

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra řídicí techniky FEL

## Návrh robota 'Ludvíka' pro propagační účely fakulty

Návrh humanoidního robota ze stavebnice LEGO,  
jeho naprogramování a možnosti využití  
programovacích jazyků pro LEGO EV3

**Matěj Štětka**

Vedoucí: Ing. Martin Hlinovský, Ph.D.  
Obor: Kybernetika a Robotika  
Květen 2020



## Poděkování

Děkuji mému vudoucímu Ing. Martinovi Hlinovskému, Ph.D. za možnosti a rady, které mi během práce na bakalářské práci dal.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 21. května 2020

## Abstrakt

Cílem této bakalářské práce je stavba humanoidního robota ze stavebnice LEGO Mindstorms EV3 pro propagační účely fakulty FEL a pro spolupráci s PKF - Prague Philharmonia. Při stavbě bylo využito především součástek z LEGO. Robot byl naprogramován v jazycích Lego Mindstorms EV3 Home Edition a Python.

**Klíčová slova:** LEGO, Mindstorms, EV3, Python, Micropython, MindSensors, ev3dev

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

## Abstract

The goal of this bachelors thesis is to build a human-like robot from LEGO EV3 for promotional purposes of faculty and for cooperation with Prague Kid Philharmonia PKF. Most of the parts used to build this robot were from LEGO. The robot was programmed in Lego Mindstorms EV3 Home Edition and in Python.

**Keywords:** LEGO, Mindstorms, EV3, Python, Micropython, MindSensors, ev3dev

**Title translation:** Design of robot named Ludvik for Promotion of the Faculty — Build of LEGO EV3 robot and his options for different programming languages

## Obsah

<b>1 Úvod</b>	<b>1</b>	<b>4 Programování</b>	<b>39</b>
1.1 Motivace	2	4.1 Lego Mindstorms EV3 Home Edition	39
<b>2 Seznámení s LEGO MINDSTORMS EV3</b>	<b>3</b>	4.2 Program robota Ludvíka	40
2.1 Řídicí kostka	5	4.3 Řídicí program v ovladači	42
2.2 Programovací jazyk LEGO MINDSTORMS EV3 Home Edition	6	4.4 Hlavní program	45
<b>3 Stavba robota</b>	<b>9</b>	4.5 Zvuky	49
3.1 Stavba pásů	9	4.6 Nový ovladač	49
3.2 Stavba trupu	13	<b>5 Python</b>	<b>53</b>
3.3 Stavba ramen a paží	18	5.1 EV3DEV	53
3.4 Stavba hlavy	23	5.1.1 Propojení více kostek	54
3.5 Elektronika a kabeláž	24	5.1.2 Změna hostname	56
3.6 Stavba hrudní částí	29	5.1.3 Errory	56
3.7 Změny na robotovi	31	5.1.4 python3/r	56
3.8 Ovladač	37	5.1.5 Bluetooth error 13 (Permission Denied)	57
		5.2 MicroPython	58
		5.2.1 Výhody a nevýhody MicroPythonu	59

5.2.2 Instalace MicroPythonu . . . . .	59
5.2.3 OVLADAČ PSP-nX-V4 . . . . .	60
5.2.4 Multiplexory . . . . .	62
<b>6 Web</b>	<b>65</b>
<b>7 Závěr</b>	<b>67</b>
<b>Literatura</b>	<b>69</b>

## Obrázky

2.1 Kompletní základní set [p1] . . . . .	3	3.9 Nasazení pilířů . . . . .	17
2.2 Krátký spojovací díl [p2] . . . . .	4	3.10 Zpevnění pilířů . . . . .	18
2.3 Dlouhý spojovací díl [p3] . . . . .	4	3.11 Rameno se čtyřmi motory . . . . .	19
2.4 Pásový díl [p4] a rámeček [p5] . . . . .	5	3.12 Rameno se sedmi motory . . . . .	20
2.5 Řídicí kostka EV3 [p6] . . . . .	6	3.13 Loketní kloub . . . . .	21
2.6 Blokový kód pro kontrolu vzdálenosti . . . . .	6	3.14 zápěstí . . . . .	22
2.7 Blokový kód pro regulaci jízdy po čáře [p7] . . . . .	7	3.15 typy násad . . . . .	22
3.1 První verze jedné pásové soustavy	10	3.16 mechanismus zápěstí . . . . .	23
3.2 Test zatížení robota . . . . .	11	3.17 Ukázka hlavy . . . . .	24
3.3 Finální verze pásové soustavy . .	12	3.18 LEGO kabel [p8] . . . . .	25
3.4 První verze nosné konstrukce . . .	13	3.19 Multiplexor . . . . .	26
3.5 Druhá verze nosné konstrukce . .	14	3.20 Zapojení multiplexorů . . . . .	27
3.6 Finální verze pilíře (1) . . . . .	15	3.21 Autobaterie . . . . .	28
3.7 Finální verze pilíře (2) . . . . .	15	3.22 LiIon baterie . . . . .	28
3.8 Plně zatížený pilíř . . . . .	16	3.23 Zásobník na letáky . . . . .	30
		3.24 Zapojení audio portu . . . . .	31
		3.25 Mechanické resetování lokte . . .	32
		3.26 Mechanické resetování hlavy . .	33

3.27 Překroucená osa . . . . .	34	4.14 Ukázka funkce reset . . . . .	48
3.28 Vytisknuté kolo a jeho podpory .	35	4.15 PSP_nX_V4 [p9] . . . . .	50
3.29 První změna reproduktoru . . . .	36	4.16 PSP blok . . . . .	50
3.30 Ovládač . . . . .	37		
4.1 Kód na změnu I2C . . . . .	40		
4.2 Možnosti multiplexorů . . . . .	41		
4.3 Schéma řídicího programu . . . . .	42		
4.4 Komunikační část ovládacího programu . . . . .	43		
4.5 Ovládání pásů . . . . .	43		
4.6 Rozpoznání funkcí . . . . .	44		
4.7 přijmutí signálu . . . . .	45		
4.8 Ovládání pásů (1) . . . . .	45		
4.9 Ovládání pásů (2) . . . . .	46		
4.10 Prázdná funkce . . . . .	46		
4.11 Ukázka funkce (1) . . . . .	47		
4.12 Ukázka funkce (2) . . . . .	47		
4.13 Ukázka plné funkce . . . . .	48		



## Tabulky

2.1 Základní stavební díly .....	4
----------------------------------	---





# Kapitola 1

## Úvod

“Pro mnoho lidí vyvolává slovo robot obraz mechanické bytosti oděné v kovu, zdobené nejrůznějšími blikajícími světly, tlačítky a dokonce i legračním zvukem. Vskutku se takovíto roboti stali postavami ve vědeckých sci-fi příbězích, románech, filmech a televizních pořadech.”[Fri]

Takto si představí robota většina lidí, ale je to opravdu pravda? Na začátku této bakalářské práce by bylo vhodné připomenout, co je robot zač a kde se vůbec tento nápad zrodil. Na začátku 20. století byl velký pokrok techniky a každým dnem se vynalezlo něco nového. Když vynecháme dvě světové války, tak lidstvo v tomto století udělalo největší pokrok v historii, a to nejen ve vědě, ale také v literatuře. V roce 1920 vydal český spisovatel Karel Čapek vědecko-fantastické téma “Rossumovi Univerzální Roboti”, příběh, který se po celém světě stal slavným pod zkratkou R.U.R.

Karel Čapek si v této novele pohrává s myšlenkou tvorby humanoidních bytostí, které by byly stejné jako lidé, ale neměly by city a potřeby. Díky tomu by byly tyto bytosti vhodné pro každodenní práci. V knize se Rossum (majitel továrny) stane uznávaným po celém světě a začne tyto roboty prodávat do celého světa. Nicméně se po několika letech roboti vzbouří a snaží se vyhladit lidstvo.

Do tohoto tématu nebude dále zabrušováno, nicméně z této novely vzešlo slovo robot, které je odvozeno od slova “robota” neboli práce. Toto slovo se během následujících let chytlo ve všech jazycích a ani dnes tomu není jinak. Karel Čapek se tímto dostal do historie lidstva, ale i když slovo zůstalo stejné, roboti vypadají v dnešní době jinak, než si to on představoval. Jak už bylo

řečeno v prvním citátu, tak každý člověk si pod slovem robot představí něco kovového, co svítí a vydává nějaké zvuky. Jako příklad mohu použít první, co si představím já, a to jsou roboti ze Star Wars, C3PO nebo R2D2.

V dnešní době je robotické skoro vše i v normální domácnosti, ale roboti nevypadají jako lidé, nýbrž jsou upraveni tak, aby sloužili co nejlépe účelům dané práce.

## 1.1 Motivace

Zadání této bakalářské práce bylo navrženo v návaznosti na předešlé bakalářské práce Robotu hrajícího na harmoniku [Šr] a Robotu hrajícího na piano[Ně]. Katedra řídicí techniky již dlouho využívá právě LEGO Mindstorms EV3, a to jak k organizacím Robosoutěže[Rob], tak i jako výukový materiál pro studenty 1. ročníku bakalářského programu Kybernetika a robotika. Doposud se v tomto předmětu využívalo starší generace LEGO NXT a programovacích jazyků Lego Mindstorms EV3 Home Edition a NXC. Jelikož katedra plánuje přejít na nový model LEGO EV3, tak je nutno tyto roboty otestovat a doplnit mezery, které by mohly překážet ve výuce. Z tohoto důvodu bude třetí část této bakalářské práce zaměřena právě na využití programovacího jazyka Python pro EV3 a jeho propojení se senzory třetí strany MindSensors[Min]. V tomto propojení a v propojení více kostek LEGO EV3 jsou stále mezery.

Začátkem června roku 2019 bylo domluveno propojení mezi Ing. Martinem Hlinovským, Ph.D a PKF - Pražskou Filharmonií, kde padl první nápad na stavbu robota, který by vystupoval v koncertech PKF dětem.

Stavby tohoto robota jsme se ujali já a můj kolega Bc. Martin Šrámek. Cílem tohoto projektu bylo postavit humanoidního robota vysokého přes 130cm, který by se uměl pohybovat a byl ideálně postaven z dílů LEGO a řízen pomocí kostek EV3.

V rámci tohoto projektu jsme měli s kolegou jasný cíl, ale nebylo jasné, jak bude vypadat finální tvar a jestli takovou masu LEGO kostky vůbec unesou. V rámci práce na projektu jsme měli rozdělenou práci tak, že Bc. Martin Šrámek vymýšlel převážně hardwarové řešení a já softwarové, z tohoto důvodu nebudou probrány do detailu všechny technické části robota, především ty, které nebyly vytvořeny mojí zásluhou.

## Kapitola 2

### Seznámení s LEGO MINDSTORMS EV3

Na stavbu robota bylo využito výlučně dílů LEGO a to především ze stavebnice řady Mindstorms EV3. Základní dva sety, které byly použity pro stavbu robota jsou LEGO MINDSTORMS® CORE EDUCATION SET [Cs] a LEGO MINDSTORMS® Education EV3 Expansion Set [Esb].



Obrázek 2.1: Kompletní základní set [p1]

Jedna tato sada obsahuje:

součástka	počet
řídící kostka	1
velký motor	2
střední motor	2
akumulátor	1
barevný senzor	1
dotykový senzor	2
ultrazvukový senzor	1
gyroskopický senzor	1

**Tabulka 2.1:** Základní stavební díly

Zbylé součásti setu jsou kabely a stavební kostky, z tohoto důvodu budou ze seznamu vynechány. Kompletní seznam dílů je volně přístupný na oficiálních stránkách LEGO. Za zmínku stojí pouze některé specifické součástky a to dva typy spojovacích dílů námi přezdívané jako “písek” z toho důvodu, že se pořád někde sypou. Těmi to díly jsou dlouhý modrý spojovací kolík[kola] a krátký černý spojovací kolík [kolb]. Tyto dvě součástky jsou fundamentální ke stavbě robota. Jelikož se stavebnice z LEGO dílů neshledávají s žádnými háčky nebo lepivými částmi, tak jediný způsob jak spojit vše dohromady je právě pomocí těchto dílů.



**Obrázek 2.2:** Krátký spojovací díl [p2]



**Obrázek 2.3:** Dlouhý spojovací díl [p3]

Dalšími dvěma součástkami jsou základní stavební prvek robota vypadající jako rám okna námi přezdívaný “rámeček” [Rá] a pásový díl[díř].



**Obrázek 2.4:** Pásový díl [p4] a rámeček [p5]

Pro představu, s jednou touto sadou lze postavit robota o velikosti 30x30x30cm. Stavebnice je určena pro středoškolský věk a lze podle návodů sestavit roboty určené k různým účelům. Jako příklad lze uvést robota, který sleduje čáru pomocí světelného senzoru nebo robota který se pomocí ultrazvukového senzoru vyhýbá překážkám a je schopen vyřešit průjezd bludištěm. Je mnoho dalších návodů na různé chodící roboty atd. ale tito roboti už nejsou zcela odlazeni a dle mého názoru nejsou plně funkční.

## 2.1 Řídicí kostka

Hlavním stavebním dílem je EV3 kostka, která je “mozkem” celého robota. EV3 bylo vydáno 1. Zářím 2013 a je to zatím třetí verze programovatelné kostky od LEGO MINDSTORMS. Tento malý počítač byl navržen s ARM9 a využívá 64MB ram. Díky tomu je robot schopen zvládat mnohé úkony a v rámci skutečnosti, že se jedná o “dětskou hračku” je schopen provádět složitější výpočty, které lze využít různými směry. Bohužel není ani v dnešní době tento počítač využit na 100% a to hlavně z důvodu přednastaveného grafického programovacího jazyku. Pro použití všech jeho funkcí je zapotřebí flashnout software na nějaký jiný programovací jazyk jako například Python, C, Java. . . I přes tato omezení lze některé úlohy vyřešit v Lego Mindstorms EV3 Home Edition a to je i cílem stavby tohoto robota.



Obrázek 2.5: Řídicí kostka EV3 [p6]

## 2.2 Programovací jazyk LEGO MINDSTORMS EV3 Home Edition

K zakoupenému setu LEGO Mindstorms je zdarma ke stažení přednastavený grafický programovací jazyk LEGO MINDSTORMS EV3 Home Edition [Esa].

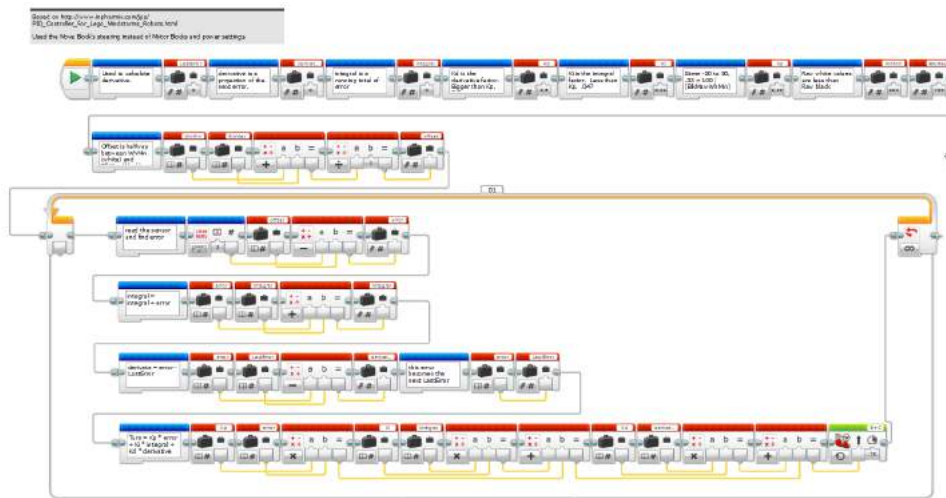
Tento programovací jazyk je určen především pro studenty s nevelkou znalostí programovacích jazyků a je navržen intuitivně.



Obrázek 2.6: Blokový kód pro kontrolu vzdálenosti

Na obrázku [Obrázek 2.6] lze vidět ukázkou základního využití, kde zelené bloky udávají pohyb motorů a žluté vytvářejí podmínky a cykly. Tento program řekne robotovi, aby jel rovně dokud neuvidí před sebou stěnu a poté se zahne směrem vpravo. Díky cyklu okolo celého programu se kód vrátí zpět na začátek a celý proces opakuje dokud není vypnut.





