

IDF Exporter

19. Oktober 2016

Inhaltsverzeichnis

1 Einführung in den IDFv3 Exporter **2**

2 Spezifizierung des Bauteil Modells was vom Exporter benutzt wird **2**

3 Erstellung einer Datei mit Bauteilkonturen **5**

4 Richtlinien zum Erstellen von Konturen **6**

4.1 Benennung der Bauteile 6

4.2 Kommentare 7

4.3 Geometrie und Einträge der Teilenummer 7

4.4 Pin Ausrichtung und Positionierung 7

4.5 Hinweise zu Abmessungen 8

5 Tools für IDF Bauteilkonturen **9**

5.1 idfcyl 9

5.2 idfrect 10

5.3 dxf2idf 11

6 idf2vrml **12**

Referenz Handbuch

Copyright

Dieses Dokument ist geschützt © 2010-2015 durch deren Beitragende welche nachfolgend aufgeführt sind. Sie können es nach den Bedingungen der GNU General Public License (http://www.gnu.org/licenses/gpl.html), Version 3 oder später, oder der Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/), Version 3.0 oder später verteilen oder verändern .

Alle Markenrechtsnamen in diesem Guide gehören den rechtmäßigen Eigentümern.

Beitragende

Cirilo Bernardo

Übersetzung

Carsten Schönert <c.schoenert@t-online.de>, 2016

Feedback

Bitte alle Bug Reports, Vorschläge oder neue Versionen an folgende Adressen richten:

- KiCad Dokumentation: https://github.com/KiCad/kicad-doc/issues
- KiCad Software: https://bugs.launchpad.net/kicad
- KiCad Software i18n Übersetzung: https://github.com/KiCad/kicad-i18n/issues

Datum der Veröffentlichung und Softwareversion

26.01.2014

1 Einführung in den IDFv3 Exporter

Der IDF Exporter exportiert ein IDFv3 [1] konformes Board (.emn) und Bibliothek (.emp) um mechanische Dimensionen an ein mechanisches CAD Paket zu übertragen. Der Exporter kann aktuell die Abmessungen der Platinenaußenkanten und Ausschnitte in der Platine exportieren als auch alle Pads und Montagelöcher inklusive Langlöcher und Bauteilumrisse. Das sind die grundlegendsten mechanischen Daten die benötigt werden um mit mechanischen Designern interagieren zu können. Alle anderen Einheiten die in der IDFv3 Spezifikation beschrieben sind werden derzeit nicht exportiert.

2 Spezifizierung des Bauteil Modells was vom Exporter benutzt wird

Der IDF Exporter benutzt 3D-Model Datei Attribute welche ursprünglich vom 3D-Betrachter benutzt worden sind. Seit dem 3D-Betrachter, IDF und möglicher Weise auch zukünftige mechanische CAD Exporter generell an verschiedenen Dateiformaten interessiert sind ist es möglich die 3D-Model Dateiattribute zu benutzen um Modelle für multiple Exporter zu spezifizieren.

Innerhalb des Footprint Editors oder in Pcbnew, bearbeiten Sie die Footprint Eigenschaften und klicken auf den Tab für die 3D-Settings (siehe Figur 1), klicken Sie auf *3D Form hinzufügen* und wählen den Filter *IDFv3 Bauteildateien (*.idf)* (siehe Figur 2). Wählen Sie die gewünschte Datei mit den Außenlinien und geben Sie alle nötigen Werte für Offset und Drehung ein. Beachten Sie das nur die Werte für Offset und die Z-Drehung vom IDF Exporter benutzt werden, alle anderen Werte werden ignoriert. Der Offset muss auf Basis der IDF Board Ausgabemaßeinheit (mm oder 1/1000) angegeben werden, das IDF Koordinatensystem ist ein Rechte-Hand Koordinatensystem mit +Z in Richtung des Betrachters, +X ist rechts vom Betrachter und +Y ist oben. Die Drehung muss in Grad angegeben werden, eine positive Drehung ist eine Drehung entgegen dem Uhrzeigersinn wie in der IDFv3 Spezifikation beschrieben. Mehrere Außenlinien können mit einem geeigneten Offset kombiniert werden um einfache Baugruppen wie ein DIP Gehäuse in einer Buchse wiedergeben zu können. [**Fehler:** In Diskussionen wurde entschieden das für den Offset von Z als Einheit Zoll benutzt werden soll, was dann konsistent mit der Einheit für den Offset des VRML Modells ist. Es mag ebenso hilfreich sein nicht die Werte vom Offset (X,Y) zu ignorieren. Diese Handlungsweise die hier vorgeschlagen ist wird sich in der Zukunft irgendwann verändern.]

