Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #47812

Electronic System Design: A Hands-on Course on Creating a Professional Electronic Product

Dr. Ilya Mikhelson, Northwestern University

Dr. Ilya Mikhelson is an Associate Professor of Instruction in Electrical and Computer Engineering at Northwestern University. His technical work focuses on signal processing and embedded systems. Besides teaching, Dr. Mikhelson has worked with dozens of students on independent projects, and has created 3 brand new, projects-heavy courses in Electrical Engineering. He is constantly experimenting with pedagogical innovations, and is very passionate about engineering education.

Electronic System Design: A Hands-on Course on Creating a Professional Electronic Product

Abstract

A new curriculum has been developed for an advanced embedded systems course. In this course, students construct a webcam from the ground up, resulting in a professional, aesthetic product. Along the way, they practice surface-mount soldering, PCB design, web design, 3D design, and embedded programming. The course is a guided journey in creating a device at the level of a minimal viable product at a startup company, and feeds into a followup class where students pursue the journey without constraints, with novel projects. To assess the efficacy of the curriculum, several markers were analyzed. First, results of anonymous surveys were reviewed. Second, anecdotal evidence was reviewed. Third, the success of students in the followup course was evaluated. Overall, the findings show that the course is effective in empowering the students to be independent designers who have valuable skills to industry.

1 Introduction

Creating a practical course is a delicate balance. On the one hand, if it is too practical, students will only learn the tools presented and will not be able to generalize. On the other hand, if it is too theoretical, students will know the foundations of practical tools but will have no practice putting that knowledge to use. Most classes in the ECE department at Northwestern University address this tradeoff by focusing on the theory, but also having labs for practical training. In the author's experience, however, this is often not enough. In order to learn what a complex project entails, students must build a complex project.

Numerous project based courses have been designed. In [1], Song and Dow created a course that offered projects to an introductory class. In [2], Mikhelson designed a course where students build a comprehensive project throughout the course of the term, also in an introductory class. In [3], Yildiz et al designed a course where students learn microcontrollers through several projects. In [4], Bell and Horowitz created a course centered around four large projects. In fact, in [5], Guo et al reviewed 76 such courses.

What makes the proposed curriculum unique, though, is the emphasis on creating a single, complete product. In fact, the course is centered around this tenet, with all classroom material stemming from the necessary facets of such design. To the author's knowledge, there is no other comparable curriculum.

In the proposed curriculum, the course focuses heavily on practice. However, theory is also emphasized in order to allow students to adjust to whatever tools a company or laboratory may demand. Furthermore, the class emphasizes the theory behind "rules of thumb" in order to allow students to make educated decisions in the design process. This approach overall helps students to foster creativity [6], as none of the practical aspects have "correct" answers, and the students gain the technical knowledge to choose their own direction amidst uncertainty.

The overall goal of the course is simple: students should be able to look at any challenge and immediately know that they could design a system to address it. A lot of this comes down to overcoming their own uncertainty and fear of the unknown. The class obviously does not teach every paradigm that might exist in embedded design. Instead, it teaches students how to approach problems systematically and how to make a plan and execute it.

The rest of the paper will show how this goal was accomplished. Section 2 will present the design of the course. Section 3 will present evidence regarding the success of the course, as well as some discussion around the findings. And Section 4 will summarize the findings and present next steps.

2 Class Design

At Northwestern University, all engineering degrees end with a capstone design class. In electrical and computer engineering, this involves two courses: one to prepare students for independent design, and one to showcase the design skills through a considerable project. The course detailed in this manuscript is the former.

The overall goal of the course is to provide a rigorous preparation for the general skills that are necessary to make a professional embedded system. To this end, there are two main components: learning and practice. In the classroom, students learn a bevy of pertinent skills, which will be detailed in Section 2.2. In the laboratory, students put their knowledge to use to create a complex, albeit guided, project, which will be detailed in Section 2.1. The laboratory component will be presented first because everything centers around it. Finally, resource use will be discussed in Section 2.3.

