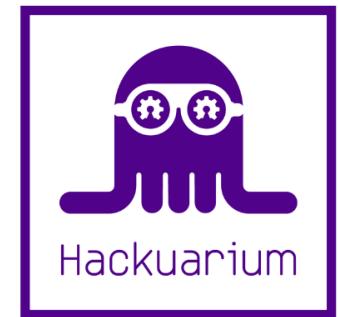


# Laser cut box

---

<https://github.com/Hackuarium/laser-cut-box>

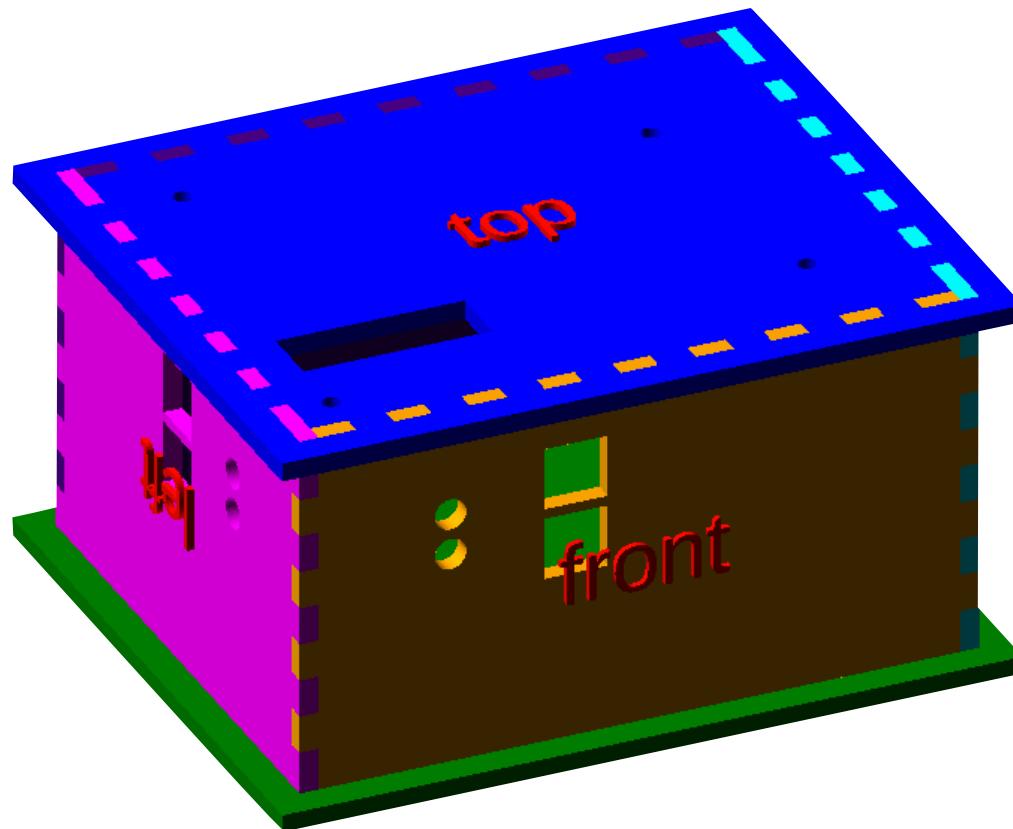
luc@patiny.com

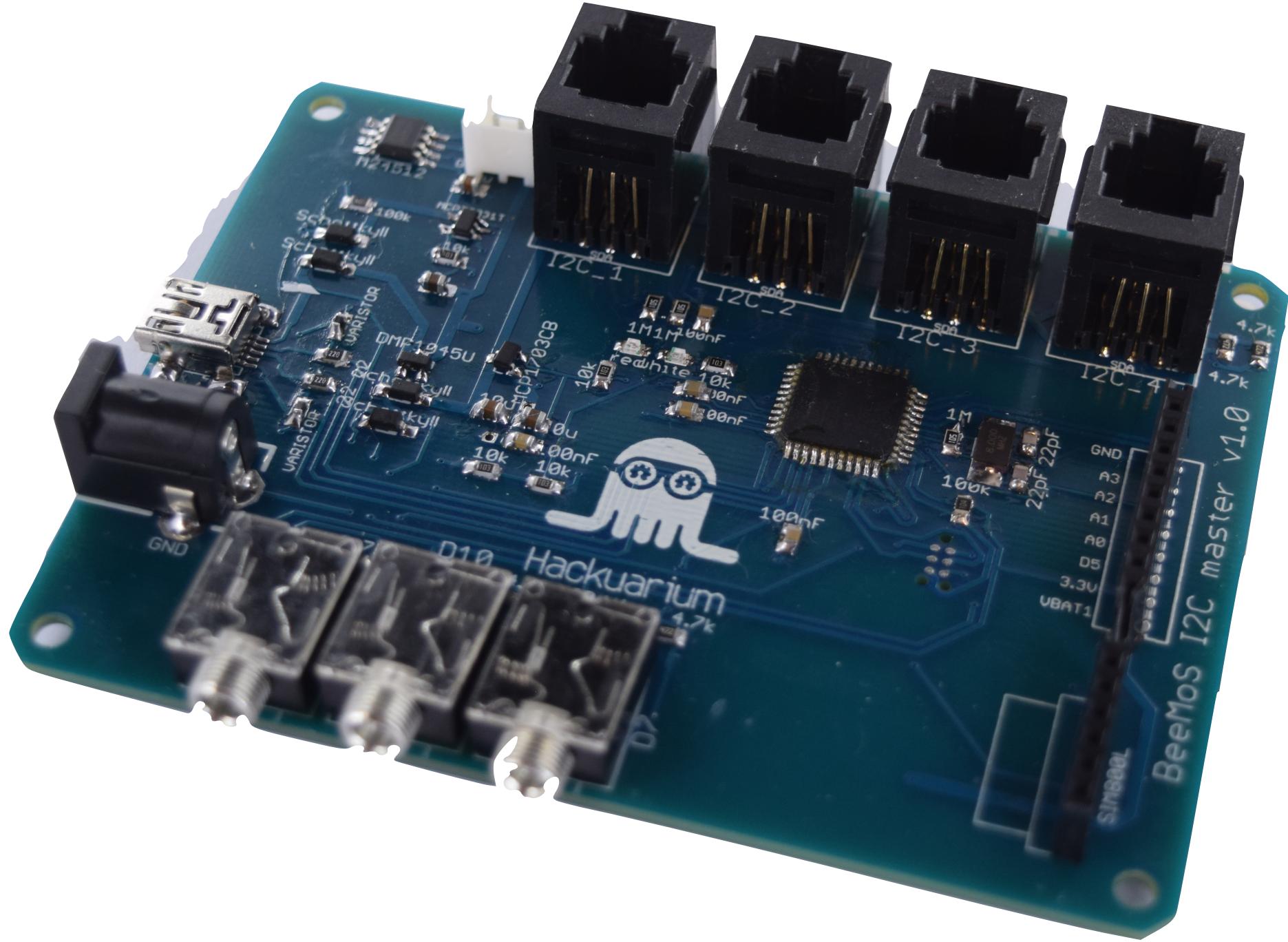


# Goal

---

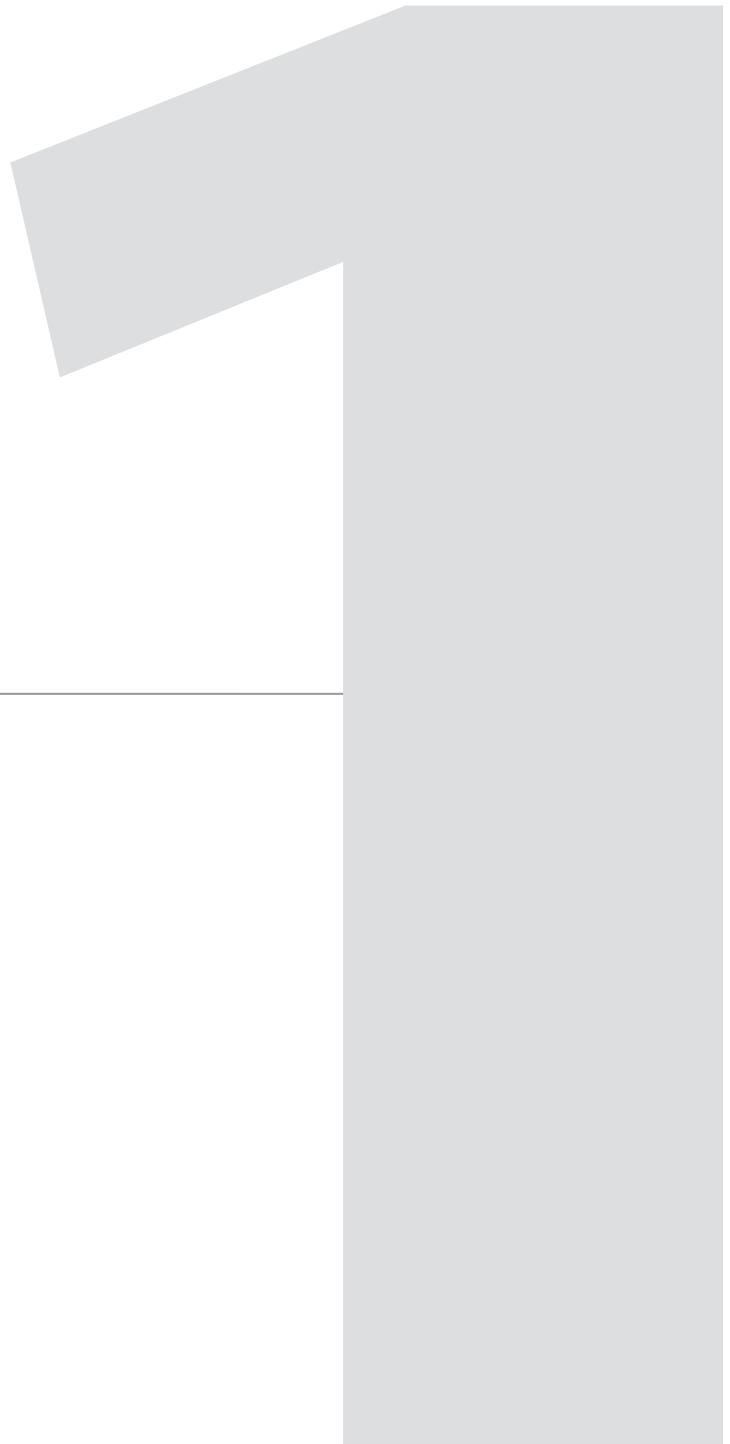
- **Quickly design a box for PCB based projects**





