



Flashing LED Strand with MakeCode

Created by Kathy Ceceri



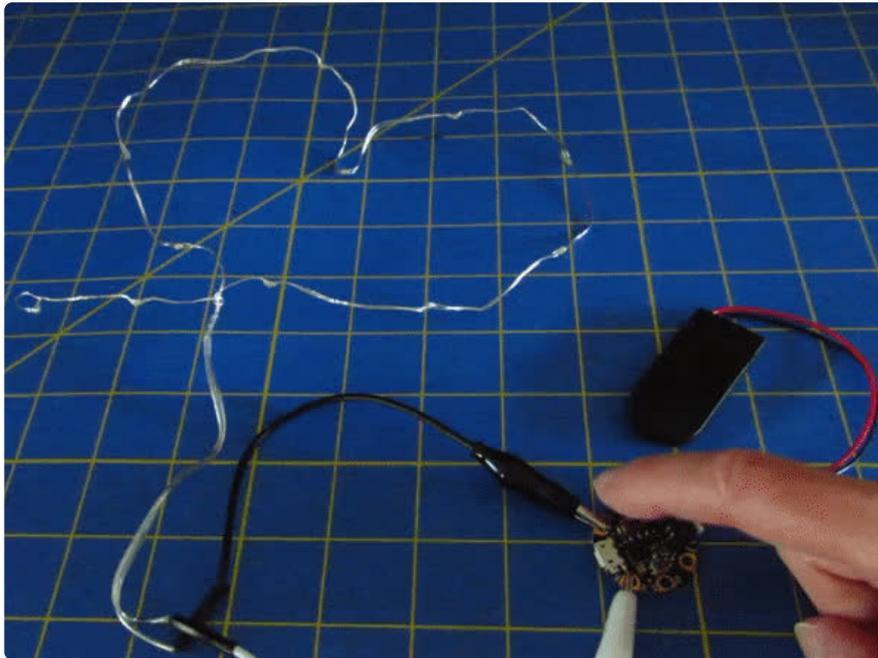
<https://learn.adafruit.com/flashing-led-strand-with-makecode>

Last updated on 2024-06-03 03:13:39 PM EDT

Table of Contents

Overview	3
• Parts	
Prep the LED Strand	7
Attach the Gemma M0	10
Code it in MakeCode Maker	14
• How to Download MakeCode to Gemma M0	
• What's Happening With the Code?	
Add to Your Wearables	18
• Other Ways to Use the Gemma LED Strand	
• Attachment Tips	

Overview



Are you a teacher on a budget looking for a programmable light-up system your students can use at home and keep for their own projects? Or maybe you want an inexpensive, simple way to incorporate lights and a microcontroller into your wearables permanently. Here's a quick and easy solution -- perfect for Halloween, cosplay, or just making outfits that really stand out!

This guide will show you how to use a Gemma M0 microcontroller and "fairy lights" -- a strand of one-color LEDs strung along a lovely silver wire that can be bent to hold its shape -- to add lights to wearables and decorations.

Although the Gemma doesn't have the functionality of the [Adafruit Circuit Playground Express board](http://adafru.it/3333) (still my favorite board for adding light effects to wearables), it does have a color-changing DotStar LED built in.

And with free, online, beginner-friendly [MakeCode Maker](https://adafru.it/C9N) (cousin to the version of [MakeCode](https://adafru.it/wWd) used to program the CPX), you can program the LED strand of lights to blink, dim, and brighten in unison.

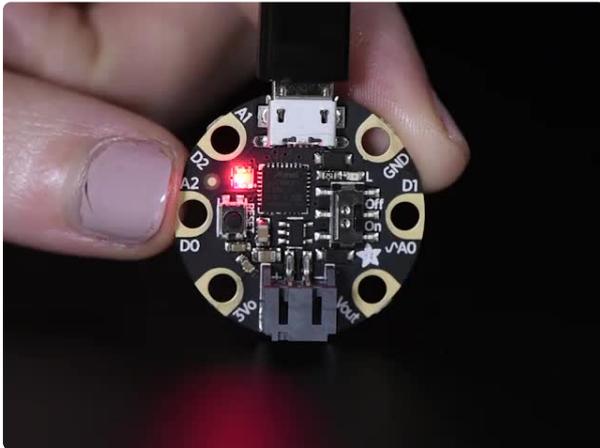
Make your light patterns run on a continuous loop, or use the touchpads on the Gemma M0 to control and change the light effects. Add these lights to a hat for Halloween or cosplay, or any wearable that can hold a board and battery pack.

As an intro coding project for kids and other beginners, the drag-and-drop MakeCode programming lets you get creative, fast!

You can use the same technique to connect a strand of lights to a Circuit Playground Express or a BBC Micro:bit, and still control them with MakeCode!

Parts

These are the parts you'll need at minimum. The Wire Light LED Strand also comes in warm white, red, green, and blue -- choose your favorite!



[Adafruit GEMMA M0 - Miniature wearable electronic platform](https://www.adafruit.com/product/3501)

The Adafruit Gemma M0 is a super small microcontroller board, with just enough built-in to create many simple projects. It may look small and cute: round, about the...

<https://www.adafruit.com/product/3501>



[Wire Light LED Strand - 10 Cool White LEDs + Coin Cell Holder](https://www.adafruit.com/product/894)

Add sparkle to your project with these lovely silver wire LED strands. These strands are very interesting, they use two silver wires that are coated so they don't short if they...

