

# Towards a Functional Licence for Open Hardware

*Author: Andrew Katz, Moorcrofts LLP*

*Andrew Katz, Partner, Moorcrofts LLP Solicitors, James House, Mere Park, Dedmere Road, Marlow, Buckinghamshire SL7 1FJ, UK*

DOI: [10.5033/iffosslr.v4i1.69](https://doi.org/10.5033/iffosslr.v4i1.69)

## Abstract

Open hardware lags open source software in maturity. The two main licences assert a form of copyleft. This paper argues that copyleft's applicability to hardware is problematic, and concludes by proposing a simpler non-copyleft licence, based on Apache 2.0, for hardware.

## Keywords

Open Hardware, Open Source Hardware, Licensing, Copyleft

There are currently two main licences vying for serious consideration as open hardware licences. They are the TAPR<sup>1</sup> Open Hardware License, and the CERN<sup>2</sup> Open Hardware Licence. Both of these licences are intended to assert a form of copyleft on open hardware, the intention being that, as with free software, open hardware must be distributed in a way that guarantees availability of the underlying design documents, and provides the right to reuse, adapt and redistribute them, with the same rule applied downstream as those designs and hardware based on them are re-distributed. Once open hardware has become free, there is, in this philosophy, no way of closing the design again.

Like free software advocates, many open hardware designers are concerned with entities “stealing their designs” by turning the design proprietary: a concern similar to that expressed by proponents of the GNU General Public License (GPL) who wish to avoid free software becoming non-free. (Other open hardware designers are more concerned that their designs are made as easy to use – from a licensing perspective – as possible and believe that, like many open source software advocates, the best way to achieve this is by attaching a licence which does not restrict entities from incorporating their open designs into closed proprietary designs or from merging them with projects which have adopted a different form of open licence).

---

<sup>1</sup> <http://www.tapir.org/ohl.html> (all URLs in this paper were accessed on 11 April 2012)

<sup>2</sup> <http://www.ohwr.org/projects/cernohl/wiki>, and see Myriam Ayass's article in this issue Ayass, Myriam; Serrano, Javier, (2012) 'The CERN Open Hardware Licence', *IFOSS L. Rev.*, 4(1), pp. 71 - 78 DOI: [10.5033/iffosslr.v4i1.65](https://doi.org/10.5033/iffosslr.v4i1.65)

When drafting the GPL, Richard Stallman cunningly tweaked the copyright licence bargain to make it a licence condition that GPL code, when distributed, would itself be subject to the GPL, and because the condition impinges on derivative works as well<sup>3</sup>, the ecosystem of GPL works would continually expand, guaranteeing an ever larger pool of free software<sup>4</sup>. Open hardware designers have been attracted to this mechanism, and have tried to attach a copyleft-style licence to hardware. The CERN and TAPR licences are both examples of this approach.

This paper argues that there are significant problems in applying a form of copyleft to hardware and that a more practical way forward is to use a permissive, non-copyleft form of licence. The Apache 2.0 license suggests itself as one starting point. Adapting it for hardware would avoid many of these problems, and have the additional advantage of a licence which is familiar, well understood and respected.

## What is Hardware?

Not surprisingly, hardware projects which have been considered as candidates for openness are typically electronic devices, but there are counterexamples. The author first became involved in open source hardware through an open car project, the Riversimple<sup>5</sup> Hyrban hydrogen fuel-cell car.

Thinking about open hardware in terms of mechanical devices is a useful thinking tool. Electronic devices, especially those with programmable components, are more akin to software than hardware, and it's more effective to think about open hardware in terms of more traditional mechanical devices.

A licence drafted to cover the relatively narrow scope of electronic designs may not be appropriate to more traditional “heavy metal” hardware.<sup>6</sup> It helps to consider a number of different use cases when examining the effectiveness of an open hardware licence: software, FPGAs, analogue electronic circuits, hydraulic and fluidic circuits, mechanical memories,<sup>7</sup> mechanical sub-assemblies, stormtrooper helmets and Michelangelo's David being examples on a loosely defined spectrum of hardware in order from the “softest” to the “hardest”. An effective open hardware licence should address the full range of hardware (and frequently will also, incidentally, address associated software).

---

3 Whether GPL2 intends to impinge on a wider subset of works than those that are simply derivative is a matter of debate, for example, see *The Time Travel Problem* below. GPL3 is more explicit in that it intends to be limited to derivative works only.

4 Open source advocates may argue that freeing software by coercion is unnecessary and counterproductive.

5 <http://www.riversimple.com/>

6 For example, a circuit board layout is a 2D graphic work which retains its 2D nature when reproduced from a mask. Under the UK Copyright Act, copying a 2D work in 2D is a restricted act, whereas reproduction of a 2D work (unless it is an artistic work), is not a restricted act. Section 51, Copyright, Designs and Patents Act 1988.

7 The Friden Flexowriter, an entirely electromechanical device, had a mechanism of rods and cams which enabled it to translate alphanumeric characters to and from a paper tape punched in Baudot code. That mechanism undeniably amounted to a mechanical ROM, the content of which would, presumably attract copyright as an independent literary work, and possibly database right as a database of mappings of characters against code).

