MacroPot Assembly Guide

The most recent version of this document is located here:
https://docs.google.com/document/d/1nHQZJlmbKfMoiNE-2jQ2WAEOWn6yFhDtXKQVzbqB8Ts

MacroPot assembly process consists of three main steps. Electrical work - soldering all components, mechanical - drilling an enclosure and painting, and initial firmware initialization.
It may take upto an hour to finish the assembly (not counting a paint to dry) but depends on your level of experience. It's recommended that you solder a PCB first, and so you can test the unit before assembling into the enclosure.

There is also a video version of this guide: https://youtu.be/n0h99QKWodY

Assembling a PCB.

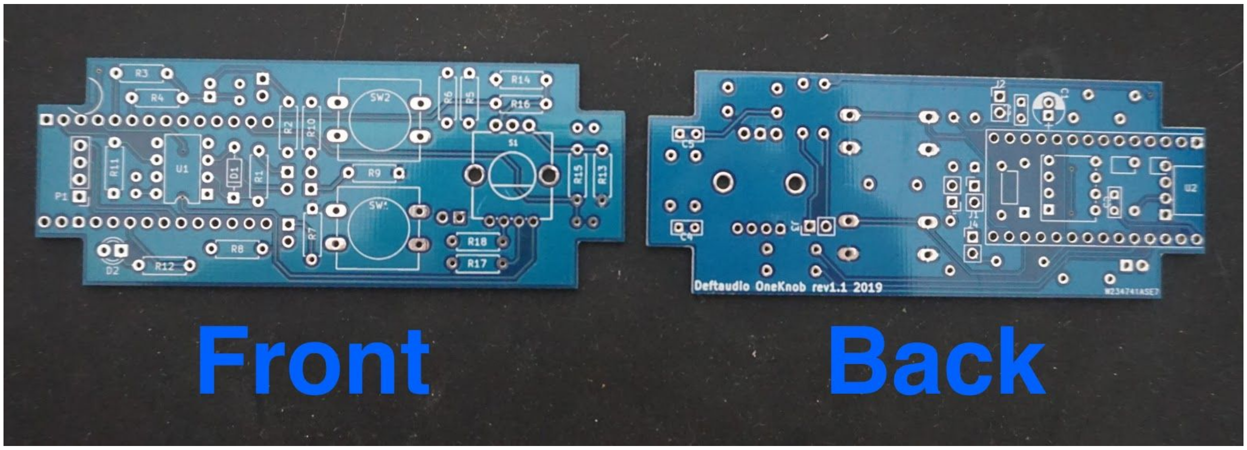
This is a step where you can save some time, if you don't use some of MacroPot functionality. For instance, you're not planning to use PWM audio output or you don't need all HW MIDI ports. You just don't populate those parts and leave it empty.

Here are the components you may consider optional:

PWM audio output	R2, R3, R4, C1, C2, J2
HW MIDI input	R1, R11, D1, U1, C3, J1
HW MIDI output 1	R5, R6, J3
HW MIDI output 2	R9, R10, J5
HW MIDI output 3	R7, R8, J4

Overall sequence doesn't matter much, however, here are some recommendations to simplify the process, so soldering can be more convenient:

1. Get familiar with the board design fist. Here are the front and back side of the board:

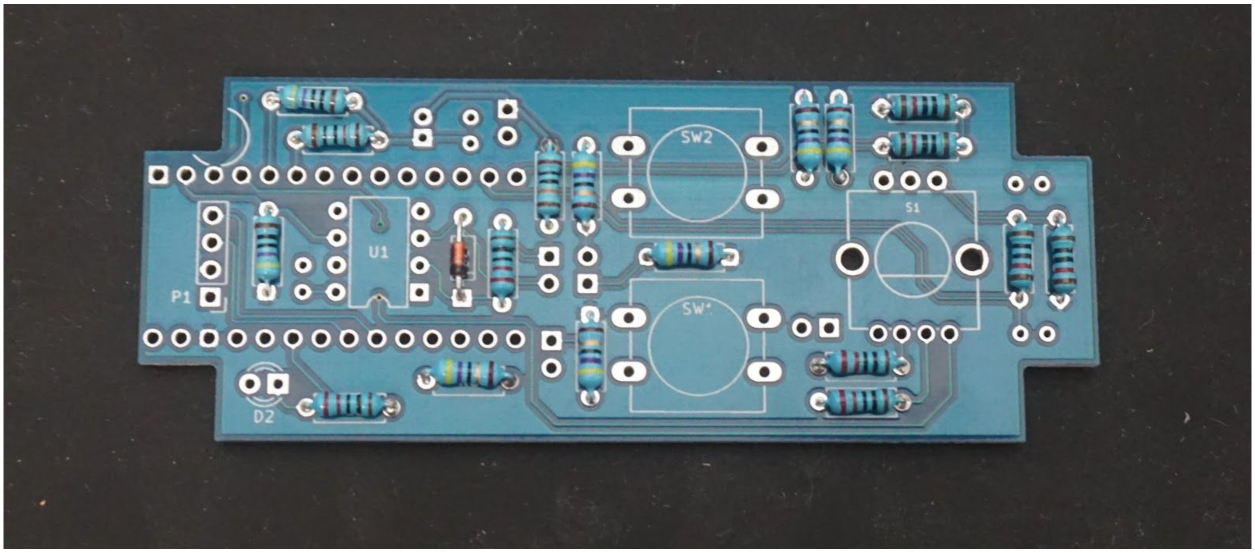


2. Decide on the functionality and components you need.
3. Start with soldering smaller parts first, such as resistors and diodes, so you lay the board on them and don't need to hold them in a position.
4. Get to bigger components later, leave headers to the end.

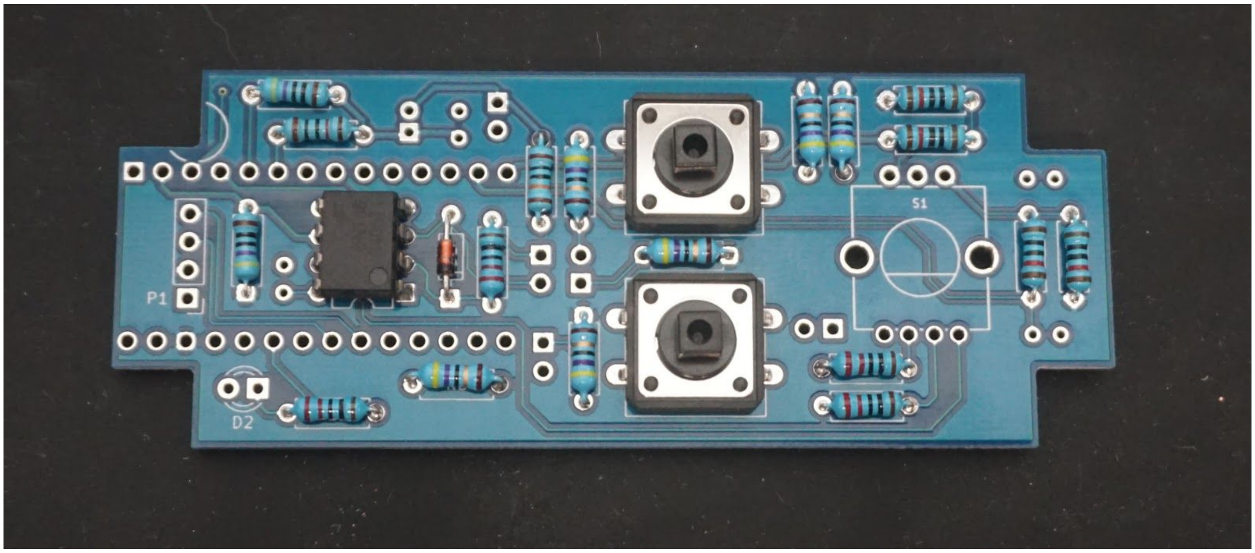
Soldering steps:

- Start with resistors on the front side and D1 diode. Those are:

D1	1N4148
R1, R12, R17, R18	220 ohm
R3(originally 475, 470 is Ok), R11	470 ohm
R13, R14, R15, R16	10k
R4(originally 127, 120 is Ok)	120 ohm
R5, R6, R7, R8, R9, R10	47 ohm
R2	100k

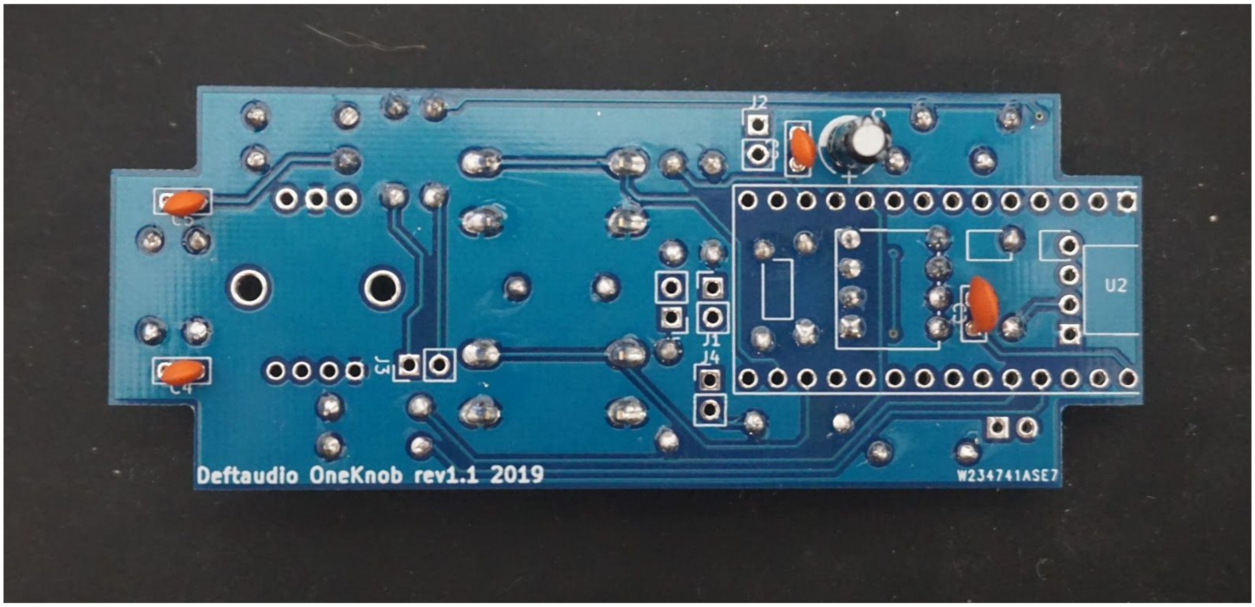


- Add U1 (6N138) without socket (important) and two push buttons

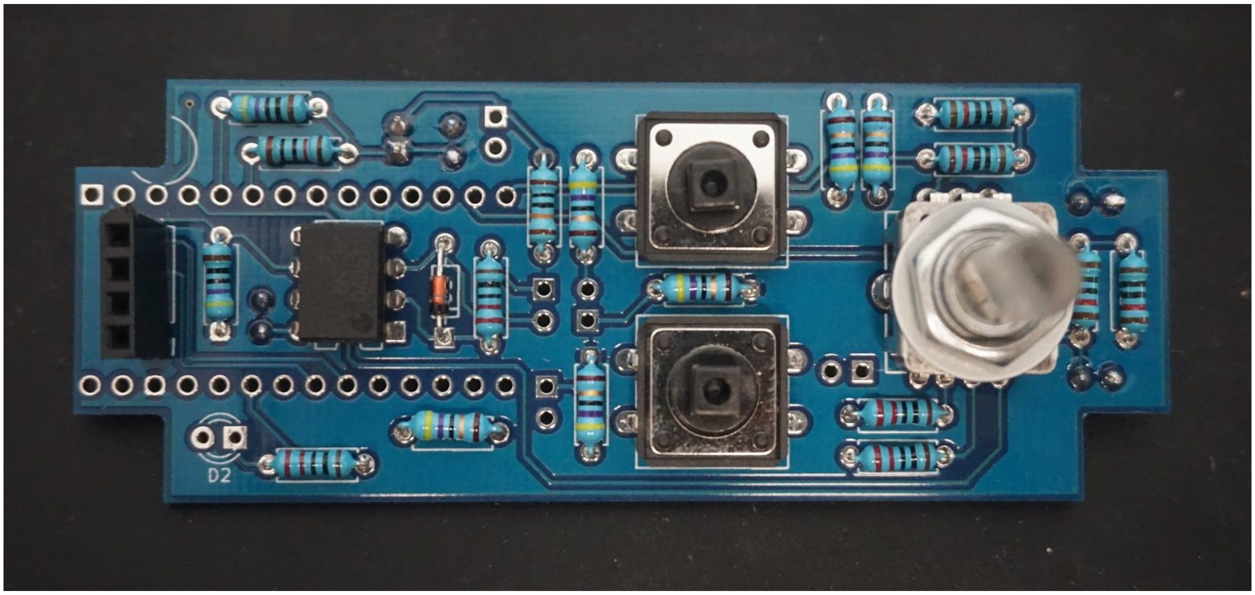


- Flip the board over and populate capacitors on the back side. Those are:

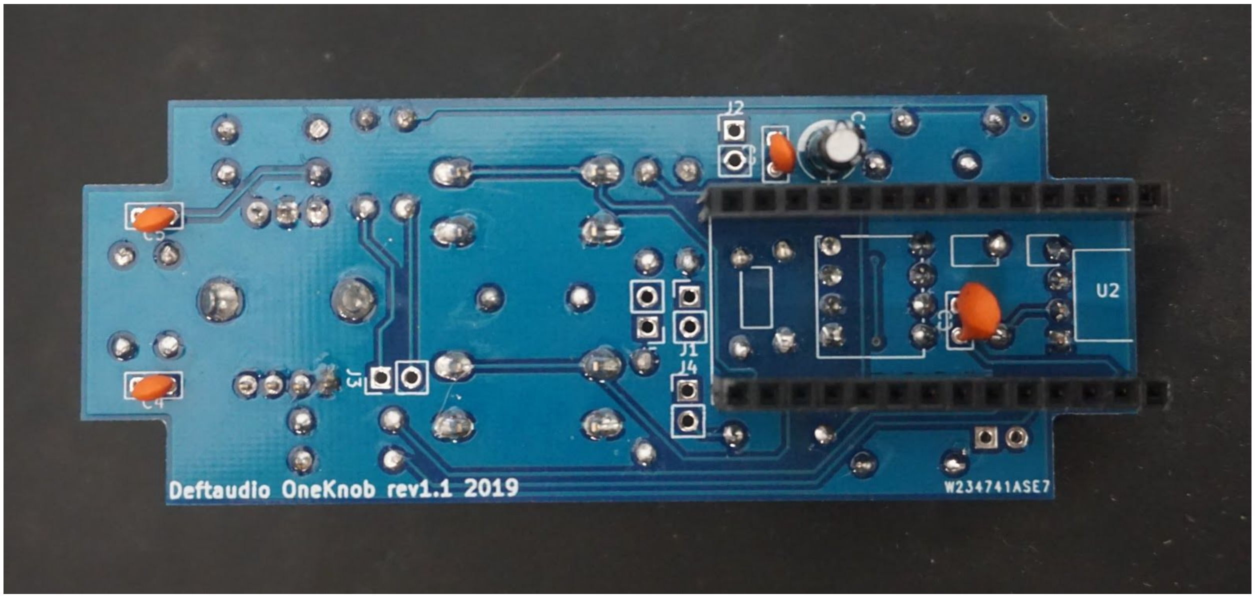
C2, C4, C5	103 (0.01uf)
C3	104 (0.1uf)
C1	10uf



