Making Hardware with KiCad and Friends

Werner Almesberger
werner@almesberger.net

July 27, 2012

Overview

- KiCad introduction
- Collaborative design in Qi-Hardware
- Tools to improve KiCad
- Workflow overview

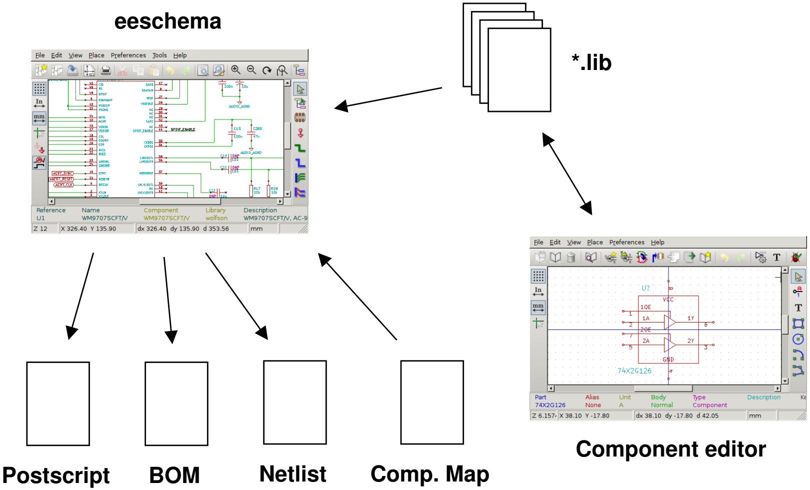
`downloads.qi-hardware.com/people/werner/fisl13.pdf`

KiCad

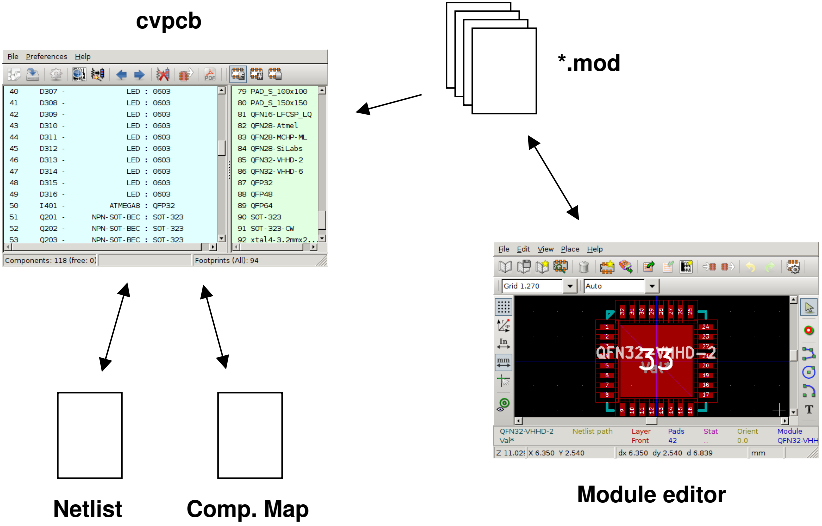
- Complete EDA solution
- Free Software: GPL, LGPL
- Development team by Jean-Pierre Charras, Dick Hollenbeck, and many others
- C++, wxWidgets
- Multi-platform: Linux, Windows, Mac
- Text files $\rightarrow$ extensible

