EE 193-05: Networked Embedded Systems

Spring 2024
T/Th 10:30-11:45am Anderson 309

Get ready to build things.

This course is a learning-by-doing introduction to building and deploying real embedded systems. We'll talk about some theory — how to pick a microcontroller, comparisons between various wireless protocols for IoT, software development patterns for embedded devices, and more. But the bulk of your learning will come from working in a team to rapidly develop and iterate on a real embedded system. For Spring 2024, we will be developing a wireless sensor network to gather dense temperature readings across campus. If all goes well, by mid-May we will have built over 30 sensor nodes from scratch, deployed them across campus, developed dashboards to monitor them, and dealt with enough failures to get a glimpse of working with IoT at scale.

After successfully completing this course, you will be able to:

- Design and assemble a complete embedded system:
  - Choose components based on engineering design criteria (performance, capacity, size, cost, etc)
  - Develop and maintain a complete bill of materials
  - Design, manufacture, and assemble a printed circuit board
- Manage power consumption on a battery-powered device.
- Write driver code by reading a protocol description or register map in a datasheet (rather than googling for an Arduino library).
- Compare and contrast various wireless protocols (BLE, WiFi, LoRa, SigFox, LTE/5G) and explain when to use each.
- Address challenges related to running embedded/IoT devices in the field, such as OTA updates and remote debugging.

Where should I look for information?

The **course website** will be the hub for the course schedule and various documents: http://www.ece.tufts.edu/ee/193AES/

There is no "textbook" for the course. We will use several books which are available online through Tisch library, along with some free books, podcasts, and blog posts.

Announcements and course communication will happen on **Slack**. The invite link will be sent via email on the first day of class.

We will make heavy use of **Github** and **Google Drive** for collaboration.

How will you and I evaluate your progress?

During the first few weeks of the semester (Iteration 1), you will complete a series of projects individually to build up some fundamental skills:

- Designing and manufacturing a circuit board
- Writing a driver from scratch to read values from a sensor
- Gathering some performance metrics with your microcontroller

- Analyzing the power consumption of a microcontroller

I don't intend to assign numerical grades to your work; this would be tedious and arbitrary and ruin half the fun of building things. Instead, we will discuss the results of your work in class so that everyone can learn from the different approaches and converge on some best practices. And you'll know if you did a good job, because your stuff will work. You don't need a grade to tell you that.

In the second part of the semester (Iterations 2 and 3), you'll be working closely with your team to build and deploy a set of sensor nodes. We'll have multiple design reviews where your team will present your proposed design and respond to questions from myself and the class. Again, I won't be scoring the design reviews; the focus is on learning from each other and refining your work so that you end up with a functional product.

Overall project grading is roughly summarized below. I expect that all teams will achieve an A.

- **A**: Your project is complete and functional. There are still some issues to fix, but successful prototyping and iteration have led to a refined design with a certain amount of polish. Your documentation is detailed enough that a future team could pick up your work and build on it.
- **B**: Your project is mostly functional. It feels hacked together rather than refined. Your documentation provides some helpful information, but future teams would probably just learn from it rather than build on it.
- **C**: Your project does not work or is missing major functionality. Your documentation is disorganized or incomplete to the point that future teams would probably ignore it and start over from scratch.

Finally, you will have multiple opportunities to receive feedback from your teammates, and to comment to me about whether your team is functioning effectively.

What else do I need to know?

Instructor:

Steven Bell `sbell@ece.tufts.edu`

Curtis 001-C (Directions: http://www.ece.tufts.edu/en/1EK/finding_my_office.html)

Office hours:

- Tuesday and Thursday after class
- Wednesdays 1-4pm, Nolop makerspace
- I'm also available other times by appointment, just email me and we can find a time! I'll generally be on campus Tues/Wed/Thurs, and available by Zoom on Mondays and some Fridays.

To minimize distraction, I generally only check email a few times a day. However, I will make a strong effort to answer all messages within 24 hours on weekdays.

Prerequisites:

This will be a highly collaborative course, and I expect that each person will bring different skills to the project. At a minimum you should have some experience with writing code and building circuits, but we're looking for a mix of expertise in software, circuit design, mechanical design, and web development. You must have junior, senior, or graduate standing to take the course.

If you have any questions about prerequisites or your ability to succeed in this course, please talk to me.

Late policy:

This course is going to move fast: we're planning to build three iterations of a design in a semester, roughly one every four or five weeks. And because it is highly collaborative, it is likely that falling behind on your part will delay your teammates.

If you don't think you will be able to turn in an assignment on time, you must send me a message via email or Slack *before the deadline* and tell me the date by which you will submit it. If your teammates are depending on your work, then you should include them on the message as well. I will grant any reasonable request, but I will hold you to the new deadline you specify.

Academic integrity:

I expect you to behave like a professional in this course, which (at the very minimum) means not doing anything that would get you fired or get the company sued. You will be learning from many external resources and building on code that others have written, so you need to be very explicit about what work is your own and what you have borrowed from other sources.

Specifically:

- You need to document where you got information. This can take different forms — perhaps a URL in your code referencing a StackOverflow answer, a citation in your formal documentation, or maybe just a note in your build journal.
- Any source code that you incorporate into your design must have a documented license which would permit you to sell the product without releasing all of your source code. (We'll discuss this more in class!)
- Collaboration is highly encouraged, but mooching off of your classmates is not acceptable. Your future employer wants you to collaborate with your colleages. Your future employer does not want you to ask your colleagues to do your work for you.

ADA accomodations:

If you need special accommodations (flexibility with certain deadlines, larger type on handouts, etc.), please initiate the process with the Tufts StAAR center (https://students.tufts.edu/staar-center/accessibility-services). Accomodations cannot be granted retroactively, so please do this sooner rather than later.