# A Modern Wearable Devices Course for Computer Science Undergraduates

Chris Gregg Raewyn Duvall Kate Wasynczuk

Department of Computer Science
Tufts University
161 College Avenue
Medford, MA 02155

## ABSTRACT

A problem that many tech companies face today is that many computer science students entering the work force lack fundamental skills for understanding the entire process of a system that is not solely software. Some students may take a series of courses on analog and/or digital circuits, but the integration with modern devices is sorely missing from most curricula. We designed the Tufts University Comp 50: Wearable Devices course to introduce the basics of digital and analog circuits to students with software-driven backgrounds by studying the intricacies of the production of wearable electronic devices. The course focused on the skills needed to design hardware, software, and a chassis for a final wearable product that was novel and potentially marketable. The primary objective was to provide a course that serves as an introduction to digital electronics but with a tangible goal to produce a high-fidelity prototype that student teams presented at the end of the semester. Given the nature of modern wearable devices, which are small, energy efficient, and strongly favor connectivity to other devices, we developed the curriculum around designing a surface-mount Printed Circuit Board (PCB), and we outfitted the student kits with coin-cell battery powered, Bluetooth-connected, Arduino-compatible devices that they needed to learn how to program and connect. We also integrated iOS development into the course so that students' final projects could communicate with both their phones or tablets, or to the Internet via these devices. As the "wearble devices" field is relatively new, this paper discusses the decisions we made for the set-up of this class, what worked and what did not, and what we would change and improve when we teach it again.

## CCS Concepts

•**Applied computing** → **Education;** *Collaborative learning;* •**Computer systems organization** → **Embedded and cyber-physical systems;** •**Hardware** → **Integrated circuits;** *PCB design and layout;*

## Keywords

ACM proceedings; Wearable Devices; Digital Electronics, Human Computer Interaction, Arduino, Mobile Computing

*SIGCSE '17 Seattle, WA USA*

© 2016 ACM. ISBN 123-4567-24-567/08/06...$15.00

DOI: 10.475/123_4

## 1. INTRODUCTION

In the last decade, *wearable devices* have become ubiquitous. Examples include GPS watches and other wrist-worn fitness devices (e.g., the FitBit[1]), microprocessor-based glasses (e.g., Google Glass[2]), wearable cameras (e.g., GoPro[3]), and microprocessor-embedded clothing ("smart clothes"). Many of these devices also include wireless Bluetooth[4] or WiFi[5] technology so they can be connected in real-time to a smartphone or to a computer. The market has exploded for wearable devices, and companies need engineers who can understand the requirements for wearable technology in order to design and build these devices. All of these devices have significant hardware and software components. Until recently, however, the tools necessary to build and design prototypes of these sorts of devices has been both expensive and arcane, and frequently required an extremely low-level understanding at both the hardware and software level. Creating PCBs and enclosures for such devices required access to expensive fabrication tools that were generally out of reach of non-corporate entities.

Within the last five years, however, the onset of easily programmable, miniature microprocessors and micro-formfactor computers has taken the electronics hobbyist community by storm. Combined with inexpensive and widely-available fabrication tools, it has become possible for a hobbyist to rapidly prototype a device and to iterate on the design reasonably quickly. Both the Arduino[6] and Raspberry Pi[7] devices have inspired a tremendous number of free or inexpensive software tools and hardware components, and open Software Development Kits (SDKs) for smartphone devices have enabled easy integration between small microprocessors, users' smartphones, and the Internet. Finally, PCB fabrication has become extremely inexpensive through online fabrication shops that produce PCB runs specifically for the hobbyist market. For example, we were able to have three 1inch×1inch PCBs fabricated for each student team for a cost of $5 per team.

With this in mind, we decided to design a semester-long course based on learning how to completely design a wearable device from PCB hardware design up to high-level mobile device integration through software. We wanted a course that could modernize learning about digital circuits, and we also wanted the course to demonstrate the importance of system engineering to a complete design.

## 2. COURSE OVERVIEW AND LOGISTICS

When the instructors sat down to decide the structure of the course, we set out to answer the following questions (broad answers in parentheses):

- What do we want students to learn? (Wearable device design from digital circuits and rudimentary PCB design through software integration with mobile devices)

- In what order should we present key concepts? (Start with digital circuits and PCB design, followed by Arduino development, culminating in iOS integration).
- What projects are most conducive to these learning objectives? (Team-based, fixed-scope projects with one or two integrated sensors).

