

# Hierarchische Schaltpläne als Schaltplan-Bausteine in KiCad Rev. C - Entwurf

Dipl. Ing. Bernd Wiebus alias dl1eic

23. Dezember 2013

Geschrieben mit  $\text{\LaTeX}$   
und Texmaker 3.3.4.

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>3</b>
<b>2</b>	<b>Hierarchische Schaltpläne in KiCad und ihre Verwendung als Bausteine bei der schnellen Entwicklung neuer Schaltungen</b>	<b>4</b>
<b>3</b>	<b>Das Erstellen von Bausteinen als hierarchische Schaltpläne in KiCad</b>	<b>5</b>
<b>4</b>	<b>Das Einfügen von Bausteinen in Schaltpläne</b>	<b>9</b>
4.1	A . . . . .	9
4.2	B . . . . .	13
<b>5</b>	<b>Zusammenfassung „Buildingblocks/Bausteine“</b>	<b>14</b>
<b>6</b>	<b>Alternative Vorgehensweise/Vorgehensweise für nachträgliche Änderungen</b>	<b>17</b>
<b>7</b>	<b>Das Verdrahten der Bausteine</b>	<b>17</b>
<b>8</b>	<b>Das Zuweisen von Footprints und Values - Grundsätzliches</b>	<b>18</b>
<b>9</b>	<b>Das Zuweisen von Footprints und Values - detailliertes Beispiel</b>	<b>21</b>
9.1	Annotation . . . . .	21
9.2	Annotation - An Subschemen orientiert . . . . .	22
9.3	Zuweisung von Werten . . . . .	26
9.4	Netzliste erstellen und Footprints zuweisen . . . . .	31
9.5	Das Ergebnis in PCBnew . . . . .	31
9.6	Auflösen der Abhängigkeiten zwischen den Subschemen . .	38
<b>10</b>	<b>Ausblick - Anwenden der Buildingblock Methodik auch bei PCBnew</b>	<b>38</b>
<b>11</b>	<b>Verbesserungsvorschläge?</b>	<b>39</b>
<b>12</b>	<b>Impressum</b>	<b>39</b>
	<b>Index</b>	<b>40</b>

# 1 Vorwort

Vorläufig und unvollständig/unvollendet!

Ohne Gewähr auf Richtigkeit!

Mit Vorsicht genießen!

In nachfolgendem Text soll erklärt werden, wie für den von Jean-Pierre Charras und seinem Team unter einer GNU Lizenz erstellten Schaltplan Editor EESchema, das Bestandteil von KiCad ist, aus hierarchischen Schaltplänen Schaltplan-Bausteine (Buildingblocks) erstellt werden können.

Diese können als „Buildingblocks“ aus einer Bibliothek verwendet werden, um daraus modular andere Schaltpläne schnell und effizient zusammenzustellen. Ich möchte an dieser Stelle den Ausdruck Module für die Bausteine vermeiden, weil er in der KiCad-Terminologie einen Footprint meint, und damit missverständlich ist. Der Ausdruck Baustein bzw. Buildingblock trifft die Sache aber genauso gut.

Leider unterstützt KiCad selber diese Methode nur bedingt durch das offene Konzept, aber (noch) nicht durch spezielle Funktionen. Darum sind an einigen Stellen Workarounds und Behelfslösungen nötig. Diese Anleitungen beziehen sich auf Eeschema Version: (2013-11-29 BZR 4513)-product Release build Platform: Linux 3.2.0-4-686-pae i686, 32 bit, Little endian für Linux Debian Wheezy.

Wegen des schnellen Voranschreitens der Arbeit von Jean-Pierre Charras und seiner Mitautoren ist diese Anleitung möglicherweise teilweise veraltet.

Weitere Informationen zu KiCad finden Sie hier unter

<http://iut-tice.ujf-grenoble.fr/kicad/>

und

<http://www.kicad-pcb.org/display/KICAD/KiCad+EDA+Software+Suite>.

Einen Wikipedia-Artikel zu KiCad finden Sie unter

<http://de.wikipedia.org/wiki/KiCad>

und eine deutschsprachige FAQ-Seite zu KiCad unter

<http://www.mikrocontroller.net/articles/KiCAD>.

Zum Inhaltsverzeichnis

## 2 Hierarchische Schaltpläne in KiCad und ihre Verwendung als Bausteine bei der schnellen Entwicklung neuer Schaltungen

KiCad unterstützt hierarchische Schaltpläne. Diese sind ursprünglich gedacht worden, um kompliziertere Schaltpläne durch Unterteilung in Unterschaltpläne bzw. Subschaltpläne übersichtlicher zu machen, doch können Sie umgekehrt natürlich auch verwendet werden, um aus vorhanden Unterschaltplänen von häufig benutzten Schaltungsteilen schnell neue, andere Schaltpläne zu erstellen.

Beispiel: Sehr viele Schaltungen verwenden einen Eingangsgleichrichter mit Siebkondensatoren, Sicherungen ec. , der oft noch von einem Längstregler zur Erzeugung einer stabilen Spannung gefolgt wird. Haben Sie nun einen großen unübersichtlichen Schaltplan, so ist es sinnvoll, diese Baugruppen in separaten Schaltplänen unterzubringen. Im ursprünglichen Hauptschaltplan verbleibt nur ein Kasten mit Anschlüssen und der Verweis auf diesen Unterschaltplan.

Haben Sie aber einmal solche Unterschaltpläne erstellt, so ist es einfach, diese bei der Erstellung neuer Schaltungen, in denen Sie genau die selben Bauteilgruppen verwenden, wiederum einzubinden. Sie ersparen sich dadurch viel Zeit beim Erstellen der Schaltpläne. Ebenso ist es, wenn Sie die gleiche (oder ähnliche) Gruppe mehrmals verwenden. Sie binden eben diesen Unterschaltplan mehrmals in Ihren Hauptschaltplan ein. In letzter Konsequenz könnten Sie es soweit treiben, das Ihr kompletter Hauptschaltplan nur noch aus Unterschaltplänen und den Verbindungen dazwischen besteht.

Da KiCad das Bauteil als Symbol im Schaltplan vom verwendeten Footprint (Modul) auf der Platine trennt, ist die verwendete Technologie fast egal. Sie zeichnen halt Bauteile als Symbol im Schaltplan, und entscheiden später (bei KiCad im Programm CVpcb), welcher konkrete Footprint und damit welche Technologie für ein Bauteil verwendet wird. Damit das aber alles gut klappt, ist einiges zu beachten. Gelegentlic müssen auch Schaltpläne im Filesystem von Hand kopiert bzw. umbenannt werden.

Zur Zeit ist die Verwendung der hierarchischen Schaltpläne als Bausteine nur eine eher zufällige Gelegenheit, die aber praktisch ist. Sie sind aber eigentlich nicht dafür gedacht worden. Möglicherweise könnte KiCad zu diesem Zwecke später einmal handliche Werkzeuge zur Verfügung stellen bzw. spezielle Export Formate schaffen, die noch mehr Möglichkeiten, auch die Footprints und Boards betreffend, eröffnen würden. Siehe dazu den Ausblick

### 3 Das Erstellen von Bausteinen als hierarchische Schaltpläne in KiCad

Sie können natürlich Schaltpläne aus anderen Quellen als aus Ihrer eigenen Hand beziehen, aber zuerst muss natürlich einmal irgendwo ein passender Schaltplan in KiCad erstellt worden sein. Auch wenn ich an dieser Stelle die grundlegenden Funktionen von KiCad als bekannt voraussetze, hier noch einmal eine kurze Wiederholung als Beispiel. Der Schwerpunkt wird dabei auf die Erstellung des Schaltplanes als wiederverwendbarer Buildingblock gelegt.

**A** Starten Sie KiCad und erstellen ein neues, leeres, Projekt . Dazu oben links im Toolbar unter Datei > Neu > leeres Projekt wählen. Im Ordner „Testprojekt“ wird die Projektdatei „Testprojekt.pro“ angelegt. Siehe Abbildung 1. Die Namensendung des Projektfiles mit .pro ist obligatorisch in KiCad. Anschließend finden Sie im KiCad Projektbaum das Projekt Testprojekt.pro, dem noch nichts zugeordnet ist. Siehe Abbildung 2.

**B** Starten Sie, um den Schaltplan zu erstellen, das Programm EESchema (Abbildung 3). Beim Starten werden Sie vermutlich eine Fehlermeldung bekommen, weil der Schaltplan Versuch1.sch noch nicht existiert. Quittieren Sie diese einfach, und Sie sitzen vor dem neuen, noch leeren, Schaltplan. Sobald Sie einmal den Schaltplan speichern, und EESchema verlassen, wird dieser im Ordner Testprojekt als Testprojekt.sch angelegt . Weitere, diesem Projekt zugeordnete Schaltpläne erzeugen Sie aus EESchema heraus, indem Sie oben in der Toolleiste unter Datei entweder NEU oder „Speichern aktueller Schaltplan unter“ wählen (wie üblich). Siehe Abbildung 4.

**C** Zeichnen Sie den Schaltplan wie gewohnt. Wenn Sie den Schaltplan als Baustein verwenden wollen, müssen ALLE Anschlüsse als hierarchische Pins ausgeführt werden. Ich habe den Verdacht, dass es bis auf Ausnahmen es GEFÄHRLICH sein könnte, in Bausteinen von vorneherein globale Label zu verwenden.

