A Hands-On Online Summer Arduino Workshop for Middle School Students

Dr. Shari Klotzkin, Binghamton University

Shari Klotzkin has a PhD in Aerospace Engineering from University of Southern California and has taught undergraduate mechanical engineering classes at Penn State, Temple University, and Binghamton University. Currently she is working on a project at AMAG Consulting.

Dr. Howard S Kimmel, New Jersey Institute of Technology

HOWARD KIMMEL is Professor-Emeritus of Chemical Engineering and Retired Executive Director of the Center for Pre-College Programs at New Jersey Institute of Technology. In 2019 Dr. Kimmel was a recipient of the Presidential Award for Excellence in Science, Mathematics and Engineering Mentoring, one of 15 awardees nationwide. In addition, Dr. Kimmel has received numerous awards in recognition of his service, including: ASEE 1985 Vincent Bendix Minorities in Engineering Award, and ASEE CENTENNIAL MEDALION for "Significant Lasting Impact on Engineering Education," 1993. The NJIT Foundation Overseers Public and Institute Service Award, 1981 (First Recipient) and in 2005; and the Allan R. Cullimore Distinguished Service Award (NJIT) for 1991.

Dr. David Klotzkin, State University of New York at Binghamton

David Klotzkin received the B.S. degree in electrical engineering from Rennselaer Polytechnic Institute, Troy, NY, in 1988, the M.S. degree in materials science from Cornell University, Ithaca, NY, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, in 1997 and 1998, respectively. His research interests are in optoelectronics devices, including semiconductor lasers, waveguide amplifiers, organic light emitters, and photonic-crystal based planar lightwave circuits. His industrial experience includes three years of graphics hardware design at IBM corporation from 1988-1991, and several years of semiconductor laser design for telecommunications applications at various companies, including Lucent Technologies and Agere Systems. In 2002, he joined the Electrical, Computer Engineering and Computer Science Department at the University of Cincinnati. In 2008, he moved to the Electrical and Computer Engineering Department of Binghamton University.

A Hands-On Online Summer Arduino Workshop
for Middle School Students

The pandemic has had an enormous impact on daily life, from long-distance travel to access to restaurants, and one significant but less-noted area is in the access to pre-college summer activities. Educational camps and programs are a big part of many children's summer schedules, and these were generally not available in 2020.

To address this gap, a week-long Arduino workshop was developed to give middle school students an opportunity to actively engage in fun and educational STEM activities. The challenge was to see if an engaging STEM program on electronic hardware could be delivered remotely [1]. With the aid of a very capable electronic simulator program, the content could be effectively delivered and even implemented on real hardware.

The class met twice daily for a week and culminated with each student presenting their individual project on the final day. The lessons were structured as a brief demo by the instructors, followed by a mini-project, such as interfacing with an LCD display. While working on the activities, students were assigned to one of three breakout rooms, each one with an instructor.

The key software was the Tinkercad Circuits simulator, which could realistically simulate an Arduino microprocessor, including the programming, and associated electronic components. Students created Arduino programs using either a beginner-friendly graphical interface or the text editor. The graphical interface is based on the block coding of the Scratch programming language and is a proven successful introduction to programming for kids as young as elementary school age [2].

The projects (programs and electronic components) were simulated in Tinkercad and then implemented on real Arduino. The challenge then (to the students, and to the instructors helping them remotely) was being able to exactly replicate the connections and wiring on the breadboard in real life. Most students did this successfully and some realized very intricate projects.

Overall, these students spent a productive week during a difficult year and emerged with a working knowledge of electronic hardware and Arduino programming.

1. Introduction

The pandemic has had an enormous impact on daily life, from long-distance travel to access to restaurants, and one significant but less-noted area is in the access to pre-college summer activities. Educational camps and programs are a big part of many children's summer schedules, and these were generally not available in 2020.

To address this gap, a week-long Arduino workshop was developed to give middle school students an opportunity to actively engage in fun and educational STEM activities. The challenge was to see if an engaging STEM program on electronic hardware could be delivered remotely [1]. Most of the middle school students had little prior experience in both programming and electronic components. The use of the Tinkercad Circuits simulator enabled the instructors to introduce both aspects online to ease the transition to real hardware. The Tinkercad software includes a graphical programming interface to make developing and debugging Arduino codes more accessible to students. However, the simulator is powerful enough to allow participants to experiment with, create and debug worthwhile and innovative projects virtually.

2. Methodology

The goal of the project was to provide a fun experience for the students while learning about engineering. The program was implemented as follows: It was advertised in the local middle schools and on the Binghamton University web page and reached its cap of thirty registrants almost immediately. An Arduino kit (Elegoo Uno Super Starter Kit) was recommended but not required, though almost all (28/32) of the students purchased the kit.

