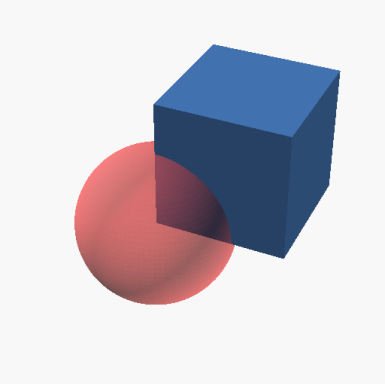

OpenScad examples

Table of Contents

1. Document information

Links to Document

Document online

Document Source

Document PDF

test repo for building openscad files into different outputs
This is still work in progress but can already build a png and stl of each scad file in the opescad directory.
See the online or pdf versions for the images as te readme is realy only the source and right now is not WYSIWYG!

2. Objects

2.1. Object - Axle

required a screw and didn’t have one.

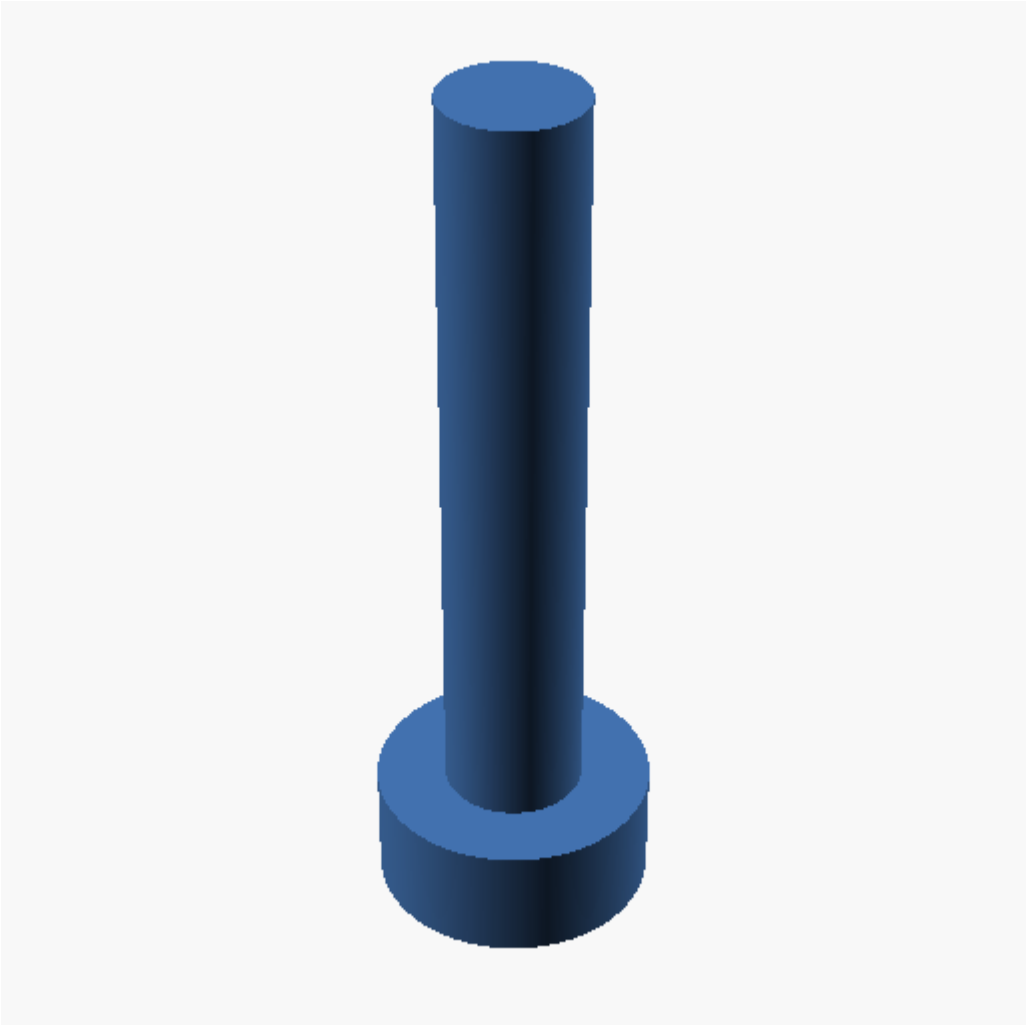

Figure 1. image

Listing 1. Openscad source

```
// axle for bearing for filament roller
// had no screw printed one ...
// the free end can be melted when the axle has been inserted so that no
fastener is required
$fn=360;
cylinder (h=22,d=3.5);
cylinder (h=3,d=7);
```

2.2. Object - ikeabung

This was a replacement foot for an IKEA shelf.

The actual foot was screwed in with a bolt on the underside.

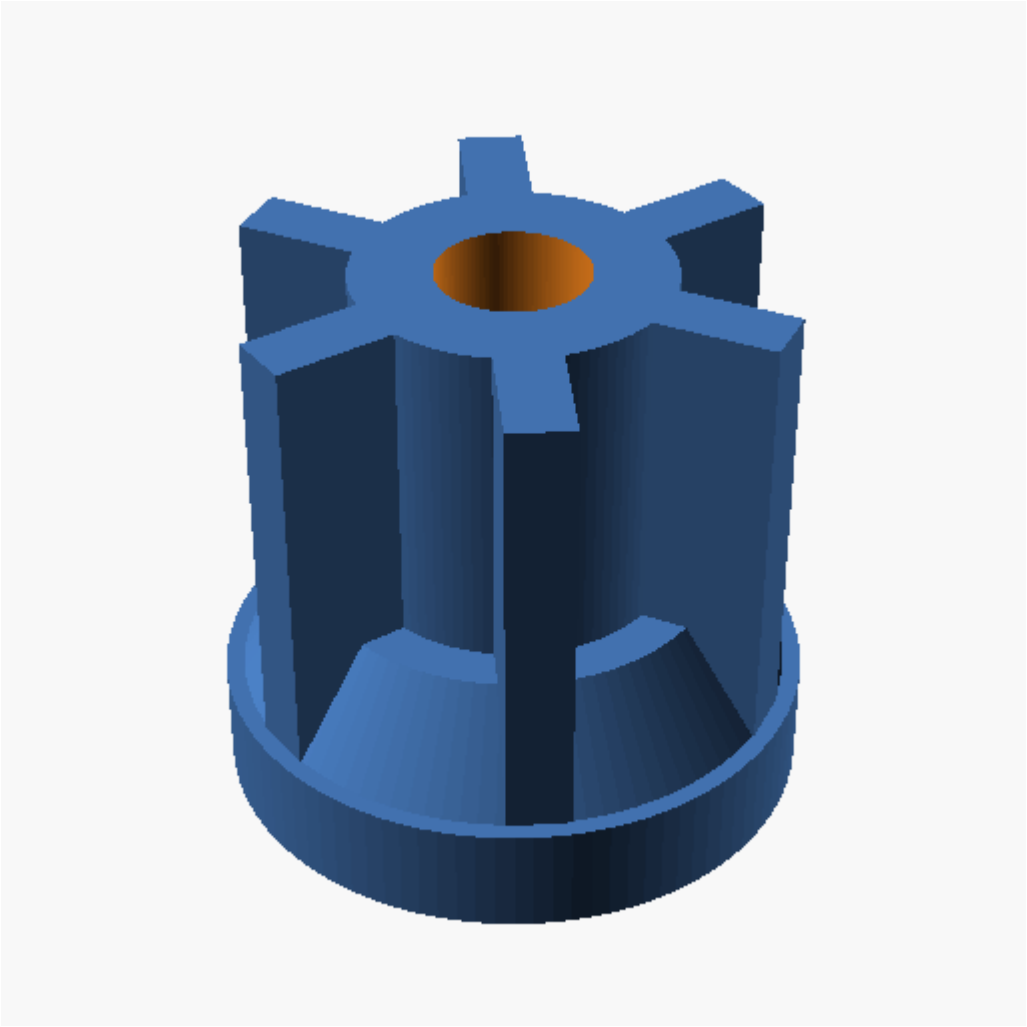

Figure 2. image

Listing 2. Openscad source

```
$fn=100;
totH=30;
baseH=6;
baseW=32;
wingW=3.5;
wingD=8;
centreD=17;

for (i = [0:360/6:360]) {
rotate([0,0,i]) translate([((baseW-2)/2)-wingD,-wingW/2,baseH])
cube([wingD,wingW,totH-baseH]);
}

difference(){
    union(){
        cylinder(h=totH,d=centreD);
        cylinder(h=6,d=32);
        translate([0,0,baseH]) cylinder(h=6,d1=baseW-2,d2=22);
```

```
        }
    translate([0,0,-.1]) cylinder(h=totH+.2,d=8.2);

    translate([0,0,-.1]) cylinder(h=8.1,d=15,$fn=6);
}
```

2.3. Object - fridgeDoorInterimHandle

Figure 3. image

Listing 3. Openscad source

```
$fn=360;
Height=100;
Diameter=18;
HolePos=(Height/2);
HoleDiam=3;
HoleDepth=10;
difference () {
    hull() {
        translate([0,0,0])
```

```
            cylinder(h=1,d2=Diameter,d1=Diameter-2);
        translate([0,0,Height])
            cylinder(h=1,d1=Diameter,d2=Diameter-2);
    }
    translate([0,0,HolePos])
        rotate([90,0,0])
            cylinder(h=HoleDepth,d=HoleDiam);
}
```

2.4. Object - Test

Just a simple example and also used as the logo for this repo.

