Design and Implementation of Internet of Things and Cloud Based Platform for Remote Health Monitoring and Fall Detection

Ala' F. Khalifeh[1](✉), Adham Saleh[1], Mahmoud AL-Nuimat[1], Dhiah el Diehn I. Abou-Tair[1], and Nasim Alnuman[2]

[1] School of Electrical Engineering and IT, German-Jordanian University, Amman, Jordan
ala.khalifeh@gju.edu.jo
[2] The School of Applied Medical Sciences, Amman, Jordan

Abstract. With the proliferation of the Internet of Things (IoT) and cloud computing technologies in various fields, remote health monitoring and fall detection are two vital applications that are expected to adopt these technologies. This is due to the fact that it not only provides efficient way for logging the patients' health information, thus providing an electronic record for all the vital health signs that the patient is monitoring utilizing various medical IoT devices, but also it can be used to send an alert message to the healthcare personnel in case of detecting any abnormal behavior hence providing an immediate assistance. Fall detection is another important application of these technologies, where wearable sensors can be used to send an alert message to the healthcare personnel in case of detecting unpredicted fall. In this book chapter, the design and implementation of a simple and cost effective healthcare monitoring and fall detection system that utilizes of-the-shelf electronic components is provided. The system consists of a microcontroller, medical sensors and communication module that are used to collect the patients' information and send it to the cloud for further processing and analysis. Furthermore, a fall detection system that utilizes wearable sensors is proposed where it can detect unpredicted fall. One unique feature in this system that it utilizes voice recognition technology to interact with the patient after detecting a fall, thus verifying if the patient needs an assistant or not, which in turn reduces the false alarms and improves the system accuracy.

Keywords: E-health · Medical sensors · Microcontroller · Cloud computing · Remote monitoring · Fall detection · Voice recognition

1 Introduction

The advent of new technologies has reshaped many applications and scenarios in our daily life. The Internet of Things (IoT), cloud computing, smart phones and systems are few examples of these technologies. The IoT is the technology that will enable many tools, instruments, machines, equipment to communicate with the Internet for the

H. M. Fardoun et al. (Eds.): REHAB 2016, CCIS 1002, pp. 84–97, 2019.
https://doi.org/10.1007/978-3-030-16785-1_7

purpose of sending information or receiving and executing certain commands. Cloud computing is another emerging technology [1] that facilitates the process of data storing, processing, and manipulation while reducing the complexity and cost at the end devices, thus providing a set of scalable, reliable and affordable services and applications that can be easily executed by smartphones and devices that have low cost and specifications. These evolving technologies have reformed people life in many aspects, before that, no one imagined that controlling house appliances, manufacture machines and equipment, farm irrigation systems etc., can be achieved by simply launching a mobile application especially designed to achieve these functionalities. Further, remote monitoring and sensing, where different vital parameters and symptoms can be measured, logged and sent for further processing has been greatly benefited from these technologies, which in turn improved the performance of many applications, reduced the operational cost, and speeded up the decision making process [2, 3].

Electronic healthcare services named sometimes as e-health systems are expected to be totally revolutionized utilizing these technologies. This is due to the fact that current systems that are used to monitor and log the patients' health situation are neither capable to cope-up with these new technologies nor had the needed software and hardware capabilities to be fully automated and connected to the Internet. Electronic Health Record (EHR) system is one of these important services that both the patients and doctors are anxiously looking forward, although EHR services are available in some counties [4], but many countries still lack the needed technologies for delivering these services. Another important application is remote health monitoring, which has paramount importance especially for elderly people, who may have emergency cases that need an immediate assistant. For example, an IoT based system can be used by elderly people to detect an unpredicted fall and send an alert message asking for an immediate help. To address these challenges, many companies provided e-health solutions that can facilitate patients' health monitoring, yet these systems are not affordable and cost effective to many people.