The class met twice daily for a week and culminated with each student presenting an individual or group project on the final day. The lessons were structured as a brief demo by the instructors followed by a mini-project. For example, one project involved interfacing with an LCD display. To assist the students with the mini-projects, they were each assigned to one of three breakout rooms, each one with an instructor. The program website [2] gives the complete schedule, list of activities and the final projects completed (with names elided).

The key software was the Tinkercad Circuits simulator [3], which could realistically simulate an Arduino microprocessor, including the programming and associated electronic components. Students created Arduino programs using either a beginner-friendly graphical interface or the text editor. The graphical interface is based on the block coding of the Scratch programming language and is a proven successful introduction to programming for kids as young as elementary school age [4].

In general, the projects (programs and electronic components) were first simulated and debugged in Tinkercad. The computer program was then uploaded to an actual Arduino. The challenge then (to the students, and to the instructors helping them remotely) was being able to exactly replicate the connections and wiring on the breadboard in real life. Most students did this successfully and some realized very intricate projects.

2.1 Participants

The demographics from the initial registration information are shown in the following table.

Table 1 Participant demographics.

a.

| Grades:	4th -5th	6th	7th	8th	9th
# Participants:	2	1	12	8	9

b.

Male	Female
21	11

2.2 Software Environment

Tinkercad is well known for its 3D modeling package, but the circuits simulation software is a more recent addition and is a useful educational tool. It runs in a web browser, so it is accessible from almost any computer or Chromebook. Though participants can sign up for their own Tinkercad accounts, an account is not required to belong to a Tinkercad class.

Tinkercad Circuits allows users to put together complex circuits using electronic components from a parts library (Figure 1). The parts can be dragged from the library to a workspace where they are connected and simulated. If the user adds a programmable component, such as an Arduino microprocessor, a coding panel can be pulled out over the parts library. The user has a choice between seeing text, code blocks or both. However, to edit the text code, the user must switch to text only and cannot go back to blocks.

To use a real Arduino, the Arduino Create software is needed and can either be installed on a computer or run in a web browser. Since it interacts with real hardware, it requires a USB connection and installation of a plug in [5].

During the workshop, the instructors met with participants over Zoom. The exercises that participants did during the zoom meeting were presented on a webpage created using Blogspot, which allows for easy presentation and updates of the online material. Multiple pages can easily be created, organized and hidden or displayed as needed. Within minutes, clarifications can be added to the webpage and made available to participants.
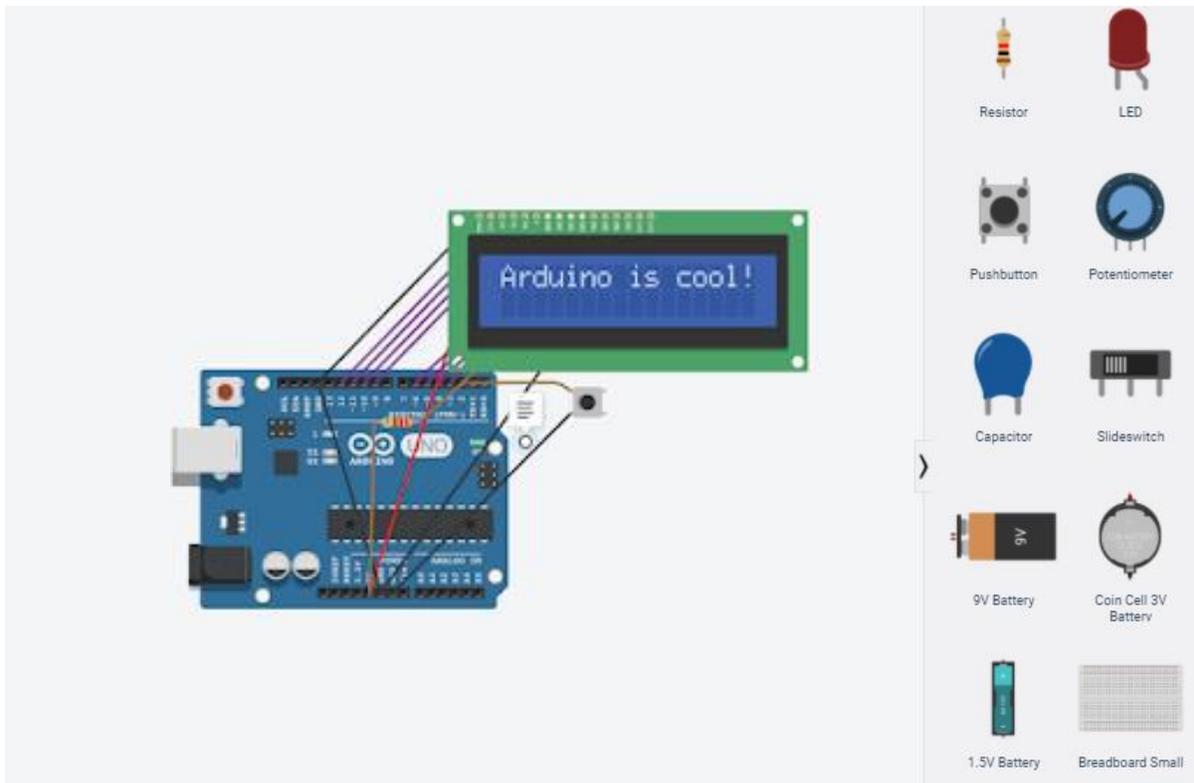
Figure 1: This view of the Tinkercad window shows a part of the library of components on the right side. The larger window on the left is the workspace.

