OPEN HARDWARE & FREE SOFTWARE: **TEUFEL MYND – A CASE STUDY**

fsfe

CRADLE TO CRADLE ELECTRONICS INITIATIVE

CRADLE TO CRADLE NGO

Open Doors, Closed Loops

When our partners at the Free Software Foundation Europe put the Teufel Mynd to the test, they bought it with their own money, opened it up, and studied every design file. They acted on a conviction we share: you can only trust, improve and build on what you are allowed to see.

At first glance, Free Software and Cradle to Cradle Design come from different worlds: one from code, the other from materials and products. Upon closer inspection, though, they turn out to be two expressions of the same hopeful idea. Openness, transparency and cooperation are not obstacles to innovation. They are the very things that make it possible.

C2C Design begins with an optimistic question. It's not about being "less bad", but making products that are genuinely good: kept in biological and technical cycles, made from known and healthy materials, powered by renewable energy, and designed so that every component can become a raw material for the next use. Free Software and Open Hardware rest on four freedoms: to study, share, modify and redistribute improved versions. They are a natural ally of this ambition, and the connection runs deeper than that.

Transparency: You can only keep a material in the loop if you know what it is made of. Open design files and a complete bill of materials do for a technical product what ingredient transparency does for a circular one: they turn a black box into something that can be understood and recovered. The same logic drives the digital product passport.

Circularity: The deeper promise of openness, for us, is circularity. A product documented in the open, with parts that can be named and re-ordered, can be taken apart cleanly at the end of a use phase, so its materials flow on into the next technical cycle instead of becoming waste. Open design helps turn "circular by intention" into "circular in practice".

Building on each other: Free Software thrives because everyone builds on shared, visible work. Open Hardware carries the same spirit: once a design is published, a community can pick it up, improve it, and imagine new uses its makers never had to anticipate on their own.

This, really, is the heart of it. Genuine C2C Design, and innovation that creates value for as many people as possible, rarely happens alone. It grows between teams, between a company and its suppliers, between business and research, and between people. Even in a competitive world, the most future-fit results often come from working in the open and building on each other.

This case study is, on its surface, the story of a speaker. We invite you to read it as something larger: a hands-on example of what is possible when good design meets openness, and when people choose to build something together.

That, to us, is where circular design and free software meet. And the possibilities, as this analysis shows, are already in our hands.

Nora Sophie Griefahn and Tim Janßen, Co-Founders and CEOs of Cradle to Cradle NGO

Open Hardware and Free Software: Teufel Mynd – a case study

FSFE's volunteer and open hardware passionate, Nicole Faeber read about the Teufel Mynd speaker announced as an "Open Source Project", so she decided to put it to the test. We, the FSFE, bought one and she dug in. Curious what she found? Read along with us!

Open hardware and Free Software are about more than tinkering: they give users the freedom to inspect, modify, and truly understand the technology they rely on every day. Open hardware means the design of a device is publicly available: anyone can study how it's built, improve it, or manufacture it themselves. Free Software goes hand in hand with this: it guarantees that the code running on your devices can be use, study, share, and improve by anyone. Together, they ensure that whoever owns a device gets to decide what runs on it, free from manufacturer lock-in or hidden restrictions.

This article has not been sponsored by Teufel in any way. We bought the device to test from our own money and did all research on our own.

In May 2025 the German audio hardware maker company Teufel announced a new product, the Mynd Bluetooth speaker box. The speaker in itself is yet another Bluetooth speaker box, the remarkable feature of it is that it was announced as an "Open Source Project" with "high repairability and sustainability", with all mechanical and electronics design files as well as the firmware source code published as "open source". So they mean the Mynd to be a Free Software and open hardware product.

So we wanted to put this to a test!

TL;DR - A Quick Summary

The Teufel Mynd project is a pretty impressive example of an Open Source Hardware project. All the mechanical and electronics design files are made public under a permissive license granting the four freedoms - to study, to share, to modify and to distribute modified versions of it. The data is complete enough to recreate the whole hardware, the mechanical enclosure as well as the electronics PCBs.

There are only two caveats:

1. The loudspeakers themselves are, at least very likely, Teufel proprietary and Teufel does not sell them as spare parts.

2. The firmware for the Bluetooth module along with some DSP and USB type-C controller firmware are not Free Software. But at least the files are provided as binary blobs so one can build a complete firmware update file.

