

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

Využití robota LEGO Mindstorms EV3 - návrh robota hrajícího na kytaru pro propagaci FEL

Adam Chmiel

Vedoucí práce: Ing. Martin Hlinovský, Ph.D.
Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Chmiel** Jméno: **Adam** Osobní číslo: **492300**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra měření**
Studijní program: **Otevřená informatika**
Specializace: **Internet věci**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití robota LEGO Mindstorms EV3 - návrh robota hrajícího na kytaru pro propagaci FEL

Název bakalářské práce anglicky:

Usage of the LEGO Mindstorms EV3 - Design of the Guitar Playing Robot for Promotion of the Faculty

Pokyny pro vypracování:

1. Seznamte se s robotem postaveným ze stavebnice Lego Mindstorms EV3 umožňujícího zahrát doprovod na klasickou kytaru.
2. Upravte konstrukci mechanismu držícího akordy tak, aby zvládal zahrát jiné rozložení akordů.
3. Rozšiřte seznam skladeb, které bude možné zahrát na nové konfiguraci hardwaru.
4. Naprogramujte utilitu na PC se seznamem skladeb pro následné spouštění.
5. Popřípadě upravte řídicí program takovým způsobem, aby umožňoval parsovat písničky z předem domluveného formátu dat.
6. Vytvořte webové stránky k realizovanému projektu (popis, princip činnosti, vysvětlení navrženého softwaru, fotogalerii a popřípadě návod na stavbu robota).

Seznam doporučené literatury:

- [1] James Floyd Kelly - LEGO MINDSTORMS NXT-G programming Guide, Second Edition
- [2] Daniele Benedettelli - Programming LEGO NXT Robots using NXC
- [3] <https://www.youtube.com/watch?v=cXgB3llvPHI>
- [4] <https://www.youtube.com/watch?v=EN-7cMjmFv0>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Martin Hlinovský, Ph.D. katedra řídicí techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **03.02.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Martin Hlinovský, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat vedoucímu Ing. Martinu Hlinovskému, Ph.D za poskytnutí všech prostředků k vytvoření práce. Také bych mu chtěl poděkovat za rady a jeho věčnou ochotu poradit.

Dále bych chtěl poděkovat Matěji Štětkoví za radu ohledně tvorby uživatelského rozhraní

V poslední řadě bych chtěl poděkovat mé rodině a přátelům za neutuchající podporu během studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.
V Praze dne x. x. 2023

Abstrakt

Práce se zabývá rozšířením již předem vytvořeného robotického kytaristy. Mechanismus držící akordy byl upraven a rozšířen o nové akordy. K vytvoření mechanismu byl použit 3D tisk. Dalším rozšířením bylo přidání nových písniček. Dále se práce zaměřuje na komunikaci EV3 kostky s počítačem a vytvořením uživatelského rozhraní. V poslední části práce se teoreticky řeší možnosti parsování písniček.

Klíčová slova: robot, LEGO Mindstorms EV3, 3D tisk, uživatelské rozhraní, SSH, MIDI

Vedoucí práce: Ing. Martin Hlinovský, Ph.D.

Abstract

This thesis deals with the extension of a previously created robotic guitar player. The chord holding mechanism has been modified and extended with new chords. 3D printing was used to create the new mechanism. Another extension was the addition of new songs. Furthermore, the work focuses on the communication of the EV3 cube with a computer and the creation of a user interface. In the last part of the thesis, the possibilities of parsing songs are theoretically addressed.

Keywords: robot, LEGO Mindstorms EV3, 3D printing, user interface, SSH, MIDI

Title translation: Usage of the LEGO Mindstorms EV3 - Design of the Guitar Playing Robot for Promotion of the Faculty

Obsah

1 Úvod	1
2 Rozšíření mechanismu držícího akordy	3
2.1 Hlavní části mechanismu držícího akordy	3
2.2 Volba akordu čtyřakordové konfigurace	4
2.3 Osmiakordová konfigurace	5
2.3.1 Omezené možnosti lego dílků .	6
3 Tvorba vlastních součástek	7
3.1 Modelovací software - Blender ...	7
3.1.1 Nastavení Blenderu	7
3.2 Návrh součástek pro osmiakordovou konfiguraci	9
3.2.1 Návrh těla součástek	9
3.2.2 Návrh ramen součástek	10
3.2.3 Pořadí akordů	10
3.3 Modelování těla 3D tisknutých součástek	13
3.4 Modelování ramen součástek ...	15
3.5 3D tisk	16
3.6 Nedostatky osmiakordové konfigurace	17
4 Rozšíření seznamu skladeb	19
4.1 Proměnné pro vytváření písniček	19
4.2 Tvorba písniček	21
5 Uživatelské rozhraní Guitar GUI	23
5.1 Způsoby komunikace a připojení ke kostce	23
5.2 Guitar GUI	24
6 Parsování písniček z předem určeného formátu dat	27
6.1 MIDI protokol	27
6.2 Struktura MIDI souboru	27
6.2.1 MIDI události	28
6.2.2 Meta události	29
6.3 Nevýhody formátu MIDI souborů	29
7 Závěr	31
Literatura	33
A Materiály v odevzdaném archivu	35

Obrázky

2.1 Hlavní části 1	4
2.2 Hlavní části 2	4
2.3 Lego dílek design ID:42165. Převzato z [10]	6
3.1 Změna jednotek na milimetry ...	8
3.2 Změna měřítka mřížky	9
3.3 Stará tříramenná součástka. Převzato z [11]	10
3.4 Držení původních akordů	11
3.5 Držení nových akordů	11
3.6 Sjednocené schéma	12
3.7 Boolean Modifier	14
3.8 AxleBase.stl	14
3.9 Pětiramenná součástka	15
3.10 Jednoramenné součástky	16
4.1 Dílek ve tvaru slzy	20
4.2 Brnkací vzor Let it Be	20
5.1 Guitar GUI	24
6.1 Tóny a jejich přiřazená čísla. Převzato z [9]	28
6.2 Seznam nástrojů. Převzato z [8]	29

Tabulky

3.1 Počet součástek	13
3.2 Počet instancí	17

Kapitola 1

Úvod

Bakalářská práce se zabývá rozšířením robota hrajícího na kytaru. Bakalářská práce úzce navazuje na práci slečny Petrčkové, která ve své práci navrhla a postavila funkčního robota, schopného plnohodnotného kytarového doprovodu. K zajištění úspěšné práce jsem musel podrobně prozkoumat a nastudovat každý aspekt robota.

Téma práce jsem si vybral, protože jsem nadšený fanoušek stavebnice LEGO, ale na rozdíl od slečny Petrčkové rozhodně nejsem kytarista. Dalším důvodem byla unikátnost tématu. Na internetu nenajdete kromě jednoho YouTube videa v podstatě nikoho, kdo by se pokoušel vytvořit takového robota. Unikátnost tématu mi dala velký prostor pro kreativitu. Na začátku jsem měl robota, kterému jsem nerozuměl a body v zadání, které jsem musel splnit. Během práce jsem musel vymyslet řešení, vypracovat ho a nakonec otestovat. V poslední řadě mě fascinovala možnost, že by programovatelná kostka a dílky LEGO stavebnice, dokázaly něco tak sofistikovaného, jako je hra na kytaru.

Moje práce je rozdělena na několik částí, které odpovídají jednotlivým částem zadání. První částí je rozšíření mechanismu, umožňující držení akordů. Chtěl jsem zachovat akordy, které robot již uměl a přidat další. Zde jsem rychle narazil na omezené možnosti LEGO dílků, a proto jsem se rozhodl vytvořit si LEGO dílek vlastní, pomocí 3D tisku. Zároveň jsem se snažil zachovat základní funkce robota, vytvořené slečnou Petrčkovou.

Druhou částí je rozšíření repertoáru robota. Více akordů sice umožňuje zahrát více písniček, ale proces vytvoření písničky pro kytarového laika, jako jsem já, může být matoucí. V této části se zabývám postupem programování písniček. Také vysvětluji několik principů kytarové hry a způsob jak je převést na kód.

Třetí částí je vytvoření grafického rozhraní pro výběr písniček. To má umožnit uživateli ovládat kostku přes počítač. S funkčním grafickým rozhraním si bude moc kdokoli, nezávisle na množství znalostí o robotovi vybrat jednu z naučených písniček a nechat robota ji zahrát.

V poslední části se zabývám možností parsovat písničky do robota z předem domluveného formátu dat. V ideálním případě by uživatel pouze nahrál písničku v předem domluveném formátu do robota a ten by byl schopný ji zahrát.

Kapitola 2

Rozšíření mechanismu držícího akordy

V následující kapitole budou vysvětleny základní principy mechanismu, který drží akordy. Nejdříve bylo nutné pochopit jak celý mechanismus funguje po stránce mechanické i programové. Bylo nutné se rozhodnout, které části zachovat a které pozměnit. Základní princip držení akordů byl ponechán, ale konstrukce se musela upravit, stejně jako programová část. Čtyři původní akordy nebyly změněny a čtyři nové akordy byly přidány. Při popisu částí je kladen důraz na ty části mechanismu, které jsou využívány pro vytvoření mojí práce.

Kapitola je rozdělena na dvě části. První se zabývá původním mechanismem, navrženým slečnou Petrčkovou. V ní je popsáno vše potřebné, pro rozšíření mechanismu. V druhé části je popsána nová konfigurace, obsahující nové akordy.

2.1 Hlavní části mechanismu držícího akordy

V celém textu je používáno jednotné názvosloví pro důležité části mechanismu držícího akordy. Hlavními pojmy jsou: čtyřakordová konfigurace, osmiakordová konfigurace, pražcová osa (nebo jen osa), pražcové závaží a součástka.

