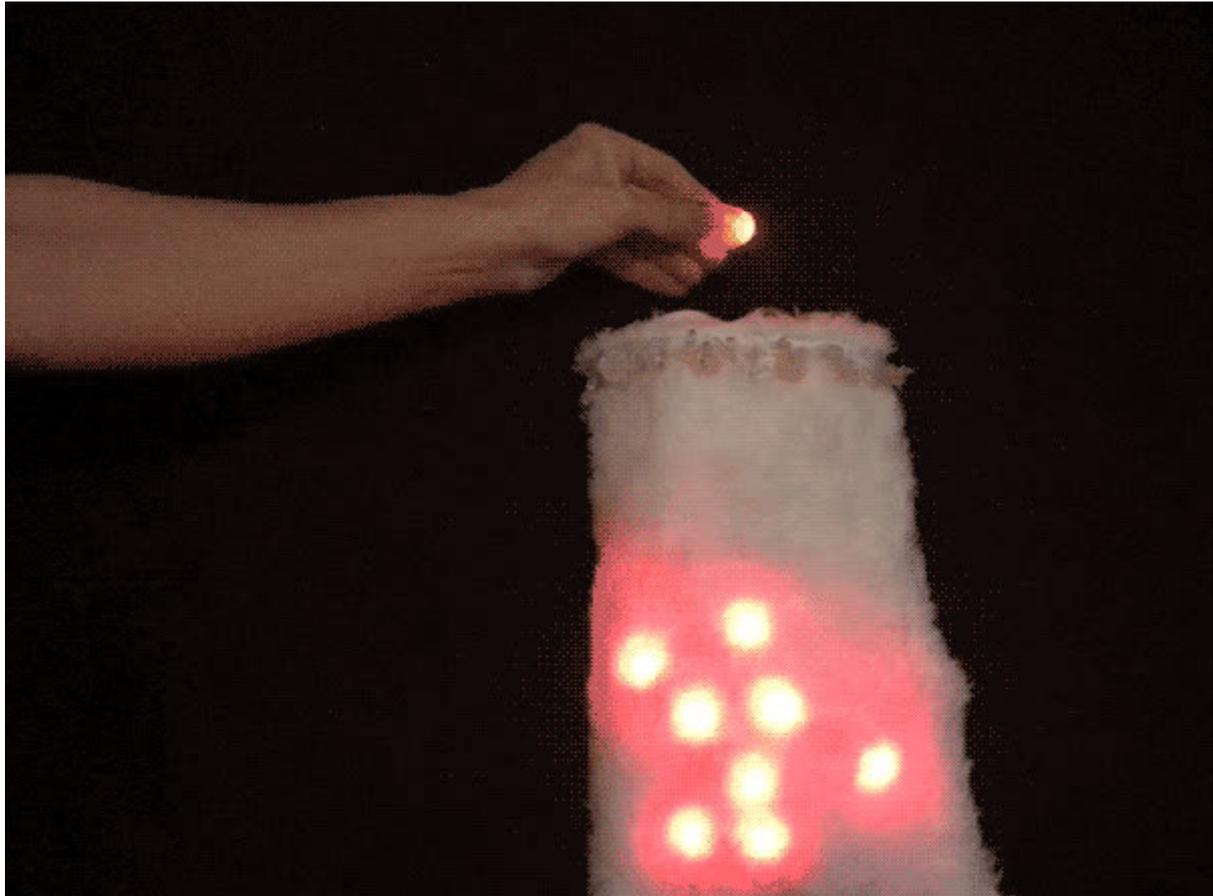




Magic Light Bag of Holding

Created by Erin St Blaine



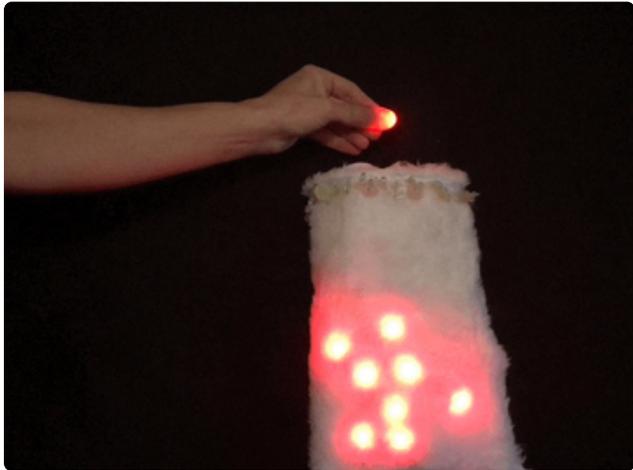
<https://learn.adafruit.com/magic-light-bag-of-holding>

Last updated on 2024-06-03 02:32:14 PM EDT

Table of Contents

Overview	3
• Parts	
Wiring Diagram	5
Assembly	6
Code with MakeCode	11
• Filling the Bag with Lights	
• Emptying the Bag	
• Resetting the Trick	
Use It	21
• More Ideas	

Overview

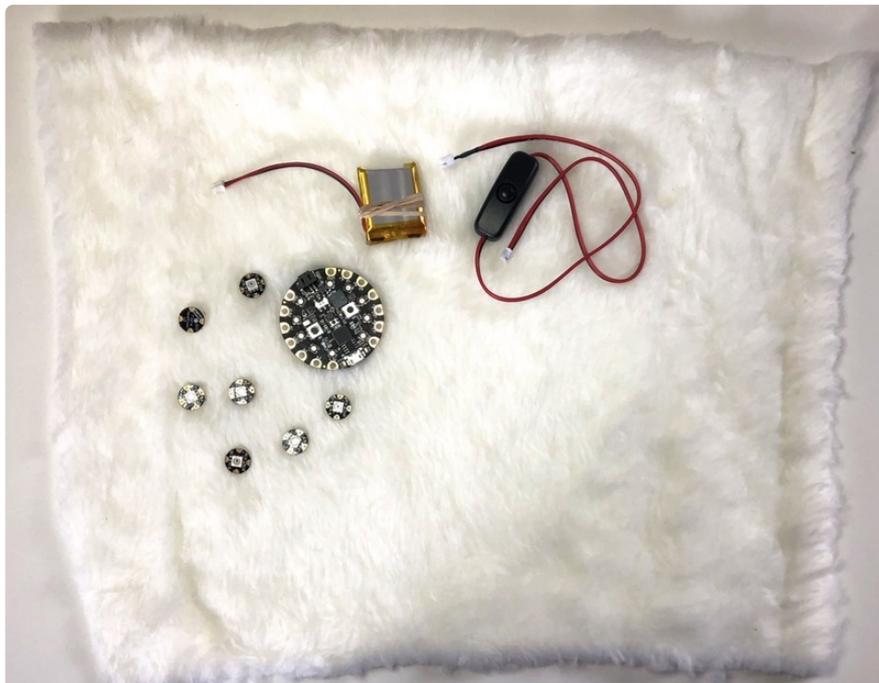


Create a bag that magically collects and holds lights! Pull up to ten colored lights from nowhere, and drop each one into your bag and watch them light up. Turn the bag upside down and all the lights will spill out and vanish.

