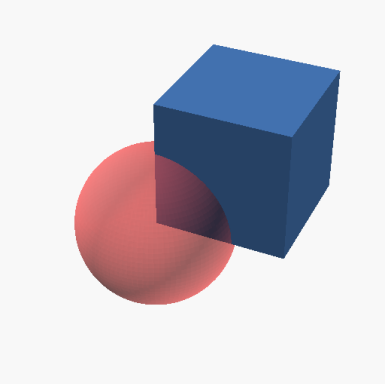

OpenScad examples

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

Table of Contents

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

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.

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

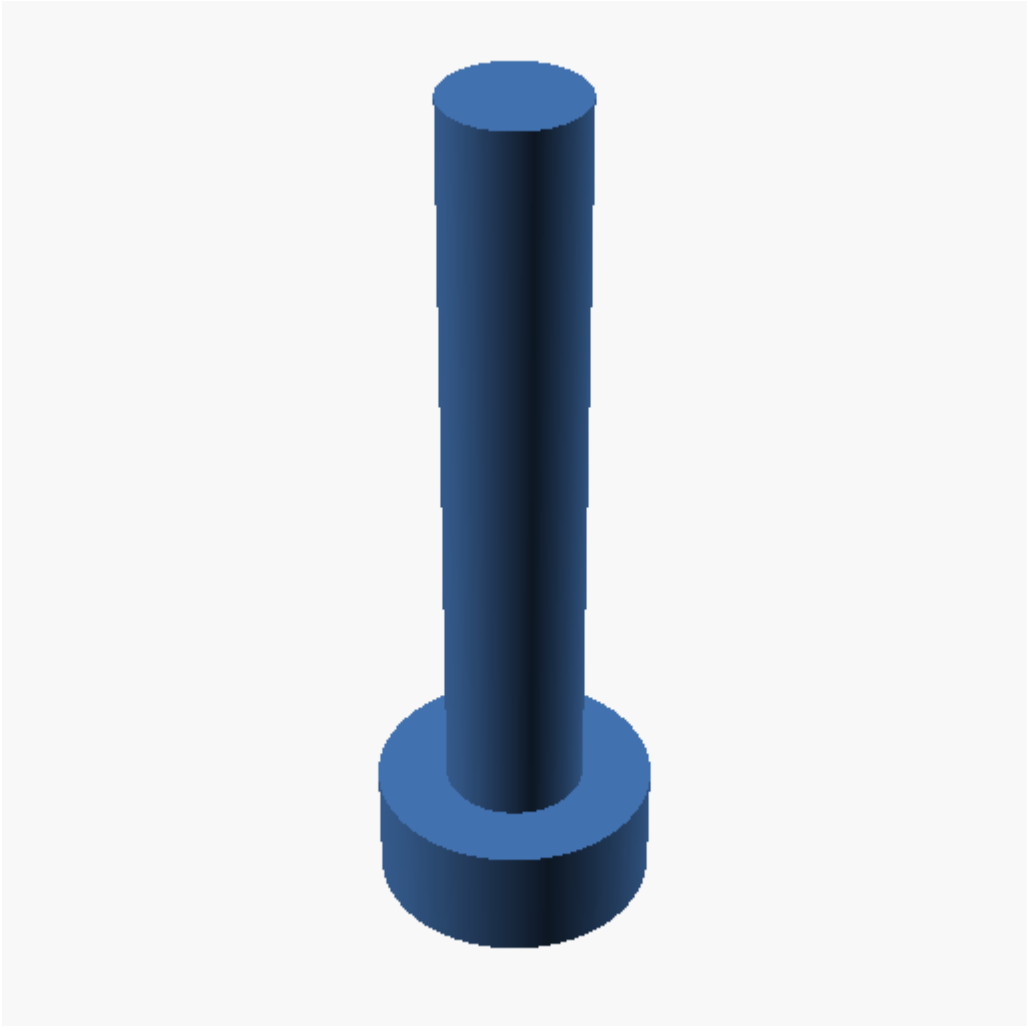

Figure 1. image

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 - CasingLED

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

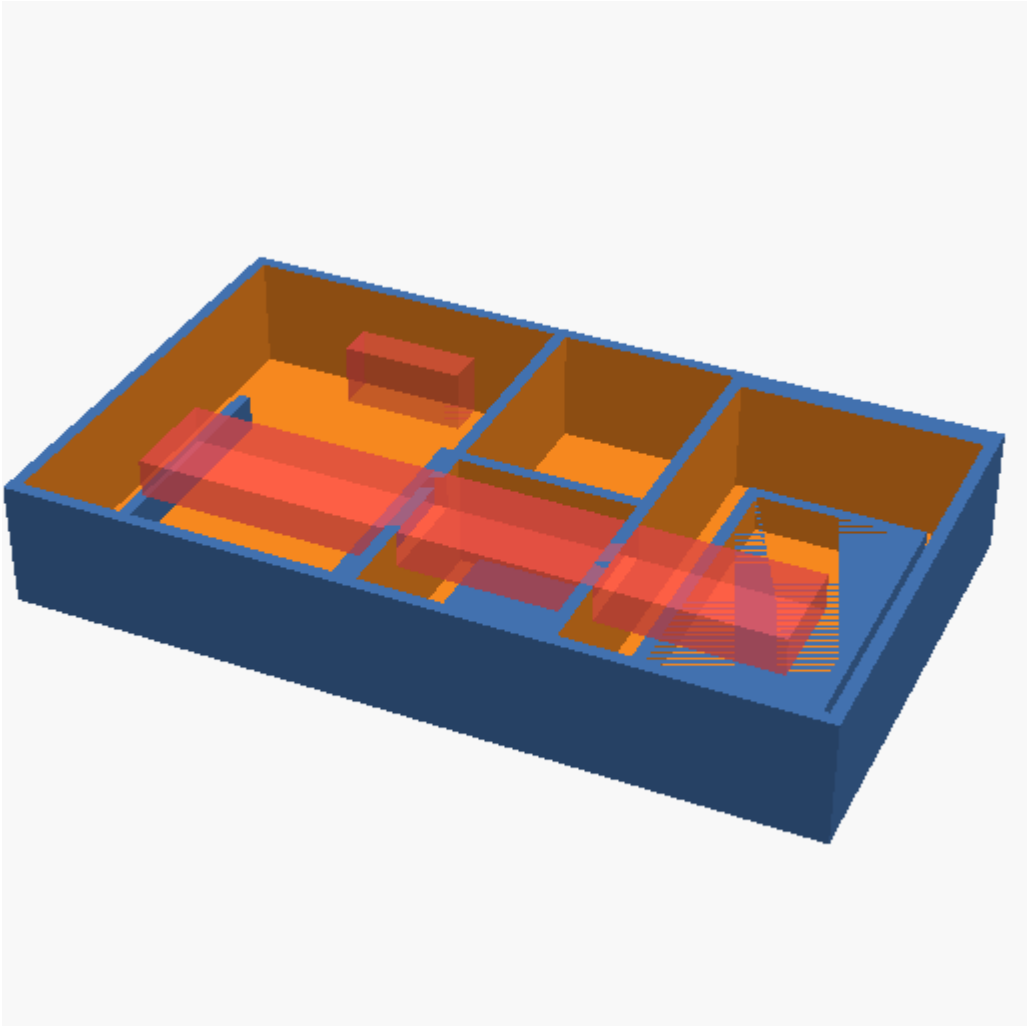

Figure 2. image

Listing 2. 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
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
            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);
        }
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
}
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();
```

2.3. Object - Hose_Adaptor

Not one of mine:
Created by Paul Tibble - 18/7/19
* https://www.thingiverse.com/Paul_Tibble/about

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

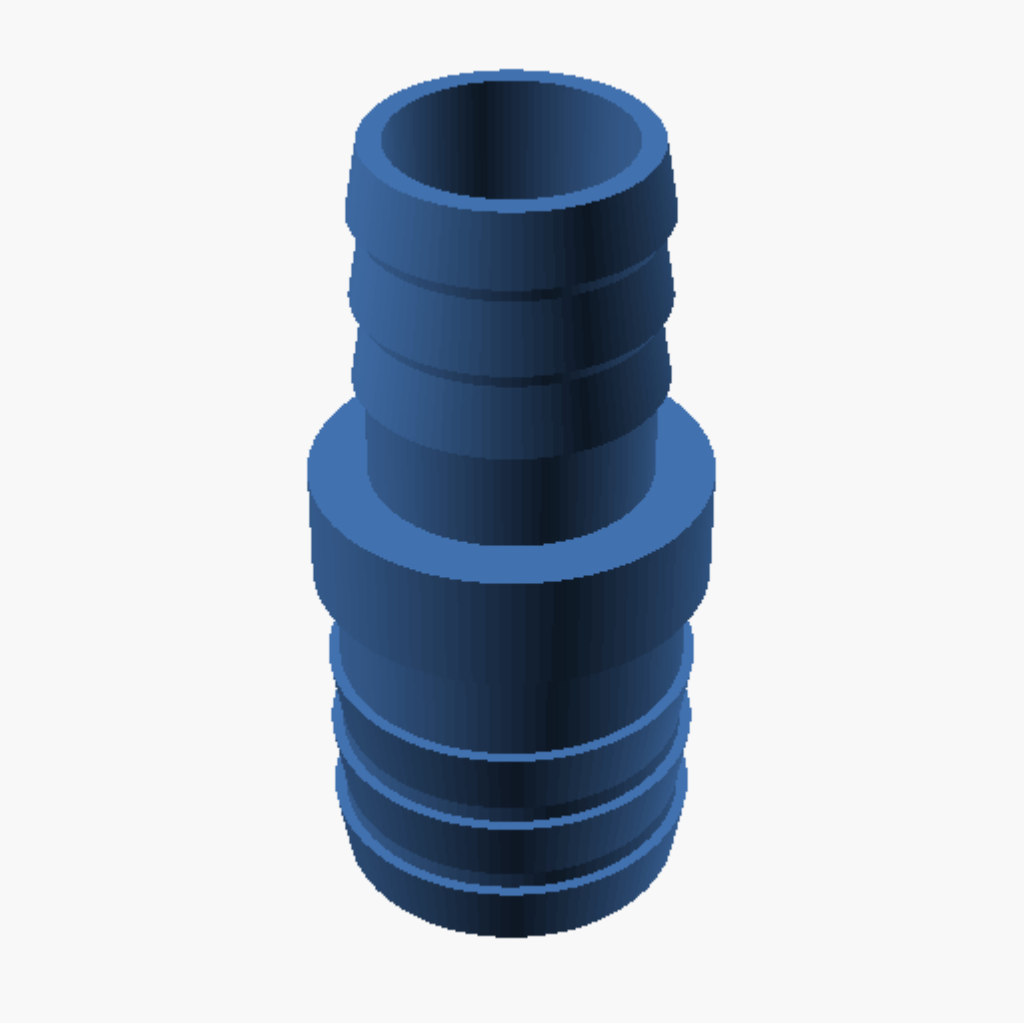

Figure 3. image

Listing 3. Openscad source

```
//////////////////////////////////////////////////////
// Created by Paul Tibble - 18/7/19                 //
// https://www.thingiverse.com/Paul_Tibble/about    //
// Please consider tipping, if you find this useful.  //
//////////////////////////////////////////////////////

$fn = 100*1;

// Outer Diameter (bottom)
outer_diameter_1 = 15;
// Wall Thickness (bottom)
wall_thickness_1 = 2;
// Rib Thickness (bottom), set to Zero to remove
barb_size_1 = 0.5;
// Length (bottom)
length_1 = 15;
// Outer Diameter (top), should be smaller than or equal to Outer Diameter
(bottom)
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
outer_diameter_2 = 12;
// Wall Thickness (top)
wall_thickness_2 = 1;
// Rib Thickness (top), set to Zero to remove
barb_size_2 = 0.5;
// Length (top)
length_2 = 15;
// Middle Diameter
mid_diameter = 17;
// Middle Length
mid_length = 5;


