Generalised and Versatile Connected Health Solution on the Zynq SoC

Dina Ganem Abunahia[1], Hala Raafat Abou Al Ola[1], Tasnim Ahmad Ismail[1], Abbes Amira[1(✉)], Amine Ait Ali[2], and Faycal Bensaali[2]

[1] Department of Computer Science and Engineering, Qatar University, Doha, Qatar
da1105224@student.qu.edu.qa, ha1104576@student.qu.edu.qa, ti1207239@student.qu.edu.qa, abbes.amira@qu.edu.qa

[2] Department of Electrical Engineering, Qatar University, Doha, Qatar
amine.aitsiali@qu.edu.qa, f.bensaali@qu.edu.qa

Abstract. This chapter presents a generalized and versatile connected health solution for patient monitoring. It consists of a mobile system that can be used at home, an ambulance and a hospital. The system uses the Shimmer sensor device to collect three axes (x, y and z) accelerometer data as well as electrocardiogram signals. The accelerometer data is used to implement a fall detection system using the k-Nearest Neighbors classifier. The classification algorithm is implemented on various platform including a PC and the Zynq system on chip platform where both programmable logic and processing system of the Zynq are explored. In Addition, the electrocardiogram signals are used to extract vital information, the signals are also encrypted using the Advanced Encryption Standard and sent wirelessly using Wi-Fi for further processing. Implementation results have shown that the best overall accuracy reaches 90% for the fall detection while meeting real-time performances when implemented on the Zynq and while using only 48% of Look-up Tables and 22% of Flip-Flops available on chip.

1 Introduction

According to the UN World Health Organization statistics, the percentage of elderly will keep increasing to reach as much as twice the percentage of children in year 2050 (32% vs. 16%) [1]. As a result, the demand for medical attention is increasing. This is mainly because around one third of the elderly population over the age of 65 falls each year, and the risk of falls increases proportionately with age. Falls happen more frequently among elderly due to the following reasons: (1) Cerebrum degeneration or some diseases causing step imbalance; (2) Side effects of taking medications for people with chronic disease might cause dizziness and slow movements; (3) Health problems that might cause people to faint temporarily; and (4) Poor lighting environments, slippery floors and objects on the way of movement [13]. Other groups require special attention, such as Diabetics, who are more likely to get low blood glucose causing fainting, spinal muscular

Y. Bi et al. (eds.), *Intelligent Systems and Applications*,
Studies in Computational Intelligence 751,
https://doi.org/10.1007/978-3-319-69266-1_22

atrophy patients who have movement inconsistency, neurological patients, like Parkinsons patients who suffer from imprecise movement, and cardiovascular patients who can have heart attacks and collapses. Hence, the necessity of developing integrated systems for healthcare has increased worldwide. This need is managed according to world-class standards to improve the health of the worlds population, and meet the needs of existing and future generations, and provide a healthy and lengthy life for all citizen to a greater extent. All health services will be accessible to the entire population. This work aims to develop a reconfigurable connected health platform using Zynq System on Chip (SoC) [16], in order to be used either indoors or in an ambulance environment to equip them with health monitoring technologies. The proposed wireless system deploys Bluetooth connectivity to establish communication between a prototyping board equiped with the Zynq SoC, and a Shimmer [3] wearable device that acquires both acceleration and Electrocardiogram (ECG) signals. The system is made of several stages: data acquisition using the Shimmer sensing device, data processing and analysis based on the Zynq SoC device, and the last stage is concerned with alerting and sending the encoded medical report to the doctor in case of a risk. The processing stage involves the implementation of k-Nearest Neighbors (KNN) classifier for fall detection using the Shimmer accelerometer data as well Advanced Encryption Standard (AES) and feature extraction using the ECG data. The structure of the remaining parts of the chapter is as follows: Sect. 2 describes the literature review and related work, Sect. 3 gives an overview of the proposed system, and Sect. 4 discusses the software and hardware implementation. Section 5 is concerned with the results and analysis. Section 6 concludes the chapter.

2 Literature Review

The area of connected health development systems for fall detection has been intensively searched. These systems can be implemented using different techniques, such as vision based falling detection. In this technique, some used special sensors as in [6], and others deploy surveillance systems as in [4] and [12]. In [2], an Omni camera was used to detect falling events. A new approach was proposed in [8] in which a MapCam (Omni camera) is used along with the personal information of each individual being captured on the camera which enhances the percentage of accuracy. Another technique is using ambience based devices which endeavor to fuse audio and visual data, besides sense through vibrating data. Image and video sensing can be achieved using multiple approaches: one method is by using signal strength measurements to track the estimated location of the user [7]. Another method is by extracting wavelet-based features from raw sensor and apply them to a TEO-based sound activity detector [14]. The last approach is uses wearable devices which are divided into two categories: motion based and posture based devices which are based on following [15]: Motion sensing method using an accelerometer and location sensing method using both accelerometer and gyroscope.This approach [11] uses sixMTw sensors, where each unit contains three tri-axial devices: accelerometer, magnetometer and gyroscope. Machine learning techniques based classifiers to distinguish between fall and daily life activities has been used. On the other hand, the system in [10] uses wearable camera and accelerometer for fall detection. It combines gradient local binary pattern features with edge orientation histograms in order to provide higher sensitivity. In [15] the proposed fall detection system uses a wearable device of single tri-axial accelerometer, and an algorithm that is based on thresholds of summing acceleration and rotation angle information. The summation acceleration is used as the first step to distinguish between high intensity movements from others. Moreover, there has been research efforts to accelerate some of the algorithms on hardware, such as implementation on heterogeneous computing platforms. For instance, a fall detection application on a heterogeneous computing platform, Zynq- 7000 SoC was deployed in [9]. The proposed solution in this system aims to use the power of the ARM cortex A9 processor of the Zynq platform together with the OpenCV libraries to achieve an efficient solution in terms of power consumption and execution time of the fall detection algorithm deployed in the system. The system design will be partitioned between the FPGA existing in the Zynq SoC, the Cortex A9 processor and the Graphical Processing Units (GPU) that will be also deployed for computer vision.

