SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-100863-111124

VYTVORENIE OVLÁDAČA V PROSTREDÍ ROS PRE MOBILNÉHO ROBOTA

BAKALÁRSKA PRÁCA

2023 **Filip Lobpreis**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-100863-111124

VYTVORENIE OVLÁDAČA V PROSTREDÍ ROS PRE MOBILNÉHO ROBOTA

BAKALÁRSKA PRÁCA

Študijný program:	Robotika a kybernetika
Názov študijného odboru:	kybernetika
Školiace pracovisko:	Ústav robotiky a kybernetiky
Vedúci záverečnej práce:	Ing. Michal Dobiš
Konzultant:	Ing. Michal Dobiš

Bratislava 2023 **Filip Lobpreis**

Slovenská technická univerzita v Bratislave
Ústav robotiky a kybernetiky

Fakulta elektrotechniky a informatiky
2022/2023

ZADANIE BAKALÁRSKEJ PRÁCE

Autor práce:	Filip Lobpreis
Študijný program:	robotika a kybernetika
Študijný odbor:	kybernetika
Evidenčné číslo:	FEI-100863-111124
ID študenta:	111124
Vedúci práce:	Ing. Michal Dobiš
Vedúci pracoviska:	prof. Ing. Jarmila Pavlovičová, PhD.
Miesto vypracovania:	Ústav robotiky a kybernetiky
Názov práce:	**Vytvorenie ovládača v prostredí ROS pre mobilného robota**
Jazyk, v ktorom sa práca vypracuje:	slovenský jazyk
Špecifikácia zadania:	Mobilná robotika využívaná v kombinácií s logistickými alebo servisnými úkonmi sa stáva čoraz viac populárnejšou. Úlohou študenta je naštudovať si mobilné robotické zariadenie, ktoré bude mať k dispozícií na Národnom centre robotiky a k nemu príslušné materiály. Študent bude pracovať s reálnym hardvérom a otvoreným systémom, ktorý bude potrebné preštudovať a pochopiť jeho fungovanie. Cieľom práce bude následne vytvoriť nadradený ovládač implementovaný v ROS (Robotickom operačnom systéme), ktorý bude schopný riadiť daného robota. Úlohy: 1. Analyzujte súčasný stav riešenia a prostredie Robotického operačného systému. 2. Analyzujte možnosti a metodiku implementácia riadiaceho balíka pre daný robot 3. Navrhnite spôsob implementácie a architektúru riešenia 4. Implementujte riadiaci systém pre mobilného roba 5. Vypracujte dokumentáciu k dosiahnutým výsledkom. 6. Vyhodnoťte dosiahnuté výsledky.
Termín odovzdania práce:	02. 06. 2023
Dátum schválenia zadania práce:	
Zadanie práce schválil:	

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Robotika a kybernetika
Autor:	Filip Lobpreis
Bakalárska práca:	Vytvorenie Ovládača v prostredí ROS pre mobilného robota
Vedúci záverečnej práce:	Ing. Michal Dobiš
Konzultant:	Ing. Michal Dobiš
Miesto a rok predloženia práce:	Bratislava 2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean et est a dui semper facilisis. Pellentesque placerat elit a nunc. Nullam tortor odio, rutrum quis, egestas ut, posuere sed, felis. Vestibulum placerat feugiat nisl. Suspendisse lacinia, odio non feugiat vestibulum, sem erat blandit metus, ac nonummy magna odio pharetra felis. Vivamus vehicula velit non metus faucibus auctor. Nam sed augue. Donec orci. Cras eget diam et dolor dapibus sollicitudin. In lacinia, tellus vitae laoreet ultrices, lectus ligula dictum dui, eget condimentum velit dui vitae ante. Nulla nonummy augue nec pede. Pellentesque ut nulla. Donec at libero. Pellentesque at nisl ac nisi fermentum viverra. Praesent odio. Phasellus tincidunt diam ut ipsum. Donec eget est. A skúška mäkčeňov a dĺžnov.

Kľúčové slová: kľúčové slovo1, kľúčové slovo2, kľúčové slovo3

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Robotics and cybernetics
Author:	Filip Lobpreis
Bachelor's thesis:	Implementation of ROS Driver for Mobile Robot
Supervisor:	Ing. Michal Dobiš
Consultant:	Ing. Michal Dobiš
Place and year of submission:	Bratislava 2023

On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures, or else he endures pains to avoid worse pains.

Keywords: ROS, BlackMetal

Pod'akovanie

I would like to express a gratitude to my thesis supervisor.

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek

DDS Data Distribution Service

IPC Inter Process Communication

LAN Local Area Network

ROS Robot Operating System

Zoznam algoritmov

Zoznam výpisov

Úvod

Ukazka upraveneho templatu FEIStyle.cls (`https://github.com/Kyslik/FEIStyle`) s pouzitim Times New Roman fontu. Nastavenie fontov som prebral z oficialneho IEEE templatu, vlozil som ho do `FEIStyle.cls` na riadky 228-230.

Tu bude krasny uvod s diakritikou atd.
A mozno aj viac riadkovy uvod.

