Real-time Embedded Systems
ECE 4501/6501, CS 6501
Fall 2020

Instructor: Homa Alemzadeh
Assistant Professor, ECE
Office: Olsson Hall (Link Lab) 259
Email: alemzadeh@virginia.edu

Teaching Assistants: TBD

Class Time: Tuesdays and Thursdays, 11:00 AM - 12:15 PM
Location: All the course activities will be carried out online.
Class meetings and office hours will be done via Zoom.
Recordings of lectures will be made available for asynchronous learning.

Office Hours: Online via Zoom - Check course website and UVA Collab

TA Office Hours: Online via Zoom - Check course website and UVA Collab

Course Website: http://faculty.virginia.edu/alemzadeh/teaching/ece_6501/index.html

Primary Textbook:
Jonathan W. Valvano, *Embedded Systems: Real-Time Operating Systems for ARM® Cortex™-M Microcontrollers*, Volume 3, Fourth edition, January 2017, ISBN: 978-1466468863.
Outline: http://www.ece.utexas.edu/~valvano/arm/outline3.htm

Additional References:
- Edward Lee and Sanjit Seshia, *Introduction to embedded systems: A cyber-physical systems approach*, MIT Press, 2016. (**Free PDF:** http://leeseshia.org/download.html)
- Philip Koopman, *Better embedded system software,* Drumnadrochit Education, 2010.

Course Description
Embedded systems are special-purpose computers (e.g., micro-controllers, DSP processors) at the core of Cyber-Physical Systems (CPS) that monitor and control the physical processes through real-time interactions with sensors and actuators. They are tightly integrated with the electronic and physical components and must operate within real-time performance, battery, and size constraints. More than 90% of manufactured micro-processors go inside airplanes, automobiles, medical devices, digital cameras, toys, home appliances, and smart buildings. What are the building blocks of an embedded system? How can we design an embedded system and make sure it satisfies specific functionality, reliability, and timing requirements? How can we bridge the gap between the inherently sequential embedded software with the intrinsic concurrency in the physical world? How can we execute multiple data acquisition, processing, and control tasks on resource-constrained microcontrollers while satisfying real-time constraints?

This course will help you answer these questions by providing the foundational knowledge and hands-on experience in design and validation of embedded computing systems, with a focus on embedded C programming and real-time operating systems for ARM® Cortex™-M Microcontrollers. In the second half of the class, we will explore related topics and applications in safety and security, cyber-physical systems (CPS), internet of things (IoT), and robotics through paper presentations and discussions.

Learning Goals and Objectives

This course is intended for the first-year graduate (GRAD) and the third/fourth-year undergraduate (UG) students in Electrical and Computer Engineering and Computer Science as well as the first-year graduate students in the NRT CPS program. It will help you develop the foundational knowledge and technical skills to design, implement, and validate real-time embedded systems. By the end of the course, you will be able to:

1. Describe the design principles for real-time systems and the key building blocks of embedded system architectures and real-time operating systems.
2. Explain how memory management, software/hardware interfacing, interrupt handling, multitasking, thread management and communication, and task scheduling are done in real-time operating systems.
3. Develop C/C++ programs within the ARM embedded programming environments.
4. Design and evaluate an embedded system based on a given specification and validate if the functional and timing requirements are satisfied.
5. Apply acquired knowledge and skills in the class to design and implementation of embedded systems in collaboration with your team members.
6. Relate to the real-world applications of RTOS and associate with related and emerging research areas, such as safety and security of embedded systems, Cyber-Physical Systems (CPS), Internet-of-Things (IoT), and robotics.
7. Construct new knowledge, critique ideas, and effectively present advanced topics.

Assessment and Evaluation

	Undergraduate (ECE 4501)	**Graduate (ECE 6501/CPS 2)**
Class Participation/Activity	10%	10%
Homework	20%	10%
Mini Projects	30%	30%
Midterm Exam	20%	20%
Final Project	20%	20%
Grad Student Presentations	0%	10%

Class Participation (10%): Active class participation is key in enhancing your team-work and critical thinking skills and improving the learning environment. Every week we will have an *in*-class activity in which you work in groups of two/three on solving problems related to the concepts taught in lectures or discuss topics presented in class (Objectives 1, 2, and 5). In-class activities are not graded, but will be followed by problems that you will complete individually in the homework. These activities will prepare you for mini-projects and final project, in which you apply the concepts learned in class to design of real embedded systems. Other important components of class participation are course evaluations via online anonymous surveys and peer evaluations by providing comments/feedback to your classmates (Objective 7).

Homework (UG: 20%, GRAD: 10%): Homework is to reinforce the concepts taught in the lectures and practiced in class activity through individual work. The main purpose of homework is to enhance your understanding of core concepts, encourage critical thinking, and to prepare you for the mini projects through small programming or written exercises (Objectives 1 and 2). Homework will be given roughly every week and will constitute a lower percentage of the final grade for the graduate students.

