Mentoring and Training Guide

for Open Source Hardware in Academia

Introduction

One of the key incentives for creating open source hardware is its educational use. Questions such as *"is it better to learn on the job with hands-on experience or with extensive prior training?"* motivated us to write this guidebook on mentoring and training. While we may never get a definitive answer, the widespread availability of open source hardware eliminates the need to decide on one learning style over another, instead allowing students to learn in a manner that is best catered to their individual needs.

In addition to accommodating different learning styles, open source hardware also serves to bring together creators from different countries, age groups, levels of education, and more. This not only increases the user base of open source hardware, but also diversifies the set of educational tools available for students.

The recommendations in this guidebook are based on evidence compiled from:

- Feedback 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 whether and how hardware developed in large international collaborative settings can benefit open source science, with an emphasis on 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.

Based on the feedback collected from these various sources, we have synthesized to following tips meant to guide both educators and students on how best to take advantage of the teaching opportunities that open source hardware presents.

MENTORING TIPS

- Listen to the students
- Make Open Source the Norm
- Facilitate Teamwork
- Promote Diversity
- Write Accessible Documentation

1 Listen to the Students

by Aruna Das

The graphic above illustrates one example of a productive student-mentor relationship that can arise from the adoption of open source hardware. The student is tasked with a project and utilizes open source hardware that is readily available online and OSHWA certified. The student then proceeds to inform their PI about the open source guidelines, effectively introducing the advantages of open source hardware projects to their PI.

While we often think of a student-teacher relationship as unidirectional, with information flowing from the teacher to the student; the reality is that it is actually a mutual learning process. Especially with this generation of students who grew up with the internet, they are used to information being easily accessible. If they don't know the answer to a question, their first instinct will be to "Google it." Because of their familiarity with openly sourced information, students themselves can even act as catalysts for open source hardware by proposing the idea to their PIs.

In addition to being open to ideas proposed by the student, PIs should also align projects with students' interests. Undergraduate researchers are eager to learn, and it is the job of their mentors to cultivate this excitement. They likely already have an idea of the topics that they are interested in, and PIs should try their best to accommodate these interests.

Listening to students also means listening to feedback. PIs should also solicit students to test their documentation and listen to student feedback. When students identify a certain area of the documentation to be confusing, PIs should take such feedback seriously and modify their documents accordingly. Our students, for example, encountered difficulties when it came to testing open hardware documentation. One of the high school students reflected of how their lack of prior knowledge proved to be a significant obstacle:

"Coming into the project we had minimal experience with open source hardware. Most of us had barely heard of the term before, let alone tested and used advanced open hardware designs .. While the documents were intimidating at first, we also lacked some of the prerequisites necessary to execute the tests."

They continued: *"Another area that we found challenging was understanding the purpose and the mechanisms behind each component of the board and the testing process. Even after reading the documents and gaining a basic understanding of the workings of the ...system, we did not have enough of an understanding of the technology to the level of the smallest components."*

This is a good example of useful feedback on the documentation from the student to the professor. The student clearly and specifically states that aspects of the documentation that were unclear and/or confusing. When a PI encounters feedback of this nature, it is important to listen to the students and make any necessary changes to the project. Open-mindedness and willingness to learn from students is one of the hallmarks of an effective educator.

