Real-Time Biosignal Acquisition and Telemedicine Platform for AAL based on Android OS

Hugo Silva[1,2,3], André Lourenço[1,2,4] and Nuno Paz[3,4]

[1] Instituto de Telecomunicações, Lisboa, Portugal

[2] Instituto Superior Técnico, Lisboa, Portugal
[3] PLUX Engenharia de Biosensores, Lisboa, Portugal

[4] Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal

Abstract. Among other strategic action points, the Ambient Assisted Living Joint Programme framework has defined telemedicine and remote monitoring as major applications for improved quality of care. With the advent of novel, game-changing mobile platforms, the technological basis is now in place to provide more capable and usable systems for this practice. In this paper, we present a real-time biosignal acquisition and telemedicine platform based on Android OS, as both the system and the available handsets present highly desirable features for the field. The proposed platform was used in the context of a continuous real time monitoring of ECG signal showing one simple example of its applicability.

1 Introduction

Telemedicine and mobile monitoring of health, and other related parameters, are referred as one of the major applications toward improved quality of care and independent living in an Ambient Assisted Living (AAL) framework [1]. This is a rapidly growing field within clinical medicine, and modern Information and Communication Technologies (ICTs) are providing new opportunities, through the introduction of innovative ways of monitoring wellbeing and conveying information both to healthcare professionals and users [2].

Although the history of telemedicine can be dated back to the Middle Ages, where bonfires were used as a way of transmitting information about the bubonic plague throughout Europe, only recently, with the advent of wireless and mobile communication technologies, novel tools have been made available that are able to provide more effective remote monitoring and telemedicine methods and systems [3].

These systems are useful to assist the general population, however they are particularly important in severe cases where the fact that the patient is unattended among visits to the practitioner and/or moving the patient between his/her home and the hospital or healthcare centers represents a great ordeal; Chronic Obstructive Pulmonary Disease (COPD) and Amyotrophic Lateral Sclerosis (ALS) are just a few examples of such cases [4, 5].

Silva H., Lourenço A. and Paz N..
Real-time Biosignal Acquisition and Telemedicine Platform for AAL based on Android OS.
DOI: 10.5220/0003303801110121
In *Proceedings of the 1st International Living Usability Lab Workshop on AAL Latest Solutions, Trends and Applications* (AAL-2011), pages 111-121
ISBN: 978-989-8425-39-3

Smartphones in particular, are becoming a *de facto* tool for telemedicine applications as they evolve to become always-on, always-connected devices [6], with increasingly higher computing capabilities and intuitive user interfaces [7]. Together with biosignal acquisition and sensing devices, current ICTs have the potential of enabling practitioners to remotely access a comprehensive set of follow-up and diagnosis parameters.

In this paper we present a telemedicine platform for real-time biosignal acquisition, display, and transmission based on Android OS. Within the existing mobile platforms landscape, Android OS is gaining particular attraction due to its hardware interoperability, versatility, and usability characteristics. The proposed platform was devised as a tool for ambulatory and/or home-care within an AAL framework, providing: a) interface with measurement instrumentation through Bluetooth; b) local data acquisition and visualization; c) local data compression and storage; and d) data transmission to a remote server over TCP/IP.

The rest of the paper is organized as follows: Section 2 provides an insight on the motivation behind the use of Android OS; Section 3 describes the related work on this topic; Section 4 details the proposed platform; Section 5 describes an application of this platform to the remote monitoring of hear rate data; and finally, 6 outline the main results and discussion.

2 Mobile Platforms

2.1 Landscape

A wide variety of mobile development platforms is currently available, of which Symbian, Blackberry, iOS, Android, J2ME, and Windows Mobile, currently represent the mainstream. Among these, the first four have the most representative market shares [8]; each has its own singularities, namely:

a) **Symbian** is currently the most widespread platform, with a share above 40% of the smartphone market today. Despite the fact that it provides full hardware access and that applications are programmed in a variant of the highly efficient C++ language, Symbian lacks a unified structure across editions. Among other aspects, this forces applications to be specifically written for each user interface, and even binary installers need to be revised, as different editions largely differ between them;
b) **Blackberry** devices are mostly know by their e-mail capabilities. Although the OS and applications are mostly supported in Java, and data transmission is very well supported, RIM's proprietary APIs must be used in order to take advantage of the device capabilities; this severely limits the developed applications to Blackberry handsets. Furthermore, this platform was initially developed for keyboard-based operation, which makes user interaction slightly below that provided by platforms designed taking into account touchscreen devices;
c) **iOS** is featured in what can be considered the most desired consumer electronics devices in the market such as the iPhone, iPad, and iPod. Recognized as the best user experience in its class and with great computing capabilities, the iOS has a severe handicap development-wise as application development is subject to Apple's

