# BiosignalsStudio: A Flexible Framework for Biosignal Capturing and Processing

Dominic Heger, Felix Putze, Christoph Amma, Michael Wand, Igor Plotkin, Thomas Wielatt, and Tanja Schultz

Cognitive Systems Lab (CSL)
Karlsruhe Institute of Technology (KIT), Germany
{dominic.heger,felix.putze,christoph.amma,
michael.wand,tanja.schultz}@kit.edu,
{igor.plotkin,thomas.wielatt}@student.kit.edu

**Abstract.** In this paper we introduce BiosignalsStudio (BSS), a framework for multimodal sensor data acquisition. Due to its flexible architecture it can be used for large scale multimodal data collections as well as a multimodal input layer for intelligent systems. The paper describes the software framework and its contributions to our research work and systems.

## 1 Introduction

In modern statistically based AI research nearly all systems are fundamentally dependent on sensory input data. They infer task-relevant patterns from the outside world and process them by recognition algorithms, which often require large amounts of training data to statistically model relevant information.

From a research perspective, data acquisition is a critical and non-trivial task, particularly when human test persons are involved. We see the need for a tool that supports this task and integrates seamlessly into the development workflow. This recording tool should be flexible, extensible, usable for non-experts and portable to different plattforms.

In this paper we introduce BiosignalsStudio (BSS), a flexible framework for multimodal sensor data acquisition, which is designed as a component of human-machine interaction (HMI) systems for both the development phase with large-scale data collections and the deployment phase for online recording. BSS is available for research purposes by contacting the authors.

## 2 Motivation

Designing and implementing a multimodal HMI system typically involves the recording of data for two kinds of purposes. First, a large data corpus has to be collected to train statistical models of, for example, user states or actions. Second, once the system is working, data has to be recorded and passed to a classification system with minimal time delay. The setup for the two scenarios

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 33–39, 2010.

may differ in recording parameters, like sensors, the underlying hardware, operating systems, and the requirements to the recording software. In the collection stage, data is archived after recording, while for the deployed system, data has to be passed on to the classifier in real-time. To avoid usage of different tools for the same task, we developed the BSS framework, which can be used in both scenarios. In the development phase BSS can be used to collect multimodal data corpora for analysis and system training and in the deployment phase it provides real-time sensor data as a multimodal input layer for the recognition system.

At the Cognitive Systems Lab, BSS is in use for several large-scale data collections, each with a different focus. All of them have in common that they require the recording software to capture the biosignals and all necessary meta information in a reproducible and coherent way. BSS offers a convenient and streamlined format to store all signals, corresponding timestamps, comments and the configuration of the recordings for each experiment.

For the integration into end-to-end systems, the requirements on the recording software are different. We need stable real-time recording of multiple parallel streams and the possibility to pass the recorded data to a classification system on the same or a different machine. As the recording software may come to use in very different settings, it should be portable, light-weight and integrate seamlessly into new environments.

BSS offers all of those features. By using the same software for data collection and recording for runtime classification, we can use the same tool-chain in both cases. It also provides visualization and data storing facilities at hand.

## 3 Related Work

There is a huge variety of commercial data-recording tools available. However, to our best knowledge, no tool offers all of the advantages of the BSS.

The first group of software products are commercial systems bundled with recording devices, distributed by hardware manufacturers. They usually have proprietary interfaces and cannot be used in combination with other manufacturers devices or custom developed systems. This causes problems when changing sensor setups (e.g. for testing in different environments) or using sensors of different origin in parallel. Depending on the software's transparency, it is difficult to implement online processing of the recorded data or additional stimulus presentation. Examples for this kind of software are gRecorder [1] or LabChart [2].

The second group of software products are more general frameworks which allow recording from arbitrary devices. The most prominent representative of this group is certainly LabView [3]. With a graphical programming interface, LabView allows the design of complex arrangements of building blocks for data acquisition and processing to form a complete and potentially parallel data flow graph. Many sensor manufacturers provide LabView modules which can be integrated as data sources. While LabView definitely is a very powerful tool, there are major differences to the BSS which make the latter more suited for the application in human-computer interaction: While LabView requires a large framework to

