

**Jihočeská univerzita v Českých Budějovicích**  
**Přírodovědecká fakulta**

**Využití open-source programů pro tvorbu výrobků  
na 3D tiskárně a hobby CNC zařízení.**

Bakalářská práce

**Tomáš Malý**

Školitel: Ing. Michal Šerý, Ph.D.

České Budějovice 2020

## **Bibliografické údaje:**

Malý, T., 2020: Využití open-source programů pro tvorbu výrobků na 3D tiskárně a hobby CNC zařízení. [Use of open-source programs for creating products on 3D printers and hobby CNC equipment. Bc. Thesis, in Czech] - 63 p., Faculty of Science, University of South Bohemia, České Budějovice.

## **Anotace:**

Bakalářská práce se zaměřuje na využití open-source programů při tvorbě na hobby CNC a 3D tiskárně. Práce obsahuje popis jednotlivých programů. Jsou zmíněny možnosti programů, podrobný postup tvorby a následná ukázka.

## **Klíčová slova:**

Hobby CNC, 3D tiskárna, Open-source programy, FreeCAD, OpenSCAD, KiCad, Inkscape, FlatCAM

## **Annotation:**

The bachelor's work focuses on the use of open-source programs in the creation of hobby CNC and 3D printers. The work contains a description of individual programs. The possibilities of the programs, the detailed process of creation and the subsequent demonstration are mentioned.

## **Keywords:**

Hobby CNC, 3D printers, Open-source program, FreeCAD, OpenSCAD, KiCad, Inkscape, FlatCAM

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích .....

.....

Tomáš Malý

## **Poděkování**

Rád bych poděkoval svému vedoucímu bakalářské práce, Ing. Michalovi Šerému, Ph.D., za věcné rady a vedení práce. Také bych chtěl poděkovat své rodině a přítelkyni za podporu během studia.

# Obsah

Úvod.....	8
Cíle práce .....	9
<b>I Teoretická část .....</b>	<b>10</b>
<b>1 Základy řízení pomocí G-kódu .....</b>	<b>10</b>
1.1 Historie .....	10
1.2 Definice vztažných bodů CNC stroje .....	10
1.3 Informace obsažené v programu .....	11
1.3.1 Geometrické .....	11
1.3.2 Technologické .....	11
1.3.3 Pomocné .....	11
1.4 Struktura programu.....	12
<b>2 OpenSCAD .....</b>	<b>16</b>
2.1 Základní informace.....	16
2.2 Import a export souborů .....	16
2.3 Prostředí programu .....	17
2.4 Komentáře .....	17
2.5 Datové typy .....	18
2.6 Modelování 2D objektů.....	19
2.6.1 Čtverec.....	19
2.6.2 Kruh.....	19
2.6.3 Polygon.....	20
2.6.4 Text.....	20
2.6.5 Vysunutí 2D objektu do prostoru .....	21
2.6.6 Rotace 2D objektu .....	21
2.7 Modelování 3D objektů.....	22
2.7.1 Krychle .....	22
2.7.2 Koule .....	22
2.7.3 Válec.....	23
2.7.4 Mnohostěn .....	24
2.8 Transformace.....	24
2.8.1 Rotace.....	24
2.8.2 Posun .....	25

2.8.3	Zrcadlení.....	25
2.8.4	Změna velikosti .....	25
2.8.5	Změna barvy .....	26
2.9	Zaoblení hran.....	27
2.9.1	Minkowski.....	27
2.9.2	Hull.....	27
2.10	Booleovské operace.....	28
2.10.1	Rozdíl .....	28
2.10.2	Sjednocení .....	29
2.10.3	Průnik .....	29
<b>3</b>	<b>FreeCAD .....</b>	<b>31</b>
3.1	Orientace v prostředí programu.....	31
3.2	Základní postup vytvoření modelu.....	31
3.2.1	Skica .....	31
3.2.2	Prostorová tvorba.....	33
3.2.3	Výkres .....	33
<b>4</b>	<b>KiCAD.....</b>	<b>34</b>
4.1	Export .....	34
4.2	Výroba plošných spojů .....	34
4.3	Práce v softwaru .....	34
4.3.1	Návrh schématu .....	34
4.3.2	Návrh PCB .....	36
<b>5</b>	<b>Inkscape .....</b>	<b>37</b>
5.1	Postup vytvoření G-kódu.....	37
<b>6</b>	<b>FlatCAM .....</b>	<b>38</b>
6.1	Frézování PCB .....	38
6.2	Vrtání děr pro kontakty součástek.....	39
6.3	Frézování měděné vrstvy.....	40
6.4	Oboustranné frézování PCB .....	41
6.5	Výřez desky .....	41
<b>II</b>	<b>Praktická část .....</b>	<b>43</b>
<b>7</b>	<b>Návod postupu návrhu v jednotlivých softwarech .....</b>	<b>43</b>

7.1	Softwary OpenSCAD a FreeCAD .....	43
7.1.1	Základní deska Arduina.....	43
7.1.2	Ochranná krabička pro Arduino .....	46
7.1.3	Víko ochranné krabičky .....	49
7.2	Software KiCAD .....	52
7.3	Software FlatCAM .....	55
7.3.1	Frézování kontaktů .....	55
7.3.2	Vrtání děr.....	55
7.3.3	Frézování zbylé mědi .....	56
7.3.4	Výřez desky .....	57
<b>Závěr.....</b>	<b>.....</b>	<b>58</b>
<b>Seznam literatury .....</b>	<b>.....</b>	<b>59</b>
<b>Externí odkazy .....</b>	<b>.....</b>	<b>63</b>

# Úvod

Tuto práci jsem se rozhodl vypracovat za účelem seznámení s open-source programy pro tvorbu na 3D tiskárně a hobby CNC. V dnešní době je to z mého pohledu velmi aktuální téma. Hodně lidí se zabývá 3D tiskem za účelem tisknout náhradní díly, prototypy nebo objekty, které nelze vyrobit jiným způsobem. Další část se zaměřuje na frézování nebo laserové vypalování částí konstrukcí v rámci modelářství z různých materiálů (balza, dřevo, plast). Pro prototypovou tvorbu se využívá i frézování DPS (deska plošných spojů; v angličtině PCB). Pro každou z těchto aplikací je potřebné vymodelovat objekt v daném softwaru. Tyto softwary je možné zakoupit a pravidelně platit licenci. Pro domácí využití je nesmyslné pořizovat software za několik desítek tisíc, a proto bych vás chtěl seznámit s několika základními open-source programy, které jsou vhodné pro domácí i komerční účely.

V teoretické části je krátké seznámení a historie G-kódu, který je řídicím prvkem pro ovládání CNC zařízení. Dále jsou zmíněny vybrané open-source programy s popisem prostředí a krátkým postupem práce.

V praktické části jsem se zaměřil na ukázkou jednotlivých programů na konkrétním návrhu. Tato část slouží jako manuál pro tvorbu a orientaci v programech. Ke každému z nich jsou přiložené obrázky dané pracovní plochy. Ukázka také obsahuje generování G-kódu pro koncové zařízení.

Všechny zmíněné softwarové produkty v této práci jsou svobodné a volně dostupné na webových stránkách vývojářů daného programu.



# Cíle práce

Cílem této práce je na základě několika typových výrobků prostudovat možnosti open-source programů pro tvorbu a následnou realizaci na 3D tiskárně typu Průša nebo CNC zařízení a vytvořit návod na využití vybraných softwarových produktů.

## Dílčí cíle

- možnosti programů
- omezení programů
- postup tvorby
- praktická ukázka

# I Teoretická část

## 1 Základy řízení pomocí G-kódu

### 1.1 Historie

Programování prostřednictvím G-kódu se využívá zejména u CNC strojů. Tyto stroje se pomalu vyvíjejí již od počátku 50. let 20. století. V Americe byly vyvinuty první numericky řízené stroje tzv. NC stroje. Jednalo se o klasický stroj vybavený jednoduchým řídicím systémem s elektronkovými obvody, který zpracoval vložené informace do čtecího zařízení, převedl a následně odeslal informace do místa výkonu požadovaným příkazem. V této době byl nejpoužívanější typ nosiče děrná páska nebo štítek.

Rozvoj byl urychlen na začátku 70. let z důvodu vývoje elektroniky. V této době se automatické systémy inovovaly na první NC systémy se základními integrovanými obvody.

V dnešní době jsou CNC stroje konstrukčně velmi podobné těm z minulého století, avšak liší se technologickými možnostmi a parametry. Odlišnosti a zdokonalení jsou viditelné zejména v oblasti: výměny nástrojů, mezioperační dopravy obrobků, třískového hospodářství, přesnosti výroby 0,001mm, opakovatelnosti. [1]

Číslicově řízené výrobní stroje CNC (Computer Numerical Control) jsou charakteristické tím, že ovládání pracovních funkcí stroje je prováděno řídicím systémem pomocí vytvořeného programu. Informace o požadovaných činnostech jsou zapsány v programu pomocí alfanumerických znaků. Vlastní program je dán posloupností oddělených skupin znaků, která se nazývá bloky nebo věty. Program je určen pro řízení silových prvků stroje a zaručuje, aby proběhla požadovaná výroba součástky. [2]

### 1.2 Definice vztažných bodů CNC stroje

M - nulový bod stroje (stanoven výrobcem stroje – výchozí počátek souřadného systému)

W - nulový bod obrobku (jeho polohu definuje programátor, váží se k němu všechny programované souřadnice drah v NC programu, jeho poloha je měřena od bodu M)

N - nulový bod nástrojového držáku (stanoven výrobcem stroje – referenční bod nástrojového držáku, ke kterému se vztahují rozměry všech nástrojů)

P - nulový bod nástroje (fréza - bod leží v ose nástroje na jeho čele)

R - referenční bod (jeho poloha dána výrobcem stroje) – po zapnutí stroje slouží k nalezení výchozího počátku souřadnicového systému M; nemá význam, pokud má stroj absolutní odměřování [3]

## **1.3 Informace obsažené v programu**

### **1.3.1 Geometrické**

Popisují dráhy nástroje, které jsou dány rozměry obráběné součásti, způsoby jejího obrábění a popisují příjezd a odjezd nástroje k obrobku a od něho. Rozlišuje pracovní posuvy, rychloposuvy a dále dráhu nástroje (po přímce nebo po kruhovém oblouku). Jde tedy o popis drah nástroje v kartézských souřadnicích, kdy pro tvorbu programu potřebujeme rozměry z výrobního výkresu. U frézky je v programu uveden popis v osách X, Y, Z danými funkcemi (a často i v dalších osách dle konstrukce stroje jako např. rotace okolo jednotlivých os A, B, C), danými funkcemi, které stanoví norma ISO a také jednotliví výrobci řídicích systémů. [2, 4]

### **1.3.2 Technologické**

Stanovují technologii obrábění z hlediska řezných podmínek (jsou to zejména otáčky nebo řezná rychlost, posuv, případně hloubka třísky).

### **1.3.3 Pomocné**

Jsou to informace, povely pro stroj pro určité pomocné funkce (např. zapnutí čerpadla chladicí kapaliny, směr otáček vřetene, dopravník špon atd.). [2]

## 1.4 Struktura programu

„Na začátku programu je před prvním řádkem (blokem) uveden znak %, za znakem je uvedeno číslo programu - to platí pro většinu řídicích systémů. Před tímto znakem lze uvádět informace, které stroj nerozpozná. Např. poznámky, jako je název součásti atd. Poznámky lze uvést i za znakem v programu, ale je nutné je dát do závorky. Jsou řídicí systémy, které tento znak nepožadují, potřebné poznámky se obvykle uvádějí v programu například funkcí G.“  
[2, s. 33]

Tabulka 1 - Složení programu v bloku [2]

Příklad				Název	Doporučené pořadí adres jednotlivých slov ve větě je: N G (M) X Y Z F S T D, nemusí se dodržovat, záleží na daném řídicím systému. Doporučuje se dodržovat pro větší přehlednost a kontrolu
N 40 G 00 X 100 Z -50				Blok (věta)	
N 40	G 00	X 100	Z -50	Příkaz (slovo)	
N	G	X	Z	Adresa	
40		00		Významová část	
100		50		Rozměrová část	

Tabulka 2 - Význam jednotlivých písmen [2]

Význam nejpoužívanějších adres:	
Písmeno	
X Y Z	Základní osy souřadnicového systému - pohyby v osách.
A B C	Rotace kolem základních os.
I J K	Parametry interpolace nebo stoupání závitů ve směru os.
P Q R	Pohyb paralelně podél základních os.
R	Některé systémy používají R jako parametr v podprogramech.
U V W	Druhý pohyb paralelně se základními osami.
T	Nástroj.
D	Paměť korekce nástrojů.
G	Přípravná (geometrická) funkce.
M	Pomocná (přídavná) funkce.

N	Číslo bloku (věty).
F	Posuv.
S	Otáčky vřetene. Konstantní řezná rychlost.
L	Volání programu.

„Pozn.: Některé z uvedených písmen abecedy jsou pro výrobce řídicích systémů závazná, některá doporučená. Neobsazená písmena abecedy jsou volná, výrobci je obsazují dle specifik svých řídicích systémů, podle možností daných strojů, pro které jsou především určena.“  
[2, s.35]

Tabulka 3 - Informace obsažené v programu [2]

Název a příklad	Užití
Věta (blok) Věta musí začínat písmenem N a číslem např. N40 Př. N 40 G 00 X 100 Z -50 (obvyklé u výukových systémů, ale nemusí u většiny systémů ve výrobní praxi)	Čísluje se obvykle po desítkách, aby bylo možné dodatečné vložení dalších vět například při opravě programu. Řídicí systém obvykle seřazuje bloky podle čísel vzestupně a v tomto pořadí je čte a stroj vykonává zadané příkazy. Vzestupnost čísel slouží též pro lepší orientaci programátora v programu. Pokud by následující věta obsahovala některé stejné instrukce, nemusí se psát, jsou platné do té doby, než budou přepsány = modální funkce.
Přípravné (hlavní) funkce G (Go) Př. G00; G01; G42 Př. G0; G1; G42	Zpracování geometrické informace. Některé systémy připouštějí vložit i více G funkcí do jedné věty. Dvojmístné číslo se nemusí použít, pokud je první číslo 0. Některé systémy používají více než dvojmístná čísla G a také M.
Pomocné funkce M (Machine) Př. M04, také M4	Vyvolávají činnosti mechanismu stroje. Některé se také týkají řídicího systému.
Informace o dráze Př. X20 Z-30	Jsou zadány cílovým bodem v souřadnicích absolutně - G90, nebo přírůstkově G91.
Funkce nástroje T (Tool)	T a D se udávají obvykle dvojmístným číslem vzájemně

Korekce nástroje D Př. T01 D01	souvisejícím. Př. T01 D01 současně zpracovává, přikazuje k danému nástroji dané korekce. Některé řídicí systémy mají jiné řešení přikazování korekcí k nástrojům.
Posuv funkce F (Feed)	Velikost posuvů u frézky je zadána v mm za minuty [mm·min <sup>-1</sup> ]
Otáčkové funkce S (Speed)	Velikost otáček je zadána, za minutu [min <sup>-1</sup> ], řezné rychlosti [mm·min <sup>-1</sup> ]

Tabulka 3 - Přehled nejpoužívanějších G-funkcí [2]

Označení funkce	Název funkce	Použití	
G00	Lineární	Programuje se v souřadnicích os, uvádí se cílový bod v souřadnicích, případně další adresy.	
G01	interpolace		
G02	Kruhová		
G03	interpolace		
G17	Pracovní rovina	X - Y	Určení roviny, ve které se provádí pracovní posuvy a rychloposuvy.
G18		Z - Y	
G19		Y - Z	
G33	Řezání závitů	Určuje se proměnlivá hloubka třísky a počet hlazení bez přídatku.	
G54-59	Posuvy nulového bodu	Posuny absolutně i přírůstkově, na začátku i v průběhu programu.	
G90	Absolutní	Programování - popis drah nástroje v souřadnicové soustavě.	
G91	Přírůstkové	Programování - popis drah nástroje, o kolik se posune v osách.	

Tabulka 4 - Přehled nepoužívanějších M-funkcí [2]

Označení funkce	Název funkce	Použití
M03	Otáčky vřetene	Ve směru hodinových ručiček Proti směru hodinových ručiček
M04		
M05	Zastavení vřetene	
M06	Výměna nástroje	Do této funkce se doplňují délkové korekce.
M07-08	Zapnutí čerpadla	Chlazení, mazání obrobku při obrábění.
M09	Vypnutí čerpadla	
M17	Konec programu	Vrací do hlavního programu (hlavní programy mohou používat podprogramy, v těchto je odvolání, které vyvolá podprogram).
M30	Konec hlavního programu	Návrat na začátek hlavního programu.

