Dance-Reactive Tutu Sparkle Skirt

Created by Erin St Blaine

https://learn.adafruit.com/dance-reactive-tutu-sparkle-skirt

Last updated on 2023-08-29 03:58:30 PM EDT
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>Wiring Diagram</td>
<td>5</td>
</tr>
<tr>
<td>Code with MakeCode</td>
<td>6</td>
</tr>
<tr>
<td>• Photon Rainbow</td>
<td></td>
</tr>
<tr>
<td>• Troubleshooting</td>
<td></td>
</tr>
<tr>
<td>Electronics Assembly</td>
<td>15</td>
</tr>
<tr>
<td>Build the Tutu</td>
<td>17</td>
</tr>
</tbody>
</table>
Overview

You can dance if you want to. You can leave the world behind.

Make a tutu that reacts to your dancing. This is a great project to do with kids, or to make for all your fabulous dancer friends. (Because.. if your friends don't dance.. well, they're no friends of mine.)

The magic lies in Microsoft MakeCode's drag-and-drop code editor in combination with Adafruit's Circuit Playground Express and its onboard sensors. Play with triggering animations when you jump, cartwheel, tilt left or right, or shake and shimmy. This is really way too much fun.

Materials Needed

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
Adafruit NeoPixel LED Dots Strand - 20 LED 4” 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!
https://www.adafruit.com/product/3631

3 x AAA Battery Holder with On/Off Switch and 2-Pin JST
This battery holder connects 3 AAA batteries together in series for powering all kinds of projects. We spec'd these out because the box is slim, and 3 AAA's add up to about...
https://www.adafruit.com/product/727

You Will Also Need

- A tutu. I used this one from Amazon ( ) -- it has 3 layers of tulle and one layer of satin
- Elastic or ribbon for ties
- A needle and thread and/or sewing machine
- A soldering iron ( ) & solder ( )
- Wire cutters ( ) and wire strippers ( )

If you’re new to making things light up, you can get all the electronics tools you need in one convenient kit! This kit also comes with a stand for your soldering iron, and an "eraser" -- solder sucker that makes it easy to fix any mistakes you make with wiring.
Wiring Diagram

The LED strand has 3 wires. The red-striped wire goes to the Circuit Playground's Vout pin, the middle wire goes to A1, and the third wire goes to G.

Be sure you are wiring to the IN end of the strip. LED strips have an IN and OUT end and they won't work if you wire them the wrong way. If you look closely at the back of the pixels you can make out an arrow pointing in the direction of data flow (from in to out). This is really hard to see, so it's usually reliable to start from the end of the strip that has the male connector. If you find your project isn't working, you may have a strip that's manufactured differently -- try starting at the other end.
Code with MakeCode

I've built some code that triggers a color-changing Photon animation when I jump or move quickly. This is fun to work with and customize to suit your movements, and you can get some great results with a little trial-and-error. Here's a link to the finished project if you want to skip ahead and work backwards.

Tutu Code on MakeCode.com

To follow along and learn how I did it, point your browser at www.makecode.com ()

Select the Circuit Playground board, since that's the one we're working with today. While you're here, take a second to notice all the other development boards MakeCode will work with. So many possibilities! Select "New Project".
You'll find yourself in the MakeCode editor window. From here, you can drag and drop snippets of code from the different blocks listed in the middle, into the workspace on the right.

We'll start with the `on_start` loop, which will run some code when the Circuit Playground Express first boots up. For now, drag the `forever` loop down and out of the way.

Click on the LIGHT block at the top of the list. The code in this block deals with the Circuit Playground's onboard lights. When you click this tab, a new block called NEOPIXEL appears. Click on this one. The options in this block deals with any light strips you add, so that's what we'll be using today.

First we'll tell our Circuit Playground that we soldered a light strand with 20 lights onto pin A1. Then we'll tell it how bright we want our lights to be. Drag the two code snippets indicated out onto your workspace and place them inside the `on_start` loop.