Advocate for Proper Recognition

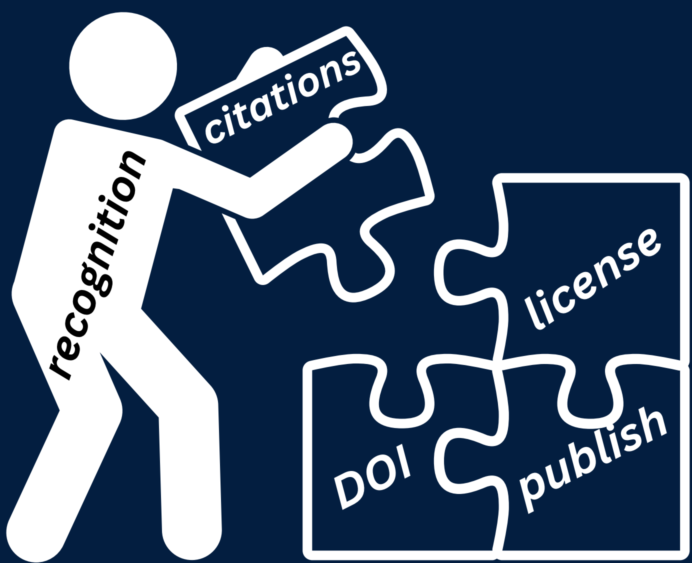
Hardware is an aspect of scientific discoveries that is often overlooked. Publicity, funding, and media tend to revolve around the groundbreaking scientific discovery, but neglect to recognize the foundation of hardware that allowed the discovery to be possible. The reality is that the designers of hardware are often lost in the hundreds of authors that are cited on a publication. For this reason, we strongly recommend that PIs advocate for proper recognition for their mentees in order to set an example for the mentee to advocate for themselves in the future. Below are a few ways of obtaining such recognition:

DOI: One way to accomplish this is through a DOI. A DOI allows for the permanent identification of a publication, thus facilitating proper citation and designation of credit to creators. Traditionally, DOIs were reserved for publications only, but now tools such as Zenodo allow for creators to obtain a DOI for their GitHub page as well. Thus, users of Open Source Hardware can then cite the GitHub page of the original creator.

Citation Instructions: Another important step that we recommend to ensure proper designation of credit is to include a section on the GitHub page for your project on how to properly cite. This step should be done in conjunction with obtaining a DOI, as the DOI can also be listed on the citations page.

License: Creators can also take the step to pursue appropriate licensing for their project. One commonly used license in software is the Creative Commons License, which allows a creator's work to be shared, used, and altered as long as the user abides by the terms stipulated by the license. Creators should research the specific license that is the most applicable to their project, but in general, licenses help to ensure that proper credit is awarded to the original creator.

Publish: Going even further than obtaining a license, we encourage individuals working on hardware to publish in journals. Open source hardware journals include Hardware X and the Journal of Open Source Hardware. Publishing hardware projects in these journals will increase the credibility and awareness of open source hardware as a whole.

