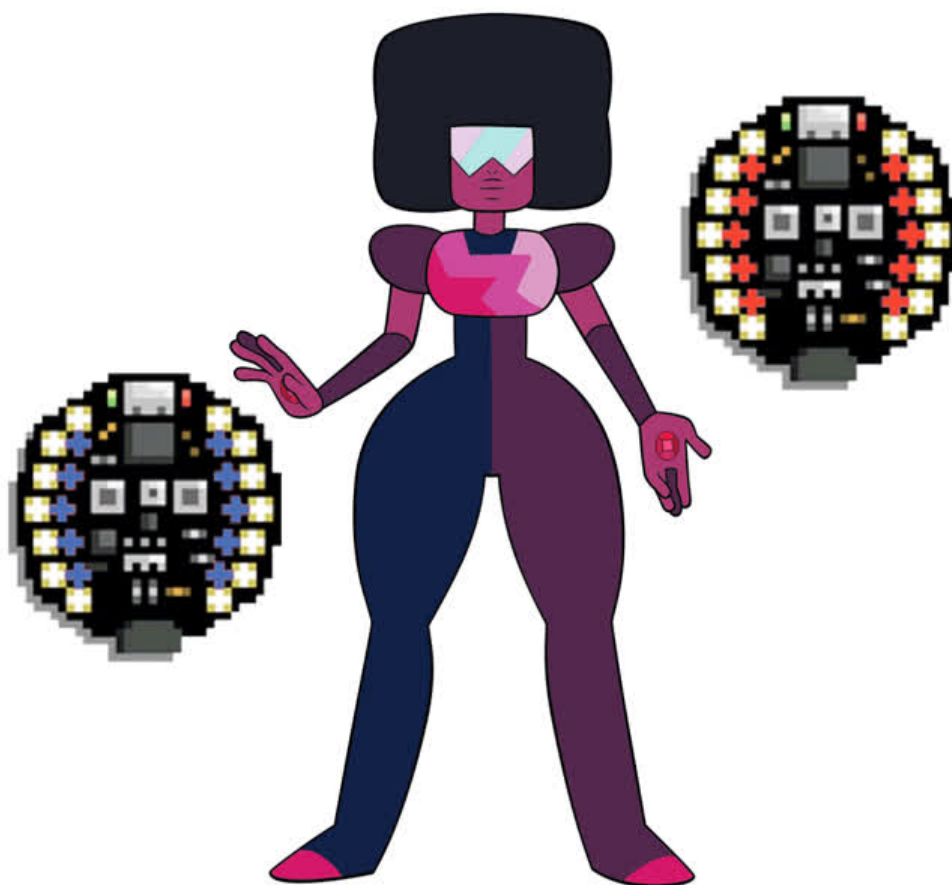




Cartoon Network MakeCode: Garnet's Palm Gems from Steven Universe

Created by John Park



<https://learn.adafruit.com/cartoon-network-makecode-garnet-s-palm-gems-from-steven-universe>

Last updated on 2024-06-03 02:38:17 PM EDT

Table of Contents

Overview	3
<hr/>	
• Parts	
Solo Fusion Gem Code in MakeCode	5
<hr/>	
• Solo Gem Fusion	
• Getting Started with MakeCode	
• Code Design	
• Brightness and Show Ring Setup	
• Button 'B' Click	
• Unfuse the Gem	
• Music	
• Theme Song	
Dual Fusion Gems Code in MakeCode	13
<hr/>	
• No Music	
• Infrared Communications	
Build the Palm Gems	19
<hr/>	

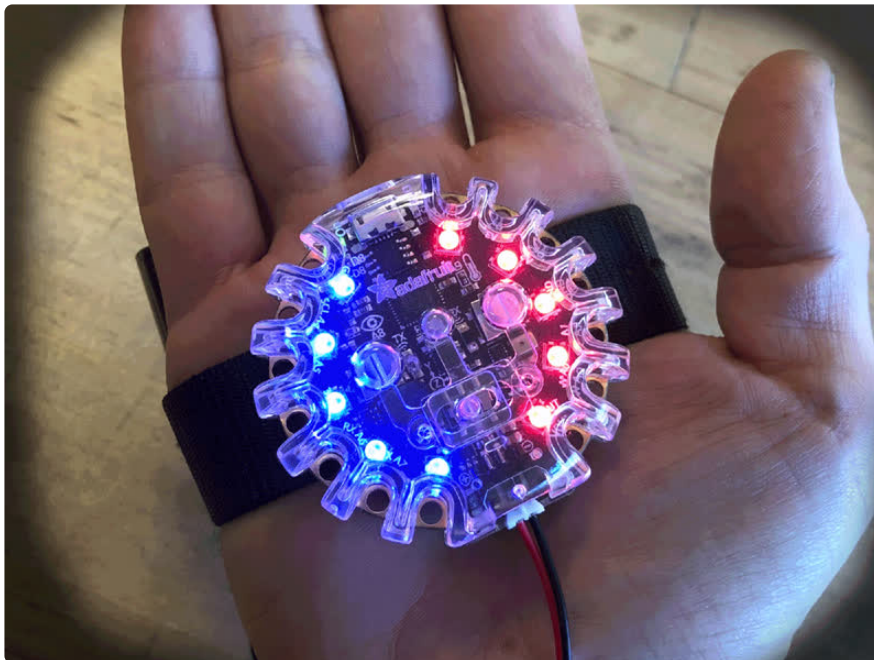
Overview



Make your own color-shifting Fusion Gems to turn Ruby and Sapphire into Garnet!

Use MakeCode to create two different projects -- a solo Circuit Playground Express gem that transforms colors, or a pair of them that react to each other!

Plus, you can create the Steven Universe theme song in MakeCode using a function and the built-in synthesizer engine!

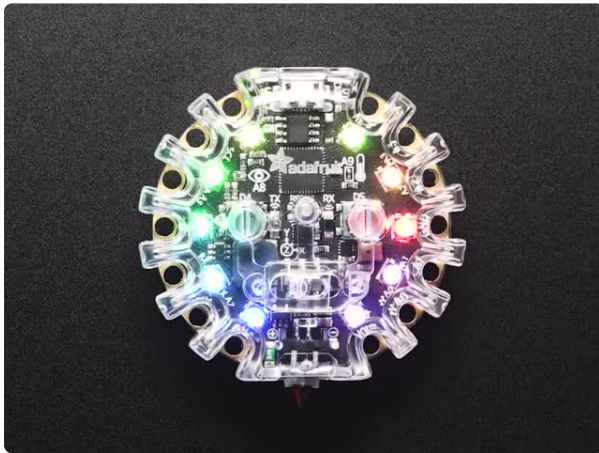


Parts

You can build this project with a single Circuit Playground Express (CPX), or do an optional two CPX version!

