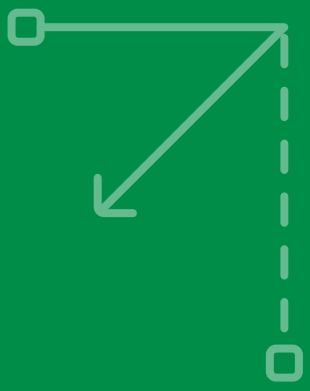
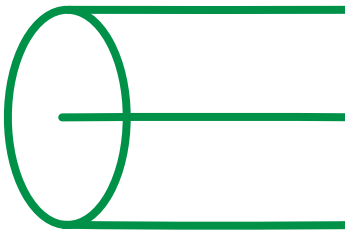
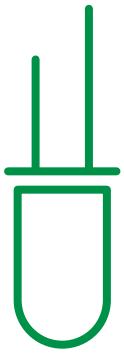
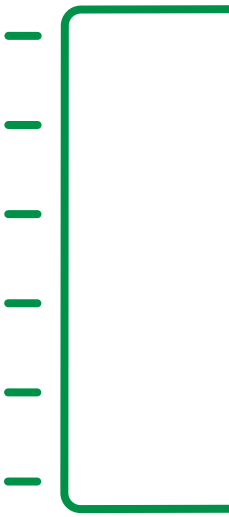
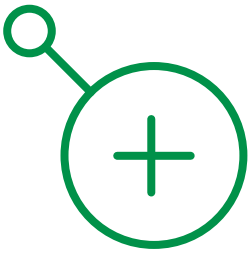
Official Guide to Tinkercad Circuits

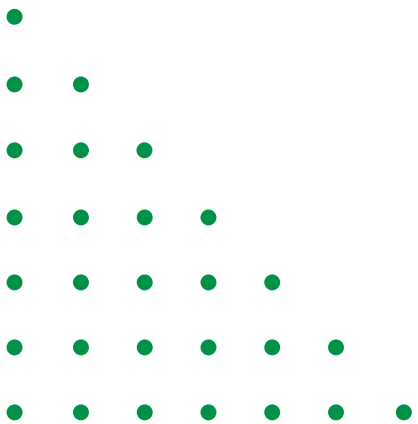




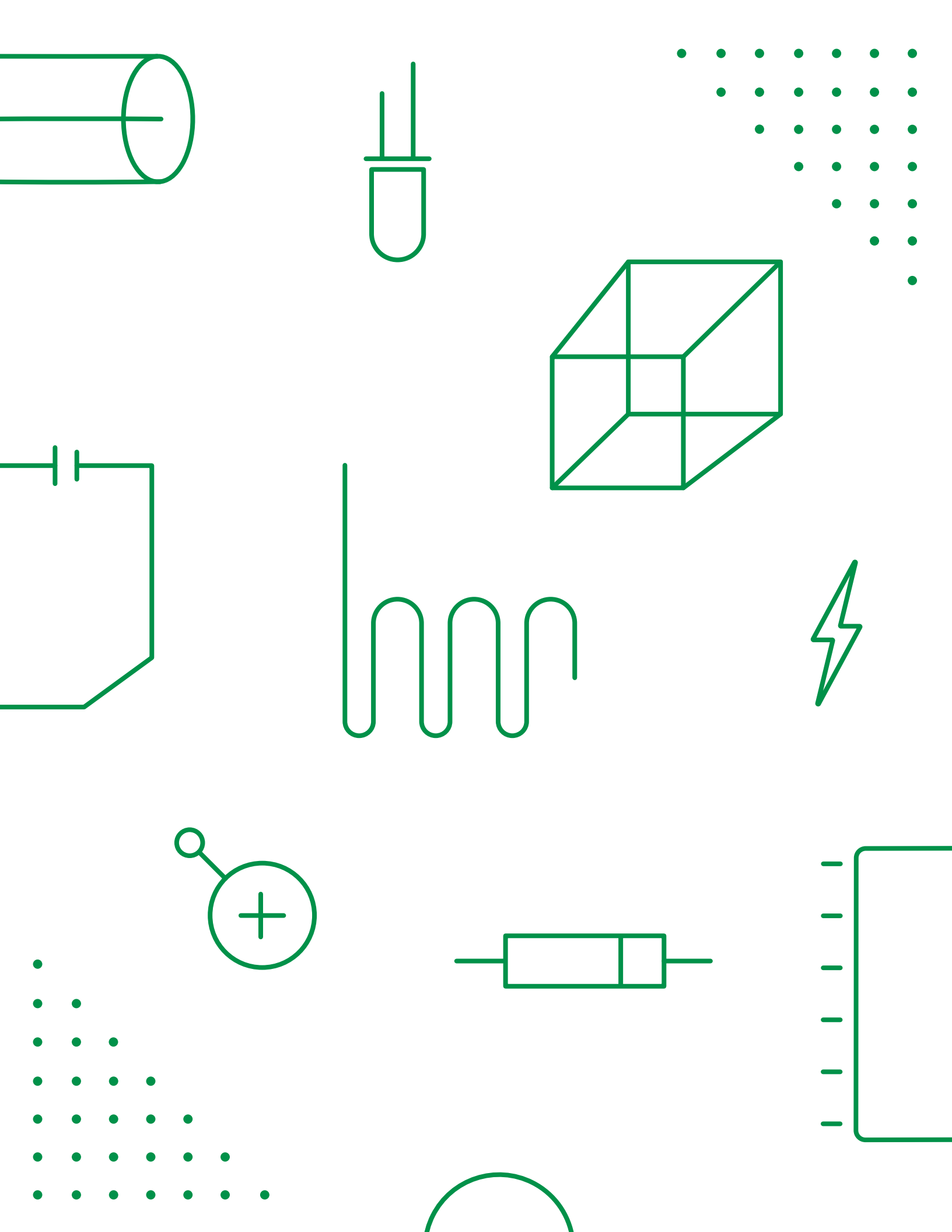








TINKERCAD AUTODESK Tinkercad

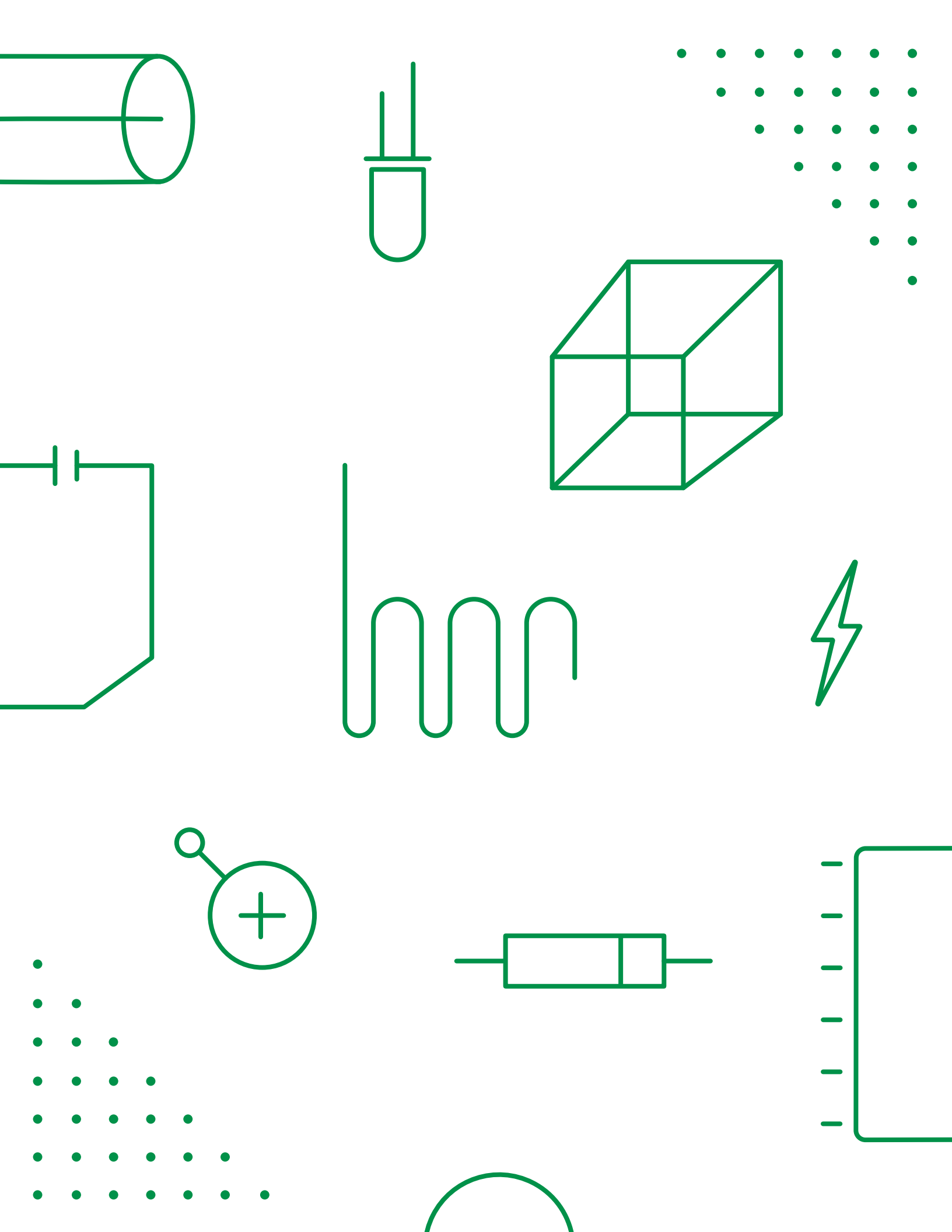


Official Guide to Tinkercad Circuits

Tinkercad Circuits is the easiest way to get your students started with learning electronics. Using our interactive circuit editor, students can explore, connect, and code virtual projects with a bottomless toolbox of simulated components.

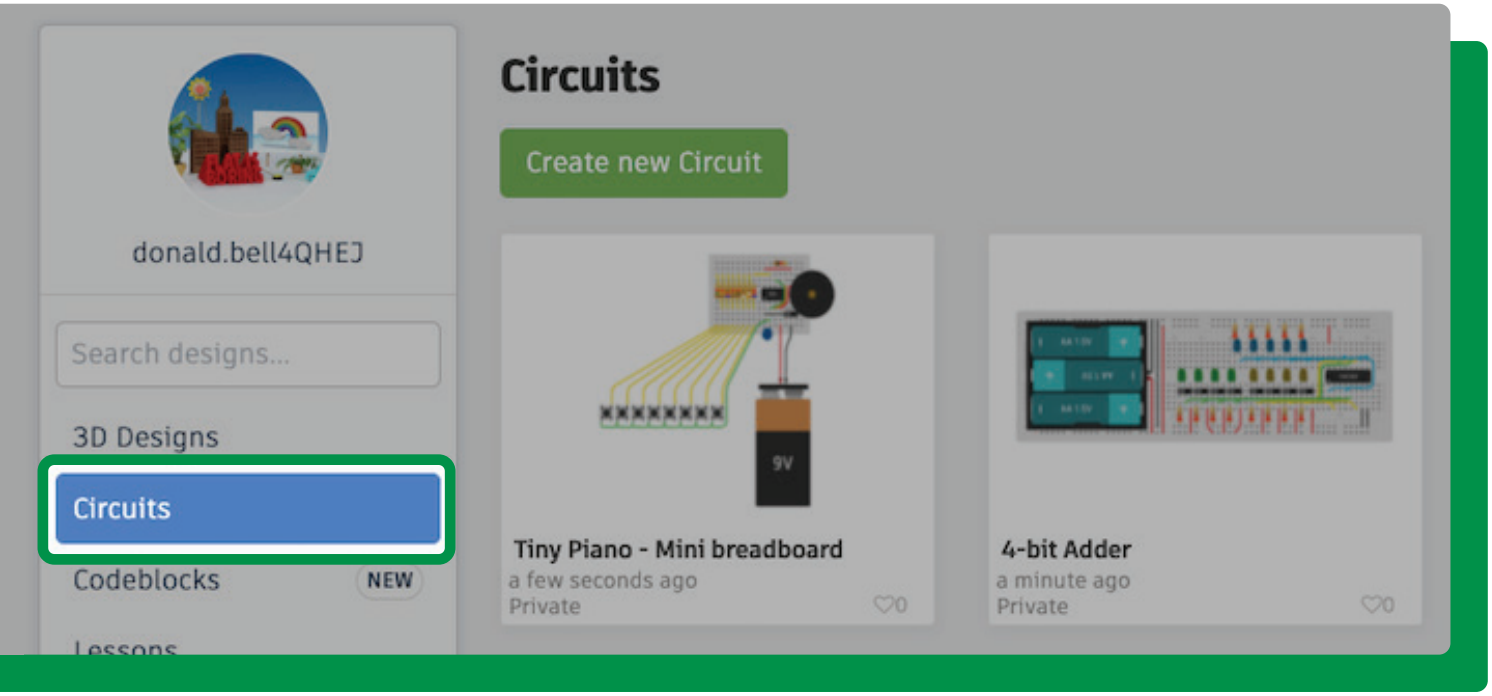
Available in 16 languages, on any computer with an internet connection, Tinkercad Circuits is an unmatched resource for electronics education.

