

Printed Circuit Board Workbench for [FreeCAD](#) (PCB)

Flexible Printed Circuit Board Workbench for [FreeCAD](#) (FPCB)

marmni (marmni@onet.eu)

Copyright 2013, 2014, 2015

<http://sourceforge.net/projects/eaglepcb2freecad/>



Index

Introduction	4
Requirements.....	5
Supported files.....	5
Installation	6
GNU/Linux.....	6
Windows.....	6
Configuration	8
STP file format colors definition.....	8
Accessing the Workbench	10
Set PCB module as main workbench.....	11
Menu Bar	12
Toolbars	13
PCB View toolbar.....	13
PCB Settings toolbar.....	14
Displaying toolbars.....	17
Specification Tree	18
Customizing Workbench	20
Assign models	24
Example of adding part to database.....	30
Multi model definition for one part.....	32
Download models	34
Display modes	35
Layers	37
Python.....	38
Grouping parts	39
Open/Import board	40
Unit system.....	41
Create new project	43
Python.....	44
Create PCB	45
Cut to board outline	49
Create glue path	51
Add annotation	52
Add model	58
Update models	62
Create constraint area	64
Explode	67
Bounding box	71
Python.....	73
Export board	74
Supported files.....	74
Unit system.....	76
Python.....	77
Export Bill Of Materials (BOM)	78
Python.....	80
Export hole locations	82
Python.....	85
Export hole locations report	87
Python.....	88
Create drilling map	89
Python.....	91
Create drill center	92
Python.....	93
Add assembly	94
Python.....	95
Update assembly	96

Python.....	97
Export to Kerkythea	98
Objects properties	99
Annotation/Object Name/Object Value.....	99
Board.....	100
Part model not found in database.....	100
Explode.....	101
Part model found in database.....	101
Constraint area.....	102
Glue path.....	102
Main assembly object.....	102
File format	103
Scripts	105
Eagle.....	105
Razen.....	105
Errors code	106
Licence	107
Changelog	108
ToDo list	112
Errors	113

INTRODUCTION

[ENG]

Mod allow you to import/create PCB boards in FreeCAD. Scope of mod:

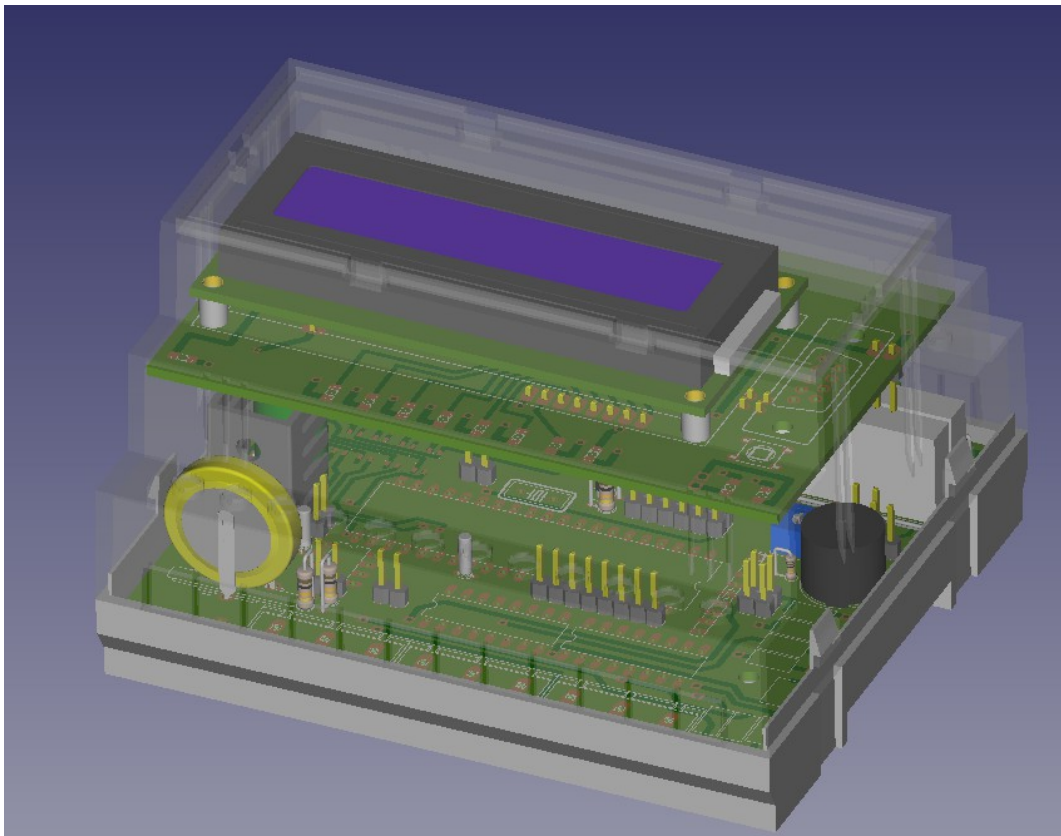
- support for many different layers,
- possible to choose colours, transparency and names for each layer,
- mod allows you to import IGES/STP models with colours,
- possible to show holes/vias independent.

[PL]

Moduł pozwala na importowanie/tworzenie płytek PCB w programie FreeCAD.

Możliwości modułu:

- wsparcie dla wielu różnych warstw,
- wyświetlanie otworów, przelotek niezależnie od siebie,
- możliwość wyboru koloru, przezroczystości oraz nazwy dla poszczególnych warstw,
- importowanie modeli zapisanych w formacie IGS/STP wraz z kolorami.



Requirements

FreeCAD-PCB require FreeCAD in version 14.0 or never. Module was tested on Windows and GNU/Linux.

Supported files

- Eagle (*.brd),
- Razen (*.rzp),
- FreePCB (*.fpc),
- gEDA (*.pcb),
- FidoCadJ (*.fcd),
- KiCad (*.kicad_pcb),
- IDF v2/v3,
- HyperLynx (*.HYP).

INSTALLATION

Unpack downloaded zip file and copy extracted folder to direction where FreeCAD is installed (subfolder Mod).

GNU/Linux

Example:

FreeCAD path:

~/Programs/FreeCAD

So copy mod to folder

~/Programs/FreeCAD/Mod

You can also copy files to folder ***~/.FreeCAD/Mod***.

Next change read/write permission to 777. Please don't forget about parameter -R!

Example:

chmod 777 -R PCB

Windows

Example:

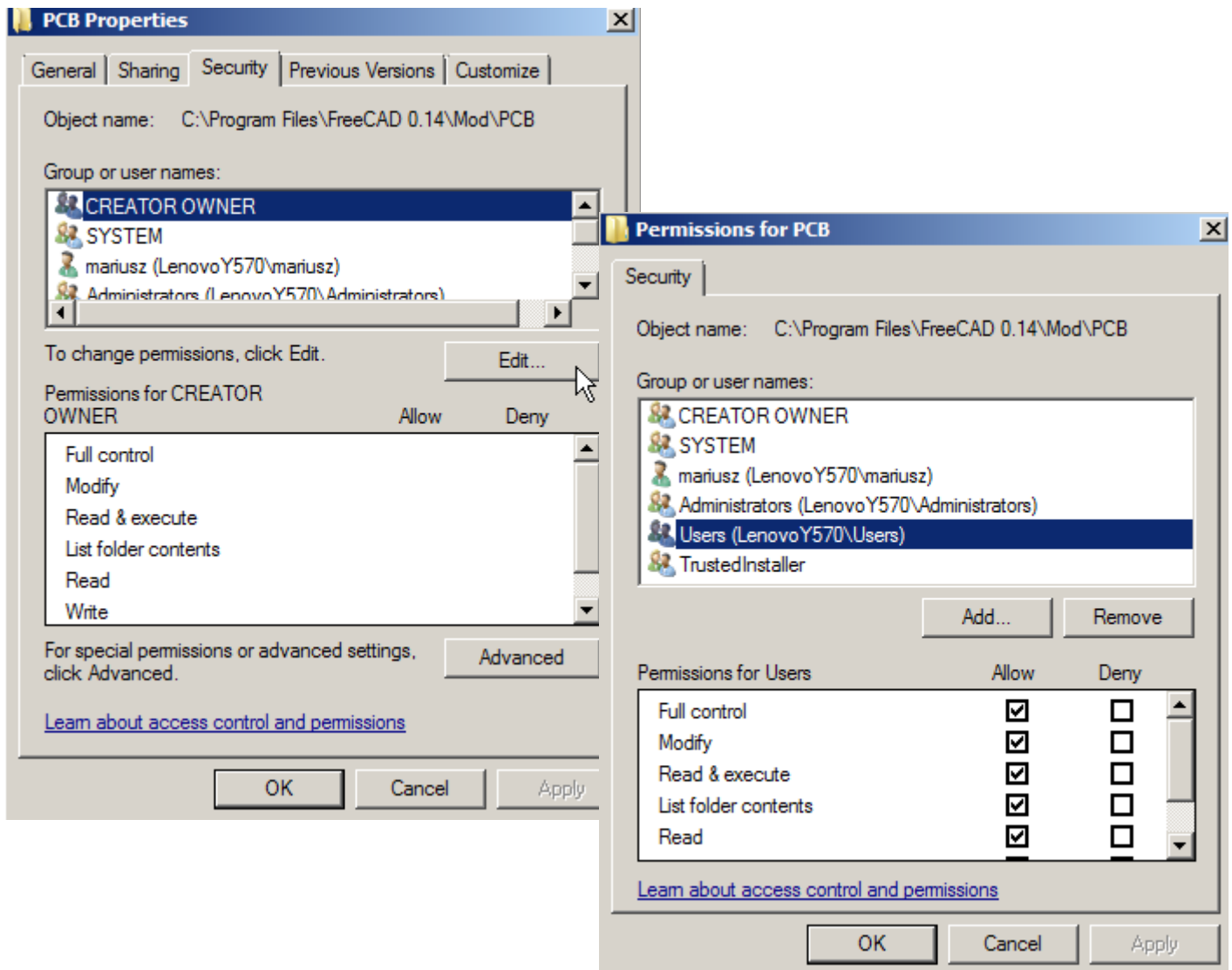
FreeCAD path:

C:/Program Files/FreeCAD-0.14

So copy mod to folder

C:/Program Files/FreeCAD-0.14/Mod

Next change read/write permission for all users. Click right button on folder PCB and choose Properties → Security → Edit → Users and mark all checkboxes under 'Allow' option.



CONFIGURATION

At this moment some settings need to be configured in file PCBconf.py. You can open this file in any text editor (please avoid Notepad).

STP file format colors definition

During loading board, You can meet with error connected with missing STP color definition. To fix that problem just add new color definition in PCBconf.py file in spisKolorowSTP variable.

For example:

Missing color name:

red

Actual situation:

```
spisKolorowSTP = {  
    "white": (1.0, 1.0, 1.0),  
    "black": (0.0, 0.0 ,0.0)  
}
```

Write to file:

```
spisKolorowSTP = {  
    "red": (1.0, 0.0 ,0.0),  
    "white": (1.0, 1.0, 1.0),  
    "black": (0.0, 0.0 ,0.0)  
}
```

Where:

"red": (1.0, 0.0 ,0.0), => „colorName”: (R / 255, G / 255, B / 255)

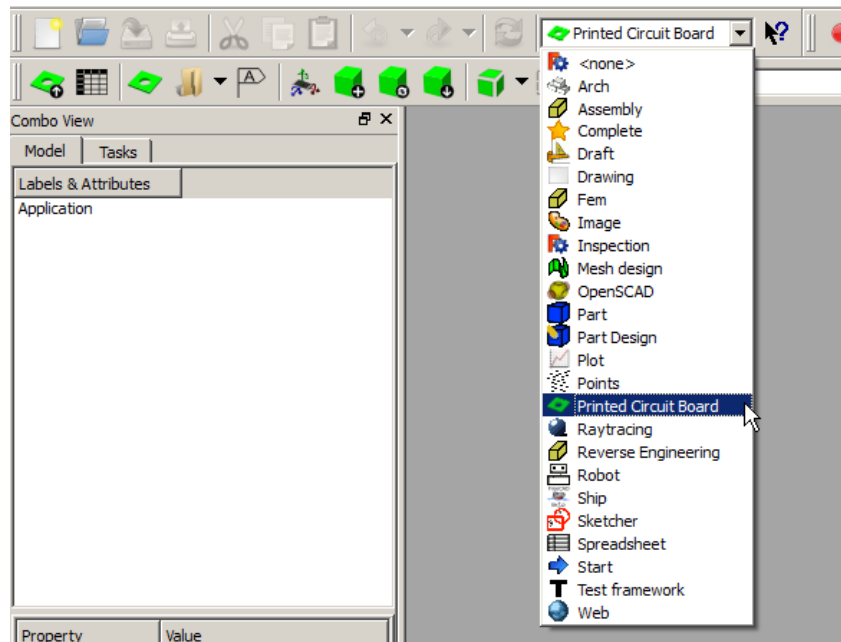
Caution!

Please do not change anything else in the PCBconf.py file!
More configuration options You can find in [Customizing Workbench](#) section.

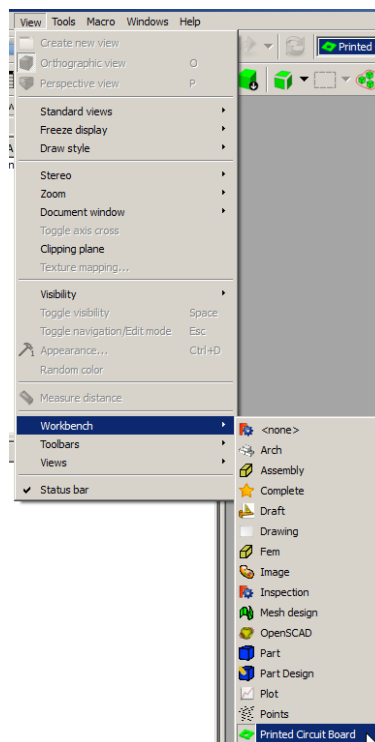
ACCESSING THE WORKBENCH

There are two methods to access to the PCB workbench:

1. In toolbar 'File' locate drop down list and choose 'Printed Circuit Board'.

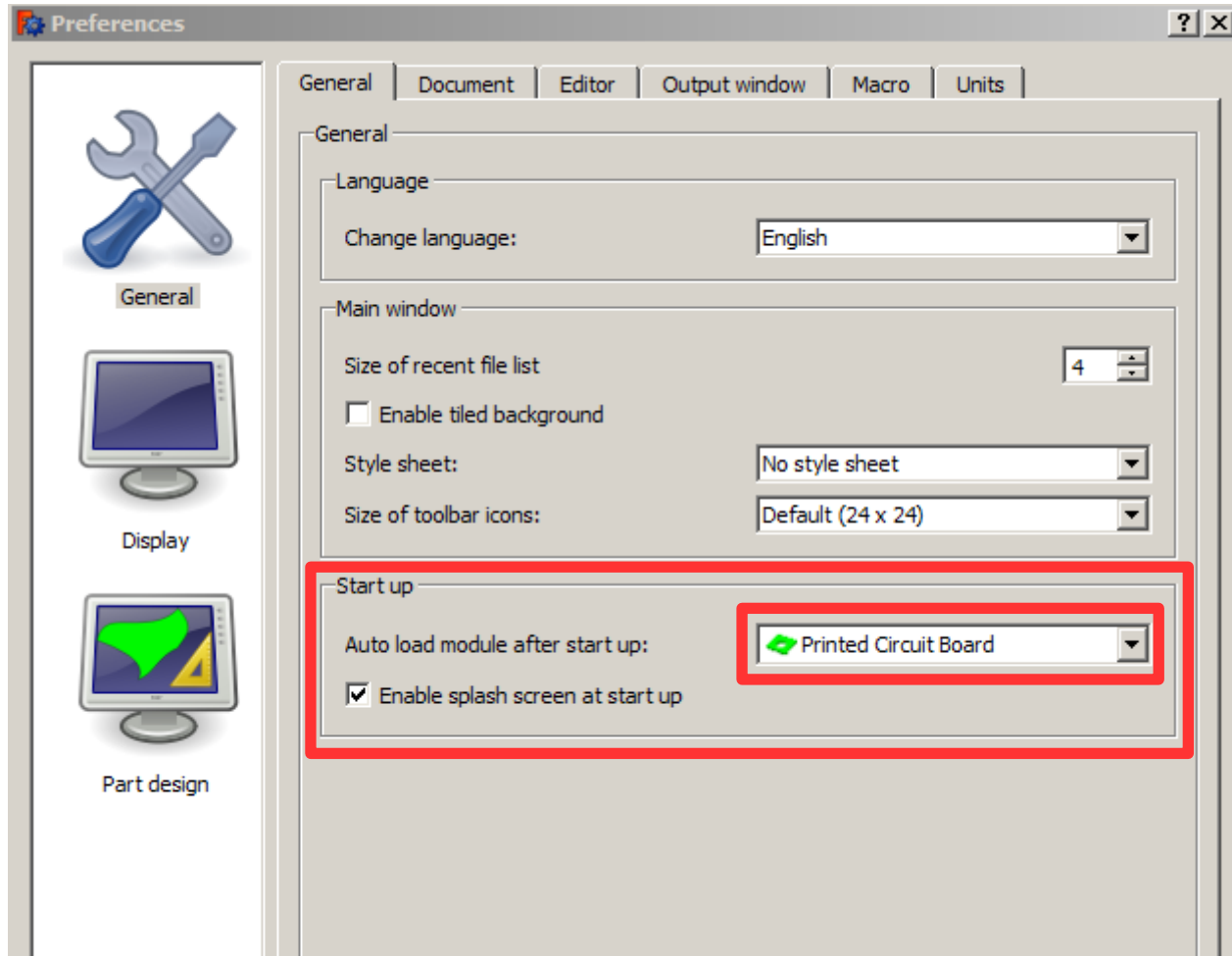


2. From top menu bar choose View → Workbench → Printed Circuit Board.



Set PCB module as main workbench

There is possibility to set PCB module as main workbench. To do this choose from top menu bar Edit → Preferences, in settings window choose General and tab General. In displayed tab You should find 'Start up' section, where You can set which workbench should be loaded after FreeCAD start.



MENU BAR

Menu bars are not available.

TOOLBARS

Three toolbars are available in PCB workbench:

1. PCB View.
2. PCB Settings.
3. Sketcher.

This section describes the various icons available in mentioned toolbars.

PCB View toolbar



Change display mode to Shaded

See [Display modes](#) section



Change display mode to Flat Lines

See [Display modes](#) section



Change display mode to Wireframe

See [Display modes](#) section



Change display mode to Internal View

See [Display modes](#) section



Layers settings

See [Layers](#) section



Cut to board outline

See [Cut to board outline](#) section



Ungroup models in 'Parts' folder

See [Grouping parts](#) section



Group models in 'Parts' folder

See [Grouping parts](#) section



3D rendering: export to Kerkythea

See [Kerkythea](#) section



Load file as assembly

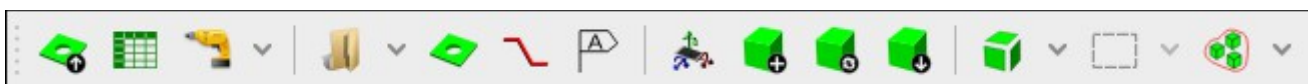
See [Add assembly](#) section



Update selected assemblies

See [Update assembly](#) section

PCB Settings toolbar



Export PCB

See [Export board](#) section



Export BOM

See [Export Bill Of Materials \(BOM\)](#) section



Export hole locations

See [Export hole locations](#) section

Export hole locations report

See [Export hole locations report](#) section

Create drilling map

See [Create drilling map](#) section

Create drill center

See [Create drill center](#) section



Create new project

See [Create new project](#) section



Create PCB

See [Create PCB](#) section



Create glue path

See [Create glue path](#) section



Add annotation

See [Add annotation](#) section



Assign models

See [Assign models](#) section



Add model

See [Add model](#) section



Update models

See [Update models](#) section



Download models

See [Download models](#) section



Explode

See [Explode](#) section



Create constraint area

See [Create constraint area](#) section

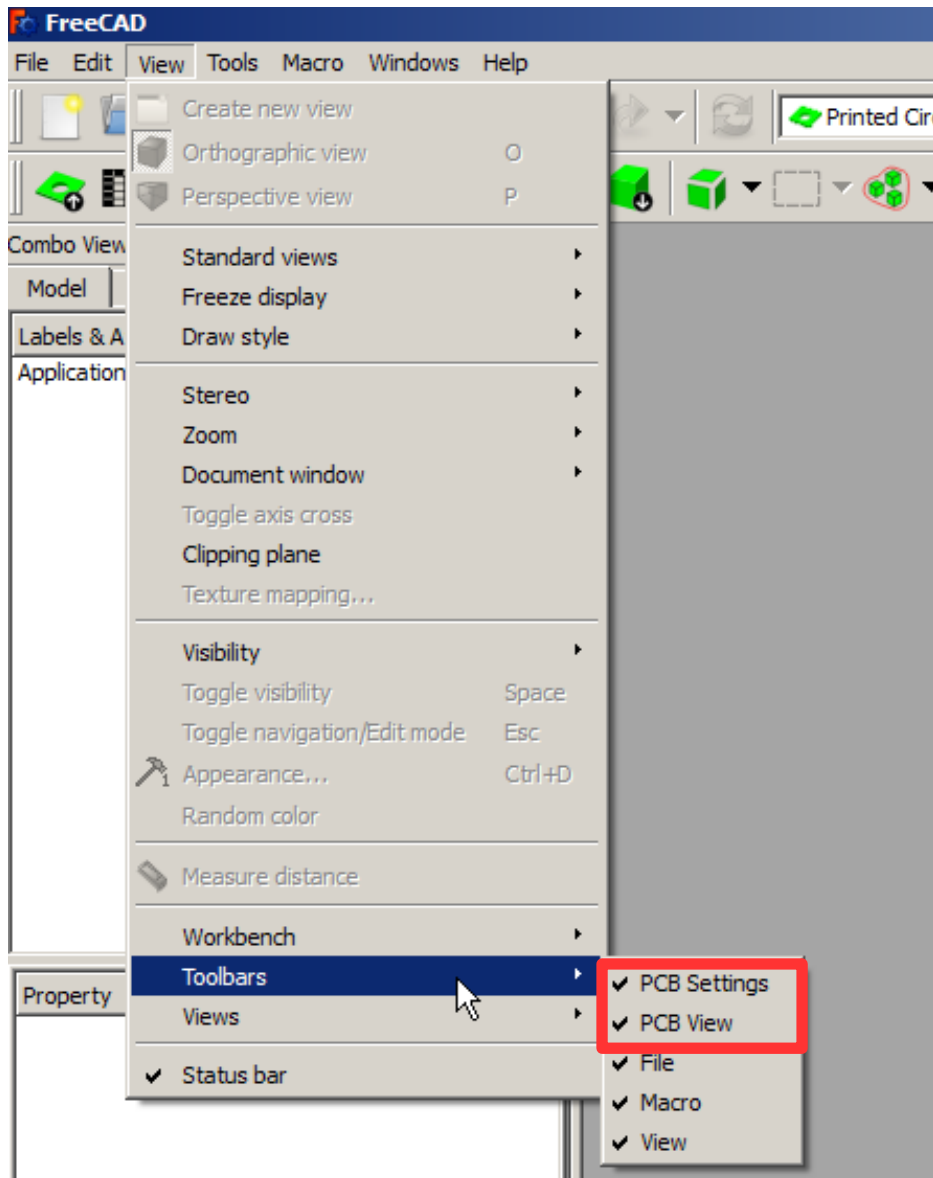


Bounding box

See [Bounding box](#) section

Displaying toolbars

When mentioned toolbars are not displaying after choosing PCB workbench in main FreeCAD window, You need to do it manually. From top menu bar choose View → Toolbars and mark toolbars from Printed Circuit Board workbench.



