



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**  
DEPARTMENT OF COMPUTER SYSTEMS

**SAMOŘIDITELNÝ MODEL AUTÍČKA PRO NXP CUP**  
SELF-DRIVING CAR MODEL FOR NXP CUP

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. VOJTECH ŠÍMA**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. JOSEF STRNADEL, Ph.D.**

**BRNO 2025**

## Zadání diplomové práce



Ústav: Ústav počítačových systémů (UPSY)  
Student: **Šíma Vojtěch, Bc.**  
Program: Informační technologie a umělá inteligence  
Specializace: Vestavěné systémy  
Název: **Samořiditelný model autička pro NXP Cup**  
Kategorie: Vestavěné systémy  
Akademický rok: 2024/25

163036

### Zadání:

1. Detailně se seznamte s pravidly soutěže NXP Cup. Přehledově shrňte známé vybavení a jeho parametry, vývojová úskalí, přístupy a řešení využité dosavadními účastníky této soutěže a výsledky dosavadních kol soutěže.
2. Shrňte požadavky kladené na samořiditelný model autička pro NXP Cup. Zvolte vhodnou sadu prvků (podvozek, elektronika, senzory, pohon, napájení atp.) pro konstrukci modelu.
3. Sestavte a oživte prototyp modelu, implementujte programové vybavení pro základní řízení pohybu modelu a sběr dat o pohybu a okolí modelu. Navrhněte rozšíření prototypu s cílem dosažení jeho konkurenceschopnosti v soutěži NXP Cup.
4. Rozšíření prototypu z bodu 3 realizujte.
5. Připravte sadu několika, soutěži NXP Cup blízkých, úloh umožňujících ověřit splnění požadavků kladených na model. Splnění požadavků experimentálně ověřte a zhodnotěte.
6. Zúčastněte se soutěže NXP Cup.
7. Shrňte chyby/nejistoty a vlivy stěžejní pro samořízení modelu, kriticky zhodnotěte vlastnosti použitého přístupu k řízení modelu, identifikujte příčiny a navrhněte řešení případných nedostatků.
8. Diskutujte možné směry pokračování v řešení zadaného tématu a rozveděte ty, které považujete za nejvíce perspektivní.

### Literatura:

- NXP Cup (EMEA's Rolling Robots competition). Dostupné z <https://nxpcup.nxp.com>.
- Welcome to NXP Cup. Dostupné z <https://nxp.gitbook.io/nxp-cup>.
- Jahn, F. Samořízení modelu auta v neznámém prostředí pomocí SLAM. Brno, 2023. Bakalářská práce. FIT VUT v Brně.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

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

Vedoucí práce: **Strnadel Josef, Ing., Ph.D.**  
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.  
Datum zadání: 1.11.2024  
Termín pro odevzdání: 21.5.2025  
Datum schválení: 31.10.2024

## **Abstrakt**

Cílem této diplomové práce je na základě požadavků kladených na model autíčka pro účast v soutěži NXP Cup navrhnout, sestrojit a zprovoznit konkurenceschopný samořiditelný prototyp modelu pro pohyb na dráze. Návrh vychází z pravidel soutěže, informací získaných z předchozích ročníků, osobních zkušeností bývalých účastníků a z požadavků potřebných na celkovou funkčnost modelu. Je popsána teorie týkající se vestavěných systémů, autonomních vozidel a jednotlivých součástek, ze kterých se model skládá. Na tomto základě je pak vytvořena implementace, při které pro hlavní navigaci slouží kamera Pixy 2, doplněna dalšími pomocnými senzory. Během pohybu na dráze dochází ke sběru dat, která slouží k vylepšování algoritmu a jsou také dále zpracována a prezentována. Model se zúčastnil soutěže NXP Cup. Spolu s touto prací byl vytvořen průvodce ke zprovoznění modelu i k implementaci, který může sloužit jako výchozí bod při navazování na tuto závěrečnou práci a model v ní vytvořený v příštích letech.

## **Abstract**

The aim of this master's thesis is to design, build and operate a competitive autonomous prototype of a model car for the NXP Cup competition. The design is based on the rules of the competition, information gathered from previous years, personal experiences of former participants and the requirements needed for the overall functionality of the model. The theory regarding embedded systems, autonomous vehicles and the individual components that make up the model is described. On this basis, an implementation is then developed and the Pixy 2 camera is used for main navigation, supplemented by additional auxiliary sensors. Data is collected while the vehicle is moving along the track, which is used to improve the algorithm and is also further processed and presented. The model participated in the NXP Cup competition. Along with this thesis, a guide to the model and the implementation was created, which can serve as a starting point for building on this thesis and the model created in it in future years.

## **Klíčová slova**

NXP Cup, autonomní vozidlo, vestavěný systém, mikrokontrolér, senzory, Pixy 2, model, dráha, motory, sledování čáry, vektory, detekce, navigační algoritmus

## **Keywords**

NXP Cup, autonomous vehicle, embedded system, microcontroller, sensors, Pixy 2, model, track, motors, line tracking, vectors, detection, navigation algorithm

## **Citace**

ŠÍMA, Vojtěch. *Samořiditelný model autíčka pro NXP Cup*. Brno, 2025. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Josef Strnadel, Ph.D.

# Samořiditelný model autíčka pro NXP Cup

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Josefa Strnadela, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
Vojtěch Šíma  
17. května 2025

## Poděkování

Velmi rád bych poděkoval mému vedoucímu práce panu Ing. Josefу Strnadelovi, Ph.D. za odborné vedení práce, cenné rady při řešení problémů a odpovědi na otázky v průběhu celé práce i na konzultacích.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Výchozí stav projektu</b>	<b>4</b>
2.1	Specifikace požadavků . . . . .	4
2.2	NXP Cup . . . . .	5
2.3	Model a obecné principy pro jeho ovládání . . . . .	7
2.4	Příklady dříve implementovaných řešení . . . . .	9
<b>3</b>	<b>Související téma potřebná k řešení projektu</b>	<b>11</b>
3.1	Vestavěné systémy . . . . .	11
3.2	Autonomní mobilita . . . . .	12
3.3	Základní fyzikální veličiny související s pohybem modelu . . . . .	16
3.4	Mikrokontrolér . . . . .	17
3.5	Prostředky na čipu mikrokontroléru . . . . .	19
3.6	Vybraná komunikační rozhraní a vývojová prostředí . . . . .	21
<b>4</b>	<b>Návrh řešení</b>	<b>24</b>
4.1	Volba základních řídicích prvků . . . . .	24
4.2	Napájení modelu . . . . .	29
4.3	Senzory pro primární navigaci . . . . .	31
4.4	Senzory pro detekci čáry . . . . .	35
4.5	Senzory pro detekci překážky . . . . .	38
4.6	Magnetické senzory . . . . .	42
4.7	Ukládání a přenos dat z modelu . . . . .	43
<b>5</b>	<b>Řešení a implementace</b>	<b>45</b>
5.1	Rozšiřující deska a platforma pro uchycení senzorů . . . . .	48
5.2	Ovládání pohybu modelu . . . . .	49
5.3	Zaznamenávání uražené vzdálenosti . . . . .	50
5.4	Infračervené senzory . . . . .	51
5.5	Barevné a ultrazvukové senzory . . . . .	51
5.6	Kamera Pixy 2 a navigační algoritmus . . . . .	53
5.7	Sběr, vyhodnocení a zpracování dat . . . . .	57
<b>6</b>	<b>Zhodnocení a náměty na rozšíření</b>	<b>58</b>
6.1	Zhodnocení řešení a dosažených výsledků . . . . .	58
6.2	Možnosti rozšíření řešení . . . . .	67

<b>7 Závěr</b>	<b>69</b>
<b>Literatura</b>	<b>70</b>
<b>A Schéma desky plošného spoje</b>	<b>73</b>
<b>B Vývojový diagram</b>	<b>75</b>
<b>C Struktura přiloženého datového obsahu</b>	<b>77</b>

# Kapitola 1

## Úvod

Tématem této diplomové práce je navrhnutí, sestavení a následné naprogramování konkurenčeschopného autonomního modelu autíčka, jehož cílem pak bude co nejlépe zvládnout projekt soutěžní dráhu a obstát v soutěži NXP Cup.

Cílem této práce je seznámit se s aktuálními možnostmi vybavení pro model autíčka, které budou odpovídat regulím soutěže, shrnout a zhodnotit dosud použité přístupy k řešení této problematiky v minulých ročních soutěžích a na základě těchto informací pak vhodně zvolit základní elektronické komponenty, které se budou starat o napájení a základní pohyb modelu. Následně dojde k rozšíření funkčnosti pomocí kamery a rozšiřujících senzorů, přičemž bude vytvořen algoritmus, který se bude starat o autonomní pohyb modelu po dráze, jež je v této soutěži specificky dána.

Výsledné řešení bude na základě analýzy dosažených výsledků i dat zhodnoceno a bude navrženo jeho vylepšení či případné rozšíření, které poskytne možnosti, díky kterým bude na tuto práci možné navázat v budoucích ročních soutěžích. Spolu s tím bude vytvořen pomocný průvodní dokument právě pro budoucí účastníky soutěže, který jim pomůže se zprovozněním modelu a pochopením principu jeho ovládání. Bude také popsán implementovaný algoritmus, který díky tomu bude možné dále posouvat a zdokonalovat. Součástí tohoto dokumentu je i sada úloh, jejichž splnění by mělo ověřit požadavky kladené na model pro účast na soutěži.

Hlavní motivací pro výběr tohoto tématu byla především charakteristika zadání, jenž není předem definováno a omezeno na pouhé programování bez hmatatelných výsledků. Práce je založena na tom, že dochází k návrhu a následné práci s elektronikou, která pak pomocí mikrokontroléru dostane potřebnou funkcionalitu. Dalším zajímavým faktorem práce je, že zde je přímé napojení na účast v soutěži NXP Cup, které může přinést cenné a zajímavé zkušenosti v oblasti autonomních modelů a závodění s nimi.

# Kapitola 2

## Výchozí stav projektu

V této kapitole budou shrnuté informace, souvislosti a požadavky, které byly známé na počátku závěrečné práce a z nichž bylo vycházeno v následujících kapitolách. Budou popsány základní pojmy a témata, která jsou stežejní pro sestrojení a vytvoření samořiditelného modelu autíčka pro NXP Cup. Také dojde ke shrnutí požadavků na výsledky a výstupy, které mají během této práce vzniknout.

### 2.1 Specifikace požadavků

V rámci této sekce budou rozsáhleji popsány požadavky na to, co má být výsledkem a finálním výstupem této práce. Tyto požadavky vycházejí z formálních cílů uvedených jako body zadání diplomové práce.

Mezi základní cíle, které je nutné zvládnout pro možnost absolvovat cíle následující, patří seznámení se s pravidly soutěže a vyhodnocení požadavků na model jako takový, které bude nezbytné splnit, aby bylo možné se zúčastnit soutěže. Dále pak provedení rešerše přístupů zvolených od ostatních týmů v minulých ročnících, a také k rešerší teoretických oblastí týkajících se autonomního řízení, vestavěných systémů a principu součástek samotných. Dojde také k popsání známých přístupů, které jsou nutné k pochopení fungování práce jako celku. Samotná účast na soutěži patří mezi jeden ze stežejních cílů této diplomové práce.

Na základě těchto vytvořených požadavků dojde k volbě vhodných součástek, které povedou k sestavení a oživení základního prototypu modelu. Ten pak bude dále rozšiřován především z hlediska funkcionality tak, aby bylo možné se na konci zúčastnit soutěže.

Výsledky pak budou vyhodnoceny, dojde ke shrnutí chyb a nejistot výsledného řešení a na základě vyhodnocení příčin těchto nejistot bude navrženo řešení těchto nedostatků. Nasbíraná data budou vyhodnocena a budou vytvořeny výstupy mapující to, jak se model pohyboval na dráze. Dojde také k navržení možností, jak toto řešení rozšířit a jaké jsou další možné směry, ve kterých je možné na řešení navázat či v jakém směru bude možné ho rozšířit například v rámci jiné práce.

Mezi požadavky je také zahrnuto postupné testování práce, ke kterému bude postupně docházet. V průběhu toho dojde k navržení sady několika úloh, které jsou blízké soutěžnímu prostředí, a jejich splnění bude moci sloužit případným dalším účastníkům pro kontrolu, zda model splňuje základní požadavky na to, aby se mohl zúčastnit soutěže.

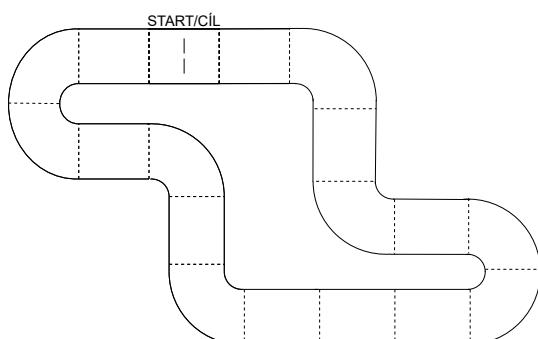
## 2.2 NXP Cup

NXP Cup<sup>1</sup> je soutěž především vysokoškolských studentských týmů z oblasti EMEA<sup>2</sup>. Každý tým má za úkol navrhnout, sestavit a následně naprogramovat model autonomního robota (vozidla). Soutěž probíhá pod záštitou firmy NXP Semiconductors, jejíž součástky, především samotné mikrokontroléry, obstarávající ovládání vozidel, jsou v modelech použity. Cílem je pomocí vhodně zvolených součástek dosáhnout takového stavu, aby se model dokázal pohybovat co nejrychleji, ale zároveň i velmi přesně v rámci dráhy, aby dokázal uspět v soutěži jako celek.

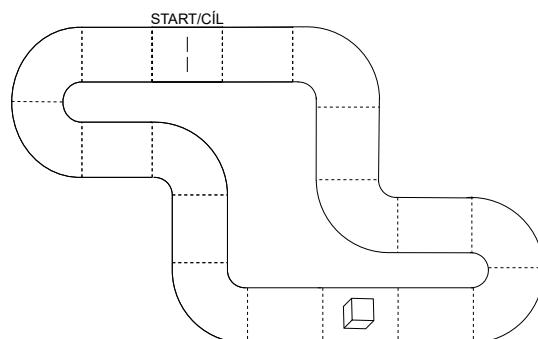
V soutěži musí každý tým absolvovat dvě disciplíny. První disciplínou je měřený závod na jedno kolo, kdy je cílem dosáhnout co nejlepšího času na předem neznámé trati. Sestrojené vozidlo musí vždy alespoň dvěma koly zůstat uvnitř vyznačené dráhy.

Ta je na bílém matném podkladu, šířka je 55 cm a vytvořena je pak černými krajními čárami, kdy každá má šířku 2 cm. Skládá se z rovných úseků a zatáček různých úhlů, jejichž pořadí je pro dané soutěžní kolo vygenerováno náhodně. Vždy se však jedná o trať přibližně o velikosti 7 x 4 metry (ve finálových kolech, ze kterých existují videozáznamy, se jedná o trať s většími rozdíly). Na trati je vždy umístěno minimálně jedno křížení (trať je v tomto místě projeta dvakrát, křížení se však vyskytuje na rovných úsecích a úhlech, ve kterém k němu dochází, je 45° nebo 90°). V místě křížení není trať plně ohraničena, což klade vyšší nároky na navigační algoritmus a jedná se tedy o prvek, který zvyšuje obtížnost. Jednoduchý návrh trati bez křížení je možné vidět na obrázku 2.1.

V závislosti na tom, jakého času je dosaženo vzhledem k ostatním týmům, je ustanoven celkové pořadí a dle něj jsou následně rozdeleny body (nejrychlejší tým by měl obdržet 650 bodů, druhý 550, třetí 450 a každý dále umístěný potom vždy o 50 bodů méně). Druhá disciplína se nazývá *Mouser lap of honor* a následuje ihned po tom, co vozidlo projede cílem v předchozím, časově měřeném kole. Na náhodné místo ve vyznačené trati je umístěna překážka z materiálu podobného polystyrenu bílé barvy o rozměrech 20 x 20 x 20 cm (názorně provedeno na nákresu trati na obrázku 2.2). Překážka je umístěna doprostřed rovného úseku dráhy (myšleno mezi čarami, které vytvářejí dráhu samotnou) a úkolem soutěžního týmu je zajistit, aby vozidlo zastavilo v dostatečné vzdálenosti od této překážky a nedotklo se jí. Pokud se povede tento úkol splnit, dostane daný tým v tomto kole dalších 150 bodů.