Scan this QR code to access a digital version of this Offical Tinkercad Guide.
autode.sk/tinkercad-guide-circuits



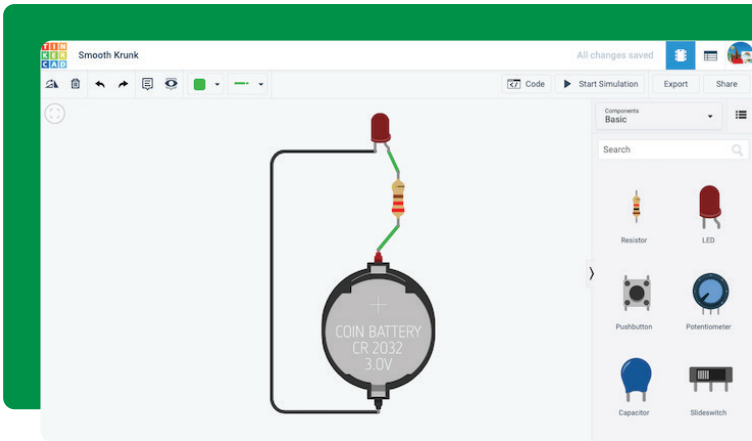
01 Exploring Circuits

After signing in to Tinkercad you'll find a dashboard of your recent designs. By default, this dashboard will showcase designs made with Tinkercad's 3D editor. To see a view of your Circuit designs, simply click the Circuits link in the left menu (highlighted below).



From this Circuits view you can scroll through any of your existing creations, or create something new by clicking the Create New Circuit button.

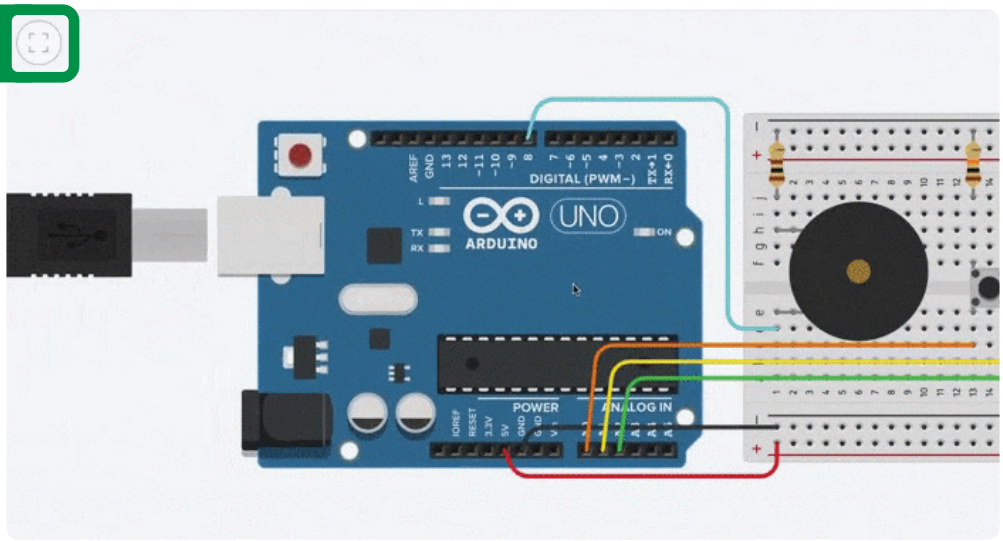
Tinkercad's Circuits editor has a similar layout to its 3D design editor. You'll find a large workspace on the left for creating your design. On the right side you'll see a panel filled with components you can drag and drop into the workspace to create your circuit.



Unlike Tinkercad's 3D design editor, the workspace in Circuits is two-dimensional. You can move your components around by selecting and dragging them, or pan the view around your design by clicking and dragging the empty space around it.

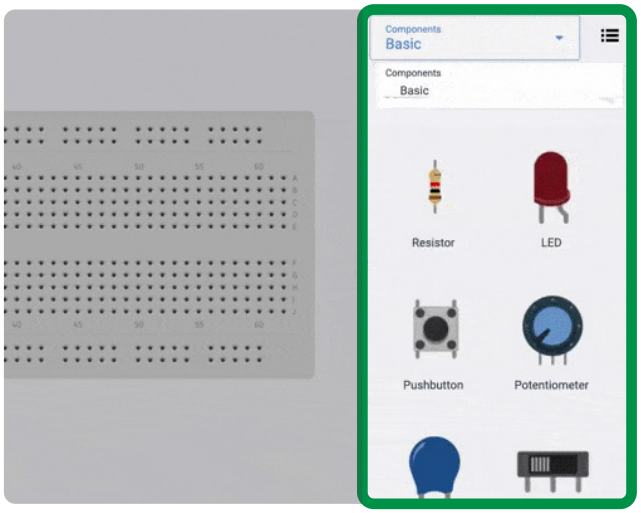
You can also zoom in and out of your design by using the scroll wheel on your mouse, a two-finger gesture on your trackpad, or a key combination of Command + and Command -.

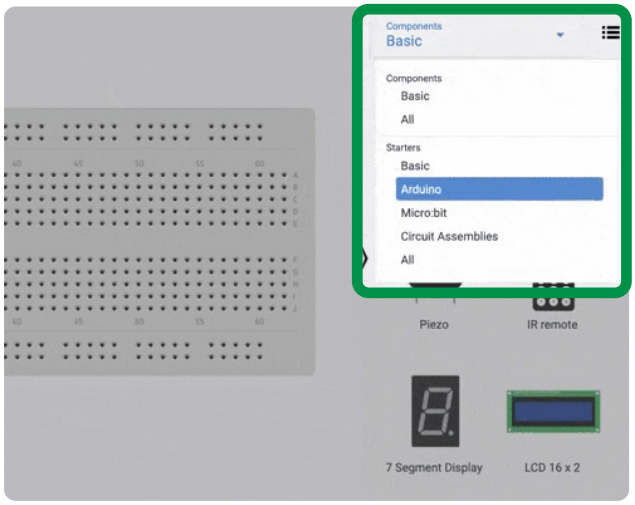
A "Zoom to fit" button is located in the top left corner of the workspace, which will center and zoom your design to fill the window. Pressing the letter F on your keyboard works as a handy shortcut for this same command.



If this is your first time using the Circuits editor, we encourage you to explore the different buttons and options available to you in the menu bar across the top. Hovering your mouse over any of the buttons should reveal an explanation for what it does, as well as any keyboard shortcuts that accomplish the same command.

Likewise, spend a minute browsing some of the other options and menus available in the component panel. By default, the Circuits editor presents you with a selection of the most popular basic components for learning electronics. To access more components, use the dropdown menu to select the All Components view, or search for specific components using the search bar beneath the menu.