//do not change these
inner_diameter_1 = outer_diameter_1 - (wall_thickness_1*2);
inner_diameter_2 = outer_diameter_2 - (wall_thickness_2*2);

module create_profile() {
    ////////
    // Middle
    ///////

polygon(points=[[inner_diameter_1/2,length_1],[mid_diameter/2,length_1],[mid_dia
meter/2,length_1+mid_length],[inner_diameter_2/2,length_1+mid_length]]);
    //////
    //length 1
    /////
    translate([inner_diameter_1/2,0,0])square([wall_thickness_1,length_1]);
    //barb 1

translate([outer_diameter_1/2,0,0])polygon(points=[[0,0],[0,(length_1/5)],[barb_
size_1,(length_1/5)]]);
    //barb 2

translate([outer_diameter_1/2,length_1*0.25,0])polygon(points=[[0,0],[0,(length_
1/5)],[barb_size_1,(length_1/5)]]);
    //barb 3

translate([outer_diameter_1/2,length_1*0.5,0])polygon(points=[[0,0],[0,(length_1
/5)],[barb_size_1,(length_1/5)]]);
    //////
    //length 2
    /////

translate([inner_diameter_2/2,length_1+mid_length,0])square([wall_thickness_2,le
ngth_2]);
    //rib 1

translate([outer_diameter_2/2,(length_1+mid_length+length_2),0])polygon(points=[
[0,0],[0,-1*(length_2/5)],[barb_size_2,-1*(length_2/5)]]);
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
    //rib 2
    translate([outer_diameter_2/2,(length_1+mid_length+length_2)-
length_2*0.25,0])polygon(points=[[0,0],[0,-1*(length_2/5)],[barb_size_2,-
1*(length_2/5)]]);
    //rib 3
    translate([outer_diameter_2/2,(length_1+mid_length+length_2)-
length_2*0.5,0])polygon(points=[[0,0],[0,-1*(length_2/5)],[barb_size_2,-
1*(length_2/5)]]);
}

rotate_extrude(angle = 360, convexity = 10) create_profile();
//create_profile();
```

2.4. Object - KitchenDoorHoleStopper

Figure 4. image

Listing 4. Openscad source

```
//plug for door hinge mounting hole (WHITE)
// door replaced by sliding glass door 27/11/2021
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
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);
}
```

2.5. Object - Knurl

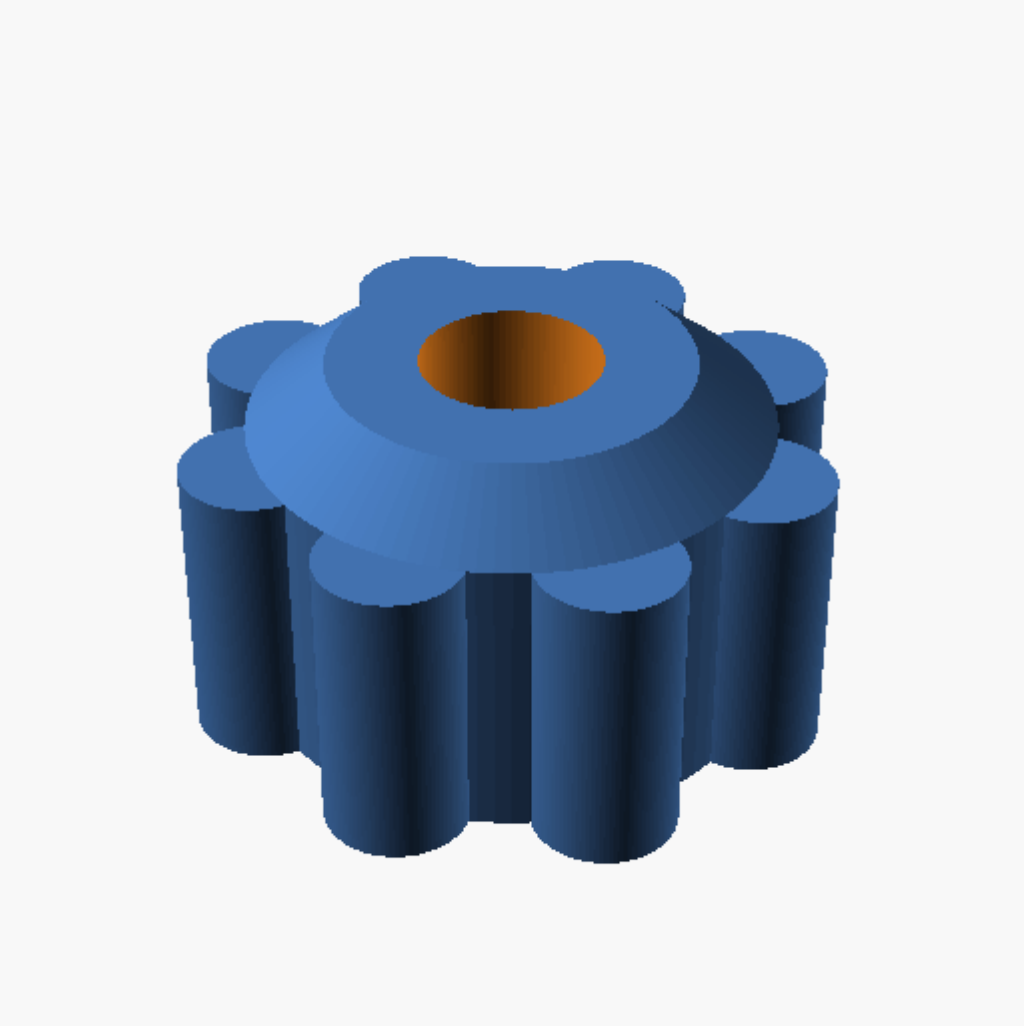

Figure 5. image

Listing 5. Openscad source

```
diam=9;
diamOut=14.4;
holeDiam=5;
height=8.2;
$fn=100;
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
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();
```

2.6. Object - Library-container

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

Figure 6. image

Listing 6. Openscad source

```
module nibLeft(x,y,z,nibR) {
    translate([x,y/10,0]) cylinder(h=z,r=nibR);
    translate([x,9*(y/10),0]) cylinder(h=z,r=nibR);
}
module nibRight(x,y,z,nibR) {
    translate([0,y/10,0]) cylinder(h=z,r=nibR);
    translate([0,9*(y/10),0]) cylinder(h=z,r=nibR);
}
module nibBottom(x,y,z,nibR) {
    translate([x/10,0,0]) cylinder(h=z,r=nibR);
    translate([9*(x/10),0,0]) cylinder(h=z,r=nibR);
}
module nibTop(x,y,z,nibR) {
    translate([x/10,y,0]) cylinder(h=z,r=nibR);
    translate([9*(x/10),y,0]) cylinder(h=z,r=nibR);
}
module containerOpenLid(x,y,z,rimThick,bottomThick,nibYN,nibR) {
rimR=rimThick/2;
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
//all 8 corners defined first
//corners should be AROUND the contained cube defined by x y z
corner000=[0,0,0];
corner0=[-rimR,-rimR,-(rimR+bottomThick)];
corner0x=[x+rimR,-rimR,-(rimR+bottomThick)];
corner0y=[-rimR,y+rimR,-(rimR+bottomThick)];
corner0xy=[x+rimR,y+rimR,-(rimR+bottomThick)];
corner0z=[-rimR,-rimR,z];
corner0xz=[x+rimR,-rimR,z];
corner0yz=[-rimR,y+rimR,z];
corner0xyz=[x+rimR,y+rimR,z];   //draw the debug contents
translate([0,0,-bottomThick]) cube([x,y,bottomThick]);
module corner0() {
    translate(corner0) sphere(r=rimR);
}
module corner0x() {
    translate(corner0x) sphere(r=rimR);
}
module corner0y() {
    translate(corner0y) sphere(r=rimR);
}
module corner0xy() {
    translate(corner0xy) sphere(r=rimR);
}
module corner0z() {
    translate(corner0z) sphere(r=rimR);
}
module corner0xz() {
    translate(corner0xz) sphere(r=rimR);
}
module corner0yz() {
    translate(corner0yz) sphere(r=rimR);
}
module corner0xyz() {
    translate(corner0xyz) sphere(r=rimR);
}
if (nibYN=="nibY") {
    nibBottom(x,y,z,nibR);
    nibLeft(x,y,z,nibR);
    nibRight(x,y,z,nibR);
    nibTop(x,y,z,nibR);
}
union(){
    //floor
    hull(){
        corner0();
        corner0x();
        corner0y();
        corner0xy();
    }
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
        //left
        hull(){
            corner0();
            corner0z();
            corner0y();
            corner0yz();
        }
        //right
        hull(){
            corner0x();
            corner0xy();
            corner0xyz();
            corner0xz();
        }
        //top
        hull(){
            corner0y();
            corner0yz();
            corner0xyz();
            corner0xy();
        }
        //bottom
        hull(){
            corner0();
            corner0z();
            corner0x();
            corner0xz();
        }
    }
}

// example
//caseRim=1.5;
//$fn=100;
//odH=10;
//odW=156;
//odD=73;
//odJSH=6;
//#containerOpenLid(odW,odD,odH,caseRim,odJSH-caseRim,"nibY",.6);
//cube([odW,odD,odH]);

module containerVertSlot(x,y,z,rimThick,bottomThick,nibYN,nibR) {
    rimR=rimThick/2;
    //all 8 corners defined first
    //corners should be AROUND the contained cube defined by x y z
    corner000=[0,0,0];
    corner0=[-rimR,-rimR,-(rimR+bottomThick)];
    corner0x=[x+rimR,-rimR,-(rimR+bottomThick)];
    corner0y=[-rimR,y+rimR,-(rimR+bottomThick)];
    corner0xy=[x+rimR,y+rimR,-(rimR+bottomThick)];
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
corner0z=[-rimR,-rimR,z];
corner0xz=[x+rimR,-rimR,z];
corner0yz=[-rimR,y+rimR,z];
corner0xyz=[x+rimR,y+rimR,z];   //draw the debug contents
translate([0,0,-bottomThick]) cube([x,y,bottomThick]);
module corner0() {
    translate(corner0) sphere(r=rimR);
}
module corner0x() {
    translate(corner0x) sphere(r=rimR);
}
module corner0y() {
    translate(corner0y) sphere(r=rimR);
}
module corner0xy() {
    translate(corner0xy) sphere(r=rimR);
}
module corner0z() {
    translate(corner0z) sphere(r=rimR);
}
module corner0xz() {
    translate(corner0xz) sphere(r=rimR);
}
module corner0yz() {
    translate(corner0yz) sphere(r=rimR);
}
module corner0xyz() {
    translate(corner0xyz) sphere(r=rimR);
}
if (nibYN=="nibY") {
    nibLeft(x,y,z,nibR);
    nibRight(x,y,z,nibR);
}
union(){
    //floor
    hull(){
        corner0();
        corner0x();
        corner0y();
        corner0xy();
    }
    //left
    hull(){
        corner0();
        corner0z();
        corner0y();
        corner0yz();
    }
    //right
    hull(){
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
            corner0x();
            corner0xy();
            corner0xyz();
            corner0xz();
        }
    }
}
// example
//caseRim=1.5;
//$fn=100;
//odH=10;
//odW=15;
//odD=73;
//odJSH=6;
//containerVertSlot(odW,odD,odH,caseRim,odJSH-caseRim,"nibY",.6);
//cube([odW,odD,odH]);

