8 How Deep Is Your Love? On Open-Source Hardware

David Cuartielles

This chapter looks at the history of open-source hardware design from the viewpoint of one of the creators of the Arduino project. Arduino is an open-source hardware platform that can be programmed from free IDE (Integrated Development Environment) software and comes with Creative Commons licensed documentation. According to many observers, it has changed how many people think about the way products can be licensed in order to reach the market.

Half academic experiment and half product, Arduino keeps on challenging business models around hardware as well as the fields touched by it, including educational tools and connected objects (the so-called Internet of Things).

Arduino grew from a five-man operation into a small multinational corporation with offices in Turin, Lugano, Malmö, Bangalore, and Taipei, and with representatives in San Francisco, New York, and Boston. It now employs 35 people dedicated to the creation of educational experiences in the world of electronics.

What the GNU[1] movement and its open-source counterpart[2] brought to the world of computation is a new type of thinking about how to share knowledge. The sharing happens thanks to the existence of legal frameworks that allow the creators to pass their work to others who can then build upon it. Arduino has used several different licensing models, mostly because it was one of the first projects to have a need to license physical objects and because when it was founded (in 2005) there was no proper licensing model for hardware.

This chapter is based on personal notes on events the author attended. It is written as the four acts of a love story (denial, white lies, open marriage, love) in which different non-fictional characters introduce the "drama of open hardware."

Everything starts at the Open Hardware Summit 2012 (OHS12) in New York, an event that gathered people from all around the world to discuss the state of openness of design tools such as 3D printers, microcontroller boards, and blueprints for workshop machinery.

Denial

Talk about the overall state of affairs, but still do not introduce the conflict, this is all about setting up the scene, who was there, what they were doing, what they were known for, how they relate to the others. … No sex, strictly business, so leave gossip aside (even if that is the juicy part).

Cuartielles, notes

It is September 2012, and the second Open Hardware Summit is about to begin. There is a crowd of people willing to listen to their open-hardware heroes talk about their findings. The event had almost sold out before opening its doors, and I am not sure I will be able to get a ticket. Dave Mellis, my friend since we first met at Ivrea in 2005 and Arduino's co-founder, coordinated the peer-review committee for the event and surely had a ticket. This year, all I did was review a couple of submissions for talks, so I wasn't on the free-entrance list. I was far too busy throughout the summer writing a book about embedded electronics connected to phones, and forgot to purchase a ticket.

Luckily, on the way to Eyebeam, where the OHS was taking place, Dave and I met up with Bre Pettis. Bre, known for being one of the founders of the hackerspace NYC Resistor and of Makerbot, a 3D printing company, looks tired; he has had an exhausting week dealing with the launching of Makerbot's latest printer and software. He was featured on the cover of *Wired*—in the geek culture, equivalent to being on the cover of *Time*. As one of the event's sponsors, Bre's company has two free tickets, and his partner can't make it. Bre and I first met a couple of years ago, and probably have spoken four times since then. He offers me his other ticket.

One of the nice things about the OHS is that there is no VIP queue—everyone has to stand in the same queue to enter the event. I stand there with Dave and Bre, and see familiar faces all around. I see Ayah from littleBits and Mary, a former student of mine at CIID.[3] Michael from Arduino SF and Katia from Turin are already inside. I recognize some people from Dave's MIT crowd carrying heavy suitcases; they will demonstrate their new fablab tools, he says. Bre hands me his second badge and I enter Eyebeam.

It was five years ago that I first visited Eyebeam. I asked for permission to visit the space where the artists in residence developed their projects. H. C. Steiner, maintainer of PureData (Steiner et al. 2013) and my host in New York on that trip, was a resident there. It was through him that I got a quick introduction to the space and got to know some people there. One of them was Ayah Bdeir, then a recent MIT graduate interested in using microcontrollers in education and now an entrepreneur selling littleBits, a small electronics kit from which children can quickly build devices (Bdeir 2012). Her innovation is a Magsafe-like[4] connector for small circuits. Out of that simple idea, Ayah created a company that has been featured all over the media; her story has been told as an example of the "American Dream." Ayah was also behind the first OHS in 2010, which she initiated with Alicia Gibb, the current president of the Open Source Hardware Association.

I first got to know Alicia back in 2009, when she got in touch with the Arduino Team (this is how the core founders of the Arduino platform are known as a group) through e-mail. She was looking for some information about how people used the Arduino platform. Alicia was in the middle of writing her MSc in Theory, Criticism, and History of Art, Design and Architecture at Pratt Institute (Gibb 2010). That thesis was presented in 2010 in conjunction with an exhibition. Alicia teamed up with Ayah in promoting the Open Hardware Summit immediately after. It started as a meeting place for individuals, institutions, and companies to discuss openness in the creation of physical goods.

Alicia's thesis documented the state of the art of Arduino at many educational institutions. A tool created by us (the Arduino Team) to help our students to enter the world of embedded electronics had spread by word of mouth to most of the world's universities. I find some of the cases that Alicia mentioned in her thesis interesting. On page 53 there is a reference to DMDuino, an Arduino compatible board, produced by Professor Kim's students at Hongik University. This development happened in 2007, when I was teaching at another Korean design school. As a matter of fact, I gave a couple of talks for Professor Kim's students while I was living in Seoul, and one of those talks sparked the process of making that particular board.