Delight and amaze your friends, UPS guy, kids, or pets with this simple trick. It's really easy to customize in MakeCode, with endless possibilities for colors, triggers, and magic effects.

First we'll lay out our NeoPixels and solder them together. Then we'll sew them into a simple fabric bag.

We'll program the Circuit Playground Express using MakeCode, Microsoft's drag-and-drop code editor. It's easy to customize your magic effects with all the different sensors and options the Circuit Playground has to offer.



Parts

Circuit Playground Express Microcontroller -- the brains!

1 x Circuit Playground Express

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

Circuit Playground Express Microcontroller -- the brains!

2 x NeoPixels

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

Individual NeoPixels in a pack of 4

1 x NeoPixel Strand

<https://www.adafruit.com/product/3630>

Could be used instead if you don't like soldering

1 x Battery

<https://www.adafruit.com/product/1578>

LiPoly Battery - 500mAh

1 x Battery Cable with Switch

<https://www.adafruit.com/product/3064>

Extension cable with on/off switch

1 x Battery Charger

<https://www.adafruit.com/product/1304>

USB Battery Charger

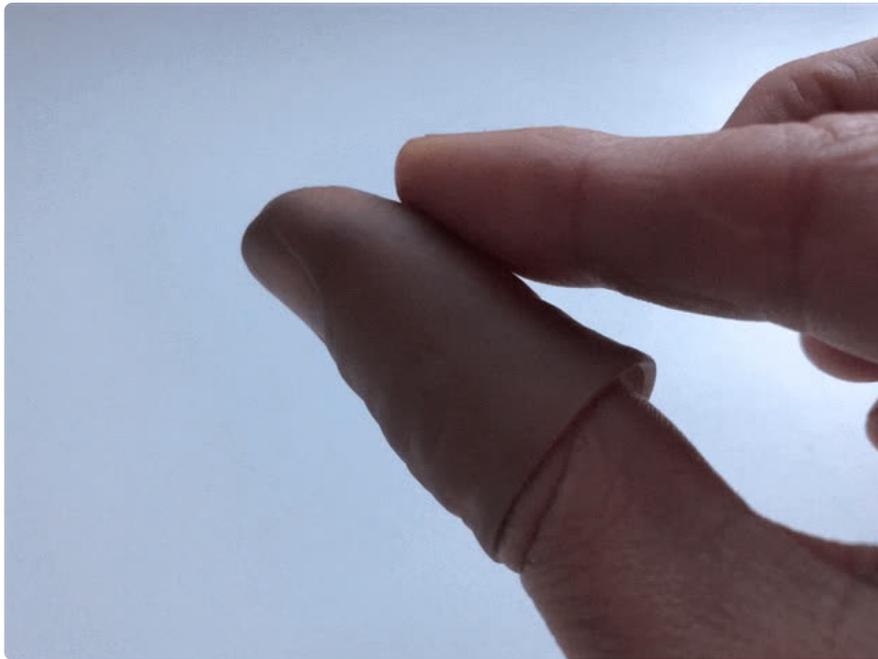
Other Materials Needed

- [Fun fur \(https://adafru.it/MD5\)](https://adafru.it/MD5) or other pretty fabric, about 16" x 14"
- Felt or fleece lining, same size plus a bit more
- Pretty trims or ribbons or jewels for decoration
- [E6000 glue \(https://adafru.it/MD6\)](https://adafru.it/MD6)
- Needle & thread and/or sewing machine
- Rubber band for the battery

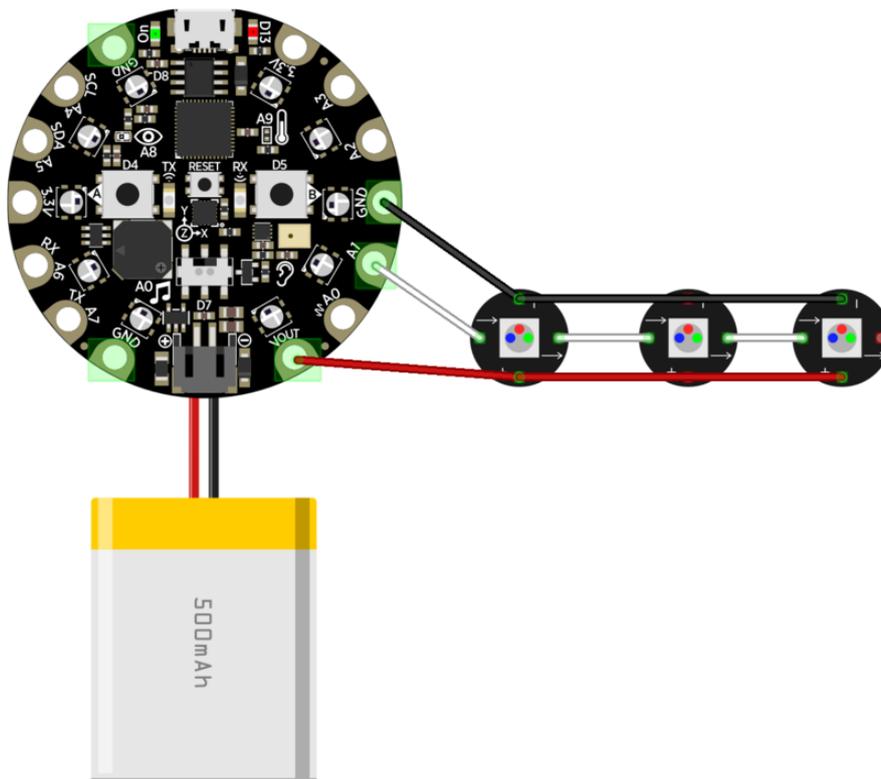
To do this trick the way I'm showing it, you'll also need a magic light-up thumb tip.

These are sold under the brand name [D-lite \(https://adafru.it/CV1\)](https://adafru.it/CV1) (among others) and come in a variety of colors and a small or large size.

