Journal of Sensor Technology, **2013, 3, 47-56**
http://dx.doi.org/10.4236/jst.2013.33009 Published Online September 2013 (http://www.scirp.org/journal/jst)

System Architecture and Design Flow of Smart Mobile Sensing Systems

Won-Jae Yi, Jafar Saniie
Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA
Email: wyi3@iit.edu, sansonic@ece.iit.edu

Received June 6, 2013; revised July 6, 2013; accepted July 14, 2013

ABSTRACT

This study presents the architecture and design flow of smart mobile sensing systems that performs wireless sensor data transmission, data analysis and display in real-time. Multiple wireless protocols are used for sensor data transmission including the Bluetooth, cellular data network and Wi-Fi for Internet access, and Near Field Communication (NFC). An Android smartphone is utilized to demonstrate the design concept of an Intelligent Personal Communication Node (iPCN) and to perform real-time sensor data acquisition, processing, analysis, display and transmission. Tested sensors include acceleration, temperature, electrocardiography (ECG) and phonocardiography (PCG). For computational capability tests, we have observed the signal processing performance of the smartphone by implementing fast Fourier transform (FFT) of the received ECG signal, and QRS detection algorithm for spontaneous heart beat rate (HBR) estimation. This system has also been tested for multiple sensor node communication and on-demand sensor data acquisition. The smart mobile sensing system can also be applied to any environment that requires real-time sensing and wireless remote monitoring.

Keywords: Android Smartphone; Signal Processing; Wireless Sensor Networks

1. Introduction

Smartphones are the fastest growing component in the IT market and are the biggest phenomenon of our daily life. There are more than 100 million smartphone users in the United States [1] and over 1 billion users throughout the world [2]. A smartphone has extensive capabilities as a valuable data communicator and analyzer. A typical Android smartphone embeds an ARM-based processor that offers sufficient processing power for the operating system and a wide variety of applications. Also, a typical Android device is equipped with multiple wireless adapters including the Internet accessible cellular data network and Wi-Fi adapters, a Bluetooth transceiver, and a Near Field Communication (NFC) reader. These multiple wireless technologies are able to manage real-time data communications on-demand and can be accessed concurrently without interrupting other protocols.

The smartphone can be applied for delivering a more efficient and convenient real-time patient health monitoring because of its extensive communication and computation capabilities. Examples of patient health monitoring attributes include blood pressure, heart beat rate (HBR), body temperature changes, breathing patterns, patient movement detections, hemoglobin saturations and electrocardiography (ECG) [3-6].

The treatment experience for patients and the work efficiency for medical professionals can be enhanced by using wireless body sensor networks (BSN). As shown in **Figure 1**, a wireless BSN consists of body sensors, a Personal Communication Node (PCN) and a central computing server for the medical professionals. Wearable body sensors are desired for mobile system integration [7,8] and must be designed to consume low power in order to provide long battery life [9,10] for extended and continuous patient monitoring. A PCN in the wireless BSN serves as a real-time data communicator with multiple wireless protocols [3,9,11-13] to retrieve sensor data and simultaneous data streaming to a designated central server. Data received on the central server is stored to allow medical professionals to remotely access a patient's test data, and then using data analysis for patient diagnosis. For example, an inquiry application can be implemented on the central server to observe a patient's status remotely [14]. The system can be further extended by integrating GPS (Global Positioning System) data to track a patient's position in real-time [15]. In practice, one central server is sufficient to provide patients' history

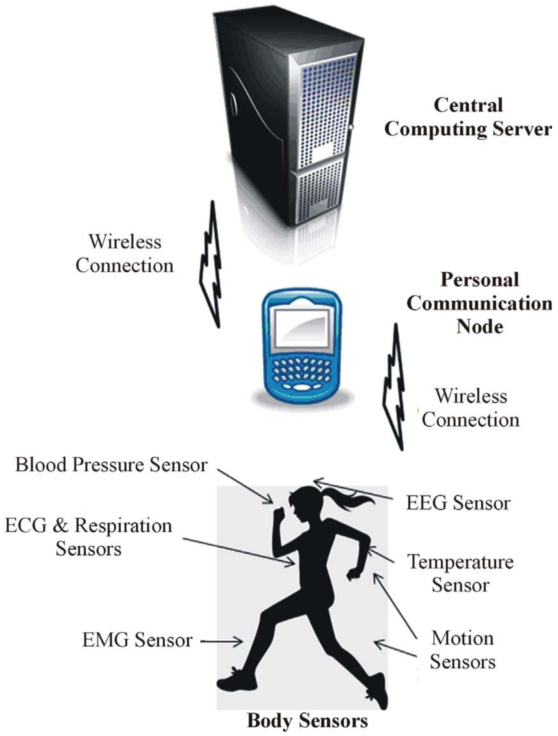


Figure 1. Wireless body sensor network (BSN) overview.

and diagnosis [16], but it is also possible to integrate multiple servers to establish a global and comprehensive infrastructure for a substantial diagnostic database [17].

In this paper, we introduce the system architecture and design flow of smart mobile sensing systems using Android smartphones. This system is capable of data communications with multiple sensors via the Bluetooth connection, data processing and analysis using the on-board ARM-based microprocessor, user-friendly data display, and data streaming to a central server via the Internet connection. The proposed smart mobile system establishes all data communications in real-time and can be easily integrated to any remote monitoring system [18].

This paper is organized as follows: Section 2 illustrates the overall design flow and the architecture of the smart mobile system and emphasizes sensor data interpretation algorithms; Section 3 describes results of various sensor data acquisition components; Section 4 presents the computation and communication capabilities of the smartphone by observing ECG signal processing results and experimental challenges using multiple sensor connections; and finally, Section 5 summarizes the findings and comments on the future trends of a smart mobile sensing system.

2. System Design

2.1. Sensors and Communication Link

We have used an eZ430-RF2560 module [19] to demonstrate the design flow and architecture where a sensor module requires a processing unit, wireless communications and low-power consumptions. This particular system consists of a MSP430BT5190 microcontroller [20] and a CC2560 Bluetooth transceiver [21]. Also, it is equipped with on-board sensors such as a CMA3000-D01 accelerometer [22] and a TMP106YZC temperature sensor [23]. The sensor data can be transferred to the smartphone using the Bluetooth Serial Port Profile (SPP) protocol. In addition, external sensor data such as ECG and phonocardiography (PCG) signals can be interfaced. These peripheral components allow the sensor data to be mobilized, and the device to operate in low power. **Table 1** presents the estimated power consumption of the sensor module where estimation is calculated with common 2 AAA alkaline LR03 batteries for the power supply. The result shows that it meets the requirement for a portable sensor module.

