

# IDF Eksporter

## Table of Contents

|   |    |
|---|----|
| Wprowadzenie do eksportera IDfv3 .....                      | 3  |
| Określanie modeli komponentów użytych przez eksporter ..... | 3  |
| Tworzenie pliku z zarysami komponentów .....                | 5  |
| Wskazówki przy tworzeniu konturów .....                     | 7  |
| Nazwy pakietów .....  | 7  |
| Komentarze .....  | 7  |
| Wpisy Geometry i Part Number .....                          | 8  |
| Orientacja wyprowadzeń oraz ich ustawianie .....            | 8  |
| Wskazówki dotyczące wymiarów .....                          | 9  |
| Narzędzia dla Kształtów IDF .....                           | 9  |
| idfcyl .....  | 10 |
| idfrect .....   | 11 |
| dxf2idf .....   | 12 |
| idf2vrml .....  | 13 |

## *Podręcznik użytkownika*

### **Prawa autorskie**

Copyright © 2014-2015. Ten dokument jest chroniony prawem autorskim. Lista autorów znajduje się poniżej. Możesz go rozpowszechniać oraz modyfikować na zasadach określonych w General Public License (<http://www.gnu.org/licenses/gpl.html>), wersja 3 lub późniejsza, albo określonych w Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), wersja 3.0 lub późniejsza.

Wszystkie znaki towarowe użyte w tym dokumencie należą do ich właścicieli.

### **Współtwórcy**

Cirilo Bernardo

### **Tłumaczenie**

Kerusey Karyu <[keruseykaryu@o2.pl](mailto:keruseykaryu@o2.pl)>, 2014-2015.

### **Kontakt**

Wszelkie zauważone błędy, sugestie lub nowe wersje dotyczące tego dokumentu prosimy kierować do:

- W sprawie dokumentacji: <https://gitlab.com/kicad/services/kicad-doc/issues>
- W sprawie oprogramowania: <https://gitlab.com/kicad/code/kicad/issues>
- W sprawie tłumaczeń interfejsu użytkownika (i18n): <https://gitlab.com/kicad/code/kicad-i18n/issues>

### **Data publikacji i wersja oprogramowania**

Opublikowano 26 stycznia 2014.

# Wprowadzenie do eksportera IDFv3

IDF eksporter eksportuje płytki do formatu pliku (.emn) oraz biblioteki (.emp) zgodnych z IDFv3 [1] do pliku wymiary pomiędzy narzędziami CAD. Eksporter obecnie eksportuje obrysy płyty oraz wycięcia, wszystkie pola lutownicze i otwory w nich, w tym także podłużne oraz kształty komponentów; jest to najbardziej podstawowy zestaw danych mechanicznych wymaganych przy współpracy z projektantami mechanicznymi. Wszystkie inne podmioty opisane w specyfikacji IDFv3 nie są obecnie zaimplementowane.

## Określanie modeli komponentów użytych przez eksporter

IDF eksporter korzysta z atrybutów pliku modelu 3D, który pierwotnie używany jest przez przeglądarkę 3D. Ponieważ przeglądarka 3D, IDF i ewentualnie przyszłe eksportery mechanicznego CAD są skupione na formatach plików różnego rodzaju, można użyć atrybutów modelu 3D, by określić modele dla wielu różnych eksporterów.

Z programu Edytor Footprintów lub Pcbnew, należy dokonać edycji parametrów footprintu oraz kliknąć na zakładkę Ustawienia 3D (zobacz [Rysunek 1](#)), kliknąć na "Dodaj kształt 3D", oraz wybrać filtr "Pliki IDFv3 (\*.idf)" (zobacz [Rysunek 2](#)). Wybrać żądany plik z zarysem oraz wprowadzić wszystkie niezbędne wartości dla przesunięcia i rotacji modelu. Należy pamiętać, że tylko przesunięcie oraz obrót w osi Z jest używany przez eksporter IDF; wszystkie inne wartości są ignorowane. Przesunięcie musi być określone w jednostkach jakich używa IDF (mm lub thou) oraz zorientowane w ten sam sposób jaki stosuje IDF, jest to reguła prawej dłoni z osią Z zwiększającą się w kierunku oglądającego, osią X zwiększającą się w prawo, a Y w górę. Rotacja musi być w stopniach, gdzie dodatnie wartości oznaczają obrót przeciwnie do ruchu wskazówek zegara, jak zostało to określone w specyfikacji IDFv3. Skomplikowane kształty mogą zostać utworzone przez połączenie odpowiednich przesunięć by tworzyć proste zespoły takie jak np. obudowa DIP umieszczona w podstawie. [**BŁĄD:** W trakcie dyskusji zdecydowane zostało, że jednostką w osi Z muszą być cale, co jest zgodne z jednostkami przesunięć w modelach VRML. Zignorowanie przesunięć (X, Y) może także być użyteczne. Zachowanie takie może w późniejszych wersjach zostać zmienione.]