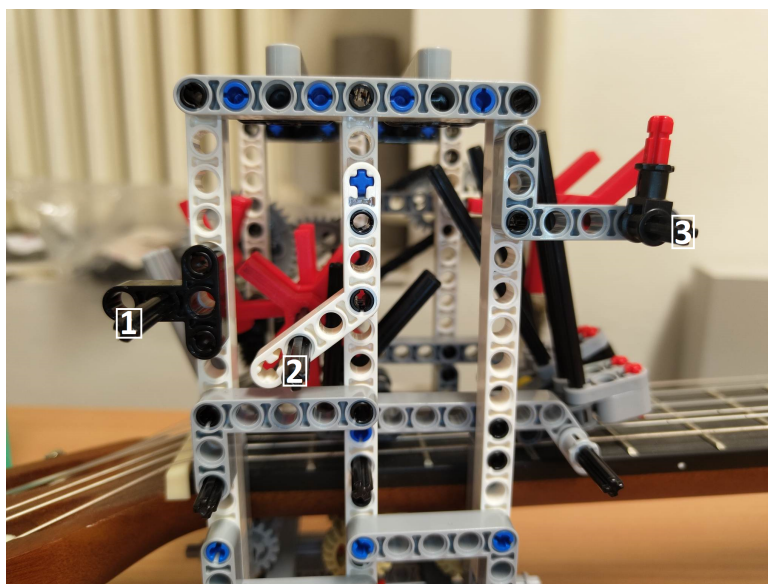
Čtyřakordová konfigurace je původní mechanismus, vytvořený slečnou Petrčkovou. Konfigurace umí čtyři akordy: **C**, **G**, **Am** a **F**. Zatímco osmiakordová konfigurace je rozšíření, umožňující zahrát k původním čtyřem akordům i čtyři nové: **D**, **E**, **Em** a **A**.

Pražcové osy jsou osy, které drží součástky, stlačující pražcová závaží. Celkem jsou tři (1), (2), (3) na obrázku 2.1], každá pro jeden využívaný pražec. Samotná osa je standardní LEGO dílek s profilem ve tvaru znaménka plus. S otáčením motoru, jež ovládá držení akordů se otáčí i osy a s nimi součástky.

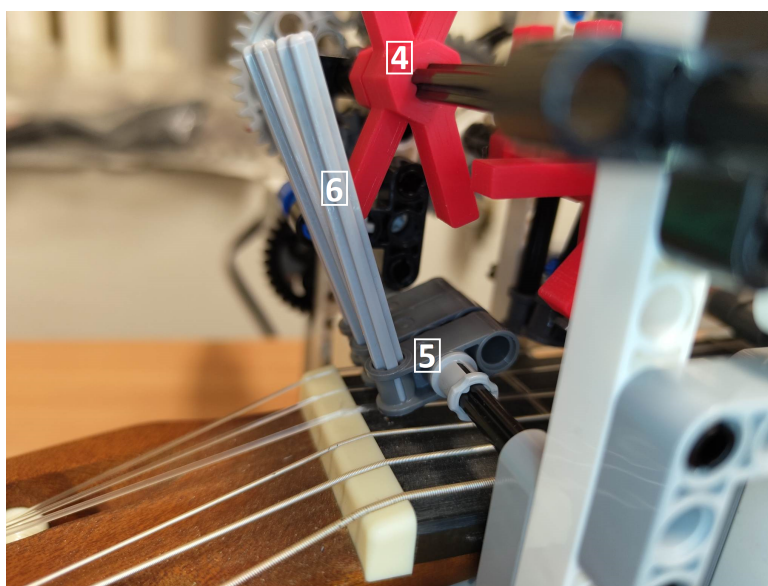
Součástky (4) jsou části, vytvořené na 3D tiskárně. Součástky jsou usazeny na pražcových osách. Jejich funkce je stlačování pražcových závaží. Správným natočením součástky se struna zmáčkne a imituje tak lidskou ruku při hraní jednotlivých akordů. Na použitých obrázcích jsou nové vytisknuté součástky, ne původní, vytvořené z LEGO dílků.

Pražcová závaží mají dvě části. Tyč (6) na kterou tlačí součástky a tělo (5), které je připevněno těsně nad strunou. Zatlačením na tyč se celé závaží pohne, čímž dojde ke stlačení struny. Chceme aby rameno součástky tlačilo na tyč

závaží co největší silou, k tomu dochází když je rameno kolmo k tyči. Pokud bude závaží stlačeno málo, zvuk struny bude tlumený.



Obrázek 2.1: Hlavní části 1



Obrázek 2.2: Hlavní části 2

2.2 Volba akordu čtyřakordové konfigurace

Celý mechanismus uvádí do pohybu jeden střední servo motor a navazující kaskáda ozubených koleček. Když se motor otáčí, otáčí se i všechny tři prahčové osy. Čtyři hratelné akordy, jsou seřazeny následovně: **A**m, **F**, **C** a **G**.

Motor je naprogramovaný tak, aby se otáčel o přirozený násobek 90ti stupňů. Jedno otočení motoru o 90 stupňů přehodí držený akord na následující v sekvenci. Motor je naprogramovaný tak, aby se točil pouze jedním směrem a celá sekvence se opakuje v kruhu. Otočení o 90 stupňů, když je zvolený akord G, nás dostane na začátek celé sekvence a zvolí se akord Am. Pokud se má motor otočit třikrát o 90 stupňů (např. z Am na G), tak motor se neotočí po částech, ale rovnou o 270 stupňů.

Pořadí akordů záleží pouze na tvaru součástek na osách. Obecně můžeme říci, že konfigurace má čtyři možné pozice. Mezi pozicemi volíme otáčením motoru po násobcích 90ti stupňů. Pro číslování pozic bylo použito indexování od nuly (Na pozici 0 je akord Am, na pozici 1 akord F, atd.).

Nahrazení jmen akordů očíslovanými pozicemi je jistá míra abstrakce. Ta nám pomůže s výpočtem požadovaného úhlu pro otočení motoru, když chceme měnit pozice. Pro čtyřakordovou konfiguraci je vzorec následující:

$$A = \{[(4 + p) - c] \% 4\} \cdot 90 \quad (2.1)$$

Použité proměnné:

- c - číslo současné pozice mechanismu
- p - číslo chtěné pozice
- A - výsledný úhel

Příklad: Na kytáře je zvolený akord C(2), chceme aby se vybral akord Am(0). Pro tento příklad bude c=2 a p=0. Po dosazení do vzorce dostaneme:

$$A = \{[(4 + 0) - 2] \% 4\} \cdot 90$$

$$A = 180$$

Vzorec 2.1 můžeme zobecnit pro konfigurace obsahující alespoň jednu pozici. Uvažujme n možných pozic, číslovaných od 0 až po n-1, kde n>0. Zobecnění pro tyto podmínky bude následující:

$$A = \{[(n + p) - c] \% n\} \cdot (360 \div n) \quad (2.2)$$

Výsledný vzorec bude platit pro všechna n>0.

2.3 Osmiakordová konfigurace

Nyní už víme vše potřebně o původní čtyřakordové konfiguraci a můžeme začít tvořit konfiguraci osmiakordovou. Původní čtyři akordy jsou součástí nejpoužívanější akordové posloupnosti C-G-Am-F. Můžeme najít obrovské množství písniček, které si vystačí pouze s těmito akordy. Tato posloupnost je v klíči C dur.

Nové akordy byly vybrány z tutoriálových videí, určené pro kytaristy začátečníky. Mnoho videí kromě posloupnosti C-G-Am-F zmiňovalo ještě

posloupnost G-D-Em-C. Jedná se o tu samou sekvenci akordů jen transponovanou z klíče C dur do klíče G dur. Posloupnost obsahuje dva akordy, které robot neumí: **Em** a **D**. Jako další dva akordy jsem se rozhodl přidat **A** a **E**, protože se ve videích často vyskytovaly.

Program robota musíme upravit tak, aby uměl motorem otáčet o násobky 45ti stupňů. Pak bude robot umět vybírat libovolnou pozici z osmi možných. Do vzorce 2.2 dosadíme $n=8$ a získáme:

$$A = \{[(8 + p) - c] \% 8\} \cdot 45$$

Jediné co zbývá, je vytvořit z lega nové součástky, které budou moci mít až 8 ramen.

2.3.1 Omezené možnosti lego dílků

Klasické dílky lego stavebnic nejsou vhodné pro vytvoření součástek pro osmiakordovou konfiguraci. Pokud spojíme dva dílky, většinou budou mezi sebou svírat úhel 90 stupňů. Existují zahnuté nosníky, jako je například lego dílek s designovým číslem 42165(2.3).



Obrázek 2.3: Lego dílek design ID:42165. Převzato z [10]

Tento dílek by mohl sloužit jako součástka pro osmiakordovou konfiguraci, kdyby neměl dvě zásadní vady. Součástku bychom chtěli přichytit na osu v místě, kde se láme(1). Otvor není ve tvaru osy(2), takže by se protácel. Mohli bychom ho udržet v pevné pozici, kdybychom využili jiný dílek, který by spojoval nosník a osu. Součástka by ale byla moc široká a tlačila přes více pračcových závaží. Dalším nedostatkem je úhel zlomu nosníku, ten je 53 stupňů, ale my potřebujeme násobky 45ti stupňů. Standardní nosníky mají dva možné úhly zlomu: 53 stupňů a 90 stupňů. Pokoušel jsem se z dostupných dílků(nejen nosníků) navrhnout součástku, která by vyhovovala všem požadavkům osmiakordové konfigurace, bohužel však neúspěšně. Rozhodl jsem se vytvořit vlastní součástku pomocí 3D tisku.

Kapitola 3

Tvorba vlastních součástek

Následující kapitola se zabývá procesem návrhu, modelování a tisku vlastního LEGO dílku. K vytvoření dílku byly využity možnosti 3D tisku. Pro tuto možnost jsem se rozhodl ze jednoho hlavního důvodu. Nepodařilo se mi vytvořit potřebnou součástku z dostupných sériově vyráběných LEGO dílků. Konstrukce buď nebyla dostatečně pevná a nebo nebyla schopná držení akordů v nové osmiakordové konfiguraci.

Kapitola začíná zmínkou o použitém softwaru a jeho nastavení. Běžný modelář má k dispozici řadu programů, placených i volně dostupných. Je důležité zmínit, že potřebujeme dva druhy programů. Jeden k vytvoření modelu a exportu souboru .stl. Druhý ke slicování modelu a vytvoření souboru obsahující G-code, tedy instrukce pro 3D tiskárnu.

V další části jsou popsány postupy pro vytvoření potřebných součástek pro robota, včetně procesu návrhu vhodných součástek pro nové rozložení akordů. Samotné vytvoření modelu součástek není nijak zvlášť složité. Je ovšem důležité, aby návrh splňoval několik kritérií, potřebných pro fungování nové osmiakordové konfigurace.