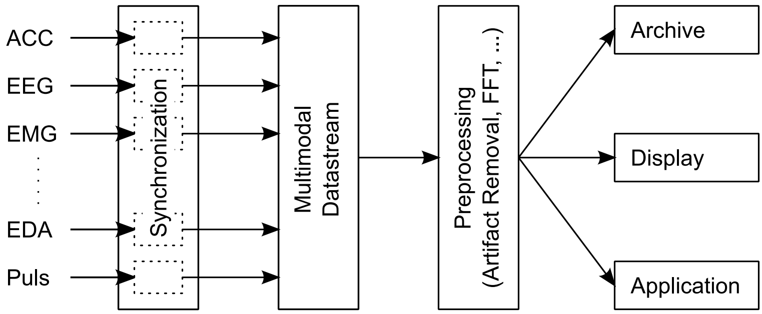


**Fig. 1.** Overview of the data flow oriented BSS Architecture. BSS can take various modalities in parallel as input, a selection of implemented input modalities is depicted on the left-hand side. Synchronization is performed before merging data streams to ensure minimal synchronization errors. Different output modalities can be selected.

be installed, the BSS only requires a Python interpreter and a few packages and is therefore highly portable. As it is implemented in an open, popular general-purpose programming language, the addition of existing or new modules for signal recording and processing is very simple in the BSS. In [4], an interesting framework called OmniRoute for recording different sensor modalities in the context of affective computing is proposed. However, the development of OmniRoute was discontinued and the framework is not available to the public.

Biofeedback software, such as BioEra [5] or BrainBay [6] has elaborated visualization features, has real-time capabilities and supports different recording hardware. However, these systems are neither designed for large-scale data collection, nor do they have the flexibility to get easily integrated into research applications.

## 4 BiosignalsStudio

### 4.1 Architecture

**Overview:** The BSS framework was designed to be extensible, multimodal, flexible, portable, and to allow short development cycles for application developers. A sophisticated modularization is essential for the system. A simple application based on the framework merely requires the configuration and definition of data connections between existing modules. Figure 1 gives an overview of the architecture.

A module is a component designed to cover one specific task. The fine-grained separation of tasks ensures the greatest possible flexibility of the framework. For example, the tasks of recording audio data and saving it in a specific format are carried out by two different modules. This allows for a flexible migration to another audio data format through replacement of the data saving module. Modules with similar tasks are interchangeable without further modifications due to standardized module interfaces.

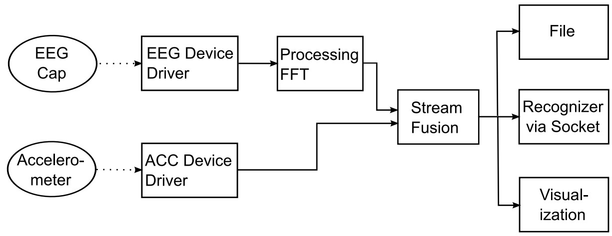


**Fig. 2.** Example module configuration for a recording setup with an EEG cap and an accelerometer as input modalities. Sensor data is fed into the framework via input modules (device drivers) and processed by a FFT and a fusion module. Data is then written to a file, send to another application and visualized on screen.

Multiple modules have to be connected to build a functional unit. Modules are connected through streams which represent the data flow in a specific application. All modules implement input and/or output streams for exchanging data with other modules. The input/output interface can be connected with any number of data senders or receivers. This allows the distribution of the same data across multiple modules, e.g. for saving and graphically representing the recorded data. Generic standard formats for the data exchange ensure reusability and interoperability of the modules.

The design goal of multimodality is realized by the option to instantiate and to start any number of modules in parallel, which allows recording, synchronization and analysis of multiple data sources at the same time. Despite some Python limitations concerning multi-core development, the system can execute performance-critical modules as sub-processes to use multiple cores. Furthermore, Python is designed to easily cooperate with other programming languages (in particular, C++) so that these languages also can be used to extend BSS.

**Streams:** The Framework has a data flow oriented architecture. All data is organized in data streams. A stream has a source and a target, the source is typically a device driver for a sensor and a target a file or socket. Data streams consist of modules through which the data is sequentially passed through. The framework offers abstract base classes, that implement the standardized interface to read data from streams and write data to streams. A stream can be split into different individual streams or different streams can be merged together. Figure 2 shows a typical example application scenario. All classes derived from the module base class can be easily plugged together to construct new data streams, thereby ensuring reusability of written modules.

