An Integrative Approach to Embedded Systems Courseware
Submission Type: Work in Progress

Harry Powell, Todd DeLong, and Homa Alemzadeh
University of Virginia

Abstract
In 2012 a course, *Introduction to Embedded Computing*, in Electrical and Computer Engineering was introduced at the University of Virginia. This course was taught in a studio style environment and distinguished itself by reinforcing concepts from across multiple areas of electrical and computer engineering, i.e., filtering simple controls, and motors, as key elements of our embedded computing laboratory sequence; a very simple CPU was employed. In 2017 an upper-level elective course in *Advanced Embedded Computing* was introduced. In this class, students work with a more capable ARM platform and implement a basic RTOS kernel with thread management, synchronization, and real-time scheduling features. In this work-in-progress paper, we discuss current course offerings and a new bridge course under development that explores introductory RTOS concepts and links embedded computing to advanced ECE courses at the second- and third-year levels.

Keywords
Embedded Computing, Embedded Systems Programming, RTOS, Electrical Engineering Concepts

Introduction to Embedded Computing
Introduction to Embedded Computing was introduced at the University of Virginia as a third-year replacement course for a much earlier course in computer architecture. As such this course was designed to introduce students to low-level embedded systems programming and interfacing while keeping the material and concepts at an introductory level. This required students to understand the basics of the underlying architecture while developing an understanding of how architecture and programming were related. Additionally, the laboratory work was designed to reinforce concepts from all aspects of an electrical and computer engineering undergraduate program.[1] The course has been taught in a studio-style which enhances the strengths of project-based learning while keeping elements of a formal lecture in place.[2]

An essential element of this course is project work based on hardware that has been designed with particular educational goals in mind. This hardware was created in-house, and each experiment is designed to introduce elements of embedded interfacing, embedded programming, and general electrical engineering concepts.

Advanced Embedded Computing
In 2017 we added an upper-level elective course in *Advanced Embedded Computing*, now in its second offering. This course is intended for first-year graduate (GRAD) and third/fourth-year undergraduate (UG) students in Electrical and Computer Engineering. It helps the students develop the foundational knowledge and skills to design, implement, and validate embedded real-time systems. The learning goals and objectives include being able to describe the principles of embedded system architectures and Real-Time Operating Systems (RTOS) and explain how

memory management, interrupt handling, thread management, task scheduling, and software/hardware interfacing are done in real-time operating systems.

Also, students should be able to design and evaluate an embedded system based on a given specification and validate if the functional requirements are satisfied. They should be able to develop C/C++ programs in conjunction with real-time operating systems. The combination of these topics is intended to allow the students to be able to relate to the real-world applications of embedded systems and associate it with emerging areas such as Cyber-Physical Systems (CPS), Internet-of-Things (IoT), and robotics.

In this class, students work with a more capable ARM platform and implement a basic RTOS kernel with thread management, synchronization, and real-time scheduling features. They finally design a handheld computer game with several tasks (foreground and background threads (ISRs)) implemented on top of the RTOS for sampling data from ADC converters, processing the data, and displaying results on LCD. Throughout this final game design experience, they learn how to deal with concerns such as FIFO communication between threads with different speeds, mutual exclusive access to shared resources using semaphores, deadlock, priority inversion, and other task management issues. In the last part of the class, the students are exposed to the more advanced topics and research trends in embedded systems, such a commonly used RTOS, the design of embedded systems for cyber-physical systems and Internet of Things applications, and safety and security issues of embedded platforms.

Hands-on embedded C programming projects are the main component of this class and contribute to 50% of the final grade. There are five mini-projects in which the students apply the concepts learned in the lectures to design and implementation of an RTOS on the TI TM4C123 micro-controller:

Mini Project 0 - Getting familiar with ARM Programming Environment

Mini Project 1 - Embedded I/O: Graphics LCD and Periodic Timer Interrupts

Mini Project 2 - Basic RTOS Kernel: TCB Management, Basic Scheduling, Timing Features

Mini Project 3 - RTOS Kernel: Spinlock Semaphores, Sleep Functionality

Mini Project 4 - RTOS with Blocking Semaphores and Priority Scheduling

For the final project, students work in teams of 3-4 students on the design of the hand-held computer game on top of their RTOS. There they extend their basic game functionality with innovative designs and features. At the end of the semester, each team presents a demo of their system to the class, and the top three designs will be selected by the TAs and students.

Intermediate Embedded Computing

In the first several offerings of *Advanced Embedded Computi*ng, several crucial gaps in understanding have been observed. Students perceive a rather sizeable conceptual leap from working with the "bare metal" environment of the introductory course to managing tasks in a real-time operating system environment. This has led us to work on the development of an intermediate course that would prepare students for a smoother entry to the advanced course as well as help them prepare for the fourth year *Capstone* experience.

To accomplish the creation of this "transition" course, the authors are going to capitalize on one of the more unique aspects of the introductory course - that to a great extent it is not a stand-alone course. That is, students are required to use the knowledge and skills learned in other courses (like *Fundamentals of Electrical Engineering 1*) to satisfy project requirements in the embedded course. For example, one project requires the students to design a Finite State Machine (FSM), implemented in software, which produces a linear triangle signal. The triangle signal is then modulated using Pulse-Width Modulation to vary the intensity of an LED. To verify that they have correctly implemented the triangle signal, they are required to design and implement an RC low-pass filter (which they learn in *Fundamentals 1*) to filter the relatively high-frequency modulation and recover the low-frequency modulated triangle signal.

