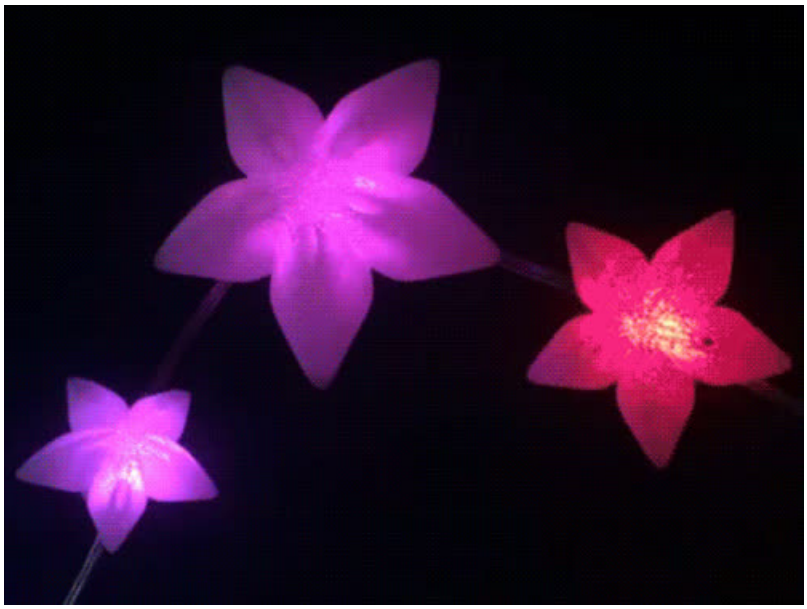
adafruit learning system

StarFlower Neopixel Strand with MakeCode

Created by Erin St Blaine



Last updated on 2018-08-22 04:05:43 PM UTC

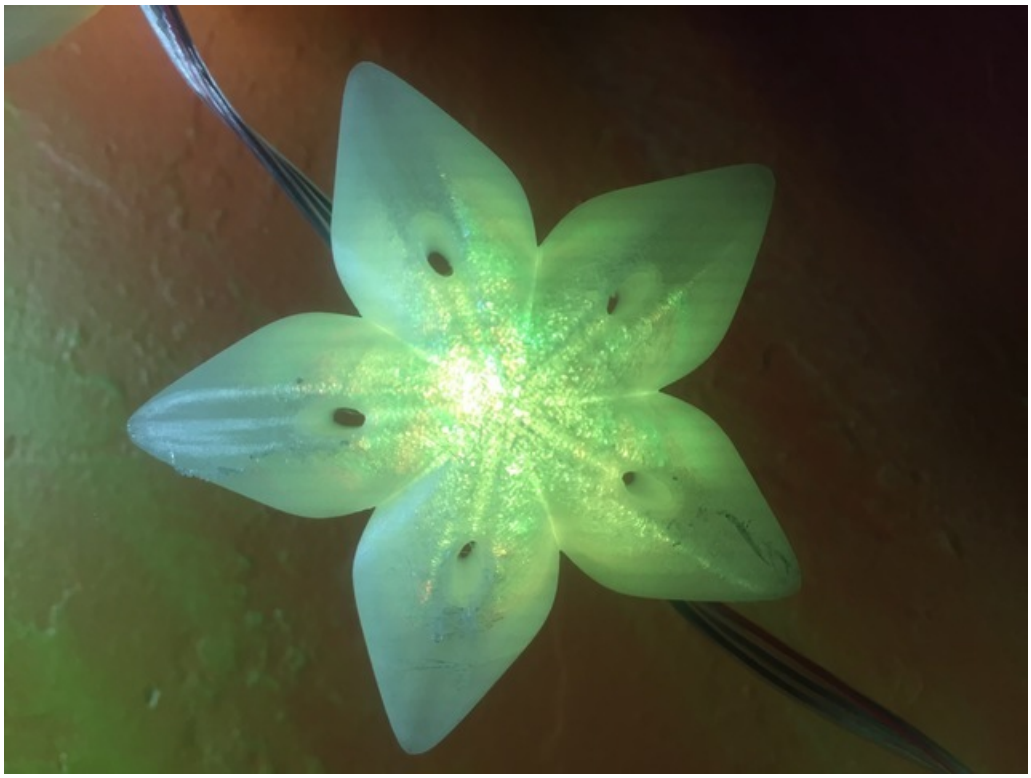
Guide Contents

Introduction

Make beautiful 3d printed glowing flowers to string around your room or your back yard. Use the Circuit Playground Express' onboard sensors to make the lights react to their environment: when it gets dark outside the lights come on automatically and shine bright, and when it's light out they will turn themselves quietly off.

