Best Practices of Open Source Mechanical Hardware

A guide with practical advice for sharing product-related documentation

Version 1.0 / February 2017

Jérémy Bonvoisin, Kerstin Carola Schmidt
Technische Universität Berlin

What makes a software program "open source" is that its source code is publicly available. What is pretty clear for software is less clear for hardware—especially for mechanical hardware. What does the term "source" refer to in "open source hardware"? According to the Open Source Hardware Association (OSHWA): "*Open source hardware is hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design.*"[i] What does it mean to "make a design publicly available"? What are the documents that enable others to study, distribute, make or sell a design? This short guide provides practical answers to these questions and is dedicated to those who want to make a piece of mechanical hardware open source.

Preamble

The above cited definition implies four degrees of freedoms regarding a considered piece of hardware:

- Freedom to study: i.e. the right to access enough information to understand how the piece of hardware (referred herein as the product) works and to retrace the logic behind its design;
- Freedom to modify: i.e. the right to edit the product definition documents and to tweak or develop the product further for any purpose;
- Freedom to make: i.e. the right to use the product definition documents to manufacture the piece of hardware;
- Freedom to distribute: i.e. the right to share or sell the product definition documents as well as the physical products fabricated according to these documents.

A precondition for the realization of these degrees of freedom is that hardware must be released with its "source code", which is termed in the referred definition as "documentation" or "design files". However, what information has to be shared in order to allow any interested person to study, modify, make and distribute a piece of mechanical hardware—in other words, what is the source of open source mechanical hardware—is a question that is not trivial and remains unanswered in this definition.

While sharing a CAD file would allow anyone to study the product design, this may be insufficient information to manufacture it. While sharing fully defined and detailed assembly instructions would allow anyone with sufficient production means and knowhow to manufacture the product, this may be insufficient to allow them to participate in the further design of the product. Is a product open source as soon as enough documentation is provided to ensure one of these freedoms? Or is it mandatory to provide enough documentation so that all of these freedoms are ensured? Is it enough to publish CAD files or is it mandatory to equally share assembly instructions?

In the absence of clear answers to these questions, this guide strives to provide practical advice on information you may want to share with your community in order to make your piece of mechanical hardware open source. It shows what you *can* do, but does not pretend telling what you *must* do. It is based on the practical assumption that the type of information you may want to share depends on your motivation to go open source:

- Is your motivation to demonstrate your customers how good your product is designed? Then put particular emphasis on publishing detailed and commented design files. See section "enable others to study your product".
- Do you want your innovative product to be manufactured by others so it can be rapidly adopted worldwide? Then put particular emphasis on publishing clear assembly instructions and a parts list. See section "enable others to make your product".
- Do you want to use open source as a means to improve your product by receiving feedback from others to or by even letting them take part in your product development project? Then put particular emphasis on publishing documents others can edit and tell potential contributors what you expect from them. See section "enable others to contribute to your design".

Note that the maturity of your product also influences the content and the level of detail of your product documentation. If you are still at an early design phase, you may have only some description of the product concept and requirements in the form of sketches and text. If your product is fully defined, you may have complete CAD drawings and a stable parts list. But independent from the maturity of your design, the general rule of thumb is to share the information you have at the time you have it.

In the following, the possible components of making your product open source are presented.

This document is to be considered as complementary to the best practices of open source hardware published by the Open Source Hardware Association[ii].

Enable others to study your product

If your motivation for going open source is to allow others to study your product, then you may want to publish your design files, i.e. the files you created to define your product. Design files are for example 2D drawings or 3D models, circuit board layouts, schematics or additional technical drawings

that show the structure of your product in detail. Sharing these files will allow others to understand how your product is designed. In the best case, you will share your design files in their **original format**, that is, in the format in which you created them. Even better is to use free and open-source software applications and open file formats so that others do not have to buy expensive licenses to access the information you are sharing with them.

Alternatively, you can share your design files in **export file formats** such as STL, PDF or PNG. Exporting your original files in those formats will however inevitably lead to loss of information. Therefore, sharing these files shall only be considered as additional measure, for example in cases you are using costly software that others may not be able to afford.

In any case you may:

- share these files in a design repository to allow others to browse your files online (e.g. GitHub);
- give the possibility to download packaged files with one click;
- clearly separate original design files and exports by labelling them accordingly.