## 2 OpenSCAD

### 2.1 Základní informace

OpenSCAD je svobodný software pro tvorbu 3D modelů, dostupný na Linux/UNIX, MS Windows a Mac OS X. Od běžných interaktivních modelovacích softwarů se liší tím, že objekty jsou tvořeny pomocí zdrojového kódu v editoru. Je vhodným nástrojem pro modelování a tvorbu objektů, zejména mechanického typu, pravidelných tvarů nebo součástek strojů. Není vhodný pro nepravidelné a křivkové součásti.

Poskytuje dvě možnosti navrhování objektů. První z nich je vytváření objektů pomocí 2D náčrtu a následného vysunutí do prostoru. Druhou možností je přímo vytvořit 3D objekty.

Software umožňuje načítání AutoCAD dat jako je DXF a čtení jejich parametrů. Jako výstup lze vytvořit soubory DXF, STL nebo OFF, které jsou potřebné pro následné vytvoření modelu na koncovém zařízení. Celý přehled podporovaných formátů viz níže. [5]

### 2.2 Import a export souborů

#### Import souborů

3D typy souborů - STL, OFF, AMF, 3MF

2D typy souborů - DXF, SVG

Lze importovat i obrázek ve formátu PNG.

Importování lze provést pomocí příkazu přímo v kódu:

```
import_dxf("název_souboru.dxf");  
import_stl("název_souboru.stl");  
surface("název_souboru.png");
```

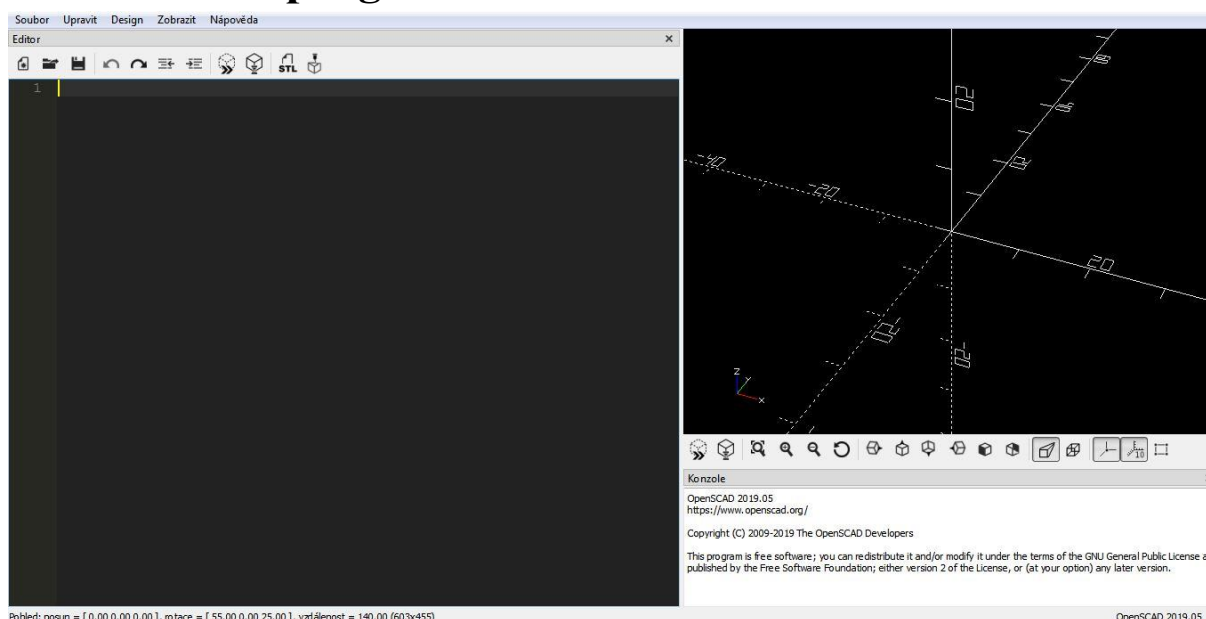
#### Export souborů

Exportování modelu je možné až po renderování F6. Cesta k exportu *Soubor* → *Exportovat*.

Podporované formáty - STL, OFF, AMF, DXF, SVG, CSG, PNG [6]



## 2.3 Prostředí programu



Obrázek 1: Prostředí OpenSCAD [Tomáš Malý]

V levé části programu se nachází editor pro zapisování zdrojového kódu. V pravé horní části se nachází grafické zobrazení a pod ním konzole, která vypisuje informace prováděných operací nebo chybová hlášení.

Vzhled editoru a grafické zobrazení prostředí (světlé, tmavé a barevné) je možné libovolně změnit v horní liště nástrojů *Úpravy* → *Předvolby*.

### Zobrazení objektu

Pro generování objektů můžeme použít rychlý náhled F5, který není detailní a nezobrazuje některé výřezy. Pro přesné vykreslení všech hran a ploch využijeme tzv. renderování (vykreslení) F6, které u komplexních objektů může trvat déle.

## 2.4 Komentáře

Komentáře jsou nedílnou součástí každého kódu, pro lepší přehlednost ve složitých i jednoduchých projektech.

Mohou sloužit nám programujícím, když se ke kódu vracíme po delší době, aby nám pomohly s lepší orientací v kódu. Nebo také budoucímu programátorovi, který by kód využil nebo upravoval. [7]

Komentáře se řídí stylem C++

```
// toto je jednořádkový komentář  
/* takto ohraničíme víceřádkové komentáře */
```

## 2.5 Datové typy

### Matematické funkce

Software podporuje většinu matematických funkcí, jako jsou funkce sinus, cosinus, tangens, mocninu, odmocninu, logaritmus aj.

Celý přehled podporovaných matematických funkcí najdete viz Seznam literatury [8]

### Čísla

Nejpoužívanější hodnoty v OpenSCAD jsou zapisovány v desítkové soustavě. Je možné zadávat kladná čísla (20), záporná čísla (-20), desetinná čísla (20.5) a velká čísla pomocí exponenciálního zápisu ( $1.2468e7 = 1,2468 \cdot 10^7$ ).

### Booleovy hodnoty

Jsou to hodnoty *true* a *false*. Používají se při centrování těles (čtverec, krychle, válec) vůči souřadnému systému (viz níže).

### Proměnné

Proměnné se vytvoří pomocí příkazu nebo identifikátoru, přiřazením výrazu a zakončené středníkem. Využívají se pro snadnou a rychlou změnu parametru v komplexních i jednoduchých projektech.

## Syntaxe

promenna=hodnota/funkce;

Vytvoření kostky pomocí proměnné:

```
strana_a=25;  
strana_b=50;  
strana_c=100;  
cube ([strana_a, strana_b, strana_c]);
```

Podkapitola *Datové typy* byla čerpána z webového zdroje dostupného viz Seznam literatury [9]

## 2.6 Modelování 2D objektů

### 2.6.1 Čtverec

Při vytvoření čtverce se primárně umístí dolní levý roh do středu souřadného systému [0, 0] *false* (není potřeba zadávat  $\rightarrow$  *square* (*x*, *center = false*); = *square* (*x*); ). Při volbě *true* je umístěn střed čtverce do středu souřadného systému.

## Syntaxe

**square**(*x*);                   - zadaná hodnota do závorky udává stejnou délku v ose x a y

**square**([*x*,*y*]);               - při zadávání více hodnot je nutné použít hranaté závorky, první hodnota určuje délku strany v ose x a druhá hodnota délku v ose y  $\rightarrow$  možnost tvorby obdélníku

Příklad vytvoření obdélníku o rozměrech 50 mm x 30 mm:

```
square([50,30]);
```

### 2.6.2 Kruh

Při tvorbě se primárně umístí střed kruhu do středu souřadného systému [0, 0].

Pro způsob vykreslování detailů je možné před příkaz použít jeden z uvedených parametrů:

\$fa - minimální úhel každého fragmentu

\$fs - minimální obvodová délka každého fragmentu tvořící obvod kruhu

\$fn - pevně daný počet fragmentů v 360° rozsahu

### Syntaxe

`circle(r=poloměr);` - možnost zadávání pomocí poloměru

`circle(d=průměr);` - možnost zadávání pomocí průměru

Příklad vytvoření kruhu o poloměru 5 mm:

```
circle(r=5);
```

### **2.6.3 Polygon**

Umožňuje vytvoření mnohoúhelníku pomocí [x, y] souřadnic. Je možné vytvořit komplexní útvary.

`points` - vkládání libovolného počtu bodů [x, y] určující vrcholy mnohoúhelníku

`paths` - vybírání bodů z položky `points` pomocí vektorů a určení jejich pořadí od 0 do n-1

### Syntaxe

```
polygon(points=[[x1,y1],[x2,y2], ... ], paths=[[v1,v2, ...], ... ]);
```

Příklad vytvoření čtverce s trojúhelníkovým výřezem:

```
polygon(points=[[0,0],[60,0],[60,60],[0,60],[10,10],[50,10],[30,50]],  
paths=[[0,1,2,3],[4,5,6]]);
```

### **2.6.4 Text**

Vytváří text jako 2D objekt, u kterého můžeme měnit velikost a styl písma. Tento nástroj podporuje používání diakritiky ve vytvářeném textu podle zvoleného fontu písma.

Samotný text musí být vložen do uvozovek. Dále můžeme měnit velikost písma příkazem *size*.

Jako další můžeme měnit styl písma, který musí být také vložen do uvozovek. Celý přehled písem je k nalezení v ovládacím panelu *Nápověda* → *Seznam písem*.

### Syntaxe

```
text ("Zadání textu", size=30, font="Andalus");
```

## 2.6.5 Vysunutí 2D objektu do prostoru

Všechny zde zmíněné objekty vytvořené ve 2D lze následovně vytáhnout do prostoru. K vytažení objektu do třetího rozměru nám pomůže příkaz *linear\_extrude ( ) { ...}*.

### Syntaxe

```
linear_extrude(výška vysunutí){objekty}
```

Příklad vytvoření pravidelného lichoběžníku o tloušťce 3mm.

```
linear_extrude(3){
  polygon(points=[[0,0],[60,0],[50,30],[10,30]],
    paths=[[0,1,2,3]]);
}
```

## 2.6.6 Rotace 2D objektu

Z vytvořeného 2D objektu je také možné vytvořit rotační součást. Základní myšlenkou je navrhnout tvar na rovině x-y a poté jej rotovat okolo osy z. Základním prvkem je kruh, ale je také možné rotovat polygon.

Parametr *convexity* určuje maximální počet protnutých stran jedním paprskem skrz celý tvar a není potřeba ho vyplňovat, pokud bude zobrazení objektu správné. Pokud je potřeba parametr vyplnit, je nutné zvolit vždy celé číslo, např.: 10.

### Syntaxe

```
rotate_extrude(angle=360, convexity = 10){ ... }
```

Příklad vytvoření háku s hrotem a okem pro uchycení:

```
$fn=50;
translate([0,55]){
```

```

rotate_extrude(angle=270, convexity = 10)
    translate([40,0]) circle(5);
}

translate([40,55,0]) rotate([90,0,0]) cylinder(r1=5, r2=0, h=15); //hrot

rotate_extrude(angle=90, convexity=10)
    translate([15, 0]) circle(5);

translate([15,-80,0]) rotate([-90,0,0])cylinder(r=5,h=80);

translate([15,-90,0]){ //oko
rotate_extrude(angle=360, convexity = 10)
    translate([10,0,0])circle(5);
}

```

Kapitola Modelování 2D objektů byla inspirována z internetových zdrojů dostupných v Seznamu literatury: [11, 12, 13, 14]

## 2.7 Modelování 3D objektů

### 2.7.1 Krychle

Při tvorbě je krychle primárně umístěna v prvním kvadrantu *false* (není potřeba zadávat stejně jako u čtverce). Volba *true* vycentruje střed krychle do středu souřadného systému [0, 0, 0].

#### Syntaxe

```

cube(x, centre=true/false); // zadaná hodnota vytvoří krychli

cube([x,y,z], center=true/false ); // při zadání všech souřadnic lze vytvořit kvádr

```

Příklad vytvoření kvádrů o rozměrech 50 mm x 30 mm x 20 mm:

```
cube([50,30,20]);
```

### 2.7.2 Koule

Koule se primárně vytvoří uprostřed souřadného systému.

Pro způsob vykreslování detailů je možné před příkaz použít jeden z uvedených parametrů:

\$fa - minimální úhel každého fragmentu (bez zadání je číslo primárně 12, menší číslo → hladší povrch)

$\$fs$  - minimální obvodová délka každého fragmentu tvořící obvod koule

$\$fn$  - pevně daný počet fragmentů v  $360^\circ$  rozsahu - pokud je číslo zadáno, budou hodnoty  $\$fa$  a  $\$fn$  ignorovány

### Syntaxe

```
sphere(r=poloměr);  
sphere(d=průměr);
```

Příklad vytvoření koule o poloměru 25 mm:

```
sphere(r=25);
```

### 2.7.3 Válec

Vytvoření s použitím *false* umístí osu válce do osy z a základnu válce do souřadnic [0, 0, 0].

S použitím *true* je osa válce umístěna do osy z a osy x, y protínají válec v polovině výšky.

Při zadání rozdílných poloměrů/průměrů je možné vytvořit kužel nebo komolý kužel.

Možnost použití parametrů  $\$fa$ ,  $\$fs$ ,  $\$fn$  jako u předchozích objektů.

Při zadání parametru  $\$fn$  je možné vytvořit třiboký, čtyřboký jehlan nebo více (např.  $\$fn = 4$ ).

### Syntaxe

```
cylinder(h=výška, r1=poloměr základny, r2=poloměr vrcholu, center=true/false);  
cylinder(h=výška, d1=průměr základny, d2=průměr vrcholu, center=true/false);
```

Pozn.: Pokud zadáme pouze r/d pak se  $r1 = r2$  a  $d1 = d2$ .

Příklad vytvoření válce o výšce 50 mm a průměru 25 mm:

```
cylinder(h=50, r=25);
```

Příklad vytvoření komolého kužele o výšce 50 mm, poloměr základny 25 mm a poloměr vrcholu 10 mm:

```
cylinder(h=50, r1=25, r2=10);
```

Příklad vytvoření kužele o výšce 50 mm s poloměrem základny 25 mm:

```
cylinder(h=50,r1=25,r2=0);
```

Příklad vytvoření čtyřbokého jehlanu vysokého 60 mm. Vrcholy základny jsou umístěny na osách x, y a jsou úhlopříčně vzdáleny 50 mm:

```
$fn=4;  
cylinder(h=60,r1=25,r2=0);
```

## 2.7.4 Mnohostěn

Vytváří se pomocí bodů *points* tvořících vrcholy mnohostěnu o daném tvaru. Tyto body jsou řazeny pomocí uspořádání vektorů *faces*, které slouží k tvorbě stěn mnohoúhelníku. Vektory [v1, v2, v3] jsou číslovány od 0 do n-1.

### Syntaxe

```
polyhedron (points = [[x, y, z], ... ], faces=[[v1, v2, v3], ... ]);
```

Příklad vytvoření trojúhelníkového hranolu:

```
polyhedron(points=[[0,0],[100,0],[100,100],[0,100],  
                  [100,0,100],[0,0,100]],  
            faces=[[0,1,2,3],  
                  [0,1,4,5],  
                  [1,2,4],  
                  [0,3,5],  
                  [2,3,5,4]]);  
//základna  
//stěna na rovině Z-X  
//základna  
//přední část  
//pravý bok  
//levý bok  
//horní strana
```

Kapitola Modelování 3D objektů byla inspirována z internetových zdrojů dostupných v Seznamu literatury: [14]

## 2.8 Transformace

Důležitým pravidlem pro kombinaci několika transformací najednou je jejich pořadí. Transformace zapisujeme vždy před upravovaný objekt v takovém pořadí, v jakém chceme změny provést.

### 2.8.1 Rotace

Umožňuje rotaci kolem os souřadného systému.



Pro rotaci více objektů nebo sestavu objektů můžeme příkaz rozšířit o složené závorky "{...}", do kterých vložíme celou sestavu.

### Syntaxe

```
rotate([stupeň_x, stupeň_y, stupeň_z]) { ... }
```

```
rotate(a=stupně, v=[x,y,z]) { ... }
```

V druhém případě nahradíme písmena označující vektory  $x$ ,  $y$ ,  $z$  za 0 nebo 1 podle toho, okolo jaké osy chceme objekt rotovat.

## 2.8.2 Posun

Posunutí objektů pomocí souřadnic  $[x, y, z]$ . Pro posun více objektů nebo soustavy objektů můžeme příkaz také rozšířit složenými závorkami jako u rotace.

### Syntaxe

```
translate ([x, y, z]) { ... }
```

## 2.8.3 Zrcadlení

Tento nástroj je schopen zkopírovat objekt a zrcadlit ho dle zvolené roviny.

