CompSci 335
Inside the Box: How Computers Work
Course Syllabus
Fall, 2024: MW 4:00 - 5:15 PM ET (2H30M/wk), 3 Cr., CS-142

Chip Weems
E-mail: weems@cs.umass.edu
Phone: 413-545-3163 (or email request for call)
Office: CS-342
Office Hours: M 10:30 - 11:30
And by appointment

TA:Emily Pruc epruc@umass.edu. Hours: Tu 10 - 11, in LGRT-220, and by appointment

Optional Textbook: Computer Organization and Architecture, Themes and Variations, Alan Clements, Cengage Learning, 2014 (very helpful for filling in beyond the lectures)

Required Hardware: To use the logic simulator and the development environment for the microcontroller you will need a computer that runs Windows, MacOS, or Linux and has a Java virtual machine installed. You will also need a USB-A port or adapter to power the microcontroller kit and to download code to it. A stock Chromebook will not be sufficient.

Course Objectives: As computer scientists, we tend to program using abstractions significantly removed from the hardware on which our code executes. Yet there are many cases where misunderstanding how the hardware works leads to poor performance, errors, security vulnerabilities, and other modes of failure. Software can't completely hide physics! A deeper understanding of the underlying mechanisms of computers also helps us distinguish real technological advances from marketing hype. In this course, we pull back the covers to see how machines operate at the lowest level, and examine the electronic, mechanical, and physical principles behind the operations. We will also explore enhancements to machine architecture beyond the basics.

Course Notes: Slides, assignments, participation exercises, handouts, Echo360 recordings, etc., are all available from the Canvas site. If you miss a class, it is important to check Canvas for updates. I do my best to update them as soon as possible, but other commitments sometimes cause me to fall behind for a few days.

Grading: In class exercises and participation in class: 50% Logic labs: 16%
Assembly labs: 12% Project proposal: 2%
Term project: 16% Project Presentation: 4%

Grade Scale:
A: 93%, A-: 89%, B+: 85%, B: 80%, B-: 77%, C+: 74%, C: 70%, C-: 67%, D+ 63%, D: 60%

These percentages are estimates, and I reserve the right to make minor changes as necessary.

In class exercises and attendance/participation in class: Most class periods will have an in-class exercise to prepare for or solidify the material presented in the lecture portion. Those may include building a circuit with the logic simulator or prototyping board, demonstrating something with the microcontroller, engaging in a small group discussion, and other activities. The first two periods, during add/drop are not counted. The remaining 25 will be 2% each. Many of these cannot be made up if you miss class, so **attendance is important.**

If you have an unavoidable conflict let me know in advance and I will try to work something out. Some of the exercises will be things that can be made up outside of class, so do ask if that's possible. Don't just assume that it's not possible. 2% seems like a small amount, but it can add up quickly to a significant fraction of your grade. See the Extension Requests section below. **This work takes the place of exams. There are no exams, but the final exam period will be used for project demonstrations.**

Submission Methods:

Photo submissions: Some of the in-class exercises will involve construction of a circuit and/or executing code on the microcontroller. Evidence of having participated in the exercise will be in the form of submitting a photo of something on the prototyping board. Thus, you will need a way to shoot photos of your work (a cell phone is fine). Images should be submitted as jpeg or png files, as a normal file submission in Canvas.

Screenshot submissions: The logic labs and some of the in-class exercises will be done with software packages (Logisim-Evolution, Arduino environment), and part of demonstrating completion will be capturing a screen image. Make sure that you know how to do this for your operating system. These images should also be in jpeg or png form.

Discussion submissions: Some of the in-class work will involve discussions and problem solving by randomly assigned groups. These assignments will be submitted as either a Quiz or a Survey in Canvas (in this case, a Quiz just happens to be what Canvas provides that has the necessary flexibility for some types of question - it's not actually a quiz).

Other activities: These will be normal Canvas submissions, such as worksheets, code files, etc.

In-class Group Submissions: Part of the submission will ask you for the names of people in your group. However, everyone in the group should submit a copy, to ensure they get credit.

Lab Homework

Due Dates: Homework is generally due at the start of class one week after it is handed out.

Late Policy: Much of the homework is preparation for work we'll be doing in class, so it is important to keep up. It is better to turn in incomplete wok and get partial credit than to fall behind and start missing assignments. If you know you have a specific time conflict, make arrangements with me in advance for a separate assignment for late submission. In some cases, feedback will be given when the homework shows a lack of understanding, and a makeup assignment will be given – the goal of the homework is for you to go through an experience that helps you to learn key concepts. It is not focused on producing a product.

