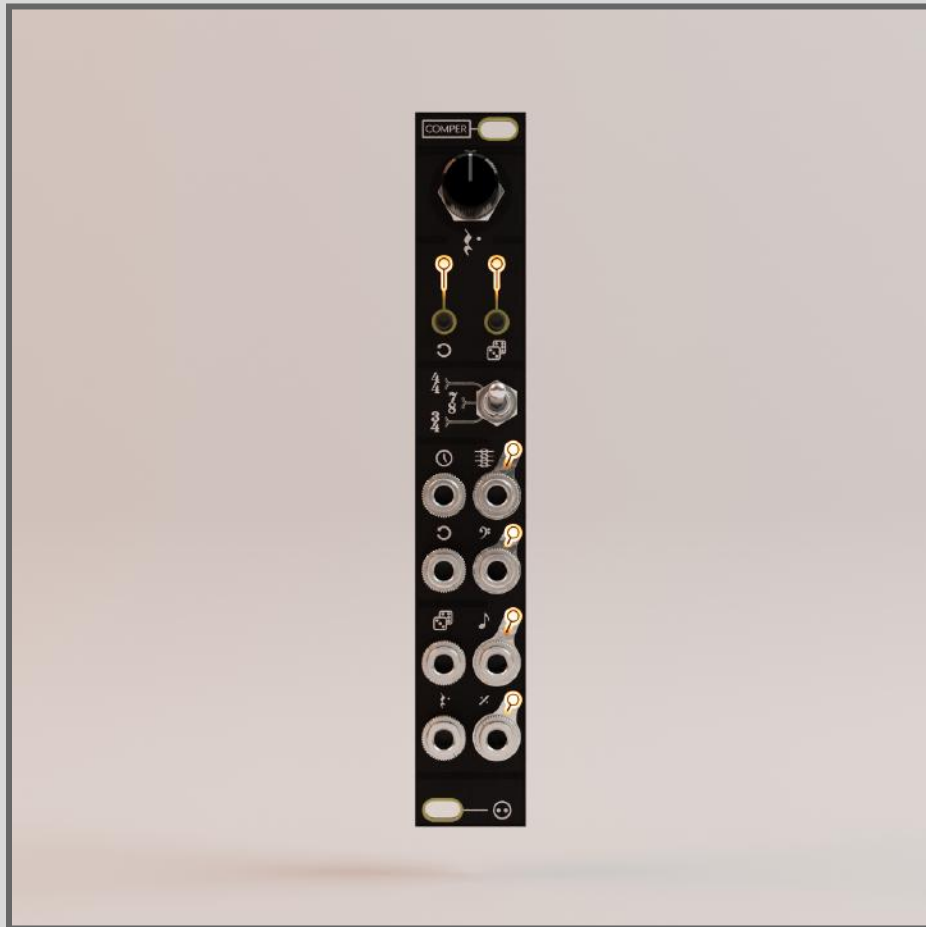# NO<u>H</u>

COMPER - Build Guide

V1.0

Introduction-

This build document aims to help you with the assembly of the module. It is quite difficult to build and is not recommended as a very first build.

It is highly recommended that you read through the entire guide once before starting with the assembly.

Soldering a module together is always a relaxing and gratifying moment when done properly, don't hesitate to take breaks when you feel you need to and double-check steps before progressing in your assembly.

Requirements-

To complete this build you need:

- A Soldering Iron and Solder
- Pliers and Side Cutters
- A Multimeter (optional but **extremely** recommended for debugging)
- Safety Glasses for assembly
- The Thonny IDE if the firmware needs to be installed / updated- Thonny is available here

Make sure you also take a look at the Bill of Material (BOM) to check whether you have all the required components to complete the build.

Build Guide-

• If not already mounted, start by placing the keyed header, be mindful of its orientation and match it with the indication on the PCB.

