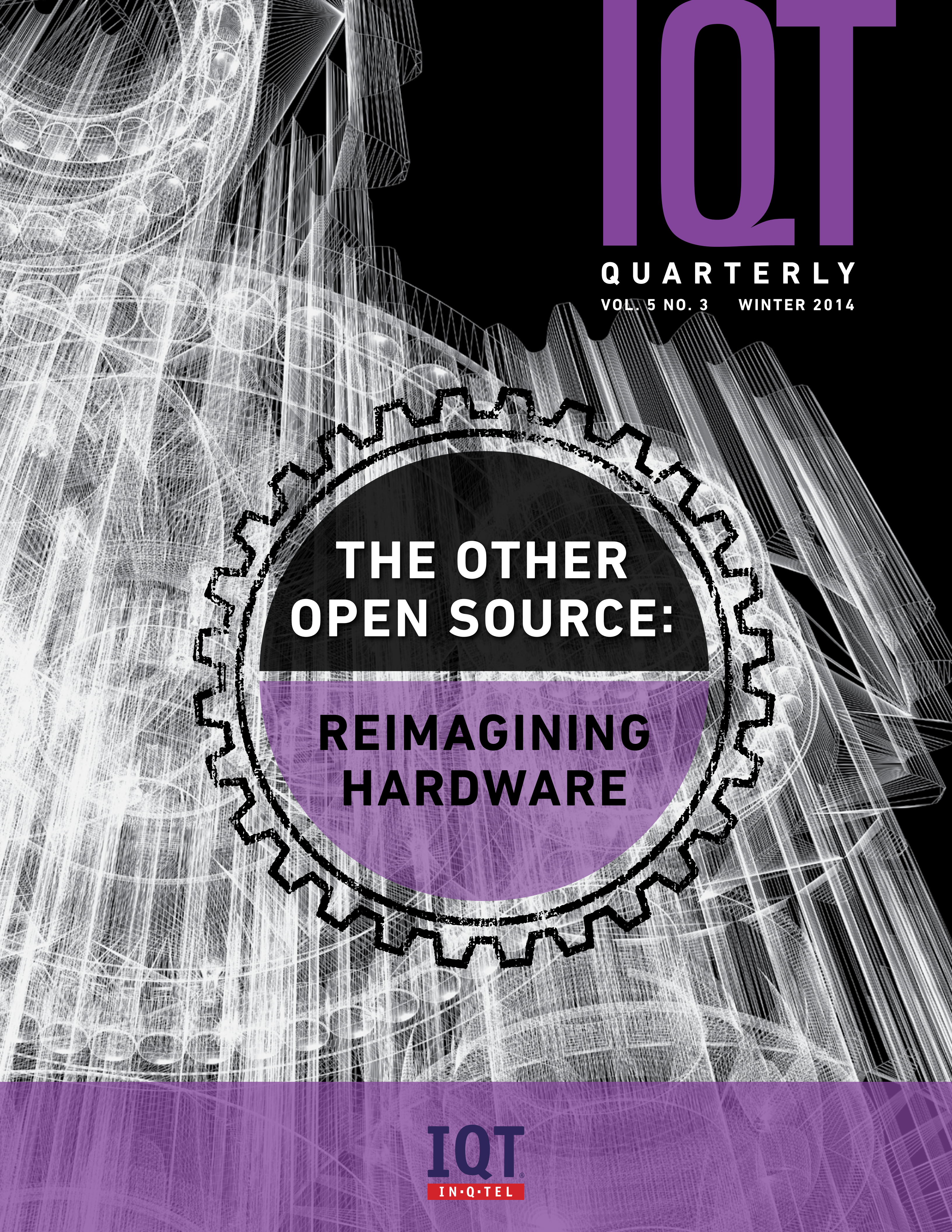
IQT
QUARTERLY
VOL. 5 NO. 3 WINTER 2014



THE OTHER OPEN SOURCE: REIMAGINING HARDWARE

IQT
IN-Q-TEL

IQT Quarterly is a publication of In-Q-Tel, Inc., the strategic investment firm that serves as a bridge between the U.S. Intelligence Community and venture-backed startup firms on the leading edge of technological innovation. *IQT Quarterly* advances the situational awareness component of the IQT mission, serving as a platform to debut, discuss, and debate issues of innovation in the areas of overlap between commercial potential and U.S. Intelligence Community needs. For comments or questions regarding IQT or this document, please visit www.iqt.org, write to iqtquarterly@iqt.org, or call 703-248-3000. The views expressed are those of the authors in their personal capacities and do not necessarily reflect the opinion of IQT, their employers, or the Government.

Design by Lomangino Studio LLC
Printed in the United States of America

IQT QUARTERLY

Identify. Adapt. Deliver.™

TABLE OF CONTENTS

OPEN SOURCE HARDWARE

By Sydney Ulvick

At first blush, the concept of open source hardware as a business is difficult to comprehend. Why do engineers develop hardware for others to use at no charge, or invest resources in improving others' designs for the benefit of all? Traditional engineering management is akin to herding cats. Doesn't somebody need to be in charge? And how do these people pay for their groceries?

The open source hardware movement found its origins in Open Source Software (OSS). Wikipedia defines open source software as "computer software with its source code made available and licensed with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose." Note that "open source" doesn't necessarily mean "free of charge" (though it is often free). The license that goes along with the software provides the terms of sharing, which typically require the user to also share modified, improved, or updated versions.

But what are the benefits of going open source? Peter Wayner elaborates in his *InfoWorld* article:

- Low-cost marketing; open source marketing is self-perpetuating
- Reduction in support costs; the community optimizes the software for its own purposes
- Reduction in development costs; the community substantially augments in-house resources
- Open-sourcing code to push back against a rival; a non-open source rival pays more for development and IP protection, and has to charge more
- Tapping open source to launch a competitor; open source licenses permit reproduction for sale[1]

These benefits show how OSS reduces cost and increases competitiveness, but how can it generate revenue? The most common OSS business model seems to be providing support services for adopters of the open source technology. The flagship example of this model is Red Hat, Inc., which has built a robust business supporting the Linux operating system.

What is Open Source Hardware?

The Open Source Hardware Association defines open source hardware (OSH) as "hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design."

The challenge to OSH lies in the logistics. For open source software, the logistical platform is well standardized and commoditized: the personal computer. The process of improving software is also well understood: write, test, and debug code. It does not require outsourced production or an investment in manufacturing, and producing copies costs approximately nothing.

For hardware, performing "open source" can be either simple or complex, depending upon the technologies involved. Understandably, the most common examples of open source hardware involve electronics; electronics design and manufacture are commoditized. However, a cash layout is still required to spin and populate a circuit board, and debugging iterations require further cash resources. Test equipment is also required, which can be complex and expensive. Both OSS and OSH require a computer with programming software, but OSH may also require utilities for embedded software, pricey

CAD packages for board layout, mechanical design, thermal modeling, or the like. By traditional logic, the developer of a hardware product would protect his or her intellectual property in order to limit risk before cash outlays become too extreme. Further investment would be required to support a staff of technologists to fully develop and optimize the product for a defined target market before introducing the hardware product for sale.

In contrast, consider Arduino and Raspberry Pi, arguably the two flagship examples of OSH. Both of these are electronics prototyping platforms with powerful on-board capabilities, which permit users to freely adapt these systems to a myriad of applications requiring computing power and a circuit interface to the real world. Both platforms were also released in an open source format. The user communities evolved the platforms for their particular applications, which by licensing terms are also open source. The advanced optimized systems can now each be obtained for around $35, and are also found at the core of numerous commercial products. A variety of companies sell the hardware (including Arduino, which originated the technology), and a variety of companies provide support to others who desire to incorporate these systems. Arduino itself makes approximately $1 per board; the bulk of its revenue comes from support services for entities that use the Arduino system. Interestingly, Arduino staff can also access the open source "literature" for solutions to be re-employed as custom services by Arduino.

Other technologies do not lend themselves to the OSH model so easily. Complex mechanical systems seem to be missing from the OSH community, although there are examples of open source communities for additive manufacturing. Whole systems also seem to be much rarer than open source subsystems and components, though systems comprised entirely of integrated open source components are on the market. This author suspects that for each technology type, a natural dollar value threshold will exist whereby the capital outlay for equipment required to produce copies will simply cost too much for an open source approach.

In summary, here's a version of how all of this could work. You're an entrepreneur with an idea for a component or subsystem. You post your idea on Kickstarter and successfully hit your dollar threshold. You use the money to produce a prototype, which you then provide to the world as open source. The idea is a good one, so others make improvements, and a significant number share their improvements with you directly. The iterative process further optimizes the product for markets where it is enjoying traction, leading to even greater consumption. Others are copying your work, but you were the originator and accordingly are the focal point for feedback. Your version of your product is therefore of higher quality, and your brand is the primary brand associated with the product. You build a significant business providing support services to others. Ultimately, someone contacts you to mass produce your product because you are the brand holder. The mass producer could simply copy it themselves, but the brand presence is needed for sales. This also undercuts motivation by a foreign entity to produce the product without permission.

With the right technology, open source hardware offers a new paradigm to be considered. The skillful entrepreneur might build an open source hardware business whose returns pay far beyond the grocer's bill.

Open Source Hardware for Mission Requirements

The IQT mission is to identify, adapt, and deliver innovative technology solutions to support the missions of the U.S. Intelligence Community. The advantages of the model are significant: lower initial and long-term costs, faster development, and enhancements that meet IC mission requirements. This statement mirrors the advantages of open source hardware, where "to meet IC mission requirements" could be replaced with "to meet the needs of [insert market here]." Open source hardware represents a new paradigm in tech development, providing an emerging rich source of technology solutions to meet mission capability needs. IQT looks forward to continuing its existing engagement in open source hardware.

Dr. Sydney Ulvick *is In-Q-Tel's Senior Vice President, Field Deployable Technologies Practice. Ulvick's 33-year professional career began as a bench chemist and progressed through the founding and operation of several small businesses before he landed at IQT in 2003. Past and present focus markets have involved biotechnology, nanotechnology, materials science, electronics, sensors, power, optics, and imaging. Ulvick holds a B.A. in biochemistry from the University of Colorado and a Ph.D. in physical chemistry from Rice University.*

REFERENCES

[1] https://www.infoworld.com/d/open-source-software/greed-good-9-open-source-secrets-making-money-228428

A LOOK INSIDE: THE *OTHER* OPEN SOURCE

The proliferation and benefits of open source software are well established, but what happens when innovators apply the same concepts to hardware designs, documentation, and schematics? In this edition of the *IQT Quarterly*, we shift our attention to the other open source — hardware. This growing movement is enabling collaboration and innovation across a range of technologies, including wireless communications, aerial vehicles, test and measurement tools, and even biology.

Gabriella Levine and Alicia Gibb of the Open Source Hardware Association (OSHWA) begin the issue by discussing the characteristics, benefits, and future of the open source hardware community, which was founded in the belief that sharing ideas, designs, and methodologies makes it easier to engineer new solutions to complex problems. The authors suggest that the most successful open hardware projects are the result of freely sharing documentation, source code, and CAD designs to adapt and build upon technologies.

Next, Peter Semmelhack of Bug Labs describes the ongoing paradigm shift in the way hardware products are designed and built. He raises unique hardware challenges absent from other open source movements — including economics and complex components — and explains open source hardware's link to the Internet of Things.

Ebrahim Bushehri's article examines the emergence of open source in the RF domain, where new technologies have enabled flexibility and power efficiency at a low cost. Mobile phones, tablets, and full radio networks are just a few of the communications tools that have reaped the benefits of freely exchanging ideas.

John Cherbini of 3D Robotics describes how aerial vehicles can benefit from open source technology. To keep up with the industry's pace of innovation, 3D Robotics has made a commitment to community collaboration, incorporating open source projects ranging from autopilot hardware design to customer-facing interface software.

Next, the founders of Kickstarter-backed Red Pitaya explain their open source test and measurement instrument.The company's online ecosystem includes a marketplace of open source applications and a repository of the corresponding source code.

Bruno Sinopoli and Antonio Rizzo highlight UDOO, an open source computing platform that can run Linux or Android. Another Kickstarter success story, UDOO features an embedded open source Arduino-compatible board and gives users full, unlimited access to its hardware schematics.

Finally, IQT's Kevin O'Connell shares insights on open source biology. While biology has lagged behind other high tech fields in open source adoption, innovative life scientists are working to address industry-specific barriers including resources, intellectual property issues, regulation, and standardization.

BROADENING THE OPEN SOURCE LANDSCAPE: Insights from the Open Source Hardware Association

By Gabriella Levine and Alicia Gibb

Open source hardware is an alternative to the patent intellectual property (IP) structure. The communally written and accepted definition states, "Open source hardware is hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or the hardware based on that design. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it.

Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs."[1]

More organizations in industry, design, and technology are adopting the open hardware definition as part of their missions and values, broadening the landscape of source files available for use. Open source hardware has been applied to electronics, fashion, furniture, musical instruments, bio-engineering, and much more. Arduino, a microcontroller and IDE (Integrated Developer Environment) software platform developed for hobbyists to make electronic prototypes, has expanded the world of hardware development for electrical engineers, artists, hobbyists, and even youth.[2] Open hardware projects cover a range including industrial machines (Open Source Ecology), 3D printers (RepRap), environmental disaster relief efforts (Protei, OpenRelief), space programs (DIY Space Exploration, Mach 30), and underwater robotics (OpenROV).

The Open Source Hardware Association (OSHWA) is a pending 501(c)3 nonprofit founded to educate people about open source hardware, collect data from the community, and voice community standards. OSHWA aims to represent the open hardware movement globally.[3]

Innovating Based on Models

Technology has always been innovated based on other people's successes, from the discovery that Earth was round to the invention of the telephone, steam engine, or airplane. Although patent laws were originally designed to protect inventors' ideas and benefit the public good, today patents can sometimes constrain further innovation. Open source is founded upon the belief that the more designs and processes can be open and shared, the quicker that innovation can happen. Open source hardware generally benefits consumers because it enables them to test, alter, and iterate upon the product, thereby allowing for competition within the free market.

