Towards Social Hardware Development

By Andrew Back, Co-founder, SolderPad

It is increasingly rare to find end-to-end design and manufacture taking place under one roof. This "vertical disintegration" has led to a reconfiguration of the electronics industry value chain, creating a growing need for collaboration across many specialist firms and coordination across countries. As the intense pressure to specialise continues to drive outsourcing, now more than ever before, the development of hardware needs to become a social act.

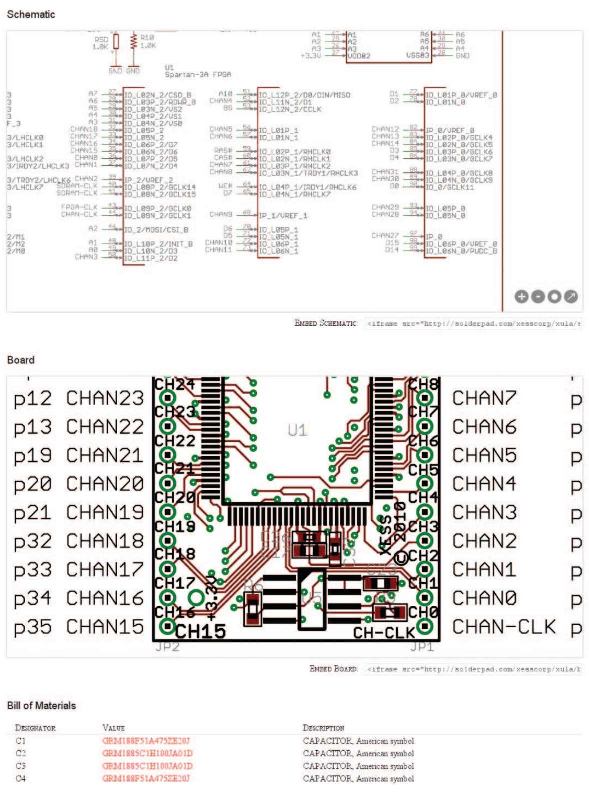
Open Source

In the past it was commonplace for design and manufacture to take place within the confines of a single organisation. This situation has changed over time and has been driven by technological progress and intensified competition, coupled with the emergence of rapidly growing new markets and reductions in trade barriers. This results in the proliferation of firms that specialise in specific areas of design, manufacture and assembly, and shifting locations of supply and demand. These significant changes to the value chain have brought with them the challenge of collaboration across organisational boundaries. Where there may be multiple time zones, first languages, engineering design tools and workflows in use, it is generally not practical or cost effective to co-locate entire project teams or mandate the use of common tools.

A Promising Pattern

The development of open source software provides one of the most visible and compelling examples of distributed cross-organisation collaboration. Shared practices based on simple tools and lightweight process have enabled low cost interaction and the highly social production of technology. Projects have ranged in size from the trivial and with a single contributor, to those that are incredibly complex and that have thousands. There is no single programming language, development environment or governance framework employed across open source software projects, and those in use may vary according to technical requirements, personal tastes or the principles of a particular technical community. However, they generally share the same set of basic working practices. Projects have a small proportion of "committers" who act as a quality gate and decide which contributions will be accepted, and may also delegate work and assume responsibility for things such as overall architecture. In open source software development, the source code is made available to all, and this may be modified by a developer providing a bug fix, or extended in order to add a new feature. Where new files have been created, a developer may simply send these to a committer, and where an existing file has been modified this can be compared against the original to provide a file that contains only the differences, and this "diff" or "patch" file sent to the committer. Since the patch contains only the changes, it is much smaller and the committer can apply it to a copy of the original file to recreate the new updated version. Version Control Systems (VCS) are central to open source software projects and enable them to track who has made revisions to files, revert these where necessary, and "tag" a release of all the files at a particular point in development, e.g. with a version number. It is these VCS repositories to which project committers have write access and are thus able to commit updates. The fundamental operation of open source software projects is incredibly simple, and the secret to their success lay in the fact that they eschew needless complexity in collaboration tools and practices. Instead choosing to work with solutions that get the job done with the minimum of fuss and least likelihood

