The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #42283

Tinkercad—Not Just for Kids

Prof. Branimir Pejcinovic, Portland State University

Branimir Pejcinovic received his Ph.D. degree from the University of Massachusetts, Amherst. He is a Professor and former Associate Chair for Undergraduate Education at Portland State University, Electrical and Computer Engineering department. He has led department-wide changes in curriculum with emphasis on project- and lab-based instruction and learning. He was awarded the best paper award by the ECE division of ASEE in 2017 for his work on freshman engineering course development. His research interests are in the areas of engineering education, microwave absorber design, ferroelectrics, photovoltaics, THz sensors, signal integrity, and semiconductor device characterization, design, and simulation. He is a member of IEEE and ASEE.

Dr. Melinda Holtzman, Portland State University

Melinda Holtzman received her Ph.D. in Electrical Engineering from the University of Nevada, Reno. She is a Teaching Assistant Professor and undergraduate advisor in the ECE department at PSU.

Tinkercad - Not Just for Kids

1. Introduction

Covid-19 caused a great deal of disruption across all levels of education largely by forcing teachers and students to quickly adapt to some model of online delivery and learning. This has been especially difficult for engineering faculty and students due to the high reliance on hands-on experimentation in labs and other modes of face-to-face learning such as projects. In our case, this has meant lectures delivered over Zoom and labs done by students by themselves at home. For example, many of our electronics courses used Analog Discovery from Digilent and more advanced courses used nanoVNA-s.

In addition to this switch to "personal" instrumentation we were forced to rely on simulation much more than before Covid. This is due to many reasons, such as simplicity of use, wide availability, and use across multiple courses. The use of virtual tools in engineering education has been recognized as an effective method for providing hands-on experiences and fostering learning in diverse engineering disciplines [1,2]. Additionally, the integration of physical experimentation with virtual simulations has the potential to improve students' educational outcomes [3]. Introducing simulation tools like LTSpice from Analog Devices or Multisim from National Instruments is almost standard practice in ECE programs. However, this introduction usually happens at the earliest on the Sophomore level during the fundamentals of circuit analysis courses.

Introducing such powerful but also complex tools at the freshman level, especially for the first engineering course, is problematic. The complexity of the interface and conceptual difficulty of interpreting schematics and different types of simulations can be intimidating and demotivating for students just testing the waters in the electrical or computer engineering field.

Recently, the well-known 3-D design tool Tinkercad from Autodesk was expanded to include simulation of electrical and electronic circuits [4-7]. The software is web-based and has a fairly intuitive interface that was meant to be friendly even for middle and high school students. Over time, its simulation capabilities have expanded and now cover areas that go well into the requirements of sophomore-level circuits courses. In addition, Tinkercad now has a built-in interface to Arduino and BBC Micro:bit microcontrollers which enables programming either in their native language (C-like for Arduino) or a special graphical user interface. This now opens up opportunities for exploring not just analog and digital circuits but also microcontroller projects. In [8], authors discuss how to integrate virtual and hands-on experimentation with the focus on Arduino sensors and programming. In [9], authors are focused on simulating electrical circuits. The question for instructors now becomes how to incorporate the most effective ideas that were developed out of necessity when they return to face-to-face instruction.

2. Curricular and institutional context

Our Electrical and Computer Engineering department at Portland State University (PSU) is in a major urban center, and we serve a very diverse population, including many transfer and international students. The academic year is based on quarters, lasting 11 weeks. We have ABET accredited BS degrees in electrical engineering and computer engineering, but the first two years are almost identical for both programs.

2.1 ECE 101: Exploring Electrical Engineering

Like many other programs, we introduced a freshman course designed to introduce students to electrical and computer engineering, to engage them in some fun but educational projects, and introduce them to campus life, all without imposing strict prerequisites. In our course we want our students to gain the ability to:

1. Solve engineering problems.
2. Perform research on areas of electrical engineering.
3. Write technical reports and summaries.
4. Perform simple lab experiments.
5. Complete a project involving both design and technical elements.
6. Work on a team.

