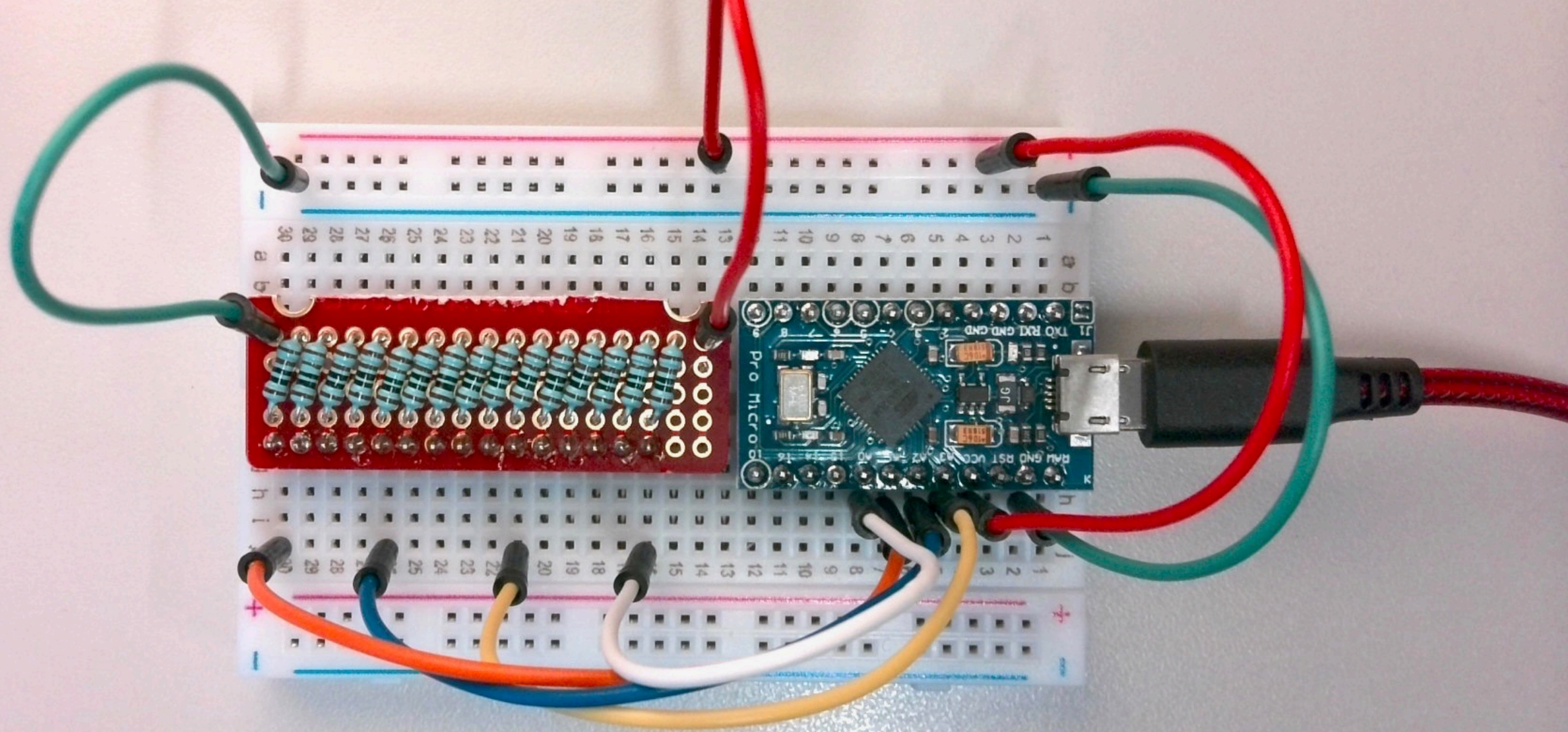


EXPLORATION

UWTMC

Matt Borland - 2019



WE'RE MAKING AN INSTRUMENT THAT MAKES USERS EXPLORE A MUSICAL SPACE

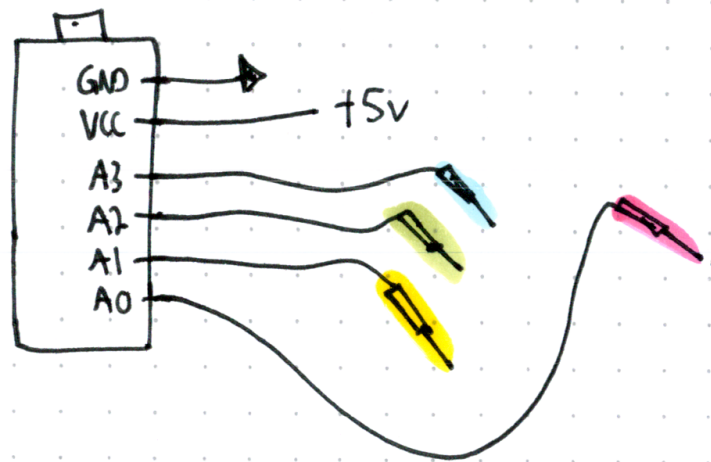
.....

We'll use a voltage divider and a "patching" paradigm to send sequences of MIDI data to our computer which we'll turn into sound with a software synthesizer!

EXPLORATION!

GO EXPLORE!

Arduino Pro Micro



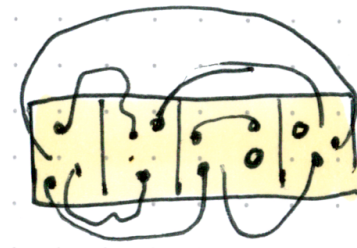
"PATCHING" TO DIFFERENT VOLTAGES ON THE VOLTAGE DIVIDER WILL ALLOW US TO INPUT DIFFERENT CONTROL SIGNALS TO THE ALGORITHM RUNNING ON OUR ARDUINO!

VOLTAGE DIVIDER



$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

VOLTAGE DIVIDERS LET US USE PRECISE VOLTAGES TO DO OTHER THINGS IN OUR CIRCUIT.



"MODULAR SYNTHS"
USE A SIMILAR
PATCHING PARADIGM
AND "CONTROL VOLTAGE"
OR
"CV"

THERE ARE 16^4 UNIQUE COMBINATIONS TO EXPLORE

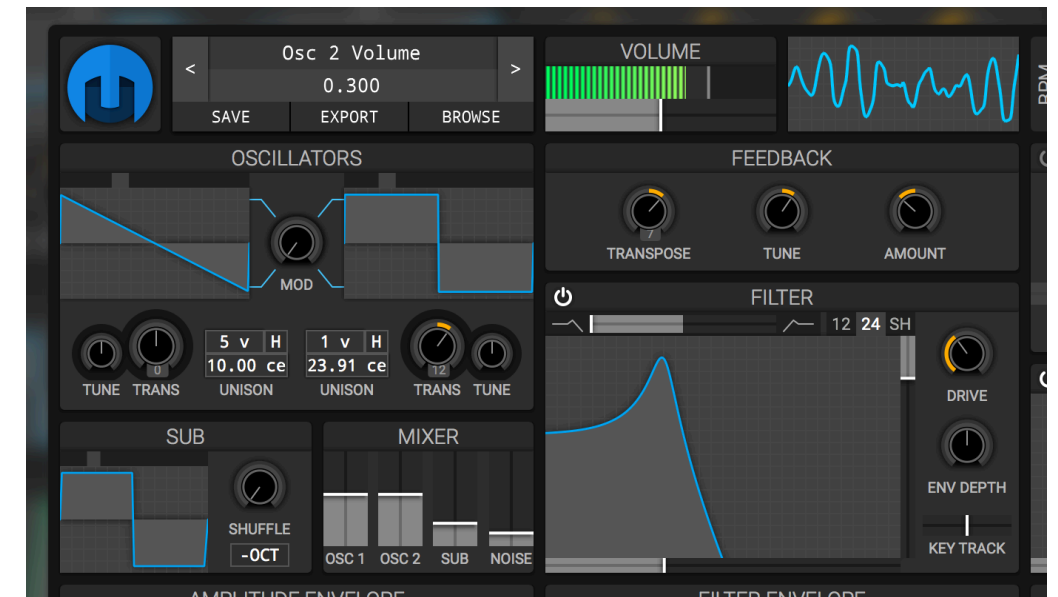
65,536

UW-TMC EXPLORATION 2019

DOWNLOADS

You need two pieces of software. Both are free and multi-platform!

Download the Arduino IDE



Arduino IDE - a software platform used to program your microcontroller.

<https://www.arduino.cc/en/Main/Software>

Helm - a software synthesizer to make musical sounds with your computer.

<https://tytel.org/helm/>

FILES

Workshops this term are at CML! All sessions run 3:30-5:30pm at Critical Media Lab, located in Communitech, 151 Charles St. W., Kitchener.

Sept. 18th: Deformation - MIDI Balloons: [DeformationFiles.zip](#)

Oct. 2nd: Gravity - Shruti Box: [GravityFiles.zip](#)

Oct. 23rd: Continuity - Seaboard and Bop Pad: [ContinuityFiles.zip](#)

Nov. 6th: Complexity - Modular Synthesis: [ComplexityFiles.zip](#)

Nov. 20th: Exploration - Co-play Patch Table: [ExplorationFiles.zip](#)

Each session will start with an explanation of a musical interface and a discussion of its expressivity, followed by a hands-on build with the sensing technology being explored, then the opportunity to develop improvisational skills through jamming with other participants on the instruments you've just built. Make sure to bring your laptop to be able to program and power your instrument!



Files are available as a ZIP at <https://uwtmc.com>

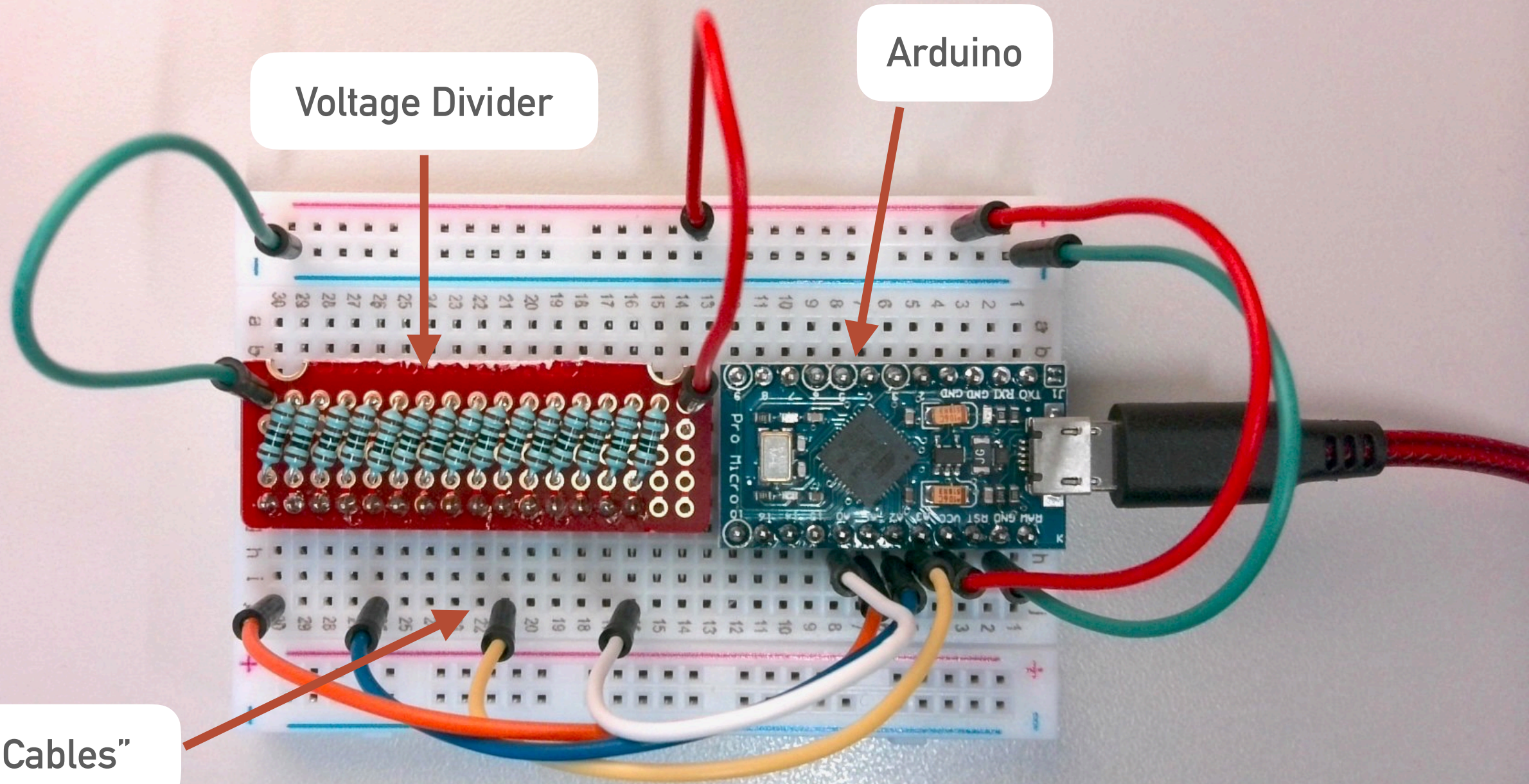
HOOKING UP YOUR CIRCUIT – COMPONENT NAMES

.....