3 System Design

The designed system consists of three stages: data acquisition using the Shimmer sensor, data processing and analysis on the Zynq SoC prototyping board and finally alerting system. The system uses three hardware platforms: the Shimmer wireless healthcare sensing device, the NI myRIO Zynq SoC prototyping board, and a PmodBT2 UART Bluetooth module. The overall system is described in Fig. 1. The data acquisition phase uses an MMA7260qt 3-axis accelerometer to acquire acceleration as well as ECG Sensors to measure the heart muscle electrical activity. Both sensors are integrated into the Shimmer sensing device which is placed on the users chest. The ECG electrodes are connected to the right arm, left arm, right leg, and left leg from one side, and on the white, black, green, and red channels of the Shimmer from the other end respectively. A Bluetooth module is used to establish a connection between the Shimmer device and the Zynq SoC prototyping board, through this connection, the signals are transferred via Bluetooth to the Zynq SoC prototyping board using the integrated RN-42 Bluetooth module inside the Shimmer as a sender, and the Bluetooth module as a receiver. Data processing and analysis is performed on the Zynq SoC prototyping board using LabVIEW as a programming software environment. If an abnormality is detected, the alerting system is launched to check if the detection is a false alarm or if the user needs help. This system also helps to assess the state of the user after the fall. Finally, if a real fall is detected, an email is sent using Wi-Fi connection with an attached medical report to the health care providers such as hospitals or ambulances.

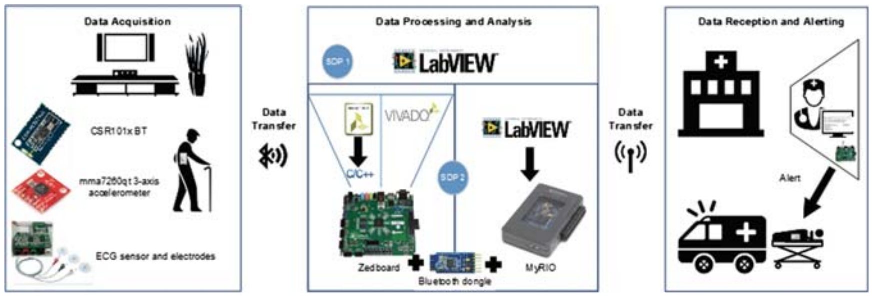

Fig. 1. System overview

4 System Implementation

The proposed solution consists of two main hardware platforms, the Shimmer sensing device and the Zynq SoC prototyping board which is myRIO. The Shimmer is used to acquire tri-axial acceleration and ECG signals from the user and then those signals are sent in real time via Bluetooth connection to the myRIO board which is located in the users side. This prototyping board has the implemented algorithm for fall detection and other algorithms used for ECG analysis and processing on the programmable logic. If the fall is detected, myRIO will send an alert via Wi-Fi to the doctors smart device. This email consists of a medical record about the information of the fall, as well as, ECG signals both encoded using AES. Consequently, the medical care can take action to reduce the risk of the fall or any related risks that might occur. The flowchart in Fig. 2 summarizes the different steps.

The system is implemented in multiple phases; the first phase involves software implementation, in this phase all the processing and testing is performed on a PC. The second phase involves the implementation of all algorithms on hardware that is capable of optimizing PC's performance. For that purpose, NI myRIO is chosen since it has the Zynq SoC which includes the ARM processor where simple code will be executed and an FPGA based programmable logic where computationaly intensive algorithms will be executed. The system setup is illustrated in Fig. 3.

4.1 Software Implementation

The software implementation where results are initially verified is important for building the system. The algorithms have been tested separately then they were combined to build the entire system. After the system is implemented and tested on a computer, computationally intensive parts are identified and implemented on the PL side of the Zynq SoC prototyping board for hardware acceleration while the remaining parts are implemented on the PS side.

