EECS 461: Embedded Control Systems, Fall 2009

CLASS TIME: 12:00-1:30 Tuesday and Thursday

LAB TIMES: Monday 2:00-5:00; Tuesday 3:00-6:00; Wednesday 9:30-12:30; Thursday 3:00-6:00

PLACE: 1005 Dow Bldg (lecture), 4342 EECS Bldg (lab)

INSTRUCTOR: J. S. Freudenberg

OFFICE: 4425 EECS bldg

PHONE: (734) 763-0586

EMAIL: `jfr@eecs.umich.edu`

GRADUATE STUDENT INSTRUCTORS: Zhaori Cong, Jessica Szemraj

OFFICE HOURS: 2:00-3:00 Monday and Wednesday, but feel free to stop by anytime, or email me to set up an appointment

PREREQUISITE: EECS 373 or EECS 216 (formerly EECS 306) or approval of instructor

GRADING

- Homework/Laboratory Assignments: 50%
- Quizzes (tentatively scheduled October 29th and December 8th): 30%
- Project: 20%

TEXTBOOK: There isn't a single textbook that covers all the topics of this course. We will provide handouts on most topics of the class. A good review of hardware fundamentals, interrupts, and operating systems is found in *An Embedded Software Primer* by David E. Simon, Addison Wesley, 1999, ISBN: 0-201-61569-X.

USEFUL REFERENCES: The following list of books provide useful background for various lecture topics:

- J. Lemieux, *Programming in the OSEK/VDX Environment*, CMP Books, 2001.
- D. Auslander and C. J. Kempf, *Mechatronics: Mechanical Systems Interfacing*, Prentice-Hall, 1996.
- J. Ledin, *Embedded Control Systems in C/C++*, CMP Books, 2004.
- R. Soja and M. Bannoura, *MPC5554/MPC5553 Revealed*, AMT Publishing, 2005.
- A. Burns and A. J. Wellings, *Real-time systems and programming languages*, Pearson, 2001.
- C. M. Krishna and K. G. Shin, *Real-Time Systems*, McGraw-Hill, 2001.
- W. Wolf, *High-Performance Embedded Computing: Archutectures, Applications, and Methodologies*, Morgan Kauffman, 2007.

In addition, a very interesting and useful website is `www.embedded.com`. This site has many articles relevant to the industry and this course.

WEB PAGE: `www.eecs.umich.edu/courses/eecs461`

EMAIL ALIAS: `eecs461@eecs.umich.edu`

It is NECESSARY to subscribe to the email alias, as important information may be distributed that way. To do so, send an email message to "`eecs461-request@eecs.umich.edu`" with "subscribe" in the subject line. You will receive a return message – you must read this message and follow instructions to confirm your subscription to the email alias. To remove your name from the alias, follow the same procedure, but with "unsubscribe" in the subject line.

OVERVIEW: The vast majority of microprocessors are not used in desktop or laptop computing applications. Instead, they are *embedded* in other technological systems, such as cell phones, appliances, and automobiles. One unique feature of embedded applications is that the goal of the design is not directly related to the performance of the microprocessor, but rather to the performance of the overall system. When I drive my car, I don't care what kind of microprocessors (if any) are being used to control the engine, transmission, and brakes; I only care that they work, and work reliably! When I make toast, I don't know or care what kind of microprocessor is in my toaster (it has one!)

In EECS 461 you will learn how to use a microprocessor as a component of an embedded control system. The specific embedded system we will be working with is a *haptic interface*, which uses force feedback to enable a human to interact with a computer through the sense of touch. The *skills* we shall develop in doing so are applicable to a broad range of embedded system applications.

There are many differences between programming a general purpose computer and programming for an embedded application. Among these is the fact that embedded computations must be synchronized with the evolution of a physical system. Embedded software is often safety critical, and thus cannot fail. There are also constraints on memory and power consumption not present in general purpose computing.

Lecture Topics

- Sampling. Position and Velocity Measurements. Encoders. Quadrature Decoding.
- The MPC5553 Time Processing Unit (TPU) and its Quadrature Decoding function.
- Pulse Width Modulation (PWM). Frequency response of PWM signals. DC motors. Amplifiers. Interface electronics.
- Haptic interfaces. Virtual worlds. Human-computer interaction. Applications to games, manufacturing and robotics, drive-by-wire aircraft and automobiles, flight and surgery simulators, video editing. "Artifacts" due to microprocessor implementation of the virtual world.
- Algorithms. Feedback systems. PID control. Logic control and finite state machines. Numerical integration. Implementing a virtual world on a microprocessor.
- Concepts from real time operating systems. Interrupts. Shared data. Latency. Round-robin architectures. Single vs. multitasking. Semaphores.
- Real time computation. Rate monotonic scheduling.
- Modeling. Use of Matlab/Simulink/Stateflow to simulate the interaction of a virtual world with a human operator through the haptic interface.
- Networking. Control Area Network (CAN) vs. Ethernet protocol. The CAN unit on the MPC5553.
- Rapid prototyping and autocode generation.

Laboratory

During the first several weeks of the semester, the laboratory exercises will develop an embedded controller for a haptic interface. The software will be written in *C*. We will implement the controller over a CAN network to study performance degradation due to networking delay. We will then recreate our work using rapid prototyping tools to generate *C* code directly from a Simulink model of the haptic virtual world. We shall structure the generated code as task states in the OSEKTurbo real time operating system.

Project

Late in the semester we will complete a small project using the hardware, software, and haptic interfaces in the embedded systems laboratory. This semester's project will be to develop an adaptive cruise control system allowing different lab groups to interact over the network and receive both visual feedback from a computer monitor as well as haptic feedback.