Voltage Divider

Arduino

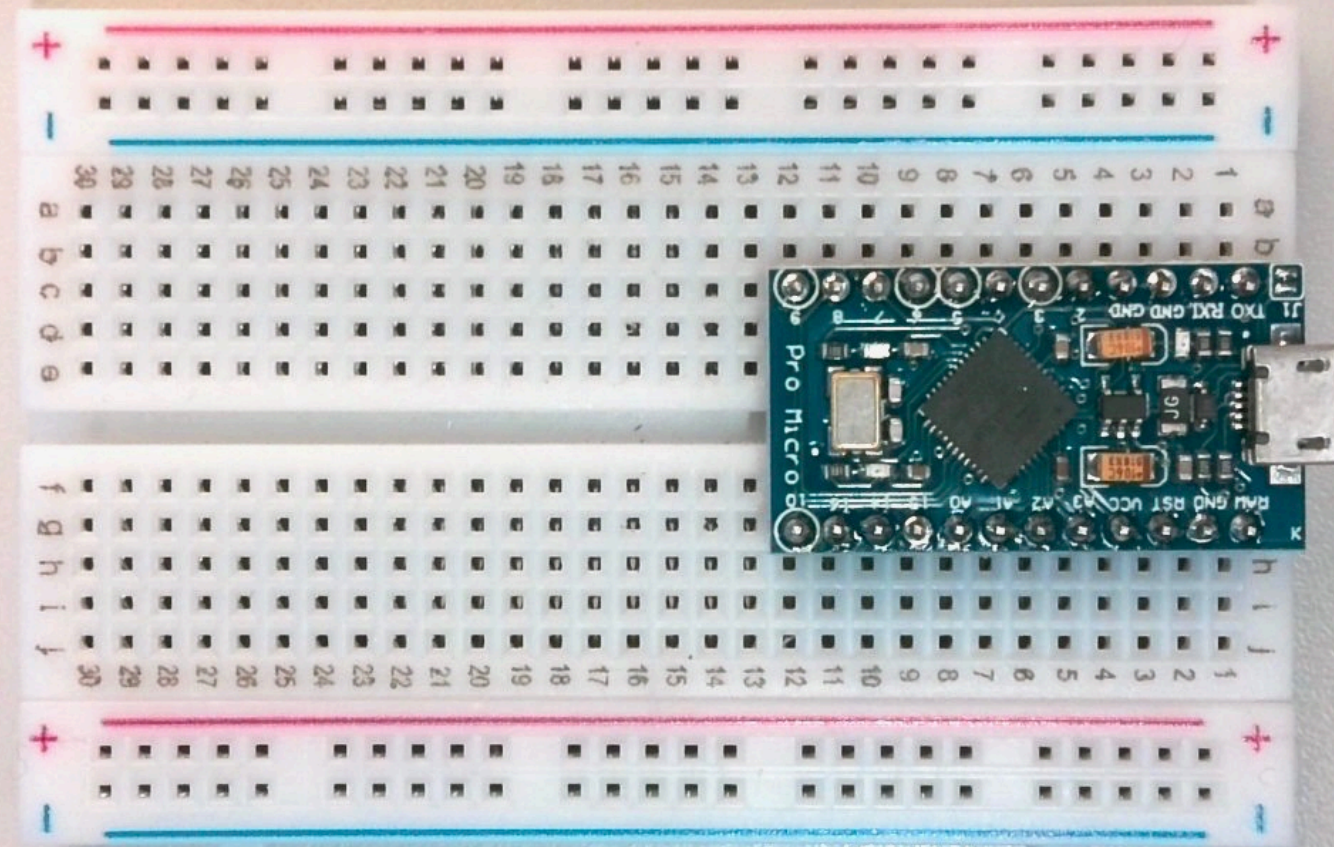
“Patch Cables”



HOOKING UP YOUR CIRCUIT!

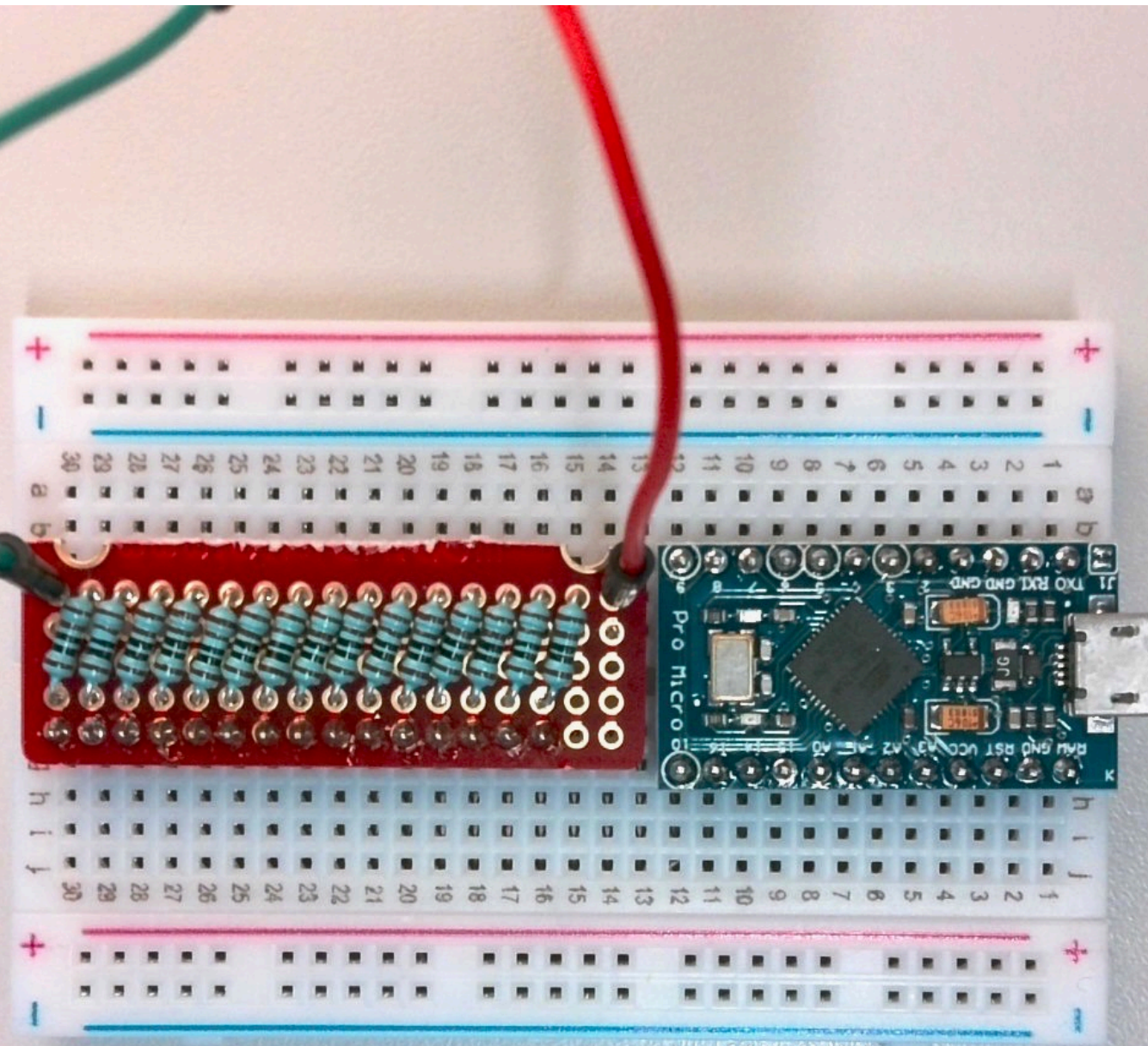
.....

Start with just your arduino plugged into the breadboard.



HOOKING UP YOUR CIRCUIT!

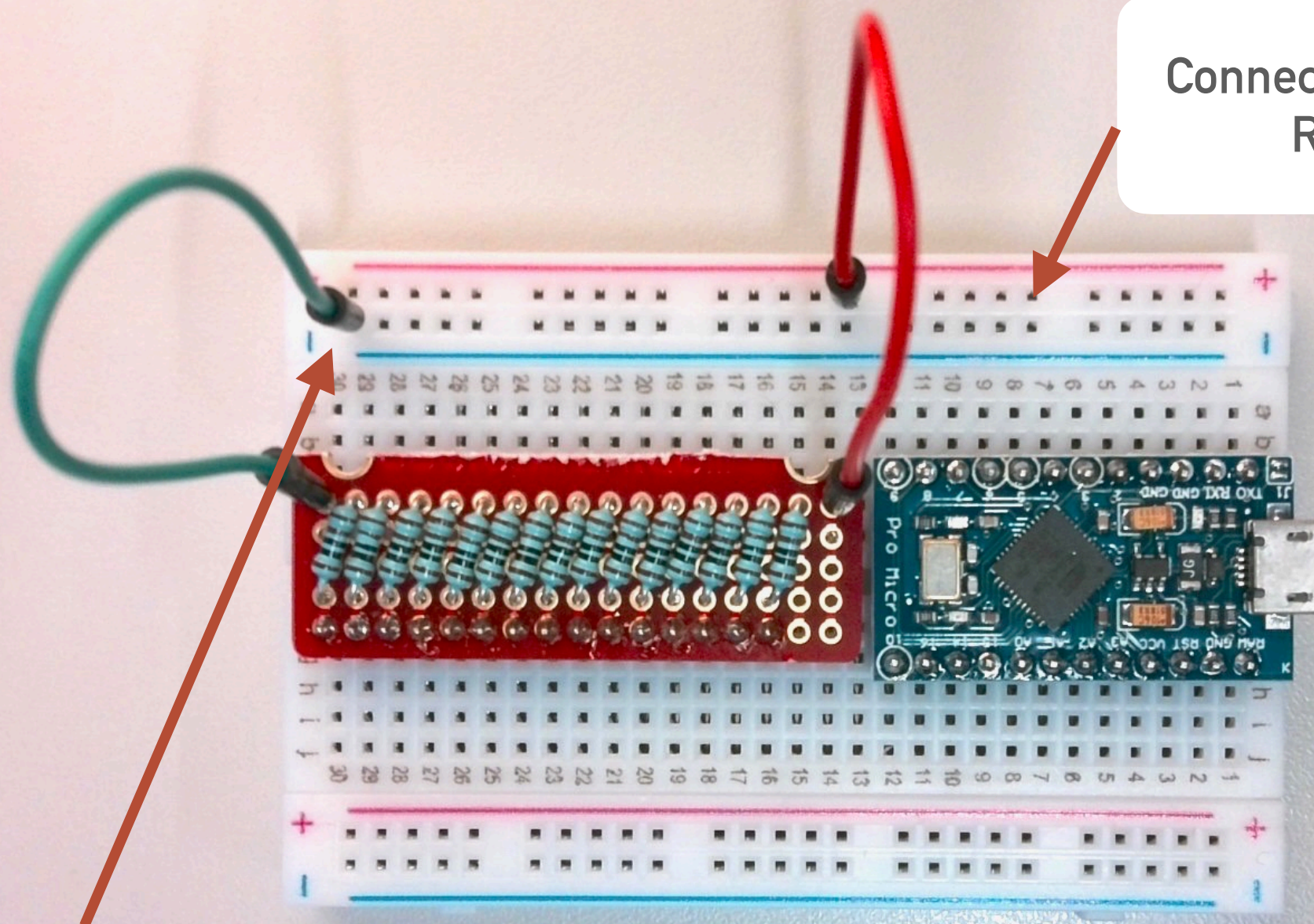
.....



