# Biomedical Signal Processing: Designing an Engineering Laboratory Course Using Low-Cost Hardware and Software

**Mr. Felipe L. Carvalho, Florida Atlantic University**

Felipe L. Carvalho is a graduating senior in the Electrical Engineering program at Florida Atlantic University (FAU), Boca Raton - FL. At FAU, he is a member of the Innovation Leadership Honors Program and as part of his undergraduate studies, is currently working on his Honors Project "Biomedical Signal Processing." Additionally, he is a co-op at BlackBerry, where he works closely with principles of telecommunications and software testing. He is a Tau Beta Pi certified member, where he holds the position of the Chair of Induction and Student Outreach Committee. Mr. Carvalho is also an IEEE student member for the 2013-2014 academic year. His research interests include Biomedical Engineering, Electromagnetics, Power, and Energy Systems.

**Dr. Ravi T. Shankar, Florida Atlantic University**

Ravi Shankar is a senior professor with the engineering college at Florida Atlantic University (FAU), Boca Raton, FL. He has research and teaching interests in computer engineering and science, and electrical and biomedical engineering (see http://faculty.eng.fau.edu/shankar/ ). He is the director of a college-wide center that focuses on systems issues (http://csi.fau.edu/ ). In that capacity, he has built up teaching and research collaborations among professors and students from multiple colleges (arts and letters, business, education, engineering, K-12, and science). The collaborative work is documented at several subject specific FAU websites. He has a PhD from the University of Wisconsin, Madison, WI, and an MBA from Florida Atlantic University, Boca Raton, FL. His doctoral dissertation was on early and noninvasive detection of atherosclerosis. He has been issued 7 US Patents, with 3 of them in the biomedical area licensed for commercialization by Florida Atlantic University. He is a Fellow of the American Heart Association.

# Work in Progress—Biomedical Signal Processing: Designing an Engineering Laboratory Course Using Low-Cost Hardware and Software

**Abstract**

A Biomedical Signal Processing (BSP) laboratory course has a cost associated with its implementation and maintenance. This cost can sometimes be a barrier to be overcome by institutions intending to launch their own comprehensive laboratories for the processing and analysis of biomedical signals (biosignals). As a result, when it comes to BSP, many courses are designed to provide the students with a dry and theoretical application of mathematical methods only, forgetting to focus on hands-on projects to boost the students' learning curve. This work in progress describes the design process of a comprehensive BSP laboratory course using low-cost hardware and software. We describe the methods utilized to generate the documentation needed for the course, and also a detailed list of the materials used and their main features. We divided the design process in two main categories: 1) analog circuit design; and 2) digital signal processing and software development. We provide details about these two categories and sub-divide the structure of the course into seven laboratory experiments to be completed individually or by a group of students. After the completion of this project, a collection of seven laboratory guides will be fully designed and this hands-on BSP course will be ready to be disseminated and implemented. In addition, all the materials and documentation created throughout this project are periodically uploaded into the group's Smart System online webpage (http://smartsystems.eng.fau.edu/biomedical-signal-processing/) for easy and free access.

## Introduction

The characteristics of a Biomedical Signal (biosignal) can be quite intimidating for students. This type of signal – characterized by high common mode signal, low amplitude differential signal, low frequency noise, drift, and cycle-to-cycle variability – is normally challenging to process and to analyze. Based on survey, students tend to do poorly in Biomedical Signal Processing (BSP) courses[1]. Because it is usually a difficult task for students to master BSP courses well, and because of its high implementation costs, a number of institutions still do not offer a course in BSP. Hence, this project is focused on the design of a BSP laboratory course for undergraduate students using low-cost hardware and software.

There is an unmet need to expose undergraduate students to Biomedical Engineering (BME) in a simple and straightforward manner. Application of digital signal processing to processing of biosignals can be fairly difficult. A well designed and implemented laboratory course can suppress the details and provide easy to use interfaces to get students started in the right direction. We hope this will enhance interest and embolden them to undertake further studies and research in the field in later years.