2.1 Laboratory Work

The idea for the course, as well as its followup capstone design course, originated from the author's experience cofounding a startup company. The startup involved creating an internet-of-things, connected embedded system. The goal of the course, then, is to prepare students to be able to create such a product on their own. To achieve this, students need to learn more than just embedded systems. They must also be comfortable with wireless communications, 3D printing, and web design. To obtain this knowledge, students build a webcam from scratch. In order to build up the project from the ground, the quarter is split into eight tasks, which will be detailed below.

The main architecture of the webcam consists of three main components: a microcontroller (SAM4S8B [7]), a camera module (OV2640 [8]), and a WiFi module (ESP32 [9]). Even though the ESP32 is capable of being the main microcontroller, we use it only as a peripheral WiFi module in order to focus on low-level programming using the hardware abstraction layer of the SAM4S [10], which most students have never seen. The OV2640 module outputs a JPEG image directly over its parallel interface, and the SAM4S has a parallel input port with DMA capabilities.

The students start by assembling breakout boards for the three main components (which the instructor created and manufactured). Then, they connect the boards using jumper wires and finalize their connections. After this, they design their own PCB and get it manufactured. While the PCB is getting manufactured, students create a website for their camera (hosted on the ESP32, which acts as a server), and also make a 3D-printed enclosure for their PCB. Finally, when the boards arrive, students assemble the whole camera and demonstrate its functionality.

2.1.1 Task 1: Component Research

For the first task, students are thrown into the deep end immediately. After one lecture discussing the class project, students are tasked with speccing out all of the components that will be used. In reality, this is a simplification of the true process, because all of the main components are specified, and operating conditions are also given. Nonetheless, this is still many students' first time diving deep into datasheets and thinking about component compatibility.

2.1.2 Task 2: Soldering

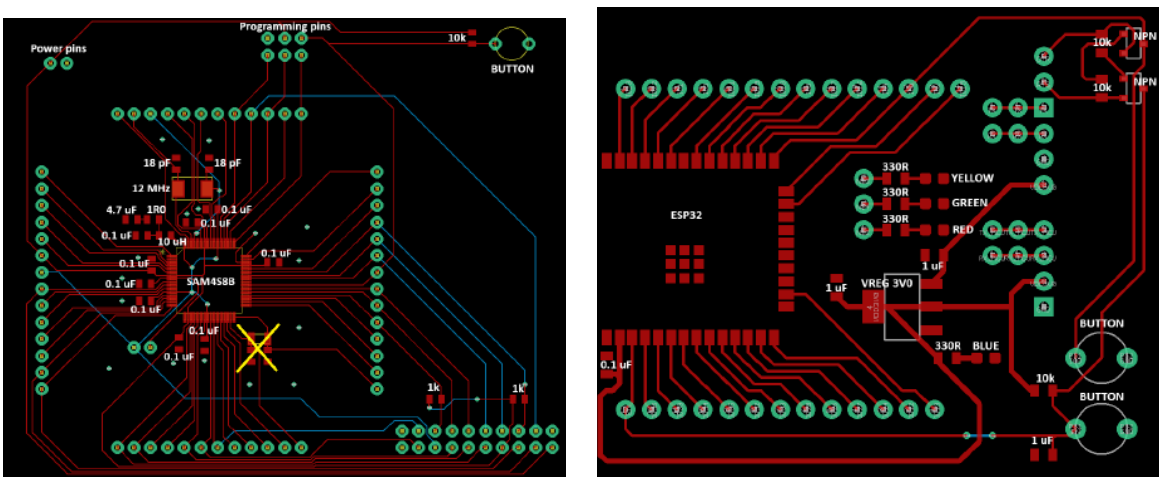
For the second task, students must demonstrate surface-mount soldering proficiency. This consists of two parts. For the first part, students must solder each of four components on a practice board. The components are a TSOP-48, an SOT-223, a QFN-20, and an 0805. For the second part, students must demonstrate understanding of proper care for laboratory equipment.

This is done by each student recording a video of them properly shutting down the equipment after soldering. This involves tinning the tip of the iron, letting the hot air gun cool down, turning off the fume extractor, and turning off all equipment.

2.1.3 Task 3: Embedded Programming Practice

While the first two tasks were just to get everyone up to speed, the third task begins the class project. The goal of the project is to program the microcontroller that runs the webcam to turn on an LED when a button is pressed, essentially a "hello world" program.