There are many examples of successful businesses openly sharing software, such as Mozilla and Linux, but the rise of the open hardware trend is just beginning. This growing trend is founded in the belief that sharing ideas, designs, and methodologies can bring technological innovation and manufacturing mainstream on local and global scales, making it easier to engineer new solutions to complex problems. Open hardware projects that facilitate free sharing of documentation, source code, and CAD designs are an approach to proliferate innovation.

How to Create Open Source Hardware

As the previously shared open source hardware definition explains, one must document the complete and preferable versions of the files for a design, rather than an intermediate or obfuscated version. For mechanical components and physical designs, the preferable versions are the original CAD files, and for circuit boards, the original schematic and board layout files. The open hardware community has generated a list of best practices for documenting and sharing work related to a piece of hardware so that others can use and modify the work.[4]

Unfortunately, a technology that attempts to be open will often incorporate original design files in proprietary formats from expensive software tools because an open source software equivalent does not exist. In this case, it's helpful and encouraged to offer versions of the design in alternative or intermediate formats that can be viewed and edited with common or free programs. Such formats include PDFs of circuit schematics, Gerbers for circuit board layouts, and IGES or STL files for mechanical objects. These allow people without access to expensive or proprietary software to make use of the design as best possible. However, releasing the *original* files as well defines the core of open source hardware best practice.

Many individuals and companies that produce open source hardware publish design files on their websites when a product goes on sale (e.g., Arduino).[5] Others store their files in online version control systems (e.g., GitHub or Google Code), so that they are public throughout the design and development process. Further, there are websites specifically designed for sharing hardware designs, like Thingiverse and Instructables.[6,7] If the inventor adheres to the open source hardware definition, then he or she may use the open hardware logo to denote to the community that the project is open and the source files are publicly available.

Collaboration is Vital in the Open Source Hardware Community

The open source hardware community is made up of a diverse set of people and backgrounds. Participants often categorize themselves as DIY-ers, engineers, makers, hackers, artists, and activists, and often a combination of these categories. Communities of these people participate in the DIY movement, the rise of makerspaces/hackerspaces (places where like-minded people collaborate and innovate in science, tech, and art), the maker movement, and the open source hardware movement. Similar to the sharing ethos that occurs in the maker and DIY cultures, the open source hardware movement began as a way for people to share information and documentation for fabricating hardware. Several companies and open source hardware projects have branched from other open projects as a result of the knowledge and skill sharing in collaborative work facilities.

In 2012, OSHWA conducted a survey to collect data about the open source hardware community. The survey had 2,000 participants, although this is not representative of the entire open source hardware community. The survey found that 44 percent of the participants were using open source hardware for their jobs/careers. Only 14 percent of participants reported that none of their income comes from building open source hardware, while 86 percent of participants make some or all of their income from open source hardware. Fifty-two percent reported living in the U.S., though this number could be high as the survey was in English, and U.S.-centric despite OSHWA's best efforts to reach out internationally.

Why Go Open?

Patents were created to incentivize inventors and spur innovation in exchange for 20 years of exclusive rights. Patentees have to disclose to the public how their innovation was created. In today's patent system, 20 years may no longer be a realistic timeframe for the pace of technology innovation in the digital age. The barriers and frustrations that the patent system has created are steering inventors to adopt a new alternative to patents: open source hardware. It is vastly easier to innovate on a technology which is open with free, publicly available source files. Open source hardware creates products not driven by building monopolies, but driven by capitalistic pursuits and technological innovation in an open environment. This type of information sharing leads to powerful opportunities for companies and individuals to learn from each other.

To further illustrate these ideals, Nathan Seidle, former OSHWA board member and open hardware business owner of SparkFun Electronics, was invited to testify to the House Subcommittee on Courts, Intellectual Property and the Internet.[8] Seidle uses open source hardware rather than patents because his products are innovated within weeks, not years. His products also get copied and reproduced by consumers and users. Patented works get copied, too, but Seidle reports that it is more lucrative to out-innovate a copied product than to litigate. Open source hardware companies value a large community using, sharing, and making derivative products, working towards a common goal of bettering the world of electronics and prototyping tools.

Individuals and companies value open source hardware to make technologies more accessible and attainable by a broader audience. Additionally, open source hardware piggybacks off the DIY movement by valuing giving others design files to build things themselves and fix them when broken. People find it beneficial that open source hardware aligns with the DIY and maker ethics, valuing the ability to control, alter, and personalize the items which one owns. As products swing back to personalization from mass market goods, open source hardware makes personalization of goods possible. Not only does personalization benefit the consumer, but the fact that companies can build off of, curate, and improve other open source hardware products also means the consumer is getting a better product. Inventors are creating the marketplace and alternate IP system that they want to be part of.

The Future of Open Source Hardware

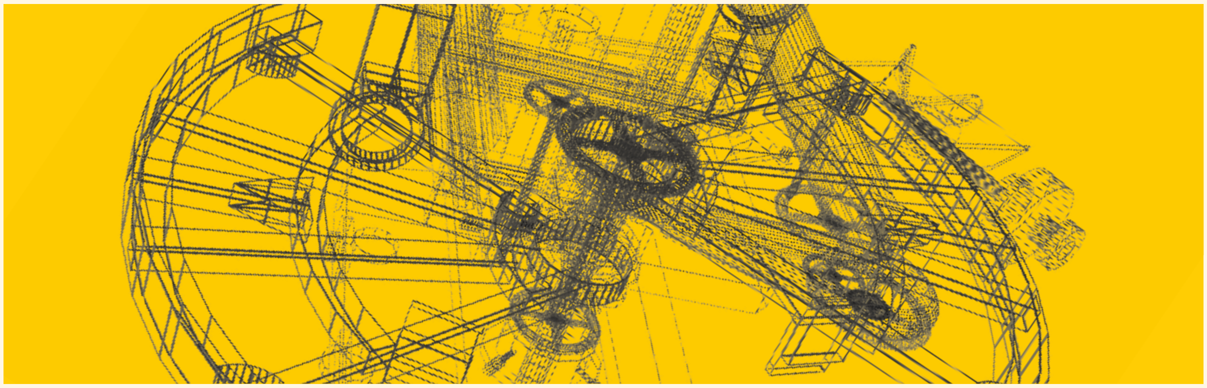
OSHWA hosts an annual Open Hardware Summit, which continues to grow each year and attract attendees and sponsors from bigger and bigger businesses. Open source hardware tools, such as open source laser cutters (e.g., Lasersaur) and open source jigsaws are now making it to mainstream markets.[9,10] Along with these new advancements and growth within the community, OSHWA recognizes that people want more options for their hardware, even in the open source niche. OSHWA is looking to develop a labeling system that would graphically represent which parts of a project were open source (e.g., the mechanicals, the electronics, the process, etc.), which parts can be easily fixed if broken, which parts can be recycled, and which parts have instructions for troubleshooting. As open hardware increases the options that inventors have when releasing their technologies, OSHWA hopes to grow the number of innovations using open source hardware and continue to relay the benefits of open source hardware to the general public. Q

Alicia Gibb *is the founding President and Executive Director of the Open Source Hardware Association (OSHWA). Prior to becoming an advocate for OSHWA, Gibb was a researcher and prototyper at Bug Labs, where she ran the academic research program and the test kitchen, an open R&D lab. She is a member of NYC Resistor, co-founder of the Open Hardware Summit, and a member of the advisory board for Linux Journal. Gibb holds a degree in Art Education, an M.S. in Art History, and an M.L.I.S. in Information Science from Pratt Institute.*

Gabriella Levine *is President of the Board of the Open Source Hardware Association, and a hardware designer interested in the relationship between technology and ecology. Levine's sculptural and robotic works include Protei Inc. (open source sailing drones), Sneel.cc (biomimetic swimming snake robots that sense environmental data), and OCP (Ocean Collaboration Platform developing marine technologies). She teaches at ITP (Interactive Telecommunications Program, NYU) and CIID (Copenhagen Institute of Interaction Design), and has presented globally at symposia and lectures including the Open Hardware Summit, Startup Festival, TEDxNavesink, and Unreasonable@State. Gabriella holds a Master's degree from ITP, Tisch School of the Arts, NYU.*

REFERENCES

1 Open Source Hardware Association. 2013. "Definition (English)." http://www.oshwa.org/definition/
2 Arduino. http://arduino.cc
3 Open Source Hardware Association. http://www.oshwa.org
4 Open Source Hardware Association. 2013. "Best Practices for Open-Source Hardware." http://www.oshwa.org/sharing-best-practices/
5 Arduino. http://arduino.cc
6 Thingiverse. http://thingiverse.com
7 Instructables. http://instructables.com
8 Testimony of Nathan Seidle, Committee on the Judiciary Subcommittee on Courts, Intellectual Property and the Internet U.S. House of Representatives. "Innovation in America: The Role of Technology." August 1, 2013. http://judiciary.house.gov/hearings/113th/08012013/080113%20Testimony%20-%20Seidle.pdf
9 Lasersaur Manual. http://lasersaur.com
10 ShopBotTools CNC Routers. http://www.shopbottools.com



A CRITICAL PARADIGM SHIFT IN COMPLEX HARDWARE DESIGN

By Peter Semmelhack

The allure of open source hardware is simple: bring the success and benefits of open source software to the world of hardware. Why? Because large organizations designing, building, and producing new electronic devices face a daunting reality. Prototypes of innovative new products typically take more than a year to complete at costs exceeding a million dollars. Compare this to today's agile software teams who can rapidly produce low-cost, innovative new application prototypes and demonstrations in weeks, if not days.

This was not always the case. A number of years ago, budding software entrepreneurs needed to run the venture capital gamut to unlock the investment dollars needed to start their businesses. But that seems like ancient history now. Open source software has completely changed the landscape. Want to become the next Larry Ellison? You may not need a VC to get started. All of the fundamental building blocks are now available for free, not least of which is a complete, commercial grade operating system called Linux. Today, believe it or not, most of the Internet runs on open source software platforms.

Hardware is Hard

Open source hardware promises similar benefits. End users can enjoy much greater control over the products they purchase and manufacturers are afforded much faster innovation cycles, time to market improvements, and, perhaps best of all, the opportunity to engage and interact with their customers in valuable new ways. MIT professor Eric von Hippel, in his influential book *Democratizing Innovation*, makes the case that this type of openness creates a unique win-win for both producers and consumers of hardware products.

Although its name is enticingly similar to open source software and its target benefits are easy to understand, the reality of open source hardware is devilishly different. The most obvious distinction is the simple economic requirements encountered when dealing with atoms versus bits. Software developers never have to worry that they will run out of "if-then" statements, nor deal with a four week backlog on "for" loops. They do not have to worry about minimum part order quantities or unit shipments delayed at customs. The list goes on. This fundamental difference lies at the root of the challenges faced by open source hardware advocates as they try to find Red Hat-like success.

There is another important difference between the worlds of software and hardware development. Whereas open source software has only two main components — the binary executable code (.exe files on PCs) and the corresponding human-readable source code — open source hardware has many. A typical hardware product requires the following elements, all of which comprise the "source code" for a modern electronic device:

- **Bill of Materials (BOM):** the chips, resistors, capacitors, etc. that populate the circuit board
- **Schematic:** the logic/blueprint of how the physical elements interact to create the final product
- **Gerber File:** the design of the circuit board's physical layout

And while a software engineer can usually open a source code file and make sense of it with any standard code editor, each of the hardware product "source" files requires specialized tools and skill sets to understand and manipulate. This makes sharing more complicated and cumbersome.

It's tempting to compare open source hardware to 3D printing due to the physical nature of the end product, and indeed there are many similarities in the value proposition. But the crucial difference lies in the simplicity of creating the final "executable." A 3D printer requires a single, specialized file to work its magic. That one file, when fed into the printer's electronic brain, produces the final product. That simplicity does not apply to electronic devices, though there may be a time in the distant future when 3D printers are capable of actually printing integrated circuits and all the other complicated components that make up a working system. But for now, that ability is still a fantasy.

Despite these challenges, open source hardware represents a critical paradigm shift in how complex hardware products are conceived, designed, built, produced, and distributed. It's a change supported by advances in technology, but, more compellingly, it's being driven by a new generation of tinkerers and do-it-yourselfers, a trend sometimes called the "maker movement." This new breed of innovator is the driving force behind much of the excitement around new hardware design generally and open source hardware specifically.

A perfect example of this group of doers' influence on the world of hardware innovation is exemplified by their recent ability to sidestep VCs entirely for their initial fundraising efforts. Instead, they tap directly into their own community for investment capital via crowdfunding. While it's beyond the scope of this article to jump into the pros and cons of this method of raising money, one needs only to visit the website of the popular startup Kickstarter to get an idea of its attraction and disruptive potential.

The Internet of Things and Open Source Hardware

The future of open source hardware is tightly linked to the emergence of another trend currently gaining momentum — the "Internet of Things" (IoT). For those unfamiliar with the term, the "Internet of Things" refers to any *thing* (hardware) connected to the Internet that's not a mobile phone, tablet, or similar device. This is usually defined by the existence, or absence, of a graphical user interface. The defining characteristic of a "thing" is that, in most cases, the real user interface is located somewhere other than on the thing itself. Take, for example, the FitBit, a popular health-monitoring device. The small gadget you wear on your belt or wrist is a study in simple, minimalist design with little in the way of a user interface (e.g., buttons or displays). Access to the user's personalized health information is via an integrated mobile application. This digital interface is physically separate from the FitBit device, but connected wirelessly, allowing for seamless communications.