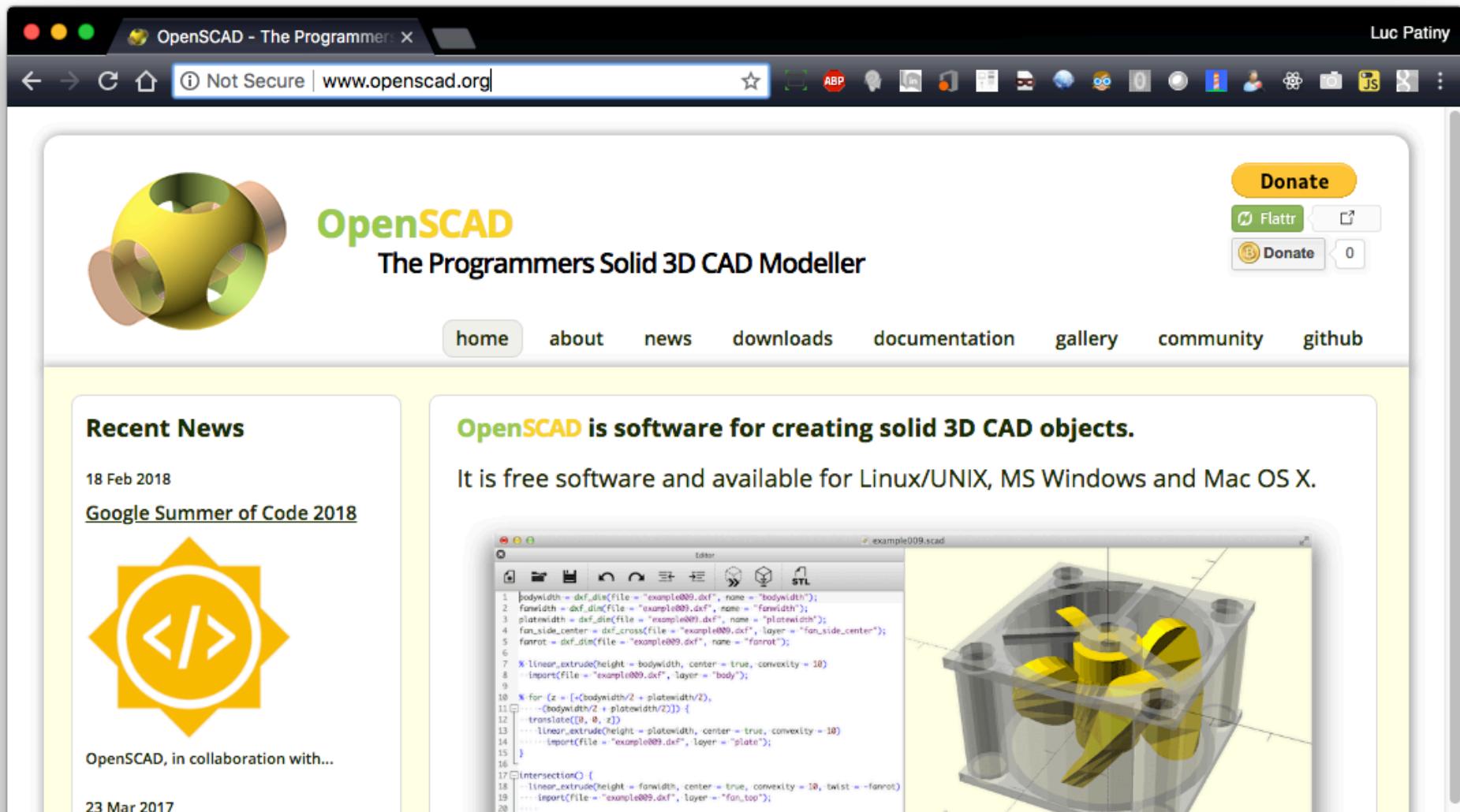
# Installation

---



# Install OpenSCAD

<https://github.com/Hackuarium/laser-cut-box>



The screenshot shows a web browser window displaying the official OpenSCAD website at [www.openscad.org](http://www.openscad.org). The page features a large yellow 3D model of a complex gear-like object on the left. To its right, the text "OpenSCAD" is written in green and yellow, followed by "The Programmers Solid 3D CAD Modeller". A navigation bar below includes links for "home", "about", "news", "downloads", "documentation", "gallery", "community", and "github". On the far right, there are donation buttons for "Flattr" and "Bitcoin". A "Recent News" sidebar on the left lists items from February 2018, including "Google Summer of Code 2018" and "OpenSCAD, in collaboration with...". The main content area contains a section about OpenSCAD's capabilities and a code editor window showing a portion of an SCAD script for a "example009.scad" file. The script includes code for extruding and importing files to create a 3D model of a fan inside a case.

OpenSCAD  
The Programmers Solid 3D CAD Modeller

home about news downloads documentation gallery community github

Recent News

18 Feb 2018

[Google Summer of Code 2018](#)

OpenSCAD, in collaboration with...

OpenSCAD is software for creating solid 3D CAD objects.  
It is free software and available for Linux/UNIX, MS Windows and Mac OS X.

```
1 bodywidth = dxf_dim(file = "example009.dxf", name = "bodywidth");
2 fanwidth = dxf_dim(file = "example009.dxf", name = "fanwidth");
3 platewidth = dxf_dim(file = "example009.dxf", name = "platewidth");
4 fan_side_center = dxf_cross(file = "example009.dxf", layer = "fan_side_center");
5 fanrot = dxf_dim(file = "example009.dxf", name = "fanrot");
6
7 linear_extrude(height = bodywidth, center = true, convexity = 10);
8 import(file = "example009.dxf", layer = "body");
9
10 for (z = [-(bodywidth/2 + platewidth/2),
11 ... -(bodywidth/2 + platewidth/2)]) {
12 ... translate([0, 0, z]);
13 ... linear_extrude(height = platewidth, center = true, convexity = 10);
14 ... import(file = "example009.dxf", layer = "plate");
15 }
16
17 intersectionO {
18 ... linear_extrude(height = fanwidth, center = true, convexity = 10, twist = -fanrot);
19 ... import(file = "example009.dxf", layer = "fan_top");
20 }
```

# Download the project

<https://github.com/Hackuarium/laser-cut-box>

The screenshot shows a GitHub repository page for 'Hackuarium / laser-cut-box'. The page includes the repository name, a summary of activity (5 commits, 1 branch, 0 releases, 1 contributor), a file browser, and download options. A red circle highlights the 'Clone or download' button.