The first part of the task involves soldering. A PCB is provided to construct a breakout board for the microcontroller, shown in Fig. 1(a). Another PCB is provided to construct a breakout board for the WiFi module, shown in Fig. 1(b). Students must construct both. While this particular task does not use the WiFi chip, all of the voltage regulation happens to be on that board, so it is used purely as a power supply at this point.



(a) (b)

Figure 1: (a) Layout of microcontroller breakout PCB, (b) Layout of WiFi module breakout PCB.

One more board that must be constructed is a programming header. The microcontroller breakout board does not have a USB interface. Instead, it breaks out the native programming pins of the microcontroller. The programmer that the class uses, the J-Link [11], provides a 10-pin interface, while the microcontroller requires only six pins. Therefore, students solder another board that is an interface between the two.

Once everything is constructed, students run a test using the J-Link to verify that their soldering is correct. Once it is confirmed that the soldering is functional, students can program the button-LED code. The code itself is not very difficult, but does use various facets of the microcontroller including interrupts. The microcontroller is one that most students have never used before, and there is no wrapper like Arduino, so students have to get accustomed to its hardware abstraction layer.

2.1.4 Task 4: Full Webcam Using Breakouts

For the fourth task, students create the full webcam using breakout boards. Two of the breakout boards were already created in Task 3, so students start by creating the third one, namely the camera module. With all three breakout boards complete, students make all of the appropriate connections between them. This involves power distribution, a UART bus between the microcontroller and WiFi module for general communication, a SPI bus between the microcontroller and the WiFi module for image transfer, an I2C bus between the microcontroller and the camera module for camera configuration, and a parallel data bus between the microcontroller and camera module. Additionally, there are various control signals between the boards. For each of these, students have to dive into the datasheets of the components to figure out which pins to use.

Finally, to test their work, students run a testbench on the circuit. The testbench is provided to the students as a hex file that they upload to the microcontroller. The results of the various tests can be viewed over a serial monitor (e.g. Tera Term [12]). If the connections are all correct, every test will pass. If anything is wrong, the testbench tells students exactly which part failed. This helps them to narrow down their debugging efforts.

2.1.5 Task 5: PCB Design

For the fifth task, students focus on designing their final printed circuit board (PCB). After completing Task 4, they should be confident in their connections, even if they do not yet have the firmware for the camera completed. To create the final PCB, students combine all of the essential parts of the breakout boards. Additionally, there is a size constraint on the board so as to add an extra challenge. All designs are 2-layer boards for cheap, quick manufacturing. To complete the task, students submit their final Gerber files and the course staff places an order.

It should be noted that students are provided with all appropriate libraries for each component used in the class. That is, they do not have to create any custom library components. Since creating custom libraries is an essential part of PCB design proficiency, there is a separate subtask within this task that asks students to create a custom component, and to create a simple breakout board for this component.

2.1.6 Task 6: Website Design

For the sixth task, students create a website for their webcam. The website is hosted by the WiFi module, and when someone connects to it, it displays a live stream from the camera using websockets. In order to make this task more fun, students pretend that they are creating an entire home automation website. They make a home page with a navigation bar, where one of the options is the webcam. This allows them to get more practice with web design, while also allowing a lot of creativity.

2.1.7 Task 7: 3D Printing

For the seventh task, students create a custom enclosure for their PCB. All of the work is done in Onshape [13], an online CAD suite, which allows for easy collaboration between students and help from the instructor. There is a set of guidelines that students must adhere to, but otherwise they have free reign over their design.

After the enclosure is designed, students must print it. The laboratory for this course has four 3D printers, but there are also lots of printers around the university. By around the middle of this task, the PCBs are typically arriving, so students can try them out and make any refinements to their designs as necessary.

2.1.8 Task 8: Final Design

For the eighth task, students document their final design. This entails assembling their custom PCB and demonstrating the full operation of the webcam. This is often the hardest part of the course, even though every individual component has already been verified. The reason for this is that it is very difficult to discern a hardware issue from a software one. Poor soldering connections create strange errors. Missing traces between components necessitate workarounds. Connections between components that were accidentally inconsistent with the breakout webcam from Task 4 require software changes. It is very rare for a camera to work smoothly from the first attempt.