Obrázek 2.1: Obrázek ukazující možné rozložení trati pro závod.



Obrázek 2.2: Po dokončení kola je na náhodné místo (na rovince) umístěna překážka.

<sup>1</sup>NXP Cup – <https://nxpcup.nxp.com/>

<sup>2</sup>EMEA – Europe, Middle East, and Africa (souhrnné označení regionu Evropy, Blízkého východu a Afriky)

Soutěž probíhá ve více kolech a je možné se z lokálních kol, která se konají na jednotlivých univerzitách po celé Evropě, nominovat na celkové mezinárodní finále. To se v roce 2025 bude konat v Eindhovenu, kde jsou pro nejlepší týmy připraveny hodnotné ceny i finanční ohodnocení. Hlavním cílem soutěže je však prohloubit znalosti v technických obořech, osvojit si znalosti a získat důležité praktické zkušenosti v oblasti vývoje robotiky a autonomních vozidel.

### 2.2.1 Pravidla soutěže NXP Cup pro ročník 2025

Pro ročník 2025, kterého se bude sestrojený model autonomního autíčka účastnit, byla firmou NXP vydána pravidla [17], jejichž zkrácený výčet toho nejdůležitějšího je vypsán níže. Představení soutěže, průvodní informace a rady k součástkám či implementaci organizátor poskytuje zde<sup>3</sup>.

#### Požadavky pro složení týmu:

- Každý tým se musí skládat z jednoho až tří členů týmu.
- Každý člen musí být studentem registrované školy, nezávisle na ročníku či oboru studia.
- Členové nemusí být studenty stejné školy, musí však všechni splňovat pravidla.
- Všechny přihlášené týmy se účastní stejně soutěže.
- Na vývoji, sestavování či programování se smí podílet jen a pouze členové týmu.
- Každý tým může vytvořit více autonomních vozidel, každé však musí mít svoji vlastní registraci, jiné jméno týmu a musí být osazeno a založeno na jiném mikrokontroléru, který se stará o chod modelu.

#### Každý tým musí použít jeden z následujících modelů:

- Model vytisknutý na 3D tiskárně, který poskytl a podrobně popisuje organizátor soutěže NXP.
- Jakýkoliv model, který byl doporučen v předchozích ročnících soutěže.
- Jakákoliv komerční stavebnice, kterou je možné složit do podoby modelu, který bude velikostí odpovídat trati.
- Vlastní návrh modelu, nezávisle na materiálu či velikosti.

#### Pravidla která musí všechny výše zmíněné modely splňovat:

- Model může být poháněn až čtyřmi motory (kartáčovými i bezkartáčovými). Ty mohou být umístěny pro každé kolo zvlášť i samostatně pro každou nápravu.
- Model může obsahovat maximálně 4 kola.
- K řízení musí být použito mikrokontroléru či vývojové desky od firmy NXP. Mohou být vytvářeny i vlastní desky, musí však být osazeny mikroprocesorem od NXP. Může být použito více desek než jedna.

---

<sup>3</sup>Informace o soutěži od NXP – <https://nxp.gitbook.io/nxp-cup>

- Pro primární navigaci musí být použita kamera, ostatní senzory mohou být použity pro zlepšení povědomí o okolí vozidla.
- Model musí být autonomní a nemůže být jakkoliv řízen na dálku, v soutěžním pokusu nesmí být vybaven jakýmkoli bezdrátovým zařízením.
- Pokud firmy NXP produkuje určitý typ senzorů, měly by tyto senzory být použity. V opačném případě je možné použít jakékoli jiné.
- Pro pohánění všech součástí (samotného modelu i elektroniky) může být použita pouze jedna baterie. V případě použití LiPo baterie se musí mít maximálně 3 články o maximálním napětí 12 V a kapacitě 6000 mAh.

## 2.3 Model a obecné principy pro jeho ovládání

V této sekci bude popsán model, který byl zvolen jako stěžejní pro tuto práci. Na něm bude postaveno celé řešení v podobě elektroniky a na ní navázané implementace. Dále dojde k představení jeho dvou základních ovládacích prvků – stejnosměrných servomotorů a stejnosměrných bezkartáčových motorů.

### 2.3.1 Zvolený model pro tuto práci

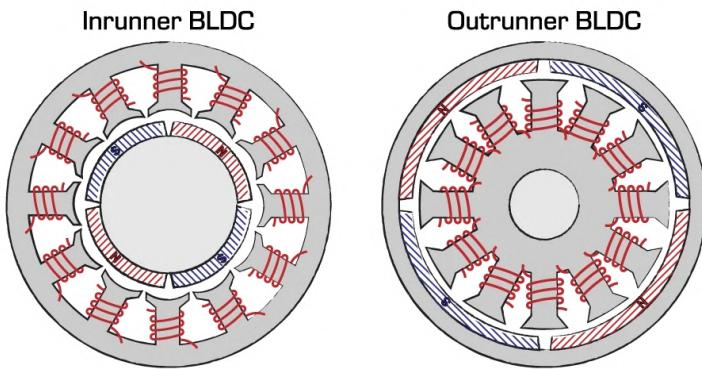
Jedná se o model vozu ROB0170, který byl dříve veden jako jeden ze vzorových pro použití v soutěži NXP Cup. Má délku 34 cm a šířku 17 cm. Skládá se z kovového podvozku, na kterém jsou přiděleny jednotlivé nápravy. Na zadní nápravě se nachází pro každé kolo jeden bezkartáčový motor, který se v kombinaci s ozubenými kolečky, které zajišťují přenos otáčivého pohybu až ke kolu, stará o pohyb modelu vpřed. U přední nápravy se nachází servomotor, který se stará o natáčení předních kol a tím tedy i zatáčení modelu. Model byl už sestaven, protože na fakultě byl dříve využit k jiné závěrečné práci. V průběhu samotné práce byl však model nejednou rozebrán a znova opět sestaven. Osazeny jsou pneumatiky o průměru 7 cm. Veškerá další elektronika, zdroj napětí či další senzory musí být do modelu vhodně umístěny a připojeny.

### 2.3.2 Bezkartáčové motory

K pohonu modelu slouží dva bezkartáčové stejnosměrné motory, oba umístěny na zadní nápravě, která je tak tou jedinou hnanou, starající se o pohyb modelu vpřed. Každé kolo má svůj vlastní motor, jehož rotace je pomocí dvou ozubených koleček převedena z hřídele motorku na hřídel samotného kola. Teoreticky tedy může být na každém kole jiná úroveň otáček, nicméně v práci vždy byla snaha dosáhnout na obou kolech stejných otáček.

Bezkartáčové stejnosměrné motory, často také označované pouze zkratkou BLDC z *anglického Brushless DC motor* jsou jedním ze základních typů motorů. Na rozdíl od kartáčových motorů, které byly jeho přímým předchůdcem, tento typ eliminuje tření vznikající na kartáčích a k pohybu rotoru dochází s minimálním třením se statorem. Díky tomuto (a samotné absenci kartáčů a komutátoru) jsou tyto motory efektivnější, mají delší životnost díky menšímu opotřebení způsobenému absencí tření a jejich chod je tišší. Motor samotný je řízen elektronicky například z mikrokontroléru, díky čemuž můžou být jeho otáčky a tudíž výkon regulovány dle potřeby uživatele. K tomu však je nutné použít externí regulátor otáček motoru.

Princip motoru je založen na magnetismu. Ve statoru, což je pevná část motoru, jsou umístěny elektromagnety a v rotoru, což je rotující část motoru, jsou umístěny magnety permanentní. Je rozlišováno mezi motory outrunnerovými a inrunnerovými v závislosti na tom, zda jsou tyto permanentní magnety umístěny na vnitřní straně rotoru (viz obrázek 2.3). V případě motorů **A2212/15T**, jenž jsou osazeny a detailněji popsány v části 4.1.3, se bavíme o typu outrunnerovém s magnety na vnější straně [12].



Obrázek 2.3: Ukázka rozdílného umístění permanentních magnetů mezi outrunnerovými a inrunnerovými motory. Převzato z [12].

K samotnému pohybu motoru dochází tak, že jsou pomocí elektronického regulátoru otáček měněna magnetická pole uvnitř statoru, díky čemuž jsou magnety v rotoru střídavě odpuzovány a přitahovány, což vede k tomu, že je vytvářen rotační pohyb.

Výhodou motorů je jejich především efektivita (vyšší účinnost, výkon, životnost a tiští chod) a možnost vysoké přesnosti při jeho využívání. Nevýhodou pak může být vyšší pořizovací cena či nutnost řídit motor právě pomocí elektronického regulátoru. Tyto motory jsou využity v širokém množství výrobků, ať už v elektromobilech, domácích spotřebičích, elektrickém náradí či RC modelech, což je i případ této práce. Model sice není řízen pomocí radiových vln, avšak princip samotného pohybu je díky použití těchto součástek stejný [16].

### 2.3.3 Servomotory

Zatáčení modelu je obstaráváno pomocí servomotoru, na který jsou připojená ramena, jež vedou k oběma předním kolům, tudíž natočení hřídele servomotoru vede k natočení kol a změně směru jízdy.

Servomotory (používáno také pouze servo) jsou malé, ale výkonné a energeticky nenáročné motory, které jsou využívány tam, kde je potřeba nastavit něco do přesné polohy. Hlavním rozdílem oproti běžným motorům popsaným v předchozí části je, že nedochází k tomu, že se periodicky otácejí kolem své osy a pouze se mění rychlosť otáčení, ale natáčí se pouze v určitém rozsahu mezi různými pozicemi.

Skládají se ze série ozubených kol (plastových či kovových), malého stejnosměrného motoru, potenciometru a řídícího obvodu. Všechny součástky jsou umístěny v jednom pouzdře a z toho jsou vývody v podobě vodičů (řídící signál a napájení) a hřídele, ke které je připojeno to, co je třeba polohovat. V případě modelu autíčka se jedná o ramena směřující k předním kolům modelu.

Když je stejnosměrný motor v pohybu, potenciometr dává zpětnou vazbu elektronice. Ta vyhodnocuje, zda je již servo v pozici, jenž byla nastavena pomocí řídícího signálu. Jakmile je této pozice dosaženo, dojde k ukončení napájení a motor se zastaví [24].

Řídící signál pracuje na předem dané frekvenci, v rámci které jsou stanoveny minimální a maximální hodnota, které stanoví hranice rozsahu daného serva. Detailněji je pak tento princip popsán přímo při představení použitého serva v návrhu řešení v části 4.1.2.

## 2.4 Příklady dříve implementovaných řešení

Jelikož soutěž NXP Cup (dříve se jednalo o Freescale Cup, poté však tato firma byla odkoupena firmou NXP Semiconductors a na základě toho byl změněn i název soutěže) má za sebou již více než 10 ročníků, během kterých se postupně vyvíjely modely i technologie, a docházelo také k postupným změnám v principu a formátu soutěže samotné.

Předchůdcem této soutěže byla Freescale Race Challenge, při které se k závodění používala autodráha Carrera, na níž závodily malé modely autíček, ve kterých byla osazena deska plošného spoje (všechny soutěžní týmy dostaly jednu a tu samou od organizátora soutěže). Na základě dat z akcelerometru, který detekoval zrychlení v jednotlivých osách, bylo možné detektovat zatačky na dráze a reagovat na ně snížením rychlosti tak, aby model nevypadl z dané dráhy [3]. Oproti dnešnímu formátu soutěže, kde je třeba řešit důležité faktory jako okraje dráhy a to, aby model dráhu neopustil, bylo dříve třeba naprogramovat pouze takový algoritmus, který reguluje napětí jdoucí z dráhy k motorům v autíčku podle toho, jaká data získává právě z akcelerometru.

V prvních ročnících soutěže v novém formátu byla oproti současným ročníkům dráha v jiné podobě – místo ohrazení 2 čarami na vnějšku se zde vyskytovala pouze jedna černá vodící čára, která byla umístěna uprostřed, jak je možné vidět na obrázku níže 2.4. Díky tomu se v implementovaném algoritmu jednotlivé týmy pravděpodobně mnohem více soustředily na to, aby sledovaly přímo tuto vodící čáru, po které se snažily pohybovat přímo (ať už pomocí kamery, nebo různých senzorů určených právě na sledování čáry). Informace o tom, zda v této době bylo také pravidlo o primárním použití kamery, se nepodařilo dohledat. Je pravděpodobné, že zde bylo menší riziko opuštění dráhy, protože při chybě, jenž způsobila opuštění vodící čáry, bylo více prostoru na to zareagovat a vrátit se zpět na dráhu.



Obrázek 2.4: Ukázka jak vypadala trať v dřívějších ročnících. Převzato z [1].



Obrázek 2.5: Ukázka jak vypadala trať v ročníku 2025 na soutěžním dni.

V dřívějších ročnících pak byla využívána pouze lineární kamera – ta snímá pouze jeden řádek pixelů a vyhodnocuje, jaká se v jednotlivém pixelu vyskytuje barva, na základě čehož se může potom algoritmus rozhodovat. Výhodou použití těchto kamer je především jejich vyhodnocovací rychlosť. Pro letošní ročník je však již doporučena jako hlavní navigační prvek kamera Pixy 2, která sleduje již kompletní 2D prostor před vozidlem a předzpracovává data tak, že do mikrokontroléru již v případě zvoleného módu sledování čar pouze posílá souřadnice začátků a konců jednotlivých čar, které kamera detekovala. Na základě těchto souřadnic pak musí být implementován algoritmus, který bude rychle a přesně vyhodnocovat následující kroky, jež budou muset být provedeny, aby se model udržel mezi čarami na závodní dráze. Podrobněji je tato kamera a její použití v práci zmíněno později, a to v části [4.3.1](#).

V průběhu zimního semestru 2024/2025 proběhla na fakultě prezentace soutěže NXP Cup a byl zde i zástupce týmu, který se v minulém ročníku soutěže umístil na druhém místě v evropském finále. Byl i názorně předveden model, se kterým se účastnili. Během diskuze bylo získáno plno zajímavých informací jak o modelu, tak i o tom, jak soutěž probíhá. Hlavním faktorem jejich úspěchu je to, že tým soutěží s několik let starým modelem, který pouze vylepšují v detailech. Mezi ně patří například, že pneumatiky pro soutěž si tým vyrábí sám ze speciálního silikonu z důvodu, aby měl co největší přilnavost a tudíž výhodu oproti ostatním týmům. Za zajímavost považuji fakt, že i když v posledních ročnících soutěže bývá doporučována kamera Pixy 2, tento úspěšný tým má svůj algoritmus stále založen na kameře lineární.

## Kapitola 3

# Související téma potřebná k řešení projektu

V této kapitole budou shrnuta základní téma a fakta související s budoucím řešením práce, jako jsou například autonomní vozidla a jiné dopravní prostředky, vestavěné systémy či fyzikální veličiny, které bude potřeba během návrhu a vývoje samořiditelného modelu autička zohlednit a pracovat s nimi.

### 3.1 Vestavěné systémy

Vestavěný systém je ve své podstatě kombinace hardwaru a softwaru, která je vytvořena, aby vykonávala konkrétní funkce. Mohou být programovatelné, většinou však jsou tvořeny a pak hromadně používány pro jednu pevně danou funkci (slouží k ovládání jednoho konkrétního zařízení), což je hlavní rozdíl oproti osobním počítačům, které jsou používány k mnohem širšímu spektru obecnějších úloh a úkolů. Vestavěné systémy jsou úzce zaměřeny na jednu konkrétní problematiku či použití. Informace v této kapitole byly čerpány z [9].

Vestavěné systémy jsou založeny na mikroprocesoru či na mikrokontroléru, který se stará o funkčnost celého vestavěného systému. Dále se zde vyskytují paměti, A/D převodníky zajišťující převod analogového signálu do digitální podoby a vstupní či výstupní zařízení, která jsou zde k tomu, aby mohly být získávány podstatné informace z okolí, nebo aby naopak mohly být prezentovány uživatelům systému. Jedná se například o tlačítka, různé druhy senzorů, diody či zobrazovací zařízení různých typů i velikostí.