Jak tylko modele zostaną określone dla wszystkich pożądanых elementów, z poziomu Pcbnew wybierz menu **Plik**, a następnie **Eksportuj**, i finalnie **Eksportuj do IDFv3**. Pojawi się okno dialogowe (patrz [Rysunek 3](#)), które pozwala na ustawienie pliku wyjściowego IDF oraz jednostek miary (mm lub mils). Wyeksportowane pliki IDF mogą być przeglądane w oprogramowaniu typu CAD, na przykład darmowy [FreeCAD](#) lub przekształcane do postaci VRML z użyciem narzędzia *idf2vrm* oraz przeglądane przez dowolną przeglądarkę VRML.

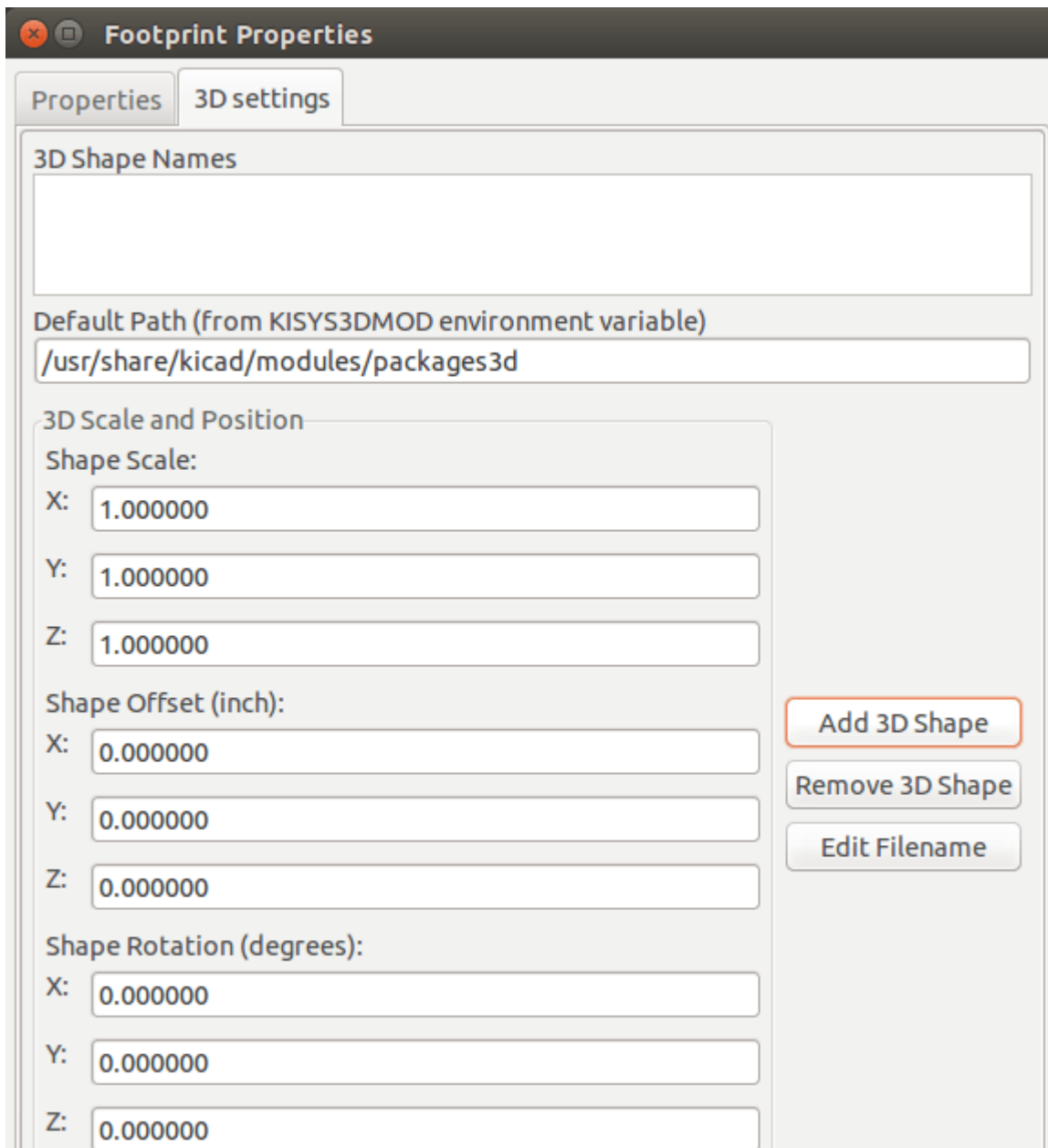


Figure 1. Właściwości modułów, ustawienia 3D

