



Trellis M4 Expressive MIDI Controller

Created by John Park



<https://learn.adafruit.com/trellis-m4-midi-keypad-and-tilt-controller>

Last updated on 2024-03-08 03:12:52 PM EST

Table of Contents

Overview	3
<ul style="list-style-type: none">• Parts Used• Assembly and Familiarization	
MIDI Essentials	5
<ul style="list-style-type: none">• Electronic Music Gear• MIDI Messages• Trellis M4 MIDI	
Firmware Upload	7
<ul style="list-style-type: none">• Download• Install the Firmware• Bootloader Mode• Drag and Drop• Test MIDI Output	
Software Synthesizers	11
<ul style="list-style-type: none">• Synthesizers• Desktop/Laptop	
Play Music with Helm Synthesizer	17
<ul style="list-style-type: none">• Play Music• Assign CC	
Using with iOS Devices	19
<ul style="list-style-type: none">• Using with iOS Devices	
Custom Code in Arduino	23
<ul style="list-style-type: none">• MIDI Channel• Note Range• CC Channel• Test MIDI Output	

Overview



You can play music with the **Trellis M4** by turning it into a great input device for synthesizers! It can send standard **MIDI** (Musical Instrument Digital Interface) note data over USB MIDI to a software synth on your computer, laptop, (including web browser based synths!), or mobile device.

Using the built-in accelerometer (thanks to [Analog Devices for the ADXL343 \(https://adafruit.it/CVU\)](https://adafruit.it/CVU)!), we can also send control change (CC) data, such as mod wheel, and filter cutoff values, and even pitch bend settings just by tilting the Trellis M4!

You can even use the Trellis M4 to control hardware synthesizers by using a conversion device that acts as a USB MIDI host and sends out traditional MIDI over a 5-pin DIN cable or more modern TRS 3.5mm MIDI connection. It's even possible to go really old school and convert to gate and CV (control voltage) for a modular synth or pre-MIDI device..

Parts Used



[Adafruit NeoTrellis M4 with Enclosure and Buttons Kit Pack](https://www.adafruit.com/product/4020)

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>



[iOS Lightning to USB OTG Cable](https://www.adafruit.com/product/3940)

Your iOS phone or tablet may not have a USB port on the bottom but that doesn't mean you can't use it to connect USB devices. Secretly known as a 'Camera Connector' or...

<https://www.adafruit.com/product/3940>



[USB Patterned Fabric Cable - A/MicroB](https://www.adafruit.com/product/2008)

Oooh it's so soft! And that beautiful pattern! If you have to have visible cables then you might as well have the nicest fabric bound cable with a eye-catching snake-like...

<https://www.adafruit.com/product/2008>



Little Rubber Bumper Feet - Pack of 4

Keep your electronics from going barefoot, give them little rubber feet! These small sticky bumpers are our favorite accessory for any electronic kit or device. They are sticky, but...

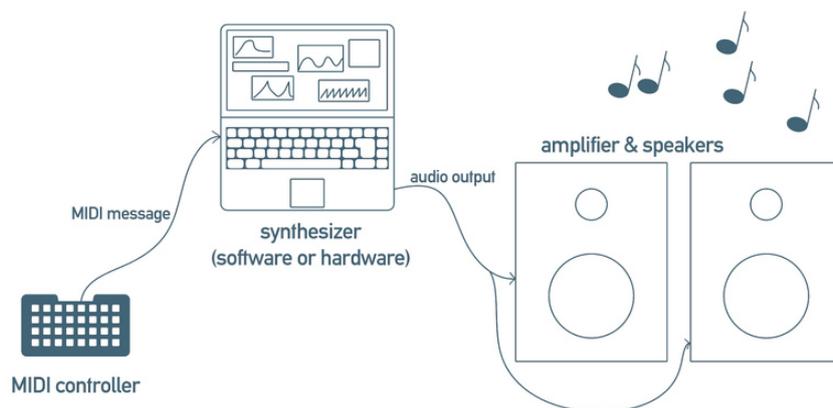
<https://www.adafruit.com/product/550>

Assembly and Familiarization

Take a look at the [Trellis M4 main guide \(https://adafru.it/C-C\)](https://adafru.it/C-C) for lots more info and the [assembly page \(https://adafru.it/D0j\)](https://adafru.it/D0j) in case you haven't yet built the enclosure.

MIDI Essentials

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.



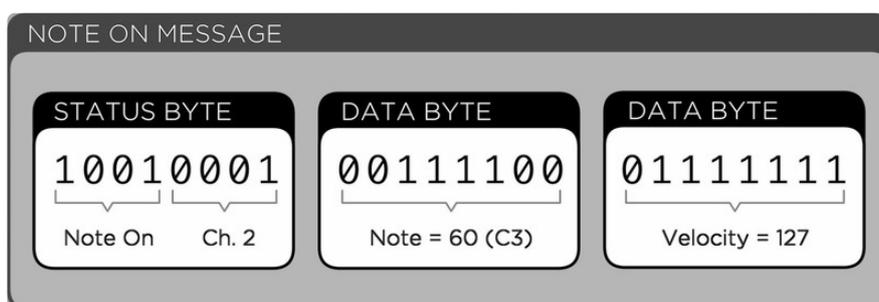
Electronic Music Gear

Electronic music gear is often made up of multiple discreet 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.