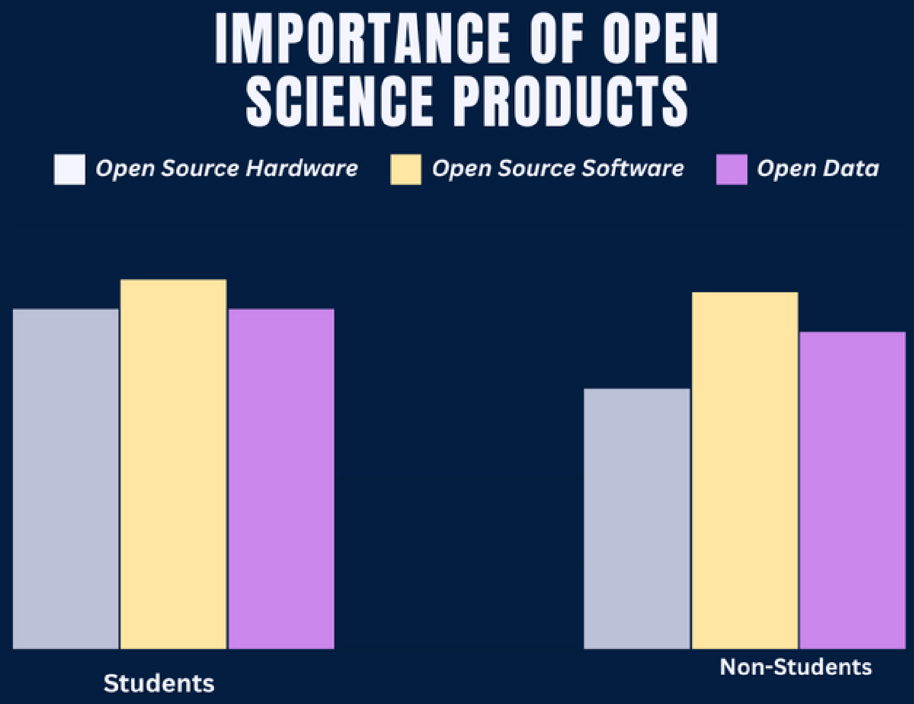


Make Open Source the Norm

PIs and creators should strive to make open source the norm when it comes to hardware in academia. This can be accomplished with a holistic approach, which includes education inside the classroom as well as in the lab.

Curriculum

First, we suggest that educators ought to propose to their respective universities that Open Source Hardware be incorporated into the science/engineering curriculum. By introducing students to Open Source Hardware early on and educating them on the best practices, students will not only be made aware of the abundance of Open Source Projects available to explore, but will also consider the possibility of making their own future projects open source.



When comparing the survey results of the students (including undergraduates, graduate students, and postdocs), to that of the non-students, one stark contrast that emerged was the fact that students were more open to the idea of Open Source Hardware being important to the open source ecosystem than non-students. When asked to rank the importance of Open Source Hardware, Software, and Data on a scale of 1-5, students were more likely to have similar rankings for all 3, while non-students were more likely to ranking Open Source Software and Open Data higher than Open Source Hardware.

These results demonstrate that students, being earlier academic journey, have an open mind when in comes to Open Source Hardware, and it is important to capitalize on this foundational period in their education.

Standardization

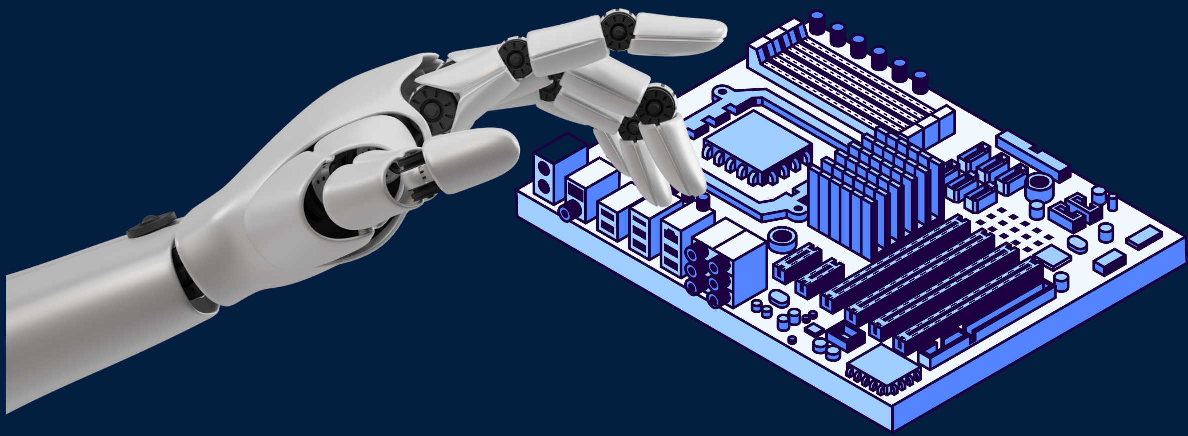
Incorporating open source hardware in the formal undergraduate curriculum serves the purpose of standardizing the open science education. According to the undergraduate survey results, a major barrier to preparedness and successful engagement in the open source hardware space is the lack of specialized and centralized educational resources pertaining to open source hardware. Prior to their project, only two of five undergraduate students had experience with open source hardware. The only three resources used in their prior experiences with open source hardware were Arduino, Raspberry Pi, and YouTube.

In the exit survey, one student noted that their *"future likelihood to engage in open source hardware is limited by the amount of resources that [they] can find online and also how accessible the materials are."*

Another student cited the lack of *"a good source to finding centralized information"* as a main disadvantage of open source hardware.

