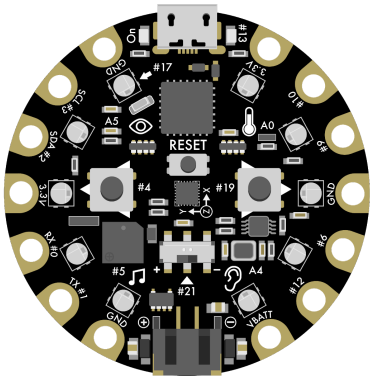
Maker Toolkit Troubleshooting Guide

Understanding the Different Elements

Programming with the Maker Toolkit and Circuit Playground requires several different elements to work together. In order to simplify troubleshooting, we should start by understanding what each of those parts is called and what they do.

Circuit Playground

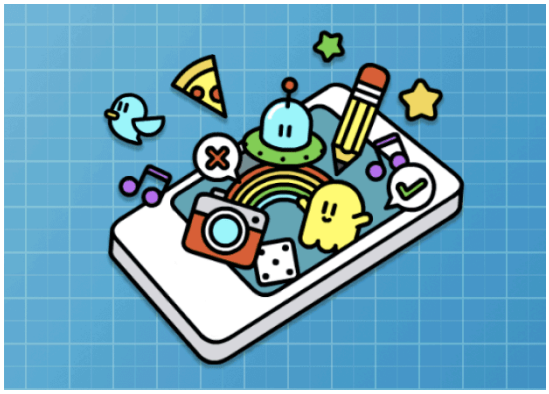
The Circuit Playground is the physical board that students create their projects with. It plugs in to the computer via USB, which it uses to communicate with App Lab and the Maker Toolkit. The Circuit Playground needs to run a special piece of software called firmata in order to communicate with App Lab.



Circuit Playground Versions

Adafruit produces two different versions of this board, the Circuit Playground Classic and the Circuit Playground Express. The Classic is a more affordable board built on the Arduino platform, while the Express uses a more advanced processor that allows it to support a wider variety of programming languages and environments, including Microsoft MakeCode and Circuit Python. At this time **we only support the Circuit Playground Classic**, but we are actively working with Adafruit to bring beta support for the Express.

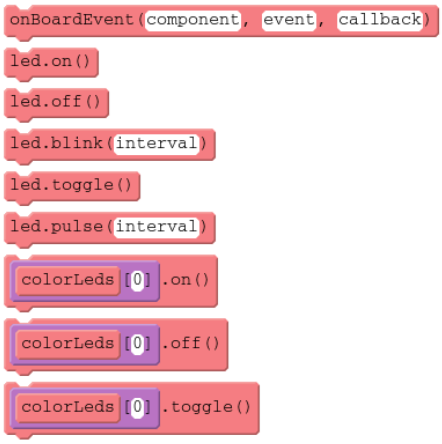
App Lab



App Lab is the programming environment that students use to program the Circuit Playground. It is the same environment that students use in CS Discoveries Unit 4 to learn the design process, and is also used in CS Principles. App Lab doesn't actually upload any code to the Circuit Playground. Instead, it communicates with the firmata software on the board over USB. This means that boards can be easily swapped among multiple students without worrying about what code was last on there, but it also means that the board must be connected to a computer for student code to run.

The Maker Toolkit

App Lab is a programing tool designed for mobile app development. In order to write programs that communicate with physical hardware, we need to add some custom commands. The Maker Toolkit is a set of additional commands that are designed to communicate with the Circuit Playground. This is effectively a library of blocks that can be turned on or off within any App Lab project.



The Maker App / Chromebook Extension

The element that makes this whole thing work is an app that can facilitate communication between App Lab and the Circuit Playground. If you are using a standard desktop operating system (Windows, Mac OS, or Linux), the Maker App is a standalone app that you can install which allows apps running on Code.org to communicate with the hardware plugged into your computer. If you are using a Chromebook, there is a separate Chromebook Extension that allows the Chromebook browser to communicate with the Circuit Playground.

Arduino Drivers

Many computers, including all Macs, come with support for the Arduino hardware that makes up the Circuit Playground. Some computers however, particularly older Windows computers, will need to have a special hardware driver installed so that the operating system can recognize the Circuit Playground.

Getting Set Up

Getting set up is a pretty straightforward process. There are a few situations that can require additional steps, which we'll address at the end of this section.

Chromebooks

1. Install the Code.org Serial Connector Chromebook Extension
2. Plug in your Circuit Playground
3. Visit studio.code.org/maker/setup to test your installation.

Other Platforms

1. Install the Maker App from studio.code.org/maker/setup
2. Plug in your Circuit Playground
3. Open the Maker App and click the gear icon to test your installation

Dealing with Installation Test Errors

Adafruit drivers installed (Windows Only)

This test can't actually test if your drivers are installed, but will be triggered if the board can't be found. Follow the instructions on Adafruit's site to make sure you have the proper drivers installed.

