



Paper-Craft Crystal Gem Lantern

Created by Erin St Blaine



<https://learn.adafruit.com/paper-craft-crystal-gem-lantern>

Last updated on 2021-11-15 07:39:33 PM EST

Table of Contents

Overview	3
• Large Crystals	3
• Smaller Crystals	5
Wiring Diagram	6
Code with MakeCode	6
Make the Gem	14
Electronics Assembly	18
Customize Your Crystal	21

Overview

Crystal Gems will always save the day!

Create a stunning paper craft crystal from laminated cellophane wrap. Add NeoPixels and watch it sparkle!

We've included two different build sizes in this tutorial -- one crystal large enough to house a Circuit Playground Express, and one with a single NeoPixel illuminating the crystal. This project can be run from a battery for cosplay applications (the crystal on Gandalf's staff perhaps?) or plugged in to the wall via USB for a more permanent installation.

We've also included a Steven Universe Warp-Pad 3D printable base, so your Crystal Gems can be displayed in style. The possibilities are endless!



Large Crystals

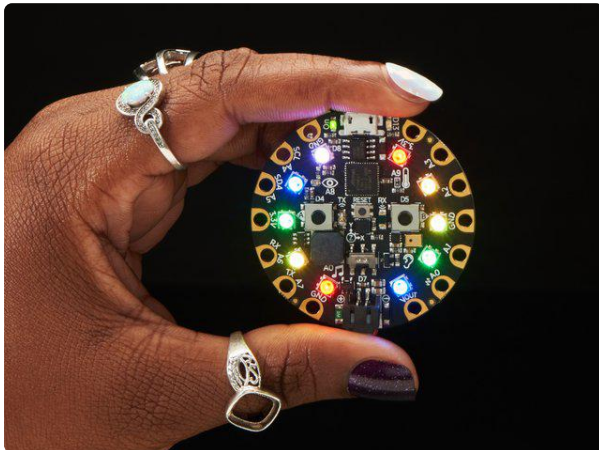
This is a simpler build that requires no soldering -- the Circuit Playground fits inside the gem and provides all the light.

Materials Needed

- Circuit Playground Express
- Iridescent Cellophane gift wrap (from the craft store)

- Laminating machine & two 3mm pouches
- Utility knife or Vinyl Cutting Machine & accessories
- Super glue
- 3D printer for the base
- Battery or USB cable for power

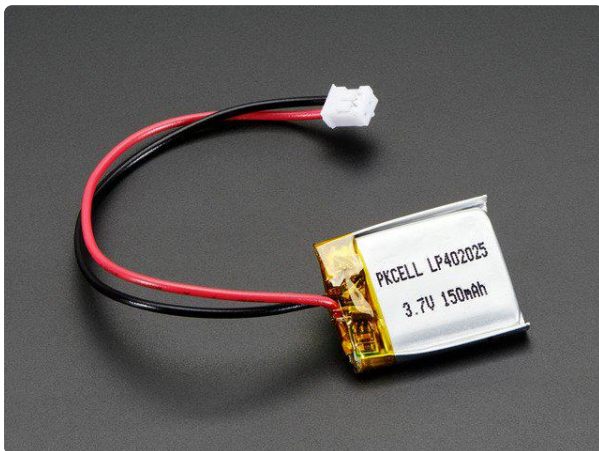
If you don't have a cutting machine or a laminating machine, you can [buy a pre-cut crystal gem kit here \(https://adafru.it/OMF\)](https://adafru.it/OMF).



[Circuit Playground Express](https://www.adafruit.com/product/3333)

Circuit Playground Express is the next step towards a perfect introduction to electronics and programming. We've taken the original Circuit Playground Classic and...

<https://www.adafruit.com/product/3333>



[Lithium Ion Polymer Battery - 3.7v 150mAh](https://www.adafruit.com/product/1317)

Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...

<https://www.adafruit.com/product/1317>



[USB cable - USB A to Micro-B](https://www.adafruit.com/product/592)

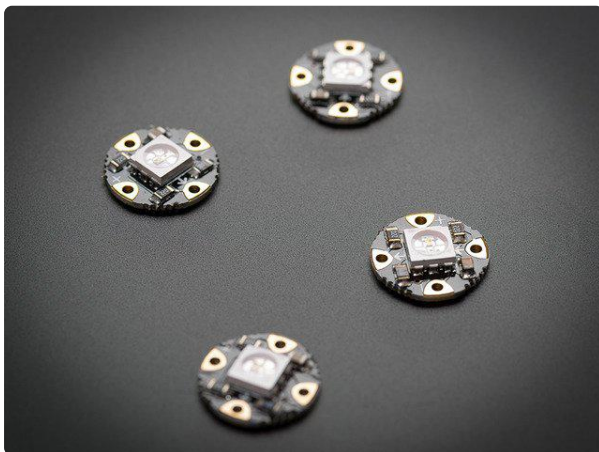
This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...

<https://www.adafruit.com/product/592>

Smaller Crystals

This build is a little more complicated since you'll be soldering individual NeoPixels together to illuminate the crystal. You can make multiple crystals and string them together, and the NeoPixels will animate in concert with each other. It's fun to make multiple crystals in slightly different sizes to create a gorgeous sparkling art piece.