But, to be clear, the Internet of Things is not *the Internet*. While this may sound obvious, the extraordinary amount of hype currently surrounding all things IoT is causing everyone to overlook this critical distinction. The reality is, unlike a piece of software, you can't upload a thing onto the Internet and, therefore, immediately avail yourself of all its benefits. In addition, the simplicity of the phrase "Internet of Things" belies another fact. Unlike the relatively homogenous markets for PCs and smartphones, the devices we all normally use to access the Internet, the IoT is a huge conglomeration of smaller, fragmentary markets, each with its own distinct technical and user requirements.

A common phrase in the world of IoT is, "the next Facebook will come from the Internet of Things!" It sounds plausible, given futurists' projections of billions, if not trillions, of devices soon to be connected to the Internet. But let's take a moment to unpack that prediction.

It is every Internet entrepreneur's dream to build a product that "goes viral." That is, to develop an application or website that is so useful, interesting, and/or provocative that a significant portion of the Internet's users suddenly find it impossible to live without. But let's look at everything that has happened, technically, up to this point to enable this "viral" event to occur.

Over the past 20 years, much of the modern world has successfully equipped themselves with some form of personal computer. There are literally hundreds of millions of PCs connected to the Internet. Most of these computers run a standardized, well-documented hardware and software platform (WinTel or Mac). If they are interacting on the Web, they are running additional collections of standard technologies — network communications (Ethernet, TCP/IP, HTTP), document formatting (HTML, XML), and presentation (Flash, CSS, Javascript). Since all of this standardized technology is in place, a good idea contained in the appropriate technical packaging can go viral with very little systemic friction — exactly like a human virus (we're all running similar "platforms" ourselves).

But if you try to equate the PC world — a completely integrated stack of standard technologies — to the world of IoT, the analogy crumbles. It falls apart precisely because there is no existing platform equivalent, no PC-like device, with hundreds of millions of identical gadgets just waiting to be connected to the Internet. For the sake of argument, let's say an entrepreneur comes up with a new mobile application that could help diabetics live longer, healthier lives. Further, let's say that this new application requires an accurate way to measure a user's blood sugar levels, via a glucometer or similar device. This seemingly minor additional requirement of a new "thing," a glucometer, represents the key challenge.

A mobile application that utilizes the built-in sensors available on a typical smartphone can experience rapid adoption, much like a website, since the technical underpinnings are there and ready to go. But in the diabetic application example, no commercially available smartphone includes any type of sensor for measuring blood sugar levels. As a result, there is no chance of a viral event occurring because first, a specialized hardware platform requires wide adoption. This is why, historically, hardware startups have required significant capital to get off the ground. Not only do the devices need to be designed and built, they also need substantial marketing and distribution to get into people's hands. But more relevantly, given the fragmented nature of IoT markets, lining up large investments from traditional VCs to launch into these small markets is very difficult.

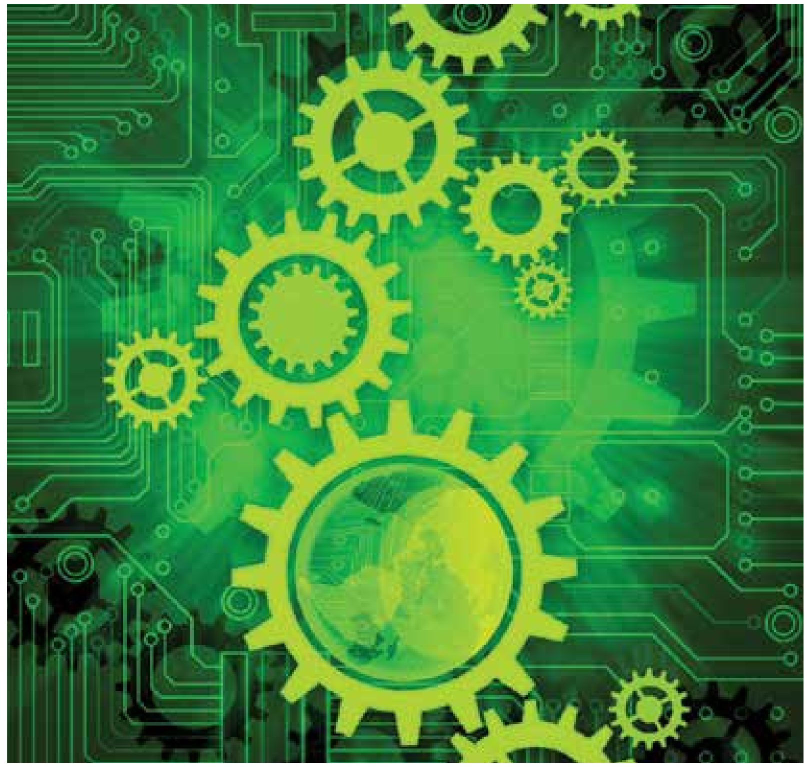
This is relevant to our current discussion of open source hardware simply because if IoT markets are fragmented and idiosyncratic, the current model for hardware innovation won't suffice. A new, low-cost and broad based approach will be necessary, one that does not require huge upfront investments and shares the risks amongst a community of like-minded innovators. Products for these markets will require specialized tool sets that traditional providers will find too costly to develop, but user communities won't. Sharing, the open source hardware community's most vital characteristic, will be at the center of IoT innovation.

As a contributing member of the open hardware movement, Bug Labs has actively participated in the open source hardware community since its inception. From contributing significant hardware and software IP to founding the Open Hardware Summit in 2010, Bug Labs has been an avid producer and consumer of open source hardware "source code." The company's customers range in size from raw startups to Fortune 100 global corporations. They all share the same basic goals: lower the costs and reduce the resources necessary to innovate in hardware. And they all are demonstrating via real-world, business-centric products and applications that open source hardware IP, and the attendant communities that surround it, is on its way to becoming a critical foundation for new technical and business model innovation.

Peter Semmelhack *is the founder and CEO of Bug Labs, the company behind BUG, the award-winning modular, open hardware and software company. Prior to starting Bug Labs, Semmelhack was the founder and CEO/CTO of Antenna Software. As a founding member of the rapidly growing open hardware movement, Semmelhack is a frequent speaker at events around the world.*

OPEN INNOVATION ENABLING NEXT GENERATION COMMUNICATIONS TECHNOLOGIES

By Ebrahim Bushehri



We live in an age where intellectual property is vital to continuing innovation and development. On the other hand, openness is just as important. This is something that progressive establishments and governments have recognized for centuries.

Patent Protection and the Shift to Open Source

Although the concept can be traced back 2,000 years to the Greek colony of Sybaris, France was the first country to introduce a comprehensive system designed to encourage inventors to give up their secrets in exchange for a temporary monopoly over sales in the 16th century.

The idea of making technology available for public inspection — and enhancement — lies behind the concept of open innovation, one of the driving forces of the open source movement. Open innovation pioneer Eric von Hippel pointed out in his key work *Open User Innovation:* "A growing body of empirical work clearly shows that users are the first to develop many and perhaps most new industrial and consumer products."

Manufacturers can develop only so many products and often cannot second-guess what users want. But users often want to tweak products to make them perform the functions they really need. Indeed, von Hippel's own research in the late 1980s found that 80 percent of the most important scientific instrument innovations were developed by users.

Manufacturers have then simplified the task of providing these innovations to others by incorporating them into their product lines.

Applying the Concept in Hardware

The concept of open innovation is most obvious today in the software domain. Although most often associated with the academic community through the efforts of pioneers such as Richard Stallman and Linus Torvalds, free or open source software was first adopted by IBM to help support its emerging mainframe business decades before the company began supporting Torvald's Linux operating system.

A developing trend is that of open source hardware. It is helping to remove a number of barriers to innovation in electronics as the functionality of products increases.

Open source hardware makes it possible to modify and extend designs to build new products more easily than with conventional off-the-shelf hardware, where key elements are difficult to examine without recourse to reverse engineering, which is inconvenient and may even be illegal in some jurisdictions. The changes in technology are helping to make the design intent of open designs easier to understand.

The rise of the field-programmable gate array (FPGA) has blurred the distinction between hardware and software by making it possible to dynamically create on-chip circuits synthesised from a hardware description language (HDL). The FPGA makes it easier to use off-the-shelf circuitry, modify it, and apply it to a new application. Not only that, the hardware can be swapped in and out dynamically if the application demands it. The raw programming bitstream employed by an FPGA is difficult to reverse engineer. But the HDL design files that are used to generate that bitstream can contain extensive documentation and pointers to design intent.

Similarly, open formats used by PCB and mechanical CAD tools can provide useful information for engineers trying to integrate multiple devices in one design. Thanks to these advances, a growing range of open source hardware is appearing, some of which has attracted widespread support.

Open Source RF Hardware

Following the success of open source hardware in the digital domain, where entire ecosystems have been formed around initiatives such as Raspberry Pi, Arduino, and BeagleBoard, the next natural step is open source in the RF domain. The core boards for the digital platforms are supplied with all documentation and source files needed to recreate and adapt them. This approach makes it easy to purchase off-the-shelf boards and I/O interfaces and then adapt the designs so they can be production optimized for the target applications, instead of the user being forced to buy off-the-shelf versions that may have the wrong form, fit, and cost parameters for volume use.

Earlier this year, Myriad RF, which sought to increase adoption and innovation in the niche RF sector by simplifying access and lowering the costs of hardware, replicated this model in the RF domain. The nonprofit initiative, led by Lime Microsystems, created support networks and open source boards, which could be adapted using their free, editable design files in the same way as in the digital domain.

The idea is to simplify RF adoption in the design of any wireless system to the extent that a vast community of users can easily create innovative solutions and access community support to accelerate the development and ultimately rollout of their products.

Myriad RF is already gaining momentum and has been joined by Fairwaves' open source base station, Nuand's bladeRF board, and Loctronix's SDR and navigation platform. All four systems, it should be said, rely on field programmable RF transceivers — such as Lime's LMS6002 — to ensure they can be adapted for virtually any application.

Open Source Hardware for Wireless Systems

Open source hardware has made it past prototypes and into finished systems such as mobile phones and tablets — and more recently the networks themselves. Projects such as the Greenphone and Openmoko demonstrated that it is possible to build complete handsets with open source hardware and software support. As a result, users were able to integrate the devices' core functionality into their communications projects.

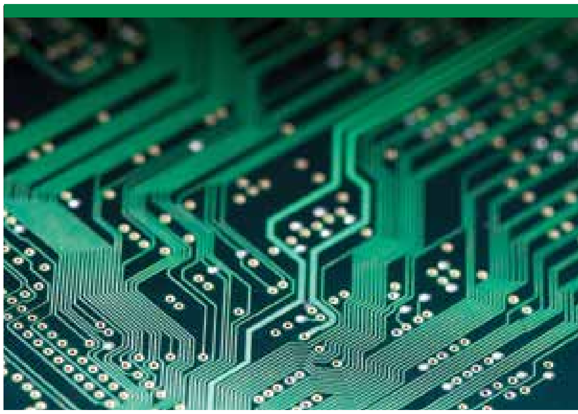
In these handsets, parts of the modem firmware have remained closed to respect the IP rights of technology providers. But the core technology behind communications represents a new frontier for open innovation. The reason for the shift towards open innovation is that flexibility and power efficiency are emerging as key requirements in communications hardware. The highly restrictive legislative environment that has dominated wireless communications is giving way to a more permissive framework that allows for greater degrees of freedom and experimentation.

On the infrastructure side, Fairwaves' open source base station was used in Nijmegen, Holland to add instant capacity and ensure emergency service communications when the Dutch city hosted one of the biggest festivals of the year and saw an influx of one million people.

But the power of open source RF hardware can be best seen in the small, remote, and rural Mexican village of Yaviche, which in 2013 became one of the first to benefit from open source mobile networks. The problems of rural telecoms are well known: mobile communications are essential to rural communities in developing nations, but how does a telecom operator rollout a network costing more than a small community could ever repay through subscriptions? The village in Mexico worked with the Rhizomatica project and Fairwaves to deliver a local network that can connect to phones in the village and nearby farms and hills for just a few thousand dollars, enabling free calls, access to emergency services, and allowing the village doctor to conduct rounds by phone.

Where is the Next Growth?

Multi-standard radios are now common in handsets, and users have become accustomed to their devices switching between radio modes as they move in and out of range of networks. Technologies such as white-space radio and event-driven radio require further flexibility.



The highly restrictive legislative environment that has dominated wireless communications is giving way to a more permissive framework that allows for greater degrees of freedom and experimentation.

White-space radio, for example, could actively sniff a broad region of spectrum to find frequencies that are not used at a particular point in time to allow interference-free transmission for short periods of time. White-space radio will help to optimize use of radio spectrum as wireless communication, particularly for applications such as machine-to-machine communication, becomes more prevalent.

Event-driven radio, on the other hand, optimizes energy consumption, using simple, very low-bitrate channels to notify wireless nodes of an incoming packet so they need only activate a more sophisticated transceiver for the time it takes to accept the transmission.

