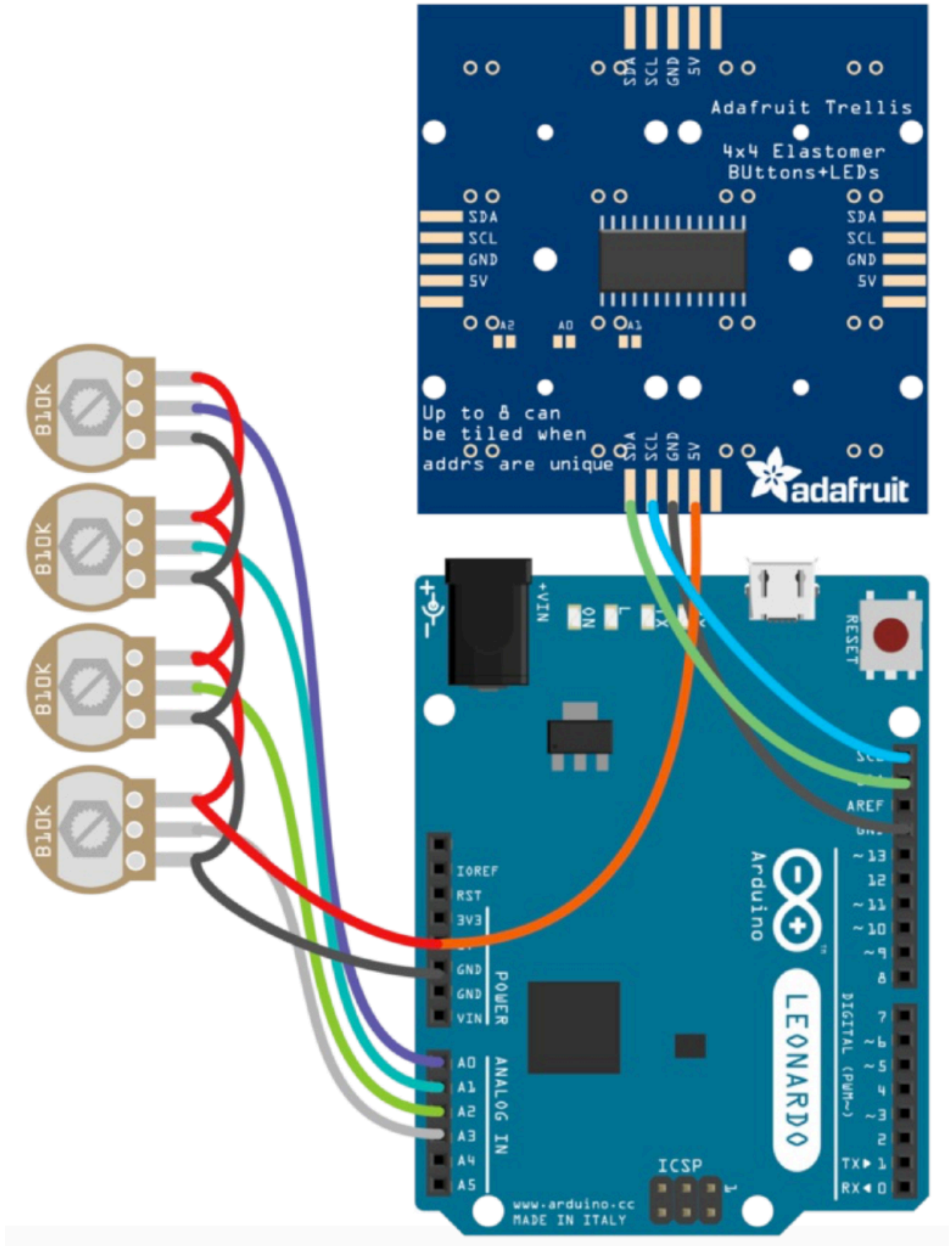
DIY Midi Controller

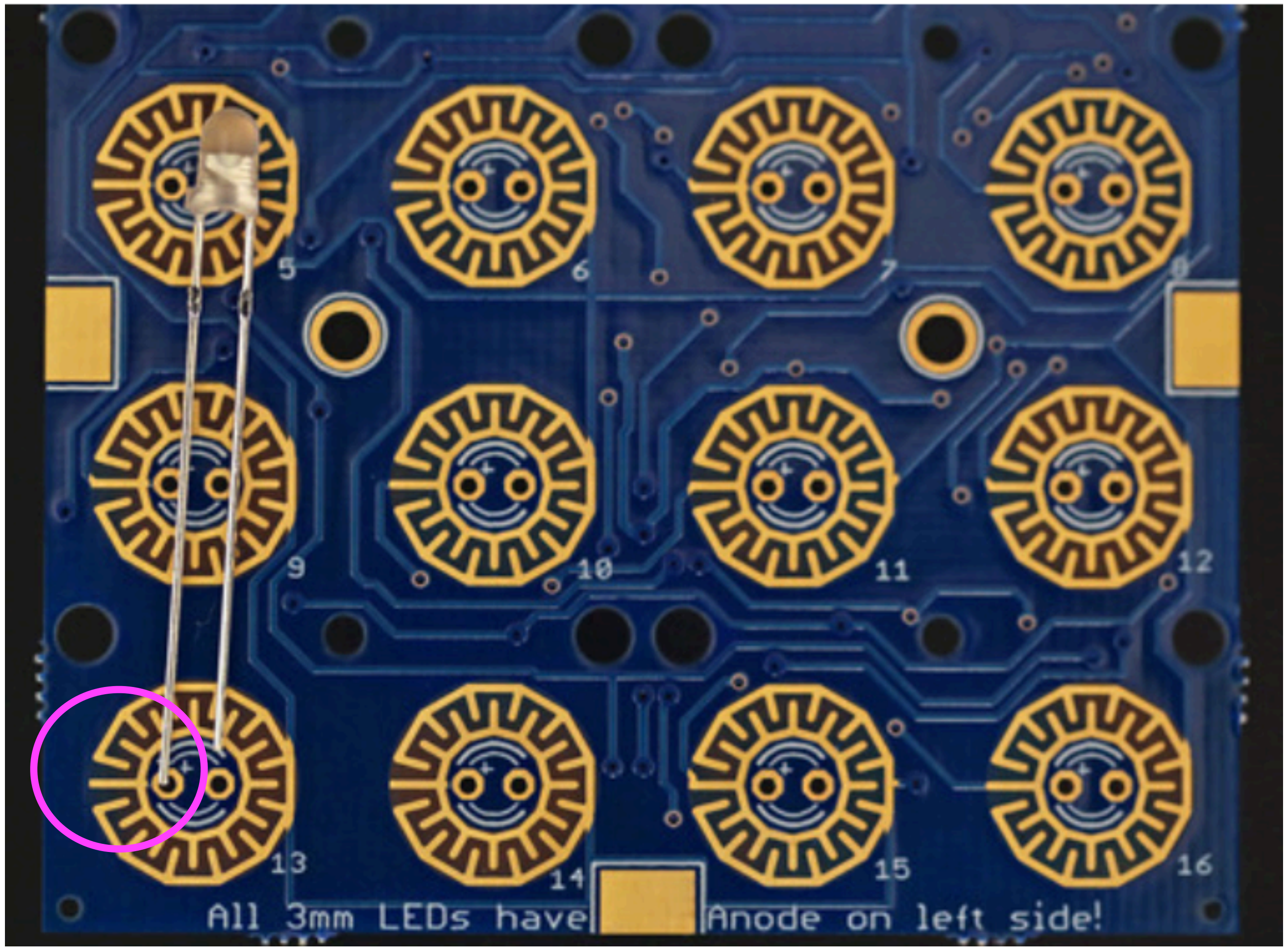
The Edge

Circuit Diagram



