



# Classic MIDI Synth Control with Trellis M4

Created by John Park



<https://learn.adafruit.com/classic-midi-synth-control-with-trellis-m4>

Last updated on 2024-03-08 03:14:17 PM EST

# Table of Contents

<b>Overview</b>	<b>3</b>
<ul style="list-style-type: none"><li>• <a href="#">Parts</a></li></ul>	
<b>MIDI Connections</b>	<b>7</b>
<ul style="list-style-type: none"><li>• <a href="#">Electronic Music Gear</a></li><li>• <a href="#">MIDI Messages</a></li><li>• <a href="#">NeoTrellis M4 MIDI</a></li><li>• <a href="#">Adapting MIDI to the 3.5mm TRS Plug</a></li><li>• <a href="#">TRS Adapter</a></li><li>• <a href="#">STEMMA Connector</a></li><li>• <a href="#">MIDI Adapter</a></li><li>• <a href="#">Optional Fanciness if You're Feeling Extra</a></li></ul>	
<b>MIDI Keypad &amp; MIDI Arpeggiator</b>	<b>16</b>
<ul style="list-style-type: none"><li>• <a href="#">Download</a></li><li>• <a href="#">Install the Firmware</a></li><li>• <a href="#">Bootloader Mode</a></li><li>• <a href="#">Synth Connection</a></li><li>• <a href="#">MIDI Arpeggiator</a></li><li>• <a href="#">Customize with Arduino</a></li></ul>	
<b>Code with CircuitPython</b>	<b>20</b>
<ul style="list-style-type: none"><li>• <a href="#">CircuitPython Prep</a></li></ul>	

# Overview



Play your classic synthesizers using the NeoTrellis M4 by sending **MIDI** (Musical Instrument Digital Interface) data over UART (serial port) to a traditional MIDI DIN-5 connection. No USB MIDI required!

The MIDI standard is excellent for getting note and CC (and pitch bend, mod wheel, etc.) data from one device to another. Many modern controllers can only send USB MIDI, and lots of classic synths only have a DIN-5 connector. The Trellis M4 can send serial data over its UART port. With a little bit of plug adapter magic, we can get all of that data sent right on over to the standard DIN-5 MIDI cable!

## 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...

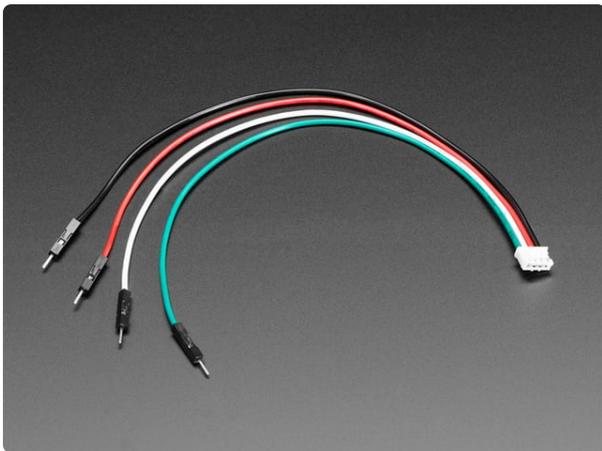
<https://www.adafruit.com/product/4020>



### 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 Male Header Cable - I2C STEMMA Cable - 200mm

This cable will let you turn a JST PH 4-pin cable port into 4 individual wires with high-quality 0.1" male header plugs on the end. We're carrying these to match up with any...

<https://www.adafruit.com/product/3955>



### 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>



### Monoprice 5-Watt Guitar Amplifier, Portable Recorder

Always be ready rock out or record with this 5-watt Guitar Amplifier, Portable Recorder, and USB Audio Interface from Monoprice. This handheld 5-watt amp...

<https://www.adafruit.com/product/4402>



### Stereo 3.5mm Plug/Plug Audio Cable - 6 feet

This basic cable comes with two 3.5mm (1/8" headphone jack size) stereo connectors. It's fairly straight forward, you'll commonly need these to connect two audio devices...

<https://www.adafruit.com/product/876>

In addition to the parts above, you'll need:

- Standard DIN-5 Male to DIN-5 male MIDI cable
- 3.5mm TRS Male to DIN-5 female MIDI adapter. Pinouts vary, unfortunately, as this was only recently standardized by the MIDI organization. We can use the now-standardized pinout of the "Type A" adapter, or you can simply swap two of the wires coming from the JST PH 4-pin cable to accommodate "Type B" connectors.



Here's a generic Type A adapter that works well (<https://adafru.it/D4Q>)



These Type B [Malekko Heavy Industry adapters](https://adafru.it/D4t) (<https://adafru.it/D4t>) work great, and they're super bright pink! At only \$6 for a two pack it's hard to beat the price.

And, if you're interested in this project, you'll probably have favorite synthesizer with a DIN-5 female MIDI-IN port on it! This covers most synthesizers built since 1983, so there are plenty to choose from.