1 ROS

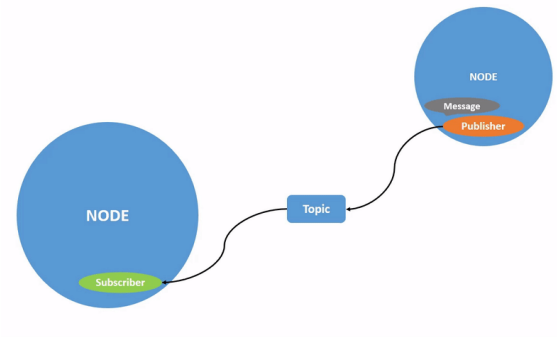
Robotický operačný systém (Robot Operating System) je súbor voľne dostupných softvérových knižníc a nástrojov, ktoré nám vytvárajú vhodné podmienky na písanie aplikácií pre mnohé druhy robotov. ROS má dve verzie. Vo všeobecnosti sa stretneme s tým, že pod názvom ROS1 alebo ROS sa myslí ROS verzie 1. Pod názvom ROS2 sa myslí ROS verzie 2. Aby nenastali nejasnosti budeme v tomto dokumente označovať ROS verzie 1 ako ROS1 a ROS verzie 2 ako ROS2 V prípade, keď budme hovoriť o spoločných vlastnostiach a funkcionalitách, ROS1 a ROS2 budeme označovať dokopy ako ROS.

1.1 ROS

Komunikácia v ROSe je zabezpečená cez IPC (Inter Process Communication), TCP/IP UDP/IP komunikáciou pomocou troch zakladacích metód: **topiky** (Topics), **služby** (Service) a **akcie** (Actions)

1.1.1 Topiky

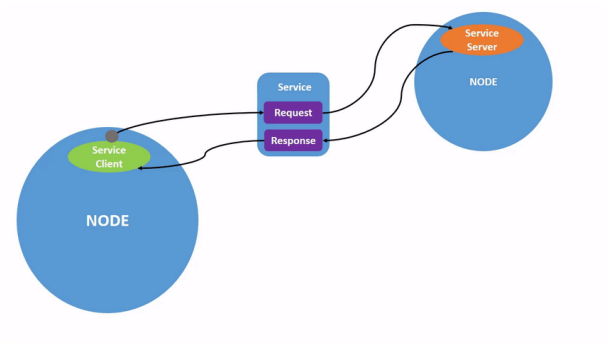
Topiky (IPC) sú najjednoduchší spôsob komunikácie. Vieme si ich prirovnať k UDP/IP protokolu. Definujeme si jedného poskytovateľa (publisher) a jedného príjemcu (subscriber). Medzi týmito dvoma účastníkmi sa následne posielajú správy (messages), ktoré sme si dopredu definovali.



Obr. 1.1: Vizualizacia topicu v ROSe [1]

1.1.2 Služby

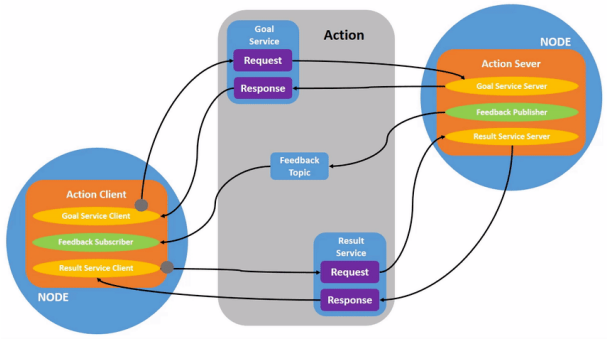
Služby (TCP/IP) nám poskytujú rovnaký spôsob komunikácie ako topiky, až na to, že sa správy medzi servisom a klientom posielajú cez LAN (Local Area Network). Služby sa využívajú pri komunikácii medzi viacerými zariadeniami.



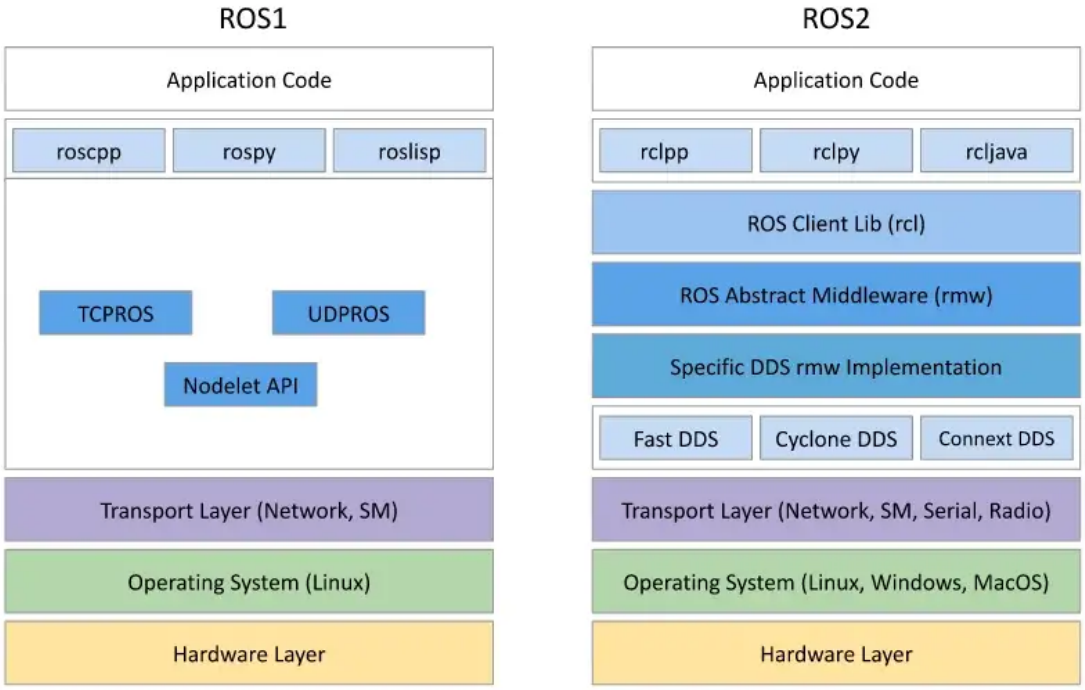
Obr. 1.2: Vizualizácia služby v ROSe [1]

1.1.3 Akcie

Akcie (TCP/IP) sú najzložitejším spôsobom komunikácie. Tento spôsob bol pridaný do ROS1 až neskor. V druhej verzii ROSu je tento typ komunikácie medzi troma základnými. Sú založené na službách a máju 3 stavy. Najprv pošle klient serveru akú akciu má vykonať, server mu potvrdí, že túto požiadavku dostal. Server začne následne vykonávať danú akciu a posielať klientovi priebežné správy o priebehu. Keď server skončí pošle klientovi výsledok akcie a klient mu obratom potvrdí obdržanie výsledku.