After everything is finally working, students document their efforts through a video walkthrough and a report. In the video, they show every detail of the operation. In the report, they talk about all of the design choices, tradeoffs, and challenges.

2.1.9 Firmware Design

Starting around six weeks into the project, around the end of Task 4, students start working on the overall firmware of the webcam. This happens in the background alongside the other tasks. Students are given comprehensive documentation and have to implement numerous functions to achieve the desired result.

The firmware orchestrates the full operation. It starts by configuring the camera module to output images in the proper format and configuring the WiFi module to accept images over SPI. Then, the WiFi module is instructed to connect to the internet, and the program waits until the connection is complete. After a connection is made, the firmware waits until a user connects to the webcam streaming website (i.e. a client opens a websocket connection to the server). Once that happens, the firmware grabs images from the camera module, writes them to the WiFi module, and streams them to the website.

While all of these operations are looping, there is an asynchronous button that allows the camera to be provisioned to a new WiFi network. The firmware must detect this press at any time and put the WiFi module into provisioning mode to allow reconfiguration.

Altogether, this work involves using polling, interrupts, and DMA. Students must write numerous custom functions and be comfortable with all levels of programming (well, perhaps excluding assembly) as they work with bit-shifts and bit-masks, hardware abstraction functions, and high-level algorithms, for example to determine the length of a JPEG image from raw bytes. They must also be comfortable with the intricacies of I2C and SPI in order to utilize those communication channels appropriately.

2.2 Lectures

The goal of the lectures is to provide the necessary background to complete all of the laboratory work, as well as to provide relevant information that goes beyond what is needed for the project. The assumption is that students are well-versed in circuits and programming already (through the prerequisites to the course).

Lectures start by giving a jump-into-the-deep-end treatment of component selection. The discussion focuses on identifying necessary components, determining their compatibility (e.g. voltages and tolerances), and determining their appropriateness in the design (e.g. power consumption, timing requirements). The instructor guides students through the speccing process by illustrating on Digikey and diving into datasheets. Most students walk away from this lecture a little overwhelmed, as they have never done such exercises previously, even though they are very familiar with all of the theory. However, through Task 1 (Section 2.1.1), students gain comfort with this process.

Next, students learn how to perform surface mount soldering through demonstration and practice. Everyone has soldered before (again, though the prerequisites), but very few have worked with such small and surface mount components. One lecture is spent discussing surface mount technology and the various tools of the trade for soldering, as well as showing a live demonstration under a close-up camera. The next class is spent in the laboratory doing a guided soldering session. This gets students up to speed for successful completion of Task 2 (Section 2.1.2).

After soldering, some lecture time is spent on oscillators. Everyone knows the concept of an oscillator, but very few know the theory. The instructor walks through the math of an unstable feedback system that creates the right conditions for oscillation, and discusses how the feedback can be made using various components such as crystals or RC networks. This is all presented in the context of the datasheet of our microcontroller, which allows users to choose between an internal RC oscillator and an external crystal. Most students come in unsure about why the datasheet calls for load capacitors alongside the crystal, and come out understanding both the internal and external components.

After this interlude, focus shifts to embedded programming. The next several lectures are spent on walking through the programming environment for our microcontroller, as well as performing various demonstrations and writing collaborative code to illustrate some core functionalities. Specifically, class time is used to write a program to blink an LED (i.e. using a pin as an output), to blink an LED when a button is pressed (i.e. introducing inputs), and using a button as an interrupt. For most students, this is the first time they get down to the level of activating individual clocks for peripheral functions, or even thinking that peripherals need a clock. This gives the necessary background to complete Task 3 (Section 2.1.3).

At this point, the next logical demonstrations would show UART, SPI, and I2C. However, many students are not familiar with their theory, so all of the theory behind these protocols is explained first. During the explanations, students also see traces of the protocols on a logic analyzer, which serves double duty by showing how to use the tool as well as how these signals look in real life. Once the theory is complete, examples are shown in the embedded programming environment.