The key to adopting these and other technologies lies in agility and understanding the interaction between the key components. That understanding can best come by analyzing both the hardware (digital and RF) and software that comprise the system — and making modifications for the target application as needed. Open source technology provides the necessary underpinning for these systems to evolve and accelerate the pace of innovation in wireless technology. Q

Dr. Ebrahim Bushehri *is the founder of Myriad RF, a family of open source hardware projects providing design for low-cost and fully configurable RF boards. Busherhi is also CEO of Lime Microsystems, a fabless semiconductor company that develops RF/analog mixed signal solutions. His experience spans more than 20 years in directing and managing design teams for the implementation of high performance integrated circuits within the wireless communication market. Previously, he was the head of Middlesex University Microelectronics Centre (MUMEC) collaborating with top-tier organizations such as Nokia, QinetiQ (formerly Defence Evaluation and Research Agency), and Fraunhofer IAF. Bushehri was a professional group committee member of Institution of Electrical Engineers (IEE) and is a member of Institute of Electrical and Electronics Engineers (IEEE).*

REFERENCES

Technology Systems Referenced

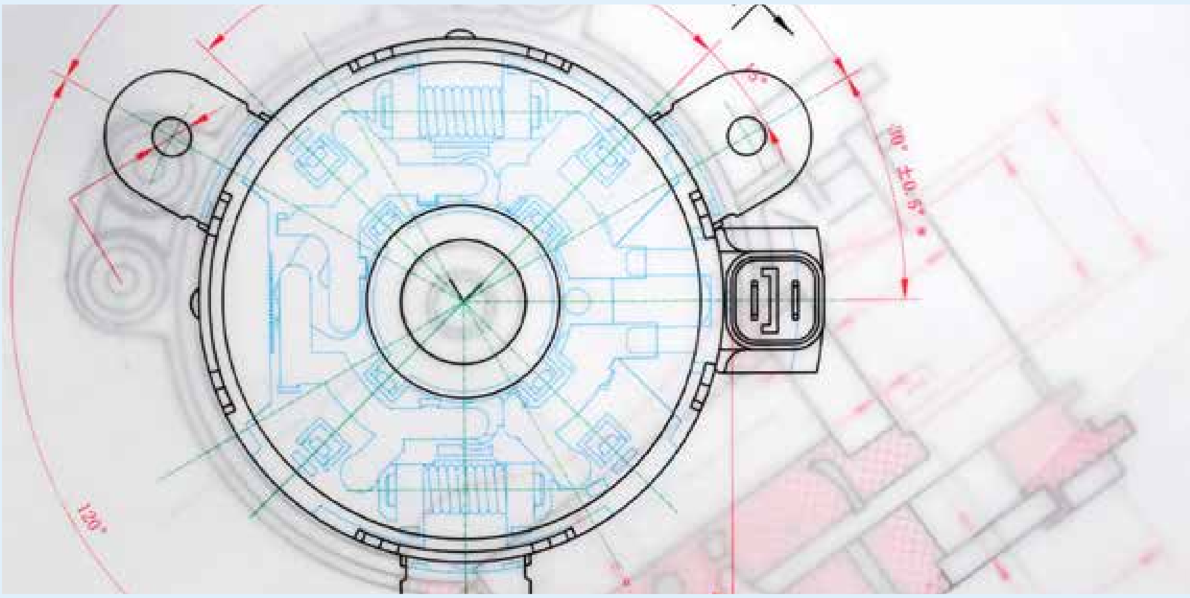
1. Raspberry Pi - http://www.raspberrypi.org/
2. Arduino - http://arduino.cc/
3. Beagleboard - http://beagleboard.org/
4. Myriad RF - http://myriadrf.org
5. Fairwaves - http://fairwaves.ru
6. Nuand - http://nuand.com/
7. Field programmable RF: http://en.wikipedia.org/wiki/Field_Programmable_RF
8. LMS6002 from Lime Microsystems: http://www.limemicro.com/products/LMS6002D.php
9. Loctronix - http://www.loctronix.com/news/Insider/I3-2-A1.html
10. Greenphone - http://en.wikipedia.org/wiki/Greenphone
11. Openmoku - http://wiki.openmoko.org/wiki/Main_Page
12. Rhizomatica - http://rhizomatica.org/

Open Source Network Implementations

1. Lime Microsystems. September 3, 2013. "World's Largest Open Source RF Network Deployed for World Leading Festival." www.limemicro.com/news/?pr=largest_open_source_gsm_network&sector=industry.
2. EE Times. November 7, 2013."Open-Source Radio Links Mexican Village." http://www.eetimes.com/author.asp?section_id=36&doc_id=1320012
3. Myriad RF. 2013. "Open Source and Rural Telecoms." http://myriadrf.org/open-source-and-rural-telecoms/

LEVERAGING THE OPEN SOURCE COMMUNITY TO ACCELERATE AERIAL VEHICLE INNOVATION

By John Cherbini



Seeding a New Industry

Flying robots have been around for more than a century. Over the past few decades, their use has expanded to a variety of applications, including targets, reconnaissance, radioactive sampling, and most popularly as weapons of war. We have reached an inflection point where core use cases for flying robots are shifting dramatically. No longer do these "drones" need to carry missiles to be valuable. Today, drones are employed to gather data effectively and efficiently across a variety of vertical applications, carrying sensor payloads that are less than a pound.

Only a few companies produce the inertial navigation system required to enable a vehicle to perform repetitive autonomous tasks. Even fewer of these companies are sharing their work openly with the public in order to take advantage of the economies of scale that occur when building products with community participation. The key to optimizing this unique open business model is to understand its advantages and disadvantages.

Cost, complexity, and closed development are common traits of previous drone generations. Small production-run hardware, rare materials, and proprietary development formerly kept the technology in the hands of those who could buy their way through steep barriers to entry. By using commodity-based hardware and community-driven software development, the cost of this technology has been lowered by orders of magnitude.

The logistics of open source development shoots chills up the spines of large corporations: "Who controls the code?" "How do we keep our competitors from leveraging this?" "How do we generate revenue unless we protect our intellectual property?" These are all valid questions. The "open" or "closed" debates that ensue often delay innovation in large organizations. Nonetheless, some groups have found that working within the constraints of open source development while leveraging the advantages of a distributed and flexible workforce can produce high quality, profitable results.

We are working in an era where high fidelity and complex sensors have become commonplace. It's the "spoils of the cell phone wars," as Chris Anderson, CEO of 3D Robotics, has called it. In fact, many of the acceleration and gyroscopic sensors used in current smartphones are functionally equivalent to the rare, proprietary sensors that are used in military unmanned vehicles. Instead of requiring millions of dollars to implement an Attitude Heading Reference System (AHRS), these can be assembled for twenty dollars. In 2014, we no longer have to buy our way across enormous barriers to entry; we can build our way through them with pocket change.

Still, simply having access to low-cost sensors will not produce a robot that flies itself.

These sensors must be assembled in a reliable package and replicate the previously proprietary calls to the sensors to produce reliable solutions that operate autonomous vehicles. There are a few ways to approach this problem. One is to bankroll a cadre of mechanical and electrical engineers to develop a single thread of innovative applications. Another, arguably better, approach is to engage and cultivate a community of experts who can pool their various talents and skill sets to find solutions that replicate — and then improve upon — the closed offerings of the past.

Community is a term that has many positive connotations, especially in today's crowdsource-addicted culture. From sharing business startup costs to real estate to cars, the use of a community model optimizes the value of resources, from personal to intellectual property.

At the same time, distributed access carries negative connotations, especially for big businesses. First, there are the free-riders who give nothing back and solely benefit from the efforts of others. Then there's the idea that free and shared equates to untested and inferior. Within traditional business models, these negative connotations typically outweigh the positives, and many companies decide to go it alone and produce proprietary, closed-source products.

There are a few notable exceptions to this standard proprietary IP business model. Google's Android, Red Hat's Fedora, and IBM's Eclipse and Apache have all demonstrated that there are profitable business models that rely upon valuable community collaboration. 3D Robotics (3DR) has also taken this open approach to drive innovation.

Staying Open in a Proprietary-Minded Competitive Landscape

3DR is a leader in the open unmanned aerial vehicle (UAV) industry. Founded in 2009 by Chris Anderson and Jordi Munoz, 3DR has grown into a thriving VC-backed enterprise with more than 170 employees across North America.

With over 35,000 square feet of manufacturing space, four pick-and-place lines produce thousands of finished electronic products in the course of a single shift. Three Computer Numerical Control (CNC) lines output additional components for final assembly of fully autonomous fixed-wing, multirotor, and ground-based vehicles. 3DR's manufacturing facility operates at an enterprise level due to the decades of high quality electronics and assembly manufacturing skills developed by the likes of Samsung and Foxconn.

The UAV industry is evolving at a breakneck pace. 3DR has risen to the challenge by leveraging its community of experts to accelerate time to market. The company's most popular autopilot hardware design is based on open source technology. This means anyone who has surface-mount device manufacturing skills can replicate 3DR hardware. The primary driver behind this decision is to continue pushing the innovation curve by allowing others in the community to augment and iterate on the design and improve reliability and functionality.

As with any community-based effort, the outcomes that 3DR has experienced are sometimes surprising. Some have taken the design and simplified the functionality and footprint to meet new customer requirements. These positive efforts warrant respect, adding value to the industry and furthering the community as a whole. Less valuable efforts have resulted in unmodified clones of the original design (including adoption of the 3DR name and logos). While this may meet certain customer requirements, it is not in the spirit of the community-based collaboration model.

3DR's firmware and customer-facing interface software is also open source. This code has been developed through the DIY Drones community, and supported and sponsored by 3D Robotics. DIY Drones is now the largest robotics community in the world with more than 46,000 active members. Discussions in the community elevate thought leadership in the space, revolving around the UAV industry as a whole, and not just 3DR. As with any open, online community, members participate at different levels, ranging from lurkers and consumers to active developers, beta testers, code contributors, and community managers. Some community members have even become full-time 3DR employees.

Although 3DR certainly benefits from the open source community, there are times when staff engineers are required to deliver on customer requirements. This delineation highlights the nuances of the open source business model. One cannot always count on the community to deliver on all business needs. Nonetheless, 3DR contributes internal work to the community, enriching the common knowledge base and catalyzing further innovation across a variety of applications, only some of direct interest to the company. This synergy is used to drive innovation and "turn the crank" faster than is possible in traditional business models.

It could be argued that lowering barriers to entry by passing on lower prices to customers through decreased test and development cycle costs is the largest advantage to the open source business model. An equally valid argument is that quality is the most critical advantage. With developers all over the world, work is rapidly cycled from one developer's sunset to another's sunrise. Every aspect of the hardware and software is constantly being reviewed and improved. As a result, thousands of contributors add value to the code line without requiring thousands of developers on the payroll. Customers can glean the benefits of this model by riding the leading edge of this innovation cycle, without the associated costs. More sensitive customers have the ability to adopt, test, certify, and standardize versions of the hardware and software to meet their individual requirements.

In terms of future innovation, the open source community generates ideas and allows 3DR to explore new use cases. Members often discuss potential applications and market verticals. While 3DR has chosen to focus resources on delivering value in agriculture, mining, construction, conservation, and cinematography, the DIY Drones community leverages the open source code to research and develop hundreds of different global applications.

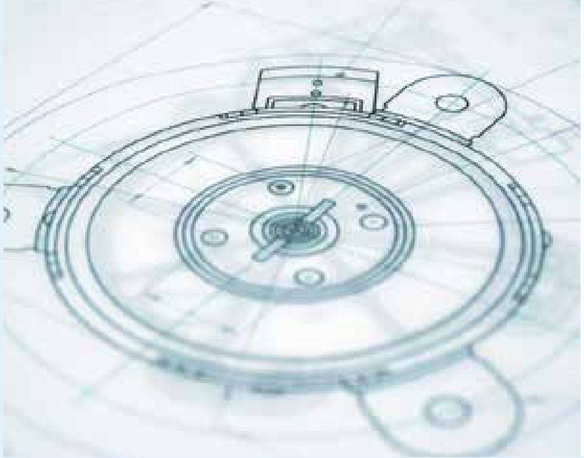
Where Do We Go From Here?

Open source robots have made their way from the hobbyist's workbench into the hands of mainstream consumers. Autonomous data collection no longer requires countless hours behind a soldering iron. Drones are quickly gaining value for a variety of new applications — for collecting data in new and creative, repeatable ways. Over time, drones will become easier to use, more reliable, and more capable. Eventually, they will even be trusted to safely transport packages to your doorstep.

Here in the U.S., the Federal Aviation Administration ultimately commands the fate of commercial drones. Over the next year, they will conduct research at approved test sites and create a certification process for smaller drones to share the national airspace. The Association for Unmanned Vehicle Systems International predicts that during the first three years of commercial integration, more than 70,000 jobs will be created in the U.S. with a global economic impact of more than $13.6 billion.

Regulations surrounding drone usage vary significantly around the world. In Australia and Canada, where rules around commercial drone usage are more liberal, innovative applications will likely be deployed even sooner. We are at a crossroads where the demand for gathering data effectively and efficiently is growing, but the technology is being developed under the auspices of an uncertain regulatory environment. How this demand will translate into operational regulations and innovative applications for lightweight unmanned vehicles across the world is yet to be seen.

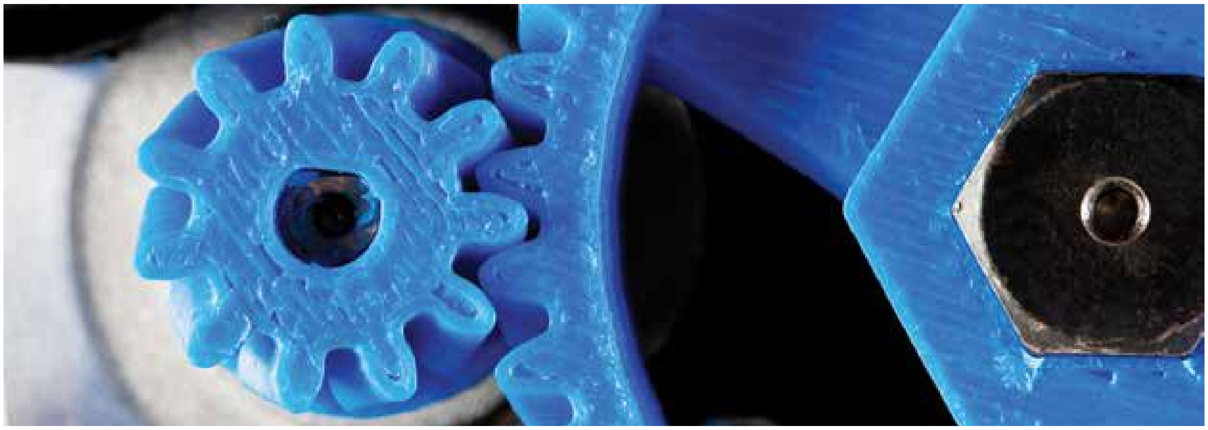
The unmanned vehicle consumers will ultimately decide how closed, proprietary systems with enormous development costs and longer innovation cycles will co-exist with equivalent open, modifiable systems with lower development costs and the ability to innovate at a faster pace.