2.3 Hardware

Parents were encouraged, though not required, to purchase an Arduino kit. We recommended the Elegoo Super Starter kit available on Amazon for around $37 because it has a good selection of accessories for beginners, as well as excellent documentation. However, the activities use generally available parts that should be included in most beginner Arduino kits. Most of the workshop activities could be accomplished with either the Tinkercad simulation or the Arduino kits, and most participants did both.

Parents were encouraged to test the Arduino Create software [5] in advance of the workshop start. This allowed some initial troubleshooting and saved time and effort later.

2.4 Workshop Schedule

The general workshop schedule was as follows:

Day 1: Introduction to Tinkercad, parts of a circuit, connecting the Arduino microprocessor to the computer and uploading a code to light up an LED.

Days 2-3: Students created circuits using components, such as the potentiometer, piezo electric buzzer, motor, LCD screen, and a variety of sensors. Most of the components used were available in both the Arduino kit and in Tinkercad.
Day 4: Students worked on a project of their own choosing and created slides for a presentation.
Day 5: Some last-minute work finishing up projects and final presentations.

The first four days included both a morning and afternoon session. Each session began with a short introduction to scientific or engineering principles, a demonstration generally in Tinkercad, and then activities for the participants to work through on their own. Detailed directions for these activities are available on program website.

While doing the hand-on activities both in Tinkercad and on a real Arduino, participants remained on Zoom. They were split into breakout rooms (each with about ten students and an instructor) to get individualized attention.

Each hands-on session, explained on the website, included a Main Activity in Tinkercad to illustrate the major points of the lesson, a Try-it Live activity which includes pictures of the real components, and any other information needed to successfully put together a circuit. Finally, a Challenge Activity was usually available, which had additional activities that could be done in either Tinkercad or a real Arduino. Though participants without the kit could not do the Try It Live activities, the Challenge Activities at the end of the lesson could be done either partly or completely in Tinkercad. In reality all the participants completed the Main Activities successfully, but some decided to leave the program without doing the challenge activities, which is easy to do in a virtual workshop. This benefitted the program in two ways: 1. Those with a shorter attention span could easily leave to do something else, and 2. The instructors could provide more individualized attention and encouragement to those that had more difficulty getting the circuits to work.

The Challenge Activity often involved adding another component to a circuit put together earlier in the lesson and could be done in either in Tinkercad or with real components. For instance, in the lesson on connecting sensors, including the temperature sensor, to the Arduino, the challenge activity involved adding an LED light that would turn on when the temperature gets too hot. While the online Tinkercad tutorials were not specifically included in this workshop, several basic activities were similar and one of the Learn Arduino projects was adapted (the Crystal Ball project).

Figure 2: Use of both Tinkercad and a real Arduino for an ultrasonic sensor project.

The morning session started at 9:30AM and the afternoon at 1:30PM. Each session lasted around 2 hours. By the end of each session, the instructors were able to answer questions individually so that all participants were able to finish the Main Activity and Try It Live activity. Many students had the time and were motivated to try the Challenge activity.

The final projects and presentations were successful, and some were fairly involved, considering limited time and resources. One of the completed projects is shown in Figure 2.

2.5 Coding

 Arduino projects require both hardware and coding, both of which are simulated in Tinkercad. Tinkercad offers both coding using Scratch-like blocks, called Codeblocks and standard text coding. With either blocks or text coding, the Arduino program can be saved on the disk in standard text format and uploaded to a real Arduino using Arduino Create.

Figure 3 shows an example of the block coding on the left and the text coding on the right. Notice that the steps to initialize the variables and the Arduino pin at the beginning of the text program are written automatically by Tinkercad. By using the block coding option, participants can successfully write a working Arduino code without having to get bogged down in syntax and to memorize key words and function names. For instance, the Tinkercad temperature sensor code block even computes the temperature from the sensor reading, so users do not need to know the voltage-number-to-temperature conversion. In addition, the text code shows the conversion, which can be seen when the blocks are converted to text. This way, users can learn about the need for the conversion without having to spend time looking up a tedious equation.