Round, Awesome Microcontroller Board
2 x **Circuit Playground Express**
Round, Awesome Microcontroller Board

<https://www.adafruit.com/product/3333>



Adafruit Circuit Playground Express or Bluefruit Enclosure

We've got nice cases for many of our beloved boards, but the Circuit Playground Express and

<https://www.adafruit.com/product/3915>



3 x AAA Battery Holder with On/Off Switch, JST, and Belt Clip

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/3286>



Alkaline AAA batteries - 3 pack

Battery power for your portable project!

These batteries are good quality at a good price, and work fantastic with any of the kits or projects in the shop that use AAA's. This is a...

<https://www.adafruit.com/product/3520>



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>

Solo Fusion Gem Code in MakeCode

Solo Gem Fusion



Getting Started with MakeCode

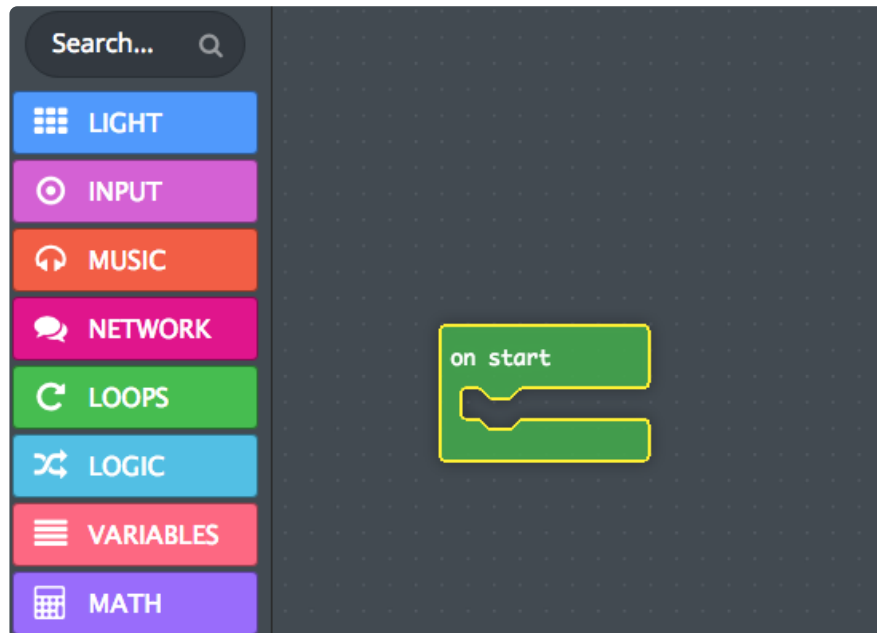
If you're new to MakeCode, [head to this guide to get started \(https://adafru.it/wWd\)](https://adafru.it/wWd). Once you're able to upload MakeCode to your Circuit Playground Express, return here.

Code Design

We'll break up our code into two sections: Lights and Music. In the light section, we'll use blocks from the **Light** category as well as tell the program what to do when buttons from the **Input** category are pressed.

Startup

In MakeCode, you can set up parameters for your project that only need to be stated once at startup -- these things go into the **on start** block. To start out drag an **on start** block out from the **Loops** category and onto your MakeCode canvas.

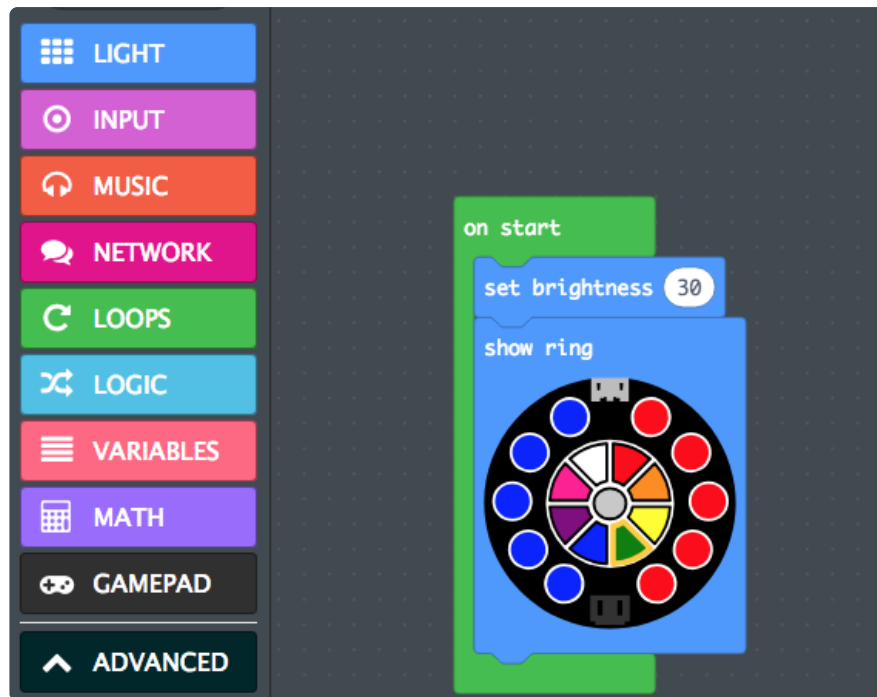


Brightness and Show Ring Setup

Next, get a **set brightness** block and a **show ring** block from the **Light** category and place them into the **on start** block. This will set the default brightness of the ten NeoPixel ring that surrounds the Circuit Playground Express, and allow us to specify the color pattern.

We'll give it an extra bit of brightness by setting the value to **30**.

Click on the "pie wedge" color pickers to choose a color and then click the NeoPixels in the circle to set them to half red and then half blue.



Button 'B' Click

Next, we'll create the Garnet fusion effect and a way to initiate it! Add an **on buttonA click** block from the **Input** category. Change the dropdown to be **button B**. This block will be used to detect when you click the button on the right side of the Circuit Playground Express, and then we'll specify what to do.

Animation

The first thing we want to happen is to run a swirling color animation. From the **Light** category, pick the **show animation for 500ms** block and add it to the **on button B click** block. In the animation block, click on the little colorful icon and choose a different animation -- we'll pick the comet animation as shown below. We also want to tell it to run for a little bit longer than the default half of a second -- set the value to 1500ms, which is a second and a half.