## The TAPR Open Hardware License

The TAPR Open Hardware License was drafted by John Ackermann, an attorney and radio amateur<sup>8</sup>, as a response to designers of electronics comprising the Tucson Amateur Packet Radio System. It is expressly intended to apply copyleft to hardware, and details of its rationale and drafting process can be found in Ackermann's article in the *University of Dayton Law Review*<sup>9</sup>. Ackermann is refreshingly candid about the challenges arising from applying copyleft to open hardware. A copy of the licence can be found in the article<sup>10</sup>

The licence is expressly intended to be a contract (unlike the GPL, which was drafted to be a bare licence subject to conditions<sup>11</sup>). It primarily deals with design documentation (which includes CAD files, board layouts and mechanical drawings), requiring anyone who uses design documentation of covered hardware to comply with the licence obligations, and specifically, to make the design documentation (including any modifications to them) available to any recipient of the hardware (there are also obligations to attempt to pass details of the obligations back to the upstream licensors).

Clauses of particular relevance are:

*1.5 By (a) using, copying, modifying, or distributing the Documentation, or (b) making or having Products made or distributing them, you accept this Agreement, agree to comply with its terms, and become a "Licensee."....*

and

### *Making Products*

*5.1 You may use the Documentation to make or have Products made, provided that each Product retains any notices included by the Licensor (including, but not limited to, copyright notices on circuit boards).*

*5.2 You may distribute Products you make or have made, provided that you include with each unit a copy of the Documentation in a form consistent with Section 4. Alternatively, you may include either (i) an offer valid for at least three years to provide that Documentation, at no charge other than the reasonable cost of media and postage, to any person who requests it; or (ii) a URL where that Documentation may be downloaded, available for at least three years after you last distribute the Product.*

These clauses raise some issues which are referred to below. The other clauses of the contract are similar to a FOSS licence, and cover limitation of liability, patent licensing and so on. As they are not pertinent specifically to copyleft they are outside the scope of this paper. The licence provides

<sup>8</sup> N8UR

<sup>9</sup> [http://www.tapr.org/Ackermann\\_Open\\_Source\\_Hardware\\_Article\\_2009.pdf](http://www.tapr.org/Ackermann_Open_Source_Hardware_Article_2009.pdf) (2009) Volume 34, number 2, page 183

<sup>10</sup> And here: [http://www.tapr.org/TAPR\\_Open\\_Hardware\\_License\\_v1.0.odt](http://www.tapr.org/TAPR_Open_Hardware_License_v1.0.odt)

<sup>11</sup> Although whether it is a bare licence or contract is not a matter for the Free Software Foundation, but a matter of judicial interpretation from jurisdiction to jurisdiction.

that any software (including firmware) in the project is not covered by the licence, but is governed by whatever (generally open source) software licence is applicable to it<sup>12</sup>. There is also a non-commercial version of the TAPR License, but like the Creative Commons *non-commercial* option, this is neither a free nor open source licence.

John Ackermann has indicated that he is happy to engage in an updating exercise for the TAPR licence.

## The CERN Open Hardware Licence

The CERN Open Hardware Licence, discussed elsewhere in this issue, has a similar aim. Also primarily covering design documentation, it has undergone a more structured revision process and is currently at version 1.1. The next version is currently under active discussion.

A clause of particular concern is:

*4.1 The Licensee may manufacture or distribute Products always provided that the Licensee distributes to each recipient of such Products a copy of the Documentation or modified Documentation, as applicable, and complies with section 3.*

This raises issues which are discussed below.

## Copyleft and Open Hardware

Copyleft in software has detractors: from the proprietary software companies who see it as a “viral” mechanism which could “infect” their precious proprietary codebase, to the proponents of an open, permissive development model, who argue that openness does not need to be forced, but, as a better model, openness will inevitably succeed. The arguments as they relate to software relate also to hardware, but there are some differences in emphasis, as well as arguments apply solely to hardware. These arguments are explored below.

## The Legal Arguments

For there to be a licence, there first must be something unlawful.

A licence is a permission to do something which would otherwise be unlawful. It therefore follows that there has to be something unlawful in the first place, for which the licence can grant permission. In software licences, the rights granted are, in the main, related to copyright, and permissions are permissions to do what would otherwise be contrary to copyright.

Copyright impinges on software at almost every stage of its use and exploitation: the acts of

---

<sup>12</sup> Although there are edge cases – for example the programming of FPGAs, where the boundary is not clear. See section *Boundary Problem* below.

running and copying the software are controlled by copyright. The extent to which distribution is governed by copyright is covered in Heather Meeker's article in this edition.

The upshot is that, at each of these stages, permission of the copyright owner is required to avoid breaching copyright law, and thus at each point, the copyright owner has the opportunity to grant a licence and apply licence conditions. The authors of software licences take advantage of this in order to exercise control at many different times.

Richard Stallman and Eben Moglen notably exploited the opportunity to apply conditions to the distribution of software in the GPL, and provided that any distribution of GPL code must be accompanied by (or allow access to) the corresponding source.

This does not apply to hardware to anything like the same degree. Running a software program like a spreadsheet requires a software licence. Using a hardware device like an abacus (or a difference engine)<sup>13</sup> does not.

Hence, any licence which tries to echo the GPL by requiring the distribution of hardware to be accompanied by the source will necessarily be limited in its effectiveness by virtue of the extensive opportunities for making use of the underlying design without having to rely on the licence.