GitHub - Hackuarium/laser-cut-box GitHub - Hackuarium/laser-cut-box Luc Patiny

GitHub, Inc. [US] | https://github.com/Hackuarium/laser-cut-box

Features Business Explore Marketplace Pricing Search Sign in or Sign up

Hackuarium / laser-cut-box Watch 6 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Insights

Create a box for a PCB using laser cutting machine <https://hackuarium.github.io/laser-cut-box/>

pcb laser-cutting printed-circuit-boards box box2d box3d openscad

5 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

Ipatiny move images

demo add 3d example and demo folder

docs move images

pcbbox add 3d example and demo folder

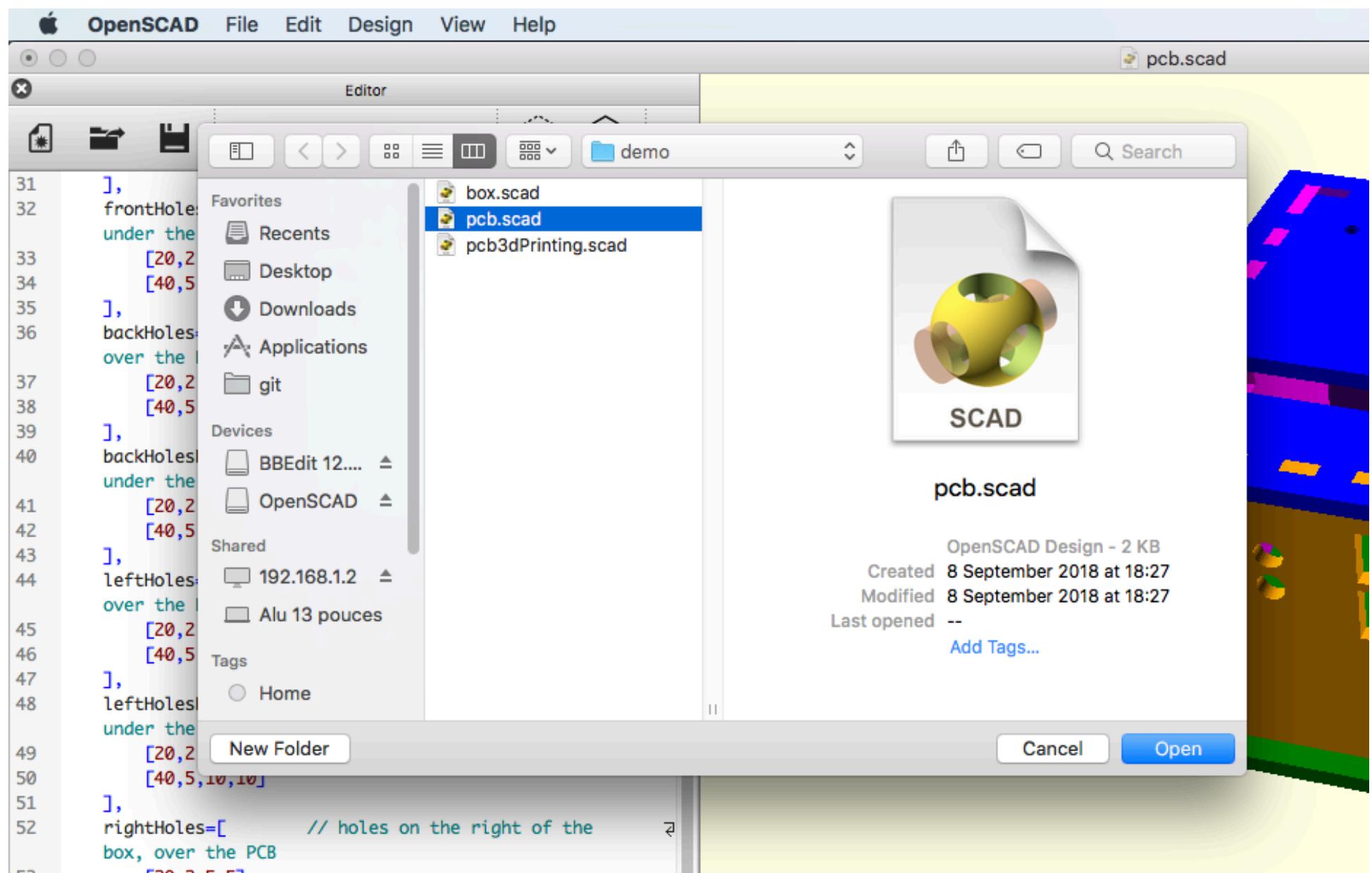
README.md move images

Clone with HTTPS <https://github.com/Hackuarium/laser-cut-box>

Use Git or checkout with SVN using the web URL.

Open in Desktop Download ZIP

# Open: demo → pcb.scad



**Code editor**

**Preview**

**Render**

The screenshot shows the OpenSCAD application interface. On the left is the **Code editor**, displaying the SCAD source code for a 3D model. The code defines various holes and labels for a box. In the center is the **Preview** pane, showing a 3D rendering of the model. On the right is the **Render** pane, showing a more detailed 3D rendering of the model. At the bottom is the **Console** pane, displaying the output of the compilation process.

```
31 ],
32 frontHolesB=[ // holes in front of the box,
33     under the PCB
34         [20,2.5,5],
35         [40,5,10,10]
36 ],
37 backHoles=[ // holes in the back of the box,
38 over the PCB
39         [20,2.5,5],
40         [40,5,10,10]
41 ],
42 backHolesB=[ // holes in front of the box,
43     under the PCB
44         [20,2.5,5],
45         [40,5,10,10]
46 ],
47 leftHoles=[ // holes on the left of the box,
48 over the PCB
49         [20,2.5,5],
50         [40,5,10,10]
51 ],
52 leftHolesB=[ // holes on the left of the box,
53 under the PCB
54         [20,2.5,5],
55         [40,5,10,10]
56 ],
57 rightHoles=[ // holes on the right of the
58 box, over the PCB
59         [20,2.5,5],
60         [40,5,10,10]
61 ],
62 rightHolesB=[ // holes on the right of the
63 box, under the PCB
64         [20,2.5,5],
65         [40,5,10,10]
66 ],
67 showLabels=false, //should we show the labels
68 labelsSize=10, // size of the labels
69 3d=true, // 3d rendering or just 2d ?
70 show="all" // used for 3D printing.
71 Possible values:
72             // all (default), top, bottom,
73             fulltop (5 faces), fullbottom (5 faces)
74 );
75 
```

viewport: translate = [ 54.00 44.00 30.80 ], rotate = [ 60.20 0.00 17.90 ], distance = 336.10

pcb.scad

Editor

Console

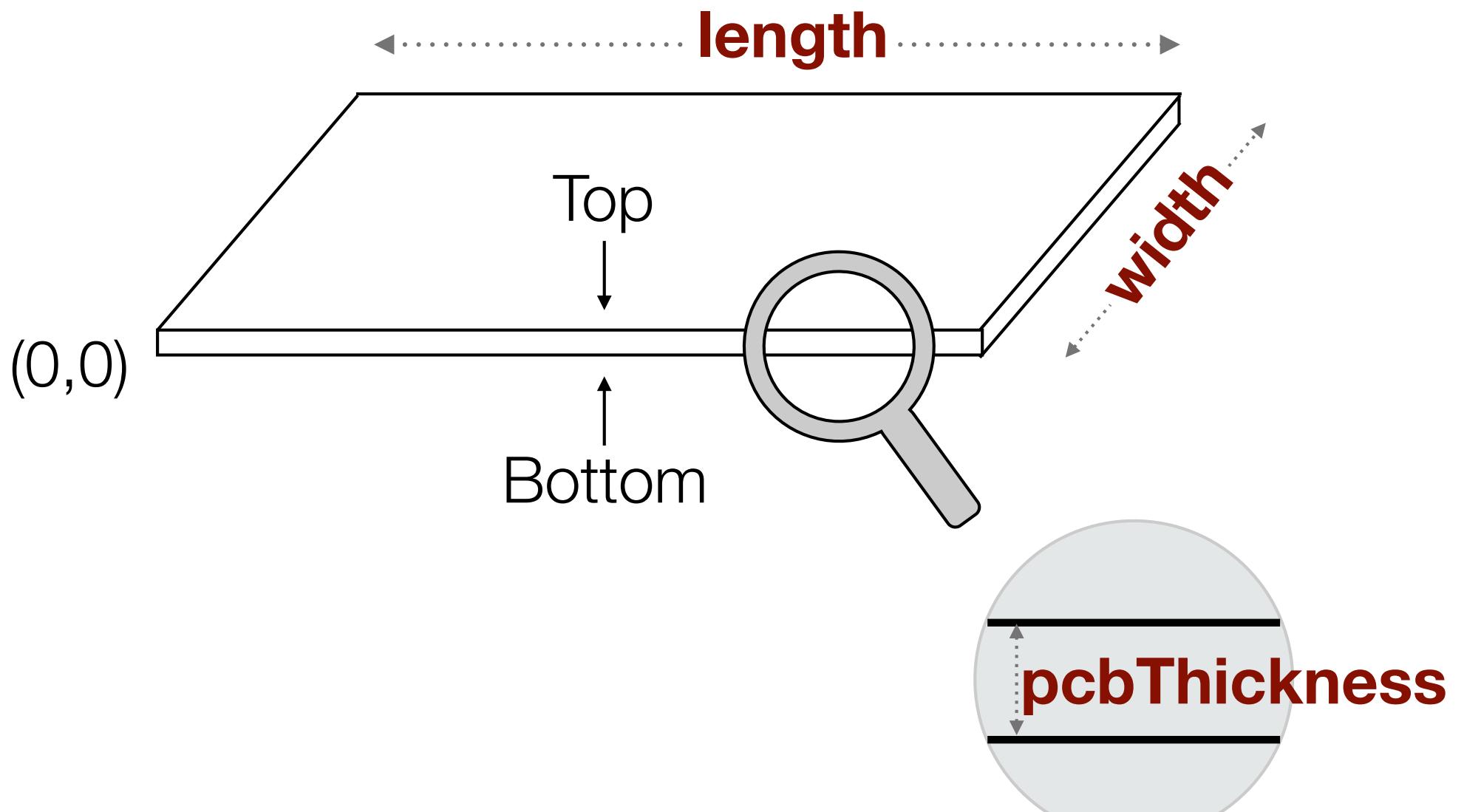
Saved backup file: /Users/lpatiny/Documents/OpenSCAD/backups/pcb-backup-JjKM7847.scad  
Compiling design (CSG Tree generation)...  
Compiling design (CSG Products generation)...  
Geometries in cache: 259  
Geometry cache size in bytes: 4400000  
CGAL Polyhedrons in cache: 3  
CGAL cache size in bytes: 8962536  
Compiling design (CSG Products normalization)...  
Normalized CSG tree has 6 elements  
Compile and preview finished.  
Total rendering time: 0 hours, 0 minutes, 0 seconds

