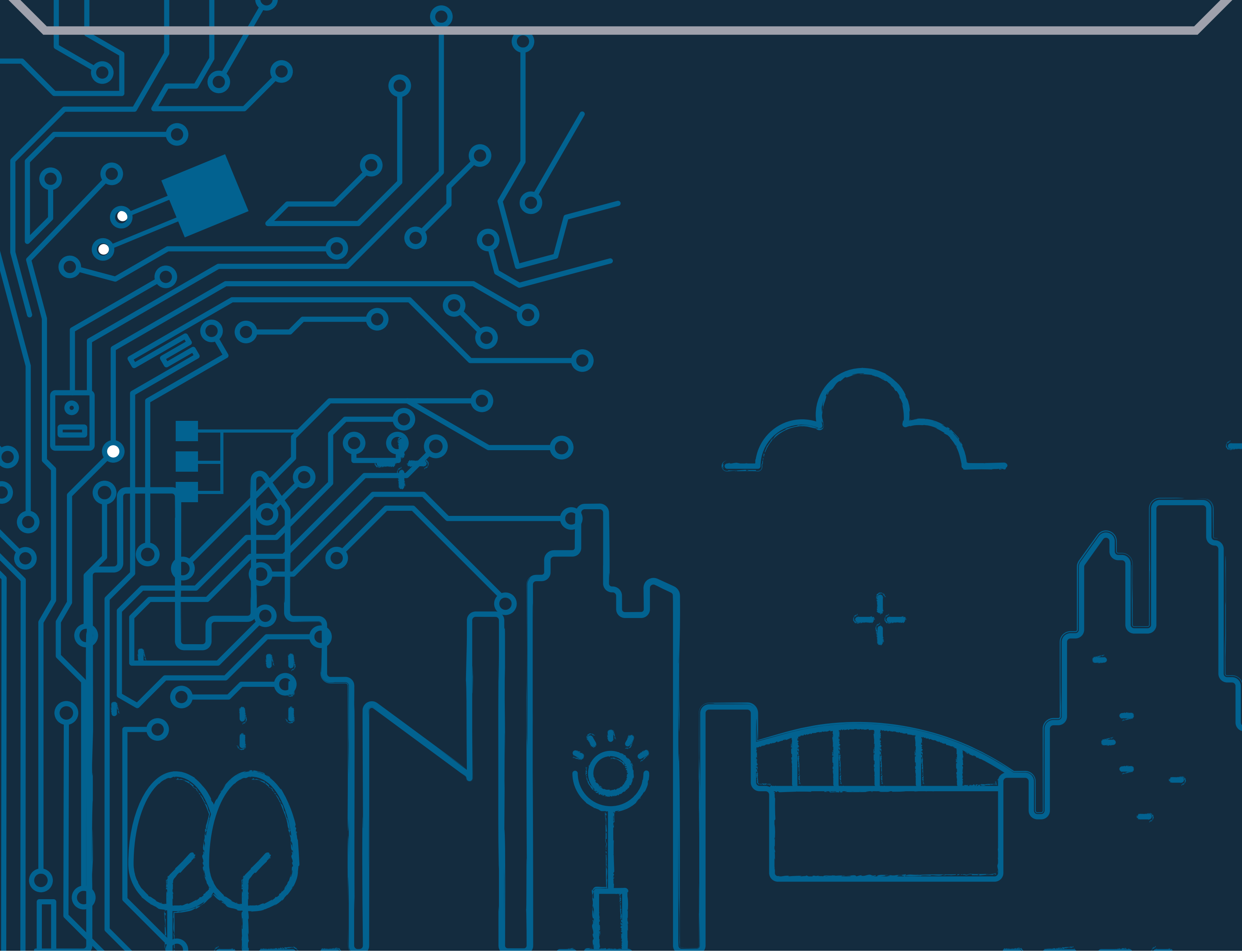
GUIDELINES FOR OPEN SOURCE HARDWARE

IN ACADEMIA AND LARGE SCIENTIFIC COLLABORATIONS

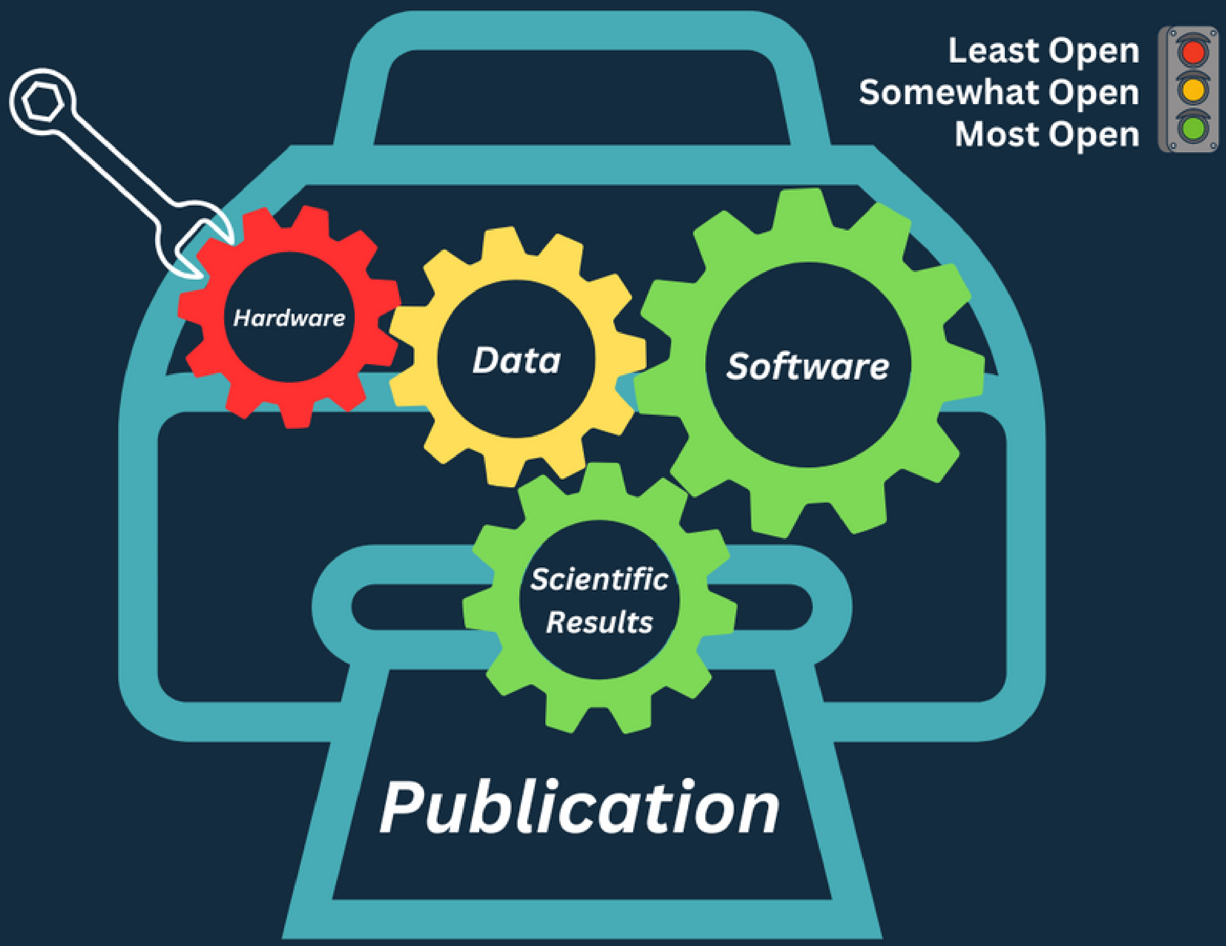


INTRODUCTION

Creating Open Source Hardware is not as simple as passing schematics to the user. Instead, there are numerous critical, often multifaceted, and rarely documented challenges. What threshold knowledge, training, and experience is necessary to translate a design into a physical object? What are the overlooked critical motivators of design choices that are only documented in the designer's mind? What practices make a design valuable for everyone?

The recommendations described in this guidebook arise from the following:

- Feedback provided by past and recent designers, users, and testers of hardware (built by the Columbia Experimental Gravity group) in an open-source context from the perspective of their diverse careers and life experiences.
- Input collected from members of LIGO-Virgo-KAGRA, IceCube, and VERITAS Collaborations* on the feasibility of moving hardware developed in large international collaborative settings to open source science, and specifically open source hardware.
- Recent experience of high-school and college students who participated in an end-to-end exercise of testing publicly-available documented hardware that was written over a decade and a half ago.



ROADMAP FOR OPEN SOURCE HARDWARE

Include a Path for Growth

Design

1. Broadly and globally accessible
2. Longevity
3. Future extensions and upgrades
4. Upgrades and downgrades
5. Future proofing
6. Export controls

Documentation

1. Additional Use Case Scenarios
2. Test Environment

Distribution

1. Easy to find online
2. Certification

Adopt Open Source Practices

- Build an inclusive team
- Use inclusive documentation norms
- Write detailed documentation, use guide, test documents and troubleshooting guide
- Include description of purpose

Make Accessible Tools the Standard

- Strive to use free software
- Avoid proprietary and obsolete parts
- Avoid components with hidden physics
- Avoid misuse of parts

Decide on the Circle of Openness

At least two target audiences:

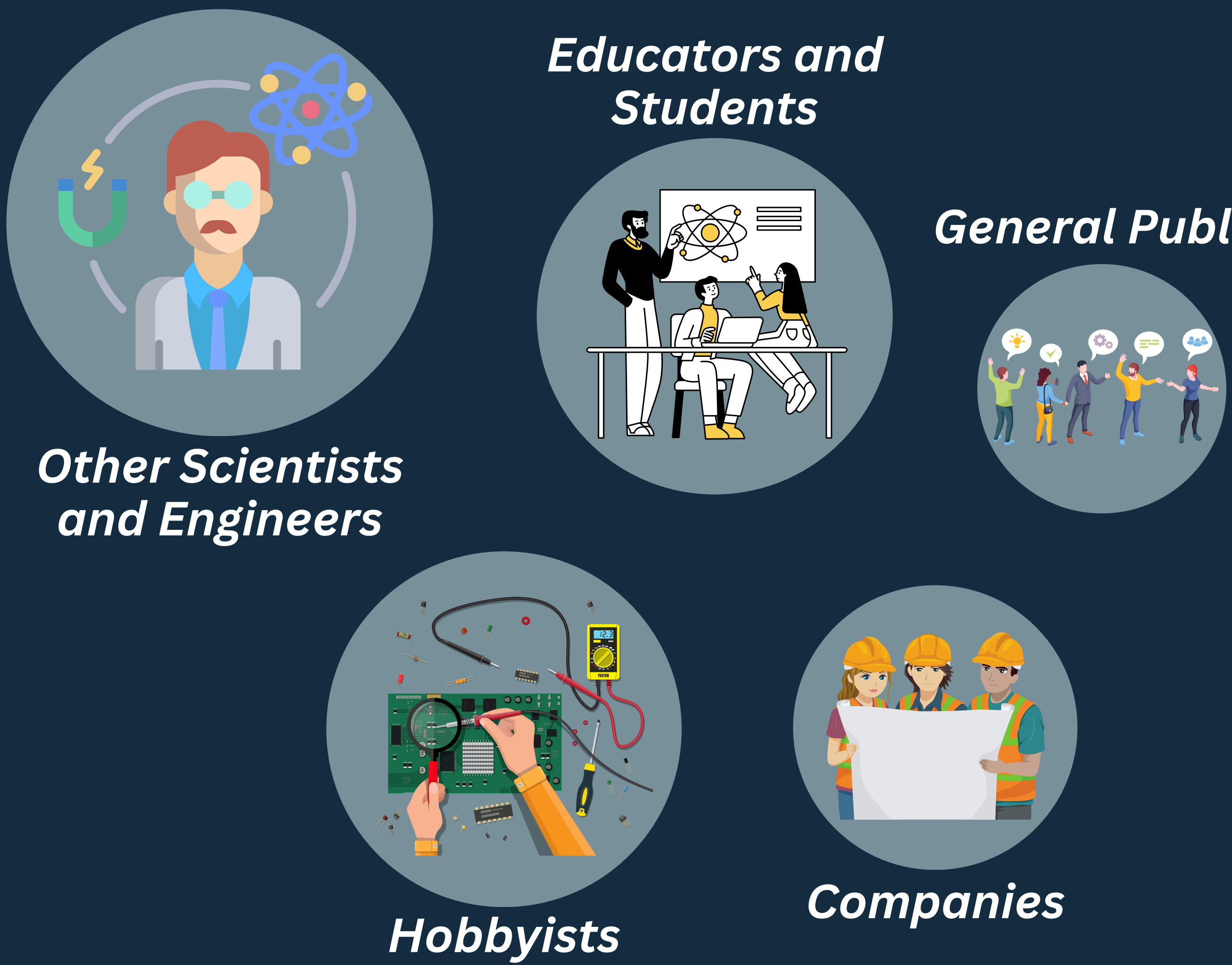
1. Undergraduate researchers
2. Entry-level enthusiasts

Start

1 DECIDE ON A CIRCLE OF OPENNESS

An open source hardware is by definition free to use, study, and modify for any desired purpose beyond that originally specified. Since it is hardware, there is an associated cost to building, which other scientists, researchers, and engineers may willingly provide through allocated research funds and company resources. On the other hand, open source hardware should be available to and easy to use for anyone, including educators and their students of various ages, as well as hobbyists and the general public. The circle of openness indicates different levels of access, dictated by the prior knowledge required to reproduce the hardware, the time investment, and learning curve associated with it.

Understanding one's target audience is crucial to a successful open source project. Our survey results indicated that scientists and engineers in large scale scientific collaborations believe that other scientists and engineers are the primary target audiences of open source hardware projects. In the figure below, the circle size is proportional to the survey results (i.e. a larger circle size indicates a greater number of respondents citing the given group as the target audience).



When the target audience of open source hardware is perceived to be a group of one's peers, the design documentation will specifically be prepared for other experts in the field. As a consequence, it will likely omit crucial background information to the project under the assumption that the reader is already familiar with the subject. This false impression leads to open source documentation that is written at a level far above that which can be useful to an entry-level scientist and, by extension, the general public.

In order to maximize the exposure and accessibility of open source hardware, we suggest considering the following audience targets when determining the circle of openness:

1. UNDERGRADUATE RESEARCHER

Design, document, and optimize the cost of your hardware while targeting an undergraduate researcher in an academic laboratory setting. Such individuals are close to the earliest entry point to academic research. They are new to the field, but also ready to take the challenge of not only recreating hardware but modifying for new use cases. Undergraduates represent the bridge between the academic world and the general public. They are the future experts in academia, industry, as well as in entrepreneurial markets.

2. ENTRY-LEVEL ENTHUSIASTS

We also encourage providing a reduced-cost version of your hardware that is designed and documented with entry-level hobbyists and K-12 educators (and their students) in mind. Current K-12 students are future undergraduates in science and engineering and the future entrepreneurs of the world. The choices of purpose may range from a hands-on demonstration built for an education and public outreach exhibit to a certified open source hardware project that a sixth grade physics class can build in school, to even a small sensor network that one may want to use in a miniature 'spice garden' in an eighth floor kitchen of an urban apartment. The main audience of open source hardware, therefore, are students of all ages who have limited experience doing hands-on work. Consequently, it is crucial to be mindful of planning the design to be within the limits of the students' level of knowledge as well as the accessibility of the materials. Among the things to be aware of:

- ***Safe Use:*** Design and document your design in a way that allows for safe use even for those without prior experience. For instance, parts of the design that require a particularly high voltage or a a dangerous tool should be very well documented if not avoided altogether.
- ***Cost Effectiveness:*** Be mindful of the tools and the funding to which student researchers have access. They often do not have the resources to spend on expensive, customized components. Try to use parts that a student can realistically have access to or purchase.

