Playa Festival Bike
Created by Erin St Blaine
Guide Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>Planning</td>
<td>5</td>
</tr>
<tr>
<td>Essentials</td>
<td>5</td>
</tr>
<tr>
<td>Extras</td>
<td>5</td>
</tr>
<tr>
<td>Pie-in-the-Sky Ideas</td>
<td>5</td>
</tr>
<tr>
<td>Painting &amp; Decoration</td>
<td>7</td>
</tr>
<tr>
<td>Rear Fiber Optics</td>
<td>11</td>
</tr>
<tr>
<td>Materials List</td>
<td>11</td>
</tr>
<tr>
<td>Adafruit Parts Used</td>
<td>11</td>
</tr>
<tr>
<td>Additional Parts Used</td>
<td>11</td>
</tr>
<tr>
<td>The Vision</td>
<td>12</td>
</tr>
<tr>
<td>Wiring Diagram</td>
<td>15</td>
</tr>
<tr>
<td>Important Note about Power!!</td>
<td>15</td>
</tr>
<tr>
<td>Software</td>
<td>16</td>
</tr>
<tr>
<td>Install CircuitPython</td>
<td>16</td>
</tr>
<tr>
<td>Adafruit Circuit Python Libraries</td>
<td>16</td>
</tr>
<tr>
<td>Upload code.py</td>
<td>16</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>17</td>
</tr>
<tr>
<td>Important Note about Power!!</td>
<td>18</td>
</tr>
<tr>
<td>Electronics Assembly</td>
<td>19</td>
</tr>
<tr>
<td>Wiring</td>
<td>19</td>
</tr>
<tr>
<td>Playa-Proofing Your Electronics</td>
<td>19</td>
</tr>
<tr>
<td>Battery Case</td>
<td>20</td>
</tr>
<tr>
<td>Important Note about Power!!</td>
<td>21</td>
</tr>
<tr>
<td>Mount to the Bike</td>
<td>25</td>
</tr>
<tr>
<td>Frame Lights</td>
<td>31</td>
</tr>
<tr>
<td>Wiring the Lights</td>
<td>33</td>
</tr>
<tr>
<td>Code with MakeCode</td>
<td>34</td>
</tr>
<tr>
<td>Code It Yourself</td>
<td>34</td>
</tr>
<tr>
<td>Hue, Saturation and Value</td>
<td>36</td>
</tr>
<tr>
<td>Testing It Out</td>
<td>36</td>
</tr>
<tr>
<td>Electronics Assembly</td>
<td>39</td>
</tr>
<tr>
<td>Mount and Seal</td>
<td>41</td>
</tr>
<tr>
<td>Sealing it Up</td>
<td>42</td>
</tr>
<tr>
<td>Repair Kit</td>
<td>44</td>
</tr>
<tr>
<td>Go Further</td>
<td>45</td>
</tr>
</tbody>
</table>

© Adafruit Industries

https://learn.adafruit.com/playa-festival-bike

Page 2 of 46
Get your bike ready for festival season!

This guide will detail taking an old Trek mountain bike from ordinary to Extra-blingbling! We'll decorate the heck out of it, add lights, and then waterproof and playa-proof everything so it's far less likely to break while you're riding around a festival at night.

We'll talk about different light strip options and code for maximizing battery life and brightness, project timelines and management, and how to make sure that all your connections are as bulletproof as possible, so you can ride with confidence.

It's already mid-June at the time of writing, and I already feel a bit behind on this project. I'll post progress pics as I go, and build the guide out as I encounter problems and pitfalls.

Ready to go?
The very first step is to make a list of all the wondrous features you'd like for your bike. I talked to some longtime Burner friends and got a ton of great ideas. I also made a Pinterest board (https://adafru.it/F1r) with some fun ideas I'd like to try.

