

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Koncept vozidla autonomně řízeného mikroprocesorem
v reálném čase
Bakalářská práce

Autor: Miroslav Škoda
Studijní obor: AI3

Vedoucí práce: Ing. Pavel Blažek, Ph.D.

Hradec Králové

duben 2020

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

vlastnoruční podpis

V Hradci Králové dne:

Poděkování:

Děkuji vedoucímu bakalářské Ing. Pavlu Blažkovi Ph.D. za cenné rady a hlavně trpělivost.

Anotace

Práce se zabývá výrobou autonomně řízeného modelu automobilu v měřítku 1:28. V první části se zaměřuje na základní informace a výběr stěžejních stavebních prvků. Druhá část obsahuje samotný výběr metod pro snímání pohybu, charakteristikám pohybu vozidla a stanovení nejlepšího řešení s ohledem na měnící se okolní podmínky na dráze. Na tuto část pak navazuje implementace řešení, vývoj a testování modelu pro ověření vhodnosti vybrané koncepce.

Výsledkem práce je autonomní model automobilu, který je schopen na základě snímaných dat v reálném čase vykonávat jízdu po speciální závodní dráze určené pro modely 1:28. Konkrétní uplatnění a přínos práce je zejména pro vývojáře v oblasti autonomního řízení, firmy anebo také modeláře, zabývající se řízením vozidel bez vnějšího zásahu člověka.

Annotation

Title: Concept of real-time microprocessor-controlled vehicle

The work deals with the production of an autonomously controlled 1:28 scale model car. The first part focuses on basic information and selection of key building elements. The second part details the selection of the motion sensing method, the vehicle's movement characteristics and the determination of the best solution with respect to the changes in ambient track conditions. For this part, you can then use the chosen solution, develop and test of the model to find the suitability of the selected concept.

The result of the work is an autonomous model car which is able to drive on a special track for 1:28 models based on real-time data. The specific application and benefit of the work is especially for developers in the field of autonomous driving, companies or modelers, dealing with driving vehicles without external human intervention.

Obsah

1	Úvod.....	1
2	Cíl práce.....	2
3	Metodika zpracování.....	3
4	Teorie autonomního pohybu	4
4.1	Úvod do autonomního pohybu.....	4
4.1.1	Teoretické základy.....	5
4.1.2	Dosavadní vývoj a trendy.....	6
4.1.3	Možnosti pohybu.....	9
4.2	Modely vozidel.....	10
4.2.1	Obecné využití.....	11
4.2.2	Požadované vlastnosti.....	11
4.2.3	Chování vozidla.....	12
4.2.4	Varianty provedení.....	12
4.3	Technologie snímání okolí.....	15
4.3.1	Senzory indukční.....	15
4.3.2	Senzory kapacitní.....	16
4.3.3	Senzory optické.....	17
4.3.4	Senzory ultrazvukové.....	18
4.3.5	Senzory radarové.....	19
4.3.6	Kamerové snímání.....	20
4.3.7	Lidar vs. Radar.....	21
4.3.8	Shrnutí variant a rozhodnutí.....	22
4.4	Základy regulace.....	22
4.4.1	ON / OFF regulace.....	23
4.4.2	PID regulace.....	24

4.4.3	Shrnutí variant a rozhodnutí	25
4.5	Popis řídicí elektroniky.....	25
4.5.1	Napájení	26
4.5.2	Ovládání modelu.....	26
4.5.3	Mikro kontrolér.....	26
4.5.4	Periferie.....	27
4.6	Popis mechanické části.....	28
4.6.1	Mechanismus přední nápravy.....	28
4.6.2	Rychlost automobilu	29
5	Předpoklady a realizace modelu	29
5.1	Výchozí podmínky	29
5.1.1	Velikost modelu.....	30
5.1.2	Typ a vlastnosti dráhy	30
5.1.3	Způsob hodnocení.....	31
5.2	Analýza dráhy.....	32
5.2.1	Rychlost.....	32
5.2.2	Plynulost a dynamika jízdy.....	32
5.2.3	Změna pohybu	33
5.2.4	Ideální trajektorie.....	33
5.3	Návrh realizace na základě teorie	33
5.3.1	Požadované vlastnosti.....	34
5.3.2	Model vozidla 1:28.....	34
5.3.3	Další periferie.....	36
5.3.4	Ultrazvukové senzory	38
5.3.5	Arduino mikroprocesor	39
5.3.6	Tištěné díly pomocí 3D tiskárny.....	41

5.4	Programování.....	44
5.4.1	Hlavní smyčka.....	45
5.4.2	Popis a funkce jednotlivých částí.....	49
5.5	Mechanismus řízení rychlosti.....	55
5.5.1	Konfigurace PID knihovny.....	56
5.5.2	Detekce rychlosti.....	57
5.6	Snímání okolí pomocí 3 ultrazvukových čidel.....	58
5.6.1	Požadované vlastnosti a princip.....	59
5.6.2	Testování v praxi.....	59
5.7	Snímání okolí pomocí 5 ultrazvukových čidel.....	60
5.7.1	Požadované vlastnosti a princip.....	61
5.7.2	Testování v praxi.....	62
5.8	Vyhlazování sbíraných dat.....	63
5.8.1	Klasifikace a určení chyby.....	64
5.8.2	Zpracování a výpočet.....	64
5.9	Kalibrace jízdy a testování vozidla.....	64
5.9.1	Měření.....	65
5.9.2	Shrnutí jízdy.....	66
6	Shrnutí výsledků.....	67
7	Závěry a doporučení.....	68
8	Seznam použité literatury.....	69
9	Přílohy.....	71

Seznam obrázků

Obrázek 1 - Asistenční systémy vozidla a jejich technologie.....	6
Obrázek 2 - Pontiac 1935 Phantom	7
Obrázek 3 - GM Firebird III 1958 s ovládáním bez volantu	7
Obrázek 4 - Mercedes-Benz 410	8
Obrázek 5 - Lego Mindstorms NXT	13
Obrázek 6 - mBot Ultimate 2.0 - 10-in-1 Robot Kit.....	13
Obrázek 7 - VEX IQ Challenge 2019/20 robot.....	14
Obrázek 8 - OPENZ V3B RC podvozek	15
Obrázek 9 - Indukční senzor PNP OMRON.....	16
Obrázek 10 - Kapacitní senzor NPN SICK.....	17
Obrázek 11 - Optická kontrolní linka NETTO	18
Obrázek 12 - Ultrazvukové čidlo HC-SR04.....	19
Obrázek 13 - Radarové čidlo DFROBOT kombinující přijímač i vysílač.....	19
Obrázek 14 - Raspberry PI Camera Module V2.....	20
Obrázek 15 - Lidar vs. Radar porovnání	21
Obrázek 16 - Regulační smyčka.....	23
Obrázek 17 - Dvoupolohová regulace On/Off.....	23
Obrázek 18 - Jednotlivé složky PID regulace	24
Obrázek 19 - Arduino NANO popis rozhraní.....	27
Obrázek 20 – Originální podvozková platforma 1:28 Mini-Z.....	30
Obrázek 21 - Závodní dráha v Třebechovicích pod Orebem	31
Obrázek 22 – Podvozková platforma IWAVER boční pohled.....	35
Obrázek 23 – Karoserie VW Transporter T1.....	36
Obrázek 24 - Flysky GT2B s přijímačem	37
Obrázek 25 - OLED panel 128x32 i2c	37
Obrázek 26 - Obrázek– Srovnání Raspberry Zero a Arduino Nano	39
Obrázek 27 – Doplněk Arduino Camera.....	40
Obrázek 28 - Arduino Nano 2.0 zapojení.....	41
Obrázek 29 - 3D model předního nárazníku Slicer	42
Obrázek 30 – 3D model zadního nárazníku Slicer	43

Obrázek 31 – Opakující se smyčka loop()	47
Obrázek 32 - Metoda readDistFront()	49
Obrázek 33 - Výpočet vzdálenosti pomocí odrazu	49
Obrázek 34 - Metoda calculDirection()	50
Obrázek 35 - Úsek metody bočních čidel.....	51
Obrázek 36 – Průběh nájezdu do zatáčky	51
Obrázek 37 - Úsek metody čidel náklonových.....	52
Obrázek 38 - Úsek metody kombinující čidla	53
Obrázek 39 - Průjezd zatáčkou před najetím.....	53
Obrázek 40 - Průjezd zatáčkou po najetí	54
Obrázek 41 - Metoda calculateAndWriteSpeedLimited()	55
Obrázek 42 - Konfigurace PID knihovny.....	56
Obrázek 43 - Metoda readSpeed()	57
Obrázek 44 - Rotační mikropínač.....	58
Obrázek 45 - Model se 3 aktivními čidly v přední části vozu	59
Obrázek 46 - Model s 5 aktivními čidly v přední části vozu.....	61
Obrázek 47 – Testovaný finální model s 5 aktivními čidly	61
Obrázek 48 - Vyfiltrování extrémních hodnot.....	64
Obrázek 49 - Testovaná pravotočivá závodní zatáčka	65
Obrázek 50 - Postupy správného projetí zatáčkami	66

Seznam tabulek

Tabulka 1 - Stupně automatizace vozidla.....	5
Tabulka 2 - Testování průjezdu pravotočivou zatáčkou se 3 čidly	60
Tabulka 3 - Testování průjezdu pravotočivou zatáčkou s 5 čidly	63

1 Úvod

Historie autonomních dopravních prostředků se datuje již na začátek 20. století, kde hovoříme spíše o začátku velmi raných pokusů, ne o plnohodnotném startu. Teprve 21. století je ve znamení rapidního rozvoje a jedno z nich je právě v tomto odvětví, konkrétně autonomní řízení automobilu.

Pokusy o autonomní řízení byly zaznamenány již v roce 1925 kdy společnost Houdina Radio Control provedla úspěšné pokusy s dopravním prostředkem, který byl řízen několika servomotory. Automobil tak dokázal několik metrů jet autonomně tzn. bez zásahu a asistence řidiče.[1] Ve srovnání s prvními pokusy o autonomní pohyb, má současný pokrok v daném oboru, díky moderním technologiím, dramaticky výraznější progres. Leader současného trendu je společnost Tesla Motors, díky úrovni autonomie stupně 3, která je typická tím, že automobil je schopen autonomně převzít část nutné pozornosti pro provoz vozidla.

Autor této práce se zajímá o trendy v automobilovém průmyslu, je aktivní modelář a je velkým příznivcem programování zejména v oblasti mikroprocesorů a vývojových platforem. Hlavní důvod pro výběr tohoto tématu je zájem autora se v tomto směru dále vzdělávat a najít další hlubší využití této platformy.

2 Cíl práce

Cílem je vytvořit konceptuální model dopravního prostředku, který bude schopen jízdy jak za asistence řidiče, či jiného vstupního zdroje, tak jízdy zcela samostatné, tedy autonomní. Ta bude zahrnovat aktivní vyhýbání překážkám a plně souvislou jízdu. Stejně tak bude v reálném čase reagovat na neočekávané situace například překážku před vozidlem nebo náhlou změnu trajektorie dráhy. Způsob, jakým bude autonomní vstup reagovat na tyto situace bude předem jasně definován a bude kladen důraz na samostatné řešení vzniklých těchto situací. Koncepce modelu je směřována tak, aby v budoucnu mohl být systém rozšířen, například o umělou inteligenci a více interagujících modelů na závodní trati, které spolu budou spolupracovat.

3 Metodika zpracování

V rámci závěrečné práce bylo třeba prostudovat teoretické zdroje zabývající se oblastí autonomně řízených dopravních prostředků, používanými technologiemi a získat přehled o dostupném materiálu pro realizaci praktické části.

Před samotnou realizací praktické části bylo nutné stanovit jasné cíle a definovat limity pro stavbu. Prvním krokem bylo navržení struktury uzavřené testovací závodní dráhy, jež bude určena v první řadě modelům dopravních prostředků v měřítku 1:28, což odpovídá přibližně délce 15 cm. V závislosti na prvním kroku bylo třeba uvažovat o samotném modelu dopravního prostředku. V tomto případě se jednalo o speciální model automobilu osazeného specifickou elektronikou. Výstupem bylo samotné autonomní chování vozu.

Praktická část zahrnovala tvorbou vozidla z několika dílčích částí a implementací vlastního softwaru pro vývojovou platformu Arduino a následně nespočetné množství iterací na testovací dráze, které spočívaly v řešení kolizí a odstranění nedostatků. Součástí této části je popis všech aspektů vysledovaného autonomního chování. Pro dosažení cíle bylo třeba provádět analýzy, z kterých bylo potřeba vyvozovat závěry, které bylo možné implementovat do řídicího systému.

4 Teorie autonomního pohybu

Autonomní pohyb pro dopravní prostředky v 21. století znamená na určitý čas převzít plnou kontrolu k vykonávání činností, které by za normálních okolností fyzicky vykonával operátor. Mezi tyto činnosti patří interakce s vlivy prostředí, změna rozhodnutí, korekce chyb v rozhodování a schopnost předvídat. Tyto uvedené činnosti je nutné vykonávat s maximální možnou rychlostí a přesností, a to s ohledem na výkon systému samotného. Systémem rozumíme komplexní řešení, které sbírá z okolí vstupy a mění je na výstupy. Způsob, jakým algoritmem uvnitř tyto data zpracovává, je to nejdůležitější, čím může daný systém disponovat. Systém má zadaný jasný cíl a pomocí autonomních algoritmů reaguje v reálném čase na měnící se okolí a přizpůsobuje svůj chod a průběh. Samostatně se rozhoduje o svém chování a rozhodování, bez ohledu na vznikající chyby v měření nebo měnícím se okolí.

4.1 Úvod do autonomního pohybu

Autonomní pohyb se dá obecně rozdělit do skupin rozlišených podle stupně automatizace dopravního prostředku. Jsou seřazeny od nejnižší úrovně, která pouze varuje před nebezpečím, až po stupeň nejvyšší, který plně přebírá kontrolu nad vozidlem. Americká vládní agentura NHTSA zavedla 4 základní skupiny autonomizace dopravních prostředků v roce 2013. O 3 roky později rozšířila svou původní škálu o další stupeň. Od roku 2016 stupnice obsahuje 5 základních skupin, které budou dále popsány. Další možný vývoj povede k širšímu rozdělení do dílčích mezi skupin.

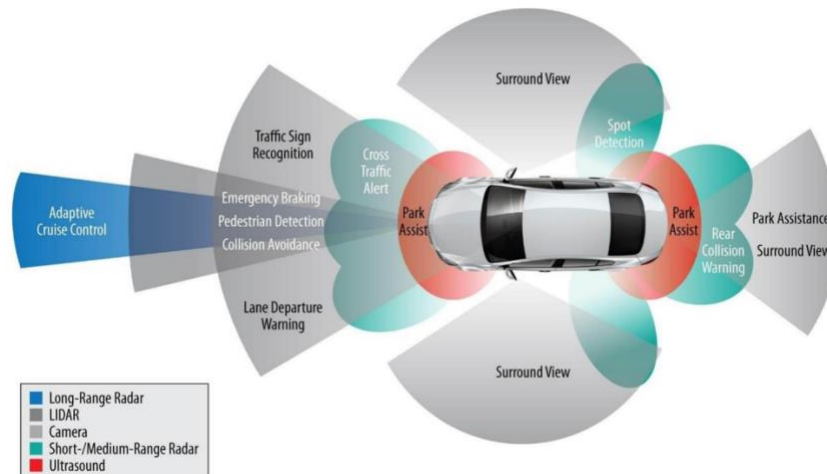
4.1.1 Teoretické základy

Stupeň automatizace	Charakteristika	Příklad
Nultý – bez automatizace	Nulová autonomie. Dopravní prostředek je ovládán zcela řidičem.	Ukazatel venkovní teploty.
První – nízká automatizace	Dopravní prostředek je řízen řidičem. Ve vozidle jsou nainstalovány asistenční systémy, které v řízení výrazně napomáhají.	Adaptivní tempomat, asistent řízení v pruhu.
Druhý – částečná asistence	Dopravní prostředek je řízen řidičem, ale za určitých podmínek je systém připraven převzít řízení.	Autonomní parkování.
Třetí – částečný dohled	Řidič nemusí mít ruce na volantu, ale musí být připraven řízení kdykoliv převzít.	Autopilot jízdy po dálnici.
Čtvrtý – vysoká automatizace	Schopno vysoké automatizace jízdy. Pokud řidič vyžaduje kontrolu nad vozidlem, může řízení převzít.	Autopilot na běžné silnici.
Pátý – plná automatizace	Vozidlo je schopno provádět všechny akce bez vědomí lidského faktoru. Řidič zadává cílovou destinaci.	Řízení vozidla za pomoci zvolené destinace.

Tabulka 1 - Stupně automatizace vozidla

(Zdroj: online <https://www.tpsd-ertrac.cz/file/oblast-autonomni-vozidla>)

Základní rozřazení autonomních řídicích systému je podle různých úrovní dle stupně automatizace, které přehledně shrnuje tabulka 1. Jsou seřazeny od nejprimitivnějšího typu autonomního řízení až po plně autonomní režim jízdy.



Obrázek 1 - Asistenční systémy vozidla a jejich technologie

(Zdroj: online <https://www.tpsd-ertrac.cz/file/3-etapa-oblast-autonomni-vozidla>)

Dostupné technologie snímání (obr. 1.) jsou důležitá pro správnou vizualizaci reálného prostředí. Automobily dnes pracují s Radarem dlouhé vzdálenosti, který snímá situaci před vozidlem od 100 m a dále. Obvykle je také kombinovaný s krátkovzdálenostní variantou. Dále je přítomný Lidar založený na principu světelného paprsku. Všechny systémy doplňuje kamerové snímání okolo vozidla. [2]

4.1.2 Dosavadní vývoj a trendy

Autonomně se pohybující dopravní prostředky byly ještě do nedávna vzdálenou vizí, která se teprve nyní stává skutečností. Hlavním zohledňovaným faktorem je pochopitelně bezpečnost s důrazem na schopnost řešit krizové situace.