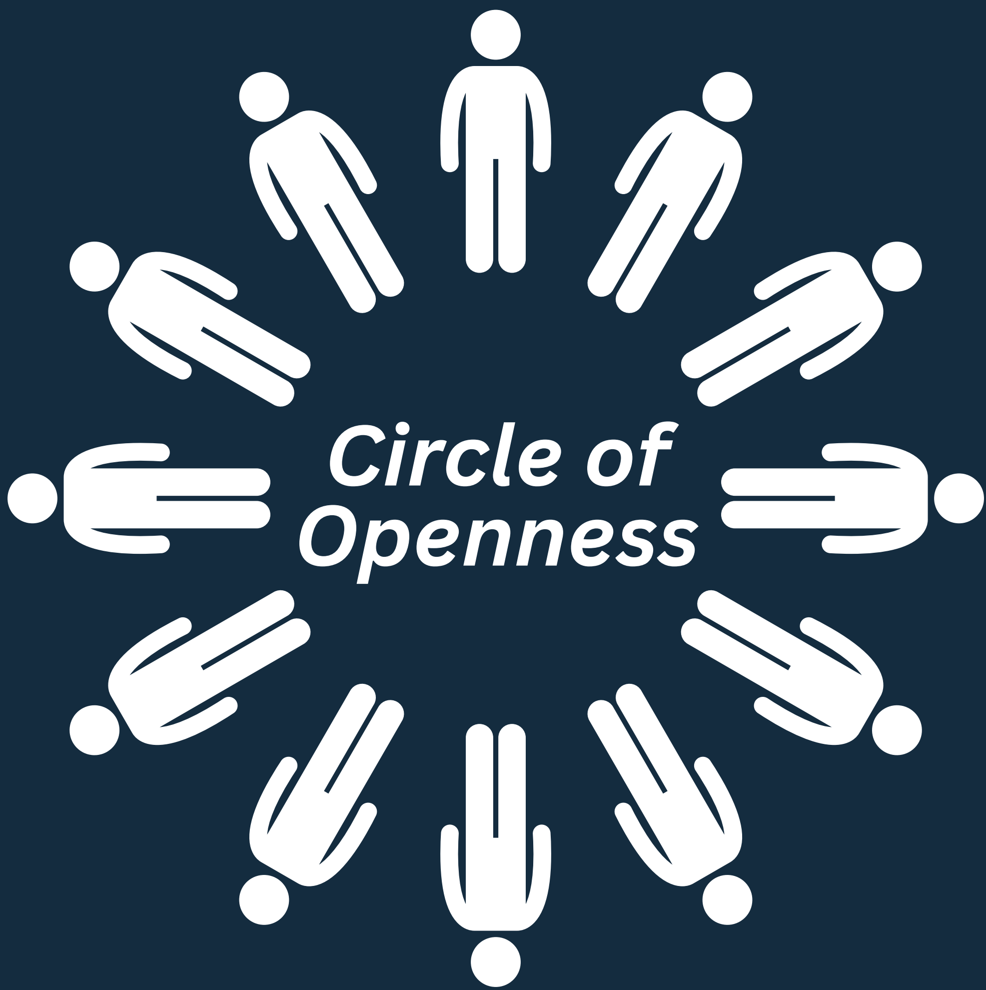
Taking it a Step Further

There are additional measures that creators can take to make their hardware appeal to a greater audience, thus broaden the 'circle of openness ' of their project. One example suggested in our survey responses, and that is a "difficulty level indicator" on projects. The respondent said:

"Naturally, open source projects are going to come in gradations of difficulty, and having a designator on a project will allow users to select a project closest to their skill level without having to sift through dozens of projects."

This proposed measure would serve to increase the accessibility of open source hardware and appeal to a more diverse demographic.

The desired circle of openness will set the open source hardware project goals, parameters, and deliverables. Naturally, as academic research targets creating and discovering the previously unknown, projects with steep learning curves will continue to persist. Documenting hardware in accordance with open source practices is nonetheless critically beneficial for any large scale academic collaborative project.



2 MAKE ACCESSIBLE TOOLS THE STANDARD

Does the availability of manufacturing files make hardware fully open source? Definitely not. Without access to modifiable design files, others cannot fully understand a project, much less contribute to it.

There exist numerous types of design software, whether the hardware in question is an electronics board, a complex mechanical element, or an optical system. The choice of which to use should take account its level of openness and accessibility. A behind-the-paywall design software makes the full project less accessible, thereby limiting growth. While many types of software is advertised as "free," it is important to be aware of certain limitations, e.g. limitations on the number of layers of a printed circuit board, limitations on the maximum size of a design, or limitations on importable part libraries.

The critical dependence of modern designs on the complex simulation and experience-based development process necessitates the communication of the design flow and design rationale beyond the end result. One also needs to keep track of design software updates as well as compatibility (up- and downward as well as cross-platform). A design that is originally created with a free version of software can become hidden behind a paywall when companies update their business model. Using design software that comes with a commitment to being fully open source and forever free is strongly advised.