**Essentials**

- A good headlight for seeing in the dark
- NeoPixels on the frame so you can be seen
- A basket for carrying water and other stuff

**Extras**

- A tall tail-fin "flag" so you can be seen from a distance
- Daytime color -- paint, bling, and fun fur
- Audio sound system / bluetooth speaker

**Pie-in-the-Sky Ideas**

- POV Spoke wheels that show images as you move
- Multiple bikes synched together for group riding
- A "find my bike" feature that makes it blink in the dark from a distance

There are so many ideas to have. One feels almost attacked by the flock of ideas!
The planning stage is a great time to get realistic about what your essentials are, how much time you have, and what can be realistically done with a reasonable battery pack and budget.

For this project, I'm going to start with the essentials, then move to the extras if I've got the time. The pie-in-the-sky ideas probably should have been started September of last year, the day I got home from the last festival of the season, but I'll give some suggestions for after-market solutions you can buy.
I've ordered my LED supplies and enclosures from the Adafruit store! While I wait for the UPS guy, I'll start by decorating the bike.

I want to paint the bike before I add lights, and then add bling and other exciting add-ons after the lights are on.

The first step is to disassemble the bike as much as possible so I can spray paint the frame. I want to paint the basket also, so I left that in place. I started with a couple coats of white primer, so my fancy paint is a lot less likely to flake off.
I chose three anodized paint colors -- a brilliant blue, shiny purple, and metallic green. I want them all intermingling with each other and shining prettily. The label emphasizes that this paint looks best over metallic shiny chrome. so my next layer of paint was shiny silver. I used Mirror Effect paint as well as regular silver paint to cover up the primer on the bike.
Finally, once the silver dried, I sprayed the three colors of metallic paint on, blending them willy-nilly so the bike has a flowing, gradient-style "mermaid" look to it.

This stuff is really gorgeous. The photos don't quite do it justice. Painting over the silver really does give it a depth that I'm quite happy with.

Once the paint dried, I wanted to add some fun fur to the seat. I found a scrap of rainbow fur in my closet and glued it onto the existing seat with a generous amount of E6000. I didn't worry too much about
getting a perfect "fit" -- there was no sewing involved, I just sort of draped the fabric over the top of the seat and then trimmed around the edges, making sure to get plenty of glue all around the sides.

I've got a lot more bling to add, but I want to wait until the electronics are in place before going any further. The UPS guy just knocked on my door, so the next step will be planning out where my lights are going and where to mount enclosures and batteries. It's also time to do some wiring diagrams and figure out exactly how to lay everything out.
Rear Fiber Optics

Materials List

Adafruit Parts Used

- 3w RGB LED (https://adafruit.it/CXi)
- Feather M4 Express (https://adafruit.it/Cmy)
- PropMaker FeatherWing (https://adafruit.it/Fje) with headers
- Waterproof 3xAA battery case with switch (https://adafruit.it/w3B)
- Female JST connector (https://adafruit.it/drM)

Additional Parts Used

- Clear 1" and 3/4" heat shrink
- 1/8" Adhesive heat shrink
- 18" long 3/4" polycarbonate tube
- 5 1/2' long 1/2" polycarbonate tube
- Fiber optics & light pipe in varying sizes, at least 6' long
- Gaffer's tape
- Hot Glue

You'll also need a soldering iron & accessories, some ordinary heat shrink, and a heat gun.
The Vision

I want to mount something tall and striking to the back of the bike so I can be seen from a distance (and for ease of finding my bike among the crowd). This should be:

- Bright and tall -- maybe around 6-7 feet high
- Add motion when I ride
- Maybe create shade? A parasol attachment? Though this might be problematic with wind
- Removable, for bike transport

My first idea was to buy a rear bike rack and figure out how to use that as the mount for my "flag". I bought one from Amazon (yes, okay, I bought the cheap one) and it absolutely didn't fit. The screws were the wrong length and the wrong width, the metal bars weren't nearly long enough, and the stabilizer mount simply wouldn't fit at all. After a frustrating morning, I've scrapped that idea and am trying something a little different.

I went to the plastic store and picked up a couple of acrylic tubes that fit nicely inside each other. My idea is to create a fiber optic system, with bright lights mounted permanently to the bike, with the actual fiber optics removable for transport. This way I'll be able to add interest up high without having to string long wires or use connectors, which could break or allow corrosion into my system.
After prototyping, I realized my mistake: acrylic is the WRONG material. It's cheap and easy to work with but it's extremely brittle. My bike fell gently down against the wall and it cracked and broke. Back to the plastic store for Polycarbonate tubing! It's much sturdier.

