

Recreate or Modify PCB Layouts with Open-Source Software

Follow this step-by-step process for a cost-efficient way to duplicate a physical or published printed circuit board.

Dr. Erich S. Heinzle, VK5HSE

A frequently encountered challenge is the duplication of an existing printed circuit board (PCB) layout — one that was published or sold commercially but is no longer available, or an existing design needing replacement or updating with new components. As shown in Figure 1, this is easily done with free, open-source software (see the sidebar, “Software Used,” for details on the required programs).

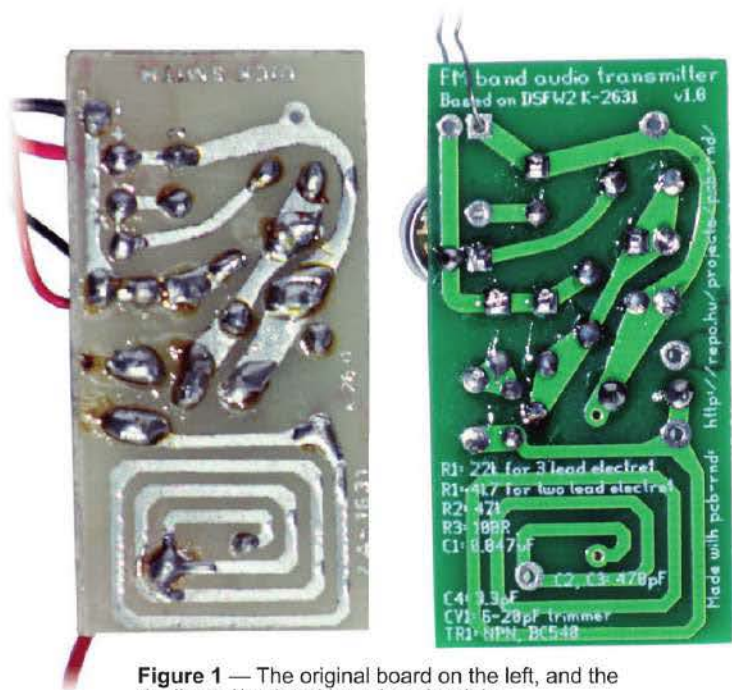


Figure 1 — The original board on the left, and the duplicated/updated board on the right.

To follow this process, you will also need a computer running an open-source operating system. As the requirements for the PCB layout software are minimal, even a Pentium 4 with 512 MB of RAM and a monitor with as little as 800 × 600 resolution will work. For lower-end hardware, lubuntu (www.lubuntu.net), a lightweight version of the Ubuntu GNU/Linux operating system, or the related Debian (www.debian.org) operating system with a lightweight window manager/desktop environment such as XFCE or LXDE, are ideal. These are easily installed from a boot CD or bootable USB drive containing a freely downloadable boot image from the internet. Alternatively, the FreeBSD operating systems and Apple's OS X operating systems can also be used.

Software Used

■ *Graphic Image Manipulation Program (GIMP)*: an open-source graphics package available for GNU/Linux, MacOS X, and Windows. If a PDF is the source of the layout, *GIMP* can import the PDF page at a specified resolution (300 dpi is adequate). Other graphic file formats can be opened with *GIMP*. It is available at www.gimp.org.

■ *pcb-rnd*: allows an image to be displayed under the PCB layout being designed. It is available as a standard package in GNU/Linux distributions such as Ubuntu and Debian, as well as being available as source code, which can be downloaded and compiled locally. *pcb-rnd* can save in other PCB layout formats such as *gEDA PCB*, *Protel Autotrax*, *KiCad*, and *FidoCadJ*.

This tool also enjoys parametric footprint support, padstacks, slotted pads, polygonal copper features, and arbitrary text in subcircuits (i.e. footprints). The layout editor supports .png, .eps, and postscript export for home PCB fabricators, and Gerber output format for commercial board manufacture.