Change the default 30 to 20 since we have 20 lights. Then set the brightness to whatever you'd like. This tutu is pretty diffused with all the layers of tulle and satin, so I turned my brightness all the way up to 255.

Adding Motion Reactivity

The Circuit Playground has a 3-axis accelerometer on board. This gives us lots of different options for motion reaction. You can find them under the INPUT tab. I used the 3g input for my tutu, but you can play with all the different options and see what works with your movement.
Click the INPUT tab. Drag an instance of `on shake` into your workspace and open up the menu under shake. Choose 3g for now, but take note of all the things you can choose from for customization later on.

We've set up our light strand, set the brightness, and made a motion trigger. Now we need to tell our lights what we want them to do when we move. I decided on a photon of colored light shooting across the tutu whenever I jump.

**Photon Rainbow**

We'll use the Photon functions (also under the NEOPixel tab, just scroll down a bit) to create some custom animations. Photon is a very clever tool that's super hackable. I keep on discovering that I can create new fun effects whenever I play with it.

For this project, I want an animation that dramatically changes the color of the tutu whenever I jump. We'll use a variable to change the color every time the animation runs.
Drag an instance of `strip photon set pen hue` and `strip photon forward by 0` into your `on 3g loop`.

Now is a great time to start testing your code. You can download it as many times as you’d like and see what it's doing in real time on the Circuit Playground Express.
Start by giving the file a name and saving it to MakeCode. I called mine "Tutu". Then click the Download button, and plug your Circuit Playground into your computer via the USB port.

Click the Reset button on the face of the Circuit Playground. All the onboard lights will turn green and a drive will appear on your computer called CPLAYBOOT. This means it's in bootloader mode and you can program it. Drag the file you just downloaded onto that drive.

Move your Circuit Playground quickly -- give it a shake -- and see what happens. We set a hue of 10 (orange) so each time you shake it, an orange light appears. This isn't quite what we want -- we want the whole strip to light up. To accomplish this, we need to add another loop.
Go to the LOOPS tab and select \texttt{repeat 4 times}. Drag this onto your workspace and place it inside your \texttt{on 3g} block with the \texttt{photon} commands inside, as shown. Since we have 20 lights, change the \texttt{4} to \texttt{20}. Now the photon will move forward by 1 light, 20 times in a row -- which means all our lights will get used.

Download again and see what happens now. Looking better! We've got the whole strip lighting up whenever we shake the Circuit Playground. But it's going wicked fast, and not changing colors yet. Let's slow it down and add some color variation.
Go to LOOPS and drag an instance of `pause 100 ms` onto your workspace, inside your `repeat` loop. Change `100` to `20` for now. This will determine the speed of your animation, and you can adjust it to suit your preference. I like a pause of just `20 ms`, it makes for a nice active photon.

Next we'll make a variable. Variables allow us to change a number on-the-fly. We've got `photon hue` set as a number right now (10) and we can change this up or down to change the color of the lights.
Click the VARIABLES tab and make a new variable. Call it `hue`.

Drag an instance of the new `hue` variable into the `strip photon set pen hue` block, to associate our variable with the hue of the photon pen.

Then drag an instance of `change item by 1` onto your workspace. Put it inside the `repeat` loop along with the `pause`. Select your new variable `hue` from the dropdown in `change item by 1` to make it `change hue by 1`. Then change the `1` to a `3` for now. Here's another place to customize -- the higher the number, the more each light will change color. 3 gives a nice color gradient.
Lastly, let's give our variable hue a starting color. Drag an instance of VARIABLES > set item to 0 into your on start loop. Set it to the hue variable. Choose a number between 0 and 255 to pick a starting color.

Troubleshooting

If the CPLAYBOOT drive is not appearing, here are some things to try:

- Be sure you have a Circuit Playground Express. The Classic will not work with MakeCode.
- Use a different USB cable. Some cables are "charge-only" and won't pass data. Also try using a different USB port on your computer (mine is sometimes twitchy).
- With the Circuit Playground plugged into your computer, press the tiny reset button. The lights should all turn green. If they don't, try double-clicking the reset button.