V prvopočátku je nutné začít vývojem výpočetní techniky jako takové. Dnes se autonomnímu řízení věnují přední světové technologické firmy.



Obrázek 2 - Pontiac 1935 Phantom

(Zdroj: online <https://www.zive.cz/clanky/samoridici-auto-jezdilo-po-silnicich-uz-v-roce-1993-nemelo-radar-ani-gps/sc-3-a-195526/default.aspx>)

Historie sahá až do dvacátých let minulého století. Prvním automobilem byl Pontiac přezdívaný jako „phantom auto“ (obr. 2.). Automobil ovládali servomotory a aktuátory řízené na dálku.



Obrázek 3 - GM Firebird III 1958 s ovládáním bez volantu

(Zdroj: online <https://www.zive.cz/clanky/samoridici-auto-jezdilo-po-silnicich-uz-v-roce-1993-nemelo-radar-ani-gps/sc-3-a-195526/default.aspx>)

Pontiac Firebird III (obr. 3.) se lišil oproti sériovému provedení GM Firebird implementovaným dálkovým ovládáním, které pracovalo na krátkých vlnách s dosahem až 5 km. Tehdy se jednalo o jeden z nejúžasnějších a nejzajímavějších automobilů doby a také o výstřelek vědy a techniky. Dále disponoval například tempomatem, ABS nebo klimatizací.



Obrázek 4 - Mercedes-Benz 410

(Zdroj: online <https://www.zive.cz/clanky/samoridici-auto-jezdilo-po-silnicich-uz-v-roce-1993-nemelo-radar-ani-gps/sc-3-a-195526/default.aspx>)

Německá firma Bosch v roce 1993 demonstrovala svůj systém automatického řízení na světle zelené dodávce Mercedes-Benz 410. (obr. 4.) Mercedes byl vybaven elektronickým ovládním řízení, plynového pedálu, brzd a automatickou několikastupňovou převodovkou. Vozidlo bylo testováno na silnicích v uzavřené oblasti a během jízdy dosahovalo rychlosti až 90 km/h. Dokázalo se nejen udržet na silnici, ale také najít nejlepší cestu do zadaného cíle. Automobil sbíral data pomocí několika kamer umístěných před vozidlem a uvnitř vozidla. Informace byly následně zpracovávány palubním počítačem v zadní části vozu. Auto mělo k dispozici digitální mapy, ale nemělo k dispozici satelitní navigační systém GPS.

Kromě firem jako Bosch nebo Mercedes, pracovali na vývoji také další velcí hráči jako GM nebo GE nebo IBM. Přelom tisíciletí nastartoval rychlé pokroky v oblasti výkonu počítačů i samotných technologií pro autonomní řízení. Nasazení těchto systémů v osobních vozidlech umožnil také proces miniaturizace a sériové výroby pro běžné obyvatele.

Nejnovější algoritmy pro autonomní řízení automobilů se zabývají schopností učit se za provozu. Tyto systémy jsou velmi často založeny na strojovém učení. Bohužel se o autonomním řízení dá říci, že ještě zdaleka není bezchybné a 100% bezpečné. Postup doladování a vychytávání chyb je v tomto směru velmi žádoucí.

V roce 2017 podle statistiky Americké vládní agentury NHTSA zemřelo na silnicích v USA 37,5 tis. lidí. O rok později, v roce 2018 podle statistiky zemřelo 36 tis. lidí. Tento mírný pokles nehodovosti můžeme přikládat novějšímu vozovému parku a nadčasové výbavě nových modelů vozidel. Mezi faktory, které dále ovlivňují tuto skutečnost, můžeme řadit dokonalejší zádržný systém a modernější aktivní prvky ve vozidlech. Poté vyšší míra spolupráce automatizačních brzdících systémů, které řidiče dopravního prostředku informují anebo řidiči přímo zasahují do řízení.[3]

S postupným příchodem a pronikáním autonomních dopravních prostředků se liší také míra nedůvěry a opatrnosti, což se projevuje v legislativě jednotlivých států. Konkrétně v USA státy Michigan, Nevada a Washington DC omezují použití autonomních dopravních prostředků.

Navzdory všem kladům, existují i možná rizika. Jedním z nich je například možný hackerský útok. Jelikož by při nesprávném fungování autonomního systému mohlo dojít k vážnému selhání, měl by i nadále být možný lidský zásah do plně zautomatizovaného dopravního prostředku. Možnou hrozbou je snadný teroristický útok, který by mohl vést ke kolapsu celého systému. Proto je potřeba dbát na nejvyšší bezpečnostní opatření v místech, kdy jedno neuvážené rozhodnutí nebo bezpečnostní díra změní situaci a dojde tak k nevyhnutelné katastrofě.

Smyslem autonomních systémů je převzít co možná nejvíce lidské činnosti na sebe, nebo výrazně ulehčit práci. Má to však i stránky stinné. Bohužel, díky nim spousta z méně kvalifikovaných zaměstnanců může přicházet o práci a živobytí. [4]

4.1.3 Možnosti pohybu

Autonomní řízení dopravy má mnoho variant. Nejznámější a asi všemi nejvyužívanější je automobilová doprava. Avšak je zde řada dalších dopravních prostředků, které využívají autonomních pomůcek.

Jednou z nich může být železniční doprava, která postupně automatizuje. Za první plně automatizovanou linkou se považuje linka číslo 14 zvaná Météor v Paříži. Přepravuje osoby již od roku 1998 a ročně linka přepraví více než 64 mil cestujících.[5] Ryze Česká společnost AŽD například plánuje vydat na cestu plně

autonomní vlak, ve kterém bude strojvedoucí pouze kontrolovat aktuálně probíhající jízdu.

„Samořiditelné vlaky jsou mojí vysokou prioritou. První jednotka bude jezdit v roce 2020, říká generální ředitel AŽD Praha Zdeněk Chrdle“[6].

Dalším typem jsou například letadla, ve kterých existuje všeobecně známý autopilot. Nejnovější technologie umožňují bezpečný let a přistání po předem určené trajektorii za pomoci radiomajáků. Podobným způsobem je řízená i lodní doprava, i když k mému zklamání se automatickému řízení v lodích neříká autokapitán.

V dnešních městech již jsou plně funkční samořídící vozy metra. Praha není v tomto směru výjimkou. Do budoucna jsou v metropoli v plánu dokonce soupravy bez jakékoliv kontroly strojvedoucího. Tento návrh se projednává v souvislosti s chystaným metrem trasy D.[7]

4.2 Modely vozidel

Modely vozidel nebo také jiným názvem RC modely, což je zkratka z anglického slova Radio Controlled, jsou v podstatě zmenšeniny originálních velkých dopravních prostředků. Nemusí to být nutně zmenšenina automobilu, ale může se také jednat o modely motorek, letadel, ponorek apod. Modely můžeme dále dělit podle více hledisek. Prvním hlediskem je velikost podvozku. V praxi se setkáváme nejvíce s modely velikosti 1:10. Jedná se o modely s délkou přibližně 40 cm. Model určený pro bakalářskou práci má velikost 1:28 a je přibližně 15 cm dlouhý. Modely členíme také podle přenosové frekvence na novější s 2.4 GHz přenosovým pásmem, které bakalářská práce využívá a na starší modely s přenosovým signálem o frekvenci 40 MHz nebo dokonce 27 MHz. Poslední členění je podle určení použití modelu. Liší se hlavně pořizovací cenou a kvalitou zpracování. Pokud je model určen na profesionální závody v týmech, je pravděpodobnější, že pořizovací cena byla několikrát vyšší, než u konkurenčního modelu pro obvyklé domácí použití. Rozdíly nejsou na první pohled vidět, ale každá ušetřená vteřina na závodní okruhu je bonus pro řidiče modelu.

Na druhé straně jsou na trhu dostupná již naprogramovaná hotová řešení ve formě stavebnic. Tyto řešení ale nabízí velmi omezené možnosti rozšíření a jsou odkázány na svoje úzké využití bez možnosti přidávání a upravování konfigurace.

4.2.1 Obecné využití

Modely v obecné rovině chápeme víceméně jako přesné odrazy skutečného světa. Model nikdy nebude splňovat všechny vlastnosti reálného automobilu, avšak musí přesně vystihnout charakter a podstatu modelované věci.

Obecně jsou modely využívány v mnoha odvětvích. Například můžeme jmenovat armádní použití. Tyto modely jsou speciálně konstruovány na nejtvrďší podmínky, do kterých jsou speciálně navrhovány. Zachraňují ostatním lidem život, pronikají do míst, která jsou lidem zapovězena. Dalším odvětvím je doprava. Modely jsou zde používány pro analýzu situací, které jsou pro běžné simulace nepředstavitelné anebo by její simulace trvala několik let. Neméně početná je skupina amatérských modelářů, která používá modely nejrůznějších velikostí k hobby účelům a usnadňují si každodenní práci. Kromě toho také aktivně přispívají do online repositářů a společně se dělí o nápady na internetových stránkách. Naopak pro komerční firemní účely jsou k dostání na trhu speciálně určené robotické modely pro průmyslovou automatizaci. Pořizovací cena těchto robotických unikátů bývá často největším problémem. V těchto odvětvích se do budoucna očekává další vlna automatizace s vyvíjejícím se průmyslem. Robot může buď pomáhat, nebo dělat lidskou práci, popř. zajišťovat i komfort společnosti.

4.2.2 Požadované vlastnosti

Definování požadovaných vlastností modelu je klíčové pro úspěšný návrh bakalářské práce. Vybraný model musí být ze všech hledisek nejlepším řešením s ohledem na miniaturní velikost modelu. Důvod, proč bylo vybráno právě řešení na bázi RC modelů, je možnost rozšíření funkcionality a variabilita řešení. Dopravní prostředek, v tomto případě automobil ve velikosti 1:28, je speciálně upraven ke svému účelu, tedy myšleno platformu autonomně řízeného modelu automobilu na

závodní dráze. Toto řešení je náročnější na samotnou výrobu, která je nutná k provozu a naprogramování vývojové platformy uvnitř. Přepracována je hlavně řídicí elektronika uvnitř, včetně regulátorů a serv nutných k pohonu. Byly přidány senzory, které snímají okolí a dále byla přidána platforma Arduino, která rozhoduje o všech vstupech a výstupech. Pomocí speciálně upraveného Mini USB kabelu je realizována komunikace mezi počítačem. Servo je originálně shodné se závodním modelem, které se používá v ostrém provozu. Pohonná jednotka je zajištěna za použití motoru velikosti 130 pro stejnosměrné napětí 6 V a dvojitého H můstku zapojeného do vývojové desky Arduina.

4.2.3 Chování vozidla

Nejdůležitější fakt, který ovlivňuje jízdu samotnou, je vnímání okolí, rozhodování a reakce řidiče. Dále je třeba přizpůsobovat řidičské reakce aktuálnímu stavu vozovky. Typický příklad je snížení rychlosti do zatáčky. Dále je velmi složitým procesem například i jen udržení automobilu ve svém jízdním pruhu.

Při modelování reakcí automobilu začneme stanovením cílů, které má vozidlo vykonávat. K nahrazení vjemů řidiče poslouží ultrazvuková čidla, vývojová deska Arduino a Servo motor s pohonnou jednotkou. Tyto součástky jsou zodpovědné za reakce a jízdu vozidla. Řidičovy reakce jsou jako model zapsány v Arduinu, které pomocí vstupu z čidel posílají výstupy do servo motoru a H můstku, která řídí pohyb. Pokud čidla zaznamenají překážku, je žádoucí, aby systém okamžitě reagoval, případně zastavil. Dále je žádoucí, aby byl systém schopný udržet automobil, pokud možno na středové trajektorii mezi dvěma mantinely. Pokud bude muset vykonat model změnu směru, zatočí ve správném směru. Je velmi žádoucí, aby model, pokud to dovolí technologie, udržel ideální stopu a nenarazil do žádné překážky.

4.2.4 Varianty provedení

Řešení chytrého robotického vozidla nabízí firma DJI. Vozidlo DJI Robomaster S1 je edukativní, interaktivní a vhodné pro školy jako edukativní robot. Ovládá se

bud' přes Wi-Fi připojení mobilním telefonem nebo bezdrátově přes laptop pomocí technologií Bluetooth.

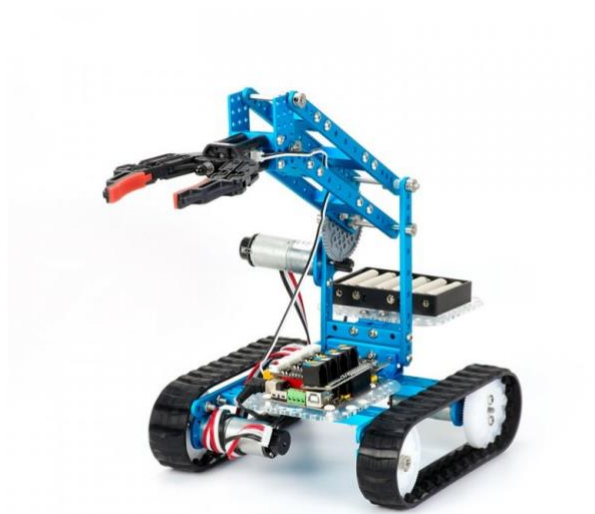


Obrázek 5 - Lego Mindstorms NXT

(Zdroj: online

https://simple.wikipedia.org/w/index.php?title=Lego_Mindstorms_NXT&oldid=6767140)

Hotové řešení Lego Mindstorms (obr. 5.) je programovatelná robotická stavebnice založená na prvcích ze stavebnice Lego. Dále jsou na trhu k dostání velmi podobná řešení.

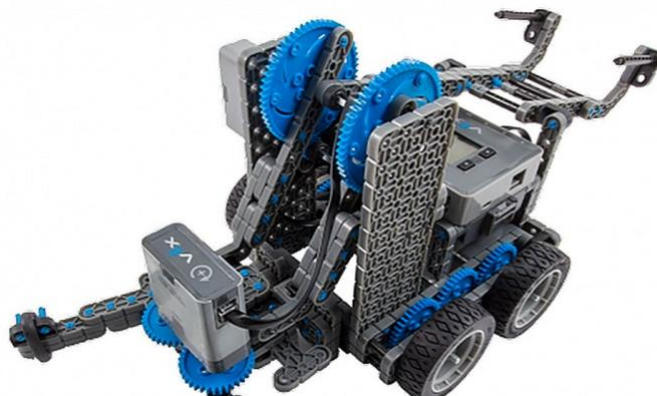


Obrázek 6 - mBot Ultimate 2.0 - 10-in-1 Robot Kit

(Zdroj: online <https://www.alza.cz/mbot-ultimate-2-0-10-in-1-robot-kit-d4588294.htm>)

Lego a jeho alternativy (obr. 6.) jsou velice intuitivní jednoduchá řešení. Svým vzhledem a použitelností jsou určena i pro menší děti. Stavebnice je vybavena USB

vstupem pro přímé programování přes PC. Hlavní výhodou je v její variabilitě a schopnosti postavit, jakkoliv velký model, který bude sloužit přesně danému účelu. Nevýhodou této stavebnice je pevnost dílků, ze kterých je postaven. Je určen spíše pro širší veřejnost s možností vysoké kreativity. Na stavbu závodního modelu v relativně malém balení není tato stavebnice vhodná.



Obrázek 7 - VEX IQ Challenge 2019/20 robot
(Zdroj: online <https://www.rapidonline.com/vexiq/challenge>)

Hotové řešení od společnosti VEX IQ Robotics (obr. 7.) je velmi sofistikovanou stavebnicí, která zaujme především provedením na velmi vysoké úrovni z hlediska elektronického vybavení a také díky pevnosti dílů. Disponuje kovovými zmenšenými díly dle své reálné předlohy. Nevýhodou tohoto modelu je nedokonalost v přední nápravě, která nezatáčí jako reálný automobil. Zatáčení by se dalo připodobnit k systému tanku. Otočka je realizována odlišnou rychlostí otáčení kol na pravé a levé straně. Také tento model postrádá odpružení náprav, kvůli kterým je na nerovném povrchu velmi nestabilní.



Obrázek 8 - OPENZ V3B RC podvozek

(Zdroj: online <https://cults3d.com/en/3d-model/game/openz-v3b-chassis-1-28-rc>)

Řešení na bázi RC modelů (obr. 8.) vychází z mnoha let zkušeností. Mnoho výrobců vyrábí ještě více variant modelů. Velmi známý výrobce je bezesporu Tamiya nebo LRP. Není lehké najít podvozek, který bude ideálním základem pro hladký provoz na rychlostní dráze a který bude splňovat požadavky bakalářské práce. V tomto ohledu není brán zřetel na výrobce, nýbrž záleží na vhodné velikosti, značce, pevnosti, určení, použitelnosti servo motoru a na velikosti pohonného ústrojí.

4.3 Technologie snímání okolí

Kvalita snímání okolí je základní veličinou jako vstupní data k vyhodnocení výstupů do pohonného motoru. Vývojový kit Arduino musí dostatečně rychle a ideálně ještě s předstihem mapovat cestu před automobilem. Proto je rozhodnutí výběru vhodného čidla klíčové pro úspěch a cíl autonomního automobilu. Další aspekt je vhodná vstupní a výstupní vybavenost vývojové desky. Výhodou je i2c a 10bit AD převodník.

4.3.1 Senzory indukční

Indukční senzory jsou založeny na mechanismu obsahující cívku a pohybující se části, které mění svůj odpor v závislosti na přibližovaném kovovém předmětu.



Obrázek 9 - Indukční senzor PNP OMRON

(Zdroj: online <https://www.tme.eu/cz/details/e2bm12ks04m1b1/indukcni-cidla-valcova-dc/omron/e2b-m12ks04-m1-b1>)

Snímatelná vzdálenost indukčního čidla (obr. 9.) je obvykle až 10 cm. Se vzrůstající vzdáleností narůstá nepřesnost v měření. Čidla jsou relativně velmi přesná a miniaturní. Cena za kus čidla se pohybuje v závislosti na kvalitě výrobce od 50 Kč do 300 Kč. Také snímací úhel je velmi vysoký, obvykle kolem 90 stupňů. Jelikož čidlo je z principu velmi přesné, je také odezva čidla velmi malá. Čidlo je nevhodné na snímání jakýchkoliv jiných materiálů, než jsou feromagnetika. Z tohoto důvodu nelze čidlo použít na snímání okolí na závodní dráze, jež tvoří měkká guma a výplně.