For fiber optics, I want something that will shine screamingly bright. We're just using one LED, so I don't need anything that does fancy animations. A 3W RGB LED (https://adafruit.it/Fjf) will work perfectly for this project. 3 watts is a LOT of brightness! (Do not shine into remaining eye...)

I'm going to make the fiber optic "tail feathers" a separate system entirely from the rest of the lights on the bike. 3W LEDs draw a lot of power, so I want to give them their own battery pack and controller. I like the idea of two totally separate systems, so if one system breaks or runs out of juice, I've still got another light system on my bike, and all is not lost.

My idea is to mount the 3W LED and microcontroller to the bottom of the outer 3/4" tube, then mount the whole assembly to the bike. The smaller tube will contain the fiber optics, and will simply slip into the larger tube when I'm ready to ride. This way I will be able to very easily disconnect the large tangly delicate fibers and put them INSIDE my car while the bike rides on the bike rack on the OUTSIDE of my car.
The RGB LED has 4 pads -- R, G, B, and +. The PropMaker Wing has matching pads in the center of the board.

The PropMaker wing will stack on top of the Feather M4 Express, and the battery pack plugs into the Feather.

**Important Note about Power!!**

Do NOT leave the battery pack plugged in to the Feather while you also have a USB cord plugged in. The Feather has onboard charging, so it will try to charge your AA batteries, which could result in battery damage and / or Feather damage.
Software

Install CircuitPython

The Adafruit Feather M4 ships with CircuitPython but lets go ahead and update it to the latest version. It's super easy with the circuitpython.org website, just click the link below to launch the page. There you can choose to install stable release or beta.

Quick Start

- Connect board to computer via a known good USB and double press the reset button.
- Download the CircuitPython UF2 and upload to the FEATHERBOOT drive.
- Open CIRCUITPY drive and upload the required libraries (listed below) and code.py

[Links to CircuitPython website]

Adafruit Circuit Python Libraries

Download the CircuitPython library bundle and unzip the folder. Create a new folder in the CIRCUITPY drive and name it "lib". The following libraries are required to run the code properly. Double check to ensure all of the files and folders are inside the lib folder on the CIRCUITPY drive.

- adafruit_bus_device (directory)
- adafruit_lis3dh.mpy
- neopixel.mpy
- adafruit_rgbled.py
- simpleio.mpy

[Links to CircuitPython library bundle]

Upload code.py

Click the link below to download the project code. Save it as code.py on your CIRCUITPY drive.
import time
import board
import adafruit_rgbled
import digitalio

POWER_PIN = board.D10
enable = digitalio.DigitalInOut(POWER_PIN)
enable.direction = digitalio.Direction.OUTPUT
enable.value = True
# Pin the Red LED is connected to
RED_LED = board.D11
# Pin the Green LED is connected to
GREEN_LED = board.D12
# Pin the Blue LED is connected to
BLUE_LED = board.D13

# Create the RGB LED object
led = adafruit_rgbled.RGBLED(RED_LED, GREEN_LED, BLUE_LED)

# Optionally, you can also create the RGB LED object with inverted PWM
# led = adafruit_rgbled.RGBLED(RED_LED, GREEN_LED, BLUE_LED, invert_pwm=True)

def wheel(pos):
    # Input a value 0 to 255 to get a color value.
    # The colours are a transition r - g - b - back to r.
    if pos < 0 or pos > 255:
        return 0, 0, 0
    if pos < 85:
        return int(255 - pos * 3), int(pos * 3), 0
    if pos < 170:
        pos -= 85
        return 0, int(255 - pos * 3), int(pos * 3)
    pos -= 170
    return int(pos * 3), 0, int(255 - (pos * 3))

def rainbow_cycle(wait):
    for i in range(255):
        i = (i + 1) % 256
        led.color = wheel(i)
        time.sleep(wait)

while True:
    # rainbow cycle the RGB LED
    rainbow_cycle(0.1)