An Android smartphone in our system communicates with the sensor node through Bluetooth SPP protocol to receive the sensor data in real-time. On the sensor module, a UART connection between the sensor and microcontroller is set at the baud rate of 115, 200 bps, achieving approximately 15 KB/s of data transfer rate. This is sufficient enough to provide real-time sensor data acquisition and can be fully maintained under the Bluetooth SPP protocol. A typical Android smartphone equipped with two possible options to connect to the Internet: cellular data networks and a built-in Wi-Fi adapter. Both data paths are suitable to provide the necessary bandwidth to achieve real-time sensor data communication between the smartphone and the central server.

In addition, a typical Bluetooth connection requires a pairing process prior to establishing data connections by manually selecting the target Bluetooth device. This manual procedure can be eliminated by using an NFC tag

Table 1. Estimated sensor system power consumption.

	Power Consumption	
	Maximum (Active)	Minimum (Idle)
MSP430BT5190 Microcontroller	0.88 mA - 1.84 mA	1.2 µA - 2.1 µA
CC2560 Bluetooth Transceiver	39.2 mA	40 µA
CMA3000-D01 Accelerometer	70 µA - 90 µA	unknown
TMP106YZC Temperature Sensor	100 µA	50 µ - 85 µA
Typical ECG Module	180 µA	unknown
Total	40.43 mA - 41.41 mA	91.2 µA - 127.1 µA
Operational Time using 2 AAA Batteries	1 day	327 - 456 days

which provides the MAC address of the targeted Bluetooth device for automatic and instantaneous connection with an application launch. In our design, the NFC tag is programmed to hold the Bluetooth MAC address of the sensor module. NFC tag information can be read using an embedded NFC reader on the Android smartphone by bringing the smartphone into close proximity of the tag as shown in **Figure 2**. Reading different NFC tags using the smartphone makes it possible to store the target information of multiple devices on the Android application. This capability allows multiple Bluetooth connections for collecting multiple sensor data. The design procedure of the multiple Bluetooth connections in our system is introduced in Section 4.

Both automatic and manual Bluetooth connections must be available on the Android smartphone in case there is no NFC reader. A comparison of a Bluetooth connection using the NFC and manual Bluetooth connection is described in **Figure 3** where the manual Bluetooth connection needs tedious steps to initiate the connection, bringing more and unnecessary complexity to users. Our developed system is capable of using both connection methods for any Android device including non-NFC equipped smartphones. After establishing the Bluetooth SPP connection, the Android application flow entails data acquisitions, interpretations, displays and data transmission to central server. The application software receives and interprets sensor data to display them on the smartphone. Then, these data are transferred to the central server via the Internet connection.

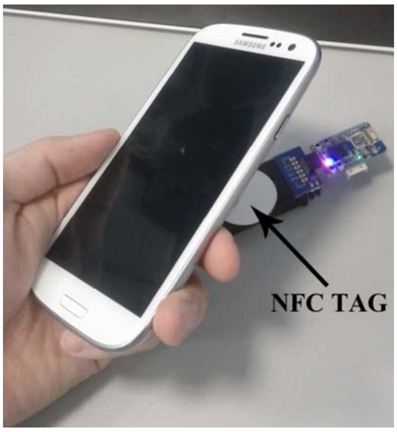


Figure 2. NFC tagging to initiate the Android application.

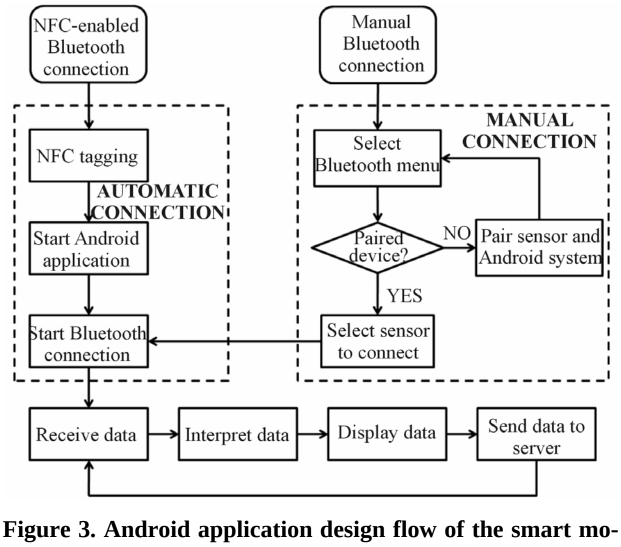


Figure 3. Android application design flow of the smart mobile sensing system.

2.2. Intelligent Personal Communication Node (iPCN)

The Intelligent Personal Communication Node (iPCN) is a system based on an Android smartphone that governs sensor data including signal processing and data transmission as shown in **Figure 4**. Also, a limited data processing at the sensor node level can be achieved using embedded processor (*i.e.*, smart sensor) which helps to reduce the computational load on iPCN.

In practice, signal processing can be achieved on the smartphone without significant hampering of the operating system and utilization of computational resources. For example, the QRS detection [24] of the ECG signal can be achieved in real-time under the Android operating system environment. The QRS complex, which consists of Q, R and S waves of the ECG signal, represents right and left ventricular depolarization of the heart and is useful to determine heart abnormalities [25]. Also, this processing is essential to determine the cardiac cycle by measuring RR intervals (intervals between two R peaks of the ECG signals) within the QRS complex [25]. The QRS detection requires band-pass filtering technique along with adaptive threshold adjustments as shown in **Figure 5**. A real-time QRS detection algorithm using a

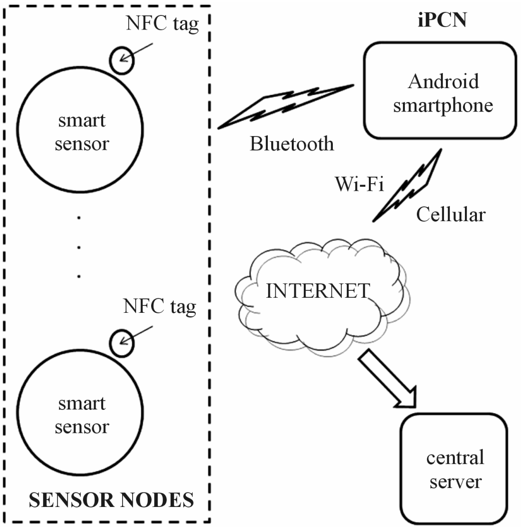


Figure 4. Smart mobile sensing system architecture.