Trellis M4 MIDI

We can use the Trellis M4 to send any kind of MIDI message we like, right over USB. 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 Trellis 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.

Firmware Upload



In order to make it simple to get up and running with your Trellis M4 Expressive 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 Trellis M4, 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.

Trellis M4 MIDI CC.UF2 firmware

<https://adafru.it/C-D>

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

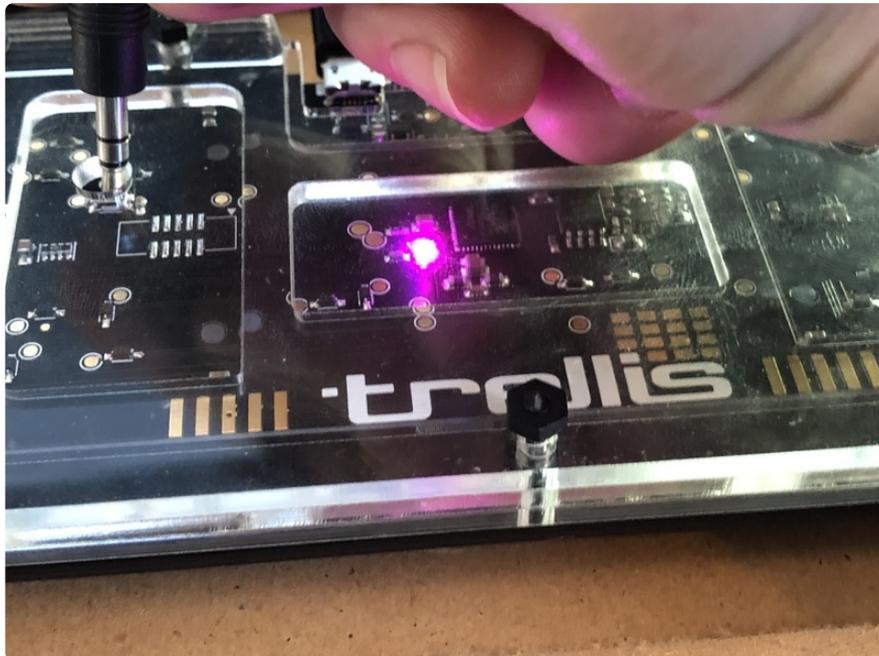
Install the Firmware

Plug your Trellis 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 Trellis M4 will start up, and you'll see a purple indicator light on the back side.

Bootloader Mode

Now, we'll put the Trellis M4 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.

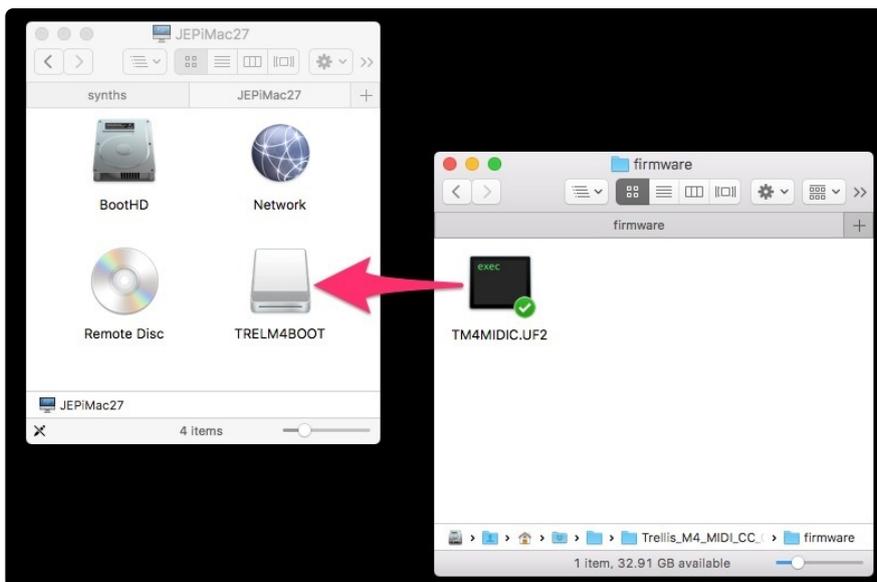




Once you have double-clicked the reset button, the indicator LED will turn green. You'll notice a new USB drive appear on your computer named **TRELM4BOOT**, this is the bootloader USB storage built right into the Trellis M4. It is now ready to receive the firmware file.