První vestavěné systémy byly vytvářeny v 60. letech minulého století, kdy byla snaha v programu Apollo vyvinout takový integrovaný obvod, který zredukuje velikost i hmotnost digitálního systému, který se nacházel ve velitelském i lunárním modulu na misích, jejichž úkolem bylo přistání na Měsíci. Tento vestavěný systém pomáhal astronautům shromažďovat a následně využívat důležité informace o letu v reálném čase.

Protože se jednalo o finančně velmi nákladnou záležitost, probíhal i v následujících letech vývoj těchto systémů především ve vesmírném programu či pro armádní účely. Až později, v 70. letech mohly díky snížení cen integrovaných obvodů začít vznikat vestavěné systémy, které byly určeny pro širší a obecnější využití. Původní systémy, které dříve byly analogové či později číslicové, mohly tedy postupně být nahrazovány systémy softwarovými. Jejich výhodou je především to, že je možné návrhy postupně měnit, vylepšit či rozšiřovat dle potřeby.

### **3.1.1 Typy vestavěných systémů**

Vestavěné systémy lze podle funkčních a technických požadavků dělit na tyto základní typy:

#### **Mobilní vestavěné systémy**

Jedná se o menší, snadno přenositelné zařízení a jsou uzpůsobena k tomu, aby fungovala po určitou dobu bez zdroje napájení. Typicky jde o fotoaparáty či mobilní telefony, zde se však už jedná o poměrně složitá zařízení, jejichž zařazení do vestavěných systémů je rozporuplné.

#### **Síťové vestavěné systémy**

Jedná se o zařízení, jenž jsou připojena k internetové síti, pomocí které spolu mohou vzájemně komunikovat a poskytovat si data. Jedná se například o bezpečnostní kamerové systémy, bankovní a pokladní systémy, se kterými se setkáváme na pokladnách v obchodech, v kioscích při nákupu jízdenek nebo při používání bankomatů a platebních terminálů. Dalším takovým zařízením může být například tiskárna, termostat atd.

#### **Nezávislé vestavěné systémy**

Jedná se o systémy, které plní specifické úkoly, ale nijak nekomunikují s okolním světem a jsou na nich nezávislé. Může se jednat o zařízení jako kalkulačka, meteostanice či MP3 přehrávač, ale také o systémy, jenž se vyskytují v domácích spotřebičích, jako jsou například myčka nádobí či automatická pračka. V těchto případech si uživatel vůbec nemusí uvědomovat, že pracuje s nějakým vestavěným systémem.

#### **Vestavěné systémy pracující v reálném čase**

Do této kategorie spadají takové systémy, u kterých je klíčové především to, aby jimi získaná data byla vyhodnocena v co nejkratším (minimálním) časovém intervalu. Důležité je zde také spolehlivost těchto výstupů, protože se jedná o kritické úlohy, jaké jsou například systémy využívané v letectví (prvky autopilota), zdravotnictví (kontrola srdeční aktivity) či ve vojenském průmyslu (navádění či odklonění raket).

Tato práce je založena na tom, že mikrokontrolér (více popsán v sekci 3.4) reaguje na data, která jsou zaslána ze senzorů (popsány postupně v kapitole 4). Dal by se tedy klasifikovat jako vestavěný systém pracující v reálném čase díky nutnosti rychlé a spolehlivé reakce, ačkoliv tyto reakce a selhání v návaznosti na ně nejsou tak kritické jako u aplikací zmíněných v předchozím odstavci.

## **3.2 Autonomní mobilita**

Pojem autonomní (samořiditelná) mobilita označuje takové dopravní prostředky, které jsou schopné pohybovat se samostatně bez nutnosti toho, aby do jejich chodu zasahoval řidič, případně aby jeho vliv byl co nejmenší, v závislosti na tom, o jakou úroveň autonomie jde. Tyto prostředky využívají pro orientaci v prostoru technologie jako umělou inteligenci, senzory, kamery či GPS. Získaná data z těchto technologií zpracovává algoritmus, který se stará o všechna rozhodnutí, jež daný dopravní prostředek provede a povedou k pohybu v prostoru.

Tento pojem zahrnuje nejen autonomní auta, na která je vzhledem k plánovanému každodennímu osobnímu použití kladen největší důraz a budou popsána detailněji kvůli podobnosti s touto prací. Dále také například letadla či vlaky, u nichž je rozšíření těchto technologií v provozu již dále. Nemohou být také opomenuty například drony, ty zaznamenaly v posledních letech velký rozmach, především však bohužel díky faktu, že jsou ve velkém nasazovány ve válečných konfliktech jako levnější a mnohdy i efektivnější varianty k vedení útoku. Ve všeobecném měřítku však autonomní doprava slibuje zvýšení bezpečnosti, efektivity a snížení dopravních komplikací, protože všechny tyto aspekty mohou být vylepšeny právě tím, že bude eliminován lidský faktor a bude nahrazen algoritmem, který by v ideálním případě měl být téměř neomylný.

### 3.2.1 Autonomní řízení v automobilové dopravě

V této části bude detailněji popsáno autonomní řízení v automobilové dopravě. Jeho základní principy a techniky jsou dobrým výchozím bodem pro vytvoření samořiditelného modelu autíčka. To se stejně jako auta v silniční dopravě musí bezpečně a spolehlivě pohybovat ve specifickém prostředí.

Cílem zavádění jakékoli autonomní dopravy je zvýšení efektivity a bezpečnosti provozu. V autonomní dopravě se v zásadě rozlišují různé úrovně autonomie v závislosti na tom, jak moc do řízení zasahuje samotný řidič. Se zvyšující úrovni autonomie klesá role řidiče až k úrovni nejvyšší, kde se již počítá s plnou autonomií a z řidiče již stává pouze pasažér. Jednotlivé úrovně jsou popsány níže a při jejich vysvětlení bylo čerpáno z [26].

#### Úroveň 0 – bez automatizace

Řidič plně ovládá vozidlo. Systém ho může upozornit na hrozící situaci v podobě překážky na silnici, oznamuje překročení maximální rychlosti, vyjetí z jízdního pruhu nebo například hlídání mrtvé úhly. Do této kategorie také spadá parkovací asistent, který pouze řidiče upozorňuje na to, pokud se vozidlo příliš přiblíží jinému vozidlu či překážce. V těchto případech systém nijak neovlivňuje pohyb vozidla, ten plně ovládá řidič.

#### Úroveň 1 – asistence řidiče

Řidič stále musí plně ovládat vozidlo a musí být kdykoliv schopen zareagovat, systém však již může vstupovat do řízení, některé úkony provádět a ulehčit tím řidiči práci, nicméně ten stále je zcela zodpovědný za pohyb vozidla. Může se jednat například o adaptivní tempomat upravující rychlosť vozu podle rychlosti vozidla před ním, nebo jednoduchý asistent jízdy v pruhu. Dalším příkladem může být nouzové brzdění, přičemž v těchto případech již systém vstoupí do řízení, aby zabránil nebezpečí v podobě hrozící srážky.

#### Úroveň 2 – částečná automatizace

Řidič stále dohlíží na prováděné činnosti, ale některé z nich již dokáže systém sám zvládnout bez zásahu řidiče. Ten však stále musí být připraven zasáhnout, kdyby hrozilo nebezpečí. Jedná se například o popojíždění v koloně při malých rychlostech nebo o automatické parkování, kdy je pouze zadáno, kam se má zaparkovat, a systém již vozidlo pomocí kamer a senzorů samostatně zaparkuje.

### **Úroveň 3 – podmíněná automatizace**

Systém už se plnohodnotně stará o jízdu a řidič již do ní nijak nemusí zasahovat, ale musí být bdělý pro případ, že bude třeba řízení převzít. Jedná se však o specifické využití, především na dálnicích, na kterých již dokáže vozidlo předjízdět, řešit kolizní situace nebo brzdit a vyhýbat se v případě hrozící srážky.

### **Úroveň 4 – vysoká automatizace**

Tato úroveň vychází z předchozí úrovně, ale zde již dokáže auto pohybovat úplně všude a dokáže samostatně dojet z bodu A do bodu B. Stále se zde vyskytují pedály i volant. Řidič tak kdykoli může převzít řízení, například při špatném počasí, kdy ho systém k tomu může vyzvat. Pokud k převzetí nedojde, z bezpečnostního důvodu se auto automaticky zastaví.

### **Úroveň 5 – plná automatizace**

Úroveň, kdy se již o vše stará autonomní systém a řidič již zde přechází do 100 % role pasažéra. Na této úrovni se již nepočítá ani s teoretickým zásahem, protože se zde nevyskytují ani pedály s volantem.

Aktuálně v České republice vývojově na úrovni 3 autonomního řízení, přičemž předchozí úrovně jsou již běžně využívány. Zavádění vyšších úrovní naráží na legislativní omezení, jejichž změny se v roce 2025 řeší. V některých zemích, například v Německu, jsou tyto možnosti již dostupné. Cílem je zvýšení bezpečnosti a plynulosti dopravy [27].

Ve světě se testují i prototypy úrovně 4, často v podobě autonomních vozítek pro úklid či přepravu, například na letištích. Nejrozšířenější provoz vozidel této úrovně mobility je pak v USA, kde společnost Waymo nabízí robotaxi služby v oblasti San Francisca, přístupné veřejnosti od roku 2024.



Obrázek 3.1: Ukázka autonomního robotaxi od společnosti Waymo jezdící po různým městech USA. Převzato z [15].

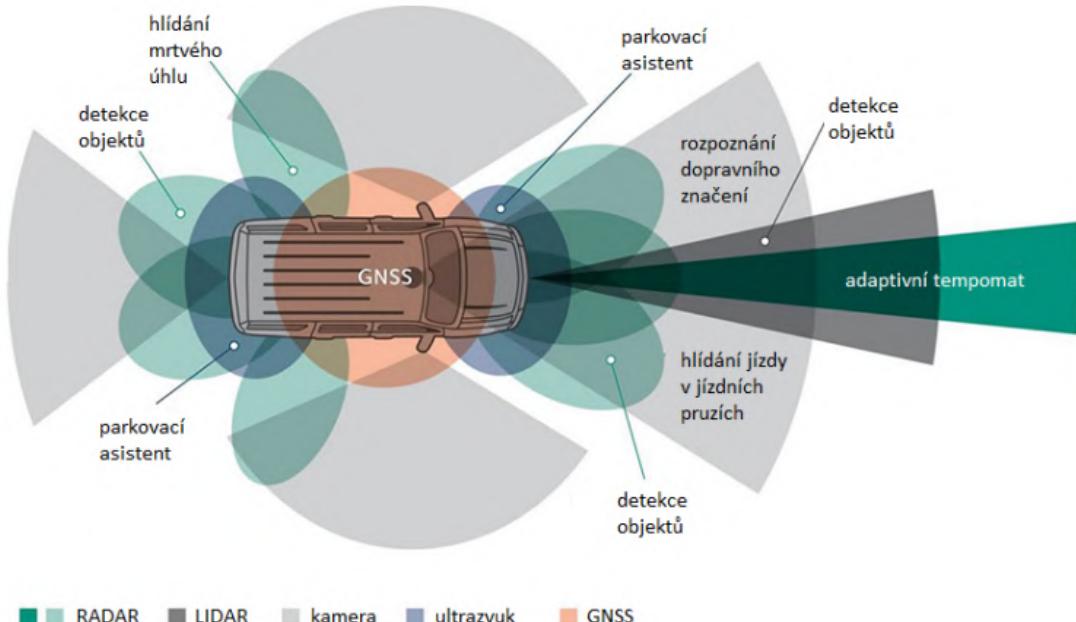
#### **3.2.2 Technologické principy autonomního řízení**

K tomu, aby se vozidlo mohlo bezpečně pohybovat po komunikaci či kdekoli jinde, potřebuje mít dostatek informací o svém okolí, na jejichž základě se pak bude moci rozhodnout o tom, jaký další manévr bude provádět.

Mezi hlavní senzory, které se používají, patří **kamery** sledující ve všech směrech okolí vozidla. Ty můžou zachytit i taková místa, která by pro řidiče v normálním vozidle byla v mrtvém úhlu nebo se mohou starat o sledování vodorovného i svislého dopravního značení. Dalším senzorem je **radar**, založený na vysílání a následném přijímání mikrovlnných vln, které se odrazily od překážek před vozidlem. Na stejném principu pracují dodnes radary, které monitorují oblohy, například na řídících věžích na letištích. Na podobném principu funguje také velmi používaný **lidar**, který však používá k monitorování překážek laserové paprsky, díky čemuž je možné získávat mnohem přesnější informace o objektu, včetně detailů velikostně v jednotkách milimetrů.

Dále potřebuje mít vozidlo přesné informace o své poloze. K tomu potřebuje kvalitní **mapové podklady**, které budou vždy aktuální. Aby s nimi mohlo pracovat, potřebuje znát svou přesnou polohu v reálném čase, o což se stará **přijímač globálního navigačního družicového systému**. Tento signál může být někdy problematické získat (například ve vysoké a husté zástavbě nebo v horských údolích), proto ještě systémy disponují i **akcelerometry** či **gyroskopy**, které také mohou určovat polohu vozidla. Tyto senzory se dnes již běžně vyskytují v nových modelech aut, kde slouží k doplňování informací při pohybu například v tunelech nebo garážích. Signálem z družic se určí absolutní a přibližná pozice vzhledem k celému světu, osazené senzory pak poskytují naopak velmi přesné informace o bezprostředním okolí.

Důležité také je, aby autonomní vozidlo identifikovalo i jiná pohybující se vozidla a zvládalo předvídat i jejich pohyb – polohu, rychlosť nebo směr jízdy. Pro komunikaci s těmito vozy, případně s novou moderní dopravní infrastrukturou, by mělo být vše **datově propojené** pomocí obousměrné datové komunikace, aby o sobě jednotlivá vozidla a všichni další aktivní i pasivní účastníci silničního provozu vzájemně věděli [26]. Hlavní je, aby byla sbíraná data využívána co nejrychleji a co nejpřesněji, protože jen tak bude zaručeno, že bude plně autonomní doprava spolehlivá a bezpečná.



Obrázek 3.2: Ukázka toho, jak může být detailně monitorováno celkové okolí vozidla pomocí jednotlivých senzorů, kamer či družicových systémů. Obrázek převzat z [26].

### 3.3 Základní fyzikální veličiny související s pohybem modelu

V této kapitole budou teoreticky představeny základní fyzikální veličiny ovlivňující pohyb modelu se na dráze. S těmito veličinami je třeba počítat při vývoji, případně je možné je zaznamenávat a následně vyhodnocovat. Na základě znalostí získaných z těchto dat pak může být vylepšován algoritmus, případně mohou být použity pro zaznamenání parametrů dráhy či o tom, jak se na ní model choval.

Pohyb jako takový je definován jako **změna polohy hmotného bodu** (hmotným bodem nahrazujeme pro zjednodušení skutečné těleso, jedná se o jediný bod s totožnou hmotností, který by byl těžištěm skutečného tělesa). K popisu pohybu se používají tři základní veličiny – poloha, rychlosť a zrychlení. Oblast fyziky zabývající se pohybem a těmito veličinami se jmenuje **kinematika**.

Trajektorie je množina bodů, kterými hmotný bod prochází při svém pohybu. Fyzikální veličina délka  $s$ , která má základní jednotku metr ( $m$ ), je délka trajektorie, kterou hmotný bod opíše za čas  $t$  (jednotka ( $s$ )). Podle trajektorie můžeme pohyb rozlišit do těchto kategorií:

- **pohyb přímočarý** – trajektorií je zde část úsečky, někdy také nazýván pohybem posuvným
- **pohyb křivočarý** – trajektorií je zde křivka či její část, v případě kružnice či její části se jedná o pohyb otáčivý
- **pohyb složený** – pohyb, který je kombinací dvou předchozích pohybů

#### 3.3.1 Rychlosť hmotného bodu

Rychlosť hmotného bodu je základní fyzikální veličina popisující pohyb. Udává změnu polohy hmotného bodu v čase, tedy jaký čas  $t$  je potřeba na uražení dráhy  $s$ . Jedná se o vektorovou veličinu (což znamená, že vždy má nějaký směr a velikost). Označována je písmenem  $v$  a základní jednotkou jí je metr za sekundu  $\frac{m}{s}$ , v praxi se pak také používá jednotka kilometr za hodinu  $\frac{km}{h}$  a vztah mezi nimi je  $1 \frac{m}{s} = 3,6 \frac{km}{h}$ . Rozlišujeme mezi rychlosťí okamžitou a rychlosťí průměrnou.