Plug in your voltage divider to the breadboard as shown. Be careful to align it as shown.

HOOKING UP YOUR CIRCUIT!

.....

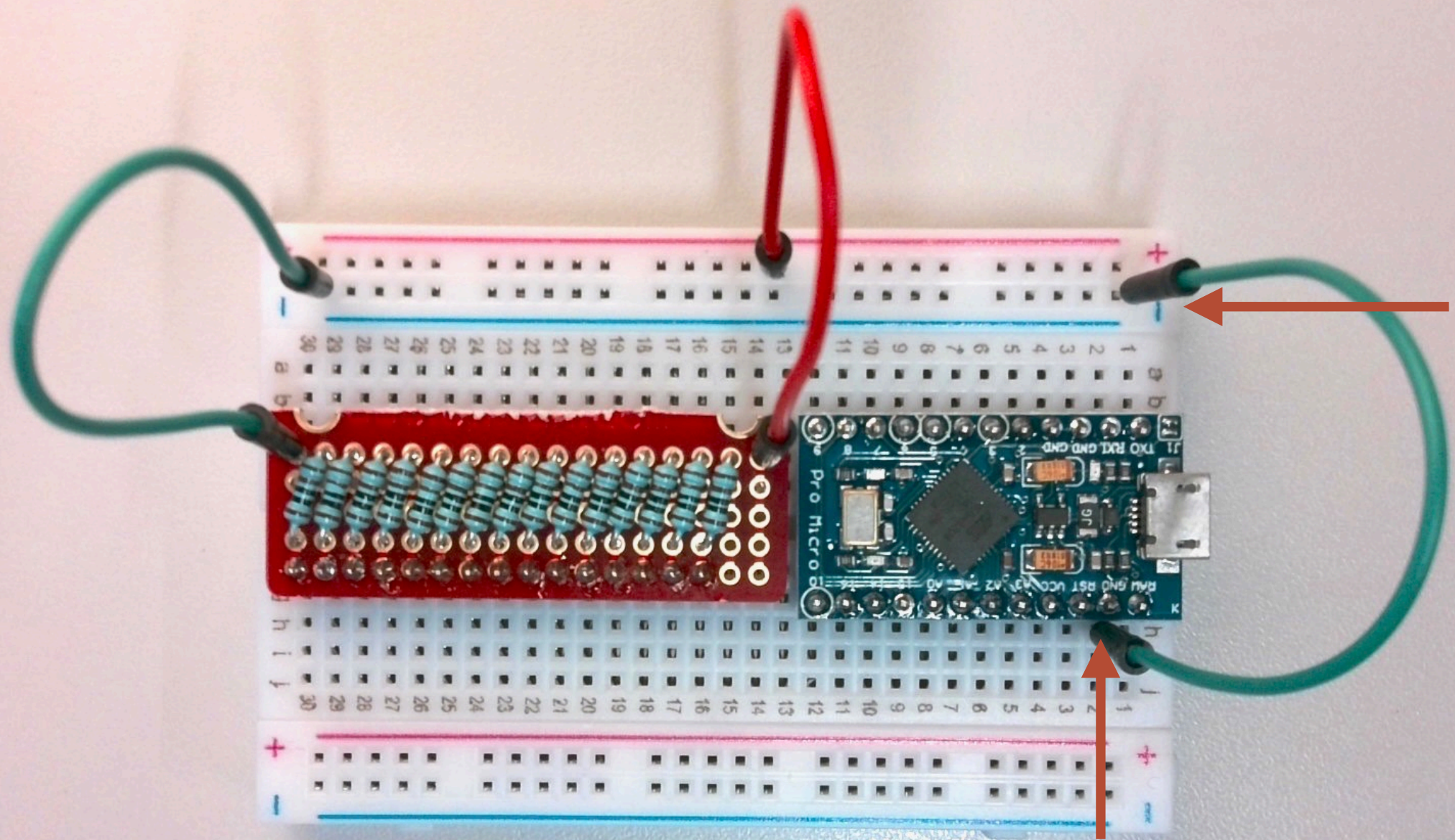


Connect the RED wire to the RED power rail.

Connect the GREEN wire to the BLUE power rail.

HOOKING UP YOUR CIRCUIT!

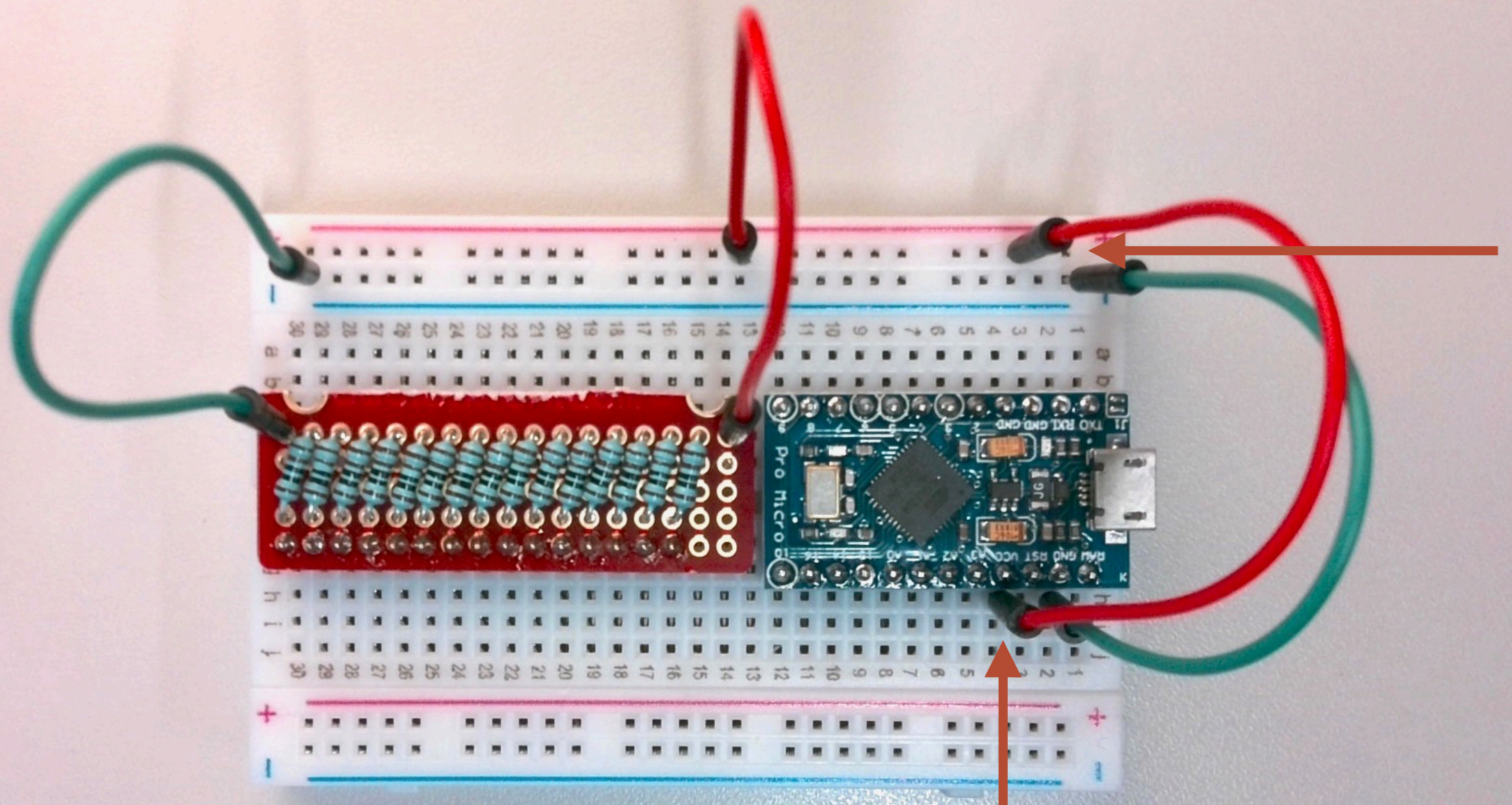
.....



Connect the GND pin on the arduino to the BLUE power rail.

HOOKING UP YOUR CIRCUIT!

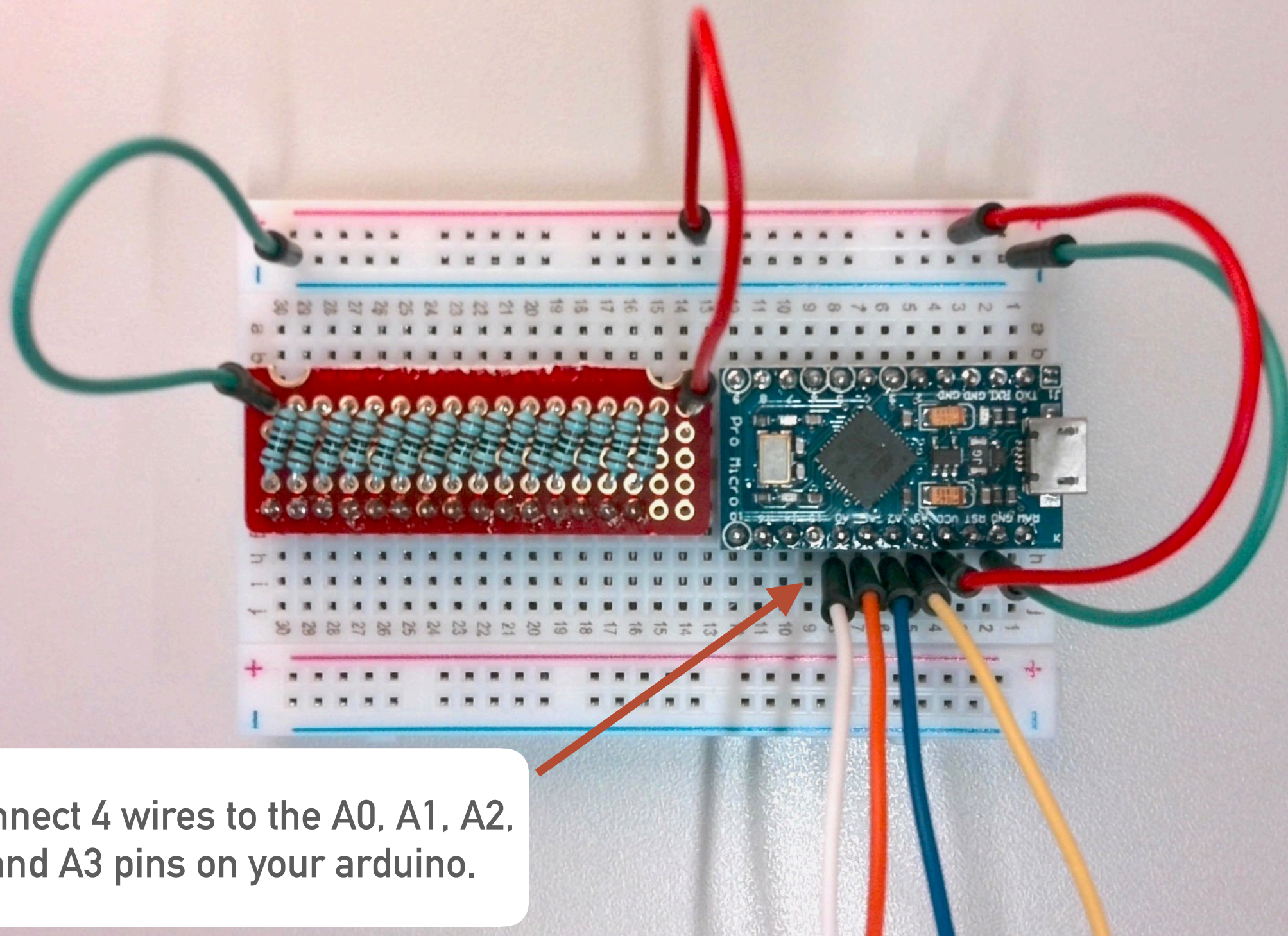
.....



Connect the VCC pin on the arduino to the RED power rail.

HOOKING UP YOUR CIRCUIT!

