



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**NEKONVENČNÍ MODEL VOZIDLA SE SAMOŘÍDICÍMI
SCHOPNOSTMI**

UNCONVENTIONAL VEHICLE MODEL WITH SELF-DRIVING ABILITIES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

REBEKA TYDOROVÁ

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. MICHAL BIDLO, Ph.D.

BRNO 2025

Zadání bakalářské práce



164823

Ústav: Ústav počítačových systémů (UPSY)
Studentka: **Tydorová Rebeka**
Program: Informační technologie
Název: **Nekonvenční model vozidla se samořidícími schopnostmi**
Kategorie: Vestavěné systémy
Akademický rok: 2024/25

Zadání:

1. Nastudujte základy problematiky samočinného řízení vozidel se zaměřením na modely (vytvořené např. z vhodné stavebnice) provozovatelné v interiérových podmínkách.
2. Po dohodě s vedoucím práce zvolte vhodný koncept vozidla odlišný od tradičních podvozků na čtyřech kolech s jednou řízenou nápravou. Vozidlo realizujte z vhodné stavebnice v podobě fyzického modelu.
3. Pro model vozidla z bodu 2 navrhnete a implementujte alespoň jeden scénář použití zahrnující prvek samořízení. Vytvořte příslušný řídicí software.
4. Pro daný rozsah vybraných parametrů provozních podmínek (např. míra osvětlení prostoru, rychlost pohybu apod.) proveďte sadu experimentů s cílem optimalizovat spolehlivost a přesnost řízení v definovaných podmínkách.
5. Zhodnotte dosažené výsledky a diskutujte možnosti pokračování projektu.

Literatura:

- Dle pokynů vedoucího projektu.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 a 2 zadání, demonstrace rozpracovaného konceptu samořízení z bodu 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Bidlo Michal, doc. Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1.11.2024
Termín pro odevzdání: 14.5.2025
Datum schválení: 31.10.2024

Abstrakt

Cielom tejto bakalárskej práce bolo navrhnuť a zrealizovať nekonvenčný model autonómneho vozidla schopného pohybu v interiérovom prostredí na základe vizuálnych podnetov. Dvojkolesový model bol vytvorený pomocou 3D tlače a riadený platformou Raspberry Pi. Vozidlo bolo testované v troch rôznych scenároch: (1) sledovanie čiary pomocou priebežného vyhodnocovania jej polohy, (2) smerovanie k najvzdialenejšiemu detegovanému bodu na trase a (3) navigácia podľa sekvencie farebných značiek. Každý prístup bol navrhnutý, implementovaný a experimentálne overený v rôznych svetelných a pohybových podmienkach. Výsledky testovania poukázali na výhody a obmedzenia jednotlivých metód – zatiaľ čo prvý prístup bol jednoduchý na implementáciu, vykazoval nízku stabilitu. Druhý poskytoval plynulejšie riadenie, no mal problémy s ostrými zákrutami. Tretí model tak isto nezvládal prudké zákruty a bol citlivý na svetelné podmienky, ale vynikal v plynulosti a rýchlosti. Hlavným prínosom tejto práce je demonštrácia možností realizácie rôznych prístupov k autonómnemu riadeniu aj s jednoduchým a cenovo dostupným hardvérom. Zistenia môžu slúžiť ako základ pre ďalší vývoj autonómnych modelov s dôrazom na spoľahlivosť v rôznych podmienkach a optimalizáciu riadiacich algoritmov.

Abstract

The goal of this bachelor's thesis was to design and develop an unconventional model of a self-driving vehicle capable of indoor navigation based on visual cues. A two-wheeled model was created using 3D printing and controlled via the Raspberry Pi platform. The vehicle was tested in three different scenarios: (1) following a line through continuous assessment of its position, (2) driving toward the closest detected point on a predefined route, and (3) navigating using a sequence of colored signs. Each scenario was designed, implemented, and tested under various lighting and movement conditions. The testing results revealed the strengths and limitations of each method. The first method was simple to implement but exhibited low stability. The second offered smoother control but struggled with sharp turns. The third also had difficulty with sharp turns and was sensitive to changes in lighting, but it excelled in speed and fluidity. The main contribution of this work is the demonstration of different approaches to autonomous vehicle navigation using simple and affordable hardware. These findings provide a foundation for further development of autonomous models, with a focus on improving reliability under varying conditions and optimizing control algorithms.

Klíčové slová

autonómne vozidlo, nekonvenčný podvozok, model vozidla, Raspberry Pi, sledovanie dráhy, vizuálna navigácia

Keywords

autonomous vehicle, unconventional chassis, vehicle model, Raspberry Pi, path tracking, visual navigation

Citácia

TYDOROVÁ, Rebeka. *Nekonvenční model vozidla se samořídícími schopnostmi*. Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Michal Bidlo, Ph.D.

Nekonvenční model vozidla se samořídícími schopnostmi

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracovala samostatne pod vedením pana doc. Ing. Michala Bidla, Ph.D. Uviedla som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpala.

.....
Rebeka Tydorová
13. mája 2025

Podakovanie

Touto cestou by som rada podakovala vedúcemu mojej bakalárskej práce, doc. Ing. Michal Bidlo, Ph.D, za jeho ochotu, trpezlivosť a odborné usmernenie, ktoré mi pomohli úspešne dokončiť túto prácu. Ďakujem aj mojej rodine a priateľom za podporu, ktorú mi poskytovali počas celého štúdia.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 4 |
| 2 | Teoretický základ | 5 |
| 2.1 | Autonómne riadenie vozidiel | 5 |
| 2.2 | Alternatívne koncepty vozidiel | 9 |
| 2.3 | Použité technológie a stavebnice | 14 |
| 3 | Návrh a realizácia modelu | 20 |
| 3.1 | Voľba koncepcie vozidla | 20 |
| 3.2 | Elektrické príslušenstvo | 22 |
| 3.3 | Návrh riadiaceho softvéru | 23 |
| 4 | Návrh scenárov autonómnej jazdy a riadiacich algoritmov | 26 |
| 4.1 | Scenár 1 – jazda po čiare | 26 |
| 4.2 | Scenár 2 – vylepšenie plynulosti jazdy | 27 |
| 4.3 | Scenár 3 – jazda po značkách | 30 |
| 5 | Experimentálna časť a testovanie | 31 |
| 5.1 | Definovanie parametrov testovania | 31 |
| 5.2 | Analýza výsledkov | 31 |
| 5.3 | Zhodnotenie a diskusia | 34 |
| 6 | Záver | 37 |
| | Literatúra | 38 |
| | Zoznam príloh | 40 |
| A | Model vozidla | 41 |

Zoznam obrázkov

| | | |
|-----|--|----|
| 2.1 | Príklad senzorickej fúzie, prevzaté z [33] | 8 |
| 2.2 | Model vozidla s diferenčným pohonom, prevzaté z [31] | 10 |
| 2.3 | Model trojkolesového vozidla, prevzaté z [30] | 10 |
| 2.4 | Model vozidla s holonomným podvozkom, prevzaté z [13] | 11 |
| 2.5 | Model vozidla s pásovým podvozkom, prevzaté z [17] | 11 |
| 2.6 | Model vozidla s kĺbovým podvozkom, prevzaté z [35] | 12 |
| 2.7 | Model viacnohého robota, prevzaté z [7] | 12 |
| 2.8 | Model vozidla so sférickým podvozkom, prevzaté z [38] | 13 |
| 2.9 | Model vozidla s kolesovo-nožičkovým systémom, prevzaté z [2] | 13 |
| 3.1 | Prototyp prvého modelu vozidla vytvorený z kartónu | 20 |
| 3.2 | Prvý model skonštruovaný zo súčiastok vytlačených na 3D tlačiarni | 21 |
| 3.3 | Kolesá rôznej veľkosti a hustoty výplne testované pre model vozidla | 21 |
| 3.4 | Druhý model vymodelovaný v prostredí Tinkercad a vytlačený na 3D tlačiarni | 22 |
| 3.5 | Schéma zapojenia elektrického príslušenstva modelu vozidla | 23 |
| 3.6 | Bloková schéma softvéru pre riadenie modelu vozidla | 23 |
| 3.7 | Spracovanie obrazu – detekcia navigačných značiek | 24 |
| 3.8 | Vývojový diagram riadiaceho algoritmu | 25 |
| 4.1 | Rozmery zorného poľa kamery pripevnenej na model vozidla | 28 |
| 4.2 | Výpočet uhla medzi stredom obrazu a nájdeným bodom | 28 |
| A.1 | Prototyp prvého modelu vozidla | 41 |
| A.2 | Prvý model vozidla | 42 |
| A.3 | Druhý model vozidla v prostredí Tinkercad | 43 |
| A.4 | Druhý model vozidla | 44 |

Zoznam tabuliek

| | | |
|-----|--|----|
| 2.1 | Porovnanie výhod a nevýhod alternatívnych podvozkových koncepcií | 14 |
| 2.2 | Porovnanie platforiem pre autonómne vozidlá na základe kľúčových vlastností pre účely tejto práce | 17 |
| 4.1 | Namerané hodnoty pohybu modelu | 27 |
| 5.1 | Zvládnuté zákruty pri prednastavenej rýchlosti prvého scenára | 32 |
| 5.2 | Zvládnuté zákruty pri zvýšenej rýchlosti prvého scenára | 32 |
| 5.3 | Zvládnuté zákruty pri maximálnej rýchlosti prvého scenára | 32 |
| 5.4 | Zvládnuté zákruty pri prednastavenej rýchlosti druhého scenára | 33 |
| 5.5 | Zvládnuté zákruty pri prednastavenej rýchlosti tretieho scenára | 34 |
| 5.6 | Zvládnuté zákruty pri maximálnej rýchlosti tretieho scenára | 34 |
| 5.7 | Porovnanie scenárov podľa kľúčových vlastností | 35 |

Kapitola 1

Úvod

Autonómne riadiace systémy sa v posledných desaťročiach stali kľúčovou súčasťou technologického pokroku v rôznych oblastiach nášho života. Ich hlavnou úlohou je umožniť zariadeniam samostatne sa rozhodovať a vykonávať jednotlivé zadané úkony bez neustáleho zásahu a kontroly zo strany človeka. Využívajú sa napríklad v automobilovom priemysle pri vývoji samojazdiacich áut, v priemyselnej automatizácii, v letectve, v zdravotníctve, či pri vývoji inteligentných domácností. Stretávame sa s nimi však už aj v reštauračných zariadeniach, kde robotická obsluha roznáša jedlo a nápoje. V jadre týchto systémov sa nachádza schopnosť vnímať okolie, spracovávať vstupné údaje zo senzorov a následne podľa nich riadiť činnosť výstupných zariadení – najčastejšie motorov.

S rastúcou dostupnosťou výpočtovej techniky a jej rastúcim výkonom sa v posledných rokoch otvárajú dvere k vývoju jednoduchších autonómnych systémov aj pre študentov a menšie výskumné tímy. Významnú úlohu v tomto smere zohráva aj platforma Raspberry Pi – miniatúrny počítač s nízkou spotrebou a cenovou dostupnosťou, ktorý poskytuje dostatočný výpočtový výkon na realizáciu rôznych projektov, od spracovania obrazu až po ovládanie hardvéru v reálnom čase. V kombinácii s ďalšími dostupnými komponentmi, ako sú kamery, snímače či motorové moduly, predstavuje Raspberry Pi atraktívnu platformu pre výučbu základov robotiky a umelej inteligencie.

Jednou z klasických úloh, ktorá sa často rieši v rámci vzdelávacích projektov, je návrh autonómneho mobilného zariadenia – napríklad malého robotického vozidla, ktoré sa dokáže samostatne pohybovať v prostredí podľa vizuálnych alebo iných podnetov. Takýto projekt v sebe spája viaceré zložky – elektroniku, programovanie, spracovanie obrazových dát, návrh algoritmov a praktické testovanie v reálnych podmienkach. Okrem rozvoja technických zručností pomáha tento typ úlohy študentom pochopiť základné princípy, na ktorých sú postavené aj zložitejšie autonómne systémy používané v praxi.

Táto bakalárska práca sa zaoberá návrhom a realizáciou jednoduchého autonómneho vozidla riadeného mikropočítačom Raspberry Pi. Vozidlo sa orientuje na základe vizuálneho vstupu z kamery a jeho pohyb je riadený tak, aby sledovalo zvolené vizuálne prvky umiestnené v priestore. Cieľom práce je preskúmať možnosti nízkonákladového hardvéru v oblasti autonómneho riadenia, navrhnúť základné algoritmy na rozpoznávanie obrazu a aplikovať ich v praxi pri ovládaní pohybového systému. Práca má zároveň slúžiť ako ukážka toho, že aj s jednoduchými prostriedkami je možné vytvoriť funkčný a zaujímavý robotický systém.

Kapitola 2

Teoretický základ

Táto kapitola poskytuje teoretický základ potrebný na pochopenie fungovania autonómnych systémov a opisuje kľúčové technológie, ktoré umožňujú samostatnú jazdu vozidiel bez zásahu vodiča. Venuje sa nielen historickému vývoju autonómneho riadenia a jeho základným princípom, ale aj sensorom, ktoré vozidlám umožňujú vnímať okolie, a algoritmom, ktoré zabezpečujú plánovanie a realizáciu pohybu.

Zároveň sa zameriava na alternatívne podvozkové koncepcie vhodné pre malé autonómne platformy, ktoré sa využívajú najmä v oblasti výskumu a výučby. Tieto netradičné riešenia umožňujú efektívne testovanie algoritmov v prostrediach, kde klasická automobilová konštrukcia nemusí byť ideálna.

Záver kapitoly bude venovaný predstaveniu vývojovej platformy, nástrojov a softvérového prostredia, ktoré boli využité pri implementácii systému.

2.1 Autonómne riadenie vozidiel

Autonómne vozidlá predstavujú jednu z populárnych tém súčasnej doby. Cieľom tejto technológie je vytvoriť vozidlá, ktoré dokážu bezpečne a spoľahlivo vykonávať všetky činnosti vodiča bez nutnosti jeho zásahu [24]. V tejto časti sa zameriame na vývoj autonómneho riadenia od jeho prvých pokusov až po súčasné technologické trendy, popíšeme základné princípy fungovania autonómnych systémov a technológie, ktoré ich umožňujú. Táto kapitola bola prevzatá z [33].

História a vývoj autonómnych vozidiel

Aj keď sa pojem *autonómne vozidlo* javí ako novodobá záležitosť, jeho korene siahajú až do 16. storočia, kedy Leonardo da Vinci vytvoril dizajn samohybného vozíka. Tento vynález sa často označuje nielen ako prvé autonómne vozidlo, ale aj ako prvý robot v dejinách. Nešlo síce o motorové vozidlo v dnešnom slova zmysle, skôr o mechanizmus poháňaný pružinou, no niektoré jeho časti sa nápadne podobajú prvkom, ktoré poznáme aj v súčasných vozidlách – napríklad predprogramovateľný smer pohybu, parkovacia brzda alebo pružinový pohon. V modernej dobe bol podľa da Vinciho návrhu postavený funkčný model, ktorý potvrdil, že jeho myšlienka bola technicky uskutočniteľná.

Ďalší známy pokus o samoriadiace vozidlo sa objavil v roku 1925, keď elektrotechnik Francis P. Houdina predstavil v uliciach New Yorku vozidlo ovládané na diaľku pomocou rádiových vln. Hoci tento projekt skončil nehodou a bol následne ukončený, znamenal prvý reálny experiment so vzdialeným riadením vozidla v bežnej premávke. O niekoľko rokov

neskôr, počas svetovej výstavy Futurama v New Yorku v roku 1939, spoločnosť General Motors predstavila víziu budúcnosti, v ktorej sa mali vozidlá pohybovať po cestách vybavených rádiovými signálmi a magnetmi. Neskôr General Motors rozvíjala tieto myšlienky aj prostredníctvom série futuristických koncepčných vozidiel Firebird.