SPECIFICATION TREE

There are few object types directly connected with PCB workbench. They can be identified in the 'Combo view' by the specific icons.



Board



Constraint area



Explode



Layer



Part model found in database



Part model not found in database



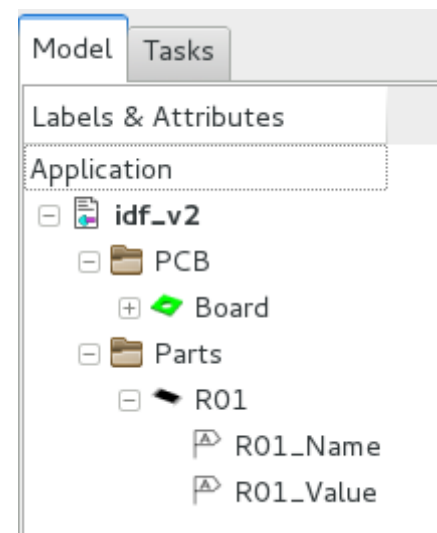
Annotation/Object Name/Object Value



Glue path



Main assembly object





Main assembly subcomponent

More info about mentioned objects You can find in [Objects properties](#) section.

CUSTOMIZING WORKBENCH

To access to the PCB workbench settings You need to choose from top menu Edit → Preferences: section PCB.

Preference tab for module contain three groups:

1. General

The screenshot shows the 'General' tab of the PCB Workbench Preferences dialog. It features three sub-sections: 'Board', 'Parts', and 'Libraries'. The 'Board' section includes a 'Default software' dropdown set to 'Eagle', a 'Default board thickness' spinner set to '1,5', and checkboxes for 'Load board thickness from file (if available)' (checked) and 'Import holes' (unchecked). The 'Parts' section has checkboxes for 'Import parts' (checked), 'Group parts' (checked), 'Colorize elements' (checked), 'Adjust part name/value' (unchecked), 'Generate report with unknown parts' (unchecked), and 'PCB-Decals (for IDF)' (checked). The 'Paths' field contains 'Path_1,Path_2,Path_3'. The 'Database' section has a 'Path' field with a browse button. The 'Libraries' section lists 'FidoCadJ' with path '/home/mariusz/Programy/FidoCadJ/lib' and 'Razen' with path '/home/mariusz/Programy/razen/1.0.0/libraries'.

Section	Option	Value / State
Board	Default software	Eagle
	Default board thickness	1,5
	Load board thickness from file (if available)	<input checked="" type="checkbox"/>
Parts	Import parts	<input checked="" type="checkbox"/>
	Group parts	<input checked="" type="checkbox"/>
	Colorize elements	<input checked="" type="checkbox"/>
	Adjust part name/value	<input type="checkbox"/>
	Generate report with unknown parts	<input type="checkbox"/>
	PCB-Decals (for IDF)	<input checked="" type="checkbox"/>
Paths	Path_1,Path_2,Path_3	
Database	Path	[Browse Button]
Libraries	FidoCadJ	/home/mariusz/Programy/FidoCadJ/lib
	Razen	/home/mariusz/Programy/razen/1.0.0/libraries

This section contains default settings for import process:

- Default software: this field allow You to set default used by you software,
- Board thickness: default value is 1.5mm,

- Paths to: database, 3D models, extra libraries,
- Checkboxes associated with importing parts/colors/holes,
- Checkbox associated with generating report with unknown parts.

If checkbox 'Group parts' is checked, imported parts will be splitted to groups according to Category they belong.

**Caution!**

For more information about grouping parts see '[Grouping parts](#)' section.

**Caution!**

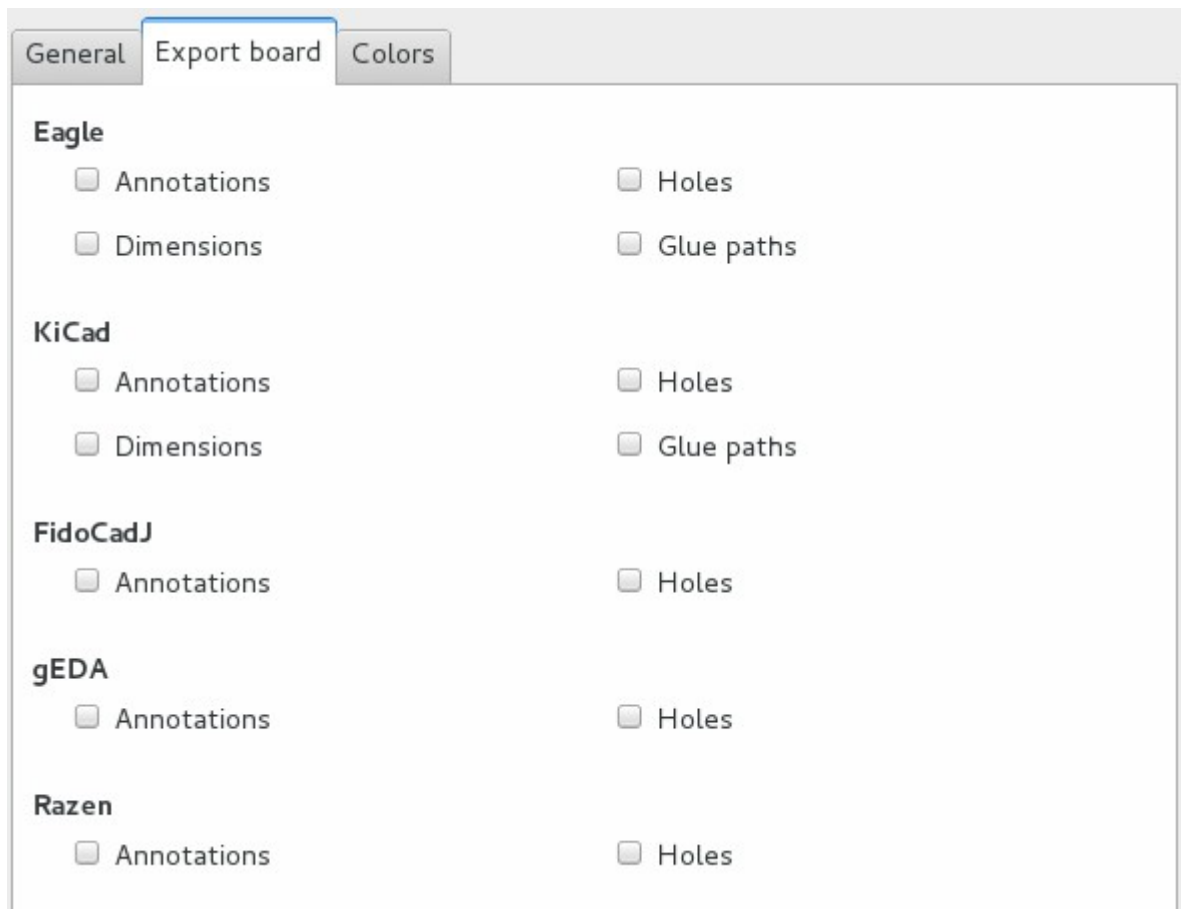
It is recommended to keep parts and database.cfg outside PCB folder.

**Caution!**

To set libraries for FidoCadJ you can indicate folder or main jar file.

2. Export board

Default settings associated with exporting board to one of supported formats can be set in this tab.



3. Colors




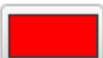



All default color can be set in 'Colors' section.

General Export board **Colors**





Board

Color 

Constraint areas

Place Outline Top 	Place Outline Bottom 
Place Outline 	
Route Outline Top 	Route Outline Bottom 
Route Outline 	
Route Keepout Top 	Route Keepout Bottom 
Via Keepout 	
Place Keepout Top 	Place Keepout Bottom 

Layers

Path 	Silk 	Pad 
Annotations 	Measures 	Center drill 
Glue 		

Window 'Assign models' allow for assigning 3D models to corresponding part from one of supported software.

[illegible]

Window comprises two main columns. Left column comprises function necessary to manage parts in database. While right side comprises form where You can set (or edit) package data.

Search block contains prev/next button and entry field for search term

Collapse/Expand all items from Package list

Reload database

Import new entries from different database file

Create database copy

Convert models from one software to another

Delete selected models from database

Manage Categories:
- add new category
- edit existing category
- remove selected category

Package list contains categories /parts names and their description

Name	Description
+ Batteries	
+ Buzzers	
+ Capacitors	
+ con-harting	
+ con-phoenix	
+ Connectors	
+ Crystals	
+ Diodes	
+ Display	
+ Goldpins	
+ Heatsinks	
+ Inductor	
+ Jumpers	
+ Led	
+ Packages	
+ Packages-*BGA	
+ Potentiometers	
+ Rectifiers	
+ Relays	
+ Resistors	
+ Switch-dil	
+ Varistors	

Caution!

Categories are stored directly in FreeCAD.

Tools → Edit parameters => Preferences → Mod → PCB and variable partsCategories



Caution!

After deleting model from database it is not possible to undo this operation!

Basics setting

Package type is model name (any text)

→ Package type

These field contains relative or absolute path to 3D model

→ Path to element

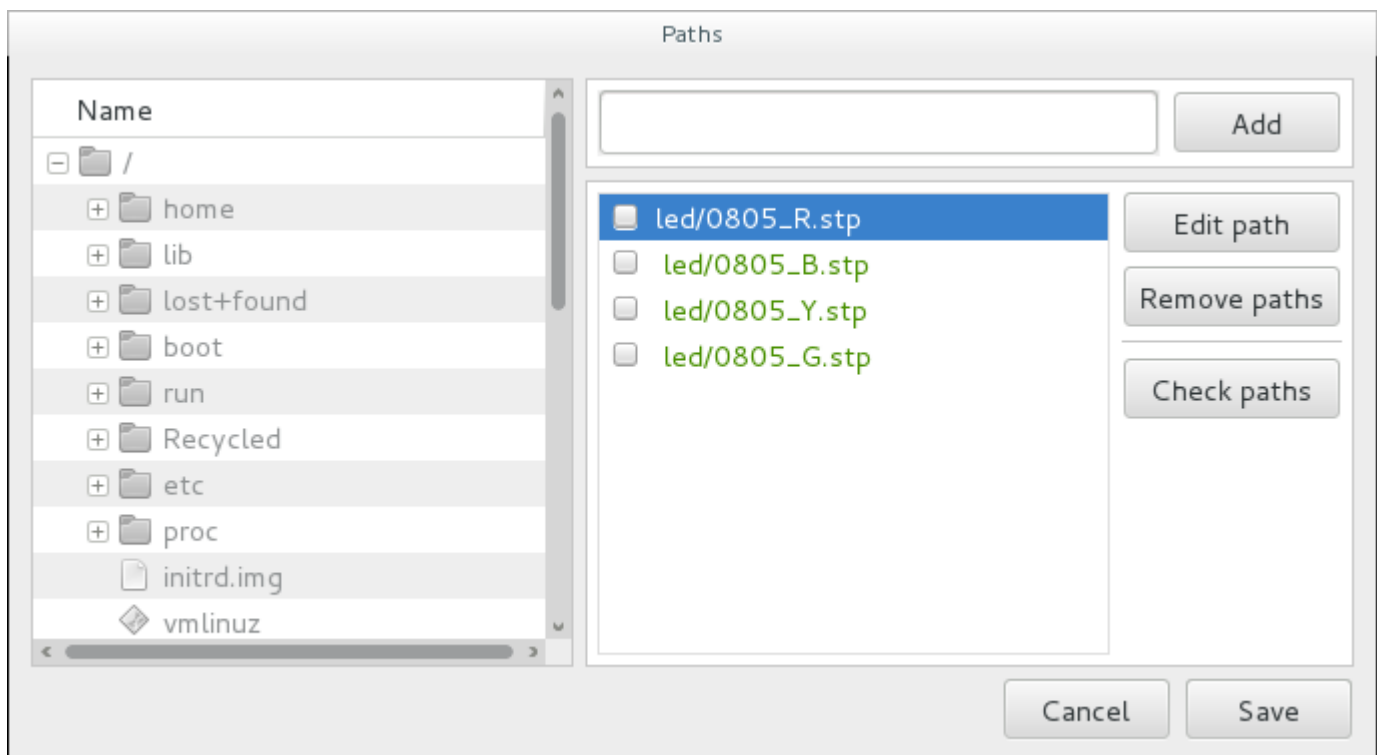
Optional parameter: path to part datasheet (web site or local file)

→ Datasheet

Set category for model

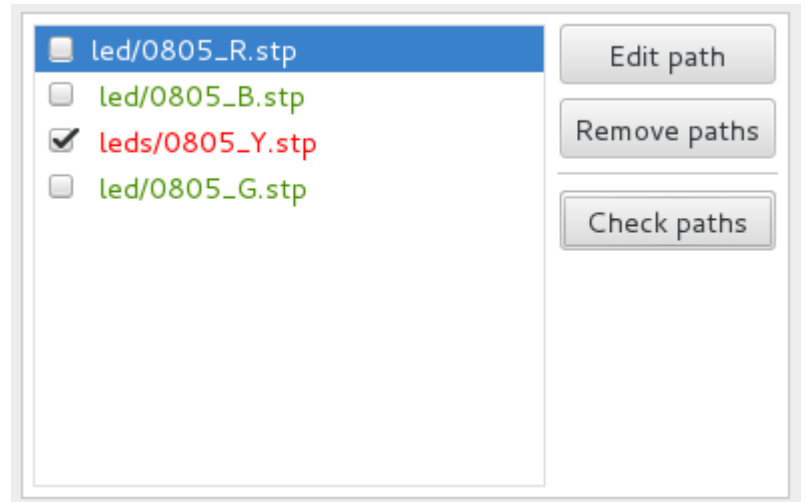
→ Category

From version 3.9 it is possible to set paths in gui.

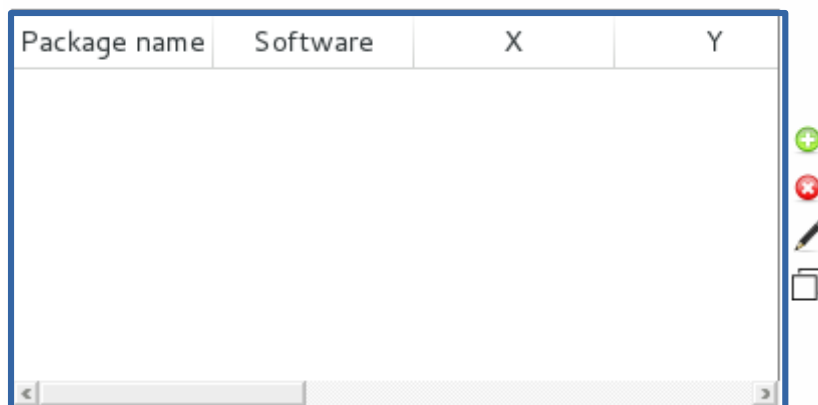


Add

To check new paths you need to click button 'Check paths'.



First sections contain all package parameters, from one of supported softwares, necessary to load specific 3D model.

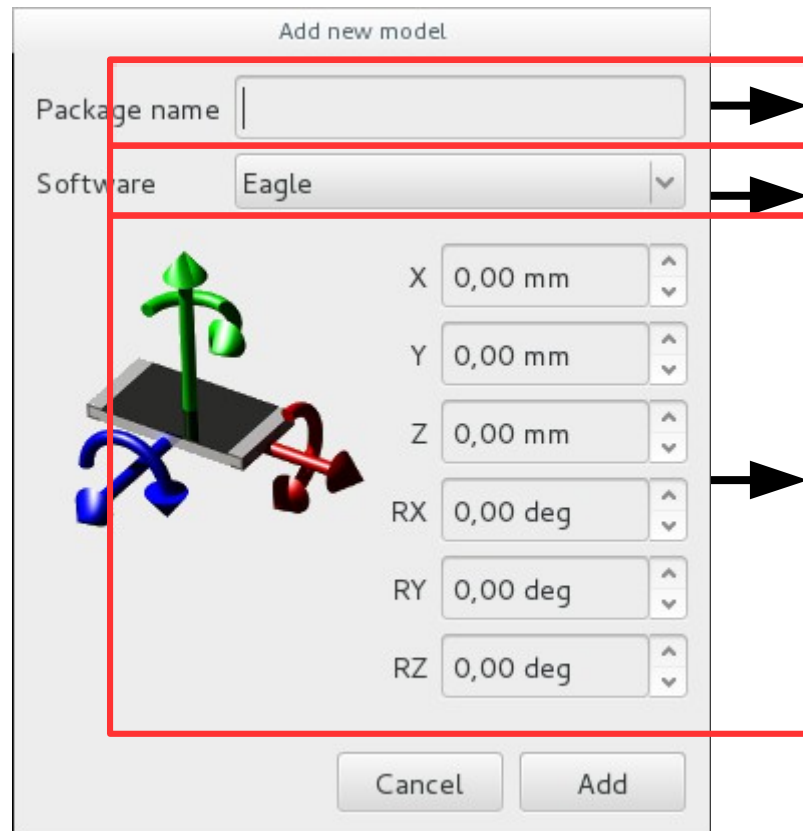


Last button allows you to copy existing entry and save it in database under new name.

This field contains package type name taken from software used by You to create PCBboards.

From drop-down list You need to choose software name for with this entry will be connected.

X, Y, Z, RX, RY, RZ parameters are correction values used to correctly placement 3D model.

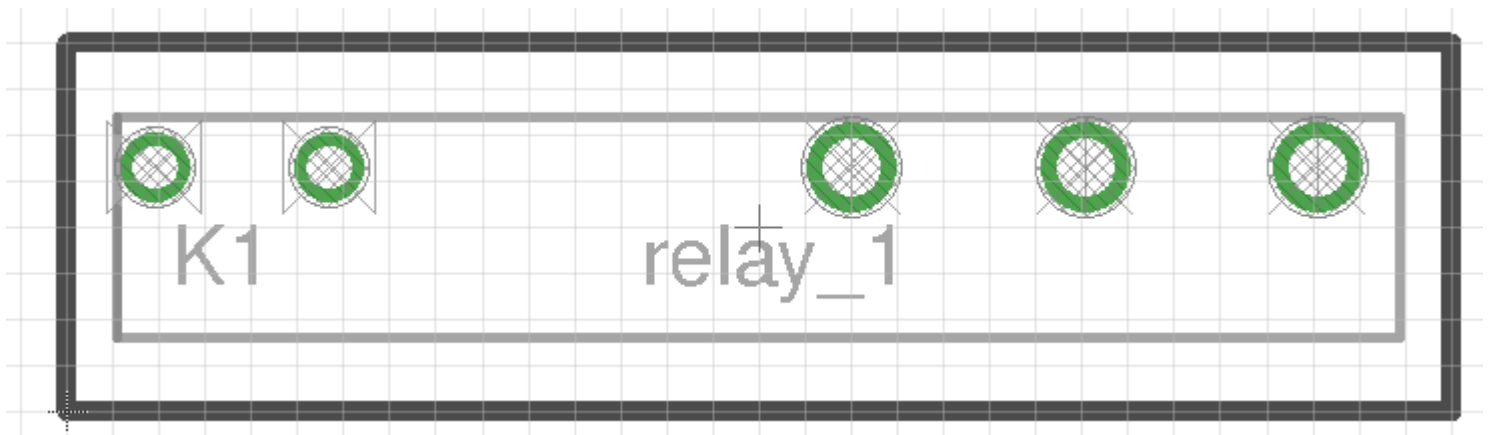


Adjust part name/value

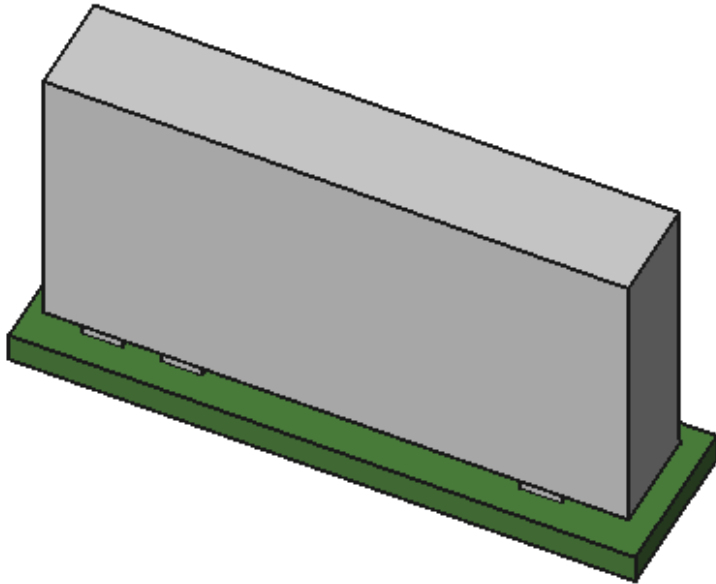
Option 'Adjust part name/value' allows to automatic placing objects name/value in specific position.

	Parameter	Visible	X	Y	Z	Size	Color	Align
<input checked="" type="checkbox"/>	Name	True	0,00mm	0,00mm	0,00mm	1,27mm		center
<input type="checkbox"/>	Value	True	0,00mm	0,00mm	0,00mm	1,27mm		center

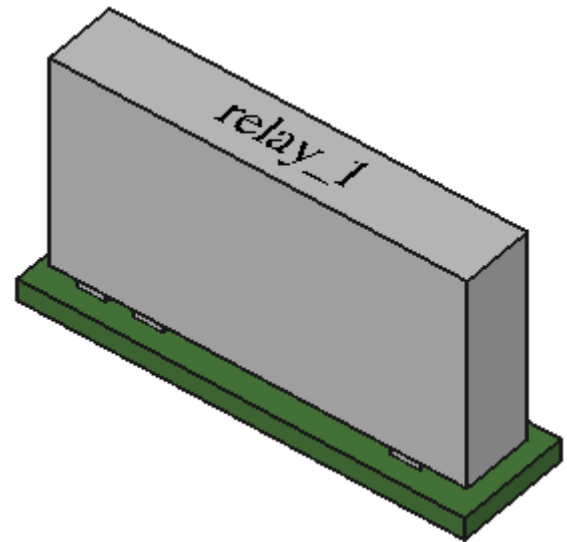
Example



	Parameter	Visible	X	Y	Z	Size	Color	Align
<input type="checkbox"/>	Name	True	0,00mm	0,00mm	14,95mm	3,00mm		center
<input checked="" type="checkbox"/>	Value	True	0,00mm	0,00mm	14,95mm	3,00mm		center



Result without option 'Adjust part name/value'



Result with option 'Adjust part name/value'

Set socket for model

To add socket for model just mark checkbox for 'Add socket' and from drop down list choose socket 3D model name. In drop down list You will find only models marked before as sockets.

☐ **Add socket**

Socket

Set model as socket

To set model as socket just mark checkbox for 'Set as socket' sign. In spinbox specify socket height.

☐ **Set as socket**

Height

Footer from right column contain buttons to saving form entries to database

Save button will save form as new entry in database or will update package parameters (in edit mode).

'Save as new' will save existing entry in database under new package name.