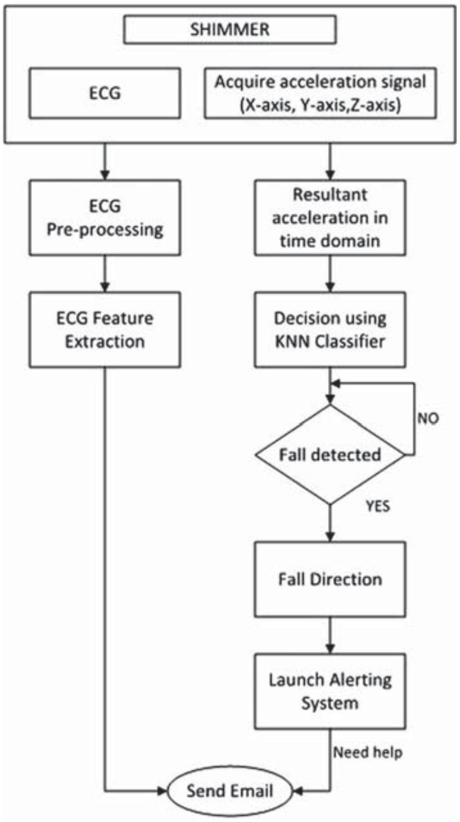

Fig. 2. Solution steps

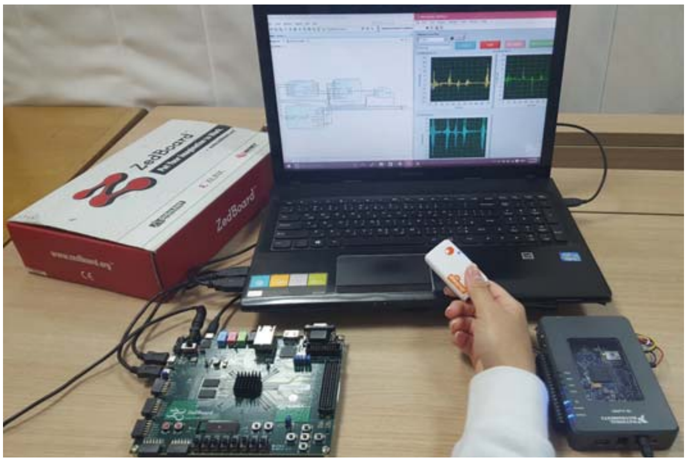

Fig. 3. System setup

4.1.1 Fall Detection

A fall detection database is constructed using the acquired tri-axial acceleration signals from the Shimmer platform. The database samples are collected to construct a base and reference for the algorithm, through including different scenarios to improve the accuracy of the system. As a result, the database consists of twelve subjects that perform various daily activities and falling scenarios. Each of the twelve people performed multiple activities. The first type of activity which is “Activity of Daily Life (ADL)” involves four subtypes which are “Jumping”, “Jogging”, “Picking something from the ground” and “Running”. This specific choice of ADL scenarios is made due to the fact that they produce high acceleration resultant compared to other daily activities. Thus, they can act as a separation line for distinguishing between falling and daily activities. This decision was made after a series of trials. The seconfd type of activity which is “Fall” involves four subtypes as well which are: “Back fall”, “Left fall”, “Right fall” and “Front fall”. For each person, three hundred samples are acquired for each of the eight scenarios, then the resultant acceleration is calculated for each tri-axial sample according to equation (1).

$$\text{Resultant acceleration} = \sqrt{Ax^2 + Ay^2 + Az^2} \tag{1}$$

Among these three hundred resultant acceleration, only the three highest values are taken and added to the database. This is applied for all of the subjects which lead to: 12 people x 8 scenarios x 3 resultant acceleration extracted = 288 database samples. The Shimmer device was placed in the same position for all subjects which is the chest. Furthermore, all experiments were done in the same environment to achieve the requirements of a reliable system. The provided system should distinguish between ADL and fall scenarios. Figure 4 represents different experiments demonstrating different fall and ADL scenarios together with the acquired acceleration signals.

The Shimmer platform can generate 3-axis acceleration using 3-axial accelerometer. The acquired signals are transmitted via Bluetooth to the myRIO

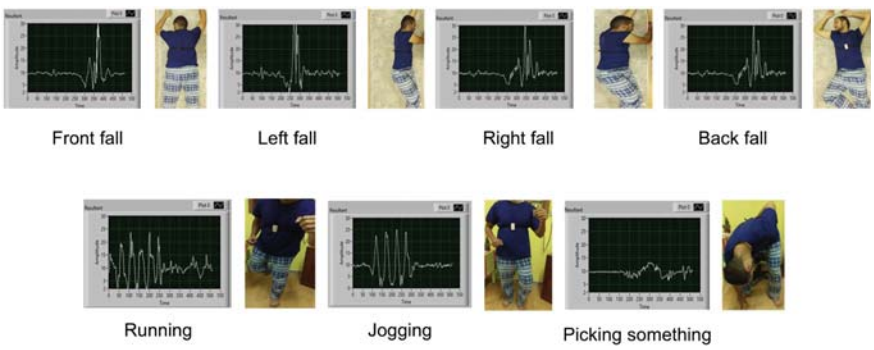

Fig. 4. Acceleration signals for ADL and various fall directions

for analysis in LabVIEW. Through LabVIEW application, the resultant of the three axis acceleration (Ax, Ay, Az) is calculated using Eq. (1) and then the resultant acceleration is sent to a sub vi that implements the KNN classification algorithm. The KNN classification is chosen to implement the fall detection algorithm because it requires less computation time and less data. However, it provides a high percentage of classification accuracy. The purpose of the KNN algorithm is to use a database in which the data points are separated into distinct pre-defined classes to predict the class of a new data point coming in. Predicting the class of the new data point can be achieved using one of the following distance functions: (2), (3) or (4).

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{2}$$

$$\sum_{i=1}^{k}|x_i - y_i| \tag{3}$$

$$(\sum_{i=1}^{k}(|x_i - y_i|)^q)^{\frac{1}{q}} \tag{4}$$

The design of the proposed system uses the Euclidean distance for detecting the class of a new sample. The KNN classifier sub vi takes the sample resultant from time domain then calculates the distance between the new sample and all distance samples stored in the database and then all calculated values are stored in array X that is sorted ascendingly. After storing and sorting, the K smallest values of the distances array are selected and then the class of the sample is predicted depending on the class of the K selected samples. The simplified KNN algorithm pseudo code is shown in Fig. 5 where X are the training data stored in the database; Y presents class labels of X also stored in the same database and x is the new data point to be predicted.

In order to specify the fall direction of the user, several experiments were done to see the effect of the fall direction on of x- axis, y-axis, and z-axis values. Then a decision was made by choosing suitable values of x-axis and z-axis since the orientation of these two axes are affected with different directions: front, back, left and right. The process of detecting the direction of the fall is summarized in Fig. 6.

4.1.2 ECG Analysis

ECG features provide information about the heart rate, the condition of tissues within the heart, the conduction velocity as well as various abnormalities. It provides evidence for the diagnosis of cardiac diseases. The ECG analysis system contains several stages, starting with electrodes setup and wiring, to ECG acquiring, processing and analysis, and finally, results demonstration. Figure 7 shows an overview of the ECG analysis system.

