Use of FreeRTOS in Teaching Real-time Embedded Systems Design Course

Dr. Nannan He, Minnesota State University, Mankato

Nannan He received the Ph.D. in computer engineering from Virginia Tech. From 2012 to present she is an Assistant Professor at the ECET department in Minnesota State University at Mankato. Her teaching and research interests are in safety-critical embedded software, real-time embedded systems, and software verification. She is an IEEE member and reviewers for many conferences and journals in EDA field.

Dr. Han-Way Huang, Minnesota State University, Mankato

Ph.D. in computer engineering from Iowa State University Professor of ECET Department of Minnesota State University, Mankato
Use of FreeRTOS in Teaching Real-time Embedded Systems Design Course

Abstract

This paper presents our experiences of teaching the course “Real-time Embedded Systems Design” by applying the free and open source Real-Time Operating System (RTOS) called FreeRTOS. The emphasis is placed on how we adopted FreeRTOS as a real-world RTOS example in both lectures and lab sessions from exercises preparation, lab equipment setup to lab organization. Compared with existing real-time computing courses, the main difference of this course is that we focus on teaching students the design and application development of real-time embedded systems from the practitioner’s point of view, instead of introducing research or theoretical topics. FreeRTOS is a real-time kernel/scheduler designed to run on a microcontroller for embedded applications. It supports a large number of underlying microcontroller architectures and has become the leading real-time computing platform for microcontrollers. In this course, it has been applied to conducting experiments with multitask scheduling algorithms and the real-time interfacing with microcontrollers for all our lab sessions and course projects. Our primary experiences indicate that FreeRTOS is a richly featured, cost-efficient and well supported RTOS for teaching real-time systems design and developing microcontroller-based real-time applications.

Introduction

Nowadays, with the emergence of new processors and methods of processing, communications and infrastructure, modern industrial automation systems require high real-time control capabilities. There is an on-going work to achieve the education goal of increasing the technical depth and broaden training by investigating deterministic timing techniques in complex real-time automation systems at Minnesota State University, Mankato. As an important exploration step towards this goal, a new real-time embedded system design course has been offered to electrical engineering and computer engineering senior or graduate level students. At the same time, the goal is also an important guideline in the course preparation and teaching practices, as a result some special features of this course are formed compared with most existing real-time systems design courses.

Real-time computing courses are mostly offered to computer science major students at the senior or graduate level, with the aim of equipping them with the knowledge of conducting the scientific research in this area. These courses typically focus on the theoretic topics in real-time computing, such as various reference models of real-time systems, schedulability theory, design and timing analysis of multi-task scheduling algorithms and operating systems. However, the
real-time embedded systems design course presented in this paper emphasizes engineering issues of designing and developing real-time systems in practical embedded applications like automation. The course is taken by senior electrical engineering and computer engineering students and some graduate students pursuing their Master degree. Other engineering students with the appropriate software background could also take the course. It emphasizes teaching students real-time systems design and applications from the practitioner’s point of view, instead of targeting at research and theoretical topics as conventional real-time computing courses. After taking this course, students are expected to demonstrate the ability of correctly defining and designing real-time systems, and the ability of basic real-time application development. Active learning and hands-on learning are the fundamental teaching approaches applied to this real-time systems design course.

This new course covers the topics on RTOS relevant topics, such as multi-task scheduling, system services, and resource policies and some application issues for developing real-time systems, such as microcontroller, requirement analysis, performance analysis and verification of real-time system design. Among these topics, RTOS is one of the core components of our new course. In this course, we employed an existing free, open source Real-time Operating System called FreeRTOS as a case study of RTOS in both lectures and lab sessions.

FreeRTOS is a real-time kernel/scheduler designed to be small enough to run on a microcontroller. It provides the real time scheduling functionality, inter-task communication, timing analysis and synchronization primitives for teaching RTOS. It also offers the rich example projects as the bases for developing embedded real-time systems. Moreover, FreeRTOS supports a large number of underlying microcontroller architectures including advanced ARM Cortex™-Mx series, and has become the standard RTOS for microcontrollers. To simplify the structure of the application code, The FreeRTOS software provides time-related Application Programming Interfaces. As a result, complex embedded real-time applications can be efficiently built to meet their real-time processing deadlines on top of FreeRTOS. In this course, FreeRTOS was applied to conducting experiments with multitask scheduling algorithms and real-time interfacing with microcontrollers for all our lab sessions and course projects.

This paper presents the primary experiences of teaching real-time embedded systems design to engineering students, with the emphasis of how we adopted FreeRTOS as a real-world RTOS example in teaching to improve the teaching effectiveness. The description of this course is first given, including course contents, learning outcomes and instruction approach. Next, a survey of existing real-time operating/runtime systems is reported. Then, this paper describes the software FreeRTOS and how we make use of FreeRTOS in lab assignments and course projects from exercises preparation, software setup and implementation. Finally, the paper gives a conclusion and discusses the future work.
2. Real-time embedded systems design course description

Our real-time embedded systems design course targets the learning of real-time systems design and applications from the practitioner’s point of view. It has been offered for two years. This course is organized as two hours of lecture and three hours of laboratory per week. It has three main objectives.

- To improve students’ awareness of real-time specifications in critical automation controllers and other embedded systems.
- For engineering students to apply modern development tools and advanced techniques to designing and analyzing performance of small-scale real-time systems.
- To enable students to develop real-time applications to solve problems with specific timing requirements.

Moreover, in order to accomplish the basic instruction approach of active learning and hands-on learning, this course has an experiential component. It allows students to apply advanced techniques learnt from this course to develop an understanding of their advantages and disadvantages in different applications.

Course learning outcomes

Our overall aim is to equip students with the knowledge of designing real-time systems and developing real-time applications to solve engineering problems in practice. In the development of this course, we identify course learning outcomes that stem from this aim and extend to learning activities. We developed twelve learning outcomes, classified into three core components as shown in Figure 1.

1. To demonstrate the ability of correctly defining real-time systems
   1.1 To identify problems as hard, firm or soft real-time system;
   1.2 To articulate and contrast different definitions in real-time systems.

2. To demonstrate the ability of real-time systems design
   2.1 To comprehend formal methods based specification approaches and utilize modeling tools;
   2.2 To understand the impact of hardware for real-time performance;
   2.3 To analyze the scheduling feasibility of a set of independent tasks and derive schedules;
   2.4 To understand resource policies and system services for inter tasks communication and synchronization;
   2.5 To understand the challenges and applications of performance analysis techniques;
2.6 To understand real-time issues on advanced distributed control networks such as SCADA;

3. To demonstrate the ability of basic real-time application development
   3.1 To understand real-time software testing, verification and system integration.
   3.2 To be aware of performance optimization techniques.
   3.3 To utilize modern tools to simulate executions and critique different implementation choices.
   3.4 To comprehend the architecture, functions and applications of one or two existing RTOS.

**Course contents**

The topics covered in this course include real-time scheduling approaches such as clock-driven scheduling and static and dynamic priority driven scheduling, resource handling, timing analysis, RTOS, hard and soft real-time systems, distributed real-time systems, concepts involved in the modeling, design, analysis and verification of real-time systems. Course materials were drawn from two text books 1, 2, FreeRTOS tutorial book and reference manual, ARM Cortex-M microcontrollers’ datasheets, websites, and other publications. Table 1 shows the classification of these topics. In this table, the time schedule of each top-level topic is given for this one semester course (~ 15 weeks). Please note that the topic 7 - Case studies (FreeRTOS) is not labeled with a specific schedule because its sub topics are provided in the combination with other topics throughout the semester.

<table>
<thead>
<tr>
<th>Table 1. Course topics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Fundamentals (week 1)</strong></td>
</tr>
<tr>
<td>Basic concepts and misconceptions</td>
</tr>
<tr>
<td>Multidisciplinary design challenges</td>
</tr>
<tr>
<td><strong>2. Hardware for real-time systems (2nd-3rd)</strong></td>
</tr>
<tr>
<td>Process architecture</td>
</tr>
<tr>
<td>Architectural advancements</td>
</tr>
<tr>
<td>Peripheral interfacing</td>
</tr>
<tr>
<td>Distributed real-time architecture</td>
</tr>
<tr>
<td><strong>3. Real-time operating systems (3rd - 6th)</strong></td>
</tr>
<tr>
<td>Multi-task scheduling</td>
</tr>
<tr>
<td>System services for application programs</td>
</tr>
<tr>
<td>RTOS selection issues</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Instruction approach

Active learning and hands-on learning are fundamental teaching approaches applied to this real-time system design course. All of the classes were held in the laboratory. For this course, this setting eases the flexible adoption of a variety of teaching methods, depending on the characteristics of different course topics in sequence. The main teaching formats and material employed in this course are presented as the following.

At the beginning, we used power point slides presentation and class discussion to introduce students the topics on defining real-time systems. These topics are basis for further learning. Thus, it is important to help students to set up a solid and comprehensive foundation. In the class discussion, some questions are designed to enable students to reflect on key concepts in real-time systems, and to encourage active learning. Here are some examples: 1) Are real-time systems synonymous with ‘fast or high performance’ systems?, 2) “In the statement ‘All practical systems are ultimately real-time systems’, what is your idea of the degree of ‘real-time’?, 3) “Where could a response time requirement of a system come from?”. As the discussions proceed, students gradually deepen their comprehension at the same time are inspired to judiciously observe and analyze real-time applications. The initial homework exercises in this course are extracted from practical application problems. One example question is: considering an automation system for the car assembly, describe three different event scenarios, classify the system as hard, firm or soft real-time under each of these scenarios. Such exercises are designed with open-end questions. The goal is for students to think and give the justification for their answers, and to fortify students’ understanding so as to be able to apply it in practice.

The main purpose of introducing formal requirement engineering techniques is for students to develop an appreciation of automated formal or semi-formal methods in rigorously specifying real-time system requirements. An example sub-system requirements document for “Fuel Management” from AIRBUS was introduced as a case study. Students are convinced that the formal specification like State chart is not just a scientific research issue, in fact has been widely used in requirement specification of safety-critical embedded systems in industry to avoid the ambiguity caused by conventional text-based specifications. Later on, they showed great enthusiasm in learning formal or semi-formal approaches. In the course evaluation, students made several comments that the requirement engineering offered in this course was one of the most beneficial aspects to them.

There is the increasing number of microcontrollers (MCUs) supporting real-time applications. In this course, the following MCU development boards and Integrated Development Environment (IDEs) are employed to be available for students.

- PIC24, dcPIC (Explorer 16 Development board from Microchip)
- ATmel SAM4S-EK (ARM Cortex-M4 microcontroller from Atmel)
- ATmel SAM4L-EK (ARM Cortex-M4 microcontroller for low power from Atmel)
The above MCUs are selected as they are all supported by the FreeRTOS software which is the main RTOS studied in this course.

The last part of this course offers students the topics which are served for the application development core component. As real-time systems are often applied in the ‘critical’ embedded applications with respect to reliability, safety and security, verification and validation (V&V) is an important issue in real-time application development. An on-going research on the model-based V&V is incorporated in this course so that students could be exposed with latest V&V advances.