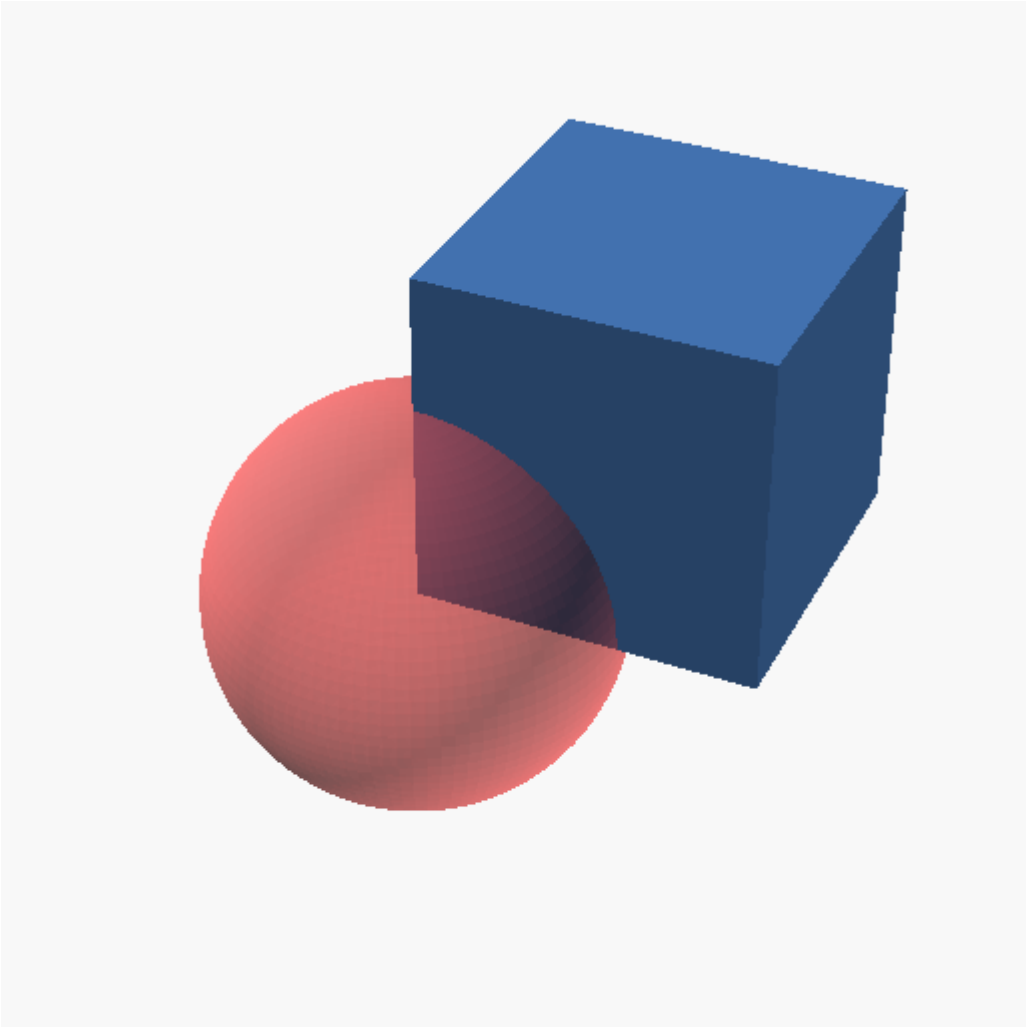

Figure 4. image

Listing 4. Openscad source

```
//just a simple test drawing
$fn=100;
cube([10,10,10]);
#sphere(d=12);
```

2.5. Object - fastener

This is a fastener for a writing Desk.
The idea is to add a magnet to hold it up and to print it so that it does not require a bearing.

- V1 is the first prototype for a first print test and fitting test
  - fits well and axle didn’t print free so need update
- V2 added a better axle but didn’t get printed
- V3 added a better cutout and is printed
  - The cutout is currently a dummy pending getting the axle to work to try it out with magnets taped into place
  - axle prints freely so moving on to screw holes, magnets, and covers
- V4 Added final OCD logo and screw caps etc.
  - Mounted and working.

Figure 5. image

Listing 5. Openscad source

```
$fn=100;
mainLength=50;
mainD=15;
mainH=10;
axleD=10;
axleDout=axleD+3;
ringH=2;

magnetX=17;
magnetY=5;
magnetZ=2;

module axle(xxlX,xxlY) {
    translate([0,0,-xxlY/2])cylinder(h=mainH+xxlY,d=axleD+xxlX);
    translate([0,0,((mainH-ringH)/2)]) cylinder(h=ringH,d=axleDout+xxlX);
    translate([0,0,(mainH/2)-((axleDout-axleD)/2+ringH/2)])
cylinder(h=(axleDout-axleD)/2,d1=axleD+xxlX,d2=axleDout+xxlX);
    translate([0,0,(mainH/2)+(ringH/2)]) cylinder(h=((axleDout-
axleD)/2),d2=axleD+xxlX,d1=axleDout+xxlX);
}
module clip() {
    difference() {
        union(){
            hull(){
                cylinder(d=mainD,h=mainH);
                translate([mainLength,0,0]) cylinder(d=mainD,h=mainH);
            }
            translate([7,-3.5,mainH]) linear_extrude (height=1.5) {
                text("OCD",size=8);
            }
        }
        //magnet
        translate([mainLength-magnetX,-magnetZ/2,(mainH-magnetY)/2+1])
cube([magnetX,magnetZ,magnetY+10]);
        //holder
        holderW=19;
        holderRin=33;
        holderRout=holderRin+holderW;
        difference(){
            translate([0,0,-.1]) cylinder(h=3+.1,r=holderRout);
            translate([0,0,-.11]) cylinder(h=3+.22,r=holderRin);
        }
    }
}
module magnetCap(){
    //magnet cap
    difference(){
        cylinder(h=2.8,d=11);
```

```
        translate([0,0,-.1]) cylinder(h=2,d=10);
    }
}
module screwCap() {
    //screwcap axle
    cylinder(h=2,d=7.5);
    translate([0,0,2]) cylinder(h=1,d=axleD);
}

//add the clip
difference () {
    clip();
    axle(1,1);
}

//add the axle and drill a hole in it for a srew
difference(){
    axle(0,0);
    translate([0,0,-.05]) cylinder(h=mainH+.1,d=4);
    translate([0,0,mainH/2]) cylinder(h=(mainH/2)+.1,d=7.5);
    //next two lines just a visual
    //#translate([0,0,mainH+2]) screwCap();
    //#translate([42,0,-.5]) magnetCap();
}
translate([0,-27,3]) rotate([0,180,0]) screwCap();
translate([0,-15,3]) rotate([0,180,0]) magnetCap();
```

2.6. Object - geoTest

This is not one of mine and I just kiked it as a good example.
I did correct some mistakes in it though.

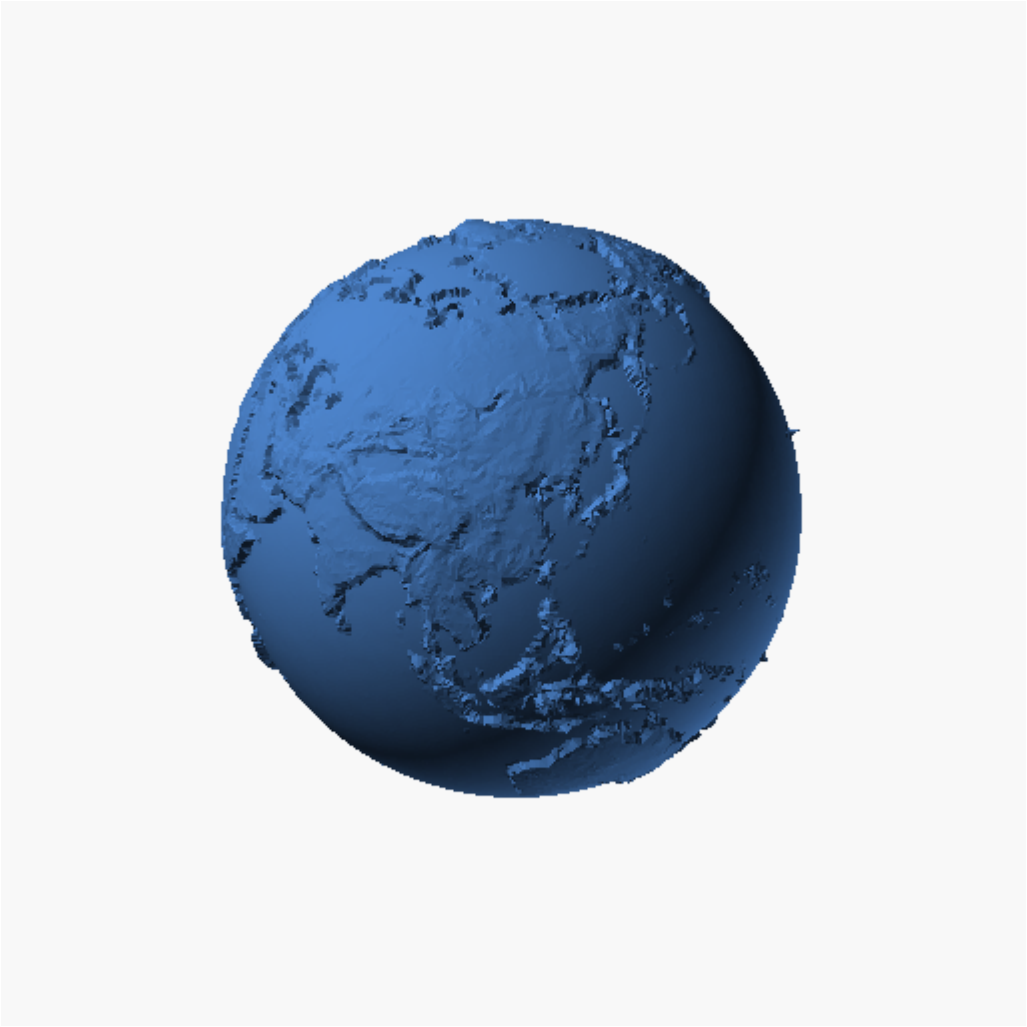

Figure 6. image

Listing 6. Openscad source

```
// Geody Planet 1 - SCAD
// Geody - https://www.geody.com/
// OpenSCAD - http://www.openscad.org/

wwrad=40; // Radius of the Planet
wrad=wwrad/20; // Radius of the Spot
wradp=wwrad-wrad/2; // Distance of the Spot from the center of the Planet
wres=50; // Resolution of the Spot

latx=48.782345; lonx=9.180819;

rotate(a=[0,0,270]) { import("geody_earthmap.stl", convexity=4); } // download
from https://www.geody.com/geody_earthmap.stl
// sphere(r=wwrad, $fn=wres); // Test Planet

translate([(-wradp)*cos(latx)*cos(lonx),(-
wradp)*cos(latx)*sin(lonx),wradp*sin(latx)]){sphere(r=wrad, $fn=wres);}
```

