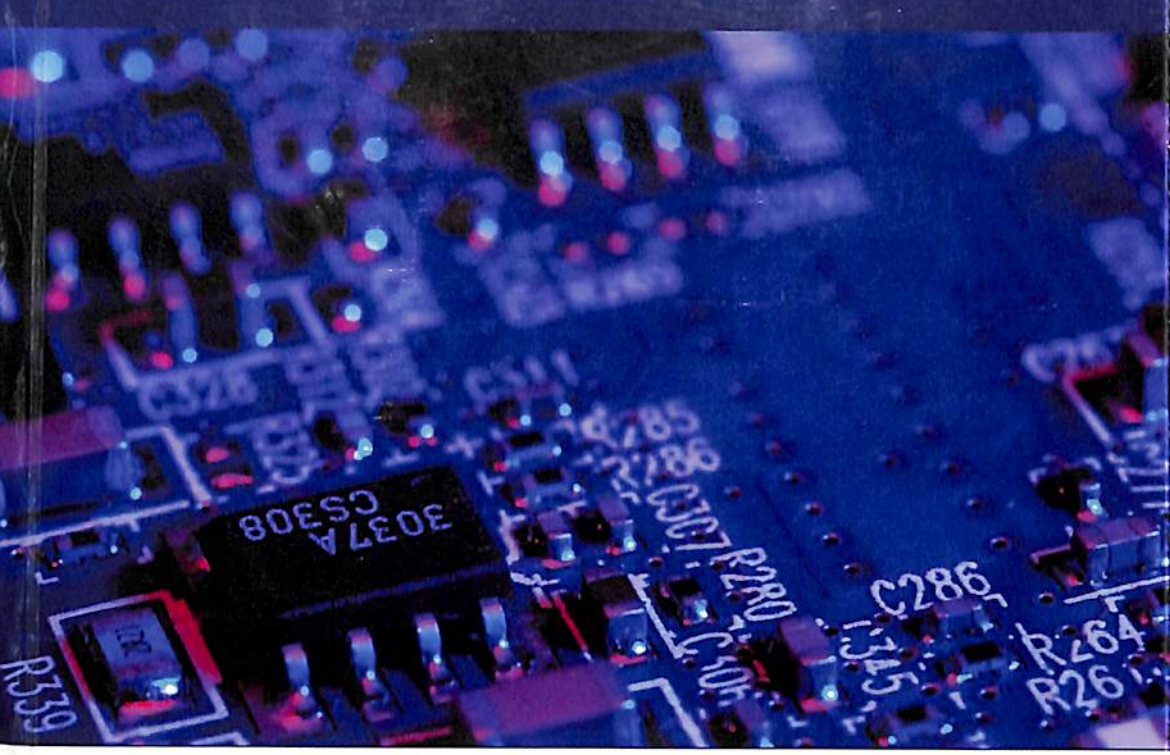

Partha Pratim Ray
Rebika Rai

Open Source Hardware: An Introductory Approach

LAP LAMBERT Academic Publishing

Open Source Hardware:An Introductory Approach covers many aspects of preliminary thoughts on Open Source Hardware. This book is unique of its own kind to present concepts, evolution, and futuristic exploration in the areas of newly emerging freedom on computer hardwares. This book contains design prospects, business models of IP sharing of Open Source Hardware. One of vital topics covered in this book is Open Source Hardware licences where definition and comparisons between the various available licenses have been highlighted. Further, the case studies on the two most promising Open Source Hardware platforms, Arduino and BeagleBoard have been explored. Relationship between Open Source Software and Hardware is presented lucidly. Lastly, another emerging area of Computer Science, Internet of Things (IoT) has been merged with the prospective of enlisting the present scenario of IoT in Open Source Hardware. This book will be useful for the engineering students, practicing engineers, and researches in the relevant field.

Partha Pratim Ray

He pursued Master of Technology in Electronics from the WBUTech, India. Rebika Rai is the Co-Author of this book. She did Master of Technology in Computer from SMITechnology, India. The authors are currently working as Assistant Professors in Computer Department, Sikkim University, India. Authors published several papers in journals of repute.

