VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

AUTONOMNÍ ŘÍZENÍ MODELU VOZIDLA

AUTONOMOUS MODEL-CAR DRIVING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **RADOMÍR STOJAN**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MICHAL BIDLO, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Stojan Radomír**
Program: Informační technologie
Název: **Autonomní řízení modelu vozidla**
Autonomous Model-Car Driving
Kategorie: Vestavěné systémy

Zadání:

1. Seznamte se s modelem vozidla postaveného na platformě NXP Cup Kit.
2. Vozidlo doplňte vhodnými senzory tak, aby bylo možné zkoumat možnosti jeho autonomního řízení.
3. Zvolte vhodné uspořádání úlohy a pro toto navrhněte alespoň jeden algoritmus s cílem dosažení co nejvyššího stupně autonomie modelového vozidla pro danou úlohu.
4. Proveďte sadu experimentů za účelem zjištění schopností vozidla.
5. Zhodnoťte dosažené výsledky a diskutujte možnosti pokračování projektu.

Literatura:

- Dle pokynů vedoucího projektu.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 a 2 zadání, demonstrace prototypu systému z bodu 3 zadání.

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

Vedoucí práce: **Bidlo Michal, Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 14. května 2020
Datum schválení: 25. října 2019

Abstrakt

Tato práce se zabývá problematikou autonomních parkovacích systémů. Na základě analýzy současných studií zabývajících se tvorbou parkovacích systémů byl sestaven model vozidla na platformě NXP Cup Alamak osazený senzory pro detekci překážek a zjištění rychlosti a ujeté vzdálenosti. Pro tento model byl navržen autonomní řídicí algoritmus. Podařilo se vytvořit algoritmus, pomocí něhož je model schopen samostatně vyhledat parkovací místo, vyhodnotit, jakým způsobem do toho místa zaparkovat a dovolují-li to podmínky parkovacího místa, provést parkovací manévr.

Abstract

This thesis focuses on the topic of creating autonomous parking systems. Based on the analysis of existing studies, a parking algorithm is presented for a vehicle model NXP Cup Alamak. This model was retrofitted with sensors in order to accurately measure vehicle speed and distance to surrounding obstacles. The proposed algorithm allows the vehicle model to autonomously detect and analyse available parking spaces, determine how to park in these parking spaces and to perform the parking maneuver if the dimensions of the parking space allow it.

Klíčová slova

Autonomní parkovací systém, Arduino, ESP-32, laserový senzor vzdálenosti, NXP Cup car Alamak, Inkrementální rotační enkodér.

Keywords

Autonomous parking system, Arduino, ESP-32, laser distance sensor, NXP Cup car Alamak, Incremental rotary encoder.

Citace

STOJAN, Radomír. *Autonomní řízení modelu vozidla.* Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Bidlo, Ph.D.

Autonomní řízení modelu vozidla

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Bidla, Ph.D. a uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Radomír Stojan
28. května 2020

Poděkování

Tímto bych chtěl poděkovat Ing. Michalu Bidlovi, Ph.D. za vedení této práce, dále pak Ing. Vojtěchu Mrázkovi, Ph.D. za pomoc s prací v době nemoci mého vedoucího a Petru Stehlíkovi za pomoc s výběrem součástek, které byly použity při tvorbě této práce.

Obsah

Kapitola 1

Úvod

V posledních letech je patrné čím dál větší rozšíření a nasazení různých autonomních systémů. Jejich hlavním smyslem je napodobit náročnou práci člověka tak, aby některé úlohy analýzy a rozhodování mohly být prováděny automaticky bez zásahu člověka. Jednou z oblastí, ve které dnes probíhá intenzivní výzkum, je autonomní řízení vozidel. Pomineme-li např. již běžně montované senzory, které řidiči usnadňují orientaci v provozu nebo třeba systémy, které mají za cíl předpovídat a eliminovat nebezpečné situace v dopravě, je hlavní důraz kladen i na autonomní řízení, kdy je značná část tohoto procesu přenechána počítači. Typickým představitelem řešení tohoto problému, který dnes již bývá dostupný, je autonomní parkovací systém.

Autonomní parkovací systémy jsou jedny z pokročilých asistenčních systémů, které se postupně stávají běžnou výbavou moderních automobilů. Cílem těchto systémů je asistovat řidiči při provádění parkovacích manévrů nebo dokonce úplně eliminovat potřebu zásahu člověka při provádění této rutinní činnosti a zajistit tak vyšší úroveň komfortu a bezpečnosti pro posádku vozidla i její okolí. Parkování je činnost nejčastěji prováděná v nízkých rychlostech, které obvykle nepřevyšují 10 $km \cdot h^{-1}$. Snahou je data ze senzorů zpracovat a interpretovat tak, aby celý proces parkování mohl být proveden bezpečně a plynule. Důraz je kladen na přesnost a rychlost zpracování dat o poloze, rychlosti a okolí vozidla. Existuje řada studií, které se tímto problémem zabývají a jejich souhrn bude předmětem samostatné kapitoly.

Cílem této práce je navrhnout a implementovat autonomní parkovací systém pro model vozidla na platformě NXP Cup Alamak. Tento model vhodně doplnit senzory tak, aby bylo možné detekovat vzdálenost k překážkám, které se nacházejí v okolí vozidla a přesně měřit rychlost a ujetou vzdálenost vozidla. Navržený systém musí zpracováním těchto údajů dosáhnout co nejvyšší autonomie u zvoleného parkovacího scénáře. Pomocí experimentů budou ověřeny schopnosti modelu a parkovacího systému.

Kapitola 2

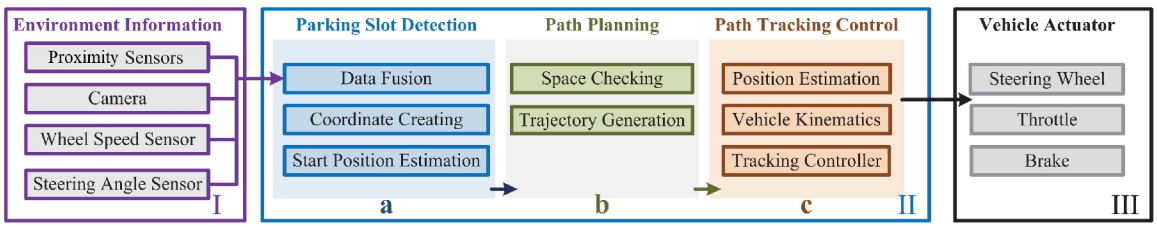
Autonomní parkovací systémy

Tato kapitola obsahuje shrnutí základních principů autonomních parkovacích systémů, popis obecné architektury takového systému a techniky, které se v této oblasti využívají. V samostatné sekci je podrobněji rozebrána vybraná studie, která dobře vystihuje základní principy tvorby parkovacích systémů. V závěru je uveden stručný souhrn algoritmů, které byly v posledních letech navrženy pro řešení této problematiky.

2.1 Architektura parkovacího systému

Architektura parkovacího systému (obrázek 2.1) závisí na přístupu, který byl zvolen při tvorbě parkovacího systému, avšak obecně lze v každém systému nalézt tyto 3 základní rysy:

- Senzorový systém,
- Řídící systém,
- Systém akčních členů.



Obrázek 2.1: Architektura parkovacího systému. Sekce I. obsahuje získávání informací o vozidle a jeho okolí pomocí senzorového systému vozidla. Sekce II. se zabývá zpracováním těchto údajů a řízením akčních členů vozidla, tyto úkony provádí řídicí systém. Sekce III. obsahuje akční členy vozidla. Převzato z [13].

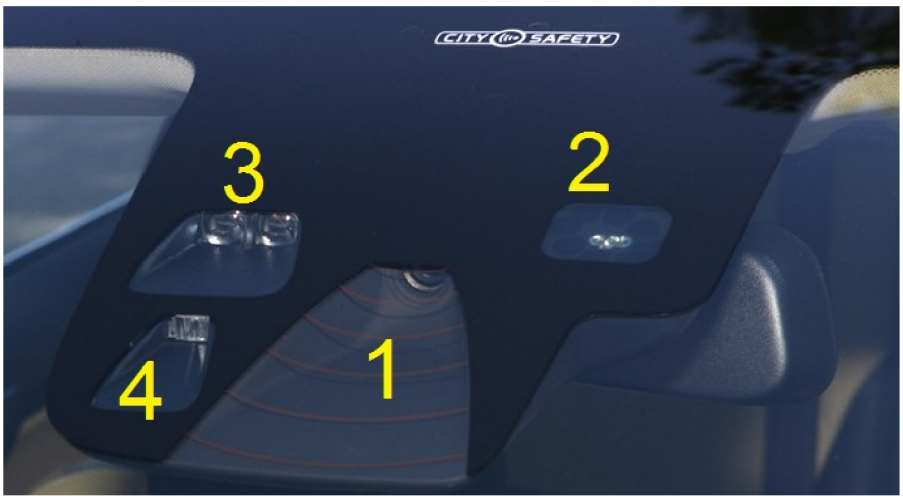
2.1.1 Senzorový systém

Aby bylo možné efektivně řídit proces parkování, je zapotřebí mít přesné údaje o vozidle a jeho okolí. K tomuto účelu slouží senzorový systém vozidla (obrázek 2.1, sekce I.),

Mezi hlavní informace o vozidle, které je třeba sledovat, patří rychlost vozidla a úhel natočení kol přední nápravy. K měření rychlosti je možné použít celou řadu senzorů, např: akcelerometry, GPS, magnetické, optické a mechanické otáčkoměry atp. V praxi se k měření rychlosti vozidla i úhlu natočení kol přední nápravy často používají různé varianty rotačních enkodérů. Příklady takových komponent je možné vidět na obrázcích 2.2.

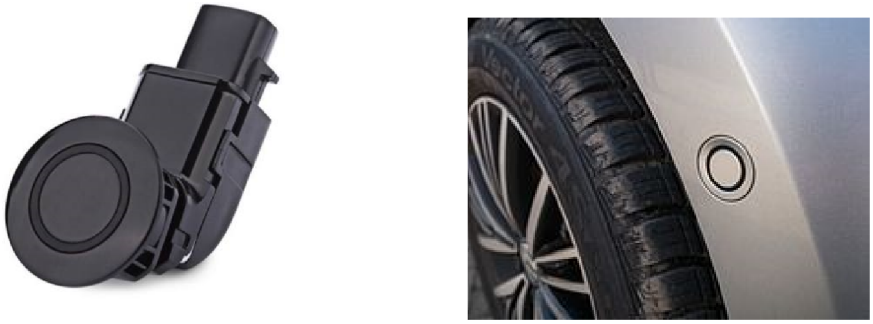
Obrázek 2.2: Elektromechanický rotační enkodér od firmy OMRON (vlevo), optický rotační enkodér od firmy BROADCOM (vpravo)

K získávání informací o prostředí kolem vozidla se v dnešní době používají kamerové systémy a senzory přiblížení. Kamerové systémy (obrázek 2.3) se používají primárně ke sledování dění před vozidlem, sledování tzv. mrtvých úhlů nebo jako parkovací kamery, které asistují řidiči při manuálním parkování. U některých vozidel se ovšem využívají kamerové systémy i pro účely autonomního parkování.



Obrázek 2.3: Kamerový modul City Safety. 1. běžná kamera, 2. dešťový senzor, 3. přijímač LIDAR, 4. vysílač LIDAR.

Pro účely parkování se používají primárně senzory přiblížení. V praxi se nejčastěji používají senzory ultrazvukové (obrázky 2.4), ale senzorů pro měření vzdálenosti existuje celá řada, například laserové senzory vzdálenosti, elektromagnetické senzory přiblížení a radar. Podrobnější popis vlastností a funkcionality vybraných senzorů bude předmětem samostatné kapitoly.



Obrázek 2.4: Ultrazvukový senzor. Samostatný modul (vlevo), senzor zabudovaný v nárazníku (vpravo).

2.1.2 Řídicí systém

Úkolem řídicího systému (2.1, sekce II.) je zpracovávat data ze senzorů a vyhledávat a analyzovat parkovací místa. Poté, co je nalezeno vhodné parkovací místo, je řídícím systémem podle dostupných údajů o poloze vozidla a jeho okolí vypočtena parkovací trajektorie, kterou se poté řídicí systém snaží co nejpřesněji provést řízením akčních členů vozidla. Celý řídicí proces lze rozdělit do 3 částí:

- Vyhledávání parkovacího místa,
- Výpočet parkovací trajektorie,
- Realizace parkovací trajektorie.

Během vyhledávání parkovacího místa (2.1, sekce II. a) se vozidlo pohybuje prostorem, řídicí systém spojuje dostupná data ze senzorů a mapuje prostředí okolo vozidla. Dle úrovně autonomie řídicího systému může být pohyb vozidla řízen člověkem nebo částečně či plně automatizován.

U systémů s nižší autonomií je obvykle přenecháno řízení člověku a systém pouze vyhledává vhodná místa k parkování, pokud je takové místo nalezeno, je dále nabídnuto řidiči, který musí výběr místa potvrdit, aby mohl začít proces parkování. U systémů s vyšší autonomií je i pohyb vozidla prostorem přenechán řídicímu systému, přičemž finální výběr parkovacího místa může být přenechán člověku nebo plně automatizován. Některé pokročilé systémy [6] se dokonce pokoušejí napodobit práci profesionálního řidiče tak, že řidič přijede s vozidlem do místa, v jehož blízkosti si přeje zaparkovat a poté může posádka vozidla vystoupit a vozidlo samočinně najde v okolí parkovací místo, do kterého zaparkuje. Zaparkované vozidlo přitom není třeba hledat, jelikož systém poskytuje funkci, kdy je vozidlo schopné samostatně vyhledat pozici řidiče a na tuto pozici se přemístit.

Tvorba parkovací trajektorie a její realizace (obrázek 2.1, sekce II. b,c) jsou spolu komplementární. Způsobů, kterými je možné vytvořit parkovací trajektorii, existuje celá řada a jsou závislé na algoritmu, který byl zvolen při tvorbě parkovacího systému. Na základě zvoleného algoritmu může být vytvořená trajektorie stálá nebo proměnná. Stálou trajektorii není možné po zahájení parkovacího manévru nijak měnit a dojde-li při takovém manévru k události, která nedovoluje vozidlu pokračovat (např. vstup chodců do parkovacího místa nebo chyba při analýze parkovacího místa) bývá takový manévr ukončen. Naopak u proměnných trajektorií se systém může do určité míry přizpůsobovat měnícímu se okolí i během provádění parkovacího manévru, takové systémy však obvykle bývají výpočetně mnohem náročnější. Jakmile je parkovací trajektorie stanovena, realizační část parkovacího systému se snaží řízením akčních členů řídit pohyb vozidla tak, aby bylo dosaženo maximální přesnosti mezi reálnou a požadovanou pozicí vozidla v každém bodě parkovacího manévru.

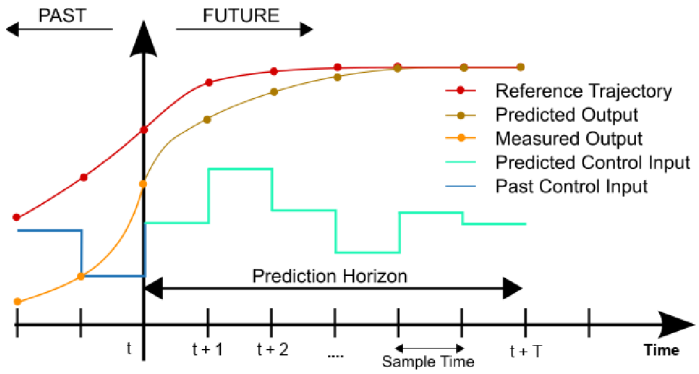
2.1.3 Systém akčních členů

Úkolem akčních členů (obrázek 2.1, sekce III.) je řídit pohybovou soustavu vozidla na základě příkazů od řídicího systému. Hlavní prvky vozidla, které je třeba řídit akčními členy, jsou plynový pedál, brzdový pedál a volant. Mezi nejčastěji používaný akční člen patří servomotory.

2.2 Metoda řízení modelů s predikcí

Metoda řízení modelů s predikcí (ang. model predictive control, dále jen MPC) [3] je heuristická řídící metoda, která byla představena v 70. letech 20. století. Jedná se o pokročilou metodu řízení, která se používá primárně pro řízení průmyslových a automatizačních procesů s omezujícími podmínkami.

