

# iScope – Viewing Biosignals on Mobile Devices

Christian Breitwieser<sup>1</sup>, Oliver Terbu<sup>1</sup>, Andreas Holzinger<sup>2</sup>, Clemens Brunner<sup>1</sup>,  
Stefanie Lindstaedt<sup>3</sup>, and Gernot R. Müller-Putz<sup>1</sup>

<sup>1</sup> Institute for Knowledge Discovery, Graz University of Technology, Inffeldgasse  
13/IV, 8010 Graz, Austria

`{c.breitwieser,clemens.brunner,gernot.mueller}@tugraz.at,`  
`o.terbu@gmail.com`

<sup>2</sup> Institute for Medical Informatics, Statistics and Documentation,  
Medical University Graz, Auenbruggerplatz 2/V, 8036 Graz, Austria  
`andreas.holzinger@meduni-graz.at`

<sup>3</sup> Know Center GmbH, Inffeldgasse 13/VI, 8010 Graz, Austria  
`slind@know-center.at`

**Abstract.** We developed an iOS based application called iScope to monitor biosignals online. iScope is able to receive different signal types via a wireless network connection and is able to present them in the time or the frequency domain. Thus it is possible to inspect recorded data immediately during the recording process and detect potential artifacts early without the need to carry around heavy equipment like laptops or complete PC workstations. The iScope app has been tested during various measurements on the iPhone 3GS as well as on the iPad 1 and is fully functional.

**Keywords:** biosignal, ECG, EEG, iOS, monitoring, application, visualization.

## 1 Introduction

Recording, processing, storing, and analyzing biosignals is a major part within the broad field of neuroscience or health informatics. Common biosignals in this case include the electroencephalogram (EEG), the electrocardiogram (ECG), and many others. Such biosignals are often either recorded, stored and processed later on or directly processed online during the recording process. In both cases, a clean and artifact free signal is a crucial requirement; a recording heavily contaminated with biological and/or technical artifacts could render the recorded data useless. Thus, online monitoring of the recorded signals is an important aspect during biosignal recording processes.

Many companies already provide tools to monitor such signals like the Brainvision recorder from BrainProducts (Brain Products, Gilching, Germany) or their open-source acquisition program PyCorder. Other open-source tools [1] include the BCI2000 Source Signal Viewer [2], the OpenViBE Signal Display [3] or the tools4BCI TiAScope<sup>1</sup>. All these viewers provide the user with the possibility to

<sup>1</sup> `tools4bci.sf.net`

monitor the acquired signals. However, they are partly bound to hardware from one manufacturer or restricted to single operating systems, and they all require a running personal computer to display the recorded signals. Often, in case of patient measurements and other measurements out of the lab, a full-blown measurement system can be a high burden. This might be due to limited space, safety regulations or just the inconvenience to carry around an array of laptops.

Furthermore, as presented in [5,6], flexibility and standardization are important, in the field of brain-computer interface (BCI) research [7]. BCIs, as described with a functional model by Mason et al. [8], rely on realtime processing of brain signals, or in the case of hybrid BCI systems [9], also on processing of other types of input signals like buttons, joysticks, and many more. Reliable processing and an early detection of artifacts or other disturbances in the recorded data are a crucial requirement to provide accurate feedback to the user.

To cope with these issues, we present a lightweight and portable app running on iOS devices (iPhone, iPad, . . .) to monitor recorded biosignals online, using the tools4BCI TiA (TOBI interface A) protocol [4]. Figure 1 shows a schematic illustration of the data flow and at which point the iScope is used to display the acquired signals. With this application, it becomes possible to monitor the biological signals in realtime and immediately determine the signal quality, hence avoiding recording useless data due to bad signal quality.

