



von Iznogood

<iznogood/at/iznogood-factory.org>

Über den Autor:

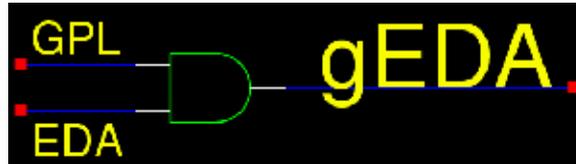
Ich arbeite mit GNU/Linux schon seit einiger Zeit. Im Augenblick benutze ich Debian. Obwohl ich Elektrotechnik studiert habe, arbeite ich hauptsächlich an Übersetzungen für die GNU/Linux Gemeinde.

Übersetzt ins Deutsche von:

Guido Socher

<guido/at/linuxfocus.org>

Wie man gEDA benutzt



Zusammenfassung:

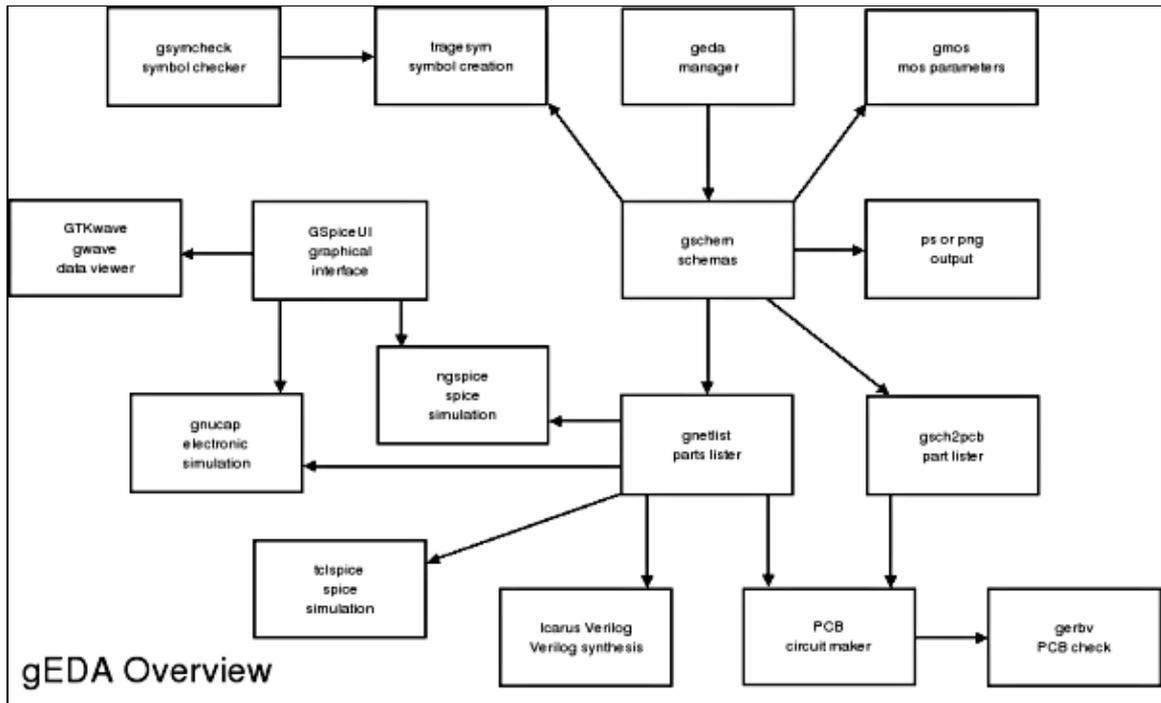
Hier werden wir sehen, wie man eine gedruckte Schaltung mit gEDA erstellt.

Einführung

Warum noch einen Artikel zu EDA (Electronic Design Automation)? Nun es gibt mehr als nur einen Weg und der gEDA Weg ist ein "freier" Weg. Das beste an gEDA ist, daß es ein ganzer Satz von Werkzeugen ist. Einige mit grafischer Oberfläche und einige kommandozeilenbasiert. In diesem Artikel werde ich zeigen, wie man ein PCB (Printed Circuit Board, gedruckte Schaltung) entwirft.