Wenn für alle gewünschten Bauteile die Modelle festgelegt worden sind wählen Sie in Pcbnew **Datei → Export** und dort **IDFv3**. Eine Dialogbox öffnet sich (siehe Figur 3) welche es erlaubt den Ausgabenamen und die Ausgabeeinheit (mm oder Mils) einzustellen. Die exportierte IDF Datei kann mit der freien mechanischen CAD Software FreeCAD betrachtet werden oder durch das Tool *idf2vrml* in ein VRML umgewandelt werden um dann mit einen geeigneten VRML Betrachter angeschaut werden zu können.

[1] http://www.simplifiedsolutionsinc.com/images/idf_v30_spec.pdf

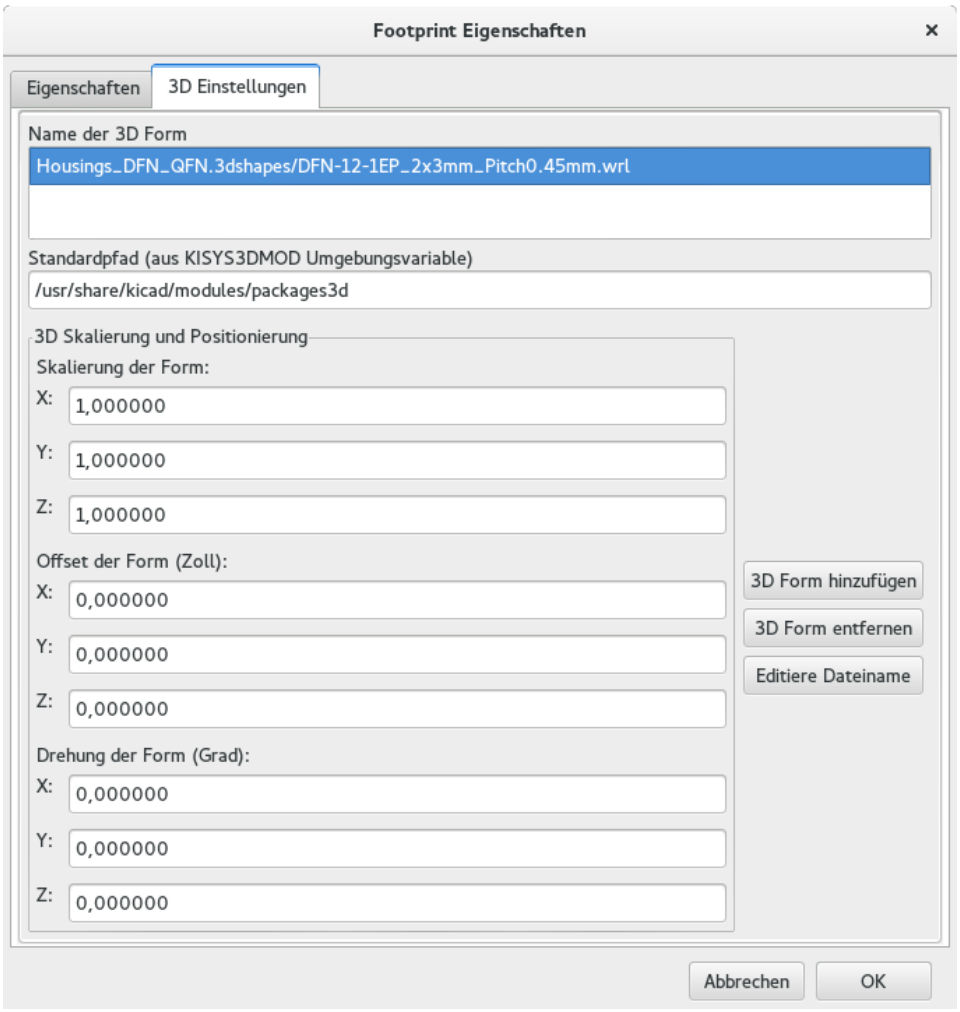


Abbildung 1: Footprint Eigenschaften, 3D Settings