Similar to the acceleration signals, Shimmer LabVIEW library has been used to acquire ECG signals. First of all, the sampling frequency is assigned to 51.2 Hz,

Algorithm 1: k-Nearest Neighbor

```
Classify(X, Y, x) // X: training data, Y: class labels of X, x: unknown sample
for i= 1 to m do
    Compute distance d (Xi, x)
End for
Compute set I containing indices for the k smallest distances d (Xi, x)
Return majority label for {Yi where ϵ I}
```

Fig. 5. Algorithm 1

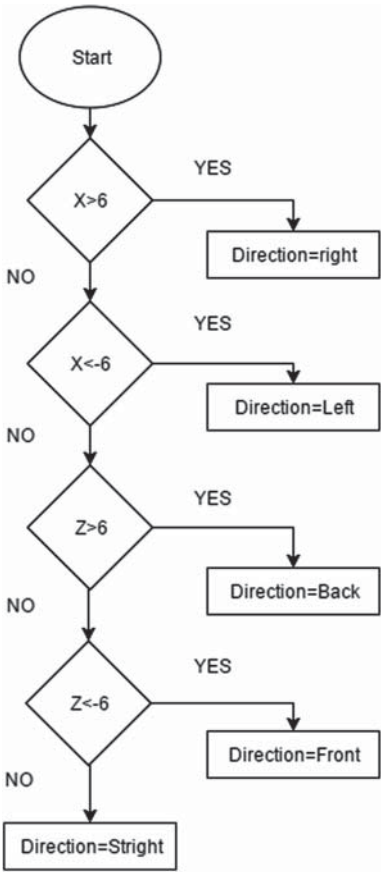

Fig. 6. Flowchart for fall direction

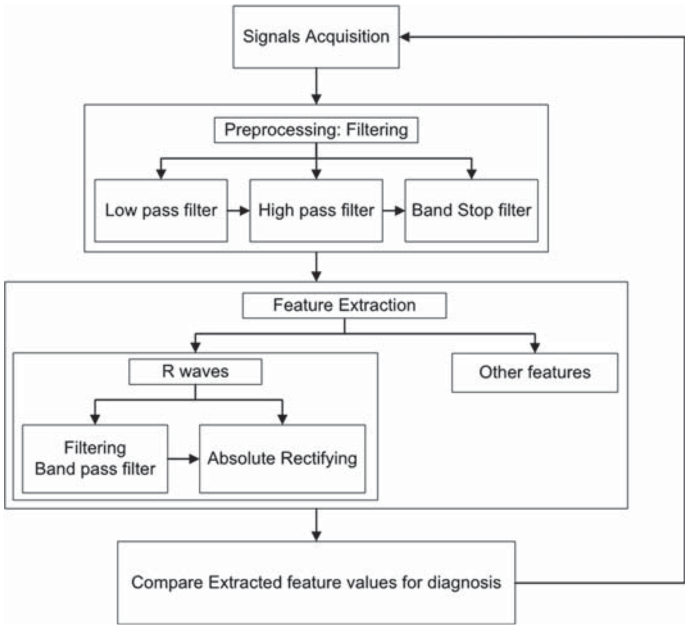

Fig. 7. ECG analysis system overview

which is suitable for the ambulatory ECG monitoring, i.e. monitor the heart while doing normal daily life activities. The Analog to Digital Converter (ADC) output of each ECG channel uses signed 24-bit digital format. The relationship between the ECG signal in mVolts and the ADC output is given by the Eq. (5).

$$\text{ECG Signal in mVolts} = \frac{((\text{ADC Output} - \text{ADC Offset}) \times \text{ADC Sensitivity})}{Gain} \tag{5}$$

Considering that the ADC Output is measured, and the ADC Sensitivity is described in equation (6):

$$\text{ADC Sensitivity} = \frac{V_{ref}}{\text{ADC Max}} = \frac{2420 \text{ mVolts}}{2^{23} - 1} \tag{6}$$

The values for the gain and ADC offset 5 must be inserted in Eq. (5). The gain is software configurable, whereas the nominal value of the ADC offset is 0. The channel inputs are connected, to calculate the offset of each channel. For example, for channel 1 (lead II), the mean value of the ADC output is calculated with the RA and LL inputs connected. This can be done if the uncalibrated data is saved into a file and the mean value is calculated for each channel. The following step after signals acquisition and extracting is filtering. A band stop filter is needed if the signal is experiencing interference from mains electricity. A 50 Hz frequency eliminator is required. Also, a high pass filter is necessary to eliminate low-frequency components of the signal. The cut-off frequency of