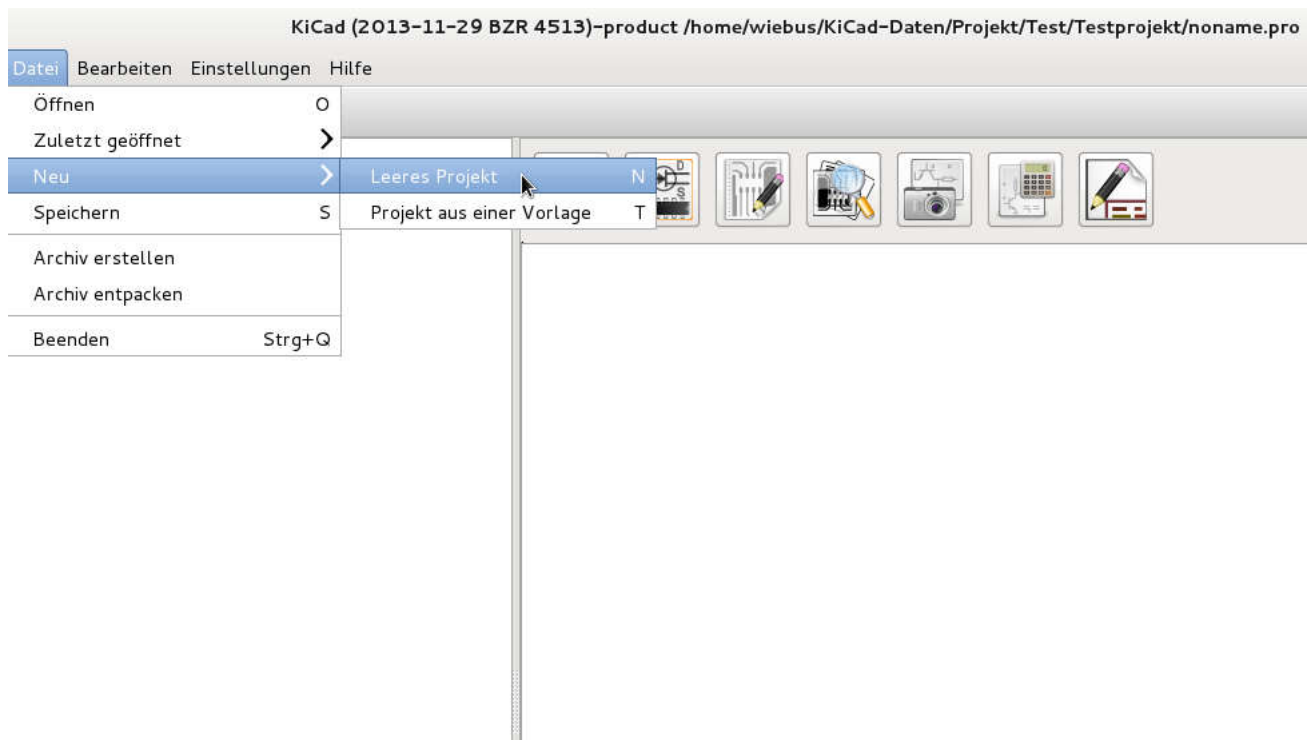


Abbildung 1: Anlegen eines „Testprojektes“ in KiCad

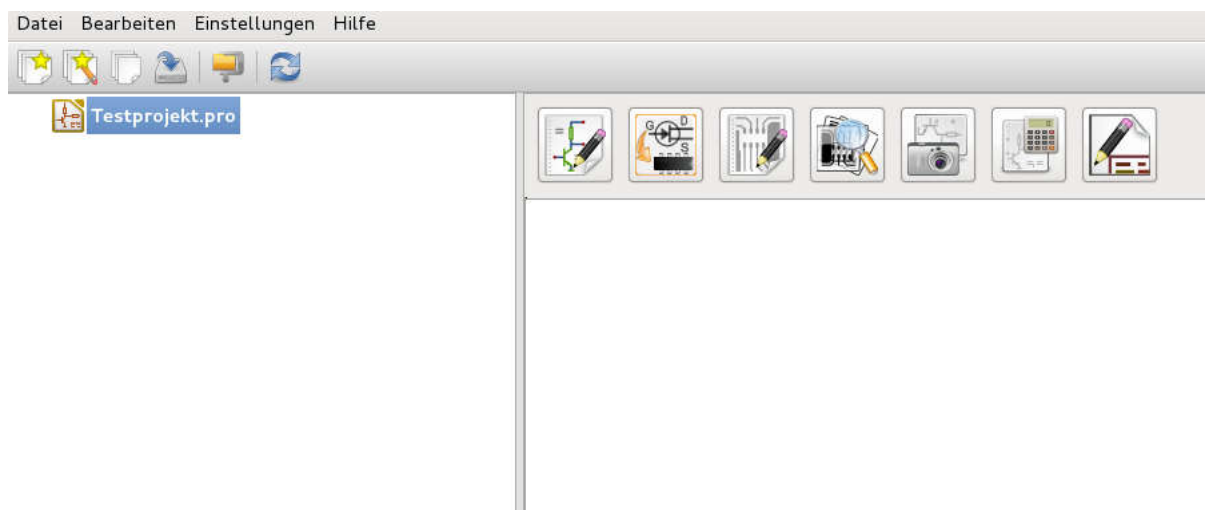


Abbildung 2: Anlegen eines „Testprojektes“ in KiCad: Ergebnis

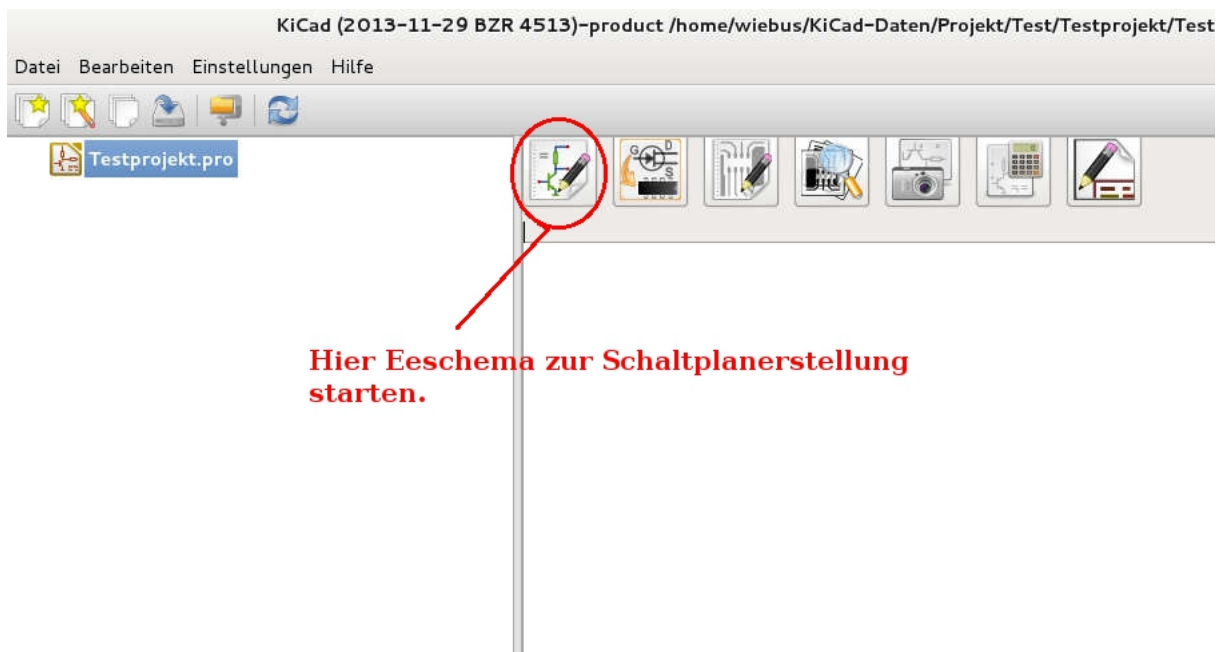


Abbildung 3: Eeschema zur Schaltplanerstellung starten.

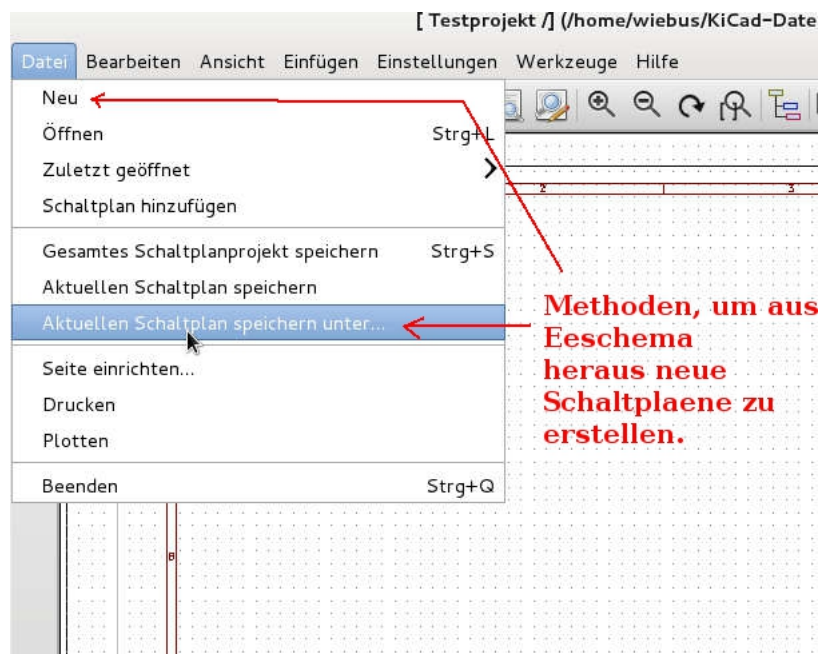


Abbildung 4: Neue Schaltpläne aus Eeschema heraus erzeugen.

Der Grund ist Übersichtlichkeit. Stellen Sie sich vor, Sie hätten in den Bausteinen die Masse jeweils als globales GND definiert. Nun verwenden Sie den Block mehrmals, aber mit unterschiedlichen Bezügen auf unterschiedliche Massen. Dann könnten Sie ein Probleme bekommen, weil irgendwo alle Massen global verbunden werden, aber es nicht offensichtlich ist, dass, wieso und wo dieses passiert.

Zum Erstellen hierarchischer Label verwenden Sie den entsprechenden Button in der rechten Toolleiste (Abbildung 5). Diese hierarchischen Label bilden später, nach Einfügung in einen Hauptschaltplan, die Schnittstellen zu diesem. Siehe hierzu auch die Abbildung 12.

Der fertige Baustein 317Regler.sch ist in Abbildung 6. zu sehen. Er stellt eine Standard Längstregelschaltung mit einem LM317 dar. Die Datei 317Regler.sch ist in den Ordner „Testprojekt“ gebracht worden, wenn Sie nicht gerade erst dort erzeugt wurde. Tatsächlich habe ich aber diese Datei aus einem anderen Ordner, wo ich solche Bausteine vorgefertigt bevorratete, dorthin kopiert.

KiCad erwartet, dass sich die Schaltplandateien von hierarchischen Unterschaltplänen im gleichen Projektordner wie der Hauptschaltplan befinden. Ausserdem ist es sicherer und übersichtlicher, diese Datei in den Projektordner zu kopieren, weil ja an ihr eventuell geändert werden muss. Und diese Änderungen würden die Originaldatei, die ja möglicherweise noch anderswo verwendet werden soll, verfälschen würden. Bedenken Sie dabei, dass dieses Baustein als Teil einer Bibliothek anzusehen ist, die komplette Unterschaltpläne enthält, die hier und anderswo immer wieder verwendet werden sollen. Achten Sie bitte auch auf die Datei 317Regler-cache.lib. Sie wird spätestens angelegt, wenn Sie den (Original)Schaltplan speichern und EESchema verlassen. Diese „-cache.lib“ enthält eine Library mit den im Schaltplan verwendeten Symbolen. Sie ist wichtig, wenn Sie die Bibliothek in ein anderes Projekt bringen wollen, wo möglicherweise eine andere Symbollibrary existiert, die mit ihrer nicht zusammenpasst. Diese XXX-cache.lib sollte der Übersichtlichkeit halber bis auf den „-cache.lib“ Teil immer den gleichen Namen wie die Schaltplandatei haben, obwohl dieses nicht zwingend notwendig ist.

Die Projektdatei des Buildingblocks selber benötigen Sie dazu nicht. Diese ist nur nötig, wenn Sie den Buildingblock selber im Original bearbeiten.

Zum Inhaltsverzeichnis



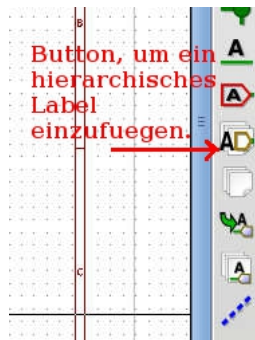


Abbildung 5: Ein hierarchisches Label zufügen

## 4 Das Einfügen von Bausteinen in Schaltpläne

### 4.1 A

Nun wollen Sie diesen vorproduzierten Buildingblock 371Regler.sch in den Hauptschaltplan Testprojekt.sch als hierarchischen Unterschaltplan einfügen.

Dazu muss er aber schon existieren, unabhängig davon, aus welcher Quelle er kommt.

Nehmen wir an, Sie würden diese Längstregelung in dem Schaltplan dreimal verwenden wollen. Dazu wird ein entsprechender hierarchischer Unterschaltplan angelegt, indem Sie den entsprechenden Modus Button, wie in Abbildung 7 gezeigt, anwählen. Dann erstellen Sie den Subschaltplan als Rechteck, indem Sie mit einem linken Mausklick zuerst seine obere linke Ecke, und mit einem zweiten linken Mausklick seine untere rechte Ecke festlegen. Wenn Sie den zweiten Mausklick tätigen, poppt ein Fenster auf, in dem Sie die Schaltplandatei und den Schaltplannamen eintragen können (siehe Abbildung 8). Per Defaulteinstellung wird hier nach einem Algorithmus ein unverwechselbarer, aber weitestgehend nichtssagender Dateiname und Schaltplanname eingetragen. Das sollten Sie so nicht übernehmen, auch dann nicht, wenn Sie keinen schon existierenden Schaltplan übernehmen wollen, sondern vergeben Sie sinnvolle Namen, an denen Sie den Schaltplan bzw. die ihn enthaltene Datei gut wiedererkennen können. Der Dateiname und der Schaltplanname müssen nicht übereinstimmen. Tatsächlich können sie das auch nur, wenn der verwendete Block nur einmal im Schaltplan auftaucht. Der Schaltplanname darf nur einmal im Schaltplan vorkommen, aber mehrere Schaltpläne mit darum voneinander abweichendem Namen können auf die gleiche Schaltplandatei verweisen. Weil das hier auch so sein wird, wird der erste Schaltplanname zu „Sheet317Regler-1“ gewählt, und auf die schon existierende Datei