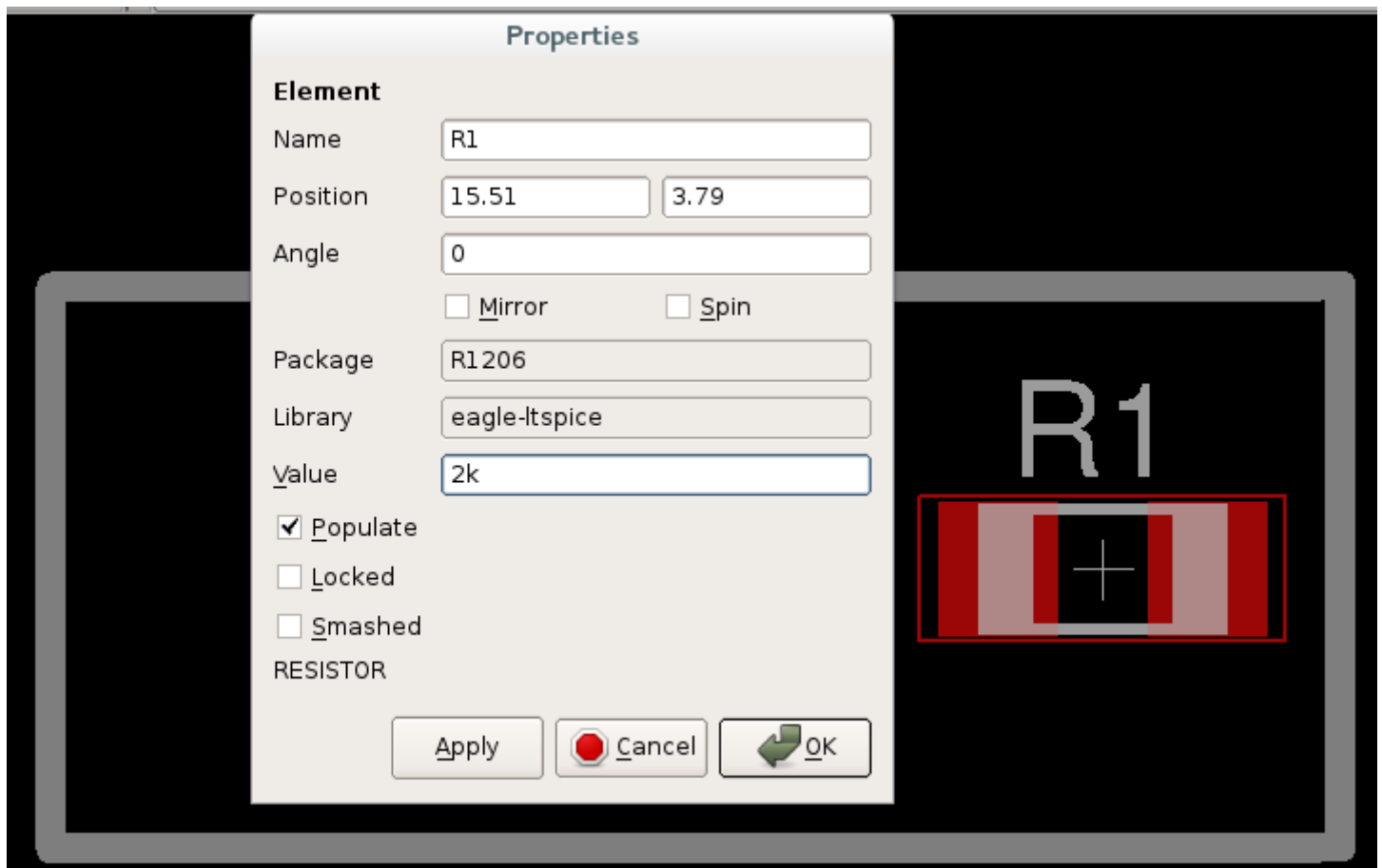
'Clean/New' button clean form.

Close button will appear only for GNU/Linux users.



Example of adding part to database

In Eagle we prepared board containing one part: R1 SMD resistor 2k in package 1206.



Open 'Assing models' window and:

1. In 'Package type' field write resistor_smd_1206. Package name is free text.
2. In 'Path to element' write ***resistors/R1206***.

**Caution!**

Model extension.

resistors/R1206 mean that script will search at first file R1206.igs and then R1206.stp,
so course path to element can also ook like
resistors/R1206.igs or ***resistors/R1206.stp***.

**Caution!**

resistors/R1206 – script will search model in one of relative paths set in
configurations.

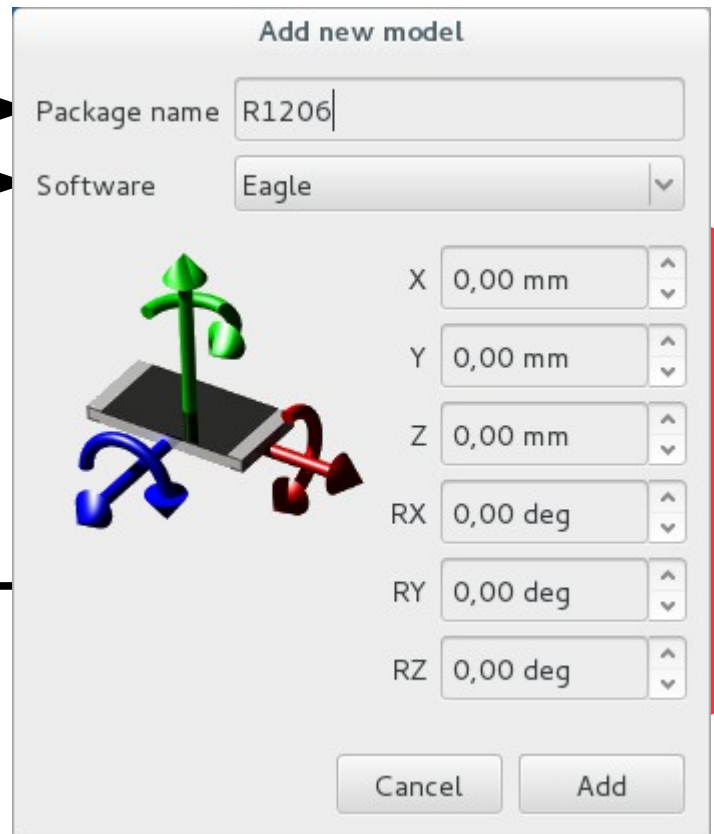
3. Datasheet field may remain empty.
4. Package category may remain as 'None'.
5. Next step explains how to add new 'connection' between 3D model and part added on board.

To create board we use Eagle so in 'Package name' we should give value shown next to 'Package' field from part properties window in Eagle

Software field we set on Eagle.

That mean all parts R1206 from board, created in Eagle will be connected with 3D model R1206.igs/.stp.

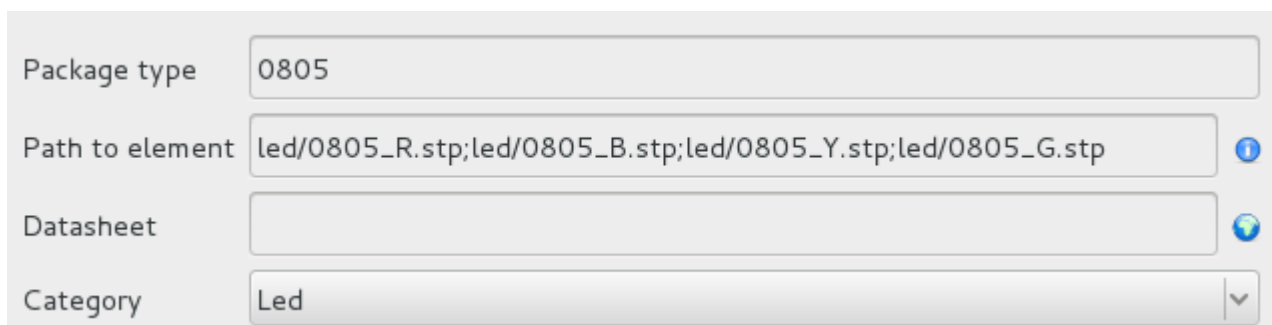
Each part is placed in space (X, Y, Z, RX, RY, RZ). Parts specified in Eagle contains only X, Y and rotation around Z axis. Sometimes is necessary to set some correction values to achieve specified in board part position – these values dependence from used 3D model.



6. Click add button.
7. This model is not socket and doesn't contain it, so we will skip specific checkboxes.
8. To save form in database click button 'Save'.
9. Thats all, now just reload board or update 3D models for R1206 packages.

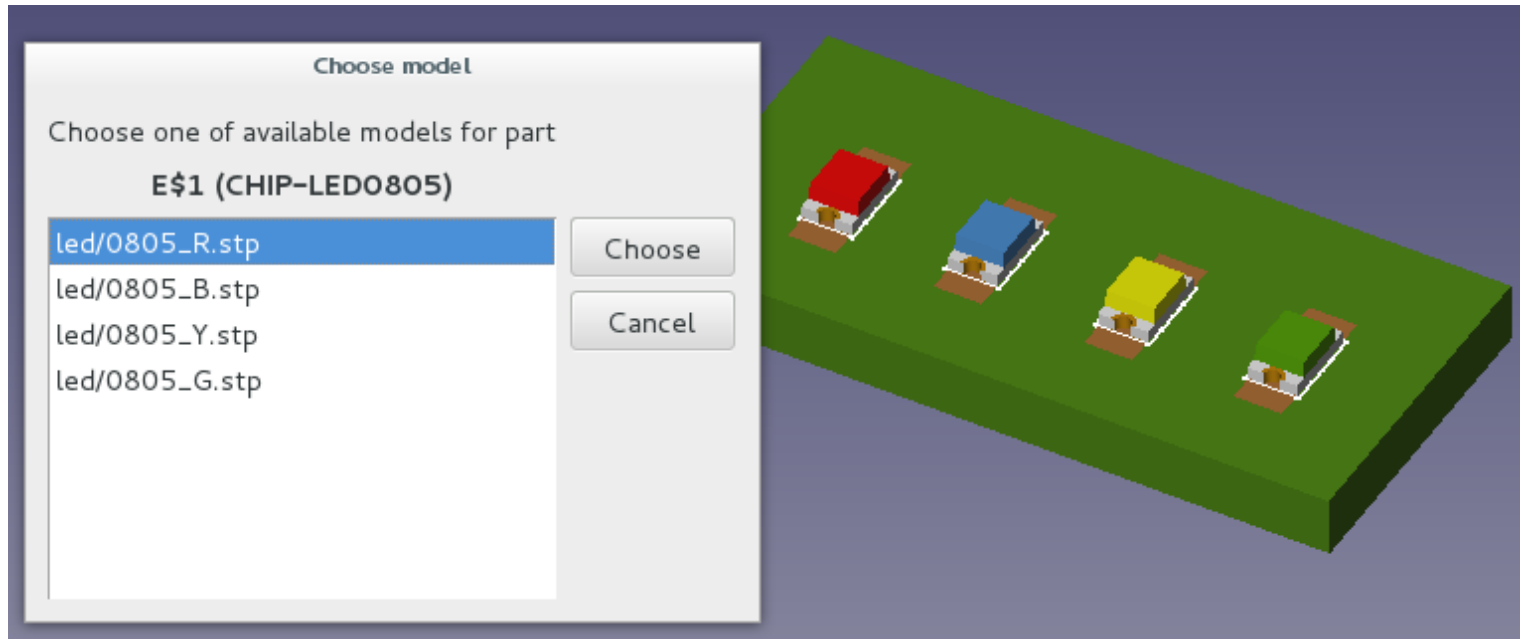
Multi model definition for one part

There is a possibility to set more than one 3D model for one package. To do this just split path to all models by character ';' in field 'Path to element'.



This function is useful for parts which only different is color – the same correction values are set for all models.

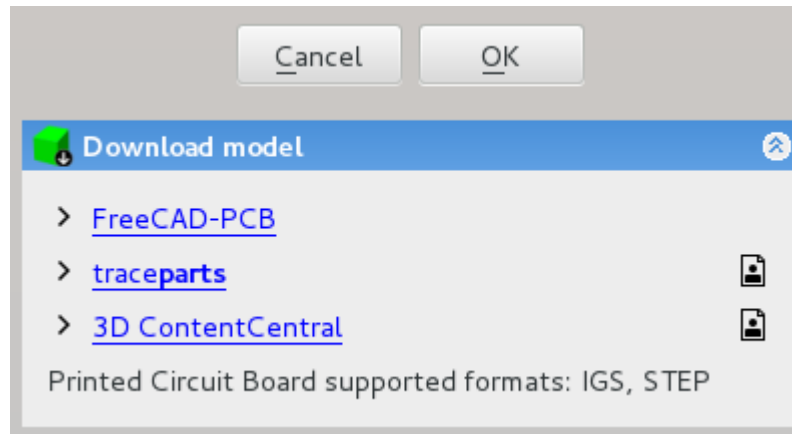
For packages where we set multi models, special window will appear during board loading or parts updating.



DOWNLOAD MODELS



Download model window appear in Task tab. Mentioned window contains links to sites when You can download for free 3D models.



Icons definition:



Registration is necessary to download models

Printed Circuit Board workbench supported formats:

- IGS,
- STEP - at this moment colours are supported only for step files saved/created in FreeCAD. So to avoid problems with colours create part in FreeCAD or open an existing component in FC (made in another program) and save it as step with colours.

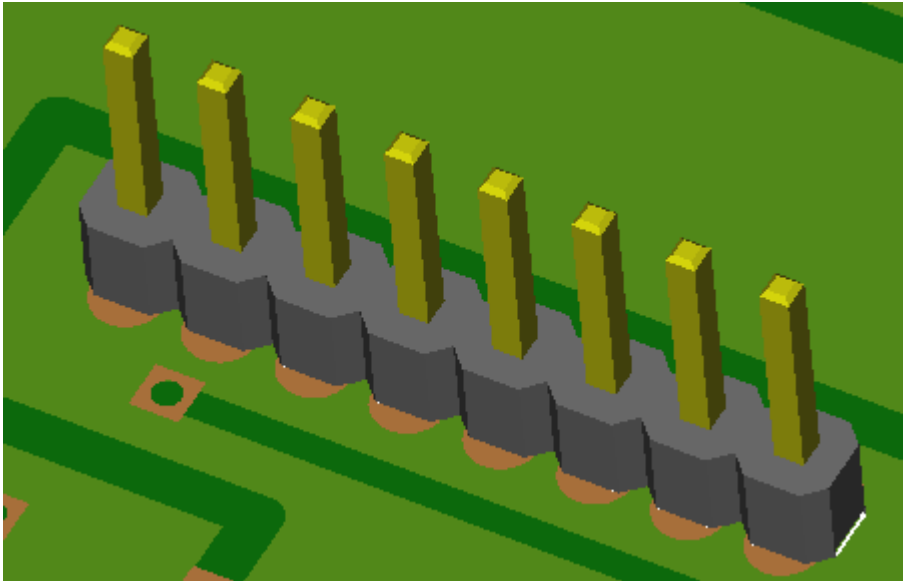
There is also possibility to search for concrete model. To do this just right click on missing model in specification tree and choose PCB model → Find model on-line.

DISPLAY MODES

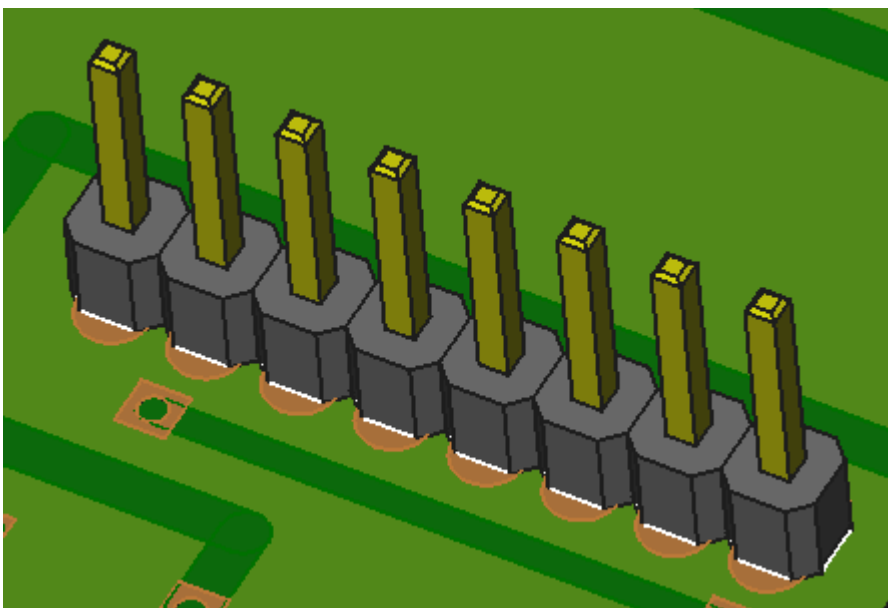
Display modes function can quickly and in easy way change display representation of shapes in project.

Available types:

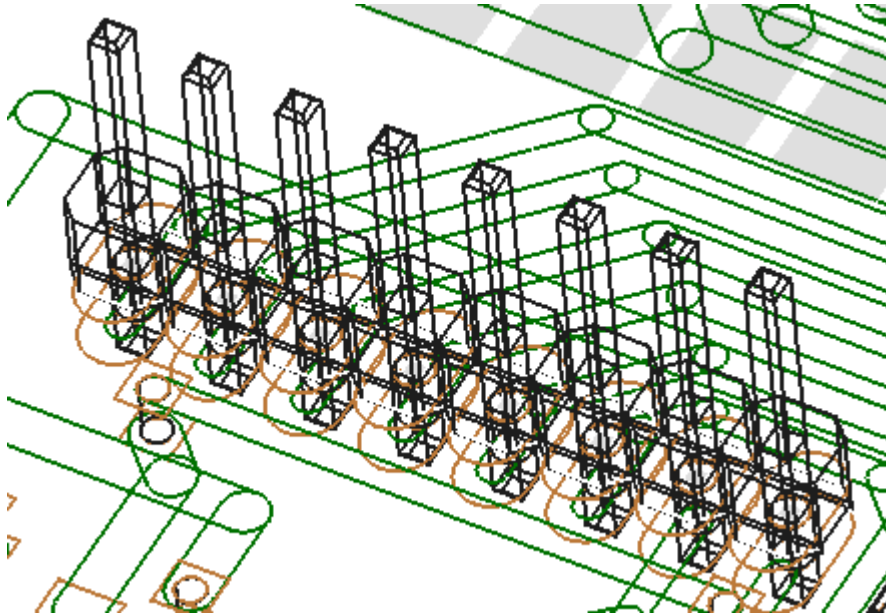
1. Shaded: border lines are hidden.



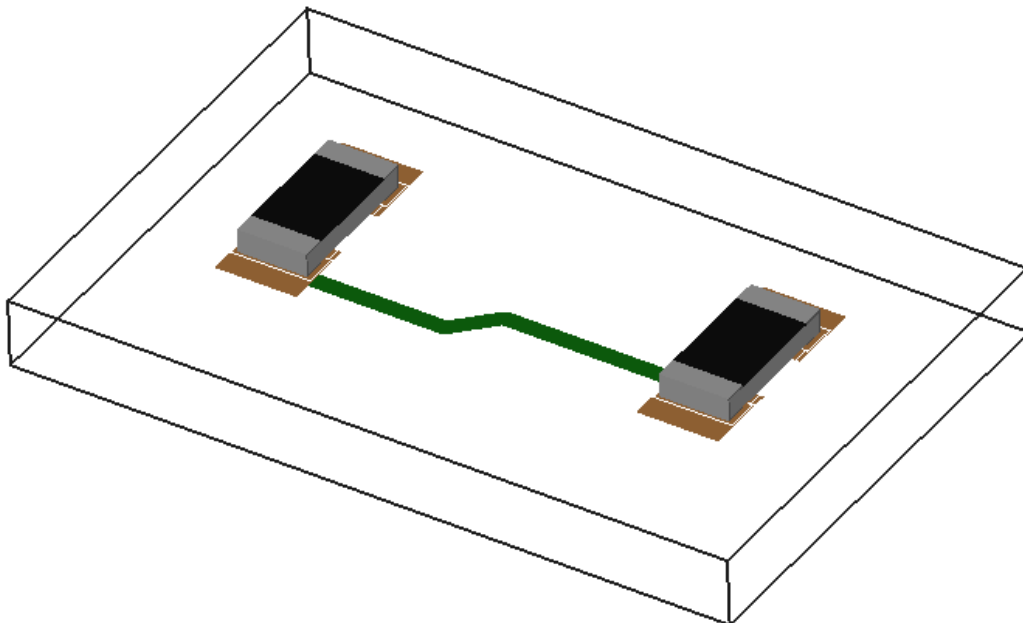
2. Flat lines: surfaces and border lines are displayed in one time.



3. Wireframe: only border lines are displayed.



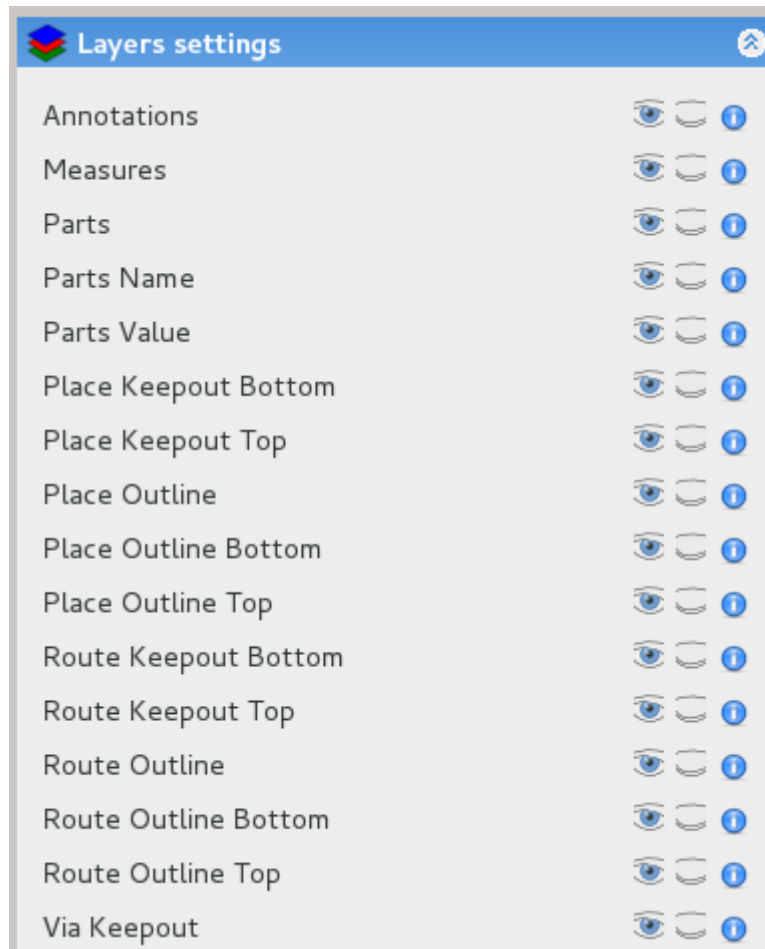
4. Internal View: for board only border lines are displayed, rest is displayed in Flat lines mode



LAYERS



Layers settings windows help in managing of currently displayed board layers. Layers settings window appear in Task tab.



Each line consists of four parts:

- Layer name,
- Button Show All – show all objects of this type,
- Button Hide All – hide all objects of this type,
- Information button – display information about layer.