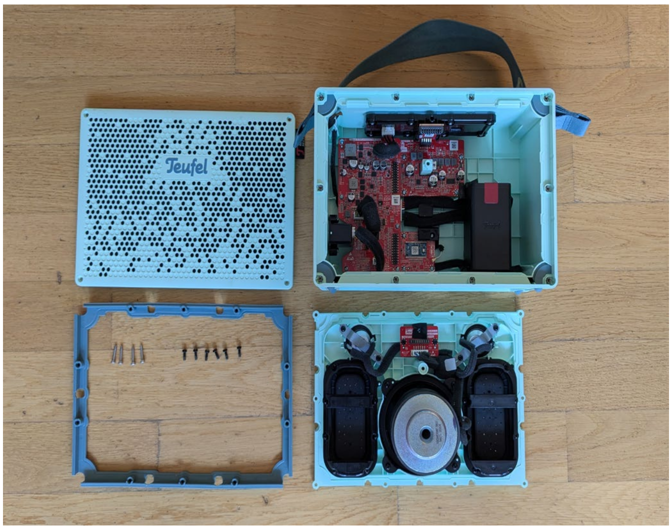
So the project fulfills the Open Source Hardware Definition.

Let's dive into the details!

Repairability

Many current consumer electronic devices these days are not repair friendly. Cost pressure and ease of mass production make the non-repairable options, like glue instead of screws, very appealing to manufacturers.

The Mynd is very different in this regard. The case is held together by screws, not glue. The parts are pieced together tightly with proper rubber seals, not glue strips. The screws are clearly visible and marked, not hidden away under some sticker label or warranty seal. The different screw sizes and types are documented on the bottom of device itself. Also the starting point to pry the casing parts apart after taking out the screws is marked on the case. So opening the device is really made as easy as possible, no guessing, the steps for dis- and reassembly are clear and pretty much self explanatory.



Mynd opened

There is just one caveat, availability of spare parts. So far Teufel only offers the battery pack and carrying belts as spare parts, not the Printed Circuit Boards (PCBs) or speaker chassis themselves. Maybe it is possible to convince the Teufel service center to buy these, but Teufel's general policy for spare parts says: "Please note: We do not ship circuit diagrams, toner, or electronic components such as circuit boards, capacitors, and resistors". For the average customer, this limits repairability.

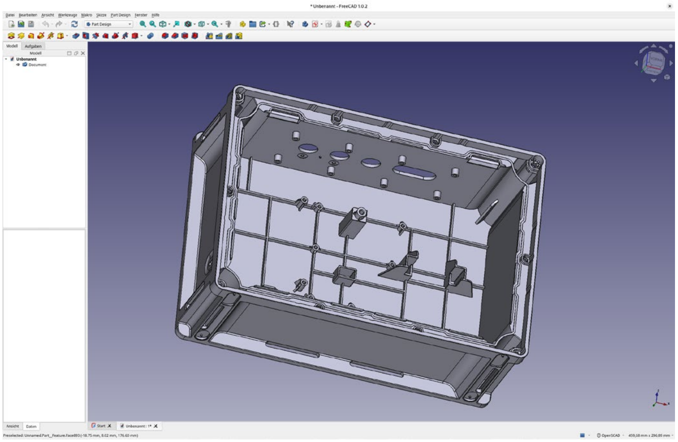
Open Hardware

As promised in the product announcement, Teufel made all hardware design files publicly available on GitHub. All these files are, where applicable, licensed under the Creative Commons "Attribution-ShareAlike 4.0 International" license which allows the four freedoms: Use, study and modify, redistribute the original and modifications.

This is pretty remarkable!

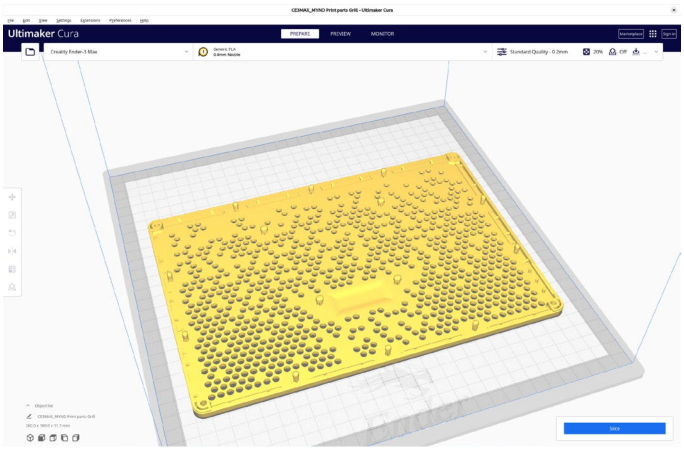
3D CAD, the casing

All components of the case are split into separate 3D CAD files and are available in 3DM, STL and STP format. In total there are 11 different parts that make up the case / chassis. Here we have the main case STP file opened in FreeCAD:

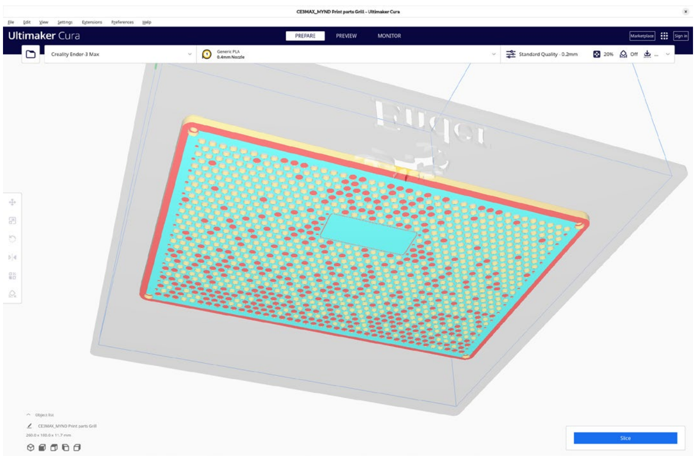


Teufel Mynd speaker opened and dismantled

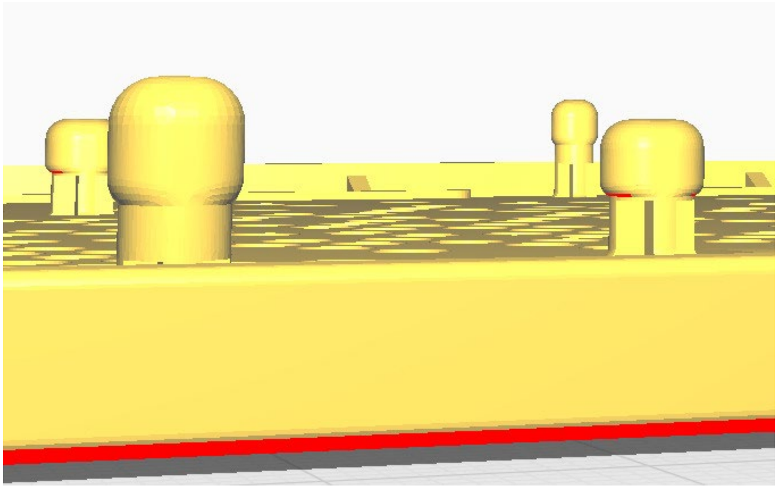
So yes, everything is there, the full case and all its parts! Can you remake it, like 3D print? Well, let's see, here is the front grill STL file opened in the Cura 3D printer slicer:



This is looking pretty good! Since the Mynd speaker box is quite big you of course need a large 3D print room, the front grill is 26cmx18cm. But the problem is in the details. The 3D CAD design has not been optimised for 3D printing but rather for injection mold production so this makes 3D printing difficult and we can see the issue when looking at the print plate from underneath:



All the red parts are free floating, i.e. these do not touch the print surface and will create issues while printing. With some luck and a good printer these will still come out OK but they may also end up quite ugly. Another a bit problematic part are the "knobs" on the top side which hold the front grill tight in rubber gaskets of the speaker fixture:



These little knobs also have some overhangs which can create issues while printing.

So is the CAD design indeed "open hardware"? Yes! It is open licensed and you can use Free Software tools like FreeCAD or Cura slicer to tinker with it, even try to reproduce parts e.g. using 3D printing. The 3D printing limitations (overhangs etc.) are not a detriment to the open hardware nature, nothing requires open hardware to be easily reproducible.

Electronics, schematics and PCBs

Next let's have a look at the electronics design, also for this all relevant files are available in the same hardware GitHub repository under the same "Attribution-ShareAlike 4.0 International" license. Again the files are made available in three file formats: Altium Designer, KiCAD and PDF. The PDF files are PDF exports of the schematics and the printed circuit board (PCB) layout. The Altium Designer and KiCAD formats are the much more interesting ones since these contain the editable schematics and PCB layout! Since we only use Free Software we cannot use the Altium files, so we focus on the KiCAD version for now - I assume that the original design was made using Altium and was then exported to KiCAD.

Excursion: Electronics making

To design and finally make a piece of electronics, a number of pieces are needed. First you need to have some idea what you are going to make, of course. Based on this you need to start choosing components for your design - which chips to use, which power supply these need, how this needs to be designed with which parts etc. So from a very early stage the component selection is a critical first step, we'll also come back to this a little later.

Next you need to start to draft your electronics design, i.e. how all the components are connected to each other. This is done in the so called schematics. Every single connection of each of the components is made here. The schematics usually does not look similar to the final PCB, it is a rather abstracted view on the hardware and focuses on the parts choice and the connection between the parts.