During the workshop, both block and text codes were included in the activity directions. For the first few Arduino projects, the participants were given the program, but they had to put it into Tinkercad themselves. After that, they were encouraged to construct at least part of the Arduino code themselves with some basic information about how to get started.

As students become more comfortable with coding, they can start off their program using block coding and then switch to text to further modify and test the code before uploading it to Arduino. Most of the activities during the workshop could be accomplished with block coding. Both the LCD screen, which introduced the idea of loading an external library and so did not have a code block, and input from the serial monitor, required text coding.

Though the workshop did not specifically use the ultrasonic sensor, because of the ease of the block coding in Tinkercad, several students were able to incorporate it into their final project. Tinkercad's code block for the ultrasonic sensor automatically writes a function in the text coding to read from the ultrasonic sensor and any required set up code (Figure 4). Whether Codeblocks or text were used for the Arduino code in Tinkercad, the student can then save it and write it to a real Arduino.

Almost all Arduino codes that work in Tinkercad also work on the real Arduino. Since the instructors had direct access to the students' work in the Tinkercad simulation, problems could be addressed and debugged even working remotely.

Figure 3: Tinkercad has both text coding (right) and code blocks (left). When using code blocks, Tinkercad also creates the text verson.

```
1  /*
2     Code to use PWM to fade an LE
3  */
4
5  int brightness = 0;
6
7  int direction = 0;
8
9  void setup()
10 {
11    pinMode(10, OUTPUT);
12 }
13
14 void loop()
15 {
16    if (brightness <= 5) {
17       // switch to increasing bei
18       direction = 1;
19    }
20    if (brightness >= 255) {
21       // switch to decreasing bri
22       direction = -1;
23    }
24    brightness += (direction * 5)
25    // LED should be connected to
26    analogWrite(10, brightness);
27    delay(30); // Wait for 30 mil
28 }
```



```
1  long readUltrasonicDistance(int triggerPin, i
2  {
3    pinMode(triggerPin, OUTPUT);   // Clear the t
4    digitalWrite(triggerPin, LOW);
5    delayMicroseconds(2);
6    // Sets the trigger pin to HIGH state for 1(
7    digitalWrite(triggerPin, HIGH);
8    delayMicroseconds(10);
9    digitalWrite(triggerPin, LOW);
10   pinMode(echoPin, INPUT);
11   // Reads the echo pin, and returns the sound
12   return pulseIn(echoPin, HIGH);
13 }
14
15 void setup()
16 {
17   Serial.begin(9600);
18
19 }
```
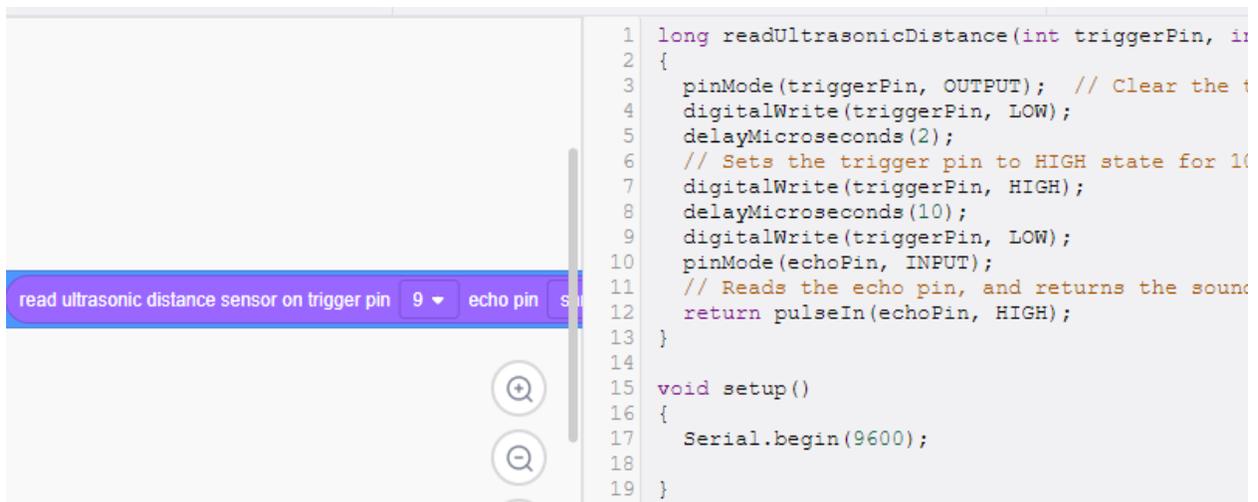
Figure 4: The ultrasonic sensor code block is an example of how all the text coding required by the Arduino is automatically created by Tinkercad Codeblocks.

