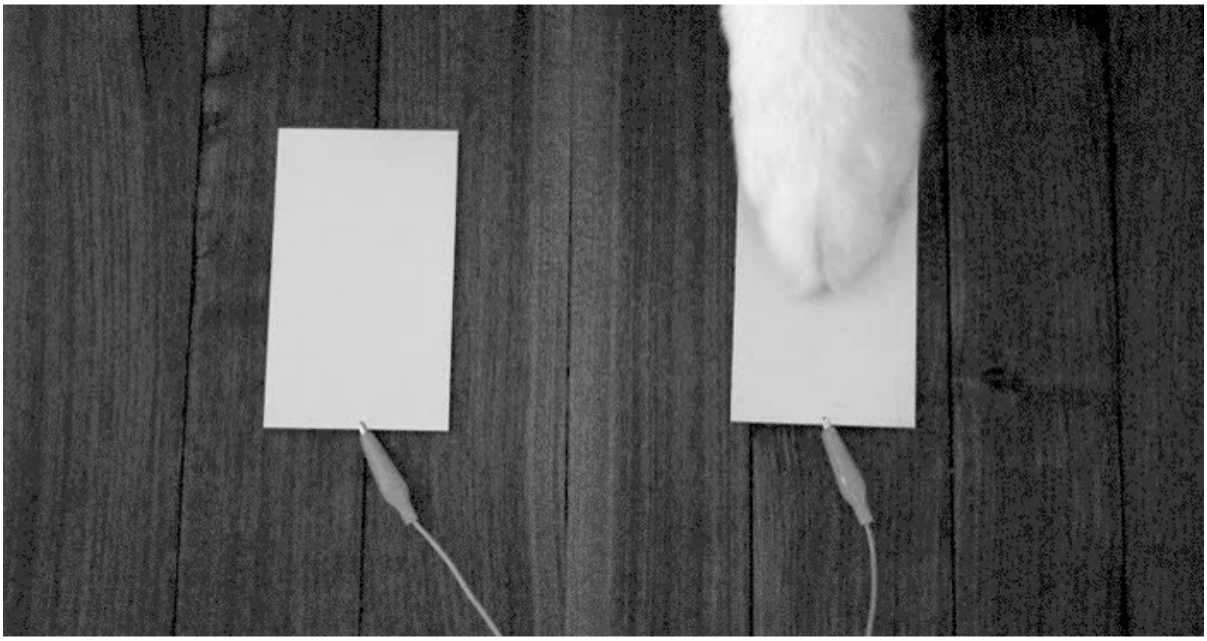
# Capacitive Touch Drum Machine

Created by Todd Treece



https://learn.adafruit.com/capacitive-touch-drum-machine

Last updated on 2024-06-03 01:41:29 PM EDT

# Table of Contents

# Overview

Have you (or your dog) ever wanted a drum machine? If so, this project is for you. We will use an Arduino Leonardo, a MPR121 capacitive touch shield, and a strip of sixteen RGB NeoPixels to make a simple drum machine that you can use to control USB MIDI or serial MIDI devices.

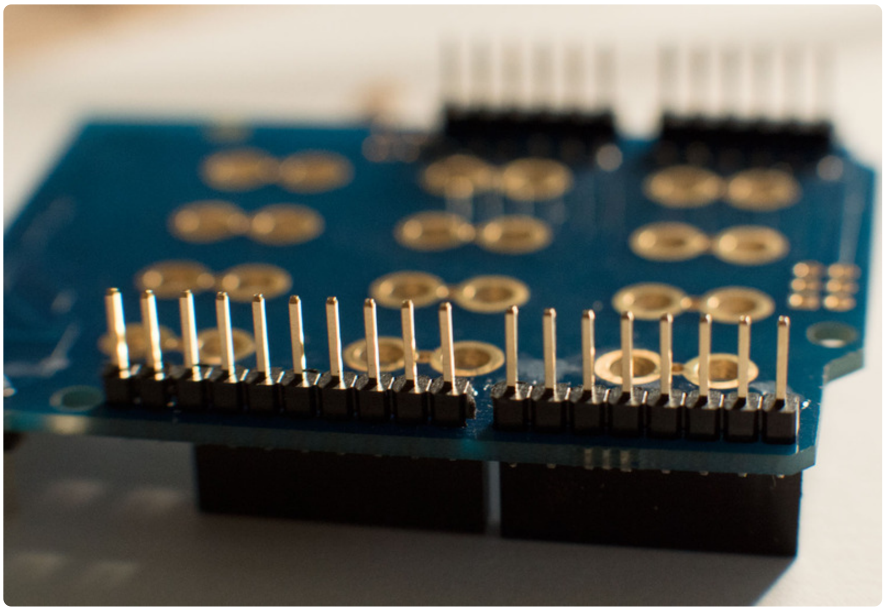
This tutorial is based on the same hardware setup used in the Make a Sequencer from a Tree (https://adafru.it/f3G) video. The only difference is that the MPR121 capactive touch breakout was used in the video instead of the MPR121 shield we will be using in this tutorial, but they both will work equally well for this project. If you would like to learn more about how the metal buttons were created in that video, you can check out our guide on using metal inlay for capactive touch buttons (https://adafru.it/Cfp).