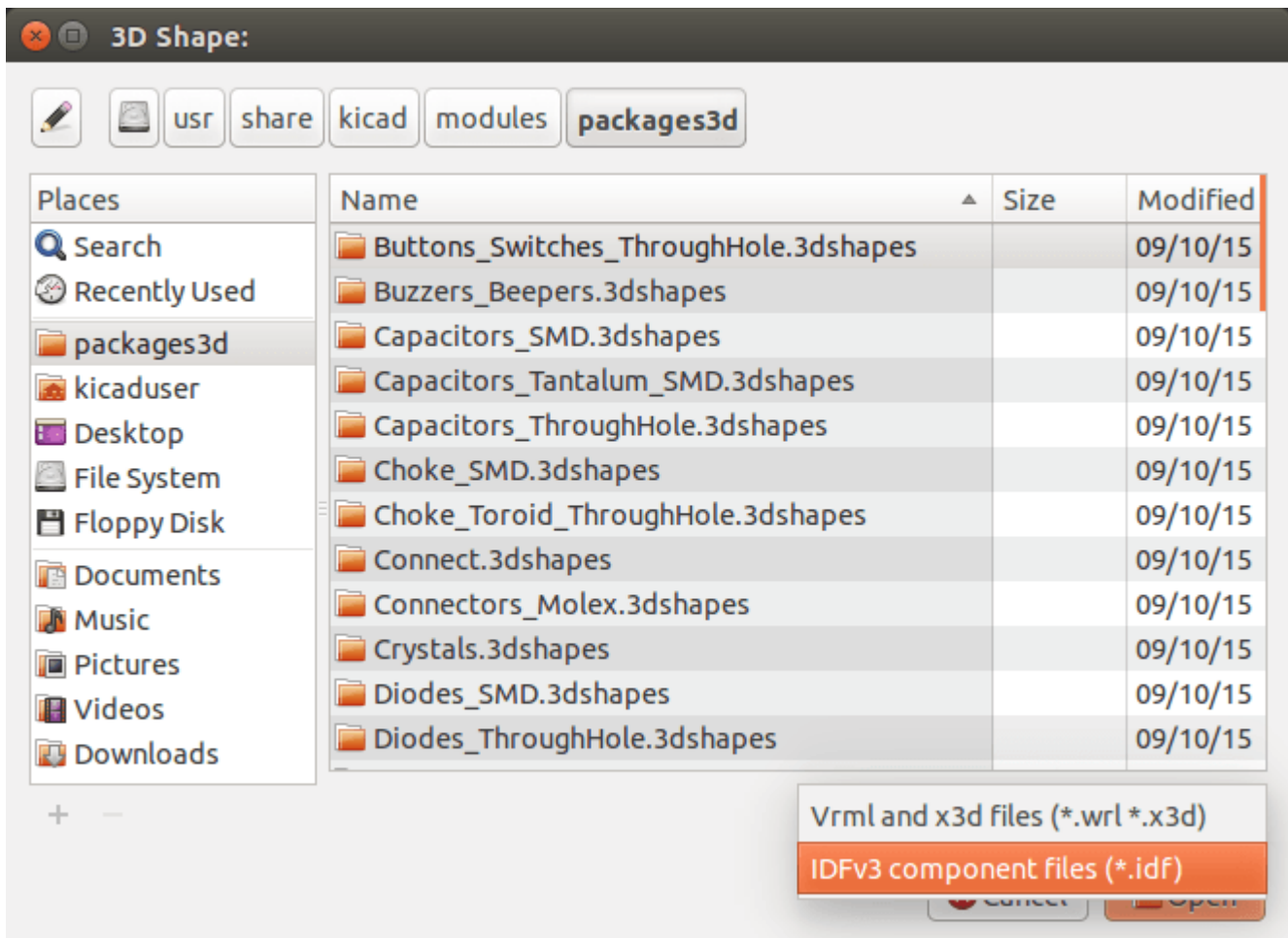


Figure 2. Wybór zarysów komponentów IDF

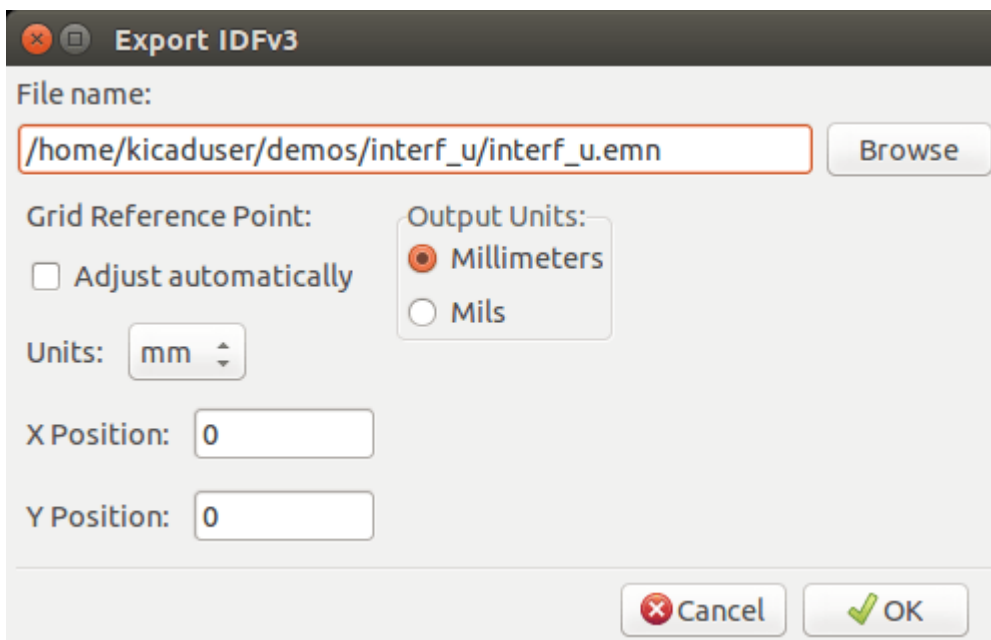


Figure 3. Ustawienia wyjściowe IDF

## Tworzenie pliku z zarysami komponentów

Plik z zarysem komponentu (\*.idf) zawiera pojedynczą sekcję .ELECTRICAL lub .MECHANICAL jak opisano w specyfikacji. Sekcje mogą być poprzedzone dowolną ilością linii komentarza; linie komentarza są

kopiuwane przez eksporter do pliku biblioteki i mogą zostać użyte do śledzenia danych meta, takich jak odnośniki do dokumentacji używane do określania zarysu oraz rozmiaru komponentu.

Sekcja zarysu komponentu zawiera pola, które są ciągami znaków, liczbami całkowitymi lub ułamkowymi. Ciąg jest kombinacją znaków, które mogą zawierać spacje; jeśli ciąg zawiera spacje to musi być ujęty w cudzysłowy. Cudzysłowy nie mogą pojawiać się wewnątrz ciągu. Liczby ułamkowe mogą być reprezentowane za pomocą notacji dziesiętnej lub wykładniczej, ale notacja dziesiętna jest jednak bardziej czytelna. Punktem dziesiętnym musi być kropka, a nie przecinek. Plik IDF musi składać się tylko z 7-bitowych znaków ASCII; zastosowanie 8-bitowych znaków może doprowadzić do zachowań niemożliwych do zdefiniowania.