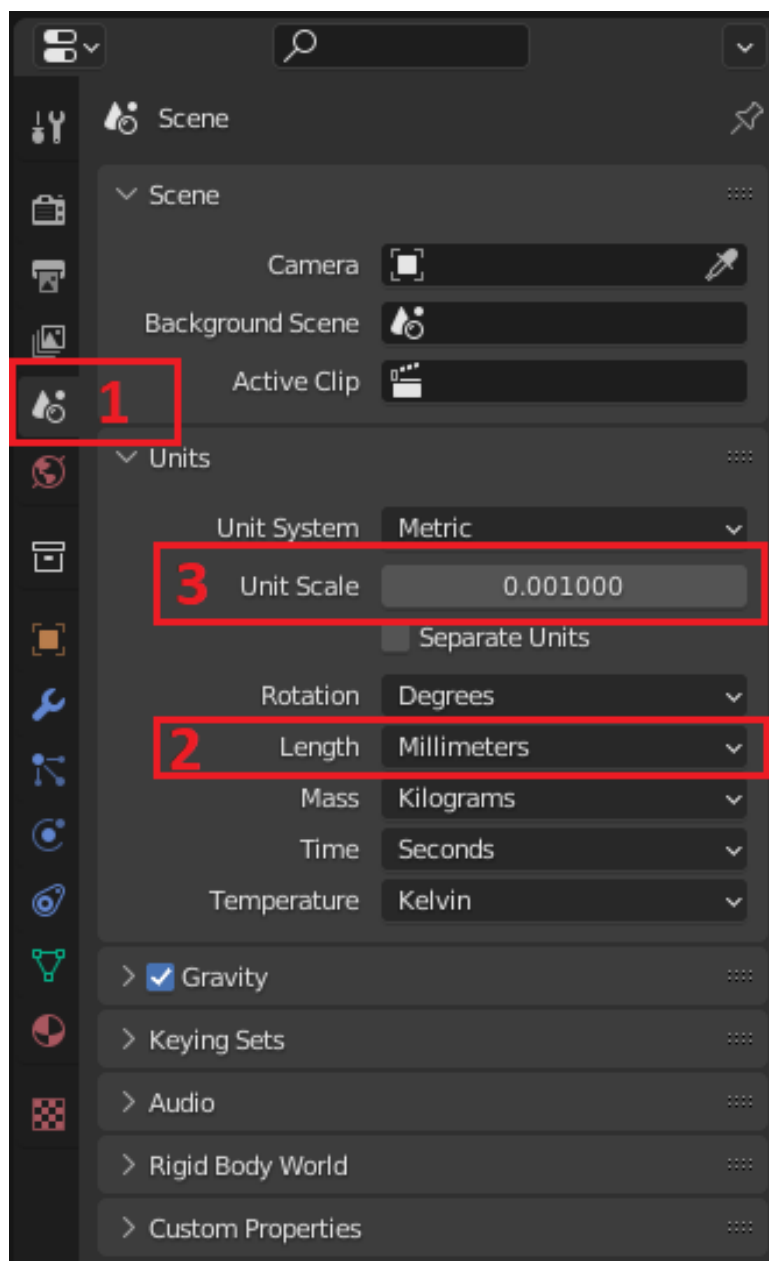
3.1 Modelovací software - Blender

K modelování součástek jsem použil program Blender. Jeho hlavní využití spočívá ve vytváření 3D modelů do videoher, animací či vizuálních efektů. Obsahuje všechny potřebné funkce i pro vytváření modelů pro 3D tisk. Blender má velkou uživatelskou komunitu, která vytvořila spoustu dodatečně nainstalovaných rozšíření pro 3D tisk. Pro vytvoření součástky, ale stačí základní Blender verze 2.8 či novější. Přestože primární účel Blenderu není vytváření modelů určených pro 3D tisk, byl vybrán ze dvou hlavních důvodů. Za prvé je zdarma a volně šířitelný. Za druhé Blender jsem již používal a jeho ovládání mám již dobře zažitě.

3.1.1 Nastavení Blenderu

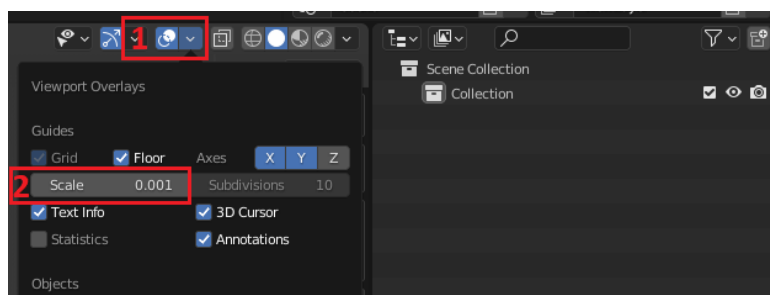
Ve výchozím nastavení je Blender nastaven na jednotky nevhodné pro 3D tisk - metry. Pro modelování 3D součástek potřebujeme Blender nastavit na milimetry. Správně nastavené jednotky lze vidět na obrázku 3.1. V bočním

panelu vybereme záložku **Scene properties** (1), zde musíme změnit v části units Length z **Meters** na **Millimeters** (2). Tato změna ale pouze změní používané jednotky a ne skutečnou velikost objektů, proto musíme změnit Unit Scale (3) z **1** na **0.001**. Druhé nastavení, které musíme změnit je měřítko



Obrázek 3.1: Změna jednotek na milimetry

pomocných naváděcích čar, které tvoří na obrazovce mřížku. Její velikost je stále nastavena na metry. Toto nastavení se mění rozklikem šipky vedle ikony **Show Overlay**(1) a změněním hodnoty **Scale**(2) z **1** na **0.001**. Správné nastavení mřížky můžeme vidět na obrázku 3.2.



Obrázek 3.2: Změna měřítka mřížky

Výsledkem bude mřížka o délce hrany 1 milimetr. Všechny nově přidané nebo importované objekty budou nyní v správně nastaveny v milimetrech.

3.2 Návrh součástek pro osmiakordovou konfiguraci

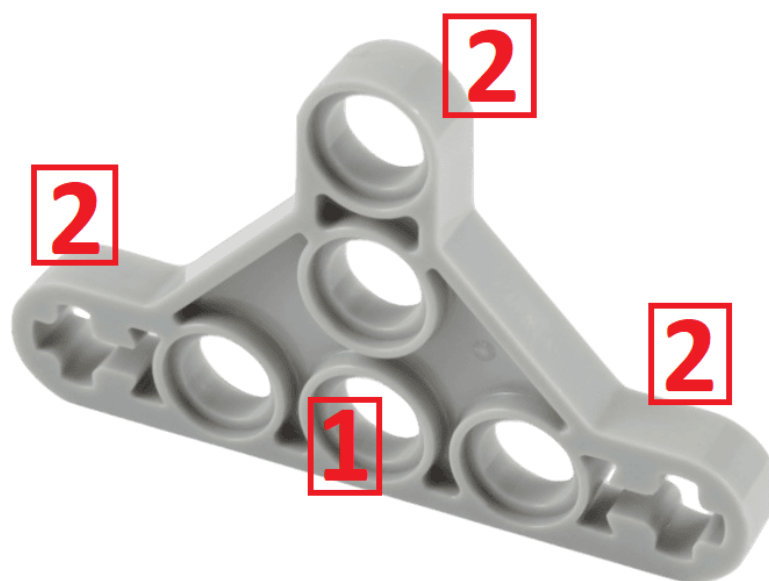
Návrh součástek byla časově nejnáročnější část procesu vytvoření nové osmiakordové konfigurace. Tato konfigurace kromě původních čtyř akordů (**A_m**, **F**, **C**, **G**), obsahuje čtyři akordy nové (**A**, **E_m**, **D**, **E**). Pro navrhnutí součástek je potřeba si uvědomit několik kritérií, které musí splňovat.

Součástky budou mít dvě hlavní části: tělo a ramena. Tělo bude zaručovat správné přichycení součástky na pohyblivou osu. Tělo musí pevně sedět na ose, pokud by bylo moc volné, mohlo by se protočit. Ramena budou fungovat jako síla tlačící na pražcové závaží.

Pražcové osy na kterých se budou součástky pohybovat nejsou všechny stejně daleko od sebe. Osy pro první a druhý pražec jsou blíže sobě než osa pro druhý a třetí pražec. Součástky musí mít dostatečně dlouhá ramena, aby struna byla pořádně stlačena. Kvůli tomu, že jsou osy prvního a druhého pražce blíže, může dojít k zaklesnutí součástek do sebe a rozbití mechanismu. Pořadí akordů se musí vhodně navrhnout, tak aby do sebe součástky nezaklesnuly.

3.2.1 Návrh těla součástek

Návrh součástek a jejich částí se inspiruje LEGO dílky, které zastávaly funkci součástek u čtyřakordové konfigurace. Na obrázku 3.3 vidíme LEGO dílek, který funguje jako tříramenná součástka. Tvar těla(1) součástky se odvíjí od počtu akordů použité konfigurace. Jelikož chceme mít osm akordů, bude mít tělo tvar pravidelného osmibokého hranolu. Uprostřed hranolu bude otvor ve tvaru pražcové osy. Lego dílek na obrázku má velké tělo(1) a kratší ramena(2). U vlastní součástky bylo žádoucí, aby tělo bylo co nejmenší a ramena delší než ve čtyřakordové konfiguraci.



Obrázek 3.3: Stará tříramenná součástka. Převzato z [11]

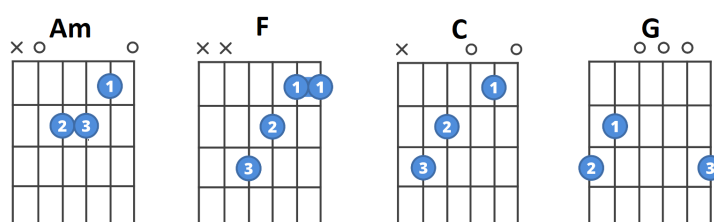
■ 3.2.2 Návrh ramen součástek

Rameno si můžeme představit jako prodloužení jedné z osmy stěn hranolu. Počet ramen součástky se rovná počtu pozic, ve kterých je struna stlačena. Původní čtyřakordová konfigurace byla použita k určení potřebné délky ramen pro jednotlivé osy. Ramena součástek na první a druhé ose jsou stejně dlouhá. Ramena součástek na třetí ose jsou o 5mm delší.

