Advances in Technology Innovation, vol. 3, no. 3, 2018, pp. 118 - 125

A GI Proposal to Display ECG Digital Signals Wirelessly Real-time Transmitted onto a Remote PC

Marius-Corneliu Rosu[1,*], Hamed Yassin-Kassab[1], Adriana Iliesiu[2], Serban Georgica Obreja[3]

[1]Department of Applied Electronics and Information Engineering, University Politehnica of Bucharest, Romania.

[2]Department of Cardiology, University Carol Davil of Bucharest, Romania.

[3]Department of Telecommunications, University Politehnica of Bucharest, Romania.

Received 21 July 2017; received in revised form 26 September 2017; accepted 17 December 2017

Abstract

The sensors, as wireless communication system, comply the 7-layer model Open Systems Interconnection (OSI). In this paper, a point-to-point transmission model was used. The ECG signal is transmitted from the Router Sensor (RS) to an end Coordinator Node (CN) plugged-in to the laptop via USB port; RS acquires ECG signal in analogical mode, and is also responsible with sampling, quantization and sending it wirelessly direct to CN. The distance between RS and CN is a single-hop transmission, and does not exceed the range of the XBeeS2Pro transceivers. The communication protocol is ZigBee. Remote viewing of the transmitted signal is performed on a Graphical Interface (GI) written under MATLAB, after the signal has been digitized; the choice of MATLAB was motivated by future developments. Particular aspects will be highlighted, so that the reader to be edified about the results obtained during laboratory experiments. Recording demonstrate that the purpose exposed in title has been reached: Direct link in Real-Time was established, and the digital ECG signal received is reconstituted accurately on MATLAB GI; signal received on laptop is compared with the analog signal displayed on oscilloscope.

Keywords: GI, real-time ECG, wireless-transmission, ZigBee, MATLAB

1. Introduction

Remote viewing of a digital signal is performed on a Graphical Interface (GI), written in a programming language, enabling the entire signal overview and detailed views. Visualization by GI, it showed to be faster than standard methods. Different levels of visualization and fast navigation in bio-signals are essential conditions for a GI. Many expectations are demanded from specialists when is used a software for more tasks. The truth sometimes looks quite different: The software does not work as is expected, and because of some bugs unpleasant consequences and unwanted effects will occur.

GI is usually the main interface; the only one can say between the user and the given system. Traditionally testing with GI is done manual involving money and time; that's why the results are often unreliable. So better for testing are specialized software which can avoid these problems and time period to enter on market can be shorted. Thus, scripting language for GI should allow rapid testing and debugging as well as the possibility to check GI with dedicated programs.

* Corresponding author. E-mail address: marius-corneliu.rosu@sdettib.pub.ro

1.1. Graphical Interface Design Issues

Engineering for GI becomes a subject in the software industry starting with 90s. GI had replaced the Text-Based Interface (TBI) with one based on graphics, and for users it becomes more intuitive and easier to work with. In [1], it is shown the need to transpose the TBI to the GI platform in an automatic way. This must include the GI builder and compiler code for event-driven language C/C++. Applications that use GI are created based on processes requirements, and during development these may change. Typically, GI are modeled as state machines defined by windows, widgets, widgets parameters and values; usually shaping GI starts from implementing requirements which a system must meet.

Long-term bio-signals' acquisitions are the main sources of informations about the patient's condition. Hence, a broad set of issues into a GI design must be harmonized. Modeling was also an important obstacle in the tests performed on the basis of patterns because they were not used more intensively in the industry. It has been shown, however, that in well-defined applications, models generation methods can be used with good results. Consequently, the automatic generation of models has become widely used in designers thinking. Various techniques used to generate models of GI are synthesized in Table1. For a more thorough documentation, the reader is guided to consult the reference [2].

In Table1, most techniques require human intervention; the effective implemented solution largely depends on the degree of human intervention. Stages of drafting, in short, are the following:

(1) Design and modeling of the main components for optimal response of the Monitored System (MS).