In this book chapter, we provide a more detailed description of our previous work [5] where a simple, cost effective IoT based healthcare monitoring and logging system that utilizes cloud computing is proposed, where the systems design and main components in terms of hardware and software are described in details. Further, the software code used in programing the IoT devices and the cloud network are offered to the public domain as an open source project, which will facilitate its adoption and widespread implementation thus driving people toward adopting and utilizing EHR monitoring systems, which will simplify the healthcare monitoring process, and assisting the physicians in providing the most effective medications and healthcare service to their patients. Furthermore, a fall detection system that can detect an unpredictable fall, and send an alert message to the remote healthcare personnel asking for an immediate human intervention is proposed. However, in order to improve the system efficiency and reduce the false alarms, a voice controlled system that utilizes voice recognition is proposed that will be launched once a fall incident is detected, this system will open a dialogue with the patient in order to verify whether it was a real fall that needs an assistant or it just a normal movement or false alarm that do not require an immediate help. The proposed healthcare monitoring system is based on an Arduino microcontroller [6] equipped with a set of medical sensors. The sensors measure the patients' health conditions and send them through a microcontroller to a cloud platform to form permanent medical records for the patients, which can be easily accessible by patients and doctors. The fall detection system consists of a w wearable device that utilizes an Arduino microcontroller, gyroscope and accelerometer sensor, and voice recognition module, that can detect a fall and conduct a voice verification process to check whether the patient needs an assistant or not.

The contribution of the book chapter is two folds; first, the hardware components used in the proposed system is inexpensive and affordable, further, the developed code of the proposed system is provided to the public as an open source project, which will facilitate its adoption. Second, the proposed fall detection system with the voice recognition interaction functionality is relatively new. The rest of the book chapter is organized as follows: Sect. 2 provides a brief literature review for the most related papers. Section 3 describes the proposed e-health monitoring system and its implementation. Section 4 describes the proposed fall detection system. Section 5 summarizes the paper and proposes future work.

2 Literature Review

The idea of developing an IoT health monitoring system by itself is not novel, however, very few systems are made available to the public as open source projects, and proposed cost effective solutions, which may limit the ability of deployment and adoption. Furthermore, in the literature, there are several fall detection systems and algorithms, but they just detect the fall and do not interact with the patients to confirm the fall, which may result in false alarms. The authors in [7] developed a system for ambulatory use that collects and measures the patients' health conditions using a set of wireless sensors. The collected data is stored in a local database that is sent to the physician server for further analysis and processing. Furthermore, an alarming signal is sent to both patients and doctors if any abnormal health condition is detected.

Mukherjee et al. [8] developed an e-health system that can, in real-time, monitor the patients' health conditions through a set of sensors available at the patients' premises, which in turn sends the captured data to a remote server using a smartphone or communication device for further processing and analysis, where an alert message is sent to the medical doctors in case an emergency is detected. E-health monitoring systems can also include extra features and applications like patients' fall detection. The authors in [2, 3] proposed a System of systems network architecture that can be used to provide remote health monitoring for patients, who can be residing at their homes or at the hospital, and using medical IoT sensors that are connected to the remote medical center.

Falls affect millions of people every year and could cause injuries especially for the elderly [9]. In fact, falls have been considered one of the top three causes of death in elderly people. In the literature, several algorithms and systems have been proposed in the field of elderly fall detection. In [9], Rajasekaran et al. developed a system composed of a network of wearable sensors and general computing capabilities for the individual event detection, alerts, and communications with various medical informatics services. The system purpose is to provide monitoring services for elderly patients under drug therapy after infarction, data collection in some cases, and remote consultation for elderly people. Moreover, Alazrai et al. [10, 11] proposed two approaches for fall detection, one is based on depth-map video sequences based on view-invariant human activity representation [10], while the other one detects fall using anatomical-plane-based representation [11].

There are currently some fall detection systems available that detect these falls and allow the user to obtain an assistance manually or automatically if a fall occurs. Exemplary fall detection systems can consist of personal help buttons (PHBs) or environment-based and automatic detection systems and wearable sensors. For instant, Peng et al. [12] developed a fall detection system that can detect and expect a fall after analyzing the data captured by a sensor designed for that purpose. Then the fall detector system uses the captured data to adapt and fine-tune the fall detection process according to the patient physical characteristics. Commercially, many companies started providing fall detection devices as on the shelf products; for instance, AutoAlert is a sample product from Philips company, which combines accelerometers and barometric sensors with a tuned algorithm to detect many types of falls with a claimed accuracy of 95% [13]. Apple watch series 4 offers an application that detects hard fall and connect the user to the emergency with touch screen interface [14]. Khan and Hoey concluded in their review of fall detection techniques that fall detection researcher collected their data in special settings that may be not applicable in all daily life situations and environments, which limits the accuracy of the different algorithms and techniques available and reduces the performance of classifiers significantly when the dataset used in classification is different from the training data [15]. Based on that, we believe a good fall detection system needs a further assessment technique like voice orders form the user confirming the situation.