3. Results

3.1 Final Projects

By the end of the workshop, students had learned some basics of constructing circuits and using the Arduino microprocessor and were able to create projects mostly independently. Since the assignment was very open-ended, the projects ranged in complexity from simple blinking lights and songs using the piezoelectric buzzer to backup warning cameras (Figure 5) and weather stations (Figure 6). Some participants were adventurous enough to experiment with components that were not specifically covered during the week, such as the proximity sensor and remote-control input.

For the project, they were encouraged to first debug it in Tinkercad before putting it together on the real Arduino. Those without a strong programming background were able to use block coding in Tinkercad to create the code for the Arduino.
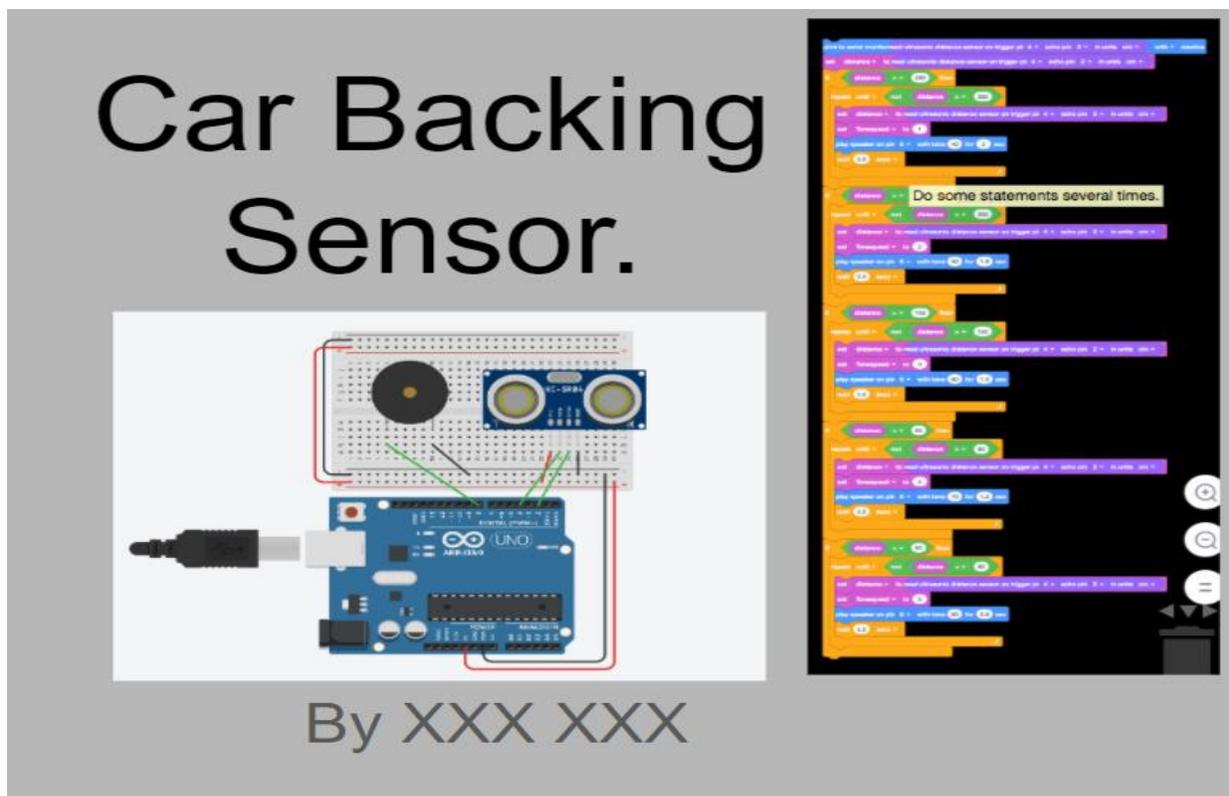

Figure 5: This project example shows the block coding that was used. Many projects included both a Tinkercad simulation and a real Arduino circuit.

