

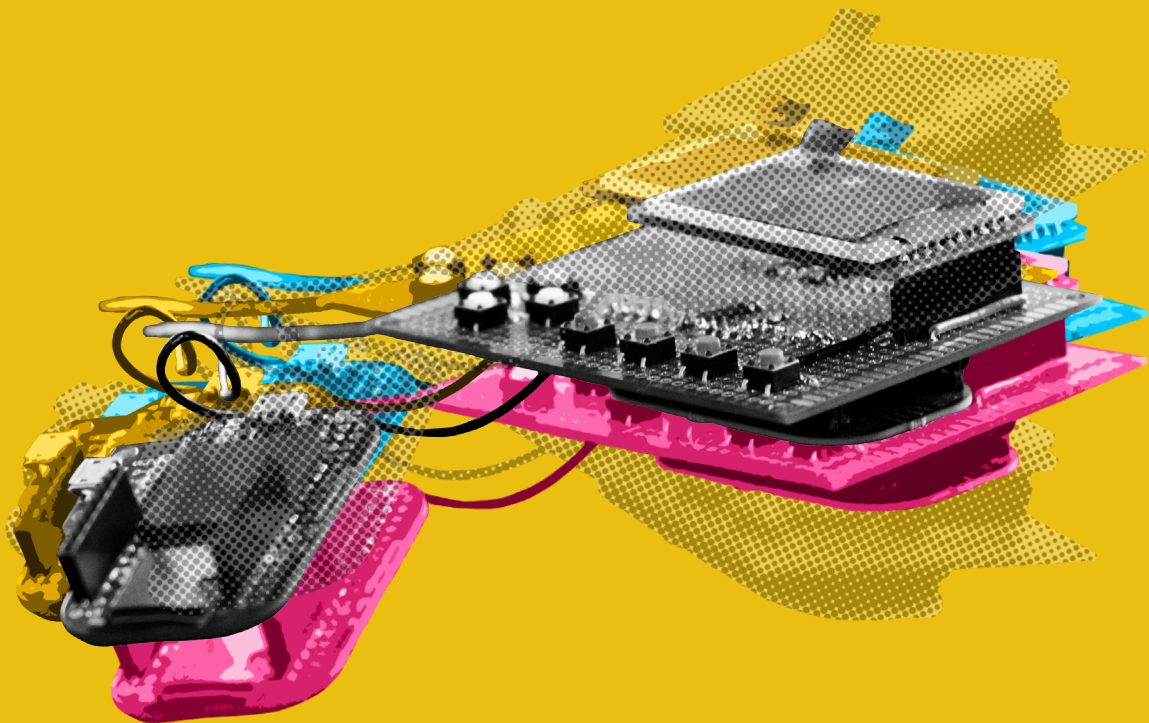


**ENGELBERG CENTER**  
on Innovation Law & Policy  
NYU School of Law

**OCTOBER 2020**

# Open Source Hardware Weather Report 2020

Clarissa Redwine and Michael Weinberg



**OSHWA**  
OPEN SOURCE HARDWARE ASSOCIATION



## Engelberg Center on Innovation Law & Policy

The Engelberg Center on Innovation Law & Policy at New York University School of Law provides a unique environment where scholars can examine the key drivers of innovation as well as the law and policy that best support innovation. By fostering interdisciplinary and collaborative research on innovation law and policy, the Engelberg Center attracts legal scholars and practitioners, technologists, economists, social scientists, physical scientists, historians, innovators, and industry experts who study the incentives that motivate innovators, how those incentives vary among creative endeavors, and the laws and policies that help or hinder them.

## Open Source Hardware Association

The Open Source Hardware Association (OSHW) aims to foster technological knowledge and encourage research that is accessible, collaborative, and respects user freedom. OSHWA's primary activities include hosting the annual Open Hardware Summit and maintaining the Open Source Hardware certification, which allows the community to quickly identify and represent hardware that complies with the community definition of open source hardware.

## Acknowledgements

This report is the result of discussions with a wide swath of the open source hardware community. We want to thank everyone who made themselves available to discuss their experiences in open source hardware and support this report, including the following people who agreed to be acknowledged below. While their input helped shape this report, their inclusion in this list is not necessarily intended to imply endorsement.

Kwabena Agyeman	Marcin Jakubowski
Mitch Altman	Toni Klopfenstein
Leon Anavi	Sophi Kravitz
Eric Carlson	Tod E. Kurt
Charles Duan	Catarina Mota
Lenore Edman	Jie Qi
Salman Faris	Elliott Rouse
Zach Fredin	Shah Selbe
Drew Fustini	Jan Suhr
Alicia Gibb	Phillip Torrone
Josh Greenberg	Jeffrey Yoo Warren
Santana Jackson	Juliana Zhang