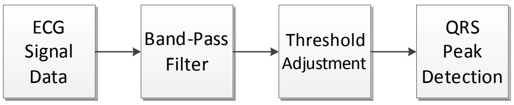


Figure 5. QRS detection algorithm flow.

moving average filter [24] under the Android smartphone is tested in this study and it will be discussed in Section 4.

One of the main goals for the iPCN application is to relay the received sensor data to the server and display them in readable formats. In order to collect and stream various sensor data correctly, individual data interpretation methods must be determined. In this section, three different types of sensor data interpretation are introduced.

The on-board accelerometer on the eZ430-RF2560 produces 3 axes values, having an 8-bit data for each axis. Combined x, y and z-axis data are considered as one data set. The Android smartphone application processes retrieved data sets to display and send to the central server as described in Algorithm 1. Data is either drawn in graph form or printed in numerical values for users to observe their current sensor data stream.

Algorithm 1: Accelerometer Data Application

```
buffer = received accelerometer data
While acknowledges all data sets do
    x-axis = buffer [1st byte]
    y-axis = buffer [2nd byte]
    z-axis = buffer [3rd byte]
    Draw graph (x, y, z) or print text (x, y, z)
    Send data to server (x, y, z)
End While
```

Similar to the above method, retrieved temperature sensor data from the eZ430-RF2560 is also processed on the iPCN where each temperature sensor data is represented in 8-bits. Similar to the accelerometer data communication, the temperature sensor always sends current temperature data including consecutive redundant temperature data. Data redundancy must be taken care of for this particular sensor type on the iPCN application. The application compares incoming temperature sensor data with the previous temperature sensor data to determine consecutive data redundancy as shown in Algorithm 2. Only non-redundant data are used to display current temperature data in text on the iPCN and to be sent to the central server. However, all received data are used for drawing graph on the smartphone. In this particular example, a warning message is generated for user's convenience when the received temperature sensor data reaches a user-defined value. This concept entails the ability of creating a smart warning system and can be of great importance in certain applications such as medical healthcare monitoring.

Algorithm 2: Temperature Sensor Data Application

```
buffer = received temperature data
While acknowledges all data sets do
    temperature = buffer [byte]
    If (no temperature changes) then
        Draw graph (temperature)
    Else If (temperature changed) then
        temperature = new temperature
        Draw graph (temperature) or print text (temperature)
        If (temperature > = 35) // user-defined range
            Generate "Warning" Popup // alert user
        End If
        Send data to server (temperature)
    End if
End While
```

For further system evaluation and to demonstrate data transmission feasibility, ECG and PCG signals at a sampling rate of 250 Hz [26] are used. For this type of sensor data, all data are scaled from floating point format to 32-bit integers to make it suitable for byte-sized Bluetooth SPP data transmissions. Thus, the application needs to convert and scale back the received data to their original format as shown in Algorithm 3. Then, similar to the Algorithm 1 and Algorithm 2, the smartphone sends the data to the graph application and to the server through the Internet.

Algorithm 3: ECG/PCG Sensor Data Application

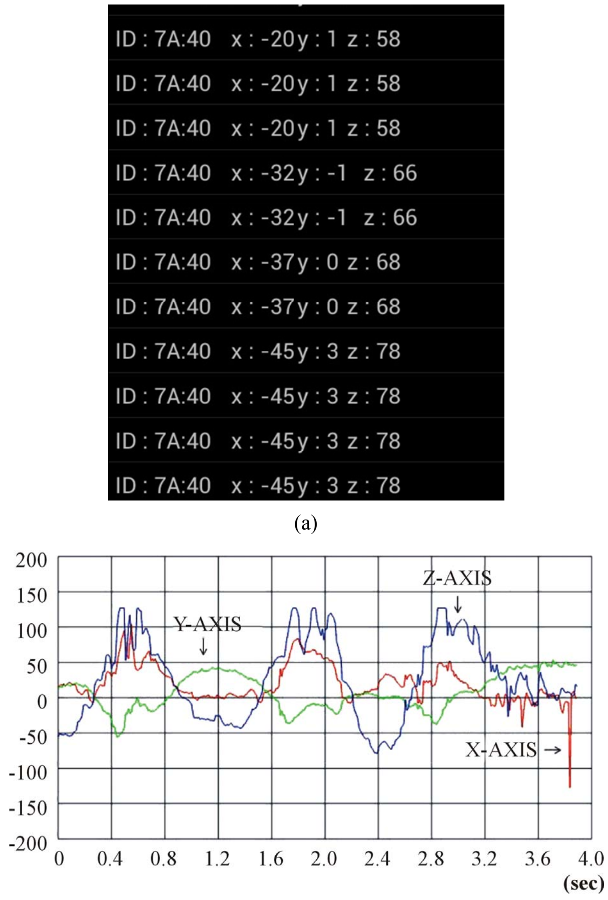
```
buffer = received ECG/PCG data
While acknowledges 4 bytes of data do
    data = (buffer [1st byte] & 0 xff ) << 24
    data + = (buffer [2nd byte] & 0 xff) << 16
    data + = (buffer [3rd byte] & 0 xff) << 8
    data + = (buffer [4th byte] & 0 xff)
    Convert (data to original format)
    Draw graph (data)
    Send to server (data)
End While
```

3. Sensor Data Acquisition

Our customized Android software applications are verified under various Android operating systems from versions 2.3 to 4.2 for data acquisitions and transmissions in real-time using accelerometer data, temperature data, and ECG and PCG signals. An Android smartphone with an embedded NFC reader is required to execute automatic application launch and the automatic Bluetooth connection. For non-NFC equipped Android devices, the user needs to launch the application and select the target sensor node manually to start data acquisitions, processing, display and transmissions.