Weather Station
By: XXX XXX
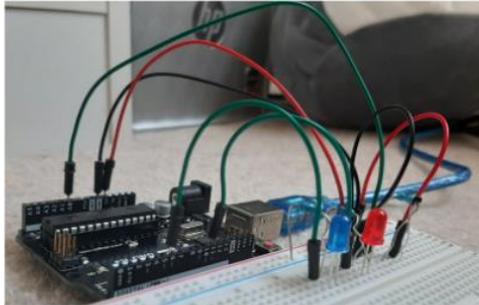
For this project, I used a thermistor, 2 LEDs, 3 resistors, 7 wires, and of course a breadboard as well as the Arduino UNO. My goal was to make the blue LED light turn on when the temperature was low and make the red LED turn on when the temperature was high.
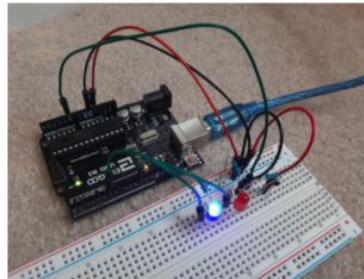
This circuit was near a vent in my room that was blowing cold air, so when I first uploaded the code, the blue LED light turned on while the red one stayed off.
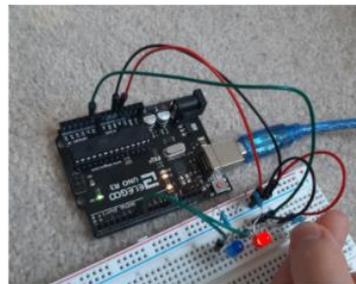
I put my fingers around the thermistor to increase the temperature, which caused the red LED to turn on and the blue one to turn off.

Figure 6: This project did not include a Tinkercad simulation because the thermistor in the kit is different from the temperature sensor in Tinkercad.

3.2 Challenges

The workshop made use of the classroom feature of Tinkercad which allows the instructors to examine students' designs and help students fix problems with their circuits. To examine a student's circuits and Arduino code, Tinkercad gives the instructor a copy of the project. Then the instructor can figure out a solution without changing the student's work and then suggest ways for the student to fix it. Most issues with setting up Arduino Create were successfully addressed before the workshop with the parents.

The real challenges arouse with participants trying to replicate the Tinkercad circuit in real life. A working circuit in Tinkercad reproduced exactly in real life would work as expected with very few exceptions. However, real life circuits are much messier (see Figure 6) than a Tinkercad circuit (Figure 7), more difficult to connect wires properly and easier to break.

Also, sometimes the participants needed some help properly matching the correct component in the Arduino kit with the Tinkercad part (for example, they confused the three pin and four pin distance sensors).
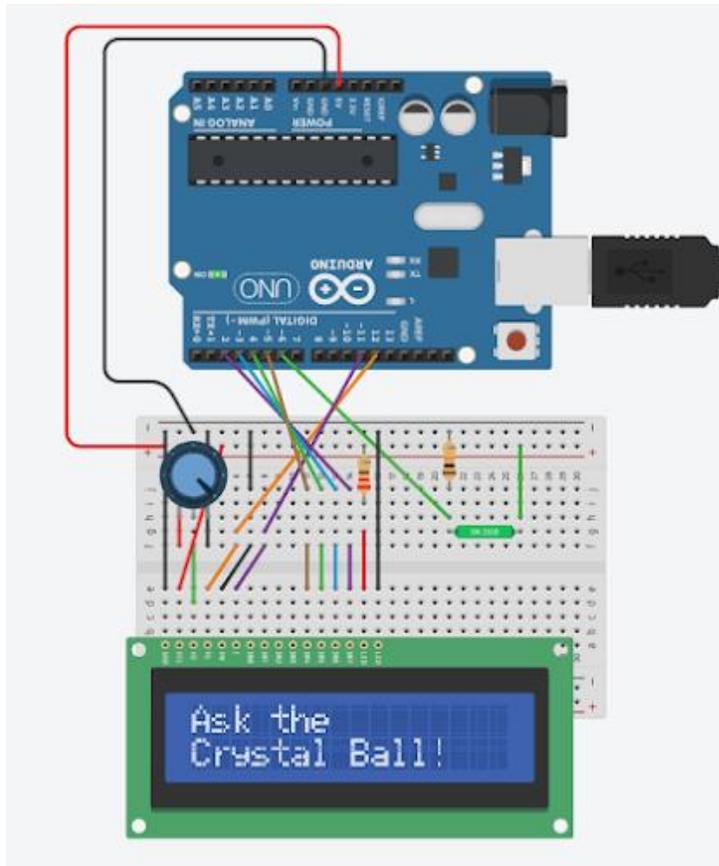
Figure 7: In Tinkercad Circuits the wire connections are relatively easy to follow.

Had the workshop been in person, the instructors would have been able to easily identify problems with the circuit. Since the instructors and participants interacted online only, problems were identified and solved by having the participant show the circuit over Zoom. Though these issues were eventually solved, it was more time consuming, and the instructors were dependent on the participant holding the circuit up to the camera at exactly the correct angle to examine it. It was a learning curve for the participants to figure out how to connect wires on a breadboard. If this workshop were to run again, perhaps incorporating a few activities that illustrate how to connect wires in a breadboard would be helpful. Also, some of the real electronic parts are prone to breaking. LEDs are fragile, but the kits came with plenty of extra. The part that caused the most difficulty was the DC motor; the wires on many of the motors fell off. Fortunately, some of the parents were able to soldier the wires back on to the motor. If the workshop had been in person, soldiering wires back onto the motor would have been trivial. In addition, some extra circuits components that were not included in the kits could have been available for experimentation.

