Mobile Sensor Data Collector using Android Smartphone

Won-Jae Yi, Weidi Jia, and Jafar Saniie

Department of Electrical and Computer Engineering, Illinois Institute of Technology
3301 S. Dearborn St. 103SH, Chicago IL, 60616, U.S.A.

Abstract— **In this paper, we present a system using an Android smartphone that collects, displays sensor data on the screen and streams to the central server simultaneously. Bluetooth and wireless Internet connections are used for data transmissions among the devices. Also, using Near Field Communication (NFC) technology, we have constructed a more efficient and convenient mechanism to achieve an automatic Bluetooth connection and application execution. This system is beneficial on body sensor networks (BSN) developed for medical healthcare applications. For demonstration purposes, an accelerometer, a temperature sensor and electrocardiography (ECG) signal data are used to perform the experiments. Raw sensor data are interpreted to either graphical or text notations to be presented on the smartphone and the central server. Furthermore, a Java-based central server application is used to demonstrate communication with the Android system for data storage and analysis.**

I. Introduction

Mobile communication devices are designed to achieve multiple purposes but mostly are focused on voice and short messaging services. Wireless technology has the benefit of improving data mobility, using different protocols such as Wi-Fi and Bluetooth. In the medical field, many studies introduced body sensor networks (BSN) for healthcare applications. BSN improves the patient's monitoring system with the help of the modern technology. This can be done by various wearable sensors equipped with wireless capabilities [1], [2]. In addition, as seen in various researches [3], [4], it is desirable to develop a low power system. Different types of sensors can be used for monitoring movements [5], temperature changes, heart-beat, blood pressure and more to establish a patient monitoring system [6]. Bluetooth is one of the widely available options for managing wireless networks to simultaneously connect up to 7 ancillary devices [7].

In this paper, we introduce a microcontroller system that communicates via Bluetooth with the smartphone for data collections, and streams data simultaneously to the central server for data storage and analysis via the Internet. This system provides a solution for mobile patients by forming a wireless BSN in Bluetooth and Wi-Fi/cellular Internet connections with a common Android smartphone which can monitor the patient status via wireless data transmission.

II. System Design

Figure 1 represents the Mobile Sensor Data Collector that involves Bluetooth, Near Field Communication (NFC) and wireless Internet connections for collecting, streaming, storing and analyzing sensor data in real-time. Three different sensors transfer sampled data to the MSP430BT5190 [8] which communicate with the CC2560 Bluetooth transmitter via UART and sends data to a smartphone using the Android and Bluetooth system. On the phone, it displays received data on the screen and streams to the server for storage and data analysis. The term "real-time" in this paper is used to express that data transfers are achieved without perceivable delays among the devices. Also, since the Android system is capable of running application software in the background mode, the application used in this paper has the ability to transfer data during a phone call.

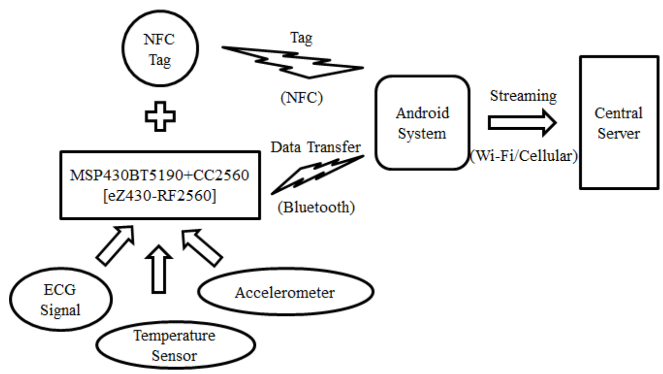


Figure 1. Overall Design of Mobile Sensor Data Collector

A Java-based UDP server application is used to collect data sent from the smartphone via the Internet. When receiving data from the smartphone, the server application displays and saves all received data to a text file for later analysis. For experimental purposes, this server was implemented with an ordinary desktop to demonstrate our fundamental idea. Also, UDP protocol was chosen over TCP because UDP usually achieves faster transmission than the TCP protocol by not waiting for an acknowledgment signal back to the origin.