V nasledujúcich desaťročiach sa vývoj autonómnych systémov posúval ďalej. V rámci vesmírneho výskumu bol vyvinutý vozík schopný autonómnej navigácie po povrchu Mesiaca, čo inšpirovalo výskumné tímy k vytvoreniu prvých pozemských vozidiel s použitím počítačového videnia. V 80. rokoch minulého storočia vznikli prvé prototypy vozidiel, ktoré dokázali jazdiť rýchlosťou až 20 míľ za hodinu bez ľudského zásahu.

V Európe zohral dôležitú úlohu projekt PROMETHEUS, ktorý združoval výskumné inštitúcie a automobilky. Vyvrcholil v roku 1994 jazdou na diaľnici, kde vozidlo zvládlo sledovanie jazdných pruhov, obchádzanie iných vozidiel a zmenu jazdného pruhu – a to všetko bez zásahu vodiča.

Významným milníkom bola séria súťaží v rokoch 2004 až 2007, ktorú organizovalo Americké Ministerstvo obrany (Defense Advanced Research Projects Agency – DARPA). Súťažné tímy museli zostrojiť autonómne vozidlo a úspešne prejsť stanovenú trasu. Prvý ročník dopadol neúspešne, kde sa žiadnemu tímu nepodarilo prejsť cieľovou čiarou. Nasledujúci rok však niekoľko vozidiel prešlo až do konca, čo dokázalo, že technológia autonómneho riadenia môže fungovať. Vďaka tomu boli inšpirované viaceré firmy a výrobcovia automobilov ako BMW, Audi, Daimler, Google, Tesla, Uber a Baidu a spustili vlastné programy vývoja autonómnych vozidiel.

História vývoja autonómnych vozidiel tak ukazuje rýchly prechod od teoretických vízií k praktickému testovaniu a implementácii. Hoci úplne autonómna jazda zatiaľ nie je bežne dostupná, technologický pokrok neustále napreduje a prináša nové výzvy v oblasti bezpečnosti, legislatívy a spoločenského prijatia [25].

Základné princípy samočinného riadenia

Samoriadiace vozidlo je vo väčšine literatúry definované ako vozidlo, ktoré je schopné samostatne vykonávať úlohy vodiča. Využíva na to plne automatizovaný systém obsahujúci pokročilý softvér, ktorý spracováva informácie v reálnom čase, a tým umožňuje vozidlu pohotovo reagovať na vonkajšie podnety [24].

Aby bolo vozidlo schopné jazdiť autonómne, musí zvládať tri základné úlohy:

- **Vnímanie prostredia,**
- **Plánovanie,**
- **Vykonanie rozhodnutí.**

Vnímanie prostredia by sa dalo prirovnať k ľudskému videniu. Vozidlo zisťuje, kde sa nachádza a čo sa deje okolo neho. Hlavné funkcie v tejto oblasti zahŕňajú:

- **Lokalizáciu** – určenie presnej polohy vozidla v priestore,
- **Mapovanie** – vytváranie a aktualizáciu mapy okolia,
- **Detekciu objektov** – rozpoznávanie a klasifikáciu objektov v okolí pomocou senzorov.

Plánovanie a rozhodovanie funguje na základe získaných informácií. Cieľom je identifikovať bezpečný a efektívny spôsob, ako sa vozidlo dostane do cieľovej destinácie. Plánovanie prebieha v troch vrstvách:

- **Plánovanie trasy (route planning)** – najvyššia vrstva, ktorá určuje optimálnu trasu z bodu A do bodu B s ohľadom na aktuálne dopravné podmienky a mapové podklady. Podobný princíp poznáme z bežných GPS navigácií.
- **Plánovanie správania (behavioral planning)** – rozhodovanie o správaní vozidla v konkrétnych situáciách. Systém zvažuje dopravné pravidlá, predvída správanie ostatných účastníkov premávky a rozhoduje napríklad o zmene jazdného pruhu, dávaní prednosti alebo obchádzaní prekážky.
- **Plánovanie pohybu (motion planning)** – najnižšia vrstva, ktorá určuje konkrétnu trajektóriu pohybu vozidla, teda napríklad smer jazdy, rýchlosť či akceleráciu. Výsledkom je pohybový plán, ktorý vozidlo vykoná.

Tieto tri vrstvy spolupracujú a neustále sa aktualizujú v reálnom čase, aby vozidlo dokázalo reagovať aj na dynamicky sa meniace situácie.

Posledným krokom je vykonanie rozhodnutí z predchádzajúcich krokov, čo vo finále predstavuje otočenie volantom, pridanie plynu alebo zabrzdzenie. Táto fáza je zároveň hlavným cieľom riadenia vozidla a sústreďí sa na plynulú, bezpečnú a predvídateľnú jazdu.

Základné princípy samočinného riadenia teda spočívajú v tom, že vozidlo dokáže samostatne vnímať, vyhodnocovať a konať – všetko bez priameho zásahu človeka. Vývoj týchto systémov je zložitý a zahŕňa široké spektrum disciplín, od sensoriky a pokročilých algoritmov, cez strojové učenie, až po kybernetiku a dopravné inžinierstvo.

Senzory a technológie používané v autonómnych systémoch

Aby mohlo autonómne vozidlo bezpečne fungovať, musí neustále zhromažďovať informácie o svojom okolí. Na tento účel využíva kombináciu rôznych sensorov a technologických riešení, ktoré spolu vytvárajú komplexný obraz o prostredí. Tieto systémy spolu tvoria základ autonómneho riadenia.

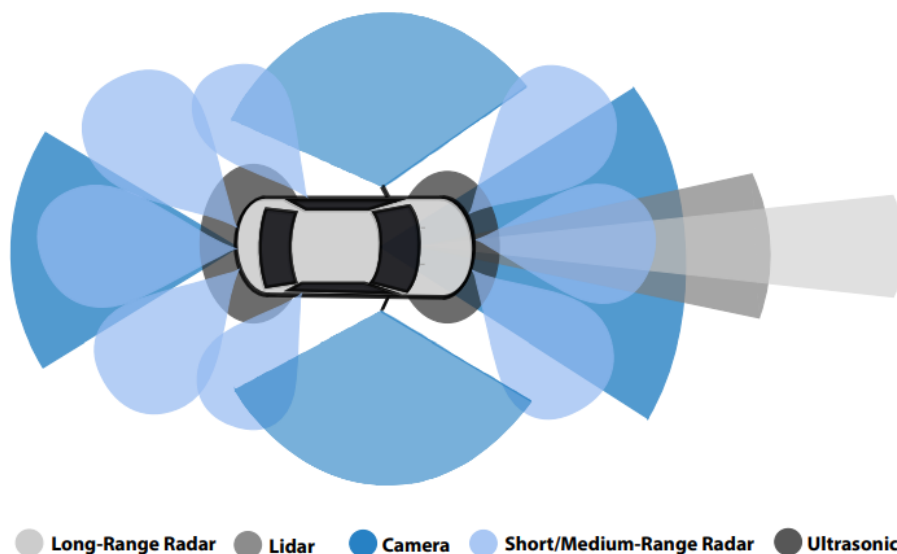
Základným prvkom autonómneho systému sú senzory. Každý typ senzora má svoje silné aj slabé stránky, preto sa vo vozidlách používajú spoločne v rámci takzvanej sensorovej fúzie:

- **Kamery** – poskytujú vizuálne informácie podobné ľudskému oku. Umožňujú rozpoznávať dopravné značky, semaforey, vozidlá, chodcov a iné prvky cestnej premávky. Ich výhodou je vysoké rozlíšenie a nízke náklady, no za nevýhodu možno považovať horšie fungovanie pri slabom osvetlení alebo zlom počasí.
- **Lidar (Light Detection and Ranging)** – tento senzor pracuje na princípe odrazu laserových lúčov a dokáže presne vytvárať trojrozmerný model okolia. Vďaka vysokej presnosti je využívaný najmä pri detekcii prekážok a pri mapovaní okolia. Ich nevýhodou je skutočnosť, že laserové lúče sa odrážajú aj od malých častíc ako sú hmla, dažďové kvapky alebo prach, čím sa zvyšuje ich citlivosť na okolité prostredie v prípade nepriaznivých podmienok.
- **Radar** – využíva rádiové vlny na detekciu objektov, ich veľkosť, rýchlosť a smer. Je spoľahlivý aj v daždi, hmle či v noci, čo z neho robí dôležitý doplnok kamier a lidarů.

Poskytuje taktiež dostatočné rozlíšenie aj na väčšie vzdialenosti. Medzi jeho nevýhody patrí úzky uhol záberu a slabšia detekcia nemetalických objektov.

- **Ultrazvukové senzory** – fungujú na základe detekcie zvukových vln s frekvenciami nad 20 kHz. Pracujú na krátke vzdialenosti, typicky pri parkovaní alebo manévrovaní v tesnom priestore. Detegujú blízke objekty a prekážky v okolí vozidla. Dobre fungujú vo väčšine situácií, ako aj pri nepriaznivom počasí ako dážď, hmla či sneh, nerobia im problém ani rôzne svetelné podmienky. Nepriaznivo ale na ich funkciu vplýva vysoká rýchlosť vozidla alebo vetra. S ich funkciou môžu taktiež interferovať aj iné vysokofrekvenčné zvuky z okolia.
- **Odometrické senzory** – jedná sa o špeciálne senzory vyvinuté na meranie vzdialenosti prejdenej vozidlom. Výpočet prejdenej úseku spočíva vo vynásobení počtu otočení kolesom obvodom pneumatiky. Poskytujú presné informácie o vzdialenosti, ktorú prešlo vozidlo. Sú ale citlivé na chyby spôsobené nerovnými povrchmi, šmykaním kolies alebo inými faktormi prostredia.

Tieto senzory spolu vytvárajú systém senzorickej fúzie, kde sa dáta z jednotlivých zariadení kombinujú a navzájom korigujú 2.1. Výsledkom je spoľahlivý model okolia, ktorý slúži ako základ pre lokalizáciu, mapovanie, detekciu objektov a následné plánovanie pohybu vozidla.



Obr. 2.1: Príklad senzorickej fúzie, prevzaté z [33]

Okrem sensorov sú neoddeliteľnou súčasťou autonómnych systémov aj technológie, ktoré umožňujú spracovanie údajov, plánovanie pohybu a samotné vykonanie riadiacich rozhodnutí:

- **Senzorová fúzia (Sensor Fusion)** – predstavuje proces, pri ktorom sa kombinujú dáta z viacerých sensorov, ktoré sú popísané vyššie. Výsledkom je komplexnejší a presnejší obraz okolia vozidla, než aký by poskytol ktorýkoľvek senzor samostatne.
- **Navigačný systém** – zahŕňa systémy, ktoré slúžia na určenie polohy vozidla a plánovanie trasy. Patrí sem:

- **GPS (Global Positioning System)** – pomocou digitálnej mapy a počítača prítomného priamo vo vozidle sú vozidlá schopné automaticky lokalizovať svoju polohu na mape, vypočítať trasu a navigovať vozidlo do cieľovej destinácie.
 - **SLAM (Simultaneous Localization and Mapping)** – umožňuje vozidlu súčasne vytvárať mapu neznámeho prostredia a určovať svoju presnú polohu v nej.
 - **IMU (Inertial Measurement Unit)** – umožňuje autonómnym vozidlám určiť svoju pozíciu a smer, akým sú otočené pomocou akcelerometrov, gyroskopov a magnetometrov.
- **Komunikácia** – zahŕňa komunikáciu medzi vozidlami, vozidla s infraštruktúrou (napríklad dopravné značky, semaforey), vozidla s chodcami a vozidla so sieťou. Jej cieľom je zvýšiť bezpečnosť cestnej premávky, zvýšiť efektívnosť riadenia vozidla a optimalizovať využitie energie.
 - **Riadenie** – cieľom je vykonávať rozhodnutia na základe zozbieraných dát z ostatných technológií. Zahŕňa udržiavanie vozidla medzi čiarami, automatické upravovanie rýchlosti na udržanie vzdialenosti od ostatných vozidiel a zmenu pruhov.
 - **Umelá inteligencia a strojové učenie** – využívajú sa na predikciu správania účastníkov cestnej premávky, zlepšovanie rozpoznávania objektov a adaptáciu na nové situácie. Vďaka tomu sú autonómne systémy čoraz inteligentnejšie a bezpečnejšie.

Kombináciou týchto senzorov a technológií vzniká komplexný systém, ktorý umožňuje vozidlu samostatne vnímať, vyhodnocovať a reagovať na okolie, čo tvorí základ autonómneho riadenia.

2.2 Alternatívne koncepty vozidiel

Tradičné podvozkové koncepcie založené na štyroch kolesách s jednou riadiacou nápravou sú dlhodobo overené a často využívané v oblasti automobilového priemyslu. V oblasti mobilnej robotiky a modelov autonómnych vozidiel, najmä tých určených do interiérových podmienok, však existuje množstvo alternatívnych riešení, ktoré prinášajú špecifické výhody, ale aj nevýhody. Najmä v interiéri, kde je málo miesta, môžu byť vhodnejšie iné riešenia, ktoré sú jednoduchšie na stavbu, ľahšie ovládateľné alebo stabilnejšie. Táto kapitola predstavuje rôzne netradičné typy podvozkov, ktoré sa dajú použiť pri stavbe malých autonómnych vozidiel.

Prehľad netradičných podvozkových riešení

Medzi najčastejšie alternatívne podvozkové koncepcie patria:

- **Diferenčný pohon** – jednou z najjednoduchších a najrozšírenejších koncepcií v mobilnej robotike, obrázok 2.2. Vozidlo má dve hnacie kolesá umiestnené na jednej osi, pričom každé z nich je poháňané samostatným motorom. Riadenie sa dosahuje zmenou rýchlosti kolies – rovnaká rýchlosť znamená pohyb vpred alebo vzad, rozdielna rýchlosť spôsobí otáčanie. Výhodou tohto riešenia je jednoduchosť mechanickej konštrukcie a nenáročnosť na riadiaci softvér. Nevýhodou je nižšia stabilita pri vyšších rýchlostiach a menej presné otáčanie v porovnaní s vozidlami s riadenou nápravou [39].



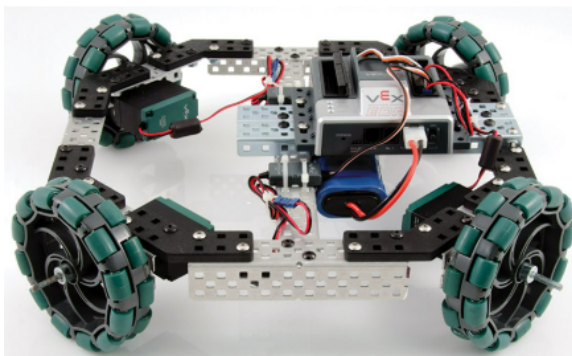
Obr. 2.2: Model vozidla s diferenčným pohonom, prevzaté z [31]

- **Trojkoľesové vozidlá** – zvyčajne využívajú jedno riadiace koleso a dve hnacie kolesá, obrázok 2.3. Trojkoľesové modely sú často kompaktnéjšie ako štvorkolesové a poskytujú dobrú obratnosť pri nízkych rýchlostiach. Zároveň vyžadujú menej mechanických dielov, čo znižuje ich zložitosť. Nevýhodou je však obmedzená stabilita, najmä pri rýchlejšej jazde alebo pri prekonávaní prekážok [34].



Obr. 2.3: Model trojkoľesového vozidla, prevzaté z [30]

- **Holonómne podvozky** – tieto podvozky umožňujú vozidlu pohybovať sa v ľubovoľnom smere bez nutnosti meniť orientáciu celého podvozku. Využívajú špeciálne kolieska, ktoré vďaka nakloneným valčekom zabezpečujú kombinovaný pohyb, obrázok 2.4. Výhodou je mimoriadna manévrovateľnosť a možnosť presného polohovania v obmedzenom priestore, čo je ideálne pre robotiku v interiéri. Nevýhodou je vyššia mechanická zložitosť, náročnejšie programovanie a zvýšené nároky na senzorkú spätnú väzbu [37].



