

Kurz vorgestellt – Elektronik-CAD-System KiCAD

Die Komplexität auch von Schaltungen der Hobbyelektronik wird immer höher, manche Bauelemente sind ohne eine gedruckte Leiterplatte nicht mehr zu verarbeiten, und die Betriebssicherheit solcher Schaltungen auf Lochrasterplatten lässt oft zu wünschen übrig.

Wir stellen in einem kurzen Abriss ein Open-Source-Platinenentwurfsprogramm vor, das es auch dem Hobbyelektroniker ermöglicht, direkt aus dem Schaltungsentwurf heraus professionelle Platinen zu erstellen.

Komplettes System

Programme zur Erstellung von Platinenlayouts gibt es zahlreich, die meisten sind für professionelle Anwender bestimmt und damit auch sehr teuer. Eine sehr beliebte Ausnahme ist EAGLE [1] von CadSoft, mit einem Wermutstropfen, es unterliegt Beschränkungen in der Platinengröße. Sehr preiswert ist auch Sprint-Layout von Abacom [2], allerdings ist es ein reines Platinenentwurfsprogramm ohne Anbindung an einen Schaltplaneditor.

Da kommt ein Programm wie KiCAD [3] genau richtig. Es ist als Open-Source-Programm für MS Windows,

Linux und Mac OS X nicht nur kostenlos, sondern auch sehr leistungsfähig mit den zusammenhängenden Hauptkomponenten Schaltplaneditor und Platinendesigner. Dass hier sogar ein 3D-Betrachter und ein Gerber-Viewer zum Programmpaket gehören, ist eine große Leistung der Entwickler. Entsprechend hat KiCAD eine große Anhängercommunity, auch in Deutschland [4]. Was ist KiCAD nun? Das von Jean-Pierre Charras entworfene System ist eine komplette CAD-Suite zum Entwerfen und Layouten von Schaltungen. Es besteht aus vier Hauptkomponenten, dem Schaltplaneditor (EESchema), dem Tool CvPCB, dem Layouteditor (PCBnew) und dem Gerber-Viewer GerbView. Dazu kommen weitere Komponenten wie Bitmap2Component oder der bereits erwähnte 3D-Viewer. Nur der Vollständigkeit halber sei erwähnt, dass mit KiCAD bis zu 16-lagige Platinendesigns zu erzeugen sind, in aller Regel werden dem Hobbyelektroniker zwei Lagen reichen.

Der Schaltplaneditor EESchema

Nach dem Starten des Programms öffnet sich zunächst ein Projektmanager (Bild 1), wie man ihn in ähnlicher Form auch von anderen Programmen kennt. Hier sind alle Daten des aktuell bearbeiteten Projekts übersichtlich zusammengefasst, und die einzelnen Komponenten des Programms direkt per Button erreichbar. Wie bei solchen Programmen üblich, sind alle Daten eines Projekts in einem Archiv abgelegt.

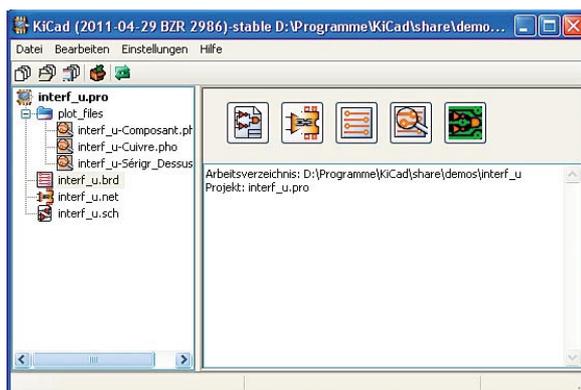


Bild 1: Über den Projektmanager wird das gesamte Projekt übersichtlich verwaltet.