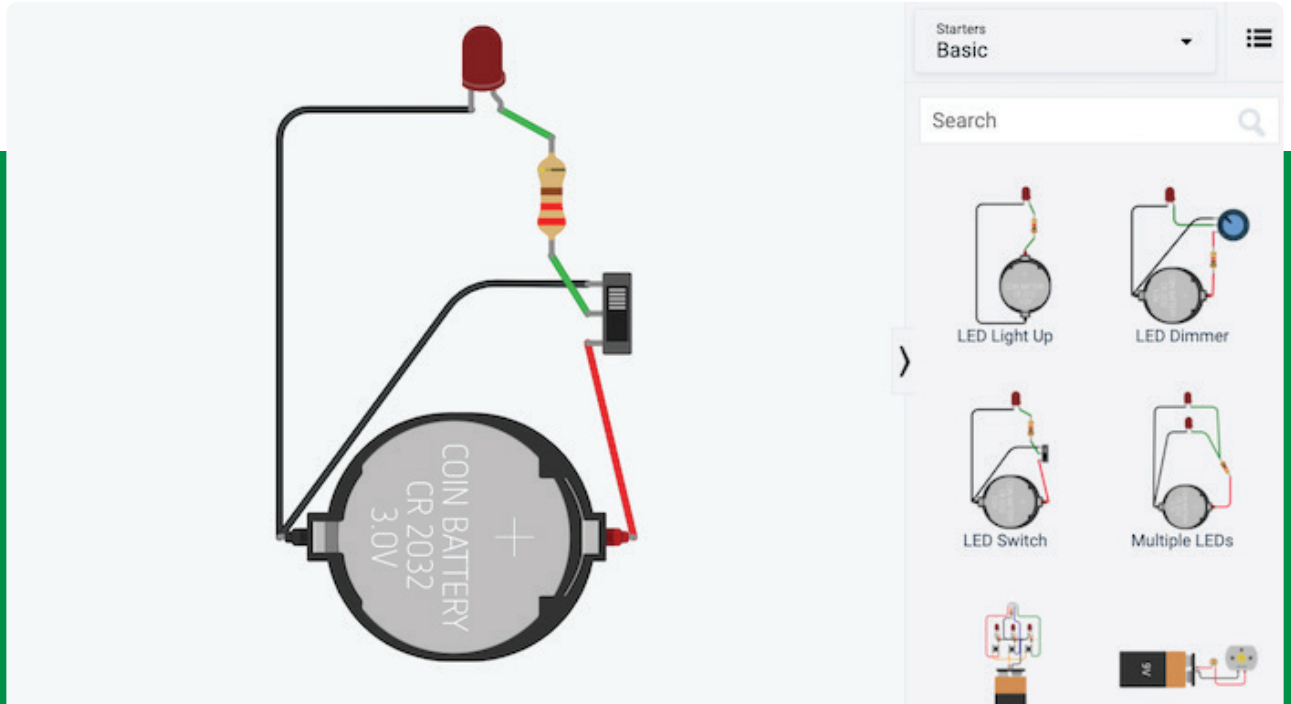




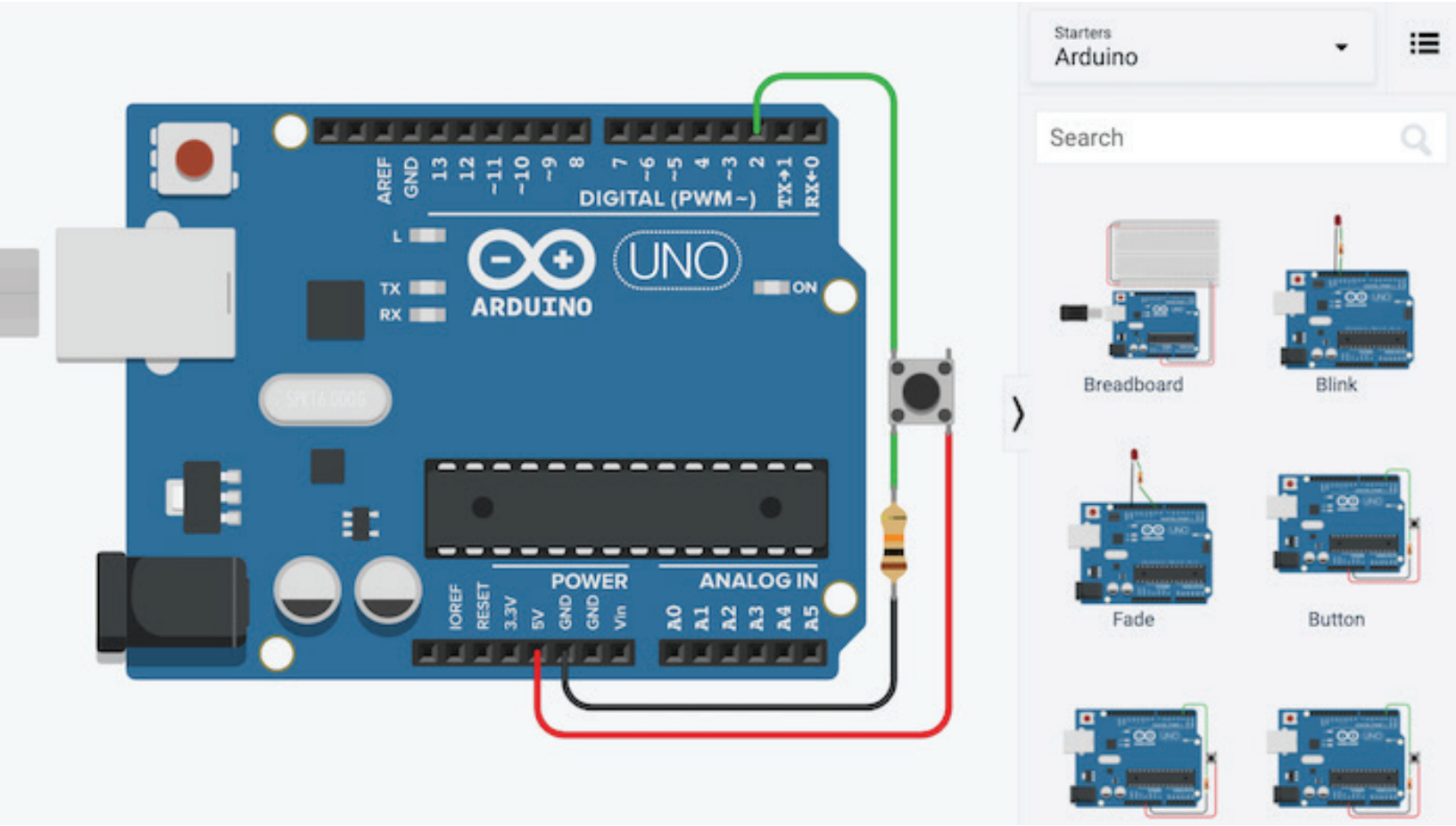
You'll also notice that we have more than just components in this menu. Further down, you'll find a selection of Starters. These are pre-made circuit examples that students can drag into the workspace, simulate, edit, and remix.

These Starters fall into four main categories: Basic, Arduino, Micro:bit, and Circuit Assemblies. Every one of our Starters comes to life in some way when the Start Simulation button is pressed.

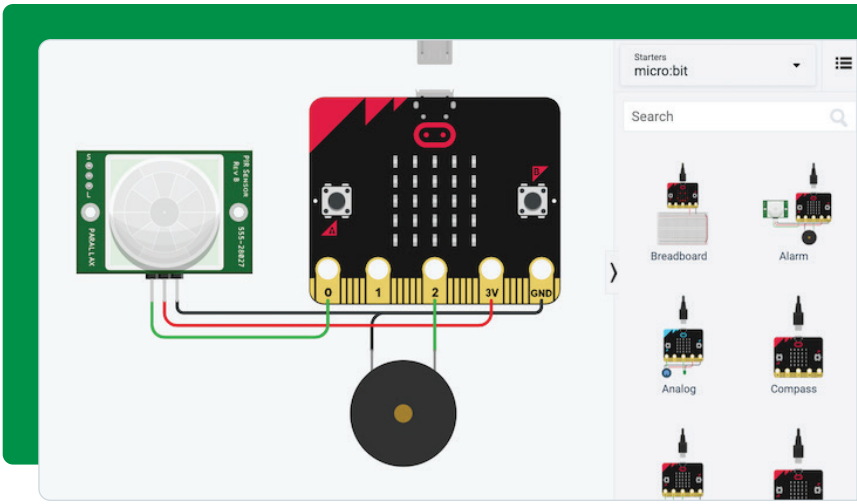
Basic Starters are made from the kind of common electronic components typically used to introduce students to electronics (LEDs, batteries, hobby motors, resistors, and switches). These examples use no microcontrollers, and no code.



With Arduino Starters, students can see the kind of advanced interactions that are possible with programmable microcontrollers. Each of the Arduino Starters include a code view, which students can directly edit using a built-in blocks-style interface, a text-based editor, or a combined view.

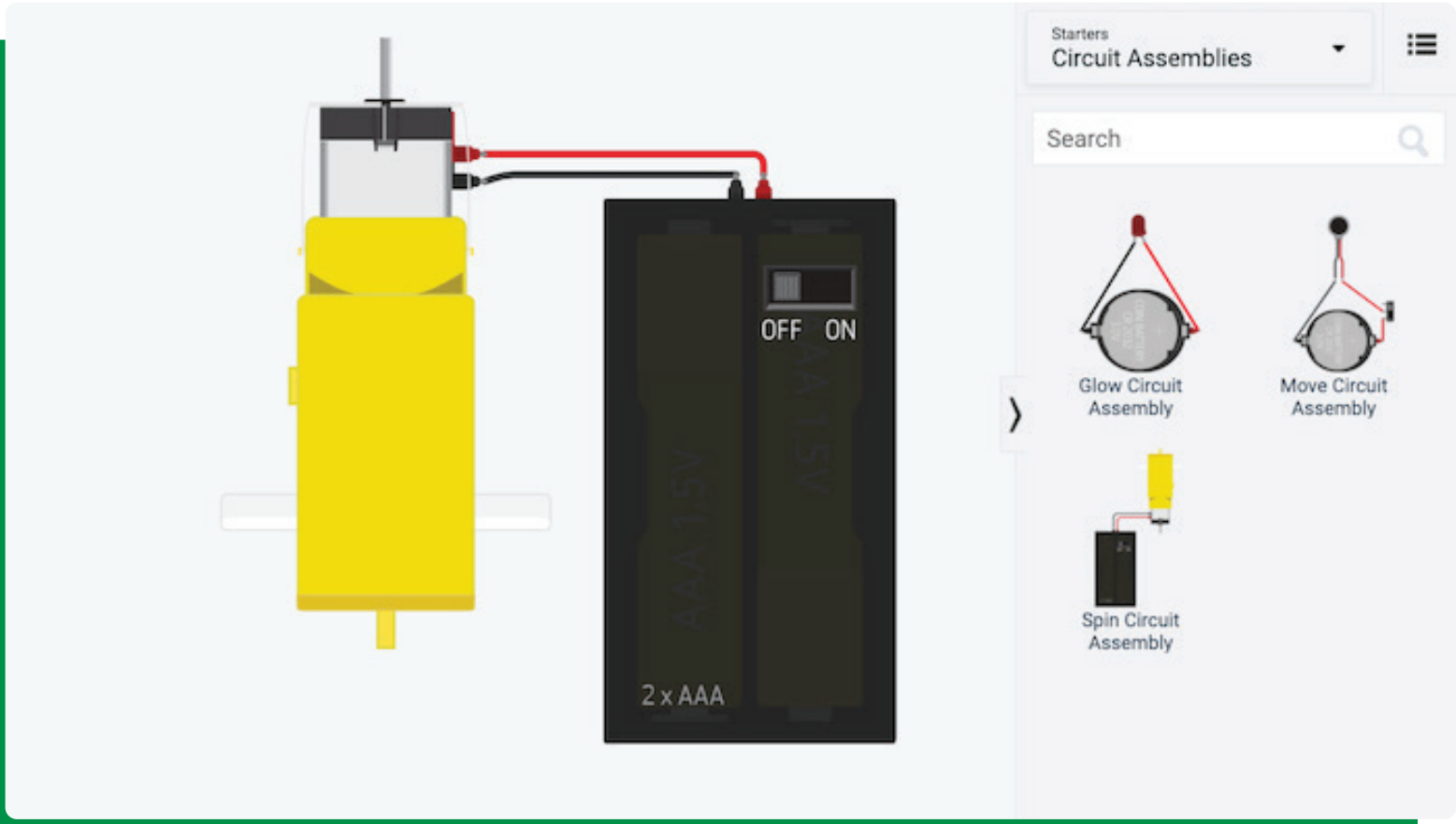


Our most recent addition are the micro:bit Starters. Similar to the Arduino starters, these are microcontroller projects that use the popular micro:bit educational electronics board. They also include an editable code view.



Finally, we have the Circuit Assemblies Starters. There are only a handful of these and they're directly linked to projects that tie-in both 3D design and basic electronics. These projects include:

- Glow Circuit at **autode.sk/tinkercad-glow**
- Move Circuit at **autode.sk/tinkercad-move**
- Spin Circuit at **autode.sk/tinkercad-spin**



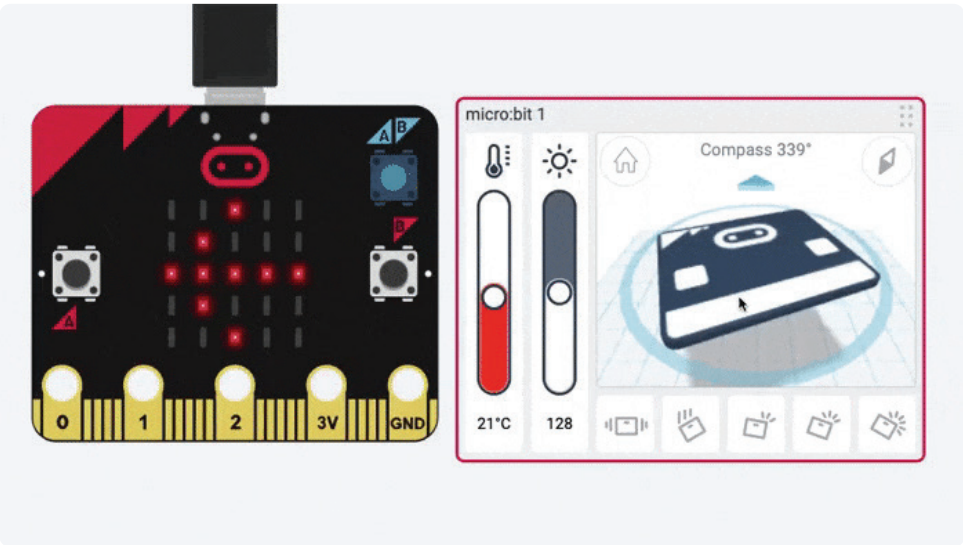
If you've exhausted our Starters and you're still hungry for more our Gallery page, at **autode.sk/tinkercad-gallery-circuits**, includes a selection of community contributed designs to spark your inspiration. Alternatively, you can use the Circuits view of our search tool, at **autode.sk/tinkercad-search-circuits**, to locate specific designs that may be useful to you.

02 Micro:bit in Tinkercad

The BBC micro:bit is a popular and inexpensive circuit board designed for students to learn electronics and coding. As the newest addition to Tinkercad circuits, it benefits from having our most recent starter examples. Learn more about BBC micro:bit at microbit.org.

Micro:bit Starters

The quickest way to start exploring micro:bit in Tinkercad is to drag out one of the example designs from the Starters menu. Each of these designs can be brought to life in some way using the simulation mode, and include code that students can copy, modify, and build on.