##### Rychlosť okamžitá

Okamžitá rychlosť v rovnici (3.1) označena jako  $v$  je definována jako podíl aktuálního přírůstku dráhy  $\Delta s$ , ke kterému dojde za časový úsek  $\Delta t$ . Tento časový úsek je velmi malý, v ideálním případě co nejmenší (limitně se blížící nule). V praxi se pak tato rychlosť využívá, když potřebujeme vědět, jak rychle se aktuálně pohybujeme (například aktuální rychlosť vozidla na silnici udávaná tachometrem v závislosti na otáčení kola v čase). V této práci bude mít spíše informativní charakter a bude získávána právě z otáčení kol.

$$v = \lim_{\Delta t \rightarrow 0} \frac{\Delta s}{\Delta t} \quad (3.1)$$

##### Rychlosť průměrná

Rychlosť průměrná, pro rozlišení v rovnici (3.2) označena jako  $v_p$ , je definována jako podíl celkové uražené dráhy  $s$ , kterou hmotný bod urazil za celkový čas  $t$  a právě času  $t$ . Získáváme tím informaci, jakou stálou rychlosťí bychom se museli konstantně pohybovat, abychom

v daném čase tuto dráhu urazili. Pokud bychom pak tento časový nekonečně zmenšovali, dostali bychom se zpět k rychlosti okamžité z rovnice (3.1).

$$v_p = \frac{s}{t} \quad (3.2)$$

Na základě rychlosti můžeme pohyb rozdělit do dalších dvou kategorií, a to:

- **pohyb rovnoměrný** – rychlosť je při tomto pohybu vždy stejná (konstantní), hmotný bod tedy urazí ve všech stejně velkých časových intervalech vždy stejný úsek dráhy
- **pohyb nerovnoměrný** – rychlosť je zde proměnlivá, díky čemuž hmotný bod urazí ve stejně dlouhých časových intervalech různé úseky dráhy [18]

## 3.4 Mikrokontrolér

Mikrokontrolér (často také používána pouze zkratka MCU<sup>1</sup>) je ve své podstatě jednočipový počítač, který je navržen a následně naprogramován pro specifické úkoly. Velmi často je to součást vestavěných systémů, kde se mikrokontroléry starají o chod celého zařízení, díky čemuž se v dnešní době vyskytují téměř všude, kde je třeba nějaké řízení pomocí malého zařízení a není potřeba nějakého složitějšího operačního systému. Často fungují tak, že se zde vyskytují nějaké vstupní informace (například z nějakého typu senzoru), které jsou v mikrokontroléru zpracovány a předány na výstup (prezentace výstupu závisí na konkrétní aplikaci).

Výhodou také je poměrně malá spotřeba elektrické energie, příznivá cena díky použití jednoduchých a levných procesorů (jsou ve většině případů pro dané aplikace naprostě dostačující) a také to, že do jednoho čipu jsou zabudovány další moduly, díky čemuž již není potřeba používat další externí zařízení a pracují naprostě samostatně nezávisle na okolí.

Výrobci také hojně vyrábějí vývojové desky, které obsahují vše potřebné k tomu, aby se uživatel mohl připojit pomocí vstupně-výstupních pinů či periferií a naprogramovat to, co potřebuje. Právě těchto desek bylo využito v této práci, konkrétně vývojové desky FRDM KL27Z z portfolia firmy NXP Semiconducotrs (detailněji je popsána v části 4.1.1) pro hlavní chod aplikace a desky Arduino Uno především pro testovací účely.

Při pojednávání o mikrokontrolérech je potřeba také zmínit mikroprocesory, jenž jsou zaměřeny spíše na maximalizaci výkonu a nemají na jednom čipu zabudované žádné paměti a periferie. Často se vyskytují spíše ve stolních počítačích, v dnešní době však již jsou rozdíly mezi mikrokontroléry a mikroprocesory smazávány především díky levnější výrobě, která také dovoluje hustější a složitější výrobu čipů. Díky tomu právě mikrokontroléry často přebírají v některých případech funkce takové, které dříve byly řešeny na univerzálních počítačích s mikroprocesory [10].

### 3.4.1 Architektura procesorů

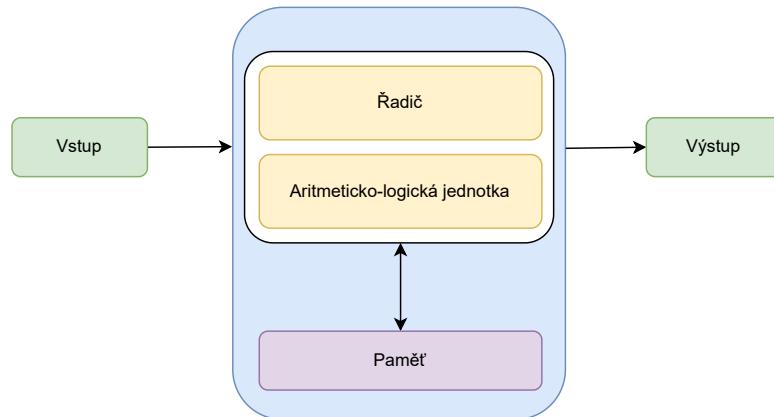
Jak byly postupně v minulém století vyvíjeny první počítače, vznikaly různé architektury, které nám popisují, jaké je uspořádání součástí uvnitř počítače. Do dnešní doby se zachovaly a používají se především dvě – Von Neumannova a Harvardská.

---

<sup>1</sup>MCU – Microcontroller unit (mikrokontrolerová jednotka)

## Von Neumannova architektura

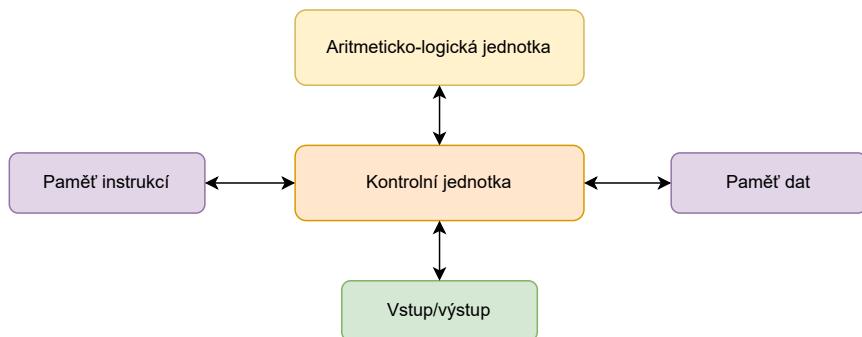
Jedná se o principem jednodušší architekturu, především díky složení z menšího množství součástí. Typická zde je společná paměť pro data i pro samotný program, který se pak skládá z jednotlivých strojových instrukcí, jenž jsou řízeny řadičem a vykonávají se sekvenčně. Díky společné paměti může být využita jediná sběrnice pro přenos dat i instrukcí, což však je i limitujícím faktorem, především kvůli rychlosti. Jedná se o univerzální a flexibilní architekturu, proto se v dnešní době využívá v osobních počítačích.



Obrázek 3.3: Ukázka blokového schématu procesoru při použití architektury Von Neumann. Převzato z autorovo bakalářské práce [23].

## Harvardská architektura

V této architektuře dochází k oddělení paměti pro data a paměti pro program. Díky tomu je možné využívat širší sběrnice pro připojení paměti dat. To je výhoda umožňující rychlý přístup k datům i instrukcím současně, což zvyšuje výkon aplikace. Díky tomu, že musí být použito více sběrnic i paměťových jednotek, se jedná o náročnější architekturu jak z hlediska technického, tak finančního. Více se využívá tam, kde jsou úlohy specifického rázu a nejsou tolik obecné, proto v dnešní době bývá využívána především v mikrokontrolérech [5].



Obrázek 3.4: Ukázka blokového schématu procesoru při použití Harvardské architektury. Převzato z autorovo bakalářské práce [23].

## 3.5 Prostředky na čipu mikrokontroléru

V této sekci budou postupně popsány teoreticky stežejní principy, které se využívají později v implementaci k ovládání motorů či servomotorů, převodu analogových hodnot napětí na hodnoty diskrétní nebo k měření délky pulzu či periodickému získávání hodnot ze senzorů.

### 3.5.1 Pulzně šířková modulace (PWM)

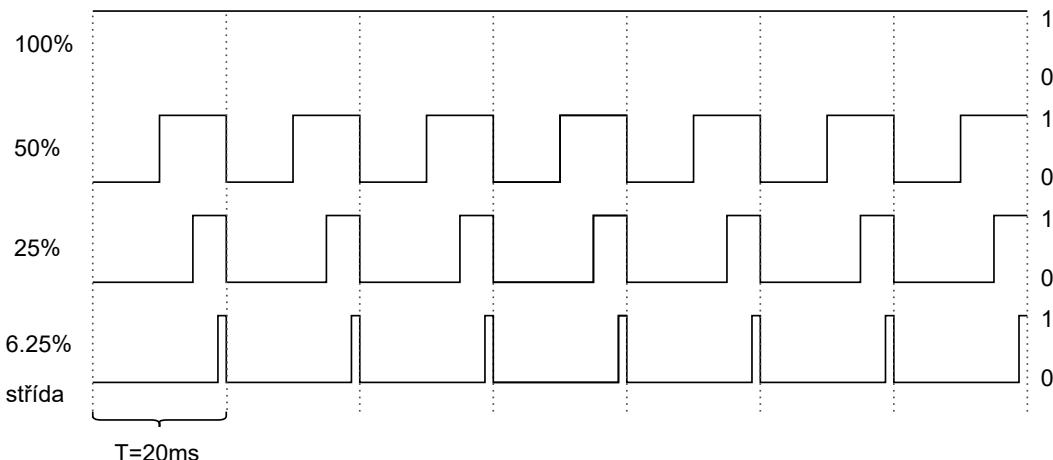
PWM<sup>2</sup> je metoda, při které dochází ke změnám nastavení parametrů řídícího signálu, který slouží k řízení či k regulaci elektronických zařízení.

Jedním z parametrů je samotná hodnota signálu – může se nacházet pouze ve dvou logických stavech. Buď v nulovém stavu (low), který značí napětí s hodnotou 0 V, nebo ve vysokém (high), který naopak značí maximální napětí (liší se v závislosti na konkrétním použití). V digitálním světě jsou to jediné hodnoty, které můžeme nastavit, a právě díky PWM můžeme simulovat jakoukoli hodnotu napětí mezi těmito stavy.

Dalším parametrem je pak perioda  $T$ , která nám udává časový úsek (jeden cyklus), během kterého jsou hodnoty signálu sledovány. Ta vychází ze spínací frekvence signálu (značí počet možných přepnutí mezi jednotlivými stavy) a je k frekvenci převrácenou hodnotou. Následně je získána **střída**, která značí poměrně, jakou část časového úseku byl signál v logické 0 a 1. Jedná se o hodnotu, která je vyjádřena v procentech [8].

Na ukázkovém výpočtu 3.3 níže vidíme příklad, kdy střída dosahuje hodnoty 50 %. Při délce periody 20 ms se signál v logické 1 nachází přesně polovinu času, tedy 10 ms. Graficky je tento jev také znázorněn na obrázku 3.5 v druhém řádku.

$$strida = \frac{dobaLogicke1}{T} \times 100 = \frac{10}{20} \times 100 = 50\% \quad (3.3)$$



Obrázek 3.5: Ukázka toho, jaké jsou výsledné hodnoty střídy v procentech v závislosti na tom, jaké jsou hodnoty signálu. Perioda 20 ms byla zvolena jako vzorová především proto, že byla v práci často využívána při řízení motorů (při frekvenci 50 Hz). Podobné grafické znázornění je možné zobrazit při sledování PWM signálu na osciloskopu.

<sup>2</sup>PWM – Pulse Width Modulation (pulzně šířková modulace)

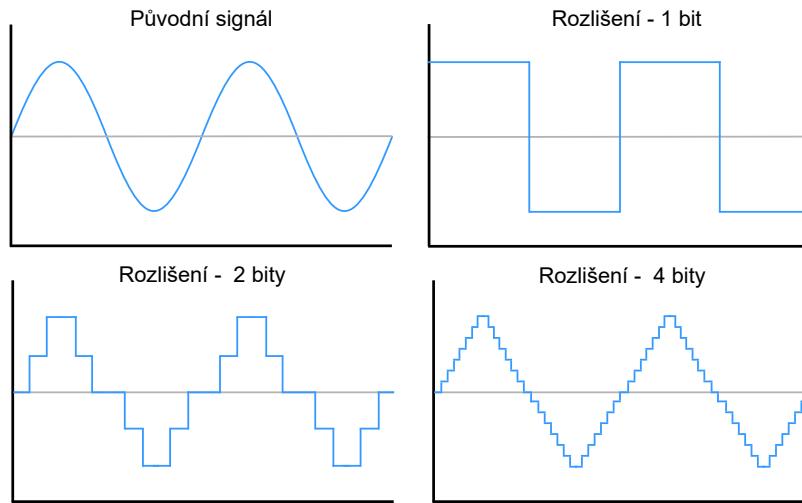
Využití najde PWM například při regulování otáček stejnosměrného motoru, pak hodnota střídy udává, kolik procent z maximálního možného výkonu motoru bude využito (podobně tomu tak je také u ventilátorů), nebo pro regulování svítivosti diody, kdy vyšší hodnota střídy znamená vyšší napětí a tedy i svítivost. Jelikož pracujeme s vysokými frekvencemi, nejsme schopni okem zaznamenat, že dioda bliká při přechodech mezi jednotlivými stavami, pouze zaznamenáme výslednou intenzitu. V této práci se také PWM používá pro ovládání servomotoru, kdy je dán rozsah střídy při dané délce periody, ve kterém je možné měnit natočení kol. Detailněji je toto popsáno v části 4.1.2.

Další příklad využití principu PWM je u regulátorů napětí, kdy dochází k spínání MOS-FET tranzistorů a v závislosti na poměru sepnutého a rozepnutého stavu dochází k regulaci napětí. Detailněji je pak tato součástka a její fungování představeno v části 4.2.2.

### 3.5.2 AD převodníky

Jelikož mikrokontroléry pracují vždy s digitální hodnotou, je třeba řešit, jak převádět data ze senzorů, které poskytují své hodnoty v podobě analogové. Ty v závislosti na měřené hodnotě generují na výstupu napětí úměrné měřené veličině, například teplotě u teplotního čidla či naměřené vzdálenosti u senzoru vzdálenosti. K tomu slouží AD převodníky, což jsou součástky, které tento spojitý signál převedou na signál diskrétní.

Vstupní signál je vzorkován a následně dochází k jeho kvantifikaci do různých diskrétních úrovní. To, kolik jich je a jaké tedy je jeho rozlišení, je určeno tím, o kolikabitový převodník se jedná. V případě, že by se jednalo pouze o jednobitový, mohl by vstupní signál být rozdělen pouze mezi dvě hodnoty – 0 a 1. S narůstajícím počtem bitů se zvyšuje to, s jakou přesností jsou hodnoty kvantifikovány, což znázorňuje obrázek 3.6.



Obrázek 3.6: Ukázka toho, jak je vzorkován původní signál (sinusoida) pomocí AD převodníků se zvyšujícím se rozlišením (počtem bitů převodníku).

Důležitá je u převodníků především vzorkovací frekvence, která musí být alespoň dvakrát vyšší než nejvyšší frekvence v signálu, aby se obnovil původní analogový signál, čemuž se říká Nyquistův teorém. Kdyby tento teorém byl porušen, mohlo by dojít k aliasingu a data by nebyla rekonstruována do digitální hodnoty správně [25].

### 3.5.3 Čítače

Důležitou roli hrají u mikrokontrolérů také čítače, díky kterým je možné přesně určovat časové úseky. Čítač pracuje na určité frekvenci a při každém jednotlivém ticku dojde k jeho inkrementaci. Při frekvenci 1000 Hz tak dojde k přičtení hodnoty každou 1 ms.