(2) Establishing of minimum parameters required to cover the MS.

(3) Generate automatically if possible with a predetermined method (e.g. supervised learning) the pattern of algorithms and test parameters.

(4) Testing the Monitored System by running the generated patterns and parameters.

To explore the events, GI inputs must be stimulated while outputs return the results; also, the created GI for the MS, permit to check the implementation (software or hardware) for the system requirements.

2. Signal Consideration

Various technologies are needed to develop an effective GI as discussed in the previous chapter; design and development require a multidisciplinary approach. Long-term acquisitions of bio-signals are important; they constitute the main source of information on the patients' condition. Visualization tools are also important: They allow the inspection of complex bio-signals in a fast manner and friendly use.

In terms of scope, this paper aims to depict the possibility of using the programs dedicated mainly to signal processing (MATLAB this case). In terms of variety, bio-signs are impossible to classify rigorously: the only certainty is that these are with very low values (the useful ECG signal, between -2 and +2mV) , while interferences, artifacts, and noises, distort them, having a major influence (e.g. electrods offset level between -300 and +300mV). A bio-signals classification is not universally valid. In Fig. 1, we drew a classification attempt, and Fig. 2 illustrates once again that it is practically unachievable for a strict and unique classification.

From these, ECG signal is graph transposition of heart electrical activity, and the mechanical activity of the heart is somewhat regular with the exception of being patients with Cardiovascular Disease (CVD), so ECG may be considered a quasi-periodic waveforms. Particularities can be summarized as follows:

(1) ECG is ideally for electronic reporting

(2) ECG signals are weak;

(3) Bad influences have interferences, artifacts, and noises;

(4) Bandwidth is considered depending on the intended purpose, monitoring or diagnosis

(5) Some areas of the ECG signal frequencies must be preserved;

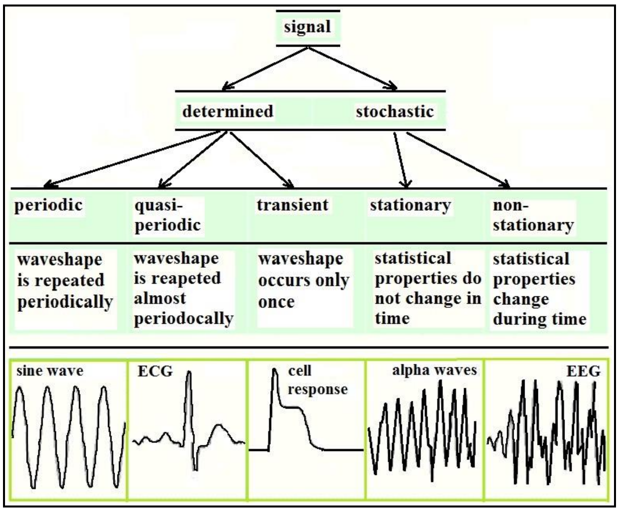

signal
determined
stochastic
periodic
quasi-periodic
transient
stationary
non-stationary
waveshape is repeated periodically
waveshape is reapeted almost periodocally
waveshape occurs only once
statistical properties do not change in time
statistical properties change during time
sine wave
ECG
cell response
alpha waves
EEG