• You can solder a pin on each side, check whether it sits flat on the PCB and solder the rest of the pins if it does.
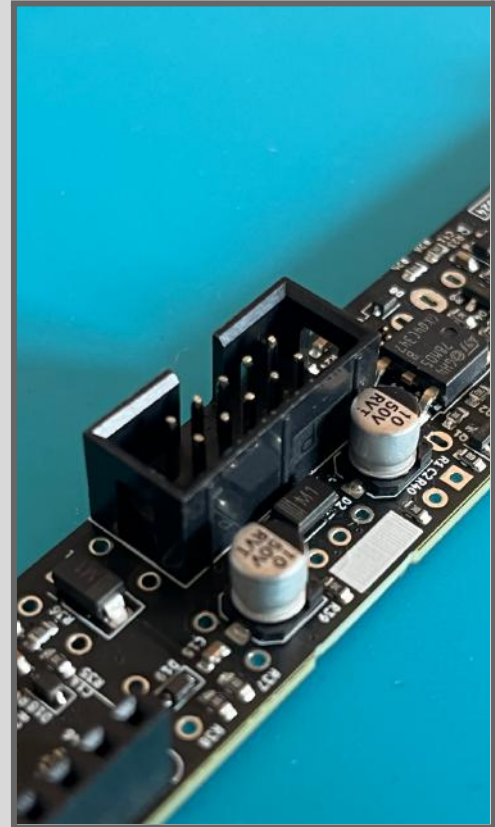
• Again, these components should be already mounted. If not, mount the PCB sockets and solder them in place

• You can solder a pin on each socket, check whether they sit flat on the PCB and solder the rest of the pins if they do.
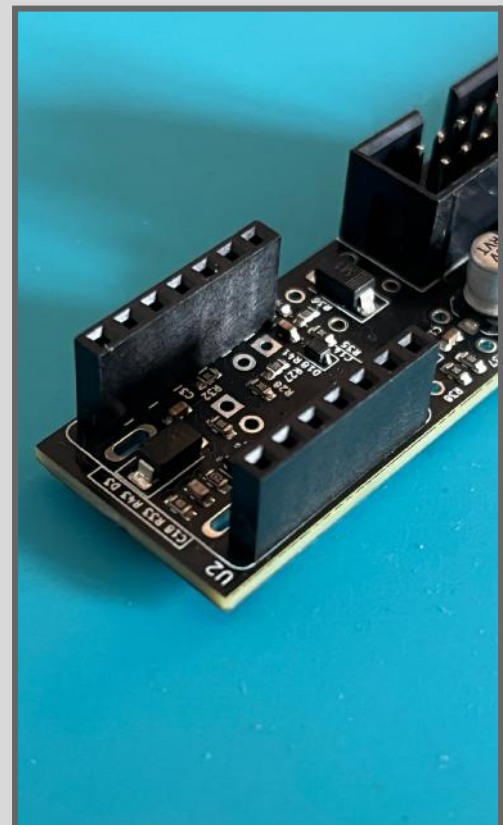
• Flip the PCB around and insert the six LEDs **without soldering them**.

• Polarity matters for these components, the shorter leg of the LED goes in the square pad on the PCB.

• Again, **do not solder them**.
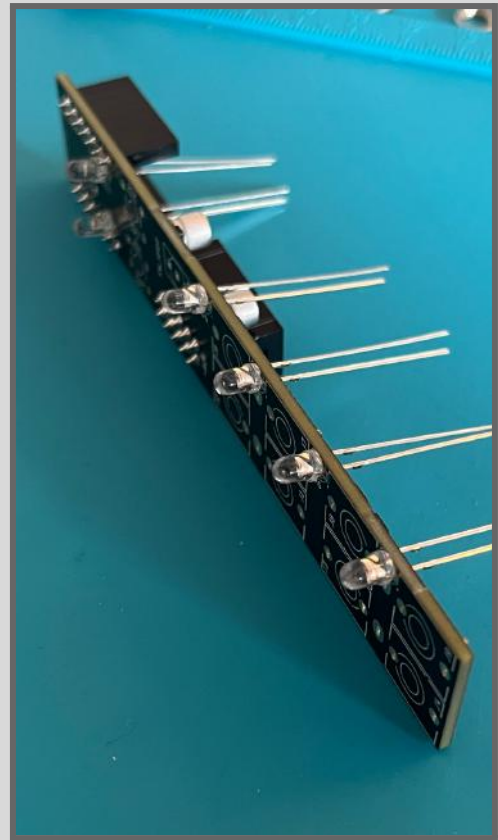
• Mount the potentiometer and the two buttons **without soldering them**.

• The polarity of the buttons does not matter. If you struggle to fit the potentiometer you can flatten the mounting pins with pliers.