### Syntaxe

```
mirror (v=[x, y, z]) { ... }
```

Za hodnoty označující vektory  $x$ ,  $y$ ,  $z$  zadáváme 0 nebo 1 podle toho, okolo jaké osy se objekt zrcadlí.

## 2.8.4 Změna velikosti

Změní velikost daného předmětu dle zadaných hodnot ve 2D nebo 3D. Lze měnit například kruh v elipsu nebo čtverec v obdélník. Příkaz se zadává před transformovaný objekt.

Při použití příkazu s vyplněním `auto=true` stačí do upravovaných rozměrů zadat pouze jeden ze tří parametrů. Ostatní rozměry se transformují ve stejném poměru s původním objektem. Vhodné při potřebě změnit symetricky všechny strany objektu.

## Syntaxe

```
resize(newsize=[x,y,z]);  
resize([x,y,z]);  
resize([x,y,z], auto=true);
```

### **Příklad použití *auto=true*:**

Původní objekt – kvádr o rozměrech 30 mm x 50 mm x 25 mm. Nový rozměr v ose x = 90. To je třikrát větší než původní, a proto se kvádr v ose y a z zvětší také třikrát.

```
resize(newsize=[90,0,0],auto=true) cube([30,50,25]);
```

## **2.8.5 Změna barvy**

Určuje vzhled vymodelovaných částí, které chceme odlišit od celku. Změna se projeví pouze pro náhled v F5. Při renderování (vykreslování) F6 není zatím rozlišení barev dostupné.

Je možné zadávat přímo hodnoty barev (Red, Green, Blue), které nabývají hodnot od 0 do 255. Hodnoty R, G, B mají rozsah pouze 0 až 1, a proto je nutné zadávat hodnoty ve zlomku např.: R/255.

Další možností je zadat tzv. hexvalue. Je to hodnota v hexadecimální soustavě, která se skládá z čísel od 0 do 9 a písmen od A do F. Například hodnota #FF0000 představuje červenou barvu. Každá pozice čísla určuje hodnotu určité barvy a dělí se na (RR, GG, BB). [15]

Libovolné barvy lze namíchat a poté vygenerovat jejich hexadecimální hodnotu například na webu viz Seznam literatury. [16]

## Syntaxe

```
color(c = [R/255, G/255, B/255] { ... }  
color("#hexvalue") { ... }  
color("colorname") { ... }
```

Příklad vytvoření krychle o hraně 10 mm s oranžovou barvou pro různé syntaxe:

```
color(c=[237/255,66/255,12/255]) cube(10);  
color("#ED420C") cube(10);  
color("orange"); // odlišný odstín oranžové
```

## 2.9 Zaoblení hran

Zaoblení hran je jeden z nejpoužívanějších nástrojů v tomto softwaru. Používá se pro propojování různých rovin objektů, zakulacení rohů z konstrukčního důvodu nebo lepšího vzhledu. Pro zaoblení hran je možné použít dva podobné nástroje *minkowski*( ) a *hull*( ).

### 2.9.1 Minkowski

Používá se pro zaoblení hran, například u vytváření krabičky a je možné ho využít pouze pro prostorovou tvorbu.

Příkaz *minkowski* musí obsahovat základní těleso např.: kvádr a válec, podle kterého budou hrany zaobleny. Minkowského součet funguje tak, že na každou stranu délky a šířky kvádru je přičten poloměr válce a k výšce kvádru je přičtena výška válce.

Pokud bude základní těleso kvádr o rozměrech 100 mm x 50 mm x 20 mm a válec o poloměru 10 mm a s výškou 15 mm, poté budou rozměry vytvořeného tělesa následující:

Délka:  $100 + 10 + 10 = 120$  mm

Šířka:  $50 + 10 + 10 = 70$  mm

Výška:  $20 + 15 = 35$  mm

Příklad vytvoření zaobleného kvádru. Pokud bychom chtěli, aby byly výsledné rozměry kvádru 100 mm x 50 mm x 30 mm s poloměrem zaoblení 10 mm, je potřeba počítat s minkowského součtem a zvolit vhodné rozměry viz níže v kódu:

```
$fn=50;
minkowski(){
    cube([80,30,20]);
    cylinder(r=10,h=10);
}
```

### 2.9.2 Hull

Příkaz *hull* vytváří celé těleso pomocí válce, nebo kruhu a je možné ho využít ve 2D i 3D tvorbě.

Je možné vytvořit pomocí válce/kruhu zaoblené tvary libovolných průřezů (čtverec, obdélník, trojúhelník, mnohoúhelník atd.).

Budeme-li chtít zaoblený kvádr, vytvoříme ho pomocí čtyř posunutých válců. Nesmíme zapomenout, že posunutím válců určujeme polohu jejich středů, a proto je potřeba u posunutí odečíst/přičíst poloměr válců tak, aby nám vznikl požadovaný rozměr.

Pro vytvoření kvádrů s poloměrem zaoblení 5 mm a rozměrech 100 mm x 50 mm x 10 mm jsme zvolili následující rozměry v kódu:

```
$fn=50;
hull(){
    translate([5,5]) cylinder (r=5,h=10);
    translate ([95,5]) cylinder (r=5,h=10);
    translate ([95,45]) cylinder (r=5,h=10);
    translate ([5,45]) cylinder (r=5,h=10);
}
```

Kapitola Transformace byla inspirována z internetových zdrojů dostupných v Seznamu literatury: [17]

## 2.10 Booleovské operace

### 2.10.1 Rozdíl

Možné používat při tvorbě 2D i 3D, ale není doporučeno kombinování. Použití zejména při vytváření děr, výběrů, drážek nebo dutin. Založeno na odečítání objektů od prvního výchozího objektu zadaného v kódu.

Mezi složené závorky zadáváme jako první objekt, od kterého budeme odebírat všechny následující zadané tvary.

#### Syntaxe

```
difference( ) {...}
```

Příklad odebrání čtvercového průřezu tělesa skrz válec:

```
difference ( ) {
    cylinder(h=50, r=10, center=true); //vytvoření válce v ose z
    cube([20,10,10], center=true); //odebrání kvádrů skrz válec
}
```

## 2.10.2 Sjednocení

Využívá se pro sjednocení dvou nebo více objektů do jednoho celku.

Nutné použít například při vytváření děr skrz složený objekt.

### Syntaxe

```
union ( ) { }
```

Příklad sjednocení dvou válců tvořící kříž, ve kterém je uprostřed vytvořena čtvercová díra:

```
difference ( ){  
    union ( ) {  
        cylinder (h = 100, r = 10, center = true);  
        rotate(a = 90, v = [1,0,0] )cylinder (h = 100, r = 10,  
        center = true);  
    }  
    cube ([20,10,10], center = true);  
}
```

Pozn.: Kdyby v tomto případě nebyl použit příkaz *union( )*, byl by ze základního válce nejprve odebrán druhý válec a poté kvádr. V tom případě by vzniklo jiné těleso.

## 2.10.3 Průnik

Při použití vzniká průnik dvou objektů → zůstává pouze společná část obou těles. Zbytek je odstraněn.

### Syntaxe

```
intersection ( ){ }
```

Příklad průniku dvou válců:

```
intersection( ) {  
    cylinder(h = 20, r=5, center = true, $fn=50);  
    rotate (a=90,v=[90,0,0]) cylinder (h = 20, r=5, center = true, $fn=50);  
}
```

Kapitola Booleovské operace byla inspirována z internetových zdrojů dostupných v Seznamu literatury: [18]

Video ukázka tvorby v softwaru OpenSCAD je dostupná v Externích odkazech na konci práce.

## 3 FreeCAD

Jedná se o parametrický CAD modelovací program, který je vyvíjený od roku 2002 a je zcela svobodný. Program umožňuje vytvářet modely ve 2D i 3D, z nich dále vytvořit výkresy nebo sestavu objektů. Software podporuje import a export dat typu DXF, SVG, STEP, STL aj.

Modelování je založeno na vytvoření skici, která musí být zcela zakótována a celý profil musí být uzavřen. Při kreslení skici můžeme použít základní geometrické vazby. Po vytvoření skici můžeme náčrt vysunout do prostoru nebo jej rotovat okolo zvolené osy. [19]

### 3.1 Orientace v prostředí programu

Styl navigace je možné zvolit kliknutím pravého tlačítka na pracovní plochu a zvolit danou navigaci. Software nabízí rozsáhlou řadu navigací, které se používají v jiných modelovacích softwarech, jsou na výběr navigace typu CAD, OpenInventor, Blender aj. [20]

### 3.2 Základní postup vytvoření modelu

Při vytváření modelu zvolíme v hlavní nabídce pracovních prostředí *Part design*, kde se zpřístupní veškeré potřebné prvky.

1. Vytvořit nový prázdný dokument (Ctrl + N)
2. Vytvořit novou skicu
3. Plně zakótovat výkres
4. Vytvořit prostorový objekt

#### 3.2.1 Skica

Zvolíme *Vytvořit novou skicu*, dále si zvolíme *Základní rovinu*, na které bude náčrt ležet.

Po zvolení roviny se zpřístupní nástroje pro tvorbu náčrtu. Při vytváření obrysu součásti se v levé části pracovní plochy (záložka *Tvorba*) zobrazí počet stupňů volnosti objektu, všechny prvky tvořící náčrt a jejich vazby. Počet stupňů volnosti můžeme snížit pomocí zakótování

výkresu nebo přidáním vazeb do té doby než dosáhneme počtu volnosti 0 a program nám umožní skicu uzavřít.

Základní prvky pro tvorbu 2D objektů a jejich možnosti vytvoření:

- bod
- úsečka - pro tvorbu je možné použít písmeno L
- oblouk - oblouk je možné vytvořit pomocí středu a koncové bodu, koncových bodů a okrajového bodu
- kruh - střed a okrajový bod nebo 3 okrajové body
- elipsa - elipsa pomocí středu, hlavní poloosy a bodu; oblouku elipsy pomocí středu, hlavní poloosy a koncových bodů
- splajn-křivka - B splajn pomocí kontrolních bodů
- lomená čára
- obdélník
- pravidelný mnohoúhelník
- drážka

Základní vazby:

- bodová vazba - umístění bodu na objekt
- svislá
- vodorovná
- rovnoběžná - rovnoběžná vazba mezi dvěma úsečkami
- kolmá - kolmá vazba mezi dvěma čarami
- tečná
- rovnost dvou prvků
- symetrická - vytvoří symetrii mezi dvěma body vzhledem k čáře nebo bodu

Základní kóty:

- vodorovná - kótování mezi dvěma body nebo zvolení úsečky
- svislá - kótování mezi dvěma body nebo zvolení úsečky
- fixující - možnost použití mezi bodem a úsečkou



- průměr - zadá průměr kružnice nebo oblouku, lze zakótovat i poloměr
- úhel - zadá úhel mezi dvěma čarami

Kapitola Skica byla inspirována z internetových zdrojů dostupných v Seznamu literatury: [21]

### 3.2.2 Prostorová tvorba

Pro vytvoření prostorového objektu využijeme vytvořenou skicu, kterou můžeme dále vysunout do prostoru o určité šířce, rotovat okolo zvolené osy, vytvořit těleso pomocí prvku tažení podél trasy nebo plné těleso pomocí profilování.

Prostorové prvky:

- vysunutí - vysunutí náčrtu do třetího rozměru
- rotace - zvolení skici, osy rotace, úhlu rotace
- tažení - umožňuje spojení dvou prvků v různých rovinách (např.: kruh - čtverec) po přímce a vytvoření 3D tvaru
- profilování - lze táhnout určitý tvar, podobně jako u tažení, podél předem dané trasy

Vytvořené objekty můžeme nadále upravovat pomocí zaoblení hran nebo sražení.

V prostorové tvorbě lze využít Booleovských operací sjednocení, průnik, rozdíl.

Prostorové prvky lze také vytvořit v pracovním prostředí *Part*. Ve výběru je možné zvolit krychli, válec, kouli, kužel nebo prstenec. Všechny tyto objekty je možné dále upravovat podle daných parametrů v levém dialogovém okně v záložce *údaje*.

### 3.2.3 Výkres

Aby mohl být zhotovený model vyroben, je potřeba vytvořit výkresovou dokumentaci. K tomu slouží pracovní prostředí *TechDraw*, které se nachází v horní liště nástrojů.

Hlavní pohled výkresu, který je výchozí pro ostatní pohledy, je tvořen z pohledu v modelovacím prostředí 3D objektu. Ve výkresu lze zvolit několik pohledů a miniaturu celé součásti. Dále lze vytvořit řez součásti, detail a vložit poznámku.

Nakonec je potřeba vše zakótovat a popsat, aby bylo možné objekt vyrobit. [22]

Video ukázka tvorby v softwaru FreeCAD je dostupná v Externích odkazech na konci práce.

## 4 KiCAD

KiCAD slouží pro projektování desek plošných spojů DPS neboli PCB (Printed circuit board).

Výstupem jsou data ve formátu Gerber, což je svobodný vektorový formát ASCII pro tvorbu PCB. Gerber obsahuje informace o PCB jako je měděná vrstva, pájení a pozice vrtaných otvorů. [23]

### 4.1 Export

Exportovat data lze z *Návrhu PCB*. Po vytvoření a dokončení celého projektu je možné zvolit typ souboru, který požadujeme jako výstup celé desky (Gerber, Postscript, SVG, DXF, HPGL, PDF). Pro vrtání se jedná o typ souboru Excellon. Cesta pro export: *Soubor* → *Kreslit*.

### 4.2 Výroba plošných spojů

Plošné spoje se vyrábí z nevodivé sklolaminátové desky, která je většinou z obou stran potažená slabou měděnou vrstvou. Na měděném povrchu se vytvářejí vodivé cesty pro spojení elektronických obvodů.

Při sériové výrobě se na desku přenese návrh součástek a cest v podobě leptu vzdorné pasty. Zbylá nepotřebná měď je pomocí leptání odstraněna a měděný povrch zůstane pouze pod vrstvou nanesené pasty.

V našem případě pro kusovou a prototypovou výrobu je možné plošné spoje vytvořit prostřednictvím frézování. V tom případě Gerber data konvertujeme na G-kód. [24]

### 4.3 Práce v softwaru

Prvním krokem po otevření programu je volba *Vytvořit nový projekt*. Po vybrání úložiště se automaticky vytvoří dva soubory. Jeden slouží jako editor pro tvorbu elektronických schémat. Druhý slouží k navrhování tištěných spojů.

#### 4.3.1 Návrh schématu

Je určen pro návrh elektrického obvodu včetně součástek.

Po zvolení nabídky *Umístění symbolu* v pracovním prostoru se otevře nahraná knihovna v programu se standardizovanými součástkami (přibližně 15 000 součástek, které jsou uvedeny pod anglickým názvem). Vyhledávání je tříděno podle typu součástky, nebo je možné vyhledávání pomocí názvu.

Pokud knihovna neobsahuje hledanou součást, je možné si ji vytvořit v *Editoru součástek* v základním okně. Součástka je tvořena pomocí pouzdra, kontaktů. Vytvořenou součástku je možné uložit do nově vytvořené knihovny, která bude platná lokálně pro vytvářený projekt nebo globálně pro celý software.

Po umístění všech potřebných elektronických součástek na pracovní plochu, je potřeba všechny propojit a k tomu slouží nástroj *Umístit vodič*, který je v pravé části ovládacího panelu.

V následujícím kroku je potřeba všechny součástky očíslovat a k tomu slouží nástroj *Očíslovat schéma*, který se nachází v horní liště nástrojů. Tato část je důležitá pro následující krok a tím je přiřazení pouzdra, které najdeme v hlavní nabídce Nástroje → Přiřazení pouzder.

Přiřazení pouzder slouží pro tvorbu a vyobrazení reálné součástky ve 3D náhledu celého návrhu. Poté je potřeba soubor uložit a tím se okno pouzder uzavře.

Dále využijeme v horním ovládacím panelu nástroj *Generovat netlist*, který uloží vybraná pouzdra a schéma. [25]

Pro rychlejší návrh schématu je vhodné používat zkratky nástrojů.

"w" - umístit vodič

"c" - kopírování komponentu (stejně i pro Návrh PCB)

"v" - změna hodnoty komponentu

"r" - rotace komponentu (stejně i pro Návrh PCB)

"m" - posunutí komponentu (stejně i pro Návrh PCB)

### 4.3.2 Návrh PCB

Při otevření návrhu PCB je potřeba jako první *Načíst netlist*, který je uložen ve vytvořeném projektu. Z netlistu se načtou součástky a schéma návrhu.

Zobrazené součástky je potřeba umístit do prostoru podle praktického rozvržení plánované desky.

V pravé části se nachází ovládací panel se *Správce vrstev*, který rozděluje jednotlivé typy čar. Základní potřebnou vrstvou je *F.Cu*, která vyznačuje přední vrstvu mědi (při tvorbě oboustranné PCB použijeme i *B.Cu*, která tvoří zadní stranu) a jsou z ní tvořeny vodivé spoje. Pro ohraničení tvaru desky je potřeba zaškrtnout vrstvu *Edge.Cuts*.