Fig. 1 Bio-signals wave shapes attempt classification

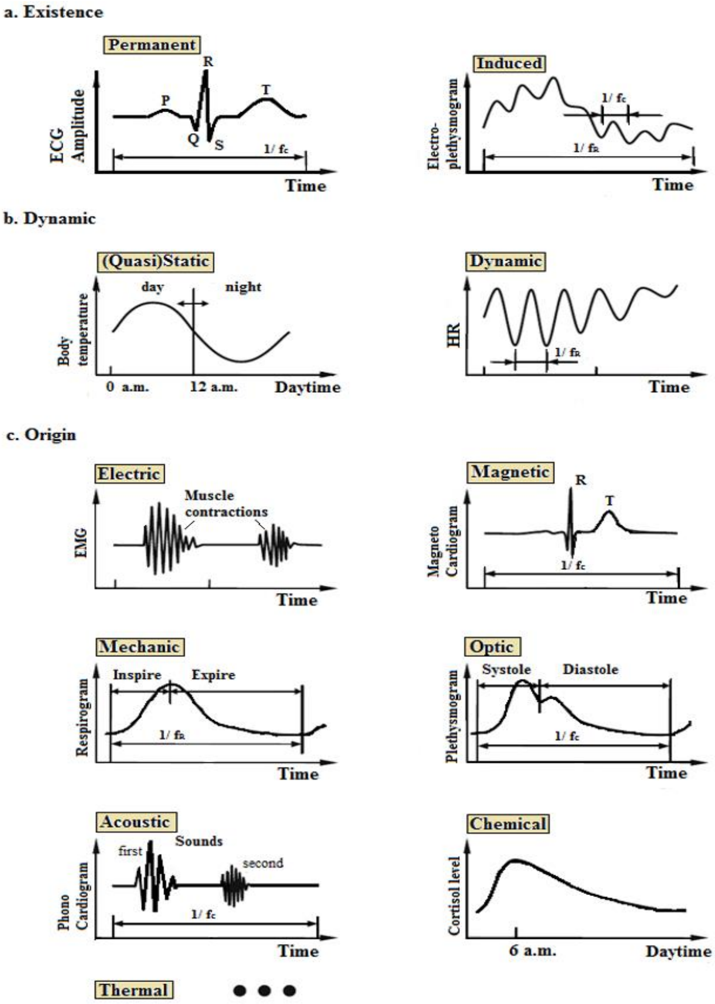


Fig. 2 Second approach for Bio-signal classification (after [3])

Note: *In figures are indicated, heart rate f_C, respiratory rate f_R, and other additional information*

Table 1 Techniques for generating GI Models

Task.	**GI Models**		
	Technique name	**Description**	**Remarks**
1	*Manual modeling*	Modeling the test system; Defining the required test area; Control of entry, while on exit is observed the events using test scripts; The model represents the real situation and test scripts that generates him; System testing by the so-called compliance testing	Requirement specifications, interface, and so on, should accurately describe the behavior of the system; Requirements are shown in natural language by texts and diagrams; Just persons with analytical abilities can create a formal and structured model
2	*Automatic modeling*	The initial objectives of model are reduced; Enables implementation of the compliance testing requirements;	The model is basically a reverse engineering task; Implementation is the source for redesign of the work; Can reveal bugs that cause accidents in implementations
3	*Semi-automatic modeling*	Modeling techniques, performed most part automatically	Are assisted manually by a person during the process; Revised, corrected and extended manually after generating the initial model.
4	Reverse Engineering	Extract design artifacts, and derive abstractions less dependent of implementation.	There are two approaches: *static*, analyzing source code or binary representation, without running the system, while *dynamic*, examines the external behavior of the system by running it.
5	Static Analysis	Built for important elements of source code figure behavioral and structural;	Several issues make the static analysis a complicated task: Inheritance, polymorphism, and dynamic binding, make difficult to know which methods are going to be executed; that's why For this, the method was not successful in generating patterns for GI testing purposes.
6	Dynamic Source Code Instrumentation analysis	During the execution of a system, observations are made by instrumentation instructions implemented as source code, which can be on GI controlled and observed	Source code is the best way to get information, because the level of abstraction and sources of information are controlled by reverse engineering;
7	Dynamic Byte code analysis	It is used to avoid repeated operations	Less invasive than source code can be used in Java
8	Dynamic GI APIs	The technique is used to collect information on GI status, and to control the GI from one state to another	The model was created for understanding the behavior of the original GI, in aiming to build the correct shell; It is used an independent source code for interaction.
9	Manual GI exploration techniques	Explore manually by the user the implemented GI, while the background process record user actions.	All capture or playback GI tests tools, utilizes manual methods of exploration; Model is used to generate test cases able to find errors.
10	Fully automated GI exploration techniques	Generation of GI testing models; GI models are obtained as well as runtime behavior automatically.	The process works in ripping mode, starting with first GI window and continuing by opening all other sub-windows; The behavior of the structure and execution of GI are extracted.
11	User-assisted GI automated exploration techniques	It is a mix between automatic exploration and user-assisted manual exploration.	It consists of three phases: 1. Extracting structural information about each GI window; 2. interaction with GI; 3. Takeover of behavioral information through automatic process. The model is used for abstract cases.
12	Feedback GI Refinement	All the processes described above can be improved by this technique; This technique requires that an initial test case or suite to be created, manually or automatically, and to be run on software.	Various sources of feedback are possible, e.g. the code coverage report. Feedback serves to enhance the GI model, allowing for better test coverage and test generation with lower costs of execution