These are so much fun! The way they work is that the light comes on when you squeeze your finger against your thumb. Simple, delightful, and endless possibilities for pranks and magic trickery.



Wiring Diagram



fritzing

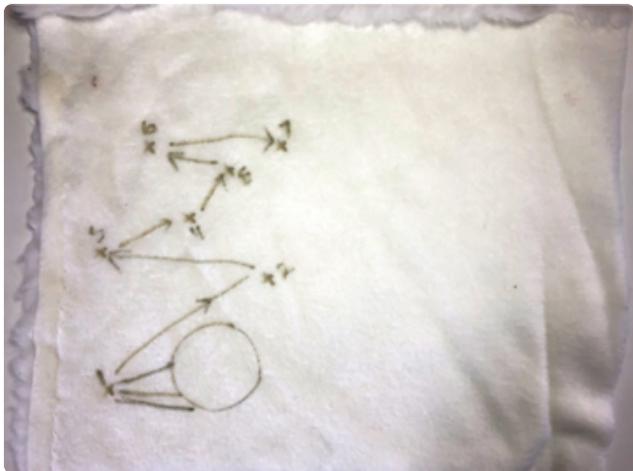
The individual NeoPixels will be wired up to pin A1 (to the data IN pad), VOUT to + and G to -.

Each NeoPixel's OUT pad will be wired to the IN pad on the next pixel.

If you want to add an on/off switch, you can plug the battery into a [switched battery extension cable](http://adafru.it/3064) (<http://adafru.it/3064>). These are really handy since they're so clicky and tactile that you can turn your project on or off purely by feel and squeezing through the bag.

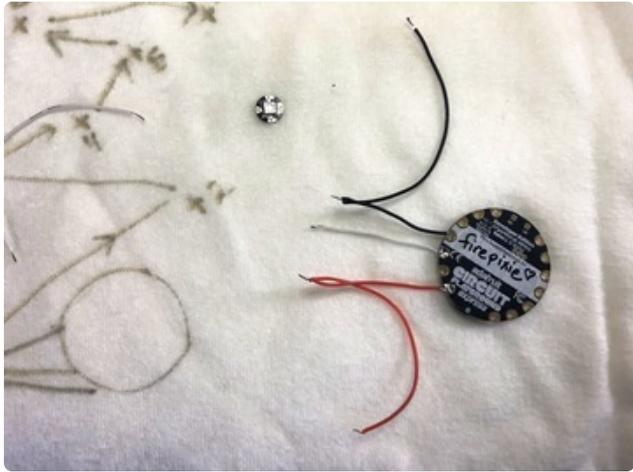
Assembly

This shows how to wire up the bag using individual NeoPixels. You could also use the [NeoPixel Light Strand](http://adafru.it/3630) (<http://adafru.it/3630>) -- these pixels are already soldered together and could be wired up to the Circuit Playground quickly and easily. It doesn't give you quite as much control over spacing, and your bag will feel a little stiffer, but it's a great option if you don't want to do a lot of soldering.



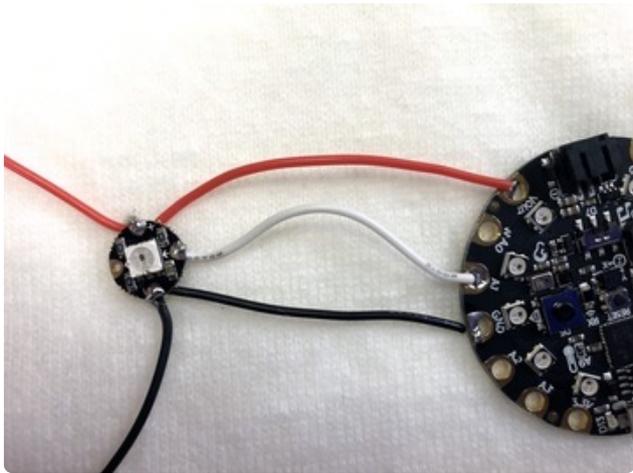
Cut your fabric to the size you want your bag. Mine is about 14" by 16", making a 14" x 8" bag when it's folded over and sewn.

Lay out the pixels and the Circuit Playground on the back of your fabric on the side that will become the front of the bag. Remember to place the Circuit Playground face down, so the pixels shine outwards. Mark where you want each pixel to go.

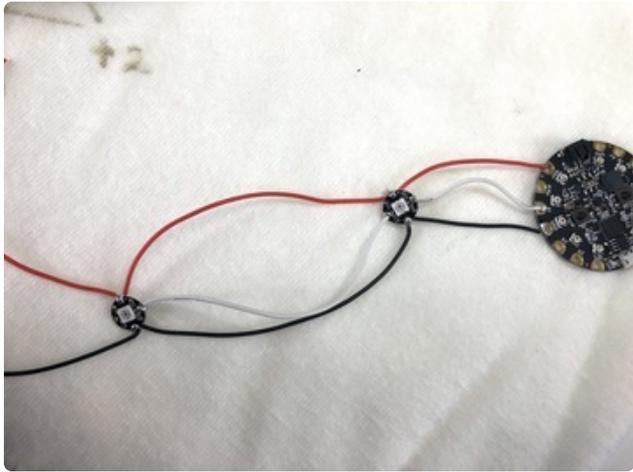


Now we'll wire up the pixels. Measure out the white, red, and black wire going to pixel #1 against your design and solder them to the A1, VOUT and G pads as shown in the wiring diagram.

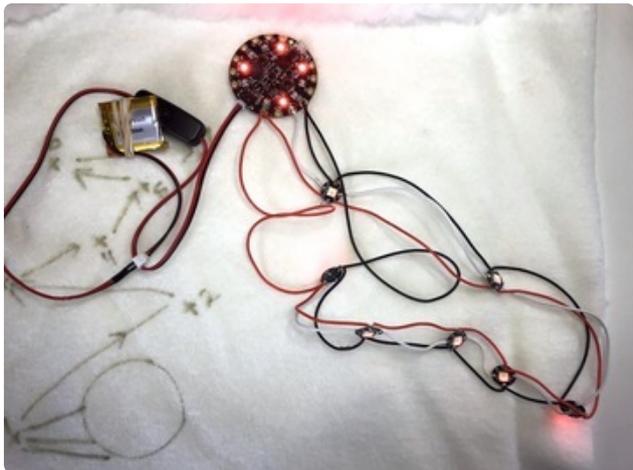
Measure out the red and black wires going to pixel #2 and twist them together with the first wires. You'll want to solder both into the + and - holes on the pixel at the same time.