Of note, *pcb-rnd* can be made to run on MacOS X, and a dedicated internet relay chat (IRC) channel exists if help is needed with installation and use. Users of older operating system versions may prefer to build and install the latest tools from source code. It is available at repo.hu/projects/pcb-rnd.

■ *gerbv*: a viewer for Gerber-formatted files, available at gerbv.geda-project.org. On Ubuntu- and Debian-based systems, *gerbv* can be installed from the package `sudo apt-get install gerbv`.

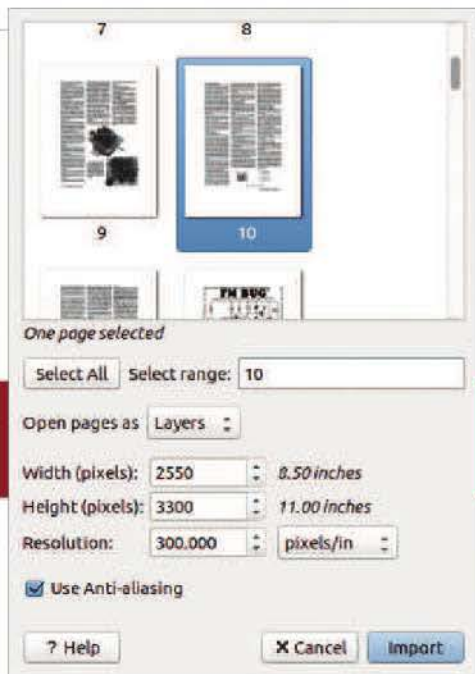


Figure 2 — FM bug layout by Colin Mitchell imported into GIMP from PDF.

The Process

1 Input an image of the PCB layout's copper layer to your computer. For physical printed circuit boards, a flatbed scanner or a smartphone camera will work. Save the image as a JPEG, PNG, BMP, or TIFF file. For layouts published in portable document format (PDF), import the PDF into a graphic design program, or take and save a screenshot.

2 Crop the image and correct any distortion with the *Graphic Image Manipulation Program (GIMP)* (see Figure 2), an open-source graphics package available for GNU/Linux, Mac OS X, and Windows. After loading the image, crop the PCB region within the image (see Figure 3). Rotation of the board image may be necessary for board scans (see Figure 4). Following cropping, correct any distortion with the box transform tool to make each corner a right angle. The overall dimensions and aspect ratio of the image are not important, as the PCB layout software will scale the image to match the specified layout size (see Figure 5).

3 Save the image as either a JPEG or PNG format.

4 The preferred PCB layout software is the open-source *pcb-rnd*. It is available as a standard package in GNU/Linux distributions like Ubuntu and Debian, and as source code that can be downloaded and compiled locally. In Ubuntu- or Debian-based distributions, the command for installing from a package is `sudo apt-get install pcb-rnd`.

Launch *pcb-rnd* with the PCB image displayed in the background of the editor. Assuming the PCB layout image is saved as `MYLAYOUT.PNG`, this can be done in a terminal window with `pcb-rnd -bg-image MYLAYOUT.PNG` (see Figure 6).

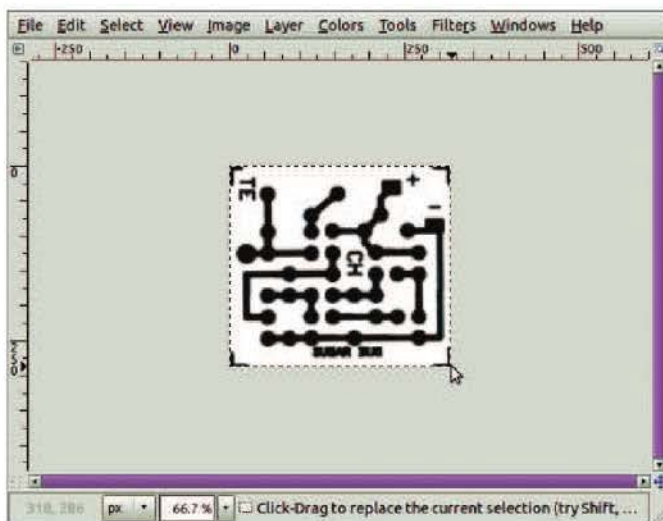


Figure 3 — Cropped layout image in GIMP.