III. Experiment Results

As shown in Figure 1, all experiments are initiated using an NFC tagging process to start the Android application and initiate the Bluetooth connection automatically. In this particular smartphone, the NFC tag reader is located on the backside. The user needs to tap on the NFC tag as shown in Figure 2 to run the program. The NFC tag containing the

978-1-4673-2527-1/12/$31.00 ©2012 IEEE

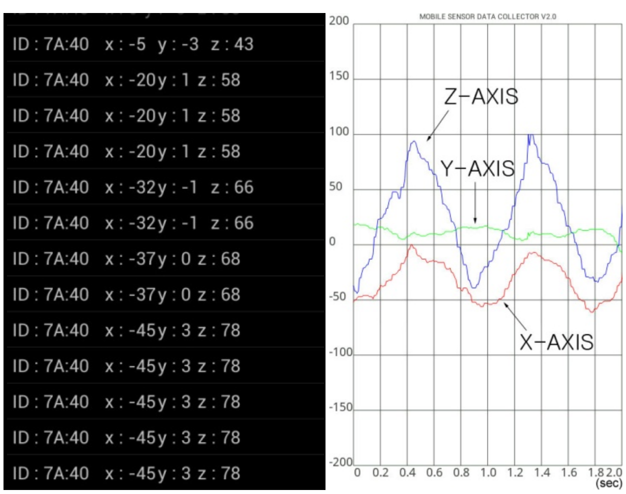
Bluetooth MAC address of the CC2560 Bluetooth device is attached to demonstrate where the tag should be located.

Figure 2. Initiating connection process

Up to 7 ancillary sensor nodes can be simultaneously connected to the Android system. However, a single sensor Bluetooth connection was employed for testing purposes.

A. Accelerometer Data Collection

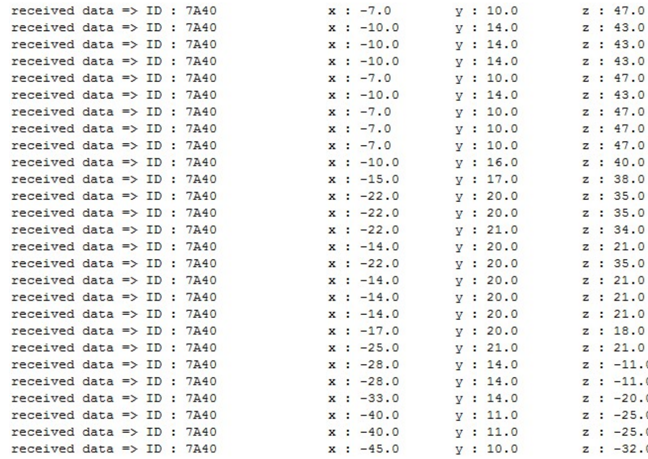
In this paper, the Android 2.3.3 and 4.0.3 operating systems are tested using Google Nexus S to display collected data and stream data to the server. The design of the new system is achieved first by collecting sensor data from the MSP430BT5190, transferred via the CC2560 Bluetooth transmitter. Then, the Bluetooth transmitter sends data to the smartphone, which displays the collected data in real-time. As an example, Figure 3(a) shows the accelerometer data collected and displayed on the smartphone in text and Figure 3(b) shows the data in the graphical notation.



(a) (b)

Figure 3. Received real-time acceleration data display
(a) text notation; (b) graphical notation