Základním prvkem MPC je model procesu, který chceme řídit. Obvykle se jedná o modely lineární, ale lze použít i modely nelineární, ty však bývají výpočetně náročnější. Předpokladem pro dosažení dobrých výsledků je vhodná transformace reálného procesu a jeho hlavních rysů do návrhu modelu. Hlavní výhodou algoritmů MPC je možnost optimalizovat následující stav systému na základě aktuálního stavu systému, předpovědi budoucích stavů systému a porovnáním s požadovaným stavem systému v daném časovém okamžiku. Příklad časového průběhu, který znázorňuje tento princip, je možné vidět na obrázku 2.5. Jednotlivé křivky představují stavy systému a hodnoty vstupních či výstupních proměnných v daném časovém okamžiku. Červená křivka představuje požadovaný stav systému. Oranžová křivka ukazuje reálný stav systému. Hnědá křivka znázorňuje předpovídaný budoucí stav systému. Tmavě modrá křivka představuje reálné hodnoty vstupních proměnných. Světle modrá křivka znázorňuje předpovídané budoucí hodnoty vstupních proměnných.



Obrázek 2.5: Příklad časového průběhu pro řídicí systém využívající metodu MPC[2]. Osa x představuje časový rámec systému. Na ose y jsou znázorněny jednotlivé stavy, vstupy a výstupy systému.

Algoritmus pracuje iterativně, tedy vyhodnocení aktuálního stavu spolu s předpovědí budoucích stavů se opakuje v každé iteraci algoritmu. V každém kroku algoritmus vyhodnotí aktuální stav systému a na jeho základě předpoví budoucí stavy systému v intervalu $\langle t, t+T\rangle$, kde T je počet předpovědí budoucích stavů systému. Při předpovědi budoucích stavů je navíc možné aplikovat omezující podmínky systému i na jeho předpovídané stavy, čímž je možné docílit velmi přesných a realistických předpovědí budoucích stavů systému. Podle dostupných informací je poté provedena optimalizace následujícího stavu systému a na základě toho algoritmus provede potřebné úkony pro dosažení tohoto stavu pomocí akčních členů systému.

Za účelem zlepšení přesnosti algoritmu je možné kromě pevných omezujících podmínek systému do modelu přidat i tzv. měkké omezující podmínky. Takové podmínky umožňují některým prvkům systému na určitou dobu překročit pevně stanovené omezující podmínky modelu, což může přispět ke zlepšení výsledků a umožnit modelu nalézt i řešení, která by jinak nebyla dostupná.

2.2.1 MPC v parkovacích systémech

Ve studii [13] je představen automatický parkovací systém na základě kinematického modelu vozidla. Je navržena varianta MPC algoritmu s přidáním měkkých omezujících podmínek, přičemž algoritmus koordinovaně řídí rychlost a úhel natočení předních kol vozidla. V sekcích níže jsou podrobněji popsány jednotlivé součásti řídicího systému navrženého v [13].

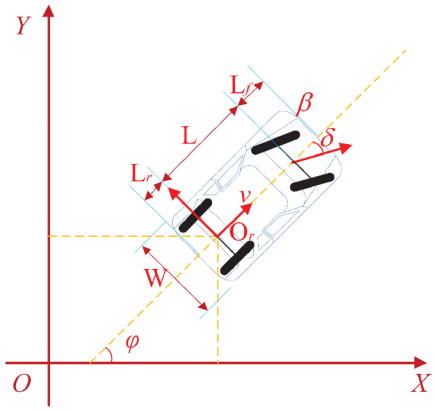
Kinematický model vozidla

Kinematický model vozidla představuje matematickou reprezentaci reálného vozidla a jeho pohybových vlastností. Jedná se o lineární model s omezujícími podmínkami. Zanedbáme-li dynamické vlastnosti vozidla a budeme brát v potaz pouze jeho pohybové vlastnosti, za ideálních parkovacích podmínek je možné kinematický model vozidla popsat rovnicí 2.1.

[2]Převzato z https://commons.wikimedia.org/wiki/File:MPC_scheme_basic.svg

$$\begin{bmatrix} x \\ y \\ \varphi \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ \tan\delta/L & 0 \end{bmatrix} \begin{bmatrix} v \\ \delta \end{bmatrix} \tag{2.1}$$

Kde v představuje rychlost vozidla, x,y jsou souřadnice středu zadní nápravy O_r vozidla v euklidovském prostoru. φ značí směr (úhel) vozidla v prostoru vůči ose x, L představuje rozvor náprav vozidla a δ je úhel předních kol vozidla. Podrobný diagram kinematického modelu spolu s popisem všech klíčových vlastností je možné vidět na obrázku 2.6.



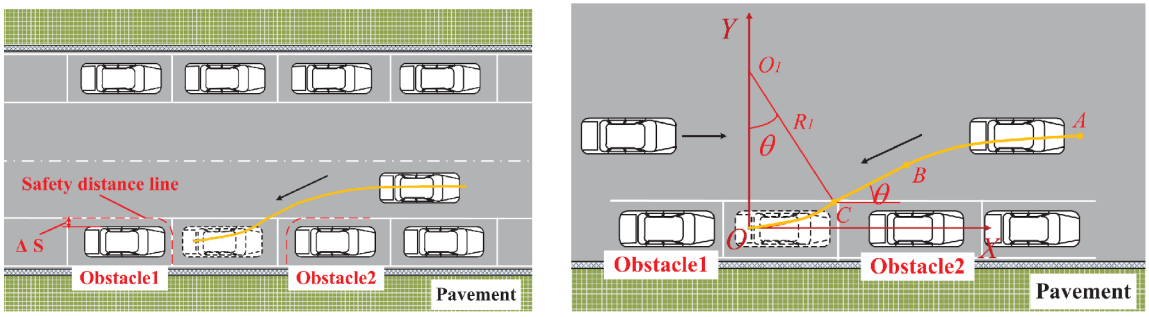
Obrázek 2.6: Diagram kinematického modelu vozidla. v značí rychlost vozidla, L značí rozvor náprav vozidla, L_f a L_r jsou délky přední a zadní nápravy, φ značí směr (úhel) vozidla v prostoru, δ značí úhel předních kol vozidla, W je šířka vozidla, β reprezentuje obrys vozidla, O_r je střed zadní nápravy vozidla a x,y jsou souřadnice středu zadní nápravy v euklidovském prostoru. Převzato z [13].

Tvorba parkovací trajektorie

Proces tvorby parkovací trajektorie se skládá ze 2 kroků:

- Detekce parkovacích míst v okolním prostředí pomocí senzorového systému vozidla a analýza jejich velikosti L_p.
- Pokud platí, že $L_p >= L_{pmin}$, kde L_{pmin} je minimální velikost parkovacího místa a L_p je velikost nalezeného parkovacího místa, je vypočtena optimální parkovací trajektorie v závislosti na velikosti vozidla, parkovacího místa, pozici vozidla a parametrů okolního prostředí.

Na obrázcích 2.7 je možné vidět příklad tvorby parkovací trajektorie a realizaci parkovacího manévru pro podélné parkování. Nechť body A,B,C,O leží na parkovací trajektorii. Proložením vhodné křivky těmito body vznikne parkovací trajektorie. Body A a O představují počáteční a koncový bod parkovací trajektorie. Sekce $\overline{AB}$ je počáteční parkovací křivka. Sekce $\overline{BC}$ je úsečka, která spojitě a plynule navazuje na sekce $\overline{AB}$ a $\overline{CO}$, jinými slovy se jedná o tečnu ke křivkám $\overline{AB}$ a $\overline{CO}$ v bodech B a C. Sekce $\overline{CO}$ je finální parkovací křivka kruhového tvaru se středem v bodě O_1 a poloměrem R_1. Vozidlo navíc v každém okamžiku parkovacího manévru musí udržovat minimální odstup od jakékoliv překážky. Tato vzdálenost je definována jako bezpečná vzdálenost ΔS

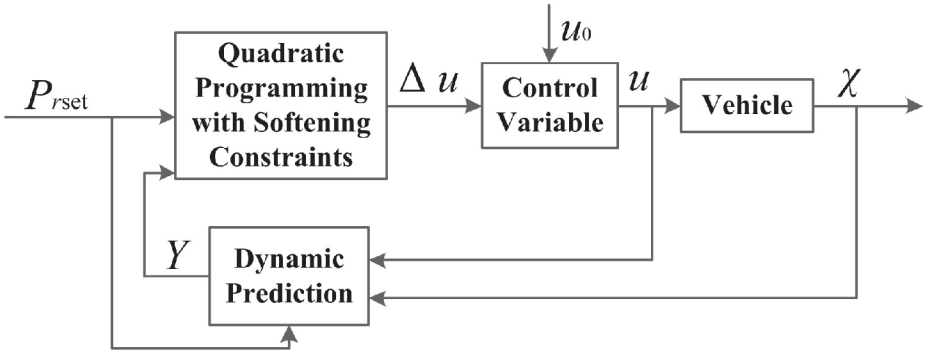


Obrázek 2.7: Příklad podélného parkování. Na obrázku vlevo je příklad parkovací trajektorie s vyznačenou bezpečnou vzdáleností ΔS. Na obrázku vpravo je podrobný popis této trajektorie a její jednotlivé části. Převzato z [13].

Návrh parkovacího kontroléru

Proces realizace parkovacího manévru po stanovené trajektorii je realizován sledováním sady referenčních bodů P_{rset}, které jsou rovnoměrně rozprostřeny po parkovací trajektorii a vznikly diskretizací spojité parkovací křivky $\overline{ABCO}$. Blokové schéma parkovacího kontroléru je možné vidět na obrázku 2.8. Prvním krokem je zjištění aktuálních informací o stavu vozidla v daném okamžiku pomocí senzorového systému vozidla. Tyto informace jsou reprezentovány stavovým vektorem χ. V následujícím kroku je možné provést predikci budoucích stavů systému Y na základě kombinace stavového vektoru vozidla χ, poslední hodnoty řídící proměnné u a referenčního bodu P_r pro daný časový okamžik. Dalším krokem je získání inkrementu řídící proměnné Δu pomocí optimalizace metodou kvadratického programování s měkkými omezujícími podmínkami. Posledním krokem je získání aktuálního stavu řídící proměnné u, přidáním inkrementu řídící proměnné Δu k poslední hodnotě řídící proměnné u.

Kvadratické programování [1] je jedna z metod nelineárního programování, což je optimalizační procedura pro nalezení minima či maxima funkce, jejíž proměnné jsou omezeny určitými podmínkami, kde tato funkce, omezující podmínky nebo obojí jsou nelineární. V případě kvadratického programování je tato funkce kvadratická a omezující podmínky jsou lineární.



Obrázek 2.8: Blokové schéma parkovacího kontroléru MPC s měkkými omezujícími podmínkami. Převzato z [13].

2.3 Souhrn studií

V dnešní době se používá celá řada metod, jakými je možné řešit problematiku autonomního parkování, stručný popis zajímavých přístupů se nachází níže.

Jednou z nejpoužívanějších metod pro tvorbu parkovacích systémů je použití teorie fuzzy logiky a fuzzy množin, které dokáží dobře vystihnout vlastnosti parkovacího procesu, které je velmi obtížné vystihnout přesným matematickým popisem. Příkladem může být [2], kde je za použití fuzzy logiky vytvořen parkovací systém, jehož pravidla byly vytvořeny na základě znalostí profesionálního řidiče. Pro sledování parkovací trajektorie je využita kamera a funkce příslušnosti tohoto systému byly optimalizovány pomocí genetického algoritmu, za účelem dosažení vyšší přesnosti řídicího systému.

Evoluční a genetické algoritmy se s úspěchem používají v různých sférách výzkumu a parkovací algoritmy nejsou výjimkou. V [12] je vytvořen řídicí systém vozidla, který za pomoci genetického algoritmu hledá optimální parkovací trajektorii. Na základě parametrů vozidla jsou odvozeny omezující podmínky parkovacího algoritmu. Tento přístup přináší větší flexibilitu a takový systém je schopen lepšího přizpůsobení se okolním podmínkám a využití dostupné místo než běžné lineární a nelineární systémy.

V [7] se autorům povedlo vytvořit autonomní parkovací systém podélného parkování pro situace, kdy rozměry parkovacího místa nejsou předem známy nebo je není možné zjistit. Architektura toho systému je hybridní a používá neuro-fuzzy kontrolér pro řízení parkovacího procesu. Testování a simulace ukázaly, že navržený systém je skutečně schopen parkovat bez znalosti velikosti parkovacího místa a to při použití pouze tří běžných ultrazvukových senzorů.

Další možností, která se v dnešní době stává velmi populární je použití neuronových sítí pro zpracování dat ze senzorů nebo dokonce pro celý řídicí systém. Takový přístup je představen ve studii [5], kde je pro řízení modelu použita vícevrstvá, dopředná neuronová síť. Tato práce se rovněž zabývá možností paralelizace výpočtů na systémech s více procesory. Za tímto účelem je navržena a otestována nová metodologie, pomocí níž bylo dosaženo kvalitnějších výsledků oproti konkurenčním systémům s podobnou architekturou.

Některé studie se zabývají parkováním ve větším měřítku. Příkladem může být [9], ve které je navržen systém, jehož účelem je vhodně navigovat vozidla k parkovacím místům na základě obsazenosti parkovacích míst v oblasti vozidla. Systém na základě dat o poloze, které mu vozidla poskytují například pomocí GPS nebo mobilních sítí, vyhodnocuje obsazenost jednotlivých parkovacích míst a parkovacích komplexů. Na základě těchto údajů doporučuje vozidlům, které chtějí parkovat, vhodná parkovací místa v jejich okolí. Pro tvorbu systému byla použita konvoluční neuronová síť. Experimenty prokázaly funkčnost systému i bez dostupnosti dat o poloze vozidel v reálném čase a schopnost systému lépe optimalizovat využití parkovacích míst například v parkovacích komplexech.

Kapitola 3

Platforma NXP Cup Alamak

Tato kapitola obsahuje podrobný popis Platformy modelu NXP Cup Alamak. V samostatných sekcích jsou popsány jednotlivé soustavy modelu a části, které bylo třeba doplnit, spolu s obrázky a diagramy.

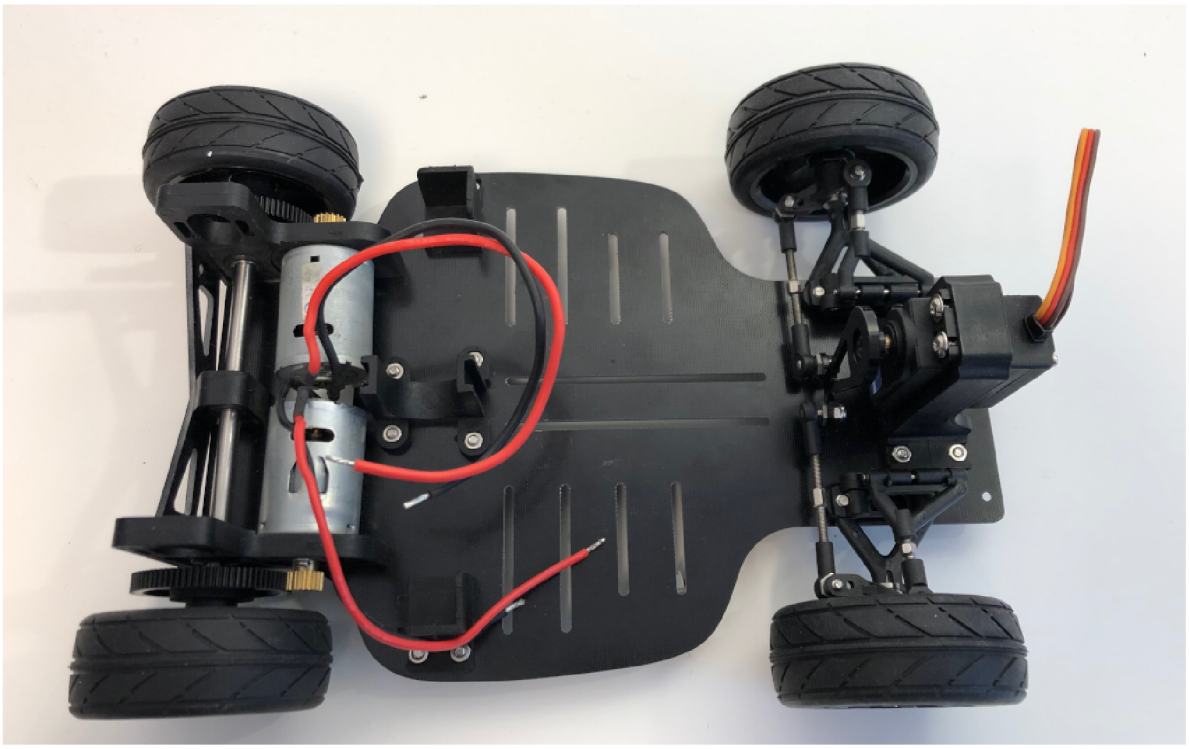
Model vozidla NXP Cup Alamak vznikl spoluprací firem Landzo a NXP jako platforma vozidla pro soutěžní účely v soutěži NXP Cup. Cílem této soutěže je v týmech sestavit model a naprogramovat jeho řídicí systém tak, aby bylo možné projet stanovenou dráhu v nejrychlejším možném čase bez toho, aniž by model vybočil ze stanovené dráhy. Typ modelu, který je použit v této práci, byl používán v ročníku 2017/2018 soutěže NXP Cup. Podrobný popis platformy je možné najít na gitbooku výrobce [10].

