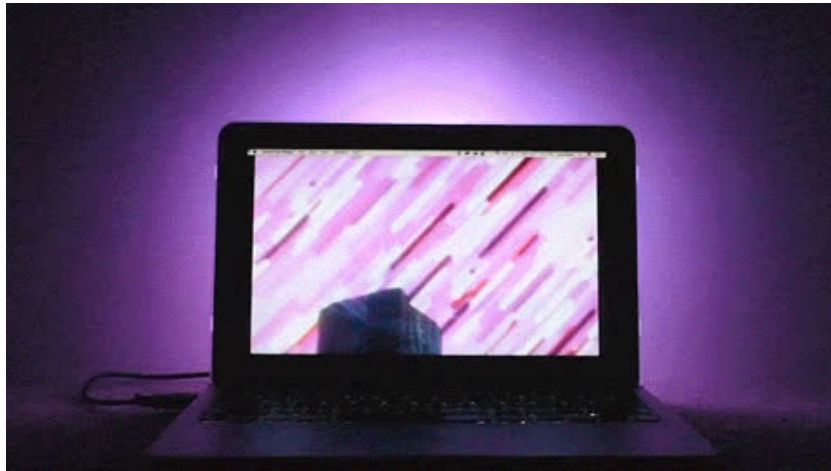


□

Adalight for Circuit Playground

Created by Phillip Burgess



Last updated on 2017-01-12 07:20:59 AM UTC

Guide Contents

| | |
|-----------------------------------|---|
| Guide Contents | 2 |
| Overview | 3 |
| Download & Install | 4 |
| Arduino IDE | 4 |
| Processing IDE | 4 |
| Adalight ZIP | 4 |
| Upload Code to Circuit Playground | 5 |
| Run the Processing Sketch | 6 |
| What It Does | 8 |

Overview



One of my first projects with Adafruit was [Adalight](http://adafru.it/tDD) (<http://adafru.it/tDD>), an ambient lighting effect for media PCs, similar to the *Ambilight* feature of mid-2000s Philips TVs. Basically, matching what's on the screen to a set of background lights to make the display pop!

All the pieces of the Adalight project — a microcontroller, addressable “smart” LEDs and a USB connection — are present today on the diminutive Circuit Playground board. Why not replicate this classic project on today's affordable hardware?

This small remake is no substitute for the gargantuan Adalight builds that some users have created...but for laptops or small all-in-one PCs, it can be a fun project for a board you may already be learning with. There's **no soldering** and **no extra power supply** this time around!

Here's what the original Adalight could do:

For small screens, and only a few minute's time, we can have a pint-sized version easily working, just some software on your PC and on the Circuit Playground board. If you love it, you can always upgrade to the mega-AdaLight version!

Download & Install

Three pieces of software must be installed to follow this project:

Arduino IDE

If you've used Circuit Playground before, the **Arduino** software is **already installed** on your system. If not, please read [Circuit Playground Lesson #0 \(http://adafru.it/rb4\)](http://adafru.it/rb4) for guidance on downloading and installing the software and getting your computer talking to Circuit Playground.

Before continuing, you should know how to upload a sketch (such as the example “Blink” sketch) to the Circuit Playground board. Read through the getting-started guides (such as the one linked above) if this is unfamiliar territory.

Processing IDE

Next, download the **Processing** IDE [from the Processing web site \(http://adafru.it/aMI\)](http://adafru.it/aMI). *Processing* is a programming language for Windows, Mac and Linux with audiovisual applications in mind. The first section of the [Getting Started page \(http://adafru.it/aMJ\)](http://adafru.it/aMJ) explains how to install the software.

The Arduino and Processing IDEs look VERY similar! If you encounter problems with the Adalight code, you may have loaded it into the wrong IDE.