3.1. Accelerometer Data

Accelerometer data acquisition is achieved by the on-board accelerometer on the eZ430-RF2560, which is connected via the UART serial connection to the MSP-430BT5190 microcontroller. Then, the microcontroller transmits the processed data in real-time to the smartphone through the CC2560 Bluetooth transceiver [21]. **Figure 6(a)** is the result of acquired accelerometer data in text along with the four last digit of the Bluetooth MAC address on the sensor node. **Figure 6(b)** is the result of acquired accelerometer data in graph from a gently swiveled accelerometer. **Figure 7** is the acquired accelerometer data on the server simultaneously from the smartphone via the Internet connection.



```
ID : 7A:40   x : -20y : 1 z : 58
ID : 7A:40   x : -20y : 1 z : 58
ID : 7A:40   x : -20y : 1 z : 58
ID : 7A:40   x : -32y : -1  z : 66
ID : 7A:40   x : -32y : -1  z : 66
ID : 7A:40   x : -37y : 0 z : 68
ID : 7A:40   x : -37y : 0 z : 68
ID : 7A:40   x : -45y : 3 z : 78
ID : 7A:40   x : -45y : 3 z : 78
ID : 7A:40   x : -45y : 3 z : 78
ID : 7A:40   x : -45y : 3 z : 78
```

(a)

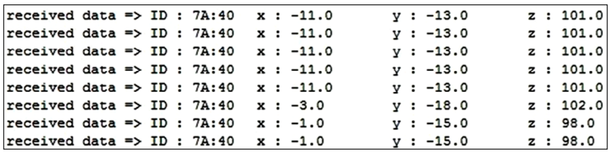
(b)

Figure 6. Received real-time accelerometer data display on the smartphone in (a) text; (b) graph.

Many studies are explored for movement detections using the accelerometer data analysis. However, they all share the same idea for the analysis; their systems compare the incoming data with pre-recorded templates in real-time [4,5,12,27]. To improve accuracy, numerous attempts must be achieved to set a template for a desired detection movement. Multiple accelerometer data are preferred for sophisticated movement detections [28], including posture and movement transition detections.

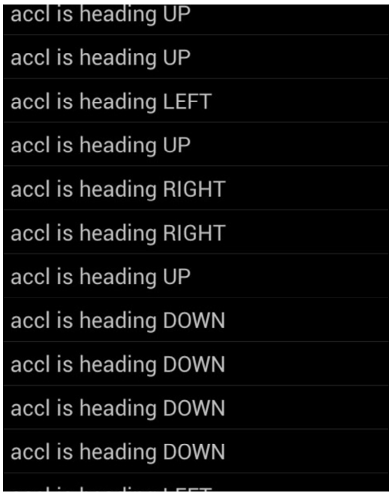
In this study, we have explored a simple accelerometer data analysis using the data retrieved on the smartphone. The x-axis interpretation can be accomplished when the absolute value of x-axis data is greater than the absolute value of y-axis. Depending on the orientation of the device, x-axis data between −60 and −50 represents a steady LEFT, whereas between +50 and +60 represents a steady RIGHT. For the other two directions, similar rules apply to detect the direction observing y-axis data. **Figure 8** shows a simple accelerometer analysis of the raw data interpretation, detecting four two-dimensional directions: UP, DOWN, LEFT, RIGHT.

We have extended this accelerometer data interpretation



```
received data => ID : 7A:40   x : -11.0        y : -13.0        z : 101.0
received data => ID : 7A:40   x : -11.0        y : -13.0        z : 101.0
received data => ID : 7A:40   x : -11.0        y : -13.0        z : 101.0
received data => ID : 7A:40   x : -11.0        y : -13.0        z : 101.0
received data => ID : 7A:40   x : -11.0        y : -13.0        z : 101.0
received data => ID : 7A:40   x : -3.0         y : -18.0        z : 102.0
received data => ID : 7A:40   x : -1.0         y : -15.0        z : 98.0
received data => ID : 7A:40   x : -1.0         y : -15.0        z : 98.0
```

Figure 7. Received real-time accelerometer data on the server.



```
accl is heading UP
accl is heading UP
accl is heading LEFT
accl is heading UP
accl is heading RIGHT
accl is heading RIGHT
accl is heading UP
accl is heading DOWN
accl is heading DOWN
accl is heading DOWN
accl is heading DOWN
```

Figure 8. Two-dimensional accelerometer movement detection.

to the Snake Game sample from Android Developers [29]. The original game requires touch gestures on the screen to play the game. The touch screen inputs are replaced by the accelerometer using the above method to interpret the raw data. **Figure 9** shows the snake's movement by using the accelerometer on the microcontroller. This particular experiment emphasizes that the accelerometer could be used for subject movement detections using 3 axes data. Also, multiple accelerometer data integration for sophisticated movement detections will increase the movement estimation accuracy.

3.2. Temperature Sensor Data

The temperature sensor on the eZ430-RF2560, detecting the room temperature in real-time is used for the data acquisition. Similar to the previous method, the acquired data can be displayed in text or graph on the smartphone screen. A heat gun is used to demonstrate room temperature changes on the embedded temperature sensor as shown in **Figure 10**.

Figure 11 is the result of the temperature sensor data acquisition on the smartphone in text and graph. In particular, as shown in **Figure 11(b)**, a system warning popup message is implemented when the temperature goes beyond 35 degrees Celsius to alert the user on the graph application. This application also sends data to the central server simultaneously for data storage and analysis, as shown in **Figure 12**. As described in the previous section, consecutive redundant temperature data are only considered for the graph application, but discarded when the data is represented in text.

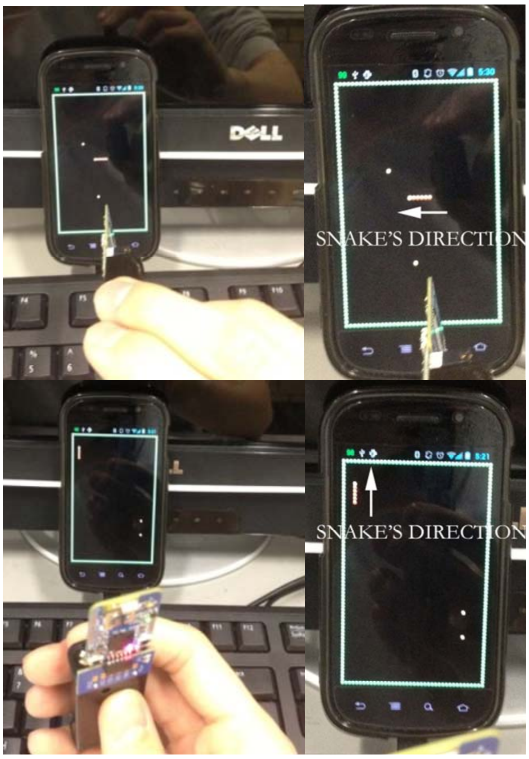


Figure 9. Remote controlling Snake Game.