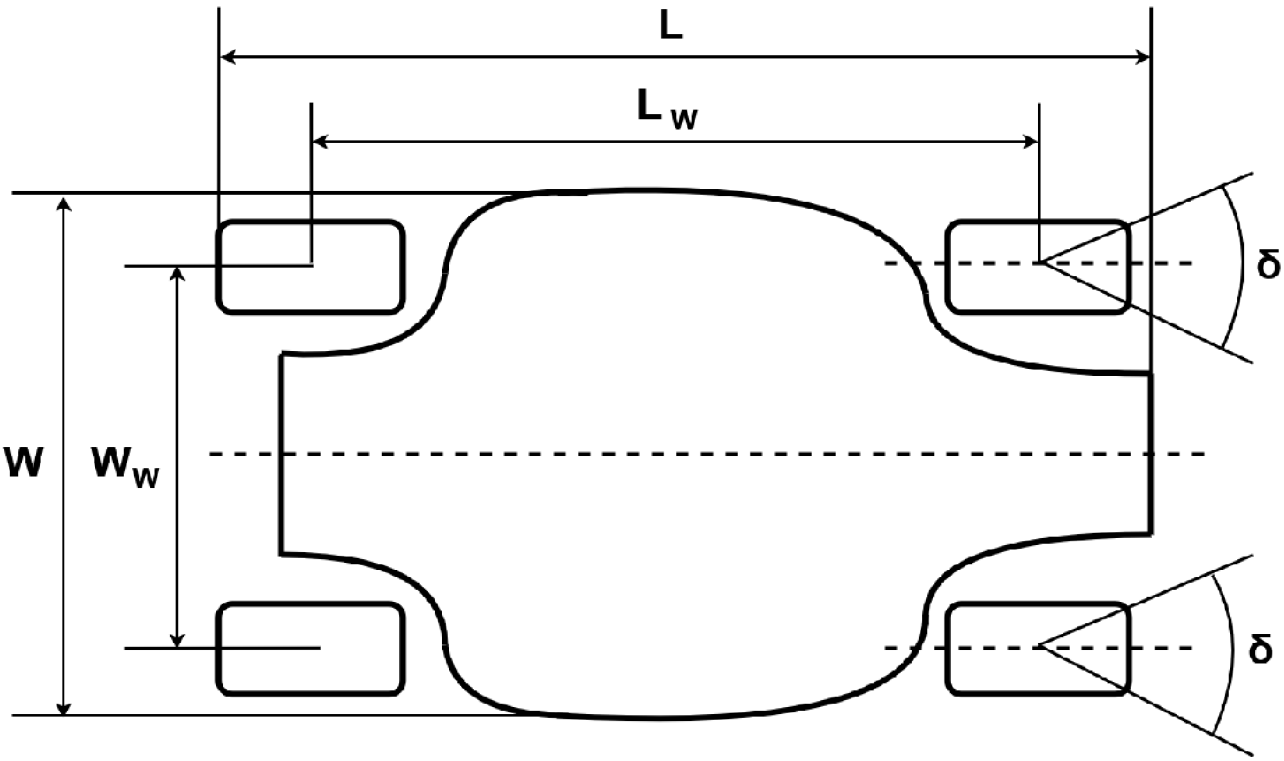
Tato platforma byla pořízena jako hotový výrobek, nicméně během prvních experimentů s ní se ukázalo, že trpí řadou nedostatků, které značně komplikují její použitelnost. Proto budou na příslušných místech této kapitoly rovněž zmíněny identifikované nedostatky a navržený způsob jejich řešení k přizpůsobení původní platformy potřebám této práce. Souhrnně tedy tato kapitola prezentuje celou hardwarovou platformu a její úpravy jsou jedním z přínosů této práce.

3.1 Podvozek

Podvozek modelu vozidla NXP Cup Alamak a jeho diagram je možné vidět na obrázcích 3.1. Podvozek je od výrobce osazen dvěma motory, které pohánějí zadní nápravu a servomotorem, který otáčí koly přední nápravy. Parametry podvozku jsou následující:

- Délka podvozku $L = 270\ mm$.
- Šířka podvozku $W = 180\ mm$.
- Rozvor náprav $L_w = 185\ mm$.
- Šířka náprav $W_w = 155\ mm$.
- Rádius natočení předních kol $\delta = 40°$.
- Hmotnost = 690 g.





Obrázek 3.1: Podvozek modelu NXP Cup Alamak. Obrázek podvozku (nahoře), diagram podvozku (dole).

3.2 Napájecí soustava

Hlavním napájecím zdrojem modelu je 6-ti článkový Ni-MH akumulátor (obrázek 3.2) s kapacitou 2500 mAh. Jmenovité napětí akumulátoru činí 7,2 V, přičemž napětí při maximálním nabití se pohybuje okolo 8,8 V.

Obrázek 3.2: Baterie modelu NXP Cup Alamak. Jedná se o 6-ti článkový Ni-MH akumulátor s kapacitou 2500 mAh a jmenovitým napětím 7,2 V.

Originální napájecí obvod pro servomotor se nachází na systémové desce, ovšem volba velmi silného servomotoru spolu se špatně dimenzovaným napájecím obvodem zapříčinila, že napájecí obvod se při pohybu servomotoru náhodně vypínal. S tímto problémem se setkali i jiní řešitelé, používající stejnou platformu, a podrobnější popis tohoto problému je možné najít v diplomové práci [11]. Bylo tedy třeba navrhnout náhradní řešení a pro napájení servomotoru byl zvolen impulzní DC-DC step-down měnič s regulátorem LM2596[1] (obrázek 3.3), který umožňuje plynule regulovat výstupní napětí v rozsahu 3 – 33 V. Maximální výstupní proud činí 3 A a výkon až 15 W, to je pro potřeby servomotoru zcela dostatečné. Napájení zbytku systému a periferií zprostředkovává motorová a systémová deska. Tyto desky dále regulují vstupní napětí z akumulátoru na napěťové úrovně 5 V a 3,3 V. Kompletní schéma znázorňující napájecí soustavu modelu se nachází v příloze B.1.

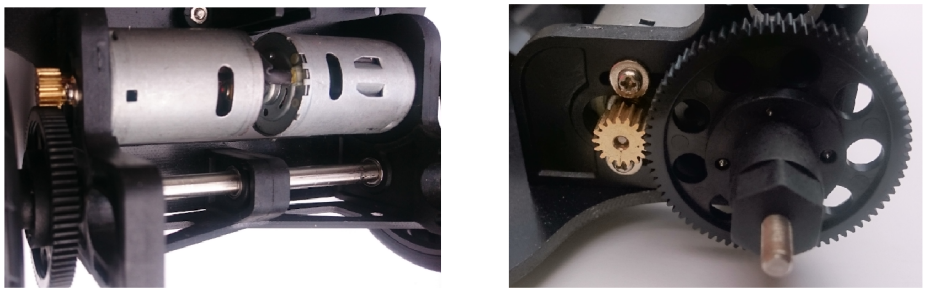
Obrázek 3.3: Napájení servo motoru. Impulzní DC-DC step-down měnič.

[1] Technický list step-down regulátoru LM2596: https://www.ti.com/lit/ds/symlink/lm2596.pdf?ts=1590578148496.

3.3 Pohybová soustava

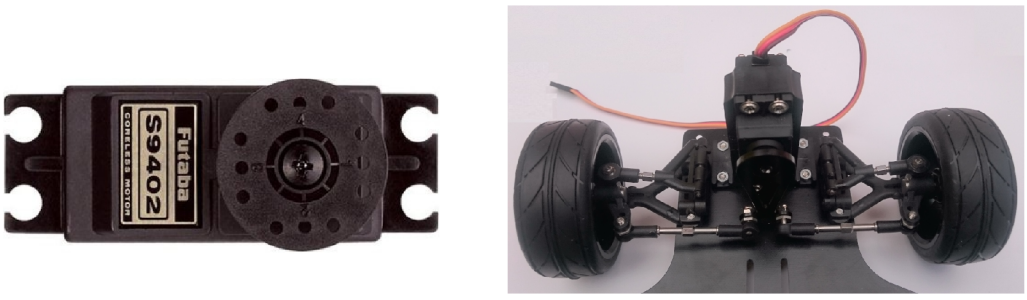
Pohon zadní nápravy zajišťují dva nezávisle řízené stejnosměrné motory od neznámého výrobce (obrázky 3.4). Řízení výkonu motorů je realizováno pomocí pulzně-šířkové modulace [4] (dále jen PWM). Každý motor je napájen pomocí H-můstku, který je tvořen dvěma polovičnímy můstky BTN7960[2]. Řídicím vstupem poloviční ho můstku je PWM signál, kterým je řízen výkon motoru. Výběrem poloviční ho můstku, na který je PWM signál přiveden je možné řídit směr otáčení motoru, přičemž řídící signál druhého poloviční ho můstku musí být vždy v nule.

Výstupem motoru je kovové ozubené kolo se 17 zuby, které otáčí větším plastovým ozubeným kolem se 78 zuby a to je připevněno ke kolu zadní nápravy. Výsledný převodový poměr je tedy přibližně 4,588:1 (otáčky motoru : otáčky zadního kola).



Obrázek 3.4: Motory zadní nápravy. Uložení motorů (vlevo), Výstupní převod motoru a finální převod kola zadní nápravy (vpravo).

O řízení přední nápravy se stará servomotor Futaba S9402[3] (obrázky 3.5). Napájecí napětí servomotoru je stejnosměrné a mělo by se pohybovat v rozsahu 4 – 7,2 V. Jedná se o velmi silný servomotor s kovovými převody a krouticím momentem až 8 $kg \cdot cm$ při napětí 6 V. Řízení servomotoru je analogové, tedy k jeho řízení je třeba jednoho PWM signálu. Pohybový rozsah servomotoru je přibližně 170°, což je pro potřeby modelu více než dostatečné, jelikož uchycení kol přední nápravy neumožňuje servomotoru natočení o více než 10° na každou stranu od pozice středu.



Obrázek 3.5: Servomotor Futaba S9402. Samostatné servo (vlevo), servo osazené na modelu NXP Cup Alamak (vpravo).

[2]Technický list poloviční ho můstku BTN7960: https://www.mouser.com/datasheet/2/196/Infineon-BTN7960-DS-v01_01-en-785559.pdf.

[3]Servo Futaba S9402: https://www.astramodel.cz/cz/katalog/futaba/futaba-servo-s9402-8-0kg-cm-0-10s-60-mg-wp-p399.html.

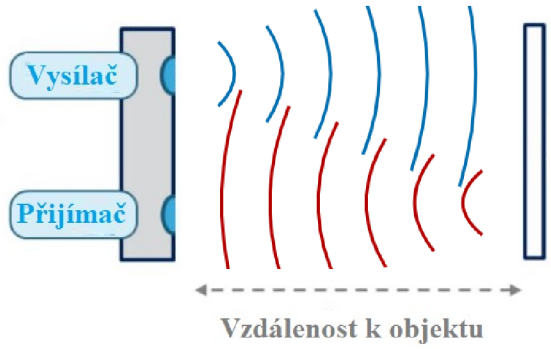
3.4 Senzorová soustava

Model NXP Cup Alamak v základu neposkytuje žádnou senzorovou výbavu, která by byla vhodná pro účely této práce. Bylo tedy nutné navrhnout senzorový systém a model jím doplnit. Princip fungování, výhody a nevýhody jednotlivých senzorů, které byly zváženy pro použití v této práci budou předmětem následujících sekcí.

3.4.1 Senzory přiblížení

Pro měření vzdálenosti objektů v okolí vozidla se v dnešní době používá celá řada senzorů. V této práci se podrobněji zaměříme na skupinu tzv. time-of-flight (ToF) senzorů přiblížení, které měří vzdálenost k objektu pomocí doby letu signálu, od vysílače k objektu a zpět k přijímači. Princip je možné vidět na obrázku 3.6. Vzdálenost k objektu je možné obecně popsat pomocí rovnice 3.1.

$$Vzdalenost = \frac{Dobaletusignalu \cdot Rychlostsignalu}{2} \tag{3.1}$$



Obrázek 3.6: Princip měření vzdálenosti pomocí ToF[4]senzoru.

Ultrazvukový senzor HC-SR04

Ultrazvukové senzory jsou v praxi jedny z nejpoužívanějších ToF senzorů přiblížení. Jako měřicí signál jsou použity zvukové vlny s frekvencí obvykle v rozsahu 40 – 250 kHz. Rychlost šíření zvukových vln prostorem je silně závislá na teplotě a druhu média, kterým se vlny šíří. Ve vzduchu za pokojové teploty tato rychlost činí přibližně 343 $m \cdot s^{-1}$.

Pro použití v této práci byl zvažován ultrazvukový senzor HC-SR04 na obrázku 3.7. Jedná se o velmi jednoduchý ultrazvukový senzor, který při každém měření vysílá 8 zvukových vln s frekvencí 40 kHz. Měřicí rozsah senzoru je 2 – 400 cm a velikost senzoru činí 45x20x16 mm.

[4]Obrázek převzat z: https://www.onelectrontech.com/stmicroelectronics-vl53l1x-tof-proximity-sensor-detects-distance-up-to-4-meters/, upraveno.

Obrázek 3.7: Ultrazvukový senzor vzdálenosti HC-SR04.

Tento senzor má ovšem řadu nevýhod, například to, že senzor samotný neposkytuje jako výsledek naměřenou vzdálenost, ale pouze informaci o tom, kdy signál dorazil překlopením logické úrovně na pinu Echo z 0 na 1. Výpočet samotné vzdálenosti tedy musí provést řídicí systém. Další nevýhodou je skutečnost, že senzor je stavěný pro použití se systémy s 5V logikou a jelikož logika systému v této práci je 3,3 V, bylo by třeba každý senzor doplnit převodníkem logických úrovní. Dalším problémem se ukázala být velikost samotného senzoru, který je pro použití na modelu vozidla NXP Cup Alamak příliš velký. V neposlední řadě je přesnost senzoru ovlivnitelná na základě úhlu, který svírá překážka vůči senzoru. Tento problém je podrobněji popsán v diplomové práci [11]. Z těchto důvodů byla tedy možnost použití senzoru HC-SR04 v této práci vyloučena.

Laserový senzor VL53L0X

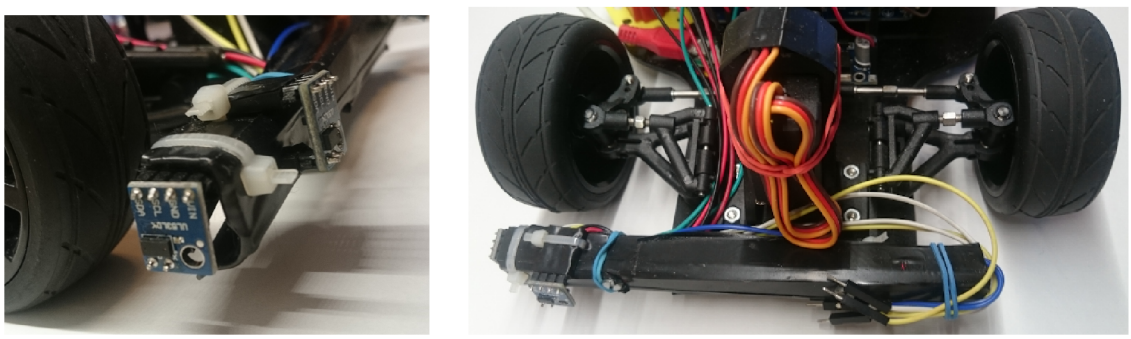
Laserové senzory vzdálenosti pracují na podobném principu jako ultrazvukové senzory, hlavní rozdíl je ovšem v typu použitého měřicího signálu, kterým je u těchto senzorů světelné záření. Rychlost světelného záření ve vakuu činí 299 792 458 $m \cdot s^{-1}$. Taková rychlost umožňuje velmi rychlá měření i na delší vzdálenosti, klade ovšem mnohem větší nároky na výpočetní techniku senzorů, jelikož světelné záření je schopné urazit vzdálenost jednoho metru přibližně za 3 nano sekundy.

Senzor použitý v této práci je laserový senzor vzdálenosti VL53L0X od firmy STMicroelectronics. Samostatný modul senzoru spolu s osazením na desce je možné vidět na obrázcích 3.8. Velikost samostatného senzoru je 4,4x2,4x1,0 mm a velikost senzoru s deskou je 29,5x22x2mm. Detekční rozsah senzoru je přibližně 2 – 2000 cm. Deska poskytuje 6 pinů: VIN, GND, SDA, SCL, GPIO1 a XSHUT. Piny VIN a GND slouží k napájení senzoru, ten je napájen stejnosměrným napětím v rozsahu 2,6 - 3,3 V. Piny SCL a SDA slouží ke komunikaci se senzorem pomocí sběrnice I^2C. Na pinu GPIO1 může senzor při určitých událostech generovat přerušení. Pin XSHUT slouží k ovládání napájení senzoru.