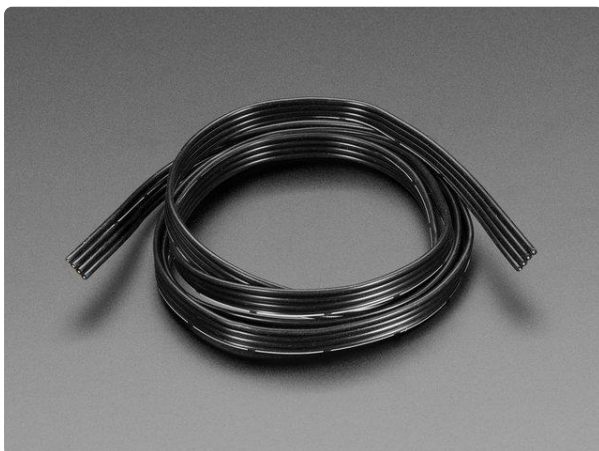
- Circuit Playground Express
- Individual NeoPixels (1 for each crystal)
- Soldering Iron & Accessories
- Iridescent Cellophane Gift Wrap (from the craft store)
- Laminating machine & 3mm pouches
- Wire or ribbon cable with 4 strands
- Utility knife or a Vinyl Cutting Machine & accessories
- Super glue
- Battery or USB cable for power



[Flora RGB Smart NeoPixel version 2 - Pack of 4](https://www.adafruit.com/product/1260)

What's a wearable project without LEDs? Our favorite part of the Flora platform is these tiny smart pixels. Designed specifically for wearables, these updated Flora NeoPixels have...

<https://www.adafruit.com/product/1260>



[Silicone Cover Stranded-Core Ribbon Cable - 4 Wires 1 Meter Long](https://www.adafruit.com/product/3892)

For those who are fans of our silicone-covered wires, but are always looking to up their wiring game. We now have Silicone Cover Ribbon cables! These may look...

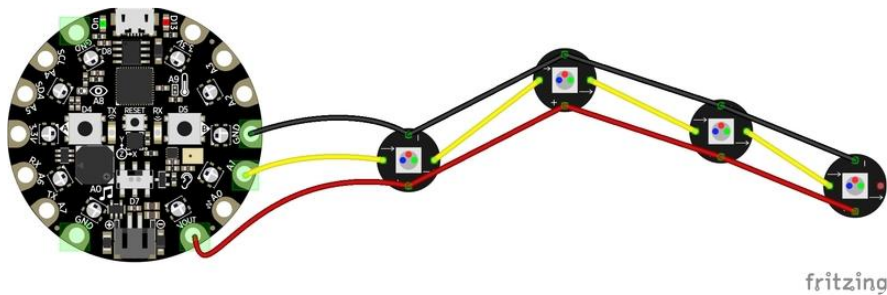
<https://www.adafruit.com/product/3892>

Wiring Diagram

Large Crystals

The larger crystal is big enough to fit the Circuit Playground Express and a small battery inside, making wiring very simple. Just plug in the battery or USB cable and you're done!

Smaller Crystals



For the smaller crystals, it works best to put a single NeoPixel inside, since the Circuit Playground won't fit. You can also string multiple crystals together to make a hanging lamp or glowing crystal cluster. These are wired as follows:

- VOUT to +
- A1 to IN
- GND to -

If you have more than one crystal, wire each OUT pin to IN on the next NeoPixel in the sequence.

Code with MakeCode

Head to www.makecode.com (<https://adafru.it/Dyi>) and select the Circuit Playground Express. Create a new project and give it a name. I called mine Crystal Fire Gem.

Here are the completed projects, if you just want to download the code and go. The "Large Crystal Fire Gem" version uses the lights onboard the face of the Circuit Playground Express. The "Small Crystal Fire Gem(s)" version assumes you've soldered NeoPixels onto pin A1 and runs the lights on them.

The step-by-step below shows the Large Gem / Circuit Playground Onboard lights code. At the end, we'll show you how to modify it for a NeoPixel strand instead.