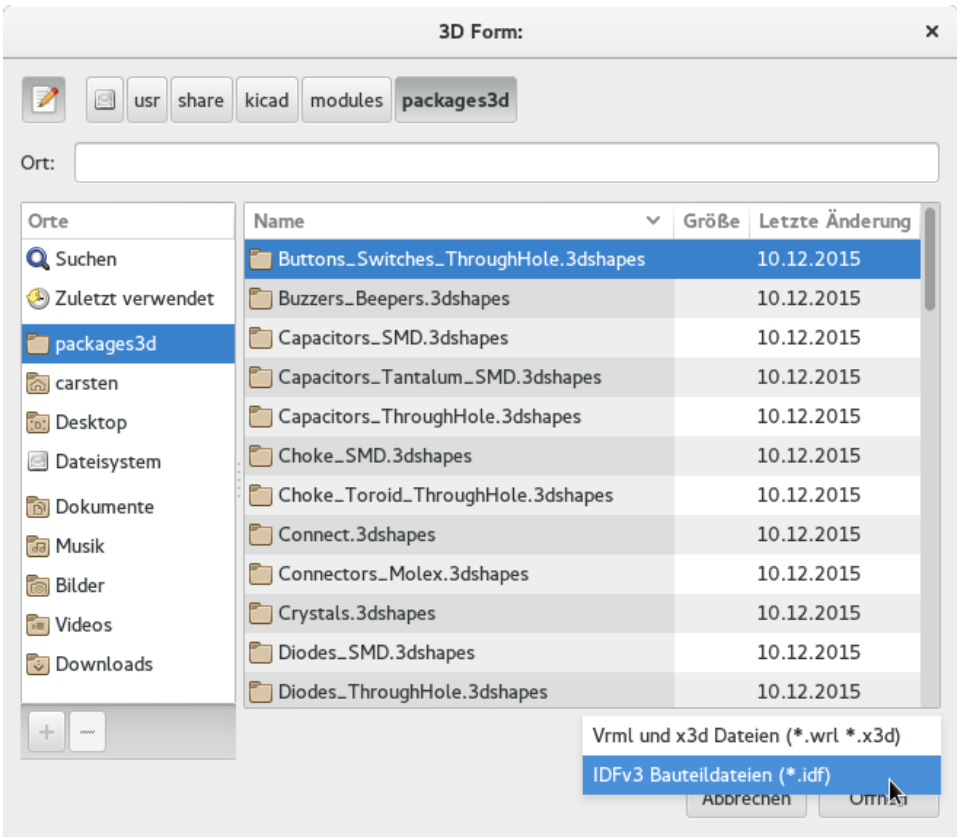


Abbildung 2: Auswahl IDF Bauteilkonturen

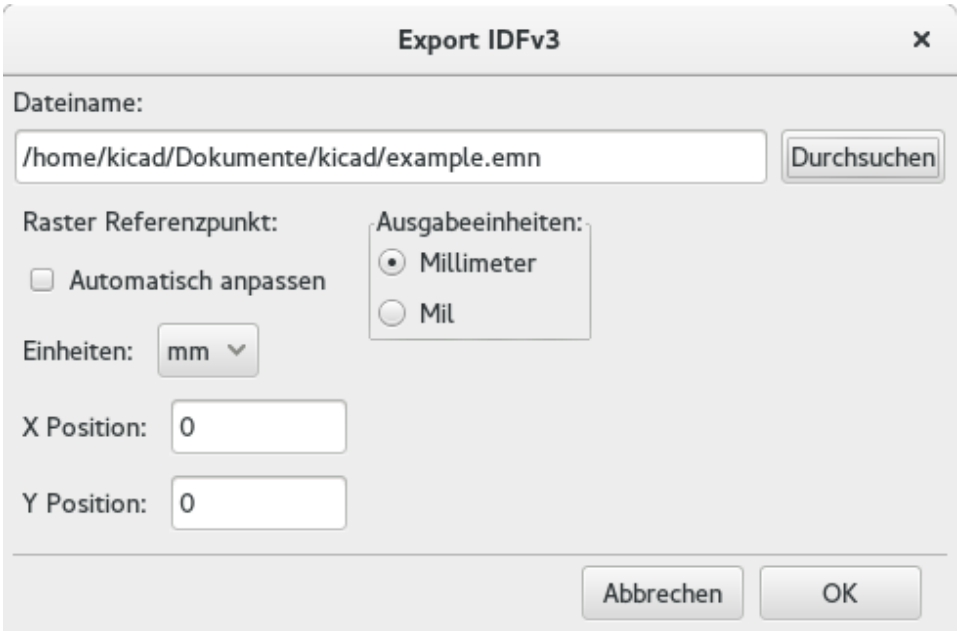


Abbildung 3: IDF Ausgabe Einstellungen

3 Erstellung einer Datei mit Bauteilkonturen

Die Datei mit den Bauteilkonturen (*.idf) enthält eine einzigen Abschnitt .ELECTRICAL oder .MECHANICAL wie in der Dokumenten Spezifikation beschrieben. Diesem Abschnitt kann ein Bereich von zahlreichen Zeilen zur Dokumentation vorangehen, diese Zeilen werden vom Exporter in die Bibliotheksdatei kopiert und können benutzt werden um Metadaten weiterzugeben wie z.B. Referenzen zu den Dokumenten die benutzt wurden sind um die Außenmaße und Dimensionen von Bauteilen zu erfahren.