Solder the white wire into the IN pad on the pixel. Then measure out the white wire going to pixel #2 and solder it to pixel #1's OUT pad.



Continue measuring and soldering each pixel in order, making sure the wire lengths are about right for your design.



Once you've got all the pixels soldered up, plug your Circuit Playground into your computer and download the test sketch (linked below). Make sure all the pixels light up! If they don't, check to make sure your solder connections are all good, and that you've soldered each pixel in the right orientation.

It's easy to get the IN and OUT pads mixed up, and much harder to fix after your project is assembled, so don't skip this step!

Click "Download" and drag the resulting file onto the **CPLAYBOOT** drive that appears when you plug in your Circuit Playground and click the Reset button.

This sketch will light up a strand of pixels on pin A1 with a rainbow animation. Pretty!

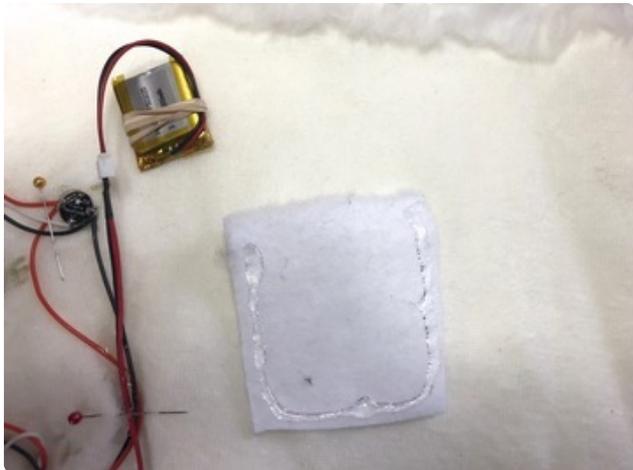
You can move on to the coding step from here, or continue and finish the bag assembly first.



Attach your pixels to your fabric following your design. I used E6000 glue to glue the pixels down, and then I used a needle and thread to secure the Circuit Playground face-down against the fabric. Sewing is best -- it's not a great idea to soak your Circuit Playground board in glue.

Remember to face the pixels down into the fabric so they show from the outside. Also, do NOT turn the pixels on while the glue is wet. The wet glue can act as a conductor and short out your pixels (so be patient!).

Make a pocket for your battery inside the bag, near the top where it's easy to get to for charging. Glue or hand-sew a small piece of felt on the inside of the bag. Thread the battery extension cable up through the bottom of the pocket before you secure it down.

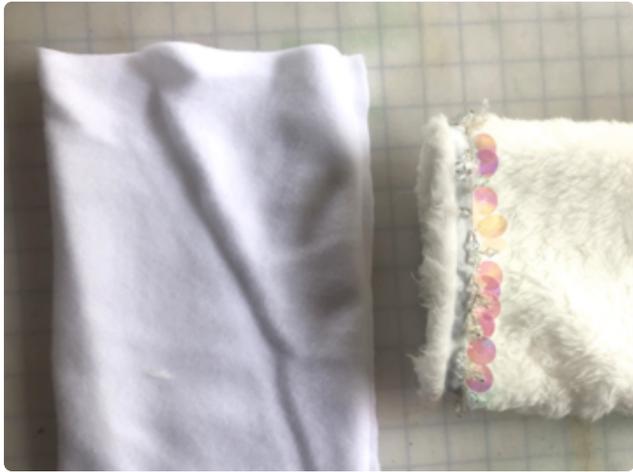


Carefully wind a rubber band around the battery and the wires, holding the wires against the body of the battery. This will provide strain relief -- the solder points for these wires are really delicate and break easily unless the wires are secured.



Cut a piece of felt or fleece the same size as your fabric. This will become an inner liner for your bag, so you can show it empty and/or use it to hold things, without getting tangled up in the wires.





Fold both pieces of fabric with RIGHT sides together, then sew along the side and bottom edge. Turn the main bag right-side out, but leave the inner bag inside-out.



Place the inner bag inside the main bag and sew carefully around the top edge, leaving an opening so you can get the battery in and out.

You can add trim or decorations to your main bag at this time too.



If you are using a fun-fur type fabric, you can cut a small slit right by the USB port of your Circuit Playground Express for easy programming.

Fun fur will hide this just fine. Other types of fabric you may need to program through the battery case opening (so make sure you left that big enough!)

Code with MakeCode

We'll use Microsoft's drag-and-drop code editor to create the magic. MakeCode is an easy way to get up and running with the Circuit Playground Express. No prior coding knowledge is needed, and it's an easy way to experiment and learn to think like a coder.

To get started, go to [makecode.adafruit.com \(https://adafru.it/wpC\)](https://adafru.it/wpC) and then choose **New Project**. You'll find yourself in the MakeCode Editor. From here, you can click on any of the colored tabs and drag blocks of code onto your workspace, then preview it using the Circuit Playground Express pictured on the left.

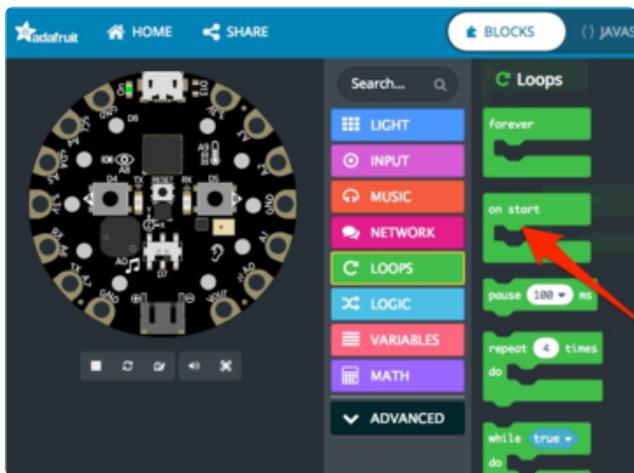
Once your code is written, plug your Circuit Playground Express into your computer via its USB port and click the "reset" button. All the lights will turn green and your Circuit Playground will appear as a drive on your computer called **CPLAYBOOT**. Simply drag your downloaded code onto this drive to program the Circuit Playground Express. Easy!

Head over to this [Intro to MakeCode \(https://adafru.it/AEp\)](https://adafru.it/AEp) guide for more info on getting started with MakeCode.