2.7. Object - KitchenDoorHoleStopper

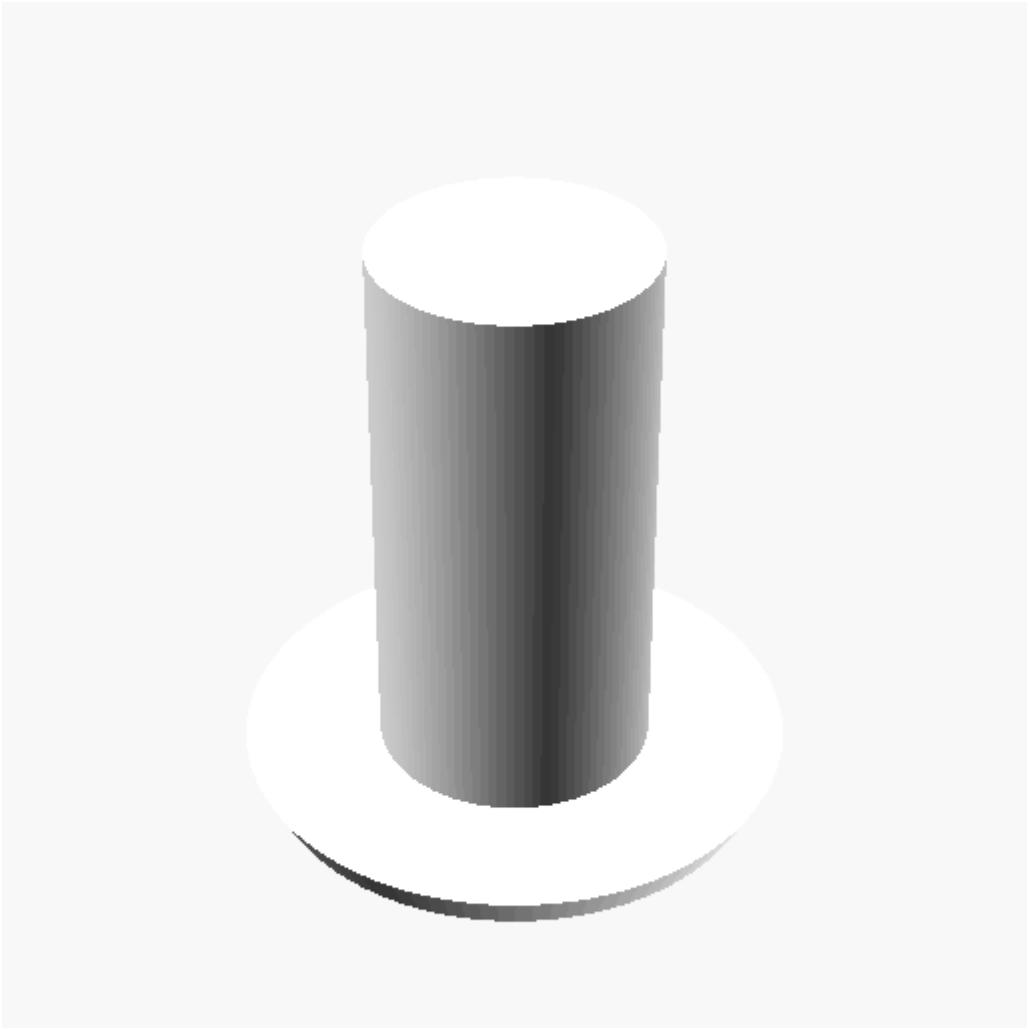

Figure 7. image

Listing 7. Openscad source

```
//plug for door hinge mounting hole (WHITE)
// door replaced by sliding glass door 27/11/2021
totDepth=15;
insertDiameter=7;
lidDiameter=14;
lidHeight=1;
$fn=100;
color ([1,1,1]) {
    cylinder(h=totDepth,d=insertDiameter);
    cylinder(h=lidHeight,d2=lidDiameter,d1=lidDiameter-lidHeight);
}
```

2.8. Object - strikeplate

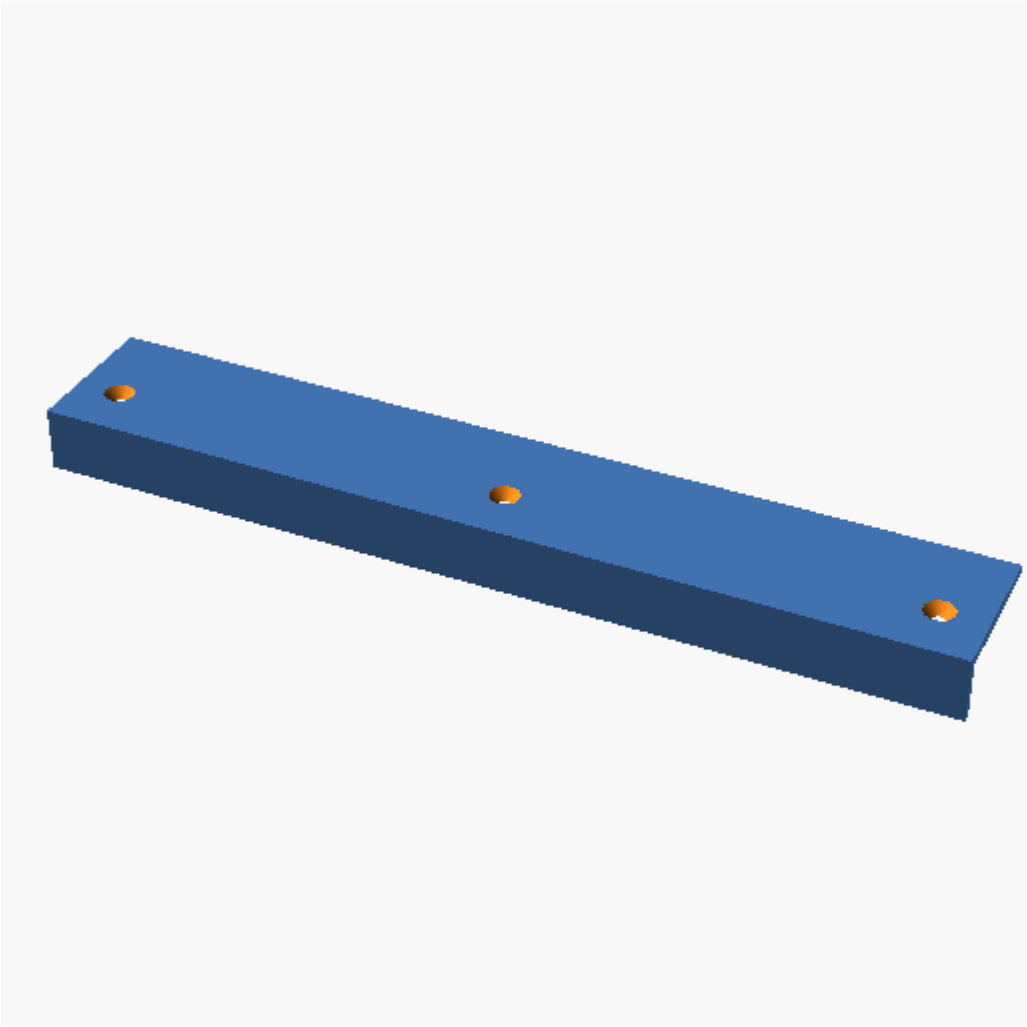

Figure 8. image

Listing 8. Openscad source

```
$fn=100;
SPlength=170;
SPwidth=28;
SPmaterialStrength=2;
;
module strikePlate () {
    cube ([SPlength,SPwidth,SPmaterialStrength]);
    translate ([0,0,-10])
        cube ([SPlength,SPmaterialStrength,10]);
}

module screw () {
    *cylinder(h=8,d=3);
    cylinder(h=3,d1=2,d2=7);
}

difference(){
    strikePlate();
```

```
    translate([8.5,10.5,-0.1]) screw();
    translate([85,10.5,-0.1]) screw();
    translate([SPlength-8.5,10.5,-0.1]) screw();
}
;
```

2.9. Object - CasingLED

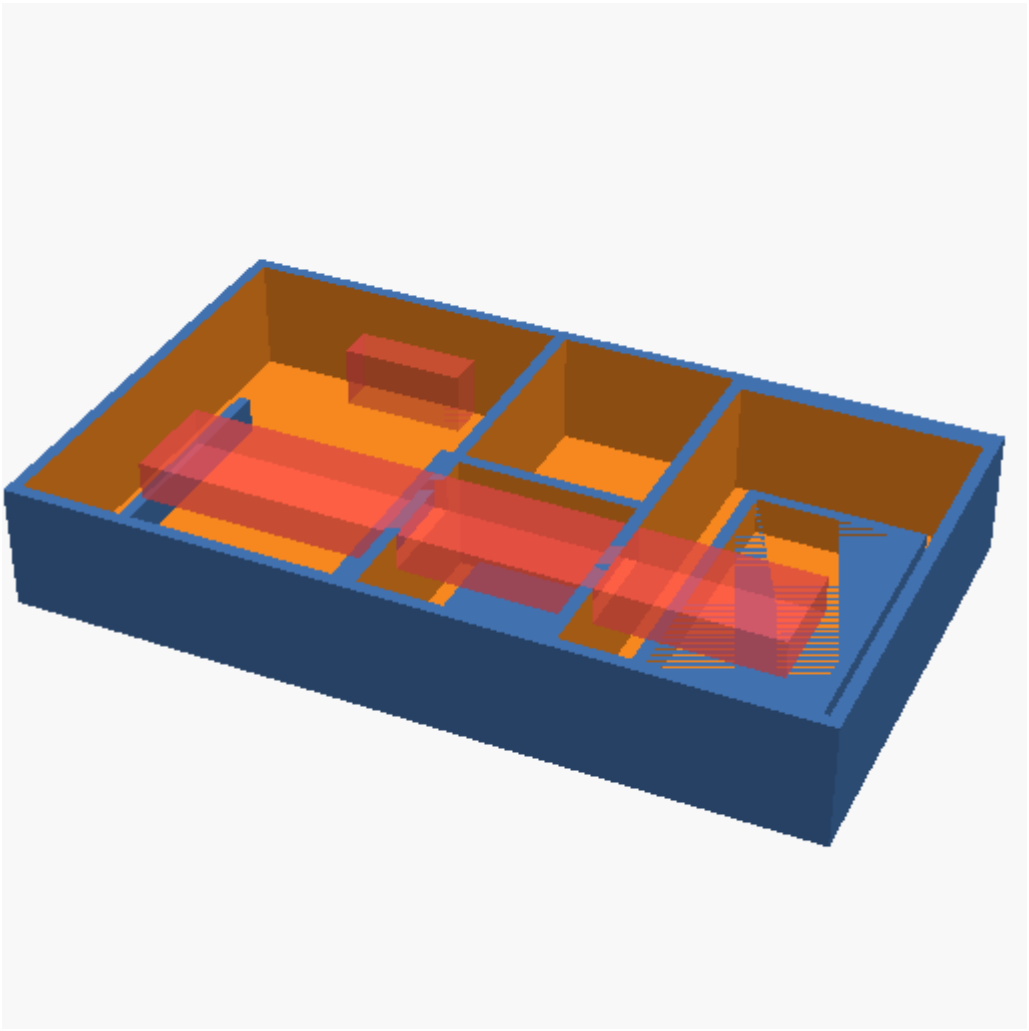

Figure 9. image

Listing 9. Openscad source

```
// OPENSCA Model for encloser for Tine's table
// Curently 3 devices Waatuino, 3.3v to 5v and esp wemos d1 mini
$fn=100;
module wemos(){
    difference(){
        union(){
            //wemos d1 mini 26mmx35mm h7mm
            cube([26,35,10],center=false);
            #translate([9,32,0]) cube([10,5,5],center=false);
```

```
        }
        translate([3,5,0]) cube([1,20,3]);
        translate([21,5,0]) cube([1,20,3]);
    }
}
module v5v3(){
    union(){
        difference(){
            //volatege level shifter 5v 3v 14mmx16mm h7mm
            cube([14,16,10],center=false);
            translate([3,3,0]) cube([8,10,3]);
        }
        translate([4,4,0]) cube([6,8,3]);

    }
}
module blanker(){
    //volatege level shifter 5v 3v 14mmx16mm h7mm
    cube([14,18,10],center=false);
}
module wattuino(){
    //wattuino arduino 5v clone 22mmx32mm h7mm
    union(){
        difference(){
            cube([19,34,10],center=false);
            translate([2.5,4,0]) cube([14,26,3],center=false);
        }
        translate([3.5,5,0]) cube([12,24,3],center=false);
    }
}
module casing(){
    //outer casing
    cube([63,37,10],center=false);
}
module cabling(){
    //cabling boom
    cube([50,8,3],center=false);
}
module xadow_pin(){
    union(){
        translate([0,0,0]) cylinder(h=1,r1=1,r2=1);
        translate([0,0,1]) cylinder(h=3,r1=1,r2=.5);
    }
}
module xadow_gsm(){
    difference(){
    union(){
            //Xadow madule
            //turns out the GSM module has exactly  25.37mm X 20.30mm / 1'' X
0.8''
```

```
            //approx 2mm hole 17.5mm x18mm
            cube([25.4,20.3,.75],center=false);
            translate([3,1.5,0]) xadow_pin();
            translate([21.4,1.5,0]) xadow_pin();
            translate([3,18.5,0]) xadow_pin();
            translate([21.4,18.5,0]) xadow_pin();
            }
            #translate([25.4,20.3,0]) cylinder(h=1,r1=1,r2=1);
        }

}
module enclosure(){
    //outer casing and substract
    difference(){
        casing();
        translate([1,1,1]) wemos();
        translate([28,1,1]) v5v3();
        translate([43,1,1]) wattuino();
        translate([28,18,1]) blanker();
        #translate([7,7,7]) cabling();
        }
}

enclosure();
*xadow_gsm();
```