Die Sektion mit den Bauteilkonturen kann Felder für Zeichenketten, Ganzzahlen oder Gleitkommazahlen enthalten. Eine Zeichenkette ist eine Kombination von Buchstaben die auch Leerzeichen enthalten dürfen, diese müssen aber quotiert werden. Quotierungsmarkierungen erscheinen nicht in der späteren Zeichenkette. Gleitkommazahlen können in Dezimal- oder Exponentialschreibweise geschrieben werden, die dezimale Schreibweise sollte jedoch wegen der besseren Lesbarkeit für Menschen bevorzugt werden. Der Dezimalpunkt muss ein Punkt und darf kein Komma sein. Die IDF Datei darf nur aus 7-Bit ASCII Zeichen bestehen, die Verwendung von 8-Bit Zeichen führt zu einem undefinierten Verhalten.

Eine IDF Datei besteht aus SECTIONS (Abschnitten) welche wiederum aus RECORDS (Datensätzen) besteht, diese enthalten FIELDS (Felder). Die IDF Datei mit den Konturen besteht aus nur einem Abschnitt der dann entweder .ELECTRICAL oder .MECHANICAL benannt werden muss. Ein Datensatz ist eine einzelne Zeile mit Text und kann eine oder mehrere Felder enthalten. Felder sind eine Reihenfolge von Zeichen die durch ein oder mehrere Leerzeichen getrennt sind, welche nicht zwischen Quotierungen liegen. Alle Felder eines Eintrages müssen in einer Zeile liegen, Einträge dürfen nicht über mehrere Zeilen verteilt liegen.

Die Abschnittsmarkierung (.ELECTRICAL oder .MECHANICAL) gilt als der erste Datensatz (Record 1) des Abschnitts. Datensatz 1 muss von Datensatz 2 gefolgt werden, welcher vier Felder hat:

1. Geometrischer Name: Eine Zeichenkette in Kombination mit der Teilenummer, diese muss eine eineindeutige Identifikation der Bauteilkonturen ermöglichen. Für standardisierte Bauteile kann man die Bauform benutzen um den geometrischen Namen zu bilden, zum Beispiel ”SOT-23”. Für die Eindeutigkeit der Bauteile benutzt man die Teilenummer des Herstellers und ergänzt damit den geometrischen Namen.

2. Part Number: Obwohl diese schon in der Benamung Verwendung findet, zum Beispiel BS107, ist es besser diesen Eintrag zu benutzen um das Bauteil zu beschreiben. Wenn der geometrische Name zum Beispiel ”TO-92” ist dann kann man den Eintrag *Teilenummer* benutzen um das Layout und die Orientierung der Pads für dieses spezielle TO-92 Gehäuse zu beschreiben.

3. IDF Einheit: Dies muss entweder MM oder THOU sein, und es gilt nur für die Einheiten die diese einzelne Bauteilkonturen beschreibt.

4. Höhe: Diese Gleitkommazahl gibt die normale Höhe des Bauteils an, die Maßeinheit ist die die unter Feld 3 angegeben ist.

Datensatz 2 muss von einer Anzahl von Einträgen für Datensatz 3 gefolgt sein welcher die Umrisse des Bauteils spezifiziert. Datensatz 3 hat vier Felder:

1. Loop Index: 0 (Konturenpunkte sind im Uhrzeigersinn angegeben) oder 1 (Konturenpunkte sind entgegen dem Uhrzeigersinn angegeben)

2. X-Koordinate: Eine Gleitkommazahl.

3. Y_Koordinate: Eine Gleitkommazahl.

4. Öffnungswinkel: Eine Gleitkommazahl. Wenn der Wert 0 ist dann wird eine gerades Liniensegment vom vorherigen Punkt zu diesem Punkt gezogen. Wenn der Wert 360 ist dann wird der vorherige Punkt als Mittelpunkt eines Kreises benutzt und

dieser Punkt spezifiziert einen Punkt auf dem Kreis. Geben Sie nie einen Wert von -360 an um einen Kreis zu spezifizieren da zumindest ein großes mechanisches CAD Programm damit nicht fehlerfrei zusammen arbeitet. Wenn der Wert negativ ist dann wird der Bogen im Uhrzeigersinn vom vorherigen Punkt aus zu diesem Punkt gezogen, bei einem positiven Wert wird der Bogen entgegengesetzt zum Uhrzeigersinn gezogen.