Layout and design by Objectively



# Contents

<b>Introduction</b>	<b>4</b>
A Note on Terminology	6
<b>A Brief Background on Open Source Hardware</b>	<b>8</b>
<b>Open Source Hardware as a Tool</b>	<b>10</b>
Open Source Hardware Has Real Benefits	10
Open Source Hardware Requires Real Investment	11
<b>Hardware is Not Software</b>	<b>13</b>
<b>When Does it Make Sense to Use Open Source Hardware?</b>	<b>16</b>
When Using it to Set a Standard	16
When Encouraging Hackers	17
When Designing to Teach or Train	17
When Targeting a Non-Commodity Market	17
When You Want to Turn Users into Advocates	18
When You Have Trusted Documentation and Manufacturing Quality That Set You Apart	18
When Engaging Your Community is at the Center of the Project	18
<b>When Does it Not Make Sense to Use Open Source Hardware?</b>	<b>20</b>
When it Targets a Community Primarily Interested in Non-Modifiable or Expandable Projects	20
When Maintaining Public Documentation is in Conflict with Other Goals	20
When Maintaining a Community is in Conflict with Other Goals	21
When Open Source Hardware is Viewed as a Replacement for Marketing	21
<b>There Are Unresolved Challenges in the Open Source Hardware Community</b>	<b>22</b>
Is Open Source Hardware Documentation a Barrier to Opening a Project?	22
Should the Community Distinguish Between Levels of Openness?	23
Is There a Place for Almost, or Aspirationally, Open Hardware?	23
Is it Productive to View Open Hardware as More than a Dichotomy?	24
Is “Just Copying” Wrong?	24
<b>Opportunities for Open Hardware</b>	<b>26</b>
Be More Intentional About Diversifying Open Source Hardware	26
Expand Academic Pathways	27
Openness Requirements in Grantmaking	27
Create Model Repositories	28
Develop Open Components	28
Improve Open Design Tools	28
<b>Conclusion</b>	<b>29</b>
<b>Appendix</b>	<b>30</b>
Top Open Source Hardware Certifications by Country	31
Top Open Source Hardware Certifications by Entity	31

# Introduction

**O**pen source hardware has blossomed in the past 10 years. In the decade since the first set of gatherings that would become the annual Open Hardware Summit, the open source hardware model has proven itself to be much more than an idea. Open source hardware projects form the basis of robust community initiatives, expansive scientific research, and multi-million-dollar businesses. The open source hardware community has developed a rich collection of organizations such as the Open Source Hardware Association (OSHW), the Gathering for Open Science Hardware (GOSH), and the Community Bio Summit. There is no question that open source hardware works as a business, as a community, and as an approach to creating.

One reason for open source hardware's success is that it is a diffuse, dynamic approach to creation driven by a diverse community. Although this diffuse nature is ultimately a strength, its diffuse nature can also make it hard to fully understand the state of the community at any moment in time. This resource—which we are styling as a weather report—is designed to address that challenge by capturing the current moment in open source hardware. Like a weather report, it describes current conditions and attempts to project into the future.

This weather report documents the state of open source hardware at the end of 2020. Existing members of the community can use it to understand what is happening in other corners of the ecosystem, to help organize around universal challenges, and to collectively support opportunities. Newcomers to open source hardware can use the report to understand the current state of open source hardware, and to determine if an open source hardware approach fits well with their goals.

This resource is styled as a report, not a book. It prioritizes identifying, organizing, and briefly introducing key concepts over exploring them in the fullest possible detail. That approach is intended to make it easy to use this resource to survey the state of open source hardware. Diving deeply into any element of open source hardware is beyond its scope.

The open source hardware community is far from monolithic. In fact, not all communities that work within the open source hardware framework identify themselves as doing so. Even self-acknowledged open source hardware communities can come from different traditions, sometimes only slightly aware of one another.

That can mean that some parts of the open source hardware community wrestle with problems solved long ago by others. Similarly, all too often, disparate corners of the community are working on the same problems in parallel without the benefit of sharing resources. Some people can become frustrated when they try to join the community and are unable to find a guide that allows them to learn from what came before them.

This report is intended to begin a process of distilling information from disparate parts of the community into a common reference point. While no report can be fully comprehensive, identifying common successes and challenges can make it easier for the community to coordinate within itself.

This report was originally designed to flow from an in-person workshop following the 2020 Open Hardware Summit. When the summit became virtual as a result of the COVID pandemic, the workshop became a series of one-on-one interviews with many of the original invitees, as well as other members of the community. The issues, questions, and challenges raised during these interviews formed the basis of this report.

This weather report captures a shared moment within the sprawling open source hardware community. It identifies what works in open source hardware today, along with what does not work. Combined, these will help everyone understand when open source hardware can be effective—and when it is a poor match to an initiative's goals.

It also identifies questions and roadblocks that are shared across the community. Hopefully that will make everyone struggling with those challenges feel less alone. It may also help the community coordinate to find solutions. We hope that this report will help identify areas of opportunity for the community to build consensus, tools, and platforms that support open source hardware. We also hope that it will help connect those disparate corners of the community to each other.

Open source hardware is a tool. Like any tool, there are some problems that open source hardware strategies lend themselves to. There are others where open source hardware simply does not make sense. By capturing the state of open source hardware in 2020, we hope that existing community members can learn from their peers, new community members can find their way in, and former community members can find their way back.



## A Note on Terminology

The open source hardware community is broad. It includes multi-million-dollar companies, one-person hobby projects, and many initiatives in between. While this diversity is a strength for the community, it can also create linguistic challenges.

### OPEN HARDWARE VS. OPEN SOURCE HARDWARE

Although the use of “Open Hardware” vs “Open Source Hardware” once reflected two sides of a split within the community,<sup>1</sup> at this point most people use the terms interchangeably. This report follows that current convention and does not assign any relevance to the distinction.

### OPEN HARDWARE PROJECTS

It can be complicated to identify the *thing* at the heart of open source hardware. *Hardware* can be too narrow because it fails to reflect the supporting documentation and code that make up a critical part of openness. Similarly, *product* can suggest a focus on commercial open source hardware at the expense of other parts of the community. While products make up an important, vibrant part of the open source hardware community, a commercial orientation is not required to be part of the open source hardware community.

This report uses *project* to refer to the hardware, software, documentation, and community connected to a given piece of open source hardware. The term encompasses commercial offerings as well as non-commercial offerings. Similarly, the *project team* is the group of individuals who are primarily responsible for the project. The team may be within a company, an entire company, an NGO, an academic research lab, or an informal collection of individuals.