Obrázek 3.8: Laserový senzor vzdálenosti VL53L0X. Samostatný modul (vlevo), modul osazený na desce (vpravo).

Pro měření je použito světelné záření o vlnové délce 940nm, které je pro lidské oko neviditelné a zcela bezpečné. Vysílaný paprsek má tvar kužele, který se postupně rozšiřuje pod úhlem 25°. Velkou výhodou oproti senzoru HC-SR04 je to, že o výpočet vzdálenosti se stará samotný senzor a není tedy potřeba zatěžovat výpočtem řídící systém. Výsledkem měření senzoru je vzdálenost k překážce v milimetrech. Komunikace se senzorem probíhá pomocí sběrnice I^2C [4]. Podrobnější popis komunikace je možné najít v dokumentaci výrobce[5].

Pro účely vyhledávání parkovacích míst, detekci překážek a parkovaní byly použity 2 senzory VL53L0X, jejich umístění na modelu NXP Cup Alamak je možné vidět na obrázcích 3.9.



Obrázek 3.9: Osazení laserových senzorů vzdálenosti VL53L0X na modelu vozidla NXP Cup Alamak.

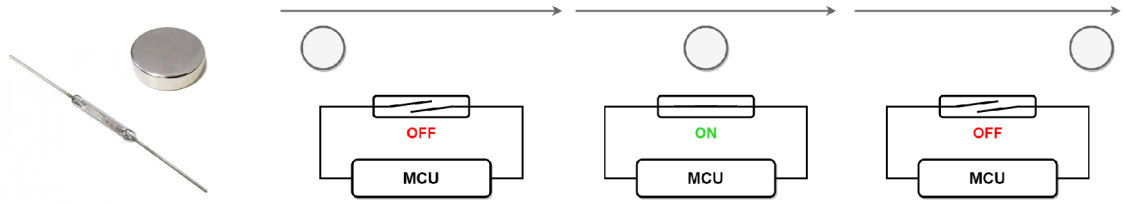
[5]Dokumentace laserového senzoru vzdálenosti VL53L0X: https://www.st.com/resource/en/datasheet/vl53l0x.pdf.

3.4.2 Senzory rychlosti a ujeté vzdálenosti

Jelikož platforma modelu Alamak v základu neposkytuje žádnou možnost pro měření rychlosti a ujeté vzdálenosti modelu, bylo třeba tuto funkcionalitu doplnit, protože přesné informace o rychlosti a ujeté vzdálenosti vozidla jsou pro účely parkování a efektivního řízení modelu naprosto nezbytné.

Magnetický spínač

Jako jednoduchá možnost měření rychlosti vozidla bylo uvažováno použití magnetických spínačů spolu s neodymovými magnety. Princip měření je možné vidět na obrázcích 3.10. Neodymové magnety byly rovnoměrně usazeny na kola zadní nápravy a do jejich blízkosti bylo umístěno magnetické relé. Při rotaci kola dochází k přiblížení magnetů k magnetickému relé, které v určité vzdálenosti od magnetu sepne a vyšle přerušení řídícímu systému, který přerušení zpracuje a získá tak informaci o rychlosti kol a ujeté vzdálenosti.



Obrázek 3.10: Magnetický spínač. Spínač spolu s neodymovým magnetem (vlevo), princip měření (vpravo).

Nevýhodou tohoto řešení se ukázala být malá přesnost a chybovost takového systému. Magnety od sebe musí mít určitou minimální vzdálenost, aby nedocházelo k rušivým vlivům od okolních magnetů. To znamenalo, že maximální počet magnetů použitelných na jednom kole byl 5. Při takovém množství bylo možné dosáhnout maximální přesnosti měření přibližně 4 cm. Výpočet přesnosti měření je dán rovnicí 3.2.

$$P = \frac{\pi \cdot W_d}{M_c} = \frac{\pi \cdot 6,5}{5} \doteq 4,08cm \tag{3.2}$$

Kde P je maximální přesnost měření, W_d je průměr kol modelu a M_c je počet magnetů osazených na kole modelu. Taková přesnost je pro účely této práce nedostatečná, proto bylo třeba nalézt jiné řešení.

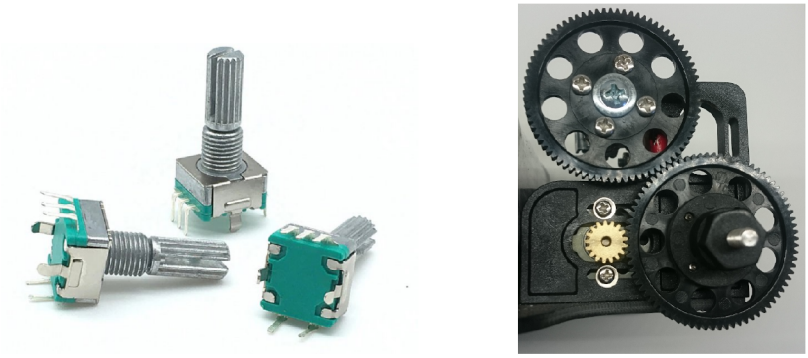
Rotační enkodér

Enkodéry[6] jsou používány takřka v každém odvětví průmyslu a automatizace, kde je třeba dosáhnout přesného měření lineárních nebo rotačních pohybů. Tyto senzory obecně převádějí rotační nebo lineární pohyb na digitální či analogový signál, který umožňuje zjistit rychlost a směr takového pohybu. Existují dva hlavní typy enkodérů: inkrementální a absolutní. Inkrementální enkodéry udávají pozici vztaženou k nějakému počátečnímu bodu,

[6]Základní popis enkodérů: https://www.rls.si/en/blog/how-to-select-an-encoder.

nejčastěji k nule. K této pozici se potom při pohybu inkrementálně přičítá nebo odčítá hodnota, jejíž velikost je úměrná uražené vzdálenosti. Absolutní enkodéry udávají přesnou pozici v určitém intervalu, například u lineárního pohybu 1 metr, u rotačního 1 otáčka (360°). Při překročení tohoto intervalu potom dojde k vynulování pozice enkodéru.

Senzor použitý v této práci je inkrementální rotační enkodér EN11[7] od firmy TTElectronics (obrázky 3.11), který při natočení hřídele vytváří 2 analogové signály. Tento enkodér kromě měření rotace zároveň poskytuje funkci tlačítka a obsahuje 5 výstupních pinů: 3 pro měření rotačního pohybu a 2 pro funkci tlačítka. Měřící hřídel senzoru poskytuje neomezený pohyb v obou směrech otáčení a senzor generuje 20 impulsů na 1 otáčku hřídele. K převodu rotačního pohybu z kola modelu Alamak je použito stejné plastové ozubené kolo, které tvoří finální převod u motorů pohybové soustavy. Jelikož kolo modelu a hřídel enkodéru používá stejné ozubené kolo, jejich převodový poměr je 1:1.



Obrázek 3.11: Rotační enkodér EN11 od firmy TTElectronics (vlevo), propojení s pohybovou soustavou modelu (vpravo).

Rozlišení senzoru je možné dále ovlivnit způsobem kódování signálů A a B. U inkrementálních enkodérů se používají 3 hlavní způsoby kódování[8]:

- *X1* - jsou počítány pouze náběžné nebo sestupné hrany jednoho ze signálů a rozlišení senzoru zůstává stejné.
- *X2* - jsou počítány náběžné i sestupné hrany jednoho ze signálů a rozlišení senzoru je dvojnásobné.
- *X4* - jsou počítány náběžné i sestupné hrany obou signálů a rozlišení senzoru je čtyřnásobné.

V této práci je použito kódování *X4* a výsledné rozlišení senzoru je tedy 80 impulsů na 1 otáčku kola modelu. Je tedy možné dosáhnout měření ujeté vzdálenosti s přesností přibližně 0,255 cm. Přesnost měření je dána rovnicí 3.3.

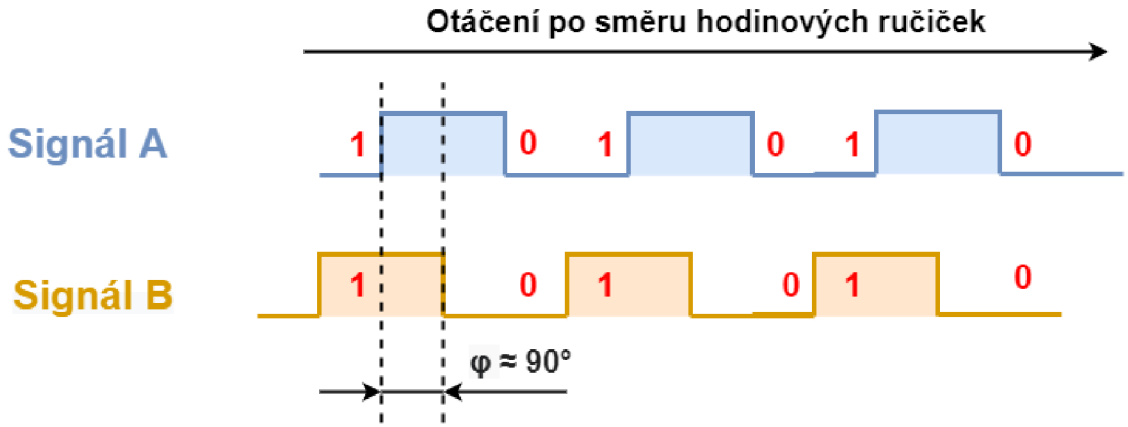
$$P = \frac{\pi \cdot W_d}{I_c \cdot R_a} = \frac{\pi \cdot 6,5}{20 \cdot 4} \doteq 0,255cm \tag{3.3}$$

[7]Dokumentace rotačního enkodéru EN11: https://www.ttelectronics.com/TTElectronics/media/ProductFiles/Encoders/Datasheets/EN11.pdf.

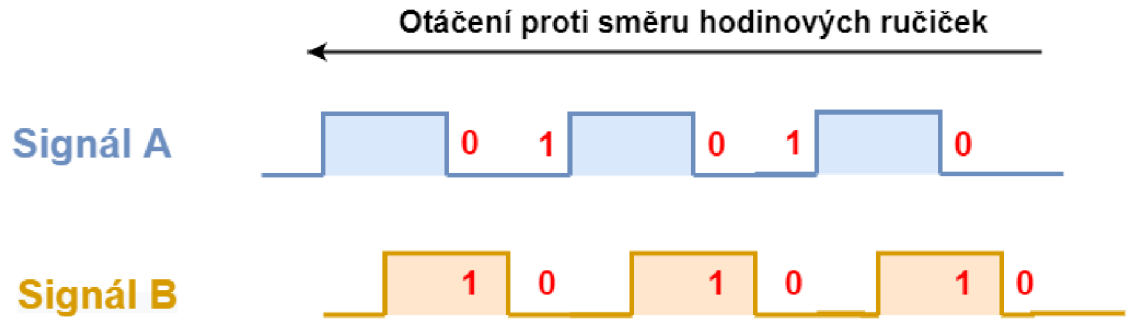
[8]Způsoby kódování u inkrementálních enkodérů: https://www.motioncontroltips.com/faq-what-do-x1-x2-and-x4-position-encoding-mean-for-incremental-encoders/.

Kde P je maximální přesnost měření, W_d je průměr kol modelu a I_c je základní počet impulsů na 1 otáčku hřídele enkodéru a R_a je násobitel rozlišení senzoru pro zvolený typ kódování.

Pro měření rotačního pohybu slouží piny A,B z obr. 3.12 a 3.13, na kterých senzor měří napětí vůči zemi. Piny A a B jsou pomocí pull-up rezistorů připojeny na kladné napětí řídícího systému. Při rotaci hřídele enkodéru dochází k postupnému uzemňování pinů A a B, čímž vznikají obdélníkové signály. V případě enkodéru EN11 bude těchto signálů 20 na jednu otáčku hřídele. Enkodér je konstruován tak, že obdélníkové signály jsou vůči sobě fázově posunuty přibližně o 90° a signál B předbíhá signál A. Tento fázový posun umožňuje zjistit, jakým směrem se otáčí hřídel enkodéru. Budeme-li se dívat na hodnoty obou signálů v momentech, kdy signál A mění svou logickou hodnotu, budou při otáčení hřídele po směru hodinových ručiček logické hodnoty obou signálu stejné. Naopak při otáčení hřídele proti směru hodinových ručiček budou logické hodnoty obou signálů opačné. Tento princip je znázorněn na obrázcích 3.12 a 3.13.



Obrázek 3.12: Princip detekce směru otáčení hřídele enkodéru. ukázka otáčení po směru hodinových ručiček, φ značí fázový posun signálů.

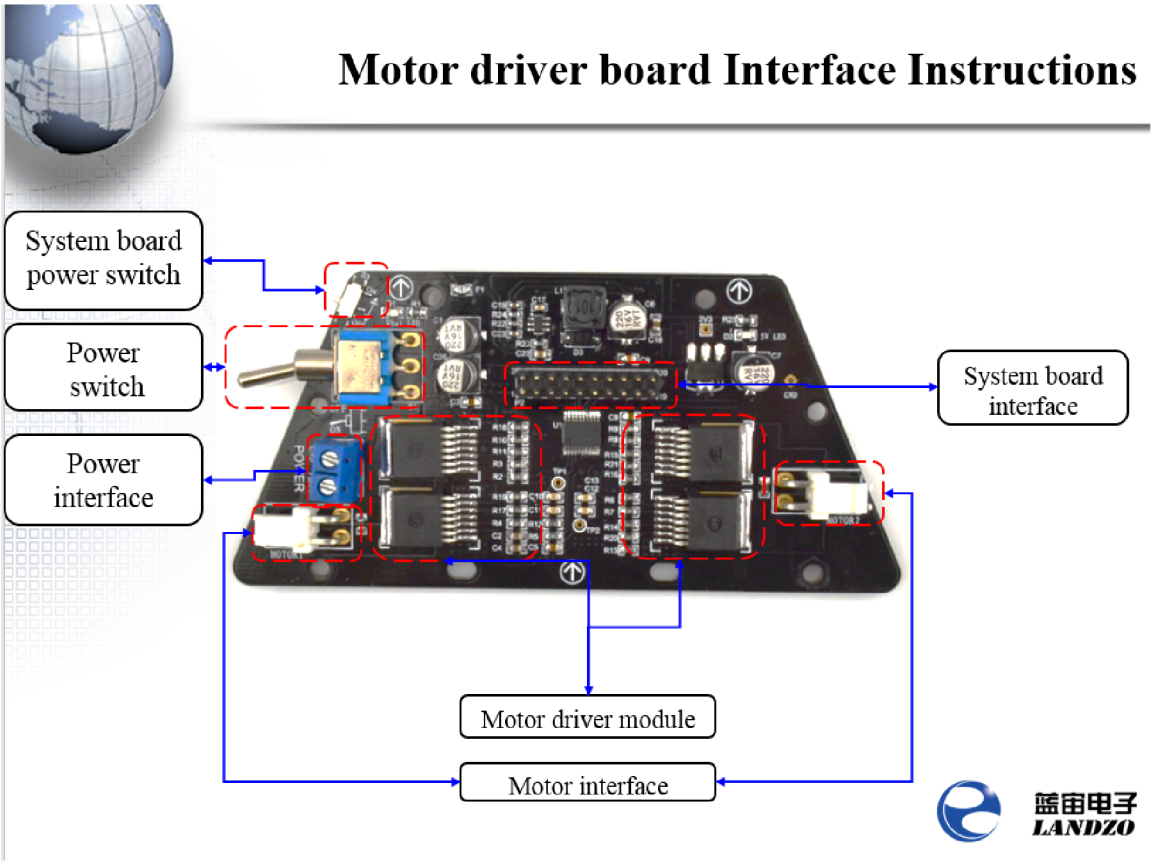


Obrázek 3.13: Princip detekce směru otáčení hřídele enkodéru. ukázka otáčení proti směru hodinových ručiček.

