

MotorPot Assembly Instruction v1.0

General guidance

MotorPot is 12HP Eurorack module for enclosures with 48mm depth or more required. It can be powered from a Eurorack power bus or use external 12V power for the stepper motor. This is helpful to isolate those rails to avoid high current consumption which can approach 2A@12V. MotorPot has a highly modular design which allows it to test it during the assembly in pieces before putting it all together.

Main circuit (controller and other ICs) are powered from Eurorack PSU, either 5V or 12V. **It's recommended to use 5V rail, which means U5, C3 and D6 might not be installed.**

To get started, please, familiar yourself with the latest BOM and schematics on Github:

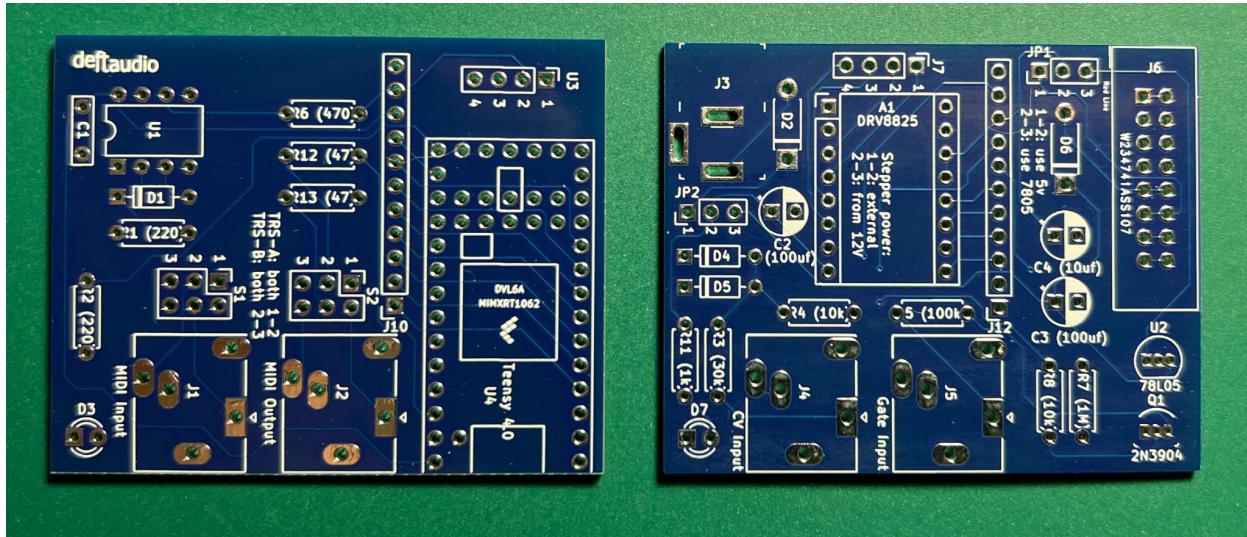
https://github.com/Deftaudio/Midi-boards/tree/master/Eurorack_MotorPot

Also, you may find it helpful to follow an introduction video to familiarize yourself with its basics:
<https://youtu.be/hMhLqPYqAdg>

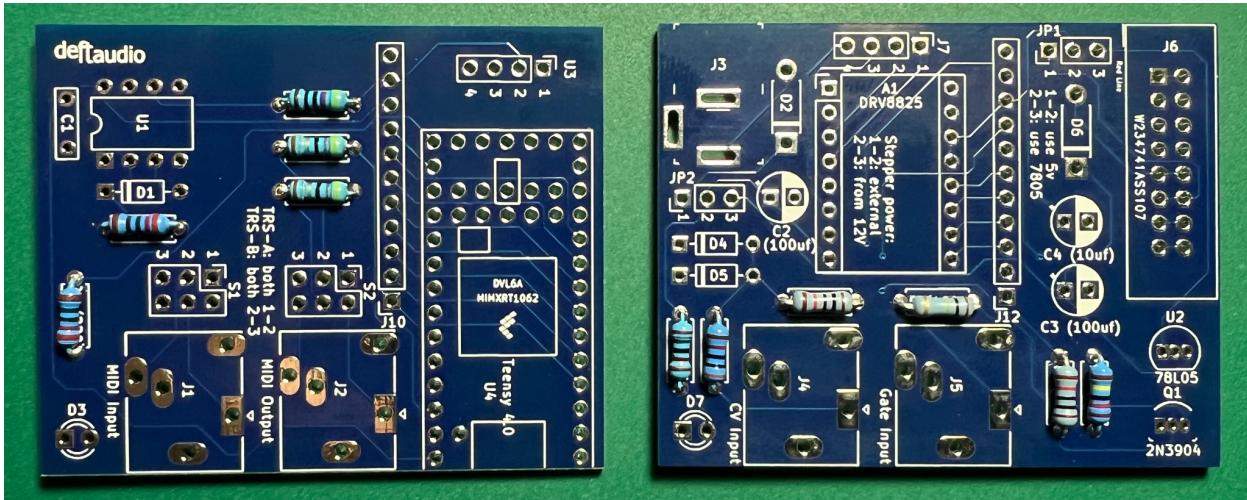
Board Assembly

MotorPot consists of two boards stacked together, stepper motor, display and aluminum faceplate.