Když je dosaženo nejvyšší možné hodnoty, dojde k přetečení, o němž procesor informuje v podobě přerušení. Hodnota, do které se čítá, je určena podle toho, o kolikabitový čítač se jedná. Často se využívají například 16ti bitové čítače, které přetečou při hodnotě 65535. Díky tomu je například možné měřit časové úseky mezi nástupnou a sestupnou hranou nebo použít čítače k periodickému vyvolávání částí kódu, tak jako tomu bylo využito v této práci a je to popsáno později v implementaci.

## 3.6 Vybraná komunikační rozhraní a vývojová prostředí

V této sekci budou popsána použitá komunikační rozhraní, která byla použita pro připojení jednotlivých zařízení, jako jsou senzory či kamera a komunikaci s nimi. Budou také popsána jednotlivá vývojová prostředí, ve kterých docházelo k vývoji výsledného řešení této práce.

### 3.6.1 I<sup>2</sup>C

Jedná se o sériové rozhraní, které je založeno na standardu **I<sup>2</sup>C**<sup>3</sup>. To, že se jedná o sériové rozhraní, stejně tak jako ve všech později popsaných rozhraních v následujících sekcích znamená to, že jsou data posílána bit po bitu za sebou po jedné datové lince.

Sběrnice se v případě I<sup>2</sup>C skládá pouze ze dvou linek – **SCL**, která slouží k přenosu hodinového signálu a každý impuls je ekvivalentem k jednomu přenesenému bitu informace. Tento signál generuje *master* zařízení, které se zde vyskytuje vždy jedno a stará se o řízení komunikace. Ostatní zařízení pak mohou být na sběrnici připojena v režimu *slave*, kdy každé musí mít svou unikátní sedmibitovou adresu (může být připojeno až 127 takovýchto zařízení), podle které je určeno, komu jsou data poslána. Druhou linkou pak je **SDA**, po které dochází k samotnému přenosu dat. Díky tomu, že je zde pro data pouze jedna linka, pracuje směrnice v režimu *half-duplex*, kdy data mohou vždy proudit pouze jedním směrem mezi zařízeními [20].

Nejčastěji se v aplikacích používá například k připojení senzorů či displejů, v případě této práce na ní byly připojeny ve fázi návrhu řešení ultrazvukové senzory, barevné senzory či laserový senzor, nakonec ale žádný nebyl ve finálním řešení použit. Konektor umožňující připojení jakéhokoliv I<sup>2</sup>C zařízení je však na rozšiřující desce vyveden a může být kdykoliv použit v případě potřeby.

### 3.6.2 SPI

Sériové komunikační rozhraní **SPI**<sup>4</sup> je standard podobný dříve zmíněnému I<sup>2</sup>C, nicméně se jedná o starší rozhraní, které se skládá v základu ze čtyř linek v případě připojení jednoho zařízení. Je zde linka **SCLK** s hodinovým signálem a linka **CS**, která slouží pro určení toho, které *slave* zařízení zrovna bude komunikovat se zařízením *master*. Každé takto připojené zařízení pak tedy znamená jeden vodič připojený k mikrokontroléru navíc.

Pro přenos dat pak slouží linky **MISO** – Master In Slave Out a **MOSI** – Master Out Slave In, které slouží k přenosu dat mezi jednotlivými zařízeními, díky čemuž tato

<sup>3</sup>I<sup>2</sup>C – Inter-Integrated Circuit (meziintegrovaný obvod)

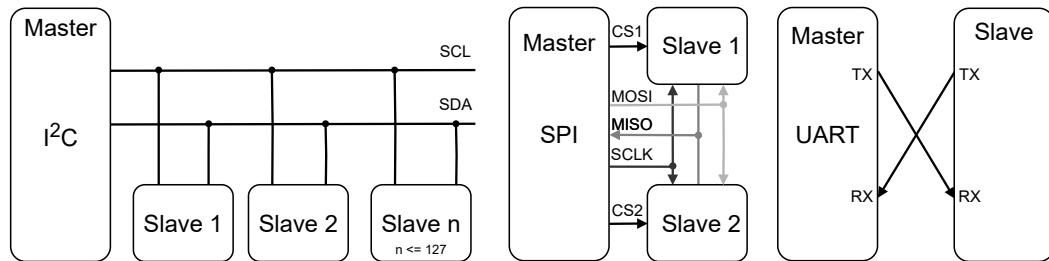
<sup>4</sup>SPI – Serial Peripheral Interface (sériové periferní rozhraní)

periferie pracuje v režimu *full-duplex*. Díky tomu se jedná o rychlejší sběrnici a i její princip je jednodušší než v případě I<sup>2</sup>C. Často se využívá k připojení komunikační rozhraní či sběrnici, především díky dříve zmíněné rychlosti. Je také standardem pro připojování SD karty, což bylo zvažováno a testováno i v této práci, nakonec však pomocí SPI byla připojena pouze kamera Pixy 2.

### 3.6.3 UART

**UART**<sup>5</sup> je nejstarší z dosud zmíněných rozhraní a jak již název značí, jedná se o asynchronní rozhraní, čímž se liší od předchozích dvou zmíněných. V praxi to znamená, že tu není žádný hodinový signál, který se stará o synchronizaci komunikace. Rychlosť rozhraní je zde pevně dána při inicializaci a k datům jsou přidávány start, stop a paritní bit, aby mohlo být rozeznáno, kdy jsou posílána data.

Absence hodinového signálu je výhodou v tom, že používají pouze 2 vodiče. Konkrétně se jedná o **TX** a **RX**, které jsou do kříže propojeny mezi zařízeními, z čehož však plyne i nevýhoda v podobě možnosti připojit pouze jedno zařízení. Omezená je i velikost rámce, ve kterém jsou data posílána, nejčastěji se jedná o 8 bitů. I přes to se však jedná o velmi rozšířené a používané rozhraní, dříve se využívalo k připojování tiskáren či modemů, dnes se využívají například v GPS či Bluetooth modulech. V případě této práce byl UART využit pro posílání logovacích dat, která pak dále byla přeposlána přes Bluetooth modul do externího zařízení [4]. Obrázek 3.7 porovnává všechna 3 zmíněná rozhraní z hlediska propojení jednotlivých zařízení.



Obrázek 3.7: Ukázka toho, jak jsou propojena master a slave zařízení u jednolitvých komunikačních rozhraní.

### 3.6.4 USB

**USB**<sup>6</sup> je univerzální a nejrozšířenější sériová sběrnice, která je využívána pro komunikaci mezi jednotlivými zařízeními a periferiemi, která již za dobu své existence od první verze prošla mnoha vylepšeními v podobě konektorů i jednotlivých standardů (především přenosových rychlostí). V této práci a celkově ve světě vestavěných systémů či mikrokontrolérů nejsou nijak důležité rychlosti, takže na zařízeních byly konektory a standardy starších verzí, ale nezpůsobovalo to žádné problémy.

Na desce Arduino UNO je dnes již v nových zařízeních nepoužívaný konektor USB typ B, nicméně zde se vyskytuje pravděpodobně kvůli své robustnosti (tento vývojový kit se často využívá ve školství, kde právě robustnost a odolnost proti zničení jsou pozitivními faktory), na desce od NXP je pak také starší, ale robustnější typ mini USB, na opačných

<sup>5</sup>UART – Universal Asynchronous Receiver-Transmitter (univerzální asynchronní přijímač/vysílač)

<sup>6</sup>USB – Universal Serial Bus (univerzální sériová sběrnice)

stranách pro připojení k zařízení se pak standardně vyskytuje USB-A. Této sběrnice je využito k tomu, že při jejím připojení dochází k napájení mikrokontrolérů (u desky od NXP v případě testování, kdy se model autonomně nepotřebuje pohybovat v prostoru) pomocí standardní hodnoty napětí 5 V, která je právě pro tuto sběrnici typická. Slouží také k tomu, aby do zařízení byl nahráván samotný kód, který je pak na zařízení spuštěn a vykonává danou funkci.

### 3.6.5 OpenSDA

Jedná se o otevřený sériový ladící adaptér s mikrokontrolérem Kinetis K20, který je vestavěn do vývojových desek od firmy NXP (dříve Freescale). Jedná se o pomyslný most mezi počítačem a mikrokontrolérem, kdy je toto rozhraní použito jak k samotnému programování, tak také k ladění a sériové komunikaci, ke které je využit USB kabel.

Díky použitým UART-USB převodníkům je tak možné na komunikačním COM portu sledovat například výpisy, které jsou z programu vyvolány pomocí PRINTF nebo si program zastavovat za běhu aplikace dle potřeby a sledovat, jaké hodnoty se aktuálně nacházejí v proměnných [11].

### 3.6.6 MCUXpresso IDE

MCUXpresso IDE<sup>7</sup> je integrované vývojové prostředí, které bylo vyvinuto firmou NXP a slouží k vývoji softwaru na mikrokontrolerech založených na jádru Arm® Cortex®-M, produkovaných primárně též firmou. Jedná se o poměrně jednoduché prostředí, které je jednoduché na používání a je založeno na dříve používaném prostředí Eclipse.

V prostředí se kromě klasického psaní kódu, jeho sestavení a nahrání do mikrokontroléra dají pohodlně v grafické podobě využívat konfigurační nástroje k nastavení vstupně-výstupních pinů, využitých periferií či časovačů a hodin. Umožňuje importování velkého množství vzorových knihoven a projektů, na kterých mohou být postaveny pokročilejší funkce. Poskytuje také možnosti detailního ladění, které je často využíváno v případě vývoje a testování projektu, samozřejmostí je pak sledování aktuálních hodnot proměnných či konzole pro zobrazení vypisovaných hodnot.

### 3.6.7 Arduino

Arduino<sup>8</sup>, jakožto jednoduchá vývojová platforma, bylo používáno společně s vývojovou deskou Arduino UNO především k prvotnímu testování jednotlivých součástí, například senzorů nebo motorů. Sloužila pak jako následná kontrola funkčnosti, pokud se nepodařilo zprovoznit funkčnost na desce s NXP. Pomocí Arduina také bylo možné programovat a pohodlně upravovat nastavení jednoho typu regulátoru otáček či měnit nastavení desky HC-05, jenž sloužila pro přenos dat a bude popsána později v sekci 4.7.

Výhodou je zde veliká rozmanitost knihoven, díky kterému můžou být kód rychle sestaven a nahrán do Arduino desky s připojenými senzory, které tak mohou být ihned otestovány. Především právě u senzorů bylo testováno, s jakou citlivostí či na jaké vzdálenosti jsou schopny reálně měřit a fungovat. Díky tomu bylo vyhodnoceno, zda vůbec má smysl implementovat jejich funkce ve složitějším prostředí, nebo najít jinou, vhodnější alternativu.

<sup>7</sup>MCUXpresso IDE – <https://www.nxp.com/design/design-center/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE>

<sup>8</sup>Arduino – <https://www.arduino.cc/>

# Kapitola 4

## Návrh řešení

V této kapitole bude popsán návrh řešení především v oblasti výběru vhodné elektroniky. Budou popsány jednotlivé součástky jak z hlediska jejich teoretického principu, tak z hlediska parametrů samotných součástek a toho, proč zrovna daná komponenta byla pro osazení v modelu zvolena. Samotnému návrhu řešení předcházelo ucelení představy o tom, jak se má model chovat a jaké elektronické součástky budou potřeba k tomu, aby se docílilo takového stavu, kdy bude moci model v autonomním režimu ovládat sám veškeré součástky dle potřeby, aby dosáhl co nejlepších výsledků v soutěži.

### 4.1 Volba základních řídicích prvků

V této sekci budou popsány elektronické součástky, které byly vyhodnoceny jako základní řídicí prvky této práce. Jedná se o vývojovou desku, motory a servo. Řešení napájení všech součástek je popsáno detailněji v sekci 4.2.

#### 4.1.1 Vývojová deska FRDM KL27Z

Během práce na tomto projektu bylo vyzkoušeno více vývojových desek z portfolia NXP, kdy každá měla nějaké výhody a nevýhody. Byly testovány i novější desky z řady FRDM (Freedom), které například měly i integrovaný slot na SD kartu, nicméně vyskytl se problém s testováním a laděním samotné aplikace, což znemožňovalo možnost jakéhokoliv vývoje, tudíž pak byla zvolena starší, ale pro tento projekt výkonově i vybavením dostačující deska **FRDM KL27Z**.

Nutné je také zmínit pro soutěž pořadatelem doporučenou vývojovou desku S32K144-Q100, kterou bylo také možné vyzkoušet. Díky jinému osazenému čipu je třeba k vývoji použít jiné vývojové prostředí od firmy NXP než bylo využíváno s deskou FRDM KL27Z, tudíž by bylo třeba přepsat i dosud vytvořený kód a seznámit se s rozdílným přístupem k programování. Samotné prostředí však prokazovalo nedostatky z hlediska optimalizace (zdlouhavé spouštění a načítání používaných knihoven) a i charakter desky jako takové spíše vybízí k tomu, že je najde využití spíše v průmyslových či automotive aplikacích, čemuž odpovídá i výkon, osazené komponenty a samotná velikost desky. Ta je vzhledem k povaze tohoto projektu až přehnaně vybavená z hlediska poskytnutých periferií, ale i na úrovni samotných vstupně-výstupních pinů. Proto nakonec bylo rozhodnuto tuto doporučenou desku nevyužít a zůstat na již zprovozněné platformě.

Deska FRDM KL27Z je levnou vývojovou platformou a je osazena mikrokontrolérem MKL27Z64VLH<sup>1</sup>, jenž je postaven na procesoru ARM Cortex-M0+ s frekvencí 48 MHz. Díky osazenému procesoru je optimalizována tak, aby mohla být efektivní a zároveň měla nízkou spotřebu. Poskytuje vše potřebné pro projekt – dostatek vstupně-výstupních pinů, ADC piny pro převod analogových vstupů na digitální hodnotu či naopak, ale i možnosti využití všech důležitých sériových komunikačních rozhraní (UART, SPI i I<sup>2</sup>C), dvě programovatelná tlačítka či LED<sup>2</sup> pro vizuální indikaci. Je možnost využít více typů časovačů, kdy byly použity téměř všechny, jenž deska nabízí, některé pro řízení pomocí PWM a zachytávání délky vstupních hran, jiné pak pro periodické dotazování a kontrolování různých příznaků či pro měření časových úseků. Drobou nevýhodou desky je to, že využívá logiky na úrovni 3,3 V, proto muselo být pro připojování některých senzorů nebo kamery Pixy 2 využito převodníků logické úrovně napětí, aby nedošlo k poškození samotného mikrokontroléra. Nicméně na této logice pracuje většina desek, tudíž by situace byla totožná i v případě zvolení jiné alternativy.

Napájení může být realizováno pomocí přivedení napětí na pin VIN (hodnota až 9 V, která je pak na samotné desce regulována), nebo pomocí připojeného USB, díky němuž je také možné používat OpenSDA, což je integrovaný ladicí obvod, viz část 3.6.5.

Deska byla uchycena k modelu pomocí přípravku, který byl vytisknut na 3D tiskárně a je umístěn přibližně ve středu modelu, aby všechny připojované komponenty byly co nejbližše.

#### 4.1.2 Servo MG996R

Na modelu je osazeno Servo MG996R, což je servo s poměrně velkou silou, které se může použít v modelářství (zatáčení modelů aut, natáčení klapek na letadlech) nebo například v robotech. Přímo na jeho výstup jsou připojena ramena, která se starají o natáčení kol přední nápravy.

Výhodou tohoto serva je to, že ozubená kolečka pro převod pohybu z motorku na kola jsou kovová a nikoli plastová, jak to bývá běžné u levných a malých serv. Konstrukce modelu byla vytvořena pro použití primárně s tímto servem, tudíž je přesně uchyceno pomocí šroubků k modelu.

Nevýhodou pak je to, že v případě, že se se servem něco stane, musí být model téměř kompletně rozebrán, aby bylo možné servo vyjmout. Během jednoho testovacího dne bylo jedno servo zničeno a muselo být nahrazeno za totožný kus.