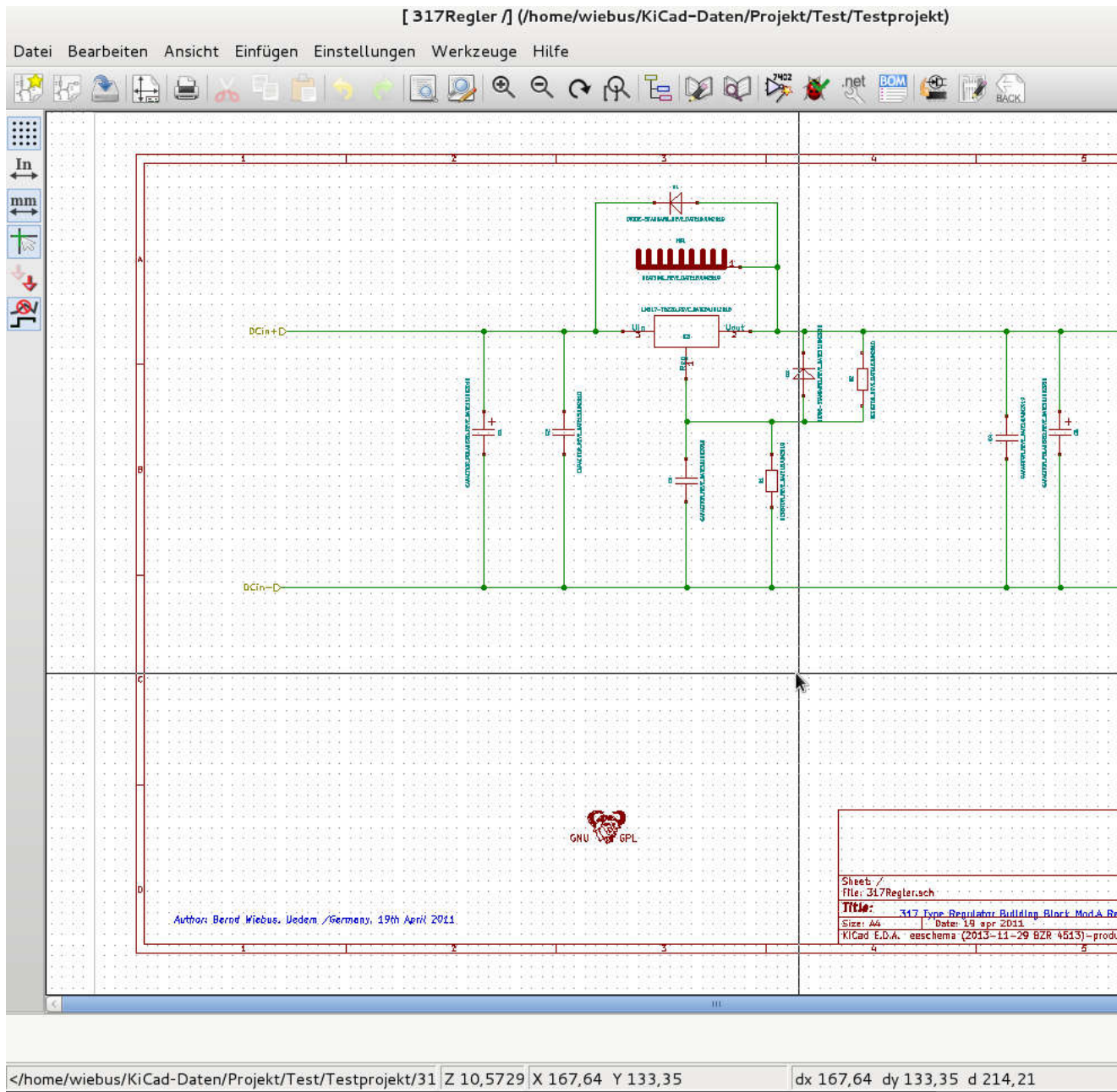


Abbildung 6: Ein Schaltplan in KiCad

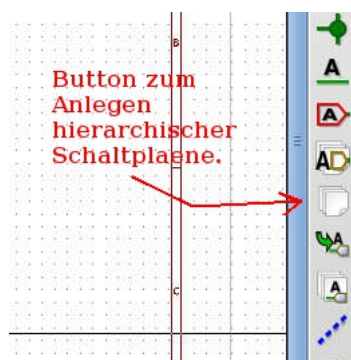


Abbildung 7: Button zum Anlegen eines Subschaltplanes

„317Regler.sch“ verwiesen. Wenn Sie das ganze nun mit <Enter> bestätigen, stellt Kicad fest, das die gewählte Schaltplandatei schon existiert und fragt zur Vorsicht nach, ob ein Schaltplan aus dem Inhalt dieser Datei erstellt werden soll . Das bestätigen Sie mit „JA“.

Nun haben Sie einen (fast) fertigen Unterschaltplan, wie in Abbildung 9 zu sehen ist.

#### Zum Inhaltsverzeichnis

Sie können diesen Unterschaltplan öffnen, in dem Sie entweder einen Doppelklick darauf machen, solange Sie KEIN Tool aktiviert haben, oder indem Sie oben aus der Toolleiste den Button zum Navigieren in der Schalplanhierarchie auswählen (Abbildung 10).

Wenn Sie dieses machen, werden Sie feststellen, dass alle Symbole durch Kästchen mit Fragezeichen rsetzt wurden. Dieses bedeutet, dass die oben erwähnte Cache-Bibliothek für die Symbole nicht in die Liste eingetragen wurde (Abbildung 11). Wählen Sie dazu aus der oberen Menüleiste Einstellungen > Bibliothek. Es poppt ein Fenster auf. Dort wählen Sie auf der rechten Seite oben den Button „Hinzufügen“ und wählen aus der erscheinenden Dateiauswahl die Cache-Bibliothek für die Symbole aus. In dem Falle hier „317Regler-cache.lib“ im Ordner „Testprojekt“ . Anschließend sollte der Inhalt des Unterschaltplanes so aussehen, wie in Abbildung 12.

Dabei ist zu erinnern, dass das Original des Unterschaltplanes mit hierarchischen Labeln an den Anschlüssen zum übergeordneten Schaltplan ausgeführt war .

Wenn nicht, muss das nachgeholt werden. Es ist aber zu empfehlen, bei Schaltplänen, die als Unterschaltpläne verwendet werden sollen, insbesondere, wenn sie dazu entworfen werden, um in eine Bibliothek von „Building-

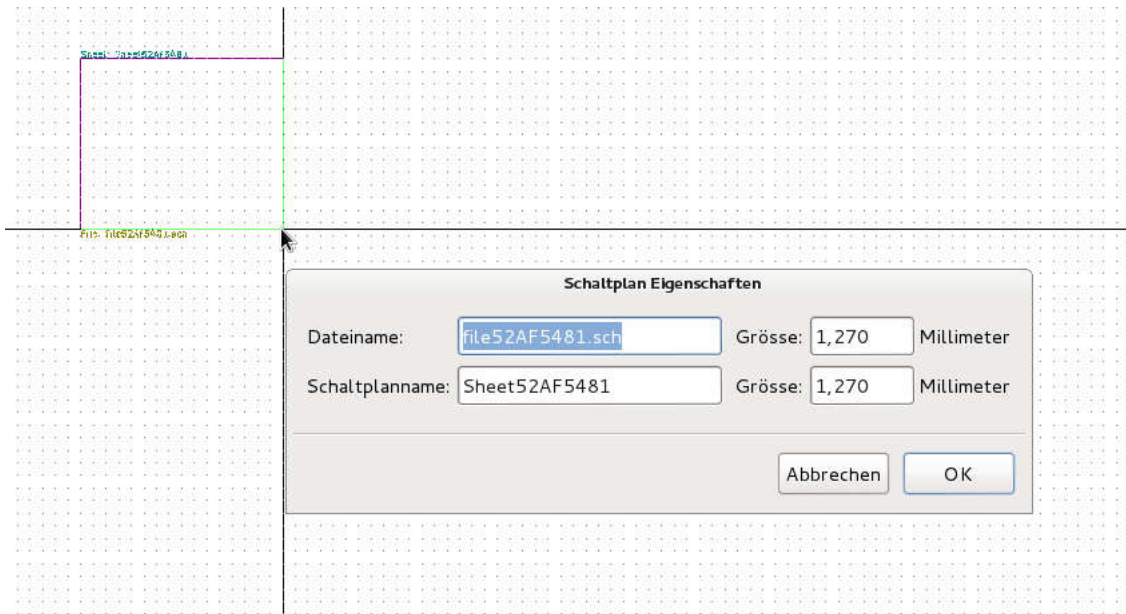


Abbildung 8: Anlegen eines Subschaltplanes

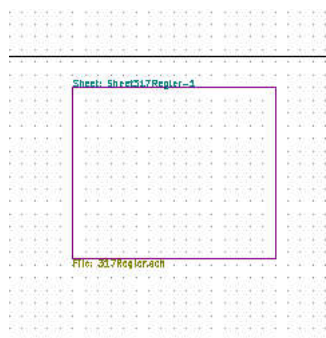


Abbildung 9: Fertiger Subschaltplan

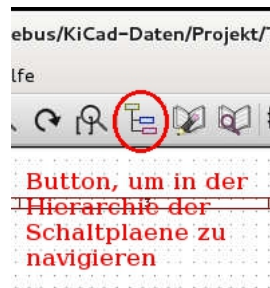
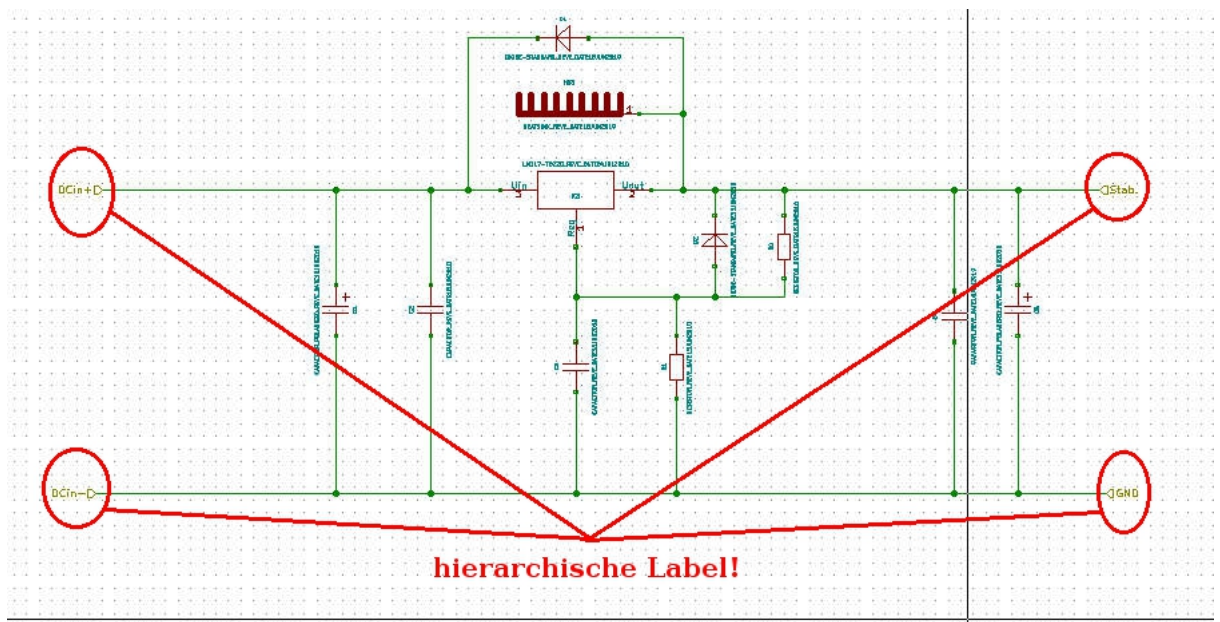


Abbildung 10: Button zum Navigieren in der Schaltplanhierarchie

blocks“ eingefügt zu werden, dieses von vorneherein zu machen.

Zum Inhaltsverzeichnis



## 4.2 B

Im Hauptschaltplan sollen aber nun drei solcher hierarchischer Unterschaltpläne verwendet werden. Einen gibt es schon, zwei fehlen noch. Die restlichen beiden können Sie einerseits auf dem gleichen Wege erstellen wie den ersten, wobei Sie immer auf die gleiche Datei „317Regler.sch“ verweisen wie beim ersten, aber nun die abweichenden Schaltplannamen „Sheet317Regler-2“

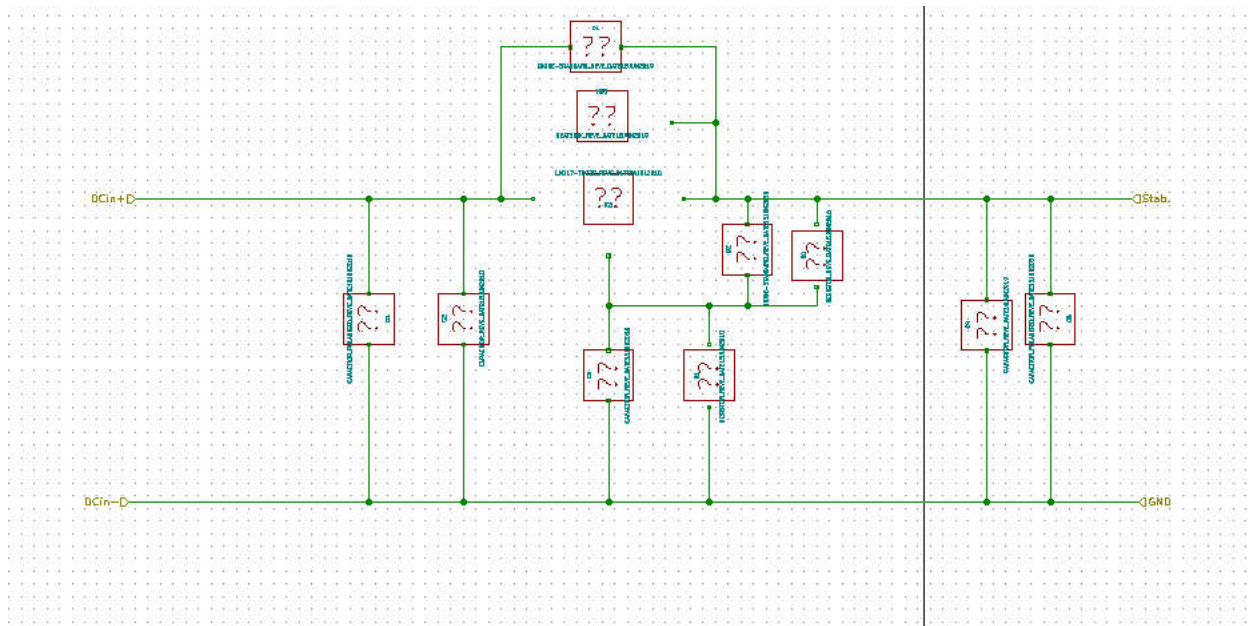


Abbildung 11: Fehlende Cache-Bibliothek für die Symbole des Unterschlupfes

und „Sheet317Regler-3“ verwenden. Das ist aber umständlich. Etwas einfacher geht es, indem man den schon vorhandenen Unterschlupf als Gruppe markiert, und kopiert neu ablegt. Anschließend muss der neue Unterschlupf mit einem rechten Mausklick zum Editieren geöffnet werden, und die Schaltplannamen entsprechend in „Sheet317Regler-2“ und „Sheet317Regler-3“ geändert werden.