Ring Color

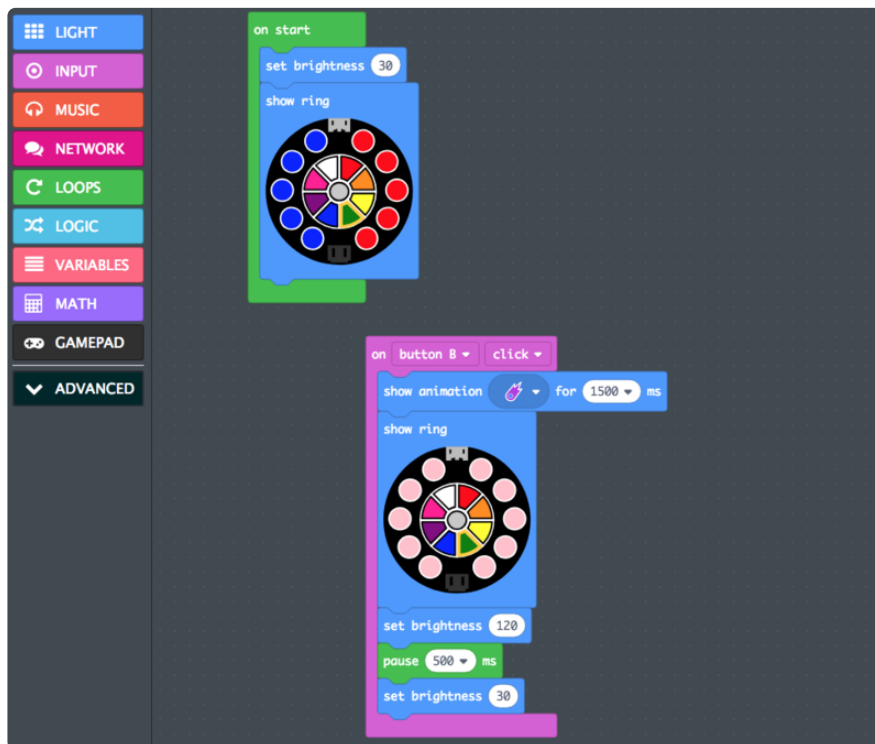
Once the animation stops we want to set the ring to a magenta color. Add a **show ring** block and change the color of each NeoPixel to magenta. (Note, the graphic shows up as a lighter, less saturated pink than it will in reality.)

Flash Effect

We also want to do a dramatic bright flash effect. To do so, add a **set brightness** block (hint: you can right-click the one in your **on start** block and choose the **duplicate** menu option) then set it up to **120**. This is pretty bright!

Add a **pause 100ms** block from the **Loops** menu next and change the value to **500ms**.

Then, set the brightness back down with another duplicate of the **set brightness** block, set to a value of **30**.

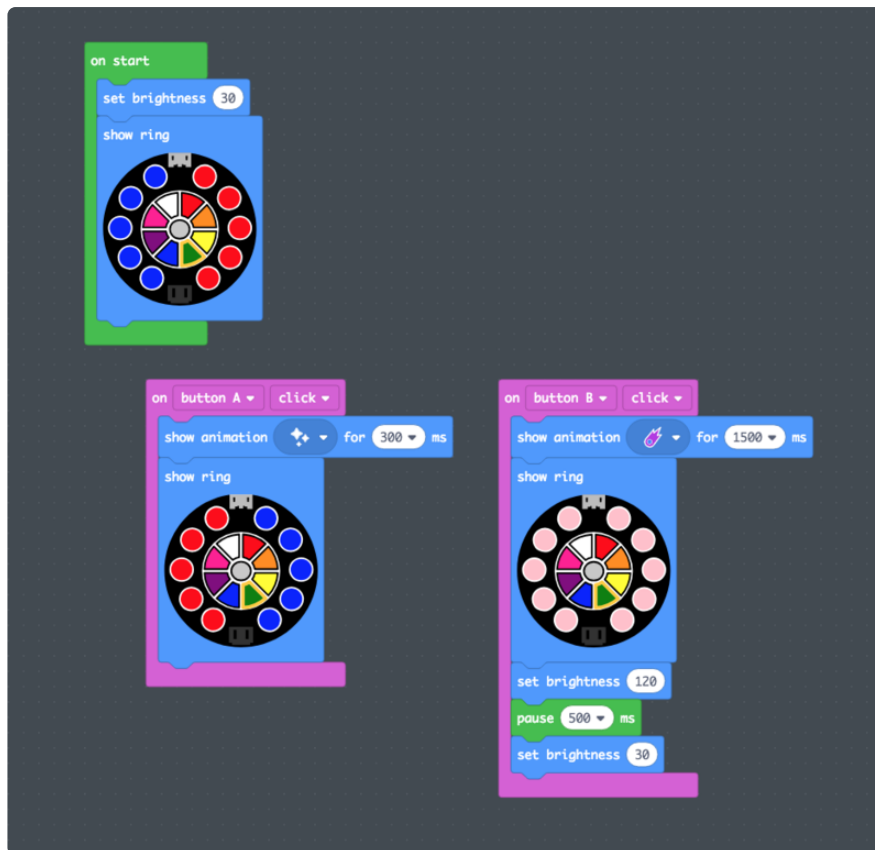


You can test this out now by downloading the code to your Circuit Playground Express that's plugged in over USB. It will start up, display red and blue NeoPixels, then you can press the right button and watch your fusion effect!

Unfuse the Gem

Now that we can fuse to Garnet, let's unfuse to Ruby and Sapphire! You can right-click and duplicate the entire existing **on button B click** block and then use that as the starting point. Drag the bottom three blocks to the left side of MakeCode in order to trash them, as we won't do the set brightness flash effect.

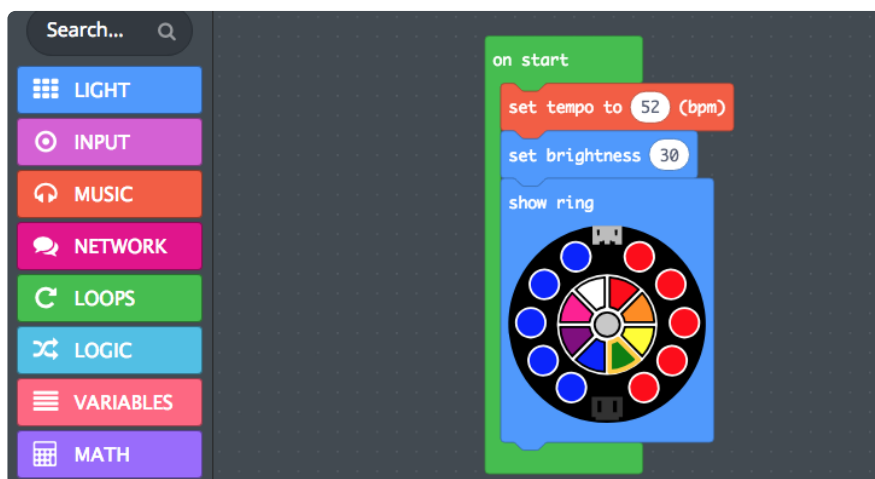
Then change the animation to 300ms of **sparkle** and set the colors back to red and blue as shown here.



Music

Let's add to the fun of the fusion by adding music and sound effects next! You can easily add a pre-made song from the **Music** category, but we'll go a bit deeper by re-creating the Steven Universe theme song.