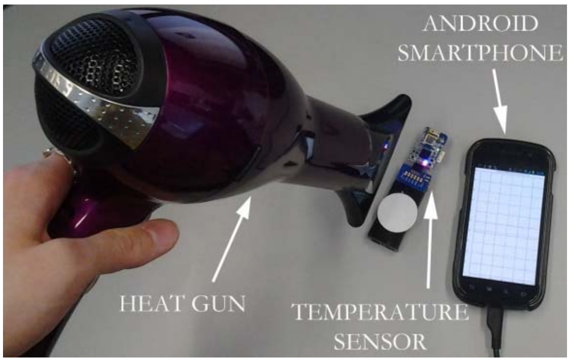
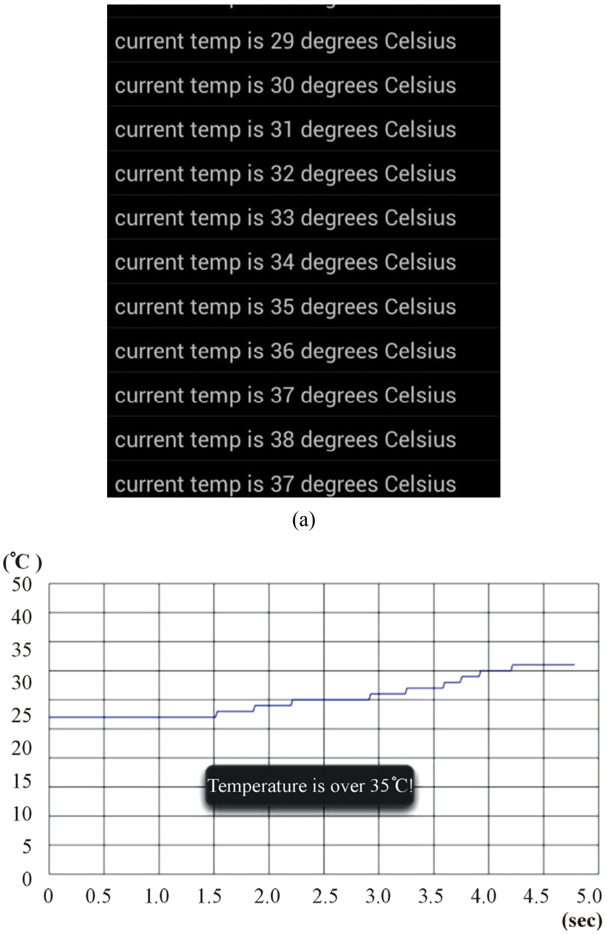


Figure 10. Testing temperature sensor data transmission.



current temp is 29 degrees Celsius
current temp is 30 degrees Celsius
current temp is 31 degrees Celsius
current temp is 32 degrees Celsius
current temp is 33 degrees Celsius
current temp is 34 degrees Celsius
current temp is 35 degrees Celsius
current temp is 36 degrees Celsius
current temp is 37 degrees Celsius
current temp is 38 degrees Celsius
current temp is 37 degrees Celsius

(a)

(b)

Figure 11. Received real-time temperature data display on the smartphone in (a) text; (b) grap.

3.3. ECG and PCG Data

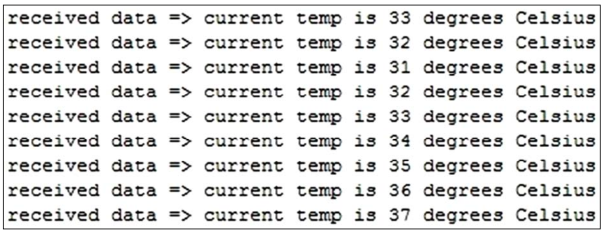
Pre-recorded ECG and PCG signal data from an online database [26] are used to demonstrate the feasibility of simultaneous sensor data acquisition and display on the smartphone, and stream to a central server. For example, the ECG and PCG data are sent to the smartphone using Bluetooth SPP protocol, and plotted for display (see **Figure 13**) and simultaneously transmitted to a central server through the Internet connection.

4. Smartphone Computation and Communication

4.1. Computation Evaluation

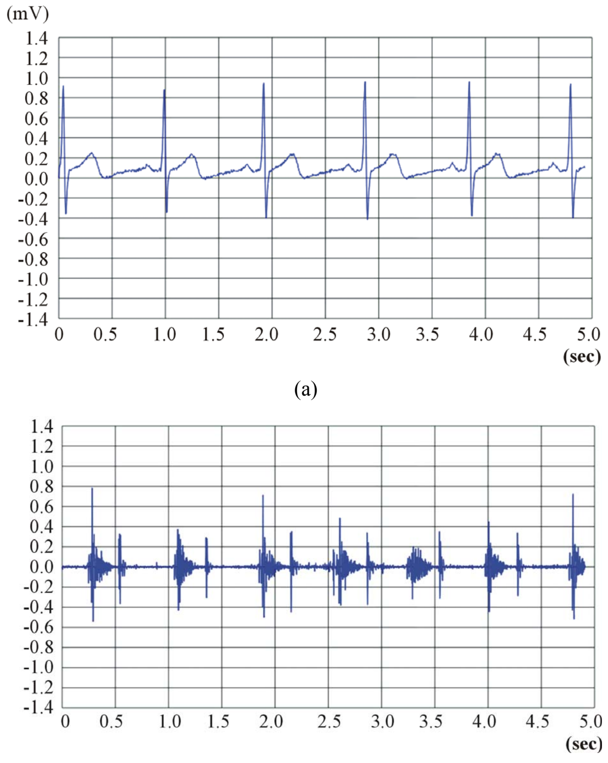
To evaluate the computational capability of the smartphone for signal processing, we have implemented fast Fourier transform (FFT) of the received ECG data on the Android smartphone as shown in **Figure 14**.