The primary goal was that by the end of the semester, students should have an understanding of digital circuits and device design that will be useful for a modern grasp of the integration between hardware, software, and external devices in a project design. We envision this course replacing our university's required digital electronics course, giving students a more practical introduction to the course material.

Each of the instructors has a unique background. The primary instructor has degrees in Electrical Engineering and in Computer Engineering, and is a professor in the Computer Science department at the university. He primarily teaches introductory undergraduate courses, and has taught a number of project-based courses. The two other instructors and course designers were undergraduate computer science majors. One is studying in the engineering school, and she has taken both an analog and digital circuits course, as well as an Android-based design course. The other student is studying in the liberal arts school, has a background in human factors, and has self-taught herself basic digital circuit design. The instructors also had advice from a consultant who currently manages the university's maker spaces, and was the owner of a wearable devices company. The two undergraduate students were the primary course designers, and they used the course design and implementation as their own senior capstone project.

## 2.1 Syllabus and Overall Course Goals

The ambitious syllabus for the course was built around four key areas:

1. Digital electronics knowledge and design, to include designing rudimentary, small surface-mount PCBs (we targeted 1 inch × 1 inch designs) that would be fabricated externally to the university.[1]
2. Arduino programming with Low Energy Bluetooth integration.[2]
3. iOS / OS X programming. We gave the students basic code that allowed communications between an iOS / OS X device through a minimal device application, and the teams modified this code for their specific devices.
4. Device packaging design. Students were expected to use laser cutters and/or 3D printers located in the university's maker spaces to design thoughful enclosures for their final projects.

Because students were expected to already have at least two semesters of C++ programming background (as provided by our undergraduate curriculum), we spent the majority of the instruction on digital electronics theory, design, PCB layout, and soldering. When we did discuss programming (e.g., for Arduino sketches or iOS coding), these lessons were faster-paced and we expected the students to pick up on the topics more quickly. Overall, we sequenced the course instruction to cover each of the above concepts in roughly the given order, continually stressing the idea that we were building an entire integrated system.

[1]We used OSH Park[12] for PCB fabrication, with a turnaround of about two weeks on the designs.

[2]The student kits had the miniature LightBlue Bean[8] Arduino-compatible device that had an integrated Bluetooth, accelerometer, and temperature sensor.

## 2.2 Course Prerequisites

Students were not expected to have any electronics background. The course was popular during registration, with over seventy students attempting to enroll. We ended up with twenty-six students on the roster, and only four had meaningful electronics experience (although those students were helpful in providing extra instruction during group electronics design activities, and in the final projects).

The roster ended up with roughly a 50/50 male/female ratio, and about twenty percent of the class was comprised of traditionally underrepresented minority students. The students were by and large motivated and engaged in the course material, and the course was well-attended. We pitched the course to the students as a unique opportunity to learn valuable (and fun!) electronics and programming skills, and the students appreciated the opportunity to take the course. Because of the low student-to-teacher ratio, we were also able to give students a decent amount of individual (or team) attention.

## 2.3 Student Materials and Course Purchases

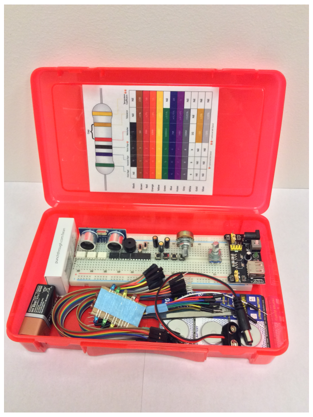

Figure 1: The Student Kit

In order to design the student materials kit, we worked with a budget of roughly \$40, which turned out to be reasonable based on bulk-purchasing the kit materials and building the kits prior to selling them to the students. The instructors made over forty individual orders for the course, which took a considerable amount of time and planning. In future versions of the course, we plan on finding a ready-built kit that has most of the materials, and augmenting the kit with a minimal amount of bulk purchases.