4.3.2 Senzory kapacitní

Senzory kapacitní (obr. 10.) snímají okolí pomocí dvou rovnoběžných desek. Princip fungování je takový, že pokud dojde k pohybu před senzorem, je okamžitě naměřena změna v detekované kapacitě. Z tohoto důvodu je vhodné tyto čidla použít tam, kde je nutno měřit velmi drobné odchylky změny polohy nebo zrychlení.



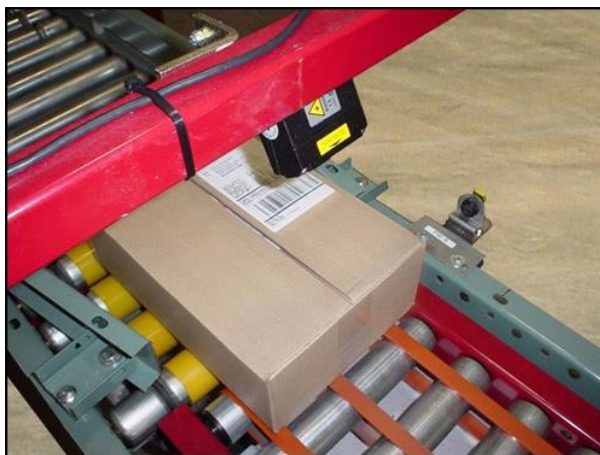
Obrázek 10 - Kapacitní senzor NPN SICK

(Zdroj: online <https://www.tme.eu/cz/details/cm18-12nnp-ec1/kapacitni-snimace/sick>)

Snímatelná vzdálenost tohoto senzoru je až 2 m. S rostoucí vzdáleností velmi prudce klesá přesnost měření. Rozlišení čidla je přibližně v řádech centimetrů. Tento pokles se vzrůstající vzdáleností zhruba platí na intervalu do 25 cm. Poté je rozlišení a přesnost velmi strmá funkce, což znamená, že již nelze rozpoznat překážku. Cena čidla je variabilní a pohybuje se od 100 Kč do 500 Kč. Nejedná se tedy o drahou záležitost, v porovnání s optickými čidly. Odezva čidla je velmi nízká, pohybuje se v řádech milisekund. Úhel interakce čidla od osy čidla na překážku neboli zorný úhel zachycuje až do 90 stupňů. To dovoluje použití zařízení tohoto typu v širokém průmyslu. Čidlo zjistí změnu, pokud se změní vzdálenost mezi deskami anebo vlastnosti materiálu pomocí plochy překryvu desky.

4.3.3 Senzory optické

Optické senzory jsou v posledních letech velmi žádaným zbožím v průmyslu. Hlavní využití v tomto směru mají převážně pro kontroly kvality výstupu. Skládají se z částí optického senzoru, nejčastěji infračervené kamery, zdroje světla a předmětů, které tento snímač kontroluje. Jsou nevhodné pro implementaci v autonomním automobilu malé velikosti.



Obrázek 11 - Optická kontrolní linka NETTO
(Zdroj: online <http://www.vahynetto.cz/1049-kamerove-snimace.html>)

Mezi hlavní důvody patří vysoká energetická náročnost, nutnost použití výkonného procesoru a velikost celé soustavy, což je patrné na ilustraci. (obr. 11.) Dále je nutnost klást vysoké nároky na čistotu prostředí ve kterém se optické senzory využívají, jelikož i nepatrně malá prachová částice zachycená na optickém senzoru může být klasifikována jako falešná závada. Snímaná vzdálenost úzce souvisí s optickou soustavou, která je s optickým senzorem spojena.

4.3.4 Senzory ultrazvukové

Zkratka SONAR anglických slov Sound Navigation and Ranging, neboli ultrasonické senzory, které používají vržené zvukové vlny k detekci překážky. Čidlo se skládá ze dvou hlavních částí. Z vysílače, který vysílá v pásmu nad 18 kHz ultrazvukový tón, který je poté znovu snímán a podle času ve kterém dorazí zpět do čidla se odhaduje jeho vzdálenost od překážky. Čím je čas od vyslání pulzu a přijetím delší, tím větší je vzdálenost, která je mezi překážkou a čidlem. Tento typ senzorů je spolu s radarovým snímáním hojně používán v automobilovém průmyslu. Využívá se k detekci různých překážek a k rychlému rozpoznání spánku řidiče nebo je využíván v parkovacích asistentech. Hodí se na vzdálenosti do 10 m. Přesnost čidel bývá okolo centimetru a snímací úhel je oproti ostatním čidlům relativně malý. Přibližně se jedná o úhel okolo 25 stupňů. Snímače mají horší odezvu, která se

pohybuje okolo 50 ms. Cena se pohybuje kolem 250 Kč za kus. Avšak cena závisí na kvalitě, na výrobci a na místě využití.

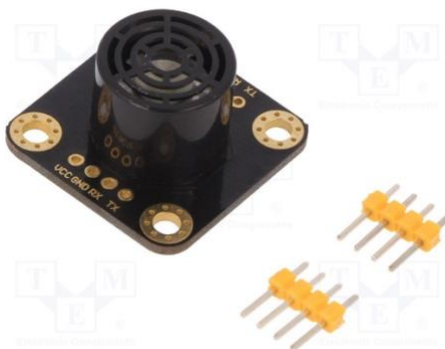


Obrázek 12 - Ultrazvukové čidlo HC-SR04

(Zdroj: online <https://www.tme.eu/cz/details/oky3261/cidla-vzdalenosti/okystar>)

Nejenom na ilustraci výše (obr. 12.), ale i skutečné čidlo je velmi malé. Má zanedbatelnou energetickou náročnost a obsluha vývojovým kitem Arduino je velmi jednoduchá. Úhel snímání čidla není široký a tím pádem je více než vhodný na snímání překážek okolo miniaturního automobilu. Horší odezva okolo 60 ms však může při rozměrech a pohybu uvažovaného modelu působit komplikace.

4.3.5 Senzory radarové



Obrázek 13 - Radarové čidlo DFROBOT kombinující přijímač i vysílač

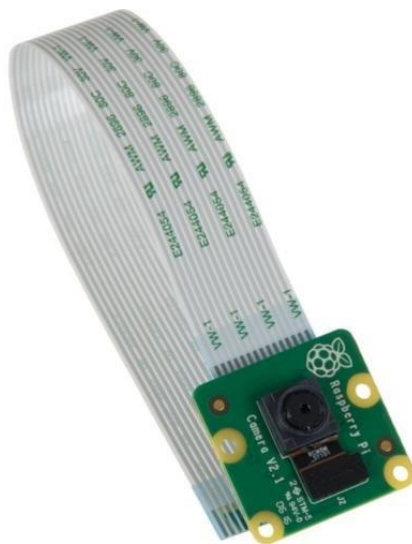
(Zdroj: online <https://www.tme.eu/cz/details/df-sen0153/cidla-vzdalenosti/dfrobot/sen0153>)

Zkratka RADAR anglického slova Radio Detection and Ranging značí systém využívající elektromagnetické vlnění k odhalení překážek před čidlem. Stejně jako ultrazvuková čidla i radarové čidlo používá vysílač a přijímač. Také existují varianty, které sice nejsou úplně běžné, ale přesto se s nimi setkáváme. (obr. 13.) Příklad čidla kombinující vysílač a přijímač do jednoho pouzdra od firmy DFROBOT.

Čidlo jako takové z principu měří čas, za který čidlo dostane odpověď zpět. Dále jsou radarová čidla ideální na použití v automobilovém průmyslu jako detektor hlídání vzdálenosti před vozidlem. Používá se v koncernu Volkswagen jako adaptivní tempomat, který hlídá vzdálenost před vozidlem nebo bezpečnostní brždění při zaznamenání nebezpečné překážky před vozidlem. Nevýhoda je vysoká pořizovací cena a energetická náročnost výpočtu dat z čidla. Snímatelná vzdálenost je až 100 m hlavně díky velmi úzkému úhlu snímání, který čítá 5 stupňů.

4.3.6 Kamerové snímání

Nedílnou součástí je také technologie kamerového snímání okolí a následné zpracování obrazu ve výpočetní stanici.



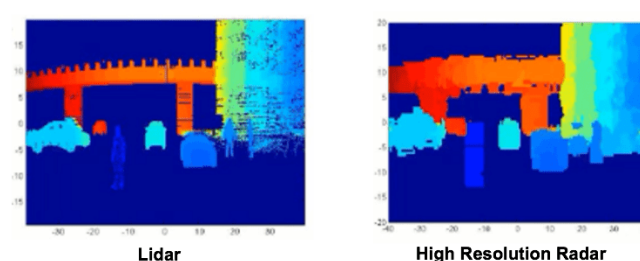
Obrázek 14 - Raspberry PI Camera Module V2

(Zdroj: online https://www.tsbohemia.cz/raspberry-pi-camera-module-v2_d347172.html)

Kamerové snímání se dostává do běžných automobilů, a tak se kamery řidiči snaží usnadnit situaci, rozhodování a reflexy. Stabilita a výkonnost systému velmi

závisí na výkonnosti a algoritmu strojového učení. Systém lze velmi dobře rozšířit pro použití na modelech automobilů až po vojenské letouny, kterým zajišťuje „vidění“. Nevýhodou systému je relativně snadná manipulace s daty, kdy může dojít k chybě jen pouhým zakrytím čočky nebo změnou podmínek. Další sledovaný aspekt je vysoká energetická náročnost při kvalitním a použitelném úsudku. Pokud chceme vyrobit robota, který nebude chybovat za jakýchkoliv vnějších podmínek, je vhodné použít všechny výše zmíněné technologie k velmi kvalitnímu snímání v reálném čase. Nejspíše použijeme velikostně adekvátní kameru (obr. 14.) s vysokým rozlišením.

4.3.7 Lidar vs. Radar



Obrázek 15 - Lidar vs. Radar porovnání
(Zdroj: online <https://semiengineering.com/radar-versus-lidar>)

Technologie jsou si velice podobné, avšak záleží na konkrétní aplikaci a určení. Obě technologie využívají odraz od překážek. Můžeme sledovat několik měřitelných veličin. Velmi zajímavé je srovnání napříč přesností, použitím paprsku, a chováním v horších venkovních podmínkách. Ilustrace (obr. 15.) porovnává výslednou představu, zejména o kvalitě obou uvedených technologií.

LIDAR je přesnější systém na monitorování menších překážek, který si bez problémů poradí i s pohyblivou překážkou a nedokonalostmi v reálném provozu. Nevadí mu tma a je velmi všestranný. Dále náročnější na mechanické vlivy a vlivy okolí, jako je například počasí. Pokud bychom řídili autonomní automobil opatřený pouze systémem Lidar za deštivého počasí anebo se sněhem, je velice pravděpodobné, že tento systém bude vyřazen velmi rychle z provozu. Proto při

návrhu nového automobilu musí tyto aspekty být brány v potaz a jsou často kombinovány ještě s dalšími optickými systémy. Lidar neboli měření vzdálenosti pomocí odrazu zpět k čidlu. Jedná se o technologii, která detekuje objekty před čidlem a měří jejich velikost a přesnou dispozici. Technologie Lidaru se objevila na trhu po Radaru a Sonaru a na rozdíl od rozhlasových nebo zvukových vln využívá ke skenování prostředí laserové světelné pulzy.

RADAR neboli Radio Detection and Ranging se používá k detekci objektů na velkou vzdálenost. Radar snímá rychlost objektu a jeho dispozici od čidla. Prodleva zvukové ozvěny nám umožňuje měřit, jak daleko se objekt nalézá. Doba, po kterou se zvuk šíří prostorem, je určena vzdáleností zdroje zvuku a povrchu, který odráží zvukové vlny. Dopplerův jev nám zároveň umožňuje definovat rychlost pohybujícího se objektu. Technologie systému byla vynalezena již v roce 1940.[8]

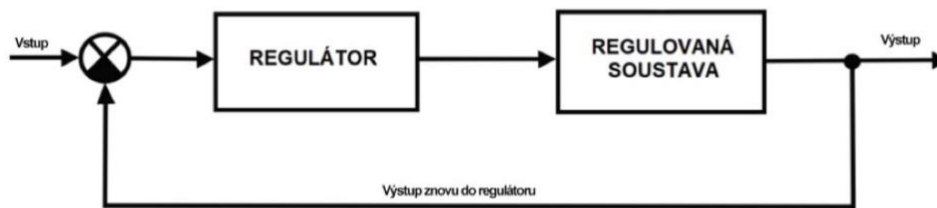
4.3.8 Shrnutí variant a rozhodnutí

Vzhledem k velikosti a spotřebě energie bylo zvoleno snímání vzdálenosti pomocí ultrazvukového čidla. Vzájemnou komunikaci zajišťují již hotové napsané knihovny. Na modelu bude kvůli nutnosti snímání situace okolo modelu osazeno více shodných čidel. Dále je vybrané ultrazvukové čidlo vhodnější na snímání kratších vzdáleností, kdy je přesnější než čidlo radarové, které se hodí spíše na vzdálenosti delší než 1 metr.

4.4 Základy regulace

Veličinu regulujeme, pokud chceme sledovanou hodnotu nebo veličinu udržet ve stanovených mezích. V průběhu regulace se zjišťuje míra hodnoty veličin a srovnávají se s hodnotami, které by podle předpokladu měly nabývat. Čím menší jsou odchylky, tím přesnější je regulátor.

Regulace používá svůj vlastní okruh na principu čtení vlastních výstupních hodnot. (obr. 16.) Jedná se o tzv. regulační smyčku. Dále předpokládáme nastavení pro konkrétní hodnoty.

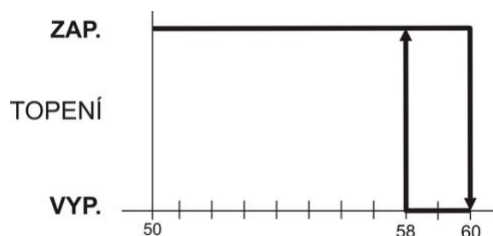


Obrázek 16 - Regulační smyčka
(Zdroj: vlastní tvorba)

Nastavená hodnota je vstupní parametr do regulátoru. Regulátor nastaví výchozí hodnotu a vydá pokyn regulované soustavě. Regulovaná soustava vykoná pokyn a jeho výstupem je hodnota, která se dostává zpět do regulátoru. Výsledné hodnoty se v průběhu času zpřesňují, pomocí postupné korekce ze zpětné vazby.

4.4.1 ON / OFF regulace

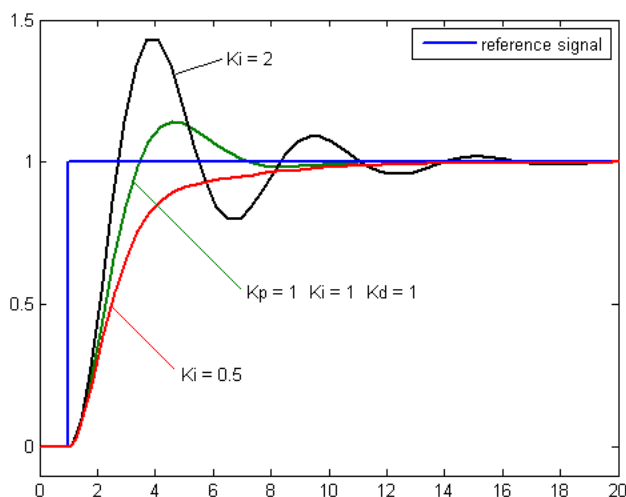
Tento druh regulace je ten nejjednodušší pro implementaci. Jednoduše po dosažení nastavené hodnoty provádí sepnutí či vypnutí výstupu. Tento fakt nazýváme dvoupolohovou regulací. Můžeme také hovořit o určitém pásmu, kdy regulátor čeká na dosažení určitých mezí. Jinými slovy se tento interval nazývá hystereze. Teoretické základy regulátoru jsou velmi triviální pro realizaci ještě jednodušší, jelikož je velmi široce využívána a existuje mnoho dokumentovaných zdrojů.



Obrázek 17 - Dvoupolohová regulace On/Off
(Zdroj: vlastní tvorba)

4.4.2 PID regulace

Jedná se o nejpoužívanější a s vhodným nastavením nejpřesnější regulaci veličiny. Zahrnuje proporcionální, integrační a derivační člen. Všechny 3 členy jsou na výstupu sečteny. Regulace pomocí PID může být také čistě spojitá funkce nebo dvoupolohová. Častěji se používá regulace spojitá, kterou budeme využívat na plynulé dosažení nastavené rychlosti modelu.



Obrázek 18 - Jednotlivé složky PID regulace

(Zdroj: online https://en.wikipedia.org/w/index.php?title=PID_controller&oldid=945158329)

Proporcionální složka regulátoru určuje rozdíl mezi aktuální naměřenou hodnotou a hodnotou, kterou se snaží regulátor dosáhnout. Je to nejjednodušší člen, který podle naměřených hodnot, které chybí do hodnoty předem dané, lineárně upravuje svůj výkon výstupu. Proporcionální složku jako takovou můžeme použít zcela samostatně, na rozdíl od dalších. Ilustrace (obr. 18.) znázorňuje černou barvou chování regulátoru s převahou proporcionální složky. Poté ale musíme brát v úvahu zvýšenou mírou překmitů soustavy. Další složky nám pomáhají regulátor lépe udržet a rychleji stabilizovat.

Integrační složka je o trochu složitější část systému PID regulace než proporcionální složka. Její hlavní parametr je vyjádřen v časové konstantě. Vyjadřuje změnu na vstupu a podle toho upravuje stejnou měrou výstup. Pokud bude odchylka na vstupu regulátoru od nastavené hodnoty vysoká, bude také snaha integrační složky regulátoru upravit výstup co nejrychleji. Integrační člen je často

neprávem opomíjen, přitom je velmi důležité tento parametr nastavit jako druhý. Na ilustraci (obr. 18.) je znázorněn zelenou barvou. Pokud se soustava dostává do překmitů a nelze ji stabilizovat, téměř vždy za tuto anomálii může právě tento člen.