To, jakou rychlosťí dokáže servo měnit jednotlivé úhly či jaký má točivý moment, závisí na tom, jaké je použito napájecí napětí. Rozsah bývá uveden v technické dokumentaci serva. Servo je možné napájet jak menším napětím, například přímo z mikrokontroléru, nebo také může být použit externí zdroj napájení, jako to bylo využito v případě této práce, kdy byla využita hodnota 7 V, jejíž využití je podrobněji popsáno v sekci 4.2 týkající se napájení součástek v modelu. Parametry použitého serva jsou přehledně znázorněny v tabulce 4.1.

Ovládání serva spočívá v tom, že je dána frekvence, která udává, jak často může docházet ke změně pozice a tedy natočení motoru. Nejčastěji bývá použita frekvence 50 Hz, stejně tak tomu bylo v případě použitých motorů. Na vstup motorku je tak posílan PWM signál a je generována strída, určující, v jaké poloze se zařízení bude nacházet.

<sup>1</sup>Dokumentace MKL27Z64VLH4 – <https://www.nxp.com/docs/en/data-sheet/KL27P64M48SF2.pdf>

<sup>2</sup>LED – Light-Emitting Diode (světelná dioda)

Servo MG996R		
Hmotnost serva	55 g	
Točivý moment	9,4 kgf/cm při 4,8 V	11 kgf/cm při 6 V
Rychlosť otáčení o $60^\circ$	0,17 s při 4,8 V	0,14 s při 6 V
Rozsah serva	$120^\circ$	
Operační napětí	4,8 V až 7,2 V	
Frekvence	50 Hz	

Tabulka 4.1: Technické parametry serva MG996R.

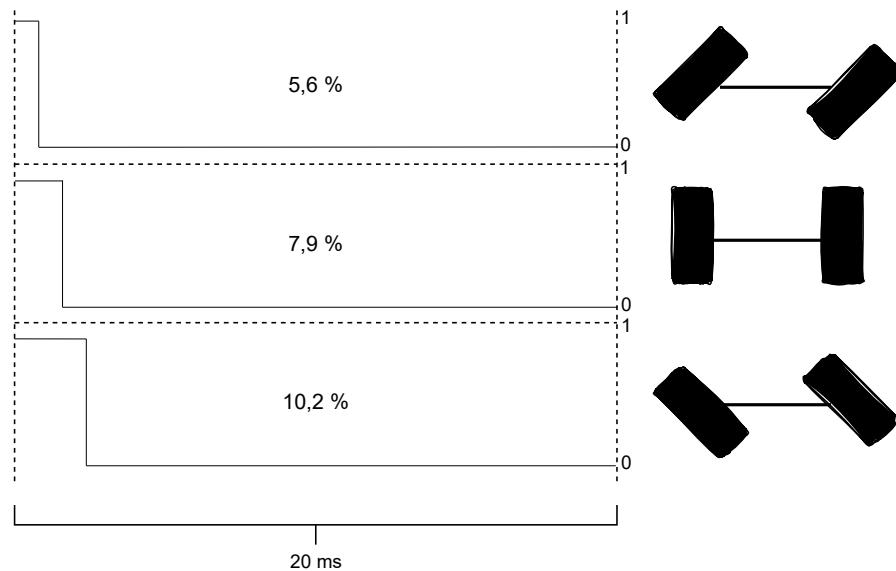
Zajímavostí je, že u těchto motorů je využita pouze malá část z možného rozsahu střídy. Konkrétně v případě použitého serva pro ovládání zatáčení modelu je možné využít pouze přibližně 7,2 % z rozsahu, který jde využít k nastavení jednotlivých úhlů natočení.

Když je generována střída pro úhel  $0^\circ$ , což je stav, kdy jsou kolečka vytočena nejvíce vpravo pro maximální úhel zatáčení, tak má střída hodnotu 4,3 %. Pro hodnotu  $60^\circ$ , což je přibližná střední hodnota pro pohyb rovně, se jedná o 7,9 % a pro  $120^\circ$ , což je hodnota maximálního natočení pro zatáčení vlevo při 11,5 % střídy. Rovnice (4.1) ukazuje, jaká je hodnota střídy v procentech pro změnu úhlu o jeden stupeň.

$$1^\circ = \frac{\text{rozsahstřidy}}{\text{rozsahserva}} = \frac{11,7 - 4,5}{120^\circ} = \frac{7,2}{120^\circ} = 0,06\% \quad (4.1)$$

Tyto hodnoty byly zjištovány pokusným procházením celého rozsahu a hledáním ideální polohy, především pro střední hodnotu, aby model při nastavení tohoto úhlu jel rovně. Muselo docházet k pravidelným kalibracím v závislosti na tom, jak byla upravována ramena, která spojují servo s kolečkem.

Hodnoty nakonec ještě lehce upraveny po zničení jednoho kusu tak, aby nebylo možné kola vytáčet na maximální možnou úroveň, co poskytuje model servu. Díky tomuto snížení byl lehce zmenšen rozsah zatáčení, ale již se pak nikdy nestalo, že by se servo zaseklo. Hodnoty lze vidět na obrázku 4.1.



Obrázek 4.1: Ukázka generování střídy pro nastavování jednotlivých poloh natočení kol.

### 4.1.3 Outrunnerový motor A2212/15T 930 kV

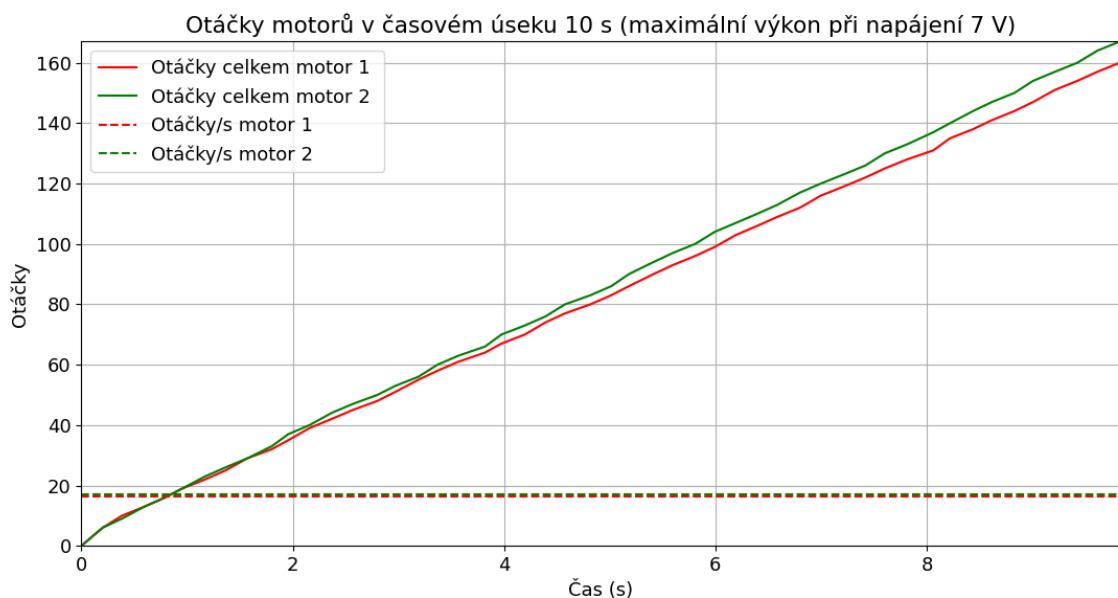
Jedná se o univerzální základní motor, který je využíván především v modelářství nebo v dětských hráčkách. Díky své velikosti je spíše vhodný pro modely do váhy 1 kg. Označení 2212 označuje průměr (22 mm) a délku (12 mm) statoru, 15T značí pak počet vinutí na cívkách.

V modelu jsou použity dva motory, každý pro jedno kolo zadní nápravy, napájené pomocí regulovaného vstupního napětí, které poskytuje baterie. K řízení jejich otáček je použito PWM. Jak motor vypadá a ukázka jeho zapojení s regulátorem otáček je možné vidět v následující části na obrázku 4.3.

Parametr 930 kV značí, že na každý jeden V napájecího napětí je v ideálním případě dosaženo 930 otáček za minutu. V realitě jsou hodnoty dosti rozdílné, především v důsledku tření (v částech motoru, na připojených kolejích). Výsledky testovacího měření výkonu těchto motorů jsou vidět na obrázku 4.2 níže. Rozsah napájecího napětí spolu s dalšími technickými parametry motoru je uveden v tabulce 4.2.

Motor A2212/15T	
Hmotnost motoru	47 g
Účinnost	80 %
Napájecí napětí	7 až 12 V
Napěťová konstanta	930 kV
Počet Li-Pol článků	2 až 3

Tabulka 4.2: Technické parametry motoru A2212/13T



Obrázek 4.2: Ukázka toho, jaký byly naměřeny otáčky při napájení motorů 7 V (kola se nacházela ve vzduchu a běžela naprázdno). Otáčky jsou průměrně přibližně na úrovni 18 ot/s, tedy přibližně 1080 za minutu, což jsou hodnoty výrazně menší než teoretický výkon 6510 ot/min které uvádí výrobce.

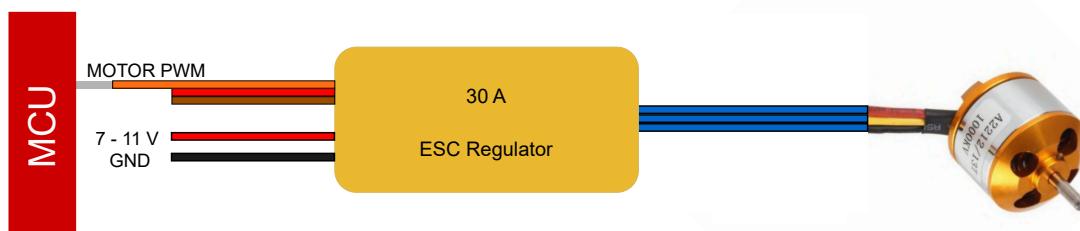
#### 4.1.4 Regulátor otáček

Jelikož moderní bezkartáčové motory, které byly popsány v části 2.3.2, již nestačí řídit pouhým snižováním či zvyšováním napětí, je třeba použít regulátor, který přijímá řídící signály a podle nich řídí otáčky samotného motoru. Regulátor otáček (používána zkratka ECS<sup>3</sup>) je tedy takový mezičlánek mezi zdrojem napájení, ovladačem, v našem případě mikrokontrolérem, jehož signály jsou řídící, a samotnými motory.

Při hledání vhodného návrhu řešení se ukázalo jako poměrně složité zprovoznit a naučit se ovládat samotné BLDC motory pomocí ECS. Při převzetí modelu byly osazeny regulátory **BLHeli-S**, které se na první pohled jevily jako ideální řešení. Regulátory mohly být připojeny přes Arduino k počítači, kde pak bylo možné v doporučeném programu pohodlně nastavit to, při jaké hodnotě střídy budou maximální/minimální otáčky, jak hlasité bude indikační pípnání samotných motorků a plno dalších užitečných parametrů. Problém však nastal při připojení k mikrokontrolérovi, kdy se nepodařilo najít správnou sekvenci signálů, která by sloužila k tomu, že akceptuje nastavení, které v regulátoru již bylo nahráno pomocí programu, nebo dojde k novému nastavení a inicializaci právě pomocí určité posloupnosti signálů zaslaných do zařízení pomocí PWM. Motory většinou při pokusech o nastavení rozsahu otáček pouze pípaly, později se pak již také zasekly a začaly se přehřívat.

Nakonec tedy byly zvoleny pro výsledný návrh jiné, v podstatě bezejmenné regulátory<sup>4</sup>. Ty jsou z hlediska principu a ovládání na první pohled výrazně jednodušším zařízením. Pro jejich inicializaci stačí zaslání pouze signálu značícího maximální otáčky a po určité chvíli signálu značícího otáčky minimální.

Problémem tohoto typu je nekompletní a nepřesná dokumentace, která se dokonce značně liší od reality. Nepodařilo se podle žádného návodu k tomuto regulátoru vstoupit do programovacího módu, ve kterém by bylo možné nastavit mód brzdění modelu. Dokumentace či různé návody v podobě videí popisují různé postupy, nicméně žádný se nepodařilo přivést do reality i při testování na různých platformách. Regulátor pracuje také na frekvenci 50 Hz a rozmezí, ve kterých je nastavována rychlosť pomocí PWM, se pohybuje v hodnotách střídy přibližně mezi 1 ms a 2 ms. Využívá se tedy pouze přibližně 5 % celkového rozsahu, což je tradiční hodnota, která je pro řízení motorů pomocí regulátorů využívána, nicméně je v rozporu s dokumentací součástky, kterou poskytuje prodejce, u kterého byla daná součástka pořízena, která mluví o možnosti využití střídy v plném rozsahu [7]. Princip ovládání a problémy, které bylo třeba dálé vyřešit, jsou detailněji popsány v implementační části práce, konkrétně v části týkající se implementace ovládání pohonu 5.2.



Obrázek 4.3: Ukázka toho, jak je propojen motor s regulátorem otáček a mikrokontrolérem. Nepřipojené vodiče by mohly sloužit jako napájení pro další součástky v případě potřeby.

<sup>3</sup>ECS – Electronic Speed Controller (elektronický kontroler rychlosti)

<sup>4</sup>ESC regulátor – <https://www.gme.cz/v/1507908/regulator-otacek-pro-bezkartacove-motory-30a>

## 4.2 Napájení modelu

V této sekci bude detailně popsáno, jak bylo pro model navrženo napájení a celkové rozvedení potřebných úrovní napětí pro jednotlivé součástky dle potřeby.

### 4.2.1 Baterie

K napájení modelu byla využita LiPol baterie, která je dle pravidel jediným zdrojem napájení při běhu modelu. Tento typ baterií patří v dnešní době k nejčastěji používaným zdrojům napájení v oblasti RC modelů, ale také v téměř všech běžně používaných elektronických zařízeních (mobilní telefony, notebooky atd.).

Výhodou tohoto baterií je to, že jsou velmi tenké, lehké a mohou být vytvarovány do téměř jakéhokoliv tvaru podle toho, kde přesně budou použity. Poskytují větší kapacitu, netrpí tolík na samovybíjení, které je při správném používání téměř zanedbatelné, a okolní teplotní prostředí na ně téměř nemá vliv. V běžných podmínkách se dají označovat za velmi spolehlivé [13]. Jejich nevýhodou může být riziko vznícení nebo dokonce výbuchu při mechanickém poškození. Je tedy nutné dodržovat pravidla stanovená výrobcem při nabíjení, vybíjení i skladování. Pokud dojde k vybití baterie pod hodnotu 3 V, může dojít k výraznému zhoršení vlastností či k úplnému zničení.

Konkrétně byla využita baterie **DroneArt** s kapacitou 2300 mAh. Jedná se o tříčlánkovou baterii s označením 3SP1, což znamená, že se jedná o články zapojené do série, kdy každý jednotlivý článek má napětí 3,7 V, celkem tedy baterie poskytuje napětí 11,1 V (v reálnitě ale až 12 V). Baterie má dva výstupy – balanční konektor, který musí být zapojen při nabíjení a zaručuje to, že se budou všechny články v baterii nabíjet stejně a nedojde k situaci, kdy jeden článek bude nabité výrazně více než jiný. Druhým výstupem je konektor XT-60, který už je pouze konektorem typu "samec". Aby baterie mohla být odpojována, byl na tento konektor pouze přiveden opačný konektor s výstupem v podobě vodičů, které pak již byly přichyceny v 2,54 mm konektoru na osazenou rozšiřující desku mikrokontroléru, na které bylo následně napětí dále rozvedeno a regulováno dle potřeby.

Při testování a ladění běhu aplikace bylo samozřejmě využito kromě této baterie i napájení přes USB, nicméně to sloužilo pouze k napájení samotného mikrokontroléru pro potřeby nahrání programu. Jelikož USB rozhraní poskytuje pouze napětí 5 V, nebylo možné jím napájet a testovat motory a další součástky, proto baterie byla zapojena téměř ve všech případech, kdy se na zařízení cokoliv testovalo.