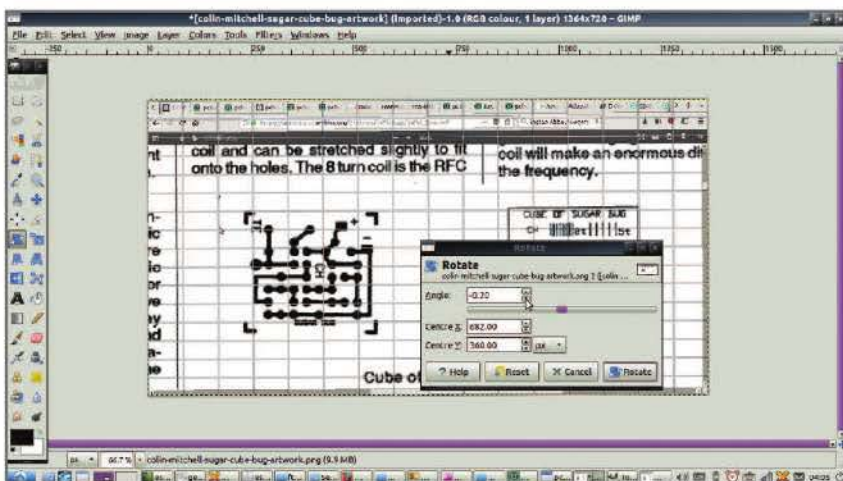


Figure 4 — Rotation of the image.

5 As the image will stretch to suit the layout size, it can be adjusted in the **PREFERENCES** dialogue (see Figure 7). If the PCB size is known, specify this as the layout size. If the PCB size is unknown, determine the required dimensions and scaling by identifying a known mounting pad type with readily determined dimensions on the board. An integrated circuit with 0.1-inch pin spacing would be ideal.

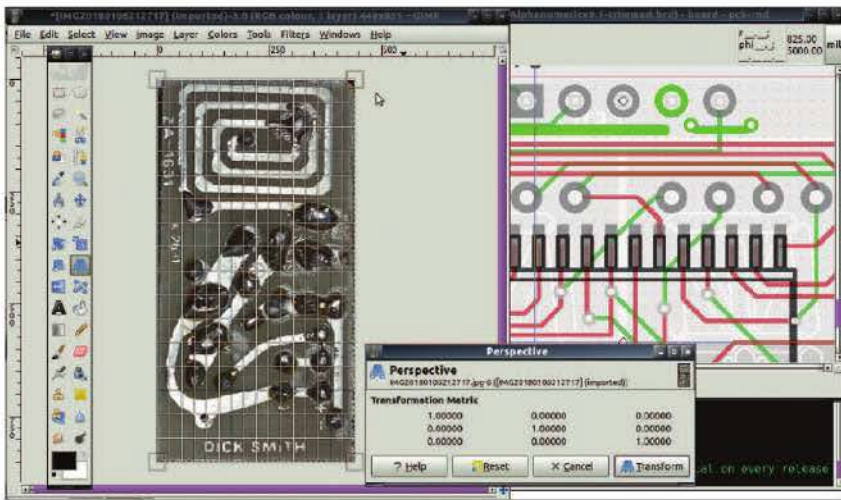


Figure 5 — Cage transformation of the PCB photograph.

6 In *pcb-rnd*, footprints — the mounting pads for components on PCBs — are called *subcircuits*. *pcb-rnd* supports multiple footprint formats and footprint library types, including *gEDA PCB*, Protel Autotrax, Eagle binary, Eagle XML, and *KiCad*, or it can use existing PCB layout files in those formats as footprint libraries. *pcb-rnd* also supports real-time internet-based footprint browsing and retrieval from sites such as gedasymbols.org. Invoke the footprint library window with the `w` and `l` key commands. Having identified a known footprint type, place a matching footprint into the layout editor over the feature on the background image.