<https://www.adafruit.com/product/894>



[2 x CR2032 Coin Cell Battery Holder - 6V output - On/Off switch](https://www.adafruit.com/product/783)

This tiny coin cell battery holder is ideal for small portable or wearable projects. It holds two 20mm coin cells (2032 or CR2032 are the most popular size) in series to generate 6V...

<https://www.adafruit.com/product/783>



[CR2032 Lithium Coin Cell Battery](https://www.adafruit.com/product/654)

A perfect match for our sew-able coin cell holder. This non-rechargeable coin cell is CR2032 sized: 20mm diameter, 3.2mm thick. It...

<https://www.adafruit.com/product/654>



[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>

Optional Parts

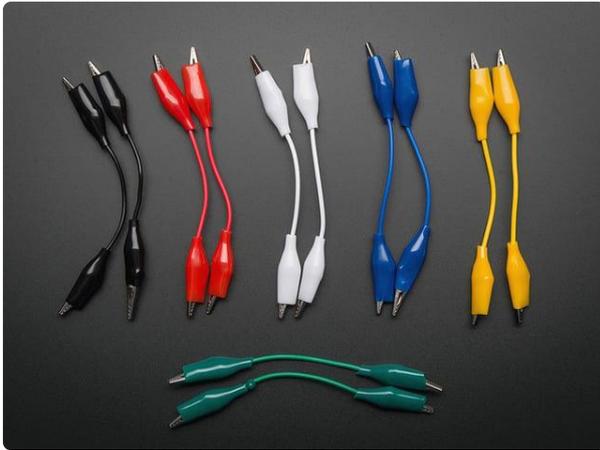
Depending on what you want to do with your programmable fairy lights, you might find some of the following useful:



[JST-PH Battery Extension Cable - 500mm](https://www.adafruit.com/product/1131)

By popular demand, we now have a handy extension cord for all of our JST PH-terminated battery packs (such as our Lilon/LiPoly and 3xAAA holders). One end has a JST-PH compatible...

<https://www.adafruit.com/product/1131>



Short Wire Alligator Clip Test Lead (set of 12)

Connect this to that without soldering using these handy mini alligator clip test leads. Approximately 4.5" overall cables with alligator clip on each end, color coded. You get 12...

<https://www.adafruit.com/product/1592>



Bolt-On Kit for Circuit Playground, micro:bit, Flora or Gemma

You have a Circuit Playground Express, and want to connect some wires to it for adding LEDs or sensors or speakers? You can use our...

<https://www.adafruit.com/product/4103>



Hakko Professional Quality 20-30 AWG Wire Strippers

These are the finest wire strippers we have used, and if you have to do a lot of wiring, you will agree! They have soft rounded grips - very comfortable to use, and precision ground...

<https://www.adafruit.com/product/527>

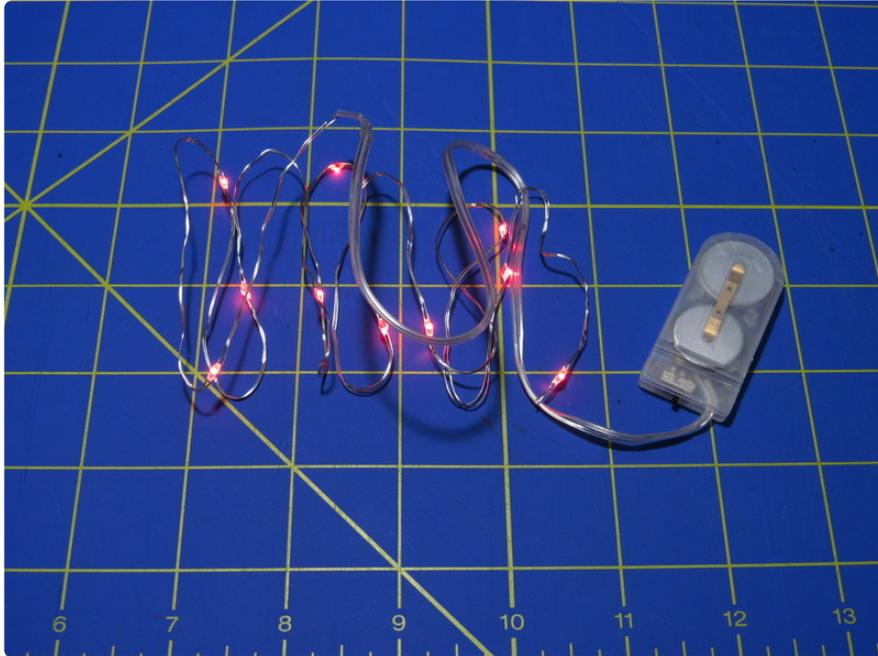


Multi-Colored Heat Shrink Pack - 3/32" + 1/8" + 3/16" Diameters

Heat shrink is the duct tape of electronics which I guess makes this heat shrink the colorful and exciting duct tape they sell at craft stores. This heat shrink comes in six...