if (library) {} else  {
    echo("trying to compile a library!");
    linear_extrude(height = 4) {
        text("trying to compile a library!");
        }
    }
```

2.7. 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.

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

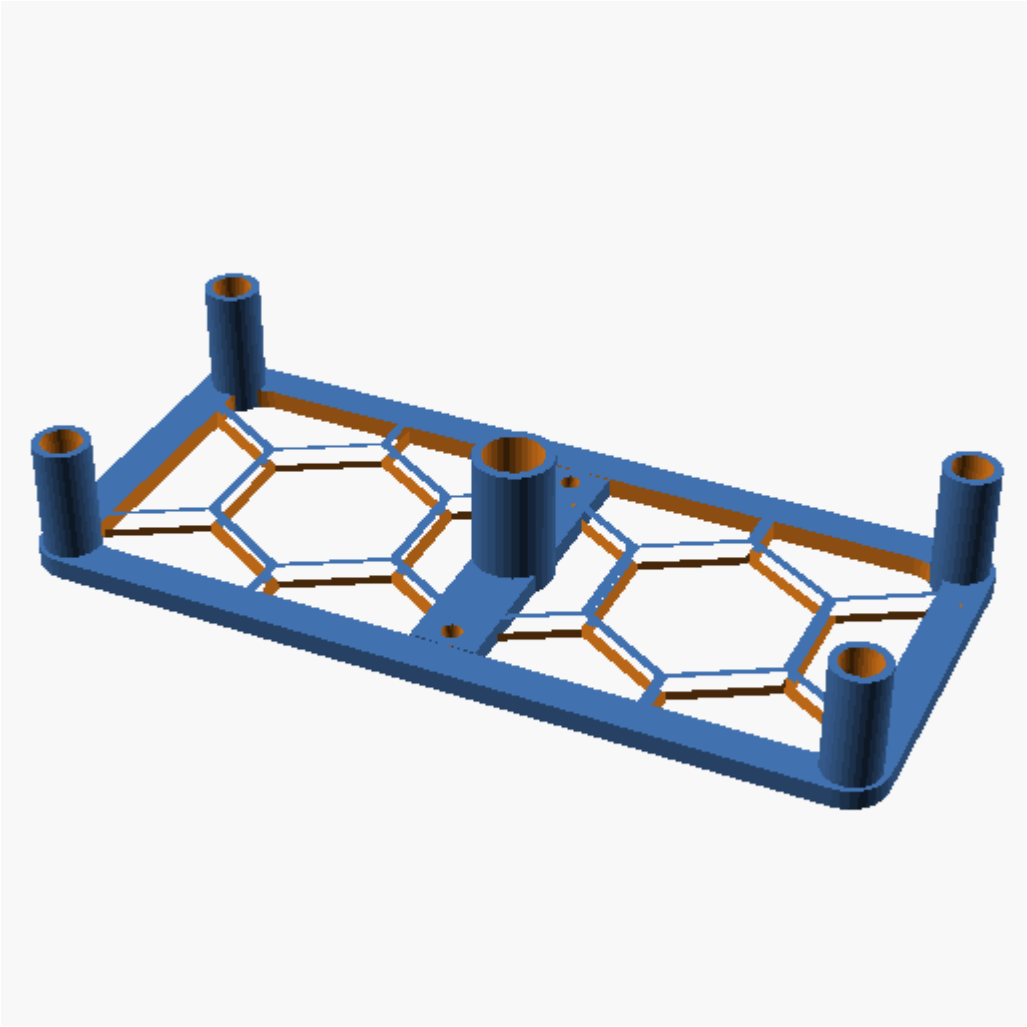

Figure 7. image

Listing 7. 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;
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
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);
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
        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);
}
```

2.8. Object - Test

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

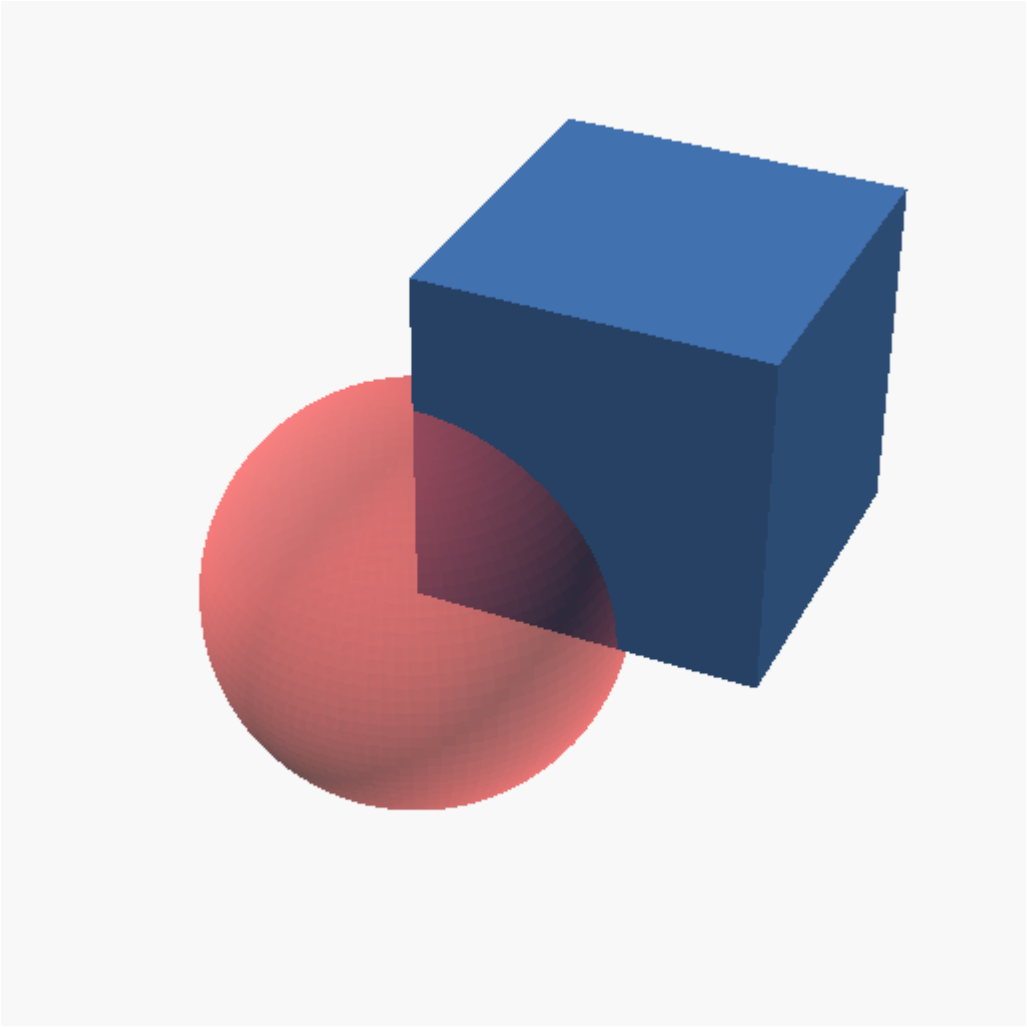

Figure 8. image

Listing 8. Openscad source

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

2.9. Object - VacuumPreAmplifierBase

vacuumpreamp

housing a retro vacuum tube china preamp.

Have a nice wooden box that is looking for some use as a housing The pre-amp is a cheap vaccum tube type sourced from aliexpress https://a.aliexpress.com/_B0MVMZ

I've created an openscad model of the Box based on some measurments with a calliper. The model is designed to help asses where to drill holes and to print a guide to drill the holes. The preamp has a power input (12v~) an in and an output (headphone jack) and a Volume

potentiometer. Also the housing is to expose the Vacuum Tubes to the interested viewer. Since the lid is hinged and the relative position of the tubes to the lid, when opening, is difficult to eyeball the model was created to try out different Hole placments as well as providing a template for Drill guides.

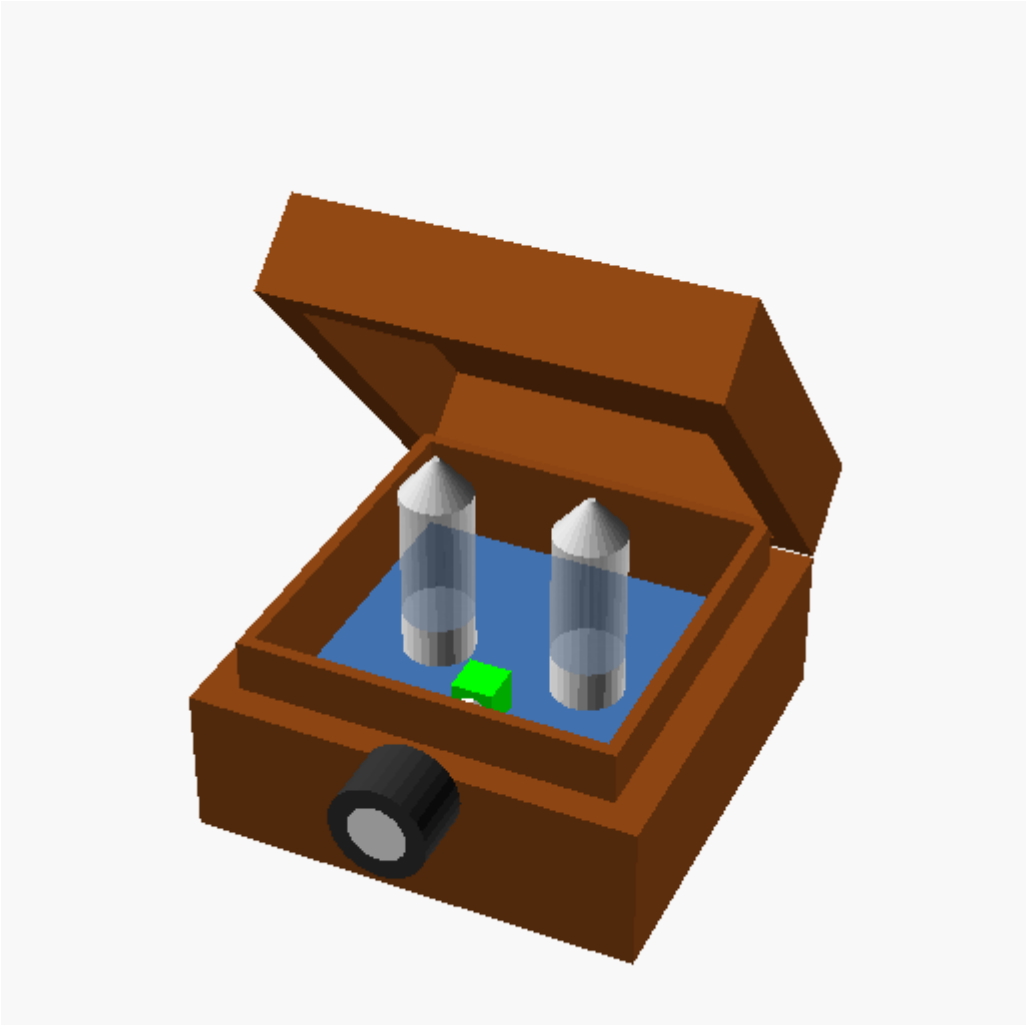

Figure 9. image

Listing 9. Openscad source

```
box1X=105.5;
box1Y=106;
box1Z=36;
box1BaseH=3;

box2X=89;
box2Y=91;
box2Z=45.5;

box3X=83;
box3Y=85;
box3Z=50;
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
lidX=box1X;
lidY=box1Y;
lidZ=23;
lidDepth=20.3;
lidStampR=20;
lidHingeAngle=50;
lidAnimZ=0;

preampBoardX=77;
preampBoardY=66;
preampBoardZ=1.5;
preampTubeR=17/2;
preampTubeH=42;
preampTubeBaseH=10;
preampTubeTipH=51;
preampTubeC=[200/255,200/255,200/255];
preampKnobR=11.5;
preampKnobH=16;
preampAxleH=29;

brown=[139/255,69/255,19/255];
gold=[255/255,215/255,0/255];
Blue=[0/255,0/255,200/255];

module box(){
    color(brown)
    difference(){
        union(){
            cube([box1X,box1Y,box1Z]);
            translate([box1X/2-box2X/2,box1Y/2-box2Y/2,box1BaseH])
cube([box2X,box2Y,box2Z]);
        }
        translate([box1X/2-box3X/2,box1Y/2-box3Y/2,3])
cube([box3X,box3Y,box3Z]);
        // star the next line to see inside the box
        *translate([-.5,-.5,-.5]) cube([box1X+1,box1Y*.85+1,box1Z/2+1]);
    }
}

//lid
module lid(){
    color(gold) translate([(box1X/2),(box1Y/2),lidZ+.0001])
cylinder(h=1,r1=lidStampR,r2=lidStampR);
    color(brown) translate([0,0,0])
        difference(){
            translate([0,0,.001]) cube([lidX,lidY,lidZ]);
            translate([box1X/2-box2X/2,box1Y/2-box2Y/2,0])
cube([box2X,box2Y,lidDepth]);
    }
}
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
module tube () {
    union(){
        color(preampTubeC,.5) translate([0,0,preampTubeBaseH])cylinder(h=42-
preampTubeBaseH,r1=preampTubeR,r2=preampTubeR);
        color([1,1,1])cylinder(h=preampTubeBaseH,r=preampTubeR);
        translate([0,0,preampTubeH]) color(preampTubeC)
cylinder(h=preampTubeTipH-preampTubeH,r1=preampTubeR,r2=1);
    }
}

translate([(box1X-box3X)/2,((box1Y-box3Y)/2)+(box3Y-preampBoardY)-
1,box1BaseH+21])
union() {
    //board
    cube([preampBoardX,preampBoardY,preampBoardZ]);
    //tubes
    translate([15+preampTubeR,15+preampTubeR,preampBoardZ]) tube();
    translate([52+preampTubeR,15+preampTubeR,preampBoardZ]) tube();
    //Volume Knob Base
    translate([38,0,preampBoardZ]) color([0,1,0]) cube([10,10,10]);
    //volume knob
    translate([43,-(preampKnobH+preampAxleH),preampBoardZ+5]) rotate([270,0,0])
    union(){
        difference() {
            color([50/255,50/255,50/255]) cylinder(h=preampKnobH,r=preampKnobR);
            translate([0,0,-.001]) cylinder(h=1,r=(preampKnobR/100)*60);
        }
        color([255/255,255/255,255/255])cylinder(h=1,r=(preampKnobR/100)*60);
        //knob axle
        translate([0,0,preampKnobH]) color([1,1,1])cylinder(h=preampAxleH,r=3);
    }
}
//draft base
translate([15,box1Y-((box1Y-box3Y)/2)-1,box1BaseH])
color([0,0,0])cube([8,1,21]);
//enclosure
box();
translate([box1X,box1Y,(box1Z)+lidAnimZ+.5]) rotate([lidHingeAngle,0,180])
lid();
```