In 2014, we no longer have to buy our way across enormous barriers to entry; we can build our way through them with pocket change.

One of my favorite sayings is, "a washing machine stopped being a robot when it became a washing machine." The key to this change in vernacular is ease of use. Reduced costs and faster innovation cycles also allow open source companies like 3DR to drive higher consumer adoption rates and increased customer satisfaction. Today's washing machines require only a couple of inputs to produce reliable output. Soon it will be so easy for aerial vehicles to gather valuable data across multiple industries that they will no longer be regarded as robots. They'll simply be drones.

***John Cherbini** is Vice President of Sales and Marketing for 3D Robotics. He leads enterprise delivery as well as online and channel sales and services for customers worldwide. Coming from a long background in hardware, software, and services, Cherbini came to 3D Robotics after five years of operating and integrating 3DR products into his own projects. His early grasp of the 3DR platform's commercial enterprise applications is helping him build scalable sales and product processes to drive rapid growth.*



TEST AND MEASUREMENT TURNS TO OPEN SOURCE AND KICKSTARTER

By Aleš Bardorfer, Borut Baričevič, and Rok Uršič

Kickstarter-backed Red Pitaya is a compact open source test and measurement instrument promising to replace many expensive laboratory instruments. Its users will benefit from an online ecosystem that consists of Bazaar — a marketplace where open source applications are available within a single click — and Backyard — an organized repository containing the corresponding open source code and tools necessary for developing applications.

In the early days of the electronics industry, test and measurement (T&M) equipment such as oscilloscopes, signal generators, and the like came with fixed functionality and performed a small, rigid set of tasks. Over the last few decades, manufacturers have leveraged the programmability of digital signal processing (DSP) to offer broader feature sets tailored for specific applications. However, the ability to program these instruments and determine their final feature sets was restricted to whatever the manufacturers did before the products left the factory.

Even with broader feature sets, many instruments on the market today don't have the functionality customers may require. This is especially true for interdisciplinary research groups at institutes and universities worldwide, which focus on building large, complex systems based on smaller subsystems, with instruments being one of the latter. In such cases, it is often essential to use the instruments at much finer levels than the vendors who designed the equipment enabled. Typically, integration problems arise in closed loop feedback systems involving two or more instrument subsystems that require a hard real-time response with very little jitter. Often, the interfaces to the instruments prove inadequate to achieve the required overall integrated system performance. Closed source, DSP-oriented instruments prevent such access.

Case Study of an Open Source Hardware Startup: Red Pitaya

Instrumentation Technologies is known in the field of particle accelerators for its family of sophisticated instruments called Libera.[1,2] These programmable instruments measure and stabilize different properties of high energy particle beams in closed feedback loops, enabling particle accelerators to reach their full potential and become powerful scientific tools. But engineers at Instrumentation Technologies also wondered what would happen if they created instrumentation — or let us say a signal processing platform — that was conveniently priced, easy-to-use, user-customizable, and open source.

Encouraged by the ideas behind the Lean Startup movement, a team of engineers at Instrumentation Technologies "got out of the building" and organized in-person interviews and Skype calls with people and groups using T&M equipment.[3] The team listened carefully and tried to understand the users' working environment and the issues they faced. This was a surprisingly rewarding experience, which also helped identify opportunities the users were not able to articulate by themselves.

To facilitate this process, the team went through a succession of a few minimum viable products (MVP), the first one being only a draft block diagram on a sheet of paper.[4] Progressing through a set of MVPs, they developed a signal processing platform, which they called the Red Pitaya.

But the team did not stop there; during the interviews they identified two more areas of difficulty. The first one is associated with starting up T&M instruments and the second is the complexity of customizing these platforms to their specific needs. As a result, the team decided to enhance the Red Pitaya platform through the addition of an ecosystem that allows the user to customize the device well after the initial sale. The Red Pitaya ecosystem consists of an application store called Bazaar and an open source development environment called Backyard. The core idea behind Bazaar is to establish an attractive Internet marketplace where people can freely browse and select from an inventory of applications. Each of these turns Red Pitaya into a specific instrument or measurement tool in a simple and straightforward manner. Backyard, on the other hand, is for those who would like to understand how the applications are built, improve them, or create their own from scratch. Backyard acts as a platform to support and encourage a development community.

The Red Pitaya team decided to use Kickstarter to gauge market interest in the concept and fund the first production lot.[5] The campaign was launched on July 22, 2013 and ended on September 20, 2013. During that time, the team managed to collect $256,125 in pledges, exceeding the initial goal of $50,000 by a factor of five. Instrumentation Technologies decided to spin-off Red Pitaya as a separate company.

System Architecture Considerations

Good instrumentation in general requires excellence in two domains. The analog domain, which is moving to ever higher frequencies, still dictates the noise floor and measurement performance of an instrument. But it is the digital domain that is gaining momentum due to its stability, noise immunity, time invariance, and ability to perform very complex processing, combined with standard connectivity. Thus, signal processing is increasingly moving from the analog world to the digital. There is another advantage to the digital world: it is reconfigurable. This creates an attractive opportunity for instrument developers to build a variety of functionalities into a single platform and share them with others. It is therefore natural to create an Internet repository in which one stores "rules" about how the platform is configured to become an instrument, and tools for their development. This simple idea led to the development of the Red Pitaya ecosystem with Bazaar and Backyard.

The basic approach in modern signal processing systems is to be as generic as possible at the analog front-end electronics, and to sample the signals with fast analog-to-digital converters (ADC) as soon as possible in the processing chain. On the digital side, the trend is to process the signals and send them to the digital-to-analog converters (DAC) as late as possible in the processing chain, and again to be as generic as possible in terms of the analog back-end electronics. With such an architecture, all the specifics of the processing/measurement system are concentrated in the digital domain, while the analog electronics are kept simple and generic. This mostly digital modern architecture creates a great opportunity for the customization of hardware to perform a number of tasks across several applications spaces. The possibilities are limited only by the bandwidth of the analog front and back ends, and the computational resources of the digital domain — primarily the field-programmable gate array (FPGA) and the CPU resources.

Based on the Red Pitaya team's experience, an FPGA and CPU form the basic blocks of a signal processing system and have always been a winning combination in reconfigurable instrumentation. This combination allows for freedom in partitioning the parts of signal processing between the high-performance FPGA and the easily programmable CPU. In particular, almost any instrument faces the problem of reducing huge amounts of input data from the raw sampling point to the point of the instrument's outputs. Typical examples of measurement results include a signal plot of an oscilloscope, a frequency-domain plot of a spectrum analyzer, or a stream of position coordinates of a basketball player being tracked by processing the output of a ceiling-mounted camera. All these measurement outputs represent much less data compared with the raw instrument input signals, the latter being sampled at very high frequency. This reduction of data from the instrument's inputs to its meaningful outputs is therefore the main job of the digital signal processing embedded in the instrument.

Having both the FPGA and the CPU available on the signal processing system enables the developer to freely decide which parts of DSP processing to implement on the FPGA and which parts on the CPU. There are differences between them in terms of processing suitability, but both can perform digital signal processing. In general, the FPGA handles ultrafast, yet simple DSP operations, but is less suitable for complex procedural operations. The CPU, on the other hand, excels at slower, but arbitrarily complex procedural operations. CPUs are also good at running standard interactive interfaces, such as Web servers. Despite the large improvements in FPGA development tools in recent years, it is still generally easier to write procedural software to run on the CPU, compared with RTL coding and synthesis of digital structures in the FPGA.

The freedom of partitioning the DSP processing between the FPGA and the CPU brings another advantage — namely, the ability to rapidly prototype a performance-limited, but fully functional system. Implementing most of the DSP on the CPU allows prototype demonstrations at early stages of the development project, suitable for marketing, while at its later stages, this strategy allows for a smooth transition of the performance critical part of the DSP to the FPGA for a final product with the same functionality, but full performance.

THE INSTRUMENT

Red Pitaya is a modern signal processing platform with multiple analog and digital inputs and outputs. There are two main types of processing chains present on Red Pitaya in terms of speed. The first type is the ~50 MHz bandwidth signal processing chain achieved by leveraging the extremely fast and low-jitter hard real-time processing capabilities of FPGAs. The other Red Pitaya processing chain, at a bandwidth of ~50 kHz, is achieved through the CPU, with the ability to run a hard real-time operating system.

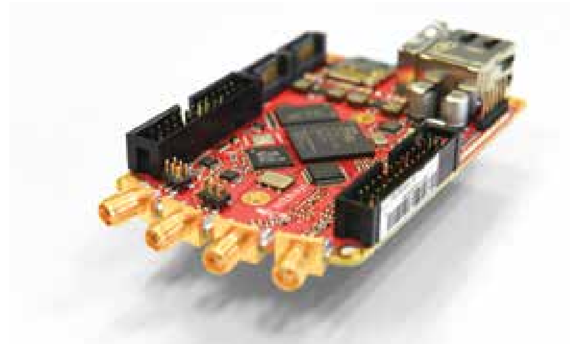
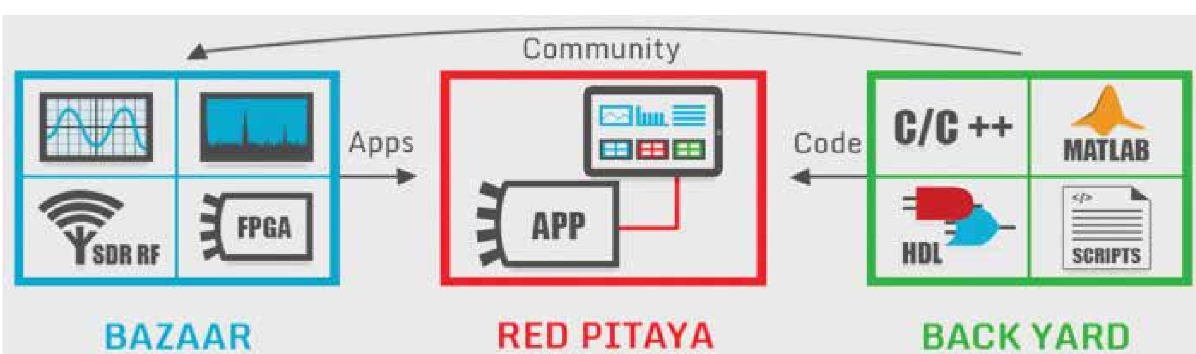


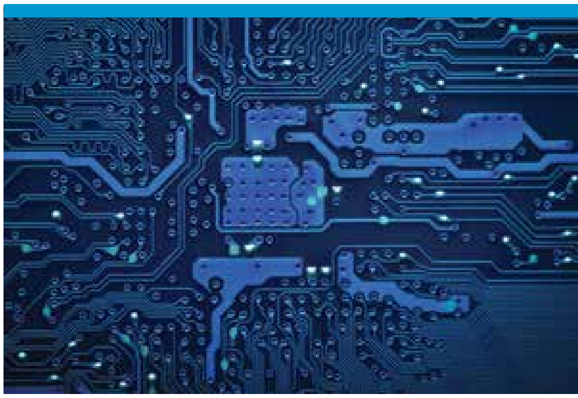
Figure 1 | The Red Pitaya hardware

The Red Pitaya system is based on the Xilinx® Zynq®-7010 All Programmable SoC. Red Pitaya boasts fast two-channel, 125 M sample/second signal-acquisition and signal-generation capabilities, which can be combined with FPGA based DSP processing in between to form hard real-time feedback loops. In addition to fast signal processing, the system includes several slower (~100-kHz) I/O channels, leveraging Xilinx Analog Mixed Signal (AMS) technology, along with several digital I/Os. Distributed processing is possible if you daisy-chain several Red Pitaya modules via fast serial connectors. In this way, you can build a complex system, requiring more inputs and outputs, using several interconnected Red Pitaya subsystems. The CPU runs the Linux operating system and supports standard peripherals, such as 1000BASE-T Ethernet, USB OTG, Micro SD storage, and USB serial console.

THE ECOSYSTEM

The goal of the Red Pitaya ecosystem is to bring people together and allow their creativity to accomplish something more significant than they could alone. In general, ecosystems try to be inclusive, and they catalyze productivity. At the base of the ecosystem is a connected community, an informal group of individuals and companies who share common purposes. These





The goal of the Red Pitaya ecosystem is to bring people together and allow their creativity to accomplish something more significant than they could alone.

purposes include shared technology, and business practices such as encouraging differentiation and cooperation to help all members succeed at the end.

The Red Pitaya ecosystem consists of:

- **Red Pitaya:** a high performance tool with a convenient price tag.
- **Bazaar:** an Internet marketplace where open source applications are available within a single click for immediate experience and use.
- **Backyard:** an organized repository containing the corresponding open source code and tools necessary for developing applications.

Initial applications in the Bazaar include a two-channel, 125 M sample/second oscilloscope, a spectrum analyzer, a signal generator, and a PID controller. The beauty of open source is that motivated enthusiasts can develop custom applications to fit their specific needs. As with every open source customization, chances are that others will be able to use the modified or new instrument or application as well. Additional applications are expected to evolve over time within the Red Pitaya ecosystem.

The Red Pitaya team's vision is to enable everyone use technologies that were only available to advanced research laboratories and industry for a fraction of the cost.