Pražcové osy se neotáčejí stejným směrem. Osy prvního a třetího pražce se otáčejí proti směru hodinových ručiček, osa druhého pražce po směru (při pohledu na mechanismus ze strany, na které není motor). Při vytváření součástek musíme dbát na směry otáčení osy a umístění pražcového závaží (vlevo nebo napravo od osy). Směr otáčení osy určuje pořadí ramen pro jednotlivé pozice mechanismu. Umístění závaží zase udává, které stěna osmibokého hranolu je pozice číslo nula.

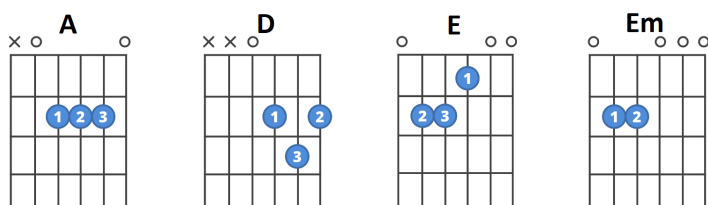
■ 3.2.3 Pořadí akordů

Původní čtyřakordová konfigurace umí držet akordy v tomto pořadí: **A_m**, **F**, **C** a **G**. Držení těchto akordů vidíme na obrázku 3.4. U akordu **F** vidíme, že se hraje jen přes čtyři struny (horní dvě mají nad sebou nakreslený křížek), robot ale umí jen hru přes všech 6 nebo přes 5 spodních strun. Akord lze zahrát i přes pět spodních strun, některé zdroje uvádí použití pěti strun, některé zase čtyř. Záleží na osobní preferenci hráče. Robot jej bude hrát přes strun pět, jelikož pro hru přes 4 struny by se musela přestavět celá konstrukce, která se stará o pohyb trsátka.



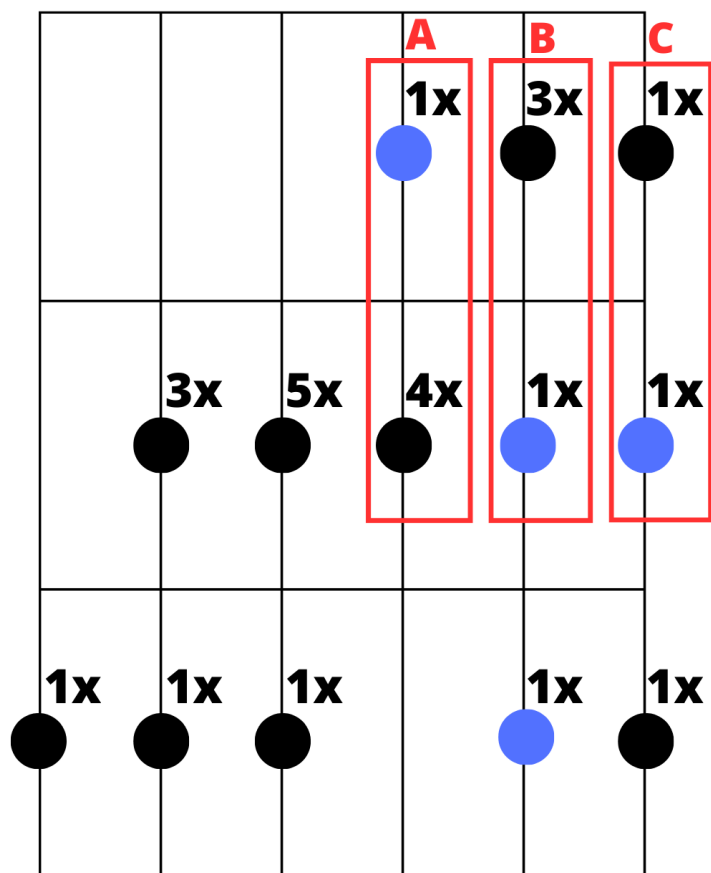
Obrázek 3.4: Držení původních akordů

Původní čtyři akordy chceme zanechat a přidat k nim čtyři akordy nové. Držení nových čtyř akordů (**A**, **D**, **E** a **Em**) je vidět na obrázku 3.5. Akord **D** se ze stejného důvodu, jako akord **F** bude hrát přes pět strun.



Obrázek 3.5: Držení nových akordů

Nyní musíme všechny akordy „sjednotit do jednoho“. Všechny použité struny na jednotlivých prazcích nakreslíme do jednoho schématu. Ke strunám napíšeme počet akordů, které je využívají. Na sjednoceném schématu lze dobře vidět možná místa konfliktu mezi součástkami (místa kde by se součástky mohli do sebe zaseknout). Sjednocením všech osmi akordů, které bude robot umět hrát, dostaneme následující schéma 3.6.



Obrázek 3.6: Sjednocené schéma

Černé body zvýrazňují součástky, které používají původní i nově přidané akordy. Modré body zvýrazňují součástky, které používají pouze nově přidané akordy. Pro modré body bude zapotřebí přidat pražcové závaží nad strunu. Vedle každého bodu je napsaný počet akordů vyžadující zmáčknutou strunu na této pozici. Jednodušeji řečeno, číslo vedle bodu udává počet ramen součástky. Červené obdélníky **A**, **B** a **C** zvýrazňují místa možného konfliktu. Konflikty mohou vzniknout pouze mezi součástkami ovládající pražcové závaží 1. a 2. pražce, kvůli menší vzdálenosti mezi závažími 1. a 2. pražce.

Celkem se na sjednoceném schématu vyskytuje 13 bodů přes tři pražce. Potřebujeme tedy 13 správně umístěných pražcových závaží a 13 vytištěných součástek. Seznam potřebných součástek s počtem kusů je v tabulce 3.1.

Počet ramen	Počet kusů
1	9
3	2
4	1
5	1

Tabulka 3.1: Počet součástek

Po vytvoření sjednoceného schématu musíme akordy správně seřadit. Součástky s největším počtem ramen budou zaujímat nejvíce místa. Moje řešení se snaží minimalizovat obsah součástek, které mají největší počet ramen. Pětiramenná součástka nemůže vytvořit konflikt, na rozdíl od součástky se čtyřmi rameny (červený obdélník A na obrázku 3.6).

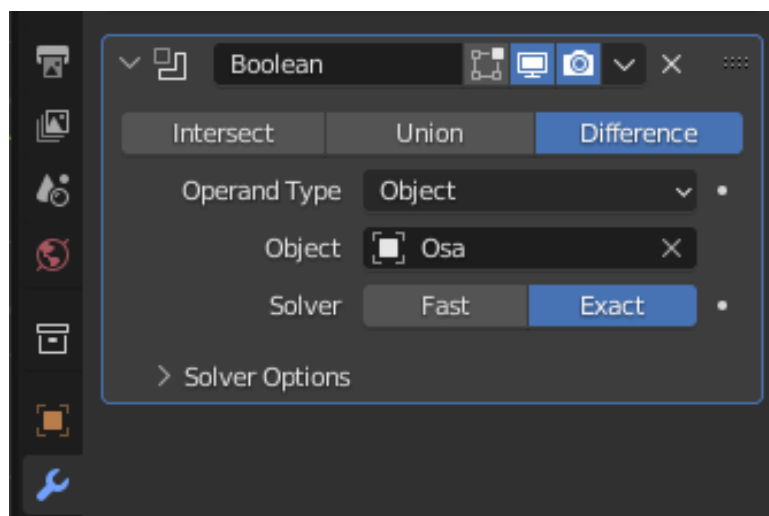
Čtyřramenná součástka stlačuje pražcové závaží struny **G** (třetí nejtenčí) na druhém pražci. Tuto pozici využívají akordy: **Am**, **F**, **A** a **D**. Zvolíme tedy tyto čtyři akordy jako první v celé sekvenci. Nyní se musíme podívat na součástku, se kterou by mohlo dojít ke konfliktu. Tou je jednoramenná součástka na stejné struně o pražec výše. Tuto pozici využívá akord **E**. Vložíme ho jako další po akordu **D**. Tento postup byl zopakován pro zbylé dvě místa možného konfliktu, tím vznikla finální sekvence všech osmi akordů: **Am**, **F**, **A**, **D**, **E**, **C**, **G** a **Em**

Sekvenci ještě bylo zapotřebí otestovat. Místo tisknutí součástek byly vytvořeny primitivní modely součástek z papíru, které sloužily k simulaci otáčení součástek na osách o 45 stupňů. Tento postup byl příjemnější, než si otáčení součástek na osách představit v hlavě.

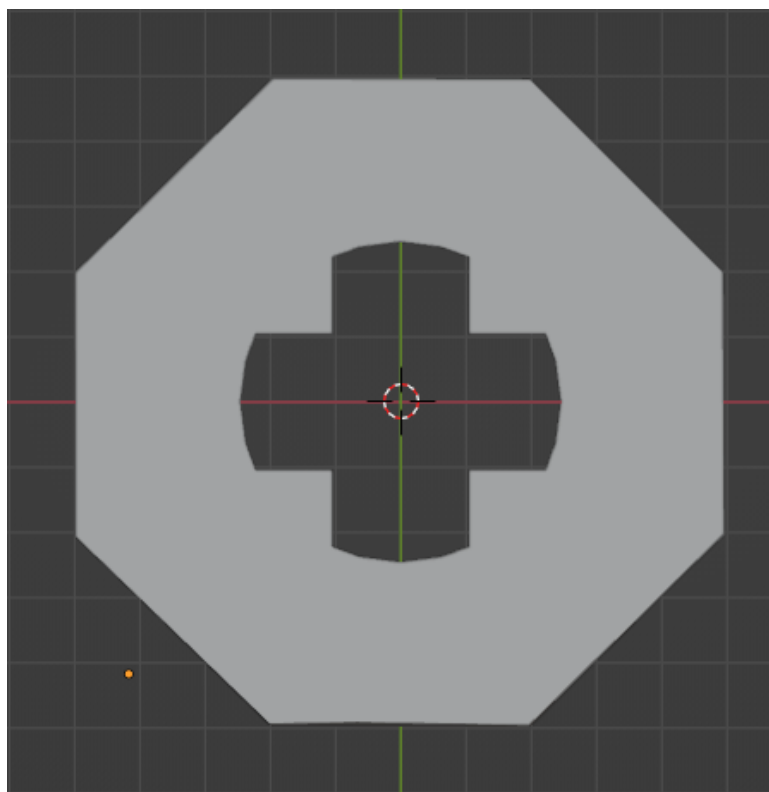
3.3 Modelování těla 3D tisknutých součástek

Největší výzvou při vytváření modelů součástek bylo vytvoření těla, které bude pasovat dobře na pražcovou osu. Na začátku byl vytvořen pravidelný osmiboký hranol o výšce 6mm. Ten byl umístěn doprostřed modelovací plochy, tak aby počátek souřadnicové soustavy byl přímo uprostřed spodní podstavy hranolu. Dále jsem si na internetu našel model pražcové osy ve formátu .stl. Model osy jsem importoval do Blenderu, osu správně natočil a vycentroval, aby procházela přesně prostředkem hranolu. Hranolu byl přidán modifier Boolean, s navolenou možností Difference a zvolena osa jako objekt k odečtení (3.7). Po aplikování modifíku se od hranolu odečte osa a uprostřed hranolu vznikne otvor ve tvaru pražcové osy.