# Parameters

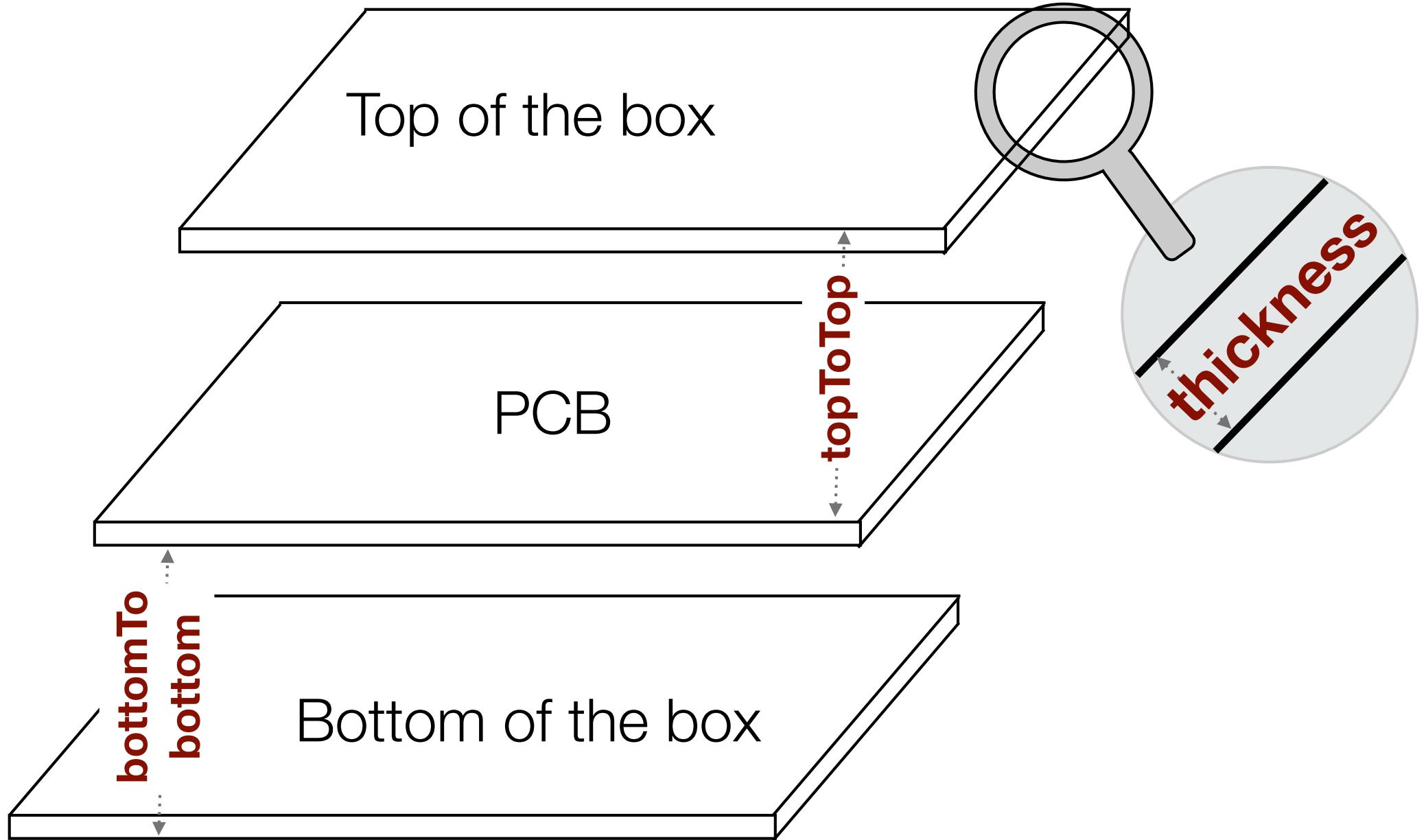
---

# PCB

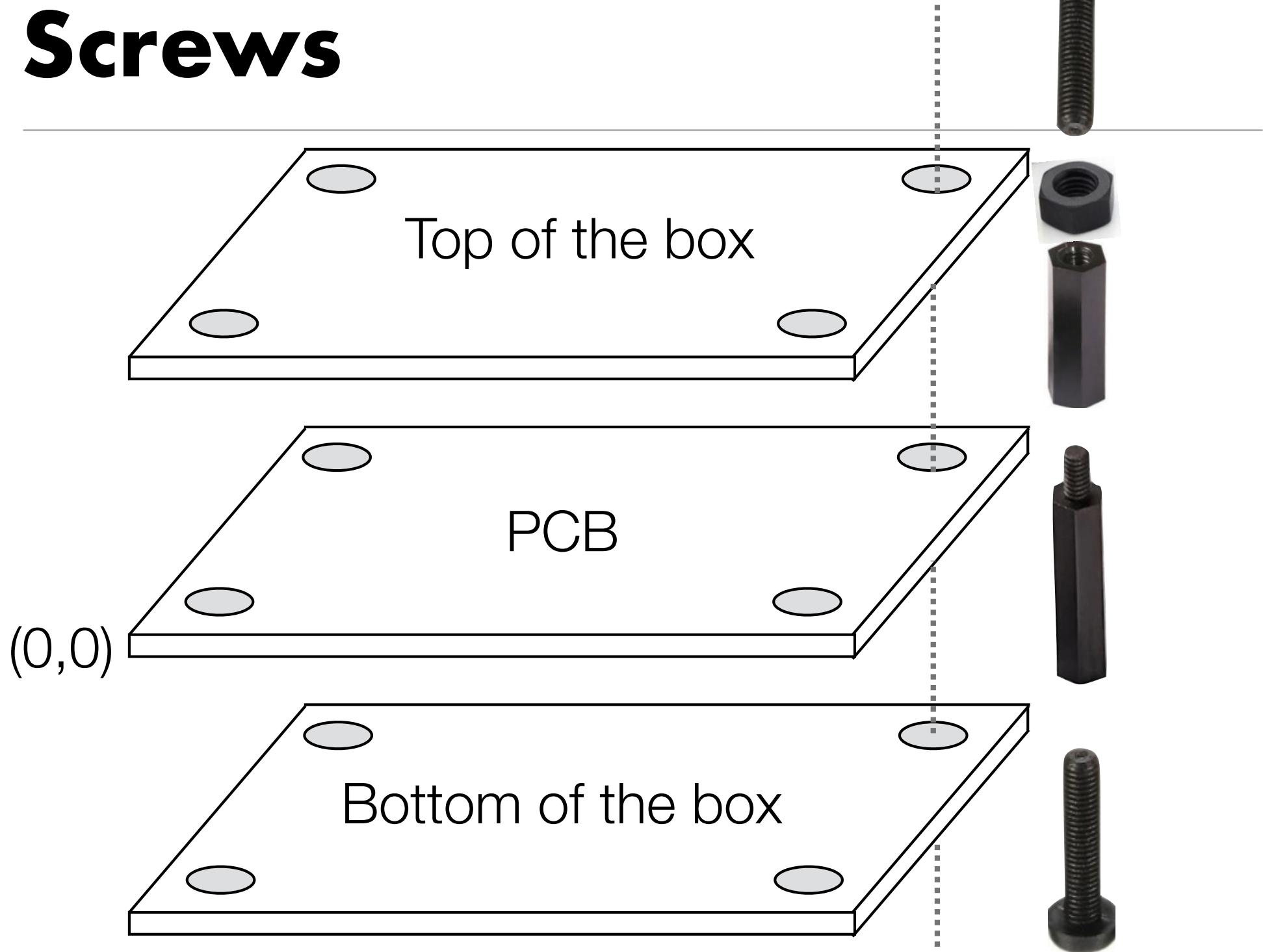
---



# Distance top / bottom



# Screws



# **screws**

---

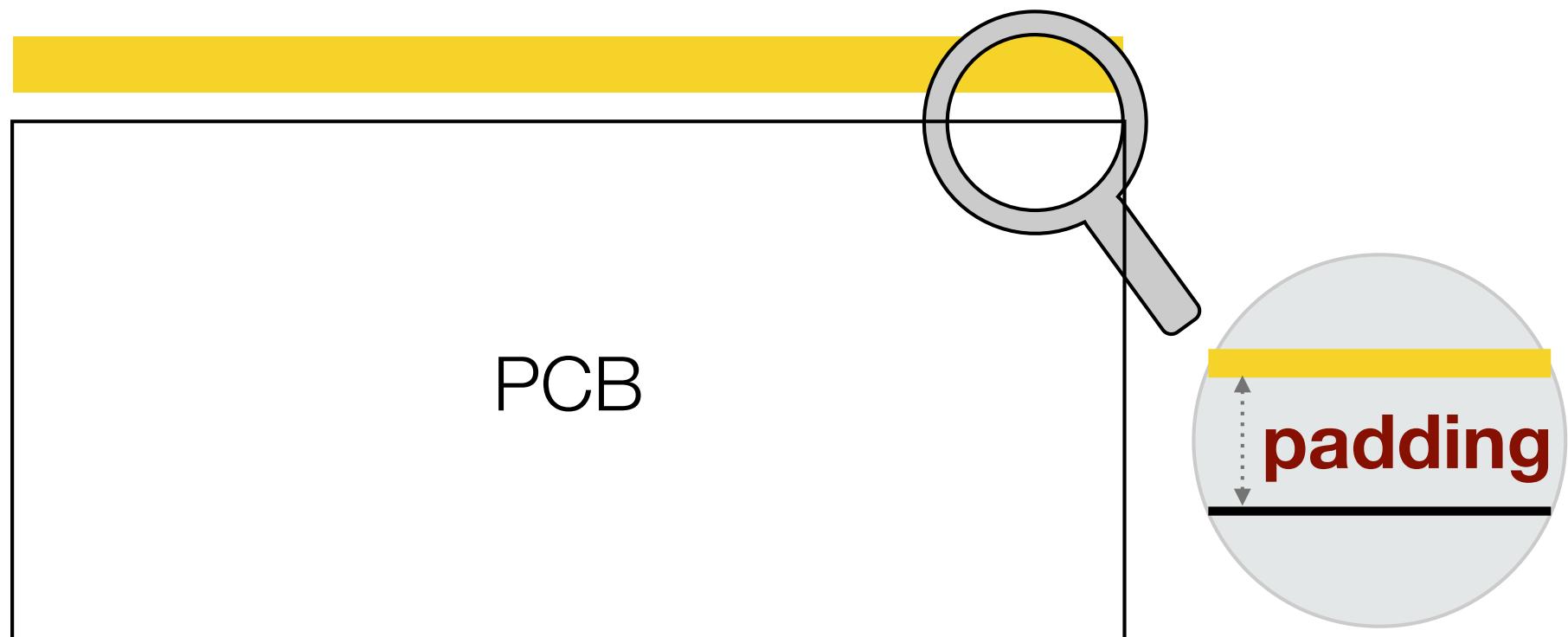
```
screws=[  
    [ 5 , 5 ] ,  
    [ 10 , 70 ] ,  
    [ 85 , 15 ] ,  
    [ 80 , 60 ]  
],  
screwDiameter=3 ,
```