Derivační člen je vyjádřen též v časové konstantě. Reaguje na rychlost změny, tedy s jakou rychlostí se daný systém přibližuje k nastavené hodnotě. Pokud derivační člen zjistí náhlou změnu, která není v minulém trendu stoupání vstupní hodnoty, náležitě upraví výstup, aby nedošlo k velmi vysokému překmitu. Systém, kde budeme předpokládat vysoký podíl derivačního členu, bude relativně nestabilní a nelineární. V praxi můžeme hovořit například o zahřívání velmi malého tělesa s obrovským výkonem a snahou těleso zahřát na přesně nastavenou teplotu. Derivační člen zjistí velmi rychlé přehřátí nastavené teploty, a bude se snažit kompenzovat výstup záporným vlivem. Reálné použití derivačního členu je téměř vždy posledním nastavovaným parametrem, protože ne vždy plnou mírou záleží na rychlosti změny překmitu. Na (obr. 18.) je derivační složka znázorněna červenou barvou.[9]

4.4.3 Shrnutí variant a rozhodnutí

Použití jednoduché ON/OFF regulace je velmi jednoduché. Touto regulací je ošetřeno kritické napětí baterie, což chrání akumulátor před zničením a pod vybitím. Pokud je kritická hranice dosažena, Arduino odpojí elektroniku z provozu. Na plynulý chod modelu a dosažení rychlosti modelu je použita spojitá PID regulace s pevně nastavenou rychlostí modelu.

4.5 Popis řídicí elektroniky

Řídicí elektronika modelu je rozdělena do několika bloků. Hlavní systémovou část, která obsluhuje výstupy do servo motoru a pohonné jednotky, tvoří platforma Arduino Nano V2 s 16 MHz AVR mikroprocesorem. Napájení modelu je tvořeno baterií velikosti 18650 a motor o velikosti 130. Servo motor má hmotnost 3 g. Obvyklá elektronika přijímače je přítomna ve formě 2,4 GHz modulu pro bezdrátový příjem signálů.

4.5.1 Napájení

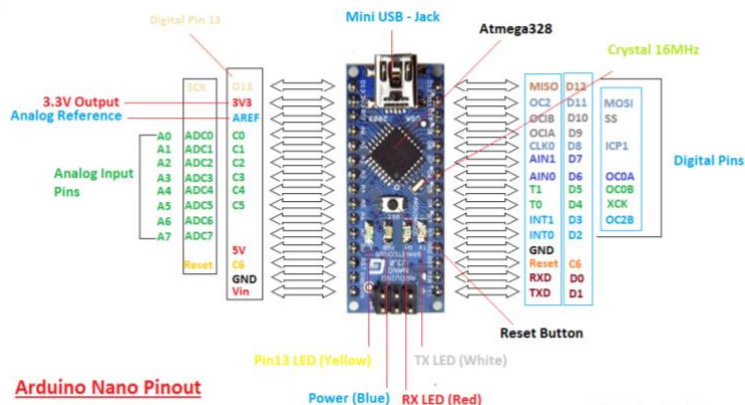
Baterie o velikosti 18650 s nominálním napětím 3.7 V značky Panasonic a jmenovitou kapacitou 3200mAh s technologií Li-Ion HV. Pro plné nabití baterie je vyžadováno napětí 4,25 V. Tento typ baterií je díky vysoké kapacitě hojně využíván v přenosných laptotech. Baterie má ideální vlastnosti pro uplatnění v modelu a vzhledem k miniaturní velikost má dostačující kapacitu. Maximální možný odebíraný proud činí 6 A, což dovoluje zásobovat motor nepřerušovaným a stabilním napětím s nízkým vnitřním odporem baterie. Jedinou nevýhodou je relativně krátká výdrž na jedno nabití. Nijak však neomezuje možnosti provozu a funkčnost modelu.

4.5.2 Ovládání modelu

Ovládací část modelu je navržena přímo na provoz závodního modelu. Je použit 2.4Ghz přijímač a vysílač, který komunikuje za pomoci 3 kanálů, což zajišťuje dostatečnou rychlost a přesnost přenosu informací. Přijímač umístěný v modelu automobilu je poměrně lehký, miniaturní a energeticky nenáročný pro dané použití. Komunikace mezi přijímačem v automobilu a řídicí jednotkou Arduino je realizována pomocí PWM modulace a její velikostí cyklu. Komunikace má 8 bit šíři, je tedy definována celkem 255 stavy pro každý kanál. [10]

4.5.3 Mikro kontrolér

Vývojový kit Arduino Nano ve verzi 3.0 (obr. 19.) je přepracovaný návrh již zažitého kontroléru. Vychází z vývojové desky Arduina UNO, která je osazena procesorem ATmega 328P.



Obrázek 19 - Arduino NANO popis rozhraní

(Zdroj: online <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html>)

Mikro kontrolér architektury AVR pracuje s kmitočtem 16 MHz a vnitřní paměť 32 kB slouží pro ukládání projektů. Ve srovnání s vývojovou deskou Arduino UNO nesoucí stejný procesor dostalo Arduino Nano menší pouzdro a skladnější balení. Vývojová deska má integrovaný USB vstup pro rychlé programování přes PC utilitu Arduino IDE. Arduino vyžaduje napájení nejméně 5 V stabilizovaných a odrušených od překmitů. Dále je možno platformu napájet 7–15 V, kdy si deska upraví své napájení sama. Na celkem 14 digitálních a analogových výstupů je možno připojit periferie. Analogových vstupů je celkem 8 s rozlišením 10 bitů. V modelu jsou použity pro čidlo otáčení kol a monitorování jejich rychlosti. Dále existuje možnost komunikovat sériově nebo přes protokoly jako například SPI, I2C. Oproti laboratorní desce typu Arduino UNO mají výstupy omezeny proud na 40 mA a také jsou méně energeticky náročné.

4.5.4 Periferie

Jako snímač vzdálenosti od bariér byl vybrán ultrasonický senzor HC-SR04. Senzor se používá nejen v aplikacích v robotice, ale využití má i jako jednoduché čidlo pro detekci překážek. Aplikací je spousta ať už od zavírání dveří, přes detekci schodů, až po foto pasti a mnoho dalších. Velmi nízká cena okolo 60 Kč dělá z čidla snadno dostupnou periferii za přijatelnou cenu. Energetická náročnost čidla je 15 mA. Nosná frekvence 40 kHz a maximální snímaná vzdálenost je 6 m.

Senzor, který zpracovává rychlost modelu je rotační mikropsínač se třemi kontakty, které jsou propojeny s vývojovou deskou. Mikropsínač má velmi podobnou funkčnost jako optická závora. Využívá principu rotačního kola se štěrbinami. Při rotaci se mění hodnoty na sepnutý a rozepnutý stav na výstupních pinech. Výstupy čidla jsou snímány pomocí analogových vstupů, které jsou naprogramovány na překročení určité meze, kdy Arduino zaznamená přechod stavu napětí na pinu. Frekvence, s jakou snímá přechody jednotlivých stavů otočného čidla, uvažujeme jako aktuální rychlost modelu vozidla.

4.6 Popis mechanické části

K mechanické části popisovaného modelu neméně patří jeho podvozková platforma. Tu tvoří základ od čínské firmy IWAVER. Konkrétně model O2M velikosti 1/28. Je dostatečně pevný pro uložení veškeré elektroniky. Nejkritičtější pro implementaci do podvozkové platformy je baterie a další velikostně náročnější periferie. Odpružena je přední i zadní náprava. Podvozek disponuje, na rozdíl od levnějších variant klasickým servo motorem a továrním elektromotorem pro 5 V s délkou 20 mm.

Robustní konstrukce tohoto modelu je vhodná i pro dosažení větších rychlostí. Celý hotový model váží při plném složení necelých 700 g. Při porovnání s originálním závodním speciálem hovoříme o násobcích váhy, kterou je podvozek nucen snést. Z tohoto důvodu je velká pravděpodobnost, že dříve či později selže některá z pohyblivých částí podvozku.

4.6.1 Mechanismus přední nápravy

Servo motor váhy 3 g je dostačující k pohybu přední nápravy. Přední náprava je odpružená, tak aby bylo možno model plynule ovládat i na nerovném povrchu. Výrobce SG není úplně podstatný. Na trhu je daleko více možných variant. Liší se svojí rychlostí a proudovým odběrem. Servo je analogovým typem, a tak používá pro kontrolu zpětného úhlu natočení analogový potenciometr.

4.6.2 Rychlost automobilu

Pohonné ústrojí modelu tvoří stejnosměrný elektromotor. Nezatížený motor dosahuje při volnoběžných otáčkách okolo 18 000 RPM a je převodován pastorkem se 6 zuby pro pomalejší pohyb. Pomalejší převodový poměr kvůli limitacím ze strany podvozku použít nelze. Pravá a levá strana zadní hnané nápravy je oddělena diferenciálem pro snadnější zatačení. Diferenciál je ze vnějšku tvořen velkým ozubeným kolem, do kterého zapadá pastorek z elektromotoru. Jsou použity velmi měkké pneumatiky pro maximální kontakt s vozovkou.

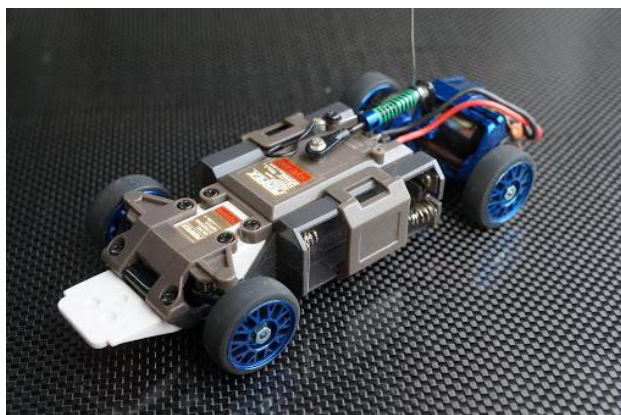
5 Předpoklady a realizace modelu

5.1 Výchozí podmínky

Prvotní model byl osazen 3 senzory v přední části vozu. Volba senzoru padla na ultrasonický senzor HC-SR04 pro snímání vzdálenosti z více úhlů. Byl vybrán hlavně díky velmi nízké ceně a jednoduché použitelnosti knihoven pro Arduino. Další výhodou je, že je velmi přesný a energeticky nenáročný.

Oproti závodním továrním modelům, byl autonomní model konstruován s nejvyšší prioritou samočinné jízdy. Rychlost a přesnost modelu v manuálním režimu není prioritní. Stejně tak ani velikost a váha není klíčová. Dále není třeba se zaobírat projetím celého závodního okruhu nejvyšší rychlostí. Závodní dráha Kyosho Mini-Z tvoří základnu, na které jsou testovány závodní modely série Mini-Z velikosti 1:28 a další velikostně podobné závodní speciály.

5.1.1 Velikost modelu



Obrázek 20 – Originální podvozková platforma 1:28 Mini-Z

(Zdroj: online <http://blog.rcmart.com/rc-128-mini-chassis-showdown-mini-z-vs-gl-racing-gla-vs-atomic-bz>)

Rozměry čistě podvozkové platformy (obr. 20.) jsou následující: Délka vozidla je 13 cm, šířka 5 cm a výška od vozovky 6,5 cm. Na výše uvedené délce není zohledněn modul na snímání vzdálenosti, který bude umístěn v přední části vozidla. Karoserie je tvořena kovovým základem s nalakovaným povrchem, který je zpestřován náhradními pneumatikami a předním nárazníkem. Vozidlo je poháněno zadní nápravou. Přední náprava je nehnaná a slouží ke změně směru vozidla. Pneumatiky jsou určené do vnitřních prostor. Šířka pneumatiky je 8,5 mm bez vzorku.

5.1.2 Typ a vlastnosti dráhy

Dráha samotná má rozměry 535 cm na délku a 435 cm na šířku. Avšak tvar dráhy netvoří dokonalý obdélník. (obr. 21.)



Obrázek 21 - Závodní dráha v Třebechovicích pod Orebem
(Zdroj: vlastní tvorba)

Závodní dráha obsahuje díly ze stavebnice Mini-Z Grandprix Circuit (obr. 21.) varianty s 12 zatáčkami a jedním dlouhým úsekem. Levotočivých zatáček je celkem 8. Zbylé 4 zatáčky jsou pravotočivé. Dráha obsahuje most, kde je umístěn čítač počtu kol a světelné signály na odstartování jízdy automobilu. Na okrajích dráhy se nachází vyvýšené mantinely, které navádí automobily zpět do dráhy. Jsou umístěné po celém obvodu dráhy. Výška mantinelu je 3 cm a je na celé šířce dráhy konstantní.

5.1.3 Způsob hodnocení

Mantinely ohraničující dráhu byly využity k navádění automobilu po jeho ideální stopě. Nelze předpokládat, vzhledem k neustále se měnícím podmínkám prostředí, naprostou bezchybnost úsudku měření. V modelu budou implementovány algoritmy na vyhlazení chyb, které toto riziko minimalizují.

Jízda je hodnocena úspěšnými projezdy zatáčkami v závislosti na četnosti projetí. Pokud automobil bude postaven před zatáčku, je pro něj klíčové správné rozhodnutí o jejím průjezdu. Je zde vysoké riziko výskytu chyb v porovnáním s přímým směrem jízdy. V průběhu času a realizace programování a stavby modelu hraje velkou roli počet úspěšně projetých zatáček. Na tomto počtu je úspěšnost programování vyhodnocována a je volen další postup.

5.2 Analýza dráhy

Délka dráhy je při projetí modelu jejím středem 10 m. Závodníci musí danou trasu projet jedno kolo za čas mírně překračující 5 vteřin. V celé své ploše je tvořena shodným povrchem, bez změny. Je sestavena z bloků, které mají na okrajích zámky, které se do sebe zasouvají na principu puzzle. Jediné nerovnosti, které musí automobil zvládnout jsou právě tyto spoje. Bloky mají černou barvu, avšak vzhledem k tomu, že měření modelu není založeno na principu rozpoznávání barev, tedy není vedeno na podél žádné barevné stopy, ale je postaveno na principu měření vzdálenosti od mantinelů, tak i barva povrchu v tomto případě nehraje roli. Segmenty mají tvar čtverce s délkou strany 50 cm. Dráha je široká 80 cm. S touto hodnotou budeme dále v kapitole naprogramování a vyhlazení chyb z čidel počítat.

5.2.1 Rychlost

Rychlost závodních speciálů 1:28 se pro tento typ dráhy dosahuje až 40 km/h v rovných úsecích. Použité motory typu 130 vyžadují napájení okolo 6 V a špičkový proud až 10 A. Použitý slabší motor pro nižší maximální napájecí napětí 3 V plně dostačuje k plynulému pohybu modelu vpřed nebo vzad. Rychlost modelu stoupá se zvyšujícím se požadavkem na rychlejší jízdu. Ovládán je pomocí PWM modulace. Motor přenáší točivý motorek skrz 6 zubový pastorek na 42 zubů pevně spojených s tělem diferenciálu. Rychlost vozidla je snímána a jde jako vstup do řídicí jednotky Arduino. Tento vstup je v modelu omezen na 10 cm/s, kvůli přílišné chybovosti způsobenou vysokou rychlostí modelu a nestabilitou vyvolanou konstrukčním řešením podvozkové platformy.

5.2.2 Plynulost a dynamika jízdy

Při realizaci bylo použito ovládání pomocí PID regulace a snímání rychlosti kol. Při nastavení PID velmi záleží na hmotnosti modelu a na velikosti tření pohyblivé soustavy.

Vzhledem k tomu, že elektromotor nedokáže roztáčet kola automobilu plynule lineární rychlostí v závislosti na přísunu proudu, je nutné regulovat

překmity a korekce vzniklé přílišným nebo naopak nedostatečným výkonem elektromotoru. Dále je síla elektromotoru silně závislá na aktuálních otáčkách. Nejvíce je namáhaný elektromotor a celá regulační soustava při snaze uvést model do pohybu. Předchozí problémy by řešila suchá spojka, která by oddělovala točivý moment, který je předáván z elektromotoru na vozovku. [11]

5.2.3 Změna pohybu

Řídící jednotka komunikuje se servo motorem pomocí jednoduchého PWM signálu. Jednotka má obvykle 3 vývody. Červený, který slouží pro napájení. Další je černý, kterým se zařízení uzemňuje. Poslední je žlutý, který komunikuje s ovládací elektronikou Arduino. Rotační pohyb motoru se na výstupu převádí na pohyb, který mění úhel natočení kol. Pro přenos pohybu ze serva na proporcionální natočení kol přední nápravy je použita mezinápravová plastová spojka.

5.2.4 Ideální trajektorie

Optimální jízda, kterou model projíždí trať, se nazývá ideální stopa. V zatáčkách je tvořena oblouky optimálního poloměru. Model má projet zatáčkou plynule s co možná nejméně změnou rychlosti. Bezchybná ideální stopa je velmi komplikovaná i pro závodníky, zajišťuje projetí trati s minimálními zásahy do regulace otáček motoru pohánějícího model.[12]

5.3 Návrh realizace na základě teorie

Kapitola zahrnuje návrh možné realizace po nastudování teoretického základu a dostupných komponentů. Uvedený postup je dostačující pro zmíněný autonomní provoz na Mini-Z závodní dráze. Postup výroby je uveden v chronologickém pořadí s ohledem na zdokonalování funkčního celku směřující k dosažení požadovaného cíle. Tato vylepšení eliminovala nežádoucí účinky a faktory pro provoz na dráze.