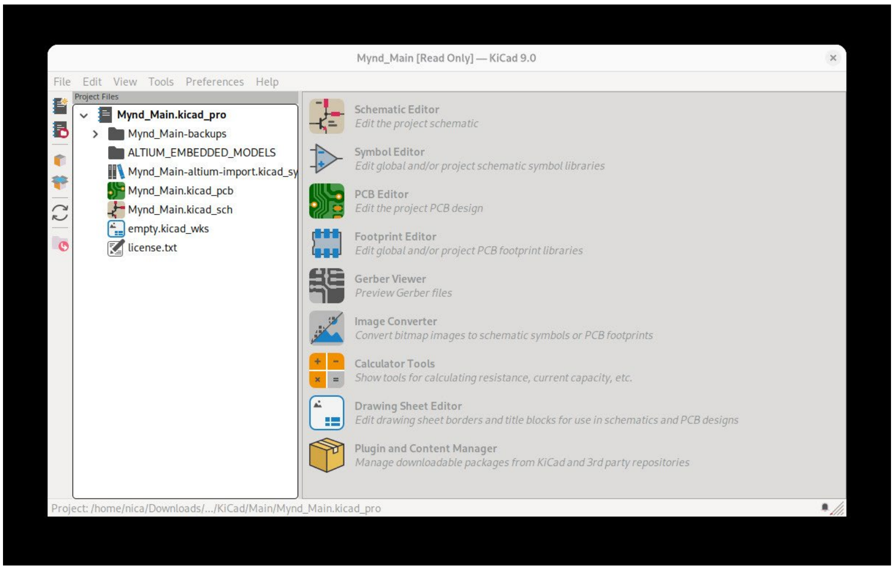
Once the schematics is somewhat ready the next phase is a real time sink, creating a PCB layout from the schematics. Tools like Altium or KiCAD assist with it, they know the physical sizes and constraints of the components and visualize it for you. But in the end the engineer has to properly place them on the virtual PCB. Placing is not only a matter of just putting them somewhere but you also need to know electrical constraints and keep routing the connections in mind. Creating a PCB from known schematics can take days and weeks and needs a lot of experience. Adding to that there may be mechanical constraints as well, e.g. known maximum PCB sizes or shapes that need to be taken into account.

Once the schematics and PCB layout is done the PCB can be ordered from a PCB manufacturer and all electronic components can be ordered from the various distributors. In many cases you need to order from more than one distributor because not even the large ones carry all components - and fingers crossed everything is in stock!

Once PCBs arrive and all parts are at hand you can finally start to assemble the PCB and power it up for the first time: Good luck the "magic smoke" stays within the parts!

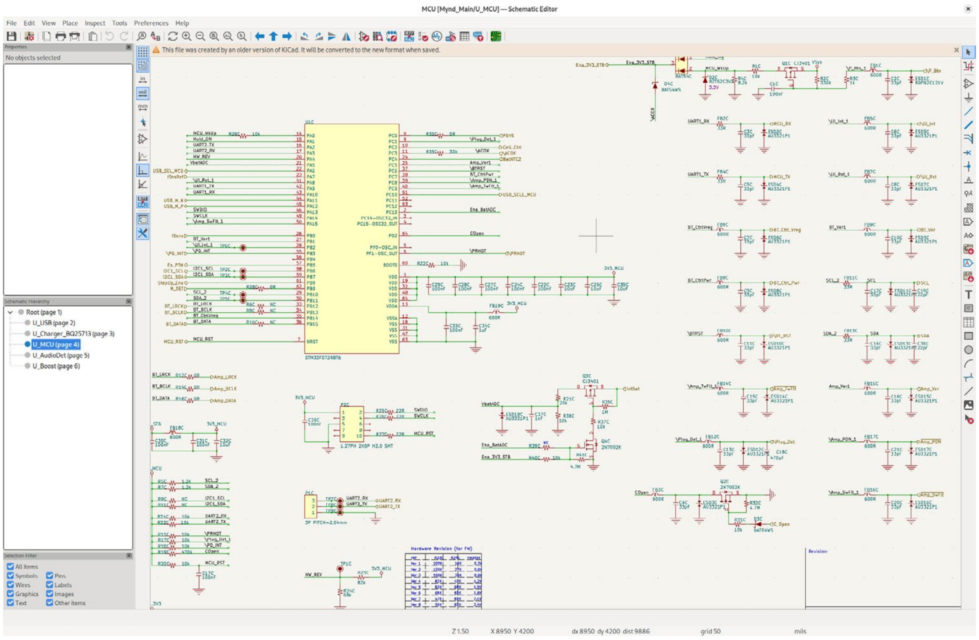
The main board

So let's have a look at the main PCB as an example:

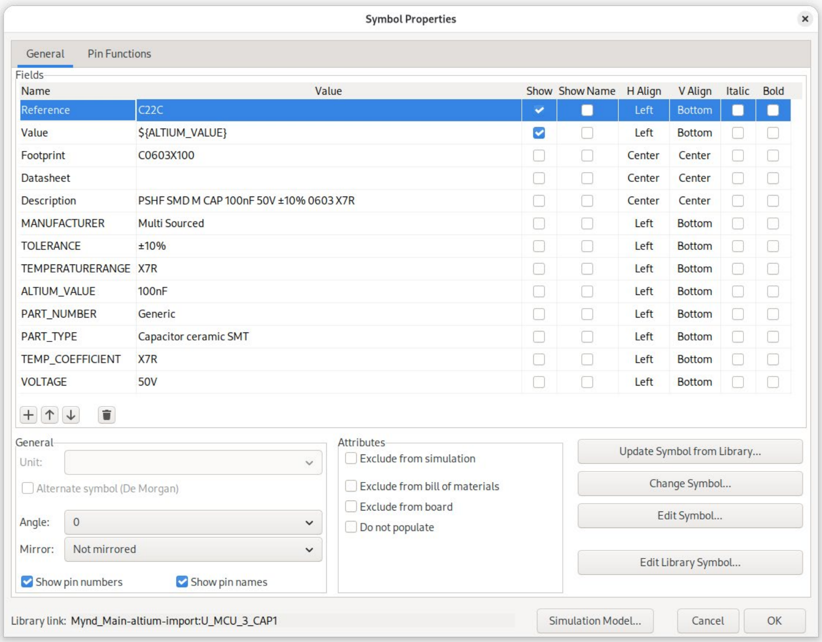


The Teufel engineers created a full KiCAD project for each PCB that can be directly and completely opened in KiCAD with all parts - schematics, footprints, 3D components and the full PCB design. Here is one page of the main PCB schematics for the main microcontroller:

Schematics



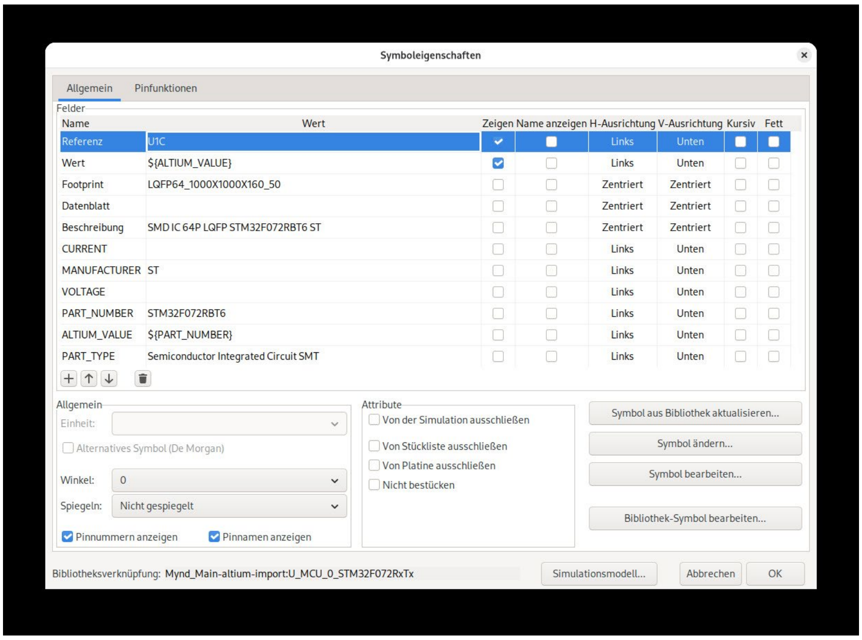
The schematics only show a limited level of detail, some details necessary for actually making a device are not shown. For example the component with designator "C22C" right of the microcontroller unit (MCU), in the schematics you can only see that it is a capacitor with 100nF capacity. But the actual specification needed contains a lot more information, which is also contained in the KiCAD project:



Example component C2C

So here you can see that it is rated for maximum voltage of 50V, has a temperature coefficient of X7R, a tolerance of +/-10% and that the physical so called "footprint" is 0603, which means a surface mount device (SMD) 6mil long and 3mil wide - 'mil' is 100th of an inch, a common size in PCB design, so about 1.6mm in length and 0.8mm wide. A pretty small component. Each and every component needs to be specified this way.