**Obrázek 2.7:** Blokový kód pro regulaci jízdy po čáře [p7]

Dalším příkladem [**Obrázek 2.7**] je program sledující černou čáru regulovaný pomocí PID regulace [PID]. Takovýto program již není zcela intuitivní a uživatel musí mít znalosti s programovacími procesy. Červené bloky se dají vysvětlit jako ALU. Lze s nimi zacházet jako s podmínkami, konstantami nebo i proměnnými. V tomto programu je v prvních dvou řádcích nastaveno několik konstant a proměnných jako jsou odečty chyb ze senzorů a hodnoty PID. Zbytkem programu je funkční cyklus, který si vypočítá okamžitou odchylku robota od čáry a v posledním (a i jediném zeleném bloku) nastaví hodnotu dvou pohybových motorů tak, aby se přiblížil zpět na okraj čáry. Základní princip jízdy robota po čáře zde nebude řešen, jelikož zatím nebyl u Robota Ludvíka vyuzit. Pro více informací lze navštívit tutorial [Tut].



## Kapitola 3

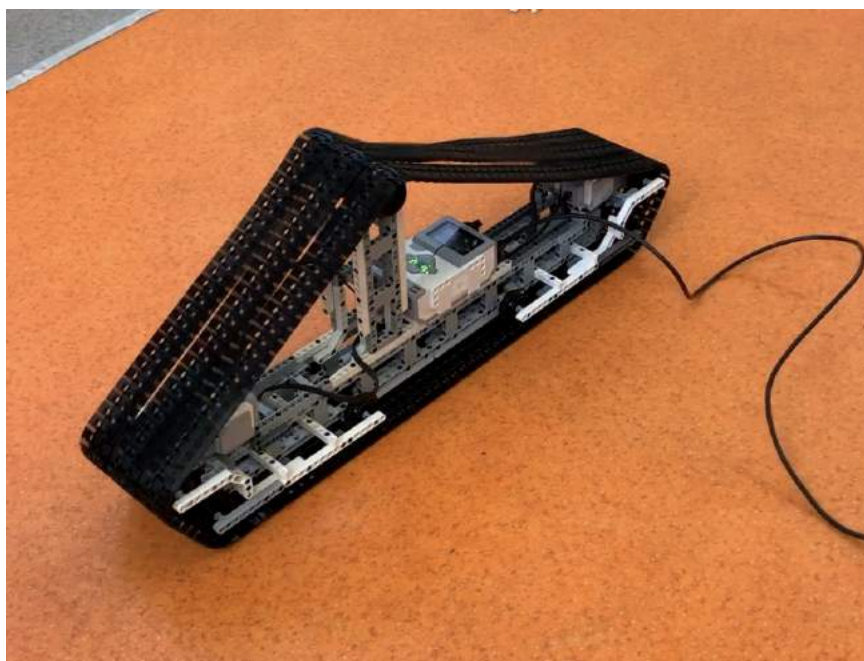
### Stavba robota

Celý robot byl navržen v několika etapách tak, aby splňoval předem dané požadavky. Největším problémem celého projektu bylo vytvoření konstrukce dostatečně robustní na to, aby udržela celého robota a při pohybu se nerozpadla. Jak už bylo řečeno v předchozí kapitole, tak normální velikost robota tomuto projektu zcela neodpovídá. Z tohoto důvodu bylo rozhodnuto začít nejdříve podvozkem a poté stavět trup. Až v poslední řadě paže a hlavu. Tato kapitola je rozdělena po jednotlivých částech stavby v chronologickém pořadí.

#### 3.1 Stavba pásů

První, a dá se říci, že i fundamentální myšlenkou stavby, byl podvozek. Velikost požadovaného robota zapříčinila to, že špatně zapojené LEGO díly by celkovou váhu robota neunesly. Proto bylo rozhodnuto robota postavit na pojízdných pásích, nikoliv na nohách nebo jiných podporách. Dalším důležitým výpočtem bylo vytvoření systému pásů, které by byly dostatečně pevné, ale zároveň byly schopny pohybovat celou hmotností robota odhadovanou na necelých 35 kilogramů.

Velikost pásů a jejich rozvoru byla zvolena tak, aby odpovídala rozložení tanků z druhé světové války, tedy aby rozvor pásů byl stejný jako délka jednoho pásu dotýkajícího se země. Dále bylo zvoleno vytvořit každý pás ze třech pruhů pásových dílů LEGO, aby odpovídal šířce 10 centimetrů. Tento postup se ukázal jako správný pro dostatečnou manévrovatelnost robota.



**Obrázek 3.1:** První verze jedné pásové soustavy

Každý pás byl postaven do tvaru trojúhelníku proto, aby bylo možné upravovat délku pásu pro ideální napětí dílků tak, aby byly stále napnuté, ale nepřetrhávali se. Dále byly do každého pásu vloženy 4 motory EV3, které byly spojeny osou a ozubenými koly tak, aby ovládaly všechny 3 pásy simultánně.

K robotovi byla připojena EV3 kostka a pomocí jednoduchého programu, který ovládal oba motory vpřed bylo testováno zda zvolené řešení bude vhodné a nebo nikoliv. Prvním testem byla pouze jízda rovně po rovné ploše, tento test proběhl bez jakýchkoliv komplikací a proto bylo v druhém testu vyzkoušeno přejetí malé překážky jako je práh u dveří. I tento test proběhl v pořádku. V následujícím kroku byla postavena kopie tohoto pásového systému a otestována na stejných prvcích.

V poslední řadě byly vytvořeny 2 vzpěry, které spojily oba pásové systémy dohromady dle předem daných rozměrových požadavků. Tento spojený systém byl podroben více testům. První dva testy byly stejné jako v prvním kroku a proběhly bez komplikací, dalším testem bylo otočení na místě. Tento pohyb byl pro robota velice důležitý, aby byl schopen projíždět zúženými uličkami, dveřmi atd. Díky správným výpočtům na začátku stavby byla i tato konstrukce schopna splnit daný požadavek a otáčela se na jednom místě bez potřeby pohybu. Posledním testem bylo zatáčení při jízdě, které sice fungovalo, ale byl nalezen problém v tom, že pásy mohly mít rychlost pouze 0% nebo 100% díky tomu byl pohyb robota trhavý. Tento problém šel vyřešit softwarovým

řešením, kde bylo využito přidávání a ubírání rychlosti po 10%.

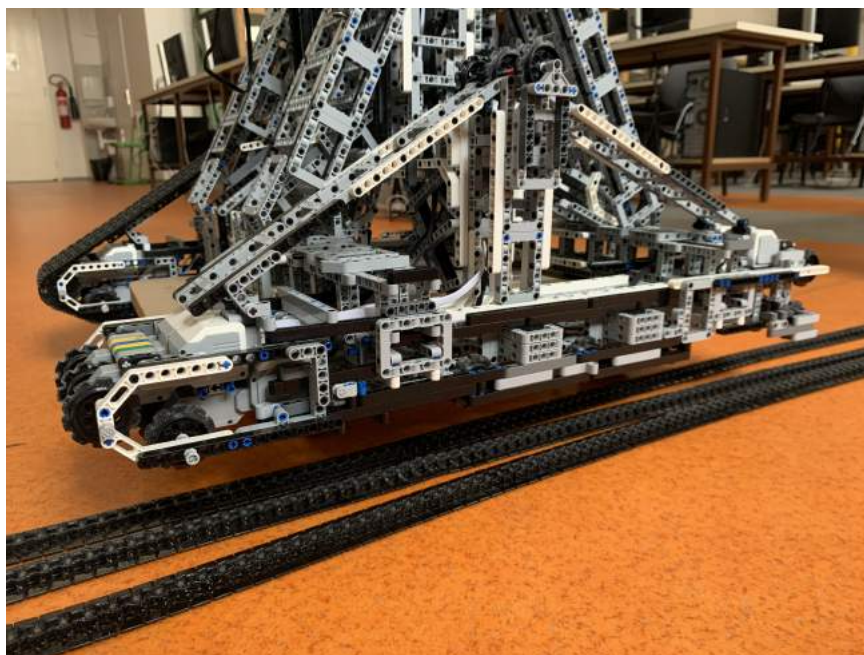


**Obrázek 3.2:** Test zatížení robota

Robot byl těmito testům podroben znovu s 10 kilogramovým závažím. V prvním testu byl zaznamenán problém s rychlostí, kde se robot začal pohybovat až na 30%. V druhém testu bylo zjištěno, že přejezd hrany vyšší než 3 centimetry bude pro robota problémový. Tento fakt spíše jen potvrdil teorii, že robot nebude schopen vyjízďet schody. Třetí test neprokázal žádnou novou změnu. Robot potřeboval více jak 20% výkonu, aby se začal pohybovat

nicméně tato chyba byla nalezena již v testu prvním. V posledním testu se objevila chyba, která z velké části není odladěna ani do dnešní doby a dle mého názoru s ní Robot Ludvík bude muset žít i nadále. Touto chybou je smýkavost části pásů při otáčení. Jelikož má pás určitou šířku, tak se jedna ze stran musí smeknout, aby dohnala zbytek pásu. U tanků se s touto věcí počítá a dalo by se říct, že “jen rozryjí” půdu pod sebou. U robota Ludvíka je problém v tom, že díky této skutečnosti nejde na pásy přidat žádné protiskluzové součástky a díky tomu má na určitých površích problém s jízdou. Většina povrchů byla do určité míry vylazena, nicméně povrch kobercový je pro robota peklem. Je schopen jízdy pouze vpřed a nikoliv do stran, jelikož se do koberce chytanou částí pásu a ten se poté vytrhne. Všechny tyto problémy byly zaznamenány.

Během dokončování stavby bylo zjištěno, že výkon čtyř motorů v pásových systémech je nedostatečný a proto byla konstrukce pozměněna na systém šesti motorů v každém pásu.



**Obrázek 3.3:** Finální verze pásové soustavy

Hlavní úpravou bylo prodloužení pásu a přidání dvou motorů, které zajistily dostatečnou energii k pohybu stroje. Tato rekonstrukce zapříčinila problém s přední hranou pásu, která začala být skoro kolmá se zemí, takže přejíždění větších nerovností nebylo zcela vhodné.

## 3.2 Stavba trupu

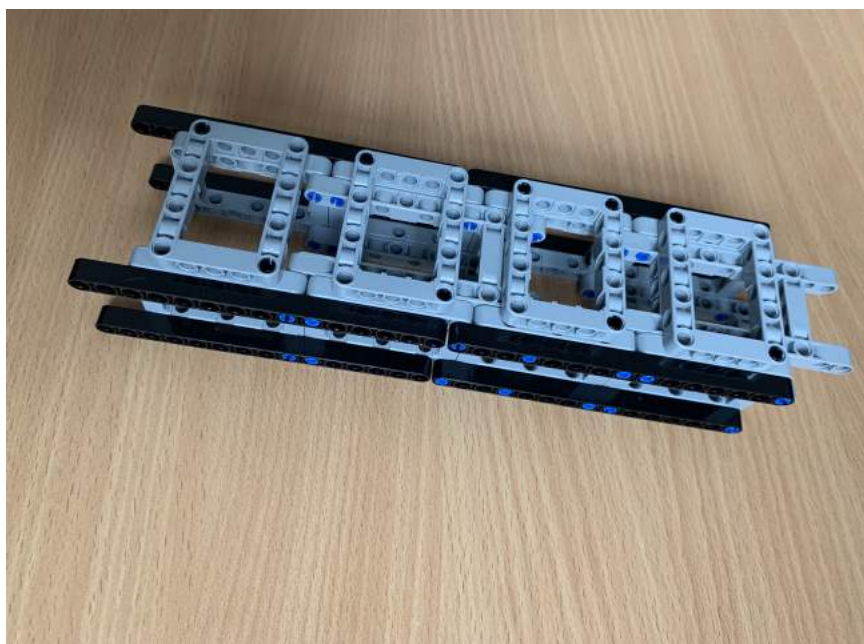
Jelikož byl pohyblivý spodek hotov, tak bylo na řadě vytvoření nosného trupu robota, za možné řešení byla zvolena možnost 2 nosných pilířů zabudovaných do podvozku spojených ve výšce 120 centimetrů.

Tato stavba se potýkala se třemi různými řešeními, než byl vytvořen funkční prvek. První verzí bylo pouze spojení rámečků do čtvercového tvaru. Tato verze se ukázala již po 30 centimetrech délky nedostatečná, nejen že se v rotaci vylamovala, ale její pevnost nebyla nikde blízko robota požadovaných rozměrů.



**Obrázek 3.4:** První verze nosné konstrukce

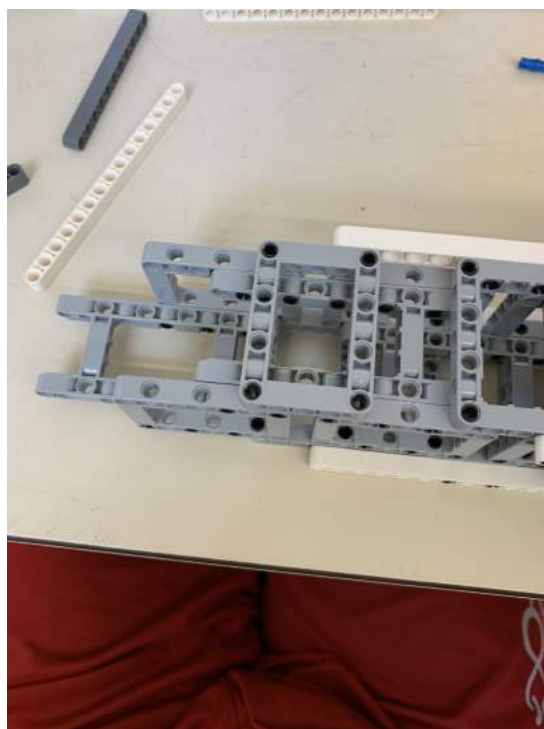
Další verzí bylo vylepšení tohoto systému o druhý plášť tak, aby se tlak na díly vyvíjel protichůdně a tlačily jeden do druhého. Tento systém se zdál funkční a vydržel dostatečný tlak i tah. Nicméně v torzi se opět některé díly vylamovaly a díky této skutečnosti bylo zřejmé, že i tento model bude pouze prototypem.



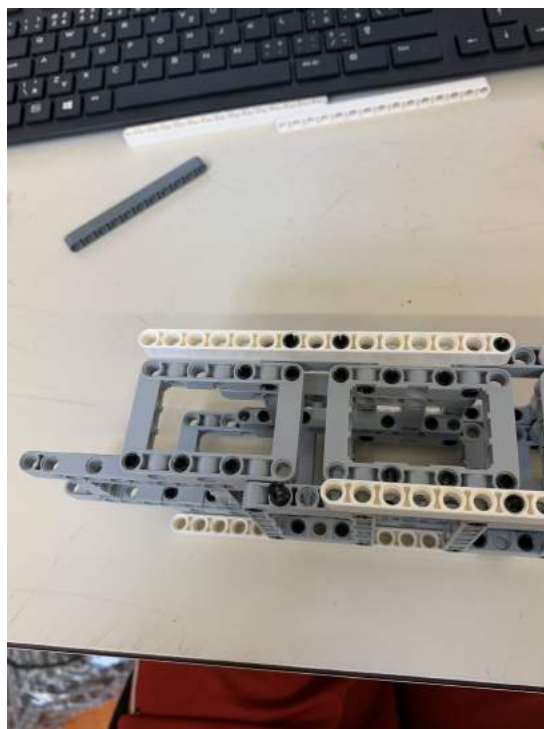
**Obrázek 3.5:** Druhá verze nosné konstrukce

Třetí a finální verze byla vytvořena na více pokusů. Hlavní myšlenkou bylo vytvoření dutého pilíře, tak aby šel nastavovat, ale nemohl se rozpojit nikde ve středu stavby. Po několika neúspěšných pokusech byl vytvořen model dokola se přidávajících dílů, které navzájem tlačily do protichůdných stran, navíc byl tento model opatřen díly tak, aby se jednotlivé části v torzi nevylomily, ale tlačily zpět na kostru robota.