This guide will cover modeling the flowers in Fusion360, connecting the light strand to a Circuit Playground Express, and creating animations and reactivity with the MakeCode editor.

There's so much you can do with the Circuit Playground Express. This project is a jumping-off place to creating your own designs and making the lights do whatever you can dream up.



1 x Circuit Playground Express
Circuit Playground Express
ADD TO CART

1 x Neopixel Strand
4" Pitch Neopixel Strand
ADD TO CART

1 x 2 pin cable
JST 2-pin cable
ADD TO CART

1 x 3 pin cable
JST 3-pin cable
ADD TO CART

1 x Screw Terminal

2.1mm Screw Terminal Adapter

ADD TO CART

1 x Power Supply

5v 2A power supply

ADD TO CART

1 x 1N4001 Diode

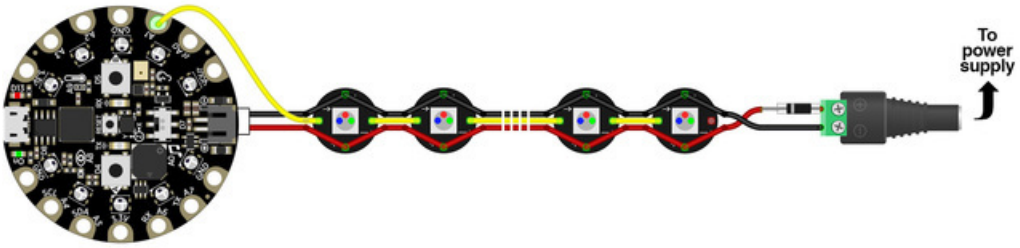
1N4001 Diode

ADD TO CART

Tools Needed

- 3d Printer (or 3d printing service)
- Soldering iron & accessories **OR**
- Wire strippers & alligator clips
- Heat gun
- Tiny screwdriver

Wiring Diagram



The great thing about these NeoPixel strands is that most of the wiring is already done for you; we just need to make some connections at the ends.

We'll show making an adapter for the microcontroller, then attach the NeoPixels with their included connector. Power connects at the opposite end of the light strand. That's something frequently overlooked with NeoPixel projects: while color data *must* travel in a specific direction from "in" to "out," **power can go either way.** In the diagram above, a 5V power supply is connected at the *end* of the strand, and we tap off this at the *start* of the strand to power the microcontroller, where the strand's data input is also connected.

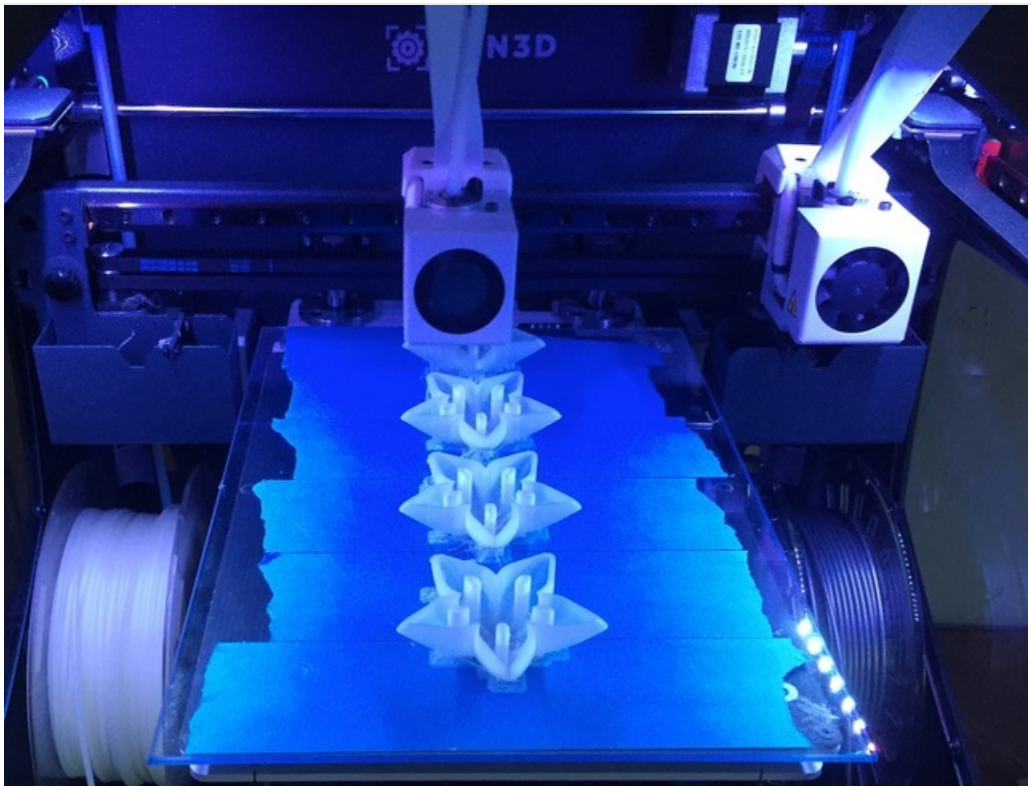
You can easily chain multiple light strands together with their included connectors. Just remember that if you have more than a few strands chained together, you may need a beefier power supply. (https://adafru.it/eCD) A modest 2 Amp supply is good for a couple strands, or more, depending on how you set the brightness level.