Collaboration: If someone asks you for help on the lab homework, try to take the role of a teacher and help them find the solution on their own; don't just give them an answer. If it is clear that people are submitting answers that are merely copies of each other, the credit for the one solution will be divided among the copies. Of course, the TA and I are here to help, and welcome questions, so you can always suggest that someone contact us.

Grading Tip: The homework involves using tools to gain a deeper understanding of how a computer works at the level of the digital logic, and the native instruction set. These are more about having an individual experience, than getting the right answer (although the goal of the experience is for you to get something working).

I typically lean toward generosity for partial credit, but the TA and I can do that only when you've made an attempt. If you leave a blank on an in-class exercise or don't submit a lab, then it can't be given any credit. **Always try!**

Lab Homework Summaries

Logic Labs: There are four digital logic labs, each of which has an associated handout that walks you through a step-by-step process of creating a series of circuits in Logisim. The handout asks you to experiment with the circuits and answer questions at various stages in the construction process and submit screen shots. They are briefly summarized here:

Logic lab 1: Use Logisim to simulate and explore simple combinational Boolean logic circuits: AND, OR, NOT, NAND, NOR, XOR, rediscover DeMorgan's Law. (1 hour)

Logic Lab 2: Use Logisim to build a full adder, then a multibit adder/subtracter. (1 hour)

Logic Lab 3: Use Logisim to build an RS flip flop, a D flip flop, a register, and then a simple processor datapath with multiple functions. (2 hours)

Logic Lab 4: Extend the data path to include a program counter, memory, and branch logic. By this point you will have seen how a computer works to execute instructions. (2 hours)

Logisim-evolution simulator download info can be found here: https://github.com/logisim-evolution/logisim-evolution#download

Assembly Programming Labs: Three projects that are designed to get you up to speed on mixed C and Assembly programming for the ARM processor using the Arduino development environment. They give you a sense of how software interacts with hardware.

Programming Lab 1: Build an assembly language pause loop subroutine that can be called from C or assembly language, using the C calling convention, to pause for a specified amount of time. Starts with in-class work to get pieces working (2 hours)

Programming Lab 2: Use assembly to get lights connected to the microcontroller to illuminate to show the low order 8 bits of a series of values passed in by a C program, with the display time controlled by your pause subroutine (I/O port driver and calling the subroutine from Lab 1). Also starts with in-class work on analogous components. (3-5 hours)

Programming Lab 3: Change the Lab 2 program to do all of the looping in assembly, given pointer(s) to array(s) of values from C (nested looping and calling, array access). Class covers the principles of each part, but you flesh out and combine them. (4 to 6 hours)

Semester Project: Using the the microcontroller on the prototyping board and additional components as desired, develop a project of your choice that you think will impress the rest of the class. You may add components of your own to those on the board - there are many sources online, such as Adafruit and Sparkfun, and I have many that I can loan. (8 to ? hours)

The term project will be done in teams of two, although individual projects may be allowed for people with prior experience. It's helpful if the team can identify a clear division of labor, but team members should also be helping each other through challenges. This is a place where collaboration is encouraged. Unless you are doing a multi-microcontroller project, the expectation is that team will run the final demonstration on just one of their microcontrollers.

You will submit a brief report describing the development process (including who did what), and what you learned from it (especially any challenges that were overcome), along with your code. It is OK to include code from the web, but you must clearly identify which code is yours and which is imported, and from where. In addition to the report, the grading criteria for the project will include the depth of learning involved (which can be shown through factors such as algorithmic complexity and/or use of different features of the microcontroller), and how well the project fulfills its goals.

There will be a proposal form to fill out for your project, so you can get feedback on it before you get too far along. Credit is given for submitting it.

The project will be publicly demonstrated during the final exam period, with the whole department invited to attend. Each presentation will be evaluated in part (very small fraction) by other students in the class on the basis of its impressiveness. Mostly the evaluation is based on how well it conveys the operation of the project and the effort that went into its development. Submitting impressiveness evaluations for the presentations will be treated as an in-class exercise.

Project Teams: When you are in a team, then your grade is partially dependent on your performance as a team. Some ways you can help to ensure that your team does well:

Be realistic with your partner in discussing your abilities and time commitments when you divide up the project work. If possible, get together and work collaboratively, taking time to brainstorm ideas for how you will make your project impressive, and how you will adapt if a part isn't working out. Paired programming is a recognized software engineering technique in industry, and is known to yield higher productivity, especially on small, intensive assignments. In many cases, you'll learn more and learn it more quickly through working together directly.

Make sure you have each other's schedules, phone numbers, e-mail addresses, etc. Establish a clear policy of when it is OK to contact one another. Let each other know when you are going to be unavailable with enough advance warning for good contingency planning.

