Introduction to Embedded Systems

Course Information and Syllabus

Please read the cautionary note about scheduling your lab activities and plan course activities accordingly.

Course Overview

The term “computer” usually conjures up in the minds of many people the image of a mainframe, a minicomputer, a PC, a workstation or a laptop computer. However, computers have always been embedded into all sorts of everyday items from automobiles and planes to TVs, in-house entertainment centers and toasters. These are usually called embedded computers or embedded systems, and actually account for more than 90% of all the world’s manufactured processors. In general, users of embedded systems see a specialized function (such as a High-Definition TV) and do not directly think of the computer embedded within the system. Such embedded computers are gaining importance as an increasing number of systems use embedded processors, RAM, disk drives, and networks. Embedded systems range in size from simple toasters and mini-robots to large-scale systems deployed in process control, manufacturing, power generation, defense systems, telecommunication systems, automotive systems, air traffic control, avionics, video-on-demand and video-conferencing systems. Embedded systems also differ from their conventional PC or workstation cousins in several ways. Embedded systems are typically used over long periods of time, will not (or cannot) be programmed or maintained by its end-users, and often face significantly different design constraints such as limited memory, low cost, strict performance guarantees, fail-safe operation, low power, reliability and guaranteed real-time behavior. These embedded systems often use simple executives (OS kernels) or real-time operating systems with typically small footprints, support for real-time scheduling and no hard drives. Many embedded systems also interact with their physical environment using a variety of sensors and/or actuators. This introductory course on embedded computing focuses on these issues germane to embedded systems.

Topics covered in this course will include:

• Embedded processor architecture and programming,

• I/O and device driver interfaces to embedded processors with networks, video cards and disk drives,

• OS primitives for concurrency, timeouts, scheduling, communication and synchronization,

• Real-time resource management techniques, and

• Application-level embedded system design concepts such as basic signal processing and feedback control.

A hands-on lab component will provide students with direct experience on both the hardware and software commonly used in embedded system design.

Classroom lectures will also be augmented with handouts and other materials.

Goals of the Course

This course is designed with two complementary goals:

1. To understand the scientific principles and concepts behind embedded systems, and
2. To obtain hands-on experience in programming embedded systems.

The first goal will enable you to transcend the "*hot technology du jour*" and the "*buzzword of the month*" trends, and apply concepts as technologies mature/obsolesce and new technologies are born.
The second goal will instill practical skills using hands-on intensive experience and will serve as the path by which you will be "learning by doing".

Specific Goals

- Understand the "big ideas" in embedded systems
- Obtain direct hands-on experience on both hardware and software elements commonly used in embedded system design.
- Understand basic real-time resource management theory
- Understand the basics of embedded system application concepts such as signal processing and feedback control
- Understand, and be able to discuss and communicate intelligently about
  - Embedded processor architecture and programming
  - I/O and device driver interfaces to embedded processors with networks, multimedia cards and disk drives
  - OS primitives for concurrency, timeouts, scheduling, communication and synchronization

Caution on Scheduling Your Lab Activities

Due to hardware constraints, you may have to wait for workstation space in the course laboratory. So, *please*, *please*, **plan to work on your lab projects early and do not wait until the last minute**. You may miss your project deadline just because you had to wait for a workstation to become available. You will not be given grace extensions just because you could not get access to the lab equipment on your schedule. **You *will* have adequate time for each project to be carried out in the lab; this is ample time for you to schedule completion of your project despite these lab constraints.**

Use the “blackboard” facilities available on the course web-page to send e-mail to the instructors, TAs, the course secretary, class students or all of the above. A discussion board is also available for posting questions, comments and answers.

Special Circumstances This version of the course will be broadcast live to an AIT campus in Greece (Athens Institute of Technology). As such, a batch of students will join us remotely on an interactive audio-video hookup adding to the already rich diversity of students on CMU campus. We encourage students on both the CMU and AIT campuses not to be distracted by such a distance-learning infrastructure – it is likely to be the wave of the future! Please engage yourself in the activities of the class by being interactive, raising questions (and offering answers, too) and we strongly expect that the differences posed by the distances essentially go away.

Course/Office Hours

Class **CMU campus students**: Wednesday & Friday 09:30 - 11:20 am (Eastern Time)
AIT campus students: Wednesday & Friday 04:30 - 06:20 pm (Greece Time)

Very Important: We have a limited number of hardware kits, relative to the class size. Please do *not* postpone working on your project until the last minute; you may not get access to the lab equipment workstations at the times you like. Failure to obtain access to a workstation at the last minute will not be considered a valid reason for postponement of a project deadline, or for elimination of penalties in your project score.

Evening recitations will be conducted as the situation warrants and will be relatively infrequent. When conducted, they will normally be in the evening (possibly 7-9pm). Due to the time-zone differences, it is likely that the CMU and AIT campus students will have separate recitation sessions conducted by their respective local personnel.

