EECS 461: Embedded Control Systems, Winter 2008

CLASS TIME: 1:30–3:00 Monday and Wednesday

LAB TIMES: Monday, 3:30–6:30; Tuesday, 1:30–4:30; Wednesday, 10:00–1:00; Thursday, 1:30–4:30

PLACE: 1311 EECS (lecture), 4342 EECS Building (lab)

INSTRUCTOR: J.A. Cook

OFFICE: 4238 EECS Building

EMAIL: jeffcook@eecs.umich.edu

GRADUATE STUDENT INSTRUCTORS: Jeff Roder (M, Th), John Schmotzer (Tu, W)

OFFICE HOURS: 10:00 – Noon, Monday and Wednesday, but feel free to stop by or email me to set up an appointment

PREREQUISITE: EECS 373 or EECS 216 or approval of the instructor

GRADING:

- Homework: 25%
- Laboratory Assignments: 25%
- Quizzes (tentatively scheduled for February 20th and April 2nd): 30%
- Project: 20%

Homework is due at the **beginning** of class. Late homework will not be accepted. The *Homework Policy* and *Lab Policy* are posted on the course website, and included in the syllabus. Please read them and follow instructions.

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

TEXTBOOK: There is no required textbook for this course. The primary reference on real time operating systems is *An Embedded Software Primer* by David E. Simon, Addison Wesley, 1999, ISBN: 0-201-61569-X. This book also has a good overview of hardware fundamentals. Lecture notes and other useful information will be posted periodically on the course website.

OTHER USEFUL REFERENCES:

- 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.

In addition, an interesting and useful website is www.embedded.com.

EMAIL ALIAS: eecs461@eecs.umich.edu

To subscribe to the email alias, 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. It is NECESSARY to subscribe to the email alias, as important information may be distributed that way.

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 is being used to control the engine, transmission, and brakes; I only care that they work, and work reliably. 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, or force feedback system. The skills we shall develop are applicable, however, to a broad range of embedded system applications.

Lectures

The lectures cover a wide range of topics, including:

- Sampling. Position and Velocity Measurements. Encoders. Quadrature Decoding.
- The MPC5553 Time Processing Unit (TPU) and its Quadrature Decoding function.
- Features of the MPC5553 Microcontroller.
- Pulse Width Modulation (PWM). Frequency response of PWM signals. DC motors. Amplifiers.
- Interface electronics.
- Haptic interfaces. Virtual worlds. Human-computer interaction.
- Artifacts due to microprocessor implementation of the virtual world.
- Algorithms. Feedback 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. Software architecture.
- 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 MPC555.
- Rapid prototyping and autocode generation.

Laboratory
You will have a lab partner. 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 by creating a Simulink model of the haptic virtual world and using Real Time Workshop to automatically generate the embedded software. Labs will begin **Monday, January 14th**. Due to MLK day, the lab section normally meeting on **Monday, January 21st** will meet on **Friday, January 25th**, time TBD.

Project
Late in the semester we will complete a driving simulator project using the hardware, software, and haptic interfaces in the embedded systems laboratory.

Homework Policy

(1.) You may consult with one another on homework problems.

(2.) The solutions you hand in must be written up by yourself, and represent your own intellectual output.

(3.) Under no circumstances may you look at solutions from previous years. Nor may you use Matlab and Simulink code from previous years.

Lab Collaboration Policy There are 5 parts to a lab:

(1.) Pre-lab questions: These are questions asked about the concepts that the lab addresses, and often relate to topics covered in lecture. Some pre-lab questions also test your knowledge of the MPC5553 and its peripherals. Such questions can be answered by referring to the relevant sections of the MPC5553 manual or to any programming notes that accompany a lab. All of these questions are to be answered **INDIVIDUALLY**.

(2.) Pre-lab code, for use in lab: Some labs will come accompanied with header files and skeleton library files and instructions about how certain functions should behave. These functions will be used during the in-lab, and you can, and should, work with your partner on writing them before coming to lab.

(3.) In-lab experiments: You will work together with your partner on anything in the "In-Lab" section of a lab handout.

(4.) Post-lab questions: These are questions about any concepts you would have learned about during your In-lab, or inference questions to test your understanding of what you observed in lab. You may discuss any concepts for these questions with your lab partner, however with **NO ONE** outside your group. You and your partner are required to write up your own solutions **SEPARATELY FROM ONE ANOTHER** for all of the post-lab questions.

(5.) Code to be submitted with the post-lab: You are required to submit, as part of your post-lab, a copy of your code. You should turn in your own copy of the code along with your answers to the pre-lab and post-lab questions. **You may not, under any circumstances, refer to code written by students in previous semesters**.