These data are being sent to the central server either via Wi-Fi or cellular networks for storage and analysis at the same time. Figure 4 shows the received data from the smartphone displayed on the server. The server also saves data to a text file in the designated directory for data analysis.



```
received data => ID : 7A40    x : -7.0     y : 10.0    z : 47.0
received data => ID : 7A40    x : -10.0    y : 14.0    z : 43.0
received data => ID : 7A40    x : -10.0    y : 14.0    z : 43.0
received data => ID : 7A40    x : -10.0    y : 14.0    z : 43.0
received data => ID : 7A40    x : -7.0     y : 10.0    z : 47.0
received data => ID : 7A40    x : -10.0    y : 14.0    z : 43.0
received data => ID : 7A40    x : -7.0     y : 10.0    z : 47.0
received data => ID : 7A40    x : -7.0     y : 10.0    z : 47.0
received data => ID : 7A40    x : -7.0     y : 10.0    z : 47.0
received data => ID : 7A40    x : -10.0    y : 16.0    z : 40.0
received data => ID : 7A40    x : -15.0    y : 17.0    z : 38.0
received data => ID : 7A40    x : -22.0    y : 20.0    z : 35.0
received data => ID : 7A40    x : -22.0    y : 20.0    z : 35.0
received data => ID : 7A40    x : -22.0    y : 21.0    z : 34.0
received data => ID : 7A40    x : -14.0    y : 20.0    z : 21.0
received data => ID : 7A40    x : -22.0    y : 20.0    z : 35.0
received data => ID : 7A40    x : -14.0    y : 20.0    z : 21.0
received data => ID : 7A40    x : -14.0    y : 20.0    z : 21.0
received data => ID : 7A40    x : -14.0    y : 20.0    z : 21.0
received data => ID : 7A40    x : -17.0    y : 20.0    z : 18.0
received data => ID : 7A40    x : -25.0    y : 21.0    z : 21.0
received data => ID : 7A40    x : -28.0    y : 14.0    z : -11.0
received data => ID : 7A40    x : -28.0    y : 14.0    z : -11.0
received data => ID : 7A40    x : -33.0    y : 14.0    z : -20.0
received data => ID : 7A40    x : -40.0    y : 11.0    z : -25.0
received data => ID : 7A40    x : -40.0    y : 11.0    z : -25.0
received data => ID : 7A40    x : -45.0    y : 10.0    z : -32.0
```

Figure 4. Received real-time acceleration data on server

An axis value representation depends on the raw sensor data and this raw data could differ from the sensors. There are 3 axes provided from the sensor and each set of data needs to be interpreted. For this particular device used in this paper, x-axis data between -60 and -50 represents LEFT, between +50 and +60 represents RIGHT. This rule applies similarly to the other two axes. This differs from other sensors where the data output of acceleration is normally represented in terms of m/s^2. However, a translation algorithm shares the same idea. Figure 5 is the result of translating the accelerometer data based on accelerometer movements.

Figure 5. Accelerometer data translation

This type of the accelerometer translation was extended to the Snake Game sample provided by Android Developers [9]. The original game uses touch screen inputs to control the snake. The touch screen inputs were replaced by accelerometer movements to provide data in LEFT, RIGHT, UP and DOWN. The data analysis was done on the phone itself for test purposes. Figure 6 shows the movement of the snake on the phone that is controlled by accelerometer data from the MSP430 eZ430-RF2560.

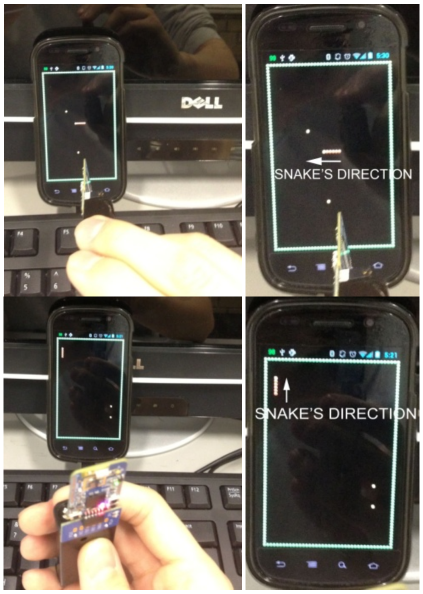


Figure 6. Remote controlling Snake Game

This example emphasizes that accelerometer data can be adapted for the patient movement detection system. Multiple accelerometers could be implemented to produce more advanced movement analysis.