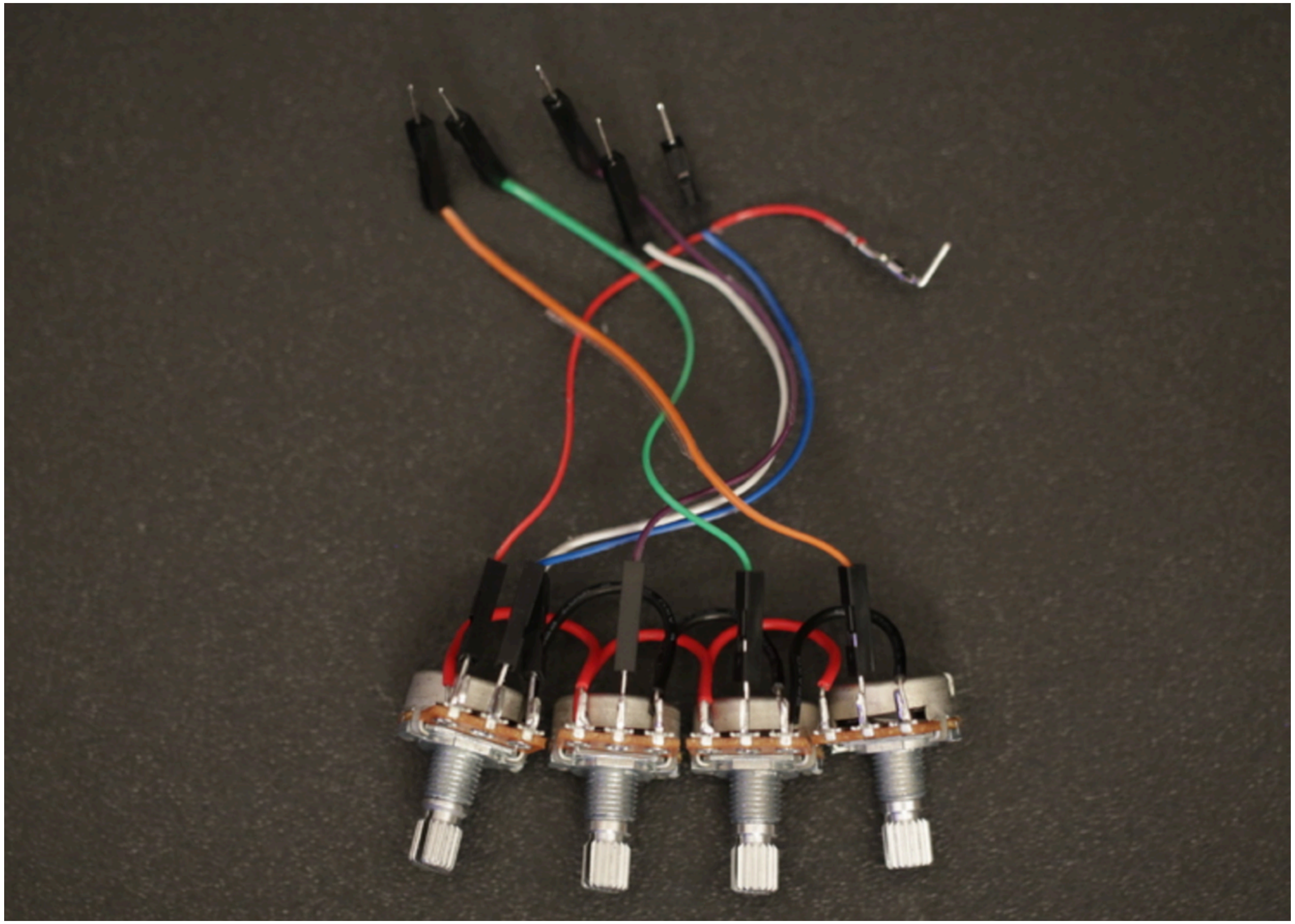
- Source: https://cdn-learn.adafruit.com/downloads/pdf/mini-untztrument-3d-printed-midi-controller.pdf