We have tested two different types of Android smartphones, Nexus S and Galaxy S III, where the Nexus S represents the low-end smartphone and the Galaxy S III represents relatively most up-to-date high-end smartphone, in terms of CPU power. As shown in **Table 2**, both Android smartphones' performance is satisfactory and practically can handle FFT computations in the order of tens of milliseconds when the data size is a few thousand samples. This result proves that most of the common Android smartphones available in the market have sufficient computing power to accomplish signal processing.



```
received data => current temp is 33 degrees Celsius
received data => current temp is 32 degrees Celsius
received data => current temp is 31 degrees Celsius
received data => current temp is 32 degrees Celsius
received data => current temp is 33 degrees Celsius
received data => current temp is 34 degrees Celsius
received data => current temp is 35 degrees Celsius
received data => current temp is 36 degrees Celsius
received data => current temp is 37 degrees Celsius
```

Figure 12. Received real-time temperature data on the server.



(a)

(b)

Figure 13. Received real-time sensor data on the smartphone of (a) ECG signal data; (b) PCG signal data.

4.2. QRS Detection and Heart Beat Rate (HBR) Estimation

ECG signal analysis is useful to determine the heart activity which can be analyzed by observing the ECG signal characteristics in real-time. In our study, the HBR is calculated by identifying RR intervals based on the result obtained from QRS peak detections using a moving average filter [24] on the Android smartphone. This method can determine the HBR in real-time and does not need to wait for 60 seconds to calculate the heart beats per minute (BPM) since every measurement of RR intervals can be converted into the HBR. **Figure 15** shows the HBR in BPM alongside with the display of ECG signal in real-

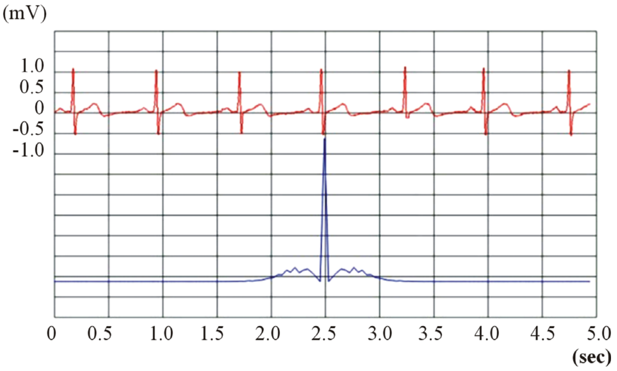


Figure 14. ECG signal and FFT result displayed on the smartphone.

Table 2. Smartphone and FFT computation performance comparison.

N Samples for FFT	Computation Time	
	Nexus S Android Smartphone	Galaxy S III Android Smartphone
128	10.27 ms	0.26 ms
256	17.40 ms	1.89 ms
512	34.10 ms	6.64 ms
1024	44.10 ms	12.59 ms
2048	65.86 ms	21.79 ms

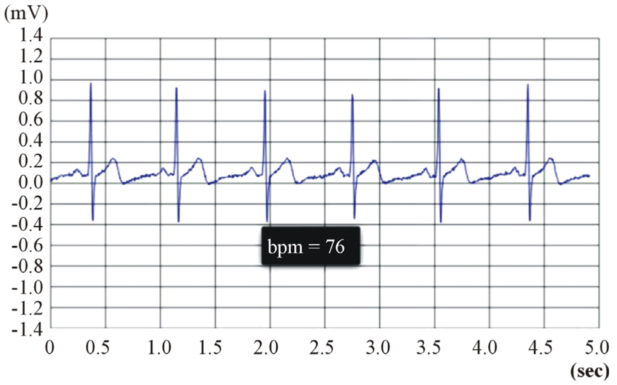


Figure 15. Received real-time ECG signal data and HBR in BPM.

time.

4.3. Multi-Sensor Connections

According to the Bluetooth protocol specification, multiple concurrent connections up to 7 slaves to 1 master node are possible [30]. This can be applied to multiple sensor data transfers without interrupting other sensor data acquisition processes. However, this configuration produces unstable smartphone system operations. To overcome this issue, our design avoids concurrent data transmissions between the smartphone and the sensor nodes, but connects to different sensors and commits sensor data transfer instantly when an interrupt is triggered on that particular sensor.

In order to connect to different sensor nodes, the information of the sensor nodes must be registered to the Android application. This can be achieved by reading multiple Bluetooth MAC addresses using NFC tags. As the NFC and the Bluetooth can be used concurrently on the Android system, the new sensor node information can be obtained at any time even during Bluetooth data transmissions. When registering new sensor node information to the Android application, it only connects to the sensor node and awaits for the data transfer requests. By this arrangement, less system resources of the Android operating system is required for the operation. **Figure 16** is the design flow of multiple sensor node connection.

The application is designed to remove NFC tag information if a repeat tag action by the same sensor node is recognized. This action will help to avoid repeatedly registering the same sensor node. However, this does not apply when there is only one MAC address registered on the queue. Furthermore, the removed MAC address information can be restored by reading the NFC tag. The application provides the information to the user about the connected devices in real-time. As shown in **Figure 17**, two different accelerometers are used to demonstrate connection transitions from one node to the other. Differentiating the types of the sensor data can be accomplished by identifying the Bluetooth MAC address information on the sensor or by programming the information on to the NFC tag. **Figure 18** is the result displayed on the server where the sensor data type can be differentiated by determining the ID field which is the last four digits of the Bluetooth MAC address.

