Avoiding the Syntax: An Accessible Approach for Introducing First-Year Engineering Students to Microcontrollers

Will Sierzputowski
Department of Engineering Education
Ohio Northern University
Ada, OH 45810
Email: w-sierzputowski@onu.edu

Karli Katterle
Department of Engineering Education
Ohio Northern University
Ada, OH 45810
Email: k-katterle.1@onu.edu

Josephine Palmer
Department of Engineering Education
Ohio Northern University
Ada, OH 45810
Email: j-palmer.4@onu.edu

Tylar Dazey
Department of Engineering Education
Ohio Northern University
Ada, OH 45810
Email: t-dazey@onu.edu

Dr. Todd France
Department of Engineering Education
Ohio Northern University
Ada, OH 45810
Email: t-france.1@onu.edu

Introduction

At Ohio Northern University, all incoming engineering students are required to complete a two-course introductory engineering experience. This sequence gives all students, regardless of specific degree path, a toolbox of skills that all engineers need. This course takes place before most other engineering content courses, effectively making it many students' first foray into learning practical engineering skills in areas such as utilizing the design process, prototyping, and presenting in front of both technical and non-technical audiences. These two courses attempt to have students develop a universal language that they can use in future interactions with different types of engineers and to give students a baseline understanding of the tools, skills and knowledge that are used across all the engineering disciplines.

Giving all students a baseline understanding of tools, skills and knowledge serves two main purposes. First, it consolidates key concepts all engineers are expected to understand, limiting the college's duplicated efforts across departments. For example, students learn the basics of Excel, technical report writing, and CAD modeling. Additionally, students are taught critical components of the design process, including idea generation, concept reduction, and prototyping. These classes also serve to teach students to compromise and overcome conflict when on teams.

Many engineering students enter college unsure of the specific discipline that is the best fit for them; exposure to various engineering disciplines in this first-year course sequence helps ensure that students pursue the path in which they are most interested. This exposure also gives students valuable interdisciplinary experiences, a recommendation that has become more common among post-secondary educators hoping to limit the effect of degree-specific "silos" at the college level. It is important that all engineers know enough about other disciplines so as to create a common language and foster effective teamwork for years to come. Few engineering problems exist solely within one discipline, and the interconnection of all fields is more vital than it ever has been.

In past iterations of the course, there has been an unintended prevalence of mechanical engineering content, so to better balance the courses, a new module was created that heavily focuses on electrical and computer engineering. The mechanical focus is concerning given the research of Steiner. His work shows a student preference to performing group roles that they are already comfortable with. This can be extended to topic area of decisions as well [1]. A student who has no experience with circuits (many students that are entering college) will be less likely to want to pursue a solution involving circuits and this becomes a cyclical problem.

This module consisted of 5 days devoted to the basic concepts of programming and circuit design using a microcontroller kit (the Elegoo EL-KIT-003 UNO Project Super Starter Kit based on the Arduino Uno). This module began with procedural learning (i.e., following specific instructions) and transitioned to student exploration into the design and creation of feedback-based systems. Due to the limited available time and activity goals – which emphasized understanding the structure of coding and familiarity with if-then statements rather than the syntax of a specific program – instruction was given using a free online drag-and-drop programming environment. This let students and instructors alike focus on the logical thought process involved in programing, allowing these students to more easily transition into standard programming. Removing the element of syntax greatly increased the work that students were able to accomplish in the time frame and with skill sets they already had. This approach is supported by Schunn[2], mitigating the traditional "basics first" engineering curriculum model, instead diving quickly into exploration and design as a motivator to learn foundational skills.

Lesson Design:

In the interest of providing a positive introductory experience in the fields of electrical and computer engineering, salient topics were chosen for the microcontroller module. Immediately preceding this module, a separate lab was run focusing various resistor types (carbon, light-dependent, and thermistors). With this foundational knowledge, the module dove into the exploration of variable resistors and using them as digital sensors. Except for resistors and making circuits only using resistors, any further knowledge of circuits and their design was not taught (although many students knew more from high school courses).