3. Motivation and Previous Achievements

The increasing of hardware applications in medicine had generated the issues of visualization into a manner that can be understood by the human brain; signal analysis in medicine has now become a method of interpretation and research since it can offer feedback on the patient's condition or state of the experiment by processing it. So, viewing and processing techniques involve researchers or clinicians direct in selecting interesting parts of analyzed bio-signal. The acquisition on a longer period of time is a method that allows continuous monitoring, and was proposed in [4]; the method has, however, barriers consisting of vast amounts of end point with double column. It becomes important for remote monitoring continuous Real-Time view. Design of visualization tools is related to a correct choice of data information storage structure; signals must be recorded in a format that gives possibility for *posteriori* data accessing.

Several standards for storing biological data and inter-communication have been developed over time. A standard format that allows saving the data on multiple channels is European Data Format (EDF), and it is described in [5, 6]. Division of Health Science and Technology from Massachusetts Institute of Technology (MIT) developed research resources for an online investigation that can be applied to complex biological signals [7]; it consists of the following main components:

(1) *PhysioBank* is the online database of physiologic signals;

(2) *PhysioToolkit*, is the bio-signal analysis tools: The ECG data stored in EDF format can be viewed in the EDF browser, an open source tool that allows signal processing, such as filtering, power spectrum, or heart rhythm;

(3) *PhysioNet* the web forum [8];

The GI's requirments for viewing and processing of the long term bio-signals are not easy to be achieved. Other ideas in visualizing large data sets, but not for bio-signals, is web mapping services consisting in a multilevel visualization of an very large datasets; another area requiring long-term visualization are algorithms for data mining [9].

Displayed solutions described above refer to portions of signal finite in time. For bio-signs with virtually infinite duration, ECG in this paper, visualization imposes a new concept of GI' s that have to display in real-time the monitored signal wirelessly transmitted.

4. The Concept

Biological signals can be acquired through various methods. As was mentioned, visualization, analysis, and processing, are essential in long term monitoring. In this paper, we intend to see a new possibility for building a GI by using other resources than the classic ones.

4.1. Data file formats

It should be recalled once again that in long term monitoring, the amount of received information is enormous. Also, the data format in a standardized mode will facilitate cooperation and exchange of data between centers; a brief list of formats is described in further:

- ***.txt**: Text, known by users as computer files, with a wide use, those are basic file system; can be accessed and modified thru text editors. ***Disadvantage:*** *slow access in data*;
- ***.edf**: EDF is just a basic format to archive and exchange bio-signals; sampling frequencies and physical sizes are not imposed. ***Disadvantage:*** *implementation of EDF unit is time consuming task ;*
- ***.dat**: PhysioBank is characterized by the archive of physiologic signals recorded digital. Each Physio-Bank is possible to have more than one record divided in three files: **.hea file* - header information; *name or URL* - Short text file describing signals; **.dat* – binary signal file containing digital samples;
- ***.h5**: **HDF5** - Hierarchical Data Format 5 for storing and managing data with simple structure that include: *Datasets* and *Groups;*
- ***.mat**: MATLAB format, the most known in biomedical engineering area; data are saved in binary. Even though is not Open Source, it is a powerful tool;

Since ECG acquisition we made is under MATLAB format, the idea was to try to realize a GI also under MATLAB tool.