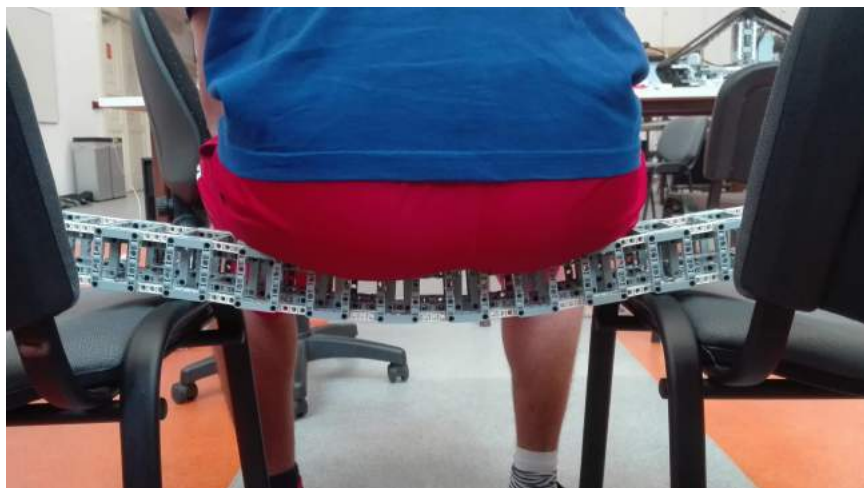


**Obrázek 3.6:** Finální verze pilíře (1)



**Obrázek 3.7:** Finální verze pilíře (2)

Vytvořený model se ukázal jako dostatečný i v metrové délce. Velkou nevýhodou byla nemožnost demontáže ve středních částech, ale tento fakt nijak neovlivnil cíl stavby. Model byl podroben několika testům tlaku, tahu i torze a ve všech přešel. Posledním testem bylo položení metr dlouhého pilíře mezi dvě židle a položením co největší hmotnosti na jeho střed. Celý pilíř obstál i váze sedícího člověka.



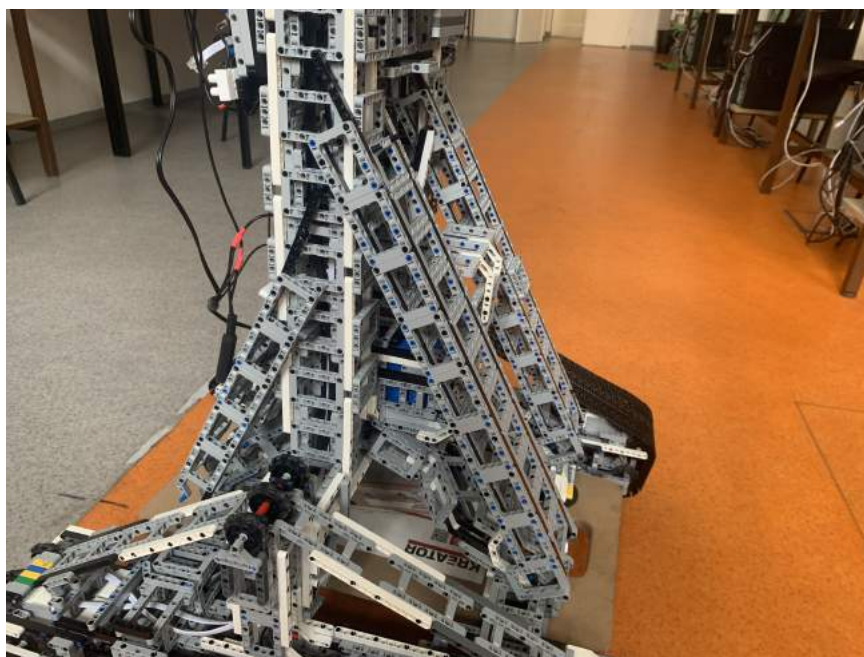
**Obrázek 3.8:** Plně zatížený pilíř

Tento pilíř byl sestaven dvakrát a stejným způsobem byl spojen s prvním modelem ve výšce 120 centimetrů tak, aby připomínal trup a ramena robota. Nadále byl uchycen do podvozku.



**Obrázek 3.9:** Nasazení pilířů

Tato vytvořená konstrukce byla podrobena stejným čtyřem jízdním testům. Hlavním zjištěným problémem byl kyv celé konstrukce. Z tohoto důvodu byly přidány šikmé vzpěry do podvozku tak, aby zamezily kyvu celého robota. Tento problém nebyl zcela eliminován, ale byl poznamenán a později doladěn softwarovým řešením regulace rozjezdu a brzdění.



**Obrázek 3.10:** Zpevnění pilířů

Skutečnost kyvu robota bohužel zamezila tvorbu jakéhokoliv odpružení podvozku od robota, avšak zatím nebyl nalezen problém, kde by se robot díky tomuto nedostatku trápil.

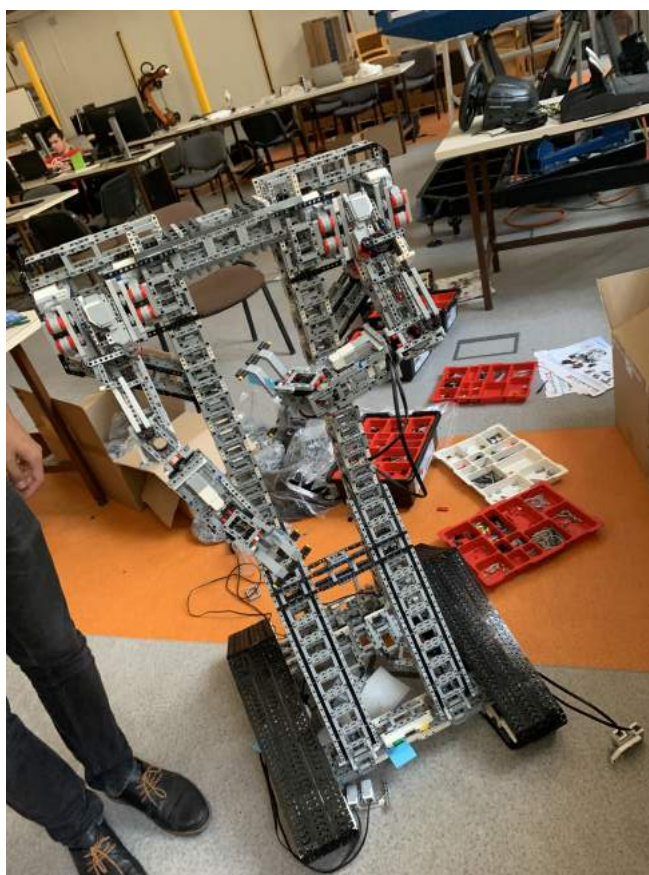
### 3.3 Stavba ramen a paží

Tvorba ramen a rukou byla vytvořena na tři různé verze. V této části musím předat veliký kredit Bc. Martionovi Šrámkovi za jeho geniální sestavení jednotlivých kloubů tak, aby byl robot schopen pohybovat končetinami všemi směry. Z tohoto důvodu bude napsán postup stavby a vysvětleny změny postupně u jednotlivých kloubů, nicméně nebude zacházeno nikterak do detailů.

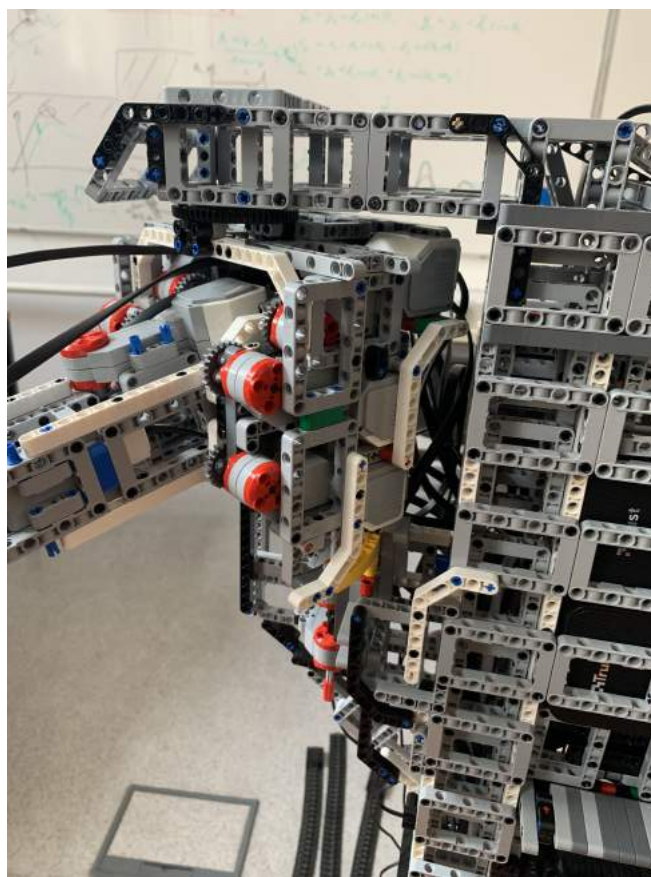
Cílem pohybu rukou bylo zvednutí a otočení v ramenu, zvednutí lokte a rotace zápěstí společně s mechanismem chycení předmětů.

Každý LEGO motor velký má sílu pouze 47 newtoncentimetrů. Z tohoto důvodu bylo nutné navrhnout celou paži tak, aby byla co nejlehčí. Hlavní myšlenkou bylo přidání co nejvíce motorů do ramenního kloubu tak, aby

nezatěžovaly paži při zdvihu. Prvním pokusem byl systém čtyř motorů, které šly protichůdně a rameno se mohlo zvedat pouze vpřed. Tento systém byl sice funkční, ale opíral se o dva nedostatky. Prvním z nich byl stále nedostatečný výkon a paže nešla zvednout výše než do horizontální pozice, druhým problémem byla nemožnost jakékoliv rotace paže do stran. Robot s tímto návrhem sice nějaký čas fungoval, ale nakonec bylo zvoleno řešení, kde v rameni fungovalo šest motorů na zdvih paže a jeden motor otáčel celou paží do stran. Zvolené řešení bylo založeno na technickém LEGO dílu Gear Rack[GR]. Díky těmto kolejničím se celý kloub otáčí na jednom místě a nijak nezatěžuje zbytek paže.

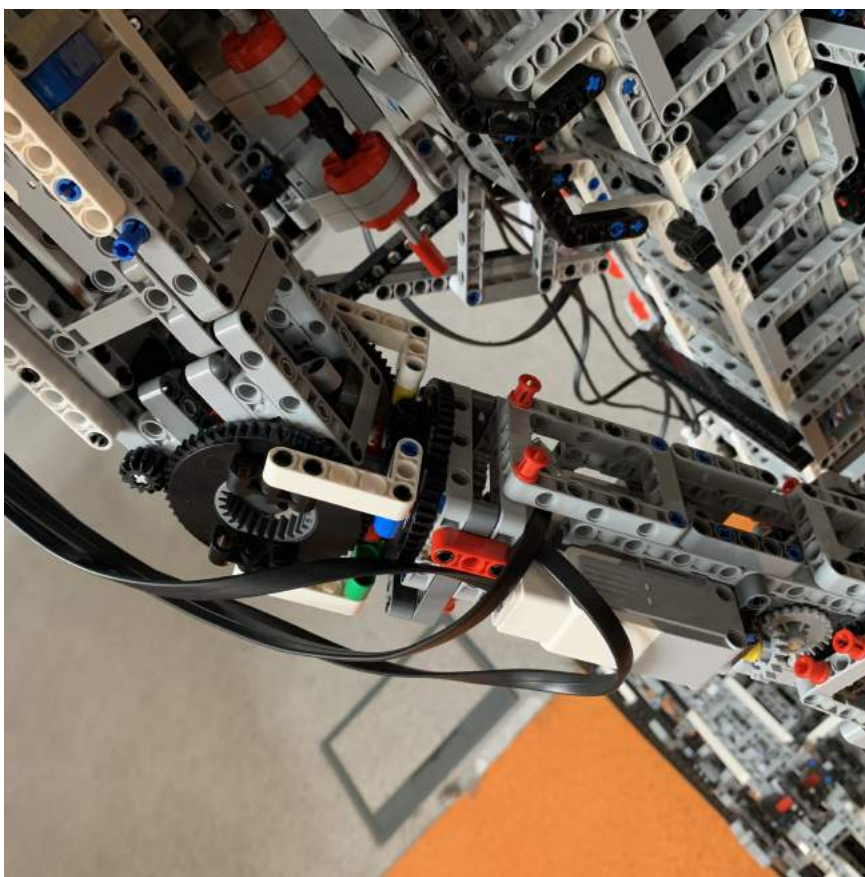


**Obrázek 3.11:** Rameno se čtyřmi motory



**Obrázek 3.12:** Rameno se sedmi motory

V rámci loketního kloubu bylo rozhodnuto využití dvou pohybů. Ohnutí lokte a rotace předloktí. Ohnutí bylo sestaveno tak, aby byla váha systému zatížena co nejbližší rameni. Bylo využito hřídele a velký motor byl usazen v ramenním kloubu. Otočení předloktí bylo postaveno na rotaci středního motoru přes dvojitý ozubený díl[día], který je vhodný právě pro tento účel, z důvodu pevnosti celého kloubu a možnosti rotace o 360 stupňů.

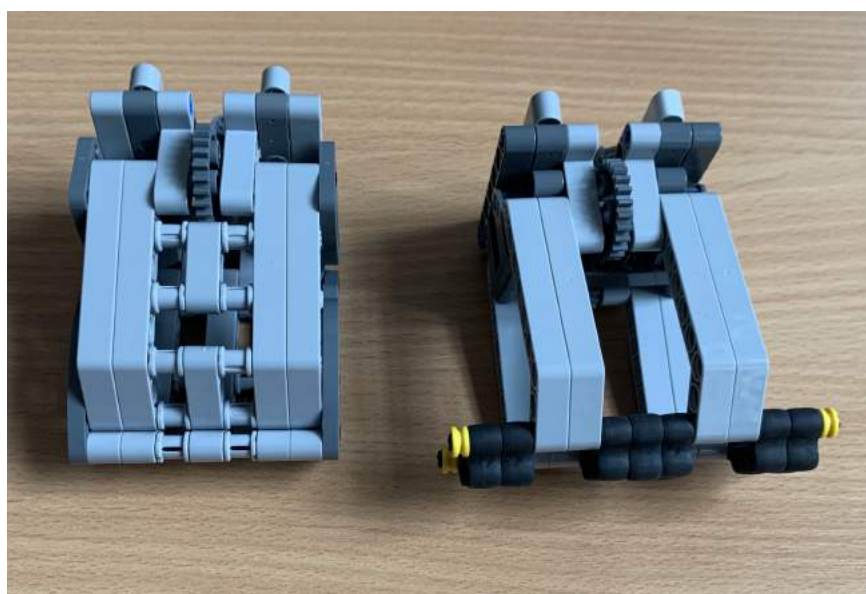


**Obrázek 3.13:** Loketní kloub

Finální částí robota je zápěstí, které bylo postaveno tak, aby bylo možno měnit nástavce tak, aby sloužily co nejlépe právě danému účelu. Hlavní tvar ruky jsou dvě protichůdné části složené z L dílů LEGO, tak aby bylo možné sevřít nějaký předmět. Mechanická část je vyřešena středním LEGO motorem, který otáčí osu na které je šnek. Ten otáčí protichůdně dvěma ozubenými koly, která už pak otevírají a nebo zabírají ruku.



Obrázek 3.14: zápěstí



Obrázek 3.15: typy násad

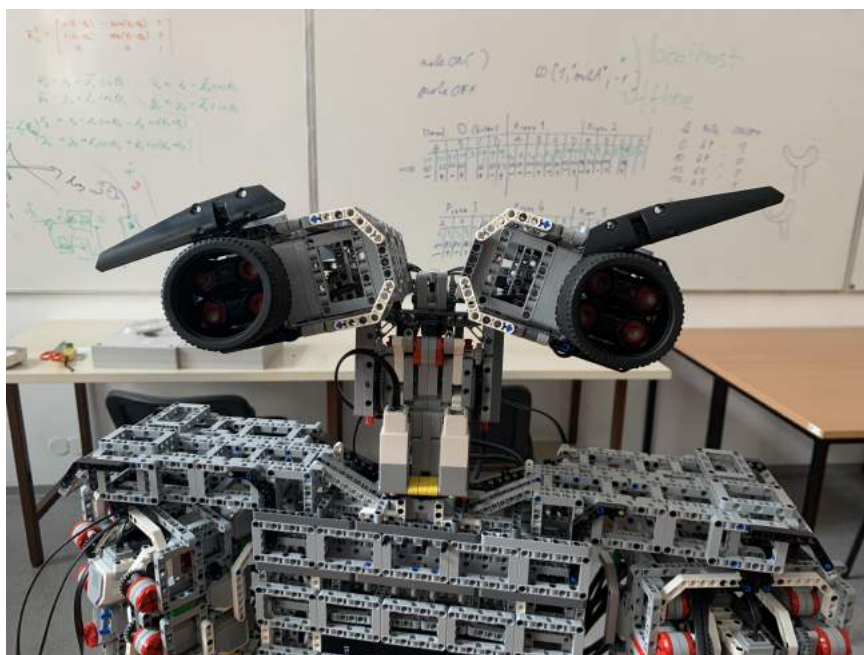




**Obrázek 3.16:** mechanismus zápěstí

## ■ 3.4 Stavba hlavy

Hlava byla vytvořena za účelem vyjadřování co největší lidskosti. Při vymýšlení tvaru bylo diskutováno několik možností tvarů a to včetně tvarů pusy očí a jiných částí. Nakonec bylo rozhodnuto, že robot nebude mít ústa jako taková a na místo tomu bude nad oči přidáno obočí. Tento návrh se vyplatil hlavně z důvodu, že lidské emoce jsou složité napodobit a proto byl přechod pouze na hýbání obočí a hlavou dobrým krokem. Při stavbě hlavy bylo dbáno na to, aby uměla základní rotace vlevo, vpravo tak i vzhůru a dolů. Dalšími dvěma pohyby je právě pohyb obočí a také pootočení části hlavy. Díky těmto funkcím dokáže robot vyjádřit základní emoce, jako štěstí, zlobu, mrzutost...