Troubleshooting

If it's not working, make double-sure you have all the above libraries installed in your "lib" folder. Also be sure all your wire connections are tight and the wires are connected up correctly.
More about CircuitPython can be found here! (https://adafru.it/cpy-welcome)

**Important Note about Power!!**

Do NOT leave the battery pack plugged in to the Feather while you also have a USB cord plugged in. The Feather has onboard charging, so it will try to charge your AA batteries, which could result in battery damage and / or Feather damage.
**Electronics Assembly**

**Wiring**

Solder four colored wires to your 3W LED. These work differently than NeoPixels -- you have one wire for red, one for green, and one for blue, and a power wire as well. The aluminum pads on this LED can be a little tricky to solder to. I found it easiest to tin the wire heavily, then heat the aluminum pad on the LED and press the tinned wire down until it stays.

Be sure you get these wires connected solidly. We'll seal the LED later, but you don't want wires popping off when you bend them.

If needed, solder the included headers onto your PropMaker Wing. Solder the other end of the wires from the LED to the pads marked R, G, B and + near the center of the board. I soldered these on from underneath the PropMaker Wing, so the wires and delicate connections will be sandwiched between the PropMaker Wing and the Feather when everything is assembled.

Next, stack the PropMaker Wing on top of the Feather M4 Express and solder in place. You can use female headers or just solder the PropMaker's headers directly to the Feather.

**Playa-Proofing Your Electronics**

We're going to use a couple different varieties of heat shrink to encase our electronics in plastic so they're 100% waterproof and dust-proof.

First up: **Adhesive Heat Shrink**

If you haven't used this stuff, well, it's great. Adhesive heat shrink already comes with hot-melt glue inside, so when you heat it up to shrink it down, the glue melts and makes a super strong, waterproof connection. This stuff is really useful for playa-proofing your electronics.

Shop for [Adhesive Heat Shrink on Amazon](https://adafru.it/Fjg)
Next up: **Clear 1" Heat Shrink**

Heat shrink comes in GIANT size! A clear piece of 1" or 3/4" heat shrink sealed with hot glue at the ends will make a slim, cheap, waterproof enclosure for your electronics.

Once it's sealed, it's not easy to update code or fix loose connections, so be 100% sure everything is working the way you want before sealing. But, it is not too hard to remove the plastic for updates or fixes if needed. 99% alcohol will remove hot glue easily and completely, so if you need to get in there you can, without breaking your setup.

Shop for [extra-large clear heat shrink on Amazon](https://adafruit.it/Fjh)

**Battery Case**

Our waterproof battery case doesn't come with a JST connector, so you'll need to solder one on. This connection will be exposed, so we'll double-heat shrink it to make sure it's bulletproof.

Heat-shrink both wires individually, then add a piece of adhesive heat shrink over the top of both connections.
Upload your code if you haven't already, and plug your battery into the Feather. The 3W LED should come on and shine super bright, running a rainbow animation. Wiggle the wires around and make sure nothing comes loose!

**Important Note about Power!!**

Do NOT leave the battery pack plugged in to the Feather while you also have a USB cord plugged in. The Feather has onboard charging, so it will try to charge your AA batteries, which could result in battery damage and / or Feather damage.

Slip a piece of 3/4" clear heat shrink over the LED, with all wires neatly coming out one side. Squirt a small amount of hot glue inside each end (you need less than you think) and while the glue is wet, use a heat gun to shrink the heat shrink down. Let it dry and trim the edges if needed.

Once everything is completely encased and cooled, give it another test to make sure it's still working reliably. If you need to bust into this thing to repair a loose wire, or if you want to update the code, 99% alcohol will remove the hot glue. Just cut the heat shrink off then soak in alcohol and the glue will peel right off.
The Feather and PropWing are a little trickier to encase, simply because we don’t want the reset button getting pressed accidentally by the tight plastic. I carefully added a little hot glue around the edges of the reset button, making sure not to get any IN the button mechanism. This will make a little shelf the case can rest on, so the feather doesn’t get accidentally reset. But the plastic is soft enough that I can still press the button if I push hard, and reset the board.
With the battery plugged in, and when you’re 100% sure everything works, slide a piece of 1” heat shrink over the Feather and PropMaker Wing. Squirt a little glue in each end and shrink it down. Trim the ends if needed.
Your setup is now fully waterproof! I threw mine in the pool just to test it, and it's still blindingly bright at noontime in full sun! This will drive our fiber optics really well.