Python

To manage layers by Python You need to import PCBlayers

```
import PCBlayers
```

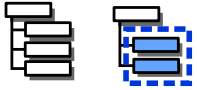
This module contain few functions:

1. `display(TypeId)`: display layer with a given ID
TypeId: specific list
2. `blank(TypeId)`: blank layer with a given ID
TypeId: specific list
3. `toggle(TypeId)`: toggle visibility state of layer with a given ID
TypeId: specific list
4. `state(TypeId)`: plirt state of layer with a given ID
TypeId: specific list
5. `list()`: return list of all available layers
'layerName: {'info': 'info', 'id': TypeId}
6. `listTXT()`: print all available layers

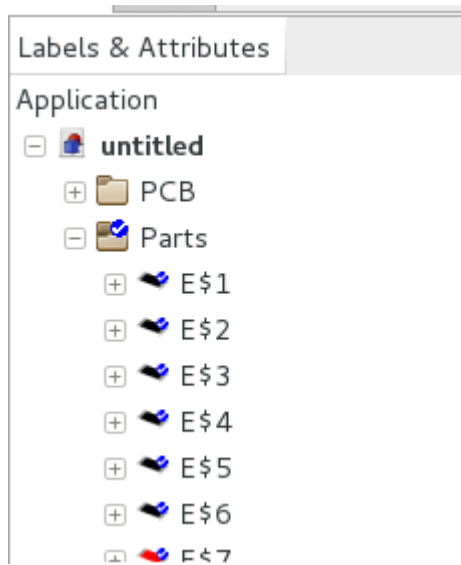
Example:

```
import PCBlayers as layer  
  
lay = layer.list()  
lay = lay["Annotations"]['id']  
  
layer.blank(lay)  
layer.state(lay) # False  
layer.toggle(lay)  
layer.state(lay) # True
```

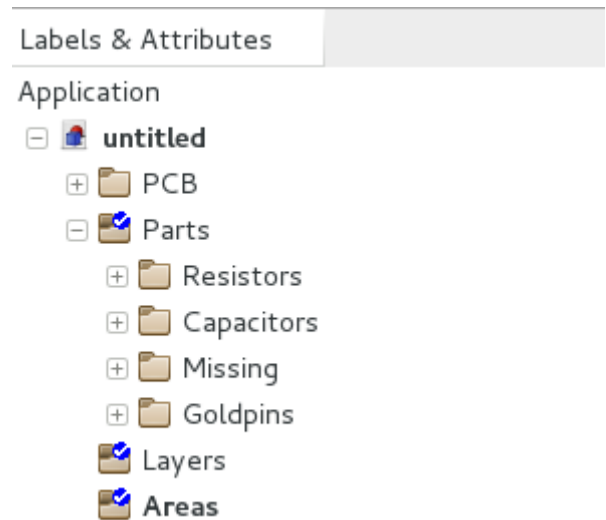
GROUPING PARTS



These options allow you to group/ungroup parts, according to Category they belong.



Ungrouped parts



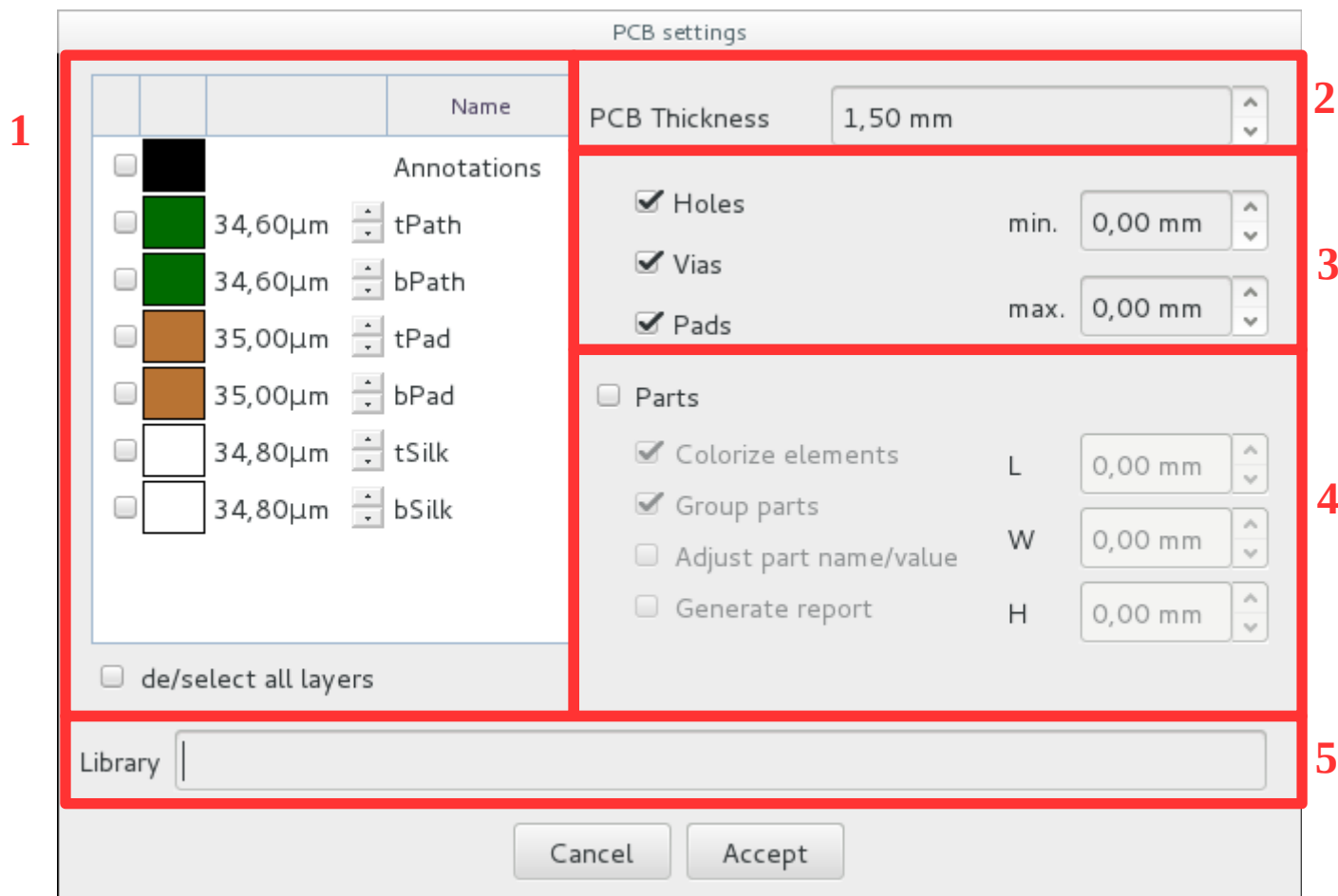
Grouped parts

These options are also available in:

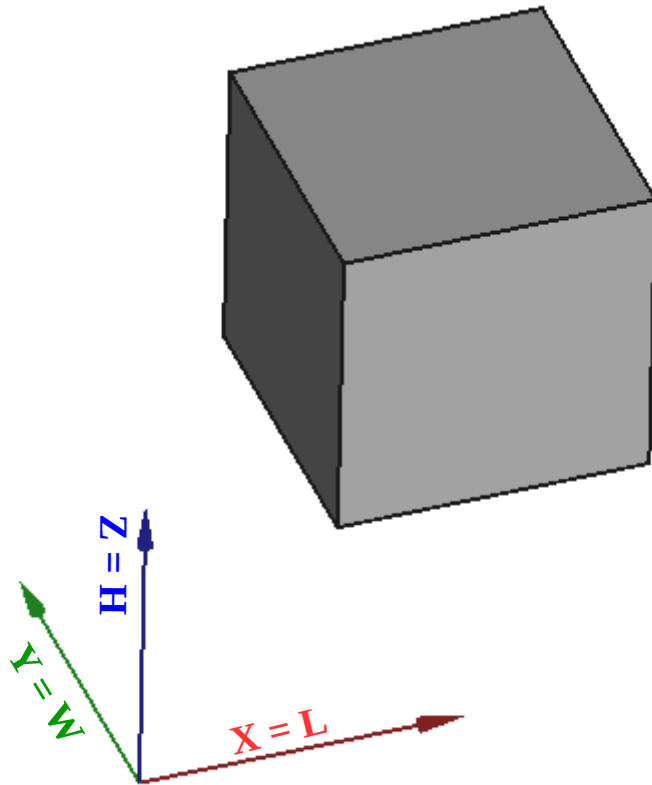
- open/import window,
- update parts window,
- add new model window.

OPEN/IMPORT BOARD

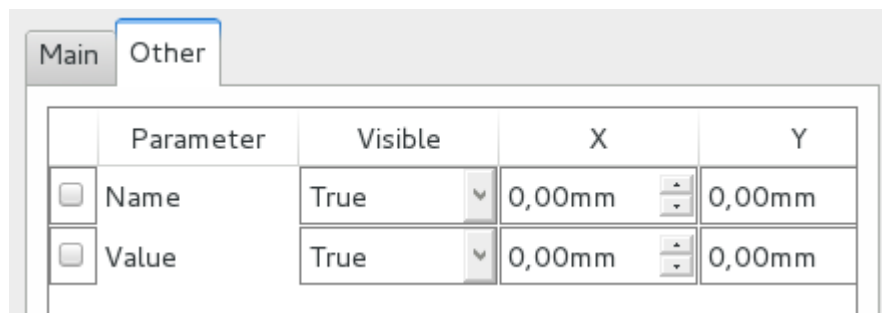
During open/import process special window will appear, in which we can set basic parameters of the board.



1. In first section You can choose, which layers will be loaded. Available layers depends from loading file type. Layer name and color are editable.
2. This section allow You to set PCB thickness. If file contain board thickness this value will be displayed in this field. Default value is 1.5[mm].
3. Third section contain basic settings about importing holes.
Here You can decide what type of holes You want to import (hole/vias/pads) and set imported holes diameter range (min/max). Both parameter can be set separately.
4. Fourth area contains basic settings about importing parts.
Here You can decide if You want to import parts, decide if they should contain colors, etc. Fields L/W/H allow You to decide about minimum length/width/height of 3D models which will be imported. All three parameter can be set separately.



Option 'Adjust part name/value' allows to automatic placing objects name/value in specific position. This option is connected with functions from 'Assign window'.



5. In some cases it is necessary to define path to specific program libraries.

Unit system

During board loading process units are changed to millimeters [mm].

Below You can find information, which part of imported files are supported by Printed Circuit Board workbench.

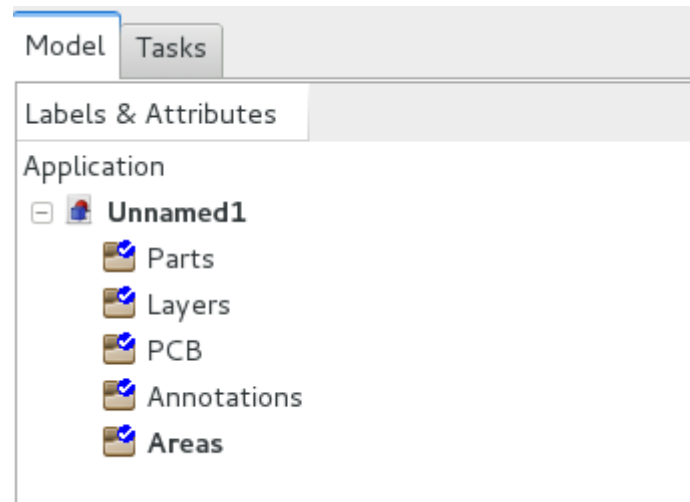
Soft name	PCB											
		Holes/Vias	Parts	Border	Measures	Soldermask	Keepout layers	Paths	Pads	Soldermask ARC	PCB round corners	Annotations
Eagle	brd	■	■	■	■	■	■	■	■	■	■	■
gEDA	pcb	■	■	■	■	■	■	■	■	■	■	■
FreePCB	fpc	■	■	■	■	■	■	■	■	■	■	■
KiCad	kicad_pcb	■	■	■	■	■	■	■	■	■	■	■
ZenitPCB	zpc	■	■	■	■	■	■	■	■	■	■	■
TARGET 3001!	t3001	■	■	■	■	■	■	■	■	■	■	■
FidoCadJ	fcd	■	■	■	■	■	■	■	■	■	■	■
SCOOTER		■	■	■	■	■	■	■	■	■	■	■
Razen	rzp	■	■	■	■	■	■	■	■	■	■	■
IDF v2	idf	■	■	■	■	■	■	■	■	■	■	■
IDF v3	idf	■	■	■	■	■	■	■	■	■	■	■
IDF v4		■	■	■	■	■	■	■	■	■	■	■
TurboPcb	apcb	■	■	■	■	■	■	■	■	■	■	■
DipTrace	asc	■	■	■	■	■	■	■	■	■	■	■
HyperLynx	HYP	■	■	■	■	■	■	■	■	■	■	■

■	Yes
■	No
■	Never
■	In progress
■	Future

CREATE NEW PROJECT



Before starting project it is recommended to press Create new project button. It will create necessary groups in Specification Tree used later by script.



Meaning of groups:

- Areas: group will store constraint areas,
- PCB: group will store board object,
- Layers: group will store layers,
- Parts: group will store all parts used in project,
- Annotations: group will store annotations.

Caution!



It is recommended to press Create new project button before starting project, but it is not necessary – script will check if specific group exist and, if desired will create it.

Python

To manage groups by Python You need to import PCBgroups module

```
import PCBgroups
```

This module contain few functions:

1. createGroup_Parts(): create Parts group.
2. createGroup_Layers(): create Layers group.
3. createGroup_Annotations(): create Annotations group.
4. createGroup_Areas(): create Areas group.
5. createGroup_PCB(): create PCB group.
6. setProject(): create all available groups.

Example:

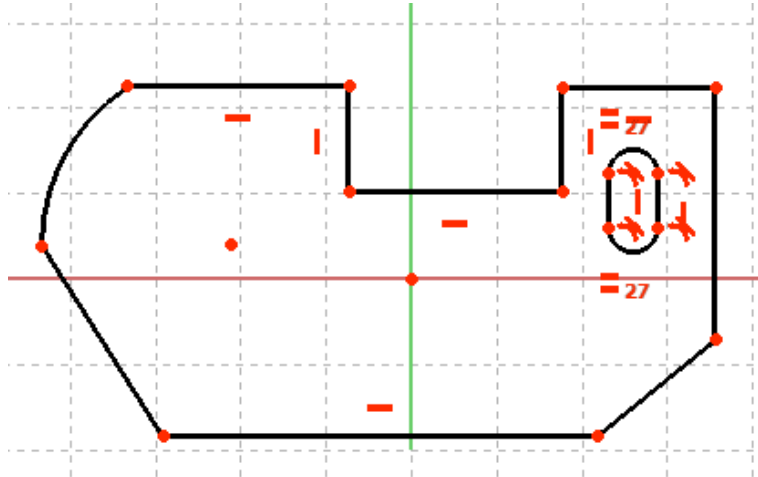
```
import PCBgroups as group  
pcbG = group.createGroup_PCB()  
pcbG.Label() # u'PCB'
```

CREATE PCB

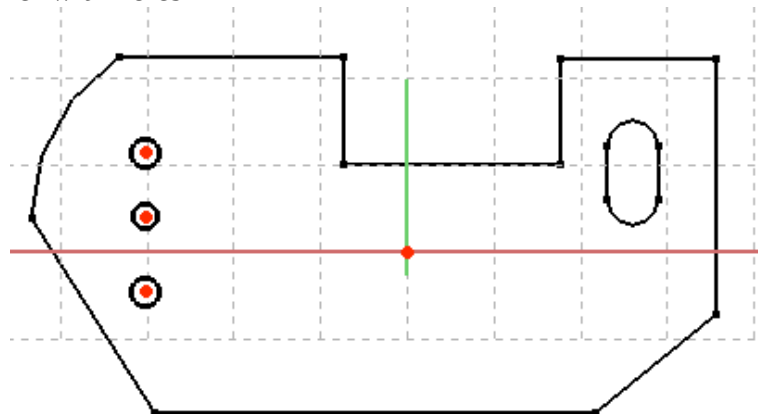


This task explains how to create a board and design its geometry from scratch.

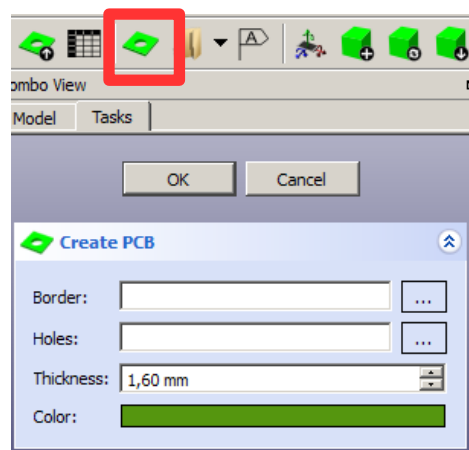
1. Create Sketcher with contour of the board



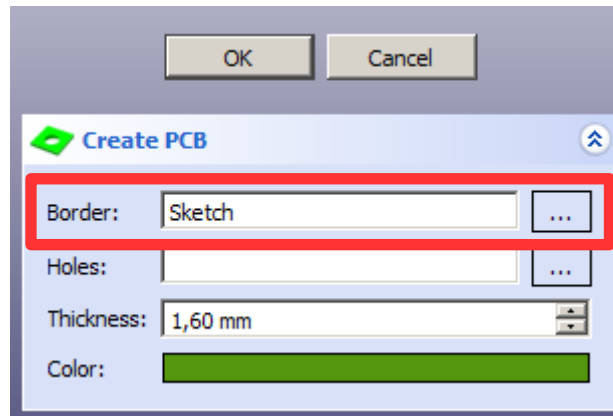
2. Create new Sketcher with holes



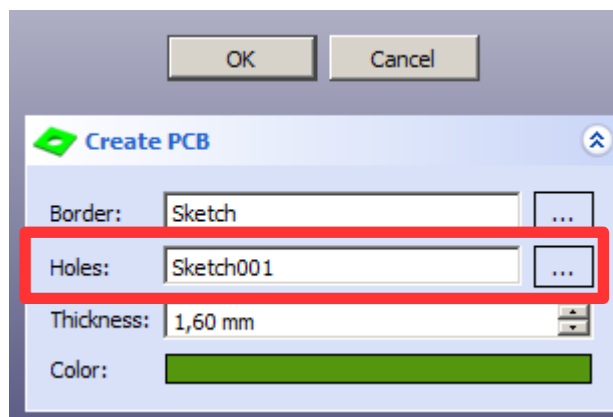
3. Click the **Create PCB** button



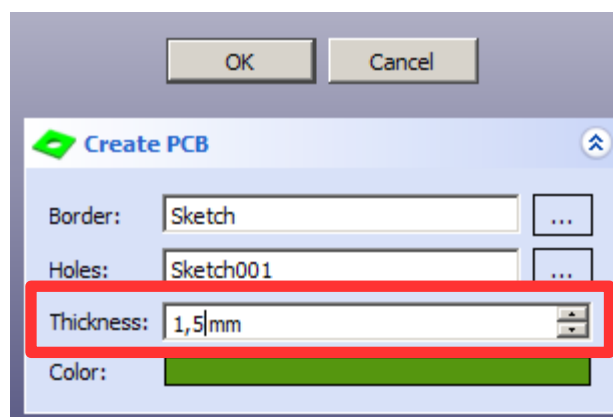
4. As **Border** point to Sketcher with contour of the board



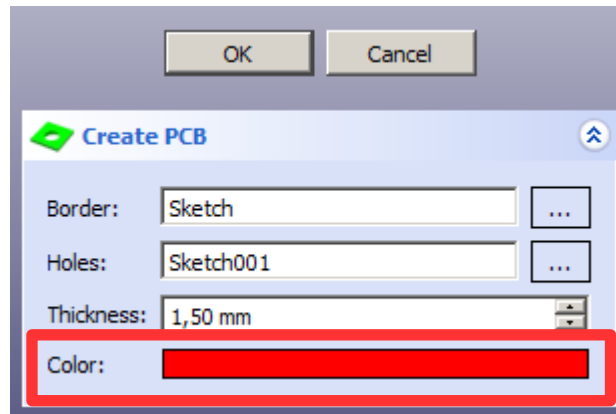
5. As **Holes** point to Sketcher with holes



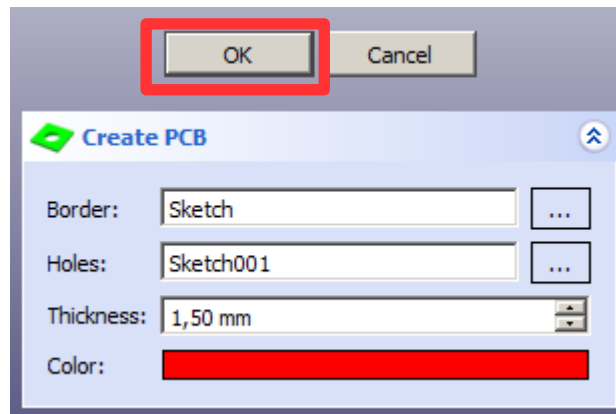
6. Set board **Thickness**



7. Set board **Color**



8. Click **OK** button to finish



The board should be generated according to settings.
The board has been created from scratch.



Caution!

Only one board can be generated per project

Caution!

Even if there will be no holes in board, proper Sketcher need to be done

Caution!