2.10. Object - midletoninset

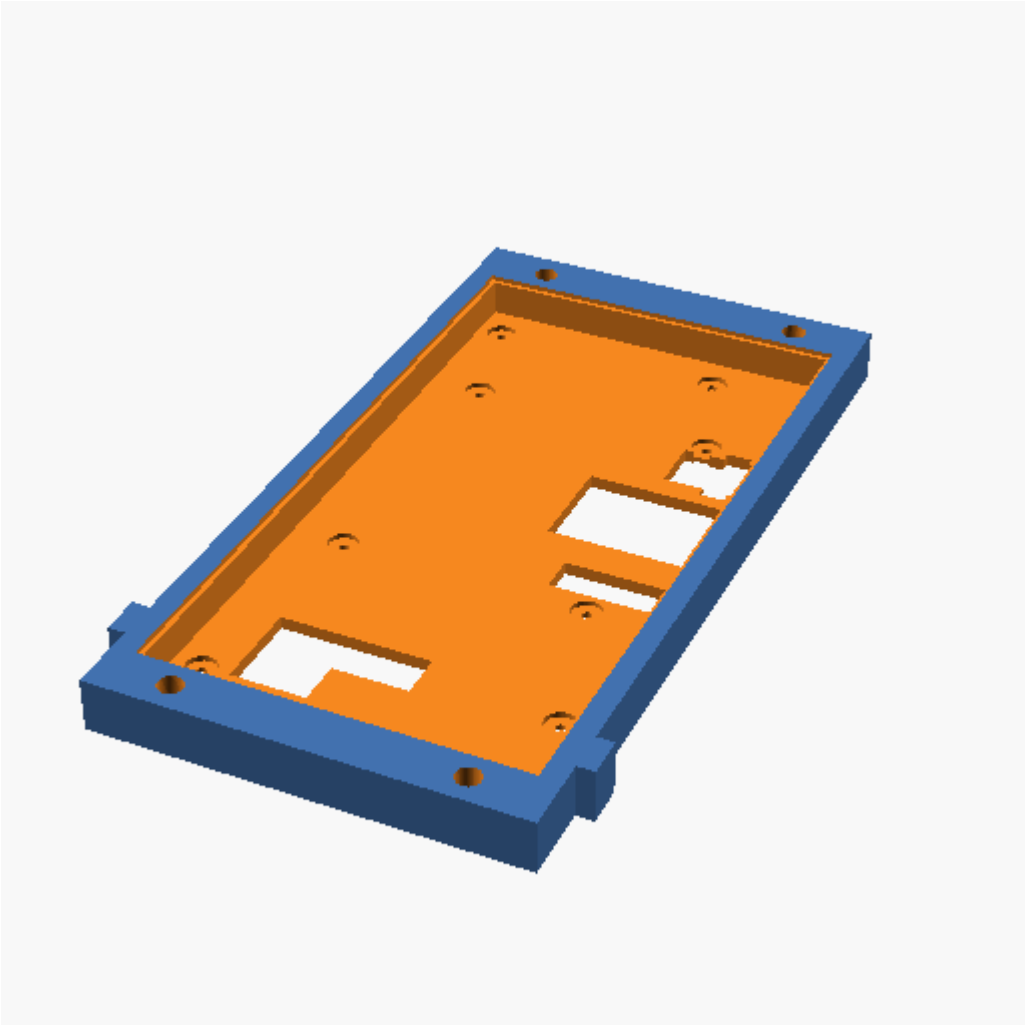

Figure 10. image

Listing 10. Openscad source

```
//This is an inlet for a whiskey presentation box from Midleton
$fn=50;
//Lower Notches
LowerNotchDepth=3.5;
LowerNotchLength=8;
LowerRNotchLengthOffset=15;
LowerLNotchLengthOffset=14.3;
module LLnotch(LowerLNotchLengthOffset){
    //Lower Left Notch
    translate([-LowerNotchDepth,LowerLNotchLengthOffset,0])
        cube([LowerNotchDepth,LowerNotchLength,BoxHeight]);
}
module LRnotch(LowerRNotchLengthOffset){
    //Lower Right Notch
    translate([BoxWidth,LowerRNotchLengthOffset,0])
        cube([LowerNotchDepth,LowerNotchLength,BoxHeight]);
}
```

```
//Variables for screen
ScreenTopY=75;
ScreenTopX=141;
ScreenTopZ=1;
ScreenEdge=1;
ScreenMaxDepth=7;
module waveshareHDMIscreen(wiggle){
    //full hd screen top face
    //and yes it has rounded corners but let's just start simple.
    union(){
        //Screen dimensions
        cube([ScreenTopY,ScreenTopX,ScreenTopZ+wiggle]);
        translate([ScreenEdge,ScreenEdge,-ScreenMaxDepth])
            cube([ScreenTopY-(2*ScreenEdge),ScreenTopX-
(2*ScreenEdge),ScreenMaxDepth+wiggle]);
        //connecting cable at the edge.
        translate([57,0,-ScreenMaxDepth]) cube([7,7,ScreenMaxDepth]);
        //USB for touch with offseted connector - wiggle through
        translate([12,19,-ScreenMaxDepth-10])
            cube([30,9,ScreenMaxDepth+10]);
        translate([12,10,-ScreenMaxDepth-10])
            cube([15,12,ScreenMaxDepth+10]);
        //USB for power - wriggle through
        translate([65,95,-ScreenMaxDepth-10])
            cube([5,15,ScreenMaxDepth+10]);
        translate([57,97,-ScreenMaxDepth-10])
            cube([17,11,ScreenMaxDepth+10]);
        //HDMI connector - Wriggle through might not work... might have to make
hole larger
        translate([44,72,-ScreenMaxDepth-10])
            cube([30,20,ScreenMaxDepth+10]);
        //Audio?
        translate([51,56.75,-ScreenMaxDepth-4])
            cube([23,7.5,ScreenMaxDepth+4]);
        //The screw holes
        Standoffs();
        //The mounting holes for the displaycover
        translate([0+10,0-5,-ScreenMaxDepth-6])
            cylinder(h=20,d=4.8);
        translate([0+10,ScreenTopX+5,-ScreenMaxDepth-6])
            cylinder(h=20,d=4.8);
        translate([ScreenTopY-10,0-5,-ScreenMaxDepth-6])
            cylinder(h=20,d=4.8);
        translate([ScreenTopY-10,ScreenTopX+5,-ScreenMaxDepth-6])
            cylinder(h=20,d=4.8);
    }
}
StandoffDepth=9;
StandoffSpace=1;
StandoffScrewHead=2;
```

```
module HolePeg(offset1){
    //standoff
    translate([0,0,-StandoffDepth+1]+offset1)
        cylinder(h=StandoffDepth-1,r=3.05);
    //screwwshaft
    translate([0,0,-StandoffDepth-StandoffSpace+1]+offset1)
        cylinder(h=StandoffDepth+StandoffSpace-1,r=1);
    //Screw head
    translate([0,0,-StandoffDepth-StandoffSpace-StandoffScrewHead+1]+offset1)
        cylinder(h=StandoffScrewHead,r=3);
}
module Standoffs(){
    //Outside holes
    //one
    *HolePeg([6,9,0]);
    HolePeg([6.5,9.75,0]);
    //the rest
    HolePeg([69,22,0]);
    HolePeg([6,132.5,0]);
    HolePeg([53,132.5,0]);

    //inside holes
    HolePeg([11.5,52.5,0]);
    HolePeg([60.5,52.5,0]);
    HolePeg([60.5,110.5,0]);
    HolePeg([11.5,110.5,0]);
}

// Midleton box measurements
//Real total Height
//BoxHeight=61;
//Display inset Height
BoxHeight=10.5;
//testprint
//BoxHeight=8.5;
BoxWidth=83.8;
LowerPartLength=162.5;
//testing value
//LowerPartLength=50;
LowerPartWallThickness=1.5;
LowerPartFloorThickness=1.5;
module Displaymodule() {
    //Lower part of the box
    difference(){
        //Outercube
        cube([BoxWidth,LowerPartLength,BoxHeight]);
        //subtract for inner space

*translate([LowerPartWallThickness,LowerPartWallThickness,LowerPartFloorThicknes
s])
```

```
            cube([BoxWidth-2*LowerPartWallThickness,LowerPartLength,BoxHeight-
(2*LowerPartFloorThickness)]);
    }
    LLnotch(LowerLNotchLengthOffset);
    LRnotch(LowerRNotchLengthOffset);
}
//Displaymodule();
//Standoffs();
//waveshareHDMIscreen();

// put it all together
difference(){
    Displaymodule();
    //Screen
    translate([(BoxWidth-ScreenTopY)/2,(BoxWidth-ScreenTopY)/2+6,BoxHeight-
ScreenTopZ])
        waveshareHDMIscreen(.1);
    //for testprint only
    *translate([-10,10,2.5])cube([100,130,15]);
    *translate([10,-10,2.5])cube([65,160,15]);
}
//remove for print... only for animation
*translate([((BoxWidth-ScreenTopY)/2),(BoxWidth-ScreenTopY)/2+6,(BoxHeight-
ScreenTopZ)+30*(1-$t)]) waveshareHDMIscreen(0);
```

2.11. Object - internal-volume

the internal Volume of a presentation box to test ideas on.