Large Crystal Fire Gem MakeCode

<https://adafru.it/EJr>

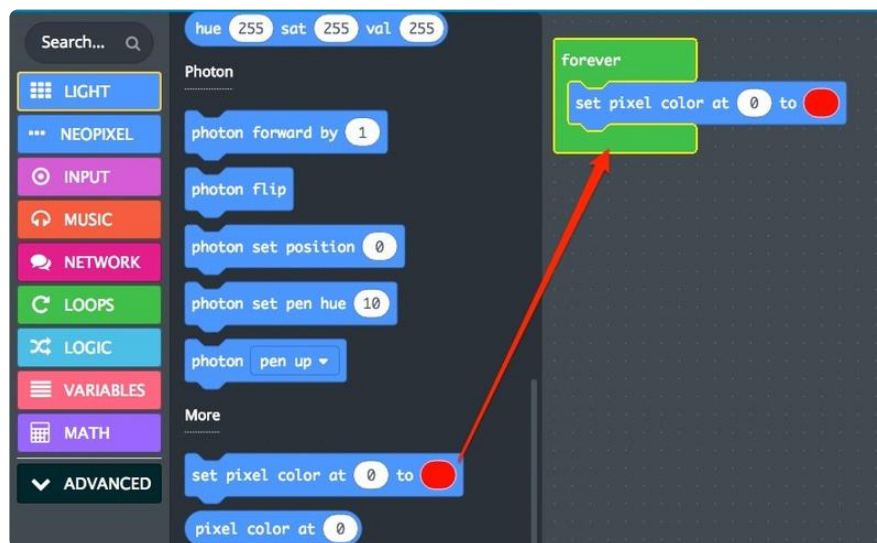
Small Crystal Fire Gem(s) MakeCode

<https://adafru.it/EM6>

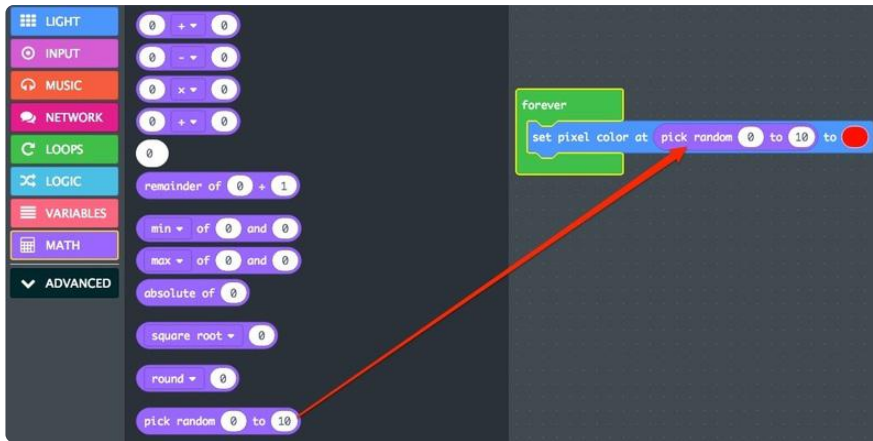
Fire Light Flicker Animation

For this lamp, I wanted a flickering fire effect. To achieve this, we'll add a bit of randomness to the hue and value fields, as well as to which pixel we're affecting. The more randomness we can add, the more natural the flame effect will seem.

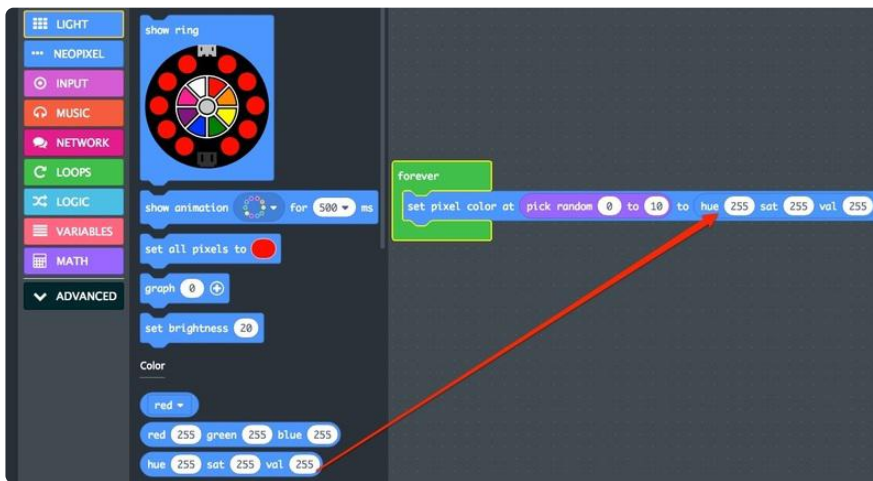
From LIGHT drag an instance of `set pixel color at 0 to red` into your `forever` loop.



From the MATH tab, drag an instance of `pick random 0 to 10` to replace the `0`. Now our lights will be randomly assigned to the 10 lights on the Circuit Playground.



From the LIGHT tab, find the `hue 255 sat 255 val 255` block. Use it to replace the `red` in the block you just made.



This little bit of awesomeness will give us lots of control over the colors. Hue refers to what we usually think of as a color: red, orange, green, etc. Red is at 0, yellow is around 50, blue 160 and it cycles back around to red at 255.

Saturation (sat) controls how intense the color is. For example, given a hue of 0 (red), saturation=0 comes out plain white, 100 is sort of pinkish, and 255 is full, bright red.

Value (val) refers to the amount of color. Think of it as intensity or brightness.

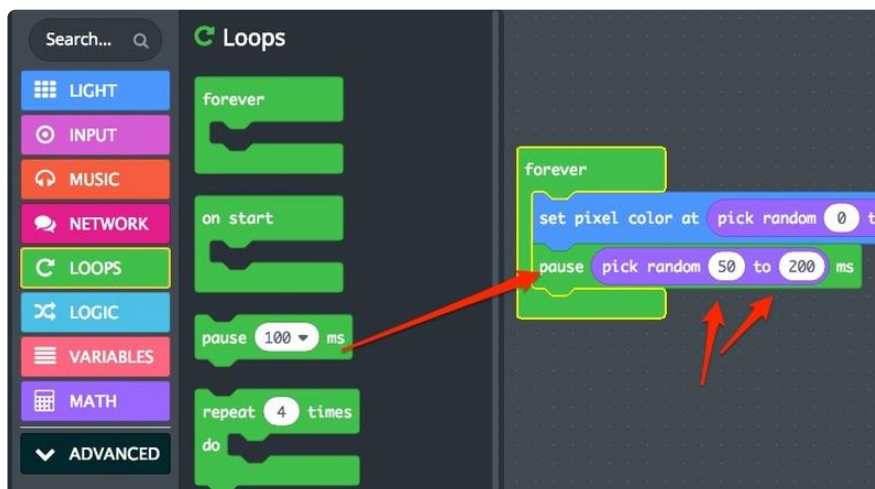
There's lots of info about HSV colors available on [Wikipedia \(https://adafru.it/CxV\)](https://adafru.it/CxV) if you want to know more.