Plik IDF składa się z *SEKCJI*, które składają się z *REKORDÓW*, które składają się z *PÓL*. Dla plików konturów IDF może istnieć tylko jeden typ sekcji i musi być jednym z *.ELECTRICAL* lub *.MECHANICAL*. Rekord jest pojedynczą linią tekstu i może zawierać jedno lub więcej pól. Pola to sekwencje znaków rozdzielonych przez jedną lub kilka spacji, które nie pojawiają się w cudzysłowach. Wszystkie pola rekordu muszą znajdować się w jednej linii; rekordy nie mogą obejmować wielu wierszy.

Sekcja nagłówka (*.ELECTRICAL* lub *.MECHANICAL*) jest uważana za pierwszy rekord (Record 1) sekcji. Rekord 1 musi występować razem z rekordem 2, który posiada cztery pola:

1. Nazwa Geometrii: ciąg znaków, który razem z numerem części musi formować unikalny identyfikator kształtu komponentu. Dla standardowych obudów, nazwa obudowy stanowi dobrą wartość dla nazwy geometrii, na przykład "SOT-23". Dla unikalnych obudów numer części według producenta również stanowi dobry wybór dla nazwy geometrii.
2. Numer części: choć oczywiście przeznaczony na numer części, np. "BS107", to lepiej jest użyć tego ciągu aby móc opisać pakiet. Na przykład jeśli nazwa geometrii to "TO-92", pole Numer części może być użyte do opisywania układu pól lutowniczych lub orientację tego konkretnego pliku konturu obudowy TO-92.
3. Jednostki IDF: musi zawierać jedną z dwóch wartości MM lub THOU i ma zastosowanie wyłącznie do jednostek opisujących ten pojedynczy składnik konturu.
4. Wysokość: jest to liczba rzeczywista reprezentująca wysokość nominalnego składnika przy użyciu jednostek określonych w polu 3.

Po rekordzie 2 musi wystąpić kilka rekordów 3, które określają kontur elementu. Rekord 3 składa się z czterech pól:

1. Indeks pętli: 0 (punkty konturu są określane przeciwnie do ruchu wskazówek zegara) lub 1 (punkty konturu są określone w kierunku ruchu wskazówek zegara)
2. położenie X: liczba rzeczywista ułamkowa
3. położenie Y: liczba rzeczywista ułamkowa
4. Kąt zawarty: liczba ułamkowa. Jeśli wartość wynosi 0 to odcinek linii prostej jest rysowany od poprzedniego punktu do tego punktu. Jeśli wartość wynosi 360, poprzedni punkt określa środek okręgu i punkt ten określa punkt na okręgu; nigdy nie należy podawać wartości -360, gdyż co najmniej jeden z wiodących pakietów mechanicznego CAD nie będzie się poprawnie zachowywał w takiej sytuacji. Jeśli wartość jest ujemna to jest rysowany łuk w prawo z poprzedniego punktu do tego punktu, a jeśli wartość jest dodatnia, to łuk jest rysowany w lewo.

Tylko jedna zamknięta pętla jest dopuszczalna i nie jest możliwe opisywanie wycięć. Ostatni punkt musi być taki sam jak pierwszy chyba, że kontur jest okrągły.

Pierwszy przykład pliku IDF:

```
# prosty cylinder - może reprezentować kontur kondensatora elektrolitycznego
.ELECTRICAL
  "cylinder" "5mm OD, 5mm height" MM 5
  0 0 0 0
  0 2.5 0 360
.END_ELECTRICAL
```

Drugi przykład pliku IDF:

```
# odwrócone T
# komentarz dodany by zaprezentować możliwość komentowania
.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
```

## Wskazówki przy tworzeniu konturów

Podczas tworzenia konturów, zwłaszcza gdy są one udostępniane innym osobom, odpowiednie podejście przy ich tworzeniu i nazewnictwo pomoże innym szybciej zlokalizować pliki oraz umieszczać komponenty bez większych kłopotów.

### Nazwy pakietów

Należy starać się zawierać pewne informacje o konturze w nazwie pliku, aby dać użytkownikowi ogólne pojęcie o tym, jaki kontur zawiera plik. Na przykład cylindryczna obudowa z wyprowadzeniami osiowymi może odzwierciedlać niektóre z rodzajów kondensatorów foliowych, jak również niektóre typy rezystorów, więc sensowne jest, by starać się identyfikować taki obrys jako poziomy lub pionowy z wyprowadzeniami, dodając kilka ważnych informacji o odpowiednich wymiarach: średnica, długość i rozstaw. Jeśli element posiada unikatowy kontur, warto dodać numer części według producenta i prefiks by wskazać klasę urządzenia.