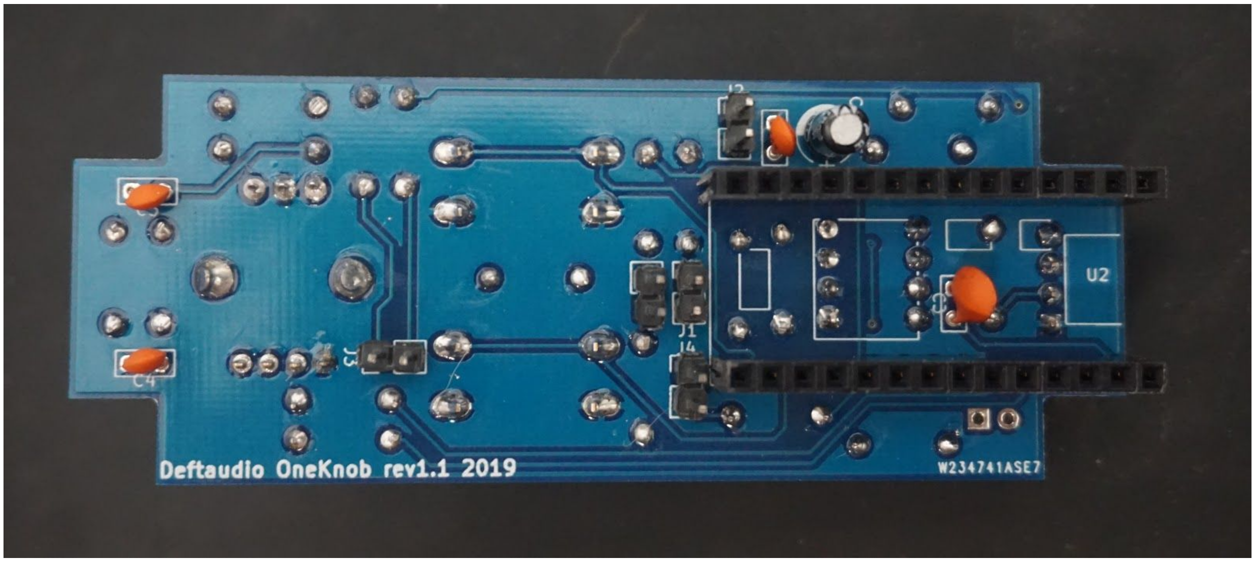
- Cut a 4-pin header female from 40 pin strip and solder it for OLED connection, solder rotary encoder:



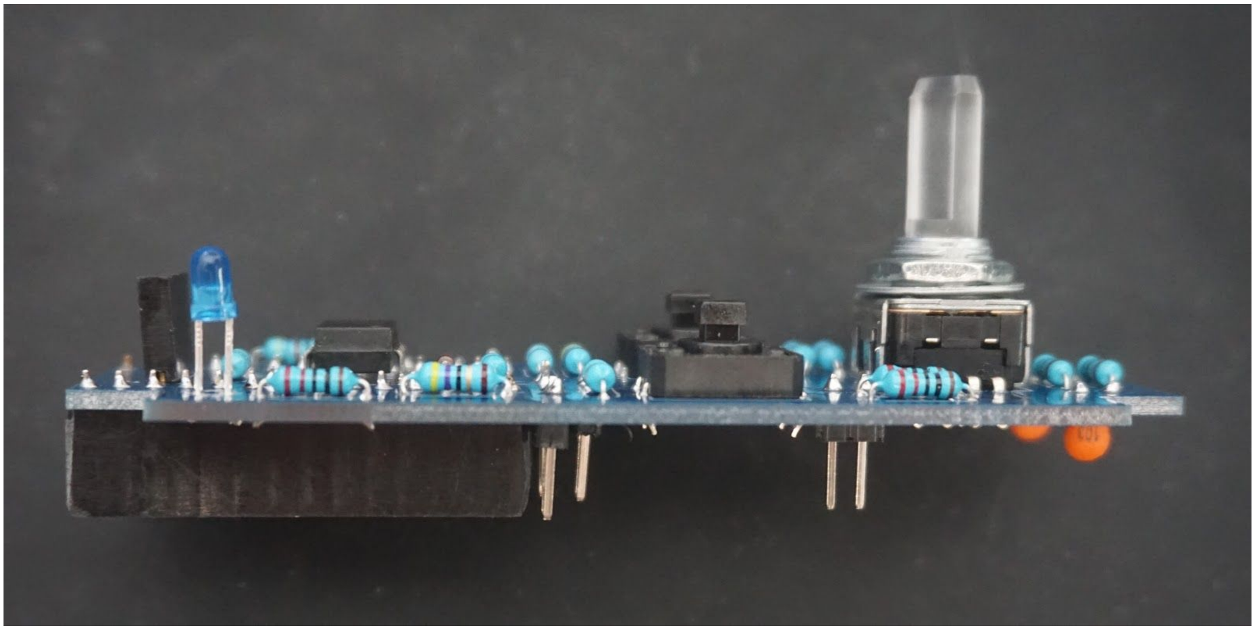
- Cut two 14-pin female strips and solder them on the backside for Teensy 3.2:



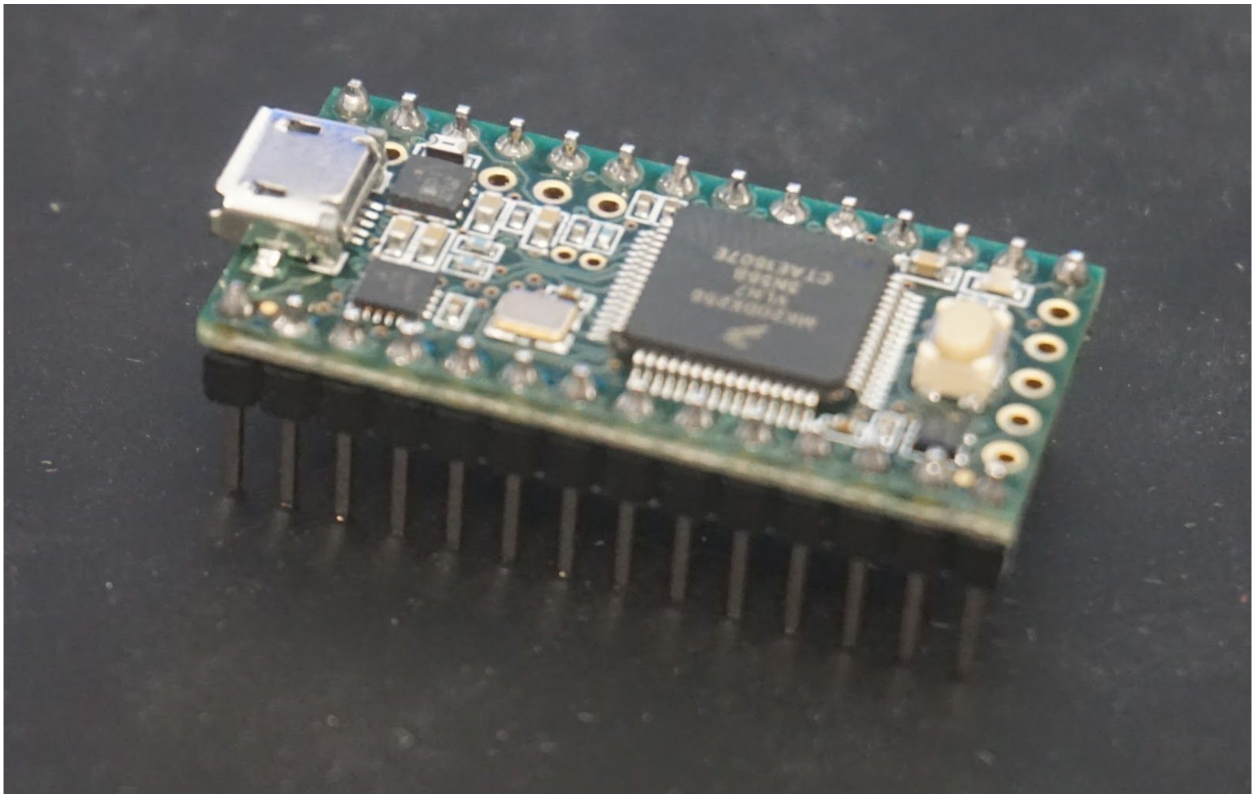
- Cut five 2-pin male headers, solder them into J1, J2, J3, J4, J5 locations:



- Solder 3mm blue LED at about 5mm distance from the board. Follow pin orientation, bigger pin should be on the left, see here:



- Solder male pins to Teensy 3.2:



Firmware initialization.

Right after you completed board soldering, it’s a good time to start MacroPot for the first time and make sure that at least basic functions are working fine. There are two stages in this process - updating firmware and loading default settings to Teensy’s EPROM.

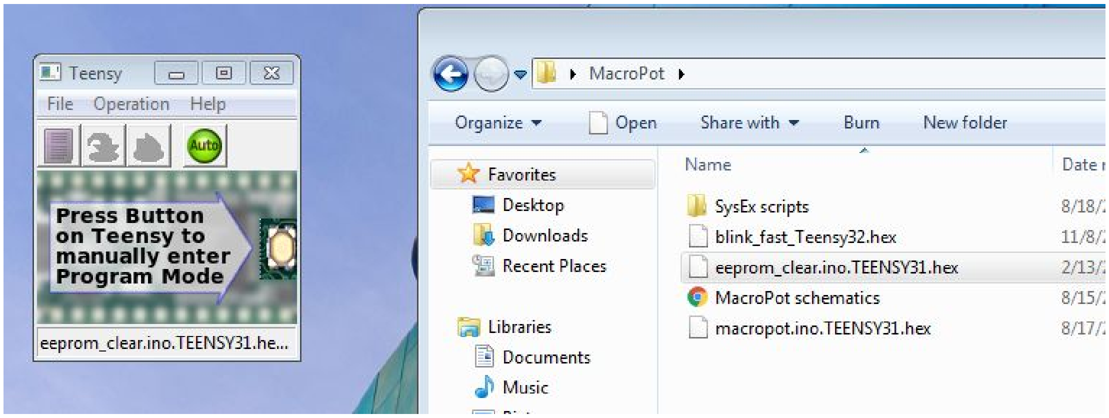
Updating a firmware from compiled *.hex file:

- If you’re experienced Arduino user, you may skip this part, it’s mainly for new users who are not familiar with Arduino programming or software development.
- *.hex file is a compiled binary, that you simply load to a Teensy board. That means, you don’t do any changes to the code by yourself, you just use what’s already available.
- While programming, Teensy can be assembled onto MacroPot board or just connected over USB without anything attached. This doesn’t matter. Obviously on the next power on it has to be attached to the board.

It’s important to understand that MacroPot saves macros and its state into nonvolatile memory (EPROM) of the Teensy controller. Main firmware image is only a program that runs on the controller. It doesn’t include any configuration parameters or pre-defined macroses. This is what needs to be configured after the firmware update. However, that’s also recommended to clean EPROM before updating main firmware, so, previous or garbage EPROM data won’t be used as macro configuration.

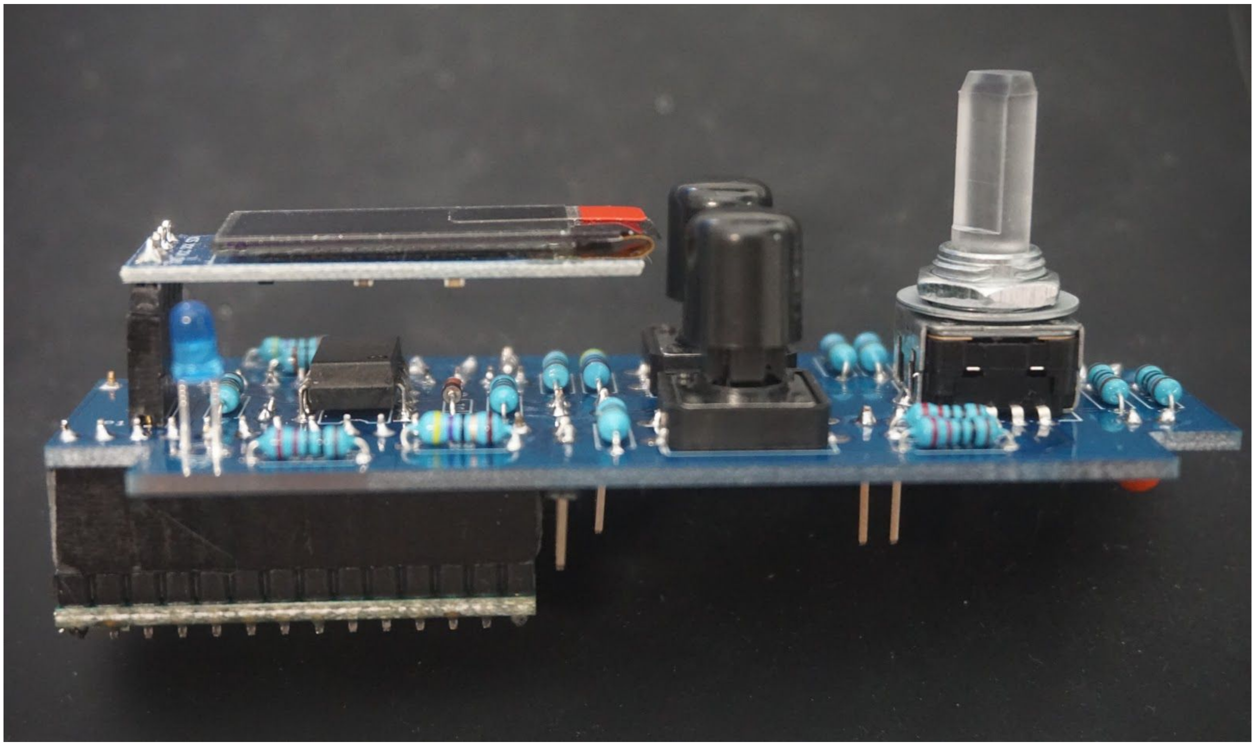
Steps:

- To start, download Teensy loader from official site: https://www.pjrc.com/teensy/loader.html
- Run it and drop eeprom_clear.hex firmware first from the MacroPot hex from the repository: https://github.com/Deftaudio/MacroPot/blob/master/Firmware/eeprom_clear.hex
- Press button on Teensy and make sure loader app updated it.