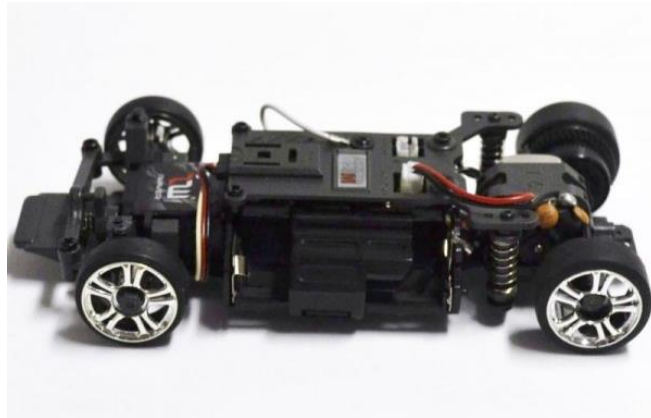
5.3.1 Požadované vlastnosti

Klíčová vlastnost pro stavbu ještě před samotnou realizací, je sestavit co nejmenší a nejjednodušší jednotku, která nesmí být komplikovaná po hardwarové a softwarové části. Klíčová je rychlost komunikace senzorů s mikro kontrolérem. Sensory měřící vzdálenost jsou na modelu umístěny ve 3 kusech. Přední snímá vzdálenost před automobilem, dva jsou umístěny na bocích. Senzor vzadu je vzhledem k zachování jednoduchosti a nízké hmotnosti vynechán. Nejdůležitější je snímat vzdálenost před automobilem a po stranách.

Automobil bude schopen reagovat na změny tratě před ním v podobě zatáček. Jakýkoliv kontakt s mantinelem není žádoucí a je započítán jako špatný pokus o projetí. Také je žádoucí projet závodní trať směrem pouze vpřed bez nutného couvání vzad.

5.3.2 Model vozidla 1:28

Pro demonstrativní účely a výrobu modelu, byl po pečlivém usouzení zvolen podvozek firmy IWAVER 1:28 (obr. 22.) od stejnojmenného čínského výrobce. Platforma je velmi kompaktní a variabilní. Dovoluje použít jakékoliv vlastní servo a svoji elektroniku, což je u takto malých modelů netypické. Podvozek je klíčovou částí, která nese veškerou elektroniku autonomního modelu. U podvozku je nutné hledět na každý detail, protože rozhoduje o výkonu a úspěšnosti řešení do budoucna. Podvozek je také klíčový v při projíždění zatáčkami, brždění, zrychlování a udržení stability na závodní dráze. O zatáčení modelu se stará vlastní servo motor, který svým natočením udává kolům směr. Zadní náprava je elektromotorem poháněna přes plastové ozubené ústrojí a přes diferenciál je přenášen točivý moment na dráhu.



Obrázek 22 – Podvozková platforma IWAVER boční pohled

(Zdroj: online https://speedrcmaresme.es/index.php?id_product=1077&controller=product)

Pro správné fungování modelu, je nutné, aby byl model umístěn na závodní dráze, která je vymezena po stranách mantinely. Při výskytu nečistot na pneumatikách může nastat nedotáčivost nebo snížený kontakt s vozovkou a nestabilita. Za tento fakt může snížená adheze neboli přilnavost pneumatik k dráze. Při velké míře nečistot se může stát, že se vozidlo neudrží na závodní trati.

Podvozek je oproti prodejní verzi značně odlehčen a zjednodušen. Box na 4 mikrotužkové baterie je vyjmut. Stejně také původní elektronika je nahrazena vývojovou deskou Arduino. Servo motor a pohonná jednotka je původní. Kvůli dalšímu snížení hmotnosti byl z automobilu vytažen tlumič nápravy a nahrazen vytištěným měkkým dorazem navrženým a vytištěným pomocí 3D tisku.

Ostatní elektronika, v první řadě platforma Arduino, baterie, regulátor rychlosti a přijímač, tvoří neméně důležitou část automobilu. Na zadní straně karoserie je LCD OLED display, který zobrazuje aktuální stav všech čidel, průběh vyhodnocování a aktuální natočení a požadavky na periferie. Pomocí LCD můžeme v reálném čase sledovat průběh vyhodnocování zatáčky a natočení servo motoru.



Obrázek 23 – Karoserie VW Transporter T1

(Zdroj: online <https://www.peckamodel.cz/wel12531wcrre-vw-t1-bus-1963>)

Pro test byla použita karosérie z hračky, která není součástí žádného výrobního portfolia známé firmy. (obr. 23) Její rozměry se přesně neshodují s parametry podvozku. K přizpůsobení byly navrženy a vytištěny díly na 3D tiskárně. Odlišný rozchod kol byl přizpůsoben vyfrézováním dostatečně velkých otvorů pro umístění nápravy. Další drobné úpravy k napasování karoserie byly taktéž provedeny vyfrézováním a vyvrtáním.

5.3.3 Další periferie

Byla použita tříkanálová vysílačka Flysky (obr. 24.), která zajišťuje komunikaci s modelem. Model FS-GT2B obsahuje spárovaný přijímač společně s vysílačem. Používá GFSK modulaci s AFHDS technologií, která přeskakuje zabraná pásma a hledá volnou frekvenci s ohledem na vnější podmínky. Důvod volby této vysílačky je také jednoduchost použití a také velmi malá energetická náročnost.



Obrázek 24 - Flysky GT2B s přijímačem

(Zdroj: online <https://www.bighobby.cz/Fly-Sky-FS-GT2B-3Ch-original-2-4GHz-prijimac-GR3E-d1145.htm>)

Bylo nutné zajistit trvalý přísun 5 V napětí do řídicích obvodů Arduina, regulátorů a servo motorů. K tomuto účelu byl použitý Step-Up měnič MT3608, který je schopný dodávat až 1 A proudu a 5 V stejnosměrného vyfiltrovaného napětí.

V poslední řadě je důležitý okamžitý přehled vyhodnocování vstupů snímaných z čidel a výstupů jdoucích do regulátoru a servo motoru.



Obrázek 25 - OLED panel 128x32 i2c

(Zdroj: online <https://www.vokolo.cz/oled-128x32-i2c>)

Tyto údaje jsou zobrazeny na OLED LCD (obr. 25.), který je umístěn v zadní části modelu. Display má rozlišení 128 x 32 bodů a jednobarevné modré podsvícení. Komunikuje s vývojovou deskou prostřednictvím I2C sběrnice, kterou poskytuje Arduino na pinech A4 a A5. Výhoda sběrnice I2C je v její naprosto jednoduché implementaci, která vyžaduje spojení se zemí, a dva datové vodiče SDA a SCL, kde SCL označuje hodinový signál, který určuje rychlost přenosu a SDA posílá pomocí sériové komunikace jednosměrně data do LCD. Poslední 4. vodič slouží k trvalému napájení 5 V, které je potřeba k samotnému fungování LCD. Použitý LCD display je unikátní svojí energetickou nenáročností, velikostí a jednoduchostí. V neposlední řadě je velice levný a snadno dostupný.

5.3.4 Ultrazvukové senzory

Ultrazvukový modul HC-SR04 byl zvolen s ohledem na aktuální skladovou dostupnost v e-shopu a vlastnosti, kterými modul disponuje. Jedná se o kompaktní modul o velikosti 44x20mm, který je již samostatně funkční a již není potřeba připojovat další součástky, které by vyžadovali zvláštní funkčnost. Jeho cena se pohybuje okolo 100Kč. Má celkem 4 piny z čehož je jeden napájecí, druhý slouží pro vstupní puls, další pro výstupní puls a samozřejmě také zem. Je možné ho napájet 5 V, nebo jiným stabilizovaným napětím o velikosti 3.3 V až 6 V.

Pro správnou funkci modelu je zapotřebí udělat vzdálenostní mapu překážek a mantinelů okolo modelu. Uvažujeme o použití až 5 čidel, která by mapovala a důkladně vyhodnocovala situaci před automobilem i po stranách automobilu.

Pro základní funkčnost a další testování byla zvolena konfigurace se 3 ks čidel, které jsou na vozidle umístěny následovně. První modul snímá situaci bezprostředně před vozidlem a varuje před blížícím se mantinelem nebo překážkou v přímém směru. Toto přední čidlo si označíme číslem 1.

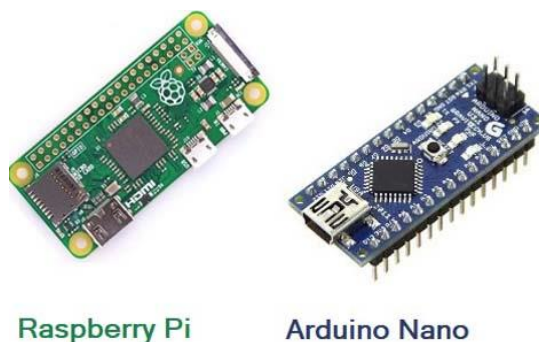
Další dva senzory jsou umístěny v náklonu podle osy vozidla pro snímání situace po stranách modelu mírně dopředu. Toto dovoluje reagovat na blížící se zatáčky ještě dříve, než se do ní model dostane. Čidla jsou umístěna v 30stupňovém

sklonu, které vykryvá mrtvé pásmo čidla, aby byla dosažena co nejvíce celistvá plocha snímání bez mrtvých míst, které čidlo nevykryje. Čidla náklonová si označíme čísly 2 a 3.

Dolní hrana čidla je umístěna přesně 45 mm nad zemí. Je to z důvodu rušivého prostoru, kde čidlo reaguje i na samotnou zem jako na překážku. Čidla jsou umístěna kolmo k zemi. Jiný náklon nepřipadá v úvahu. Vysílací a přijímací čidla jsou umístěna vedle sebe v horizontální poloze. Jsou tedy ve shodné vzdálenosti od země. Tento fakt je zvláště důležitý pro přesnost vyhodnocování a dále pro rychlost a kvalitu snímání. Pokud umístíme čidla jinak, nebo nějaký vnější činitel pohne nebo změni náklon čidel o nějaký hypotetický úhel, bude nutné čidla znovu kalibrovat. Zvláště pak mechanismus vyhlazování fiktivních a nereálných hodnot, se kterým si daný systém musí umět vypořádat. [13]

5.3.5 Arduino mikroprocesor

Mikroprocesor je hlavní vyhodnocovací část modelu vozidla. Na výběr byla možná varianta poháněná elektronikou Raspberry PI, konkrétněji variace Pi Zero, která je nejvíce odlehčená základní deska se stále obrovským potenciálem a výkonným procesorem. (obr. 26.)

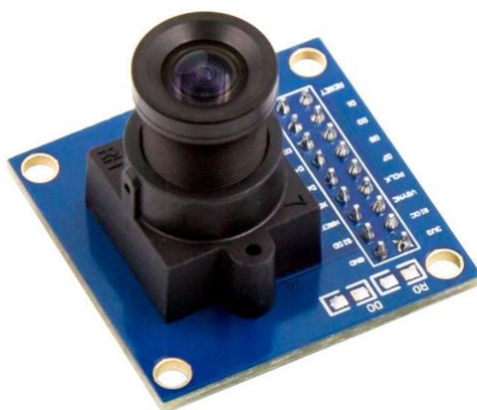


Obrázek 26 - Srovnání Raspberry Zero a Arduino Nano

(Zdroj: online <https://www.circuito.io/blog/arduino-vs-raspberry-pi>)

Další variantou je použít procesor Arduino NANO (obr. 26.), který disponuje oproti Pi Zero několikanásobně nižším výkonem a jednoduššími knihovny. Dále je jeho cena velmi nízká. V neposlední řadě Raspberry oproti Arduino velmi ztrácí,

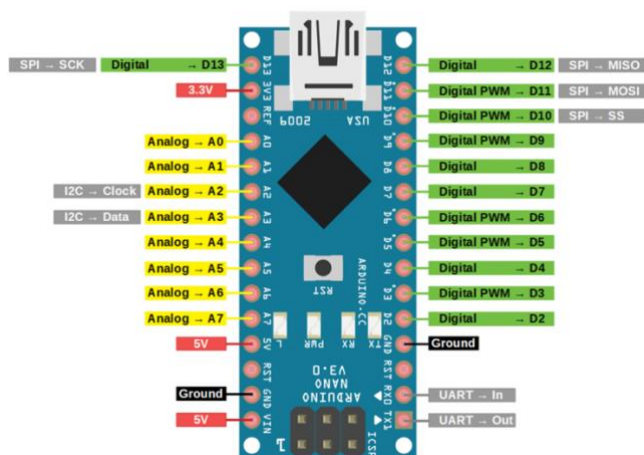
co se týče spotřeby elektrické energie. Arduinu s procesorem AVR stačí při nečinnosti jen cca 10 mA. Tento fakt je dán rozdílným typem mikroprocesoru. Kromě již zmíněné energetické náročnosti, vyšší pořizovací ceně a nutnosti komunikovat s periferiemi po 3,3 V datové větvi jako hlavním komunikačním napětím, byl nakonec zvolen Arduino Nano. Dále se Raspberry PI hodí spíše pro modely vozidel využívající rozpoznávání obrazu, kde je vysoký výkon žádoucí a Arduino nemůže nízkým výkonem konkurovat.



Obrázek 27 - Doplněk Arduino Camera

(Zdroj: online <https://arduino-shop.cz/arduino/900-arduino-vga-kamera.html>)

Doplňky na Arduino umožňující připojení externí kamery (obr. 27.) sice existují, avšak není v silách Arduina rozhodovat v reálném čase o nadcházející trajektorii vozidla, a ještě k tomu v dostatečné přesnosti a výkonosti.



Obrázek 28 - Arduino Nano 2.0 zapojení

(Zdroj: online <https://diyi0t.com/i2c-tutorial-for-arduino-and-esp8266>)

Mikrokontrolérová vývojová deska Arduino Nano ve verzi 2.0 (obr. 28.) disponuje mikroprocesorem ATmega 328 v nejmenším vyráběném pouzdře. Procesor běží maximální rychlostí 16 MHz s integrovaným oscilátorem. Je založen na architektuře AVR a jeho provozní logické napětí je 5 V. Disponuje 22 směsnými digitálními piny, a z toho 6 z nich umožňují PWM modulaci, která je použita na ovládání motoru vozidla. Zelenou barvou vyznačeno na ilustraci (obr. 28.). Dále disponuje 8 analogovými piny umožňujícími pomocí vestavěného 10 bit AD převodníků čítat a měřit napětí. Žlutou barvou znázorněno (obr. 28.). Arduino Nano disponuje 32 KB integrovanou FLASH pamětí na nahrání potřebného programu, 2 KB SRAM a 1 KB EEPROM paměti. Velká výhoda tohoto řešení je jednodušnost a jednoduchost použití. Nemusíme řešit USB převodník, protože je již v desce integrován. Nahrávání kódu do mikroprocesoru je otázkou chvíle. [14]

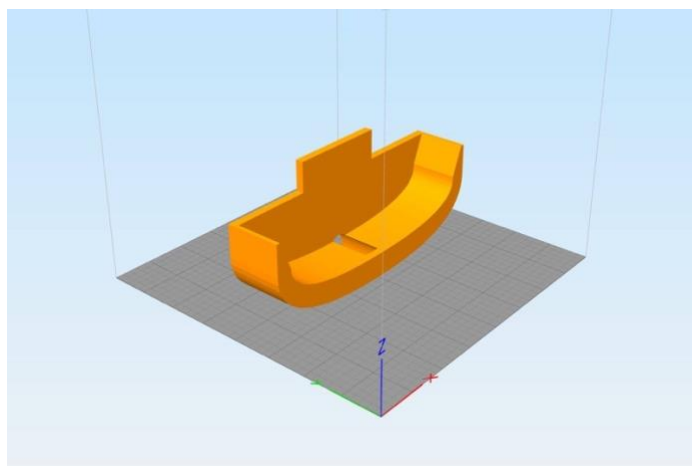
5.3.6 Tištěné díly pomocí 3D tiskárny

Pokud se díváme na konstrukční řešení a velikost modelu 1:28, je nám jasné, že nelze vměstnat všechnu elektroniku včetně větší baterie, mikrokontroléru a ovládání motoru a dalších nutných periferií do malého modelu karoserie Mini-Z. Na všechny doplňující výtisky byla použita 3D tiskárna Rebel sCube, která plně dostačuje všem komerčním i nekomerčním tiskům jakékoliv kvality.

Díly jsou nakresleny za pomoci svobodného modelovacího software openCAD. Poté jsou exportovány ve formátu STL, který definuje plochy objektů, ze kterých je poté složen 3D model. Při exportu do tiskárny je potřeba získat pohybová instrukční data, kterými se tiskárna bude řídit. Tyto data jsou uložena ve formátu gcode a vloženy na SD paměťové médium. Tomuto procesu se říká slicing, neboli nařezání a je klíčový pro bezchybný tisk. Pro účely bakalářské práce se osvědčil software Slicer, jehož výřez je ilustrován na ilustraci 29 a 30.[15]

Byla použita tisková struna Filamentum materiálu PETG oranžové barvy. Všechny díly byly vytištěny na 100% výplň pomalým tiskem, díky které jsou díly extrémně pevné a netrpí lámavostí a křehkostí.

Řešením je přeskládat elektroniku do velikostně většího balení, které je dostupné a zároveň nebude kazit dojem malého modelu. Řešení je koncept dodávky ve stejném měřítku. Karoserie VW Transporter se nabízí jako řešení, které pojme veškeré potřebné komponenty, a ještě trochu místa dokonce zbude. Nevýhodou tohoto řešení je prvotní naprostá nekompatibilita karoserie s nabízenou podvozkovou platformou.