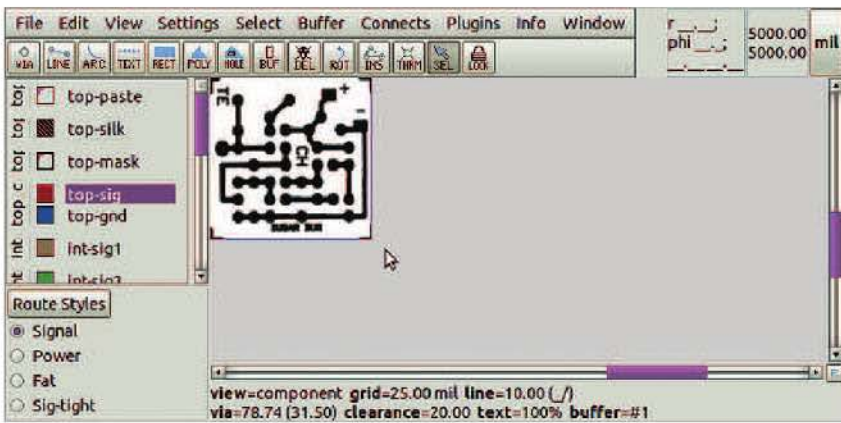


Figure 6 — The layout editor with the background image.

7 Carefully manipulate the board size in the layout preferences to match placed footprints with the scaled footprints in the background image. An initial guess at the board size and some simple calculations to determine the needed scaling of features can speed this process. Once the board size has been determined, draw a set of lines around the perimeter of the board on the outline layer to define the board outline when exporting files for manufacture (see Figure 8).

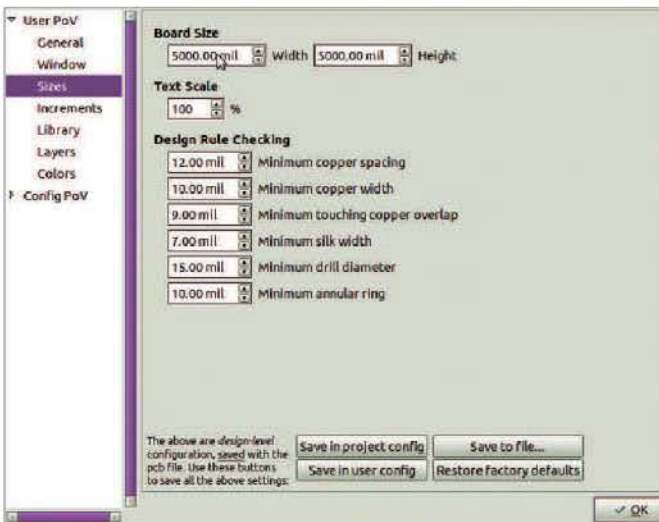


Figure 7 — PCB layout size preferences.

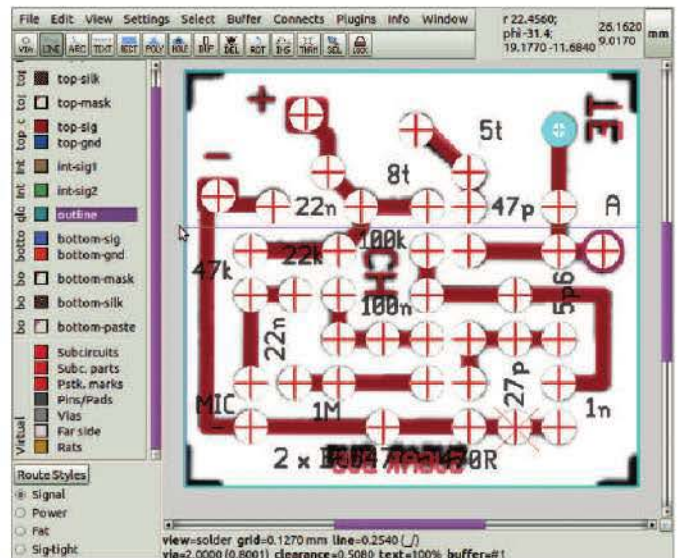


Figure 8 — Add the outline to the board.