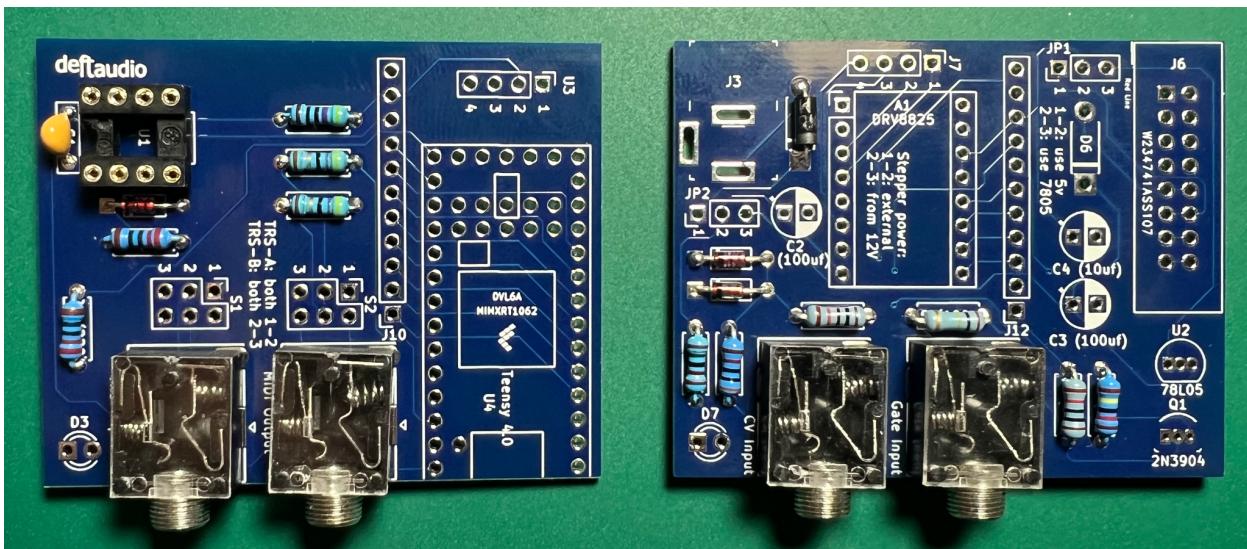
Put two boards side by side without any parts populated:



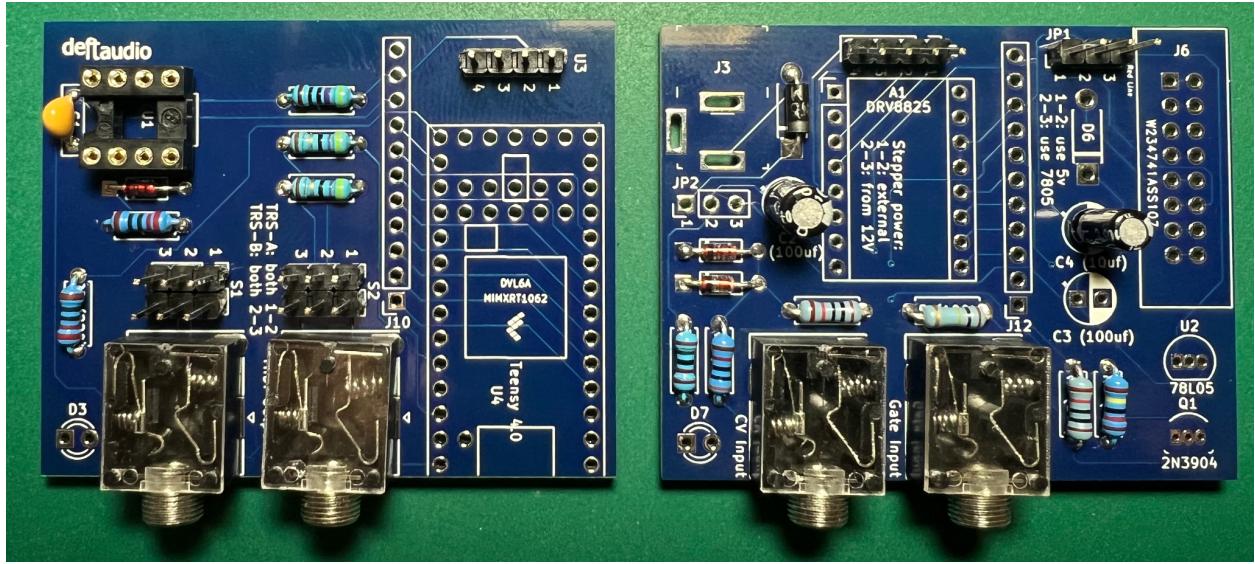
Start assembling by putting together passive components with low profile first, such as resistors, diode:



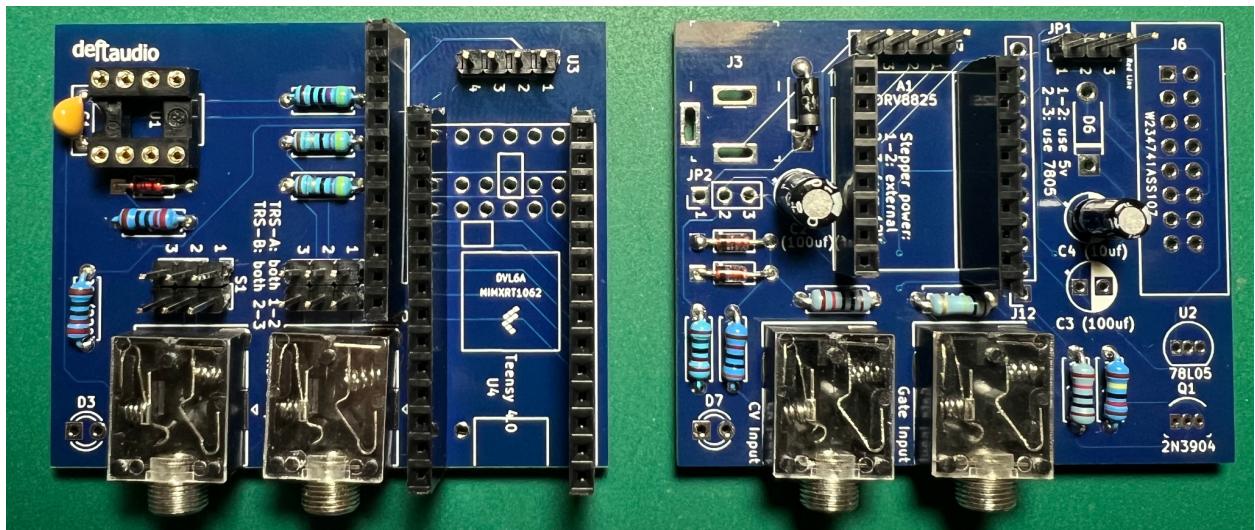
Then add IC sockets and 3.5mm connectors:



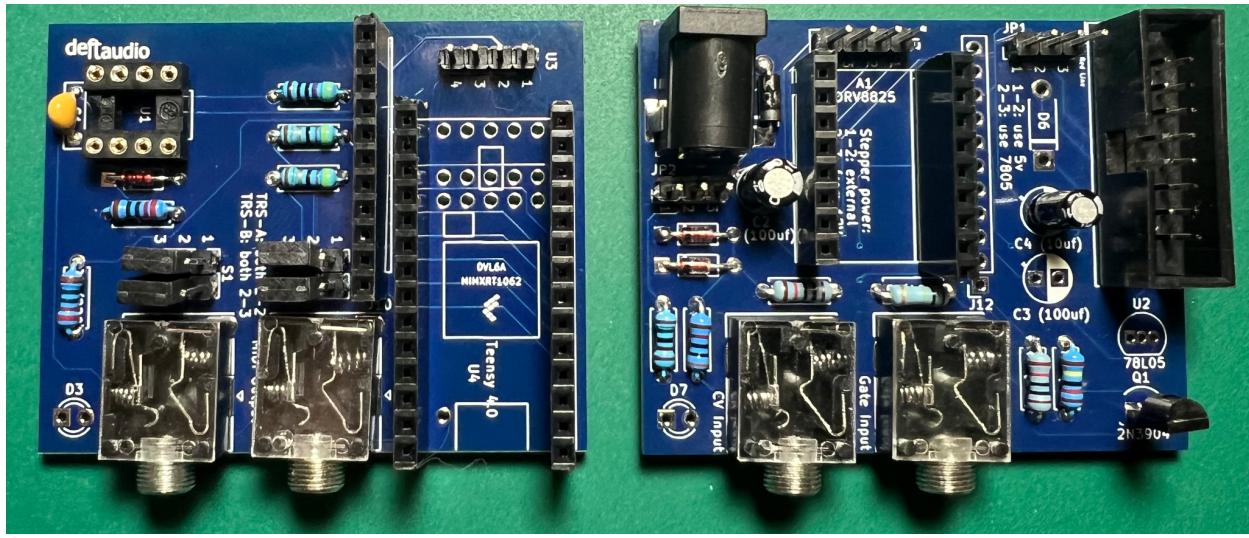
Add jumper headers, don't populate J12:



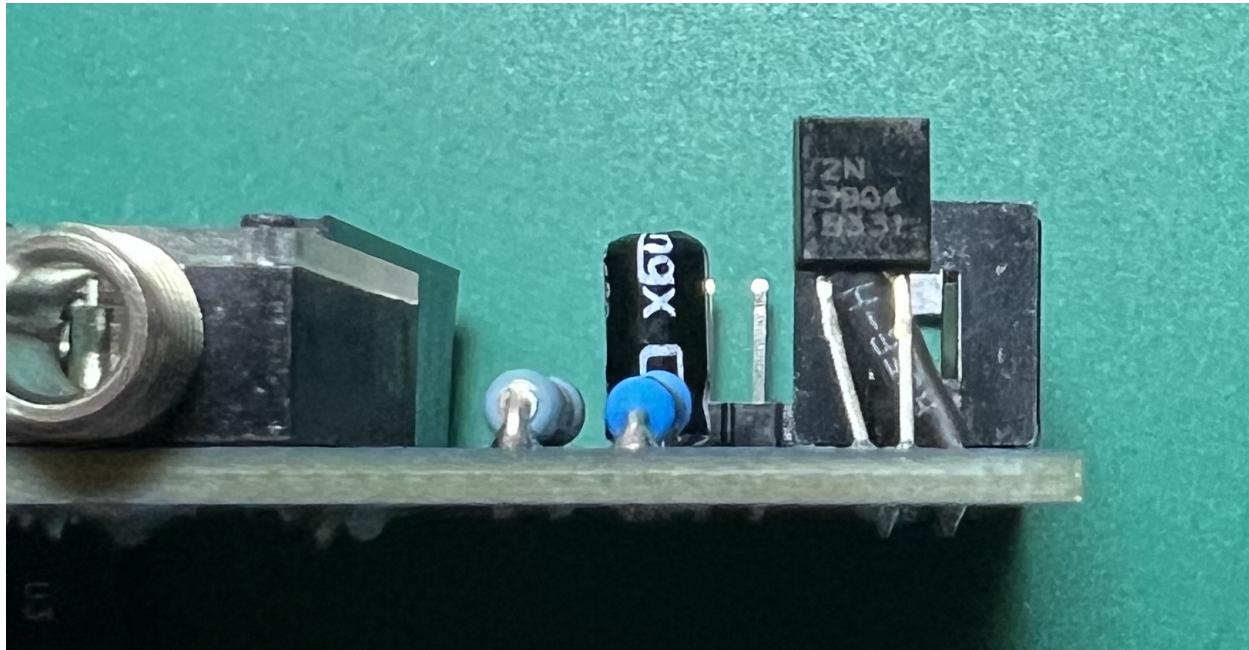
Add Teensy and DRV8825 female headers:



Add Eurorack male socket, S1-S2 jumpers and 2N3904 transistor. **Important: due to footprint mistake 2N3904 (or 2N2222) transistor need to flip legs 2 and 3, i.e. leg 3 goes to 2, and leg 2 goes to 3.**

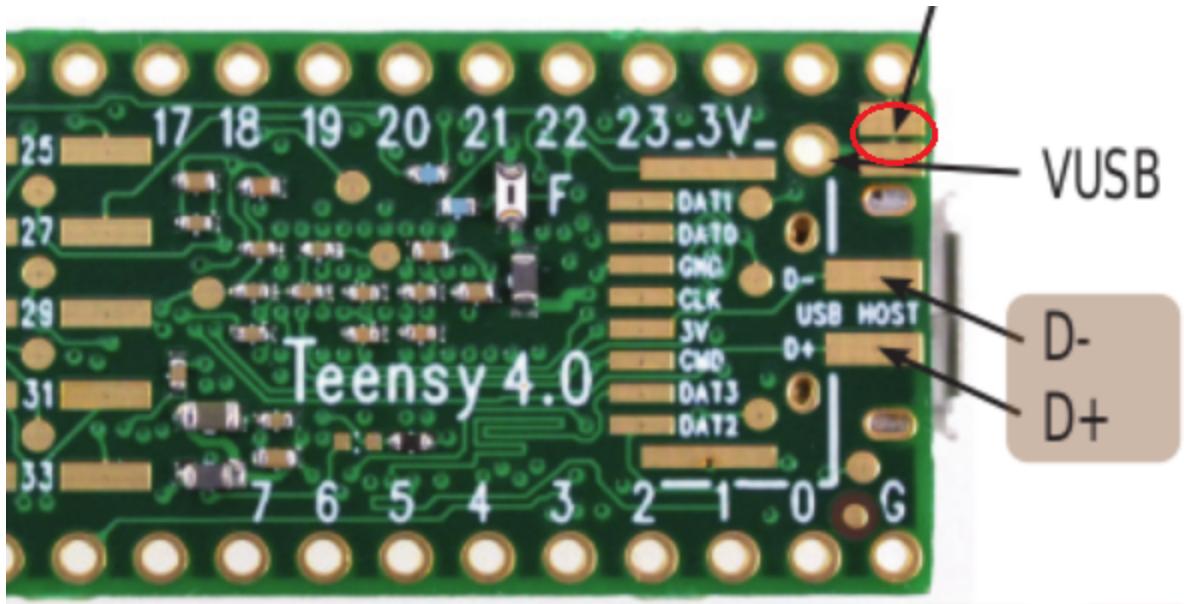


Here is a closer look of 2N3904 transistor:

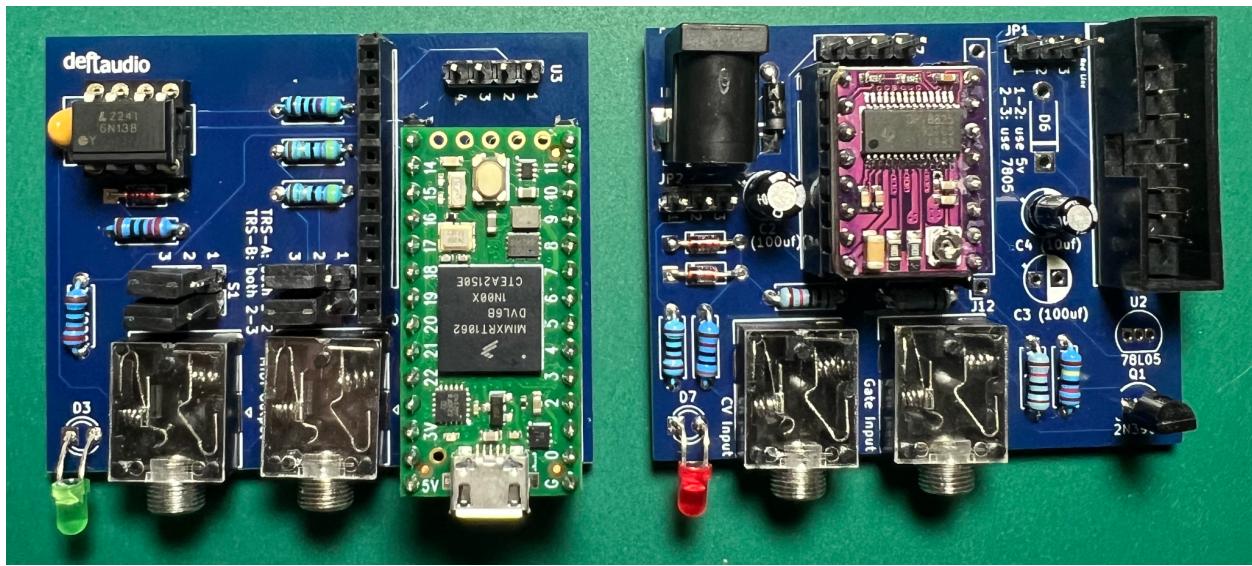


Solder Teensy pins and cut VUSB

It's important to note that by default Teensy 4.0 microcontroller is powered from a USB bus and supplies 3.3V and 5V to the rest of the circuit. For the purpose of the standalone operation, the circuit can be powered from Eurorack connection, which illuminates having USB cable connected all time. However, connecting USB cable could cause Teensy damage when powered externally. To avoid this, a special track should be cut on the Teensy board. It's marked in red, it's a tiny line that connects two pads that need to be cut. By doing so, Teensy can no longer be powered from the USB cable.



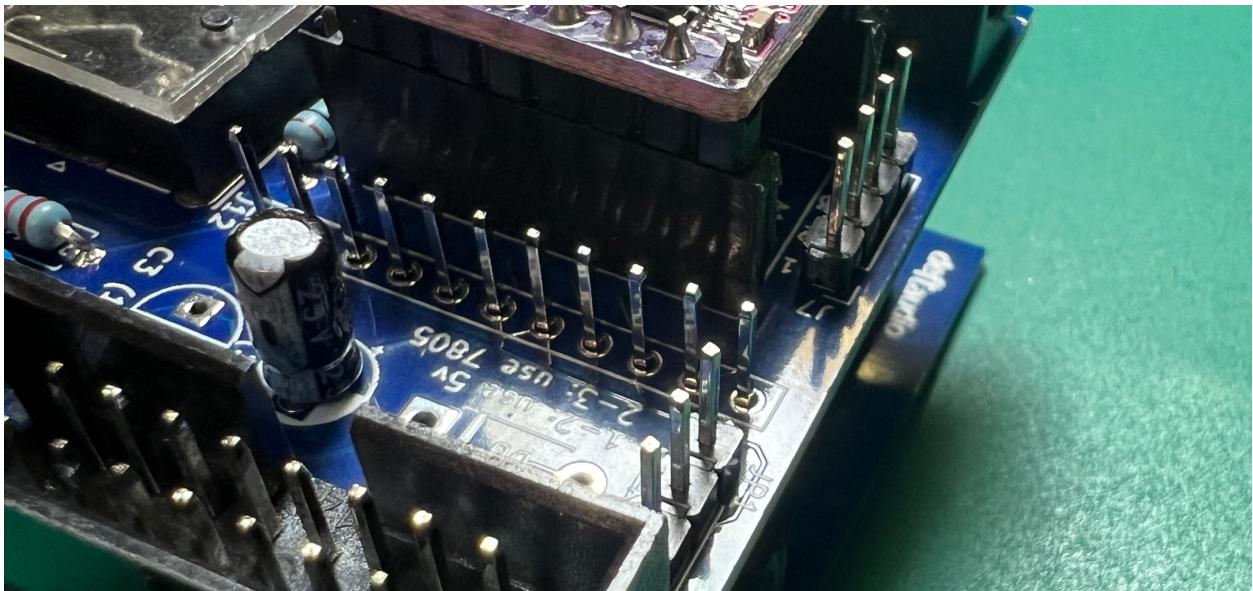
Plug in Teensy, 6N138, solder LEDs:



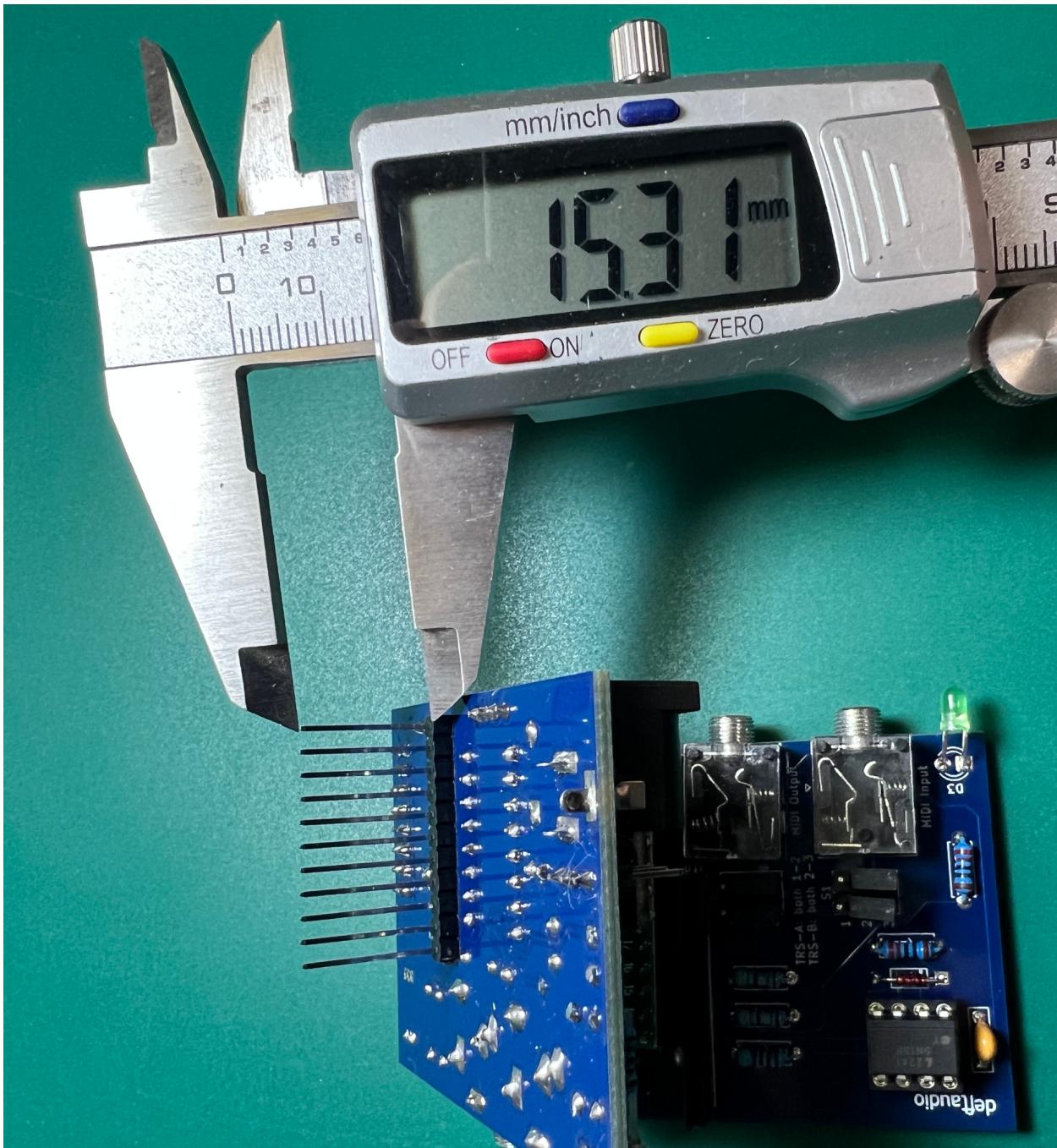
Take 11pin long pinheader and insert it into female header J10. Mount both board to the front panel and squeeze them together, so long pins will go through the J12 connector. Make sure that boards are parallel:



Now solder J12 pins in place, below is the look:



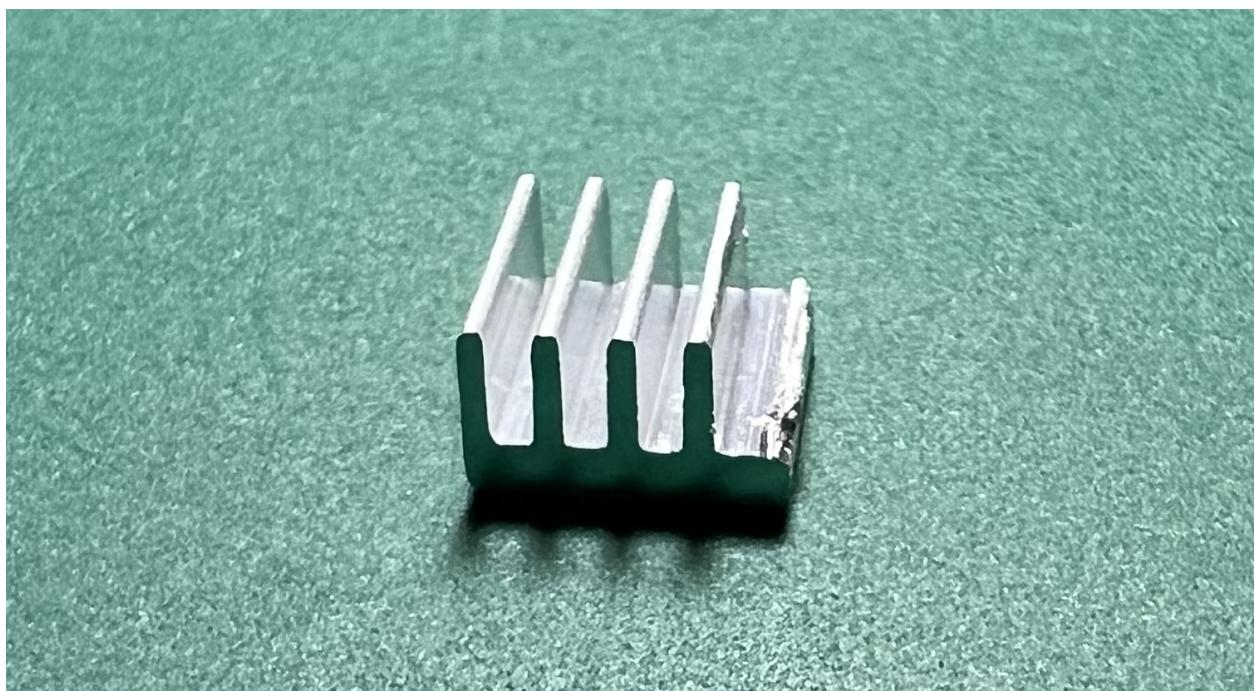
You may also refer to these measurements of the long pin header. Instead of squeezing, solder as normal and the cut the extra length:



Cut the extra pins of the OLED display from the solder side. So they won't make a circuit short when attached to the front panel:



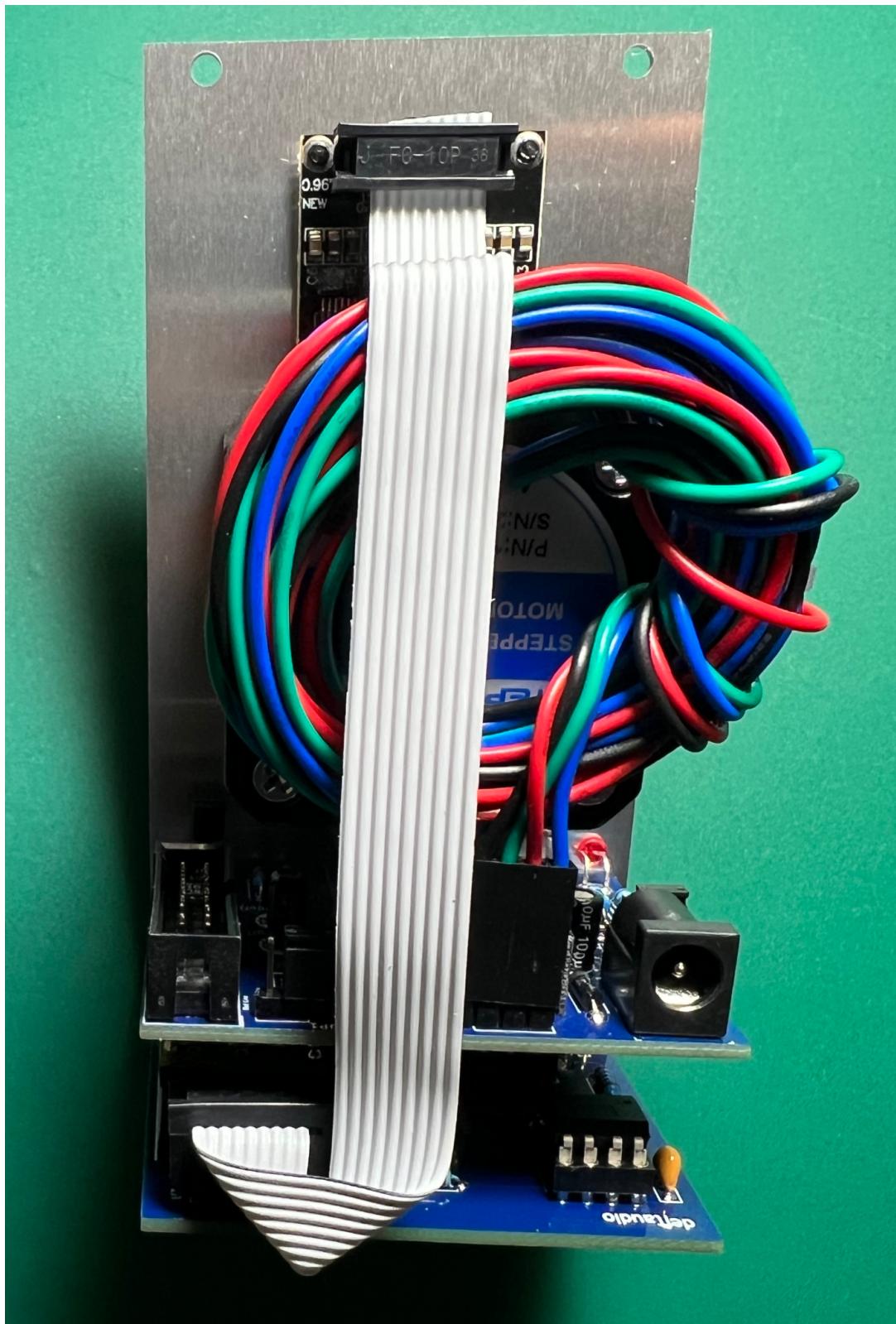
Cut on side of the heatsink as shown below:



Assemble everything together, connect step motor (black wire is the pin1):



Make a 10-pin cable for the OLED, note only 4 pins are used, GND is pin1. Follow the pin1 orientation.



Finally, set JP1 and JP2 in 1-2 position.

