

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MODULÁRNY ROBOT NA EDUKAČNÉ ÚČELY
BAKALÁRSKA PRÁCA

2023

MICHAL MIHÁLIK

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MODULÁRNY ROBOT NA EDUKAČNÉ ÚČELY
BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná Informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: RNDr. Zuzana Berger Haladová, PhD.
Konzultant: Ing. Jozef Vaško

Bratislava, 2023
Michal Mihálik



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Michal Mihálik
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Modulárny robot pre edukačné účely
Modular educational robot

Anotácia: Práca má hardvérovo-softvérový charakter. Pôjde o vytvorenie modulárneho robota, teda sady súčiastok a softvéru k nim. Z tých sa potom dajú postaviť viaceré robotické zostavy (napr. robotické rameno, štvornohý robot...) a ovládať sadou jednoduchých funkcií. Cieľom je vytvoriť nástroj, s ktorým si ľudia môžu vyskúšať zjednodušené programovanie a skladanie robotov podľa ich predstavy. V softvéri bude implementovaný nejaký druh kinematiky a interface cez I2C protokol pre komunikáciu s inými mikrokontrolérmi/robotmi. Na projekt bude dohliadať Jozef Vaško z tvorivej dielne Fablab ktorá patrí pod CVTI SR.

Vedúci: RNDr. Zuzana Berger Haladová, PhD.
Konzultant: Ing. Jozef Vaško
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.
Dátum zadania: 24.10.2022

Dátum schválenia: 24.10.2022
 doc. RNDr. Damas Gruska, PhD.
 garant študijného programu

.....
 študent

.....
 vedúci práce

Čestné prehlásenie: Čestne prehlasujem, že som túto prácu vypracoval samostatne iba s použitím uvedených zdrojov a na základe konzultácií so školiteľkou a konzultantom.

.....

Pod'akovanie: Chcem podakovat' moje školiteľke RNDr. Zuzane Berger Haladovej, PhD. za jej čas a trpezlivosť. Ďalej chcem podakovat' môjmu konzultantovi Ing. Jozefovi Vaškovi a Mgr. Pavelovi Petrovičovi, PhD. za skúsenosti a vedomosti s ktorými sa so mnou podelili. Na záver by som chcel pod'akovat' mojim rodičom, ktorí to so mnou celý ten čas vydržali.

.....

Abstrakt

Práca sa venuje robotike. Výsledkom je modulárny robot, ktorý sa používa ako vyučovací pomôcka. Vychádza z existujúceho projektu MoKraRosA, ktorý vylepšuje. Ponúka niekoľko novovytvorených alebo vylepšených funkčných modulárnych súčiastok pre robota a softvér na ich ovládanie. Riešenie prináša funkčnú robotickú stavebnicu, ktorá sa v budúcnosti dá upravovať a rozširovať. Motiváciou projektu bolo vytvorenie učebnej pomôcky pre vyučovanie programovania a robotiky pre stredné školy a dospelých.

Kľúčové slová: robotická stavebnica, robot, modulárny, edukácia

Abstract

The work is devoted to robotics. The result is a modular robot that is used as a teaching aid. It is based on the existing MoKraRosA project, which it improves. It offers several newly created or improved functional modular parts for the robot and the software to control them. The solution provides a functional robotic kit that can be modified and expanded in the future. The motivation of the project was the creation of a teaching aid for teaching programming and robotics for secondary schools and adults.

Keywords: robotic kit, robot, modular, education

Predhovor

Táto práca vznikla popri mojej práci v dielni Fablab. Môj konzultant mi tento projekt navrhol ako možnú bakalársku prácu. Fablab vždy hľadá nové typy technológií a vyučovacích metód. Projekt ma hlavne zaujal svojím využitím ako vzdelávacia pomôcka, takto dostal projekt zaujímavý rozmer. Táto práca je hlavne určená ľuďom ktorí majú záujem o robotov určených na vzdelávanie a DIY robotiku. Tento projekt bol od začiatku zamýšľaný ako open-source projekt, takto sa môže ľahko vyvíjať a verejne používať.

Obsah

Predhovor	vii
Úvod	1
1 Očakávané použitie	3
1.1 Fablabu	3
1.2 Cieľové skupiny	3
1.3 Predchodca tohto projektu - MoKraRosA	3
1.4 Poučenia z predchádzajúceho projektu	4
2 Prehľad problematiky	5
2.1 Podobné práce	5
2.1.1 Pôvod edukačných robotov	5
2.1.2 LEGO Mindstorms	6
2.1.3 Otto DIY	7
2.1.4 Kitronik :MOVE mini mk2	8
2.1.5 Ozobot Evo	9
2.1.6 Makeblock mBot2	10
2.1.7 Hodnota tohto projektu	10
2.2 Teória	11
2.2.1 Robotická kinematika	11
2.2.2 I^2C	12
2.3 Prehľad technológií	13
2.3.1 Arduino	13
2.3.2 Servo motory	13
2.3.3 3D printing	14
2.3.4 LEGO Technic štandardy	16
3 Návrh a implementácia	17
3.1 Hardvér	18
3.2 Softvér	23

4 Použitie v praxi	27
4.1 Konštrukcia robota	27
4.2 Nastavenie systému robota	28
4.3 Testovanie prototypu v role používateľa	28
4.3.1 Zostava robota pre testovanie	28
4.3.2 Testovací program	29
4.4 Závery z testovania	33
Záver	35
Literatúra	37
Príloha A	41

Zoznam obrázkov

1.1	robot MoKraRosA v zostave pavúk	4
2.1	Seymour Papert s robotom Turtle	6
2.2	sada LEGO Mindstorms Robot Inventor	7
2.3	dvojnohý robot OTTO	8
2.4	robot :MOVE mini mk2	9
2.5	robot Ozobot Evo	9
2.6	robot mBot2	10
2.7	Pozicionálne servo SG90	14
2.8	3D tlačiareň Original Prusa Mini	16
3.1	diely hlavného modulu bez elektroniky	19
3.3	modul koleso	20
3.4	modul kliešte	20
3.5	modul ultrazvukový senzor	21
3.6	modul Mp3 prehrávač	21
3.7	modul ultrazvukový senzor	21
3.8	Nano Strong doska	22
3.9	náčrt obvodu prototypu	23
3.10	diagram knižnice Modbot	24

Úvod

V predkladanej práci vytvárame modulárneho robota Modbot na edukačné účely. Hra od nepamäti patrí k základným spôsobom oboznamovania sa človeka so svojím prostredím a novými poznatkami. Silu chute hrať sa využívame ako neoddeliteľný prvok v kurzoch programovania a robotiky pre študentov stredných škôl a dospelých. Im má slúžiť aj Modbot ako edukačná hračka a sprievodca v objavovaní možností programovania a robotiky. Edukačné roboty začali vznikať hneď po vzniku prvých programovacích jazykov zameraných na vzdelávanie a existujú už niekoľko desiatok rokov. Seymour Papert, tvorca počítačového jazyku Logo vo svojej knihe[30] uvádza:

Čo nezvládne ozubené koleso, môže počítač. Počítač je Proteus strojov. Jeho podstatou je jeho univerzálnosť, jeho sila simulovať. Pretože môže mať tisíc podôb a môže slúžiť tisíciam funkcií, dokáže osloviť tisíc chutí. (preklad z angl. Michal Mihálik)

Týmto chcel poukázať na možnosti, ktoré počítače môžu poskytnúť aj vo vzdelávaní.

V dnešnej dobe existuje viacero rôznych robotických stavebníc zameraných na vzdelávanie v oblasti informatiky. Využívajú sa vo vzdelávacích inštitúciách, ale nájdu sa aj v rukách bežných ľudí. Tieto stavebnice ponúkajú vzdelávanie jednoduchou hravou formou, kde dieťa alebo aj dospelí môžu sami skúmať a skúšať ako fungujú roboty.

Rôzne obmedzenia existujúcich stavebníc nás motivovali vytvoriť vlastnú stavebnicu. Naša práca ponúka alternatívu, niečo nové a odlišné. Stavebnica je súčasťou projektu pre dielňu Fablab. Robotická stavebnica Modbot nadväzuje na starší projekt MoKraRosA a vylepšuje niektoré z jeho pôvodných súčastí. Podľa požiadaviek dielne musí byť stavebnica modulárna a musí byť dostatočne jednoduchá na použitie vo vzdelávaní. Tieto dve veci usmerňovali ale zároveň aj sťažovali vývoj, ktorý však priniesol zmysluplný výsledok. Naším záujmom a vkladom v tejto práci bol zameranie sa na modularitu robota a možnosti využitia tejto modularity pri vzdelávaní.

Kapitola 1

Očakávané použitie

V tejto kapitole vysvetľujeme motiváciu vzniku projektu vytvorenia nášho modulárneho robota (Modbot). Opisujeme cieľové skupiny, ktorým je práca s robotom určená a vstupné očakávania definované pred vytvorením robota. Vysvetľujeme základ, na ktorom sme stavali - predchodcu tohto projektu a opisujeme vylepšenia prinášané našou prácou.

1.1 Fablabu

Projekt modulárneho robota na edukačné účely bol vytvorený v organizácii Fablab. Je to verejná dielňa fungujúca pod sekciou vedeckých knižníc Centra vedecko-technických informácií Slovenskej republiky (ďalej CVTI SR)[21]. CVTI SR spadá pod Ministerstvo školstva, vedy, výskumu a športu Slovenskej republiky. Fablab Bratislava je postavený na Charte Fab[1], ktorá bola vytvorená 20.októbra 2012 po otvorení prvého Fablab priestoru na univerzite MIT v Bostone. Fablab Bratislava poskytuje priestor na rozvíjanie projektov a nápadov, použitie strojov zdarma a viacero kurzov a workshopov na rozvíjanie vzdelania o rôznych technológiách a výrobných procesoch.

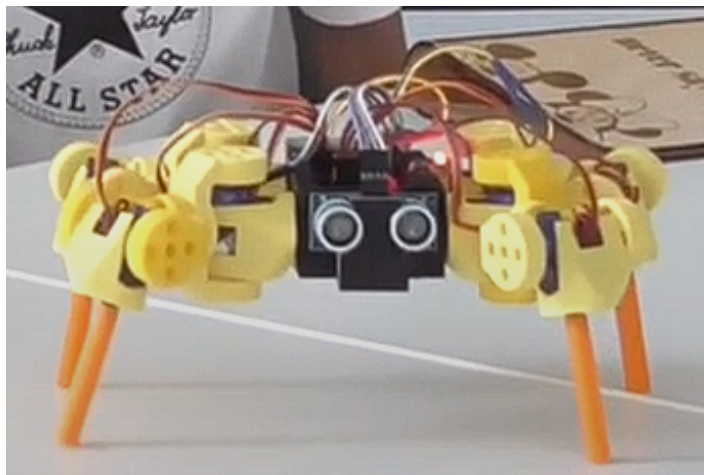
1.2 Cieľové skupiny

Robot má nájsť využitie v kurzoch robotiky prebiehajúcich vo Fablabe. Kurzy majú rôznu náročnosť v závislosti od vekovej kategórie účastníkov. Tento robot je určený pre kurzy žiakov stredných škôl a dospelých. Tí už zvyčajne majú mentálnu kapacitu na pochopenie zložitejších procesov a postupov potrebných na skladanie a programovanie takýchto robotov. Taktiež už majú potrebné vedomosti o základnej geometrii a aritmetike používanej pri programovaní robota, napr. súradnicové osi, záporné čísla a uhly.