Was ist gEDA?

gEDA ist ein Satz freier Werkzeuge (frei im GPL Sinn). gEDA wird permanent weiterentwickelt. In dem folgenden Diagramm kann man die einzelnen Werkzeuge und ihre Aufgaben sehen. Mit einem Klick auf das Diagramm kann man das Bild vergrößern.



In der Mitte befindet sich gschem. Ein grafischer Schaltbild- und Symboleditor. Dieses Programm kann von geda, dem Manager, aufgerufen werden. Mit geda kann man nicht nur gschem starten, sondern auch eine Spice-simulation, einen PCB Designer und einen Wellenformbetrachter.

Oben links ist tragesym, mit dem man Schaltungssymbole entwerfen kann und gsymcheck zum Überprüfen dieser Symbole. Am wichtigsten für diesen Artikel ist gschem. Interessant ist, dass es zwei Möglichkeiten zum Erstellen der Netzlisten gibt. Eine Besonderheit von gEDA ist, dass alle Programme mit reien Textdateien arbeiten. Die Dateierweiterung zeigt dabei, wofür sie gut sind.

- .sym für Schaltungssymboldateien
- .sch für Schaltbilder
- .net für Netzlisten
- .pcb für PCB Dateien

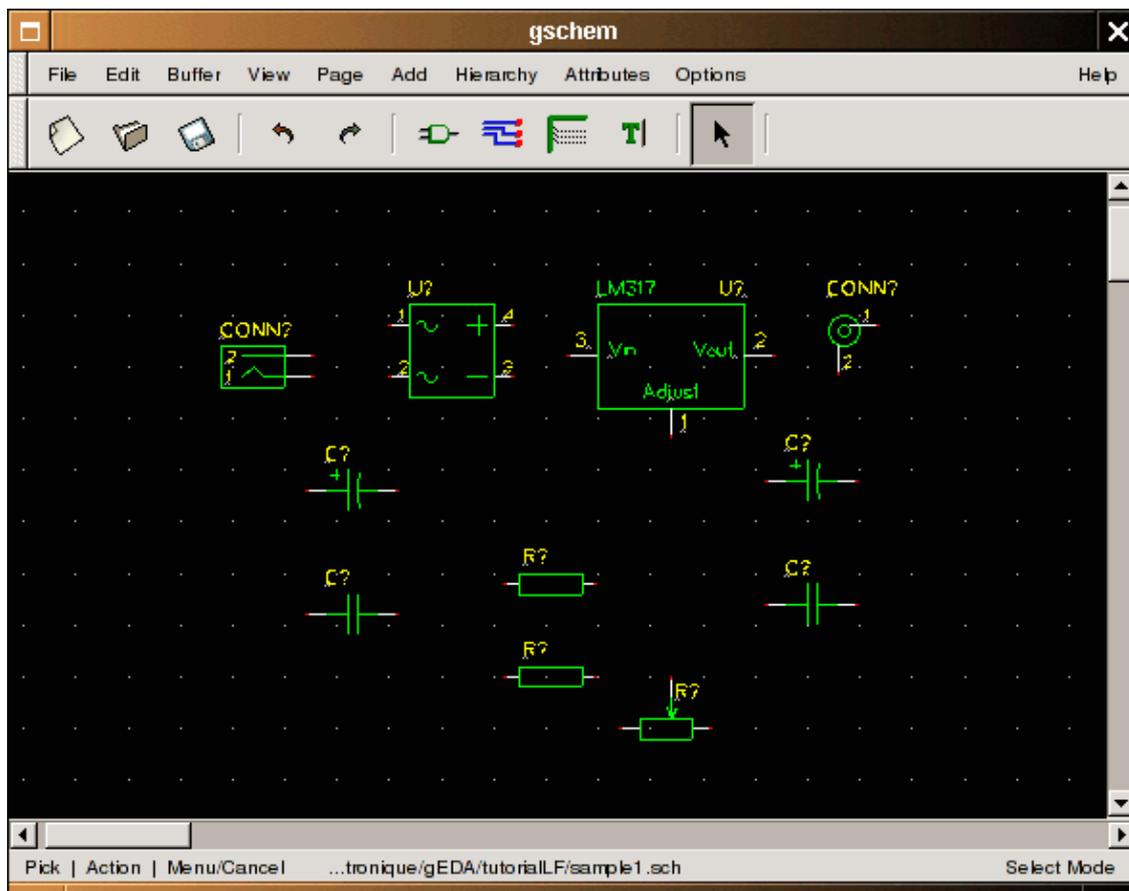
gnetlist ist der "offizielle" Weg, um Netzlisten für Spicesimulation mit tclspice, ngspice, gnucpa, Verilog Synthese mit Icarus oder zum Entwerfen einer gedruckten Schaltung zu erzeugen. gsch2pcb ist das Net-Listen Programm für gedruckte Schaltungen. Es übersetzt .sch Dateien in pcb Dateien. Am Ende kann man die pcb Datei dann mit gerbv auf Fehler überprüfen.