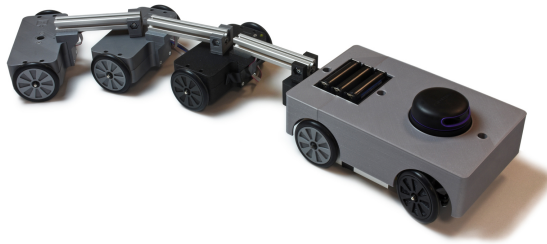
Obr. 2.4: Model vozidla s holonomným podvozkom, prevzaté z [13]

- **Pásové podvozky** – známe najmä z vojenskej techniky a terénnych robotov, využívajú namiesto kolies pásy, obrázok 2.5. Tie poskytujú vysokú trakciu a schopnosť prekonávať nerovnosti či prekážky. Takéto podvozky sú vhodné do nerovného alebo šmykľavého terénu. V interiéri však môžu pôsobiť nemotorne, majú vyššiu spotrebu energie a sú náchylné na opotrebovanie pásov [14].



Obr. 2.5: Model vozidla s pásovým podvozkom, prevzaté z [17]

- **Kĺbové podvozky** – tieto konštrukcie obsahujú aspoň jeden otočný kĺb medzi časťami podvozku, ktorý umožňuje zatáčanie, obrázok 2.6. Používajú sa hlavne pri väčších robotických platformách alebo v priemyselnom prostredí. Výhodou je schopnosť manévrovať v úzkych priestoroch, no realizácia takéhoto systému z bežne dostupných stavebníc je technicky náročnejšia a vyžaduje pokročilé riadiace algoritmy [11].



Obr. 2.6: Model vozidla s kľbovým podvozkom, prevzaté z [35]

- **Hexapody a viacnohý roboti** – základom je konštrukcia, ktorá je inšpirovaná chôdzou živočíchov, napríklad hmyzu. Robot má viacero nôh, obrázok 2.7, ktorými napodobňuje biologický pohyb a je tak schopný prekonať nerovný terén. Má vysokú stabilitu, no jeho mechanika a riadenie sú zložité a majú vyššiu energetickú náročnosť [7].



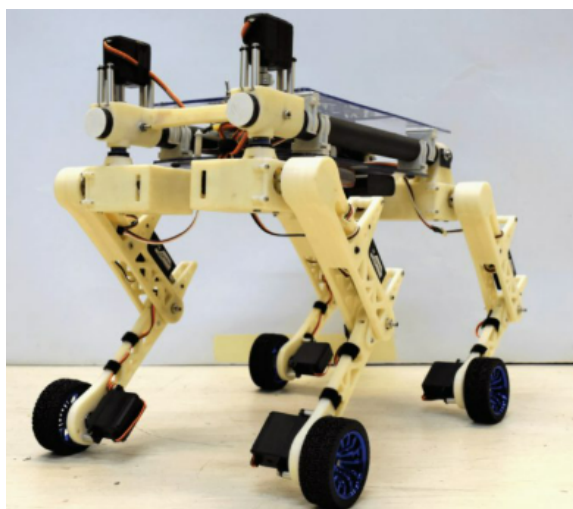
Obr. 2.7: Model viacnohého robota, prevzaté z [7]

- **Sférický podvozok** – pohybujú sa pomocou jednej centrálnej gule, obrázok 2.8, ktorá je ovládaná vnútornými mechanizmami. Zabezpečuje tak plynulý pohyb a poskytuje možnosť pohybu v ľubovoľnom smere. Riadenie je ale náročnejšie a robot nie je vhodný na prekonávanie terénnych nerovností [21].



Obr. 2.8: Model vozidla so sférickým podvozkom, prevzaté z [38]

- **Kolesovo-nožičkové systémy** – kombinujú výhody kolesa (rýchlosť) a nôh (prispôsobivosť terénu), obrázok 2.9. V bežnom režime sa pohybujú ako štandardné vozidlo, no v prípade prekážky v ceste dokážu zdvihnúť podvozok a pomocou nôh prekážku prekročiť alebo sa otočiť na mieste. Jedná sa ale o roboty so zložitejšími riadiacimi algoritmami a drahšou realizáciou [32].



Obr. 2.9: Model vozidla s kolesovo-nožičkovým systémom, prevzaté z [2]

V tejto kapitole boli predstavené a porovnané rôzne alternatívne podvozkové koncepcie, ktoré sa líšia nielen mechanickým riešením, ale aj spôsobom riadenia a vhodnosťou pre konkrétne aplikácie (stručný prehľad v tabuľke 2.1). Zatiaľ čo diferenčný pohon predstavuje jednoduché a spoľahlivé riešenie pre základné úlohy autonómneho pohybu, holonomné systémy ponúkajú vyššiu manévrovateľnosť na úkor konštrukčnej a softvérovej náročnosti. Pásové a kĺbové podvozky zas vynikajú v špecifických podmienkach, kde sú potrebné schopnosti prekonávať prekážky alebo manévrovať v úzkych priestoroch.

| Koncept | Výhody | Nevýhody |
|-----------------------------|---|---|
| Diferenčný pohon | Jednoduchá konštrukcia, nenáročný softvér, vhodné pre malé modely | Horšia stabilita pri vyššej rýchlosti, menej presné zatáčanie |
| Trojkoľosové vozidlo | Kompaktnosť, úspora materiálu, dobrá obratnosť | Nižšia stabilita, nevhodné pre nerovný terén |
| Mecanum/Všesmerový podvozok | Holonómny pohyb, výborná manévrovateľnosť v malom priestore | Zložitá mechanika a riadenie, vyššie nároky na snímanie a výpočty |
| Pásový podvozok | Vysoká trakcia, schopnosť prekonávať prekážky | Ťažkopádnosť v interiéri, vyššia energetická náročnosť |
| Kĺbový podvozok | Flexibilita pohybu v úzkych priestoroch | Zložitá konštrukcia a riadenie, náročná realizácia v modelovej mierke |
| Hexapod/viacnohý robot | Výborná stabilita, schopnosť chôdze v zložitom teréne, redundancia pohybu | Veľmi zložitá mechanika a riadenie, vyššia spotreba energie |
| Sférický podvozok | Všesmerový pohyb, unikátny dizajn, plynulý chod | Potreba aktívnej stabilizácie, technicky náročný na realizáciu |
| Kolesovo-nožičkové systémy | Kombinácia výhod nôh a kolies, flexibilita pri prekážkach | Vysoká zložitosť, veľké nároky na výpočtový výkon a softvér |

Tabuľka 2.1: Porovnanie výhod a nevýhod alternatívnych podvozkových koncepcií

Pre účely tejto práce zameranej na autonómne modely vozidiel v interiéri je vhodné zvoliť taký koncept, ktorý poskytuje dostatočnú presnosť riadenia, možnosť jednoduchého zostavenia z dostupných stavebníc a zároveň prináša istý stupeň netradičnosti oproti štandardnému štvorkolesovému podvozku. Z tohoto dôvodu bol zvolený dvojkoľosový/diferenčný model vozidla s tretím stabilizačným bodom.

2.3 Použité technológie a stavebnice

Teraz sa pozrieme na technológie a stavebnice, ktoré boli využívané pri vývoji modelu autonómneho vozidla. Všetky súčasti výsledného projektu boli vyberané na základe ich kompatibility, dostupnosti a vhodnosti pre vybraný koncept.

Prehľad dostupných platforiem

Stavebníc a platforiem vhodných na realizáciu modelu autonómneho vozidla je nespočetne veľa. Ponúkajú možnosť zrealizovať nápady začiatočníkov pomocou jednoduchších systémov, ako aj náročné a prepracované projekty skúsených odborníkov. Uvedené budú iba najpopulárnejšie a najvhodnejšie stavebnice, ktoré sú esenciálne pre túto prácu:

- Jednou z najznámejších platforiem je práve **Arduino** vďaka svojmu širokému spektru využitia a ľahko použiteľnému hardvéru a softvéru. Arduino mikrokontrolér je možné ovládať pomocou ich multiplatformového vývojového prostredia a programovacieho jazyka, ktorý je nadstavbou jazyka C/C++ [1]. Ide o open-source platformu, takže

dokumentácia, schémy a zdrojové kódy sú voľne dostupné, čo robí Arduino lákavým či už pre začiatočníkov alebo pokročilých. Umožňuje jednoduché ovládanie senzorov, motora a ďalších periférií. Súčiastky pre zostavenie vozidla či iného zariadenia sú ponúkané v súpravách a taktiež samostatne na ich stránkach [5]. Arduino mikrokontrolér nie je energeticky náročný a je vhodný pre projekty, ktoré vyžadujú pohotovú reakciu. Keďže neobsahuje operačný systém, kód beží priamo na mikrokontroléry, čo umožňuje spracovávať požiadavky v reálnom čase. Neprítomnosť operačného systému ale spôsobuje, že Arduino nie je vhodné pre náročnejšie systémové nastavenia, neumožňuje sieťové funkcie ani paralelné vykonávanie úloh. Nie je tiež určené pre náročné výpočetné úlohy ako spracovanie obrazu kvôli svojmu nízkemu výpočetnému výkonu a limitovanej pamäti [4, 28].

- Ďalšia platforma je **ESP32** od spoločnosti Espressif Systems, ktorá je v porovnaní s klasickým Arduino pokročilejšia. Ich mikrokontrolér ponúka navyše bezdrôtovú komunikáciu a vyšší výpočtový výkon, vďaka čomu je vhodná na projekty, kde sa využíva diaľkové ovládanie alebo diaľková komunikácia. To má ale za následok vyššiu spotrebu energie. Vývoj projektov na tejto platforme je možný pomocou rôznych programovacích prostredí, ako napríklad rámec ESP-IDF, ktorý bol vyvinutý Espressif spoločnosťou, ale podporuje aj Arduino IDE. Medzi programovacie jazyky patrí MicroPython, JavaScript či C/C++. Pre začiatočníkov môže byť táto platforma náročnejšia kvôli zložitejšej konfigurácii. Aj keď dokáže zbierať a prenášať dáta, jeho výpočtový výkon a hardvérové zdroje výrazne limitujú spracovanie obrazu či videa [6, 22].
- Jednou z možností pre začiatočníkov sú výrobky od spoločnosti **Legó**, ktorá ponúka LEGO Mindstorms EV3 (staršia verzia) a SPIKE Prime, ktoré obsahujú programovateľný blok, motory a senzory. Umožňujú vytvárať rôzne robotické modely vďaka vysokej modularite. Programovanie je možné pomocou jazyka Scratch, blokov ikon a slov. Je určené hlavne pre základné školy na výučbu základov robotiky a algoritmického myslenia. Ich grafické programovanie je intuitívne, avšak tieto platformy majú obmedzený výpočtový výkon, ktorý nie je vhodný pre spracovanie obrazu v reálnom čase. Ďalšou nevýhodou je vysoká cena oproti ostatným platformám a limitovaná možnosť pripojenia vlastných komponentov [10, 27, 36].
- Zaujímavá platforma je **NVIDIA Jetson**, ktorá sa zameriava na vývoj aplikácií využívajúcich umelú inteligenciu. Má vysoký výpočtový výkon a umožňuje paralelné spracovanie viacerých senzorov a neurónových sietí. Umelú inteligenciu je možné využiť na spracovanie obrazu v reálnom čase, no má vyššie nároky na napájanie, čo môže byť v prípade mobilných modelov problematické. Medzi nevýhody patrí aj vyššia cena a zložitejšia konfigurácia [8, 9].
- V neposlednom rade medzi platformami na vývoj autonómnych vozidiel je tu **Raspberry Pi**, ktorý je vhodný pre výučbu aj prototypovanie v oblasti robotiky. Je to malý, cenovo dostupný počítač, poskytujúci vysoký výpočtový výkon a pamäť s možnosťou rozšírenia. V porovnaní s mikrokontrolérmi ako Arduino či ESP32 ponúka plnohodnotný operačný systém, vďaka čomu umožňuje spúšťanie komplexného softvéru ako sú Python, OpenCV, alebo knižnice na spracovanie obrazu, neurónových sietí a riadenie senzorov. Nevýhodou môže byť vyššia spotreba energie a môže byť náchyľnejší na zlyhanie pri výpadku napájania [4, 16, 19, 28].

Okrem vyššie uvedených platforiem existuje mnoho ďalších riešení, ktoré možno využiť pri návrhu autonómnych systémov rôznej náročnosti. V praxi sa často využíva aj kombinácia viacerých platforiem, napríklad mikrokontrolér Arduino alebo ESP32 zabezpečuje základné riadenie a zber dát zo sensorov, zatiaľ čo výkonnejšia jednotka ako Raspberry Pi alebo NVIDIA Jetson spracúva obraz či pokročilé výpočty [4, 28]. Výber vhodnej stavebnice preto závisí od konkrétnych požiadaviek projektu, úrovne odborných znalostí používateľa a dostupných zdrojov. Prehľadné porovnanie jednotlivých platforiem pre autonómne vozidlá na základe kľúčových vlastností pre účely tejto práce je uvedený v tabuľke 2.2.

| Platforma | Výpočtový výkon | Podpora spracovania obrazu | Energetická náročnosť | Cena | Zložitosť použitia | Vhodné pre |
|------------------|------------------------|-----------------------------------|------------------------------|------------------|---------------------------|---|
| Arduino | Nízky | Nie | Nízka | Nízka | Jednoduchá | Začiatočníkov, pohotovú reakciu |
| ESP32 | Stredný | Čiastočne | Stredná | Nízka | Stredne zložitá | IoT, bezdrôtové riadenie, malé projekty |
| Raspberry Pi | Stredne vysoký | Áno | Stredná až vyššia | Nízka až stredná | Jednoduchá | Výučba, prototypy, kamerové systémy |
| LEGO Mindstorms | Nízky až stredný | Nie | Nízka | Vysoká | Veľmi jednoduchá | Školské projekty, začiatočníci |
| NVIDIA Jetson | Vysoký | Áno | Vysoká | Vysoká | Zložitá | AI, neurónové siete, pokročilá vizuálna analýza |

Tabuľka 2.2: Porovnanie platforiem pre autonómne vozidlá na základe kľúčových vlastností pre účely tejto práce

Použitá platforma, nástroje a vývojové prostredie

V tejto časti sa bližšie zameriame na konkrétnu platformu, vývojové nástroje a konštrukčné prostriedky, ktoré boli použité pri realizácii modelu autonómneho vozidla. Popíšeme dôvody výberu riadiacej jednotky, materiálu konštrukcie modelu, vývojové prostredie, ako aj softvérové knižnice, ktoré zohrávajú kľúčovú úlohu pri ovládaní komponentov a spracovaní obrazu.

Ako riadiaca jednotka pre realizáciu modelu vozidla bola zvolená Raspberry Pi, konkrétne Raspberry Pi 4 Model B. Dôvodom je jej výpočtový výkon, ktorý je potrebný pre spracovanie obrazu, ako aj modul pre napojenie kamery a GPIO piny, vďaka ktorým môžeme jednoducho pripojiť a ovládať ďalšie potrebné súčasti modelu, ako sú motorčeky. Raspberry Pi je tiež kompaktné (má veľkosť kreditnej karty), takže výsledný model veľmi nezataží. Ďalšou skvelou funkciou je možnosť vzdialeného ovládania pomocou wi-fi signálu, takže pre zasielanie inštrukcií nie je potrebné mikroprocesor pripájať cez kábel. Výhodou pre túto prácu je aj možnosť programovania v jazyku Python, ktorý poskytuje veľké množstvo užitočných knižníc [4, 16, 19, 28].

Pre sfunkčnenie hlavnej časti – Raspberry Pi – bolo potrebné do mikroprocesora zapojiť SD kartu s nainštalovaným operačným systémom. Rozhodla som sa pre odporúčaný Raspberry Pi OS. Po konfigurácii potrebných nastavení je možné sa k tomuto zariadeniu pripojiť buď pomocou siete cez SSH alebo môžeme pristupovať priamo pomocou ostatných periférií. K tomuto prístupu potrebujeme displej, kábel s micro HDMI portom, klávesnicu a myšku.