• Again, **do not solder anything yet**.

• Mount the SPDT switch and all the jack sockets to the PCB **without soldering them**.

• The polarity of the switch does not matter either. The jack sockets share their ground pins with the row above or below as shown in the picture on the right.

• Again, **do not solder anything yet**.
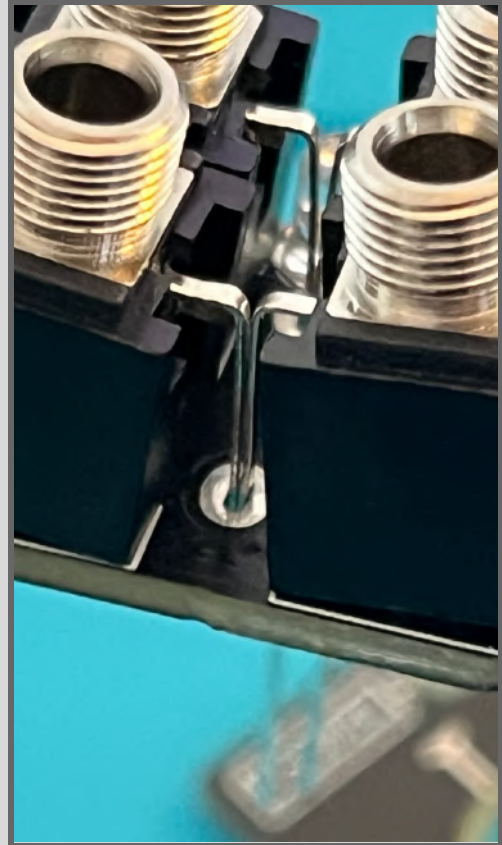
• Check if everything still sits flat with the PCB as nothing is soldered yet.

• You can add washers to the potentiometer and the switch so that they sit level with other components from the front panel.

• Again, **do not solder anything yet**.
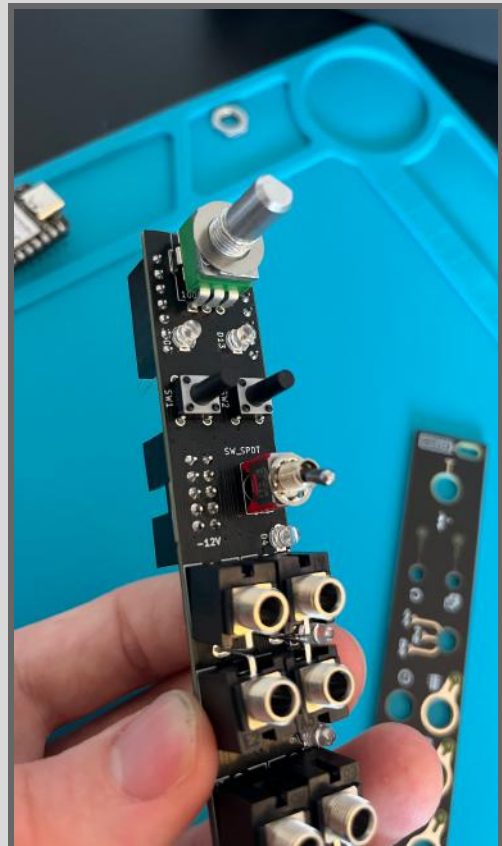
• Put the front panel on and hand tighten the nuts for the potentiometer, the switch, and all of the jack sockets

• Check again if everything is still well put in place and sits flat on the PCB and the front panel before starting to solder.

• **You can start soldering**.

• To solder the LEDs in place, push them towards the front panel and solder **only one** pin of that LED.

• Then, reflow the pin (i.e. make the solder melt again) and while the solder is liquid use the leads to orient it correctly (**make sure you do not burn yourself as the reflowed lead will be hot**).