For example, if A possesses a piece of object code which is only available under the GPL, unless A has already violated the GPL, A will be able to make free use of that code herself. It may be the case that A does not possess the corresponding source (either because A's provider didn't provide it to her under the GPL, or A has not requested the source from the provider). However, A will be unable to distribute that code to any third parties, unless A can fulfil her own obligations under the GPL by delivering the source code to the recipient, or making it otherwise available in compliance with GPL.<sup>14</sup>

However, generally speaking,<sup>15</sup> using a piece of hardware, or transferring a piece of hardware from one person to another does not potentially contravene any intellectual property rights, and therefore does not require any licence on which copyleft-type requirements can impinge. This makes it difficult to effectively implement a copyleft licence for hardware which is effective, if the trigger is to be distribution of the physical hardware. For application of the copyleft to the design documents, see below.<sup>16</sup>

<sup>13</sup> But possibly not an analytical engine

<sup>14</sup> To muddy the waters, this also suggests a possible way of circumventing the GPL: under the Computer Programs Directive (2009/24/EC), where a copy of a program has been put into circulation in the European Economic Area with the consent of the copyright owner, the copyright owner's right to control further circulation of that copy ceases. How, therefore, can the GPL be enforced if a person who receives a copy of a GPL program and does not obtain the source, then passes the copy to another person, relying on the directive? There are a number of counterarguments to this but a paper on open source hardware is not the place to discuss them.

<sup>15</sup> Patents and some design rights may complicate this argument.

<sup>16</sup> One question, which has not been extensively explored, even if the distribution of a piece of hardware itself does not require a licence under an intellectual property right, can such distribution cause the distributor to lose rights under an existing licence which it *does* require for other purposes, as a means of enforcing the licence. For example, can the licence to replicate the source code to code W be lost if the licensee fails to distribute a related piece of software S in accordance with copyleft, even if that piece of software S is not a derivative work of W, and therefore the licence of W's proprietor is not required for the distribution.

An option which can be considered is whether a contractual mechanism can be applied. In effect, each licensee is contractually obliged to impose a licence on a subsequent owner of the hardware, where that licence requires the subsequent owner to comply contractually with the terms of the licence, and to only pass derived hardware onto a third party once that third party has been bound in a similar manner.

The difficulty is that the contract creates an *in personam* relationship between the parties, so that the recipient of the derived hardware from the licensee will only be bound if the licensee has fulfilled his part of the bargain with the licensor and imposed the licence terms on the third party.<sup>17</sup>

If the licensee is in breach and fails to impose the contract on the subsequent recipient, then the licensor may have a claim against the licensee for breach of contract,<sup>18</sup> but will have no right of action against the recipient under contract, who will then be free and clear to pass the hardware on free of any contractual restriction. Thus the chain of contracts will become long and fragile, and any failure to impose contractual terms on a subsequent recipient will break the chain. It is also arguable that requirement to impose contractual terms on third parties is, in itself, an unenforceable restraint of trade.<sup>19</sup>

### The Time Travel Problem

If a licensee fails to comply with a licence condition, then it is axiomatic that the licensee is in breach of copyright. However, the situation is not always quite so simple. It is usually the case that it is possible to determine whether the condition is fulfilled at the point when the otherwise infringing act takes place. For example, where a licensee under the GPL passes a complete copy of the relevant source code alongside the binary, the distribution is the restricted act, and the attached condition is fulfilled by providing the source at the same time. However, licences sometimes fall into the trap of trying to make a licence conditional on the licensee *doing something in the future*.

It would be a bizarre but functional condition of a licence agreement to assert that the licensee has to wear a bowler hat while making a copy of design documentation. However, it would be difficult to see how it would be possible to make it a licence condition that a licensee can copy design documentation so long as he will be wearing a bowler hat *in a week's time*. In the intervening week between copying, and the requirement to wear the hat, is the licensee in a state similar to that of Schrödinger's unfortunate cat, neither infringing nor non-infringing, until the point occurs, in a week's time, at which compliance with the condition can be determined?

By adopting a wording similar to that contained in copyleft licences like the GPL, which make distribution contingent on a number of conditions (for example, to make the source code available), the drafters of both hardware copyleft licences have unwittingly fallen into this trap. The assumption<sup>20</sup> is that distribution of a copyright work requires the licence of the copyright

17 And, in England and Wales, until the passing of the Contracts (Rights of the Third Parties) Act 1999, it would, owing to privity of contract, be difficult to establish a mechanism by which the licensor would have a direct contractual right of action against the licensee, although possibly some form of agency could have been applied.

18 The remedy for this will be damages, which will be difficult to assess, given that the licensor has made it clear his willingness to license on a zero-cost basis.

19 e.g. *Dunlop Pneumatic Tyre Co Ltd v Selfridge & Co Ltd* [1915] UKHL 1

20 Unfortunately, this assumption is not necessarily correct, and the question of whether a specific distribution is a

owner, and that, accordingly, the restricted act of distribution can only be carried out if it can be determined that the condition is fulfilled at the time the distribution is taking place. The GPL, therefore, to that extent, works, because the condition can be determined at the point that the licence is being relied on<sup>21</sup>.

However, it's not quite so simple to apply this to hardware. If distribution of hardware is not, in itself, a restricted act under copyright law, then a condition, like that contained in clause 4.1 of the CERN OHL, or clause 5.2 of the TAPR OHL, is difficult to interpret.

Either it is a condition, breach of which has the effect of terminating (or allowing the licensor to terminate) the licence, or it is a condition which somehow retrospectively makes previous acts of the licensee (like copying the documentation) become unlawful (as described above). A third possibility is that the clause is a contractual obligation (for example, you agree that if you distribute the hardware, you will also, as a contractual obligation, transfer the design documentation).

Looking at each of these in turn:

If breach of the condition allows termination of the licence, then it does not itself make the specific restricted act unlawful, but may make subsequent restricted acts unlawful. We have already established that manufacture of the hardware itself (and distribution of it) may well *not* be a restricted act. No licence is needed, so although the licensee will be restricted from making copies of the design documentation, for example, this is not going to provide a major hindrance to exploitation of the hardware<sup>22</sup> as it may well remain possible to continue to manufacture the hardware without being in breach of the licence. However, copying and adapting the design documentation would no longer be lawfully possible (and it may be that this residual copyleft effect is sufficient to provide the normative effect that the licensor will be seeking).