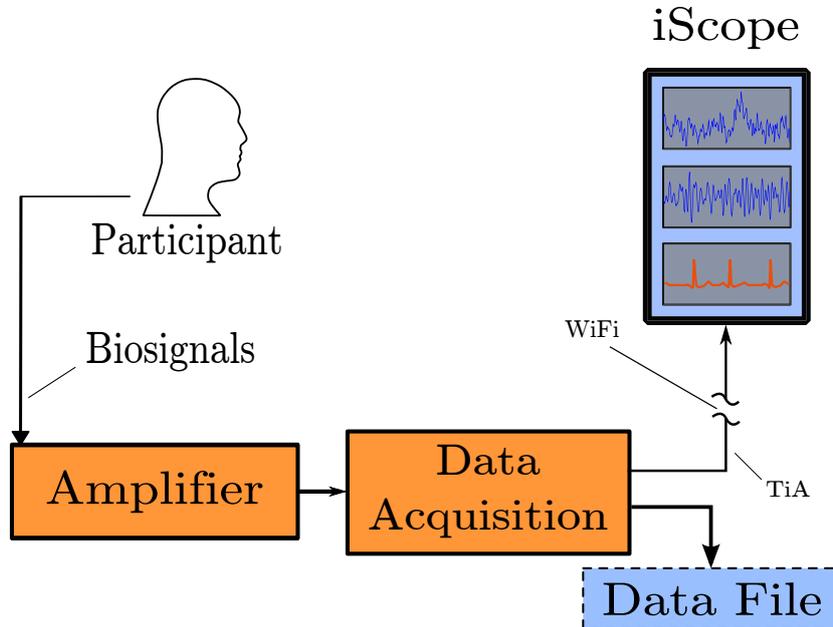
## 2 Methods

### 2.1 Requirement Analysis

As presented in [4], biosignal acquisition can be subdivided into different signal types such as EEG, ECG, and many more. Such a distinction is a meaningful way to transmit and display signals, as different signal types like EEG or ECG have different characteristics (amplitude, vulnerability to interferences, . . .). Thus, individual treatment for different signal types is further also a meaningful way to design the iScope. Due to a limited resolution and display size of iPods, iPads and others, the selection of individual channels is an important feature too, as it might become a problem to draw all signals at once [10]. Additionally, a user should be free to choose the signals to display.

Signals can be analyzed in the time as well as in the frequency domain. Often, the observation of a signal in the frequency domain reveals information, hardly visible in the time domain. As the fast fourier transform (FFT) is a powerful and relatively low resource consuming way to transfer a time domain signal into the frequency domain, this feature is a valuable requirement too.

Usually it is desired to inspect a certain time window of recorded data, e.g., the last 10 seconds, but this requirement can change during a recording out of various reasons. Thus, a time-zoom function is a crucial requirement. With this function it is possible to steplessly change the timeframe of the displayed signals. Additionally, the possibility to zoom exactly one channel to full screen is implemented, as a detailed inspection of single channels is a necessary feature too.



**Fig. 1.** Schematic illustration of the data flow to the iScope. Biosignals are recorded from a participant by an amplifier and acquired by a data acquisition module. This module then delivers the recorded data via the TiA protocol to the iScope, where it can be monitored in realtime in different ways.

## 2.2 Design Principles

iOS devices are usually designed to fulfill low power consumption requirements, which are unfortunately usually related to limited processing power of the respective device. Thus, especially in case of data acquisition with many channels or high sampling rates, an iOS device might be too slow to display all channels at once. Furthermore, the resolution is also restricted in some way. A meaningful way to cope with those two problems is the introduction of a fixed amount of graphs visible on the display and the possibility to scroll through all signals. Hence, only a fixed number of signals has to be displayed at once, resulting in a scalable behavior based on the number of displayed channels and their sampling rates.

To avoid freezing of the device in case of insufficient processing power, the only supported protocol is the user datagram protocol (UDP), which is also offered by TiA. In case of a full network buffer, samples are automatically dropped, reducing the amount of samples to process. In case of any sample loss, the user will be notified with red lines in the individual graphs, padding the lost samples.

To fulfill the usual look-and-feel of the iOS environment, the whole control of iOS is realized with finger gestures. Moving one finger up or down on the display is used to scroll through the different graphs, a two finger pinch gesture is used

to in- or decrease the displayed timeframe and double taps are used to zoom the tapped channel to full screen mode.

## 3 Results

### 3.1 The iScope App

Figures 2 and 3 show different user screens of the iScope app, either presenting acquired signals in the time- or the frequency domain.

It can be seen, that the channel selection is functional and also the distinction of different signal types is properly supported. Furthermore, this selection can be modified during the running app. iScope is able to display multiple graphs at once, also from different signal types, as visible in Figure 2(b), where EEG signals are shown together with an ECG signal. Figure 2(c) illustrates the zoomed ECG signal for a timeframe of 1.7 s with additional information.

Figure 3(a) shows FFT spectra of multiple channels, which are computed and plotted in realtime. The zoomed FFT view mode is presented in Figure 3(b), which provides additional information compared to the multi graph view. In this case, the spectrogram is computed for a time period of one second and updated every 0.2 s. This parameters can be changed in the iScope configuration to fit the individual users needs.