This background is necessary to get started on the firmware design (Section 2.1.9).

Before the next big topic, another interlude is taken by presenting the details of camera sensors, image storage and transmission, and the JPEG protocol. All of these have to be understood in detail to be fully comfortable with the code that controls the image flow.

The next big topic is PCB design. The lectures start by showing how to make a schematic, progressing from simple to complex. Then, layouts are presented, again progressing from simple to complex. In the context of layouts, there is an extended discussion on signal integrity. The lectures show how signals can become corrupted, and how proper layout design can mitigate these issues. Students learn how to think from the ground up instead of memorizing rules of thumb. After learning about signal integrity, students learn how to design custom components and custom breakout boards. At this point, students are ready to complete Task 5 (Section 2.1.5).

After PCB design, lectures focus on web design. The purpose of this class is not to make students experts in front end design, so the focus is on the basics. Namely, students learn about HTML, CSS, and JavaScript, as opposed to modern frameworks. The lectures focus mostly on teaching through example, so students can practice during the lecture time as well. After this set of lectures, students can complete Task 6 (Section 2.1.6).

The last big topic is 3D design. In these lectures, students learn how to design custom parts in a computer-aided design (CAD) suite. As with web design, the goal is not to make the students experts in CAD, but rather to give them the skills to create functional prototypes for novel situations. After these lectures, students can tackle Task 7 (Section 2.1.7).

For the rest of the lectures, there are various topics. One lecture is used to demonstrate how to efficiently debug embedded systems with surface mount components. Another lecture is used to discuss ethics in embedded systems [14, 15, 16, 17]. Finally, the last lecture brings an invited speaker who has experience with embedded systems in industry to talk about their experiences.

2.3 Resource Requirements

As with any laboratory-based course, this course is resource-intensive, both in terms of equipment and time. First, this course requires some specialized equipment, namely surface-mount soldering stations complete with necessary tools (e.g. tweezers, solder paste, and flux paste). Each laboratory station also has a power supply, a 16-channel logic analyzer, and a computer with 2 monitors to aid in CAD work.

Second, there are a decent number of expendable components, namely the microcontroller, WiFi module, camera, accompanying components, and PCBs. The author also designed some PCBs specifically for surface mount soldering practice, so components for these boards also have to be replenished. Luckily, the university's laboratory budget is sufficient to cover these recurring expenses, so students do not pay anything out of pocket.

Third, the course is very time-intensive for the teaching staff. This is natural for a course where students make a full embedded system from scratch, since errors can be caused by soldering issues, hardware malfunctions, or software issues. Even with years of experience, debugging a single group's work can take a full hour. In order to make the course feasible, Northwestern University makes great use of peer mentors. Peer mentors are undergraduate or graduate students who have taken the course recently. Their main role is to hold office hours. With the support of one TA and three peer mentors, the burden becomes manageable.

3 Results and Discussion

The evaluation of the efficacy of the course is based on quantitative and qualitative feedback received over the lifetime of the course, which has run seven times so far. Every quarter, students fill out an anonymous survey at the end of the course, administered by the university. This survey contains both quantitative and qualitative evaluations of various facets of the course. Additionally, there is anecdotal qualitative feedback in the form of emails the instructor receives from alumni. Furthermore, the success of students in the followup course can be evaluated to determine how effective the course was in preparing them for unguided, complex projects. The quantitative feedback is presented in Section 3.1, and the qualitative feedback is presented in Section 3.2.

3.1 Quantitative Feedback

A summary of the quantitative ratings for the course is presented in Table 1. All ratings are on a 6 point scale, where 6 is the highest rating and 1 is the lowest. The total number of students represented in Table 1 is 159.

Table 1: Summary of quantitative feedback.

Prompt	2017 - 2024 average	2017 - 2024 median
Provide an overall rating of the course	5.7	5.7
Estimate how much you learned in the course	5.6	5.6
Rate the effectiveness of the course in challenging you intellectually	5.4	5.5

Overall, the quantitative responses have been very positive. Even though students spend a lot of time on the course, as shown in Fig. 2, they seem to enjoy it and learn a lot.