If it is a condition with retrospective effect, then the time travel problem arises.

If it is a condition which is (solely) a contractual obligation, then this may place a specific obligation on the licensee, but it still only creates an *in personam* right as considered above, and does not have effect as an operative condition on third parties, because the act in question is not a restricted act under copyright law<sup>23</sup>.

---

restricted act is likely to vary from jurisdiction to jurisdiction. GPL3 tries to clarify this issue by using “propagate” and “convey” and limiting them to acts which are specifically restricted under copyright law.

- 21 However, this does leave open the question as to what happens where the licensee opts to offer to make the corresponding source available for a period of three years, and fails to honour that offer. Thankfully, this is not a paper about GPL enforcement.
- 22 Another issue with termination is that licences rarely explicitly deal with termination (although CERN OHL does). Automatic termination is always problematic, as inadvertent breaches are easy, and will trigger automatic termination. Termination with notice has its own problems (the licensor needs to know about the breach, for one thing). Also, how easy is it for the licence to be reinstated? Does the termination apply to all instances of that licence irrespective of the hardware it applies to? All instances of that licence for any iteration of that hardware? All instances of that licence for one iteration of that hardware? Or just the specific instance of that licence as it was applied to the specific design documents downloaded at a specific time (so that re-downloading the design documents would reinstate the licence)?
- 23 The wording of both the TAPR and CERN licences is not effective to make the obligations contractual. They both say, in effect, the licensee *may* manufacture and distribute the hardware, *provided that* she also makes the design documentation available. If the permission (“you may”) is not required (because the underlying act is not contrary to copyright law, or other intellectual property law), the condition (“provided that”) is irrelevant, and can be ignored.

These uncertainties cause problems for an effective copyleft hardware licence. They can be addressed to an extent by drafting<sup>24</sup>, but the problems they raise provide another reason to question the appropriateness of copyleft in a hardware context.

### Copyright and the Design Documents

It's important to distinguish between the hardware itself and the associated design documents. The design documents will generally be subject to copyright, and reproduction, adaptation and distribution of design documents to the public will therefore require a copyright licence. Thus any appropriate document licence such as one of the Creative Commons licences or the GNU Free Documentation Licence can be applied, with copyleft adopted (or not) accordingly. However, documentation licences do not, in themselves, require the distribution of the design documents with the related hardware.

### The Boundary Problem

For a copyleft mechanism to work, there needs to be a clear boundary, such that certain interactions between a copyleft piece of software ("CS") and a non-copyleft piece of software ("NCS") mean NCS can only be distributed subject to the copyleft licence, and certain other interactions allow NCS to be distributed free of the copyleft licence (but subject of course to whatever other licence, if any, may be required in respect of NCS).

Typical technical questions in a software context are, "does copying a snippet of GPL and incorporating it into my app require me to distribute the whole app under the GPL", or "does dynamically linking my app to a GPL library require me to distribute the whole app under the GPL?" There is much debate about this.<sup>25</sup> In the world of hardware it is a significantly greater problem. The types of interaction are much greater: would bolting a copyleft wheel to your proprietary car mean that (assuming hardware copyleft is possible) you could not sell your old car without being able to provide the design document to the wheel, or the whole car?<sup>26</sup>

---

Better wording (to make this work contractually) would be "You are under a contractual obligation to provide the design documentation to any recipient of the hardware [by an appropriate means]". It may even be possible, in some jurisdictions, to make the requirement enforceable by recipients under third party beneficiary doctrine. For example, recipients would be a permissible class of potential enforcers under the Contracts (Rights of Third Parties) Act 1999 in England and Wales.

24 One possibility the author is considering, in connection with the CERN licence, is that undertaking any restricted act (such as copying the design documents themselves, creating derivative works, or, if restricted, making articles and distributing them) is conditional on the licensee having made available to the public the complete design documentation from an easily locatable and publicly indexed place. As it stands, this is onerous (any trivial act of copying, or amending the design documentation, even before distribution, would trigger the condition), so the licensor undertakes not to *enforce* the terms of the licence unless and until an article made to the design (or part of it) has been passed to the public (and only in relation to breaches taking place after the passing of the design). This sidesteps the time travel problem, as the licensee would technically be in breach, but would be safe from enforcement until the design, or its instantiation in a physical object, had been passed to the public.

25 See the linking interactions document referenced in Bain, Malcolm (2010) 'Software Interactions and the GNU General Public License', *IFOSS L. Rev.* 2(2), pp 165 – 180 DOI: [10.5033/ifoossr.v2i2.44](https://doi.org/10.5033/ifoossr.v2i2.44) which is devoted to discussing in-depth interactions solely in relation to one version of one licence: GPL2.

