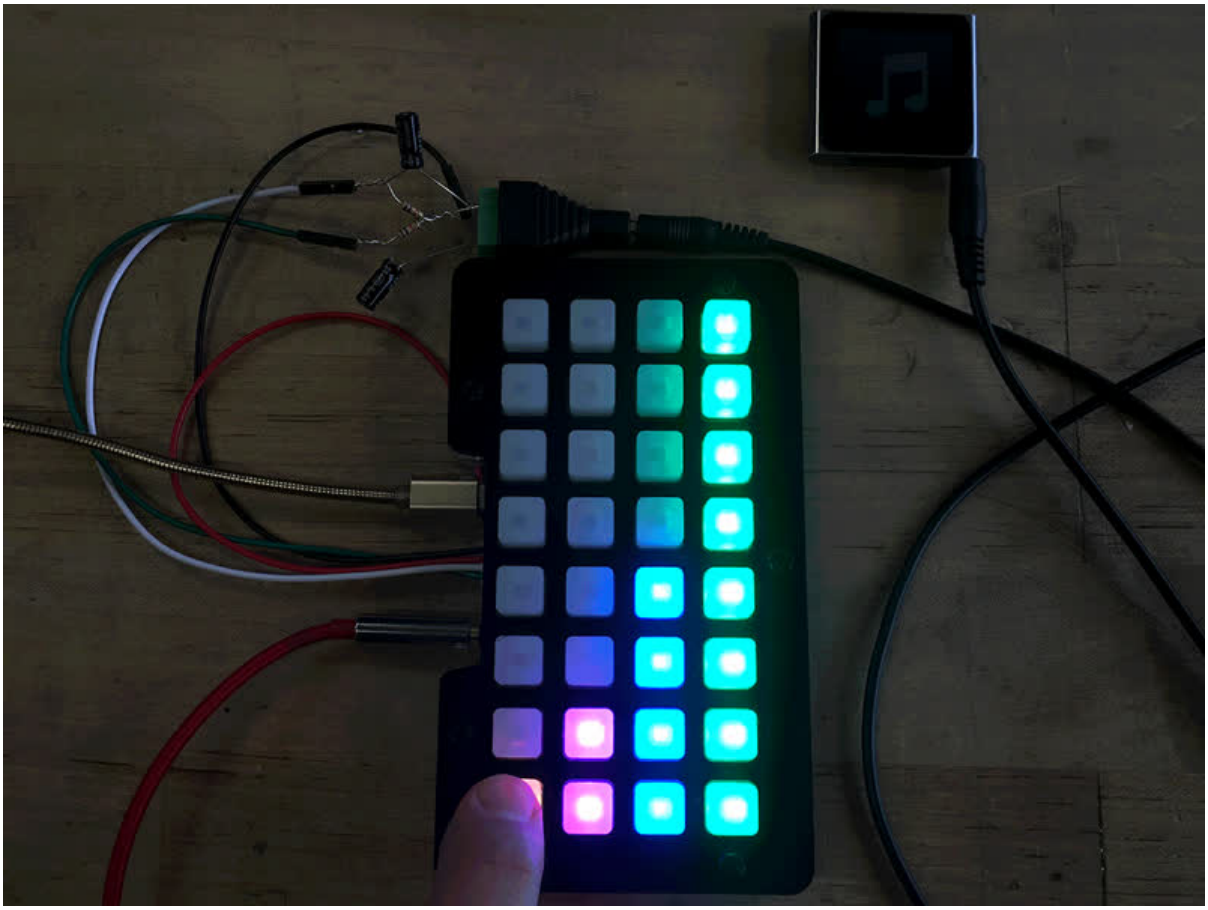
Trellis M4 Audio Filter Visualizer

Created by John Park



https://learn.adafruit.com/trellis-m4-audio-visualizer-and-filter

Last updated on 2024-03-08 03:15:24 PM EST

Table of Contents

Overview 3

- Parts

Filters and Visualisation 7

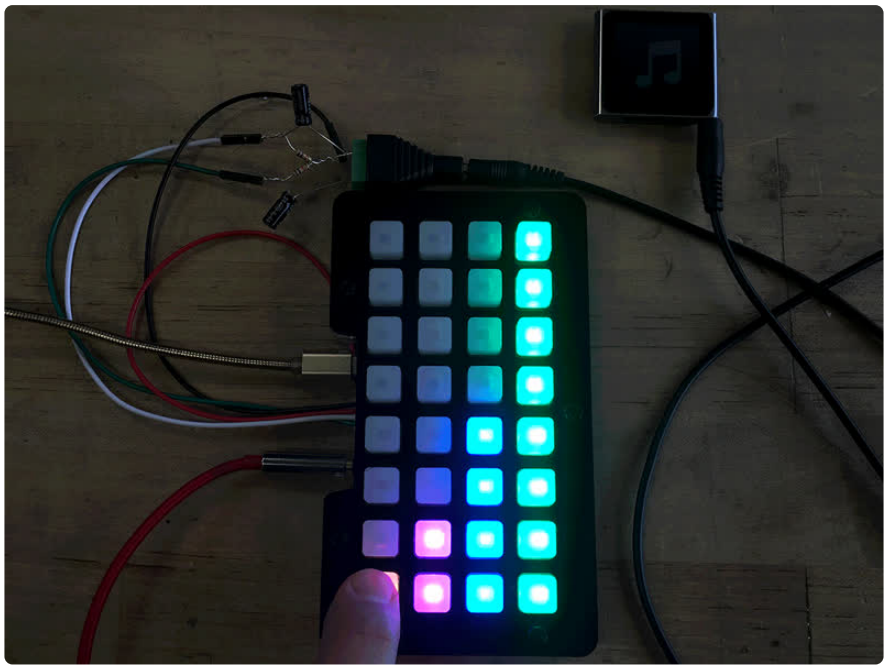
- Audio Library
- FIR Filter
- FFT Spectrum Visualization

Audio Input Circuit 9

Use the Audio Filter Visualizer 12

- Firmware
- Deeper Modification

Overview



Colored lights bouncing to the music is a must have for any party. Now you can bring the party to your NeoTrellis M4 Express with the Audio Visualizer and Filter! With realtime Fast Fourier Transform (FFT) frequency spectrum analysis, the music coming into your NeoTrellis STEMMA adapter (from a computer, phone, Walkman, turntable pre-amp, etc.) is converted into colors and lights on the buttons to display the magnitude of the sound in different frequency bands.