A SolderPad workshop at the DesignSpark-sponsored Open Source Hardware Camp


Schematic

Embed Schematic: `<iframe src="http://solderpad.com/xesscorp/xula/s`

Board

Embed Board: `<iframe src="http://solderpad.com/xesscorp/xula/b`

Bill of Materials

Designator	Value	Description
C1	GRM188F51A475ZE20J	CAPACITOR, American symbol
C2	GRM1885C1H100JA01D	CAPACITOR, American symbol
C3	GRM1885C1H100JA01D	CAPACITOR, American symbol
C4	GRM188F51A475ZE20J	CAPACITOR, American symbol

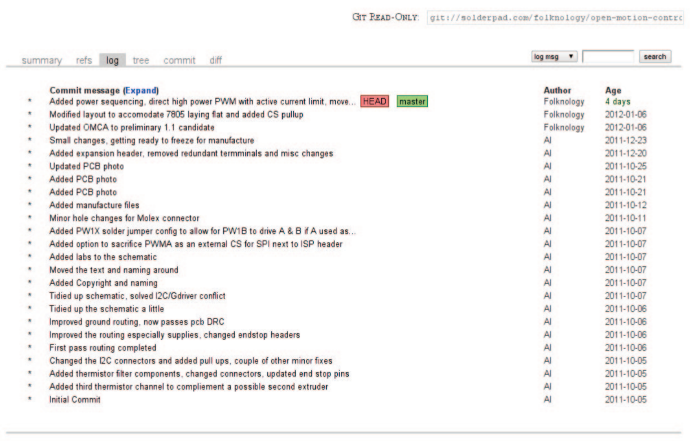
Detail of a SolderPad Project Showing Pan/Zoom Interface

of contention in a multi-contributor environment. The many benefits that this simplicity affords is something that enterprises are waking up to, as many now work to replace complex and expensive proprietary collaboration tools with the very same ones that are being used by open source projects. Adopting similarly agile processes, the principle difference is that software development is being done in private rather than in public. Whilst there are clearly differences between the development practices of software and those used in the development of hardware such as electronics, there is much valuable learning that can be taken from the success of open source software projects; from the remarkably lightweight processes used to support collaboration, to the pragmatic and vanity free technology that is used at the interface between collaborators.

New Challenges

Where collaboration is centred upon text such as specifications, bills of materials, and HDL and software source code, only the most simple tools are required in order to make sense of, comment upon and edit files. However, this represents only a small part of the hardware design story and many inputs and outputs are pictorial in form and require the use of special tools that are able to work with specific proprietary file formats. These could be, for example, mechanical drawings and 3D designs, and schematic diagrams and printed circuit board layouts. The tools required can be extremely expensive, and the situation is made worse by the fact that so many different file formats exist and a given tool will generally work with a subset of these at best. The Electronic Data Interchange Format (EDIF) is a standards-based attempt at producing a common vendor-neutral format for storing netlists and schematic diagrams. However, this has enjoyed very limited success for various reasons that include loose specifications which have resulted in many incompatible "flavours" of EDIF existing across tools, and market forces that have impeded uptake of the standard. Freely available tools such as DesignSpark PCB [1] offer help in the way that teams are able to install these at zero cost, and are thus able to avoid having to make a financial investment in tools that may only find use on one or a small number of projects. Further challenges to open source-style collaboration exist in the design tool file formats, as these are typically binary and cannot be easily compared to produce a file containing only the differences. Even where files are text-based they tend to be rewritten entirely each time they are saved. A minor design change may result in a file with completely reordered contents, and a simple comparison with the previous revision could suggest significant change when this is not the case. It is likely that to achieve the levels of agility, efficiency and scalability found in open source software projects, the fundamental tools and processes employed in support of collaborative hardware development will have to be equally pragmatic and lightweight.