978-3-659-46591-8

Partha Pratim Ray
Rebika Rai

Open Source Hardware: An Introductory Approach

LAP LAMBERT Academic Publishing

Impressum / Imprint
Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek: The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Coverbild / Cover image: www.ingimage.com

Verlag / Publisher:
LAP LAMBERT Academic Publishing
ist ein Imprint der / is a trademark of
OmniScriptum GmbH & Co. KG
Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Deutschland / Germany
Email: info@lap-publishing.com

Herstellung: siehe letzte Seite /
Printed at: see last page
ISBN: 978-3-659-46591-8

Open Source Hardware:
An Introductory Approach

First Edition

Preface

As the years passed, and computers shrunk in size yet increased in number, there was no medium for programmer to intermingle and trade information. With the rise of internet in late 90's, Open source mechanism brought together the community therein developing a podium where programmers would gather and exchange information to build an open source software / hardware to be used in present as well as future.

Open Source Software is emerging towards the positive stake and the exposure is being made in an effectual manner. Numerous proprietary software companies are identifying the value of the so called "open source". On the other hand Open Source Hardware came in the picture which facilitated the user to grab the information about the hardware for free. The information comprised of hardware design (i.e. mechanical drawings, schematics, PCB layout data, HDL source code and integrated circuit layout data), in addition to the software that drives the hardware, are all released.

We have a sincere and strong belief that without using almost any jargons and mathematics, we can study more aspects of "Open Source" technology. The idea is that even the person with little or no background in computers will be able to grasp the main concepts of it.

- **Orientation:** An attempt has been made to cover the topics in an appropriate sequence.
- **Style:** An attempt has been made to keep the language very simple with the explanation readily understandable.
- **Contents:** An attempt has been made to cover the important features especially the case study on the emerging Open Source Hardware i.e. Arduino and BeagleBoard.
- **Visual Approach:** The book features numerous illustrations and diagram for an easier grasp of the concept highlighted.
- **Pedagogical Features:** Every chapter contains the list of Key concepts and references.

The chapter wise orientation of the book is as follows:

Chapter 1 provides the background material of the subject. The chapter covers the introduction to an Open Source like Open Software, Hardware, comparisons of Open source with free software / closed source, and also highlights the importance of an Open future using various open source software and hardware's.

Chapter 2 deals with the history of Open Source and its evolution.

Chapter 3 focuses on the Open Source hardware development. We take a look at the concepts of Open Hardware and Design Alliance, future of Open Hardware, Elements of Open Hardware projects, Open Source Hardware processes.

Chapter 4 covers the Business models for Open Source Hardware Design highlighting the concepts of IP sharing in hardware and

Software, Strategies for Commodity producers and Economic Motivations.

Chapter 5 introduces the concepts of Open Source Hardware foundation that deals with the aspects of OSHWA (Open Source Hardware Associations), OSHW principles, definitions, OSHW survey.

Chapter 6 explains the various Open Source Hardware Licenses and their comparisons.

Chapter 7 focuses on the Case Study of the emerging Open Source Hardware especially Arduino and Beagleboard.

Chapter 8 explains the relations between the Open Source Hardware and Open Source Software.

Chapter 9 covers Internet of Things that highlights the aspects of IoT Consortium, Cloud Computing and IoT based on Clouds.

A number of appendices provide the solid base of Case studies, Licenses and other techniques.

We hope that the book has a significant use for the professional, teachers and students exploring field of **"Open Source Hardware".**

3rd July, 2013
Gangtok, India

Partha Pratim Ray
Rebika Rai

Acknowledgements

It gives us an immense pleasure to express our deepest appreciation to all those who provided us the possibility to complete this book.

A special gratitude we would give to our Prof. Dr. T. B. Subba, Vice Chancellor, Sikkim University, India whose contribution in stimulating suggestions and encouragement helped us to coordinate our work.

Furthermore we would also like to acknowledge with much appreciation the crucial role of Prof. Dr. J. P. Tamang, Dean (School of Sciences), Sikkim University, India who have been an extreme support and inspired us in all the way.

Last but not least, many thanks go to our family and friend, who have invested their full effort and support that helped us to achieve our goal.

Partha Pratim Ray

Rebika Rai

Sikkim University, Gangtok, India

Contents

Introduction 1

Key Concepts

Open Source | Open Source Software | Open Source Hardware | An Open Future | Open Source | Closed Source | Free Software | Proprietary Software.