Obrázek 3.17: Ukázka hlavy

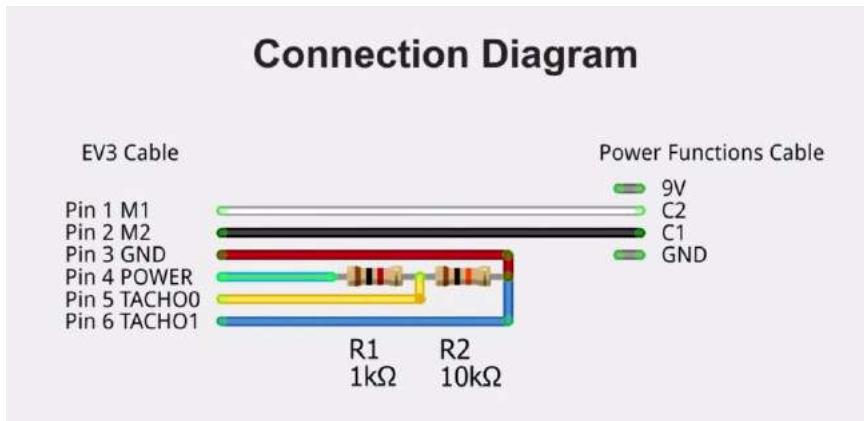
### 3.5 Elektronika a kabeláž

LEGO Mindstorms EV3 se opírá o fakt toho, že na každé kostce je pouze 11 portů.

Mezi těmito porty jsou microUSB na vstup do počítače, poté USB na daisychain (Daisychain je propojení více elektronických prvků za sebou. Více na Daisy Chain[Dc], dále vstup na SDkartu a nakonec dvakrát čtyři LEGO porty. Horní čtyři jsou na ovládání motorů a spodní čtyři jsou na ovládání senzorů. Z tohoto faktu vyplývá, že jedna kostka dokáže řídit pouze malého robota.

Robot Ludvík má v celé své kráse 36 motorů a 8 senzorů. Díky této komplikaci bylo zapotřebí vymyslet takovou variantu, aby robot fungoval. Naskytlo se několik možností. První z nich byla možnost daisychainu kostek za sebou pomocí usb. Toto řešení bylo zamítnuto z důvodu maximální možnosti propojení, která je pro LEGO EV3 na 4 kostkách. Další možností bylo využití bluetooth a komunikace všech kostek. I toto řešení bylo zavrhnuto hned ze dvou důvodů. Prvním z nich je nedměrná cena potřebných kostek a druhou je nedopracovanost grafického EV3 programu v bluetooth komunikaci.

Dalším nápadem bylo rozdvojení kabelů. Bylo otestováno několik možností. Při rozdvojení kabelu se sice motory otáčely, ale na různé strany a pod slabým výkonem.



Obrázek 3.18: LEGO kabel [p8]

Jak si lze z obrázku [Obrázek 3.18] výše povšimnout, tak každý LEGO kabel má v sobě jak pohon tak i zpětnou vazbu. Z tohoto důvodu nebylo možné regulovat 2 motory jedním kabelem. Navíc u senzorů tento mechanismus nemohl fungovat vůbec, jelikož bez I2C komunikace nemohou senzory fungovat.

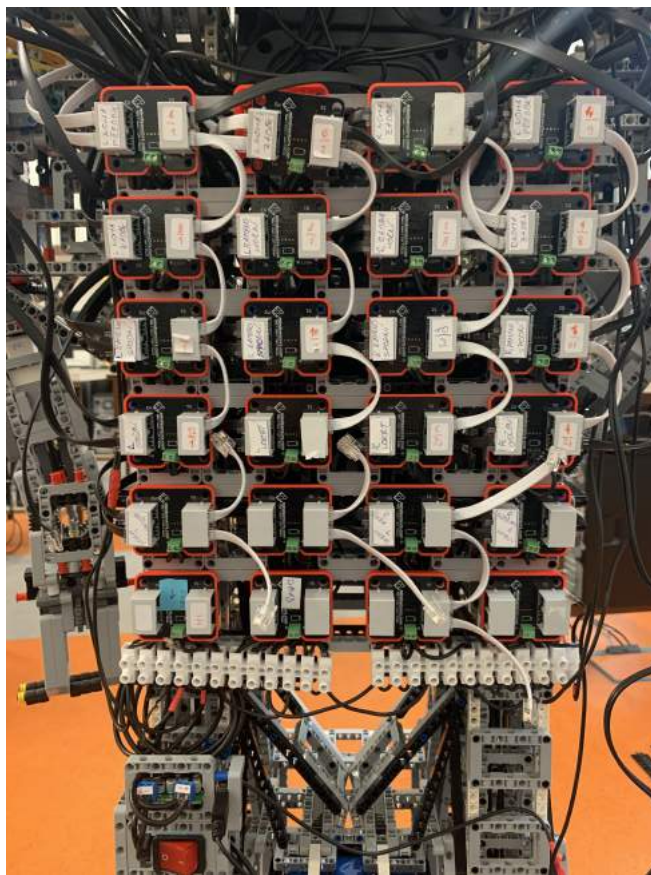
V rámci testování bylo také zjištěno, že kostka předává do motorů nedostatečnou energii a při plném výkonu všech čtyřech motorů do jedné paže není kostka schopna celou paži zdvihnout.

Po pečlivém probrání všech možností bylo rozhodnuto využití součástek LEGO třetí strany z webových stránek MINDSENSORS. Tento výrobce se specializuje na elektronickou část pro LEGO MINDSTORMS a vytváří různé vylepšení použitelné na robotech. Mezi tyto vylepšení patří například, teplotní čidla, kamery, nebo právě námi potřebné multiplexory. Pro stavbu robota bylo nakonec rozhodnuto využití 24 multiplexorů[Mul].



**Obrázek 3.19:** Multiplexor

Velikou výhodou je externí napájení každého multiplexoru zvlášť. Tato věc umožnila dostatečný výkon ve všech končetinách robota. Tento multiplexor funguje pomocí I2C komunikace a lze mu přednastavit vlastní adresu a použít ho na daisy chain. V rámci adres jednotlivých multiplexorů bylo zjištěno, že největší možné číslo zapojených multiplexorů za sebe je šest. Z tohoto důvodu byla na robota využita jedna kostka která ovládá na každém svém portu šest multiplexorů.



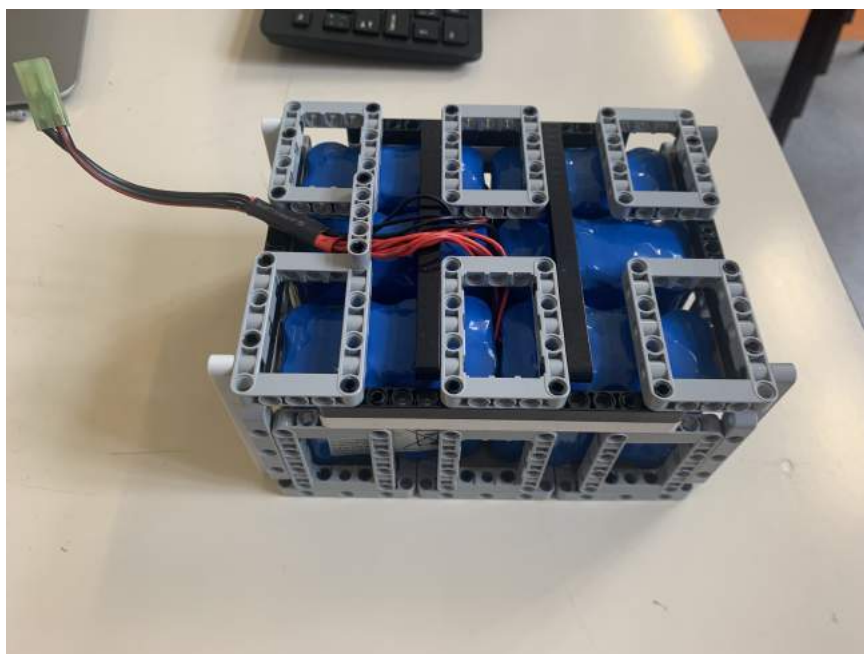
**Obrázek 3.20:** Zapojení multiplexorů

Pro celkové zapojení bylo zapotřebí sehnat baterii, která by napájela nejen EV3 kostky, ale i samotné multiplexory. V první verzi byla využita autobaterie[Aut], která byla umístěna mezi pásy pod robota, tak aby měl robot těžiště co nejnižší.

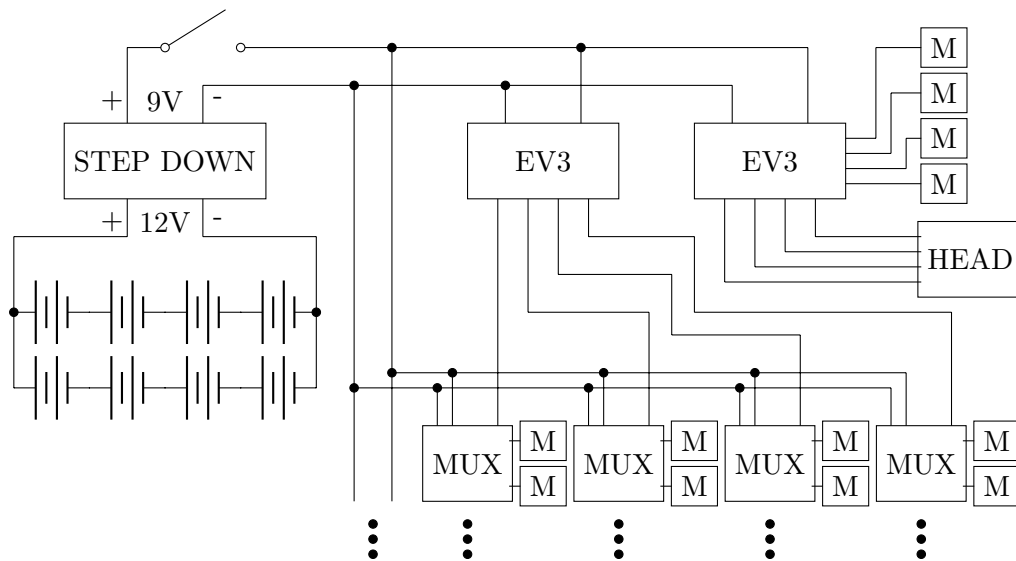


Obrázek 3.21: Autobaterie

V dnešní době robot využívá LI-ION baterie[LiI] baterie, které mají větší výdrž a navíc jsou podstatně lehčí, což robotovi umožňuje využití větší rychlosti.



Obrázek 3.22: LiIon baterie



Na schématu robota lze vidět řešení, které bylo využito. Na baterii je umístěn ON/OFF switch a nad ním je stepdown, který převádí 12V z autobaterie na 9V potřebných pro robota. Dále je stepdown připojen na dvě svorkovnice, které paralelně rozvádí napětí do všech multiplexorů a kostek. Zároveň je možné na schématu vidět jednotlivé propojení multiplexorů v daisy chainu.

### 3.6 Stavba hrudní části

Poslední částí stavby bylo vytvoření podávacího mechanismu letáků, neboť v rámci požadavků na robota bylo nutno vytvoření stroje, který bude schopen komunikovat s okolím a rozdávat letáky. Tento mechanismus byl postaven na principu dvou gumových kol, která se přitlačí na jednotlivé papíry a lehce jsou vysunuty z hrudníku. Robot si poté do hrudníku sáhne a rukou si papír přichytí.



**Obrázek 3.23:** Zásobník na letáky

Pod tento mechanismus byly přidány reproduktory Leto Compact Trust 2.0[Let] pro lepší zvučnost robota obzvlášť pokud bude říkat souvislou řeč. Problémem EV3 je, že na audio nemá žádný výstup. Proto byla řídicí kostka rozebrána a na místo původního malého reproduktoru byl připájen vstup na audio jack 3.5.

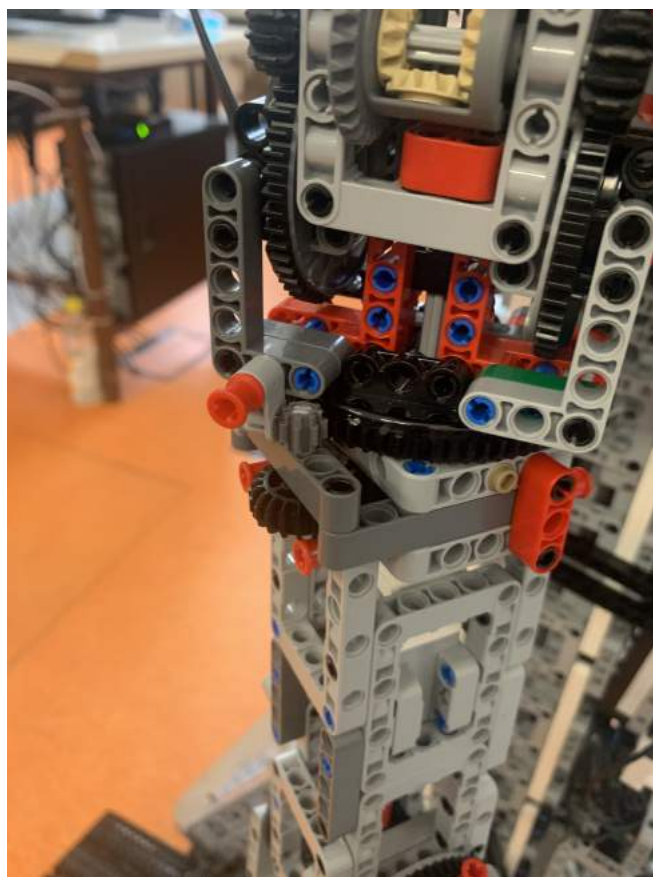




Obrázek 3.24: Zapojení audio portu

## ■ 3.7 Změny na robotovi

Během několika koncertů PKF, které Ludvík absolvoval bylo zjištěno mnoho nedostatků. Některé byly již zmíněny (např. baterie). V této práci nebudou uvedeny veškeré změny, nýbrž jen ty podstatné. Jednou z nich je přidání kulaté LEGO součástky do loketního kloubu, které je využíváno jako zarážky pro kalibraci.



**Obrázek 3.25:** Mechanické resetování lokte

Dále byly k hlavě přidány dva LEGO díly tvaru L, tak aby se hlava nepředklonila příliš a mohla se zvednout zpět. U tohoto problému bylo zjištěno, že velký LEGO motor zvedající hlavu nemá dostatečný výkon na plynule proběhnutí zdvih hlavy.



**Obrázek 3.26:** Mechanické resetování hlavy

Po prvním vystoupení robota Ludvíka bylo zjištěno, že původně sestrojený podvozek celý nápor robota nezvládá a celý se rozpadá. Z tohoto důvodu bylo vytvořeno nové vyztužení podvozku a v rámci této přestavby byly předělány i pásy robota. Některá kola, která poháněla pás měla nejspíše při funkci takovou sílu, že dokázala nejen ohnout, ale i přelámat standardní LEGO OSY.



**Obrázek 3.27:** Překroucená osa

Tento problém byl fatální. V rámci řešení nových možností bylo navrženo využití kovové středové osy. Tento nápad byl, ale zavržen z důvodu trvalé deformace, již ne na ose, ale na motoru samotném. Z tohoto důvodu bylo využito nových kol, která byla umístěna do kritických bodů pásů. Samotné osy bohužel ani tuto změnu nevydržely. Finálním řešením bylo využití 3D tisku.

Ozubená kola pohánějící pás byla vymodelována tak, aby do nich bylo možné vložit ložisko a poté byla nasazena na LEGO konstrukční díl délky 15. Toto řešení bylo dostatečné a je využíváno dodnes.