This is accomplished through a series of labs and projects. Labs introduce common lab equipment and instrumentation. Projects are team-based and can be structured, or students can pick their own. This course is followed by another two freshman courses. The first one delves in depth into problem solving and how to program microcontrollers using Python. Engineering problems and projects worked on in this course are more complex and the approach is more structured. The third course in the sequence covers C-programming and details of programming and interfacing. By the end of the first year, students have enough technical knowledge and experience in working in the labs and on teams that they can undertake a substantial project during their sophomore year that can be thought of as a mini-capstone project.

An issue we have had is the transition between the relatively easy ECE 101 and the more rigorous ECE 102. A lot of new material is introduced in ECE 102, and some students have difficulty with the pace. One way we are addressing this is to introduce some of the new material 'lightly' in ECE 101, so when students see it again in ECE 102 it is not all new. To this end, we added a lab in ECE 101 to introduce microcontrollers and their programming using Tinkercad. This lab is discussed in more detail in the Implementation section below.

2.2 PIN 101: Introduction to Computer Hardware

Based on our experience at PSU and in partnership with a large university in China, we developed a first-year course called Introduction to Computer Hardware. The course has the same goals as our ECE 101. While there are some cultural and curricular differences, the biggest challenge has been the scale: while courses at PSU typically have 40-60 students, courses in China have 240 students. This brings special logistical problems, which were exacerbated by lockdowns during Covid-19. Over the last three years, these courses were primarily taught remotely by the instructor from PSU with microcontroller related parts delivered by a local instructor.

Despite Covid restrictions, we set out to have as much hands-on experience and lab work as possible. To this end, we developed a series of five labs specifically for this course:

- Three labs are intended to introduce electrical circuits
- Two labs introduce Arduino Uno and its applications in simple projects

The first three labs are described in detail in the sections below.

Students are also expected to complete a team-based micro-project that combines two Arduino sensors or other components, followed by a larger project. Due to logistical constraints, we have to limit the selection of projects to those listed in Table I.

Table I. Projects used in introducing Arduino programming.

● Simple electronic organ, ● Control LED-based color generator, ● Music player, ● Motor speed control	● Moving object detector, ● Traffic light controller ● Variable speed desktop fan.

Obviously, we are only able to introduce the very basics of circuits and microcontroller applications, but that aligns with the goals of these courses: gaining initial experience in solving engineering problems, performing lab experiments, and completing a technical project. Simultaneously, we hope to excite and motivate students by engaging them in authentic engineering work.

In the following we will first describe Tinkercad, followed by practical examples of how we use it in different courses that introduce electrical engineering to freshmen students. Next, we will discuss some preliminary results of surveys related to students' preparedness for hardware and software projects as well as student satisfaction with the introduction of Tinkercad as a simulation tool.

3. What and why of Tinkercad

Tinkercad is a free, web browser-based app that covers 3D design, electronics, and coding [4]. It was started in 2011 as a 3D modeling tool and was acquired by Autodesk in 2013 [6]. Around 2017 electronic circuit simulation was added. While the app can be used by anyone with a web browser and internet access, it has special features that make it accessible for K-12 use, such as being Ad-free and kidSAFE certified [7].

Currently, the app supports Arduino Uno and Micro:bit microcontrollers. Programming of these can be accomplished through a graphical interface called CodeBlocks or through a more traditional, text-based IDE. Microcontrollers can be interfaced with electronic circuitry so that code development can be done in parallel with hardware development. One example illustrating the close equivalence between the simulation and actual circuit is given in Figure 1, taken from the set of tutorials available from [7].

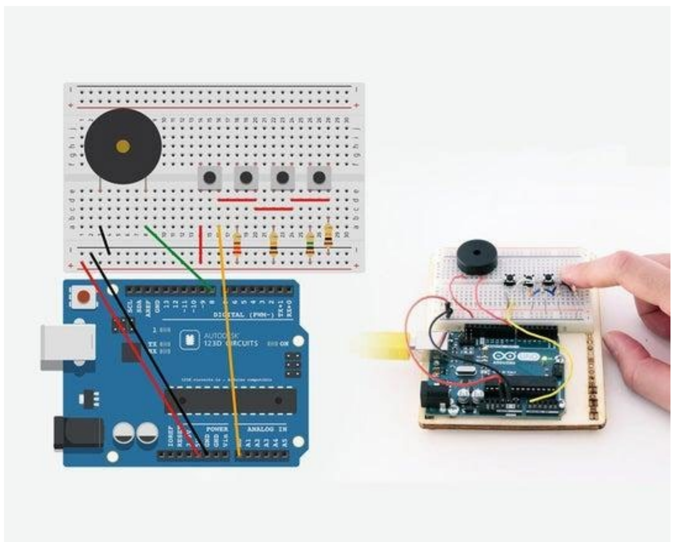

Figure 1. Illustration of simulation and implementation of a keyboard instrument [7]