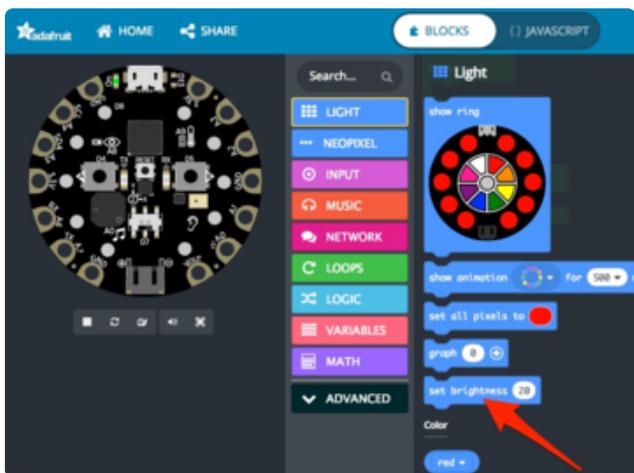
If you want to skip right to the end and work backwards, here's the completed MakeCode project.

Set Up the Lights

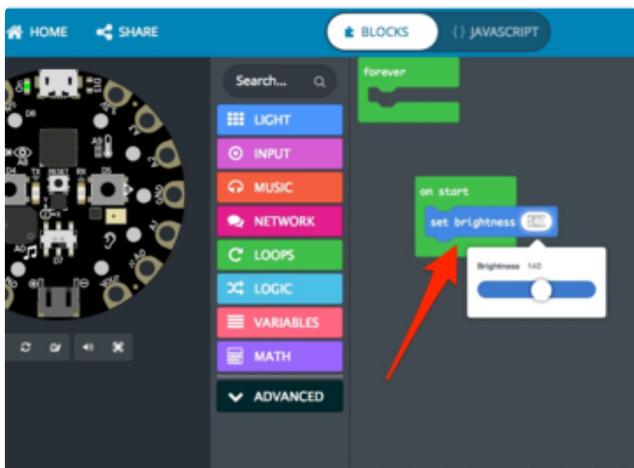
First, let's tell the Circuit Playground that we have a strand of lights wired up to pin A1. We'll also set the brightness of our lights.

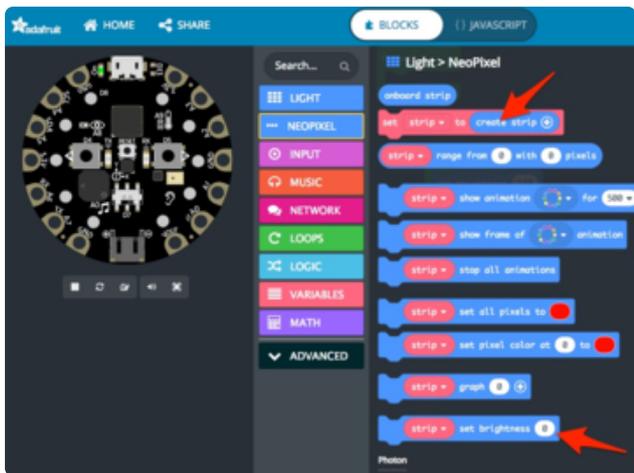


Click on the "Loops" tab and drag an instance of `on start` onto the workspace. Anything inside this bracket will happen just once, when the Circuit Playground is powered up.



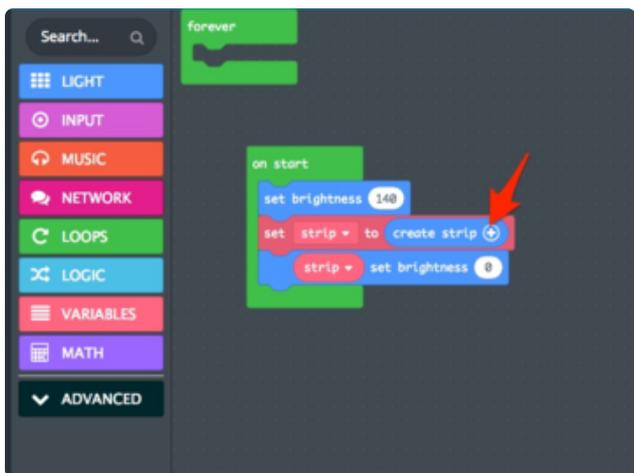
Click on the "Light" tab and find `set brightness 0` and drag it out into your `on start` loop. Set the brightness to something moderate, maybe 140. This will set the brightness of the lights on the front of the Circuit Playground Express.



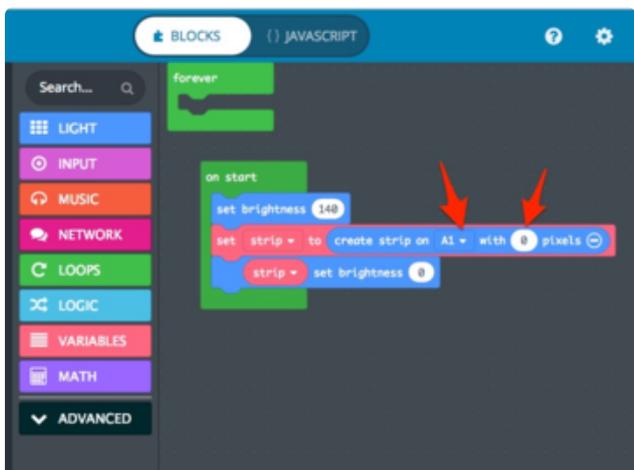


Click on the **LIGHT** tab again, and the **NEOPIXEL** tab will appear right under it.

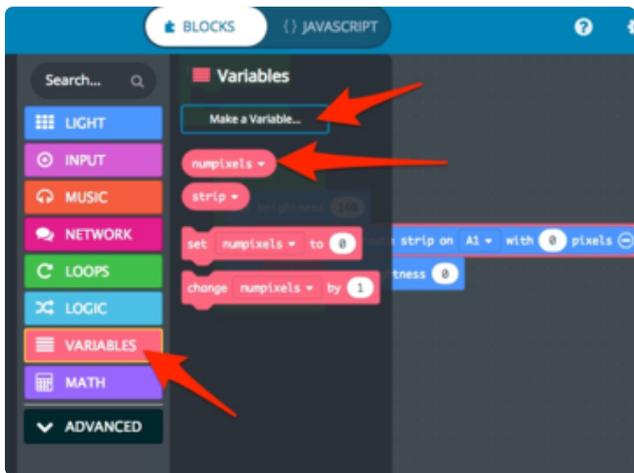
From **NEOPIXEL**, drag an instance of **set strip to create strip** into your **on start** loop.



Click the **+** next to create strip and MakeCode will add some modifiers. Be sure **A1** is selected (since that's the pad we soldered our lights to), then click the next **+** to expand the tab further, giving you the option to specify how many pixels are in your strip.