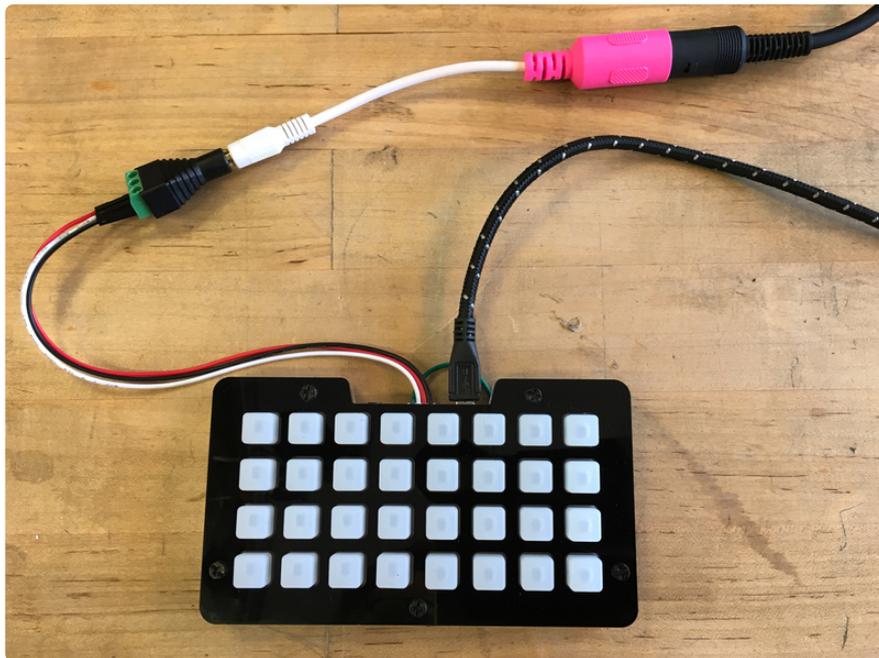


I'm using the Trellis M4 to control this modern all-analog minimoog "inspired" synthesizer, the Behringer Model D, which has no keyboard, but a MIDI IN jack designed for this type of use. It's highly affordable, and it sounds super thick and great.

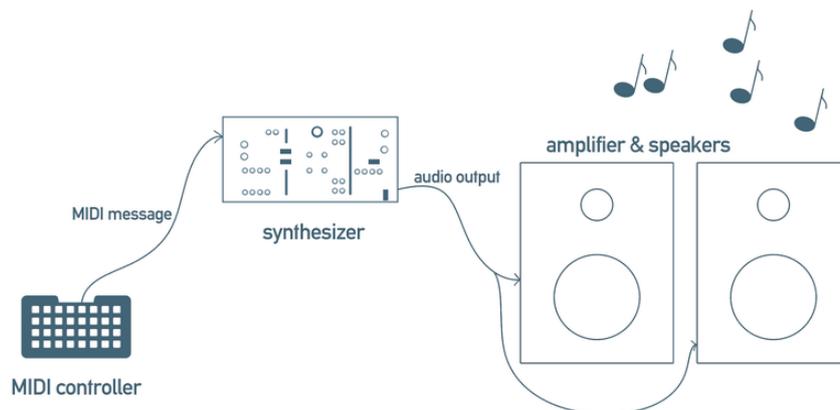


---

# MIDI Connections



**MIDI** is a venerable protocol (dating back to 1983) that is used to communicate between synthesizers, controllers, sequencers, sample players, computers, mobile devices, drum machines, and other electronic music making devices.



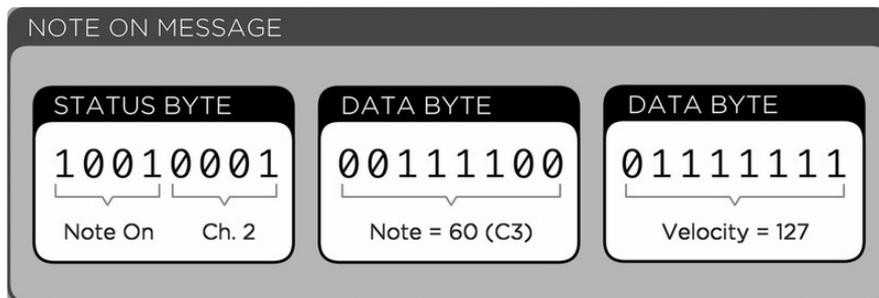
## Electronic Music Gear

Electronic music gear is often made up of multiple discrete components that each have their specialized task to perform. They can communicate with each other through data messages, control voltages, gate and trigger signals, and/or audio signals in order to come together as a whole, united system capable of being performed, recorded, played back, and ultimately, producing audible music!

# MIDI Messages

A simple and very common use case is to have a controller, such as a piano-style keyboard, send **Note On** and **Note Off** data to a music synthesizer (including software synths on your computer or mobile device).

Press a key and a message is sent telling the synth to play a specific musical note. Release the key and a message is sent to the synthesizer telling it to stop playing that note.



Getting a bit fancier than simple on/off messages, MIDI can also be used to send "continuous controller" **CC** messages, typically the result of turning a knob or pushing a slider on the keyboard controller. These can be used to sweep through the cutoff frequency of a low pass filter, or modulate a tremolo, and many, many other parameters.

A similar scheme is also used to send pitch bend info -- often represented by a pitch bend wheel on the left side of a MIDI keyboard.

Learn more about MIDI here <https://www.midi.org/articles/an-intro-to-midi>

## NeoTrellis M4 MIDI

We can use the NeoTrellis M4 to send any kind of MIDI message we like. [In a previous project, we showed how to send MIDI over USB to control software synthesizers \(https://adafru.it/D3L\)](#). Here we'll send MIDI the classic way, as a serial data stream over a DIN-5 connector that can control any traditional synth.

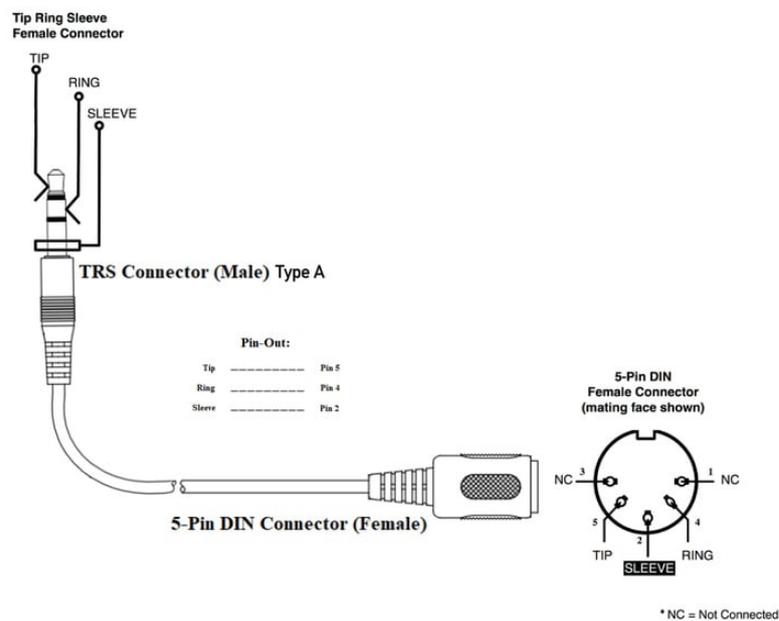
In this project, we've mapped the 32 buttons to play the "keys" on your synth, from a low C (MIDI Note 32) up to a high G (MIDI note 64). You can hold many buttons at once to create chords if you're using polyphonic synth software.