Tělo součástky jsem si na 3D tiskárně vytiskl a snažil se ho na pražcovou osu nasadit. Otvor byl ale moc malý a tělo součástky nešlo na osu navléct. Otvor uprostřed hranolu bylo potřeba zvětšit. Po několika neúspěšných pokusech se podařilo vytvořit tělo součástky, které na osu pasovalo pevně. Aby došlo k zachování symetrie těla, byl otvor zvětšen o 0,05 milimetrů na každé straně.



Obrázek 3.7: Boolean Modifier

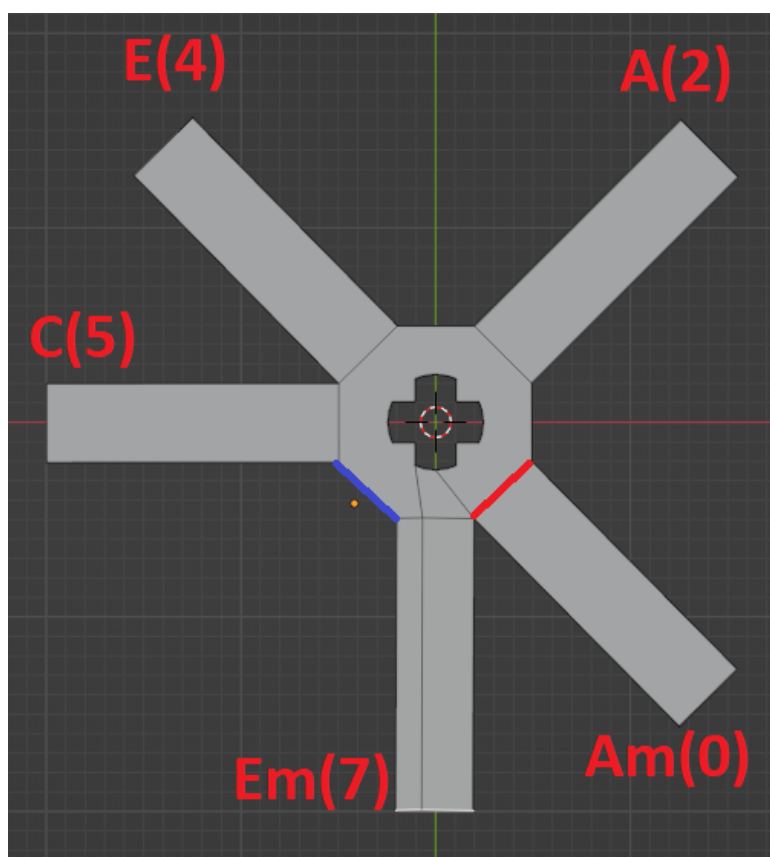


Obrázek 3.8: AxleBase.stl

Na obrázku(3.8) můžeme vidět finální model těla součástky(pohled shora). Model těla se jmenuje AxleBase.stl. Všechny modely(.blend) a jejich exportované verze(.stl) jsou ke stažení na mém gitlabu.

3.4 Modelování ramen součástek

S funkčním modelem těla součástky bylo možno začít tvořit finální modely součástek. Postup tvoření součástek je vysvětlen na modelu pětiramenné součástky(3.9). Prvním krokem je sepsání všech akordů, které součástku využívají a jejich pozice. Pro pětiramennou součástku to jsou: $A_m(0)$, $A(2)$, $E(4)$, $C(5)$ a $E_m(7)$. Součástky na druhé pražcové ose mají pražcové závaží napravo od osy. Na obrázku je červenou čarou vyznačena pozice číslo 0. Pokud by závaží bylo nalevo od osy, bude pozice 0 na místě modré čáry. Pozice jsou vždy u všech součástek číslovány v opačném směru než je směr otáčení osy(po jednom otočení po směru hodinových ručiček bude na předešlé pozici 0 nyní pozice 1). Nyní musíme na každé pozici, kde je akord vytvořit rameno.

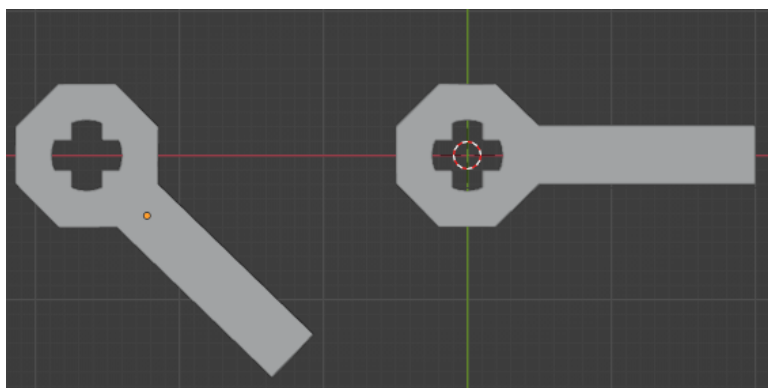


Obrázek 3.9: Pětiramenná součástka

Ramena se modelují pomocí extrudování stěn hranolu. Extrudování vytvoří kopii vybrané stěny, kterou můžeme potom přemístit, a spojí kopii s původní stěnou. K extrudování nejdřív vybereme chtěnou stěnu a zmáčkneme klávesu E, nyní na numerické klávesnici zadáme délku ramena. Zadání délky přes numerickou klávesnici vyextruduje stěnu podél normálového vektoru stěny, přesně jak potřebujeme. Pro součástky na první a druhé pražcové ose je délka ramen 15mm, pro součástky na ose třetí 20mm. Postup byl zopakován

pro všechny součástky s více než jedním ramenem. Hotový model je nutno vyexportovat jako soubor .stl. Pomocí tohoto postupu jdou vytvořit i součástky jednoramenné, ale existuje způsob jednodušší.

Jednoramenné součástky můžeme libovolně převracet a otáčet když je nasazujeme na osu. Součástku vlevo na obrázku 3.10 můžeme využít pro všechny akordy se sudou pozicí, tu napravo pro všechny akordy s pozicí lichou. Např.: Jednoramenná součástka pro akord Am(0), může sloužit jako součástka pro akord A(2), když ji nasazujeme na osu, otočíme ji o 90 stupňů ve směru číslování pozic.



Obrázek 3.10: Jednoramenné součástky

3.5 3D tisk

Dalším krokem ve vytváření osmiakordové konfigurace je tisk vymodelovaných součástek. Součástky byly tištěny na tiskárně od české firmy *PrusaReaserch*, ta k tiskárnám vytvořila i vlastní slicovací software *PrusaSlicer*. Všechny modely ve formátu .stl byly naimportovány do PrusaSliceru. K jednoramenným součástkám byly přidány další instance, aby jich byl potřebný počet. Finální počet tisknutých součástek je v tabulce 3.2. Stisknutím klávesy A se optimalizuje uspořádání součástek. Jako materiál pro tisk byl zvolen nejběžnější filament PLA. Nastavení kvality tisku bylo použito 0.15mm QUALITY s hustotou výplně 15%. Po zkontrolování počtu instancí a nastavení tisku, stiskneme vlevo dole Slicovat a program vytvoří G-Code. Ten nahrajeme na SD kartu, vložíme ji do 3D tiskárny, na displeji zvolíme náš soubor a počkáme než bude tisk hotov. Celková doba tisku byla 1h a 35m.

Když je tisk hotov, počkáme ještě chvíli, než vychladne nahřívací deska a vylámeme hotové součástky. Hotové součástky mohou mít hrubé okraje a v otvoru uprostřed součástky se může vyskytovat přebytečný materiál. V tomto případě je očistíme nožem nebo pilníkem. Je důležité vyzkoušet každou součástku nasadit na osu. Může se stát, že součástka bude moc volná, nebo pevná. Nestalo se mi, že by nešla vůbec na osu nasadit, někdy je zapotřebí trochu více zatlačit. Pokud jde navléct opravdu velmi těžce, nejde sundat, nebo je moc volná, můžeme ji zkusit vytisknout znovu s lepší kvalitou tisku. Po úspěšném nasazení doporučuji součástku ještě několikrát sundat a opět-

vaně nasadit, vnitřní strany otvoru se tak trochu obrousí a nasazování půjde lépe.

Název souboru .stl	Počet instancí
5stick_short	1
4stick_short	1
3stick_short	1
3.1stick_short	1
1stick_short	2
1.1stick_short	2
1stick_long	2
1.1stick_long	3

Tabulka 3.2: Počet instancí

Nyní už jen zbývá nasadit vytisknuté součástky na pražcové osy a přidat nová pražcová závaží. Mechanismus začneme rozebírat odshora a postupně se dopracujeme až k pražcovým závažím. Nad třetím pražcem je 5 závaží a domnívám se, že není možné aby jich bylo 6. Závaží by byla moc zmáčknutá u sebe a zatlačením na jedno by se jich pohybovalo více. Při nasazování součástek na osy, jsem musel několikrát zkontrolovat jestli je jejich pořadí a natočení správně. Celý proces vyžadoval značné množství trpělivosti. Po nasazení všech součástek, byl mechanismus poskládan zpátky a osmiakordová konfigurace byla hotova.