We've added a diode near the power supply to protect our pixels from too many volts. A lot of inexpensive power supplies aren't super accurate: they're labeled as 5v and they give off 5v-*ish*, and it's sometimes closer to 5.5 or 6.

These neopixel strands can be pretty finicky, and they really don't like more than 5v. Adding a diode between the power supply and the LED strand will compensate by dropping the voltage just enough, so we get around 4.5-5.7 volts.

Alternatively, you can use a switching power supply (https://adafru.it/CjB) with a 3 or 4.5 volt option, or one that's rated for 4.5 volts or lower. **Just be sure you don't switch the supply up to 6 or 12 volts,** or you could fry your controller and pixels!

3D Printing



Organic T-Splines

The shape was 3D modeled in Autodesk Fusion 360. It was created using t-splines, which is a part of the sculpting environment and toolset. The design is free to download and modify.

I've included three different sizes of flower, all of which will clip neatly onto the light strand. Choose your favorite size or print a few of each size to add lovely variety to your light strand.

https://adafru.it/ABL

https://adafru.it/ABL

https://adafru.it/ABM

https://adafru.it/ABM

3D Printing

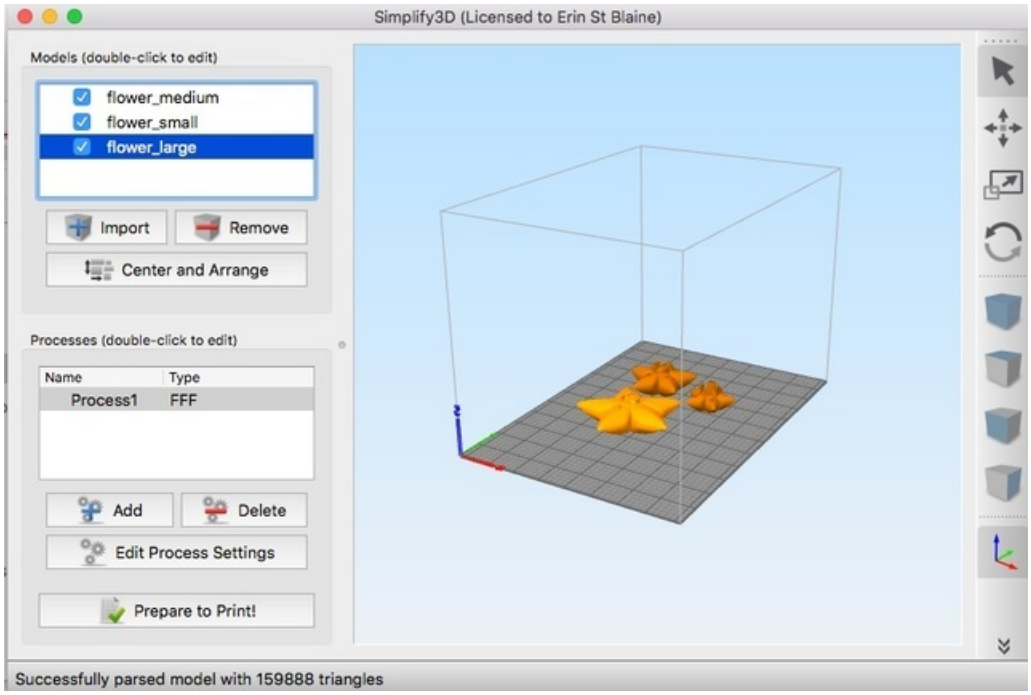
Because this part features no flat edges, it will require support material to 3D print properly. Most slicing software for 3D printers are capable of producing support material. A raft is also necessary for properly adhering the part to the bed.