approval and the standard set of APIs does not allow low level access to the device resources (e.g. Bluetooth);

d) **Android** was developed by Google and it is available under an open source license. Specifically designed for touchscreen devices, this platform is based on a modified version of the Linux kernel. While the Android platform is built upon Java applications, it does not use the established Java standards as J2SE or J2ME, this being the main drawback of Android as it limits the compatibility between existing applications written for these platforms and the Android OS;

e) **J2ME** debuted as a subset of the Java language, with a reduced set of APIs targeted at mobile phones and other devices that implemented the Virtual Machine. This platform is available on nearly every handset and application development follows the philosophy of write once, run anywhere making portability a keystone of this platform. Nonetheless, the J2ME APIs were initially designed for connection limited devices, with small displays and reduced computing power, therefore limiting the developed applications capabilities. Furthermore, applications are bound to the VM, which greatly conditions the access to core hardware features;

f) **Windows Mobile** belongs to the first wave of mobile platforms for smartphones ever to appear; designed to mimic in some extent the environment of the desktop versions of Windows. Although highly robust and versatile, since applications are programmed in C++ and the APIs provide full hardware access, Windows Mobile has not been able to adequately keep up with the competition regarding usability, and changes among versions of the OS generally do not take backward compatibility into account.

2.2 Outline

In all platforms, except Android and Windows Mobile, applications must either be digitally signed or undergo a thorough approval process by the manufacturer prior to their deployment or for being made available to end-user. Still, after these processes, the access to core resources is typically restrained or sand-boxed, limiting the functionality of the applications and/or requiring a great deal of attention from the end-user to grant permission in the operations considered most sensitive in terms of resource access. In either case, the user experience, functionality, and attainable automation levels are, in general, highly restrained.

Android however, is an open source platform, currently supported by a wide range of devices, it is hardware independent, and greatly focused on interaction via touch reducing the learning curves. As of 2010, over 70 devices from different makers can be found on the market with a wide price range and most of them with large touchscreen displays; these allow users with different cognitive and motor skills to use the device as screens can be easily adapted to the end-users needs.

Given this scenario, our research found the Android platform to be the most versatile in an AAL framework. Among other advantages regarding its competition, Android provides: a) Advanced UI with touch-driven interaction; b) Portability; c) Exposed APIs for Bluetooth radio access; d) Unrestrained hardware access through unified APIs to access the different services; e) Multihoming over WiFi or cellular network (2G/3G); and f) Unprecedented security and application permissions management policy.

3 Related Work

Other authors have proposed approaches for telemonitoring and real-time signal acquisition from biosignal measurement devices through mobile platforms. [9] proposes telemedicine system for home care and patient monitoring based on a mobile phone. In their approach, the mobile phone collects the data from existing monitors through a cabled RS232 interface and then sends them at distance through TCP/IP for a central computer. Furthermore, the client application does not provide local feedback to the user and is based on Java MIDP, initially designed for devices with reduced interface capabilities and with severe security restrictions for hardware resources access, which in the current landscape may not be the most adequate platform in a AAL framework.

In a work by [10], a GSM-based, mobile telemedicine platform for blood pressure monitoring is presented. This system is based on a wireless platform, that integrates a Bluetooth receiver and a GSM module; the Bluetooth connectivity allows the wireless data acquisition from a Bluetooth-enabled sphygmomanometer, while the GSM module provides the cellular network access layer for data transmission to a remote server. While dedicated hardware devices can be highly optimized for power and efficiency, they're scalability and versatility or AAL applications is generally highly limited. They perform well the functions for which they were designed, but new features require firmware or hardware changes which are difficult to implement on new devices and to propagate among devices already in use.