Speichern Sie alles. Beachten Sie, dass der Schaltplannamen nicht zwangsweise mit dem Namen der Schaltplandatei identisch ist.

Zum Inhaltsverzeichnis

## 5 Zusammenfassung „Buildingblocks/Bausteine“

Das Verwenden von Buildingblocks ist also einfach das Verwenden von hierarchischen Schaltplänen mit Schaltplandateien, welche schon vorgefertigt existieren.

Dazu wird eben beim Anlegen des hierarchischen Subschaltplan Blattes auf diese schon existierende Datei verwiesen, die sich dabei im Projektordner

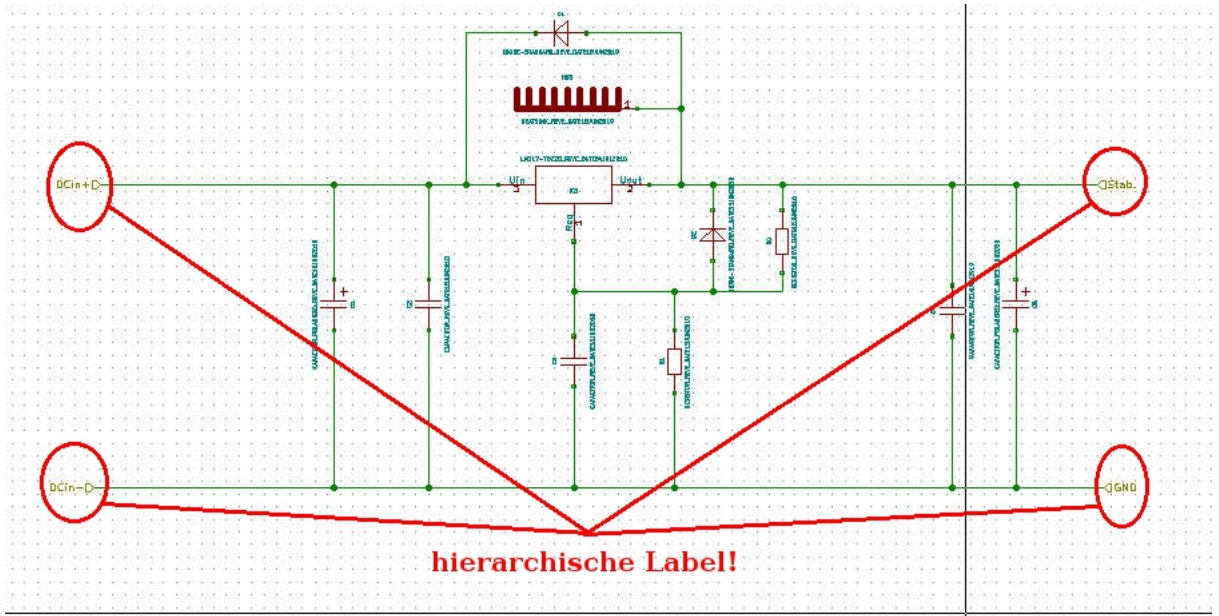
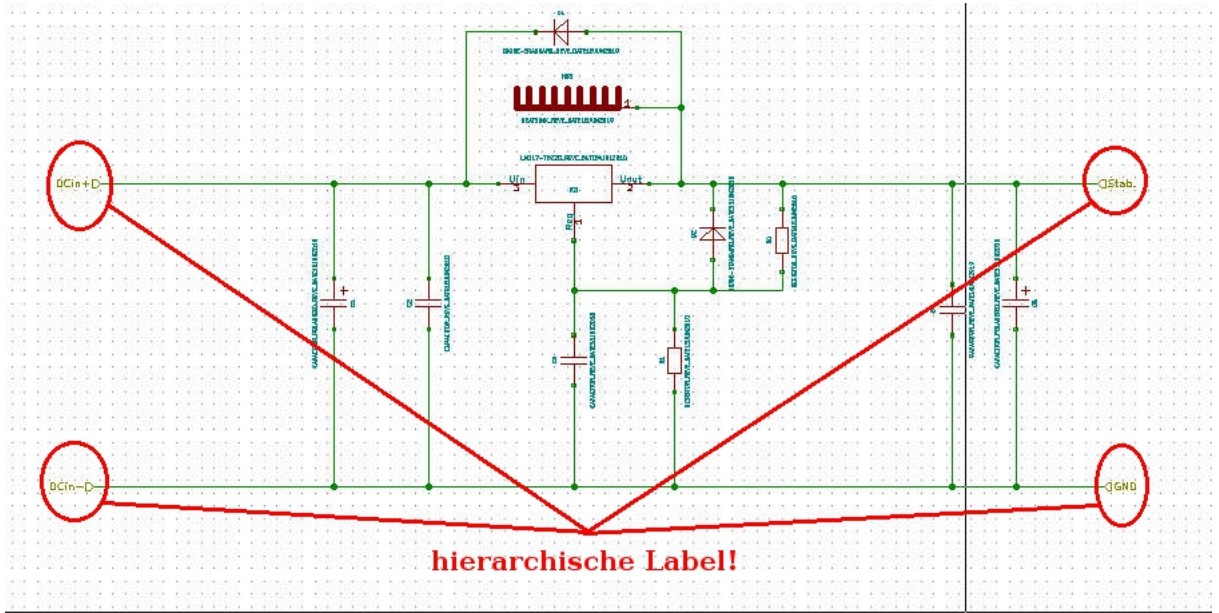


Abbildung 12: Fertiger Unterschlupplan mit hierarchischen Labeln

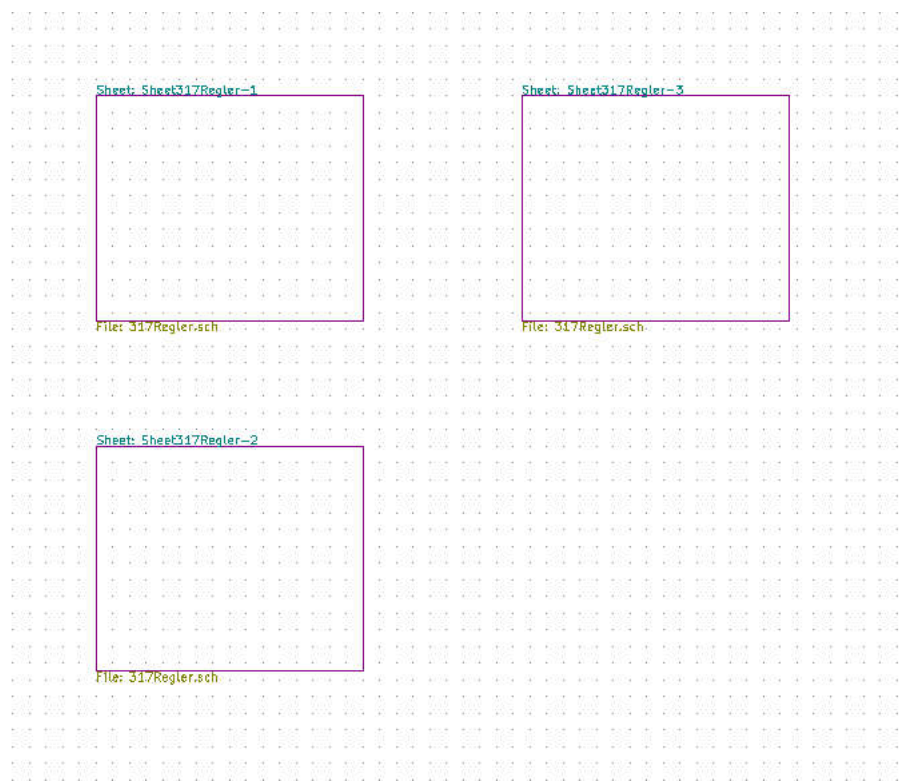


Abbildung 13: Drei Unterschaltpläne



befinden muss, und der Pfad zum Symbolcache dieser Schaltplandatei muss in die Liste der verwendeten Symbollibraries eingetragen werden.

Zum Inhaltsverzeichnis

## **6 Alternative Vorgehensweise/Vorgehensweise für nachträgliche Änderungen**

Die andere Reihenfolge, also zuerst das Anlegen von hierarchischen Unterschaltplänen, und dann das Hinzufügen der Buildingblocks, funktioniert auch, ist aber etwas umständlicher. Es kann aber nützlich sein, auch diesen Weg zu kennen, weil er bei Änderungen bestehender Schaltpläne genutzt werden muss. Legen Sie dazu zuerst den hierarchischen Unterschaltplan an. Dann wählen Sie aus der Menüleiste oben unter Datei die Aktion „Alle Schaltpläne speichern“. Es werden leere Subschaltpläne mit dem vorher gewählten Dateinamen erzeugt. Beenden Sie Eeschema, und löschen Sie mit einem Dateiverwaltungsprogramm diese Dateien und ersetzen Sie diese Dateien durch Dateien gleichen Namens, die Sie durch Kopieren und Umbenennen aus den Buildingblock Originaldateien erzeugt haben. Vergessen Sie nicht, die Symbol-Caches in die Liste der verwendeten Bibliotheken einzutragen. Dieses könnte passiert sein, wenn Sie irgendwo Fragezeichen als Symbole vorfinden, wie z.B. in Abbildung 11 gezeigt.

Zum Inhaltsverzeichnis

## **7 Das Verdrahten der Bausteine**

Gehen Sie in den Hauptschaltplan und wählen Sie dazu aus der rechten Werkzeugleiste mit dem passenden Button das Importieren hierarchischer Pins (Abbildung 14). Klicken Sie nun in einen Subschaltplan, und Sie erhalten ein hierarchisches Label aus dem Subschaltplan als Pin außen am Subschaltplan, den Sie mit der Maus verschieben und mit einem Mausklick platzieren können. Dies können Sie nun wiederholen, bis alle Anschlüsse der hierarchischen Schaltpläne als Pins an den Symbolen der Unterschaltpläne vorhanden sind. Siehe Abbildung 15. Diese Pins entsprechen den hierarchischen Labeln im inneren des Subschaltplanes. Diese Pins können Sie nun als ganz gewöhnliche Pins in KiCAD untereinander und mit weiteren Bauteilen



Abbildung 14: Button um hierarchische Pins aus Unterschaltplänen zu importieren

verdrahten. Siehe dazu Abbildung 16.

[Zum Inhaltsverzeichnis](#)

## 8 Das Zuweisen von Footprints und Values - Grundsätzliches

Wenn zwei (oder mehr) Subschaltpläne angelegt werden, die auf die gleiche Baustein -Datei verweisen, ist das erst einmal kein Problem. Die Annotation ist in der Lage, diese zwei (oder auch noch mehr solcher] Schaltpläne auseinanderzuhalten. Die Bauteile gleicher Position in den unterschiedlichen Subschaltplänen werden korrekt durchgezählt und auch richtig unter diesen Referenznummern mehrfach in der Netzliste angezeigt . Dort können ihnen mit CVpcb dann Footprints zugewiesen werden. Es können durch unterschiedliche Referenznummern unterschiedenen Bauteilen auch unterschiedliche Module/ Footprints zugewiesen werden. Inwieweit das aber sinnvoll ist, ist im Detail zu überlegen.

Da diese Subschaltpläne aber alle auf die gleiche Schaltplandatei verweisen, und dort nur einmal ein Value eingetragen werden kann, sind auch durch unterschiedliche Referenznummern unterschiedene Bauteile in gleicher Position immer mit dem gleichen Value versehen, selbst wenn sie unterschiedliche Footprints zugewiesen bekommen haben.

Es hängt nun von ihrem persönlichen Umgang mit Value und Footprint in der Stückliste (BOM „Bill of materials“) ab, wie das zu Handhaben ist.