Our goal for the student kits was to provide a basic electronic design capability with a broad enough range of sensors and components to demonstrate a wide range of design ideas. The kits included a breadboard, power supply, and a multimeter. We also purchased extra "sensor modules" kits[9] to allow the students to experiment with a larger number of sensors, and to see what kinds of sensors are available. Table 1 lists all of the items in the students' kits, and Figure 1 shows the components in the kit without the multimeter.

The most expensive component in the kit (more than half the cost of the kit) was the LightBlue Bean (LBB) Arduino compatible device[8]. Figure 2 shows the size of the LBB relative to a breadboard. We chose this device for a number of reasons. First, it is battery operated, small ($1.79 \times 0.80 \times 0.33$ inches), and it comes with Low Energy Bluetooth, an onboard 3-axis accelerometer, tri-color LED, and an onboard external temperature sensor. Because of its Arduino compatibility, it can run the myriad of *sketches* (the Arduino term for "program") that are available to Arduino users. The LBB has eight

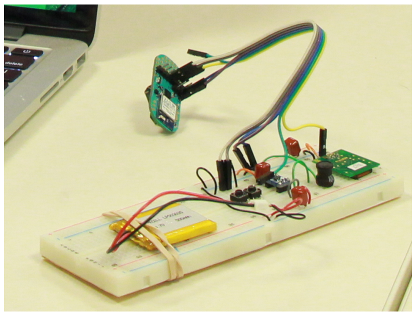

Figure 2: Light Blue Bean Connected to a Breadboard

digital input/output pins, two of which can be used for analog input. Furthermore, the LBB has a free iOS / OS X SDK that enables relatively easy integration with iOS and OS X devices. These features made the LBB an excellent choice for wearable device design, and the students quickly acclimated to programming and designing with the LBB. We should note that LBB-like devices are becoming cheaper and more fully-featured (e.g., with integrated WiFi) at a tremendous pace, and the LBB may not be the best choice for future versions of the course.

The breadboard power supply can be seen in Figure 3. This component allows students to power both sides of the breadboard independently, with either 3.3V or 5V. Components that are compatible with Arduino devices generally require either one or the other voltage, which made it easy for students to integrate many different sensors into their designs for testing.

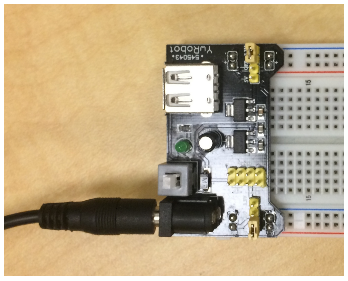

Figure 3: Breadboard Power Supply

We had the students purchase inexpensive multimeters for the course (they were less than $6 each). Although they worked reasonably well, they were not the most robust devices, and many of them did not remain completely working for the entire course. We wanted the students to be able to test and debug their projects while being away from the lab, but for future classes we would recommend paying more for better meters.

We chose the rest of the components of the kit to provide both basic parts for student projects (e.g., switches, buttons, wires), and to have sensors or components that we utilized for instructional purposes, and for projects. For example, the shift register was the basis for Project 1, and the Ultrasonic Motion detector was the basis for Project 2.

### 2.4 Classroom Setup

The course was held in an electrical engineering laboratory, which was outfitted with lab benches, high quality digital multimeters, DC power supplies, oscilloscopes, and soldering irons. The students frequently used the multimeters and soldering irons, and we provided instruction on the other equipment, as well. All course instruction took place in the lab.

All students brought laptop computers to class, and because of our decision to use iOS and OS X tools, we augmented the lab with an extra Macintosh computer for the few students who did not own Mac computers. We originally wanted to support multiple operating systems, but the logistics was prohibitive, and the software tools for other operating systems were not as robust (or, in the case of the Chromebook OS, not available).

Students brought their kits to class, and we augmented the course materials with consumable parts such as solder, solder flux, extra wires and batteries, and surface-mount and through-hole components (e.g., resistors, capacitors, etc.).