It is imperative to ensure equal accessibility for all potential users. The responses to our survey second this sentiment, with one respondent stating:

> *"Encourage open source tools. It's easier to use e.g. KiCad for some electronics design if the original hardware design also already used it."*

Software	Cost	Free Trial	Operating System
ALTIUM DESIGNER	$$$	Yes	Windows
EAGLE	$	Yes	Windows/Mac /Linux
Ki	Free	N/A	Windows/Mac /Linux
CIRCUITMAKER	Free	N/A	Windows

As this respondent points out, it is far easier to use and modify an open source project if the original hardware design uses open source tools. Tools such as KiCad, for example, will also broaden the audience of an open source project by lowering the barrier to entry for hardware.

A project that uses only free software is more likely to be used than one that uses software with a large paywall and/or that which only professionals have access. This also benefits the creator of the hardware by increasing potential traffic to a project.

Beyond the software used in the design, one must also consider the specific parts that are purchased and built into the assembled hardware. We encourage standardization of design when it comes to the component parts of a hardware project, as multiple projects which contain similar parts will save users the time and money. We list a few points of caution that will hinder a standardization of design:

- A fabulous design can be easily poisoned by proprietary blocks or devices. Even a single obsolete or hard-to-replace component can prove disastrous from an open source point of view, due to their hidden costs. Such elements/parts should be avoided completely.
- Amazing sensors highly advanced capabilities and shortcuts may come onto the market with professional-looking data sheets, but such technology often lacks detailed calibration instructions and proper description of components. Such designs that involve extensive hidden physics can be as problematic and confusing and should therefore be avoided.
- Parts are sometimes grossly misused as cost-saving tactics in clever-looking designs. Unfortunately, not only are these kludges hard to understand, but they can also compromise design integrity, longevity, and safety. Hard-to-understand designs can hide serious problems, making it best to avoid them both as a user and as a designer.

The standardization of the tool-set used during the design, production, testing, and documentation of a hardware project will determine the level of accessibility. A fully open project has no hidden physics, misuse of parts, or behind-the-paywall elements and enables future extensions in a fully collaborative manner.

We note that large international scientific projects set design and review requirements for their hardware as well as the associated tools and documentation. The longevity of such projects necessitates those not involved in the original design to be able to fully understand it as well as make the necessary replacements, repairs, or adjustments, even when the original designers are no longer accessible. Designing and building hardware via open source tools from the get-go is preferable and should be adopted as the standard by collaboration, as it enables new participants and members to easily contribute even to an older project. Considering the increasing labor costs of experts, a proper open source design is always the best cost-saving choice in the long term for both small and large teams in academia.

3 ADOPT OPEN SOURCE PRACTICES

Open source hardware may begin as an open source project, but as complexity increases, it inevitably becomes a team venture. Beyond the hardware, the team's other significant product is the documentation. Proper documentation enables the hardware to succeed in the open source project world and in the wild, surpassing its original academic purpose.

The requirement that anyone can contribute to open source projects encourages the designers in charge to seek to build an inclusive team. A "Statement of Inclusivity" accompanying the project documentation will encourage brilliant minds worldwide to contribute and collaborate with the original hardware development team. Globally appreciated and respectful use of terminology is a key factor in the broad success of open source projects. Documentation norms that are inclusive of neurodiverse developers (e.g. color and font-type setting) are encouraged and should be adopted from the beginning. The documentation starts before the design process by selecting and setting up a widely-used, version-control-capable documentation interface that is accessible for the entire team and scalable. The documentation system should be as automated and comfortable as possible to allow for a continuous barrier-free documentation process in near real-time.

Learn from Open Software Practices

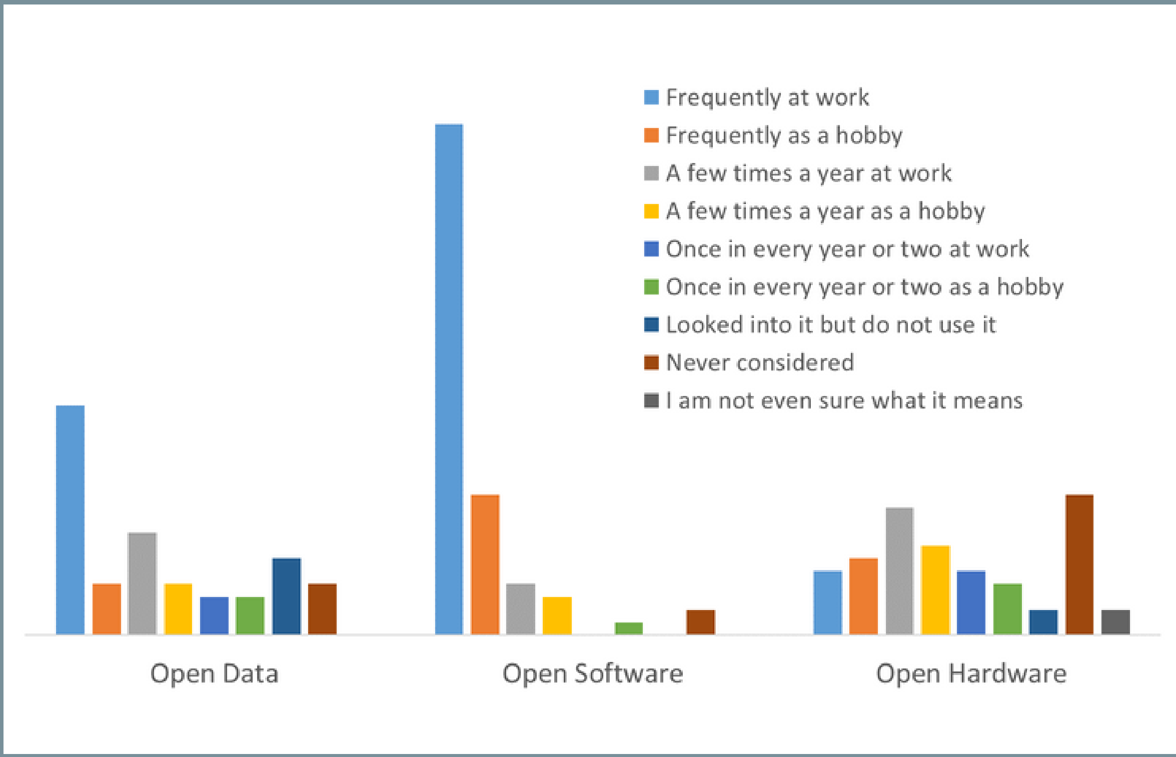
Open source software is already an influential tool in academic research and teaching. Consequently, proponents of open source hardware ought to learn from the successes of open source software and implement, when applicable, analogous practices that made it so widely successful.