There is already extensive literature on projects using Tinkercad, both in print as well as video tutorials. From our perspective, we want to explore using it to lower the barrier to engaging in authentic engineering projects. We hope this will happen through:

a) Easy access
b) Intuitive interface that closely mimics actual hardware
c) Early exposure to combined hardware and software projects
d) Combining the simulation with building projects in the lab and testing them in real life.

Next, we discuss how this is currently implemented.

3.1 Implementation: ECE 101 lab

A short class lecture introduces Tinkercad with a simple circuit example, then briefly introduces microcontrollers in general and the Arduino platform in particular. Students are provided with some simple references if they want to explore more on their own, e.g., [12], [13].

For the lab, students first simulate a simple circuit with familiar components - power supply, LED, resistor and multimeter - to gain experience with the program. One experiment they do here is to vary the value of the current-limiting resistor. They can see the LED brightness change, and the LED eventually fails if the resistor is too small.

In the main part of the lab, they connect the LED and resistor to an Arduino. They start with a built-in code to blink the LED. Even if they haven't studied C++ yet, they can understand each line of the code with a brief explanation. They then add a second LED and edit the code to make the LEDs blink both simultaneously and sequentially. Finally, they 'tinker' with their circuit, creating a light show of at least five LEDS which turn on and off in some colorful pattern.

3.2 Implementation: PIN 101: lab exercises, pre-lab prep

This course has no formal prerequisites and in order to get to more interesting design and project-based problems we have to first cover a few basics, such as:

1. Resistive circuits, including the parallel and series combinations
2. Ohm's law
3. Voltage and current dividers
4. Time-dependent effects such as R-C charging

The math and physics required for explaining these topics is relatively straightforward and can be built intuitively. This approach is reflected in the selection of lab topics which are then followed by applications, such as using a timer IC NE555 and linear voltage regulator LM317. While these applications may look intimidating at first glance, they require only a handful of components. The LM317 lab illustrates the application of voltage division and Kirchoff's laws, and results in a useful circuit. Similarly, the application of NE555 illustrates a practical design that utilizes R-C charging and voltage division to produce a capacitance meter or to control LED blinking.

After refining the labs during the first two years, in 2023 we decided to introduce a formal requirement for pre-lab exercises. This was driven by two realizations:

a) Many students were coming to labs without doing the required reading or watching videos
b) It is possible to almost exactly replicate the lab content on Tinkercad.

The former problem is not really surprising but any solution to it cannot be too burdensome for either students or instructors. Once we realized that point b) was possible it was fairly straightforward to design pre-lab assignments that required completion of Tinkercad simulations of some fraction of the actual lab assignment. For example, the second lab is about observing R-C charging. In the lab students build a simple circuit on the breadboard while the same circuit is simulated in Tinkercad, as shown in Figure 2. The simulation also enables asking “what if” questions that may be too time consuming in an actual lab setting. Similarly, reconnecting a DMM for voltage or current measurements is much less risky in simulation than in an actual lab setting. We believe that the fact that the simulation directly illustrates the actual experiment is very important because it reduces the cognitive load for novices. Figure 2 can be compared with Figure 3 that illustrates the corresponding schematic. The schematic would need to be explained in detail and many new concepts introduced such as ideal wires, connecting nodes, idealized sources, etc. Of course, these concepts will have to be explored further and solidified in follow-on courses.