### Table 2. A survey of RTOSs

<table>
<thead>
<tr>
<th>Name</th>
<th>License</th>
<th>Platforms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeRTOS</td>
<td>Free</td>
<td>ARM, IA32, Cortex-M3, PIC, MSP430, STM32</td>
<td>A real-time kernel designed to be small enough run on a MCU, be portable to a large number of MCUs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.freertos.org">http://www.freertos.org</a></td>
</tr>
<tr>
<td>CooCox CoOS</td>
<td>BSD</td>
<td>ARM, STM32, NXP LPC1000, TI LM3s8963,</td>
<td>An embedded real-time multi-task OS special for ARM Cortex M series.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.coocox.org/CoOS.htm">http://www.coocox.org/CoOS.htm</a></td>
</tr>
<tr>
<td>QNX</td>
<td>Mixed</td>
<td>ARM, IA32, MIPS, PowerPC, SH-4, xScale</td>
<td>A commercial Unix-like RTOS, targeting at general purpose usage, being widely used with in a variety of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>devices including cars and mobile phones.</td>
</tr>
<tr>
<td>Window CE or WinCE</td>
<td>Proprietary</td>
<td>ARM, MIPS, x86, SuperH</td>
<td>An OS and kernel developed by Microsoft for embedded systems; a variety of IDE supporting development for WinCE</td>
</tr>
<tr>
<td>scmRTOS</td>
<td>Free</td>
<td>ARM, Cortex-M3, MSP 430, AVR</td>
<td>A free tiny preemptive RTOS intended for use with Single-Chip MCUs. The key features are max speed and min code/RAM size requirements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="http://scmrtos.sourceforge.net/ScmRTOS">http://scmrtos.sourceforge.net/ScmRTOS</a></td>
</tr>
<tr>
<td>Micrium uc/OS-II</td>
<td>Proprietary</td>
<td>ARM, ST32 PIC24</td>
<td>A priority-based pre-preemptive real-time multitasking operating system kernel, featuring unlimited application tasks.</td>
</tr>
<tr>
<td>CoDeSys SP Runtime System</td>
<td>Proprietary</td>
<td>X86, ARM, Infineon TriCore</td>
<td>Full-feature OS; Component based architecture for customer adaptations to various platforms.</td>
</tr>
</tbody>
</table>
3. RTOSs survey

Every real-time system contains some operating system (OS) like functionalities: 1) Providing an interface to the input/output hardware; 2) providing coordination of virtual concurrency in a uniprocessor environment; 3) providing true concurrency with multi-core processors and distributed system architectures. In general, a RTOS has three main principal goals: 1) to offer a reliable, predictable, and low-overhead platform for multi-tasking and resource sharing; 2) to make the application software design easier and less hardware bound; 3) to make engineers in various industry fields to concentrate on their core product knowledge. However, full-feature operating system (OS) is not always the good solution in embedded systems for two main reasons. First, it is hard to meet design constraints of low-end embedded systems, such as system cost and complexity, response time and the punctuality. Second, the MCU is heavily occupied by the system software, instead of executing the time-critical application.

Nowadays, there are many RTOSs available from both commercial and research communities. These RTOSs have different license types, target usage and supported platforms. In order to select the suitable RTOS for our teaching purposes, we first did a survey of existing RTOSs whose target usage is in embedded applications, as shown in the table below. We follow three selection principles: 1) the license is low-cost or free; 2) The RTOS is portable to a large number of MCUs; 3) The RTOS is well-documented. Compared with other RTOS systems, FreeRTOS is free and supports most common MCUs from multiple manufactures. Moreover, it is easy to access a comprehensive FreeRTOS reference manual and lab exercises developed using FreeRTOS.

4. Introduction to FreeRTOS

FreeRTOS is chosen as the case study of RTOS to study multitask scheduling policies and real-time interfacing with MCUs. It provides real time scheduling functionality, inter-task communication, timing analysis and synchronization primitives. On top of FreeRTOS, complicated Cortex-M4 MCU applications can be efficiently built to meet their hard real-time requirements. It supports organizing these applications as a collection of independent threads of execution.

Since each of the MCUs used in our lab has one processor core, only one task can be executing at any one time. By computing the scheduling policy designed by the application developer, FreeRTOS is responsible of deciding which task (i.e., one thread of execution) should be executing at a time. Using the priority-based scheduling as an example, the application code first assigns higher priorities to tasks that implement hard real-time requirements, and lower priorities to tasks that implement soft or firm real-time requirements. Then, FreeRTOS could take care of the rest of the work and determine which task should be executing by examining the priority
assigned to each task by the applications. Thus, hard real-time tasks can always guarantee to be executed ahead of soft real-time tasks.

Besides helping ensure an application meets its processing deadlines, FreeRTOS can bring other benefits. To help students better understand the usage of this RTOS, three other main benefits are introduced through lab exercises. First, the kernel is responsible for timing and provides time-related APIs to the application. This allows the structure of the application code to be simpler and the overall code size to be smaller. Second, the idle tasks which are created automatically by the RTOS can be used to measure spare processing capacity, to perform checks in the background, or to place the processor into a low-power mode. Third, gatekeeper tasks provided by the RTOS can be used to serialize access to peripherals.

5. Application of FreeRTOS

Educators have explored hands-on RTOS development for enhancing student learning in real-time systems course . The lab assignment based on the FreeRTOS is the important experiential component of this course, which aims at giving student rich hands-on experience in building real-time systems.

Three things are prepared for tutoring each lab session before students start working on the lab assignment. First, a list of questions is designed for students to figure out the answers during each lab session. These questions serve as the guidelines to assist students’ hands-on learning. Second, the concepts or algorithms related to each lab work, which have been introduced in lectures, are reviewed. Thirdly, as software programing is the main task in each lab, a set of relevant API functions are introduced to students for the efficient programming. These APIs are provided by either FreeRTOS or certain libraries included in a particular IDE. A standard demo project which incorporates MCU development board, simulator, logic analyzer and miniature RTOS with all basic features, is provided to students. They make use of this demo project as the basis to construct more application specific projects, which could achieve more efficient development compared with creating application from scratch. According to the course evaluation report, students gave the feedback that they benefit most from the hands-on programming experience.
Lab assignments on FreeRTOS

The lab assignments based on FreeRTOS consist of five key components. Figure 2 shows core lab components. More details are presented in the following:

1. Task Management
   Include 8 lab exercises on creating tasks, creating tasks, experimenting with task priorities, using Block state to create a time delay, combining continuous processing tasks and periodic tasks, defining an idle task hook function, changing task priorities, and deleting a task.

2. Queue Management
   Have 4 lab exercises on creating queue for task-to-task communication, sending data to a queue, receiving data from a queue, managing the stored data, blocking on a queue, and the effects of task priorities when sending to and receiving from a queue.

3. Interrupt Management
   3 lab exercises are arranged on deferred interrupt processing, using binary semaphores / counting semaphores to synchronize a task, and using queues within an interrupt service routine.

4. Resource Management
   2 lab assignments are given on mutual exclusion, mutex, priority inheritance and gatekeeper tasks for protecting resource access.

5. Memory Management
   1 lab assignment on different memory allocation schemes to illustrate their pros and cons.

Other tools are also employed for system development in this course, for example, FreeRTOSplusTrace from Atmel is used for simulating and observing execution traces for debugging.

6. Conclusion

This paper presents our primary experience of teaching the new course “Real-time Embedded Systems Design”. First, the description of this course is presented. Second, we introduce a survey of RTOSs portable to ARM platform, especially an open source RTOS – freeRTOS with which the lab assignments are designed. In the future, we hope to apply another IDE Keil µVision which is dedicated for ARM programming to this course, refine lab assignments and propose new projects on developing FreeRTOS-based real-time applications.

References


DICOM, MRI and Bioinstrumentation using Matlab and Simulink

Dr. Mohammad Rafiq Muqri, DeVry University, Pomona
Prof. Shih Ek Chng, DeVry University, Pomona

Professor College of Engineering and Information Science
DICOM, MRI and Bioinstrumentation using MATLAB and Simulink

BMET students interested in biomedical signal processing, digital imaging and communications in medicine (DICOM), picture archiving, communication system (PACS), and bioinstrumentation are deprived of the opportunity to take background courses such as 2D-signal processing, computer communications, radiography, and sensors and instrumentation. Compare to traditional electrical engineering students, the lack of hands-on lab experience becomes more apparent when students are working on capstone senior projects.

One strategy we used to solve this issue was to include a mixed capstone project group comprising of computer, electronic, and biomedical engineering students. This strategy worked for few groups but the success rate was less than thirty five percent due to the students’ lack of preparation and disadvantage of knowledge compared to traditional electrical engineering students.

To rectify this situation we proposed and developed this teaching module which incorporates well selected signal processing, biomedical imaging and instrumentation topics which make extensive use of MATLAB, Simulink, and LabVIEW tools. This teaching module includes a detailed description of associated core lab exercises, student responses and recommendations. This low cost program consists of a series of theory modules coupled with a hands-on laboratory component using readily available test equipment and graphical capabilities of MATLAB, Simulink and LabVIEW software. As such this paper concerns problem solved and lessons learned while developing computer-assisted instruction strategy to improve the current state of learning in the classroom. This will introduce the students to new topics not covered in traditional courses. The goal of this paper is to pass on information useful to anyone contemplating related work, where similar occurrences are likely. This paper will explain how this learning and teaching module was instrumental in progressive learning of these students, improving their performance and successful culmination of capstone senior projects.

Setting up a Biomedical Instrumentation Laboratory

Teaching students how to use specialized instruments and equipment that are currently used in the medical field do not serve the purpose for engineering students trying to apply the knowledge of engineering into medicine. Most equipment is designed for taking measurements only which limit students from practicing engineering. They are mainly used for research purposes because they are too complex and very costly. The development of a Biomedical Instrumentation Lab for the undergraduate program requires thoughtful planning especially for most teaching institutions offering BS-BMET program; it has limited budget for acquiring specialized instrument and equipment to be used for conducting experiments mainly for educational purposes. Developing a single platform that is flexible enough to perform most undergraduate physiological experiment in lab become possible with the Biomedical Workbench developed by National Instrument. This platform allows students to develop their own test system with the latest measurement technologies while providing an environment for developing an application in the capstone project.
The instrumentation setup which can be used for most human physiology laboratory experiment is shown in figure 1 below.

![Biomedical Instrumentation Setup Diagram](image)

**Figure 1 Biomedical Instrumentation Setup**

Typical experiments using this basic instrumentation setup includes Cardiovascular Physiology test, Neurophysiology test, and Pulmonary Ventilation test. The following are a compilation of experiments for each category.

A. Cardiovascular Physiology Experiments
   1. Electrocardiogram and Heart Sounds
   2. Electrocardiogram and Peripheral Circulation
   3. Exercise, the Electrocardiogram and Peripheral Circulation
   4. Blood Pressure, Peripheral Circulation and Body Position
   5. Blood Pressure, Peripheral Circulation and Imposed Conditions

B. Neuro Physiology Experiments
   1. Electroencephalography (EEG)
   2. Electromyography (EMG)
   3. Evoked potentials (EP)
   4. Reflexes and Reaction Times

C. Pulmonary Ventilation
   1. Breathing Parameters at Rest and After Exercise
   2. Breathing and Gravity
   3. Factors that Affect Breathing Patterns
The instrumentation setup can be divided into 4 major parts. I.) The Silver (Ag)-Silver Chloride (AgCl) electrodes, II.) high performance bio-potential amplifier with build-in hardware filter from iWorx, III.) National Instrument Elvis II+, and IV.) Laptop/PC running Biomedical Workbench, LabVIEW, and MATLAB.

I.) Ag-AgCl electrodes — Each 4-mm diameter reusable button scalp electrodes is made of silver covered with silver chloride coating. This electrode has impedance typically less than 100 Ω which is most suitable for detecting EEG signal voltage typically less than 50 μV. To maximize skin contact and reduce artifacts due to movement of the electrode during acquisition of EEG signal, electrode gel is applied in between the electrode and the scalp. The electrodes are connected to the Isolated Module which is connected to the Bio-Potential Amplifier.

II.) Bio-Potential Amplifier with build-in hardware filter — The pre-amplifier is a low noise instrumentation amplifier with very high CMRR 85dB up to 200 Hz and input impedance greater than 10GΩ. To provide enough signal level for voltage acquisition by Elvis II plus, the amplifier is equipped with 8 selectable voltage gain of up to x5k. Hardware analog filter is integrated into the amplifier to allow preprocessing of raw signal by removing unwanted signal pickup from body movement or noise of 60 Hz riding on the baseline of the EEG signal. EEG signal frequency typically ranging from 0.5Hz the low end of Delta wave to 30 Hz the high end of Beta wave. The High pass and Low pass filter cutoff frequency is set to 0.3Hz and 50Hz respectively on the Bio-Potential amplifier to meet this preliminary bandwidth requirement.