2.10. Object - buttonBack

Button backing for the monitor.

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

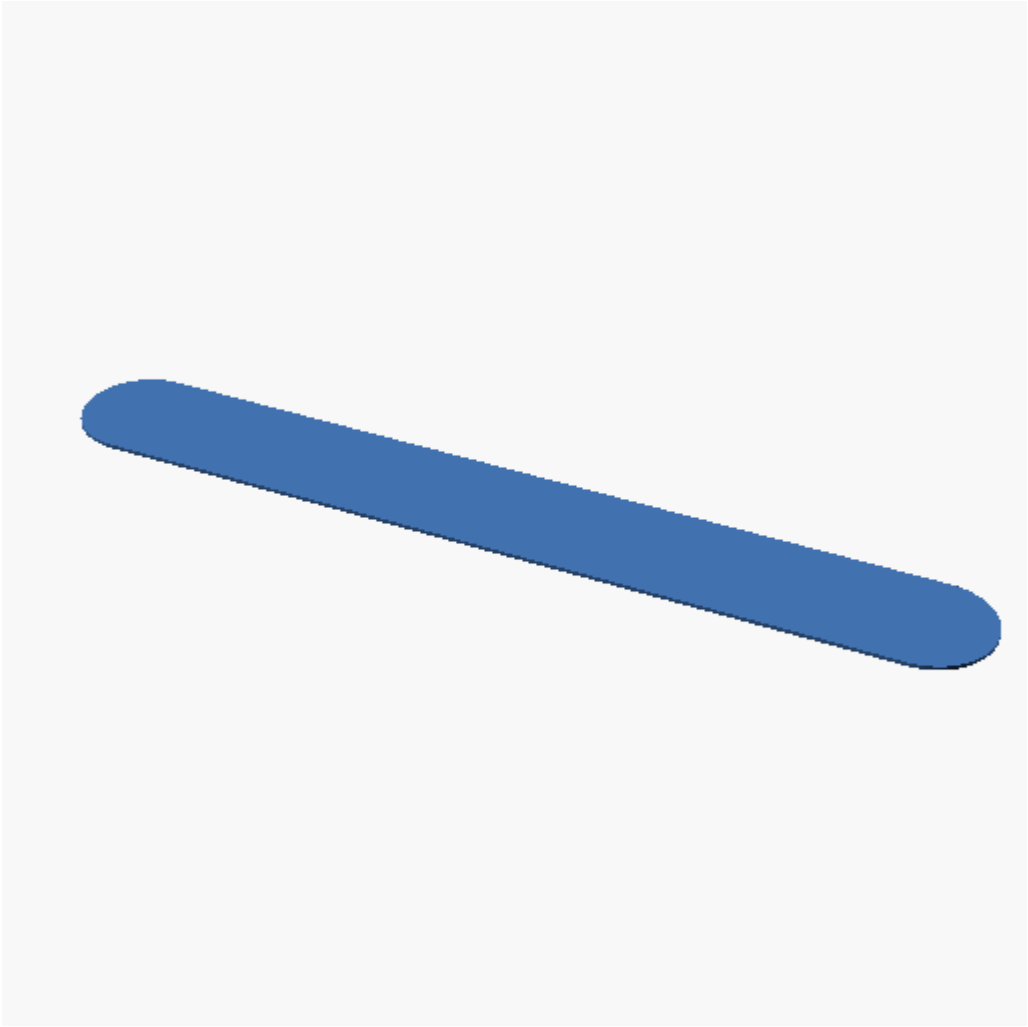

Figure 10. image

Listing 10. 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);
}
```

2.11. 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.

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

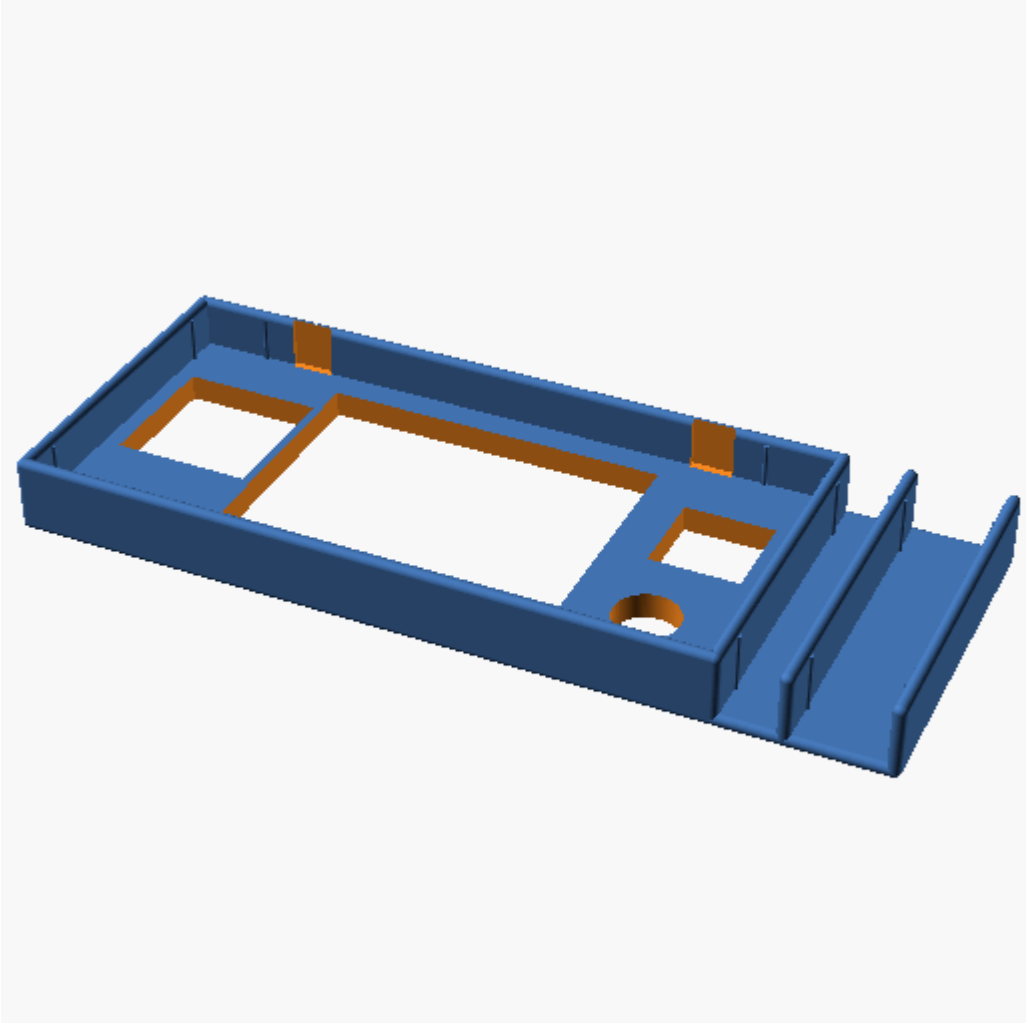

Figure 11. image

Listing 11. 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;
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
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]);
    }
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
}
//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);
```

2.12. 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.

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

Figure 12. image

Listing 12. 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-
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
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);
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
    //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();
```

2.13. Object - fridgeDoorInterimHandle

Figure 13. image

Listing 13. Openscad source

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

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
        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);
}
```

2.14. 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.

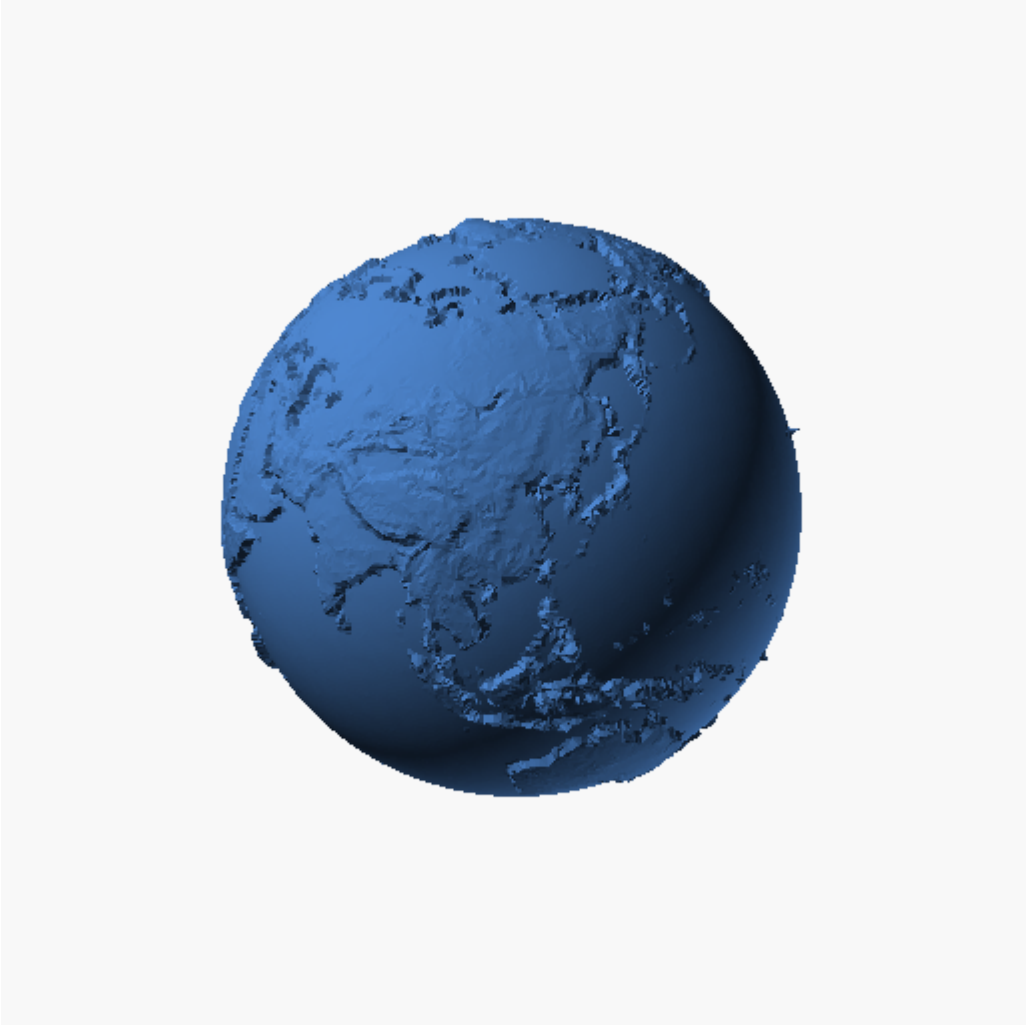

Figure 14. image

Listing 14. Openscad source

```
// Geody Planet 1 - SCAD
// Geody - https://www.geody.com/
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
// 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);}
```

2.15. Object - ikeabung

This was a replacement foot for an IKEA shelf.