Obr. 1.3: Vizualizácia akcie v ROSe [1]



Obr. 1.4: Porovnanie štruktúr ROS1 a ROS2 [2]

1.2 ROS1

ROS bol prvýkrát vydaný v roku 2007. Ide o softvér, ktorý sa začal vyvíjať so zámerom zjednodušiť programovanie a ovládanie robotov. Od doby, kedy vznikol prešiel mnohými verziami a úpravami. Jeho neoddeliteľnou súčasťou sú štrukturalizovanie programu do uzlov (nodov), komunikácia medzi uzlami, podpora viacerých programovacích jazykov ako sú C, C++ alebo Python a vytváranie balíčkov dostupných širokej verejnosti.

Štrukturalizovanie základov ROS1 je spravené monoliticky čo najstabilnejším spôsobom. Na počiatku programu musíme spustiť hlavný program (roscore), ktorý zabezpečuje vytváranie jednotlivých uzlov. Komunikácia medzi uzlami je zabezpečená prostredníctvom prepojenia uzlov cez LAN/WLAN alebo IPC komunikaciu. Uzly môžu byť spustené aj na iných zariadeniach. Roscore ďalej poskytuje parametre jednotlivým uzlom z parametrového servera. Jeho najväčšou úlohou je zabezpečenie komunikácie uzlov v programe. Nezáleží na tom, či sú uzly na jednom alebo viacerých zariadeniach.

Aj napriek mnohým výhodám má ROS aj nedostatky, ktoré sa ťahajú už od jeho počiatkov. Sú to napríklad:

- Nepostačujúca distribuovanosť systému. Ak prestane fungovať roscore, prestane ísť celý program,
- ROS1 je písaný v starom štandarde a to vnáša do programu technologický dlh a bezpečnostné riziká,
- Kvalita komunikácie sa nedá ovplyvniť

Kvôli takýmto problémom sa začala vyvíjať nová verzia ROSu, ROS2. V roku 2025 sa skončí podpora poslednej distribúcie ROS1 menom Noetic. Preto je odporúčané začínať nové projekty v ROS2.

1.3 ROS2

Ako už bolo spomenuté zámerom vývoja ROS2 je zlepšenie funkcionality a bezpečnosti systému. ROS2 nie je spätne kompatibilný. Podstata toho, ako sú zoskupované uzly a ako spolu komunikujú je diametrálne odlišná od ROS1. Z tohto dôvodu bol vyvinutý takzvaný rosbridge, ktorý zabezpečuje kompatibilitu medzi verziami. Nie je to ale trvalé riešenie. Odporúčané je nástroj využívať a počas toho prepisovať kód z verzie 1 do verzie 2. Komunikácia prebieha v ROS2 rovnakým spôsobom ako v ROS1. Pomocou topikov, služieb a akcií.

Táto podobnosť končí na najvyššej vrstve. Ako sme videli na Obr. 1.4. Štruktúra ROS2 je rozdelená do viacerých vrstiev. Najdôležitejšie je pre nás vedieť, že komunikácia je spracovávaná modelom DDS (Služba distribúcie údajov) z anglického (Data Distribution Service). Tento model zlepšuje výkon, stabilitu a bezpečnosť modelu oproti ROS1. Je založený na TCP/UDP protokole. Z obrázku vyčítame aj lepšie rozloženie modulov. To zabezpečuje jednoduchšie prispôsobovanie systému pre nové funkcionality. Podpora operačných systémov sa v ROS2 rozšírila aj o Windows, Mac OS či operačné systémy reálneho času. Operačne systémy nie sú jediné rozšírenie ohľadom kompatibility. S ROS2 je možné programovať už aj v Jave či Matlabe. Tvorcovia mysleli aj na programátorov a pridali rozšírené možnosti testovania, debugovania či nasadzovania programu do reálneho využitia.

ROS2 ma necentralizovanú štruktúru, a preto pri spúšťaní programov už nie je potrebné mať spustený roscore. Ak teda spadne jeden proces druhé budú fungovať naďalej. V ROS1 sme vedeli ovplyvniť počet uchovaných správ pokým nepretiekol zásobník, ktorý ich uchovával na neskoršie použitie. V ROS2 vieme zmeniť kvalitu komunikácie. Vieme si zadefinovať, či by sme radšej stratili niektoré správy, ale dostali by sme všetky rýchlo. Alebo aby sa zabezpečilo,

že dostaneme všetky správy, ktoré boli vyslané, aj keby to trvalo dlhšie. Dokonca si vieme zadefinovať maximálny čas, ktorý budeme čakať na ďalšiu správu.

Pri všetkých týchto zlepšeniach nemôžeme zabudnúť aj nasledovný nedostatok. Keďže ROS2 je mladší ako ROS1 nájdeme k nemu menej dokumentácie. Pridaním veľkého počtu funkcionalít začal vznikať problém pre začiatočníkov s porozumením niektorých kódov. Avšak tento problém je nedostatkom, ktorý časom zanikne. V čase písania tejto prace pribudli na stránke dokumentácie 2 strany popisujúce pokročilejšie Funkcionality druhej verzie ROSu.