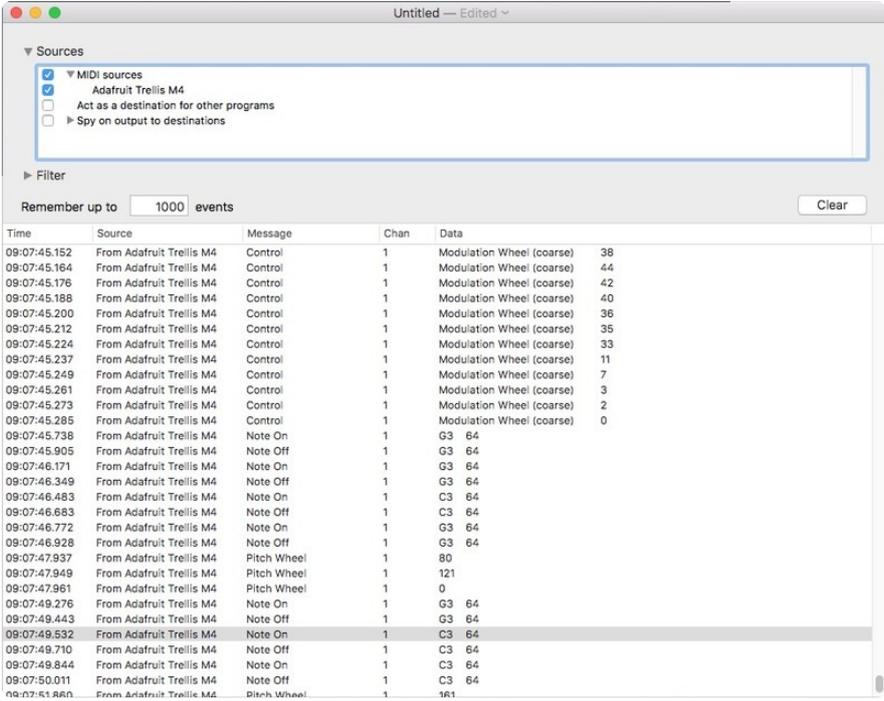
Drag and Drop

Now, drag the **TM4MIDIC.UF2** file onto the **TRELM4BOOT** drive. The file will copy over in a few seconds and then the Trellis M4 will automatically restart itself (you'll see the **TRELM4BOOT** drive disappear, don't worry, this is normal!). The status LED will return to purple, indicating regular operation mode.



Congratulations, you've updated the firmware and you're ready to play!

If you ever need to manually switch out of bootloader mode, simply press the reset button one time.



Time	Source	Message	Chan	Data
09:07:45.152	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 38
09:07:45.164	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 44
09:07:45.176	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 42
09:07:45.188	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 40
09:07:45.200	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 36
09:07:45.212	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 35
09:07:45.224	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 33
09:07:45.237	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 11
09:07:45.249	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 7
09:07:45.261	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 3
09:07:45.273	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 2
09:07:45.285	From Adafruit Trellis M4	Control	1	Modulation Wheel (coarse) 0
09:07:45.738	From Adafruit Trellis M4	Note On	1	G3 64
09:07:45.905	From Adafruit Trellis M4	Note Off	1	G3 64
09:07:46.171	From Adafruit Trellis M4	Note On	1	G3 64
09:07:46.349	From Adafruit Trellis M4	Note Off	1	G3 64
09:07:46.483	From Adafruit Trellis M4	Note On	1	C3 64
09:07:46.683	From Adafruit Trellis M4	Note Off	1	C3 64
09:07:46.772	From Adafruit Trellis M4	Note On	1	G3 64
09:07:46.928	From Adafruit Trellis M4	Note Off	1	G3 64
09:07:47.937	From Adafruit Trellis M4	Pitch Wheel	1	80
09:07:47.949	From Adafruit Trellis M4	Pitch Wheel	1	121
09:07:47.951	From Adafruit Trellis M4	Pitch Wheel	1	0
09:07:49.276	From Adafruit Trellis M4	Note On	1	G3 64
09:07:49.443	From Adafruit Trellis M4	Note Off	1	G3 64
09:07:49.532	From Adafruit Trellis M4	Note On	1	C3 64
09:07:49.710	From Adafruit Trellis M4	Note Off	1	C3 64
09:07:49.844	From Adafruit Trellis M4	Note On	1	C3 64
09:07:50.011	From Adafruit Trellis M4	Note Off	1	C3 64
09:07:51.860	From Adafruit Trellis M4	Pitch Wheel	1	161

Test MIDI Output

You can test to see that your Trellis M4 is outputting MIDI signals properly by using a MIDI utility. These will report all activity coming from any MIDI controller plugged into your system. Here are some to try:

- [MIDI Monitor \(https://adafru.it/rBH\)](https://adafru.it/rBH) for mac os
- [MIDI-OX \(https://adafru.it/nkF\)](https://adafru.it/nkF) for Windows
- [MIDI Wrench \(https://adafru.it/C-1\)](https://adafru.it/C-1) for iOS
- [MIDI Scope \(https://adafru.it/C-2\)](https://adafru.it/C-2) for Android
- [KMidimon \(https://adafru.it/rBI\)](https://adafru.it/rBI) for Linux
- [Web MIDI Monitor \(https://adafru.it/C-3\)](https://adafru.it/C-3) for Chrome browser