Mini Projects (30%): Mini-projects are hands-on experience programming exercises that enhance your technical skills and help you learn how to apply the concepts learned in the lectures, class activities, and homework into practice (Objectives 3, 4, and 5). There will be roughly three to four mini projects and they will be done individually. Every other week, after you submitted your mini-projects, we will discuss them in the class and you will evaluate your classmate's submissions based on the given rubric (Objective 7). You will then have a chance to review and revise your submission.

Midterm Exam (20%): Midterm exam will be based on the lectures, class activities, homework, and mini projects to test the material covered in the class (Objectives 1-4). Although the exam will be closed book, you will have access to the relevant datasheets and functional descriptions.

Final Project (20%): Building up on the knowledge and skills learned through lectures, class activities, and mini projects, you will work in teams of two/three on a project related to design and evaluation of an embedded system with a real-world application. Around the middle of the semester, each team will select a project and set the milestones, timeline, and division of tasks for completing their project. Each team will present their progress towards the completion of project and their final results through presentations in the class as well as a final written report. The projects will be evaluated based on the proposed ideas, the techniques used for design and validation, applying concepts learned in the class, demonstrating working prototypes and measurable outcomes, as well as the quality of final presentation and final report. All the students will be involved in the evaluation by giving feedback on the presentations by each team and grading it based on a given rubric. Each student will be also evaluated by me based on the quality and creativity of their ideas, questions, and evaluation. (Objectives 1-7)

Graduate Student Presentations (UG: 0%, GRAD: 10%): Graduate Student presentations will be done in the second half of the semester and are planned to expose you to the advanced topics and emerging research areas in embedded systems and enhance your critical thinking skills. To differentiate the requirements for the graduate class from those for the undergraduate class, the students enrolled in the graduate section will work with me on selecting a relevant topic of interest and will present it to the class. Each presentation will follow with a class activity or discussion on the topic. All of students are expected to participate in evaluating the presentations in terms of quality, content, and arguments using a given rubric. (Objectives 6 and 7)

Pre-requisites

A basic knowledge of computer architecture and embedded systems is required. A working knowledge of computer assembly and C/C++ programming is required for mini projects. Preferred undergraduate requirements include ECE 3430 and ECE 4435 or CS 3330.

Online Tools and Resources

Due to the COVID-19 pandemic, all the course activities will be carried out online. Class meetings (including lectures and in-class activities) and office hours will be done via ***Zoom***. Video recordings of lectures along with the lecture notes and related reading material will be posted on the ***UVA Collab website*** for asynchronous learning. We will use ***Piazza*** within the Collab website for posting announcements, running polls, and Q&A. **GitHub Classroom** will be used for posting and submitting mini projects and final project. Each student must actively follow the announcements and posts on the Collab and Piazza and show active participation by posting questions, answering questions posted by their classmates, and taking part in polls and online surveys in a timely manner.

Course Schedule

The following schedule presents the tentative activities for each week and throughout the semester. This schedule is subject to change depending on our progress and interests. I will inform you of any updates or changes on the Collab or Piazza and at the beginning of each class. http://faculty.virginia.edu/alemzadeh/teaching/ece_6501/schedule.html.

Course Policies

Academic Integrity: By enrolling in this course, you have agreed to abide by and uphold the Honor System of the University of Virginia (https://honor.virginia.edu). I expect every student in this course to fully comply with all of the provisions of the University's Honor Code. Any attempt to take credit for work done by another person is considered as plagiarism and a violation of the honor code. You are encouraged to study in groups and work together for in-class activities, some of the mini projects and the final project. You may also refer to online material, code, and research articles for your projects and presentations. However, you must properly credit your team members whom you collaborated with and cite any resources or individuals you consult or any code that you use. All suspected violations will be forwarded to the Honor Committee, and you may, at my discretion, receive an immediate zero on that assignment.

Late Policy: There will be a **10% penalty** for late assignments **(per school day).** You will also have **a grace period of three days** for the whole course to address any unexpected events such as sickness, traveling, other deadlines, interviews, etc. This means that you can always submit late assignments (except the presentation slides and the final project), but if you don't use more than three late days, you will not be penalized in any way. The late penalties and the grace period will be accounted at the end of the semester. Also, your lowest grades for homework and class activity will be dropped.

Disabilities: If you need any special assistance, please contact me **as soon as possible** by Email or during office hours. The University of Virginia strives to provide accessibility to all students. If you require an accommodation to fully access this course, please contact the Student Disability Access Center (SDAC) at (434) 243-5180 or sdac@virginia.edu. If you are unsure if you require an accommodation, or to learn more about SDAC services, you may contact them at the number above or by visiting their website at www.sdac.studenthealth.virginia.edu.