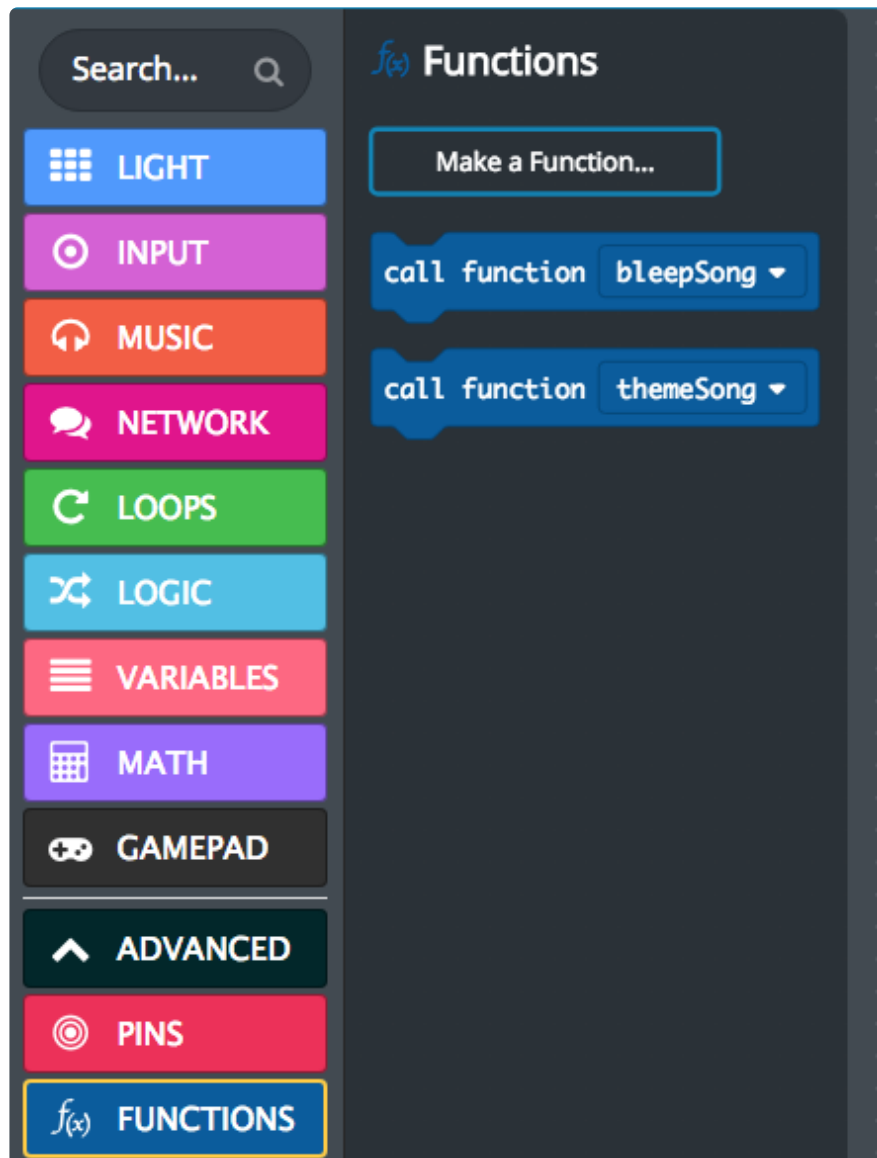
First, we can set the song tempo (beats per minute) in the **on start** block by adding a **set tempo to ___ (bpm)** block. Later, you can adjust the value to speed up or slow down playback tempo.



Functions

To keep our code a bit organized, we'll use **function** blocks. These can hold a set of code blocks on their own, yet be accessed from elsewhere, such as within a button click block.

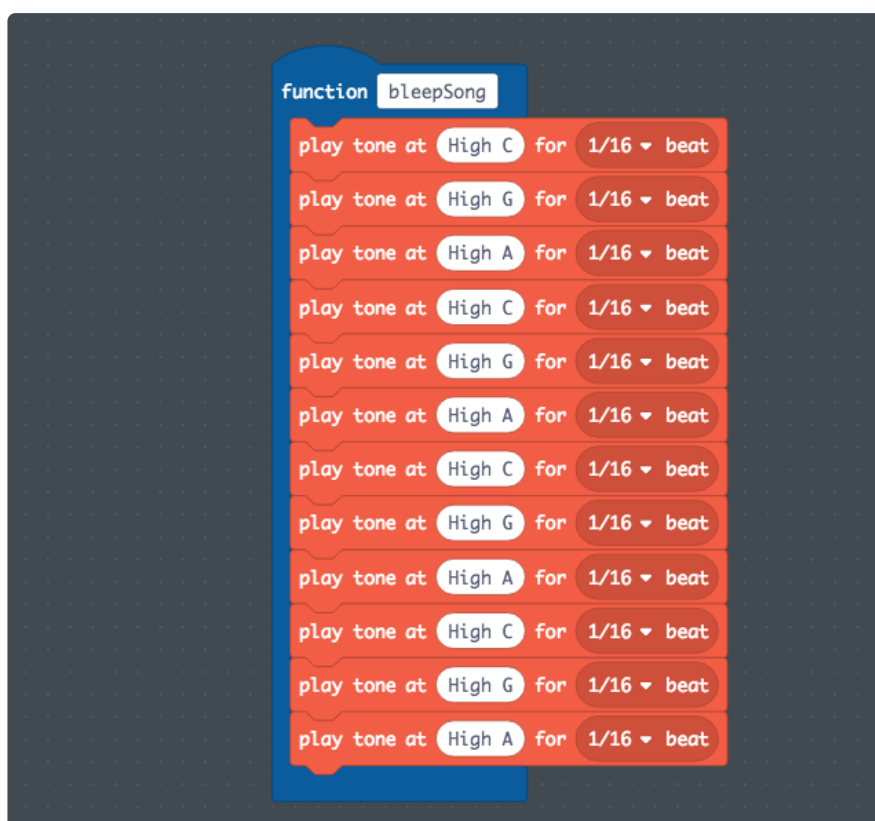
Click on the **Advanced** button and then the **Functions** button. Then click on **Make a Function...** and type the name **bleepSong** when prompted, then click **OK**.



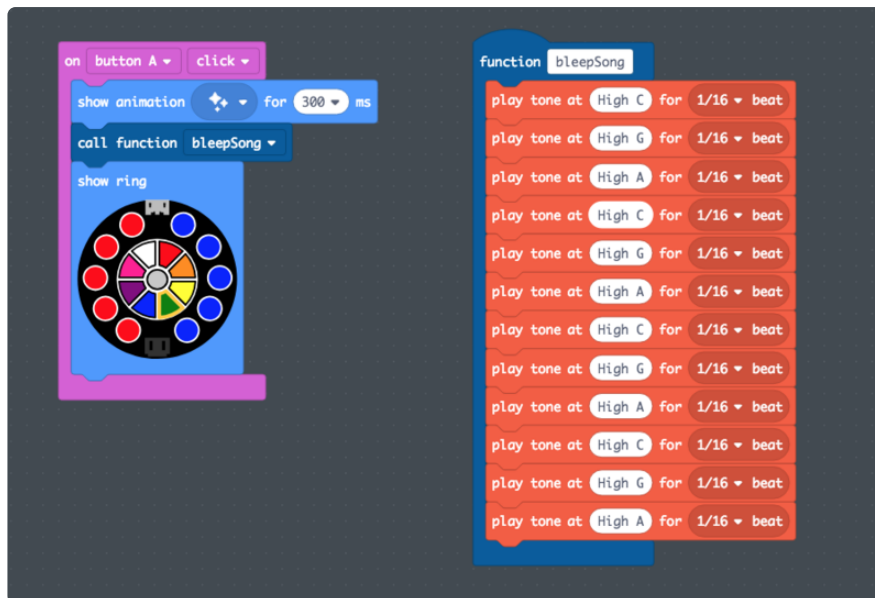


Now, you can start adding individual notes of your song from the **Music** category to the **bleepSong** function. Drag in the **play tone at Middle C for 1/2 beat** block.