In diesem Artikel werden wir nur über gschem, gsch2pcb und PCB reden. Wir beginnen mit gschem.

Platzieren von Komponenten mit gschem

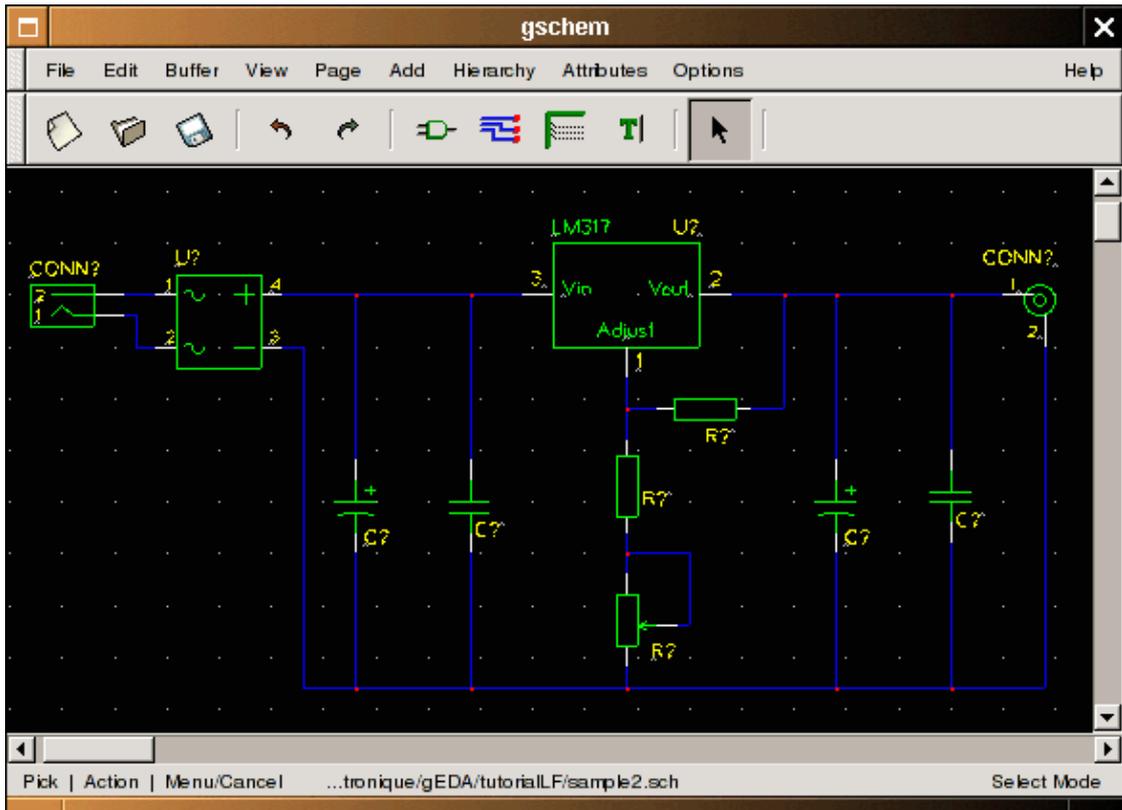
Ich werde hier nicht weiter über die Installation reden. Unter GNU/Debian ist das einfach `apt-get install geda-gschem geda-symbols geda-utils pcb`. Bei anderen Systemen kann man die nötigen Dateien von <http://geda.seul.org/source.html> oder <http://pcb.sourceforge.net/> für PCB herunterladen. OK, laßt uns gschem öffnen. Es gibt zwei Möglichkeiten. Den Befehl `gschem` alleine eintippen oder in Verbindung mit dem Dateinamen: `gschem sample.sch`. Jetzt können wir die Komponenten platzieren. Wieder haben wir zwei Möglichkeiten: mit der Maus und "Add Component" oder mit der Tastatur und "i". Wir selektieren `BNC-1.sym`, `pwrjack-1.sym` unter `connector`, `2 resistor-2.sym`, `1 resistor-variable-2.sym`, `2`