### OPEN HARDWARE, MAKER, AND LOW-COST HARDWARE

This report focuses on open hardware. While open hardware can overlap significantly with maker and low-cost hardware, the terms and concepts are not interchangeable. These terms are also not universally defined.

*Open hardware* is just that—hardware that is intentionally and purposefully open. Open hardware can be created by an enthusiast at home or by a multi-million-dollar manufacturing infrastructure. It is publicly documented and openly licensed, with source files and code available to all. Ideally, open hardware meets the community definition of open source hardware maintained by the Open Source Hardware Association.

*Maker hardware* can be, but is not necessarily, open. It tends to be created using distributed manufacturing tools, and is often designed to be user-buildable and user-serviceable. Maker hardware tends to be oriented away from large industrial supply chains and toward individual or small group creators.

*Low-cost hardware* is defined by its cost. It may be open or closed, DIY or industrial. Its relatively low cost may make it more attractive to work with and deploy in challenging or resource-limited environments because it is inexpensive to replace.

Although each of these types of hardware share connections, it is helpful to recognize their distinctiveness. Unlike “open hardware” and “open source hardware,” they cannot be used interchangeably.

<sup>1</sup> See the “Open Hardware vs. Open Source Hardware” section of OSHWA’s open source hardware history: <https://www.oshwa.org/research/brief-history-of-open-source-hardware-organizations-and-definitions/>

A gathering of the open source  
hardware community at the 2016  
Open Hardware Summit



# A Brief Background on Open Source Hardware

**W**hile this report celebrates the tenth anniversary of the Open Hardware Summit, the open source hardware community has roots that are much deeper. The Open Source Hardware Association (OSHWa)'s [history](#) of open source hardware begins with the work of [Bruce Perens](#) in 1997, and includes development of the [TAPR open source hardware license](#) in 2007 and the OHANDA four freedoms mark in 2009.

2010 was the year of a gathering that would become the annual Open Hardware Summit and the creation of the community-derived [Open Hardware Definition](#). In 2012 OSHWA was formed as a home for the Open Hardware Definition and institutional organizer of the Open Hardware Summit. OSHWA also developed a collection of best practices and guidance documents for creating open source hardware in response to community feedback and questions. These documents became a common reference point for the community as it worked to develop open source hardware.

The Open Hardware Summit quickly grew into an annual focus for the open source hardware community. By 2016, OSHWA had also designated October as Open Hardware Month to make it easy to hold local celebrations of open source hardware beyond the annual summit. These local celebrations included events around the world, featuring community members from an increasingly broad set of locations working within an increasingly broad set of contexts.

2016 also heralded the first Gathering on Open Science Hardware (GOSH) at CERN. GOSH's interests focus specifically on integrating open hardware into the sciences. The first meeting produced the [GOSH manifesto](#), codifying the goal of reducing barriers between diverse creators and users of scientific tools to support the pursuit and growth of knowledge.



As the community expanded, it became increasingly hard to easily identify hardware that complied with the open hardware definition. In response, OSHWA created the open source hardware certification program in 2016. [Version 2 of the program](#), released in 2018, also provided an opportunity to [consolidate existing guidance documents and best practices](#). In a further attempt to help bring standardization to open hardware documentation, in 2019 the Open Know-How Working Group released [Version 1.0 of the Open Know-How Manifest Specification](#).

During the same period, GOSH developed and released the [Global Open Science Hardware Roadmap](#). The result of years of community-focused development that included over 100 contributors, the 2018 roadmap charts a course to make open science hardware ubiquitous by 2025. It emphasizes a “learn-support-grow” model for creating a sustainable open science future. Perhaps more importantly, the roadmap [continues to evolve and expand](#) through continuous community input.

Open source hardware licensing also evolved during this period. Early licenses, such as the [TAPR Open Hardware License](#) and [Solderpad Open Hardware License](#) attempted to adapt open source software licensing practices to hardware. CERN also developed a set of licenses, releasing a popular [version 1.2](#) in 2013. In 2020, CERN released [version 2.0](#) of its license, now a suite of licenses designed to give users a choice between different approaches to openness.

The institutions around open source hardware grew as well. The [Journal of Open Hardware](#), the [Journal of Open Engineering](#), and [HardwareX](#) emerged to document academic open hardware research. The Deutsche Institut für Normung (DIN)—a German standardization organization—issued [DIN SPEC 3105](#) for open source hardware.



Open hardware flourished during this decade of institutional growth. The [RepRap](#) project created an entire desktop 3D printing industry. A wide range of projects fused the worlds of art and technology. In the wake of the earthquake and tsunami that devastated the Fukushima Daiichi Nuclear Power Plant, [Safecast](#) created an open source geiger counter to track the radiation. More recently, ad hoc international teams have created scores of [open source medical devices](#) to fight the COVID pandemic. Throughout, open source hardware proved that it worked, driving adoption well beyond communities drawn to it because of an inherent belief in the power of openness.

The result of this history is a rich community of individuals, NGOs, and companies applying open source hardware principles to a wide range of challenges. Oftentimes one of its greatest challenges is speaking to itself. That is why this report consolidates information from a wide range of discussion into a single resource.

# Open Source Hardware as a Tool

**O**pen source hardware is a tool. Like any tool, it can be effective in some situations and detrimental in others. Distinguishing between those situations requires understanding what it really means to use open source hardware. This section describes the costs and benefits of an open source hardware approach in relation to the creators' goals. It will help you understand what it really means to build a successful open source hardware project and if open source hardware development is the right tool for the job.

## Open Source Hardware Has Real Benefits

Open source hardware can bring tremendous benefits when it is used correctly. Some of these benefits are obvious. Others emerge in time.