Your music then passes through to the TRRS headphone jack on the NeoTrellis so you can listen to it. But that's not all! We've also built in a low pass filter and band pass filter so you can trigger awesome sounding effects at the push of a button!

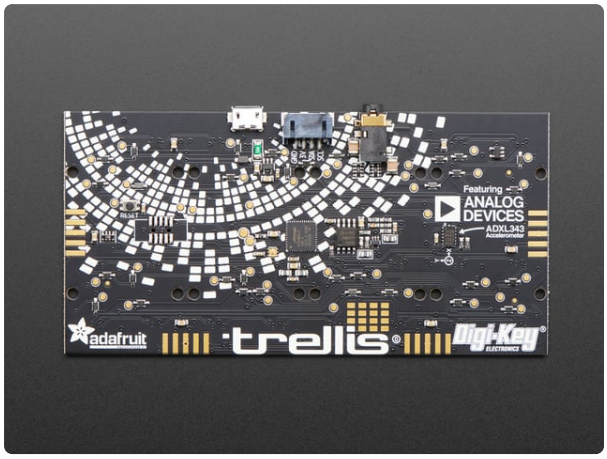
We'll also build a small circuit with capacitors and resistors to center the incoming audio around the proper voltage for the NeoTrellis M4's analog-to-digital (ADC) converter.

Parts

Adafruit NeoTrellis M4 with Enclosure and Buttons Kit Pack

So you've got a cool/witty name for your band, a Soundcloud account, a 3D-printed Daft Punk...

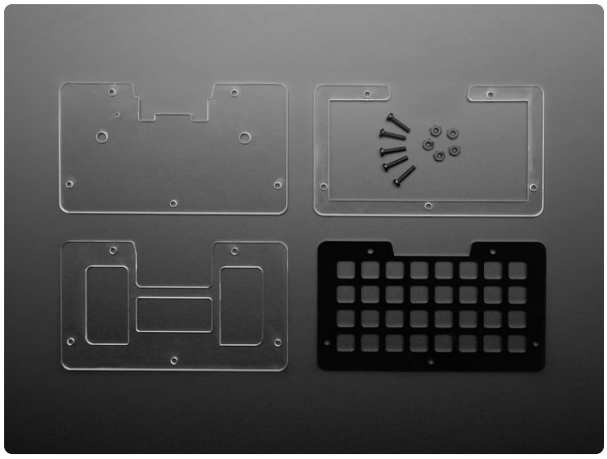
https://www.adafruit.com/product/4020



Adafruit NeoTrellis M4 Mainboard - featuring SAMD51

We got a big ol' blender and tossed in an ItsyBitsy M4, two NeoTrellis boards and an electret mic amp - turned on the 'mix' button and out came the NeoTrellis M4 - a super...