- array of X / Y
- unlimited number of holes
- allows to fix the PCB
- holes on top and bottom
- should specify diameter

# padding

---

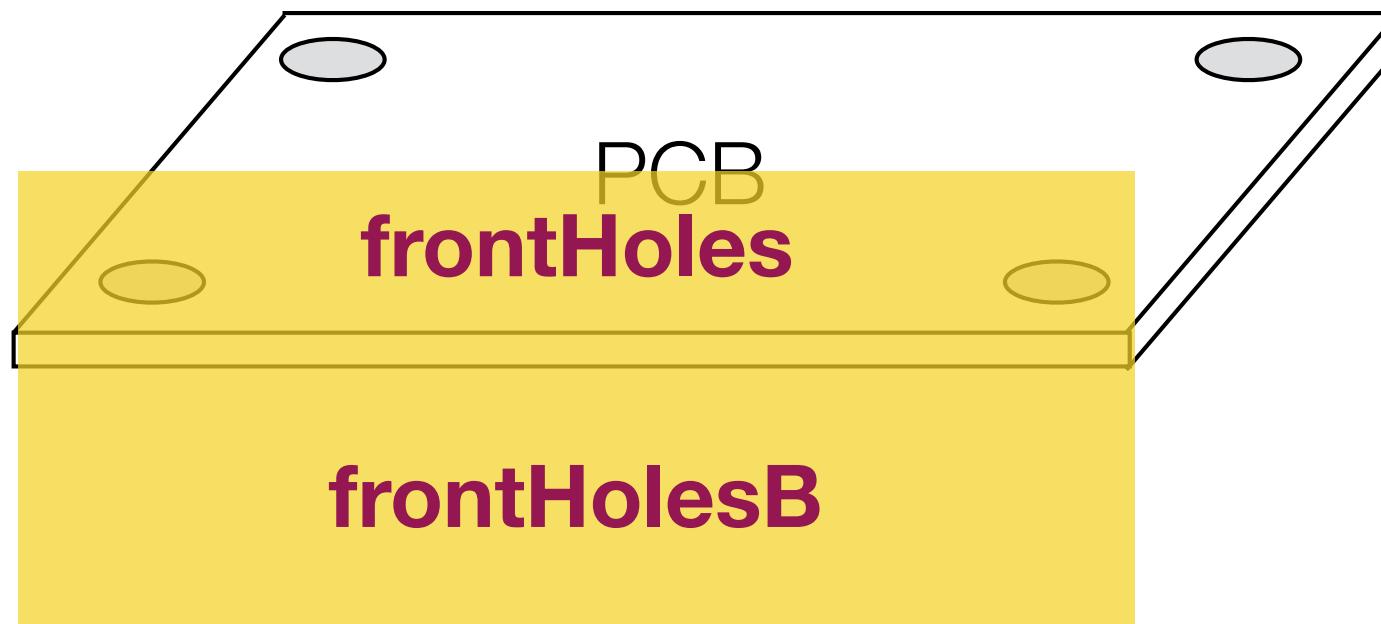
- Distance between PCB and side walls



# Front and holes

---

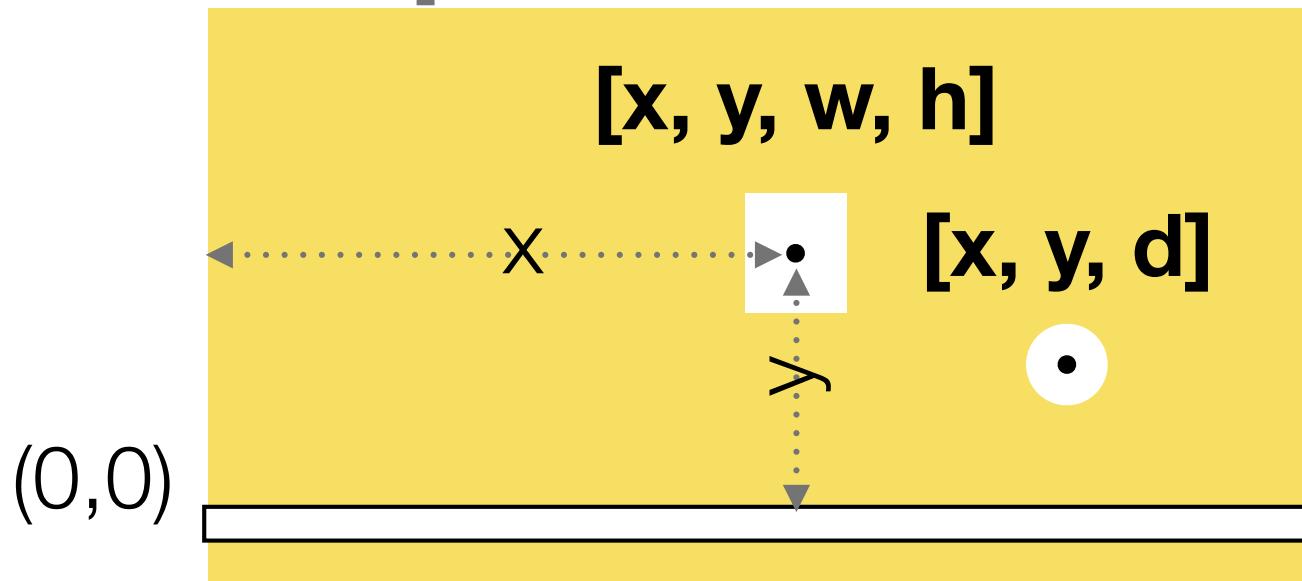
- 2 parts: over the PCB and under the PCB