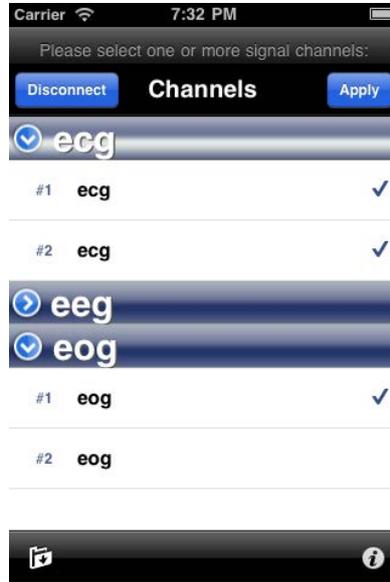
A mobile application is not just the pure 1:1 transformation of a possible desktop solution, instead a complete adaptation on the mobile context of use, as it is essential in the medical area to consider the aspects of mobility and to bring a clear added value for the end user [11]. One possible solution is to adapt to the individual context automatically, so to say making the App smart [12].

iScope further supports various other little features, as re-grouping of graphs, automatic or manual scaling, the possibility to take screenshots, save them to the device or send them per email and other little things.

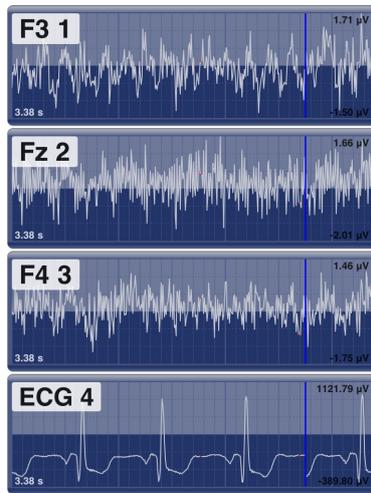
## 4 Discussion

iScope was successfully tested and evaluated on different iOS devices. It has already been used in various measurements within and outside the lab and has proved its usefulness. It is easy and intentional to use, as common iOS apps. It further proved it's stability during measurements with a duration longer than two hours.

The current iScope implementation is based on iOS 4, but porting to newer iOS versions should become no problem and will be a straight forward process [13]. Furthermore, it has mainly been tested on different iOS devices like the iPhone 3GS or the iPad 1. Considering this two devices, a main difference in computational power could be recognized. While the iPhone 3GS was merely able to process and display 3 channels (while receiving 10 channels) at once without juddering, the border of the iPad 1 was much higher. Due to the higher



(a) Selection Screen



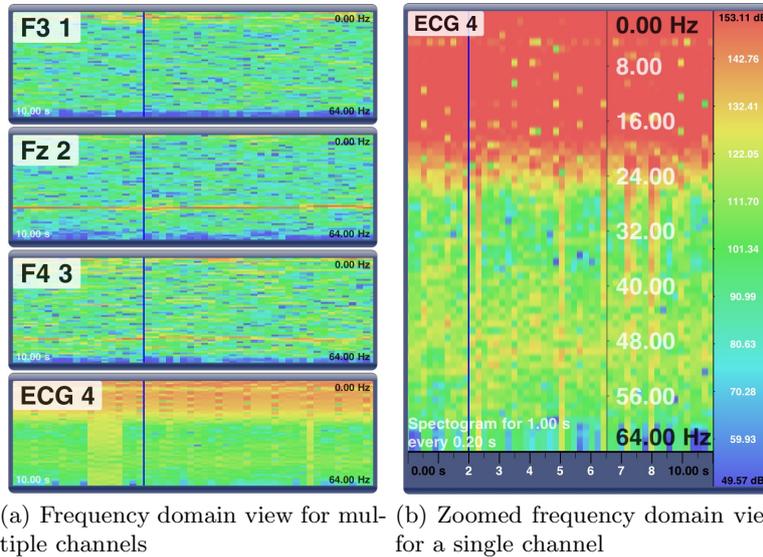
(b) Visualization Screen



(c) Detailed Visualization

**Fig. 2.** User screens within the iScope app. The first screen shows the selection screen, where individual channels or entire signal types can be switched on or off. The second screen shows the usual visualization screen in the time domain with four signal graphs. Screen number three presents the zoomed mode of the fourth signal graph.

computational power of the iPad 1, it was smoothly possible to show 4 channels and calculate the FFT view at the same time, while receiving 32 channel with 128 Hz sampling rate without any sample loss.



**Fig. 3.** Multi graph and zoomed FFT view within the iScope app. Red colors illustrate a higher power and blue colors a lower power. Concrete values are visible in the zoomed presentation mode.