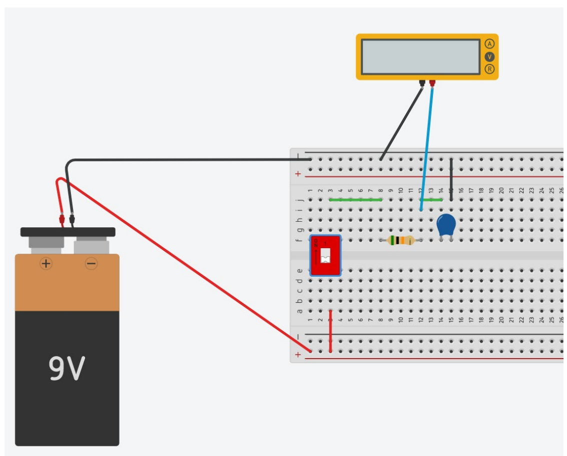

Figure 2. Example of Tinkercad setup for simulation of charging of an R-C circuit and measuring capacitor voltage.

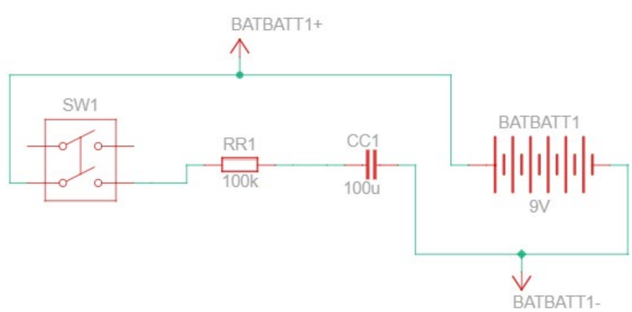

Figure 3. Circuit schematic of the R-C charging circuit from Figure 2.

Another lab experiment that is surprisingly easy to do in Tinkercad is simulation of the astable operation of an NE555 timer, which is shown in Fig. 4. Combined with a simple LED or buzzer as a load, it is easy to adjust components and directly observe the effects of various variables. Note that the NE555 operation is based on two concepts that were introduced in the two previous labs: voltage division which is used to set thresholds, and R-C charging which sets the frequency of pulses. Even though students come well prepared for the lab experiment this is still a challenging experiment mainly because it requires attention to details in wiring and debugging.

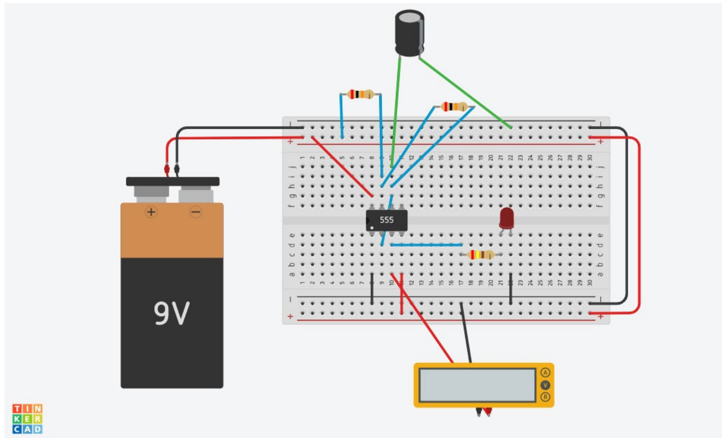

Figure 4. NE555 astable circuit simulation in Tinkercad.

In the future, we intend to build upon this set of circuit-based experiments to include analogous experiments using Arduino Uno. For example, it is straightforward to control LED blinking frequency with a few lines of code vs. using an NE555. Similarly, measuring a voltage divider on a breadboard easily leads to measuring thermistor resistance in a voltage divider and determining its temperature using Arduino. This way we hope to illustrate various approaches to solving similar problems. We have started this work, and we believe that Tinkercad will be even more useful in this setup due to its ability to combine Arduino code with external circuits.

4. Assessment

Given that this is a freshmen introduction to electrical and computer engineering class, we cannot expect much in terms of prior knowledge of circuits, hardware in general, or programming. Nonetheless, there is likely to be a wide variety of backgrounds of students at both institutions, especially at the home institution.