3.5 Řídicí soustava

3.5.1 Motorová deska

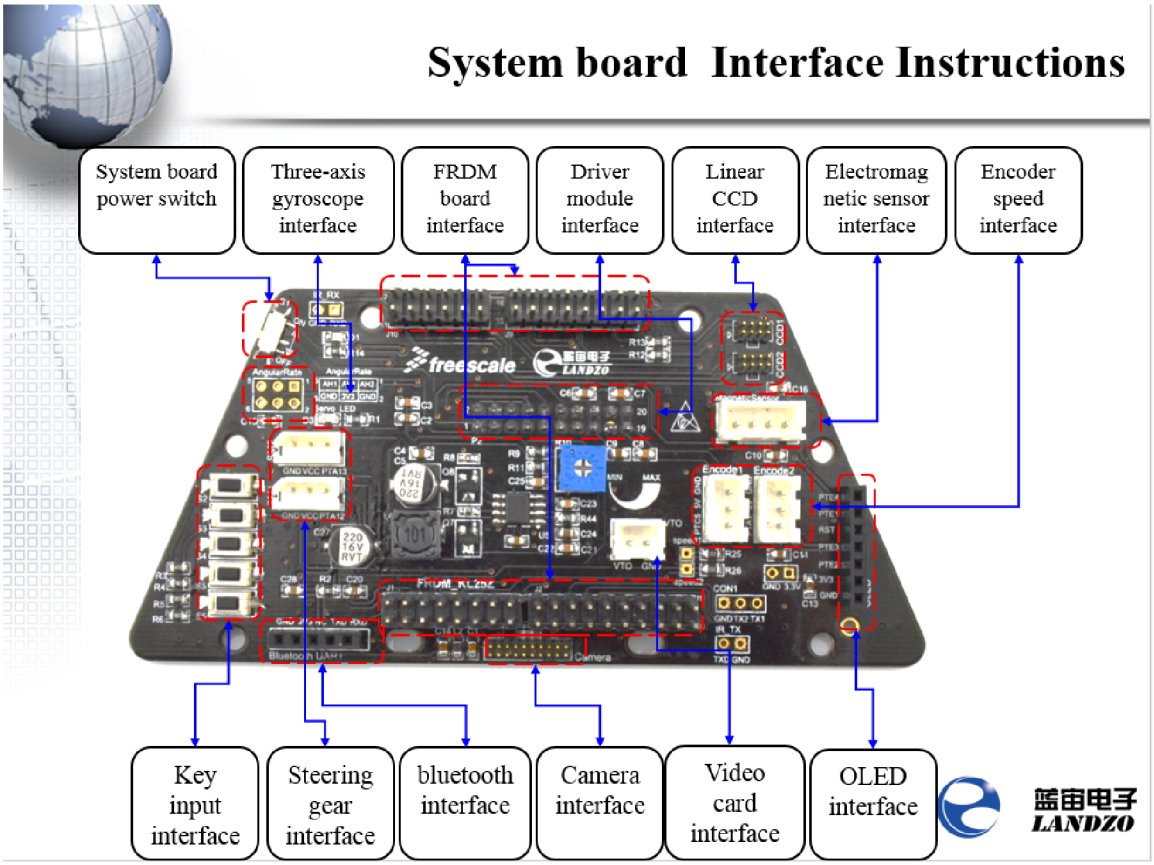
Hlavním úkolem motorové desky je řízení motorů na základě PWM signálů z mikrokontroléru a regulace napětí z hlavního napájecího zdroje na napěťové úrovně 5V a 3,3V. Maximální vstupní napětí pro motorovou desku je 12V, To je více než dostatečné pro použití s originálním akumulátorem, který je základem napájecí soustavy 3.2. Motorovou deska spolu s popisem dostupných periferií a konektorů je možné vidět na obrázku 3.14. Kompletní schéma motorové desky se nachází v příloze C.1.



Obrázek 3.14: Motorová deska spolu s popisem periferií a konektorů. Převzato z [10]

3.5.2 Systémová deska

Systémová deska se stará o zprostředkování napájení a komunikace mezi jednotlivými periferiemi systému, motorovou deskou řídícím mikrokontrolérem. Na této desce se nachází značná část periferií systému a jejich rozhraní. Systémovou desku spolu s popisem veškerých periferií a konektorů je možné vidět na obrázku 3.15. Kompletní schéma systémové desky je možné nalézt v příloze D.1.



Obrázek 3.15: Systémová deska spolu s popisem periferií a konektorů. Převzato z [10].

3.5.3 Platforma ESP32

ESP32 je série mikrokontrolérů s 3,3V logikou od firmy Espressif systems a integrovanými Wi-Fi a dual-mode bluetooth moduly. Jedná se o nástupce velmi známé a oblíbené platformy ESP8266 od stejného výrobce. ESP32 našla své uplatnění primárně ve světě IoT, avšak díky svému výkonu, který předčí i mnohé desky od firmy Arduino, se jedná o oblíbenou volbu i pro tvorbu řídicích systémů.

Důvodem výběru této platformy oproti NXP FRDM-KL25Z byla primárně obtížná integrace senzorů do existujícího a pevně daného systému modelu Alamak. Ten neposkytoval dostatek možností pro doplnění senzorů jak po hardwarové tak i softwarové stránce. Samotná deska FRDM-KL25Z neobsahovala dostatečný počet volných pinů pro osazení a řízení všech potřebných senzorů, musely by tedy být použity konektory, které jsou vyhrazeny pro jiné periferie, což by mohlo značně ztížit implementaci celého systému. Softwarové možnosti platformy NXP FRDM-KL25Z taktéž nebyly vhodné, jelikož hlavní 2 programovací prostředí: Mbed a Zephyr OS, neposkytují v základu prostředky pro komunikaci se senzory. Naproti tomu platforma ESP32 umožňuje programování například v prostředí Arduino, které obsahuje velké množství existujících prostředků pro komunikaci se senzory a umožňuje bez problému ovládat i zbylé prvky systému. Tato platforma je tedy pro tvorbu parkovacího systému navrženého v této práci mnohem vhodnější.

Série mikrokontrolérů ESP32 využívá jako své výpočetní jádro 32-bitový mikroprocesor Xtensa LX6 od firmy Tensilica. Tento mikroprocesor je v závislosti na typu mikrokontrolérů dostupný v jednojádrové nebo dvoujádrové konfiguraci a jeho frekvence může být 160 nebo 240MHz. Mikrokontrolér použitý v této práci je typ ESP-WROOM-32. Je osazen dvoujádrovou variantou mikroprocesoru Xtensa LX6 s frekvencí v rozmezí 80 – 240 MHz. Frekvenci procesoru je možné v daném rozmezí libovolně měnit a jednotlivé jádra je možné nezávisle vypínat a zapínat. Kromě hlavního mikroprocesoru obsahuje ESP-WROOM-32 ještě jeden nízko-výkonový koprocesor. který umožňuje obsluhovat přerušení, například z periferií a časovačů, ikdyž je hlavní procesor zcela vypnutý. Základní paměť mikrokontroléru tvoří 520 KB SRAM paměti a 4 MB flash paměti, která je rozšiřitelná až o 16 MB externí paměti pomocí rozhraní SPI.

Obrázek 3.16: ESP-WROOM-32. Samostatný modul (vlevo) a modul osazený na desce (vpravo).

Mikrokontrolér ESP-WROOM-32 je osazen na vývojové desce, která poskytuje možnost pro programování a napájení přes rozhraní microUSB a připojení dalších periferií. Samostatný mikrokontrolér spolu s osazením na vývojové desce je možné vidět na obrázcích 3.16. Deska obsahuje integrovaný USB-UART kontrolér CP2102 od firmy Silicon Labs a dvě tlačítka Reset a Boot. O stabilizaci napětí pro mikrokontrolér na úroveň 3,3 V se stará napěťový

regulátor AMS1117. Provozní vstupní napětí a proud vývojové desky jsou dány standardem USB a jejich hodnota je 5 V a 500 mA.

Vývojová deska poskytuje 38 pinů, z nichž 34 jsou programovatelné GPIO piny. Výhodou mikrokontroléru ESP-WROOM-32 je možnost připojit téměř jakoukoli funkci, kterou mikrokontrolér poskytuje (I^2C, PWM, SPI atp.) na libovolný ze 34 GPIO pinů. To je možné díky použití matice GPIO pinů a multiplexorů. Podrobný popis jednotlivých pinů a jejich omezení lze nalézt ve specifikaci výrobce [8]. Vývojová deska v základu poskytuje podporu dvou programovacích prostředí: Arduino a NodeMCU.

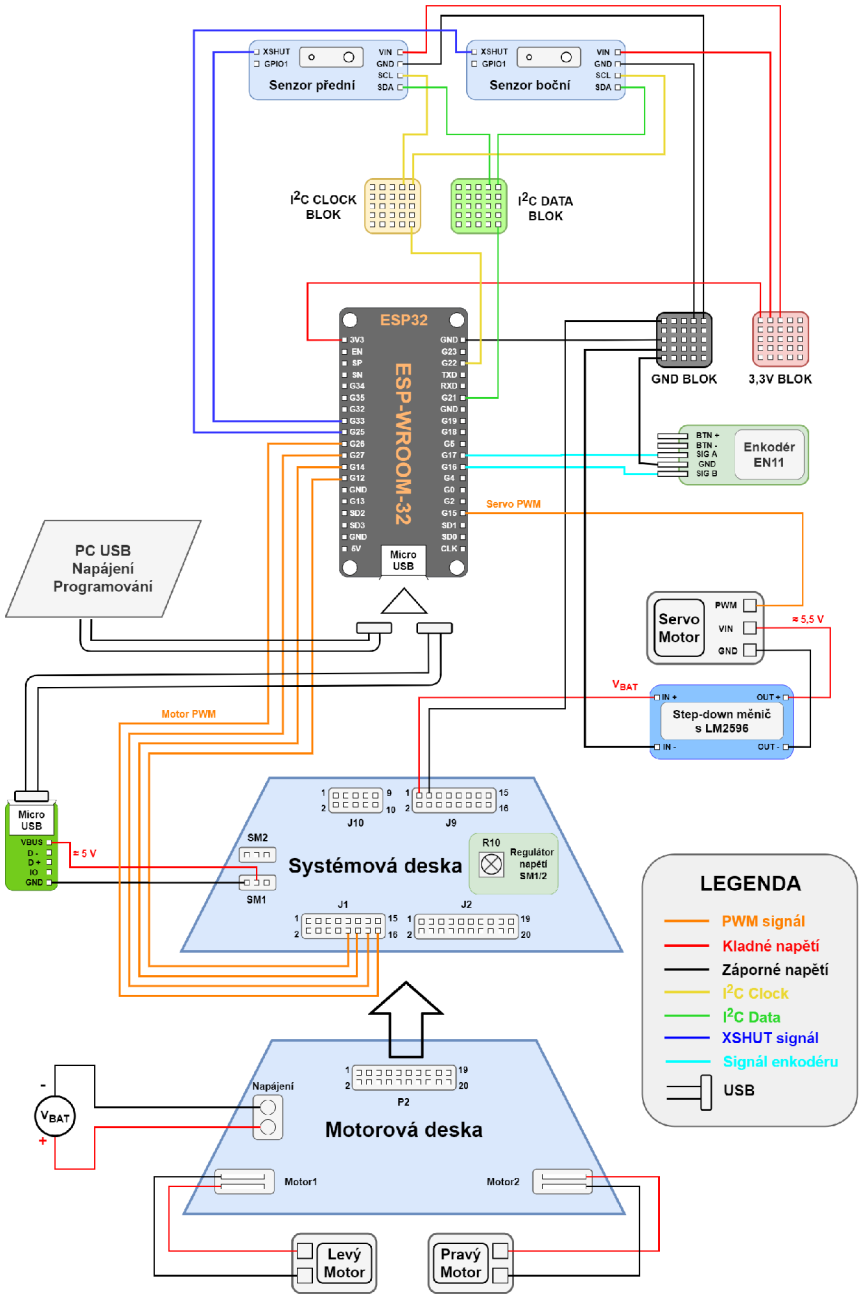
Mikrokontrolér ESP-WROOM-32 poskytuje následující periferie a rozhraní:

- **Základní periferie a rozhraní:**
  - 2x 12-bit ADC převodník
  - 2x 8-bit DAC převodník
  - 10x kapacitní dotykové senzory
  - SD/SDIO/MMC kontrolér pro podporu SD karet
  - 3 kanály 16-bit Motor-PWM
  - 16 nezávislých kanálů 80MHz 16-bit LED-PWM
  - 3x UART
  - 2x I^2C
  - 2x I^2S
  - 4x SPI (volně nebo pro externí paměť)
  - SPI/SDIO slave kontrolér
  - Hallův senzor
  - IR vysílač/přijímač
  - Ultra nízko-výkonový analogový předzesilovač
- **Bezdrátová rozhraní**
  - Wi-Fi 802.11 b/g/n
  - Bluetooth 4.2
- **Zabezpečení a ostatní technologie**
  - Secure boot
  - WEP, WPA/WPA2, PSK/Enterprise
  - Hardwarová akcelerace šifrování: AES/SHA2/RSA-4096
  - Hluboký spánek se spotřebou 5 μA
  - Probuzení

Podrobné informace o mikrokontroléru ESP-WROOM-32 je možné nalézt v dokumentaci výrobce [8]. Schéma s popisem jednotlivých částí mikrokontroléru je možné vidět v příloze E.1.

3.6 Zapojení systému

Na obrázku 3.17 je možné vidět kompletní schéma zapojení systému, spolu se všemi doplňky základního vybavení modelu NXP Cup Alamak, navrženými úpravami a periferiemi.



Obrázek 3.17: Schéma zapojení systému.

Kapitola 4

Návrh řídicího algoritmu

Tato kapitola obsahuje popis navrženého řídicího algoritmu. V samostatných sekcích se nachází vizualizace architektury algoritmu a stručný popis jeho jednotlivých částí. V další sekci je popsáno prostředí a knihovny, které byly použity k tvorbě řídicího algoritmu.

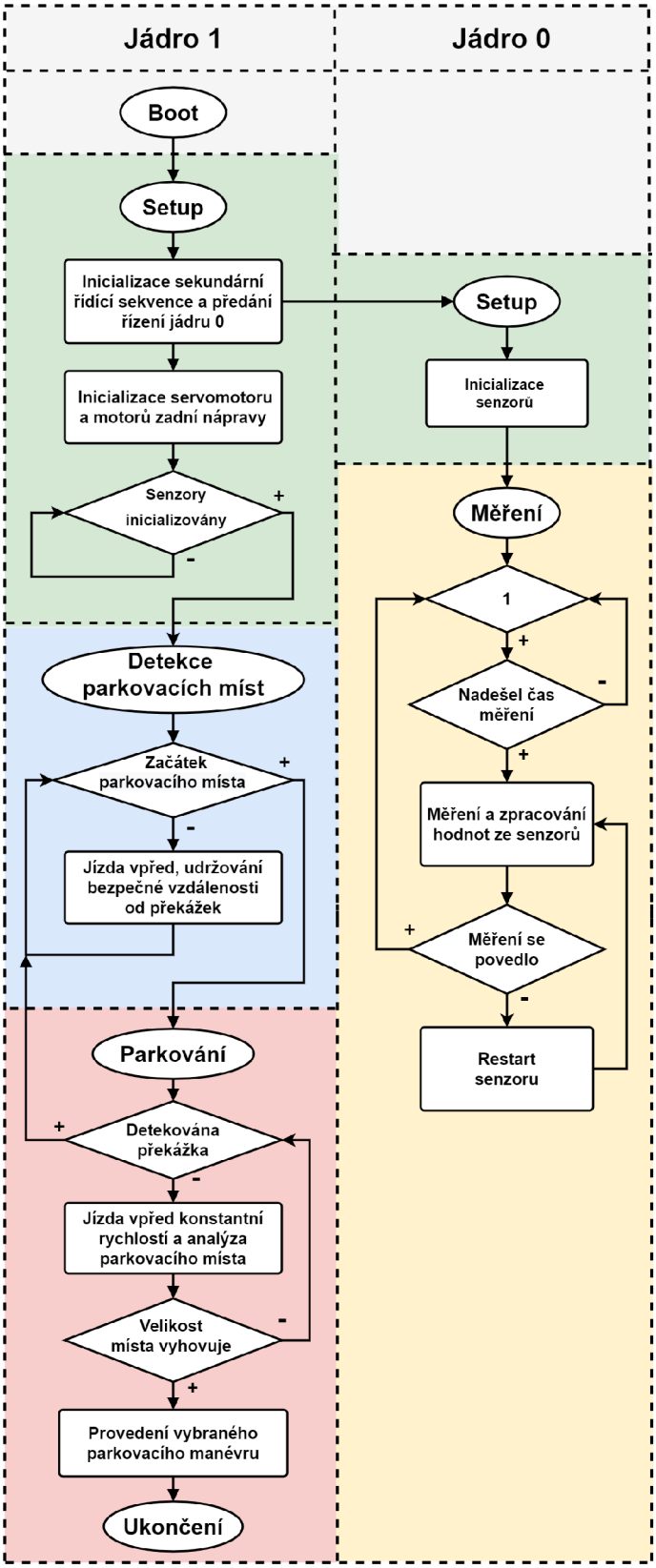
4.1 Architektura řídicího algoritmu

Při návrhu řídicího algoritmu byly zváženy možnosti, které poskytuje mikrokontrolér ESP-WROOM-32 a na jejich základě byla navržena architektura řídicího algoritmu. Tu je možné vidět na obrázku 4.1. Jelikož mikrokontrolér ESP-WROOM-32 poskytuje dvě výpočetní jádra, bylo možné některé aspekty algoritmu paralelizovat a dosáhnout tak lepšího rozložení výpočetní zátěže a rychlejší odezvy celého systému.