Let's take a look at the version of that sequencer we will be making in this tutorial.

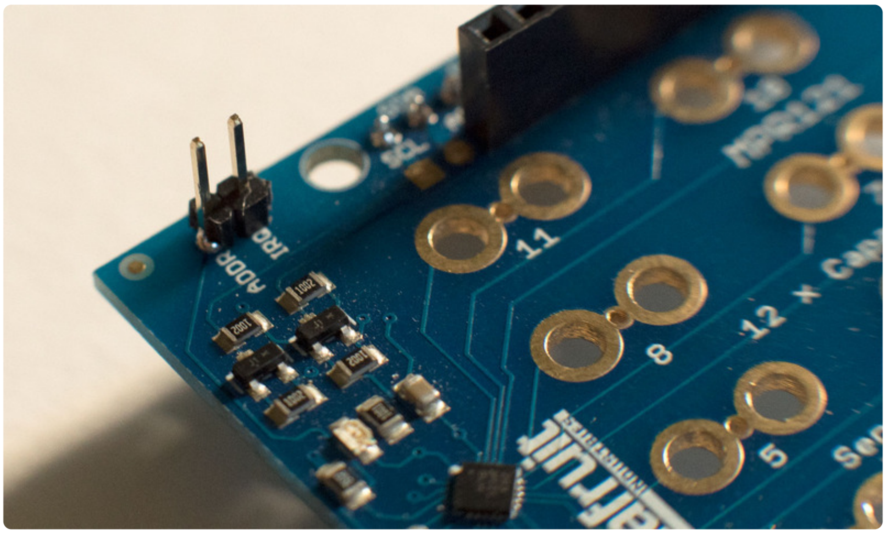
Now that you have a handle on what we are making, let's take a look at how to wire everything up.

# Wiring

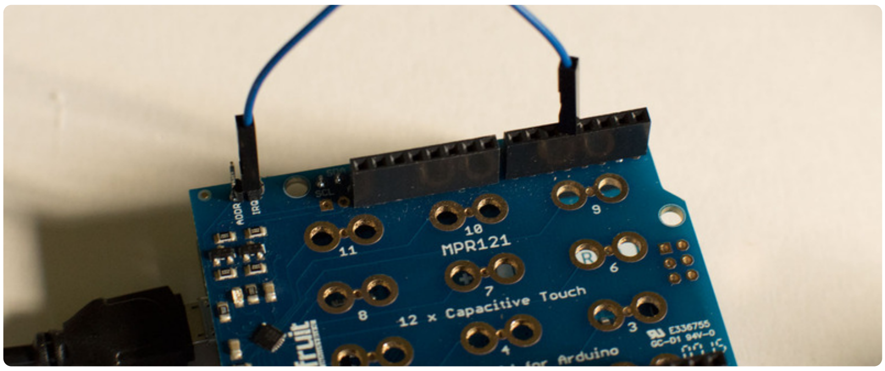
There are only a few things to wire up since we are using the MPR121 shield. First, you will need to solder the headers to the MPR121 shield. If you need help learning how to solder, check out our soldering guide (https://adafru.it/dgp). You can use male headers or shield stacking headers if you have some handy.



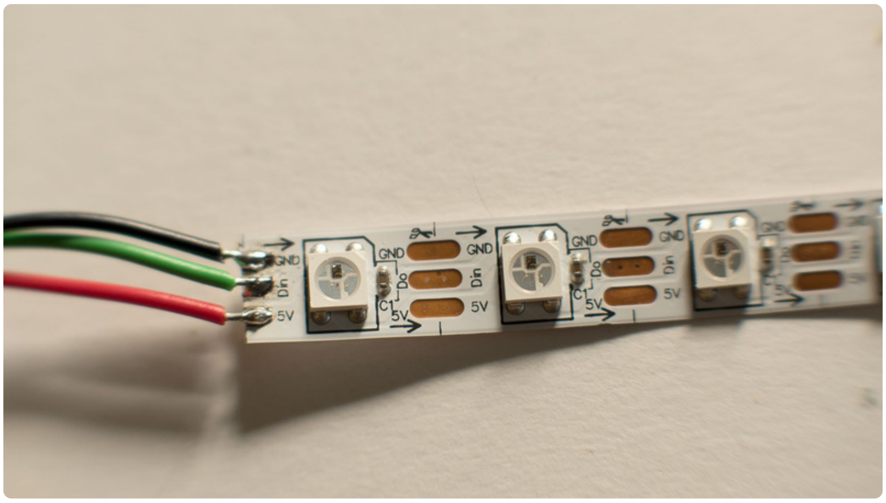
Next, you will need to add two male header pins to **ADDR** and **IRQ** pins on the top of the MPR121 shield. Make sure the long portion the pins are facing up as shown in the picture below.