The 3W LED will get hot, but not hot enough to melt the glue. I imagine that you could break this if you really tried, but it would take some doing.

Note: this does make a waterproof light, but the battery case doesn't seem to be officially rated for underwater use, so .. I wouldn't put it in water deeper than a couple feet. "Waterproof" and "submersible" are not really the same thing. A battery case with a rating of IP68 would be certified submersible, but there's no official rating on this one, so go carefully.

More about IP Ratings Here (https://adafruit.it/Fji)
Mount to the Bike

Next we'll make the fiber optic "flag".

I used a variety of different sizes of side-glow and end-glow light pipe and fiber optic cable for my bike flag. You'll want long pieces that will dangle out of the top of your smaller 1/2" tube.

Wiedamark (https://adafruit.it/Fji) is a great source for fibers of varying sizes and lengths.

Even them up at the end and cut them so they're all lined up together. I added a couple rubber bands around the fibers so they fit tightly inside the tube without sliding around.

The LED will be attached to the larger outside tube which will be permanently attached to the bike. The smaller tube containing the fiber optics slides into the larger tube, and slides back out again for transport.
Use gaffer's tape or duct tape to secure the LED to the back of the microcontroller. Then, securely tape the larger tube over and around the LED.
I drilled a couple holes in my bike frame and attached two screw clamps from the hardware store to hold the tube in place at the angle I wanted. Then I zip-tied the battery pack to the frame with the on/off switch facing outwards. Make sure everything is really secure!
Slide the fiber optic tube into the outer tube and push it down all the way until the fibers catch the light from the LED and glow!
The cool thing about this setup is that you actually get some variation in colors. The 3W LED actually has three LEDs inside: a red, green, and blue. Some of the fibers will be positioned directly above one of these lights, and some will be more generally placed, so you get a lovely color variation in your fibers.
I finished up by adding a few crystal gems (https://adafruit.it/Fjk) and other diffusers to a few of the strands of light pipe. Let your flag express your style.
Frame Lights

I want to add lights to the frame that will animate in the same colors I painted the bike. I'm using NeoPixel strand lights at 2" pitch, for a couple of reasons:

1. They are already 100% waterproof -- each light is encased in plastic, and these lights are very hard to break
2. The strand-light style will allow me to wrap them around the bike frame, decorating it like a Christmas tree. I can get lights on all sides of the bike really easily.
3. The spread between the pixels means I can cover the whole bike using fewer lights than if I were to use a NeoPixel strip. This means my batteries will last a lot longer, since I'm driving fewer pixels. My hope is that this strip will last through several nights of use without having to change the batteries.

Your browser does not support the video tag.

Adafruit GEMMA M0 - Miniature wearable electronic platform
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...
$9.95
In Stock
Add to Cart

Adafruit NeoPixel LED Dots Strand - 20 LEDs at 2" Pitch
Attaching NeoPixel strips to your costume can be a struggle as the flexible PCBs can crack when bent too much. So how to add little dots of color? Use these stranded NeoPixel dots!...
$27.50
In Stock
Waterproof 3xAA Battery Holder with On/Off Switch

Keep your power source safe and toasty in these waterproof 3xAA battery holders. They're just like classic switched battery holders, but designed for survival! The case has a rubber...

$3.95  
In Stock  
Add to Cart

If soldering is not your jam, never fear: you can get this bolt-on kit for the Gemma M0 for attaching the lights and connectors to the board. Just wind the wires around through the Gemma's soldering holes and tighten the screws down to hold the wires in place. Dreamy!

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...