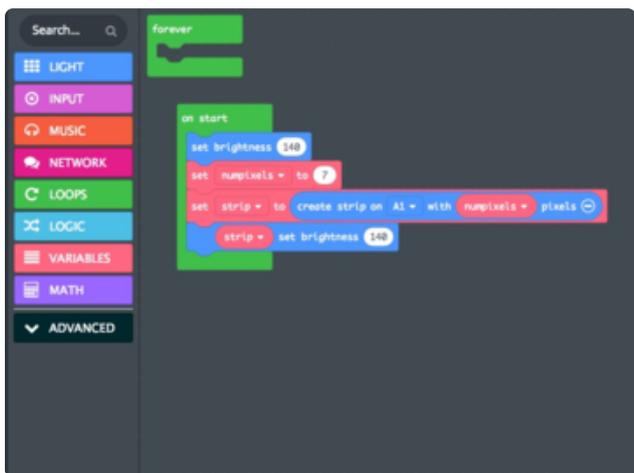


It's good practice to make this number a variable. That way if you decide to change the number of pixels in your strip later on, or if we want to reference it later in the code, we can do so easily.



Click the **VARIABLES** tab and make a new variable. Call it `numpixels`. Drag it out into your workspace and into the `create strip` instance, replacing the `0`.

Drag an instance of `set numpixels to 0` into your `on start` loop above `set strip`. Enter the number of pixels you've added to your project -- I have 7.



Then, go back to your **NEOPIXELS** tab (under the **LIGHTS** tab) and drag an instance of `strip set brightness 0`. Change this to `140` also.

Filling the Bag with Lights

Each time we "drop" a magic light into the bag, we want one light to turn on. We've got ten lights on the face of the Circuit Playground, and seven more spread out along the inside of the bag. But if we used all ten of the Circuit Playground's lights, it wouldn't look quite right. Dropping random lights into a bag would never light up a perfect circle! So we'll just use a few of the Circuit Playground's lights, to make it seem more random and jumbled, and then move on to our light strip.

Decide how you want to trigger your effect. You can use the buttons on the face of the Circuit Playground, or the capacitive touch pads around the edge. Capacitive touch works great with sheer fabric, but if you're using fun fur, the fabric is too thick to trigger the capacitive touch -- so maybe not the best option in this case. You could also use the "tilt" or "shake" features. It's all dependent on your magician's flair and style. You could even use multiple triggers to make it even more mysterious!

I decided to use "shake" -- so each time I shake the bag, another light will appear.

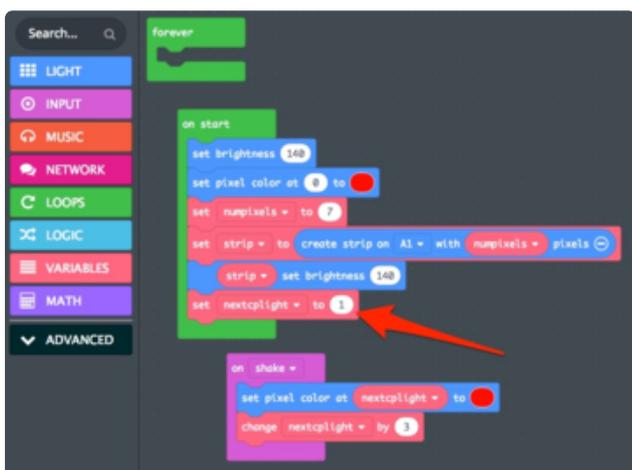
Add Variables

First we need to create a couple more variables. Go to your **VARIABLES** tab and create a variable called `nextlight`. We'll use this one to refer to the lights on the strip. Create one more called `nextcplight`. We'll use this one to refer to the lights on the Circuit Playground Express.



From the **INPUT** tab, drag an instance of `on shake` (or whatever your preferred trigger is) onto the workspace. From the **LIGHT** tab, drag `set pixel color at 0 to red` into this loop. Then go grab the `nextcplight` variable you just made and drag it into this block to replace the `0`. Finally, change the color to whatever color you'd like the light to be. My magic thumb tip is red, so I'll leave mine red.

Now we'll add some code that changes the variable `nextcplight` to refer to a different light. I want to use just 3 of the Circuit Playground Express' lights (to avoid the perfect circle look), so I'll tell it to start at light 1 and then skip ahead lights -- so we'll light up pixels 1, 4, and 7.

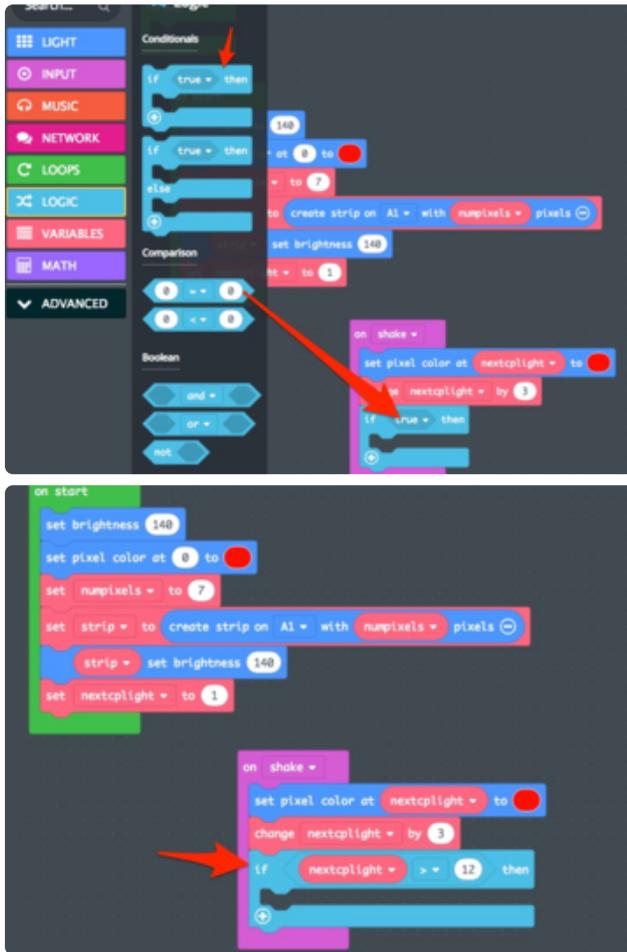


Go to the **VARIABLES** tab and drag an instance of `set nextcplight to 0` into your `on start` loop. Set this to `1`. Then drag an instance of `change nextcplight by 1` into your `on shake` loop and set this to `3`.