But that's not all! We can also play very expressive pitch bend and modulation with the NeoTrellis M4. This is all thanks to the Analog Device ADXL343 accelerometer built right onto the board! Tilt left and right to pitch bend down and up, and tilt forward and backward to gradually adjust the filter or tremolo or whatever you like on MIDI CC 1.

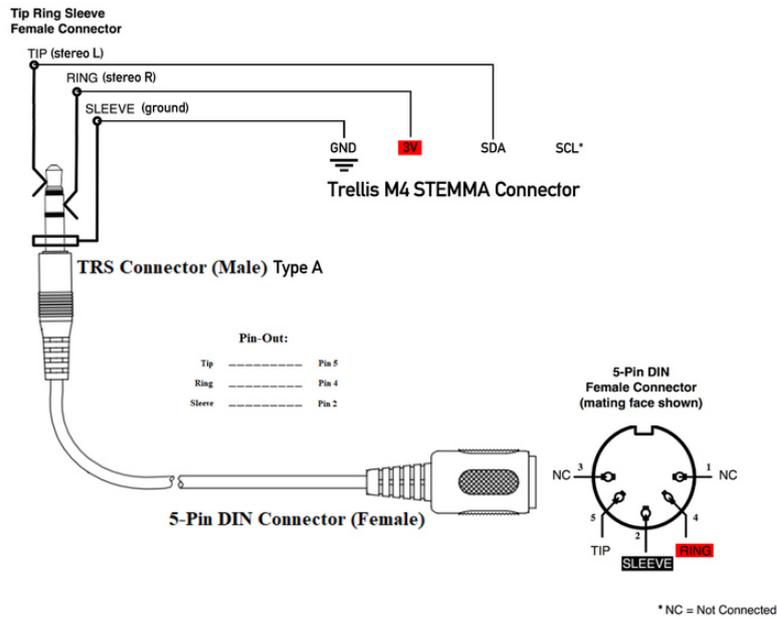
## Adapting MIDI to the 3.5mm TRS Plug

In an effort to provide classic MIDI on small devices, where the large DIN-5 connector is impractical, many manufacturers have started to use a three-conductor 3.5mm plug and jack with an adapter. The 3.5mm TRS (tip, ring, sleeve) connectors and wires are what we're familiar with from most stereo headphone cables. Since we only use three of the five conductors in a DIN-5 cable, this works out well. [This site has a great chart of compatibility \(https://adafru.it/D9p\)](https://adafru.it/D9p) for various manufacturers equipment and the different pinouts of TRS to DIN-5 connectors.

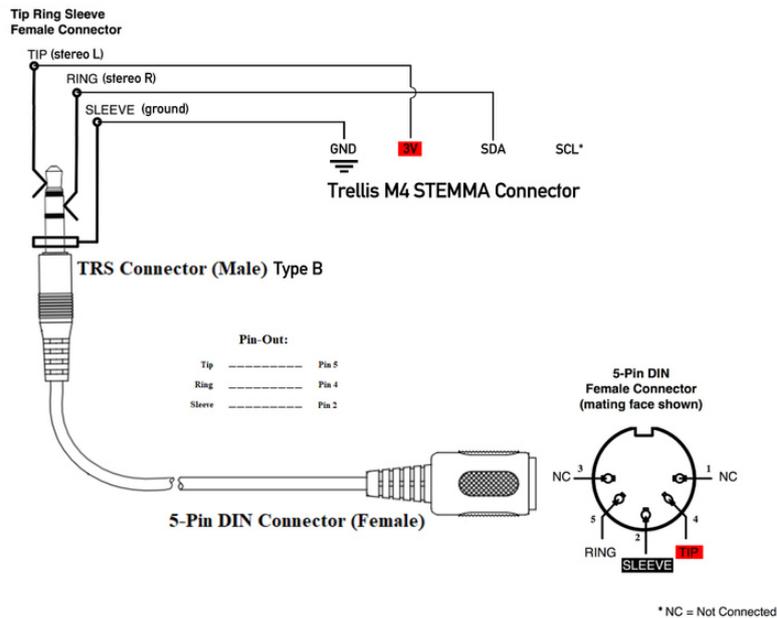
Here, you can see how the female 3.5mm TRS jack on one device can be plugged into a DIN-5 female MIDI jack on another.