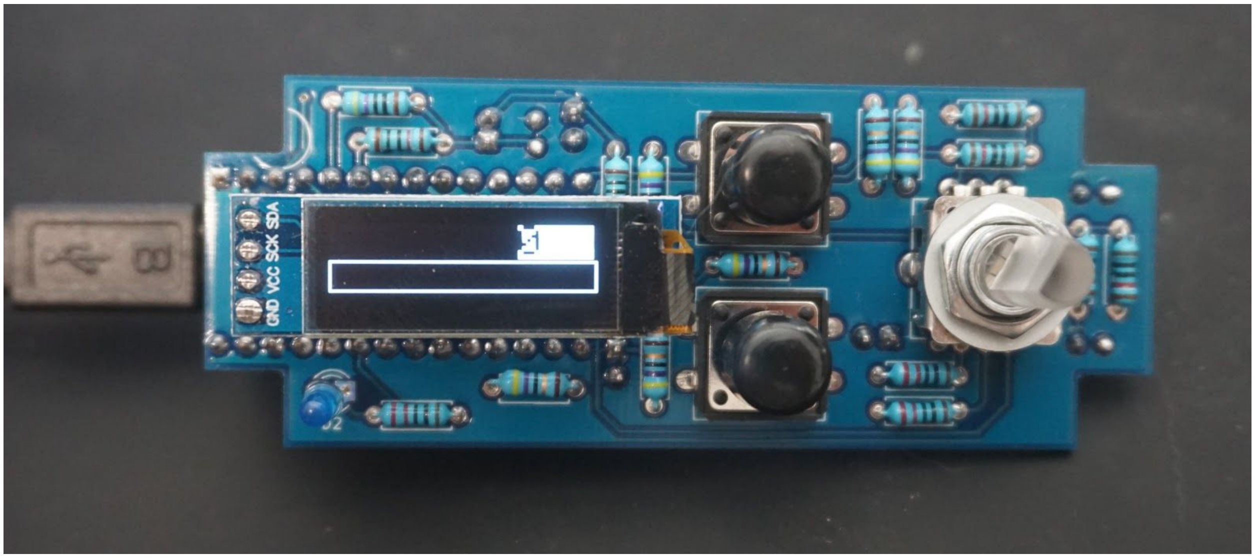

Press Button on Teensy to manually enter Program Mode

eeprom_clear.ino.TEENSY31.he...

- Repeat steps for main firmware: https://github.com/Deftaudio/MacroPot/blob/master/Firmware/macropot_1.0.1.hex
- Make sure the board is complete, i.e. attach Teensy, OLED screen(temporary) and push button caps



- Once connected MacroPot will start with empty configuration, i.e. no macro name and a progress bar



Alternative: Updating a firmware from Arduino project:

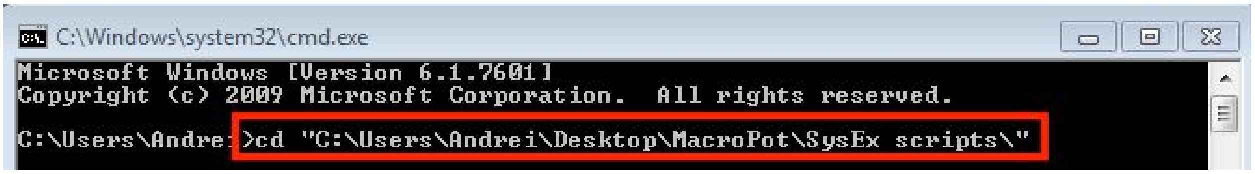
- This step assumes that you take MacroPot source files, compile them and load to Teensy. It’s for experienced users how are willing to enhance MacroPot functions in future by contributing into the software development.
- Make sure you’re running latest Arduino version.
- Download and install Teensyduino: https://www.pjrc.com/teensy/td_download.html
- Open MacroPot project file.
- Set a controller to Teensy 3.2 in MIDIx4 mode.
- Set a running speed to 72Mhz, you don’t need to overclock Teensy for basic firmware.
- Compile and load eeprom_clear tutorial sketch following by the main firmware.

Loading default macro configuration

MacroPot is deeply programmable controller over SysEx messages. However, to get you started, there are default scripts created to program its state. This minimizes your learning time, as default macroses designed to cover many common scenarios. Follow the download repository https://github.com/Deftaudio/MacroPot/tree/master/SysEx%20scripts you’ll find MS Windows and Mac OS X versions. They are designed to use cross-platform command line SendMIDI utility by Geert Bevin: https://github.com/gbevin/SendMIDI Just put a recent binary to the script directory first.

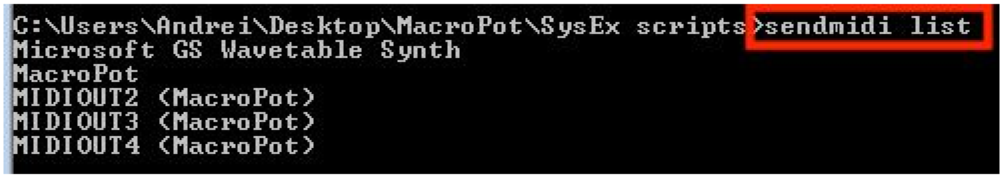
Here is an example of running SendMIDI under Windows,

- Run Command Line “cmd” and change a directory to the location of scripts:



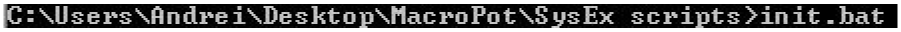
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Andrei>cd "C:\Users\Andrei\Desktop\MacroPot\SysEx scripts\"
```

- Connect MacroPot and execute “sendmidi list”. You’ll see all MIDI output devices connected to the system. As MacroPot has four virtual USB ports, there four separate devices below. You can use any of them for programming, but it’s easier to pick the first one due to a naming convention. Scripts do that by default.



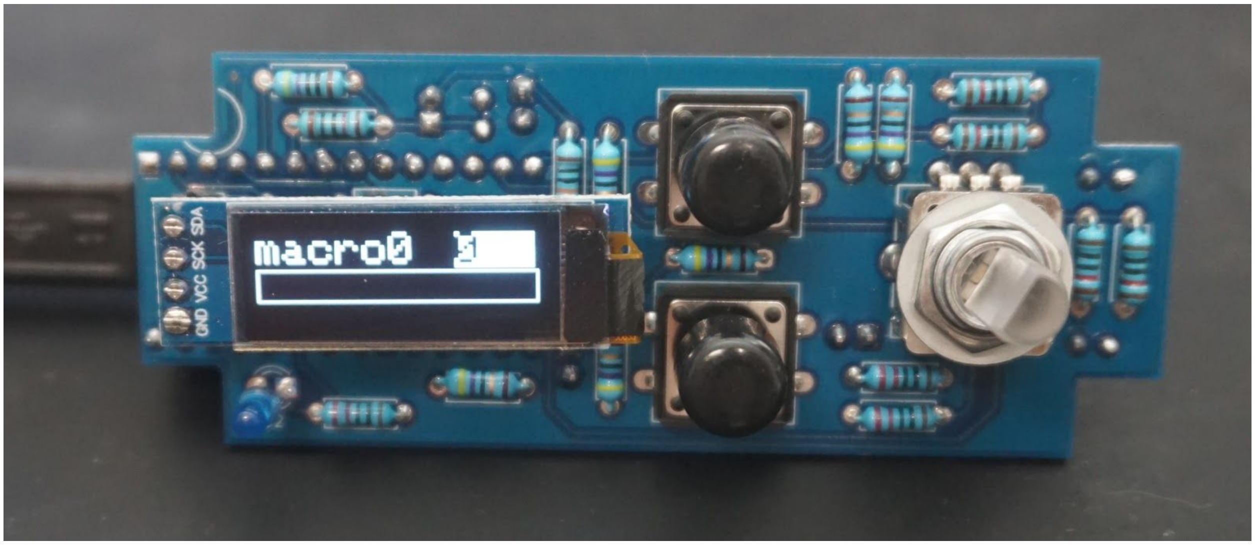
```
C:\Users\Andrei\Desktop\MacroPot\SysEx scripts>sendmidi list
Microsoft GS Wavetable Synth
MacroPot
MIDIOUT2 (MacroPot)
MIDIOUT3 (MacroPot)
MIDIOUT4 (MacroPot)
```

- Press simultaneously Up and Down buttons on MacroPot controller, until “Sysex Mod” appears on the screen.
- Execute init.bat (Windows) or init.sh (OS X, don’t forget to make it executable first)

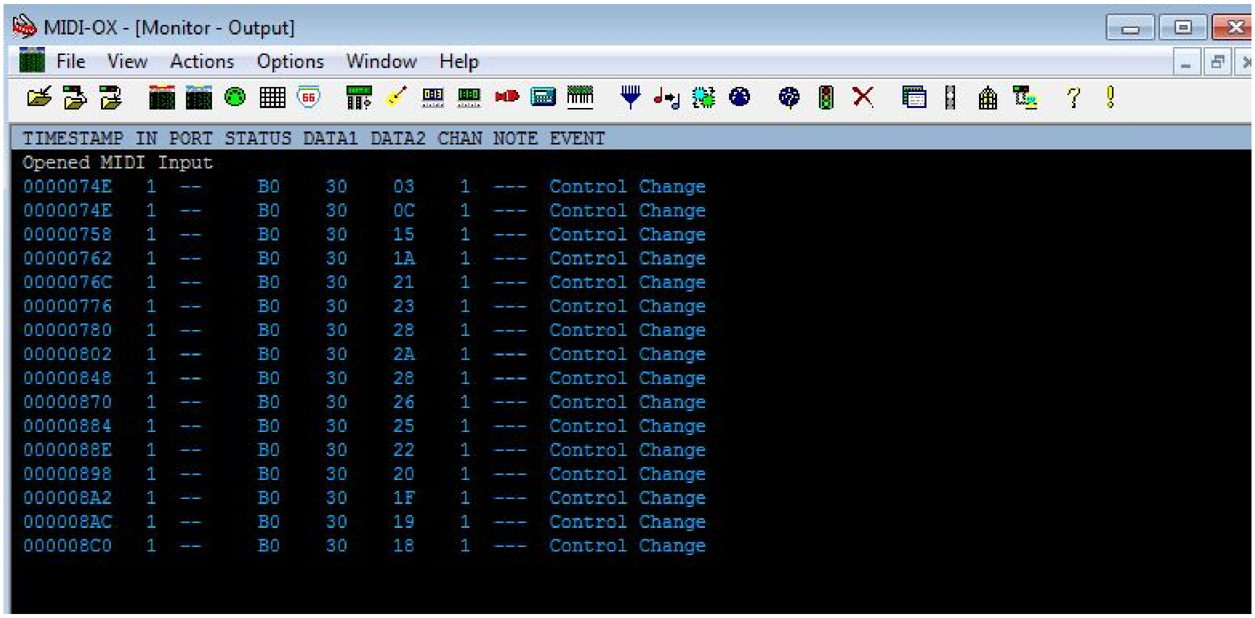


```
C:\Users\Andrei\Desktop\MacroPot\SysEx scripts>init.bat
```

- Once script finishes, press Up button, now you should be able to see a default macro name:



- At this point, you may run some basic tests, even without hardware MIDI ports connected. You may use MIDI OX or similar MIDI monitor sniffing USB MIDI traffic.

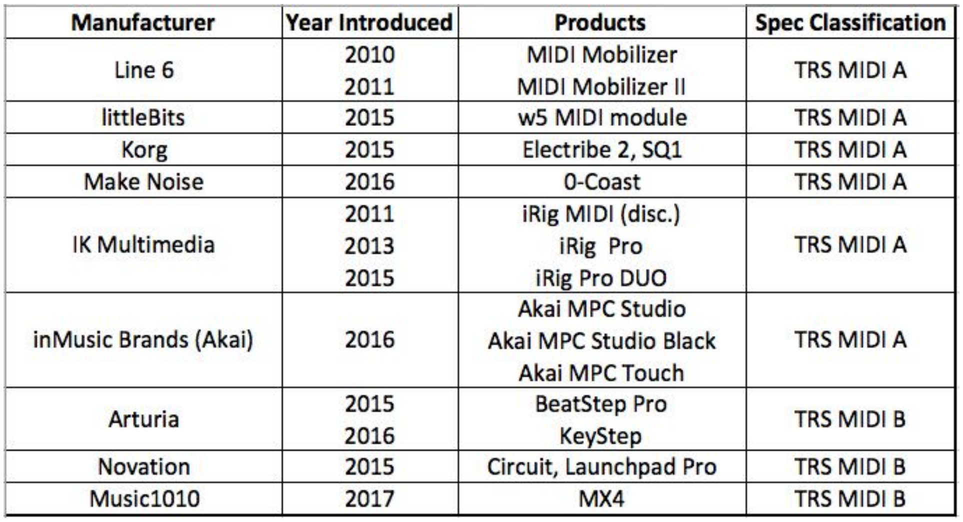

MIDI-OX - [Monitor - Output]
File View Actions Options Window Help