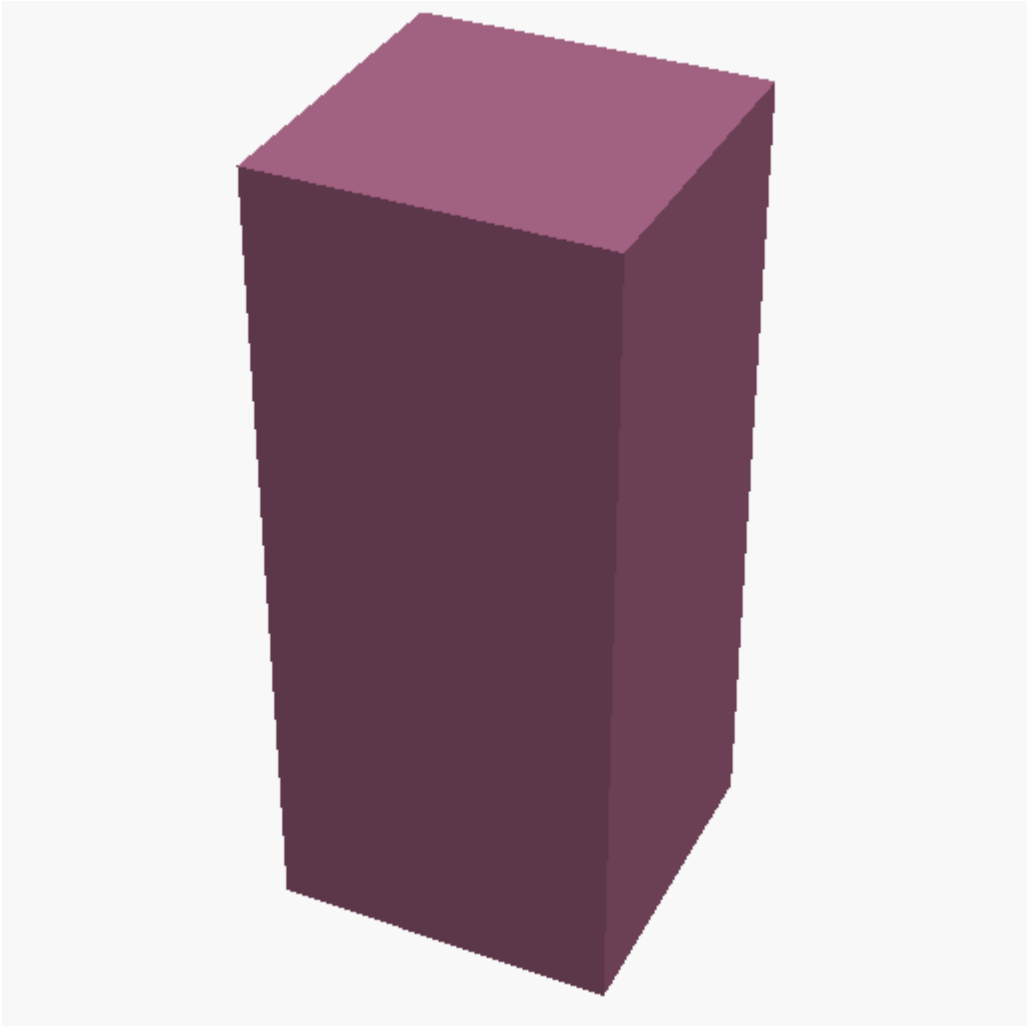

Figure 11. image

Listing 11. Openscad source

```
//inside of midleton wooden box with double doors
height=260;
width=111.1;
depth=108.5;

#cube([width,depth,height]);
```

2.12. Object - steeringaxle

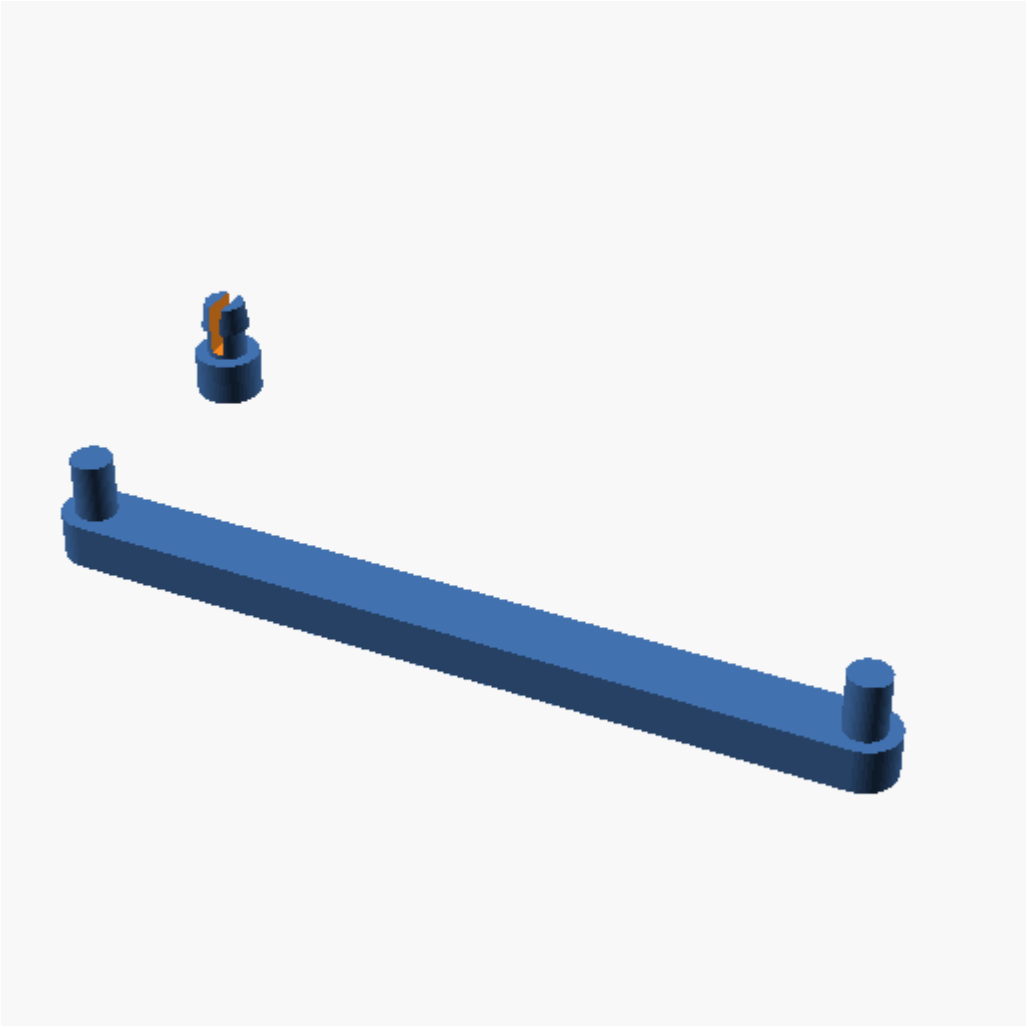

Figure 12. image

Listing 12. Openscad source

```
$fn=50;

module tabbedCylinder(){
    difference(){
        union (){
            cylinder(h=2,d1=3,d2=3);
            cylinder(h=4.6,d1=1.8,d2=1.8);
            translate ([0,0,3.4]) cylinder(h=1.2,d1=2.2,d2=1.8);
        }
        translate ([0,0,3.5]) cube([.6,2.5,2.5],center=true);
    }
}
module EndCylinder(){
        union (){
            cylinder(h=2,d1=3,d2=3);
            cylinder(h=4.6,d1=1.8,d2=1.8);
        }
}
```

```
module steeringAxle(){
    //axis
    translate ([1.5,0,0]) cube([34,3,2]);
    //connector
    translate ([1.5,1.5,0]) EndCylinder();
    //connector
    translate ([35.5,1.5,0]) EndCylinder();
}

steeringAxle();
translate([0,15,0]) tabbedCylinder();
```

2.13. Object - buttonBack

Button backing for the monitor.

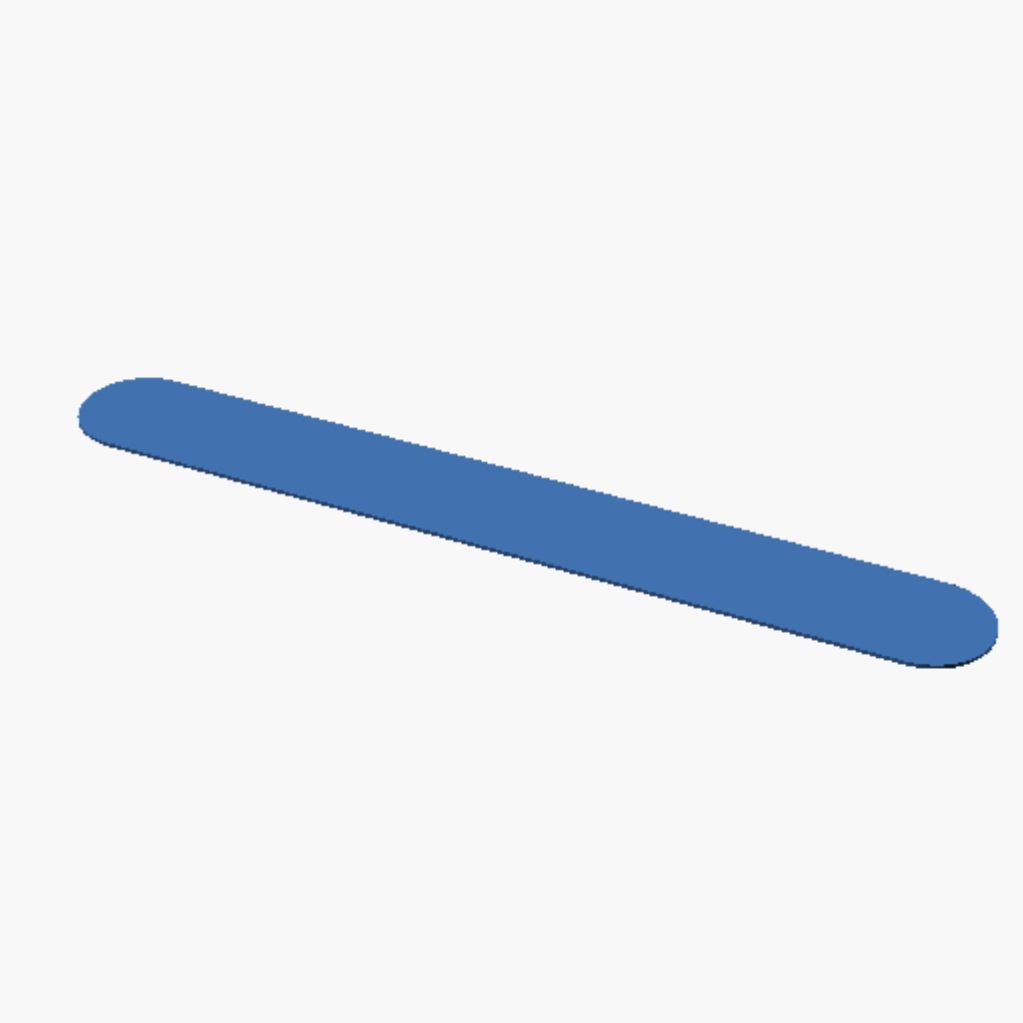

Figure 13. image

Listing 13. Openscad source

```
$fn=100;
holeEndD=16.1;
holeLength=120.1;
buttonHolderHeight=.5;

hull(){
    cylinder(h=buttonHolderHeight,d=holeEndD);
    translate([holeLength-holeEndD,0,0])
cylinder(h=buttonHolderHeight,d=holeEndD);
}
```