**Obrázek 3.28:** Vytištěné kolo a jeho podpory

Finální změnou bylo vyměnění reproduktorů za jiný model, neboť zvuk vycházející z původně nainstalovaných neodpovídal potřebné síle zvuku v plně obsazené a zahlučené místnosti. Zvolený reproduktor Arys Compact SET[Ary], byl vybrán především díky jeho rozměrům. S výškou 182 milimetrů přesně zapadá do trupu robota a jeho 12W stačí na momentální potřebu robota.



Obrázek 3.29: První změna reproduktoru

## 3.8 Ovládač

První verze robota byla založena na dálkovém ovladači, který využíval dvě EV3 kostky v daisy chainu a to pouze z důvodu možnosti přidání dalších čtyřech tlačítek. Tato stavba byla využita v roce 2019 a nyní je nahrazena upraveným PSP ovladačem (Viz kapitola programování). Celkovou myšlenkou ovladače, bylo vytvoření něčeho co bude mít hodně tlačítek pro testování různých funkcí a nebylo nikterak komplikované na stavbu. Z tohoto důvodu bude uveden pouze obrázek.



Obrázek 3.30: Ovládač





## Kapitola 4

### Programování

Hlavní částí této bakalářské práce je programování pro LEGO MINDSTORMS EV3. Tato kapitola je rozdělena na programování v Lego Mindstorms EV3 Home Edition (v kterém byl doteď Robot Ludvík programován) a následně je se psáno shrnutí všech dalších možností a jejich realizace.

#### 4.1 Lego Mindstorms EV3 Home Edition

Co se týče Lego Mindstorms EV3 Home Edition, tak bude v první řadě shrnuto několik jeho výhod a nevýhod.

Mezi velké výhody tohoto programu patří především grafické zpracování a jeho využití. Uživatel nemusí mít nikterak velké znalosti z programování a i přesto dokáže naprogramovat funkčního robota. Navíc je program velice intuitivní, alespoň pro základní funkce. Veškeré segmentování do jednotlivých bloků je přehledné a snadno dekodovatelné. Použití vícevláknových funkcí je naprosto triviální a uživatel nemusí mít vůbec ponětí, jak ve skutečnosti tyto procesy probíhají. Díky tomu je tento program skvělým stavebním programem pro začínající robotiky, kteří se snaží proniknout do tajů stavby a programování autonomních robotů.

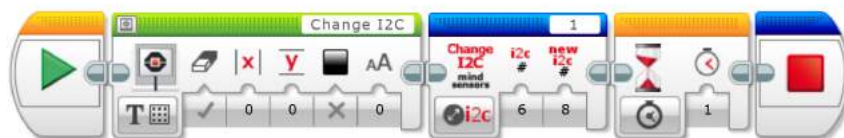
Nevýhod se v tomto programu bohužel najde více a to především pro pokročilejší programátory. Největším problémem je zatížení grafické karty u

větších a obsáhlejších programů. Pokud se uživatel snaží naprogramovat něco složitějšího, jako například právě Robota Ludvíka, program přestává velice rychle stíhat a zasekává se. Velice často se stává, že celý program zamrzne a uživatel je nucen některé části psát znovu. Pro tuto realitu funguje pouze jediné řešení a to je neustálé ukládání rozdělané práce. Druhou velikou nevýhodou je nepřehlednost větších programů. Software má sice možnost ukládání segmentů do jednotlivých bloků což je to samé jako v textových programovacích jazycích volání funkci, nicméně tento postup opět “zalagovává” program rychleji. Na druhou stranu pokud chce uživatel napsat něco jako například PID regulaci, tak se počet použitých bloků zvyšuje takovým způsobem, že se v kódu lze velice rychle ztratit. S tím souvisí fakt, že lze vkládat komentáře přímo do kódu, ale tyto komentáře se nedají přichytit k jednotlivým blokům a díky tomu se stane, že jakákoliv změna v kódu posune bloky jinam a komentáře už nejsou relevantní. Všechny ostatní problémy už souvisí především se všemi výše popsányými.

## 4.2 Program robota Ludvíka

Robot je rozdělen na dva programy. Prvním z nich je program CONTROL, který je nahraný v ovladači robota a druhým kódem je MAIN, který je přímo v robotovi. Tyto dva kódy spolu komunikují pomocí bluetooth rozhraní.

Jednotlivé motory jsou ovládány pomocí externích multiplexorů, které mají přednastavené EV3 bloky, které byly naimportovány do Lego Mindstorms EV3 Home Edition. Multiplexor je naprogramován tak, že komunikuje pomocí I2C protokolu. Ten funguje na sériové komunikaci. Pro více informací I2C[I2C]. Všechny multiplexory mají nastavenou stejnou adresu a proto je k použití nutné ji změnit. Pro tento účel je už připraven na stránkách MindSensors kód[kó], který přepíše jednotlivé adresy [Obrázek 4.1].

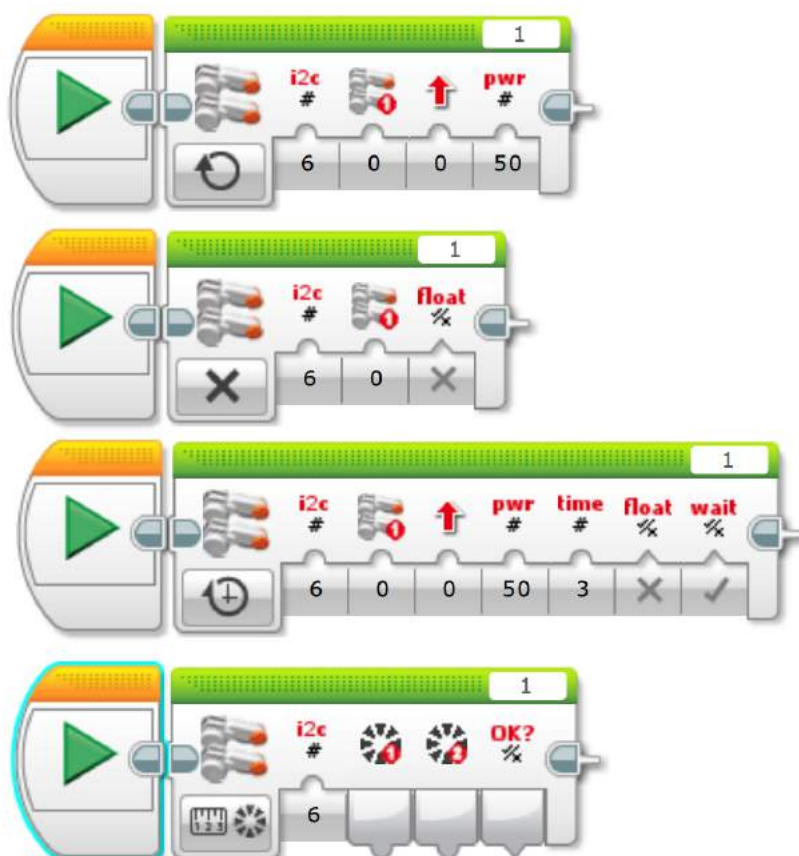


Obrázek 4.1: Kód na změnu I2C

V modrém bloku lze jednoduše napsat původní adresu I2C a poté novou adresu. Nevýhodou je, že multiplexor na jiné adrese nelze najít a proto byl

vytvořen program, který otestuje všechny adresy a vrátí všechny připojené. Tato funkce byla využita při nastavování, jelikož ne vždy změna I2C fungovala.

Následně použitý multiplexor má hned několik možností [Obrázek 4.2].

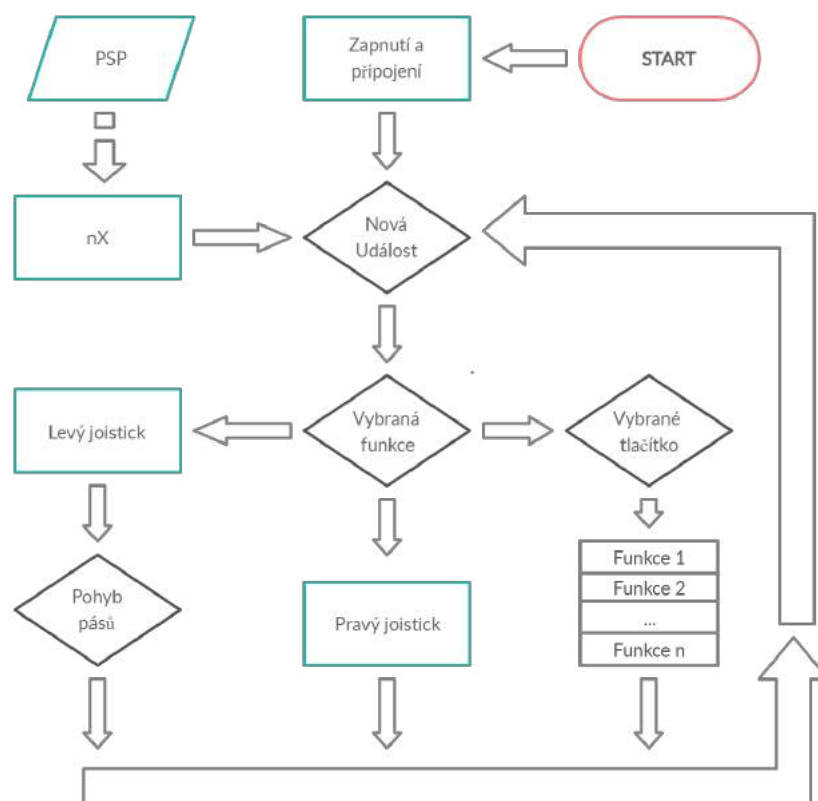


Obrázek 4.2: Možnosti multiplexorů

Jako příklad budou vysvětleny čtyři z nich. První funkce byla použita na robotovi nejvíce, jedná se pouze o příkaz pohybu motorů na dané adrese. Druhou funkcí je zastavení motorů opět na zadané adrese. Zbylé dvě funkce nebyly využity, ale lze s nimi odečítat ze senzorů a jiným způsobem je řídit.

### 4.3 Řídic program v ovladači

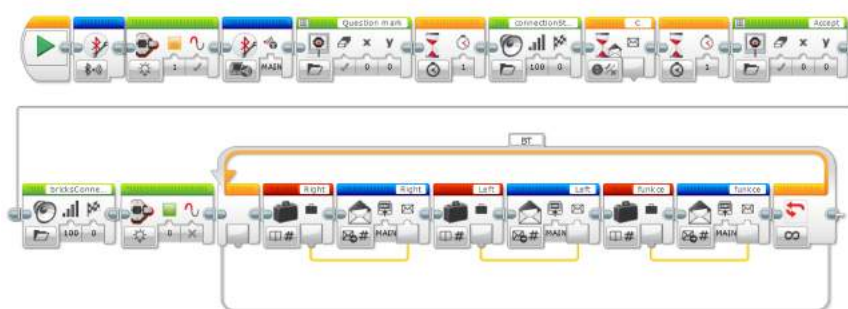
Kód v ovladači byl vytvořen ve dvou verzích, prvním kódem bylo využití dvou EV3 kostek v daisy chainu a využití osmi tlačítek.



**Obrázek 4.3:** Schéma řídicího programu

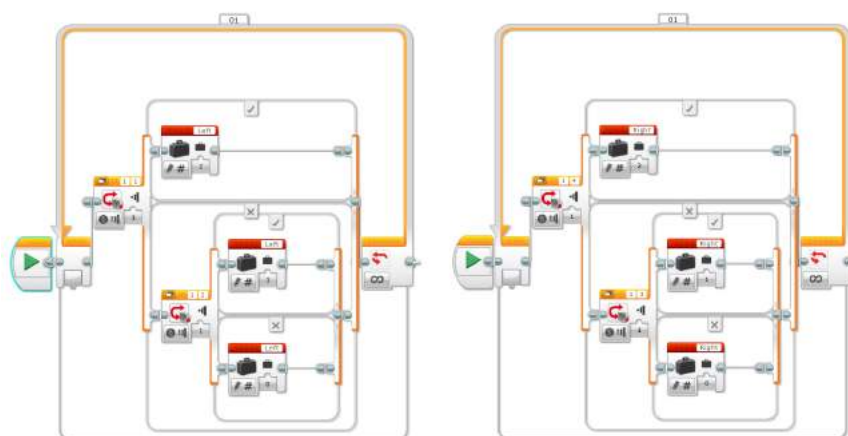
Schéma celého programu [**Obrázek 4.3**] bylo založeno na obecné funkčnosti robota. Program si v každém cyklu odečte hodnoty z ovladače, které poté použije pro rozhodnutí zda má robot někam jet a nebo splnit zadanou funkci.

První vytvořenou částí byla bluetooth komunikace. Lego Mindstorms EV3 Home Edition má přednastavené knihovny pro bluetooth, kde stačí pouze zadat jaký typ chce uživatel poslat jako například INT/STRING/LOGIC a poté stačí napsat jméno druhého EV3 se kterým chce komunikovat. Pro realizaci tohoto kódu je nutné mít zapnutý bluetooth rozhraní na obou kostkách.



**Obrázek 4.4:** Komunikační část ovládacího programu

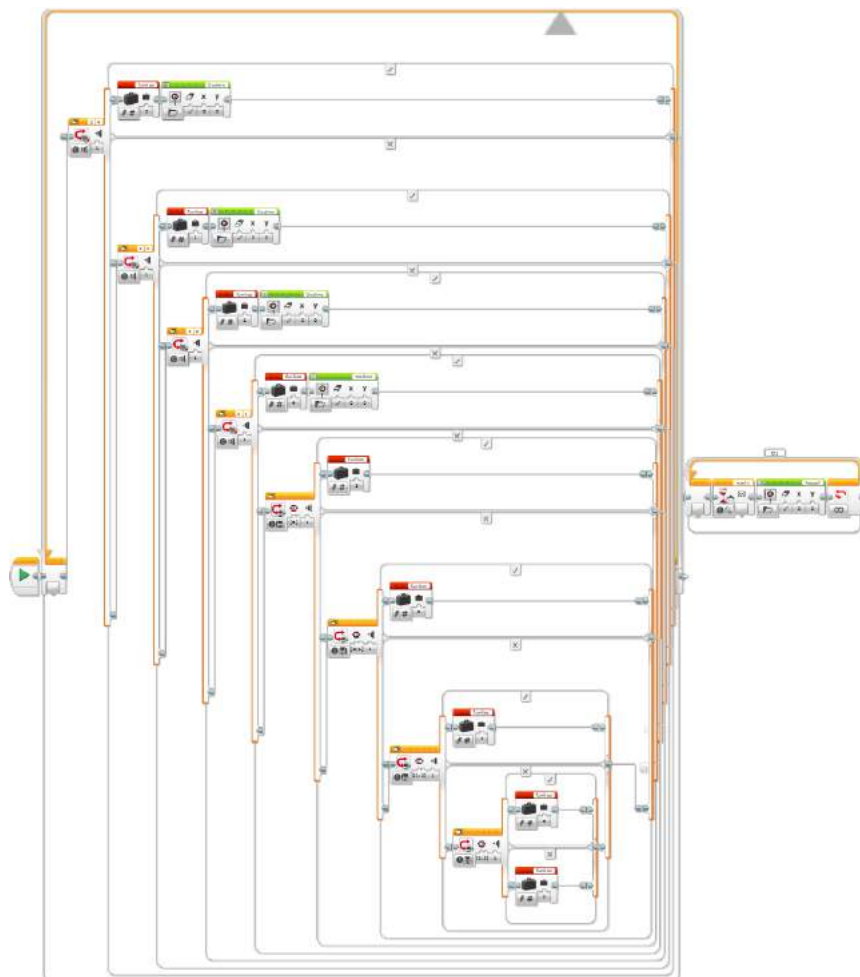
Komunikační segment kódu [Obrázek 4.4] zvolený na robotovi Ludvíkovi nejdříve zapne rozhraní bluetooth a poté jako indikaci spuštění začne EV3 žlutě blikat. Poté pošle program handshake do robota zobrazí na displeji značku otazníku a vydá zvuk. Poté čeká na odpověď handshake. Pokud dostane zprávu zpět, přestane blikat, vydá znovu zvuk a začne svítit červeně. Poté zapne komunikační loop programu. Tento loop má tři oddělené zprávy, které posílá. První dvě jsou binární hodnoty pohybu pásu pro obě strany. Kód odesílá 0,1,2 což je interpretováno jako stop, vzad, vpřed. Tato funkce je stejná pro obě strany. Třetí funkce odešle hodnotu zapnuté funkce programu. Funkcí lze odeslat do velikosti INT, ale na ovladači bylo využito pouze osmi dotykových senzorů a poté pět tlačítek na robotovi, z tohoto důvodu robot využívá maximálně třinácti funkcí. Zpráva je odeslána opět v integeru 0,1,...,13 kde 0 značí, že nebyla spuštěna žádná funkce. Tyto tři zprávy jsou odděleny z důvodu potřeby pohybu robota při vykonávání jednotlivých funkcí.