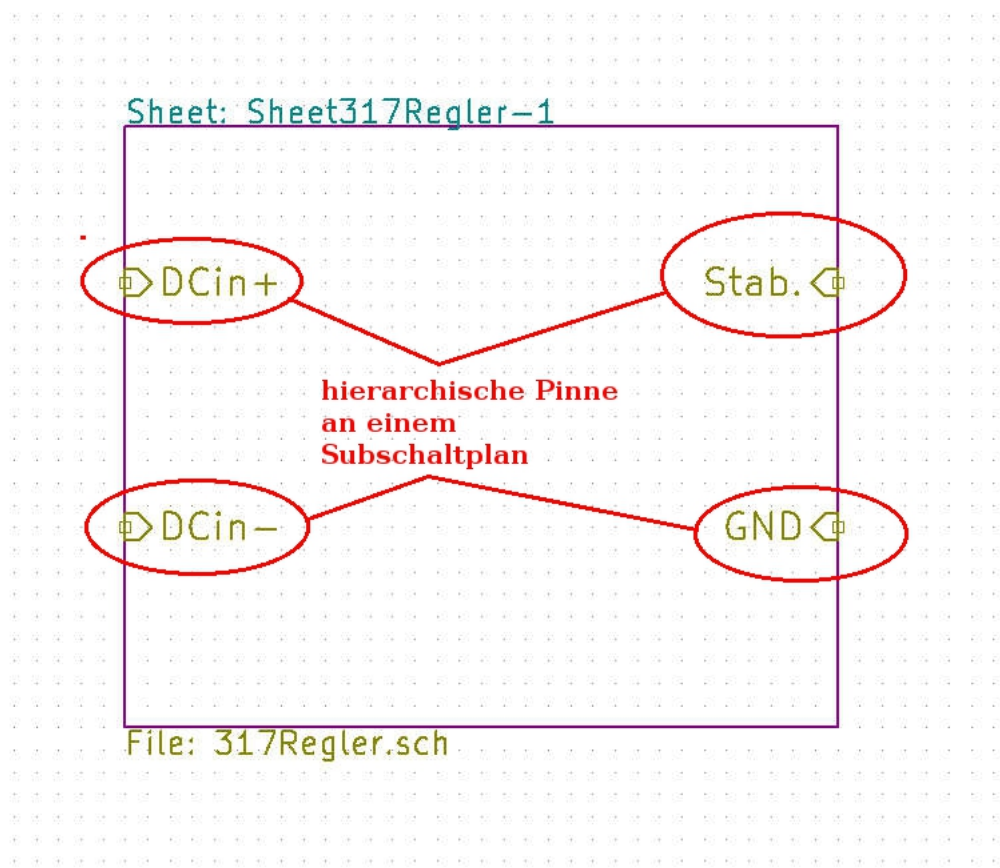


Abbildung 15: Mit hierarchischen Pins versehener Subschaltplan

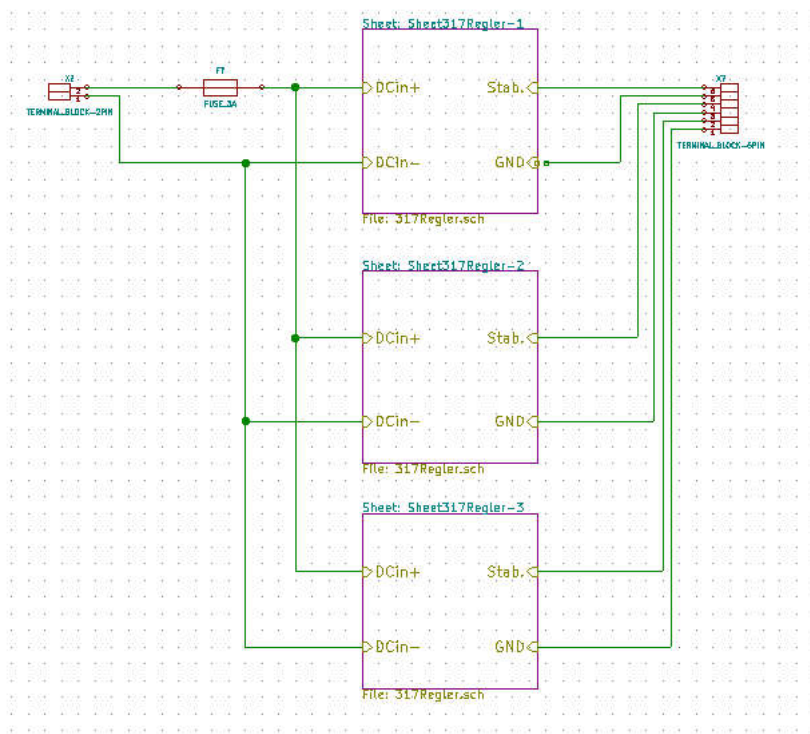


Abbildung 16: Verdrahtete Subschaltpläne

Unterschiedliche Footprints mit daraus resultierender unterschiedlicher Technologie aber gleichen „Value“-Eigenschaften sollten eigentlich auch im Value erkennbar sein. Z. B. als Value nicht nur 10k angeben sondern umfassender 10k/0805 oder 10k/TH-1/3W-RM10mm.

Die sauberste Lösung wäre daher, sobald sich die Subschaltpläne auch nur in einer Kleinigkeit unterscheiden, egal ob im Value (dann ist es sowieso zwingend nötig) oder im Footprint, eine Kopie des Subschaltplandatei unter anderem Dateinamen anzulegen, und dann darauf zu verweisen. Dann unterscheiden sich die Subschaltpläne nicht nur im Namen, sondern auch in der hinterlegten Datei, und in zwei Dateien können dann sehr gut auch unterschiedliche Values eingetragen werden.

Wie das im Detail zu verstehen ist, zeigt nachfolgendes Beispiel.

Zum Inhaltsverzeichnis

## 9 Das Zuweisen von Footprints und Values - detailliertes Beispiel

### 9.1 Annotation

Wir wenden auf den so erstellten Schaltplan die Autoannotationsfunktion an. Diese wird über den in Abbildung 17 zu sehenden Button gestartet. Es poppt ein Menü auf, in dem verschiedene Einstellungen getätigt werden können. Siehe Abbildung 18.

„Starte Annotation“ startet eben dieselbe. Wenn jetzt Warnungen über „Mehrfachelemente“ erscheinen, wie in Abbildung 19, so ist dieses ein Hinweis, das Referenzen doppelt vergeben wurden. In unserem Falle ist dieses passiert, weil der importierte Schaltplan „317Regler.sch“ schon eine Annotation hatte, die nun durch das Kopieren mehrfach auftritt.

Es gibt grundsätzlich drei Möglichkeiten, dem abzuweichen.

A) Durch Verwenden eines Ursprungsschaltplanes OHNE Annotation. D.h. bei „317Regler.sch“ hätte direkt zu Anfang keine Annotation vorhanden sein dürfen, bzw. eine vorhandene hätte gelöscht werden müssen. Dazu ist es aber nun zu spät.

B) Indem man die vorhandene Annotation löscht. Siehe Abbildung 20 unten.

C) Indem man Eeschema die Möglichkeit einräumt, doppeltbelegte Annotationen zu ersetzen. Siehe Abbildung 20 in der Mitte.

Für den angenommenen Fall würde ich aus Erfahrung die Methode C vorziehen. Methode B und C sind sich relativ ähnlich, und für den Fall hier läuft es auf

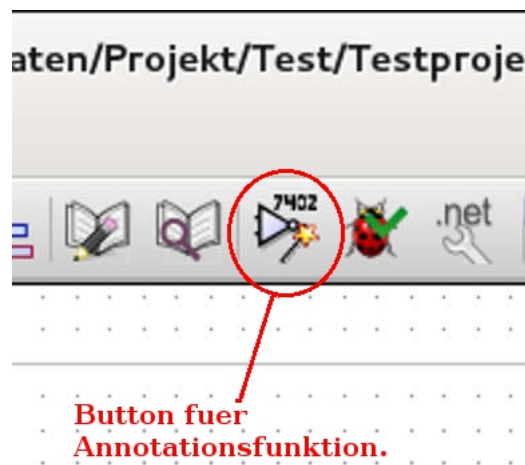


Abbildung 17: Der Button für die Annotationsfunktion

das Gleiche hinaus.

Methode A) ist nur schwer stringent einzuhalten. Theoretisch würde man die Bibliotheken ohne Annotation vorhalten, aber praktisch braucht man sie auch dort, wenn die Bibliothek bearbeitet oder präsentiert werden soll. Und oft wird dann anschließend das Löschen vergessen.

[Zum Inhaltsverzeichnis](#)

## 9.2 Annotation - An Subschaltplänen orientiert

Wenn Subschaltpläne verwendet werden, ist es oft sinnvoll und angenehm, die Strukturierung der Nummerierung an die Unterschaltpläne anzupassen, so dass anhand der Nummern auf den Unterschaltplan geschlossen werden kann. KiCad kann das automatisch erledigen, indem im Annotationsfenster entweder "Verwende erste freie Nummer bis Schaltplannummer x 100" oder „Verwende erste freie Nummer bis Schaltplannummer x 1000“ aktiviert wird. Siehe Abbildung 21.

Der Effekt ist folgender:

Alle Bauteile im Hauptschaltplan erhalten als erste Ziffer der Referenznummer eine 1. Alle Bauteile im ersten Unterschaltplan erhalten als erste Ziffer der Referenznummer eine 2. Alle Bauteile im zweiten Unterschaltplan erhalten als erste Ziffer der Referenznummer eine 3. usw.