III.) Elvis II plus — Elvis II+ is a Modular Engineering Educational Laboratory Hardware platform by National Instrument for use in data acquisition. It is connected to PC with a high speed USB cable. The output of the Bio-Potential amplifier is connected to the analog input (AI) which is configured to measure voltage. Since EEG signal contains multiple channels of data corresponding to different positions on the brain, up to 8 analog input channels can be connected to Elvis II plus at any given time.

IV.) Biomedical Workbench — The diagram in figure 2 shows the latest Biomedical Tool Kit from National Instrument. This application software is used for real-time acquisition or generation of simulated bio signals.
Appendix A illustrates the procedure for Instrument Control Settings on this application and the experimental results from Evoked potentials in Neurophysiology experiment using the Biomedical instrumentation setup described above. This experiment detects the electrical signals of the brain in response to light stimulation of the eyes.

**Advantages of using the Biomedical Workbench**

The illustration above shows how simple the Biomedical Workbench software together with the Elvis-II plus hardware platform is to acquire real time biomedical signal. One advantage of using the software is to be able to create multiple virtual channels simultaneously with individual filtering parameters from one acquired real time signal. This feature is especially important for analyzing EEG signal because the acquired biosignal for several given bandwidth of interest can be generated in one acquisition. The ECG Feature Extractor to extract ECG features for heart rate variability analysis is also proven to be useful as well as the Noninvasive Blood Pressure Analyzer. To create a meaningful lab, student should take the acquired data and process through MATLAB or LabVIEW program to try to match with the results from the Feature Extractor and Analyzer from Biomedical Workbench and draw conclusions about the accuracy of their own algorithm. The Feature Extractor and Analyzer is by no means a model answer but can be used as a reference or secondary resource to justify the results from analysis.

Another useful feature is the file format converter. Institutions currently using different hardware equipment that generate different data file format can be easily converted to .tdms file format for use in Biomedical Workbench. Figure 3 lists the file formats that can be converted and saved.
For radiographic imaging experiment, the 3D image reconstructor helps student visualize 3D model from a set of 2D image slices. This software accepts external image files from real applications in different file formats including DICOM, BMP, JPEG, and PNG. Instructor can provide sample DICOM files downloaded from the website to the students. By using this feature, students can view the detailed internal structure in three dimensions, which provides detailed information about the region of interest (figure 4).
Implementation of a real time DSP system

In our university, we have introduced the Tower System Microcontroller for two Eight-Week sessions to the students before the capstone project. Students are required to know how to setup and connect the Tower board (Figure 5) and ADC to DAC data acquisition board (APPENDIX figure B1) for the filter experiments. Students design the digital filter for a given design specification with the MATLAB SPTOOLS graphical filter design editor using a Parks-McClellan iterative algorithm for digital filter coefficients determination. The filter will then be implemented using the CodeWarrior, an integrated development environment (IDE), for the creation of program that runs on the Tower System Microcontroller (APPENDIX figure B2). Signal conditioning using operation amplifier for anti-aliasing and anti-imaging filter is also included in the curriculum to enhance their knowledge in analog filter design. Performance of the Tower embedded system board can be tested by using the NI Elvis Instrument Launcher (APPENDIX C). The Function Generator has frequency sweep capabilities which allow Bode Analyzer to create a frequency response automatically of the real time DSP system. In the capstone project, this real time DSP system can be connected to the Elvis II plus data acquisition platform for real time biosignal processing application.

![Figure 5 Tower System Microcontroller Setup](image)

Reading image file using MATLAB

In Capstone project, student who is interested in 2D signal processing application is to use MATLAB Image Acquisition toolbox to acquire images and video from hardware. Students use different commands in MATLAB to experiment and practice them on their favorite sample images. The following image formats are supported by MATLAB:

- BMP
- HDF
- JPEG
- PCX
- TIFF
- XWB
Most sample images found on the Internet are JPEG-image that is widely used compression standards for images. An image named xxxx.jpg is stored in the JPEG format. APPENDIX D show how an image can be loaded into MATLAB. A digital image is composed of pixels which can be thought of as small dots on the screen. Each pixel is represented by a binary number which describe the color of the pixel.

In general, there are four basic representation of a pixel:

1. **Binary.** Each pixel is either black or white. Since there are only two possible values for each pixel, we need only 1 bit per pixel, as such they are efficient in terms of storage. Images for which a binary representation may be suitable include text (handwriting or printed), architectural plans, or fingerprints.
2. **Grayscale.** Each pixel is a shade of gray, normally from 0(black) to 255 (white). This range implies that each pixel can be represented by 8-bits (1-byte). Other gray scale ranges are also used, but generally they are a power of 2. Such images arise in medicine (x-rays), and images of printed works etc.
3. **True color.** Color digital images require three values to be recorded for each pixel: one red component, one green component, and one blue component. These components are combined with different weightings to produce a range of colors. If each of these components has a range 0-255, this gives a total of \(256^3 = 16,777,216\) different colors in the image. Since total number of bits required for each pixel is 24, such images are also called 24-bit color images.
4. **Indexed.** Most color images have only a small subset of the more than 16 million possible colors. For convenience of storage and file handling, the image has an associated color map, or color palette, which is simply a list of all colors used in that image. Each pixel has a value that does not give its color (as for a RGB image, but an index to the color in the map).

Images from one image type can be converted to another image type by using MATLAB functions as shown in the table below.

<table>
<thead>
<tr>
<th>Function</th>
<th>Use</th>
<th>format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ind2gray</td>
<td>Indexed to grayscale</td>
<td>x = ind2gray(y, map);</td>
</tr>
<tr>
<td>gray2ind</td>
<td>Grayscale to indexed</td>
<td>[x, map]=gray2ind(y);</td>
</tr>
<tr>
<td>rgb2gray</td>
<td>RGB to grayscale</td>
<td>x = rgb2gray(y);</td>
</tr>
<tr>
<td>gray2rgb</td>
<td>Grayscale to RGB</td>
<td>x = gray2rgb(y);</td>
</tr>
<tr>
<td>rgb2ind</td>
<td>RGB to indexed</td>
<td>[x,map]=rgb2ind;</td>
</tr>
<tr>
<td>ind2rgb</td>
<td>Indexed to RGB</td>
<td>x=ind2rgb(y,map)</td>
</tr>
</tbody>
</table>

**Table 1  MATLAB image converter functions**

APPENDIX D shows sample of MATLAB commands for reading and converting 2D image. Student can further process the converted 2D image for some specific applications.
Digital Signal Processing using Java

Last but not the least, the four Java programs or code snippets which is shown in APPENDIX E were used by engineering technology students before even they have been exposed to control theory or DSP courses. This was accomplished by a class lecture in OBP which was followed by examples of simple substitutions in the Java program code; however most motivated students will want to do more than that. The Faculty here at our university attempted to indulge students in such activities. Figure E1 is an example of an instructor lead program which the student edited, compiled and displayed the output. The purpose of this program is to show how general Java workhorse discrete Fourier Transform and other control theory methods 7 can be introduced at an earliest stage to engineering technology students with the tools and concepts they will further reinforce in future DSP courses.

The TestDFT application class shown in APPENDIX figure E2, uses class Fourier and invokes its methods. Highly efficient algorithms for computing the DFT were first developed in the 1960s. Collectively known as Fast Fourier Transforms (FFT’s), they all rely upon the fact that the standard DFT involves redundant calculation. Strictly speaking, there is no such thing as ‘the FFT’ 3. Rather, there is a collection of algorithms with different features, advantages, and limitations. An algorithm which is suitable for programming in a high level-language on a general purpose computer may not be the best for special purpose DSP hardware. What the different algorithms have in common is their general approach – the decomposition of the DFT into a number of successively shorter, and simpler, DFTs.

There are various ways of explaining FFT decomposition. We can show that a DFT can be expressed in terms of shorter, simpler, DFT’s by dividing the signal x[n] into subsequences. The method which is widely used in DSP literature is also referred to as conventional decomposition. Then there is also an alternative approach known as index-mapping. It should be clear in our mind that conventional decomposition and index mapping are just two ways of looking at the same problem and there is no essential difference between them.

Supposed we have a signal with N sample values, where N is an integer power of 2. We first separate x[n] into two subsequences, each with N/2 samples. The first subsequence consists of even number points in x[n], and the second consists of odd number points- Writing n = 2k, when n is even, and n = 2k + 1 when n is odd. We can thus express the original N-point DSP in terms of two N/2 point DFTs. Now we can take the decomposition further, by breaking each N/2- point subsequence down into two shorter, N/4-point subsequences. The process can continue until, in the limit, we are left with a series of 2-point subsequences, each of which requires a very simple 2-point DFT. A complete decomposition of this type gives rise to one of the commonly used radix-2, decimation in time, FFT algorithms.

Now the students are ready to implement an FFT as a Java method. It is called the fastFFT( ) Method as shown in figure E3 and is also defined in the Fourier class.
In the next step, the students apply this method to compute the FFT on a 2Hz cosine wave. They were instructed to take 64 data samples over a 2-second sample period. The program first computes the FFT to obtain the frequency spectrum for a 2Hz cosine wave. Then the program was used by students to perform an inverse Fourier transform that reconstructs the 2Hz cosine wave from its frequency spectrum. Next, we implement the FFT as a Java method. It is called the fastFFT( ). The next thing to do is to test the fastFFT( ) by applying it to the composite cosine signal that were processed earlier. The amplitude time history for a signal containing three different frequency components is generated and sent to fastFFT () method. The TestFFT class source code is shown in figure E4. These are just a few of the representative Biomedical Instrumentation, DSP using Java programming, Image processing using MATLAB laboratory modules to which students get exposed.

**Feedback and Assessment**

Continuous examining the evolving needs of our students and employers for career-oriented higher education programs as basis for development of additional programs is our university mission and purpose. Agencies accrediting our programs are also increasingly focused on student outcomes and achievement. Student outcomes are the skills and abilities students are expected to demonstrate at graduation. One of the student outcomes in our program is the ability to conduct standard tests and measurements; to conduct, analyze, and interpret experiments; and to apply experimental results to improve biomedical systems and processes.

We have offered this DICOM, MRI, Bioinstrumentation course as a special topics course which is one way new courses are piloted locally at our university campus. This course is offered to target student outcome as described above. This course is instrumental in the progressive learning of the students by relating and applying fundamentals of circuit analysis, analog and digital electronics, micro-computing, power electronics, electro-mechanics, and various energy and material concepts to bioengineering systems. Although special topics are not evaluated in the same manner as standard session-long courses, feedback directly from students indicates that initial offering was well-received.

A compilation of feedback was from four junior students enrolled in the course for credit, and three sophomore students attended the lectures and discussions but did not receive the credit.

The feedback from the student indicates that:

1. Students gained an appreciation of the extensive background required by our engineering and computer information sciences technology program. MRI, DICOM and Bioinstrumentation puts many concepts in calculus, linear algebra and java programming to practical use in a context that has been described as “cool” by the students.

2. Students gained an appreciation of the natural sciences courses required by our engineering program, particularly physics. Often students reported that the physical concepts related to a particular topic (for example electrostatics, resonance, Amperes Law, and Faraday’s Law of electromagnetic induction ) made much more sense after implementing the concept in the context of the Magnetic resonance Imaging and bioinstrumentation applications.
3. Students gained an appreciation for the difficulties involved in developing and debugging complex software system. The DFT and FFT coding was for many students, the first java programming experience with a non-trivial code base that has to be designed and written from scratch and leverages the power of java data structures.

