Syllabus: ECE652 Safety-critical Real-time Software

Sebastian Fischmeister
Electrical and Computer Engineering
University of Waterloo
sfischme@uwaterloo.ca
April 5, 2018

Course Title: Safety-critical Real-time Software

Course Short Title: Safety-crit RT Software

Instructor: Prof. Sebastian Fischmeister

1 Calendar Description

Concepts, theory, tools, and practice to understand, design, and write embedded software. This covers topics such as embedded computing hardware, embedded computing software, modeling of timing and real-time systems, dependability and safety, security, validation, and performance of embedded systems.

2 Prerequisite Topics

computer architecture, operating systems, programming languages, C

3 Prerequisite Courses

- Operating systems course
- Compilers course
- Programming course

4 Antirequisite Courses

- ECE455 Embedded Software

5 Lab Description

No lab

6 Tutorial Description:

No tutorials

7 Project

The course project can be either a software program or a literature survey report on a topic relevant to the course.

8 Reading Material

- Jane Liu. Real-time Systems, Kluwer, 2000. Chapter 1-6
- Wolf, Marilyn. Computers as Components: Principles of Embedded Computing System Design. Elsevier, 2012. Chapter 1-4
- Lee, Edward Ashford, and Sanjit A. Seshia. Introduction to embedded systems: A cyber-physical systems approach. MIT Press, 2016. Chapter 12-15
- Lyu, Michael R. Software fault tolerance. John Wiley & Sons, Inc., 1995. Chapter 1-5
- Seacord, Robert C. Secure Coding in C and C++. Pearson Education, 2005. Chapters 1-9 (exclude C++ portions)
- Shostack, Adam. Threat Modeling: Designing for Security. John Wiley & Sons, 2014. Chapter 2-3
- Jain, Raj. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons, 1990. Chapter 1-11

- Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem." ACM Transactions on Programming Languages and Systems (TOPLAS) 4.3 (1982): 382-401.
- Regehr, John, and Usit Duongsaa. "Preventing Interrupt Overload." ACM SIGPLAN Notices. Vol. 40. No. 7. ACM, 2005.
- Fischmeister, Sebastian, and Insup Lee. "Temporal Control in Real-time Systems: Languages and Systems." Handbook of Real-Time Systems and Embedded Systems by (2007).
- Driscoll, Kevin, et al. "Byzantine Fault Tolerance, from Theory to Reality." International Conference on Computer Safety, Reliability, and Security. Springer, Berlin, Heidelberg, 2003.
- Rushby, John. "Critical system properties: Survey and Taxonomy." Reliability Engineering & System Safety 43.2 (1994): 189-219.

9 Major topics

- **Part 1: Embedded computing**
  - Lecture syllabus
  - Embedded systems are difficult
  - Instruction sets
  - Computing units
  - Embedded computing platforms
- **Part 2: Timing**
  - Reference model for real-time system
  - Common approaches to scheduling
  - Worst-case execution time problem
  - Clock-driven scheduling
  - Priority-driven scheduling
  - Programming timing constraints
- **Part 3: Safety**
  - Overview of dependable systems
  - Software fault tolerance
  - Interrupt scheduling
  - Byzantine Generals
  - Basics of software qualification
  - Basics of safety assessment
  - ISO 26262 & IEC 62304 & DO 178C
- **Part 4: Security**
  - Security overview
  - STRIDE
  - Attack trees
  - String vulnerabilities
  - Integer vulnerabilities
  - Formatted output vulnerabilities
  - Concurrency vulnerabilities
  - Recommended security practices
- **Part 5: Correctness**
  - Invariants and temporal logic
  - Equivalence and refinement
  - Reachability and model checking
- **Part 6: Performance**
  - Introduction to benchmarking
  - Common mistakes
  - Performance metrics
  - Probes and monitors
  - Ratio games and visualization

10 Grading

Item	Weight
Quiz 1-3	20
Final	50
Project	20

You need at least 50% in each category to get a positive grade.