Enable others to replicate your product

If you want others to replicate your product, you may publish a parts list and assembly instructions. Below, you will find some recommendations regarding these two elements.

Share a parts list

A well-made and clear parts list (also termed as bill of materials [BOM] in engineering jargon) is an essential document for those who want to replicate your product. This document will help them understand which parts they have to source or manufacture before assembling your product. The ISO 7573:2008 standard provides a list of "minimum requirements for parts lists" and may help you with setting up your BOM properly. Unfortunately this document is not open source and not everybody can access it freely. Here is some advice about how to structure your BOM.

A BOM may contain the following information for each part/component of your product:

- **Name of the part**: a short denomination allowing to quickly identifying its function.
- **Reference designation**: This is a unique reference code linking the parts list and your technical documentation. With this code, the maker will know which component refers to which CAD file. Reference designation and name of the part may eventually be the same, as long as the reference designation is unique and unambiguous.
- **Photo or illustration**: Putting a photo next to the textual reference of each part will help getting a quick overview of your parts list. This will allow someone to search more efficiently for parts in the parts list.
- **Exact specifications**: these are precise descriptions of the characteristics the part will have to fulfil, including the metrics and the units of measure, preferably in SI units. Avoid vague descriptions such as "you will need 20 small bolts, 10 large ones, a good battery and a long wire". This information may be insufficient to find the appropriate components and to reliably reproduce the functionality of your product. Here some examples of unambiguous specifications you can give:

- o Bolts: diameter, length, kind of thread. For example, "M3x20" labels a metric screw thread with 3 millimetres diameter and 20 millimetres length.
  - o Nuts: standard, diameter, height. As an example, an ISO 4032:1999 M4 nut would be a hexagon nut with 3.2 mm height and 7 mm diameter.
  - o Battery: Type, voltage, typical capacity in mAh, technology. For example: LR6, 1.5 V, 2 850 mAh, alkaline.
- **Quantity**: that is, how many of such parts are necessary. You may also indicate whether some extra parts should be bought as a precautionary measure, in case the assembly of your product goes wrong.
- **Procurement information**: that is, where the parts can be ordered. A list with possible suppliers will help makers of your product to save time. In the best case, you can recommend a supplier by providing a direct URL hyperlink to the part on the supplier's website. Stating an alternative procurement option can make sense as well.
- **Cost**: it may be of interest to know the cost of each part in order to calculate the total procurement costs of the product. You can either provide estimates or alternatively make references to market prices and provide URL hyperlinks to one or more suppliers.

Here are a few best practice examples you can look up: Sunzilla, Lasersaur, μdelta.

Share assembly instructions

Once makers know which parts they need and how to procure them, they may require guidance to assemble your product. The international standard IEC 82079-1:2012 provides some guidance in that regard. It defines the creation, structuring and presentation of technical manuals in terms of comprehensibility, content, structure, appearance, design and language. Yet again, this document is not open source and not everybody can access to it. Here is some advice about the two main components of clear and understandable assembly instructions: text and illustrations.

Text

A good instructional text finds the balance between richness of details and clarity. It should be detailed enough and mistake-proof, but not filled with unnecessary information that could lead to confusion. Concise language is essential if you want to reach a wide audience of readers, so you may want to minimize unnecessary technical jargon.

Illustrations

Illustrating assembly instructions makes them more clear and vivid. There are at least two kinds of illustrations you can provide:

- *Images* provide an easy way of illustrating assembly steps. They help makers to quickly identify what is to be done and allow them to proceed per imitation. Images can either be photos of assembly steps or drawings such as CAD models. Even better is to annotate images with text and symbols. For example, an arrow can depict the path along which a part has to be moved in an assembly step.
- *Videos or animated GIFs* may be more powerful in their significance than photos, since they carry more information. In cases of very complex assembly steps, videos or GIFs might better express instructions that are difficult to formalize with text and static images.

Here are a few best practice examples you can look up: [Ultimaker,](#) [Precious Plastic.](#)

Enable others to contribute to your design

If you want to profit from collective intelligence to improve your product and build a community that actively participates in your product development project, there are at least two things you may consider: sharing editable information and publishing a contributing guide.