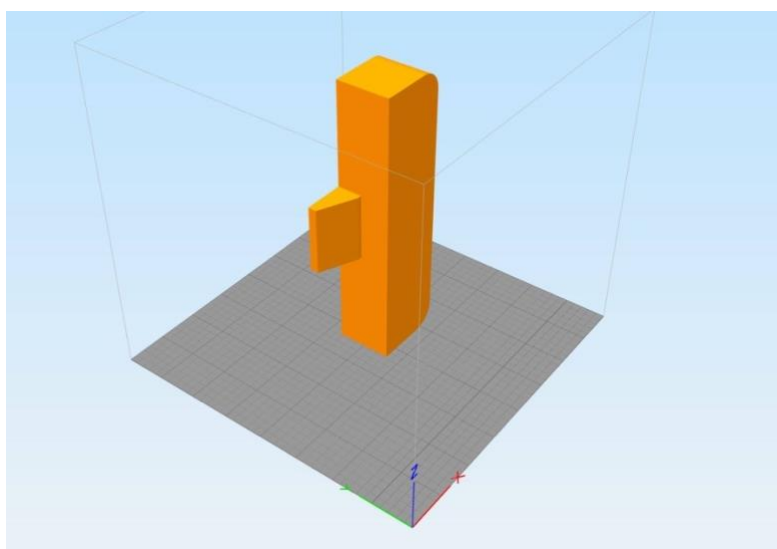


Obrázek 29 - 3D model předního nárazníku Slicer
(Zdroj: vlastní tvorba)

První věc, se kterou je nutné se vypořádat, je uchycení karoserie, tak aby byla pevně konstrukčně spojena se zbytkem modelu. K tomu účelu je zapotřebí karoserii mírně zvednout, odhadem o 1 cm výšky a vystředit. (obr. 29.) Jinak

karoserie nebude dosedat na nápravy a bude zapotřebí řešit větší vykrojení zadních náprav a u přední nápravy maximální dorazy rozmezí serva, které by se o karoserii zbytečně zadíralo. Je zapotřebí také uchytit modelovanou karoserii na zatím ještě neobsazené konstrukční mezery a vruty. Díly jsou modelovány s ohledem na maximální jednoduchost a funkčnost stavby.

Přední část vozidla je řešena do oblouku, který kopíruje karoserii a lícuje s ní. Uprostřed se uchycuje na jeden šroub M2, který trvale spojuje karoserii a tisknutý díl. Na podvozkovou platformu je díl uchycen pomocí vysouvacího mechanismu, který se do karoserie nasune pod úhlem a zajistí taktéž šroubem M2 z každé strany. Šrouby se zapouští do výtisku kvůli možné kolizi s povrchem. Přední náprava se také počítá jako deformační zóna tohoto modelu. Z tohoto důvodu je zachován původní tvar nárazníku a celá styčná dosedací plocha přesně kopíruje původní řešení.



Obrázek 30 – 3D model zadního nárazníku Slicer

(Zdroj: vlastní tvorba)

Zadní část karoserie lemuje 3D výtisk ve tvaru velkého nárazníku (obr. 30.), který lícuje se zadní stranou karoserie a přidává ji pevnost. Uprostřed je zacvakávací mechanismus podobný přední části, který se taktéž vsouvá do podvozkové platformy a zajišťuje se ve prostřed šroubem M2, který se zapouští do výtisku

z důvodu možného kontaktu s povrchem. Toto řešení zadního nárazníku také řeší odlišný mezinápravový rozchod původního řešení Mini-Z.

V poslední řadě je za pomoci 3D tisku vyřešen mechanismus odpružení zadní nápravy. Staré řešení využívalo části karoserie, které již nemohly být použity pro současné řešení z důvodů úspory místa. Jedná se hlavně o odpružení zadní nápravy, které je nově vyřešeno systémem listových per, které se používá například v nákladních automobilech. V zadní části je vytištěn dlouhý list, který dosedá na zadní nápravu a díky vlastnostem vytištěného materiálu se na delší vzdálenost mírně ohýbá.

Řešení je dostačující, jelikož nebrání plnému využití zadní nápravy jako hnací nápravy. Toto řešení také dovoluje jednoduše zvýšit výšku zadní nápravy nad povrchem pomocí jednoduchého přidání dalšího listu, který bude tvrdší než ten předchozí.

5.4 Programování

Programování mikroprocesoru Arduino probíhá pomocí programu Arduino IDE, což je oficiální název pro integrované vývojové prostředí pro mikroprocesory Arduino a další jsou většinou založené na architektuře AVR. Díky prostředí Arduino IDE můžeme komunikovat s mikroprocesorem pomocí USB portu počítače a přímo do něj nahrávat kód z vývojového prostředí. IDE obsahuje pokročilý textový editor, který nám pomáhá při psaní a následném debugingu kódu. Programováno je v jazyce C++ nebo C. Po vytvoření nového projektu se začíná psát do první části metody `void setup()` sloužící k zapsání kódu, který se spustí pouze jednou. Druhá část programu se zapisuje do smyčky `void loop()`, která probíhá opakovaně až do vypnutí nebo odpojení Arduina. Tyto dvě části jsou důležité, což znamená, že při smazání některé z nich, kompilátor napsaný kód nezkompiluje a neumožní ho nahrát do Arduina.[16]

5.4.1 Hlavní smyčka

Ještě před spuštěním hlavní smyčky programu, je zapotřebí inicializovat všechny knihovny a vyrobit konstruktory. Toto vytvoříme pomocí příkazu `include`. Postupujeme od shora dolů.

```
#include <PID_v1.h>
```

```
#include <U8glib.h>
```

```
#include <Servo.h>
```

V první řadě inicializujeme LCD display, který zavoláme příkazem a pojmenujeme ho `mujOled`.

```
U8GLIB_SSD1306_128X32 mujOled(U8G_I2C_OPT_NONE);
```

Dále je potřeba inicializovat knihovnu PID, která se stará o chod regulace rychlosti modelu. Zavoláme ji příkazem a pojmenujeme ji `myPID`.

```
PID myPID(&InputFreq, &Output, &SetSpeed, consKp, consKi, consKd, P_ON_M, REVERSE);
```

Knihovna pobírá parametry vstupní frekvence, nastavované rychlosti, a samotné konfigurace PID hodnot. Poslední atributy udávají použitý mód, který nám říká, že výstupní hodnota je měněna pouze s aktuální naměřenou hodnotou bez časových dat o průběhu minulé hodnoty. Reverzní mód jako takový používáme proto, aby motor nemusel inkrementovat celý použitelný rozsah od 0 až do 255 postupně. Bez této drobnosti by trvalo přibližně 25 s, než by se model dal do pohybu. Jednodušší je vzít maximální hodnotu 255 a PID nechat automaticky vypočítávat inverzní hodnotu do kýžené hledané rychlosti.

Output poté vrátí hodnotu od 0 do 255 podle toho, jaké rychlosti má být dosaženo.

```
double consKp = 3, consKi = 4, consKd = 0;
```

Hlavní smyčka `loop()` nese všechny důležité metody v chronologickém pořadí pro co nejvyšší přehlednost kódu. Všechny klíčové funkce jsou rozepsány do metod, které naměřené hodnoty buď upravují anebo vytváří nové hodnoty, které se poté dále použijí.

```
myPID.SetMode(AUTOMATIC);
```

```
myPID.SetSampleTime(10);
```

Funkce nám říká, že `myPid()` bude nabývat automatických hodnot 100 x za vteřinu. Hodnota 10 je uvedena v milisekundách a označuje po jakém časovém úseku se má znovu spustit.

```
servo.attach(2);
```

Na závěr je nutné se postarat o správné přiřazení funkce datového vodiče do serva, které je korektně alokováno na pin 2 jako PWM výstup.


```

//loop
void loop() {

    steeringConstOld = steeringConst;

    String zprava = "";
    zprava3 = "";

    modeSelect();

    if (!mode) {

        // mod manual
        steeringConst = map(pulseIn(steering, HIGH), 950, 1950, 30, 140);
        throttleConst = map(pulseIn(throttle, HIGH), 950, 1950, 0, 255);

        servo.write(steeringConst);
        throttleM();

    } else {

        // automatic mode
        moded = false;
        calculDirection();

        isSteerChanged();

        applySteerLimit();

        servo.write(steeringConst);

        calculateAndWriteSpeedLimited();

        readDistFront();
        readDistLeft();
        readDistRight();
        readDistRight2();
        readDistLeft2();

        readSpeed();

    }
}

```

Obrázek 31 – Opakující se smyčka loop()

(Zdroj: vlastní tvorba)

Hlavní smyčka programu (obr. 31.) začíná uložením data o posledním úhlu natočení z minulého průchodu. Pokud se jedná o první průchod, pak je do řetězce vložena výchozí hodnota 88, která symbolizuje přesnou polovinu rozsahu. Servo reaguje na hodnoty od 0 až do 180 stupňů. Hodnota prostřední je tedy 90. Drobné výchyly hodnot jsou způsobeny nedokonalostmi servo motoru a přední nápravy.

Dále začínáme na čistém nepopsaném řetězci pro LCD display, kterému říkáme zprava a zprava3. Tyto řetězce jsou použity na konci loop() metody. Tyto

řetězce jsou na začátku cyklu prázdné, teprve postupně jsou tyto zprávy naplňovány podle aktuálních hodnot.

Následuje funkce `modeSelect()` která nám zjišťuje aktuální mód ovladače. Pokud je aktivován běh automatického automobilu, funkce mění hodnotu proměnné `mode` na 1, kterou poté dále porovnává pomocí větve `if`.

Manuální operace probíhají za asistence řidiče, kdy mikro kontrolér přejímá hodnoty z přijímače a dále je pomocí metody `map`, upravuje na použitelné rozsahy hodnot. Následně je exportuje do serva a regulátoru. O to se stará metoda `servo.write()` a `throttleM()`. Vstupy jsou pomocí funkce `pulseIn()` zapisovány do hodnot od 0 až do 255. Funkce počítá čas v mikrosekundách symbolizující čas vlny od horní hrany do další horní hrany.

Ve větvi `else` je popsána část kódu, která se koná, pokud je aktivován automatický mód, který vypočítává trajektorii. Do proměnné `moded` je zapisován údaj, který reaguje na změny ve směru automobilu. Pokud model vykonává průjezd zatáčkou, proměnná se nastavuje na 1 a když zatáčka skončí, vrací se na 0. Dále je proměnná využita při čtení konkrétní vzdálenosti, kdy je vyčítání a rychlost ovlivněna tímto parametrem. Metoda `calculateDirection()`, `isSteerChanged()` a `applySteerLimit()` jsou klíčové metody pro správnou funkci modelu. Jsou napsány tak, aby vracely hodnoty, které jsou hned použitelné pro export do serva motoru a na výpis do LCD. Celý vyhodnocovací systém je popsán právě v těchto metodách.

Metoda `calculateAndWriteSpeedLimited()` počítá aktuální rychlost modelu a upravuje jízdu modelu k vyšší plynulosti. Předposlední metoda `readDist()` je určena k importu aktuálních dat jízdy z čidel. Každý loop průchod jsou tyto metody spouštěny opakovaně. Poslední metoda `readSpeed()` načítá aktuální rychlost modelu, který se pohybuje po dráze. Pokud není zaznamenán pohyb, metoda vrací nulovou rychlost a PID s touto hodnotou v příštím průchodu počítá.

Poslední metody, které jsou pravidelně spouštěny, jsou metody na vykreslení textu na LCD displeji. Řetězec se před exportem pomocí funkce `printf()` sestaví a po předchozím smazání celého LCD znovu napíše. LCD píše řetězce zleva podle jednotlivých řádků. Pokud přeskakujeme na různé místa na LCD, je nutné znát souřadnice vykreslovaného bodu.

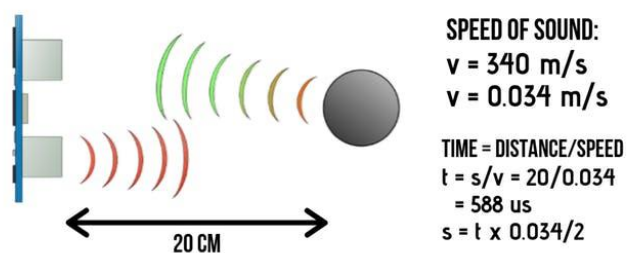
5.4.2 Popis a funkce jednotlivých částí

```
void readDistFront() {  
  
    digitalWrite(12, LOW);  
    delayMicroseconds(2);  
    digitalWrite(12, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(12, LOW);  
    // pomocí funkce pulseIn získáme následně  
    // délku pulzu v mikrosekundách (us)  
    odezva = pulseIn(11, HIGH, 200000);  
    // přepočet získaného času na vzdálenost v cm  
    if (odezva / 58.31 < 99) {  
  
        if (odezva / 58.31 < 1) {  
            vzdalenost = Input;  
        } else {  
            vzdalenost = odezva / 58.31;  
            Input = vzdalenost;  
        }  
    } else {  
        vzdalenost = 99;  
        Input = vzdalenost;  
    }  
}
```

Obrázek 32 - Metoda readDistFront()

(Zdroj: vlastní tvorba)

Vyčítání z předního senzoru je realizováno pomocí metody `readDistFront()`. (obr. 32.) Metoda nejprve vyšle do modulu impulz, dlouhý přesně 5 ms a poté pomocí metody `pulseIn()` vyčte odezvu. Proměnná `odezva` ukládá počet mikrosekund, se kterými dále ve výpočtu počítáme.



Obrázek 33 - Výpočet vzdálenosti pomocí odrazu

(Zdroj: <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>)

Rychlost zvuku 346,3 m*s-1 při teplotě vzduchu 25 °C znamená, že za 1 ms urazí pulz v metrech 346,3/1000000 což je 0,0003463 metru. Převáděno na cm to je 0,03463 cm / ms. Vzhledem k tomu, že signál jde od čidla k předmětu, kde se odrazí a zase zpět, musíme tuto vzdálenost ještě vydělit číslem 2. Výsledek je, že se vzdálenost bude rovnat počtem mikrosekund násobených číslem 0,017315 anebo vydělený 58.31. (obr. 33.) Výsledek porovnáváme s maximálními měřitelnými údaji a rovnou s nesmyslnými hodnotami počítat nebudeme.[17]

```
void calculDirection() {  
    int steeringConstCurr;  
  
    if (TbocniR) {  
        moded = true;  
        steeringConstCurr = (95 - ((50 - bocniR)));  
        steeringConst = steeringConstCurr;  
    }  
  
    if (TbocniL) {  
        moded = true;  
        steeringConstCurr = (((50 - bocniL)) + 95);  
        steeringConst = steeringConstCurr;  
    }  
}
```

Obrázek 34 - Metoda calculDirection()

(Zdroj: vlastní tvorba)

Metoda calculDirection() se stará o logiku vyhodnocování tvaru trati okolo vozidla. (obr. 34.) Do proměnné steeringConstCurr se ukládá aktuální stav všech vyhodnocovacích podmínek tak jak se za sebou provádějí. Dále jsou podmínky sestaveny podle míry přesnosti úsudku správného rozhodnutí. V praxi to znamená, že pokud máme k dispozici údaje o protějších bočních čidlech a model se pohybuje po rovné dráze, vždy se použije tato podmínka. V tu chvíli jsou ostatní irelevantní.

První úsek kódu náleží podmínce bočního čidla v přední části vozu. Podmínky mají velmi podobnou funkci. Do proměnné steeringConstCurr se ukládá hodnota natočení serva. Hodnota bocniR přichází jako údaj uvedený v cm. Jedná se o vzdálenost od překážky bočního čidla v přední části vozu. Hodnota 95 je zde uvedena jako prostřední hodnota servo motoru a jedná se o nulovou rovnoběžnou pozici. Hodnota 50 cm symbolizuje střed hypotetické dráhy, jejíž šířka včetně automobilu je 90 cm. Tento úsudek se používá pouze pokud není jiný kvalitní odhad

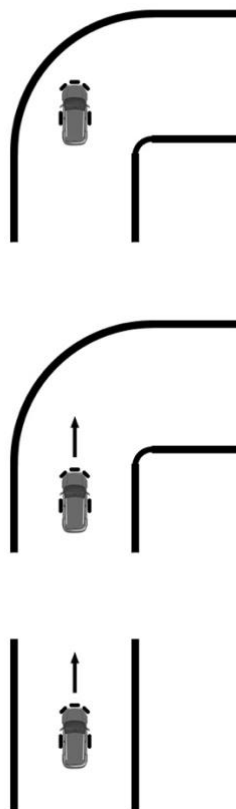
dráhy před modelem a nejsou údaje z čidel, které jsou umístěny naproti sobě. Poté je nutné tedy hodnoty odhadovat.

```
if (TbocniR && TbocniL) {  
  if (vzdalenost > 40) {  
    moded = true;  
    double ins = (bocniR - bocniL) * 0.5;  
    steeringConstCurr = ins + 88;  
    steeringConst = steeringConstCurr;  
  }  
}
```



Obrázek 35 - Úsek metody bočních čidel

(Zdroj: vlastní tvorba)



Obrázek 36 - Průběh nájezdu do zatáčky

(Zdroj: vlastní tvorba)

Ilustrace (Obr. 36.) znázorňuje použití pouze bočních senzorů na modelu. Pokud uvažujeme pouze data o vzdálenosti z každé stran, vozidlo bude moci projíždět rovnými úseky. Pokud vozidlo projíždí rovným úsekem a jsou k dispozici údaje z bočních senzorů a vzdálenost před automobilem větší než 40 cm, automaticky se ostatní senzory vyřazují a přebírají se data právě jen z bočních senzorů. Sestavení podmínek je patrné z ilustrace (obr. 35.) Toto je nejjednodušší úsudek řízení modelu a výstupu dat do serva. Označíme ho jako A.

Jak napovídá ilustrace (obr. 36.), toto řešení je nejkvalitněji použitelné, pokud automobil jede po rovné dráze, pokud se s modelem přiblížíme k zatáčce a vzdálenost je menší než 40 cm, tyto čidla se již prioritními nestávají a používá se řešení níže. (obr. 37.)