A remote patient monitoring system using a Java-enabled 3G mobile phone is presented in [11]. Designed for bedridden patients, this system allows doctors to remotely monitor their patients through a mobile phone; patients are connected to standard bedside monitors, which in turn relay the data to a remote information server through a central monitoring station. Through a 3G mobile phone, doctors can use a Java application to access the remote information server and monitor at distance multiple parameters from their patients. For continuous monitoring in an AAL framework, devices should perform their measurement functions without conditioning the individuals throughout their regular activities and daily life, in which case, the fact that it relies on bedside monitors limits the practical applications of this system.

[12] have devised a multipurpose mobile platform for ubiquitous personal health information assessment, that can be used for a wide range of applications. It is an open source framework that includes a set of interfaces for communication with devices, storage, network connectivity and user interface. The mobile application is able to collect data from Bluetooth end-devices, provide local feedback with summary data, and stream it through 3G/GPRS network to a remote server. Although extremely complete and modular, this platform is based in the Java MIDP framework.

Recent work by [13] has described a smartphone based telemedicine system for recording limited lead body surface potential maps by means of a Bluetooth-enabled, 12-lead ECG end-device. A software application developed for Windows Mobile and the .NET compact framework collects the data in real-time from the end-device and stores it on a removable solid state memory card. Among all smartphone platforms, Windows Mobile is currently among the group with lowest market shares; moreover, most handsets designed for this OS require a stylus for proper operation as the UI was not designed to take advantage of the modern devices features. For AAL applications more user-friendly and intuitive platforms are required.

Previous approaches to this problem have explored different solutions and platforms. We address this problem in a novel approach, through the Android OS mobile platform and the integrated capabilities provided by modern smartphones, whose advantages were previously commented. In its current version, our system interfaces with a general purpose biosignal acquisition hardware platform, nonetheless, we devised a software-based architecture with plugin end-device and processing blocks. Furthermore, combined with high-level facilities provided by the OS such as the Android Market, the proposed platform presents increased versatility for AAL applications since it can be remotely updated with new features.

4 Proposed Approach

4.1 System Overview

Figure 1 presents an overview of the global architecture for the proposed approach. Sensors (A) are applied to the subject with the purpose of simultaneously monitoring one or more parameters. An end-device (B), the acquisition unit, performs the analog-to-digital conversion and data encoding of the biosignals coming from the different sensors. Furthermore, the end-device (B) provides Bluetooth wireless connectivity to the smartphone (C) based on the Serial Port Profile (SPP).

The data streamed by the end-device (B) is continuously collected and decoded in real-time on the smartphone (C) which performs storage, instant display, and transmission to a remote database (D) where the data becomes available for observation by healthcare professional and other designated caregivers. In the smartphone, algorithmic transformations are also performed over the signals, which include data compression (based on deflate algorithm), and signal processing operations consisting of transfer functions computation and automated biosignal modeling.

Local real-time monitoring of incoming data is shown in the smartphone (C) screen to provide local feedback to the user, based on our UI screen architecture; also, the data is stored in standard ASCII format on the local memory. When a connection to the remote database (D) is available, the stored data is then compressed for network payload optimization, and streamed in real-time to the remote database (D) using an atomic transaction model.

For data transmission at distance, our platform uses the multihoming capabilities of Android OS. A connection to the remote database (D) for data transmission is performed through standard TCP/IP protocol over an established WiFi, 2G/3G cellular network connection. Since current devices and associated mobile providers data plans are designed in an always-on, always-connected approach, our platform is able to provide near-real time continuously updated data at the remote server. The remote server (D) is able to handle incoming concurrent connection from multiple smartphones.

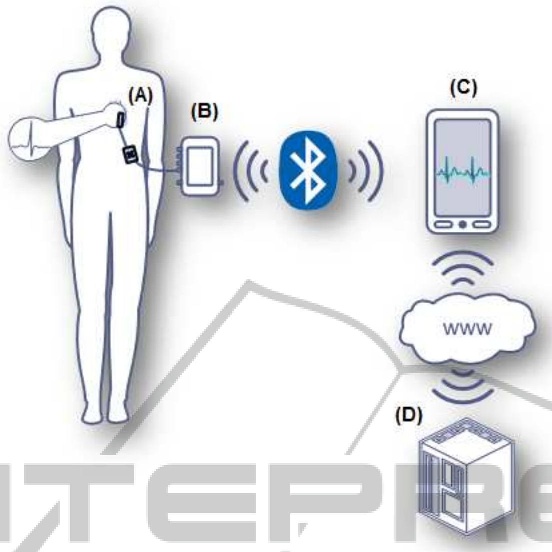


Fig. 1. Global system architecture overview.