Texas Instruments Inc (TI) has developed low-cost boards and software that will simplify exposure to DSP concepts[3]. TI's Digital Signal Processing (DSP) baseboard and emulator are potentially free for any university developing and/or offering DSP labs; our innovation here is in providing added value. The developed laboratory guides will allow many institutions to quickly launch their own BSP labs. We have already made available online a series of tutorials

covering different laboratory guides' topics, and by the completion of this project, we intend to upload all the laboratory guides and complementary documentation into the group's Smart Systems webpage (http://smartsystems.eng.fau.edu/biomedical-signal-processing/). Our goal is to make BSP widely available, hence our decision to design around low-cost hardware and software, and to provide free and easy access to all of the documentation generated during this project.

**Materials**

One challenge to be overcome by institutions offering BSP laboratories is the cost of implementation and maintenance of such courses. Hence, this project is primarily concerned with minimizing the costs associated with the design and implementation of a BSP undergraduate comprehensive laboratory course. We hope that the chosen low-cost hardware and software, combined with our tutorials and laboratory guides made available online in our group's webpage will stimulate many institutions to launch and offer their own BSP lab courses.

With the cost limitation in mind, we decided to utilize TI's digital signal processor OMAP L-138 LCDK as our main hardware in the processing and analysis of biosignals. This board also requires an additional Joint Test Action Group (JTAG) emulator to interface with the operational system. As far as hardware materials, these two portable and low-power boards provide sufficient hardware functionality to design lab experiments for BSP. For more specific experiments, such as the design of an analog Electrocardiogram (ECG) amplifier, other electrical components – such as resistors, capacitors and operational amplifiers – must be available for use.

Below is a the detailed list of materials that our group is currently working on to design the proposed BSP laboratory course:

1. OMAP L-138 Digital Signal Processor LCDK

TI's OMAP L-138 is a robust, portable, and low-cost development board. The baseboard is Windows compatible, and also contains a fast ARM9 core processor, which enables the direct execution of 8-bit Java bytecode in hardware[2]. Because of the ubiquitous use of the Windows operational system in the academic setting, we chose this board mainly because of its straightforward USB-port connectivity to Personal Computers (PCs) running Windows XP or later.

The dual-core architecture of the OMAP device provides benefits of DSP by enabling applications that require a high-level operating system. In addition, this board provides numerous possibilities for interaction with other hardware peripherals that can easily allow a wider range of applications not only for BSP, but also for DSP in general. The board also contains a built-in AC to DC converter (coder and decoder), which allows the sampling and reconstruction of analog biosignals to be processed onto the same hardware environment.

Many reasons led our group to choose for the TI's OMAP L-138, but in summary, this dual-core, low power, portable and easy to use DSP board fulfill the requirements we foresaw, such as low-cost and versatility.

2. XDS100v2 JTAG emulator

The XDS100 emulator is a low-cost USB-interface JTAG hardware reference design. It is compatible with Windows and its main functionality is to provide an efficient and the necessary connection between the baseboard and a PC. Similarly to the OMAP DSP, this emulator is also low-cost and low power, and it only requires a USB port to be powered. This emulator in parallel with the TI's baseboard provides the sufficient hardware for the processing of digital signals.

### 3. Code Composer Studio™ v5

TI's Code Composer Studio™ (CCS) is an integrated development environment (IDE) for TI's embedded processor families. CCS provides an efficient IDE for the development and debugging of embedded applications. The software is based on the Eclipse open source software framework, which offers an excellent software framework for building software development environments and it is becoming a standard framework used by many embedded software vendors.

CCS also includes a compiler for the OMAP family, a source code editor, project build environment, simulators, and real-time operating system. We are utilizing the CCS environment to develop, debug, test, and implement our algorithms. CCS is compatible with C and C++ high-level programming languages, which is also a convenient feature, since both C and C++ are the widely used programming languages. In summary, the free and online available CCS software combines the advantages of the Eclipse software framework with advanced embedded debug capabilities from TI resulting in a compelling feature-rich development environment for the deployment of embedded systems.