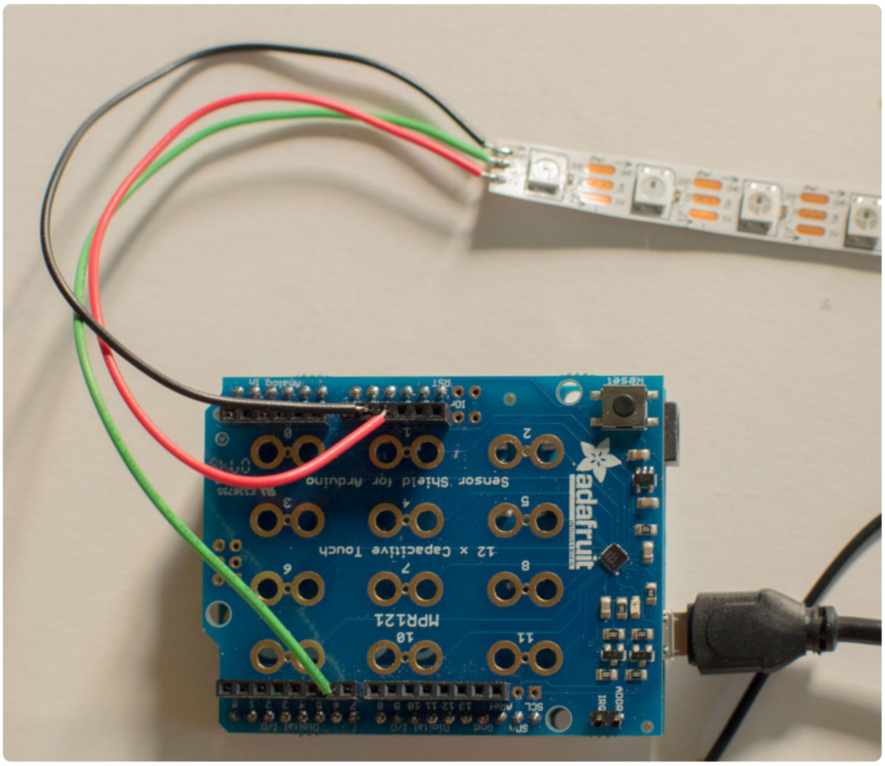
Use a female/male jumper cable to connect the MPR121 **IRQ** pin to **digital pin 4**.



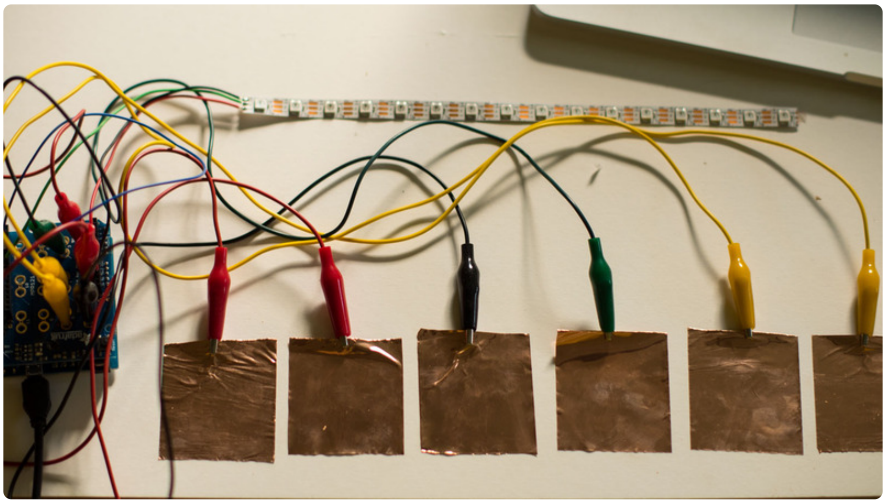
Next, grab your strip of 16 NeoPixels and solder three lengths of wire to the **5V**, **DIN**, and **GND** pins on the NeoPixels. If you are using two NeoPixel sticks, you will also need to solder them together. If you are using a strip of NeoPixels, you can cut a strip of 16 NeoPixels for use with this project.



Seat the MPR121 shield on top of the Leonardo, and attach the NeoPixel **5V** and **GND** wires to the **5V** and **GND** pins on the MPR121 shield. Attach the NeoPixel **DIN** wire to **digital pin 6** on the MPR121 shield.



Use alligator clips to connect pads 0-5 on the MPR121 shield to fruit, metal, or anything else that is electrically conductive. In the example below, I'm using copper tape.



Next, we will look at the uploading the code to the Leonardo.

# Code

For this project you will need version **1.6.3 of the Arduino IDE** with the Adafruit Arduino Board Manager proxy added to your Arduino preferences. If you haven't configured the proxy yet, check out our guide (https://adafru.it/jcv) on setting up your Arduino IDE to use the proxy.