### **AN ENGAGED COMMUNITY**

The community is the not-so-secret key to most open source hardware projects' success. An open hardware project's community feels personally invested in the project. That can make the community a peerless advocate for the project, recruiting new users and identifying new uses. Communities are also challenging for competitors to copy.

The open nature of open source hardware means that the community has an incentive to improve the hardware and contribute those improvements back to the project. Open licenses are designed to prevent project leaders from simply integrating community suggestions into proprietary products.

The community can also be a buffer against competitors. Many open hardware projects survive in an environment where they compete against lower-priced competitors on the strength of their community advocacy and loyalty.

Furthermore, open source hardware is not merely an invitation to modify or improve the hardware itself. A project's community is often the source of accessories, new applications, and complementary projects that expand the range of uses for the underlying projects.

## SETTING A STANDARD

Open source hardware projects can become a de-facto standard in their field. The players in a space feel confident building around open projects because their very openness eliminates the risk that the results of this collective effort will become unavailable in the future. Open source hardware gives everyone a common reference point that can evolve along with the needs of the community.

An open standard allows everyone who relies on it to collectively improve the standard without fear of empowering a proprietary competitor. It can also allow everyone to create their own version of the standard without running afoul of restrictive licensing terms.

## COLLABORATIVE DEVELOPMENT

One barrier to collaborative development is the fear that one's work will be "taken advantage of" by one player in the space. Open licensing can force all of the collaborators to play by the same rules, encouraging a sense of common mission that can inspire collaborative development. The very openness of open source hardware acts as a hedge against each collaborator's work disappearing behind a proprietary veil.

## NETWORK OF ADVOCATES

Many successful open source hardware projects draw an explicit connection between their openness and the strength of their advocacy networks. The community built up around open source hardware is constantly looking for new applications and opportunities for the project. These communities are also often sources of support when disputes arise between open source hardware projects and competitors. This support can take a range of forms, from public advocacy to financial support.

# Open Source Hardware Requires Real Investment

Open source hardware does not just happen. It requires real work on the part of the community supporting the project. Some of the most significant costs cited by open source hardware community members are:

## MAINTAINING PUBLIC DOCUMENTATION

Open source hardware relies on open documentation. That includes schematics and design files for the hardware, software, and documentation. One key obligation of any open source hardware project is maintaining a publicly accessible repository of documentation related to the project.

As discussed later in this report, there is something of a split in the community as to how significant of a burden maintaining public documentation places on the project maintainers. This split largely flows from how closely a project's internal workflow inherently mirrors what is required of public documentation. In some cases,

maintaining public versions of documentation requires building an entire parallel infrastructure. In others, it simply involves making existing design documents public. Whatever the case, publishing and maintaining the public documentation will require some amount of resources from the project team.

Nonetheless, it would be short sighted to simply view documentation as a cost of developing open source hardware. Robust documentation fuels real collaboration, which is a foundational strength of open source hardware. Documentation is what allows a local modification to be adopted worldwide, and an individual insight to become the global standard. Documentation is the foundation of open source hardware.

## **COLLABORATING WITH THE COMMUNITY**

There are many examples of both open and closed hardware projects that rely on robust communities for their success. In both cases, these successful communities do not simply appear. Instead, they are the result of careful work and resources allocated by the projects. The importance of intentional community building is not a unique challenge for the open source hardware community. However, because community building is often a key element of open source hardware success, it is worth examining here.

Most successful open source hardware projects rely on an active community to champion its use, provide feedback, and contribute improvements. These successful projects prove that it is quite possible to build successful open source, community-based projects.

Community engagement can take a number of different forms depending on the nature of the project. Regardless of the form, cultivating and sustaining this community requires dedicated time from the project team. Open source hardware projects that overlook the importance of community engagement or view it as a secondary

concern are rarely successful. Launching an open source hardware project without devoting resources to building and maintaining a community is similar to launching a product without dedicating resources to marketing. The project will exist in the world, but it is unlikely that users and customers will find it.

Open and closed projects that rely on community engagement require dedicated resources to manage those communities. While open source hardware projects that intend to rely on communities must take steps to support those communities, those obligations are functionally identical to the steps that any community-dependent project needs to take. If the team does not have the resources to devote to community cultivation, neither an open nor a closed community-based strategy is likely to be successful.

## **ANSWERING QUESTIONS**

Closely related to community engagement more broadly, successful open source hardware projects tend to receive questions from their community. While these questions can help the project team identify problems, as well as improve the hardware and accompanying documentation, they can also divert resources away from the core development of the hardware itself. If the project team lacks the interest or capacity to handle questions (or even create a system to triage them), they are less likely to find open source hardware to be a successful approach.

Similarly, scaling the capacity to answer questions should be part of any project's roadmap. That does not mean that the team needs to start with the capacity to engage with many more questions than they receive at the start. Instead, a structure for answering questions should be designed to scale the same way production and sourcing might—over time and in response to growth.

## COMPETING AGAINST COPIES

Open source hardware is designed to be copied. This is key to its strengths in fostering community, collaboration, and diverse applications. However, it can also result in projects competing directly with copies.

Successful open source hardware projects use many strategies to operate in a world with copies. Cultivating active communities, innovating faster than competitors, fostering a reputation for superior build quality, and developing exceptional user support are all ways to distinguish successful projects from their competitors.

While each of these strategies can be successful, they require intentionality, resources, and time to implement. If these types of strategies—and other strategies highlighted in this report—are poor fits for a project, open source hardware may not be an appropriate strategy.

## Hardware is Not Software

Although open source hardware shares a history and ideology with open source software, the two are distinct in meaningful ways. Understanding the differences between hardware and software is especially important for community members joining the open source hardware world from the open source software world.

