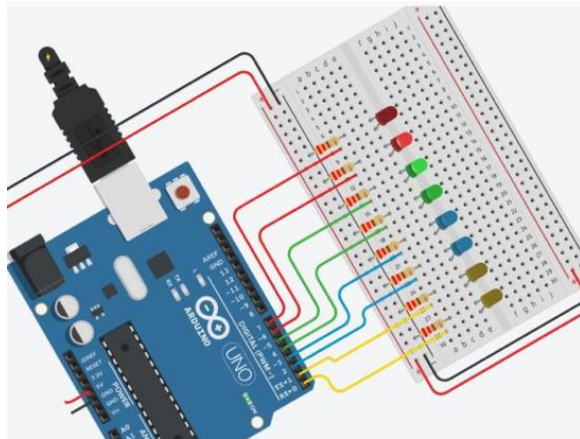




Lesson Plan: Program an LED Light Show



Grades: 6 – 12 (US)

Duration: 2 hours

Skills:

- Circuits
- Debugging
- Programming

Subjects:

- Electronics
- Computer Science
- Technology

Lesson Plan Overview

Students will learn the basics of using the Tinkercad Circuits interface to build a simple battery-powered LED circuit. Next, they will learn how to connect an Arduino to the circuit and program it to blink the LEDs. Finally, they will apply what they learned to make their own LED light show. Students will also learn some best practices for building circuits, and how to debug and troubleshoot their circuits and code. The estimated duration is provided for each part of lesson, giving you the flexibility to adjust as needed. This lesson requires no previous experience with circuits or programming.

Acknowledgment: The Active Learning Initiative at Cornell University

Learning Objectives

- Build a basic LED circuit.
- Connect and program an Arduino to control the circuit.
- Expand the circuit to add more LEDs.
- Build neat, organized circuits.
- Troubleshoot when things don't work as expected.



Standards

- **ISTE Standard 4 | Innovative Designer:** Students use a variety of technologies within a design process to identify and solve problems by creating new, useful, or imaginative solutions.
- **ISTE Standard 5 | Computational Thinker:** Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
- **NGSS MS-ETS1-2:** Evaluate competing design solutions using a systematic process to determine how well they meet criteria and constraints of the problem.

For more information about standards used in this lesson plan, visit these websites:

- ISTE: www.iste.org/standards/for-students
- Next Generation Science Standards: www.nextgenscience.org/search-standards

Materials

This is a list of materials each student will need to complete this lesson.

- Computer with access to the Internet and Tinkercad
- Slideshow software, like Google Slides, PowerPoint, or Keynote (optional)

You'll find links below where you can directly download and print files from the [Tinkercad Lesson Plan page](https://www.tinkercad.com/lessonplans/program-an-led-light-show) (<https://www.tinkercad.com/lessonplans/program-an-led-light-show>) that your students will need for this lesson:

- [Evaluation Rubric](#) [PDF with fillable text fields]
- [Example Tinkercad file](#) [link to Tinkercad]
- [Light Show Intro](#) [video]
- [Create a New Circuit](#) [video]
- [Create a New Circuit](#) [PPTX]
- [Build a Basic LED Circuit](#) [video]
- [Build a Basic LED Circuit](#) [PPTX]
- [Add an Arduino and Breadboard](#) [video]
- [Add an Arduino and Breadboard](#) [PPTX]
- [Blink an LED](#) [video]
- [Blink an LED](#) [PPTX]
- [Add a Second LED](#) [video]
- [Add a Second LED](#) [PPTX]
- [Light Show](#) [video]
- [Light Show](#) [PPTX]
- [Registers](#) [video]
- [Registers](#) [PPTX]

Setting Up Your Class

There are three options for each section of this lesson, allowing you to customize it to your classroom format:

- You can use a projector (or share your screen if students are learning remotely) to demonstrate and guide students through using Tinkercad Circuits yourself. This option works well if the entire class will be working together at the same pace, as you can pause periodically to make sure all students are caught up.
- You can project (or screen share) pre-recorded video instructions for each section, and then let students work independently. Students can also watch the videos at their own pace if they are working remotely, or if they are in a classroom and have their own headphones.
- Downloadable slides with complete instructions are available for each section. This option works well in a classroom environment where you want students to work at their own pace, but they cannot watch videos separately (e.g. they do not have headphones), or as a supplement to the videos if students are working from home at their own pace.

To introduce your students to the lesson, you can start with the introductory video in the next section. You can also guide them through or ask them to explore the tutorials available on the [Tinkercad Learn Arduino page](#).



Project Overview [1m]

1. Introduction Video [1m]

[Watch this brief video for an overview of the Program an LED Light Show project.](#) (Click the link to watch the video.)