Arduino IDE v1.6.3 Download

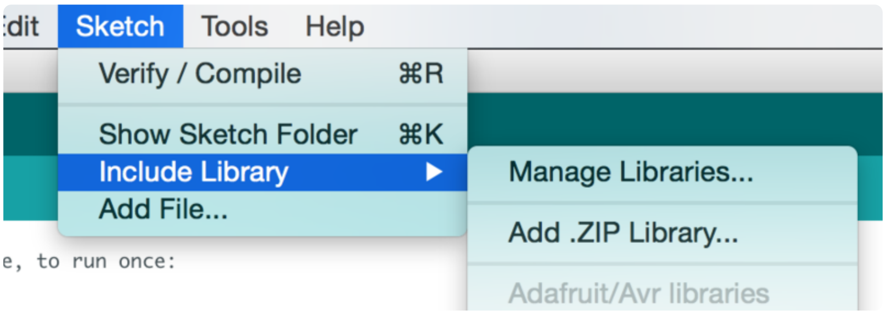
https://adafru.it/f1P

## Virtual MIDI Synths

You will need a virtual MIDI synth installed on your computer for this project. For Windows, the best one seems to be VirtualMIDISynth (https://adafru.it/f3J), and for OS X SimpleSynth (https://adafru.it/f3K) is your best bet.

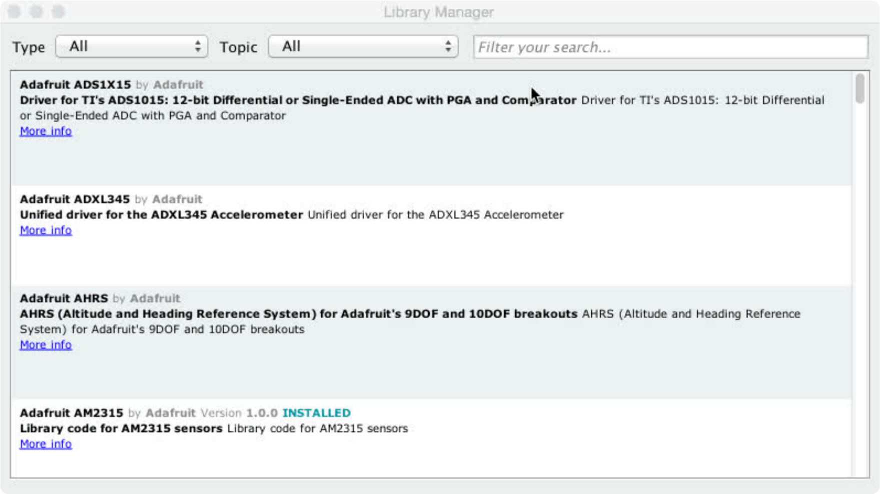
## Installing the Arduino Library Dependencies

You will need to install a few things using the new Arduino Library and Board Managers. First, open up the Arduino Library Manager under the **Sketch->Include Library** menu. Select the **Manage Libraries...** item from the menu.



You will need to use the search box to find and install these three libraries:

- **FifteenStep**
- **Adafruit NeoPixel**
- **Adafruit MPR121**



**Adafruit ADS1X15** by Adafruit
**Driver for TI's ADS1015: 12-bit Differential or Single-Ended ADC with PGA and Comparator** Driver for TI's ADS1015: 12-bit Differential or Single-Ended ADC with PGA and Comparator
More info

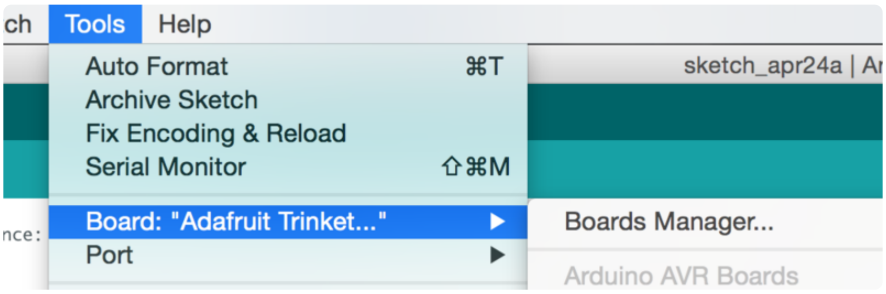
**Adafruit ADXL345** by Adafruit
**Unified driver for the ADXL345 Accelerometer** Unified driver for the ADXL345 Accelerometer
More info

**Adafruit AHRS** by Adafruit
**AHRS (Altitude and Heading Reference System) for Adafruit's 9DOF and 10DOF breakouts** AHRS (Altitude and Heading Reference System) for Adafruit's 9DOF and 10DOF breakouts
More info

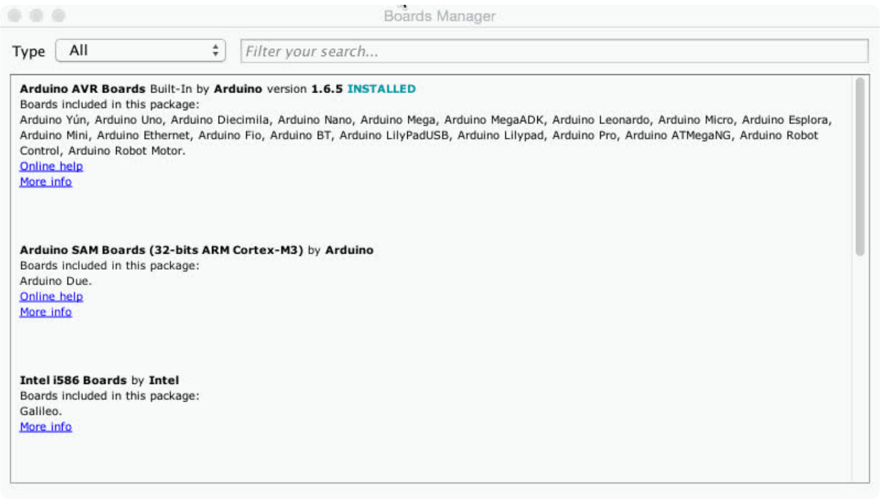
**Adafruit AM2315** by Adafruit Version **1.0.0** **INSTALLED**
**Library code for AM2315 sensors** Library code for AM2315 sensors
More info

## Adding USB MIDI Support

Next, you will need to install USB MIDI support for the Leonardo using the Boards Manager. Click on the **Boards Manager** item under the **Tools->Board** menu.