Cílem práce bylo realizovat funkční prototyp modelu vozidla, které bude schopno autonomně provádět vybrané parkovací manévry. K dispozici jsou 2 pevně dané parkovací manévry, konkrétně:

- parkování do řady,
- podélné parkování.

Navržený algoritmus pracuje tak, že vozidlo jízdou vpřed konstantní rychlostí vyhledává parkovací místa po své pravé straně, přičemž udržuje stanovenou bezpečnou vzdálenost od okolních objektů. Rozměry nalezených parkovacích míst jsou analyzovány a vyhovují-li potřebám zvoleného parkovacího manévru, je tento manévr proveden. K dispozici jsou celkem 4 způsoby vyhledávání a analýzy parkovacího místa: podélné, do řady, libovolně s prioritou podélného a libovolně s prioritou do řady. Při analýze místa podélně či do řady algoritmus uvažuje pouze rozměry vybraného parkovacího manévru a na jeho základě hledá vhodné parkovací místo. U vyhledávání s prioritou jsou brány v potaz rozměry obou parkovacích manévrů. Algoritmus se snaží primárně najít místo pro prioritní manévr, je-li však místo nedostatečné pro prioritní manévr, ale zároveň dostatečné pro manévr sekundární, je proveden sekundární parkovací manévr do nalezeného parkovacího místa.



Obrázek 4.1: Blokové schéma řídícího algoritmu.

4.2 Popis realizace a funkce navrženého algoritmu

Řídící algoritmus, který byl v rámci této práce navrhnut a implementován, byl vytvořen v prostředí Arduino IDE v programovacím jazyce C++ a skládá se ze 4 hlavních částí:

- inicializační část (obr. 4.1, zelená),
- část měření senzorové soustavy (obr. 4.1, žlutá),
- část vyhledávání parkovacích míst (obr. 4.1, modrá),
- parkovací část (obr. 4.1, červená).

Inicializační část

Po aktivaci modelu začne jádro 1 vykonávat program uložený v paměti mikrokontroléru. Jádro 0 se po startu mikrokontroléru nachází v režimu spánku. V inicializační části algoritmu (obrázek 4.1, zelená) se vytvářejí a inicializují jednotlivé prvky řídícího algoritmu. V prostředí Arduino IDE je tato část algoritmu reprezentována funkcí *Setup()*. Jádro 1 vytvoří sekundární řídící úlohu *Sensor_Task* a probudí jádro 0, které začne tuto úlohu paralelně vykonávat. V dalším kroku provede Jádro 1 inicializaci akčních členů systému. Je vytvořena instance třídy *Servo* a dvě instance třídy *Motor*. Konstruktory těchto tříd uvedou akční členy do výchozí pozice, tedy servomotor se nachází v nulové pozici a kola přední nápravy míří rovně a motory zadní nápravy jsou zastaveny. Poté přejde jádro 1 do smyčky, ve které čeká, dokud není hodnota proměnné *start_flag* rovna jedné. Tuto hodnotu proměnné nastaví jádro 0 ve chvíli, kdy je dokončena inicializace senzorového systému. Mezitím jádro 0 inicializuje senzorový systém vozidla. Dále je vytvořeno komunikační rozhraní se senzory přiblížení VL53L0X prostřednictvím Arduino knihovny pro I^2C komunikaci *Wire*. Poté jsou postupně inicializovány senzory přiblížení a rotační enkodér. Pokud se inicializace senzorového systému povede, jádro 0 nastaví hodnotu proměnné *start_flag* a přejde do stavu měření. Jádro 1 přechází do stavu vyhledávání parkovacího místa.

Část měření senzorové soustavy

V měřící části (obrázek 4.1, žlutá) jádro 0 řídí senzorový systém a ukládá výsledky měření. Proces měření se opakuje v nekonečné smyčce. Senzory přiblížení VL53L0X měří vzdálenost ve výchozím režimu přesnosti, v tomto režimu provede senzor jedno měření přibližně za 33 ms. Pro měření vzdálenosti jsou použity 2 senzory VL53L0X, měření bočního senzoru je považováno za primární a probíhá maximální možnou rychlostí. Měření čelního senzoru je považováno za sekundární a probíhá v intervalu daném konstantou *front_reading_interval*, jehož výchozí hodnota je 100 ms. Při volbě hodnoty tohoto intervalu je nezbytné brát v potaz čas měření senzoru *Timing_budget*. Toto je doba, za kterou senzor provede jedno měření vzdálenosti a nepřímo určuje přesnost měření senzoru. Její hodnotu je možné nastavit funkcí *setMeasurementTimingBudget* a její výchozí hodnota je 33 ms. Je tedy logické, že nemá cenu nastavovat interval kratší, než je doba provedení jednoho měření senzoru vzdálenosti. Při měření může někdy dojít k chybě komunikace, senzor přestane reagovat a je třeba jej restartovat, k tomuto účelu slouží funkce *reboot()*. Pro účely lepšího vyhodnocování a řízení se ukládá posledních 5 měření vzdálenosti z bočního senzoru a 3 měření ze

senzoru čelního. Měření ujeté vzdálenosti probíhá maximální možnou rychlostí. nejrychleji však každých 10 ms, což je hodnota zpoždění na konci každého cyklu měřicí smyčky. Aby nedocházelo k dlouhým výpadkům měření ujeté vzdálenosti v momentě, kdy probíhá měření senzorů přiblížení, které může trvat řádově desítky až stovky milisekund, je před každým měřením senzoru přiblížení rovněž provedeno měření ujeté vzdálenosti.

Část vyhledávání parkovacích míst

Ve fázi vyhledávání parkovacího místa (obrázek 4.1, modrá) jádro 1 vyhodnocuje údaje ze senzorů a s jejich pomocí řídí akční členy systému. Vyhodnocením dat o ujeté vzdálenosti řídící systém udržuje konstantní rychlost jízdy směrem vpřed. Údaje ze senzorů přiblížení slouží k udržení stanovené bezpečné vzdálenosti vozidla od nejbližší překážky, prevenci kolizí a k vyhledávání začátku parkovacího místa. Kontrola rychlosti probíhá v intervalech daných konstantou *speed_control_interval*, jehož výchozí hodnota je 100 ms. Při každé kontrole rychlosti je zjištěna ujetá vzdálenost od poslední kontroly a tato hodnota je porovnána s konstantou *speed_target*, která udává požadovanou ujetou vzdáleností mezi intervaly. Aby nebylo rozhodování prováděno na základě „ostrých hodnot“, je k požadované ujeté vzdálenosti při porovnání z obou stran přidána tolerance, daná konstantou *speed_tolerance*. Na základě výsledku porovnání je rychlost obou motorů zadní nápravy upravena tak, aby bylo dosaženo konstantní rychlosti modelu. Kontrola vzdálenosti k překážkám a prevence kolizí probíhá v intervalu stanoveném konstantou *distance_control_interval*. Její výchozí hodnota je 200 ms. Při každé kontrole je posledních 5 hodnot měření bočního senzoru zprůměrováno, aby byl snížen vliv chybných a nepřesných měření. získaný průměr boční vzdálenosti je podobně jako u kontroly rychlosti porovnán s požadovanou bezpečnou vzdáleností *desired_distance* $\pm$ *distance_tolerance*. Na základě výsledku porovnání je proveden zásah do řízení servomotoru tak, aby byla dosažena požadovaná boční vzdálenost. Údaje z předního senzoru přiblížení slouží primárně k detekci a prevenci kolizí s překážkami, které by boční senzor nedokázal detekovat. Pokud je průměr boční vzdálenosti větší než *parking_treshold*, je detekován začátek parkovacího místa a řídící algoritmus přechází do fáze parkování.

Parkovací část

Ve fázi parkování (obrázek 4.1, červená) vozidlo pokračuje konstantní rychlostí vpřed, přičemž jádro 1 vyhodnocuje jednotlivá měření bočního a předního senzoru přiblížení a ujetou vzdálenost. Pomocí hodnot bočního senzoru jsou analyzovány rozměry parkovacího místa. Tato fáze v řídícím algoritmu představuje funkci *parking()*. Pokud vozidlo urazí vzdálenost, která je větší nebo rovna minimální požadované velikosti parkovacího místa *parking_length* a nedojde přitom k detekci překážky, která by neumožňovala provést parkovací manévr, je místo považováno za vyhovující a řídící algoritmus zaparkuje do tohoto místa vybraným manévrem. Pokud dojde během analýzy rozměrů parkovacího k nalezení překážky, řídící algoritmus se okamžitě vrací do fáze vyhledávání parkovacího místa (4.1, modrá). K dispozici jsou dva pevně dané parkovací manévry, jeden pro podélné parkování a jeden pro parkování do řady. Způsoby vyhledávání a analýzy místa jsou k dispozici celkem 4: podélné, do řady, libovolně s prioritou podélného a libovolně s prioritou do řady. Výběr způsobu vyhledávání a analýzy parkovacího místa jo proveden nastavením proměnné *parking_type* na požadovanou hodnotu.

4.3 Použité nástroje a knihovny

Pro účely řízení akčních členů modelu byly vytvořeny knihovny *Servo* a *Motor*, které obsahují stejnojmenné třídy. Knihovny jsou implementovány v jazyce C++. Pro řízení motorů i serva je třeba vytvořit a nakonfigurovat řadu PWM signálů. K tomuto účelu slouží kontrolér LED PWM mikrokontroléru ESP-WROOM-32 [8] a jeho funkce *ledcSetup(channel, frequency, resolution)* a *ledcAttachPin(pin, channel)*, které umožňují vytvořit až 16 kanálů PWM signálů. K řízení střídy PWM signálů slouží funkce *set_duty_cycle(unsigned int duty_cycle)*. Směr otáčení motorů je možné řídit funkcí *set_direction()* a k jejich zastavení slouží funkce *stop()*. Rozlišení a frekvenci pwm signálů je možné měnit funkcemi *set_resolution* a *set_frequency*.

Důležité parametry, konstanty a proměnné řídicího algoritmu se nacházejí v konfiguračním souboru *alamak_config.h*. Volbou hodnot těchto parametrů je možné měnit jednotlivé aspekty řídicího algoritmu.

Pro řízení laserových senzorů vzdálenosti VL53L0X existuje v prostředí Arduino IDE celá řada knihoven. Mezi nejpoužívanější patří knihovny od Adafruit[1] a Pololu [2]. V této práci je použita knihovna Pololu, která neposkytuje takové možnosti jako Adafruit, ale její použití je jednodušší a pokročilejší funkcionalita knihovny Adafruit nebyla pro potřeby této práce nezbytná. Knihovna Pololu staví na originálním API výrobce senzoru VL53L0X a umožňuje měření vzdálenosti formou jednotlivých měření nebo formou měření, která se automaticky opakují ve stanoveném intervalu. Kromě samotné komunikace se senzorem knihovna umožňuje:

- Měnit a číst I^2C adresu senzoru,
- Zapisovat a číst registry senzoru,
- Nastavit práh detekce objektů,
- Nastavit trvání měření v milisekundách,
- Nastavit dobu odezvy senzoru (timeout),
- Detekovat výpadek senzoru (timeout).

Pro měření otáček rotačního enkodéru je použita knihovna ESP32Encoder[3]. Tato knihovna byla speciálně vytvořena a optimalizována pro použití s vývojovými deskami ze série ESP32. Narozdíl od běžných knihoven, které k měření nejčastěji používají přerušení, tato knihovna používá dvoukanálový 16-bitový čítač impulzů, který je součástí každého mikrokontroléru ESP32 a díky tomu zbytečně nezatěžuje systém přerušení mikrokontroléru. Signální piny enkodéru jsou připojeny na slabší pull-down rezistory, aby nedocházelo k chybnému čtení pozice, pokud je enkodér odpojen. Tato knihovna umožňuje současně obsluhovat až 10 enkodérů.

[1] Knihovna Adafruit pro komunikaci se senzorem vzdálenosti VL53L0X: https://github.com/adafruit/Adafruit_VL53L0X.

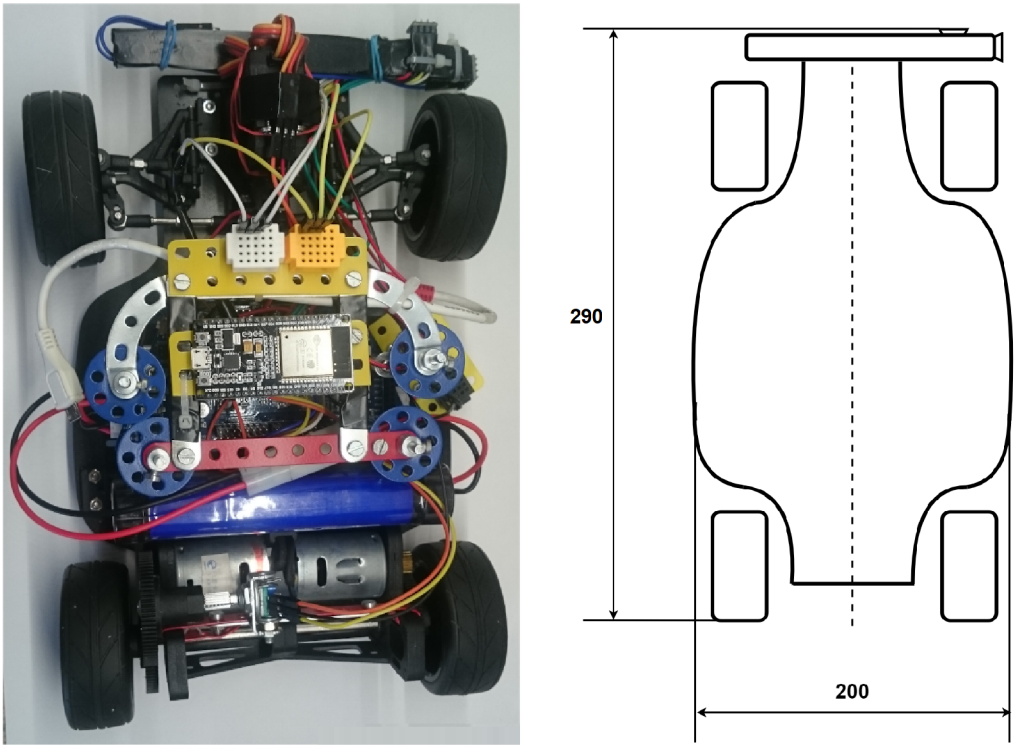
[2] Knihovna Pololu pro komunikaci se senzorem vzdálenosti VL53L0X: https://github.com/pololu/vl53l0x-arduino.

[3] Knihovna pro komunikaci s inkrementálním rotačním enkodérem EN11: https://github.com/madhephaestus/ESP32Encoder/.

Kapitola 5

Experimenty a vyhodnocení výsledků

V této kapitole jsou prezentovány a vyhodnoceny výsledky experimentů, které byly provedeny s modelem vozidla NXP Cup Alamak a jeho senzory. Provedené experimenty parkovacích manévrů slouží ke stanovení minimální velikosti parkovacích míst, které vyhledává řídicí algoritmus. Zbylé experimenty slouží jako ukázka funkce konkretních částí řídicího algoritmu a ověření přesnosti vybraných senzorů. Model vozidla, se kterým byly provedeny experimenty, je možné spolu s jeho rozměry vidět na obrázcích 5.1.



Obrázek 5.1: Experimentální model vozidla. Obrázek vozidla (vlevo), diagram popisující rozměry vozidla (vpravo).

5.1 Přesnost senzorů vzdálenosti VL53L0X

V následujících experimentech je představena a vyhodnocena přesnost laserových senzorů vzdálenosti VL53L0X. Veškeré experimenty probíhaly tak, že senzor byl pevně usazen na rovném podkladu a naproti němu byl umístěn objekt o velikosti 10x10 cm. Tento objekt byl poté posouván do určených vzdáleností a výsledky byly zaznamenány do tabulek v párech reálná-naměřená vzdálenost. Vzdálenosti jsou uváděny v milimetrech.

Přesnost režimů měření vzdálenosti

V tomto experimentu byly porovnány jednotlivé režimy měření vzdálenosti, jež poskytuje knihovna Pololu, která je použita pro řízení senzorů vzdálenosti VL53L0X v této práci. Tato knihovna poskytuje 3 režimy měření vzdálenosti:

- *fast-mode* - měření vzdálenosti vysokou rychlostí na úkor přesnosti, jedno měření trvá přibližně 20 milisekund.
- *standard-mode* - výchozí režim, jedno měření trvá přibližně 33 milisekund. Kompromis mezi rychlostí a přesností.
- *high-accuracy* - měření vzdálenosti s vysokou přesností na úkor rychlosti, jedno měření trvá přibližně 200 milisekund.