### 2.5 Software and PCB Fabrication

All software for the class was Free and Open Source Software (FOSS). We used the Fritzing[10] Computer Aided Design (CAD) software package for virtual breadboard design, schematic design, and PCB fabrication design. Although Fritzing is not a perfect tool, it is excellent for the classroom environment and is easier to learn than a number of other electronics CAD software. Students mocked up their designs on real breadboards, transferred the designs directly to the graphical virtual breadboard in Fritzing, and then progressed through the electronic schematic design phase and directly into the PCB design phase. Fritzing also produces output that can be directly sent to the PCB fabrication house. We frequently used Fritzing breadboard diagrams in handouts, exams, and during instruction, and it was invaluable from a pedagogical standpoint.

Once the students designed their PCBs in Fritzing, the course staff uploaded and purchased the PCB fabrication directly from an online PCB design vendor[12]. The turnaround for the devices was roughly two weeks, and we had to carefully plan the final project (where the student groups had to design their own PCBs) such that the designs were ready for fabrication with enough time to finish the project, and also with enough time to quickly turn around another design if the first one failed (a not uncommon occurrence).

All Arduino sketches for the LightBlue Bean were written and debugged inside the Arduino Integrated Development Environment (IDE), and with the LBB *Bean Loader* program. Students were able to use a serial monitor for debugging and testing, and all uploading and downloading was done wirelessly via Low Energy Bluetooth.

iOS and OS X software development was done in Apple's XCode development environment. Students were able to get free developer accounts from Apple in order to test their applications on their phones and tablets.

Device enclosure designs for laser cutting and 3D printing were done with the OnShape CAD software[11], and designs were fabricated and cut at the university's maker spaces, which the students had full access to.

All of the software tools students used had varying degrees of complexity, but students are used to quickly learning new software, and as computer science majors, they readily learned the new software packages.

## 3. PEDAGOGY AND METHODS

Because there were three instructors for the course, we distributed course administration as evenly as possible. The main instructor set overall course policy, and worked directly with students for logistical concerns. One of the undergraduate instructors was responsible for desigining the assignments and creating the course web site. The other undergraduate was responsible for the majority of the assignment grading. All three instructors took turns teaching lessons on the various course topics, and the instructors had regular meetings to discuss lesson planning and course flow.

| Component | Quantity | Notes |
|---|---|---|
| LightBlue Bean | 1 | Small, Arduino compatible with Low Energy Bluetooth, Accelerometer, Temperature Sensor, and tri-color LED |
| Breadboard | 1 | Full size (5.5 cm x 17 cm) |
| Power Supply | 1 | 5V and 3V power supply that connects directly to breadboard and is powered by 9V battery or 9V wall plug |
| Multimeter | 1 | Digital and inexpensive, measures voltage, current, and resistance (including continuity buzzer) |
| Ultrasonic Transmitter | 1 | Used for Project 2; measures distance from sensor |
| Potentiometer | 1 | Analog potentiometer |
| Rotary Encoder | 1 | One of the more complex parts; was used in a lab demonstrating oscilloscopes |
| Active Buzzer | 1 | Used by some students for rudimentary music |
| Switches | 2 | Simple single-throw (connected / disconnected) |
| Tilt Switches | 3 | Used to demonstrate comparison with digital accelerometer |
| Buttons | 4 | Momentary (normally off) |
| Optocouplers | 3 | Used by students in individual projects |
| 555 Timers | 2 | Used by students in individual projects |
| Shift Register | 1 | Used for Project 1 as an example of a more advanced device |
| LEDs | 10 | Multi-color |
| Photoresistors | 3 | Some students used these for rudimentary touch sensor |
| Jumper Wires | 40 | M/M, M/F, F/F - we purchased hundreds of extra jumper wires |
| Headers | 2 | Used on first day for soldering demo – soldered to LightBlue Bean |
| 3v Batteries | 3 | Coin cell batteries to power the LightBlue Bean. We bought many extras to hand out during the course |
| 9v Battery | 1 | Used for the power supply. We bought many extras for the class. |
| Tweezers | 1 | Used for placing and holding surface mount components. |

Table 1: Student Kits

## 3.1 Lectures and Independent Learning

Lectures almost always included a hands-on component. Usually, the tasks were demonstrated during class on the projector so that students could follow along, and then experiment with the hardware on their own after the discussion. This set-up was so that they could figure out how a component worked in a system, which was vital for using it later in their projects. To supplement the lectures, students were frequently asked to watch online electronics instructional videos, or to research commercial wearable devices to discuss the properties that make them effective in the marketplace.