Adalight ZIP

Finally, [visit the Adalight page on Github \(http://adafru.it/aMK\)](http://adafru.it/aMK) and download the ZIP file, or simply use this link to download directly:

[Download Adalight ZIP File](http://adafru.it/tDz)
<http://adafru.it/tDz>

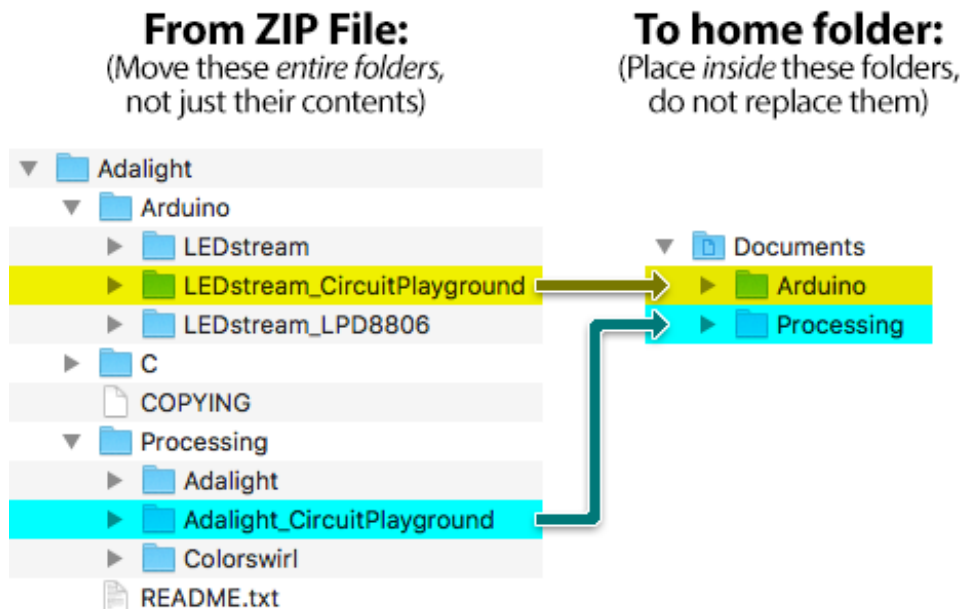
After uncompressing the ZIP file, you'll need to move some files into place.

If you've run the Arduino and/or Processing IDEs before, there will be corresponding “Arduino” and “Processing” folders inside your personal “Documents” folder (or “My

Documents” in Windows). In that case, move the contents of the Arduino and Processing folders from the Adalight ZIP file into the corresponding document folders.

If the Arduino and Processing folders don't yet exist on your system, you can just copy these from the Adalight ZIP file to your Documents folder.

The other files and folders in the ZIP file can be ignored. These are for other projects and aren't essential to its use.



Exit the Arduino and Processing IDEs if they're running. The newly-installed folders aren't visible until the next time these programs start.

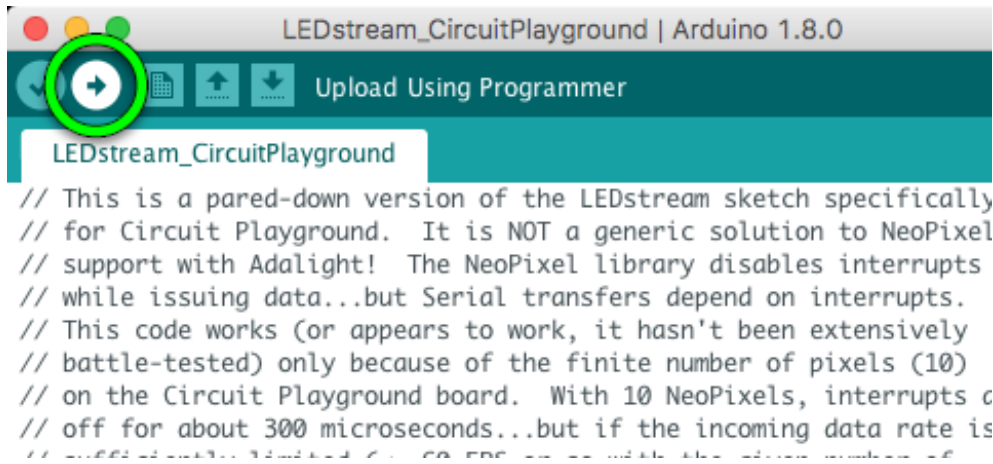
Upload Code to Circuit Playground

Launch the Arduino IDE. From the **File** menu, select **Sketchbook**, which should “roll over” to show **LEDstream_CircuitPlayground**. Select this.