For a capacitor one usually can find components with the same specification from different manufacturers which makes component sourcing a bit easier. Some distributors allow importing a Bill Of Material (BOM) into their shop system and will offer best matches. For some other components there can be a variety of choices with very similar specifications which can make picking the right one complicated. In this case an exact manufacturer part number comes to the rescue:



Example MCU

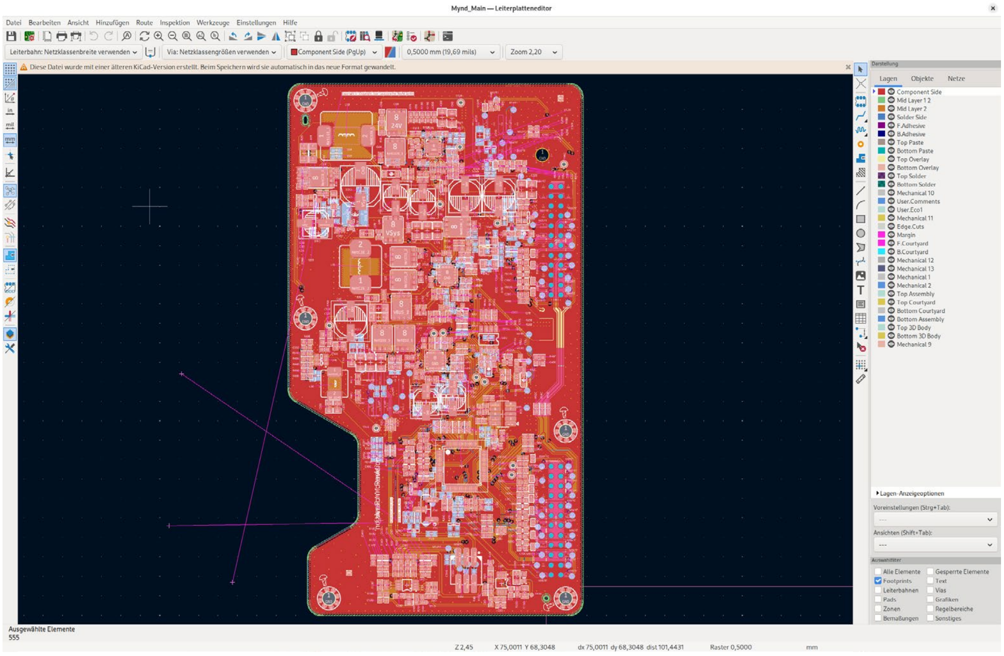
In this example there is the main PCBs microntroller unit, specified with its exact manufacturer and the manufacturer's part number, ready for ordering.

Speaking of ordering, it is also quite common, especially for commercial projects, to also specify an ordering source within the project, so that later on a BOM can be exported and handed over to e.g. a purchasing person or department for ordering all the components. In the KiCAD Mynd project there are no ordering sources.

So, are the electronics design schematics open hardware? Yes! It is licensed under an open license providing the four freedoms. All necessary information to understand the schematics is included as well as all information to make a PCB from the schematics.

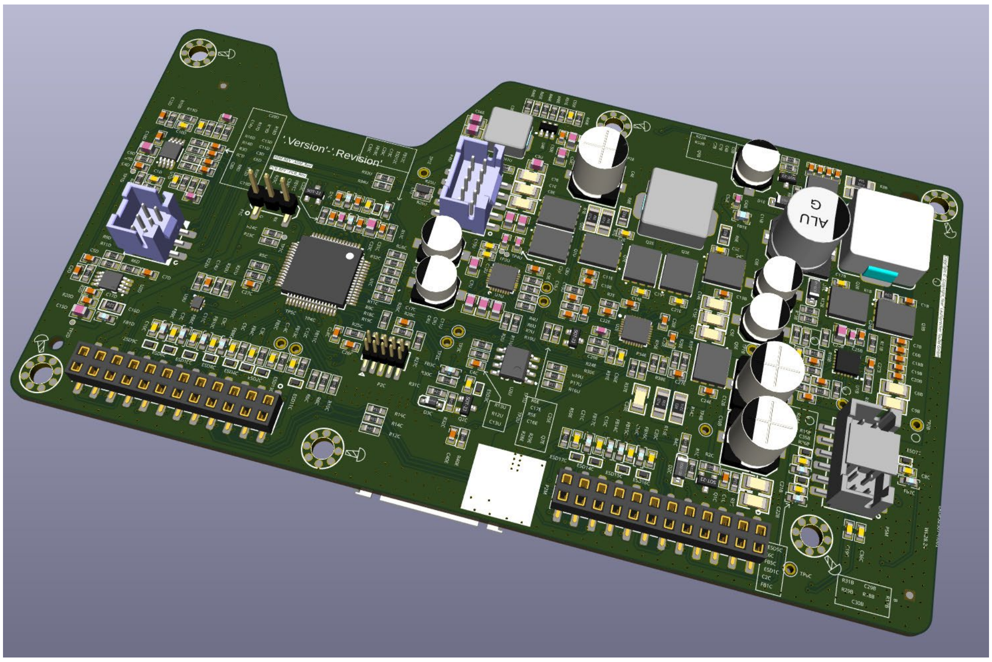
PCB layout

And now to the real fun part, the PCB layout. I won't go into the details here because there would be too much to cover. Just to give an idea here is the main PCB view in KiCAD's PCB editor:

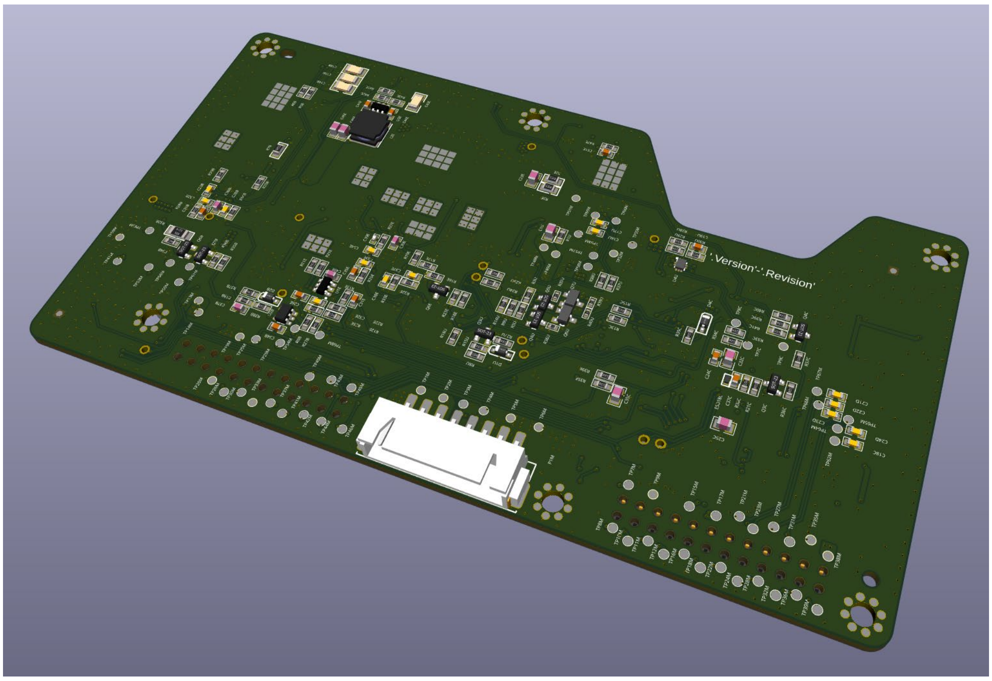


Main PCB layout, KiCAD

This is a 4-layer PCB, i.e. it has four layers of copper traces, top, bottom and two in between. This PCB alone carries 561 components! Many of them small SMD components, the smallest being some 0402 resistors (4mil x 2mil = ~1mm x ~0.5mm). And to make things even worse, there are components placed on both sides of the PCB which makes soldering even more complicated. In KiCAD there is an awesome feature to create an interactive 3D view of the designed PCB, here is the Mynd Main PCB front and back:



KiCAD 3D Mynd PCB front



KiCAD 3D Mynd PCB back

In total there are nine (9) PCBs in the Mynd speaker box, the main and amplifier PCBs being the largest and most complex. In total there are 1047 components on these 9 PCBs, about 87 different types of components. Just managing the supply and stock of 87 different component types is, I can tell from experience, a major effort.

Is the PCB layout open hardware? Yes! You have the four freedoms and the hardware design details are complete enough to rebuild the whole thing! From parts specification to PCB specification (so called "stack up") everything is there. The only issue one might run into trouble with is finding ordering sources for all the components, but that's probably outside of the scope of open hardware anyway.

Make it yourself?

Well, I have not tried, but looking at the design files, mechanical and electronics, I think building the Mynd speaker hardware from scratch should be possible. I have not found any information missing, except maybe ordering sources for the components. Placing and soldering all over 1000 components on the nine PCBs will be a major effort and with components down to 0402 will be difficult to place by hand, some pick & place setup would be very helpful along with proper SMD reflow soldering equipment. But the important part here is that there is no information missing in the design files, it's all there.

3D printing the case could be possible as well, with some additional effort to accommodate for 3D printing instead of injection molding. The result will very likely look and feel differently, but it could work.

What will definitely become a problem are the Teufel specific custom parts, i.e. the speaker chassis themselves. Other parts like the rubber gaskets and rubber seals these are also in the 3D design files but will be hard to e.g. 3D print properly.