Construction



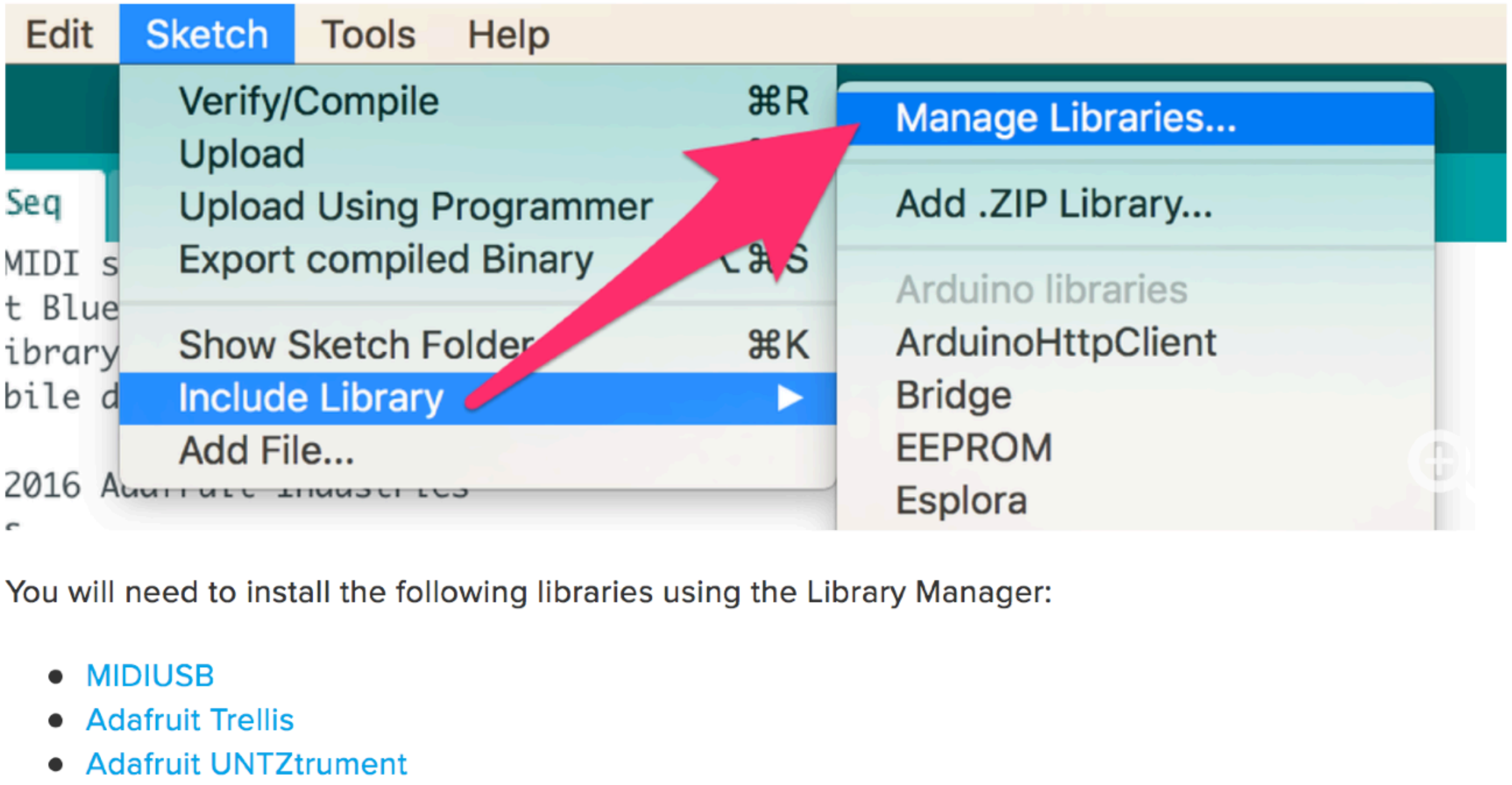
- Longer leg of LED goes into positive pin of PCB
- Source: https://learn.adafruit.com/mini-untztrument-3d-printed-midi-controller/prep-components

Construction



- Source: https://www.thingiverse.com/thing:409733

Add Libraries



You will need to install the following libraries using the Library Manager:

- MIDIUSB
- Adafruit Trellis
- Adafruit UNTZtrument

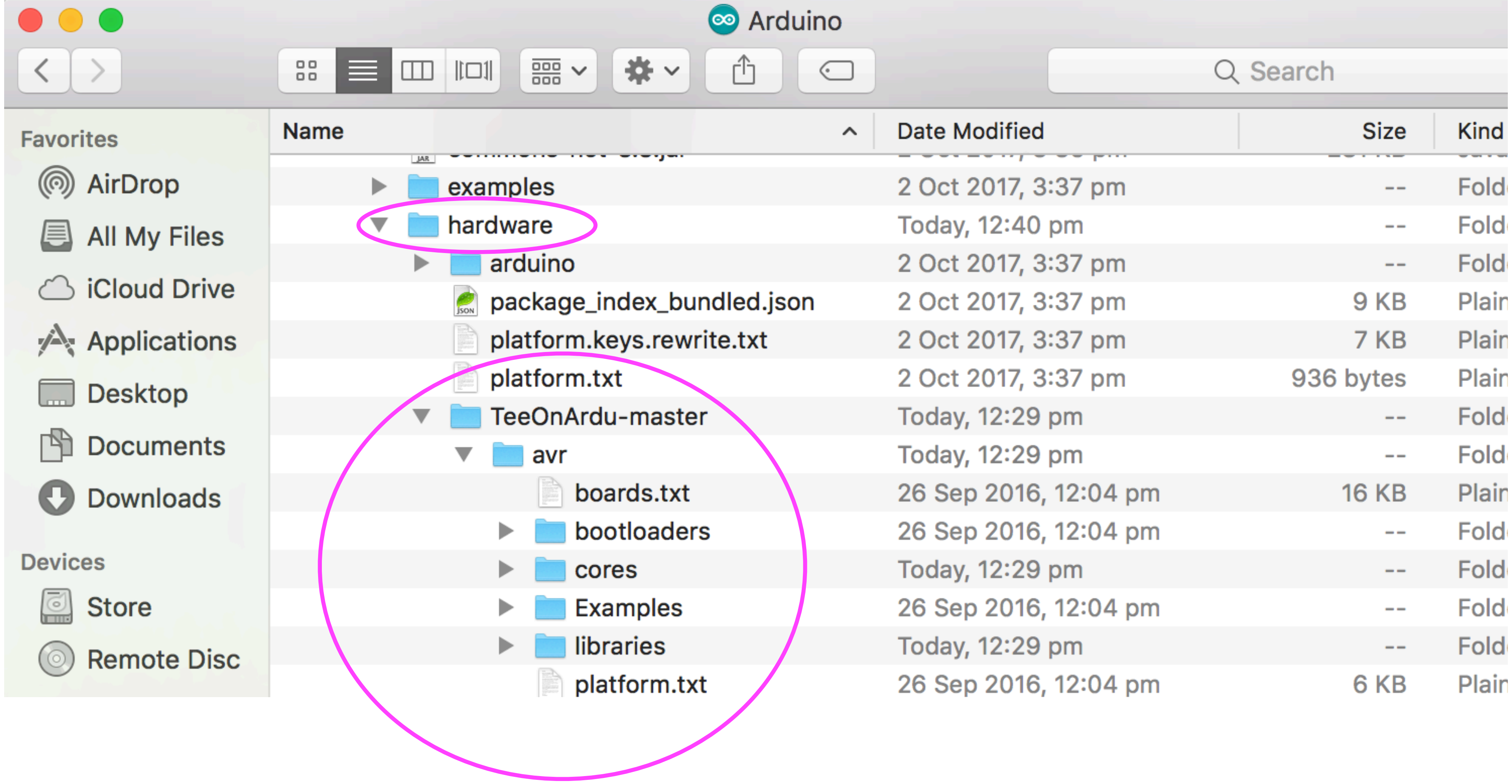
- Source: https://learn.adafruit.com/untztrument-trellis-midi-instrument/code