Share editable information

Publishing non-editable files represents a barrier to potential contributions. Sharing export CAD formats like PDF or STL may enable your community to study your design. Yet, those formats are hardly editable and they may not bear all information you had in your original CAD format (e.g. parameters and constraints). In contrast, sharing CAD files in their original processing format would allow others to access the entire information, to modify it and to make modifications.

This is not only true for CAD drawings, but also for other kinds of information and files such as product specifications, user manuals, BOMs or assembly instructions.

There are at least two ways you can share editable information:

- through editable files your community members can download/upload.
- through a Web 2.0 environment allowing online edition (e.g. a wiki).

Publish a contribution guide

If you want to build a community of co-designers, you may communicate publicly (i.e. make an open call) that interested persons are highly welcome to join and contribute in the product development. Publishing a contribution guide may be a good idea for this. Below are some advices for creating a motivational contribution guide.

State explicitly that people are welcomed and explain them *what* they can do. For example:

- take on already identified tasks/issues;
- define new tasks/issues, for example by giving improvement suggestions or reporting bugs;
- contribute to the documentation regarding product assembly or usage;
- help expressing product requirements;
- leave comments;
- support your project financially.

After letting potential contributors know *what* they can do, you may want to explain them *how* they can contribute. For example by providing them practical information with regard to the following:

- Collaborative design tools:
  - Where can the contributors find the files they need?
  - Which tools should they use?
  - How can they register?
  - How can they let others know what part of the project they are working on?

- The design process:
  - How can contributors make or suggest modifications?
  - How are contributions evaluated and integrated?
  - What rules and guidelines should be followed when contributing to the project?
- The design rationale:
  - Why is your product the way it is?
  - Why you made the specific choices you did.
- The team
  - Who is involved in the project and how?
  - Which skills are currently needed in your project?

Finally, you may want to explain potential contributors *why* they should contribute, that is, what can motivate them spending time and expertise in the project. Here are some elements that can help you with this:

- Tell them what your vision is. It might be helpful to make it clear where you are heading with your project and what you want to change.
- People may be motivated by fun, the desire to learn new skills or to create a better product, social aspects, networking, career development and even competition or rewards. You could make use of this and tell your potential contributors the benefits they can expect from contributing. For example:
  - You could explain the fields in which they can improve their skills and gain additional knowledge.
  - You could emphasize the fun factor of contributing to your project by explaining/showing the fun that you are having and that they can expect.
  - A sense of community motivates people participating in a co-creation environment. You might want to emphasize the aspects of community building and the collective spirit in your project.

General advice

Showcasing helps. You may share information helping others to understand your product and its functions. You may show renderings or previews of your product (an image of a 3D model or a photo of one of your prototypes) and concisely explain what it is about and which requirements it will meet.

Be clear about the products requirements. It may be interesting for others to understand not only the solution you are providing but also the problem you are solving. By doing so, you may explicitly state the initial problem or motivation behind your project so people understand clearly what you are trying to achieve.

Make information and documents easy to find. An intuitively and easy-to-use designed homepage helps a lot. The necessity to fill a form in order to get the documentation may be interpreted as a violation of the principle of non-discrimination against persons or groups contained in the open source definition.

Checklist: how open is your hardware?

The Open Source Hardware Association offers a certification programme allowing developers to self-certify the compliance of their product with the open source hardware definition[iii]. The clarity of this certification scheme is however restrained by the limited precision of the open source hardware definition highlighted in the introduction of the present document. It also considers openness as a binary criterion: either a product is open and can be certified (in accordance with compliance to defined threshold standards), or it is not. However, the different forms and levels of openness are not taken into account.

In order to help you determine how open your product is, we introduce the concept of the open-o-meter: a simple scale from 0 to 8 where a product gets one point for each of the following aspects:

1. design files are published
2. assembly instructions are published
3. a BOM is published
4. a contribution guide is published
5. the published CAD files are in editable format
6. the published assembly instructions are in editable format
7. the published BOM is in editable format
8. the specified license allows commercial use

If your product gets 8 points, it conforms to the best practices of open source hardware. If your product gets 0 points, it does not seem to be open at all and should not be labelled as open source. If your product gets between 1 and 7, well, continue your efforts!

Acknowledgments

---

[i] http://www.oshwa.org/definition/
[ii] http://www.oshwa.org/sharing-best-practices/
[iii] http://certificate.oshwa.org/