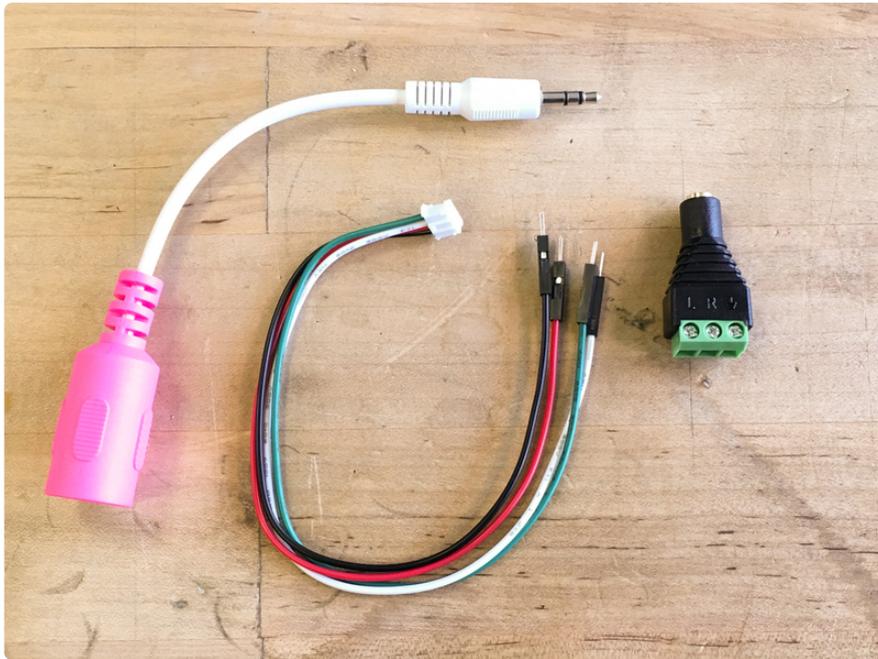
In our scenario, we'll connect the Trellis M4's UART connection to this adapter so that we can send MIDI messages out to a synthesizer with a MIDI-IN DIN-5 female jack. Here are the connections for a Type A cable:



On a Type B adapter the tip and ring connections are reversed, so we'll wire the Trellis M4 to the adapter like this:



The outputs of the M4 are current limited, and we're using only 3.3V logic, so the wiring works fine without adding resistors. BUT if you want to be in full compliance with the MIDI electrical specification, add a 47 Ohm resistor in series to the 3V and SDA lines.

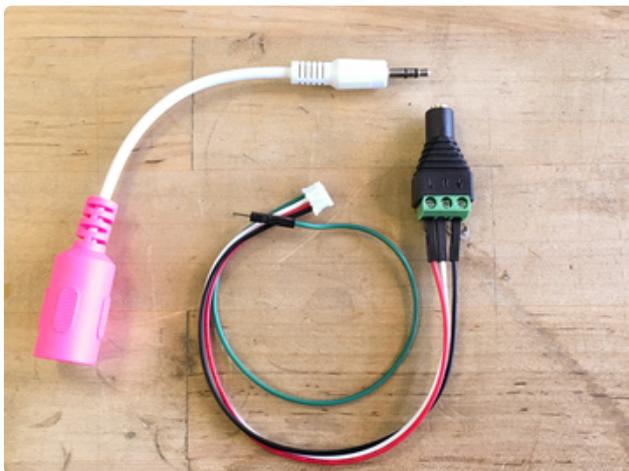


## TRS Adapter



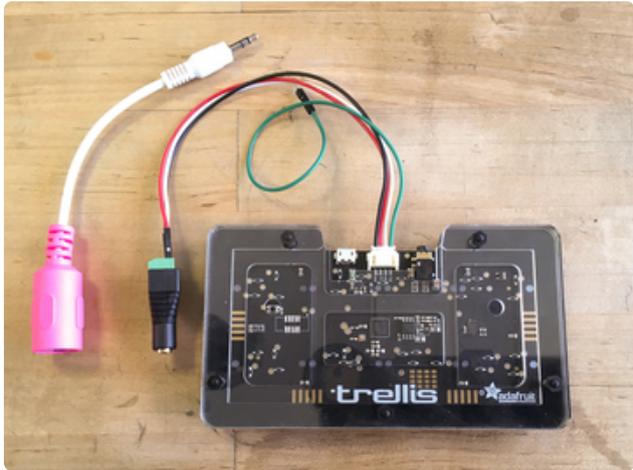
Use a small screwdriver to loosen the three screw terminals in the 3.5mm female TRS adapter. Then, insert the JST PH cable wires as shown:

Black to GND  
White to R  
Red to L



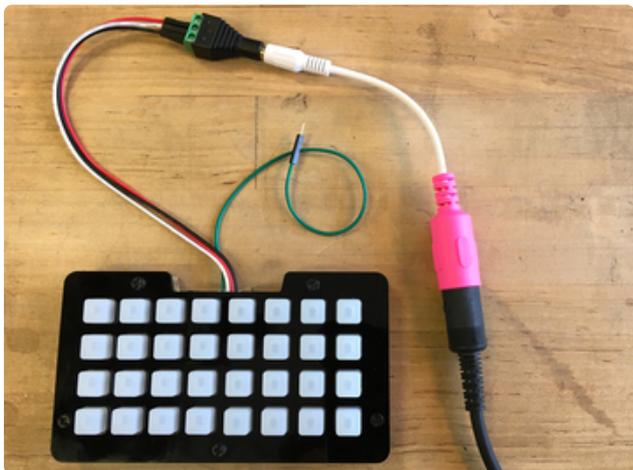
The green wire will not be connected to anything.