Nur geschlossener Loop ist erlaubt, und es ist nicht möglich eine Aussparung festlegen. Der letzte angegebene Punkt muss der gleiche sein wie der erste Punkt, wenn der Umriss kein Kreis ist.

Beispiel IDF Datei 1:

```
# a simple cylinder - this could represent an electrolytic capacitor
.ELECTRICAL
    "cylinder" "5mm OD, 5mm height" MM 5
    0 0 0 0
    0 2.5 0 360
.END_ELECTRICAL
```

Beispiel IDF Datei 2:

```
# an upside-down T
# a comment added for the sake of adding comments
.ELECTRICAL
    "Capital T" "5x8x10mm, upside down" MM 10
    0 -0.5 8 0
    0 -0.5 0.5 0
    0 -2.5 0.5 0
    0 -2.5 -0.5 180
    0 2.5 -0.5 0
    0 2.5 0.5 180
    0 0.5 0.5 0
    0 0.5 8 0
    0 -0.5 8 180
.END_ELECTRICAL
```

4 Richtlinien zum Erstellen von Konturen

Wenn Konturen erstellt werden, und besonders wenn man die Arbeit mit anderen teilt, ist ein konsistentes Design und Benennen von Dateien hilfreich um die richtigen Dateien schnell finden zu können damit das Platzieren von Bauteilen ohne große Anstrengungen durchgeführt werden kann.

4.1 Benennung der Bauteile

Versuchen Sie durch den Dateinamen an die Benutzer ein paar Informationen über die Bauteilkonturen mitzugeben damit der Benutzer ein grobe Idee davon bekommt was es für eine Kontur ist. Zum Beispiel können axial verdrahtete zylindrische Bauteile einige Kondensatoren aber auch einige Typen von Widerständen repräsentieren. Daher macht es Sinn einer Kontur Informationen wie horizontale oder vertikale Verdrahtung mitzugeben und ebenso ein paar zusätzliche Informationen wie die relevanten Dimensionen.

Der Durchmesser, die Länge und Abstände sind die wichtigsten. Wenn ein Bauteil eine eher allgemeine Kontur besitzt dann ist die Hersteller Teilenummer und ein Hersteller-Präfix hilfreich um die zugehörige Klasse des Bauteils zu kennzeichnen.

4.2 Kommentare

Benutzen Sie Kommentare um den Benutzern weitere Informationen über die Bauteilkontur mitzugeben, zum Beispiel einen Hinweis auf die Referenzen die benutzt worden sind um die Dimensionieren in Erfahrung zu bringen.

4.3 Geometrie und Einträge der Teilenummer

Bedenken Sie sorgfältig die Werte die Sie in die Felder für die Geometrie und die Teilenummer eintragen. Zusammen bilden diese Zeichenketten eine eindeutige Kennung für das MCAD-System. Die Werte der Einträge haben idealerweise eine Bedeutung für einen Benutzer, aber dies ist nicht unbedingt notwendig. Diese Werte werden primär vom MCAD-System als Unique-ID benutzt. Im Idealfall sind die gewählten einzelnen Werte in diesen Feldern auch innerhalb einer größeren Zusammenstellung von Konturen eindeutig, eine gute Auswahl wird zu weniger Konflikten besonders in komplexen Boards führen.

4.4 Pin Ausrichtung und Positionierung

Für durchkontaktierte Bauteile gibt es keine weit akzeptierten Schemas für die Ermittlung der Pin Ausrichtung und Bauteilmittelpunkte in 3D-Modellen. Wenn es nur 2 Pins an einem Bauteil gibt müssen, aus Gründen der Konsistenz, diese in einer horizontalen Anordnung (siehe Figur 4) entlang der X-Achse liegen und bei 3 Pins versuchen Sie zwei Pins in einer horizontalen Anordnung auf der X-Achse anzuordnen. Bei Bauteilen mit einer Polarität wie Tantal oder Elektrolyt Kondensatoren muss am Pin 1 der positive Pol liegen, Dioden müssen die Kathode auf Pin 1 liegen haben. Dies ist nötig aus Gründen der Kompatibilität zu Schaltplansymbolen die eine Ausrichtung für SMT Bauteile besitzen, beachten Sie aber, dass viele bestehende KiCad Schaltplansymbole und Footprints die Anode an Pin 1 haben.

Anmerkung

In den letzten Versionen der KiCad Footprints auf GitHub ist in den THT und SMT Bauteilen die Anode jetzt auf Pin 2.