2.14. Object - screen_mounting_tabs

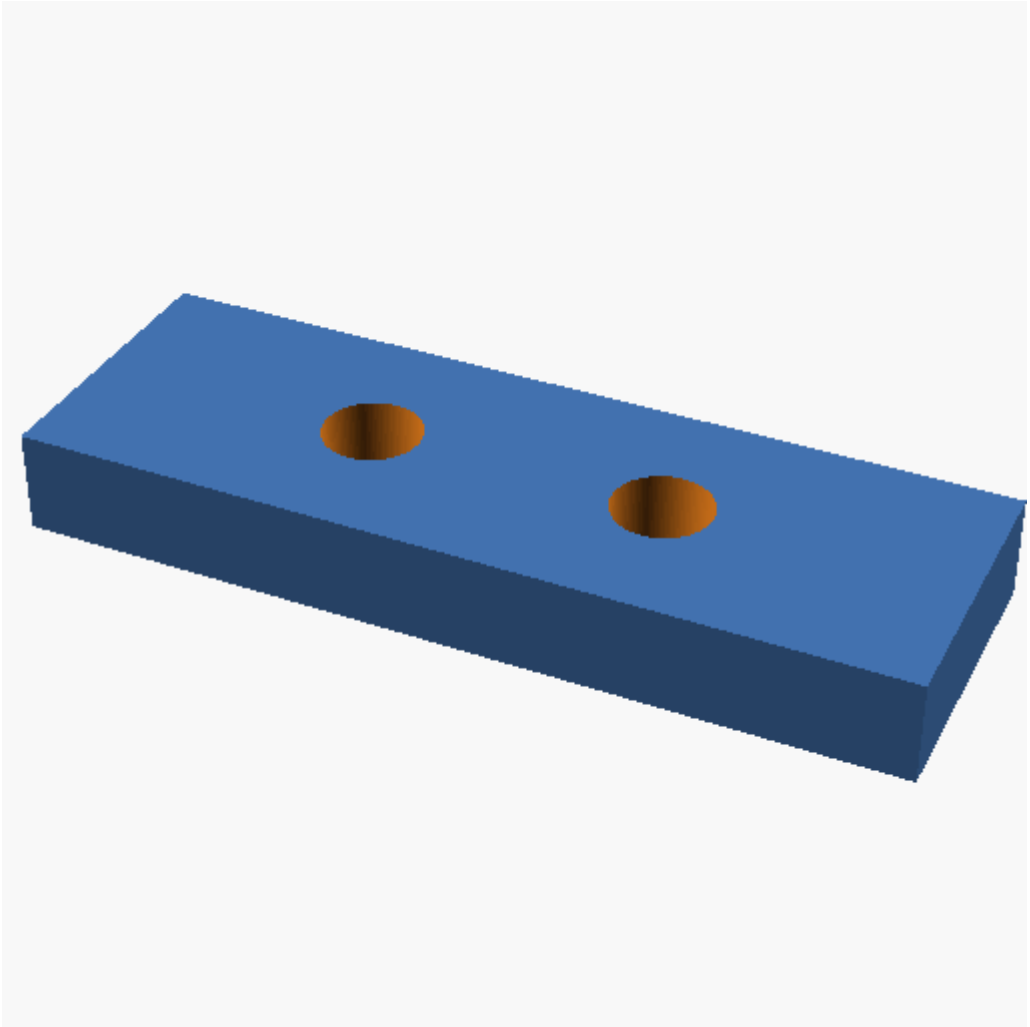

Figure 14. image

Listing 14. Openscad source

```
$fn=100;
tab_height=.3;
```

```
tab2bottom=2.4;

plus=.1; // this is to make parts larger than the hole they are to make
plusH=plus/2;

hole_d=2;
tab1_hole_spacing=8;
tab2_hole_spacing=6;
shim_height=tab2bottom-tab_height;
shim_depth=6;
shim_width=18;

module tab1(spacing){
    difference(){
        cube([shim_width,shim_depth,shim_height]);
        translate([shim_width/2-spacing/2,(shim_depth/2),-plusH])
            union() {
                cylinder(h=shim_height+plus,d=hole_d);
                translate([spacing,0,0]) cylinder(h=shim_height+plus,d=hole_d);
            }
    }
}

//for the bottom tabs
//tab1(tab1_hole_spacing);

//for the top tabs
tab1(tab2_hole_spacing);
```

2.15. Object - case

A case for an odroid handgeld console and accessories.

The edges are rounded and there are cutouts for the parts that protrude from the console.

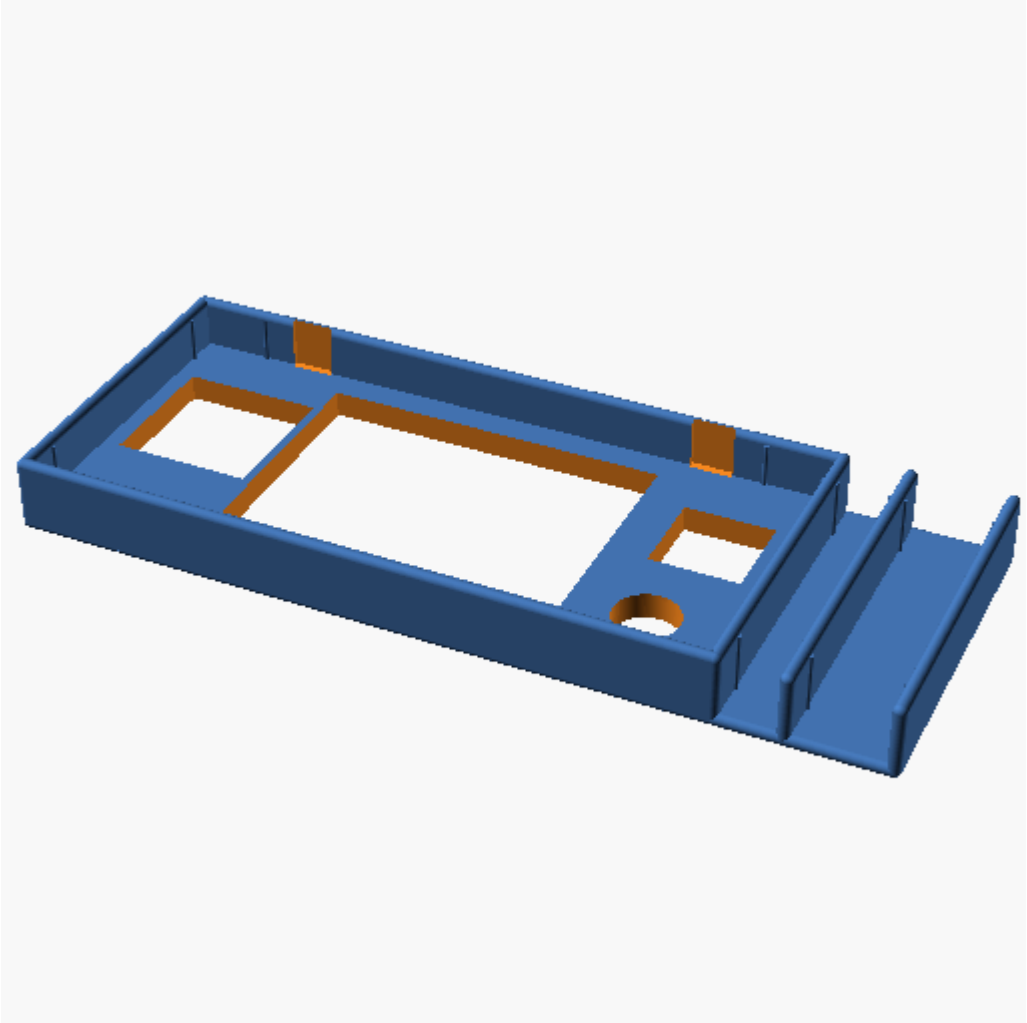

Figure 15. image

Listing 15. Openscad source

```
include <Library-container.scad>
caseRim=3;
holdersR=.7;
$fn=100;
kbD=82;
kbW=210;
kbH=7;
library=true;

odH=10;
odW=156;
odD=73;

odJSW=28-13;
odJSR=odJSW/2;
odJSoffX=13;
odJSoffY=11;
```

```
odJSH=6;
odBRW=118-38;
odBRD=12-5;
odBRoffX=38;
odBRoffY=5;
odCRW=28-7;
odCRD=58-37;
odCRoffX=7;
odCRoffY=37;
odTBW=152-120;
odTBD=61-27;
odTBoffX=120;
odTBoffY=27;
odTLoffX=23;
odTLoffY=odD;
odTLW=33-23;
odTLD=2;
odTH=20;
odTRoffX=123;
odTRoffY=odD;
odTRW=odTLW;
odTRD=odTLD;
odDPW=odBRW;
odDPD=67-14;
odDPoffY=14;
odDPoffX=odBRoffX;

//the odroid travel case with cutouts for buttons etc
difference(){
    //the container itself
    translate([caseRim/2,caseRim/2,odJSH])
containerOpenLid(odW,odD,odH,caseRim,odJSH-caseRim,"nibY",.6);
    offset=.01;
    //the cutouts
    translate([1.5,.75,-offset/2]) union() {
        translate([odW-odJSR*2-odJSoffX,odJSoffY,0]+[odJSR,odJSR,0])
cylinder(h=odJSH+offset,r=odJSR);
        translate([odW-odBRW-odBRoffX,odBRoffY,0])
cube([odBRW,odBRD,odJSH+offset]);
        translate([odW-odCRW-odCRoffX,odCRoffY,0])
cube([odCRW,odCRW,odJSH+offset]);
        translate([odW-odTBW-odTBoffX,odTBoffY,0])
cube([odTBW,odTBW,odJSH+offset]);
        translate([odW-odTLW-odTLoffX,odTLoffY,odJSH-.1])
cube([odTLW,odTLD,odTH+offset]);
        translate([odW-odTRW-odTRoffX,odTRoffY,odJSH-.1])
cube([odTRW,odTRD,odTH+offset]);
        translate([odW-odDPW-odDPoffX,odDPoffY,0])
cube([odDPW,odDPD,odJSH+offset]);
    }
```

```
}
//add on some slots for peripherals
floorDepth=0;
//microuter slot
translate([caseRim/2+caseRim+odW,caseRim/2,floorDepth])
    containerVertSlot(12,odD,odH+odJSH,caseRim,floorDepth-caseRim,"nibY",.6);
//micro USB 3 Port Hub
translate([caseRim/2+2*caseRim+odW+12,caseRim/2,floorDepth])
    containerVertSlot(19.5,odD,odH+odJSH,caseRim,floorDepth-caseRim,"nibY",.6);
```

2.16. Object - RPi_zero_mount

This also is NOT one of mine but I've cleaned it up a bit as it wasn't displaying correctly. I needed it for a pi cluster and as it's quite good I didn't reinvent the wheel here.