1.4 Rozdiely

Čo je určite dobrou správou pre všetkých programátorov, ktorí robili v prvej verzii a sú zvyknutí na jej štandardy a funkcionalitu sa nemajú čoho obávať. Prechod z ROS1 na ROS2 je dosť priamočiary. Čo sa zmenilo je spôsob písania kódu, ale koncepty funkcionality ostali všetky rovnaké. V tejto sekcii nebudeme písať konkrétne kódy, budeme len opisovať čo je podobné a čo zasa rozdielne medzi verziami spomínaného systému. Keďže celý projekt bol písaný v programovacom jazyku C++ tak sa aj tieto zmeny budu týkať hlavne C++.

1.4.1 Štandard jazyka

Pokým ROS1 bola písaná v štandarde C++98 tak ROS2 je už písaná v novom štandarde. To zahŕňa inicializovanie templatov a ich používanie. Definície a deklarácie templatov sú na knihu samú o sebe, preto do detailov nebudeme zachádzať. Stačí nám vedieť, ako ich inicializovať. V prvej verzii sme definovali publishera (publikovateľa) všeobecného a definovali sme mu cez aký topic má posielať správy. V druhej verzii naväzujeme publishera na špecifický tip správy akú posielame. Nemôže sa teda stať, že takýto program by sme skompilovali a následne keď ho spustíme tak by spadol.

1.4.2 Inicializácia nody

Tak isto ako v prvej verzii aj v druhej verzii musíme definovať nodu. Rozdiel je v tom, že prvá verzia obsahovala NodeHandle a druha verzia obsahuje priamo Node. V druhej verzii je zaužívaným štandardom tuto nodu precediť a použiť polymorfizmus pri objekte, ktorý musí existovať počas celej doby vykonávania programu. Pri prvej verzii tomu tak nebolo. Museli sme vytvoriť už spomenutý NodeHandle. Ten sa nemusel využiť ako base trieda a nemusel ani existovať počas celého behu programu.

1.4.3 Komunikácia

DDS (Služba distribúcie údajov) je stredný protokol používaný v ROS2 na komunikáciu medzi uzlami. Je to systém správ publikovania (publish) / odoberania (subscribe), ktorý umožňuje uzlom komunikovať medzi sebou bez toho, aby poznali identitu ostatných uzlov. DDS je

štandardný protokol, ktorý sa okrem ROS2 používa vo viacerých odvetviach. [3] Druh komunikácie je v ROS2 rozšírený ešte o akcie 1.1.3.

1.4.4 Vlákna

Prvá verzia ROSu je založená na jedno vláknovom, synchrónnom komunikačnom modeli, zatiaľ čo ROS2 je založený na viac vláknovom, asynchrónnom komunikačnom modeli. Táto zmena znateľne zrýchľuje program, ale prináša väčšiu komplexnosť do programu.

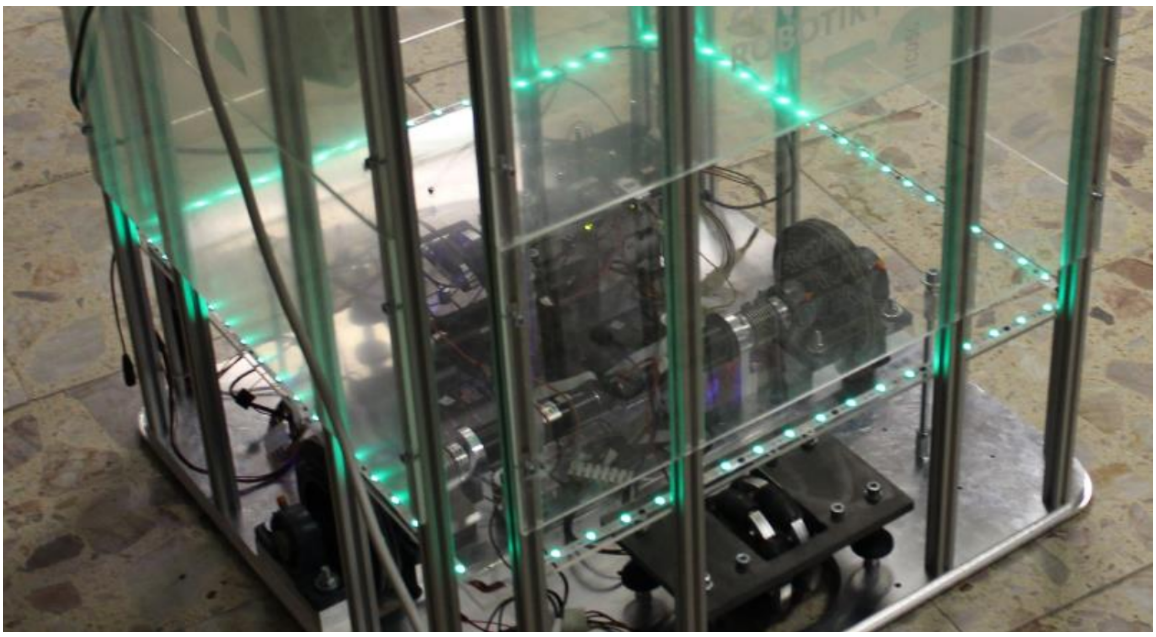
1.4.5 Kompilácia

Zmenou verzii sa zmenil aj spôsob kompilácie programu. ROS1 bol skompilovaný pomocou `catkin` build systému. Catkin je založený na programe `cmake`. Jeho nastavenie dependencií je konfigurované pomocou súboru `package.xml` . ROS2 prešiel na viac nastaviteľný systém `Colcon`. Tento systém je na rozdiel od catkinu založený na Pythone a jeho dependencie sa nastavujú pomocou `setup.py` súboru. V prípade colconu si môžeme definovať spôsob kompilácie t.j. môžeme nastaviť, ako sa budú spracovávať dependencie. Jednou s najviac používaných možností je `ament_cmake`. Je založený na cmaku a spolupracuje s colconom. Medze ament_cmake je založený na cmaku vieme mu definovať dependencie pomocou xml súboru ako v ROS1. Je to jeden zo spôsobov, ako zmenšiť rozdiel medzi ROS1 a ROS2.