Software Synthesizers

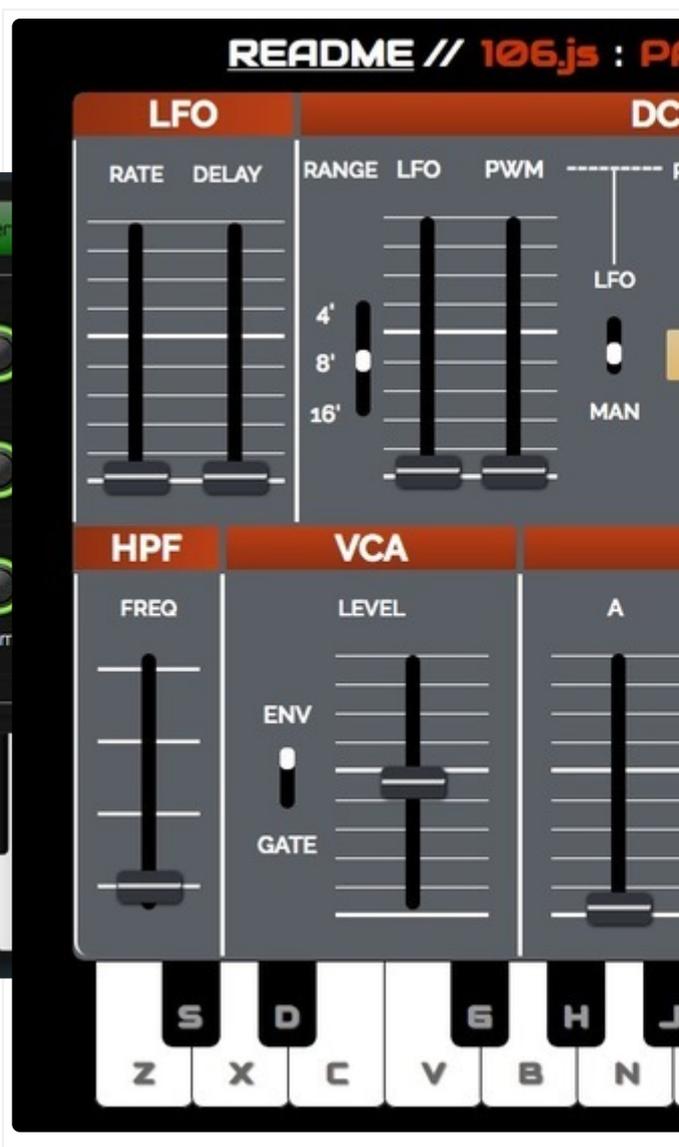


There are many, many, many, many (did we mention "many"?), synthesizers out there in the world that you can control using MIDI note and CC controls.

You may already have a favorite software synth, and chances are it'll work with the Trellis M4. In case you don't have one already picked out, here are some good ones to try.

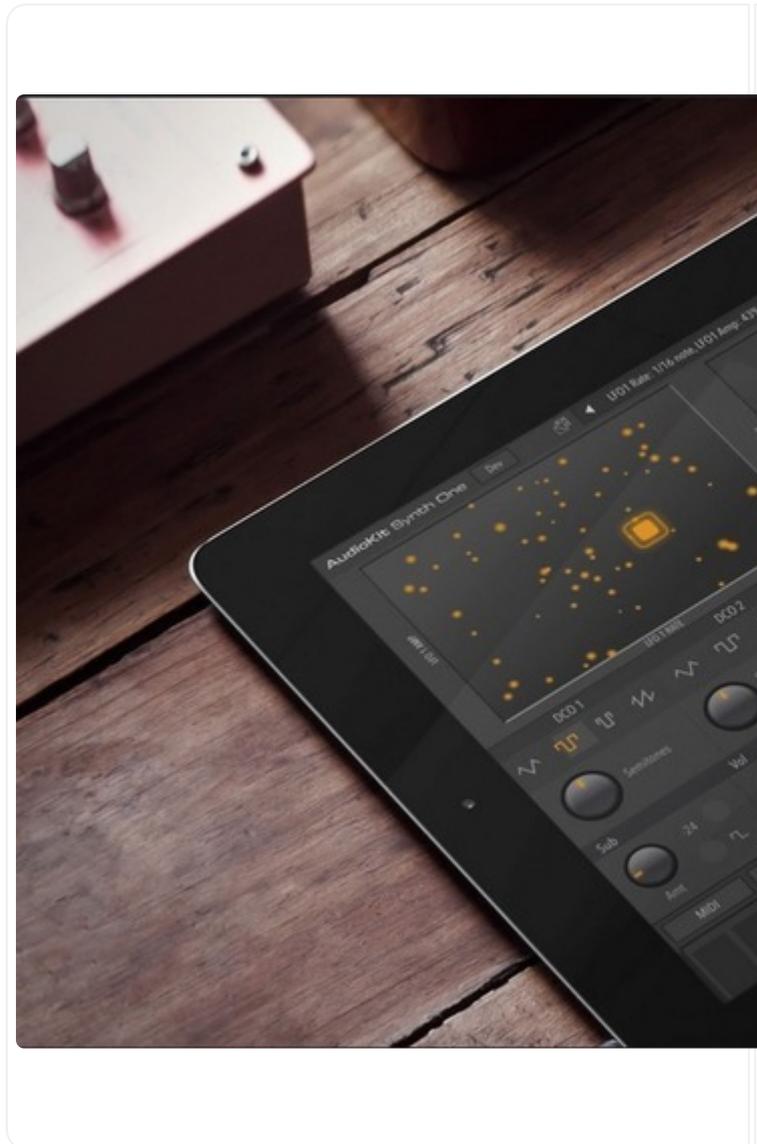
Synthesizers

Nearly any software synth (or DAW (digital audio workstation) you find will allow you to use USB MIDI to control the notes, pitch bend, and CC input. Here are some good ones to try:



Chrome Web Browser

- Viktor NV-1 <https://nicroto.github.io/viktor/> (<https://adafru.it/C-4>)
- Juno-106.js <http://juno-106.js.org/> (<https://adafru.it/C-5>)



iOS

free open source

- [AudioKit Synth One \(https://adafru.it/C-6\)](https://adafru.it/C-6) (iPad only)

free

- [GarageBand \(https://adafru.it/C-7\)](https://adafru.it/C-7)
- [DRC Polyphonic Synth by Imaginado \(https://adafru.it/C-8\)](https://adafru.it/C-8) (iPhone and iPad)



Android

free

- [DRC Polyphonic Synth by Imaginado \(https://adafru.it/C-9\)](https://adafru.it/C-9)

Desktop/Laptop



Linux / Windows / mac os

free open source

- [Helm](https://adafru.it/C-a) (https://adafru.it/C-a)
- [VCV Rack](https://adafru.it/C-b) (https://adafru.it/C-b)
- [Pure Data](https://adafru.it/C-c) (https://adafru.it/C-c)
- [Ardour](https://adafru.it/C-d) (https://adafru.it/C-d)

Windows / mac os

commercial

- [Ableton Live](https://adafru.it/C-f) (https://adafru.it/C-f)

- [Propellerhead Reason \(https://adafru.it/C-g\)](https://adafru.it/C-g)
- [Cubase \(https://adafru.it/C-h\)](https://adafru.it/C-h)
- [Fruity Loops FL Studio \(https://adafru.it/C-i\)](https://adafru.it/C-i)



Windows

free

- [PreSonus Studio One \(https://adafru.it/C-e\)](https://adafru.it/C-e)



mac os

free

- [GarageBand \(https://adafru.it/Ctp\)](https://adafru.it/Ctp)

commercial

- [Logic Pro \(https://adafru.it/C-j\)](https://adafru.it/C-j)

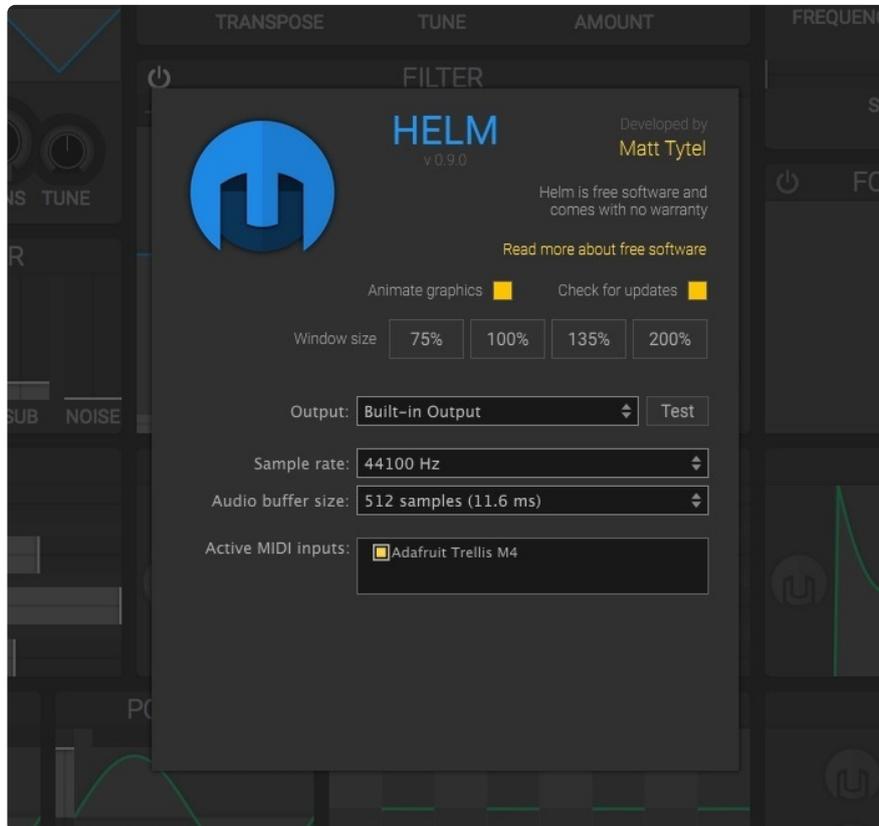
Play Music with Helm Synthesizer

Play Music

Once you've picked a synth (the Chrome browser ones are really quick to try out, hint, hint!) plug in your Trellis M4 and get playing! It's very likely the synth will recognize it

immediately, but if not, check the preferences and choose the "Adafruit Trellis M4" MIDI device.