Dr. Aleš Bardorfer *is a passionate engineer, climber, mountaineer, and co-founder of Red Pitaya. He obtained his B.Sc. and Ph.D. in Electrical Engineering from University of Ljubljana, Slovenia, followed by a fruitful career in various fields of robotics, haptic interfaces, virtual/augmented reality, medicine, rehabilitation, and particle accelerators. Before co-founding Red Pitaya, he worked as a researcher, developer, teacher, and system designer at various universities, institutes, and companies in Slovenia, the U.S., the U.K., France, and Australia. Bardorfer is fond of the open source and Linux communities and is currently active in Red Pitaya system and software design.*

Borut Baričevič *co-founded Red Pitaya after a long experience developing high performance instrumentation for particle accelerators. He graduated in Electronics Engineering at Trieste University, with a degree thesis about microwaves. He worked for particle accelerator laboratories on high-power RF design projects. Afterwards, Baričevič continued his career at Instrumentation Technologies where he worked mainly on RF design and signal processing applied to accelerator diagnostics instrumentation and low-level RF systems. Currently, Baričevič is active on Red Pitaya user experience, application, and community development. He's still involved in hardware and signal processing design.*

Rok Uršič *is an avid entrepreneur, co-founder of Red Pitaya, and founder and CEO of Instrumentation Technologies. Before founding his first company, Uršič worked in different R&D and management positions at particle accelerator laboratories in Italy, the U.S., and Switzerland. Uršič obtained his B.Sc. in Electrical Engineering from the University of Ljubljana, Slovenia.*

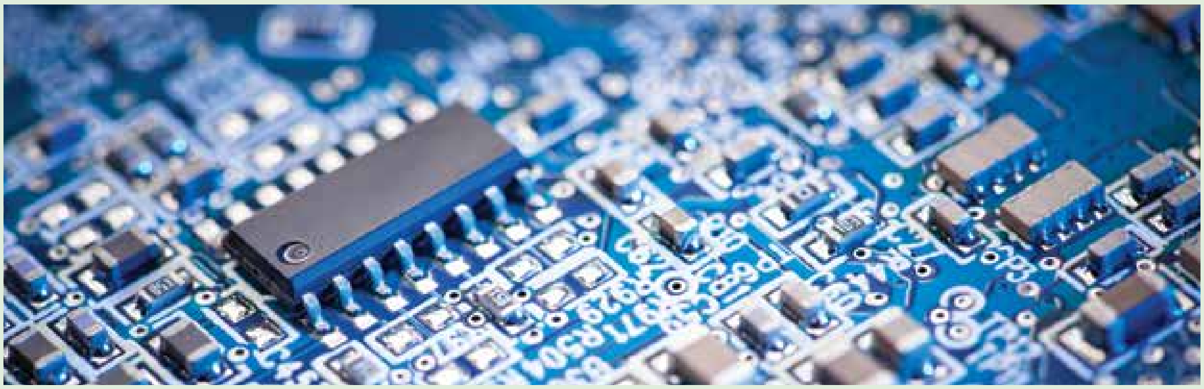
REFERENCES

[1] Instrumentation Technologies. <www.i-tech.si>.
[2] Instrumentation Technologies. "Accelerator Products." 2013. <http://www.i-tech.si/accelerators-instrumentation/>.
[3] Blank, Steve. "Why Real Learning is Outside the Building, Not Demo Day." *Steveblank.com*. September 5, 2013. <http://steveblank.com/2013/09/05/why-real-learning-is-outside-the-building-not-demo-day/>.
[4] Ries, Eric. "The Lean Startup Methodology." *The Lean Startup*. 2013. <http://theleanstartup.com/principles>.
[5] "Red Pitaya: Open Instruments for Everyone." *Kickstarter*. 2013. <http://www.kickstarter.com/projects/652945597 red-pitaya-open-instruments-for-everyone>.

TRANSFORMATIVE COMPUTING PLATFORMS:

Android, Linux, and Arduino in an Open Hardware Mini PC

By Bruno Sinopoli and Antonio Rizzo



Isaac Newton, paraphrasing a metaphor of Bertrand of Chartres, wrote in one of his letters, "If I have seen further it is by standing on the shoulders of giants," highlighting the importance of building upon past discoveries to rapidly advance knowledge. In the information age, the nontrivial integration of cyber and physical domains to design complex systems demands reuse of both software and hardware components to tame complexity and enable rapid development.

The UDOO project is an attempt to contribute to this need in the embedded electronics world by creating a revolutionary computing platform where users not only have full, unlimited access to all the hardware schematics and software, but they can contribute as a community by shaping its evolution. Its flexibility, ease of use, ability to interact with the physical world, high computing performance, energy efficiency, rugged design, and low cost make it the ideal solution for prototyping products in a large variety of application domains, such as robotics, home and building automation, interactive design, and military applications. Moreover, its minimal footprint (4.33 in. x 3.35 in., or 11 cm x 8.5 cm) allows for easy embedding into objects and spaces, making it an ideal choice in the Internet of Things.

UDOO is the quintessential open source project, as the concept not only applies to the software (e.g., operating system), but also to the hardware design and implementation. Every detail, ranging from the schematics to the pin layout, is available to the community. The documentation is readily downloadable and continuously updated to include improvements and feedback from the community. This allows the community to fully understand the board's functionality, and therefore push its limits and unleash its potential. The open source model aligns with the efforts of the community, transforming a potentially competitive effort into a collaborative one and increasing overall quality of the design. In this context, the ultimate beneficiary is the end user, who can leverage a solid platform to build new products and services.

UDOO breaks the mold of today's single board computers, bringing together two ARM CPUs in one compact and low-cost board. Equipped with an ARM Cortex-A9 i.MX6 Freescale processor for heavy computation with low power consumption, alongside the Arduino Due's ARM SAM3X for real-time operations with digital and analog sensors and actuators, UDOO is a revolution in the Do-It-Yourself (DIY) and educational markets. UDOO can run either Android or Linux, with an Arduino-compatible board embedded, and a wealth of options for communication and human-machine interfaces. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. UDOO is a powerful prototyping board for software development and design; it is easy to use and with a few steps, users can start creating projects with minimal knowledge. UDOO merges different computing worlds in one; each with its own strengths and weaknesses, all of which are useful today in education, DIY, and rapid prototyping endeavors.

Thanks to its features, UDOO can be used as a stand-alone computer, able to run any sort of Linux distributions for ARM architecture such as Ubuntu (Linaro), Debian, Yocto, Arch Linux, and XBMC, giving the user all the necessary tools for programming needs. The Arduino IDE, for example, allows users to build and upload sketches to the Arduino-compatible embedded board without any additional hardware or cable connections. UDOO also runs the latest Android operating system including all the features of the Google OS. The Accessory Development Kit (ADK) 2012 expands the possibilities of hardware manipulation by making UDOO an Android device and its own accessory.

One of the key innovations of UDOO is the direct connection between the two processors, connected by a UART Serial port delivering a two-way channel with a tested 115,200 baud rate. This seamless integration not only addresses the needs of many designers working in the Internet of Things, but it also offers opportunities to improve energy efficiency. In a scenario where computation is needed only when an event from the physical world is triggered, the ARM processor used in the Arduino can be used to wake up the rest of the system, thus providing the duty cycling capability necessary to ensure long-lasting operations in battery power mode.

UDOO's Kickstarter campaign launched on April 9, 2013, to a very warm reception that raised almost $650,000 (27 times the requested amount) by the June 8 closing date, and started shipping on time. Loyal to its nature and mission, UDOO made some key changes suggested by Kickstarter backers.

Thanks to an international network of collaborators from several universities, UDOO has been used in

Figure 1 | UDOO's open hardware, low-cost board

various research projects and educational activities at institutions such as Carnegie Mellon, UC Berkeley, and UC San Diego. Students at Carnegie Mellon are using UDOO for a large number of projects ranging from art installations to robotics, including the design of an automated roofing robot and a lunar rover participating in the Google Lunar XPRIZE. Of particular interest is the latter, where students have been considering UDOO as the computing platform of choice for their rover. As a result, they have subjected UDOO to all the harsh environmental conditions the moon would offer, with satisfactory results to date. Professor Miller Puckette at UCSD is using UDOO to build an ultra portable 6-channel guitar processor for advanced effect processing. Aidilab, one of the creators of UDOO, is using UDOO for several projects in the area of interaction design.

Prototyping with physical computing toolkits has become a widespread method for technology development, design exploration, and creative expression. The board allows researchers and designers to quickly create and explore new interaction techniques and design devices in the prototyping sessions of a project. UDOO provides a full suite for developing prototypes based on the Android platform alone or combined with ADK 2012.

In addition to fast prototyping, this platform offers enormous opportunity in education. In particular, tools like UDOO can raise the level of interest in technology in K-12 education, a crucial goal toward keeping the country at the forefront of high-tech. UDOO makes it possible for users with different skills in coding and electronics to learn easily. Users can choose the configuration that fits their level of skills and grow in

The goal is for companies and inventors to use UDOO to build high-level prototypes of products and services quickly and easily.

the direction they prefer. Step-by-step "how to work with electronics" tutorials will be available for every level of expertise, while there will be ready-to-go Arduino sketches for those interested in working with codes. With UDOO, teachers can teach basic and advanced topics with the same low-cost hardware. They can use the same platform to explore different topics, from interaction design to firmware programming to robotics, as well as the possible synergies among topics.

As important as the board is the creation of a community of users. Open source projects thrive only when a large number of users contribute with suggestions, feedback, and by posting their own developments. UDOO's aim is to create a seamless educational and professional community that could grow according to the challenges posed by the new emerging technologies (e.g., Internet of Things, cloud computing, cluster computing, etc.).

UDOO is building this community starting from a small set of selected universities (Carnegie Mellon; University of California, San Diego; Aarhus University, Denmark; University of Siena, Italy; Ontario College of Art and Design, Canada) with the aim of bootstrapping the design process of the board and its accessories and then opening up to the rest of the world.

UDOO will support the community through educational events, competitions, and an educational website for teachers and students. Furthermore, while keeping the academic world as the main partner in this educational endeavor, UDOO aims to address the basic process of computing literacy in primary school and beyond, through academic projects like Scratch and Squishy Circuits.

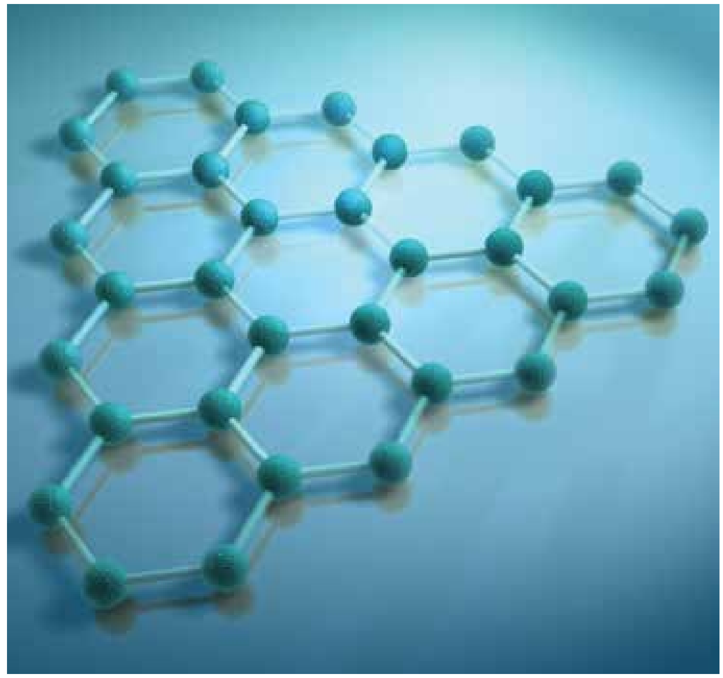
UDOO plans to make its portal a reference point for the open hardware community. UDOO is generating a tremendous amount of interest, with users posting not only feedback, but verifying claimed capabilities, finding new ones, and even contributing software improvements and accessories and designs, such as a file to print an enclosure. Ultimately, the community will dictate the evolution of the product, according to their needs and the market's needs. UDOO wants the portal to become a place where not only the community shares freely, but also serve as a B2B and B2C platform for the users who wish to sell their products and designs. Q

Dr. Bruno Sinopoli *is an Associate Professor in the Department of Electrical and Computer Engineering at Carnegie Mellon University. Sinopoli received his M.S. and Ph.D. in Electrical Engineering from the University of California at Berkeley in 2003 and 2005, respectively. Previously he received the Dr. Eng. degree from the University of Padova. His research interests include networked embedded control systems, distributed estimation and control with applications to wireless sensor-actuator networks and system security. Sinopoli was awarded the 2006 Eli Jury Award for outstanding research achievement in the areas of systems, communications, control, and signal processing at U.C. Berkeley.*

Antonio Rizzo *is a Professor of Cognitive Science at the University of Siena. Previously, he served as Director of the Academy of Digital Arts and Science – ArsNova in Siena, Chair of the European Association of Cognitive Ergonomics, and Head of the Human Factor Group of the Italian National Railways. Rizzo's main interests are in the domains of interaction design and cultural-historical psychology.*

OPEN SOURCE WETWARE:
Molecular Biology Breaks Out of the Ivory Tower

By Kevin P. O'Connell



Many technology entrepreneurs and innovators take their inspiration from companies that have risen to the top of their industries from humble beginnings in garages, basements, and dorm rooms. Apple, Amazon, and Google are among the most famous examples, but behind them are many others, in fields spanning a wide array of hardware and software products. One technology area less prominent in this evolution from garage to stock exchange is the life sciences. This article examines developments in life science innovation driven by lessons learned from other tech fields, examines barriers that set biology innovation apart from other tech fields, and highlights a variety of ongoing challenges in this space.