TeeOnArdu installation

1) Create "avr" folder inside TeeOnArdu-master folder and put all contents inside "avr" folder

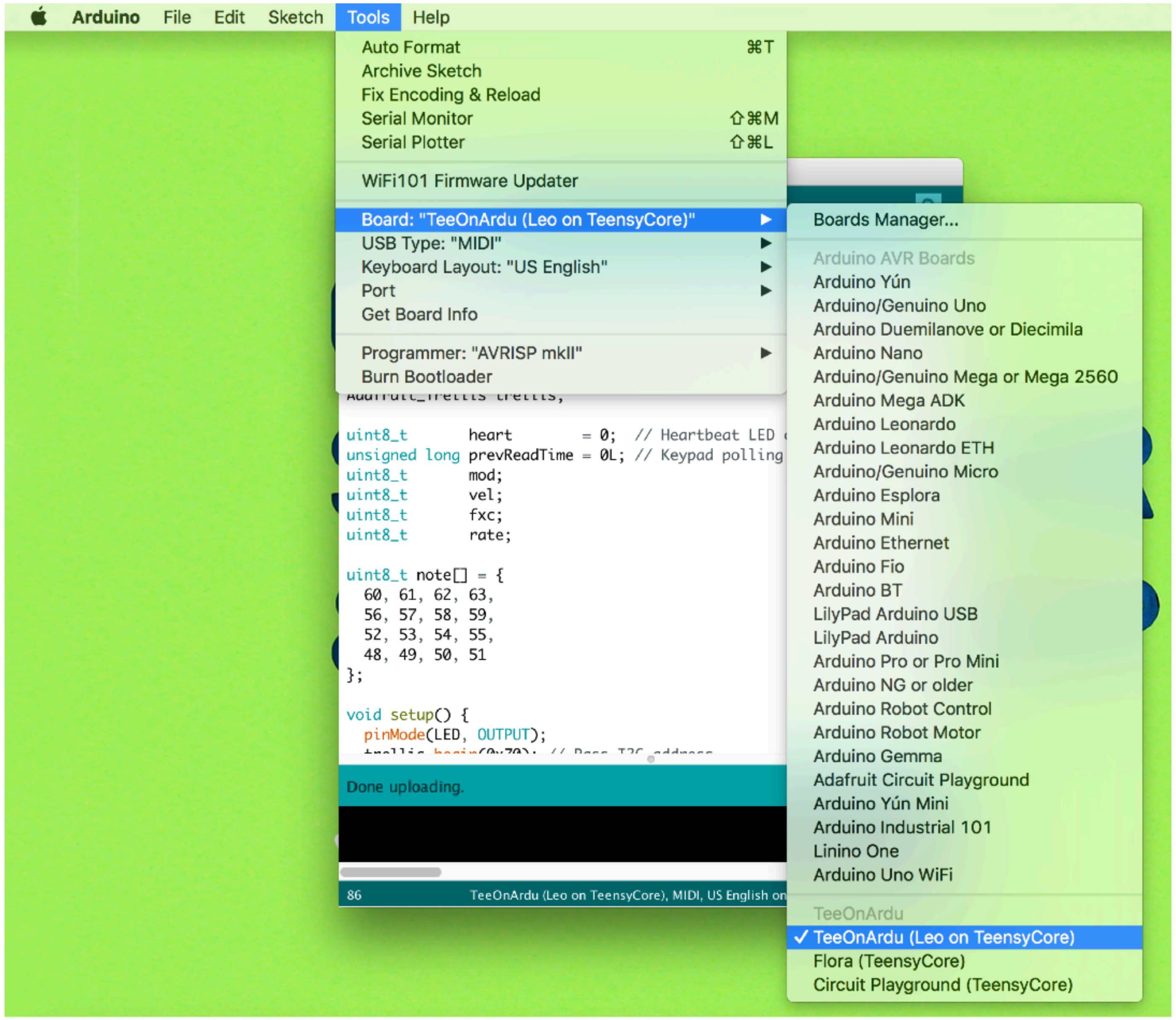
TeeOnArdu: https://github.com/adafruit/TeeOnArdu

2) Put TeeOnArdu-master folder inside "hardware" folder (on Mac have to right click on Arduino app and "show package contents" and navigate to "hardware" folder [Arduino.app/Contents/Java/hardware])



Name	Date Modified	Size	Kind
▶ examples	2 Oct 2017, 3:37 pm	--	Fold
▼ hardware	Today, 12:40 pm	--	Fold
▶ arduino	2 Oct 2017, 3:37 pm	--	Fold
package_index_bundled.json	2 Oct 2017, 3:37 pm	9 KB	Plain
platform.keys.rewrite.txt	2 Oct 2017, 3:37 pm	7 KB	Plain
platform.txt	2 Oct 2017, 3:37 pm	936 bytes	Plain
▼ TeeOnArdu-master	Today, 12:29 pm	--	Fold
▼ avr	Today, 12:29 pm	--	Fold
boards.txt	26 Sep 2016, 12:04 pm	16 KB	Plain
▶ bootloaders	26 Sep 2016, 12:04 pm	--	Fold
▶ cores	Today, 12:29 pm	--	Fold
▶ Examples	26 Sep 2016, 12:04 pm	--	Fold
▶ libraries	Today, 12:29 pm	--	Fold
platform.txt	26 Sep 2016, 12:04 pm	6 KB	Plain