4. Students spent a significant amount of time on the signal processing with MATLAB, Simulink, applications of FFT and Java programming assignments (presumably relative to their other course work) but the results were satisfying. We did not receive any complaints about the level of effort required by, nor the time spent on the programming assignments.

Student’s performance in the initial course offering and in the course of capstone projects was exceptionally high. This result was due to a biased sampling; the four juniors taking the special topic course initiated the effort, and the sophomores that attended regularly were invited by the instructor. We hope to see better understanding of basic principles and excellent performance in the future versions of the course, but a wider range of students will undoubtedly test the validity of mixed performance in course of capstone senior projects.

**Conclusion**

With proper guidance, monitoring, and diligent care, the biomedical engineering technology students can be exposed earlier to medical instrumentation, Java data structures and the basics of MATLAB. Recent assessment results have shown that with the layout of training modules before taking the capstone project, the fear of BMET students taking the capstone projects is not only eliminated, but has also built up their confidence and improved the quality and creativity of their projects. With proper conditioning and judicious course selection, students will become more motivated and will help reinforce the best practices in implementing capstone senior projects.
Bibliography

9. ZhijunGu, NI Biomedical Start up Kit 3.0, July 2010, https://decibel.ni.com/content/docs/DOC-12646
APPENDIX A: Instrument Control Settings and Results for Evoked potentials (EP) experiment

There are two Instrument Control Settings on this application. When the selector switch is on the Logger Settings mode, step 1 of 3 for Setting involve selecting the physical channel (AI), Input Range, and Terminal Configuration. The diagram in figure A1 shows one Physical Channel Setting. Up to 8 analog input channels (ai0-ai7) can be acquired simultaneously on Elvis II plus. Input Range is set to -5/+5V. This allows acquisition of amplified raw EEG signals up to +/- 5V after the Bio-Potential amplifier. The input range can be changed to improve signal resolution. Differential Input is selected to amplify the signals between the +/- electrodes.

![Figure A1 Physical Channel Setting](image)

Step 2 involves setting the filtering parameters for the 4 main frequency ranges of the EEG signal. Figure A2 to A5 shows the filter parameters used for Delta, Theta, Alpha, and Beta Wave. A tabulation of the four filters parameters is shown in figure A6. Butterworth filters are used for all 4 virtual channels with Passband ripple of 0.1dB and Stopband attenuation of 60 dB.
<table>
<thead>
<tr>
<th>Virtual Channel Name</th>
<th>Passband edge frequency (Hz)</th>
<th>Stopband edge frequency (Hz)</th>
<th>Bandpass Filter Designed Order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower</td>
<td>Upper</td>
<td>Lower</td>
</tr>
<tr>
<td>Delta Wave</td>
<td>0.5</td>
<td>3</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta Wave</td>
<td>4</td>
<td>7</td>
<td>3.5</td>
</tr>
<tr>
<td>Alpha Wave</td>
<td>8</td>
<td>13</td>
<td>7.5</td>
</tr>
<tr>
<td>Beta Wave</td>
<td>14</td>
<td>30</td>
<td>13.5</td>
</tr>
</tbody>
</table>

**Figure A6**  Tabulation of the four filters parameters

Step 3 in figure A7 shows all 4 virtual channels and the physical channels are enabled for 30 seconds duration of logging time. Logging will stop automatically after the 30 second duration has past.

**Figure A7**  Logger setting

Significant increase of Alpha Wave amplitude was detected as shown in figure A8 when eyes were closed and relaxed as compared to figure A9 when eyes were open and concentrated starring at the computer monitor.
Figure A8  Eyes wide opened and stared at the computer monitor

Figure A9  Eyes closed and relaxed
APPENDIX B: TOWER_ADCDAC_AXIOM board

This board features one ADC and one DAC. The user can access the channels of these chips from the green connection strips located on the right and left of the board. Note that the ADC inputs are on the left side (TB1) and the DAC outputs are on the right side, as shown in the picture below. Also note that there are two DAC output strips. The output of the DAC is bipolar on the upper right connector (TB4) with 4 outputs, but unipolar on the lower right connector with 8 outputs (TB2).

![Figure B1 ADC/DAC AXIOM Board](image)

```c
// Filter Array

#define FIR_SIZE 143

int FIRinput[FIR_SIZE]; // Holds moving average filter inputs from ADC
int FIRcoeffs[FIR_SIZE] =
0.187, -0.13, 0, -0.15, 0.15, 0.27, 0, -0.33, 0.121, 0.10, -0.241, 0.25, -0.188, 0, -0.142, 0.236, -0.151, 0.438, 0, -0.562, -0.433, 0, -0.69, 0, -0.398, 0,
-0.763, 0.827, 0.491, 0.274, 0, -0.435, 0.1448, 0, -0.139, 0,
0.311, 0.1051, 0.708, 0.509, 0.0561, 0, -0.7101, 0, -0.6561,
0.509, 0.5022, 0.101, 0, -0.131, 0, -0.1379, 0, -0.1448, 0, -0.0433, 0,
-0.174, 0.493, 0.857, 0.763, 0, -0.89, 0, -0.433, 0,
-0.161, 0.418, 0.150, 0, -0.161, 0, -0.1382, 0, -0.185, 0.342, 0, -0.16,
0.185, -0.285, 0.241, 0, -0.81, 0.121, 0, -0.293, 0.379, 0,
0.156, 0.370, 0, -0.150, 0, -0.187, 0;

// Function main

void main(void)
{
    int RFL0(); // Initialize the PLL to a system clock of 25 MHz
    setSampleFreq(16); // Set sampling frequency in Hz
    // The integer value argument to the sampling frequency
    // Please note that the oscillator frequency on some Tower boards has been measured as much as 4% off. This will directly affect the frequency response of any and all digital filters with the most notable effect being shifting the cutoff frequencies up or down by approximately the same frequency percentage as the oscillator offset.
    // Initialize the FIR input array
    // Initialize the FIR input array
    if (FIRinput[10] == 0)
        for (i = 0; i < FIR_SIZE; i++)
            FIRcoeffs[i] = 0;
        return;

    // Main
        return;
}
```

![Figure B2 Implementation of FIR filter coefficient in CodeWarrior](image)
APPENDIX C: NI ELVISmx Instrument Launcher

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Multimeter</td>
<td>DC voltage, AC voltage, Current (DC and AC), Resistance, Capacitance, Inductance, Diode test, Audible continuity</td>
</tr>
<tr>
<td>Oscilloscope</td>
<td>Two channels, scaling and position adjustment knobs, modifiable timebase, autoscaling, digital or analog hardware triggering, cursors for accurate screen measurements</td>
</tr>
<tr>
<td>Function Generator</td>
<td>SMHz, output sine, square, or triangle waveforms, amplitude selection, and frequency settings, DC offset setting, frequency sweep capabilities, amplitude and frequency modulation</td>
</tr>
<tr>
<td>Variable Power Supply</td>
<td>Positive (0 and +12 V) or negative (−12 and 0 V)</td>
</tr>
<tr>
<td>Bode Analyzer</td>
<td>Set the frequency range of the instrument, choose between linear and logarithmic display scales</td>
</tr>
<tr>
<td>Dynamic Signal Analyzer</td>
<td>Continuous or single scan measurements, apply various window and filtering options</td>
</tr>
<tr>
<td>Arbitrary Waveform Generator</td>
<td>Uses Waveform Editor (included with NI ELVISmx), load waveforms, generate two waveforms simultaneously, run continuously or once</td>
</tr>
<tr>
<td>Digital Reader</td>
<td>Reads digital data from eight consecutive lines at a time (0, 7, 8, 15, 16, 23), continuous single reading,</td>
</tr>
<tr>
<td>Digital Writer</td>
<td>Manually create a digital pattern or select predefined patterns (ramp, toggle, walking 1s), control eight consecutive lines, continuous or single write, TTL compatible</td>
</tr>
<tr>
<td>Impedance Analyzer</td>
<td>Capable of measuring the resistance and reactance for passive two-wire elements at a given frequency.</td>
</tr>
<tr>
<td>2-wire Current-Voltage Analyzer</td>
<td>Conduct diode parametric testing, view current-voltage curves; full flexibility in setting parameters such as voltage and current ranges</td>
</tr>
<tr>
<td>3-wire Current-Voltage Analyzer</td>
<td>Conduct transistor parametric testing, view current-voltage curves; base current settings for measurements of NPN and PNP transistors</td>
</tr>
</tbody>
</table>

Figure C1  NI ELVIS Instruments
APPENDIX D  Sample MATLAB commands for reading 2D image

Images are read into MATLAB environment using function imread, whose syntax is

```matlab
g >> imread (‘file-name’)
```

1. Here file-name is a string containing the complete name of the image file (including any applicable extension). For example, the command line

```matlab
g >> I = imread (‘x-ray-white2.jpg’)
```
reads the JPEG image x-raywhite2 into image array I. Then type

```matlab
g >> imshow(I)
g >> J = im2double(I);
g >> whos
```

2. Depict and explain the output for this step .

3. Now try using this command line, which is really three commands on one line.

```matlab
g >> figure, imshow (I), pixval on
```

The three commands we use here are :

- **figure**, which creates a figure on the screen. A figure is essentially a window in which a graphics object can be placed. Objects may include images or various types of graphs.

- **imshow (I)**, which displays the matrix I as an image.

- **pixval on**, which turns on the pixel values in our figure. This is a display of the gray values of the pixels in the image. They may appear at the bottom of the figure in the form

```matlab
c, r = p
```

where c is the column value of the given pixel; r its row value, and p its gray value.
5. Given below are some examples of MATLAB commands and their typical outputs.

```matlab
>> I = imread('D:\MyDocuments\My Pictures\ImageP\cell1.jpg');

>> whos

Name      Size      Bytes     Class  Attributes
I         512x512   262144    uint8

>> imshow(I)

>> J = im2double(I);

>> whos

Name      Size      Bytes     Class  Attributes
I         512x512   262144    uint8
J         512x512   2097152  double
```

For this step 6 use your favorite digital color image.

