Chinese Dragon Puppet with Motion-Reactive Flame Effect

Created by Erin St Blaine


Last updated on 2023-08-29 04:54:24 PM EDT
Overview

I know how the birds fly, how the fishes swim, how animals run. But there is the Dragon. I cannot tell how it mounts on the winds through the clouds and flies through heaven. Today I have seen the Dragon.

-Confucius

Chinese dragons are iconic symbols of Chinese culture and festivals, with their vibrant colors and intricate designs. This tutorial will show you how to turn an inexpensive Chinese dragon decoration into an animated stick puppet, and then how to take your dragon puppet to the next level, adding motion-reactive flame lights that create a stunning visual effect to mesmerize your audience.

This is an easy beginner-level project that involves a little bit of soldering. You can download our code sample, or learn to make your own flame effects using MakeCode, Microsoft's drag-and-drop code editor. This is a fantastic project to do with kids.
Parts

Circuit Playground Express
Circuit Playground Express is the next step towards a perfect introduction to electronics and programming. We’ve taken the original Circuit Playground Classic and...
https://www.adafruit.com/product/3333

USB cable - USB A to Micro-B
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

Adafruit NeoPixel Digital RGB LED Strip - White 60 LED
You thought it couldn't get better than our world-famous 32-LED-per-meter Digital LED strip but we will prove you wrong! You wanted...
https://www.adafruit.com/product/1138

1 x On/Off Switch
JST 2-pin Extension Cable with On/Off Switch - JST PH2
https://www.adafruit.com/product/3064

1 x Battery
Lithium Ion Polymer Battery - 3.7v 500mAh
https://www.adafruit.com/product/1578
1 x **Battery Charger**
Switched JST-PH 2-Pin SMT Right Angle Breakout Board

1 x **Silicone Wire Cable**
Silicone Cover Stranded-Core Ribbon Cable - 4 Wires 1 Meter Long - 26AWG Black

1 x **Heat Shrink**
Food-Grade Heat Shrink - 3/8" diameter 12" long

Additional Tools & Materials

- Chinese Dragon Decoration: [I used this one from Amazon](https://www.amazon.com/).
- Soldering iron & accessories
- Hot Glue gun
- Zip ties
- Two wooden dowels for control rods
- A handful of fiberfill pillow stuffing for diffusion

Wiring Diagram

The following connections are made between Circuit Playground Express and the NeoPixel strip:

<table>
<thead>
<tr>
<th>Circuit Playground Express</th>
<th>NeoPixel Strip</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOUT</td>
<td>+5V</td>
</tr>
<tr>
<td>A1</td>
<td>DI or DIN</td>
</tr>
</tbody>
</table>
Remember: NeoPixel strips are directional. Be sure you're soldering to the IN end of the strip. Check the arrows printed on the PCB, they should be pointing in the direction of data flow.

The on/off switch plugs into the JST connector, and the battery plugs into the other end of the switch wire.

## Electronics Assembly

Cut a length of NeoPixels to fit inside your dragon. He squishes and stretches, but I found that 42 pixels of 60 pixels per meter NeoPixels was about the right length for my dragon. Note down how many pixels you have: you'll need this number later when you're coding.

Cut about 8 inches of ribbon cable wire.

This ribbon cable has 4 wires, and we only need 3. One of the wires has a stripe on it. Let's keep that one, to help us keep track of which wire is which, and get rid of the fourth wire on the opposite edge. Pull that one off and save it for another project.

Separate the remaining wires a bit and strip about 1/4" of shielding from each wire. Solder to the Circuit Playground: striped wire to VOUT, middle wire to A1 and the remaining wire to GND.

For more detailed soldering instructions, check out our [How to Solder NeoPixels Guide](https://docs.adafruit.com/).
Solder the other end of the wires to your NeoPixel strip. The striped wire goes to +, the middle wire to DI, and the third wire to G.