## STEMMA Connector

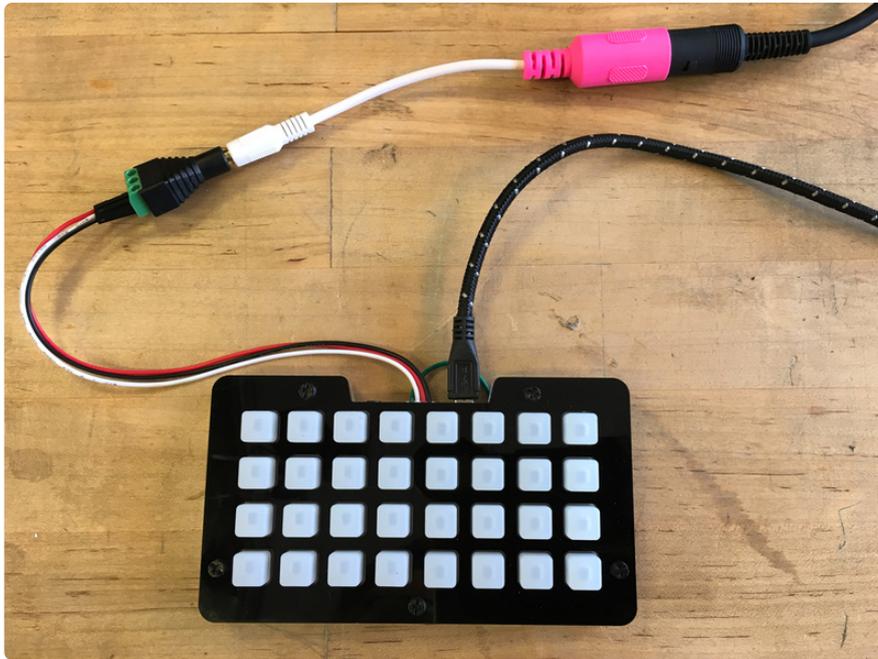


Next, plug the JST PH cable into the STEMMA connector on the NeoTrellis M4. This will allow the 3V, SDA (serial data), and GND connections to be made with the TRS adapter.

## MIDI Adapter



Now, plug the MIDI adapter into the TRS breakout jack. You can also plug a standard MIDI DIN-5 cable into the adapter.



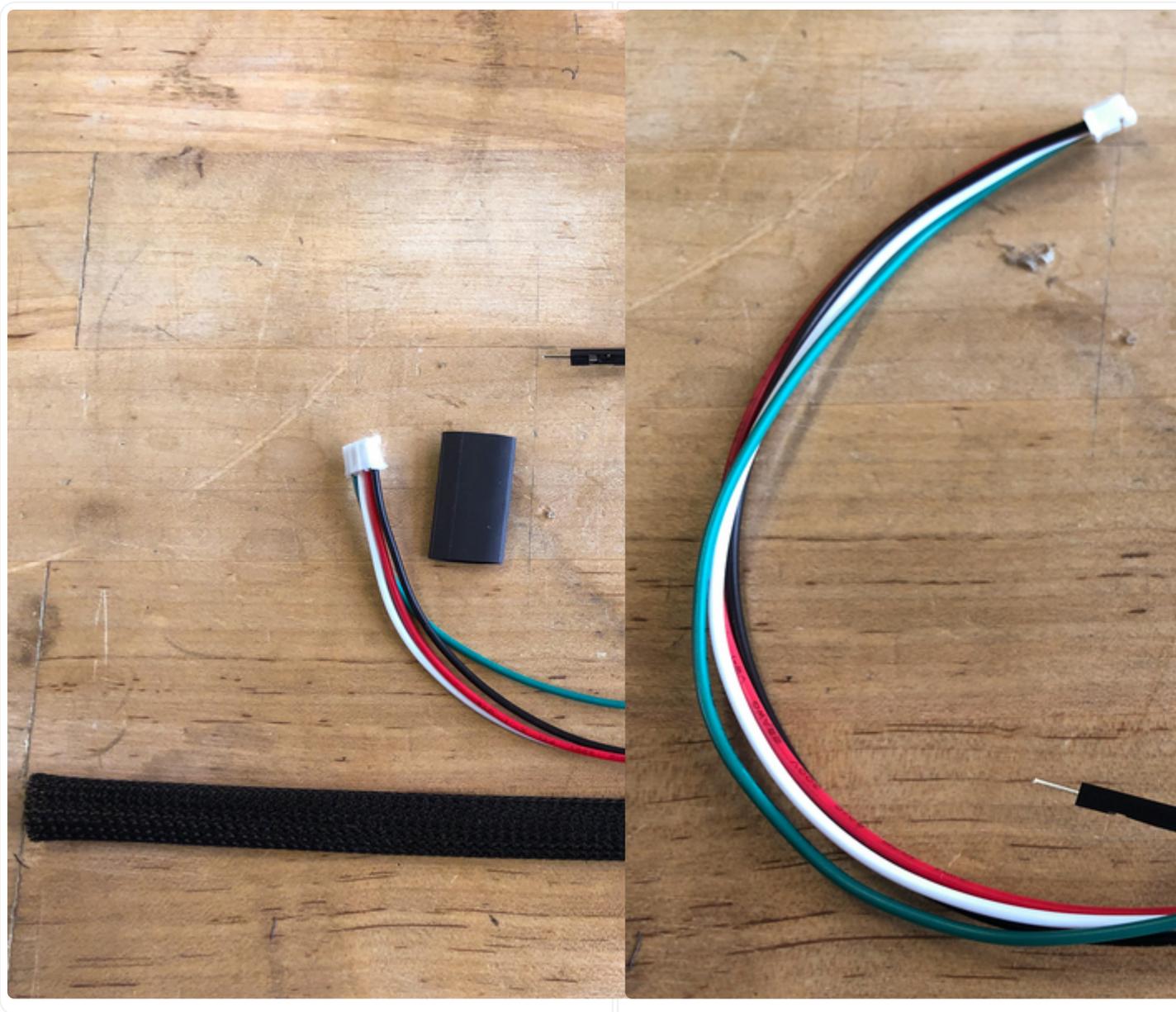
Next, we'll upload the MIDI player or MIDI arpeggiator firmwares onto the NeoTrellis M4 and play the synth!