Try downloading the code and see if it works. Plug your Circuit Playground Express into your computer and press the reset button. Drag the downloaded code file onto the **CPLAYBOOT** drive that appears. Try shaking / triggering your effects and watch your lights come on.

Now let's do the same thing for the light strip. We want to use the same trigger (on shake) but we don't want these lights to come on until all three lights on the Circuit

Playground face have come on. We'll need to add a conditional statement (if/then) to make this work.



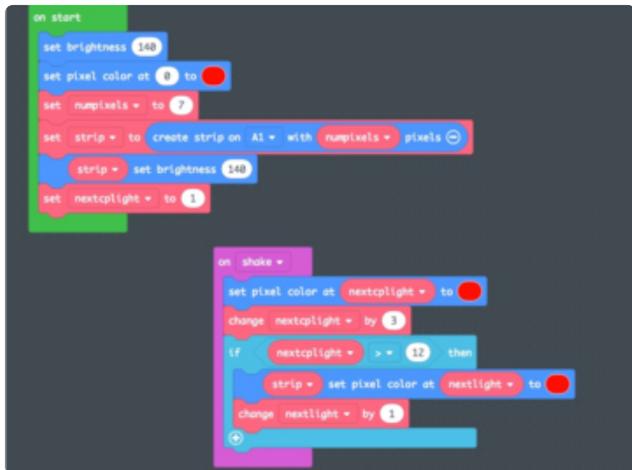
Under the **LOGIC** tab, drag an **if true then** loop into your **on shake** loop. Then drag a comparison block to replace the **true**.

Change the first **0** in the comparison block to your **nextcplight** variable. Then change the second **0** to **12** (because we're counting by 3s, remember?). Finally, change the operator **=** to **>** (greater than). Now the code means "if the last light-number we lit is greater than the number of lights onboard the Circuit Playground Express, move on to the next thing".

The next thing is to light the lights on your LED strip. So let's repeat the same idea, only using the blocks and values that will work for our strip.

Grab `strip set pixel color at 0 to red` under the **NEOPIXEL** tab.

(Remember, the **NEOPIXEL** tab will change your light strip, whereas the **LIGHT** tab changes the lights on the face of the Circuit Playground Express).



```
on start
  set brightness to 140
  set pixel color at 0 to red
  set numpixels to 7
  set strip to create strip on AI with numpixels pixels
  strip set brightness 140
  set nextlight to 1

on shake
  set pixel color at nextlight to red
  change nextlight by 1
  if nextlight >= 12 then
    strip set pixel color at nextlight to red
    change nextlight by 1
```

Drag in your `nextlight` variable (the one we made to refer to the light strip), and a `change nextlight by 0` block from the **VARIABLES** tab. Change the `0` to a `1` to light up the lights sequentially.

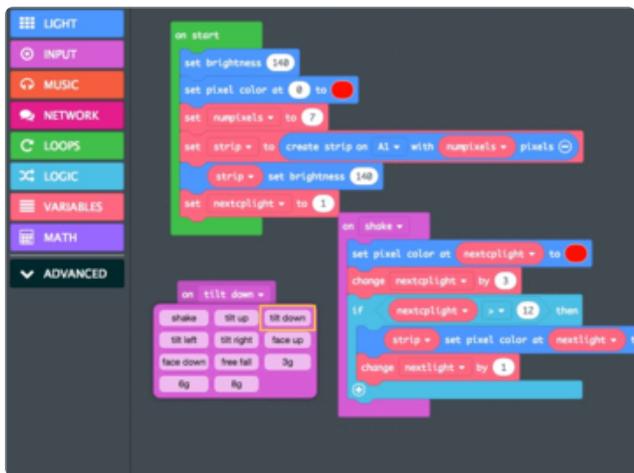
Try downloading this code again and see if it works the way you want.

Emptying the Bag

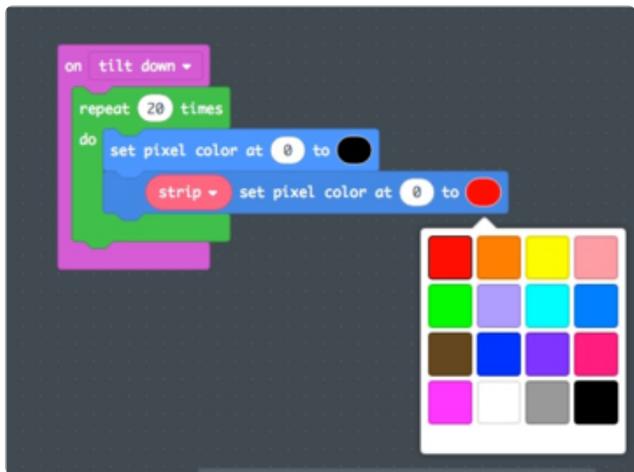
Hooray! We've gotten the bag to fill up with magic lights each time it's shaken. Now let's add a function that makes it appear to empty out when we're done.

We'll use the "tilt" input for this part. I want the bag to empty out when I turn it completely upside down, so the "tilt down" input is the one I'll use.

You could also make it trigger with a loud sound, a fast swing, or if you shine a bright light at the outside of the bag. So many possibilities!



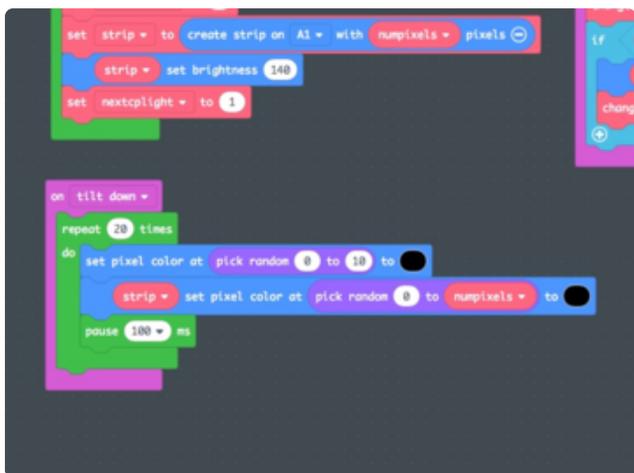
Drag another instance of **on shake** to your workspace, and change the input to **tilt down**. Under **LOOPS**, drag an instance of **repeat 4 times** into this loop. Change **4** to a number about twice as big as your total number of lights. I have 10 lights total (3 on the Circuit Playground Express and 7 in my strip) so I chose 20.