3.2 Qualitative Feedback

3.2.1 Surveys

For the qualitative feedback, Table 2 shows a few snippets of what was said on the latest feedback survey.

It should be noted that this class represents a self-selected group of students who are passionate about embedded system design, so the qualitative feedback is mostly positive. The critical feedback focuses mostly on potential restructures to the course tasks. Of note, though, is that

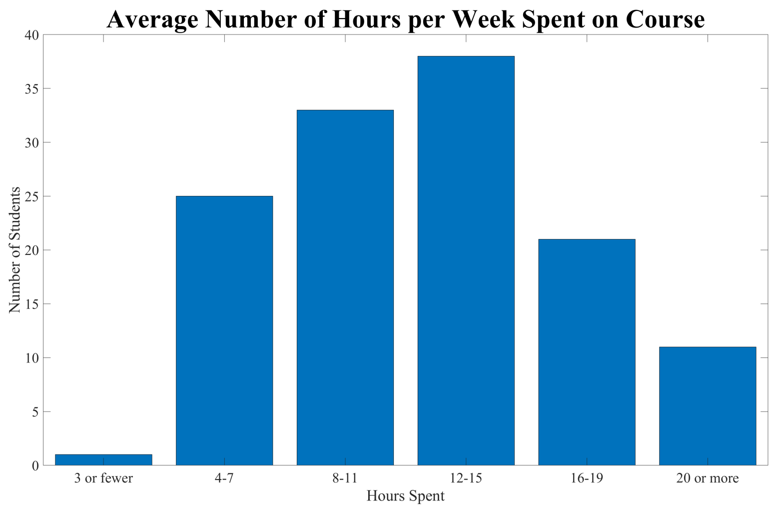


Figure 2: Average number of hours per week spent on course outside of class time.

Table 2: Samples of anonymous student feedback.

Honestly this is a class you need to take if you're planning to go into anything electronics–related.
Great class! very hands on and interesting, you learn actual skills that are useful in designing a product. at the end you have something you can be proud of.
This is one of the best EE classes at Northwestern, and I'd highly recommend anyone interested in embedded systems to take it. Even if you already have prior experience with embedded systems, you will still learn a lot.
Very interesting course, learn a lot of skills applicable to industry.
You learn a ton here – the breadth of knowledge is crazy.
Great, but time consuming class that will make you want to rip out your hair sometimes.

almost no one says that the course should be made easier or less time consuming. Students really do appreciate the debugging efforts that they must put forth, recognizing that such skills cannot be taught in the classroom, but rather have to be experienced.

3.2.2 Alumni Outreach

Over the years, the author has received several emails from former students along the same lines. Here are two excerpts:

> *"Choosing to take [this class and the followup, project-based class] was one of the best decisions I've ever made. I don't think that any other set of classes contributed more to my professional growth than these. I still keep my finished webcam from [this class] on my work desk and I look at it to remind myself of the sheer struggle of building it and making it function, and that none of my work tasks are impossible if done slowly, carefully, and the right way!"*

> *"Your classes were by far my favorites at Northwestern, and they taught me so much. In particular, the [followup, project-based class] was an absolute joy, and your guidance on [my project] helped to expand my knowledge in ways that would have otherwise been impossible. I also want to thank you in particular for your help with this project since it directly helped me get a job soon after graduation. I was asked before an interview to prepare a technical presentation of my choice, and I chose to present the project I made in your class. Everyone at the presentation absolutely loved it, and the amount of in-depth detail I had because of your class and guidance really stood out to them, and they almost immediately hired me, and with the highest college-conversion salary they've ever given!"*

Based on these emails and other anecdotal evidence (through casual conversation with alumni), the class serves students well as they move into industry beyond college. Additionally, the skills they learn in this class are easily transferable to new tools. For example, we design PCBs in a particular software, but students have told me that switching to a different one at their company was easy given what we learned.

3.2.3 Performance in Followup Class

The followup class to this one is a capstone design class. In the class, students work in small teams to complete a large project. This involves everything from component selection and sourcing, to creating necessary breakout boards, to creating a full system and iterating as necessary, to finalizing the product and making it "market-ready".