`www.kicad-pcb.org`

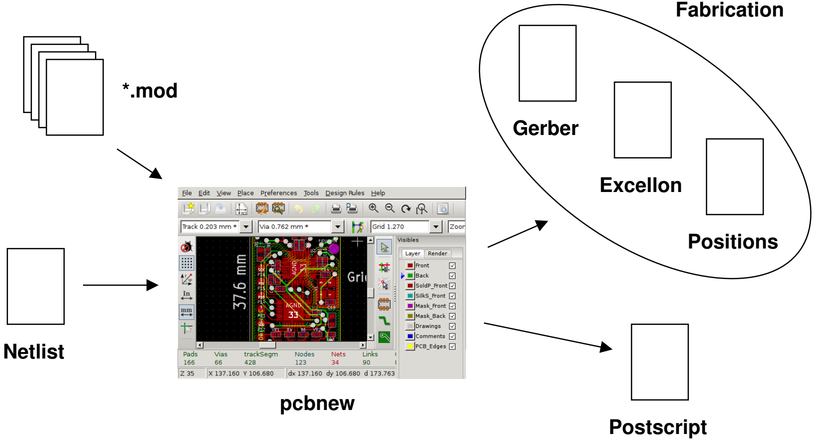
KiCad: Schematics



KiCad: Footprints



KiCad: Layout



Qi-Hardware

Structure and goals:

- Collection of loosely connected projects
- Informal gathering of like-minded developers
- Copyleft Hardware with Free Software
- Development and manufacturing
- Toolmaking

Products:

- Ben NanoNote (Handheld computer)
- Ben-WPAN (IEEE 802.15.4 wireless)
- Milkymist One (FPGA-based Video synthesizer)

`www.qi-hardware.com`

Collaborative Design

- Follow other people's work
- Review other people's work (projects and libraries)
- Shared design information
- Shared version control (git)
- Shared procedures and rules

KiCad Limitations

- No integrated version control system support
- Integration leads to weak peripheral tools
  - Footprint editor
  - Component catalog
  - Module catalog
- No scripting (coming)
- Scattered libraries

Qi-Hardware Adaptation

- Keep things simple: only Linux
- Command-line-oriented use
- Extend KiCad with patches
- External scripts/programs
- Own component and module libraries with common rules

Libraries

Goals:

- Consistent naming
- Known origin of design information
  E.g., IPC-7351
- Documented development process

To do:

- Improve organization
- Better integrate background information
- Reviews !

`tiny.cc/kicad-libs-components`

`tiny.cc/kicad-libs-modules`

What goes into git

KiCad:

- Project file: *project*.`pro`
- Schematics: *project*.`sch`, *subsheet*.`sch`
- Footprint mapping: *project*.`cmp`
- Layout: *project*.`brd`

Qi-Hardware:

- Makefile (`sch`, `brd`, `fab`, `clean`)
- Bookshelf (for dsv)
- Project-local libraries

What doesn't

- Netlist: *project*.net
  Generated with eeschema from *.sch and *project*.cmp
- Caches and backups: *sheet*.bak, *project*.000, *project*-cache.lib
- Postscript: *project-sheet*.ps, *project-layer*.ps
- Gerbers: *project-layer*.g??
- BOMs: *project*.lst, *project*.cvs
- Fab outputs: *project*.drl, *project*.pos
- And so on: *project*.cad, *project*.erc, *project*.dsn, *project*.rpt, *project*.wrl, ...

Project File (*project*.pro)

Contains:

- Project settings
- Lists of libraries

Issues:

- Local absolute paths
- Default libraries
- Timestamps → commit noise

Solution:

- Maintain paths manually
- `purge` script:
  `tiny.cc/wernermisc-bin-purge`

Makefile

Objectives:

- Convenience shortcuts
- Combine tools
- Share common procedures
- Avoid mistakes

```
front:
    pcbnew --plot=ps --plot-fill-all-zones \
       --layers=Front --plot-mirror \
       $(PROJECT).brd
    lpr $(PROJECT)-Front.ps
```

KiCad Command-Line Patches

Original hack by Werner Almesberger, clean rewrite by Wolfgang Spraul, soon rewrite for scripting.

- Idea: command-line access to main output functions
- eeschema
  - Visualization: Postscript (and SVG, DXF)
  - BOM $\rightarrow$ BOOM
  - Netlist, ERC
- pcbnew
  - Fabrication: Gerbers, drill (Excellon), component positions
  - Visualization and DIY: Postscript
  - Attributes: mirror, size, origin, fill zones, ...