Within a stream, data itself is organized in channels, typically one channel corresponds to a one-dimensional sensory input device. Besides the data, a channel can contain arbitrary additional meta information, for instance, time stamps needed for synchronization can be saved in the meta information.

**Synchronization:** A key issue in multimodal data acquisition is the synchronization of the different data sources to be able to correlate the modalities with

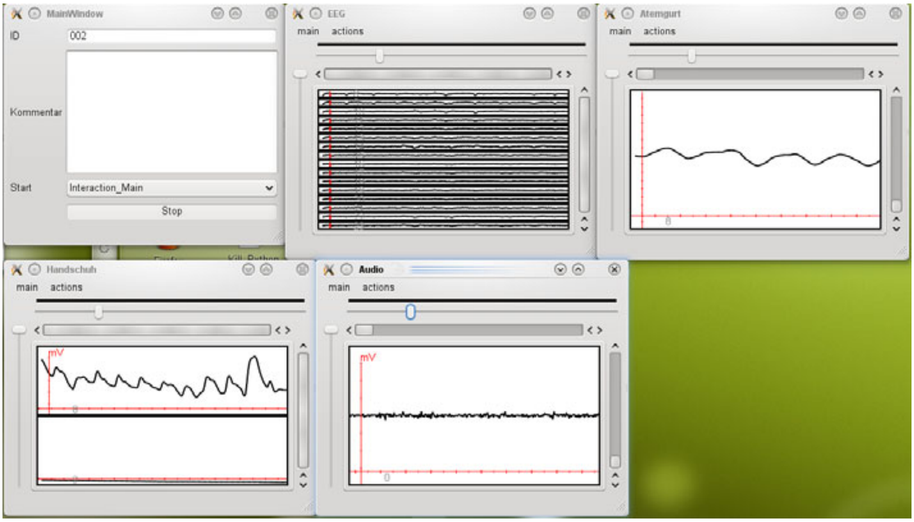


**Fig. 3.** Screenshot of the visualization module. A live visualization of sensor data from the following devices is shown: a 16 channel EEG (top row, middle), a respiration belt (top row, right), a wireless sensor glove measuring pulse (second row, left), a microphone (second row, right). All windows are resizable and scalable for signal inspection.

each other. Our framework offers a simple but effective approach to deal with this problem. Whenever a device driver module reads new input data, a timestamp depending on the system time is attached to the meta information of the data. The threaded device drivers ensure that new data is read with a minimum delay and thereby the timestamp is as close as possible to the actual time the data was acquired by the sensor.

**Visualization:** The BSS framework also includes modules to visualize acquired signals in real-time through OpenGL. The visualization of different signals can be freely arranged in individual windows and layouts. Figure 3 shows an example with different signal sources. The visualized signal can be mapped by an arbitray function (e.g. scaling to correct constant offsets, logarithmic axes, etc.).

### 4.2 Available Modules

The BSS is packaged with a variety of modules which can be used to build new applications. There are different types of modules available with different roles within a project:

**Input Modules:** Input modules insert new data into the stream. The data can for example be acquired by reading hardware devices or by artifical generation. The current BSS comes with input modules for accessing the biosignal recording interface by Becker Meditec (including EEG, EMG, Respiration, EDA, Plethysmography), by recording interfaces from generic audio devices and video cameras, modules for recording from different input devices (keyboard, steering wheel, etc.), and signal generators for testing. New input modules for different devices can be added easily, in particular when using serial connections (e.g. USB, Bluetooth). It is sufficient to develop a new device driver input module that implements the functionality needed to communicate with the device.

**Processing Modules:** The BSS is not limited to simple input-output stream topologies. There is a selection of intermediate processing modules that help to manipulate the incoming data before it is sent to other components. Typical processing modules that are available in the BSS are converters between different data formats, filters and mathematical operations, especially for enhanced visualization (Butterworth filters, Fast Fourier Transform, etc.), and internal channel manipulators (e.g. channel merger, temporal storage, etc.).

**Output Modules:** Data recording with the BSS is typically only the first step in a more complex data processing chain. The BSS therefore offers a variety of output channels. This includes writing to files (with timestamps), online visualization for signal inspection, and generic socket connections to access other components of a human-machine interface, for example statistical classifiers.