2 Robot

Robot, s ktorým sme pracovali bol výsledkom tímového projektu viacerých študentov z roku 2019. Pri vysvetľovaní a opisovaní robota sa budeme odvolávať na dokumenty, stránky a kód, ktorý napísali. Všetky tieto údaje si sprístupnené na mobilnom robote v záložke `$(HOME)/Desktop/Blackmetal` [4].

Robot je v tvare kvádra. Jeho šírka je 60cm a je vyzdvihnutý nad zem o 1.5cm. Nachádza sa na kolesách o polomere 8cm. Jeho kostra, až na oceľové pláty, ktoré držia robot, je spravená z hliníku. Konkrétne z hliníkových tyčí, ktoré sú pospájané plexisklovými plátmi. Jeho podobizeň vidíme na nasledujúcom obrázku.



Obr. 2.1: Zobrazenie spodnej časti mobilného robota [4]

Na obrázku ďalej vidíme olemovanie robota pasom s LEDkami. Tie svietia nasledovným spôsobom. Keď sa robot nehýbe všetky LEDky svietia na zeleno. Keď sa robot pohne do nejakej strany, LEDky znázornia jeho pohyb tým, že svietia na strane, do ktorej sa robot hýbe. Keď nastane situácia, kedy počítač ovládajúci motory prestane komunikovať s Arduinom, ktoré sa stará o detekciu stavov robota tak LEDky začnú blikať červeno-modrými farbami.

Ako bolo spomenuté LEDky znázorňujú pohyb robota. Ten sa pohybuje za pomoci diferenciálneho podvozku s dvoma podpornými všesmerovými kolesami. Motory robota sú pripojené na meniče. Tie sú ovládané priamo príkazmi z počítača.

2.1 Komunikácia s robotom

S robotom sa vieme spojiť pomocou dvoch portov. Jeden port je otvorený na prijímanie požiadavok (requestov) a ten druhy je na monitorovanie stavu robota. Port *664* je otvorený pre tisíc užívateľov, ktorí môžu len sledovať stav robota. Druhy port je na prijímanie requestov *665* a je otvorený len pre jedného užívateľa.

2.1.1 Logovanie

Spomínaný port *664* je otvorený jednému užívateľovi. Keď sa užívateľ pripojí začne dostavať nepretržite správy typu JSON, ktoré hlásia stav robota. Spravy, ktoré dostávame sú nasledujúceho formátu

```
{"state":1,"direction":1}
```

Hodnoty sa pri stave (state) a ani pri smere (direction) nemenia. Sú to stále jednotky. Pokým robota nezastavíme buď príkazom, alebo stlačením tlačidla vypnutia, tak sa tieto správy budu posielať. Môžeme potom začať polemizovať o tom či by nebolo lepšie už tieto spravy využiť na to čo reálne spomenutý JSON reťazec ukazuje.

2.1.2 Ovládanie

Port *665* je sprístupnený na prijímanie a odosielanie požiadavok a ich odpovedí. Príkazy sa na počítač posielajú cez sieť z externého počítača vo formáte **JSON**. Študenti, ktorí navrhovali systém posielania požiadavok (request) a odpovedí (response) robili tieto správy ručne. Preto nastávajú situácie, kedy robot pošle správu, ktorá nespadá do štandardu písania JSON textu. Z tohto dôvodu sme nemohli použiť už existujúci kód, ktorý by nám zjednodušil prehľadávanie týchto sprav. Podla dokumentácie sa robot mal ovládať správami typu [5]

```
{"UserID":1,"Command":3,"RightWheelSpeed":50,"LeftWheelSpeed":50}
```

Význam jednotlivých parametrov:

- **UserID** - Znázorňuje ID užívateľa, ktorý je pripojený na robot. Predvolená hodnota je 1.
- **Command** - Číselná hodnota znázorňujúca príkaz, ktorý ma robot vykonať

    0. Prázdny príkaz slúžiaci na overenie spojenia
    1. Núdzové zastavenie
    2. Normálne zastavenie
    3. Príkaz nastavujúc rýchlosti kolies mobilného robota
    4. Prázdny príkaz
    5. Prázdny príkaz

6. Príkaz pýtajúci si aktuálnu rýchlosti pravého a ľavého kolesa. Tento príkaz nebol sprave navrhnutý v kóde robota. Vracal nám žiadaná hodnotu namiesto aktuálnej. Museli sme ho prepísať
7. Pripravenie motorov robota
8. Príkaz pýtajúci si aktuálnu pozíciu pravého a ľavého kolesa.