In Figures 5 and 6, we present the results of an initial survey that students fill at the start of the courses. In Figure 5, the question is:

> *How much experience do you have working with electrical circuits, either real hardware or in simulation?*

The scale is descriptive:

- None at all
- A little bit - I learned a bit about them but have forgotten most of it; I would be largely starting from zero.
- A moderate amount - enough to get me started without much trouble.
- Quite a bit - I am familiar with electrical circuits but not an expert.
- A great deal - expert level; I am very familiar with electrical circuits.

In Figure 6, the question is

How would you rank your programming skills (in any programming language)

And the scale is from Excellent to Terrible.

Three sets of survey results are presented:

1. From Fall 2020 for ECE 101 course at home institution
2. From Fall 2021 and 2022 for PIN 101 course in China

As expected, most students have very little background in either circuit design, hardware, or programming. However, there is a good number of students who are very confident in their abilities in these areas. Making this course interesting and valuable to all students is an ongoing challenge. One surprising result is that roughly ⅓ of students in ECE 101 deemed their programming skills as "Terrible", much more so than their Chinese counterparts. All this data verifies what we suspected - despite all the efforts of various middle and high schools to introduce topics relevant to engineering, we must assume that the majority of our students will have little or no experience in these areas. As will be shown next, we believe that Tinkercad can help us solve these problems.

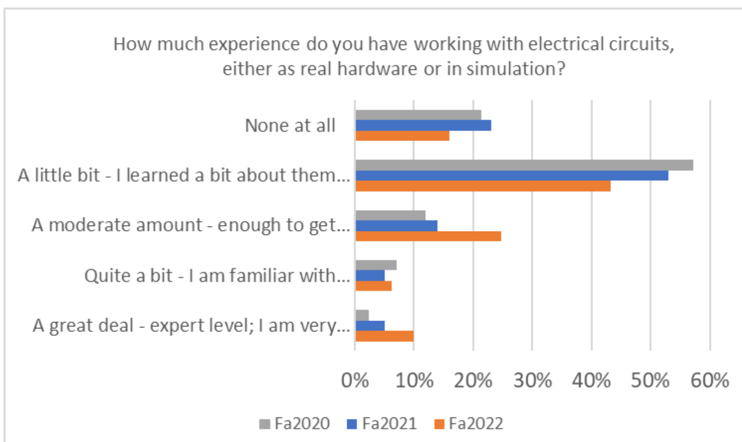

Figure 5. Student responses regarding their prior experience with electrical circuits.

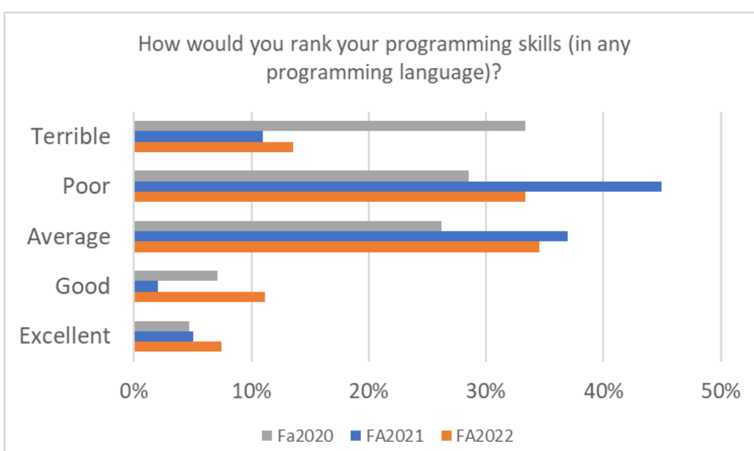

Figure 6. Student responses regarding their programming skills.

4.1 Self-assessment of effectiveness

Tinkercad was first introduced in our introductory classes in China followed by its introduction at PSU in the 2023-24 academic year. An obvious question is if students find Tinkercad useful and to this end we devised a simple end-of-the-term survey asking the following seven questions, starting with "Tinkercad is … ":

1. easy to learn
2. useful for improving my understanding of circuits
3. useful to prepare for labs (we didn't do this, but would it be useful to do a Tinkercad simulation before performing a lab?)
4. useful for introducing Arduinos
5. useful for understanding Arduino programming
6. I would like to continue using Tinkercad in future classes
7. Tinkercad would improve my motivation to study ECE

The results for ECE 101 in Fall 2023 term are presented in Figure 7. Class size was 51 and 23 students filled in the survey for a response rate of 45%. Overall, students liked Tinkercad's ease of use and its usefulness in learning different concepts. Somewhat surprisingly, almost 80% of students agreed strongly that it is useful in introducing Arduino (question 4) and for Arduino programming (question 5).