Before the creation of the module, it was known that the first-year students had very different educational backgrounds. It would thus be inefficient and tedious for students with previous microcontroller experience to go through multiple days of lecture based introductory lessons. This choice is supported by the research of Recktenwald which reflects a strong student preference to project based learning [3]. At the same time lecturing would not give the students an opportunity to properly explore the microcontroller and its various components. With these ideas in mind, procedural, self-directed activities were prepared. It was estimated that the procedural component of the module would take students 3 days to complete on average, leaving 2 additional days for further exploration and design.

In order to allow students to program and understand programmatic flow and logic without having to teach them the syntax in a more traditional language, an online program called TinkerCAD [4] was used. One component of TinkerCAD is a drag-and-drop programming language that automatically generates C++ code that can be compiled and ran on an Arduino. Students could thus program in a visual interface, simply copy the translated code into the official Arduino IDE, Integrated Development Environment, then compile and upload it to the Arduino. This syntax workaround greatly reduces the background knowledge needed to program and allows all effort to be placed into understanding order and flow of the program.

The first day consisted of introducing students to the Arduino Uno that serves as the microcontroller in the selected kit. The use of various input/output (I/O) pins was explained and the concepts of Voltage source and ground pins was introduced. The closed-loop nature of functioning circuits was reiterated, as was the effect resistors had on voltage. Then students made a simple button powered LED circuit from both a physical picture of the circuit and a schematic drawing. Notably, students were genuinely excited after seeing a single LED turn on when they depressed the button. Students were then thrown directly into programming.

The first program the students created used the serial communication ability of an Arduino to send simple messages to a computer (Arduinos have a built-in library for easily communicating to a device that supports serial communication, which in this case is used to send text based information to the connected computer). Upon learning how the basic communication worked, students then were led through the construction of simple digital circuits that used a button connected to a digital I/O pin as an input. Students' programs read the value of the pin in question, then wrote this read value using the serial communication so they could visibly see what the Arduino was reading from the pin when the button was pressed (0 when not pressed, 1 when pressed).

In order to teach students how to use analog input pins, they constructed a simple circuit that used a voltage divider to change the voltage drop across a variable resistive element such as a thermistor, photoresistor and a potentiometer. They changed conditions as appropriate for each element and saw how the value the Arduino was reading was changing based on these changes.

They recorded these values on a chart, which served as a quick reference point for instructors to check students' progress and ask questions to evaluate students' understanding of the process.

Now that students understood how input could be read, simple outputs were taught, such as from an LED and a buzzer.

Students then were introduced to the if-then statement. Using this they and created a circuit that lit an LED when a button was held.

Many of the activities place an emphasis on understanding instead of task completion by their instructors. Students were encouraged to work together and ask lots of questions when they came to obstacles. This classroom environment is similar to what Harlen states as effective [5]. Students then moved on to create systems that utilized resistive elements and analog inputs to change the state of an output. Specifically students were tasked with creating a system that utilized lights that turned on at different light value through the use of a photoresistor. Students then expanded this to have a three-state system, where there are three different trigger points and three different outputs based on the currently read value.

After these introductory lessons, students were given the opportunity to experiment with any of the kit components or functions that interested them. To help focus their efforts, they were asked to identify a specific authentic need (i.e. a temperature control unit that turns on a fan reduce the temperature being read by a sensor), then design and program a circuit to fulfill this need.

Implementation:

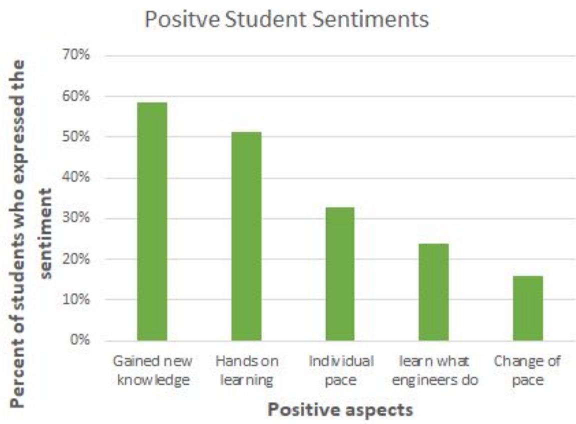
Due to the student-centered nature of this module, instructors were actively engaged, but not in the traditional lecture-style sense. However, any question a student may have had, whether to expand on what they were learning, clarify points, or seek assistance, were addressed by the instructor. To assist each instructor with handling these individual questions, upperclassmen from the electrical and computer engineering programs volunteered to help. These students' assistance was invaluable both in terms of being able to help multiple students at once as well as providing deeper technical knowledge that the first-year instructors may have lacked (these instructors had various levels of microcontroller experience themselves).

Analysis:

Before the module, students were given optional anonymous surveys to evaluate their knowledge and comfort level with circuits, microcontrollers and programming. The survey consisted of 7 Likert scale questions. One hundred four of 161 first-year students completed this pre-survey. After the module, students were again given a survey with the same 7 Likert scale questions, and also asked to give open-ended qualitative feedback on what they liked and disliked about the module. One hundred twenty-seven students filled out the post-survey.

The qualitative data was analyzed by narrowing down students' comments into several key categories. The most emergent positive "themes" (all majors combined) are shown in Figure 1.

The most commonly cited strengths were related to gaining new knowledge (58%) and the hands-on nature of the module (51%).



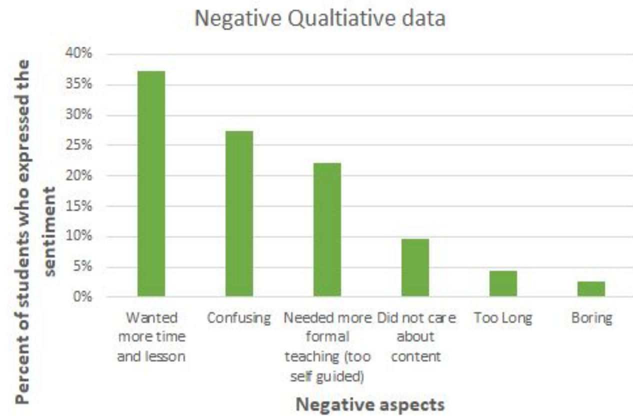


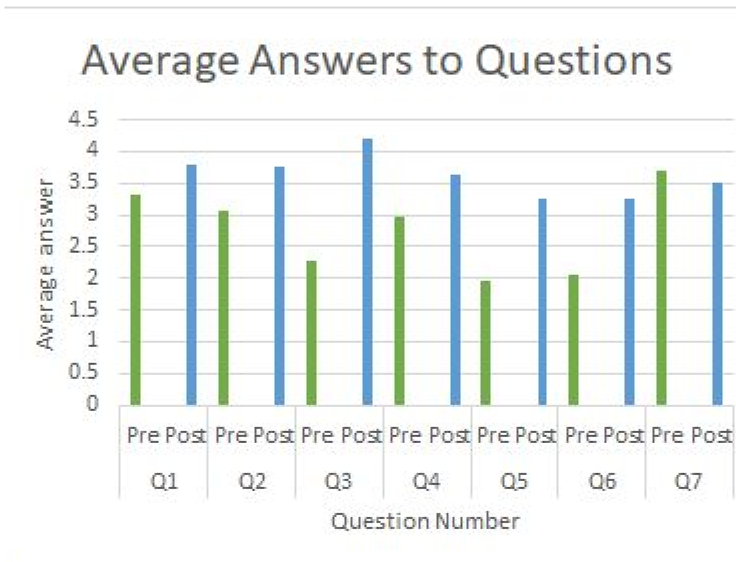
Figure 1: Positive Student Sentiments

Figure 2: Negative Student Sentiments

Figure 2 shows the negative aspects that students identified in the course. Thirty-seven percent of the students expressed that they wanted more time and lessons in the module. This may speak to the success of the lesson or may speak to the fact that students wanted more time to complete the given material. The pace of learning could have been streamlined – 27% of student expressed that the module was confusing in at least one part and thus more explanation and clearer information should have been given in sections.