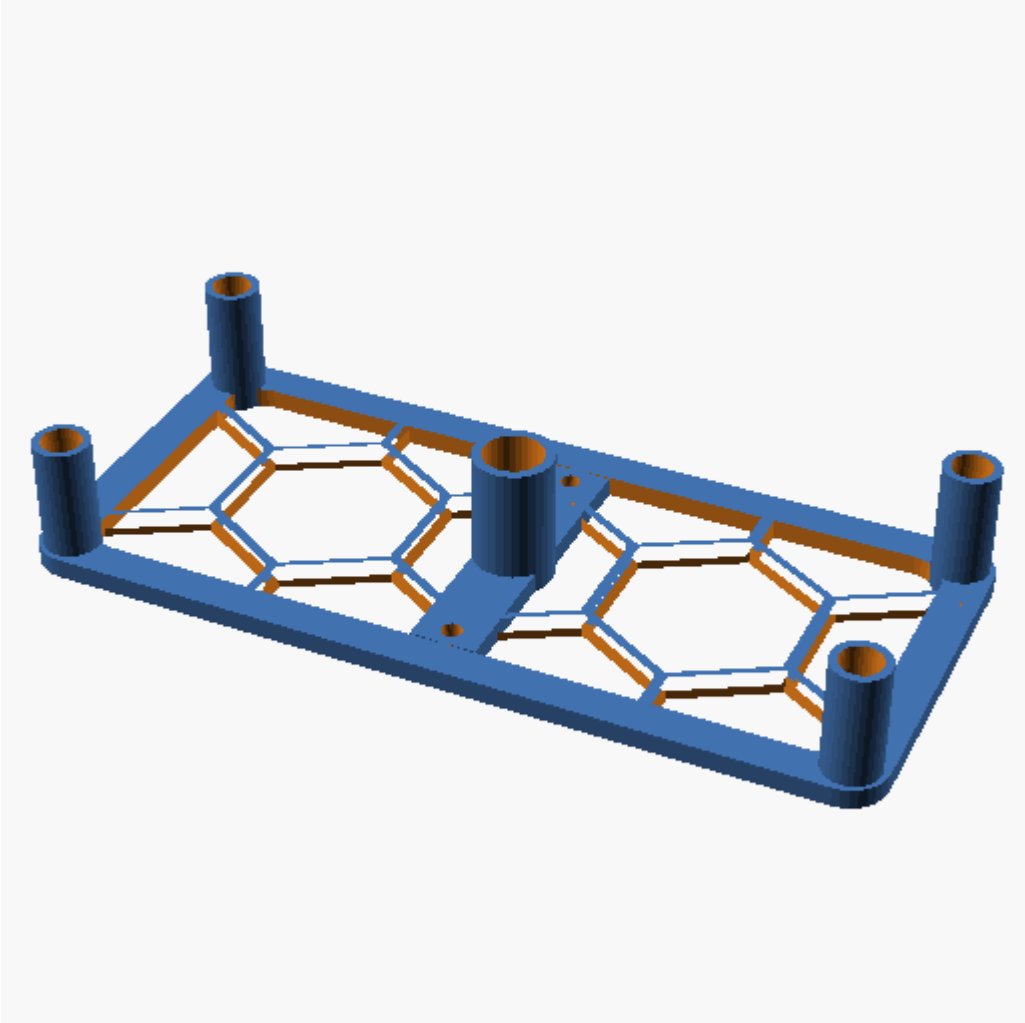

Figure 16. image

Listing 16. Openscad source

```
/* [Base] */
//type = 1; //[1:"Hexagon Grid",2:"Skeleton"]
```

```
/* [Hidden] */
$fn = 32;
zero_x = 64;
zero_y = 29;
zero_z = 1.5;

mounts_z = 8.5;
mounts_radius = 2.1;
screwholes = 2.6;
screwholes_radius = 1.5;
screwholes_depth = 10.7;

base_x = zero_x - 2*3.0;
base_y  = zero_y  - 2*3.0;
base_z = zero_z;
mount_x = zero_x/2 - screwholes;
mount_y = zero_y/2 - screwholes;
mount_z = zero_z + mounts_z;
screwhole_base_z = mount_z - screwholes_depth;

module baseplate(){
    translate([-zero_x/2+3,-zero_y/2+3,0])
        minkowski(){
            cube([base_x,base_y,base_z/2]);
            cylinder(r=3.0,h=base_z/2);
    }
}

module mounts(){
    translate([0,0,0]) cylinder(r=3.0,h=mount_z);
    translate([-mount_x, -mount_y, 0]) cylinder(r=mounts_radius,h=mount_z);
    translate([-mount_x, +mount_y, 0]) cylinder(r=mounts_radius,h=mount_z);
    translate([+mount_x, -mount_y, 0]) cylinder(r=mounts_radius,h=mount_z);
    translate([+mount_x, +mount_y, 0]) cylinder(r=mounts_radius,h=mount_z);
}

module hexagon (radius=8,latticeWidth=8,latticeLength=16,spacing=1,height=2){
    linear_extrude(height) {
        for(j = [0:latticeWidth-1]) {

translate([((sqrt(3)*radius)+spacing)/2*(j%2),sqrt((pow(((sqrt(3)*radius)+spacin
g),2))-(pow((((sqrt(3)*radius)+spacing))/2,2)))*j,0]) {
                for(i = [0:latticeLength-1]) {
                    translate([(sqrt(3)*radius*i)+spacing*i,0,0]) {
                        rotate([0,0,30]) {
                            circle(radius, $fn = 6);
                        }
                    }
                }
```

```
            }
        }
    }
}

module hex_border(){
    difference(){
        baseplate();
        holes();
        translate([0,0,-.01]) scale([0.9,0.8,1.02]) baseplate();
    }
}

module holes(){
    translate([0,0,screwhole_base_z+0.4]) {
        translate([0,0,0]) cylinder(r=screwholes_radius*1.5,h=screwholes_depth);
        translate([-mount_x,-mount_y,0])
cylinder(r=screwholes_radius,h=screwholes_depth);
        translate([-mount_x,+mount_y,0])
cylinder(r=screwholes_radius,h=screwholes_depth);
        translate([+mount_x,-mount_y,0])
cylinder(r=screwholes_radius,h=screwholes_depth);
        translate([+mount_x,+mount_y,0])
cylinder(r=screwholes_radius,h=screwholes_depth);
    };
}

module result(){
    difference(){
        translate([-2.5,-base_y/2,0]) cube([5,base_y,base_z]);
        translate([0,10,-3]) cylinder(d=1.5,h=10);
        translate([0,-10,-3]) cylinder(d=1.5,h=10);
        holes();
    }
    translate([0,0,0])
    hex_border();
    difference(){
        translate([0,0,0]) cylinder(r=3.0,h=mount_z);
        holes();
    }
    difference(){
        mounts();
        holes();
    }
    difference(){
        baseplate();
        holes();
        translate([-zero_x/2-5,-zero_y/2+1.5,-0.1]) hexagon();
    }
}
```

```
difference(){
    result();
    translate([0,10,-3]) cylinder(d=1.5,h=10);
    translate([0,-10,-3]) cylinder(d=1.5,h=10);
}
```

2.17. Object - schuko

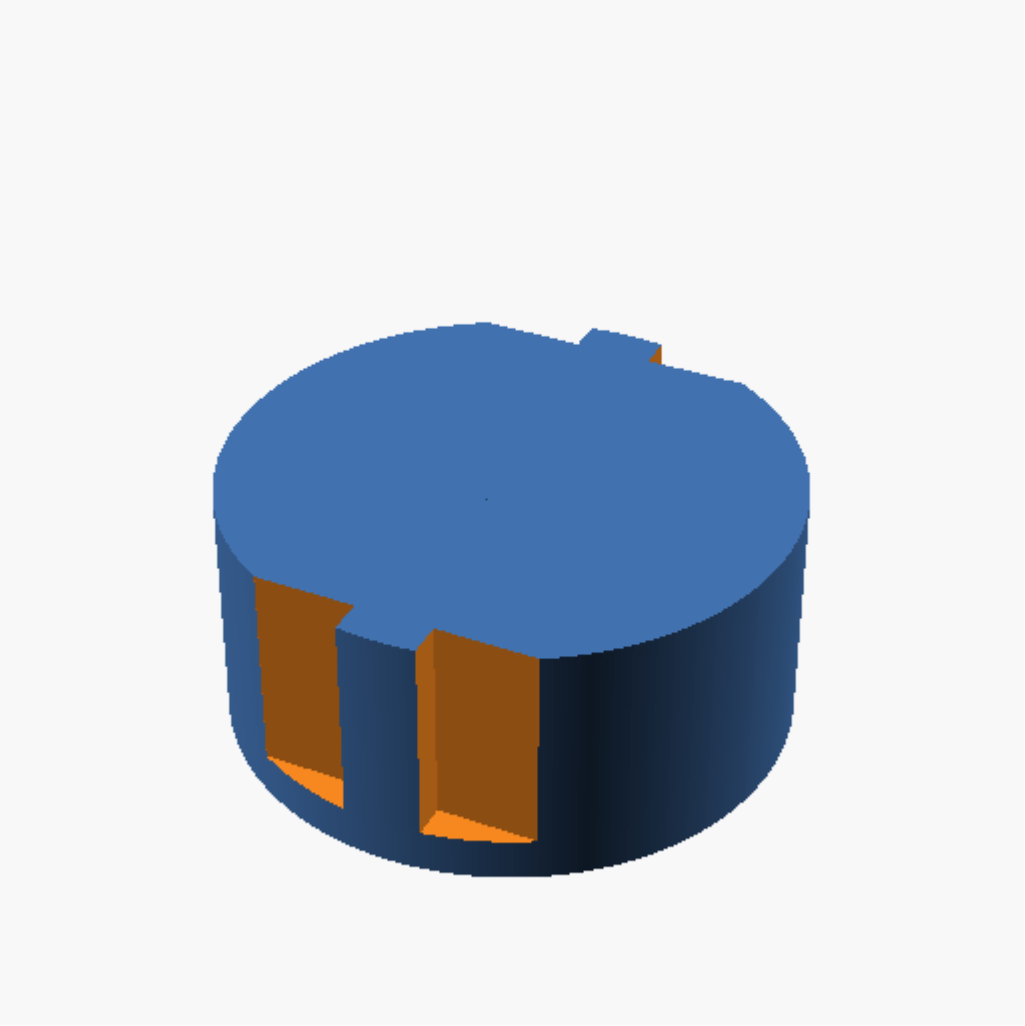

Figure 17. image

Listing 17. Openscad source

```
/*
 Parametric Schuko CEE 7/3 socket

 Copyright 2017 Anders Hammarquist <iko@iko.pp.se>
 Licensed under Creative Commons - Attribution - Share Alike

 Made using a negative "profile punch" that can be extracted
 and used to "punch" a schuko socket into any sufficiently large solid.
```

```
*/

// Diameter of cover
coverdiameter = 50; // [50:100]

// Thickness of cover
coverthickness = 4.8; // [2:0.2:15]

// Center screw offset (extreme values disables screw hole)
screwoffset = 0; // [-11:0.5:11]
// This is the socket punch. Includes cut-out for
// earthing contacts and holes for pins and center screw.
// Maximum screw offset from center is 10mm (use a larger
// value to remove the hole for the screw).
module schuko(screwoffset=0, screwdia=3.5, screwhead=6.5, screwsink=3)
{
    module earthing()
    {
        intersection() {
            union() {
                translate([-22,-2,3])
                    cube([6,4,20]);
                translate([-19,-2,17.5])
                    rotate([0,-30,0])
                        cube([15, 4, 4]);
            }
            translate([-22,-3,3])
                cube([22,6,20]);
        }
    }

    difference() {
        union() {
            translate([0,0,-1])
                cylinder(r=39/2, $fn=300, h=18.5);

            // Earthing cutouts
            color([1,1,1]) {
                earthing();

                rotate([0,0,180])
                    earthing();

            }

            // Power pins
            translate([0,10,0])
                cylinder(r=7/2, $fn=300, h=30);
            translate([0,-10,0])
                cylinder(r=7/2, $fn=300, h=30);
```

```
            if (abs(screwoffset) <= 10) {
                // Center screw
                translate([screwoffset,0,0])
                    cylinder(r=screwdia/2, $fn=300, h=30);
                translate([screwoffset,0,0])
                    cylinder(r=screwhead/2, $fn=300, h=17.5+screwsink);
            }
        }

        // Side key profile
        translate([5.4/2,16.9,3])
            cube([7,3,20]);
        translate([-5.4/2-7,16.9,3])
            cube([7,3,20]);
        translate([5.4/2,-20.4,3])
            cube([7,3.5,20]);
        translate([-5.4/2-7,-20.4,3])
            cube([7,3.5,20]);
    }
}
difference () {

difference () {
    cylinder(r=39/2, $fn=300, h=17.5);
translate ([-27.3/2,-27.8/2,0]) cube([27.3,27.8,10]);
rotate([0,0,0]){
    difference(){
        union() {
            translate([0,0,0])
                cylinder(r=44/2, $fn=300, h=21.5);
            // Lip
            rotate_extrude($fn=100) {
                polygon(points=[[0,0], [coverdiameter/2,0],
                [coverdiameter/2+0.2*coverthickness,coverthickness],
                [0,coverthickness]]);
            }

            // Pin guard: 9.5 x 28.5 x 3mm (rounded ends)
            translate([-4.75,-14.25,21.5])
                cube([9.5, 28.5, 3]);

            // center screw standoff: 6 x 2.5 (above pin guard) x 2 - 3
            // ( 8mm inside, 14 - 12.2 mm outside)
            translate([-7.25, -3, 21.5])
                cube([2.5, 6, 5.5]);
            translate([4.75, -3, 21.5])
                cube([2.5, 6, 5.5]);

        }
```

```
        schuko(screwoffset=screwoffset);
    }
}
}

}
```