$1.50  
In Stock  
Add to Cart

In addition, you'll need:

- Large clear heat shrink in 1" and 1/2" sizes
- Wire strippers & wire cutters
- Hot glue gun
- Heat gun
- Small screwdriver
- Zip ties

Shop for extra-large clear heat shrink on Amazon (https://adafru.it/Fjh)
Wiring the Lights

We'll wire the battery pack directly to the Gemma, since the waterproof pack doesn't come with a JST connector attached. Red goes to \texttt{VOUT} and black goes to \texttt{G}.

The NeoPixels will use the same pads for + (\texttt{VOUT}) and - (\texttt{G}), and the third wire will attach to \texttt{D1}.
Code with MakeCode

MakeCode is Microsoft's ritzy drag-and-drop code editor. It makes it easy to get up and running with the Gemma M0.

Find the Gemma M0 code editor at [maker.makecode.com](https://adafru.it/DWO)

Here's my completed project, which runs a green, blue and purple gradient along the LED strand. Click the Download link at the bottom, plug your Gemma M0 into your computer, click or double-click the reset button and drag the file onto the resulting GEMMABOOT drive.

Or, follow along below to create your own project from scratch.

Code It Yourself

Head to [https://maker.makecode.com](https://adafru.it/C9N) and select **New Project**. Choose the Gemma M0 board. Give your project a name. I called mine **Bike**.

Click the **LIGHT** tab. Another tab titled **MORE...** will appear. Click on that one to find lots of options for attaching and controlling a NeoPixel strand soldered to the Gemma M0. Drag an instance of **set strip to create WS2812 strip on LED with 30 pixels** out into your workspace and place it inside the **on start** loop, which is usually there by default in a new project. (If it's not there, you can find it under the **LOOPS** tab). Change the pin to **D1** instead of **LED**, and enter 100 pixels, or however many you've got on the bike.

Note: NeoPixels are also called WS2812 pixels, and DotStars are called APA102 -- make sure you've got the right ones selected.

Next, drag an instance of **strip set brightness** into your **on start** loop. Set the brightness to whatever you'd like. I want them as bright as possible, so I chose 255.
Above the **LIGHT** tab there’s another tab called **PIXEL**. Anything in here controls the onboard NeoPixel on the face of the Gemma M0. I want this turned off on my bike - - no stray status lights for me, thank you -- so I dragged an instance of **set pixel color** into my **on start** loop and made it **black**. This will make sure the pixel stays off.

I want to be able to animate and change the colors of my NeoPixels over time. The way to do this is to assign the color a variable, and then change the variable this way or that, causing the lights to animate. Go to the **VARIABLES** tab and create a new variable called **hue**. Then, drag an instance of **set hue to 0** into your **on start** loop and pick a starting hue. I want a purple-pink-blue gradient, so I chose **140** as a starting hue which will give me a nice blue color.

Go back to the **LIGHT > MORE...** tab and drag an instance of **set strip pixel color at 0 to red** into your **forever** loop. Things in this loop will run forever, over and over. Then drag an instance of **strip rotate by 1** beneath it. Change the 1 to a 3. This snippet will create the animation on the strip, as soon as we tell it what to do.
Hue, Saturation and Value

Hue, saturation and value are a means of defining a specific color using just numbers, on a scale of 0-255.

Hue refers to the color on the spectrum -- red, blue, purple, etc. Red is at 0, blue is at around 140, purple comes in at around 180, and the spectrum loops around back to red at 255.

Saturation refers to how pastel or how vivid the colors are: 0 is a plain white, no matter what your hue is -- any completely unsaturated color just turns white. 255 is fully vivid and saturated with color.

Value, for our purposes, refers to the brightness or intensity of the color. The higher the value, the more red or blue is added.

You can read far more than you ever wanted to know about this on Wikipedia (https://adafruit.it/CxV). What matters to us is that we can mess with the NeoPixel colors in all sorts of fun ways using hue, saturation and value.

Under the LIGHT tab you'll find a block for hue 255 saturation 255 value 255. Drag this into your set strip pixel color block, replacing the red. Then, go into your VARIABLES tab and grab an instance of your new favorite variable, hue. Use this variable to replace the 255 in the hue 255 block. Now we can mess with the pixel hue by changing the hue variable.

So, let's do exactly that. Grab an instance of change hue by 1 from your VARIABLES tab and put it into your forever loop. I made mine a little more dramatic by changing the hue by 2. Try some different values in here to see what you like.