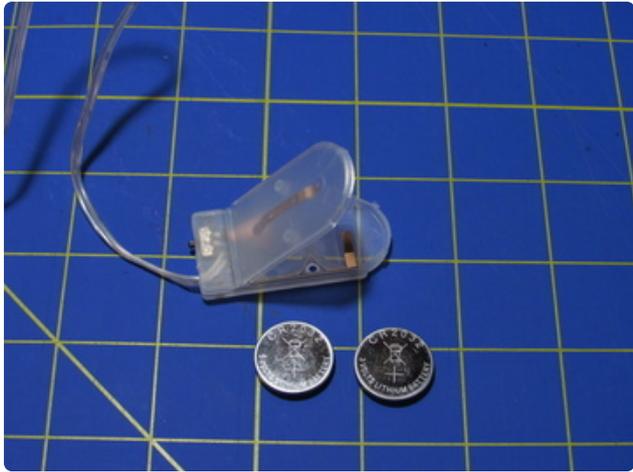
<https://www.adafruit.com/product/1649>

Prep the LED Strand



To connect the LED Strand to the Gemma M0 board, you'll need to do a little prep work first. Basically, you will disconnect the attached battery holder and expose the metal part of the wires. When you attach it, its power will come through the board, which requires its own battery holder.

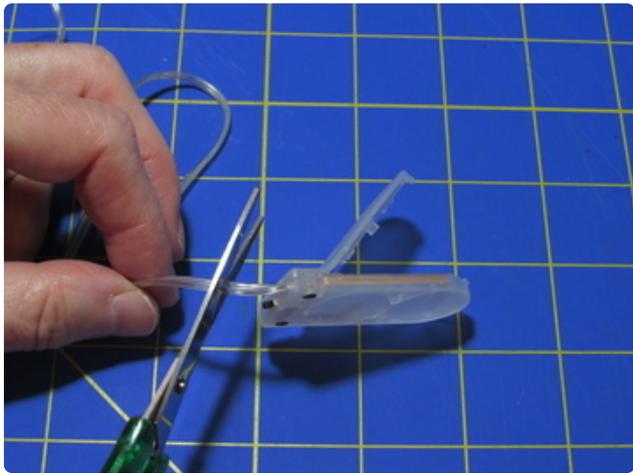
In the step that follows this one, you'll find some options for attaching the Gemma to the large alligator pads.



Detach the Battery Pack

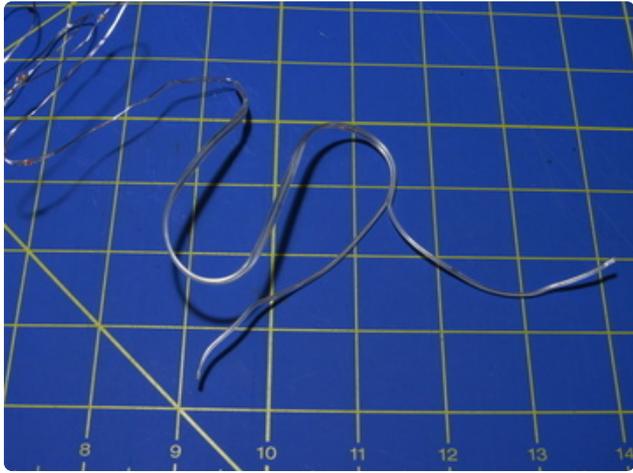
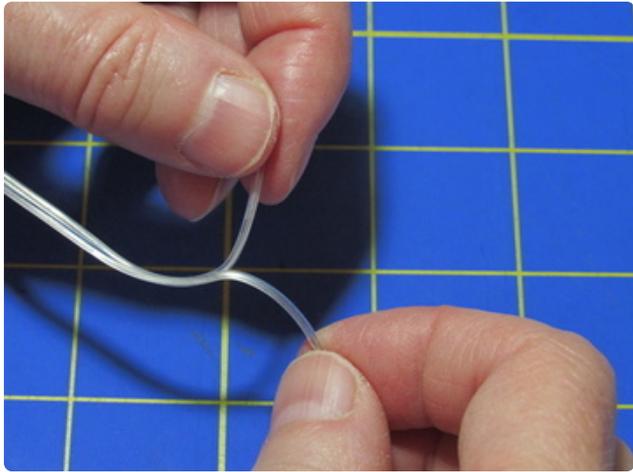
If you want to test the LED strand, insert the coin cells into the holder and turn it on.

Be sure to remove the batteries before going further!



With scissors or wire cutters, cut off the attached battery pack.

If you leave a couple of inches of wire attached, you can save it for some other use.



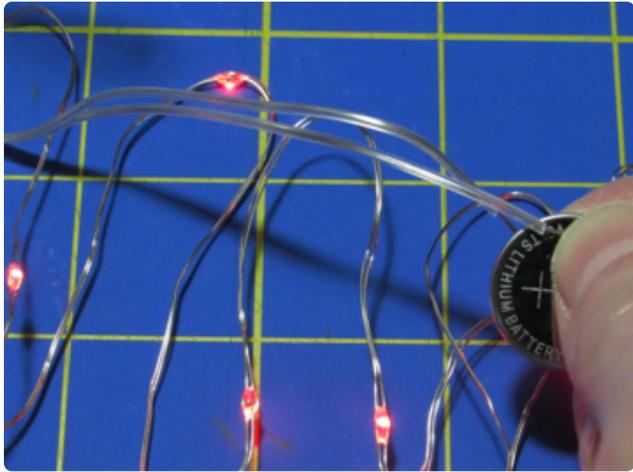
Prep the Wires

There are two wires going to the strand of lights, connected by a soft plastic coating.

Grab the cut ends of the wire, and carefully pull the wires apart for 3 or 4 inches.