### EVERY PIECE OF HARDWARE IS UNIQUE

Creating perfect copies of software is easy. Every user who downloads a program creates a copy that is identical to the original and to every other copy. That means that anyone can become a distributor of a piece of software, offering perfect versions to all comers.

Hardware is very different. Creating hardware relies on manipulating the physical world, which means navigating imperfections. Given the exact same design files, 10 different manufacturers will create 10 pieces of hardware that differ from one another in ways large and small.

As a result, the manufacturer and distributor of hardware—the manufacturer and distributor of a given physical piece of hardware, not the creator of the original design—has more influence on the quality of an individual instance of that hardware than the distributor of software would have on an individual copy of the software. The entity responsible for the actual piece of open source hardware in your possession is much more important than the server hosting the open source software you download.

Many successful open source hardware projects use this difference to their advantage. They cultivate reputations for high-quality manufacture, not just high-quality open source designs. Two providers of identical software may be reduced to competing on cost. Two providers of identical hardware will still compete on cost. However, they will also compete on build quality in a way that is not relevant in the software world.

### OPEN SOURCE HARDWARE IS NOT EXPECTED TO BE FREE (AS IN BEER)

In addition to being perfect, reproducing software is also free. The lack of a marginal cost for software reproduction has created a wariness in the open source software community around selling software. This wariness also creates a generalized suspicion of payments connected to the use of open source software in many parts of the community.

In contrast, there are real marginal costs for reproducing hardware. Hardware is made up of stuff, and building and shipping stuff (as opposed to bits) costs money. Even an open source hardware project operating at cost will have to charge users (or raise funds) in order to get hardware to users.

One result of this dynamic is that the open source hardware community is much more comfortable with the idea of charging for—and paying for—hardware projects. That comfort extends beyond simply covering costs into supporting the building of sustainable businesses. This is why open source hardware companies built on selling the hardware itself form a significant portion of the open source hardware community. It also provides a welcoming path for new members of the community hoping to create financially successful open source hardware.

## SCALE MATTERS IN HARDWARE

Because each piece of hardware needs to be physically assembled, scale can be much more important in the world of open source hardware than it is in open source software. The process of creating and distributing 5, 50, 500, or 5,000 copies of a software package is largely identical. In hardware, the processes for creating and distributing 5, 50, 500, or 5,000 pieces of hardware can be fundamentally different. Parts availability, assembly steps, and quality control processes can be very different, depending on how many pieces of hardware you are creating.

As a result, it is much more important to plan for your initial distribution scale in hardware than it is in software. While this is not a unique requirement when comparing *open source* hardware to *closed source* hardware, it does become unique when comparing open source *hardware* to open source *software*. Many members of the open source hardware community come from the open source software community (as opposed to the closed source hardware community), so recognizing this distinction has been hard won.

Lab workshops and collaborations at the 2018 Gathering for Open Science Hardware in Shenzhen, China



# When Does it Make Sense to Use Open Source Hardware?

If open source hardware is a tool, when is it most effectively deployed? Some of the most successful open source hardware projects incorporate one—or many—of the following scenarios.

## When Using it to Set a Standard

Open source hardware is well positioned to define a standard because everyone involved has equal control of it into the future. A community or industry that does not have mutual trust can work together to build an open standard that defines a collective, interoperable starting point for further innovation.

One of the most prominent examples of this type of standard setting is the form factor and layout of the original Arduino microcontroller boards. The original Arduino's shape, size, and pin layout formed the basis of countless add-ons designed to sit on top of the Arduino. The open nature of the Arduino gave these additional builders confidence that, even if Arduino itself abandoned the shape, it would be easy for others to continue to manufacture boards that would be compatible with the accessories. Similarly, a new generation of “eather-compatible” boards, built to be compatible with the open source microcontroller boards developed by Adafruit, prove that standards can continue to evolve in the open community as needs and interests diversify.



Openness has also fueled standardization in device interoperability across companies. Adafruit's [STEMMA](#), Seeed Studio's [Grove](#), and Sparkfun's [Qwiic](#) are largely mutually compatible ways to connect development boards, sensors, and other components.

Open source hardware has also helped to standardize open source scripting languages to hardware. [Circuitpython](#) is now supported in a [wide range](#) of open source electronics, and the open source nature of those electronics makes it easy to adapt them for even more uses. There are also active projects bringing [Go](#) and [Rust](#) to electronics. The evolution from IDEs to scripting environments has made it easier for many creators to quickly move into hardware, and to shift between different platforms.

## When Encouraging Hackers

Some hardware is specifically designed to be hacked, modified, improved, and torn apart. It assumes that its community has the technical skills to make modifications and the inclination to do so.

Open source hardware is perfect for these kinds of communities. The documentation makes it easy for the community to understand how the hardware works and the licensing eliminates legal barriers to modifying—and distributing modifications of—the hardware. Open source hardware becomes a clear invitation to hacking communities to gather around the project and contribute to its evolution.

## When Designing to Teach or Train

Similarly, many open source hardware projects are explicitly designed to teach users about the hardware itself. Even if those users do not start with the skills required to modify the hardware, open documentation can help them explore its workings and understand how various elements work together. It can also incentivise learners, giving them the confidence that once they understand the hardware, they will be free to take it in new directions.

This approach to teaching can be focused on the hardware itself by teaching users how a given piece of hardware operates. It can also be focused on teaching users broader concepts connected to hardware design and operation, with the open hardware itself becoming merely a tool toward that larger goal.