Po pripojení k Raspberry Pi bolo nutné doinštalovať niekoľko balíčkov a knižníc, ktoré budeme využívať pri ovládaní súčastí modelu vozidla a samotnom algoritme [16, 19, 23].

```
sudo apt update
sudo apt upgrade -y

sudo apt install python3 idle3
sudo apt install -y python3-opencv
sudo apt install -y python3-picamera2

sudo rpi-update
sudo reboot

sudo apt install --reinstall qt5-qmake qtbase5-dev qtchooser
qtbase5-dev-tools qtbase5-dev
sudo apt install --reinstall libqt5gui5 libqt5x11extras5 qtwayland5
sudo apt install qt5-default

sudo add-apt-repository ppa:beineri/opt-qt-5.15.2-focal
sudo apt update
sudo apt upgrade -y

sudo apt install --reinstall libgl1-mesa-dri libegl1-mesa libgles2-mesa
sudo apt install libgl1-mesa-glx libegl1-mesa libgles2-mesa
sudo apt install mesa-utils

sudo apt install xfonts-base xfonts-100dpi xfonts-75dpi
```

```
sudo apt install --reinstall libxcb-xinerama0

sudo apt update
sudo apt upgrade -y

sudo apt install -y opencv-data
```

Výpis 2.1: Použité príkazy na konfiguráciu Raspberry Pi po inštalácii systému

Ako programovací jazyk riadiaceho softvéru bol zvolený Python vďaka jeho jednoduhosti a širokej podpore knižníc [19]:

- **GPIO zero** – táto knižnica slúži ako rozhranie na komunikáciu medzi Raspberry Pi a ostatnými komponentami, ktoré su pripojené cez GPIO piny. V našom prípade ide o ovládanie motorov [26].
- **Picamera2** – určená pre komunikáciu s kamerovým systémom Raspberry Pi, konkrétne pre kameru pripojenú pomocou plochého kábla priamo do konektora na mikroprocesore. Dobre spolupracuje aj s knižnicou OpenCV [23].
- **OpenCV** – open-source knižnica, ktorá obsahuje stovky algoritmov počítačového videnia. V tejto práci je využívaná pre spracovanie obrazu z kamery, ako napríklad detekciu objektov, morfologické operácie, zmenu farebného priestoru. . . [29]
- **Numpy** – poskytuje viacrozmerné polia a rýchle logické a matematické operácie s nimi. Nápomocná bola pri manipulácii s obrazovými dátami a výpočtoch v spojení s OpenCV [29, 12].

Pri vývoji výsledného modelu vozidla bolo nutné použiť viacero vývojových nástrojov. Okrem výberu platformy bolo dôležité rozhodnutie aj to, na čom bude celý model fyzicky postavený. V tejto práci nebola použitá žiadna z bežne dostupných stavebníc, ako napríklad LEGO či podobné hotové riešenia. Hlavným dôvodom tohto rozhodnutia bola potreba väčšej flexibility pri návrhu konštrukcie, ako aj možnosť lepšej optimalizácie rozmerov a tvaru podľa konkrétnych komponentov a požiadaviek modelu.

Použitím 3D modelovania a následnej 3D tlače jednotlivých častí vozidla bolo možné vytvoriť presne takú konštrukciu, aká bola potrebná – napríklad s presnými úchytmi pre Raspberry Pi, motorčeky, batériu či kameru. Tento prístup je prívetivejší v prípade potreby jednoduchých úprav a zmien počas vývoja. Zvolený spôsob zároveň otvára priestor pre využitie vlastnej fantázie a rozvoj technickej tvorivosti, čo vedie k lepšiemu porozumeniu celého procesu návrhu a realizácie modelu.

Webová aplikácia Tinkercad bola výborným nástrojom pri vytváraní 3D modelu. Okrem možnosti modelovania 3D dizajnov a elektrických obvodov ponúka aj blokové programovanie. V tejto práci bolo prínosné jednoduché a intuitívne modelovanie bez potreby pokročilých znalostí, ktoré táto platforma umožňuje. Môžeme tu nájsť aj predpripravené objekty rôznych aj pokročilých tvarov (napríklad kolesá), ktoré si vieme upraviť podľa potreby a ušetriť tak veľa času pri vývoji [3].

Spojením vlastnej konštrukcie vytlačenej na 3D tlačiarňi a výkonného vývojového prostredia Raspberry Pi vzniklo flexibilné riešenie vhodné pre autonómne vozidlo. Tento prístup nielenže umožňuje efektívnu integráciu hardvéru a softvéru, ale zároveň podporuje individuálnu tvorivosť a praktické nadobúdanie zručností v oblasti návrhu, programovania aj fyzickej realizácie.

Kapitola 3

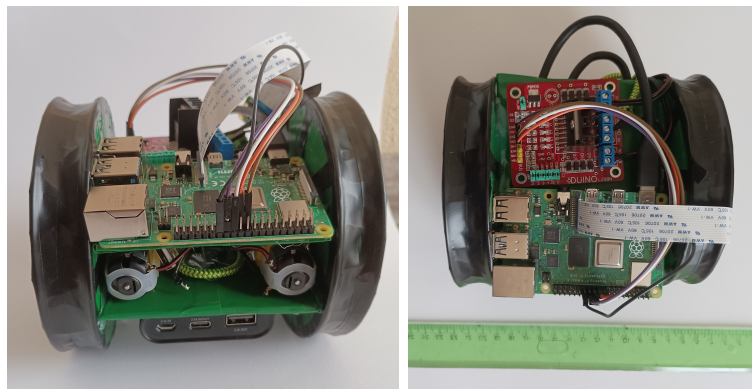
Návrh a realizácia modelu

V tejto kapitole sa pozrieme na praktickú časť práce, a to konkrétne na návrh a realizáciu nekonvenčného modelu vozidla so samoriadiacimi schopnosťami. Zoznámime sa s dvojkolesovou koncepciou z pohľadu mechanickej časti, použitými elektronickými komponentami a riadiacim softvérom.

3.1 Voľba koncepcie vozidla

Na začiatok bolo potrebné vytvoriť dvojkolesový model vozidla, do ktorého by bolo možné efektívne umiestniť všetky komponenty potrebné na fungovanie vozidla. Medzi tieto komponenty patria motorčeky, základná doska Raspberry Pi, H-mostík, kamera a napájací zdroj. Z pohľadu mechaniky modelu predstavujú kolesá jedinú aktívnu mechanickú časť. Zvyšok konštrukcie slúži ako nosná a upevňovacia sústava, ktorá zabezpečuje stabilitu a správne rozloženie jednotlivých častí.

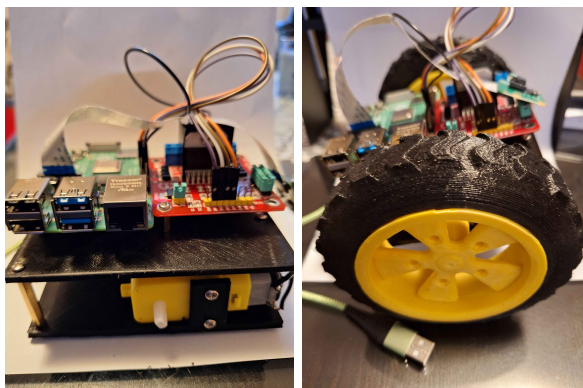
Ako prvý bol vytvorený jednoduchý prototyp z kartónu, ktorý môžete vidieť na obrázku 3.1. Týmto sme získali lepšiu predstavu o rozložení komponentov a celkovom rozmere modelu.



Obr. 3.1: Prototyp prvého modelu vozidla vytvorený z kartónu

Po doladení detailov bol návrh prenesený do prostredia Tinkercad, kde bol prototyp vozidla vymodelovaný a následne vytlačенý na 3D tlačiarňi z materiálu PLA, obrázok 3.2. Základ vozidla tvorili dve rovné dosky, umiestnené nad sebou, ktoré boli prepojené pomocou kovových tyčiek. Cieľom návrhu modelu bolo vytvoriť podvozok tak, aby sa zeme dotýkali

výhradne kolesá, čím sa zníži trenie a zlepši pohyblivosť modelu. Aby sa zabránilo prevráteniu vozidla, bolo potrebné umiestniť jeho ťažisko čo najnižšie. Najprístupnejším riešením bolo umiestnenie najťažšej časti – napájania – na spodnú stranu dolnej dosky. Priestor medzi doskami vyplňali káble a motorčeky, na ktoré boli napojené kolesá pomocou na mieru vytvorených výrezov. Na hornej doske boli umiestnené Raspberry Pi a H-mostík. Kamera bola pripevnená na dosku v tvare štvorca napojeného zvisle na hornú dosku vozidla. Všetky komponenty okrem kolies boli napevno prichytené ku konštrukcii skrutkami.



Obr. 3.2: Prvý model skonštruovaný zo súčiastok vytlačených na 3D tlačiarňi

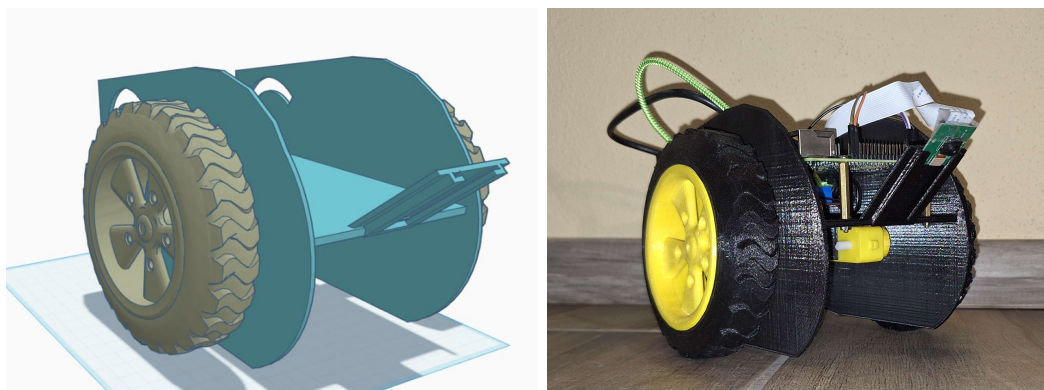
Ďalším krokom pri tvorbe vozidla bolo testovanie kolies. Dôležité bolo správne vybrať ich hrúbku, veľkosť, pružnosť a materiál, obrázok 3.3. Disk kolesa bol zhotovený z PLA materiálu, na pneumatiku bol použitý materiál TPU. Vzhľadom k použitiu rôznych materiálov a tiež dosiahnutia čo najnižšej hmotnosti na jednotlivé komponenty kolesa bolo ďalej potrebné správne nastavenie hustoty výplne pneumatiky. Na 3D tlačiarňi sme nastavili viaceré hustoty výplne a testovali sme ich pružnosť. Najlepšie sa javila 50% hustota výplne, ktorá bola zvolená na zhotovenie výsledného modelu kolesa. Správna veľkosť bola vybratá na základe vykonania testov rôznych rozmerov kolies. Modely s tenkými kolesami boli nestabilné, hrubé kolesá predstavovali veľkú záťaž pre motorčeky a prinášali aj priveľkú zotrvačnosť. Veľkosť kolies bola zvolená primerane k celkovej výške modelu.



Obr. 3.3: Kolesá rôznej veľkosti a hustoty výplne testované pre model vozidla

Pri testovaní snímania obrazu kamerou pripevnenou na zhotovené vozidlo bolo zistené, že model nie je využiteľný pre ciele tejto práce. Vozidlo nebolo možné stabilizovať pri rozbiehaní a spomaľovaní tak, aby kamera zvládala snímať dostatočne kvalitný obraz. Z tohoto dôvodu bolo rozhodnuté vytvoriť nový, vylepšený model 3.4.

Hlavným cieľom druhého návrhu bolo zvýšiť stabilitu vozidla. Body dotýkajúce sa zeme (kolesá) boli preto obohatené o tretí, stabilizačný bod. Ten už však nebol v podobe kolesa ale polguľovitého výstupku umiestneného v zadnej časti vozidla pod priestorom na napájanie. Vďaka stabilizačnému výstupku sa nemusí celá zadná časť modelu dotýkať zeme, čo znižuje trenie a vozidlo tým pádom potrebuje menšiu silu na prekonanie trecieho odporu. Na zvýšenie stability bolo presunuté napájanie modelu do zadnej časti vozidla. Váha modelu vozidla sa teda nachádza v zadnej časti v oblasti výskytu stabilizačného výstupku, tým pádom je zabránené prevráteniu vozidla a akémukoľvek nežiadúcemu pohybu. Doska, na ktorej je umiestnené Raspberry Pi a H-mostík, zostala nezmenená. V prednej časti je napojená plošina na kameru, ktorá je v tomto modeli už naklonená. Vďaka presunu powerbanky mohla byť spodná doska modelu odstránená. Motorčeky boli pripevnené na spodnú stranu dosky. Pridaním tretieho stabilizačného bodu a umiestnením najväčšej váhy do zadnej časti vozidla sme zaistili potrebnú stabilitu pre spoľahlivé snímanie obrazu, čo bolo kľúčové pre ďalší vývoj autonómneho riadiaceho softvéru.



Obr. 3.4: Druhý model vymodelovaný v prostredí Tinkercad a vytlačný na 3D tlačiarni

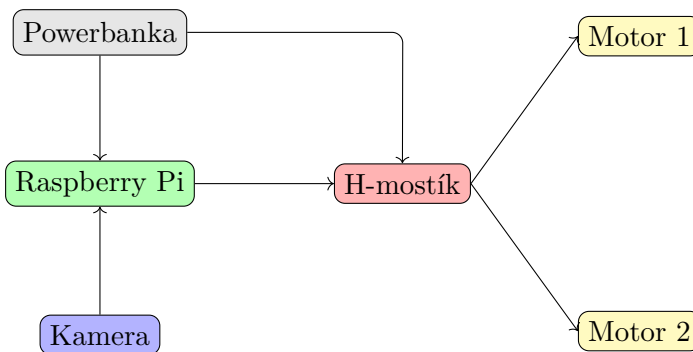
3.2 Elektrické príslušenstvo

Pre funkčný model bolo nevyhnutné vytvoriť zapojenie všetkých elektrických súčiastok, čo v tomto prípade predstavuje:

- **Raspberry Pi 4 Model B** – hlavná riadiaca jednotka,
- **H-mostík** – riadenie motorčekov,
- **DC motorčeky** – pohyb vozidla,
- **Raspberry Pi kamerový modul rev1.3** – snímanie obrazu pre navigáciu,
- **a powerbanka** – jediný zdroj napájania celého systému.

Začala som tým, že som prispájkovala vodiče ku každému motorčeku a tie boli pripojené ku vopred určeným svorkám na H-mostíku. H-mostík je prepojený so zariadením Raspberry Pi pomocou 4 káblikov, kde 2 vodiče slúžia na ovládanie každého motorčeka zvlášť. Na Raspberry Pi doske sú pre tieto účely GPIO piny, ktoré umožňujú softvérové riadenie výstupov. Ďalej sa tu nachádza pre model dôležitý konektor na zapojenie kamery, ktorá bola umiestnená v prednej časti modelu tak, aby mala nerušený výhľad na dráhu

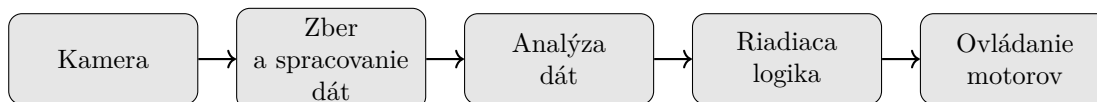
vozidla. Celý systém je napájaný prostredníctvom powerbanky, ktorá je pripojená cez USB port do Raspberry Pi a pomocou upraveného USB kábla aj do H-mostíka. Tento kábel bol prestrihnutý, zbavený nadbytočnej izolácie, na koncoch káblov boli medené časti pocínované a pripojené do svoriek, čím sa zabezpečilo stabilné a nezávislé napájanie motorov bez zataženia napájania z Raspberry Pi. Použitie Powerbanky ako jediného zdroja napájania zabezpečilo mobilitu celého riešenia bez potreby externých zdrojov. Prehľadné zapojenie elektrických komponentov je znázornené na obrázku 3.5.