# frontHoles

---

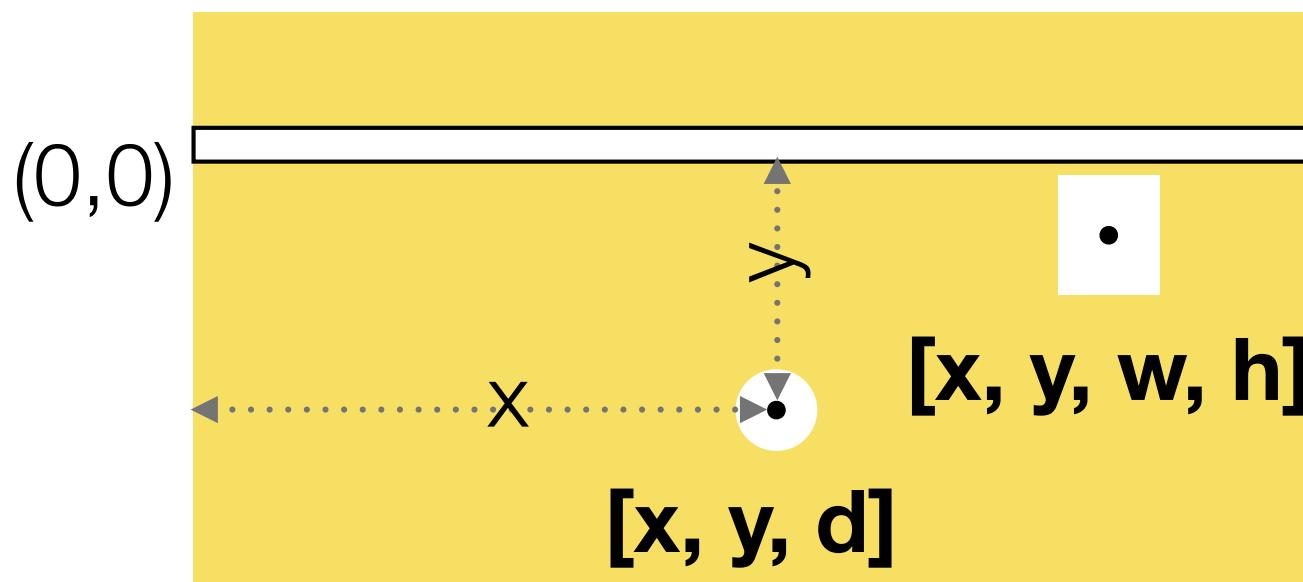
- **[ $x, y, w, h$ ]: centered rectangle**
- **[ $x, y, d$ ]: centered circle of diameter  $d$**
- **(0,0) is top left of PCB**



# frontHolesB (bottom)

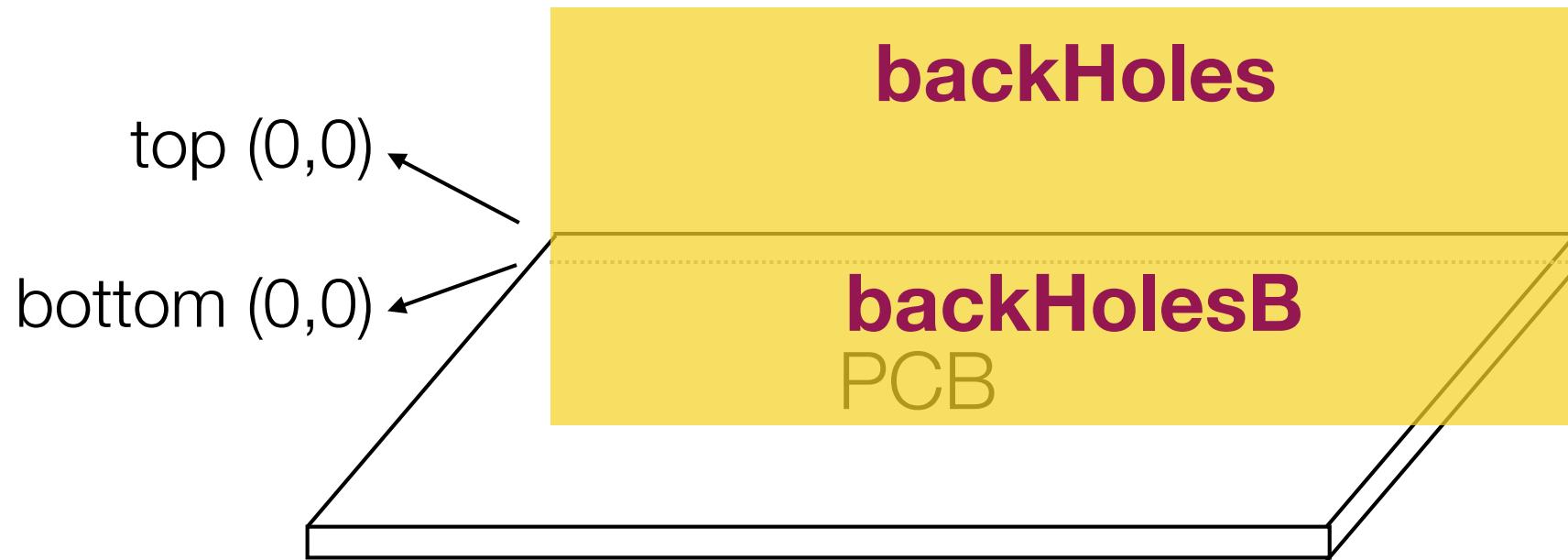
---

- **[x, y, w, h]: centered rectangle**
- **[x, y, d]: centered circle of diameter d**
- **(0,0) is top left of PCB**
- **y > 0 !**