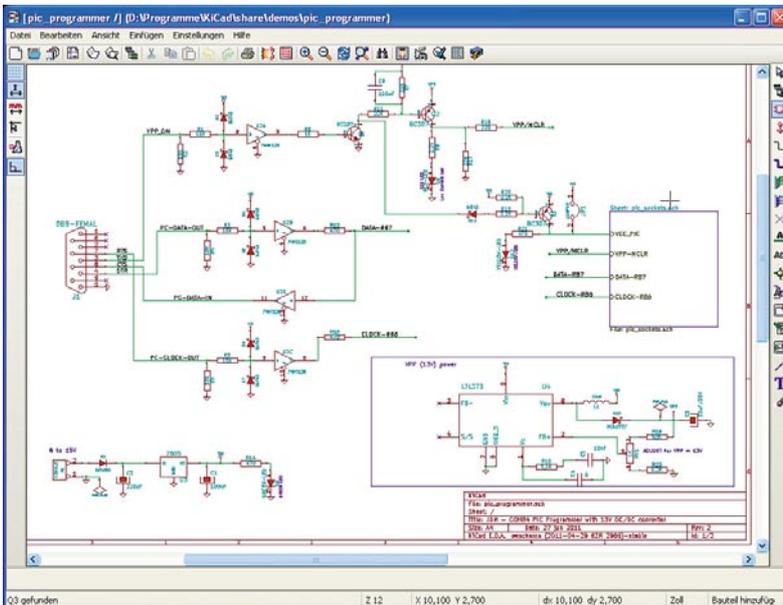


Bild 2: Der Schaltplaneditor von KiCAD – übersichtlich und komfortabel

Wählt man den Button des Schaltplaneditors (EESchema) an oder öffnet zum ersten Kennenlernen eines der mitgelieferten Beispiele (Dateiendung .sch), so öffnet sich der Schaltplaneditor (Bild 2). Hier findet man sich, auch wegen der deutschen Bedienoberfläche, schnell zurecht, ein ausführliches englisches Tutorial ist unter „Hilfe“ hinterlegt, deutsche Einstiegstutorials finden sich im Internet, z. B. unter [4]. Die mitgelieferten Beispiele eignen sich sehr gut für erste Übungen.

Bereits in der Grundausstattung ist eine riesige Bauteilbibliothek (Bild 3) mit umfangreichen Suchfunktionen vorhanden, weitere sogenannte Libs sind via Internet verfügbar, z. B. unter [5]. Hier arbeitet eine aktive Gemeinde an der Vervollkommnung und aktuellen Anpassung der Bibliotheken. Über den Bibliotheksbrowser ist auch jeweils ein Link zum Datenblatt des geladenen Bauelements verfügbar.

Natürlich kann man auch selbst Bauteile über einen Bauteileditor (Bild 4) entwerfen und in die Bibliothek einfügen.

Beim Entwerfen des Stromlaufplans sind relativ wenige Designregeln zu beachten, lediglich auf eine exakte Platzierung der elektrischen Verbindungen und das Hinzufügen von Netznamen ist genauestens zu achten. Offensichtliche Fehler findet der „electric rule check“ (Bild 5). Auch eine Stückliste ist automatisch erzeugbar. Über ein Annotationswerkzeug (Nummerierung der Bauteile) kann die Nummerierung automatisch nach verschiedenen definierbaren Regeln ausgeführt werden.

Am Schluss ist eine Netzliste zu erstellen, damit man mit Hilfe des Programmteils CvPCB den Schaltplansymbolen reale Bauteile (Footprints, Bild 6) zuordnen kann, die wiederum der Layouteditor als Arbeitsgrundlage benötigt. Die Netzliste ist neben dem Format für den eigenen Layouteditor (PCBnew) auch für andere verbreitete Layoutprogramme erzeugbar.

Der Layouteditor PCBnew und das 3D-Tool

Ist die Schaltung abgespeichert und die Netzliste erzeugt, kann es an das Layouten gehen.

Dazu wird der Layouteditor geöffnet und nach wenigen Grundeinstellungen wie Platinengröße, Lagen usw. die eben erstellte Netzliste geladen. Das Routen übernimmt nach dem Platzieren der Autorouter, wahlweise lässt sich auch ein Web-basierter externer Router einbinden. Natürlich ist jederzeit ein manuelles Eingreifen möglich und auch nötig, alles schaffen selbst teuerste Autorouter nicht. Hier sollte man sich bei den ersten Versuchen nicht entmutigen lassen, sondern sich Schritt für Schritt anhand der Online-Hilfe in das interaktive Platzieren und Routen einarbeiten.

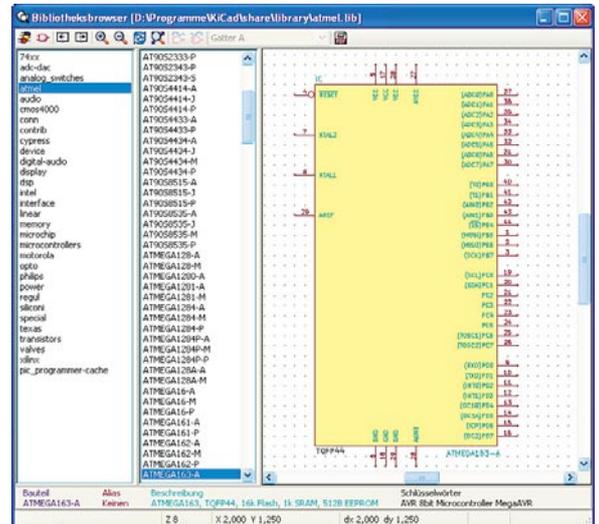


Bild 3: Die Bauteilbibliothek ist riesig, mit eigenen Bauteilen sowie Bibliotheken aus der KiCAD-Community ergänzt.