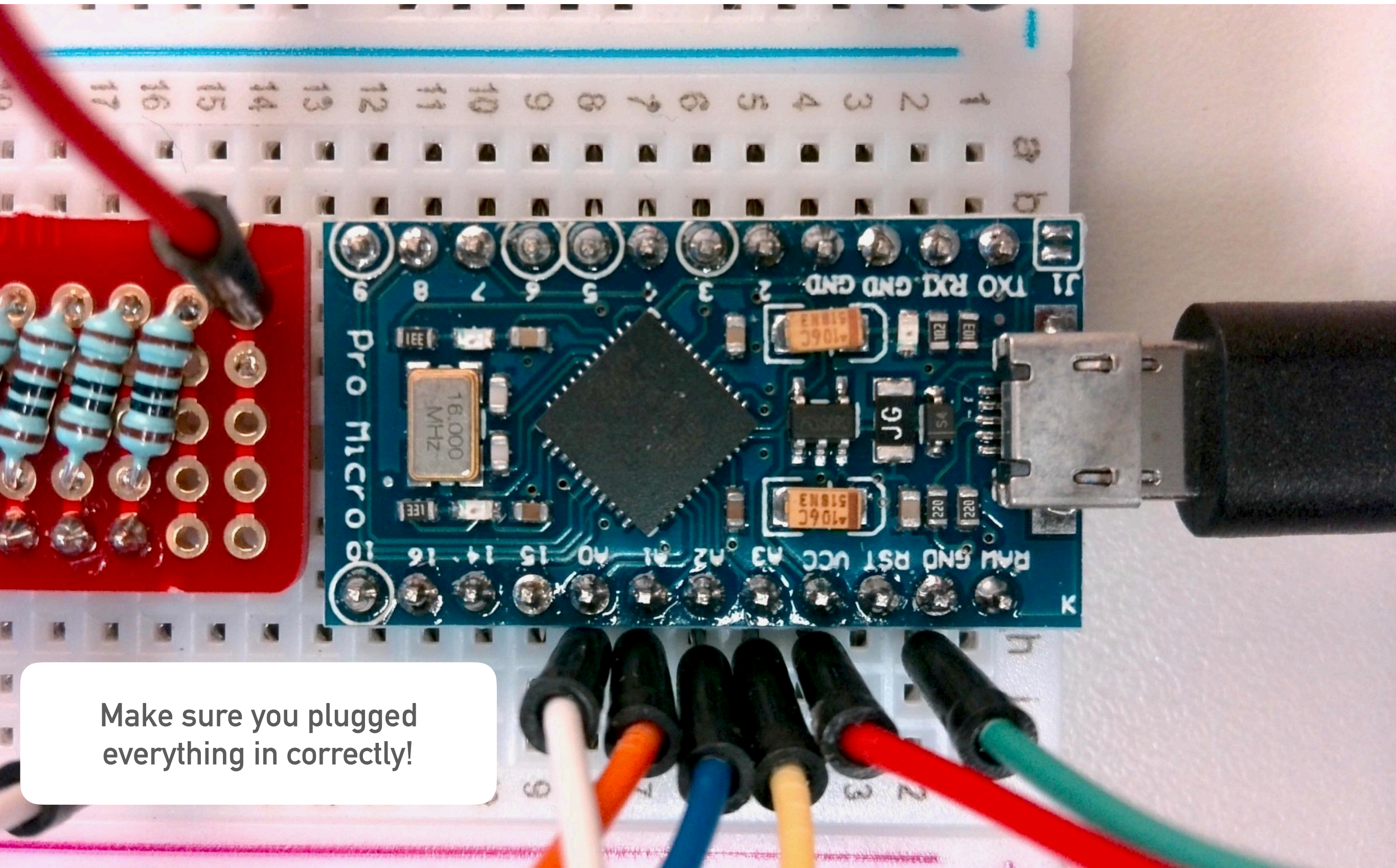
.....



Connect 4 wires to the A0, A1, A2, and A3 pins on your arduino.

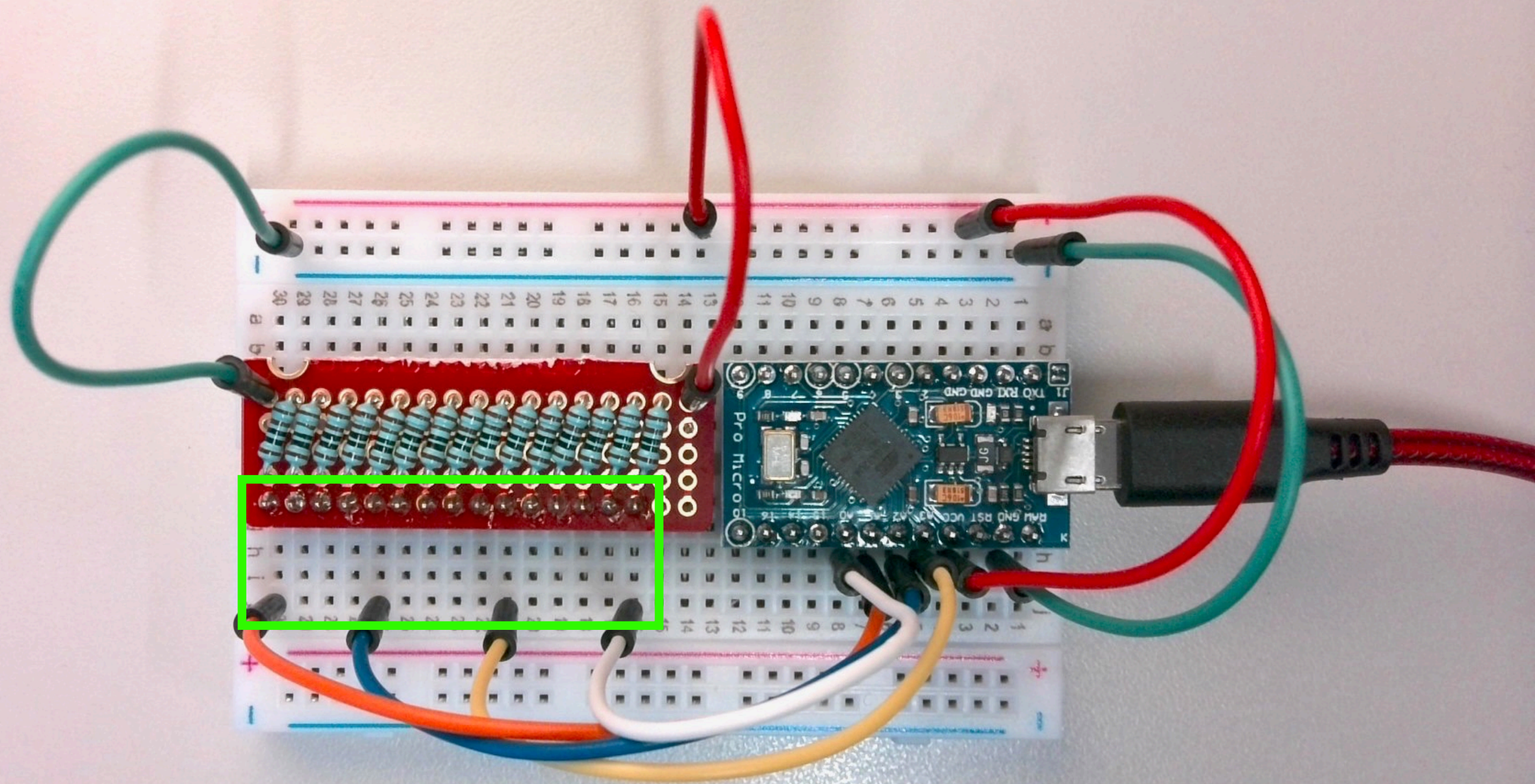
HOOKING UP YOUR CIRCUIT!

.....



Make sure you plugged everything in correctly!

HOOKING UP YOUR CIRCUIT!



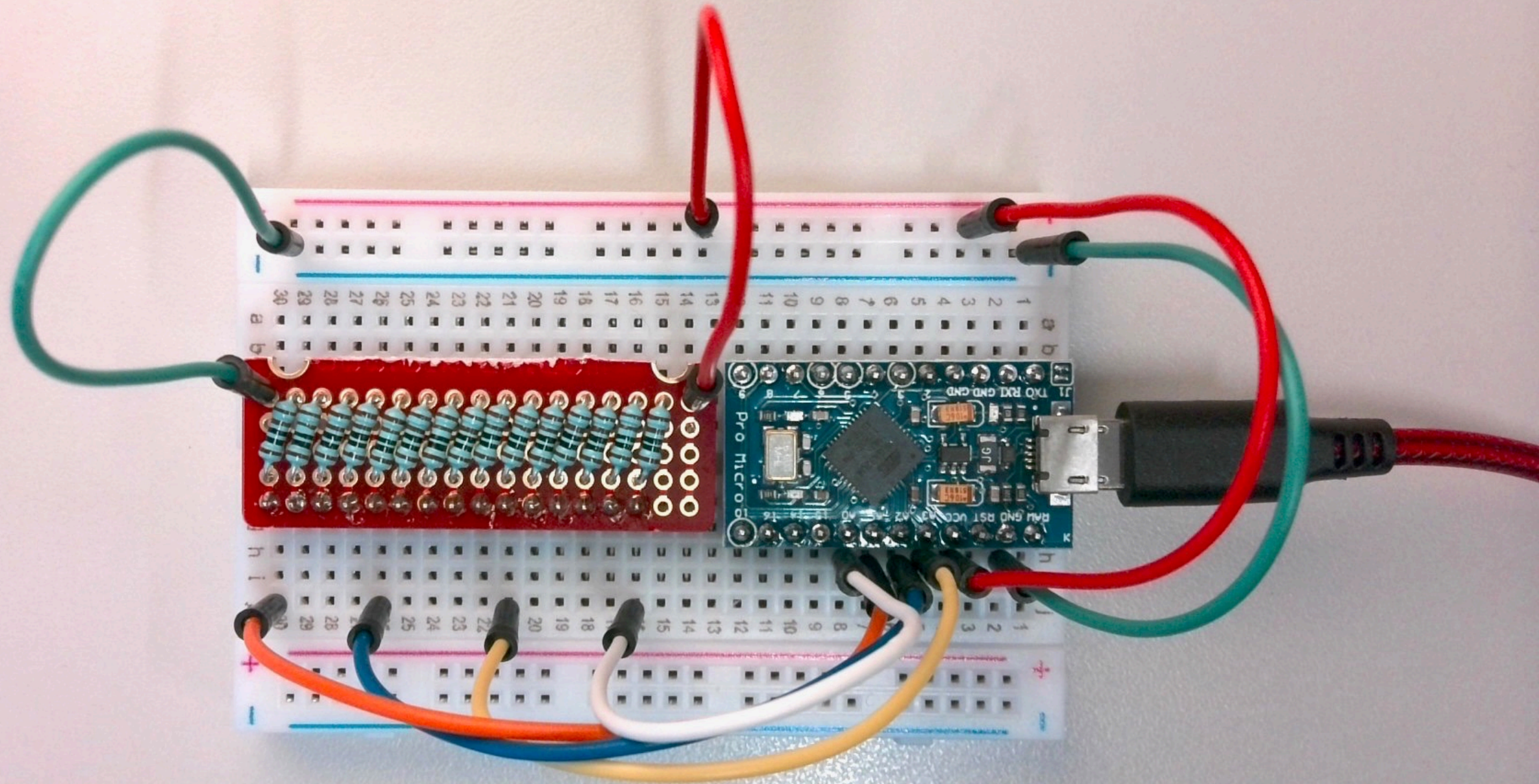
Connect the other ends of those 4 wires to any of the 15 pins of the voltage divider. It doesn't matter which ones you use. Changing which pins these 4 wires are connected to on the voltage divider will change the way the instrument sends MIDI messages to your computer.

SAFETY CHECK!

*BEFORE PLUGGING THE
USB CABLE INTO YOUR
COMPUTER GET MATT TO
CHECK YOUR BOARD SO
YOU KNOW IT'S SAFE AND
NOTHING WILL BREAK!*

PLUG IN THE USB CABLE, THEN GO PROGRAM IT!

.....



ON YOUR COMPUTER

Open “*Exploration.ino*”

Arduino IDE - a software platform used to program your microcontroller.

