Open Hardware

Andrew Katz
OpenUK Future Leaders’ Group
17 July 2020
What is Open [Source] Hardware
Electronics (circuit boards)
Silicon Chips
Cases, racks, furniture
Mechanical devices - motors, engines, cars, boats
3D artistic works
3D printed works
Fluids
OSHWA Guidelines
Based on 4 freedoms and 10 OSI criteria
“Open source hardware is hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design.”
Question: can a chocolate cake be open hardware? Is the recipe a design document?
Software: software is automatically protected by copyright, therefore to open source software you need a licence. Is the same true of hardware?
Other examples:
Fashion (UK vs. Italy)
Database rights (EU vs US)
The IP Challenges:
Copyright, patent, design rights (registered and unregistered), plant breeders’ rights, semiconductor topography (mask) rights, database rights…
With software: many chances for copyright to apply: downloading, installing, running, distributing....
With hardware: IP will not impinge so often. You don’t need a licence every time use a hammer.
What licences are there?
FOSS licences: BSD, MIT, Apache, GPL
Content licences: Creative commons
HW Specific Licences: CERN OHL, Solderpad, TAPR, OCP Permissive and Copyleft
Issues with openness...
Software is made of 1s and 0s.
Hardware is made of atoms.
With software, your instructions will get you to 1 and 0 level.
With hardware, does the design need to get you to atoms?
...components: do you need to provide all the instructions?
And if you don’t have all those instructions, is the design truly open?
Other openness issues
Field of use (e.g. noncommercial)

IP Restrictions: e.g. patents

Requirement to provide complete source to components
In software: you can (largely) claim that any software released (as source) under and OSI/FSF approved licence is Open Source or Free Software
Hardware: more complex. OSHWA definition goes further. You also need:
1. The design must be public
2. Interfaces to software must be documented, or provided under an OSI licence
3. Certification requires you to use open components where possible
Hardware: it’s not so easy to apply a definition of openness.
EC Study: taxonomy for placing projects on an openness spectrum
Can copyleft/reciprocity work in open hardware?
Less opportunity for IPR to impinge
The boundary problem: horizontal (assemblies),
vertical (components).
Economic issues: may be easier to reverse engineer.

See https://www.jolts.world/index.php/jolts/article/view/69
Development issues:
Software: all in digital domain, tools (e.g. GCC) generally free. Can develop complex software on a cheap computer.
Development issues:

Hardware:
- Expensive design/development software
- Complex and expensive physical tools (lathes, milling machines, 3D printers...)

Shipping

Space

Environmental constraints

Cost and quality of feedstock
Many hardware projects are much more like software: FPGAs for example.
Does “hardwareness” matter? Introducing a scale to measure hardwareness across design, build, test, productise.
Economic impact:
Cost to replicate the Linux Kernel: £0
Cost to replicate a car: £xx,000
Economic impact: Cost to replicate an FPGA: $1.
Summary:
OSH much less mature than OSS
There are similarities, but big differences.
Basic definitions - e.g. openness, not so clear.
Licensing situation more complex (many more IPRs). But many fewer licences!
Dangerous to assume what works in OSS works in OSH
The real world has much more of an impact with OSH
There’s a lot of study to be undertaken...