Again given below are some more examples of MATLAB commands and their typical outputs.

```matlab
>> I = imread('D:\MyDocuments\My Pictures\ImageP\dv-portall.jpg');

>> whos

Name      Size      Bytes     Class  Attributes
I         227x730x3  497130    uint8
J         512x512   2097152  double

>> K = rgb2gray(I);

>> whos

Name      Size      Bytes     Class  Attributes
I         227x730x3  497130    uint8
J         512x512   2097152  double
K         227x730   165710   uint8

>> imshow(K)

>> I = imfinfo('D:\MyDocuments\My Pictures\ImageP\cell1.jpg')
```
I =

<table>
<thead>
<tr>
<th>Filename:</th>
<th>[1x43 char]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileModDate:</td>
<td>'13-Apr-2013 21:12:43'</td>
</tr>
<tr>
<td>FileSize:</td>
<td>98531</td>
</tr>
<tr>
<td>Format:</td>
<td>'jpg'</td>
</tr>
<tr>
<td>FormatVersion:</td>
<td>''</td>
</tr>
<tr>
<td>Width:</td>
<td>512</td>
</tr>
<tr>
<td>Height:</td>
<td>512</td>
</tr>
<tr>
<td>BitDepth:</td>
<td>8</td>
</tr>
<tr>
<td>ColorType:</td>
<td>'grayscale'</td>
</tr>
<tr>
<td>FormatSignature:</td>
<td>''</td>
</tr>
<tr>
<td>NumberOfSamples:</td>
<td>1</td>
</tr>
<tr>
<td>CodingMethod:</td>
<td>'Huffman'</td>
</tr>
<tr>
<td>CodingProcess:</td>
<td>'Sequential'</td>
</tr>
<tr>
<td>Comment:</td>
<td>{}</td>
</tr>
</tbody>
</table>

```matlab
>> H = figure, imshow('D:\MyDocuments\My Pictures\ImageP\cell1.jpg'), pixval on
```

```matlab
>> H = figure, imshow('D:\MyDocuments\My Pictures\ImageP\cell1.jpg'), impixelinfo
```

```matlab
>> H = figure, imshow('D:\MyDocuments\My Pictures\ImageP\cell1.jpg'), imdistline
```
public class Fourier {
    public static double[] discreteFT(double[] fdata, int N, boolean fwd) {
        double X[] = new double[2*N];
        double omega;
        int k, ki, kr, n;
        if (fwd) {
            omega = 2.0*Math.PI/N;
        } else {
            omega = -2.0*Math.PI/N;
        }
        for(k=0; k<N; k++) {
            kr = 2*k;
            ki = 2*k + 1;
            X[kr] = 0.0;
            X[ki] = 0.0;
            for(n=0; n<N; ++n) {
                X[kr] += fdata[2*n]*Math.cos(omega*n*k) + fdata[2*n+1]*Math.sin(omega*n*k);
                X[ki] += -fdata[2*n]*Math.sin(omega*n*k) + fdata[2*n+1]*Math.cos(omega*n*k);
            }
        }
        if (fwd) {
            for(k=0; k<N; ++k) {
                X[2*k] /= N;
                X[2*k + 1] /= N;
            }
        }
        return X;
    }
}

Figure E1 DFT program
public class TestDFT {
    public static void main(String args[]) {
        int N = 64;
        double T = 2.0;
        double tn, fk;
        double fdata[] = new double[2*N];
        for(int i=0; i<N; ++i) {
            fdata[2*i] = Math.cos(4.0*Math.PI*i*T/N);
            fdata[2*i+1] = 0.0;
        }
        double X[] = Fourier.discreteFT(fdata, N, true);
        for (int k=0; k<N; ++k) {
            fk = k/T;
        }
        for (int i=0; i<N; ++i) {
            fdata[2*i] = 0.0;
            fdata[2*i+1] = 0.0;
            if (i == 4 || i == N-4) {
                fdata[2*i] = 0.5;
            }
        }
        double x[] = Fourier.discreteFT(fdata, N, false);
        System.out.println();
        for (int n=0; n<N; ++n) {
            tn = n*T/N;
        }
    }
}

Figure E2  TestDFT Program
public class Fourier {
    public static void fastFFT(double[] fdata, int N, boolean fwd) {
        double omega, temp, tempi, fscale;
        double xtemp, cosine, sine, xr, xi;
        int i, j, k, n, m, M;

        j=0;
        for(i=0; i<N-1; i++) {
            if (i<j) {
                temp = fdata[2*i];
                tempi = fdata[2*i + 1];
                fdata[2*i] = fdata[2*j];
                fdata[2*i + 1] = fdata[2*j + 1];
                fdata[2*j] = temp;
                fdata[2*j + 1] = tempi;
            }

            k = N/2;
            while (k <= j) {
                j -= k;
                k >>= 1;
            }
            if (fwd)
                fscale = 1.0;
            else
                fscale = -1.0;
            M = 2;
            while (M < 2*N) {
                omega = fscale*2.0*Math.PI/M;
                sin = Math.sin(omega);
                cos = Math.cos(omega) - 1.0;
                xr = 1.0;
                xi = 0.0;
                for (m=0; m<M-1; m+=2) {
                    for (i=m; i<2*N; i+=M*2) {
                        j = i + m;
                        temp = xr*fdata[j] - xi*fdata[j+1];
                        tempi = xr*fdata[j+1] + xi*fdata[j];
                        fdata[j] = fdata[i] - temp;
                        fdata[j+1] = fdata[i+1] - tempi;
                        fdata[i] += temp;
                        fdata[i+1] += tempi;
                    }
                    xtemp = xr;
                }
            }
        }
    }
}
\[
x_r = x_r + x_r \cos - x_i \sin; \\
x_i = x_i + x_temp \sin + x_i \cos;
\]

\[
M *=2;
\]

\[
\text{if} (\text{fwd} ) \{ \\
\text{for} (k=0; k<N; k++) \{ \\
fdata[2*k] /= N; \\
fdata[2*k + 1] /= N; \\
\}
\}
\]

**Figure E3** FFT program

```java
public class TestFFT {
    public static void main(String args[]) {
        int N = 64; double T = 1.0; double tn, fk;
        double fdata[] = new double[2*N];

        for(int i=0; i<N; ++i) {
            fdata[2*i+1] = 0.0;
        }
        Fourier.fastFT(fdata, N, true);
        System.out.println();
        for(int k=0; k<N; ++k) {
            fk = k/T;
            System.out.println( "f[" + k + "]" = " + fk + " Xr[" + k + "]" = " + fdata[2*k] + " \\
                Xi[" + k + "]
            = " + fdata[2*k+1]) ;
        }
    }
}
```

**Figure E4** Test FFT program
Development of a Suit of Virtual Experiments for Physics and Chemistry Undergraduate Laboratories

Miss Oluymemisi Oladayo Satope, iLabs OAU

Satope Oluymemisi is a developer at iLabs OAU and a final year student in the department of Electronic and Electrical Engineering OAU. She has worked with iLab in robotics education for high school students. Also, as the chairperson of Women in Engineering OAU Student Branch, she has been involved in several tech programmes for high school girls and students in general. Presently she is working on online education with online laboratories for physics and chemistry experiments in iLabs.

Mr. Isaiah Oreoluwa Boboye, Obafemi Awolowo University
Mr. Olawale Babatunde Akinwale, Obafemi Awolowo University

Olawale B. Akinwale earned his first degree at the Department of Electronic and Electrical Engineering, Obafemi Awolowo University, Ile-Ife, finishing with first class honors in 2004. He obtained his second degree from the same department in 2011. He is a lecturer at the Obafemi Awolowo University Ile-Ife in Electronic and Electrical Engineering, majoring in Instrumentation. He is also a lab developer in the OAU iLab Research Group. He developed the first reported robotic arm remote lab in Africa making use if the MIT iLab shared architecture and National Instruments LabVIEW. His interests include online experimentation, methods in enhancing pedagogy, machine learning and artificial intelligence, and home automation.

Prof. Lawrence O Kehinde P.E., Obafemi Awolowo University, Ile-Ife, Osun State

Professor Lawrence Kunle Kehinde, a former Departmental Chair, Engineering Dean and University Deputy Vice Chancellor, received his B.Sc. 1st class Honours in Electronics (1971), and a PhD, Control Engineering (1975), at the University of Sussex UK. As a Fellow of the international Atomic Energy Agency, USA; He had his Post Doctoral Studies in Nuclear Instrumentation at the University of California, Berkeley USA (1977-1978). He has spent most of his years as a Professor of Instrumentation Engineering at the Obafemi Awolowo University (OAU), Ile-Ife, Nigeria where he still teaches. He was the Rector of the first private Polytechnic in Nigeria. He recently concluded a 3-year Visiting Professor term at the Texas Southern University, Houston Texas USA. He has worked in Techno-Managerial position as the Director of ICT at OAU for years. His major field is Instrumentation Designs and has designed various equipment. He was the founding Principal Investigator of the University’s iLab research and he currently designs remote and virtual experiments for remote experimentation in Science and Engineering. He is at present the Coordinator of a State Research and Educational Network in Nigeria. He is a Chartered Engineer, a Fellow of both the Computer Association of Nigeria, and Computer Professionals of Nigeria and a Member of IEEE and ASEE. He is also a reviewer for journals. He has over 75 publications in Journals and Proceedings. He also jointly held two British Patents in the past.

Prof. Olabode Idowu Asubiojo, Obafemi Awolowo University
Abstract

Remote Labs have been able to provide students and instructors with the avenue to perform experiments anytime. Several virtual laboratories are readily available on the internet some of which were also developed by the iLab OAU team. However, most of the virtual laboratories developed are focused on desktops and also require the installation of one or two plugins or runtime for the execution.

Inadequate funding and lack of equipment is a bane of undergraduate laboratory practice in Physics and Chemistry. If it is possible for students to have virtual equivalents of these experiments, they can easily perform them on-line thereby increasing their understanding and speed of completion if and when they get to the real lab.

Using Physics and Chemistry laboratory manuals of freshmen at a university, the authors have developed virtual laboratories for selected experiments. These include Calorimetry- (Specific Heat Capacity, Latent Heat of Fusion of Ice), Ohm’s law, pH measurement and Acid base Titration. This work involves a synergy of faculties in Physics, Chemistry, Electronics and Computer Engineering. This paper presents three of the developed experiments. The idea is to eventually upgrade all designed experiments in future to a MOOCs suite that a number of universities can use.

Introduction

In the study of science, experiments are indispensable for development of skills to deal with physical processes and instrumentation. Experiments are performed to verify the theories taught in class, analyse systems and ultimately translate laboratory experience into real life application. The use of traditional laboratory has been adversely affected over time as a result of rising cost of laboratory equipment and increasing number of students’ enrolment into schools. As the number of student participation rises, there is a need for a complementary provision of platforms for experiments to be performed or at least perform them with a close replica of the actual laboratory experience.

Virtual laboratories are software applications that use the theory behind the experiment to model a laboratory setting. Virtual laboratories have a number of advantages. Firstly, with virtual laboratories, the student can repeat a particular experiment multiple times with different parameters to observe different scenarios with relatively no extra cost per performance of the experiment. Secondly, and this partly derives from the above, virtual laboratories allow students to learn from failures that may arise without causing any real damage to systems. Learning from failure is one of the objectives for engineering education as defined by ABET. Another benefit of virtual laboratories is that they can be adapted to Open On-line Courses for several students in different locations, hence opening up to the benefits of Massive Open On-line Courses (MOOCs).
Many emerging technologies have been used to develop laboratories. Recent researches have indicated the use of techniques and software such as LabVIEW and Matlab/Simulink, Java applet, Flash, Ajax etc. However, many of these laboratories require the user to have one software or the other installed on their systems before using the laboratories.

To solve this problem, the work reported here not only develops virtual laboratories for undergraduate experiments but also uses a technology which is readily available on most browsers and does not require any pre-installation by the user. This helps to bring the laboratory experience as close as possible to the users.

Developing The Virtual Laboratories

Two points which were borne in mind during the development process of the virtual laboratories are as stated by Dr Mark Schulz in his paper:

1. Not all experiments are well suited to being performed online and some are more so than others.
2. Remote experiments cannot completely replace traditional laboratories. They cannot completely deliver the traditional laboratory experience but can be very useful in cash-strapped institutions as well as in well-funded institutions for additional student exercises.

Intricate to making a virtual laboratory is pedagogy. A virtual laboratory is simply a model of a traditional (physical) laboratory. As with every model, the virtual laboratory only represents the "essential characteristics" of the traditional laboratory. The most essential characteristic of any laboratory is its pedagogy i.e. the aim of the experiment and the skills the student is expected to learn by performing the experiments. For example, in a titration experiment, while the student is expected to be able to identify colour change and hence the point at which to stop the titration, there is little or no pedagogic value in the student knowing how much force he would need to turn off the tap of the burette. Hence, in modelling a titration experiment, mouse clicks or arrow keys on the keyboard can be used to open and close the burette's tap but good graphics or a webcam must be used to show the colour of the solution in the conical flask during the titration process.

Once the pedagogics of the laboratory had been sorted out, technical / technological issues were then taken into account. These included issues like which online technologies to use and how much required bandwidth would be acceptable. Research has shown that there is pedagogical value in implementing a realistic virtual laboratory interface. Presenting a realistic interface hence meant taking a careful look at graphics. The technical / technological issues were thus divided into two major parts: Graphics and Programming.