- **RightWheelSpeed** - Nastavenie rýchlosti pre pravé koleso
- **LeftWheelSpeed** - Nastavenie rýchlosti pre ľavé koleso

Z tohto kusu kódu je jasné, že sa majú posielať celé čísla a na základe tohto vstupu sa bude robot hýbať. Čo sme zistili až po skompilovaní a spustení tímového projektu je, že sa majú posielať desatinné čísla z intervalu 0 až 1. Po spomenutí tejto skutočnosti môžeme uviesť vyznám jednotlivých parametrov. Toto bolo písané v dokumentácii, ktorá nám bola dodaná na začiatku programu. Môžeme preto príklad prepísať na reťazec, ktorý by fungoval

```
{"UserID":1,"Command":3,"RightWheelSpeed":0.50,"LeftWheelSpeed":0.50}
```

2.2 Par slov k parametrom reťazca

UserID

Táto možnosť je v momentálnom stave robota nevyužitá. Počet zariadení, ktoré sa môžu pripojiť na port, cez ktorý sa dá robot ovládať je 1. Je to ale dobrá možnosť na rozšírenie kódu. Keď sa budú môcť pripojiť viacerí užívatelia, tak sa bude musieť vyriešiť, koho príkaz bude mať akú prioritu.

Command:4

Tento príkaz je prázdny. My sme ho ale neskôr prepísali na príkaz, cez ktorý sa dá nastaviť žiadaná pozícia kolies robota (natočenie). Táto funkcionalita nie je v takom stave ako sme si priali.

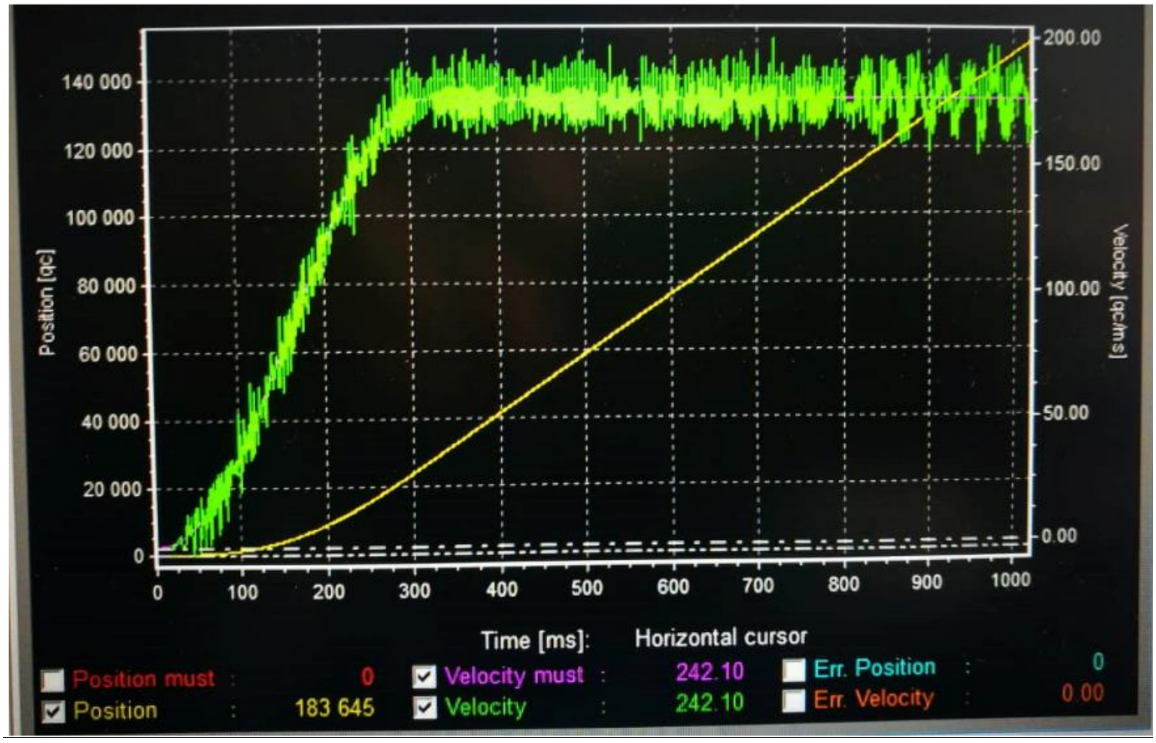
3 Oprava chyb na robote

Ako bolo spomenute vyzssie ked posleme command s cislom 6 tak nam robot vrati aktualne rychlosti kolies. Pri skusani tejto funkcionality sme narazili na problem. Ked sme sa robota spytali na jeho rychlosti dostali sme retazec, ktory obsahoval nahodne velke cisla. Tieto cisla za menili ked sme zadavali nejake hodnoty pre rchlosti kolies aby sa robot hybal.

```
{"LeftWheelSpeed"=236223201280 "RightWheelSpeed"=4294967296}
```

Tu vidime priklad obdrzanej spravy. Jeden z napadov bolo premenit toto cislo na desatinne cisla, kedze sme mu posielali desatinne cisla. Problem je v tom, ze ked posielame request na nastavenie rychlosti kolies tak kod na robote funguje tak ze si to premeni na cele cisla v rozsahu 0 az 1000 a to je hodnota, na ktoru nastavy rychlost otacania kolies. Na druhu stranu ked si vypytame od robota rychlosti kolies tak zobrie informaciu z enkoderov a posle nam to bez spracovania.

Po dokladnom prestudovani kodu sme zitili, ze hodnoty ktore nam posiela nie su ani vytahovane z enkoderov spravnou funkciou. Prote sme ju zmenili a zacali sme dostavat hodnoty, s ktorymi by sa uz mohlo dat pracovat.



Obr. 3.1: Prechodová charakteristika rýchlosti kolies [4].

3.1 Lorem Ipsum

Tento text bol prevzaty zo stranky `www.lipsum.com` [6], preto ho treba riadne odcitovat.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum [6].

Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English [6]. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

Where does it come from? Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source [6]. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum"(The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32 [6].

4 Recitácia

Citujem všetky zdroje v **bibliography.bib**, [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22].
Good luck.