Be sure you're soldering to the IN end of the strip. You can see little arrows pointing in the direction of data flow if you look closely.

Testing

Our next step will to seal the strip ends so the wires won't pull out with all that shaking going on. But before you do that, it's important to test your strip and make sure your solder joints are good. It's much easier to fix a faulty connection before you've sealed it in glue.

Head over to the code section and upload the sample code. Give your Circuit Playground a shake and make sure the lights come on and stay on. If everything works, move on to the next step. If not, check below for some troubleshooting tips.

Sealing

Slide a piece of clear heat shrink over the strip and squirt a little hot glue inside, so the solder joints are covered in glue. While the glue is wet, use a heat gun to shrink the heat shrink down so it completely seals the end of the strip.

Do the same with the other end.

Code with MakeCode

We'll use Microsoft's drag-and-drop code editor to create the animations. 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.

For this project we'll be using maker.makecode.com, which is the beta version of MakeCode. At the time of writing, it has a few more features that will give us more control over our pixels, but it does change frequently so if something's not where you expect, look around.

Once your code is written, you'll simply plug your Circuit Playground Express into your computer via its USB port and click the "reset" button. Be sure the USB cable is a known good "data + power" cable and not a small charge only cable. All the lights will turn green and your Circuit Playground will appear as a drive on your computer called CPLAYBOOT. Drag your downloaded code onto this drive to program the Circuit Playground Express. Easy!

If you get a flash drive named CIRCUITPY instead, no worries, press the reset button twice and CPLAYBOOT should show up. It might take one or two tries but it does work.

Head over to this Intro to MakeCode 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.

Flame on Shake Code

Code Design & User Interface

For this project, I want to design a flickering flame animation and trigger it when I shake the puppet's head. After a few seconds of flickering, I want the lights to slowly fade to black until they're triggered again with another shake.

Head to maker.makecode.com, choose New Project and select the Circuit Playground Express. 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 emulator pictured on the left.
On Start & Functions

Click on the LOOPS tab and drag an instance of on start onto the workspace. (It may already be there) Anything inside this bracket will happen just once, when the Circuit Playground is powered up. You'll also see a FOREVER loop. Anything in this bracket will happen over and over, forever.

First we'll create our variables. Click the VARIABLES tab and add a variable called strip, and one called bright.
Set Up the Lights

Click on the LIGHT tab. Any block inside this tab will control the lights on the face of the Circuit Playground. We don't need to do anything to get these lights set up, they're ready to go.

Click on the the MORE tab. Blocks in this tab will control the NeoPixel strip we added. We need to tell MakeCode we added lights, so drag an instance of `set strip to create WS2812 on LED with 30 pixels` into your `on start` loop.

I soldered my NeoPixel strip to pin A1, and my strip has 42 pixels in it, so I changed this line so it reads `set strip to create WS2812 strip on A1 with 42 pixels`.

Brightness Setup

I want to be able to change the brightness slowly over time, so I've made brightness into a variable. This way we can use simple math to fade the lights up or down.

In the `on start` block, set your global brightness. I've set mine to 150, which is medium-bright (your max here is 255).

Then in the `forever` loop, I've set up the brightness of both the onboard lights (found in the LIGHT tab) and the NeoPixel strip (found in the MORE tab) so they use the `bright` variable.
Flame Animation

MakeCode comes ready with a few premade LED animations, one of which is a flame-type effect. I like this well enough on the Circuit Playground face, but I don't think it looks quite as good on the LED strip; I want a little more control over the flicker for the body of the dragon. No worries, we can use both.

Find the FUNCTION tab under ADVANCED. Create a new function and call it `flame`. Then, add `FUNCTION --> call flame` to the `forever` loop. We'll move it out of there later, but while we're building the animation it's nice to have it just run.

Now you can mess around with the different light controls available until you have something that looks like a flame. Here's what I ended up with.

The first two lines deal with the pixels on the Circuit Playground's face. I gave a background color of orange, then added the pre-made flame animation over the top.