Continued on page 34 >



Git Read-Only: git://solderpad.com/folknology/open-motion-contrc

summary refs log tree commit diff

	Commit message (Expand)	Author	Age
*	Added power sequencing, direct high power PWM with active current limit, move... HEAD master	Folknology	4 days
*	Modified layout to accomodate 7805 laying flat and added CS pullup	Folknology	2012-01-06
*	Updated OMCA to preliminary 1.1 candidate	Folknology	2012-01-06
*	Small changes, getting ready to freeze for manufacture	Al	2011-12-23
*	Added expansion header, removed redundant termminals and misc changes	Al	2011-12-20
*	Updated PCB photo	Al	2011-10-25
*	Added PCB photo	Al	2011-10-21
*	Added PCB photo	Al	2011-10-21
*	Added manufacture files	Al	2011-10-12
*	Minor hole changes for Molex connector	Al	2011-10-11
*	Added PW1X solder jumper config to allow for PW1B to drive A & B if A used as...	Al	2011-10-07
*	Added option to sacrifice PWMA as an external CS for SPI next to ISP header	Al	2011-10-07
*	Added labs to the schematic	Al	2011-10-07
*	Moved the text and naming around	Al	2011-10-07
*	Added Copyright and naming	Al	2011-10-07
*	Tidied up schematic, solved I2C/Gdriver conflict	Al	2011-10-07
*	Tidied up the schematic a little	Al	2011-10-06
*	Improved ground routing, now passes pcb DRC	Al	2011-10-06
*	Improved the routing especially supplies, changed endstop headers	Al	2011-10-06
*	First pass routing completed	Al	2011-10-06
*	Changed the I2C connectors and added pull ups, couple of other minor fixes	Al	2011-10-05
*	Added thermistor filter components, changed connectors, updated end stop pins	Al	2011-10-05
*	Added third thermistor channel to compliement a possible second extruder	Al	2011-10-05
*	Initial Commit	Al	2011-10-05

Detail of a SolderPad Project Showing Commit Log

< Continued from page 33

Areas of commonality will exist and some of the tools and processes used in open source software development may be repurposed, but it is possible that new technology and process will also be required.

SolderPad

SolderPad [2] is an online service that attempts to bring some of the benefits of open source software development to electronic engineering. It provides a place to share, discover and collaborate on electronic design, and the current release supports public hosting and is aimed squarely at open source hardware projects. The Git VCS technology is at the core of the platform and is used to manage both project files and meta-data, with support planned for the ability to also manage repository contents via the Web API. The initial release of the site works with simple design exports of images and JSON formatted bills of materials, and thus sidesteps getting into the complexity of structurally understanding the various native design tool file formats.

Designs are presented as a datasheet with:

• schematic diagrams and PCB layouts that are navigable via a pan/zoom interface;
• a bill of materials that allows search and pivot by part;
• explicit licensing and copyright information;
• a text description;
• tagging.

Conclusion

The electronics industry is presented with a significant opportunity to increase agility, productivity and the ability to scale in collaborative projects, through the adoption of lightweight process and pragmatic technology at the interface between partners. It may be that clues as to how this can be achieved in practice are to be found in the operation of open source software projects and, more recently, the emergent open source hardware movement. SolderPad is an example of a service that embraces open source techniques in support of achieving such goals, but this is only the beginning of the journey and there will be others. o

[1] http://www.designspark.com/pcb
[2] http://solderpad.com

The Git Distributed Version Control System

Git is The Version Control System that is used to support the development of Linux and that shares the same creator, Linus Torvalds. It puts an emphasis on performance and safeguarding against corruption, and this is not surprising when you consider that the Linux kernel comprises over 11 million lines of code spread across tens of thousands of files, and with thousands of contributors submitting updates. The Git VCS is one of an increasingly popular new breed that are distributed in nature. Which is to say that every developer has a copy of the entire repository and the development history, and it is this local copy or "clone" to which they commit their changes. Should they have write access to a project's master repository hosted on a server somewhere—i.e. be a project committer—they are then able to periodically "push" to this any changes that were committed locally. In the event that they do not have this level of access they may submit their changes to someone who does. Git provides enhanced support for many powerful development practices, such as providing a very low cost means of creating a "branch" where parallel development may safely proceed in a non-linear fashion, e.g. to test something out. With support for later merging this branch back into the development "trunk" if deemed appropriate, or else simply deleting it, many branches may exist at any one point in time and switching between them is trivial. Other less visible but equally important features include the cryptographic authentication of history, whereby it is made impossible to change old revisions of files and accountability is assured. One of the few downsides of Git is that the learning curve can be a little steep if you do not have much experience of such systems. However, crossing this hurdle is an investment well worth making as it will more often than not pay significant dividends in terms of productivity gains. What started out as a tool to support development of the Linux kernel is now incredibly popular, and in April 2011 the project hosting provider GitHub reported that they alone were hosting over 1,000,000 projects and that this figure was growing by 4,500 each day. This does not include other service providers, and individuals, projects and companies that host their own Git repositories.

To find out more about Git see: http://git-scm.com/

FIND IT:

**For more information visit:
www.rs-components.com/electronics**

Share your views...

www.designspark.com/eTech