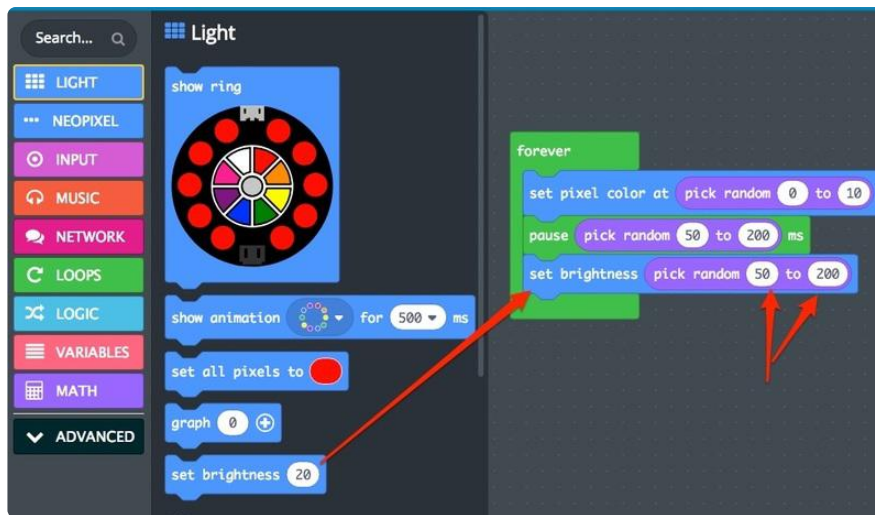
Grab two more instances of `pick random` from the MATH tab and drag them in to replace the `hue` and `val` settings. For an orangey-red flame, change the hue to 5-20. Experiment with different numbers here to find the color you like.

The `val` setting translates to brightness (more or less). I chose a range from 100-255.

Next we'll add some speed randomness and control to the flickering. Drag an instance of LOOPS > `pause 100 ms` into your forever loops. Then add a randomness block and try values of 50 and 200. Take a look at the emulator on the left side to see how you like it.



Mine wasn't quite flickery enough, so I added another block that adds more randomness to the overall brightness. Drag an instance of LIGHT > `set brightness` into your `forever` loop and try some different random values. I liked 50-200.



Download your code to your Circuit Playground Express and see how you like it. Does it look flamey enough? Play with the values in all the different blocks you made until you absolutely love the effect.

On/Off Switch

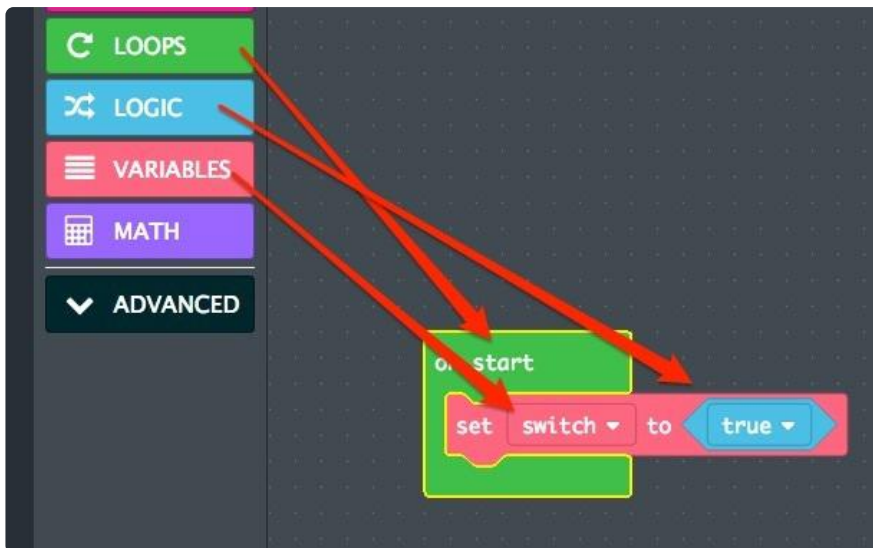
It's not too hard to add some code that will turn the lights on and off with any number of triggers. The Circuit Playground Express comes ready-made for this type of thing, with onboard capacitive touch pads and sensors already set up. All you need to do is code the switch and decide how you want to trigger it.

In this example I'm triggering the switch with the capacitive touch pad A4 on the Circuit Playground. You can also use the sound or acceleration triggers just by changing the INPUT loop you choose.

From your LOOPS tab, drag an instance of `on start` into your workspace. Anything in this loop will happen just once, when your Circuit Playground Express boots up.

Next, from VARIABLES, create a new variable called `switch`. Drag `set switch to 0` into your `on start` loop.

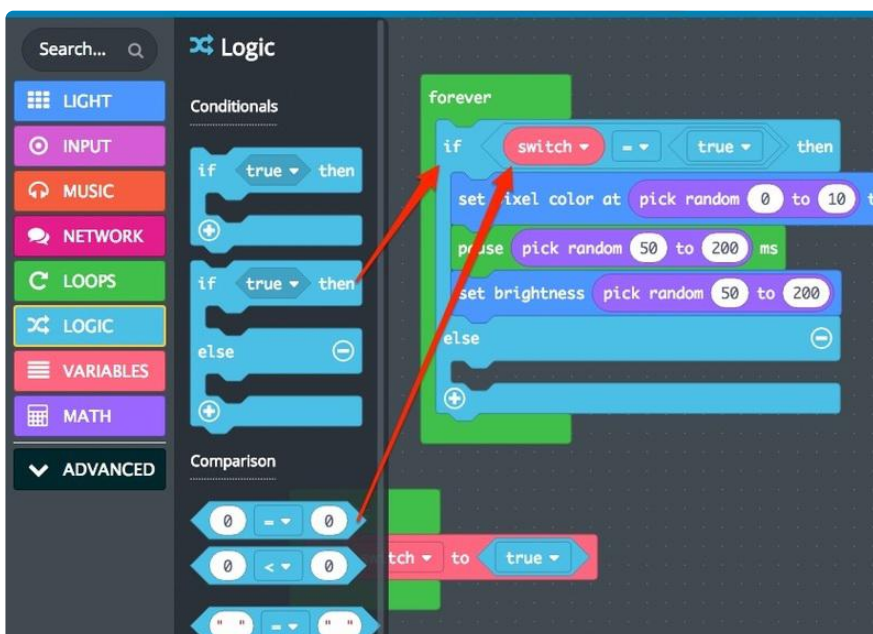
Then, from LOGIC, replace the `0` with `true`. This will allow us to tell the switch to turn the lights on when true, or off when false.