Print them face down with the clips up in the air. This makes it easier to remove the supports without accidentally breaking off the clips.

If you don't have acccess to a 3D printer, can you upload the STL file to 3D Hubs (https://adafru.it/tC1) and have a local

hub 3D print and send the parts to you.

This part is solid in design, so to make it hollow you'll want to print with 0% infill.



Slice Settings

Below are some recommened slice settings.

- Glow in the Dark, clear, or white PLA @ 220C extruder / 65C bed
- Bluetape on non-heated beds
- **Support and Raft required**
- **0% Infill**
- 2 shells/perimeters
- 60mm/s printing speed

Support Settings

- support type: everywhere
- 60 degree overhangs
- support infill desnity 15%
- horizontal offset from part 0.30mm

Raft Settings

- 3 surface layers
- offset from part 3.00mm
- separation distance 0.14mm
- raft infill 50%

Removing Support Material

A pair of flush diagonal cutters can help remove support material from the part. It can also be removed by hand if it's of high quality.

Assembly

NeoPixel color data must travel from the “in” end of the strand to the “out” end. Examine the backs of the pixels *very* closely. Printed on the PCB, you might see either some data direction arrows, or the words “IN” and “OUT” between pads. The weatherproofing epoxy makes things murky and you may need to look at several pixels before finding one. This is important! If you connect the microcontroller to the wrong end, the lights won't work.

If you have multiple LED strands, plug them into each other to create one long überstrand.

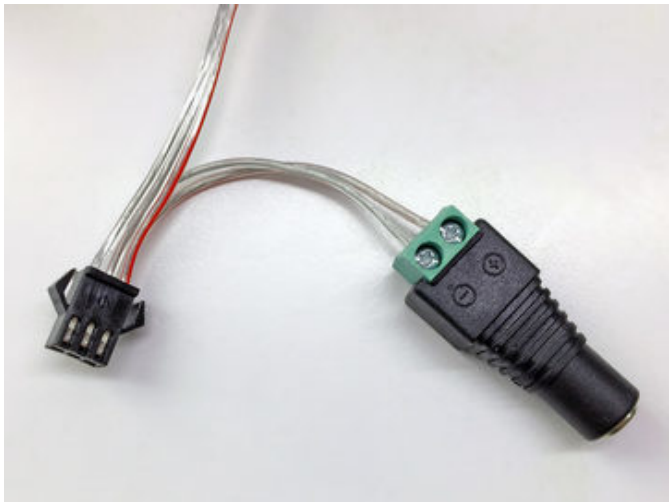
Each strand should have a pair of extra wires with exposed ends, for connecting power. Only **one** of these pairs is required. The rest should have their exposed tips trimmed flush or covered with tape or heat-shrink tubing to prevent electrical shorts.

For 1 to 3 linked strands, let’s use the pair of wires at the **end**. For longer strands, choose a pair near the **middle.**

Find the silver stripe on your diode.

Diodes have a specific *polarity,* passing current in only one direction...the silver stripe is the + end. So we want to connect the + side of the screw terminal to the “dark” end with no stripe.

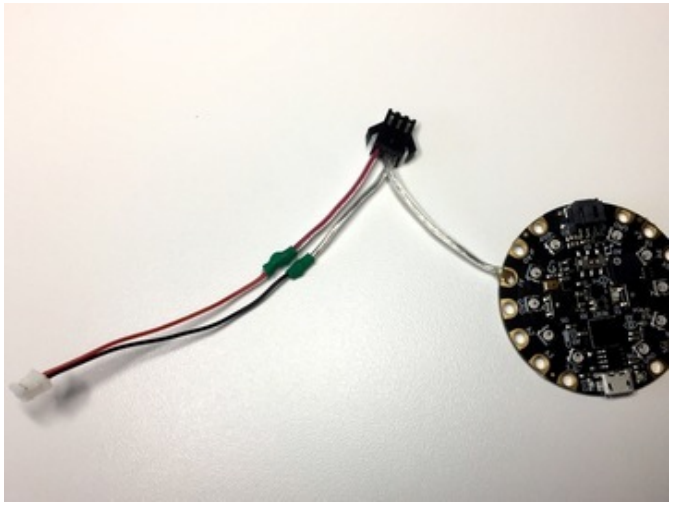
Bend the diode's leg on this side and slip it into the port on the screw terminal. Solder the other leg to the RED wire coming from the LED connector.