### 4. MATLAB®

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. With MATLAB, one can analyze data, develop algorithms, and create models and applications. In this project, MATLAB provides the main computational tool and environment for the use of the Digiscope program.

### 5. Digiscope

Digiscope is a MATLAB program that allows students to design and run digital filters on pre-sampled ECG signals without doing any programming[6]. The program comes with a library containing a series of pre-sampled ECG signals and an extensive list of signal processing tools and techniques. This free and straightforward MATLAB program is being incorporated into our course as a step prior to the design of algorithms to be implemented onto the OMAP DSP hardware.

Since the use of simulation tools under appropriate technical supervision (i.e. by a Teaching Assistant or a Professor) generates better results in the learning process[3], even using pre-sampled values, Digiscope may assist the students with a better insight on what the expected outcomes will be like prior to the actual code development and implementation.

### 6. Windows PC

The OMAP baseboard is compatible with both Linux and Windows operational systems. For convenience purposes, in this project we are having the board to interact with the Windows

operational system only – that is, Linux is out of this project's scope. TI requires a minimum of 1 GB of memory, 300 MB disk space, and 1.5GHz single core to run the Code Composer Studio (CCS) v5 software. In parallel, Mathworks requires a minimum of 1 GB of memory for MATLAB only, 1024 MB RAM, and Pentium 4 or better processor.

## Course Background

The proposed BSP Laboratory course is targeted to undergraduate students majoring in Electrical Engineering (EE), Biomedical Engineering (BME), or related fields. The course is being designed as a 3000-level course, that is to say, for students in their third year into engineering programs. For EE majors, the course would be offered as an elective, while for BME majors it would be a required class. The proposed laboratory is being designed as a 3.00 credits class – preferably offered twice a week, giving the students sufficient time to complete each experiment.

In order to create an efficient learning environment, we also propose this course to be offered to a maximum of 30 students per section. We believe that 15 groups is the ideal number for a hands-on course, allowing one Professor and one Teaching Assistant (TA) to supervise and assist the students with questions, troubleshooting, and circuit analysis.

By the completion of each experiment, each group needs to turn in a laboratory report comprising of all tables and graphs embedded in the laboratory guide, and their C-language code implemented. In addition, each group must demonstrate their Digiscope simulations to the TA as it will be part of the student's grade.

Each experiment shall be graded as follows:

| | |
|---|---|
| Pre-lab | 15% |
| Digiscope simulations | 15% |
| C/C++ code | 30% |
| Tables and Graphs | 40% |

Table 1: Experiments' Grading Breakdown

## Students Background

We are targeting this course to students who are familiar with basic computer programming, and with electrical circuits concepts. A basic understanding of a high-level computer programming language – preferably C – can be helpful, but is not required, as we intend to develop this skill throughout the course. The laboratory guides will provide a series of walk-through tutorials that will lead the students to successful results even if they don't have a strong background in computer programming. However, a solid understanding of mathematical algorithms and electrical signals is necessary for this class. For this reason, classes such as "Intro to C" and "Circuits 1" would provide all the prerequisites necessary for this course.

Though MATLAB is a tool used throughout the proposed course, there is no need for prior experience with it – as it will mainly be used as the platform necessary to run the Digiscope program. Thus, incoming students are not expected to know MATLAB language. Nonetheless,

we wish to introduce the students to certain MATLAB computations during specific experiments, but all the knowledge necessary to perform such simulations shall be provided in the laboratory guides in a simple and straightforward manner.

**Cost of Implementation**

Assuming an order of 15 lab kits containing the necessary hardware components for this course's deployment, the implementation costs would be roughly $ 4,050.00, which includes 15 OMAP L-138 LCDK, and 15 JTAG XDS100 Emulators.