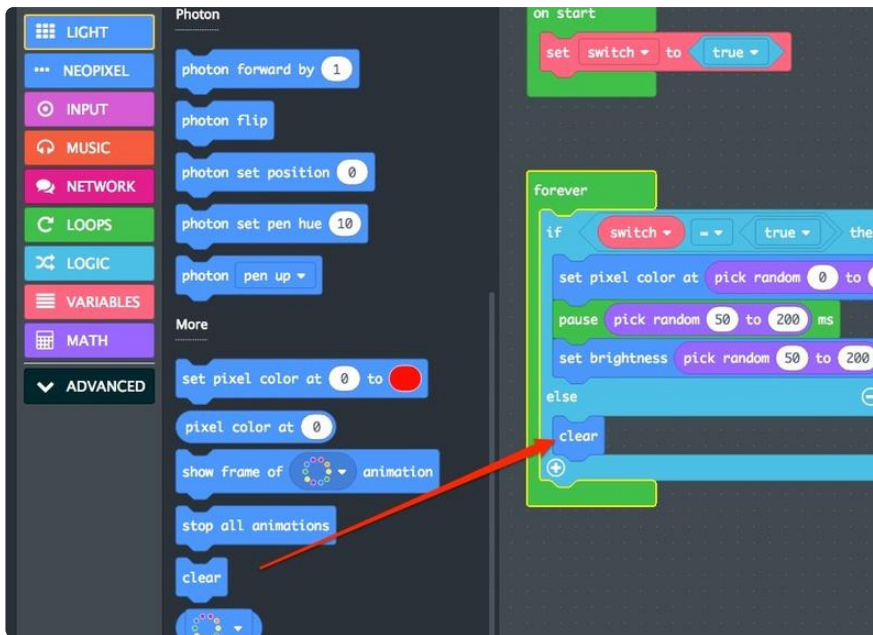
Now let's add a conditional loop to our `forever` loop. From LOGIC, drag an instance of `if / then` into your `forever` loop, and put the existing code blocks inside the first if slot.

From LOGIC, replace the `true` with a comparison block, and make it read `if switch = true then`

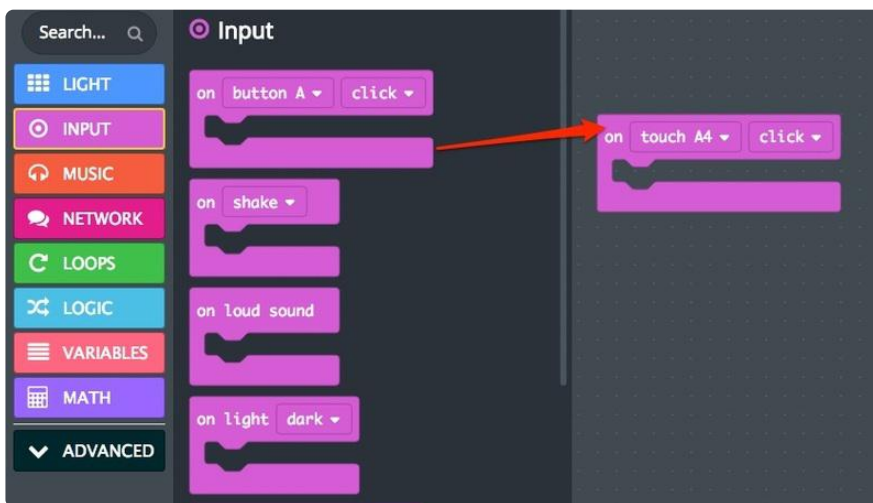
This will make the flame code you wrote earlier happen when the variable `switch` is set to `true`.



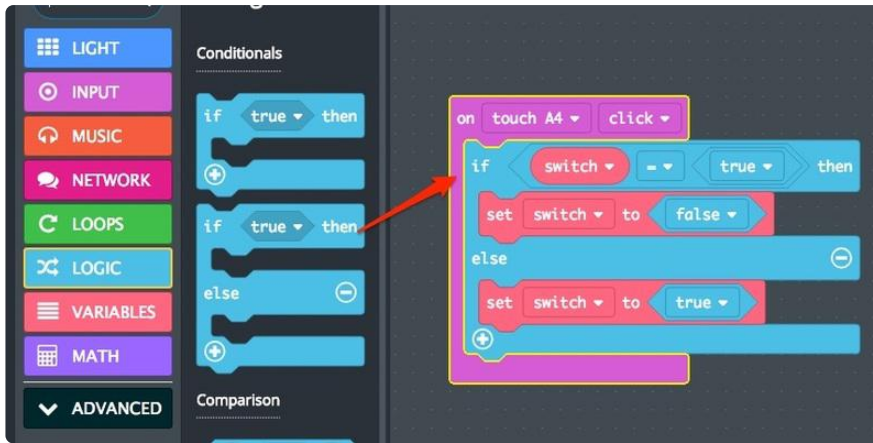
In the else block, add LIGHT > `clear` to turn all the lights off.