8 Now place additional individual footprints on each of the background image footprint features. On the silkscreen of a PCB, the reference designator (or *refdes*) is the information that refers the component to the schematic. If the *refdes* is known for a component, it can be specified. Select the footprint and invoke the edit properties window with **e** and **p** keys to alter the *refdes*. The silkscreen *refdes* is an optional subcircuit attribute in *pcb-rnd* and can be deleted if not needed.

9 Components on the layout can now be modified if necessary. In Figure 9, the original variable capacitor footprint in the discontinued Dick Smith Electronics (DSE) FM bug has been modified to accommodate a more available two-pin type. Simple boards designed in the days of tape and transparencies may be duplicated with padstacks (pads that interconnect the top side traces to bottom side traces; also called untented vias) and tracks, an approach demonstrated with the “Sugar Cube Bug” design by Colin Mitchell in Figure 10.

Footprints can be duplicated in *pcb-rnd* by adding the silkscreen, copper, and hole features to match those of the original feature, and then selecting the set of features and converting them to a subcircuit (see Figure 11).

10 Once components have been placed, lay out the tracks. The route style dialogue allows the track width to be varied, even on the same electrical net. It also allows arbitrary placement, various snap-to-grid options, and polygonal copper features. Track widths can be measured to precisely specify the width, although a reasonable visual match is sufficient for most layouts not operating in the microwave region. Arcs on copper, silkscreen, and outline layers are supported on *pcb-rnd*, making it easy to duplicate designs with photoresist tape and sweeping curves. Arc properties can be inspected and modified with the “Edit Properties” key commands **e** and **p**.

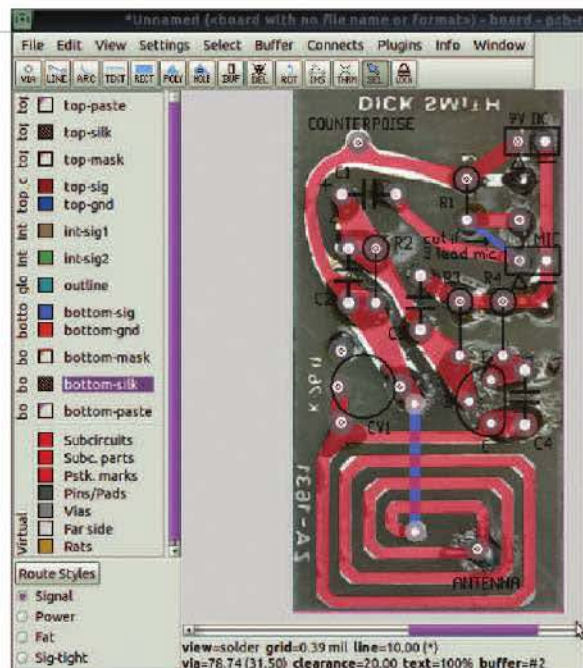


Figure 9 — The FM bug layout showing a two-pin trimmer as well as the original three-pin trimmer.