3.6 Nedostatky osmiakordové konfigurace

S dokončenou osmiakordovou konfigurací bylo možné začít vytvářet nové písničky, které by nové akordy využili. Až při prvním testu písničky jsem narazil na vadu mého řešení. Občas se stává že zvuk některých strun zní jinak než by měl, nebo tlumeně. Problémem je samotné množství akordů konfigurace. K chybě dochází když struna nemá být v současném natočení součástek stlačená, ale v následujícím ano. Úhel otáčení po 45 stupních je moc malý a rameno součástky se závaží dotýká i když ještě nemá. Tato malá síla někdy stačí k tomu, aby strunu utlumila. Přidávání písniček byla poslední část zadání, které jsem se věnoval a když jsem chybu objevil neměl jsem už dostatek času, abych implementoval konkrétní způsob jak chybu odstranit. Možné způsoby odstranění vady:

- Zmenšení počtu akordů - Kdybychom měli například jen šest akordů a motor by se otáčel po 60 stupních, je možné že by se problém vyřešil. Museli bychom opět použít 3D tisknuté součástky. Tělo součástky by se změnilo na šestiboký pravidelný hranol.
- Přestavení mechanismu držící akordy - Zvětšením vzdálenosti mezi pražcovými závažími a strunami by se mohl problém odstranit. Pravděpodobně

by se ale musel předělat celý mechanismus.

- Změna pražcových závaží - Navrhnout nové pražcové závaží, které by vadu eliminovalo. Opět by se asi musel předělat celý mechanismus

Je nutno říct, že zmíněné způsoby jsou jen návrhy, které mě napadli, jako první při identifikaci příčiny problému.

Kapitola 4

Rozšíření seznamu skladeb

Následující kapitola se zabývá postupem vytváření písniček pro robota. Robot již dvě písničky uměl: Hey, Soul Sister od kapely Train a Castle of Glass od skupiny Linkin park. Tyto dvě písničky byly použity jako vzor při vytváření nových. Proces vybírání nových písniček byl časově náročný. Při vyhledávání písniček jsem často narazil na internetové články, vygenerované umělou inteligencí s nepravdivými informacemi a nefungujícími referencemi. Každá písnička lze zahrát více způsoby a hráči do doprovodu často přidávají vlastní prvky, typické pro jejich styl hraní. Robot má omezené možnosti, tudíž není možné implementovat všechny způsoby hry na kytaru. Pro úspěšné vytvoření písničky je potřeba znát alespoň základní principy kytarové hry a způsob jak je převést na instrukce pro robota. Pro hraní robot vyžaduje dva údaje: **timeSignature** a **BPM**. TimeSignature určuje počet dob v taktu a BPM počet úderů za minutu. Vysvětlení jejich vlivu na hru je v následující části.

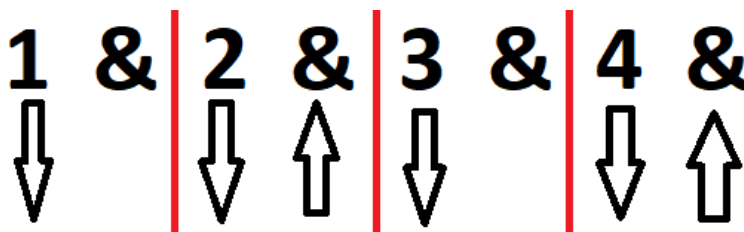
4.1 Proměnné pro vytváření písniček

Při hraní na kytaru pravá ruka vytváří sekvenci pohybů trsátka přes struny. Tato sekvence se většinou neustále opakuje dokola a změnou akordů vzniká písnička. Kombinace těchto pohybů se nazývá brnkací vzor. Počet pohybů ve vzoru se odvíjí od taktu písničky. Všechny písničky, které robot umí jsou ve čtyř-čtvrtovém taktu. To znamená že jeden takt má čtyři doby. V kytarové hře na jednu dobu připadají dva pohyby přes struny trsátkem. Vzory se mohou skládat ze 3 různých pohybů: úder dolů, úder nahoru nebo vynechání úderu. Ramenem s trsátkem pohybuje velký motor. Jedno otočení motoru o 360 stupňů způsobí jeden pohyb trsátka směrem dolů a jeden pohyb zpět směrem nahoru. To přesně odpovídá jedné době taktu. Na tyči ovládající trsátko je nasazen lego dílek ve tvaru slzy((1) na 4.1), který způsobuje, že se rameno při pohybu zpět nahoru zvedne a trsátko se strun nedotkne. Dílek lze nahradit jiným dílkem, který bude trsátko zvedat při pohybu dolů a nechá strunu znít při pohybu nahoru. Musí se tak ovšem provést před začátkem hry.



Obrázek 4.1: Dílek ve tvaru slzy

Na obrázku 4.2 můžeme vidět brnkací vzor nové robotovi písničky Let it Be od kapely Beatles. Červené čáry od sebe oddělují jednotlivé doby taktu. Šipky zobrazují směr pohybu přes struny. Nejdřív bychom tedy chtěli úder dolů s pomlčkou směrem nahoru a poté aby trsátko udeřilo přes struny při pohybu dolů i nahoru a ještě jednou stejnou sekvenci. To ovšem není možné. Během hry by se musela rozpojit tyč ovládací rameno s trsátkem a vyměnit dílek ve tvaru slzy za jiný. Písničky, které umí robot zahrát používají jednodušší vzor, pouze s úderem směrem dolů a pomlčkou (1. a 3. doba na obrázku 4.2). Při výběru písniček byl kladen důraz na to, aby originální vzor byl co nejvíce podobný, tomu co umí robot.



Obrázek 4.2: Brnkací vzor Let it Be

TimeSignature se rovná počtu dob v jednom taktu, neboli kolikrát se bude opakovat sekvence úder dolů a pomlka při zahrání jednoho akordu. U některých písniček se mění akord na část taktu. V tomto případě změníme timeSignature na počet dob trvání akordu a poté ho změníme zpět.

Z BPM (beats per minute) se vypočítává úhlová rychlost otáčení motoru. Údaj je možné zjistit z libovolného internetového zdroje a často bývá uveden v akordovém přepisu písničky. Byly vybírány písně s BPM menším než 120. Při použití vyššího BPM se rameno s trsátkem rozpadalo, tato vada byla odstraněna přestavěním ramena. Při vyšším BPM než 120, ale dochází ještě k jedné chybě. Po krátké době hraní dojde k přeskočení zubů na ozubených kolečkách a prohodí se pořadí úderů. Rameno bude zvednuté při pohybu dolů

a udeří přes struny při zpátečním pohybu. Po chvíli dojde k úplnému rozbití mechanismu a musí se celý restartovat.

4.2 Tvorba písniček

Všechny písničky jsou uloženy ve složce **Songs** a jednotlivé písničky jsou pojmenovány SongX.py. K vytvoření písničky je zapotřebí zkopírovat konstruktor třídy SongX z již zhotovených písniček. Nyní si připravíme pořadí a počet zahrání akordů konkrétní písničky. U většiny písniček lze nalézt alespoň dvě sekvence akordů, které se opakují. Těmi jsou sekvence pro sloku a pro refrén. Pro zvětšení přehlednosti kódu můžeme vytvořit metody, které zahrají celou sloku a refrén. Tyto pomocné metody poté zavoláme v hlavní metodě play.

```
#Metody play() a playVerse() písničky Wild Thing od
#skupiny The Troggs
def play(self):
    self.playVerse()
    self.playVerse()
    self.playVerse()
    self.playVerse()
    self.playChorus()
    self.playVerse()
    self.playVerse()
    self.playVerse()
    self.playVerse()
    self.playChorus()
def playVerse(self):
    #Takt má dvě doby. Dolů-Pomlka-Dolů-Pomlka
    self.timeSignature = 2
    self.playChord(a,self.currentPosition)
    #V písničce se objevují, krátké pauzy. Ty jsou
    #simulovány pomocí wait().
    wait(250)
    self.playChord(d,self.currentPosition)
    reset(self.robot.pickMotors)
    self.playChord(e,self.currentPosition)
    wait(250)
    self.playChord(d,self.currentPosition)
    #Slečna Petrčková ve své práci doporučuje
    #po několika akordech resetovat pozici motorů
    reset(self.robot.pickMotors)
```

Ke každému souboru SongX.py se musí vytvořit soubor mainX.py. Do něj je importována třída SongX požadované písničky. Soubor mainX.py obsahuje konstruktor robota, ten obsahuje atribut song, který je potřeba přepsat. Přepsání importu a přiřazení písničky jsou jediné dvě změny, které jsou nutné

pro vytvoření nového souboru mainX.py.

Celkem byly přidány 4 nové písničky, jsou jimi: Let it be od kapely Beatles, Knockin' on Heaven's Door od Boba Dylana, Wild Thing od skupiny The Troggs a Mad World od Garyho Julese. Písničky je možno spouštět přímo z kostky zvolením příslušného mainX.py souboru, nebo pomocí nového grafického rozhraní.

```
from songs.Song5 import * <- Import třídy Song5
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import Motor
from pybricks.parameters import *
from pybricks.media.ev3dev import Image, ImageFile, SoundFile
import _thread
import time
class Robot:
    """Class representing robot with all its parts."""
    def __init__(self):
        # Initialize the EV3 brick and motors.
        self.ev3 = EV3Brick()
        self.mainMotor = Motor(Port.A, Direction.CLOCKWISE,
                               [40, 24])
        self.sideMotor = Motor(Port.B, Direction.CLOCKWISE)
        self.chordMotor = Motor(Port.D, Direction.CLOCKWISE,
                                 [[36, 12], [8, 40]])
        self.pickMotors = [self.mainMotor, self.sideMotor]
        self.song = Song5(self) <- Výběr písničky
```

Kapitola 5

Uživatelské rozhraní Guitar GUI

Kapitola se zabývá možnostmi komunikace s kostkou LEGO EV3 MINDSTORMS a implementací uživatelského rozhraní s názvem Guitar GUI. Rozhraní využívá komunikace přes protokol SSH k vytvoření vzdáleného přístupu a ovládání EV3 kostky pomocí UNIX-ových příkazů. Uživatelské rozhraní slouží k výběru písničky a spuštění příslušného programu na kostce.

5.1 Způsoby komunikace a připojení ke kostce

Kostku je možné připojit k počítači třemi způsoby: pomocí mini-USB kabelu, přes Bluetooth nebo Wi-Fi. Pro programování robota je vhodné využít rozšíření vývojového prostředí Visual Studio Code, které je možné doinstalovat přímo v něm. Návod na instalaci je dostupný v dokumentaci verze MicroPythonu pro EV3 LEGO MINDSTORMS(odkaz zde). Při využití rozšíření ve Visual Studio Code nezáleží, jestli je připojení ke kostce přes mini-USB kabel, Bluetooth nebo Wi-Fi. Pro využití Wi-Fi je potřeba připojit ke kostce bezdrátový adaptér a provést konfiguraci. Při modifikaci programu robota byl použit mini-USB kabel, kvůli nejmenší odezvě mezi zařízeními.