An example Tinkercad Circuits file is available here as a teacher reference: <https://www.tinkercad.com/things/dcoobTV5YGM>. You can also show your students this simulation at the beginning of class, but we do not recommend sharing the link to the circuit so that students are encouraged to create their own.

Build and Program a Circuit [45m]

2. Create a New Circuit [10m]

Teacher Instructions	Student Instructions
<p>Show the Create a New Circuit video or the slideshow to your students (<i>please see links below</i>). Make sure they can find the Circuits section of Tinkercad, since it defaults to the 3D Designs section when you log in. Explain that it is a good habit to give the circuits names that will make sense in the future, instead of leaving the randomly-generated names. If you come back to your circuit a week or a month from now, will you be able to tell what it does based on the name?</p>	<p>Watch the video or slideshow (<i>please see links below</i>) to learn how to make a new circuit in Tinkercad. Then open a new window in your browser and log into Tinkercad. In the Tinkercad Circuits workspace, create a new circuit, and give it a name like "My First Circuit" Your teacher will either demonstrate how to do this, or ask you to follow along with the video.</p>

Materials:

- [Create a New Circuit](#) [video]
- [Create a New Circuit](#) [PPTX] (*Please click the link to download the slideshow.*)
- Computer with audio

Outcome:

- Create a new circuit in Tinkercad Circuits.

3. Build a Basic LED Circuit [10m]

Teacher Instructions

Show the [Build a Basic LED Circuit video](#) or the [slideshow](#) to your students (*please see links below*). Check to make sure students are comfortable with the basics of the Tinkercad Circuits editor at this point. They need to understand how to click and add parts from the components drawer on the right, move them around in the workplane, and connect them with wires. Students who struggle with using the interface will have difficulty with the rest of the lesson.

This is a good point to start to emphasize debugging. If a student's LED does not light up, help them ask questions to figure out why. For example, do they have an open circuit with one wire disconnected? Is the connection to the battery backwards?

Student Instructions

Watch the video or slideshow (*please see links below*) to learn how to build a basic LED circuit. In a separate window, build a series circuit in Tinkercad with a 9V battery, LED, and 1 kilo-ohm ($k\Omega$) resistor. Make sure the long leg of the LED is connected to the positive (red) terminal of the battery. When you click "Start Simulation," the LED should light up.

Materials:

- [Build a Basic LED Circuit](#) [video]
- [Build a Basic LED Circuit](#) [PPTX] (*Please click the link to download the slideshow.*)
- Computer with audio

Outcome:

- Build a battery-powered LED circuit.

4. Connect an Arduino and Breadboard [10m]

Teacher Instructions

Show the [Add an Arduino and Breadboard video](#) or the [slideshow](#) to your students (*please see links below*). It's important to emphasize *good wiring practice* here. Many students may connect wires in a straight line between two points. This makes the circuit messy and harder to debug as they add more wires.

Encourage them to carefully route the wires to keep things neat, as demonstrated in the video. Color-coding wires is also important. It is standard convention in electronics to use red for power/5V and black for ground/negative.

Student Instructions

Watch the video or slideshow (*please see links below*) to learn how to add an Arduino and breadboard. Then in a separate window, start a new circuit in Tinkercad. Add an Arduino and a breadboard to the circuit, rotate them both 90 degrees so they are upright, and put them next to each other. Connect the Arduino's 5V and GND pins to the breadboard's power (+) and ground (-) buses respectively. Use neat, color-coded wiring.

Materials:

- [Add an Arduino and Breadboard](#) [video]
- [Add an Arduino and Breadboard](#) [PPTX] (*Please click the link to download the slideshow.*)
- Computer with audio

Outcome:

- Set up an Arduino and breadboard so you are ready to start controlling LEDs with the Arduino.



5. Blink an LED [10m]

Teacher Instructions

Show the Blink an LED video or the slideshow to your students (please see links below). As the circuit becomes more complex, it becomes more important to emphasize good wiring practice and debugging. Some students might not get their LED to blink on the first try. Avoid fixing it for them – guide them through the troubleshooting process to make sure everything is connected properly on the breadboard. Since they are using built-in example code, there should not be any errors in the code at this point.

Student Instructions

Watch the video or slideshow (*please see links below*) to learn how to blink an LED. Then in a separate window, connect an external LED to Arduino pin 13 using the breadboard in Tinkercad. This time you will need to change the value of the resistor – set it to 220Ω. Click the Code button and use the Blocks dropdown menu to select "Text." When you click "Start Simulation," the LED should blink on and off.

Materials:

- [Blink an LED](#) [video]
- [Blink an LED](#) [PPTX] (*Please click the link to download the slideshow.*)
- Computer with audio

Outcome:

- Connect a single LED to the Arduino and make it blink using built-in example code.