Po dokončení Návrhu PCB je možné v ovládacím panelu *Pohled* → 3D prohlížeč (Alt+3), otevřít reálný náhled vytvořené desky se součástkami a tištěnými spoji. [26]

Video ukázka tvorby v softwaru KiCAD je dostupná v Externích odkazech na konci práce.

## 5 Inkscape

Jedná se o svobodný editor založený na vektorové grafice.

Při použití potřebných rozšíření, která jsou již základně nainstalována, nebo je možné je doinstalovat dodatečně, můžeme program také využít jako generátor G-kódu. Využívá se zejména pro 2D grafiku jako je laser nebo gravírování. Návrhy je možné tvořit přímo v prostředí programu nebo je importovat.

### 5.1 Postup vytvoření G-kódu

1. Při vytvoření nebo importování návrhu zvolíme velikost objektu v milimetrech v horní části pracovního prostoru.
2. Dále otevřeme nabídku *Extensios* → *Gcodetools* → *Tools library*.
3. Pro většinu použití v nabídce *Tools library* zvolíme možnost *cylinder* a potvrdíme.
4. Vytvořenou zelenou tabulku umístíme vedle návrhu, aby jej nepřekrývala a upravíme parametry podle potřeby (průměr nástroje, otáčky, posuv, hloubku řezu).
5. V záložce *Gcodetools* otevřeme *Orientation points*. V poli *Z surface* → 0, *Z depth* 0,1 a jednotky zvolíme milimetry [mm].
6. Otevřeme nastavení cesty G-kódu *Extensios* → *Gcodetools* → *Path to Gcode*. V otevřeném okně a nabídce *Preferences* zvolíme název souboru s koncovkou *název.ngc*. Dále vybereme místo uložení souboru, bezpečnou výšku nástroje nad plochou, jednotky → mm a *Post-processor* → none. V nabídce *Path to Gcode* nastavíme *Cutting order* → *Subpath by subpath*.
7. Po potvrzení nastavení se obrázek změní v jednotlivé dráhy nástroje a uloží se jako soubor do zvolené složky s příponou .ngc obsahující celý G-kód.

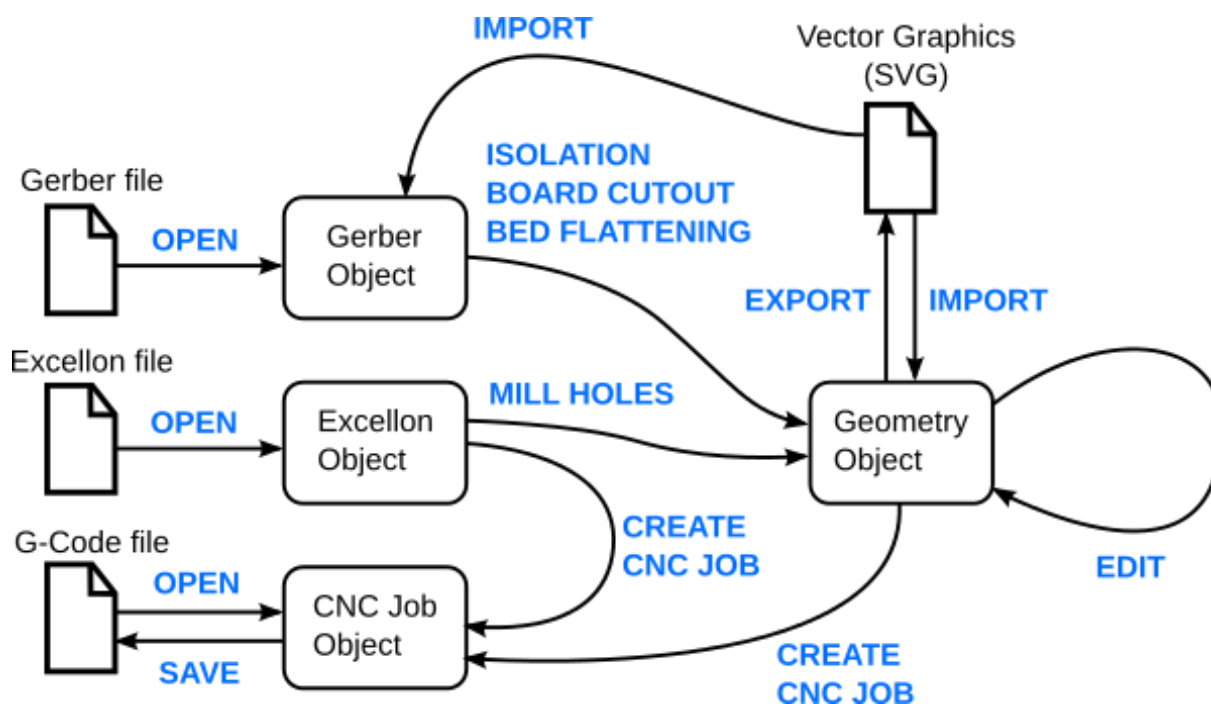
Tato kapitola byla inspirována z internetových zdrojů dostupných v Seznamu literatury: [27, 28]

## 6 FlatCAM

Jedná se o program pro přípravu na CNC zařízení pro tvorbu PCB. Dokáže importovat Gerber soubory, které jsou vytvořeny v PCB Cadu (v našem případě KiCad) a dále z nich vytvořit G-kód. [29]

### Podporované typy souborů pro import:

- Gerber - typické nadefinování měděné vrstvy na PCB
- Excellon - obsahuje specifikaci vrtáků, jejich velikost a souřadnice
- G-Code - instrukce pro CNC zařízení pro frézování nebo vrtání
- SVG - vektorová grafika



Obrázek 2: Diagram zobrazující možné úkony se soubory [30]

Na začátku vždy založíme nový projekt (CTRL + N) nebo se po načtení externího souboru automaticky vytvoří nový projekt sám.

### 6.1 Frézování PCB

Otevřeme potřebný Gerber soubor s informacemi o měděné vrstvě *F.Cu* nebo *B.Fu*. Soubor se automaticky analyzuje, zpracuje a vykreslí geometrii.

V záložce *Selected* zapíšeme *Tool diam* (průměr nástroje), který využijeme pro izolované frézování a klikneme na *Generate Isolation Geometry*. To vytvoří nový geometrický objekt s koncovkou *.iso* → po přiblížení objektu je vidět nově vytvořená geometrie okolo kontury.

**V nově vytvořeném souboru je potřeba vyplnit technologické údaje:**

Cut Z - jedná se o hloubku záběru v ose Z; obvykle hodnota -0,002 mm nebo -0,05 mm

Travel Z - bezpečnostní vzdálenost od desky mimo frézovací cyklus

Feedrate - řezná rychlost udávaná v jednotkách podle nastavení aktivního projektu

Tool diam - průměr nástroje; je potřeba zadat stejný rozměr nástroje jako při tvorbě kontury

Po vyplnění dialogového okna klikneme na *Generate CNC Job object* a tím se vytvoří nový soubor uložený do projektu.

Modře zvýrazněné dráhy jsou výřezy mědi a žlutě zvýrazněné dráhy jsou přejezdy nástroje mimo řez. V záložce *Project* se zobrazuje strom vytvořených souborů, které je možné pomocí pravého tlačítka myši a možnosti *Enable plot/Disable plot* zobrazovat a skrýt. To je užitečné pro větší přehlednost v případě více operací.

Před závěrem je možné přímo přidat pomocné funkce na začátek kódu. Úpravy je možné provádět i po uložení souboru při otevření v poznámkovém bloku. [31]

## **6.2 Vrtání děr pro kontakty součástek**

Načtení souboru typu excellon. Automaticky se načtou souřadnice děr.

V dialogovém okně se vytvoří tabulka s rozměry děr a jim přidělený vrták. Je možné ke každému vrtáku vytvořit samostatný G-kód nebo zvolit jeden průměr vrtáku vyplněný v okně *Drill Tool dia* a vytvořit všechny díry stejné.

## Údaje pro vyplnění:

Cut Z - hloubka vrtání

Travel Z - výška nástroje od desky mimo záběr v materiálu

End move Z - vzdálenost nástroje od desky po poslední operaci

Feedrate Z - rychlost posuvu nástroje

Spindle speed - otáčky vřetena za minutu

Po vyplnění všech potřebných parametrů vygenerujeme G-kód → *Create Drills GCode*, který následovně uložíme do zařízení a také se vytvoří nová složka v záložce *Project* → *CNC Job*. Stejně jako u frézování se zobrazí béžovou barvou dráha nástroje. [32]

## 6.3 Frézování měděné vrstvy

Frézování nefunkčních ploch je potřebné pro snížení rizika zkratu způsobeného prachem, rzi apod., nebo při používání ve vysokofrekvenčních obvodech. Umožňuje odebrání měděné vrstvy neuvedené v Gerber souboru.

### Postup:

Otevřeme Gerber soubor jako v předešlých případech.

V záložce *Selected* rozklikneme okno s popisem *Non-copper regions*. Zadáme *Boundary Margin* (označení velikosti ohraničení), pokud požadujeme zaoblené okraje ohraničení, zaškrtneme *Rounded Geo*. Dále klikneme na *Generate Geo* a tím se nám vytvoří nový geometrický objekt obsahující ohraničení objektu okolo Gerber se zadaným okrajem. Poté se Gerber objekt odečte od ohraničené části, což vytvoří geometrický objekt tvořený polygony na místě neobsahující měď.

Po vytvoření okraje a nové geometrie, vyplníme dialogové okno s parametry pro nástroj jako u předchozích postupů frézování.

Pro volbu trasy nástroje zvolíme v záložce *Selected* nástroj *Paint Tool*, který nám umožní podrobnější nastavení. Důležité je, aby v položce *Object* byl zvolený soubor s koncovkou



*\_noncopper*, který byl v předchozím kroku vytvořen. Dále zadáme *Tool Dia* (průměr nástroje), který použijeme na odfrézování přebytečné mědi. Dále zvolíme *Overlap Rate* (překrytí jednotlivých drah nástroje) v [%] pro zajištění odebrání veškeré mědi i při frézování v 90° drahách.

Vyplnění parametru *Margin* je volitelné. Určuje pouze vzdálenost nástroje od mnohoúhelníků tvořících vodivé cesty v případě frézování větším průměrem nástroje.

Volba metody frézování je zcela individuální na tvořené desce, a proto si ji můžeme zvolit podle potřeby. Metoda *Standard* pracuje na principu zmenšování mnohoúhelníku o průměr nástroje, dokud nezůstane žádná oblast. V metodě *Seed-based* se dráha počítá uvnitř cílového objektu, která začíná bodem a zvětšuje se po kružnicích (přirůstající rozteč o průměr nástroje a překrytí), do té doby než se neprotne s okrajem mnohoúhelníku. Poslední metodou je *Straight lines*, která vytváří rovnoběžné dráhy.

Po vyplnění všech potřebných údajů v záložce *Paint Tool* vytvoříme geometrii drah → Create Paint Geometry. To vytvoří červeně zvýrazněné dráhy a vrátí nás zpět do záložky *Selected*, kde jako v předchozích postupech vytvoříme G-kód (Generate CNC Job Object). [33]

## 6.4 Oboustranné frézování PCB

Při tvorbě některých PCB je potřeba vytvořit vodivé cesty na obou stranách desky. K tomu je potřeba využít frézování obou stran.

Tento způsob výroby je podrobně popsán v kapitole *2-side PCB*, která je volně dostupná na webových stránkách softwaru viz odkaz v Seznamu literatury. [34]

## 6.5 Výřez desky

Tento nástroj se nejčastěji používá při vyřezávání PCB z většího formátu sklolaminátové desky.

### Postup:

Je možnost automatického vytvoření výřezu, který zvolí program, nadefinovat jej manuálně nebo načtením Gerber souboru.

Tato práce a postup bude popisovat výřez desky pomocí načtení Gerber souboru. Postup ostatních možností lze vyhledat na webových stránkách zdroje této kapitoly.

V prvním kroku je potřeba načíst Gerber soubor s koncovkou *Edge\_cuts.gml*, který definuje okraje desky. Tento soubor jsme vygenerovali například z programu KiCad a otevřeli jej.

V otevřené záložce *Selected* zvolíme nástroj pro vytvoření kontury desky *Cutout Tool*.

Ve výběru objektu pro výřez je potřeba zvolit již načtený soubor a zbylé parametry vyplnit podobně jako u předchozích operací.

Při vytvoření geometrie kontury je možné zvolit místa po obvodu, kde nebude materiál odebrán a bude stále součástí celého archu. V parametrech se navíc objevuje hodnota pro *Gap size* (velikost mezery), která určuje délku nefrézované části. Je zde 7 možností pro umístění mezer: *None* - žádná mezera; *LR* - vpravo, vlevo; *TB* - nahoře, dole; *4* - na každé straně jedna; *2LR* - dvě vlevo, dvě vpravo; *2TB* - dvě nahoře, dvě dole; *8* - dvě na každé straně. Pro vytvoření geometrie zvolíme *Generate Freeform Geometry*. [35]

Na závěr otevřeme nově vytvořenou geometrii a stejně jako u předchozích postupů vytvoříme G-kód.

Video ukázka tvorby v softwaru OpenSCAD je dostupná v Externích odkazech na konci práce.

## II Praktická část

### 7 Návod postupu návrhu v jednotlivých softwarech

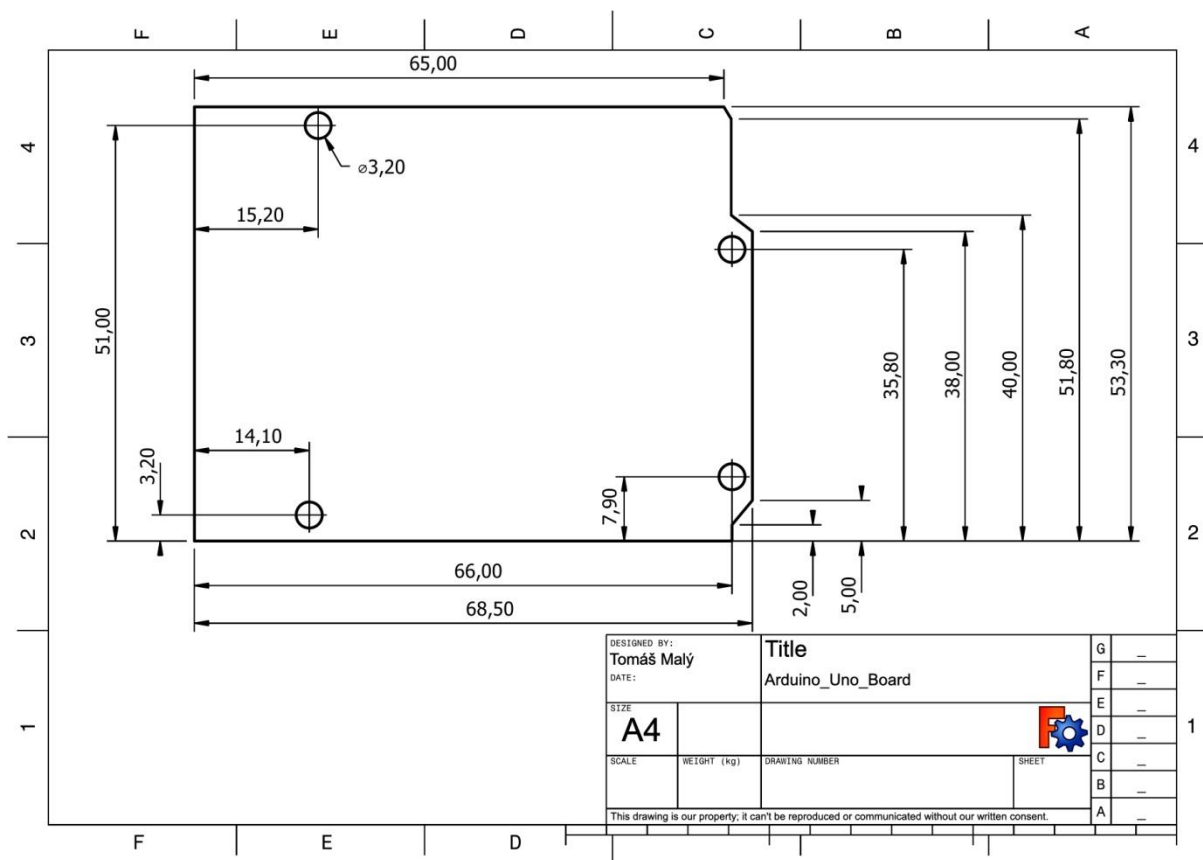
#### 7.1 Softwary OpenSCAD a FreeCAD

Pro ukázkou a porovnání těchto dvou softwarů byl vytvořen návrh krabičky pro základní desku Arduino Uno. Na tomto návrhu krabičky jsou založeny ukázky a porovnání.

