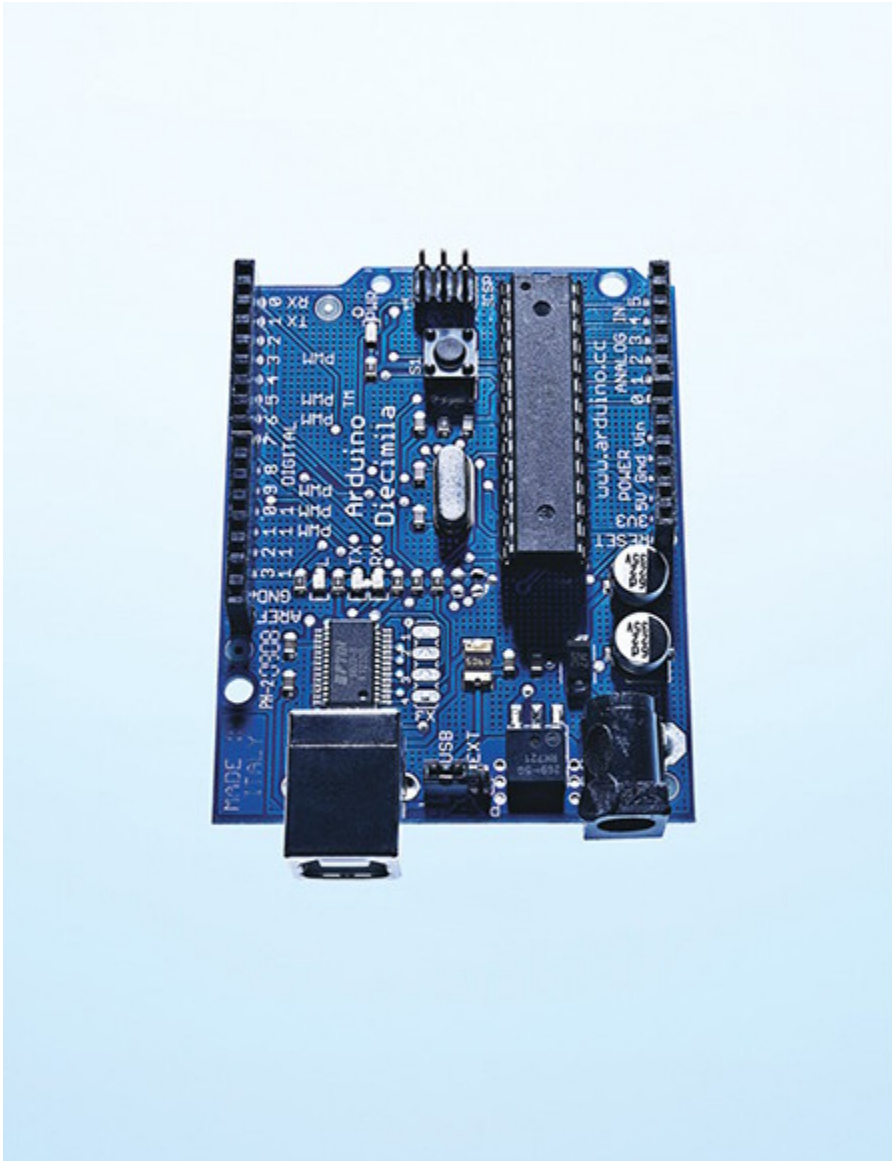


[<< Back to Article](#)

WIRED MAGAZINE: 16.11

Build It. Share It. Profit. Can Open Source Hardware Work?

By Clive Thompson 10.20.08



A circuit board for the masses: the Arduino microcontroller.