**Obrázek 4.5:** Ovládání pásů

Část programu pro jízdu [Obrázek 4.5] je triviální a funguje pro oba pásy stejně. Kód v nekonečném cyklu načítá pomocí dvou vnořených If funkcí

hodnotu na dotykových senzorech. Nejdříve ověří zda byl zvolen pohyb vpřed, pokud ano, tak odešle hodnotu 2. Pokud se tak nestane, tak zkontroluje pozici druhého senzoru a podle něj odešle hodnotu buď 1 nebo 0.



Obrázek 4.6: Rozpoznání funkcí

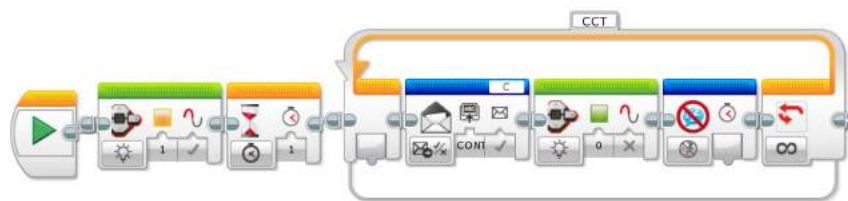
Hlavní částí kódu v ovladači je vlákno [Obrázek 4.6], které odečítá jednotlivé spuštěné funkce. Opět jako u pásového vlákna i toto odečítá postupně hodnoty jednotlivých senzorů a pokud je některý z nich kladný, tak ho odešle do robota. Nevýhodou tohoto kódu byla nemožnost zmáčknutí dvou funkcí naráz. Nicméně tato chyba neměla na robota žádný vliv, jelikož byl robot ovládán tak, aby využíval pouze jednu funkci naráz. V kódu je možnost povšimnutí zobrazení kříže na displeji jako indikace, že daná funkce právě probíhá.

Shrnutím lze napsat, že jako takový byl první kód pro ovládání naprosto

triviální a fungoval, dostatečně pro fungování robota.

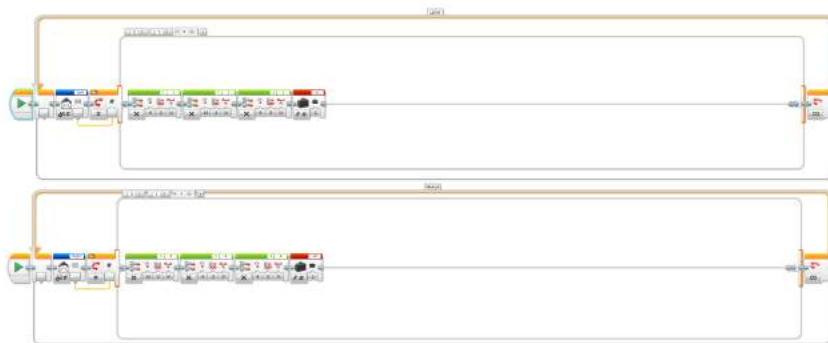
## 4.4 Hlavní program

Program v robotovi byl na rozdíl od ovládacího programu často přepisován a to právě kvůli právě použitým funkcím. V následující kapitole bude program opět popsán, ale z funkcí bylo vybráno jen několik exemplárních.



Obrázek 4.7: přijmutí signálu

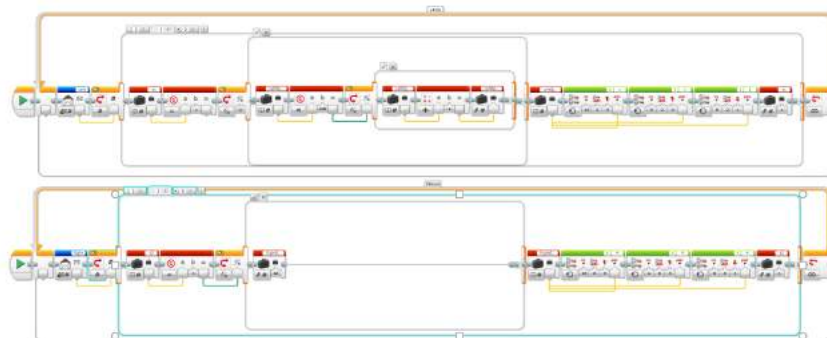
První část kódu [Obrázek 4.7] slouží pouze pro handshake a poté je využit blok, který robota nenechá vypnout. Tento blok je velice důležitý, jelikož EV3 nebere přijmutí zprávy bluetooth jako “použití robota” a díky tomu se po půl hodině program sám vypne.



Obrázek 4.8: Ovládání pásů (1)

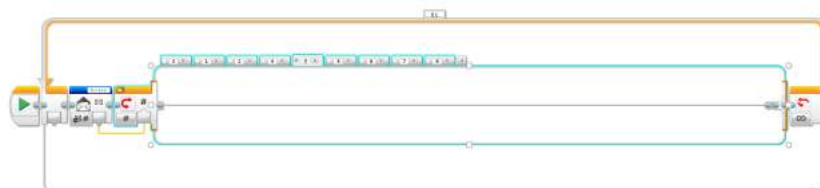
Další částí je použití pásů. Tento kód měl několik změn. Z tohoto důvodu bude ukázáno jen jedno řešení a popsány jeho změny. Na obrázku [Obrázek 4.8] lze vidět funkci STOP pro robota. Tato funkce funguje opět jako nekonečný cyklus, kde přijímá hodnoty pohybu pásů a podle nich se rozhoduje,

kterou část programu pustit. Bylo využito funkce, která je ekvivalentní struktuře if/elseif/else. Pokud přijde hodnota 0, tak robot zastaví všechny motory, ale využije u nich funkci “coast”, která zaručí, že motory doběhnou na volno běh a nezarazí robota na místě. Tato věc byla velice důležitá při testování, jelikož bez této změny se robot nepřiměřeně moc kýval.



Obrázek 4.9: Ovládání pásů (2)

Jelikož jsou pásy postaveny symetricky [Obrázek 4.9], tak hodnota 1 je pro jeden pás stejná jako pro druhý hodnota 2. V poslední části funkce je inverzně napsaný stejný program. Na obrázku lze vidět, že pro levý pás program načte hodnotu a pokud odpovídá hodnotě 1, tak si překontroluje zda je proměnná *s1* rovna naposled stisknutému programu. Pokud ano provede navýšení rychlosti robota. Toto je část, která byla přidána později a funguje jako regulace pro robota. V prvních testech se robot rozkýval při rychlém rozjezdu. Tato část zaručila jeho pomalejší zrychlení, kde v prvním běhu zapne motory na 60% a postupně přidává pokud stále dostává stejný povел k jízdě. Následně je hodnota odeslána do motorů a uložena nová hodnota do *s1*. Pro poslední možnost programu se právě program dostal poprvé do funkce jízdy daným směrem, a proto je hodnota nastavena pouze na 60%.



Obrázek 4.10: Prázdná funkce

Vlákno pro jednotlivé funkce bylo nastaveno stejným způsobem. Program si načte hodnotu funkce a tu spustí, dokud není celá provedena, tak nenačítá další funkci. Na obrázku [Obrázek 4.10] lze vidět hodnotu 0, která nezapíná



funkci žádnou.



**Obrázek 4.11:** Ukázka funkce (1)

Jako dalším příkladem [Obrázek 4.11] byla zvolena funkce u které robot zastříhá obočím. U všech funkcí byla nastavena zpětná vazba, kde odešle robot do ovladače FALSE u proměnné READY a zamezí použití další funkce. V tomto případě následně zapne motor na hlavě a obočí, které zvedne a položí v rychlém intervalu. Následně odešle TRUE a robot začne čekat na další funkci.



**Obrázek 4.12:** Ukázka funkce (2)

Tento příklad [Obrázek 4.12] byl zvolen pouze jako ukázka použití funkce bloku. Pod tyrkysovým blokem se ukrývá delší kód. Program vykoná vše postupně jako v předešlém kroku, jen pro uživatele je tento postup přehlednější.



Obrázek 4.13: Ukázka plné funkce

Pod tyrkysovým blokem se ukrývá funkce HandIn, kterou lze vidět na obrázku [Obrázek 4.13]. Díky této funkci si robot ze zásobníku v trupu vezme do ruky připravený leták a podá ho před sebe. Nebude zde zacházeno do podrobností. Ve zkratce si robot v postupné sekvenci nastaví ruku tak, aby směřovala před zásobník z kterého poté vyjede papír a ten robot posléze chytne a ruku předpaží. Na konci kódu robot využívá funkce RESET který je jednou z fundamentálních částí robota, neboť se potřebuje v nějaké pozici dostat do svého “středu pracovního prostoru”, aby mohl znovu používat další funkce.



Obrázek 4.14: Ukázka funkce reset

Celý program RESET lze vidět na obrázku [Obrázek 4.14]. Program, využitím co nejmenší síly (tato hodnota byla optimalizována na 10-20% pro různé klouby), posune všemi končetinami do nulové pozice, kde se resetuje. Jedinou výjimkou je předloktí, kde je kloub postaven na otočném kole, které nemá bod v kterém by se mohlo zaseknout, z tohoto důvodu byla použita druhá řádka obrázku stavební kolík. Ten byl zapojen tak, aby do něj robot narazil při otáčení. Pokud by po nárazu ovšem pokračoval, došlo by k poškození robota a proto byla přidána funkce pro odečet proudu v motoru, díky kterému robot pozná, že narazil na překážku a zastaví se.

## 4.5 Zvuky

Mezi tyto funkce jsou v některých funkcích přidávány zvukové bloky, ve kterých podle požadavků robot mluví. Jelikož zatím Robot Ludvík neumí autonomně komunikovat, je nutné předem nastavit co kdy bude říkat. Toho je docíleno pomocí překladače do zvukového formátu[nz]. Chtěný text je přeložen do robotické mluvy a ta je posléze uložena jako mp3. Následně je nutno zvuk vložit do Lego Mindstorms EV3 Home Edition pomocí Sound Editoru. Následně lze ve zvukovém bloku požadovaný segment najít a použít. Nevýhodou tohoto postupu je nízká kvalita zvuku, která je docílena díky špatnému překladači v Lego Mindstorms EV3 Home Edition a další nevýhodou je možnost použití maximálně osmi vteřin zvuku. Pro delší proslovy je nutné využití více bloků.

## 4.6 Nový ovladač

Nový ovladač. Během podzimu roku 2019 bylo rozhodnuto, že ovladač není pro robota dostačující a byl objednána PSP-nX Combo[PSP] opět z MindSensors. Tento ovladač sliboval mnoho nových funkcí a stejně jako multiplexory měl už přednastavené funkce pro Lego Mindstorms EV3 Home Edition, C, Java...



**Obrázek 4.15:** PSP\_nX\_V4 [p9]

Tento ovladač je předělánkou celosvětově známého ovládacího panelu platformy PlayStation. Se sedmnácti tlačítky a dvěma joisticky lze ovládat téměř cokoliv. Toto řešení je využito i u mnoho dalších robotů na katedře Robotiky. Speciální jednotkou je PSP-Nx-v4 [Obrázek 4.15], který převádí signály do EV3 kostky. Jeho zapojení pro Lego Mindstorms EV3 Home Edition je triviální. Lze naimportovat přednastavený blok a využít jeho funkcí.



**Obrázek 4.16:** PSP blok

Ovládací blok [Obrázek 4.16] má více funkcí, avšak většina z nich není dostatečně modifikovatelná. Bylo zjištěno, že funkce BIT SEPARATOR velice jednoduše rozliší veškeré pohyby na ovladači. Tato funkce byla využita v

programu. První nastavení je pouze I2C adresa ovladače. Tato adresa se nemusela nijak změnit, jelikož je z logických důvodů využíván právě jeden ovladač. První čtyři výstupy jsou hodnoty ve vertikálních a horizontálních osách joysticků ovladače. Tyto výstupy vrací hodnoty -100,100 což je pro manipulaci s rychlostí ideální. Levý joystick byl zvolen jako ovládací pro jízdu robota.

Zbylé výstupy jsou binární hodnoty 0,1, které značí zda je tlačítko zmáčknuté, či nikoliv. Řídící program byl upraven tak, aby jednotlivá tlačítka zaznamenávala různé funkce.

Jedním z hlavních výpočtů robota bylo převedení hodnot z joysticků do jízdy. Zde byla využita základní matematika, kde si robot určí maximální výkon vpřed pomocí vertikální hodnoty a poté v procentuální míře sníží rychlost jednoho z pásů na hodnotu určenou horizontální hodnotou. Díky této části se robot pohyboval plynule i při zatáčení. Největším problémem bylo převedení hodnot do záporného stavu tak aby při nulové jízdě vpřed a maximální jízdě do stran byl otočen na místě. Toto bylo vyřešeno pouze opačným výkonem na obou pásech. Posledním ošetřením hodnot bylo omezení na -100% až 100%, jelikož bylo zjištěno, že se některé přijaté hodnoty od této reality odlišují.



# Kapitola 5

## Python

Cílem této kapitoly je seznámení s jiným programovacím jazykem pro EV3 a to je Python. Bude zde obeznámeno jak s EV3DEV, což je neoficiální boot Linuxu na EV3 běžící v Pythonu a také bude probrán MicroPython, který je od roku 2019 podporován oficiálně. Dále budou ukázány příklady toho, co lze v Pythonu tvořit a co bylo v rámci této bakalářské práce zjištěno. V poslední řadě budou shrnuty výhody a nevýhody jednotlivých jazyků.

### 5.1 EV3DEV

Ev3dev[EV3] je Debian-Linux flash podporující programovací jazyk Python. Instalace EV3DEV je detailně popsána na jejich oficiálních stránkách, proto zde bude popsáno pouze shrnutí. Je nutné stáhnout debian stretch z jejich stránek a pomocí libovolného programu jako je například Etcher flashnout stretch na microSD. Po vložení do kostky se na ní spustí stretch debianu a uživatel může kostku využívat jako Linuxový počítač. Pro připojení do kostky je nutné připojit SSH[SSH], přes které jde kostka ovládat.

Kód jde napsat v libovolném prostředí pro Python, je pouze nutné naimportovat knihovnu ev3dev2. Ta obsahuje všechny potřebné funkce pro ovládání robota.

```
#!/usr/bin/env python3
from ev3dev2.motor import LargeMotor, OUTPUT_A, OUTPUT_B, SpeedPercent, MoveTank
from ev3dev2.sensor import INPUT_1
from ev3dev2.sensor.lego import TouchSensor
```

```
from ev3dev2.led import Leds
```

Na obrázku lze vidět základní přednastavení při psaní kódu, kde je naimportováno několik motorů a senzorů. V ev3dev lze napsat jakýkoliv kód, který je omezen pouze znalostí Pythonu, využívání funkcí je intuitivní a nic ve struktuře programování nemění.

### ■ 5.1.1 Propojení více kostek

Jedním z hlavních problémů které byly řešeny, bylo propojení více jak čtyř kostek EV3 dohromady. Toto řešení sice nebylo využito na robotovi Ludvíkovi, nicméně stojí za zmínění z důvodu možných využití v budoucnu.

Řešeným způsobem bylo propojení šesti EV3 pomocí bluetooth. K tomuto řešení bylo využito knihovny PyBluEz[PyB], která je dlouhodobě rozsáhlá v bluetooth komunikaci v rámci programovacího jazyku Python. Jelikož se s EV3DEV kostka chová jako debian, tak lze pomocí terminálu instalovat programy. Toto má v rámci řešení závažný problém. EV3DEV stretch je “osekán”, tak aby byl pro EV3 co nejjednodušší a proto nelze nainstalovat PyBluEz v prvním kroku.