• Once you are happy with how the LED sits, remove the soldering iron and the LED will stay in place. You can then solder the second pin and clip the leads.
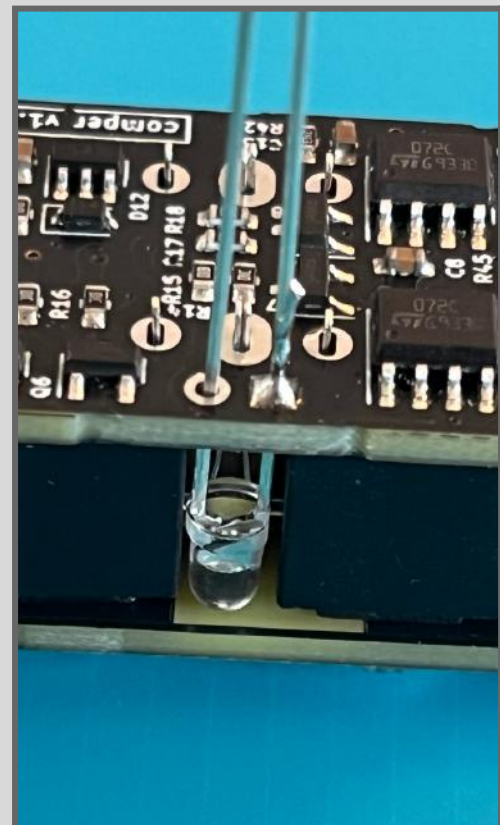
• You can then solder the rest of the components. Though because the PCB is very dense it is important to be extremely careful and aware of where your soldering iron is.

• Simply remember to **have your soldering iron in an orientation that puts it as far away from other components as possible** (as shown on the picture)**.**
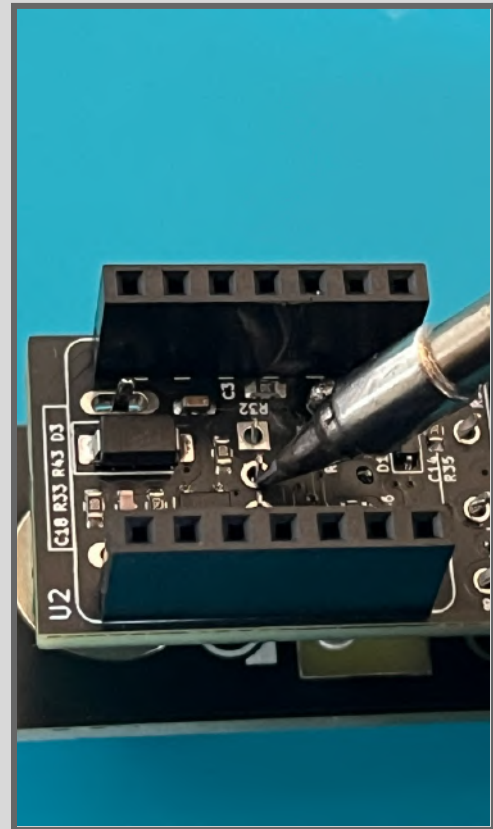
• Once all the components are properly soldered in place and double-checked, you can insert the male headers into both sockets, with the longer side of the header going in.

• Insert the microcontroller board in these male headers and solder the pins. This will ensure that they are parallel and fit in the socket

• The next step only applies if your MCU is **BRAND NEW**. Comper DIY kits should have the firmware pre-installed.
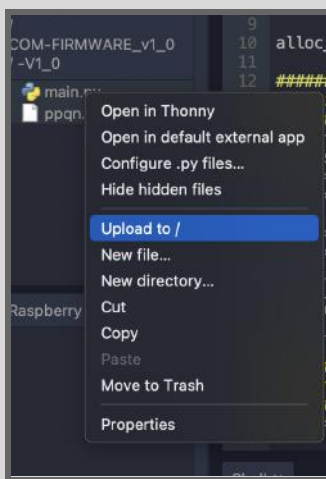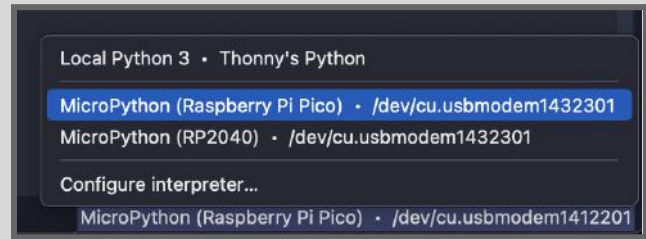
Installing/Updating the Comper Firmware

• Get on the Comper [product page](#) and download the firmware file. It should be a compressed zip file with the correct version of MicroPython (1.23.0) and another folder inside of it.