The caliber of projects that students have made has really spoken for itself. Students have made a sign language interpreter (with embedded machine learning), a persistence of vision display, smart toys for kids, a webcam that follows whoever is speaking, a food scale that automatically recognizes what is on it, and much, much more [18]. These are the same students who came into the preceding class with only a marginal understanding of how such systems work. By the end of the class, they were confident enough to tackle these challenges, and knowledgeable enough to persevere.

3.3 Challenges and Improvements

While the course is currently on its ninth iteration, there are still challenges and certainly room for improvement. First, the lecture structure is not ideal. A constant struggle is deciding what to spend time on in lecture. On the one hand, the class is very practical and therefore necessitates becoming acquainted with several tools. On the other hand, teaching students how to use a tool in the classroom sometimes seems wasteful, as they could read the documentation independently. However, if class time were instead used to dive deeper into the mathematics and theory of the phenomena that occur during embedded system design, this could also be viewed as wasteful since very little of it would be applied directly during the project. A certain balance has been reached with respect to practice versus theory, but there is always a tension between the two. Perhaps the class would benefit from a flipped structure, or a partially-flipped structure.

Another challenge in the course relates to helping students. Many issues in the project are due to improper soldering, and the only way to get better is to practice. However, it is not feasible to give detailed feedback on each student's every attempt. Therefore, it would be good to have some kind of automated feedback system that could scan the connections on the practice components to determine if they are connected well. The author has found such commercial systems, but they are prohibitively expensive, so an in-house solution would be better. As with any such endeavor, though, creating it will take time, so it is a future improvement for now.

A big challenge comes from evolving industry demands and available tools. The rise of generative AI has presented a unique challenge. For the main heft of the course, this is not a problem, because the course uses an uncommon microcontroller, so resources for it are very limited, and generative AI does a very poor job writing code for it. On the other hand, the topic of web design may need to be rethought, as generative AI generally does a great job writing the kind of web code that is taught in the class. A grander question is how web design teaching should change as a whole due to this paradigm, but as for the scope of this class, it may simply make sense to show how to use generative AI for this purpose. As that would free up some time in the curriculum, one thought is to insert a chapter on embedded machine learning or real-time operating systems, as those are very in demand currently [19, 20].

Following up on industry demands, the class would benefit from getting feedback from industry. Specifically, a structure could be put in place to follow up with alumni after a given amount of time to see how this class has benefited them directly, as opposed to relying on alumni to reach out to the instructor. This would help to better steer the class's evolution as useful facets could be reinforced and less useful parts could be dropped.

Another place for improvement is the grading structure of the course. This course is a perfect candidate for ungrading [21, 22], specifically standards-based grading [23], as the goal is to make a comprehensive project by the end. In an ideal world, the course would start by giving the class all of the Tasks (Section 2.1), and telling them they must complete them all by the end of the term. In reality, some feedback has indicated that students want even more granular tasks and more frequent due dates! The truth is, most students are not very good at independent time management, so a more structured course is beneficial to them. Therefore, the author is still thinking about how to make a push towards the ideal case while taking reality into account.

4 Conclusion

Overall, the class seems to have succeeded in its goal, which is to empower students to build anything they can imagine. Through a rigorous curriculum, students learn many skills that are directly transferable to their hobbies and jobs. Though the work is hard and the hours can be quite long, students have been motivated enough to succeed. This is evidenced strongly by the results in Section 3.

However, no course is perfect, and this is no exception. There is still plenty of room to grow and evolve, as discussed in Section 3.3. The primary focus will be to keep up with industry trends, but the other challenges can be addressed incrementally, hopefully improving the course over time.

References

[1] Jiahui Song and Douglas E Dow, "Project-based learning for electrical engineering lower-level courses," in *2016 ASEE Annual Conference & Exposition*, 2016.

[2] Ilya Mikhelson, "Introduction to electrical engineering: Empowering and motivating students through laboratory-focused teaching," in *2024 ASEE Annual Conference & Exposition*, 2024.

[3] Faruk Yildiz, Reg Recayi Pecen, and Keith L Coogler, "An industry related project-based microcontroller course," in *2013 ASEE Annual Conference & Exposition*, 2013, pp. 23–171.