5 Rovnice

Pytagorova veta je definovana vztahom:

$$a^2 + b^2 = c^2 \tag{5.1}$$

Dosadenim (5.1) do linearneho modelu

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{5.2}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{5.3}$$

zistime, ze tento LaTeXtemplate funguje, dokonca mozeme vkladat aj inline rovnice $D = b^2 - 4ac$.

6 Možnosti anonymizácie

Anonymizácia znamená zmena alebo úprava údajov tak, aby sa podľa nich nedala jednoznačne určiť osoba, ktorej tieto údaje patria [8]. Existuje niekoľko spôsobov, ktorými môžeme dosiahnuť rôznu úroveň anonymizácie na internete: od mazania cookies súborov po ukončení prehliadania webových stránok až po používanie operačných systémov, ktoré sú na anonymite založené; od bezplatných možností až po komerčné verzie.
Nasleduje priblíženie niektorých možnosti anonymizácie.

6.1 Súkromné prehliadanie

Najpoužívanejšie internetové prehliadače súčasnosti majú v sebe zabudovanú funkcionalitu, ktorá dokáže čiastočne anonymizovať prístup na internet. Táto funkcionalita blokuje ukladanie navštívených stránok do histórie a nezaznamenáva súbory, ktoré sa stiahnu z internetu. sw a hw sú skratky.

6.2 Anonymná sieť

Anonymná sieť je sieť serverov, medzi ktorými dáta prechádzajú šifrované. V anonymných sieťach dáta prechádzajú z počítača používateľa, odkiaľ bola požiadavka poslaná, cez viaceré proxy smerovače, z ktorých každý správu doplní o smerovanie a zašifruje vlastným kľúčom. Cesta od ...

6.3 Funkcionalita

Rozšírenie tiež okrem splnenia špecifikácie malo pre prehľadnosť a overenie funkčnosti zobrazovať údaje, ktoré boli na server odoslané. Zoznam údajov odoslaných na server, sa mal ukladať do krátkodobej histórie, aby nemal používateľ k dispozícií len najnovšie údaje, ale aj údaje odoslané v nejakom časovom období. Nejaky listing z priloh C.1.

6.3.1 Funkcionalita2

Samozrejmosťou bolo nastavenie zapnutia rozšírenia pri štarte, prípadne interval zmeny odosielaných údajov.

6.4 Vzhľad

Dôležitou požiadavkou kladenou na rozšírenie bolo príjemné používateľské rozhranie. Z tohto dôvodu malo rozšírenie obsahovať zoznam modifikovaných vlastností a tlačidlo pre prístup k nastaveniam rozšírenia v jednoduchej a praktickej forme. Predpokladaný vzhľad je zobrazený na obrázku č. 6.1. Dôležitou požiadavkou kladenou na rozšírenie bolo príjemné používateľské rozhranie.[7] Z tohto dôvodu malo rozšírenie obsahovať zoznam modifikovaných vlastností a

Tabuľka 6.1: Moduly a ich funkcie pri anonymizácii

	Funkcia													
						Modifikácia								
Modul	zobrazenie hlavičky	blokovanie skriptov	zmena IP	zmena lokalizácie	zmazanie/blokovanie cookies	blokovanie trackerov	popis	používateľský agent	kódové označenie prehliadača	názov prehliadača	verzia prehliadača	platforma	výrobca prehliadača	označenie výrobcu prehliadača
User agent switcher							X	X	X	X	X	X	X	X
Ghostery					X	X								
Better privacy					X									
Anonymox			X	X	X		X	X						
Modify headers					X			X						
Request policy						X								
Live HTTP headers	X													
User agent awitcher for chrome							X	X						
Header hacker							X	X	X	X	X	X	X	X
Mod header							X	X	X	X	X	X	X	X
Script no		X												
No script		X												
Proxify it			X	X										
I'm not here				X										
Get anonymous personal edition		X	X	X	X	X								
Anonymous browsing toolbar			X	X										
Easy hide your IP and surf anonymously			X	X				X	X	X	X			

Obr. 6.1: Predpokladaný vzhľad rozšírenia

tlačidlo pre prístup k nastaveniam rozšírenia v jednoduchej a praktickej forme. Predpokladaný vzhľad je zobrazený na obrázku č. 6.1.

```
/* Hello World program */

#include<stdio.h>

struct cpu_info {
      long unsigned utime, ntime, stime, itime;
      long unsigned iowtime, irqtime, sirqtime;
};

main()
{
      printf("Hello World");
}
```

Výpis 1: Ukážka algoritmu

Algoritmus 1 Ukážka príkazov pre algorithmic

```
   <text>
   if <condition> then
       <text>
   else
       <text>
   end if
   if <condition> then
       <text>
   else if <condition> then
       <text>
   end if
   for <condition> do
       <text>
   end for
   for <condition> to <condition> do
       <text>
   end for
   for all <condition> do
       <text>
   end for
   while <condition> do
       <text>
   end while
   repeat
       <text>
   until <condition>
   loop
       <text>
   end loop
Require: <text>
Ensure: <text>
   return <text>
   print <text> {<text>} and , or , xor , not , to , true, false
```

Záver

Conclusion is going to be where?
Here.