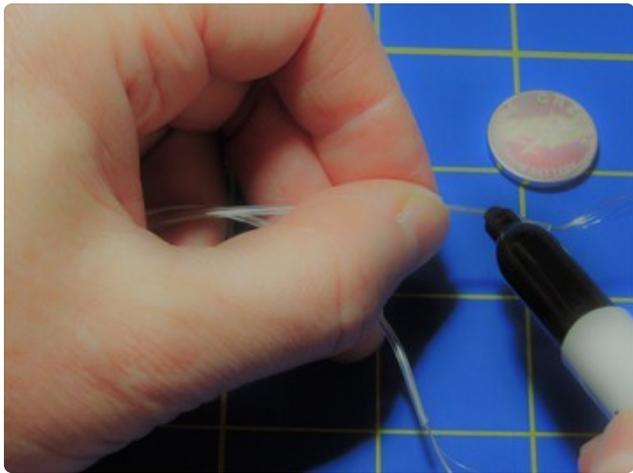
Strip a good half inch or so of the plastic insulation from the end of each wire to expose the metal wires inside.

To keep the bunch of thin wires together, give them a twist.



Test and Mark the Wires

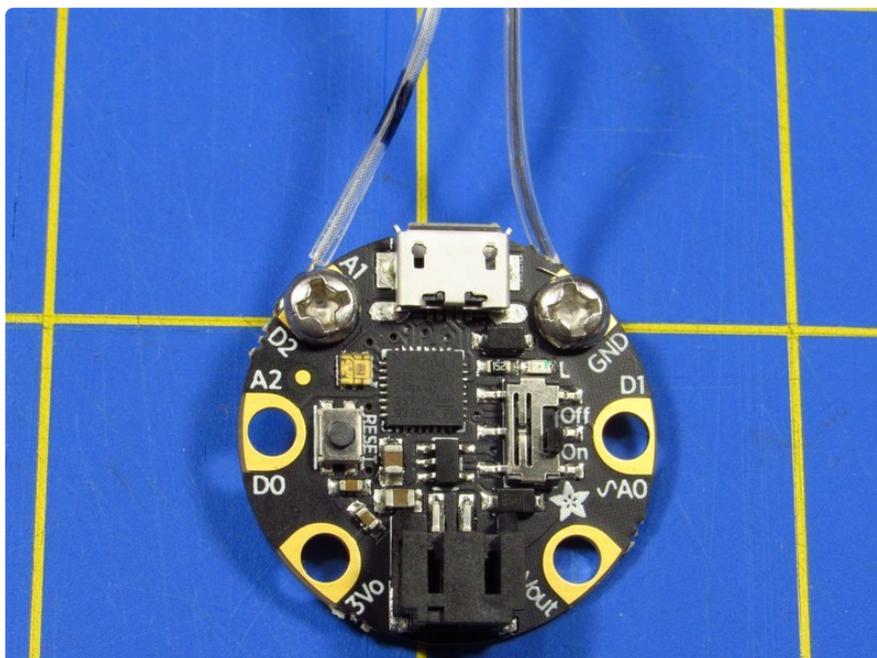
LEDs are polarized components, meaning they will only work if the positive wire is attached to the positive side of the battery and the negative wire is attached to the negative side. So next, you need to figure out which is which!



Take one of the coin cell batteries and pinch it between the two wires. Make sure each wire is only touching one side of the battery, and don't let the wires touch each other!

When the lights come on, make note of the wire touching the positive (+) side of the battery. Mark it with a pen or piece of tape.

Attach the Gemma M0



To use the Gemma M0 to control the LED strand, you have to connect the wires from the lights to the board.

Each wire will be hooked up to one of the Gemma's pins (those numbered connection points around the edge of the board):

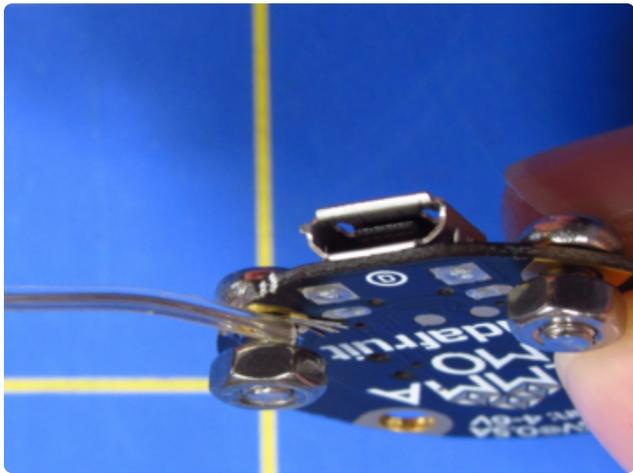
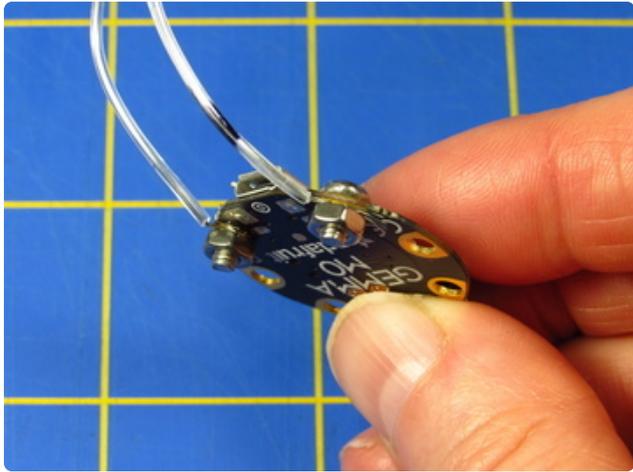
- The negative lead -- the wire you did not mark -- will be attached to the pin labeled GND, which stands for ground. This is the ground for your circuit.
- The positive lead -- the wire that has the Sharpie mark on it -- will be connected to the pin labeled A1/D2.