Search for MIDI using the search box, and install the **Leonardo & Micro MIDI-USB** package.

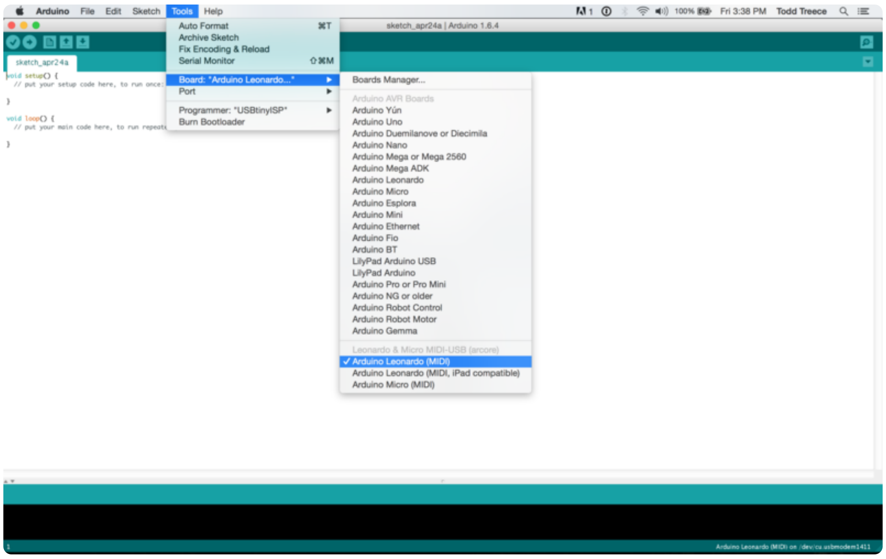


**Arduino AVR Boards** Built-In by **Arduino** version **1.6.5** **INSTALLED**
Boards included in this package:
Arduino Yún, Arduino Uno, Arduino Diecimila, Arduino Nano, Arduino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor.
Online help
More info

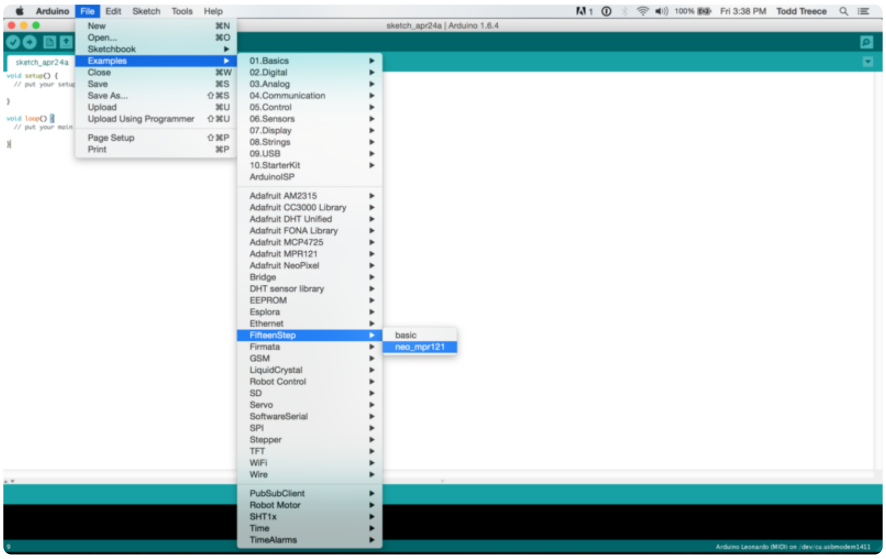
**Arduino SAM Boards (32-bits ARM Cortex-M3)** by **Arduino**
Boards included in this package:
Arduino Due.
Online help
More info

**Intel i586 Boards** by **Intel**
Boards included in this package:
Galileo.
More info

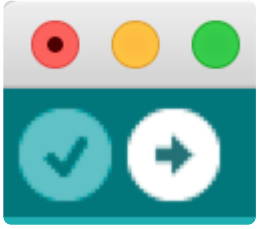
Next, make sure you have selected the **Arduino Leonardo (MIDI)** from the **Tools->Board** menu, and plug in the Leonardo to your computer's USB port using a micro USB cable.