**Interactive Modules:** Interactive components are part of recording setups in which stimuli are presented to the user or in which user input is required. They can be started in parallel to the recording modules, in fact most interactive modules also act as input modules for logging their events. Interactive modules can be of any type, the BSS comes with a set of cognitive tests (e.g. Flanker, Oddball, Stroop, etc.) and modules for multimodal stimulus presentation.

## 5 Applications

The BiosignalsStudio is already applied in a variety of diverse applications for the investigation of human-machine interaction and human cognition. This includes both, large-scale data collection and online classification tasks. The following section gives an overview over the most interesting ones, highlighting the different requirements for the BSS framework.

**HMI in Dynamic Environments:** In a driving simulator, we are working on cognitive interaction systems in the car that are able to adapt to inner states of the driver [7], [8]. To detect those inner states, a variety of different biosignals is continuously recorded, including video, voice, several physiological signals, and steering angle. With these many input streams, synchronous recording of parallel streams is critical. As with many input devices comes an increased chance of hardware failure, the BSS has some abilities to recover from situations of lost connections or missing data.

**Cognitive Fitness:** To investigate the effect of physical activity on cognitive fitness, the BSS contains a whole battery of cognitive tests which can be easily selected and started from a convenient graphical user interface. This is especially useful for larger data collections where student researchers with limited training are employed to start and supervise the recordings. To facilitate later segmentation and labeling of the data, logging of events during a test (e.g. button pressing, stimulus presentation) is completely integrated into BSS.

**Brain Computer Interfaces:** For research on the recognition of imagined body movements and unspoken speech using EEG streams, we employ BSS to collect data of various test persons to investigate interpersonal differences. BSS has the ability to present a variety of stimuli to the test person (written words, sounds, videos) which are synchronized into the biosignal stream for labeling.

**Online Workload Classification:** For the development of an empathic humanoid robot that is able to detect and react to its user's mental workload, we developed an online workload recognizer based on EEG data of which the BSS is an integral part [9]. The recognizer works in real-time and receives the data stream from the BSS module via a socket connection.

**Airwriting Recognition:** For the online recognition of letters and whole words written in the air [10], arm movement is captured by recording acceleration sensor and gyroscope data generated by a wireless sensor glove. As the glove was self-designed, we developed and integrated a new BSS module to capture the data via a bluetooth interface. Python is available for various plattforms, including smartphones, so we currently strive for the integration of the system, including BSS, on a mobile device.

## References


[1] Guger Technologies OG, Herbersteinstrasse 60, 8020 Graz, Austria: g.Recorder Biosignal Recording Software, `http://www.gtec.at`

[2] ADInstruments Pty Ltd, Unit 13, 22 Lexington Drive Bella Vista, NSW, Australia: LabChart (2153), `http://www.adinstruments.com`

[3] National Instruments Corporation, 11500 N Mopac Expwy, Austin, TX, USA: LabView, `http://www.ni.com/labview`

[4] Mader, S., Peter, C., Göcke, R., Schultz, R., Voskamp, J., Urban, B.: A Freely Configurable, Multi-modal Sensor Sytem for Affective Computing. In: Proceedings of the Affective Dialogue Systems Workshop, Kloster Irrsee, Germany (2004)

[5] PROATECH LLC: BioEra, `http://www.bioera.net`

[6] Veigl, C.: An Open-Source System for Biosignal- and Camera-Mouse Applications submission for the Young Researchers Consortium of the ICCHP 2006 (2006), `http://www.shifz.org/brainbay`

[7] Putze, F., Schultz, T.: Cognitive Dialog Systems for Dynamic Environments: Progress and Challenges. In: Proceedings of the 4th Biennial Workshop on DSP for In-Vehicle Systems and Safety (2009)

[8] Putze, F., Jarvis, J.-P., Schultz, T.: Multimodal Recognition of Cognitive Workload for Multitasking in the Car. In: Accepted for Publication in the Proceedings of the 20th International Conference on Pattern Recognition (2010)

[9] Heger, D., Putze, F., Schultz, T.: Online Workload Recognition during Cogitive Tests and Human-Computer Interaction. In: Dillmann, R., et al. (eds.) KI 2010. LNCS (LNAI), vol. 6359, pp. 402–409. Springer, Heidelberg (2010)

[10] Amma, C., Gehrig, D., Schultz, T.: Airwriting Recognition using Wearable Motion Sensors. In: Proceedings of the First Augmented Human International Conference (2010)