Photo: James Day

Check this out," [Massimo Banzi](#) says. The burly, bearded engineer wanders over to inspect a chipmaking robot—a "pick and place" machine the size of a pizza oven. It hums with activity, grabbing teeny electronic parts and stabbing them into position on a circuit board like a hyperactive chicken pecking for seeds. We're standing in a one-room fabrication factory used by [Arduino](#), the Italian firm that makes this circuit board, a hot commodity among DIY gadget-builders. The electronics factory is one of the most picturesque in existence, nestled in the medieval foothills of Milan, with birdsong floating in through the open doors and plenty of coffee breaks for the white-coated staff. But today Banzi is all business. He's showing off his operation to a group of potential customers from Arizona. Banzi scoops up

one of the boards and points to the tiny map of Italy emblazoned on it. "See? Italian manufacturing quality!" he says, laughing. "That's why everyone likes us!" Indeed, 50,000 Arduino units have been sold worldwide since mass production began two years ago. Those are small numbers by Intel standards but large for a startup outfit in a highly specialized market. What's really remarkable, though, is Arduino's business model: The team has created a company based on giving everything away. On its Web site, it posts all its trade secrets for anyone to take—all the schematics, design files, and software for the Arduino board. Download them and you can manufacture an Arduino yourself; there are no patents. You can send the plans off to a Chinese factory, mass-produce the circuit boards, and sell them yourself — pocketing the profit without paying Banzi a penny in royalties. He won't sue you. Actually, he's sort of hoping you'll do it.

That's because the Arduino board is a piece of open source hardware, free for anyone to use, modify, or sell. Banzi and his team have spent precious billable hours making the thing, and they sell it themselves for a small profit — while allowing anyone else to do the same. They're not alone in this experiment. In a loosely coordinated movement, dozens of hardware inventors around the world have begun to freely publish their specs. There are open source synthesizers, MP3 players, guitar amplifiers, and even high-end voice-over-IP phone routers. You can buy an open source mobile phone to talk on, and a chip company called VIA has just released an open source laptop: Anyone can take its design, fabricate it, and start selling the notebooks.

Banzi admits that the concept does sound insane. After all, Arduino assumes a lot of risk; the group spends thousands of dollars to make a batch of boards. "If you publish all your files, in one sense, you're inviting the competition to come and kill you," he says, shrugging.

Then again, Linux sounded pretty insane, too, back in 1991, when Linus Torvalds [announced it](#). Nobody believed a bunch of part-time volunteers could create something as complex as an operating system, or that it would be more stable than Windows. Nobody believed Fortune 500 companies would trust software that couldn't be "owned." Yet 17 years later, the open source software movement has been crucial to the Cambrian explosion of the Web economy. Linux enabled Google to build dirt-cheap servers; Java and Perl and Ruby have become the lingua franca for building Web 2.0 applications; and the free Web-server software Apache powers nearly half of all Web sites in the world. Open source software gave birth to the Internet age, making everyone—even those who donated their labor—better off.

Can open source hardware do the same thing?

Every open source project begins with an itch that needs scratching. Linux was launched when Torvalds decided he didn't like the operating systems available to him. The top three—Microsoft's DOS, Apple's operating system, and Unix—were all expensive and they were closed; Torvalds wanted a system he could tinker with. As it happened, a lot of other geeks wanted the same thing. So when Torvalds began working on Linux and sharing his code, other hackers were willing to pitch in and help improve it for free—creating a virtual workforce that was infinitely bigger and smarter than Torvalds himself. That is the central benefit of open source projects: They're like a barn raising in which everyone gets to use the barn. Somebody has a problem and creates a tool to solve it. And once the tool is created, hey—why not share it? The hard work has already been done. Might as well let others benefit.



Team Arduino: Gianluca Martino, Massimo Banzi, and David Cuartielles

Photo: James Day

Arduino began the same way. Banzi was a teacher at a high tech design school in Ivrea, Italy, and his students often complained they couldn't find an inexpensive, powerful microcontroller to drive their arty robotic projects. In winter 2005, Banzi was discussing the problem with [David Cuartielles](#), a Spanish microchip engineer who was a visiting researcher at the school. The two decided to design their own board and enlisted one of Banzi's students—[David Mellis](#)—to write the programming language for it. In two days, Mellis banged out the code; three days more and the board was complete. They called it the Arduino, after a nearby pub, and it was an instant hit with the students. Almost anyone, even if they didn't know anything about computer programming, could use an Arduino to do something cool, like respond to sensors, make lights blink, or control motors. Then Banzi, Cuartielles, and Mellis put the schematics online and spent 3,000 euros to make the first batch of boards.