Open Source Origins

Contributing to the diversity of technologies being developed is the increasing availability of the tools and knowledge needed to pursue innovation. A major source of such tools is the growing movement in open source technology. The open source movement in engineering, especially software, is a reaction to perceived restrictions in the ability of developers to innovate and collaborate freely, exchange ideas, and commercialize the fruits of their development. It is also a philosophical reaction to the perceived sequestering of technology benefits behind legal ownership of intellectual property (IP). Many young entrepreneurs bring with them out of academia a desire to continue sharing knowledge in the academic fashion and a desire to develop products of broad benefit to society. They also believe that innovation accelerates when more people are able to work on ideas simultaneously. Advances such as Linux and its derivative operating systems, the Mozilla browser, Arduino processors, and Raspberry Pi microcomputers are all well-known results of open collaboration, and their easy availability for use is the result of free licensing of their code, schematics, blueprints, and other source materials. Open source technology has made available software code libraries that form the modules from which many products are built.

Life Scientists Take Notice

Why has open source practice lagged in the life sciences? The answers lie in both economics and in the nature of biotechnology and its deep roots in basic science. In academia, life science knowledge sharing is robust in both journal publication and in the release of raw genetic data, as evidenced in the sheer volume of published works and genetic databases readily available online.

However, the cost of turning basic science information into economically sustainable products has historically been staggering; a small molecule drug made by a pharmaceutical company will typically require 10 years and hundreds of millions of dollars to identify and test. The cost and time required to develop products, the laboratory and manufacturing infrastructure needed, and the intense regulatory oversight from government agencies has largely kept individual innovators from working on novel products in molecular biology and is the reason that most biotech startups spin out of universities. The cost of productization for many life science products is also a major driver for protecting biotech IP via patenting and vigorous, litigious patent protection.

Also, unlike other tech fields, molecular biology products have until recently been driven by de novo creation of individual products, rather than product development from industry standard parts, modules, or libraries. Certainly there are some basic tools that have been commercialized (restriction enzymes and ligases, which serve as the cutting and joining tools for DNA manipulation, are prominent examples), but these are tools; there are far fewer actual genetic or protein "building blocks" available for incorporation into novel products. Often the few that are available carry liberal use licenses for research, but onerous licenses for incorporation into commercial products, if such licenses are available at all.

Molecular biologists watching their engineer colleagues across campus and in the marketplace have been keenly aware that their own infrastructure needs, regulatory oversight and IP issues, and the "hand-crafted" rather than "built-from-standardized-parts" nature of life science innovation present challenges to the individual entrepreneur and innovator. These challenges stand in stark relief from the open knowledge distribution necessary to conceive of new inventions, as available in the open literature, taught in university classrooms, and increasingly available free in massively open online courseware (MOOCs) from top-tier providers like MIT and Harvard.

Life Scientists React

Open source biology is a technological and cultural reaction to these challenges. It is loosely organized, and shares a "hacker" (or perhaps a "maker") ethos with engineering innovators. It asks, fundamentally, how the barriers of resources, IP issues, regulation, and standardization can be lowered or eliminated, so that the tools and collaboration that fuel innovation in other fields can be available to molecular biologists. Here is how open source biology is addressing some of these issues:

Parts: In 2003, researchers at MIT began to assemble and distribute "BioBricks," DNA sequences that encode defined biological structures or genetic regulatory functions.[1] These genetic parts can be assembled in novel ways to convey new functions to microbial cells, such as producing light, odors (like wintergreen), and pigments in response to pre-programmed external stimuli (temperature, exposure to light, exposure to chemicals, and so on). The Registry of Standard Biological Parts now contains over 10,000 parts, and sets standards for documentation, assembly, and packaging to facilitate the submission of new parts to the registry, and to ensure interoperability and usefulness of new parts in the context of the entire catalog.

Community: The BioBricks Foundation is part of The Synthetic Biology Network, which also includes groups devoted to knowledge sharing, organizing conferences, curating the Registry of Standard Biological Parts, and organizing an annual, global competition for innovative molecular biology projects based on the Registry. Beyond this group-of-groups, "DIY-Bio" communities have formed in cities in North America, Europe, Asia, and Australia.[2] Such groups vary in their level of activity but typically include meet-ups, teach-ins, space sharing, and blogging in addition to supporting the creation of shared lab space. Many DIY-Bio communities have established "biohacker spaces," such as BioCurious in the San Francisco Bay Area, where members can perform experiments, share instrumentation, teach, and collaborate outside traditional venues such as universities and biotech startup incubators.[3] BioCurious has a group that is collaborating on the design of a 3D printer that will be able to pattern cells into tissue- and organ-like structures. Such instrumentation projects often involve "makers" leveraging open source electronics and software to accomplish an open source biology goal.

Resources: The infrastructure needed to perform the most sophisticated work in molecular biology is still a formidable challenge, but some of the most powerful techniques, such as DNA amplification and cloning via the Polymerase Chain Reaction, can now be performed on low-cost thermocyclers with openly shared, license-free designs. Molecular biology no longer relies heavily on radioactive chemicals to trace the progress of experiments, removing the expense and effort in using those substances. Likewise, the cost of both DNA sequencing and of synthesizing designed DNA sequences has plummeted, making synthetic biology more accessible. The rise (and then fall) of venture-backed biotech companies, as well as the pace of innovation in molecular biology equipment, has created supply and demand for second-hand equipment that is serviceable and affordable, especially for groups or co-ops who are pooling resources. Last but not least, the advent of crowdsourcing has brought some DIY-Bio projects a financial boost: both BioCurious and an inventor of a license-free PCR device have received funding through Kickstarter and Indiegogo campaigns.

Licensing and IP: Unlike software, whose IP is typically protected by copyrighting, molecular biology IP is usually protected by patenting, a much more expensive process. The BioBricks foundation has created the BioBrick Public Agreement (BPA), a "scalable contract" between an

Open source biology asks, fundamentally, how the barriers of resources, IP issues, regulation, and standardization can be lowered or eliminated, so that the tools and collaboration that fuel innovation in other fields can be available to molecular biologists.

individual who wants to make a BioBrick part free to use and someone else who wants to use it freely.[4] It is unclear at this writing whether BPAs have been tested in litigation, but clearly the community is striving to create the legal underpinnings for an analog to open source software. In another recent development, the US Supreme Court ruled in mid-2013 that human genes cannot be patented, which loosened patent protections for many biotech products and conversely made genetic information accessible to a wider field of innovators.[5]

Software: Programs designed to analyze, catalog, search, match, and design DNA sequences are integral to the information-intensive nature of biotechnology. This intersection of software and biotechnology is referred to as bioinformatics. The open source ethos of software developers has been integral to bioinformatics since its inception, and the core algorithms that underlie even commercially available DNA analysis packages are sourced from the open literature. Besides commercially available software suites, there is a vast array of molecular biology analytical software tools available free online, many created by graduate students and post-doctoral researchers.

Impacts and Issues

Has the open source biology movement had the same impact on biology that open source code has had on software development? In short, no, or perhaps not yet. Despite the efforts of the open source biology community, the cost of working in molecular biology is still high relative to writing code. That said, BioBricks' standardized parts, distributed to participants in the iGEM (International Genetically Engineered Machines) competition, have enabled undergrad and graduate workers to produce an impressive array of inventions. The iGEM competition began in 2004 and went international the next year. In 2012 alone, over 170 teams from North America, Latin America, Europe, and Asia built genetic components into organisms that:

- Detect and digest environmental pollutants
- Micromachine copper wafers
- Make cavity-fighting antibiotics, vitamins, building materials, and fuels
- Duplicate logic functions in genetic signaling networks
- Prevent the transfer of synthetic genes to naturally occurring organisms

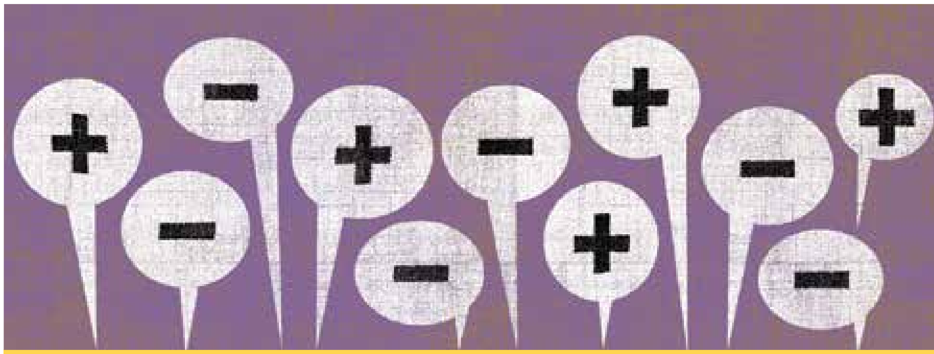
While the movement has begun to show how standardizing biological parts can stimulate innovation, it still remains to be seen whether and how that innovation will be manifest in the marketplace, given the remaining barriers to scaling up manufacturing, the lack of open source experience in the biotech IP arena, and the similar lack of experience with open source biology regulatory considerations.

What challenges does open source biology pose from a safety and security perspective? The same infrastructure challenges that so far limit the economic impact of open source biology also prevent, to a large degree, the kinds of projects that would raise serious and immediate public safety concerns. DIY-Bio work typically occurs at BioSafety Level 1 (BSL1), which is comparable to work done in high school and freshman college labs. Eckard Wimmer's lab at Stony Brook University demonstrated as early as 2002 that it is possible to assemble a viable poliovirus genome entirely from DNA synthesized by a commercial vendor, but projects of that scale and complexity are still far beyond the capabilities of biological "garage entrepreneurs" or local DIY-Bio groups.[6] Without the containment facilities available at institutional laboratories, would-be DIY-Bio malefactors run the significant risk of becoming their own first victims. While a complete discussion is beyond the scope of this article, developing and testing the means to effectively disseminate an engineered pathogen is similarly complex and resource-intensive. Commercial vendors of synthetic DNA cooperate with government and law enforcement to monitor the sequences ordered by customers, and the tools and chemistry to make one's own synthetic DNA is still beyond the means of DIY-Bio groups.[7] That said, DIY-Bio groups are loosely, if at all regulated and depend on member oversight to ensure safety and compliance with ethical pledges not to engage in dangerous work. BioCurious, the Bay Area biohacker space, provides paid members with safety training and lab waste removal services, but groups vary widely in their level of organization and oversight. The barriers to increasingly sophisticated molecular biology work continue to drop and the potential for misuse unique in the open source technology world will correspondingly increase with time. Therefore, the challenge for public safety officials will be to ensure that open source biologists establish and maintain an ethic of responsibility, and remain engaged in public dialog about the impact of their work.

Dr. Kevin P. O'Connell *is Vice President with IQT's Field Deployable Technologies Practice. He has been with In-Q-Tel since 2007. His 28-year career began in applications of molecular microbiology to problems in agriculture, and progressed to research and development projects in biological defense. He was a scientist and principal investigator with a Department of Defense laboratory for 10 years, where he focused on pathogen detection and genetic characterization. He is the author of over 50 peer-reviewed journal articles, book chapters, and other publications, edited a recent book on biological defense, and holds eight patents. O'Connell holds a B.S. degree in Life Sciences from MIT and an M.S. and Ph.D. in Bacteriology from the University of Wisconsin-Madison.*

REFERENCES

1 http://biobricks.org/
2 http://diybio.org/local/
3 http://biocurious.org/
4 http://biobricks.org/bpa/
5 Association for Molecular Pathology et al. v. Myriad Genetics, Inc, et al. Certiorari to the US Court of Appeals for the Federal Circuit, N. 12-398. Argued April 15, 2013-Decided June 13, 2013. (http://www.supremecourt.gov/opinions/12pdf/12-398_1b7d.pdf)
6 Cello J., Paul A. V., and Wimmer, E. 2002. Chemical synthesis of poliovirus cDNA: generation of infectious virus in the absence of natural template. *Science* 297:1016-1018. (http://www.ncbi.nlm.nih.gov/pubmed/12114528)
7 Keasling, J.D and Venter, J. C. 2013. Applications of Synthetic Biology to Enhance Life. *The Bridge* (a publication of the National Academy of Engineering), 43:47-58. (http://www.nae.edu/Publications/Bridge/88361/88366.aspx)



TECH CORNER

To supplement the ***IQT Quarterly***'s focus on technology trends, *Tech Corner* provides a practitioner's point of view of a current challenge in the field and insight into an effective response.

FLEXIBLE AND OPEN SOURCE: THE BRAVE NEW WORLD OF RF

A technology overview from IQT portfolio company Lime Microsystems

On September 27th, 2013, in the tiny, remote Mexican village of Yaviche, a revolution began.

A project that brought mobile access to the 700 people living there for just a few thousand dollars saw the first mobile calls made. This project can be replicated throughout the developing world, and bring benefits of wireless technology.

Key to this affordable and life-saving implementation was an emerging technology: flexible RF and the open source business models that it has enabled.

From mobile networks implemented by the residents of a small village in Mexico's Oaxaca mountains to portable 2G base stations for disaster recovery teams, to machine-to-machine systems for asset tracking, these advances in flexible RF technology are allowing greater innovation in the RF sector and greater simplicity for those wanting to implement these technologies.

Figure 1 | Yaviche Santa Maria, a remote village of 700 in the Oaxaca mountain range[1]

4G and the Internet of Things: Why a Revolution is Needed

The obvious example that highlighted the need for flexibility was Apple's third generation iPad. It was launched in 2012 to provide access to 4G networks and with them, download speeds of 100 Mbps. However, the only LTE bands the product's transceivers accessed were 700MHz and 2.1GHz. At the time, these bands were only used in the U.S. which left potential users in other parts of the world with no access — albeit that didn't stop those in Australia and the UK buying the device for its 4G connectivity.

Indeed, according to the analyst house Strategy Analytics, 40 LTE (E-UTRA to be more precise) bands exist globally, with over 35 assigned and several pending approval.