### Komentarze

Należy użyć komentarzy w pliku IDF, aby dać użytkownikom więcej informacji o konturze, na przykład odniesienie do źródła wykorzystywanego przy jego wymiarowaniu.

## Wpisy Geometriy i Part Number

Należy dokładnie przemyśleć jakie wartości zostaną przekazane do wpisów Geometriy oraz Part Number. Razem wzięte, treści te działają jako unikatowe identyfikatory dla systemów MCAD. Idealnie byłoby gdyby wartości tych łańcuchów miały pewne znaczenie dla użytkownika, ale nie jest to konieczne: wartości te są przeznaczone przede wszystkim dla systemu MCAD i używane jako niepowtarzalny identyfikator. Idealnie gdyby wybrane wartości były unikatowe w każdej dużej kolekcji konturów; wybierając odpowiednie wartości spowodujemy również mniej kolizji, zwłaszcza w przypadku złożonych płytek.

## Orientacja wyprowadzeń oraz ich ustawianie

Dla komponentów przewlekanych nie istnieją powszechnie akceptowane standardy dotyczące orientacji pinów oraz centrowania komponentu w modelach 3D. Dla spójności, jeśli są tylko 2 piny muszą być one w układzie poziomym (patrz [Rysunek 4](#)) wzdłuż osi X, a dla 3 pinów należy starać się utrzymać 2 z nich w układzie poziomym według osi X. Spolaryzowane komponenty takie jak kondensatory elektrolityczne lub tantalowe muszą posiadać dodatkowo wyprowadzenie na pinie 1, a dla diod pinem numer 1 musi być katoda; taki układ zachowuje zgodność z orientacją określoną dla komponentów SMT; należy jednak pamiętać, iż wiele z istniejących schematów w programie KiCad posiada anodę na pinie numer 1.

### NOTE

W najnowszych wersjach programu KiCad footprinty w repozytoriach GitHub mają anodę na pinie 2 dla elementów THT, tak samo jak w przypadku elementów SMD.

W przypadku komponentów DIP centrum konturu musi znajdować się w środku prostokąta opisanego w miejscu pinów, a pin 1 najlepiej by znajdował się w lewym górnym rogu; będzie to utrzymywać spójność ze znormalizowaną orientacją komponentów SMT; jednakże, taki model zostanie obrócony o -90 stopni w stosunku do większości istniejących footprintów w programie KiCad i modeli VRML. W przypadku elementów takich jak poziome kondensatory lub obudowy TO-220, najlepiej umieścić wyprowadzenia w rzędzie według osi X a korpus wyciągnięty do góry (patrz [Rysunek 4](#)). Niespolaryzowane pionowe elementy z wyprowadzeniami osiowymi muszą mieć wyprowadzenia z prawej strony; spolaryzowane pionowe elementy przewlekane mogą mieć wyprowadzenia z każdej strony, w zależności od tego, czy pin 1 znajduje się na dolnym końcu (wyprowadzenie po prawej) lub na górnym końcu (wyprowadzenie po lewej).

### NOTE

W obecnej wersji programu KiCad footprinty elementów THT zorientowane są pinami wzdłuż osi Y zamiast osi X i pin 1 elementu jest w punkcie zaczepienia niżeli w środku obudowy. Zorientuj oraz umieść kształt komponentu według własnego uznania; pozwoli to uniknąć konieczności określenia niezerowego obrotu dla konturów komponentu IDF. Ponieważ eksporter IDF obecnie ignoruje wartości przesunięcia (X, Y), ważne jest, by użyć odpowiedniego punktu zaczepienia w zarysach komponentów IDF.

W komponentach SMT orientacja, centrum pakietu oraz kontur są definiowane według różnych standardów. Należy użyć standardu właściwego dla własnego systemu pracy. Należy także pamiętać, że wiele komponentów nie odpowiada żadnym standardom; w takich przypadkach komponent wykraczający poza ramy najlepiej jest identyfikować przez numer części według producenta, umieszczonego w nazwie pliku konturu. Generalnie, obudowa SMT to kontur w postaci prostokąta obejmującego również wyprowadzenia; obudowa jest zorientowana tak, że pin 1 jest tak blisko jak to tylko możliwe lewego górnego rogu, a lewy górny róg jest zwykle sfazowany by był rozróżnialny.