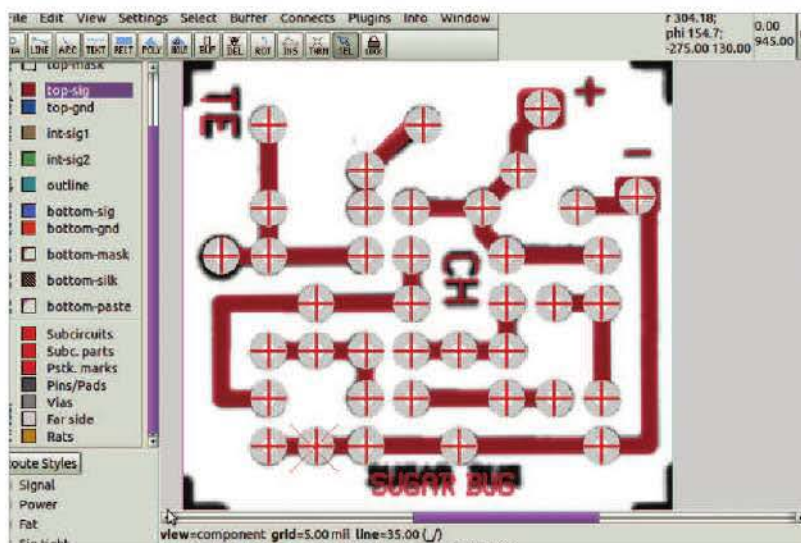


Figure 10 — A screenshot of the sugar cube layout.

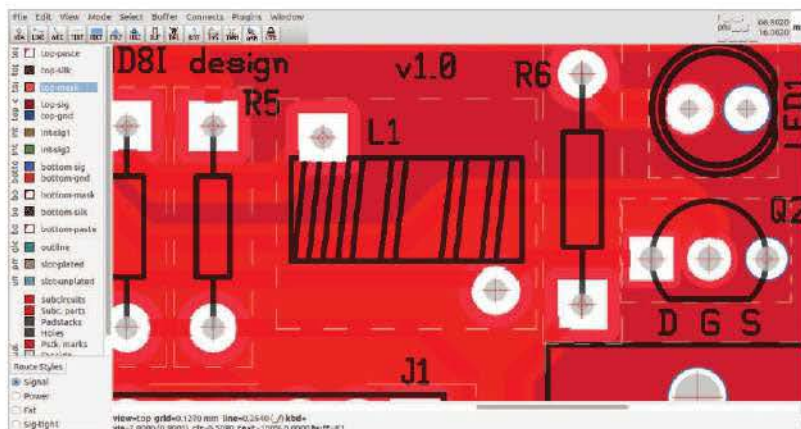


Figure 11 — The FT37-43 subcircuit/footprint.

11 *pcb-rnd* allows rectangular or polygonal copper pours to be placed on the layout. Some pins on components will need to connect to the copper pours functioning as ground or power planes. Hover the cursor over a pin and use the command `:padstackedit()` to check and adjust the clearance of that pin on the silkscreen, copper, or solder mask layers. Thermal “spokes” can be applied to pins connected to a ground plane. The thermal tool is used on the pin when the corresponding copper layer is selected to add thermals. *pcb-rnd* also supports thermals on solder mask-defined (SMD) pads (see Figure 12).

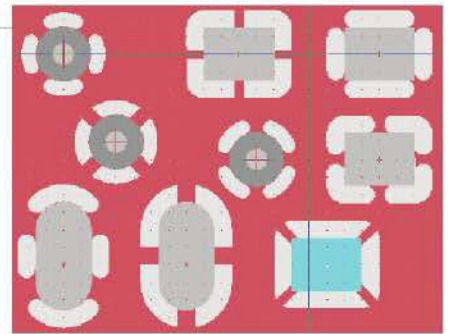


Figure 12 — Thermals applied to pins and pads in *pcb-rnd*

12 You can now place text onto the layout. The *pcb-rnd* editor allows arbitrary text rotation and supports multiple fonts (see Figure 13). Thus, you can enhance older designs with silkscreen component outlines, component labels, component values, version information, and instructions.

13 For double-layer PCBs, save the current layout as **MYLAYOUT.LHT** and close the layout editor. Now undertake steps 1, 2, and 3 for the other side of the PCB, but mirror the image left to right before saving as **MYLAYOUT2.PNG**. For step 4, do the following: `pcb-rnd --bg-image MYLAYOUT2.PNG MYLAYOUT.LHT`. Step 5 is not necessary as the layout size is already correct.