<https://www.arduino.cc/en/Main/Software>



```
Arduino File Edit Sketch Tools Help
Exploration | Arduino 1.8.7

Exploration
// Programmed using the knockoff Pro Micro boards - use Arduino/Genuino Micro as board type
// Watch out - your board's name might change and you'll have to select it again in Tools/Port
// after you program it the first time

// This program uses a resistor ladder and 4 analog inputs to provide
// Generative music a "patch"able environment - think of it
// as a modular synthesizer sequencer that sends out MIDI notes
// changing the algorithm below will change the music it makes!

// Libraries
// You need to install both libraries - search for them under Tools/Manage Libraries, and then
// search for the word in front of the .h
#include <elapsedMillis.h>
#include <MIDIUSB.h>

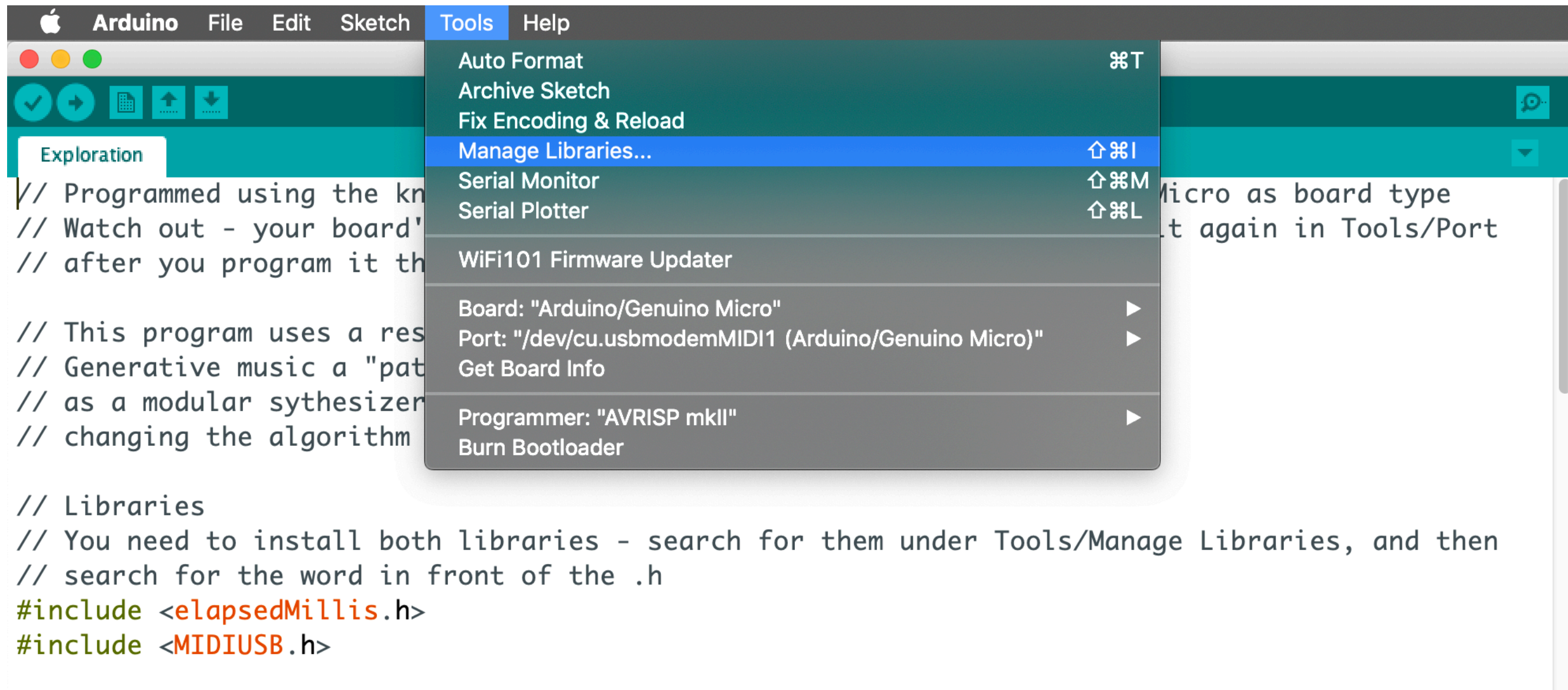
// Parameters we'll use to control the system output
int A = 0;
int B = 0;
int C = 0;
int D = 0;
int AB = 0;
int AmB = 0;
int AC = 0;
int AmC = 0;
int AD = 0;
int AmD = 0;
int BC = 0;
int BmC = 0;
int BD = 0;
int BmD = 0;
int CD = 0;
int CmD = 0;

// MIDI Parameters
int voiceAchannel = 0; // Send voice A to Midi Channel 1 (zero-indexed)
int voiceBchannel = 1; // Send voice B to Midi Channel 2 (zero-indexed)
int voiceCchannel = 2; // Send voice C to Midi Channel 3 (zero-indexed)
int CC = 100; // send control change variables on CC 100

/Users/mattborland/ownCloud/TEACHING/UWTMC/Public Sessions/F2019/Session 5 - Exploration/Files/Explor
Arduino/Genuino Micro on /dev/cu.usbmodemMIDI1
```


LIBRARIES – INSTALL THESE TO ADD FUNCTIONS TO YOUR PROGRAM

.....



The screenshot shows the Arduino IDE interface. The 'Tools' menu is open, and 'Manage Libraries...' is highlighted. The background shows a code editor with C++ code for a MIDI synthesizer and a serial monitor window.

Tools Menu Options:

- Auto Format ⌘T
- Archive Sketch
- Fix Encoding & Reload
- Manage Libraries... ⌘⇧I
- Serial Monitor ⌘⇧M
- Serial Plotter ⌘⇧L
- WiFi101 Firmware Updater
- Board: "Arduino/Genuino Micro" ▶
- Port: "/dev/cu.usbmodemMIDI1 (Arduino/Genuino Micro)" ▶
- Get Board Info
- Programmer: "AVRISP mkII" ▶
- Burn Bootloader

Code Editor Content:

```
// Programmed using the known board type
// Watch out - your board type may change
// after you program it the board type may change

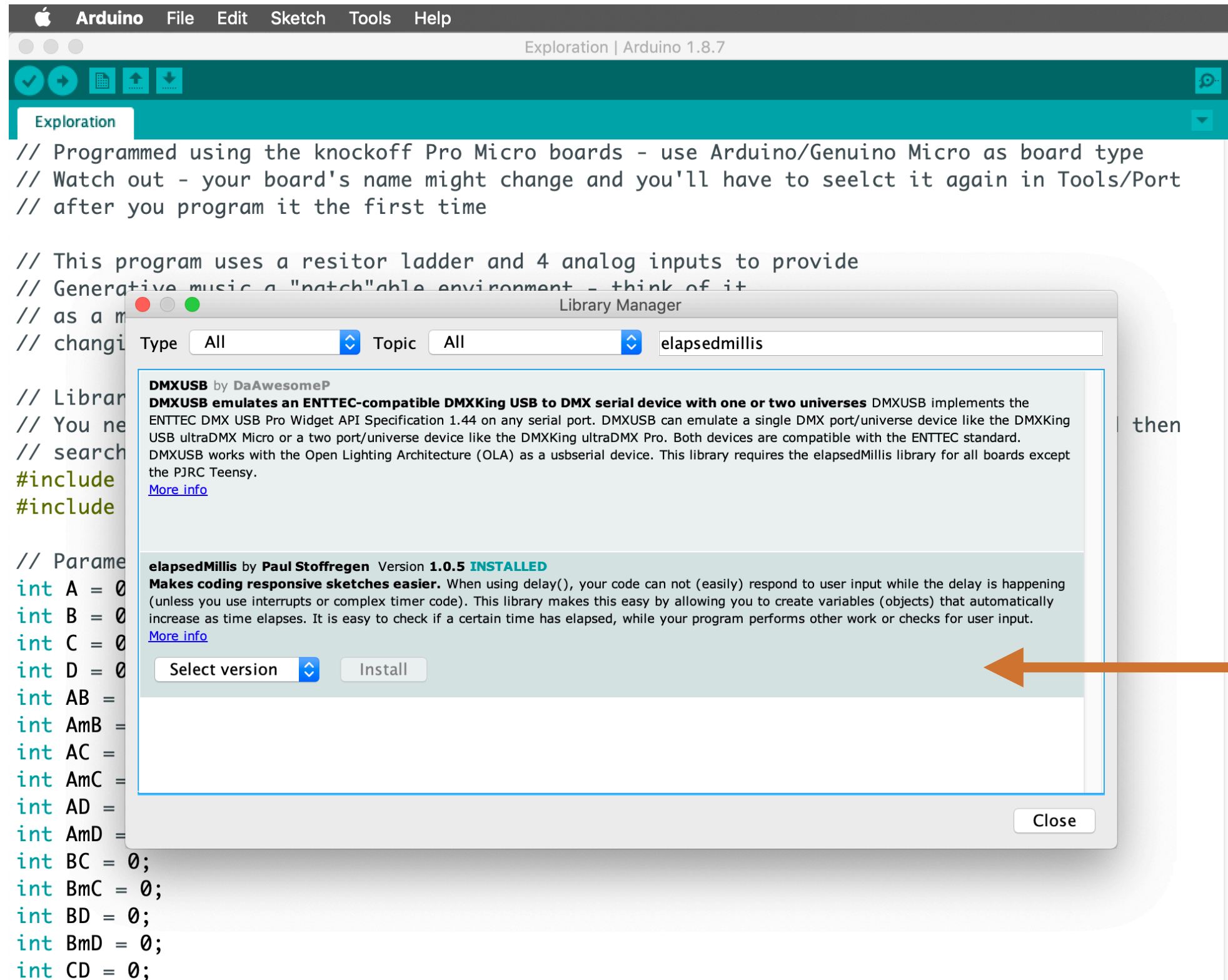
// This program uses a resampling algorithm
// Generative music a "pattern"
// as a modular synthesizer
// changing the algorithm

// Libraries
// You need to install both libraries - search for them under Tools/Manage Libraries, and then
// search for the word in front of the .h
#include <elapsedMillis.h>
#include <MIDIUSB.h>
```

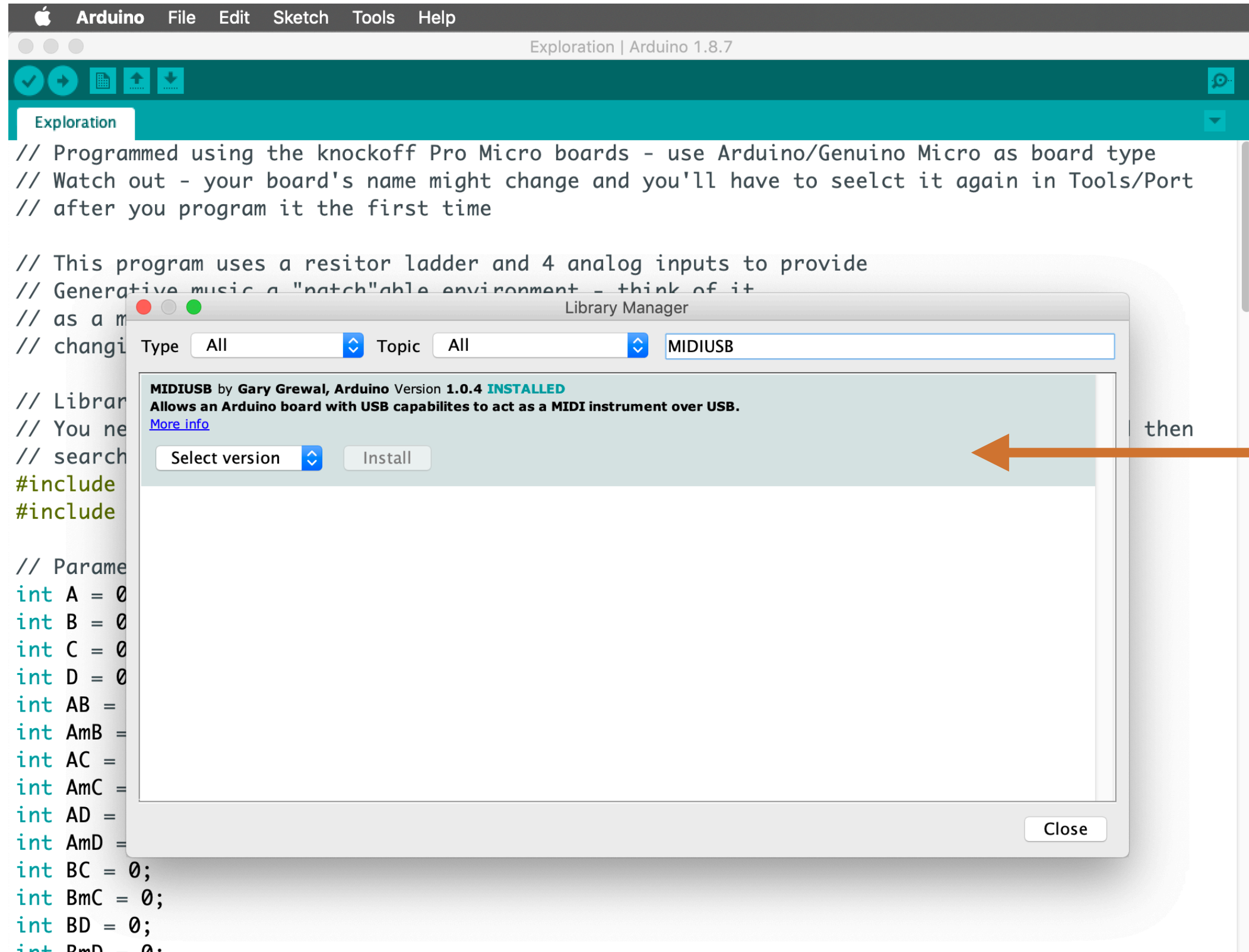
Serial Monitor Content:

```
Micro as board type
t again in Tools/Port
```