Pro vytvoření návrhu krabičky a desky plošných spojů bylo zapotřebí změřit originální desku Arduina a vytvořit její model. Podle modelu desky bylo dále možné vytvořit návrh krabičky. Níže uvedené výkresy jsou vytvořeny v softwaru FreeCAD dle odměřených reálných hodnot desky a podle nich je dále navržena krabička.

U každého z výkresů je znázorněn krátký postup vytvoření modelu dané součástky v softwaru OpenSCAD a FreeCAD.

##### 7.1.1 Základní deska Arduina



Obrázek 3: Výkres desky [Tomáš Malý]

## Postup vytvoření desky v OpenSCAD

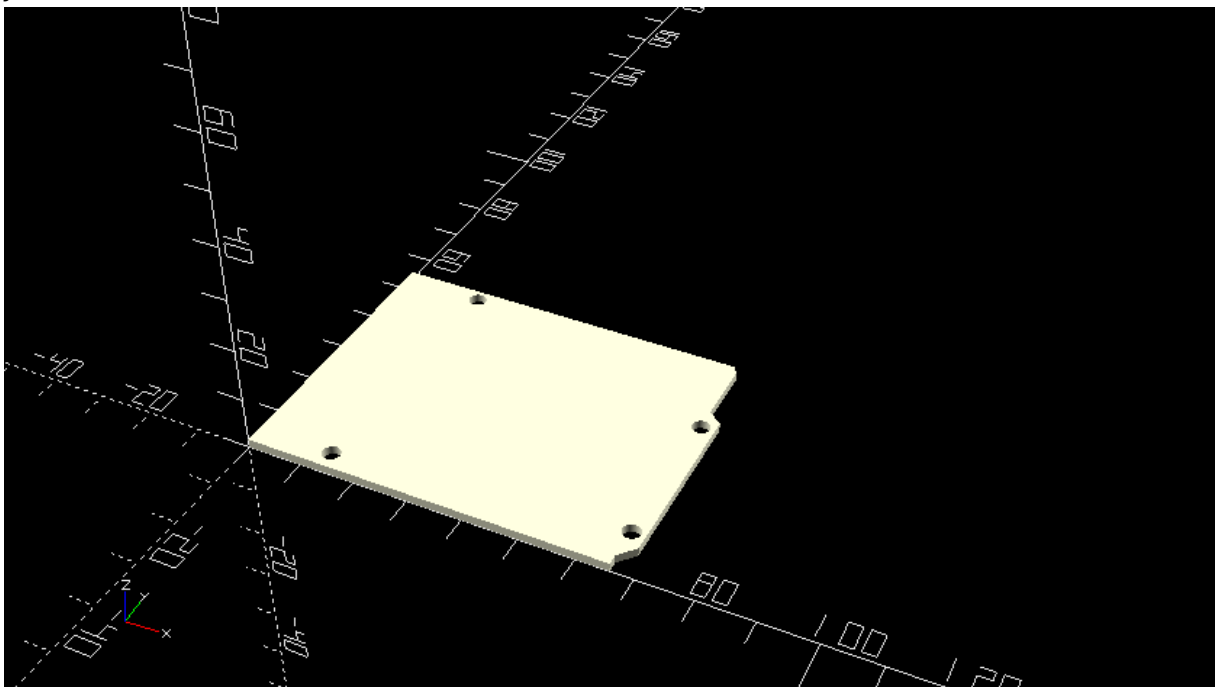
Pro vytvoření desky využijeme 2D modelovací prvek `polygon` (mnohoúhelník). Ten je tvořen vrcholy desky podle výkresu ve tvaru `[x,y]` a cestou, kterou jsou body spojovány.

Od vytvořeného `polygonu` pomocí příkazu `difference` odečteme čtyři kružnice, které budou sloužit jako aretační díry.

Na závěr je do zdrojového kódu použit příkaz `linear_extrude`, který vytáhne do prostoru vytvořený 2D náčrt o požadovanou tloušťku (v našem případě 1,6 mm).

### Ukázka zdrojového kódu tvořící model desky

```
$fn=500;
linear_extrude(height=1.6){
  difference(){
    polygon(points=[[0,0],[66,0],[66,2],[68.5,5],[68.5,38],[66,40],
                  [66,51.8],[65,53.3],[0,53.3]],
           paths=[[0,1,2,3,4,5,6,7,8]]);
    translate([14.1,3.2,0]) circle(r=1.6);
    translate([66,7.9,0]) circle(r=1.6);
    translate([66,35.8,0]) circle(r=1.6);
    translate([15.2,51,0]) circle(r=1.6);
  }
}
```



Obrázek 4: Model desky OpenSCAD [Tomáš Malý]

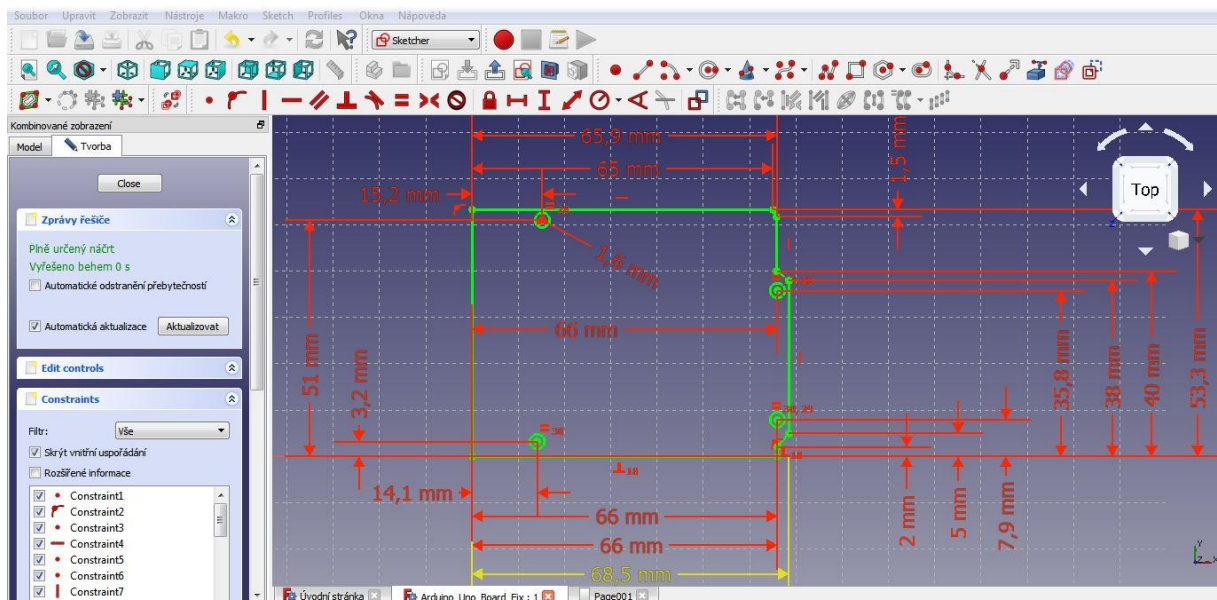
## Postup vytvoření desky v FreeCAD

V prvním kroku jsme vytvořili nový prázdný dokument. Dále jsme pro vytvoření modelu zvolili modelovací prostředí *Part Design*, ve kterém se zpřístupní lišta s nástroji pro tvorbu objektů.

V následujícím kroku jsme vytvořili novou skicu a její umístění jsme zvolili na základní rovině XY.

Po vytvoření skici se otevře nová lišta nástrojů, která slouží k vytváření geometrických útvarů a jejich zakótování. Celý obvod jsme přibližně narýsovali pomocí lomené čáry. Všechny rozměry, které byly předem odměřeny, jsme zakótovali od počátku souřadného systému. Poslední byly vytvořeny aretační otvory pomocí kružnice.

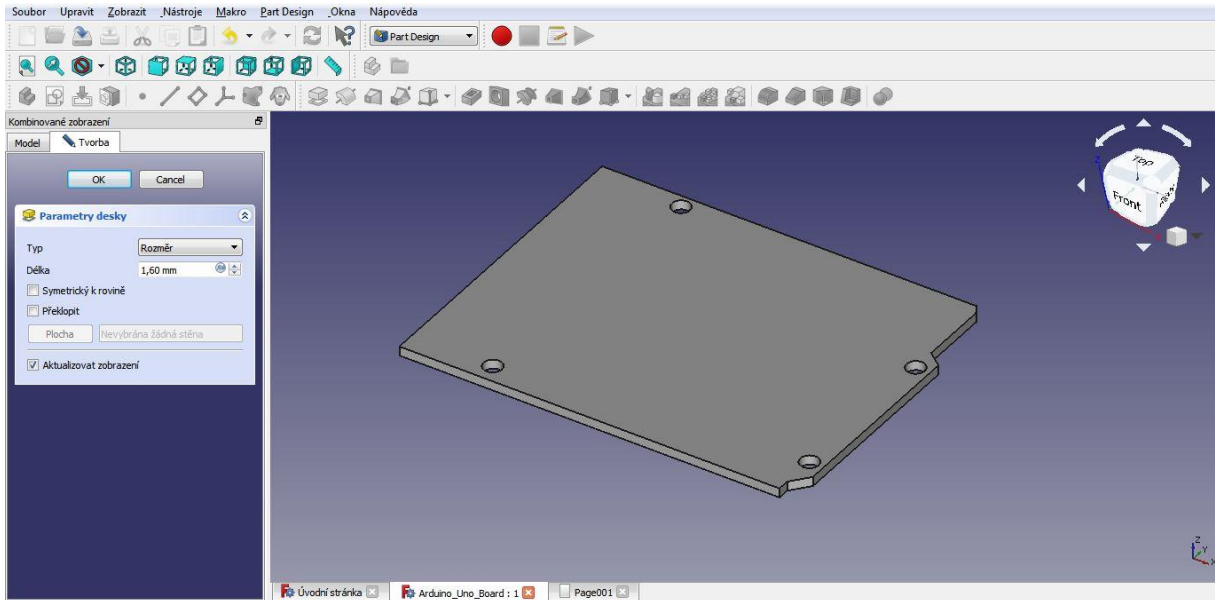
Pro kontrolu správnosti zakótování slouží tzv. řešič, který se nachází v levé části dialogového okna a průběžně ukazuje a řeší počet stupňů volnosti tvořené skici. Pro znázornění chybějící kóty nebo vazby je možné kliknout na počet stupňů volnosti a ve skice se část zobrazí jako tmavě zelená. Pro kontrolu jaká část není ještě určena, také stačí myší uchytit libovolnou část skici a pokusit se s ní posunout. Pokud je skica hotová, změní svoji barvu na světle zelenou a *Řešič* vypíše *Plně určený náčrt*, viz obrázek č. 5.



Obrázek 5: Skica desky FreeCAD [Tomáš Malý]

Poslední krok pro vytvoření 3D modelu je vytáhnout 2D skicu do prostoru podél osy Z.

K tomu slouží nástroj *Přidat vysunutím vybranou skicu*. V levém dialogovém okně se otevře nabídka s parametry vytahované desky, viz obrázek č. 6. Je možné si zvolit typ, rozměr a směr vysunutí. Po vybrání všech potřebných parametrů se dialogové okno potvrdí a objekt je vytvořen.

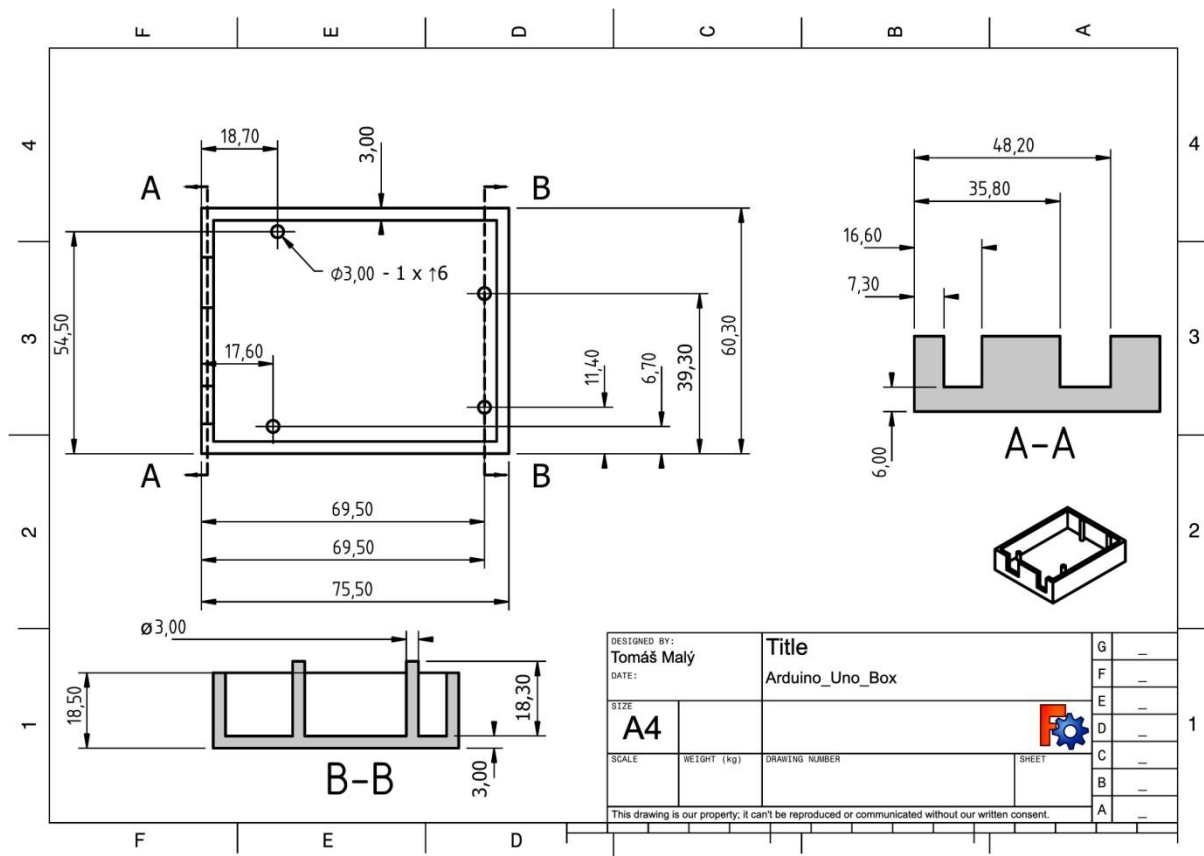


Obrázek 6: Model desky FreeCAD [Tomáš Malý]

### 7.1.2 Ochranná krabička pro Arduino

Ochrannou krabičku jsme navrhli tak, aby vyhovovala praktickému použití Arduino. Krabička by měla omezit přístup prachu a nečistot, zabránit poškození součástek na základní desce a připojovacích konektorů. Při manipulaci s deskou zabránit porušení pracovního stolu ostrými hroty zespodu PCB.

Deska je také zasunuta přes čtyři kolíky, které jí zabrání v pohybu po krabičce. Tři z těchto kolíků také slouží k aretaci ochranného víka.



Obrázek 7: Výkres krabičky [Tomáš Malý]

## Postup vytvoření krabičky v OpenSCAD

Základní tvar ochranné krabičky je tvořen kvádrem, ze kterého jsou postupně odebrány tři kvádry.

První vytváří prostor pro základní desku. Jeho rozměry a posunutí jsou zvoleny tak, aby vznikla krabička o tloušťce stěny 3 mm.