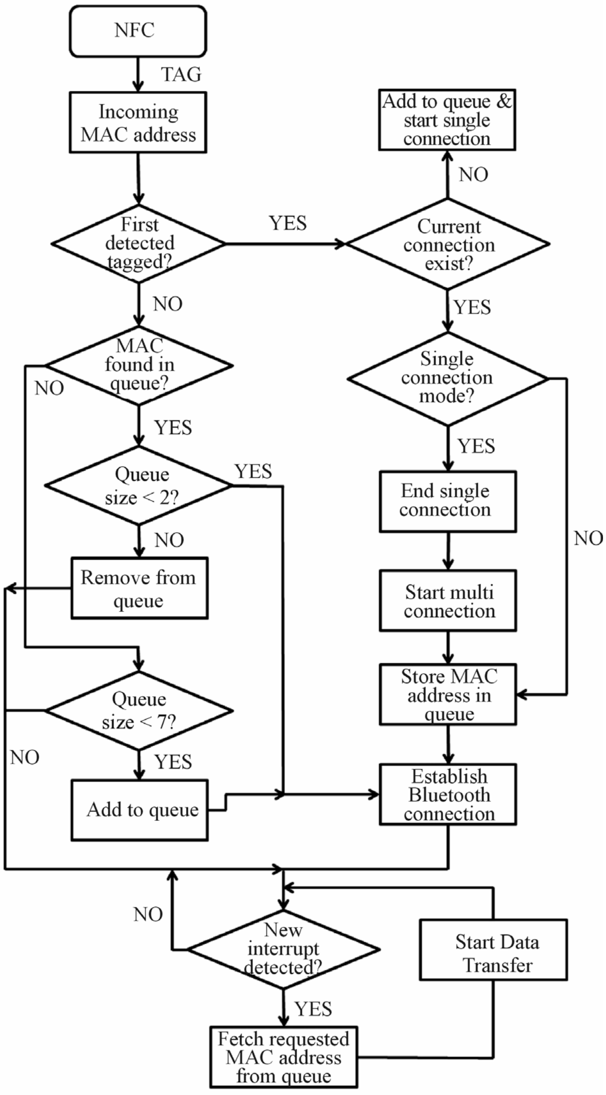
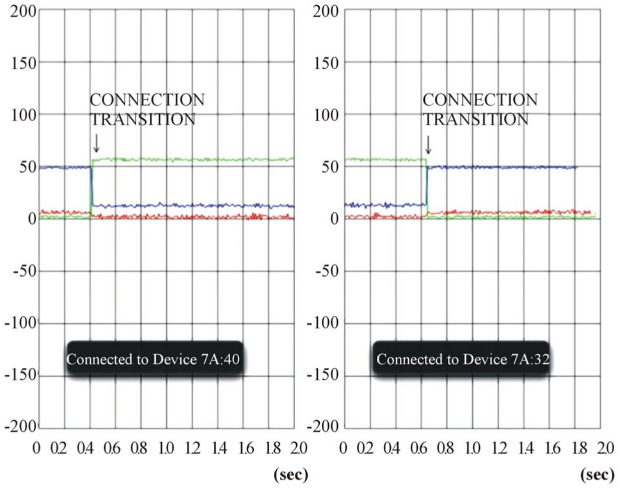


Figure 16. Android application flow of multiple sensor connection.

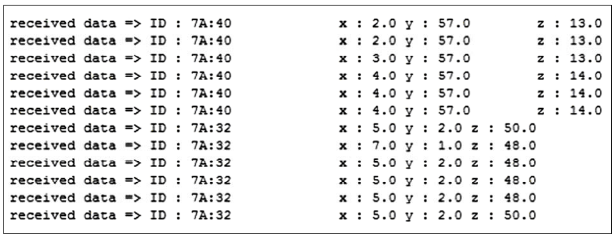
5. Conclusion

In this paper, the system architecture and design flow of a smart mobile sensing system using an Android smartphone are introduced. The smartphone in our design is able to achieve sensor data acquisition and transmission using multiple wireless protocols in real-time as well as data display. The sensor node is desired to be a low-power and efficient system to improve user-experience,



(a) (b)

Figure 17. Received real-time accelerometer data on the smartphone during transitions (a) from Device 7A:32 to 7A:40; (b) from Device 7A:40 to 7A:32.



```
received data => ID : 7A:40            x : 2.0 y : 57.0        z : 13.0
received data => ID : 7A:40            x : 2.0 y : 57.0        z : 13.0
received data => ID : 7A:40            x : 3.0 y : 57.0        z : 13.0
received data => ID : 7A:40            x : 4.0 y : 57.0        z : 14.0
received data => ID : 7A:40            x : 4.0 y : 57.0        z : 14.0
received data => ID : 7A:40            x : 4.0 y : 57.0        z : 14.0
received data => ID : 7A:32            x : 5.0 y : 2.0 z : 50.0
received data => ID : 7A:32            x : 7.0 y : 1.0 z : 48.0
received data => ID : 7A:32            x : 5.0 y : 2.0 z : 48.0
received data => ID : 7A:32            x : 5.0 y : 2.0 z : 48.0
received data => ID : 7A:32            x : 5.0 y : 2.0 z : 48.0
received data => ID : 7A:32            x : 5.0 y : 2.0 z : 50.0
```

Figure 18. Received real-time accelerometer data on the server during transitions from Device 7A:40 to 7A:32.

and to deliver critical information on-demand. Various sensor data including accelerometer, temperature sensor, ECG and PCG data are used to demonstrate the feasibility of the system for real-time sensor data transmission and processing. FFT computation and QRS detection are tested on the Android smartphone for real-time signal processing. This system design has improved the efficiency for user interactions by combining NFC technology with the Bluetooth protocol. This arrangement eliminates monotonous Bluetooth pairing and connection procedures, and enables automatic executions of the Android application. In addition, multiple sensor node connections via the Bluetooth are possible using the NFC by registering multiple Bluetooth device information on the Android application. The developed system has the capability to be further extended not only to day-to-day patient health status monitoring, but also to subjects working in hazardous environment that require continuous monitoring. Any complex systems demanding multiple sensors for remote monitoring safe operations and/or integrity of multipart structures used in transportation systems, civil structures or equipment are also applicable for this design.

REFERENCES

[1] A. Alexander, "Smartphone Usage Statistics 2012," 2012. http://ansonalex.com/infographics/smartphone-usage-statistics-2012-infographic

[2] T. Mogg, "US Smartphone Users Now over 100 Million, Android Increases Market Share," 2012. http://news.yahoo.com/us-smartphone-users/now-over-100-million-android-041611789.html

[3] I. Korhonon, J. Parkka and M. Van Gils, "Health Monitoring in the Home of the Future," *IEEE Engineering in Medicine and Biology Magazine*, Vol. 22, No. 3, 2003, pp. 66-73. doi:10.1109/MEMB.2003.1213628

[4] H. Tanaka, R. Kimura and S. Ioroi, "Equipment Operation by Motion Recognition with Wearable Wireless Acceleration Sensor," *Third International Conference on Next Generation Mobile Applications, Services and Technologies* (*NGMAST* '09), Cardiff, 15-19 September 2009, pp. 114-118. doi:10.1109/NGMAST.2009.96

[5] C. J. James and S. Kumar, "Detection of Posture and Motion by Accelerometer Sensors," 3*rd International Conference on Electronics Computer Technology* (*ICECT*), Kanyakumari, 8-10 April 2011, pp. 369-371.