In this work, an inexpensive and open source health monitoring system is proposed, that can be used to generate an e-health record that can be used to monitor the patient health situation, the system can also be used to analyze the logged information and detect any abnormal conditions. Furthermore, we proposed a simple yet efficient fall detection system that utilizes our previous knowledge and experience in voice recognition and dialogue systems [16, 17] is proposed, up to our knowledge there is no other system proposed in the literature that combines the detection of unpredicted fall with voice recognition, such that the system verifies whether the patient needs an assistant or not utilizing a voice recognition dialogue system, which in turn reduces the possibility of sending false alarms and improves the system accuracy.

3 E-health Monitoring System Design and Implementation

In the following sections, the design and implantation details of the proposed e-health monitoring system is provided. As depicted in Fig. 1, the e-health monitoring system consists of four main parts: a set of medical sensors, microcontroller, cloud infrastructure and web-based interface.

3.1 Medical Sensors

In order to select the most vital medical sensors, a research has been conducted in exploring the most important vital signs for medical emergencies. According to John Hopkins hospital in the US [18], vital signs are measurements of the body's most basic functions. Four main vital signs are identified and are normally monitored by medical doctors. They are the body temperature, heart pulse rate, respiration rate (rate of breathing) and blood pressure. Furthermore, in order to cover as much as possible of various health conditions, more sensors than the previously mentioned can be easily integrated into the proposed system. In particular, Electroencephalography (EEG) to measure the muscles activity, Electrocardiography (ECG) to record and monitor the heartbeat, Glucometer (blood sugar level meter) and oxygen level in the blood, patient posture and position which can be used to identify anomalous body movements and their connections with some diseases, sweating rate, and finally, the heart electrical activity (Fig. 2).

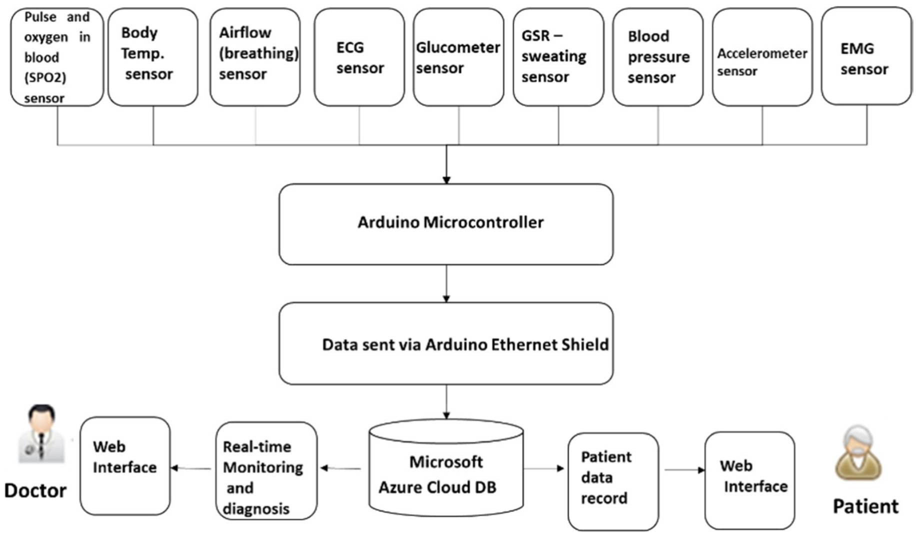


Fig. 1. E-health monitoring system architecture

Fig. 2. The various medical sensors that can be used in this platform all connected to the e-health sensors shield. The figure depicts the following sensors: (1) Blood pressure sensor, (2) Oxygen in blood sensor, (3) Glucometer sensor, (4) Body temperature sensor, (5) Electrocardiogram sensor, (6) Electromyography sensor, (7) Airflow sensor, (8) Galvanic skin response sensor, (9) Patient position sensor [19].