Technologies Used for Design Process of the Virtual Laboratories

To provide the ease of use of virtual laboratories across various devices and platforms, the authors employed HTML5 which is a technology readily available on most browsers alongside JavaScript that has contributed immensely to interaction on web. HTML5 also makes it possible to have one lab which works both on desktop and laptop computers as well as all mobile devices. Other technologies which have been used to create lab clients include ASP.net, C#, Java, LabVIEW, MATLAB and Adobe Flash. There are also a
plethora of software which are used for developing native apps for mobile devices, apps based on iOS, Android and Windows Phone 8 operating systems.  

Construct 2 which is an HTML5 based multimedia content creator developed by Scirra Limited was used in the development. Construct 2 provides an easy to learn drag and drop programming platform with the use of a visual editor and a behaviour-based logic system. Development in Construct is achieved with the use of behaviours and event systems which allows the developer to think logically and grasp real programming concepts easily. Flexibility and extensions can be achieved with the use of third-party plugin and those created by the developer himself with the use of the available JavaScript SDK. Some of the available objects and plugins (inbuilt and third-party) in Construct 2 are shown in figure 1.

![Figure 1: Construct 2 with inbuilt and third-party plugins](image)

**Graphics**

In order to make the graphics interface as realistic as possible, the favoured ways of obtaining images for the developed labs could be to take pictures of actual laboratory setups or to download images of the apparatus from the internet. Images can however also be drawn using Photoshop and CorelDraw.

Image sizes are a main concern in virtual laboratories as large image sizes means large application sizes which requires a larger bandwidth to perform the experiments or else long times for the loading of the images. On the other hand, small image sizes generally mean low image resolutions which can greatly reduce the pedagogical value of the virtual laboratory.

**Programming**

The programming component is the functionality component. Here, the logic for the virtual laboratory was developed while constantly referring to the pedagogics of the traditional
The Developed Virtual Experiments

We will now present three of the developed virtual experiment suites, which are associated with physical first year labs in Physics and Chemistry.

a. **Acid-Base Titration Experiment**

Titration is a process by which the concentration of solution, called the analyte or titrand can be calculated by use of another solution of known concentration (called the titrant or titrator). Titration is the process of adding a measured volume of the analyte to a known volume of the titrant until full reaction has taken place between these two solutions. This point of full-reaction is often found by observing the colour of an indicator which had been added to the titrant before the titration started. In acid-base titration, an acid is placed in the burette and is added to a base in a conical flask until the solution in the conical flask has been neutralized. The concentration of the analyte can now be found using equation 1.

\[ C_A V_A = C_B V_B \]

where
- \( C_A \) is the concentration of the acid solution used
- \( V_A \) is the volume of the acid
- \( C_B \) is the concentration of the base
- \( V_B \) is the volume of the base

While the above is accurate, it should be noted that the base in the conical flask can as well be the analyte in which case titration would involve adding a measured volume of an acid of known concentration (titrant) to a known volume of a base of unknown concentration (analyte). Equation 1 still holds true in this case.

**Apparatus:**
- Burette, Beaker, Pipette, Conical (Erlenmeyer) flask, Retort stand

**Materials:**
- Titrant, Analyte, pH indicator

**Procedure:**
1. The burette is placed on the retort stand.
2. The burette is filled with the acid.
3. The beaker is filled with a known volume of the base solution.
4. The pipette is used to take a known volume of the base solution from the beaker to the conical flask.
5. A few drops of the indicator are added to the base solution in the conical flask.
6. The conical flask containing the base-indicator mixture is placed beneath the burette on the retort stand.
7. The tap on the burette is opened to titrate while carefully watching the solution in the conical flask for a colour change which would indicate the solution has been neutralized.
8. At the point of neutralization, the tap is closed.
9. The volume of the acid used is recorded.
10. Calculation of the unknown concentration is done using equation 1.

The Developed Virtual Acid Base Titration Experiment:

The graphic design of this experiment was broken down into three modules (shown in figure 2). Each module was handled by a different layout.

Module one: Introduction
Lab Technologist introduces the experiment to the students.

Module two: Experiment
Experiment module itself.

Module three: Help
Help section where a lab technologist is called for assistance

Figure 2: Modules of the Titration lab suite.

Figures 3 to 5 show some screen shots in the graphic representation of the experiment

Figure 3: A Screen shot showing the laboratory technologist introducing the experiment (module 1)
The complete virtual experiment for the titration experiment can be called at the following site: [http://ilab-titration.appspot.com](http://ilab-titration.appspot.com)

b. **Measurement of Specific Heat Capacity**

Calorimetry is the measurement of the flow of heat energy into or out of a system. While heat flow may be difficult to be directly measured, its effect (temperature change) can easily be measured. We indirectly measure heat flow by measuring the change in temperature in a system. A change in the temperature of a substance is an indicator of heat having flowed into or out of a substance. Hence, heat flow can be measured by measuring the amount of
temperature change a substance undergoes given that the mass of the substance and the specific heat capacity of the substance are known.

Calorimetry is done by a device called a calorimeter. Using a calorimeter, heat flow can be calculated using equation 2.

\[ q = mC \Delta T \]

where
- \( q \) is the amount of heat that flowed
- \( m \) is the mass of the substance
- \( C \) is the specific heat capacity of the substance
- \( \Delta T \) is the change in temperature of the substance

**Apparatus:**
- Calorimeter, heater, digital scale, thermometer

**Material:**
- Water, metal samples (with known specific heat capacities).

**Procedure:**
1. The mass of the empty calorimeter is measured and recorded.
2. The calorimeter is filled about half way with water.
3. The mass of the half-filled calorimeter is measured and recorded.
4. The mass of the water in the calorimeter is now calculated.
5. The temperature of the water in the calorimeter is measured.
6. A metal sample is selected from the available samples
7. The mass of the selected sample is measured and recorded.
8. The sample is heated to about 100°C.
9. The heated sample is transferred to the calorimeter filled with water.
10. The thermometer is watched until a steady temperature is attained.
11. The change in temperature is computed and then the heat flow is computed using equation 2.

**The Developed Specific Heat Capacity Experiment:**

The experiment was set up into five major layouts which handled all the procedures. Each layout is made up of a lab assistant with guidelines on how to navigate the layout. At the end of the experiment the necessary details and standard parameters needed are provided for the student. Figures 6 to 9 show some screen shots of the Specific Capacity Experiment.
Figure 6: Screen shot showing filling of calorimeter with water (Step 2 of the experiment procedure)

Figure 7: Screen shot showing measuring of the mass of selected metal sample B (Step 7 of the experiment procedure)
The complete virtual experiment for the specific heat capacity experiments can be called at the following site: http://ilab-heatcapacity.appspot.com

c. **Latent Heat of Fusion of Ice**

Latent heat of fusion is the heat which melts a solid at its melting point without raising its temperature. The word latent comes from the fact that though the heat is being applied to the
solid, its temperature does not rise (hence, the heat seems to be "hidden"). Instead, the solid melts. This only occurs at the solid's melting point.

The amount of heat needed to melt a solid at its melting point (latent heat of fusion) is dependent on the mass of the solid. Equation 3 gives the relationship between the heat absorbed by the solid and the latent heat of fusion.  
\[ Q = ML_f \]  
where
- \( Q \) is the amount of heat absorbed by the solid,
- \( M \) is the mass of the solid
- \( L_f \) is the latent heat of fusion.

**Apparatus:**  
Calorimeter, heater, digital scale, thermometer

**Material:**  
Ice, water

**Procedure:**
1. The mass of the empty calorimeter is measured and recorded.
2. The calorimeter is filled about half way with water.
3. The mass of the half-filled calorimeter is measured and recorded.
4. The mass of the water in the calorimeter is now calculated.
5. The temperature of the water in the calorimeter is measured.
6. Several dry pieces of ice are added to the calorimeter until the temperature of the mixture is between 5-10°C below room temperature.
7. When all the ice has melted the equilibrium temperature is measured and recorded.
8. The new mass of the calorimeter (plus water plus water from the melted ice) is measured.
9. Equation 3 is used to compute the latent heat of fusion of the ice.

**The Developed Latent Heat of Fusion Experiment:**

The experiment was set up into four major layouts which handled all the procedures. Each layout is made up of a lab assistant with guidelines on how to work through the layout. At the end of the experiment the necessarily details and standard parameters needed are provided for the student. Figures 10 to 12 show some screen shots of the Latent Heat of Fusion Experiment.
Figure 10: Screen shot showing the student taking temperature reading of water (Step 5 of the experiment procedure)

Figure 11: Screen shot showing the student taking temperature reading after all ice has melted (Step 7 of the experiment procedure)
The complete virtual experiment for the latent heat of fusion experiments can be called at the following sites: http://ilab-latentheat.appspot.com

Lab Assessment

This is a work in progress and the virtual labs are yet to be fully deployed. However, in order to improve the interface and functionality of the developed laboratories and also get a general overview of these lower level students’ responsiveness to the virtual laboratory initiate, an initial quick survey for the titration virtual lab was carried out using 25 first year students from different science and engineering departments in Obafemi Awolowo University.

In the survey, after carrying out the virtual experiment, students were asked to respond to the following 5 questions using a 5-point scale where 1 is bad and 5 is excellent:

1. How well did the lab communicate the supposed experiment?
2. How was the user experience?
3. Have you performed the experiment in a real laboratory before? If yes, how well does the virtual lab compare to the real lab?
4. How easy was it getting your way around the lab?
5. How appealing was the lab interface?

Students’ overall responses are shown below in table 1.
Table 1: Students' assessment of the virtual lab

<table>
<thead>
<tr>
<th>Question</th>
<th>1 (Bad)</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5 (Excellent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Question 2</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>Question 3</td>
<td>No</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>23</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Question 4</td>
<td></td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Question 5</td>
<td></td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

A general comment section was also provided for the students. From this section it was obvious from the various comments, that the virtual experiment was a new experience for most of the students, the experience was however a good one as most of them are looking forward to it being a full part of their curriculum. While this initial assessment looks rather positive, it is inconclusive due to the few number of students involved and the fact that the work is still ongoing.

It is planned to include these experiments as part of the laboratory coursework for the students in the next sessions. The students would be expected to use the virtual laboratories as a support to the real laboratories for the upcoming session. The results and report prepared from these experiments would also be graded. Finally, each student would be expected to fill a feedback from after performing the experiment, to help improve the laboratories further.

Cost

Since this is an on-going work, the final cost has not been determined. However, Table 2 presents the expenditure so far, for the developed virtual laboratories. It should be noted that while the following purchases were made for the particular laboratories developed, the cost of the purchased software is independent of the number of laboratories developed. Hence, the cost of each virtual laboratory reduces with the increase in the total number of virtual labs developed.

Table 2: Budget for virtual labs.

<table>
<thead>
<tr>
<th>S/N</th>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Construct-2</td>
<td>$119</td>
</tr>
<tr>
<td>2.</td>
<td>Microsoft Visual Studio</td>
<td>$1,199.00</td>
</tr>
<tr>
<td>3.</td>
<td>The GIMP (Graphics Editing Tool)</td>
<td>Freeware</td>
</tr>
</tbody>
</table>

The developed virtual labs are currently being hosted on Google's free app platform, the Google App Engine (http://appengine.google.com). While this is presently free, it presents limitations which could prove vital for virtual labs. For example, the Google App Engine would make it difficult to have a service broker which handles authentication and a separate
database from the experiment engine's database hence making authentication and user monitoring quite impossible.

**Challenges**

Although it is strongly desired to depict reality as much as possible, limitations arise when trying to render 3D actions in 2D. As optimization is considered in terms of memory usage, animations implemented in design also have to be minimized to an extent.

Construct 2 which is a new tool that is experiencing development every time with the release of different versions requires the developer to update his work sometimes. This is the case when some new updates in the software provides a means for the developer to produce a better version of some actions which have already been inefficiently implemented hence, a need to go back and rework some of the implementation.