Office Hours: As listed above for each instructor *and* by appointment.

Textbooks

No regular textbooks will be used in the course. Copies of all reading materials, and lecture handouts, will be available online on the blackboard course web-site. . You are responsible for printing your own hard-copies of handouts to bring to the class-room for each lecture. To give you enough time to do this, electronic versions of the slides for each lecture will be available off the course's BlackBoard site, inside the "Lectures" folder, 24 hours before each lecture.
Exception: The instructors will bring the handouts for quizzes and exams to class.

References

- "*Embedded Systems Design*" by Steve Heath.
- The IEEE POSIX standard.
- "*Principles of Concurrent and Distributed Programming*", by Ben Ari.
- "*Real-Time Systems*", by Jane Liu.
- "*Meeting Deadlines using the Rate-Monotonic Analysis Approach*" by Leblanc and D. Roy.

Programming in C

Because the projects involve extensive use of C programming, it is a good idea to own a C book. One good choice is "*Practical C Programming*"' by Steve Qualline, which can be ordered from O'Reilly & Associates (http://www.oreilly.com/) by calling (800) 998-9938.
Another is "*Portable C*"' by Henry Rabinowitz and Chaim Schaap, which should be available at the CMU bookstore.
In addition to a general C book, we recommend "*C Traps and Pitfalls*" by Andrew Koenig for novice C programmers -- it illustrates common problems encountered by those new to C.

Assembly Language Programming

You will need to know Intel Xscale® (or ARM) assembly-language programming fairly well, to get good grades in this course. If you have not learnt or done assembly-language programming in the past, you have to dedicate considerable time and effort to come to speed on this component of the course.
Things to remember:
You will need to know assembly language programming to just get Project 1 done.
Significant amounts of literature and training materials with examples are available online on the

course web-site.

In addition, use Internet search engines (using keywords like "ARM Assembly Language Programming Tutorial" to track down and read more examples until you feel comfortable.

Assignments/Grading

The course will be graded on an absolute scale, with a total of 1000 possible points partitioned as:

200 points from Quizzes (5 of them, each worth 50 points). The lowest quiz score will be dropped and there will be *no* make-up quizzes.

300 points from Exams (2 of them, each worth 150 points)

500 points from Projects (5 of them, each worth 100 points)

Grading Scale

90% - 100%: A
80% - 89%: B
70% - 79%: C
60% - 69%: D
00% - 59%: R

Wk #	Start Date	Wednesday	Friday	*Notes*
1	Aug 27	Logistics	Introduction to Embedded Systems	*Project #1 out on Fri*
2	Sep 03	Intel Xscale® Assembly Language and C	Intel Xscale® Exception Handling and SWI	*Recitations will be held to provide more detailed information*
3	Sep 10	Intel Xscale® Monitor, Program Loading and Initialization	Profiling and Code Optimization ***Quiz #1***	*Quiz #1 on Fri*
4	Sep 17	Memory, I/O and Microcomputer Bus Architectures	The X-Board: Memory-Mapped I/O and Devices **Project #1 due**	*Project #2 out on Fri*
5	Sep 24	Timers and Interrupts	Interrupts and Serial Comm	*Hardware kit intro*
6	Oct 01	Buffering and DMA ***Quiz #2***	Processes and Process Scheduling	*Quiz #2 on Wed*
7	Oct 08	Memory Management, Overlays and Virtual Memory **Project #2 due**	Concurrency	*Project #3 out on Wed*

8	Oct 15	Traditional OS Processes and Scheduling ***Quiz #3***	**Mid-Semester Break (no class)**	*Quiz #3 on Wed* *Half-way there!*
9	Oct 22	***Mid-Term Exam***	Synchronization & Deadlocks	*Exam #1 on Wed*
10	Oct 29	Resource Management in Embedded Real-Time Systems **Project #3 due**	Resource Management in Embedded Real-Time Systems	*Project #4 out on Wed*
11	Nov 05	Applying RMA in Real Life ***Quiz #4***	A/D & D/A Conversion	*Quiz #4 on Wed*
12	Nov 12	Basics of Feedback Control	Basics of Signal Processing	*Breadth Coverage*
13	Nov 19	Embedded Middleware **Project #4 due**	Real-World Embedded OSs ***Quiz #5***	*Project #5 out on Wed;* *Quiz #5 on Fri*
14	Nov 26	**Thanksgiving Break (no class)**	**Thanksgiving Break (no class)**	*Enjoy the break!*
15	Dec 03	Course Summary and Exam Preparation	**Project #5 due**	*Course wrap-up*
16	Dec 10	***Final Exam Date TBA***	***Final Exam Date TBA***	*Finals Week*

[2] This schedule is tentative and subject to revision to accommodate instructors and travel schedules.