14 You can now view the reverse side of the board by hitting **ALT** and **TAB** to flip the board in the layout editor. Next, undertake steps 6 – 12 to add the necessary footprints, tracks, copper pours, and text on the reverse layer to match the background image.

15 After saving your work, export the layout in whichever supported format suits your use. Use PNG, postscript, or encapsulated postscript (EPS) to allow the copper layers to be printed on a transparency for photo-

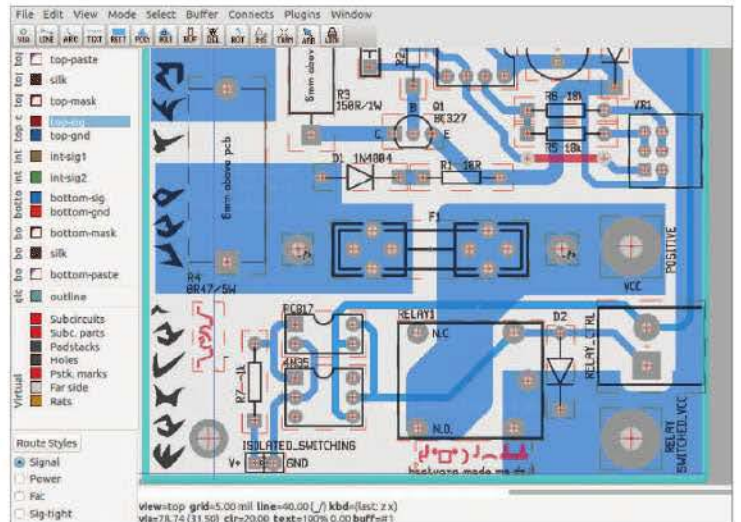


Figure 13 — Multiple font support in *pcb-rnd*.

resist exposure, or on suitable paper for toner transfer. Mirrored outputs are supported, and postscript export allows scaling to compensate for printer-introduced distortion. Use the industry standard Gerber format

Naming Your Files

pcb-rnd allows CAM (computer-aided manufacture) rules to be specified for particular manufacturers, but default industry standard naming conventions will be suitable for most manufacturers. Typical naming conventions employ file endings to identify the various layers. A common format is the following:

- MYLAYOUT.GTO - TOP LAYER SILKSCREEN**
- MYLAYOUT.GTS - TOP LAYER SOLDER MASK**
- MYLAYOUT.GTL - TOP LAYER COPPER FEATURES**
- MYLAYOUT.GTP - TOP LAYER SOLDER PASTE STENCIL**
- MYLAYOUT.GBO - BOTTOM LAYER SILKSCREEN**
- MYLAYOUT.GBS - BOTTOM LAYER SOLDERMASK**
- MYLAYOUT.GBL - BOTTOM LAYER COPPER**

MYLAYOUT.GBP - BOTTOM LAYER SOLDER PASTE STENCIL

MYLAYOUT.CNC - EXCELLON FORMT DRILL FILE, Sometimes MYLAYOUT.TXT

Silkscreen layers are used for text, labels, and artwork. Solder mask layers are the epoxy layer (often green) coating the tracks but leaving the pins and pads exposed for soldering. Solder mask layers are generated automatically by the software when Gerber files are generated. *pcb-rnd* allows features to be drawn on the solder mask as apertures both on the board and in subcircuits. Top and bottom paste layers are used to create stencils for bulk application of solder paste prior to SMD device placement and soldering. They are unnecessary for layouts without SMD components, or for SMD components where hand application of solder is planned.