Educational and recreational technologies took off in the 1980s. I am lucky to be among the first Spaniards who got access to computers back then. I don't recall Spain as having a lot of people who would build their own computers; however, there was a big interest in augmenting them and building circuits that would be controlled with early PCs.

As the PCs became more specialized and harder to access, the interest in "homebrew" electronics decreased. In the city of Zaragoza, where I come from, we jumped from having several small repair shops where it was possible to buy discrete electronic components to having only one. At that time, Zaragoza had three engineering schools—all of them offering courses in electrical engineering—with more than 6,000 students in all.

The small electronics stores probably were killed off by the arrival of the Internet and the engineers' early adoption of it for purchasing purposes, in conjunction with the arrival of cheaper consumer products that people tended to throw away rather than get them fixed. The electronics stores transformed into consumer electronics stores as the hobbyist engineers faded away.

When Arduino came along, there was no market for it. It was not intended for a market, either; it was intended to introduce design and art students to electronics in an easy way. We didn't anticipate its potential; we just wanted something cheap and useful to give an introduction to digital electronics. Those two characteristics were precisely what gave Arduino the reach it has today.

There was one person who anticipated the potential not just of Arduino, but of the maker community in general: Chris Anderson, then the main editor of *Wired*. The hacker movement had brought the concept of the more political hacklabs and that of the less political hackerspaces or makerspaces into popular culture. Those concepts somehow led Anderson to foresee that the maker community could eventually enter into a different category, and that makers were going to need tools and guidance. Anderson recognized that the maker movement had the potential to have an effect similar to that of the arrival of the PC in the 1980s.

Chris, a great speaker, is often invited to speak about innovation, technology, and other subjects. At OHS12, he gave a keynote address titled "Microeconomics for Makers: Business Models for the New Industrial Revolution" (Anderson 2012a).

OHS12 happened in September 2012, just a couple of days before the World Maker Faire in New York (WMFNY), an event arranged by *Make*, a magazine dedicated to the maker movement. The two previous editions of OSH took place on WMFNY's grounds, at the New York Hall of Science in Queens. However, the 2012 edition moved location in an attempt to detach the events from each other.

The September 2012 issue of *Wired* had a photo of Bre Pettis and, inside, an article on Makerbot—the 3D printer Bre and his team had developed—written by Chris Anderson himself (Anderson 2012b).

White Lies

this paragraph is about the illusion of open source, about people building things open for a while, long enough to get traction, and how they then discover that the market is a bitch and they want out, because it is hard to accept that things are the way they are and radical openness is tough. Not even Arduino is radically open

Cuartielles, notes

In the spring of 2007 I was an artist in residence at the Hangar in Barcelona. For three months I lived in a rental near the Hangar. I didn't manage to do any of the art works I had planned for the period. Arduino had become popular with artists by then, and I had to respond many requests for help and many invitations to give a talk or a course.

In a talk I gave at Dorkbot Barcelona, I made my standard presentation of the period: I spoke about the robot a Chinese student hacked using the first self-made Arduino in 2005, how the Involuntary Dance Machine project from Malmö emulated Stelarc's early works about body control at a fraction of the development time, how the boards changed to accommodate users' needs, and so on. At the end of the talk, a highly intoxicated young man approached me. At the time, I wasn't used to the idea of fandom. Why should anyone care about the things we do beyond the fact that they are useful? The young man asked a question that has followed me ever since: "How does it feel to be making something like this, something people look into? Don't you feel lonely when you are doing these things?"

That question happened to be extremely insightful. You might think that creating something like Arduino for a community of users is a pure participatory design (PD) process; that as part of an open-source community, you are doing everything to help your users; that they will be loyal to the brand you are building; that they will participate in the creative process—but in reality there have been situations and external influences that could have forced us to make decisions that would have compromised the basic ideology of our project.

This is the white lie of open source when it is brought into the mainstream. You cannot tell people that you will have to make design decisions in the end. Some of those might be really small ones: the distance between two connectors, the color of the silkscreen on a PCB,[5] the capacitors from this or that brand. Some others might be of great importance: the family of processors used to run the code, the way menus are structured on the UI, or the license for the design files. Eventually someone has to make these decisions, or else the conversations on the forum[6] will go on forever. The Arduino participation model operates from the point of view that it is impossible making 100 percent of the users happy.

Therefore, the concept of radical openness is just an illusion. There is no real democratic process in the making of many of the open-source projects anyway. You will find many projects where people have to take the role of benevolent dictators, as described by Eric Raymond in his essay "Homesteading the Noosphere" (Raymond 2000).

But there is yet another white lie, and this one goes deeper under the skin of the open-source movement. The fact is that open source is a trend. From bikes to computers, passing by utensils and patters for making clothes, there is an overwhelming amount of open projects showing up every week. People make open "stuff," sometimes out of good intentions, sometimes out of a willingness to gain recognition. "Open" is a buzzword, and many don't even understand the complexity behind the legal frameworks needed to make things open.