26 If it's possible to effectively place restrictions on the on-sale of a car in this way, or if freedom advocates successfully lobby for the implementation of laws which enable this to be possible, this suggests the unintended consequence of providing a framework to enable car manufacturers, for example, to quash the used car market other than through authorised dealers. Alternatively, the advocates will find themselves arguing a point which sounds like (or can easily



Very quickly, a restriction on the on-sale of any complete item containing maybe only a single copyleft part would become stifling. The alternative is to have a copyleft licence where the scope and extent of the copyleft is restricted in some way, possibly to a specific sub-assembly, by analogy with the file-level copyleft applied by the Mozilla licence. However, a “file” is a relatively well understood concept in computer science. An “assembly” in terms of mechanical engineering is less well understood. Is the assembly the wheel, the wheel+tyre, the wheel+tyre+hub, the wheel+tyre+hub+stub axle, etc.?<sup>27</sup>

A way of dealing with this (suggested by Myriam Ayass as part of the ongoing development of the CERN licence) is to deal with the boundary issue in terms of the design documentation: by requiring the recipient to pass on changes to the design documentation only at the same level of abstraction as the original design documentation was received, this means that there is no need to provide greater detail. In other words, an electronic circuit diagram can be amended and redistributed without having to provide details of how to manufacture the individual components. Tying the copyleft to the design documentation also helps as regards incorporation of sub-assemblies into larger assemblies. If the copyleft only applies at file-level, it becomes more akin to Mozilla-style weak copyleft, and is more easily manageable.

### Open Hardware and Open Source Hardware

The OHANDA<sup>28</sup> four freedoms, based on the four freedoms of the Free Software Foundation<sup>29</sup>, are:

*Freedom 0: The freedom to use the device for any purpose.*

*Freedom 1: The freedom to study how the device works and change it to make it to do what you wish. Access to the complete design is precondition to this.*

*Freedom 2: Redistribute the device and/or design (remanufacture).*

*Freedom 3: The freedom to improve the device and/or design, and release your improvements (and modified versions in general) to the public, so that the whole community benefits. Access to the complete design is precondition to this.*

Freedoms 0 and 3 require “access to the complete design”. Unfortunately, “complete design” does

---

be portrayed as) special pleading, and tying themselves in logical knots, like Richard Stallman did when he found himself arguing with the Pirate Party that their proposals to shorten the copyright term should be subject to a special *longer term of copyright* for free software, to enable copyleft to continue to function

<http://www.gnu.org/philosophy/pirate-party.html>

27 A possible solution does suggest itself in that the licensor can describe the scope of the “assembly” in granting the licence, and that subsequent licensees can expand that scope but not contract it. However, this is very reliant on the licensor and subsequent licensees coming up with a sensible definition of the scope, and makes licence hygiene complex in terms of determining whether a particular project is in compliance with all the licences relating to relevant sub-assemblies. The lower end of the scope also needs consideration. Does a full materials-science description of the metal alloys comprising the wheel need to be provided? An effective copyleft hardware licence will need to address this issue.

28 ohanda.org

29 fsf.org

not neatly map onto “source code”. A kit of parts to make an electronic egg-timer may consist of a circuit board and a number of discrete components such as LEDs, transistors, and an integrated circuit like a 555 timer. It may be “obvious” to some that in this context the complete design consists of the schematic, a list of standard components, and the board layout, but why does “complete design” not include a description to make the 555 timer (a very simple device in IC terms, but nonetheless a little black box – literally – with 8 conductive legs sticking out of it)? Whether or not this is “obvious” in the context, the question is much more difficult when considering a car. Is it acceptable to specify a standard, widely available type of electric motor for the starter motor, or is it necessary to also provide the schematics of the motor, including the materials from which the armature and bearings are made, the torque of the nuts used to secure it, the precise composition of the copper windings, and details of the materials used in lubrication, so that an engineer with access to a handy oil well, copper mine, refinery, smelter and machine shop can effectively manufacture the car from scratch?<sup>30</sup>

This is an issue for whether a particular piece of hardware complies with the OHANDA definition, but it also impinges on licence, and in particular copyleft hardware licences, because release of the complete design documentation is likely to be a condition of distribution of hardware for compliance with such a licence.

It also means that the number of compliant components available to build a compliant assembly will, of necessity, be very small, if each component needs to be available with complete design documentation. In effect, without a practical constraint, the design documentation for an open source car will require every piece of information required to synthesise the car from a bunch of atoms of the appropriate elements used in its construction. Not even Ford and General Motors have access to that amount of information.

Accordingly, a degree of realism needs to be employed, and one way to do this is to distinguish between *open hardware* and *open source hardware*.

“Open hardware” means hardware components which are readily available (whether commercially or otherwise) and for which all relevant specifications are known, such that if (without necessary access to the original design materials) someone created a component compliant with the relevant specifications, it would work in the main assembly for all expected use-cases and environmental considerations applicable to the main assembly (it also requires that such use can be made without impinging on any intellectual property rights for that use-case). Open hardware will not, in itself, be OHANDA compliant. Standard electronic components, such as 74 series ICs, transistors, resistors, capacitors etc. will generally be open hardware, but not themselves OHANDA compliant.

“Open source hardware” is any hardware which fulfils the OHANDA criteria, where “complete design documentation” means the documentation required to build the assembly from components which are either themselves open source hardware, or are open hardware.

This is consistent with requiring that a piece of software have access to a library compliant with a

---

<sup>30</sup> Of course, the equivalents in the world of software, such as GCC, can be downloaded for free and run on a very modest computer.

specific, published API. The 555 timer is a good example of a piece of open hardware, which is not necessarily open source hardware. Its specifications are known in great detail: clearly its electrical specifications are of great importance, but its physical dimensions, operating temperature range and so on are also important and necessary to enable the item to be regarded as open hardware.

It is submitted that this distinction between open hardware and open source hardware provides practical benefits in a licensing context by suggesting a way in which a copyleft hardware licence (if otherwise feasible) can be constructed which provides a practical way of determining where the boundary of design information lies: namely that design documentation of the assembly must be provided, but that the assembly can consist of components which are open hardware, and therefore only their specification, and not their design documentation, need be revealed.