Setting up MotorPot

Uploading Teensy firmware

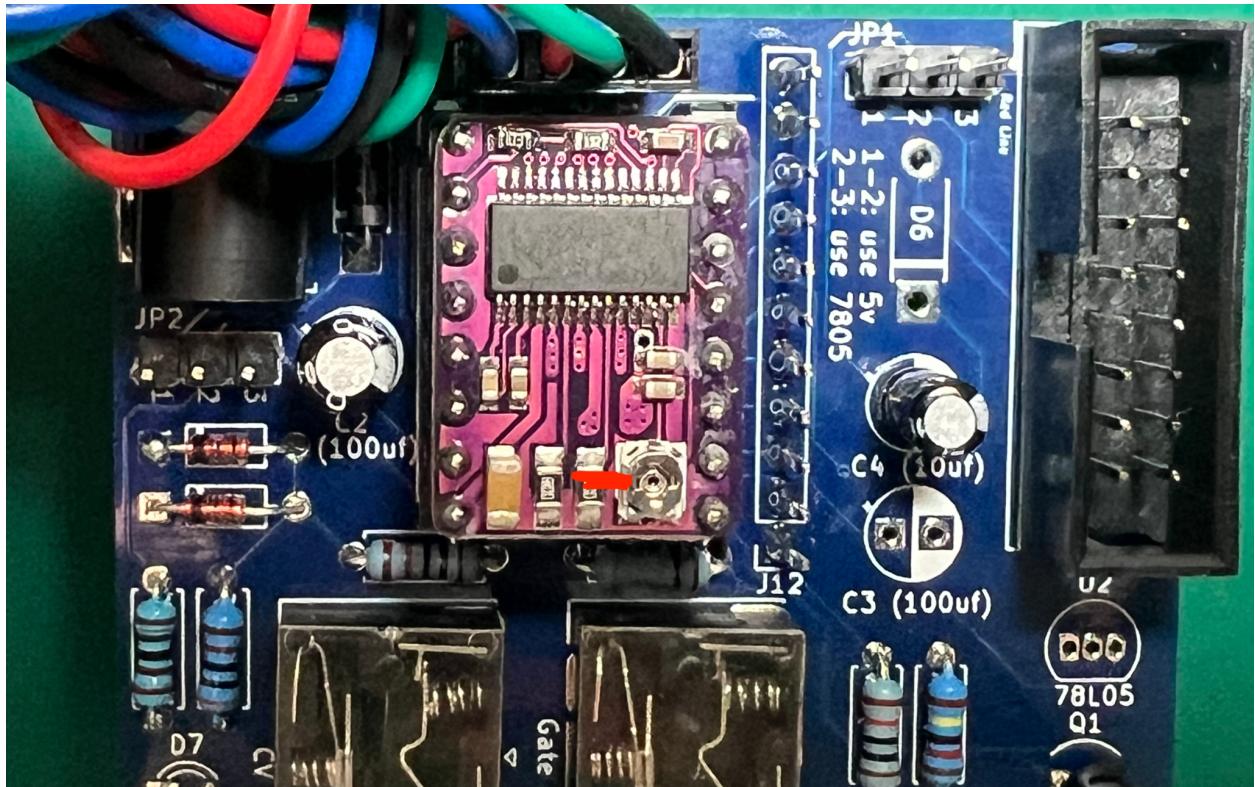
Teensy microcontroller is programmed within the Arduino framework. Starting from Arduino 2.0 it can be easily integrated into the environment without a need for a custom Teensyduino application.

- If you plan to modify the code please follow complete Arduino installation steps here: https://www.pjrc.com/teensy/td_download.html MotorPot firmware is located in the same GitHub repository.
- If you just want to load the existing firmware, it's easier to use Teensyloader instead, which is a cross platform firmware update tool that you can download separately: <https://www.pjrc.com/teensy/loader.html>
- The github link provides precompiled binaries for Teensyloader. First burn eprom_clear binary which just erases previous eprom content (in case Teensy was used for other purposes) then restart it and then program the main firmware.
- As the result at the first power on you should see MotorPot greetings on the display.

Setting DRV8825 driver current

DRV8825 is the modern high efficient and flexible step motor driver that supports up to 2.5A. The current has to be adjusted per step motor used. By default MotorPot is configured with NEMA17 17HS13-0404S1 1.8deg stepper the recommended current ~ 0.4A. Here are steps to perform:

- Disconnect stepper motor if connected
- Power on MotorPot for the first time with Teensy firmware loaded. It doesn't need to be configured with any settings, but it should be loaded as it initializes DRV8825 at the start.
- Connect the voltmeter to the ground and to the adjustment pot center. It has a metal center part, so just touch it in the center. This is a Vref voltage for the circuit.
- It should be set by the formula: $V_{ref} = \text{Current Limit} / 2$, so for the default step motor it should be around 0.2 - 0.3V set.
- Please, refer to a picture, or just use it as a reference without measuring voltage. You may see that the potentiometer key is set around 9 o'clock.



SW Customization options

Open source MotorPot firmware provides limitless opportunities to continue firmware improvement and development under BSD 3-Clause License terms. In the meantime there are several parameters that you may start tweaking right away by just changing their current values. Those are related to the step motor configuration and precision.

- You can adjust the max speed of the stepper which is set specifically to the model. For example for the default 17HS13-0404S1 step motor the max speed is set to 600. Setting higher speeds may result in the loss of torque and skipping steps. “MaxSpeed” constant is setting this.
- Another parameter that can be tuned is the microstep resolution. Step motors can be driven in higher precision (lower speed is the downside) by moving from full step operation into half step, $\frac{1}{4}$, $\frac{1}{8}$, 1/16 and even 1/32 operation for DRV8825 drivers. Consider changing “M0, M1, M2” constants for that and refer to the DRV8825 Datasheet, table 1:

<https://www.makerguides.com/wp-content/uploads/2019/02/DRV8825-Datasheet.pdf>