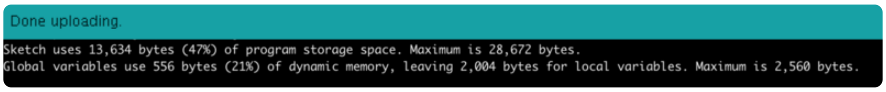
Now we are ready to open the example sketch. Open the **neo_mpr121** example from the **File->Examples->FifteenStep** menu. The example code is heavily commented, so we won't go into too much detail here. If you have any problems with it, please file an issue on GitHub (https://adafru.it/f3L).



Next, upload the example sketch to your Leonardo using the right arrow icon at the top of the sketch window.



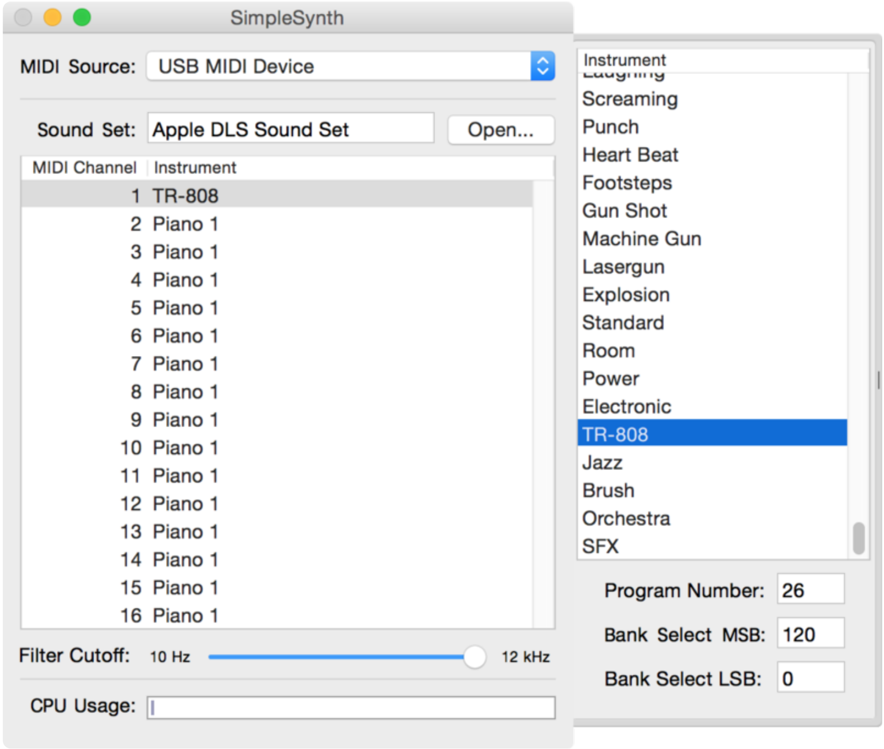
If everything goes as expected, you will see a Done Uploading message at the bottom of the window.



```
Done uploading.
Sketch uses 13,634 bytes (47%) of program storage space. Maximum is 28,672 bytes.
Global variables use 556 bytes (21%) of dynamic memory, leaving 2,004 bytes for local variables. Maximum is 2,560 bytes.
```

## Running the Example

Now that you have uploaded the sketch, you should see a sequence running on the NeoPixels. Open up your software synth that you downloaded earlier, and select the USB MIDI device as your input source. Then select a drum set from the list of available instruments. I'm using the TR-808 patch in SimpleSynth on OS X. Press a few of the pads, and your drum machine should save them to the sequence and play them back through the MIDI software synth.



MIDI Source: USB MIDI Device

Sound Set: Apple DLS Sound Set

| MIDI Channel | Instrument |
|---|---|
| 1 | TR-808 |
| 2 | Piano 1 |
| 3 | Piano 1 |
| 4 | Piano 1 |
| 5 | Piano 1 |
| 6 | Piano 1 |
| 7 | Piano 1 |
| 8 | Piano 1 |
| 9 | Piano 1 |
| 10 | Piano 1 |
| 11 | Piano 1 |
| 12 | Piano 1 |
| 13 | Piano 1 |
| 14 | Piano 1 |
| 15 | Piano 1 |
| 16 | Piano 1 |

Filter Cutoff:

CPU Usage:

Program Number: 26

Bank Select MSB: 120

Bank Select LSB: 0

Next, we will look at the simple user interface in the example sketch.

# User Interface

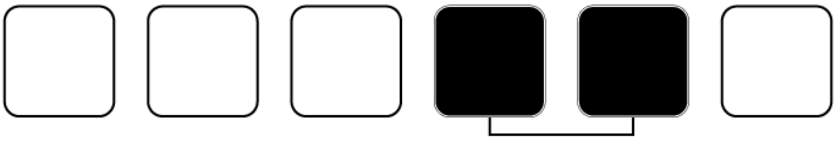
There are some handy commands buried beneith the surface of the example sketch. You can change things like tempo, shuffle, MIDI channel, sequence length, MIDI pitch, and MIDI velocity on the fly using a few simple button combinations. There is a condensed single page cheatsheet PDF download available for the user interface that will be a good reference once you have read through the full command documentation.