Finally, we'll tell the switch to change states when we touch pad A4 on the Circuit Playground. From INPUT, drag an instance of **on button A click** and change it to read **on touch A4 click**.



Add another conditional from the LOGIC tab as shown. This will create a toggle switch, turning the lights on and off each time the pad is touched.



Download your code to your Circuit Playground and test it out. Once everything's working, it's time to build the crystal gem.

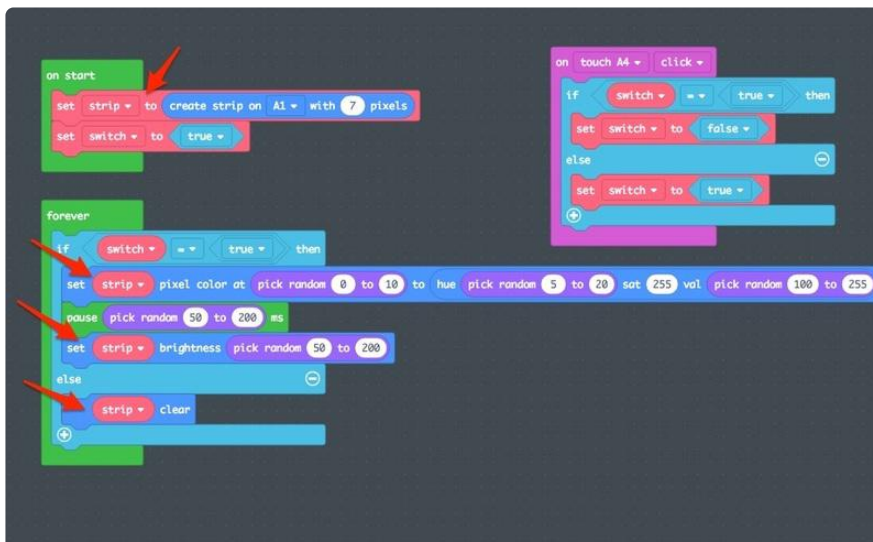
Troubleshooting

If you're having trouble getting the code to download, head over to the [Circuit Playground Express Intro Guide \(https://adafru.it/BeF\)](https://adafru.it/BeF) for help.

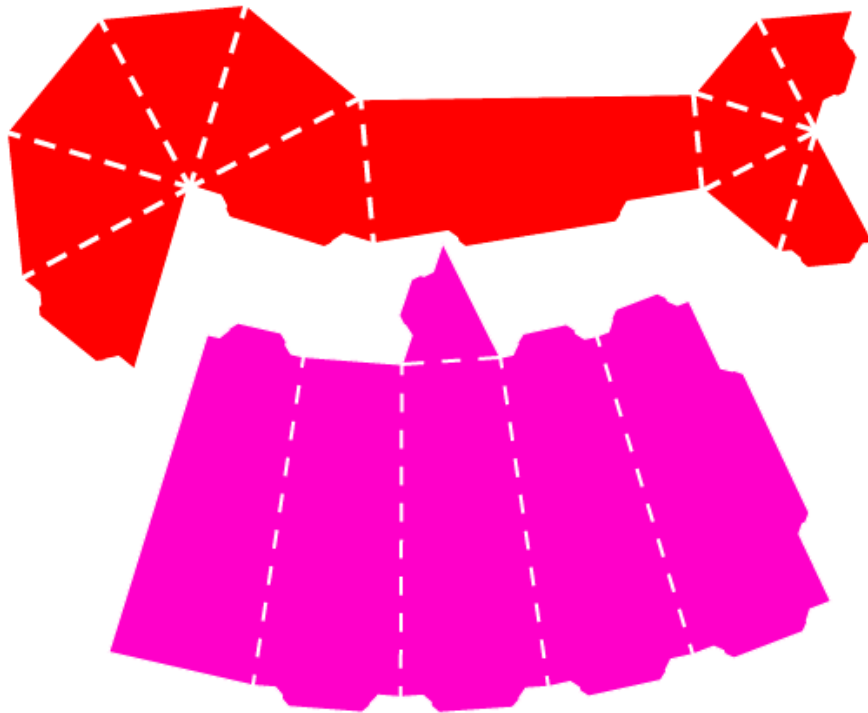
Small Gem Version

To modify the code to run on individual NeoPixels, make the following changes:

- Add LIGHT > NEOPIXEL > `set strip to create strip on A1 with (x) pixels` to your `on start` loop
- Change `set pixel color...` to `set strip pixel color...` (NEOPIXEL tab) for both instances in the `forever` loop
- Change `clear` to `strip clear` in the `else` portion of the conditional loop



Make the Gem



Download the printable template below for a large or small Crystal Gem. [Or, you can order a pre-cut gem kit here \(https://adafru.it/OMF\).](#)