Obr. 3.5: Schéma zapojenia elektrického príslušenstva modelu vozidla

3.3 Návrh riadiaceho softvéru

Kľúčovou časťou tejto práce je riadiaci softvér vozidla, bez ktorého by bol model vozidla iba kus plastu a elektrických komponentov. Táto kapitola bude preto venovaná podrobnému popisu jednotlivých blokov programu, vrátane integrácie senzorov a výstupných zariadení. Ako programovací jazyk bol vďaka svojej kompatibilitate s Raspberry Pi a veľkým množstvom užitočných knižníc zvolený python. Nižšie je znázornená bloková schéma celého programu na obrázku 3.6.



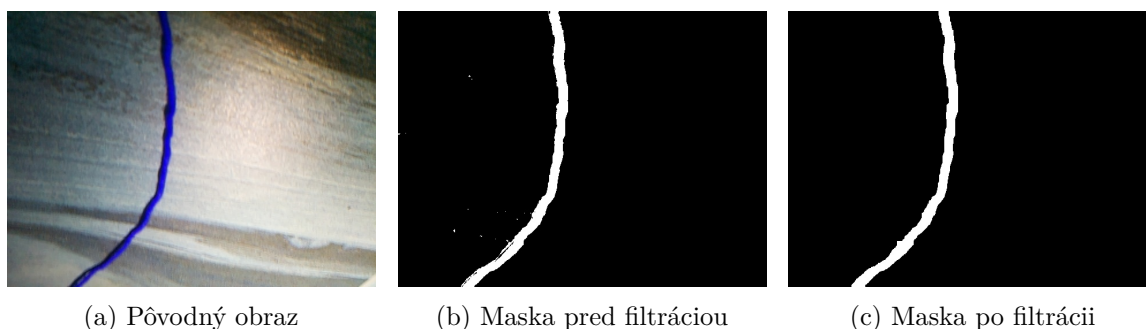
Obr. 3.6: Blokovaná schéma softvéru pre riadenie modelu vozidla

Kamera znázornená v schéme z obrázku 3.6 predstavuje senzor, ktorý zachytáva vizuálne dáta z priestoru pred vozidlom v reálnom čase. V programe je využívaná knižnica `picamera2`, ktorá umožňuje jednoduchú konfiguráciu a ovládanie kamery na platforme Raspberry Pi. Vytvorené snímky sú prevádzané do digitálnej podoby, a to do formátu RGB s rozlíšením 640 pixelov na šírku a 480 pixelov na výšku (toto rozlíšenie je pre účely práce postačujúce a zároveň nadbytočne nezaťažuje výpočtové možnosti Raspberry).

Zber dát je sprostredkovaný zachytávaním snímky pomocou funkcie `capture_array()`, ktorej výstupom je matica pixelov. Tá je ďalej spracovaná pomocou `OpenCV` knižnice, ktorá prevádza obraz do HSV (Hue, Saturation, Value) farebného priestoru, ktorý je výhodnejší pri detekcii farieb. Cieľom tohoto bloku je čo najrýchlejšie zachytiť obraz a pripraviť ho na analýzu.

Vozidlo má byť schopné samoriadenia na základe farebnej značky, ktorú predstavuje čiara alebo bod na zemi, a preto je nutné ju pri analýze zdetegovať. Najskôr je potrebné

definovať si hranice farebného priestoru pomocou trojrozmerných polí za využitia NumPy knižnice, ktorými budú odlišené významné pixely od nevýznamných. Na základe hraníc je vytvorená maska pomocou funkcie `cv2.inRange()`, ktorej výsledkom je binárny obraz so zvýraznením definovaných farieb v podobe bielych pixelov. Zvyšné pixely sú čierne, čo je možné pozorovať na obrázku 3.7b. Nasledujú morfológické úpravy pre odstránenie šumu a malých nedokonalostí. Na tieto účely sa využívajú dve operácie – vyplnenie drobných dier vo vnútri objektov a odstránenie príliš malých objektov. Na realizáciu je opäť využívaná NumPy knižnica, s ktorou je definovaná matica 5x5 – filtračná maska. Tá prechádza po obraze a pomocou funkcie `cv2.morphologyEx()` sa vyhladzuje obraz. Vo funkcii sa definuje filtračná maska a operácia, ktorá sa má použiť. Pre tento prípad boli esenciálne operácie `cv2.MORPH_CLOSE` na vyplnenie malých dier vo vnútri objektov a `cv2.MORPH_OPEN` na odstránenie šumu. Výsledkom je obraz zbavený nedokonalostí, ktoré by mohli rušiť detekciu navigačných značiek, vid' obrázok 3.7c.

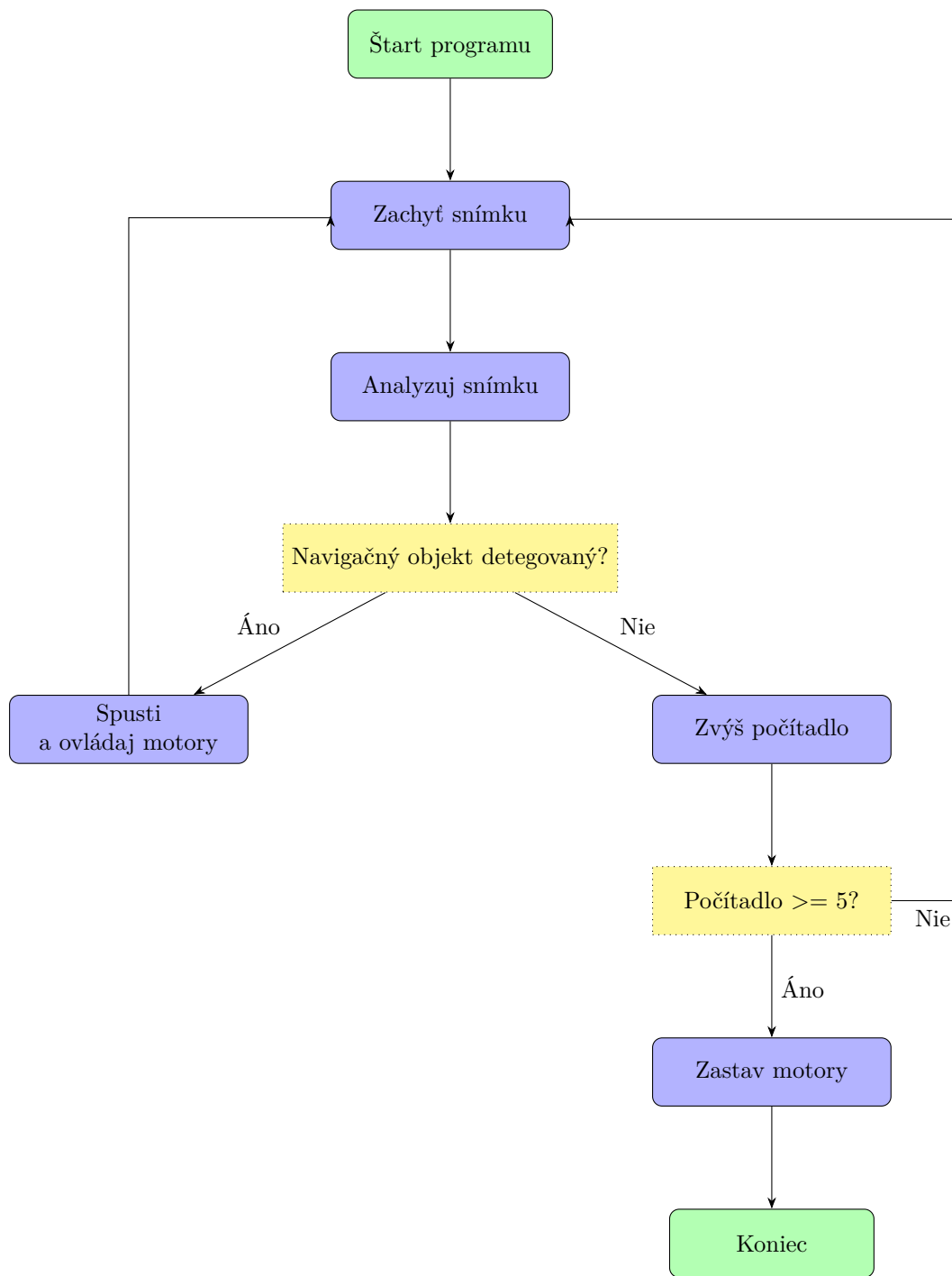


Obr. 3.7: Spracovanie obrazu – detekcia navigačných značiek

Pre ďalšie spracovanie je potrebný zoznam kontúr objektov v binárnej maske, na čo je použitá funkcia `cv2.findContours()`. Ako vstupný obraz je využívaný vyššie popisovaný filtrovaný binárny obraz a pre spracovanie je nastavené zachytávanie vonkajších kontúr, ktoré sú reprezentované ako zoznam bodov (x, y) , a teda dohromady tvoria obvod objektov.

Využitie týchto kontúr sa líši od typu scenára autonómnej jazdy, a preto je spolu s riadiacim algoritmom jednotlivých scenárov podrobne popísaný v ďalšej kapitole.

Ovládanie motorov sa nachádza v hlavnom riadiacom bloku v nekonečnom cykle a prebieha v reálnom čase. Na komunikáciu s výstupnými zariadeniami – motorčekmi – je používaná knižnica `gpiozero`, trieda `Motor`. Umožňuje jednoduché ovládanie cez dvojicu pinov (pohyb dopredu, dozadu). Po získaní obrazu z kamery a jeho analýze nastáva spracovanie výsledkov. Aby boli ošetrené chyby z dôvodu chvílkového výpadku správnej detekcie navigačných značiek, bolo zavedené počítadlo neúspešných detekcií. V prípade, že nebola v obraze nájdená kontúra, zvýši sa. Ak sa toto počítadlo dostane na číslo 5, motory sa zastavia a program ukončí, keďže je vysoko pravdepodobné, že navigačné značky skončili alebo sa od nich vozidlo vychýlilo. Pri úspešnej detekcii sa vynuluje počítadlo neúspešných detekcií a rozbehnú sa motory, ak ešte neboli spustené. Algoritmus je znázornený na obrázku 3.8. Popis korigovania smeru vozidla bude tiež popísaný v ďalšej kapitole na základe scenára.



Obr. 3.8: Vývojový diagram riadiaceho algoritmu

Kapitola 4

Návrh scenárov autonómnej jazdy a riadiacich algoritmov

Nasledujúca kapitola bude venovaná popisu návrhu a implementácie rôznych scenárov autonómnej jazdy. Po detekcii kontúr navigačných objektov v obraze boli navrhnuté viaceré scenáre, ktoré majú rôzny prístup využitia kontúrových súradníc. Prvý scenár bude jednoduchší – sústredí sa na jazdu po čiare. Nasledujúci variant vychádza z prvej a venuje sa vylepšeniu plynulosti pohybu. Na záver bude popísaný modifikovaný scenár, ktorý sa už neriadi plynulou čiarou, ale bude nahradená pokiaľ možno minimálnym počtom bodov s rôznymi rozstupmi. Tie nám umožňujú navigovať vozidlo s použitím menšieho množstva materiálu, ktorý vieme umiestniť len na pár miest, takže celá trajektória nemusí byť plynule označená.

Po podrobnej analýze uvažovaných scenárov som navrhla a implementovala algoritmy, čo najvernejšie k stanoveným návrhom. Počas vývoja došlo k niekoľkým modifikáciám, ktoré vyvrcholili do návrhov v podobe, v akej sú uvedené nižšie.

4.1 Scenár 1 – jazda po čiare

V tomto programe je model vozidla riadený čiarou. Výsledný zoznam súradníc kontúr v obraze z analýzy popísanej vyššie je využívaný na nájdenie najvyššieho bodu kontúry (v našom prípade sa jedná o najnižšiu y-súradnicu), za ktorým by sa vozidlo malo smerovať. Iteruje sa cez všetky body kontúr, porovnáva sa ich výška (y-súradnica) a hľadá sa bod, ktorý leží najvyššie v obraze. Pri úspešnom nájdení tohoto bodu sa ukladá jeho x-súradnica. Tá je potrebná na výpočet odchýlky najvyššieho bodu od vodorovného stredu a normalizáciu tejto odchýlky na hodnotu v rozsahu od -10 do 10. Na škálovanie odchýlky je využitá polovičná šírka obrazu, ktorým je vydelená odchýlka a výsledok je v rozmedzí -1 až 1. Táto hodnota je vynásobená číslom 10, aby bola dosiahnutá požadovaná škála, ako je uvedené v rovnici 4.1.

$$\text{norm_odchylka} = \left(\frac{x_{\text{top}} - x_{\text{stred}}}{\frac{\text{sirka}}{2}} \right) \cdot 10 \quad (4.1)$$

Záporné hodnoty znamenajú, že bod leží naľavo od stredu a kladné, že leží napravo. Výsledok je použitý na ovládanie motorov. Ak je hodnota záporná, znamená to, že sa vozidlo musí natočiť doľava. V praxi je to realizované pozastavením ľavého kolesa na krátky čas, ktorý je definovaný vzorcom 4.2.

$$\text{cas_pozastavenia} = \frac{\text{norm_odchylka}}{40} \quad (4.2)$$

V prípade kladnej hodnoty je pozastavené pravé koleso. Tento výpočet bol určený experimentálnym spôsobom a zvolený na základe najlepších výsledkov pri testovaní. Po natočení vozidla sa opäť vytvára a analyzuje snímka. Ak je nová normalizovaná odchýlka väčšia ako 1, znova prebieha pozastavenie kolies na základe novej hodnoty. Ak je menšia ako 1, model vozidla sa pohybuje vpred.

Tento prístup je limitovaný rýchlosťou výpočtového výkonu Raspberry Pi, ktorý zvláda spracovať jednu snímku v priemere za 0.3 sekundy. Model vozidla teda upravuje svoje smerovanie približne 3-krát za sekundu, čo viditeľne ovplyvňuje plynulosť jazdy. Model sa nakláňa z jednej strany na druhú s cieľom dostať najvyšší bod čiaru do stredu snímaného priestoru. Vďaka svojim mykavým pohybom dostal tento scenár názov *kačacia chôdza*, ktorý veľmi výstižne pomenúva situáciu.

4.2 Scenár 2 – vylepšenie plynulosti jazdy

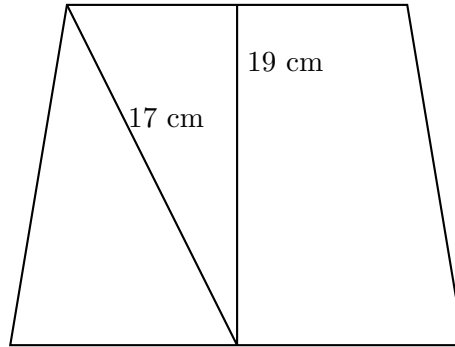
Nasledujúci scenár sa sústreďí na docielenie úbytku trhavého pohybu predošlého programu. Zachováva vyhľadanie najvyššej súradnice čiaru, ako aj výpočet odchýlky od stredu, no pre zlepšenie plynulosti jazdy bolo nutné zmeniť logiku riadiaceho algoritmu. Namiesto prepočítavania smeru vozidla za pochodu v najrýchlejšej možnej miere som sa rozhodla navigovať model vozidla rovnakým smerom najdlhší možný čas. K docieleniu tohoto princípu bolo potrebné vedieť uhol a vzdialenosť, v akej leží najvyššie položený bod značkovej čiaru v aktuálnej snímke, keďže vozidlo je možné s istotou navigovať len po kraj zorného poľa.