## Optional Fanciness if You're Feeling Extra

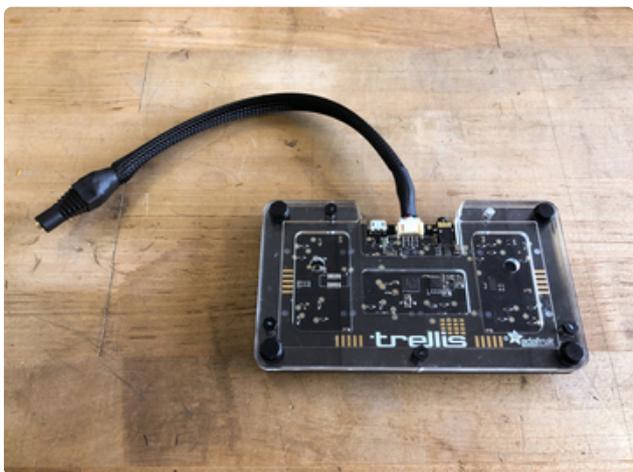
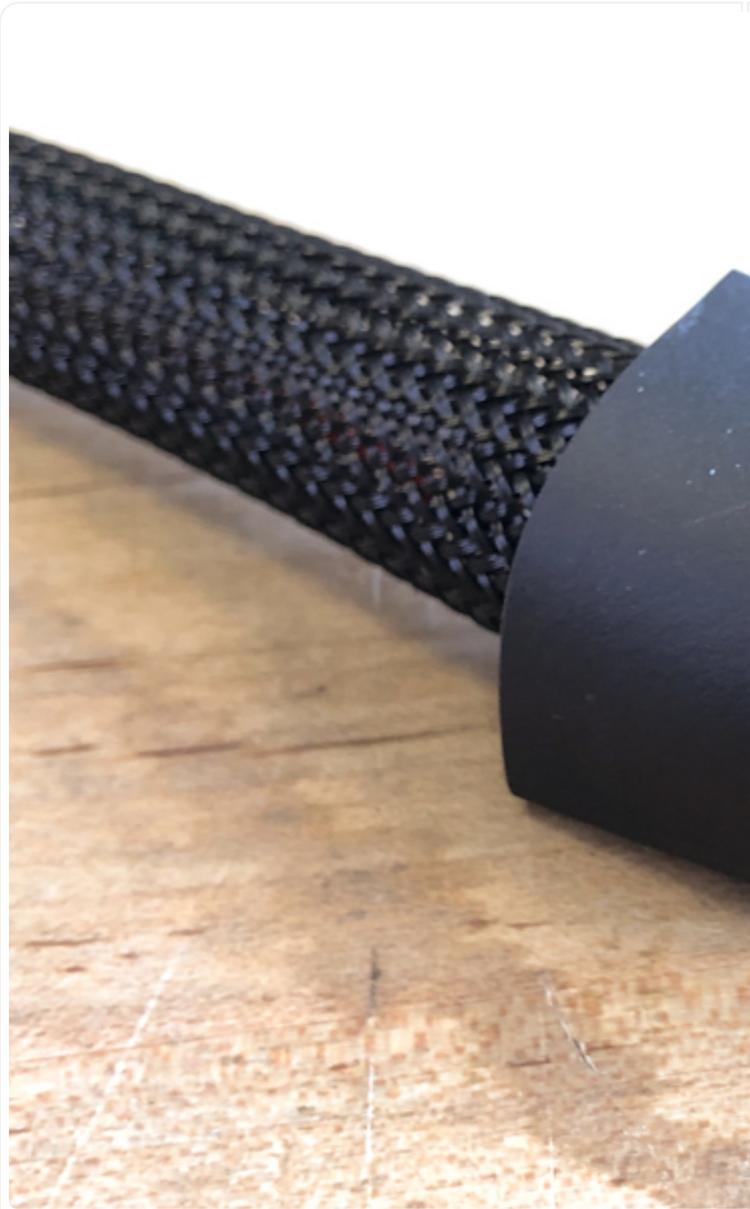


Want to neaten up your cabling job a bit so you can look the part when you play live at the club? You can get some woven cable sheathing and heat shrink tubing to take it up a notch!

Cut a length of woven sheathing, bend back the unused green wire, and slip the sheathing over the wires.



Use a few pieces of heat shrink tubing to secure each end.



Plug it in and you're ready to go!



---

## MIDI Keypad & MIDI Arpeggiator



The MIDI keypad can send note on/off messages as well as pitch bend and mod wheel CC (continuous controller) messages. This example code works on the NeoTrellis M4 with both USB MIDI over the micro USB port and classic MIDI over the UART serial port.

In order to make it simple to get up and running with your Trellis M4 Classic MIDI Controller **with no programming required**, we've created a drag-and-drop firmware you can use. You can skip to the page [Custom Code in Arduino \(https://adafru.it/D08\)](https://adafru.it/D08) if you want to customize things!

If you're just getting started with your NeoTrellis, there is lots more info [here in the main Learn Guide!](https://adafru.it/C-C) (<https://adafru.it/C-C>)

## Download

First, download the firmware file linked in the button below and save it to your computer hard drive somewhere you'll be able to find it, such as your **Downloads** folder.