## When Targeting a Non-Commodity Market

Some open source hardware projects are designed to succeed by targeting higher-end, non-commodity markets. This approach is most successful for hardware that is technically complex to manufacture or calibrate. In those situations, even if competitors are able to access the schematics and technical information about a hardware project, few are likely to have the capacity to create the hardware with the level of precision it requires to work effectively, or to support it in a way that meets user expectations.

Many users of this type of precision equipment are themselves technically sophisticated. As a result, they place a high value on openly available technical information about the hardware, and may take the opportunity to contribute improvements back to the project.

## When You Want to Turn Users into Advocates

Open source hardware can help create trust among projects and communities. This trust can create a group of user-advocates who are deeply invested in the success of the project. They work to improve the hardware as part of their collective endeavor and advocate for the use of the project in a wide variety of circumstances.

Openness plays a key role in this trust, for at least two reasons. First, it gives users confidence that their contributions will not merely be locked away by the project team without benefiting the community. Second, if the original project team does lose the confidence of the community, the community knows that it can simply take the project somewhere else.

## When You Have Trusted Documentation and Manufacturing Quality That Set You Apart

While in theory anyone can manufacture a piece of open source hardware, not everyone can manufacture it in a way that reliably matches user expectations and the original documentation. Developing a reputation for reliability, accuracy, and support can help a project succeed when it is competing against lower-quality clones.

It is true that, in some cases, clones will be “good enough” to compete against the nominally higher-quality versions. However, for applications where quality matters, open source approaches can help build a community without undermining a competitive position.

## When Engaging Your Community is at the Center of the Project

As described in this section, open source hardware provides a solid foundation for community building around projects. While strong communities are not key to the success of all projects, projects centered on community will often find that open approaches bring strong benefits.



Hallway conversation at the Open Hardware Summit 2018



## When Does it Not Make Sense to Use Open Source Hardware?



Open source hardware is not always a successful approach. Projects with the following characteristics may find that the costs associated with sustaining an open source hardware project outweigh the benefits of the approach.

### When it Targets a Community Primarily Interested in Non-Modifiable or Expandable Projects

Open source hardware allows a community of users to modify and expand the hardware in order to take it in new directions. However, not every project is targeting a market that values that type of flexibility. If a project is primarily geared toward communities who are unlikely to care about how the project works, let alone having the opportunity to modify it, the project may not be a good candidate for an open source-based approach. This is especially true because, in at least some cases, optimizing designs for openness can involve tradeoffs in other areas. If the users of the product are unlikely to engage with its open nature, the project may be better served by focusing on things that the users do prioritize.

### When Maintaining Public Documentation is in Conflict with Other Goals

Although this is not a universally held view, some members of the open source hardware community have noted that maintaining accurate and useful public documentation distracted from the core mission of the project and diverted resources away from other priorities. While some projects

embrace open source hardware in part to instill a discipline around internal documentation practices, other projects feel that rigorous public documentation requirements create an additional step in the engineering process.

If your internal project documentation requires significant revision to be shared, and that revision process will draw critical resources from elsewhere, you may find that the open approach is not a good fit for the project.

## When Maintaining a Community is in Conflict with Other Goals

Project communities do not simply appear. Fostering a community requires a real investment of time and resources, both in its initial creation and ongoing health. Some projects see this role as central, of equal value to engineering and design. Others view community development as a competitor to more important priorities.

If your project team is uninterested in investing in growing a community, that community is unlikely to develop. As such, the team should reflect on its community development prioritization before moving forward with an open source approach. If the team decides that community development is a distraction from the core of the project, an open approach may not be an ideal fit.

## When Open Source Hardware is Viewed as a Replacement for Marketing

Some projects hope that simply “open sourcing” their hardware will create a market and community for the project. As with community building, open source hardware can be a key part of a marketing strategy. It cannot, however, be a replacement for a marketing strategy.

Successful open source hardware projects still need to take the steps that any successful project takes, including marketing to communities and users. This report documents numerous ways that an open source approach can help develop and sustain a community. Nonetheless, that development and sustainability is not self-executing. Open source approaches are not a shortcut to success.



# There Are Unresolved Challenges in the Open Source Hardware Community

**T**he open source hardware community is vibrant and growing. As it grows, engaging with a broader range of users, uses, and communities, it encounters new challenges. One purpose of this report is to help document the challenges that have largely been resolved. That allows everyone to learn those lessons quickly.

Another purpose of this report is to identify challenges under active discussion within the community. At a minimum, that may help members of the community struggling with these challenges feel less alone. More ambitiously, documenting these challenges can make it easier for the open source hardware community to work together toward solutions.

## Is Open Source Hardware Documentation a Barrier to Opening a Project?

As discussed elsewhere in this report, there are divergent views about the burdens involved with publicly documenting open source hardware projects.

Some view creating public documentation as a straightforward extension of good internal engineering and design principles. This view prioritizes the act of creating clean, accurate documentation as a necessary step in any design project. Under this view, both open and closed projects should be creating documentation that is immediately usable by others as a matter of course. Within this context, the public version of the documentation is essentially identical to the internal working version. This all but eliminates any burden associated with creating public versions of documentation.

An alternative view understands internal documentation to exist at a significantly lower standard than is necessary for public documentation. This view does not prize creating complete internal documentation. In light of the state of internal documentation, the act of creating public versions of the documentation that are useful to community members outside of the project can be burdensome and distracting. The resources required to overcome this burden can prove to be a significant barrier to adopting an open approach.

It may be that both of these views are correct, inasmuch as they reflect two distinct, but equally viable, approaches to design. Nonetheless, the split is real and contributes to some level of frustration within the community. In some ways the correct approach for a given project is tied to the alignment between its internal style of documentation and its intended users. Unstructured internal documentation approaches for a project targeting beginners are likely to require significant additional work to create the public documentation. The same unstructured approach for a project targeting users comfortable with the technology may require significantly less preparation before being released.