TeeOnArdu installation

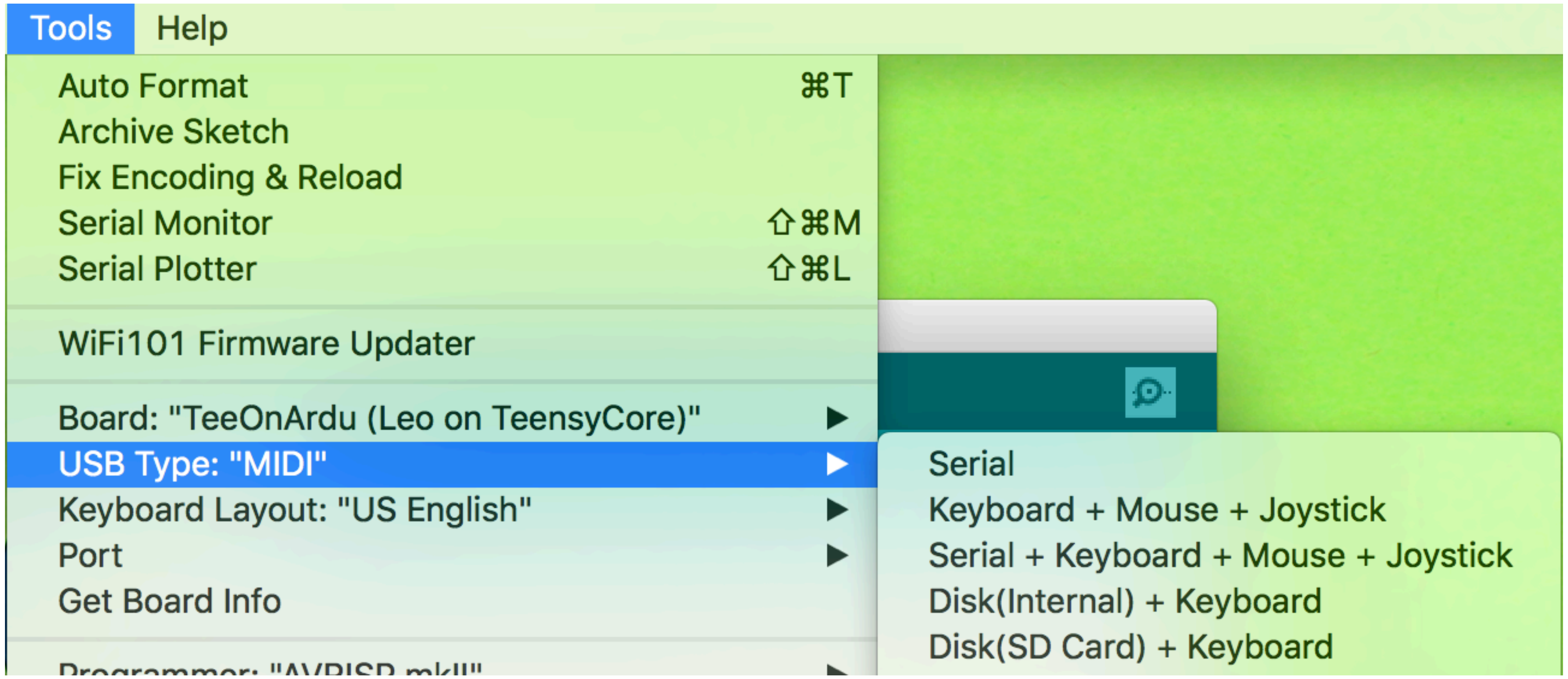


```
Adafruit_Trellis trellis;

uint8_t       heart        = 0;  // Heartbeat LED
unsigned long prevReadTime = 0L; // Keypad polling
uint8_t       mod;
uint8_t       vel;
uint8_t       fxc;
uint8_t       rate;

uint8_t note[] = {
  60, 61, 62, 63,
  56, 57, 58, 59,
  52, 53, 54, 55,
  48, 49, 50, 51
};

void setup() {
  pinMode(LED, OUTPUT);
  trellis.begin(0x70); // Pass I2C address
```

Done uploading.

86 TeeOnArdu (Leo on TeensyCore), MIDI, US English on

TeeOnArdu installation



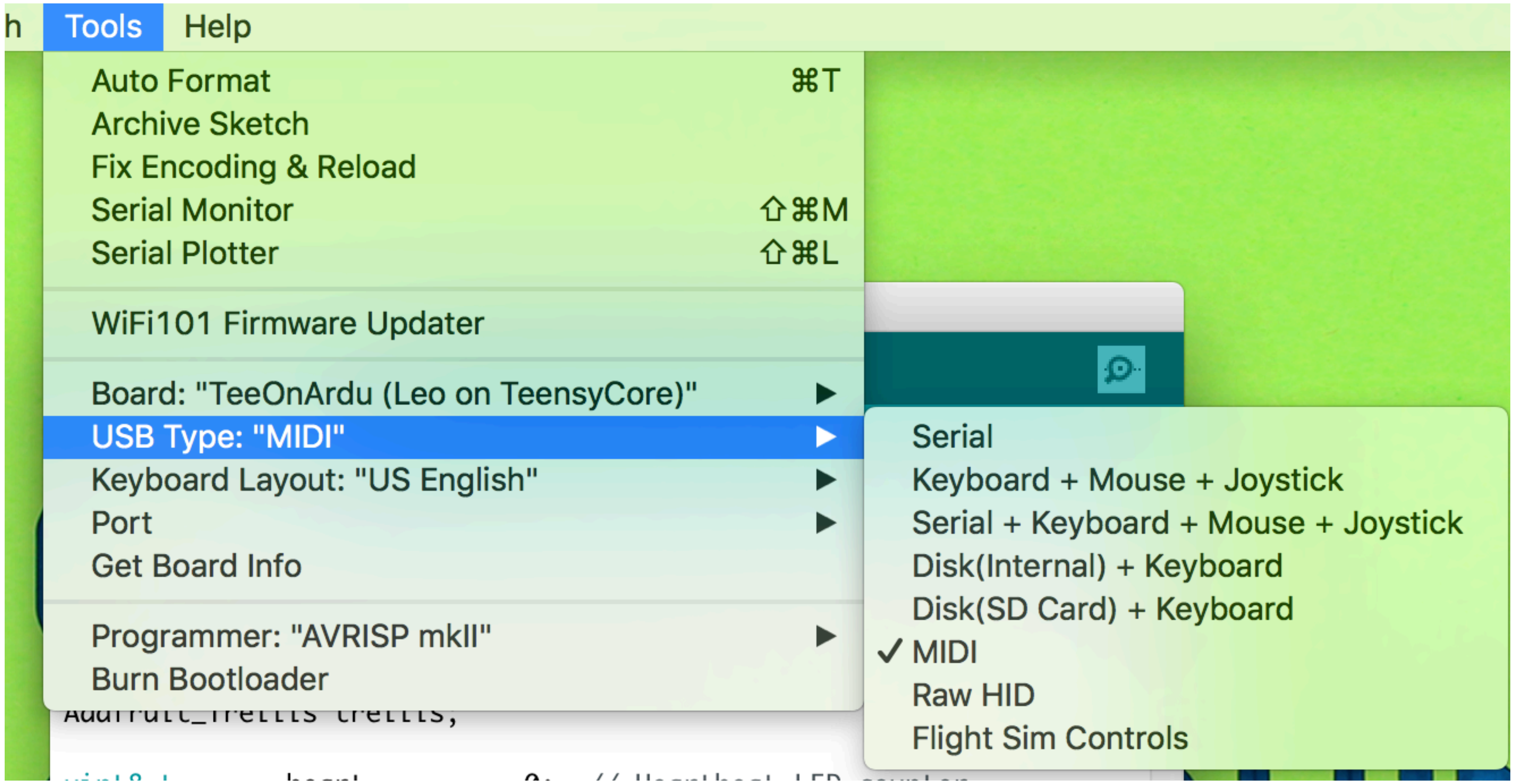
Board: "TeeOnArdu (Leo on TeensyCore)" ▶