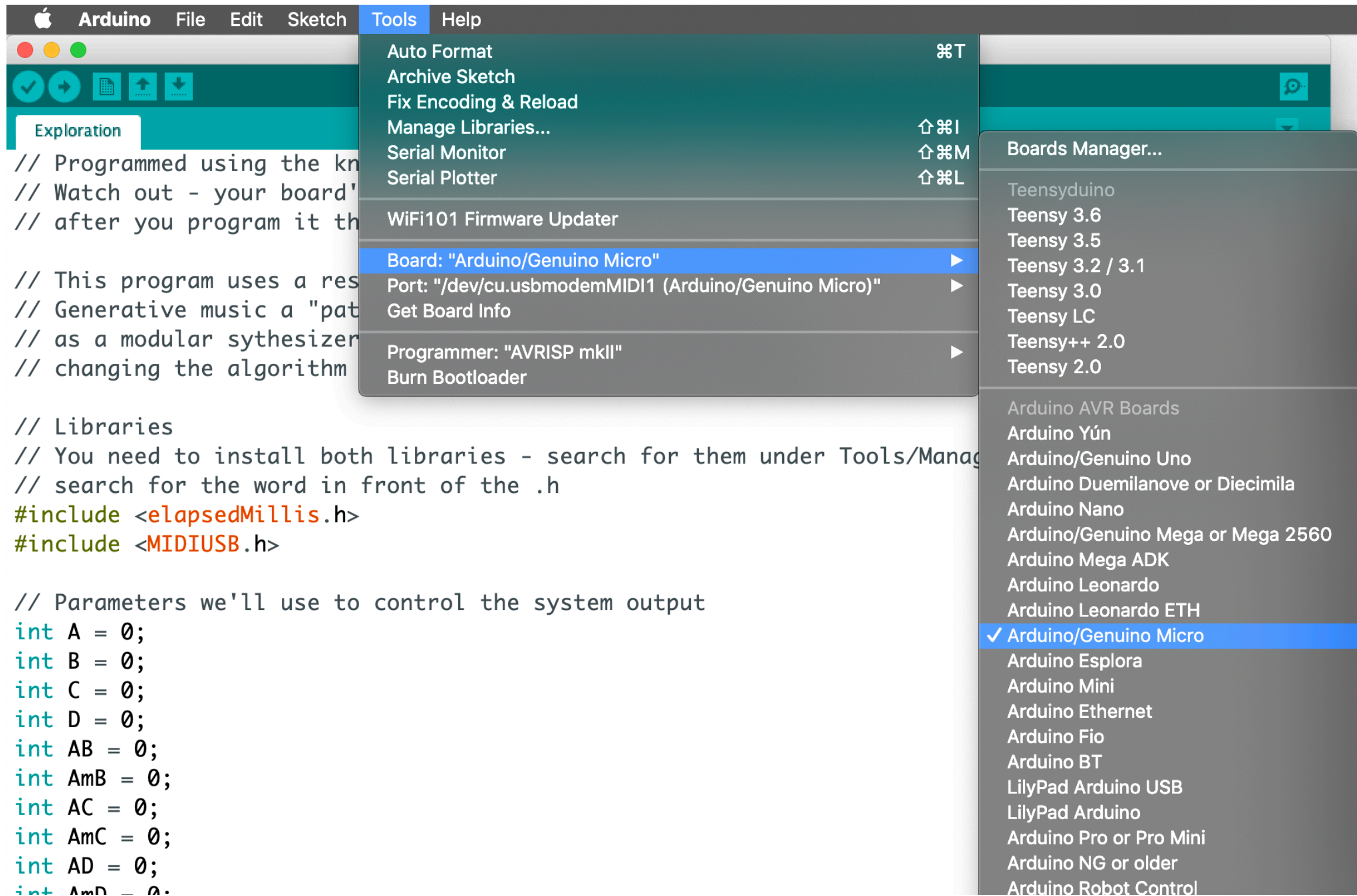

LIBRARIES – ELAPSEDMILLIS



LIBRARIES – MIDIUSB



PROGRAM YOUR DEVICE – BOARD TYPE

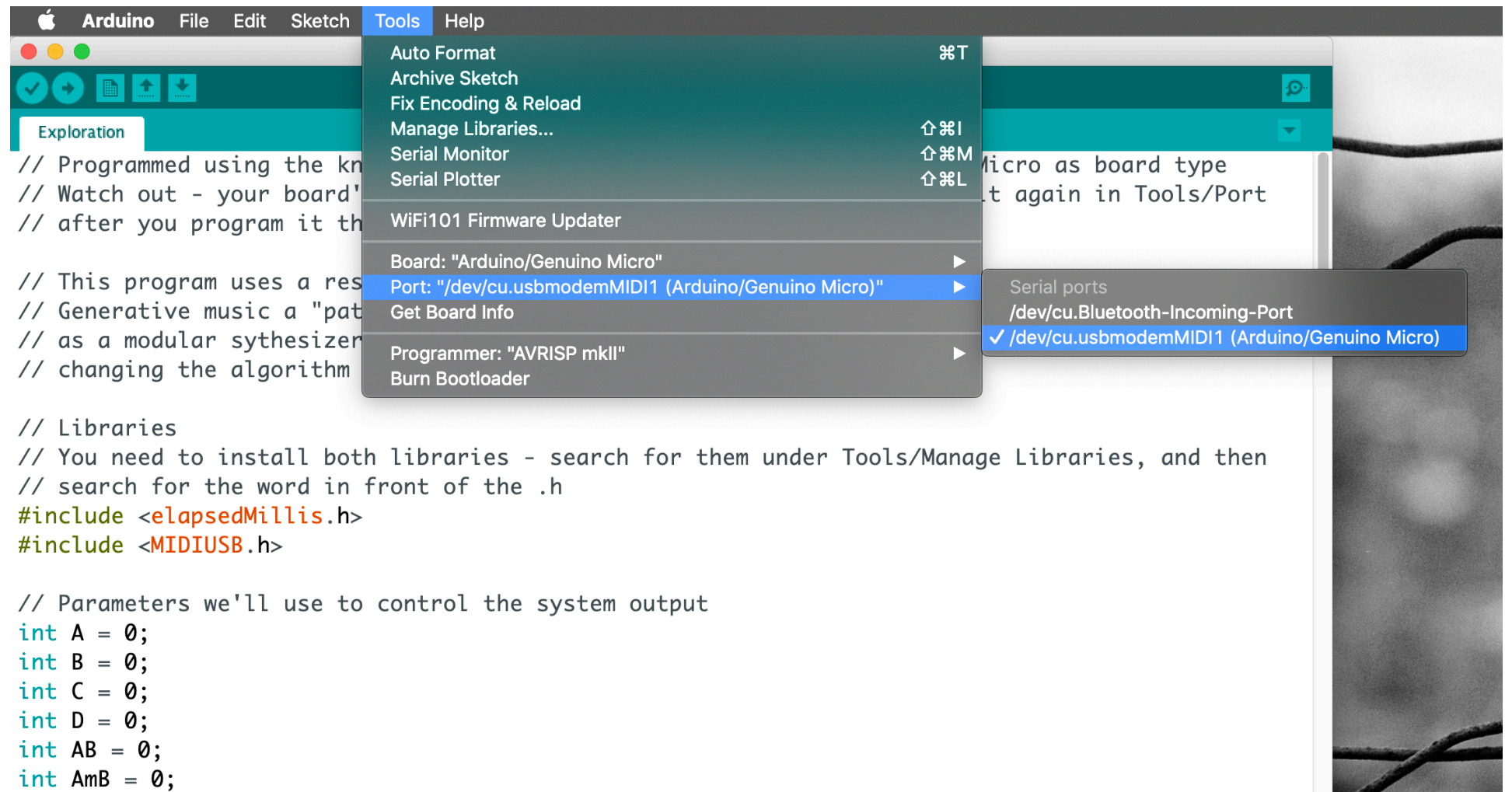


PROGRAM YOUR DEVICE – PORT

.....

Sometime this name changes after programming, so you may need to reselect this from time to time.

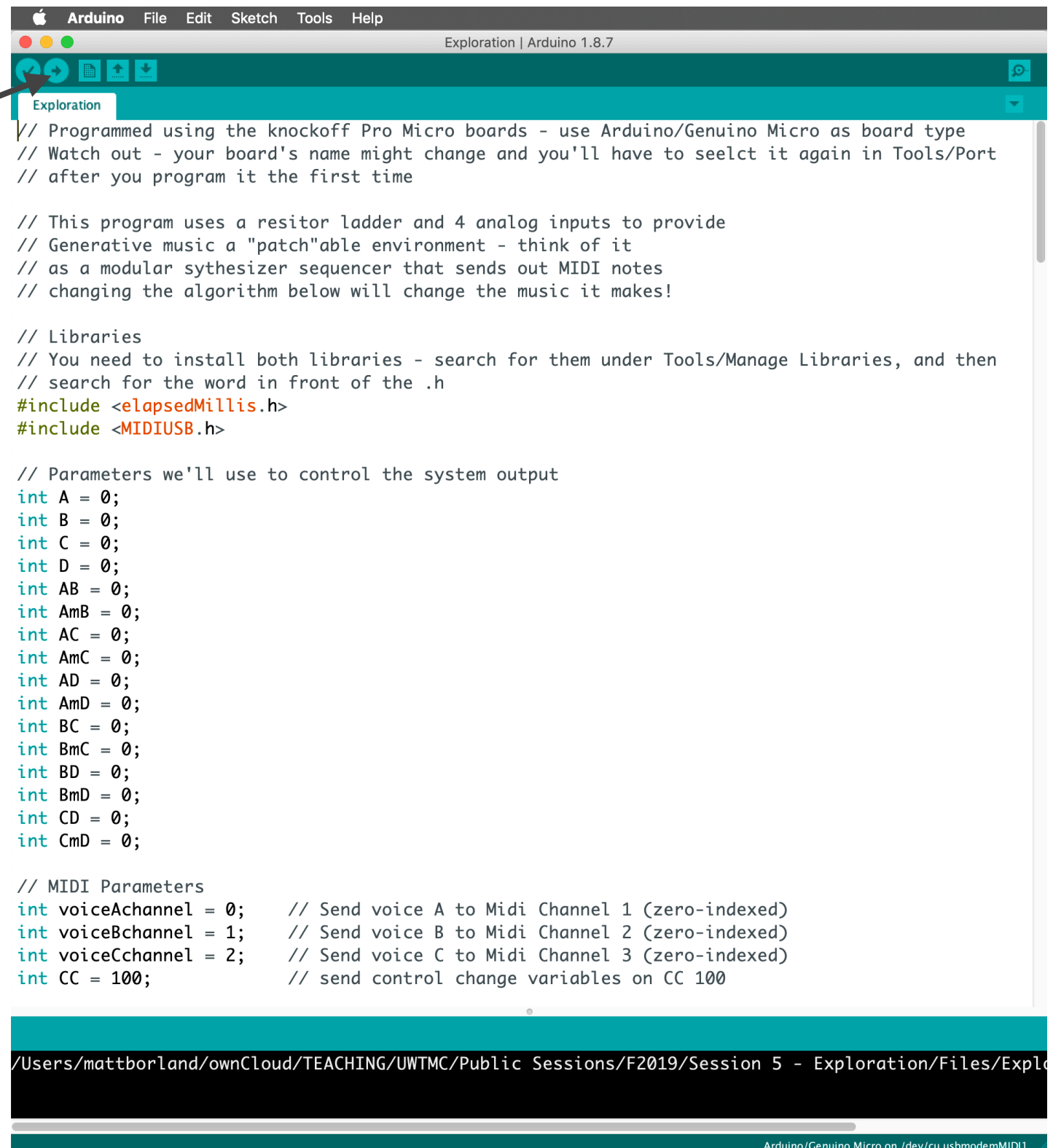
If you get an error that says “board not found” you should check it’s plugged in and that your port is selected correctly.