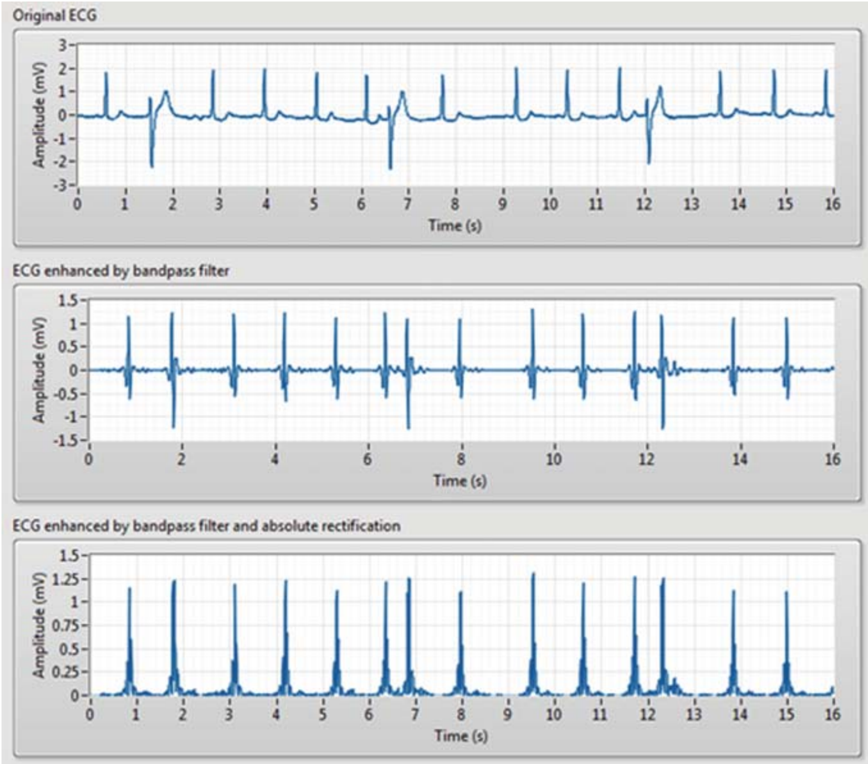

Fig. 8. ECG signal pre-processing for R waves detection

0.5 Hz is suitable for long term ECG monitoring, whereas a cut-off frequency of 0.05 Hz is recommended for ECG diagnostic. In the proposed work, the cut-off frequency is a set as a choice according to the end user application. QRS detection algorithms are the foundation of ECG analysis, and can be used to estimate the heart rate. The most widely used real-time QRS detection algorithm is The Pan-Tompkins Algorithm. First, the signal is passed through a low pass and a high pass filter to reduce the influence of the muscle noise, the baseline wander, the T-wave interference, and the power line interference. This step is accomplished in the pre-processing phase. After filtering, the signal is differentiated to provide the QRS slope information; then the signal is squared to make all data points positive and to emphasize the higher frequencies. After squaring, the signal is integrated using sliding window integration to obtain waveform feature information. The size of the sliding window depends on the sampling frequency. The last step is to adjust a threshold that differentiates between peaks of the signals, and the absence of peaks. ECG feature extraction firstly starts by R-waves detection, then extracting other features. For normal ECG signals, they can be easily detected. However, the process becomes harder for heart patients where their ECG signals are abnormal. As a result, the ECG feature extraction starts by signal enhancement, which contains two stages: filtering and rectifying. Initially, the signal is filtered using a bandpass filter, to allow signals within a specific range of frequencies to pass, and prevent others. Since R-waves of human ECG usually ranges between 10–25 Hz, these values were used as cutoff frequencies. Then, the filtered signal is rectified using absolute rectification. Figure 8 shows the processing result of an ECG signal, with negative R values, and large T-waves amplitudes. It can be noticed that after enhancement all beats can be easily detected, and heart rate variability analysis can be easily done.

Fig. 9. Heart rate calculation

The heart rate (HR) in beats per minute (BPM) is calculated from the ECG using the R waves (part of the QRS complex defined above). The characteristics of the R peak are determined by training for an interval of a few seconds; these characters are used to calculate the heart rate. S1 is the sample number of the first R-wave detected, S2 is the sample number of the last R-wave detected, and NR represents the total number of R-waves between S1 and S2 inclusively. By knowing the sampling rate in Hz, which is Fs. These values can be substituted into the Eq. (7).

$$HR(BMP) = 60 \times \frac{F_s \times (N_R - 1)}{S_2 - S_1} \tag{7}$$

Figure 9 illustrates the heart rate calculation in the system. The ECG signal results from lead I (RA-LA) and the heart rate calculated which is 95.0103.

4.1.3 Alerting System

In the proposed solution, fall detection and minimizing false alarms are two major concerns. It is critical to confirm accurate detection and correct behavior of the system. Therefore, a complete alerting system is added to the functionality of the proposed system. Mainly to notify the caregivers about any abnormal activity with the user, and to minimize false alarms even further. Generally, falls can be categorized into three conditions:

Condition 1: Normal Condition of Fall

This is the condition when a fall is detected, but the elderly stays conscious. In this case, the user can turn off the alarm by pressing the "I am fine" button. This condition is crucial for eliminating false alarms, such that if the system detects a fall in ADL, the user can inform the system that it is not a fall.

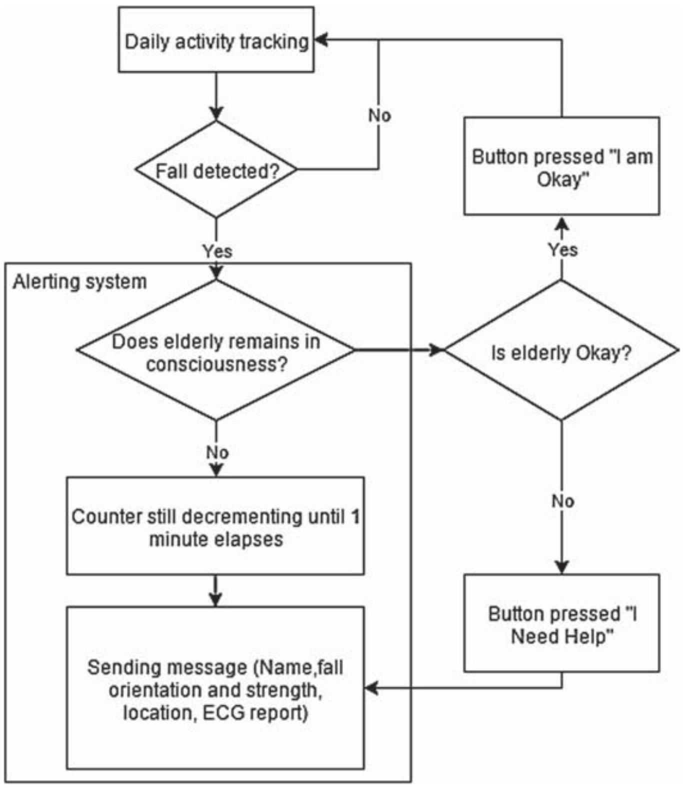

Fig. 10. Alerting system flowchart

Condition 2: Critical Condition of Fall

This condition results when a fall happens, and causes injuries, but elderly stays conscious. In this case, the user can press the button "I NEED HELP" to alert that the elderly needs help, and send the email.

Condition 3: Emergency Condition of Fall

This is the condition when the fall causes serious injuries, and the elderly is in fatal condition. As a result, an alert is sent to the emergency unit to provide emergency response and assistance as fast as it can.

For alerting system implementation, Data Dashboard for LabVIEW was used. This application runs on Android/IOS mobile phones and tablets. The system consists of two timers and two switches, and it follows the flowchart illustrated in Fig. 10. If a fall was detected, the first timer runs for 30 s to make sure that the user is conscious, if he/she did not respond within 30 s, an alert is sent immediately. If the user replied within the timeout, the second timer runs for 30 s to check if the user needs help. If the user chose "I am fine", it is more likely a false alarm that was stopped to reach the medical assistance.