It is not possible to create board from a previously created part

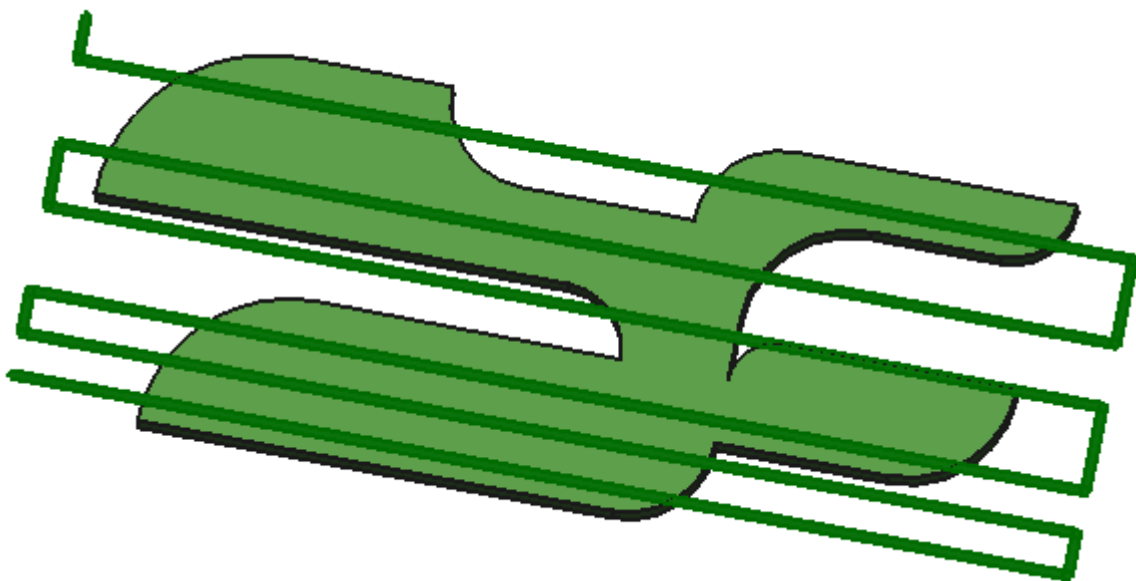
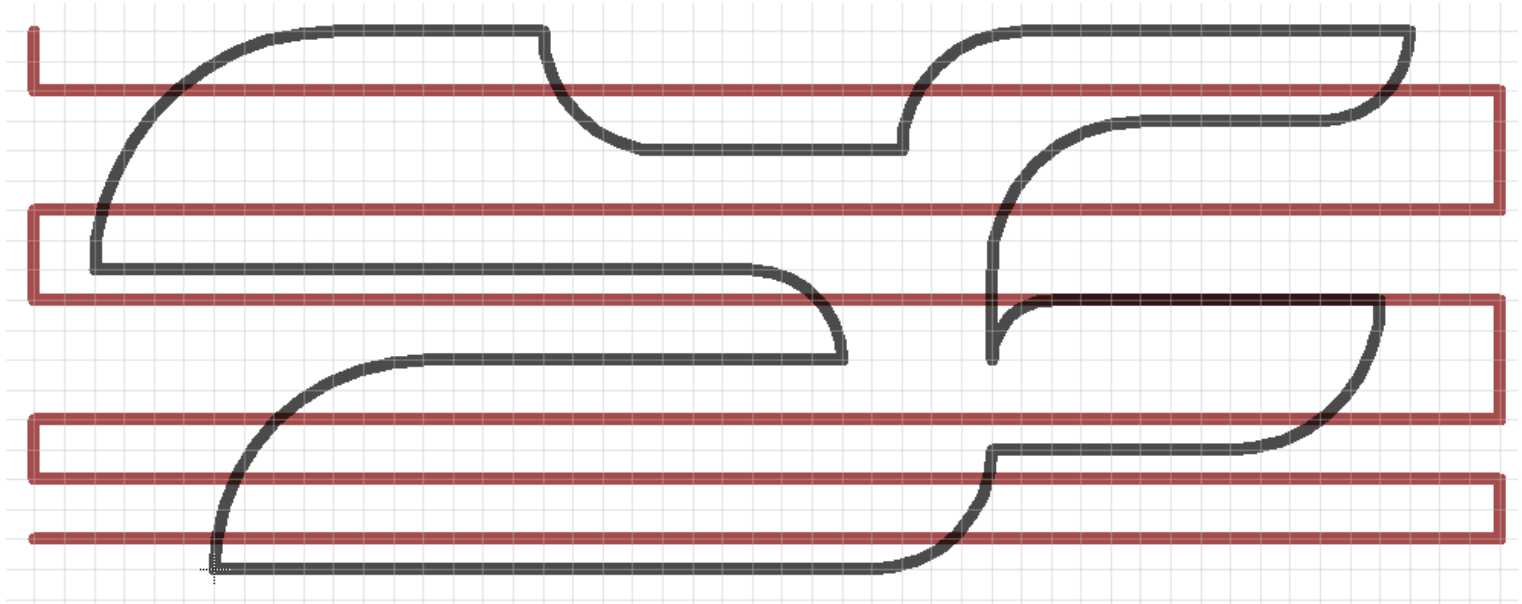
CUT TO BOARD OUTLINE

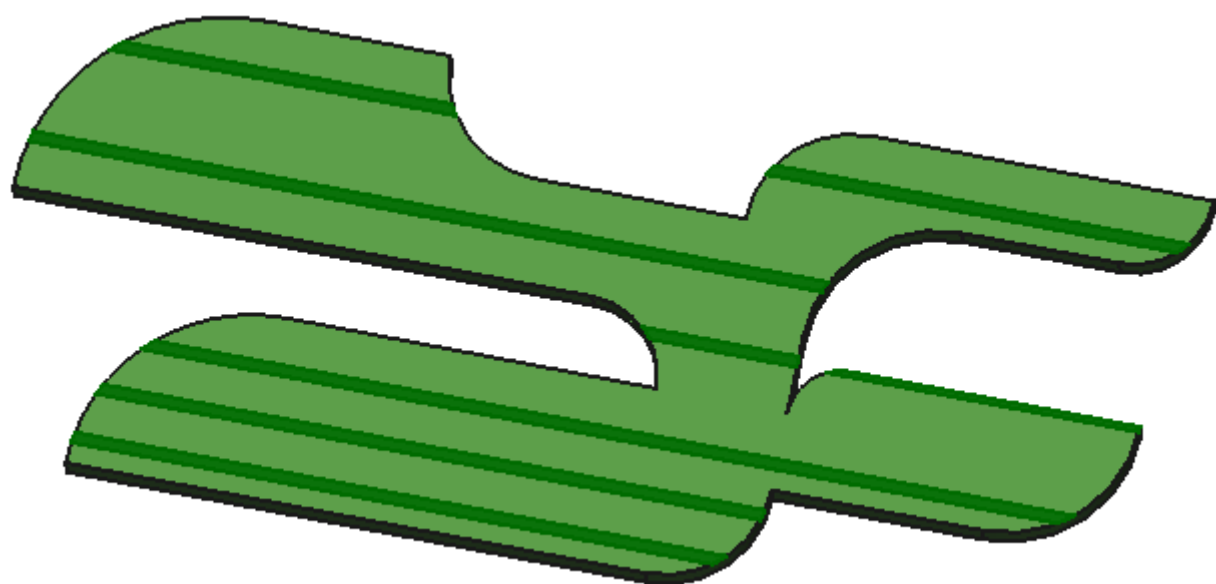


board.

Sometimes it is necessary to display board like it will look after manufacturing. To do this just use option 'Cut to Board Outline'. Function will automatically blank/display all layers/paths that are outside of the

Example





CREATE GLUE PATH



ADD ANNOTATION



This task shows how to add annotations to project.

1. Click button **Add annotation** on PCB Settings toolbar

2. Set text for the annotation

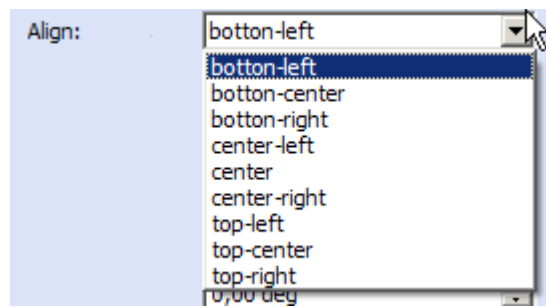
3. Set font size (in mm)

4. Set font color

To change color click on colored area.

5. Set align

There is possibility to align text, according to object center.



Align: bottom-left (X=0, Y=0)



Align: center (X=0, Y=0)



Caution!



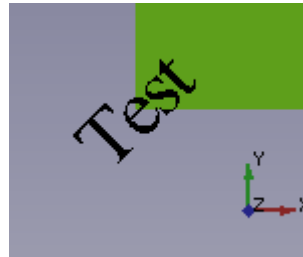
Align parameter set also point around which text will be mirrored/rotated.

6. Mirror text

Available values:

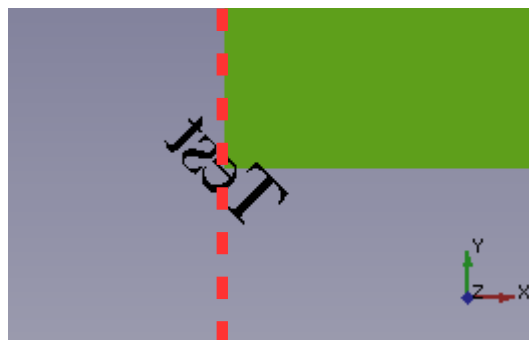
- None: do not mirror annotation,

Align: center, Rot: 45[deg]



- Global Y axis: mirror annotation according global Y axis (axis orientation is taken from 3D view space)

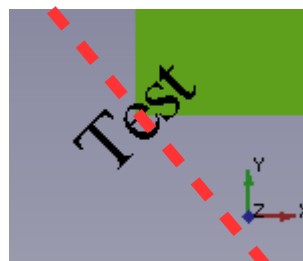
Align: center, Rot: 45[deg]



— Mirror axis

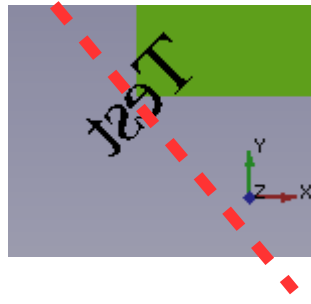
- Local Y axis: mirror annotation according local Y axis (axis orientation is taken from object)

Align: center, Rot: 45[deg], **Mirror: None**



◆ Local Y axis orientation

Align: center, Rot: 45[deg], **Mirror: Local Y axis**



Mirror axis

7. Spin text

If spin parameter is set to True (default) the texts can be displayed in any rotation, also upside down, otherwise (Spin = False) text is displayed readable (no upside down).

Align: bottom-left, Rot: 120[deg], Mirror: None, **Spin: True**



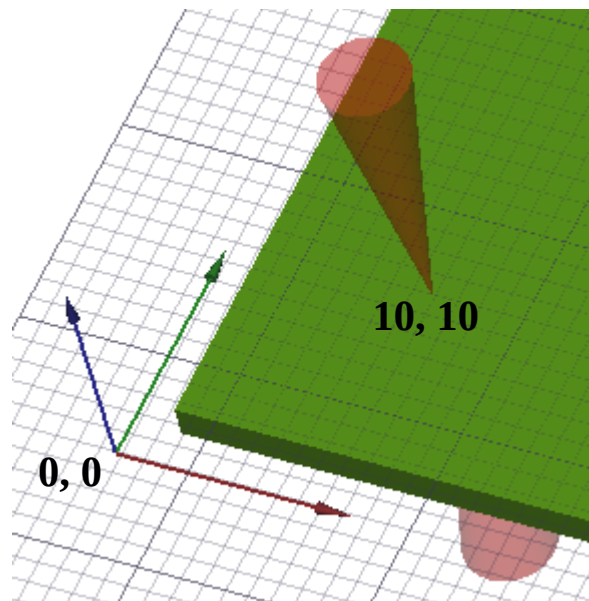
Align: bottom-left, Rot: 120[deg], Mirror: None, **Spin: False**



8. Set values for X, Y coordinates

X:	<input type="text" value="10,00 mm"/>
Y:	<input type="text" value="10,00 mm"/>

Distance is measured from 0, 0 in 3D window. Actual position (model center) is representing in 3D view by red 'arrow'



9. Set rotation value (rotation around Z axis)

Rotation:	<input type="text" value="45,00 deg"/>
-----------	--

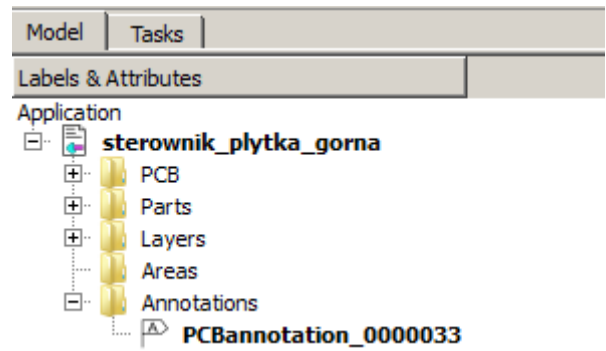
10. Choose side for new annotation on board

Side:	<input type="text" value="TOP"/>
-------	----------------------------------

**Caution!**

Component will not be automatically rotated according to chosen side.

11. Click Ok button to finish



Caution!



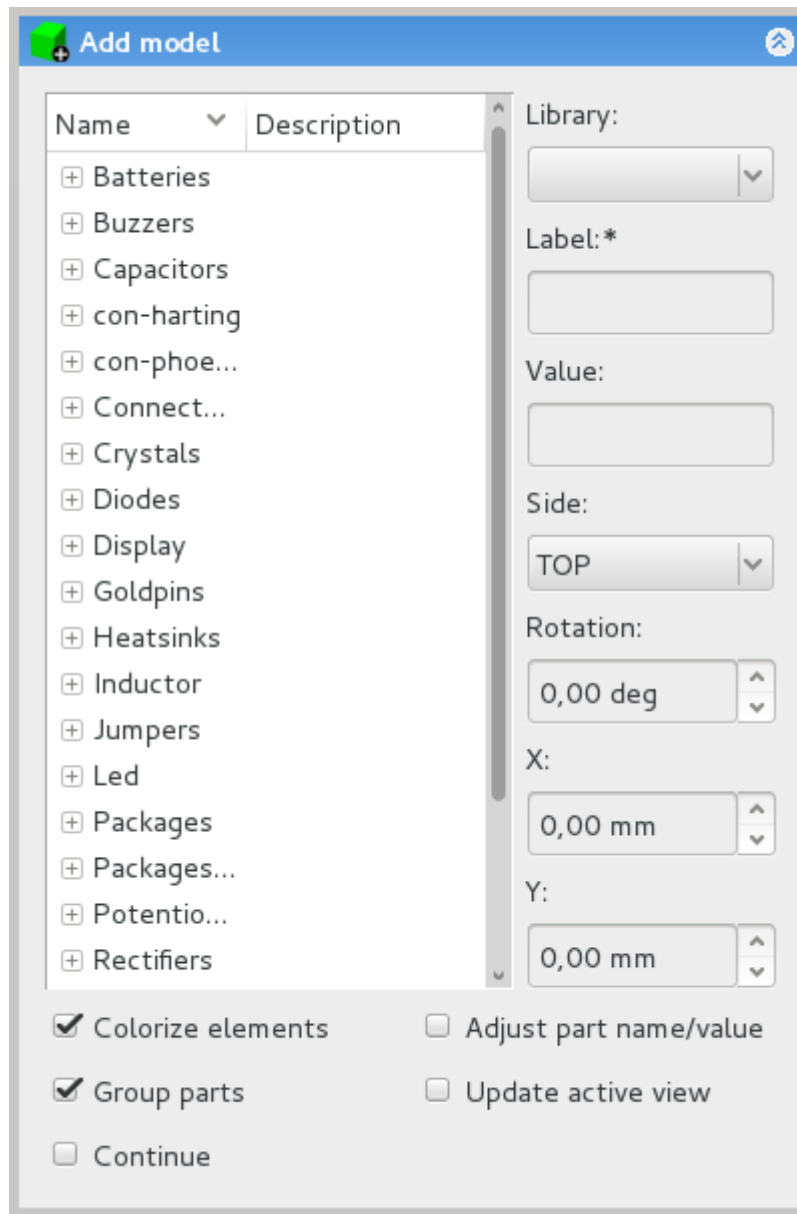
If not existing, special group (Annotations) will be created.

ADD MODEL

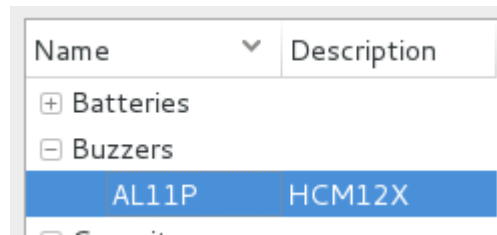


This task explains how to add new component to existing board.

1. Click button **Add model**



2. Select package – model type



3. Choose from drop-down list, from which library script should take settings



4. Write component name

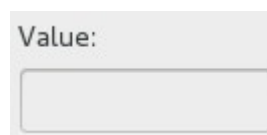


Caution!

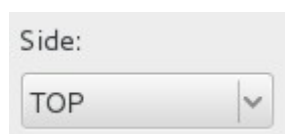


This field is mandatory

5. Add value



6. Choose side for new component on board



Component will be automatically rotated according to chosen side.

7. Set rotation value (rotation around Z axis)

Rotation:

45,00 deg

Value is in degrees.

8. Set values for X, Y coordinates

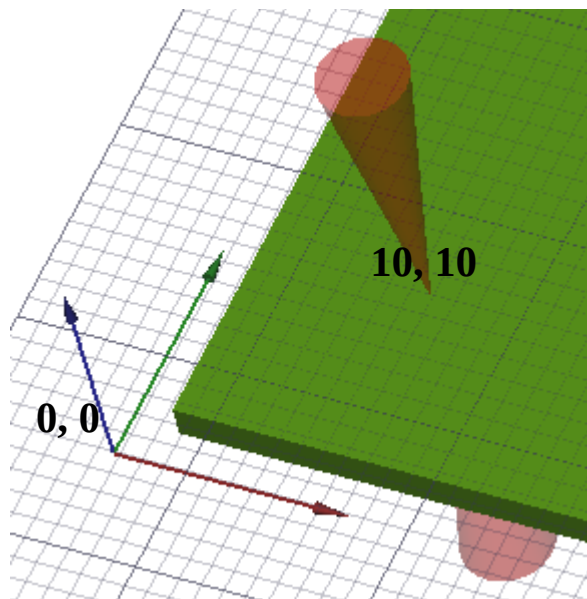
X:

10,00 mm

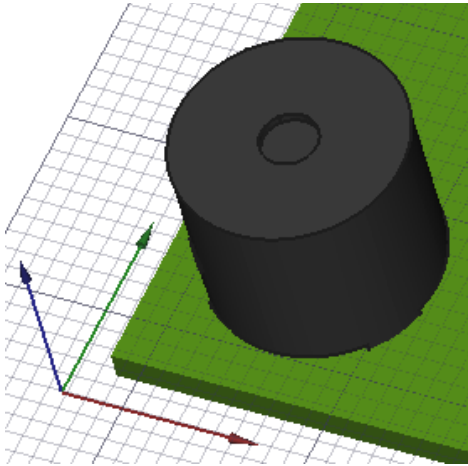
Y:

10,00 mm

Distance is measured from 0, 0 in 3D window. Actual position (model center) is representing in 3D view by red 'arrow'

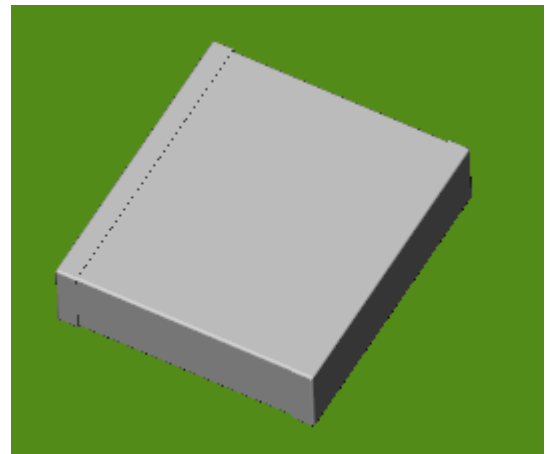
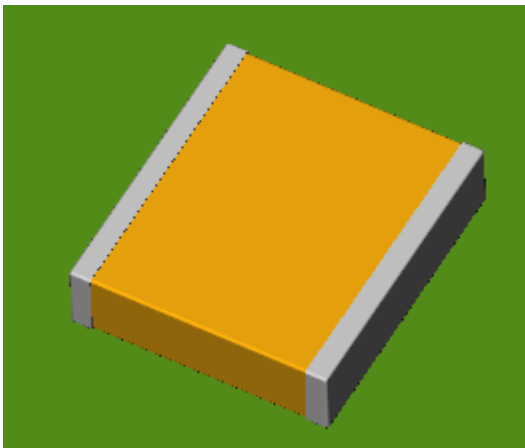


9. Click Ok button to finish



Add model tab contain five configuration options:

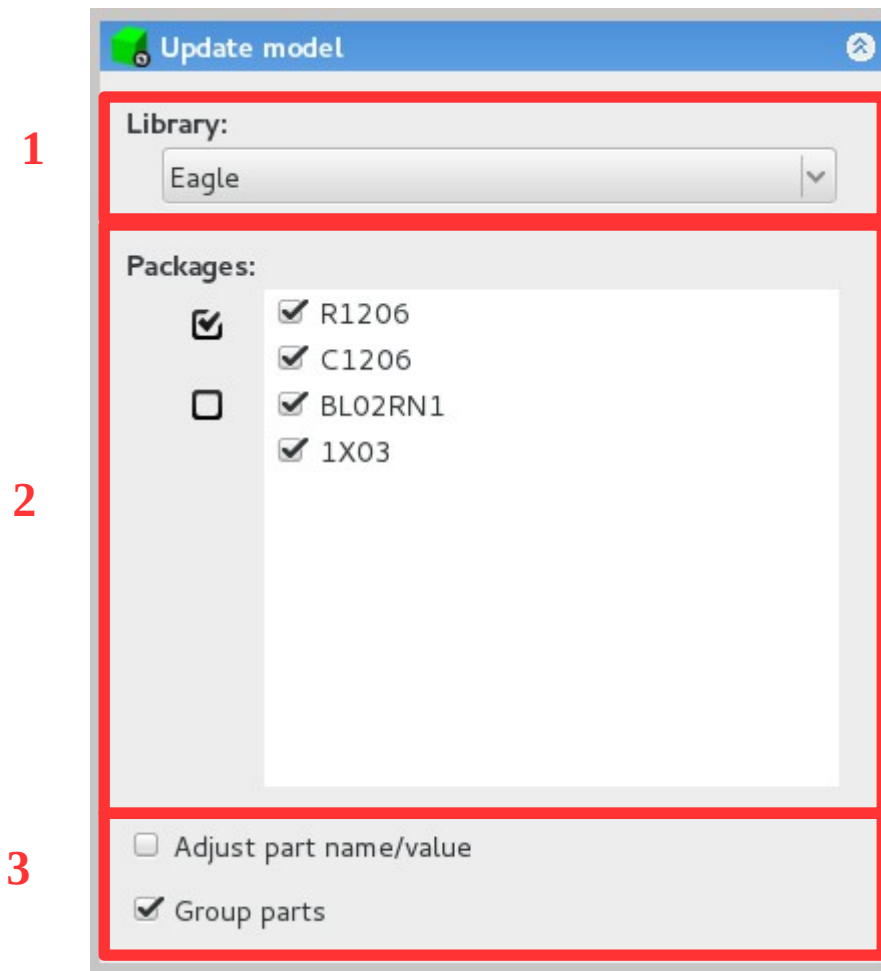
1. Colorize elements: there is possibility to add models in two modes – with colors and without.



2. Adjust part name/value – set Name/Value annotation values according to settings set in database.
3. Update active view: view in 3D window will automatically switches between TOP/BOTTOM view, dependency which side will be chosen.
4. Group parts: grouping parts in tree according to Categories.
5. Continue: normally after click Ok button Add modal window disappears, to avoid that (You want to add more than one object) just mark this option.

UPDATE MODELS

 Update models window will reload/load 3D model/settings for used in project components.



Update models tab contain three sections:

1. Library: during update process, script will search settings (eg. X, Y, Z values) in specific library,
2. Packages: contain listbox with used in project components. Checked checkbox next to model type mean that this part will be updated.
3. Configuration options:
 - Adjust part name/value – set Name/Value annotation values according to settings set in database.
 - Group parts: grouping parts in tree according to Categories.

Caution!

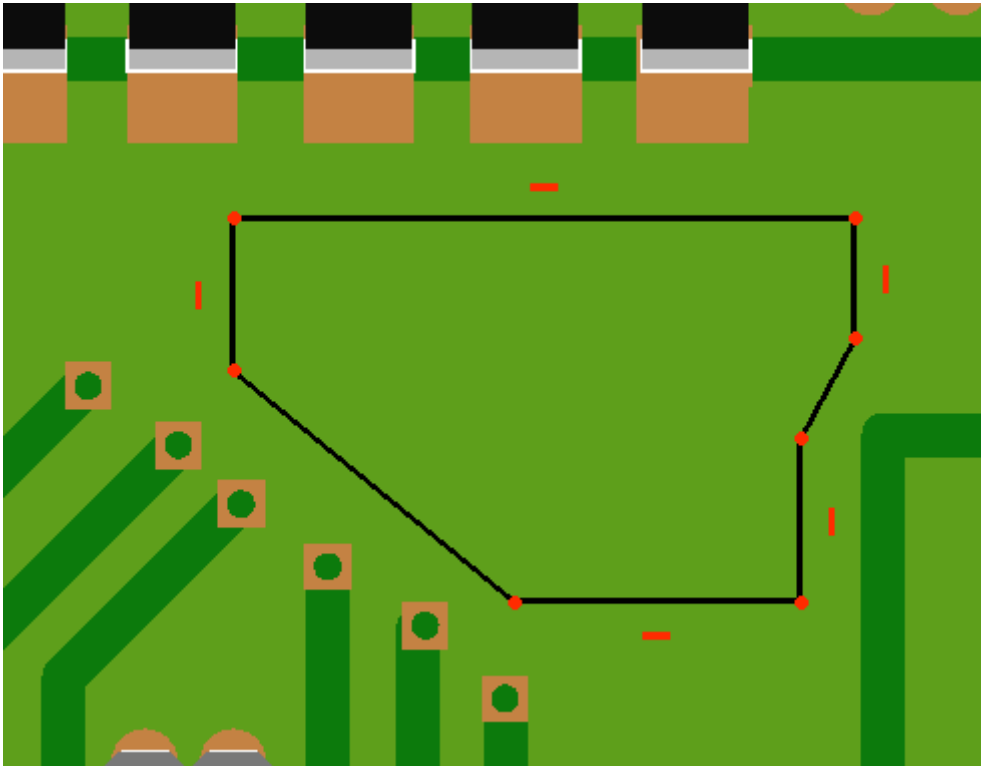
When selected component does not appear in specified library, model will be not updated.