[4] Steven Bell and Mark Horowitz, "Rethinking non-major circuits pedagogy for improved motivation," in *2018 ASEE Annual Conference & Exposition*, 2018.

[5] Pengyue Guo, Nadira Saab, Lysanne S Post, and Wilfried Admiraal, "A review of project-based learning in higher education: Student outcomes and measures," *International journal of educational research*, vol. 102, pp. 101586, 2020.

[6] Zhiqiang Liu and Dieter J Schonwetter, "Teaching creativity in engineering," *International Journal of Engineering Education*, vol. 20, no. 5, pp. 801–808, 2004.

[7] "Atsam4s8b," `https://www.microchip.com/en-us/product/atsam4s8b`, Accessed: 2025-01-13.

[8] "Omnivision takes lead in mobile cameras with new 2-megapixel sensor," `https://www.ovt.com/press-releases/omnivision-takes-lead-in-mobile-cameras-with-new-2-megapixel-sensor/`, Accessed: 2025-01-13.

[9] "Esp32," `https://www.espressif.com/en/products/socs/esp32`, Accessed: 2025-01-13.

[10] "Advanced software framework (asf)," `https://www.microchip.com/en-us/tools-resources/develop/libraries/advanced-software-framework`, Accessed: 2025-01-13.

[11] "J-link debug probes," `https://www.segger.com/products/debug-probes/j-link/`, Accessed: 2025-01-13.

[12] "Tera term home page," `https://teratermproject.github.io/index-en.html`, Accessed: 2025-01-13.

[13] "Onshape," `https://www.onshape.com/en/`, Accessed: 2025-01-13.

[14] Daniel W Tigard, Maximilian Braun, Svenja Breuer, Amelia Fiske, Stuart McLennan, and Alena Buyx, "Embedded ethics and the "soft impacts" of technology," *Bulletin of Science, Technology & Society*, vol. 44, no. 3-4, pp. 73–83, 2024.

[15] Sudeep Pasricha, "Ethics in computing education: Challenges and experience with embedded ethics," in *Proceedings of the Great Lakes Symposium on VLSI 2023*, 2023, pp. 653–658.

[16] Charles B Fleddermann, "Engineering ethics cases for electrical and computer engineering students," *IEEE Transactions on Education*, vol. 43, no. 3, pp. 284–287, 2000.

[17] Mahsa Ghorbani, Anthony A Maciejewski, Thomas J Siller, Edwin KP Chong, Pinar Omur-Ozbek, and Rebecca A Atadero, "Incorporating ethics education into an electrical and computer engineering undergraduate program," in *2018 ASEE Annual Conference & Exposition*, 2018.

[18] "Esp32 projects from northwestern university's embedded electronics class," `https://hackaday.com/2022/12/12/esp32-projects-from-northwestern-universitys-embedded-electronics-class/`, Accessed: 2025-01-13.

[19] "Embedded ai market size, share, trends, industry analysis report: By offering (hardware, software, and services), data type, industry verticals, and region (north america, europe, asia pacific, latin america, and middle east & africa) – market forecast, 2024–2032," *Polaris Market Research*, 2024.

[20] "Real-time operating system market size - by offering, by application, forecast, 2024 - 2032," *Global Market Insights*, 2024.

[21] Lorenzo Moreno-Ruiz, D Castellanos-Nieves, B Popescu Braileanu, EJ González-González, JL Sánchez-De La Rosa, CLO Groenwald, and CS González-González, "Combining flipped classroom, project-based learning, and formative assessment strategies in engineering studies," *International Journal of Engineering Education*, vol. 35, no. 6, pp. 1673–1683, 2019.

[22] Michael Trevisan, Denny Davis, Steven Beyerlein, Phillip Thompson, and Olakunle Harrison, "A review of literature on assessment practices in capstone engineering design courses: Implications for formative assessment," in *2006 Annual Conference & Exposition*, 2006, pp. 11–112.

[23] Danielle L Iamarino, "The benefits of standards-based grading: A critical evaluation of modern grading practices," *Current Issues in Education*, vol. 17, no. 2, 2014.