1.1 An Open Source

Open source commonly refers to a program in which the source code is accessible to the public for use or alteration from its original design at no cost i.e. open. It emerged in the community as a response to proprietary software owned by corporations. Open source code is typically formed as a joint attempt in which programmers progress upon the existing code and contribute the changes within the community. Programmers on the Internet read restructure and modify the source code, compelling a useful evolution of the product. The process of eradicating bugs and improving the software happens at a more rapid rate than through the traditional development channels of commercial software as the information is shared throughout the open source community and does not originate through a corporation's research and development mechanism.

The open-source model comprises of the concept of simultaneous yet different agendas and approaches in terms of production, in contrast with more centralized models of development such as those typically used in commercial software companies.

The Open Source Initiative (OSI) is a non-profit corporation with global scope formed to educate, advocate about, for the benefits of open source, and to build bridges among different constituencies in the open source community.

1.2 Open Source Software

A main principle and practice of open-source software development is peer production by bartering and collaboration, with the end-product, source-material, "blueprints", and documentation available at no cost to the public. This model is also used for the development of open-source-appropriate technologies, solar photovoltaic technology and open-source drug discovery.

An organizational benefit from the use of Open Source Software includes Reliability, Stability, Auditability, Cost, Flexibility, Freedom, Support and Accountability.

Many people use open source software (OSS) licenses such as "Creative Commons", "GPL" (GNU Public License), "BSD" or "MIT" and other similar OSI approved licenses. There may be a benefit to having a separate /unique license for hardware because hardware can be patented (unlike software) there has recently been calls for open hardware licenses.

Computer software is licensed under free software licenses and open source licenses. Software that fits the Free Software Definition may be more appropriately called free software; the GNU project in particular objects to their works being referred to as open source. However, nearly all software meeting the Free Software Definition also meets the Open Source Definition and vice versa.

1.3 Open Source Hardware

Open (source) hardware is a licensing agreement for electromechanical projects. That is to say, electronics and robotics and

other mechanical projects are open hardware if they are documented and published under an open hardware license.

An open hardware license is basically an agreement by the author(s) of the work that allows other people to use that work for free, without paying royalties, licensing fees, etc. as long as certain constraints are followed. The most common constraints are "attribution requirements," for example requiring anybody who uses the work to place the name of the original author(s) prominently on the final project. Another common constraint is "share-alike" that means that any derivative work must be released under a similar license.

Open-source hardware consists of physical artifacts of technology designed and offered by the open design movement. Both free and open-source software (FOSS) as well as open-source hardware is created by this open-source culture movement and applies a like concept to a variety of components. The term usually means that information about the hardware is easily discerned. Hardware design (i.e. mechanical drawings, schematics, bills of material, PCB layout data, HDL source code and integrated circuit layout data), in addition to the software that drives the hardware, are all released with the FOSS approach.

Various open source hardware licenses are Opencores prefers the LGPL or a Modified BSD License, FreeCores insists on the GPL, Open Hardware Foundation promotes "copyleft" or other permissive licenses", the Open Graphics Project uses a variety of licenses, including the MIT license, GPL, and a proprietary license, and the Balloon Project wrote their own license. New hardware licenses are often termed as the "hardware equivalent" of a well-known OSS license, such as the GPL, LGPL, or BSD license. Also, the noteworthy OOH license includes the TAPR Open Hardware License, Balloon Open Hardware License, Hardware Design Public License, CERN Open Hardware License (OHL), The Solderpad License.

1.4 An Open Future

Years ago, there was no market for proprietary software. At the dawn of the digital era, when a single computer occupied entire rooms, software was not a commodity to be sold. Instead, it was shared, spread around the computer room, used, copied, examined, and improved by anyone interested. An open source community thereby existed.

As the years passed, and computers shrunk in size yet increased in number, the community all but disappeared. There was no feeling of camaraderie among the microcomputer programmers from the late '70s through the '80s. There was no forum for them to interact and trade information.

With the rise of the Internet in the '90s, open source software has again brought together a community. The Internet provides a place where thousands can gather to build open source software for the present and the future. Writing software is again a community movement.

The motivations of those in this new open source community are to write quality software for themselves. Studies have found that those who perform creative work due to intrinsic motivations do better work than those who are promised rewards for performance. This applies to professional programmers as easily as to the studies' subject groups. Open source programmers write what they do because they want to, so they produce better work.

Open source programmers in the community take pride in exhibiting their work to the outside world. It's the same reason that classic car hobbyists take their cars to shows: because they want to show off their talents. Open source programmers polish their software and make it shine.