TI also offers an ECG Front End Performance Kit (ADS1298ECG), that we are considering working with it. The ADS1298ECG is compatible with the C500 processors (OMAP family), provides simultaneous 24-bit signal sampling, an ADC converter, and also a built-in programmable gain amplifier. The main features of this additional hardware involve its easy-to-use evaluation software (compatible with Microsoft™ Windows XP and Windows 7), and the ability to export data in simple test files for post processing. We are considering this product to acquire data in a more accurate and safer manner. Also, it would allow the students to record and store their own ECG signal file, which would potentially engage them more to the course.

If no partnership with TI is possible – which is unlikely since the company has an active University discounts program – and assuming the inclusion of the ADS1298ECG, the lab cost of implementation would be elevated to $ 7,035.00. Below is a detailed list including the retail price of each hardware device:

| Item | Retail Price per Unit (in dollars) | Link |
|---|---|---|
| OMAP-L138 Low-Cost Development Kit | $195.00 | http://e2e.ti.com/group/universityprogram/educators/w/wiki/2063.omap-l138c6748-lcdk-kit.aspx |
| XDS100v2 USB JTAG Emulator | $75.00 | http://e2e.ti.com/group/universityprogram/educators/w/wiki/2063.omap-l138c6748-lcdk-kit.aspx |
| ADS1298ECG Front End Performance Demonstration Kit | $199.00 | http://www.ti.com/tool/ads1298ecgfe-pdk |

Table 2: Retail Price of each Hardware Device

**Methods**

Considering that the targeted students are in their third year, each experiment needs to be carefully designed in order to retain the student's interest. It is imperative to provide the students with a series of detailed and comprehensive steps to avoid frustrations. For this reason, each laboratory guide is being designed in the form of a tutorial, that is, each document is a series of instructions-to-be-followed in order to achieve successful results. We believe that a laboratory guide formatted as such would appeal more to the targeted students, than large paragraphs of instructions.

Each laboratory guide follows the same structure:

- A. Introduction: A brief theoretical introduction to provide the students with all the information needed from circuit design, software development, or digital signal processing. In the lab guide introduction, we also offer the students with an overall motivation to conduct that particular experiment, as well as the challenges that will be tackled.
- B. Pre-lab: Before each experiment, the students should answer a series of 5 quick questions. The pre-lab's role is to allow the students to recapitulate the key points of the theory that will be needed during each experiment.
- C. Procedures: This section of the laboratory guide contains the procedures that each student/group must take in order to complete the experiment. We provide a list of steps, diagrams, circuit schematics, and screenshots in the format of a compelling and easy-to-follow guide. In this section, students can also find practical examples and code samples. The goal of each experiment is to not only introduce the students to BSP, but also to develop their computer programming skills, hence the need for detailed and clear instructions.
- D. Results: In this section, the students can find tables and graphs that need to be completed for grading purposes.

This work in progress will be considered successful upon the completion of seven laboratory guides that will provide a hands-on approach to learning about signal processing through the application of digital signal processing methods to biomedical problems. Thus, in order to achieve the final laboratory guides there is a need to first fully comprehend the hardware architecture of the OMAP baseboard and its software development environment. For this reason, during the past months, our research group has executed numerous experiments involving audio signal processing. Now we are transporting our findings and techniques with audio filtering to the design of each experiment utilizing ECG signals.

We have already developed a series of tutorials and posted them on our Smart Systems webpage. In order to achieve the project's main goals, we divided the project in two main parts 1) the analog design; and 2) code development and implementation (utilizing MATLAB's Digiscope for simulation, and CCS for the final C –language code implementation). Additionally, we structured the course into seven parts. We are designing each experiment to introduce the students to different aspects of BSP[4], in an incremental fashion, on the same original (a 3-lead Electrocardiograph) signal.

The proposed seven laboratory guides are as follows:

1. Emulation of typical external resources such as oscillators and oscilloscopes:

In this part of the project, we researched low-cost alternatives to expensive oscillators and oscilloscopes. Our goal was to utilize software and low-cost embedded hardware to generate electrical signals (thus replacing hardware oscillators), and to allow observation of electrical signals (thus replacing hardware oscilloscopes), respectively. We found two alternatives: Digiscope and WinDSK8.