capacitor-1.sym, 2 capacitor-2.sym, bridge-1.sym aus analog und lm317-1.sym aus linear. Wir erhalten folgendes Bild:

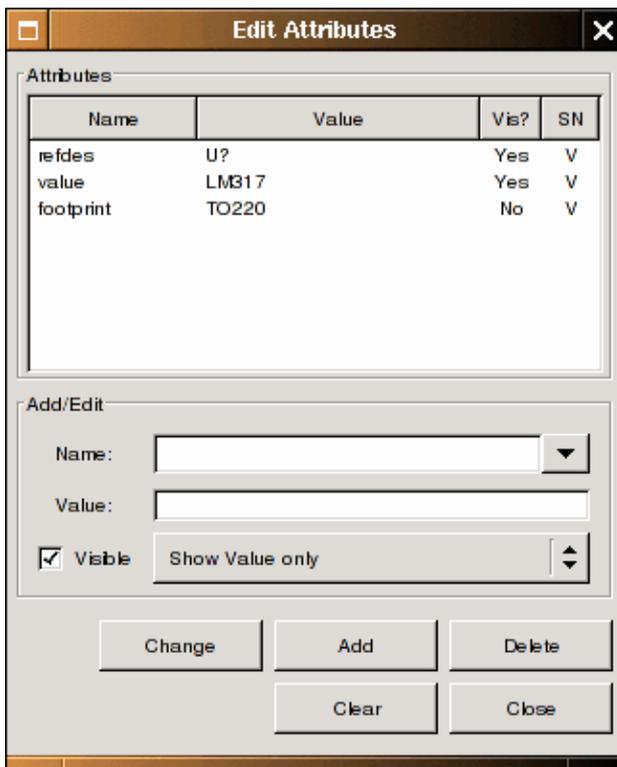


Keine Sorge, die Beispieldateien kann man von meiner Homepage (auf dem neuesten Stand) oder von diesem Server herunterladen.

Nun können wir die Komponenten verdrahten. Dazu müssen wir sie rotieren und verschieben. Da wir faul sind, benutzen wir die Tastenkombinationen. Eine Komponente wird selektiert und mit "er" rotiert, mit "ei" gespiegelt, mit "m" verschoben und mit "n" verdrahtet.



Danach kann man das Attribut einer Komponente mit "ee" editieren. Wir beginnen mit der wichtigsten Komponente: LM317. Der Attributdialog sieht wie folgt aus:



Wir schreiben LM317 als Wert und TO220L als Gehäuse (ich schlage vor, die Gehäuseform unsichtbar zu machen). Für Transistoren ist der Wert wie im Schaltbild zu sehen und die Gehäuseform R025. Kondensatoren, Potentiometer ... gehen alle analog (in der Datei zum Herunterladen ist in sample.sch das Beispiel zu sehen). Wenn wir fertig sind, speichern wir mit "fs", falls die Datei schon einen Namen hat und andernfalls mit "fa".