"We did 200 copies, and my school bought 50," Banzi says. "We had no idea how we'd sell the other 150. We didn't think we would." But word spread to hobbyists worldwide, and a few months later there were orders for hundreds more Arduinos. Turns out there was a market for this thing.

So the Arduino inventors decided to start a business, but with a twist: The designs would stay open source. Because copyright law—which governs open source software—doesn't apply to hardware, they decided to use a Creative Commons license called [Attribution-Share Alike](#). It governs the "reference designs" for the Arduino board, the files you'd send to a fabrication plant to have the boards made.

Under the Creative Commons license, anyone is allowed to produce copies of the board, to redesign it, or even to sell boards that copy the design. You don't need to pay a license fee to the Arduino team or even ask permission. However, if you republish the reference design, you have to credit the original Arduino group. And if you tweak or change the board, your new design must use the same or a similar Creative Commons license to ensure that new versions of the Arduino board will be equally free and open.

The only piece of intellectual property the team reserved was the name Arduino, which it trademarked. If anyone wants to sell boards using that name, they have to pay a small fee to Arduino. This, Cuartielles and Banzi say, is to make sure their brand name isn't hurt by low-quality copies.

Members of the team had slightly different motives for opening the design of their device. Cuartielles—who sports a mass of wiry, curly hair and a Che Guevara beard—describes himself as a left-leaning academic who's less interested in making money than in inspiring creativity and having his invention used widely. If other people make copies of it, all the better; it will gain more renown. ("When I spoke in Taiwan recently, I told them, 'Please copy this!'" Cuartielles says with a grin.) Banzi, by contrast, is more of a canny businessman; he has mostly retired from teaching and runs a high tech design firm. But he suspected that if Arduino were open, it would inspire more interest and more free publicity than a piece of proprietary, closed hardware. What's more, excited geeks would hack it and—like Linux fans—contact the Arduino team to offer improvements. They would capitalize on this free work, and every generation of the board would get better.

Sure enough, that's what happened. Within months, geeks suggested wiring changes and improvements to the programming language. One distributor offered to sell the boards. By 2006, Arduino had sold 5,000 units; the next year, it sold 30,000. Hobbyists used them to create robots, to fine-tune their car engines for ultrahigh mileage, and to build unmanned model airplanes. Several quirky companies emerged. A firm called Botanicalls developed an Arduino-powered device that monitors house plants and phones you when they need to be watered.

In one sense, Arduino's timing was perfect. There's a resurgence of DIY among geeks interested in hacking and improving hardware, fueled by ever-cheaper electronics they can buy online, build-it-yourself publications like [Make magazine](#), and Web sites like Instructables. In recent years, hackers have been aggressively cracking consumer devices to improve them—adding battery life to iPhones, installing bigger hard drives on TiVos, and ripping apart Furby toys and reprogramming them to function as motion-sensing alarm bots. Inexpensive chip-reading tools make it possible to reverse-engineer almost anything. That's how Chinese hardware copycats rip off products so quickly.

Want to join the world of Arduino developers? *Wired* editor in chief Chris Anderson already has, designing two Arduino-based autopilots for unmanned model aircraft: ArduPilot and BlimpDuino (you

can find them at diydrone.com). Here's his formula for getting your creation out and into the world.

1 Download the Arduino schematic and circuit board files from [arduino .cc](http://arduino.cc). Use the free version of CadSoft Eagle (from cadsoft.de) to modify them for your particular creation.

2 Upload your files to a board fabricator like BatchPCB. Your boards will be manufactured in Chinese robotic-electronics factories and sent to your house. Typical cost is \$10 each.

3 Order bulk electronic parts from [digikey .com](http://digikey.com) and solder the components onto the board to make a prototype. Test the board and your code. You're ready to distribute your gizmo to the masses!

4 If you want to produce and sell the product yourself, use a manufacturing service like Screaming Circuits to assemble the boards on robotic pick-and-place soldering machines.

