ECE 4760/5730: Digital Systems Design Using Microcontrollers
Author: Lecturer, Hunter Adams, ECE
Revision date: 8/16/2024
Course website: **https://ece4760.github.io**

Credits: 4 credits

Catalog Description:
Design of real-time digital systems using microprocessor-based embedded controllers. Students working in groups of 2-3 design, debug, and construct several small systems that illustrate and employ the techniques of digital system design acquired in previous courses. The content focuses on the laboratory work. The lectures are used primarily for the introduction of examples, description of specific modules to be designed, and instruction in the hardware and high-level design tools to be employed.

This is a Cumulative Design Experience (CDE) course. As such, lab exercises in this course will draw upon and synthesize as much material from the undergraduate electrical and computer engineering curriculum as possible. Students will implement concepts from differential equations, physics, a variety of coding classes, digital signal processing, analog circuits, control theory, computer graphics, robotics/mechatronics, and more in ***real, physical projects***.

Course Frequency:
Offered every Fall semester

Prerequisites:
ECE 3140 or permission of instructor

Corequisites:
NONE

Preparation Summary:
C programming: Students need to be comfortable with programming in the C language and understand the use and implications of software concurrency and interrupt handlers.

Electronic construction: Students should be familiar with construction of electronic circuitry on solderless breadboards, and will be required to learn to solder.

Textbook(s) and/or Other Required Materials:
Reading list is summarized for each lab assignment on the lab assignment webpage. Please find those webpages under "Assignments" on the course website linked here.

The readings cover the peripherals required for each lab, as well as specific techniques necessary to implement the labs. The readings are a combination of vendor (Raspberry Pi) manuals, and instructor- generated web pages. Link here.

The three main documents are RP2040 datasheet, RP2040 C SDK manual, and the instructor-generated demonstration code repository. Code repository linked here.

Class and Laboratory Schedule:
Lectures: 3 hrs/wk. Lecture time transitions to additional lab time around the start of the final project.

Lectures from 2022 are available here.

Recitations: None

Labs: There are weekly lab sessions. There will be 3 lab assignments, each of which will be 3-4 weeks long. Current labs are available here.

Since virtually all work for this course is lab work, you will be expected to be in lab 6 hours per week. Each lab will have a formal write-up, as explained on the policy page linked here. When we transition to final projects, you will be expected to be in lab 9 hours per week. **Each laboratory assignment will also require out-of-class preparation, which you are expected to accomplish independently.** Hunter will be in the lab with you for all scheduled laboratory sections.

Students enrolled in the 5000-level version of the class will have additional requirements for each laboratory assignment. These additional requirements will be assignment specific, and the assignments often change from semester to semester. There will be no extra time granted to 5000-level students to achieve these extra requirements.

	Lectures	Lab work	Reading/preparing for lab	Writing lab reports/ documentation	Total
Instruction hours per student per week (before transition)	3	6	3	4	16
Instruction hours per student per week (after transition)	0	9	3	4	16

Assignments, Exams and Projects:
Note: Syllabus subject to change prior to course start. Final syllabus posted on course Canvas or website.

Homework: Approximately four assignments (a laboratory report for each laboratory, and a final webpage). Collaboration within groups is required.

Exams: none.
Independent Study Project: Extensive, 5 week, final project with full write-up. The student final deliverable is a webpage and a technical demonstration.

- Please find details here.
- Student projects since 2022 available here.
- Student projects from 1999-2021 available here.
- Demonstration videos for 2022 (RP2040) available here.
- Demonstration videos for 2020-2021 (PIC32) available here.
- Demonstration videos for 2015-2019 (PIC32) available here.

Typical Topics Covered:
- Review software concurrency, interrupt service routines, and threads.
- Hardware concurrency: Getting good performance by using all of the concurrently executing co-processors available on the microcontroller chip. Including: DMA, five timers, SPI, UART, ADC subsystem, and others.
- Thread programming
- Precision time interval measurement/generation
- 2D graphics
- SPI DAC and noise considerations
- Scanning a keypad: connections, scan code, and state machine
- Fixed point arithmetic for speed
- Direct Memory Access (DMA) controller for fast i/o.