Übersetzen mit gsch2pcb

Zuerst müssen wir die Komponenten mit folgendem Befehl nummerieren:

```
refdes_renum sample.sch
```

Wir öffnen die Datei nochmal und sehen, daß die Fragezeichen von R?, C? U? sich in Zahlen geändert haben. Zuerst müssen wir eine "footprint library" (Gehäuseformen) installieren und eine Projektdatei erstellen. Ein gutes footprint findet man unter: [http:// web.wt.net/%7Ebillw/gsch2pcb/gsch2pcb-libs.tar.gz](http://web.wt.net/%7Ebillw/gsch2pcb/gsch2pcb-libs.tar.gz). (local copy) Lade es herrunter und installiere es mit:

```
mv gsch2pcb-libs.tar.gz ~/toTheLocationYouWant/  
tar xvzf gsch2pcb-libs.tar.gz
```

Danach erstellt man eine Projektdatei mit:

```
schematics sample.sch  
elements-dir ~/XXX/pcb-elements  
output-name sample
```

Ich habe es sampleLFprojet genannt, aber jeder andere Name geht auch. "schematics" ist für unsere gschem Datei. Elements-dir stellt für pcb das footprint Verzeichnis zur Verfügung (ersetze XXX mit deinem Verzeichnis für pcb-Elemente). Und output-name ist sample damit wir auch bei pcb die Datei sample nennen können.

Als nächstes lassen wir folgendes laufen:

```
gsch2pcb tutorialLFprojet
```

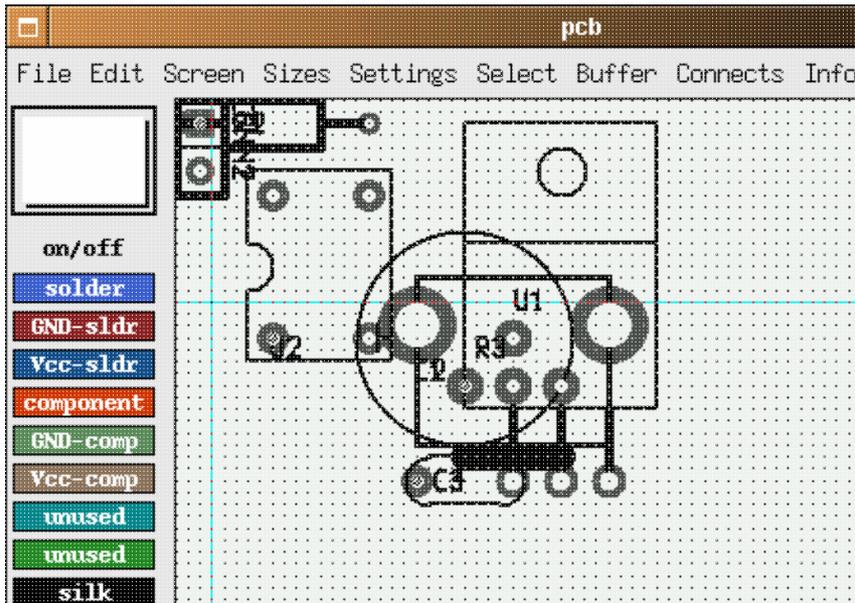
Die entstandenen Dateien sind sample.board und sample.net. Die erste führt alle Komponenten auf und die zweite ist eine net-Liste (Verdrahtungen).