Testing It Out
You may have noticed that over on the left, a little NeoPixel emulator strip has appeared beneath your Gemma M0. This emulator will run whatever your code is doing. This is a really easy way to preview your animations and decide if you like the effect, without having to download the code every time. My emulator is starting to look good, but it's running a bit faster than I'd like. Let's slow it down a bit.

Under the LOOPS tab, find \texttt{pause 100 ms} and drag it into your \texttt{forever} loop. Check the emulator again. Do you like it better? Find a speed that works for you. You can also randomize the timing (\texttt{pick random} under the \texttt{MATH} tab) if you'd like.

Almost there! We've created a customizable rainbow animation on our strip. However, I don't want a full rainbow -- I just want a partial rainbow, or a gradient animation. We can limit our \texttt{hue} variable to only show hues in one part of the spectrum using an \texttt{if / then} statement.

We'll start with a hue of \texttt{80} (green), then increase by \texttt{2} until we reach \texttt{240} (magenta), at which time we will place a limit and tell the code to circle back around to \texttt{80}.

Under the LOGIC tab, find an \texttt{if true then} block and drag it into your \texttt{forever} loop. Grab a \texttt{comparison} block to replace the \texttt{true}, then change the \texttt{=} to \texttt{>=}. Now we don't have to worry about hitting our limit number exactly.
From the VARIABLES tab, grab another hue variable and use it in the first comparison spot. Set the second comparison spot to 240. Drag an instance of `set hue to 0` and place it inside the if / then block, and set the value to 80 for a nice green.

Now the hue number will count up from 80 (since you set it to 80 in the setup block) by 2’s, until it reaches 240, at which point it will pop back down to 80. This effectively makes the NeoPixels change from green (80) to magenta (240) and then back again. Success! We’ve got our gradient.

It's easy to change these values to get the color gradient you want.

Plug your Gemma M0 into your computer using a USB cable. Click the tiny Reset button in the middle of the board and the onboard pixel will turn green. When the lights turn green, a new drive will appear on your computer called GEMMABOOT.

In MakeCode, click the Download button to download your code to a file. Drag this file onto the GEMMABOOT drive. That's it! Plug in your light strand and make sure you love the animation. If not, go back and tweak it until you love it. Our next step will be to seal up the Gemma so you won't be able to easily change the code once you've done this.
Electronics Assembly

Cut the female connector off the **OUT** end of your last NeoPixel strand. We'll attach this to the Gemma. Make sure you are cutting the correct connector off! NeoPixels only work in one direction -- you must attach the data wire to the **IN** end. On most strips, the male connector is on the **IN** end and the female is on the **OUT**, but double check your strips to be sure. There is a teeny tiny, very hard-to-see arrow on the back of the strips pointing in the direction of data flow.

Strip the wires about 1/4" with a pair of wire strippers. Twist the wires so they're tidy, then twist the red striped wire securely through the hole marked **VOUT**. Thread the middle wire through the **D1** hole, and the third wire will twist to **G**.

Solder or bolt on the MIDDLE wire only (for the moment).

Next, find your battery pack and strip the wires about 1/4". Thread the red wire into **VOUT** and the black wire into **G**, along with the connector wires.

Finish soldering or bolting the wires to the Gemma. Test by pressing the power switch on the battery pack -- the lights on the face of the Gemma should come on.
Trim any extra fuzzy wires.

If you've already uploaded your code, plug your NeoPixels into the connector and be sure they come on. If they're not coming on, check to be sure the tiny power switch on the face of the Gemma is switched to ON.

That's it for assembly of the electronics. We'll seal up the Gemma with heat shrink and hot glue a bit later, but the next step is to add the lights to the bike and run the animations in place, so you can be 100% sure you like them. A NeoPixel animation can look pretty different once the LEDs are wound around the bike frame, and once the Gemma is sealed we won't be able to update the code. Since we used a connector, we can add lights to the bike and test, then easily remove the Gemma and battery pack to seal it up.
Mount and Seal

Start by deciding where the battery pack and microcontroller will be attached. Mine will fit beautifully right underneath the bike seat, nestled up inside and out of the way. From there, wrap the lights tightly around the bike. You have a lot of control over density -- you can cover most of the bike with just one strand, or snap together 3-4 strands for NeoPixel Overload.

