EECS 461: Embedded Control Systems, Winter 2018

CLASS TIME: 12 – 1:30 Monday and Wednesday

PLACE: 1500 EECS (lecture), 4342 EECS (lab)

INSTRUCTOR: J.A. Cook

OFFICE: 4233 EECS

EMAIL: jeffcook@umich.edu

OFFICE HOURS: Monday and Wednesday 2:00–3:30. Don't hesitate to email me or set up an appointment at another convenient time.

LAB INSTRUCTORS:

Shivani Shah (shivanid@umich.edu) Mo 2:00PM–5:00PM and We 3:00PM–6:00PM

Ben Hansen (shansenb@umich.edu) Tu 2:00PM–5:00PM and Th 2:00PM–5:00PM

Jay Mulani (jmulani@umich.edu) Tu 10:00AM–1:00PM and Th 10:00AM–1:00PM

Chris Schmotzer (cschmotz@umich.edu) (We 8:00AM–11:00AM and Fr 11:00AM–2:00PM

Labs will begin **Monday, January 8**.

GRADING:

- Homework: 25%
- Laboratory Assignments: 25%
- Quizzes (likely to be administered on-line, timed via Canvas, probably the week after Spring Break and the last week of class): 30%
- Project: 20%

HOMEWORK is to be submitted **electronically** as specified in the Canvas Assignment. **Late homework will not be accepted.**The *Homework Policy* and *Lab Policy* are posted on the CTools website, and included in the syllabus. Please read them and follow instructions.

ATTENDANCE: Students are expected to attend ALL lectures and ALL labs from BEGINNING to END. I realize that there are situations, such as interviews and plant trips, that make this impossible, and we will work to accommodate students when this happens. Nonetheless, missing lecture or lab should be an exception, rather than the rule!

TEXTBOOK: There is no required textbook for this course. Lecture notes and other useful information will be posted on Canvas.

OTHER USEFUL REFERENCES:

- David E. Simon, *An Embedded Software Primer*, Addison Wesley, 1999, ISBN: 0-201-61569-X.
- 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.
- 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.

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 MPC5643L eTimer and its Quadrature Decoding function.
- Features of the MPC5643L 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 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. Late in the semester we will complete a driving simulator project using the hardware, software, and haptic interfaces in the embedded systems laboratory.

Note:

- You must attend the entire 3 hour lab section for which you have registered.
- You may not change lab sections.

Homework Policy

(1.) Homework is due electronically as specified in the Canvas Assignment. Late homework will not be accepted.

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

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

(4.) **Under no circumstances may you look at solutions from previous years. Nor may you use Matlab and Simulink code from previous years. It is a violation of the Honor Code to do so.**

Lab Collaboration Policy

(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 MPC5643L and its peripherals. Such questions can be answered by referring to the relevant sections of the MPC5643L 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.

(6.) **You may not, under any circumstances, with respect to the laboratory exercises or the final project, refer to code or models written by students in previous semesters. It is a violation of the Honor Code to do so.**

Lab Late Policy

(1.) Late pre-labs will not be accepted.

(2.) In-labs are meant to be completed in your assigned section. If you do not complete the in-lab assignment in your assigned section, then you will have one week to complete it during open lab hours.

(3.) It is the student's responsibility to notify the GSI if he or she will miss a lab section.

(4.) Late post-labs will not be accepted.

Reading Assignments There are no required textbooks for this course, but there is a lot of reading material that support the lectures and laboratory assignments. Unless otherwise specified, you will find this material on Canvas: EECS 461 001 WN2018 Files–Lecture Notes. Here's what you should be reading and when:

1. January 3, 2017:
    - `C_lecture.pdf:` This file is located in EECS 461 001 WN2018 Files–Labs–Lab1. Introduction to C programming for embedded systems, including important information required for Lab 1.
    - `EECS461_course_logistics.pdf:` What is an Embedded System?
2. January 8, 2017:
    - `Lecture2.pdf:` Interfacing a Microprocessor to the Analog World.
    - `Lecture3.pdf:` Position and Velocity Measurements.
3. January 15, 2017:
    - `Lecture4.pdf:` The World of Sensors.
    - `Stateflow_Modelling.pd` Stateflow tutorial.
4. January 22, 2017:
    - `Lecture5.pdf:` Actuators.
    - `Lecture6.pdf:` Motor Control.
5. January 29, 2017:
    - `Lecture7.pdf:` Feedback Systems.
    - `Notes_Lab3.pdf:` This file is located in EECS 461 001 WN2016 Files–Labs–Lab3. Discussion of sampling, aliasing and *beating* observed in lab 3.
6. February 5, 2017:
    - `Lecture8.pdf:` Feedback Systems.
    - `state_space.pdf:` System modeling in state space.
    - `Wall_Chatter_short.pdf:` Discussion of the "wall chatter" phenomenon observed in the "virtual wall" laboratory exercise.
    - `Wall_Chatter_long.pdf:` Extended discussion of the chatter phenomenon (optional).
7. February 12, 2016:
    - `Implementation_Virtual_Spring.pdf:` Implementation issues with the virtual spring.
8. February 19, 2016:
    - `SWArchitecture.pdf:` Embedded Software Architecture.

- SWArchitecture_Slides.pdf: Lecture slides for software architecture discussion.

9. March 5, 2016:
   - CAN_MPC5643L_W16.pdf: Controller Area Network (CAN).
   - CAN_slides.pdf: Lecture slides for CAN.

10. March 12, 2016:
    - code_generation.pdf Automatic code generation from Simulink models.
    - Analyzing_Generated_Code.pdf: Evaluating the generated code.

In early March, a Canvas folder will be created which will contain many important documents pertaining to the final project. Additionally, each lab folder has a copy of introductory material for the lab exercise. Finally, there is a folder on Canvas called "Handouts." This folder contains articles from the technical and popular press about embedded systems that you might find interesting, but which are not required reading.