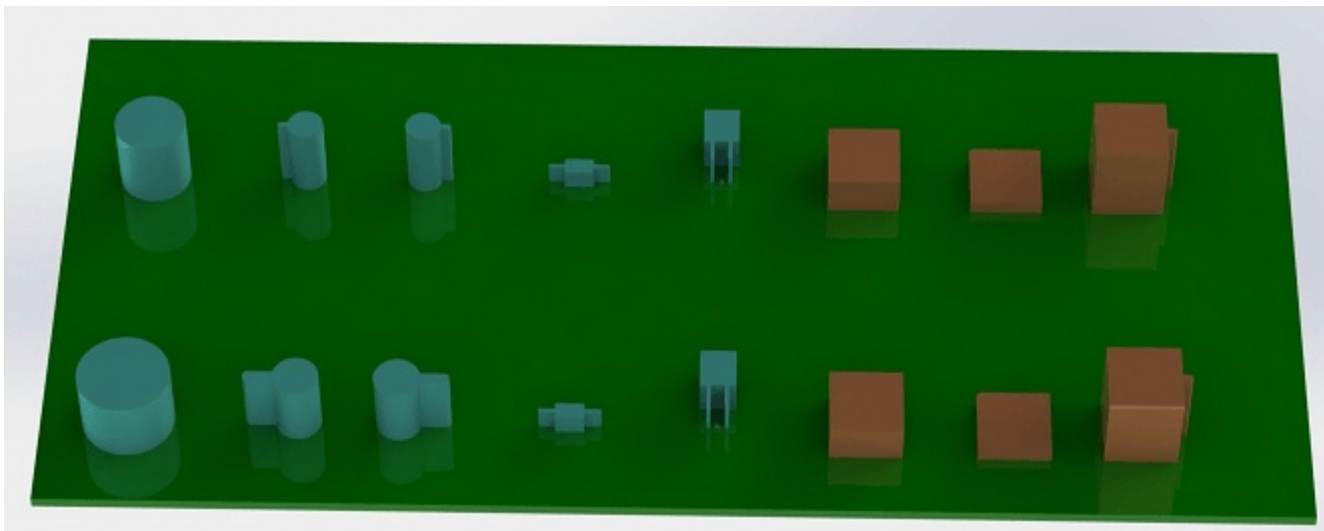


Figure 4. Próbkę konturów generowane przez programy idfrect oraz idfcyl zrenderowane przez SolidWorks.

Od lewej do prawej: (a) pionowy cylinder z wyprowadzeniami promieniowymi, (b) pionowy cylinder z wyprowadzeniem osiowym na lewo, (c) pionowy cylinder z wyprowadzeniem osiowym na prawo, (d) poziomy cylinder z wyprowadzeniami osiowymi, (e) poziomy cylinder z wyprowadzeniami promieniowymi, (f) kwadratowy zarys, zwykły, (g) kwadratowy zarys z fazowaniem, (h) kwadratowy zarys z wyprowadzeniem osiowym na prawo. Kontury na górze zostały zwymiarowane w milimetrach, podczas gdy dolne kontury zostały zwymiarowane w jednostkach calowych.

## Wskazówki dotyczące wymiarów

Celem dostarczenia "wyciąganych" konturów jest danie projektantowi mechanicznemu pewnego wyobrażenia o lokalizacji i fizycznej przestrzeni zajmowanej przez każdy z elementów. W typowym scenariuszu projektant mechaniczny zastąpi niektóre surowe modele za pomocą bardziej szczegółowych modeli, na przykład podczas sprawdzenia, czy zamontowane kątowe diody LED pasują do otworów w panelu. W większości sytuacji dokładność konturu nie ma znaczenia, ale do dobrej praktyki należy stworzenie zarysów, które przeniosą możliwie jak najlepszą informację mechaniczną. W kilku przypadkach użytkownik może chcieć, by dopasować komponent z bardzo niewielkim nadmiarem przestrzeni, na przykład w przypadku przenośnego odtwarzacza muzyki. W takiej sytuacji, jeżeli większość konturów dostarcza wystarczająco dobrą reprezentację komponentów, projektant mechanicznej części będzie musiał wymienić tylko niewiele modeli podczas projektowania obudowy. Jeśli kontury nie odbijają w pełni rzeczywistości, to projektant mechanicznej części będzie tylko tracił czas na zastępowanie modeli, tak by zapewnić im dobre dopasowanie. Mimo wszystko, jeśli umieścimy niepoprawne modele, to można oczekiwać tylko niepoprawnych rezultatów. Jeśli zaś umieścimy wiele odpowiednich informacji, można mieć pewność, że spotkamy się z dobrymi wynikami.