Another major challenge is the support for multiple screen sizes. Although the HTML5-based laboratories can work on various devices another limitation is the multiple screen sizes. It is a challenge to design images that will likely fit into screens when different platforms are used.

**Future Works**

The work is an ongoing project thus the pH measurement and Ohms law lab are yet to be developed. Also work is still being done to handle compatibility on various screen sizes. The authors focused on the development of laboratories for lower level Science and Engineering subjects as these are the subjects in which there are the largest number of students and hence greatest shortage of equipment to handle the class sizes. Pedagogic success with these experiments will serve as a pedestal to obtain funding for the advanced experiments for higher level Science and Engineering subjects. Some of the authors have also previously designed Re-crystallization and Melting Point Experiments, Electrical Conductivity Measurements and Chloride measurement Experiments and these are currently being hosted on the UK Open Science Lab site www.opensciencelab.ac.uk. These experiments can also be found at http://62.173.43.104/RMExp/RMExp.html, http://62.173.43.104/ConductivityMeasurement/ConductivityMeasurement.html, and http://62.173.43.104/ChlorideExp/ChlorideMeasurement.html

The Ohms Law experiment had been initially designed using LabVIEW but it would require installation of LabVIEW, hence the need for a new design using graphics. The issue of efficient cross platform usage will also be considered. It is to be noted however that the experiments developed for the UK Open Science Lab site were developed using Adobe Flex and Adobe Flash.

**Conclusion**

The experiments developed by this project are part of the introductory experiments used in the teaching of chemistry and physics in lower level classes in universities and also the
science classes in high schools. The project would therefore serve the purposes expected by a virtual laboratory in terms of making it easier for the students to relate to the experiment when seen in the real laboratories. They would also provide the means for students to revise the steps involved in the experiments when reading or preparing for examinations in cases where they cannot go back to the laboratory.

This work has also helped to bring virtual experiment a step closer to the users, by being able to use the laboratories without expecting the user to make any pre-installations.

References


Developing Control Experiments as a part of a Remote Laboratory Facility

Dr. Abul K. M. Azad, Northern Illinois University

Abul K. M. Azad is a Professor with the Technology Department of Northern Illinois University. He has a Ph.D. in Control and Systems Engineering and M.Sc. and B.Sc. in Electronics Engineering. He has been in academics for 15+ years, and his research interests include remote laboratories, mechatronic systems, mobile robotics, and educational research. In these areas, Dr. Azad has over 100 refereed journal and conference papers, edited books, and book chapters. So far, he has attracted around $1.7M of research and development grants from various national and international funding agencies. He is a member of the editorial board for a number of professional journals as well as an Editor-in-Chief of the International Journal of Online Engineering. He is active with various professional organizations (IEEE, IET, ASEE, and ISA) as well as a member of board of Trustees of CLAWAR Association. He has served as Chair and Co-Chairs of numerous conferences and workshops, in addition to serving on the program committees of around 30 international conferences. Dr. Azad is a project proposal reviewer with various national and international funding agencies in US, Europe, and Australia.

Pramod P Kaushik, Northern Illinois University

Pramod Kaushik, a graduate from Northern Illinois University specialized in robotics and VLSI. He had great passion towards robotics and was dedicating his time to implement robotic projects while at school. With bachelors in electronics and communication engineering from Anna University, India, Pramod had a strong background in electrical engineering. The ability to learn new tools and software quickly and to use them in projects made him an idea candidate for research assistant under Dr Abul Azad. Under the able guidance of Dr Azad, Pramod designed and implemented various projects in robotics and among them, the most notable work is "Developing Control Experiments as a part of Remote Laboratory Facility".
Developing Control Experiments as a part of a Remote Laboratory Facility

Abstract

This paper presents the customization of engineering systems for their integration with a remote laboratory web portal. The author starts with a brief discussion on remote laboratories along with the customization requirements of equipment for their use within a remote laboratory. The paper then illustrates the hardware and software customization of two engineering systems, a mobile robot, and a flexible robot manipulator. Finally, it discusses the structure of a remote laboratory portal along with some of its operational details.

1. Introduction

Smart devices are growing exponentially, and our everyday life has changed dramatically with the advent of the Internet and networking technologies. Related to these technologies, one emerging entity is the IoT (Internet of Things). This is a developing concept of making an open network of devices equipped with sensors and RFIDs (radio frequency identification) aimed at interconnecting all things electronic to make them more intelligent and programmable. According to Cisco, a leader in IoT and supported by IEEE, about 20 billion machines and devices could be linked by 2020.1 Smart devices are already being used for various remote activities, such as to check soil moisture in vineyards, control home environments, alert drivers to traffic jams, and monitor patients’ vital signs—all without human intervention. However, users will have a major role to play as they generate and use the data coming from the myriad of devices. This IoT will offer plenty of opportunities but also challenges to building ever more complex systems, dealing with lack of standards, analyzing and managing the data, and securing privacy.2 In academic areas, IoT is appearing in the form of remote laboratories where physical (real) laboratory equipment is being controlled remotely over the Internet.3,4 This paper will discuss some of the issues related to the remote laboratories.

Remote laboratories are gaining popularity among researchers and educators, and there are a number of reported initiatives in terms of system design, technology use, and pedagogical issues. These laboratories have great potential and can bring a new dimension for teaching the STEM (Science, Technology, Engineering, and Mathematics) disciplines.5,6 However, the integration of a number complex technologies and the current development structure of remote laboratories have made it difficult to develop and obtain sustainability.7

As a continuation of Internet accessible remote laboratory facility development, the lead author recently integrated a couple of control system experiments into the facility. This paper presents the design and development of two control experiments and their integration process into the remote laboratory facility. The pedagogical design details are provided in a separate paper, which is submitted to the ASEE 2014 annual conference. Section 2 illustrates the importance of remote laboratories, followed by a section describing the essential components of a remote laboratory. Section 4 presents the customization process of two engineering systems for their integration with a remote laboratory. Section 5 shows the makeup of a remote laboratory web
portal that can be used for a remote laboratory course delivery and is followed by the conclusion section.

2. **Why Remote Laboratories?**

Traditional laboratory classes are scheduled for only a limited time period. Considering the mixed ability level of the students, they sometimes want or feel the need to perform additional experiments beyond their assigned tasks.\textsuperscript{8} It is usually difficult to accommodate any extra time due to the lack of resources to keep the laboratories open, but ironically, too much experiment equipment lies idle during most of its usable lifetime. Remote experimentation facilities can provide cost effective and unlimited access to experiments and maximize utilization of available resources.\textsuperscript{9} Moreover, these facilities will allow inter-experiment collaboration among universities and research centers by providing research and student groups with access to a wide collection of expensive experimental resources at geographically distant locations.

The development process for an Internet accessible remote laboratory is complex, as it involves multiple disciplines: software, hardware, computer interfacing, Web development, Web security, user interface, and learning management.\textsuperscript{10, 11} However, a number of research groups have developed remote laboratories (and some are in the process of doing so) and have demonstrated their potential in the fields of teaching, research, and industry.\textsuperscript{12, 23, 14}

3. **Components of Remote Laboratories**

The basic purpose of a remote laboratory is to make equipment available over the Internet so it can be manipulated remotely to perform experiments needed by a student or a researcher.\textsuperscript{15, 16} There are a few simple steps to implement a remote laboratory (Figure 1):

![Figure 1: Basic concept of remote laboratories.](image)

- a) The first and foremost thing is that the equipment should have interfaceability with a computer (or with a networked device) along with the ability to exchange its input(s) and output(s) as needed to perform experiments.
- b) The next required item is a local computer that will provide the processing requirement for an experiment along with hosting a graphical user interface (GUI). The GUI will allow a remote user to perform experiments using the local computer without any physical intervention within the laboratory in which the equipment is located.
c) The third step is connecting the local computer with the Internet, allowing the GUI to be accessed from remote locations using an Internet browser.

In addition, access control to the experiment needs to be considered to ensure safety and proper use of the facility, pedagogical design issues, and management of the facility when offering multiple experiments at the same time. To address the later issues it is essential to consider developing or using a web portal that is suitable for a remote laboratory scenario.\textsuperscript{17}

4. Customizing Control Experiments

Almost all commercially available laboratory equipment is designed with an idea that it is to be used within a laboratory when students/researchers are sitting nearby. As discussed earlier, the remote laboratory concept introduces a totally different mindset in which students/researchers would be remotely located and would not have any access to the laboratory in which the equipment is residing. This makes it essential to customize commercially available equipment before it can be integrated with a remote laboratory facility. The customization involves the processing of all input(s) and output(s) needed for performing experiments and makes it available to the local computer connected to the equipment. The paper reports the customization of two pieces of equipment used for remote laboratories. The equipment is a household mobile robot and a flexible robotic manipulator. The customization involves processing input(s), output(s) signals and interfacing the equipment with a host computer, and finally developing suitable GUIs. Safety is a key factor for remote laboratory offerings, and a sub-section is provided to describe how that issue has been dealt for these experiments.

4.1 Roomba Create- Household Mobile Robot

The RoCr (Roomba Create) is a complete robot development system that allows one to program new robot behaviors without thinking much about the mechanical assembly and low-level code. The RoCr system was developed by iRobot. RoCr’s Open Interface (OI) provides a set of commands, such as ‘drive’ commands, demo commands, song commands, and sensor commands.\textsuperscript{18}

![Image of a Roomba Create.](image-url)
Through a serial communication, these commands can be used to develop new behaviors, add third party electronics, and write OI based programs to control the RoCr. An image of the RoCr is provided in Figure 2. In addition to the inbuilt sensors, one can attach additional hardware (sensors and actuators) to the RoCr via a cargo bay connector. These can be a robotic arm, light sensor, ranging sensor, light display, etc.

**RoCr Open Interface**

The OI consists of an electronic interface and a software interface for controlling the RoCr’s behavior and reading its sensors. The hardware interface includes a 7 pin Mini-DIN connector for connecting the RoCr with a PC and a DB-25 connector in the cargo bay area. The software interface allows a user to read its sensor and manipulate the RoCr’s behavior through a series of built in commands. The commands include the mode, drive motors, song, demo, and sensor status request.

The RoCr OI has four operating modes: Off, Passive, Safe, and Full. Once it receives the Start command, the RoCr can then enter into one of the four operating modes by sending a mode command to the OI. It can switch between operating modes at any time by sending an appropriate command to the OI. Depending on the requested packet ID, the RoCr sends back one or all 43 different sensor data in the form of packets. The sensing items are the bumps and wheel drops, wall, cliff left, cliff front left, cliff front right, cliff right, virtual wall, over currents, IR byte, buttons, distance, angle, charging state, voltage, current, battery temperature, battery charge, battery capacity, wall signal, cliff left signal, cliff front right signal, cliff right signal, user digital input, user analog input, charging sources available, OI mode, song number, song playing, number of stream packets, velocity, radius, right velocity, and left velocity.