Communicate! Communicate! Communicate! Tell each other what you are thinking. Don't keep thoughts inside. Remember to praise each other for jobs well done. If you feel the need to criticize, use statements that start with "I think…." or "I feel…" Avoid criticism that starts with an accusatory "You did…," or "You always…" Be very clear about who is doing what - in the final project summary, I will ask you to jointly summarize what each of you contributed.

Think of yourselves as a team. Develop some team spirit. Name your team. Develop a unique style for your presentations and demos (e.g., team colors, a logo). Get to know each other as individuals. Discuss hobbies, career goals. If you each take the time to care about one another, then you'll do what it takes to excel as a team.

Communication: The best way to reach me or the TA is via email. The course also has a Piazza forum that is accessed from the Canvas page. We will try to answer within 24 hours, but may take longer over weekends, holidays, or due to travel, illness, or other work commitments. It is always best to start assignments early, so there is time for us to respond to questions before the due date.

Extension Requests: If you have an accommodation from Disability Services that allows for extensions on assignments, you only need to tell me that you are requesting one as per your accommodation. I automatically get a letter from them at the start of the semester. If you have a chronic health issue that sometimes interferes with your ability to attend class or finish homework on time, then you should enquire with them to see if you can get an accommodation (https://www.umass.edu/disability/). If you become sick, you can also ask for an extension, but please accompany the request with some documentation, such as a note from your doctor or UHS (if you're sick, you should at least do a telehealth screening to be sure it isn't something serious), a picture of a positive Covid or flu test or thermometer showing a fever, etc. As mentioned above, if you have a known time conflict, such as an away game, please ask in advance.

Learning Outcomes: You are a unique human being with your own history and capacities. Not a vessel to be filled or an animal to be trained. Therefore, what you learn in this class will be a unique set of concepts, capacities, and understanding. The class is structured to provide you with a set of experiences in which you can see how computer hardware works, what it can and cannot do, how the lowest abstraction layers of the software stack interact with it, and what you

can accomplish through the use of a processor with limited resources. The grading criteria are an opportunity for you to demonstrate that you have engaged fully with these experiences, and that you are able to take what you have learned and apply it in a creative manner. This is not an outcome-oriented learning environment, with a goal of ensuring that every student will answer the same question on the same test the same way. It's your responsibility to bring your whole self to the experiences, and to creatively show what you've gained from them.

Course Inclusiveness Statement: No matter who you are or how you define yourself you are welcome in this class. Each person here is a human being deserving of dignity and respect. My goal is to help you learn the subject matter in a way that you will find useful, and to help you have an enjoyable and empowering experience in doing so. It is important to keep in mind that we are all coming to this class with different backgrounds. For many, this is a first hardware course, while others have some prior experience. We are all here to learn together! There are no dumb questions! From time to time, I may enlist some students to help others in class. If I ask you to help, remember that we all have different modes of learning, and there is no stigma to be associated with needing assistance. Please reach out to me if you have any concerns.

Pronouns Policy Statement: Everyone has the right to be addressed by the name and pronouns that they use for themselves. You can indicate your preferred/chosen first name and pronouns on SPIRE, which appear on class rosters. I will do my best to ensure that I address you with your chosen name and pronouns. Please let me know what name and pronouns I should use for you if they are not on the roster. Please remember: A student's chosen name and pronouns are to be respected at all times in the classroom.

University Accommodation Statement: The University of Massachusetts Amherst is committed to making reasonable, effective and appropriate accommodations to meet the needs of students with disabilities and help create a barrier-free campus. If you have a disability and require accommodations, please register with Disability Services to have an accommodation letter sent to your faculty. Information on services and materials for registering are also available on the University of Massachusetts Amherst Disability Services page

University Academic Honesty Statement: Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst. Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to: cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair. Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent (http://www.umass.edu/dean_students/codeofconduct/acadhonesty/).

University Title IX Policy Statement: In accordance with Title IX of the Education Amendments of 1972 that prohibits gender-based discrimination in educational settings that receive federal funds, the University of Massachusetts Amherst is committed to providing a safe learning environment for all students, free from all forms of discrimination, including sexual assault, sexual harassment, domestic violence, dating violence, stalking, and retaliation. This includes interactions in person or online through digital platforms and social media. Title IX also protects against discrimination on the basis of pregnancy, childbirth, false pregnancy, miscarriage, abortion, or related conditions, including recovery. There are resources here on campus to support you. A summary of the available Title IX resources (confidential and non-confidential) can be found at the following link: https://www.umass.edu/titleix/resources. You do not need to make a formal report to access them. If you need immediate support, you are not alone. Free and confidential support is available 24 hours a day / 7 days a week / 365 days a year at the SASA Hotline 413-545-0800.