6. Add a Second LED [10m]

Teacher Instructions

Encourage your students to try this before they review the slides or video for this section to find "the answer" (please see links below). You can also emphasize that there is not a single "right answer" to this part of the lesson. The example uses a second LED connected to Arduino pin 12, however, the students can choose a different pin if they want to.

Make sure students are comfortable with the following before you proceed:

- Wiring an LED to a new Arduino pin, including a resistor
- Using the `pinMode()` command to set the appropriate pin as an output
- Using the `digitalWrite()` command to turn the LED on and off

Student Instructions

In a separate window use what you have learned so far to add a second LED to your circuit in Tinkercad. You will need to choose an Arduino pin and connect it the same way you connected the first LED. Make it blink on and off at the same time as the first LED. Hint: you can do this by copying and pasting lines of code in your program; you do not need to write any new code from scratch. The only thing you need to change is the pin number. Try to do this before you look at the slides or video for this section (*please see links below*).

Materials:

- [Add a Second LED](#) [video]
- [Add a Second LED](#) [PPTX] (*Please click the link to download the slideshow.*)
- Computer with audio

Outcome:

- Connect a second LED to the Arduino, and modify the example code to make it blink.

Make Your Own Light Show [45m]

7. Design Your Light Show [15m]

Teacher Instructions

Show the [Light Show video](#) or the [slideshow](#) to your students (*please see links below*). Some students might get ambitious here – set reasonable expectations based on the time you have available. For example, you might say that everyone should start out with four LEDs and add more if they have time, instead of trying to jump right to 12 LEDs and then not having time to finish.

Note that programming a circuit with lots of LEDs can become cumbersome using the `digitalWrite()` command, since you need one line of code to turn each individual LED on or off. See the "Optional: Programming with Registers" section to learn how you can turn up to eight LEDs on or off with a single line of code.

Student Instructions

Watch the video or slideshow (please see links below) to learn how to design your light show. Then it's your time to shine! Add more LEDs and design your own LED light show in Tinkercad. You can make the LEDs blink on and off in different patterns, or even try to make simple animations like an "LED chaser."

Materials:

- [Light Show](#) [video]
- [Light Show](#) [PPTX] (*Please click the link to download the slideshow.*)
- Computer with audio

Outcome:

- Design your own LED light show.

8. Share Your Work [30m]

Teacher Instructions

For remote learning, you can have students share their screens one at a time, or put them in breakout rooms so they can share in smaller groups. For in-person learning, you can have all students leave their simulations running on their computers, and let them walk around the room to look at other students' projects.

Student Instructions

Once you have finished your light show, share it with your teacher and the other students in your class.

Materials:

- Computer with audio

Outcome:

- Show off your work to the rest of the class.

Optional: Programming With Registers [20m]

9. Learn to Use Registers [10m]

Teacher Instructions	Student Instructions
<p>This is an advanced, optional topic that you can use as an add-on to this lesson plan. Show the Registers video or the slideshow to your students (<i>please see links below</i>). The vast majority of Arduino projects use the built-in Arduino functions and libraries because they are easier for beginners to understand and use. Registers make the code harder to read, but sometimes they allow more advanced functionality or can make the code more efficient.</p> <p>In the case of this project, a single line of code using registers can replace multiple lines of code with the <code>pinMode()</code> or <code>digitalWrite()</code> commands. See this video for a more detailed explanation of why you would want to use registers, and this video series for a full set of tutorials on programming with registers in Tinkercad.</p>	<p>Watch the video or slideshow (<i>please see links below</i>) to learn how to use registers. Registers are specific locations in a microcontroller's memory. Each register is made up of individual bits that can be either 1 or 0. Many registers have special functions related to the microcontroller's hardware – for example, the Arduino's digital I/O pins. As explained in the following video and slideshow, there are registers you can use to replace the <code>pinMode()</code> and <code>digitalWrite()</code> commands in your Arduino code. This approach lets you control up to 8 LEDs with a single line of code.</p>

Materials:

- [Registers](#) [video]
- [Registers](#) [PPTX] (*Please click the link to download the slideshow.*)
- Computer with audio

Outcome:

- Use registers to blink a single LED.

10. Update Your Light Show [10m]

Teacher Instructions

Students with previous programming experience may be confused by registers. A register is not an array or a variable, even though it seems like one. Emphasize the idea that each bit (1 or 0) in a register corresponds to a physical pin on the Arduino, and each 1 or 0 has a meaning associated with it (output/input for the DDR register, and high/low for the PORT register).

Student Instructions

Program your light show using registers instead of the `pinMode()` and `digitalWrite()` commands. Does this make your code shorter and easier to read? Can you program new behavior for your LEDs that would have been difficult to program before?

Outcome:

- Program an LED light show using registers instead of Arduino commands.