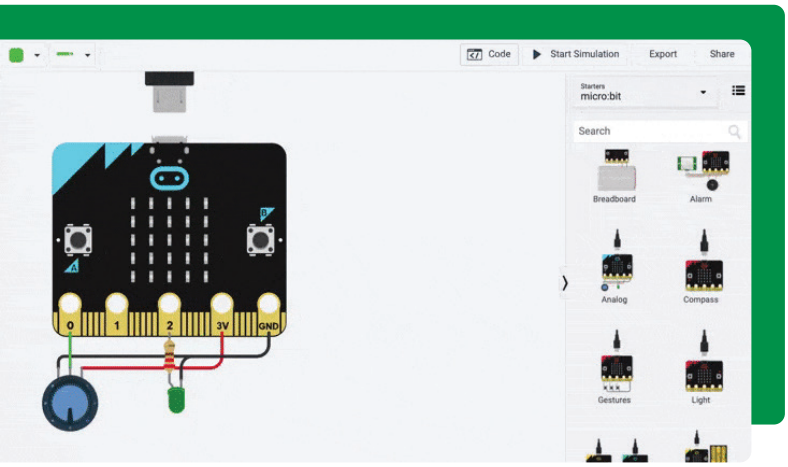


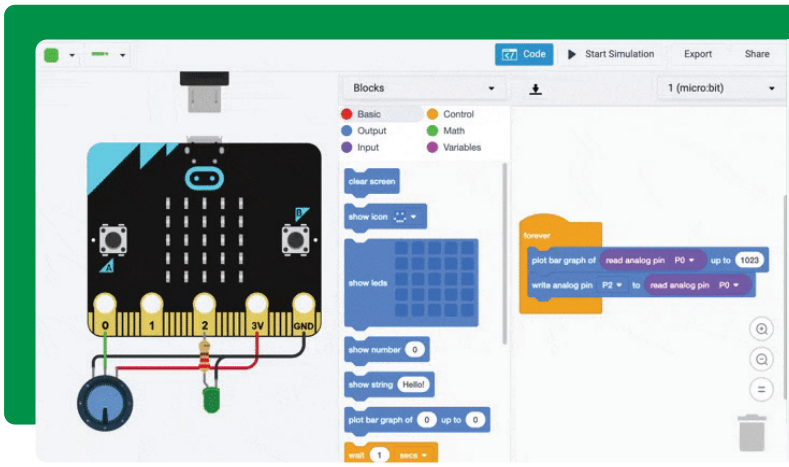
Coding micro:bit

To view or edit code on any of our programmable circuits (including Arduino), simply press the Code button to toggle the coding tools in and out of view.

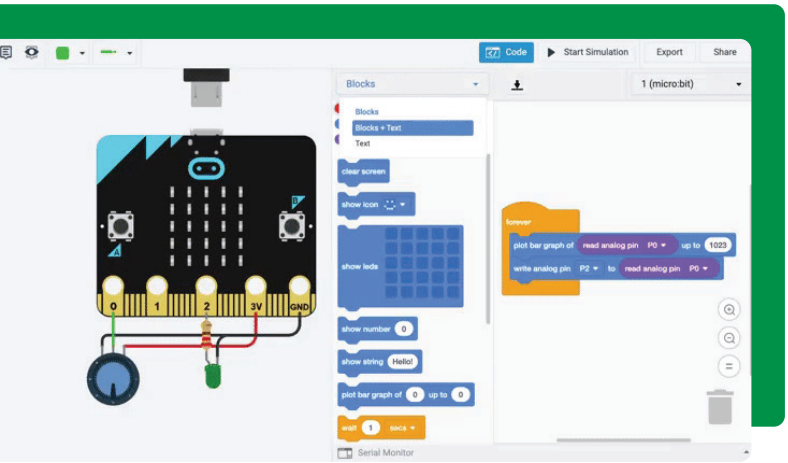
Students can code their micro:bit creations using a Scratch-based system of code blocks, similar to the system used on Microsoft's MakeCode website. Coding with blocks is a fantastic way for beginners to visualize the logic and patterns of code.

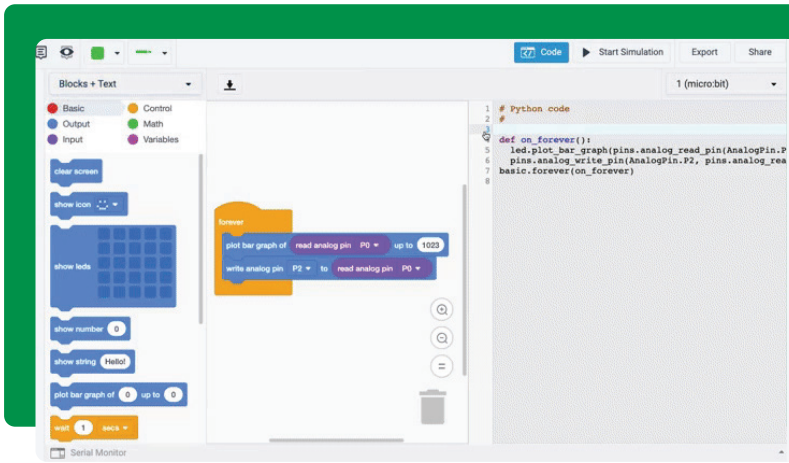
As you can see from the Starters examples, in spite of its approachability, blocks programming is capable of some relatively sophisticated interactions. We like to call it a “low floor, high ceiling” programming language.



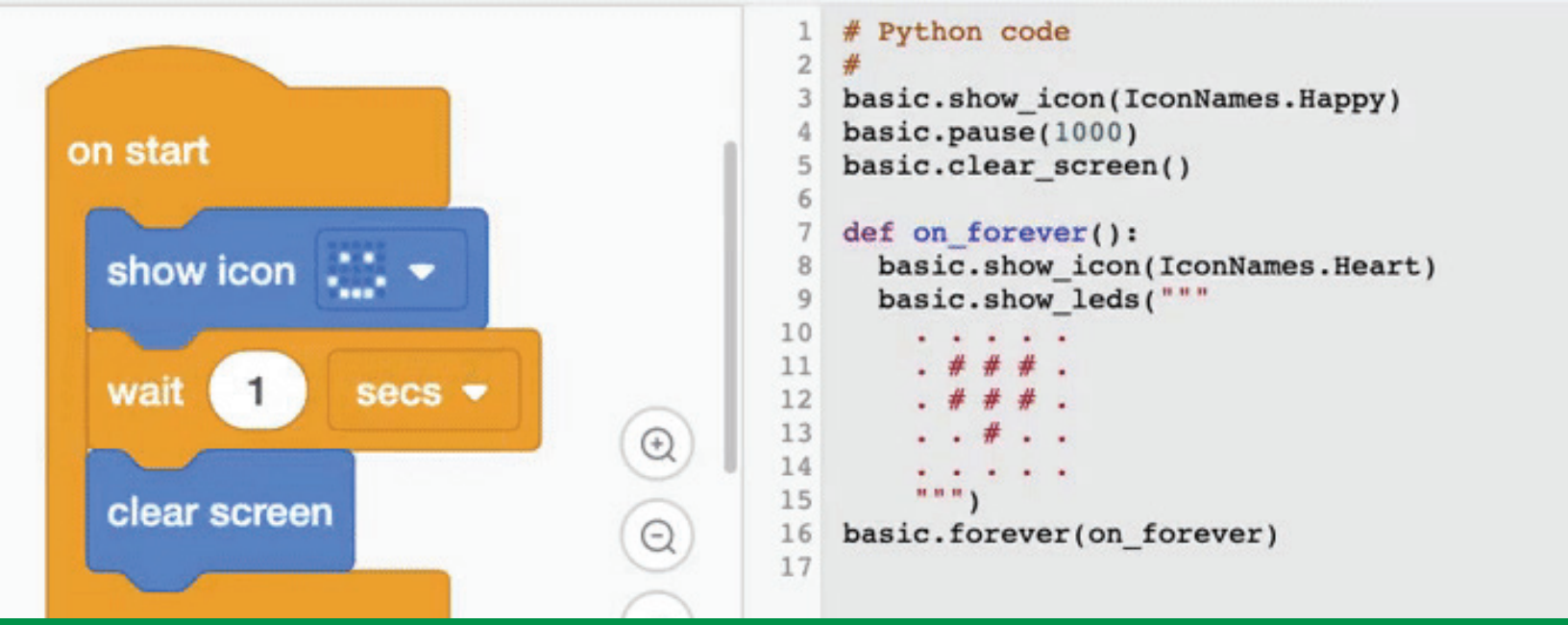


That said, your more advanced or adventurous students have the option of exploring micro:bit coding using the popular scripting language of Python. This option can be found using the dropdown menu above the blocks code tools.





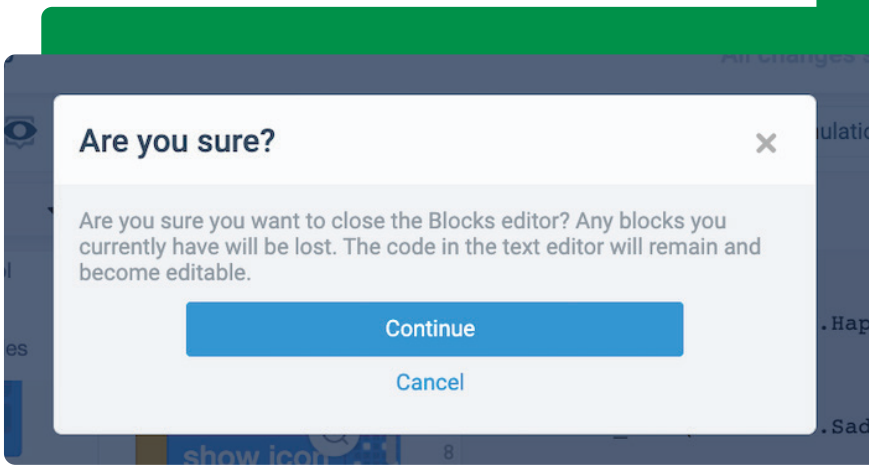
Students can change this code view from the default Blocks option, to a Blocks + Text (Python) view, or a purely Text view. The Blocks + Text view is perfect for beginners, allowing them to explore coding possibilities using familiar blocks while seeing the same concepts represented in the written language of Python.



Please note: Moving your blocks code to Python is a one-way street. When changing any blocks-based code to a fully text-based option you'll be presented with a warning that the conversion cannot be undone.

Any attempt to revert a design back to a Blocks view of the code will erase the code. We've done our best to make this as clear as possible within the editor to prevent students from accidentally deleting their code. As their instructor, you may want call special attention to this and have students duplicate a backup of their designs before exploring the different code environments.

For more information on our Python integration for micro:bit (including our Python debugging tool) check out our blog post titled Python Coding with micro:bit in Tinkercad Circuits **autode.sk/tinkercad-python**.