- Motor control: PWM, PID controllers, need for optoisolation
- Power saving modes
- Sound synthesis – Direct Digital Synthesis
- Specifying, planning, building, and testing of a project.
- Analog noise and circuit layout.
- Physical construction. Soldering, board layout.
- Debugging of mixed hardware/software systems.

Student Outcomes:
1. Be able to compute the performance of circuitry, including loading effects.
2. Be able to use a microcontroller development system and appropriate software tools (assembler and C compiler)
3. Be able to wire and debug analog and digital circuitry
4. Be able to specify a project and carry out a detailed design.
5. Be able to calculate error budgets for timing and performance
6. Consider human factors (safety, usability, etc.) when designing embedded devices
7. Work as a team to produce timely solutions for projects.
8. Produce demonstrations and documentation.
9. Be able to describe the significance of microcontrollers in our technical infrastructure and the social,
political, and ethical implications of automation and miniaturization.

Grading Rubric

This class is composed of laboratory assignments and a final project. You will be graded on these tasks, plus some consideration of class participation. The final design project demonstration is due the last week of classes. The grade breakdown is as follows:

50% for laboratory assignments. For each laboratory assignment:
- 10% for preparedness and participation, as determined by conversation with Hunter in lab, lecture attendance, and occasional in-lecture quizzes.
- 20% for weekly progress deadlines (evenly distributed across all weeks for a particular lab)
- 30% for the final laboratory demonstration
- 40% for the laboratory report

50% for the final project. This is further broken down similarly to the laboratory assignments:
- 10% for preparedness and participation, as determined by conversation with Hunter in lab, lecture attendance, and occasional in-lecture quizzes.
- 20% for weekly progress deadlines (evenly distributed across all weeks of the final project)
- 30% for the final laboratory demonstration
- 40% for the final report

For higher-resolution descriptions of how these scores are determined, please see the course policy page, linked here.

Hunter reserves the right to make small adjustments to your grade. You may help your grade by participating in class discussions. Excessive nonattendance of lecture may lower your grade. If you feel that you have been unfairly graded, you have one week from the time the assignment is handed back to request a regrade. To request a regrade, you must submit the assignment with a written description of your concern attached to the instructor (not the TA).

Academic Integrity:
Students expected to abide by the Cornell University Code of Academic Integrity with work submitted for credit representing the student's own work. Discussion and collaboration on homework and laboratory assignments is permitted and encouraged, but final work should represent the student's own understanding. Specific examples of this policy implementation will be distributed in class and are given here.

Should copying occur, both the student who copied work from another student and the student who gave material to be copied will both automatically receive a zero for the assignment. Penalty for violation of this Code can also be extended to include failure of the course and University disciplinary action.

With regard to AI . . .

- For your lab reports, you must cite and quote AI-generated content in an identical fashion to that with which you must cite and quote human-generated content. Failure to do so will be considered plagiarism.
- You may use AI-generated code at your own risk. **Any AI-generated code must be clearly labeled as such. Please see the example below. Failure to label any AI-generated code will be considered plagiarism. The course staff will not help you debug AI-generated code.**

Accommodations for students with disabilities
In compliance with the Cornell University policy and equal access laws, the instructor is available to discuss appropriate academic accommodations that may be required for student with disabilities. Requests for academic accommodations are to be made during the first three weeks of the semester, except for unusual circumstances, so arrangements can be made. Students are encouraged to register with Student Disability Services to verify their eligibility for appropriate accommodations.

Grading scale

- 97-100: A+
- 93-97: A
- 90-93: A-

- 87-90: B+
- 83-87: B
- 80-83: B-
- 77-80: C+
- 73-77: C
- 70-73: C-
- 67-70: D+
- 63-67: D
- 60-63: D-
- <60: F

Inclusivity Statement
We understand that our members represent a rich variety of backgrounds and perspectives. Cornell is committed to providing an atmosphere for learning that respects diversity. While working together to build this community we ask all members to:

- share their unique experiences, values and beliefs
- be open to the views of others
- honor the uniqueness of their colleagues
- appreciate the opportunity that we have to learn from each other in this community
- value each other's opinions and communicate in a respectful manner
- keep confidential discussions that the community has of a personal (or professional) nature
- use this opportunity together to discuss ways in which we can create an inclusive environment in this course and across the Cornell community