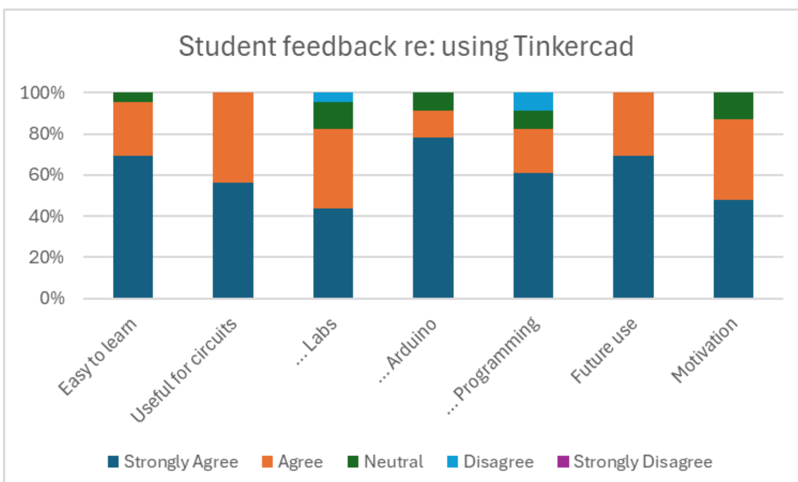

Figure 7. Results of student survey done in ECE 101 in the Fall 2023 quarter. There were 23 respondents out of 51 registered students.

Eighteen students also left comments at the end of the survey. The comments were uniformly positive, such as:

> *"Overall, I really appreciated Tinkercad! I think it would be huge mistake not to focus on it more."*
> *"Tinkercad and Arduino significantly improved my understanding of practical uses of circuits."*

"Tinkercad is really simple to use and the programming has two types block or text. For beginners I recommend the block as it would be easy to program."

Our original motivation for introducing Tinkercad was focused on circuits, but we now see that it could also be very useful in introducing programming concepts. Given the well-known problems with introducing programming concepts we should look for ways to make this introduction less intimidating. Some recent work indicates that the block-programming approach has some benefits over the more traditional text-based introduction to programming [8],[11]. In the future, we could leverage the fact that Tinkercad has both built in and compare both approaches. Finally, students did report improved motivation due to the use of Tinkercad, but they were slightly less enthusiastic about it, with 48% agreeing strongly, and 39% agreeing. Overall, these results give us confidence that introducing Tinkercad was a good idea and that we should work on expanding its use.

4.2 Transition between ECE 101 and 102

One of our motivations in introducing Tinkercad in ECE 101 at PSU was to provide a light introduction to microcontrollers and microcontroller programming in ECE 101, so that these were not entirely new concepts in ECE 102. We surveyed the students in ECE 102 in the Winter 2024 term, asking whether they found it useful to have been introduced to Arduino in ECE 101 using Tinkercad. The results of this survey question are shown in Figure 8 below. It can be seen that the majority of students agreed or strongly agreed that it was useful. While some were neutral, none disagreed. This survey does not provide us with a direct assessment of the effectiveness of this intervention, but it does provide justification for continuing with its development.

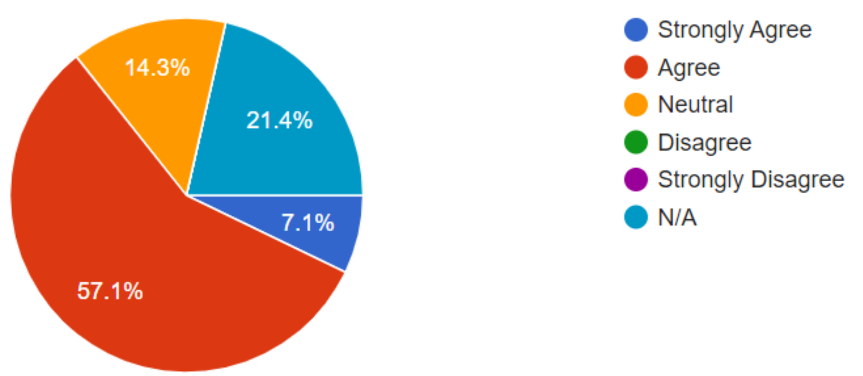

Figure 8. Student response to the question "Was it useful to have been previously introduced to Arduinos and Arduino programming in the Tinkercad lab in ECE 101?" There were 14 responses.