2.18. Object - tesa

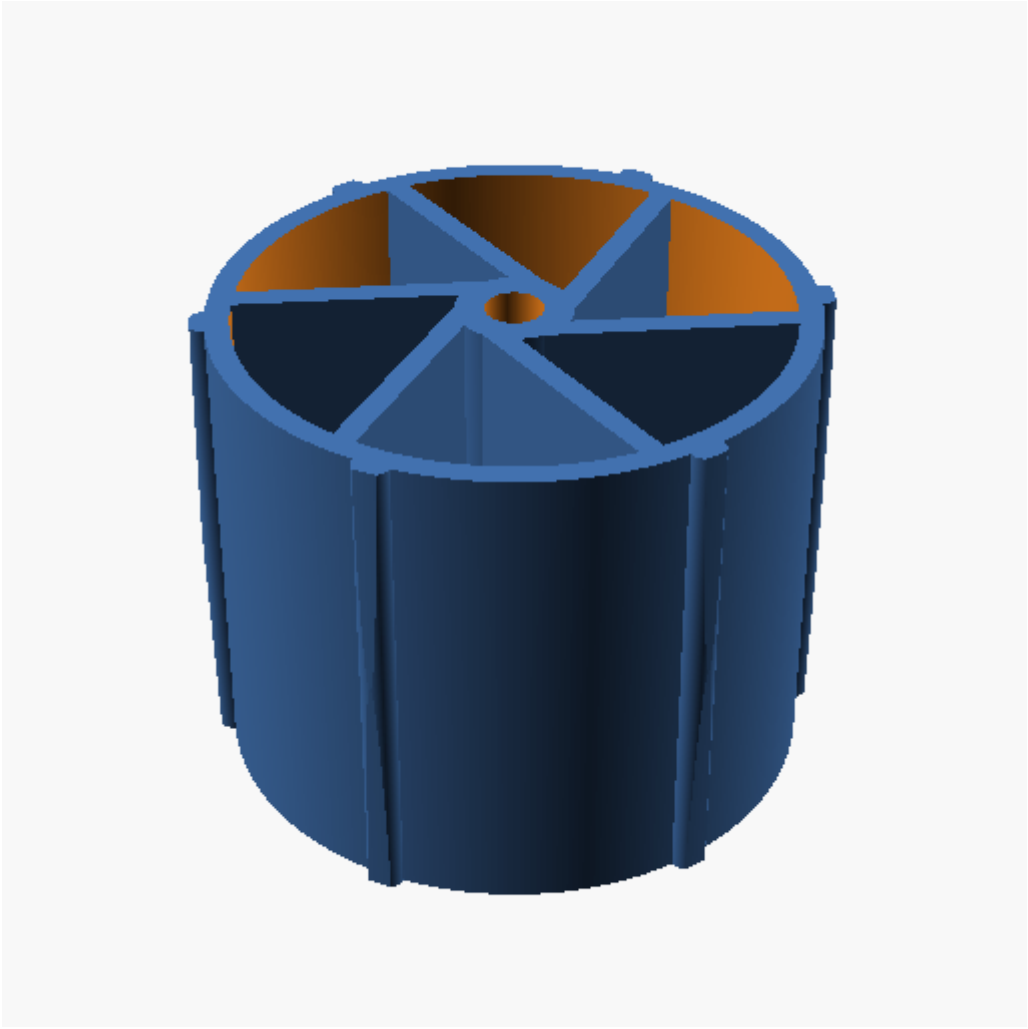

Figure 18. image

Listing 18. Openscad source

```
//Tesa roller ersatzroller
//celotape roller

$fn=360;
height=20;
outsideD=24.5;
outsideDepth=2;
axleD=2.4;
```

```
hubD=axleD*2;
nubR=.75;
module taper(){
difference(){
union(){
    translate([0,0,-.1]) cylinder(h=height+.2,d=outsideD+2+nubR);
}
union(){
    translate([0,0,-
.11])cylinder(h=height/2+.22,d1=outsideD+1.5*nubR,d2=outsideD+2*nubR);
    translate([0,0,height/2+.1])
cylinder(h=height/2+.1,d1=outsideD+2*nubR,d2=outsideD+1.5*nubR);
}
}
}
difference(){
union(){
    //outside
    difference(){
        cylinder(h=height,d=outsideD);
        translate([0,0,-.1])cylinder(h=height+.2,d=outsideD-outsideDepth);
    }
    //HUB
    difference(){
        cylinder(h=height,d=hubD);
        translate([0,0,-.1])cylinder(h=height+.2,d=axleD);
    }
    //nubs
    for (i = [0:5]) {
        translate([sin(360*i/6)*outsideD/2, cos(360*i/6)*outsideD/2, 0 ])
        rotate([0,0,0])cylinder(h = height/2, r=nubR);
    }
    for (i = [0:5]) {
        translate([sin(360*i/6)*outsideD/2, cos(360*i/6)*outsideD/2, height/2 ])
        cylinder(h = height/2, r=nubR);
    }
    //spokes
    for (i = [0:360/6:360]) {
        rotate([0,0,i])translate([1.2,0,0])cube([1,(outsideD/2)-
(axleD/2.4),height]);
    }
}
taper();
}
```

2.19. Object - Knurl

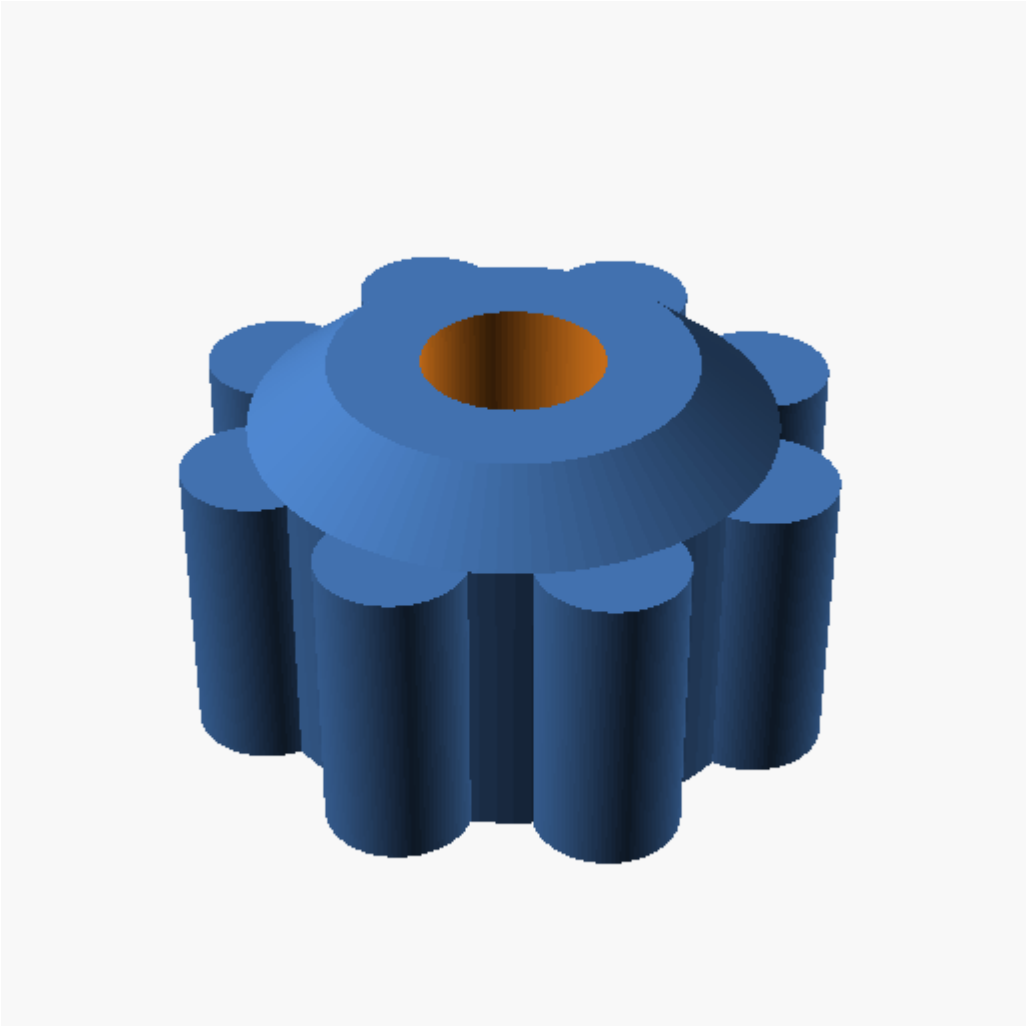

Figure 19. image

Listing 19. Openscad source

```
diam=9;
diamOut=14.4;
holeDiam=5;
height=8.2;
$fn=100;
knurlNum=8;
knurlInc=360/knurlNum;
knurlDiam=4;
insetHeight=5;
insetDiam=5;
totalHeight=10;


module knob() {
difference(){
    union(){
        cylinder(h=height, d=diamOut);
        translate([0,0,height]){
```

```
        cylinder(h=totalHeight-height,r1=7.2,r2=5);
        }
    }
    translate([0,0,-.5])
        cylinder(h=totalHeight+1, d=holeDiam);
    translate([0,0,-.5]){
        cylinder(h=totalHeight-insetHeight+.5,d1=holeDiam+5,d2=holeDiam);
    }
    }

for(i=[0: knurlInc: 360]){
   rotate([0,0,i])
        translate([diamOut/2,0,0])
            cylinder(h=height, d=knurlDiam);
}
    }


knob();
```

3. To do

Right now the github source is not perfect as the readme does not display the images when viewed in github.

Need to add further process steps for the images like meshlabserver to do furhter processing:

- glass rendering
- cleaning up the mesh
- Simplifying the mesh
- Stats

Need to add animation options.

☑ Need to add text display option for each item.

Need to add view parameters as options.