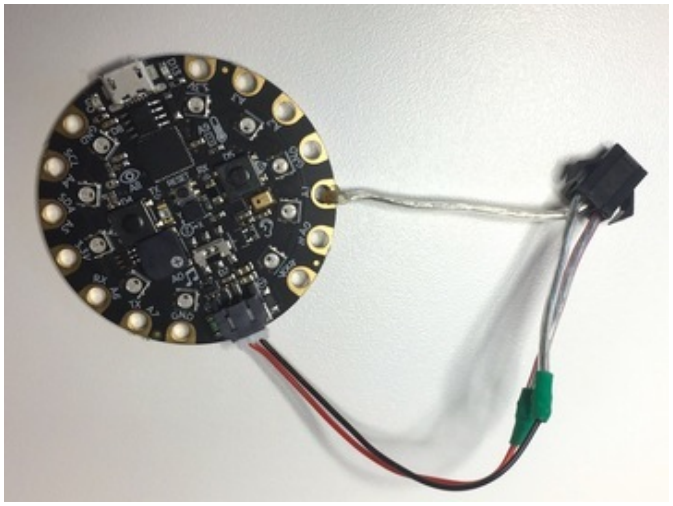


(Diode not shown) Connect a female DC power adapter using the screw connectors, being **super extra careful** to **follow the polarity markings** stamped on the DC jack:

- + (plus) connects to the strand’s **red** wire
- – (minus) to the opposite wire

You may need to strip away a little extra insulation from the wires to make a good connection with the terminals. Tug a little on each wire to make sure it's firmly in place.





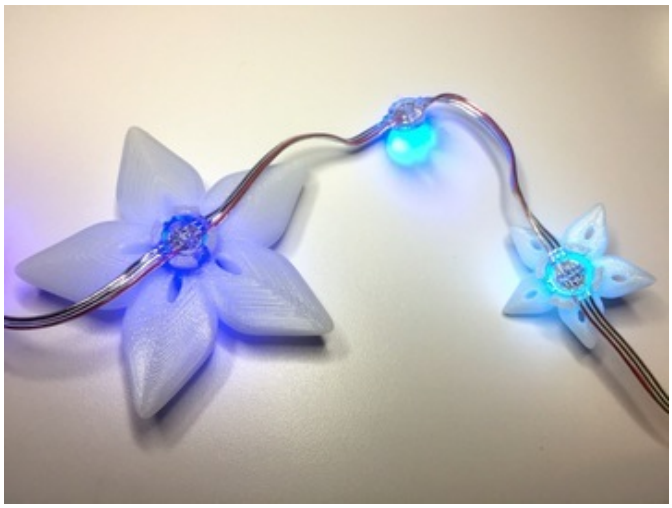
To connect the microcontroller at the “input” end of the strand, I soldered up this little **adapter cable** using a **3-pin JST plug** (to the LEDs) and a **2-pin JST cable** (to the Circuit Playground).

Don’t just follow the picture, take a good look at your actual hardware and be **super extra careful** to get the connections right:

- The **RED** wire from the LED strand should connect to the **RED** wire on the 2-pin JST plug.
- The **OPPOSITE** wire from the LED strand (furthest from the red wire) should connect to the **BLACK** wire on the 2-pin JST plug.
- The **MIDDLE** wire will get soldered to the Circuit Playground's A1 pin.

If you don’t have a **3-pin JST plug**, and you’re 100% okay sacrificing the “output” end of your LED strand, you can cut that plug off and use it as an input-end adapter. Do this at the mid-point between the plug and the last pixel, so there’s enough wire to be useful. Was your DC jack connected to the extra power wires there? That’s okay! Remember, the strand can accept power from *either* end.

If you're doing this project with kids, you can use alligator clips instead of soldering. It won't stay permanently attached, but it's a fun and easy way to get the lights up and running.



Clip the 3d printed flowers gently onto each light. If they don't seem to fit right, turn them 90 degrees -- the spacing between the clips is not exactly even; the wire goes between the wider clip spacing.

If they still don't quite want to fit, use a heat gun on the clips for just 5-10 seconds to soften them a little bit, then press the flower onto the light. When the plastic cools it will grab and hold nice and tightly.

Programming with MakeCode

The Circuit Playground Express can be programmed a number of ways: it will run Arduino code, CircuitPython, or you can program it with MakeCode.

Microsoft MakeCode for Adafruit is a web-based code editor for physical computing. It provides a block editor, similar to Scratch or Code.org, and also a JavaScript editor for more advanced users.