CREATE CONSTRAINT AREA



This task shows you how to create a constraint area. A constraint area is a 'object' represent area reservation for different purposes.

1. Create Sketcher with contour of the constraint area

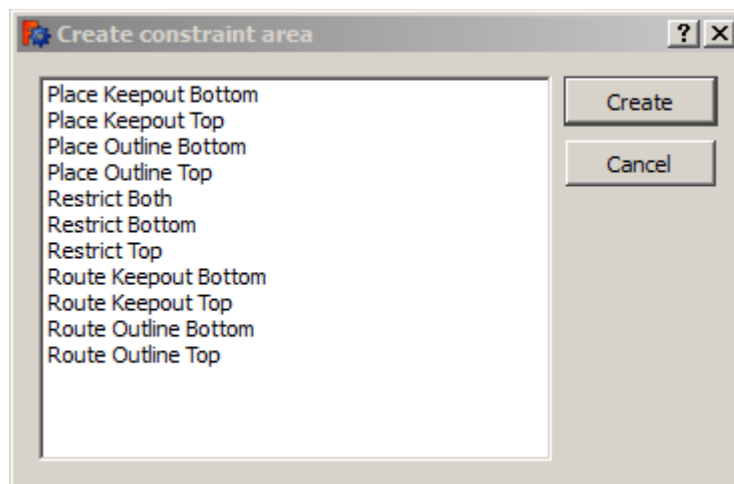


Caution!



Shape created in Sketcher need to be closed.

2. Select just created sketcher and click Create Constraint area button

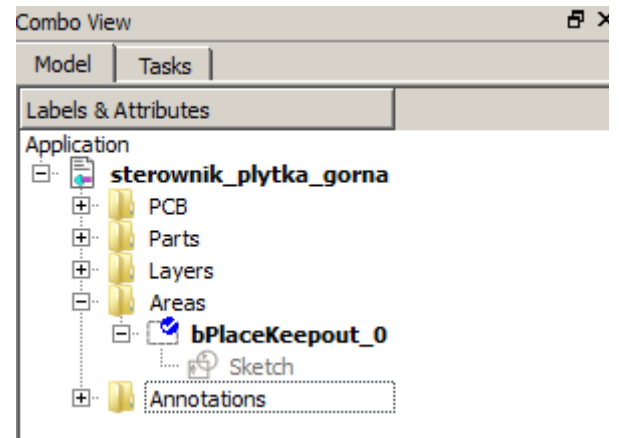
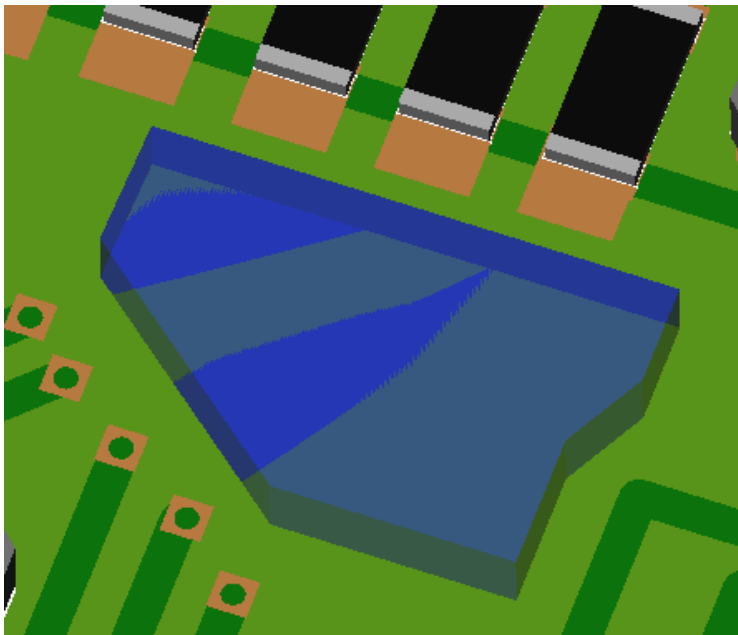


3. Choose constraint area type

Available constraint area types:

- Place Keepout Bottom:
- Place Keepout Top:
- Place Outline Bottom:
- Place Outline Top:
- Restrict Both:
- Restrict Top:
- Restrict Bottom:
- Route Keepout Bottom:
- Route Keepout Top:
- Route Outline Bottom:
- Route Outline Top:

4. Click **OK** button to finish



Caution!



If not existing, special group (Areas) will be created.

EXPLODE



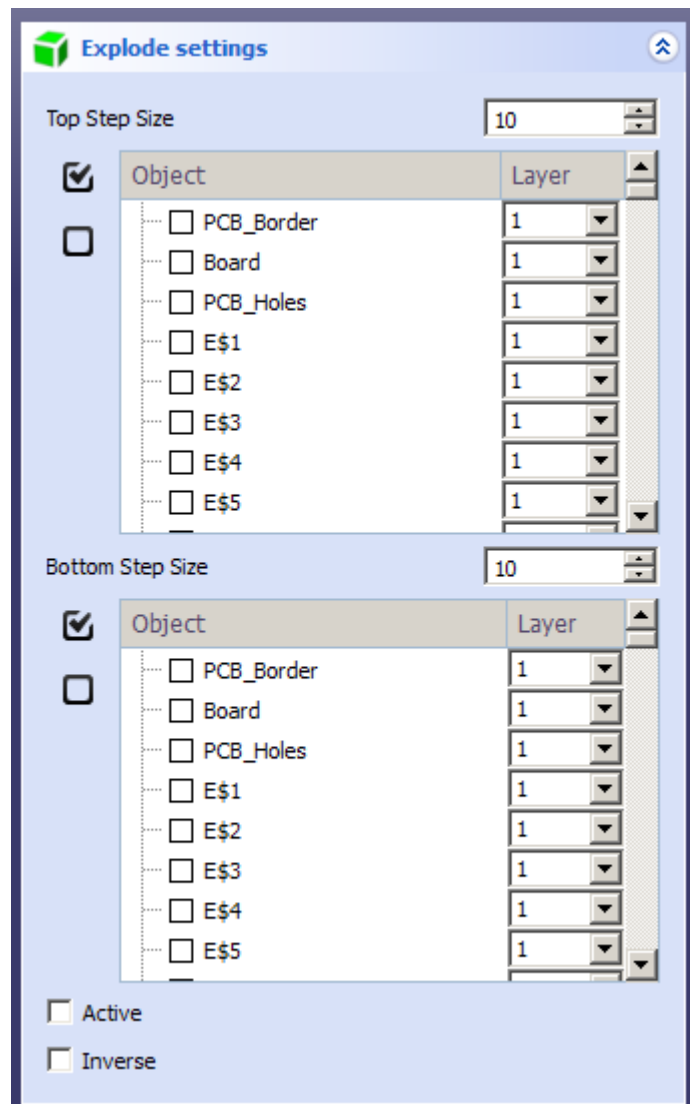
Explode function allows for assemblies to be quickly disassembled and shown in an exploded view.

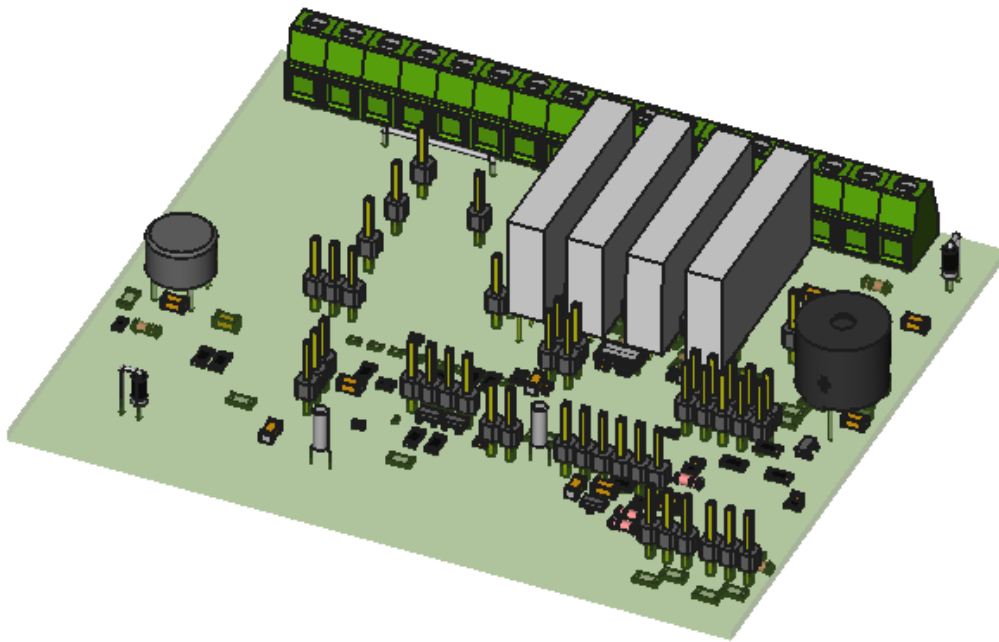
Explode contains two similar functions:

1. Fast explode: automatically explode all printed circuit board parts.
2. Explode: user selects which parts and how will be relocated.

This task shows you how to manually create a explode object.

1. Load PCB and click Explode button





2. Set Top step size

Top Step Size	10
---------------	----

Top step size determines distance between next levels.

3. Choose parts – top side

<input checked="" type="checkbox"/>	Object	Layer
<input type="checkbox"/>	PCB_Bor...	1
<input type="checkbox"/>	Board	1
<input type="checkbox"/>	PCB_Holes	1
<input checked="" type="checkbox"/>	E\$1	1
<input checked="" type="checkbox"/>	E\$2	2
<input checked="" type="checkbox"/>	E\$3	1
<input type="checkbox"/>	E\$4	1
<input type="checkbox"/>	E\$5	1

Selected parts will automatically disappear from the bottom list.

Layer parameter is a number, that will be multiplied by Top step size to determine new Z position for part after explode

4. Set Bottom step size

Bottom Step Size

Bottom step size determines distance between next levels.

5. Choose parts – bottom side

<input checked="" type="checkbox"/>	Object	Layer
<input type="checkbox"/>	E\$91	1
<input type="checkbox"/>	E\$92	1
<input checked="" type="checkbox"/>	E\$100	1
<input checked="" type="checkbox"/>	E\$108	1
<input checked="" type="checkbox"/>	E\$109	6
<input checked="" type="checkbox"/>	E\$111	3
<input checked="" type="checkbox"/>	E\$113	1
<input type="checkbox"/>	E\$116	1

Selected parts will automatically disappear from the top list.

Layer parameter is a number, that will be multiplied by Bottom step size to determine new Z position for part after explode.

6. Active

☐ Active

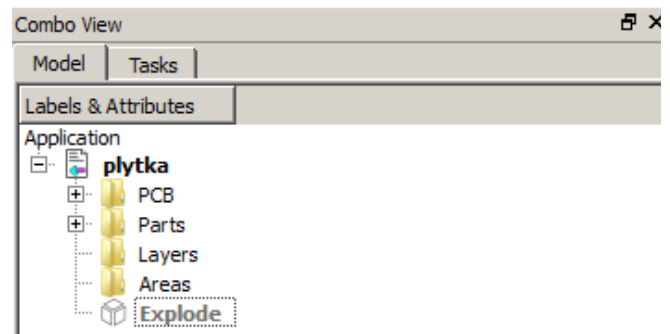
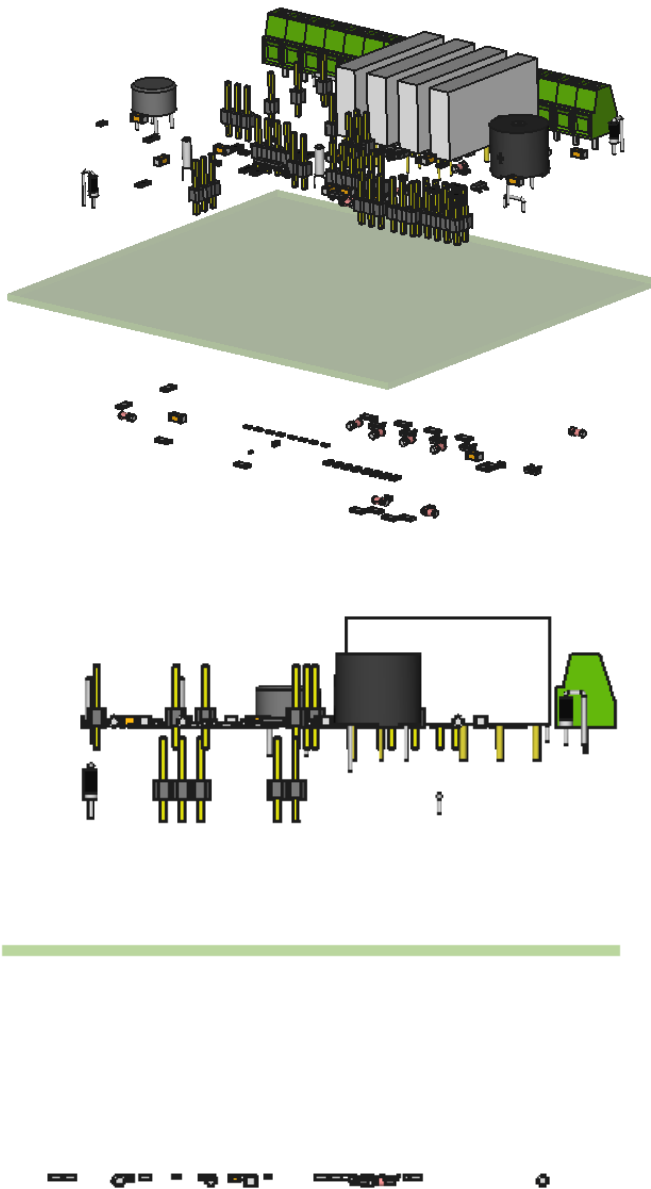
If this option will be checked explode object will be activated automatically after creation.

7. Inverse

☐ Inverse

To reverse top and bottom side check this option.

8. Click Ok button to finish



Caution!



Fast explode function automatically splits parts between top and bottom layers.

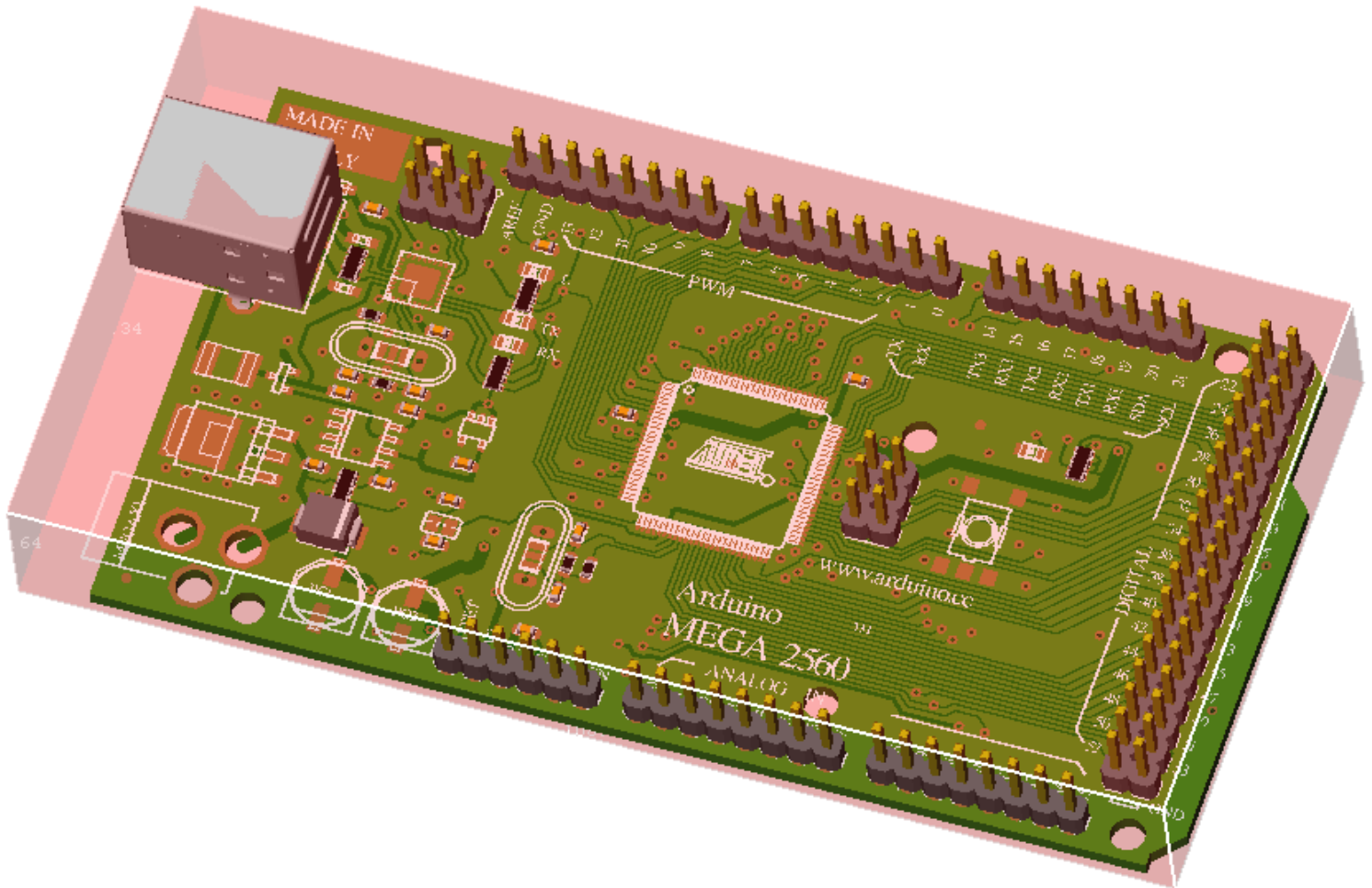
BOUNDING BOX

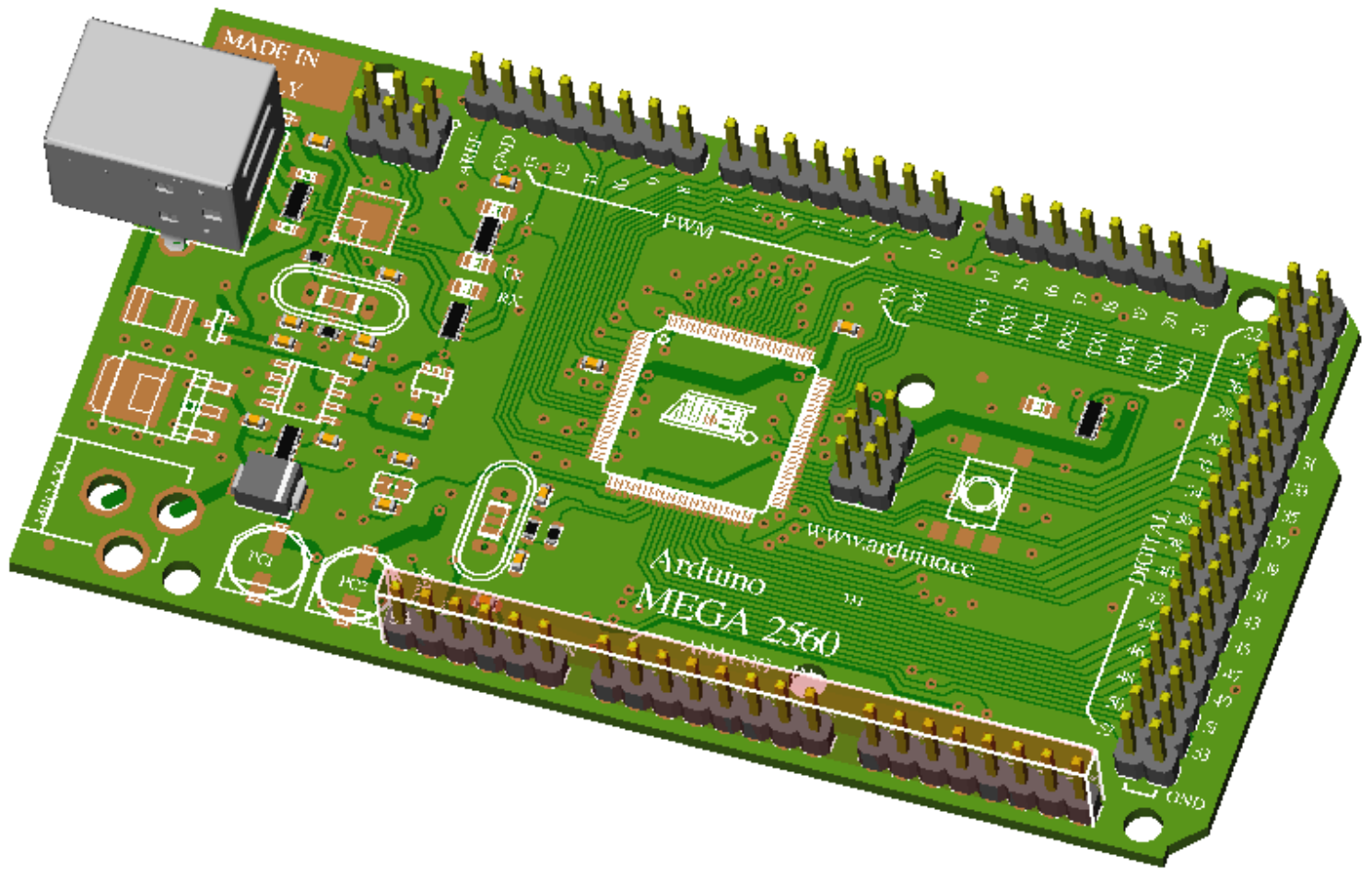


Bounding box is a smallest cuboid completely surrounds the object.

Printed Circuit Board workbench contain two function to generate bounding box:

- Bounding box – generate box for all board (board, parts, paths),
- Bounding box from selection – generate box for selected components.





Generated boxes are normal cubes so it is possible to work on them in FreeCAD.

You can generate as many bounding boxes, as You need.

Python

To create bounding box by Python You need to import PCBboundingBox

```
import PCBboundingBox
```

This module contain few functions:

1. boundingBox(): create bounding box for whole board.
2. boundingBoxFromSelection(): create bounding box for selected objects.