https://www.adafruit.com/product/3938



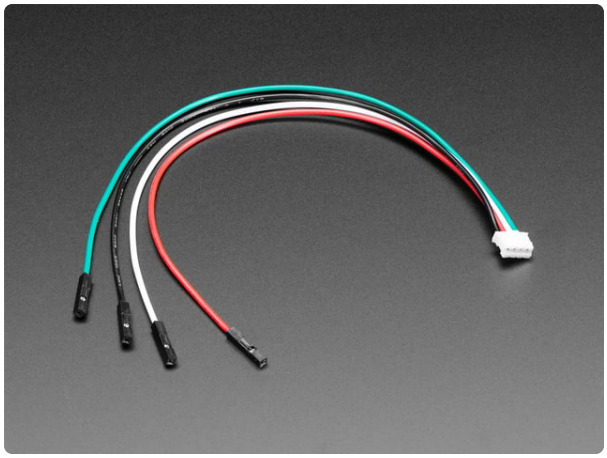
NeoTrellis M4 Acrylic Enclosure Kit

So you've got your Adafruit NeoTrellis M4, a cool/witty name for your band, a Soundcloud account,

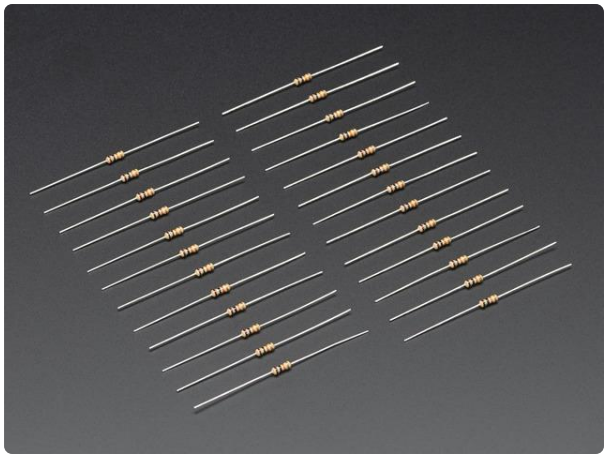
https://www.adafruit.com/product/3963

[Silicone Elastomer 4x4 Button Keypad - for 3mm LEDs]
So squishy! These silicone elastomer keypads are just waiting for your fingers to press them. Go ahead, squish all you like! (They're durable and easy to clean, just wipe with mild...
https://www.adafruit.com/product/1611

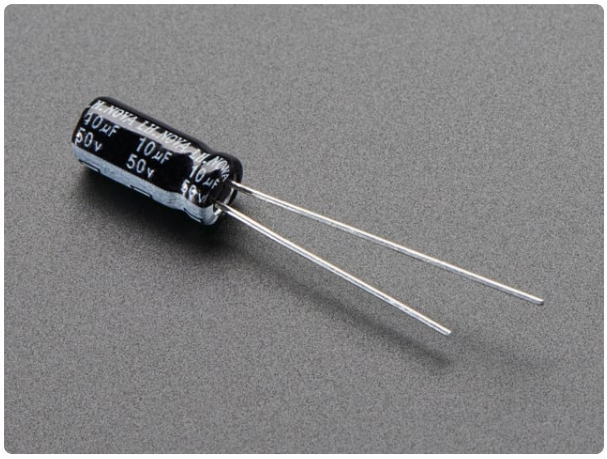
[3.5mm (1/8") Stereo Audio Jack Terminal Block]
One truth about working with audio is you always need the cable or adapter you don't have in your toolbox. That's why we love these terminal-block audio connectors so...
https://www.adafruit.com/product/2791



[JST PH 2mm 4-Pin to Female Socket Cable - I2C STEMMA Cable]
This cable will let you turn a JST PH 4-pin cable port into 4 individual wires with high-quality 0.1" female sockets on the end. We're carrying these to match up with any of...
https://www.adafruit.com/product/3950



[Through-Hole Resistors - 10K ohm 5% 1/4W - Pack of 25]
ΩMG! You're not going to be able to resist these handy resistor packs! Well, axially, they do all of the resisting for you!This is a 25 Pack of 10K...
https://www.adafruit.com/product/2784



10uF 50V Electrolytic Capacitors - Pack of 10

We like capacitors so much we made a kids' show about them. ...

https://www.adafruit.com/product/2195

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

3.5mm Male/Male Stereo Cable

Seamlessly transmit high-quality stereo audio with this 3.5mm Male/Male Stereo Cable. Ideal for "passing the AUX cord," or connecting...

https://www.adafruit.com/product/2698

In addition to the parts above you'll need some powered speakers or headphones to listen to the fly tunes.

USB Powered Speakers
Add some extra boom to your audio project with these powered loudspeakers. We sampled half a dozen different models to find ones with a good frequency response, so you'll get...
https://www.adafruit.com/product/1363