The One Laptop Per Child project (OLPC 2013) is trying to make an open-source computer running open-source software to provide access to open knowledge. OLPC's lobbyists[7] pitched the idea to whole countries and offered making millions of computers at a fraction of the money spent in books by the average kid. The hidden message behind this is that openness is good because of being cheap. People have a tendency to buy into this argument.

Where the second white lie resides is in thinking that open source is always making things cheaper. Take the maintenance of the software platform for a region as an example. There is the option to buy from a vendor like Microsoft or there is the option of hiring a local developer expert in the free Linux operating system to run the region's servers. In both cases there is an economic cost: either you pay for a pre-made system, or you pay for someone to tailor the software to your needs. The main difference is that a good open-source developer will write the code in such a way that it will be possible for other developers to reuse the code and maintain and enlarge the system.

The money needs to be spent anyway. The equation of cost needs to be analyzed every time; there is no magic formula.

Cost and openness, what matters the most? This is the issue here. If someone were giving us a machine to solve a task that was cheap enough, we would probably not think about making one ourselves. This is the basic principle of capitalism: we have learned to value our time, and we apply the simple equation of how much our time costs and what is the probability of successfully building something by ourselves. If the probability is low and the cost is high, we will never try. The price has to be right. Price seems to matter at least as much as openness.

Chris Anderson knows that openness is not really open but is trendy, and that openness doesn't necessarily make things cheaper. So he decided to step into the open-source hardware movement by exploiting one of the communities he had helped creating: DIY Drones. Chris created two different relevant communities on the side of his career as editor in chief of *Wired*: DIY Drones (Anderson 2008) and Geekdad (Anderson and Denmead 2013). At some point he decided to stick to DIY Drones, I guess because it was more rewarding both personally and economically.

Chris managed to grow a business around the idea of unmanned aerial vehicles and created the company 3D Robotics. On April 15, 2013, the company's website (Anderson and Munoz 2013) said "3D Robotics provides fully-autonomous aircraft and open-source UAV[8] technology that deliver professional performance at amazingly affordable prices." This clearly states both aspects I am trying to stress: cheap and open are elements of the present-day sales pitch. Many people think that because of being open and building on top of other projects, it has to be cheaper because it requires a smaller research-and-development effort.

3D Robotics sells all the parts needed to make a flying vehicle, but also sells fully assembled machines. The main component within 3D Robotics' UAVs is the so-called Ardupilot, an Arduino-compatible board that can run a variety of motors and can read data from onboard sensors needed for balancing and locating the flying vehicles it commands. It is the brain for all of their products. 3D Robotics is programmed using the software designed and maintained by Arduino. 3D Robotics maintains a special firmware for the Ardupilot boards, but doesn't need to be supporting the UI to program that firmware.

In 2012, I was invited to appear at a conference called Tijuana Innovadora (Cuartielles et al. 2012). I was to take part in a panel discussion with the main firmware maintainer for the Ardupilot, Jason Short. I didn't know then which was the relationship between Jason, Ardupilot, and Tijuana. The whole story wouldn't take long to unfold. Tijuana Innovadora is an initiative from a city benefactor trying to portrait the city and the region around Tijuana as a reliable business hub. The trade agreements between the U.S. and the northern states of Mexico allow U.S. companies to manufacture for lower wages (as well as with lower manufacturing regulations) at factories located on Mexican ground. 3D Robotics is one of those. And the person responsible for 3D Robotics' *maquila*[9] is a young Mexican entrepreneur who tries to spread the open-source hardware culture among the local colleges. He suggested that Tijuana Innovadora 2012 have a panel on open-source hardware, where I spoke and where I met Jason.

At some point, we were invited to visit 3D Robotics' factory in Tijuana, which is where they manufacture the Ardupilot boards at a pace of 200 per week.[10] They are also assembling the full UAVs there, and serve them to the whole world via courier. It is a convenient location, as 3D Robotics has its headquarters in San Diego, California. All of these facts were clearly stated at 3D Robotics' website.

Yet again, I knew 3D Robotics was manufacturing in Mexico, not just because of the website. Just about a month earlier, while in New York visiting the OHS12, I happened to step outside the conference building to catch some air. I accidentally bumped into Chris Anderson and Ayah Bdeir, who were outside talking about their businesses. Ayah was worried about the quality of Chinese manufacturing. Chris was pitching Mexican professionalism and how much closer Mexico is to the United States than China. My only proof that that conversation happened is a picture I took of the three of us together (figure 8.1).

People tend to identify open source with "good will" and even with "ethical hacking." However, I find Eric Raymond's analysis much better:

> Not until the Linux explosion of early 1993–1994 did pragmatism find a real power base. Although Linus Torvalds never made a point of opposing RMS [Richard Stallman], he set an example by looking benignly on the growth of a commercial Linux industry, by publicly endorsing the use of high-quality commercial software for specific tasks, and by gently deriding the more purist and fanatical elements in the culture.
>
> A side effect of the rapid growth of Linux was the induction of a large number of new hackers for which Linux was their primary loyalty and the [Free Software Foundation]'s agenda primarily of historical interest. Though the newer wave of Linux hackers might describe the system as "the choice of a GNU generation," most tended to emulate Torvalds more than Stallman.
>
> Increasingly it was the anticommercial purists who found themselves in a minority.