I made a little bleep-y arpeggio by playing C-G-C 1/6th notes as shown here:



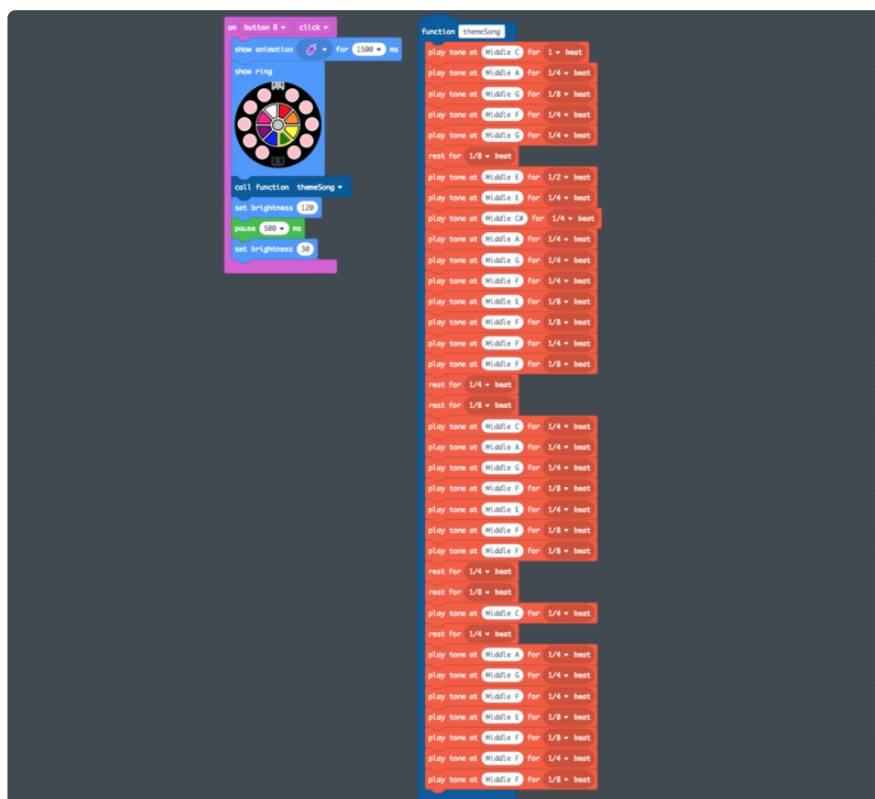
When you want to run the code that is inside of a function, you'll use the **call function _____** block found inside of **Advanced > Functions**. Here, I've placed this block inside the **on button A click** block after the animation runs. Now, when the button is pressed, the animation will play, then the bleepSong code will play, and finally, the **show ring** block will run.



You can imagine that without the function block our code could get pretty messy!

Theme Song

You can repeat this technique in order to create the theme song from Steven Universe! Here's the version I came up with, the only difference from the **bleepSong** is that it is much longer, and it also uses the **rest for ____** blocks.



Here's what the finished code looks like. If you want to check it against the final live code, [here's a link \(https://adafru.it/DJfj\)](https://adafru.it/DJfj).

You can upload it to your CPX and try it out! Press B to fuse to Garnet, and then A to un-fuse to Ruby and Sapphire!!



Next, we'll build a double Gem version using infrared communications! Or, skip that page and go straight to building the palm holder on the final page.

Dual Fusion Gems Code in MakeCode

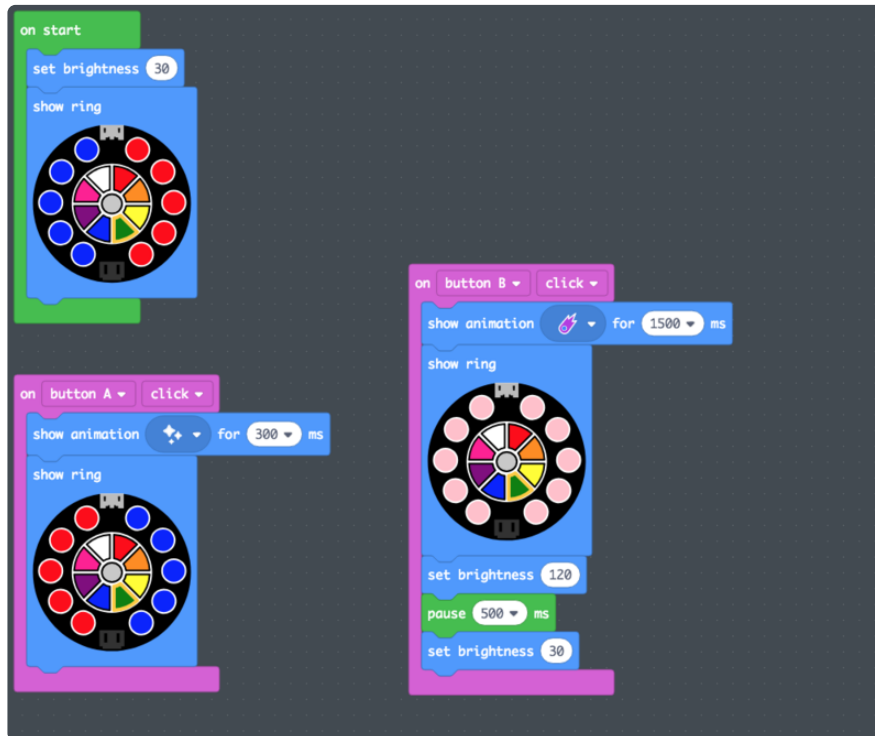


Now that you've created the solo gem fusion, it's easy to modify it to work on a dual gem setup. You'll use two Circuit Playground Express boards and then code them in a way that they can send messages to each other!

This will allow you to press the B button on one of them, for example, and they will both go into the Garnet fusion sequence fully synchronized!

No Music

Currently, there is a bug when using IR (infrared) communications and music blocks in the same code. So for this version, we'll start from [the previous code \(https://adafru.it/DJf\)](https://adafru.it/DJf) and then delete the music related blocks and functions.