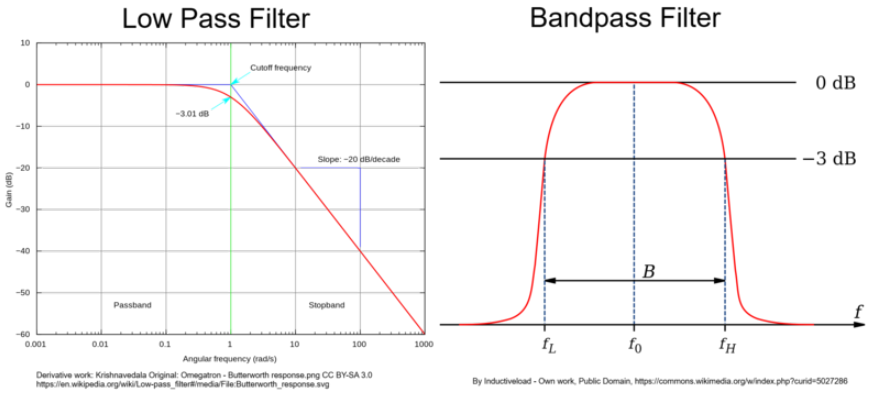
Cell-phone TRRS Headset - Earbud Headphones w/ Microphone
These earbud headphones are the perfect accessory for your FONA - they've been tested to work with our modules - but can be used with any iOS or Android device that uses a TRRS...
https://www.adafruit.com/product/1966

Filters and Visualisation

Audio Library

This project leverages the excellent Audio library (https://adafru.it/CVT) for Arduino which has been forked for the NeoTrellis M4 Express from the original Teensy Audio Library by Paul Stroffregen at pjrc (https://adafru.it/Dfv). The Audio library is described this way: "A toolkit for building streaming audio projects, featuring Polyphonic Playback, Recording, Synthesis, Analysis, Effects, Filtering, Mixing, Multiple Simultaneous Inputs & Outputs, and Flexible Internal Signal Routing. "

The two key things we're doing with code from the Audio library are filtering audio and analyzing it for visualization.



FIR Filter

The Finite Impulse Response (FIR) Filter code filters the audio based on frequency bands we specify. Wikipedia (https://adafru.it/Dfw) describes electronic filters this way:

> ...circuits which perform signal processing (https://adafru.it/Dfx) functions, specifically to remove unwanted frequency components from the signal, to enhance wanted ones, or both.

The frequency range of a song can span from 20 Hz to 20,000 Hz (the audio spectrum of human hearing) but we can use an FIR filter to selectively pass through only certain bands (ranges) of those frequencies. You can drop out all of the mid and upper bands, only allowing the low end to remain by using a **low pass** filter. This gives us a type of calm, muffled sound that's great for queuing a build-up in dance music, for example. Our example allows frequencies below 1000 Hz to pass through.

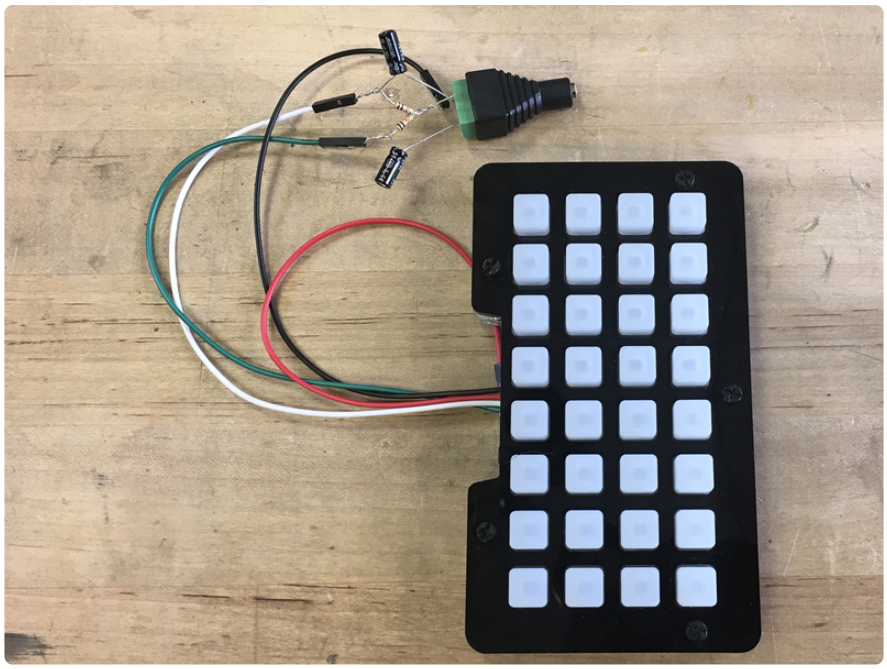
We also have a **band pass** filter that allows only mid-range to pass through (a band pass can be any band of frequencies you like, we just happen to have chosen a mid-range here). Our example allows frequencies from 1200 Hz to 1700 Hz to pass.