Open Source Hardware (OSH), the emergent field the Open Hardware Summit (OHS) is all about, unveiled the white lies to all of us in its 2012 edition. Chris Anderson's keynote speech was about how to run a business selling open-source technology. He introduced his vision of hybrid open-closed systems in order to keep the cloning of his company's original designs to a minimum. He found out what we—in the OSH business—had already discovered a long time ago: It is easy to clone our designs. The difference Chris was introducing was that it should not be possible to copy part of the designs, as he and his partner at 3D Robotics decided to make part of their designs closed to protect their IP.[11]

I wondered how much Chris had been influenced by the interview he had conducted with Bre Pettis just a couple of months before, the one that appeared in *Wired* just as OHS happened in New York. I bet that besides his personal experience dealing with IP protection, he had gained some understanding on how Bre and his partners had decided to run Makerbot Industries.

Figure 8.1
At OHS12: C. Anderson (left), D. Cuartielles (center), and A. Bdeir (right). 2012 David Cuartielles (CC:BY-NC).

Bre spoke later that same day at the conference in New York. His news was not just that Makerbot had released a new 3D printer and a new software to generate the GCode[12] for the printers; in addition, Makerbot, which had grown as an open-source business, was now going to close part of its IP. It capitalized on the contributions coming from hundreds of users and competitors that shared software, iterated the hardware and published their results online. Makerbot was highly regarded by a huge community of users and had among its original investors Adrian Bowyer, who according to Wikipedia (2013) had “spent twenty-two years as a lecturer then senior lecturer in the Mechanical Engineering Department at the University of Bath,” had “retired from academic life in 2012,” and had “invented the RepRap Project—an open-source self-replicating 3D printer.”

I had given a talk at the Technical University of Denmark in 2011. Adrian and I had given a keynote presentation about open technologies titled “Creating wealth while giving it away” (Bowyer and Cuartielles 2011). It was then that I had heard from Adrian himself that he was an investor in Makerbot. After my initial surprise, I came to realize that this was an interesting situation. The creator of something as relevant as a self-replicating machine was an investor in a company that was capitalizing on his creation. I saw it as some sort of poetic justice.

Fast forward to 2012. Bre Pettis, giving his presentation at OHS12, was about to break into tears. He had had a tough week. Part of the Makerbot concept had gone closed-source. And now, at the open-source conference, everybody was expecting to hear Bre explain why Makerbot wouldn’t release the blueprints to the mechanical construction of its latest printer and wouldn’t release the source to the application to work with the 3D models and generate the GCode.

Bre’s explanation was simple: they had invested a lot of time and money in creating their hardware and software, and they didn’t want them to be cloned. This made a lot of people unhappy. Among other contributions to the open-source community, Makerbot Industries created Thingiverse (Makerbot 2013), an online repository of 3D shapes and designs for laser cutting, broadly used by Makerbot users but also by users of other 3D printing systems, mostly because there were no alternative free online services where to easily document their projects.

One of the Thingiverse users was a young Czech contributor to the RepRap project, Josef Prusa. His immediate response to Makerbot’s new policies was to stop using the repository:

> I’m leaving Thingiverse after seeing updated Terms of use thingiverse.com/legal, over next few days I will remove all my stuff. It will be downloadable on my website josefprusa.cz or reprap.org I prefer to be owner of my own designs:-)
>
> We are not trolls, as Raldrich said.
>
> The fact that the legal ramifications of MakerBot’s TOS weren’t discovered until today doesn’t magically give them a free pass.
>
> The fact that they don’t intend (today) to exercise the rights they’ve granted themselves also doesn’t magically give them a free pass. Companies change—take a look at their stance on Open Source Hardware.” (Prusa 2012)

Josef is known as the maker of the most cloned open-source 3D printer, the so-called Prusa-Mendel model (Prusa 2013). In his “Occupy Thingiverse” statement, he invited unhappy Thingiverse users to make bad prints of cubes and upload them as a way to protest against Makerbot’s policy shift. A cube is nothing but a three-dimensional shape that is used to check whether a 3D printer is properly calibrated. People within the printing community print many cubes to check if the printer can operate at a proper speed, if it responds to changes in the size of the prints, and so on.

One thing Josef quotes in his goodbye letter to Thingiverse is how the legal terms (Makerbot 2012) of the website weren't changed in September 2012, when Bre announced Makerbot's new designs, but were changed in February of that year. I have no proof of how these legal terms were before that, but it is clear that things had started to change for the users far before the big announcement. As a matter of fact, things might have been like that since the website was created years before, but people didn't realize until September that Makerbot kept the right to use the designs posted by users, at no cost, as long as they were hosted there. On top of that, they had the right to decide whether they could be there or not.