## Should the Community Distinguish Between Levels of Openness?

The Open Source Hardware [Definition](#) establishes a clear test for what is, and is not, open source hardware. OSHWA's [open source hardware certification program](#) approaches this distinction as a binary one: hardware that meets the definition is certified as open source hardware, hardware that does not meet the definition is not.

It is also possible to view open hardware as more of a spectrum. Open hardware that incorporates proprietary components can currently meet the standards of the definition as long as those components are generally available on the market without having to sign an NDA to access them. Should the community celebrate hardware that goes beyond that obligation, by striving to integrate (and even create) open components? Should that celebration be structured to avoid reducing the recognition of hardware that “merely” meets the definition’s requirement? Would this approach further confuse the definition of open source hardware?

## Is There a Place for Almost, or Aspirationally, Open Hardware?

The open source hardware community is rightfully wary of “open washing”—the practice of hardware projects claiming to be open without meeting the definition of openness. Whether motivated by cynical marketing instincts or ignorance, promoting non-open hardware as open undermines confidence in open source hardware.

Nonetheless, there are projects that fulfill many of the requirements of openness, or that aspire in good faith to become open one day. Should the open hardware community distinguish these projects from other non-open hardware? If so, what should that distinction mean?

Furthermore, is there value in intentionally engaging with aspirationally open hardware projects? If so, what are the best mentorship and pathway-building strategies to enable aspirationally open hardware to become fully open hardware?

## Is it Productive to View Open Hardware as More than a Dichotomy?

The open source hardware community often views hardware as being either open or closed. There is an alternative approach that offers three categories: open, non-open, and closed.

Under this rubric, open hardware is hardware that fully complies with the open source hardware definition. Closed hardware is hardware that is actively closed, where the creator does not share design files and uses property rights to exclude others.

The third category, non-open hardware, identifies hardware where the project team has not taken intentional steps to embrace either approach. They may share some design files but have no systems in place to make sure that all design files are shared. While non-open creators may not openly license their rights, they also do not actively enforce their rights against others.

The non-open hardware category encompasses a vast middle ground of creators, both casual and commercial. These projects are not open in the sense that they comply with the definition. Should the community view them as fully closed?

## Is “Just Copying” Wrong?

Many members of the open source hardware community focus on the freedom that open source hardware gives them to modify, improve, and redesign hardware. This embodies a vision where open source allows others to build upon existing hardware.

While this is a key benefit of open source hardware, the open source hardware approach also allows others to simply copy the original design.

This type of copying causes some projects to avoid open hardware approaches, and causes others to leave the community. As such, many members of the community view it in a negative light. Others view copying as the best compliment their work can receive. Are there other ways to understand the behavior?

Although many exciting applications of open source hardware involve modifying the hardware itself, many celebrated uses of open source hardware are simply situations where open hardware has been used in unexpected contexts. Does that suggest that open hardware projects do not need to be modified in order to be successful, and that copies can achieve these goals as well? Is this dynamic unique to open source hardware, or is it similar to the dynamic in other open communities?

Someone who makes an identical, but lower cost, copy makes that hardware more accessible to more users. This accessibility comes at a cost—it may undermine the incentive to innovate in the first place or continue to innovate in the future—but it also brings benefits—the hardware is available in a way that it was not before. How should the community take this into account?

Is it useful to distinguish between copies that exist in a richly collaborative ecosystem and copies that exist in an otherwise low-engagement community? Put another way, are mere copies less bad if other parts of the community are actively contributing back to the project?

How broadly should the concept of “copies” be applied? Many successful open source hardware projects are replicated using locally sourced materials that differ from the original design. Should these localized versions be considered copies? If so, should the community approach them differently?



Collaborating on creating a roadmap  
for open scientific hardware at the 2017  
Gathering for Open Science Hardware in  
Santiago, Chile



# Opportunities for Open Hardware

**T**his report captures open source hardware at a 10-year anniversary. The final section suggests some opportunities for the community in the next 10 years.

## Be More Intentional About Diversifying Open Source Hardware

OSHWA has taken significant steps to diversify the open source hardware community through the structure of the Open Hardware Summit. The summit itself was founded by two female engineers: Ayah Bdeir and Alicia Gibb. It has historically included a majority of women in leadership roles, especially Addie Wagenknecht, who chaired the summit from 2013 to 2018. In addition to being an early adopter of a code of conduct, the summit reserves approximately one-third of its entire budget for support of the Ada Lovelace Fellowship program. The Ada Lovelace Fellowship program began as a program to encourage women to participate in the community, and has expanded over the years to support a wide range of participation from historically underrepresented groups. The summit also structures and regularly evaluates its speaker selection process to promote a diversity of presentations.

GOSH has also embedded principles of equity and diversity into its community from the start. It has centered these values through its manifesto and detailed [code of conduct](#). GOSH has also held major events [beyond the United States](#) and outside of the global north, including in locations such as [Chile](#) and [China](#). These foundational decisions have helped to foster robust regional communities, including [AfricaOSH](#) and, in Latin America, [reGOSH](#).

Most recently, the [community resolution to redefine SPI pin names](#) is an important step toward recognizing how structures and terminology within hardware can exclude potential participants in open source hardware.

These types of steps arguably make the open hardware community more diverse than some other open source communities. Women serve in major leadership roles in the open source hardware community. Alicia Gibb is the president of the Open Source Hardware Association. Jenny Molloy and Shannon Dosemagen are key leaders in GOSH. Women are at the center of a number of successful open source hardware companies, including Limor Fried at Adafruit and Lenore Edman at Evil Mad Scientist. Naomi Wu is one of the most vocal advocates for open source hardware approaches online and in China.