4.2 Framework

Our framework provides a comprehensive set building blocks for the development and deployment of flexible and versatile telemonitoring tools. We provide core components and interfaces for controlling the mobile application at different levels:

a) **End-device Communications**, through a communications component that is responsible for establishing and managing the communication with the acquisition unit over bluetooth;
b) **Signal Acquisition**, through an end-device dependent component, that decodes the collected signals; an interface enables the abstraction of the acquisition unit and addition of complementary or alternative devices;
c) **Signal Processing**, through a component that implements the conversion algorithms, taking the signals coming from the device as input and providing the results from the processing to the presentation layer;
d) **Data Presentation**, which is the core component of the framework; it receives data from the acquisition and processing components, stores them persistently in the solid state (SD) memory card, and enables the visualization of the the selected channel in the user display;
e) **Data Compression**, which is performed using the components with the same name; for compression we the selected the deflate algorithm (used by gzip), that uses a combination of the LZ77 algorithm and Huffman coding [14].
f) **Transmission Over TCP/IP**, through a component that takes the compressed data and transmits it to a remote server.

In figure 2 is illustrated the class diagram, where: IDevice is the interface that defines the functionality for communication with the end-devices and signal acquisition;

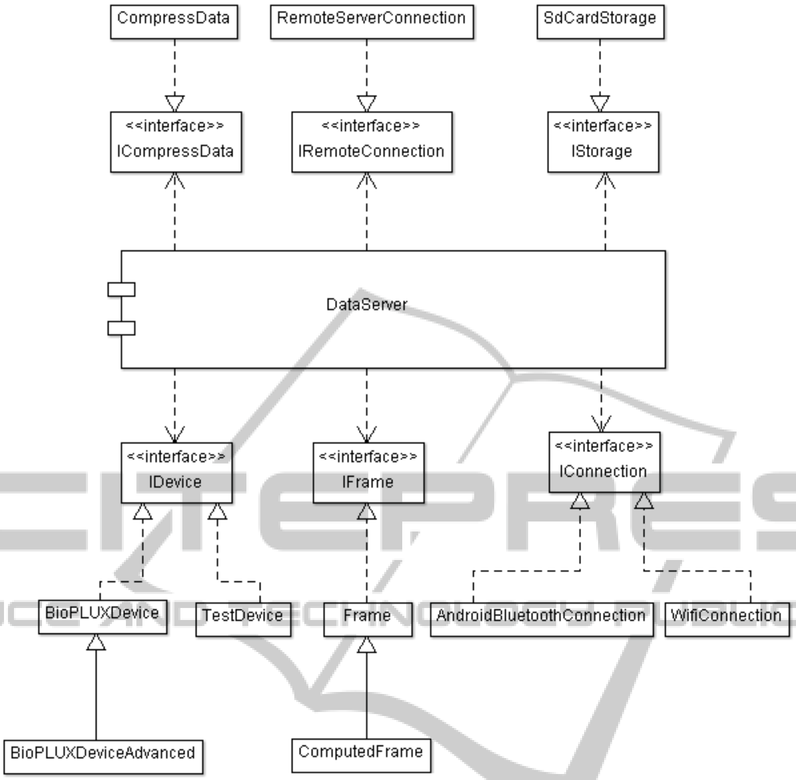


Fig. 2. Simplified class diagram.

IFrames derived objects store each collected data sample; IConvertionAlgorithm is the interface that defines the structure for signal processing blocks; DataServer is the object responsible for the data presentation; IStorage is an interface that defines the SD card data storage procedures; ICompressData is the interface that defines the data compression mechanisms; and finally IConnection and IRemoteConnection are the interfaces that define the data communication and associated mechanisms.