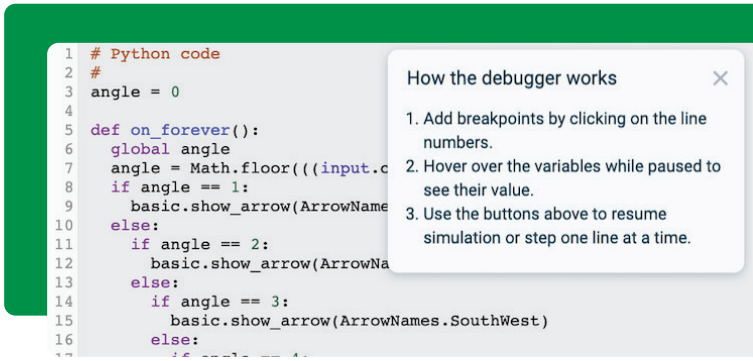


Debugging micro:bit Code

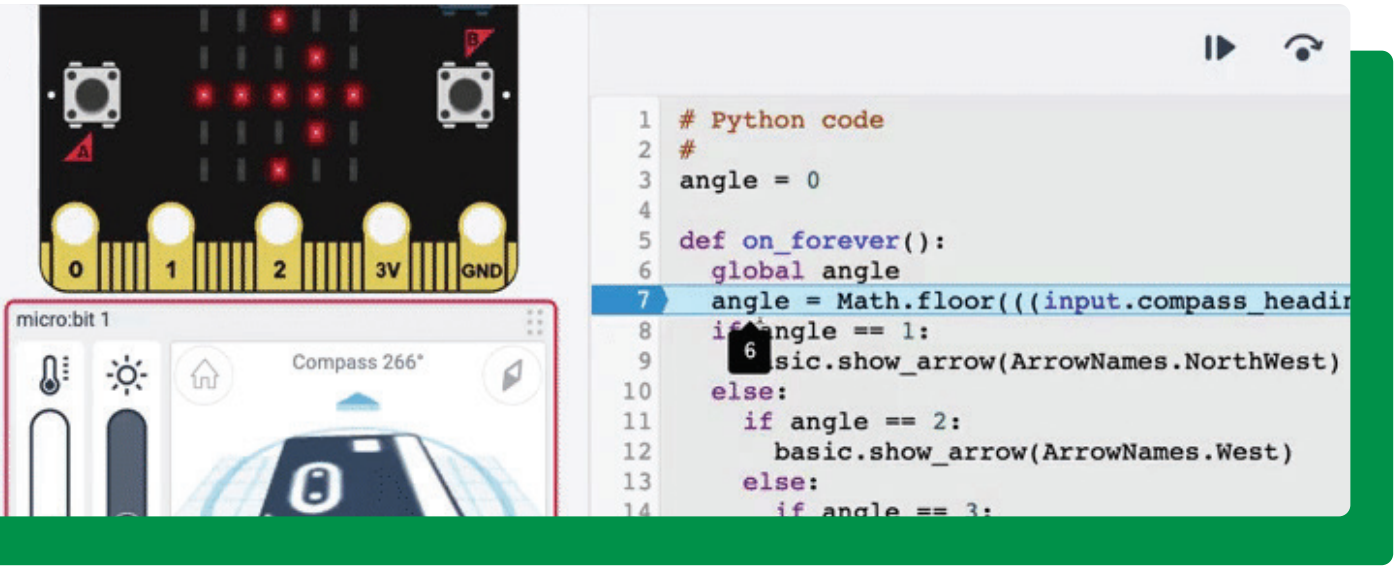
If your code isn't quite up to snuff, our built-in debugger will let you know. The debugging interface will kick-in automatically if it detects an error in your code.

Another advantage of debugging is that you can add breakpoints in your code. By selecting a line number in your code, the highlighted line marks a point where the code will pause during simulation.

When debugging with break points, two buttons will appear above the code window during simulation. You can use these buttons to either resume the execution of the code after the break-point, or step to the next line of code.



Adding breakpoints is also a useful way to reveal variable values while simulating your project.

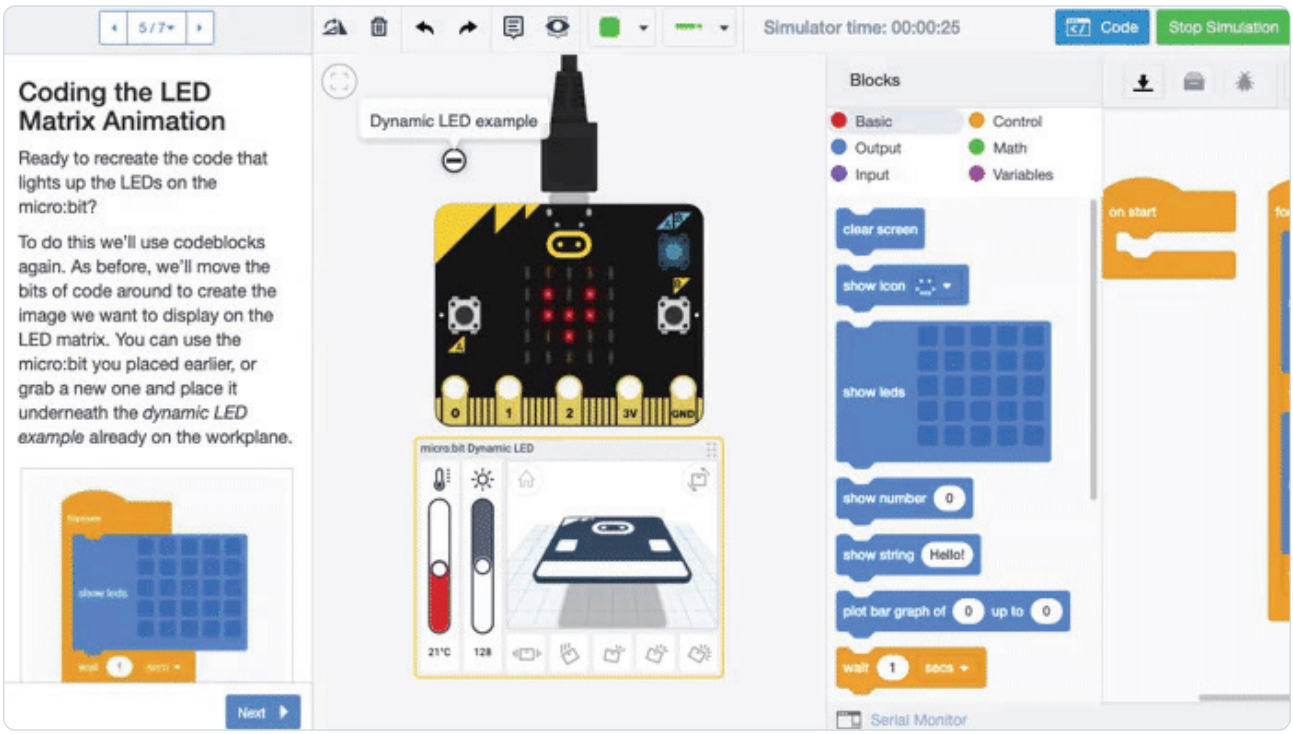


The example above shows the Compass project located in our Starters menu. When the simulation is paused at a breakpoint anywhere in the code, you can hover over a variable (in this case "angle") and see its value at that moment in time. For this example, we can play around with the board's orientation and check to see if the math accurately indicates the direction.

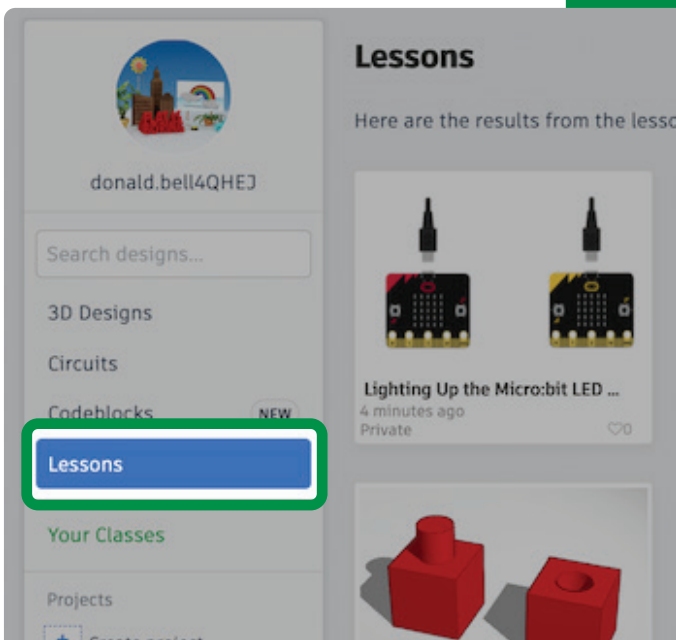
Micro:bit Lessons

For a more structured approach to learning the fundamentals of micro:bit, students can work through our self-paced, interactive lessons at **autode.sk/tinkercad-learn-microbit**.

There are five distinct lessons within this project. Similar to our interactive lessons for basic circuits and Arduino, students are provided with a sidebar outlining the instructions they need to follow to complete each task.



If you're using Tinkercad Classrooms to onboard and manage your student experience with Tinkercad, you'll be able to see a student's completed lessons in the Lessons section of their design gallery, or within the Designs section of your class dashboard.



03 Arduino in Tinkercad

For over 15 years, the Arduino ecosystem of microcontroller boards and code libraries have become an indispensable part of electronics education. But if you've ever worked with a student to connect an Arduino board to their computer, download Arduino's IDE software, update the libraries, and configure the ports, you know that it can be a bumpy road.

The Arduino simulation within Tinkercad simplifies the learning experience. It's free, works on any computer with an Internet connection, and scales to any class size. Best of all, Tinkercad Circuits provides a bottomless supply of virtual components that students can use to build and simulate their projects.

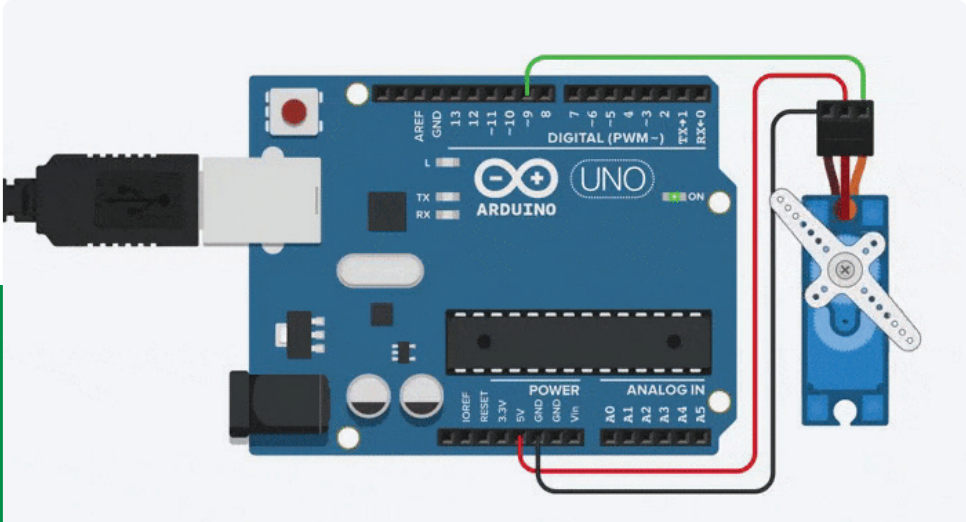
When they're ready to physically prototype their projects, Tinkercad Circuits makes it easy to export their code as a native Arduino (.ino) file that they can upload to their board.

Learn more about Arduino at **www.arduino.cc**.

Arduino Starters

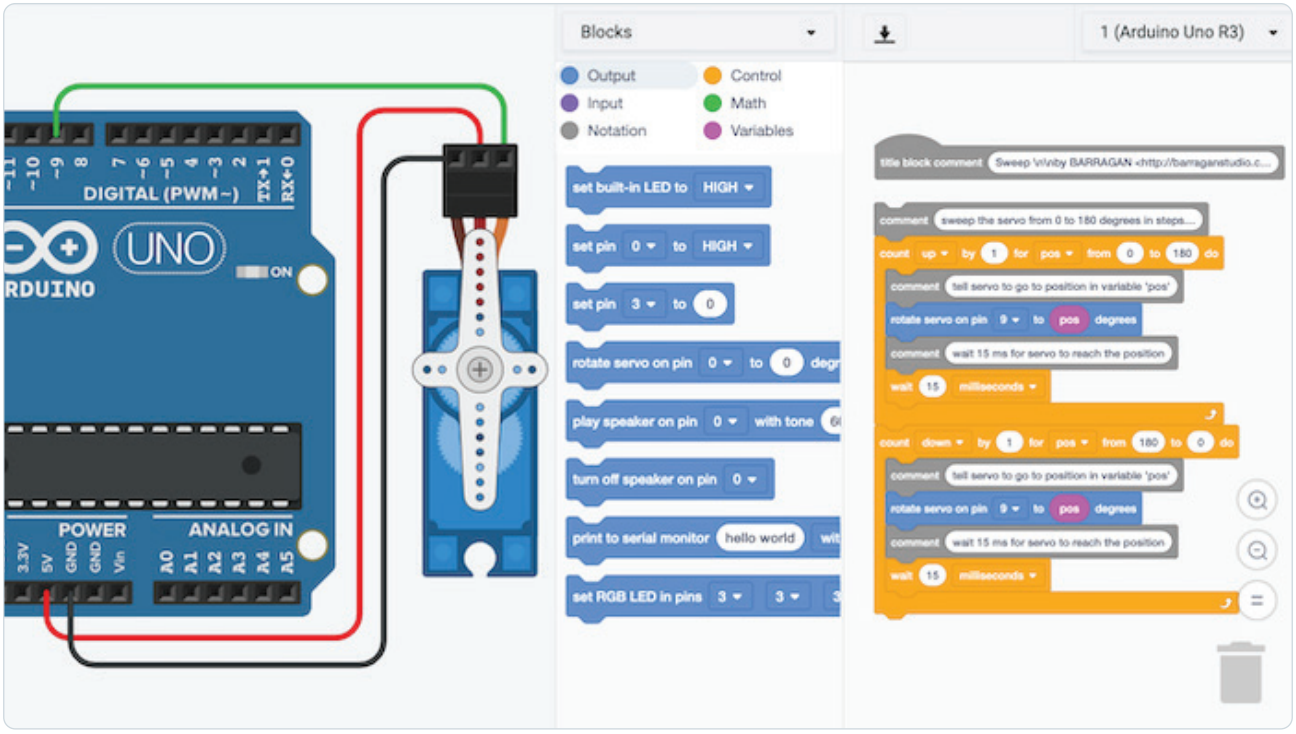
One of the fastest ways to explore the possibilities of Arduino in Tinkercad is to simply drag one of our sample Starter circuits into your workspace.