Zoznam použitej literatúry

1. *ROS2 documentation* [online] [cit. 2022-12-23]. Dostupné z : `https://docs.ros.org/en/humble/index.html`.
2. *ROS2 from the Ground Up* [online] [cit. 2022-12-23]. Dostupné z : `https://medium.com/@nullbyte.in/ros2-from-the-ground-up-part-1-an-introduction-to-the-robot-operating-system-4c2065c5e032`.
3. *ChatGPT* [online] [cit. 2022-12-24]. Dostupné z : `https://chat.openai.com/chat`.
4. BC. MAREK PACALAJ, BC. TOMÁŠ KÚTIK, BC. DOMINIK GULA, BC. DÁVID PAVLIČ, BC. DANIEL ĎURKOVIČ. *Mobilný podstavec pre robota*. 2019.
5. BC. MAREK PACALAJ, BC. TOMÁŠ KÚTIK, BC. DOMINIK GULA, BC. DÁVID PAVLIČ, BC. DANIEL ĎURKOVIČ. *Dokumentacia k softwaru robota BlackMetal*. 2019.
6. *Lorem Ipsum* [online] [cit. 2020-11-30]. Dostupné z : `https://lipsum.com/`.
7. BRATKOVÁ, Eva (zost.). *Metody citování literatury a strukturování bibliografických záznamů podle mezinárodních norem ISO 690 a ISO 690-2: metodický materiál pro autory vysokoškolských kvalifikačních prací* [online]. Verze 2.0, aktualiz. a rozšíř. Praha: Odborná komise pro otázky elektronického zpřístupňování vysokoškolských kvalifikačních prací, Asociace knihoven vysokých škol ČR, 2008-12-22 [cit. 2011-02-02]. Dostupné z : `http://www.evskp.cz/SD/4c.pdf`.
8. BORGMAN, Christine L. *From Gutenberg to the global information infrastructure: access to information in the networked world*. First. Cambridge (Mass): The MIT Press, 2003. ISBN 978-3-16-148410-0.
9. GREENBERG, David. Camel drivers and gatecrashers: quality control in the digital research library. In: HAWKINS, B.L and BATTIN, P (eds.). *The mirage of continuity: reconfiguring academic information resources for the 21st century*. Washington (D.C.): Council on Library and Information Resources; Association of American Universities, 1998, s. 105–116.
10. LYNCH, C. Where do we go from here? the next decade for digital libraries. *DLib Magazine* [online]. 2005, vol. 11, no. 7/8 [cit. 2005-08-15]. ISSN 1082-9873. Dostupné z : `http://www.dlib.org/dlib/july05/lynch/07lynch.html`.

11. DĚŤA, Hugh a RYCHLÍK, Tomáš. *A big paper: Podtitul* [online]. 2. vyd. Praha: Academia, 1991 [cit. 2011-01-12]. Pokusná edice. ISBN 978-3-16-148410-0. Dostupné z : http://pokus.cz.

12. DĚŤA, Hugh, RYCHLÍK, Tomáš, DALŠÍ, Pepa, SPOUSTA, Pepa, SKORO, Moc, ALE, Nestačí a HODNĚ. *Úplně úžasná knížka*. 3. vyd. Praha, 1991.

13. DĚŤA, Hugh, RYCHLÍK, Tomáš, DALŠÍ, Pepa, SPOUSTA, Pepa, SKORO, Moc, ALE, Nestačí and HODNĚ. *Úplně úžasná knížka*. 3rd ed. Praha: MIT Press, 1991.

14. FREELY, I.P. A small paper: Podtitulek. *The journal of small papers*. 1997, roč. 1, č. 3, s. 2–5. to appear.

15. JASS, Hugh. A big paper. *The journal of big papers*. 1991, roč. 23.

16. Titulek. *The journal of big papers*. 1991, roč. 12, č. 2, s. 22–44. Dostupné z DOI: 10.112.22/jkn.

17. KOLLMANNOVÁ, Ludmila, BUBENÍKOVÁ, Libuše a KOPECKÁ, Alena. *Angličtina pro samouky*. 5. vyd. Praha: Státní pedagogické nakladatelství, 1977. Učebnice pro samouky, č. 4. ISBN 978-3-16-148410-0.

18. NOVOTNÁ, Pepina. Podkapitola. In: KOLLMANNOVÁ, Ludmila, BUBENÍKOVÁ, Libuše a KOPECKÁ, Alena. *Angličtina pro samouky*. 5. vyd. Praha: Státní pedagogické nakladatelství, 1977, kap. 2., s. 22–29. Učebnice pro samouky, č. 4. ISBN 978-3-16-148410-0.

20. KNUTH, Donald. Journeys of TEX. *TUGBoat*. 2003– , vol. 17, no. 3, s. 12–22. ISSN 1222-3333. Dostupné tiež z: http://tugboat.tug.org/kkk.pdf.

21. GENIÁLNÍ, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007. ISBN 978-3-16-148410-0.

22. VLAŠTOVKA, Josef. Velmi zajímavý článek. In: GENIÁLNÍ, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007, s. 22–45. ISBN 978-3-16-148410-0.

Prílohy

A Štruktúra elektronického nosiča

/CHANGELOG.md

· file describing changes made to FEIstyle

/example.tex

· main example $.tex$ file for diploma thesis

/example_paper.tex

· example $.tex$ file for seminar paper

/Makefile

· simply Makefile – build system

/fei.sublime-project

· is project file with build in Build System for Sublime Text 3

/img

· folder with images

/includes

· files with content

/bibliography.bib

· bibliography file

/attachmentA.tex

· this very file

B Algoritmus

Algoritmus B.1 Vypočítaj $y = x^n$

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

$y \Leftarrow 1$
if $n < 0$ **then**
$X \Leftarrow 1/x$
$N \Leftarrow -n$
else
$X \Leftarrow x$
$N \Leftarrow n$
end if
while $N \neq 0$ **do**
if N is even **then**
$X \Leftarrow X \times X$
$N \Leftarrow N/2$
else {N is odd}
$y \Leftarrow y \times X$
$N \Leftarrow N - 1$
end if
end while

C Výpis sublime

```
../../ fei .sublime-project
```

Výpis C.1: Ukážka sublime-project