If it's still not working, head over to the Intro to Circuit Playground Express guide for more suggestions.

If you can drag the code onto CPLAYBOOT but your LED strand isn't lighting up, check to be sure that your code reflects the pin you soldered to (Did you solder to A1?)
Does your code reflect that?). Also be sure you've soldered your connector to the IN end of the LED strand -- it won't work if you soldered to the OUT end.

Electronics Assembly

The LED strand comes with a connector at each end. In most strands, the male connector is at the IN end and the female connector is at the OUT end. Check to be sure your strand goes along with this convention. Data needs to come from the IN end.

Since I'm using just one strand for this project, I've cut the connector at the OUT end off and soldered it to my Circuit Playground. This way I can permanently attach the lights to my tutu, but still be able to easily disconnect the Circuit Playground, in case I want to update the code or use the Circuit Playground in another project when I'm not wearing the tutu.

Find the end of the LED strand that has the female connector. This should be the OUT end of the strip. Double check this is right by looking for an arrow on the back of the LEDs pointing in the direction of data flow.

Use a pair of wire cutters to cut this connector off with a few inches of wire still attached.

Separate the three wires on the connector and strip about 1/4" from each one using a pair of wire strippers.
Wrap the bare ends of the wires through the holes on the Circuit Playground as shown, and use a soldering iron to solder them in place. The red wire will go to VOUT, the middle wire goes to A1, and the remaining wire goes to G.

Plug your LiPoly battery or your AAA battery holder into your Circuit Playground and see if the onboard light comes on. If it doesn't, be sure your AAA battery holder on/off switch is also turned on and that your batteries are charged. Also be sure you’re using alkaline batteries and not rechargeables. The Circuit Playground Express will run best on non-rechargeable batteries since their voltage is a bit higher.

Plug in your light strand and shake the Circuit Playground around a bit. Play with holding it in different orientations to get a feel for how the animations you coded are triggered.
I bought a tutu from Amazon.com that has 3 layers of tulle and a satin layer. I think they meant for the satin to go on the inside, but I turned it around so the satin was the top layer for better LED diffusion.

To make the tutu into a tie-on skirt, find the seam in the tulle and cut through all four layers. (You could leave it as a pull-on skirt if you’d like, but this way it’s easier to get in and out of and fits a wide range of sizes)

The edge you just cut is now the center front. I rounded both front edges of only the satin layer to make the tutu lay nicely on my hips, creating a "bustle" type look. Save the pieces you cut off! We’ll use this extra fabric to make the battery pocket. If you’d like, finish the lower edge of the tulle with a serger or make a small hem.
Cut two pieces of white elastic or ribbon to make the ties. Sew one to each side of the tutu at the top, underneath the satin layer.
Grab one of the scraps of satin you cut from the front corner of the tutu. Trace around the Circuit Playground and battery pack to make a pocket. Cut it out along your lines. Finish the edges if desired.

Stitch the sides and bottom to the TOP layer of tulle, just underneath the satin layer, making sure not to catch the other layers when you sew.
With the electronics in the pocket, plug in your light strand. We want to attach the lights on the outside of the bottom (innermost) layer of tulle. To get there, make a small hole in the middle and top layers of tulle so you can slip the connector through and plug it into your Circuit Playground Express.

Sew around the wires between the pixels to attach the pixels artfully to the tulle. I like to try and stagger them a bit, so the lights look more random and it doesn't look so much like a strand.

Download your code and dance around a bit. See if you like it. If not, make some changes. Make it yours!

Remember that the orientation of the Circuit Playground Express inside the pocket will change your results if you use the `tilt left` or `tilt right` functions. Make sure it's sitting in there vertically for best results. If it's slipping around a lot, try making the pocket a little smaller, or you can sew the Circuit Playground in place using some of the unused holes around the edge.