This means you can drag and drop light animations and functionality using the Circuit Playground Express' onboard sensors without ever writing a single line of code. Just snap the blocks together and watch your lights dance.

https://adafru.it/AEp

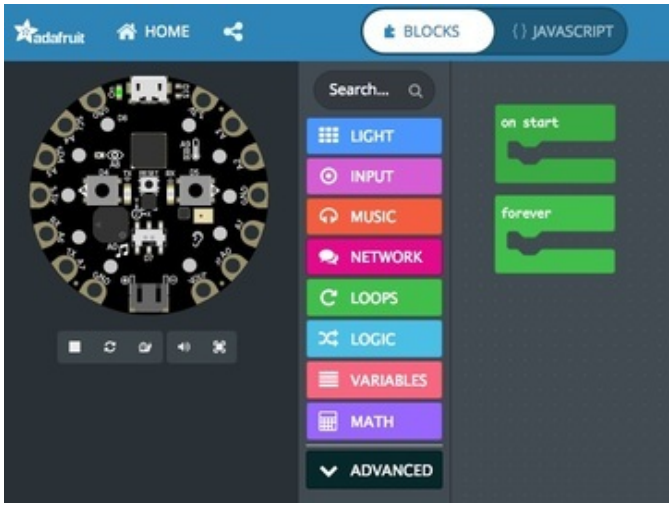
https://adafru.it/AEp

For this project, I used the Circuit Playground's onboard light sensor to turn the lights on automatically when the room gets dark. When the room brightens back up, the lights will turn off.

The lights are running MakeCode's built-in rainbow animation. Check out this guide (https://adafru.it/CjC) for more info on working with neopixel strands and building your own animations.

Go to https://makecode.adafruit.com/ (https://adafru.it/wmd)and select "New Project".

Set Up the Light Strand



We soldered our light strand to pin A1. Tell the code by adding an `on start` loop (Loops > `on start`). This block will run ONCE when the board powers up, so it's the perfect place to define our setup.

Drag the `on start` block above the `forever` block.

Next go to "Variables" and drag `set (item) to 0` into the `on start` block.

Change `(item)` to `(strip)`.

Go to the "Neopixel" bin (you may need to click the "Light" bin to make it appear). Drag `create strip on (A1) with (24) pixels` into the block you just made, replacing the "`0`" field.

In the "Neopixel" bin, find `strip > show animation () for 500 ms`. Drag this block into the `forever` loop.

Download Code

Let's test to see if it's working.

1. Plug your Circuit Playground Express into your computer with a USB cable.
2. Click the reset button.
3. Green lights will appear on the Circuit Playground's face and it will appear in your list of devices, called **CPLAYBOOT**.

If you don't see this, try double-clicking the reset button instead of single-clicking.

Click the pink **Download** button on your MakeCode screen and the code you just made will download to your computer. Drag it onto the **CPLAYBOOT** device.

Plug the light strand into power and you will see a pretty rainbow animation. Hooray!

Add Ambient Light Sensing

Let's make the lights turn on auto-magically when it gets dark, and off again in bright daylight. We can do this using the Circuit Playground Express' onboard light sensor and a little bit of experimentation.

From the pink **Input** bin, drag four blocks:

- two instances of `on light (dark)`
- two instances of `set (dark) light threshold to 0`

From the **Neopixel** bin, drag two blocks:

- two instances of `(strip) set brightness to (0)`

We want the lights to come ON when it gets dark in the room, and go OFF when it gets bright.

Place one instance of `(strip) set brightness to (0)` inside each `on light ()` loop. Leave the `(light)` one at 0, and set the `(dark)` one at 150 or so, or whatever you'd like your maximum brightness to be (on a scale of 1-255, 255 being the brightest they can go).

Next, place both the `set (dark) light threshold to 0` blocks inside the `on start` loop. Change `(dark)` to `(light)` on one of them, so one triggers above a certain brightness and the other triggers below a certain darkness.

Calibrate

Every room or environment is different, so you'll need to experiment a bit to get the thresholds right. I found that `set (bright)` to 21 and `set (dark)` to 32 works well in a window-lit bedroom -- when I turn on my overhead lights, the light strand goes off, and when the overhead lights go off, the light strand comes on.

Download the code and drag it to **CPLAYBOOT** to test and see where your light threshold needs to be. Experiment until it behaves the way you want. A bright flashlight is helpful for experimentation!

Here's the completed project that you can play with directly.