USB Type: "MIDI" ▶

Keyboard Layout: "US English" ▶

Port ▶
Get Board Info
Serial
Keyboard + Mouse + Joystick
Serial + Keyboard + Mouse + Joystick
Disk(Internal) + Keyboard
Disk(SD Card) + Keyboard

TeeOnArdu installation



Board: "TeeOnArdu (Leo on TeensyCore)" ▶

USB Type: "MIDI" ▶

Keyboard Layout: "US English" ▶

Programmer: "AVRISP mkII" ▶

MINI UNTZtrument sketch

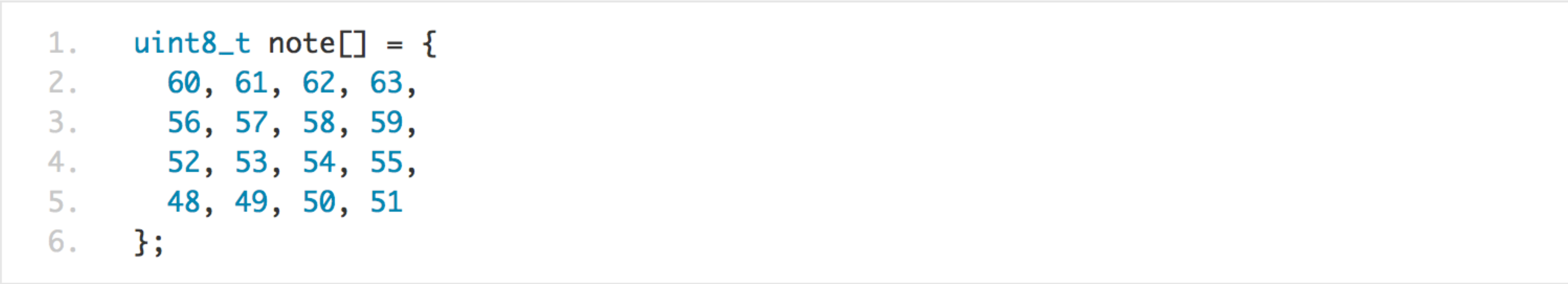
MINI UNTZtrument Sketch: https://learn.adafruit.com/mini-untztrument-3d-printed-midi-controller/software

Midi

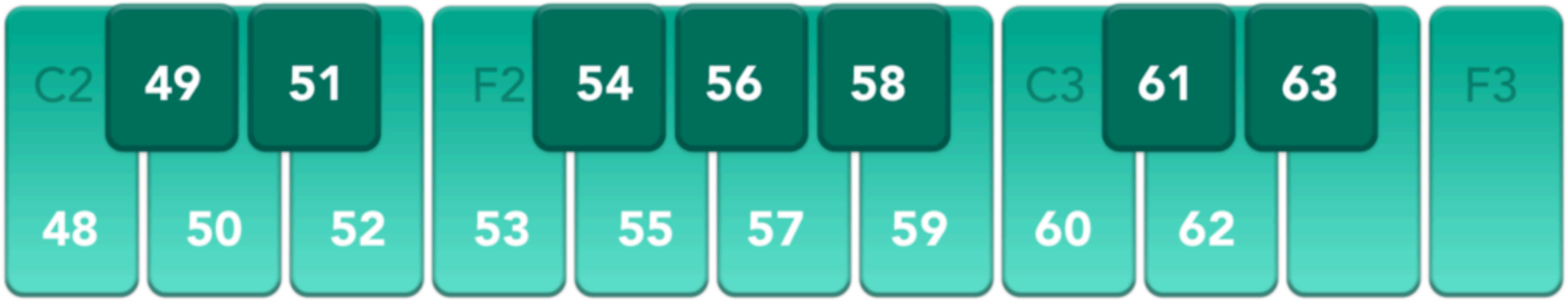
MIDI (/ˈmɪdi/; short for **Musical Instrument Digital Interface**) is a technical standard that describes a communications protocol, digital interface and electrical connectors and allows a wide variety of electronic musical instruments, computers and other related music and audio devices to connect and communicate with one another.

Source: https://en.wikipedia.org/wiki/MIDI

MIDI Note Mapping

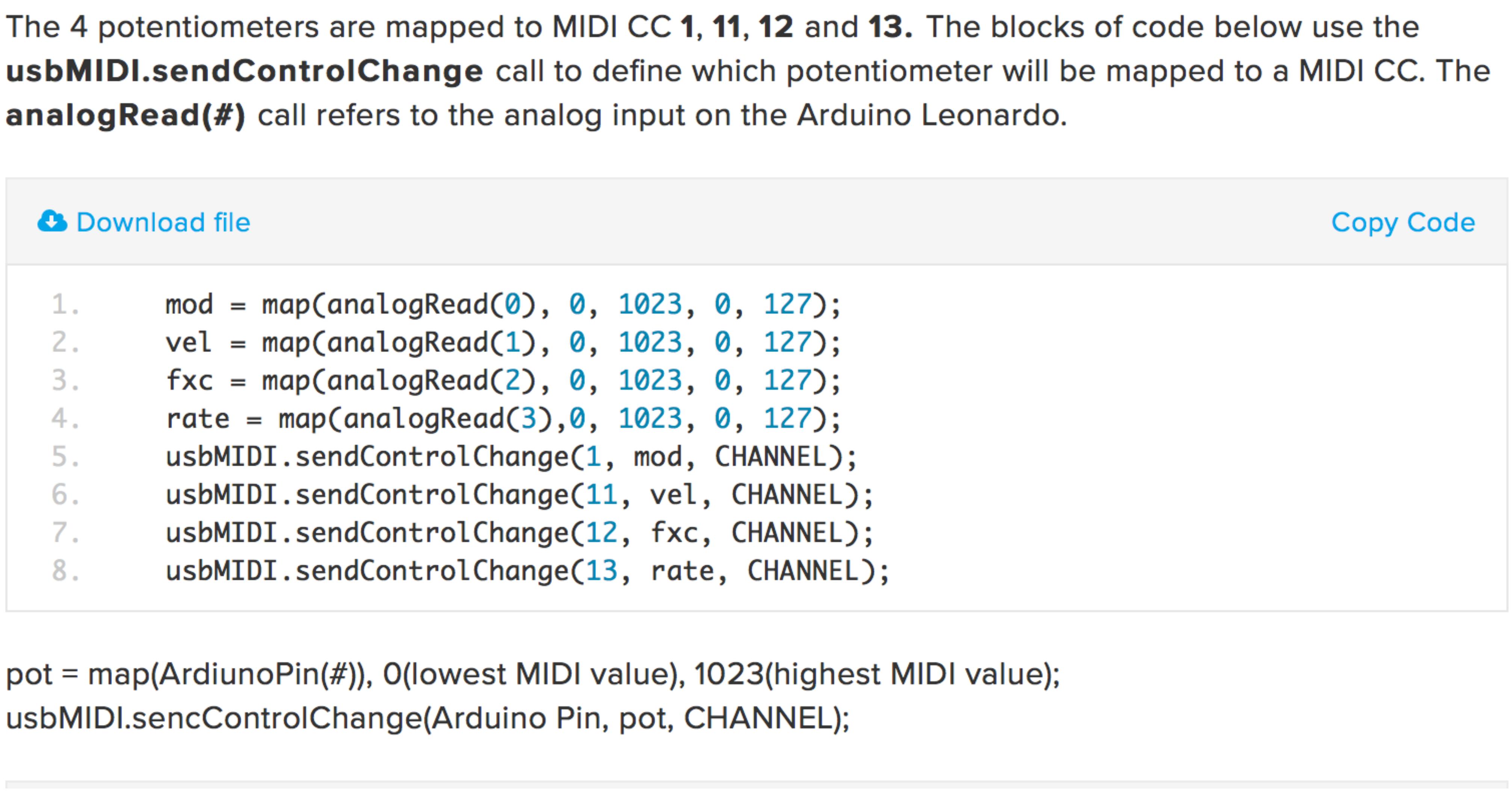


```
uint8_t note[] = {
  60, 61, 62, 63,
  56, 57, 58, 59,
  52, 53, 54, 55,
  48, 49, 50, 51
};
```