## 3.2 Examination

The original plan for the course included three quizzes to be used as assessment. However, due to time constraints with the topics covered, only one exam was administered at the midpoint of the semester. It covered Ohm's Law, pull-up/pull-down resistors, using a multimeter, designing basic circuits with LEDs in series and in parallel, and circuits integrated with the LightBlue Bean. In general, the students performed well on the exam, although there were a few students who were still not comfortable with the electrical-engineering nature of some of the questions, and their performance was poor. We offered multiple tutorial sessions for students who wanted to improve their electronics skills during the semester, and these sessions were well-attended.

## 3.3 Projects

Projects were longer assignments and were applications of the in-class instruction. Expectations for each project were detailed on handouts and were available on the course website.

### *3.3.1 Project 1*

Project 1 focused on understanding the basics of electronics by developing a circuit that uses a shift register to set a particular number of output LEDs to values given certain input. For this project, each student either worked alone or with one other person. The first part required that the students create a working Arduino sketch for the LBB that lit up random LEDs on a breadboard. The second part implemented the shift register into the randomizer. The final part of the project included a prototype of their device with a creative application that the students demonstrated to the class and invited guests. The students were required to include one new sensor or component on their device and they needed to research how to use the component independently. The shift register had to be integral to the final device. Examples of student projects included a decibel monitoring device, a visual digital thermometer, and various games that had player-input and visual or audible output.

### *3.3.2 Project 2*

Project 2 utilized PCBs designed by the instructors and Ultrasonic sensors to create a small, wearable device that measured the distance from the device to another object. The instructors assigned partners for this project so that students could practice working with others. The students first mocked up the instructor's pre-determined circuit in Fritzing to practice designing hardware on a computer as a preliminary step to creating a PCB and testing hardware physically. Next, the students practiced surface mount soldering on the given PCB to get the Ultrasonic circuit working. As with Project 1, the students were required to creatively use the ultrasonic sensor and circuit in a device they designed, with at least one new component or sensor. Example projects included a workout device that measured "punching speed," a device to alert a backwards-walking tour guide of an impending obstacle, and a theremin-like device with integrated headphones.

### *3.3.3 Project 3*

The final project was divided into three subsections: hardware, software, and user interface, and it spanned roughly half of the semester. The first week was spent determining projects and teams of four or five students by first having the students pitch ideas and then let us know which projects they were interested in pursuing. They then planned out their whole project: what they wanted their product to do by the end of the semester, what hardware they would use, and how they would interface hardware, software, and casing together for the final creation. Each phase had a team lead that would be in charge of making sure everything in their phase was completed. The next two weeks focused on ordering parts and understanding the hardware – teams investigated the individual components and then built a rough prototype on a breadboard. Understanding the hardware was critical because the students had to design their own PCB based on their planned circuit. The students used Fritzing to develop their PCBs, and at the end of the two weeks, the PCB designs were sent out for development. The students spent the following two weeks developing software, either Arduino sketch integration with their breadboard prototypes, and/or iOS integration, if necessary.

The PCBs started arriving around the start of the User Interface phase and teams started putting everything together. They soldered their components, developed casing for their products to be wearable and attractive, and finalized their applications to be intuitive. They also planned and implemented user testing (see below for details), changed a few small details to incorporate user feedback, and finally presented in front of a panel of judges at a local Maker Space, Artisan's Asylum.

Each phase had deliverables including working prototypes at that point in development, continually updated Fritzing diagrams, and a write-up about project progress and how to use their device at each stage.

## 4. USER TESTING

With about a week before their final presentations, teams were required to develop a user testing strategy to analyze their designs with a group of random test users. We advertised the test date and location, and each team was able to test their device with students and faculty who attended the event.

Most surprisingly, all of the teams modified their product in some way (e.g., updated software, additional hardware, better package design) based on the feedback they received from this testing. A common course evaluation comment was that the user testing day was extremely helpful to teams in their final push to finish the products.

## 5. RESULTS

### 5.1 Lectures

At the beginning of the course, the instructors realized that the learning curve was steeper than anticipated for the electronics material, so lectures progressed more slowly than the initial calendar plan. This required more out-of-class learning, particularly through online videos and handouts. This turned out well for the students that watched the videos, but we could not control whether or not everyone completed those assignments. In a future version of the course, we may implement post-assignment quizzes to control this better.