Board plugged in

The most common reason for this test to fail is that you are using a charging-only cable. Many USB cables that come with phones, or are sold as phone charging cables, don't include the data wires necessary to communicate with your computer. Try swapping out with a USB cable that is known to work with data devices (such as an external hard drive). This test may also fail if you plug in an unsupported board, such as the Circuit Playground Express.

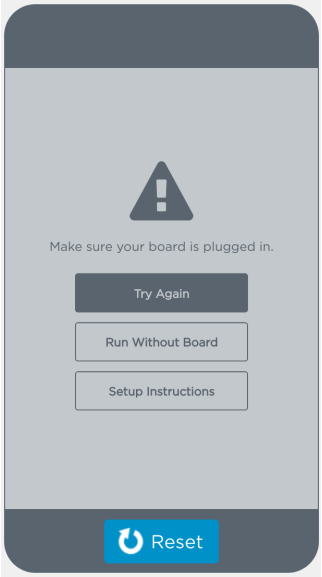
Board connectable

Failure at this point usually means that you don't have firmata, the special software that allows us to communicate with the board, installed. This could happen if you've used another tool, such as the Arduino IDE, to install a program to the board that would overwrite the firmata. Follow Adafruit's guide to installing firmata to resolve this.

Board components usable

If you hit this error, it means that we found firmata installed on your board, but it's not behaving like we expect it to. This could mean that you have an old version installed, or a firmata from a different source than Adafruit. Either way, follow Adafruit's guide to installing firmata to resolve.

Common Issues and Errors

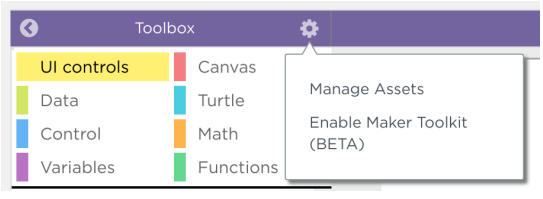


Make sure your board is plugged in screen

This error screen appears if App Lab can't connect to your board when you try to run a program. This could be caused by several different issues

- The app may have hiccuped when connecting, click the Try Again button.
- Your board may be stuck in an odd state because of a program that didn't release it properly. Click the reset button on the board (tiny button in the center) and then refresh the page.
- Your board may not actually be plugged in. Check the USB connections and try again.
- Something more substantial may be wrong. Click the Setup Instructions button and follow the troubleshooting steps from Getting Set Up

Maker Toolkit blocks missing



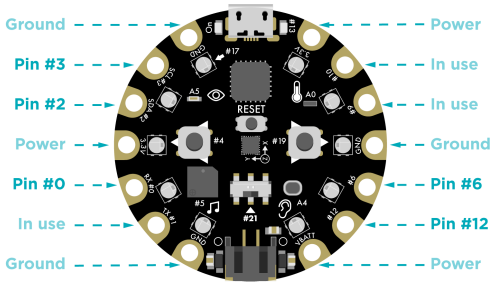
If you don't see the *Maker* category in your toolbox, it likely means that the Maker Toolkit has been turned off. Remember, this is an optional set of of blocks, which has to be activated. It should be active by default on all curriculum levels that require the board (please email us at support@code.org if it's missing on a curriculum level). When creating a new project from scratch, however, you'll need

to enable the Maker Toolkit. You can do this by clicking the gear in the top right corner of App Lab's toolbox and selecting "Enable Maker Toolkit".

External Buttons / LEDs not working

Towards the end of the unit, students learn to wire their own buttons and LEDs to the copper pads on the edge of the board. This is both an exciting opportunity to build devices that are more unique and personal and also a new layer of complexity that can require debugging. Issues with these circuits tend to fall into one of two categories:

Software Issues



- Pins 1, 9, and 10 are reserved for use by the software the communicates between the board and App Lab, so while they look like great places to wire a circuit, they can't be used. Instead, use pins 0, 2, 3, 6, or 12.
- When creating an new LED or button object, you won't find pre-built blocks in the Circuit toolbox, nor will you see the buttons referenced in the onBoardEvent dropdown - the Maker Toolkit doesn't know about the the circuits you wire in the way that it knows about the circuits built into the Circuit Playground. You'll need to switch to text mode in order to use circuits you create yourself, but you can always use the existing blocks for reference.

Wiring Issues

- Both buttons and LEDs need to form a circuit from a numbered data pin to a ground pin. Any break in that circuit (like wires that aren't cleanly touching, or a layer of tape insulating the connection) can cause it not to work.
- LEDs are polarity-sensitive, meaning that electricity can only flow through them in one direction. If an LED isn't lighting, double check that the plus side is connected to a data pin and the minus side is connected to ground.