PROGRAM YOUR DEVICE – EXPLORATION.INO

.....

With your arduino plugged in, press the ARROW button to compile the code and send it to the arduino.



```
Arduino File Edit Sketch Tools Help
Exploration | Arduino 1.8.7

Exploration
// Programmed using the knockoff Pro Micro boards - use Arduino/Genuino Micro as board type
// Watch out - your board's name might change and you'll have to select it again in Tools/Port
// after you program it the first time

// This program uses a resistor ladder and 4 analog inputs to provide
// Generative music a "patch"able environment - think of it
// as a modular synthesizer sequencer that sends out MIDI notes
// changing the algorithm below will change the music it makes!

// Libraries
// You need to install both libraries - search for them under Tools/Manage Libraries, and then
// search for the word in front of the .h
#include <elapsedMillis.h>
#include <MIDIUSB.h>

// Parameters we'll use to control the system output
int A = 0;
int B = 0;
int C = 0;
int D = 0;
int AB = 0;
int AmB = 0;
int AC = 0;
int AmC = 0;
int AD = 0;
int AmD = 0;
int BC = 0;
int BmC = 0;
int BD = 0;
int BmD = 0;
int CD = 0;
int CmD = 0;

// MIDI Parameters
int voiceAchannel = 0; // Send voice A to Midi Channel 1 (zero-indexed)
int voiceBchannel = 1; // Send voice B to Midi Channel 2 (zero-indexed)
int voiceCchannel = 2; // Send voice C to Midi Channel 3 (zero-indexed)
int CC = 100; // send control change variables on CC 100

/Users/mattborland/ownCloud/TEACHING/UWTMC/Public Sessions/F2019/Session 5 - Exploration/Files/Explor
Arduino/Genuino Micro on /dev/cu.usbmodemMIDI1
```

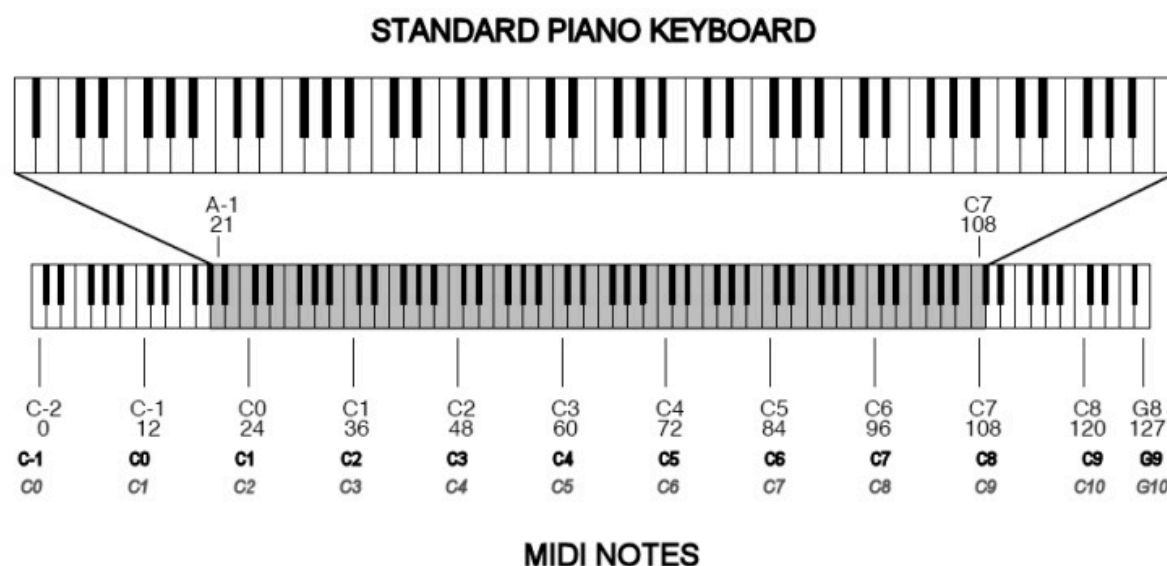

CHANGE YOUR NOTES!

.....

There is a set of 16 notes that the instrument will pick from. You can change these notes to change the key or harmonies that are created. You don't have to have 16 unique notes, they can all be the same, or can be repeated.

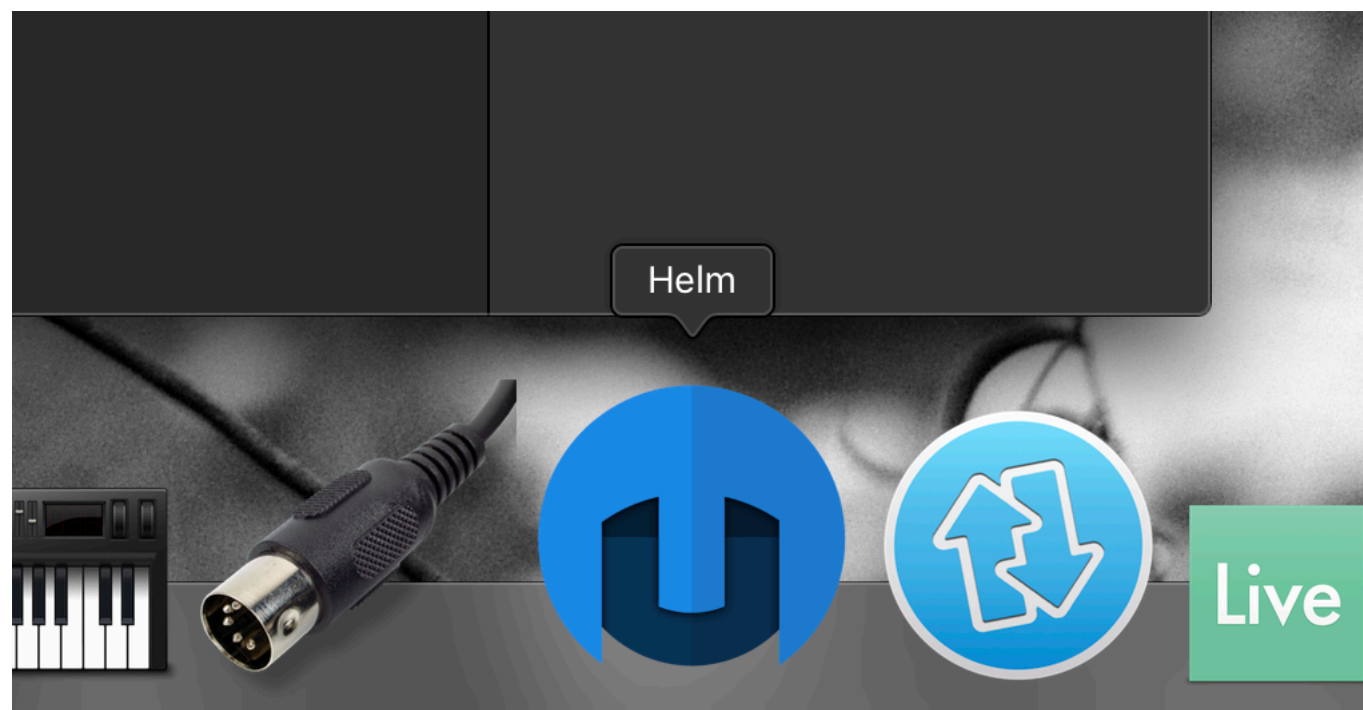
YOU DO HAVE TO HAVE 16 NOTES or the code might break and cause an error that crashes your instrument.

```
|  
// You can change which pitch plays by changing these MIDI Note Numbers  
// You need to have 16 numbers here. They don't all have to be different, but there  
// have to be 16 total.  
byte pitch[] = {43, 47, 48, 52, 55, 59, 60, 64, 67, 71, 72, 76, 79, 83, 84, 88};
```



<https://en.wikipedia.org/wiki/MIDI>

OPEN HELM – SOFTWARE SYNTHESIZER



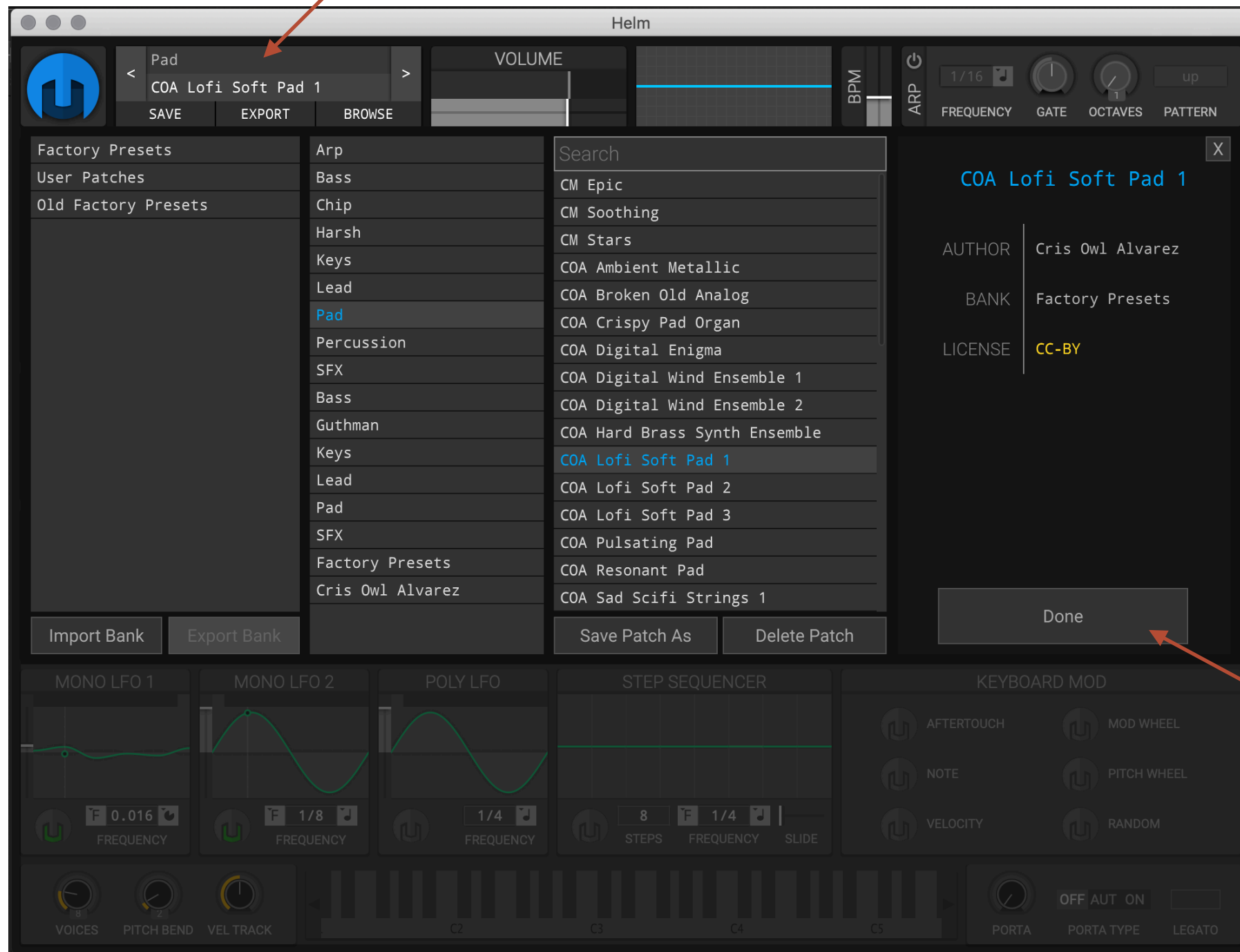
*Helm - a software synthesizer
to make musical sounds with
your computer.*