While the results suggest that resources like YouTube, Arduino, and Raspberry Pi have the potential of easing the learning curve for engagement in open source hardware, the responses also suggest that the level of outreach and awareness of open source hardware among most students is low. Further compounding the issue is that resources for open source hardware education are limited and decentralized. Consequently, students who wish to become educated in open science are expected to find outside resources on their own time. This leads to disparities in education caused by disparate levels of accessibility to such resources. Thus, formalizing open science education as a standard part of the undergraduate curriculum is a key step in ensuring equal accessibility to students.

Hands-On Experience



In addition to open source hardware practices being formally integrated into the curriculum, it is also equally important for PIs to demonstrate open source practices in their own research groups.

The job of a researcher is to educate their team, and the best way for students to learn proper open source practices is to perform them. For instance, students should be documenting hardware to open source standards throughout the course of their research. When students use hardware that is developed by others, they should be taught proper citation practices. Currently, proper practices for citation and claiming academic credit are not well-emphasized in the undergraduate curriculum. It is important to instill these habits in students early on because these methods will not only allow them to properly give credit to the work of others but will also inform them as to how to claim credit for their own future work.

By incorporating open source hardware education into both the formal curriculum taught in the classroom as well as the hands-on experience obtained in the lab, students will be made aware of open source best practices early in their academic careers and can incorporate these ideas into their future projects. In this way, educators are not only changing how science is done in their group but also how the entire scientific community operates. Along the way, they are giving the gift of "forever" tools and knowledge to their students.

3 Facilitate Teamwork

As a PI, it is imperative to emphasize teamwork on a micro- and a macro-scale. On a micro-scale, teamwork within an individual research group allows young researchers to pool information from the various group members and problem-solve effectively without constantly having to consult a PI. Our own students expressed the skills that working as a team taught them:

> "*Working on this project has reinforced my mindset that if all team members are invested in a group project teamwork is much more productive than individual work. Though manual tasks can be easily completed alone, it is much more efficient if the work is split evenly...Working with other people that lack prerequisite knowledge creates the opportunity for team members to divide topics to learn and teach them to each other in an accessible way that assumes no prior knowledge.*"

Teamwork in an open source setting should be especially emphasized because the nature of open source science is inherently that of teamwork. Open source hardware allows for larger teams than previously imaginable, with members spanning multiple countries, time zones, levels of experience, etc. It is entirely possible that a contributor in your project does not speak your native language or does not have a formal scientific background.

Because of this, *it is important to treat your own research group as a microcosm of this type of larger collaboration.* Members should practice respectful communication habits when dealing with different ideas. Platforms like GitHub have a community code of conduct to maintain respectful discourse between collaborators. PIs may find it useful to have a similar code of conduct for their research group.