[6] P.K. Baheti and H. Garudadri, "An Ultra Low Power Pulse Oximeter Sensor Based on Compressed Sensing," 6*th International Workshop on Wearable and Implantable Body Sensor Networks*, Berkley, 3-5 June 2007, pp. 144-148.

[7] M.-H. Cheng, *et al.*, "A Vital Wearing System with Wireless Capability," 2*nd International Conference on Pervasive Computing Technologies for Healthcare*, Tampere, 30 January-1 February 2008, pp. 268-271.

[8] H. Ghasemzadeh, V. Loseu and R. Jafari, "Structural Action Recognition in Body Sensor Networks: Distributed Classification Based on String Matching," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 14, No. 2, 2010, pp. 425-435. doi:10.1109/TITB.2009.2036722

[9] S.-L. Chen, H.-Y. Lee, C.-A. Chen, H.-Y. Huang and C.-H. Luo, "Wireless Body Sensor Network with Adaptive Low-Power Design for Biometrics and Healthcare Applications," *IEEE Systems Journal*, Vol. 3, No. 4, 2009, pp. 398-409. doi:10.1109/JSYST.2009.2032440

[10] J. Yoo, L. Yan, S. Lee, Y. Kim and H.-J. Yoo, "A 5.2mW Self-Configured Wearable Body Sensor Network Controller and a 12uW Wirelessly Powered Sensor for a Continuous Health Monitoring System," *IEEE Journal of Solid-State Circuits*, Vol. 45, No. 1, 2010, pp. 178-188. doi:10.1109/ISSCC.2009.4977422

[11] H. Li and J. Tan, "An Ultra-Low-Power Medium Access Control Protocol for Body Sensor Network," *IEEE 62nd Vehicular Technology Conference*, Dallas, 25-28 September 2005, pp. 2342-2346.

[12] B. P. L. Lo and G. Yang, "Key Technical Challenges and Current Implementations of Body Sensor Networks," 2*nd International Workshop on Wearable and Implantable Body Sensor Networks*, London, 12-13 April 2005. http://vip.doc.ic.ac.uk/bsn/public/UbiMonPapers/Key_Technical_Challenges_and_Current_Implementations_of_

Body_Sensor_Networks.pdf

[13] S. Hendershot, A. Hilton and M. Oo, “Wireless Wearable Motion Sensor for Use in Medical Care,” *IEEE Published Student Application Papers*, 2009.

[14] N. Zarka, M. F. Hinnawi, A. Dardari and M. A. Tayyan, “Patient Keeper” Medical Application on Mobile Phone,” *Information and Communication Technologies*: *From Theory to Applications*, Damascus, 19-23 April 2004, pp. 37-38.

[15] C. Rotariu, H. Constin, G. Andruseac, R. Ciobotariu and F. Adochiei, “An Integrated System for Wireless Monitoring of Chronic Patients and Elderly People,” 15*th International Conference on System Theory*, *Control and Computing* (*ICSTCC*), Sinaia, 14-16 October 2011, pp. 1-4.

[16] C. A. Otto, E. Jovanov and A. Milenkovic, “A WBAN-Based System for Health Monitoring at Home,” 3*rd IEEE*/*EMBS International Summer School on Medical Devices and Biosensors*, Boston, 4-6 September 2006, pp. 20-23.

[17] N. Bricon-Souf, D. Delerue, H. Bezzazi, D. Donsez and R. J. Beuscart, “A Regional Server for Medical Information,” *Proceedings of the* 19*th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Chicago, 30 October-2 November 1997, pp. 939-940.

[18] W. Yi, W. Jia and J. Saniie, “Mobile Sensor Data Collector using Android Smartphone,” *IEEE* 55*th International Midwest Symposium on Circuits and Systems*, Boise, 5-8 August 2012, pp. 956-959.

[19] Texas Instruments, “Texas Instruments Embedded Processors Wiki for eZ430-RF2560,” 2013. http://processors.wiki.ti.com/index.php/EZ430-RF2560

[20] Texas Instruments, “MSP430BT5190 Mixed Signal Microcontroller,” 2010. http://www.ti.com/lit/ds/symlink/msp430bt5190.pdf

[21] Texas Instruments, “CC2560 Bluetooth Single-Chip Solution,” 2010. http://www.ti.com/lit/ds/symlink/cc2560-pan1325.pdf

[22] Murata Electronics, “CMA3000-D01 3-Axis Ultra Low Power Accelerometer with Digital SPI and I2C Interface,” 2012. http://www.muratamems.fi/sites/default/files/documents/cma3000_d01_datasheet_8277800a.04_0.pdf

[23] Texas Instruments, “Digital Temperature Sensor with Two-Wire Interface,” 2006. http://www.ti.com/lit/ds/symlink/tmp106.pdf

[24] H. C. Chen and S. W. Chen, “A Moving Average based Filtering System with Its Application to Real-Time QRS Detection,” *Computers in Cardiology*, Thessaloniki, 21-24 September 2003, pp. 585-588.

[25] F. G. Yanowitz, “ECG Learning Center—An Introduction to Clinical Electrocardiography,” 2012. http://ecg.utah.edu

[26] A. L. Goldberger, ***et al*.**, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,” *Circulation*, Vol. 101, No. 23, 2000, pp. e215-e220. doi:10.1161/01.CIR.101.23.e215

[27] K. D. Nguyen, I.-M. Chen, Z. Luo, S. H. Yeo and H. B.-L. Duh, “A Wearable Sensing System for Tracking and Monitoring of Functional Arm Movement,” *IEEE*/*ASME Transactions on Mechatronics*, Vol. 16, No. 2, 2011, pp. 213-220. doi:10.1109/TMECH.2009.2039222

[28] N. Crampton, K. Fox, H. Johnston and A. Whitehead, “Dance, Dance Evolution: Accelerometer Sensor Networks as Input to Video Games,” *IEEE International Workshop on Haptic*, *Audio and Visual Environment and Games*, Ottawa, 12-14 October 2007, pp. 107-112. doi:10.1109/HAVE.2007.4371597

[29] Android Developers, “Samples,” 2013. http://developer.android.com/tools/samples/index.html

[30] I. H. Mulyadi, E. Supriyanto, N. M. Safri and M. H. Satria, “Wireless Medical Interface Using ZigBee and Bluetooth Technology,” 3*rd Asia International Conference on Modelling & Simulation*, Bandung, 25-29 May 2009, pp. 276-281.