paper_gem_large.pdf

<https://adafru.it/EM3>

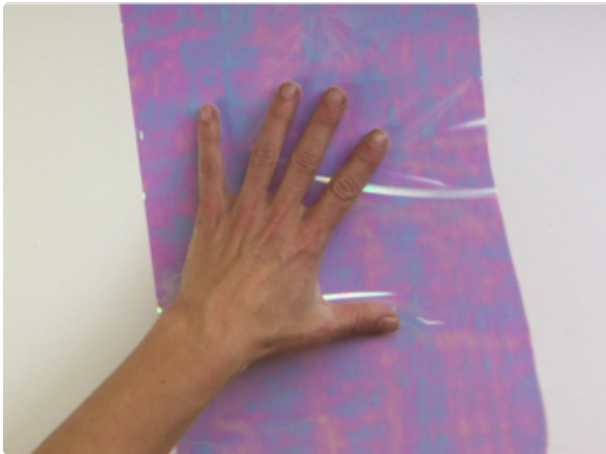
paper_gem_small.pdf

<https://adafru.it/EM4>

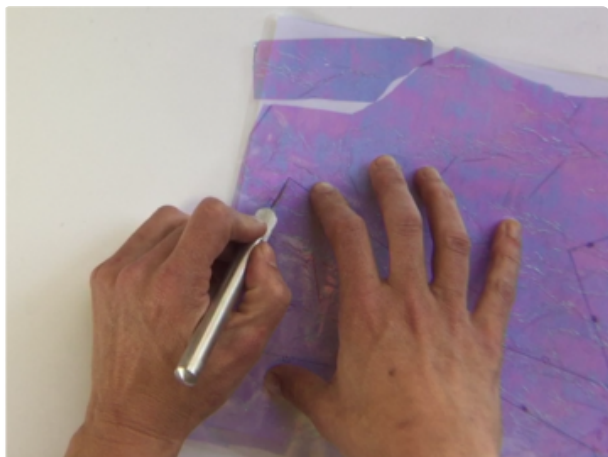


The crystals are made from laminated iridescent cellophane wrap. I'm using letter-sized 3mil thick laminate pouches from Scotch.

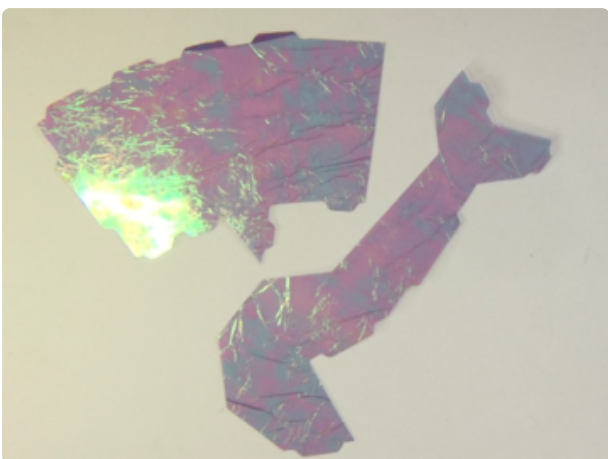
Cut a piece of cellophane to fit neatly inside the pouch. Crumple it up for more texture, and trim to be sure it's not hanging out the sides. Keep it to one layer -- multiple layers will separate after laminating.



Heat up your laminating machine and run the cellophane pouch through. It will emerge all warm and crinkly, and catch the light in a most delicious way.



Print out the paper template and place it underneath the laminated cellophane. Cut on all the solid lines and score on the dotted score lines.



Older Makers should help younger Makers with the lamination (HOT!), the cutting (SHARP!), and the Krazy glue (VERY STICKY!)

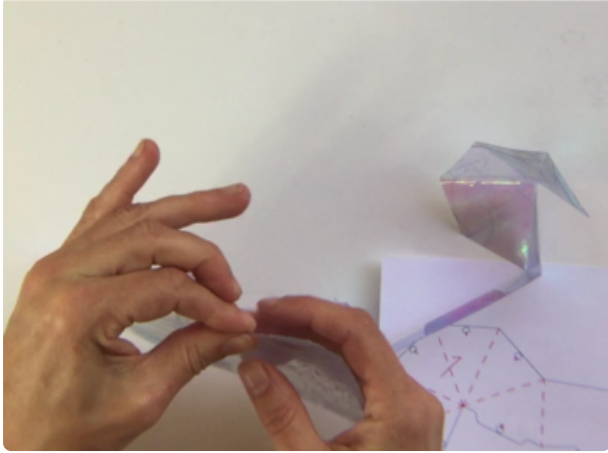
If you have a vinyl cutting machine, you can use it to cut this material too. Here is an .stl file for uploading to your favorite vinyl cutting service.

If you're using a Cricut, the best setting is Custom > Stencil Film 0.4mm

The file has two pieces, so if you scale the size up or down be sure to scale them both at the same time so they still fit together.

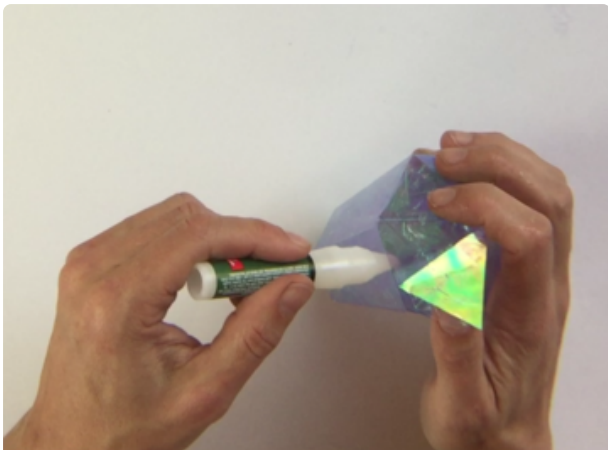
paper_gem_2part.svg

<https://adafru.it/EP5>



Fold in the same direction along all the score lines. Be sure you're folding both halves of the gem in the same direction or they won't fit together.

Gently fold down all the tabs on the edges. These don't need a sharp crease, just a bit of a bend.



Use Krazy glue on the tabs to assemble the gem. It's helpful to start by attaching the two different pattern pieces together first, then glue one tab at a time to create the gem.

Line up the corners carefully. The more accurate you can be, the better your gem will look. Also, crazy glue will set a bit faster and more easily if it's slightly above room temperature.

The antidote to crazy glue is acetone (nail polish remover). When your fingers get stuck together, and you can no longer unlock your iPhone with your fingerprint, just dip your fingertips in nail polish remover and the crazy glue will peel right off.