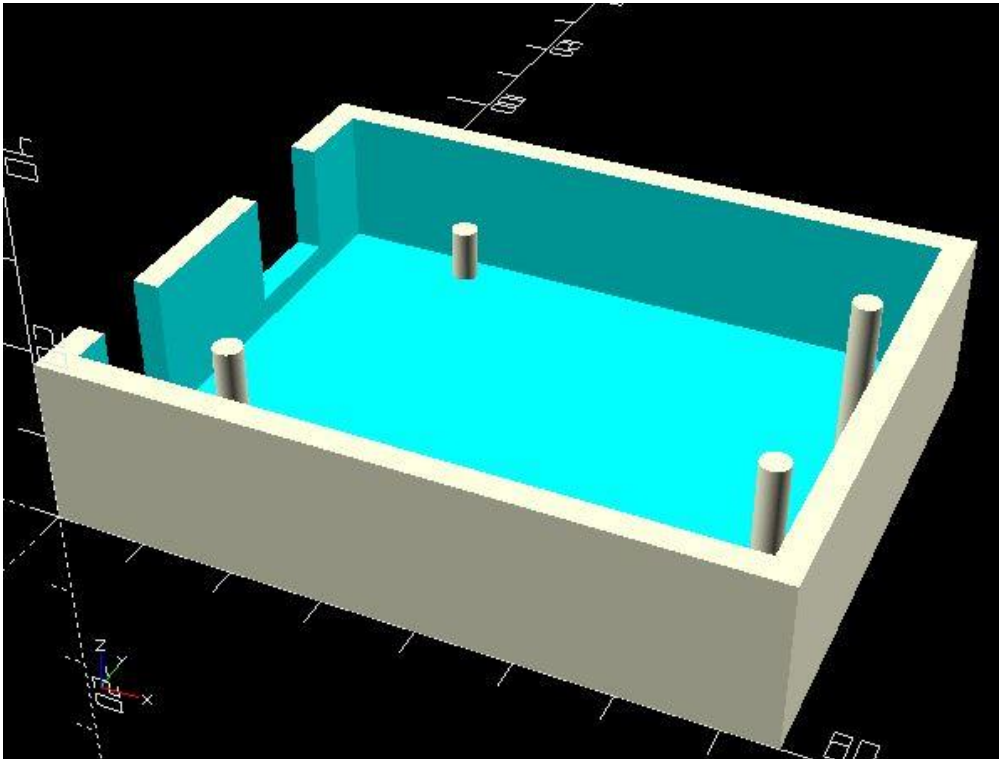
Dalšími dvěma odebranými kvádry vznikly otvory pro napájení a vstup.

Na závěr byly vymodelovány, pomocí prvku válec, čtyři aretační kolíčky, které slouží k umístění desky do vnitřního prostoru a zamezení jejího posunu.

```
$fn=100;
```

```
difference(){
    cube([75.5,60.3,18.5]);
    translate([3,3,3]) cube([69.5,54.3,15.5]);
    translate([0,7.3,6]) cube([3,9.3,12.5]);
    translate([0,35.8,6]) cube([3,12.4,12.5]);
}
```

```
translate([17.6,6.7,3]) cylinder(h=18.5,r=1.5);
translate([69.5,11.4,3]) cylinder(h=18.5,r=1.5);
translate([69.5,39.3,3]) cylinder(h=18.5,r=1.5);
translate([18.7,54.5,3]) cylinder(h=6,r=1.5);
```



Obrázek 8: Model krabičky OpenSCAD [Tomáš Malý]

### Postup vytvoření krabičky v FreeCAD

Vytvoření souboru, skici a modelu probíhalo podobně jako u Desky Arduina.

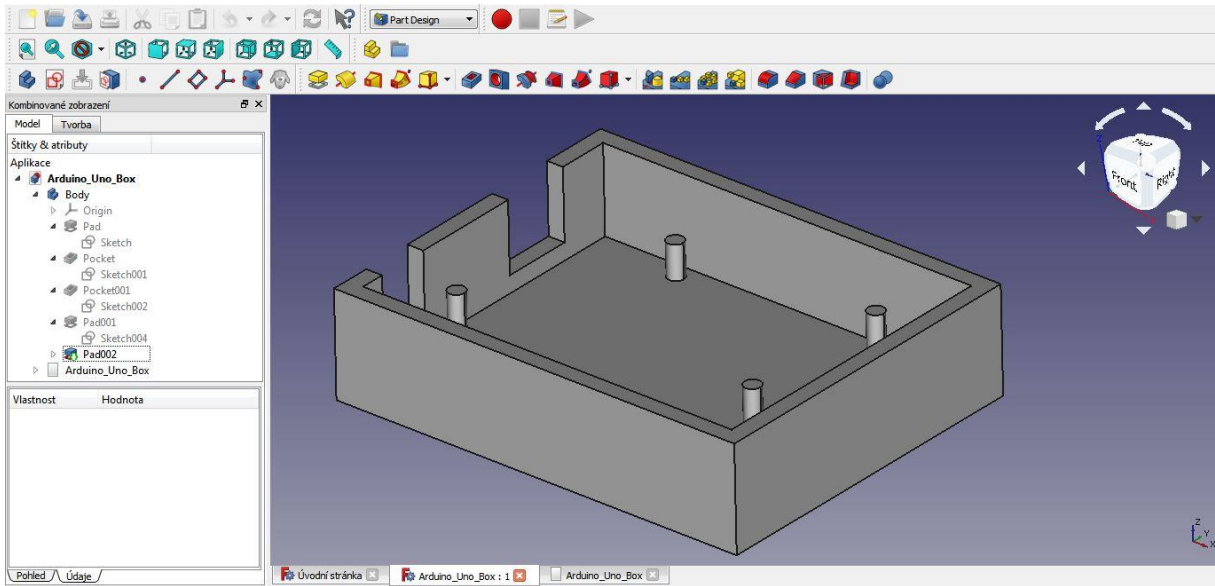
Jako první jsme vytvořili základní kvádr pomocí skici a následně ho vysunuli o výšku krabičky.

V dalším kroku jsme narýsovali na horní ploše kvádrů skicu ve tvaru obdélníku tak, aby vznikly stěny široké 3 mm. Poté jsme použili nástroj *Vytvořit kapsu pomocí vybraného náčrtu*, kde jsme vyplnili potřebné parametry podobně jako u vysunutí.

Následovně jsme vytvořili skicu na levém boku krabičky, která obsahuje dva obdélníky tvořící otvory pro vstup a výstup. Tuto vytvořenou skicu jsme odebrali skrz stěnu pro vytvoření otvorů.

Na závěr jsme vytvořili skicu na dně krabičky, kterou tvoří čtyři aretační kolíky. Poté jsme je vysunuli o potřebnou vzdálenost.

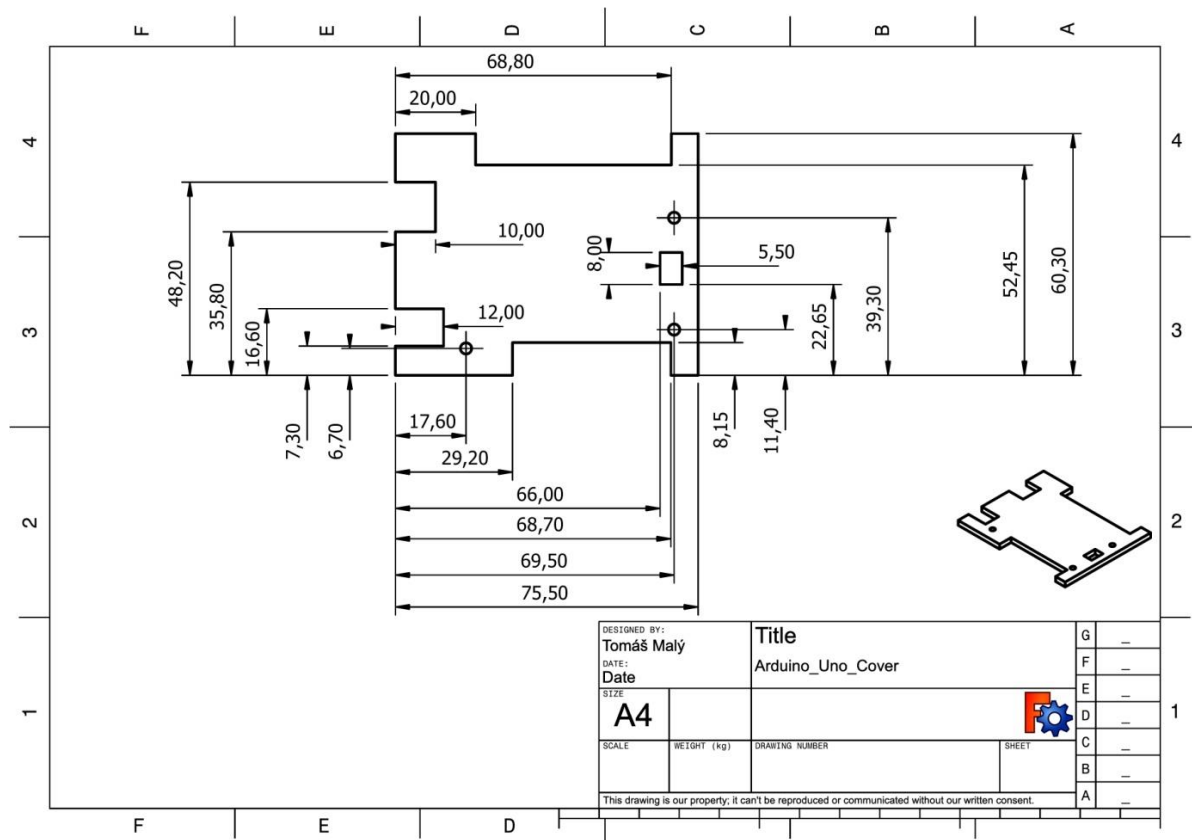




Obrázek 9: Model krabičky FreeCAD [Tomáš Malý]

### 7.1.3 Víko ochranné krabičky

Víko krabičky jsme navrhli tak, aby byl omezen přístup prachu a nečistot k elektronickým součástkám. Výřezy jsou umístěny přesně nad zapojovací konektory a jejich velikost by měla umožnit snadné zapojení a manipulaci s kabely.



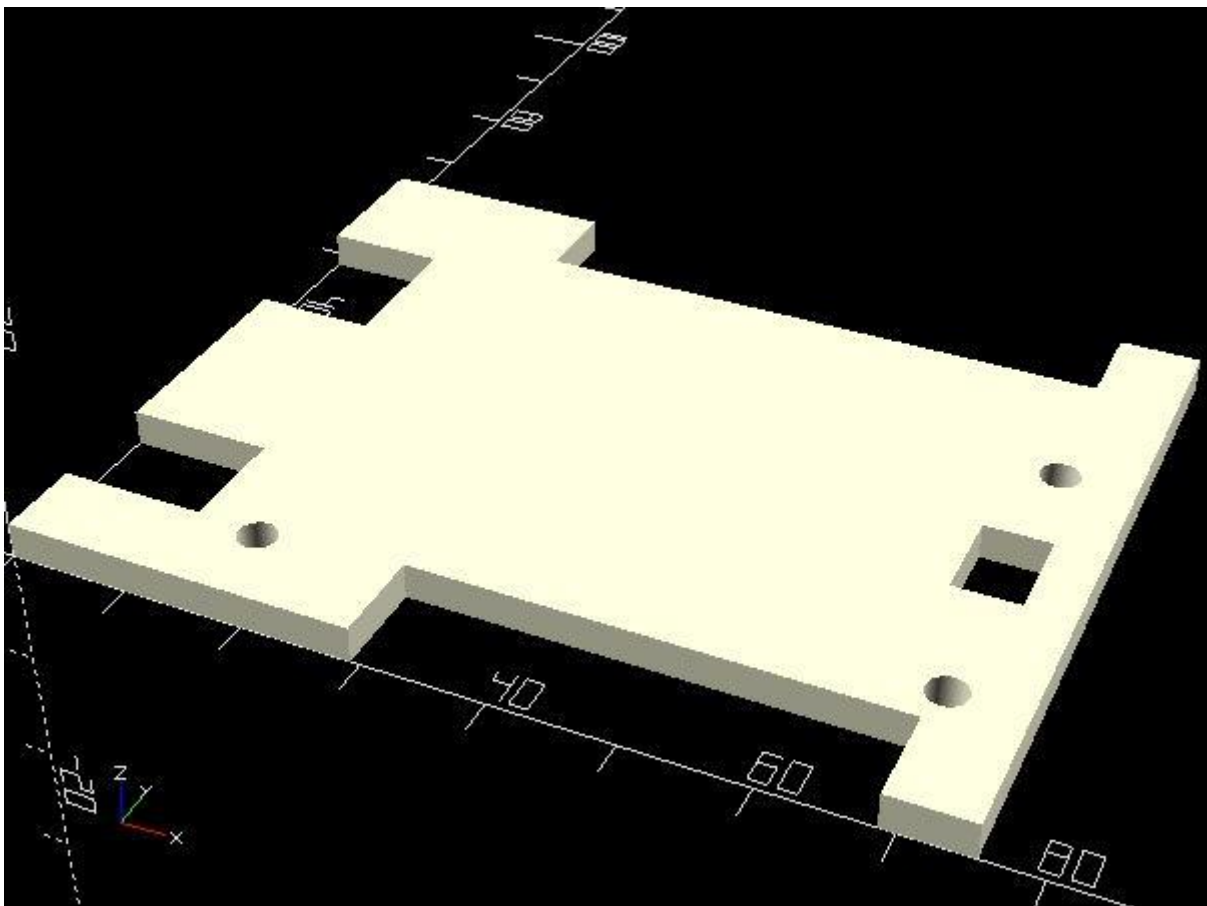
Obrázek 10: Víko krabičky [Tomáš Malý]

## Postup vytvoření víka v OpenSCAD

Při vytváření ochranného víka v OpenSCAD jsme zvolili stejnou metodu jako u základní desky. V těchto případech, kde je obvod tělesa členitý s různými výřezy, je nejjednodušší celé těleso vytvořit pomocí mnohoúhelníku, který umožňuje popis celého obvodu pomocí bodů.

Po zadání mnohoúhelníku byly dále pomocí příkazu *difference* vytvořeny tři aretační díry pro kolíky, které obsahuje krabice a jeden obdélník sloužící k zapojení konektoru.

Na závěr jsme celý 2D objekt vysunuli pomocí příkazu *linear\_extrude* do prostoru.



Obrázek 11: Model víka - OpenSCAD [Tomáš Malý]

### Ukázka programu popisující víko

```
$fn=100;  
linear_extrude(height=3){  
  difference(){  
    polygon(points=[[0,0],[29.2,0],[29.2,8.15],[68.7,8.15],[68.7,0],[75.5,0],  
      [75.5,60.3],[68.8,60.3],[68.8,52.45],[20,52.45],[20,60.3],[0,60.3],  
      [0,48.2],[10,48.2],[10,35.8],[0,35.8],[0,16.6],[12,16.6],[12,7.3],[0,7.3]],  
      paths=[[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19]]);  
  }
```

```

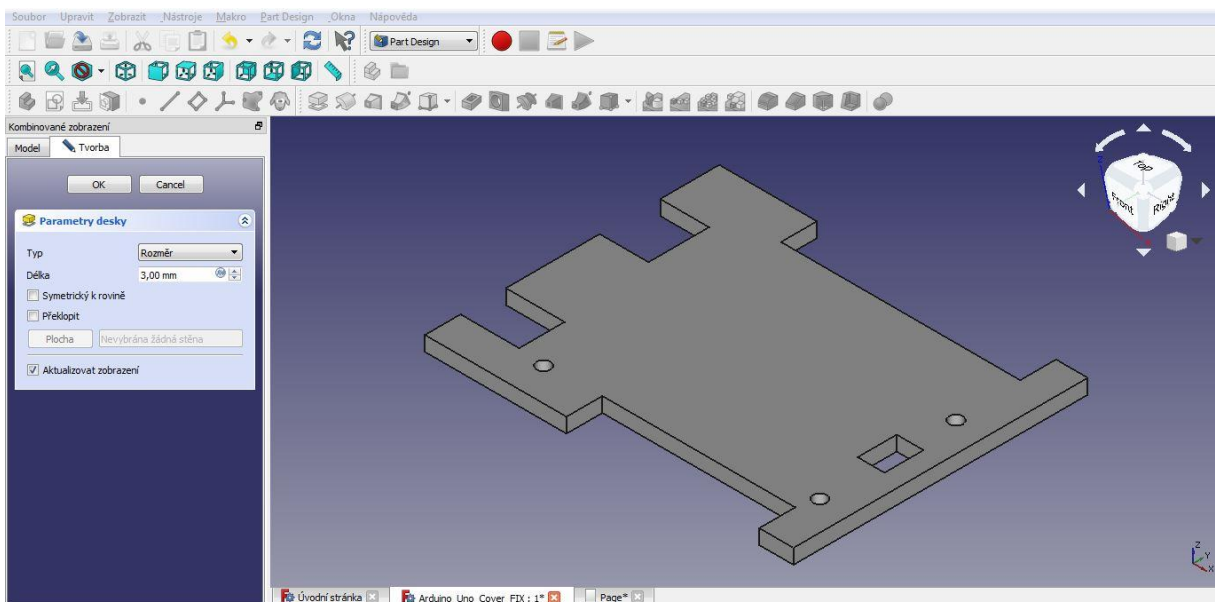
translate([17.6,6.7]) circle(r=1.6);
translate([69.5,11.4]) circle(r=1.6);
translate([69.5,39.3]) circle(r=1.6);
translate([66,22.65]) square([5.5,8]);
    }
}

```

## Tvorba víka ve FreeCAD

Začátek modelování je stejný jako u předešlých součástí. Ve skice jsme přibližně vytvořili obvod víka pomocí lomené čáry, otvory na kolíky pomocí kružnice a pravý otvor pro konektor pomocí obdélníku.

Následně jsme všechny parametry zakótovali podle výkresu a po ukončení skici jsme vysunuli víko do prostoru o 3 mm, viz obrázek č. 12.