```
TIMESTAMP IN PORT STATUS DATA1 DATA2 CHAN NOTE EVENT
Opened MIDI Input
0000074E   1  --     B0    30    03    1  ---  Control Change
0000074E   1  --     B0    30    0C    1  ---  Control Change
00000758   1  --     B0    30    15    1  ---  Control Change
00000762   1  --     B0    30    1A    1  ---  Control Change
0000076C   1  --     B0    30    21    1  ---  Control Change
00000776   1  --     B0    30    23    1  ---  Control Change
00000780   1  --     B0    30    28    1  ---  Control Change
00000802   1  --     B0    30    2A    1  ---  Control Change
00000848   1  --     B0    30    28    1  ---  Control Change
00000870   1  --     B0    30    26    1  ---  Control Change
00000884   1  --     B0    30    25    1  ---  Control Change
0000088E   1  --     B0    30    22    1  ---  Control Change
00000898   1  --     B0    30    20    1  ---  Control Change
000008A2   1  --     B0    30    1F    1  ---  Control Change
000008AC   1  --     B0    30    19    1  ---  Control Change
000008C0   1  --     B0    30    18    1  ---  Control Change
```

A note about TRS-DIN MIDI breakout cable.

At the time vendors decided to utilize TRS connectors for MIDI a common standard didn’t exist. So, every manufacturer went their own way, which basically introduced two standards, called Type A and Type B. Please,refer to MIDI.org article about that: https://www.midi.org/articles-old/updated-how-to-make-your-own-3-5mm-mini-stereo-trs-to-midi-5-pin-din-cables

Type A assumes TRS tip is connected to Pin 5, and ring to Pin 4. While Type B specs has Pin 4 on a tip and ring on Pin 5. In both cases sleeve is the ground.

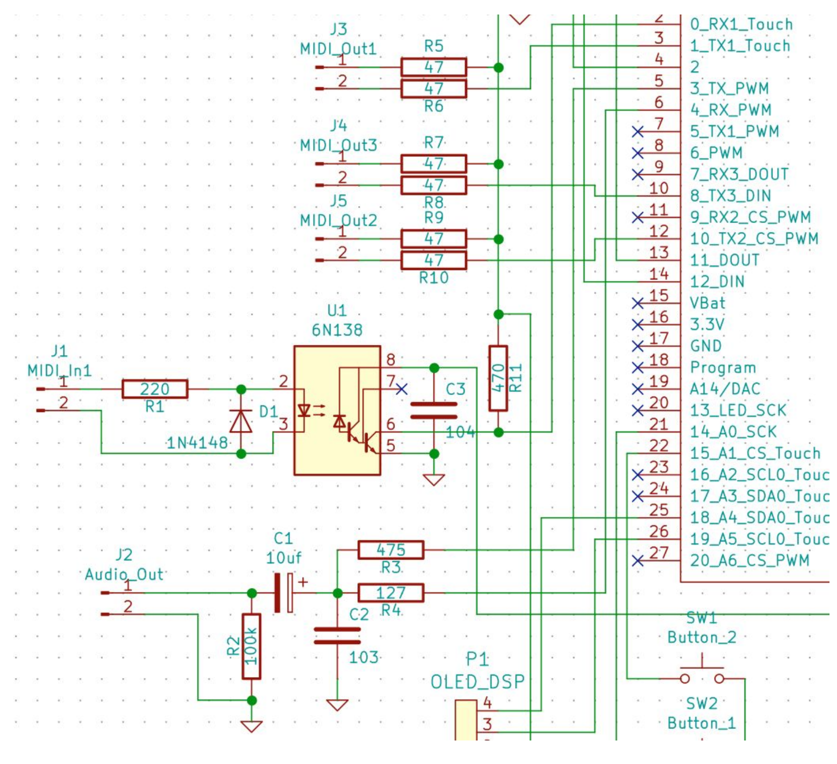


Manufacturer	Year Introduced	Products	Spec Classification
Line 6	2010 2011	MIDI Mobilizer MIDI Mobilizer II	TRS MIDI A