Pokud situaci upravíme a model umístíme do vnitřku zatáčky, je uvažování již kompletně zkreslené a není možné vytvořit kvalitní úsudek o pozici automobilu na dráze. [18]

```
if (TvzdalenostR) {
    moded = true;
    steeringConstCurr = 85 - ((40 - vzdalenostR) * 2.5);
    steeringConst = steeringConstCurr;
}

if (TvzdalenostL) {
    moded = true;
    steeringConstCurr = ((40 - vzdalenostL) * 2.5) + 85;
    steeringConst = steeringConstCurr ;
}
```

Obrázek 37 - Úsek metody čidel náklonových

(Zdroj: vlastní tvorba)

Část kódu, která rozhoduje o pohybu modelu, uvažuje parametry náklonových čidel. Dále se stará o správné fungování modelu hlavně při nájezdu do oblouku. (obr. 37.) Pokud rozebereme obrázek výše (obr. 37.), podmínky fungují hlavně, pokud čidla neposkytují dostatečně kvalitní údaje z obou stran. Poté je nutné hodnoty odhadovat a počítat s mírnými chybami v úsudku. Tyto podmínky jsou nutné ke správnému fungování, protože vždy není možné v průběhu jízdy poskytnout plynulé údaje. Nastavení těchto podmínek je tedy klíčové.

```

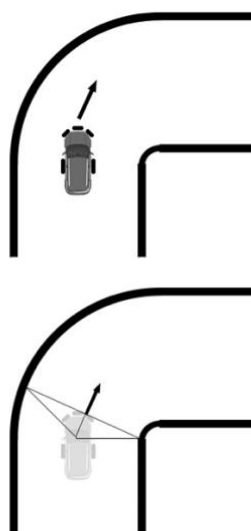
if (TbocniR && TvzdalenostL){
  if (vzdalenostL<60){
    moded = true;
    steeringConstCurr = bocniR - vzdalenostL + 98;
    steeringConst = steeringConstCurr;
  }
}

if (TvzdalenostR && TbocniL){
  if (vzdalenostR<60){
    moded = true;
    steeringConstCurr = vzdalenostR - bocniL + 98;
    steeringConst = steeringConstCurr;
  }
}

```

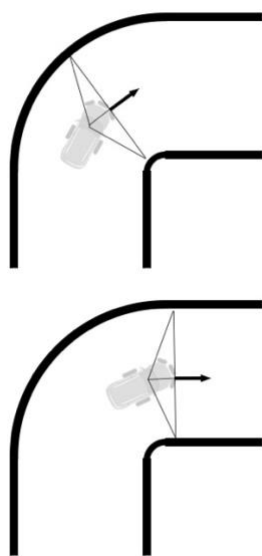
Obrázek 38 - Úsek metody kombinující čidla
(Zdroj: vlastní tvorba)

Podmínky jsou více přesné, když uvážíme hodnoty z obou čidel protistraně umístěných. (obr. 38.) Toto řešení je limitováno čtením vzdálenosti z předního čidla. Pokud je vzdálenost větší než 60 cm, toto řešení není použito. V praxi se toto řešení promítne právě v situacích, kdy model najíždí do oblouku a je na čase natočit úhel přední nápravy na správnou stranu oblouku. Pokud je vozidlo nasměrováno správně ještě před samotným obloukem, je průjezd zatáčky dokonalý. Na ilustraci (obr. 39.) správné natočení přední nápravy ukazuje černý kurzor směřující do vnitřku zatáčky.



Obrázek 39 - Průjezd zatáčkou před najetím
(Zdroj: vlastní tvorba)

Popisujeme metody, které nám pomáhají do zatáčky najíždět ve správném úhlu. Jak již bylo řečeno, je nutné, aby model před sebou měl dostatečný malý prostor neboli byl tedy v zatáčce. Ilustrace popisuje použití hypotetického tupého trojúhelníku, který je převeden do soustavy dvou pravoúhlých trojúhelníků. Servo nyní počítá s novými přesnějšími hodnotami. Vždy má tento trojúhelník stejný úhel, tedy 30 stupňů, který svírá s automobilem. Trojúhelník se liší jen svými vzdálenostmi stran. Toto je druhý úsudek řízení modelu a výstupu dat do serva. Od prvního úsudku se liší hlavně jeho složitosti. Označíme ho jako B.



Obrázek 40 - Průjezd zatáčkou po najetí

(Zdroj: vlastní tvorba)

Další průběh zatáčkou, je zobrazen na ilustracích. (obr. 40.) Pokud nastane problém a není automobil schopný vyčítat hodnoty z čidel na obou stranách, model použije hodnotu jen z jedné strany. Průjezd poté není úplně čistý. Ilustrovaný model výše (obr. 40.) zobrazuje situaci uprostřed oblouku. Používáme zde stejný tupý trojúhelník ke správnému bezchybnému projetí. Po výjezdu z oblouku a zaznamenání vzdálenosti před vozidlem delší než 60 cm, přestává platit podmínka B a model se pohybuje po rovinné trajektorii s jednoduchým A úsudkem. Zároveň je poté upravena rychlost modelu.

Nebyla zmíněna skutečnost, že pokud je model v zatáčce, tak není potřeba dosahovat vysoké rychlosti. Pokud vzdálenost před vozidlem je menší než 60cm, je rychlost modelu upravena na 5 cm / s. Původní plná rychlost, které dosahuje na rovném úseku je rovna 10 cm / s. Řízení rychlosti modelu vyhodnocuje knihovna PID library a metoda `calculateAndWriteSpeedLimited()`. Dále tyto metody jsou rozděleny na `calculateAmdWriteSpeedLimited()` a `calculateAndWriteSpeed()`.

5.5 Mechanismus řízení rychlosti

Metody `calculateAmdWriteSpeedLimited()` (obr. 41.) a dále pak `calculateAndWriteSpeed()` slouží k výpočtu a také k řízení rychlosti modelu. V běhu programu jsou použity obě a záleží na aktuálním rozhodnutí o dostupné vzdálenosti která je před modelem. Metoda `calculateAmdWriteSpeedLimited()` vrací a upravuje rychlost s ohledem na vzdálenost před vozidlem. Vzdálenost v této metodě musí být větší než 10 cm. Druhá z nich nerozlišuje vzdálenost a rychlost. Obě metody vpisují hodnoty přímo do regulátoru rychlosti na pinu 3.

```
void calculateAndWriteSpeedLimited() {
    if (vzdalenost > 10 && (!TvzdalenostL || vzdalenostL > 10) && (!TvzdalenostR || vzdalenostR > 10)) {
        if (vzdalenost < 50) {
            digitalWrite(9, 1);
            digitalWrite(10, 0);
            myPID.Compute();
            analogWrite(3, 190 - Output * LimitCons);
            zprava3 += ">";
            zprava3 += (int)(Output / 2.55);
        } else {
            digitalWrite(9, 1);
            digitalWrite(10, 0);
            myPID.Compute();
            analogWrite(3, 190 - Output * LimitCons);
            zprava3 += ">";
            zprava3 += (int)(Output / 2.55);
        }
    } else {
        digitalWrite(9, 0);
        digitalWrite(10, 0);
        analogWrite(3, 200);
        zprava3 += "----";
    }
}
```

Obrázek 41 - Metoda `calculateAndWriteSpeedLimited()`

(Zdroj: vlastní tvorba)

Metoda uvnitř rozlišuje vzdálenost modelu na závodní trati. (obr. 41.) Rozlišovací vzdálenosti jsou rozděleny do dvou bloků. Vzdálenost do 50 cm a více než 50 cm. V podmínce se rovnou zapisují hodnoty do regulátoru. Funkce podmínky je závislá na více parametrech. Na pinu 3 se zapisuje požadovaná rychlost. Metoda `myPid.Compute()` vypočítává aktuální hodnotu rychlosti a poté je tato hodnota použita v proměnné `Output`. Piny 9 a 10 jsou použity na změnu orientace motoru. Lze měnit polaritu, a tedy směr jízdy vozidla. Pokud je na pin 9 přiveden logický signál 1 a na pin 10 logický signál 0, vozidlo se pohybuje směrem dopředu.

V neposlední části je popsána jízda bržděná, která automobil postupně zastavuje. Pokud je na pin 9 a 10 přivedena logická 0, motor začíná brzdit z důvodu připojení mínusové větve na oba motorové vstupy.

5.5.1 Konfigurace PID knihovny

Použitá knihovna PID (obr. 42.) je nejjednodušší implementovatelná regulace, kterou můžeme na Arduino použít. Její funkčnost je triviální a je omezena na jen několik nastavovaných parametrů. V první řadě na proporcionální, integrační a derivační členy. Ty jsou poté použity v metodě `myPID()`.

```
//PID configure
double consKp = 3, consKi = 4, consKd = 0;

double InputR, InputL;
double Setpoint, Input, Output, Output2;
int Offset;

double SetSpeed = 5;
double InputFreq;

PID myPID(&InputFreq, &Output, &SetSpeed, consKp, consKi, consKd, P_ON_M, REVERSE);
PID myPIDR(&Input, &Output2, &Setpoint, consKp, consKi, consKd, P_ON_M, REVERSE);
```

Obrázek 42 - Konfigurace PID knihovny

(Zdroj: vlastní tvorba)

Proměnná `consKp` je nastavena na hodnotu 3 a jedná se o proporcionální složku regulátoru. `consKi` náleží integrační konstantě a má hodnotu 4, proměnná `consKd` symbolizuje derivační konstantu a pro naše účely a zjednodušení je nastavena na 0. Nemá tedy žádný vliv na ovládání rychlosti modelu. Nebo alespoň

zanedbatelný a během testování není potřeba vzhledem na rychlost modelu počítat a nastavovat přesné PID parametry. Jejich vliv na výstupní výkon a kvalitu výstupu není příliš výrazný.

5.5.2 Detekce rychlosti

Kontrola a detekce rychlosti je obsluhována metodou `readSpeed()` (obr. 43.), která je zodpovědná za korektní vyčtení hodnot z otočného snímače rychlosti umístěném na zadní nápravě.

```
void readSpeed() {  
    frequency = 100000 / (pulseIn(13, HIGH, 100000));  
  
    if (frequency < 1000) {  
        freq = frequency;  
        InputFreq = frequency;  
    }  
  
    if (frequency > 10000) {  
        if (counter > 2) {  
            freq = 0;  
            InputFreq = 0;  
        }  
    }  
  
    counter += 1;  
} else {  
    counter = 0;  
}  
}
```

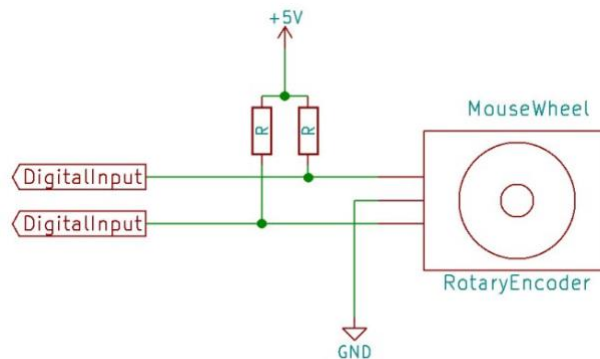
Obrázek 43 - Metoda `readSpeed()`

(Zdroj: vlastní tvorba)

Hlavní proměnná `frequency` pomocí metody `pulseIn()`, která vrací dobu, po kterou se stav na určitém pinu nachází na dolní hranici. A při přechodu do dalšího stavu vrací počet milisekund. Ostatní metody a podmínky slouží pro stabilizování hodnot, při chybném měření. Dále uvažujeme podmínku nad příliš vysokými hodnotami a při větší četnosti tyto hodnoty nebereme v úvahu. Hodnota proměnné

freq je vyfiltrovaná. Tuto hodnotu používáme do PID regulátoru jako směrodatnou hodnotu pohybu modelu vozidla.

Čidlo vyžaduje zvláštní vyčítání hodnot, protože jejím výstupem jsou 3 vodiče. Zapojení čidla k Arduino je realizováno pomocí vodičů dvou a je zapotřebí do čidla přivést napájení.



Obrázek 44 - Rotační mikropínač

(Zdroj: vlastní tvorba)

Hlavní vstup čidla do vývojové desky Arduino je přiveden na pin 13. Ilustrace (obr. 44.) popisuje zapojení, které je úplné a je žádoucí při použití, kde je vyžadována vysoká přesnost. Pokud sloučíme výstupní piny do jednoho digitálního výstupu, dostaneme výstup, který je méně náchylnější na překmity, a přitom nám stále stačí jeho rozlišovací schopnost. Takto zapojené čidlo detekuje 8 stavů v průběhu jedné úplné provedené otáčky. [19]

5.6 Snímání okolí pomocí 3 ultrazvukových čidel

První vytvořený funkční model automobilu nesoucí 3 ultrazvuková čidla zvýrazněná na schématu (obr. 45.) červenou barvou, je jako celek samostatně funkční omezený autonomní jízdy. Jsou umístěny v přední části modelu.



Obrázek 45 - Model se 3 aktivními čidly v přední části vozu
(Zdroj: vlastní tvorba)

5.6.1 Požadované vlastnosti a princip

V prvním případě je tento systém vhodnější pro projetí zatáčkou. Kvůli absenci postranních čidel, automobil nejspíše nebude zvládat rovné úseky, kde je potřeba snímání z bočních okolních senzorů. Dále je žádoucí od této konfigurace najetí do oblouku zatáčky.

Tato konfigurace čidel je plně dostačující k bezproblémovému projetí horizontu zatáčky a najetí správnou stranou do oblouku.

Výhoda této konfigurace spočívá v mnohem nižší váze modelu, který nemusí nést boční čidla a uchycení bočních čidel. S tím souvisí menší energetická náročnost a delší výdrž na baterii. Druhým faktem je, že pokud využijeme řešení 3 základních čidel, model je schopný zpracovávat loop cyklus 15x za vteřinu.

5.6.2 Testování v praxi

Výhoda této konfigurace spočívá v mnohem nižší váze modelu, který nemusí nést boční čidla a uchycení bočních čidel. S tím souvisí menší energetická náročnost a delší výdrž na baterii. Druhým faktem je, že pokud využijeme řešení 3 základních čidel, model je schopný zpracovávat loop cyklus 15x za vteřinu. Tabulka (tab. 2.) zobrazuje průjezd modelu pravotočivou zatáčkou.

Počet průjezdů	Úspěšnost	Komentář
1. průjezd	Ano	Velmi nízká rychlost, špatné najetí do oblouku, model se drží cca 10 cm od ideální trajektorie.
2. průjezd	Ne	Vysoká rychlost, kontakt s mantinelem.
3. průjezd	Ne	Kontakt s mantinelem.
4. průjezd	Ne	Zatáčka nezaznamenána.
5. průjezd	Ne	Kontakt s mantinelem, vysoká rychlost
6. průjezd	Ne	Drží se u mantinelu a následný kontakt, nezaznamenána překážka.
7. průjezd	Ano	Nízká rychlost nájezdu, ideální jízda,
8. průjezd	Ne	Kontakt s mantinelem.
9. průjezd	Ano (korekce)	Příliš vysoká rychlost, nutná korekce rychlosti, 5 cm od ideální trajektorie.
10. průjezd	Ne	Kontakt s mantinelem.

Tabulka 2 - Testování průjezdu pravotočivou zatáčkou se 3 čidly

Z tabulky je patrné, že bude nutné použít řešení více čidel. Výsledky měření nebyly dostatečné a je nutné zpracovat jiný návrh. Úspěšnost na prahu 30 % není pro model uspokojivá na plně autonomní jízdu.

5.7 Snímání okolí pomocí 5 ultrazvukových čidel

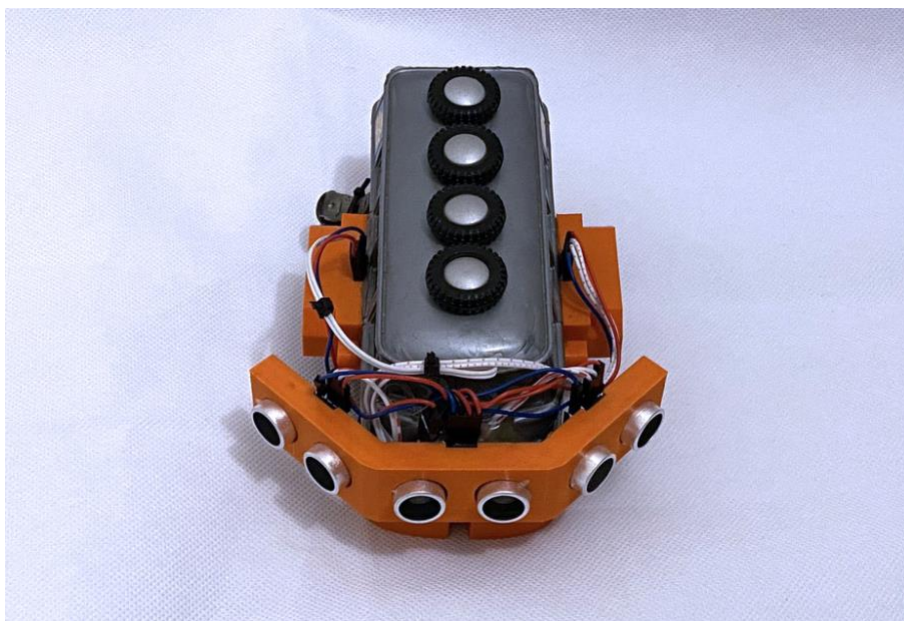
Druhý vyrobený model zapracovává do své konstrukce 5 funkčních čidel monitorujících vzdálenost okolo vozidla. (obr. 46.)



Obrázek 46 - Model s 5 aktivními čidly v přední části vozu

(Zdroj: vlastní tvorba)

Tato konfigurace modelu s 5 aktivními čidly je již finální provedení modelu automobilu z hlediska obsažených periférií. Reálný testovaný model je zdokumentován na fotografii (obr. 47.)



Obrázek 47 - Testovaný finální model s 5 aktivními čidly

(Zdroj: vlastní tvorba)

5.7.1 Požadované vlastnosti a princip

Oproti variantě se 3 čidly, přidává čidla boční na měření aktuální vzdálenosti od mantinelů bezprostředně okolo modelu. Toto řešení také přidává na váze a

zkracuje výdrž na baterii. Z hlediska systémové náročnosti obsazuje o 10 % více paměti při běhu.

Od tohoto řešení se očekává menší chybovost. Více vstupních parametrů může zásadně ovlivnit rychlost negativním směrem. Dále je zapotřebí čidla spouštět postupně, protože jejich frekvence se navzájem ruší. Není tedy možné paralelní vyhodnocování situace z více vstupních periférií, a to ani pokud se čidla vzájemně nevidí a jsou orientovány směrem od sebe.