3.2 Sensors' Microcontroller Compliance

In order to build the proposed system, it was essential to find a set of sensors that are compatible with the utilized microcontroller and at the same time, cost effective and easy to integrate and deal with. This way, the proposed solution can be adopted by many people. Arduino microcontroller is an open source, cost effective, and easy to use microcontroller that was adopted in this work. Further, we tried to search for a company that provides medical sensors compatible with Arduino microcontroller. Our search leads us to a company that is specialized in electronic kits, among of them the e-health kit utilized in the proposed system [19] developed by Libelium Inc. This company not only provides the user with the sensors and the Arduino interfacing circuit, but also the software libraries and Integrated Development Environment (IDE) needed to read the sensors' data. We used these software utilities in building the Arduino code needed to interface with the sensors and to upload the sensed data to the cloud. The developed code as well as the one provided by the company [19] are made publicly available as an open source project via the Github portal [20].

3.3 Arduino Ethernet Shield

In order to send the medical sensors' data to the cloud, it is essential to equip the Arduino microcontroller with a proper communication module. In this implementation, an Ethernet shield that can be mounted on top of the microcontroller is used, which in turn provides Internet connectivity to the cloud. The main chip of this shield is the Wiznet W5100, which provides a network (IP) stack that can establish both TCP and UDP connections [21].

3.4 Microsoft Azure Cloud

Azure is Microsoft's cloud computing platform and infrastructure utilized in this project. It provides both Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) and supports many different programming languages and frameworks [22]. In this project, a database was created that holds the sensors' data, which contains ten tables. The main table (user access) contains the user information, i.e. Username, password as well as the e-mail address and other user information, the remaining nine tables correspond to the nine medical sensors that can be used in this platform. Each of them contains the sensor value, the user ID and a counter to give a sequential number to the entered values.

3.5 E-health Web Interface

The system usability, privacy and security of the patients' data including his/her medical records is of utmost importance. Further, a secure access is needed to be provisioned to specific doctor to monitor and check the patients' data; hence, a web based interface portal has been developed that provides both the doctors and patients with an organized easy to use web interface for the patients' medical records.

3.6 Software Implementation Details

The aim of this section is to provide more insights into the system software implementation. In particular, the pseudocodes for capturing the sensors reading, connecting with the cloud and database schema used to save the patients' sensors information are fully described. Figure 3 depicts the pseudocode used to read the sensors readings. In line 1, the sensors are initialized by checking their statuses and by initializing specialized variables needed to store the sensors readings. Line 2 defines the buffer used to store the sensors' data. Lines 3 to 5 initialize some variables used in the code, mainly the variable used to save the captured sensors values (*value*), a counter (*i*) that will go through the sensors. The number of sensors are defined by the variable *numSensors* that is known and entered to the function readingSensors(). Further, the sensors' lower and upper normal values (*thresholdLowerValues*, *thresholdUpperValues*) are input to the function that define the normal range for the captured values, notice that these variables will be used to determine whether the captured data is normal or need an attention. Lines 7 to 15 will go through all the sensors, call the sensor (*i*) specific function getSensorValue(*i*) used to capture the sensors data, check if the captured value is normal or not, and in case of having a normal value, the captured data is stored in the *sensrosBuff* array which is returned by the readingSensors() function, if the captured data is abnormal, then the reading is marked for further investigation and analysis using the function markValue().