Analyzing the qualitative data that was recorded from the second survey results varied drastically between majors. Eighty-six percent of computer engineering majors who provided qualitative data said that they enjoyed the hands-on nature of the modules, 50% of the electrical and 48% of the mechanicals stated shared this sentiment and only 19% of civil engineers. Similarly, 14% of computer, 17% of electrical, 19% of mechanical and 33% of civil engineers felt that the lesson was too self guided and would have preferred a more traditional classroom.

The Likert scale questions show that students, in general, gained knowledge and increased their comfort level and ability to create and program microcontroller-based systems For the average answers to the question reference Figure 3. The only question that did not positively transform was about students' interest in learning more about electrical engineering, computer engineering, and/or computer science. It appears that students, in general are less interested in learning more about these topics. This sentiment was not shared equally by all students, more students who were already majoring in Electrical or Computer engineering were equally as ready to continue learning more. It was mainly mechanical and civil engineers who had lost interest. This clearly serves to verify choice of major for these students.



Q1	I have a reasonable understanding of what electrical engineers, computer engineers, and computer scientists do.
Q2	I have a basic understanding of computer programming structure and logic (e.g., if & else statements).
Q3	I have a basic understanding of how a breadboard works.
Q4	I can read a basic circuit schematic.
Q5	I understand the basics of how an Arduino board works.
Q6	I feel confident that I could program a microcontroller and construct a circuit to complete a useful task.
Q7	I am interested in learning more about electrical engineering, computer engineering, and/or computer science.

Figure 3:Average Survey Answers

Conclusion:

After the data was analyzed it was determined that overall, the activity was effective. It taught students basic fundamentals of microcontrollers, and offered insight into electrical and computer engineering. Students gained knowledge in areas that previous iterations of the first-year engineering course lacked. Future iterations of this module will include re-evaluation of the instructional strategy employed to alleviate the confusion that some students experienced. Additionally, more time may be devoted to this content in order to represent wider array of engineering disciplines in this introductory course.

It is perplexing why many students are now less interested in learning about electrical and computer engineering. It seems reasonable, considering how much of the collected data points towards a successful and educational experience, that many students curiosity was satisfied by this module. This is not a bad thing, as this can help people affirm that they have selected correct field of study and still fulfills one of the originals goals of the course, getting students to understand what all engineers so.

The collected data gathered shows that non electrical and computer engineering students were more interested in the subject area then there other peers. In order to increase their interest level the curricula should be adapted to contain more mechanical based systems, such as using motors, to relate it to a more physical field of study.

Overall, this lesson was very highly successful in both teaching students and helping to remove the bias towards mechanical engineering at Ohio Northern University's Introduction to Engineering course. One of the most effective choices made for this module was removing the need to understand syntax by using the TinkerCAD's drag and drop programming interface. In addition to this, running the activity with hands-on experience as a major factor allowed students, of all backgrounds, to learn, experiment and generate useful circuits. This approach, backed by data, can be recommended for similar course and can help to give students a solid understanding of programmatic logic without the need to spend large amounts of time teaching syntax.

[1] Steiner, A., Hirshfield, L., Finelli, C., & Chachra, D. (June 26-29, 2016). Investigating Task Choice in First-Year Engineering Team Projects. Paper presented at the ASEE Annual Conference and Exposition, New Orleans, LA.

[2] Schunn, C. D. (Fall 2009). How Kids Learn Engineering: The Cognitive Science Perspective. The Bridge, 39(3), 32-37.

[3] Recktenwald, G. W. (June 26-29, 2016). Six Years of Living with the Lab: A Survey of Student Experience. Paper presented at the ASEE Annual Conference and Exposition, New Orleans, LA.

[4] Tinkercad. [Online]. Available: https://www.tinkercad.com/.

[5] Harlen, W. (2006). The role of assessment in developing motivation for learning. *Assessment and Learning*, 61-80.

This opportunity is made possible through support from the National Science Foundation's Robert Noyce Teacher Scholarship Program (Award #1758335)