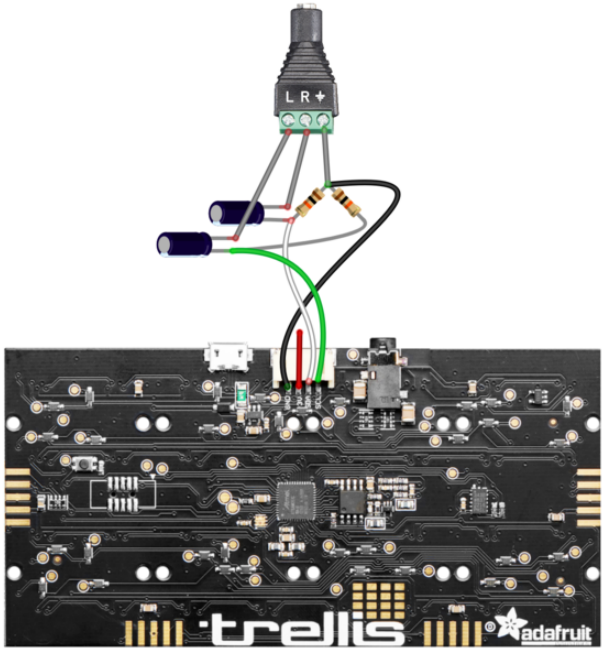
FFT Spectrum Visualization

To turn the Trellis M4's button LEDs into a spectrum analyzer, we need to measure the incoming audio's signal strength in the full frequency range.

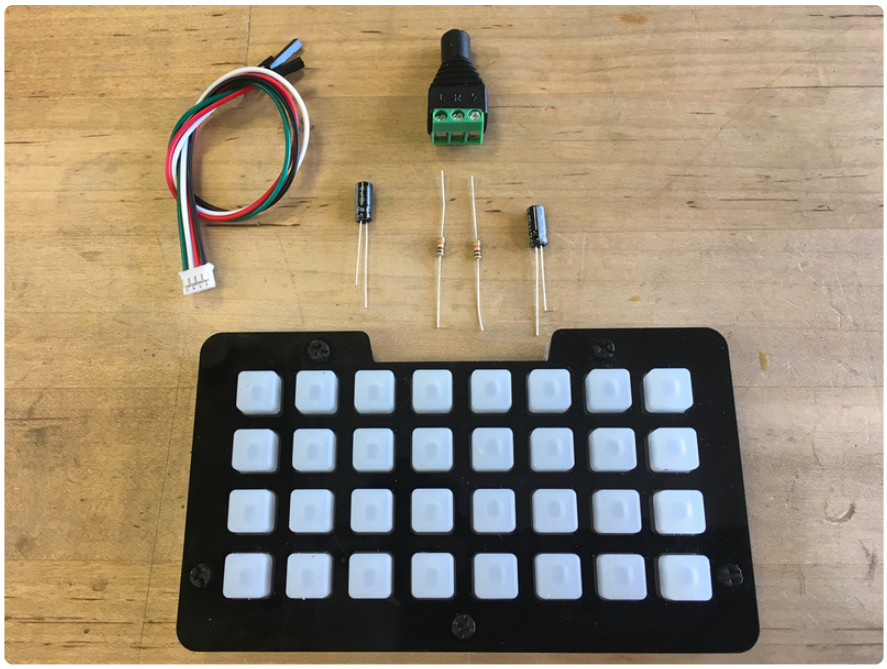
The Fast Fourier Transform code computes a 1024 point audio frequency analysis that we can use for visualization of the magnitude of the sound in various frequency range "bins". These values are then used to determine in real time which NeoPixel's to light up in the Trellis M4 buttons.