B. *Temperature Sensor Data Collection*

A temperature sensor monitoring the real-time room temperature is used to perform the experiment. The procedure of the experiment resembles the previous section but with the different data interpretation. In this particular experiment, a heat gun was used to heat up or cool down the sensor for testing purposes as shown in Figure 7. Similar to the previous accelerometer application, Figure 8(a) shows the text notation of the received data in real-time and Figure 8(b) shows the graphical notation of the received data in real-time. Particularly in the graphical notation output, we provide a warning message if the temperature exceeds more than 35 degrees Celsius. Also, the graphical notation has a range of between 0 degrees Celsius to 50 degrees Celsius for this demonstration.

Figure 9 shows the server displaying the received data from the smartphone. It delivers similar outputs compared to the accelerometer demonstration and also saves it to a text file.

Figure 7. Testing temperature sensor data transmission

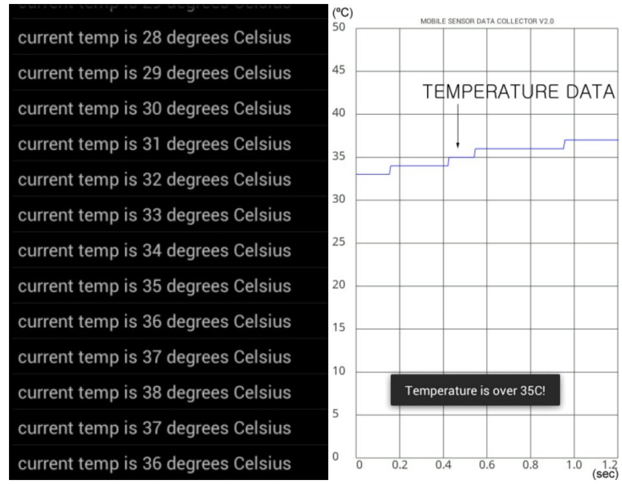


Figure 8. Received real-time temperature data display (a) text notation; (b) graphical notation

```
received data => current temp is 29 degrees Celsius
received data => current temp is 30 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
received data => current temp is 38 degrees Celsius
received data => current temp is 37 degrees Celsius
received data => current temp is 36 degrees Celsius
received data => current temp is 35 degrees Celsius
```

Figure 9. Received real-time temperature data on server

C. *Electrocardiography (ECG) Data Collection*

The ECG signal is an important part of a patient monitoring system. Currently, ECG machines are dependent on wired connections which limit their data mobility. Our system using the Bluetooth protocol for ECG signal collections greatly enhances the mobility. This ECG signal is also sent simultaneously to the server via a wireless Internet connection through the smartphone in real-time. Figure 10 shows the display of received ECG signal on the smartphone and Figure 11 shows the same result transmitted to the server in the text format.

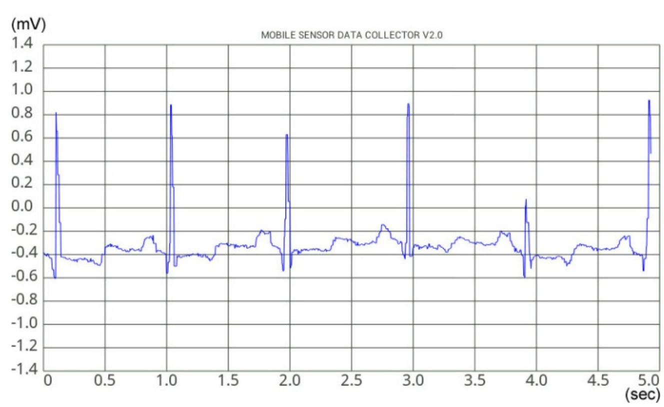


Figure 10. Received real-time ECG data in graphical notation

```
received data => -0.305
received data => -0.305
received data => -0.365
received data => -0.365
received data => -0.365
received data => -0.365
received data => -0.365
received data => -0.410
received data => -0.459
received data => -0.505
received data => -0.459
received data => -0.045
received data => 0.49
received data => 0.88
received data => 0.7000
received data => -0.3
received data => -0.420
received data => -0.394
received data => -0.160
received data => -0.315
received data => -0.334
received data => -0.329
received data => -0.350
received data => -0.339
received data => -0.325
received data => -0.325
received data => -0.325
received data => -0.325
```

Figure 11. Received real-time ECG data on server

Heart-beat rate (BPM) can be determined after analysis of the data either on the smartphone or the server. In this particular example, it represents a patient's stable condition with a normal heart-beat rate at approximately 72 BPM. This type of data can be diagnostically valuable and easily transmitted for consultations with distant experts.