Consider the following quotation from the terms of service referenced by Makerbot (2012):

> You hereby represent and warrant that your User Content does not violate the Acceptable Use Policy (defined below). You may not state or imply that your User Content is in any way provided, sponsored or endorsed by Company [Makerbot Industries]. Because you alone are responsible for your User Content (and not Company), you may expose yourself to liability if, for example, your User Content violates the Acceptable Use Policy. Company is not obligated to backup any User Content and User Content may be deleted at anytime. You are solely responsible for creating backup copies of your User Content if you desire.

This is typical for a free-of-charge service; it amounts to "you can play in my playground as long as you follow my rules, whatever you do is your responsibility." A site like Thingiverse will, for example, avoid hosting pornographic pictures. On the other hand, this provoked an interesting reaction. Sites like Dongiverse showed up to cover a niche for 3D printing technologies: the sex market. At the time of writing, the official Dongiverse website seemed to be down. However, I found a reference to it in a blog dedicated to 3D printing:

> Almost all erotic gadgets have low-profiles on 3D model sharing sites. However, one website, launched three years ago, comes with an announcement—"to parody the awkward amount of suspiciously xxx objects uploaded daily to the popular 3D model warehouse Thingiverse." (3ders 2013)

A lot of people think this site is a joke. Dongiverse, a Thingiverse for dongs, has a very similar design as Thingiverse's old version. It is a site for sharing your designs (specially sex toys).

Thingiverse allows listing objects by collection. This allows checking whether there are objects tagged in a certain way. As the reference Rrix 2013 shows, it is possible to make a search, within Thingiverse's website, for the term 'dongiverse'. This search will list some objects that are specifically designed or can be perceived as sex toys. In other words, Makerbot might not be that strict in applying the policies they declare to be following in their terms of service.

But why the fuss then? Why does it matter that a company decides to close down its IP? In my opinion, Makerbot and 3D Robotics betrayed an image they built. They flirted with the open-source community and used it to build a customer base.

We had arrived at OHS12 believing in all the little white lies of open source. Within four hours, we heard Chris and Bre give up the idea of openness.

Arduino's designs are not radically open from a process viewpoint. When we made Arduino, we built upon years of accumulated experience in teaching electronics to beginners. We didn't sit down with users to create our first boards. We knew what needed to be done. The participatory design aspects of the creation had been done in the form of fieldwork by all of us. We decided back then that using expert knowledge about users, about technology, and about the relationship between users and technology should become our development method.

This became our design process: We go out in the field and try out a certain experiment with users. Most times, we come with a new type of board and a series of exercises for people to try. Once we get sure the design is not only functionally perfect, but also an interesting tool to use, we move into production. This means making thousands of boards to send to hundreds of distributors all over the world. Because we believe in the importance of sharing knowledge, Arduino releases the so-called reference designs (the hardware design files) to the boards under a Creative Commons license.

To have a radically open design process, we probably should start opening up during the conceptualization phase and not just when the final object is released. But in the same way that other entities decided to go for not opening their designs to their users for fear of being copied, we prefer to shortcut the IP problems by keeping things secret until the day they reach thousands of people at once.

Our designs will be cloned anyway, and we will have to enforce traditional IP protection (lawyers, cease-and-desist letters) for every new design we make. So why bother even trying to protect the IP in other ways? It is better to keep it open source for people to freely build upon it. As someone said, "you have to be ready to maintain open source," and we are ready to maintain our own reference designs, but not all the variations made by hundreds of people. That is a full-time job that we don't want.

Open as in open marriage until you turn into a yuppie

Arduino thinks about manufacturing in markets for their markets (the scale factor at low prices, a different paradigm in a changing global economy)

Arduino has users, not customers

Cuartielles, notes

Open is all about love and hate, about the lie that is kept for as long as possible until someone betrays the dream. What we have to ask our open-source idols is "How deep is your love (for openness)?"

It is easy to fall in love with the idea of people making something for the greater good, but how much of that is just a naive view of reality? How much are we living in denial, blind to the fact that things might work differently? Some lovers might show a face to you and lie, but most of the times you aren't ready to ask the right questions, mostly because you might not be willing to hear the truth.

Chris Anderson sent one of his journalists, Clive Thompson, to a small conference we attended at Potsdam University in 2008. The topic was the creation of Fritzing, a program that would allow people to jump from a prototype built on a breadboard to a PCB design that could be manufactured quickly (Fritzing 2013). Clive interviewed each member of the A-Team[13] and wrote his report (Thompson 2008) carefully.

I recall receiving an e-mail message from *Wired* cross-checking the facts of that interview several months later. *Wired* has long lead times—we were interviewed in Germany in June and the article came out in November. The publication of that article marked a "before" and an "after" in the history of Arduino.

The graph in figure 8.2 shows the number of people reaching the Arduino website, grouped by continent. To create that graph, I used all the historical data that had been stored in the Arduino.cc server between 2006 and 2011. What it shows is that Europe somehow got to know about Arduino thanks to the article in *Wired*. It is my understanding that all tech-related journalists in the world look at *Wired*, then write stories related to what they see there. In other words, Chris Anderson opened Europe to Arduino, even though we are a EU-centered entity in the first place.