Das macht KiCad nun nach Präfixen getrennt. Alle Widerstände im ersten

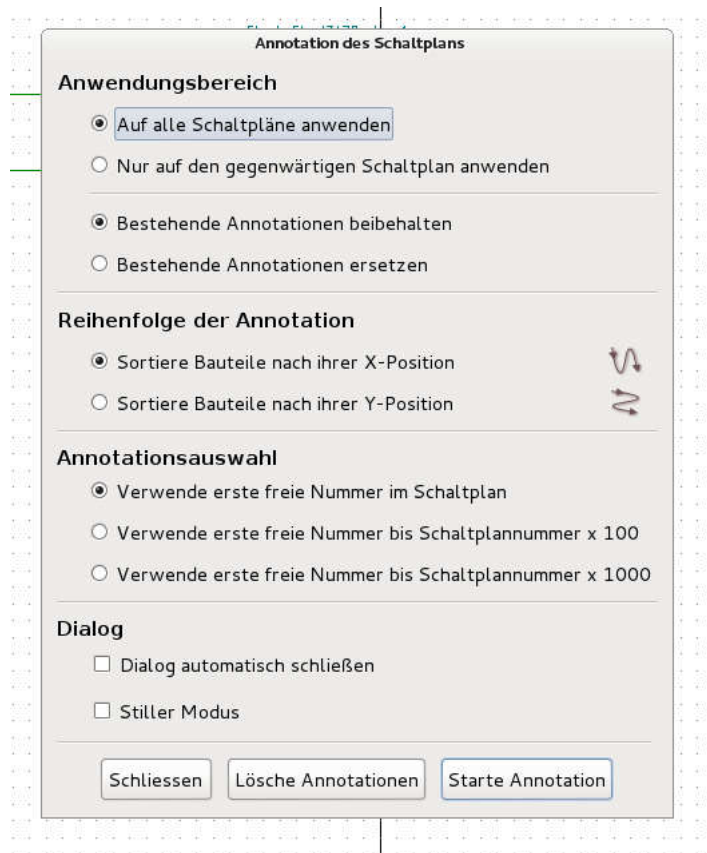


Abbildung 18: Menü der Annotationsfunktion

+

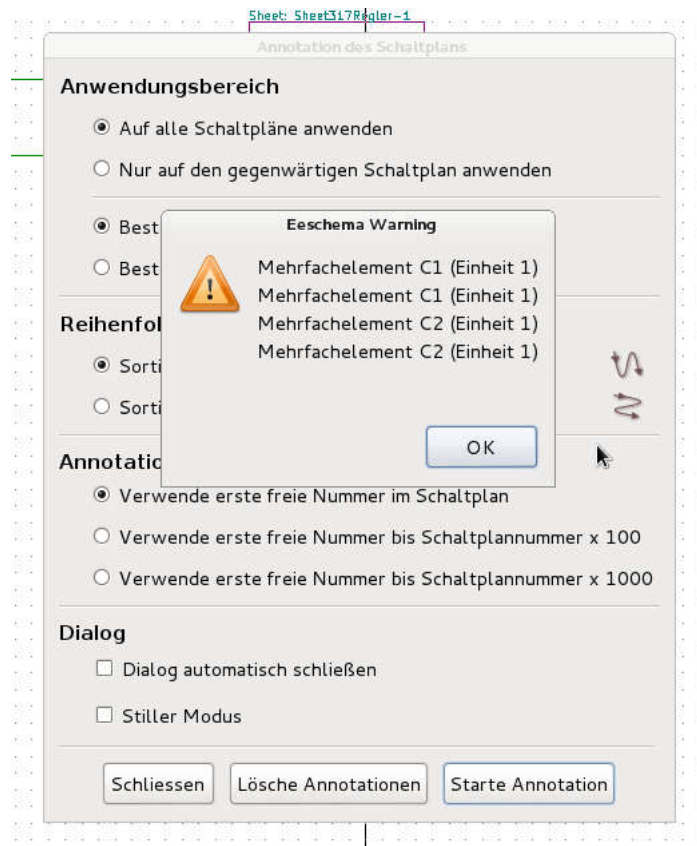


Abbildung 19: Warnung bei mehrfach vergebenen Referenzen



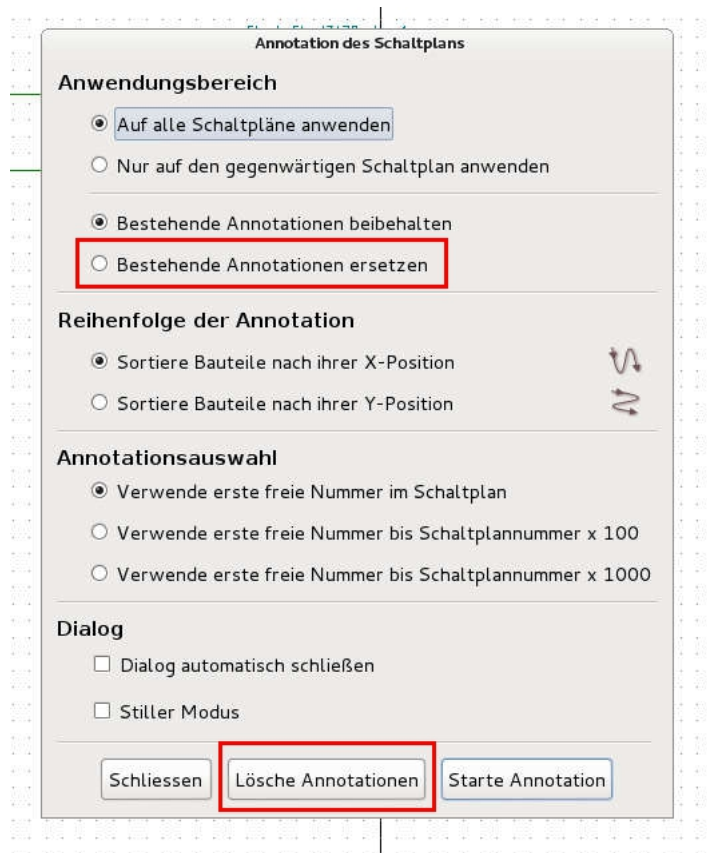


Abbildung 20: Annotationen Ersetzen bzw. Löschen

Unterschaltplan erhalten also die Nummerierung R201 bis R299, alle Kondensatoren C201 bis C299 usw. Das heißt, pro Präfix und Unterschaltplan können so 99 Bauteile strukturiert annotiert werden, wenn man die Methode “Verwende erste freie Nummer bis Schaltplannummer x 100“ gewählt hat. Kommt man damit nicht aus, so kann man noch die Methode „ Verwende erste freie Nummer bis Schaltplannummer x 1000“ wählen, die dann pro Präfix und Unterschaltplan 999 Plätze ermöglicht.

Wird die Bedingung verletzt, z.B dadurch dass ein bestimmter Präfix öfter als 99 oder 999 mal in einem Unterschaltplan vorkommt, so reagiert KiCad flexibel. Bei diesem Präfix wird einfach der Reihenfolge der natürlichen Zahlen entsprechend weitergezählt. Die anderen Präfixe folgen weiter dem Schema. Kann die Zählung des Präfixes mit dem überschrittenen Limit irgendwann wieder mit dem Schema mithalten, weil z.B. einer oder mehrere Subschemata vorkommen, in denen dieser Präfix überhaupt nicht auftaucht, so fällt sie auch wieder in dieses Schema zurück .

In diesem Falle wurde zur Annotation die Methode “Verwende erste freie Nummer bis Schaltplannummer x 100“ gewählt, um später an Hand der Bauteilbezeichnungen auf den Unterschaltplan schließen zu können. Sheet317Regler-1 enthält die 200er Nummern, Sheet317Regler-2 die 300er Nummern und Sheet317Regler-3 die 400er Nummern. Die 100er Nummern beziehen sich auf den Hauptschaltplan.

Zum Inhaltsverzeichnis

### 9.3 Zuweisung von Werten

Dem ersten Unterschaltplan (Sheet317Regler-1) werden Werte zugewiesen, indem man ihn öffnet, und nach Rechtsklick auf die Werte/Values „Wert editieren“ wählt. Das Ergebnis ist in Abbildung 22 zu sehen. Wenn jetzt die entsprechenden Stellen der Unterschaltpläne Sheet317Regler-2 und Sheet317Regler-3 (Abbildung 23 und 24) betrachtet werden, so sieht man, dass überall der gleiche Wert eingetragen ist. Nur die Bauteilreferenznummern unterscheiden sich. Das ist auch nicht weiter verwunderlich, weil alle Schaltpläne auf die gleiche Schaltplandatei „317Regler.sch“ referenzieren.