The success of open source software is well-documented among the academic community. A majority of those surveyed in the LIGO-Virgo-KAGRA, IceCube, and Veritas collaborations stated that they use Open Source Software frequently for work (Figure on next page).

Version Control, Git, and GitHub

Open source projects thrive when supported by a universal, free, and accessible platform. Such a platform simplifies collaborative work, encouraging innovation and continual development.

Version control is an invaluable tool used by open source software developers—it is the ability to update a file while the old file version is saved and remains accessible,



creating a history of edits for each file. It allows the creators of open source projects to make any changes they deem necessary to their project, and for users to see both the new and the old versions of a particular project.

Git allows for version control with the ability to "push" changes to update an existing file. GitHub, a leading cloud-based hosting service, stores Git repositories. The advantages of using GitHub is that it permits global user access, modification, and software updating, thereby improving the accessibility of open source science. Git also promotes intuitive and efficient organization of a multitude of file types. The 'Read Me' sections allow creators to clarify their project basics and the documents uploaded.

A significant number of creators have already adopted GitHub as their standard interface for open source hardware. However, Git and GitHub do come with their drawbacks, including the fact that it was designed primarily for text-based files. As a result, hardware files cannot be edited directly on the platform, and users cannot directly see the modifications made to updated files. One way to combat this is for creators to make a note of what was changed in between updates. GitHub also has a limit of 2GB for the free version of the platform.

Large international scientific collaborations are inherently diverse as they consist of people from a multitude of nationalities and education levels, including students, technical experts, professors, engineers, and scientists from all over the world. As a result, every hardware product should be documented in a globally accessible fashion for all members. Such large teams can and do influence open hardware in academia and have implications for how we do science.

At the moment of creation, design choices seem obvious. But when we look back at such choices at a later date, they may become obscure. Moreover, what is obvious to the creator may not be obvious to others. Therefore, documenting the hardware development process as it happens with attention to detail is critical. Further, the designer must also explain the how and the why behind the design at the level of the targeted "circle of openness." It is critical to understand the reasons behind the designer's choices as it makes future changes safer, faster, and easier. It also minimizes the number of design errors caused by misunderstandings.

While these details are critically important, the big picture should not be underestimated. Beyond the individual "tree level" detail, the open source hardware documentation should include a generic description of purpose or the "proverbial forest." This will attract additional users by allowing individuals to easily identify the specific hardware for their desired purpose.

The design becomes a product, ready for open source consumption, when it functions properly, its use and edge cases are tested, and the possibilities for and consequences of major and minor malfunctions are investigated. Therefore a detailed user guide, test documents, and a troubleshooting guide should accompany the design. The end user has the right to repair or modify it, and therefore needs the information provided in these documents.

The adoption of open source practices for teamwork and documentation will significantly influence how quickly new open source hardware project can gain traction, thereby contributing to awareness of open source science as a whole

4 INCLUDE PATH FOR GROWTH—ALWAYS DESIGN FOR SUCCESS

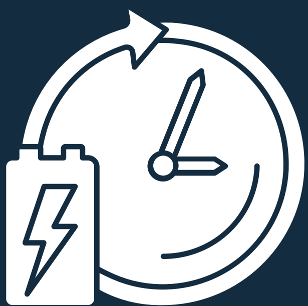
The hallmark of success is growth. The specific level of growth should be anticipated from the start, as the ability to grow rests on the initial choices of design culture, standards, tools, and accessibility. A properly documented, debugged, and published open source project will scale gracefully without putting an unbearable load on the designer. While success is never ensured, it pays to always design for success. The following is a non-exhaustive list of the factors that should be taken into consideration at each stop of the creative process:

Design

1

Broadly and globally accessible design: Institutions and economies perish every day because they underestimated what other people are capable of. The most successful designers design for everyone; that is why their products are ubiquitous from the East African savanna to the concrete jungle of New York. They allow everyone to appreciate, access, and find their design useful. It is much easier to achieve broad usefulness if diversity is represented in the design team from the start. Diverse teams who regularly work with global connections are more likely to design for maximum success.