4.2. Basis idea

As the **'tmtools'** command in MATLAB is used to establish the communication via USB port between an oscilloscope and a PC, we began to wonder if a similar issue can be applied to a CN plugged by USB onto a Remote PC, which acts as server and workstation for the received signal. The only requirement self-imposed is the ability to view in Real-Time the signal received which is transmitted in digital domain between a *Router* and a *Coordinator* node; the last one is connected to the remote PC via a USB port.

It is known that MATLAB provides programs and basic commands that can be modified for the aimed purpose. Thus, if the ECG samples from oscilloscope acquisition were made possible via the USB port in MATLAB, the same thing was wanted and in reverse operation, to display the signal wirelessly received by the *Coordinator*, in a GI written under MATLAB; the choice was motivated by flexibility and adaptability to future developments.

For this, however, we are required the following checks:

(1) If the serial device send data in "lines", through a line terminator character;

(2) How many bytes is expected to be read each time by the serial port;

Verification of the above can be done by following the instructions set forth in [10]. Since most USB drivers include the ability to act as a *"Virtual Serial Port"* or *"Virtual COM Port"* (VCP), this feature can be used to establish communication with the MATLAB interface.

In Windows, drivers for VCP are sometimes a separate feature or require additional installation steps; once installed, *serialDataStream* function is responsible for receiving and processing a continuous stream of data from a device connected to a USB. Some aspects must be specified so that the reader to be cleared up about how the connectivity can be established between *Coordinator* and PC via USB interface:

(1) First, it should be remembered that due to micro-processor ADC module in Sensor node, wireless transmission it will be made in the digital domain;

(2) In MATLAB must be connected an object with fopen(obj) command, so that serial communication with Coordinator via USB to be possible;

(3) ***obj*** from command ***fopen(obj)*** is defined at least by the parameters: '#COM port', 'BaudRate', and 'DataBits';

(4) Before running the script should be checked which '#COM port' was allocated to Coordinator, otherwise MATLAB will not know how to communicate, displaying an error in Command Window;

Further, writing in MATLAB of GI is given by the instructions of reference [11].

4.3. Wireless test results

Setting and configuration of sensors are important; sensor dedicated to ECG signal was designed for ZigBee protocols communications. The channel access technology method was preferred to connect the wireless devices "point to point"; MAC protocols thru data-link layer are responsible to allocate resources for this task (OSI level 2). Fig. 3 shows that used transceivers were wirelessly connected, and Fig. 4 shows the Coordinator hardware connected directly to the laptop via USB port; the appropriate MAC address also can be readable. Settings were made also in the XCTU configuration software.

A direct link has been established in real time between sensors; Fig.5 represents snapshots after recording during laboratory experiments, and the received signal is reconstituted accurately on GI written in MATLAB. On router AT mode setting was made, while for coordinator API mode setting was chosen; additional information and a quick reference guide for XBeeS2Pro transceivers used can be downloaded from [12].

On the other hand, energy consumption must be well controlled; controllers and radios devices are the main consumers while memory and sensors are in lower position. Those demands come because of the generic implemented solutions limited efficiency, and without taking into account the requirements of limited energy resources, Wearable Systems cannot be developed.