Here's an example using Helm.



Assign CC

The note on/off functions will work immediately, as should the pitch bend if it's supported in your software. Try holding some notes and then tip the controller left and right to bend down and up.

The forward and backward tilt will send CC 1 messages, which is usually interpreted as "mod wheel". This can be mapped by default to things like tremelo, but you may want to change it. In most software synths it's a matter of checking the help docs and reassigning it.

For example, here's how to re-assign a MIDI mapping for the x-axis tilt in Helm so that it will control Filter Cutoff Frequency:

And, here it is in action!

Enjoy playing with your new Trellis M4 Expressive MIDI Controller! If you want to further customize things, check out the page on coding it in Arduino.

Using with iOS Devices

Using with iOS Devices

Using the Trellis M4 with an iOS synthesizer, such as the excellent AudioKit Synth One, is easy, but you will need an adapter to connect the USB output of the Trellis M4 to the Lightning input of the iOS device.

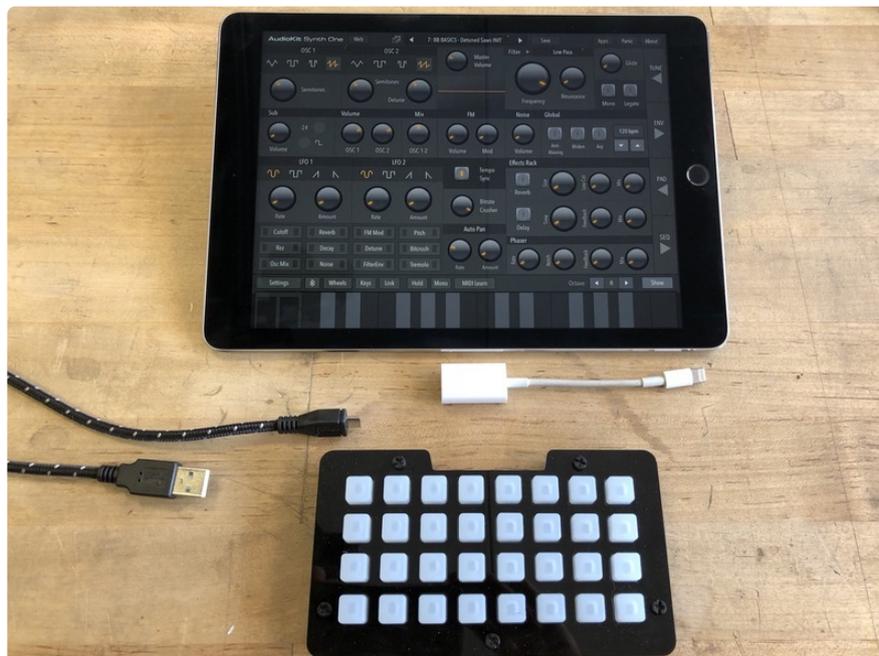
The [iOS Lightning to USB OTG cable](http://adafru.it/3940) (<http://adafru.it/3940>) allows you to connect the Trellis M4 right to your iPad, iPhone, or iPod Touch. The device will also provide power to your Trellis M4!



iOS Lightning to USB OTG Cable

Your iOS phone or tablet may not have a USB port on the bottom but that doesn't mean you can't use it to connect USB devices. Secretly known as a 'Camera Connector' or...

<https://www.adafruit.com/product/3940>







Depending on your software, you may need to select and configure the Trellis M4 as the chosen MIDI device.

Synth One found it automatically, but you can use the MIDI Learn function to assign the x-axis (forward/backward) tilt to any knob.



Press the "MIDI Learn" button

Tap on the knob you want to control -- Here I've chosen the Filter cutoff frequency



Tilt the Trellis M4 forward and backward. The range doesn't matter, it just needs to register that that MIDI CC number (1 in this case) is the once changing

Press the "MIDI Learn" button to finish



Now, as you tilt the Trellis M4, the Filter cutoff frequency changes! Have fun exploring different parameters as you play.