From the **Tools** menu, select **Board**, then **Adafruit Circuit Playground**. If this option is not present in the Board menu, the Circuit Playground software is not correctly installed. Please review the previously-mentioned tutorials to get this started.

From the **Tools** menu again, select **Serial Port**, and then the port corresponding to your Circuit Playground board.

Click the **Upload** button near the top-left of the window:



If all goes well, you should see a **‘Done Uploading’** message in the Arduino window. You won’t see Circuit Playground doing anything yet, but it’s ready and waiting for other software on your computer to connect...

Run the Processing Sketch

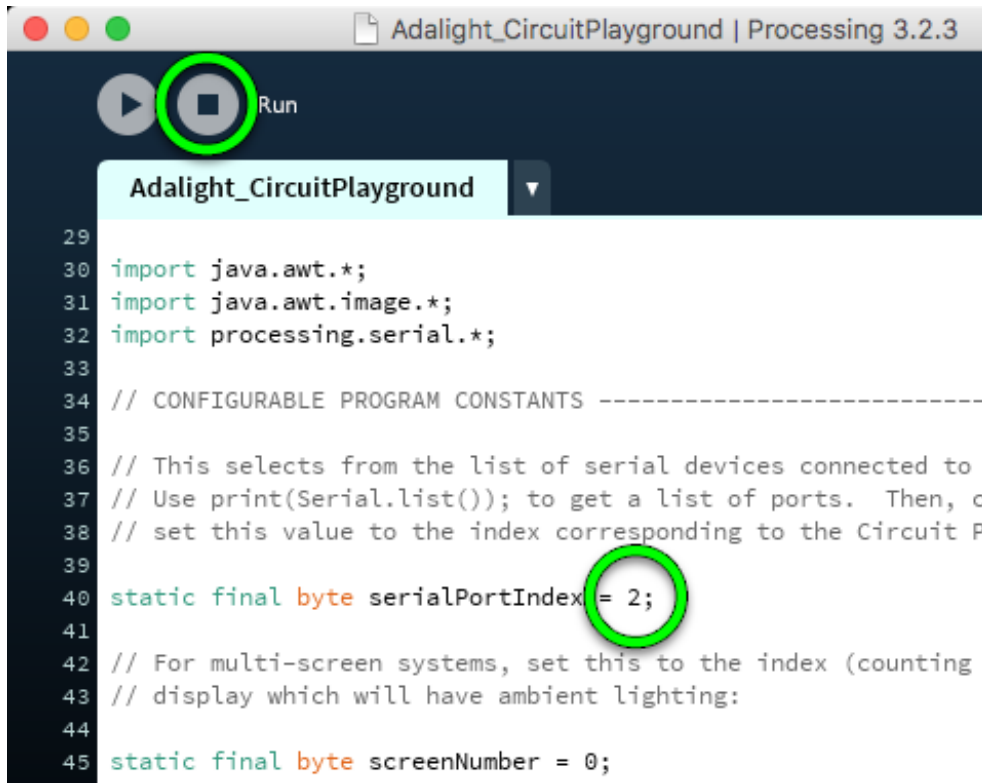
Launch the Processing IDE. After a moment, you should see a blue and white window that looks similar to the Arduino IDE, but darker.

From the **File** menu, select **Sketchbook**. This pops out a new window, which you should be able to scroll through and find the “Adalight_CircuitPlayground” sketch (if it’s not there, review the installation steps above), then open it.

You can *try* running it just as-is, but it likely won’t work...you’ll either get a small preview window (that’s a promising start), or it might just completely crash if there are no serial devices attached.

You need to edit the code first to point to the correct serial device number. When the program first starts, it prints out the names of serial devices, starting from **#0**. If you **stop the program very quickly**, you can **scroll up in the console** to see this list...**count up from zero** until you find the **serial port corresponding to the Circuit Playground device** (it’ll be the same serial port name you selected in the Arduino IDE). Then run again. If it’s not working, check your counting and try again.

This will be different on every system, so we can’t just tell you what to put there.