Meraním bola zistená vzdialenosť, ktorú vozidlo prejde za sekundu pri pohybe rovno vpred, uhol otočenia vozidla za sekundu, a tiež rozmer zorného poľa kamery, viď obrázok 4.1. Tieto hodnoty boli nevyhnutné pre výpočty časov potrebných na navigáciu vozidla na súradnice najvyššieho viditeľného bodu čiaru¹.

| Namerané hodnoty za sekundu | |
|-----------------------------|---------|
| prejdená vzdialenosť | 32,8 cm |
| otočenie vpravo | 100° |
| otočenie vľavo | 120° |

Tabuľka 4.1: Namerané hodnoty pohybu modelu

¹Rozdiel medzi jednotlivými otočeniami je spôsobený nedokonalosťami kolies modelu a mierne rozdielnym výkonom motorčekov.

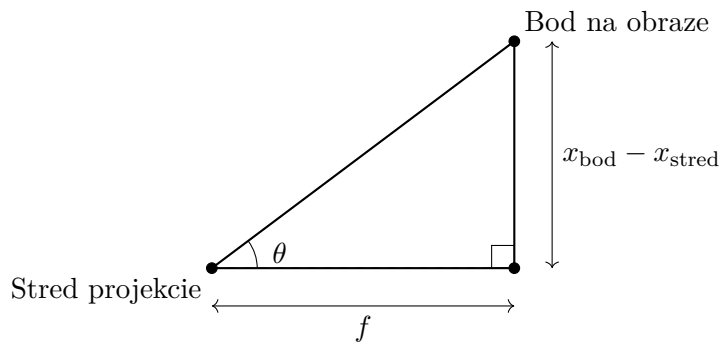


Obr. 4.1: Rozmery zorného poľa kamery pripevnenej na model vozidla

Keď už boli známe potrebné hodnoty, bolo možné spraviť vylepšený algoritmus pre navigáciu po čiare. Najskôr je určený uhol, o ktorý musí byť model natočený aby smeroval na najvzdialenejší bod čiar v aktuálnom zornom poli. Pri výpočte uhlu medzi stredom obrazu a najvyšším bodom je nutné brať do úvahy efekt rybieho oka, čiže skreslenie obrazu. Na tento účel sa využíva známy horizontálny uhol záberu kamery (FOV), ktorý je v prípade kamery Raspberry Pi Modul v1.3 približne 54° [15]. Z tohto uhla sa potom pomocou trigonometrie vypočíta ohnisková vzdialenosť v pixeloch podľa vzorca 4.3.

$$f = \frac{\text{sirka obrazu}}{2 \cdot \tan\left(\frac{\text{FOV}}{2}\right)} \quad (4.3)$$

Táto ohnisková vzdialenosť zodpovedá vzdialenosti medzi stredom projekcie a obrazovou rovinou vyjadrenej v pixeloch (obrázok 4.2). Na jej základe sa následne vypočíta uhol medzi stredom obrazu a zvoleným bodom pomocou arcustangensu, vzorec 4.4 [20]. V implementácii na výpočty slúži knižnica `math`, ktorá má v sebe zabudované funkcie na výpočet uhlu v radiánoch a prevod výsledku na stupne.



$$\theta = \arctan\left(\frac{x_{\text{bod}} - x_{\text{stred}}}{f}\right) \quad (4.4)$$

Obr. 4.2: Výpočet uhla medzi stredom obrazu a nájdeným bodom

Po vypočítaní potrebného uhlu je možné určiť, o koľko sa má model vozidla natočiť, vzorec 4.5. Tu sú využité namerané hodnoty z tabuľky 4.1, aby bolo známe, ako dlho bude trvať otočenie o požadovaný uhol.

$$\text{cas v sekundach} = \frac{|\text{uhol odchylky}|}{\text{uhol otocenia za sekundu}} \quad (4.5)$$

Jednotky výsledku sú adekvátne, keďže na meranie času je používaná knižnica `time`, ktorej parametrom využívaných funkcií je práve čas v sekundách.

Keď už je model vozidla správne natočený, ostáva ešte vypočítať vzdialenosť bodu od kamery. Opäť je využitá normovaná odchýlka bodu od stredu, vzorce 4.6 a 4.7, ktorá je vhodná pri výpočtoch skreslenia.

$$\text{odchylka} = x_{\text{bod}} - x_{\text{stred}} \quad (4.6)$$

$$\text{norm_odchylka} = \frac{|\text{odchylka}|}{\frac{\text{sirka}}{2}} \quad (4.7)$$

Po normalizácii je získaná hodnota v rozsahu 0 až 1, pričom 0 znamená, že bod leží v strede obrazu, a 1 predstavuje bod na okraji. Pri pokračovaní výpočtu, vzorec 4.8, sú zase využité namerané hodnoty, tentokrát hodnoty zorného poľa znázornené na obrázku 4.1. Bod môže byť vzdialený maximálne 19cm v prípade, že je zároveň s kamerou. Ak sa nachádza na okraji, jeho vzdialenosť sa blíži k 17cm. Hodnoty vyplňajúce priestor medzi týmito extrémami nie sú známe a ďalšie meranie by bolo nepresné z dôvodu skreslenia. V takýchto prípadoch je vhodné využiť interpoláciu [18]. Čím bližšie ku stredu bod leží, tým väčšiu váhu bude mať hodnota 19. Čím viac sa bod približuje ku kraju, tým viac sa výsledok bude blížiť k 17. Nasledujúci vzorec (4.8) zobrazuje zohľadnenie váhy extrémov na základe normovanej odchýlky:

$$\text{int_vzdialenost} = (1 - \text{norm_odchylka}) \cdot 19 + \text{norm_odchylka} \cdot 17 \quad (4.8)$$

Nakoniec sa vzdialenosť upraví na základe vertikálnej pozície bodu. Znormalizuje sa výška bodu a bude odrátaná od 1, čím sa získa normalizovaná vzdialenosť, kde 1 predstavuje najvyšší bod obrazu a 0 najnižší. Týmto medzivýsledkom sa vynásobí odhadovaná vzdialenosť z predchádzajúceho výpočtu a zohľadní sa tým vertikálna pozícia bodu, vzorec 4.9.

$$\text{final_vzdialenost} = \text{int_vzdialenost} \cdot \left(1 - \frac{y_{\text{bod}}}{\text{vyska}}\right) \quad (4.9)$$

Konečný výsledok predstavuje približnú vzdialenosť v centimetroch, ktorú musí model vozidla prejsť, aby sa dostal na najvyšší bod čiary zorného poľa kamery.

Ostáva už iba vypočítať čas v sekundách, za ktorý model túto vzdialenosť prejde, vzorec 4.10. Toto je realizované vydelením vypočítanej vzdialenosti bodu nameranou vzdialenosťou, ktorú model prejde za sekundu, viď tabuľka 4.1. Výsledok slúži na reguláciu pohybu motorčekov.

$$\text{cas} = \frac{\text{final_vzdialenost}}{\text{vzdialenost_cm/s}} \quad (4.10)$$

Keď sa model vozidla dostane na požadovaný bod, prebieha výpočet od začiatku. Týmto prístupom je docielený plynulejší pohyb vozidla po čiare, ktorý je stále dostatočne presný.

4.3 Scenár 3 – jazda po značkách

V tomto scenári je model vozidla navigovaný pomocou samostatných bodov – značiek, ktoré ležia v rôznych rozstupoch na zemi. Podobne ako v predchádzajúcich scenároch sa začína iteráciou zoznamu kontúr a hľadaním prvého zhluk bodov. Najskôr je tento zhluk obalený najmenším možným kruhom pomocou funkcie `cv2.minEnclosingCircle` a je overené, či je priemer väčší ako 5 pixelov (inak by mohlo ísť iba o šum). Ak podmienku splní, je uložená x-súradnica stredu kruhu. Do zoznamu kontúr sa kontúry ukladajú podľa polohy na obraze, a to zhora dole a zľava doprava. Stačí teda nájsť prvú vyhovujúcu kontúru, keďže sa bude rovnať najvyššie položenej značke. Súradnica nájdenej značky je opäť znormalizovaná na hodnoty -1 až 1 a vynásobená číslom 10 tak, ako to bolo v prvom scenári, vzorec 4.11.

$$\text{norm_odchylka} = \left(\frac{x_{\text{zhluk}} - x_{\text{stred}}}{\frac{\text{sirka}}{2}} \right) \cdot 10 \quad (4.11)$$

Výsledok je využitý na riadenie motorov. Ak je hodnota v rozmedzí -1 až 1, pokračuje sa rovno. Ak je menšia, zrýchli sa pravý motor a model vozidla sa otáča doľava. Ak je hodnota väčšia ako 1, otočí sa doprava. Následne ide celý scenár odznova a nové smerovanie sa prepočítava v maximálnej možnej miere.

Svojimi prvkami sa nápadne podobá na prvý scenár, no v tomto prípade trhavý pohyb nerobí problém, keďže značenie pomocou bodov smer nemení tak rýchlo, ako čiara.

Kapitola 5

Experimentálna časť a testovanie

Po navrhnutí a implementácii riadiacich algoritmov autonómnej jazdy bolo potrebné otestovať návrhy v praxi. Sústredila som sa na spoľahlivosť, plynulosť a schopnosť zvládať rôzne podmienky prostredia. Testovanie bolo zamerané na overenie funkčnosti implementácií, odhalenie ich silných stránok a nedokonalostí.

5.1 Definovanie parametrov testovania

Overovanie požadovaných vlastností bolo vykonávané nasledovne:

- **Spôľahlivosť** sa testovala pomocou rôzne náročných tratí, cez ktoré mal model vozidla prejsť, a rôznych rýchlostí pohybu modelu (boli testované iba vyššie rýchlosti, pri nižších vozidlo nie sú motorčeky schopné otáčať kolesá). Pri vyšších rýchlostiach bol kladený dôraz na schopnosť vozidla správne reagovať na zmeny v trati bez výrazného vybočenia. Začínalo sa rovnou trasou, ku ktorej sa postupne pridávali stále viac a viac strmé zákruty. V prípade, že model vozidla úspešne zvládol aspoň 90° zákrutu a dostal sa na koniec trasy, bola implementácia považovaná za spoľahlivú.
- **Plynulosť** bola hodnotená počtom zastavení a výkyvov v rámci určenej trasy. Čím väčšie hodnoty boli namerané, tým horšia bola plynulosť vozidla.
- **Rýchlosť** bola stopovaná na kratšej trase dlhej 1m. V jednom testovacom scenári išlo o rovnú trajektóriu a v druhom sa na trase nachádzala zákruta.
- **Schopnosť zvládať rôzne podmienky prostredia**, konkrétne sa jednalo o zvládanie rôznych svetelných podmienok, čiže ako budú ovplyvnené navigačné schopnosti počas zmeny svetla. Do testovacej množiny bolo zahrnuté denné svetlo, šero a umelé svetlo.

Tieto vlastnosti najviac ovplyvňujú výkon navrhnutých scenárov a vďaka ich otestovaniu vieme stanoviť hranice, za akých je model vozidla schopný správne fungovať.

5.2 Analýza výsledkov

Na základe vykonaného testovania boli zozbierané dáta o správaní modelu vozidla v rôznych podmienkach. V tejto časti budú prezentované výsledky meraní formou tabuliek a následne hodnotené pomocou porovnávacej metódy. Výsledné hodnoty meraní sú zoskupené a vyhodnotené najskôr zvlášť pre každý scenár z kapitoly 4 a na záver porovnané medzi sebou.

Scenár 1 – jazda po čiare

Ako prvá bola testovaná spoľahlivosť implementovanej metódy. Najskôr bola meraná vzdialenosť, ktorú zvládne vozidlo prejsť bez vybočenia po rovnej čiare. Model vozidla riadený prvým scenárom prešiel v priemere 1.4 metra pri prednastavenej rýchlosti. Počas chôdze bol nestabilný s neustálym nakláňaním sa z jednej strany na druhú, až kým nevybočil z čiary úplne. Výborné výsledky ale vykazoval pri prudkých zákrutách, kde zvládol prejsť zákrutou s uhlom až 120°, tabuľka 5.1. Výsledky sa nezhoršili ani pri zrýchlení o 10% pôvodnej rýchlosti, tabuľka 5.2. Pri zrýchlení o 20%, čo predstavuje maximálnu rýchlosť motorčekov, už bolo pozorované menšie zhoršenie schopnosti reagovať na prudké zákruty, tabuľka 5.3.

| Pôvodná rýchlosť | |
|------------------|----------|
| Zákruta | Výsledok |
| 0° | ✓ |
| 30° | ✓ |
| 60° | ✓ |
| 90° | ✓ |
| 120° | ✓ |
| 150° | ✗ |

Tabuľka 5.1: Zvládnuté zákruty pri prednastavenej rýchlosti prvého scenára

| Rýchlosť + 10% | |
|----------------|----------|
| Zákruta | Výsledok |
| 0° | ✓ |
| 30° | ✓ |
| 60° | ✓ |
| 90° | ✓ |
| 120° | ✓ |
| 150° | ✗ |

Tabuľka 5.2: Zvládnuté zákruty pri zvýšenej rýchlosti prvého scenára

| Rýchlosť + 20% | |
|----------------|----------|
| Zákruta | Výsledok |
| 0° | ✓ |
| 30° | ✓ |
| 60° | ✓ |
| 90° | ✓ |
| 120° | ✗ |
| 150° | ✗ |

Tabuľka 5.3: Zvládnuté zákruty pri maximálnej rýchlosti prvého scenára

Výsledné hodnoty boli priaznivé aj pri maximálnej rýchlosti, takže túto implementáciu môžeme považovať za spoľahlivú.

Plynulosť už také priaznivé výsledky nemala. Na trase dlhej 1 meter, došlo k výkyvom až 8-krát v priemere, čo predstavuje výkyv každých 12,5 centimetrov trasy. Dôvodom častých výkyvov je spôsob navigácie vozidla, ktorý prepočítava nové súradnice navigačného bodu v najčastejšej možnej miere.

Nameraná rýchlosť po rovnej trase bola priemerne 4,6 sekundy na trase dlhej 1 meter. V prípade, že rovnako dlhá trasa obsahovala zákrutu, sa čas zvýšil na 6,4 sekundy.

Ďalej bola testovaná citlivosť na svetlo, čiže ako dobre vozidlo zvláda rôzne svetelné podmienky. Začali sme denným svetlom, kde nebol pozorovaný žiadny problém pri detekcii navigačných značiek. S ubúdajúcim svetlom klesala spoľahlivosť implementovaného algoritmu. Pri testovaní za šera model nevedel správne reagovať pri prudších zákrutách a častejšie vyšiel z dráhy. Pri umelom osvetlení so žltými podtónmi bolo vozidlo zmätené – vybáčalo z trasy a videlo farbu čiary aj tam, kde nebola. Naopak pri svetle s bielym podtónom bol model vozidla schopný nasledovať čiaru rovnako dobre ako počas denného svetla. Tento rozdiel spoľahlivosti pri rôznych teplotách svetla je spôsobený tým, že žlté svetlo mení odtieň navigačných značiek. V tomto scenári ide o červenú farbu, ktorá v kombinácii so žltou mení odtieň do oranžova. Ľahko sa tak môže stať, že takto zmenená farba už nepatrí do farebného spektra definovaného v implementácii. To isté platí pri šere, pretože nedostatok svetla farbu stmavuje.