Výsledky měření vzdálenosti pro jednotlivé režimy je možné vidět v tabulce 5.1. Ve všech případech je možné vidět, že senzor má tendenci měřit nižší vzdálenost, než-li je reálná vzdálenost objektu, pokud se tento objekt nachází ve vzdálenosti menší než 100 mm. Toto je velmi patrné hlavně u režimu měření vysokou rychlostí.

Měření vzdálenosti: režim vysoké přesnosti										
Reálná vzdálenost [mm]	20	40	60	80	100	150	200	300	400	500
Naměřená vzdálenost [mm]	17	34	57	78	99	148	202	305	398	502

Měření vzdálenosti: výchozí režim										
Reálná vzdálenost [mm]	20	40	60	80	100	150	200	300	400	500
Naměřená vzdálenost [mm]	18	32	54	74	96	147	199	302	394	511

Měření vzdálenosti: režim vysoké rychlosti										
Reálná vzdálenost [mm]	20	40	60	80	100	150	200	300	400	500
Naměřená vzdálenost [mm]	15	27	50	69	91	146	193	307	388	484

Tabulka 5.1: Přesnost měření vzdálenosti senzoru VL53L0X v závislosti na zvoleném měřicím režimu.

Přesnost měření v závislosti na okolních světelných podmínkách

Při tomto experimentu byla zkoumána přesnost měření vzdálenosti při měnících se světelných podmínkách v okolí senzoru. Experimenty byly provedeny ve výchozím měřicím režimu, jelikož tento režim je použit i v navrženém řídicím algoritmu. Některé senzory, které využívají k měření vzdálenosti světelné záření, mohou být náchylné na okolní světelné podmínky. Byly provedeny 2 druhy experimentů:

- vysoká intenzita okolního osvětlení,
- nízká intenzita okolního osvětlení.

V případě experimentů s vysokou intenzitou okolního osvětlení byl přímo na senzor namířen silný zdroj světelného záření ve vzdálenosti 3 cm. Experimenty s nízkou intenzitou okolního osvětlení probíhaly v prostředí, kdy se senzor i měřený objekt nacházely téměř v absolutní tmě. V obou případech nedošlo u senzoru VL53L0X k žádnému měřitelnému zhoršení přesnosti. Výsledky experimentů je možné vidět v tabulce 5.2.

Měření vzdálenosti: protisvětlo										
Reálná vzdálenost [mm]	20	40	60	80	100	150	200	300	400	500
Naměřená vzdálenost [mm]	18	36	56	75	94	148	203	297	397	511

Měření vzdálenosti: tma										
Reálná vzdálenost [mm]	20	40	60	80	100	150	200	300	400	500
Naměřená vzdálenost [mm]	18	37	54	77	95	147	201	300	399	508

Tabulka 5.2: Přesnost měření vzdálenosti senzoru VL53L0X v závislosti na okolních světelných podmínkách.

Přesnost měření v závislosti na materiálu překážky

V tomto experimentu byla pozorována přesnost senzoru VL53L0X v závislosti na materiálu měřeného objektu. Měření byla provedena ve výchozím režimu. Postupně byly otestovány objekty různých barev a materiálů, například: plast, dřevo, papír, hliník, kůže. U žádného z těchto materiálů nebyla pozorována prokazatelná změna přesnosti měření.

Jediný experiment, při kterém došlo k významnému zhoršení přesnosti měření vzdálenosti, bylo při použití objektu z čirého skla. V tomto případě byl senzor schopen u nižších vzdáleností detekovat přítomnost objektu, ale nebyl s rozumnou přesností schopen určit jeho vzdálenost. Při vzdálenosti nad 100 mm už senzor nebyl schopen přítomnost takového objektu detekovat vůbec. Výsledky tohoto experimentu je možné vidět v tabulce 5.3.

Měření vzdálenosti: čiré sklo										
Reálná vzdálenost [mm]	20	40	60	80	100	150	200	300	400	500
Naměřená vzdálenost [mm]	0	3	17	52	64	–	–	–	–	–

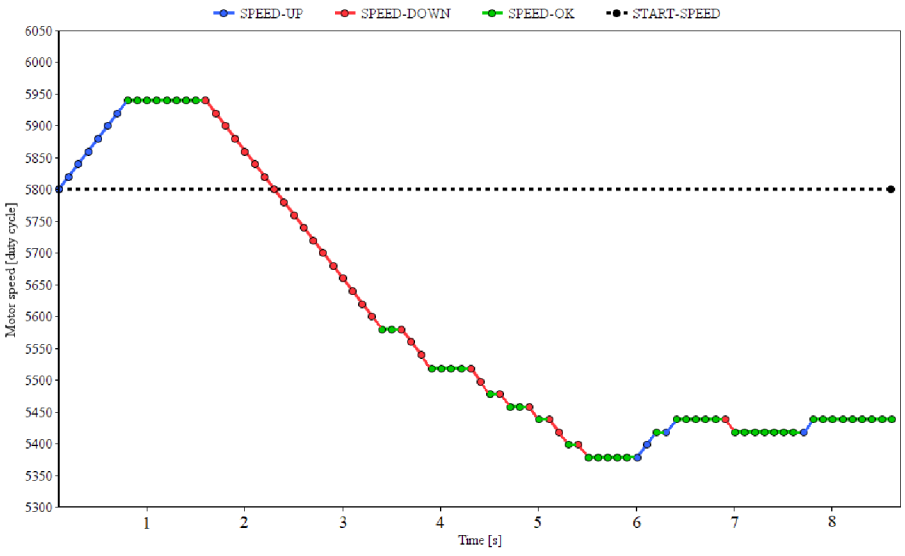
Tabulka 5.3: Přesnost měření vzdálenosti u objektu z čirého skla.

5.2 Jízda konstantní rychlostí

Cílem tohoto experimentu bylo otestovat funkci automatické kontroly rychlosti modelu. Model vozidla byl postaven na rovnou plochu bez překážek. Experiment byl proveden na povrchu s vysokou přilnavostí, aby nedocházelo k prokluzu kol hnací nápravy. Požadovaná rychlost modelu byla stanovena přibližně na 20 $cm \cdot s^{-1}$. Interval kontroly rychlosti byl nastaven na 100 milisekund. Při experimentu byla ujeta vzdálenost přibližně 2 metrů.

Výsledek experimentu je možné vidět na grafu 5.2. Na ose x je vyznačen čas experimentu a na ose y je střída PWM signálu, který určuje výkon motorů. Křivka modré, zelené a červené barvy představuje výkon motorů v daném časové okamžiku. Barva této křivky představuje rozhodnutí řídicího algoritmu o změně výkonu motorů. V modrých částech křivky řídicí systém zvyšuje výkon motorů. V zelených částech zůstává výkon motorů stejný. V červených částech křivky řídicí systém snižuje výkon motorů. Změnu výkonu motorů provádí řídicí systém na základě rozdílu reálné a požadované rychlosti modelu.

V prvních třech vteřinách experimentu je možné vidět velký zákmit rychlosti motorů, to je způsobeno tím, že rozjezd vozidla probíhá postupně a trvá určitou dobu. Proto řídicí systém zpočátku zvyšuje výkon motorů a po rozjetí modelu jej opět prudce sníží. Tento zákmit je možné částečně zmírnit vhodnou volbou počátečního výkonu motorů (v grafu 5.2 přerušovaná čára černé barvy), ale není jej možné zcela eliminovat. Mezi třetí a sedmou vteřinou experimentu dochází k pozvolnému ustálení rychlosti motorů a reálná rychlost modelu se začíná blížit požadované rychlosti. Od sedmé vteřiny dále už se rychlost modelu ustálila na hodnotě blízké požadované rychlosti a k zásahům do výkonu motorů už nedochází téměř vůbec.



Obrázek 5.2: Graf automatické kontroly rychlosti modelu.

5.3 Parkovací experimenty

Pro stanovení minimálních rozměrů parkovacího místa pro jednotlivé parkovací manévry byly provedeny sady experimentů. Každá sada představuje provedení deseti experimentů pro jednu parkovací konfiguraci. Parkovací konfigurací se rozumí nastavení parkovacího místa na konkrétní šířku W a délku L pro daný parkovací manévr. Vizualizaci parkovacích experimentů je možné vidět na obrázcích 5.3 a 5.4. Experiment se skládá ze 3 fází:

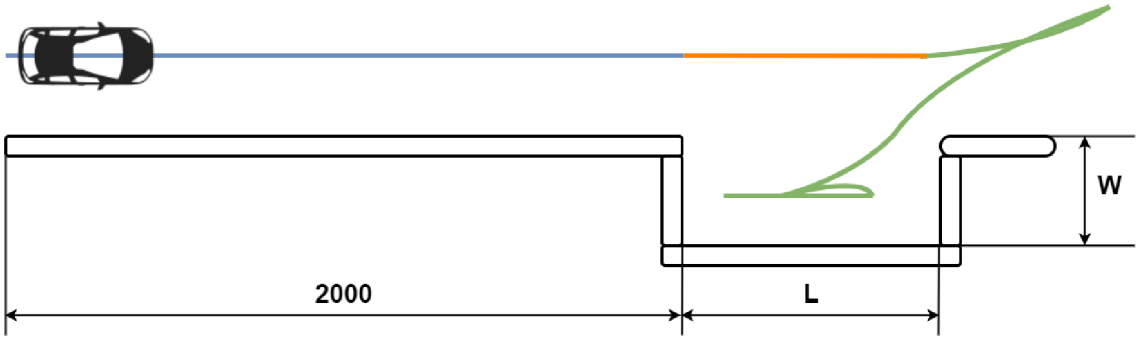
- Fáze 1 (modrá část) - jízda vpřed konstantní rychlostí a udržování předepsané vzdálenosti od okolních objektů,
- Fáze 2 (oranžová část) - identifikace parkovacího místa, jeho rozměrů,
- Fáze 3 (zelená část) - vyhodnocení a případně provedení příslušného parkovacího manévru.

V první fázi musí model vozidla urazit vzdálenost 2 metrů, přičemž musí udržovat konstantní rychlost a vzdálenost od okolních objektů. Ve fázi 2 musí model vozidla zaregistrovat začínající parkovací místo, analyzovat a jeho rozměry. Ve fázi 3 je proveden zvolený parkovací manévr. Rozhodnutí, který manévr bude proveden, se děje automaticky na základě analýzy rozměrů parkovacího místa.

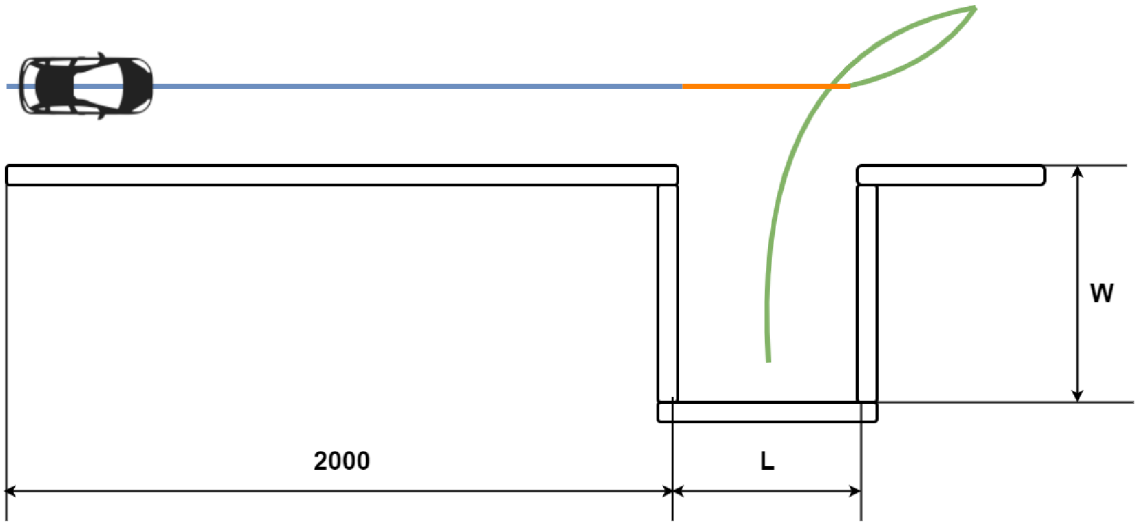
Parkovací experiment je prohlášen za neúspěšný pokud:

- v průběhu experimentu dojde ke kolizi,
- vozidlo není schopné detekovat parkovací místo,
- vozidlo po dokončení parkovacího manévru vyčnívá z profilu parkovacího místa o více než 20 mm,
- vozidlo výrazně přesáhne předepsanou rychlost nebo vzdálenost k okolním objektům.

Jinak je experiment prohlášen za úspěšný. Výsledky experimentů pro jednotlivé parkovací konfigurace a manévry je možné vidět v tabulkách 5.4 a 5.5.



Obrázek 5.3: Vizualizace experimentu podélného parkování.



Obrázek 5.4: Vizualizace experimentu parkování do řady.

Podélné parkování								
W [mm]	230							
L [mm]	400	410	420	430	440	450	460	470
Úspěšné experimenty	2	4	6	6	7	9	10	10
Neúspěšné experimenty	8	6	4	4	3	1	0	0
Úspěšnost [%]	20	40	60	60	70	90	100	100

Tabulka 5.4: Výsledky experimentů pro podélné parkování.

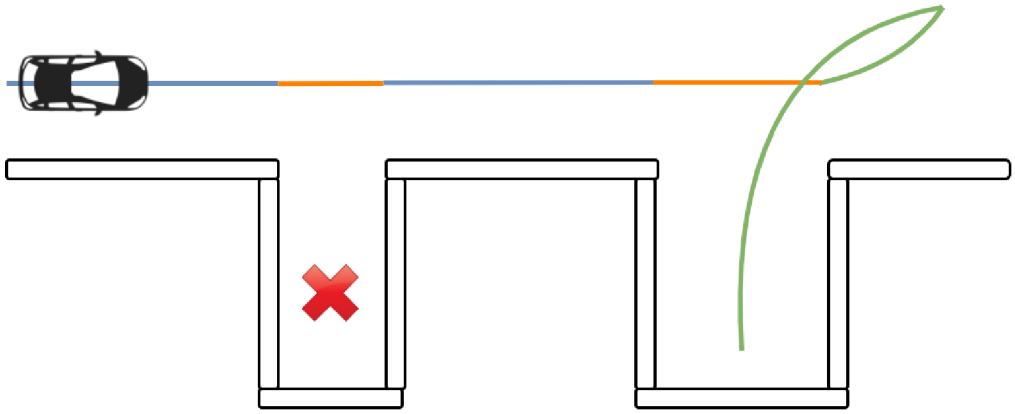
Parkování do řady							
W [mm]	320						
L [mm]	240	250	260	270	280	290	300
Úspěšné experimenty	4	6	7	9	9	10	10
Neúspěšné experimenty	6	4	3	1	1	0	0
Úspěšnost [%]	40	60	70	90	90	100	100

Tabulka 5.5: Výsledky experimentů pro parkování do řady

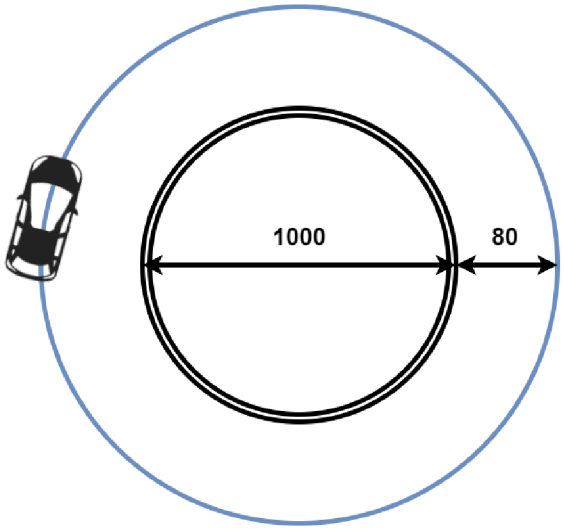
Výsledky experimentů ukázaly, že model je schopen se přizpůsobit okolním podmínkám a autonomně vyhledávat a analyzovat parkovací místa ve svém okolí. Do takto nalezených parkovacích míst je navíc schopen zaparkovat s vysokou úspěšností, při volbě vhodných minimálních rozměrů hledaných parkovacích míst. Hlavním důvodem neúspěchu parkovacích manévrů u parkovacích míst menších rozměrů byl fakt, že měření vzdálenosti ani udržování bezpečné vzdálenosti od okolních objektů nikdy nebude probíhat se 100% přesností a vozidlo se tedy při každém zahájení parkovacího manévru bude nacházet v různých vzdálenostech od okolních překážek. V případě větších parkovacích míst toto není problém, jelikož poskytují dostatečnou rezervu. U parkovacích míst menších rozměrů tato rezerva není a nároky na přesnost počáteční pozice parkovacího manévru jsou mnohem vyšší.