Once the patient's data is captured and stored into the variable *sensorsBuffer*, another function cloudConnectDBUpdate() is called. The pseudocode code of this function is shown in Fig. 4. The function starts by defining a status variable and initializing it to FALSE, this variable will be used to check whether the connection to the cloud was successfully established or not. Lines 2, 3 define the variables needed to

```
sensorsBuffer = readingSensors (numSensors, thresholdLowerValues, thresholdUpperValues)
initSensor()
Define sensorsBuffer[]
Initialize value to 0
Initialize j to numSensors
Initialize i to 1
while i < j do
    value = getSensorValue(i)
  if thresholdLowerValues [i] < value < thresholdUpperValues [i]
     sensorsBuffer[i] = value
  else :
     markValue(value, i)
  end if
i = i + 1
end while
```

Fig. 3. Pseudocode used to read the sensors' readings

```
CloudConnectDBUpdate (numSensors, sensorsBuffer )
1:   Initialize status to FALSE
2:   Initialize j to numSensors
3:   Initialize i to 1
4:   sendHttpRequest()
5:   status = connectionCheck()
6:   if status == FALSE
7:        display message "connection failed"
8:   else:
9:        while i < j do
10:            updateTable (i, sensorsBuffer[i])
11:            i = I +1
12:       end while
13:  end if
14:    terminateConnection()
```

Fig. 4. Pseudocode used to connect to the cloud and update the database

go through the sensors. Line 4 is used to establish the connection with the MS Azure cloud platform, send an HTTP POST request to the MS Azure Mobile Service data API, and check whether the system is connected successfully to the MS Azure and the database or not, if connected, it will update the tables corresponding to the medical sensors reading stored in the s*ensorsBuffer* variable using the updateTable() function.

Then the connection is terminated using the terminateConnection() function. Notice that the MS Azure cloud platform provides a time/date stamps for all the recorded data, which is very useful for monitoring the patients' health conditions over a certain period.

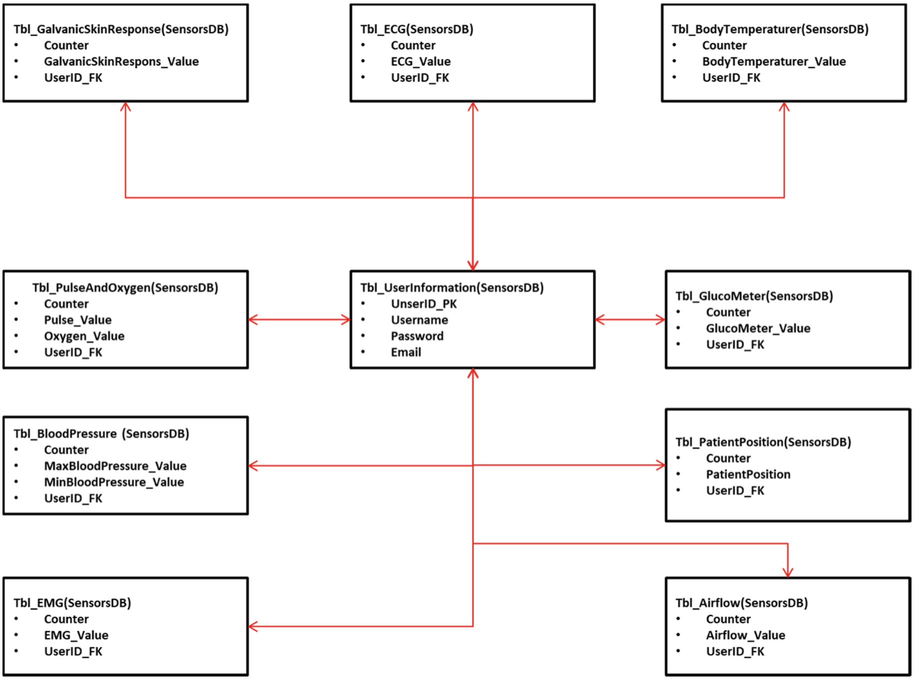


Fig. 5. Sensors' database schema

Finally, Fig. 5 depicts the database schema used to store the sensors values (SensorsDB). The database consists of several tables; the main table is the UserInformationTable, which creates a record for the patient by saving the patient basic information needed to give an access to the system and to view the records. In particular, this table saves the patient's username/password and email address. The systems also generates and saves a unique ID that is used as the database primary and foreign key. Notice that this table does not hold any private information such as the name, age, address, mobile number in order to protect the patient privacy. Further, this table is connected with the other tables that are used to hold the patient sensors' values. Notice that on each table, there is a counter field, which is incremented dynamically whenever a new record is inserted into the table.