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

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

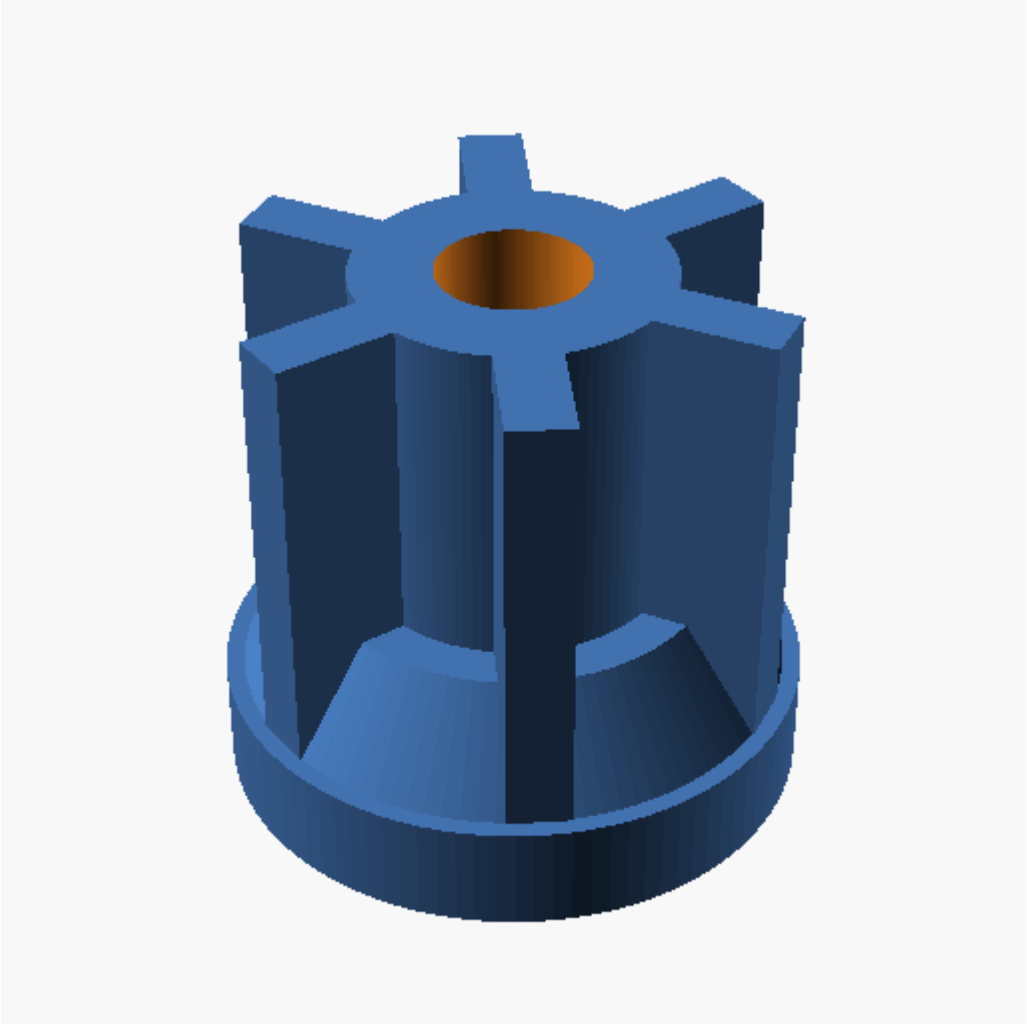

Figure 15. image

Listing 15. 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);
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

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

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

2.16. Object - internal-volume

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

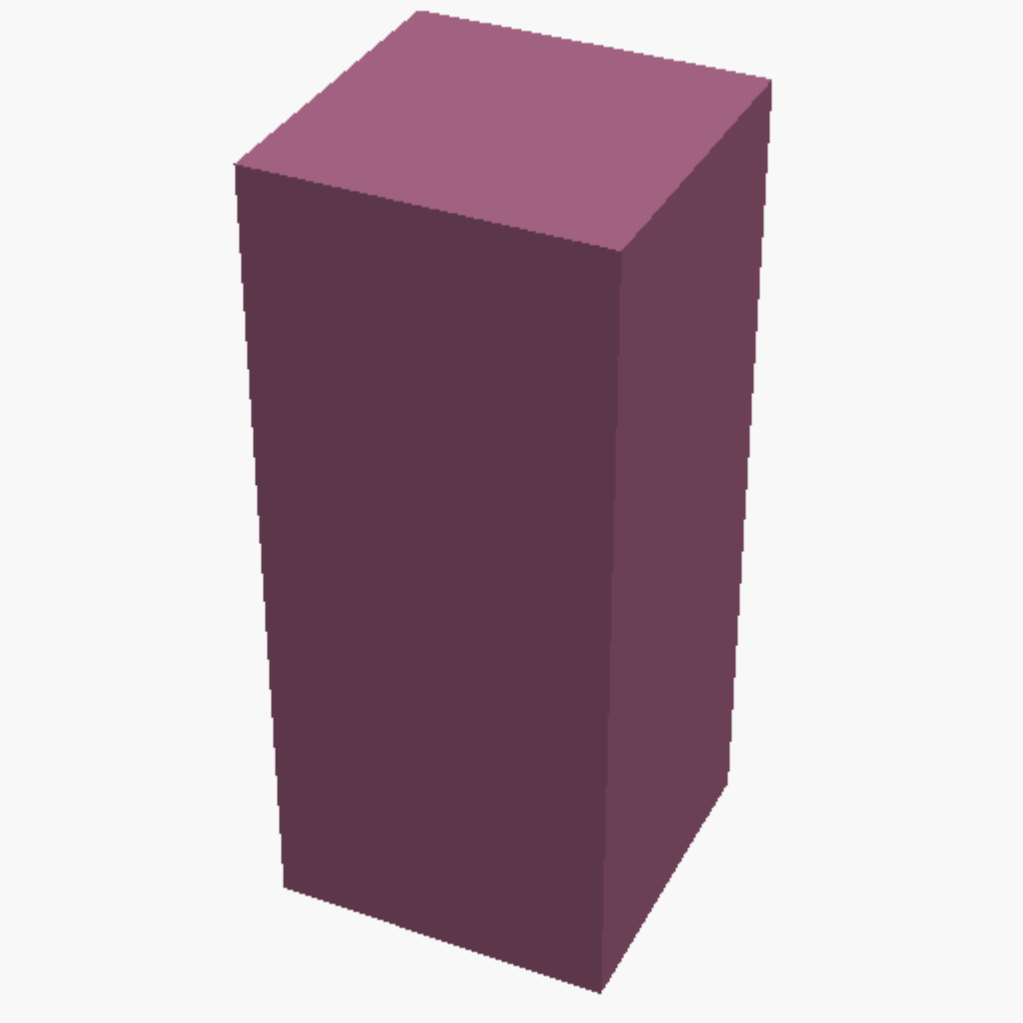

Figure 16. image

Listing 16. Openscad source

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

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

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

2.17. Object - lampRing

Figure 17. image

Listing 17. Openscad source

```
//for LED lamps in ceiling in Howth
// the originals are wider and therefore the new ones need a spacer to cover the
hole
//colour is white
//led lamps are 105mm Diameter (4 lamps)

lampD=105;
lampH=2;
holeD=99;
coverD=125;
coverInD=99;
coverH=2;
coverRidgeW=5;
$fn=100;

//lamp
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
*color("white")
    translate([0,0,coverH])
        cylinder(h=2,d=lampD);

color("white") union(){
    difference(){
        cylinder(h=coverH,d=coverD);
        translate([0,0,-.1]) cylinder(h=coverH+.2,d=coverInD);
    }
    translate([0,0,coverH])
    difference(){
        cylinder(h=lampH,d=lampD+coverRidgeW);
        translate([0,0,-.1]) cylinder(h=lampH+.2,d=lampD+1);
    }
}
```