Example:

```
import PCBboundingBox as box  
  
box.boundingBox()
```

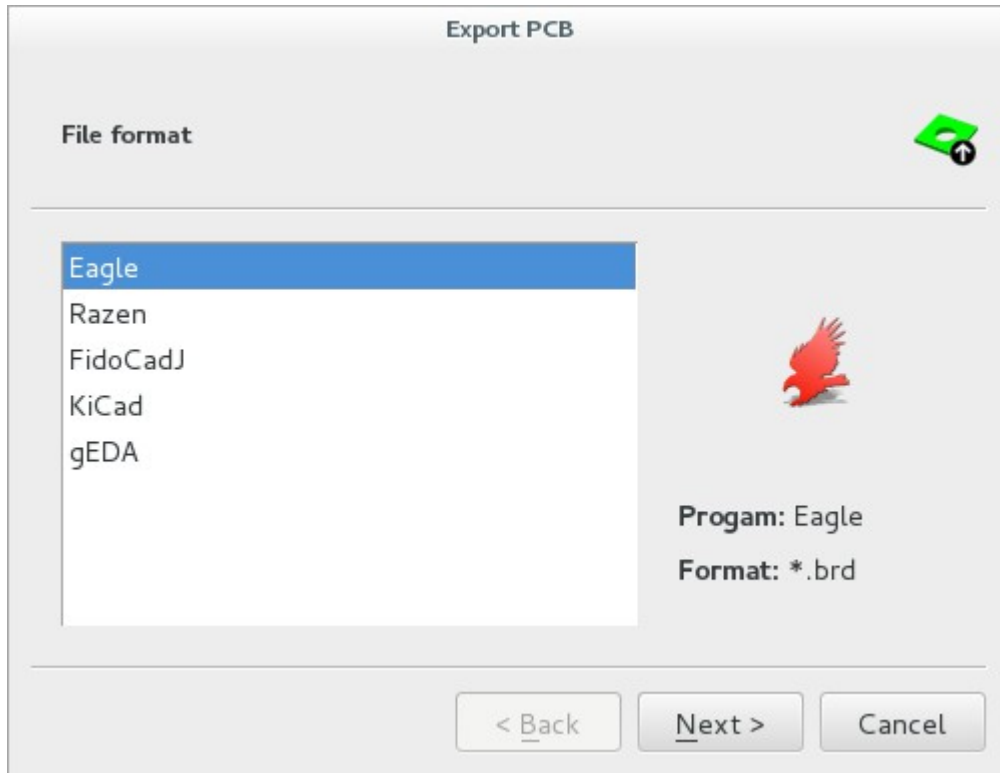
Example 2:

```
import PCBboundingBox as box  
  
a = box.boundingBox()  
a.Length # 179.546 mm  
a.Width # 104.27 mm  
a.Height # 2.155 mm
```

EXPORT BOARD



Export option allow You to save created/modified board in FreeCAD to one of supported file formats.

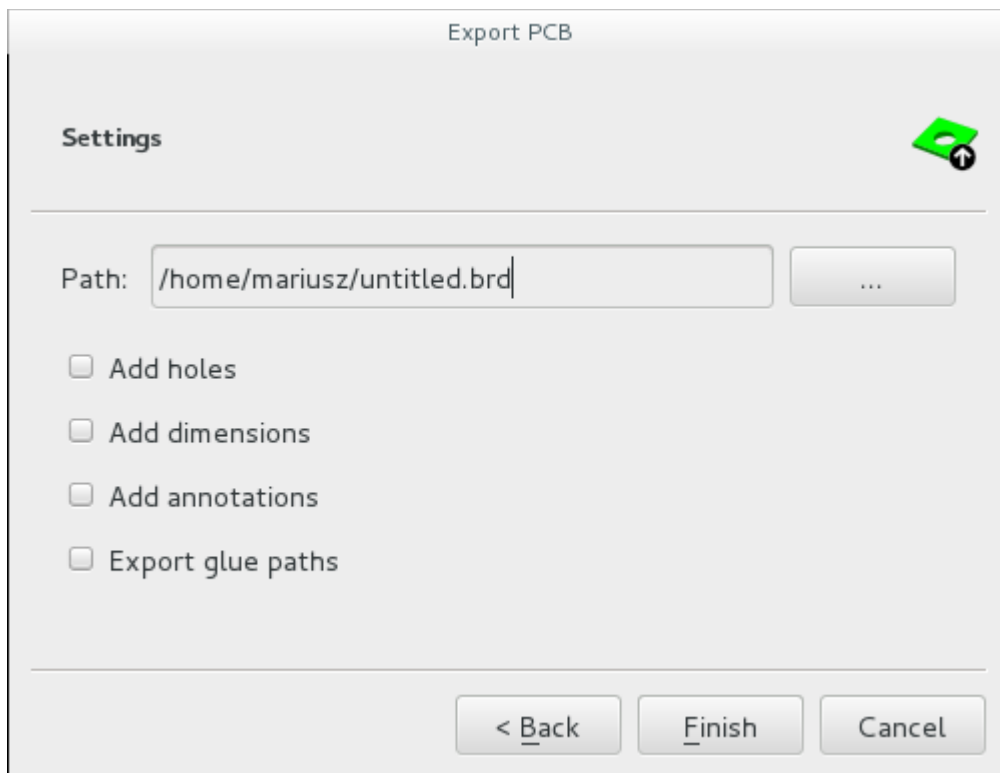


First tab in export window allow You to choose export file format.

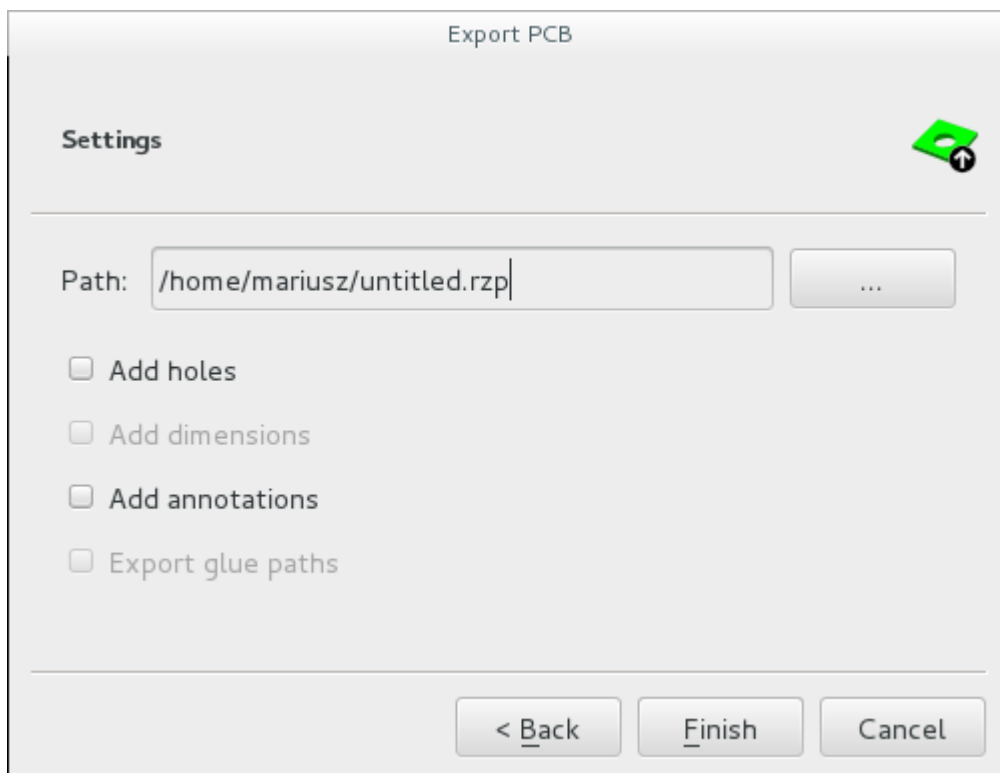
Supported files

- Eagle (*.brd),
- FidoCadJ (*.fcd),
- KiCad (*.kicad_pcb),
- gEDA (*.pcb),
- Razen (*.rzp).

After choosing file format click Next to move to settings section where You can set which parts of PCB will be exported and where to save new file.



Available options depends of selected file format.



Caution!

Default settings you can set in Preferences – tab '[Customizing Workbench](#)'.

Clicking Finish button will end Export process and script will create new file according to chosen settings.

Below You can find information, which parts of board are supported by Printed Circuit Board workbench during export process.

Soft name		PCB					
		Border	Measures	PCB round corners	Holes	Annotations	
Eagle	brd	■	■	■	■	■	■
KiCad	kicad_pcb	■	■	■	■	■	■
gEDA	pcb	■	■	■	■	■	■
FreePCB	fpc	■	■	■	■	■	■
FidoCadJ	fcd	■	■	■	■	■	■
Razen	rzp	■	■	■	■	■	■
IDF v2	idf	■	■	■	■	■	■
IDF v3	idf	■	■	■	■	■	■

■	Yes
■	No
■	Never
■	In progress
■	Future

Unit system

During board export process units are changed to millimeters [mm].

Python

To export board by Python You need to make few basic steps:

1. Import PCBexport module.
2. Choose file format, to which You want save.

Available export formats:

- Eagle (*.brd) → eagle(),
- FidoCadJ (*.fcd) → fidocadj(),
- KiCad (*.kicad_pcb) → kicad(),
- gEDA (*.pcb) → geda(),
- Razen (*.rzp) → razen().

3. Set basic export parameters.

Available settings:

- addHoles: export holes to file (if supported), default value = False,
- addAnnotations: export annotations to file (if supported), default value = False,
- addDimensions: export dimensions to file (if supported), default value = False.

4. Call function export().

Example:

```
import PCBexport
export = PCBexport.eagle()
export.addHoles = True
export.addAnnotations = True
export.export("/home/mariusz/test.brd")
```

Where:

- [/home/mariusz/test.brd](#): path and output file name (file extension is not required).

EXPORT BILL OF MATERIALS (BOM)



Option Export BOM allow You to export bill of material list to one of supported file formats.

Export BOM

Output file format: HyperText Markup Language (HTML) ▼

Output directory: /home/mariusz/Pulpit ...

Units

☒ Millimeters
☐ Inches

Options

☒ Full list
☐ Minimal header

Zero point

☒ Absolute
☐ Own

X: 7,00mm Y: 0,00mm

Export Close

Export hole location contains a number of settings that allow you to obtain the desired output file format:

1. Output file format:
 - Comma Separated Values (*.csv)
 - Text File (*.txt)
 - HyperText Markup Language (*.html)
2. Output directory: set path where file will be saved

3. Units:

- Millimeters: measure Everything in Metric , default value
- Inches: measure Everything in Inches, disabled option

4. Zero point drilling

- Absolute: base point for drilling is set in global 0, 0
- Own: set new base point for drilling
 - X: X value for new base point for drilling
 - Y: Y value for new base point for drilling

5. Extra options

- Minimal header: set whether extra data (project name, date, format) will be saved in to output file

```
Drill file
Project: sterownik
Date: 2015-04-25 16:02:37.990862
Unit: mm
Format: Decimal
Zero point drilling: Absolute (0 x 0)
```

- Option 'Full list' allow You to generate complex report for used components.

	Package	Value	Part
1	1X02		E\$33, E\$17, E\$19
2	2X06		E\$24
3	TL1105SP		S1
4	1206		LED01, LED3, LED2, LED5, LED4, LED7, LED6, LED9, LED8, LED19, LED18, LED11, LED10
5	H2M09ST		X1
6	1X08		E\$32
7	TUXGR_16X2_R2		DIS1
8	2X02		E\$20
9	R2512	150R	R44, R58, R60, R40, R52, R50, R24, R55
10	R2512	1k2	R17, R7, R3, R22, R46, R34, R37, R31
11	R0805	1k	E\$25
12	R1206	150R	R59, R12, R9
13			
14			

	ID	Package	Value	X	Y	Rotation	Side	Quantity
1	X2	PN61729		3.81 mm	38.1 mm	270 deg	TOP	1
2	POWER	1X06		39.37 mm	2.54 mm	0 deg	TOP	1
3	Z1	CT/CN0603	PGB1010604	12.065 mm	35.56 mm	0 deg	TOP	2
4	Z2	CT/CN0603	PGB1010604	12.065 mm	40.64 mm	0 deg	TOP	2
5	JP1	1X01		93.98 mm	50.8 mm	0 deg	TOP	4
6	TP2	1X01		93.98 mm	7.62 mm	0 deg	TOP	4

Python

To export BOM by Python You need to make few basic steps:

1. Import PCBexportHoles module.
2. Set basic export parameters.

Available settings:

- fileFormat:
 - Comma Separated Values → csv
 - Text File → txt
 - HyperText Markup Language → html
- filePath = def. home directory
- fileName = def. value 'untitled'
- units = mm/inch
- zeroPointDrilling:
 - -2: Absolute, def. value
 - -3: Own
 - zeroPointDrilling_X = def. value 0
 - zeroPointDrilling_Y = def. value 0
- minimalHeader = True/False, def. value False
- groupHoles = True/False, def. value False

3. Call function export().

Example:

```
from PCBexportBOM import xportBOM
export.fileFormat = 'html'
export.filePath = '/home/mariusz'
export.fileName = 'test'
export.units = 'mm'
export.fullList = True
export.export()
```


Where:

- [/home/mariusz](#): path for output file.
- [test](#): output file name (file extension is not required).

EXPORT HOLE LOCATIONS



Option Export hole locations allow You to export holes list to one of supported file formats.

Export hole locations

Output file format: Excellon (DRL)

Output directory: ...

Units

- ☒ Millimeters
- ☐ Inches

Options

- ☐ Mirror X
- ☐ Mirror Y
- ☐ Minimal header
- ☐ Group holes by diameter

Format

- ☒ Decimal
- ☐ Suppress leading zeros
- ☐ Suppress trailing zeros
- ☐ Keep zeros

Zero point drilling

- ☒ Absolute
- ☐ Own

X: 0,00mm

Y: 0,00mm

Export

Close

Export hole location contains a number of settings that allow you to obtain the desired output file format:

1. Output file format:
 - Comma Separated Values (*.csv)
 - Text File (*.txt)
 - HyperText Markup Language (*.html)
 - Excellon (DRL)
2. Output directory: set path where file will be saved
3. Units:
 - Millimeters: measure Everything in Metric , default value
 - Inches: measure Everything in Inches, disabled option
4. Format: choose format, in which values will be saved in file
Base value: 12.5[mm]
 - Decimal: without changes, value = 12.5
 - Suppress leading zeros: value = 12500
 - Suppress trailing zeros: value = 00125
 - Keep zeros: value = 0012500
5. Zero point drilling
 - Absolute: base point for drilling is set in global 0, 0
 - Own: set new base point for drilling
 - X: X value for new base point for drilling
 - Y: Y value for new base point for drilling
6. Extra options
 - Mirror X: multiply X value by -1
 - Mirror Y: multiply Y value by -1
 - Minimal header: set whether extra data (project name, date, format) will be saved in to output file

```
Drill file
Project: sterownik
Date: 2015-04-25 16:02:37.990862
Unit: mm
Format: Decimal
Zero point drilling: Absolute (0 x 0)
```

- Group holes by diameter: some output formats support grouping for holes by diameter

Diameter	X	Y
0.5	34.3	35.6
0.5	22.5	29.5
0.5	31.7	35.6
1.0	14.9	2.6
1.0	85.2	11.3
1.0	94.31	70.25
1.0	98.09	70.25
1.0	94.31	64.45
1.0	98.09	64.45
1.0	65.61	70.25
1.0	69.39	70.25
1.0	65.51	64.35
1.0	69.29	64.35
3.0	10.0	18.5
3.0	90.0	18.5

Diameter	X	Y
0.5		
	34.3	35.6
	22.5	29.5
	31.7	35.6
1.0		
	14.9	2.6
	85.2	11.3
	94.31	70.25
	98.09	70.25
	94.31	64.45
	98.09	64.45
	65.61	70.25
	69.39	70.25
	65.51	64.35
	69.29	64.35
3.0		
	10.0	18.5
	90.0	18.5
0.8		
	97.4	27.6
	97.5	23.2

Python

To export holes list by Python You need to make few basic steps:

1. Import PCBexportHoles module.
2. Set basic export parameters.

Available settings:

- fileFormat:
 - Comma Separated Values → csv
 - Text File → txt
 - HyperText Markup Language → html
 - Excellon → drl, def. value
- filePath = def. home directory
- fileName = def. value 'untitled'
- units = mm/inch
- saveFormat:
 - -2: Decimal, def. value
 - -3: Suppress leading zeros
 - -4: Suppress trailing zeros
 - -5: Keep zeros
- zeroPointDrilling:
 - -2: Absolute, def. value
 - -3: Own
 - zeroPointDrilling_X = def. value 0
 - zeroPointDrilling_Y = def. value 0
- mirror_X = True/False, def. value False
- mirror_Y = True/False, def. value False
- minimalHeader = True/False, def. value False
- groupHoles = True/False, def. value False

3. Call function export().

Example:

```
from PCBexportHoles import exportHoles
export = exportHoles()
export.fileFormat = 'html'
export.filePath = '/home/mariusz'
export.fileName = 'test'
export.units = 'mm'
export.saveFormat = -2
export.zeroPointDrilling = -2
export.groupHoles= True
export.export()
```

Where:

- [/home/mariusz](#): path for output file.
- [test](#): output file name (file extension is not required).

EXPORT HOLE LOCATIONS REPORT



Option Export hole locations report allow You to export report about needed, for drill process, tools.

Export hole locations report

Output directory:

Drill report for untitled
Created on 2015-04-25 16:21:06.077088
Drill report for plated through holes:
T1 1.40mm 0.055" (8 holes)

Total plated holes count: 8

Caution!



Output file have 'rpt' extension.

Python

To export board by Python You need to make few basic steps:

1. Import PCBexportHoles module.
2. Set basic export parameters.

Available settings:

- filePath = def. home directory
- fileName = def. value 'untitled'

3. Call function export().

Example:

```
from PCBexportHoles import exportHolesReport
export = exportHolesReport()
export.filePath = '/home/mariusz'
export.fileName = 'test'
export.export()
```

Where:

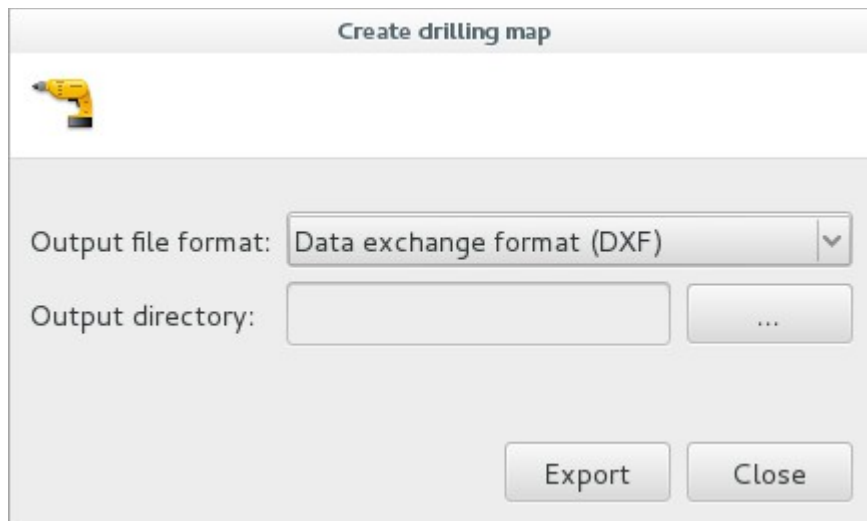
- `/home/mariusz`: path for output file.
- `test`: output file name (file extension is not required).

CREATE DRILLING MAP



Option 'Create drilling map' allow You to create 2D representation of board with marked drilling points.

Holes are splitted by diameter – each diameter value is represented by different symbol and color.



Supported formats:

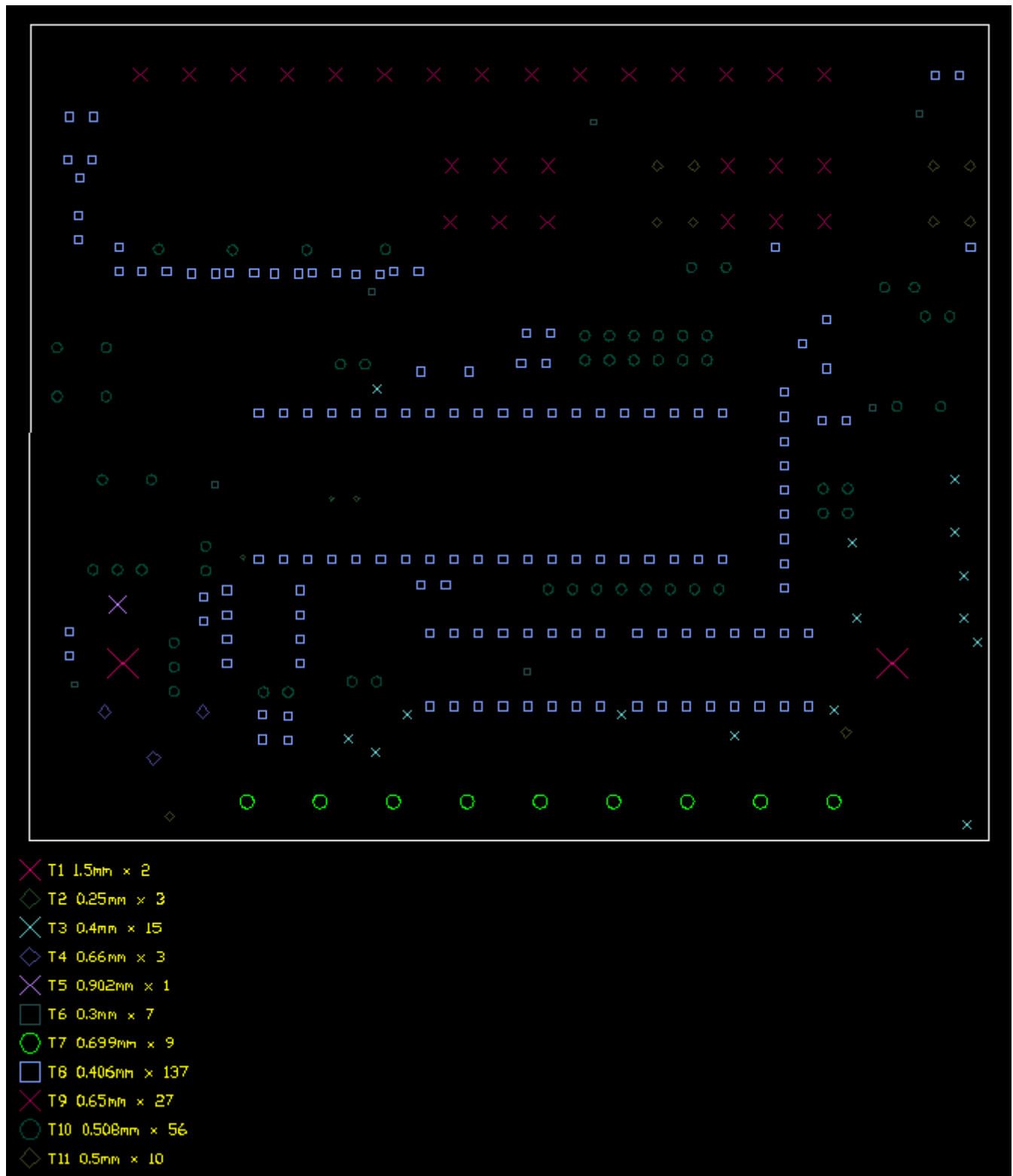
- DXF: Data exchange format (*.dxf),
- SVG: Scalable Vector Graphics (*.svg).



Caution!

Output file name is the same as project in FreeCAD.

File extension depends from selected output format.



Format in the legend: T1 1.5mm x 2

- T1: tool number,
- 1.5mm: hole diameter in [mm],
- 2: number of holes with same diameter.