Zum Inhaltsverzeichnis

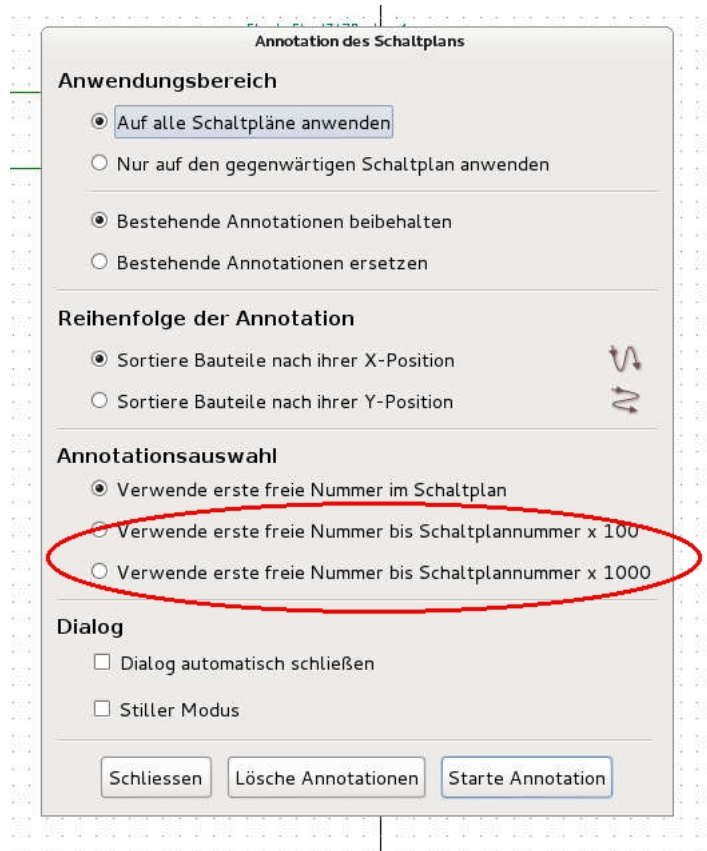


Abbildung 21: Strukturierte Annotationen

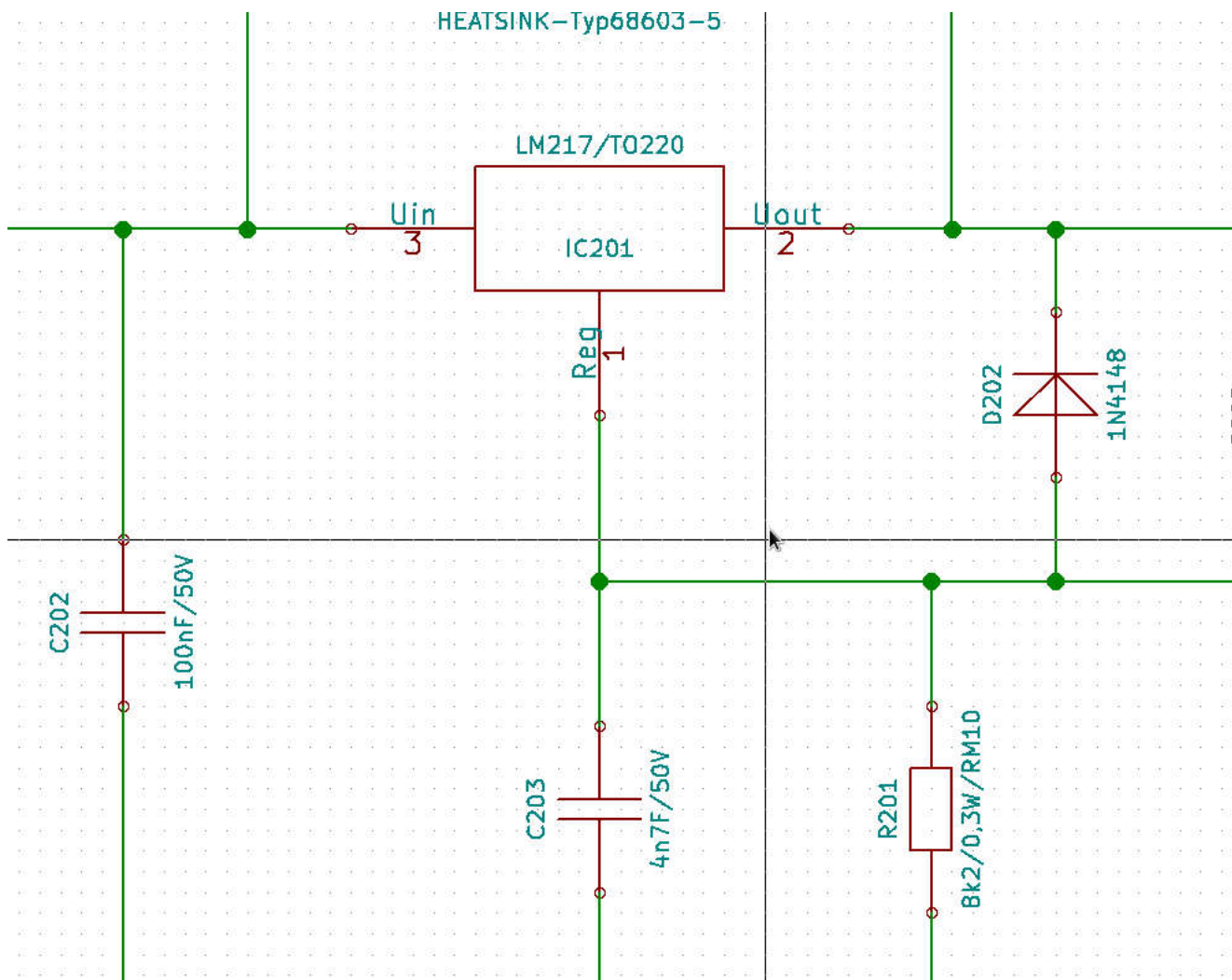


Abbildung 22: Bauteilreferenzen in Sheet317Regler-1. 200er Nummern

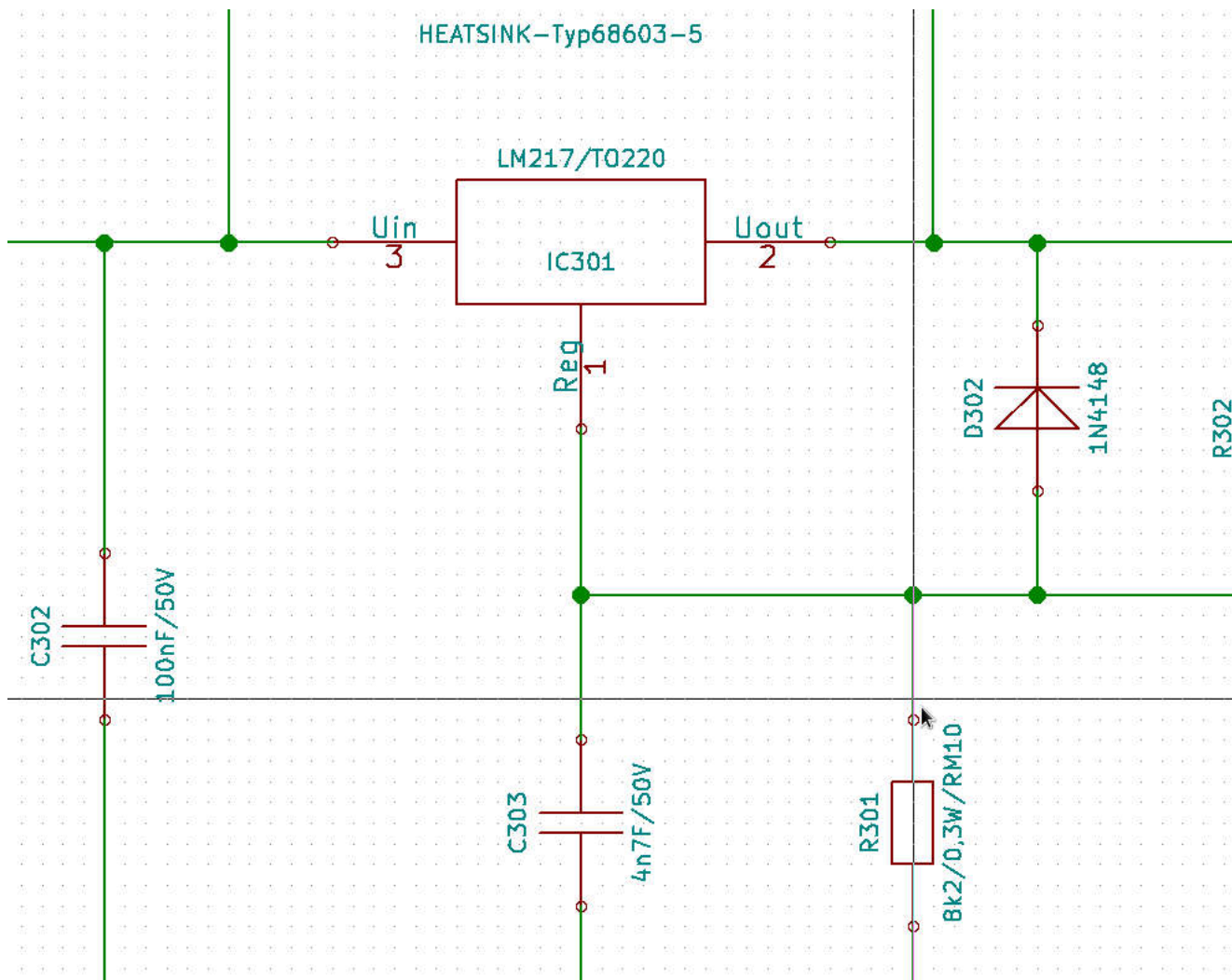


Abbildung 23: Bauteilreferenzen in Sheet317Regler-2. 300er Nummern

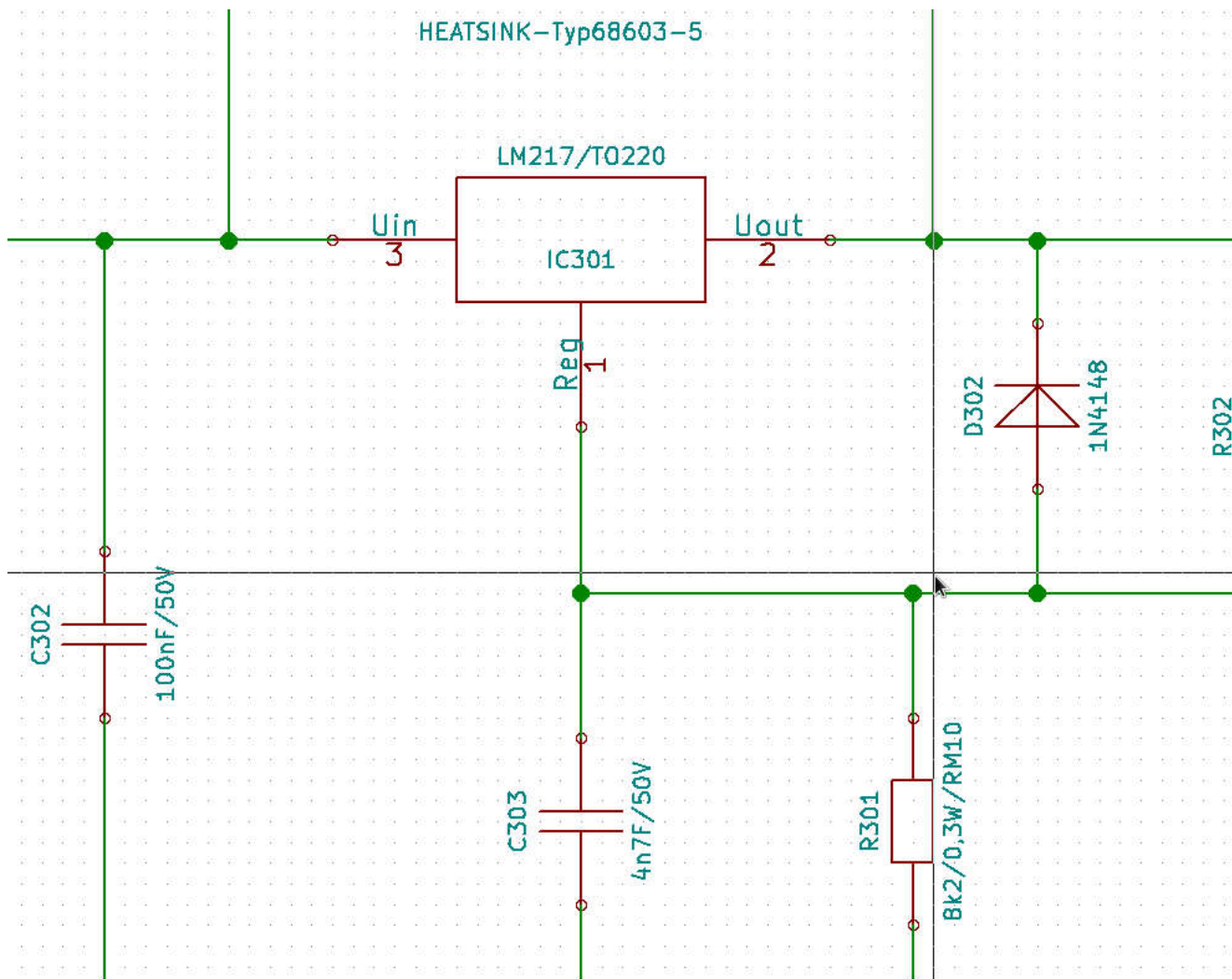


Abbildung 24: Bauteilreferenzen in Sheet317Regler-3. 400er Nummern

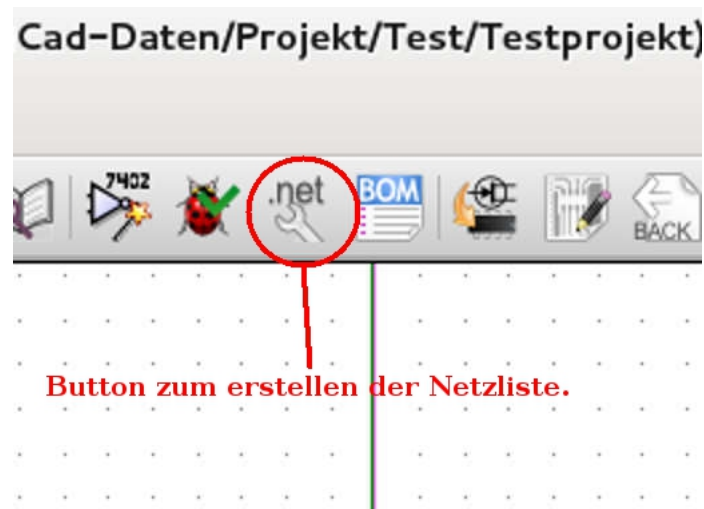


Abbildung 25: Button zum Erstellen einer Netzliste

## 9.4 Netzliste erstellen und Footprints zuweisen

Nun wird eine Netzliste erstellt . Dies geschieht über den dafür vorgesehenen Button nach Abbildung 25. Die dort vorhandenen Defaulteinstellungen werden übernommen (Abbildung 26). Anschließend wird CVpcb geöffnet (Abbildung 27) , und dort den einzelnen Bauteilreferenzen/Footprints zugewiesen. Abbildung 28 zeigt nun, dass trotz der „erzwungenen“ gleichen Werte durchaus unterschiedliche Footprints zugewiesen werden können. Das hat seinen Grund darin, dass die Footprints nicht auf den Schaltplan referenzieren, sondern auf die Netzliste, für die unterschiedliche Bauteilreferenzen durchaus unterschiedliche Positionen sind.

Beispiel: D201 und D301 haben zwar erzwungenermaßen (weil sie sich beide im Schaltplan auf „317Regler.sch“ beziehen) den gleichen Wert (1N4001), aber den unterschiedlichen Bauteilen Footprints können wegen der Entkopplung durch die Netzliste verschiedene Footprints (einmal through hole, und einmal SMD) zugewiesen werden.//

[Zum Inhaltsverzeichnis](#)

## 9.5 Das Ergebnis in PCBnew

Wenn jetzt PCBnew geöffnet wird (Abbildung 29) , muss zuerst die Netzliste eingelesen werden . Dies geschieht über den Button nach Abbildung

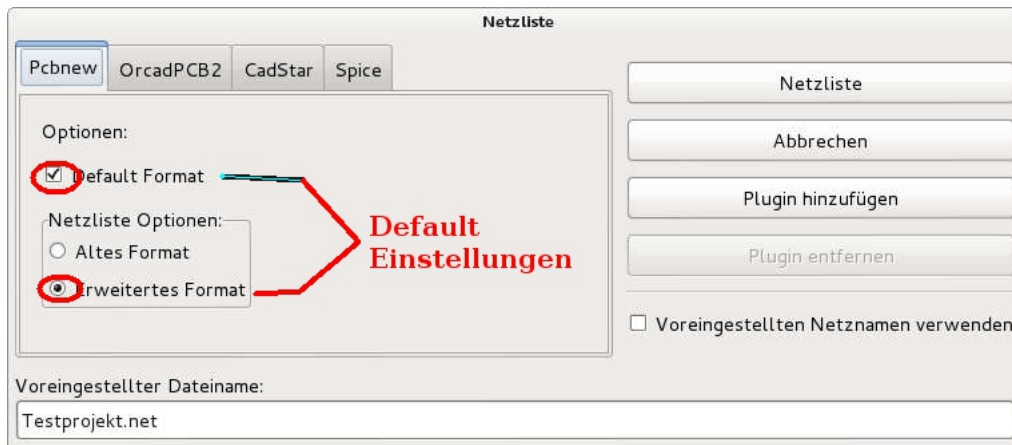


Abbildung 26: Defaulteinstellungen zur Generierung von Netzlisten

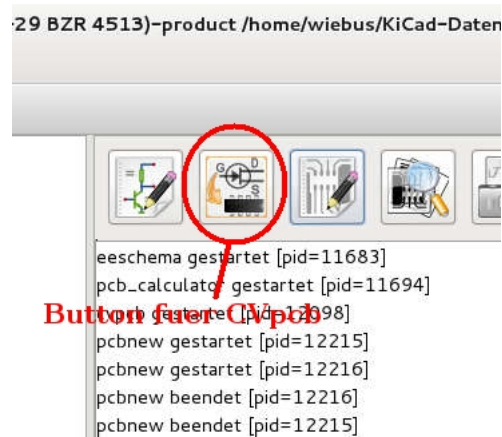


Abbildung 27: Button zum Öffnen von CVpcb



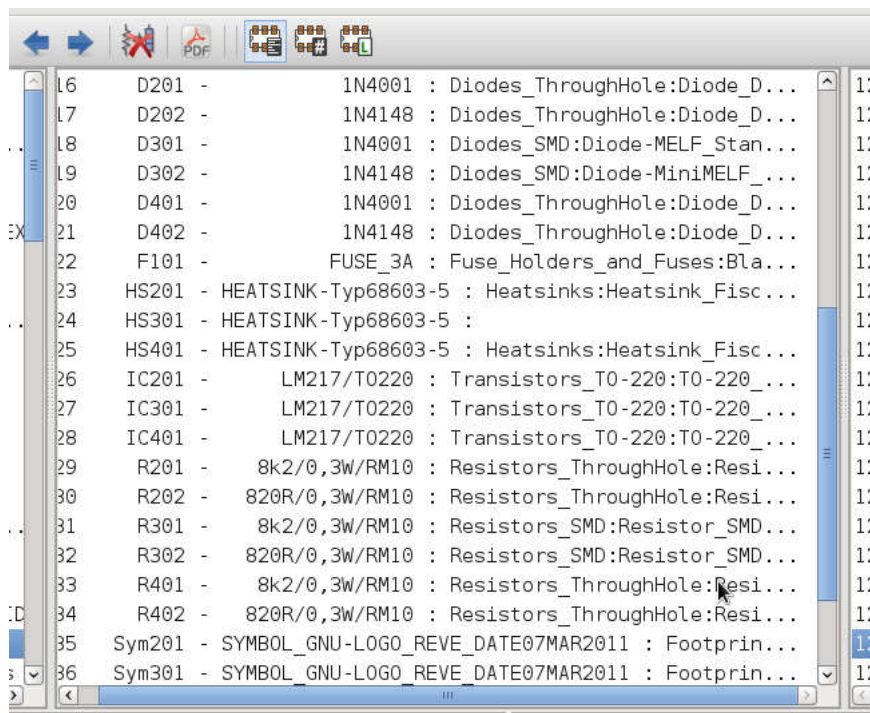


Abbildung 28: Unterschiedliche Footprints wurden in cVpcb eingetragen

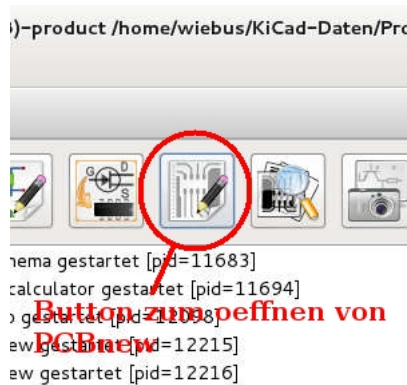


Abbildung 29: Button zum Öffnen von PCBnew

30 . Die Defaulteinstellung zum Einlesen der Netzliste in eine neue, leere Platine zeigt Abbildung 31. Nach dem Einlesen der Netzliste müssen noch die Bauteile auseinandergezogen (plaziert) werden, um die Footprints zu betrachten. Dies geschieht, indem `<t>` eingegeben wird. Es poppt ein Fenster auf, in der die Referenzbezeichnung des Bauteils (z.B. R201) eingegeben werden kann. Nach dem Drücken der `<Enter>` -Taste hängt der Footprint am Mauszeiger. Er kann damit verschoben werden und abschließend mit einem linken Mausklick plaziert werden. In Abbildung 32 sind einige solcher Bauteile gezeigt.