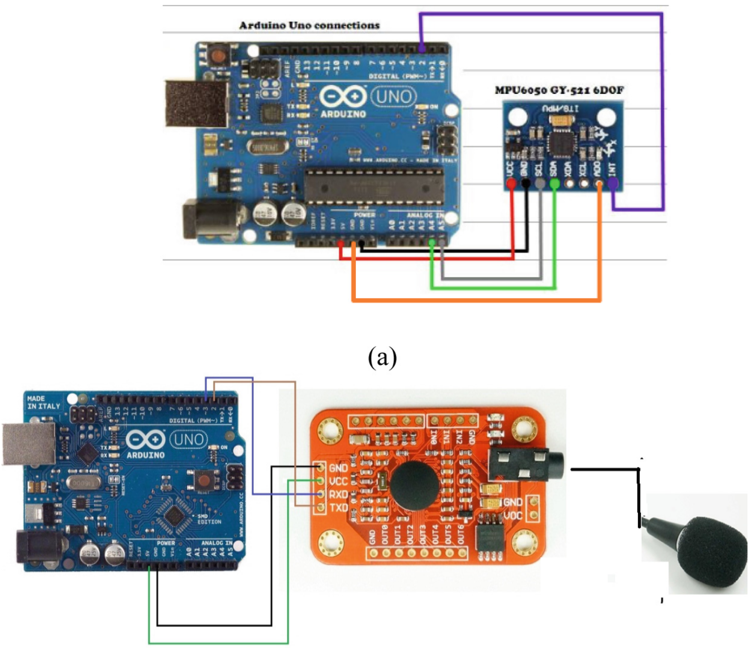
4 Fall Detection System

After discussing the health monitoring and recording system, it is also important to discuss the other proposed system that is used to detect the patient unpredictable fall and send an alert message seeking for an immediate assistant. As described earlier, this system is designed to distinguish the unpredictable fall from daily life activities and movements. As shown in Fig. 6, the proposed system consists of a microcontroller connected to two main components:

1. A gyroscope for detecting the fall, which measures the acceleration and the rotational velocity of the patients' movements.
2. A voice recognition module that allows the user to confirm whether he/she is facing an emergency condition or not.

4.1 Programming the Gyroscope and the Voice Recognition Modules

The way the system determines the patient possible fall is by measuring the acceleration and the rotational velocity of the patient movements. Several experiments that simulate the patient fall have been tested to determine the threshold values, for both the



(a)

(b)

Fig. 6. Depicts the fall-detection system which consists of (a) Arduino Microcontroller connected to InvenSense MPU-6050 gyroscope and accelerometer sensor [23], and (b) Elechouse Voice Recognition Module V3 connected to the Arduino Microcontroller from the left side, and to a microphone from the right side [24]

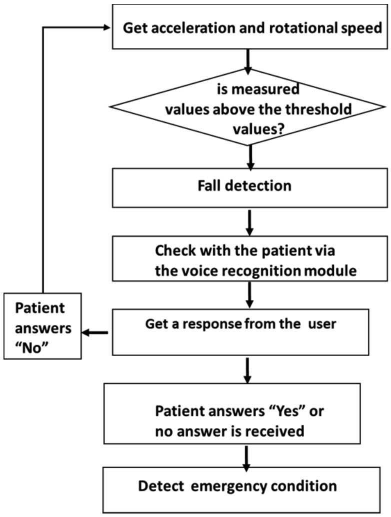


Fig. 7. A flow chart of the fall detection and emergency activation system

```
fallDetetctionVoiceRecognition ()
Initialize record = NULL
Initialize status = 0
trainVoiceRecognition()
record = saveRecords()
if voiceRecognition.clear() == 0
  display message "Recognizer cleared"
else if
  display message "No Recognition module is connected"
   EXIT
while (1)
   if  readGyro() < 3000 and readAccel() < 10000
           display message "Fall not detected"
   else if
             status = voiceRecognition()
              if (status ==1)
                    display message "Emergency is detected"
                     callEmergency();
              end if
    end if
end while
```

Fig. 8. The fall detection and voice recognition module pseudocode

acceleration and rotational velocity. When the measured values are higher than the threshold values, a fall is detected. After that, the voice activation module will start a dialogue with the patient, asking whether the patient needs an assistant or not, if the patient answers ‘Yes’, or if the module did not detect any comprehensible answer, then the system will assume that the patient needs an assistant and consequently activate an emergency call or send an alert message to the remote healthcare center, but in case the patient answered ‘No’, then the system will assume that the patient does not need any help and will not activate the automatic emergency calling process. It is worth mentioning that the automatic emergency calling and alerting process is still under further development and enhancement. Figures 7 and 8 show a flow chart of the fall detection process and the pseudocode used to program the microcontroller and the voice recognition module, respectively.