We have nearly two dozen Arduino Starter circuits to choose from. Each example includes sample code that you can view, simulate, and modify.



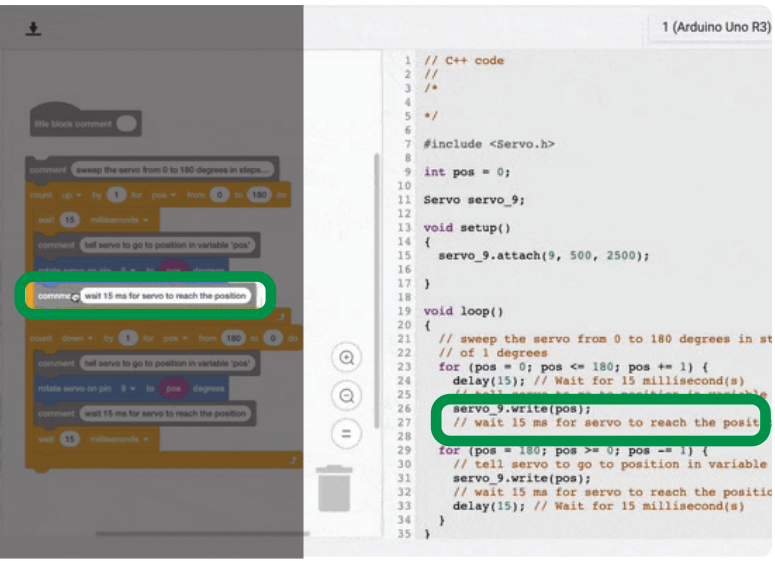
Coding Arduino in Tinkercad

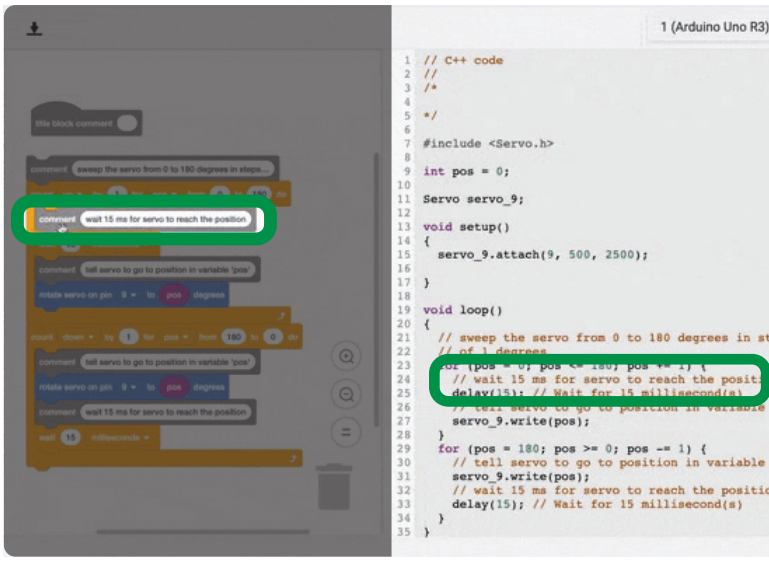
Tinkercad allows you to code your Arduino using two different approaches. Our Blocks code editor offers beginners a visual system of functions that they can drag and rearrange. All of our Arduino Starters, and most of our interactive Arduino lessons will include or refer to Blocks code.



Part of the magic of learning to code Arduino in Tinkercad is that our editor will automatically generate text-based code (C++) from students' blocks code. By switching the code view to Blocks + Text, students can see the logic of their blocks code translated to C++ code.

Modifications made to their blocks code will instantly update in the text view, providing insight into the logic and syntax of C++.

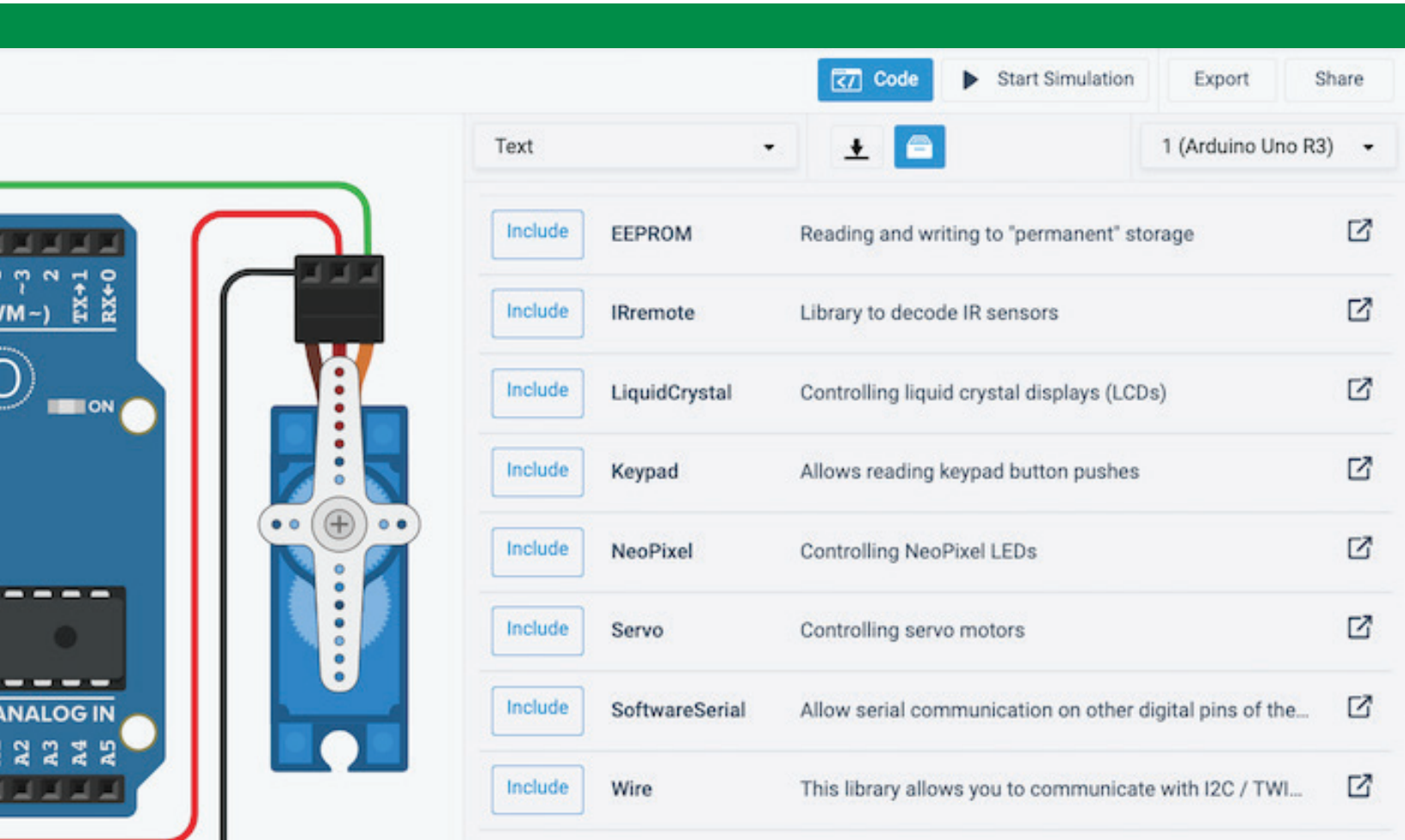




Of course, once students are ready to create their code directly in the text editor, they can switch to a pure text view. This view offers an experience similar to programming with Arduino's IDE editor.

Using Arduino Code Libraries

Tinkercad's Arduino text editor includes eleven built-in libraries. You can view and add these libraries to your project by selecting the file box icon above your code.



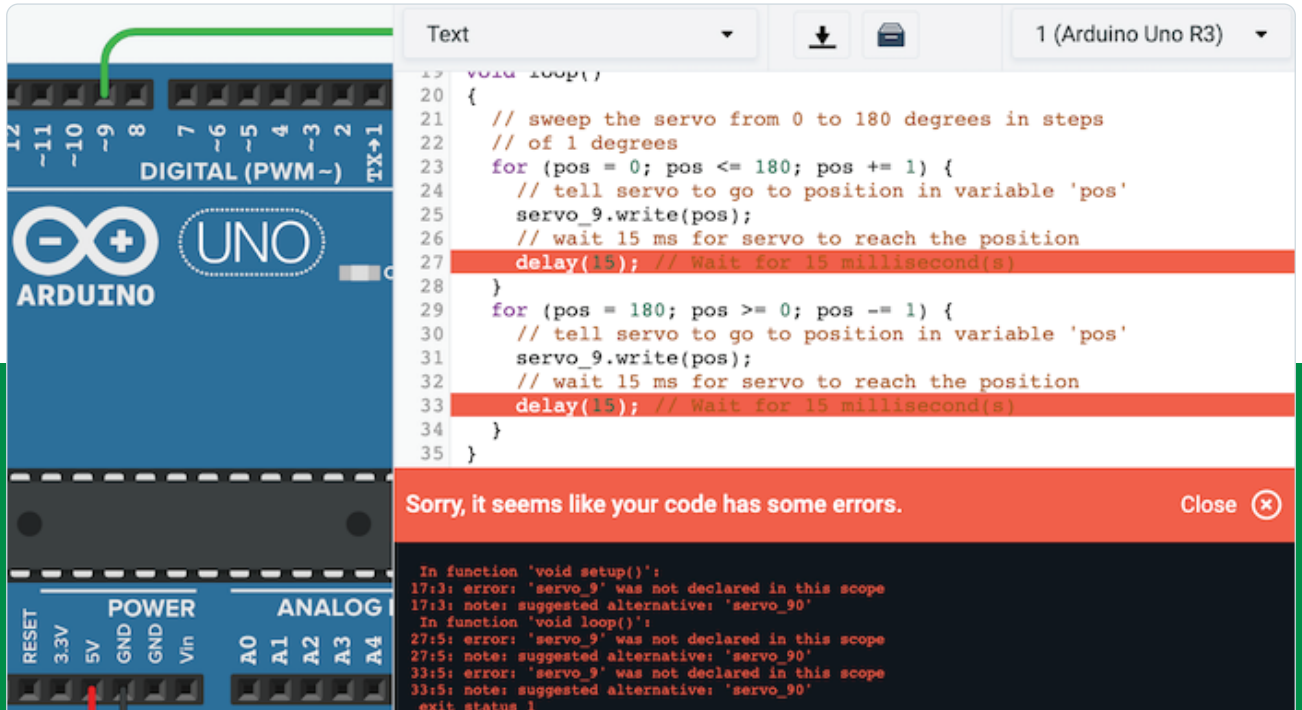
The included libraries represent some of the most popular and common libraries used in Arduino. It is possible, though, to run code that requires Arduino libraries beyond the included examples.