In between those two pins is the USB port. This is where you'll plug in the cable that connects the board to a computer for downloading code. Be sure to leave space for the cable between your wires!

You will also use the two pins on the sides of the board as touch pads to turn different light effects on and off, so leave those clear as well.

Depending on what you want to do with your lights, you have a few options for connecting the LED strand wires to the Gemma M0 that don't require soldering. They are:

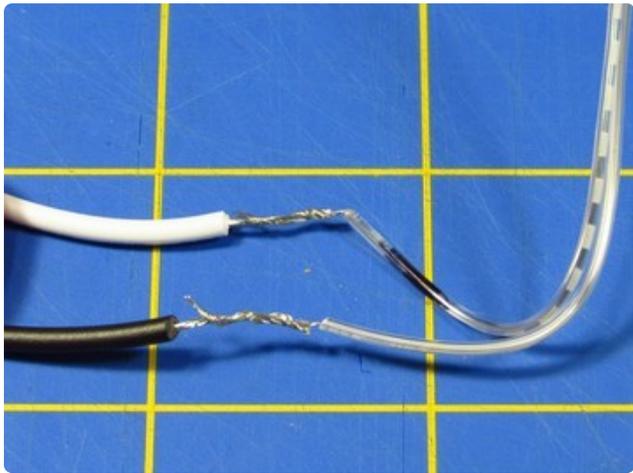
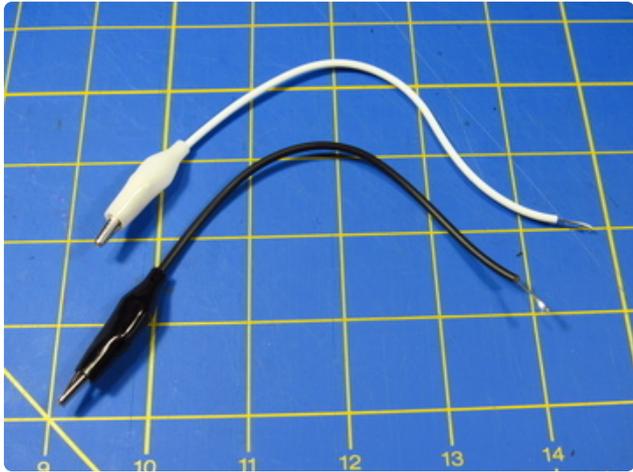
The LED light strand can get wet but the Gemma board cannot! If you want to be able to wash a project with the lights attached, make sure to pick an option that lets you detach the Gemma and batteries.



Bolt-On Option

The Adafruit-approved solution is to use the tiny screws that come in the Bolt-On Kit to hold the wires on. Just put a screw through the hole from the front of the board and screw on the hex nut loosely.

Next, wrap the wire around the shank of the screw. Then tighten down the hex nut to hold the wire against the pad.

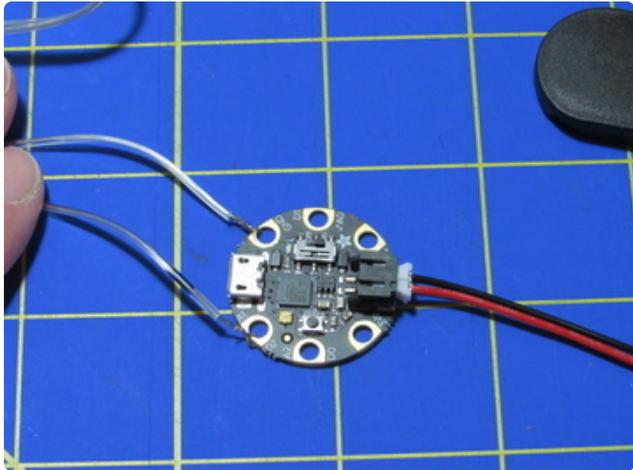


Alligator Clip Option

To make it easier to detach the Gemma M0 from the lights, you can attach a mini alligator clip test lead to the end of each wire. Be sure to make them different colors so you don't mix up the positive and the negative!

Just cut one end off the test lead, leaving one alligator clip and enough wire to connect to the LED strand wire. Strip the insulation off the test lead and twist the two wires together.

Cover the bare wires completely with electrical tape or a [heat shrink tube \(http://adafruit.it/1649\)](http://adafruit.it/1649).