Nonetheless, these efforts have not yet succeeded in cultivating as broadly diverse a community as the community needs. The open hardware community must continue to work to diversify itself both by creating a welcoming space for all creators and by actively reaching out to invite those diverse makers into the community.

In addition to striving toward a diversity of creators, the open source hardware community will continue to work to diversify how it defines hardware itself. Although there are important exceptions, today's open source hardware skews toward electronics. Outdoor equipment, board games, educational tools, and physical art installations are also open hardware and should be more fully recognized as such.

## Expand Academic Pathways

Some of open source hardware's strongest advocates and practitioners come from academia. The GOSH community has made strides in integrating open source hardware and science. Unfortunately, because academic recognition is centered on patent applications, it can often be hard for those academic advocates to receive recognition from their professional peers for contributions to open source hardware.

Academic hiring and promotion committees have processes in place to review contributions that are documented in patent applications. They are less comfortable evaluating contributions to important open source hardware projects. The open source hardware community will work to make it easier for academic communities to track and recognize open contributions from their peers to the open source hardware community.

## Openness Requirements in Grantmaking

The open access movement has been incredibly successful in building openness requirements into major grants. Today many grantmakers require research papers to be licensed under Creative Commons licenses and data to be freely available to others.

There is no reason that these types of requirements cannot extend to hardware funded by these same grantmakers. OSHWA's open source hardware certification program makes it easy for grantmakers to verify compliance with these types of requirements. Just as a Creative Commons license acts as a guarantee that research papers are openly licensed, an OSHWA certification can verify that any hardware created under a grant meets the requirements of the open source hardware definition.



## Create Model Repositories

Open source hardware is already broad enough to make it a challenge to identify one “right” way to structure a project. Nonetheless, there are steps that the community can take to make it easier for newcomers to get started in open source hardware.

One obvious approach is to create model repositories for open source hardware. These repositories can clearly structure hardware and software files. The readme files can structure documentation in a standardized way, as well as present licensing information in a common format.

These model repositories would not be intended to become a required form for open source hardware. Nonetheless, they can help shape expectations by providing clear starting points for new projects and reference structures for projects interested in organizing data in a way that is clear for users.

## Develop Open Components

One of the strengths of open source software is that any given open source software project is created from building blocks made up of other open source software components. That is rarely the case with today’s open source hardware. While the hardware itself may be open, it is usually built from proprietary—although commonly available—components.

In time, there is value in cultivating an open ecosystem of components to complement higher-level open source hardware projects. These open components can give the community even more flexibility in designing and creating hardware, as well as a deeper understanding of how the components truly work. This may feed into an even richer open source hardware ecosystem.

## Improve Open Design Tools

The open source hardware community is made up of people who can design and improve open source hardware. Today, the process of learning these design skills is an involved one. Most tools are proprietary and expensive, and free tools can lack usability and power features.

This feeds a dynamic where only a relatively small portion of the community has the skills and tools required to contribute back to open source hardware projects. Many of today’s open source hardware projects note that contributions to the hardware components of open source hardware are often dwarfed by the contributions to the project’s software components. This is, at least in part, due to the larger number of community members with the skills and tools required to contribute to software. Making it easier to access and use hardware design tools could significantly expand the pool of people with the ability to contribute back to projects.

Creating easy-to-use, powerful hardware design tools is no simple task. Nonetheless, doing so could fundamentally alter the contribution dynamic within open source hardware and greatly accelerate its growth.

## Conclusion

**T**he first 10 years of open source hardware have been incredibly successful. What started as a largely theoretical approach to hardware design has blossomed into a real, vibrant ecosystem and community. Open source hardware has proven itself to be flexible, adaptable, and robust.

This was not inevitable and is rightly celebrated by the community. This report captures some of those successes. It also recognizes that the open source hardware community continues to grow and evolve. By identifying key trends and opportunity, it helps the community communicate with itself, consider priorities, and move forward into the next 10 years.

## Appendix

**O**ne way to track the growth of open source hardware is through OSHWA's [open source hardware certification program](#). The free program verifies that hardware complies with the community definition of open source hardware. Certified hardware obtains the right to display the open source hardware certification logo:



It is also issued a unique identifier to help users find documentation for the file. The [certification directory](#) serves as a convenient starting point for finding open source hardware. It also serves as a leaderboard, tracking the top countries and entities creating certified open source hardware today.


## Top Open Source Hardware Certifications by Country

Country	Nº of Certifications
United States of America	673
Germany	90
Bulgaria	73
Mexico	24
India	17
United Kingdom	16
France	11
Spain	11
Australia	8
Switzerland	8
Japan	8

## Top Open Source Hardware Certifications by Entity

Entity	Nº of Certifications
Adafruit Industries, LLC	409
Watterott electronic	65
OLIMEX Ltd	58
Field Ready	55
SparkFun Electronics	33
LulzBot	24
ELECTRONIC CATS SAPI DE CV	15
Hummingbird Hammocks, LLC	12
ANAVI Technology	11
NeuroTinker, LLC	9
Great Scott Gadgets	8

 **Engelberg Center on Innovation Law & Policy**  
139 MacDougal Street Room 408 | New York, NY 10012

 [NYUEngelberg.org](http://NYUEngelberg.org)

 [engelberg.center@nyu.edu](mailto:engelberg.center@nyu.edu)

 [@NYUEngelberg](https://twitter.com/NYUEngelberg)

 **ENGELBERG CENTER**  
on Innovation Law & Policy  
NYU School of Law

 **ALFRED P. SLOAN**  
**FOUNDATION**