Be sure to avoid squishing any brake or gear cables -- go underneath instead of over so as not to interfere with the mechanics of the bike.

Choose a few spots to zip tie the lights in place on the bike so they don't slide around. I found just one zip tie hidden at each end of the frame sections was plenty secure.

Turn the lights on and step back a few feet. Do you like the way the animations look wrapped around the frame? If not, now is the time to update the code until it's exactly how you want it to be. Next we'll seal up the microcontroller and connectors, so be very sure everything's working how you want!
Sealing it Up

Find all the connectors between NeoPixel strands on the bike. Unplug them and slide a piece of clear 1/2" heat shrink on, then plug the connector back in and cover it with the heat shrink. You can use the same hot-glue trick here, or just use the heat shrink alone. This is really to keep dust from getting in the crevices, so you probably don't need the glue (unless you're planning on submersing the bike in water).

Slide a piece of 1" heat shrink over the Gemma. Line both ends with a bead of hot glue -- you don't need a lot, just get it from end to end -- and then use a heat gun to shrink it down while the glue is still wet. This will make a plastic seal around the Gemma that will be completely dust-proof.. but, you'll no longer be able to update the programming once this is done, so be sure you're happy!
Finally, use some hot glue to secure the sealed Gemma to the back of the battery case. Now you can mount it to the bike -- but remember to mount it using zip ties or a bungee cord, or something you can remove fairly easily so you can access the battery pack when you need to change the batteries. I chose to leave this one connector uncovered with heat shrink, since that will make it much easier to get the battery case out. This case is a bit of a pain to open so I want to be able to take it off the bike entirely.
Repair Kit

Even with all the sealing and robustifying you've done, it's possible that things will break, usually at the worst possible moment. Here's what's in my repair kit. It fits nicely inside an Adafruit box, and makes me the Hero whenever someone's lights aren't working on-site.

I've got a butane powered soldering iron which works without being plugged in. This thing is great for onsite repairs, or repairs for a bigger project that can't be moved to a power source. I also have a bit of solder, solder wick, and a cleaner sponge.

Pack extra heat shrink and zip ties -- these don't take up much room but will save you for sure. For this design I'll pack way more zip ties since I need to cut them to change the batteries.

Extra batteries are a great thing to put in your repair kit as well. And be sure you have a screwdriver that will open your battery case. I love this fancy versatile one - it's like 8 screwdrivers in one.

The Gemma M0 is so inexpensive, why not bring an entire second Gemma-and-battery-pack assembly? That way if something goes wrong you can just swap out the whole thing and not need to worry about fiddly repairs. Same with the LED strips -- these are easy to change out and plug in. You can re-heat shrink using your butane powered soldering iron or a lighter as your heat source. And if you don't use them this year, you will be glad you have them on hand for the next project!

A thread ripper from the craft / sewing store is more useful than you can imagine. Everyone should have one of these in their repair kit. It's a small razor blade that can slice through seams, heat shrink, mylar, or a variety of other materials without any danger of cutting your fingers or going through wires -- so, perfect for repairs that need to happen after your second beer.

I always carry safety pins as well. A glue gun with extra sticks is really helpful too, though you'll need to
find a place to plug it in as I've never seen a butane powered one.

Those little battery case screws are fiddly and small, and will get lost. Bring an extra battery case and some extra screws. Also throw in a roll of gaffer’s tape (not shown) for when the zip ties won't cut it.

**Go Further**

In case this isn't enough, here are a few links to commercially available products that are playa-tested and approved!

- [Monkey Lights](https://adafruit.it/FmV): POV light kit for bike spoke wheels
- [Fun fur](https://adafruit.it/FmW) in lots of weights and colors
- [Kick stand](https://adafruit.it/FmX) so you don't have to lay it in the dirt
- [Bike baskets](https://adafruit.it/FmY) in various sizes and shapes