**Modified Roomba Create**

The RoCr comes with a setup for its stand alone operation and needs to connect with a computer via a cable for any changes to its program. For its navigation, the RoCr is fitted with bump sensors. The customization task involves the wireless connection of the RoCr with a computer using a Bluetooth link, attaching a number of IR (infrared) sensors for obstacle sensing before hitting an object, and attaching a wireless camera so remote users can get a view of the RoCr’s navigating path. A block diagram of the modified RoCr is shown in Figure 3; while an image of the modified system is shown in Figure 4.
Bluetooth Interface

Bluetooth technology is used for creating wireless personal area networks (PANs). Unlike Wi-Fi networks, which can have hundreds of users, PANs were designed to be used by a single user. For this project, a BlueSMiRF was used for Bluetooth implementation. BlueSMiRF is a relatively inexpensive Bluetooth modem that implements the Bluetooth Serial Port Profile (SPP) and presents a normal 5V logic set of serial lines. When the BlueSMiRF is paired with the host computer, the serial lines are virtually connected to the computer as a normal serial port.
**Microcontroller based Infrared Obstacle Detection**

In its original form, the RoCr uses three micro-switches to sense obstacles. These switches are mounted behind a spring-loaded bumper located at the front and sides of the RoCr. To allow the RoCr to detect an obstacle ahead of time and to avoid any contact with an obstacle, the RoCr has been modified by mounting four Infrared Sensors (IR). IR are used in this project to detect obstacles, and four IR sensors are mounted in different locations. Three of the sensors are in the front and two sides of the RoCr and one on the back. The sensor outputs are connected to a microcontroller system via A/D (analog to digital) converters. The microcontroller prepares the digitized sensor outputs in a serial format before passing these via an RF transmitter. A receiver unit is placed around the host computer where the data are collected by another microcontroller system and pass the collected data directly to the host computer. The schematic of the receiver part of the IR data communication is shown in Figure 5.

In terms of its operation, IR sensors send out IR signals and receive them if they bounce back. The strength of received IR signal corresponds to the distance between the sensor and the obstacle. The IR sensors are from Sharp and is a combination of PSD (position sensitive detector), IRED (infrared emitting diode), and a signal processing circuit.\(^1\) The sensor outputs are connected to an onboard microcontroller system via A/D converters (MCP3202). The MCP3202 is a successive approximation 12-bit A/D converter with on-board sample and hold circuitry and is programmable to provide a single pseudo-differential input pair or dual single-ended inputs.\(^2\)

A Propeller microcontroller is used for this project, and it is designed to provide high-speed processing for embedded systems while maintaining low current consumption and a small
physical footprint. In addition to being fast, the Propeller provides flexibility and power through its eight processors, called cogs, that can perform simultaneous independent or cooperative tasks. The Propeller can be programmed using ‘spin language,’ a high-level object-based language developed by the Parallax. The spin provides control of the Propeller’s multi-core hardware and encourages the principles of the Propeller’s real-time application design in ways that were not represented by existing languages. The spin can execute up to 160 MIPS (million instructions per second), i.e. 20 MIPS per cog.

**Real-time Video Streaming**

One of the features of the modified RoCr is to have an onboard camera to provide a real-time video of its surroundings. A miniature wireless camera is mounted on top of the RoCr that can provide a front view from the robot. The camera uses 2.4GHz bandwidth for data transmission, and the signal is received by a base unit that converts images to PAL/CCIR/NTSC/EIA standards. The Radio AV receiver receives the transmitted signal and sends it through composite cables. The received video is then converted to digital format and passed to the computer using a Video Xpress. A block diagram shows the implementation of the wireless camera (Figure 6).

![Figure 6: Shows a block diagram for wireless camera system.](image)

**Graphical User Interface**

A GUI is one of the most important components of a remote laboratory facility. In this development, LabVIEW is the main driving force for GUI design and development. The sensors’ data collected by the host computer via the Bluetooth are passed to LabVIEW. The digitized form of video is also collected by the LabVIEW for display within the GUI. The LabVIEW design involves a connection with the Bluetooth, collecting data from the microcontroller receiver unit, and collecting the digitized video as transmitted from the RoCr. Once the data are collected, the task is to manipulate them to implement control strategies as well as display real time video through a GUI. The communication interface includes configuring the Bluetooth, by setting up the baud rate and serial port. Once communication is established between the RoCr and the host PC, communication between LabVIEW and the RoCr can be initiated. Figure 7 shows a screen shot of the developed GUI that controls the RoCr. As shown, the GUI includes all the desired control functionalities as well as sensor status and video.
The top part of the GUI provides the initialization and port setup process. The left bottom part of the GUI shows the speed and turning radius adjustment markers. The remaining area is divided for the RoCr command and control buttons, displaying video, and showing the RoCr sensor status. The available commands within the GUI are move forward, move backward, turn right, turn left, stop, load and play tunes, and change speed and turning radius. The sensor data used for making decisions (in the back end of GUI) are right bump sensor, left bump sensor, override current sensor, and IR sensors. Apart from the last one, the others are inbuilt sensors of the RoCr. In addition, the GUI displays a few other sensor statuses: home detect sensor, wheel drop sensors, battery temperature sensor, and battery charge sensor.

**Control Strategies**

The main goal of controlling the modified RoCr is to avoid any obstacle in its way and to develop an alternative path when necessary. Initially the controller will drive the RoCr in a straight-line path and check the IR obstacle sensors as well as the bounce sensors. Upon receiving an obstacle signal, the controller will automatically redirect the RoCr in a revised course. In addition to automatic control, the RoCr can also be controlled manually via a GUI, where the manual control has a higher priority than the automatic control.

Within the automatic control approach, there are three control strategies for handling three different obstacle scenarios. The scenarios are obstacle on the right, obstacle on the left, and obstacle in the front. This paper explains only one of the scenarios, obstacle on the left.
the left IR sensor detects the presence of an obstacle, the controller (within the host computer) sends a signal to the RoCr (via Bluetooth) and turns toward the right with a given turning radius for a period of time and then drives in a straight path. The turning radius and speed of the RoCr can be adjusted by a user via the GUI. Figure 8 depicts the movements for this scenario.

Figure 8: Motions after detecting an obstacle in the left side of the RoCr.

Figure 9: System block diagram for the modified flexible manipulator system.

4.2 Flexible Manipulator System

The flexible manipulator system was acquired from Quanser Inc., a company that markets educational experiment systems; most of their products are in the control discipline. A block diagram of the customized system is shown in Figure 9. The flexible manipulator system is a single-link manipulator that can move only on the horizontal plane. The manipulator link is approximately 17 inches long, 0.82 inches high, and 0.034 inches thick. At the hub, one end of the link is connected to a motor for rotation on the horizontal plane.
The motor is connected to a motor driver. Four sensors are connected to the system to collect vibration and motion information: encoder for hub angle, technometer for hub velocity, strain gauge close to the hub area for strain information, and accelerometer for end-point acceleration. The collected sensor signals are passed through a signal conditioning system for the required amplification and noise filtering. All these sensor outputs and actuator inputs are connected to an Input/Output (I/O) breakup board and then to a networked computer via a DAQ board. The DAQ board provides required interfacing of the signals to the computer using appropriate digital to analog and analog to digital converters. The DAQ board is an NI USB-6211 acquired from National Instruments and provides 16 analog inputs and 2 analog outputs, (both with 16-bit accuracy) along with a number of digital inputs and outputs. The host
computer has LabVIEW software and is used for both the controller implementation and GUI development.

Figure 10 shows an image of a GUI that is used by students to observe and validate the vibration properties of a flexible manipulator system over the Internet. The upper part of the GUI shows the control points where a student can start and stop the experimental system and choose an input signal to the system as well as control the properties of the input signal. Just below (the left hand side) is a graph window showing the shape of the desired input signal, while the right hand side window shows the remotely located experimental system. The bottom part of the window shows four pairs of sensor signals that are collected from the experimental system (both in time and frequency domains). The right hand side ones are in the time domain, and left hand side ones are in frequency domain. In addition, one can also download all the input and output signals on the host computer for further processing.

4.3 Safety Issues

Care must be taken while operating an electro-mechanical system from a remote location, especially when the remote location can be potentially unmanned. Improper use of the system can lead to equipment breakdown, damage to the system and surroundings, even the trigger of a fire. Several measures were taken to address these issues. This involves physical measures to protect the systems and limiting the control actions that a remote operator/user can perform. The RoCr is usually placed within a confined location where it can perform its activities without any serious damage to the system or its surroundings. Even while working outside the area it can protect itself from hazardous activities, which is ensured via its programming. The limitation provided via a program includes the setting of a maximum allowed speed limit for RoCr. This is to restrict impact on a collision. In case of a motor overdrive, the RoCr will turn itself down automatically. The second scenario can happen if someone directs an RoCr to an obstacle for longer time. The flexible manipulator is also protected physically and programatically. In terms of physical measure, limit switches are placed on two sides so that the flexible link can rotate only for 180 degrees, while the system control program limits the maximum amount of torque and duration one can apply, hence limiting the speed of the system.

5. Remote Laboratory Portal

Offering laboratories over the Internet involves a number of experiments being delivered at the same time to a group of student. In addition, the experiments should change with time as the course progresses. There should also be a provision for maintaining the safety of the laboratory in terms of operational hazard and integrity and to accommodate activities that will allow one to achieve pedagogical objectives. All these require a suitable portal that will provide password controlled access to the facility, activating and deactivating experiments with time, tracking student activities for pedagogical issues, and ensuring safety and system integrity.

A system diagram showing how the laboratory experiments are connected to the Internet via a web portal is provided in Figure 11. Within the diagram, the ‘Laboratory Management Server’ hosts the web portal that provides all the web applications that manage and control remote access to the facility and manage all of the laboratory-related activities. Equipment needed to perform
experiments are gathered within the ‘Equipment Bank’ where they are connected with a
Switching Matrix’. A ‘Switching Matrix’ allows one to change an experiment configuration
(hardware connection) via a ‘Host Computer’. A ‘Host Computer’ holds the GUIs along with a
provision for the video and audio feed of the experimental system. Along with local access, the
GUIs (with video and audio) are available over the Internet as a dynamic web page. As
mentioned earlier, access to the GUIs is managed by the ‘Laboratory Management Server’. A
user can access the GUIs to perform experiments from remote locations via the server.

Figure 1: Block diagram of a generic remote laboratory system.

An image of the web application page that is available via the ‘Laboratory Management Server’
is shown in Figure 12. This application page is developed by using .NET technology connected
to an SQL Server. The page provides a portal to the remote laboratory facility while facilitating
all of the requirements mentioned earlier.

Figure 12: Shows the image of administrator home page.
6. Conclusions

The paper provides an illustration of customizing two control experiments for integration within a remote laboratory facility. This involves preparing inputs and outputs for hardware interfacing with a local computer, software design, development of GUI, and finally their integration with a remote laboratory facility. The process needed additional hardware and software along with the provision of audio and video feed of the experiment environment. The developed GUIs provide an environment in which a remote user can control the systems without any physical presence within the host laboratory. After the integration of the experiments, the remote laboratory portal ensures the safety and integrity of the experiments. In addition, it provides a provision of experiment management and pedagogical design tools.

Acknowledgement

The authors would like to thank the National Science foundation for its support for the reported work. This paper is based on a NSF TUES (Transforming Undergraduate Education in Science, Technology, Engineering, and Mathematics) project, award number DUE-1140502. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References


The BlueSMiRF Silver is the latest Bluetooth wireless serial cable replacement from SparkFun Electronics. Website: https://www.sparkfun.com/products/10269

SparkFun is an online retail store that sells the bits and pieces to make any electronics projects possible. Website: https://www.sparkfun.com/

GP2Y0A21YK0F Distance Measuring Sensor Unit Measuring distance: 10 to 80 cm Analog output type. Website: http://www.parallax.com/tabid/768/ProductID/776/Default.aspx . Other product related information can be found at http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf

MCP3202 2.7V Dual Channel 12-Bit A/D Converter with SPI Serial Interface. Website : http://www.parallax.com/Store/Components/AllIntegratedCiRoCuits/tabid/154/CategoryID/31/List/0/Sort Field/0/Level/a/ProductID/232/Default.aspx and the document containing more specific details about the ADC is given in the website : http://www.parallax.com/Portals/0/Downloads/docs/prod/appkit/ltc1298.pdf


Wireless Surveillance Camera system, which was used in this project, can be found here. http://www.intelspy.com/4ca2wimicoau.html.

Video Xpress is a converter that can get analog signals from the composite cable to digital signal through USB. This product is developed by ADS Technologies Inc. and this product can be bought at http://www.tigerdirect.com/applications/SeaRoCrhTools/item-details.asp?EdpNo=3428849