4.2 Hardware Implementation

4.2.1 Implementation on Zedboard

The KNN software implementation gave a reasonably high detection rate within a short response time. The accuracy is expected to remain high with a considerable decrease in execution time if the fall detection algorithm is implemented on hardware. The Zynq SoC is chosen for hardware implementation since it contains both a processor system where software implementation is deployed and a programmable logic where hardware implementation and optimization are performed. The key element of the hardware implementation stage is to partition the system functionalities between hardware and software, and to decide on the appropriate interface between the two partitions. This decision is the most important step in hardware/software co-design approach since the system performances are directly dependent on it. Generally, software design on the Precessing System (PS) will be used to implement general purpose sequential processing tasks, an operating system, user applications and Graphical User Interfaces (GUIs), while computationally intensive data flow parts of the design are more suitably realized in the Programmable Logic (PL). For this project, the PS portion is responsible for sensor interfacing and data acquisition. Additionally, the ECG processing and encryption are implemented on the PS. Alternatively, the PL portion is responsible for KNN algorithm implementation to detect falls. The Advanced eXtensible Interface (AXI) interface handles the PS/PL communication to send acceleration signals from the PS to the PL for processing. However, the KNN algorithm is implemented on the ARM processor initially for testing and validating, after the design is verified, it is implemented on the PL. Vivado

Design Suite is used to implement the system on the Zedboard. Generally, the implementation on the Zedboard is partitioned into two sections: implementation on the PS, and the implementation on PL. PS implementation will be used to implement general purpose sequential processing tasks, an operating system, user applications and GUIs, while computationally intensive data flow parts of the design, and any software algorithms which exhibit significant parallelism can be strong candidates for implementation in PL. Figure 11 illustrates the design flow for a hardware implementation on the Zynq SoC. Through Vivado High Level Synthesis tool (HLS), an IP implementing the KNN algorithm has been designed and optimized. First a C-code is written to implement the KNN classifier for fall detection, it takes a form of a "predict" function. In addition, a C test bench is written such as it takes the main C function executing the "predict" function to self-check the results. Figure 12 shows the hardware block design of the overall design which includes the processing system, the AXI connections, and the KNN IP with the predict function.

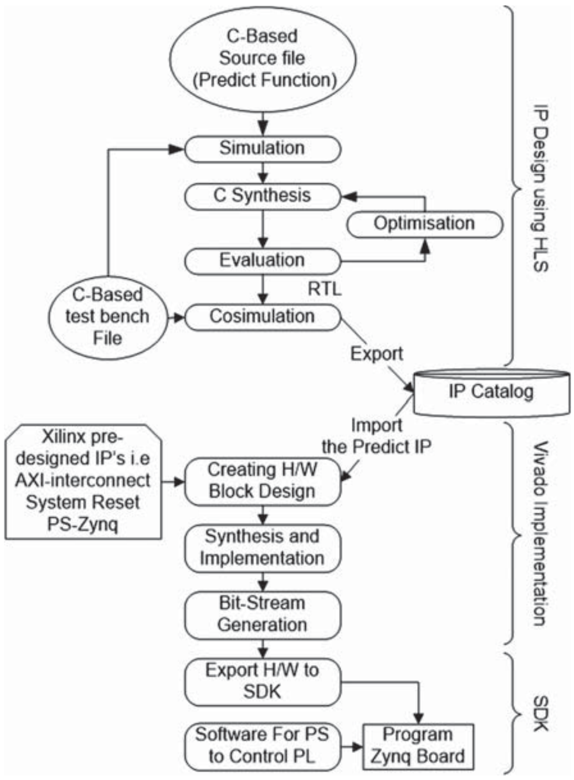

Fig. 11. Hardware implementation design flow

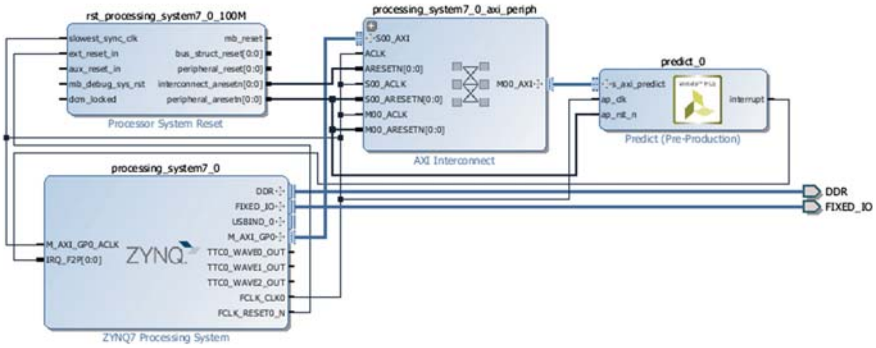

Fig. 12. Hardware block design

4.2.2 Implementation on MyRio

Even though the implementation results on the Zedboard are promising, implementing the remaining subsystems of the project is challenging. This is mainly because the Zedboard is only programmable with C/C++ and VHDL, while the software implementation of the proposed system is performed using LabVIEW. As a result and to simplify the task, NI myRIO Zynq SoC prototyping board is selected to implement the entire system. The ARM PS portion is responsible for Shimmer-myRIO communication and ECG processing, whereas the FPGA PL is responsible for KNN algorithm implementation. In this approach acceleration signals and ECG signals are acquired using LabVIEW program and transmitted to the host computer via Bluetooth. Then shared variables are used to send the values wirelessly to the myRio board using WiFi. Data was analyzed and processed on the Zynq SoC using ARM processor. As a result of the dual transmission before signals processing, data transmission consumed much time in case of weak Wifi signal causing fall detection to be delayed for about 30 s, and thus this approach was not sufficient for the application. In that matter, the