`tiny.cc/eda-tools-kicad-patches`

Data Sheet Viewer (dsv)

- Bookshelf: name and aliases → PDF or ZIP with PDF
- Download (wget) to local cache
- Hierarchical lookup by name or alias
- Example:
  `dsv avr` → ATtiny87 (`doc8265.pdf`) → xpdf

Bookshelf example:

```
N: attiny87
A: attiny167
A: avr
D: http://www.atmel.com/Images/doc8265.pdf
```

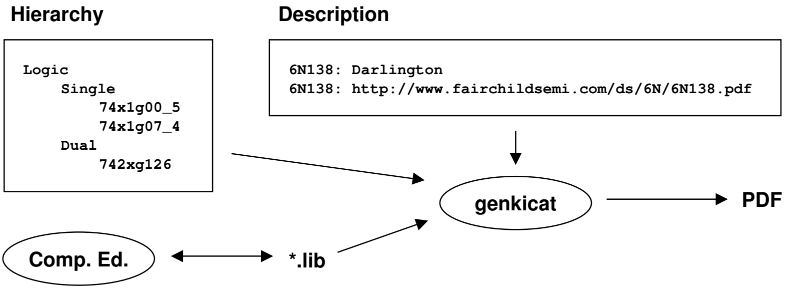
dsv Motivation

- Make sure everybody uses the same data sheet
    - Same component
    - Same manufacturer
    - Same data sheet revision
- Copyright: can't just check in PDF
- Industrial pragmatism: private mail or "internal" repository
- Qi-Hardware: share `BOOKSHELF`

`tiny.cc/eda-tools-dsv`

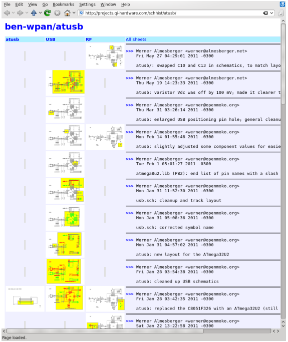
Schematics Symbol Catalog

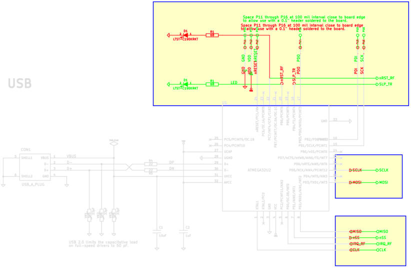
- For selection, review
- Hierarchical order with alphabetical index
- Short descriptions and data sheet links
  To do: connect to dsv
- Added value: indicate pin types



Schematics History

- Schematics revision history
- `git diff` produces gibberish





`projects.qi-hardware.com/schhist/`

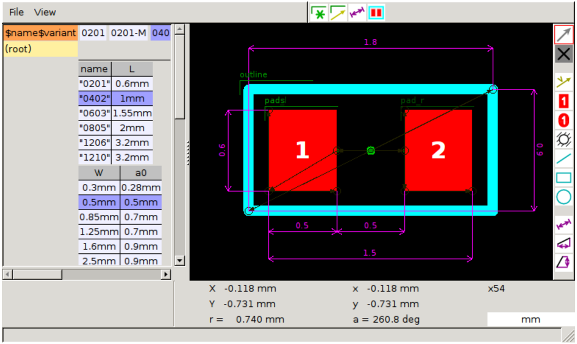
Schematics History (cont'd)

How it works:

- Walk git revision history
- Check out project whenever schematics change
- Run eeschema to make Postscript
- Convert to PNG (find diffs, with highlighting) and PDF
- Wrap in HTML

`tiny.cc/eda-tools-schhist`

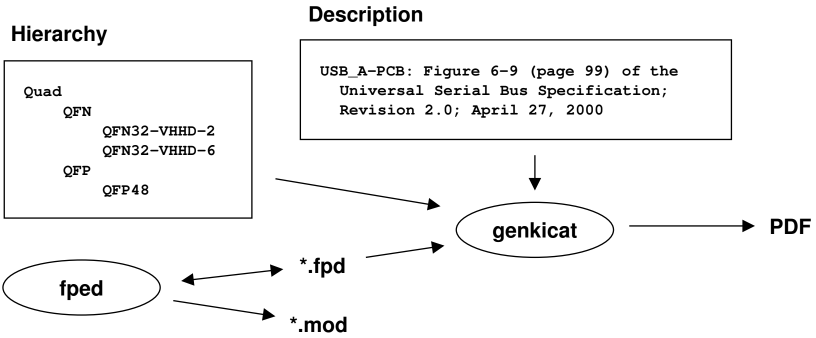
Footprint Editor (fped)