Bei DIP Gehäusen muss der Mittelpunkt der Kontur in der Mitte des Rechtecks liegen das von den einzelnen Pins begrenzt ist und Pin 1 liegt vorzugsweise oben links. Dies wird eine gewisse Konsistenz mit der standardisierten Ausrichtung von SMT-Komponenten erhalten, jedoch werden in den meisten existierenden KiCad Footprints und VRML-Modellen solche Modelle relativ um -90 Grad gedreht. Bei Bauteilen wie ein horizontal, radial verdrahteten Kondensator oder ein TO-220 Gehäuse sollten die Anschlussbeine längs auf der X-Achse liegen mit dem Gehäuse nach oben ausgerichtet (siehe Figur 4). Unpolarisierte vertikal, axial verdrahtete Bauteile müssen die Anschlüsse rechts haben, polarisierte vertikal, axial bedrahtetete Bauteilen können die Anschlüsse auf beiden Seiten haben, je nachdem ob Pin 1 ist am unteren Ende (Draht rechts) oder am oberen Ende (Draht links) liegt.

Anmerkung

In den aktuellen Versionen von KiCad sind die Footprint Module der THT Bauteile entlang der Y-Achse anstatt der X-Achse ausgerichtet und Pin 1 vom Bauteil ist nicht in der Mitte des Bauteils. Ausrichtung und Position der Bauteilkontur müssen den spezifischen Footprints entsprechen, dies wird eine sonst nötige negative Drehung der IDF Bauteilkontur vermeiden. Da der IDF Exporter aktuell die (X,Y) Offsetwerte ignoriert es ist wichtig, dass Sie den richtigen Ursprung in der IDF Bauteilkontur verwenden.

Für SMT Bauteile sind die Ausrichtung, Bauteilmitte und die Beschreibung der Kontur durch unterschiedliche Standards definiert. Benutzen Sie den passenden Standard der zu Ihrer Arbeit passt. Bedenken Sie auch, dass viel Bauteile nicht konform zu einem Standard sind, in so einem Fall ist es vermutlich das Beste das Bauteil durch die Teilenummer des Herstellers in der Datei mit den Konturen zu identifizieren. Im Allgemeinen ist eine SMT Kontur ein Rechteck welches das Bauteilgehäuse einschließlich der Anschlusspins umfasst und das Gehäuse so ausgerichtet ist, dass der Pin 1 ist so nah wie möglich an der oberen linken Ecke und der oberen linken Ecke ist, in der Regel visuell durch eine Vertiefung oder Referenz hervorgehoben.

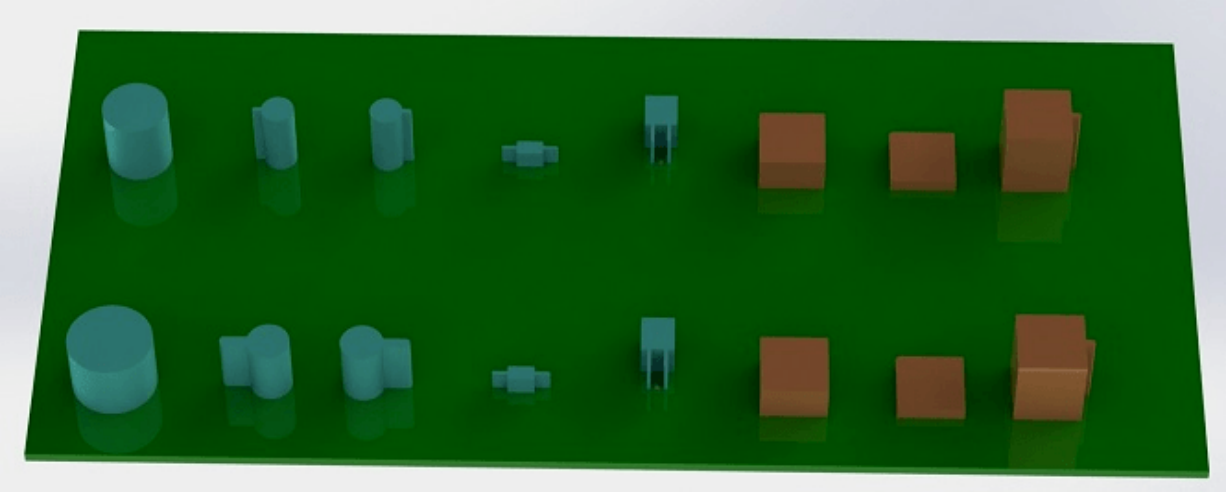


Abbildung 4: Beispiele für Konturen erzeugt von den Programmen idfcyl und idfrect, gerendert durch SolidWorks

Von links nach rechts sind (a) ein vertikal, radial verdrahteter Zylinder, (b) ein vertikal, axial verdrahteter Zylinder mit dem Anschlussdraht auf der linken Seite, (c) ein vertikal, axial verdrahteter Zylinder mit dem Anschlussdraht auf der rechten Seite, (d) ein horizontal, axial verdrahteter Zylinder, (e) ein horizontal, radial verdrahteter Zylinder, (f) eine glatte quadratische Kontur, (g) eine quadratische Kontur mit einer Abschräge, (h) eine quadratische Kontur mit axialer Verkabelung auf der rechten Seite zu sehen. Die obere Reihe wurden mit der Maßeinheit mm erstellt, die untere Reihe benutzt Zoll als Maßeinheit.