4.3 Mobile Application

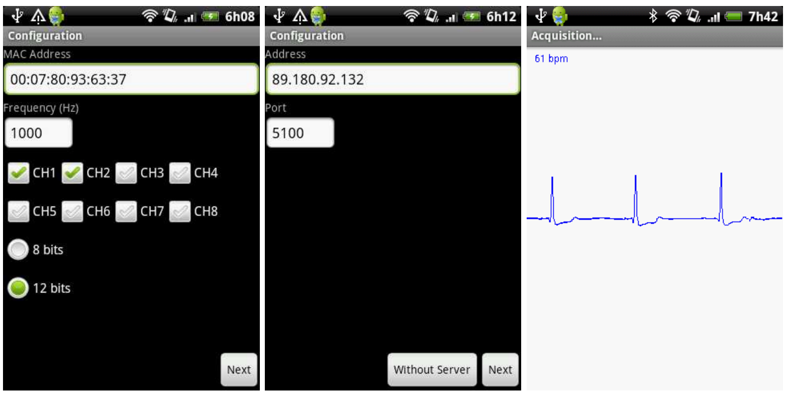
The mobile application requires Android OS version 2.0 or above, a handset with SD card storage capabilities, and WiFi or 2G/3G cellular network connectivity. It has been deployed and tested in the HTC Hero, Google Nexus One, and Samsung SPICA handset; the application runs in all devices, and has been tested successfully for all features in the first two devices. In the case of Samsung SPICA, an unresolved firmware problem with the Bluetooth interface prevents the application from executing with all features, namely the connection to Bluetooth wireless end-devices.

The developed application (see next section) has a footprint of 59KB.The generated uncompressed ASCII files with raw data that are stored in the SD card have a 199 byte header overhead, and each sample takes up 50 bytes of physical memory; Figure 3 illustrates an example of the file structure with header and data. Through the employed compression algorithm we are able to achieve an average compression ratio (size of original/size of compressed file) of **6** in the file size.

```
# bioPlux Text File Format
# Version: 1
# StartDateTime: 2010-09-14 11:23:55
# SamplingFrequency: 1000
# SampledChannels: 1 2 3 4 5 6 7 8
# SamplingResolution: 12
# AcquiringDevice: BioPlux EMG  v.2
# EndOfHeader
0    0    0    2048   0    0    0    0    0    0
1    0    0    2048   0    0    0    0    0    0
2    0    2    2048   0    0    0    0    0    0
3    0    0    2048   0    0    0    0    0    0
4    0    1    2048   0    0    0    0    0    0
```

Fig. 3. ASCII file example.

For the mobile client, the user interface takes advantage of the advanced features provided by the Android platform. Figure 4 depicts several configuration screens and an oscilloscope-like screen for real-time signal visualization. The first is used to configure the biosignal acquisition parameters by selecting what is the set of channels, sampling rate, and resolution that should be used for signal acquisition; the second, for the remote server configuration; and the later, allows the monitored channel selection and besides the raw signal is able to provide instant on-screen measurements of the computed parameters for local feedback to the user, in this case the Heart Rate of the subject in beats per minute(BPM).



(a) Channel selection screen (b) Remote Server screen (c) Real-time visualization screens

Fig. 4. User interface examples.

5 Case Study

5.1 Scenario

For testing purposes, a case study was devised for remote ECG and Heart Rate monitoring based on the presented framework. As end-device we have used a bioPLUX research generic biosignal acquisition unit, together with an ecgPLUX ECG triode sensor. Table 1 describes the main specifications of this system. Attached to this unit is an ecgPLUX active ECG triode, and its specifications are listed in Table 2. The interface with the skin is done through pre-gelled, self-adhesive round AgCl triode electrodes.

Table 1. bioPLUX research specifications.

Connectivity	Bluetooth Class II
Sampling Rate	1000Hz
Channels	8 An. + 1 Dig.
Size	84x53x18mm
Weight	86g

Table 2. ecgPLUX sensor specifications.

Gain	1000
Filtering	0.05-30Hz
CMRR	110dB
Input Impedance	$>$1MOhm

5.2 ECG Signal Processing

A signal processing block was implemented to analyze the raw signal in real-time and automatically compute the corresponding Heart Rate (HR) value.

The algorithm used for this propose is based on the determination of the R peaks of the ECG signal. We followed one of the proposed approaches of [15], based on amplitude and first derivative of the signal.

The ECG signal, $x[n]$, is rectified, and passed through a low level clipper:

$$y_0[n] = |x[n]|, \tag{1}$$

$$y_1[n] = y_0[n] \text{ if } y_0[n] \geq A$$

$$y_1[n] = A \text{ if } y_0[n] < A \tag{2}$$

The amplitude threshold, A, is determined as a fraction of the maximum amplitude of the signal in the analyzed window:

$$A = 0.4max[x[n]], \tag{3}$$

Finally, the first derivative is calculated:

$$y_2[n] = y_1[n+1] - y_1[n-1], \tag{4}$$

A R peak candidate occurs when a point in $y_2[n]$ exceeds a fixed constant threshold.
The application was deployed in an HTC Hero handset enabling the continuous real-time monitoring of the ECG signal.