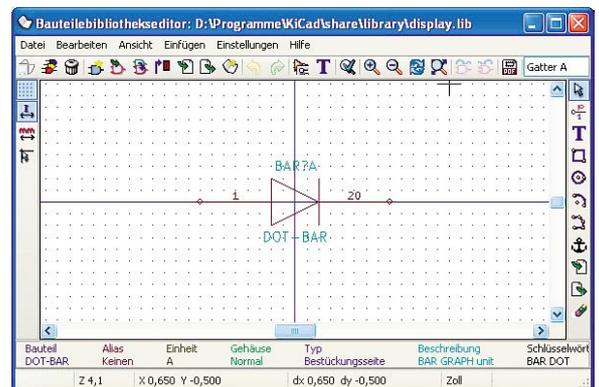


Bild 4: Über den Bauteileditor können vorhandene Bauteile modifiziert oder eigene kreiert werden.

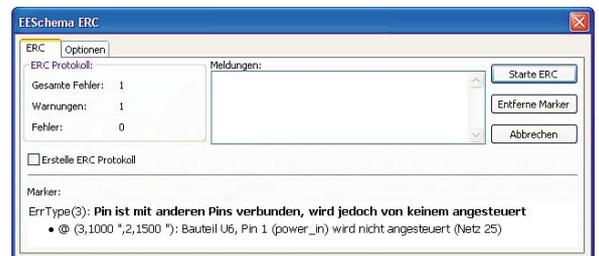


Bild 5: Designfehler in der Schaltung? Der „electric rule check“ bringt es an den Tag.

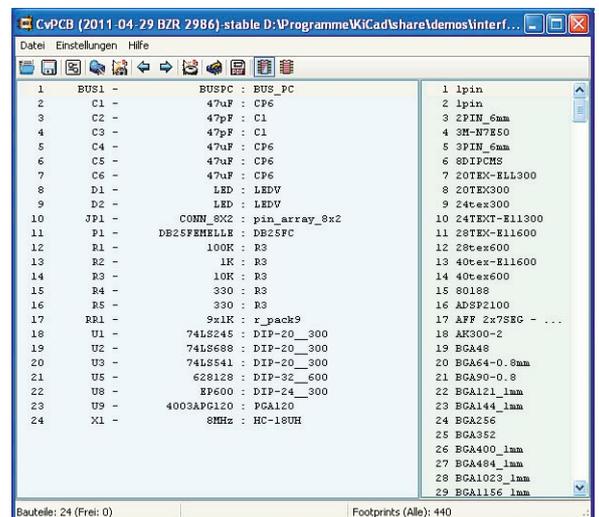


Bild 6: Im Programmteil CvPCB erfolgt die Umwandlung vom Schaltzeichen zum realen Bauteil.

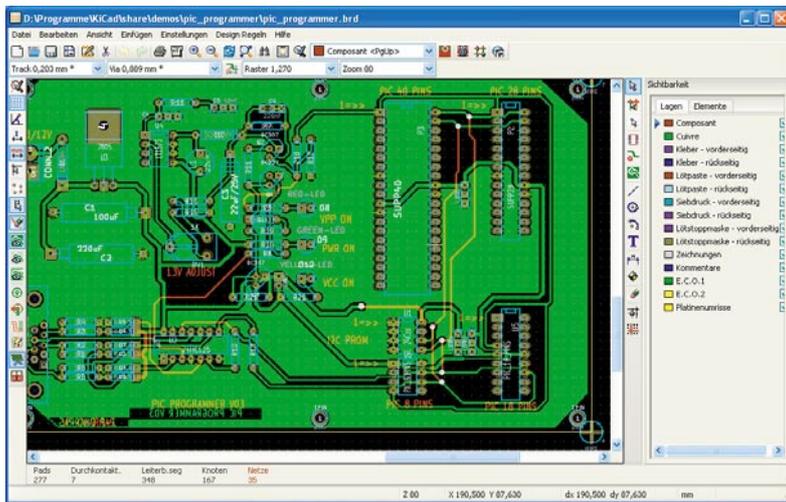


Bild 7: Das Ergebnis der Mühe, das fertige Platinenlayout

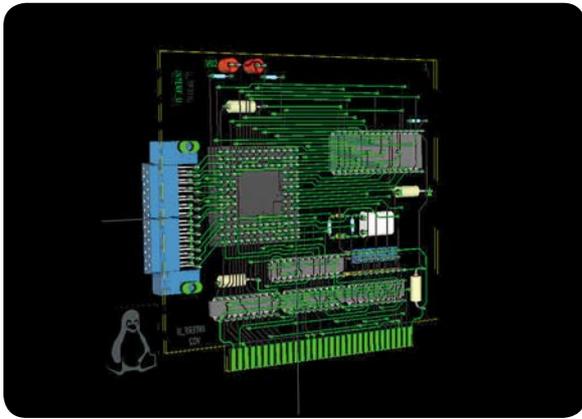


Bild 9: Detailkontrolle bis ins Letzte – hier ist die sichtbehindernde Massefläche ausgeblendet