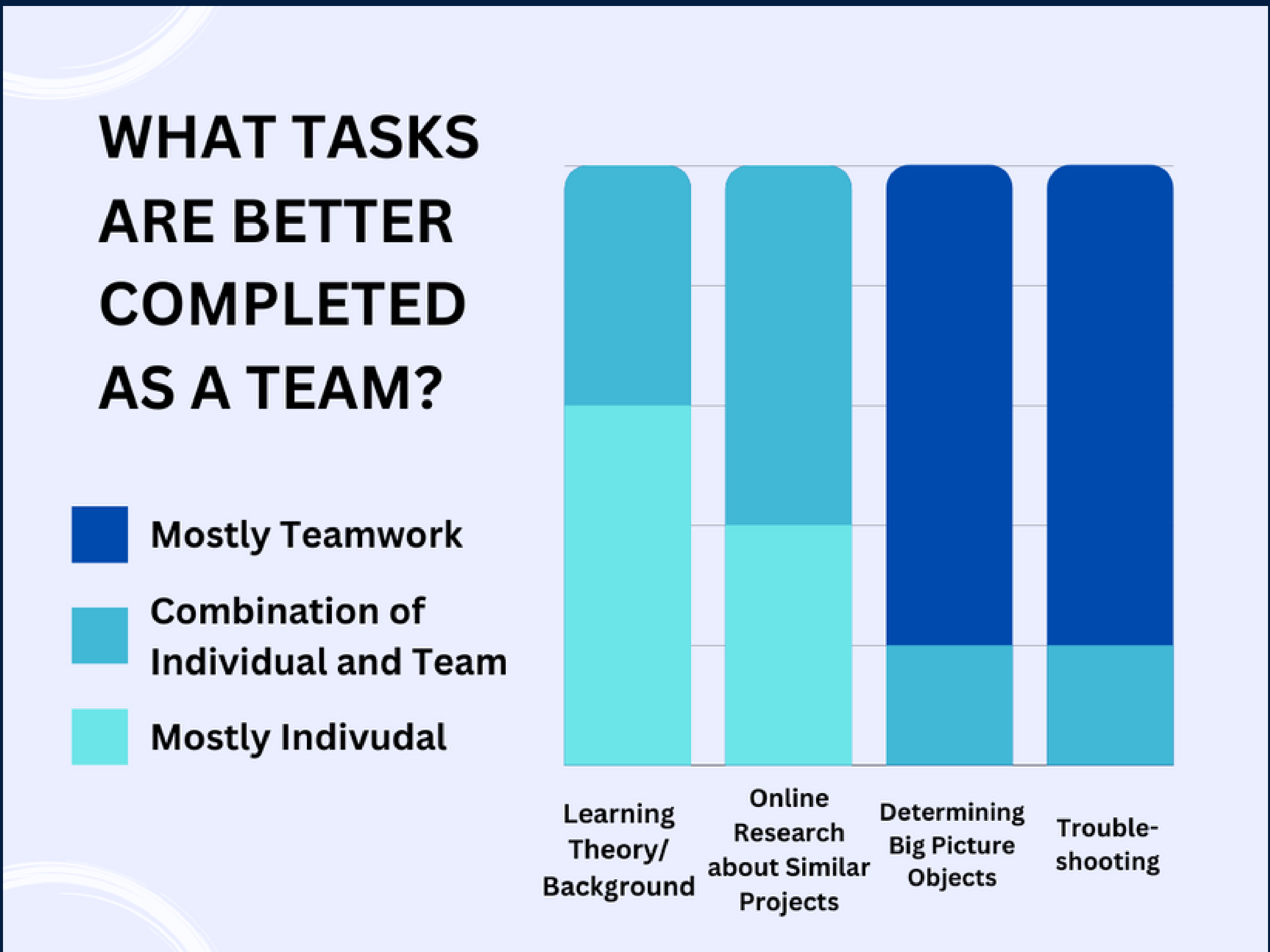
In conjunction with healthy communication strategies between team members, it is important for all members of your team to feel that their ideas are valued. Due to the aforementioned diversity in educational background and experience of the large open source hardware community, team members ought to be cognizant of and respectful of varying skill levels in their own smaller team.

In our own student teams, the members varied in age, major, and educational background. One student expressed how working with such a team improved his perspective on teamwork as a whole:

> "Working on an open source project has showed me the different ideas and perspectives different people on a team bring to the table. My previous experience informed my work by showing me that each person on a team usually has a specific talent or area of expertise that they can use to benefit the group."

Hierarchies are very present in academia and can be very discouraging to those just starting out as falling at the bottom of the totem pole means that one's ideas are often dismissed in favor of those with more experience. The job of a PI is to act as a mentor to encourage further education rather than to hold absolute authority. If students practice treating team members with respect regardless of skill level or educational background, they will be more equipped to handle the diverse audience of open source science.

While the aforementioned benefits of teamwork are invaluable part of working on an open source project, it should be acknowledged that teamwork is not necessarily suitable for all types of tasks. We surveyed our team on the types of tasks they prefer to complete as individuals or as a team, and the figure below shows the results.