### 5.2 Assignments and Hands-on Activities

Students frequently asked for more hands-on activities throughout the class to understand the application of certain subjects. The topics included circuit-building, software/device integration, and enclosure design (i.e., for laser cutting and 3D printing). The instructors planned a number of these activities outside of the regular curriculum, and they received feedback that the assignments were extremely helpful. Grading for these extra assignments was binary (completed or not), and it was included in the students' class participation grades.

### 5.3 Examination

The exam was administered during a single 75 minute class. The grades spanned from a 52% to a 100% and the average was an 83%, with only one failing grade. The student evaluations criticized the inclusion of only one exam, and a common complaint was that more frequent quizzes would have been more effective.

### 5.4 Projects

The projects were a huge success; most students enjoyed the projects assigned and they put in their best effort for understanding the electronics behind the work. They embraced the creativity that each project required, and this validated our initial idea that a project-based class would motivate the students to learn the electronics design knowledge as a necessity for creating the projects they envisioned.

1. Grades for Project 1 ranged from 73% to 100% and 88% of the class had above a 87%.

2. Project 2 was more in depth, but grades ranged from 83% to 113% and it was clear the students had a better understanding of circuits and wearable devices. Many groups went above-and-beyond the requirements (thus the high top grades), and they loved presenting their designs to the class and invited guests.

3. The final project grades ranged from 74% to 99% and the instructors were extremely pleased with the results. Each team created a high-fidelity prototype by the end of the semester and the class presented their projects at Artisan's Asylum, a local community Maker Space, where they were judged by maker space members. The judges were highly impressed with the work the students had completed. We videotaped the presentations and the teams were required to set up table displays for their devices for a post-presentation session where maker space members could ask them specific questions and see more in-depth demonstrations.

## 6. ANALYSIS

The Wearable Devices course we designed and taught was a successful realization of our original goals. In particular, our students learned a worthwhile subset of digital electronics theory and practice to replace a full-semester theory course, given that they are computer science majors. That is to say that while we did not turn any of our students into electrical engineers, we are confident that they have a basic understanding of digital circuitry to effectively work on a team where they will interface with electrical engineers. They understand many of the challenges of circuit design, and their exposure to PCB design gave them an appreciation of what is an isn't possible in an electronic device. Furthermore, we would argue that our holistic, project-based course solidified their understanding of the entire process of integrating hardware and software into a true project, and that these skills are invaluable for computer science students in today's world.

### 6.1 Lectures

Lectures were generally a success once the work-flow was understood. Because there were three instructors, it would have been more efficient to either assign one instructor to lectures or to have a very detailed plan of who is going to cover what and what exactly should go on the handouts to most mirror the lesson. The handouts, posted before class on the course website, seemed to work the best as a teaching tool for lectures, and in future iterations of this class, more handouts would be useful.

### 6.2 Assignments

In terms of learning the material, the in-class activities were the most helpful for the students. When they could follow along and see how components interacted in a circuit, the concepts stuck a lot more than just seeing the theory of each component on the projector. During this time, students who had more background in electronics would help those who had little to no previous knowledge of this technology.

### 6.3 Projects

This was where the real learning occurred – when the students had to figure out components on their own and only come to the instructors when completely stuck. These projects really promoted independent learning and the use of online resources. For example, once we taught students how to read electronic component data sheets, they readily researched the data sheets online for components they had not seen in class.

#### 6.3.1 Project 1

This project gave the instructors a good idea of where students were at the beginning of the semester, and whether the initial instruction was enough for students to work with basic electronics. The guidelines were very specific to make sure everyone had a sold groundwork to build upon for the project, but there was still room for creativity. Given the students performance, it was a successful first project.

#### 6.3.2 Project 2

This project was much more open-ended and students had a harder time not having strict guidelines to follow. The instructors' views varied between "This is college, they should learn to not require hand holding" to "yes but they won't finish the project successfully without more intervention". The final projects varied and most were good representations of the students' effort to create an actual product rather than just a class assignment. While the students may have not expected as much autonomy for this project, they were successful. They all learned about debugging hardware, creating a viable product with little input from the customer, and working with digital electronics. Overall, the educators were very impressed with how the work turned out and each team presented their projects to the class.