If we look beyond the mobile communications sector to the next big growth area, the Internet of Things (IoT), we see a similar picture. This sector is widely forecast to see explosive growth as applications such as the smart grid and intelligent industrial and home automation systems take hold. One major component of the IoT will be the smart gateway that acts as the coordination point between low power, low-cost local sensor nodes and the wider Internet. These gateways will need to support

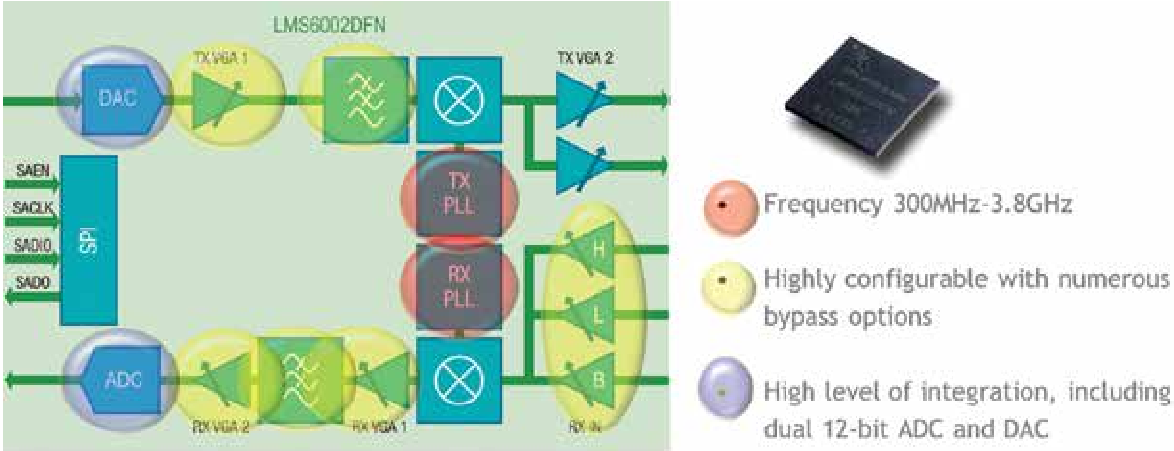


Figure 2 | LMS6002DFN FPRF (in production since 2010), first in the world for achieving the highest level of flexibility and integration

a wide range of protocols and interfaces, though not necessarily simultaneously. The problem is that both the network and transmission use multiple standards that will take time to harmonize.

The problem for traditional hardware radios is that each of these functions demands the use of fixed-function hardware, using various application specific integrated circuits (ASICs) and discrete RF and analog components to support each target air-interface standard. Although the interface-specific components allow some flexibility, such as the choice of a narrow range of channels through software control, this flexibility is limited to specific RF interfaces.

Flexible Radios

Clearly, therefore, transceivers need to be used for every standard — from FM radio to 4G and beyond. The increased need to implement RF into almost every design also requires this technology to have a plug-and-play capability and strong support — be it by the manufacturer or via an open source community. The first example of a single chip field programmable RF (FPRF) radio that has been widely adopted came from Lime Microsystems.

Alongside the RF in any modern communications system, there is a need for digital signal processing stage which runs on general or specialized processors or field-programmable gate arrays (FPGAs). These commonly used digital chipsets offer all the required flexibility to deal with multiple standards, thanks to advances in state-of-the-art silicon technologies. These devices provide the horsepower to demodulate and decode the incoming signal from the receiver or apply the complex modulations for today's protocols ready for transmission through the programmable RF subsystem.

This means FPRFs sit in the center of an ecosystem involving DSPs, FPGAs, general-purpose processors, and ASICs. But, this model also means a flexible transceiver can be applied to markets as diverse as small cells, software defined radio systems, cognitive radio, machine to machine, white space WiFi, unmanned aerial vehicle communications, and cellular backhaul.

The need for flexibility in radio systems is becoming widely recognized by equipment vendors, operators, and governments around the world. Indeed, the European

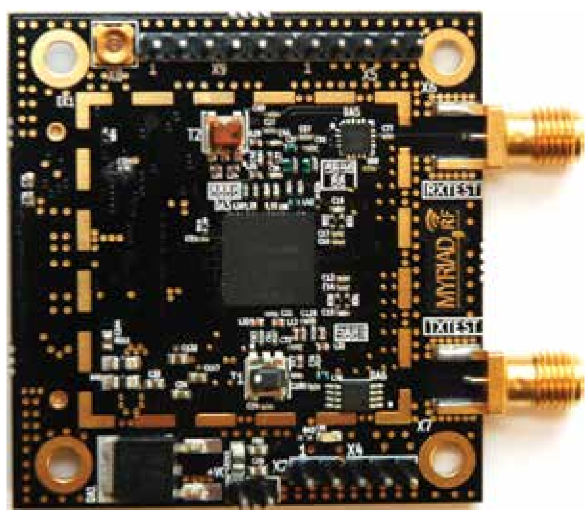


Figure 3 | A Myriad RF open source board based on Lime's FPRF transceiver technology[2]

Union's roadmap for SDR and cognitive radio (the Acropolis Project) highlights the need for flexibility and cites Eurecom's Express Mimo-2 platform as an example of best practice. This board runs four field programmable RF chips coupled with an FPGA and this puts flexibility at its heart.

Open Source Business Models

For those in the RF sector, 2013 will be seen as the year open source took off. The first four launches of open source technology came from Myriad (RF boards), Fairwaves (base stations), Loctronics (indoor navigation and SDR), and Nuand (SDR). All four came in 2013 and placed a flexible approach at the heart of the design, enabling greater functionality for users.

By adopting the open source model, these organizations are taking RF down the path taken by Linux in software and Arduino and Raspberry Pi in system level hardware.

Bringing Communications to the Masses

The number of mobile subscriptions is currently six billion (or ~80 percent of the world's population), but this hides a world of haves and have-nots. Adoption rates vary greatly, with 170 percent in Finland and less than 20 percent in Ethiopia. One-third of the world's population does not have access to affordable mobile telephony.

But open source models that place flexible architectures at the heart of their designs can change this — changing the supply chain and enabling small scale deployments on a fraction of the budgets currently required to deploy voice and data networks. This is the model that has been deployed by the Mexican village of Yaviche with the help of Rhizomatica.org and Fairwaves.

The problems of rural telecoms are well known; mobile communications are essential to rural communities in both developed and developing nations, but too costly for a telecom operator to roll out. In order to solve this, Rhizomatica.org has been negotiating with governments in Mexico and Nigeria to permit local networks on licensed spectra in locations where incumbent operators could not justify infrastructure costs. The same model can be replicated by operators in the U.S. and Europe to lower the cost of wireless infrastructure for the benefit of the scattered communities within their territories.

Rhizomatica is working as a bridge between the remote communities and the engineers/technology developers to create their own local networks enabling better access to emergency services and medical aid.

Figure 4 | To cut costs, Yaviche mounted base stations on bamboo masts[3]

The Yaviche network was implemented with help from Fairwaves, deploying its low-cost, open source base station along with the open source base station software OsmoBTS. This is mounted on bamboo poles to make use of readily available material and keep costs down.

As many as 14 high quality and 25 standard quality voice connections can be made concurrently from one base station. The importance of the network has already been demonstrated: access to medical care has increased with the local doctor beginning to make rounds via phone call, freeing up time and prioritizing patients who need immediate attention.

Changing Supply Chains

The revolution will not be limited to Yaviche, nor to just Mexico. Installations of this open source technology have also taken place in Africa (e.g., on the Indian Ocean island of Mayotte) and even in Europe, with Monaco trialling experimental network system from Spectra, which will have its first public outing in April 2014.

But this affects the supply chain, too. The traditional telecom ecosystem is a straight chain from integrated circuit vendor to operator and content supplier through the incumbent OEMs.

This supply chain may change significantly as existing OEMs will be undercut by open source alternatives. This means IC vendors and ODMs can speak directly to the operators. And a plethora of startups will launch to take advantage of new applications via the open source community. After all, this could lead to the democratization of the wireless infrastructure whereby a vast community of organizations and individuals could

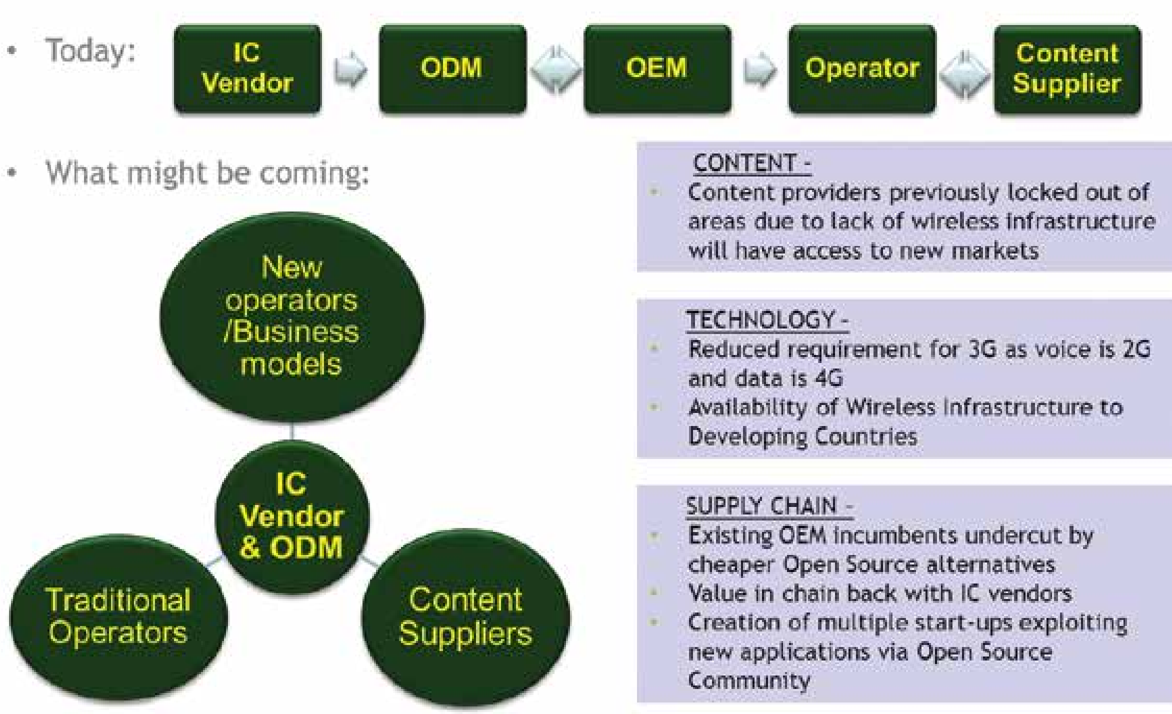


Figure 5 | Today's ecosystem and how it might change

innovate and open the market beyond a handful of suppliers who have had very little interest in engaging the open source community for the development of future wireless networks.

In short, the democratization of wireless systems is underway, and this is backed by a growing open source community — and, of course, the communities for whom it is a lifeline. Trials are underway that rely on flexible architectures and open source models to deliver the technology to the communities that need it.

But this also presents an opportunity for the operators seeking to cut costs and deliver data and voice wireless access to the vast number of people throughout the developing world with no connection — just 10 percent of India's population has access to mobile Internet — and an opportunity for those seeking to bring RF technologies to new applications, in particular those that will connect to the Internet of Things.

The wireless design market has changed dramatically in the past decade and it now addresses an exceptionally diverse array of applications. This is especially true with the rise of software defined radio and machine-to-machine communications, which provide the backbone of the Internet of Things.

The increasing complexity and intelligence in wireless environments will put greater cost pressures on discrete front-end technologies, as more users embrace programmability. A highly integrated field programmable RF architecture, such as the one designed and manufactured by Lime Microsystems, provides a solution that is not only able to react to today's challenges, but tomorrow's as well. Q

Lime Microsystems *is an IQT portfolio company that specializes in digitally configurable transceivers for the next generation of wireless broadband systems. To learn more, visit www.limemicro.com.*

REFERENCES

[1] http://fairwaves.ru/wp/wp-content/uploads/2013/10/DSC_8455_small.jpg
[2] http://myriadrf.org/wp-content/uploads/2012/10/MyriadRF1-front.png
[3] http://fairwaves.ru/wp/wp-content/uploads/2013/10/DSC_8381_small.jpg

IN THE NEWS

The *IQT Quarterly* examines trends and advances in technology. IQT has made a number of investments in innovative technologies, and several companies in the IQT portfolio are garnering attention for their unique solutions.

Connectify

Connectify develops next generation networking software for PC and mobile platforms. Its flagship product, Connectify Hotspot, allows any PC to act as a Wi-Fi access point to share a single Internet connection with other Wi-Fi enabled devices and enable groupware on the go. Connectify is based in Philadelphia, Pennsylvania. IQT initially invested in the company in September 2011. **www.connectify.me**

MedShape

MedShape has developed a range of medical devices using novel shape-changing materials. The devices utilize shape-memory alloys and polymers for a variety of orthopedic applications, including reattaching and anchoring connective tissue. MedShape recently introduced a tibial-talo-calcaneal (TTC) fusion nail that returns to its original, specific form after deformation due to its nickel and titanium makeup. MedShape is based in Atlanta, Georgia and joined the IQT portfolio in September 2010. **www.medshape.com**

OpGen

OpGen is a microbial genome analysis company. OpGen recently announced an agreement with Hitachi High-Technologies Corporation to develop a comprehensive human chromosome mapping analytical service for clinical research applications. The service will combine OpGen's market-leading Whole Genome Mapping™ technology and Hitachi's cloud-based systems, and will include bioinformatics tools to complete human genome sequence, and detect and analyze structural variations in human chromosomes. OpGen is based in Gaithersburg, Maryland. IQT initially invested in the company in October 2007. **www.opgen.com**

WiSpry

WiSpry develops and manufactures tunable RF components that can be integrated into wireless devices for improved signal performance. The company recently announced the WS1050 tunable RF capacitor, which combines three individually controllable and configurable capacitors into a single chip. The new chip reduces costs, board space, and design time while improving linear performance and design flexibility. WiSpry is based in Irvine, California and has been part of the IQT portfolio since February 2005. **www.wispry.com**

IQT®
IN·Q·TEL