Řešení tohoto problému zabralo mnoho času. V létě roku 2019 bylo o tomto řešení diskutováno a tento problém se ukázal jako velice závažný. Proto z něj bylo upuštěno. Nicméně na jaře 2020 bylo toto řešení dopracováno s pomocí Bc. Michala Urválka a jeho týmu, který v roce 2018 vytvořil robota skládajícího Rubikovu kostku pouze z LEGO EV3. Instalační řešení této knihovny není zcela triviální a zabere mnoho času. V rámci pokusů a instalací byl vytvořen dokument, kde se vylepšovala řešení jak dosáhnout úspěšné instalace. Nutno podotknout, že řešení bylo prováděno na operačním systému Windows 10 a na jiných systémech se může lišit. Řešení tohoto problému bylo následující:

1. Flash kostky na ev3dev pomocí programu Etcher
2. Při prvním zapnutí kostky je nutno ihned zapnout bluetooth na EV3. Pokud se tak neučiní, tak v dalším restartu kostky již nebude bluetooth dostupný a celá instalace by byla nutná udělat znovu. Nebylo nalezeno přesné vysvětlení, nicméně bylo zjištěno, že z důvodu duplikování adres v kernelu některé funkce z EV3DEV mizí pokud jim není přiřazena priorita. **TENTO PRVNÍ KROK JE FUNDAMENTÁLNÍ**



3. Pokud je v plánu využít propojení více kostek, tak je v této části nutno kostku přejmenovat. Viz kapitola Změna hostname
4. Připojení EV3 k internetu přes USB kabel. Kompletní návod na instalaci je na EV3DEV[USB]. Bylo zjištěno, že při restartování a opětovném použití se internet nepřipojí, pokud se tak stane, je nutno v nastavení připojení vypnout a zase zapnout tethering. (“povolit další” odškrtnout/zaškrtnout)
5. Dalším krokem je v EV3 povolit připojení k internetu. Toho je docíleno zapnutím funkce wired v nastavení kostky a kliknutím na tlačítko Connect. (Kontrolou správného postupu je napsaná IP adresa kostky v horní části obrazovky kostky.)
6. Poté je nutno se připojit do terminálu na kostce. Toho lze docílit pomocí SSH. V tomto návodu bylo využito programu Putty[Put].
  - a. Pro připojení do kostky je nutné vyplnit jméno a heslo. Defaultní přednastavení:
    - (i) Name : ev3dev (pro připojení Putty do kostky)
    - (ii) User : robot (Jméno pro připojení v terminálu kostky)
    - (iii) Pass : maker (Heslo pro připojení v terminálu kostky)
7. Pokud veškerá instalace proběhla v pořádku, tak by měl uživatel mít otevřen Linux terminál a na kostce by měla být vidět IP adresa. Pokud je tomu tak, tak lze začít s instalací. Následující instalace je nutno provádět postupně v chronologickém pořadí:
  - a. `#!/bin/sh`
  - b. `sudo apt-get update` (doba trvání instalace 3min, heslo pro sudo je opět maker)
  - c. `sudo apt-get install python3-pip python3-dev ipython3 -yes` (cca 25min)
  - d. `sudo apt-get install bluetooth libbluetooth-dev -yes` (cca 6min)
  - e. `sudo pip3 install pybluez`
  - f. `sudo chmod 777 /var/run/sdp` (pokud tento příkaz vyhodí error nebo nevyhodí nic, tak není stále potřeba nic řešit.)
8. Propojení složky s počítačem pro jednodušší pohyb se soubory lze docílit připojením SSHFS. Pro Win10 lze připojení docílit, pokud je uživatel klikne v průzkumníku na Síť a zaklikne “Přidat síťovou jednotku”, poté je nutno zadat “`\\sshfs \ev3dev \`”. (pokud byla již kostka přejmenována, tak je nutno změnit název za právě používaný.)
9. Dalším krokem je připojení knihovny bluetooth:
  - a. V kostce je nutno se donavigovat do adresáře: `/lib/systemd/system/bluetoothd`

- b. Soubor `bluetooth.service` je nutno otevřít v textovém editoru. (např. nano). Příkaz “`nano bluetooth.service`”
  - c. V otevřeném souboru je nutno změnit řádku:
    - (i) `ExecStart=/usr/lib/bluetooth/bluetoothd`
    - (ii) Na: `ExecStart=/usr/lib/bluetooth/bluetoothd -C -compat`
10. V terminálu zadat příkaz: `sudo chmod 777 /var/run/sdp`

### ■ 5.1.2 Změna hostname

Pokud chce uživatel využít více kostek, je nutno jednotlivé EV3 přejmenovat toho lze docílit pomocí následujícího návodu[Hos]:

1. Zadat příkaz: `sudo nano /etc/hostname`
2. Po otevření nano, lze smazat starý název a napsat nový
3. Zadat příkaz: `sudo reboot`
4. Zapamatovat si nový název, bude použit pro připojení v Putty (osvědčilo se napsat na kousek lepenky název kostky a přilepit ho na ní.)

### ■ 5.1.3 Errorry

Po instalaci lze narazit na dva errorry při spouštění programu:

### ■ 5.1.4 python3/r

Tento error[END] je způsoben nekompatibilitou textových editorů. Ve Windows se na konce řádek přidává několik znaků, které nejsou vidět a v Linuxu nikoliv. To lze vyřešit instalací programu `dos2unix`

```
sudo apt install dos2unix
```

Po úspěšné instalaci lze převádět soubory z jednoho systému do druhého pomocí

```
dos2unix /PATH/TO/YOUR/WINDOWS_FILE
unix2dos /PATH/TO/YOUR/LINUX_FILE
```

## 5.1.5 Bluetooth error 13 (Permission Denied)

Tento error zamezuje přístupu k bluetooth knihovně z důvodu omezení práv na kostce. Tento error má dvě řešení, první je jednoduché, ale po každém restartu kostky je ho nutno opakovat, druhé je permanentní.

1. Před každým spuštěním programu je nutno být přihlášen jako administrátor.
  - a.
  - b. Příkaz: `sudo -s`
    - (i) Name: robot
    - (ii) Pass: maker
2. Toto řešení je komplikované, nicméně permanentní. Je nutno vytvoření dvou souborů.
  - a. Nejdříve je nutné překopírovat soubor `bluetooth.device` o složku výše, toho ze docílit pomocí příkazu:
    - (i) `cp -i /lib/systemd/system/bluetooth.service /etc/systemd/system/`
  - b. Je nutno vytvoření dvou souborů. Pro vytvoření souboru lze zadat:
    - (i) Příkaz pro tvorbu souboru: `cat > filename` (Pro uložení CTRL+D a pro otevření “`cat filename`”)
    - (ii) První soubor je “`var-run-sdp.path`”. Je nutno ho vytvořit ve složce: `/etc/systemd/system/var-run-sdp.path`
    - (iii) Do souboru je poté nutno uložit:
 

```
[Unit] Description=Monitor /var/run/sdp
[Install] WantedBy=bluetooth.service
[Path] PathExists=/var/run/sdp Unit=var-run-sdp.service
```
    - (iv) Druhý soubor je “`var-run-sdp.service`”. Je nutno ho vytvořit v `/etc/systemd/system/var-run-sdp.service`
    - (v) Do souboru vložit tento text:
 

```
[Unit] Description=Set permission of /var/run/sdp
[Install] RequiredBy=var-run-sdp.path
[Service] Type=simple ExecStart=/bin/chgrp bluetooth /var/run/sdpc
```

- c. Posledními kroky je pouze aktivace vytvořených souborů, které udrží práva tak, že i po restartu bude mít kód přístup kam potřebuje a to bez nutnosti připojení administrátora.
- (i) `sudo systemctl daemon-reload`
  - (ii) `sudo systemctl enable var-run-sdp.path`
  - (iii) `sudo systemctl enable var-run-sdp.service`
  - (iv) `sudo systemctl start var-run-sdp.path`
  - (v) `sudo usermod +aG bluetooth <username>`
  - (vi) `newgrp bluetooth`

Pokud byla instalace úspěšná, tak je možné pustit ukázkový kód[Exa]

Pokud vše proběhlo v pořádku, tak už je buď na uživateli, jak komunikaci využije a nebo je možno využití mého kódu:

```
#-----SOCKET 1-----
server_sock = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
server_sock.bind(("", bluetooth.PORT_ANY))
server_sock.listen(1)
port = server_sock.getsockname()[1]

uuid = "94f39d29-7d6d-437d-973b-fba39e49d4ee"

bluetooth.advertise_service(server_sock, "Piano1", service_id=uuid,
                             service_classes=[uuid, bluetooth.SERIAL_PORT_CLASS],
                             profiles=[bluetooth.SERIAL_PORT_PROFILE],
                             )

#Sound.speak("hi", 0)
print("Start Piano", port)
#Sound.speak('Start Piano 1', 0)
client_sock1, client_info1 = server_sock.accept()
print(bricks.get(client_info1[0]), "connected!")
#Sound.speak("Piano 1 connected!")
bluetooth.stop_advertising(server_sock)
```

Tato část kódu je stejná pro všechny sockety, jen jsou změněny názvy připojených socketů. V prvních řádcích je nastaven server socket, který komunikuje, a port. Poté je připojeno uuid bluetooth komunikace. Následně je celý socket advertizován a čeká na připojení. Pokud nějaký client zareaguje a bude se chtít připojit, tak ho server přijme a spáruje se. Tento postup lze použít pro kolik klientů je potřeba, je pouze nutno vždy vypnout advertising.

## 5.2 MicroPython

Druhý řešením je oficiální MicroPython[Mic]. LEGO v roce 2018 pochopilo, že uživatelé nejsou pouze děti, ale i studenti nebo dospělí, kteří chtějí s roboty dělat složitější triky než rozpoznání barev kuličky. Z tohoto důvodu se LEGO

spojilo s EV3DEV a na jejich kernelu postavilo vlastní verzi pythonu pro EV3. Není známo z jakého důvodu se LEGO rozhodlo podpořit právě Micropython a ne už vytvořený Python a navíc proč existuje nová knihovna, která využívá nově stejné funkce, nýbrž jinak pojmenované. Při testování obou jazyků bylo zjištěno, že pokud se uživatel drží u programování pouze LEGO, tak je nová knihovna intuitivnější a jednodušší, ale jakmile se od toho odprostí, tak je programování v Micropythonu peklo.

### ■ 5.2.1 Výhody a nevýhody MicroPythonu

Výhod má Micropython hned několik. První z nich je triviální použití. Stačí mít flashnutou kostku jejich softwarem a poté ve Visual Studio Code[Cod] naprogramovat kód. LEGO má ve VSC připravený plugin, který automaticky připojí kostku k počítači a jedním tlačítkem lze nahrát kód do EV3. Další výhodou je velice jednoduchá knihovna s přednastavenými funkcemi tak, že uživatel stále potřebuje pouze základní znalosti programování.

Nevýhodou je, že složitější projekty nejsou podporovány jelikož několik externích knihoven jako např. PyBluEz chybí a jelikož je tento software oficiální, tak ho v mnoha ohledech nelze přepisovat ani pozměnit.

Pro využití jedné kostky a EV3 robotů využívající čistě LEGO sety je doporučeno využití LEGO MicroPythonu. Pro složitější operace jako je spojování kostek nebo ovládání předmětů mimo LEGO svět je doporučeno využití EV3DEV.

V rámci této bakalářské práce bylo rozhodnuto, že další krok v budoucnu bude přepsání všech kódů na robotovi Ludvíkovi do MicroPythonu, hlavním důvodem bylo rozšíření znalostí o novém Micropythonu, jelikož krom jednoho oficiálního PDF neexistuje mnoho informací jak tento software funguje.

### ■ 5.2.2 Instalace MicroPythonu

Na oficiálních stránkách LEGO lze najít podrobný návod[Ná] pro instalaci všech potřebných aplikací. Ve zkratce řečeno, stačí flashnout jejich softwarem MicroSD pomocí Etcheru a vložit ji do kostky. Poté si uživatel nainstaluje Visual Studio Code a v něm nainstaluje PlugIn LEGO EV3. Dále stačí pouze

vytvořit nový program ve VSC a s využitím přednastavených importů začít psát program.

```
#!/usr/bin/env pybricks-micropython
from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                  InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import (Port, Stop, Direction, Button, Color,
                                  SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase
import rpyc
# Write your program here
brick.sound.beep()
```

Výše je vidět základní kód, který naimportuje všechny knihovny a řekne robotovi, aby jedenkrát pípl pomocí funkce `brick.sound.beep()`. Dále lze v levém spodním rohu vidět `EV3DEV DEVICE BROWSER`. Pokud je k počítači připojeno EV3, tak se objeví právě v této části obrazovky. Všechny potřebné návody pro začátek využití MicroPythonu lze nalézt na webových stránkách LEGO[Ná].

V březnu 2020 začal vývoj nového kódu pro Robota Ludvíka, z důvodu COVID-19 nebyl kód zatím dovytvořen a některé věci nebyly dozjištěny. V následující části bude napsáno co bylo vyzkoumáno.

### 5.2.3 OVLADAČ PSP-nX-V4

První částí celého programu bylo zjištění ovládání. Oproti Lego Mindstorms EV3 Home Edition, v Pythonu už nejde “jen spojit čárou dva bloky” a robot komunikuje. Na stránkách MindSensors existuje návod na využití PSP-nX pro programovací jazyk C a Java. V rámci studií na ČVUT jsem absolvoval jak kurzy Pythonu, tak i C. Z tohoto důvodu byla Java rovnou zavrhnuta a jelikož jak Python, tak Micropython je postaven na programovacím jazyce C, tak nebylo nemožné celý kód přepsat a upravit pro Micropython.

```
def PSP_SendCommand(port, i2cAddr, command): ##tSensors port
    msg = []
    ## Build the I2C message
    msg[0] = 3
    msg[1] = i2cAddr
    msg[2] = 0x41
    msg[3] = command
    ## Wait for I2C bus to be ready
    while (nI2CStatus[port] == STAT_COMM_PENDING)
    ## when the I2C bus is ready, send the message you built
    sendI2CMsg(port, msg[0], 0)
    while (nI2CStatus[port] == STAT_COMM_PENDING)
```

Funkce `PSP_SendCommand()` využívá adresace I2C, která funguje stejně

jako pro Lego Mindstorms EV3 Home Edition. Do ovladače lze tímto kódem odeslat jakoukoliv žádost jak o stavech tlačítek, tak i o jeho hladině baterie atd.

```
def PSP_ReadButtonState(port, i2cAddr, currState): ##tSensors port
    msg = []
    replyMsg = []

    msg[0] = 2
    msg[1] = i2cAddr
    msg[2] = 0x42

    ## set all buttons to "0"
    currState["b1"] = 0
    currState["b2"] = 0
    currState["l1"] = 0
    currState["l2"] = 0
    currState["r1"] = 0
    currState["r2"] = 0
    currState["a"] = 0
    currState["b"] = 0
    currState["c"] = 0
    currState["d"] = 0
    currState["e"] = 0
    currState["f"] = 0
    currState["triang"] = 0
    currState["square"] = 0
    currState["circle"] = 0
    currState["cross"] = 0
    currState["l_j_b"] = 0
    currState["r_j_b"] = 0
    currState["l_j_x"] = 0
    currState["l_j_y"] = 0
    currState["r_j_x"] = 0
    currState["r_j_y"] = 0

    ## Wait for I2C bus to be ready
    while (nI2CStatus[port] == STAT_COMM_PENDING)

    ## when the I2C bus is ready, send the message you built
    sendI2CMsg(port, msg[0], 6) ##TODO

    ## Wait for I2C bus to be ready
    while (nI2CStatus[port] == STAT_COMM_PENDING) ##TODO

    ## read back the response from I2C
    readI2CReply(port, replyMsg[0], 6) i2c.read_from_mem(42,2,...)

    b0 = replyMsg[0]&0xff
    b1 = replyMsg[1]&0xff

    currState["b1"] = b0
    currState["b2"] = b1

    currState["e"] = (b0 >> 0) & 0x01
    currState["l_j_b"] = (b0 >> 1) & 0x01
    currState["r_j_b"] = (b0 >> 2) & 0x01
    currState["f"] = (b0 >> 3) & 0x01
    currState["d"] = (b0 >> 4) & 0x01
    currState["c"] = (b0 >> 5) & 0x01w
    currState["b"] = (b0 >> 6) & 0x01
    currState["a"] = (b0 >> 7) & 0x01

    currState["l2"] = (b1 >> 0) & 0x01
    currState["r2"] = (b1 >> 1) & 0x01
    currState["l1"] = (b1 >> 2) & 0x01
    currState["r1"] = (b1 >> 3) & 0x01
    currState["triang"] = (b1 >> 4) & 0x01
    currState["circle"] = (b1 >> 5) & 0x01
    currState["cross"] = (b1 >> 6) & 0x01
    currState["square"] = (b1 >> 7) & 0x01

    currState["l_j_x"] = (((replyMsg[2]&0xff) - 128) * 100)/128
    currState["l_j_y"] = (((replyMsg[3]&0xff) - 128) * 100)/128
    currState["r_j_x"] = (((replyMsg[4]&0xff) - 128) * 100)/128
    currState["r_j_y"] = (((replyMsg[5]&0xff) - 128) * 100)/128
```

Funkce `PSP_ReadButtonState()` je hlavní částí tohoto programu. Při každém zavolání si funkce uloží do všech hodnot každého tlačítka hodnotu 0, tedy, že tlačítko není zmáčklé nebo, že joystick není posunut. Poté program čeká na uvolnění I2C a následně odešle zprávu s dotazem na stavy tlačítek. Příchozí

zprávu si uloží do replyMsg. Posledním krokem je dekodování zprávy. V prvních 16bitech jsou uloženy hodnoty pro jednotlivá tlačítka. Program si postupně posouváním bitových operátorů ukládá do slovníku jednotlivé stavy tlačítek a následně přečte zbytek zprávy. Poslední čtyři části jsou po převedení do desítkové soustavy hodnoty -100, 100 a to pro oba joistiky v horizontální i vertikální pozici. Po tomto kroku zatím program končí. V currState jsou uloženy aktuální hodnoty ovladače. Výhodou tohoto řešení je nejen funkčnost v MicroPythonu, ale také možnost využití “dvojího” stisku tlačítek, díky kterému jde přidat více funkcí. Jako příklad lze uvést toto. Pokud bude zmáčknuto tlačítko L1, tak robot zvedne levou paži, pokud R1, tak pravou. Nýbrž pokud budou zmáčknuty obě naráz, tak zahýbe hlavou. Tato možnost nebyla u prvního řešení možná.