Arduino, and articles like the one Chris Anderson commissioned in 2008 as editor of *Wired*, helped create the buzz around the OSH field, and once he saw an opportunity, he left everything and became an open-source entrepreneur. Sadly, at OHS12 he came to present his new approach to openness: closing things down.

When we all started this adventure of open-source hardware, there was a mantra we repeated many times: If they copy you, that means you are making something good. Chris didn't think that way any longer; he believed he needed something to differentiate his products from the ones from cloning factories around the world, and closing part of the design was his approach.

Bre, while on stage, introduced a similar idea. He spoke about how much Makerbot was being copied in China (he made explicit references to this aspect) and how the Makerbot partners felt they had to keep their IP protected by not sharing the source files with others.

Ayah, on the other hand, wasn't talking to Chris about copies when I met them outside the conference in New York. They were talking about quality in the manufacturing. Chris tried to convince her about how much more convenient the Mexicans were.

I have heard these words many times coming from different manufacturers and designers. Around the world, there are good manufacturers and bad ones, no matter where they come from. There are people who are ready to deliver on a tight schedule and with high quality, and there are people who aren't.

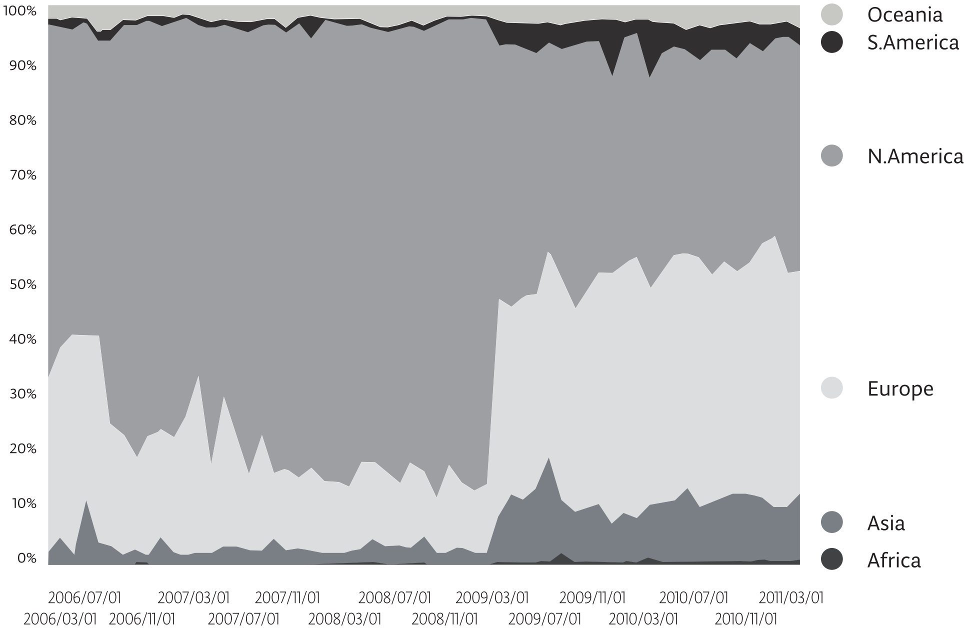

Figure 8.2
Users on the Arduino server by continent (percent, by month). David Cuartielles (CC:BY-NC).

I think that Ayah, in a very honorable way, was looking for a place to make things with the best relationship between quality and price, the same as Chris and the same as Bre. They figured out that to be able to keep that relationship, they had to stop people reproducing their designs and they had to build a narrative around it.

Since the open-source community is supposed to follow some ethical values, the narrative for OHS12 was “China is bad; we are being copied.” But this wasn’t news to us at Arduino—we have a person dedicated to hunt knock-offs from all over the world, and it is a fact that many of them come from China. However, we understood that protecting our brand image cost us the same if we were open source as if we weren’t.

We invited people to replicate our boards. We asked only that they not call their designs Arduino and that they not use our logo. This was why we put the blueprints to our reference designs online at no cost and under Creative Commons licenses. If you wanted to, you could take our design, send it to a factory, and order 1,000 boards, and you wouldn’t have to pay us anything.

Believe it or not, we still have to fight against counterfeiting. Today most of the Arduino boards are manufactured in Italy. About 43,000 units are produced per month.

That volume doesn't excuse going to China or Mexico to get things made. The expenses of keeping a team there to make sure our manufacturing requirements were met would make the boards too expensive.[14]

As an open-source project, we are ready to include modification proposals from our user community. This means we can never go into large-scale production, as that would imply not being able to modify the hardware design often. This is probably the biggest lesson we have learned so far about hardware: Don't increase production too much, because you want to be able to introduce modifications as you go. Therefore, the only way to keep prices low is to have a distributed factory structure, with manufacturing plants as close as possible to the potential users.

To spread the love, you need to be present in as many places as possible and you need to keep some balance between local and global production. I think that Arduino needs to grow in Asian, Latin American, and African markets at a local level, to be able of keeping the quality at a maximum and the price at a minimum adjusted to people's needs at different locations.