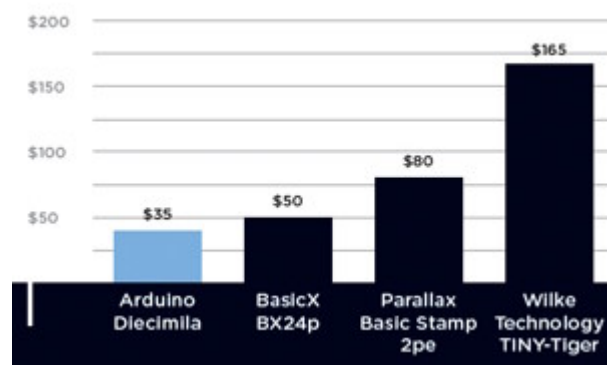
5 Alternately, an open source hardware specialist like SparkFun or Adafruit can make and sell the product for you. They'll add a profit margin and pay you a license fee .

6 Publish your revised schematics and circuit board files so that others can modify them. The cycle begins again.

This is the unacknowledged fact underpinning the open hardware movement: Hardware is already open. Even when inventors try to keep the guts of their gadgets secret, they can't. So why not actively open those designs and try to profit from the inevitable?

"Apple never open-sourced the iPod, right? But if you go down to Canal Street in Manhattan, there are copies all over the place," says [Limor Fried](#), founder of Adafruit Industries, a Manhattan company that makes and sells open source hardware ranging from the Arduino board to devices Fried designs herself. "It doesn't matter anymore whether your product is open source. Someone in another country is going to open it up and reverse-engineer it anyway."

Like the Arduino team, Fried has found that when people have access to the plans of her inventions, they suggest improvements; they almost can't help themselves. In 2006, when Fried released the design for [MintyBoost](#)—an Altoids tin crammed with AA batteries you can use to recharge your MP3 player or phone—some users complained on her forum that it wouldn't charge their devices. Other posters jumped in to analyze the problems and devise fixes; some even sketched out replacement circuitry. (MintyBoost is now Fried's most popular invention; she has sold 8,000 of the gadgets for about \$20 each.) In essence, her customers are also her tech support—available 24/7, at no cost to her.



The open source Arduino circuit board is cheaper than non-open source microcontrollers.

"But how do you make any money?" Whenever Banzai or Cuartielles describe their Arduino strategy, they're inevitably asked this question. And it's a genuine puzzle, because open source hardware isn't quite like open source software. Software costs almost nothing to reproduce; Torvalds didn't need to spend money every time someone downloaded a copy of Linux. But the Arduino team has to pay to produce its boards before it can sell them. Under traditional economic logic, this requires a patent; nobody is going to risk money inventing and selling hardware unless they can prevent competitors from immediately ripping off their designs and pouncing on their market. So how do you make money in a world of open hardware?

Right now, open design pioneers tend to follow one of two economic models. The first is not to worry about selling much hardware but instead to sell your expertise as the inventor. If anyone can manufacture a device, then the most efficient manufacturer will do so at the best price. Fine, let them. It'll ensure your

contraption is widely distributed. Because you're the inventor, though, the community of users will inevitably congregate around you, much as Torvalds was the hub for Linux. You will always be the first to hear about cool improvements or innovative uses for your device. That knowledge becomes your most valuable asset, which you can sell to anyone.

This is precisely how the Arduino team works. It makes little off the sale of each board—only a few dollars of the \$35 price, which gets rolled into the next production cycle. But the serious income comes from clients who want to build devices based on the board and who hire the founders as consultants.

"Basically, what we have is the brand," says [Tom Igoe](#), an associate professor at the Interactive Telecommunications Program at New York University, who joined Arduino in 2005. "And brand matters."

What's more, the growing Arduino community performs free labor for the consultants. Clients of Banzi's design firm often want him to create Arduino-powered products. For example, one client wanted to control LED arrays. Poking around online, Banzi found that someone in France had already published Arduino code that did the job. Banzi took the code and was done.

Then there's the second model for making money off open source hardware: Sell your device but try to keep ahead of the competition. This isn't as hard as it seems. Last year, Arduino noticed that copycat versions of its board made in China and Taiwan were being sold online. Yet sales through the main Arduino store were still increasing dramatically. Why?



Arduino gadgets: WineM coaster; Snail Light Seeker; interactive embroidery with conductive threads; and Botanicalls, which tells you when your plants need water.