2.18. Object - midletoninset

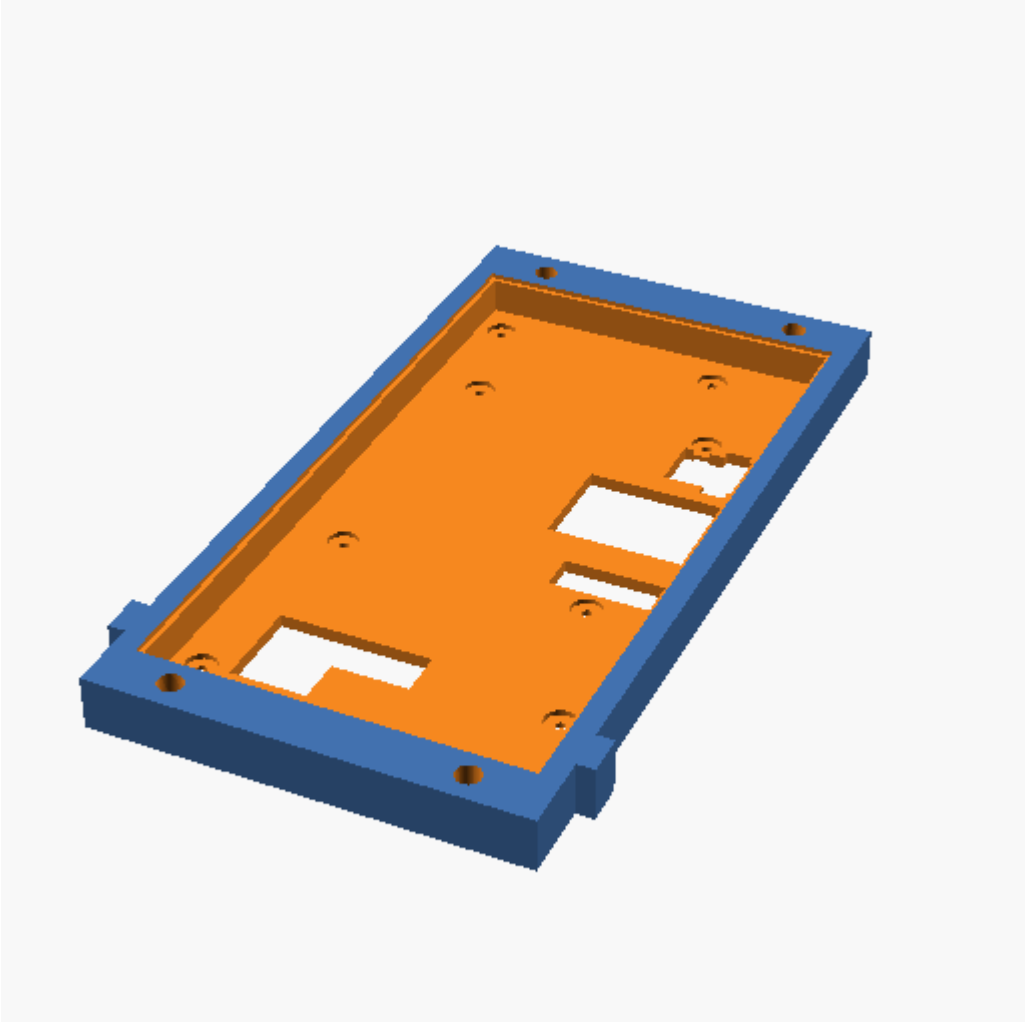

Figure 18. image

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

Listing 18. 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 throrugh
        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])
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
            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
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
//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);
```

2.19. Object - mountingplateaircraftmotor

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

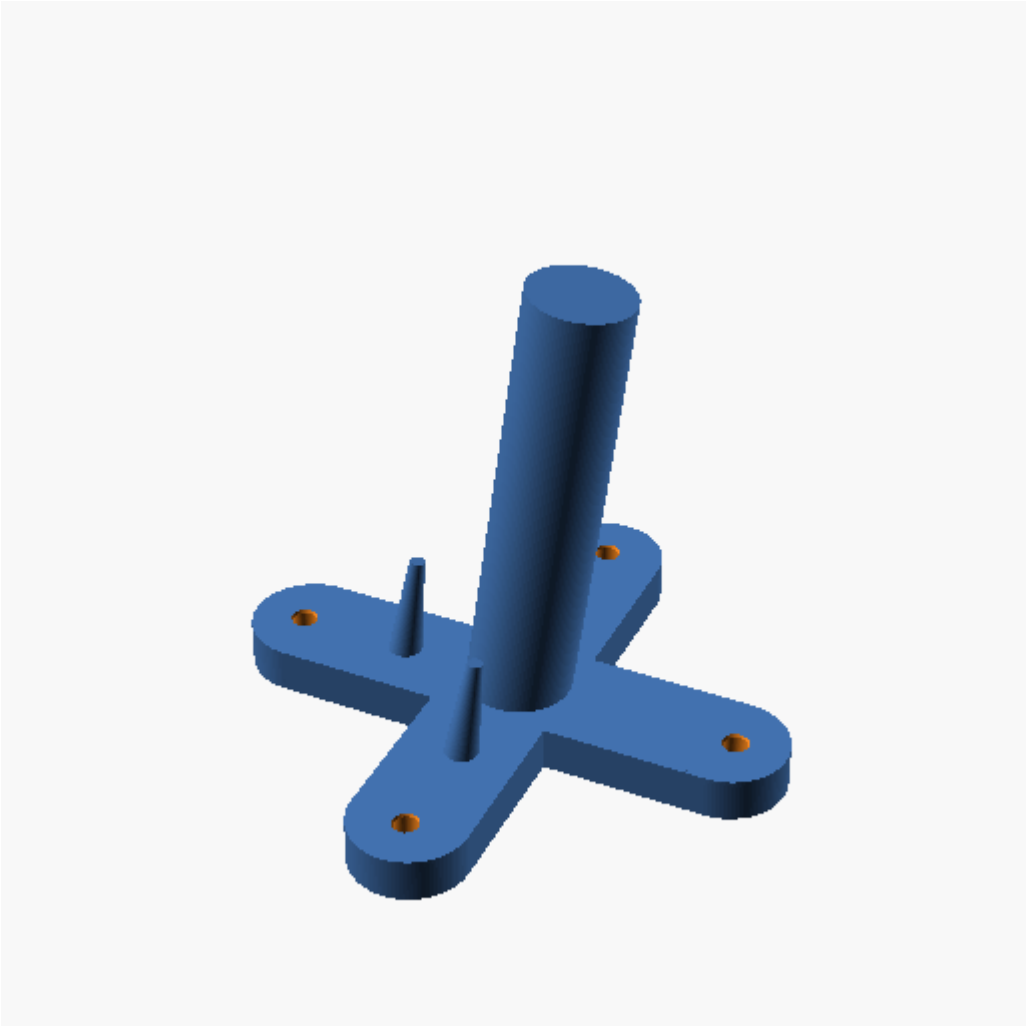

Figure 19. image

Listing 19. Openscad source

```
//second attempt with rotated arms so as to save on cutting out.
//set number of faces higher so that the cylinder doesn't lok like a pentagon
$fn=100;
//global vars
//text font Arial and so far size under 1.8 didn't show in print.....
font = "Arial";
letter_size = 2.5;
letter_height = 1.5;
line1="Ser#1";
line2="VSR";
line3="V6";
//pin depth into airframe of the wooden original was 30 test prints 20 is enough
//also of note is that I had to rotate the blasted pin by 45 and 3° as I drilled
the hole wrong. guess might be 5 or 6 (angle2) and one or two (angle1) to the
side
pinheight=30;
pinangle1=2;
pinangle2=7;
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
//radius of the mounting holes
screwhole=.81;
holeOffset=15.5;
//radius of screw hole opening
flare=1;

module letter(l) {
    // Use linear_extrude() to make the letters 3D objects as they
    // are only 2D shapes when only using text()
    linear_extrude(height = letter_height) {
        text(l, size = letter_size, font = font);
    }
}

//this is the 4 armed mount with drill holes
module mount()
    {
    union(){
        for ( arm = [0:90:360]){
            rotate([0,0,arm])
                //arm with drill hole
                difference(){
                    union(){
                        //arm
                        translate ([0,-4,0]) cube([15,8,3] );
                        //rounded tip
                        translate ([15,0,0]) cylinder(h=3, r1=4, r2=4);
                        }
                    //  subtract drill hole plus additional
                    translate([holeOffset,0,0]) cylinder(h=3, r1=screwhole,
r2=screwhole);
                    //flaring
                    translate([holeOffset,0,2.8]) cylinder(h=.5, r1=screwhole,
r2=flare);
                    translate([holeOffset,0,0]) cylinder(h=.25, r1=flare,
r2=screwhole);
                    //central mounting hole for spindle
                    cylinder(h=1.5, r1=2.5, r2=2.1);
                }
            }
        }
    }


module mountpluspin()
{
//the mount and the pin for insertion into the aircraft body
    union(){
        mount();
        //central mounting rod should be 30 long as measured but shorter    for
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
test prints
        //aslo of note is that I had to rotate the blasted pin by 45 and 3° as I
drilled the hole wrong. guess is 3° might be 5 or 6
        rotate ([pinangle1,pinangle2,45]) translate ([0,0,2])
cylinder(h=pinheight, r1=3.5, r2=3.5);
        *translate ([4,1,2.2])  letter (line1);
        *rotate(90,90,90) translate ([4,-1,2.2])  letter (line2);
        *translate ([4,-3,2.2])  letter (line3);
        }
}
//add some antispin pegs.
union (){
    mountpluspin();
    translate ([0,-8,2]) rotate ([pinangle1,pinangle2,45]) cylinder(h=8, r1=1.2,
r2=.5);
    translate ([-8,0,2]) rotate ([pinangle1,pinangle2,45]) cylinder(h=8, r1=1.2,
r2=.5);
}
```

2.20. Object - schuko

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

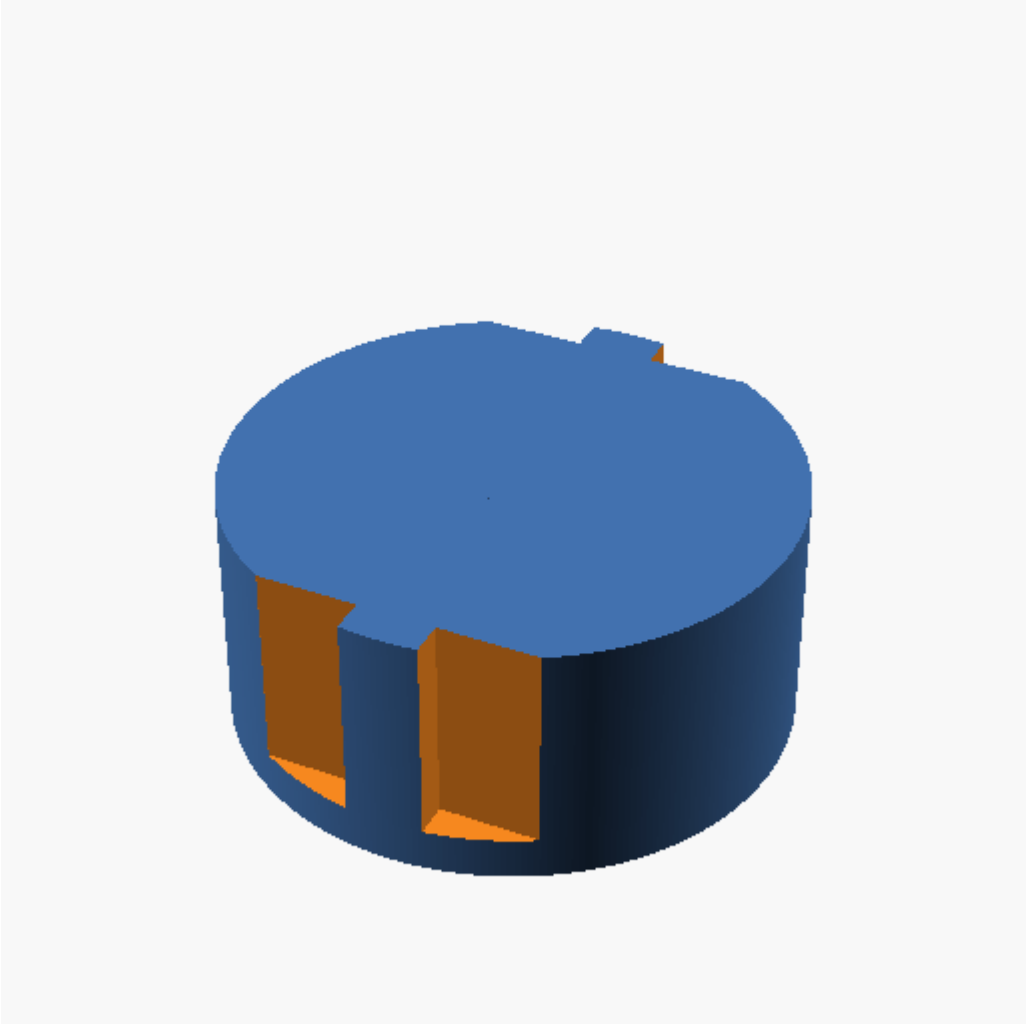

Figure 20. image

Listing 20. 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]
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
// 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);
            }
        }
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
        // 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);
    }
}
}

}
```