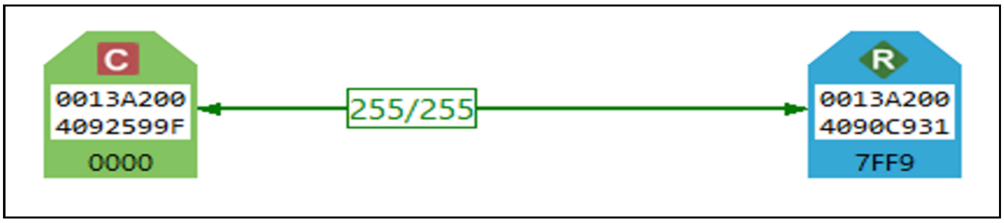


Fig. 3 XBeeS2Pro link connection

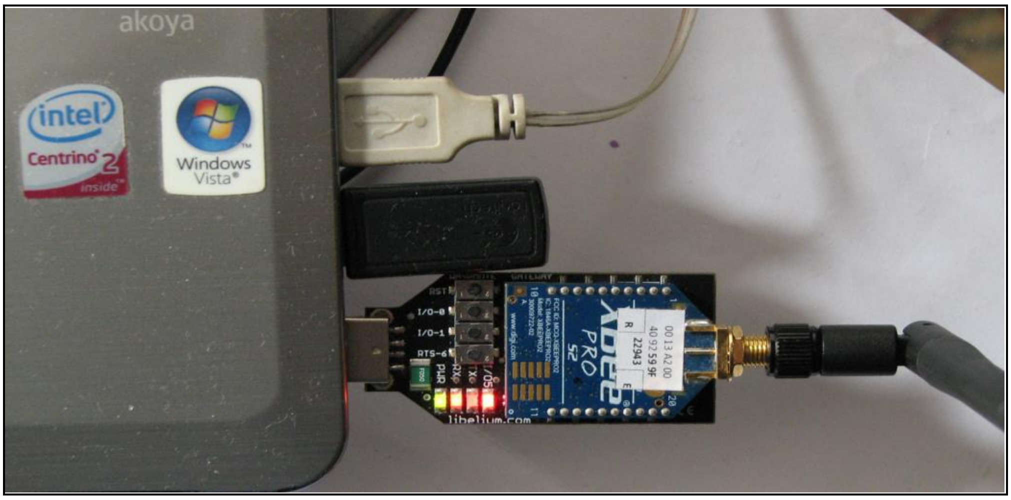
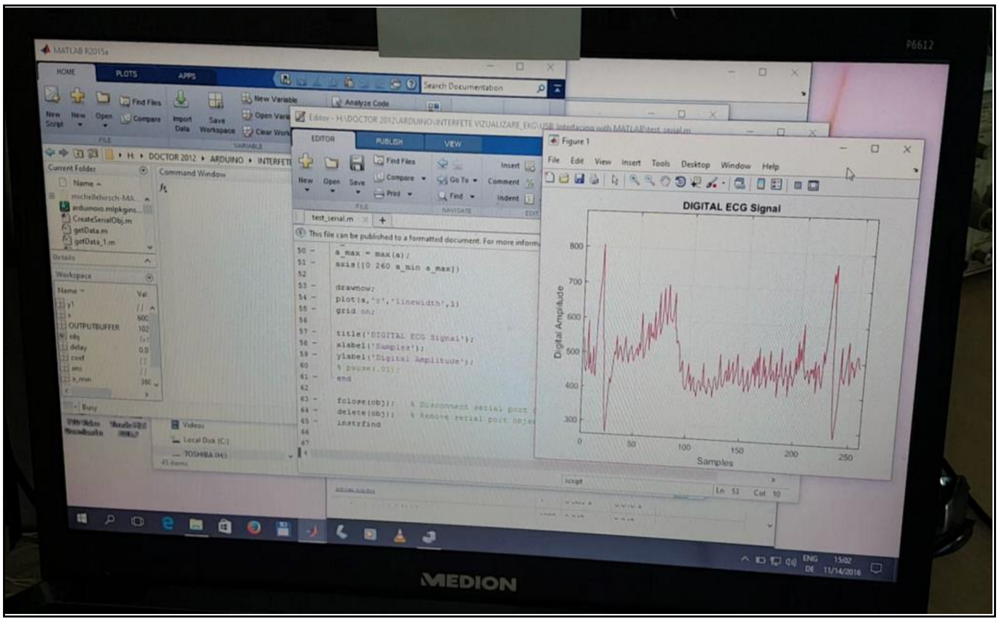