Love

> seems to me this is one of those cycles, like it happened before. Those that participated in making the first personal computers in the early 80s were making an experiment in openness, until money came into play and Apple and Commodore showed up, then all the others followed. … It's like history repeating.
>
> Cuartielles, notes

In 2005, at the Interaction Design Institute Ivrea, we had the vision that making a small prototyping platform aimed at designers would help them getting a better understanding of technology. We thought about our own interaction design students at Ivrea and at Malmö University and about their professional future. They should be able to talk to engineers about their ideation processes, build prototypes in record time, or create installations to display a concept to a certain audience.

We shared our views on what we thought would work, based on our life experiences as technology users and teaching others for several years. Putting the words into action was very natural, probably because of the empowerment one experiences when meeting like-minded people.

Arduino's growth has to be understood as part of an emergent interest in technology at a global level that circulates around the idea of co-creation throughout online knowledge exchange. There are multiple tools out there, but here are the ones I believe are interesting from a community-building point of view:

- PureData, an open-source tool for studying sound (Steiner et al. 2013)
- Processing, a tool for learning about animation (Reas and Fry 2013)

- the RepRap 3D printing movement (Bowyer et al. 2013)
- Arduino, the first successful open hardware project.

I have been involved with or in close contact with three of these (PureData, Processing, and Arduino). I see some similarities in them, but also some differences. When your goal is making the best (for education, society, the world), there isn't a pre-defined roadmap you can follow. To me, however, the process matters as much as the result. The debate and the inclusion of opinions during the making are very important. Keeping loyal to the dream, in this case openness, is also important.

If these four projects have something in common, it is the participation of users in the making, maintaining, and growth of the project's original idea into whatever it turns into. Both the do-it-yourself movement and its values are somehow a natural part of the current software and hardware culture.

The process of making prototyping boards has been like this since Motorola and Commodore introduced the first microcontrollers. The book *On the Edge* (Bagnall 2005) describes how Commodore, a company dedicated to the production of calculators, created the first single-board computer, called KIM-1, back in 1976. KIM-1 ran a processor priced at 25 U.S. dollars—only 10 percent as much as any of its competitors in the market. KIM-1 was a prototyping board that would allow engineers trying out the capabilities of the 6502 processor made by CMOS Technologies.

This board and its successors were the foundation of the first Apple computer, which was prototyped and built on Commodore technology. Since both Commodore and Apple computers had the same architecture, they could both run the same type of software. In this case, both would run Basic, created by Microsoft in the 1970s.

Despite the difference in the value of money between 1976 and 2005, we priced Arduino at a very similar level. This was a coincidence, as I doubt any of us knew anything about the initial value of Commodore's processor.

Besides the price, Arduino was similar to the KIM-1 prototyping platform in other ways. It is an open tool that can be used in building other systems. Those systems might even become competitors of the original platform, as Apple did. Arduino is the starting point for many competing projects, among them Maple (Leaflabs 2013), Freeduino (Freeduino 2012), Netduino (Netduino 2013), and Teensy (Badger 2013). Because we use open discussion tools, some of the developers of the other platforms have joined the discussion about the kind of features our system should include, as it will make it easier for them to build upon our system and to create new products.

What we see here is that before the idea of openness in software, there was an openness in hardware platforms. As the development kits are a business by themselves, there are companies that have no other choice but to make their IP available for others to use and build upon. In some cases (for example, that of Commodore, which had a hybrid model of tools and products) it resulted in the disappearance of a company; in other cases, a company was able to become much more successful.

For some reason, people like Chris and Bre don't realize that they are making the same mistakes that others made in the past. They are the Commodores of the 21st century. Let us be something else. Let us be open source.

Epilogue

A long time has passed since I began writing this chapter. I wanted to spark a discussion about the real value of openness. I hoped that my arguments would be heard by those I mentioned, and that there would be a chance to debate the steps they were taking before it was too late. However, the format of OHS12 didn't allow for discussion, and there was no time for questions directly after the presentations.

My initial idea was to persuade Bre and Chris that there was no reason to close the designs they were making, especially because they had built their business around open source and the idea of spending too many resources in protecting their IP instead of looking at the community needs and proposals to support their growth.

Almost ten months after I started writing, Chris went ahead with his plan of closing down some of the designs from 3D Robotics. He keeps on supporting his thesis of "hybrid business models" with open and closed parts put together.

Just a couple of days before I finished writing this, Makerbot was acquired by a traditional player in the 3D printing business called Stratasys (Denison 2013) in a transaction that involved 400 million U.S. dollars' worth of cash and stocks. In his article regarding the deal, D. C. Denison managed to get in touch with Adrian Bowyer, the creator of the RepRap project, who manifested his happiness for the deal, as he got a big return for his investment in the Makerbot company.

Business as usual, for everyone.

Notes

[1] GNU is the name of a project initiated by Richard Stallman to create a completely free operating system. It was first announced in September 1983.