Es zeigt sich, dass dort tatsächlich unterschiedliche Footprints auftauchen. Die Werte sind aber gleich, aus oben angegebenen Gründen. Hier kann aber der Wert verändert werden, weil die Platine über die Netzliste vom ursprünglichen Schaltplan getrennt ist. Dazu macht man einen Rechtsklick auf den Wert eines Bauteils. Hier D301 mit dem Wert „1N4001“ (Abbildung 32 rechts unten). Es poppt ein Fenster auf, in welchem der Wert editiert werden kann, z.B. auf P600. Siehe das Ergebnis in Abbildung 33.

D201 bleibt jetzt fix auf den Wert „1N4001“ notiert. Allerdings ist diese Vorgehensweise nicht sehr sinnvoll, weil letztlich eine Inkonsistenz zwischen Schaltplan und Platine besteht. Darum sollte besser der Wert im Schaltplan geändert werden, und dann über ein erneutes Erstellen und anschließendes Einlesen der Netzliste der geänderte Wert in PCBnew übernommen werden.

Zum Inhaltsverzeichnis



Abbildung 30: Button zum Einlesen der Netzliste

**Netzliste**

<b>Bauteilauswahl</b> <input checked="" type="radio"/> Referenz <input type="radio"/> Zeitstempel	<b>Nicht verbundene Leiterbahnen</b> <input checked="" type="radio"/> Behalten <input type="radio"/> Entfernen	<input type="button" value="Aktuelle Netzliste einlesen"/>
<b>Bauteilnamensquelle</b> <input type="radio"/> Aus Netzliste <input checked="" type="radio"/> Von separater cmp-Datei	<b>Zusätzliche Footprints</b> <input checked="" type="radio"/> Behalten <input type="radio"/> Entfernen	<input type="button" value="Schliessen"/>
<b>Bauteilaustausch</b> <input checked="" type="radio"/> Behalten <input type="radio"/> Ändern	<b>Single Pad Net</b> <input checked="" type="radio"/> Behalten <input type="radio"/> Entfernen	<input type="button" value="Footprints testen"/>

Dry run. Only report changes in message panel  
 Stiller Modus  
 Display all messages

Datei für Netzliste:

Meldungen:

Abbildung 31: Defaulteinstellungen zum Einlesen der Netzliste in eine leere Platinendatei

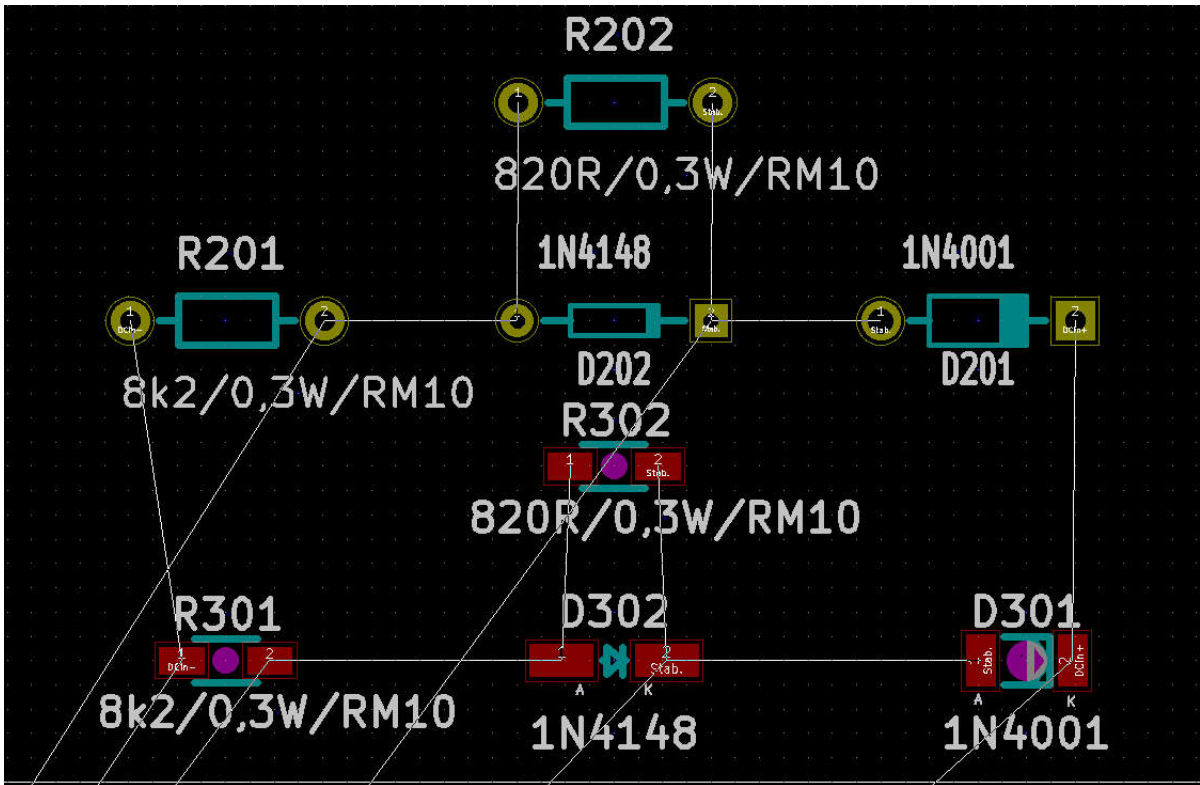


Abbildung 32: Unterschiedliche Footprints bei gleichen Werten/values

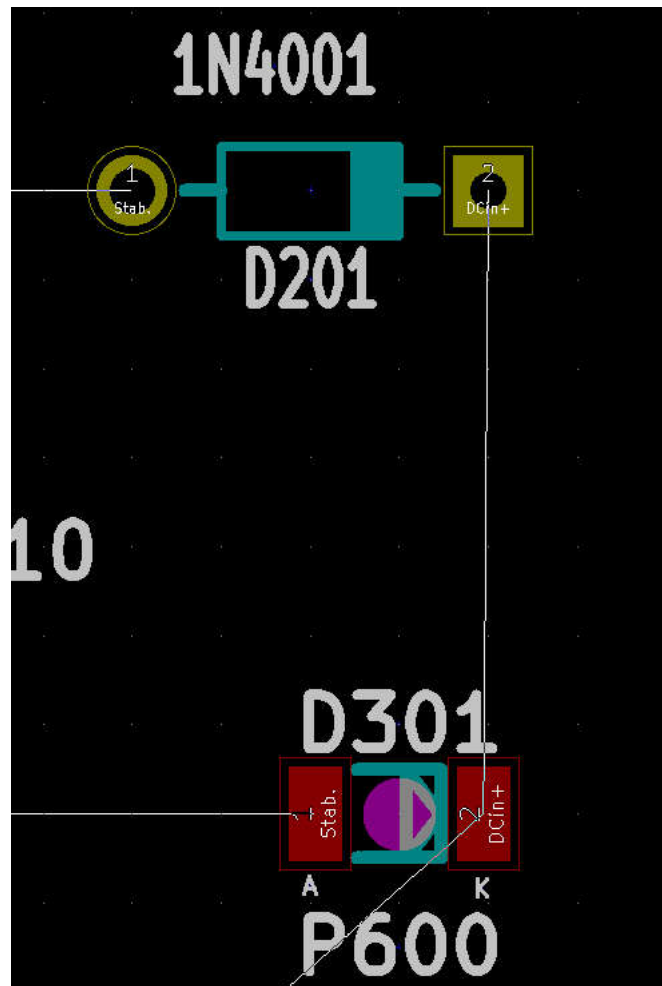


Abbildung 33: Geänderter Wert an D301

## 9.6 Auflösen der Abhängigkeiten zwischen den Subschartplänen

Um also die Werte in einem Subschartplan frei ändern zu können, ohne die Werte der anderen Subschartpläne, die auf die gleiche Subschartplandatei verweisen, mit zu ändern, muss dafür ein separater Subschartplan existieren. Dieses kann durch Kopieren und anschließendes Umbenennen des originalen Subschartplanes „317Regler.sch“ in z.B. „317Regler-2.sch“ erfolgen. Anschließend kann z.B. dem Unterschartplan Sheet317Regler-2 dieser neu erstellte Schaltplan „317Regler-2.sch“ zugewiesen werden, indem man im Hauptschartplan den Subschartplan Sheet317Regler-2 mit einem Rechtsklick anwählt und dann „Schartplan editieren“ wählt. Dort dann entsprechend die Schaltplandatei auf „317Regler-2.sch“ ändern.

Um sicherzustellen, dass die Änderungen auch übernommen werden, speichern Sie das komplette Schaltplanprojekt, schließen Eeschema, und öffnen Sie den Schaltplan erneut.

Öffnen Sie den Unterschartplan Sheet317Regler-2. Nun kann z.B. R301 auf „10k/0,3W/RM10“ und R302 auf „3k3/0,3W/RM10“ verändert werden. Wenn jetzt Sheet317Regler-1 geöffnet wird, sieht man, dass die ehemals damit synchron geänderten Werte von R201 und R202 unverändert sind. Der Subschartplan Sheet317Regler-1 verweist immer noch auf die originale Schaltplandatei „317Regler.sch“, die ja neben der Kopie „317Regler-2.sch“ weiter besteht. Die gegenseitige Abhängigkeit der hierarchischen Schaltpläne ist dadurch verschwunden. Die Subschartpläne können separat editiert werden.

Da Sheet317Regler-3 aber immer noch ebenfalls auf „317Regler.sch“ referenziert, besteht die gegenseitige Abhängigkeit zwischen Sheet317Regler-1 und Sheet317Regler-3 weiterhin. Sie könnte aber mit der gleichen Methode aufgelöst werden.

Zum Inhaltsverzeichnis

## 10 Ausblick - Anwenden der Buildingblock Methodik auch bei PCBnew

Grundätzlich kann diese Methodik in ähnlicher Weise in PCBnew verwendet werden, um mit der „append-board“ Funktion aus vorgefertigten „Buildingblock“-Boards schnell neue Boards zusammenzufügen. Allerdings ist hier mehr Arbeit nötig, da die Annotation aufwändiger zu bearbeiten ist, und durch die „Realitätsnähe“ eines konkreten Platinenlayouts die „Buildingblocks“ oft noch

im Detail modifiziert werden müssen, z.B. durch das Drehen des Buildingblocks bzw. das Verschieben einzelner Komponenten und natürlich das Verbinden mit dem Rest der Schaltung.

Ich hoffe, bald zu dieser Thematik in folgenden Revisionen dieser Anleitung Stellung nehmen zu können.

Zum Inhaltsverzeichnis

## 11 Verbesserungsvorschläge?

Wenn Sie Vorschläge und Ideen für Verbesserungen haben, so bitte ich Sie herzlich, mir diese unter der Emailadresse <mailto:bernd.wiebus@gmx.de> mitzuteilen.

## 12 Impressum

Dieses Dokument ist unter der Creative Commons Lizenz CC-BY-SA 2.0 de veröffentlicht.

Das bedeutet, dass es Ihnen auch für kommerzielle Nutzung erlaubt ist, dieses Dokument kostenlos zu verwenden, wenn Sie es unter gleichen Bedingungen weitergeben und den Autor nennen.

Autor:

Dipl. Ing. Bernd Wiebus

Weezer Str. 5

47589 Udem / Germany

Tel. +49-02825-9399977

Tel. +49-0162-6157950 (mobil)

e-mail: [bernd.wiebus@gmx.de](mailto:bernd.wiebus@gmx.de)

Zum Inhaltsverzeichnis

Zum Inhaltsverzeichnis