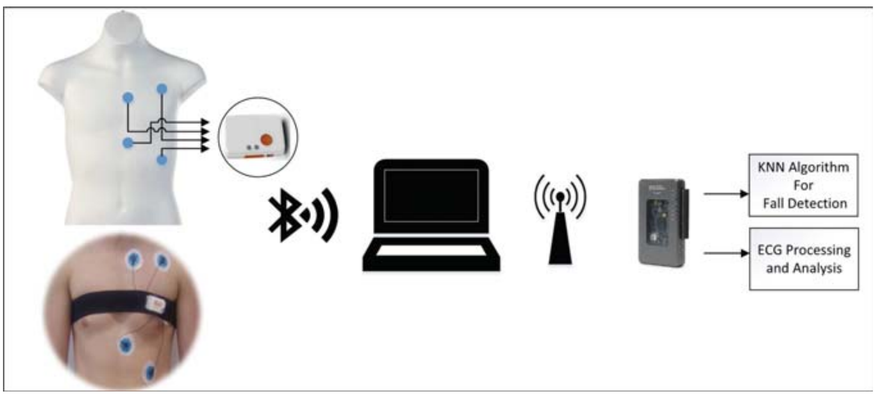

Fig. 13. System implementation using myRIO

design was improved to include direct communication between the Shimmer sensing device and the myRIO board using the pmodBT2 Universal Asynchronous Receiver Transmitter (UART) Bluetooth module. The bluetooth communication assured preserving time for data processing to satisfy the real-time performance as illustrated in Fig. 13.

5 Results and Analysis

5.1 Software Implementation

The testing phase for fall detection involves eleven subjects who participated in performing the same scenarios done with first training group. This group's results are used to evaluate the system performance. With the aim of achieving high accuracy, it is necessary to choose a suitable value of K and data training percentage, this process of trying different values of K and different percentages is referred to as cross-validation. Cross-validation can be done as follows: Fix the training percentage and vary the value of k (k = 3, 4 and 5) then fix the value of K and vary the training percentages (20, 30, and 50%)

The accuracy results for each case are shown in Table 1. After testing the system with different scenario cases, it is shown that some values of K and training percentages provide more accurate results than others. For instance, in the case when K = 5, and training percentage = 30%, the system provides the least overall accuracy of 77.14%. Whereas in the case when K = 3, and training percentage = 30%, the system provides the most overall accuracy of 90.00%. However, it is evident that the accuracy for almost all cases is below 90%. As a result of that, the system needs to be improved to increase the accuracy, this can be done in several steps: First, the database should be increased to involve at least twelve people, second, wavelet domain analysis can be used before calculating the resultant acceleration.

Table 1. Accuracy results of testing the algorithm with different values of k and different training percentages

Training percentage		K = 3		K = 4		K = 5	
		Fall	ADL	Fall	ADL	Fall	ADL
50%	Accuracy (%)	82.14	88.10	82.14	88.10	78.57	90.48
	Overall (%)	85.7		85.71		85.71	
30%	Accuracy (%)	82.14	95.23	75	92.86	50	95.23
	Overall (%)	90		85.71		77.14	
20%	Accuracy (%)	67.86	88.10	75	90.48	82.14	90.48
	Overall (%)	80		84.29		87.14	

5.2 Hardware Implementation

Vivado HLS tool supports several optimization techniques, such as loop unrolling, array portioning, and pipelining. The "Unroll Loop" approach provides dedicated hardware resources for each iteration of a loop, hence allows iterations of a given loop to be executed in parallel. The second approach, which is "Array Partitioning" allows each entity of an array to have its own data ports rather than considering it as one array entity and limiting data ports. The last approach is "Pipeline" which is applied in the system to allow pipelining of instructions and sub-functions, hence optimize the system [5]. The C code is compiled, then synthesized and a report is generated, this report contains information about timing, latency, and resource usage. This report is used to optimize the design using the previously mentioned techniques. A comparison report is generated as shown in Tables 2 and 3. Solution 1 represents the system with no optimization while solution 2 and 3 represents the design using Pipeline with different parameters. Since the pipelining approach consumed more resources than the available one, only a small part of the design was pipelined (one iteration of the loop used to compute the Euclidian distance).

Table 2. Implementation reports for solutions 1, 2 and 3 in terms of resource usage

	Used recourses			
	BRAM_18K	DSP48E	FF	LUT
Solution1	6	27	5581	9635
Solution2	6	27	5577	9608
Solution3	2	28	24015	40955

Table 3. Implementation reports for solutions 1, 2 and 3 in terms of time consumption

	Clock(ns)	Latency (clock cycles)		Interval (clock cycles)	
		Min	Max	Min	Max
Solution1	8.62	1334	16958	1335	16959
Solution2	9.53	403	16027	404	16028
Solution3	8.74	176	176	10	10

When the design is finalized, it can be used as an IP in Vivado, the design includes four IPs: the KNN block, the PS block, the AXI Interconnect block to connect the KNN IP block to the PS and finally the System Reset block.

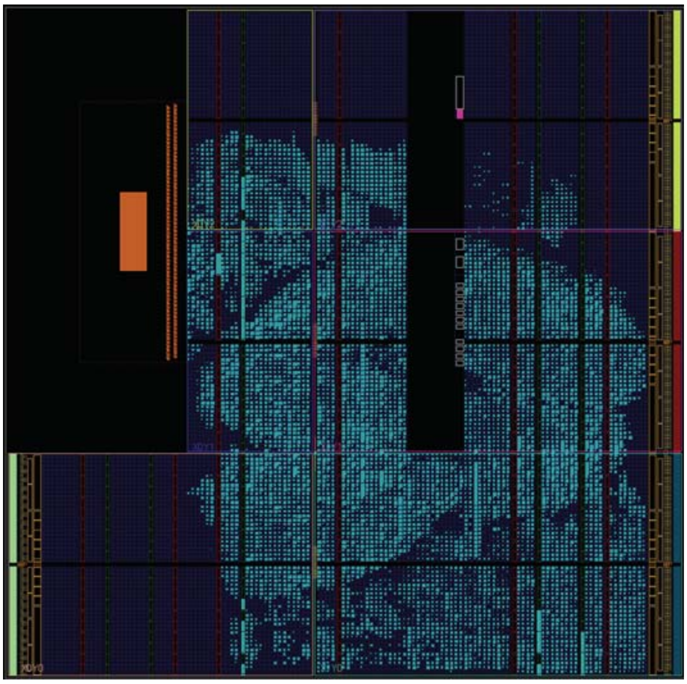

Fig. 14. Chip layout

Figure 14 shows the chip layout for the implemented best solution 3. Vivado does further optimizations when implementing the entire design on Zedboard since it uses less flip flops (FF) and lookup tables (LUTs). Table 4 shows the resource usage for the whole system that include the Processor System Reset, AXI Interconnect and the HLS Predict IP cores when the frequency is set to 667 MHz for the PS and 100 MHz for the PL. The power consumption is shown in Table 5.

