# Using Synchronously Registered Biosignals Dataset for Teaching Basics of Medical Data Analysis – Case Study

Tomasz Kocejko[1*]

[1] Department of Biomedical Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology (11/12 Gabriela Narutowicza Street, Gdańsk, Poland)

* Correspondence author: tomkocej@pg.edu.pl; ORCID: 0000-0002-2304-8942

## Abstract

Medical data analysis and processing strongly relies on the data quality itself. The correct data registration allows many unnecessary steps in data processing to be avoided. Moreover, it takes a certain amount of experience to acquire data that can produce replicable results. Because consistency is crucial in the teaching process, students have access to pre-recorded real data without the necessity of using additional equipment for data acquisition. The analyzed sample dataset consists of raw signals of ECG, Body Impedance and Body temperature recorded synchronously in laboratory conditions. The data are sampled with 250Hz sampling frequency and are framed in blocks. Students gain a chance to acquire, exchange and process the medical data in simulated conditions. Pre-recording data provides the opportunity to teach certain techniques that can be used in real life scenarios but in a control replicable environment.

**Keywords:** biosignals processing; ECG, bioimedance; synchronous measurements; education

https://doi.org/10.34808/x55q-sz53_dyr_roz7

## Background

This case study shows how pre-recorded data can be used in the course of teaching basic algorithms of biosignal processing. The data were used in a course of a medical data processing and analysis in its laboratory part. The purpose of the laboratory exercises is to prepare students for a different tasks and problems which they may encounter on the job. In the case of biomedical signal processing, these tasks can by roughly divided into

two categories: signal measurements and signal acquisition and processing. Biomedical engineers with a specialty in medical informatics should focus on acquisition and processing the most. The biomedical measurement part should be covered by medical professionals. Therefore, the laboratory exercises were built on top of the data pre-recorded in a controlled environment, maintaining focus on the precision of the measurements and reliability of the data.

## The Data

The data are a complementary part of the experiment designed to demonstrate how to use network protocols to transmit, receive and process medical data. The dataset contains biomedical signals of ECG, impedance and temperature acquired simultaneously at a 250Hz sampling rate.

## Scope of the laboratory

The whole laboratory course is divided into four stages: Data Transmission, Data Acquisition, Data Visualisation and Data Processing. After completion of this four stages, participants can take part in a challenge relying on solving a real life problem related to the biomedical signals (like detecting the heart rate based on the ECG signal). In every stage, students become familiar with code and techniques necessary to write the software in C++. Although there are more suitable programming languages for signal processing (such as Octave/Matlab or Python), the goal of this course is to prepare students to write an end-to-end application for biosignal acquisition and processing that can be deployed on embedded devices. Therefore the recommended programming language is C++ with the QT framework ([https://www.qt.io/).

## Data Transmission and Acquisition

During the class, the data (ECG, impedance and temperature) are broadcast using the UDP protocol. This simulates obtaining the data directly from the measuring devices over popular protocols like TCP/IP, UDP (Zubairi, Misbahuddin and Tasadduq, 2010) (Lamberti and Sanna, 2005), bluetooth (Mulyadi et. al., 2009) or ZigBee (Jung and Lee, 2008). Most often, such devices send data bit by bit or as information containing more than one character. The students are obliged to become familiar with the technology, protocols and algorithms used for: both data transmission and data acquisition. They also have to create software for sending and receiving the data.

The essential parts of programming are demonstrated on diagrams and followed by exemplary code like is presented in Fig. 7.1.


**Fig. 7.1.** Exemplary code for communication using UDP protocol (QT C++)

The broadcast information is wrapped in a certain data frame:
<Impedance, ECG, Temp>
where:

"<" is the beginning of the data frame,
">" is the end of the data frame
" , " is a separator
"ECG" is the ECG signal amplitude
"Impedance" is the impedance signal amplitude
"Temp" is the amplitude of the temperature

In binary form, a single batch of data is 7 bytes (56 bits). The first 24 bits are impedance samples, the next 24 bits are ECG, the last 8 bits are samples acquired by means of a thermistor.

## Data Visualisation and Processing

Knowing the data frame, students have to write software to collect the data belonging to a certain biosignal and display them in real time. They also have to create a buffer for each signal.

This task requires writing a software program that will record data into a buffer of a given length. Students are presented with a simple example, like the one presented in Fig.7.2, of how to continuously display a sin wave transmitted from a function generator. On top of that software, they have to write their own solution for biosignal visualisation.