When you open an Arduino library source file (.c or .cpp) you'll find that it is simply a clipping of C++ Arduino code. By copying and pasting this library clipping into the appropriate sections of your Arduino code, you may be able to effectively make it work.

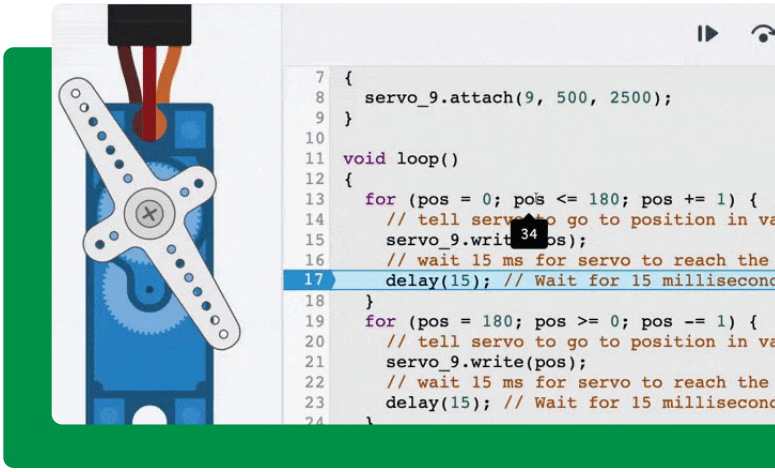
Your mileage may vary. Libraries are oftentimes made to adapt Arduino to specialized hardware or shields. If these hardware components aren't included in Tinkercad, no amount of code editing will make the project work.

Debugging Arduino Code

One of the biggest challenges in learning to code is learning how to troubleshoot problems when the code isn't working as planned. When a student's code doesn't work, Tinkercad's error console will automatically pop into view. Similar to how errors are reported in the Arduino IDE, this error console will report and highlight the suspected issues that need fixing.



Tinkercad's Arduino debugger also includes a unique and useful code break feature. By selecting lines of code, you can define moments within your code where you'd like the simulation to pause. These pauses will happen automatically automatically during simulation, allowing students to read values and troubleshoot problems. A button above their code resumes the simulation or advances to the next break.



Arduino Lessons

Our selection of Arduino lessons, at **autode.sk/tinkercad-learn-arduino**, provides students with an interactive, self-paced system for learning how to work with electronics using Arduino.

As students open these lessons, they'll find a sidebar on the left with instructions on how to complete their design.

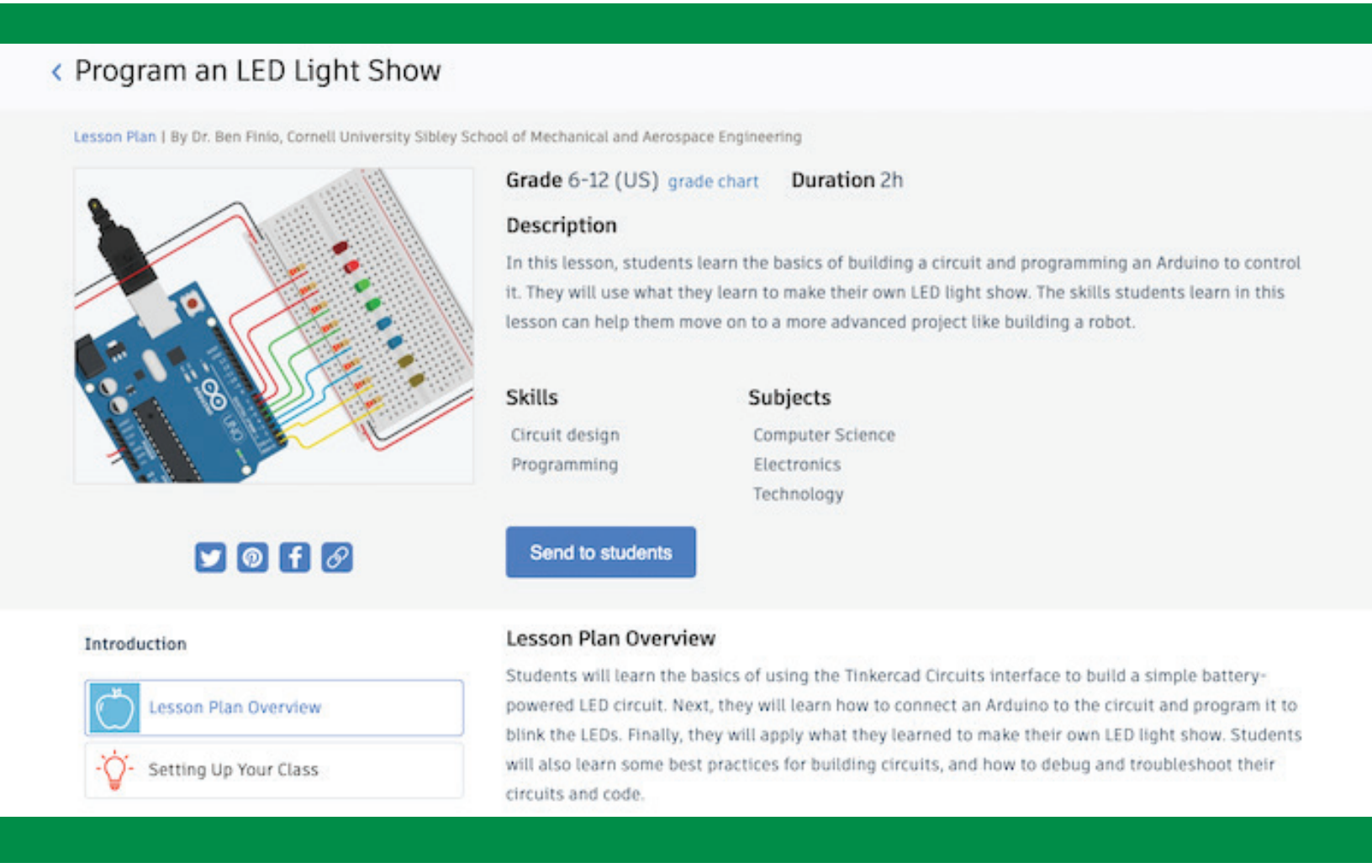
There are 28 lessons available at this time. The last 15 (represented with green title cards) are taken from the Arduino Projects Book by Scott Fitzgerald and Michael Shiloh. It is the book that Arduino includes in their official Starter Kit. Companion videos for these lessons can be found on Arduino's Remote Learning Playlist.

If you're using Tinkercad Classrooms to onboard and manage your student experience with Tinkercad, you'll be able to see a student's completed lessons in the Lessons section of their design gallery, or within the Designs section of your class dashboard.

Arduino Lessons Plan

Program an LED Light Show is a free, standards-aligned lesson plan developed by Dr. Ben Finio of Cornell University's Sibley School of Mechanical and Aerospace Engineering. In it, students learn the basics of building a circuit and programming an Arduino to control it.

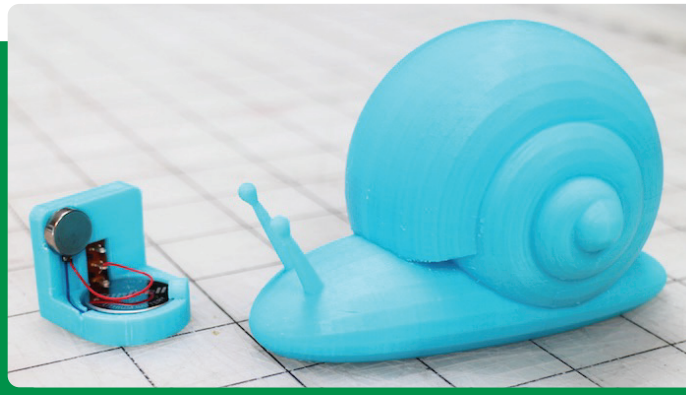
The lesson plan comes complete with video resources and PowerPoint slides and is geared towards students in grades 6-12 (US). This lesson is available at **autode.sk/tinkercad-lesson-LED-show**.



04 Bridging 3D Design and Circuits

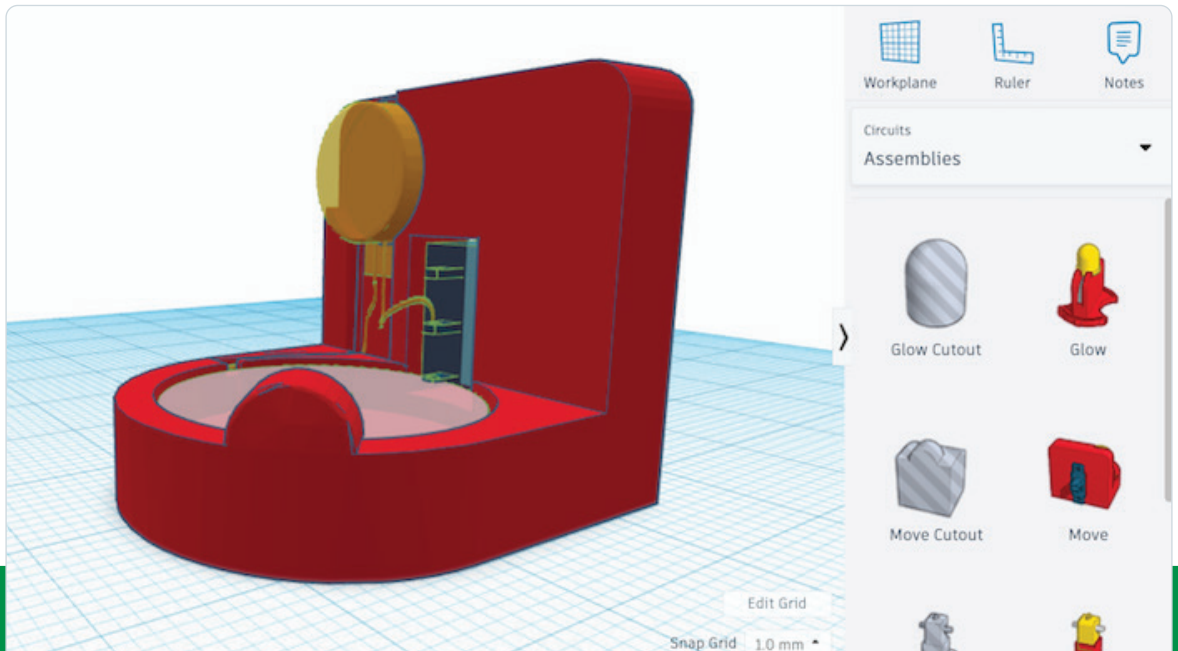
Because Tinkercad works as both an electronics editor, and a 3D design tool, naturally there are opportunities to bridge these two capabilities

On Instructables, you can find several classroom-ready activities that weave together basic electronics, 3D design and 3D printing. These activities include the Glow Circuit at **autode.sk/tinkercad-glow**, Move Circuit at **autode.sk/tinkercad-move**, and Spin Circuit at **autode.sk/tinkercad-spin**.



For an Arduino-based activity that combines 3D design and electronics, check out our guide for creating a LED Ring Butterfly at **autode.sk/tinkercad-LED-ring**.

In each of these examples, students will find pre-built 3D components in Tinkercad's 3D editor, under the Circuits Assemblies menu. The Circuits Components menu compliments these designs with accurate 3D representations of popular components.