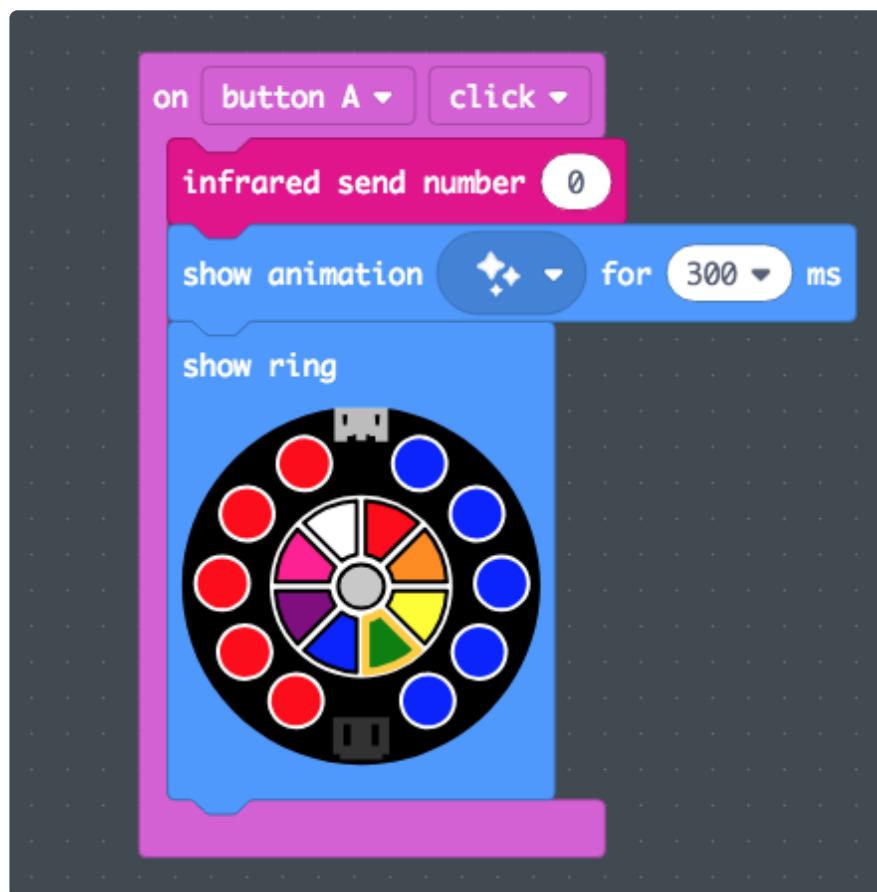
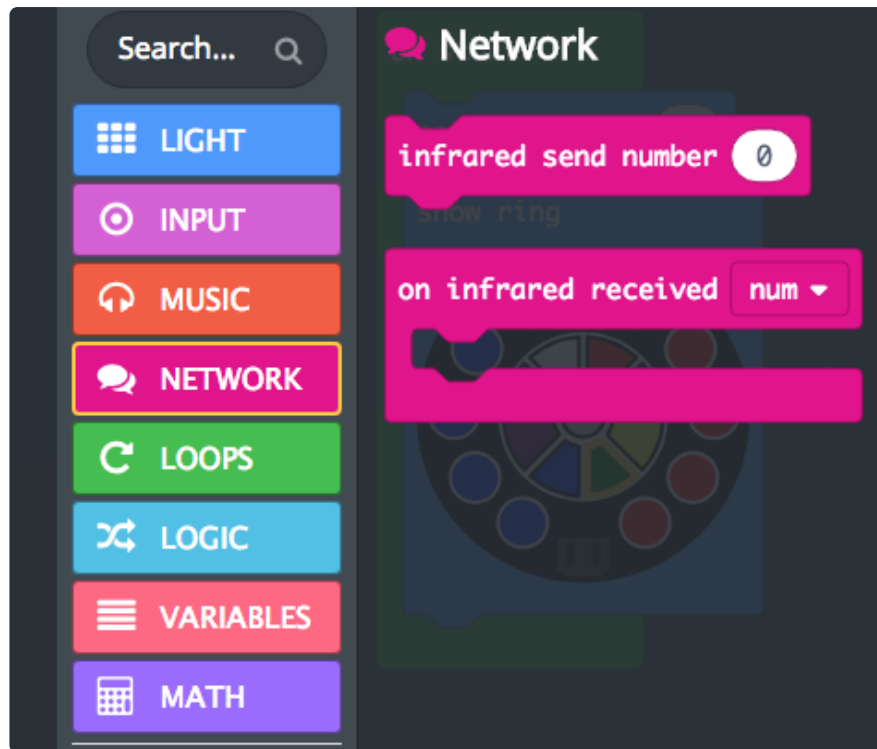


Infrared Communications

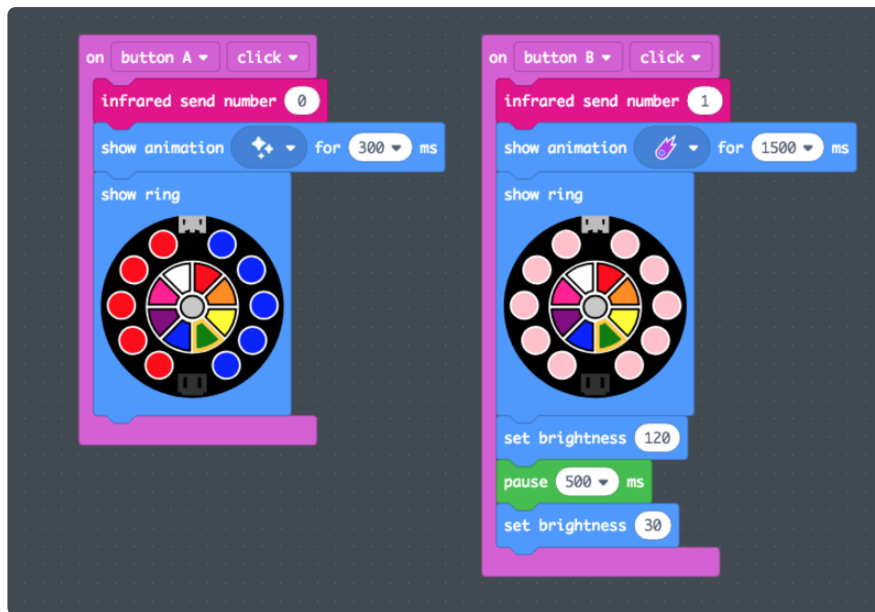
We want to send and receive messages using IR between the two Circuit Playground Express boards.

Send

Click the **Network** category and add an **infrared send number 0** block to the **on button A click** block as shown.