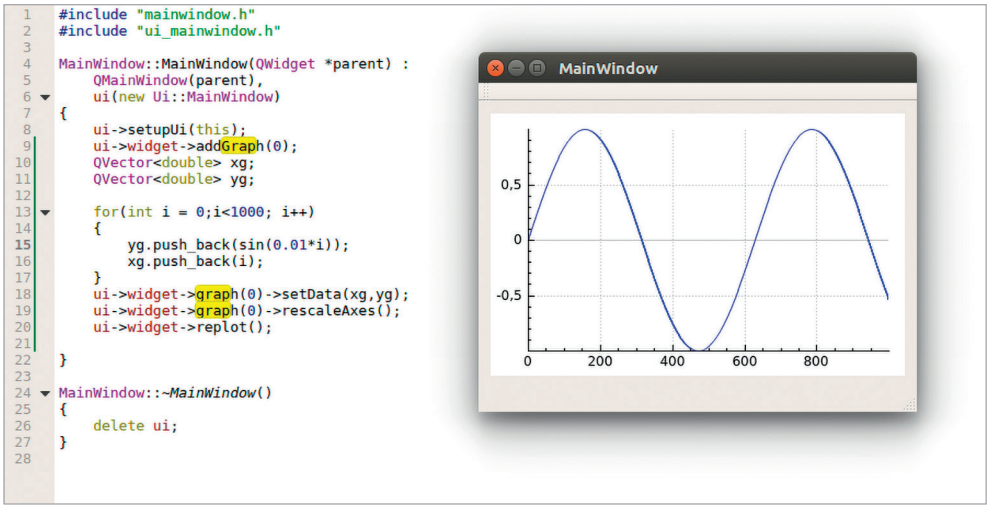


**Fig. 7.2.** Example of data visualisation using a simple buffer

Assuming that at this point students are familiar with the data structure, with the data frame and know how to split data and collect them using buffers, it is time for signal processing. This sometimes requires additional libraries for signal processing. In the described case study, students learn how to use the Armadillo C++ library for linear algebra and scientific computing [6]. Using additional libraries often requires changing the data format for the one suitable for signal processing. In this case, it is also desirable to perform signal filtration in the frequency domain. Students are provided with exemplary software for Fast Fourier Transform – FFT (Fig. 7.3) and precise comments on how to modify it for use with the available dataset.

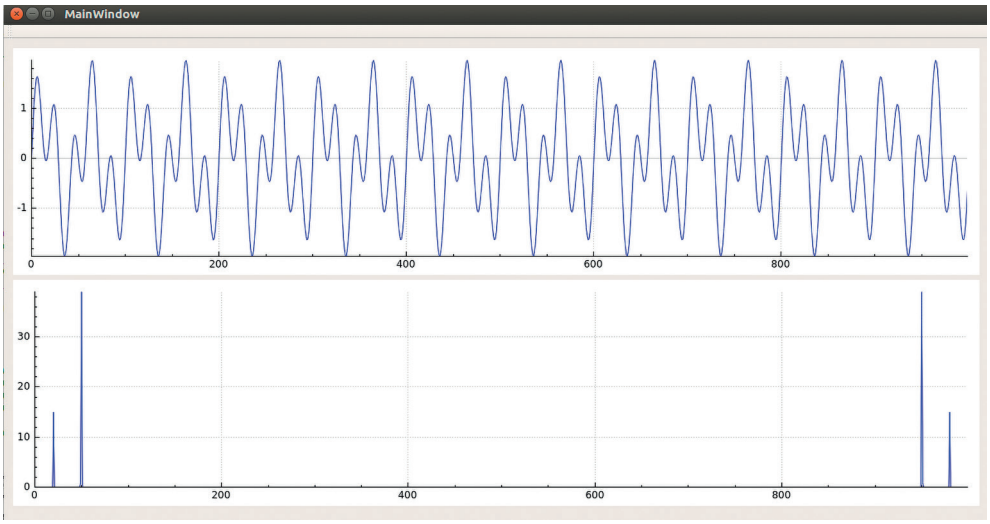


**Fig. 7.3.** Example of signal processing

## Conclusions

Dealing with biosignals is difficult as they are prone to all sorts of noise and distortions. Using the data pre-recorded in a control environment gives students a chance to focus on the algorithms. In the described case, data were broadcast to create the illusion that they came directly from measuring equipment (such as an ECG). The teaching process is more consistent due to the fact that all examples refer to the same case and to the same data. It also helps to compare results between students and different groups.

**Data quality and availability**

**Dataset DOI**:

10.34808/x3f9-fh19

**Dataset License**

CC-BY-NC

## References

[https://www.qt.io/] (Accessed: February 2021)

Jung, J.Y., and Lee, J.W. (2008) 'ZigBee device access control and reliable data transmission in ZigBee based health monitoring system'. In 2008 *10th International Conference on Advanced Communication Technology*, pp. 795–797, DOI: 10.1109/ICACT.2008.4493875.

Lamberti, F., and Sanna, A. (2005) 'A solution for displaying medical data models on mobile devices', 4th WSEAS International Conference on Software Engineering, Parallel & Distributed Systems, 16, pp. 1–7. Available at: https://dl.acm.org/doi/abs/10.5555/1365774.1365790 (Accessed: 25th May 2022).

Mulyadi, I.H. et al. (2009) 'Wireless medical interface using ZigBee and Bluetooth technology'. In *2009 Third Asia International Conference on Modelling & Simulation*, pp. 276–281, DOI: 10.1109/AMS.2009.134.

Zubairi, J.A., Misbahuddin, S., and Tasadduq, I. (2010) 'Emergency medical data transmission systems and techniques'. In *Handbook of Research on Advances in Health Informatics and Electronic Healthcare Applications: Global Adoption and Impact of Information Communication Technologies*, pp. 169–188, DOI: 10.4018/978-1-60566-030-1.ch011.