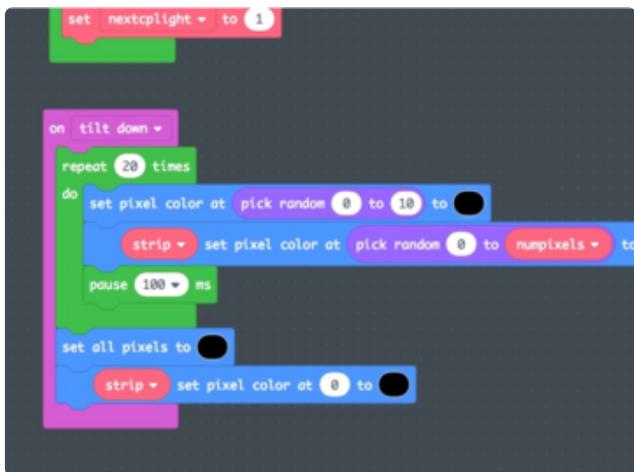
From **LIGHT** drag an instance of **set pixel color at 0 to red**, and from **NEOPIXEL** drag an instance of **strip set pixel color at 0 to red**. Change both to black -- this will turn the pixels off.



Then, we'll add some randomness so the lights go off in a random order. Under the **MATH** tab, drag two instances of **pick random 0 to 10** and replace both the **0**s in the previous two blocks. For the **strip** block, change the **10** to your **numpixels** variable (remember, that is defined as the number of pixels in your strip). For the other, we can leave it as **10** since there are 10 NeoPixels on the face of the Circuit Playground Express.



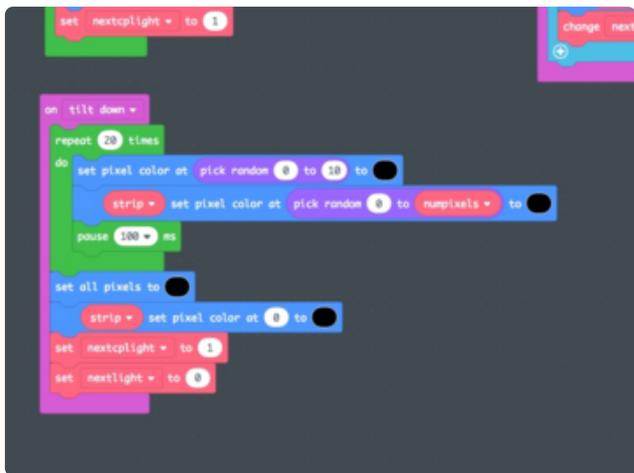
Then we'll slow it down a little. Drag an instance of **pause 100ms** from the **LOOPS** tab. You can change this variable to something pleasing -- I chose **15ms** for a fairly quick-fire emptying animation.



Let's add a failsafe just in case the random numbers don't turn all the lights off. After the **repeat** loop, drag an instance of **set all pixels to red** and **strip set all pixels to red**, and change both to **black**. This will make sure every single light gets turned off, and none appear to get "stuck" in the bag.

Resetting the Trick

One last thing to add! We want the trick to reset itself after the bag is emptied out, so you can immediately fill it again. To do this, we have to reset the **nextlight** and **nextcplight** variables back to their original state of **0** and **1**. Do this in the **on tilt down** loop, **OUTSIDE** the **repeat** loop, so the animation finishes before the variables get reset.

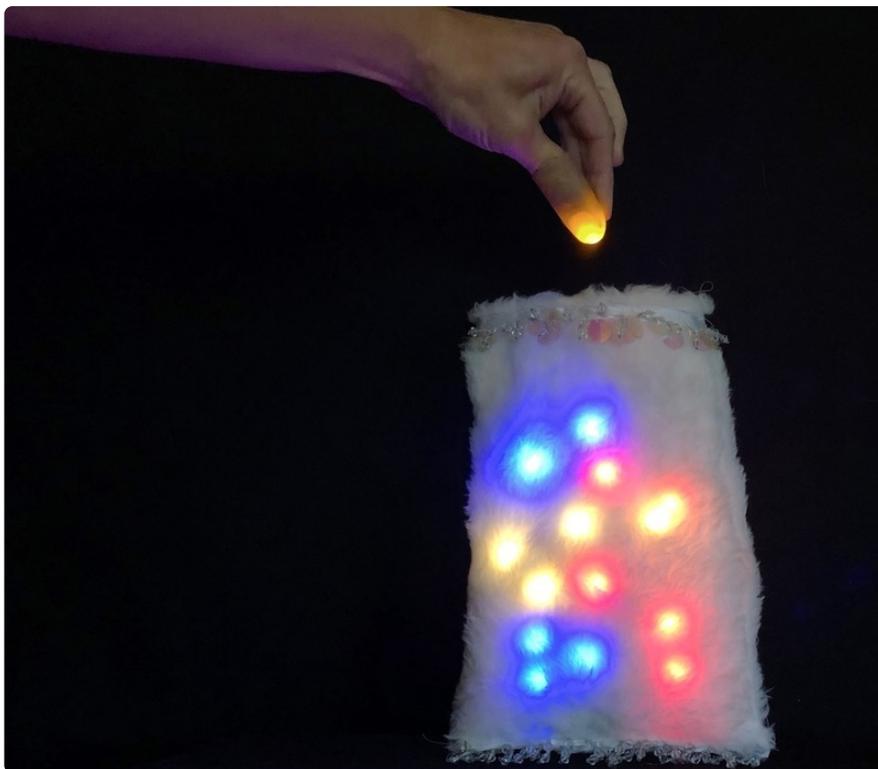


Drag an instance of `set nextlight to 0` and `set nextcplight to 0` into the `on tilt down` loop at the very end. Set `nextcplight` to `1`.

Use It

"Grab" a light out of the air with your magic thumb tip. Hold it above your bag and "drop" the light inside while giving the bag a shake. Each time you do this another light will appear inside the bag, until all your lights have been lit.

Once your bag is "full" of lights, up-end it and watch all the lights disappear. Your trick is now reset! Turn the bag right-side-up again and do it all over again.



More Ideas

Make this magic trick your own by adding other functions. Magic light thumb tips come in lots of colors, so maybe try using the buttons on the Circuit Playground Express to add multiple color effects.

You could also add sound effects -- play a sound each time a light is "captured" -- or have the lights dim or disappear when you shine a flashlight at your bag. You could trigger the lights to appear when the Circuit Playground hears a loud sound -- maybe when you shout "Abracadabra!". Or secretly hide some red jewels or giveaways inside the bag, so when you spill the "lights" out, they appear to have changed into jewels.

Make someone smile!