There still remain, however, arguments which may militate against the effective adoption of an open hardware licence, even if legally feasible, and the boundary problem is solved.

## The Economic Argument

Software, as bits, costs essentially nothing to copy. Physical items, no matter how simple, will require a number of atoms of one element or another to be reconfigured, and this resource will cost money, in terms of raw materials, components or sub-assemblies.

Both physical items and software can be reverse-engineered, (ignoring patents for the moment) and a clean-room non-infringing re-write or re-creation can, in both cases, be produced.

If B wants a piece of software with identical functionality to the Linux kernel, but without pesky GPL restrictions, then there is nothing preventing him from reverse-engineering the Linux kernel, and employing an army of software engineers to create an independent work with the same functionality, based on the functional specification obtained from the reverse-engineering process.

Because B is recreating the ideas, and not the expression, of the Linux kernel, B is not infringing copyright in the Linux kernel<sup>31</sup>. The cost of B achieving this epic task will, clearly, be enormous. The cost to B, however, of obtaining a kernel of identical functionality to the Linux kernel is, obviously, infinitesimal, if B is prepared to live with the restrictions of the GPL and adopt the Linux kernel itself.

Thus there is likely to be a cost differential of several orders of magnitude between choosing to circumvent the GPL by reverse-engineering, and choosing to accept its restrictions. The financial incentive to accept the GPL's restrictions is vast.<sup>32</sup>

Hardware is different. For any piece of hardware, there will already be a cost involved in sourcing the raw materials. Assembling atoms is much more expensive than assembling bits. The cost differential, therefore, is likely to be much smaller in proportion.

---

<sup>31</sup> The author fervently hopes this is still the case.

<sup>32</sup> But not completely insurmountable. This is, presumably the metric that Google employed when it decided that it wanted a functionally equivalent piece of software to J2ME (itself licensed under the GPL), and developed Dalvik.

Copyleft relies on this gulf between the cost of replication and the cost of circumvention. Where the cost differential is smaller, the incentive for the replicator to comply with the copyleft licence rather than go to the effort of reverse-engineering, is similarly smaller. It is easy to come up with examples, of course, where reverse-engineering the software is trivial, and reverse-engineering the hardware is difficult, but the general principle remains that a mechanical sub-assembly will frequently be easier to replicate without reference to the underlying design drawings, than a piece of software.

Once the replicator has created its own version of the hardware after the reverse-engineering process, it will then be free and clear to exploit and license that as it sees fit (and less likely to contribute back to the community than it would have been had the original designs been available to it under a non-copyleft licence).

## **There Can Be Only One**

The central premise of copyleft is that distribution of a work and its derivatives has to be on the same outlicence as the one under which it was in-licensed. Thus an app which is based on a GPL2 program can only be out-licensed under GPL2. This means that, unless licensed under GPL2-or-any-later-version (GPL2+), a software project cannot even be licensed out under, or combined with, any GPL3 code, let alone out-licensed under any non-GPL2 licence such as Apache, BSD or the Open Software Licence. Although this is a well understood problem, and some efforts are being made to tackle it (drafting in the EUPL, Mozilla 2.0 and GPL3 aims to ease the situation, with varying degrees of success), fundamentally, copyleft projects are forever stuck in an artificially reduced ecosystem of projects with compatible licences. It is likely that the FSF would enjoy seeing all software copyleft projects (and, probably, all software projects, period) licensed under GPL3+, but for as long as there are incompatible copyleft projects, this is unlikely to happen. For example, it's difficult to see how the Linux kernel, licensed under GPL2, will ever move away from GPL2, given that the consent of all of the many thousands of copyright owners, (or the re-coding of the work of unwilling owners), would be required to a move to another licence, even if it were GPL3. If a copyleft licence needs to exist, ideally, there should be only one: every new copyleft licence creates a new ecosystem which cannot interact with the other ecosystems and much of the benefit of free and open source software is accordingly lost.

A permissive, academic licence does not suffer from this problem, and accordingly the ecosystems are able to intermingle, with the attendant benefit of network effects.

## **Licences and Community**

In F(L)OSS, licences are widely regarded as being a manifesto for a particular community. Thus the GPL presents a mechanism for guaranteeing the freedom of software. It is championed by those who regard proprietary software as immoral, and an unjustifiable enclosure of the commons of human knowledge. The Apache licence is intended to permit the maximum use of software issued under it. Apache proponents believe that open source is a great way to develop software, and that companies seeking to incorporate Apache code into proprietary software will frequently

realise that to get the most value from the code, active engagement with the community is essential, and that this means, in practice, contributing back, whether in terms of code itself, or in terms of bug spotting, documentation, training and so on. Other licences have more subtle nuances (in terms of the way they deal with patents, for example). In each case, the licence reflects the values of the community that uses it. It is too early in the development of open hardware for communities to coalesce in the same way they have for open source software. However, broadly, those who see merit in the approach “we want our designs to be as broadly used as possible, and we don't care if they are used for proprietary purposes” are likely to be attracted to a permissive licence, and those who are more concerned about retaining freedom are more likely to select a copyleft-style licence. If copyleft for hardware turns out to be ineffective, how will this affect freedom-enforcers? There may be scope in a future article to investigate how other mechanisms, such as non-enforceable community norms or application of some form of certification/trademark to compliant designs and hardware, may prove to be effective. It may be dangerous to assume that the world of open source software can be closely mapped onto open hardware.