User Interface Cheatsheet PDF

https://adafru.it/f3M

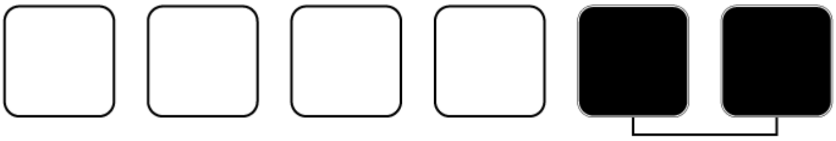
## Record Mode Toggle

The sequencer starts in record mode, but if you would like to toggle it off so you can test out the instruments without recording them to the sequence, you can press pads **4** and **5** simultaneously. You will know if you are currently recording if every fourth NeoPixel flashes red, if you are not recording, all pixels will flash blue.



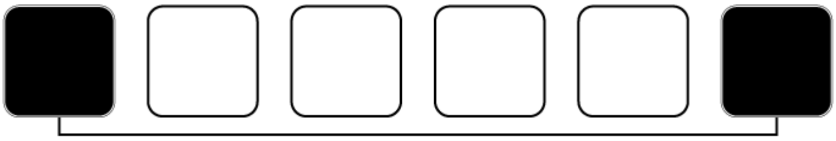
## Play & Pause Toggle

You can toggle between play and pause by pressing pads **5** and **6** simultaneously.



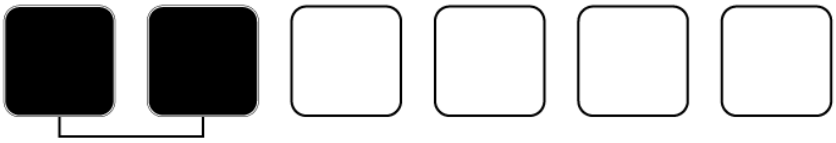
## All Notes Off

You can clear all currently playing notes from the sequence by pressing pads **1** and **6** simultaneously.



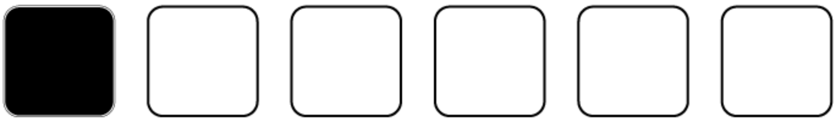
## Command Mode Toggle

Tempo, shuffle, step count, MIDI channel, pitch, and velocity are all available underneath a sub menu I'm calling Command Mode. You can enter command mode by pressing pads **1** & **2** simultaneously. You will know you have entered command mode when all of the NeoPixels flash red.

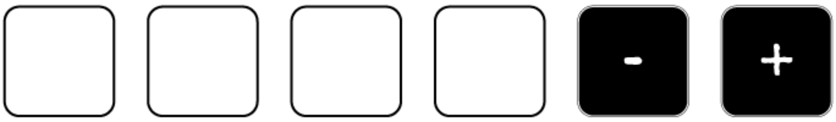


## Tempo Change

You can change the tempo by entering command mode and then pressing pad **1**.



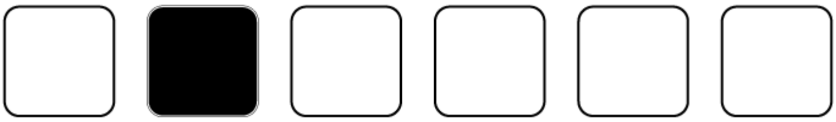
You should see the NeoPixels flash a different color at this point. You can now use pad **5** to decrease the tempo, or pad **6** to increase the tempo.



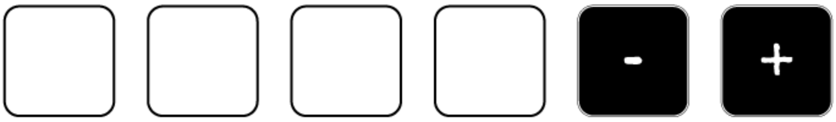
You can exit tempo change mode by using the command mode toggle pads.

## Shuffle Change

You can add or remove shuffle by entering command mode and then pressing pad **2**.



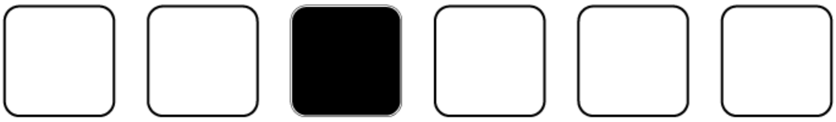
You should see the NeoPixels flash a different color at this point. You can now use pad **5** to decrease the shuffle, or pad **6** to increase the shuffle.



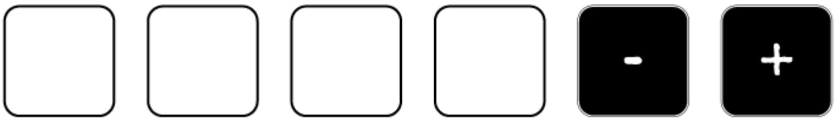
You can exit shuffle mode by using the command mode toggle pads.

## Changing the Length of the Sequence

You can change the number of steps in the sequence by entering command mode and then pressing pad **3**.