- a. *Digiscope* is a MATLAB program that allows the user to design digital filters and process normal and abnormal pre-sampled ECG signals. The script also includes a

"function generator" tool, which can be used to generate numerous types of sinusoids, triangle, and square waves.

b. *WinDSK8* is a freely available, highly useful, and versatile Windows-based program with which the user can execute a wide variety of real-time signal processing algorithms[4]. It also includes an "oscilloscope" function, which is able to acquire analog electrical and audio signals from our DSP, and display these waveforms on a computer screen.

This lab guide was structured as a tutorial, where we provided step-by-step instructions, screenshots, and block diagrams. By the end of this experiment, the students will expose exposed to the software tools mentioned above, and will also be able to have some experience in designing basic digital filters (on Digiscope). In this experiment, we guide the students to remove a 60 Hz low-frequency noise of a pre-sampled abnormal ECG signal, using Active Filters.

2   Signal Transduction – Building an analog Electrocardiogram (ECG):

This experiment covers the analog circuit design of an ECG Amplifier. We analyze the characteristics of low-pass, high-pass, band-pass, and band-stop analog filters to build an ECG amplifier.

The necessary components for this experiment include: resistors, capacitors, UA741 operational amplifiers, DC batteries, aluminum electrode tape, and an analog oscilloscope (which can be replaced by the OMAP and WinDSK8 combination).

We encourage each group to observe a member's ECG signal on the oscilloscope's screen. The goal of this lab is to teach concepts such as differential and common mode gain, and low pass filtering.

3.   ECG signal – Distinguishing normal sinus rhythm:

If the ECG Front End described in the last section (Cost of Implementation) is indeed considered, it will be implemented in this experiment. This experiment's main goal is to analyze real ECG signals – either acquired by the ADS1298ECG, or by the analog ECG amplifier built in Experiment #2.

If using the ECG amplifier built in Experiment #2, that amplifier's output can be directly connected to the OMAP's analog input, which will allow it to be further sampled and displayed by winDSK8's oscilloscope function. The ECG signal acquired by ADS1298ECG follows a similar path, but now the signal is sampled before it is connected to the OMAP L-138 baseboard. Regardless the signal acquisition tool used, both techniques allow the signal to be displayed on a computer screen, which will then permit the students to observe the normal sinus rhythm in the ECG waveform and the naturally varying heart rate.

In addition, we include the processing and analysis of both normal and abnormal pre-sampled ECG signals on Digiscope. By combining the use of Digiscope and WinDSK8, we expose the

students to two ways of distinguishing normal sinus rhythm: one from pre-sampled values, and another from real ECG signal acquired in the lab.

4. Signal Filtering – Reducing biosignal drift and offset:

This experiment's goal is to take the output of Experiment #3 for further processing. We intend to provide the basics of digital filtering and look at the advantages, disadvantages, and differences between analog and digital filters. A more mathematical approach is usually how this topic is approached. As seen in Tompkins[5], concepts such as Z-transform, Transfer Function, and the Z-plane Pole-Zero Plot are typically covered when introducing students to digital filters. This type of approach lacks, however, in hand-on exercises, which is what we are striving for in this proposed BSP laboratory.

In this experiment, we intend to implement the use of Digiscope in the first stage of the digital filter's design, which may provide the students with a better understanding of the expected outputs. Subsequently, we intend to cover simple C-language code development for signal filtering onto the OMAP's baseboard. By the end of this experiment, students will have an understanding of the theory behind digital filters, but more than that, a real hands-on experience with the processing of ECG signals acquired in Experiment #3.

The necessary hardware devices for this experiment are the OMAP L-138 LCDK, the JTAG Emulator, and a computer running Microsoft Windows™. The necessary software tools for this experiment include MATLAB (platform necessary to run Digiscope), and Code Composer Studio (C-language development environment for the OMAP family).