littleBits	2015	w5 MIDI module	TRS MIDI A
Korg	2015	Electribe 2, SQ1	TRS MIDI A
Make Noise	2016	0-Coast	TRS MIDI A
IK Multimedia	2011 2013 2015	iRig MIDI (disc.) iRig Pro iRig Pro DUO	TRS MIDI A
inMusic Brands (Akai)	2016	Akai MPC Studio Akai MPC Studio Black Akai MPC Touch	TRS MIDI A
Arturia	2015 2016	BeatStep Pro KeyStep	TRS MIDI B
Novation	2015	Circuit, Launchpad Pro	TRS MIDI B
Music1010	2017	MX4	TRS MIDI B

By the design MacroPot supports both types, all you need to do is to flip pin on a board header and so that changed tip/ring assignment. Ground pin is ignored, can be simply implemented by connecting it to the enclosure. Good news, that MIDI designed in a way, that even if you used the wrong polarity, this won’t burn the scheme. So, if port doesn’t work from the first attempt, flip the pin orientation and try again.

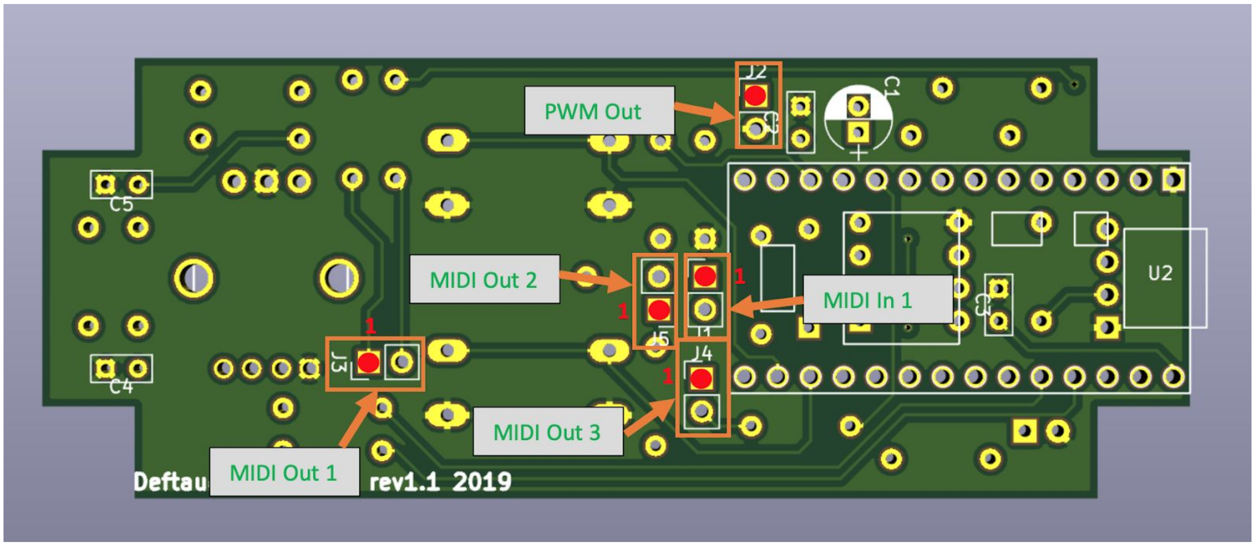
Connecting External Ports - MIDI and Audio.

Below is a connector assignment per header:

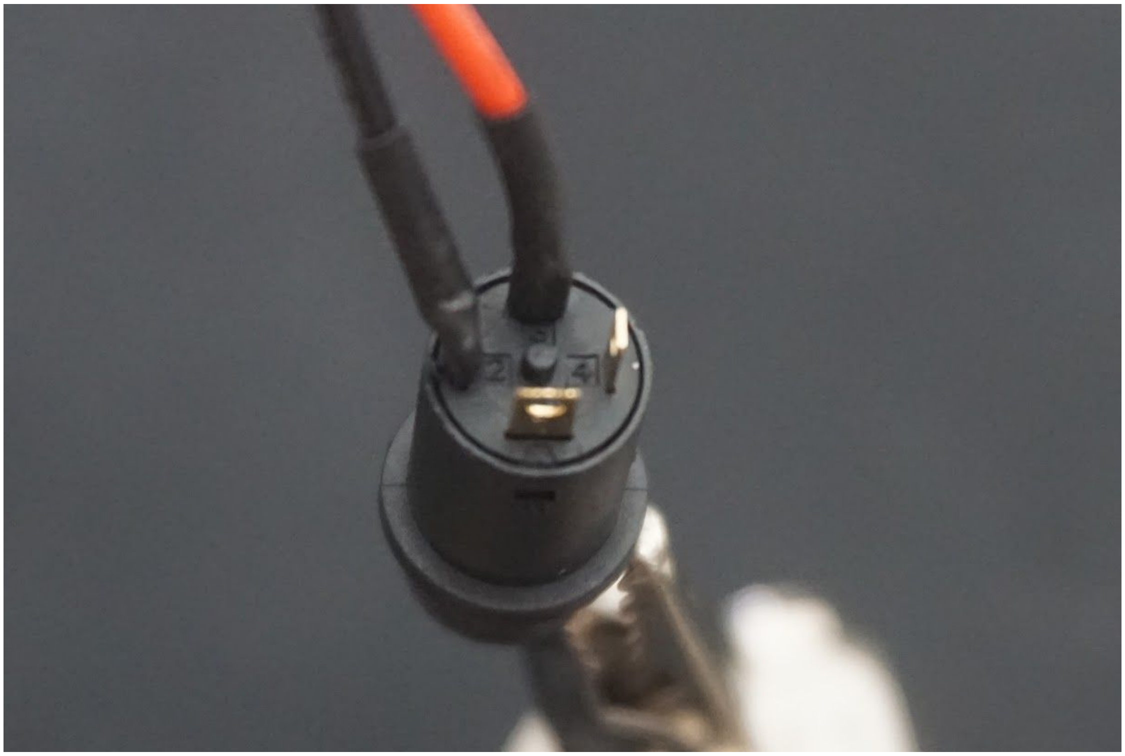
MIDI In	J1	
MIDI Out 1	J3	
MIDI Out 2	J5	
MIDI Out 3	J4	
PWM Out	J2	

If you're following **Type A** TRS-MIDI specification, then Pin 1 of the MIDI In/Out headers go to a tip, while Pin 2 goes to a ring.

- Refer to the Pin 1 orientation and header location below:



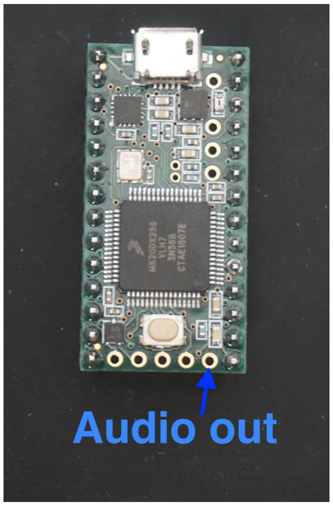
- MIDI TRS socket orientation is shown below. On this connector type a ring is labeled #2 and tip is #3. Solder wires of different colors to differentiate them:



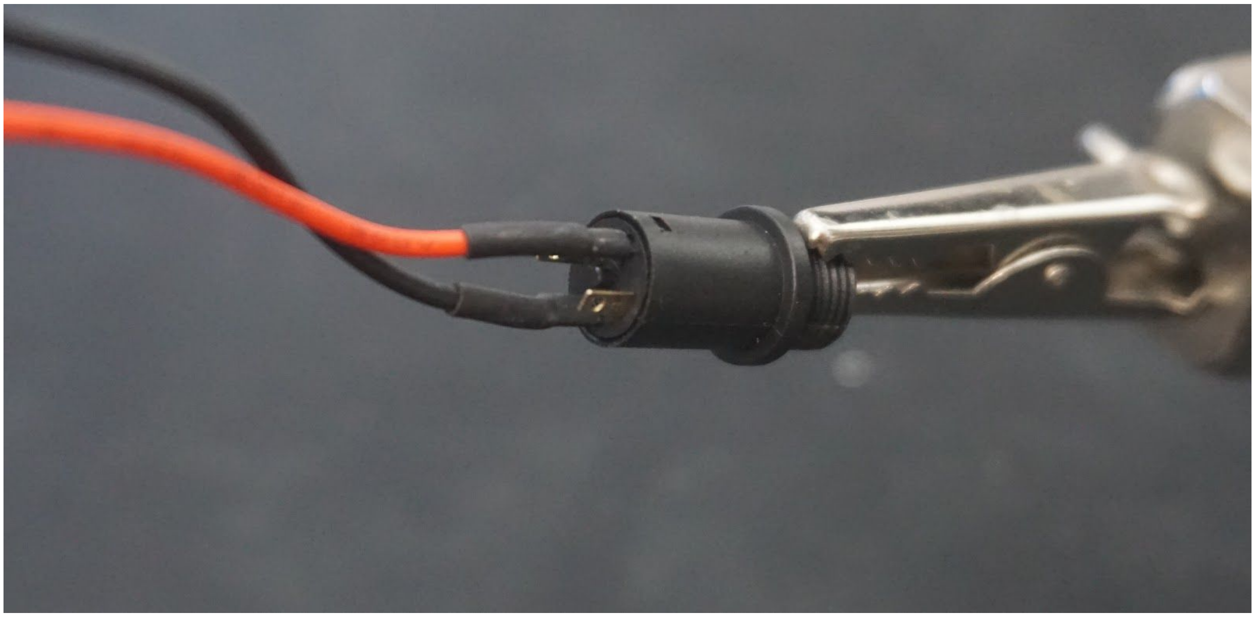
- So with that red wire means tip and black wire is for ring. Connect those accordingly to Type A or Type B specification to MacroPot headers.

Audio can be connected from two different places - PWM header output or DAC output. Those are differently programmed inside Teensy framework, you may prefer to use DAC output for better quality(12 bit) or even both of them together, if tip and ring pins utilized.