### 4.2.2 Regulátor napětí

Baterie, která je jediným zdrojem napájení, má na svém výstupu napětí teoreticky 11,1 V (reálně je hodnota vyšší až k hodnotě 12 V), což je pro všechny součástky příliš velká hodnota (teoreticky by ji zvládly motory, ale i tam byla na základě testování využita hodnota výrazně nižší). Z tohoto důvodu je nutné použít ke snížení napětí na potřebné hodnoty regulátory napětí.

K regulaci napětí byly zvažovány dva různé typy součástek – LDO<sup>5</sup>, který mění napětí na pevnou, z výroby stanovenou hodnotu. Tato součástka mění veškeré nevyužité napětí na teplo a má proto poměrně velké ztráty. Během testování byl používán modul se stabilizátorem **AMS1117**, ale podařilo několikrát zničit (nikdy nebylo vyzozorováno, proč se tomu tak stalo), bylo tedy vyhodnoceno, že bude praktičtější využít jiné alternativy.

<sup>5</sup>LDO – Low-Dropout Regulator (stabilizátor s nízkým úbytkem napětí)

Nakonec bylo zvoleno řešení pomocí DC/DC step-down měniče s řídícím obvodem **LM2596**, a to především díky tomu, že je možné si pomocí potenciometru přímo nastavit výstupní hodnoty napětí, což poskytovalo flexibilitu při testování. Díky nízkým ztrátám jde také o efektivnější variantu. V principu jde u těchto měničů o to, že MOSFET tranzistor spíná mezi nulovým a maximálním napětím. Podle toho, jak dlouho tranzistor setrvá ve stavu sepnutém, se určí střída (hodnota mezi 0 a 100 %), která určuje, jaká procentuální část původního vstupního napětí se bude vyskytovat na výstupu. Princip této regulace je tedy stejný jako regulování pomocí PWM, které bylo popsáno dříve v části 3.5.1. Toto nemůže být využito vždy, protože některá citlivá zařízení nemusí snést vysoké (byť velmi krátké) hodnoty napětí na vstupu, ale pro naše potřeby v práci je toto ideální řešení [2].

#### 4.2.3 Vysvětlení zvolených úrovní napětí

Vstupní napětí, které dodává baterie, je pomocí dvou DC/DC step-down měničů sníženo na dvě různé úrovně napětí. Jedná se o úroveň 7 V, která je přivedena na vstup serva a také to je úroveň, kterou jsou napájeny samotné motory. Motory by mohly být napájeny i bez jakékoliv regulace pomocí 11,1 V, nicméně při této hodnotě byly otáčky i při minimální hodnotě střídy moc velké (výkon BLDC motorků totiž přímo úměrně závisí na tom, jaké napětí je přivedeno na vstup). Byla proto zvolena právě hodnota 7 V, která byla otestována jako vhodná vzhledem k tomu, aby minimální střída představovala pomalou jízdu modelu. Detailněji je práce s motory a jejich výkonem popsána v části s implementací 5.2. Pro servo to naopak je maximální napájecí hodnota, díky které může docházet ke změnám polohy a tudíž natočení kol v co nejkratším časovém úseku od vytvoření změny.

Druhou hodnotou je 5 V, která slouží k napájení desky mikrokontroléra, pokud zrovna není napájen pomocí USB. Tento měnič by bylo teoreticky možné vynechat a využít samotného mikrokontroléra, který na VIN vstupu zvládá hodnoty až 9 V a pomocí svého interního LDO převádí hodnoty na 5 a 3,3 V. Aby se mikrokontrolér zbytečně nezahříval, bylo využito i druhého měniče s výstupní hodnotou napětí lehce přes 5 V, který slouží k napájení mikrokontroléra a některých senzorů.

Theoreticky mohlo být využito toho, že ESC regulátory mají v sobě interní regulátor napětí na 5 V a mohly by být pro napájení využity, nicméně při testování často docházelo k tomu, že napájení k regulátorům a zároveň motorům bylo odpojeno, tudíž bylo praktičtější využití dvou regulátorů, které vytvoří tyto požadované hodnoty automaticky v okamžiku, kdy je připojena baterie. Poslední hodnotou napětí, se kterou pracují zbylé senzory, je hodnota 3,3 V. K jejímu získání dochází přímo z vývojové desky pomocí interních regulátorů a dále je rozvedena dle potřeby.

#### 4.2.4 Převodník úrovní napětí

Při návrhu napájení se bylo potřeba vypořádat s faktum, že vývojová deska FRDM KL27Z pracuje s logickou úrovní napětí 3,3 V, což znamená, že v případě nastavení výstupních pinů na logickou hodnotu 1 se zde vyskytuje hodnota blízká právě této. Problematictější jsou však vstupní piny s digitální hodnotou směřující z některých senzorů, které pracují na úrovni 5 V. Jedná se primárně o ultrazvukové senzory HY-SRF05 či kamery Pixy 2, jenž budou popsány později v části 4.3.1 a 4.5.3. Tyto senzory by mohly přivést na vstup napětí vyšších hodnot, což by desku mohlo poškodit. V dokumentaci čipu výrobce zmiňuje toleranci pouze 0,3 V nad logickou úroveň, konkrétně tedy maximálně 3,6 V, což signály ze senzorů výrazně převyšuje. Tabulka 4.3 shrnuje, na jakých úrovních napětí jednotlivé součástky pracují.

Bylo tedy nutné použít převodník úrovní napětí. Na ten je přiveden signál s vyšším napětím a převodník se postará o regulaci a hodnotu, která může být bezpečně připojena na digitální vstup desky. V případě digitálních pinů nedochází k žádné ztrátě informace, k té by došlo pouze v případě regulace analogové hodnoty napětí, která by byla ochuzena o informace, které mohla poskytnout nad úrovní 3,3 V.

11 V	7 V	5 V	3,3 V
MOTORY			
	SERVO		
	Napájení desky		
	Kamera Pixy 2 -><-		
	HY-SFR ultrazvukový senzor vzdálenosti -><-		
	GP2Y0A41SK0F IR senzor vzdálenosti <-		
	VL53l0X laserový senzor vzdálenosti -><-		
	Modul s hallovým senzorem <-		
	TCS3200 senzor barev <-		
	HC – 05 UART převodník ->		
	CRTT5000 IR tracking senzor <-		

Tabulka 4.3: Tabulka ukazující jednotlivé součástky a to, na jakých úrovních napětí mohou pracovat. Problematické jsou žlutě zabarvené součástky se šípkou <-, která značí, že bude jejich výstup může mít až 5 V. Ten pak bude přiveden na piny mikronoktroléru a je třeba tedy využít převodníků úrovně napětí.

## 4.3 Senzory pro primární navigaci

V této sekci budou popsány dva druhy snímačů obrazu – lineární snímač snímající jeden řádek pixelů a 2D kamera Pixy 2, což je hlavní snímač doporučený pro v soutěži NXP Cup. Výhodou těchto snímačů oproti senzorům na přímou detekci čáry, které budou popsány v následující sekci 4.4 je to, že monitorují prostor před modelem, tudíž dokáží na změnu dráhy či na výskyt cílové čáry reagovat dříve než až v okamžiku, kdy se model na čáru fyzicky nachází.

### 4.3.1 Kamera Pixy 2

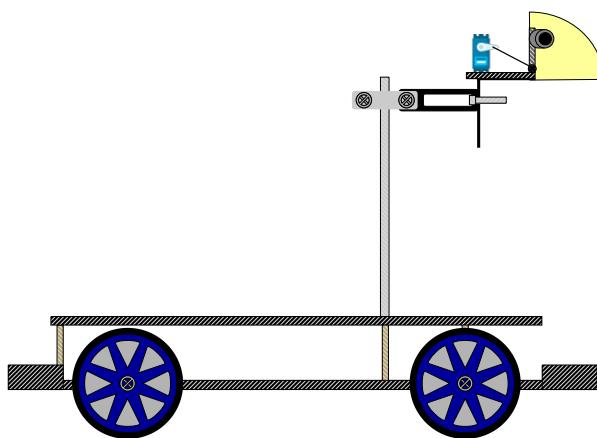
Pixy 2 je malá kompaktní kamera, snadno použitelná jako inteligentní kamerový systém pro malá autonomní robotická zařízení, kterým díky svým funkcím může poskytnout důležité informace o okolí. Jedná se o nástupce již dříve vydané verze PixyCam, nyní se však jedná o rychlejší, menší a výkonnější verzi, ke které byly přidány funkce, například algoritmy pro sledování čar. Kamera také umí detektovat průsečíky vektorů, umí číst čárové kódy a naučit se rozpoznávat a sledovat barevné objekty, přičemž může najednou sledovat až 7 různě barevných objektů. Má integrovaný zdroj světla v podobě LED a pracuje se snímkovací frekvencí 60 snímků za sekundu.

Výhodou je, že velké množství dat je již zpracováváno přímo v kameře a nezatěžují tak taklik mikronoktrolér, kterému již poskytnou předpřipravená data. V případě sledování čar například poskytne informace v podobě informací o jednotlivých vektorech (vždy 2 body, které značí vektor, který může být sledován a k němu index, který slouží k identifikaci vektoru). U sledování objektů pak poskytne informaci o tom, o jaký objekt se jedná a na

jaké souřadnici se v obraze vyskytuje. Ke komunikaci s cílovým zařízením může být použito některé z rozhraní SPI, I<sup>2</sup>C, USB či UART. Výhodou této kamery je, že může nahradit velké množství senzorů.

### Využití kamery Pixy2

Konkrétně bylo využito připojení pomocí SPI konektoru na desce s kamerou, jenž byl připojen přímo na rozšiřovací desku. Kamera samotná byla osazena do dodaného přípravku, jenž je následně uchycen na kotvící tyč, která dovoluje umístění do dostatečné výšky nad modelem. K ovládání směru a úhlu záběru jsou k dispozici 2 serva, využito však bylo pouze jedno, které poskytuje možnost nastavit zorný úhel záběru – v rozmezí od nasměrování přímo na plochu před modelem po stav, kdy kamera kouká vodorovně do prostoru před modelem.



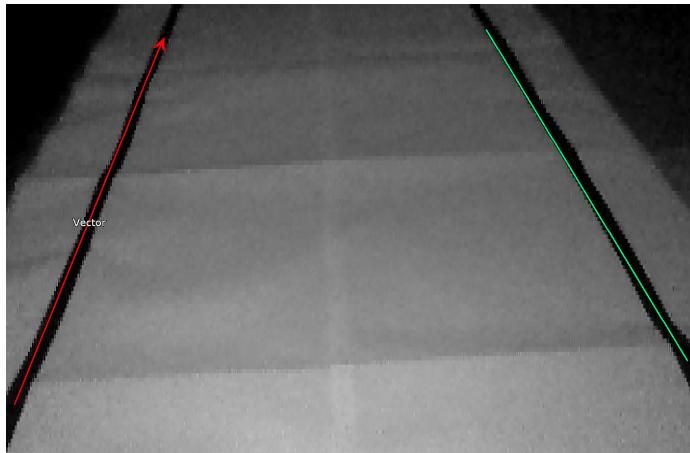
Obrázek 4.4: Zjednodušený náčrtkek toho, jak je nad modelem umístěna kamera a do jakých úhlů může být nastavena (žlutá výseč).

Poloha může být libovolně nastavena a výhodou je, že se o ovládání serva stará samotná kamera. Té je pouze přes rozhraní SPI zaslána sekvence bytů s informací o tom, do jaké pozice se má servo pootočit. Díky tomu nemusí být servo zapojeno na pin mikrokontroléru generující PWM signál. Druhé servo nebylo použito, protože se nepředpokládá, že by kamera musela být natáčena ještě v další ose.

### Software PixyMon

Ke kameře Pixy 2 je poskytnut od výrobce software PixyMon<sup>6</sup> sloužící ke konfiguraci a drobnému ovládání kamery. Po připojení pomocí USB kabelu tak mohou být nastaveny parametry kamery – především se jedná o různé nastavení snímání v závislosti na aktuálních světelných podmínkách – může být nastavena minimální či maximální délka snímaného vektoru, stejně tak jeho šířka (konkrétně byla nastavena taková šířka čáry, aby se eliminovalo zachytávání tenkých čar z okolí nebo naopak velkých černých ploch). Je také omezeno, kolik vektorů najednou může být snímáno, či nastaven jas. Všechno toto nastavení se uloží do kamery a zůstane zachováno i při připojení přes rozhraní SPI z mikrokontroléru, kde jediné, co jde nastavit přes toto rozhraní, je jas, čehož ale nebylo využito.

<sup>6</sup>PixyMon – <https://pixycam.com/downloads-pixy2/>

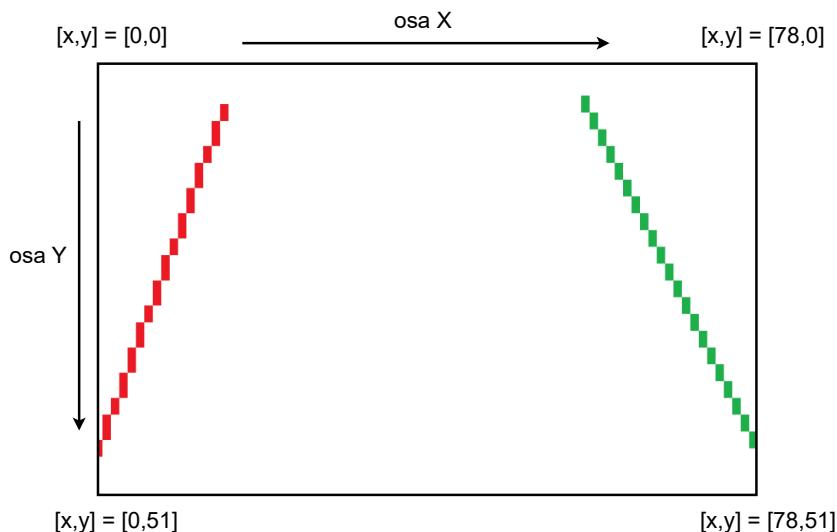


Obrázek 4.5: Ukázka toho, co je vidět v aplikaci PixyMon po připojení kamery k PC pomocí USB. Červeně je vždy označován primární vektor.

### Práce s Pixy 2

Kamera posílá přes rozhraní na vyžádání souřadnice, přičemž pro každý vektor je zaslán počáteční bod, koncový bod a také unikátní index tohoto vektoru. Kamera se vždy v dalším zachyceném snímku snaží navázat na tyto již jednou zachycené vektory. Pokud se to podaří, v dalším odeslání bude zaslán vektor se stejným indexem, což značí, že došlo k navázání na vektor předchozí. Detailněji je práce s Pixy 2 popsána v části práce, které se týká implementace řešení, konkrétně v sekci 5.6.

Rozlišení, se kterým kamera pracuje, je kvůli rychlosti zaznamenávání i přenosu poměrně nízké, konkrétně  $79 \times 52$  pixelů a je znázorněno na obrázku 4.6 níže. Díky tomu může být informace o jednom vektoru přenesena přes rozhraní v pouhých 6 bajtech (x a y souřadnice každého ze dvou bodů a 2 bajty s indexem vektoru).



Obrázek 4.6: Ukázka toho, v jakém rozlišení a systému souřadnic snímá kamera při módu sledování čar. Jedná se stejný pohled jako na obrázku 4.5.

### 4.3.2 Lineární snímače

Alternativou k použití 2D kamery Pixy 2 jsou lineární (často také nazývané 1D) kamery či řádkové snímače. Jedná se o typ snímače, který byl do soutěže dříve doporučován a také jej dodnes využívá většina týmů, především pokud pracují s modely, které se již účastnily minulých ročníků soutěže.