Custom Code in Arduino

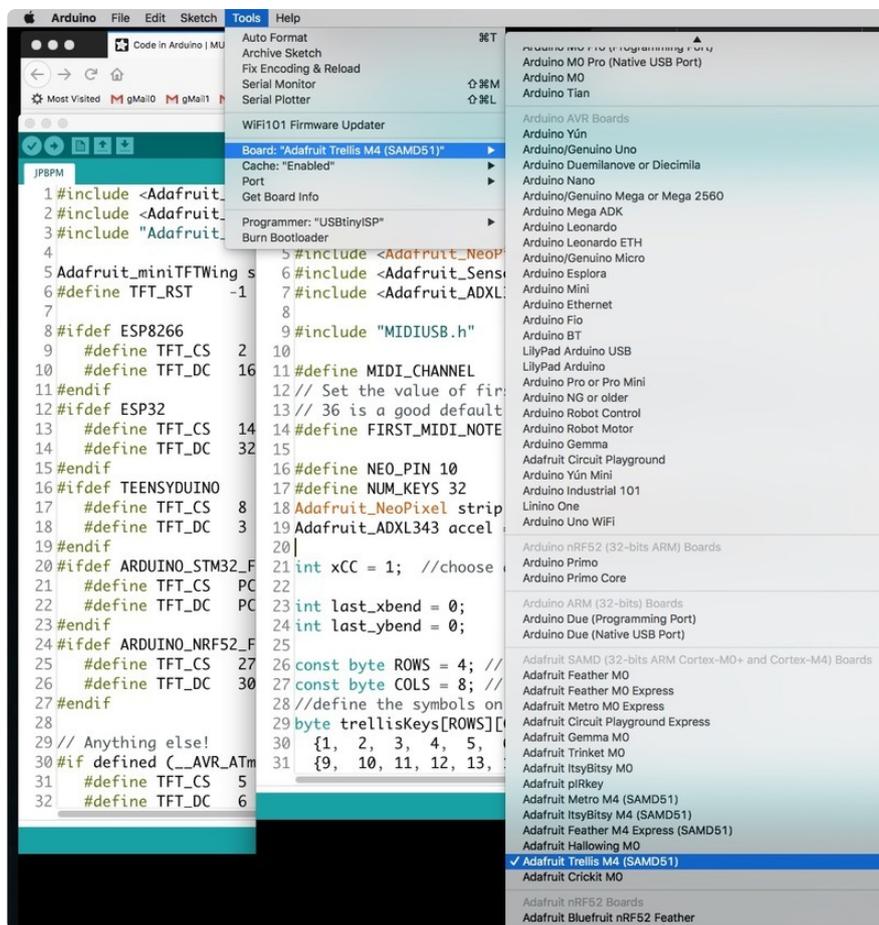
If you want to customize the Trellis M4 Expressive MIDI Controller beyond the .uf2 firmware you dragged onto the board, we'll need to use Arduino IDE to code it.

First, [get set up with Arduino IDE as detailed here \(https://adafru.it/D09\)](https://adafru.it/D09).

Next, [follow these instructions \(https://adafru.it/D0a\)](https://adafru.it/D0a) on getting the proper libraries installed. In this code we use the following libraries:

- Adafruit_Keypad
- Adafruit_NeoPixel
- Adafruit_Sensor
- Adafruit_ADXL343
- MIDIUSB

Once you've updated the board definitions as shown, you'll be able to select **Adafruit Trellis M4 (SAM51)** as your board for compiling and uploading.



Copy the code shown below and paste it into a new sketch in Arduino. Save the sketch as **midi_keypad.ino** and then upload it to your board.

Note: if you have any problem uploading to the board, double-click the reset button on the back of the board during the upload to set the Trellis M4 into bootloader mode. The on-board status LED (on the back) will be purple during normal running mode and switch to green when it is in bootloader mode.

```

// Trellis M4 MIDI Keypad CC
// sends 32 notes, pitch bend & a CC from accelerometer tilt over USB MIDI

#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL343.h>
#include <Adafruit_NeoTrellisM4.h>

#define MIDI_CHANNEL      0 // default channel # is 0
// Set the value of first note, C is a good choice. Lowest C is 0.
// 36 is a good default. 48 is a high range. Set to 24 for a bass machine.
#define FIRST_MIDI_NOTE 36

Adafruit_ADXL343 accel = Adafruit_ADXL343(123, &Wire1);

int xCC = 1; //choose a CC number to control with x axis tilting of the board. 1
is mod wheel, for example.

int last_xbend = 0;
int last_ybend = 0;

Adafruit_NeoTrellisM4 trellis = Adafruit_NeoTrellisM4();

void setup(){
  Serial.begin(115200);
  //while (!Serial);
  Serial.println("MIDI keypad & pitchbend!");

  trellis.begin();
  trellis.setBrightness(80);

  // USB MIDI messages sent over the micro B USB port
  Serial.println("Enabling MIDI on USB");
  trellis.enableUSBMIDI(true);
  trellis.setUSBMIDIchannel(MIDI_CHANNEL);
  // UART MIDI messages sent over the 4-pin STEMMA connector (3.3V logic)
  Serial.println("Enabling MIDI on UART");
  trellis.enableUARTMIDI(true);
  trellis.setUARTMIDIchannel(MIDI_CHANNEL);

  if(!accel.begin()) {
    Serial.println("No accelerometer found");
    while(1);
  }
}

void loop() {
  // put your main code here, to run repeatedly:
  trellis.tick();

  while (trellis.available()){
    keypadEvent e = trellis.read();
    int key = e.bit.KEY;
    Serial.print("Keypad key: ");    Serial.println(key);
    Serial.print("MIDI note: ");    Serial.println(FIRST_MIDI_NOTE+key);

    if (e.bit.EVENT == KEY_JUST_PRESSED) {
      Serial.println(" pressed\n");
      trellis.setPixelColor(key, 0xFFFFFF);
      trellis.noteOn(FIRST_MIDI_NOTE+key, 64);
    }
    else if (e.bit.EVENT == KEY_JUST_RELEASED) {
      Serial.println(" released\n");
      trellis.setPixelColor(key, 0x0);
      trellis.noteOff(FIRST_MIDI_NOTE+key, 64);
    }
  }

  // Check for accelerometer