1.3 Predchodca tohto projektu - MoKraRosA

Modulárny krahulský robot s Arduino, alebo MoKraRosA[3], je projekt, ktorý bol vytvorený na Letnom sústreďení talentovanej mládeže v Krahuliach v júli 2019. Projekt spočíval v navrhnutí a naprogramovaní modulárneho robota. Konečná zostava bola v tvare štvornohého

pavúka. Kvôli malému záujmu a komplikovanému zaobstarávaniu súčiastok v rámci Fablabu tento projekt ostal bez ďalšieho vývoja. Jeden exemplár robota sa stále aktívne používa ako ukážka vo Fablabe. Pavúk mal viacero verzií záverečného kódu. Tieto verzie boli spojené do jedného programu. V ňom sa dá robot používať v dvoch módoch. Prvý mód je priame ovládanie. Druhý mód je na vytváranie sekvencie pohybov, ktorá sa zapamätá a následne sa môže prehrať.



Obr. 1.1: robot MoKraRosA v zostave pavúk

1.4 Poučenia z predchádzajúceho projektu

Z predchodcu tohto projektu vyplynulo veľa inšpirácie. Vďaka nazbieraným skúsenostiam bolo možné vyhnúť sa viacerým problémom a vylepšiť celkový dizajn robota. Medzi hlavné problémy patrili:

1. Hmotnosť dielov - zamerali sme sa na odstránenie materiálu v miestach, kde nebol potrebný pre stabilitu a integritu súčiastok.
2. Stiesnený hlavný modul - rozšírili sme hlavný modul, aby vznikol väčší priestor pre elektroniku.
3. Ťažkopádny mechanizmus skladania kĺbov - zamerali sme sa na zjednodušenie konštrukcie kĺbov a odstránenie nepotrebných súčiastok.
4. Vyššie nároky na znalosť programovania - vytvorili sme knižnicu na zjednodušenie programovania a ovládania tohto robota.

Kapitola 2

Prehľad problematiky

V tejto kapitole sú predstavené niektoré podobné práce z komerčného, DIY (úrob si sám”) alebo akademického priestoru, s ktorými máme skúsenosti, potrebná teória a prehľad použitých technológií. Zdôvodňujeme hodnotu tohto projektu medzi ostatnými podobnými prácami.

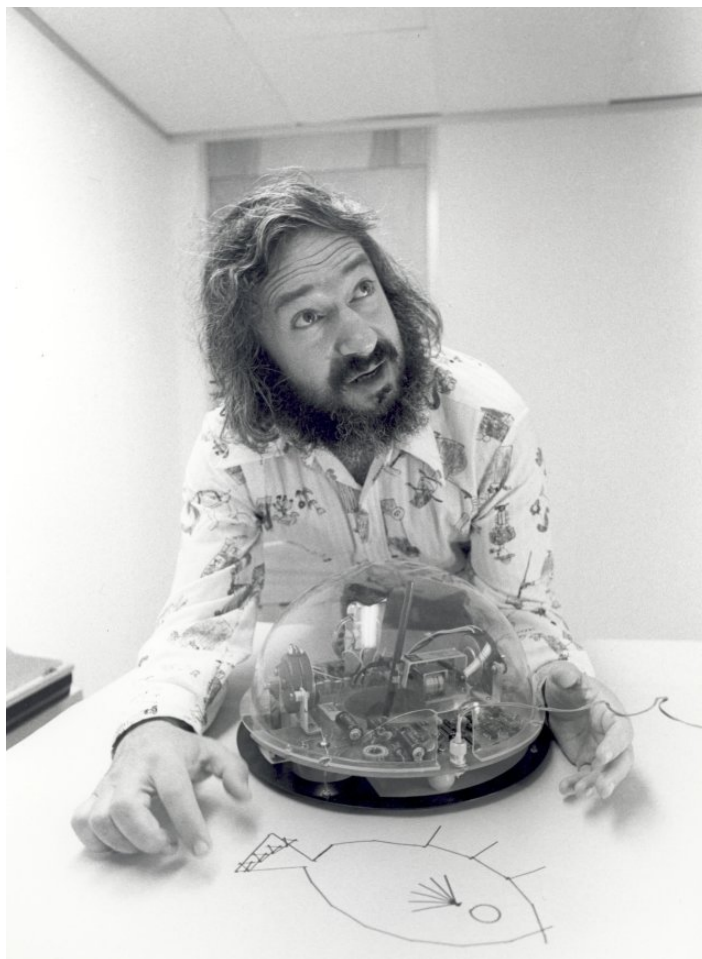
2.1 Podobné práce

Existuje viacero produktov s edukačným zameraním ako náš robot Modbot. Niektoré z nich, ktoré sú používané vo Fablabe, opíšem po krátkom predstavení pôvodu edukačných robotov.

2.1.1 Pôvod edukačných robotov

Za počiatok edukačných robotov sa dá považovať rok 1970. Vtedy Tom Callahan z MIT vytvoril prvého robota *Turtle*. Vytvoreniu tohto robota predchádzalo vytvorenie programovacieho jazyku *Logo*[15][14], ktorý bol vytvorený vo firme BBN Technologies Seymourom Papertom, Wallym Feurzeigom a Cynthiou Solomonovou. Papert dostal prvotný nápad počas práce ako konzultant pre firmu BBN Technologies na projekte pre Americké námorníctvo. Projekt vedený Wallym Feurzeigom bol zameraný na skúmanie, ako sa dajú použiť počítače na tréningovanie námorníkov. O tento nápad sa Papert podelil s Wally Feurzeigom a Cynthiou Solomonovou. Títo ľudia sa neskôr podieľali na rozvíjaní Loga. V roku 1962 Wally začal pracovať v BBN Technologies a v roku 1965 tam založil oddelenie edukačnej technológie. Skúsenosti firmy s prácou na programovacom jazyku JOSS[5] a ich derivátmi TELCOMP[6] a STRINGCOMP[4] ich viedli k presvedčeniu, že dokážu vytvoriť jazyk s nízkym prahom vstupu a vysokým stropom expertízy. Tento problém vyriešil Papert práve jazykom Logo. Jazyk fungoval na počítačoch PDP. Nevýhodou týchto počítačov bola ich vysoká cena a to, že ich monitory nepostačovali požiadavkám Loga. Preto Papert prišiel s nápadom vytvoriť robota Turtle. Zakomponovanie korytnačej grafiky do Loga tiež ponúkalo potenciál skúmať nové matematické a geometrické nápady. Pred pridaním korytnačej grafiky bol jazyk Logo v podstate programovacím jazykom, ktorý dovoľoval študentom manipulovať zoznamy. Korytnačia grafika sa stala dominantnou časťou Loga do takej miery, že jazyk Logo a Turtle

graphics sa neskôr stali synonymami. Prvú špecifikáciu robota Turtle s názvom LOGO Ambulatory Executor napísal v auguste 1969 Mike Paterson. Podľa týchto špecifikácií vytvoril Tom Callahan prvého robota Turtle 2.1. Jazyk Logo ani robot Turtle sa neuchytili vo vzdelávaní až do začiatku 80. rokov. V tom čase sa rýchlo začali roširovať osobné počítače, ktoré napomohli Logu a robotovi Turtle dostať sa do škôl. V roku 1980 Papert vydal svoju knihu *Mindstorms*, ktorá vysvetľuje koncept Loga a robota Turtle a dôležitosť počítačov vo vzdelávaní.[31]

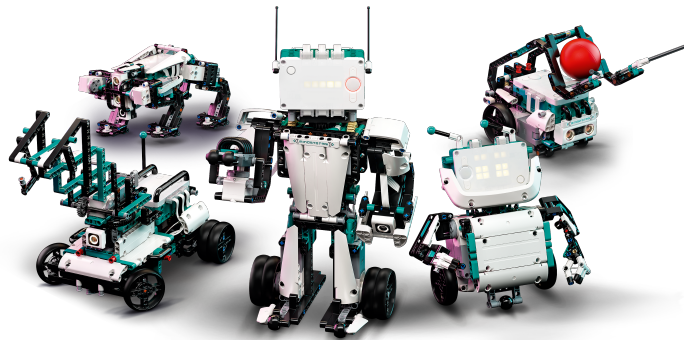


Obr. 2.1: Seymour Papert s robotom Turtle

2.1.2 LEGO Mindstorms

V roku 1985 založil Seymour Papert, Mitchel Resnick a Steve Ocko firmu *Microworlds Learning*, v ktorej začali pracovať na novom type hračky[33][29][13]. Chceli vytvoriť hračku umožňujúcu deťom vybudovať veci a stroje podľa ich predstavy, ktoré sa dali ovládať počítačom. Po dlhšom uvažovaní sa rozhodli použiť ako základ hračky LEGO kocky, vzhľadom na to že LEGO je hravé, funguje dobre ako herný systém a už existovali súčiastky ako ozubené kolieska a motory, ktoré LEGO vyrábalo. Ešte v tom roku začali spolupracovať so spoločnosťou LEGO. Neskôr sa Papert a Resnick presťahovali do MIT Media Laboratory, kde vytvorili Epistemology and Learning Group. LEGO sa v tom istom čase stalo sponzorom laboratória, takže projekt predtým korporátny sa stal akademickým. Prvá generácia programovateľnej

kocky bola vytvorená v roku 1989 a bola pomenovaná 6502 Programmable Brick kvôli procesoru, ktorý používala. Bola bezdrôtová a bežal na nej vlastný LOGO interpretér. Ďalšia bola vyvinutá v roku 1995 s menom Pocket Programmable Brick, tiež nazývaná grey brick. V roku 1996 bola vyvinutá Red Programmable Brick tiež nazývaná Model 120 Programmable Brick. Tieto kocky boli nadizajnované tak, aby boli flexibilné, čím umožňovali použitie v mnohých situáciách. Veľmi dôležitým bodom pre dizajn kocky bolo prepojenie s reálnym svetom. Kocky mali viacero I/O portov na pridanie rôznych senzorov snímajúcich svetlo, teplotu, alebo zvuk. Vznikol tlak urobiť kocku jednoduchú, lacnú a ľahko vyrobitelnú. Na softvérovej strane bolo posudzovaných viacero programovacích prostredí pre použitie s programovateľnou kockou, tradičné textové, ale aj grafické prostredia. Prvé dve generácie boli programované v textovom LOGO rozhraní. Pre ďalšiu generáciu bolo vyvinuté prostredie LogoBlocks, grafická verzia LOGO rozhrania. V tomto rozhraní bolo komplikovanejšie vytvárať zložitejšie programy, takže neskôr bolo vytvorené hybridné rozhranie. Na základe tohto výskumu LEGO začalo vytvárať svoj hrací set LEGO Mindstorms, čo trvalo ešte pár rokov. Nakoniec v roku 1998 LEGO Mindstorms prišlo do obchodov. Základom setu bola programovateľná kocka *Robotic Command eXplorers* alebo v skratke *RCX*. V ďalších dvoch desaťročiach bolo vytvorených viacero nových súprav s novými programovateľnými blokmi ako NXT, NXT 2.0, EV3 a Robot Inventor set [34]. V roku 2019 bol tiež oznámený set Spike Prime pod produktovou líniou Lego Education. V októbri 2022 Lego Group oznámilo, že Lego Mindstorms línia produktov bude ukončená koncom roka 2022 s poskytovaním podpory ešte ďalšie dva roky[34].