Eine gedruckte Schaltung mit PCB erstellen

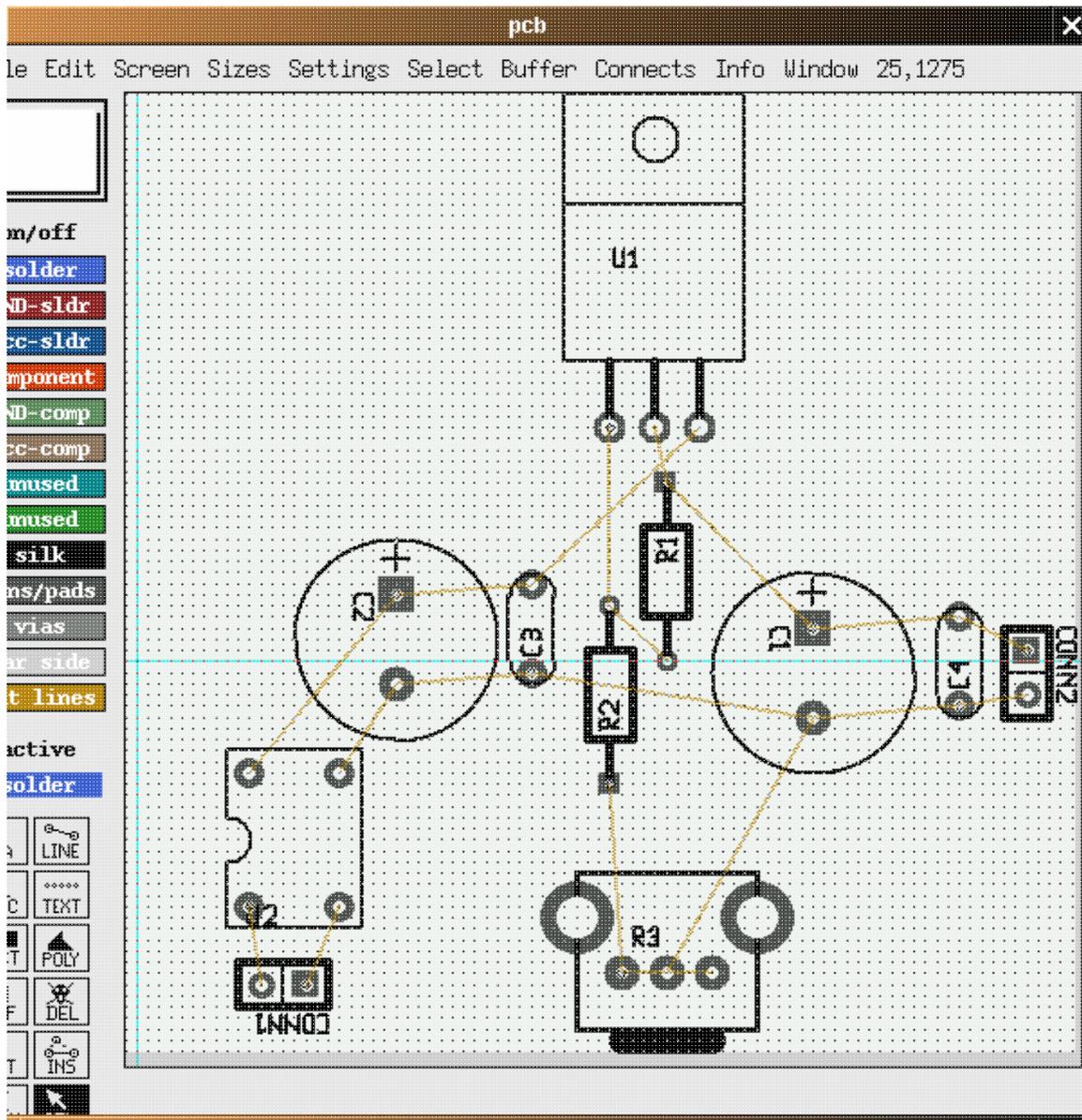
Nun können wir endlich zu PCB kommen:

```
pcb sample.pcb
```