5. Identification of the zero crossing points to average and display the averaged biosignal:

This experiment will cover techniques for averaging the biosignal. Averaging is desired in order to visually observe normal and pathological ECG signals. We will also use it to extract features for automated pattern recognition.

6. Pattern recognition from the averaged signal:

We will use the averaged signal to identify the normal sinus rhythm; this experiment will be an extension of Experiment #5, ad pattern recognition is only possible after averaging the signal.

7. Simple diagnostics from pattern recognition:

At the basic level, we wish to automatically detect cases of low and high heart rates. This is not pathological for a fairly broad range of heart rates. Students can achieve these via transcendental meditation (or deep and slow breathing) and in-place jogging, respectively. At the advanced level, we wish to extend the lab to identify pathological conditions such as myocardial infarction. For this, we will access available online databases to acquire such data. This experiment will cover techniques for feature extraction from biosignals and consequent diagnostics. This advanced part will serve to connect the lab work with real world applications.

All the experiments covered in this proposed BSP laboratory are designed to be compatible with TI's OMAP L-138 LCDK. Each laboratory guide is written upon this group's achievement of the expected outcomes. Thus, all the proposed techniques are previously tested to assure their quality. We have been designing the laboratory guides as follows:

I. Depending on the topic to be discussed in the laboratory guide—e.g. sampling, filtering, pattern recognition—we first conduct a similar experiment using non-biomedical signals. In general, we have been processing tunes (pure sinusoid signals) and audio signals. The deterministic behavior of such signals provides a better understanding of the OMAP processor, as well as accurate results that confirm the response expected.

II. After conducting the experiment using a deterministic input signal and observed the output of such system, we now modify the experiment to the context of BME. This involves the processing of real signals – such as the output signal of the ECG built in the experiment 2 – or the processing of pre-sampled signals.

III. Once the experiments are conducted, i.e. we process and analyze the proposed biosignal, we write the laboratory guides and tutorials for that particular series of BSP experiments.

**Anticipated outcomes**

The anticipated outcomes of this project are:

1) To prepare the above listed tutorials and make them available online along with all the software and circuit schematics; and

2) To develop a comprehensive Biomedical Signal Processing Engineering Lab that can be replicated easily.

**Conclusion and Project Status**

So far we have completed the laboratory guide of Experiment 1, we have accomplished the design and specification of the analog ECG[5] (Experiment 2), and signal filtering (Experiments 3 and 4). In addition, we completed the tutorial on how to design and build this ECG, the tutorials on how to get started with the OMAP processor, the tutorials on how to get started with the Code Composer Studio v5, and a brief tutorial on the OMAP L-138 processor. All of these comprehensive tutorials guides are already available online in the group's Smart System online webpage (http://smartsystems.eng.fau.edu/biomedical-signal-processing/).

**References**

1. Li, L. and Li B., *Design of Experiments in Biomedical Signal Processing Course*, 30th Annual International IEEE EMBS Conference, Vancouver, British Columbia, Canada, August 20-24, 2008.
2. Klostermann, M., Christ, O., Mankodiya, K., Vogt S., and Hofmann U.G., *OMAP 3 Based Signal Processing For Biomedical Engineering Teaching*, 17th European Signal Processing Conference (EUSIPCO 2009), Glasgow, Scotland, August 24-28, 2009.

3.  Damassa, D.A., *Simulation Technologies in Higher Education: Uses, Trends, and Implications*, ECAR Research Bulletin 3, 2010. Boulder, CO.
4.  Welch, T.B., Wright, C.H.G. and Morrow, M.G., 2011, *Real-time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs,* CRC Press, Boca Raton, FL, 436p.
5.  Tompkins, W.J. and Webster, J.G., 1981, *Design of Microcomputer-Based Medical Instrumentation*, Prentice Hall, Englewood Cliffs, NJ, 496p.
6.  Tompkins, W.J., 2000, *BIOMEDICAL DIGITAL SIGNAL PROCESSING – C-Language Examples and Laboratory Experiments for the IBM® PC,* Prentice Hall, Englewood Cliffs, 359p.