2.21. Object - screen_mounting_tabs

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

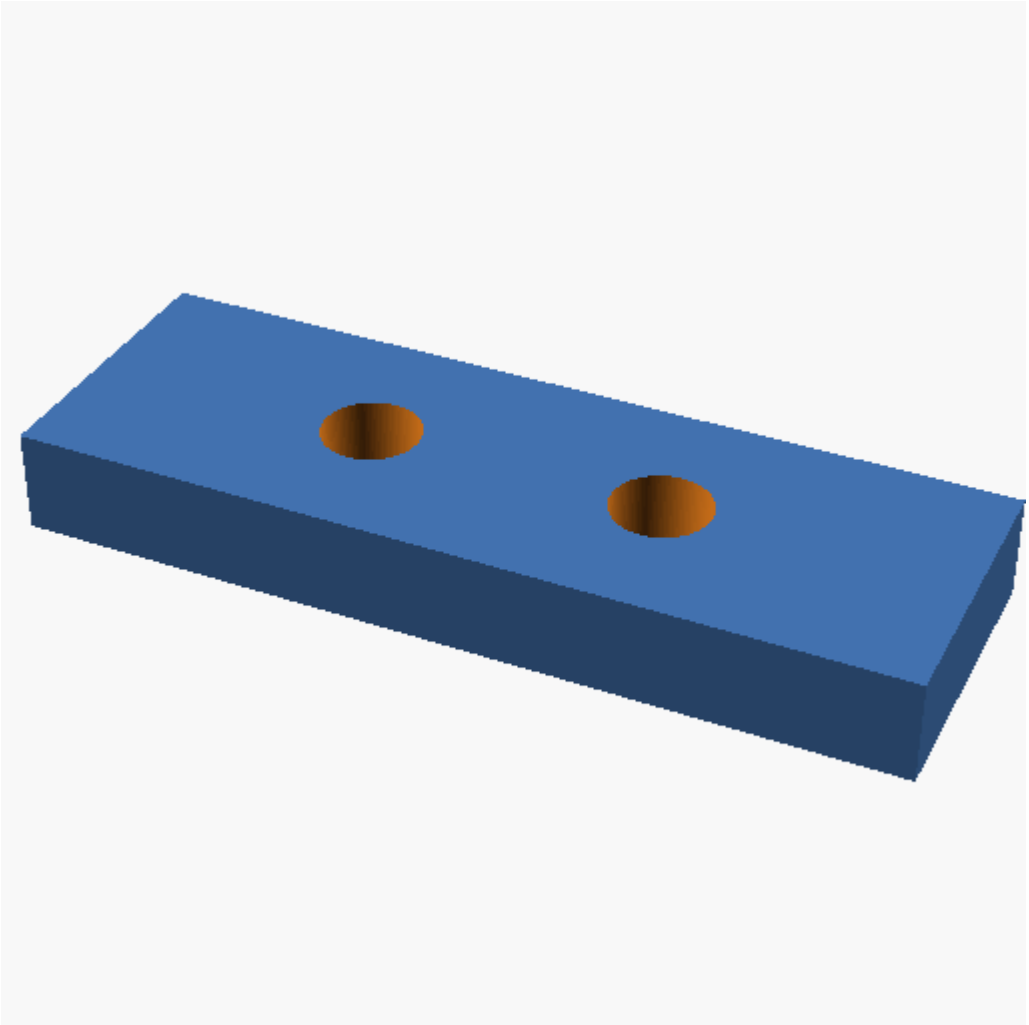

Figure 21. image

Listing 21. 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])
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
            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);
```

2.22. Object - spool

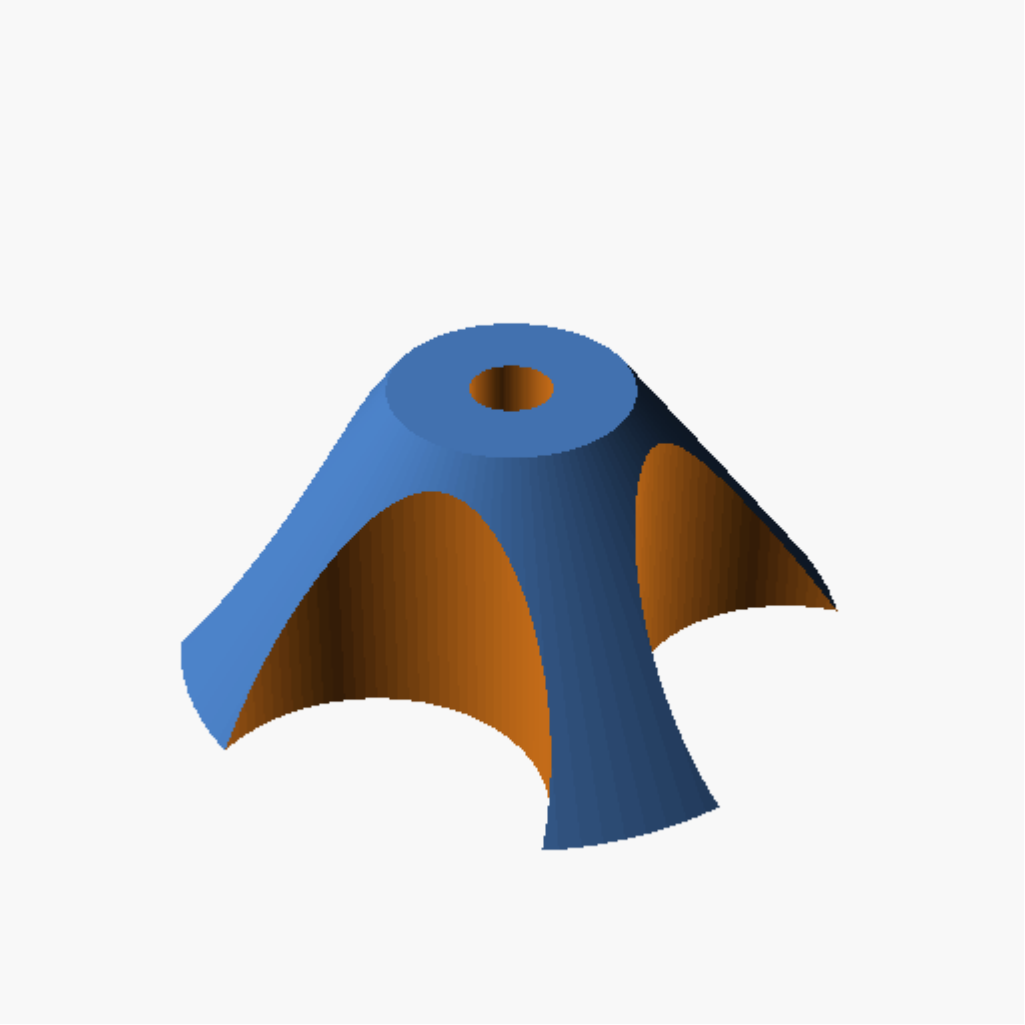

Figure 22. image

Listing 22. Openscad source

```
coneH=30;     //height of the cone
coneDin=25;   //smallest diameter of the cone
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
coneDout=70;  //widest diameter of cone
axleD=8;      //axle diameter of the axle for the 608 bearing - we;ll add for
printer tolerance
$fn=100;      //make things round
bearingH=7;   //608 skateboard bearing height
bearingD=22;  //608 skateboard bearing diameter we'll add amillimeter or two
later to account for the fitting ring
fittingD=bearingD+7;  //outer diameter of the fitting ring for the bearing
nubAngle=360/8;  //the fitting nubs for the bearing at x degree rotation
printerRadTol=.2; //add this value to the radius
nubRad=.5;    //the nub radius for the bearing fitting ring

module cone(height,inD,outD) {
    cylinder(h=coneH , r2=(inD/2) , r1=(outD/2) );
    }

module axle(height,diameter,tol) {
    translate([0,0,-.1]) cylinder(h=height,r=(diameter/2)+tol); //axle
}
module bearing(height,diameter,tol) {
    translate([0,0,-.1]) cylinder(h=height+.1,r=(diameter/2)+tol); //bearing
}

//subtract for quicker print
module removeCyls(bearingD,coneDout,coneH){
    translate([-((bearingD/2)+(coneDout/4)+4),0,-.1])
cylinder(h=coneH,r=coneDout/4);
    translate([+((bearingD/2)+(coneDout/4)+4),0,-.1])
cylinder(h=coneH,r=coneDout/4);
    translate([0,+((bearingD/2)+(coneDout/4)+4),-.1])
cylinder(h=coneH,r=coneDout/4);
    translate([0,-((bearingD/2)+(coneDout/4)+4),-.1])
cylinder(h=coneH,r=coneDout/4);
    }

module ring(inRad,outRad,height,tol) {
  difference(){
      cylinder(h=height,r=outRad+tol);
      translate([0,0,-.1]) cylinder(h=height+.2,r=inRad+tol);
    }
}

module fittingNubsCircle(nubRad,height,inRad,angle,tol) {
    rad=inRad+nubRad+tol;
    for (pos=[0:angle:360]) {
        *echo(pos);
        rotate ([0,0,pos]) translate([rad,0,0]) cylinder(h=height,r=nubRad);
    }
}
//
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
difference(){
    union(){
        difference(){
            cone(coneH,coneDin,coneDout);
            translate([0,0,-.1]) bearing(bearingH+.1,fittingD,printerRadTol);
            translate([0,0,-.1]) axle(coneH+.5,axleD,printerRadTol);
        }//
        ring( (bearingD/2)+nubRad, (fittingD/2) , bearingH , printerRadTol );
        fittingNubsCircle( nubRad , bearingH , bearingD/2 , nubAngle ,
printerRadTol );
    }//
    removeCyls(bearingD,coneDout,coneH);
}
```

2.23. Object - steeringaxle

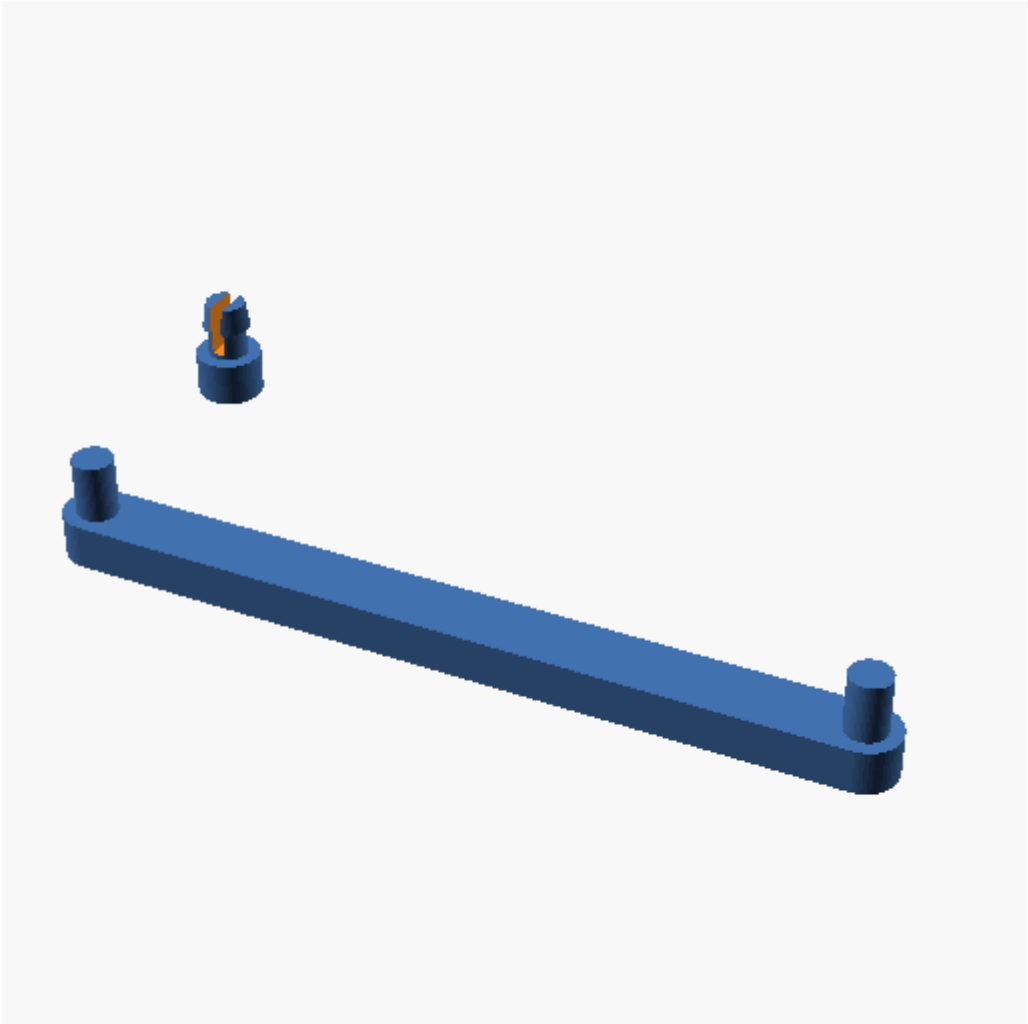

Figure 23. image

Listing 23. Openscad source