Audio Input Circuit

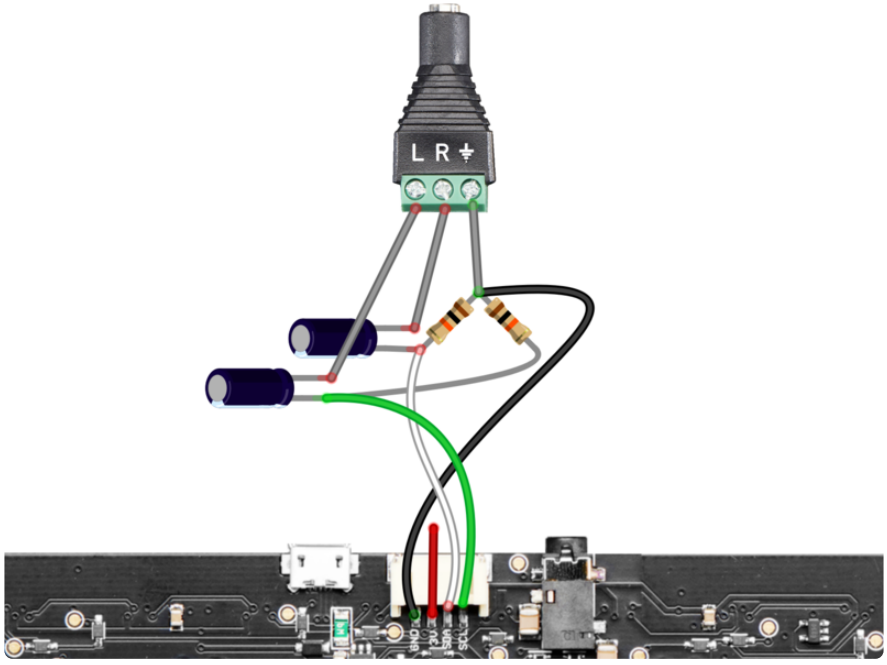


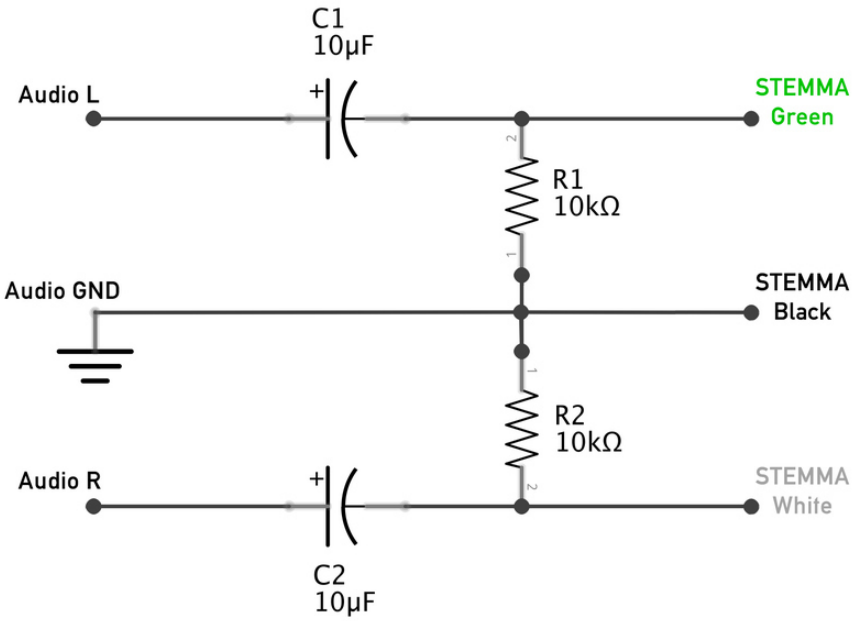


The audio signal coming from your music player needs a bit of massaging before it can be read by the analog inputs on the NeoTrellis M4. We'll build a small circuit using two capacitors and two resistors to remove the DC bias and center the incoming audio around the correct voltage for the NeoTrellis M4's analog-to-digital converter (ADC).



If you haven't already put together your NeoTrellis M4, follow these instructions on assembly (https://adafru.it/D0j).





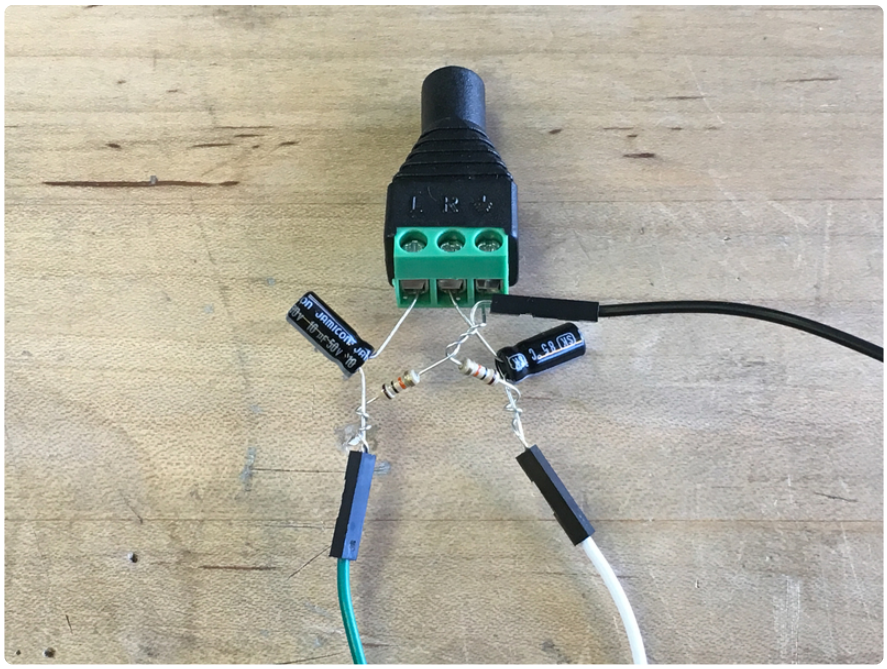
This can be done on a breadboard if you like, but if you want to keep it small and mobile, you can do without, and go for a "dead bug" style of construction. These are the connections:

- NeoTrellis M4 STEMMA connector **green** to first 10uF **capacitor negative** leg (the shorter leg, and the one with the white stripe on the body of the cap)
- First **capacitor positive** (longer leg) to audio jack terminal block **'L' terminal**
- **STEMMA** connector **green** to first **10k resistor** (either leg)
- First **resistor** to audio jack terminal block **GND**