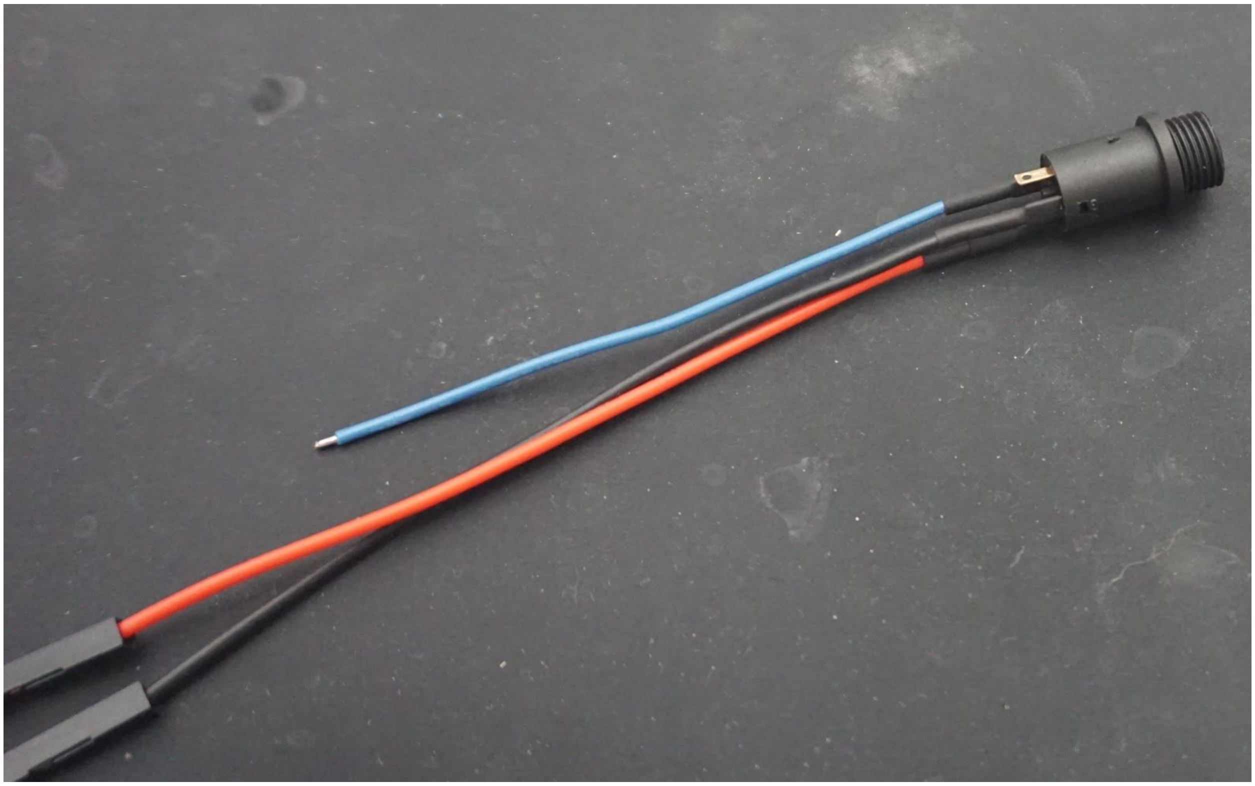
- In order to use DAC output you need to solder a wire directly to the pin shown below, as it’s not routed on the board to any headers.



- Here is an example of wiring tip and sleeve on TRS connector. Tip is market “3” and sleeve is “B”.



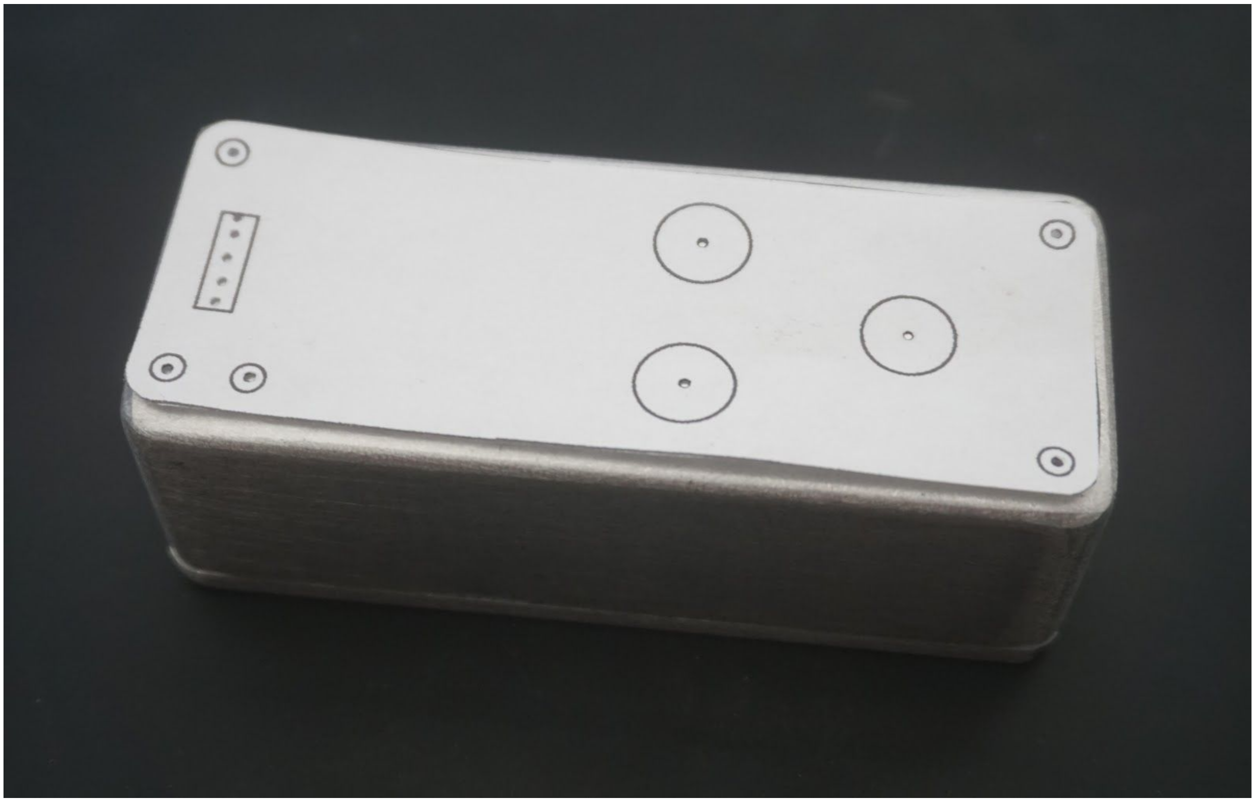
- You also may decide to connect both PWM and DAC audio outputs at the same time if you utilize TRS pinout. As such DAC output goes to a tip, PWM to a ring and ground on the sleeve:



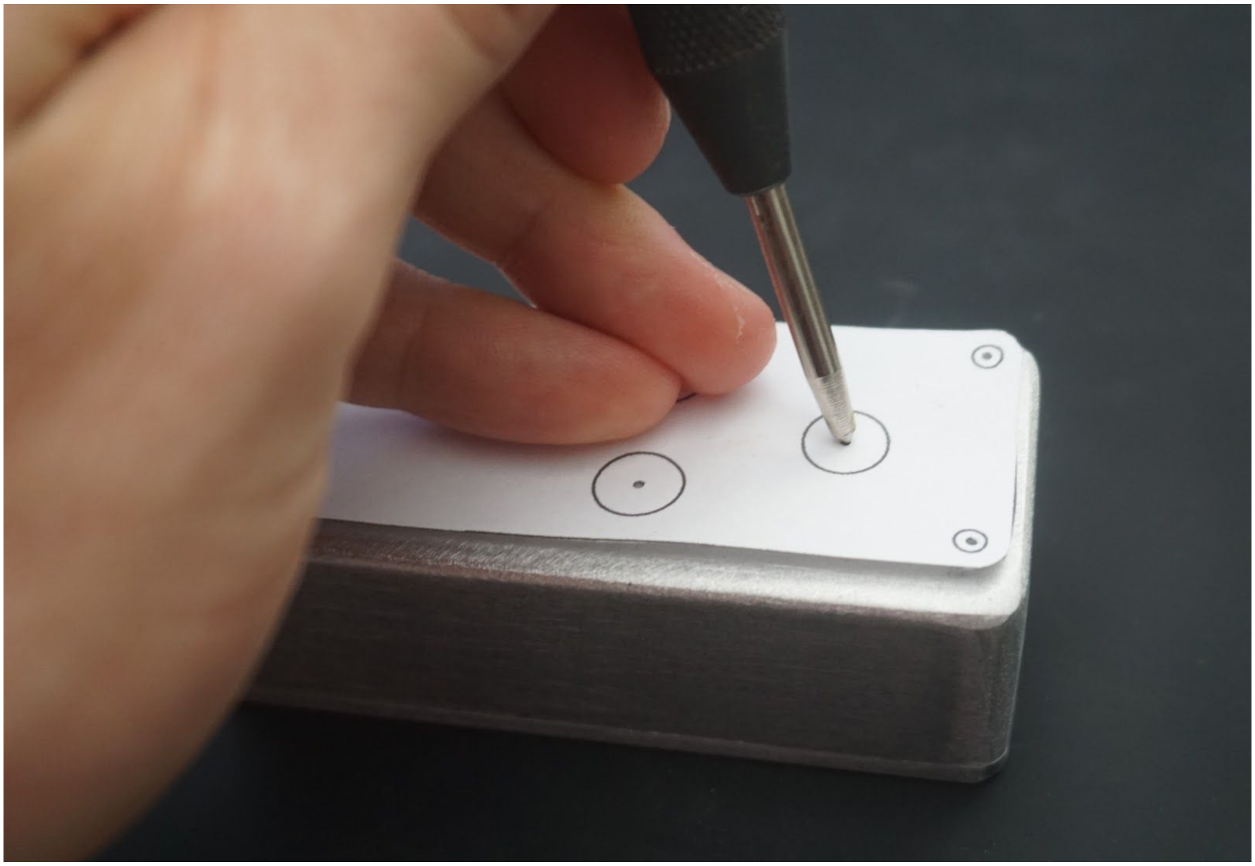
Preparing an enclosure.

Preparing an enclosure requires drilling holes, making four threads and painting. You may decide to use a drill press or a hand drill. Core tool is highly recommended to make hole with a precision.

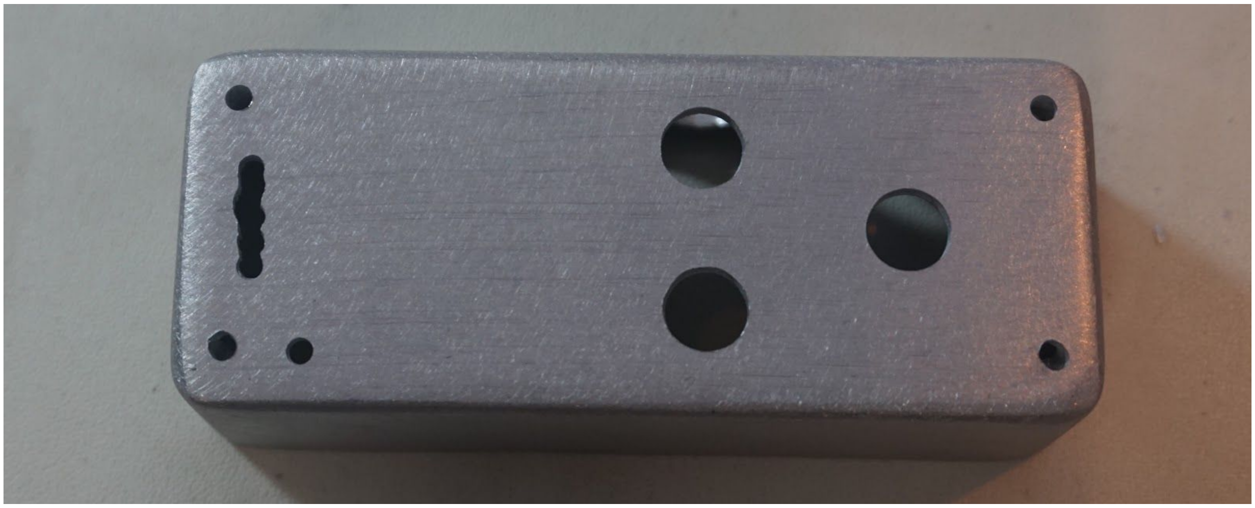
- Start by printing the draft label located here https://github.com/Deftaudio/MacroPot/blob/master/Acrylic%20panels/MacroPot-drill_allignment.pdf cutting it to the size and aligning to the enclosure:



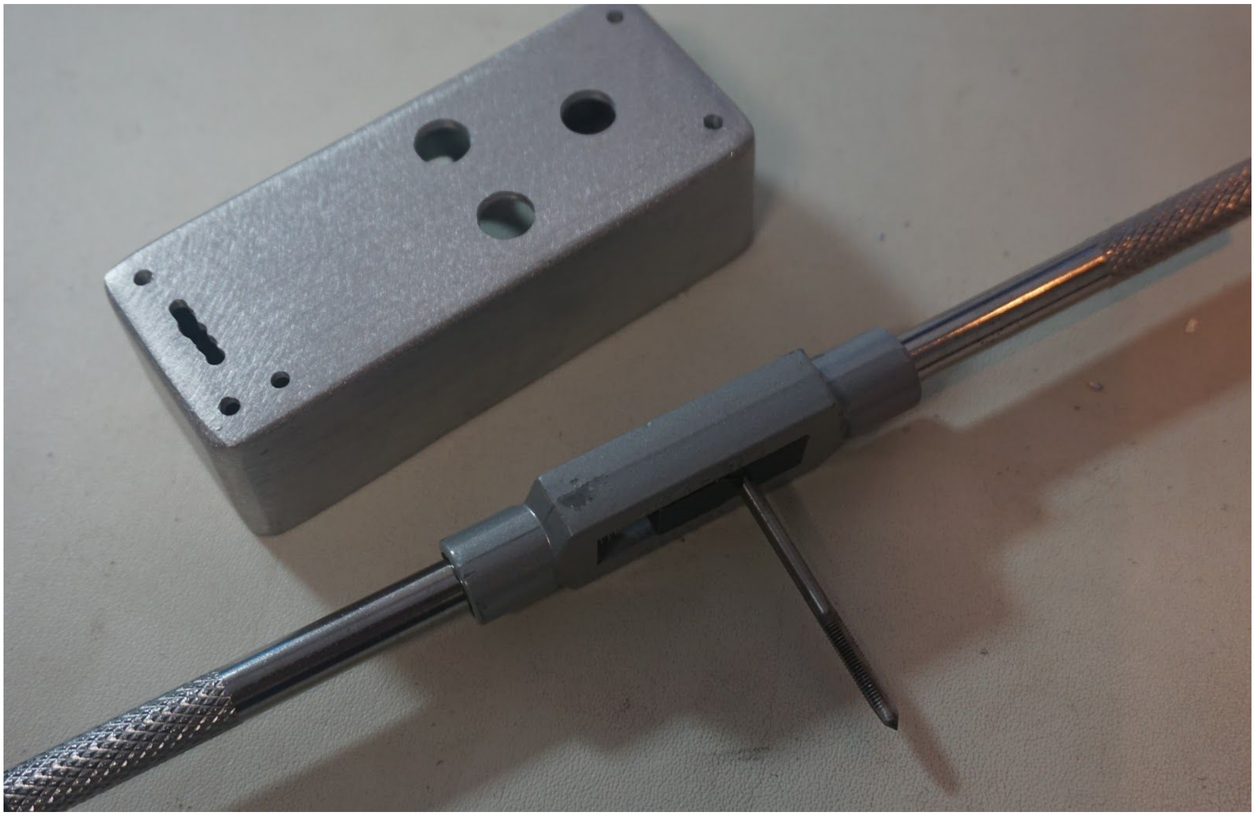
- Use a core tool to clearly mark the center of all holes. Note, for the OLED connector you'll drill multiple holes and then use a rasp for a final shape:



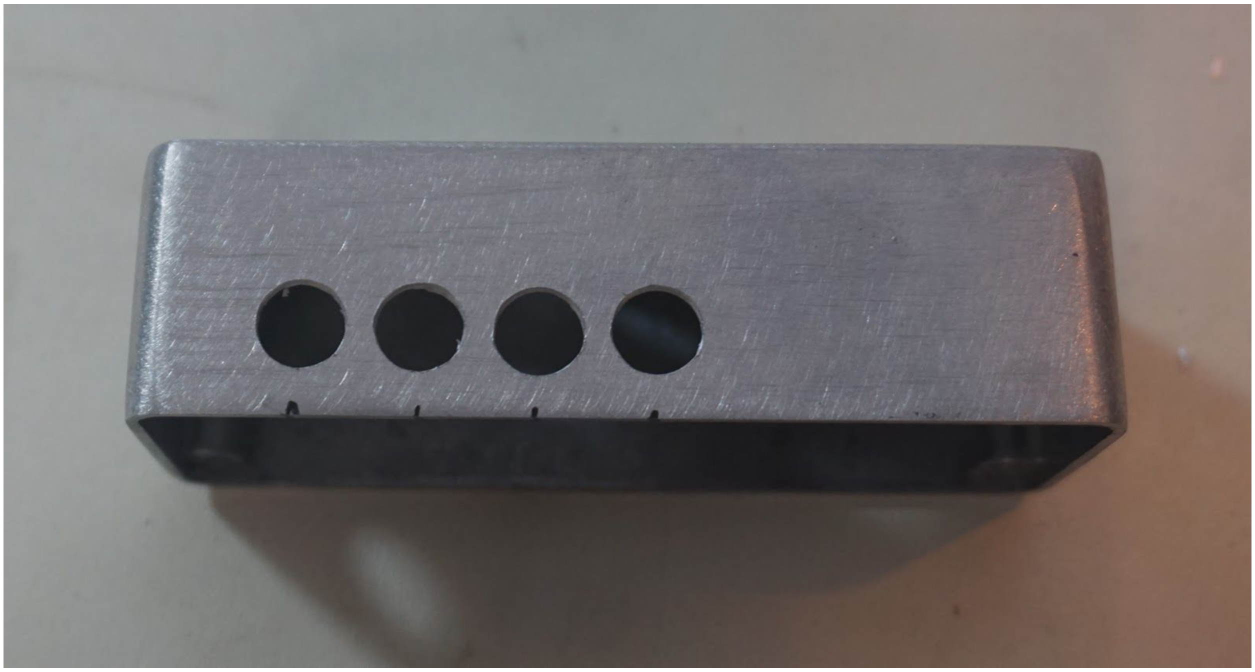
- Start drilling by using 2.5mm drill bit. That's important for corner holes that then will be 3mm threaded. Then adjust the size of the bit or better use step drill bit. Make sure button caps don't get stuck, you can make these holes significantly bigger to prevent that:



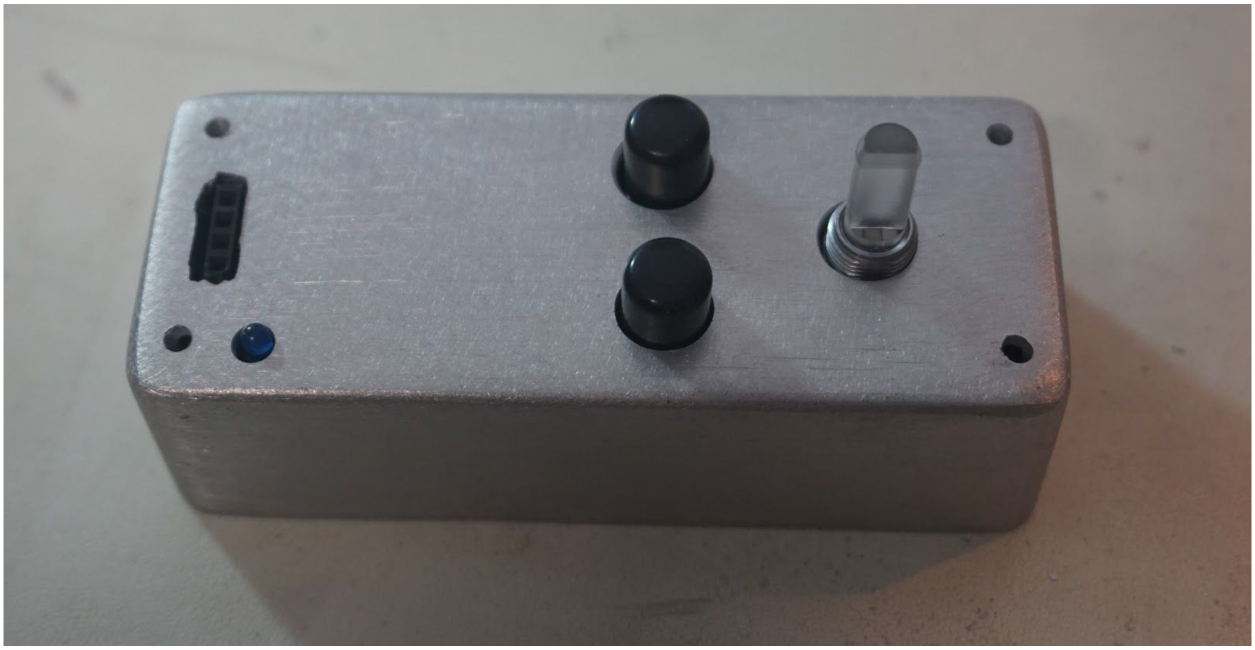
- Use a thread cutter tool for corner holes and cut 3mm thread. You can buy a thread bit on eBay, Amazon ~$4.



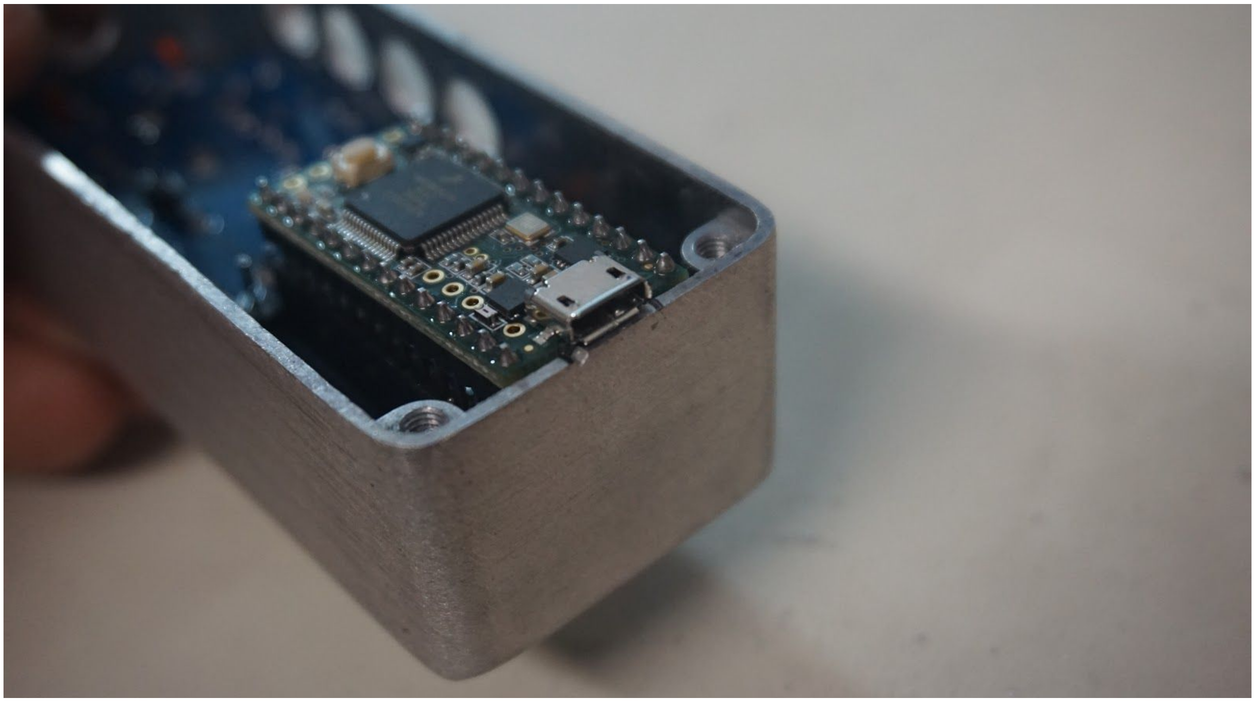