Photos: James Day

Partly because many Asian knockoffs were poor quality, rife with soldering errors and flimsy pin connections. The competition created a larger market but also ensured that the original makers stayed a generation ahead of the cheap imitations. Merely having the specs for a product doesn't mean a copycat will make a quality item. That takes skill, and the Arduino team understood its device better than just about anyone else. "So the copycats can actually turn out to be good for our business," Igoe says.

NYC Resistor, a club for hardware hackers in Brooklyn, looks like a madscientists' lab, strewn with motorized doll parts, hot-rodded electric guitars, and Tupperware containers crammed full of electronic junk. I'm here to meet with [Raphael Abrams](#), a cofounder of the group. Abrams, 33, is well known in open source hardware circles for developing the [Daisy](#), an open MP3 player. It earned him so much acclaim that he now works more or less full-time designing open projects and customizing audio hardware for other businesses, including hunting companies that hire him to develop duck and deer calls. ("I'm the go-to guy for digital animal-caller designs," he says. "It's the weirdest thing that has ever happened to me.")

Abrams is deep in conversation with Alicia Gibb, a grad student who hacks hardware in her spare time. She's talking about a matchbox-sized widget that museums use to monitor humidity and temperature in their galleries. It's made by Masterpak and retails for \$115 (similar devices can cost \$400). A single institution might need hundreds of them, so it's a lucrative little market.

But as Abrams and Gibb pick apart the gadget, they realize that the price carries a huge markup.

"This is worth about \$15 in parts," Abrams says, whistling as he pokes at the tiny electronics board inside. "It has a really cheap low-end chip. And they charge \$400 for this? Someone is getting robbed." He tosses it on the table. "You could sell it for \$80."

Gibb gets a playful look in her eye. "I'm gonna do an open source version of this thing," she says. "Wait a minute," I say. "That means any museum will be able to take your free design and fabricate copies itself? Or someone who isn't even an inventor—like me—could send your design to a Chinese factory, produce a couple of thousand devices for \$20 apiece, and sell them to museums for \$50?"

"Sure," she says, grinning.

I hear the sound of a thousand business models crumbling.

If Gibb actually pulls this off without violating any patents, the company that makes the overpriced widget is in for a shock. No more easy profits based on the obscurity of its intellectual property. It will immediately have to offer a better product or improved service—or risk going out of business.

This may be destruction—but it's creative destruction. Business models will crumble, sure, but others will be born. Open source methods illustrate a hard, cold fact about hardware: It's increasingly becoming a commodity. It is not merely that China has massively decreased the price of producing goods. It's that the price of designing goods is dropping through the floor. As [Eric von Hippel](#), an MIT professor of entrepreneurship, points out, that drop is the result of the emergence of cheap or free tools for chipmaking, 3-D modeling, and online collaboration.

"In a sense, hardware is becoming much more like software, up to the point where you actually fabricate an object," von Hippel says. "That's why you're starting to see open source techniques in hardware. Design is largely going to shift out from manufacturers to the communities."

To thrive in this next wave, hardware manufacturers will have to switch their thinking. Their job is no longer just to dream up ideas—it's equally important, maybe even more vital, to seek out innovations from users. Manufacturers used to have to guess what their customers want, but the customers already know what they want, so it's more efficient to have them design it. The value of manufacturers isn't in cool designs but in economies of scale: They produce high-quality objects cheaply or offer superb shopping and support experience.

I can't help but think there are limits to this. Passionate amateurs can create an MP3 player or a synthesizer. But what about a jet engine? Or a car? To pass regulatory tests, these products require expensive laboratory equipment, like wind tunnels for car shapes and airplane parts, or crash labs. That can't be accomplished by a bunch of loosely connected designers surfing on their laptops in a Starbucks.

[Yochai Benkler](#) isn't so sure. The Harvard professor and author of *The Wealth of Networks* predicts that smart commercial firms will share resources with open source communities. "If you want to design a car in an open source way, maybe you'll work with a corporation that has access to an expensive wind tunnel," he says. This sort of cooperation has become common for open source software. IBM and Sun Microsystems pay staff members to contribute to Linux because it's in the companies' interest to have the software grow more powerful, even if competitors benefit.