Don't glue the gem closed completely -- leave a flap open so you can access the inside.



Cut and crumple a small piece of un-laminated cellophane. This will act as a diffuser inside the crystal, and give it that magical look, as though the crystal has lots of beautiful occlusions inside. Use less cellophane here than you think. A little goes a long way, and if you stuff the crystal too full the light won't bounce around as nicely.

Set this aside for now. We'll add it as the final touch after adding the lights.

Electronics Assembly

The .stl file for a 3D printed Steven Universe Warp Pad base for the crystal is below. Print with supports!

For the larger crystal, the Circuit Playground Express simply goes inside the crystal. For the smaller one, you can hide the Circuit Playground and battery inside the bottom of the base. Hot glue works great to secure the crystal firmly in place.

Warp Pad - Large Hole .stl

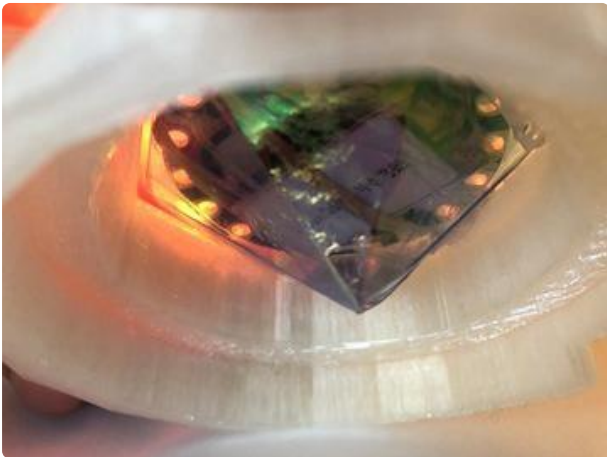
<https://adafru.it/EM7>

Warp Pad - Small Hole .stl

<https://adafru.it/EM8>



Large Crystal



Plug a battery or USB cable into your Circuit Playground Express. I like to add a rubber band to my battery to help it stay hidden.

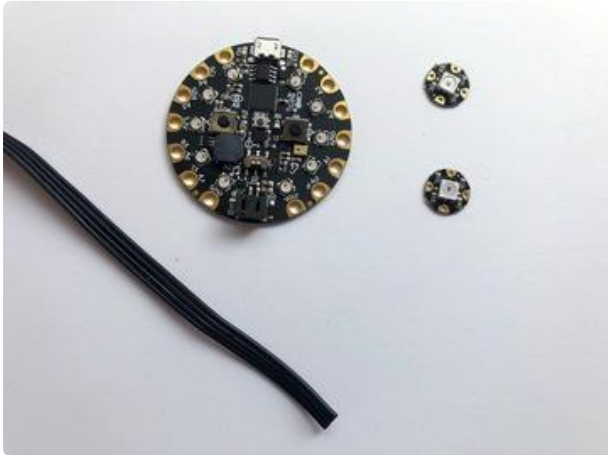
Slip the board inside the crystal and nestle it along the creased edge, with the battery underneath in the bottom of the crystal or the USB cord threading out and down through the flap you left open.



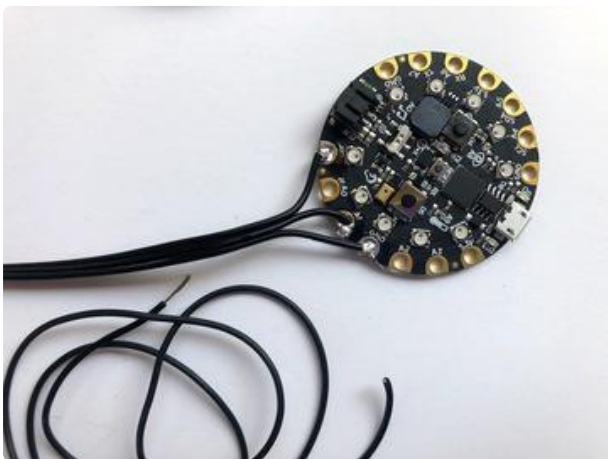
Add the crumpled cellophane to diffuse the light.

Small Crystal

This one's a bit more complicated since we need to do some wiring and soldering. The small crystal has just one NeoPixel inside, but the magic here is that you can have a series of crystals all animating together, which is definitely worth a little extra work.



Start by finding the striped wire on the ribbon cable. This wire will be your power (VOUT) wire, the next wire will be data, and the third wire will become your ground wire. We don't need the fourth wire, so pull it apart from the others and save for another project.



Solder the three remaining wires to your Circuit Playground:

- Striped wire to VOUT
- Middle wire to A1
- Third wire to G

Note: Don't use A0 instead of A1 -- A0 is the speaker pin, and does not like to run NeoPixels.



Measure out the length of wire you want. Cut it to length. If you're using just one NeoPixel, solder the striped wire to +, the middle wire to IN, and the third wire to G.



If you're using more than one NeoPixel, you'll want to solder two wires into the + and - pads, so you can pass power to the next pixel on down the line. Solder a wire from OUT on your first pixel to IN on the next pixel in the series, and so on.



Slip each pixel inside the end of your crystal and secure it in place with hot glue. Add your crumpled up cellophane diffuser, then secure the crystal's open flap with crazy glue.

Customize Your Crystal

If you want to design your own crystal, here's a video showing how I designed mine in Fusion360, then exported using the Slicer plugin to create the flattened gem pattern.

This design uses parametric scaling, so it's easy to change the size or proportions of the crystal gem, so it's exactly to your liking.