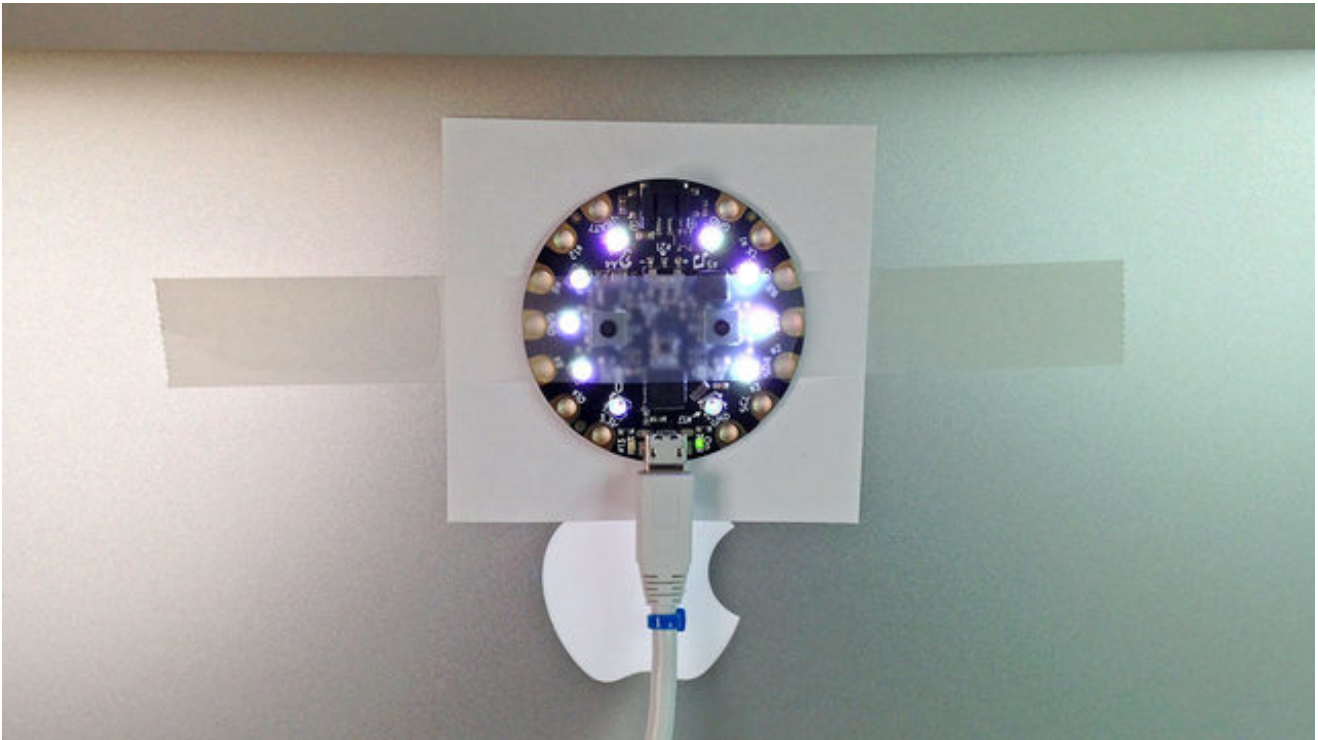


```
29
30 import java.awt.*;
31 import java.awt.image.*;
32 import processing.serial.*;
33
34 // CONFIGURABLE PROGRAM CONSTANTS -----
35
36 // This selects from the list of serial devices connected to i
37 // Use print(Serial.list()); to get a list of ports. Then, c
38 // set this value to the index corresponding to the Circuit P
39
40 static final byte serialPortIndex = 2;
41
42 // For multi-screen systems, set this to the index (counting
43 // display which will have ambient lighting:
44
45 static final byte screenNumber = 0;
```

When it works, you should see the NeoPixels on the Circuit Playground board light up.