- Source: https://learn.adafruit.com/mini-untztrument-3d-printed-midi-controller/software

Potentiometers Midi CC (‘Control change’ messages)



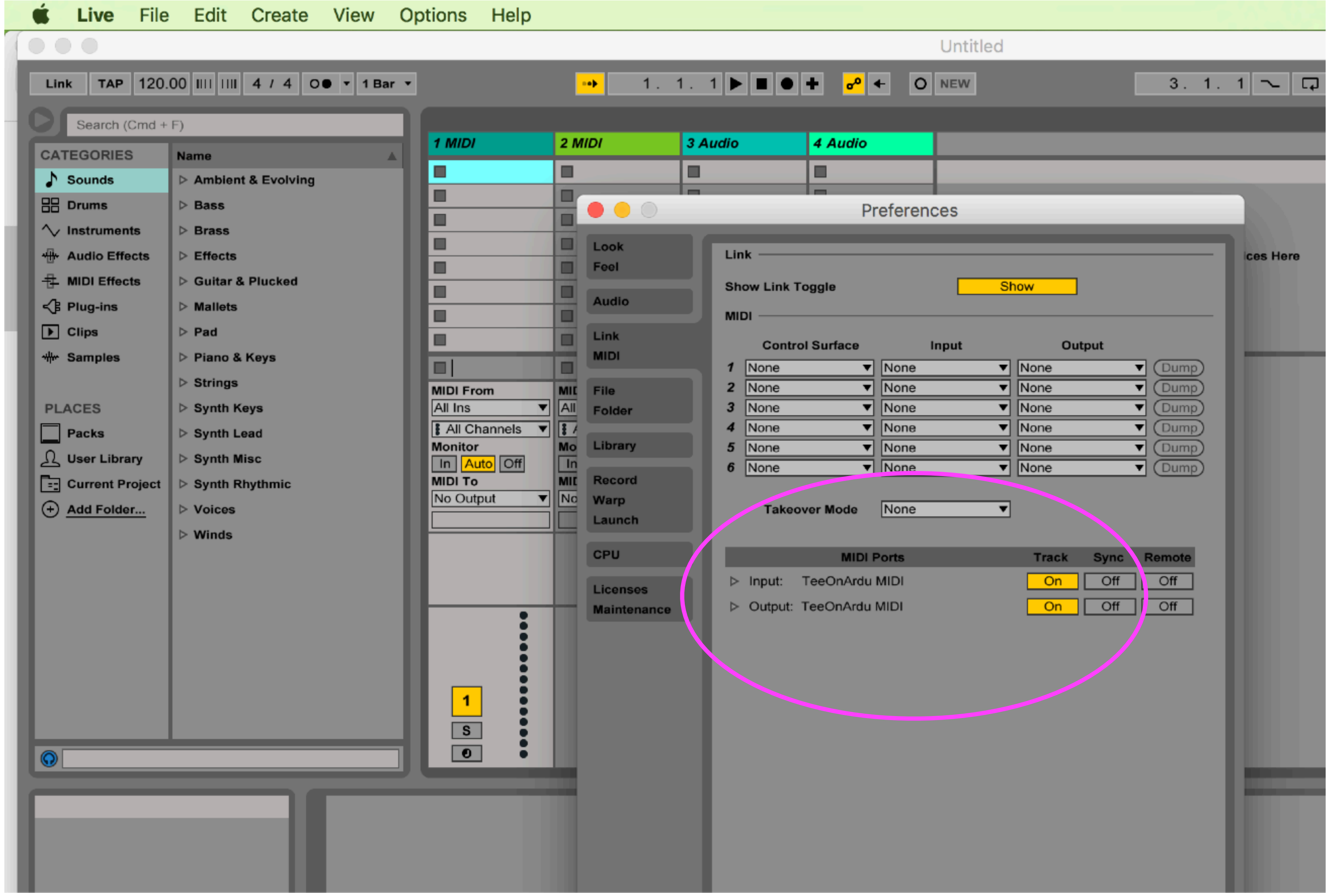
The 4 potentiometers are mapped to MIDI CC **1**, **11**, **12** and **13.** The blocks of code below use the **usbMIDI.sendControlChange** call to define which potentiometer will be mapped to a MIDI CC. The **analogRead(#)** call refers to the analog input on the Arduino Leonardo.

```
mod = map(analogRead(0), 0, 1023, 0, 127);
vel = map(analogRead(1), 0, 1023, 0, 127);
fxc = map(analogRead(2), 0, 1023, 0, 127);
rate = map(analogRead(3),0, 1023, 0, 127);
usbMIDI.sendControlChange(1, mod, CHANNEL);
usbMIDI.sendControlChange(11, vel, CHANNEL);
usbMIDI.sendControlChange(12, fxc, CHANNEL);
usbMIDI.sendControlChange(13, rate, CHANNEL);
```

pot = map(ArdiunoPin(#)), 0(lowest MIDI value), 1023(highest MIDI value);
usbMIDI.sencControlChange(Arduino Pin, pot, CHANNEL);