[2] The two main views within the world of source-accessible code are free software and open-source software. The former was started by Richard Stallman and suggests that once a piece of code is licensed under the so-called GPL license, all the code produced using it will inherit that same license and will therefore have to be open. The latter embraces a more business-oriented viewpoint and allows people building "closed" applications on top of open blocks of code.

[3] Copenhagen Institute for Interaction Design.

[4] Magsafe is a brand registered by Apple Inc. and used to label the power connectors used in the most recent family of Macintosh computers. It uses magnets to get the connector to span to the computer in an easy way.

5. PCB stands for Printed Circuit Board. In more colloquial terms, it is how you would talk about a naked circuit before populating it with electronic components.

6. Arduino's main feedback system is an online forum. As of June 28, 2013, it had 140,000 registered users.

7. OLPC's founder, Nicholas Negroponte, was once the director of MIT's Media Lab. He used his fame and influence to pursue this mission. His initial goal—never attained—was to make an open-source computer for $100.

8. UAV stands for Unmanned Aerial Vehicle.

9. A *Maquila* is a factory located in northern Mexico and manufacturing mostly for the U.S. market.

10. I was told this by my guide at 3D Robotics' Tijuana factory. He also mentioned the intention of keeping their machinery running around the clock to triple the production capabilities at some point.

11. IP stands for *intellectual property*, a term people use to refer to the designs, inventions, processes, and services they create.

12. GCode is a markup language used to represent the movements of the head of a 3D printer or a CNC machine.

13. A-Team: the Arduino Team.

14. At the time of writing, Arduino was about to launch the Yun, the first official Arduino board designed and manufactured in China. The initial batch was to number 5,000.

References

3ders. 2013. Description of Dongiverse (http://www.3ders.org/articles/20130128-falling-cost-of-3d-printers-leads-to-rise-in-personalised-sex-toys.html).

Anderson, C. 2008. About page from the DIY Drones community: The DIY Drones Mission (aka The Five Rules) (http://diydrones.com/profiles/blog/show?id=705844:BlogPost:17789).

Anderson, C. 2012a. Microeconomics for Makers: Business Models for the New Industrial Revolution. Talk given at Open Source Hardware Summit, New York.

Anderson, Chris. 2012b. The New MakerBot Replicator Might Just Change Your World (http://www.wired.com/design/2012/09/how-makerbots-replicator2-will-launch-era-of-desktop-manufacturing/).

Anderson, C., and K. Denmead. 2013. Geekdad blog (http://geekdad.com/2013/03/the-geekdads-big-adventure/).

Anderson, C., and J. Munoz. 2013. 3D Robotics, UAV Technology (http://3drobotics.com/).

Badger, P. 2013. Teensy board (http://www.pjrc.com/teensy/index.html).

Bagnall, Brian. 2005. *On the Edge: The Spectacular Rise and Fall of Commodore*. Variant.

Bdeir, A. 2012. Website of littleBits (http://littlebits.cc).

Bowyer, A., et al. 2013. Website of RepRap project (http://reprap.org).

Bowyer, A., and D. Cuartielles. 2011. Creating wealth while giving it away (http://www.youtube.com/watch?v=2j84BDoNkt8).

Cuartielles, D., et al. 2012. Panelists at Tijuana Innovadora (http:// tijuanainnovadora.com).

Denison, D. C. 2013. Reactions to the MakerBot-Stratasys Deal (http://makezine.com/2013/06/20/reactions-to-the-makerbot-stratasys-deal/).

Freeduino. 2012. Freeduino board (http://freeduino.org/about.html).

Fritzing. 2013. http://fritzing.org/about/context/

Gibb, A. M. 2010. New Media Art, Design, and the Arduino Microcontroller: A Malleable Tool. MSc thesis, Pratt Institute.

Leaflabs. 2013. Maple Board (http://leaflabs.com/devices/maple/).

Makerbot. 2012. Thingiverse Legal Terms (http://www.thingiverse.com).

Makerbot. 2013. Thingiverse: A Repository of physical design (http://www.thingiverse.com).

Netduino. 2013. Netduino Board (http://netduino.com).

OLPC. 2013. One Laptop Per Child website (http://one.laptop.org).

OSHA. 2012. Open Source Hardware Summit 2012 (http://2012.oshwa.org/).

Prusa, J. 2012. Occupy Thingiverse (post on http://www.thingiverse.com).

Prusa, J. 2013. Prusa Mendel 2 3D printer on RepRap's website (http://reprap.org/wiki/Prusa_Mendel_(iteration_2)).

Raymond, E. S. 2000. Project Structures and Ownership (http://www.catb.org).

Reas, C., and B. Fry. 2013. Official website of Processing project (http://processing.org).

Rrix. 2013. Dongiverse—a Universe of Dongs (http://www.thingiverse.com/rrix/collections/dongiverse/page:1).

Steiner, Hans C., et al. 2013. Main website for PureData software (http://puredata.info).

Thompson, C. 2008. Build It. Share It. Profit. *Wired* 16 (11): 166–176.

Wikipedia. 2013. Adrian Bowyer (http://en.wikipedia.org/wiki/Adrian_Bowyer).