```
$fn=50;
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
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();
```

2.24. Object - strikeplate

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

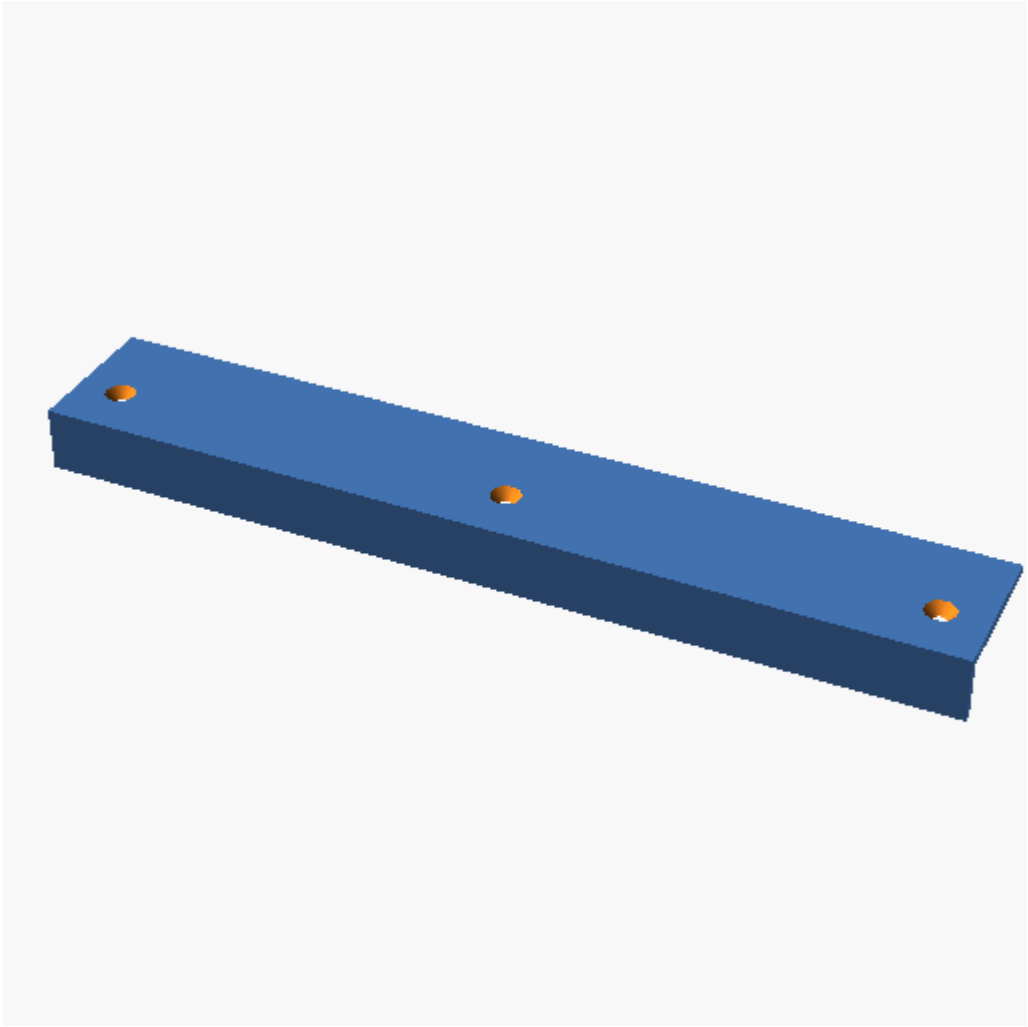

Figure 24. image

Listing 24. 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();
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

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

2.25. Object - tesa

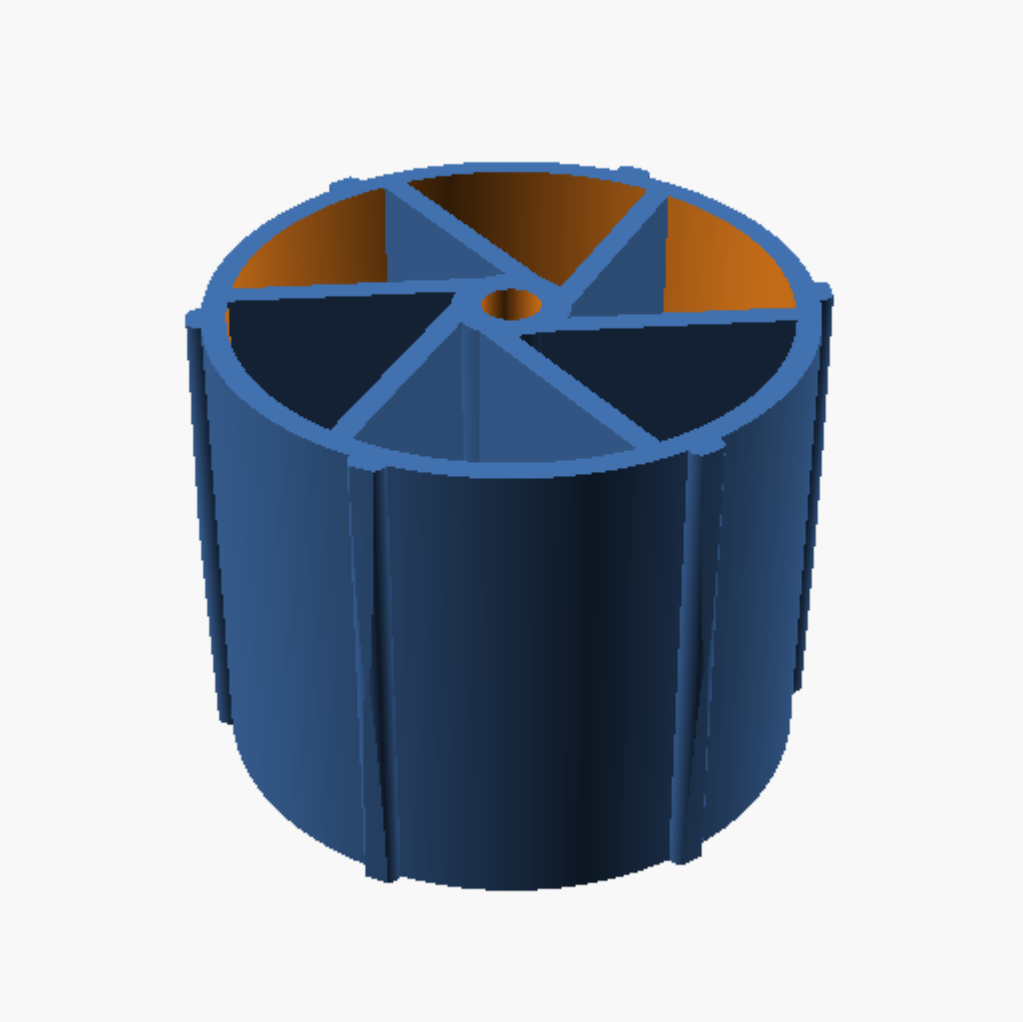

Figure 25. image

Listing 25. Openscad source

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

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

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
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();
}
```

2.26. Object - vsagcrd

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

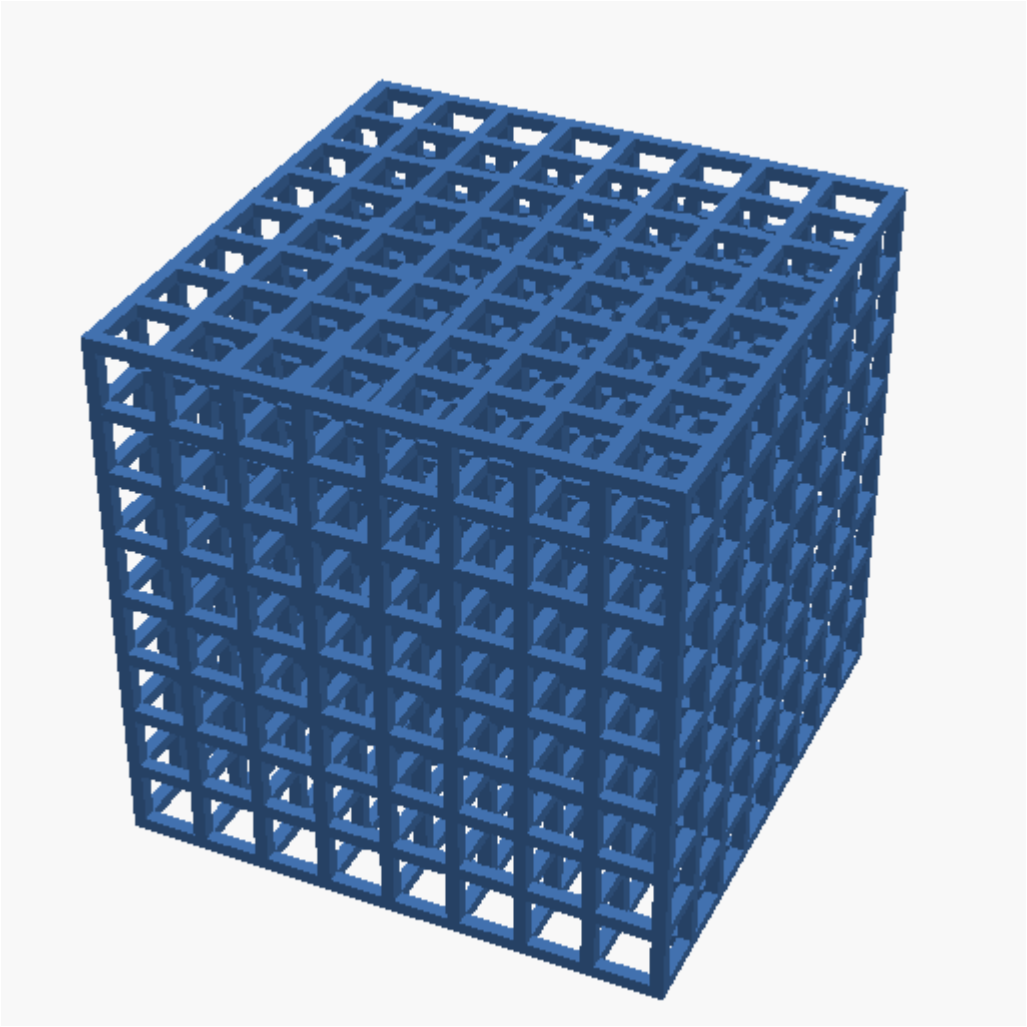

Figure 26. image

Listing 26. Openscad source

```
//Script to create a wire mesh cube with 8x8x8 empty spaces
//Virtual Space and
//Global communications research department
//logo base object with 8.2 cm side length
//consisting of the multiplied basic primitives of
//an x,y,and z axis beam iterate in one dimension in loops
//also includes the inner primitive (in 3 flavours)
$fn=100;
module x_beam(){ //just a cube with parameters in one place
    cube([82,2,2]);
}
module y_beam(){ //just a cube with parameters in one place
    cube([2,82,2]);
}
module z_beam(){ //just a cube with parameters in one place
    cube([2,2,82]);
}
module x_block(){ //primitive used for 8 points in the 3d grid
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
    *color([0,1,0]) translate ([02,02,02]) cube([8,8,8]);
    translate ([6,6,6]) sphere (r=1);
}
module vsr_cube(){ //loops for the xyz forests of beams
    union(){
        //xbeam
        for (xj=[0:10:80]){
            for (xi=[0:10:80]){
                translate([0,xi,xj])
                x_beam();
            }
        }
        //ybeam
        for (yj=[0:10:80]){
            for (yi=[0:10:80]){
                translate([yi,0,yj])
                y_beam();
            }
        }
        //zbeam
        for (zj=[0:10:80]){
            for (zi=[0:10:80]){
                translate([zi,zj,0])
                z_beam();
            }
        }
    }
}
module inner_vsr_cube(){ //the manual inner cube
    union(){
        //first set
        hull() {
            translate ([30,50,70]) x_block();
            translate ([10,10,60]) x_block();
        }
        hull() {
            translate ([70,40,50]) x_block();
            translate ([50,00,40]) x_block();
        }
        hull() {
            translate ([20,70,30]) x_block();
            translate ([00,30,20]) x_block();
        }
        hull() {
            translate ([60,60,10]) x_block();
            translate ([40,20,00]) x_block();
        }
        //second set
        hull() {
            translate ([30,50,70]) x_block();
```


(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

```
            translate ([20,70,30]) x_block();
        }
        hull() {
            translate ([10,10,60]) x_block();
            translate ([00,30,20]) x_block();
        }
        hull() {
            translate ([70,40,50]) x_block();
            translate ([60,60,10]) x_block();
        }
        hull() {
            translate ([50,00,40]) x_block();
            translate ([40,20,00]) x_block();
        }
        //third set
        hull() {
            translate ([30,50,70]) x_block();
            translate ([70,40,50]) x_block();
        }
        hull() {
            translate ([10,10,60]) x_block();
            translate ([50,00,40]) x_block();
        }
        hull() {
            translate ([20,70,30]) x_block();
            translate ([60,60,10]) x_block();
        }
        hull() {
            translate ([00,30,20]) x_block();
            translate ([40,20,00]) x_block();
        }
    }
    //fourth full set as option
    *hull() { //hull over all 8 points in 3d space
        translate ([30,50,70]) x_block();
        translate ([10,10,60]) x_block();
        translate ([70,40,50]) x_block();
        translate ([50,00,40]) x_block();
        translate ([20,70,30]) x_block();
        translate ([00,30,20]) x_block();
        translate ([60,60,10]) x_block();
        translate ([40,20,00]) x_block();
    }
}
//paint the outer framed cube 8x8x8
vsr_cube();
//paint the inner cube either as wire or solid or points
*inner_vsr_cube();
```

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified

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.

(CC BY-NC 4.0)2022 Sean Donnellan
VSR - Un-classified