The execution time is measured for the KNN algorithm on various platforms including a PC, the PS and the PL of the Zynq. A summary of the execution time can be seen in Table 6. It shows that execution on the programmable logic is the fastest. On the other hand, the execution time on the PS in reasonably similar to the one on PC.

5.3 Alerting System

A user friendly interface has been developed for the alerting system, it can be seen in Fig. 15.

Table 4. Implementation report for solution 3

	BRAM_18K	DSP48E	FF	LUT
HLS Predict	1	28	22423	25030
AXI Interconnect	0	0	475	363
System Reset	0	0	26	15
Total	1	28	22924	25408
Available	140	220	106400	53200

Table 5. Power consumption report for solution 3

Power consumption (W)						
Dynamic 1.663 W (92%)						Static (8%)
PS (82%)	Clocks (4%)	Signals (13%)	Logic (9%)	BRAM (<1%)	DSP (1%)	0.173
1.529	0.091	0.284	0.185	0.003	0.020	

Table 6. Execution time

Implementation platform	Execution time
PC	6 ms
Zynq PS	3 ms
Zynq PL	131 μs

Fig. 15. User interface

6 Conclusion

The work described in this chapter aims at presenting a connected health monitoring and alerting system that meets real time performances. A fall detection system that uses the Shimmer accelerometer data is presented and implemented on the Zynq SoC. A classification accuracy of 90% is reached by using k-Nearest Neighbors with k = 3 while meeting real-time performances using only 48% of LUTs and 22% of FFs available on chip. In addition, an alerting system is implemented, it is capable of sending reports to ambulances and hospital using Wi-Fi. The report includes vital information such as ECG signals which are collected using the same Shimmer sensor and encrypted using the Advanced Encryption Standard for increased privacy.

Acknowledgements. This paper was made possible by National Priorities Research Program (NPRP) grant No. 5-080-2-028 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. Beard, J.R., Officer, A., de Carvalho, I.A., Sadana, R., Pot, A.M., Michel, J.P., Lloyd-Sherlock, P., Epping-Jordan, J.E., Peeters, G.G., Mahanani, W.R., et al.: The world report on ageing and health: a policy framework for healthy ageing. The Lancet **387**(10033), 2145–2154 (2016)
2. Bromiley, P., Courtney, P., Thacker, N.: Design of a visual system for detecting natural events by the use of an independent visual estimate: a human fall detector. Empirical Eval. Methods Comput. Vis. **50** (2002)
3. Burns, A., Greene, B.R., McGrath, M.J., O'Shea, T.J., Kuris, B., Ayer, S.M., Stroiescu, F., Cionca, V.: Shimmer-a wireless sensor platform for noninvasive biomedical research. IEEE Sens. J. **10**(9), 1527–1534 (2010)
4. Counsell, S.R.: 2015 updated ags beers criteria offer guide for safer medication use among older adults. J. Gerontological Nurs. **41**(11), 60 (2015)
5. Crockett, L.H., Elliot, R.A., Enderwitz, M.A., Stewart, R.W.: The Zynq book: embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc. Strathclyde Academic Media (2014)
6. Kepski, M., Kwolek, B.: Fall detection on embedded platform using kinect and wireless accelerometer. In: International Conference on Computers for Handicapped Persons, pp. 407–414. Springer (2012)
7. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: principles and approaches. Neurocomputing **100**, 144–152 (2013)
8. Nait-Charif, H., McKenna, S.J.: Activity summarisation and fall detection in a supportive home environment. In: Proceedings of the 17th International Conference on Pattern Recognition ICPR 2004, vol. 4, pp. 323–326. IEEE (2004)
9. Nguyen, H.T.K., Belleudy, C., Van Tuan, P.: Fall detection application on an arm and fpga heterogeneous computing platform. Int. J. Adv. Res. Electr. Electr. Instrum. Eng. **3**(8), 11349–11357 (2014)
10. Ozcan, K., Velipasalar, S.: Wearable camera-and accelerometer-based fall detection on portable devices. IEEE Embed. Syst. Lett. **8**(1), 6–9 (2016)

11. Özdemir, A.T., Barshan, B.: Detecting falls with wearable sensors using machine learning techniques. Sensors **14**(6), 10691–10708 (2014)
12. Patel, K.V., Brennan, K.L., Davis, M.L., Jupiter, D.C., Brennan, M.L.: High-energy femur fractures increase morbidity but not mortality in elderly patients. Clin. Orthop. Relat. Res. **472**(3), 1030–1035 (2014)
13. Stevens, J.A., Hasbrouck, L., Durant, T.M., Dellinger, A.M., Batabyal, P.K., Crosby, A.E., Valluru, B.R.: Kresnow, M.j., Guerrero, J.L.: Surveillance for injuries and violence among older adults. MMWR CDC Surveill. Summ. **48**(8), 27–50 (1999)
14. Toreyin, B.U., Soyer, E.B., Onaran, I., Cetin, A.E.: Falling person detection using multisensor signal processing. EURASIP J. Adv. Signal Process. **2008**, 29 (2008)
15. Wu, F., Zhao, H., Zhao, Y., Zhong, H.: Development of a wearable-sensor-based fall detection system. Int. J. Telemedicine Appl. **2015**, 2 (2015)
16. Xilix, I.: Zynq-7000 all programmable soc: technical reference manual. ug585, vo1. 8.1. http://www.xilinx.com/support/documentation/userguides/ug585-Zynq-7000-TRM.pdf (2014). Accessed 30 Sep 2016