Open source software is growing in popularity, and the rate of growth is increasing. Publicity is getting to be more and more effective. More

and more proprietary software companies are recognizing the value of open source. For instance, Netscape released Mozilla, an open source version of Communicator, to the public in early 1998. Even Microsoft is hinting that it may make some of Windows NT's source code available to the public.

1.5 Open Source Vs Closed Source

The debate over open source vs. closed source (alternatively called proprietary software) is the vital topic in the market.

The top four reasons (as provided by Open Source Business Conference survey) individuals or organizations choose open source software are: lower cost, security, no vendor 'lock in', and better quality.

Since innovative companies no longer rely heavily on software sales, proprietary software has become less of a necessity. In this way, open source software provides solutions to unique or specific problems. As such, it is reported that 98% of enterprise-level companies use open source software offerings in some capacity.

On the other hand, open-source software does have its own flaws. One of the greatest barriers facing wide acceptance of open-source software relates to the lack of technical and general support. Open-source companies often combat this by offering support sometimes under a different product name. Acquia provides enterprise-level support for its open-source alternative, Drupal, for instance.

Many arguments pops up that open source software is inherently safer because any person can view, edit, and change code, whereas, with closed-source software individuals that do not have the permission to manipulate and update the existing software and it is to be used as per framed by the developers.

1.6 Open Source Vs Free Software

The term "open source" fosters an ambiguity of a different kind such that it confuses the mere availability of the source with the freedom to use, modify, and redistribute it. Developers have used the alternative terms Free/open source Software (FOSS), or Free/Libre /open source Software (FLOSS), consequently, to describe open source software which is also free software.

The term "open source" was originally intended to be trademark able; however, the term was deemed too descriptive, so no trademark exists.

OSI Certified is a trademark licensed only to people who are distributing software licensed under a license listed on the Open Source Initiative's list.

Open-source software and free software are different terms for software which comes with certain rights, or freedoms, for the user. They describe two approaches and philosophies towards free software. Open source and free software (or software libre) both describe software which is free from onerous licensing restrictions. It may be used, copied, studied, modified and redistributed without restriction. Free software is not the same as freeware, software available at zero prices.

The definition of open source software was written to be almost identical to the free software definition. There are very few cases of software that is free software but is not open source software, and vice versa. The difference in the terms is where they place the emphasis. "Free software" is defined in terms of giving the user freedom. This reflects the goal of the free software movement. "Open source" highlights that the source code is viewable to all; proponents of the term usually emphasize the quality of the software and how this is caused by the development models which are possible and popular among free and open source software projects.

References

1. http://www.webopedia.com/TERM/O/open_source.html.
2. Richard Stallman, Why Open Source misses the point of free software, 2011.
3. Steven Weber, The political economy of open source software.
4. Irina Guseva, Bad Economy Is Good for Open Source, Cmswire.com, 2012.

History and Its Evolution | 2

Key Concepts

Open Source Software | Open Source Definition | General Public License | Free Software Foundation | Open Source Initiatives.

The free software movement was launched in 1983. In 1998, a group of individuals advocated that the term free software should be replaced by open-source software (OSS) as an expression which is less ambiguous and more comfortable for the corporate world. Software developers may want to publish their software with an open-source license, so that anybody may also develop the same software or understand its internal functioning. With open-source software, generally anyone is allowed to create modifications of it, port it to new operating systems and processor architectures, share it with others or, in some cases, market it. Scholars Casson and Ryan have pointed out several policy-based reasons for adoption of open source – in particular, the heightened value proposition from open source (when compared to most proprietary formats) in the following categories such as Security, Affordability, Transparency, Perpetuity, Interoperability, Flexibility.

The Open Source Definition, notably, presents an open-source philosophy, and further defines the terms of usage, modification and redistribution of open-source software. Software licenses grant rights to users which would otherwise be reserved by copyright law to the copyright holder. Several open-source software licenses have qualified within the boundaries of the Open Source Definition. The most prominent and popular example is the GNU General Public License (GPL), which "allows free distribution under the condition that further developments and applications are put under the same license", thus also free. While open-source distribution presents a way to make the

source code of a product publicly accessible, the open-source licenses allow the authors to fine tune such access.