• Plug the Microcontroller (MCU) to the computer, make sure you use a USB cable that has data lines and that you keep the "B" button pressed before plugging it in.
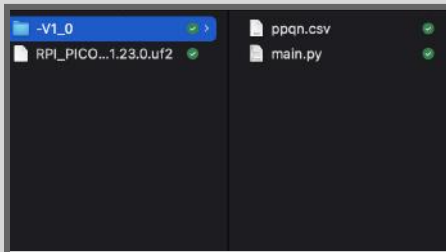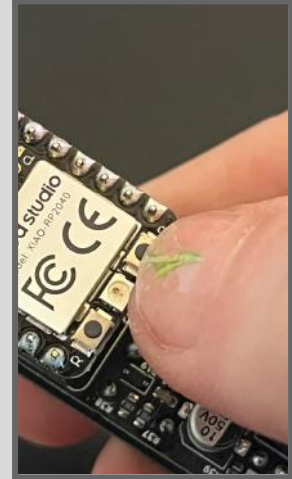
• If the MCU is new you should see it as a peripheral device once plugged in. Go to the firmware folder and drag and drop the ".uf2" file onto the MCU. This will install the correct version of MicroPython on the Pico (v1.23.0).

• Once MicroPython is installed on the MCU, install/open the [Thonny IDE](#) and connect to the MCU as a device from the bottom right menu.

• After connecting to the MCU, erase all of the documents in it if you are updating the firmware by selecting and deleting everything.

• Once that is done or if you are installing it for the first time, select all **two** files from the firmware folder you downloaded and select "upload to /" to add all needed firmware files to the MCU.

Unplug the MCU, do a [module check](#), and enjoy the new firmware :)

Inspection -

While the assembly might be finished, **it is not a good idea to plug power through before inspecting the module.**

• Start with a visual inspection, looking for any pins you forgot to solder, pins you soldered too much and shorted to other pins, or bad solder joints.

• Check the polarities of components where polarity matters (i.e. the power header, the MCU, and the LEDs).

• Look for any damage on the board, specifically whether tracks or components got damaged.

• Finally, using a Multimeter, check for shorts between the power pins (12V to GND, 12V to -12V, and GND to -12V).

Module Check (debugging) -

• Plug an external clock in (square LFO, clock) and you should see it come to life.

• All *output* LEDs should be blinking in their respective sequences.

• Pressing the reset button *blinks* its LED, while the random one lights up for *longer*.

• Move the rhythm knob fully *up* and fully *down* in 4/4 time signature. You should switch between *swing* and *straight* patterns every other bar.

That's all !
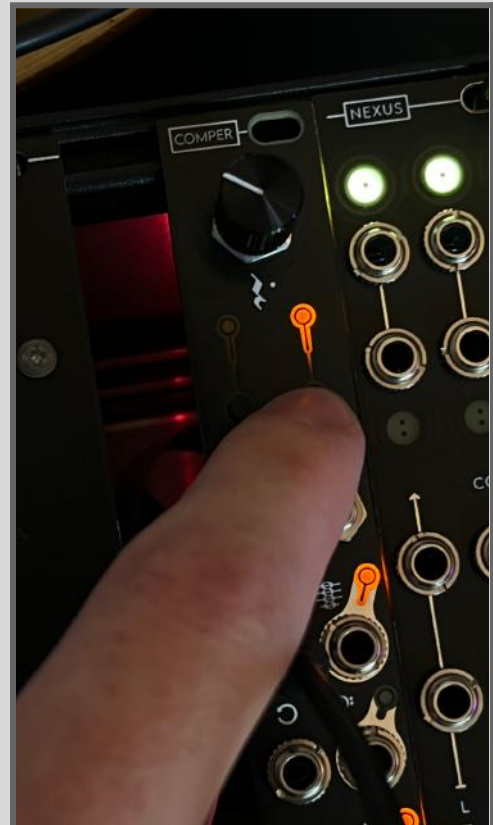
Conclusion -

This guide went through the assembly process of the module and what to be careful of when building it. If you haven't already, you can read the user manual to know what the power requirements are and how to operate it.

You can find this build guide, and other important documents regarding this module and others on the [NOH-Modular website](#).

I hope you'll find good and interesting use out of this module.