05 General Tips

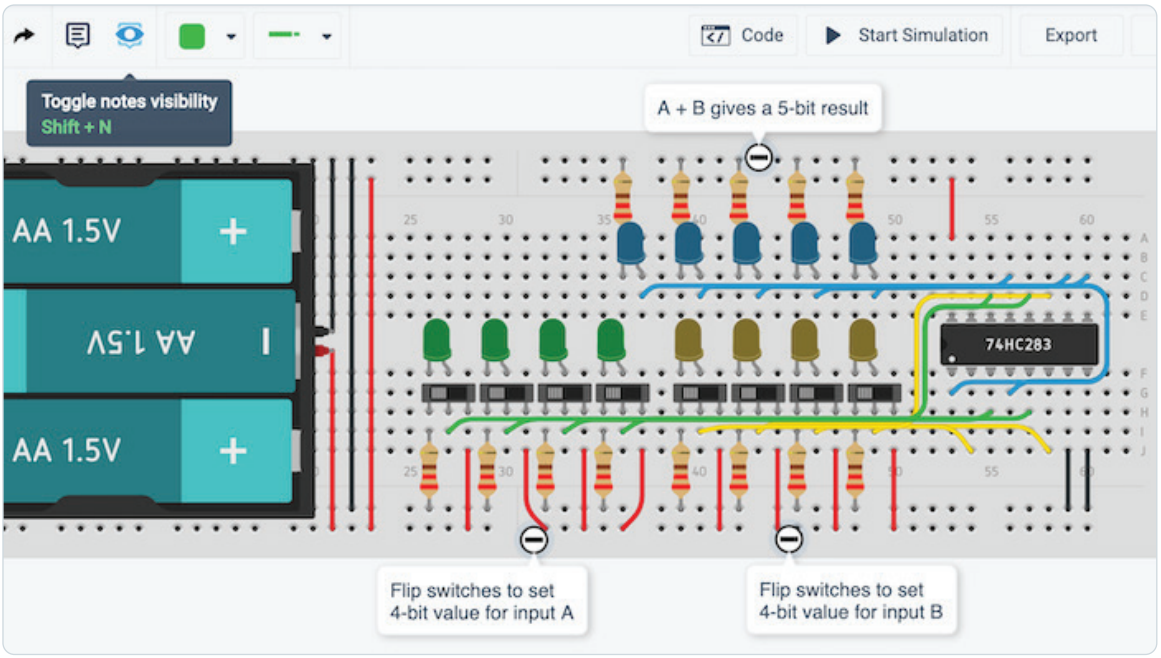
Micro:bit Starters

Within the Circuits workspace there are some handy keyboard shortcuts worth knowing.

- F - Zoom to fit
- N - Create note
- Shift + N - Hide/reveal notes
- R - Rotate
- You can quickly change the color of a wire by selecting it and pressing any number key (0-9).
- By holding shift you can select multiple items in the workspace to move or delete.

Annotation

Both educators and students can take advantage of the Notes tool for explaining designs or provisioning feedback. The Notes feature can be found in the toolbar above the Circuits workspace (or activated using the N key on your keyboard).



Once a note has been placed on a design, its placement can be further adjusted to prevent it from obscuring components. If too many notes are cluttering a design, their visibility can be toggled on and off using the button with the eye-shaped icon.

Sharing Designs

By default, anything you or your students design in Tinkercad is considered private. Private designs can't be shared between users and they will not appear in Tinkercad Gallery search results.

To share a Tinkercad Circuits design with your students you must first make it public. It's an easy change to make using the design's Properties window. We also have a step-by-step guide on making and sharing public designs in Tinkercad at **autode.sk/tinkercad-sharing**.

For a teacher to view a student's design, a student must have an individual account (in which case they can follow the steps outlined in the link above), or be a part of a Tinkercad Classrooms account that you manage. Learn about Tinkercad Classrooms at **autode.sk/tinkercad-guide-classrooms**.

Public designs also include HTML code that you can use to embed your designs on a website. Learn how **autode.sk/tinkercad-embed**. This can come in handy for including designs within a course overview page for your class, or as part of an Instructables guide.

06

Teaching Circuits with Tinkercad

In our ongoing webinar series Teaching with Tinkercad we feature educators who are putting Tinkercad to use in their classrooms and sharing what they've learned. View the complete series at **autode.sk/tinkercad-webinar**.

In the episode Making Hands-On Work in a Digital World (Ep. 3) you'll hear from Panion Tase on how he uses Tinkercad to introduce basic electronics to his students. View at **autode.sk/tinkercad-webinar-ep3**.

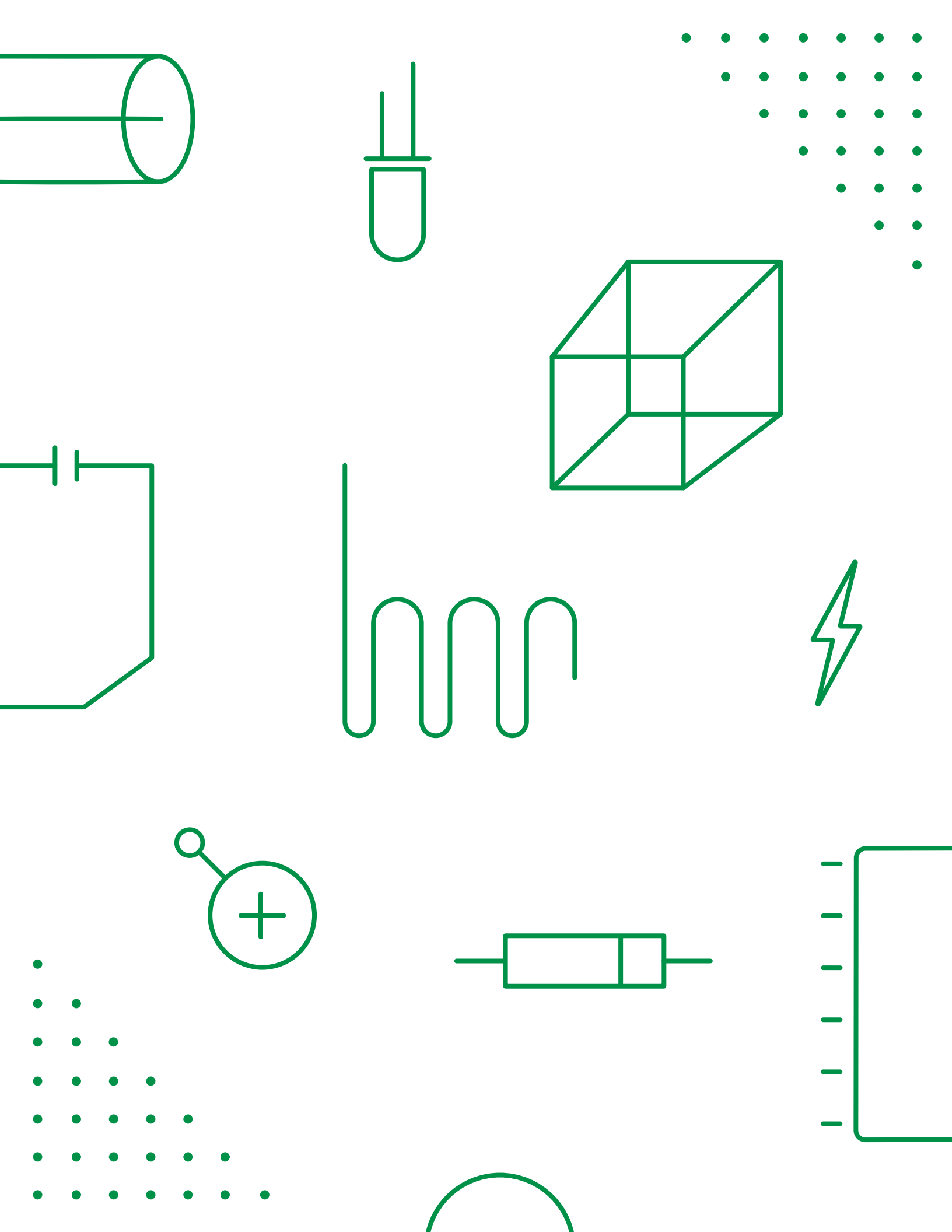
Later, in episode 5, educators Dori Friedman and Becky Stern will show you how to teach with micro:bit and Arduino in Tinkercad Circuits.
View at **autode.sk/tinkercad-webinar-ep5**.

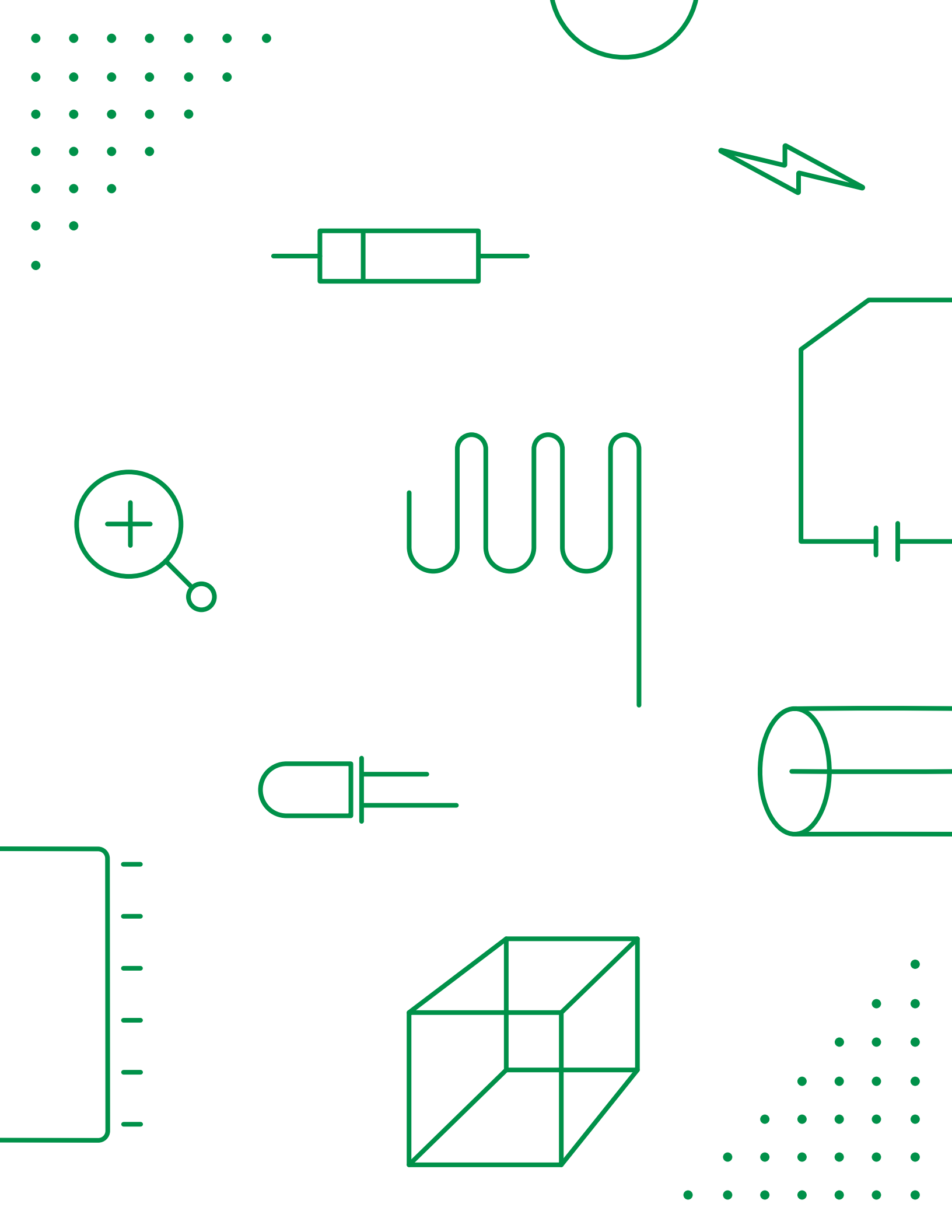
Support & FAQ

Our Tinkercad Knowledgebase includes an extensive FAQ for Tinkercad Circuits at **autode.sk/tinkercad-faq-circuits** for any questions you may have that aren't covered here. From this same page, you have the option to create a ticket with our support team for additional help.

We created Tinkercad as a tool to empower anyone in the world to learn how to design and make things, and shape the world around them. Educators like you make this all possible, and if there's anything we can do to improve the Tinkercad experience for your classroom, don't hesitate to send your suggestions to **tinkercadlearning@autodesk.com**.

-Team Tinkercad





AUTODESK Tinkercad