Obr. 2.2: sada LEGO Mindstorms Robot Inventor

2.1.3 Otto DIY

OTTO je dvojnohý robot, ktorého tvorcom je Camilo Parra Palacio[2]. Inšpirácia na vytvorenie tohto robota prišla z niekoľkých predošlých projektov zameraných na dvojnohých robotov. Prvým z nich bol BoB, alebo Bob the Biped[28], ktorý bol vytvorený Kevinom Biagininim. Mal jednoduchú kockovú hlavu, v ktorej sa nachádzal mikrokontrolér, batérie a ultrazvukový senzor a dve nohy, každá s dvoma servami. Robot mal veľa nedostatkov, napríklad ťažkopádny pohyb, káble trčiace zozadu a potrebu dodatočného opracovávania súčastiek. Po

BoBovi prišiel Zowi, ktorého pôvodným tvorcom bol Javier Isabel Hernandez. Zowi prišiel s viacerými vylepšeniami ako nabíjateľnou batériou, LED svetlami namiesto úst, bluetooth spojením, Arduino kompatibilnou doskou, aplikáciou a blokovým kódovaním. Firma BQ vypustila tohto robota na trh ako hračku s dielmi tlačenými z ABS plastu, čo vylepšilo kvalitu dielov. Tento robot bol ale stále drahý, jeho cena 99,- EUR komplikovala dostupnosť a veľké servá sťažovali skladanie robota dokonca aj pre profesionálov. Zowi ale vytvoril dobrý základ pre jeho nasledovníka, robota Tito, priameho predchodcu dnešného robota OTTO. Tvorca robota Tito, Camilo Palacio a jeho tím zobrali najlepšie veci z oboch robotov, podopĺňali medzery a výsledkom bol robot Tito. Tito bol potom ďalej vylepšovaný a nakoniec z neho vznikol robot OTTO. OTTO pozostáva zo súčiastok, ktoré sa dajú zohnať kdekoľvek a všetky diely boli doladené tak, aby boli čo najlacnejšie, najdostupnejšie a v prípade tlačených dielov jednoduché na tlač. Toto dalo OTTOvi veľkú výhodu na trhu a OTTO má teraz okolo seba veľkú komunitu ľudí, ktorí ho programujú alebo upravujú podľa svojej fantázie. Meno OTTO, pochádza zo slova "Robotto", čo v japončine znamená robot.

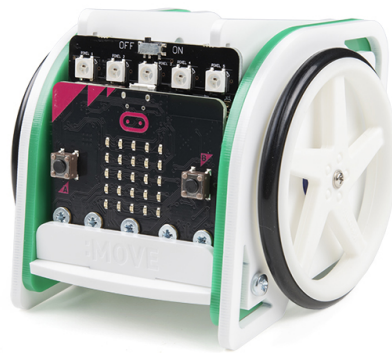


Obr. 2.3: dvojnohý robot OTTO

2.1.4 Kitronik :MOVE mini mk2

Kitronik :MOVE mini mk2[12] je dvojkolesové autíčko s Microbit mikrokontrolérom[8][20][19]. Microbit je mikrokontrolér, ktorý vznikol ako súčasť BBC Make It Digital iniciatívy[9]. V roku 2015 sa do tejto iniciatívy zapojilo 29 partnerov, medzi nimi Microsoft, Arm a Samsung, ktorí vytvorili platformu Microbit. Okolo milión Microbitov bolo distribuovaných v roku 2016 rôznym školám a knižniciam Spojeného Kráľovstva. V októbri toho roku bola vytvorená Microbit edukačná nadácia so zámerom rozšíriť Microbit aj mimo Veľkú Britániu.

Kitronik :MOVE mini mk2 sa skladá z Microbitu, ktorý už v sebe má viacero senzorov ako gyroskop, akcelerometer a teplomer, ktorý je vsadený do rodičovskej dosky. Tá ho rozširuje o farebný senzor na podvozku, pásik LED svetiel na vrchu auta, držiaka na AA batérie a dva kontinuálne servo motory. Vďaka tomu, že Microbit má v sebe zabudovaný bluetooth čip, je možné auto ovládať z diaľky, čo výrazne rozširuje možnosti použitia auta.



Obr. 2.4: robot :MOVE mini mk2

2.1.5 Ozobot Evo

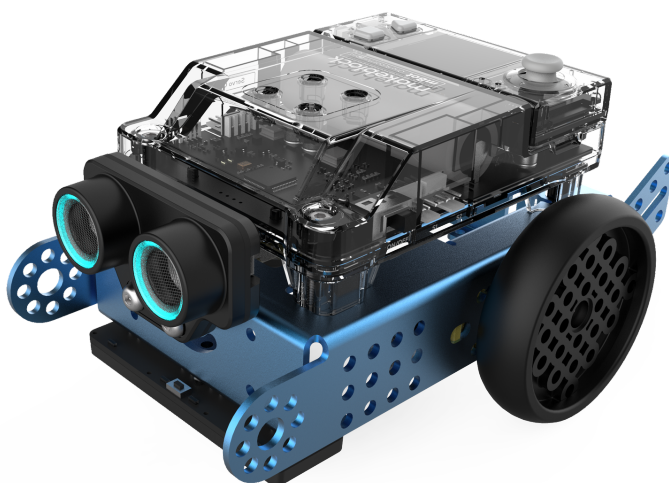
Ozobot[23] je firma zameraná na výrobu novej generácie robotov vyzerajúcich ako pôvodné korytnačky len v menšej mierke. V robotovi, ktorý sa zmestí do dlane ruky sa nachádza mikrokontrolér, Li-Po batéria, viacero senzorov blízkosti, farebné senzory, LED svetlá, bluetooth čip, mikroservá pre kolesá a malý reproduktor, všetko zahrnuté v kopulovitom priehľadnom obale. Tieto roboty sa dajú programovať dvoma spôsobmi. Prvý je za použitia farebných senzorov. Používateľ kreslí pomocou farebných fixiek čiary v určitom poradí podľa návodu a robot po ich prejdení vykoná sériu daných príkazov. Druhý spôsob je za použitia aplikácie Ozobot Blockly - aplikácie na blokové programovanie. Tento robot je zameraný na použitie v školách, nie pre jednotlivcov, takže sa predáva len po sadách 12 alebo 18 robotov. Tomuto zodpovedá aj jeho vyššia cena.



Obr. 2.5: robot Ozobot Evo

2.1.6 Makeblock mBot2

Makeblock[16] je firma pochádzajúca z Číny, zameraná na vyrábanie edukačných robotov a softvéru. Firma svoj prvý produkt začala vyvíjať ako crowdfundingový projekt na Kickstarter-i v roku 2013. Od tohto roku vyvinula už viacero robotických setov a predáva ich po celom svete. Jedným z jej výrobkov je robot-auto nazývaný mBot2 [18]. Pozostáva z viacerých senzorov, ako napríklad ultrazvukový senzor, Quad farebný senzor na podvozku, dva motory s prevodovkou. Toto všetko je napájané počítačom CyberPi, ktorý má v sebe viacero senzorov, Wi-Fi, Bluetooth, malú obrazovku, mini joystick a dve tlačítka naboku. Robot mBot2 samotný má ešte viacero portov pre rozširujúce moduly a elektroniku. Na programovanie robota sa používa aplikácia mBlock[17], tiež vyvinutá touto firmou. Programovať sa v nej dá buď blokovým programovaním alebo v jazyku Python. Táto aplikácia umožňuje programovať nielen Makeblock zariadenia, ale aj veľa ďalších, ako LEGO Mindstorms, Raspberry Pi, Microbit, Arduino a iné mikrokontroléry.



Obr. 2.6: robot mBot2

2.1.7 Hodnota tohto projektu

Hoci existujú rôzne robotické stavebnice, stále je zmysluplné vytvárať nové, lebo je mnoho funkcionalít, ktoré existujúce stavebnice neponúkajú, alebo sú príliš drahé a nedostupné pre určité cieľové skupiny. Zároveň je to spôsob, ktorým sa dá veľa o robotike a vývoji robotov naučiť. Náš projekt sa opiera práve o tento rozmer učenia sa cez experimentovanie pri vývoji a zostavovaní robota. Ponúka dobrý základ, na ktorom zručnejší študenti môžu stavať aj úplne nové riešenia.

Hodnotu tohto projektu by som zhrnul do kľúčových bodov:

1. Cena - diely sú vytlačené 3D tlačiarňami, použitá elektronika je lacná (servá, arduino, baterky),
2. Modulárnosť - používateľ si môže vytvoriť vlastné diely podľa svojich požiadaviek,

3. Otvorenosť ekosystému - jeden výrobca nemá monopol na potrebné diely,
4. Jednoduchosť programovania - vďaka vytvoreným knižniciam s možnosťou pokročilejšieho kódovania je prístupný aj pre začiatočníkov.

2.2 Teória

V tejto časti vysvetľujeme použité technológie a teóriu.

2.2.1 Robotická kinematika

Kinematika[27] v robotike je o skúmaní relácií medzi súradnicami kĺbov robota a jeho rozložení v priestore. Je to jedna zo základných a klasických tém v robotike. Všeobecne rozlišujeme medzi dvoma druhmi kinematiky podľa spôsobu akým dosiahnu cieľové súradnice a to dopredná a inverzná kinematika.

Dopredná kinematika je proces vypočítania konečnej pozície robota použitím konfigurácie a štruktúry robota ako vstupu. Dá sa to predstaviť veľmi jednoducho ako sčítavanie vektorov. Inverzná kinematika[26] je proces získania potrebnej konfigurácie z cieľovej pozície a štruktúry robota ako vstupu. Inverzná kinematika je väčšinou náročnejšia na výpočet kvôli väčšiemu počtu neznámych a možných výsledkov konfigurácie. Existujú dva všeobecné prístupy, ktorými sa dajú riešiť problémy inverznej kinematiky.

Prvým z nich je analytický spôsob. Tento spôsob využíva rovnice vhodné pre danú štruktúru robota a odvodí z nich postup pre výpočet výsledkov. Druhý spôsob je numerický. Tu dostávame počiatočný odhad konfigurácie a počítame sekvenciu konfigurácií tak, aby sa odchýlka vzdialenosti od cieľového bodu približovala k nule. Tu opíšeme dva numerické postupy, ktoré sme študovali. Prvý sa nazýva Cyclic Coordinate Descent a je to veľmi jednoduchá metóda, ktorá používa iba geometrické uvažovanie na odvodenie nasledujúcich iterátov. Zámerom je posúvať každý kĺb zvlášť a postupne znižovať odchýlku medzi súradnicami cieľového bodu a koncového efektoru, to je koniec paže po poslednom kĺbe. Ukážka postupu v pseudokóde:

1. for $i = 1, \dots, k$ do
2. $q_i \leftarrow q_{i-1}$
3. for all joints $j = 1, \dots, n$ do
4. Compute $q_j^* = \arg \min_{q_j} ||f(q_i)||$
5. Set $q_{i,j} \leftarrow q_j^*$
6. return q_k

Druhý postup je s použitím databázy. Myšlienka je veľmi jednoduchá: vypočítať tabuľku súradníc pre všetky alebo časť konfigurácií dopredu a potom z tejto databázy získavať konfiguráciu pre súradnicu, ktorá je najbližšie cieľovej súradnici. Tento spôsob je výhodný v tom, že počas používania robota sa nemusia robiť nadbytočné výpočty. Nevýhodou je, že na databázu treba veľa pamäti, ktorá často nie je dostupná v kompaktných robotoch.