Jediným zatím nalezeným problémem bylo v hardwarovém řešení levého joistiku nalezeno to, že hodnota horní vertikální pozice odpovídá pouze 87% nikoliv předpokládaným 100%.

## 5.2.4 Multiplexory

Další neprozkoumanou částí Micropythonu je využití I2C s multiplexory. V předchozí části bylo zjištěno, že upravení kódu z C do Micropythonu je funkčným řešením. Z tohoto důvodu bylo stejné řešení využito i pro multiplexory. Vyřešenými částmi je připojení k jednotlivým multiplexorům pomocí I2C, toto řešení je stejné jako u PSP-nX. Další hotovou částí je vytvořený scanner, který v EV3 vyzkouší všechny možnosti připojených komunikací na I2C a vrátí přihlášené adresy.

```
def scan_i2c(INPUT):
    add = [0x00, 0x01, 0x02, 0x03, ..., 0xfe, 0xff]
    for i in range(len(INPUT)):
        for j in range(1, 127):
            try:
                bus_test = i2c.SMBus(INPUT[i])
                bus_test.set_address(add[j])
                firmware = bytes(bus_test.read_i2c_block_data(0x00, 8)).decode().strip()
                vendor = bytes(bus_test.read_i2c_block_data(0x08, 8)).decode().strip()
                product = bytes(bus_test.read_i2c_block_data(0x10, 8)).decode().strip()
                print(i+1, " ", hex(add[j]), " ", product, "\n")
                brick.display.text(product, (20, 20+j))

            except:
                print("****")
#-----END-----functions-----

#-----i2c-----setup-----
INPUT = [ '/dev/i2c-in1', '/dev/i2c-in2', '/dev/i2c-in3', '/dev/i2c-in4' ]

bus_psp = i2c.SMBus(INPUT[3])
bus_psp.set_address(0x01)
#-----END-----i2c-----setup-----
```

Kód zatím není dostatečně rychlý a přesný tak jak by bylo vhodné. V poli INPUT si načte všechny čtyři adresy portu na EV3 a v poli add si



uloží všechny možnosti adres pro I2C. Následně je všechny vyzkouší a pokud dostane odpověď, tak vrátí jejich adresu.





## Kapitola 6

### Web

V rámci bakalářské práce byly vytvořeny webové stránky, které budou sloužit jako zdroj informací o robotovi Ludvíkovi. Celé stránky jsou pod webem Robo-soutěže. Veškeré práce na stránkách byly vytvořeny v programu Drupal[Dru].



## Kapitola 7

### Závěr

Robot Ludvík se v posledním půl roce účastnil třech koncertů PFK dětem. V prvním z nich pouze jezdil a zdravil kolemjdoucí. Při následujícím koncertu byl schopen lehké komunikace s účastníky a podal i rozhovor do televize. V posledním koncertu se dokonce účastnil celého vystoupení na hlavním podiu, kde si zahrál měsíčního robota a hned dvakrát sklidil potlesk všech 1500 diváků. Robot Ludvík byl navíc několikrát pozván do televize jako například Snídaně s Novou, dětský pořad Wifina a jiné. V tomto roce je plánována jeho účast v dalších vystoupeních PFK dětem. Navíc je robot využíván pro zahajování Robosoutěže a používán při propagačních událostech ČVUT FEL jako jsou dny otevřených dveří nebo GAUDEAMUS.

Tyto všechny věci naznačují fakt, že stavba robota byla úspěšná a splnil původní očekávání, či dokonce je předčil. Nicméně je mnoho neprobádaných částí, na kterých je potřeba zapracovat. V softwarové části je hned několik mezer. Hlavní z nich je přepsání celého robota do MicroPythonu/Pythonu. Tato věc by umožnila mnohé další možnosti, jako je připojení mikrofónu nebo kamery. Díky tomu by mohla být otestována funkčnost v autonomním fungování robota v oblastech rozpoznávání řeči a obrazu.

V hardwarové části jsou nedostatky především v podvozku, který neumí jezdit po všech površích. Dalším problémem je nedostatečná propracovanost rukou, které se mohou povolit při špatně zvoleném pohybu. Dalším upgradem je v plánu vytvoření povrchu po robotovi, který by připomínal modrou kůži.

Celkově si myslím, že cíle nejen bakalářské práce byly naplněny, ale stále je mnoho věcí na kterých bude potřeba pracovat. S trochou euforie je na

robotovi Ludvíkovi práce tolik, že bude místo i pro Diplomové práce.



## Literatura

- [Ary] Reprodaktor Arys, <https://www.trust.com/en/product/22945-arys-compact-2-0-speaker-set-black>, Accessed online: 3.4.2020.
- [Aut] Autobaterie, [https://www.autokelly.cz/Product/B0-0092S30010/8985419/39849642;38979167;38908625;40112824?fbclid=IwAR3wi1VfaDpaamaKPK4D\\_8Ah4Xory75Sv7YpB-vBL3kYfskKdxTKS6WqCxw](https://www.autokelly.cz/Product/B0-0092S30010/8985419/39849642;38979167;38908625;40112824?fbclid=IwAR3wi1VfaDpaamaKPK4D_8Ah4Xory75Sv7YpB-vBL3kYfskKdxTKS6WqCxw), Accessed online: 3.4.2020.
- [Cod] Visual Studio Code, <https://code.visualstudio.com/>, Accessed online: 3.4.2020.
- [Cs] Core-set, <https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set/5003400>, Accessed online: 3.4.2020.
- [Dc] Daisy-chain, [https://en.wikipedia.org/wiki/Daisy\\_chain\\_\(electrical\\_engineering\)](https://en.wikipedia.org/wiki/Daisy_chain_(electrical_engineering)), Accessed online: 3.4.2020.
- [Dru] Drupal, <https://www.drupal.org/>, Accessed online: 3.4.2020.
- [día] Ozubený díl, [https://www.bricklink.com/v2/catalog/catalogitem.page?P=18939c01&name=Technic%20Turntable%20Large%20Type%203%20Base%20with%20Black%20Technic%20Turntable%20Large%20Type%203%20Top,%2060%20Tooth%20\(18939%20/%2018938\)&category=%5BTechnic%5D#T=C](https://www.bricklink.com/v2/catalog/catalogitem.page?P=18939c01&name=Technic%20Turntable%20Large%20Type%203%20Base%20with%20Black%20Technic%20Turntable%20Large%20Type%203%20Top,%2060%20Tooth%20(18939%20/%2018938)&category=%5BTechnic%5D#T=C), Accessed online: 3.4.2020.

- [díb] Pásový díl, <https://www.brickowl.com/catalog/lego-black-large-tread-link-57518-88323>, Accessed online: 3.4.2020.
- [END] ERROR-NO-DIRECTORY, <https://askubuntu.com/questions/896860/usr-bin-env-python3-r-no-such-file-or-directory>, Accessed online: 3.4.2020.
- [Esa] EV3-software, <https://www.lego.com/en-us/themes/mindstorms/downloads>, Accessed online: 3.4.2020.
- [Esb] Expansion-set, <https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-expansion-set/45560>, Accessed online: 3.4.2020.
- [EV3] EV3DEV, <https://www.ev3dev.org/>, Accessed online: 3.4.2020.
- [Exa] Example, <https://github.com/pybluez/pybluez/blob/master/examples/simple/rfcomm-client.py>, Accessed online: 3.4.2020.
- [Fri] Science Friday, *The origin of the word 'robot'*, <https://www.sciencefriday.com/segments/the-origin-of-the-word-robot/>, Accessed online: 3.4.2020.
- [GR] Gear-Rack, <https://www.bricklink.com/v2/catalog/catalogitem.page?id=147858#T=C>, Accessed online: 3.4.2020.
- [Hos] Hostname, <https://www.cyberciti.biz/faq/ubuntu-change-hostname-command/>, Accessed online: 3.4.2020.
- [I2C] I2C, <https://i2c.info/>, Accessed online: 3.4.2020.
- [kola] Dlouhý kolík, <https://www.brickowl.com/catalog/lego-blue-long-pin-with-friction-6558>, Accessed online: 3.4.2020.
- [kolb] Krátký kolík, <https://www.brickowl.com/catalog/lego-black-technic-pin-with-lengthwise-friction-ridges-and-center-slots-1>, Accessed online: 3.4.2020.
- [kó] Multiplexor kód, <http://www.mindsensors.com/ev3-and-nxt/21-multiplexer-for-nxtev3-motors>, Accessed online: 3.4.2020.
- [Let] Reproduktor Leto, <https://www.trust.com/en/product/19830-letto-compact-2-0-speaker-set>, Accessed online: 3.4.2020.
- [LiI] LiIon, [http://www.batterysupports.com/144v-148v-818650-5200mah-4s2p-lithium-ion-liion-battery-pack-p-124.html?fbclid=IwAR04Mb9aysqnkMX6n32NardpsL3BM03IJkicTSjeQzUXUBnfwTHaVodqo\\_c](http://www.batterysupports.com/144v-148v-818650-5200mah-4s2p-lithium-ion-liion-battery-pack-p-124.html?fbclid=IwAR04Mb9aysqnkMX6n32NardpsL3BM03IJkicTSjeQzUXUBnfwTHaVodqo_c), Accessed online: 3.4.2020.



- [Mic] Micropython, <https://education.lego.com/en-us/support/mindstorms-ev3/python-for-ev3>, Accessed online: 3.4.2020.
- [Min] Mindsensors, <http://www.mindsensors.com/>, Accessed online: 3.4.2020.
- [Mul] Multiplexor, <http://www.mindsensors.com/ev3-and-nxt/21-multiplexer-for-nxtev3-motors>, Accessed online: 3.4.2020.
- [nz] Text na zvuk, <https://www.speechtech.cz/speechtech-text-to-speech/speechtech-tts-online-demo/>, Accessed online: 3.4.2020.
- [Ná] Návod, <https://le-www-live-s.legocdn.com/sc/media/files/ev3-micropython/ev3micropythonv100-71d3f28c59a1e766e92a59ff8500818e.pdf>, Accessed online: 3.4.2020.
- [Ně] Bc. Martin Němec, *Využití robota lego mindstorms ev3 - návrh robota hrajícího na piano pro propagaci fel*, <https://dspace.cvut.cz/handle/10467/76147>, Accessed online: 3.4.2020.
- [p1] p1, [https://thumbs.worthpoint.com/zoom/images1/1/0717/19/lego-mindstorms-ev3-education-core\\_1\\_2c61ff603939e52d20dc36c09ec13699.jpg](https://thumbs.worthpoint.com/zoom/images1/1/0717/19/lego-mindstorms-ev3-education-core_1_2c61ff603939e52d20dc36c09ec13699.jpg), Accessed online: 3.4.2020.
- [p2] p2, [https://img.brickowl.com/files/image\\_cache/larger/lego-black-technic-pin-with-friction-ridges-and-slots-2780-32-33899-38.jpg](https://img.brickowl.com/files/image_cache/larger/lego-black-technic-pin-with-friction-ridges-and-slots-2780-32-33899-38.jpg), Accessed online: 3.4.2020.
- [p3] p3, [https://img.brickowl.com/files/image\\_cache/larger/lego-blue-long-pin-with-friction-6558-32-99899-39.jpg](https://img.brickowl.com/files/image_cache/larger/lego-blue-long-pin-with-friction-6558-32-99899-39.jpg), Accessed online: 3.4.2020.
- [p4] p4, [https://img.brickowl.com/files/image\\_cache/larger/lego-black-large-tread-link-57518-88323-27-127447-38.jpg](https://img.brickowl.com/files/image_cache/larger/lego-black-large-tread-link-57518-88323-27-127447-38.jpg), Accessed online: 3.4.2020.
- [p5] p5, [https://img.brickowl.com/files/image\\_cache/larger/lego-medium-stone-gray-beam-frame-5-x-7-64179-27-720734-64.jpg](https://img.brickowl.com/files/image_cache/larger/lego-medium-stone-gray-beam-frame-5-x-7-64179-27-720734-64.jpg), Accessed online: 3.4.2020.
- [p6] p6, <https://www.lego.com/en-gb/product/ev3-intelligent-brick-45500>, Accessed online: 3.4.2020.
- [p7] p7, <https://i.stack.imgur.com/ddB8F.png>, Accessed online: 3.4.2020.
- [p8] p8, <http://www.bartneck.de/wp-content/uploads/2015/06/connection-diagram.jpg>, Accessed online: 3.4.2020.

- [p9] p9, [http://www.mindsensors.com/46-large\\_default/psp-nx-combo-with-wireless-controller.jpg](http://www.mindsensors.com/46-large_default/psp-nx-combo-with-wireless-controller.jpg), Accessed online: 3.4.2020.
- [PID] PID, <https://bricks.stackexchange.com/questions/4382/code-for-pid-for-ev3-g>, Accessed online: 3.4.2020.
- [PSP] PSP, <http://www.mindsensors.com/ev3-and-nxt/32-psp-nx-combo-with-wireless-controller>, Accessed online: 3.4.2020.
- [Put] Putty, <https://www.putty.org/>, Accessed online: 3.4.2020.
- [PyB] PyBluEz, <https://github.com/pybluez/pybluez>, Accessed online: 3.4.2020.
- [Rob] Robosoutěž, <https://robosoutez.fel.cvut.cz/>, Accessed online: 3.4.2020.
- [Rá] Rámeček, <https://www.brickowl.com/catalog/lego-medium-stone-gray-beam-frame-5-x-7-64179>, Accessed online: 3.4.2020.
- [SSH] SSH, <https://www.ev3dev.org/docs/tutorials/>, Accessed online: 3.4.2020.
- [Tut] Tutorial, [https://www.youtube.com/watch?v=AMBWV\\_HGYj4](https://www.youtube.com/watch?v=AMBWV_HGYj4), Accessed online: 3.4.2020.
- [USB] USB, <https://www.ev3dev.org/docs/tutorials/connecting-to-the-internet-via-usb/>, Accessed online: 3.4.2020.
- [Šr] Bc. Martin Šrámek, *Využití robota lego mindstorms ev3 - návrh robota hrajícího na harmoniku pro propagaci fel*, <https://dspace.cvut.cz/handle/10467/83047>, Accessed online: 3.4.2020.

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Štětka** Jméno: **Matěj** Osobní číslo: **466334**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra řídicí techniky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Návrh robota 'Ludvíka' pro propagační účely fakulty**

Název bakalářské práce anglicky:

**Design of the robot named Ludvík for Promotion of the Faculty**

Pokyny pro vypracování:

1. Popište současný stav návrhu a realizace robota Ludvíka
2. V grafickém prostředí EV3 navrhnete řídicí program pro ovládání robota Ludvíka
3. Vytvořte webové stránky k robotovi Ludvíkovi (popis současného stavu, princip a možnosti činnosti, vysvětlení navrženého softwaru, fotogalerii).
4. Diskutujte možnosti dalšího vylepšení robota Ludvíka, popřípadě navrhnete řídicí program v programovacím jazyce Python for EV3

Seznam doporučené literatury:

- [1] <https://education.lego.com/en-us/support/mindstorms-ev3/python-for-ev3>  
[2] <https://innodev.com.au/conversation/content/2017/11/12/programming-lego-mindstor-ms-python/>  
[3] <https://elearning.legoeducation.com/ev3-desktop>

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: **07.02.2020** Termín odevzdání bakalářské práce: **22.05.2020**  
Platnost zadání bakalářské práce: **30.09.2021**

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

prof. Ing. Michael Šebek, DrSc.  
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