MIDI Keypad Firmware

<https://adafru.it/D4u>

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

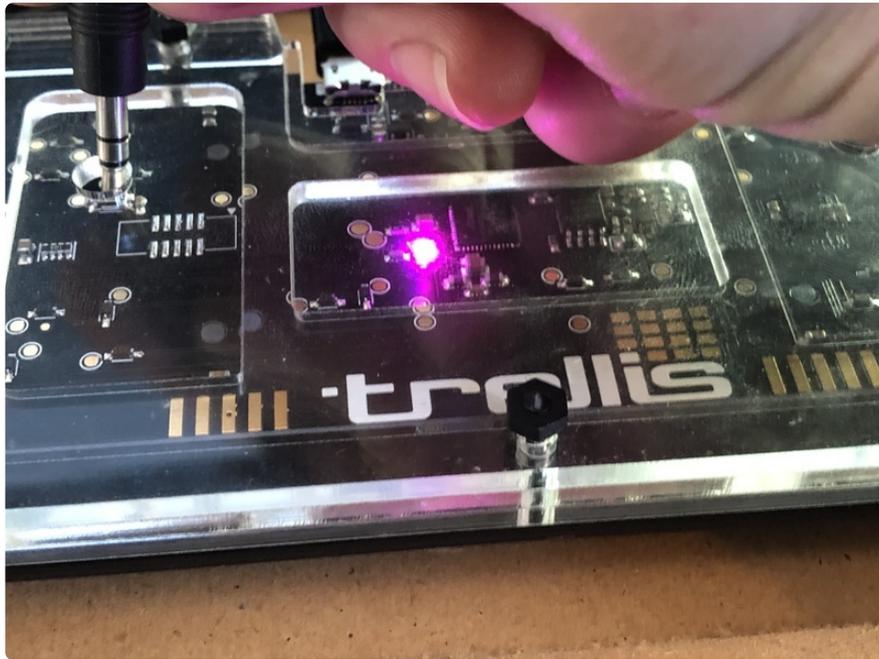
## Install the Firmware

Plug your NeoTrellis M4 into your computer with a good quality, data capable **USB** cable. Life is too short to go through the pain of accidentally using a **power-only** USB cable, so please round up any you own, cut them in half, travel to a distant land, bury them, and dance on their grave.

Your NeoTrellis will start up, and you'll see a purple indicator light on the back side.

## Bootloader Mode

Now, we'll put the NeoTrellis into "bootloader" mode. In this mode it will appear as a USB drive on your computer and will be ready to receive a new .uf2 firmware file. Use a thin, pointed object such as a headphone plug to double-click the reset button on the back side of the board.



## Synth Connection

Finally, plug your MIDI cable into your synthesizer MIDI-IN jack, power up the NeoTrellis via USB power, and play!



While playing, use the keypad as a chromatic keyboard, the forward/backward tilt as mod wheel control, and the side-to-side tilt as pitch bend.



## MIDI Arpeggiator

The MIDI arpeggiator example code works on the NeoTrellis with both USB MIDI over the micro USB port and classic MIDI over the UART serial port.

Install it this file below the same way as you did with the keypad firmware on the previous page.

## MIDI Arpeggiator Firmware

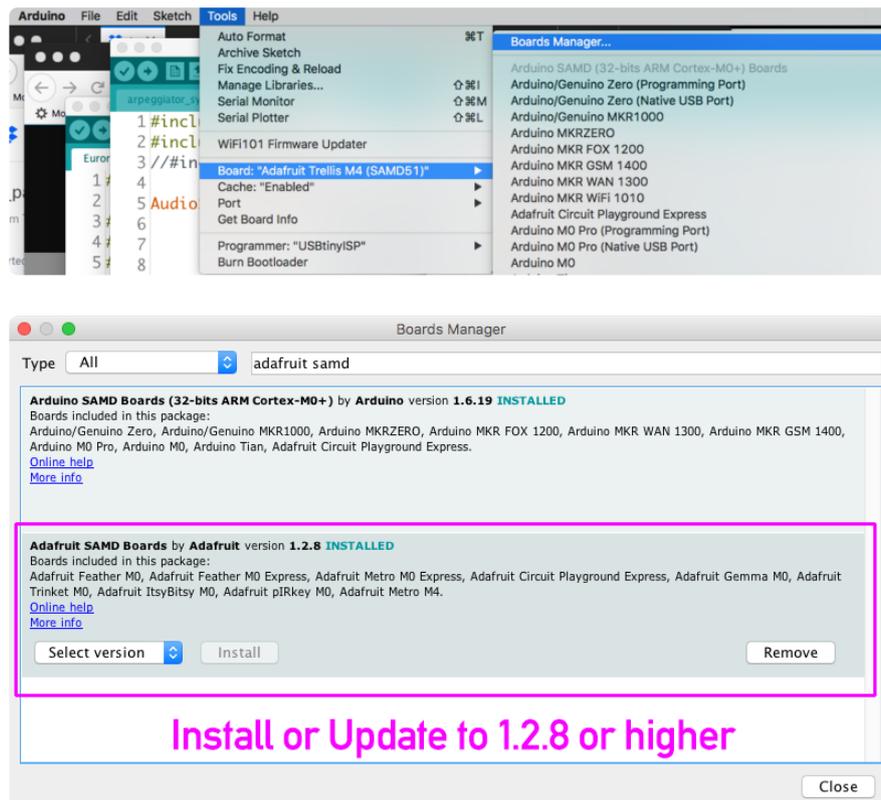
<https://adafru.it/D4v>

### Customize with Arduino

If you'd like to customize things further, you can edit the code in Arduino. Open your Arduino IDE. If you are new to Arduino, check out this guide: [Adafruit Arduino IDE Setup \(https://adafru.it/CMS\)](https://adafru.it/CMS).

[Head here \(https://adafru.it/CW5\)](https://adafru.it/CW5) to download the **Adafruit\_NeoTrellisM4** library and example code. You can learn more about Arduino libraries by [reading this guide from Adafruit \(https://adafru.it/dit\)](https://adafru.it/dit).

You'll also need to make sure you've updated your Board Definitions to version 1.2.8 or higher of the SAMD boards by heading to **Tools -> Board Manager** in the **Arduino IDE**, filter for "Adafruit SAMD" and choose the install shown below.



### Code with CircuitPython

Let's make a simple MIDI note player in CircuitPython. This will give you a good idea of how to send MIDI data over the NeoTrellis M4's UART/serial port, which you can

then adapt to make your own arpeggiators, sequencers, modal players, chord keyboards for polyphonic synths, and more.

[CircuitPython](https://adafru.it/tB7) (<https://adafru.it/tB7>) is designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the **CIRCUITPY** flash drive to iterate.

Are you new to using CircuitPython? No worries, [there is a full getting started guide here](https://adafru.it/cpy-welcome) (<https://adafru.it/cpy-welcome>).

Adafruit suggests using the Mu editor to edit your code and have an interactive REPL in CircuitPython. [You can learn about Mu and its installation in this tutorial](https://adafru.it/ANO) (<https://adafru.it/ANO>).