Fig. 4 Coordinator Node plugged into laptop



```
a_max = max(a);
axis([0 260 a_min a_max])

drawnow;
plot(a,'r','linewidth',1)
grid on;

title('DIGITAL ECG Signal');
xlabel('Samples');
ylabel('Digital Amplitude');
% pause(.01);
end

fclose(obj);    % Disconnect serial port
delete(obj);    % Remove serial port obje
instrfind
```

Fig. 5 Photos of laptop screen; ECG signal can be distinguished

5. Conclusions and Future work

Considering the problems with the viewing and processing bio-signals in the long term, and the importance of ECG signal in healthcare. This paper has presented A GI proposal to display ECG Digital Signal wirelessly transmitted onto a remote PC using MATLAB toolbox. Some requirements should be considered so that the communication to be established successfully via USB; they were detailed in Chapter 4.

Further, writing and developing of GI will be made depending on the complexity and number of determinations that are intended to be made in real time. So, viewing the transmitted digital signal must be performed on a GI which must be friendly and easy to use. At the same time, the end user must have the opportunity of immediate determinations of the main characteristics of the signal (e.g. RR distance in ECG). Also, future GI improvements should enable the designer to carry out immediately at minimal cost.

Channel access technology method was preferred for point to point wireless connection; data-link layer thru MAC protocols has the responsibility to allocate resources for this task (OSI level 2). Thanks to flexibility and adaptability, programs and functions for an automatic processing routine in parameters of interest in ECG signal will be added.

This paper discusses traditional methods for GI and proposes a new approach of a program dedicated more to processing and less for interfaces. The challenges in this direction remain undetermined, the rate of renewal of wireless technologies b eing very dynamics; tiny devices are able to improve the daily life and to ensure a better monitoring of patients.

References

[1] G. Antoniol, R. Fiutem, E. Merlo, and P. Tonella, “Application and user interface migration from BASIC to Visual C++,” Proc. 11th International Conference on Software Maintenance, 1995, pp. 76-85.

[2] A. Kull, “Automatic GUI model generation: state of the art,” Proc. IEEE 23rd International Symp. Software Reliability Engineering Workshops, November 2012, pp. 207-212.

[3] E. Kaniusas, “Biomedical signals and sensors I,” Chp. 1 Biological and Medical Physics, Biomedical Engineering, Berlin: Springer-Verlag, 2012.

[4] H. Kayyali, S. Weimer, C. Frederick, C. Martin, D. Basa, J. Juguilon, and F. Jugilioni, “Remotely attended home monitoring of sleep disorders,” Telemedicine and e-Health, vol. 14, no. 4, pp. 371-374, May 2008.

[5] “EDF-European Data Format,” http://www.edfplus.info/.

[6] B. Kemp and J. Olivan, “European data format “plus” (EDF+), an EDF alike standard format for the exchange of physiological data,” Clinical Neurophysiology, vol. 114, no. 9, pp. 1755-1761, September 2003.

[7] G. B. Moody, R. G. Mark, and A. L. Goldberger, “PhysioNet: a Web-based resource for the study of physiologic signals,” IEEE Engineering in Medicine and Biology Magazine, vol. 20, no. 3, pp. 70-75, May 2002.

[8] “PhysioNet - the research resource for complex physiologic signals,” http://www.physionet.org/.

[9] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana. “Time-series bitmaps: a practical visualization tool for working with large time series databases,” Proc. of SIAM International Conference on Data Mining, April 2005, pp. 531-535.

[10] “Warning: unsuccessful read: matching failure in format,” http://stackoverflow.com/questions/35671418/warning-unsuccessful-read-matching-failure-in-format, February 29, 2016.

[11] M. A. Hopcroft, “serialDataStream,” https://www.mathworks.com/matlabcentral/fileexchange/31958-serialdatastream/content/serialDataStream.m, March 20, 2012.

[12] “XBee S2 Quick Reference Guide,” https://www.tunnelsup.com/xbee-guide/, November 30, 2012.