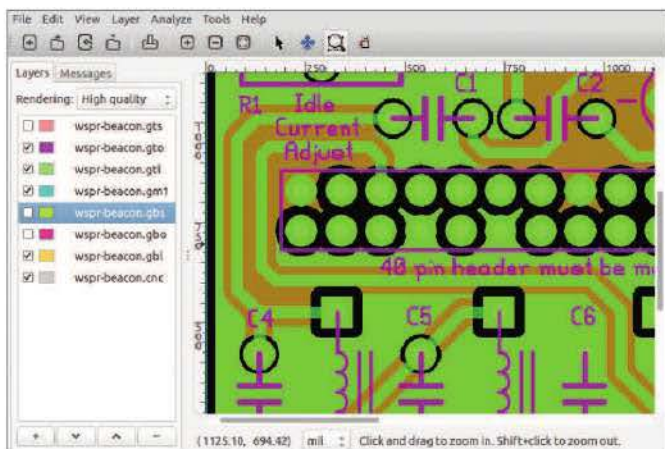


Figure 14 — Confirming connection to the ground plane and clearance distances on the header pins of a Raspberry Pi shield in the Gerber viewer *gerbv*.

RS-274X for conventional manufacturing (expected in zip files by manufacturers). See the sidebar, “Naming Your Files” for examples of how to export all your files.

16 After export, check your Gerber files for clearances between copper features prior to manufacturing. Figure 14 shows the ground connections on a Raspberry Pi WSPR shield header connected to ground planes being viewed in the Gerber file viewer *gerbv*.

Conclusion

You have now learned how to design your own PCB layouts using low-cost or free hardware and free software. You might consider exploring free, open-source schematic editors that can export netlists for use in the layout editor, allowing you to design circuits and then proceed to PCB layout with a corresponding netlist detailing the electronic components in a circuit and what nodes they are connected to. In addition, the tools described here can provide creative possibilities at little or no cost for science, technology, engineering, and mathematics (STEM) programs in schools.

All photos by the author.

Erich Heinzle, VK5HSE, is an occupational and environmental physician with a Bachelor's of Medicine and Surgery, a Master's of Public Health, and a graduate diploma in OH&S management. Erich has been a licensed amateur for 11 years, and is particularly interested in software-defined radio, digital modes, communication theory, and open-source software. In pursuit of these interests, Erich has recently completed a Bachelor's of Science in mathematics and is currently undertaking a Master's of Scientific Studies in mathematics at the University of New England, Australia. Erich is investigating the application of wavelet compression to low-bit rate audio codecs for his Master's degree research project. You can reach Erich at a1039181@gmail.com.

For updates to this article, see the **QST Feedback** page at www.arrl.org/feedback.



A range of Amateur Radio articles and technical notes are coming up in the July/August 2019 and future issues of QEX.

- John Westmoreland, AJ6BC, describes THEMIS, an experimental GPS-disciplined oscillator.
- Andrzej (Andy) Przedpelski, K0ABP, takes a different look at the phase-locked loop.
- Joseph Pingree, WB2TVB, shows how to design and print 3D components.
- Tuck Choy, M0TCC, considers pi networks with and without inductor loss in a two-part series.

- Scott Roleson, KC7CJ, constructs a receiver step attenuator.
 - John Stensby, N5DF, calculates coax loss directly from impedance measurements.
 - Jerry Whitney's, KG2BK, T-network calculator determines matching components.
 - Maynard A. Wright, W6PAP, uses *Octave* for complex characteristic impedance calculations.
 - Ulrich L. Rohde, N1UL, and Kai Siwiak, KE4PT, explain tuning short antennas for portable operations.
- QEX* is edited by Kazimierz “Kai” Siwiak, KE4PT, (kswiak@arrl.org) and is published bimonthly. *QEX* is a forum for the free exchange of ideas among communications experi-

menters. The annual subscription rate (six issues per year) in the United States is \$29. First-class delivery in the US is available at an annual rate of \$40. For international subscribers, including those in Canada and Mexico, *QEX* can be delivered by air-mail for \$35 annually. Subscribe today at www.arrl.org/qex.

Would you like to write for *QEX*? We pay \$50 per published page for full articles and *QEX* Technical Notes. Get more information and an Author Guide at www.arrl.org/qex-author-guide. If you prefer postal mail, send a business-size self-addressed, stamped (US postage) envelope to: *QEX* Author Guide, c/o Maty Weinberg, ARRL, 225 Main St., Newington, CT 06111.