Consider the [WRT54G](#) wireless router made by Linksys. It was released in 2002 as a simple \$150 router for home use. But hobbyists quickly discovered that its firmware—the software that determines the device's abilities—was based on Linux and thus legally open source. Within months, hackers had written new code that gave the device radically new features: They boosted the antenna power, turned it into a signal repeater, and constructed self-healing neighborhood mesh networks. Most of these capabilities are normally found only in devices that cost 10 times as much. Suddenly, the WRT54G market expanded. Based on the free work of amateurs, the router is now one of Linksys' all-time best-selling products.

Mani Dhillon, director of product marketing for Linksys, says the hacking has boosted the router's sales by opening up new uses. "It's a pretty strong and vocal community," he says. "We definitely credit a certain amount of the success to them."

Still, while open source hardware may be exciting, it's also confusing—even terrifying. Pioneers in the field admit they have no idea how to make the jump from small boutique hardware to mass-market devices. Banzi occasionally wonders whether he is simply being a fool by giving away some of his best work on the Arduino.

"If the Arduino chip gets bigger and better and more well known, someone in China will make it for 50

percent less. That is clear," Banzi says over dinner at a late-night Milanese restaurant famous for its coastal Italian cuisine. He stabs at his enormous bowl of orecchiette and sips some red wine, half smiling, half wincing as he imagines his work being plundered by a cut-rate offshore outfit.

"I think there's a fine line," he says, sighing, "between open source and stupidity."

It's possible that open source hardware buffs will ultimately focus not on competing with the for-profit world but in filling niches otherwise ignored.

That's what [David Rowe](#) did. Rowe is an Australian engineer who founded and then sold an Internet telephone business. He decided he wanted to help the developing world produce low-cost, high-quality telephone routers. He wanted something that would allow a company to plug in cheap, old-fashioned analog phones and place calls on inexpensive voice-over-IP networks.

"It's a huge need in Africa, but all the hardware that currently does this is, like, \$2,000 a pop," Rowe says. "African companies can't afford that." He wanted to design a device many times cheaper than that, but no existing phone-router company was interested in servicing such a low-margin market.

Rowe didn't think he could do it alone, so he organized it as an open source project. In 2005, he found a cheap chip that managed voice and data, and he wrote software for it. Sure enough, once he put the schematics online, word spread and interested hackers in Canada and Bulgaria began offering improvements. Some optimized the software; others figured out how to tweak the hardware to handle extra phone lines or how to collapse the box into a single super-powered phone line.

"We'd get stuck on a problem, and I'd hop on instant messenger and talk to the other guys and say, 'What's going on here?' I discovered that the community can figure it out a lot more quickly than I can," Rowe says.

When the time came to manufacture the device, Rowe didn't know how to find a factory. But it turns out he didn't need to. Early last year, he received a message from a Chinese firm saying it had read about the project and was interested in producing it for him. A few weeks later, the routers arrived in the mail and worked practically perfectly. Rowe commissioned the plant to begin making batches of 50. He was able to keep the unit price down to \$450 and still turn a small profit on each one. By summer 2008, he had sold a few hundred of them.

As you'd expect, Chinese competitors have already begun to manufacture routers that compete with Rowe's. He doesn't care; on the contrary, he's happy about it, because his primary goal for the devices is for them to be as cheap as possible, and fierce competition will accomplish this faster. (He and his competitors also share advice on how to improve the hardware.) A group of high tech consultants have begun selling support services to anyone who buys the router. Ideally, Rowe would like to see factories in African countries manufacture the routers, since this would bypass the punishing tariffs that make importing hardware so expensive for Africans.

Meanwhile, Rowe has become a star in high tech international-development circles, getting flown around to speak at conferences. "There's no way I would have gotten this far—and so quickly—had it been closed," he says. "This would have been a typical \$4 million or \$5 million startup if we had done it the usual way." Rowe isn't sure how the project will evolve. Will he wind up getting outcompeted, pushed out of business? Will some major hardware company offer to make the product on a massive scale?

"A lot of people got scared when I told them I was going to do this open. They were like, 'Is this going to work?'" Rowe says.

His answer: "I'm not sure."

Contributing editor Clive Thompson (clive@clivethompson.net) wrote about the making of Halo 3 in issue 15.09.