```

```

sensors_event_t event;
accel.getEvent(&event);
/* Display the results (acceleration is measured in m/s^2) */
//Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print(" ");
//Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print(" ");
//Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print("
");Serial.println("m/s^2 ");
int xbend = 0;
int ybend = 0;

if (abs(event.acceleration.y) < 2.0) { // 2.0 m/s^2
  // don't make any bend unless they've really started moving it
  ybend = 8192; // 8192 means no bend
} else {
  if (event.acceleration.y > 0) {
    ybend = ofMap(event.acceleration.y, 2.0, 10.0, 8192, 0, true); // 2 ~ 10 m/
s^2 is downward bend
  } else {
    ybend = ofMap(event.acceleration.y, -2.0, -10.0, 8192, 16383, true); // -2 ~
-10 m/s^2 is upward bend
  }
}
if (ybend != last_ybend) {
  Serial.print("Y pitchbend: "); Serial.println(ybend);
  trellis.pitchBend(ybend);
  last_ybend = ybend;
}

if (abs(event.acceleration.x) < 2.0) { // 2.0 m/s^2
  // don't make any bend unless they've really started moving it
  xbend = 0;
} else {
  if (event.acceleration.x > 0) {
    xbend = ofMap(event.acceleration.x, 2.0, 10.0, 0, 127, true); // 2 ~ 10 m/
s^2 is upward bend
  } else {
    xbend = ofMap(event.acceleration.x, -2.0, -10.0, 0, 127, true); // -2 ~ -10
m/s^2 is downward bend
  }
}
if (xbend != last_xbend) {
  Serial.print("X mod: "); Serial.println(xbend);
  trellis.controlChange(xCC, xbend); //xCC is set at top of sketch. e.g., CC 1
is Mod Wheel
  last_xbend = xbend;
}

trellis.sendMIDI(); // send any pending MIDI messages

delay(10);
}

// floating point map
float ofMap(float value, float inputMin, float inputMax, float outputMin, float
outputMax, bool clamp) {
  float outVal = ((value - inputMin) / (inputMax - inputMin) * (outputMax -
outputMin) + outputMin);

  if (clamp) {
    if (outputMax < outputMin) {
      if (outVal < outputMax) outVal = outputMax;
      else if (outVal > outputMin) outVal = outputMin;
    } else {
      if (outVal > outputMax) outVal = outputMax;
      else if (outVal < outputMin) outVal = outputMin;
    }
  }
}

```

```

    return outVal;
}
// Input a value 0 to 255 to get a color value.
// The colours are a transition r - g - b - back to r.
uint32_t Wheel(byte WheelPos) {
  WheelPos = 255 - WheelPos;
  if(WheelPos < 85) {
    return trellis.Color(255 - WheelPos * 3, 0, WheelPos * 3);
  }
  if(WheelPos < 170) {
    WheelPos -= 85;
    return trellis.Color(0, WheelPos * 3, 255 - WheelPos * 3);
  }
  WheelPos -= 170;
  return trellis.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

```

The most likely things you'll want to change are the MIDI channel, the MIDI note range of the keypad buttons, and the CC channel numbers of the accelerometer tilt axes.

MIDI Channel

You can adjust the MIDI channel by changing the value of this line:

```
#define MIDI_CHANNEL 0 // default channel # is 0
```

To change this controller to send MIDI over channel 6, for example, you could make it read:

```
#define MIDI_CHANNEL 6
```

Note Range

The Trellis M4's 32 buttons are set up in this project as chromatic keys with the low note set to C3. This corresponds to MIDI notes 36 through 68. You can change this range by adjusting this line in the code:

```
#define FIRST_MIDI_NOTE 36
```

CC Channel

The x-axis (forward/backward) tilt of the Trellis M4's accelerometer is set by default to send out a range of 0-127 values over MIDI CC channel 1. This is typically received as "mod wheel" in most synthesizers. You can change this value, in case it isn't simple in

your software synth to re-assign which knob is turned by CC 1. This line is the one to adjust:

```
int xCC = 1;
```

Once you're done making changes, upload the sketch to your Trellis M4 and test it out in your MIDI monitor utility and synth!

Test MIDI Output

You can test to see that your Trellis M4 is outputting MIDI signals properly by using a MIDI utility. These will report all activity coming from any MIDI controller plugged into your system. Here are some to try:

- [MIDI Monitor \(https://adafru.it/rBH\)](https://adafru.it/rBH) for mac os
- [MIDI-OX \(https://adafru.it/nkF\)](https://adafru.it/nkF) for windows
- [MIDI Wrench \(https://adafru.it/C-1\)](https://adafru.it/C-1) for iOS
- [MIDI Scope \(https://adafru.it/C-2\)](https://adafru.it/C-2) for Android
- [KMidimon \(https://adafru.it/rBI\)](https://adafru.it/rBI) for Linux
- [Web MIDI Monitor \(https://adafru.it/C-3\)](https://adafru.it/C-3) for Chrome browser