Scenár 2 – vylepšenie plynulosti jazdy

Testovanie druhého scenára prebiehalo podobne. Model vozidla musel prejsť po dlhej čiare bez vychýlenia z trasy, čo bolo s týmto algoritmom úspešné. Ďalej musel prekonať rovnaké

zákruty ako počas testovanie prvého scenára. Zákruty sa pohybovali v rozmedzí od 30° do 150°. Najnáročnejšia zákruta, ktorú bol tento model schopný prekonať, bola 90°, čo je stále dostačujúce, takže model je možné považovať za spoľahlivý, tabuľka 5.4. Nebolo možné otestovať väčšie rýchlosti, keďže je program závislý na nameraných hodnotách rýchlosti pohybu, ktoré boli merané pri prednastavenej rýchlosti.

| Prednastavená rýchlosť | |
|------------------------|----------|
| Zákruta | Výsledok |
| 0° | ✓ |
| 30° | ✓ |
| 60° | ✓ |
| 90° | ✓ |
| 120° | ✗ |
| 150° | ✗ |

Tabuľka 5.4: Zvládnuté zákruty pri prednastavenej rýchlosti druhého scenára

Plynulosť sa pohybovala do 3 výkyvov/pozastavení počas trasy dlhej 1 meter, čo predstavuje výkyv maximálne každých 33 centimetrov. Táto implementácia teda splnila úlohu a predstavuje plynulejší prístup pri navigácii po čiare v porovnaní s prvým scenárom.

Pri meraní rýchlosti sme sa dostali na číslo 5,2 sekundy na rovnej čiare dlhej 1 meter a 6.15 sekundy pri zákrute. Vyššie hodnoty sú následkom času potrebného na výpočet vzdialenosti čiastkových destinácií a tiež časom stráveným otáčaním vozidla správnym smerom, ktoré sa nedeje počas jazdy, ale až po príchode na čiastkovú destináciu, zhotovení snímky a vypočítaní uhla.

Keďže je tento scenár vylepšenou verziou prvého, navigačné značenie a spektrum farieb značenia v implementácii ostáva rovnaké. Tým pádom bolo pozorované rovnaké správanie na svetelné podmienky ako pri scenári číslo 1.

Scenár 3 – jazda po značkách

Tretia implementácia prešla testom schopnosti modelu vozidla ísť po rovnej trase bez akýchkoľvek komplikácií. Ak sa však na trase nachádzala zákruta, už to nebolo také jednoznačné. Pri zákrute väčšej ako 60° model vybočil z trasy, preto tento scenár nie je možné považovať za najspoľahlivejší, viď tabuľka 5.5.

Z hľadiska implementácie je vybočenie pri 90° spôsobené logikou riadenia, ktorá vraví, že nasledujeme najvyššie položený bod najbližšie k ľavému boku. Ak sa teda pokúsime o zatočenie doprava o 90°, pred modelom vozidla sa naskytne obraz dvoch bodov v rovnakej výške a ako prvý bude zdetegovaný bod naľavo, takže bude nasledovať práve ten a z trajektórie vybočí. V prípade, že sa bod nachádza napravo od vozidla, má model snahu otočiť sa, ale z dôvodu veľkého nadbiehania sa z trajektórie vychýli.

Zvýšenie rýchlosti o 10% pôvodnej rýchlosti na spoľahlivosť nemala vplyv, ako môžete vidieť v tabuľke 5.6.

| Prednastavená rýchlosť | |
|------------------------|----------|
| Zákruta | Výsledok |
| 0° | ✓ |
| 30° | ✓ |
| 60° | ✓ |
| 90° | ✗ |
| 120° | ✗ |
| 150° | ✗ |

Tabuľka 5.5: Zvládnuté zákruty pri prednastavenej rýchlosti tretieho scenára

| Rýchlosť + 10% | |
|----------------|----------|
| Zákruta | Výsledok |
| 0° | ✓ |
| 30° | ✓ |
| 60° | ✓ |
| 90° | ✗ |
| 120° | ✗ |
| 150° | ✗ |

Tabuľka 5.6: Zvládnuté zákruty pri maximálnej rýchlosti tretieho scenára

Pri tomto scenári sa navigačné body menia s nízkou frekvenciou. Počas merania plynulosti boli zaznamenané výkyvy veľmi ojedinelé, a preto môže byť táto jazda považovaná za veľmi plynulú.

Prechod rovnou čiarou dlhou 1 meter zvládol model vozidla za 5,3 sekundy a trasu so zákrutou za 5.7 sekundy. Rýchle časy sú spôsobené tým, že v tejto implementácii nezastavujeme motorčeky, ale v prípade potreby natočenia modelu vozidla jeden motorček zrýchlime.

Počas denného svetla neboli zaznamenané žiadne ťažkosti, no počas šera mal model vozidla menšie problémy pri zákrutách. Pri umelom osvetlení žltým svetlom však ukazoval rovnako dobré výsledky ako počas denného svetla. Naopak biele svetlo mu robilo problémy v podobe častého vyháčania z trasy. Dôvod tohoto správania môže byť zvolenou farbou navigačných značiek. V tomto scenári sa využíva zelená farba, ktorej žlté svetlo až tak neprekáža – zvýrazňuje ju, zatiaľ čo biele svetlo zelenú príliš zosvetľuje. Tým pádom sa opäť môže stať, že takto nasvietená farba nepatrí do definovaného farebného priestoru v implementácii.

5.3 Zhodnotenie a diskusia

Po získaní všetkých potrebných dát môžeme zhodnotiť dosiahnuté výsledky a porovnať jednotlivé scenáre medzi sebou 5.7.

Spolahlivosť, alebo inak povedané schopnosť prekonávať prudké zákruty, bola najlepšia pri prvom scenári. Svojimi výsledkami vyniká aj počas zvýšenej rýchlosti vďaka svojej častej korektúre smeru. Naopak slabšia spoľahlivosť bola pozorovaná v treťom scenári kvôli odlišnému prístupu detekcie navigačných značiek.

Miesta si ale vymenili pri hodnotení plynulosti. Tu ukazuje najlepšie výsledky tretí scenár a najhoršie prvý. Z toho vyplýva, že čím väčšia je spoľahlivosť, tým horšia je plynulosť jazdy.

Za najrýchlejší je považovaný tretí scenár, ktorý zvládol rovnú aj zakryvenú trasu v podobnom čase. Rýchlosť však môže negatívne ovplyvniť spoľahlivosť, čo sa nám v tomto prípade potvrdilo.

Farebná citlivosť bola vo všetkých scenároch podobná, no v závislosti od farby značenia sa menila citlivosť na rôzne odtiene svetla.

| | Spôľahlivosť | Plynulosť | Rýchlosť | Citlivosť na svetlo |
|----------|--------------|-------------|-----------|---------------------|
| Scenár 1 | dobrá | slabá | priemerná | stredná |
| Scenár 2 | dobrá | priemerná | priemerná | stredná |
| Scenár 3 | slabá | veľmi dobrá | rýchla | stredná |

Tabuľka 5.7: Porovnanie scenárov podľa kľúčových vlastností

Na základe tabuľky vieme povedať, že najlepšie výsledky má scenár 2 a 3. Silnou stránkou tretieho scenára je práve jeho plynulosť. Nezaostáva ani v rýchlosti, no tá môže mať nepriaznivý dopad na spoľahlivosť. Druhý scenár sa nachádza zväčša v priemerných hodnotách, silnou stránkou je jeho spoľahlivosť. Tá je výborná aj v prvom scenári, kde je zase slabou stránkou plynulosť.

Citlivosť na svetlo je spôsobená manuálnym nastavovaním farebného priestoru navigačných značiek v implementáciách. Model vozidla tak nevie reagovať na iné svetelné podmienky, ktoré majú za následok zmeny farby značenia.

Tomuto by sa dalo predísť implementáciou dynamickej adaptácie na svetelné podmienky.

Aj keď bolo testovanie vykonávané precízne so snahou dosiahnuť čo najrealistickejšie výsledky, isté faktory nebolo možné do testovania zahrnúť, kvôli čomu môžu byť výsledky nepresné:

- **Svetelné podmienky** – testovanie bolo vykonávané iba vo vnútorných priestoroch so stabilným osvetlením. Nie je preto možné garantovať, že model vozidla by sa rovnako správal aj počas meniacich sa svetelných podmienok, ako napríklad vplyvom svetiel okoloidúcich áut alebo zmenou počasia.
- **Rýchlosť** – testovanie prebiehalo vždy za konštantnej rýchlosti, ktorá bola prednastavená, a nevieme preto povedať, ako spoľahlivé by jednotlivé scenáre boli pri meniacich sa rýchlostiach.
- **Technické obmedzenia** – výsledky sú závislé od presnosti kamery, nastaveného farebného priestoru pre rozpoznávanie navigačných značiek, oneskorenia spracovania obrazu a rýchlosti otáčania motorčekov. Nepresnosť a obmedzenia vymenovaných častí mohli ovplyvniť plynulosť a spoľahlivosť jazdy.

Výsledný projekt má potenciál využitia vo viacerých oblastiach v bežnom živote. Roboti poháňaní týmito implementáciami by mohli slúžiť ako kuriéri menších zásielok na krátke vzdialenosti, kde by boli navigovaní značkami bez potreby neustáleho ľudského dohľadu.

Rovnaký princíp sa dá uplatniť v skladoch alebo v nemocničnom prostredí, kde by mohli byť týmto spôsobom prepravované materiály a lieky medzi stanovišťami so stabilnou trasou, čo by bolo výhodné pri implementácii.

Využitie nájde aj v zábavnom priemysle a v službách, kde by sa dal využiť v zábavných parkoch ako jedna z atrakcií, napríklad ako vyhliadkové jazdy po parku. Významne by sa dal uplatniť aj na prezentáciu výrobkov po vyznačených trasách pri rôznych výstavách alebo na prezentáciách nových produktov rôznorodých firiem.

V potravinárskom priemysle sa dá výsledný model využiť na donášku jedál v reštauráciách, kde by slúžil namiesto čašníkov a po zaplatení a príprave jedla v kuchyni by sa jedlo naložilo na vozidlo a to by ho donieslo k cieľovému stolu s hosťami.

Jedným z neposledných možností uplatnenia je vo vzdelávacej oblasti. Model vozidla spolu s algoritmi sú vhodné na učenie základných princípov riadenia, spracovania obrazu a prepojenia vstupno-výstupných zariadení. Je to skvelý spôsob, ako ukázať prepojenie

implementácie s hmotným výstupom. Využiť by sa teda dal na školách v rámci výučby informatiky, ale aj na krúžkoch venovaných programovaniu a robotike na základných a stredných školách.

Výsledky tohoto projektu ukazujú, že aj jednoduchý riadiaci algoritmus môže mať pri správnom prístupe dostačujúce výsledky. Využitie týchto algoritmov v praxi dáva projektu priestor na ďalšie vylepšenie a smerovanie na základe špecifických potrieb prostredí.

Kapitola 6

Záver

Cieľom tejto práce bolo navrhnuť a zrealizovať jednoduché autonómne vozidlo postavené na platforme Raspberry Pi, ktoré sa dokáže pohybovať v prostredí na základe vizuálnych podnetov. V rámci riešenia boli implementované a otestované tri rôzne prístupy k riadeniu pohybu – priebežné vyhodnocovanie polohy sledovanej čiary, smerovanie k najvzdialenejšiemu detegovanému bodu a sledovanie sekvencie farebných značiek.

Prvý prístup, založený na priebežnom vyhodnocovaní aktuálnej polohy vizuálnej stopy a následnej úprave výkonu motorov, umožnil základné sledovanie dráhy. Tento spôsob sa však v praxi prejavil ako menej stabilný, čo viedlo ku kolísavej jazde pripomínajúcej pohyb kačky. Zmena smeru bola často oneskorená, čo spôsobovalo nadmerné kormidlovanie a stratovú korekciu.

Druhý prístup, v ktorom sa analyzovala celá viditeľná dráha a určoval sa najvzdialenejší bod na čiare, smerom ku ktorému sa vozidlo otáčalo, viedol k výrazne plynulejšej a rýchlejšej jazde. Tento model vykazoval vyššiu presnosť najmä na priamych a mierne zakrivených úsekoch. Napriek tomu mal problémy v situáciách, kde sa dráha lámala pod príliš ostrým uhlom – vozidlo nedokázalo včas identifikovať nový smer a často vybočilo z dráhy, čo viedlo k jeho následnému zastaveniu.

Tretí model, zameraný na sledovanie zelených papierikov poukladaných v línii, bol spoľahlivý, pokiaľ boli dodržané vhodné svetelné podmienky. Systém sa spoliehal na presnú detekciu farby, a preto bol citlivý na odchýlky spôsobené napríklad umelým alebo slabým osvetlením. Pri správnom osvetlení však dokázalo vozidlo konzistentne sledovať trasu bez výraznejších odchýlok.

Na základe testovania je možné konštatovať, že každý z prístupov má svoje výhody a obmedzenia. Prvá metóda je jednoduchá na implementáciu, no trpí na stabilitu. Druhá prináša lepšiu efektivitu a plynulosť, ale vyžaduje kvalitné plánovanie dráhy bez prudkých zmien smeru. Tretia je veľmi plynulá a rýchla, avšak citlivá na vonkajšie vplyvy. Výsledky potvrdzujú, že aj pri použití jednoduchého hardvéru, ako je Raspberry Pi, je možné realizovať rôznorodé prístupy k autonómnemu riadeniu, pričom ich výber by mal závisieť najmä od konkrétneho scenára nasadenia a požiadaviek na presnosť, rýchlosť a prispôbivosť voči prostrediu.

Literatúra

- [1] ARDUINO. *What is Arduino?* online. 2018. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction/>. [cit. 2025-04-28].
- [2] ATHENS, N. T. U. of. *Hybrid Legged/Wheeled* online. Dostupné z: <https://nereus.mech.ntua.gr/legged/laelaps-1/>. [cit. 2025-04-30].
- [3] AUTODESK. *All you need is a 'what if...'* online. Dostupné z: <https://www.tinkercad.com/>. [cit. 2025-03-14].
- [4] BEN. *Raspberry Pi and Arduino - Combining the Best of Both Worlds* online. 2023. Dostupné z: <https://fleetstack.io/blog/raspberry-pi-and-arduino-combining-the-best-of-both-worlds>. [cit. 2025-04-28].
- [5] BLUM, J. *Exploring Arduino®: Tools and Techniques for Engineering Wizardry*. John Wiley & Sons, Inc, 2013. ISBN 978-1-118-54936-0.
- [6] BURKET, J. *OpenCV for Esp32*. 2020. Dostupné z: <https://github.com/joachimBurket/esp32-opencv?tab=readme-ov-file>. GitHub repository.
- [7] CALDERON, O. J. L. *Design of a Hexapod Robot*. Prague, 2021. Bachelor Work. Czech Technical University in Prague.
- [8] CORPORATION, N. *Get Your AI in Gear with a JetBot AI Robot Kit* online. Dostupné z: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/>. [cit. 2025-04-28].
- [9] CORPORATION, N. *NVIDIA Jetson Nano* online. Dostupné z: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/>. [cit. 2025-04-28].
- [10] DAMOUSIS, Y. *A Lego EV3 drone for surveillance and security* online. 2016. Dostupné z: <https://www.linkedin.com/pulse/lego-ev3-drone-surveillance-security-part-3-yannis-damousis-ph-d/>. [cit. 2025-04-30].
- [11] DELROBAEI, M. a MCISAAC, K. Design and Steering Control of a Center-Articulated Mobile Robot Module. *Journal of Robotics*, Január 2011, zv. 2011.
- [12] DEVELOPERS, N. *NumPy reference* online. 2024. Dostupné z: <https://numpy.org/doc/2.2/reference/index.html>. [cit. 2025-04-28].
- [13] DICK SWAN, R. T. A Look at Holonomic Locomotion. *SERVO Magazine*, September 2013.