As depicted in Fig. 8, the code starts by initializing the variables (*record*, *status*) used in the voice recognition module testing, and in the fall detection, respectively. Then the recognition module is trained to match the user voice nature and characteristics using the trainVoiceRecognition() function. Lines 4 to 8 are used to check if the voice recognition module is connected properly to the microcontroller and has a valid response or not. If it is not connected, the program will exit, otherwise, it will enter in an infinite loop where it will continuously check the values of the accelerometers and Gyroscope, if they exceeded the threshold values for the normal human motion, a fall is detected and the voice recording and recognition is activated by calling the voiceRecognition() function, which will start a dialogue with the patient to check if the patient needs a help or not, if the patient answers “Yes” or provides no answer for a specific duration of time, and if the recorded message was recognized by the voice recognition module, then the system will classify this fall as a real one and will call the emergency, if the patient answers “No”, then the system will consider this fall or movement as a normal one and will not call for an assistance.

5 Conclusion and Future Work

In this paper, we present our proposed e-health monitoring system which consists of a set of sensors connected to a microcontroller, where the patients’ sensed vital data are uploaded to a cloud platform for recording and remote assessments. Further, we proposed a fall detection system based on a gyroscope and voice recognition module that can interact with the patient if a fall is detected. We made all the developed source code available to the public and used inexpensive and affordable off-the shelf hardware components, which should facilitate the system adoption and implementation. As a future work, we are still improving the fall detection system and the voice recognition accuracy to make it more robust against surrounding noise. Further, we are working on leveraging and integrating our previous work focused on designing a mobile application that can be used as a data mule [25] for the patient health record especially when an Internet connect may not exist. Another potential feature that can be added to the fall detection system is the one proposed in [26], which utilizes the patient position while providing healthcare monitoring service. This feature can be integrated into the proposed fall detection system, thus issuing an alert message to the closest healthcare personnel to the patient location. Working on larger dataset collected from more than one researcher to better understand and cover the unseen falls as recommended by [32] may improve the performance of the systems and algorithms. Privacy in IoT and cloud environment [27] is another important aspect that should be addressed in this proposed framework, where anonymization techniques and data hiding can be used to improve the privacy of the patients who are using this system. Moreover, we are in the process of developing an expert system that can analyze the patients' data, detect, and predict any potential health problems, so the patient will be notified automatically.

Acknowledgment. The authors would like to thank the German Jordanian University for funding this project through the Grant No. GP (13/2015).