4.5 Hinweise zu Abmessungen

Der Zweck hinter den modellierten Konturen besteht darin dem Designer von mechanischen Formen einen Überblick über den benötigten Platzbedarf der jeweiligen Bauteile zu geben. Typischer Weise wird der Designer diese abstrakten Formen mit detaillierteren Modellen ersetzen, zum Beispiel wenn er sicherstellen muss, dass eine um 90° nach rechts abstehende LED später in eine Gehäuse Aussparung passen muss. In den meisten Fällen ist eine hohe Detailtreue nicht nötig, jedoch ist es eine gute Praxis den größtmöglichen Anteil an mechanischen Informationen mitzuliefern. In manchen Situationen will der Benutzer seine Platine in einem Gehäuße mit sehr kleinen Abmessungen einbauen, zum Beispiel in einen mobilen Music Player. Wenn die Modelle der jeweiligen Konturen der Bauteile ausreichend gut sind muss der Designer des Gehäuses nur wenige Bauteile während des Designprozesses anpassen. Sind die Konturen allerdings recht ungenau wird der Designer für das Gehäuse einen erheblichen Zeitaufwand damit betreiben einzelne Bauteile zu kontrollieren um sicher stellen zu können, dass alle Komponenten in das geplante Gehäuse passen. Daher gilt, je besser die Basisinformationen umso besser auch das direkte Ergebnis. Je genauer die Informationen im Vorfeld sind desto präziser auch auch die entstehenden Modelle aus diesen Informationen.

5 Tools für IDF Bauteilkonturen

Es gibt eine Anzahl an Kommandozeilen Tools ,die helfen IDF Bauteilkonturen zu erstellen. Diese Tools sind:

1. **idfcyl:** Erstellt eine Kontur von einem Zylinder in vertikaler oder horizontaler Ausrichtung mit axial oder radialen geführten Anschlussdrähten.
2. **idfrect:** Erstellt eine Kontur von einem Rechteck welches axial geführte Anschlussdrähte besitzen kann oder eine Abschrägung.
3. **dxf2idf:** Konvertiert eine DXF Zeichnung in eine IDF Bauteilkontur.

5.1 idfcyl

Wird **idfcyl** ohne weitere Argumente aufgerufen werden Benutzungshinweise für die Parameter ausgegeben:

```
idfcyl: This program generates an outline for a cylindrical component.
    The cylinder may be horizontal or vertical.
    A horizontal cylinder may have wires at one or both ends.
    A vertical cylinder may have at most one wire which may be
    placed on the left or right side.

Input:
    Unit: mm, in (millimeters or inches)
    Orientation: V (vertical)
    Lead type: X, R (axial, radial)
    Diameter of body
    Length of body
    Board offset
    *   Wire diameter
    *   Pitch
    **  Wire side: L, R (left, right)
    *** Lead length
    File name (must end in *.idf)

    NOTES:
        *   only required for horizontal orientation or
            vertical orientation with axial leads

        **  only required for vertical orientation with axial leads

        *** only required for horizontal orientation with radial leads
```

Die Hinweise werden unterdrückt wenn irgendein Argument übergeben wird. Der Benutzer kann manuell die Argumente an das Tool übergeben oder erstellt ein Skript um die Konturen zu erstellen. Das folgende Skript erstellt eine einzelne axial verdrahtete Kontur mit den Anschlussdrähten auf der rechten Seite:

```
#!/bin/bash
# Generate a cylindrical IDF outline for test purposes
# vertical 5mm cylinder,  nominal length 8mm + 3mm board offset,
# axial wire on right,  0.8mm wire dia., 3.5mm pitch
idfcyl - 1 > /dev/null <<  _EOF
mm
v
x
5
8
3
0.8
3.5
r
cylvmm_1R_D5_L8_Z3_WD0.8_P3.5.idf
_EOF
```