Obrázek 12: Model víka - FreeCAD [Tomáš Malý]

## 7.2 Software KiCAD

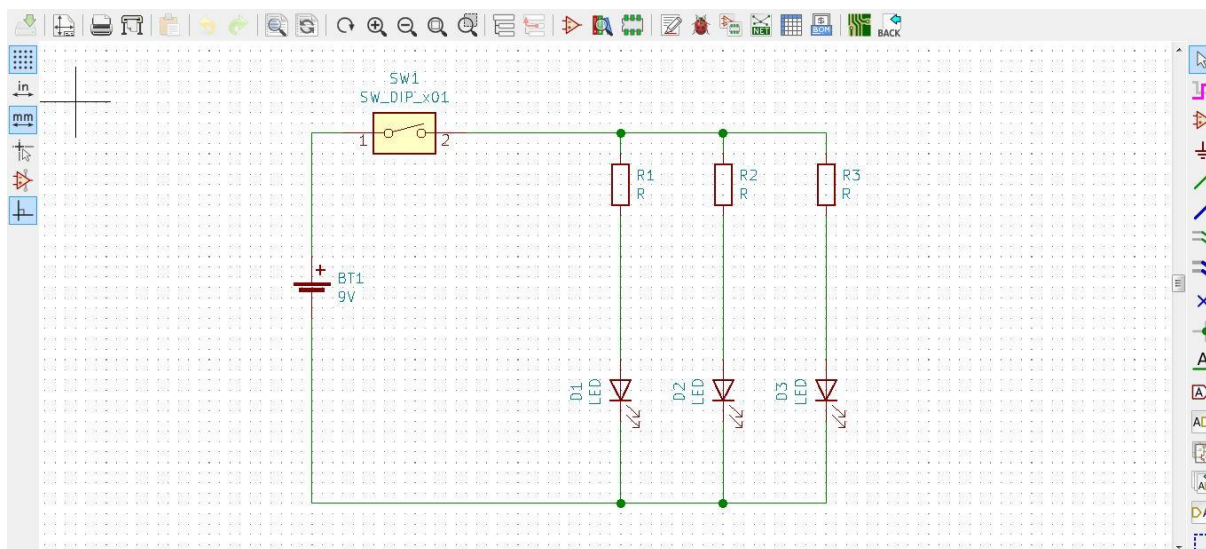
Ukázka obsahuje jednoduché schéma s popisem jednotlivých kroků.

Elektrický obvod se skládá pouze ze zdroje, LED diod, rezistorů a vypínače.

V prvním kroku jsme vytvořili nový projekt a začali s návrhem schématu. Využili jsme knihovnu součástek, ze které jsme vybrali a následně umístili na pracovní plochu všechny potřebné součástky. Součástky jsme rozmístili tak, aby bylo schéma přehledné a mohli jsme je dále propojit vodičem, podle základních pravidel elektroniky.

Po vytvoření schématu jsme využili nástroj *Očíslovat schéma*, které přiřadí každé součástce číslo. Následně jsme každé součástce přiřadili reálné pouzdro pomocí nástroje *Přiřadit pouzdra k součástkám ve schématu*, které je potřebné pro vytvoření PCB (v ukázce jsme například využili pouzdro pro zdroj typu na baterii). Přiřazená pouzdra jsme uložili.

V dalším kroku jsme prostřednictvím nástroje *Generovat netlist* vytvořili seznam součástek s přiřazenými pouzdry a uložili do složky vytvořeného projektu.



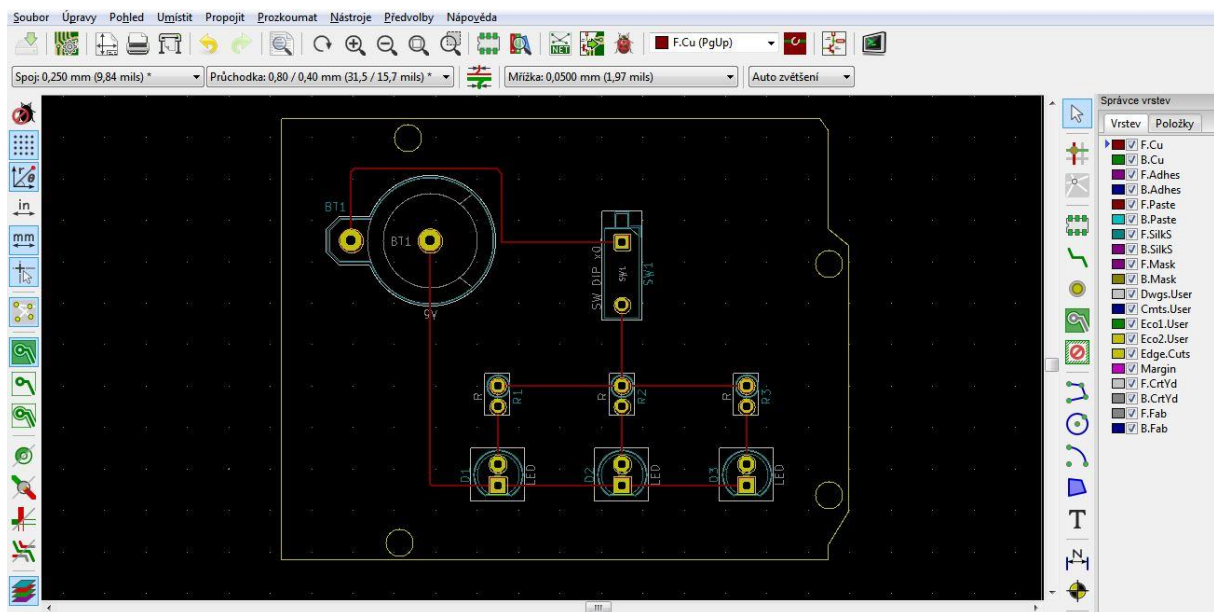
Obrázek 13: Schéma - KiCAD [Tomáš Malý]

Po zhotovení schématu a uložení seznamu součástek jsme otevřeli editor pro návrh PCB. V prvním kroku jsme si vytvořili tvar desky plošných spojů (zvolili jsme pro ukázkou tvar pro Arduino Uno). Pro vytvoření obrysu jsme zaškrtnuli v pravém poli vrstvu *Edge.Cuts* a pomocí lomené čáry jej vytvořili (popřípadě kružnice nebo oblouku).

Ve druhém kroku jsme pomocí nástroje *Načíst netlist* nahráli seznam součástek, které se automaticky zobrazily na pracovní desce.

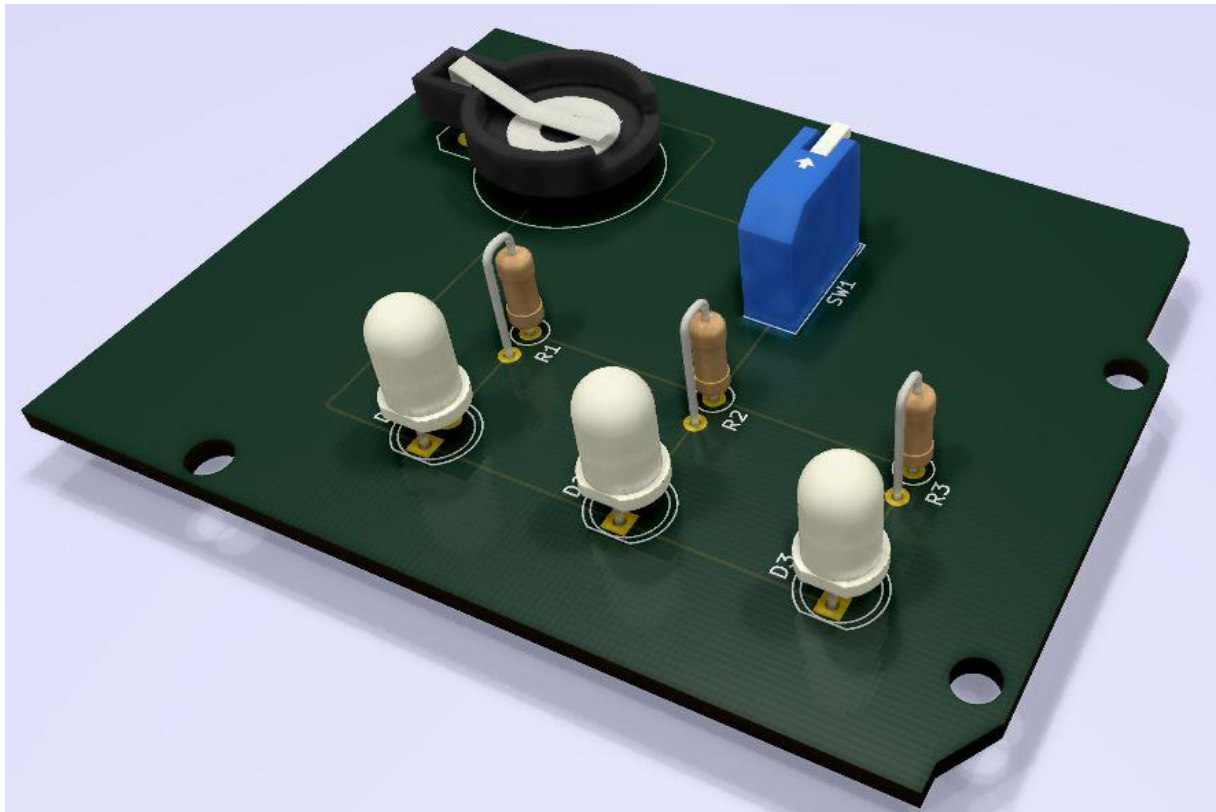
Zobrazené součástky jsou virtuálně propojeny vodičem podle schématu. Součástky jsme úhledně a prakticky poskládali na pracovní desku. Ve výběru vrstev jsme zvolili *F.Cu* pro tvorbu spoje na přední straně desky (v případě tvorby na zadní straně desky je potřeba zvolit *B.Cu*). Po připojení spoje na kontakt se zvýrazní druhý přípojný kontakt, dle vytvořeného schématu.

Pozn.: První a druhý krok lze prohodit podle potřeby. Pokud máme již daný rozměr desky (jako v našem případě) je potřeba nejdříve vytvořit tvar desky a poté umístit součástky. Jestliže máme možnost zvolit libovolné rozměry desky, nejdříve umístíme součástky.



Obrázek 14: PCB [Tomáš Malý]

Po dokončení celé desky plošných spojů jsme si pro reálnou představu otevřeli pomocí **Alt+3** 3D náhled desky, viz obrázek č. 15.



*Obrázek 15: 3D pohled [Tomáš Malý]*

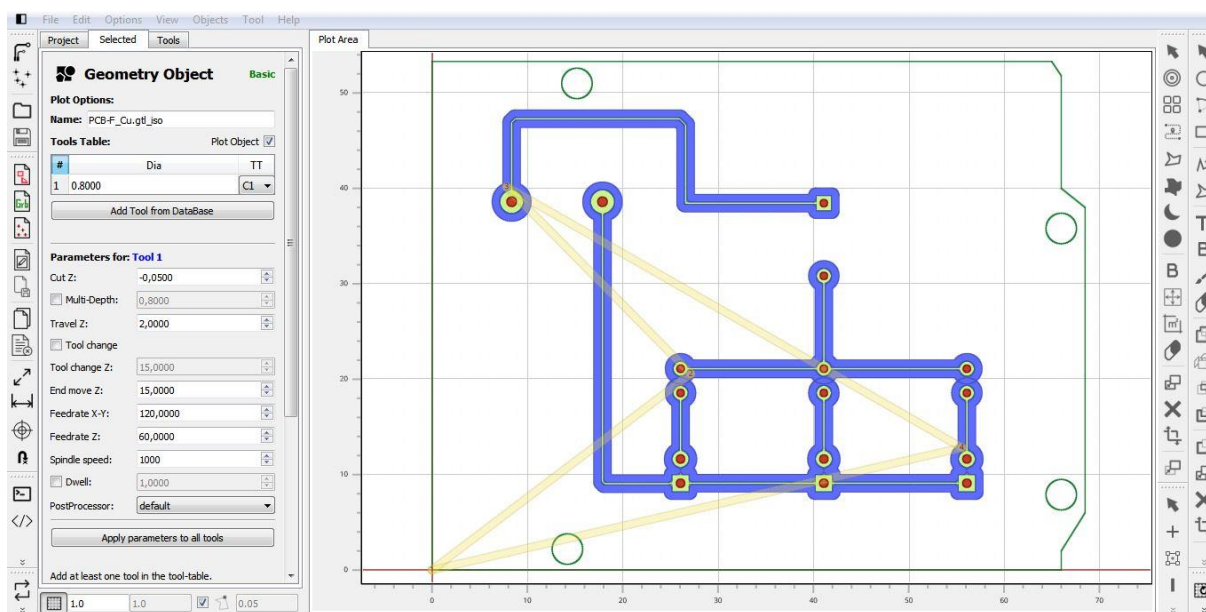
## 7.3 Software FlatCAM

Pro ukázkou tvorby v softwaru FlatCAM jsme použili PCB desku pro Arduino Uno vytvořenou v KiCAD, která je v předchozí kapitole.

### 7.3.1 Frézování kontaktů

V prvním kroku jsme vytvořili nový projekt a otevřeli vygenerovaný Gerber soubor z předešlé tvorby (v našem případě *F\_Cu.gtl*). Tento Gerber soubor slouží k vyfrézování kontaktních ploch, vyvrtání otvorů pro kontakty a odebrání zbylé mědi na desce.

Podle vytvořeného návodu v teoretické části jsme v záložce *Selected* zadali průměr nástroje 0,8 mm a vygenerovali novou geometrii. V dalším kroku jsme vyplnili parametry nástroje, viz obrázek č. 16 níže a poté vygenerovali G-kód, do kterého je možné zadat pomocné funkce a poté jej uložit nebo pouze zobrazit.



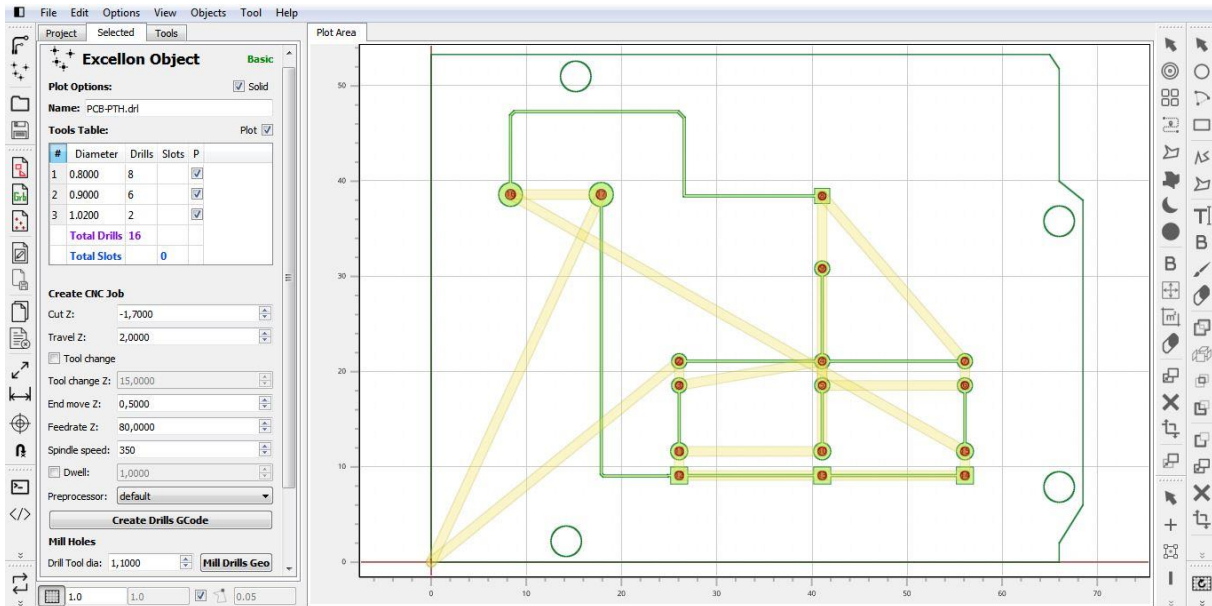
Obrázek 16: Frézování kontaktů [Tomáš Malý]

### 7.3.2 Vrtání děr

Pro lepší přehlednost dalších kroků jsme skryli pomocí pravého tlačítka myši a *Disable Plot* ve stromu operací objekty s předešlého kroku.

V další části jsme načtli Excellon soubor, který obsahuje informace o vrtání otvorů v desce (pozici, průměr, atd.). Automaticky se načtla data a zobrazilo se umístění děr.