- **STEMMA** connector **white** to second **capacitor negative**
- Second **capacitor positive** to audio jack terminal block **'R' terminal**
- **STEMMA** connector **white** to second **10k resistor**
- Second **resistor** to audio jack terminal block **GND**

- **STEMMA** connector **black** to audio jack terminal block **GND**

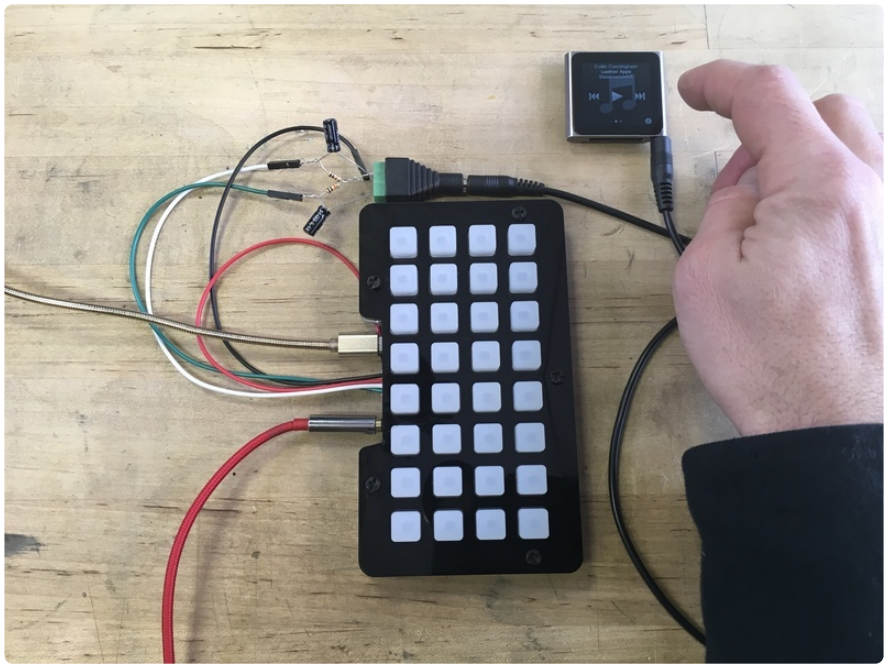
You'll find it easiest if you twist the legs of some components before inserting them into the STEMMA wire connectors as shown in this video. Just be sure that there are no accidental connections being made from the leads touching where not intended.



Plug the JST PH 4-pin cable into the STEMMA connector on your NeoTrellis M4 and you'll be able to read the incoming audio signal.

Next, let's flash the NeoTrellis M4 firmware with the filter/visualizer code.

Use the Audio Filter Visualizer



Firmware

All you need to do to use the visualizer and filters on your NeoTrellis M4 is to plug it into USB, download the firmware .uf2 file below, place your NeoTrellis into bootloader mode. Bootloader mode is entered by double-clicking the reset button on the back, and drag the firmware file onto the **TRELSM4BOOT** drive that shows up.

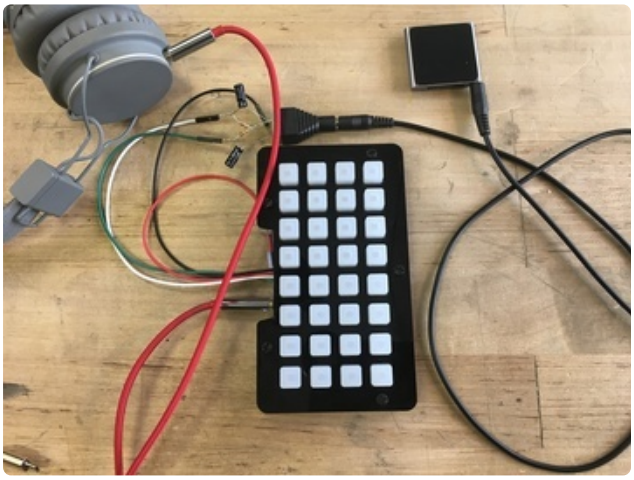
If you plug your NeoTrellis in and the drive shows up as **CIRCUITPY**, that means it is in CircuitPython mode. No Worries, press the reset button until you see **TRELSM4BOOT** which is the flash drive to place your .uf2 on. If you were to copy the .uf2 to **CIRCUITPY**, it would not run, hence ensuring the drive is **TRELM4BOOT**.