Though the Tinkercad software is very versatile, it does have some shortcomings. Some components, such as the temperature sensor, do not have an identical counterpart in the real
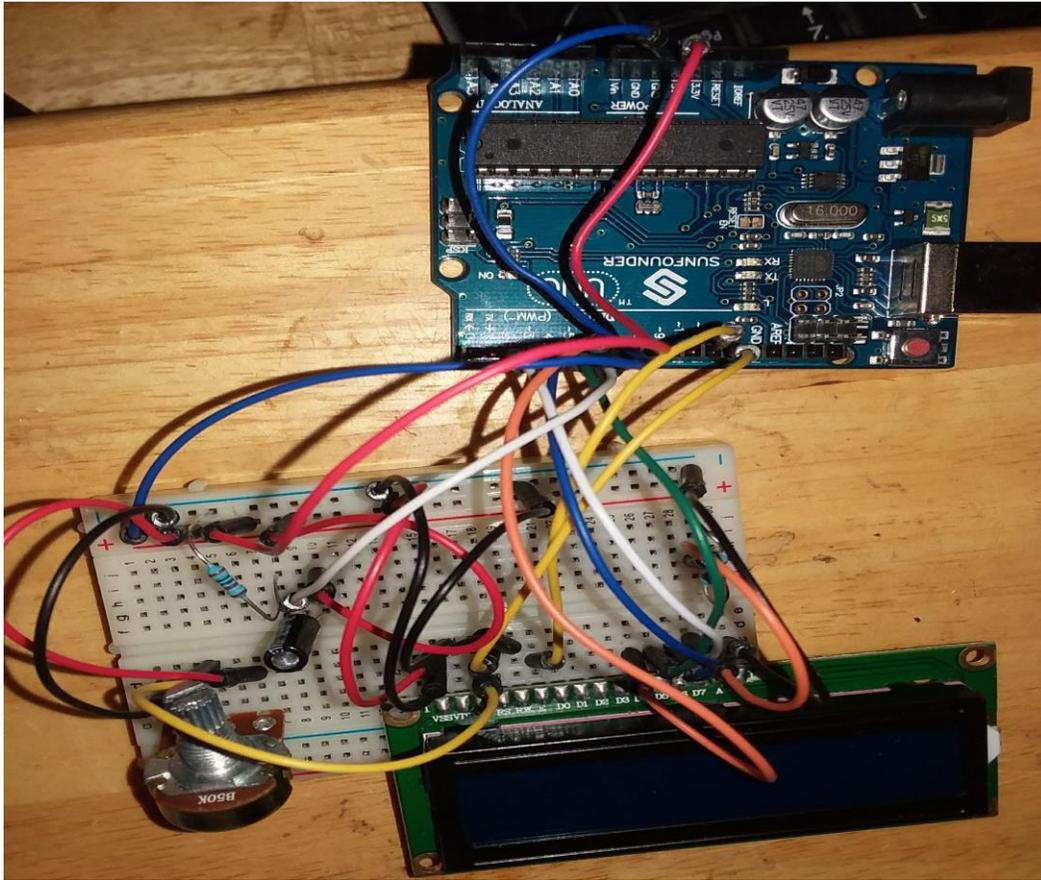
Figure 8: This is the same circuit as in Figure 7 using real components. Making sure the wires are connected correctly is more challenging.

Arduino kits, so the circuit and code required some modifications. Other components, such as the IR remote required some minor code modifications when transferred from Tinkercad to the real Arduino.

4. Discussion

The goal of this project was achieved. Rather than classroom-based lessons, this program was meant to be fun as well as a one-week learning experience for the students. Therefore, no pre-tests or post-tests were given or deemed necessary. Hence, only a qualitative study was done of the students' experiences in the workshop. This introduction to a STEM field that they may not have experienced previously would hopefully increase their interest in considering a STEM career in the future.[4]

The participants were surveyed before and after the workshop. Prior to this workshop, 75% of the students had little or no programming experience and only 4 students had experience with a text-based programming language. Over 80% of the students had little to no experience with

circuits. The post-workshop survey had 22 responses and slides for 24 final projects were created (one group project by two siblings).

4.1 Student Feedback

To help with future planning, a post-workshop survey included a few questions about the activities and about coding in particular.