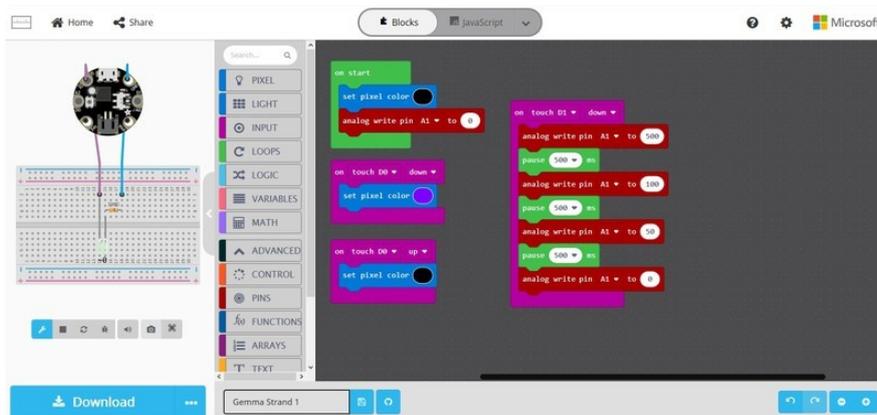
Quick and Dirty Option

Not recommended, but in a pinch just wrap the LED wires around the large holes on the board.

The problem is the thin wires will not hold up to much use without support.

You can try insulating them with electrical tape or a little [Sugru](http://adafru.it/437) (<http://adafru.it/437>), being careful not to cover any other parts of the board you need access to.

Code it in MakeCode Maker



Now that you have your LED strand hooked up to your Gemma M0, it's easy to control the lights using Microsoft MakeCode! The version to use with Gemma M0 is called [MakeCode Maker](https://adafru.it/C9N) (<https://adafru.it/C9N>). It lets you tell Gemma how to use its pins, where to look for input, and where to send output.

If you've never used MakeCode, check out the [getting started guide here](https://adafru.it/AEp) (<https://adafru.it/AEp>). The MakeCode Maker version work the same as the regular version.

In the example program shown here, one touchpad makes the onboard light turn on and off. This is a good way to test that the code is running. Since it's an RGB DotStar LED, you can make it any color you want, or even create a pattern of different colors!

The other touchpad controls all the lights in the LED Strand, in unison. In this program, they turn on, then get dimmer and dimmer until they turn off. Again, you can create any pattern you wish.

The onscreen simulator lets you see what your code will do before you download it to the Gemma board. For this project, the strand of LED lights is represented by an upside-down green LED on the breadboard.

To check out the sample program shown here, click the button below to open MakeCode Maker. Then click Edit to test it out and add to or change the programming as you wish. An explanation of the code is below.

[Click to Open Sample MakeCode](https://adafru.it/NVf)

<https://adafru.it/NVf>

How to Download MakeCode to Gemma M0

To download a MakeCode file to the Gemma M0:

- Connect the board to your computer with the USB cable.
- Press the reset button. You should see a green light and a red light on the board. This means it is in bootloader mode, and it's ready to receive code.
- Click **Download** on the MakeCode page and save the file. It will have a .uf2 extension.
- Look for a new drive on your computer labeled **GEMMABOOT**. Drag and drop (or copy and paste) the program into the drive.

If you are downloading the sample "Gemma Strand 1" program, you'll know the code was downloaded when the onboard LEDs turn off.

Try touching the pad labeled D0. The onboard DotStar LED should turn purple. When you take your finger off, the light should turn off.

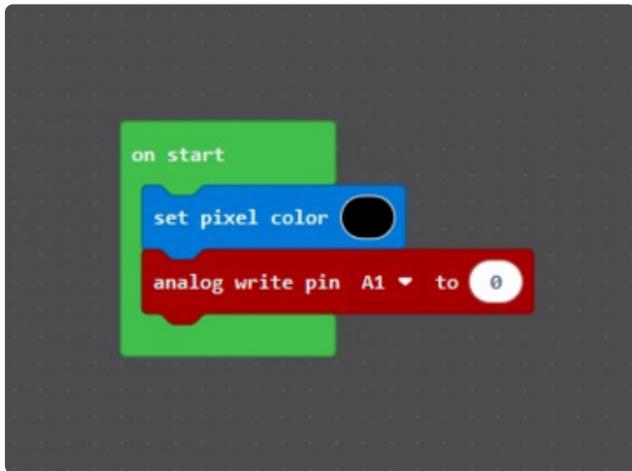
Then try touching the pad labeled D1. The strand of LED lights that you attached to the board should flick on, then get dimmer and dimmer until they go off again.

You may need to double-press the reset button to get your board into bootloader mode.

What's Happening With the Code?

Once you understand what those chunks of code are doing, you can play around with them to see what other effects you can create!

On Start

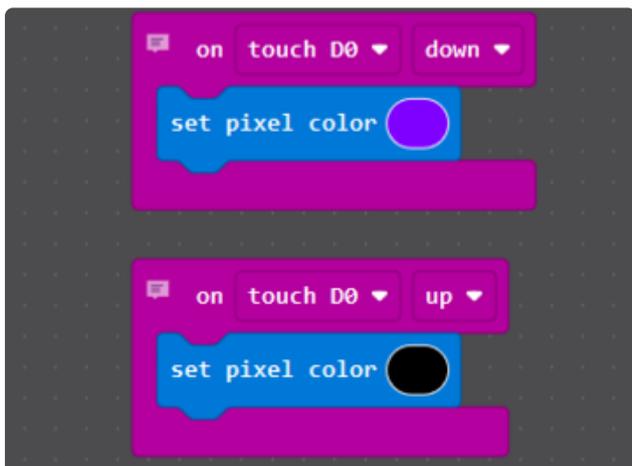


The **on start** block is found in the green Loops menu. Here, it is used to make sure the onboard LED and the strand are both off at the start.

Blocks from the blue **PIXEL** menu control the onboard LED. They give you different ways to adjust the color of the light. When you select black on the **set pixel color** block, it turns the light off.