## The Solderpad Licence

Leaving copyleft to one side, open hardware already has a number of barriers which open software does not: replication will always cost a material amount of money; the equipment required to replicate hardware is likely to be much more expensive than the cost of a simple computer and compiler/IDE; the vastly greater length of the test/fix/test cycle for hardware; the necessity of physical space for creating hardware; and difficulty of transporting hardware as opposed to bits; the challenges of collaborating effectively on hardware at a distance; the relative paucity of free and open source tools for CAD, CAM etc; the expense of testing; the complex regulatory regime around hardware certification being just a few. To introduce a number of additional hurdles suggested by copyleft seems foolhardy, unless the benefits are clear.

Unfortunately, the hoped for benefit, of preventing free riders, may not, in the light of the issues discussed above, be beneficial, especially where it may also have the effect of making an already small ecosystem even smaller.

Given the questionable effectiveness of copyleft in hardware, is it not simpler to avoid the issues entirely, and develop a non-copyleft, permissive licence? Having searched for an appropriate non-copyleft licence for hardware, and failed to find one, the author undertook to create one.

There are three ways to approach the drafting of an appropriate permissive licence: take an existing copyleft open hardware licence, and repurpose it as a non-copyleft licence, take an existing permissive open source software licence and redraft it, or draft a new licence from scratch. The author is insufficiently vain to want to create a new hardware licence from scratch<sup>33</sup>, and his favoured copyleft hardware licence, the CERN Open Hardware licence, is still subject to revision, so adopting an existing permissive software licence seemed to be a sensible course.

A review of existing permissive software licences suggested that the most well understood and widely adopted licence, which would need minimal revision (since it already had clauses dealing

---

<sup>33</sup> More plausibly, the author is too lazy.

with patents and trade marks, for example) was Apache 2.0.

Since the copyleft issues discussed above are rendered irrelevant, no additional drafting needed to be undertaken to attempt to deal with them. However, there were a few issues which can be useful to address in order to make the licence more appropriate to hardware, and the rationale for these is set out below. A diff of the current version of the licence is appended.

The licence is licensed under itself. Apache 2.0 is also licensed under itself, and since this licence is intended to be compatible with Apache 2.0, it is also capable of being self-licensed. The preamble refers specifically to Apache 2.0 and also allows the licensee to treat any work licensed under it to be licensed under Apache 2.0 in its pure form (the intention being, that it is possible to say that any work licensed under the Solderpad Hardware License is necessarily licensed in accordance with FSF and OSI criteria, as Apache 2.0 has been certified as fulfilling these criteria)<sup>34</sup>. The preamble is also intended to ensure compliance with the Apache 2.0 redistribution criteria in section 4.

Apache 2.0 explicitly deals with patents, trademarks and copyright. The main change in the Solderpad licence has been to extend the rights licensed by incorporating a new definition of “Rights”, used typically where reference to copyright alone was used, which is intended to sweep up, alongside copyright, all other relevant rights, such as design rights, semiconductor topography (mask) rights and database rights. A slightly controversial addition to clause 2 (Grant of License) provides that the licence also permits doing “...anything in relation to the Work as if the Rights did not exist” as an additional permission (still subject to the other conditions). The idea is that if the scope of intellectual property is increased from jurisdiction, then this will be picked up in the definition of “Rights”, but also related rights will be dealt with in this sweeping up clause.

The other changes are largely for clarity and are not intended to have legal effect. Thus, references to “Source” form now include net lists, board layouts and CAD files. “Object” form now includes intermediate forms such as bytecodes, FPGA bitstreams, artwork and semiconductor topographies.

“Derivative Works”, for clarity, do not include any work which physically connects or interoperates with the interfaces of the Work. “Contribution” has been extended to include designs as well as works of authorship.

Similar changes have been made to the Contributor License Agreement.

## A Note About the Name

Andrew Back kindly offered to host the modified Apache licence on Solderpad<sup>35</sup>, “a place to share, discover and collaborate on electronic projects”, and consequently, the name “Solderpad Hardware Licence” was adopted. This was partially to avoid suggesting a premature association with, or approval of the Apache Foundation for the licence, by using a name like “Apache Hardware Licence”, and partially to acknowledge Andrew's kind offer to host (as well as useful commentary

<sup>34</sup> The author is comfortable that Solderpad Hardware Licence itself is too, but it seems premature to make this assertion if the licence has not been officially approved.

<sup>35</sup> Solderpad.com

he gave during the drafting process). However, it is hoped that this is only an interim name. One possibility is that the Apache Foundation itself may consider adoption. The author has been in discussion with several board members about this licence, and it seems to be favourably viewed, but, understandably, there is no desire to formally adopt the licence in a vacuum without it being attached to a specific project under the aegis of Apache, so please contact Apache if you have a possible hardware project in mind! There has been much (too much) discussion about the name of the licence (whether it should have “Hardware” in the title, whether it should be associated with a specific hardware design repository, whether it might put off people wishing to use other repositories), but the name is not fixed.

## Conclusion

Open hardware presents challenges which do not map easily on to the challenges of free and open source software. Copyleft is particularly problematic, given that the cost of circumvention for hardware is lower than for software, that no obvious legal mechanism exists to make copyleft consistently applicable, and that the number of opportunities in the development and exploitation lifecycle for hardware for copyleft to impinge are much lower. For this reason, the author proposes that a licence based on the Apache 2.0 licence, which avoids the issues of copyleft, may be more appropriate for open hardware.

## Text of the modified Apache License<sup>36</sup>

This license is based closely on the Apache License Version 2.0, but is not approved or endorsed by the Apache Foundation. A copy of the non-modified Apache License 2.0 can be found at <http://www.apache.org/licenses/LICENSE-2.0>.

