

UCR CS122A: Intermediate Embedded and Real-Time Systems Syllabus

Basic info

Embedded systems include almost any computing system other than traditional computers. Examples include cell phones, set-top boxes, medical devices, alarm systems, automotive systems, portable music players, etc. Embedded systems is one of the fastest growing areas in computing, having high impact on people's lives, and with tremendous potential for innovative new products. If you think about it, many of the most exciting computing systems today are embedded systems -- iPods and iPhones, Mars rovers, Wii video game consoles, GPS-based people trackers, and more. The embedded systems landscape is one of tremendous innovation and impact -- new inventions, many from students like yourselves, come about every day.

Catalog entry: Systems, 5 units, Lecture, 3 hours; laboratory, 6 hours. Prerequisite(s): CS 012 or CS 013; CS 120B/EE 120B. Covers software and hardware design of embedded computing systems. Topics include hardware and software co-design, advanced programming paradigms (including state machines and concurrent processes), real-time programming and operating systems, basic control systems, and modern chip and design technologies. Laboratories involve use of microcontrollers, embedded microprocessors, programmable logic and advanced simulation, and debug environments.

Lecture M-W 6:10 pm – 7:30pm MSE 103

LAB MW 08:10 a.m. - 11:00 a.m. and TR 08:10 a.m. - 11:00 a.m ENGR2 136

Instructor: Scott Sirowy (ssirowy@cs.ucr.edu)

Teaching Assistant:

Joseph Tarango (jtarango@cs.ucr.edu)

All students must register and participate in the online course component on iLearn.ucr.edu

Recommended Items

* Embedded System Design: A Unified Hardware/Software Introduction, Frank Vahid and Tony Givargis

* Digital Design. Frank Vahid. 1st or 2nd editions.

* The C Programming Language, Second Edition, Kernighan and Ritchie, Prentice Hall, ISBN 0-13-110362-8, OR any other basic C programming book (not C++).

* VHDL for Digital Design, Frank Vahid and Roman Lysecky, John Wiley and Sons, 1st

ed, 978-0470052631. Note that this is the VHDL book, not just Digital Design. It will be used during the second half of the quarter.

Additional online material and articles handed out in class will also be required. Sources for articles include:

- * The Economist (economist.com) -- Nice technology coverage, especially their Technology Quarterly section
- * Discover Magazine (discover.com) -- Excellent discussion of science and related technology
- * HowStuffWorks.com -- Awesome description of how science and technology items work
- * Embedded Systems Programming Magazine (www.embedded.com)
- * TED.com -- Visionary talks by great minds
- * IEEE Spectrum Magazine (spectrum.ieee.org) -- EE-related articles
- * IEEE Computer Magazine (www.computer.org/computer) -- Computer technology research articles

Course grading

Grading is based on the combination of two components:

- 50 pts: Lab component
 - 30 pts: Exercises, assignments, and any lab quizzes
 - 15 pts: Course mini-project
 - 5 pts: Participation (includes attendance)
- 50 pts: Lecture component
 - 7 pts: Homework / online contributions
 - 7 pts: In-class exercises / participation / presentation
 - 36 pts: Quizzes where:
 - Quiz 1: 7 pts
 - Quiz 1: 8 pts
 - Quiz 1: 10 pts
 - Quiz 1: 11 pts

(Participation includes arriving on time and contributing to class discussions.)

Letter grades follow the usual 90/80/70/60 scale. +/- grades will be given. Students must pass (meaning 60%) the lab component and the lecture component *individually* to pass the course.

General course features and policies

* Study groups : Engineering is a social discipline, requiring good people skills. Furthermore, working with others enhances learning. Therefore, we **STRONGLY** encourage studying together. Students in a study group should solve each problem individually. Specifically disallowed is solving a problem as a group, dividing up problems among the group, and other non-individual approaches. Instead, a study group serves to

allow for explanation of what the problem is asking and of generally how to solve the problem, to enable quick assistance when one gets stuck individually solving a problem, and to check each others' solutions. Each student's solution should look distinct -- there are typically countless variations of solutions to a problem.

* Helping others : Helping others enhances one's own learning, and also, well, helps others (remember, you can all get As). Thus, we encourage you to help others in the lab if you have the time. You should not write code for others nor show your code to other lab groups. But helping find bugs, teaching how to debug, and helping to explain concepts, are fine.

* Re-grade policy: corrections must be submitted in writing and within ONE WEEK of the distribution of the graded material. Grade-database errors should also be pointed out within ONE WEEK of posting. So stay on top of your grades.

* News forum / discussion board: Reading all posts is a required part of the course. Questions not relating to an individual grade or personal circumstance should be posted to the discussion board, not emailed directly to the instructor or TAs, so that any of them can respond and so fellow students benefit from the answer.

* Academic dishonesty: Detected cases of academic dishonesty will be pursued aggressively, to protect other students. Grade sanctions and formal university incident reporting will be carried out. Copying code or deriving code for the labs from any source (current students, past students, textbooks, web, etc.) is not allowed unless explicitly authorized. Copying or deriving answers from past homework, quiz, or other solutions is not allowed. Lab partners must all contribute significantly to each lab being turned in jointly; if a partner did not contribute significantly, his/her name should not be included on the lab submission. If you suspect cheating, let the instructors know (anonymous email or notes are welcome).

* Success in computer science courses requires time. A typical student needs to spend about 15 hours per week on this course (including lecture and lab).

Lab guidelines

* Students are required to stay until the end of each lab session, working on lab-related activities. You'll have in-lab exercises, presentations, discussions, and possibly exams. Work ahead or on extra course-related material, and/or provide appropriate help to others, if you finish early.

* During lab discussion time, students should move away from their computers to the whiteboard.

* Prepare for lab before arriving. Bring relevant reference books to lab.

* All persons in lab during scheduled lab time must be formally registered in that section. No unregistered people in the lab are allowed.

* Labs may be done with one partner. Larger groups are not allowed unless explicitly authorized by the TA (rare).

* Do not bring your bike/scooter/etc. into the lab.