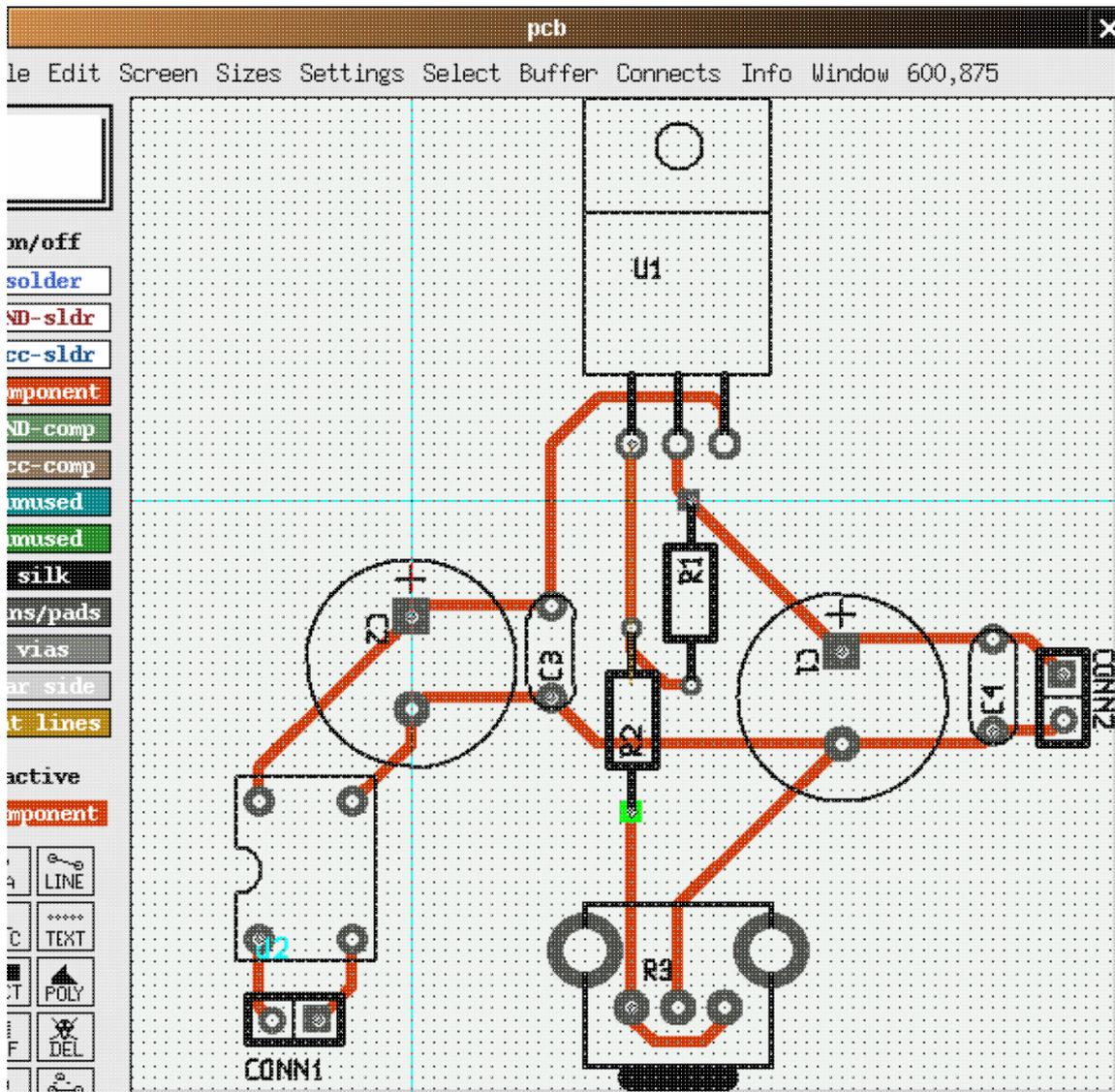
Wir erhalten folgendes Bild:



Alle Komponenten sind auf der linken Seite. Nun müssen wir die Komponenten platzieren. Wir verdrahten sie mit der sample.net Datei mit File --> load net-list file und optimieren mit Connect --> rast-nets. Es ist fast Magie.



Nun platzieren wir die Leiterbahnen wie gewünscht und setzen die Lötmaske.



Das war's: nur noch abspeichern und mit File --> print layout die Schaltung drucken.

Zusammenfassung

Das Ziel dieses Artikles war es, einfach eine direkte und schnelle Einführung zu gEDA zu geben und zu zeigen, wie einfach es ist, eine gedruckte Schaltung zu erstellen. gEDA bietet aber viel mehr. Du kannst jetzt damit herumspielen und du wirst noch viele Möglichkeiten zur Optimierung entdecken, um ein professionelles PCB zu erstellen.

Verwante Artikel

Dieser Artikel wurde inspiriert durch die Dokumentation von Bill Wilson (bill--at--gkrellm.net) gkrellm.net/gsch2pcb.html.

Mehr Informationen finden sich hier: www.geda.seul.org.

PCB gibt es unter pcb.sourceforge.net.

Falls du französisch verstehst, gibt es hier noch einiges: <http://www.iznogood-factory.org> und für

Englischsprechende hier: [hier](#).

<p><u>Der LinuxFocus Redaktion schreiben</u> © Iznogood "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Autoren und Übersetzer: en --> -- : Iznogood <iznogood/at/iznogood-factory.org> en --> de: Guido Socher <guido/at/linuxfocus.org></p>
---	--

2005-01-12, generated by lfparsr_pdf version 2.51