As this license is not currently OSI or FSF approved, the Licensor permits any Work licensed under this License, at the option of the Licensee, to be treated as licensed under the Apache License Version 2.0 (which is so approved).

This License is licensed under the terms of this License and in particular clause 7 below (Disclaimer of Warranties) applies in relation to its use.

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the Rights owner or entity authorized by the Rights owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Rights" means copyright and any similar right including design right (whether registered or unregistered), semiconductor topography (mask) rights and database rights (but excluding Patents and Trademarks).

"Source" form shall mean the preferred form for making modifications, including but not limited to source code, net lists, board layouts, CAD files, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, the instantiation of a hardware design and conversions to other media types, including intermediate forms such as bytecodes, FPGA bitstreams, artwork and semiconductor topographies (mask works).

---

<sup>36</sup> A diff file comparing this licence with the Apache 2.0 is available at <http://www.ifosslr.org/public/69-413-1-SP-1.pdf>



"Work" shall mean the work of authorship, whether in Source form or other Object form, made available under the License, as indicated by a Rights notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) or physically connect to or interoperate with the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any design or work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the Rights owner or by an individual or Legal Entity authorized to submit on behalf of the Rights owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the Rights owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

**2. Grant of License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable license under the Rights to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form and do anything in relation to the Work as if the Rights did not exist.

**3. Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

**4. Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this

License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

**5. Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

**6. Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

**7. Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

**8. Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages,

including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

**9. Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## APPENDIX: How to apply this license to your work

To apply this license to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

**Copyright** [yyyy] [name of copyright owner] Copyright and related rights are licensed under the [ ] Hardware License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at [ ]. Unless required by applicable law or agreed to in writing, software, hardware and materials distributed under this License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## Individual Contributor License Agreement ("Agreement") V2.0

Thank you for your interest in The [ ] Foundation (the "Foundation"). In order to clarify the intellectual property license granted with Contributions from any person or entity, the Foundation must have a Contributor License Agreement ("CLA") on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of the Foundation and its users; it does not change your rights to use your own Contributions for any other purpose. If you have not already done so, please complete and sign, then scan and email a pdf file of this Agreement to [secretary@\[ \].org](mailto:secretary@[ ].org). Alternatively, you may send it by facsimile to the Foundation at [ ]. If necessary, send an original signed Agreement to The [ ] Software Foundation, [ ]

Please read this document carefully before signing and keep a copy for your records.

Full name: \_\_\_\_\_

Mailing Address: \_\_\_\_\_

Country: \_\_\_\_\_

Telephone: \_\_\_\_\_

Facsimile: \_\_\_\_\_

E-Mail: \_\_\_\_\_

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to the Foundation. In return, the Foundation shall not use Your Contributions in a way that is contrary to the public benefit or inconsistent with its nonprofit status and bylaws in effect at the time of the Contribution. Except for the license granted herein to the Foundation and recipients of software distributed by the Foundation, You reserve all right, title, and interest in and to Your Contributions.

## 1. Definitions.

"You" (or "Your") shall mean the Rights owner or legal entity authorized by the Rights owner that is making this Agreement with the Foundation. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean any design or original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to the Foundation for inclusion in, or documentation of, any of the products owned or managed by the Foundation (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Foundation or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Foundation for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

"Rights" means copyright and any similar right including design right (whether registered or unregistered), semiconductor topography (mask) rights and database rights (but excluding Patents and Trademarks).

**2. Grant of License.** Subject to the terms and conditions of this Agreement, You hereby grant to the Foundation and to recipients of works distributed by the Foundation a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable license under the Rights to reproduce, prepare derivative works of (including instantiating a hardware design), publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works and do anything in

relation to the Work as if the Rights did not exist.

### 3. Grant of Patent License.

Subject to the terms and conditions of this Agreement, You hereby grant to the Foundation and to recipients of works distributed by the Foundation a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) was submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

4. You represent that you are legally entitled to grant the above license. If your employer(s) has rights to intellectual property that you create that includes your Contributions, you represent that you have received permission to make Contributions on behalf of that employer, that your employer has waived such rights for your Contributions to the Foundation, or that your employer has executed a separate Corporate CLA with the Foundation.

5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others). You represent that Your Contribution submissions include complete details of any third-party license or other restriction (including, but not limited to, related patents and trademarks) of which you are personally aware and which are associated with any part of Your Contributions.

6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

7. Should You wish to submit work that is not Your original creation, You may submit it to the Foundation separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".

8. You agree to notify the Foundation of any facts or circumstances of which you become aware that would make these representations inaccurate in any respect.

Please sign: \_\_\_\_\_ Date: \_\_\_\_\_

**About the author:**

**Andrew Katz** is a partner at boutique law firm Moorcrofts LLP, based in England's Thames Valley. Andrew specialises in technology law and has a particular interest in open design and development. He can be contacted at [andrew.katz@moorcrofts.com](mailto:andrew.katz@moorcrofts.com)

**Licence and Attribution**

This paper was published in the International Free and Open Source Software Law Review, Volume 4, Issue 1 (March 2012). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Katz, Andrew (2012) 'Towards a Functional Licence for Open Hardware', *IFOSS L. Rev.*, 4(1), pp 41 - 62  
DOI: [10.5033/ifosslr.v4i1.69](https://doi.org/10.5033/ifosslr.v4i1.69)

Copyright © 2012. Andrew Katz.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND available at <http://creativecommons.org/licenses/by-nd/2.0/uk/>

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.