<https://tytel.org/helm/>

SETUP HELM

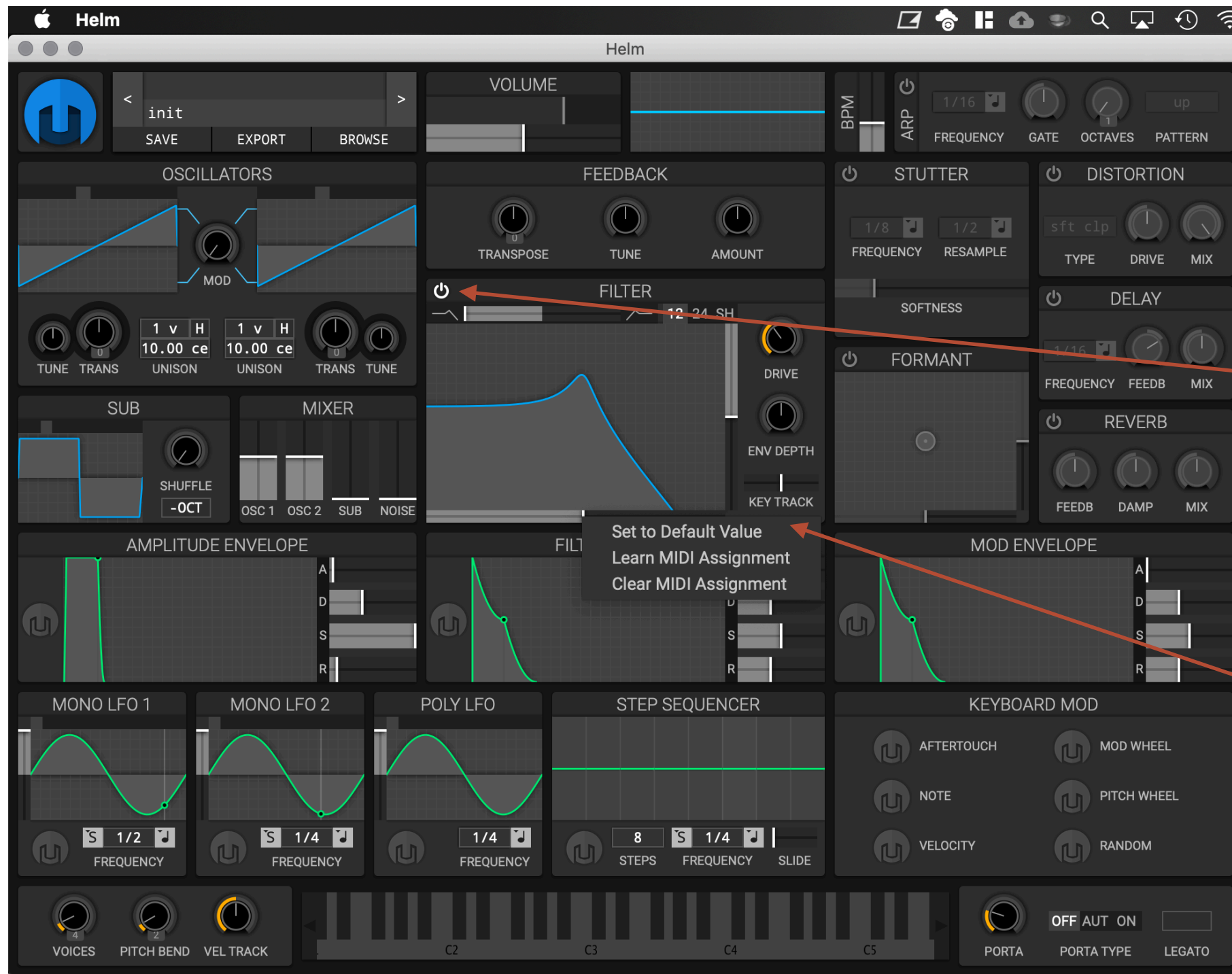
Click here to open the menu to select a sound

Make sure your laptop's speakers are on and turned up!



Try different sounds, then click done.

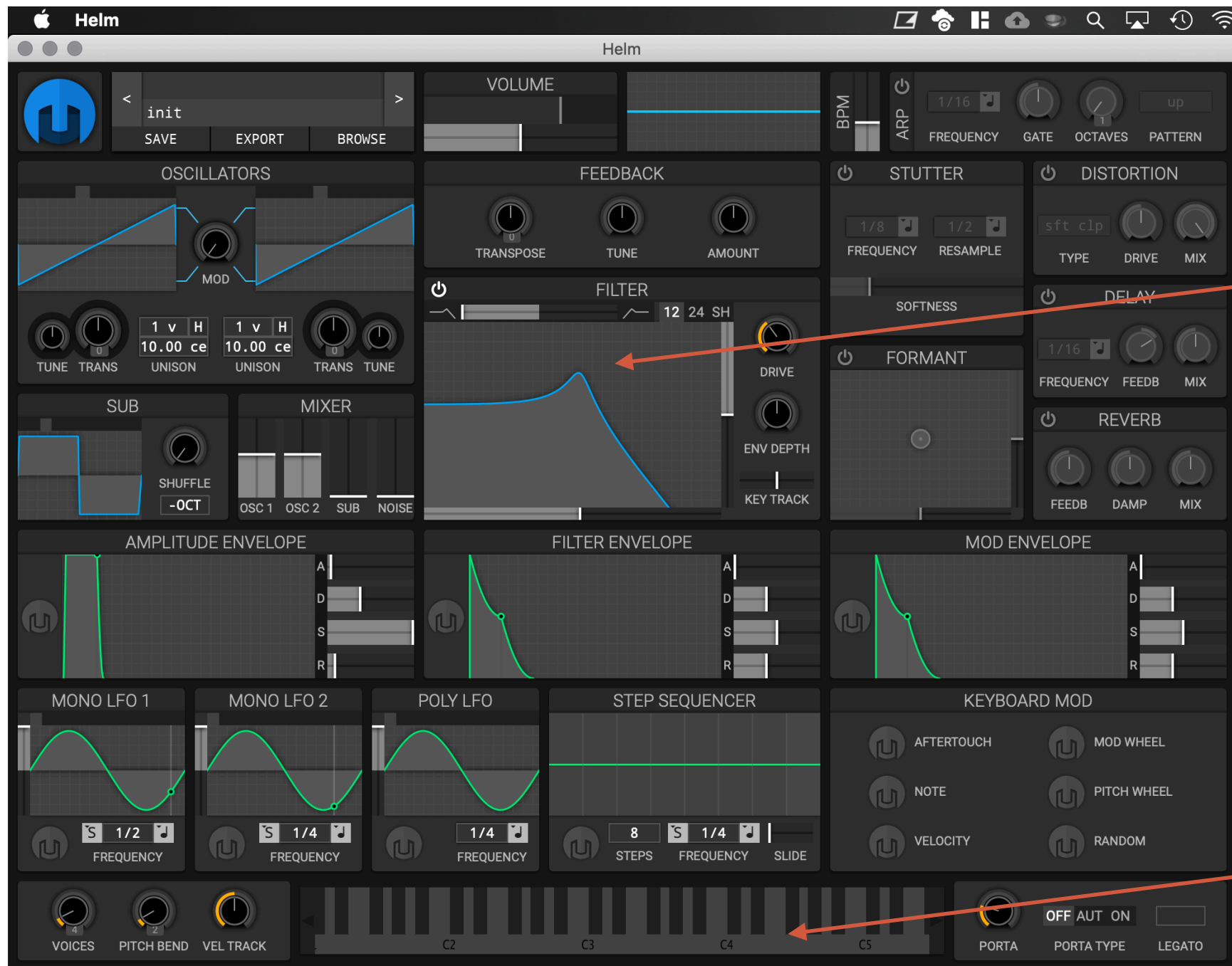
SETUP HELM



Make sure the filter is on - it shouldn't be greyed out. If it is greyed out, click the ON button here.

Right Click the bar below the filter, then click "Learn MIDI Assignment"

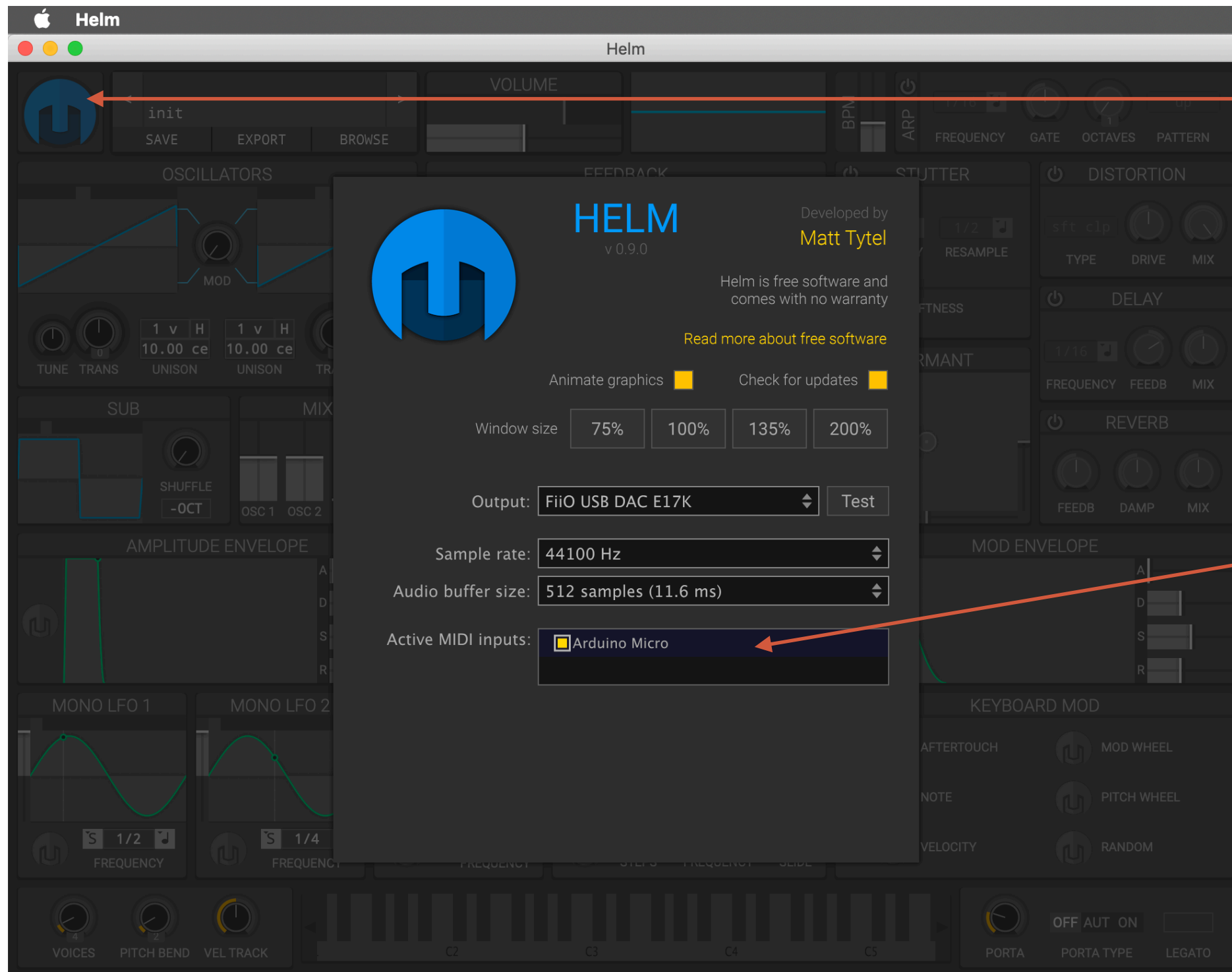
SETUP HELM



You should see the filter peak move back and forth if you have wires plugged in anywhere other than having all four in the left most pin of the voltage divider!

You should see the keys of the piano lighting up from time to time.

NOT WORKING? CHECK THE MIDI PORT



Click the HELMet to get this dialog to open.

You should see “Arduino Micro” as a MIDI input, with a yellow box showing it is selected.