2.2.2 I^2C

I^2C [11] je synchronna multi-master/multi-slave sériová komunikačná zbernica. Bola vyvinutá v roku 1982 firmou Philips Semiconductors. Je široko používaná pre pripájanie periferálnych zariadení k procesorom a mikrokontrolérom v komunikácii na krátke vzdialenosti. Pôvodný produkt dosahoval obojsmerné rýchlosti 100kbit/s, čo stačilo na prenášanie dát medzi periferálnymi zariadeniami. Postupne boli pridané nové rýchlostné režimy ako Fast Mode dosahujúci 400kbit/s, High-speed Mode dosahujúci 3,4Mbit/s a Ultra Fast-mode dosahujúci 5Mbit/s. I^2C sa skladá z dvoch liniek, sériovej dátovej a sériovej časovačovej. Časovač zodpovedá za to aby všetky zariadenia posielali správy v tom istom tempe, zatiaľčo dátová linka slúži na prenášanie samotných dát. V referenčnom dizajne majú linky 7-bitové adresovanie. I^2C má iba dve roly pre zariadenia pripojené k zbernici, master alebo slave. Master je riadiaci uzol, generuje časovanie a iniciuje komunikáciu s cieľmi. Slave je cieľový uzol, dostáva časovanie a odpovedá riadiacemu uzlu, keď sa ho niečo spýta. Okrem bitov 0 a 1 dovoľuje I^2C posielanie START a STOP signálov, ktoré ohraničujú správu. Komunikácia prebieha nasledovne:

1. Správa sa začína poslaním signálu START nasledovaným 7-bitovou adresou cieľového zariadenia a nakoniec zakončená jedným bitom, ktorý reprezentuje či iniciátor komunikácie chce písať (0) alebo čítať (1) z cieľa.
2. Ak cieľ existuje, vyšle ACK bit, teda logickú nulu pre potvrdenie adresy. Riadiaci uzol potom pokračuje buď posielaním alebo prijímaním dát a cieľ funguje komplementárne. Adresy a dátové bity sú posielané s najvyšším bitom ako prvým.
3. Za každým poslaným alebo prijatým byte-om riadiaceho uzlu je poslaný ACK bit, teda potvrdenie prijatia.
4. I^2C transakcia môže pozostávať z viacerých správ. Riadiaci uzol ukončuje správu signálom STOP, ak je to koniec transakcie, alebo môže poslať ďalší signál START aby si podržalo kontrolu zbernice pre ďalšiu správu.

2.3 Prehľad technológií

Nasleduje opis technologických postupov a produktov využitých pri tvorbe robota.

2.3.1 Arduino

Arduino[10] je open-source elektronická platforma založená na ľahko použiteľnom hardvéri a softvéri. Arduino bolo vytvorené v roku 2005 na Interaction Design Institute v Ivrei ako nástroj na rýchle prototypovanie pre študentov bez znalostí elektroniky a programovania. Arduino bolo založené na staršom projekte Wiring, ktorý bol vytvorený v roku 2003. V dnešnej dobe je Arduino známe dizajnom a výrobou programovateľných dosiek s mikrokontrolérmi pre digitálne zariadenia. Keďže je táto platforma open-source, existuje veľa dosiek rozširujúcich a obmieňajúcich funkcie a vlastnosti pôvodných dosiek. Pre projekt modulárneho robota bola vybraná doska DM Nano strong, založená na doske Arduino Nano, ktorá na rozdiel od dosky Arduino Nano poskytovala väčšie množstvo programovateľných pinov a pinov pre vývod prúdu a uzemnenie. Softvér pre Arduino pozostáva z Arduino IDE, ktoré slúži na programovanie jednotlivých typov dosiek. Pre programovanie Arduino[32] dosiek sa používa Arduino API, teda programovací jazyk založený na C++ pre dosky Arduino, ktorý pozostáva z troch častí:

1. funkcií: pre ovládanie Arduino dosky a výpočty,
2. premenných: Arduino konštanty, dátové typy a konverzie,
3. štruktúry: časti Arduino kódu ako sketch funkcie `loop()` a `setup()`, štruktúry pre cykly a podmienky, aritmetické operátory a porovnávacie operátory.

V Arduino projekte sa program nazýva *sketch*. Sketch je súbor, do ktorého sa píše program. Tento súbor má koncovku `.ino` a je vždy uložený v priečniku s tým istým menom. Priečinok môže obsahovať aj iné súbory. Okrem základného API pre Arduino existuje veľa knižníc, aj oficiálnych aj komunitných, ktoré rozširujú funkcionality kódu. Tieto knižnice sa dajú pripojiť k vlastnému projektu. Proces sťahovania programu na dosku je nasledovný:

1. Najprv sa program preloží do strojového kódu, toto zabezpečí Arduino IDE.
2. Strojový kód sa nahrá na Arduino. Mikrokontroléry sú väčšinou programované cez programátory, zapisovací softvér, ktorý umožňuje ovládať a vykonávať úlohy v rámci systému. Arduino má softvérový programátor nazývaný bootloader. Ten umožní správne nahranie nového programu na dosku.

Nevýhoda bootloaderu je v tom, že zaberá časť pamäte mikrokontroléru, ktorá sa tiež využíva na vlastný program, takže treba počítať s tým, že nebude môcť využiť celú pamäť.

2.3.2 Servo motory

Servo motory[24] sú aktuátory, ktoré umožňujú precízne ovládanie uhlu, rýchlosti a akcelerácie. Skladajú sa z motora a senzorov pre spätnú väzbu o pozícii. Pre robota Modbot boli

použité pozicionálne servá sg90, ktoré sú jedny z najčastejšie používaných serv pre malé, ale aj väčšie projekty. Tieto servá majú 180° rozsah otočenia a ovládajú sa vysielaním niekoľko milisekundových pulzov so šírkou 500-2400 mikrosekúnd, kde 1 ms pulz je 0° a 2 ms pulz je 180°. Okrem pozicionálnych serv existujú kontinuálne servá. Týmto servám sa namiesto pozície nastavuje rýchlosť otáčania.



Obr. 2.7: Pozicionálne servo SG90

2.3.3 3D printing

Myšlienka 3D tlače pochádza z polovice 20. storočia [7][25]. Vtedy bol tento koncept opísaný autorom Murrayom Leinsterom v jeho krátkom príbehu *Things Pass By*. V roku 1971 Johannes F. Gottwald patentoval Liquid Metal Recorder, zariadenie na kontinuálnu tlač kovového materiálu. Zdá sa, že toto je prvý patent opisujúci 3D tlač ako ju poznáme dnes. V roku 1984 Bill Masters podal patent jeho počítačovo automatizovaného výrobného procesu a systému. Tento patent je na zázname USPTO ako prvý patent o 3D tlači v histórii. V tej dobe boli tlačiarne ešte veľmi drahé, v hodnotách stoviek tisícov dolárov, takže si ich mohli zadovážiť iba väčšie firmy. V 90. rokoch výskum pokračoval ďalej a boli vytvárané nové procesy aditívnej výroby. Začiatkom 21. storočia sa aditívna výroba začala používať aj v zdravotníctve. Vedci vo Wake Forest Institute of Regenerative Medicine vytvorili prvý orgán pomocou 3D tlače. V tom čase sa vo verejnej sfére objavilo aj veľa open-source projektov, jedným z nich je iniciatíva RepRap ktorú v roku 2005 začal Dr. Adrian Bowyer. V roku 2009 expiroval patent na Fused Deposition Modeling tlačový proces, čo umožnilo novým firmám a jednotlivcom vytvárať vlastné FDM tlačiarne, či už komerčné alebo osobné. Objavili sa firmy ako Maker-Bot v roku 2009, Ultimaker v roku 2011 a Prusa Research[22] v roku 2012, ktoré ponúkali cenovo dostupné tlačiarne pre širokú verejnosť. V roku 2012 firma Filabot vyvinula systém na uzavrenie okruhu pre plast, čím umožnili ktorejkoľvek FDM alebo FFF tlačiarňi tlač s širokým sortimentom plastov. V roku 2014 bola demonštrovaná multimateriálová, vertikálne integrovaná platforma na tlač elektroniky, čo umožnilo tlač funkčnej elektroniky s prevádzkou do 40 GHz. V dnešnej dobe sú tlačiarne používané širokou verejnosťou na výrobu súčiastok a ozdôb do domácnosti, ale aj firmami na produkciu špecializovaných súčiastok a produktov.

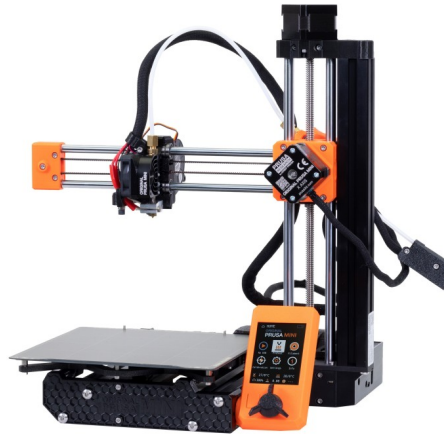
Existuje veľa druhov tlačiarň líšiacich sa stavbou, typom materiálu, s ktorým pracujú a veľkosťou. Medzi najznámejšie typy tlačiarň podľa procesu tlače patria extrúderové, práškové a stereolithografické. Okrem nich existujú ďalšie typy tlačiarň, ktoré používajú iné procesy na aditívnu výrobu. V tejto práci sa zameriavame na extrúderové, keďže máme s nimi skúsenosť a použili sme ich na výrobu súčiastok pre robota. Extrúderové tlačiarne sa všeobecne skladajú z niekoľkých častí:

1. tlačová hlava, ktorá v sebe obsahuje ohrievacie teliesko, ventilátory na chladenie materiálu, trysku na pokladanie materiálu a extruder, ktorý slúži na kontrolovaný prívod materiálu,
2. krokové motory a koľajnice, ktoré umožňujú hlave presný pohyb v 3D priestore,
3. vyhrievaná tlačová plocha,
4. riadiaci počítač,
5. zdroj zásobujúceho prúd do celého stroja.

Proces tlače pre extrúderové tlačiarne je nasledovný:

1. materiál vo forme struny nazývanej filament je vsúvaný do tlačovej hlavy,
2. tam je zahrievaný na tavnú teplotu daného materiálu,
3. a potom pokladaný na podložku v žiadanom tvare,
4. materiál sa takto ukladá po vrstvách od spodu po vrch,
5. občas je pridávaná podporná štruktúra tlačená zároveň s modelom, kvôli niektorým častiam modelu, ktoré prevísajú nad voľný priestor a potrebujú podporu.