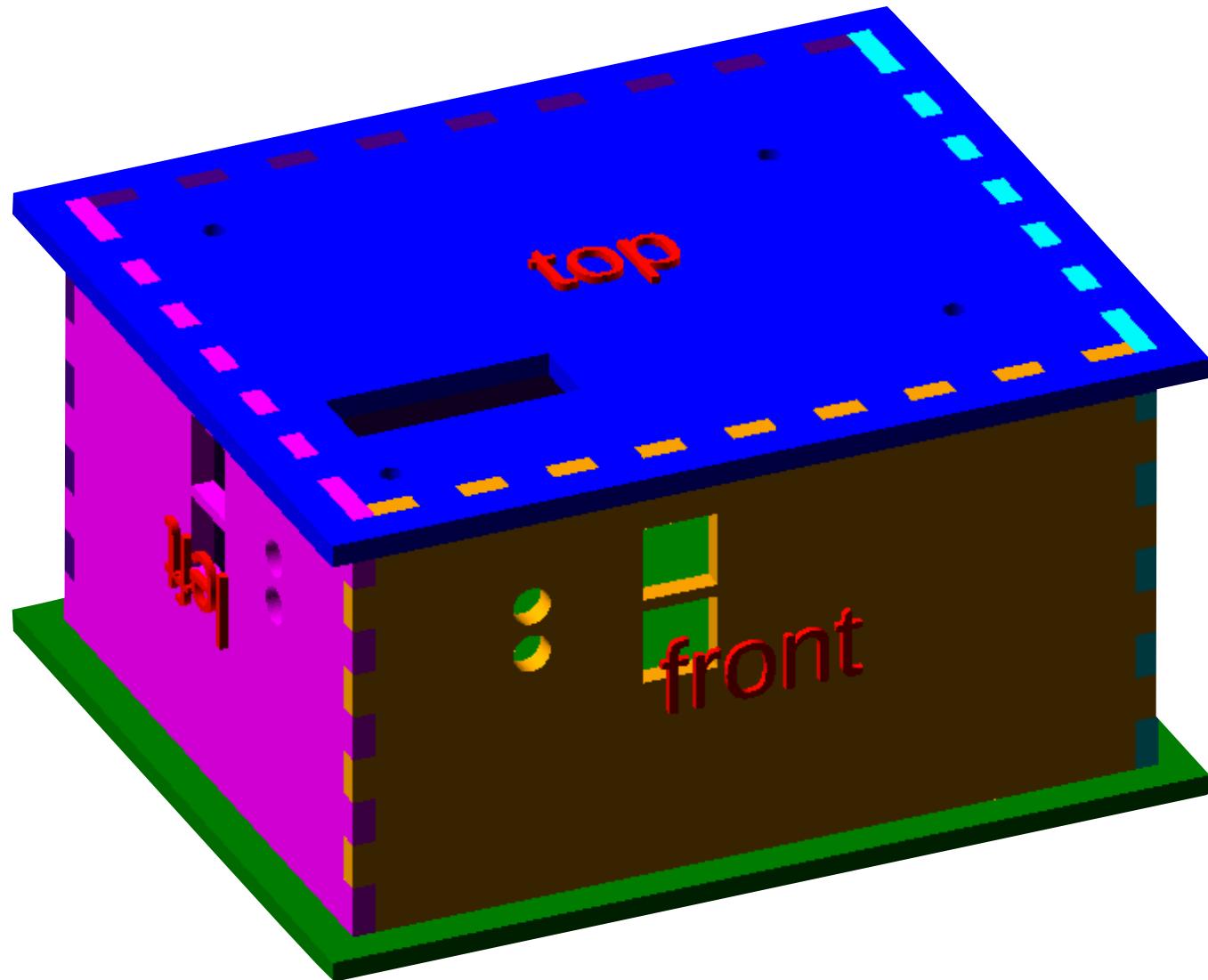
# Back

---



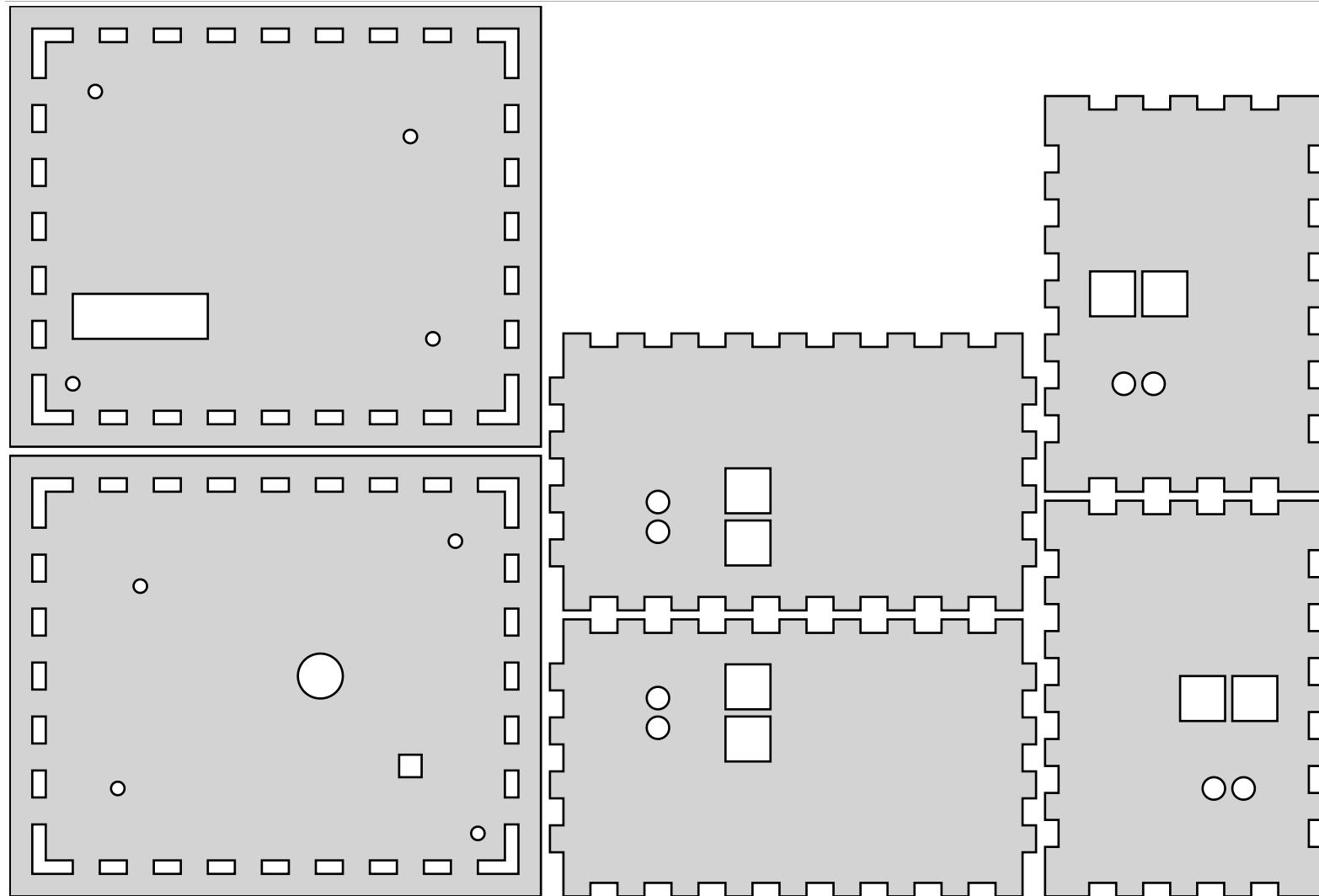
# 3d=true

---



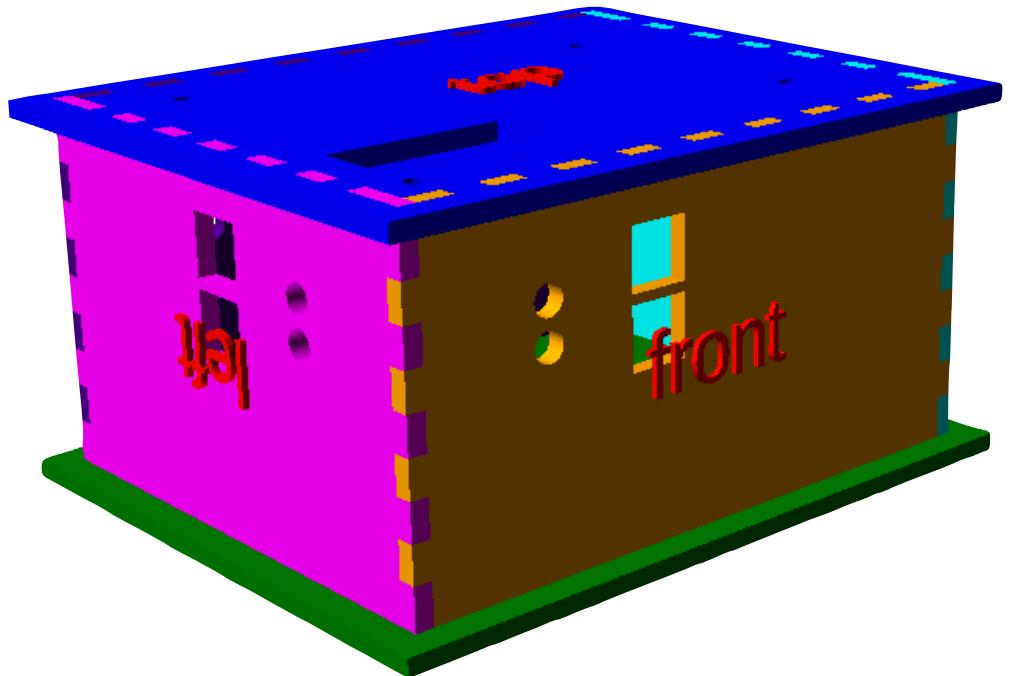
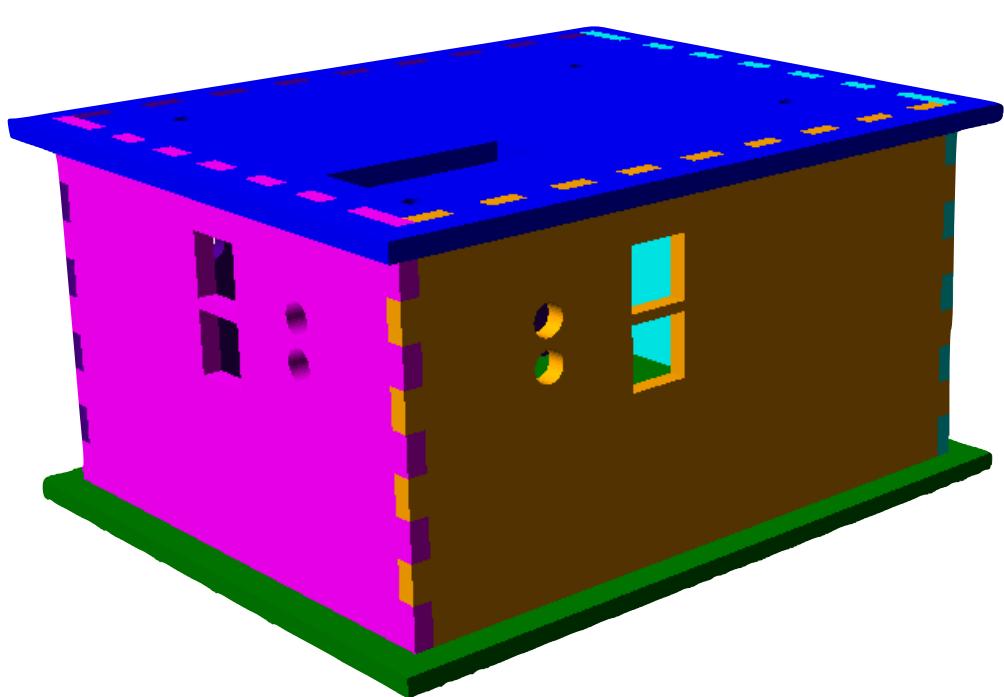
# 3d=false