- Source: https://learn.adafruit.com/mini-untztrument-3d-printed-midi-controller/software

Connecting to Digital Audio Workstation (DAW)



Link

Show Link Toggle: Show

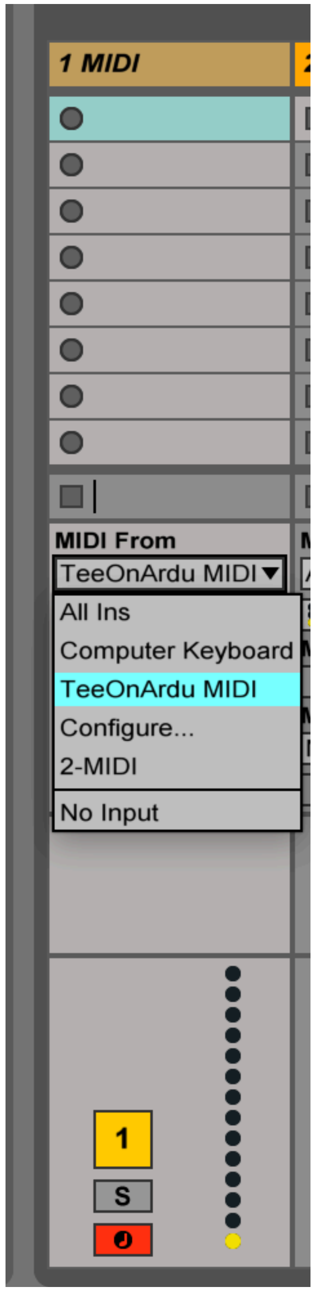
MIDI

	Control Surface	Input	Output	
1	None	None	None	Dump
2	None	None	None	Dump
3	None	None	None	Dump
4	None	None	None	Dump
5	None	None	None	Dump
6	None	None	None	Dump

Takeover Mode: None

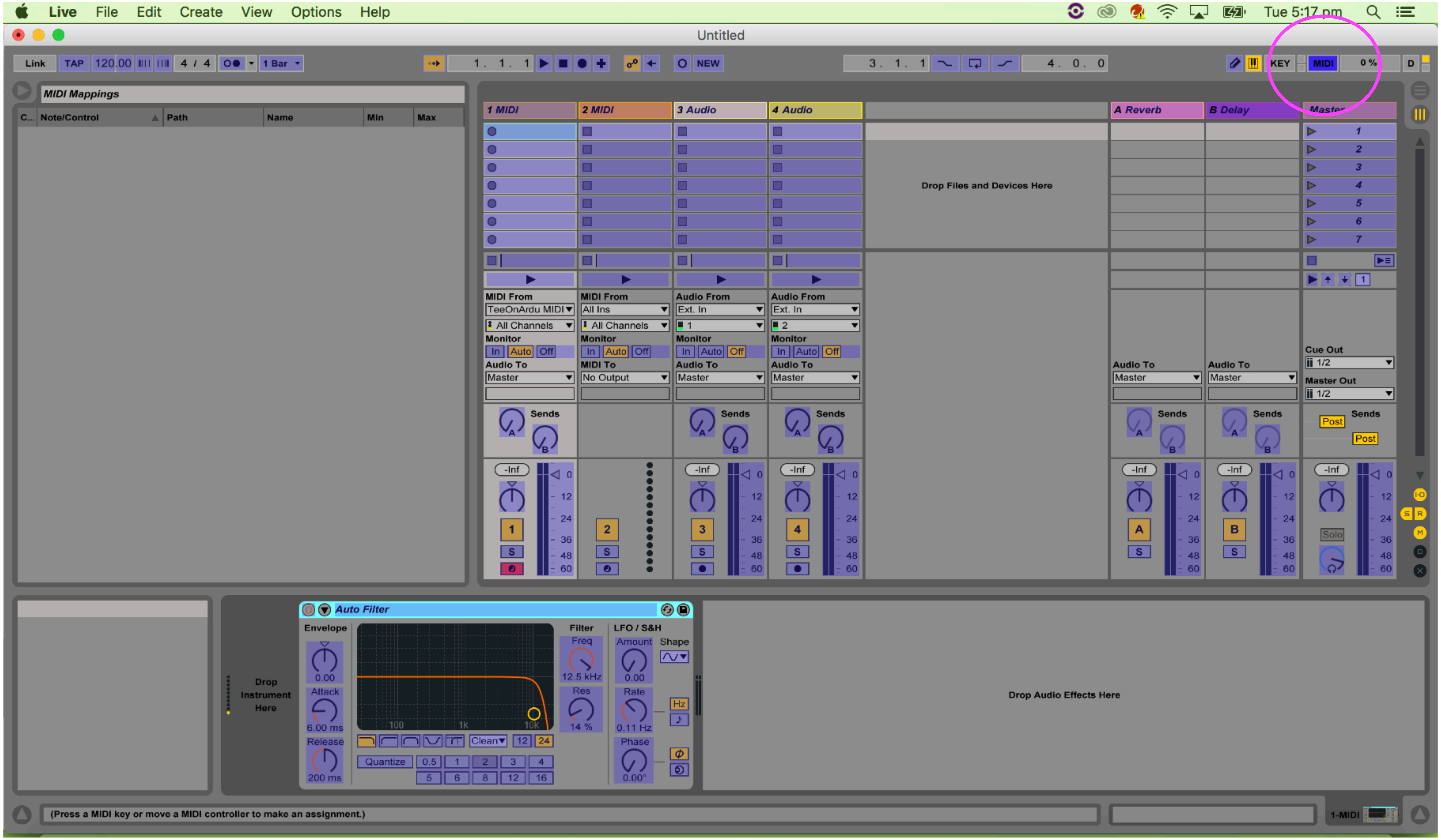
MIDI Ports	Track	Sync	Remote
▷ Input: TeeOnArdu MIDI	On	Off	Off
▷ Output: TeeOnArdu MIDI	On	Off	Off

Connecting to Digital Audio Workstation (DAW)



MIDI From: TeeOnArdu MIDI

Mapping Potentiometers



(Press a MIDI key or move a MIDI controller to make an assignment.)

Project Links

- Mini UNTZtrument Project: https://learn.adafruit.com/mini-untztrument-3d-printed-midi-controller/overview
- TeeOnArdu: https://github.com/adafruit/TeeOnArdu