5.2 idfrect

Wird **idfrect** ohne weitere Argumente aufgerufen werden Benutzungshinweise für die Parameter ausgegeben:

```
idfrect: This program generates an outline for a rectangular component.
    The component may have a single lead (axial) or a chamfer on the
    upper left corner.
Input:
    Unit: mm, in (millimeters or inches)
    Width:
    Length:
    Height:
    Chamfer: length of the 45 deg. chamfer
    *  Leaded: Y,N (lead is always to the right)
    ** Wire diameter
    ** Pitch
    File name (must end in *.idf)

    NOTES:
        *   only required if chamfer = 0

        **  only required for leaded components
```

Die Hinweise werden unterdrückt wenn irgendein Argument übergeben wird. Der Benutzer kann manuell die Argumente an das Tool übergeben oder erstellt ein Skript um die Konturen zu erstellen. Das folgende Skript erstellt eine einzelne axial verdrahtete rechteckige Kontur mit Abfasung:

```
#!/bin/bash
# Generate various rectangular IDF outlines for test purposes
# 10x10, 1mm chamfer, 2mm height
```

```
idfrect - 1 > /dev/null <<  _EOF
mm
10
10
2
1
rectMM_10x10x2_C0.5.idf
_EOF
# 10x10x12,  0.8mm lead on 6mm pitch
idfrect - 1 > /dev/null <<  _EOF
mm
10
10
12
0
Y
0.8
6
rectLMM_10x10x12_D0.8_P6.0.idf
_EOF
```

5.3 dxf2idf

Die DXF Datei die benutzt wird um die Bauteilkontur anzugeben die umgewandelt werden soll kann zum Beispiel mit der freien Software LibreCAD bearbeitet werden um die höchste Kompatibilität zu erreichen. Wird **dxf2idf** ohne Angabe von Argumenten aufgerufen so erscheinen die Benutzungshinweise und Hilfe zu den Argumenten wie schon bei den anderen Tools:

```
dxf2idf: this program takes line, arc, and circle segments
    from a DXF file and creates an IDF component outline file.

Input:
    DXF filename: the input file, must end in '.dxf'
    Units: mm, in (millimeters or inches)
    Geometry Name: string, as per IDF version 3.0 specification
    Part Name: as per IDF version 3.0 specification of Part Number
    Height: extruded height of the outline
    Comments: all non-empty lines are comments to be added to
        the IDF file. An empty line signifies the end of
        the comment block.
    File name: output filename, must end in '.idf'
```

Die Hinweise werden unterdrückt wenn irgendein Argument übergeben wird. Der Benutzer kann manuell die Argumente an das Tool übergeben oder erstellt ein Skript um die Konturen zu erstellen. Das folgende Skript erstellt eine 5mm hohe Kontur aus der Datei *test.dxf*:

```
#!/bin/bash
# Generate an IDF outlines from a DXF file
```

```
dxf2idf - 1 > /dev/null << _EOF
test.dxf
mm
DXF TEST GEOMETRY
DXF TEST PART
5
This is an IDF test file produced from the outline 'test.dxf'
This is a second IDF comment to demonstrate multiple comments

test_dxf2idf.idf
_EOF
```

6 idf2vrml

Das Tool *idf2vrml* liest einen Satz IDF Board (.emn) und IDF Bauteildatei (.emp) ein uns erstellt daraus eine VRML Datei die dann mit einem VRML-Betrachter angeschaut werden kann. Dies ist ein hilfreiches Feature für die Visualisierung der Platinenbestückungen für die Fälle wo der Benutzer keinen Zugang zu MCAD-Software hat. Der Aufruf von *idf2vrml* ohne Argumente zeigt wiederum Benutzungshinweise und die Hilfe zu den Argumenten an:

```
>./idf2vrml
Usage: idf2vrml -f input_file.emn -s scale_factor {-k} {-d} {-z} {-m}
flags:
   -k: produce KiCad-friendly VRML output; default is compact VRML
   -d: suppress substitution of default outlines
   -z: suppress rendering of zero-height outlines
   -m: print object mapping to stdout for debugging purposes
example to produce a model for use by KiCad: idf2vrml -f input.emn -s 0.3937008 -k
>
```

[**Fehler:**] Das *idf2vrml* Tool rendert aktuell die **OTHER_OUTLINE** Einheiten in einer emn Datei nicht richtig wenn diese Einheiten auf der Rückseite der Platine liegen. Sie werden aber diesen Fehler nicht vorfinden wenn Sie Dateien benutzen die von KiCad exportiert worden sind, es gibt in KiCad keine Möglichkeit derartige Einheiten so festzulegen. Im Wesentlichen wird dieser Fehler nur sehr selten auftreten, und zwar nur dann wenn Sie eine .emn Datei von einer dritten Person benutzen die eine solche Einheit auf der Rückseite einer Platine definiert hat.