2



Hardware Lifetime: The use lifetime of an open source hardware design is determined by the first component's obsolescence. Even the note of anticipated future obsolescence from the part's manufacturer makes the entire open source design obsolete, as others are unlikely to invest in such designs. While it might be attractive to save money by relying on bargain-priced obsolete or recovered parts, it is a very expensive mistake for an open source design. The frequent updates, changes, and tests are not only extremely time-consuming and error-prone, but also much more expensive than the money saved on the obsolete part's sale price. Open source hardware designers should always use components with the longest anticipated market lifetime and rely on manufacturers with a proven track record of backward-compatible designs. This practice shall make the required design updates easier, safer, and less frequent. Users will appreciate the stability and the care provided by this future-proofing practice.

It is also important to consider the fashionable topic of recycled, scavenged, or recovered parts. While on the surface this may seem like an environmentally friendly practice for open source hardware design, the hidden cost associated with it can be enormous. Such parts may be a valuable resource for the dedicated expert who is willing to take huge risks, is not recommended for the purpose of open source hardware design. Scavenged parts are not universally available, and their expected lifetime, performance, and function are often unreliable. Their biological and environmental impact are unknown and potentially unethical. Thus, in general, a successful open source hardware project plans for the future by using reliable and obtainable parts.

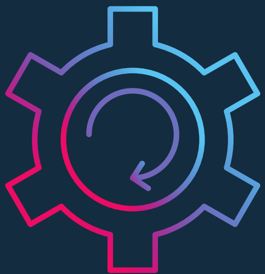
3

Environmentally friendly design: The future of science is green. Industry, academia, and education are putting a greater emphasis on the environmental impact of scientific innovations. Often times older hardware designs become obsolete not because they no longer work, but because they have an outsized environmental impact. Given that one of the core aims of open source hardware is to design a project that stands the test of time, it is crucial to use environmentally safe components when possible, with special attend to end-of-life recyclability. When a user is finished with a particular project, they should rest assured that they can safely dispose of its parts without contributing to global waste.

4

Future extensions and upgrades: An ambitious design that never comes to fruition is less than nothing, as it took time, effort, and resources from the design team and community. Since an open source hardware design can be open from the beginning, academic creators should use this freedom to their advantage. They should plan on *modular design*, where every functional building block can be released separately as soon as it is ready. This way, every level of the grandiose ultimate design is already useful and possibly being tested by potential users. This is an immense help for the creators and also allows the global community to forge their own path by potentially designing modules the original creators may not have even considered! In case the global community designs something the creators planned to do, the result is even better, as the design process had been accelerated and diversified. Embracing modularity, using standardized interfaces, and allowing for the freedom of others to pitch in are the keys for ensuring future external upgrades.

5



Consider both upgrades and downgrades for specific purposes: Often times, users have a limited budget and it is a helpful practice to allow for partial assembly and partial functionality to save money on the parts not included. In open source hardware design, "all or nothing" is less useful than a freedom of assembly completeness. Further, such a design leads to a broader user base and it can be a worthwhile investment.

6

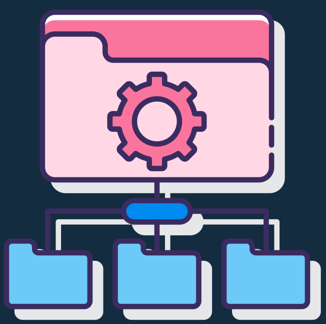
Ultimate future proofing: A future-proof open source design should always use internet-based archival and design tools that have a very long expected service life and that allow for the extraction of the entire design when its lifetime is reached. As risk of sounding paranoid, the archival quality of paper with still has a longer expected lifetime than any alternatives, especially electronic or magnetic alternatives. It is good practice for careful open hardware creators to keep printed copies of design archives. If this sounds excessive, just remember that the seismographs of the Apollo mission had to be re-digitized from paper printouts. Sometimes it is the past that offers the right advice, not the fashionable modernity.

7

Export controls: Much open source hardware happens in the US and Europe; these regions have strict export control regulations that can directly affect open source hardware designs. A design that can only be built in certain countries is not considered fully open source. Hence, close attention should be paid to export control regulations, and open source hardware designs with the goal of global availability should avoid restricted parts.

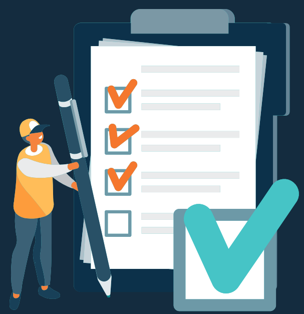
Documentation

1



Additional use cases: One should never underestimate the incredible creativity of humans and the rapid pace at which our world is changing. It is more likely than not that a design will be used for a purpose different from what its original designers intended. It is therefore critical to use a widely-accepted documentation interface with version control capabilities that allows others to easily add to, enhance, or modify the open source design and its documentation. Enabling people to discover and experiment with open source hardware brings designs to life and let them evolve.

2



Test environment: Preserving and carefully documenting the test environment of open source hardware can save a lot time down the line as it is much simpler to check for identical behavior than reinventing the wheel. Open source designers should consider the test environment and its documentation as an integral and necessary part of the design when planning for the future.