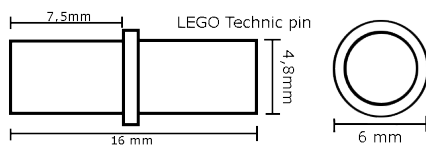
Pred tlačou sa 3D modely z rôznych typov súborov musia preložiť do jazyka Gcode, ktorý sa používa na programovanie tlačiarne. Toto sa deje v programoch nazývaných *slicer*. Tieto aplikácie v sebe väčšinou zahŕňajú aj databázu tlačiarň a materiálov a umožňujú rôzne nastavenia tlače modelu. Slicer zoberie 3D model a nareže ho na vrstvy a pre každú vrstvu na základe nastavení vypočíta najlepšiu cestu akou sa má materiál pokladať. Výsledný súbor v Gcode sa stiahne do tlačiarne. Na výrobu dielov Modbota boli použité tlačiarne Original Prusa Mini ?? a Original Prusa i3 MK3S a slicer PrusaSlicer.



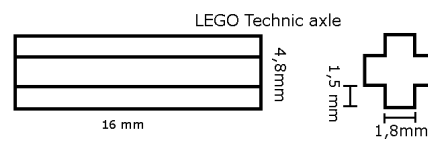
Obr. 2.8: 3D tlačiareň Original Prusa Mini

2.3.4 LEGO Technic štandardy

Na spájanie jednotlivých modulov boli použité dva typy súčiastok zo sád LEGO Technic, Pin 2.9a a Axle 2.9b. Tieto súčiastky sme vybrali preto, lebo umožnili jednoduché spájanie modulov bez použitia skrutiek alebo iných komplikovanejších spôsobov pripojenia. Jedným z dôvodov bolo aj to, že používanie týchto súčiastok je veľmi intuitívne. Nižšie ponúkame diagramy súčiastok.



(a) diagram LEGO Technic pin



(b) diagram LEGO Technic axle

Kapitola 3

Návrh a implementácia

V kapitole rozoberáme vznik návrhu robota, prebieh jeho výroby, problémy, ktoré bolo nutné riešiť a vylepšenia oproti predošlému projektu.

V tomto projekte bolo potrebné vytvoriť platformu, s ktorou sa dá názorne vysvetliť žiakom, čím sa zaoberá robotika a ktorá umožňuje vyskúšať si aplikovanie spoznaných princípov na reálne problémy.

Na začiatku prebehla konzultácia so zadávateľom, v ktorej sme identifikovali hlavné ciele a požiadavky, vzhľadom na očakávané použitie robota. Zoznámili sme sa s predchádzajúcim projektom MoKraRosA, na ktorý táto práca nadväzuje a skúsenosťami, ktoré sa ukázali pri jeho používaní. Vyjasnili sme, aké sú očakávania na funkcionality robota a preskúmali sme možnosti využitia robota vo vzdelávacích kurzoch vo Fablab. Následne prebehla ešte konzultácia s profesorom Pavlom Petrovičom, ktorý viedol pôvodný projekt. Výsledkom konzultácie boli získané cenné informácie o výbere elektronických súčiastok pre robota, o postupe pri návrhu a o pôvodnej vízii do budúcnosti.

V nasledujúcom kroku prebehla analýza, na aké požiadavky a potreby reagujú iné existujúce robotické stavebnice (zoznam stavebníc uvádzame v kapitole 2.1 Podobné práce). Ukazuje sa, že najčastejšie ide o nasledovné štyri charakteristiky:

1. variabilita použitia (používateľ môže prispôbiť robota, aby mu pomáhal riešiť rôzne úlohy),
2. jednoduchosť skladania robota,
3. jednoduchosť príkazov na programovanie,
4. modularita (t.j. možnosť zameniť niektoré diely za iné, alebo si vytvoriť vlastné prvky).

Práca s robotom má byť pre žiakov primerane náročná, aby ich neodradila prílišná zložitosť skladania a programovania. Na druhej strane však nesmie byť ani príliš jednoduchá s obmedzenými možnosťami, aby žiakov dostatočne motivovala k učeniu sa hrou s robotom.

Na základe konzultácií a analýzy vznikli dve kľúčové rozhodnutia pre túto prácu:

1. Bude ponechaný systém spájania modulov robota pomocou LEGO Technic pinov, ktorý bol veľmi jednoduchý a funkčný. Podobne sme zachovali väčšinu elektroniky, ktorá bola

použitá v projekte MoKraRosA. Redizajn hardvéru sa sústredil na vylepšenie vlastností robota vzhľadom na edukačný účel.

2. Softvér bude prepísaný od základu, keďže pôvodný kód bol jednoliaty a nedal sa rozdeliť na jednoduchšie a modulárnejšie časti.

3.1 Hardvér

Vychádzajúc zo vstupných konzultácií a analýz sme urobili nasledovné kroky.

Nanovo bolo potrebné vytvoriť obaly na elektroniku robota. Obaly kĺbov nemali dobrý systém spájania so servami a krabica s arduinom a batériami bola príliš stiesnená po napojení všetkých káblov.

Keďže sme ponechali systém spájania modulov pomocou LEGO Technic pinov, uvažovali sme o možnosti vytvárať moduly v programe zvanom Part Designer od firmy BrickLinkr, ktorý umožňuje dizajnovat' vlastné tehličky s parametrami stavebnice LEGO. Zámerom bolo rýchlo vytvoriť prototypy s danými mierkami pre LEGO piny a osy. Výhodou pre edukačné účely by bolo, že v tomto jednoduchom softvéri by si mohli súčiastky dotvárať jednoduchým spôsobom samotní žiaci. Po dôkladnom testovaní možností tohto programu a pokusnej 3D tlači vzniknutých dielov sme sa tohto postupu museli vzdať, pretože aj keď program Part Designer vedel exportovať modely v .obj formáte, modely boli defektné. Steny modelov neboli dobre uzavreté a strany často prechádzali navzájom cez seba, čo pridávalo neprimerane veľa práce na opravách chýb pred tlačou.

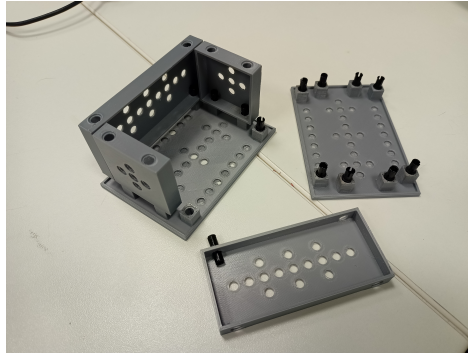
Modely sme napokon od základu vytvárali v programe Blender. Využili sme niektoré existujúce modely súčiastok voľne dostupné na internete. Ide napríklad o servá a LEGO piny, pomocou ktorých sme vytvorili jednotlivé diely. Prebehla iterácia viacerými verziami dielov pre každý modul, dokým nevznikla finálna verzia dielov, ktoré boli uznané za dostatočné:

1. Hlavný modul:

- Hlavný modul je navrhnutý tak, aby sa dal vytlačiť po častiach a potom spojiť dokopy pomocou LEGO pinov vytvárajúc dutý kváder. Pozostáva z troch typov dielov, jeden pre vrch a spodok, druhý pre dlhšiu stranu krabice a tretí pre kratšiu stranu krabice. Týmto sme dosiahli nižšiu spotrebu materiálu a jednoduchšiu tlač dielov. Každý diel bol navrhnutý s dierami pre piny na pripojenie modulov. Diely pre vrch alebo spodok boli tiež vytvorené tak, aby mali medzi sebou a ostatnými dielmi medzeru pre káble, ktoré budú viesť k ostatným modulom. V prvých prototypoch dielov sa vyskytol problém s pinmi, ktoré dobre nedržali v stenách. Preto sme urobili steny okolo dier hrubšie, aby udržali piny. Vnútri tejto krabice sa nachádzajú batérie, menič napätia, a mikrokontrolér.

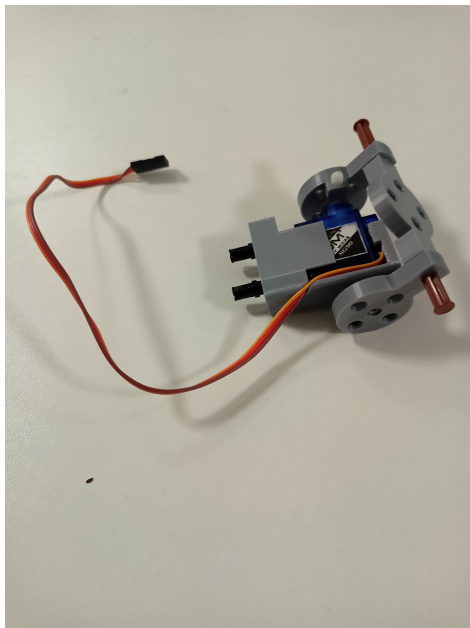
2. Modul Kĺb:

- Tento modul je tvorený zo serva a troch typov dielov. Prvý typ, spodný diel, obopína servo a má diery pre piny. Druhý typ, bočný diel, sa pripája na rameno

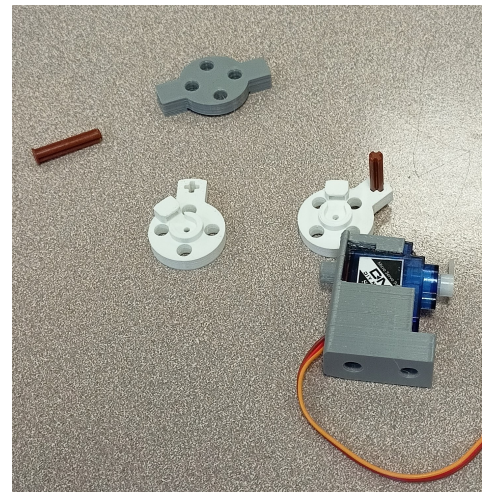


Obr. 3.1: diely hlavného modulu bez elektroniky

serva alebo na spodný diel. Takto vytvárame dva oporné body na osi otáčania pre mechanizmus. Tento diel je navrhnutý tak, aby bol obojstranný a má malé puto, ktoré umožňuje rýchle a spoľahlivé pripájanie a odpájanie ramena serva. Má tiež diery pre piny. Tretí typ, stredný diel, prepája bočné diely a má diery pre piny. Bočný a stredný diel sa počas vývoja veľmi nemenili, dolad'ovala sa na nich iba hrúbka steny a veľkosť dier pinov a puta na rameno serva. Spodný diel mal pár revízií, bol rozšírený priestor pod spodnou stranou serva pre káble a bola vytvorená malá medzera, na ktorej boli káble zachytené ako ochrana proti zatrhnutiam za kábel.