The red **analog write pin** block lets you control how much power is sent to the pin number you specify -- in this case, Pin A1. In this example, it is used like a dimmer switch to make the strand of lights dimmer or brighter, as well as turning them on or off.

Coding the Onboard LED



The **on touch** blocks are found in the **INPUT** menu.

Selecting "down" means the touch pad indicated is registering a touch. "Up" means it is not being touched.

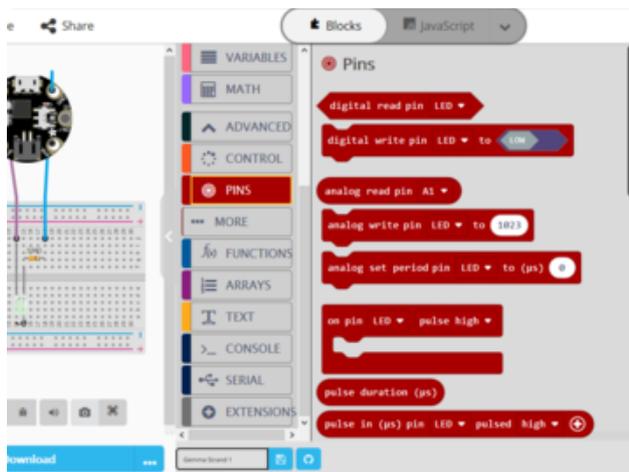
Here, they are used to make the D0 pad into an on/off switch that controls the onboard RGB light on and off.

So, when you touch pin D0, the light turns purple. Let go, and the light turns off!

Controlling the LED Strand

There are no premade blocks to control the LED strand, but you can use the red **PINS** blocks to tell Pin A1 what to do. To find the **PINS** menu, you must first click on **ADVANCED** at the bottom of the column of menus. That reveals additional menus.

The **analog write pin** block tells the pin you want to output a signal. ("Read" blocks tell the pin to look for a signal input.) Because it's analog, you can set it to different levels. (Digital controls essentially register only on or off. On the Gemma, some pins can be used for either analog (A) or digital (D), which is why they have both labels.)

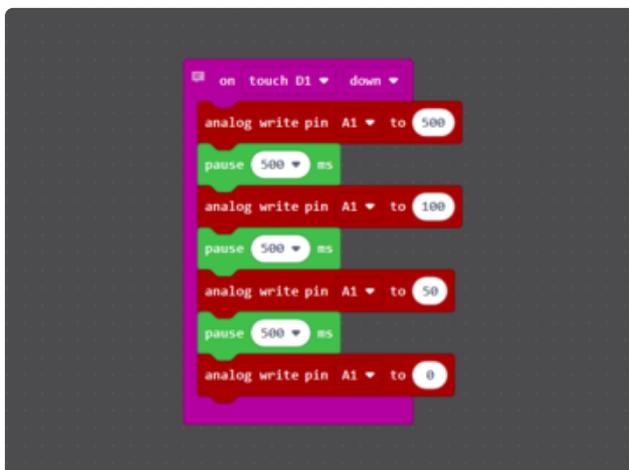


To choose which pin you want to control with the block, click on the down arrow to open the drop-down menu.

To set the amount, type a number into the white oval, or use the slider that opens when you click on the oval. Setting it to zero turns the lights off.

By default, the **analog write pin** block is set to the maximum, which is 1023.

A good rule of thumb is to set the level of the analog pin no higher than halfway. You'll make the battery last longer, and avoid burning out the LEDs.

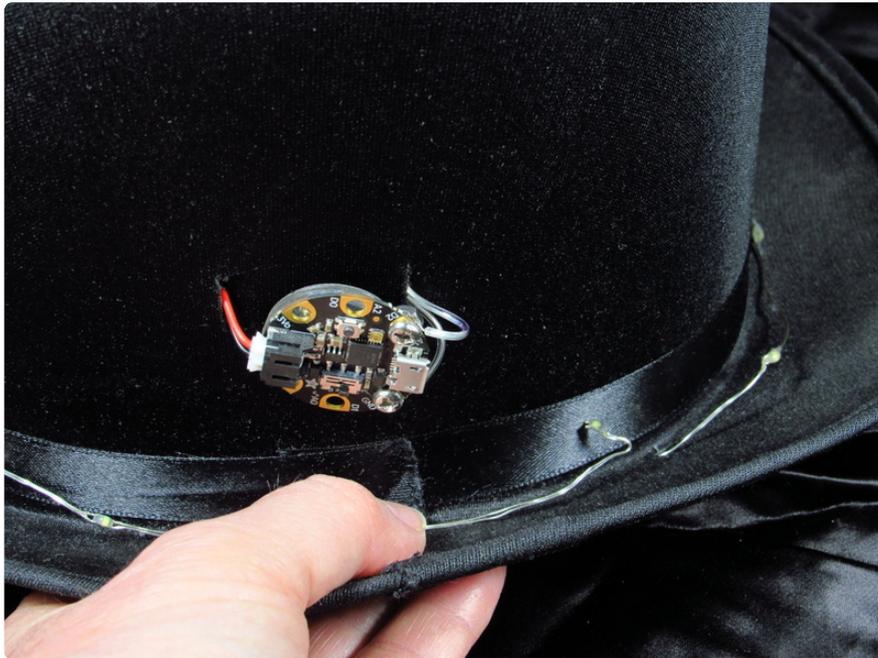


In this case, the settings go down in four steps, from 500 to 100 to 50 to 0. This makes the light get gradually dimmer.