#### 6.3.3 Project 3

This project spanned roughly half the course and was the most important to the original course goals. The students all worked exceptionally hard to create viable products and had excellent presentations at a large local maker space. Though many teams experienced the frustrations of hardware design, they all were satisfied with their final products. The teams realized that device design is iterative and that bringing together each of the components is significantly harder than it initially seems. They also learned how to overcome project difficulties based on a deadline. One team even pitched a product that did not work at all (though it had been working earlier that week), but they improvised well it enough that the judges from the maker space were suitably impressed.

After the course ended, two of the teams began working with the university's technology transfer office to investigate the possibility of patenting their design and looking for funding to commercialize the design.

## 7. TEAM FINAL PROJECTS

There were six teams, made up of between three and five students each. Each team designated one student to lead each of the three project areas: hardware, software, and design. The final projects were presented at the Artisan's Asylum maker space, and they were assessed by a panel of judges made up of the course instructors and Artisan's Asylum members. This assessment was part of the students' final grades.

The authors are happy to provide descriptions and photographs of all of the final projects, but given space considerations we will only elaborate on one project, *SunEmoji*, here.

### 7.1 SunEmoji

Team *SunEmoji* described their project as follows: "Our device is an everyday, unobtrusive wearable suitable for all people, but especially those susceptible to sunburns (e.g. children). The device will not only track and display your sun exposure over time but also train you to understand how the sun affects your body so you can practice good sun protection habits even without the device. When the device determines that it is time to re-apply sunscreen, it will notify the user via audible and visual alerts and via text message to a mobile phone."

The *SunEmoji* device used the LBB, and iPhone, and a custom PCB circuit to alert wearers when to re-apply sunscreen. The team purchased a UV-light sensitive sensor, and integrated the device into a circuit that would fit on an armband to be worn by a child. The device had a 1inch×1inch display and a speaker that would play a set of tones when it is time to re-apply sunscreen. Additionally, the team wrote an iPhone application that received a pop-up text from the device to (for instance) notify a parent, as well.

## 8. CONCLUSION

We designed our Wearable Devices course to introduce computer science students to digital electronics in a modern, fun, rewarding, and useful way. Students who took the course gained knowledge and skills that will impress future employers, and the students understand many of the details of designing and producing small, energy efficient, highly connected wearable devices. The students enjoyed the course, and we very much enjoyed teaching it.

We plan on repeating this course, and it eventually may replace our university's current digital electronics course for comptuer science engineering majors.

The biggest take away from this course has been that a version of this class is vital to current CS curricula in universities. Computer Science students should understand the entire process of developing a product and the course developed here was an effective way of doing just that. Wearable devices can be simple enough projects do complete in one semester and are a great conduit for learning basic Electrical and Computer Engineering.

## 9. REFERENCES

[1] *Fitbit Official Site for Activity Trackers & More*, https://www.fitbit.com Accessed on Jun 1, 2016.

[2] *Google Glass*, https://www.google.com/glass/start/ Accessed on Jun 1, 2016.

[3] *GoPro Official Website - Capture + share your world*, https://gopro.com Accessed on Jun 1, 2016.

[4] *Bluetooth Technology Website*, https://www.bluetooth.com Accessed on Jun 1, 2016.

[5] *WiFi Alliance*, http://www.wi-fi.org Accessed on Jun 1, 2016.

[6] *Arduino*, https://www.arduino.cc Accessed on Jun 1, 2016.

[7] *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi*, https://www.raspberrypi.org Accessed on Jun 1, 2016.

[8] *LightBlue Bean - Punch Through Design*, https://punchthrough.com/bean Accessed on Jun 1, 2016.

[9] *Arduino Sensor Kit (example)*, https://www.elektor.com/arduino-sensor-kit Accessed on Aug 16, 2016.

[10] *Fritzing*, http://fritzing.org/home/ Accessed on Jun 1, 2016.

[11] *OnShape*, https://www.onshape.com Accessed on Jun 1, 2016.

[12] *OSH Park*, https://oshpark.com Accessed on August 15, 2016.