References

1. Kiblawi, T., Khaliferh, A.: Disruptive innovations in cloud computing and their impact on business and technology. In: 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), September, pp. 1–4. IEEE (2015)
2. Khalifeh, A., Obermaisser, R., Abou-Tair, D.E.D.I., Abuteir, M.: Systems-of-systems framework for providing real-time patient monitoring and care. In: Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare, May, pp. 426–429. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2014)
3. Obermaisser, R., Abuteir, M., Khalifeh, A., Abou-Tair, D.E.D.I.: Systems-of-systems framework for providing real-time patient monitoring and care: challenges and solutions. In: Fardoun, H.M., Penichet, V.M.R., Alghazzawi, D.M. (eds.) REHAB 2014. CCIS, vol. 515, pp. 129–142. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48645-0_12
4. Ludwick, D.A., Doucette, J.: Adopting electronic medical records in primary care: lessons learned from health information systems implementation experience in seven countries. Int. J. Med. Inform. **78**(1), 22–31 (2009)
5. Khalifeh, A., Saleh, A., AL-Nuimat, M., Abou-Tair, D.E.D.I.: An open source cloud based platform for elderly health monitoring and fall detection. In: Proceedings of the 4th Workshop on ICTs for improving Patients Rehabilitation Research Techniques, October, pp. 97–100. ACM (2016)
6. Arduino microcontroller official website. http://www.arduino.cc. Accessed Oct 2018
7. Nita, L., Cretu, M., Hariton, A.: System for remote patient monitoring and data collection with applicability on e-health applications. In: 2011 7th International Symposium on Advanced Topics in Electrical Engineering (ATEE), May, pp. 1–4. IEEE (2011)
8. Mukherjee, S., Dolui, K., Datta, S.K.: Patient health management system using e-health monitoring architecture. In: 2014 IEEE International Advance Computing Conference (IACC), February, pp. 400–405. IEEE (2014)
9. Rajasekaran, M.P., Radhakrishnan, S., Subbaraj, P.: Elderly patient monitoring system using a wireless sensor network. Telemed. e-Health **15**(1), 73–79 (2009)
10. Alazrai, R., Momani, M., Daoud, M.I.: Fall detection for elderly from partially observed depth-map video sequences based on view-invariant human activity representation. Appl. Sci. **7**(4), 316 (2017)

11. Alazrai, R., Zmily, A., Mowafi, Y.: Fall detection for elderly using anatomical-plane-based representation. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, August, pp. 5916–5919. IEEE (2014)
12. Peng, Y., Jin, S., Koninklijke Philips NV: Fall detection system (2013). U.S. Patent 8,381,603
13. AutoAlert: Automatic Fall Detection by Philips Lifeline. https://www.lifeline.philips.com/medical-alert-systems/fall-detection.html. Accessed Oct 2018
14. Use fall detection with Apple Watch Series 4. https://support.apple.com/en-jo/HT208944. Accessed Oct 2018
15. Khan, S.S., Hoey, J.: Review of fall detection techniques: a data availability perspective. Med. Eng. Phys. **39**, 12–22 (2017)
16. Darabkh, K.A., Khalifeh, A.F., Jafar, I.F., Bathech, B.A., Sabah, S.W.: A yet efficient communication system with hearing-impaired people based on isolated words of arabic language. IAENG Int. J. Comput. Sci. **40**(3), 183–192 (2013)
17. Khalil, R.T., Khalifeh, A., Darabkh, K.A.: Mobile-free driving with Android phones: system design and performance evaluation. In: 2012 9th International Multi-Conference on Systems, Signals and Devices (SSD), March, pp. 1–6. IEEE (2012)
18. Johns Hopkins Medical Center website. http://www.hopkinsmedicine.org/healthlibrary/conditions/cardiovascular_diseases/vital_signs_body_temperature_pulse_rate_respiration_rate_blood_pressure_85,P00866/. Accessed July 2016
19. e-health Sensor Platform V2.0 for Arduino and Raspberry Pi [Biometric/Medical Applications]. https://www.cooking-hacks.com/documentation/tutorials/ehe. Accessed July 2016
20. Github website. https://github.com/EhealthPlatform. Accessed July 2016
21. Arduino Ethernet Shield. https://www.arduino.cc/en/Main/ArduinoEthernetShield. Accessed July 2016
22. Microsoft Azure Cloud. http://azure.microsoft.com/en-us/overview/what-is-azure/. Accessed July 2016
23. MPU-6050 Accelerometer and Gyro. http://playground.arduino.cc/Main/MPU-6050. Accessed July 2016
24. Voice Recognition Module V3. http://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf. Accessed July 2016
25. Al-Tamimi, A.K., Khalifeh, A.: Mobile mules: modular e-health information synchronization framework. In: 2014 8th International Symposium on Medical Information and Communication Technology (ISMICT), April, pp. 1–5. IEEE (2014)
26. Hababeh, I., Alouneh, S., Khalifeh, A.F.: A position aware mobile application for e-health services. In: 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), January, pp. 144–148. IEEE (2016)
27. Abou-Tair, D.E.D.I, Büchsenstein, S., Khalifeh, A.: A privacy preserving framework for the internet of things. In: 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), June, pp. 27–31. IEEE (2018)