3

Access to creators: While the original creators may be wary of user communication, feedback, or suggestions, some of the most successful open source creators embrace every piece of feedback they receive. After all, it makes their product better via diverse globally distributed engineering, visions and thought. In fact, open source creators should not merely embrace feedback from the worldwide community, but seek it out from the very beginning via a globally visible open design process. There are numerous online tools that enable instantaneous communication, such as Discord or Slack (for discussions and hangouts) as well as GitHub-Issues (preferred for design-specific direct requests, suggestions, and updates).

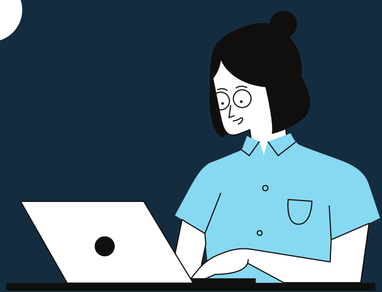
4

Design in handover: As open source projects can have extremely long lifetimes, the designs should be documented with descriptions in extreme detail so that access to the original designer is not necessary. The best open source creators strive for such perfection with design and documentation that they themselves became unnecessary for the future or success of the design. While this ultimate goal is difficult to understand for many academics, the freedom and satisfaction of a truly done and finished project is unparalleled. It is a feeling that every open source hardware designer should experience after each design is completed. A project that requires constant care is a project that was left unfinished.

A successful open source hardware project will take is own course, effectively recruiting its own next wave of creators, like radio waves leaving their antennae. A mature open source hardware project will survive without the creator's supervision. This is the ultimate success for a creator: a prospering project without the need to constantly go back and engage with it.

Distribution

1



Easy to find online: A design that is invisible, unattractive, boring, or obscure is unlikely to succeed. Professional hardware designers often make the mistake of overlooking aesthetic considerations. As proven by the creative Maker movement, designing for success does not only mean comprehensive documentation, but also requires an artistic touch. After all, it takes almost the same effort to design an attractive piece of hardware as it does to design an ugly functional object. Hardware can be attractive not only through function but through design as well. For example, some well-designed scientific instruments are immortalized in the Museum of Modern Art and the Smithsonian Museum. The popularity of Apple laptops over others can be credited largely to their aesthetic choices. It is imperative to involve artistic minds as well as scientific ones, and the earlier one does it, the better the final product will be. The compounded difference of strategic aesthetic choices on the success of the final design may prove substantial.

2

Certification: An isolated designer or design team is less likely to produce a globally impactful success than a diverse team immersed in the global thought process. Anticipating the eventual certification process and the certification of open source hardware is an extremely helpful process that enables success. Since the certification process was developed by experienced open source professionals, it amasses invaluable experience of diverse minds from all over the globe. Certified projects inherit all the knowledge and experience of these professionals, making success more likely and managing what comes after much easier. *Link to OSHWA Certification.*

Some Other Ideas Related to Open Hardware

Opening up hidden hardware—Open Source Archaeology

Often times really successful hardware projects are closed and undocumented. Unfortunately, when companies are acquired, when large scientific projects finish, or when labs are closed, projects deemed as unprofitable disappear forever. From brilliant mechanical designs of steam engines to pioneering game consoles; it happens all the time. Converting such projects to proper and certified open source hardware designs is a great humanistic goal. It is a form of archaeology where most design principles of open source hardware are valid and applicable. Thus, open source hardware movements find value in not only advancing modernity, but also in the preservation of the brilliant ideas of the past to guide future innovation.

Open Source Hardware as the Ultimate Gold Standard

The patent laws of the young United States can be seen in a way as the original open source hardware movement: they require the fully reproducible disclosure of hardware design. That was a kind of revolution, breaking down the secrecy of corporations paralyzing progress. Fortunately, the original open source hardware movement made our life infinitely better. With the emergence of the modern open source hardware movement, including its intrinsic diversity and globalization, our world can experience a second Renaissance of creativity, freedom, and explosion of untamed ideas that have the potential to improve the quality of life around the world. The choice is yours, open or closed. For us the answer is obvious.

**The LIGO Scientific Collaboration (LSC), the Virgo Collaboration and the KAGRA Collaboration, with over 2000 members together, have joined to perform gravitational wave science using their respective detectors. The IceCube Neutrino Observatory is a research facility at the South Pole in Antarctica. Over 300 scientists work together in IceCube. VERITAS is a ground-based gamma-ray instrument operating in southern Arizona; the respective collaboration has dozens of members.*

About the Authors

The authors of this work are: Zsuzsa Márka, Melinda Yuan, Aruna Das, Luke Zerrer, Sunny Hu, Aaroosh Ramadorai, Imaan Sidhu, Leonardo Lobaccaro, Antonio Lobaccaro, Alex Zhindon-Romero, Daniel Jarka, Jeremiah Alonzo, Raymond Provost, Luca Matone, and Szabolcs Márka. *We appreciate the generous support of OSHWA and the Sloan Foundation, which awarded Dr. Zsuzsa Márka with the Trailblazer Fellowship that made this work possible. We also thank the LIGO-Virgo-KAGRA, IceCube, and VERITAS collaborations and all of their members.*

OSHWA
OPEN SOURCE HARDWARE ASSOCIATION
ALFRED P. SLOAN FOUNDATION