The consensus was that tasks such as "Learning Theory" and "Online Research about Similar Projects" were best completed alone, while those such as "Determining Big Picture Objectives" and "Troubleshooting" were best completed as a team. It is important to be mindful of which activities require teamwork when assigning a task to the group.

Further, teamwork is often more productive when it is organized. When individuals are split into smaller groups, each assigned with specific tasks to complete, individuals within that group are more likely to be accountable to do their share. One of our students shared their frustrations about the fact that work was not evenly distributed across the group.

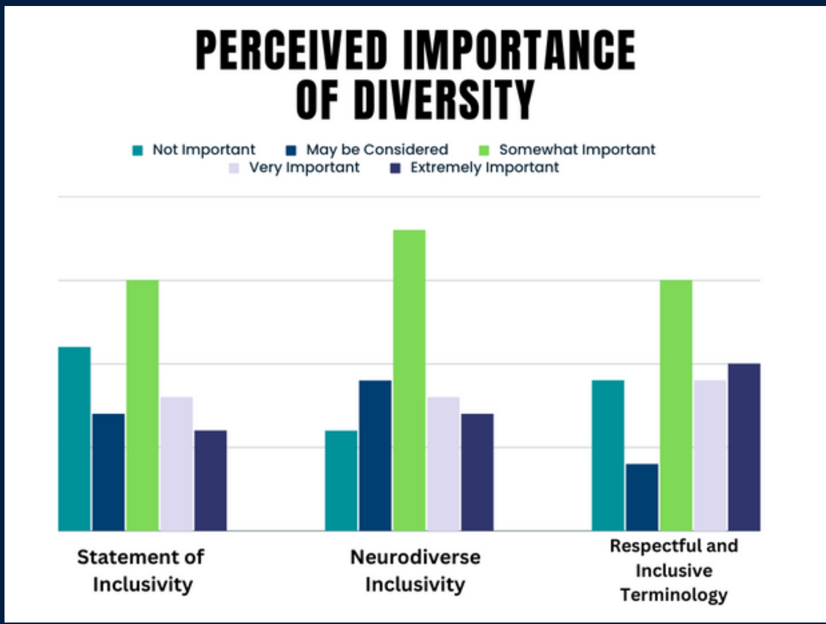
> *"For team work to be successful, everyone has to contribute their share. While this was the case in the beginning of our project, the distribution of work fell too heavily on 2 or 3 members of the group as time went on...I think our team could have been more successful if more people contributed their share."*

As expressed by this student, teamwork can often lead to some group members taking on more work than others. In order to mitigate the chances of this happening, the division of work between group members should be agreed upon beforehand, and the PI should facilitate active communication between team meetings, such as weekly check-ins.

The educational aspect of an open source project lies not only in scientific knowledge but in the skills that it requires to be a scientist. Effective teamwork is one of the skills that is indispensable in any field, but especially in academia. No research can be done alone. Even the smallest of research groups relies on work that was previously done in the field. By teaching students techniques for effective teamwork, PIs set them up for success in future endeavors.

4 Promote Diversity

Documentation should be written in a manner that promotes diversity within academia. The results from our surveys show that diversity is currently not made a priority within academic circles. When asked about the importance of including a statement of inclusivity in the project documentation, a majority of respondents answered "somewhat important," "may be considered," or "not important." Other questions regarding diversity, such as inclusivity when it comes to terminology and neurodiversity yielded similar results, as shown below.



The lack of priority afforded to diversity is a majority weakness within the academic community. By definition, diversity means the inclusion of more people. Thus, promoting diversity serves to make scientific projects accessible to a greater number of people. Promoting diversity in scientific work through changes such as adding a statement of inclusivity serves to appeal to a broader range of audiences thus increasing the awareness of Open Source Hardware as a whole.

The actual process of promoting diversity can take many different forms. The endeavor of open source science itself inherently promotes a certain level of diversity within academia by allowing greater access to educational and scientific resources. Open source projects live entirely online, and thus anyone with internet access can use and learn from them. Consequently, simply making a project open source does contribute to diversity.