## Narzędzia dla Kształtów IDF

Dostępny jest pewien zbiór narzędzi w linii poleceń, które pomagają tworzyć obrysy IDF. Narzędzia te to:

1. **idfcyl**: tworzy kontur cylindra w pionowej lub poziomej orientacji oraz osiowe lub promieniowe wyprowadzenia.
- 2.

**idfrect:** tworzy zarys prostokąta, który może mieć zarówno wyprowadzenia osiowe lub sfazowania w lewym górnym rogu.

3. **dx2idf:** konwertuje rysunki w formacie DXF na kontury komponentów IDF.

## idfcyl

Gdy **idfcyl** zostanie wywołany bez parametrów użytkownik otrzyma informacje o jego użyciu i skrót informacji o danych wejściowych:

```
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

Uwagi mogą zostać pominięte poprzez wprowadzenie odpowiednich argumentów w linii poleceń. Użytkownik może ręcznie wprowadzić informacje za pomocą linii poleceń lub osobnego skryptu generującego obrisy. Następujący skrypt tworzy pojedynczy okrągły kształt z wyprowadzeniem osiowym z prawej strony:

```
#!/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
```

## idfrect

Gdy **idfrect** zostanie wywołany bez parametrów użytkownik otrzyma informacje o jego użyciu i skrót informacji o danych wejściowych:

```
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
```

Uwagi mogą zostać pominięte poprzez wprowadzenie odpowiednich argumentów w linii poleceń. Użytkownik może ręcznie wprowadzić informacje za pomocą linii poleceń lub osobnego skryptu generującego obrisy. Następujący skrypt tworzy sfazowany kwadrat i okrągły kształt z wyprowadzeniami:

```

#!/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

```

## dxf2idf

Plik DXF używany do określenia konturów komponentów może zostać przygotowany poprzez darmowe narzędzie [LibreCAD](#) by zachować najlepszą kompatybilność. Gdy **dxf2idf** zostanie wywołany bez parametrów użytkownik otrzyma informacje o jego użyciu i skrót informacji o danych wejściowych:

```

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'

```

Uwagi mogą zostać pominięte poprzez wprowadzenie odpowiednich argumentów w linii poleceń. Użytkownik może ręcznie wprowadzić informacje za pomocą linii poleceń lub osobnego skryptu generującego obrisy. Następujący skrypt tworzy obrys o wysokości 5mm z pliku DXF '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
```

## idf2vrm1

Narzędzie *idf2vrm1* odczytuje jeden plik IDF Board (.emn) i jeden IDF Component (.emp) i tworzy plik VRML, który można podejrzeć za pomocą przeglądarki VRML. Ta funkcjonalność jest stosowana w celu wizualizacji ułożenia elementów na płytce w tych przypadkach, gdzie użytkownik nie posiada dostępu do oprogramowania MCAD. Uruchomienie *idf2vrm1* bez parametrów zwraca informacje o jego użyciu:

```
>./idf2vrm1
Usage: idf2vrm1 -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: idf2vrm1 -f input.emn -s 0.3937008 -k
>
```

[**BŁĄD:** Narzędzie *idf2vrm1* obecnie nie renderuje wpisów **OTHER\_OUTLINE** w pliku *emn* jeśli te wpisy zostaną umieszczone na spodniej stronie PCB; jednakże można zauważyć, że błąd ten nie pojawi się podczas używania plików wyeksportowanych z programu KiCad, ponieważ nie ma mechnizmu by takie wpisy umieścić. Błąd ten głównie występuje w rzadkich przypadkach, w których mogły być renderowane pliki *emn* z innych źródeł, które zawierają elementy na tylnej stronie płytki.]

---

1. [http://www.simplifiedsolutionsinc.com/images/idf\\_v30\\_spec.pdf](http://www.simplifiedsolutionsinc.com/images/idf_v30_spec.pdf)