D. Overall Data Transmission Rate (DTR)

The Data Transmission Rate (DTR) is another important part of the system considering the data size. In our system, DTR depends on the microcontroller, the Bluetooth transmitter and the wireless Internet connection speed. An UART connection between the sensor and microcontroller is established at the baud rate of 115200 bps which achieves a communication bandwidth up to 15KB/s. This emphasizes that our system is capable of the data transmission by integrating multiple types of sensors for a body sensor network system that can be important for patient monitoring, real-time data analysis and diagnosis.

IV. CONCLUSIONS

In this paper, we introduced a system using the smartphone for collecting real-time sensor data and simultaneously streaming the data to the server using Bluetooth and Internet connections. This design is the advancement over ordinary wired sensor networks which are restricted to a fixed monitoring location. In the proposed system, an accelerometer, a temperature sensor and ECG signals have been selected for data transmission using Bluetooth and wireless Internet connections. Having the Bluetooth transmitter on the smartphone, the Android system receives and displays the data on the screen in the graphical or text format and streams the collected data to the central server for data analysis, diagnosis and archiving. Taking advantage of the Android system, NFC technology was used to reduce the unnecessary Bluetooth connection process. This system is highly scalable to include more sensors to produce an upgraded patient monitoring system that is both more accurate and responsive. Furthermore, storing history of collected sensor data in the central server is extremely critical for reliable patient diagnosis.

REFERENCES

[1] M-H Cheng; L-C Chen; Y-C Hung; C-N Chen; C. M. Yang; and T. L. Yang; , "A Vital Wearing System with Wireless Capability," *Pervasive Computing Technologies for Healthcare, 2008. Pervasive Health 2008. Second International Conference on* , pp. 268-271, 2008.

[2] H. Ghasemzadeh; V. Loseu; and R. Jafari; , "Structural Action Recognition in Body Sensor Networks: Distributed Classification Based on String Matching," *Information Technology in Biomedicine, IEEE Transactions on* , vol.14, no.2, pp. 425-435, March 2010.

[3] 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," *Systems Journal, IEEE* , vol.3, no.4, pp. 398-409, Dec. 2009.

[4] J. Yoo; L. Yan; S. Lee; Y. Kim; and H-J Yoo; , "A 5.2 mW Self-Configured Wearable Body Sensor Network Controller and a 12 uW Wirelessly Powered Sensor for a Continuous Health Monitoring System," *Solid-State Circuits, IEEE Journal of* , vol.45, no.1, pp. 178-188, Jan. 2010.

[5] I. Korhonen; J. Parkka; and M. Van Gils; , "Health monitoring in the home of the future," *Engineering in Medicine and Biology Magazine, IEEE* , vol.22, no.3, pp. 66- 73, May-June 2003.

[6] H. Tanaka; R. Kimura; and S. Ioroi; , "Equipment Operation by Motion Recognition with Wearable Wireless Acceleration Sensor," *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International Conference on* , pp. 114-118, Sept. 2009.

[7] I.H. Mulyadi; E. Supriyanto; N.M. Safri; and M.H. Satria; , "Wireless Medical Interface Using ZigBee and Bluetooth Technology," *Modelling & Simulation, 2009. AMS '09. Third Asia International Conference on* , pp. 276-281, May 2009.

[8] MSP430BT5190 Mixed Signal Microcontroller, Texas Instruments, April 2010

[9] Android Developers. (2007) Snake. [Online]. Available: http://developer.android.com/resources/samples/Snake/index.html