Uživatelské rozhraní Guitar GUI využívá k volbě písničky vzdálený přístup ke kostce pomocí SSH protokolu. Vytvořit SSH je možné i v rozšíření pro Visual Studio Code(kapitola Accesing advanced EV3 features v dokumentaci). Po vytvoření se na počítači zobrazí terminál EV3 kostky, který lze ovládat pomocí UNIX-ových příkazů. Spuštění písničky se provede pomocí příkazu:

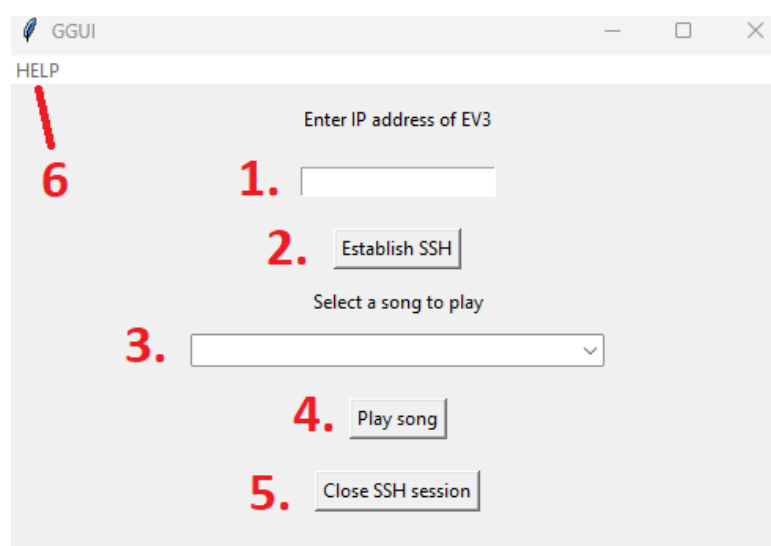
```
#Brickrun -r je příkaz pro spuštění souboru MicroPythonem
#--directory je cesta do adresáře projektu
#Všechny projekty vytvořené v rozšíření
#Visual Studia jsou v adresáři /home/robot
brickrun -r --directory="/home/robot/source" '
"/home/robot/source/main.py" <- spuštěný soubor
```

Pro použití uživatelského rozhraní Guitar GUI je potřeba připojit EV3 kostku k internetu. Pokud používáme Wi-Fi pro komunikaci mezi kostkou a počítačem, není potřeba dělat nic navíc. Jednou z dalších možností je připojit kostku přímo k internetu pomocí ethernet kabelu. Při použití Bluetooth nebo mini-USB kabelu je nutné umožnit kostce se připojit k internetu přes

připojený počítač. Návod k připojení lze najít v záložce Documentation v sekci Networking na stránce <https://www.ev3dev.org/>. Připojení k internetu přes bluetooth není možné pokud na připojeném počítači je operační systém Windows 10. Po úspěšném připojení k internetu se na kostce v levém horní rohu zobrazí IP adresa kostky. IP adresa by měla být ve tvaru 192.168.xxx.xxx. Při sdílení připojení přes mini-USB kabel může nastat chyba, kdy adresa nezačíná 192.168., tehdy je potřeba provést konfiguraci sdílení znova. Jedná se o bug, který se vyskytuje u operačního systému Windows a často se vyřeší restartováním kostky nebo rekonfigurací sdílení připojení.

5.2 Guitar GUI

Uživatelské rozhraní Guitar GUI je napsáno v jazyce Python. K vytvoření rozhraní byla použita standardní Python knihovna tkinter. K vytvoření vzdáleného přístupu přes SSH protokol byla použita knihovna Paramiko (odkaz na dokumentaci zde). Paramiko není součástí standardních pythonovských knihoven a musí se doinstalovat. Pro spuštění Guitar GUI je možno ho nainportovat jako projekt do libovolného pythonovského vývojového prostředí, nebo spustit soubor main.py. Na obrázku 5.1 je popsán postup výběru a spuštění písničky.



Obrázek 5.1: Guitar GUI

- Do pole(1) zadáme IP adresu z levého horního rohu kostky.
- Po zkontrolování IP adresy stiskneme tlačítko Establish SSH(2). Pokud bylo úspěšně vytvořeno připojení objeví se hláška SSH established successfully.
- V dropdown menu(3) zvolíme z předem vytvořených možností písničku, kterou chceme zahrát.

- Po výběru zmáčkneme tlačítko Play Song(4).
- Pokud nechceme již přehrávat další písničky stiskneme Close SSH session(5). Nyní můžeme aplikaci ukončit křížkem vpravo nahoře. V dokumentaci knihovny Paramiko je doporučeno SSH protokol vždy ukončovat. Aplikaci nelze vypnout dokud nebylo SSH manuálně ukončeno.

V levém horním rohu(6) se nachází záložka HELP. Po rozkliknutí se zobrazí nápověda pro ovládání Guitar GUI. V záložce HELP je také odkaz na návod, který obsahuje postup, jak připojit EV3 kostku k internetu. Rozhraní upozorňuje na dodržení postupu spouštění písniček. Pokud nejsou kroky provedeny ve správném pořadí, zobrazí se hlášky upozorňující uživatele na přeskočení kroku.

Guitar GUI bylo vytvořeno pomocí návodů v dokumentaci Paramiko knihovny. Rád bych se zmínil o dvou částech kódu, které knihovnu využívají.

```
#Inicializace třídy SSHClient
client = paramiko.SSHClient()
#parametry pro vytvoření připojení
#Ip adresu získáme metodou get() z pole, do kterého ji
#zadá uživatel
#username EV3 kostky je defaultně nastaven jako robot
#password EV3 kostky je defaultně nastaven jako maker
ip_address = e1.get()
username = 'robot'
password = 'maker'
#Automatické přidání klíče hosta do seznamu klíčů
client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
#Vytvoření připojení
client.connect(hostname=ip_address, username=username,
password=password)
```

Po zmáčknutí tlačítka Play Song se podle zvolené písničky přes SSH pošle příkaz ke spuštění vybraného souboru mainX.py.

```
client.exec_command('brickrun -r'
                    '--directory="/home/robot/source" '
                    '/home/robot/source/main.py''')
```


Kapitola 6

Parsování písniček z předem určeného formátu dat

Poslední kapitola se zabývá možností parsovat písničky do robota z externího souboru obsahující písničku. Běžné formáty pro přenos hudby (MP3, WMA, atd.) obsahují digitální data vytvořené pomocí A/D převodníku. EV3 Kostka je schopná data přehrát, ale to není princip robota. Robot imituje lidského kytaristu pomocí mechanických částí z lega. Robot potřebuje seznam instrukcí, které mu říkají, jaké akordy hrát. Pokud by byl takový soubor dostupný, robot by ho mohl programově přečíst a extrahovat potřebné informace pro hru. Robot by mohl použít .mid soubory využívané protokolem MIDI.

6.1 MIDI protokol

MIDI je protokol využívaný ke komunikaci mezi hudebními zařízeními. Zařízeními mohou být elektrické nástroje, syntezátory nebo zařízení vykonávající řídicí funkci v soustavě. MIDI protokol je možné využít k synchronizaci více nástrojů, vytváření hudby, použití virtuálních nástrojů nebo ovládní živých vystoupení. Pro zaznamenání instrukcí MIDI protokolu jsou využívány MIDI soubory ve formátu .mid. Na rozdíl od formátů pro přenos hudby, MIDI soubor obsahuje instrukce pro konkrétní koncové zařízení. Data v souboru jsou uložena jako proud 8-bitových dat. Pro jejich čtení je možno využít libovolný program, určený k čtení a editování binárních souborů.

6.2 Struktura MIDI souboru

Soubor MIDI se skládá ze dvou hlavních bloků. Každý blok obsahuje hned na začátku informaci o druhu bloku (Header nebo Track) a o jeho délce. Každý soubor má jeden blok, označovaný jako Header. Z něj můžeme extrahovat počet stop, které soubor obsahuje. Například každá stopa může mít přiřazený jiný nástroj a obsahovat instrukce určené pro ten nástroj.

Po header bloku následuje pro každou stopu, její vlastní track blok. Tento blok již obsahuje instrukce pro hraní stopy. Hned za informací obsahující délku bloku se nachází sekvence „MTrk” události, které se dělí do tří kategorií:

MIDI, SysEx a Meta. SysEx události se vztahují přímo k zařízení, které MIDI používá. Pro robota jsou zbytečné a můžeme je ignorovat.

6.2.1 MIDI události

MIDI události obsahují instrukce pro reprodukci zamýšleného hudebního výkonu. Najdeme zde jednotlivé noty, používaný hudební nástroj nebo hlasitost. Hlavní tři instrukce, které by robot mohl využít jsou:

- Note On - Instrukce pro zahrání tónu, po instrukci následují dva byty. První udává tón. Tóny mají přiřazená čísla a můžeme je vyhledat v tabulce na obrázku 6.1. Druhý byte určuje hlasitost. Údaj o hlasitosti můžeme ignorovat, protože robot pro něj nemá využití.
- Note Off - Instrukce pro ukončení tónu. Následují dva byty stejné jako u Note On. Pro každý Note On je ve stopě i jeho Note Off.
- Program Change - Instrukce pro vybrání nástroje stopy. Následuje jeden byte obsahující číslo nástroje. Přiřazené čísla nástroje jsou vidět na obrázku 6.2.