- [14] DONG, P.; WANG, X.; XING, H.; LIU, Y. a ZHANG, M. Design and control of a tracked robot for search and rescue in nuclear power plant. In: August 2016, s. 330–335.
- [15] FOUNDATION, R. P. *About the Camera Modules* online. Dostupné z: <https://www.raspberrypi.com/documentation/accessories/camera.html#about-the-camera-modules>. [cit. 2025-05-08].
- [16] FOUNDATION, R. P. *Raspberry Pi Documentation* online. Dostupné z: <https://www.raspberrypi.com/documentation/>. [cit. 2025-03-14].
- [17] FOXTECH. *TS5.0 100kg All Terrain High Speed Rubber Steel Crawler Tank Tracked Undercarriage Unmanned Vehicle Robot Chassis Platform* online. Dostupné z: <https://www.foxtechrobot.com/ts5-0-100kg-all-terrain-high-speed-rubber-steel-crawler-tank-tracked-undercarriage-unmanned-vehicle-robot-chassis-platform.html>. [cit. 2025-04-30].
- [18] GIBOVÁ, Z. *Fyzikálne meranie*. 1. vyd. Technická univerzita v Košiciach, 2019. ISBN 978-80-553-3318-2.
- [19] HALFACREE, G. *THE OFFICIAL Raspberry Pi Beginner's Guide: How to use your new computer*. 4. vyd. Raspberry Pi Trading Ltd, 2020. ISBN 978-1-912047-73-4.
- [20] HOFFMANN, J. Trigonometrická analýza súosového stereoskopického kamerového systému. *Posterus*, 2014, zv. 7, č. 4. ISSN 1338-0087. Dostupné z: <https://www.posterus.sk/?p=16900>.
- [21] KARAVAEV, Y. Spherical Robots: An Up-to-Date Overview of Designs and Features. *Nelineinaya Dinamika*, Január 2023, zv. 18.
- [22] KOLBAN, N. *Kolban's Book on ESP32* online. 2018.
- [23] LTD, R. P. *The Picamera2 Library*. Raspberry Pi Ltd, 2022. Dostupné z: <https://datasheets.raspberrypi.com/camera/picamera2-manual.pdf>. Build-version: 60bfb8f-dirty.
- [24] LTD, T. *What is an Autonomous Vehicle?* online. Dostupné z: <https://www.twi-global.com/technical-knowledge/faqs/what-is-an-autonomous-vehicle>. [cit. 2025-04-28].
- [25] MOBILEYE. *A Brief History of Autonomous Vehicles – from Renaissance to Reality* online. 2023. Dostupné z: <https://www.mobileye.com/blog/history-autonomous-vehicles-renaissance-to-reality/>. [cit. 2025-04-28].
- [26] NUTTALL, B. *GPIO zero* online. 2015. Dostupné z: <https://gpiozero.readthedocs.io/en/latest/>. [cit. 2025-03-14].
- [27] OLA SYROCKA, D. S. *LEGO Mindstorms EV3 Review: Comparing Home and Education* online. 2018. Dostupné z: <https://www.robocamp.eu/en/blog/lego-mindstorms-ev3-review/#programming>. [cit. 2025-04-28].

- [28] OLIYNYK, K. *Difference Between Arduino and Raspberry Pi* online. 2025. Dostupné z: <https://webbylab.com/blog/arduino-vs-raspberry-pi-comparison/>. [cit. 2025-05-02].
- [29] OPENCV. *Image Processing in OpenCV* online. 2025. Dostupné z: https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html. [cit. 2025-05-02].
- [30] P., M. *RC FPV-Trike with rear steering wheel* online. 2019. Dostupné z: <https://www.printables.com/model/3461-rc-fpv-trike-with-rear-steering-wheel>. [cit. 2025-04-30].
- [31] PUCHER, F. *DiffBot Differential Drive Mobile Robot* online. Dostupné z: <https://ros-mobile-robots.com/components/>. [cit. 2025-04-30].
- [32] SHI, Y.; ZHANG, M.; LI, M. a ZHANG, X. Design and Analysis of a Wheel-Leg Hybrid Robot with Passive Transformable Wheels. *Symmetry*, Marec 2023, zv. 15, s. 800.
- [33] SJAFFRIE, H. *Introduction to Self-Driving Vehicle Technology*. CRC Press, 2020. ISBN 978-0-429-31677-7.
- [34] SOE, M.; PHYU, K. a WIN, W. Kinematics and Control A Three-wheeled Mobile Robot with Omni-directional Wheels. In: September 2024.
- [35] S.R.O, R. C. *Autonomous Reversing T3T: Small-scale autonomous articulated Vehicle* online. Dostupné z: <https://www.rexcontrols.com/autonomous-reversing/>. [cit. 2025-04-30].
- [36] SUPPORT, L. P. R. . *LEGO® Education SPIKE™ Prime FAQs* online. Dostupné z: <https://education.lego.com/en-us/product-resources/spike-prime/troubleshooting/spike-prime-faqs/#general>. [cit. 2025-04-28].
- [37] TAJTI, F.; SZAYER, G.; KOVACS, B. a KORONDI, P. Robot Base with Holonomic Drive. *IFAC Proceedings Volumes (IFAC Papers-OnLine)*, August 2014, zv. 47, s. 5715–5720.
- [38] V. M. BUDANOV, Y. D. S. . A. M. F. Prevention of Oscillations of a Spherical Robot in Longitudinal Motion. *Journal of Computer and Systems Sciences International*, 2022, zv. 61, č. 4. Dostupné z: <https://link.springer.com/article/10.1134/S1064230722040049>.
- [39] VELAGIC, J.; KOVAC, I.; PANJEVIC, A. a OSMANOVIĆ, A. Design and Control of Two-Wheeled and Self-Balancing Mobile Robot. In: September 2021, s. 77–82.

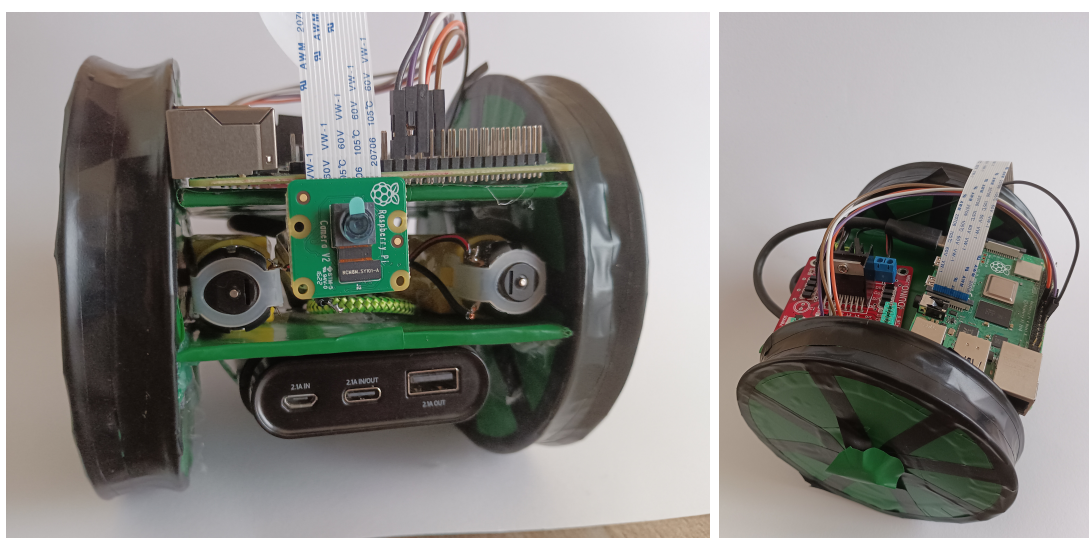
Zoznam príloh

A Model vozidla

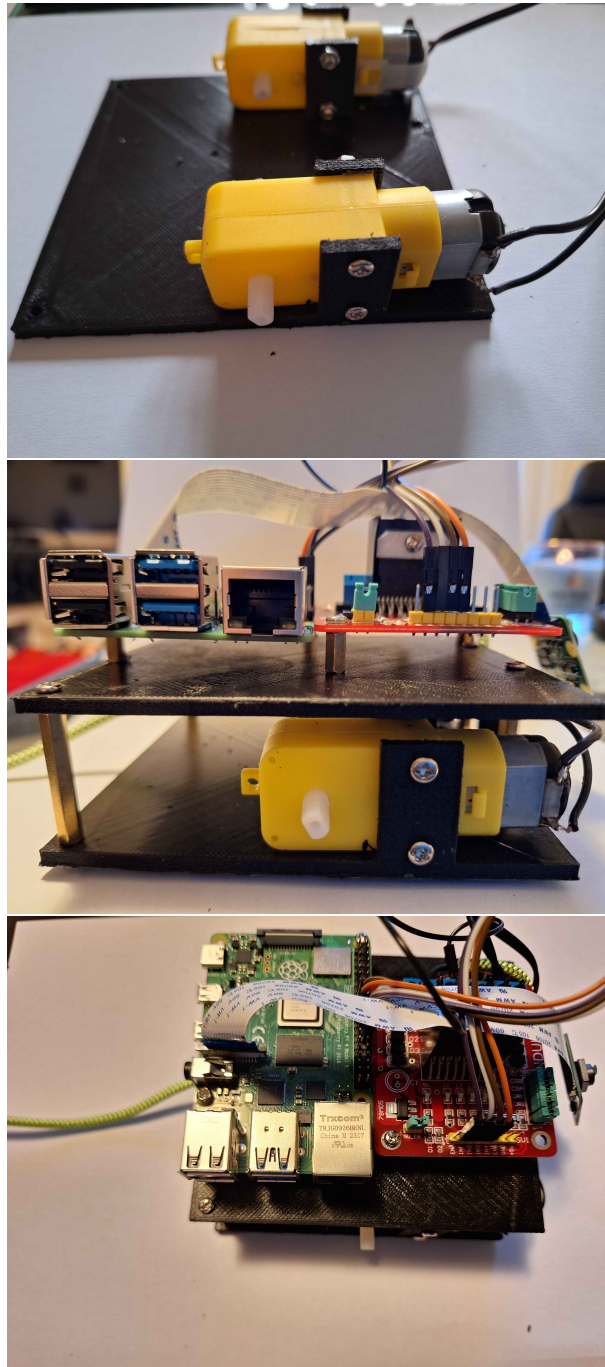
41

Príloha A

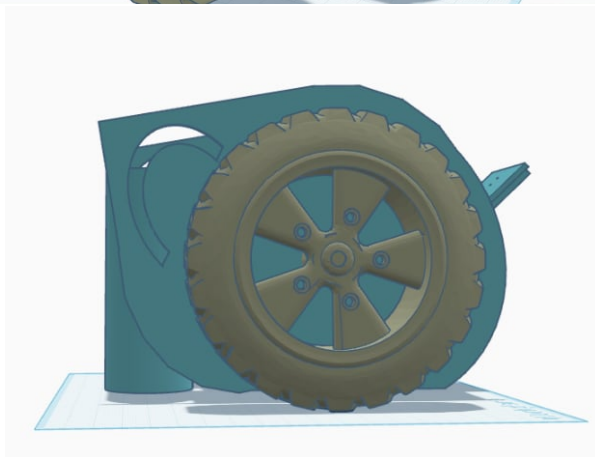
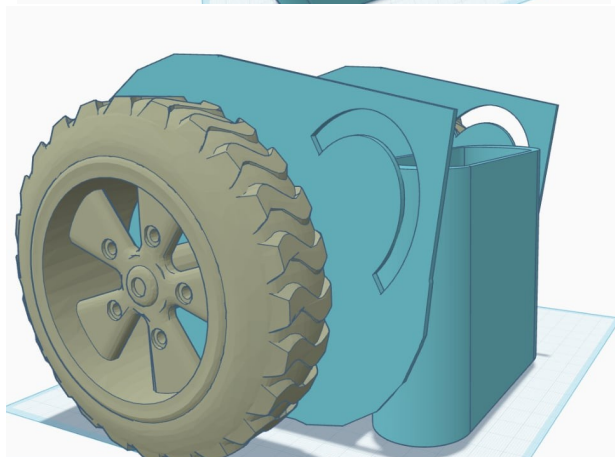
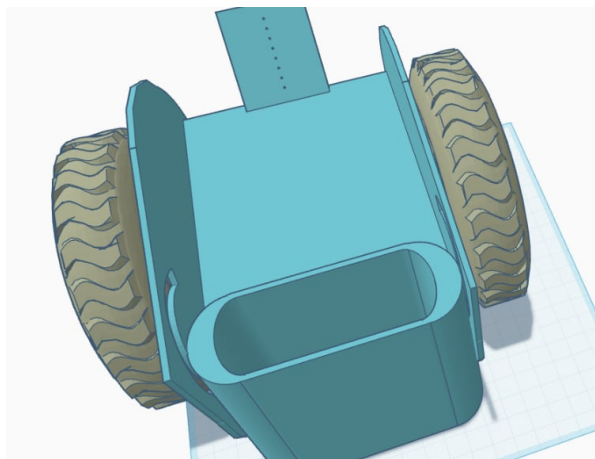
Model vozidla



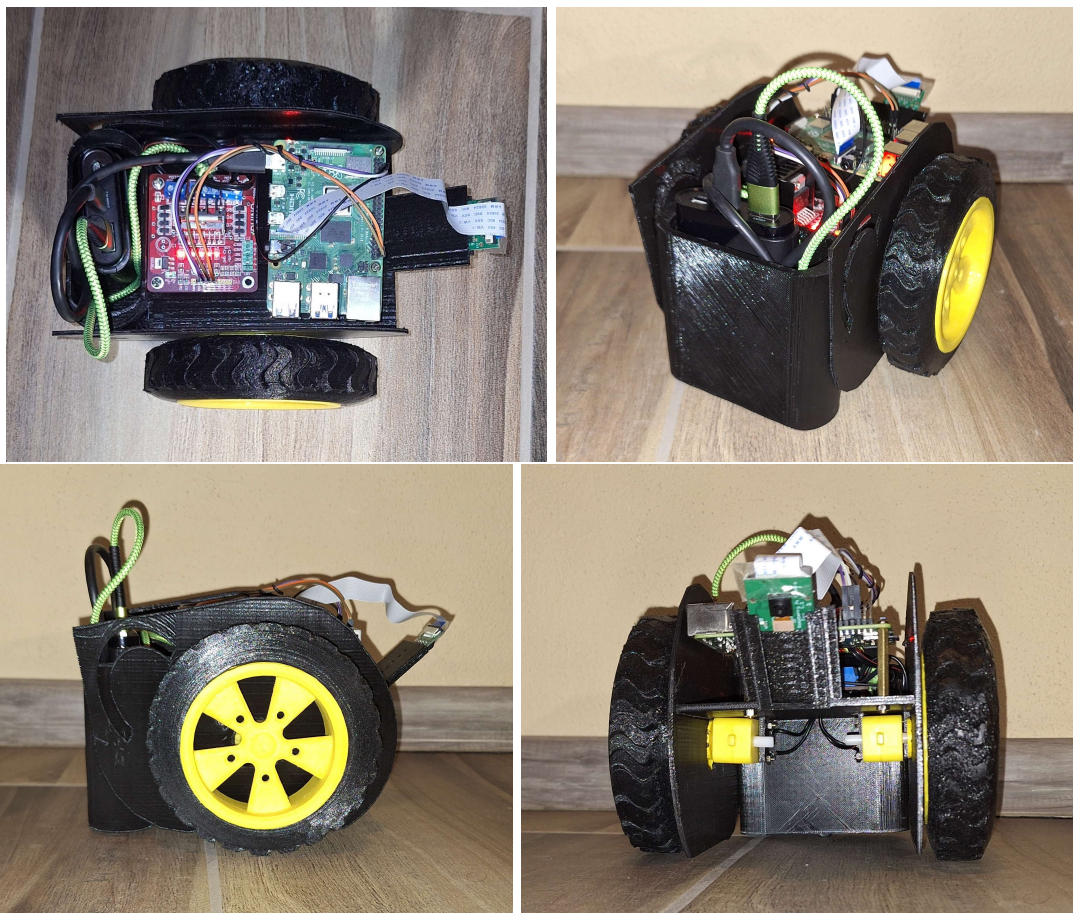
Obr. A.1: Prototyp prvého modelu vozidla



Obr. A.2: Prvý model vozidla



Obr. A.3: Druhý model vozidla v prostředí Tinkercad



Obr. A.4: Druhý model vozidla