Due to the rapid development of such mobile devices, higher sampling rates and a higher number of channels will become possible because of the increase of computational power. Better display resolutions will further facilitate the presentation of more signal graphs at once.

However, distribution of the iScope is not a trivial process. A main component of the iScope, the TiA library, is licensed under the lesser GNU public license (LGPL), which creates some issues, uploading the iScope to Apple’s app store. Thus, the iScope is currently only available on request, as it has to be built for individual devices manually. However, after resolving the licensing issues, caused by the TiA lib, the iScope app is planned to be published in the app store, to distribute this useful tool to a broader community.

**Acknowledgements.** This work is supported by the European ICT Programme Project FP7-224631. This paper only reflects the authors’ views and funding agencies are not liable for any use that may be made of the information contained herein. The authors are further grateful to the KnowCenter for financial and technical support during the development of the iScope app.

## References

1. Brunner, C., Andreoni, G., Bianchi, L., Blankertz, B., Breitwieser, C., Kanoh, S., Kothe, C.A., Lécuyer, A., Makeig, S., Mellinger, J., Perego, P., Renard, Y., Schalk, G., Susila, I.P., Venthur, B., Müller-Putz, G.R.: BCI Software Platforms. In: Allison, B., Dunne, S., Leeb, R., Del, R., Millán, J., Nijholt, A. (eds.) *Towards Practical Brain-Computer Interfaces: Bridging the Gap from Research to Real-World Applications* (2012) (in press)
2. Schalk, G., McFarland, D.J., Hinterberger, T., Birbaumer, N., Wolpaw, J.R.: BCI 2000: a general-purpose brain-computer interface (BCI) system. *IEEE. Trans. Biomed. Eng.* 51(6), 1034–1043 (2004)
3. Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., Lécuyer, A.: OpenViBE: An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *Presence-Teleop. Virt.* 19(1), 35–53 (2010)
4. Breitwieser, C., Daly, I., Neuper, C., Müller-Putz, G.R.: Proposing a Standardized Protocol for Raw Biosignal Transmission. *IEEE. Trans. Biomed. Eng.* 59(3), 852–859 (2012)
5. Müller-Putz, G.R., Breitwieser, C., Cincotti, F., Leeb, R., Schreuder, M., Leotta, F., Tavella, M., Bianchi, L., Kreiling, A., Ramsay, A., Rohm, M., Sagebaum, M., Tonin, L., Neuper, C., Millán, J.d.R.: Tools for Brain-Computer Interaction: A General Concept for a Hybrid BCI. *Front Neuroinform* 5(30), 1–10 (2011)
6. Brunner, P., Bianchi, L., Guger, C., Cincotti, F., Schalk, G.: Current trends in hardware and software for brain-computer interfaces (BCIs). *J. Neural. Eng.* 8(2), 1–7 (2011)
7. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* 113(6), 767–791 (2002)
8. Mason, S.G., Birch, G.E.: A general framework for brain-computer interface design. *IEEE Trans. Neural. Syst. Rehabil. Eng.* 11(1), 70–85 (2003)
9. Pfurtscheller, G., Allison, B.Z., Brunner, C., Bauernfeind, G., Solis-Escalante, T., Scherer, R., Zander, T.O., Müller-Putz, G., Neuper, C., Birbaumer, N.: The hybrid BCI. *Front Neurosci.* 4(30), 1–11 (2010)
10. Holzinger, A., Treitler, P., Slany, W.: Making Apps Useable on Multiple Different Mobile Platforms: On Interoperability for Business Application Development on Smartphones. In: Quirchmayr, G., Basl, J., You, I., Xu, L., Weippl, E. (eds.) *CD-ARES 2012. LNCS*, vol. 7465, pp. 176–189. Springer, Heidelberg (2012)
11. Holzinger, A., Kosec, P., Schwantzer, G., Debevc, M., Frühwirth, J., Hofmann-Wellenhof, R.: Design and Development of a Mobile Computer Application to Reengineer Workflows in the Hospital and the Methodology to Evaluate its Effectiveness. *J. Biomed. Inform.* 44(6), 563–570 (2011)
12. Holzinger, A., Geier, M., Germanakos, P.: On the development of smart adaptive user interfaces for mobile e-Business applications: Towards enhancing User Experience some lessons learned, pp. 3–16. SciTePress, INSTICC, Setubal (2012)
13. Holzinger, A., Errath, M.: Mobile computer Web-application design in medicine: some research based guidelines. *Universal Access in the Information Society International Journal* 6, 31–41 (2007)