---



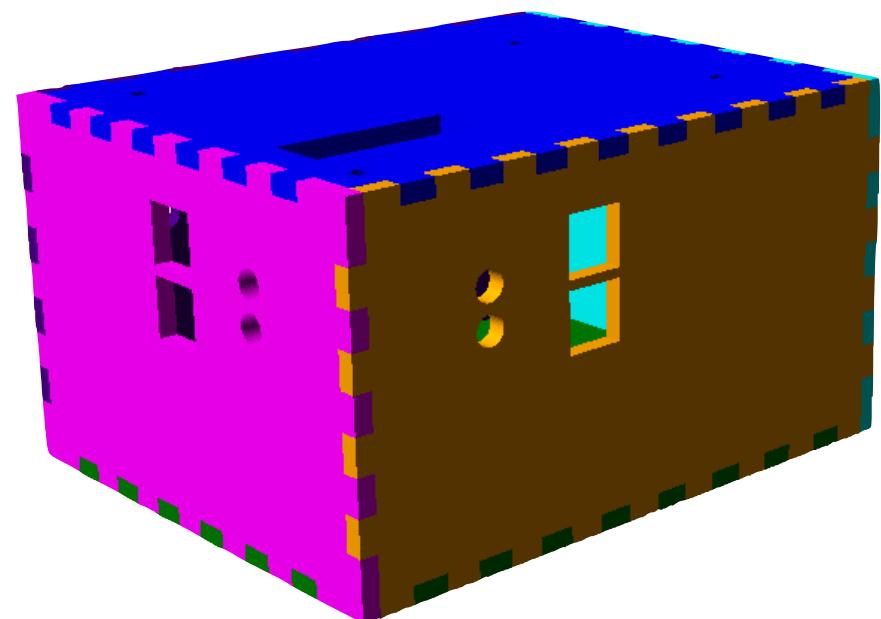
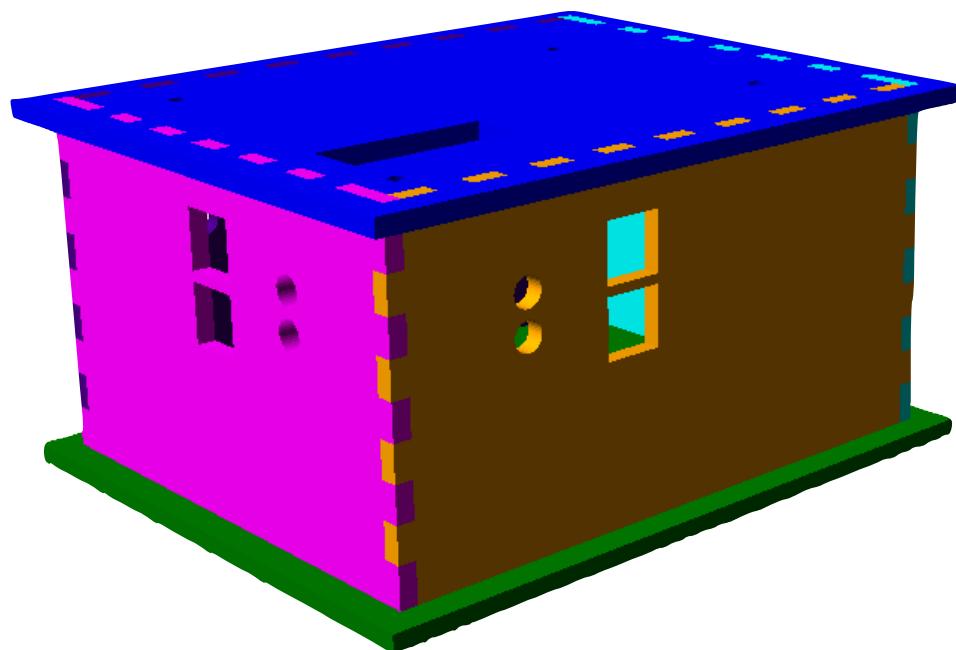
# **showLabels=false / true**

---



# **extend=10 / 0**

---

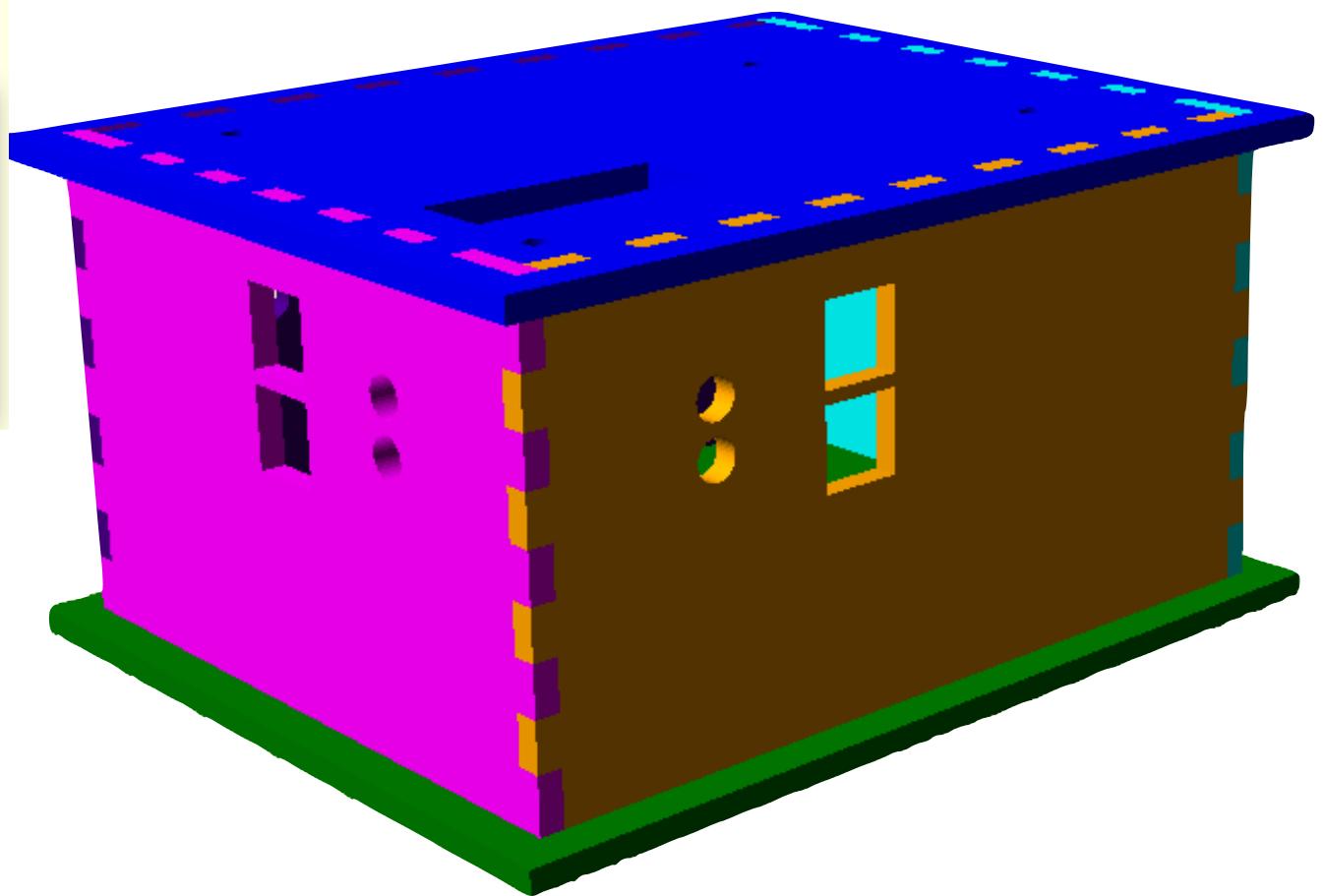
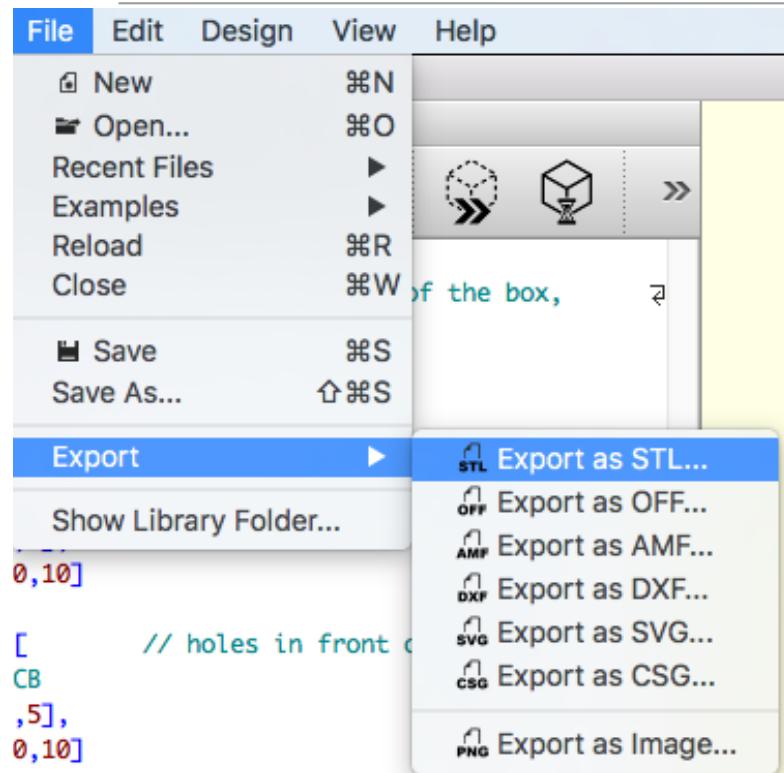


# **Exportation**

---



# export STL for 3D printing



# export DXF for laser cutting