Postupné zvyšování vstupních čidel je velmi omezené. Vývojová platforma nenabízí dostatek výkonu k provozování dalších periférií. Důvod pro použití 5 čidel, je pokrýt co největší plochu oblouku kolem vozidla v dopředném směru, tak aby plocha nepokrytých míst, byla co možná nejmenší.

5.7.2 Testování v praxi

Oproti verzi se 3 čidly je toto řešení robustnější a daleko více se podobá autonomnímu provozu modelu. Smyčka je prováděna v cyklu loop 10x za vteřinu. Dopad na výkon je vysoký, což bylo ale předpokládané. Výsledky jsou zachyceny v tabulce níže. (tab. 3.)

Počet průjezdů	Úspěšnost	Komentář
1. průjezd	Ne	Vysoká rychlost najetí do oblouku, kontakt s mantinelem.
2. průjezd	Ano	Ideální trajektorie s mírnými chybami v toleranci.
3. průjezd	Ano	Ideální trajektorie.
4. průjezd	Ne	Zvýšená rychlost modelu, průjezd ideální.
5. průjezd	Ano (korekce)	Kontakt s mantinelem, následná korekce.
6. průjezd	Ano (korekce)	Drží se u mantinelu a následný kontakt, korekce poté ideální průjezd.

7. průjezd	Ano	Ideální průjezd, pomalejší jízda.
8. průjezd	Ano (korekce)	Ideální trajektorie s mírnými chybami.
9. průjezd	Ne	Příliš vysoká rychlost, nutná korekce rychlosti, náraz do mantinelu.
10. průjezd	Ano (korekce)	Ideální trajektorie, pomalejší jízda.

Tabulka 3 - Testování průjezdu pravotočivou zatáčkou s 5 čidly

Bohužel je zde znát vysoký dopad na výkonu a rychlosti vyhodnocování. Testování končí s výsledkem 70 % úspěšnosti, což je přijatelné pro účely testování na závodní dráze.

Pokud bychom uvážili rychlost, s jakou model pracuje. Dá se říci z hodnot zaznamenaných v tabulce, že sice jsou v modelu přesnější a nepochybně kvalitnější data od vnějších čidel, ale chybovost a špatné uvážení při průjezdu tvoří spíše znatelně menší rozhodovací rychlost průchodu v samotném Arduinu.

5.8 Vyhlazování sbíraných dat

Každá jízda má zcela odlišný charakter. Není určený originální postup, který je modelem naučený. Z tohoto důvodu je uvažování nad jistou chybovostí z naměřených hodnot žádoucí.

Nad vyhlazováním je třeba se zamýšlet ještě před samotnou realizací. První možnost je použití průměrných hodnot, které jsou ale v časovém úseku velmi zkreslené. Dále nemůžeme se průměrování úplně nehodí na neustále se měnící hodnoty závislé na rychlosti změny. V tomto případě použijeme metodu vyřazení nesmyslných hodnot, které se objeví vícekrát za sebou.

5.8.1 Klasifikace a určení chyby

Každé měření v sobě obsahuje nejistotu správnosti výsledku. Pokud je hodnota nestandardně vychýlena od zavedené normální hodnoty, metody na vyhlazování jsou na místě. Tyto hodnoty nejsou úplně náhodné.

5.8.2 Zpracování a výpočet

```
if (frequency > 10000) {  
    if (counter > 2) {  
  
        freq = 0;  
        InputFreq = 0;  
  
    }  
    counter += 1;  
  
} else {  
    counter = 0;  
  
}
```

Obrázek 48 - Vyfiltrování extrémních hodnot
(Zdroj: vlastní tvorba)

Příklad metody zpracování hodnot, které jsou nesmyslné. (obr. 48.) Metoda je použita vícekrát, ilustrovaná část pochází z vyhodnocení otáčení zadních kol za účelem PID regulace. Pokud se hodnoty objeví častěji než 2x za sebou, jsou zahozeny. Dále je určen rozsah smysluplných hodnot z reálného prostředí. Do proměnné freq jsou zapisovány hodnoty, které prošly podmínkami a jsou při příštím průchodu zapisovány do ovládací metody PID.

5.9 Kalibrace jízdy a testování vozidla

Model byl testován přímo na závodní dráze v Třebelovicích pod Orebem. Zde má položená dráha 10 zatáček v celkové délce 12 m. Na dráze leží 10 poloměrem shodných zatáček. Zatáčky jsou pravotočivé a levotočivé. Dále mají shodný průměr, velikost mantinelů a nájezdovou vzdálenost.



Obrázek 49 - Testovaná pravotočivá závodní zatáčka
(Zdroj: vlastní tvorba)

Měření pro potřeby zachycení chyby byly prováděny na vyobrazené pravotočivé zatáčce. (obr. 49.) Podmínky jsou pro všechny měření naprosto shodná. Sledován je průběh řízení modelu uvnitř zatáčky, jistota a kvalita vyhlazování chyb, vzdálenost od mantinelů a odchylka od ideální stopy.

Před každou jízdou jsou z vozovky odstraněny stopy po předchozí jízdě a pokud je dráha nerovná nebo nestandardní, je opravena na další bezchybnou jízdu.

5.9.1 Měření

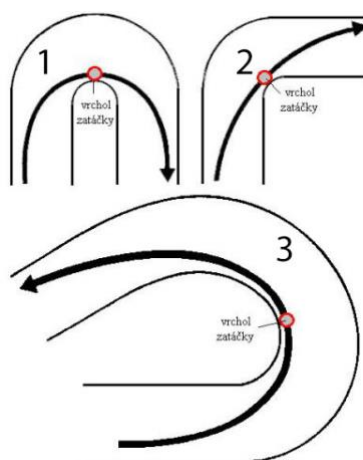
Měření úspěšnosti modelu je zásadní při debuggování a vylepšování částí kódu. Pro tyto účely je zavedeno kritérium. Počet úspěšných přejetí zatáčkou bez nutné pomoci autonomní jízdou. V případě, že model není schopen projet zatáčku, změny se vrací a testování bylo neúspěšné.

Testování průběhu jízdy na rovném úseku není pro naše měření příliš důležité, je nesrovnatelně jednodušší jízda po rovinné trajektorii v porovnání s jízdou do zatáčky.

Ke každému průjezdu je potřeba evidovat poznámky a komentáře pro další ladění a pro lepší průběh příští jízdy.

5.9.2 Shrnutí jízdy

Model dosahuje úspěšnosti 7 projetí z 10 testovaných průjezdů zatáčkami. Velmi záleží, jakou trajektorii a směr při vjezdu do zatáčky nabírá a také jakou nájezdovou rychlost model do zatáčky nabírá. [12]



Obrázek 50 - Postupy správného projetí zatáčkami

(Zdroj: online <https://rcautaliberec.webnode.cz/technicky-koutek/technika-jizdy/>)

Model je schopen provádět úspěšné projetí první a druhé vyobrazené zatáčky zde uvedené na popisujícím obrázku. (obr. 50.) Pokud nejsou zajištěny na okruhu jednoduché zatáčky ve smyslu jedno točivých na pravou nebo levou stranu, není zaručeno hladkého průjezdu a model může vykazovat chaotické chování a nesouvislou jízdu.

6 Shrnutí výsledků

Vytvořit systém, který dokáže reagovat v reálném čase na změny na dráze je komplexní problém, který je potřeba řešit vyšším výpočetním výkonem. V průběhu realizace a samotného testování bylo několikrát naráženo na samotné limity vývojové platformy Arduino. Pokud bude na dráze více než jedno autonomní vozidlo, je zapotřebí aby mezi sebou komunikovala, což v této verzi nebude dosažitelné.

Pokud vezmeme v úvahu rychlost průchodu programové smyčky, která se provádí přibližně 10x za vteřinu, není možné, aby se navýšil počet připojených periférií a z nich vyhodnocovaly další údaje. Procesor pracující na frekvenci 16 MHz již teď nedokáže, ani po optimalizaci softwaru, vyčítat rychleji a přesněji hodnoty, které zpracovává. Údaje z čidel se v omezené míře vypisují na připojený OLED panel. Bohužel bez jakékoliv historie pohybu a s mírnou latencí. Ve finálním řešení se obsazení paměti pohybuje okolo 70 %.

Další zdokonalování programu by bylo možné v případě výměny platformy za výkonnější vývojovou desku, schopnou například i komunikaci v reálném čase s jiným mikroprocesorem pro rozložení zátěže.

7 Závěry a doporučení

V bakalářské práci bylo cílem vytvořit model, který bude schopen jízdy jak za asistence řidiče, tak autonomního pohybu. Z pohledu zadání byly prostudované dostupné zdroje informací a vybrané metody zakomponované do vlastního řešení.

Autonomní robotický model postavený na platformě Arduino dokázal plnit omezené funkce autonomní jízdy na závodním okruhu. V rámci realizace bylo využito algoritmů pro sledování stopy, která bylo nutno upravit v souladu s rychlostí vyhodnocování a mechanickými vlastnostmi použitého podvozku. Dále byla v průběhu hledání vhodného řešení zkoušena kombinace s třemi a pěti čidly snímajících okolí. Ve všech zkouškách si lépe vedl model osazený více čidly, a to i za cenu pomalejších reakcí modelu.

Z praktických testů vyšlo, pokud snížíme latenci mezi snímáním a vyhodnocením signálů, získáme plynulejší a rychlejší jízdu. Použité senzory mají svoje specifické charakteristiky, které bylo třeba v návrhu zohlednit. Ve finální konfiguraci bylo dosaženo nejlepšího poměru osazené elektroniky, kapacity akumulátoru a váhy systému k velikosti modelu, přičemž byl brán ohled i na pořizovací cenu a spolehlivost komponent.

Pokud bychom použili vyhodnocování za pomoci kamer, přesnost by byla jistě vyšší, avšak na model zvoleného měřítka je metoda kvůli nedostatku místa a stabilitě neaplikovatelná. Po kompilaci programem Arduino IDE zabírá kód včetně všech knihoven na vývojové desce okolo 75 % celkového místa. Kompletní programová dokumentace modelu je obsažena na přiloženém CD disku.

Pokud budeme dále uvažovat o modelu většího měřítka, než byl pro tuto práci použitý, nebude problém integrovat plnohodnotný snímací systém založený na několika kamerách a metodu rozpoznávání obrazu. Již vhodnou velikostí by byl podvozek modelu měřítka 1:10, který se svojí velikostí blíží 50 cm délky. Uvažovaný koncept je jistě zajímavý, nabízí další konstrukční a logická řešení a mohl by tvořit základní problematiku navazující práce.

8 Seznam použité literatury

- [1] FELTON, Ryan. *The Man Who Tested The First Driverless Car In 1925 Had A Bizarre Feud With Harry Houdini* [online]. 14. únor 2017 [vid. 2020-03-23]. Dostupné z: <https://jalopnik.com/the-man-who-tested-the-first-driverless-car-in-1925-had-1792312207>
- [2] ONDRÁČEK, Tomáš. *Technologické trendy v silniční dopravě 3. etapa oblast Autonomní vozidla* [online]. 2018 [vid. 2020-03-23]. Dostupné z: <https://www.tpsd-ertrac.cz/file/3-etapa-oblast-autonomni-vozidla/>
- [3] NHTSA MEDIA. *U.S. Transportation Secretary Elaine L. Chao Announces Further Decreases in Roadway Fatalities* [online]. 22. říjen 2019 [vid. 2020-03-23]. Dostupné z: <https://www.nhtsa.gov/press-releases/roadway-fatalities-2018-fars>
- [4] KILIÁN, Karel. *Samorídící auto jezdilo po silnicích už v roce 1993. Nemělo radar ani GPS – Živě.cz* [online]. 2018 [vid. 2020-03-23]. Dostupné z: <https://www.zive.cz/clanky/samoridici-auto-jezdilo-po-silnicich-uz-v-roce-1993-nemelo-radar-ani-gps/sc-3-a-195526/default.aspx>
- [5] *14 (linka metra v Paříži)* [online]. 2019 [vid. 2020-03-23]. Dostupné z: [https://cs.wikipedia.org/w/index.php?title=14_\(linka_metra_v_Pa%C5%99%C3%AD%C5%BEi\)&oldid=17716421](https://cs.wikipedia.org/w/index.php?title=14_(linka_metra_v_Pa%C5%99%C3%AD%C5%BEi)&oldid=17716421)
- [6] SŮRA, Jan. *Strojvůdce jako okrasa? Vlaky v Česku brzy pojedou samy. Deník.cz* [online]. 2019 [vid. 2020-03-23]. Dostupné z: <https://www.denik.cz/ekonomika/strojvudce-jako-okrasa-vlaky-pojedou-samy-20190107.html>
- [7] *Metro bude moci jezdit bez strojvůdců, schválil Senát. Praha plánuje automatický provoz na ‚děčku‘ | Domov. Lidovky.cz* [online]. 31. říjen 2019 [vid. 2020-03-23]. Dostupné z: https://www.lidovky.cz/domov/provoz-metra-bude-mozny-i-bez-strojvudcu-bezobslužna-udajne-bude-linka-d.A191031_182245_In_domov_ele
- [8] NEAL, Ann. *LiDAR vs. RADAR. Sensors Magazine* [online]. [vid. 2019-06-06]. Dostupné z: <https://www.sensorsmag.com/components/lidar-vs-radar>
- [9] JOHNSON, Michael A, Mohammad H MORADI a J CROWE. *PID control: new identification and design methods* [online]. New York: Springer, 2005 [vid. 2020-04-09]. ISBN 9781846281488. Dostupné z: <http://site.ebrary.com/id/10140816>
- [10] AMARIEI, Cornel. *Arduino Servo Motors. Instructables* [online]. [vid. 2020-04-

- 09]. Dostupné z: <https://www.instructables.com/id/Arduino-Servo-Motors/>
- [11] KONFRŠT, Pavel. Všesměrový podvozek robota se servopohony a základní odometrií. 2012, 56.
- [12] SILVA, Jorge. The Best Path a DeepRacer Can Learn. *Medium* [online]. 10. prosinec 2019 [vid. 2020-04-11]. Dostupné z: <https://medium.com/myplanet-musings/the-best-path-a-deepracer-can-learn-2a468a3f6d64>
- [13] KOČIŠ, Štefan a Zdenko FIGURA. *Ultrasonic Measurements and Technologies*. [online]. Boston: Springer US, 1996 [vid. 2020-04-09]. ISBN 978-1-4613-1199-7. Dostupné z: <http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3078546>
- [14] SELECKÝ, Matúš a Martin HERODEK. *Arduino: uživatelská příručka*. 2016. ISBN 978-80-251-4840-2.
- [15] HORVATH, Joan C. *Mastering 3D printing*. Berkeley, California: Apress, 2014. Technology in action. ISBN 978-1-4842-0026-1.
- [16] MCROBERTS, Michael. *Beginning Arduino: 50 Arduino projects with step-by-step instructions and easy-to-follow diagrams ; learn how to work with motors, sensors, displays, and networking ; create simple but practical projects such as a rangefinder, an RFID reader, and an internet weather display ; no programming or electronics experience needed*. New York: Apress/Springer, 2010. Technology in action. ISBN 978-1-4302-3241-4.
- [17] ABDUL JABBAAR, Arbi. Ultrasonic Sensor HC-SR04 with Arduino Tutorial. *Arduino Project Hub* [online]. [vid. 2020-04-09]. Dostupné z: <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>
- [18] WINKLER, Zbyněk. *Odometrie (Robotika.cz > Průvodce)* [online]. 2005 [vid. 2020-03-24]. Dostupné z: <https://robotika.cz/guide/odometry/cs>
- [19] MURRAY, Mike. How Rotary Encoders Work - Electronics Basics. *The Geek Pub* [online]. 3. září 2019 [vid. 2020-04-11]. Dostupné z: <https://www.thegeekpub.com/245407/how-rotary-encoders-work-electronics-basics/>

9 Přílohy

Adresářová struktura na přiloženém CD disku

- Source-code
 - Zdrojové kódy spustitelné v programu Arduino IDE
- Libraries
 - Použité zdrojové knihovny použité v bakalářské práci
- Arduino-program-config
 - Nastavení programu Arduino IDE pro kompilaci

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: **Miroslav Škoda**
Osobní číslo: **I1600615**
Adresa: **Na Důchodě 1562, Hradec Králové, 50002 Hradec Králové 2, Česká republika**
Téma práce: **Koncept vozidla autonomně řízeného mikroprocesorem v reálném čase**
Téma práce anglicky: **Concept of vehicle real-time autonomously controlled by the microprocessor**
Vedoucí práce: **Ing. Pavel Blažek, Ph.D.**
Katedra informačních technologií

Zásady pro vypracování:

Cílem je stvořit dopravní prostředek, bez ohledu na druh, který bude schopen jízdy jak za asistence řidiče, či jiného vstupního zdroje, tak jízdy zcela samostatně tedy autonomní, autonomní jízda bude zahrnovat aktivní vyhýbání překážkám, plně souvislou jízdu, stejně tak bude v reálném čase reagovat na neočekávané situace například překážku před vozidlem nebo náhlou změnu trajektorie dráhy.

- Úvod do problematiky
- Teoretické základy autonomního pohybu
- Návrh a realizace modelu na základě teorie
- Testování modelu
- Shrnutí výsledků
- Závěr a doporučení

Seznam doporučené literatury:

1. Ondráček, Tomáš. 2018. Technologické trendy v silniční dopravě – 1. etapa, Popis problémů současného stavu, oblast autonomní vozidla
2. Ondráček, Tomáš. 2018. Technologické trendy v silniční dopravě – 3. etapa, oblast Autonomní vozidla
3. KONFRŠT, Pavel. 2012. Všesměrový podvozek robota se servopohony a základní odometrií
4. WINKLER, Zbyněk. 2005. Odometrie. In: Robotika.cz [online]. [cit. 2016-05-10]. Dostupné z: <https://robotika.cz/guide/odometry/cs>
5. Selecký, Matúš. 2016. Arduino : uživatelská příručka, Computer Press, 978-80-251-4840-2

Podpis studenta:



Datum: 25. 4. 2020

Podpis vedoucího práce:



Datum:

25. 4. 2020