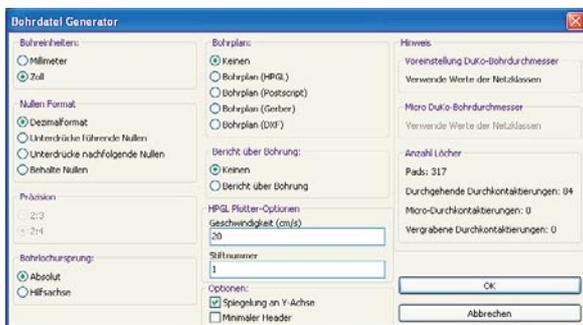


Bild 10: Der Bohrdaten-Generator

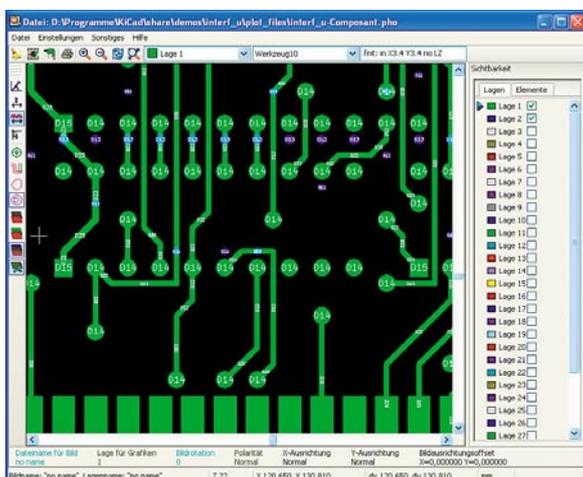


Bild 11: Über „GerView“ sind die Produktionsdaten detailliert kontrollierbar.



Bild 8: Tolles Tool – die Kontrolle der Bauteilanordnung über den 3D-Viewer

Auch hier vermeidet ein „design rule check“ (DRC) Fehler, z. B. zu geringe Abstände. Das fertige Ergebnis (Bild 7) präsentiert sich professionell und braucht sich nicht zu verstecken.

Das Auge isst mit, also soll die Bauteilanordnung auf der Platine nicht nur ordentlich aussehen, es muss auch sichergestellt werden, dass sich Bauteile nicht zu nahe kommen, behindern bzw. kollidieren oder ungünstig gegenseitig beeinflussen. Um dies abzuschätzen, kommt eine weitere Programmkomponente ins Spiel, der 3D-Viewer. Hier kann der Platinenentwurf samt Bauteilen in 3D (Bild 8) von allen Seiten betrachtet und in Ruhe kontrolliert werden, auch ohne Bauteile, Massefläche (Bild 9) usw. So kann wirklich jedes Detail kontrolliert und so manche böse Überraschung nach der teuren Platinenanfertigung von vornherein vermieden werden – für ein Open-Source-System ein sehr bemerkenswertes Feature.

Produktionsunterlagen erzeugen

Ist die fertige Layoutdatei (Dateiendung .brd) abgespeichert, können die Fertigungsdaten direkt aus dem Layouteditor heraus erzeugt werden (Bild 10 zeigt den Generator für die Bohrdaten). Will man sich selbst z. B. einen Film aus dem postscriptfähigen Laser- oder Tintenstrahldrucker ausdrucken oder das Layout in einem postscriptfähigen Computerprogramm verarbeiten, bietet sich die Postscript-Option an, ansonsten wird ein Platinenhersteller meist auf einem Gerber-Datensatz bestehen. Dieses Format können alle Hersteller verarbeiten.

Auch ein direktes Ausdrucken der einzelnen Layer ist möglich, für die Home-Brew-Platine via normalem Drucker wohl der gängigste Weg.

Um den exakten Export der Gerber-Daten für die Fertigung genau kontrollieren zu können, enthält das Programmpaket schließlich „GerView“, einen Betrachter für erzeugte Gerber-Dateien (Bild 11). Hier können sowohl die Gerber-Daten als auch die Excellon-Bohrfiles und die D-Codes, die natürlich auch einzeln anzeigbar sind, wie Bild 11 zeigt, aufgerufen und kontrolliert werden.

So kann man sicher sein, auch exakte Daten in die Produktion gegeben zu haben.

ELV



Weitere Infos:

- [1] www.cadsoft.de
- [2] www.abacom-online.de
- [3] <http://kicad.de.malavida.com/d7802-kostenloser-download-windows>
- [4] http://kicad.sourceforge.net/wiki/DE:Main_Page
- [5] www.kicadlib.org