In the intermediate course, this philosophy will be continued, establishing connections to courses like *Fundamentals of Electrical Engineering 2* and *3*.[3] For example, for the final project in *Fundamentals 2*, students design and implement two 2^{nd}-order Sallen-Key filters that are used to filter the input (in this case, their own music) into high- and low-frequency components. Each component is then used to vary the intensity of an LED, thus effectively producing a simple graphic equalization of the input signal. In the intermediate embedded course, students will re-visit the project, but will instead consider how the project requirements could be satisfied with a microcontroller-based system (versus the analog hardware in *Fundamentals 2*). We believe that investigating such trade-offs could prove to be the more interesting aspect of the project and better prepare them for *Capstone* and beyond.

Likewise, the students will be asked to revisit the final project in *Fundamentals 3*, where they design and implement an analog EKG system. In the intermediate course, they will be presented with the problem of processing the output from the EKG system (for example, to determine the heart rate) as well as designing and implementing an appropriate user interface. Students will be introduced to the need for being able to handle multiple tasks concurrently, thus laying the groundwork for a more formalized method of handling multi-tasking using an RTOS which will be discussed in the *Advanced Embedded Computi*ng class.

Our learning objectives for this course will be a.) gaining an understanding of the processes involved in transitioning a design from an analog realization to a microcontroller solution, b.) understanding structured approaches to multitasking, and c.) understanding the issues in designing a responsive user interface on an embedded computing platform.

Summary and Conclusions

We view *Embedded Computing* as a central element of our curriculum and endeavor to teach it as an integrative collection of concepts spanning the entire discipline of electrical and computer engineering, i.e., the CPU is a component that is viewed as part of a more extensive system, and not simply in isolation. Our development of both introductory and advanced level *Embedded Computing* courses following this philosophy has created a gap in student comprehension. We are developing an intermediate level course that is intended to prepare students for the upper-level course. The goals of this transitional course are the introduction of basic real-time operating system concepts within the context of applications that reinforce ideas from across the discipline. We also believe that reproducing analog projects with a digital approach not only builds embedded systems design skills, but reinforces the duality between the two techniques.

References

1. Harry C. Powell, Joanne Bechta Dugan. Embedded and Cyber-physical Computing Courseware as the Nexus of Curriculum Reform in Electrical and Computer Engineering. In: Proceedings of ASEE- SE Section. Macon Ga.; 2014.

2. Dr. Harry Powell, Dr. Ronald Willians, Dr. Maite Brandt-Pearce, Dr. Robert Weikle. Towards a T Shaped Electrical and Computer Engineering Curriculum: a Vertical and Horizontally Integrated Laboratory/Lecture Approach. In: Proceedings of ASEE Annual Conference 2015. Seattle WA.; In publication.
3. Powell H, Brandt-Pearce M, Williams R, Weikle R, Harriott L. Incorporating Studio Techniques with a Breadth-First Approach in Electrical and Computer Engineering Education. In ASEE Conferences; 2016 [cited 2016 Nov 4]. Available from: http://peer.asee.org/25661

Biographies

Harry C. Powell

Dr. Powell is an Associate Professor of Electrical and Computer Engineering in the Charles L. Brown Department of Electrical and Computer Engineering at the University of Virginia. After receiving a Bachelor's Degree in Electrical Engineering in1978 he was an active research and design engineer, focusing on automation, embedded systems, remote control, and electronic/mechanical co-design techniques, holding 16 patents in these areas. Returning to academia, he earned a Ph.D. in Electrical and Computer Engineering in 2011 at the University of Virginia. His current research interests include machine learning, embedded systems, electrical power systems, and engineering education. Dr. Powell is a member of ASEE, IEEE, and ACM.

Todd A. DeLong

Dr. Todd DeLong has been serving as a teaching faculty for over ten years, teaching courses in electrical and computer engineering, mathematics, and physics. Prior to teaching, Dr. DeLong served as a research scientist at the University of Virginia, performing research in the area of theValidation & Verification (V&V) of safety-critical systems, for transportation (rail and transit), nuclear, and implantable medical devices, working with government agencies (FRA, NRC, FDA) as well as industry suppliers. Dr. DeLong is a member of the ASEE and a senior member of the IEEE.

Homa Alemzadeh

Dr. Homa Alemzadeh is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Virginia. Homa is also an Assistant Professor by courtesy in the Computer Science Department. Before joining UVA, she was a research staff member at the IBM T. J. Watson Research Center. She received her Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign and her B.Sc. and M.Sc. degrees in Computer Engineering from the University of Tehran. Homa was the recipient of 2017 William C. Carter PhD Dissertation Award in Dependability from the IEEE Technical Committee and the IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance. Her research interests include dependable and secure computing, embedded and cyber-physical systems, and robotics. Dr. Alemzadeh is a member of ASEE and IEEE.