Extra Credit

In addition to the 1000 points, with the permission of the instructors, students may earn up to 25 points of extra credit by writing documentation for various parts of the 18-349 lab.

Evening Recitations

The evening recitation sections will only meet a few times during the term, generally for tutorial sessions or project reviews. Advance warning (including notices on the web pages) will be provided when the recitations will meet. Although it is not required, you should otherwise view the recitation as pre-scheduled time for working on your projects. However, please remember that you signed up for these times, and are required to have them available if they are needed.

Exams

There will be two exams: a mid-term exam and a final exam. The mid-term exam will be in class, in lieu of a lecture. The final exam will be scheduled later during finals week. Tentative exam dates are shown on the class schedule. Exams will focus on newer material (i.e., topics covered since the last exam), but they will be cumulative in that the class builds upon its own foundations. Do not be surprised to see a question about something that many students struggled with on an earlier quiz or exam.

Quizzes

Quizzes will be given in class throughout the term. Tentative quiz dates are shown on the class

schedule. Quizzes will test material covered during the previous few lectures. Quizzes are only given during class and **no early or make-up quizzes will be given**. The lowest quiz score will be dropped at the end of the semester; so if you expect to miss a quiz, plan on missing no more than one quiz!

Projects & Late Policy

The projects are designed to

1) Help you understand and synthesize all of the course concepts;

2) Demonstrate your abilities at correctly implementing embedded systems; and

3) Demonstrate your ability to explore and incorporate good engineering trade-offs.

We encourage you to work together with your classmates to help you understand the basic concepts. However, you are required to do your own projects. Any aid or ideas that you incorporate into your project must be documented with:

1) The extent of the help that you received, and

2) The names of the individuals who helped you.

Failure to do this will constitute cheating and result in a failing grade for the course.

Projects may be submitted after the due date, but are subject to a 10% (of the available points) per day late penalty 1.

1. Any days, *including* Saturday and Sunday, are counted as late days.

2. Due to Carnegie Mellon University policy, we cannot accept *any* work after the last day of class. Therefore, all projects must be submitted by 5:00 pm on the last day of classes - there will be *no* exceptions.

Web Pages and the Class Mailing List

Throughout the term, we will post assignments, announcements, and class notes on the web site. It's a good idea to periodically check the web-site. On the class's home web page, you will find a form for subscribing to the class mailing list. After you have submitted your request to be added to the class mailing list, you should receive confirmation within 24 hours. Anytime you want to post to the class mailing list (like a "bboard", so everyone will see your message) or send email to the TAs, course secretary and/or the instructor, please log on to the course web-page and click on “Communication” and then “Send E-mail”.

Questions, Communications and Level of Difficulty

If you have questions in class (or you find the lectures monotonous ☺), feel free to interrupt and ask questions during a lecture. The professors will try to answer questions that must of interest to many students immediately in whatever depth that is necessary. Questions of particular interest to you but not to others are also appropriate, but may be answered off-line (outside the class). Both the instructors love these individual interactions with students and encourage students to be intellectually curious. Typically, they will also try to write up details of a specific topic raised by a student for later distribution in class as handouts. Suggestions of such topics and write-ups are always welcome. Academics/researchers love to write: so take advantage of it!

The class is expected to be big (>100 students) - so get the biggest bang for your tuition dollars by contacting and interacting with the instructors directly and personally.
The instructors will also try to stimulate your thinking juices by giving out handouts and articles that discuss advanced concepts pertaining to embedded real-time systems. While you do not need

to understand these materials to do well in 18-349, they will try to provide additional food for thought. The market for embedded systems is exploding, and you should each be thinking of new and innovate ways to contribute to (and benefit from) this trend.

PLEASE READ AND ABIDE BY THE RULES

Course Policy on Cheating

We would like to promote a collaborative environment where people feel free to openly discuss and ask questions. However, when assignments are submitted, the work must be the author's own and any aid received from other people must be documented in the assignment. Simply put, cheating is submitting work as your own that is *not* your own; material handed in for grading must be the product of individual effort (or your group effort on group projects) anything else constitutes cheating.

Cheating will result in a failing grade for the course. Students are referred to the University Policy about Cheating and Plagiarism.

Maintain integrity

You are all training to become professionals. So, **act professionally**. Integrity is a core aspect of professional ethics. You need to both **exhibit integrity** and **expect integrity**.

- Do **not** use solutions from prior years.
- Do **not** hand over solutions to students for use in successive years.
- To **be fair** to most students who work hard and play fair, any and all violations will be dealt with seriously. You could be putting your degree and career in serious trouble by violating this code of ethics. Think before you act: are you ready to take the fallout? If you think you can, you would be wrong.