Download the Filter FIR NeoTrellis Firmware

https://adafru.it/Dfs

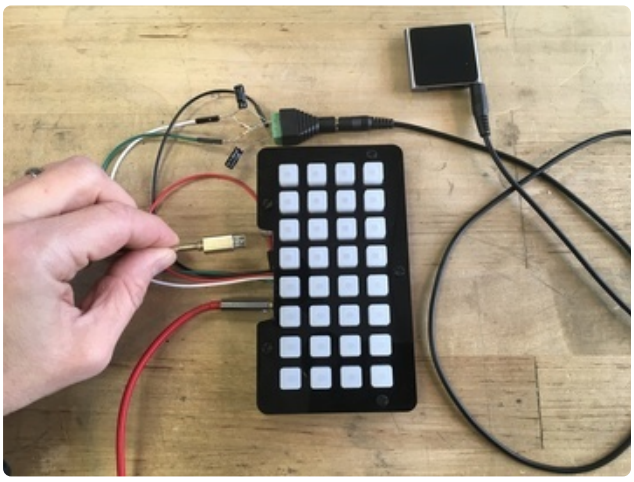
This program will replace CircuitPython, to get back to your CircuitPython projects, reinstall it by following https://learn.adafruit.com/adafruit-neotrellis-m4/circuitpython

Once you've flashed the firmware, here's how to use it:



Plug the output of a music player, such as a phone, computer, mp3 player, or Walkman, into the audio jack terminal block adapter

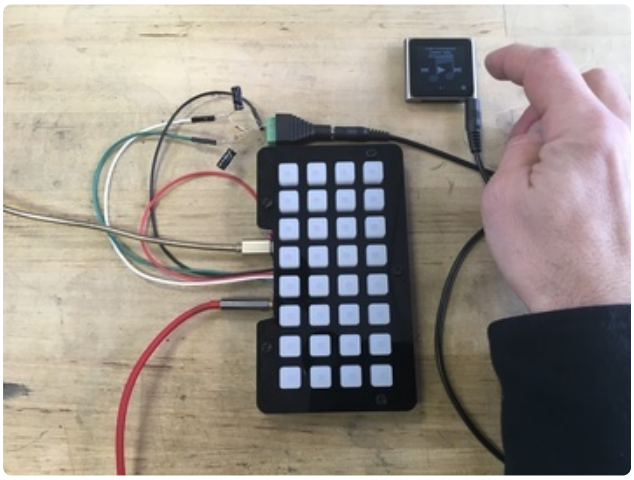
Then, connect headphones or powered speakers to the NeoTrellis M4's built in TRRS 3.5mm jack



Power the NeoTrellis M4 over USB

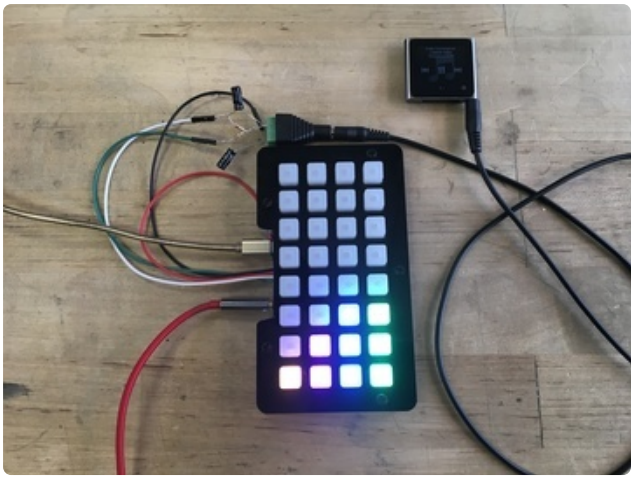
Play music on your player

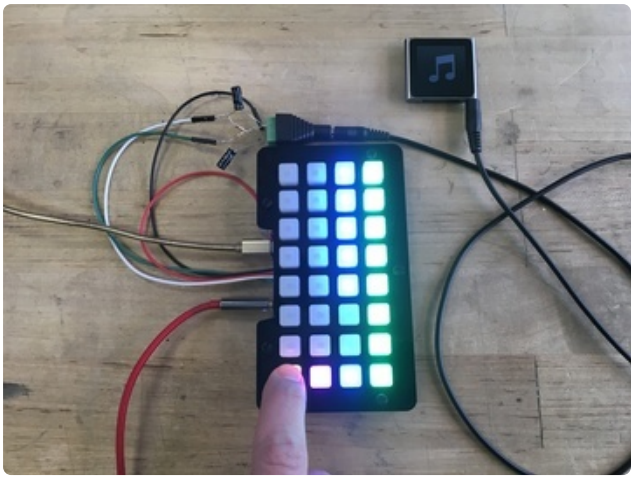
The NeoTrellis M4 will automatically apply the low pass filter, and start visualizing the music in the lower section of the NeoPixel lit buttons -- it should have a muffled sound since the mid and upper frequencies are being filtered out



Press and hold the bottom left button (NeoTrellis M4 in vertical orientation, USB jack facing left) to hear the pure, unfiltered audio, and note how the visualization changes

Let go of the bottom left button and press the one above it to switch to a band pass filter





Deeper Modification

For more advanced hacking, if you want to tweak the Arduino code, you can head to the GitHub repo (https://adafru.it/Dft) to have a look!