V záložce *Selected* je přehled všech vytvořených děr podle průměru a jejich počet. Je možné vytvořit tři G-kódy pro každý průměr, zvolit jeden průměr vrtáku pro všechny díry nebo pokud nám to zařízení umožní, použít výměnu nástroje. V našem případě jsme zvolili jeden průměr vrtáku pro všechny díry. Dále jsme vyplnili zbylé parametry pro vrtání, viz obrázek č. 17 a vygenerovali G-kód.

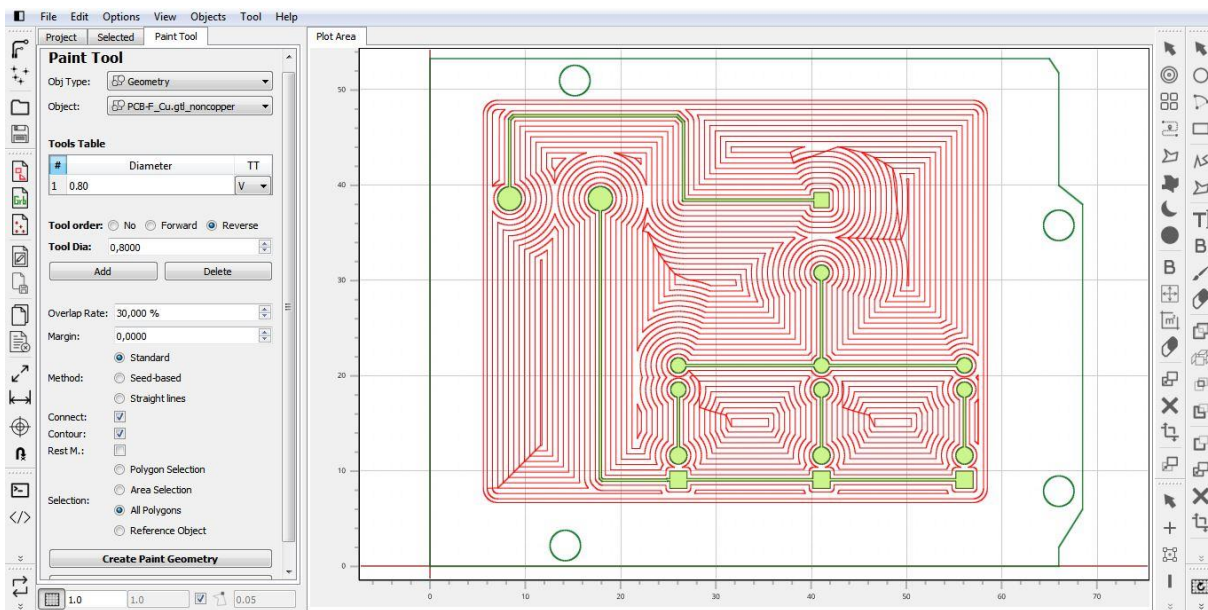


Obrázek 17: Vrtání děr [Tomáš Malý]

### 7.3.3 Frézování zbylé mědi

Otevřeli jsme již načtený gerber soubor, který jsme využili při frézování kontaktů. V záložce *Selected* jsme v části *Non-copper regions* zvolili *Boundary Margin* 1,5 mm (vnější hranici pro odfrézování mědi od vodivé dráhy na všechny strany), zaoblené rohy a vytvořili novou geometrii → *Generate Geo*. Po vytvoření geometrie vnějšího ohraničení jsme dále vyplnili parametry nástroje stejně jako u předešlého frézování a otevřeli nástroj *Paint Tool* pro výběr dráhy, překrytí a průměru nástroje. Na závěr jsme vygenerovali geometrii drah nástroje a vytvořili G-kód.

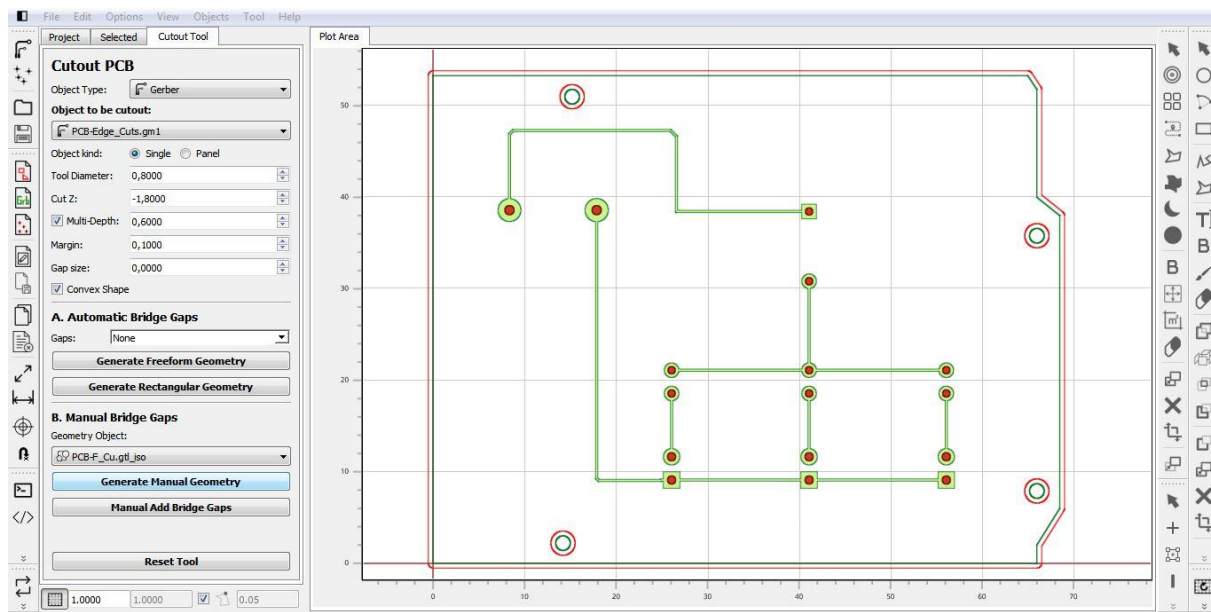




Obrázek 18: Frézování zbylé mědi [Tomáš Malý]

### 7.3.4 Výřez desky

Po otevření příslušného souboru, který definuje okraje desky, jsme podle návodu vyplnili všechna dialogová okna. V našem případě jsme zvolili průměr nástroje 0,8 mm. Pro usnadnění práce jsme zároveň při vyřezávání desky vytvořili čtyři kruhové díry pro uchycení PCB do krabičky. Tyto díry jsme pro jednotvárnost a urychlení práce zmenšili o průměr zvoleného nástroje, z důvodu frézování díry z vnější strany.



Obrázek 19: Výřez obvodu desky [Tomáš Malý]

## Závěr

Cílem této bakalářské práce bylo prostudovat tvorbu a možnosti open-source programů pro tvorbu a následnou realizaci na 3D tiskárně nebo CNC zařízení. Dále vytvořit manuál popisující jednotlivé programy s postupem práce v nich.

Ve vybraných programech jsem vytvořil manuál postupu práce. Cílovou skupinou lidí, pro které jsem manuál zkompletoval, jsou technicky vzdělaní lidé i úplní laici, kteří se s touto problematikou setkávají poprvé. Dle mého názoru se mi tento cíl podařilo naplnit.

Práce pro mě byla náročná zejména kvůli nedostatku knižních zdrojů. Většina zdrojů, ze kterých jsem čerpal, pochází z anglických pramenů. Odborné výrazy byly mnohdy těžké na překlad, ale díky této zkušenosti jsem se mohl zdokonalit i ve svých jazykových znalostech v anglickém jazyce.

Po prostudování mé práce a externích odkazů se domnívám, že člověk, který nabude těchto vědomostí, je schopen vytvořit vlastní model a vyrobit ho na koncovém zařízení.

Mezi mé další problémy patřilo převádění modelů do formátu obrázku v jednotlivých softwarech. Tento problém je způsoben tím, že tyto open-source produkty nepodporují přímé ukládání ve formátu .jpg, .png nebo je velmi nízká kvalita zobrazení. Proto jsem zvolil jako nejlepší alternativu ukládání snímků obrazovky prostřednictvím *Výstřižků* od Microsoft.

## Seznam literatury

- [1] Středisko praktického vyučování PBS Velká Bíteš, Obecný úvod do problematiky CNC programování, Dostupné z: [https://www.sosbites.cz/images/stories/Pro-studenty/studijni-materialy/VUKOV\\_TEXT - 1.ST.pdf](https://www.sosbites.cz/images/stories/Pro-studenty/studijni-materialy/VUKOV_TEXT_-_1.ST.pdf)
- [2] ŠTULPA, Miloslav. *CNC obráběcí stroje a jejich programování*. Praha: BEN - technická literatura, 2006. ISBN 978-80-7300-207-7
- [3] Ing. Petr Keller, Ph.D., Programování a řízení CNC strojů, Prezentace přednášek - 2.část, Technická univerzita v Liberci - Katedra výrobních systémů, Copyright © 2005.
- [4] Ing. Milan Chudoba, Základy programování a obsluha CNC strojů, Střední průmyslová škola, Jihlava.
- [5] OpenSCAD - About. *OpenSCAD - The Programmers Solid 3D CAD Modeller* [online]. Dostupné z: <https://www.openscad.org/about.html>
- [6] OpenSCAD User Manual - STL Import and Export - Wikibooks [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/STL\\_Import\\_and\\_Export](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/STL_Import_and_Export)
- [7] OpenSCAD Comments - Wikibooks [Online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/General#Comments](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/General#Comments)
- [8] OpenSCAD User Manual - Mathematical Functions - Wikibooks [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/Mathematical\\_Functions](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Mathematical_Functions)
- [9] OpenSCAD User Manual - Values and Data Types. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/General#Values\\_and\\_Data\\_Types](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/General#Values_and_Data_Types)
- [10] OpenSCAD User Manual/2D Primitives - Wikibooks, open books for an open world. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/2D\\_Primitives](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/2D_Primitives)
- [11] OpenSCAD User Manual/Text - Wikibooks, open books for an open world. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/Text](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Text)
- [12] OpenSCAD User Manual/2D to 3D Extrusion - Wikibooks, open books for an open world. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/2D\\_to\\_3D\\_Extrusion](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/2D_to_3D_Extrusion)

- [13] OpenSCAD User Manual/Using the 2d Subsystem. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/Using\\_the\\_2D\\_Subsystem#Rotate\\_Extrude](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Using_the_2D_Subsystem#Rotate_Extrude)
- [14] OpenSCAD User Manual/Primitive Solids - Wikibooks, open books for an open world. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/Primitive\\_Solids](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Primitive_Solids)
- [15] What Is Hexadecimal? (Hexadecimal Definition). *Lifewire: Tech News, Reviews, Help & How-Tos* [online]. Copyright © 2020 [cit. 15.4.2020]. Dostupné z: <https://www.lifewire.com/what-is-hexadecimal-2625897>
- [16] HTML color code, Alex Dixon, San Francisco, CA[online]. Dostupné z: <https://htmlcolorcodes.com/>
- [17] OpenSCAD User Manual/Transformations - Wikibooks, open books for an open world. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/Transformations](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Transformations)
- [18] OpenSCAD User Manual/CSG Modelling - Wikibooks, open books for an open world. [online]. Dostupné z: [https://en.wikibooks.org/wiki/OpenSCAD\\_User\\_Manual/CSG\\_Modelling](https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/CSG_Modelling)
- [19] OpenOffice.cz, FreeCAD: svobodná alternativa k AutoCADu. *OpenOffice.cz, Oficiální portál kancelářských balíků OpenOffice a LibreOffice* [online]. Copyright © 2015 [cit. 04.02.2020]. Dostupné z: <https://www.openoffice.cz/recenze/freecad-svobodna-alternativa-k-autocadu>
- [20] Začínáme - FreeCAD Documentation. *FreeCAD Documentation* [online]. Dostupné z: [https://wiki.freecadweb.org/Getting\\_started/cs#Navigace\\_ve\\_3D\\_prostoru](https://wiki.freecadweb.org/Getting_started/cs#Navigace_ve_3D_prostoru)
- [21] Pracovní plocha Skicář - FreeCAD Documentation. *FreeCAD Documentation* [online]. Dostupné z: [https://wiki.freecadweb.org/Sketcher\\_Workbench/cs](https://wiki.freecadweb.org/Sketcher_Workbench/cs)
- [22] Drawing tutorial - FreeCAD Documentation. *FreeCAD Documentation* [online]. Dostupné z: [https://wiki.freecadweb.org/Drawing\\_tutorial](https://wiki.freecadweb.org/Drawing_tutorial)
- [23] About KiCad, KiCad EDA. *KiCad EDA*[online], Dostupné z: <https://www.kicad-pcb.org/about/kicad/>

- [24] Výroba plošných spojů. *Microdesignum: MIDI ovladače pro pohodlné ovládání hudební techniky* [online]. Copyright © Microdesignum s.r.o. 2008 [cit. 02.06.2020]. Dostupné z: <https://www.microdesignum.cz/clanky/Vyroba-plosnych-spoju.html>
- [25] Jean-Pierre Charras, Fabrizio Tappero. Introduction to Eschema [Online]. Copyright © 2010-2018 [cit. 16.3.2020]. Dostupné z: <https://docs.kicad-pcb.org/5.1.5/en/eeschema/eeschema.html>
- [26] Introduction to Pcbnew [Online]. Jean- Pierre Charras, Fabrizio Tappero. Copyright © 2010-2015 [cit. 20.3.2020]. <https://docs.kicad-pcb.org/5.1.5/en/pcbnew/pcbnew.html>
- [27] Jindřich Fučík. Plugin modul do Inkscape pro export gcode. [online],[cit. 04.03.2020]. Dostupné z: [http://www.fucik.name/hw/laser/jfu\\_laser.php](http://www.fucik.name/hw/laser/jfu_laser.php)
- [28] Mads Aasvik.An Intro to G-code and How to Generate It Using Inkscape. [Online] Copyright © 2015 [08.02.2020] Dostupné z: <https://www.norwegiancreations.com/2015/08/an-intro-to-g-code-and-how-to-generate-it-using-inkscape/>
- [29] Introduction - FlatCAM 8 documentation. *FlatCAM: PCB Prototyping CAD/CAM* [online]. Copyright © Copyright 2014 [cit. 02.03.2020]. Dostupné z: <http://flatcam.org/manual/introduction.html>
- [30] Basics - FlatCAM 8 documentation. *FlatCAM: PCB Prototyping CAD/CAM* [online]. Copyright © Copyright 2014 [cit. 03.03.2020]. Dostupné z: <http://flatcam.org/manual/basics.html>
- [31] Isolation Routing - FlatCAM 8 documentation. *FlatCAM: PCB Prototyping CAD/CAM* [online]. Copyright © Copyright 2014 [cit. 10.03.2020]. Dostupné z: <http://flatcam.org/manual/iso.html>
- [32] Drilling - FlatCAM 8 documentation. *FlatCAM: PCB Prototyping CAD/CAM* [online]. Copyright © Copyright 2014 [cit. 10.03.2020]. Dostupné z: <http://flatcam.org/manual/drill.html>
- [33] Copper Area Clear - FlatCAM 8 documentation. *FlatCAM: PCB Prototyping CAD/CAM* [online]. Copyright © Copyright 2014 [cit. 09.03.2020]. Dostupné z: <http://flatcam.org/manual/clear.html>

[34] Two-side PCB - FlatCAM 8 documentation. *FlatCAM: PCB Prototyping CAD/CAM* [online]. Copyright © 2014 [cit. 15.04.2020]. Dostupné z: <http://flatcam.org/manual/doubleside.html>

[35] Board Cutout — FlatCAM 8 documentation. *FlatCAM: PCB Prototyping CAD/CAM* [online]. Copyright © 2014 [cit. 16.04.2020]. Dostupné z: <http://flatcam.org/manual/cutout.html>

## Externí odkazy

Tato kapitola obsahuje detailní video návody práce v programech.

OpenSCAD

<https://www.youtube.com/playlist?list=PLkRx3bM9e3yA7YhauhvCiV9v2a4RBp0Xa>

FreeCAD

[https://www.youtube.com/channel/UC\\_9HwDkwxllq5IFGkYBIH9g](https://www.youtube.com/channel/UC_9HwDkwxllq5IFGkYBIH9g)

KiCad

<https://www.youtube.com/watch?v=zK3rDhJqMu0&t=452s>

<https://www.youtube.com/playlist?list=PLEBQazB0HUyR24ckSZ5u05TZHV9khgA1O>

[https://www.youtube.com/playlist?list=PLCNJWVn9MJuORLQ3ds\\_U3D7RILfE4zdoL](https://www.youtube.com/playlist?list=PLCNJWVn9MJuORLQ3ds_U3D7RILfE4zdoL)

<https://www.youtube.com/watch?v=zK3rDhJqMu0>

FlatCAM

<https://www.youtube.com/watch?v=-Cb11heuHc>