To control how long the light stays at each level of brightness, there are **pause** blocks from the **LOOPS** menu inserted between each piece of code.

Again, you can type in a number, or choose from the drop-down menu. All times are in milliseconds (ms). One second is 1000 ms, so 500 ms is half a second.

Add to Your Wearables



The flashing LEDs look great on a hat, and they're easy to add with just a little adhesive. The quickest way is just to wrap the lights around the crown and secure the Gemma and battery holder to the brim.

To attach the Gemma to the hat, use adhesive dots or peel-and-stick Velcro dots. You may need to stack a few so the adhesive is higher than the screws holding the wires.



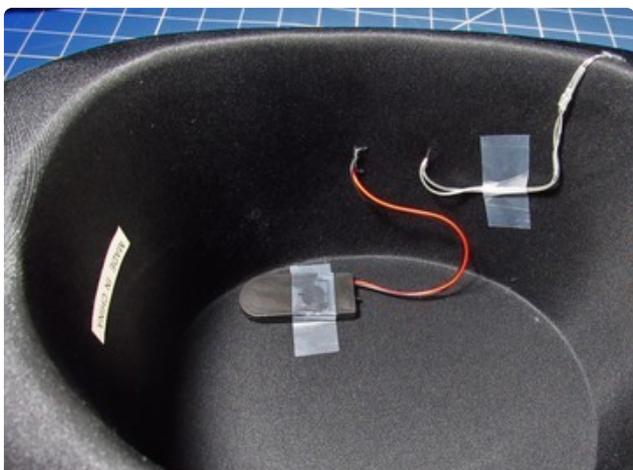
If you want to hide the excess wires inside the hat, mark two small holes on either side of where you want to place the Gemma LED. Leave enough room so the wires don't have to bend straight down, to avoid straining and possibly breaking them.



Then carefully cut a slit for the light wire. (A [ceramic craft knife \(http://adafru.it/4306\)](http://adafru.it/4306) is safe enough for kids to do this themselves, with adult help.) Thread the end of the light wire through the slit.



For the battery pack wire, cut a small flap to fit the JST plug through. Press the flap closed around the wires when you're done.



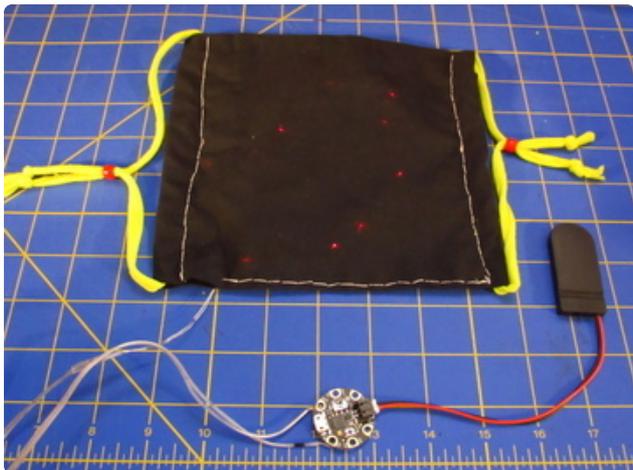
Then mark where you want the lights to come out, and cut a slit where the crown meets the brim. Bring the light wire out through that hole and wrap it around the hat.

Finally, attach the battery pack to the top of the hat. (Tape will work temporarily, but for permanent use, put adhesive under the battery pack so you can open it to replace the batteries.)

Don't forget, the Gemma board and battery holder are not washable!
Make sure to remove them before getting your project wet.

Other Ways to Use the Gemma LED Strand

The Flashing LED Strand can be used in all kinds of wearables and non-wearable projects! You can use it as a substitute for Circuit Playground Express in projects like these:



To use it with the [Light-Up Mask \(https://adafru.it/NVA\)](https://adafru.it/NVA) project, bunch up the strand of light and press it flat so it fits in the pouch. You can also slip the Gemma into the pouch, if you program the light pattern to run continuously (so you don't have to touch your face). Thread an extension cable from the Gemma to the battery pack through the mask, so you can carry the batteries in a pocket.

Other ideas:

- Add it to non-programmable LED strand projects, like the [Twinkly Earwarmer Headband \(https://adafru.it/NVB\)](https://adafru.it/NVB).
- Build them into a Halloween decoration, like the [CPX Glowing Disembodied Hand \(https://adafru.it/NVC\)](https://adafru.it/NVC).
- Make a separate wristband controller to hold the Gemma and battery pack, using the same technique as for the headband. Then use extra-long alligator wires to connect the lights to the controller. On a light-up jacket, for example, you can run the wires down the inside of your sleeve.



If your project has a pocket or pouch, you can just tuck the Gemma board and battery holder inside. If that's too long of a reach, add a JST extension cable.

Attachment Tips

- The most secure way to attach the LED strand is by sewing it on. Find instructions in the [Twinkly Earwarmer Headband project](https://adafruit.it/NVD). (<https://adafruit.it/NVD>)
- To hold the light strand on without sewing, you can try peel-and-stick adhesive dots or fabric fuse in sheets or tape.
- For a more permanent attachment, try fabric glue or even puffy paint.

Experiment! The Gemma M0 and LED Light Strand are inexpensive and versatile enough to test out in a wide ranges of projects -- and they look great!