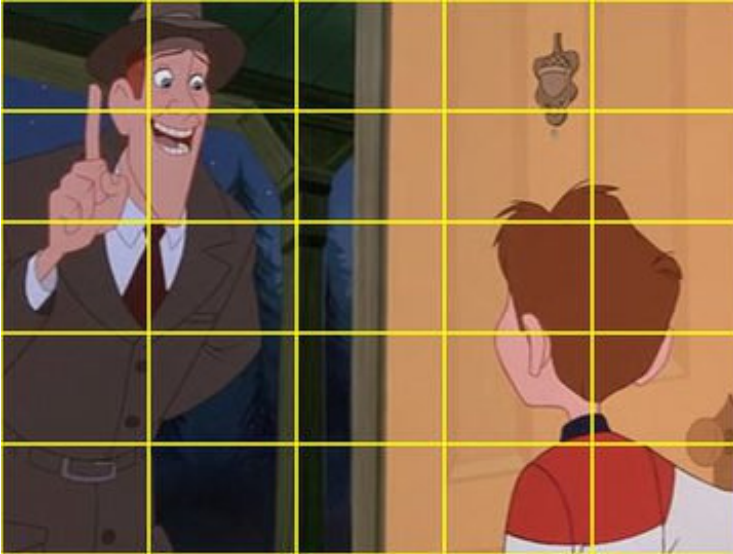
What It Does

With the **USB port at the bottom**, stick Circuit Playground to the back of your screen, facing the wall behind. Blue poster-tack or double-stick foam tape can work...or sometimes even regular clear sticky tape across the face of the board. But **DO NOT** stick Circuit Playground directly on the back of a **metal**-bodied device, like an aluminum MacBook, or you'll get an electrical short...cover the back of the board or use a mounting method that leaves a little space between board and screen.

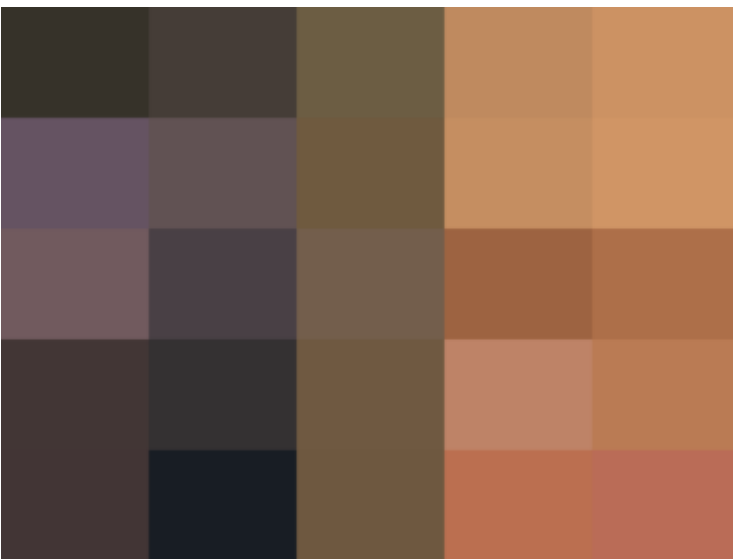


NEVER stick Circuit Playground directly on a metal computer. There should be an air gap, or tape an index card between the screen and board.

The Processing sketch clones the contents of your screen and sends this data to the Circuit Playground, which then updates the LEDs. Try moving windows around, or play a movie or a game.



The Processing code captures screen shots just as fast as it can. It then divides this image into a 5x5 grid, and takes the averages of each of those squares...



Ten of those squares correspond to the ten NeoPixels on the Circuit Playground board. It's not all one color...you can actually see motion reflected in the LEDs.



The Processing code doesn't work with all graphics cards and modes...full-screen

video, for example, may be accessing the graphics hardware directly and bypassing the normal frame buffer that Processing works with. (In one case, it works fine with full-screen movies on my NVIDIA-based iMac, but not my Intel Graphics-based MacBook Air...in the latter case, I have to keep the movie in a windowed mode, with the menu bar visible. Ah well.)

There are compatible third-party apps that may work better with full-screen video. [The original Adalight guide \(http://adafru.it/tDA\)](http://adafru.it/tDA) mentions a couple of options (Lightpack and Boblight) that may fare better in this regard, though they'll require some initial set-up to define the pixel placement as described above.