Table 2 summarizes responses of the participants to the post-workshop survey. The table shows responses of whether they agree, disagree, or have no opinion to the following statements:
I had fun! (FUN)
I learned something! (LEAR)
The projects were too easy! (EASY)
The projects were too hard! (HARD)
I want to be an engineer! (ENGIN)

Table 2 Post-workshop survey responses (N.O. is 'No Opinion')

Grade	Response	FUN	LEAR	EASY	HARD	ENGIN
5th gr. 1 stud.	Agree	1	1	0	1	0
	Disagree	0	0	1	0	0
	N. O.	0	0	0	0	1
7th gr. 7 stud.	Agree	6	7	0	0	6
	Disagree	0	0	6	6	1
	N. O.	1	0	1	1	0
8th gr. 5 stud.	Agree	4	5	0	0	2
	Disagree	0	0	5	4	2
	N. O.	1	0	0	1	1
9th gr. 9 stud.	Agree	8	9	1	3	5
	Disagree	0	0	6	4	1
	N. O.	1	0	2	2	3

All the students agreed to the statement that "I learned something." Of the 22 students, 3 had no opinion to the statement "I had fun" while the other 19 students agreed with it. Thirteen of the students indicated that they "want to be an engineer" while 5 appear to be undecided and only 4 disagreed with the statement. Most of the students disagreed with the statements regarding whether the projects were either "too hard" or too easy" indicating that they were at the correct level for middle school students. While the 5th grader indicated that the projects were too hard, only 3 9th graders agreed with those statements.

A coding question was included on the post workshop survey to see if the availability of Codeblocks offered by Tinkercad had a significant effect on the students' comfort level with using Arduino since traditional Arduino requires text coding. 60% said that they were most comfortable with code blocks, and 45% said that they were equally comfortable with text and code blocks. Several students expressed an interest in spending more time on coding during the workshop.

The end of program survey also provided some insight into the learning acquired by the students. For example:

> "I thought this was an awesome learning experience, I had a lot of fun while being an engineer and creating projects. I hope to do his next year as well!!!!!"
> "Learning about different things"
> "Being creative in order to make our projects."
> "Being able to put together circuits and learn how to use different parts"
> "The best part was probably the projects themselves as they implemented so many elements and hardware, created by us, eventually forming a work of engineering skill."
> "Thank you for such a great experience! I definitely learned from this class and enjoyed it."

The survey included space for the participants' favorite part of the workshop. At least 5 of the participants mentioned that the highlight was creating their own project at the end. A few of the comments are shown in Table 3.

Table 3: Some comments from post-workshop survey.

"I liked the block coding and how easy it was"
"I liked the fact that we didn't have to solder anything and we got to make our own circuits at the end."
"I liked that the teachers stayed on zoom until they were sure that everyone had gotten the help that they needed. It was great to always have the chance to ask a question."
I liked that we got to do our own projects on our own time". (Certainly an advantage of a virtual workshop.)
˜This program was awesome I really liked it.

5. Conclusion

By the end of the workshop, the participants had learned some STEM skills that will encourage them to try more activities in the future, and maybe even consider a career in a technical field.

Despite the challenges in delivering this content remotely, almost all of the students picked up the flow and process of programming Arduinos and breadboarding. Overall, the participants (and the instructors) felt they had a productive and educational week.

References

(1)   C. Vidal-Silva, J. Serrano-Malebran and F. Pereira, "Scratch and Arduino for Effectively Developing Programming and Computing-Electronic Competences in Primary School Children," 2019 38th International Conference of the Chilean Computer Science Society (SCCC), Concepcion, Chile, 2019, pp. 1-7, doi: 10.1109/SCCC49216.2019.8966401.

(2)   S. Klotzkin (2020, July 3) *Arduino Summer Workshop*. Retrieved February 19, 2021, from https://klotzkinsummerworkshop.blogspot.com/

(3)   Autodesk (2021) *Learn How to Use Tinkercad.* Retrieved February 19, 2021, from http://www.tinkercad.com/learn/circuits

(4)   M. Armoni, O. Meerbaum--Salant, M. Ben-Ar, "From Scratch to "Real" Programming", ACM Transactions on Computing Education, vol. 14, no.4. article no.: 25, February 2015. [Online] https://doi.org/10.1145/2677087

(5)   Arduino (2016, November 21) *Getting Started with Arduino Web Editor on Various Platforms - Arduino Project Hub*. Retrieved February 19, 2021, from (https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-on-various-platforms-4b3e4a).

(6)   Shultz, Robert, Daniel Edward Ueda, Jessica S. Ward, and Adam K. Fontecchio. "A Hands-On, Arduino-Based Approach to Develop Student Engineering Skills and Introduce Cybersecurity Concepts to K-12 Students." In *2015 ASEE Annual Conference & Exposition*, pp. 26-54. 2015.