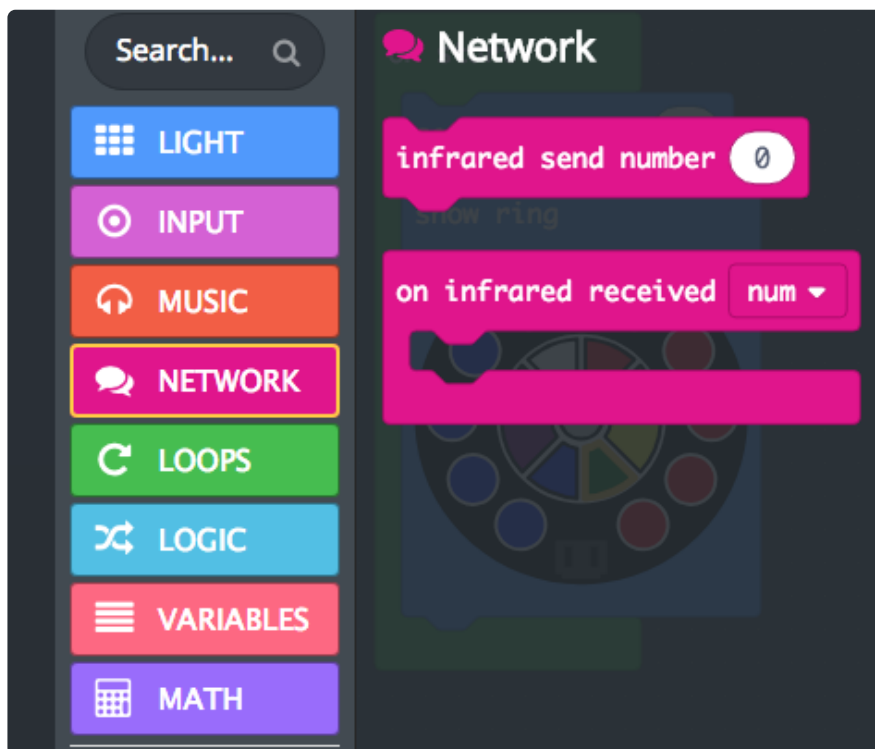
This means that whenever you click button A, an IR message will be sent out containing the number **0** as its message. We'll now duplicate this block and add it to the **button b** block, but change the message to be a number **1**.

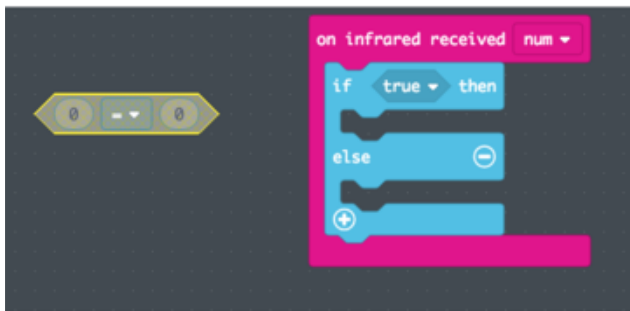


Receive

OK, now we are sending out unique messages with the two different buttons, but we need to also tell the Circuit Playground Express boards what to do when they receive a message!

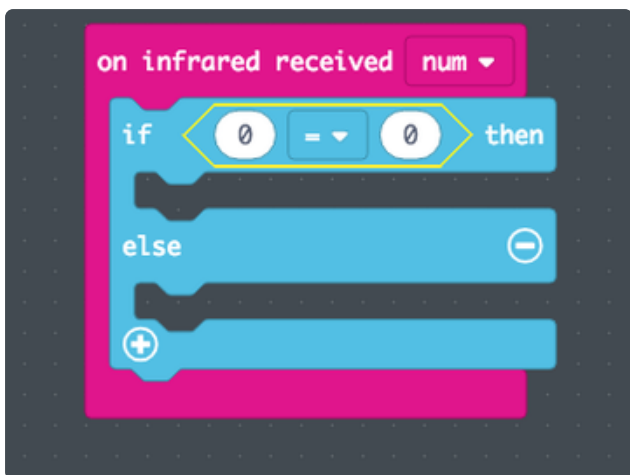
From the **Network** category, add an **on infrared received num** block.



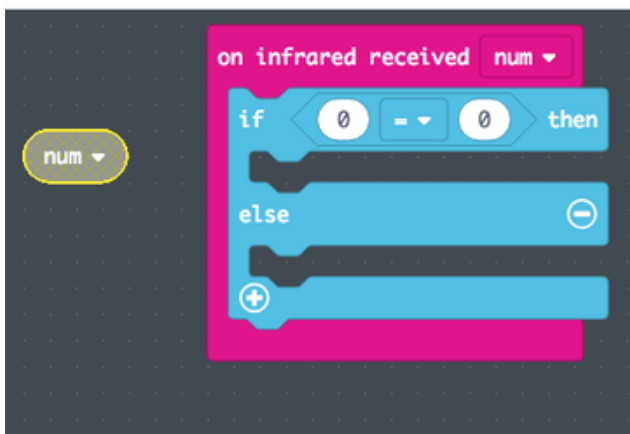


Conditionals and Comparisons

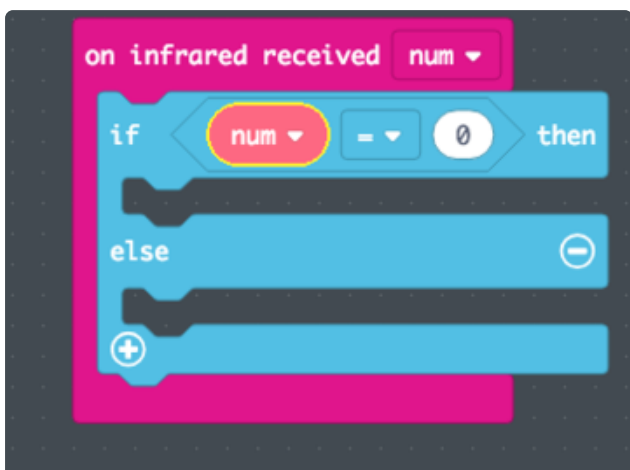
When the **on infrared received** block receives a message, we want to check to see if the message is a **0**, meaning the **A button** was pressed, or if the message is a **1**, meaning the **B button** was pressed.



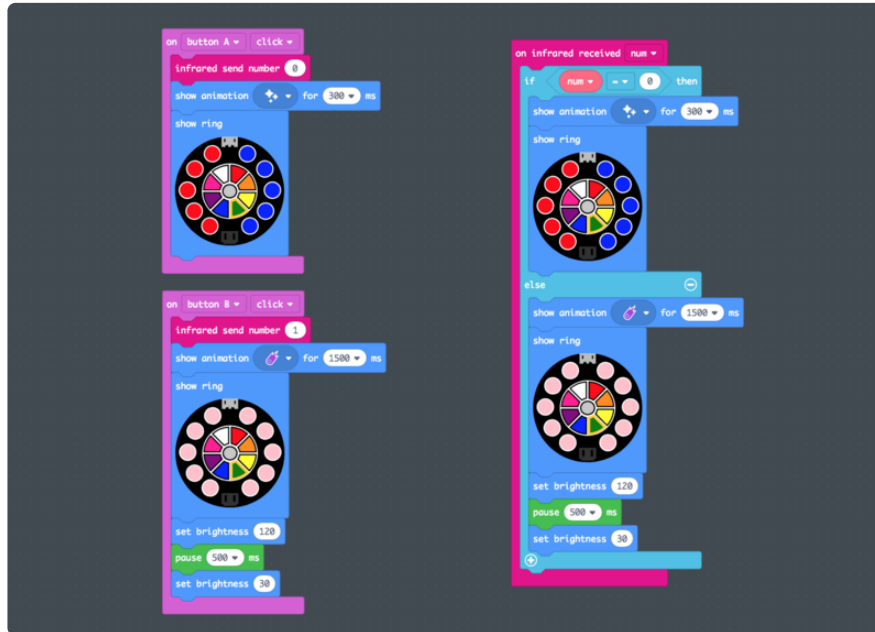
The way we do this is with a conditional block found in the **Logic** category called **if true then...else** and a comparison block called **0 = 0**.