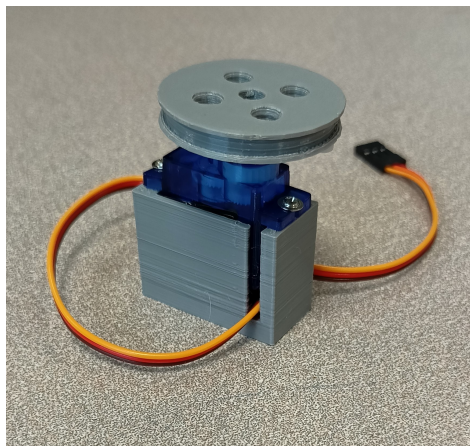
(a) modul kĺb so servom



(b) rozložený modul kĺb

3. Modul Koleso:

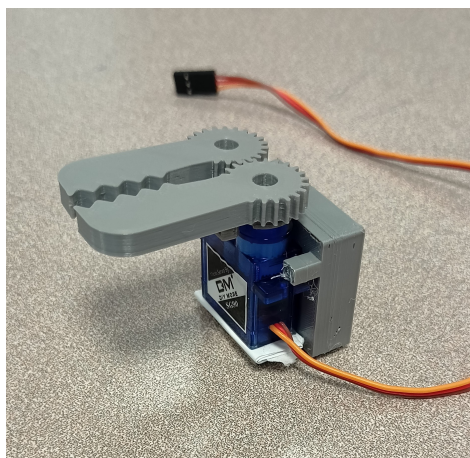
- Pozostáva z dvoch častí, jedna drží servo a druhá je nasadená na rameno serva ako koleso. Obidve časti majú diery pre piny na pripojenie ďalších modulov. Tento druh modulu je možné tiež použiť na otáčanie v zvislej osi na rozdiel od modulu Kĺb, ktorý poskytuje otáčanie v horizontálnej osi.



Obr. 3.3: modul koleso

4. Modul kliešte:

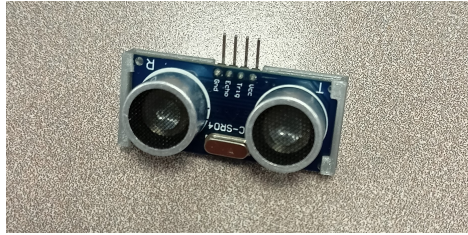
- Z veľkej časti som okopíroval mechanizmus a výzor modulu kliešťov z robota MoKraRosA. Boli urobené malé zmeny ako zväčšenie kliešťov a pridanie zúbkov pre zlepšenie úchytu.



Obr. 3.4: modul kliešte

5. Modul Ultrazvukový senzor:

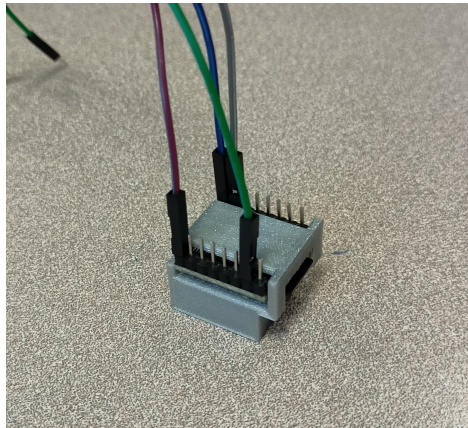
- Je zložený z ultrazvukového senzoru a dielu na držanie senzoru. Diel má diery na piny pre prepojenie s inými modulmi.



Obr. 3.5: modul ultrazvukový senzor

6. Modul Mp3 prehrávač

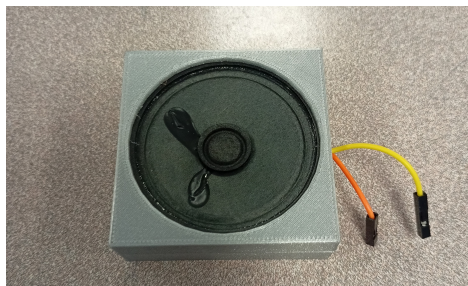
- Je tvorený z modulu DFPlayer Mini, ktorý funguje ako Mp3 prehrávač a obalu preň. Keďže doska má slot pre SD kartu, diel bol upravený tak, aby umožnil prístup k SD karte. Celý obal voľne obopína Mp3 prehrávač. Tento modul zabezpečuje prehrávanie hudby a nahrávok nahraných na SD karte. Potrebuje reproduktor, aby vydával prehrávaný zvuk.



Obr. 3.6: modul Mp3 prehrávač

7. Modul Reproduktor

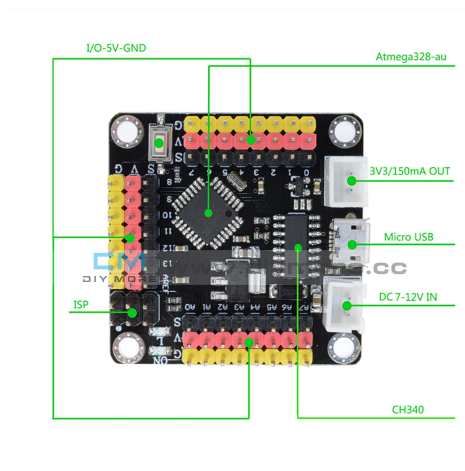
- Funguje v spolupráci s Mp3 prehrávačom. Obsahuje reproduktor s výkonom 0,5W v dvojdielnom obale. Obal sa dá pomocou pinov pripojiť na ďalšie moduly.



Obr. 3.7: modul ultrazvukový senzor

1. DIYMore Nano Strong doska

- Táto doska poskytuje 14 digitálnych dátových pinov a 8 analógovo-digitálnych dátových pinov, každý s napájacím a uzemňovacím pinom. Je to doska s mikročipom Atmega328-au, ktorý poskytuje 32KB flash pamäte a operačnú frekvenciu 16 MHz. Má Micro usb port na prepojenie s počítačom a tiež podporuje In-System Programming, skrátene ISP, čo zabezpečuje ďalších 6 pinov na doske. V jednom rohu tejto dosky sa nachádza reset tlačidlo. Táto doska je založená na Arduino Nano doske a je teda kompatibilná s Arduino vývojárskym prostredím.



Obr. 3.8: Nano Strong doska

2. SG90 servo

- Je to jedno s najbežnejšie dostupných serv na trhu. Má rozsah otočenia od 0 do 180° a točivú silu 2,5 kg/cm. Požaduje napájanie okolo 5V. Na nastavenie pozície sa posielajú pulzy cez dátový kábel, 1.5ms pulz znamená pozíciu 0°, 2 ms znamená 90° a 1 ms pulz -90°.

3. Ultrazvukový senzor HC-SR04

- tento modul obsahuje ultrazvukový prijímač a vysielač s kontrolným obvodom. Má štyri piny, napájací, uzemňovací, pin na spúšťací signál a pin na odozvu. Senzor funguje nasledovne:
 - Najprv dostane modul spúšťací signál ktorý spustí ultrazvukový vysielač.
 - Následne čaká na prijatie odrazenej zvukovej vlny.
 - Potom vyšle odozvu na základe času, ktorý prešiel od vyslania zvukovej vlny.

4. DFPlayer Mini DFR0299

- Ide o malý a lacný Mp3 modul so zjednodušeným výstupom pre reproduktor a čítačkou SD kariet. Tento modul sa dá použiť samostatne, alebo v kombinácii s

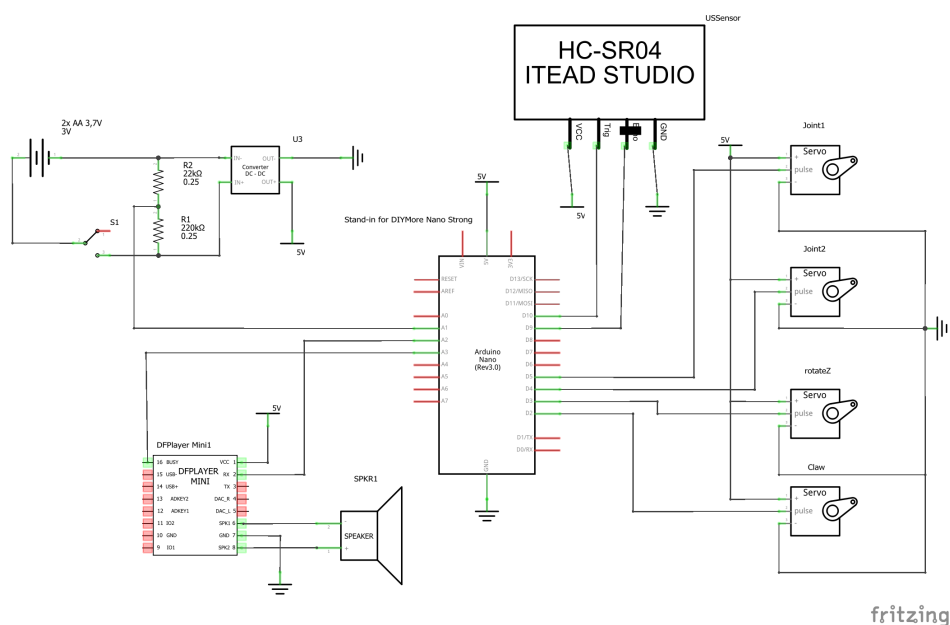
mikrokontrolérom. Poskytuje viacero vzorkovacích frekvencií, 24-bitový DAC výstup, podporu pre dynamický rozsah až 90dB, podporuje čítanie FAT16 a FAT32 súborových systémov a má viacero spôsobov ovládania. V tomto projekte je ovládaný pomocou Serial vstupného pinu pripojeného na mikrokontrolér. Existuje viacero Serial príkazov na ovládanie a získavanie informácií z modulu, ale používajú sa iba príkazy nastavenia hlasitosti a pustenia a zastavenia prehrávania skladby. Požaduje 5V napájanie.

5. Menič napätia

- Pre napájanie sú použité lítiové batérie s napätím 7,3V, takže je potrebné znížiť napätie pre mikrokontrolér a ostatné elektronické súčiastky na 5 voltov.

Celý náčrt obvodu prototypu je na nasledujúcej schéme.

Schéma sa dá rozdeliť na dve časti a to obvody v hlavnom module a súčiastky z modulov,



Obr. 3.9: náčrt obvodu prototypu

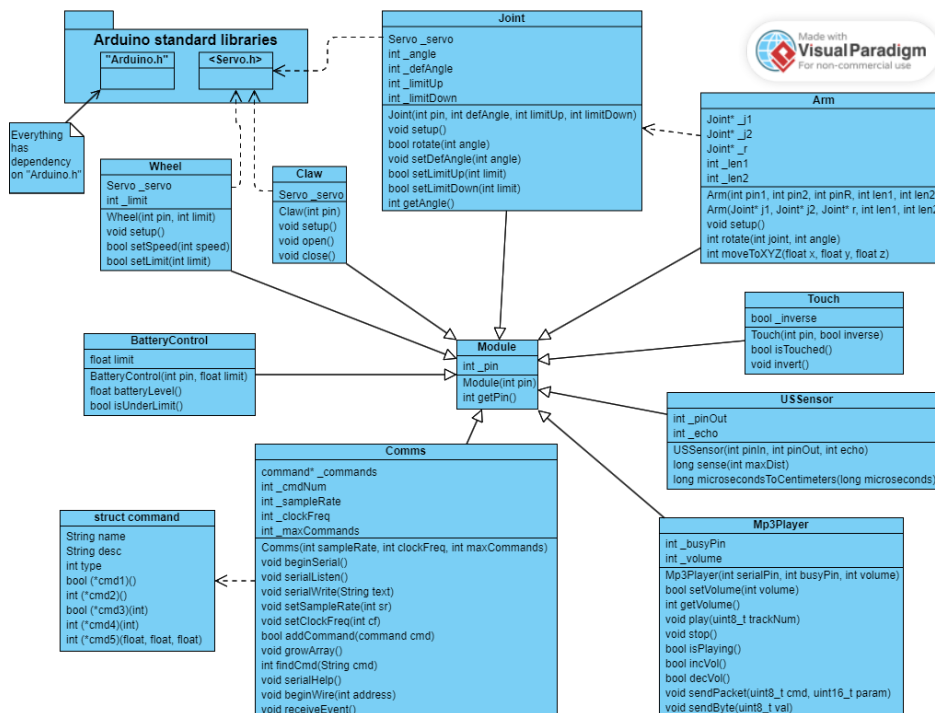
ktoré je možné pridať alebo odobrať podľa ľubovôle používateľov.