Octave #	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Obrázek 6.1: Tóny a jejich přiřazená čísla. Převzato z [9]

Prog # Instrument	Prog # Instrument	Prog # Instrument	Prog # Instrument
1. Acoustic Grand Piano	33. Acoustic Bass	65. Soprano Sax	97. FX 1 (rain)
2. Bright Acoustic Piano	34. Electric Bass (finger)	66. Alto Sax	98. FX 2 (soundtrack)
3. Electric Grand Piano	35. Electric Bass (pick)	67. Tenor Sax	99. FX 3 (crystal)
4. Honky-tonk Piano	36. Fretless Bass	68. Baritone Sax	100. FX 4 (atmosphere)
5. Electric Piano 1	37. Slap Bass 1	69. Oboe	101. FX 5 (brightness)
6. Electric Piano 2	38. Slap Bass 2	70. English Horn	102. FX 6 (goblins)
7. Harpsichord	39. Synth Bass 1	71. Bassoon	103. FX 7 (echoes)
8. Clavi	40. Synth Bass 2	72. Clarinet	104. FX 8 (sci-fi)
9. Celesta	41. Violin	73. Piccolo	105. Sitar
10. Glockenspiel	42. Viola	74. Flute	106. Banjo
11. Music Box	43. Cello	75. Recorder	107. Shamisen
12. Vibraphone	44. Contrabass	76. Pan Flute	108. Koto
13. Marimba	45. Tremolo Strings	77. Blown Bottle	109. Kalimba
14. Xylophone	46. Pizzicato Strings	78. Shakuhachi	110. Bag pipe
15. Tubular Bells	47. Orchestral Harp	79. Whistle	111. Fiddle
16. Dulcimer	48. Timpani	80. Ocarina	112. Shanai
17. Drawbar Organ	49. String Ensemble 1	81. Lead 1 (square)	113. Tinkle Bell
18. Percussive Organ	50. String Ensemble 2	82. Lead 2 (sawtooth)	114. Agogo
19. Rock Organ	51. SynthStrings 1	83. Lead 3 (calliope)	115. Steel Drums
20. Church Organ	52. SynthStrings 2	84. Lead 4 (chiff)	116. Woodblock
21. Reed Organ	53. Choir Aahs	85. Lead 5 (charang)	117. Taiko Drum
22. Accordion	54. Voice Oohs	86. Lead 6 (voice)	118. Melodic Tom
23. Harmonica	55. Synth Voice	87. Lead 7 (fifths)	119. Synth Drum
24. Tango Accordion	56. Orchestra Hit	88. Lead 8 (bass + lead)	120. Reverse Cymbal
25. Acoustic Guitar (nylon)	57. Trumpet	89. Pad 1 (new age)	121. Guitar Fret Noise
26. Acoustic Guitar (steel)	58. Trombone	90. Pad 2 (warm)	122. Breath Noise
27. Electric Guitar (jazz)	59. Tuba	91. Pad 3 (polysynth)	123. Seashore
28. Electric Guitar (clean)	60. Muted Trumpet	92. Pad 4 (choir)	124. Bird Tweet
29. Electric Guitar (muted)	61. French Horn	93. Pad 5 (bowed)	125. Telephone Ring
30. Overdriven Guitar	62. Brass Section	94. Pad 6 (metallic)	126. Helicopter
31. Distortion Guitar	63. SynthBrass 1	95. Pad 7 (halo)	127. Applause
32. Guitar harmonics	64. SynthBrass 2	96. Pad 8 (sweep)	128. Gunshot

Obrázek 6.2: Seznam nástrojů. Převzato z [8]

6.2.2 Meta události

Meta události slouží k přenosu informací nesouvisejících s přehráním tónů skladby. Poskytují dodatečné informace o skladbě. Existují tři druhy událostí, které by mohl robot využít: TimeSignature, Tempo a End of track.

- TimeSignature - Obsahuje zakódovanou informaci o taktu skladby. Význam taktu byl vysvětlen v kapitole 4.1.
- Tempo - Obsahuje zakódovanou informaci o tempu. Údaj není v jednotkách úderů za minutu (BPM), ale je možné jej přepočítat na tempo v BPM. Význam tempa byl také vysvětlen v kapitole 4.1.
- End of Track - Indikuje konec stopy. Hned po této události následuje další stopa nebo konec souboru MIDI.

6.3 Nevýhody formátu MIDI souborů

Z MIDI souboru je možno extrahovat některé údaje, které byly použity pro vytváření písniček v kapitole 4.1. V MIDI souboru chybí nejdůležitější informace pro robota, sekvence zahráných akordů. V MIDI souboru se objevují pouze informace o zahráných tónech. Zkušený kytarista po zaslechnutí tónu, dokáže tón replikovat podle zvuku, robot ale tuto možnost nemá. Tón by se musel „převést“ na akord. K implementování programu, který by byl schopný tento proces učinit možným je zapotřebí pokročilé znalosti hudební teorie. Soubor MIDI má ještě jeden nedostatek.

MIDI soubory obsahují celý doprovod k písničce, ne jen ten kytarový.

Například MIDI soubor písničky Castle of Glass od kapely Linkin park, kterou robot umí zahrát, obsahuje čtyři různé kytarové doprovody. Jednotlivé stopy je možné od sebe oddělit, tím by vznikly čtyři různé stopy, spadající do kategorie kytarových doprovodů. Z nich by se musela vybrat ta, kterou by robot zahrál.

Z dostupných formátů pro záznam hudby jsou soubory MIDI vhodné pro parsování a vytváření písniček. K implementaci funkčního řešení je potřeba, vymyslet způsob, jak převádět tóny z MIDI souboru na akordy, které umí robot zahrát.

Kapitola 7

Závěr

Cílem práce bylo rozšíření robota hrajícího na kytaru, kterého vytvořila ve své bakalářské práci slečna Petrčková. Robot měl být rozšířen o nové akordy, písničky a grafické rozhraní. Předpokladem pro vytvoření úspěšné práce bylo pochopení robota po mechanické a programové stránce na nejvyšší možné úrovni. Rozšíření o nové akordy a písničky bylo úspěšné, ale ne dokonalé. V popsaném řešení se vyskytuje chyba, které způsobuje občasné nežádoucí tlumení hraných akordů.

K robotickému hráči bylo vytvořeno grafické rozhraní Guitar GUI, které umožňuje uživateli vybírat a spouštět písničky přes počítač. Grafické rozhraní využívá knihovny Paramiko k vytvoření vzdáleného přístupu přes protokol SSH.

V poslední části byla popsána možnost parsování písniček z domluveného formátu. Tato část byla tématem mého samostatného projektu předešlý semestr. Výsledkem projektu byl závěr, že parsování písniček vyžaduje vysokou znalost hudební teorie. V této práci, jsem se z tohoto důvodu zaměřoval na jiné možnosti rozšíření robota. Moje poznatky ze samostatného projektu jsem shrnul v poslední části práce, aby mohly posloužit v budoucnosti, někomu kdo se rozhodne parsování písniček implementovat.

Největší překážkou během tvoření práce byla unikátnost tématu. Robot hrající na kytaru není primární využití stavebnice LEGO MINDSTORMS EV3, neexistuje tedy k tématu mnoho zdrojů. Vytvoření nové osmiakordové konfigurace zabralo nejvíce času, při vytváření bylo potřeba zvážit mnoho kritérií. Na druhou stranu mi unikátnost tématu umožnila rozvíjet kreativitu, představivost a schopnost řešit problémy. Rozhodně jsem za zkušenost vděčný, i za širší témat, kterými jsem se zabýval.

Robota lze nadále rozšířit o výše zmíněné parsování písniček. Dalším rozšířením by mohlo být přepracování mechanismu držícího akordy, tak aby se odstranila chyba v mém řešení.



Literatura

- [1] PETRČKOVÁ, Veronika. Využití robota LEGO Mindstorms EV3 - návrh robota hrajícího na kytaru pro propagaci FEL [online]. Praha, 2022 [cit. 2023-05-24]. Dostupné z: <https://dspace.cvut.cz/handle/10467/101323>. Bakalářská Práce. České vysoké učení technické Fakulta Elektrotecnická.
- [2] TECHNICally possible: Lego Guitar Robot: A Closer Look. YouTube [online]. 2016 [cit. 2023-05-24]. Dostupné z: <https://www.youtube.com/watch?v=EN-7cMjmFv0>
- [3] TECHNICally possible: Little Talks Guitar Cover by Lego Mindstorms EV3. YouTube [online]. 2016 [cit. 2023-05-24]. Dostupné z: <https://www.youtube.com/watch?v=cXgB3llvPHI>
- [4] The LEGO Group. Getting started with LEGO® MINDSTORMS Education EV3 MicroPython [online]. 2020 [cit. 2023-05-24]. Dostupné z: <https://pybricks.com/ev3-micropython/index.html#>
- [5] Paramiko: A Python implementation of SSHv2. [online]. [cit. 2023-05-24]. Dostupné z: <https://www.paramiko.org/>
- [6] EV3DEV [online]. [cit. 2023-05-24]. Dostupné z: <https://www.ev3dev.org/>
Dostupné z: <https://midi.org/specifications/file-format-specifications/standard-midifiles/rp-001-v1-0-standard-midi-files-specification-96-1-4-pdf>
- [7] General MIDI System Level 1. In: The MIDI Association [online]. 1996 [cit. 2023-05-24].
- [8] Standard MIDI Files 1.0. In: The MIDI Association [online]. 1996 [cit. 2023-05-24]. Dostupné z: <https://midi.org/specifications/midi1-specifications/general-midispecifications/general-midi-1/general-midi-level-1-2>
- [9] Standard MIDI-File Format Spec. 1.1, updated [online]. [cit. 2023-05-24]. Dostupné z: <http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html#BMA>

- [10] LEGO PART 32271 Technic Beam 1 x 9 Bent (7 - 3) Thick. In: Rebrickable.com [online]. [cit. 2023-05-24]. Dostupné z: <https://rebrickable.com/parts/32271/technic-beam-1-x-9-bent-7-3-thick/>

- [11] LEGO Medium Stone Gray Beam 3 x 5 x 0.5 Triangle Thin Type 2 (65193 / 99773). In: Wwww.brickowl.com [online]. [cit. 2023-05-24]. Dostupné z: <https://www.brickowl.com/catalog/lego-medium-stone-gray-beam-3-x-5-x-0-5-triangle-thin-type-2-65193-99773>



Příloha A

Materiály v odevzdaném archivu

V příložených souborech je několik složek, které obsahují zdrojové kódy, fotky a videa robota.

Ve složce new robot codes se nachází upravené a nové zdrojové kódy robota.

Ve složce Guitar GUI se nachází grafické rozhraní ke spuštění písniček.

Ve složce media se nacházejí fotky osmiakordové konfigurace a video, ve kterém robot hraje na kytaru novou písničku Wild Thing od kapely The Troggs.

Ve složce printing models se nacházejí modely součástí osmiakordové konfigurace ve formátu .blend a .stl