Add them to the received block as shown here.



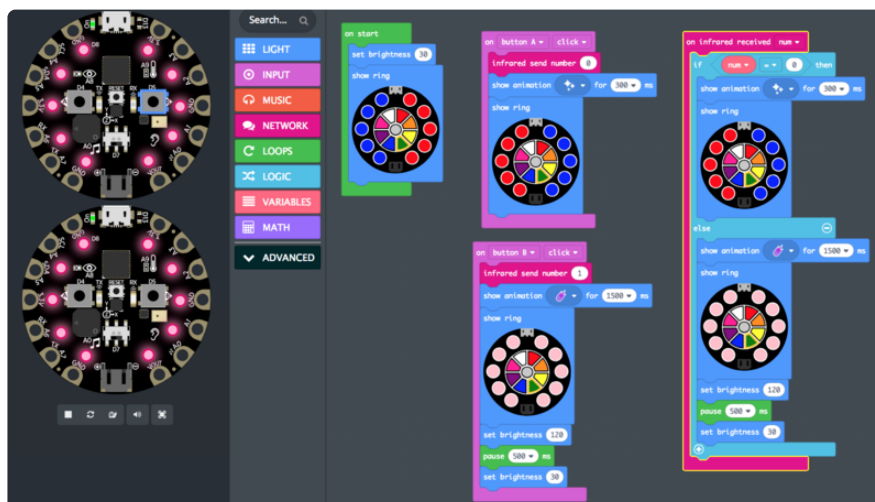
Now, we can tell the conditional block what to do in the two different cases. We'll duplicate the existing animation and light block contents of our **on button A click** block and place it into the **if** section, and then do the same with the **on button B click** block and place it in the **else** section. We don't need to duplicate the **infrared send** blocks.



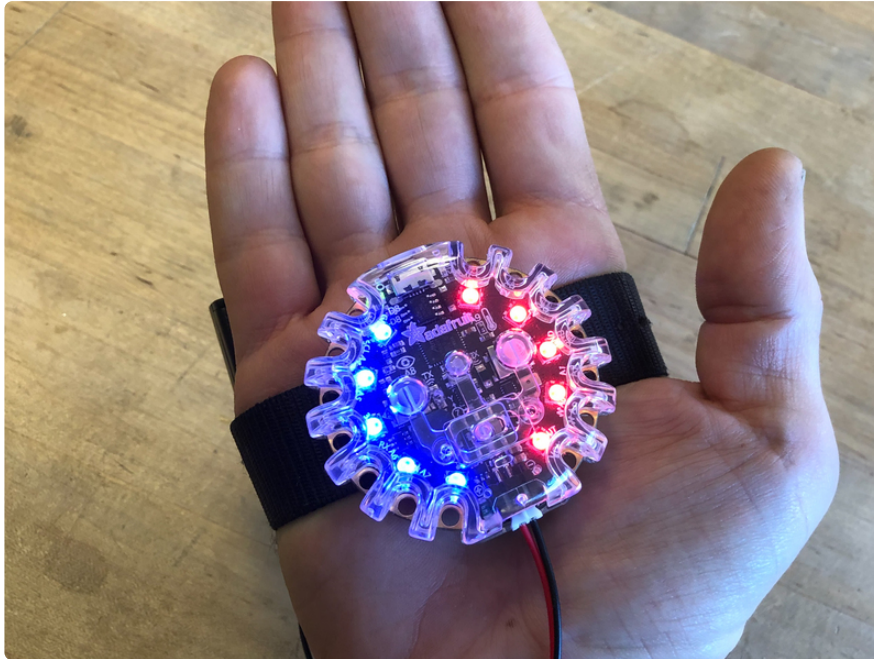
Now, as you can see, when a button is placed on one Circuit Playground Express, it will send an IR message to the second one and they will both run through the same animation and light code together!

Upload the [final code](https://adafru.it/DJg) (<https://adafru.it/DJg>) to both of your Circuit Playground Express boards (one at a time) and then plug both into power and try it out! Garnet's Fusion is complete!!

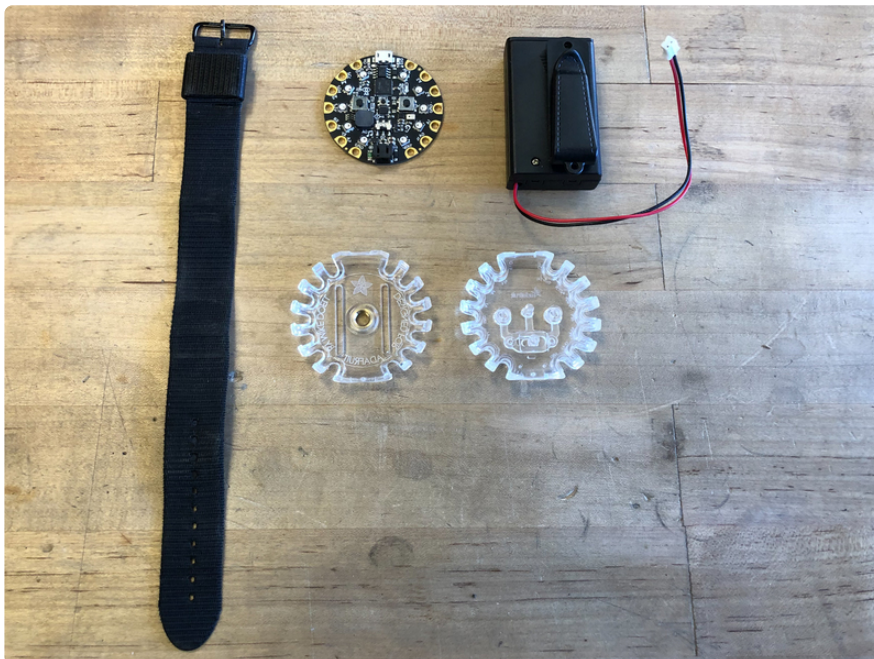
Tip: You can even try it out in the simulator! Just press one of the virtual buttons and a second CPX will appear.



Build the Palm Gems



You can hold the CPX boards in your palms and use a USB power or AAA battery box, or, dress things up a bit with Circuit Playground Express enclosures and some watch bands or straps for a more secure option suitable for a costume!





To assemble, first open the CPX enclosure and thread the band, strap, or string through the slots.



Then, place the CPX into the case and close it.

Now, plug the AAA battery pack into the CPX.



Hook the battery pack to the strap, and secure it to your hand.