First, let's prep the board for that.

## CircuitPython Prep

To get prep the Trellis M4 to run the sequencer code, follow these steps:

- Update the [bootloader for NeoTrellis](https://adafru.it/C-N) (<https://adafru.it/C-N>) from the Trellis M4 guide.
- Install the [latest CircuitPython for NeoTrellis](https://adafru.it/C-O) (<https://adafru.it/C-O>) from the Trellis M4 guide.

Click the Download Project Bundle to get the code, data files and library files.

```
# SPDX-FileCopyrightText: 2018 Dave Astels for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""
NeoTrellis M4 Express MIDI synth

Adafruit invests time and resources providing this open source code.
Please support Adafruit and open source hardware by purchasing
products from Adafruit!

Written by Dave Astels for Adafruit Industries
Copyright (c) 2018 Adafruit Industries
Licensed under the MIT license.

All text above must be included in any redistribution.
"""

import os
import parser
import sequencer
import synth
```

```

import adafruit_trellism4

trellis = adafruit_trellism4.TrellisM4Express(rotation=0)
trellis.pixels.brightness = 0.1
trellis.pixels.fill(0)

syn = synth.Synth()
seq = sequencer.Sequencer(syn)
p = parser.MidiParser()

voices = sorted([f.split('.')[0] for f in os.listdir('/samples')
                 if f.endswith('.txt') and not f.startswith('.')])
print('Voices found: ', voices)
tunes = sorted([f for f in os.listdir('/midi')
                if f.endswith('.mid') and not f.startswith('.')])
print('Midi files found: ', tunes)

selected_voice = None

def reset_voice_buttons():
    for i in range(len(voices)):
        trellis.pixels[i, 0] = 0x0000FF

def reset_tune_buttons():
    for i in range(len(tunes)):
        trellis.pixels[(i % 8, (i // 8) + 1)] = 0x00FF00

current_press = set()
reset_voice_buttons()
reset_tune_buttons()

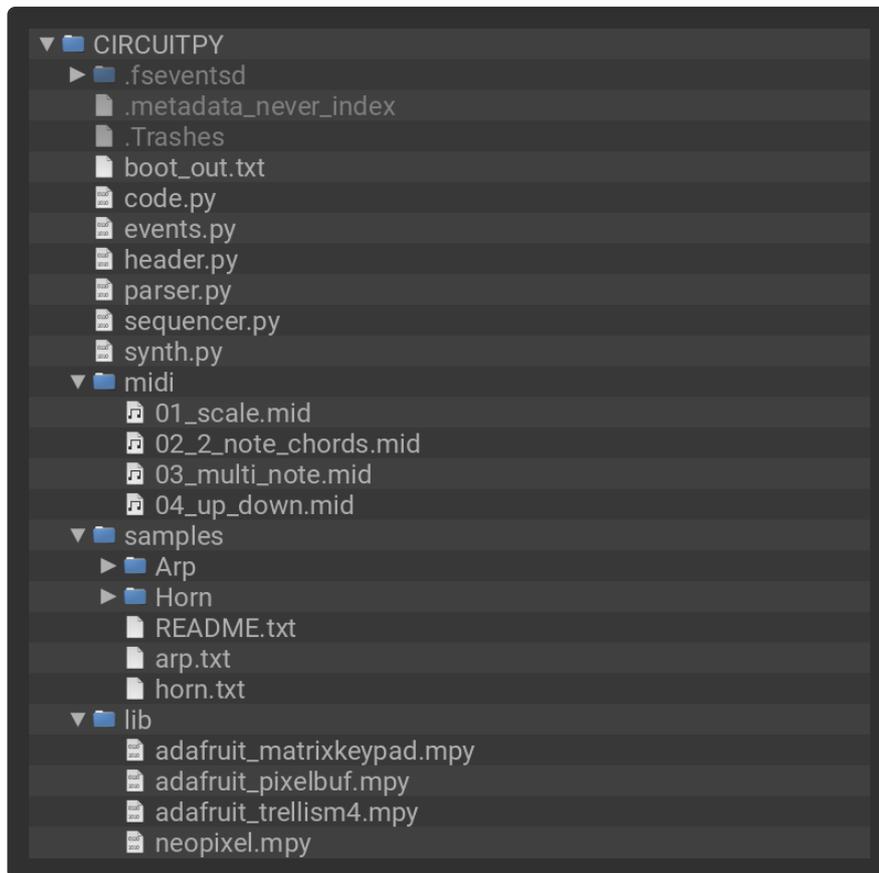
while True:
    pressed = set(trellis.pressed_keys)
    just_pressed = pressed - current_press
    for down in just_pressed:
        if down[1] == 0:
            if down[0] < len(voices):
                selected_voice = down[0]
                reset_voice_buttons()
                trellis.pixels[down] = 0xFFFFFF
                syn.voice = voices[selected_voice]
            else:
                tune_index = (down[1] - 1) * 8 + down[0]
                if tune_index < len(tunes) and selected_voice is not None:
                    trellis.pixels[down] = 0xFFFFFF
                    header, tracks = p.parse('/midi/' + tunes[tune_index])
                    for track in tracks:
                        seq.play(track)
                    reset_tune_buttons()

    current_press = pressed

```

Unzip the project bundle file on your computer and using File Explorer or Finder (depending on your operating system) copy the files over to the NeoTrellis **CIRCUITPY** drive using a known good USB data+power cable.

Your NeoTrellis **CIRCUITPY** drive should have the following files on it. Be sure the directory structure is correct and all files are present.



Hook up the NeoTrellis M4 to your synthesizer over the UART-to-MIDI adapter cabling and now when you press a button, you'll send a **MIDI note on** message, and when you release a button, it'll send a **MIDI note off**.

This can serve as your launching off point for more elaborate CircuitPython MIDI projects!