Kromě experimentů, které byly podrobně popsány výše byla provedena i řada dalších experimentů, například na obrázku 5.5 je možné vidět experiment, při kterém musí vozidlo

vynechat nevhodná parkovací místa. Na obrázku 5.6 je možné vidět experiment, při kterém musí vozidlo udržovat předepsanou vzdálenost od překážky ve tvaru kruhu. Videa těchto a dalších experimentů je možné najít na adrese v příloze[1].



Obrázek 5.5: Vizualizace experimentu parkování, při kterém vozidlo musí zvolit vhodné místo k zaparkování.



Obrázek 5.6: Vizualizace experimentu jízdy podél překážky kruhového tvaru.

[1]Videa, na kterých jsou ukázky jednotlivých experimentů, jsou dostupná na adrese: https://www.youtube.com/playlist?list=PLs65kInUaOoRcaOoELy2Pz5u8ih0II5wE.

Kapitola 6

Závěr

Práce obsahuje úvod do problematiky tvorby autonomních parkovacích systémů. V jednotlivých kapitolách byly podrobněji rozebrány techniky, které se při tvorbě takových systému v praxi používají. Byl sestaven model vozidla na platformě NXP Cup Alamak. Jednotlivé části tohoto modelu byly podrobně popsány a tento model byl doplněn vhodnými senzory za účelem realizace autonomního parkovacího systému. Na základě analyzovaných studií a možností modelu NXP Cup Alamak je navržen řídicí algoritmus, který umožňuje tomuto modelu autonomně vyhledávat parkovací místa, tyto místa analyzovat a jsou-li vyhovující, provést parkování jedním ze 2 navržených parkovacích manévrů. Byly provedeny experimenty, které ověřují schopnosti modelu, řídicího algoritmu a navržených senzorů a ukazují, že model je schopen parkovat s vysokou úspěšností s použitím obou parkovacích manévrů.

V rámci dalšího vývoje by bylo možné vybavit celý systém vetším počtem různých senzorů a implementovat algoritmus pro tvorbu dynamických parkovacích trajektorií a docílit tak vyššího stupně autonomie.

Literatura

[1] *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. Piscataway, USA: IEEE, 1991. ISBN 978-0-7381-4768-0.

[2] Aye, Y., Watanabe, K., Maeyama, S. a Nagai, I. An automatic parking system using an optimized image-based fuzzy controller by genetic algorithms. *Artificial Life and Robotics*. Tokyo: Springer Japan. 2017, sv. 22, č. 1, s. 139–144. ISSN 1433-5298.

[3] Brian Froisy, J. Model predictive control: Past, present and future. *ISA Transactions*. 1994, sv. 33, č. 3, s. 235 – 243. ISSN 0019-0578.

[4] Catsoulis, J. *Designing embedded hardware*. 2nd ed. O´Reilly, 2005. ISBN 0-596-00755-8.

[5] Chenini, H., Dèrutin, J. P. a Tixier, T. Fast parking control of mobile robot based on multi-layer neural network on homogeneous architecture. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)* [online]. 2013, s. 1–10. ISBN 978-1-4673-6129-3. Dostupné z: https://ieeexplore.ieee.org/document/6706922.

[6] Chirca, M., Chapuis, R. a Lenain, R. Autonomous Valet Parking System Architecture. In: *The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings.* 2015, 2015-, s. 2619–2624. ISBN 9781467365956.

[7] Demirli, K. a Khoshnejad, M. Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller. *Fuzzy Sets and Systems*. 2009, sv. 160, č. 19, s. 2876 – 2891. ISSN 0165-0114.

[8] Espressif Systems. *Dokumentace k mikrokontroléru ESP-WROOM-32* [online]. Verze 2.9. Espressif Systems, 2019 [cit. 2020-05-20]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[9] Nie, Y., Xu, K., Chen, H. a Peng, L. Crowd-parking: A New Idea of Parking Guidance Based on Crowdsourcing of Parking Location Information from Automobiles. In: *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society* [online]. IEEE, 2019, s. 2779–2784. ISBN 978-1-7281-4878-6. Dostupné z: https://ieeexplore.ieee.org/document/8927134.

[10] NXP. Landzo car model. *NXP Cup* [online]. Revidováno 9. 2019 [cit. 2020-05-20]. Dostupné z: https://nxp.gitbook.io/nxp-cup/developer-guide/landzo-car-model.

[11] Páral, J. *Autonomní řízení robota po jednoduché dráze* [online]. Brno, CZ, 2019. [cit. 2020-05-20]. Diplomová práce. Masarykova univerzita, Fakulta informačních technologií. Dostupné z: https://is.muni.cz/th/ze9z5/thesis.pdf.

[12] Xiong, X. a Choi, B.-J. Design of Genetic Algorithm-Based Parking System for an Autonomous Vehicle. In: *Control and Automation, and Energy System Engineering: International Conferences, CA and CES3 2011, Held as Part of the Future Generation Information Technology Conference, FGIT 2011, in Conjunction with GDC 2011, Jeju Island, Korea, December 8-10, 2011. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, sv. 256, s. 50–57. ISBN 9783642260094.

[13] Ye, H., Jiang, H., Ma, S., Tang, B. a Wahab, L. Linear model predictive control of automatic parking path tracking with soft constraints. *International Journal of Advanced Robotic Systems.* London, England: SAGE Publications. 2019, sv. 16, č. 3. ISSN 1729-8814.

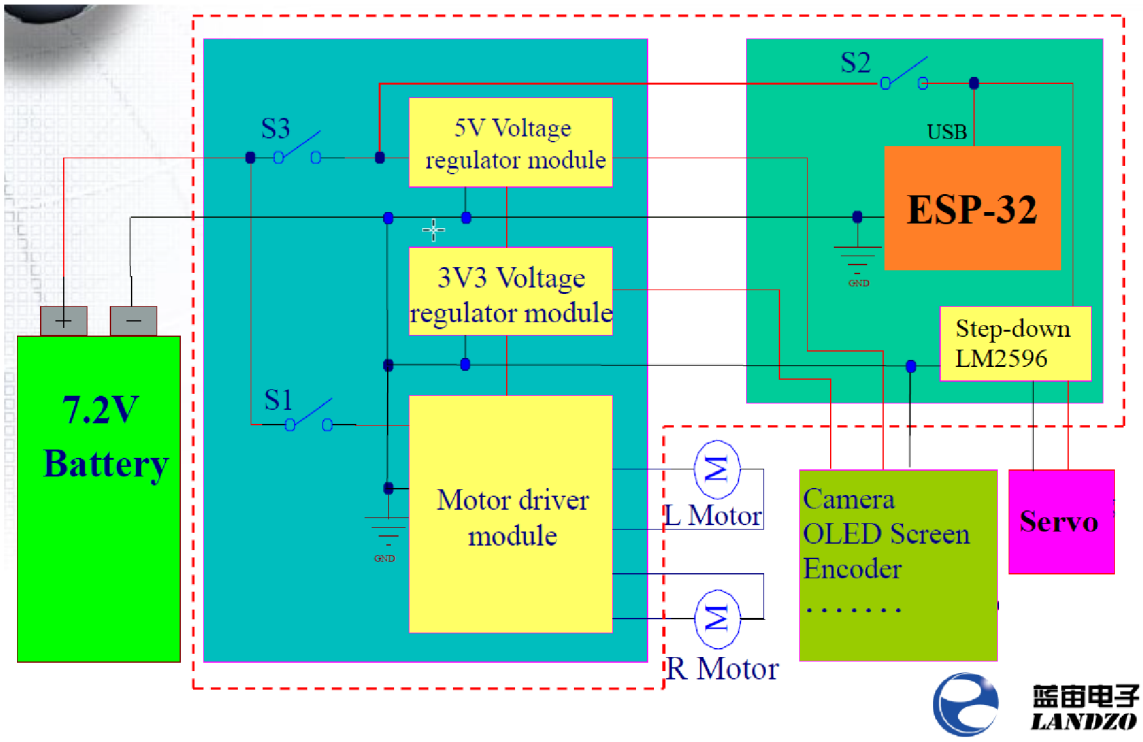
Příloha A

Struktura a obsah přiloženého paměťového média

- /Algorithm/ - adresář se zdrojovými kódy a knihovnami navrženého algoritmu.
- /Latex/ - adresář se zdrojovými kódy textu práce.
- /Text/ - adresář s textem práce ve formátu pdf

Příloha B

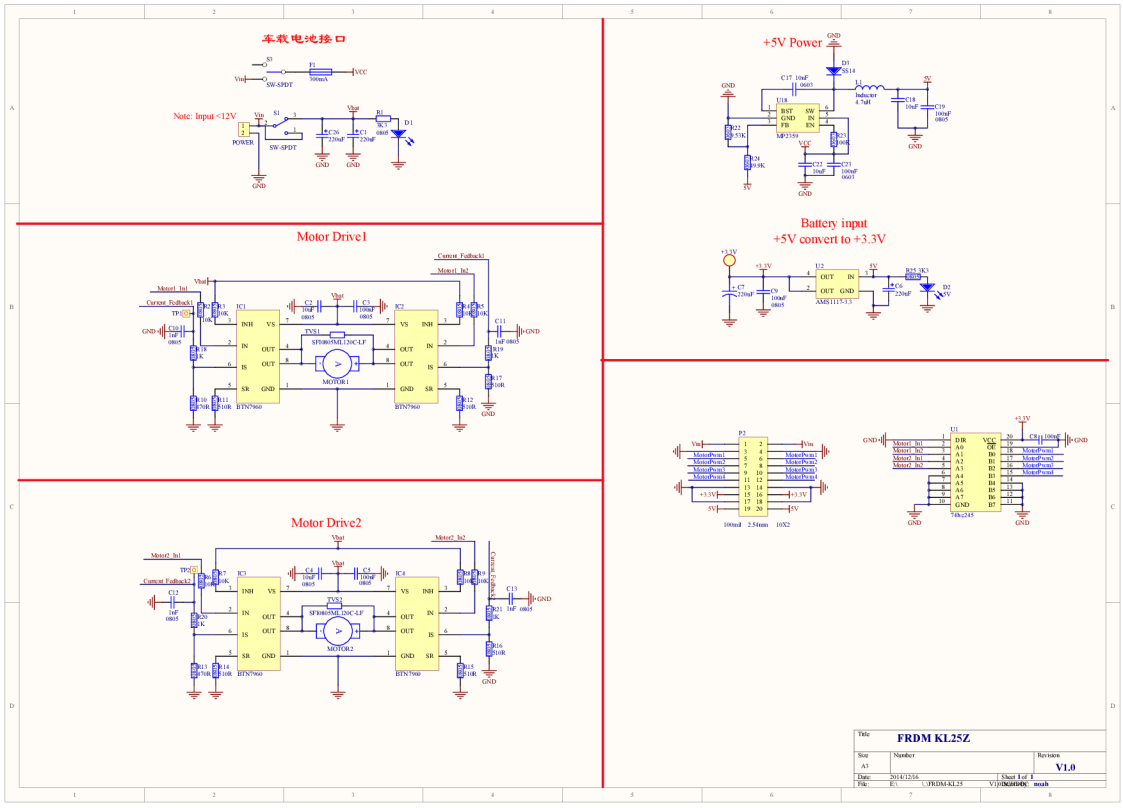
Schéma napájecí soustavy modelu



Obrázek B.1: Úplné schéma napájecí soustavy. Znázorňuje zapojení regulátorů a jednotlivých periferií, spolu s napěťovými úrovněmi.

Příloha C

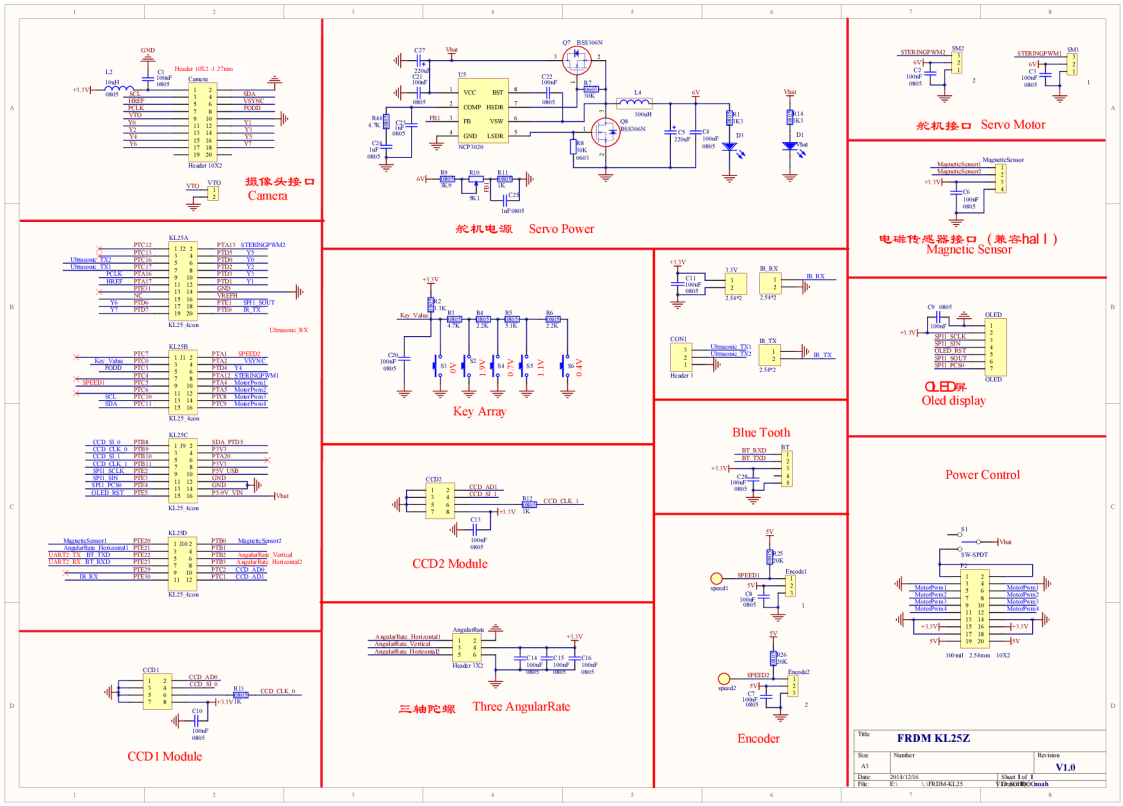
Schéma motorové desky



Obrázek C.1: Úplné schéma motorové desky. Znázorňuje zapojení jednotlivých částí motorové desky, rozložení a pojmenování jednotlivých pinů a názvy a hodnoty obvodových součástek.

Příloha D

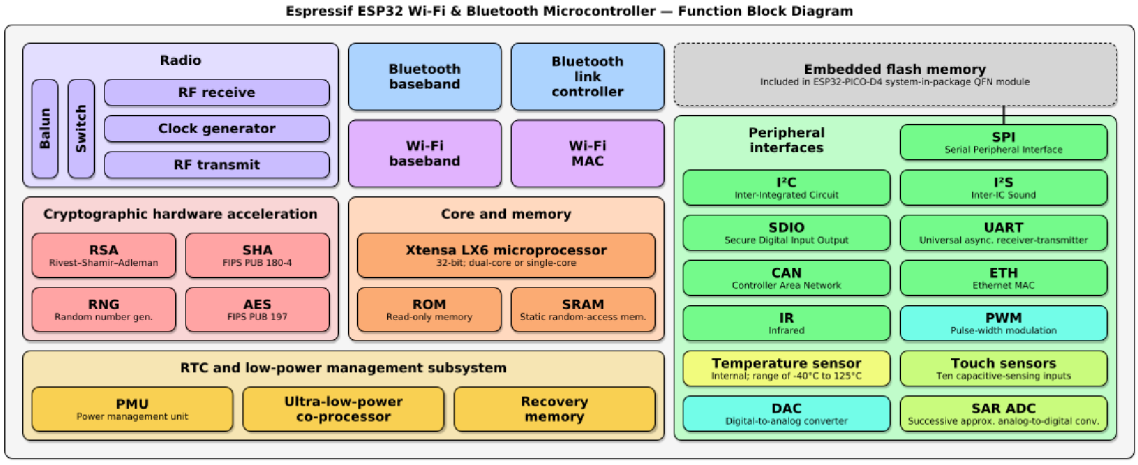
Schéma systémové desky



Obrázek D.1: Úplné schéma systémové desky. Znázorňuje zapojení jednotlivých částí systémové desky, rozložení a pojmenování jednotlivých pinů a názvy a hodnoty obvodových součástek.

Příloha E

Schéma mikrokontrolérů ESP32



Obrázek E.1: Schéma mikrokontroléru ESP32. Znázorňuje jednotlivé funkce a rozhraní mikrokontrolérů ze série ESP32.