3.2 Softvér

Za cieľ sme si stanovili navrhnuť knižnicu, ktorá bude slúžiť ako jednoduchá platforma na ovládanie jednotlivých modulov. Každý modul okrem hlavného má vytvorenú triedu, ktorá je jeho virtuálnou dvojličkou. Pomocou metód v týchto triedach je možné jednoducho ovládať každý modul zvlášť.

Zoznam tried v knižnici:

1. trieda Joint pre modul Klb



Obr. 3.10: diagram knižnice Modbot

- Obsahuje príkazy na ovládanie otáčania a nastavenie limitov na otáčanie. Trieda kĺb pracuje s pozicionálnymi servami.
2. Trieda Wheel pre modul koleso
 - Obsahuje príkazy na ovládanie rýchlosti otáčania a nastavenie limitov na rýchlosť. Trieda koleso pracuje s kontinuálnymi servami.
 3. Trieda Claw pre modul kliešte
 - Obsahuje príkazy na otváranie a zatváranie kliešťov. Trieda pracuje s pozicionálnymi servami.
 4. Trieda Mp3player pre modul Mp3 prehrávač
 - Obsahuje príkazy na spustenie a zastavenie skladby a nastavenie hlasitosti.
 5. Trieda USSensor pre modul ultrazvukový senzor
 - Obsahuje príkazy na zistenie vzdialenosti predmetu.
 6. Trieda Touch pre tlačidlá a dotykové senzory
 - Obsahuje príkazy na zistenie stlačenia tlačidla alebo dotyku a možnosť softvérovo invertovať signál.
 7. Trieda Arm pre zoskupenie viacerých modulov Kĺb

- Obsahuje príkazy na posúvanie kolekcie modulov. Bolo naplánované vytvoriť metódu na posúvanie ramena pomocou inverznej kinematiky, ale metóda sa nestihla implementovať.
8. Trieda BatteryControl na kontrolu hladiny batérie
- Obsahuje metódu na zistenie hladiny batérie a vypnutie zariadenia.
9. Trieda Comms pre komunikáciu s počítačom alebo iným zariadením
- Obsahuje metódy na čítanie príkazov, posielanie správ a pridávanie nových príkazov. Príkazy sú uložené ako objekty, ktoré obsahujú názov príkazu a pointer na funkciu.

Triedy, ktoré ovládajú servo a triedu Comms je potrebné inicializovať po hardvérovej inicializácii, preto sú v nich pridané metódy setup. Tieto triedy vytvárajú spoločne virtuálny obraz robota a poskytujú používateľovi jednoduché príkazy na jeho ovládanie. Tým, že sú všetky triedy v separátnych súboroch, je možné pridať iba tie triedy, ktoré sú z knižnice potrebné.

Kapitola 4

Použitie v praxi

V nasledujúcej kapitole sa zaoberáme použitím Modbota v praxi. Opíšeme ako sa skladá a na konkrétnom príklade ukážeme a otestujeme programovanie robota v role používateľa.

4.1 Konštrukcia robota

Skladanie Modbota sa dá rozdeliť na dve časti:

1. skladanie modulov,
2. zostavenie robota.

Skladanie modulov je zložitejšia časť konštrukcie. Potrebná je skúsenosť so spájkovaním elektrických obvodov a spájaním dielov pomocou vrutov. Pred začiatkom sa treba uistiť, že máme všetky potrebné súčiastky kúpené, resp. vytlačené.

Pri prípadnom použití v kurzoch s mladšími žiakmi je potrebné túto časť vykonať dopredu, aby účastníci kurzu mali viac času na samotnú prácu s robotom. Pre starších žiakov a študentov to však môže byť výborná príležitosť získať nové zručnosti a naučiť sa hlbšie poznatky o konštrukcii a fungovaní robota.

Skladanie jednotlivých modulov prebieha nasledovne:

1. Hlavný modul
 - Pre hlavný modul je potrebné zospájať napájací obvod s Arduinoom podľa schémy (odkaz na schému). Obal je pospájaný pomocou 16 LEGO pinov a napájací obvod s Arduinoom je voľne umiestnený vo vnútri obalu. Spínač má byť vedený von z obalu.
2. Modul Kĺb a Koleso
 - Servá treba pripevniť pomocou dvoch vrutov do vytlačených obalov a pripojiť ostatné súčiastky k statickej časti modulu.
3. Modul Ultrazvukový senzor
 - Doska so senzorom sa nasunie do obalu.

4. Modul Mp3 prehrávač

- Prehrávač sa vloží do kolísky obalu pinmi nahor. Takto bude cez otvor zozadu prístupná SD karta.

5. Modul Reproduktor

- Pre tento modul treba ku kúpenému reproduktoru prispájať káble a vložiť reproduktor do dvojdielného obalu.

6. Modul Kliešte

- Servo je potrebné pripevniť dvoma vrutmi k obalu. Potom sa pripevnia obidve strany klieští, jedna strana k obalu pomocou jedného vrutu a druhá sa nasunie a zaistí na ramene serva.

Zostavenie robota z hotových modulov je už veľmi jednoduché. Moduly sa pospájajú pomocou LEGO pinov medzi sebou a káble sa vedú do vnútra hlavného modulu, kde sa pripoja na Arduino. Túto časť konštrukcie už bezpečne zvládnu aj mladší žiaci.

4.2 Nastavenie systému robota

Modbot sa programuje vo vývojovom prostredí Arduino. Programuje sa pomocou Arduino skriptu, ktorý je založený na C++. Pred tým, než sa s knižnicou bude dať pracovať, musí sa umiestniť do priečinka `C:\Users\<name\_of\_user>\AppData\Local\Arduino15\libraries`. Podľa toho, aké moduly sa v robotovi používajú, pridajú sa triedy z knižnice pomocou `#include <názov\_triedy.h>`. Pre moduly treba vytvoriť objekty z tried. Niektoré triedy je potrebné nastaviť po inicializovaní hardvéru, takže vo funkcii `setup()` je potrebné zavolať pre respektívne objekty metódu `setup()`. Toto sa týka tried pracujúcich so servami a triedy Comms. Po tomto začiatočnom nastavení je možné programovať robota podľa uváženia.

4.3 Testovanie prototypu v role používateľa

Na testovanie prototypu bol vytvorený krátky program, pomocou ktorého sme vyskúšali základné funkcionality Modbota. V tejto časti vysvetlíme, ako bol prototyp zostavený a naprogramovaný pomocou vytvorenej knižnice.

4.3.1 Zostava robota pre testovanie

Prototyp mal nasledovné moduly:

1. 1x hlavný modul
2. 2x kĺb
3. 1x koleso s pozicionálnym servom

4. 1x ultrazvukový senzor
5. 1x Mp3 prehrávač
6. 1x reproduktor
7. 1x kliešte

Okrem týchto modulov bol k prototypu pripojený dotykový kapacitný spínač. Na vrch hlavného modulu boli pripevnené moduly Mp3 prehrávač, Reproduktor a Koleso s pozicionálnym servom. Na Koleso boli pripevnené dva moduly Kĺb v sériovom zapojení vytvárajúce rameno s tromi bodmi otáčania - jedným na osi Z a dvoma na osi X. Takto vieme dostať posun koncového bodu ramena v trojrozmernom priestore. Na konci ramena sa nachádzal modul Kliešte. Na prednej strane Hlavného modulu bol Ultrazvukový senzor. V teste bola jedna bočná stena z hlavného modulu odstránená pre ľahší prístup k micro usb portu.

4.3.2 Testovací program

Ďalej uvádzame kód, ktorý by za bežných okolností vytvoril používateľ robota. Nám poslúžil na otestovanie funkčnosti a overenie cieľov, ktoré sme si stanovili pri úpravách dizajnu robota. Kód na ovládanie prototypu postupne rozoberieme a opíšeme jeho fungovanie.

- Na začiatku súboru sú priložené všetky triedy ktoré budem používať.

```
#include <Arm.h>
#include <BatteryControl.h>
#include <Claw.h>
#include <Comms.h>
#include <Joint.h>
#include <Module.h>
#include <Mp3Player.h>
#include <Touch.h>
#include <USSensor.h>
#include <Wheel.h>
```

- Za nimi nasledujú premenné, v ktorých sú uložené objekty jednotlivých tried používané na ovládanie modulov. Tiež sa tam nachádzajú tri premenné, ktoré budú obsahovať štruktúry pre príkazy.

```
Joint upperJoint = Joint(2);
Joint rotation = Joint(5);
Arm arm = Arm(&lowerJoint, &upperJoint, &rotation, 5,5);

Claw claw = Claw(4);
Touch touch = Touch(6,false);
USSensor distanceSensor = USSensor(7,8);
```

```

Mp3Player player = Mp3Player(9,10,8);
Comms com = Comms();
struct command cmd1;
struct command cmd2;
struct command cmd3;

```

- Po vytvorení týchto objektov sa vo funkcii *setup()* zapíšu príkazy do štruktúr a nastaví sa servá a komunikácia. Štruktúry s príkazmi sa vložia do objektu triedy *Comms*.

```

void setup() {
    cmd1.name = "w";
    cmd1.desc = "picks up a thing";
    cmd1.cmd1 = pickUp;
    cmd1.type = 1;
    cmd2.name = "s";
    cmd2.desc = "puts down a thing";
    cmd2.cmd1 = putDown;
    cmd2.type = 1;
    cmd3.name = "wave";
    cmd3.desc = "arm waves at a person, after command you write a number how many times";
    cmd3.cmd3 = wave;
    cmd3.type = 3;

    lowerJoint.setup();
    upperJoint.setup();
    rotation.setup();
    claw.setup();
    com.beginSerial();
    com.addCommand(cmd1);
    com.addCommand(cmd2);
    com.addCommand(cmd3);
}

```

- Vo funkcii *loop()* sa vyskytuje kód, ktorý sa vykonáva cyklicky. Na začiatku sa zavolá metóda *com.serialListen()*, ktorá načúva, či neprichádzajú informácie z počítača. Ak nič nie je v bufferi Serial portu, tak skončí.

```

void loop() {
    com.serialListen();
}

```

Nasleduje podmienka `distanceSensor.sense()<5` ktorá sa spustí ak sa niečo bude nachádzať menej ako 5cm od senzoru. Ak je podmienka pravdivá, tak sa spustia po sebe s rozstupom 1 sekundy dve funkcie, ktoré hýbu s ramenom.

```

if (distanceSensor.sense() < 5){
    pickUp();
    delay(1000);
    putDown();
    delay(1000);
}

```

Podmienka `touch.isTouched()` je pravdivá iba vtedy, keď je zaznamenaný dotyk na dotykovom spínači. Ak sa niekto dotkne spínača, zistí sa, či sa práve prehráva nejaká hudba `player.isPlaying()`. Ak áno, tak sa hudba zastaví `player.stop()`, inak sa nastaví hlasitosť prehrávania na level 8 z 30 `player.setVolume(8)` a spustí sa skladba číslo 1 z SD karty `player.play(1)`.

```

if (touch.isTouched()){
    if (player.isPlaying()){
        delay(500);
        player.stop();
    } else {
        delay(500);
        player.setVolume(8);
        player.play(1);
    }
}
}