Naturally, open source projects are going to come in gradations of difficulty. A *designator for difficulty level* and learning curve should be added to open source projects as they allow users to select a project closest to their skill level. Such can increase accessibility of open source hardware and appeal to a more diverse demographic.

Going further, however, diversity can be promoted by actively seeking out a wide range of backgrounds, perspectives, and experiences when selecting members of an investigative group. PIs have the opportunity to create a community where all team members feel that their diverse backgrounds profoundly contribute to the ways science is done and are integral parts of a successful project. A collaboration that prioritizes diversity of every kind—racial, socio-economical, neurological, etc—will create a thriving environment free of prejudices where all feel and experience equal access and can freely and fully participate in open source hardware projects and science in general. To this end, it is crucial to develop frameworks and set goals that are in line with the values of diversity, equity, inclusion, and justice. The statement from OSHWA, shown below, is a good starting point for implementing these values:

OSHWA DEI+J STATEMENT

Any endeavour related to creating and sharing knowledge and creative work is stronger the more diverse it is. Open source hardware grows stronger from a diverse and inclusive community, and open source hardware fosters increased diversity, equity, inclusion and justice by making hardware more approachable, and easier to get started in. By reducing barriers that disproportionately prevent under-represented communities and individuals from becoming involved in hardware, open source hardware is a positive force towards equity and inclusion.

5 Write Accessible Documentation

Open source hardware ought to be written for understanding at the undergraduate level, which can only take place if collaborators know that they are working on an open hardware project from inception. According to our survey results, many professionals are skeptical of the usefulness of open source hardware projects. One respondent wrote:

> *“I guess the biggest barrier is probably that the hardware created inside the collaboration has a very specific and high-end purpose, so it is generally very specialized, expensive and so extreme in performance that it requires a high level of expertise to be manufactured, assembled and used. All things that don't go well with widespread adoption by the general public.*

Another scientist responded:

> *"Having the time (that we don't), we should create "simplified" versions for the general public, the same way as we write "science summaries" to accompany scientific papers.”*

Others highlighted the lack of (perceived) availability of open hardware projects, lack of interest, and small ecosystem.

Further, less than half of the respondents reported being aware that the hardware documentation being written would be made public. When asked about the target audience for open source hardware, the most frequently cited primary audience was other scientists and engineers. These results suggests that academics within large collaborations may be writing documentation with the impression that such documents are crafted solely for internal use or at most for use by other professionals in the field. The availability and applicability of hardware documentation to the entry level student or the general public is greatly de-emphasized. Therefore, we make a recommendation that documentation ought to be written at the level to be understood by an undergraduate student. *Undergraduates represent a bridge between the academic world and the general public.*

Different Learning Styles

Additionally, documentation should strive to accommodate different styles of learning. In all subjects, there are bound to be different types of learners, and hardware is no different.

In the "free response" survey question in which respondents were asked about the threshold of knowledge necessary to translate a design into a physical object, several of the responses stressed the importance of both theoretical and practical knowledge. For example, one respondent answered,

> *"I learned about the to-do list during the job, but I was exposed to a more generalized walk-through of how different parts of the software connected to the different parts of the hardware, which gave me a more intuitive understanding of what my job aims to do."*

As expressed by this respondent, documentation should be written in a way that is beneficial to both hands-on learners and those who prefer the theoretical.

Simplified Versions of Projects

If creators feel as if their project is too complex for the average hobbyist or student to gain any meaningful use out of it, we recommend the suggestion given by the survey respondent, which is to create less technical or 'simplified' versions of projects that they can make open source. If documentation can be simplified to be accessible to undergraduate students, it is likely to be both accessible to hobbyists, companies, and the general public, but also contains specific technical elements that will be useful to others in academia.

While we have written this guide based on feedback from our own students, the mentoring tips provided here are just the beginning. True learning comes from implementation in the classroom and the research lab. We encourage you to use the information provided here as a jumping off point to kickstart a journey into open source hardware.

**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 (Columbia university in the City of new York) the Open Source Hardware 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