Python

To create drilling map by Python You need to make few basic steps:

1. Import PCBexportDrillingMap module.
2. Set basic export parameters.

Available settings:

- fileFormat
 - Comma Separated Values → csv,
 - Data exchange format → dxf,
- filePath = def. home directory

3. Call function export().

Example:

```
from PCBexportDrillingMap import exportDrillingMap

export = exportDrillingMap()

export.filePath = '/home/mariusz'

export.fileFormat = 'dxf'

export.export()
```

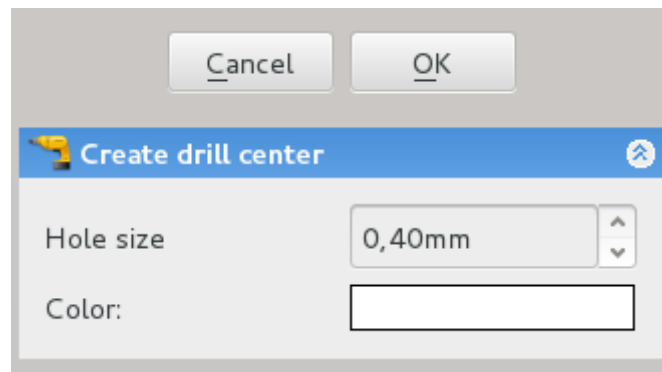
Where:

- [/home/mariusz](#): path for output file,
- [dxf](#): output file format.

CREATE DRILL CENTER



Option 'Create drill center' is useful for persons which will drill holes in PCB manually. This function allow to decrease holes sizes for better drill position during work.

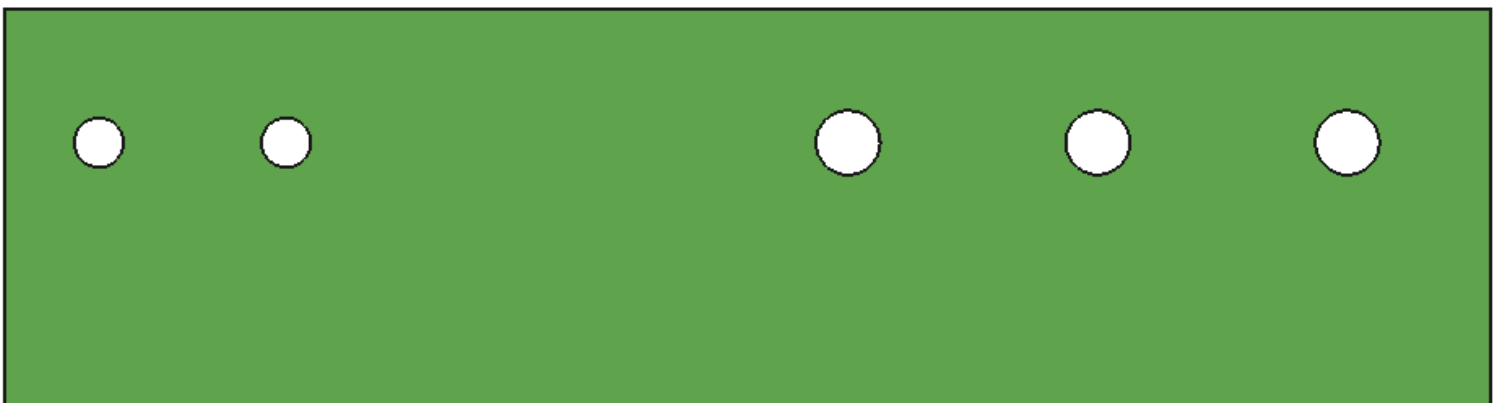


Caution!

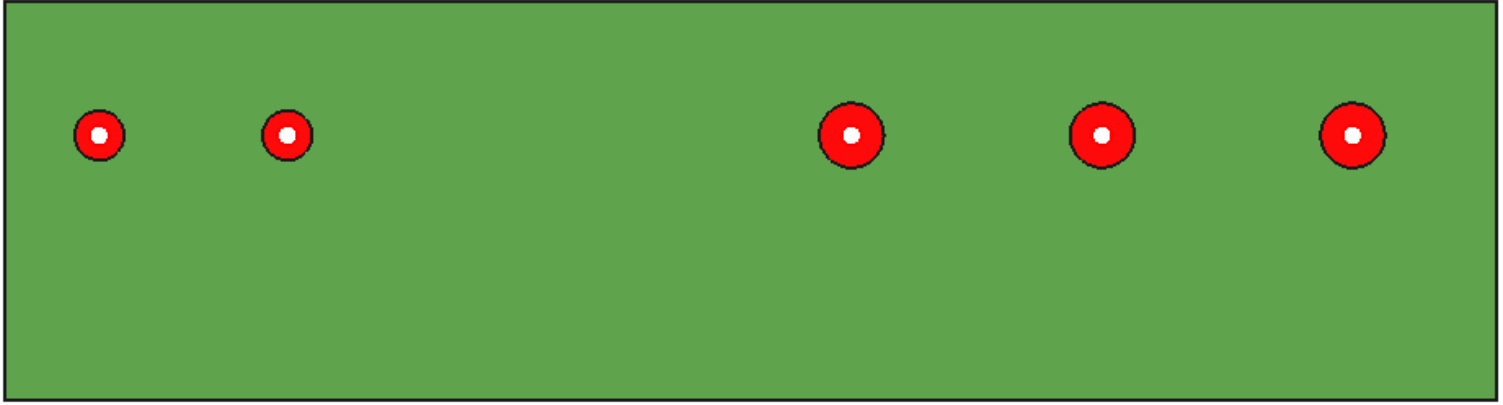


To update/change drill size, use same function. Script will automatically update layer.

Before:



After:



Python

To create drill center by Python You need to make only two steps:

1. Import PCBdrill module.
2. Call function createDrillcenter(size, color).

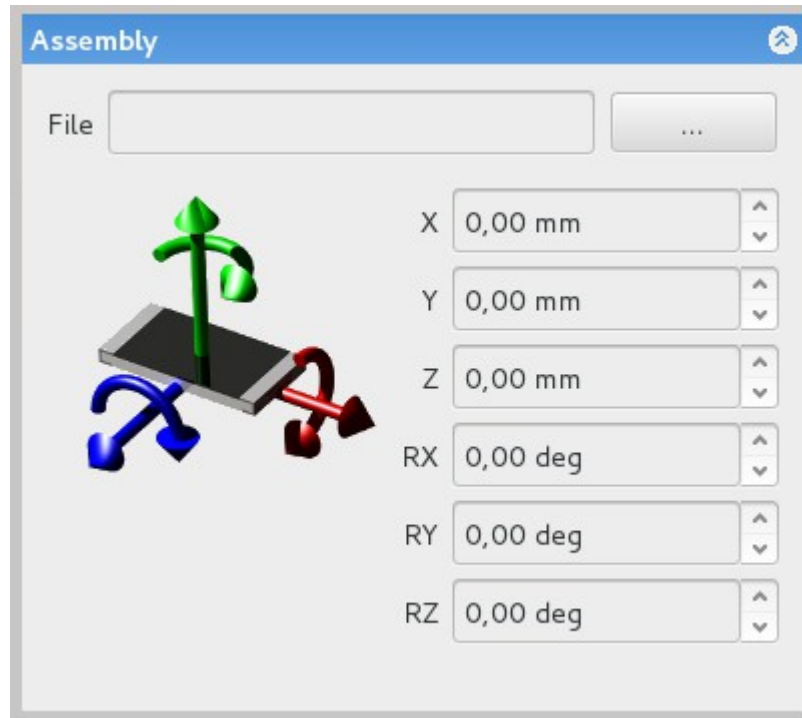
Where:

- size: new hole size in [mm],
- color: (R / 255, G / 255, b / 255).

Example:

```
from PCBdrill import createDrillcenter  
createDrillcenter(0.4, (1, 0.5, 0))
```

ADD ASSEMBLY



Python

To add assembly object by Python You need to import PCBassembly:

1. Import PCBassembly module.
2. Set basic export parameters.

Available settings:

- fileName = full path: destination path + filename,
 - x = position in X direction,
 - y = position in Y direction,
 - z = position in Z direction,
 - rx = rotation value around X axis,
 - ry = rotation value around Y axis,
 - rz = rotation value around Z axis,
3. Call function create().

Example:

```
from PCBassembly import createAssembly  
asm = createAssembly()  
asm.fileName = '/home/mariusz/dol.fcstd'  
asm.create()
```

Where:

- [/home/mariusz/dol.fcstd](#): path and file.

UPDATE ASSEMBLY



These option allow you to update loaded assemblies. Script can recognize if selected to update assembly is currently opened in FreeCAD or not.

Caution!



Auto update after file loaded does not work at the moment.

Caution!



During update process, script will keep only main assembly placement in 3D space – all deleted previously objects are reloaded.

Caution!



At one time you can select and update more than one assembly.

Python

To update assembly object by Python You need to import PCBassembly:

1. Import PCBassembly module.
2. Select assemblies to update.
3. Call function updateAssembly().

Example:

```
from PCBassembly import updateAssembly  
asm1 = FreeCAD.ActiveDocument.dol  
FreeCADGui.Selection.addSelection(asm1)  
updateAssembly()
```

Where:

- `asm1`: assembly to update.

EXPORT TO KERKYTHEA



OBJECTS PROPERTIES

Each object created in PCB workbench has unique parameters that can be set in Property View (View or Data tab). This task explains meaning each parameter.

Annotation/Object Name/Object Value

Text: text displayed by annotation object

Align: text position according to X, Y values

Mirror: mirror text

Spin: if parameter set to True text will keep rotation, parameter works for angle value $\geq 90\text{deg}$

Font: font name, parameter disabled


Size: font size

Rot: rotation value around Z axis

Side: text position on board (top/bottom side)

X: text position in X direction

Y: text position in Y direction

Property	Value
Base	
Text	[asdasdsa]
Visibility	true
Display	
Align	bottom-left
Mirror	None
Spin	true
Font	
Color	 [0, 0, 0]
Font	Hursheys
Size	4,27 mm

Property	Value
Base	
Label	PCBannotation_0000002
Placement	
Rot	0,00 °
Side	TOP
X	0 mm
Y	1,5 mm

Board

Display Holes: turn on/off holes

Holes: reference to sketch that containing holes

Border: reference to sketch that containing board outline

Thickness: board thickness

Property	Value
Base	
Label	Board
Auto Update	true
Holes	
Display	true
Holes	Board
PCB	
Border	Board
Thickness	1,6 mm

Part model not found in database

Part Name: reference to part name object

Part Value: reference to part value object

Keep Position: part will ignore changes in correction values if this value will be set to True

Package: 3D model name, parameter disabled for editing

Rot: rotation value around Z axis, parameter disabled for editing

Side: part position on board (top/bottom side), parameter disabled for editing

X: model position in X direction, parameter disabled for editing

Y: model position in Y direction, parameter disabled for editing

Property	Value
Base	
Part Name	R01
Part Value	R01
PCB	
Keep Position	false
Package	R1206
Rot	0,00 °
Side	TOP
X	3 mm
Y	-6 mm

Context menu PCB model:

- Assign model: assign 3D model to part,
- Update model: implement new model/correction values for selected part,
- Find model on-line: find 3D model in internet.

Explode

Active: turn of/off explode effect

Bottom Step Size: distance between parts placed on bottom side of board

Inverse: switch exploded parts from top to bottom and conversely

Top Step Size: distance between parts placed on top side of board

Property	Value
Base	
Label	Explode
Active	true
Bottom Step Size	10,00
Inverse	false
Top Step Size	10,00

Context menu Explode:

- Edit: edit list of models which will be exploded.

Part model found in database

Part Name: reference to part name object

Part Value: reference to part value object

Keep Position: part will ignore changes in correction values if this value will be set to True

Package: 3D model name, parameter disabled for editing

Rot: rotation value around Z axis

Side: part position on board (top/bottom side)

X: model position in X direction

Y: model position in Y direction

Property	Value
Base	
Part Name	R01
Part Value	R01
PCB	
Keep Position	false
Package	R1206
Rot	0,00 °
Side	TOP
X	3 mm
Y	-6 mm

Context menu PCB model:

- Placement model: change correction values for model in 'real time',
- Update model: implement new model/correction values for selected part.

Constraint area

Height: area height, parameter available only for some constraints areas type

Base: reference to sketch that containing area outline

Property	Value
Base	
Label	tPlaceOutline_0
Height	0,5 mm
Draft	
Base	tPlaceOutline_0

Glue path

Base: reference to sketch that containing glue path shape

Flat: if this parameter is set to True, object will ignore Width/Height parameters

Height: glue path height

Width: glue path width

Property	Value
Base	
<input checked="" type="checkbox"/> Placement	[(0,00 0,00 1,00); 0 °; (0 mm 0 mm 1,...
Label	Glue_0
Base	Sketch
Flat	false
Height	0,2 mm
Width	1,2 mm

Main assembly object

File: path to *.fcstd file

Placement: position of whole assembly in 3D space

Property	Value
Base	
Label	gora
File	/home/mariusz/Pulpit/gora.fcstd
<input checked="" type="checkbox"/> Placement	[(0,00 0,00 1,00); 0 °; (0 mm 0 mm 0 ...
Angle	0 °
<input checked="" type="checkbox"/> Axis	[0,00 0,00 1,00]
<input checked="" type="checkbox"/> Position	[0 mm 0 mm 0 mm]
x	0 mm
y	0 mm
z	0 mm

FILE FORMAT

This task explains database.cfg file format.

Each connection between 3D model and component used in ECAD software is stored in mentioned database.cfg file. All parameters can be set by Assign model window and manually by editing database.cfg file (not recomendet).

Example 3D model setting:

```
[hibhb_8788937480]
socket = [False, 0.0]
description =
add_socket = [False, None]
datasheet =
path = connectors/goldpin/1x08
soft = [[u'1X08', u'Eagle', 0.0, 0.0, 2.77, 90.0, 0.0, 0.0]]
name = 1X08
category = 10
```

Where:

```
[unique ID]
    unique ID = String
socket = [modelIsSocket, socketHeight]
    modelIsSocket = True / False
    socketHeight = Float
description = modelDescription
    modelDescription = String
add_socket = [addSocket, socketID]
    addSocket = True / False
    socketID = ID / None
datasheet = pathToDatasheet
    pathToDatasheet = String
```

```
path = pathTo3DModel
    pathTo3DModel = String
soft = [[componentName, softName, X, Y, Z, RX, RY, RZ]]
    componentName = String
    softName = String
    X = Float
    Y = Float
    Z = Float
    RX = Float
    RY = Float
    RZ = Float
name = modelName
    modelName = String
category = categoryID
    categoryID = Integer
```


SCRIPTS

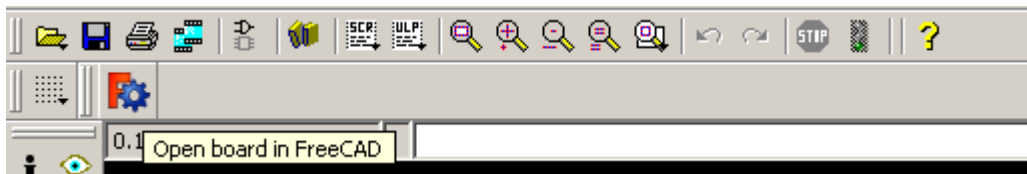
There are available few scripts which are helping exporting the boards to FreeCAD.

Eagle

Directly exporting boards from Eagle to FreeCAD [path: scripts/eagle]

- scripts/eagle/ulp/freecad.ulp – copy file to \$EAGLEDIR/ulp/
- scripts/eagle/scr/freecad.scr – copy file to \$EAGLEDIR/scr/
- scripts/eagle/bin/freecad.png – copy file to \$EAGLEDIR/bin/

In Eagle choose File → Execute Script → freecad.



On Linux to set path to FreeCAD change value of var 'programPath_LIN' in file freecad.ulp.

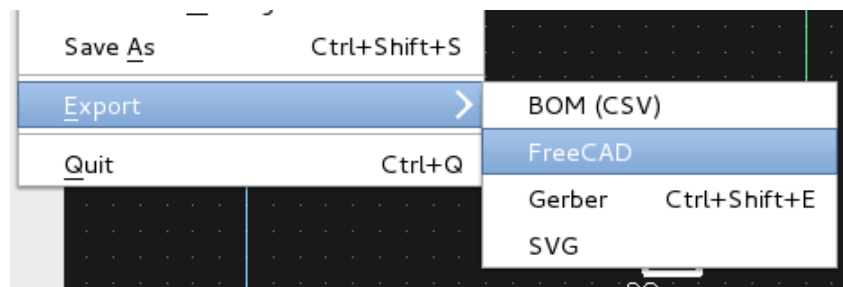
On Windows to set path to FreeCAD change value of var 'programPath_WIN' in file freecad.ulp.

Razen

Directly exporting boards from Razen to FreeCAD [path: scripts/razen]

- scripts/razen/freecad – copy folder `freecad` to \$RAZENDIR/plugin/export/

In Razen choose File → Export → FreeCAD.



On Linux to set path to FreeCAD change value of var 'programPath_LIN' in file conf.cfg.

On Windows to set path to FreeCAD change value of var 'programPath_WIN' in file conf.cfg.

ERRORS CODE

Code	Description	File
1	Function <i>getColorFromIGS()</i>	PCBpartManaging
2	Function <i>partExist()</i>	PCBpartManaging
3	Function <i>reloadList()</i>	PCBaddModel
4	Function <i>deletePackage()</i>	PCBassignModel
5	Function <i>convertDatabase()</i>	PCBassignModel
6	Function <i>reloadList()</i>	PCBassignModel

LICENCE

```
#####  
#*  
#* This program is free software; you can redistribute it and/or modify  
#* it under the terms of the GNU Lesser General Public License (LGPL)  
#* as published by the Free Software Foundation; either version 2 of  
#* the License, or (at your option) any later version.  
#* for detail see the LICENCE text file.  
#*  
#* This program is distributed in the hope that it will be useful,  
#* but WITHOUT ANY WARRANTY; without even the implied warranty of  
#* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
#* GNU Library General Public License for more details.  
#*  
#* You should have received a copy of the GNU Library General Public  
#* License along with this program; if not, write to the Free Software  
#* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307  
#* USA  
#*  
#####
```

CHANGELOG

- **Version: 3.2 (beta)**
 - Path layer: add support for arcs,
 - Added support for arcs width,
 - Multi model definition for one part,
 - Added support for lines/circles/arcs width,
 - Export hole locations/annotations
 - Preference tab for PCB mod.

- **Version: 3.1 (beta)**
 - Rewrite placement object function,
 - STEP format support for library parts,
 - Added categories to window ('Assign models')

- **Version: 3.0 (beta)**
 - Support for new pads type: oval, round rectangle,
 - New database format,
 - Add function: Layers settings,
 - New Icon Set.

- **Version: 2.9 (beta)**

- Changes in script code,
- KiCad: improvement of errors associated with the PCB import. Many thanks go to

Giovanni Di Trapani

- Round corners for PCB,
- Added new object: PCB board,
- FidoCadJ - import object name/value from file.

- **Version: 2.8 (beta)**

- Changes in script code,
- Add option: Create Bounding Box for PCB,
- Add option: Update parts directly from FreeCAD ,
- Export bill of materials (BOM),
- Export PCB layers to one of supported file formats,
- Rewrite placement object function.

- **Version: 2.7 (beta)**

- changes in script code,
- Add option: Create constraint area,
- Added ability to edit some layers,
- Use layer names from brd file,
- Initialize PCB thickness control from DRC settings in brd file.

- **Version: 2.6 (beta)**
 - rewrite FreePCB importer,
 - changes in script code,
 - add layer: path,
 - add support for:
 - IDF v2/v3,
 - FidoCadJ (.fcd),
 - KiCad (.kicad_pcb).

- **Version: 2.5 (beta)**
 - changes in script code,
 - fix bug in rotation of layers 21, 22, 51, 52,
 - fix bug in directly exporting boards from Eagle to FreeCAD on Linux,
 - add pad/via layer,
 - add support for:
 - FreePCB file format (*.fpc),
 - gEDA file format (*.pcb),
 - Razen file format (*.rzp).

- **Version: 2.4 (beta)**
 - changes in script code,
 - new command -> Assign models,
 - remove file param.py → replaced by dane.cfg,
 - new command → Convert old database to a new format,
 - directly exporting boards from Eagle to FreeCAD.

- **Version: 2.3 (beta)**
 - changes in script code,
 - add layerObject for layers 39, 40, 41, 42, 43,
 - split layer 39, 40, 41, 42, 43 into separately objects,
 - add Height property for layer 39, 40,
 - REMOVE var 'socketHeight' FROM 'CONF' => ADDED NEW ELEMENT IN 'bibliotekaDane' FROM 'PARAM' FILE.

- **Version: 2.2 (beta)**
 - changes in importing iges files,
 - changes in script code.

- **Version: 2.1 (beta)**
 - import IGES models with colours,
 - add possible to generate report with unknown parts,
 - add/cancel socket for all elements.

- **Version: 2.0 (beta)**
 - layer types supported: 20, 21, 22, 39, 40, 41, 42, 43, 47, 51, 52,
 - possible to show holes/vias independent,
 - possible to choose colours, transparency and names for each layer.

ToDo LIST

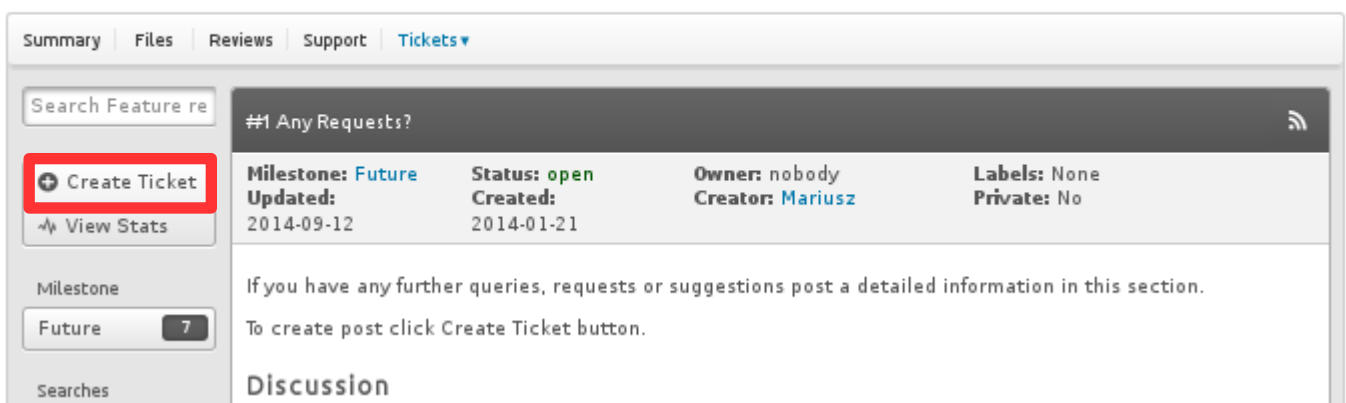
If you have any further queries, requests or suggestions post a detailed information on project site

<http://sourceforge.net/p/eaglepcb2freecad/feat-req/> or on forum <https://sourceforge.net/p/eaglepcb2freecad/forum/>.

To create post click Create Topic button.



To create post click Create Ticket button.



ERRORS

Found a bug? Post a detailed information on project page <http://sourceforge.net/p/eaglepcb2freecad/bugs/>. To create post click Create Ticket button.



[Summary](#) | [Files](#) | [Reviews](#) | [Support](#) | [Tickets▼](#)

+

Create Ticket

View Stats

Milestone

All 17

Searches

#1 Found a bug?

Milestone: All

Status: open

Owner: nobody

Labels: None

Updated: 2014-10-13

Created: 2014-01-21

Creator: Mariusz

Private: No

Found a bug? Post a detailed information in this section.
To create post click Create Ticket button.

Discussion