```

- Funkcia `pickUp()` volá sled metód, ktoré hýbu jednotlivými časťami ramena. Po zavolaní tejto funkcie sa rameno zohne a pripraví sa na uchopenie veci. Po dvoch sekundách sa kliešte zavrú a rameno sa zdvihne do začiatkovej pozície.

```

bool pickUp(){
    com.serialWrite("pick up");
    rotation.rotate(45);
    delay(300);
    lowerJoint.rotate(30);
    delay(300);
    upperJoint.rotate(20);
    delay(300);
    claw.open();
    delay(2000);
    claw.close();
    delay(300);
    lowerJoint.rotate(90);
    delay(300);
}

```

```

    upperJoint.rotate(90);
    delay(300);
    rotation.rotate(90);
    return true;
}

```

- Funkcia `putDown()` otočí a zohne rameno a otvorí kliešte. Následne sa rameno trochu zdvihne, kliešte sa zavrujú a rameno sa vráti do počiatočnej pozície.

```

bool putDown(){
    com.serialWrite("put down");
    rotation.rotate(125);
    delay(300);
    lowerJoint.rotate(30);
    delay(300);
    upperJoint.rotate(30);
    delay(300);
    claw.open();
    delay(300);
    lowerJoint.rotate(90);
    delay(300);
    claw.close();
    delay(300);
    upperJoint.rotate(90);
    delay(300);
    rotation.rotate(90);
    return true;
}

```

- Posledná funkcia `wave` dostáva číslo, ktoré hovorí koľkokrát sa má opakovať cyklus mávania. Po dokončení cyklu sa rameno zastaví a dvakrát za sebou robot cvakne kliešťami.

```

bool wave(int x){
    rotation.rotate(150);
    for (int i = 0; i < x; i++){
        lowerJoint.rotate(70);
        delay(50);
        upperJoint.rotate(70);
        delay(300);
        lowerJoint.rotate(90);
        delay(50);
        lowerJoint.rotate(110);
    }
}

```

```
        delay (30);
        upperJoint.rotate (90);
        delay (50);
        upperJoint.rotate (110);
        delay (300);
        lowerJoint.rotate (90);
        delay (50);
        upperJoint.rotate (90);
        delay (50);
    }
    rotation.rotate (45);
    delay (500);
    claw.open ();
    delay (200);
    claw.close ();
    delay (200);
    claw.open ();
    delay (200);
    claw.close ();
    delay (500);
    rotation.rotate (90);
    return true;
}
```

Video z priebehu testu uvádzame v prílohe.

4.4 Závery z testovania

Zostavenie robota prebehlo bez komplikácií. Diely vytvorené pomocou 3D tlače sa ukázali ako vyhovujúce. Po spustení programu Modbot vykonal požadované operácie podľa očakávania. Potvrdilo sa, že skladanie modulov a úvodné spustenie robota si vyžadujú vyššiu zručnosť a je vhodné (aspoň pri mladších žiakoch), aby ich realizoval zaškolený personál pred začiatkom práce s robotom.

Záver

V práci sme sa venovali vývoju robotickej stavebnice Modbot na edukačné účely, ktorá nájde svoje využitie v kurzoch dielne Fablab pre študentov stredných škôl a dospelých. Nadviazali sme na starší projekt MoKraRosa z roku 2019. Robot MoKraRosA bol sofistikovaný a preto bol aj viac komplikovaný a náročnejší na použitie v kurzoch.

Cieľom nášho projektu bolo v danom časovom rámci zjednodušiť a vylepšiť tohto robota, aby sa dal pravidelne jednoducho používať v reálnych aktivitách vzdelávacích kurzov. To sme dosiahli sústredením sa na dve oblasti. Z hľadiska hardvéru sa podarilo znížiť hmotnosť modulov a vytvoriť lepší systém ich skladania, ktorý umožní používateľom ľahšiu manipuláciu a využitie Modbota. V softvérovej oblasti sa vytvorila knižnica s jednoduchými príkazmi pre každý modul. Tá umožňuje ovládanie robota aj pre začínajúcich používateľov. Veľkým prínosom je modulárnosť Modbota a otvorenosť jeho ekosystému vhodná pre ďalší vývoj.

V budúcnosti bude potrebné otestovať stavebnicu s účastníkmi kurzov a zhodnotiť možnosti ďalšieho vývoja z hľadiska pedagogickej metodológie a vyučovacích potrieb. Už v súčasnosti vieme o niektorých drobných nedostatkoch v modeloch dielov a vo funkcionalite kódu, ktoré budeme postupne odstraňovať a vylepšovať.

Celkovo je robotická stavebnica Modbot funkčná, použiteľná vo vzdelávaní a má potenciál na ďalší vývoj. Veríme, že bude prínosom a inšpiruje ďalších ľudí k štúdiu a objavovaniu nových možností, ktoré ponúka informatika a robotika.

Literatúra

- [1] The Fab Charter angl., 2012. [Citované 2023-02-12], Dostupné z <http://fab.cba.mit.edu/about/charter/>.
- [2] how otto was born? and why is called otto?, 2017. [Citované 2023-06-01], Dostupné z <https://www.ottodiy.com/blog/story>.
- [3] Projekt Mo(dulárny) Kra(hulčí) Ro(bot) s A(rduinom) bol vytvorený na Letnom sústreďení talentovanej mládeže v Krahulciach v júli 2019, 2019. [Citované 2023-02-12], Dostupné z <https://github.com/Robotics-DAI-FMFI-UK/mokrarosa/wiki>.
- [4] Stringcomp, 2019. [Citované 2023-06-01], Dostupné z <https://en.wikipedia.org/wiki/STRINGCOMP>.
- [5] Joss, 2022. [Citované 2023-06-01], Dostupné z <https://en.wikipedia.org/wiki/JOSS>.
- [6] Telcomp, 2022. [Citované 2023-06-01], Dostupné z <https://en.wikipedia.org/wiki/TELCOMP>.
- [7] 3d printing, 2023. [Citované 2023-06-01], Dostupné z https://en.wikipedia.org/wiki/3D_printing.
- [8] About, 2023. [Citované 2023-06-01], Dostupné z <https://microbit.org/about/#top>.
- [9] About make it digital, 2023. [Citované 2023-06-01], Dostupné z <https://www.bbc.co.uk/programmes/articles/1gkww58DPmRzt2TzDp3pr9x/about-make-it-digital>.
- [10] Arduino, 2023. [Citované 2023-06-01], Dostupné z <https://en.wikipedia.org/wiki/Arduino>.
- [11] I2c, 2023. [Citované 2023-06-01], Dostupné z <https://en.wikipedia.org/wiki/I%C2%B2C>.
- [12] Kitronik :move mini mk2 buggy kit, 2023. [Citované 2023-06-01], Dostupné z <https://kitronik.co.uk/products/5652-move-mini-mk2-buggy-kit-excl-microbit>.
- [13] Lego mindstorms, 2023. [Citované 2023-06-01], Dostupné z https://en.wikipedia.org/wiki/Lego_Mindstorms.
- [14] Logo history, 2023. [Citované 2023-06-01], Dostupné z https://el.media.mit.edu/logo-foundation/what_is_logo/history.html.

- [15] Logo (programming language), 2023. [Citované 2023-06-01], Dostupné z [https://en.wikipedia.org/wiki/Logo_\(programming_language\)](https://en.wikipedia.org/wiki/Logo_(programming_language)).
- [16] Makeblock, 2023. [Citované 2023-06-01], Dostupné z <https://en.wikipedia.org/wiki/Makeblock>.
- [17] mblock, 2023. [Citované 2023-06-01], Dostupné z <https://mblock.makeblock.com/en/>.
- [18] mbot2, 2023. [Citované 2023-06-01], Dostupné z <https://education.makeblock.com/mbot2/>.
- [19] Micro bit, 2023. [Citované 2023-06-01], Dostupné z https://en.wikipedia.org/wiki/Micro_Bit.
- [20] Milestones for the bbc micro:bit, 2023. [Citované 2023-06-01], Dostupné z <https://microbit.org/impact/case-studies/milestones-for-the-bbc-microbit/>.
- [21] O fablabe, 2023. [Citované 2023-06-01], Dostupné z <https://www.fablab.sk/o-fablabe/>.
- [22] O nás, 2023. [Citované 2023-06-01], Dostupné z https://www.prusa3d.com/cs/stranka/o-nas_77/.
- [23] Ozobot, 2023. [Citované 2023-06-01], Dostupné z <https://ozobot.com/>.
- [24] Servomotor, 2023. [Citované 2023-06-01], Dostupné z <https://en.wikipedia.org/wiki/Servomotor>.
- [25] Joseph Flynt. A detailed history of 3d printing, 2023. [Citované 2023-06-01], Dostupné z <https://3dinsider.com/3d-printing-history/>.
- [26] Kris Hauser. Inverse kinematics, 2020. [Citované 2023-06-01], Dostupné z <https://motion.cs.illinois.edu/RoboticSystems/InverseKinematics.html>.
- [27] Kris Hauser. Robot kinematics, 2020. [Citované 2023-06-01], Dostupné z <http://motion.cs.illinois.edu/RoboticSystems/Kinematics.html>.
- [28] Marco. Bob the biped robot, 2019. [Citované 2023-06-01], Dostupné z <https://www.personalrobots.biz/bob-the-biped-robot/>.
- [29] Fred Martin, Bakhtiar Mikhak, Mitchel Resnick, Brian Silverman, and Robbie Berg. To mindstorms and beyond: Evolution of a construction kit for magical machines, 2000. [Citované 2023-06-01], Dostupné z https://www.researchgate.net/publication/242467788_To_Mindstorms_and_Beyond_Evolution_of_a_Construction_Kit_for_Magical_Machines.
- [30] Seymour Papert. *Mindstorms: Children, Computers And Powerful Ideas*. Basic Books, Inc., 1980.

- [31] roamerrobot@tumblr. The history of turtle robots, 2012. [Citované 2023-06-01], Dostupné z <https://roamerrobot.tumblr.com/post/23079345849/the-history-of-turtle-robots>.
- [32] Karl Söderby. Getting started with arduino, 2023. [Citované 2023-06-01], Dostupné z <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino>.
- [33] Michael Vallance, Stewart Martin, Charles Wiz, and Paul van Schaik. LEGO mindstormstm for informed metrics in virtual worlds. In *Proceedings of the 2009 British Computer Society Conference on Human-Computer Interaction, BCS-HCI 2009, Cambridge, United Kingdom, 1-5 September 2009*, pages 159–162. ACM, 2009.
- [34] Chris Wharfe. Lego is discontinuing mindstorms in 2022, 2022. [Citované 2023-06-01], Dostupné z <https://www.brickfanatics.com/lego-discontinuing-mindstorms-end-of-2022>.

Príloha A: obsah elektronickej prílohy

V elektronickej prílohe priloženej k práci sa nachádza zdrojový kód, testovací program a 3D modely súčiastok. Všetko je zverejnené aj na stránke <https://github.com/generic1707/ModBot/>.

Video z testu robota je možné nájsť na odkaze: <https://youtu.be/XplRpoystww>