- Cut rear holes for 3.5mm TRS sockets. A distance between each is 11mm. This gets them close, but still able to fit all of them within a limited space and put a screw nut:



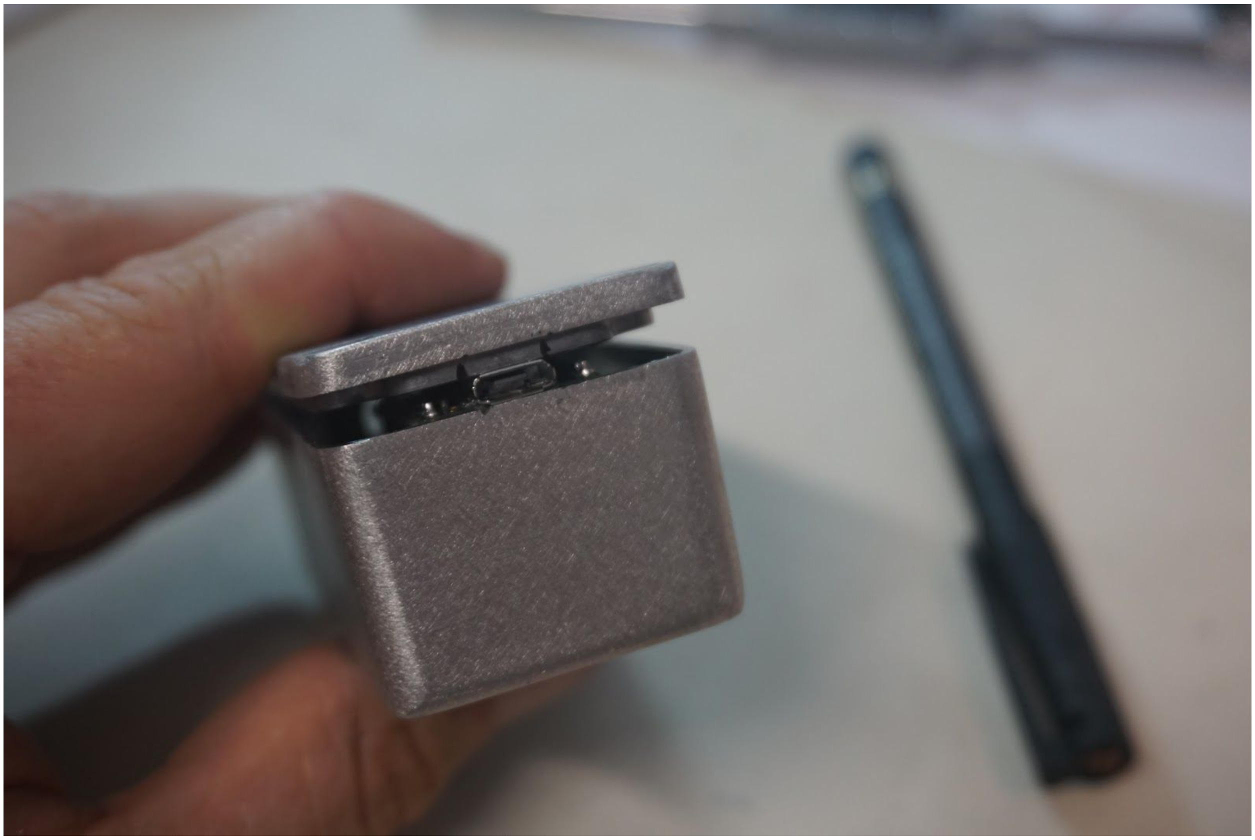
- Place the board inside (no Teensy and OLED attached) and put a washer and screw nut on the encoder:



- Place a Teensy board carefully, don't push it as USB connector stands outside and mark it's dimensions with a sharpie pen:



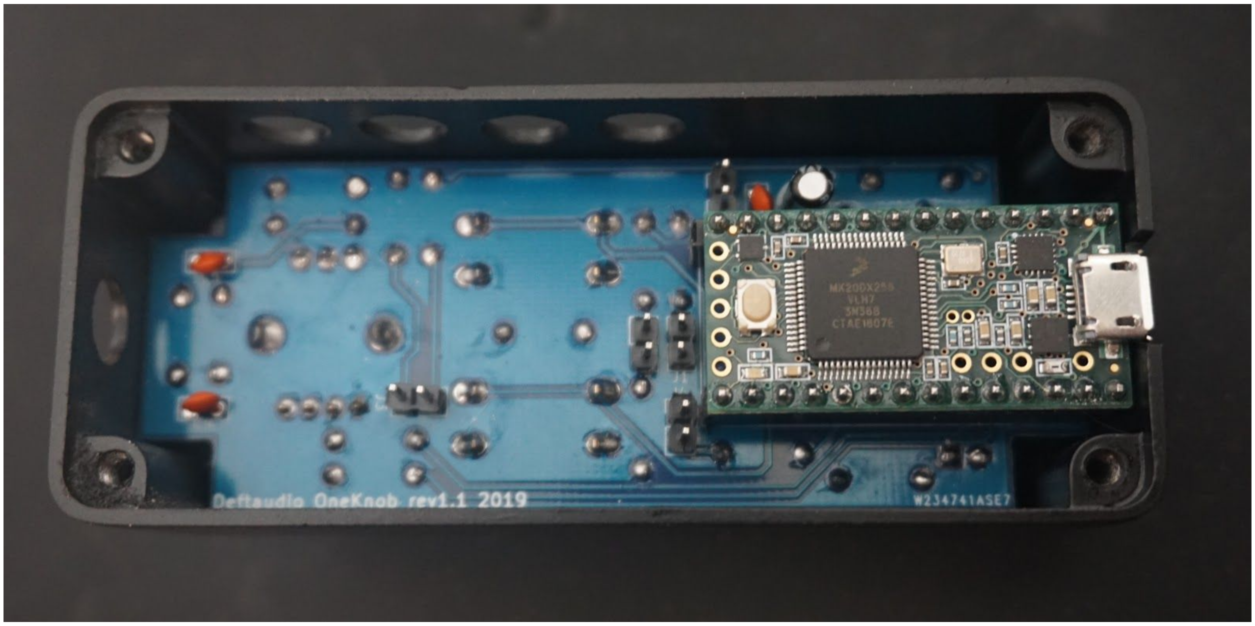
- Use a rasp to get 2 mm inside, so microUSB connector stays flat.
- Place a bottom lid, mark with a sharpie tool again and use rasp to cut a piece there too:



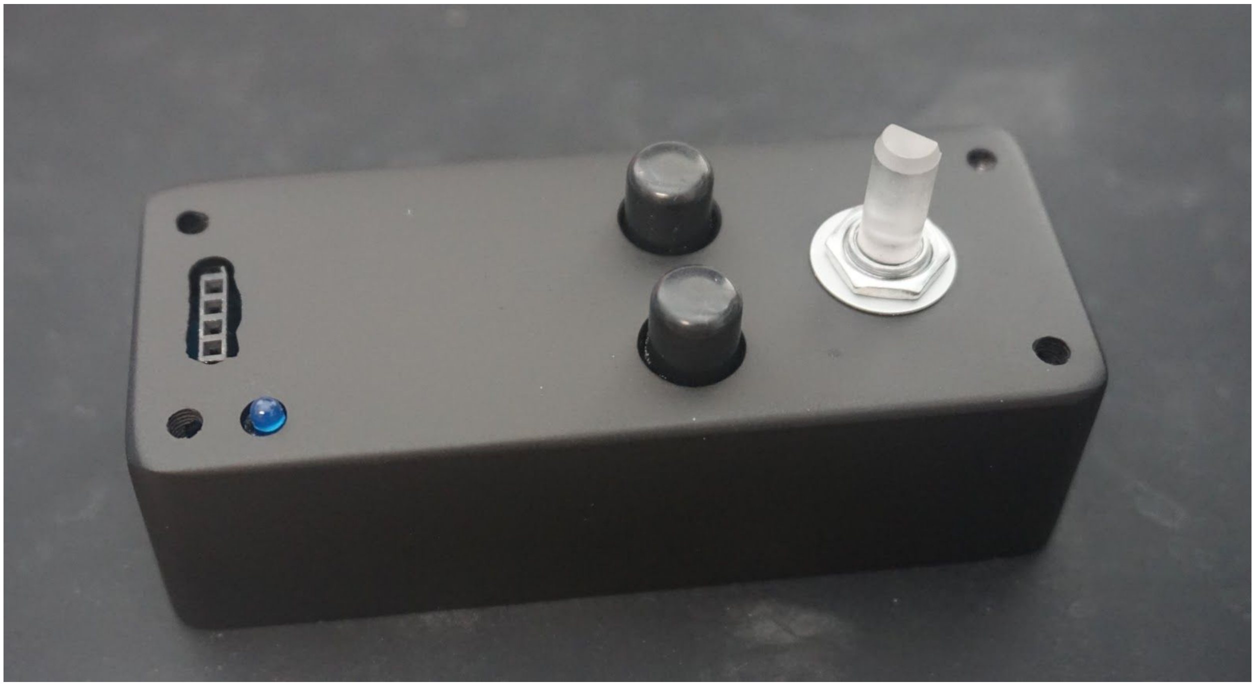
- Now it’s ready to be painted.

Final assembly.

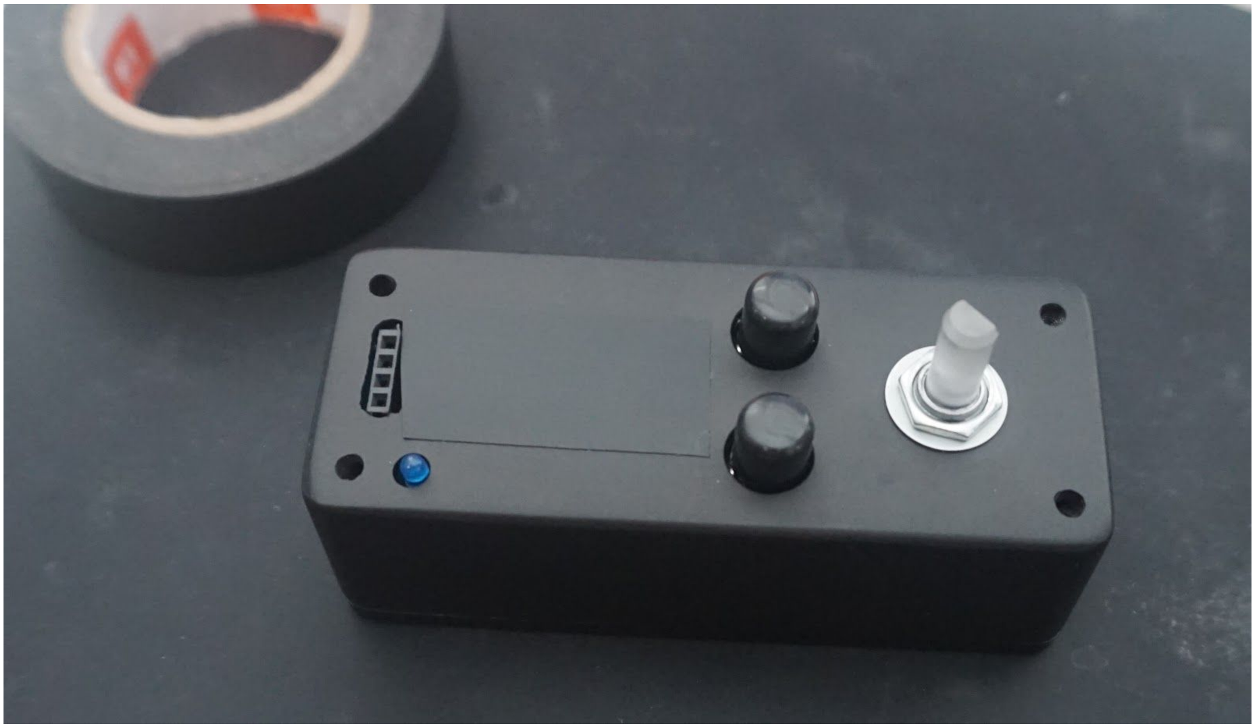
- Start an assembly by placing a board with Teensy controller attached into the enclosure:



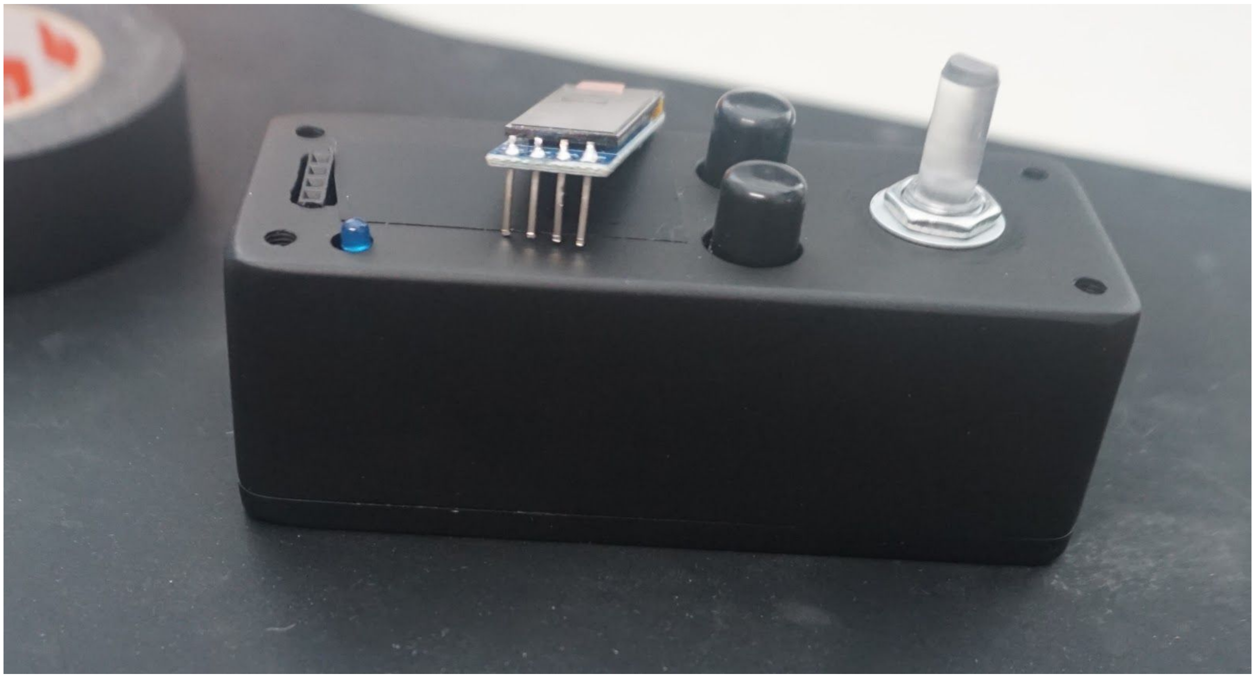
- Flip upside down, put a washer and a screw nut onto the encoder:



- Put a piece of black tape onto the place under OLED screen. This will help to protect from any accidental short such as if enclosure is not well painted.

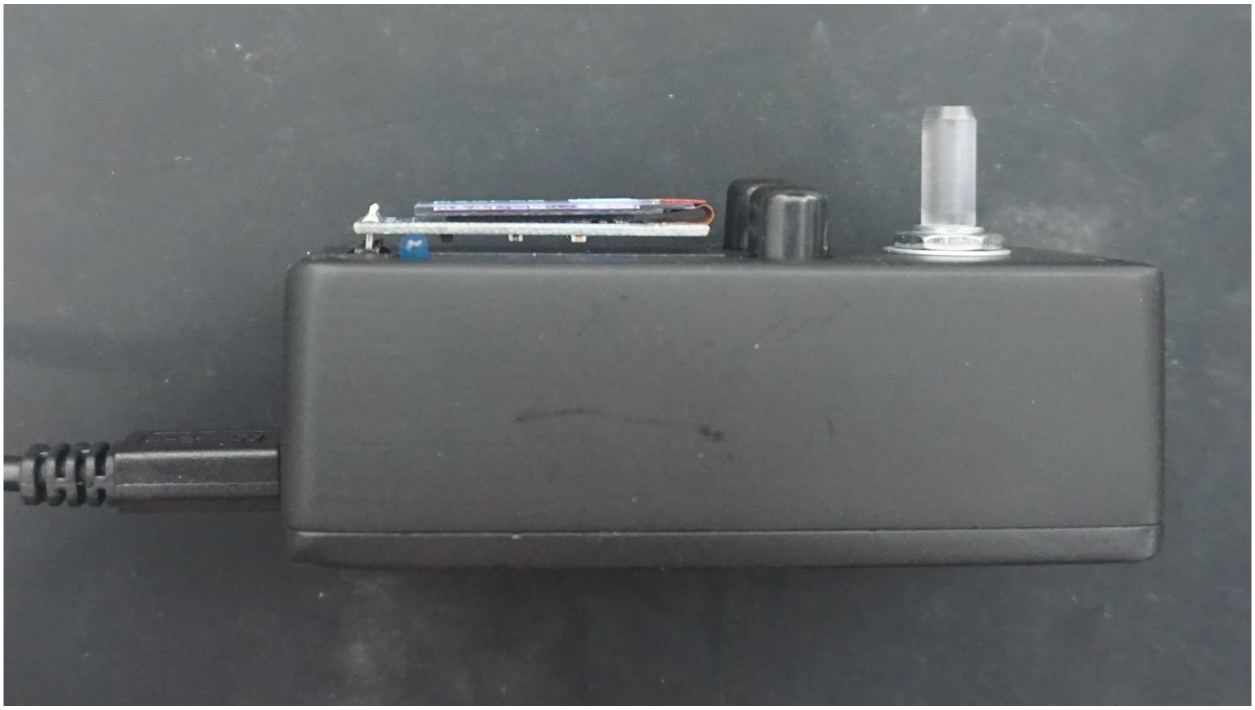


- Now you need to attach the OLED. Depends on your source of purchase it may come with soldered pins. They need to be adjusted - cut about 1-2 mm and remove black plastic.

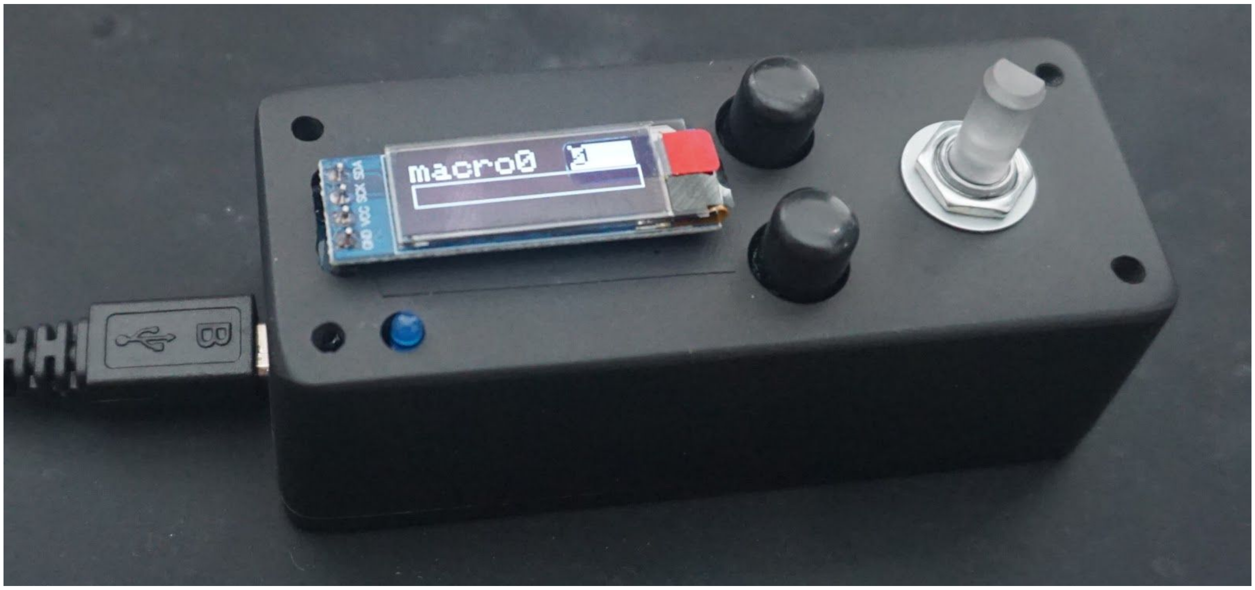


- Note: Certain OLED displays require significantly longer initialization time (up to 2.6sec) before they can operate. You can identify those by the link below. Extra initialization latency can always be added into MacroPot firmware(cause longer time to start the unit) as well as “resistor mod” applied. However, would be better if you use the right model from the beginning.