The open source label came out of a strategy session held on April 7, 1998 in Palo Alto in reaction to Netscape's January 1998 announcement of a source code release for Navigator (as Mozilla). A group of individuals at the session included Tim O'Reilly, Linus Torvalds, Tom Paquin, Jamie Zawinski, Larry Wall, Brian Behlendorf, Sameer Parekh, Eric Allman, Greg Olson, Paul Vixie, John Ousterhout, Guido van Rossum, Philip Zimmermann, John Gilmore and Eric S. Raymond. They used the opportunity before the release of Navigator's source code to clarify a potential confusion caused by the ambiguity of the word "free" in English.

Many people claimed that the birth of the Internet, since 1969, started the open source movement, while others do not distinguish between open-source and free software movements.[8]

The Free Software Foundation (FSF), started in 1985, intended the word "free" to mean freedom to distribute (or "free as in free speech") and not freedom from cost (or "free as in free beer"). Since a great deal of free software already was (and still is) free of charge, such free software became associated with zero cost, which seemed anti-commercial.

The Open Source Initiative (OSI) was formed in February 1998 by Eric S. Raymond and Bruce Perens. With at least 20 years of evidence from case histories of closed software development versus open development already provided by the Internet developer community, the OSI presented the "open source" case to commercial businesses, like Netscape. The OSI hoped that the usage of the label "open source," a term suggested by Peterson of the Foresight Institute at the strategy session, would eliminate ambiguity, particularly for individuals who perceive "free software" as anti-commercial. They sought to bring a higher profile to the practical benefits of freely available source code, and they wanted to bring major software businesses and other high-tech industries into open source. Perens attempted to register "open source"

as a service mark for the OSI, but that attempt was impractical by trademark standards. Meanwhile, due to the presentation of Raymond's paper to the upper management at Netscape—Raymond only discovered when he read the Press Release, and was called by Netscape CEO Jim Barksdale's PA later in the day—Netscape released its Navigator source code as open source, with favorable results.

Aided in large part by the growth of the Internet, OSS plays a large role in some important applications such as web servers, browsers or e-mail. The successful IPOs of various Linux distributors such as Red Hat or VA Linux has provided them with solid financing and highlighted the links between OSS and commercial vendors. Finally Microsoft, in the suit brought by the US government for alleged anti-competitive behavior, has argued that it was not a monopolist, in particular because it faced strong competition from open source software. All these factors spurred interest in the organizational structure of OSS: cooperation and contributions by geographically far-flung developers who coordinate their activity almost exclusively by e-mails and bulletin boards and who receive no direct monetary compensation for their contributions. The fact that most programmers who contribute to OSS do so without monetary compensation has been puzzling to economists.

Actually, the phenomenon is much older than the discussion of the two preceding paragraph makes it appear. During the "mainframes' age", sharing of code by programmers in different organizations was already highly developed. Then, in the 1970s, AT&T Bell Laboratories devoted much effort to the Unix operating system and to the C language. When AT&T announced plans to commercialize Unix in early 1980s, the University of California at Berkeley decided to create its own version of Unix called Berkeley Software Distribution (BSD) Unix, which was adopted by early commercial vendors such as DEC and Sun.

In reaction to the proprietary trend in software, Richard Stallman of MIT Artificial Intelligence Laboratory established, in 1983, the Free Software Foundation to promote free operating systems, applications and programming tools developed under the General Public License

(GPL or copyleft). In 1986 Larry Wall released PERL, the language that is behind most of the "live content" on the web.

After the release of Minix, a version of UNIX for the PC, Mac, Amiga and Atari ST, by Andrew Tannenbaum in 1987, Linus Torvald released a new variant of Unix, Linux, in 1989 which he copylefted in 1992. Then the story speeded up from the creation in 1994 of Red Hat, now a major Linux distributor, to the release of Netscape's source code in 1998 and it is continuing. It is quite striking that some OSS projects are large and complex and still have succeeded in achieving commercial quality. A number of open source products, such as Apache or Sendmail, dominate the market in their categories.

References

1. Eloic Peyrache, Jacques Cremer, Jean Tirole, Some reflections on Open Source Software, University of Toulouse.
2. Ghosh R. and Ved Prakash V, The Orbiten Free Software Survey, 2000.
3. Mockus A., Fielding R. and Herbsleb J., A case study of Open Source Development: The Apache Server, 2000.
4. Torvalds L., What motivates free software developers, 1999.
5. Valloppillil V, Open Source Software, a (new?) Development Methodology, 1998.