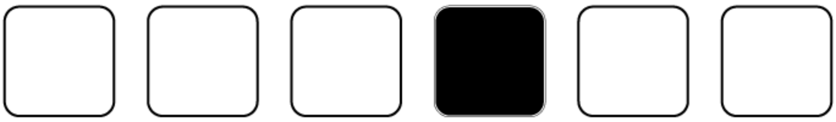
You should see the NeoPixels flash a different color at this point. You can now use pad **5** to decrease the length, or pad **6** to increase the length.



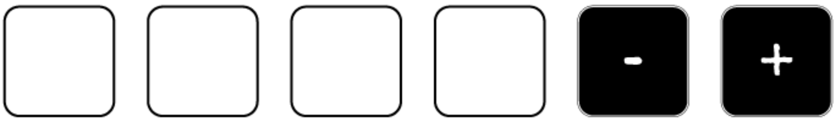
You can exit sequence length change mode by using the command mode toggle pads.

## MIDI Channel Change

You can change the current MIDI channel for new notes by entering command mode and then pressing pad **4**. The notes previously recorded will remain on the MIDI channel that was selected when they were recorded.



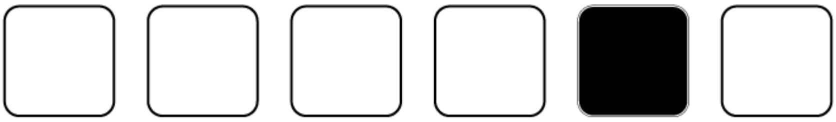
You should see the NeoPixels flash a different color at this point. You can now use pad **5** to decrease the channel number, or pad **6** to increase the channel number.



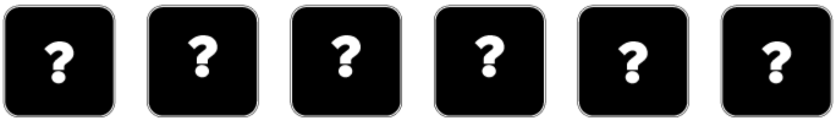
You can exit MIDI channel mode by using the command mode toggle pads.

## Pad Pitch Change

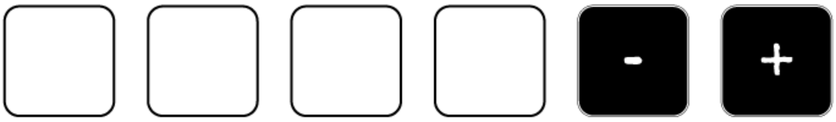
You can change the MIDI pitch of a pad is using by entering command mode and then pressing pad **5**. This change will only impact newly recorded notes. All previously recorded notes will retain their recorded pitch.



You can then select which pad you wish to change.



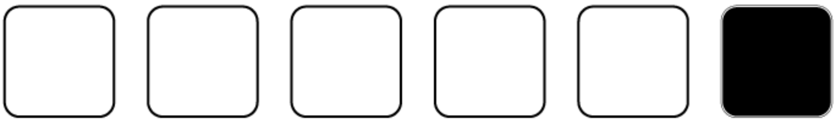
Once you have selected the pad you wish to change, you can use pad **5** to decrease the MIDI pitch value for the previously selected pad, or pad **6** to increase the MIDI pitch.



You can exit pitch change mode by using the command mode toggle pads.

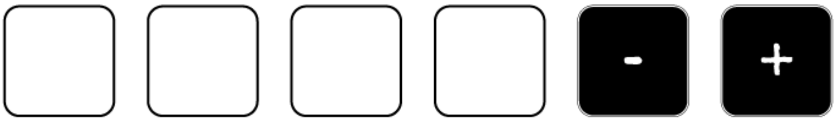
## Pad Velocity Change

You can change the MIDI velocity of a pad is using by entering command mode and then pressing pad **6.** This change will only impact newly recorded notes. All previously recorded notes will retain their recorded velocity.



You can then select which pad you wish to change.

Once you have selected the pad you wish to change, you can use pad **5** to decrease the MIDI velocity value for the previously selected pad, or pad **6** to increase the MIDI velocity.



You can exit velocity change mode by using the command mode toggle pads.

# Next Steps & Thanks

The FifteenStep MIDI sequencer library (https://adafru.it/f3N) that the example uses does not have to be used with USB MIDI. You could use it to output to a standard serial MIDI device, or use the MIDI note data for other purposes. There's an example included with the FifteenStep library that demonstrates using the library with serial MIDI out.

Another possibility is to use the MIDI data to control devices without MIDI capabilities. Below is an example of using an Arduino Uno, NeoPixels, and two MCP4725 DAC breakouts to record MIDI data from a MIDI controller and play it back as contol voltage to a modular synthesizer.

You can find the code used in this example on GitHub (https://adafru.it/f3O).

## Thanks

Thanks to Ladyada & Tony DiCola for the huge memory saving suggestions that greatly increased polyphony in the FifteenStep library, and thanks to Brennen Bearnes for the UI Cheatsheet PDF idea. Also, another big thanks to Marty McGuire and Baltimore Node (https://adafru.it/elm) for the help with laser engraving the Adafruit logo on the sequencer featured in the Make a Sequencer from a Tree (https://adafru.it/f3G) video.

## Special Thanks

A special thanks goes out to my friendly Alaskan Malamute, Biko. He graciously donated his time and beat making skills for the video at the beginning of the guide.