  http://torlus.com/floppy/forum/viewtopic.php?f=5&t=3240&p=17196#p17196

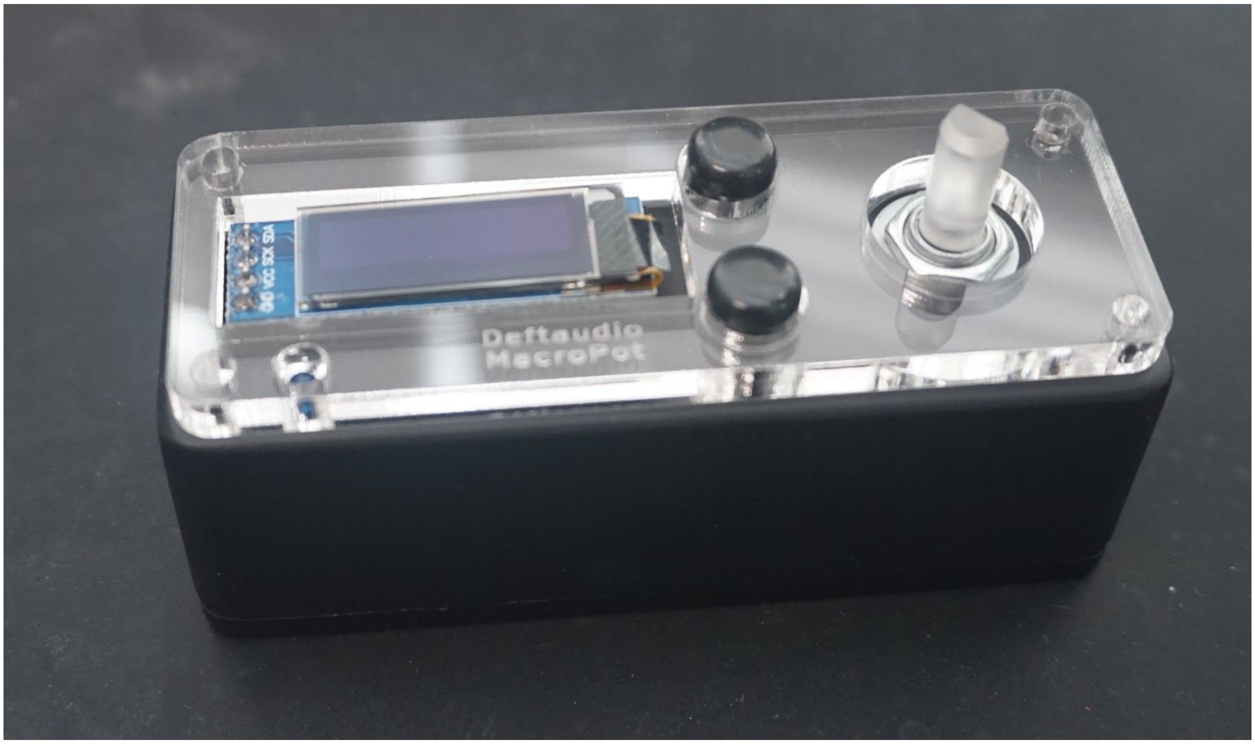
- Place OLED so it’s parallel to the enclosure and sits close to it:



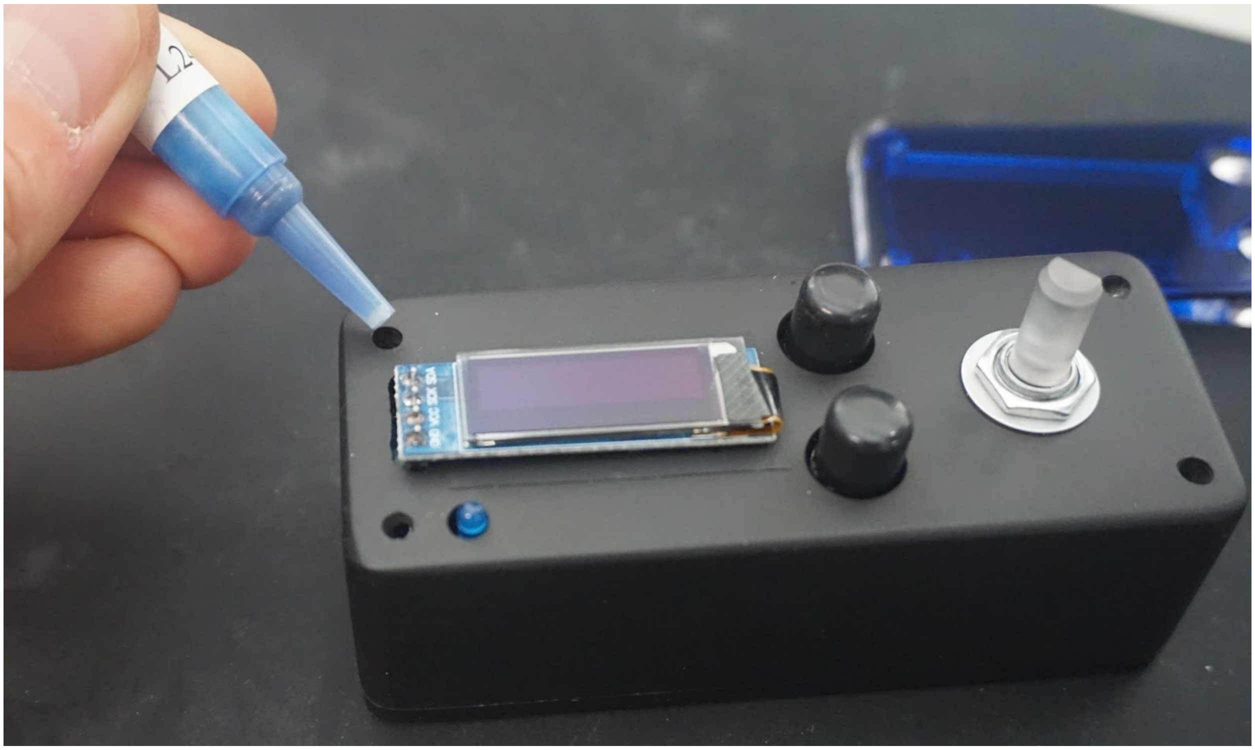
- Quickly test the unit before proceeding to further assembly:



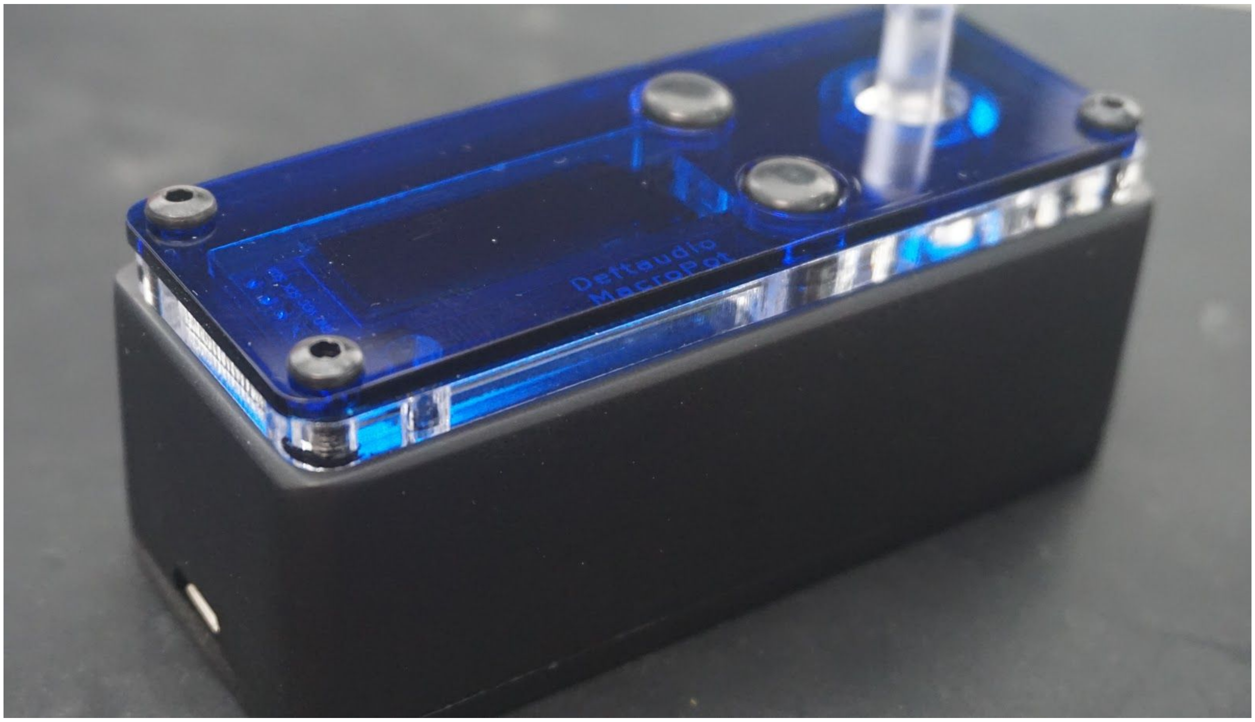
- Verify with middle acrylic layer that it covers it completely:



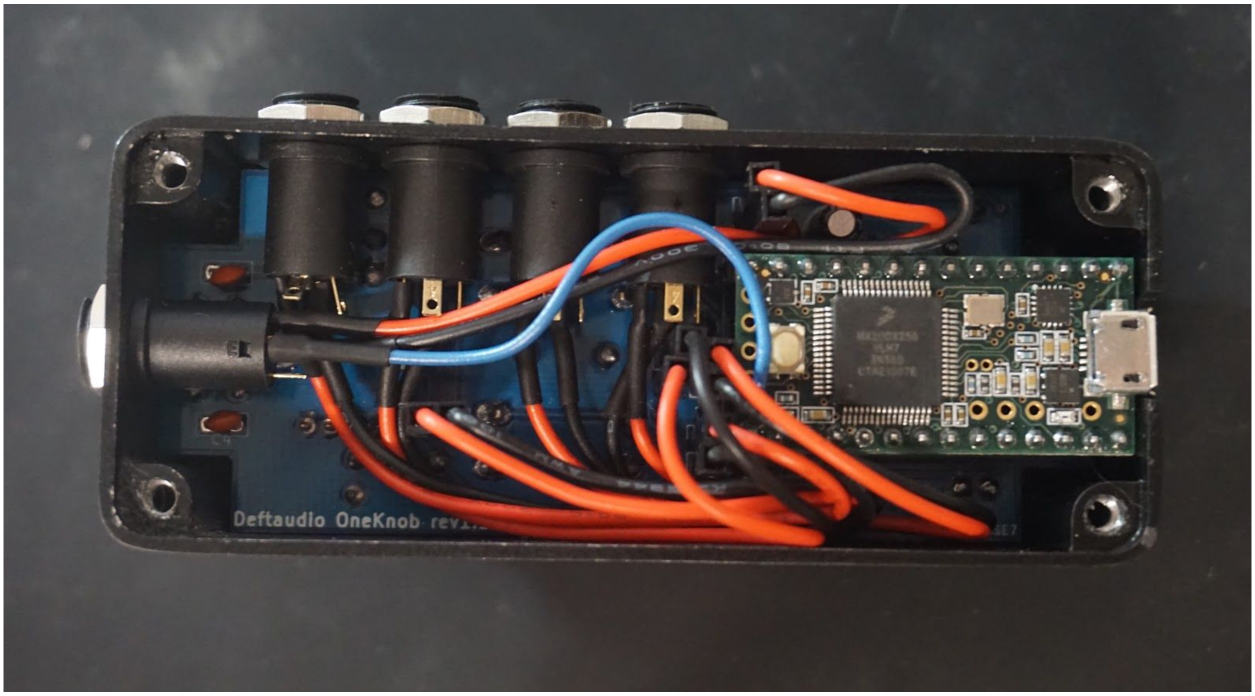
- Use a thread glue (or nail paint works well) to put a little inside 3mm screw holes. This will lock screws in the position after you tight them:



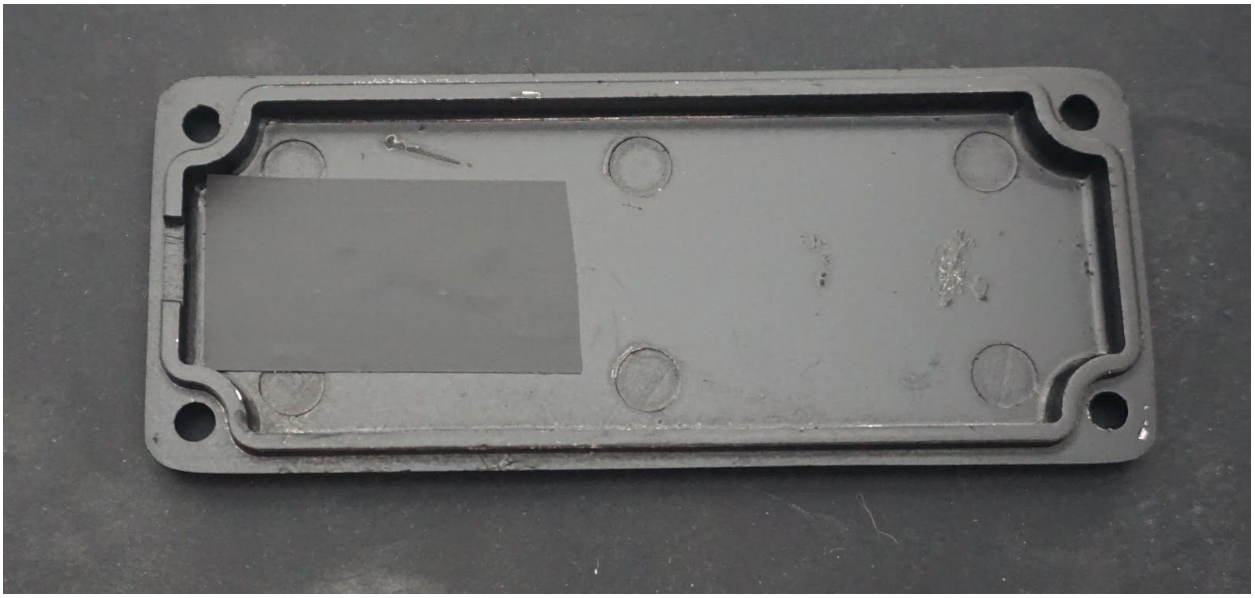
- Place a middle layer and top acrylic layer, put the screws in but don’t tight them until you have all screws and align panels. Then carefully tighten all screws. Keep in mind that aluminium is a relatively soft material. Don’t over tight to avoid a thread damage.



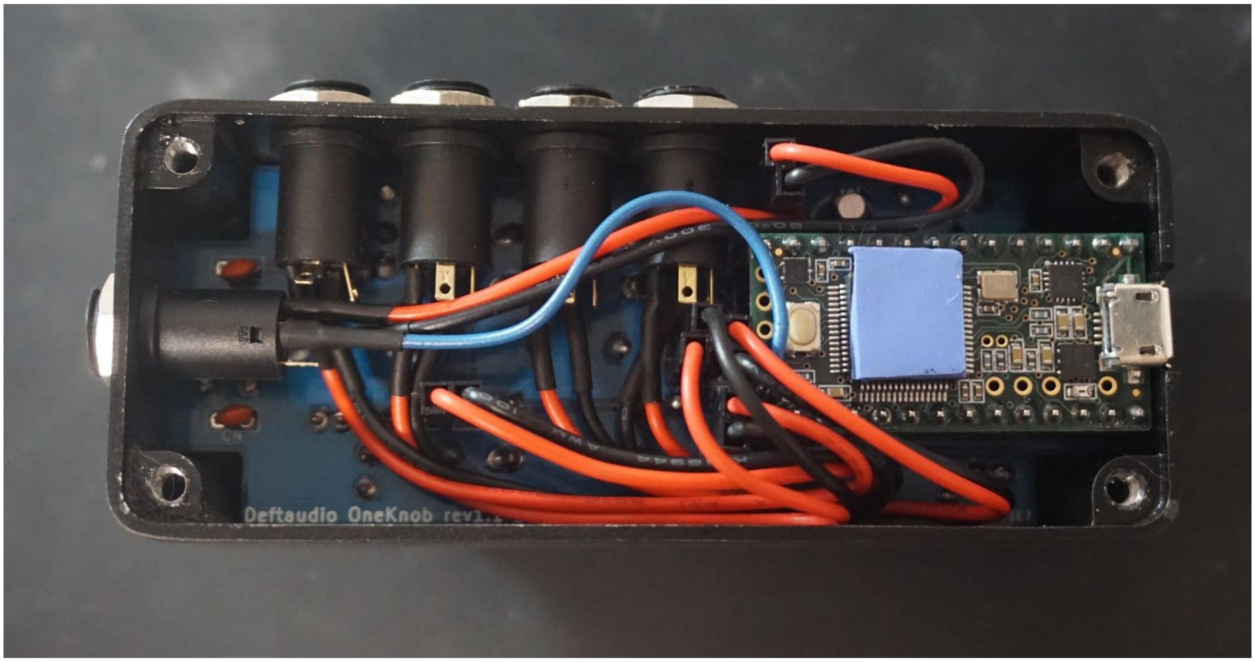
- Install all TRS connectors, starting MIDI connectors on the back side and audio out on the side. Connect per the diagram above and following desired TRS-MIDI Type A or B specification. The picture below is an example for Type A connection:



- Place a piece of black tape on the back side of the bottom lid matching the size of Teensy controller:



- Can also be a good idea to put a piece of 3mm TIM (thermal interface material) to cover a Teensy chip, so this provides some isolation, damping and thermal cooling(it’s barely warm even overclocked):



- Place a bottom lid, tighten the screws and cover them with rubber pads:

- Your device is complete. Feel free to start using it with some default macros, but shortly start learning a programming of this to explore all capabilities: https://docs.google.com/document/d/1nPIYxyJ24XFE5UrgwzTidY-a2B3J3utPwMfMKG4Hb1w/edit?usp=sharing