The rest of the lines deal with the NeoPixel strip. I gave it a background gradient from orange to red, then used the photon function, which sends a colored pixel shooting up or down the strip. I used a black pixel, which turns the lights off (black) for just a moment, and played around with adding more photons until I had a flickering effect that I liked. Adding the random numbers (under the MATH tab) gives it a really nice randomized flicker effect that looks organic.
Testing

Test your code by clicking the Download button. You may get the option to pair your board to your computer -- this is helpful as you're testing because then download becomes just one step. If you don’t want to pair the board, just drag the downloaded file onto the MAKECODE or CIRCUITPY drive that appears when you plug in your board and press reset.

Motion Trigger and Fade

Motion Trigger

I want this animation to happen when I shake the dragon's head. For this, we’ll use the `on shake` loop from the INPUT tab.

Take a look in there and notice that there are lots of other input triggers available. You can trigger your animation with a tilt, a loud sound, a button press, or a change in ambient light, among other things. The Circuit Playground Express has a wealth of sensors and inputs to choose from to bring your puppet to life.

In the `on shake` loop, we start by setting our brightness variable to our max, which will turn the LEDs on. Then we’ll call our `flame` function. I want my flames to last a few seconds, so I grabbed a `repeat` block from the LOOPS tab. More repeats will make the flames last longer. Experiment!
Fade LEDs Out

The second repeat block is our fader. From VARIABLES, choose **change bright by 0** and set the value to a negative number, to make it count down. A value of -1 will make the lights fade slowly, and a larger negative number will make the LEDs fade more quickly.

Each time this block runs in my code, the strip will dim by -2. I initially set it to 150, so running this 75 times will dim it to zero. If you want to change the dim rate, you need to also change the number of repeats so the math works out:

- Brightness 150, dim by 1: 150 repeats until we hit zero
- Brightness 150, dim by 2: 75 repeats until we hit zero
- Brightness 150, dim by 3: 50 repeats until we hit zero

![Fader Block Diagram]

Build the Puppet

I started with this [Chinese Dragon decoration](https://www.amazon.com) that I found on Amazon. It's a good medium-sized dragon, perfect for puppeting by one person.

For the control sticks, I'm using repurposed venetian blind control sticks. They're about 3 feet long with a hole in one end. Perfect! You could also use wooden dowels from the hardware store with a hole drilled in one end.
I want to attach these very securely to the head and the tail. I used zip ties: one at the top and one a few inches down. This gives me really good control of the head. I can twist or shake the stick around and the dragon responds beautifully.
Find or make a couple holes in the tail end and attach a second control rod in the same manner.

My dragon has a string running the length of the top, going through a grommet in the center of the body section. If I pull this string, it lifts the middle of the dragon up.

I ran this string through the hole in my rear control stick so I can control how much he arches his back when I pull on it. The zip tie keeps the string from pulling out but I left it loose enough that I can adjust the string’s tension with some gentle pulling.

This dragon was made to be a stationary decoration, not an animated puppet. I used more zip ties and clear packing tape to reinforce the snap-together pieces and hold them permanently together, so my dragon's head doesn't fall off when I shake him too hard.

I diffused my Circuit Playground's onboard lights with a little handful of pillow stuffing. Run some hot glue around the edges of your Circuit Playground, being sure not to get glue near the delicate components or into the reset button or USB port. Stick the fiberfill down onto the glue.
I bent the tongue piece in my dragon's mouth and attached the Circuit Playground to the front with a piece of industrial velcro. I threaded the wires through the back of the head, and slipped the NeoPixel strip down through the body.

The rest is wire management. I coiled the switch wire around my front control stick until it was held firmly in place, and attached the battery to the opposite side of the head with more industrial sticky-back velcro. This makes it easy to remove the battery for charging.

Shake it all around and make sure everything is tight. Use more zip ties, packing tape, and glue as needed. You don't want your electronics shaking loose in the middle of your parade.