5. Conclusions and future plans

We have completed several rounds of introducing Tinkercad in introductory electrical engineering courses. This was done in two very different curricula and institutional contexts: one in the US and another in China. It was first introduced as a complementary activity, first building circuits in simulation followed by building them in the lab. Secondly, we introduced it as a supplement to learning about programming Arduino microcontrollers. Both of these were successfully implemented. Results of the most recent survey indicate that students appreciate its ease of use, intuitive interface, and opportunities to experiment on their own time and outside of the lab. We believe that this will also result in increased student engagement and motivation to pursue their studies. Students in the follow-on course found it a useful way to introduce microcontrollers and microcontroller programming. We plan to expand the programming component further and to explore ways to integrate it more with the follow-on courses.

References

[1] J. O. Campbell, J. R. Bourne, P. J. Mosterman, and A. J. Brodersen, "The Effectiveness of Learning Simulations for Electronic Laboratories," Journal of Engineering Education, vol. 91, no. 1, pp. 81–87, 2002, doi: 10.1002/j.2168-9830.2002.tb00675.x.

[2] M. D. Koretsky, D. Amatore, C. Barnes, and S. Kimura, "Enhancement of Student Learning in Experimental Design Using a Virtual Laboratory," IEEE Transactions on Education, vol. 51, no. 1, pp. 76–85, Feb. 2008, doi: 10.1109/TE.2007.906894.

[3] G. Olympiou and Z. C. Zacharia, "Blending physical and virtual manipulatives: An effort to improve students' conceptual understanding through science laboratory experimentation," Science Education, vol. 96, no. 1, pp. 21–47, 2012, doi: 10.1002/sce.20463.

[4] Tinkercad, tinkercad.com (accessed on 2/4/2024)

[5] Tinkercad blog, https://www.tinkercad.com/blog (accessed on 2/4/2024)

[6] Tinkercad, https://en.wikipedia.org/wiki/Tinkercad (accessed on 2/4/2024)

[7] Tinkercad tutorials, https://www.tinkercad.com/learn/circuits (accessed on 2/4/2024)

[8] P. L. Dickrell and L. Virguez, "Combining a Virtual Tool and Physical Kit for Teaching Sensors and Actuators to First-year Multidisciplinary Engineering Students," presented at the 2021 ASEE Virtual Annual Conference, Jul. 2021.

[9] J. E. Lewis, N. Hawkins, and B. S. Robinson, "Work in Progress: Remote Instruction of Circuitry in a Multidisciplinary Introduction to Engineering First-year Course," presented at the 2021 ASEE Virtual Annual Conference, Jul. 2021. https://peer.asee.org/38193

[10] FTC news about kidSAFE program - https://www.ftc.gov/system/files/attachments/press-releases/ftc-approves-kidsafe-safe-harbor-

program/kidsafe_seal_program_certification_rules_ftc-approved_kidsafe_coppa_guidelines_feb_2014.pdf

[11] D. Sun, F. Ouyang, Y. Li, C. Zhu, and Y. Zhou, “Using multimodal learning analytics to understand effects of block-based and text-based modalities on computer programming,” Journal of Computer Assisted Learning, vol. n/a, no. n/a, doi: 10.1111/jcal.12939.

[12] H. Mafukidze, “Introduction to the Arduino”, www.circuitbasics.com/ https://www.circuitbasics.com/introduction-to-arduino/ (accessed on 2/4/2024)

[13] Arduino tutorial, https://www.tutorialspoint.com, https://www.tutorialspoint.com/arduino/index.htm (accessed on 2/4/2024)