- Fully parametric
- Automated repetition (loops, tables, sub-frames)
- GUI or text-based
- Automatic measurements
- Output: KiCad, Postscript, Gnuplot

Footprint Catalog

- Like schematics symbol catalog, but for footprints
- Added value: measurements, pad types



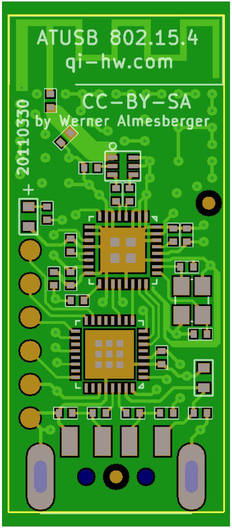
`tiny.cc/eda-tools-genkicat`

Layout History

- Layout revision history
- To do . . .

Gerber Renderer (prettygerbv)

- "Photorealistic" view of PCB
- Combines Gerber (traces, drawings) and Excellon (holes)
- Easy to see issues with solder paste, silk screen, solder mask, . . .
- Uses gerbv for rendering



`tiny.cc/eda-tools-fab-prettygerbv`
`tiny.cc/eda-tools-fab`

BOM Processor (BOOM)

- BOM $\rightarrow$ select components $\rightarrow$ find supplier $\rightarrow$ shopping list
  - Decode product numbers
  - Match characteristics (manufacturer part)
  - Find part in inventories (distributor part)
  - Select best price
- Work in progress. Rewrite from Perl to C.
- Future: Help with component selection

`tiny.cc/eda-tools-b2`

Schematics Design Rules

Work in progress. First compilation by Adam Wang.

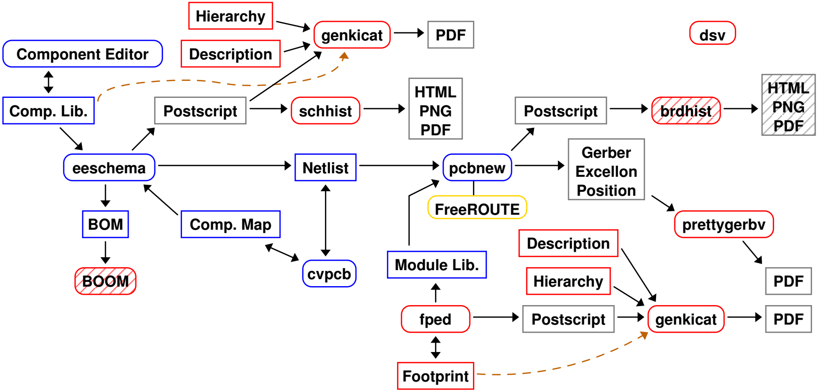
- Clean visual representation
- Compatibility with post-processors (e.g., BOOM)

Examples:

- Value naming (4k7, 10nH, ...)
- Junction style
- Naming of negated pins (e.g., nRESET)
- Text placement and size
- Checklists for reviews

`en.qi-hardware.com/wiki/Rules_on_Editing_Schematics`

Qi-Hardware Workflow



Conclusion

Experience this far:

- KiCad is a good basis for collaborative projects
- Easy to extand, thanks to open design
- Automation makes workflow less intimidating

To do:

- Spread the word
- Finish BOOM and write brdhist
- Unify meta-data (bookshelf, etc.)
- Improve quality of libraries
- Command-line support in mainline KiCad

`projects.qi-hardware.com/`

`downloads.qi-hardware.com/people/werner/fisl13.pdf`