The firmware

And now to the final part, there is also some software involved! The main MCU, a STM32 microcontroller, runs some software as well as is the module used for the Bluetooth connectivity and sound generation. The source code and build scripts were also made available by Teufel. The parts created by Teufel, i.e. the application software running on that STM32 MCU, are licensed under the MIT license. But for building it, some more dependencies are needed, like some software development kit (SDK) bits from ST-Micro (Apache 2.0, BSD 3-clause, MIT, proprietary ST SLA), bits and pieces of FreeRTOS (MIT) and some more proprietary binary only parts for the Bluetooth module, the audio DSP and, last but not least, the USB type-C port controller - yes, also these things have their own firmware these days.

So apart from the blobs everything is Free Software, which is pretty cool! The blobs though are a bit of a bummer. But is this Teufel's fault? Or does this make the whole project less "open hardware"?

My personal take on such blobs is this: While I dislike proprietary blobs in general, because they limit our freedoms to use the hardware as we like, I can at least accept certain blobs, e.g. in the case where chip+blob form a kind of union so that I can basically view the two together as just a single piece of hardware. I still don't like it, because it is hidden software, but at least I can somewhat accept it. We have this kind of situation in many cases and are somewhat OK with it - the WiFi firmware in our WiFi cards, the firmware in the touchpad controller of our laptops etc. The list is long, very long. How to properly address and fix this is another story we need to come back to another time.

The firmware package can be built using the free GNU GCC cross compiler for ARM and build the firmware for the STM32 MCU. The firmware blobs for the DSP, Bluetooth module and the USB type-C controller do not get built and it is also not documented how to program these. But at least the binaries are there.

So is the firmware Free Software? Concerning the MCU firmware, yes, it is. All firmware? No, there are certain blobs.

Especially sad is that the Bluetooth part is completely binary only in the Bluetooth firmware blob, so all pairing, Bluetooth audio protocol (A2DP with SBC) etc. is hidden in that blob and can not be changed - I would have been curious if SBC-XQ (higher quality audio codec) or TWS (True Wireless Stereo) could be added to that Bluetooth firmware.

Final verdict

So, is it open hardware?

Mechanical design / parts: Yes! Everything is documented properly, some special components are not available off the shelf and need alternative sources, especially the speakers. It would be great if Teufel would also supply these through their spare parts shop.

Electronics design: Yes! The KiCAD files contain everything one needs to completely recreate and especially repair the electronics. Some components may be difficult to source though (like the Bluetooth module).

Firmware: The Teufel made parts are Free Software, some third party firmware parts are not (USB type-C controller, DSP, Bluetooth module).

With all that it also qualifies nicely for the Open Source Hardware Definition [4] by the Open Source Hardware Association (OSHWA).

Overall I think the Teufel engineers did an extremely good job here! They definitely set the bar and have given a great example of how one can make open and sustainable consumer hardware. For some first hand insights there is a nice interview with the engineers published on the Prototype Fund website (in German).

Will the openness detriment their business? I am fairly sure it won't. Yes, maybe competitors can learn a bit about high quality engineering, but on the other hand this isn't rocket science either. Will they loose business because people will now rather build it themselves instead of buying a ready made device from Teufel? I doubt that. While I am quite convinced that one could build a complete Mynd device from the open specifications, it would take many many work hours - the logistics for the over 1000 electronic components, populating the PCBs, 3D printing or otherwise making the enclosure etc. - this is a serious undertaking and will take weeks along with dozens of person hours. A good hacking experience, for sure! But economically definitely not worth it, better go and support the company that made it with a purchase.

Outlook

Open Hardware not only means better repairability and with it better sustainability. It also means possible enhancements too! For example the Bluetooth module in the factory Mynd only supports the rather mediocre SBC audio codec. But since the electronics are made modular and the rest of the system is documented so well, why not think about replacing this limiting Bluetooth module by something else? By something more capable?

Why not replace it with an ESP32 based module plus high quality audio codec chip? Would not be the first ESP32 based audio device.

Why stop at an ESP32? There is a discussion going on in the GitHub Mynd hardware repository about replacing the proprietary Bluetooth module with a Raspberry Pi Zero 2 W based module! This would not only enable other Bluetooth audio codecs but then you would also get a complete Linux system into your Mynd, with its own media playback capabilities! Prototypes already seem to exist.

This is the brilliance of Free Software and open hardware, the limit is your imagination, it is in your hands.

Happy hacking!

Do you care about sustainability, repairability and Free Software? Did you do research about which products work best with Free Software? Have you ever tried to tinker with them? We would love to hear your experiences. Please, share your knowledge with us. We want to know which products work better when you want to be in control of your devices!

Moreover, we are looking for Free Software enthusiasts who like to check if their devices are really open hardware. If you are one of those, contact us (contact@fsfe.org), so we can write reviews together to help user to choose products that respect our four freedoms!

Read the FSFE's publication online here.