6 Conclusions

In an aging society, where life expectancy is increasing and birth rate is decreasing, there is a clear effort of addressing this problem with tools and technologies capable of helping in the self-management and delivery of improved healthcare services to people. The Ambient Assisted Living Joint Programme points telemedicine as a strategic application to address the needs of this growing market.

Other platforms have been proposed in the past that either use dedicated hardware or conventional mobile platforms that, by design, are limitations and high permissions restrictions that are preventing the faster evolution of telemedicine systems in terms of functionality and usability. In this paper we have presented a real-time biosignal acquisition and telemonitoring platform for AAL based on Android OS.

This paper presents an approach which takes advantage of the novel capabilities of that mobile platform and of the compliant handsets. We've provided a brief discussion on the current mobile platform landscape, and why Android OS is ahead of the competition in this field, we've described our proposed approach, and finally a case study is described where this framework was applied to the continuous real-time monitoring of ECG signals.

Preliminary results have shown the validity of this proposed approach for the intended application due to the always-on, always-connected philosophy of modern devices and to the capabilities introduced by Android OS as unified APIs, unrestricted hardware access, and unprecedented security management policy.

References

1. Steg, H., Strese, H., Loroff, C., Hull, J., Schimdt, S.: Europe is facing a demographic challenge ambient assisted living offers solutions. Technical report, VDI/VDE-IT (2006)
2. Yamauchi, K., Chen, W., Wei, D.: 3g mobile phone applications in telemedicine - a survey. In: Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on. (2005) 956 – 960
3. Wootton, R., Craig, J., Patterson, V.: Introduction to Telemedicine. Oxford University Press (2010)
4. Vodafone: Pulido valente hospital and vodafone foundation present telemonitoring of long term oxygen therapy (2009)
5. Vitacca, M., Comini, L., Tentorio, M., Assoni, G., Trainini, D., Fiorenza, D., Morini, R., Bruletti, G., Scalvini, S.: A pilot trial of telemedicine-assisted, integrated care for patients with advanced amyotrophic lateral sclerosis and their caregivers. Journal of Telemedicine and Telecare 16 (2010) 83–88

6. Wan, D., Mentzer, K., Nash, R.: Always on, always connected how technology will transform the future of chronic care. Technical report, Accenture (2009)
7. Zheng, P., Ni, L.: Smart Phone and Next Generation Mobile Computing. Morgan Kaufmann (2005)
8. Vergne, H.D.L., Milanesi, C., Cozza, R., Gupta, A., Lu, C., Nguyen, T., Zimmermann, A.: Competitive landscape: Mobile devices, worldwide, 2q10. Technical report, Gartner (2010)
9. Figueredo, M., Dias, J.: Mobile telemedicine system for home care and patient monitoring. In: Proceedings of the 26th Annual International Conference of the IEEE EMBS, IEEE EMBS (2006)
10. Wang, C.S., Lee, J.H., Chu, Y.T.: Mobile telemedicine application and technologies on gsm. In: Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on. (2007) 1125 –1128
11. Zhang, P., Kogure, Y., Matsuoka, H., Akutagawa, M., Kinouchi, Y., Zhang, Q.: A remote patient monitoring system using a java-enabled 3g mobile phone. In: Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE. (2007) 3713 –3716
12. Laakko, T., Leppanen, J., Lahteenmaki, J., Nummiaho, A.: Multipurpose mobile platform for telemedicine applications. In: Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on. (2008) 245 –248
13. Finlay, D., Nugent, C., Breen, C., Bond, R., Moore, G.: A smartphone based telemedicine system for recording limited lead body surface potential maps. In: Computers in Cardiology, 2009. (2009) 237 –240
14. Salomon, D.: Data Compression: The Complete Reference. Springer-Verlag (1998)
15. Friesen, G., Jannett, T., Yates, M.J.S., Quint, S., Nagle, H.: A comparison of the noise sensitivity of nine QRS detection algorithms. IEEE Transactions on Biomedical Engineering 37 (1990) 85 –98