SWE 660: Software Engineering for Real-Time Embedded Systems

Spring 2025

Contact Information

Dr. Rob Pettit
Email: rpettit@gmu.edu
Phone: 703-993-6643
Office:
Engineering 4304
Book Time With Me
(Feel free to pop in anytime my door is open!)

Course Overview

This is a graduate course in Real-Time Embedded Software Engineering. This course will count as an advanced systems course for the MS CS degree or as a 600-level Software Engineering elective for the MS SWE degree.

The course will predominantly be taught from a hands-on, project-based approach. There will also be a research and presentation component but these will also be focused on practical applications vs. theory.

Learning Objectives

Upon completion of this course, students should have:

- An understanding of the scope and type of real-time and embedded (RTE) software-intensive systems
- A general understanding of the overall software development processes for RTE systems, focusing on an agile framework
- Specific, hands-on knowledge of programming and testing RTE software
- An understanding of real-time operating systems
- Advanced and emerging topics will be explored as time permits. Some examples include the use of RTE systems supporting the Internet of Things or deploying machine learning algorithms on edge devices.

Prerequisites

CS 531 or equivalent background in systems programming

This course assumes no prior knowledge of real-time or embedded systems. Knowledge of basic programming skills and data structures will be assumed. Programming assignments will be in C.

Course Materials

Required Text: Oshana, Robert, and Mark Kraeling. *Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications*. Saint Louis: Elsevier Science & Technology, 2019. (ISBN: 0128094486). GMU Library Link

Recommended Texts:

Kernighan, Brian W., and Dennis M. Ritchie. *The C Programming Language*. 2nd ed. Englewood Cliffs, N.J: Prentice Hall, 1988. (ISBN : 0131103628) GMU Library Link

Molloy, Derek. *Exploring BeagleBone : Tools and Techniques for Building with Embedded Linux*. Second edition. Indianapolis, IN: Wiley, 2019. GMU Library Link.

McLaughlin, Brian. The BeagleBone Black Primer. Hoboken: Pearson Education, Limited, 2019. GMU Library Link.

Required Hardware:

BeagleBone Black – I recommend the DigiKey or Mouser distributors.
You will also need a starter pack of electronic components to at least include breadboard, buttons, LEDs, jumper wires, and resistors. Something small like this will work and then you can add sensors, etc. as necessary for your term project.

Class Policies

Attendance:

You must be in attendance for the first class or risk being dropped from the class*. Additionally, there will be quizzes on most class days. There are no make-ups for these quizzes and you must be present and on-time to take the quiz. Quizzes are generally given in the first 15-minutes of class and I do not allow for late starts or time extensions.

*Per GMU Academic Policy (AP) 3.1

Campus Closure or Emergency Class Cancellation / Adjustment:

If the campus closes, or if a class meeting needs to be canceled or adjusted due to weather or other concern, students should check Blackboard [or other instruction as appropriate] for updates on how to continue learning and for information about any changes to events or assignments.

Disability Accommodations

Students with a learning disability or other condition (documented with GMU Office of Disability Services) that may impact academic performance should speak with me ASAP to discuss accommodations.

Email:

You must use your Mason email account for all email correspondence having anything to do with your work at Mason. Federal laws protecting your privacy rights require that we only communicate student information directly to students –and use of the university email system is our only way to validate your identity. You may forward your campus email elsewhere, but we can respond only to a Mason email account.

Grading:

Quizzes	15%
Projects / Programming Assignments	50%
Presentation / Discussion	10%
Programming Exam*	25%

Grading Schema:

A+ [98, max]	B [84, 88)
A [92, 98)	B- [80, 84)
A- [90, 92)	C [70, 80)
B+ [88, 90)	F [0, 70)

Note that there is NO rounding of grades

Quizzes	Most classes will have a short quiz to reinforce learning
Projects / Programming Assignments	We will have a series of programming assignments designed to give you experience with embedded systems development as well as real-time, concurrent programming. There will also be a larger term project. These can be completed individually or in groups of up to 5 students. You must remain in the same group for the duration of the semester, so pick carefully.
Presentation / Discussion	To broaden our understanding of modern RTE systems, you will research a practical application of RTE software and present to the class with a facilitated discussion. This will be done with your project group.
Programming Exam	Hands-on programming exam. *Note that there will be multiple dates for the programming exam, depending on class size. We will use the final exam block as one date and then one or two other sessions as needed. I will divide you into exam block groups within the 2nd week of class so that you will know your specific programming exam date. This will be a hands-on programming exam and be representative of what you've learned through the first three programming assignments. Specifically, you will be tested on: • Understanding your BeagleBone architecture and operating system • Programming embedded software in C, including multi-threading and using digital input / output on the BeagleBone • Reading C source code and integrating legacy code with your assigned programming task

Honor Code:

You are expected to abide by the University's honor code and the CS Department's Honor Code and Academic Integrity Policies during the semester. This policy is rigorously enforced. All class-related assignments are considered individual efforts unless explicitly expressed otherwise (in writing). Group assignments are to be completed only by that group – no sharing between groups. Exams and quizzes are strictly individual efforts. Using ChatGPT or any other automated or AI-based program to complete assignments is considered cheating in the same way as copying another student's work. Review the university honor code and present any questions regarding the policies to instructor.

Cheating on any assignment will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code.

The material provided in this course is proprietary. Uploading this material anywhere without the express permission of the instructor is strictly prohibited and a violation of the GMU Honor Code.

Missed or Late Work:

I do not accept late work nor offer make-up assignments. Late or missed work will receive a zero (0) for that graded item. Quizzes are generally given within the first 15 minutes of class. There will be no time extensions for the quizzes if you show up late for the start of class. As a practice, I do drop the one lowest quiz grade for the semester.

Participation:

For the group project and assignments, each group member is expected to contribute their fair share of effort to each assignment and project deliverable. At the end of the course, each team member will complete a peer evaluation. I will also be monitoring the GitLab logs. Participation and contributions to the group do not need to be equal, but I will penalize individuals who are seen to not contribute their fair share of effort.

Respect:

I encourage questions and interaction during class and believe there are no "stupid questions". However, I do expect you to be prepared for class and to have reviewed the learning module material ahead of time for each class session. I also expect you to be respectful of each other. One common issue in recent semesters is for students to be talking at the same time I am lecturing. This is very disrespectful to me and to your peers. I will give one warning if you are talking at the same time as I am. After that, you will be asked to leave the classroom.

Safe Return to Campus:

All students taking courses with a face-to-face component are required to follow the university's public health and safety precautions and procedures outlined on the university Safe Return to Campus webpage.