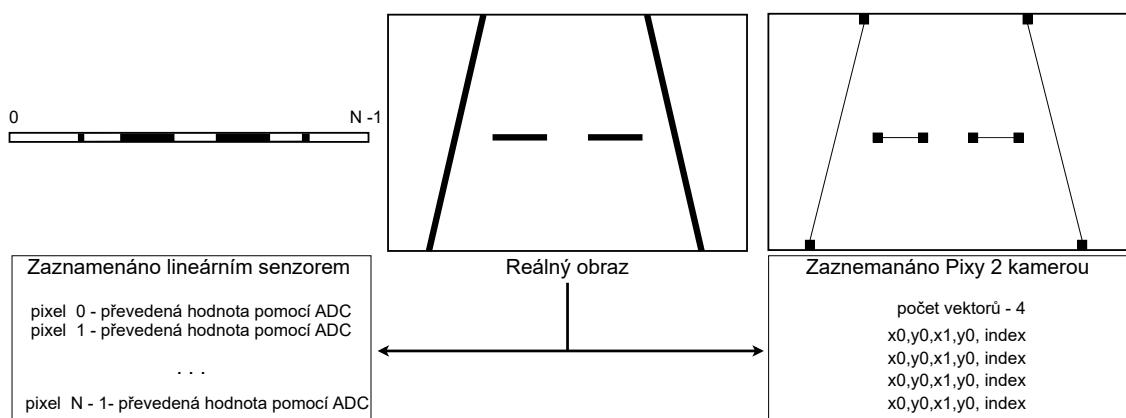
Hlavní výhodou těchto snímačů oproti dříve popsané kameře Pixy 2 je rychlosť, jakou mohou být data zpracovávána. V principu fungují tak, že je vždy zaznamenán jeden řádek pixelů, rozlišení se pohybuje v násobcích dvou, jednoduché a dostačující mají 128 pixelů. Snímač poskytnutý k této práci, který může být vidět na obrázku 4.7, jich má 2048 při použití maximálního rozlišení. Každý jeden pixel během stanoveného expozičního času vyhodnotí, jak velké množství světla bylo zachyceno, převede ho na hodnotu elektrického napětí, které je pak pomocí AD převodníků převedeno na digitální hodnotu, jež je poslána k následnému zpracování.



Obrázek 4.7: Ukázka lineárního snímače S11108, který by mohl být použit jako alternativa pro 2D kameru.

Data pak mohou být vyhodnocována na základě tohoto jednoho řádku (bylo by využito v případě navigace v soutěži), nebo může být zachytáváno více řádků a pak může dojít k rekonstrukci obrazu. Díky tomu bývají použity například pro kontrolu výrobků na výrobních pásech, mohou detektovat hrany objektů nebo například číst čárové kódy či etikety.

Pro soutěžní prostředí je výhodou těchto senzorů také to, že nepotřebují striktně mít bílé pozadí pod soutěžní dráhou, což kamera Pixy 2 nutně potřebuje, jinak není schopna spolehlivě detektovat vymezující čáry dráhy. Rozdíl mezi tím, co snímají a poskytují popisované snímače, je vidět na obrázku 4.8.



Obrázek 4.8: Ilustrace ukazující rozdíl mezi tím, co zaznamenává při jednom měření lineární snímač (N je počet počet snímaných pixelů) a informacemi o vektorech, jenž zaznamenává kamera Pixy 2.

Bylo možné testovat lineární senzor obrazu **S11108**, který má snímací rozlišení 2048 pixelů a možnou frekvenci čtení dat 10 MHz (při každé nástupní hraně synchronizačního CLK signálu je čten jeden pixel), což poskytuje mnohonásobně rychlejší možnost reagovat na změny než při použití kamery Pixy 2. Ideální umístění pro nejlepší detekční výsledky je 29 cm od snímaného povrchu. Bohužel se k tomuto senzoru díky složitější dostupnosti podařilo dostat až těsně před soutěží, tudíž již nebylo možné k němu naimplementovat řešení, především díky tomu, že by se muselo jednat o úplně jiný přístup k datům a celkové navigaci a také již nebylo na rozšířující desce počítáno s osazením tohoto typu senzoru. Nicméně díky svému potenciálu je použití tohoto řešení navrženo jako další rozšíření této práce v sekci [6.2](#).

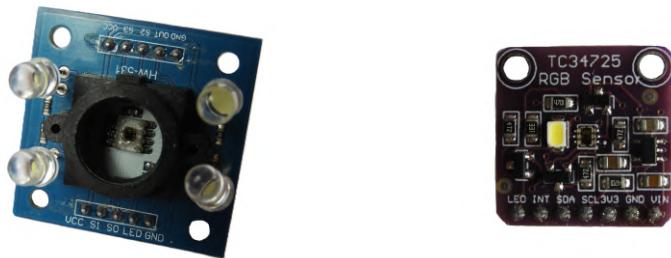
## 4.4 Senzory pro detekci čáry

V této sekci budou popsány senzory, které jsou v této práci využity pro detekci čar. Může jít o hraniční čáry, které vymezují trať, a senzor v tomto případě může nahradit kameru, nebo jako pouhý doplněk pro detekci v případě, kdy kamera nezachytí krajní čáru včas. Dalším možným využitím je detekce vodorovných krátkých čar značících konec/zacátek jednoho měřeného kola. Jedná se o barevné senzory a o optické reflexní senzory, v obou případech jde spíše o senzory doplňkového charakteru a jsou umístěny na platformě v přední části modelu.

### 4.4.1 Senzory na rozpoznávání barvy

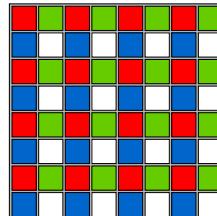
První senzory zaměřující se na rozpoznávání barvy povrchu, na který směřují, jsou senzory barvy. Ve většině případů byla detekována pouze černá a bílá barva, nicméně barevné senzory by mohly posloužit i v případě, kdy by se dráha skládala z jakékoliv jiné barvy. V domácích podmínkách například probíhalo testování funkčnosti a toho, jak mohou barevné senzory detektovat čáru i na povrchu s jinou barvou než bílou.

Byly testovány celkem dva senzory na rozpoznávání barvy, konkrétně **RGB TCS34725** a **TCS3200**. První jmenovaný je menších rozměrů a komunikace s ním probíhá přes rozhraní I<sup>2</sup>C, druhý pak má větší rozměry, ale komunikace s ním probíhá jednostranně – senzor na výstupu poskytuje signál o určité frekvenci a rozpoznávání probíhá na principu toho, jak dlouhá je délka jedné periody. Vzhledem k tomuto principu, který se z praktického hlediska více hodil i s ohledem na funkčnost ostatních senzorů, byl zvolen druhý senzor i přes větší rozměry, které mají na svědomí především 4 bílé LED, které se starají o osvětlení plochy pod senzorem pro přesnější vyhodnocování.



Obrázek 4.9: Senzory barvy, které byly v práci testovány. Zvolen a na modelu osazen byl TCS3200 (vlevo). Uprostřed jsou viditelné fotodiody, které jsou detailněji ukázány dále na obrázku [4.10](#).

Senzor TCS3200 je založen na tom, že se v něm nachází soustava fotodiod, konkrétně 16 pro každou ze základních složek barev (na tyto diody jsou umístěny barevné filtry odpovídající příslušným barvám) a 16 dalších bez filtru. Jejich rozložení je vidět na obrázku 4.10 níže. Tyto diody generují proud, který je převáděn na frekvenční signál, který může být monitorován. V případě bílého podkladu je do fotodiod vysláno více odraženého světla a díky tomu je zvýšena frekvence na výstupu [19].



Obrázek 4.10: Ukázka toho, jak jsou na čipu rozmístěny jednotlivé fotodiody, které převádějí vstupní světelné paprsky na elektrický proud.

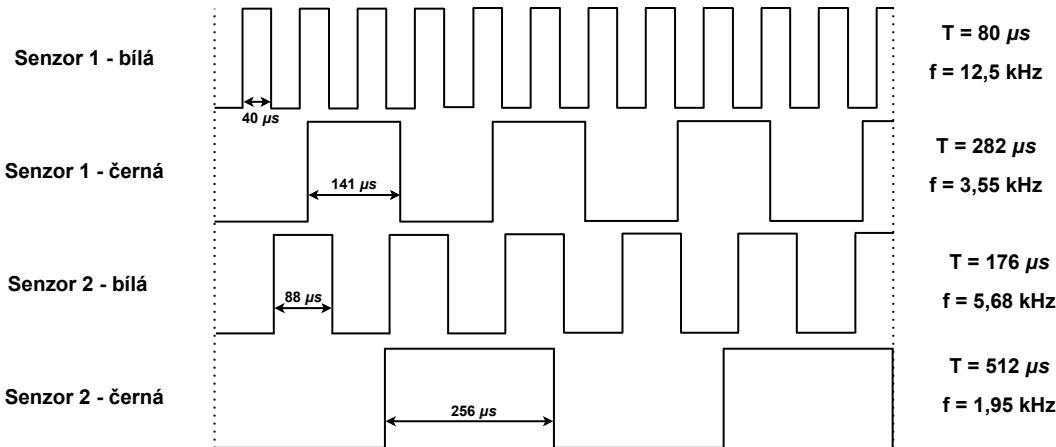
Senzor má čtyři vstupní piny, které se starají o jeho nastavení, ze kterých se dva starají o to, jak je škálována výstupní frekvence a zbylé dva určují, jakých 16 diod je použito ke snímání, konkrétně toto nastavení lze vidět v tabulce 4.4. Díky tomu je možné postupně měřit, jak senzor reaguje na prostředí pro jednotlivé filtry a pak hodnoty zkalibrovat tak, aby při použití nastavení, kdy není použit žádný filtr, byly rozeznávány všechny barvy v RGB spektru. Konkrétně v případě této práce nebylo nutné provádět složité kalibrace, protože na rozeznávání černé a bílé barvy, které je stěžejní, bylo možné i bez nich.

<b>S0</b>	<b>S1</b>	<b>Škálování frekvence</b>	<b>S2</b>	<b>S3</b>	<b>Využité filtry</b>
0	0	Není generován výstup	0	0	Červený
0	1	2 %	0	1	Zelený
1	0	20 %	1	0	Žádný
1	1	100 %	1	1	Modrý

Tabulka 4.4: Tabulka ukazující, jak pomocí vstupních pinů je možné nastavit, v jakém režimu bude barevný senzor TCS3200 pracovat.

Senzory jsou umístěny na přední platformě, kde spolu v kombinaci s infračervenými senzory mohou být umístěny téměř kdekoli na jejím okraji, aby kabely, které k senzorům vedou z rozšiřující desky, nijak nemohly bránit ve výhledu senzorům vzdálenosti, které směřují před model. Detekce může být díky vysoké frekvenci téměř okamžitá, jedno měření trvá v rozmezí 40–256  $\mu$ s.

Je možné využívat dva senzory, pro které jsou připraveny rozšiřující desce sedmipinové konektory. Zajímavé je, že se poměrně razantně liší, jakou frekvenci při totožném nastavení vstupních pinů senzory generují (viz obrázek 4.11 níže). Poměry mezi barvami jsou u obou téměř totožné, nicméně délka trvání jedné periody je u jednoho skoro dvojnásobná, tudíž v implementaci je třeba pracovat s prahovou hodnotou pro každý senzor individuálně. Střída signálu je vždy na 50 %. Hodnoty byly měřeny na domácí testovací dráze, tudíž neodpovídají přesně tomu, jaké výsledky by byly změřeny na soutěžní dráze. Kromě povrchu mají na zaznamenané hodnoty vliv i světelné podmínky.



Obrázek 4.11: Ukázka toho, jaké jsou generovány frekvence na dvou stejných senzorech při totožném nastavení pomocí pinů tak, že je frekvence škálována na 20 %.

#### 4.4.2 Infračervné senzory na sledování čáry TCRT5000

Druhé senzory, které mohou sloužit k detekci čáry, jsou infračervené senzory, konkrétně byla využívána sériově vyráběná deska, na které jsou osazeny tři jednotlivé senzory **TCRT5000** (vidět ji lze na obrázku 4.12). Tyto desky mohou také být připevněny na přední platformu, která je pro ně na míru připravena.

Princip TCRT5000 spočívá ve vysílání infračerveného záření a přijímání jeho odrazu pomocí fototranzistoru. Tyto součástky jsou oddělené plastovou přepážkou zaručující, že se nebudou navzájem rušit a ovlivňovat tím měření. Princip je podobný jako u infračerveného senzoru měření vzdálenosti, ten bude detailněji popsán později v části 4.5.1.

V tomto případě těchto senzorů však není detekována překážka, ale černá čára na světějším pozadí. Pokud je tedy senzor umístěn přímo nad povrchem dráhy, může být okamžitě čára rozpoznána. Pro správnou detekci senzor musí být v rozsahu 2–15 mm od pozorovaného povrchu. Na rovném povrchu senzory funguje velmi dobře, při nerovném (například na soutěžní dráze skládajících se z jednotlivých segmentů) se však můžou vyskytovat problémy.

Výhodou z hlediska implementace je to, že je na desce také osazen komparátor **74HC14** se Schmittovými klopnými obvody, které na vstupu mají analogovou hodnotu z jednotlivých senzorů a na výstup pak poskytnou hodnotu digitální, která je však invertovaná (pro vyšší napětí 0, pro nižší, což odpovídá detekci čáry pod senzory, naopak digitální 1). Díky tomu, že ze senzorů získáváme digitální hodnoty na digitální piny mikrokontroléru, můžeme okamžitě v obsluze přerušení zachytávat změny, pokud je detekována nástupná či sestupná hrana signálu, která nám značí změnu na dráze.



Obrázek 4.12: Ukázka sériové vyráběné desky, na které jsou osazeny tři infračervené senzory TCRT5000 a může být využívána pro detekci čáry na dráze.

## 4.5 Senzory pro detekci překážky

Tato sekce slouží pro představení jednotlivých typů senzorů, které byly vyzkoušeny a testovány za účelem toho, aby monitorovaly, zda se v prostoru před modelem vyskytuje překážka. V soutěžním prostředí se na trati může vyskytovat pouze jeden typ překážky (pokud model razantně neopustí trať), před kterým je třeba zastavit ve vzdálenosti do 10 cm od překážky, nicméně v prostředí, kde byl model testován, se vyskytovalo velké množství překážek v podobě nábytku či stěn. Ty bylo nutné detektovat, aby bylo zajištěno, že model a s ním i senzory nebudou poškozeny v případě prudších nárazů do těchto překážek.

Byly vybrány takové typy senzorů, které jsou založeny na rozdílných měřících technologiích, byla otestována jejich funkčnost a na základě výsledků byly vybrány ty, které budou popsány z hlediska principu a funkčnosti postupně v této kapitole a detailněji pak z pohledu implementace a vyhodnocování při běhu programu pak v kapitole 5.

### 4.5.1 Infračervený senzor měření vzdálenosti

Senzory, založené na infračerveném (v práci také používána pouze zkratka IR<sup>7</sup> záření), jsou v principu poměrně jednoduché – skládají se z dvou hlavních komponent, a to LED, která vyzařuje infračervené záření ve specifické vlnové délce, v případě konkrétního testovaného senzoru se jedná o 870 nm. Toto záření v případě, že narazí na překážku, se na jejím povrchu rozptylí a dojde k odražení do různých směrů. Část tohoto vyzářeného světla se vrátí zpět směrem k senzoru. Zde se pak vyskytuje druhá stěžejní součástka, a to je **fototranzistor**. V závislosti na tom, v jaké intenzitě dopadá světlo na fototranzistor, se mění hodnota v přechodu emitor-kolektor. Znamená to tedy, že výsledný proud (či napětí) je přímo řízen dopadajícím světlem, což je rozdíl oproti normálnímu tranzistoru, kde je výsledný proud řízen proudem báze-emitor.

Výhodou těchto senzorů je především to, že pracují na velmi jednoduchém principu a mají poměrně malou velikost. Jsou přesné a mají velmi nízkou výrobní cenu a také spotřebu elektrické energie.

Nevýhodou pak je možnost rušení, kdy senzor reaguje na jiné zdroje světla. To se eliminuje pomocí infračervených filtrů, fotorezistor se často vyskytuje za zatmaveným sklíčkem. Jedná se o senzory, které se používají spíše na měření kratší vzdálenost (přibližně do jednoho metru) a mohou být také citlivé na teplotu, ve které pracují. Výsledky měření jsou také závislé na materiálu a barvě překážky (na tom závisí odrazivost a to, kolik světla se dostane zpět k fotorezistoru), ke které je vzdálenost měřena [6].

Konkrétně byl testován a jako alternativní použit senzor **GP2Y0A41SK0F**. Jeho hlavní výhodou je jednoduchost, má totiž pouze napájecí piny a jeden výstupní analogový, na kterém je snímána hodnota napětí. Ta po převedení do digitální podoby pomocí AD převodníku může být dále vyhodnocována.

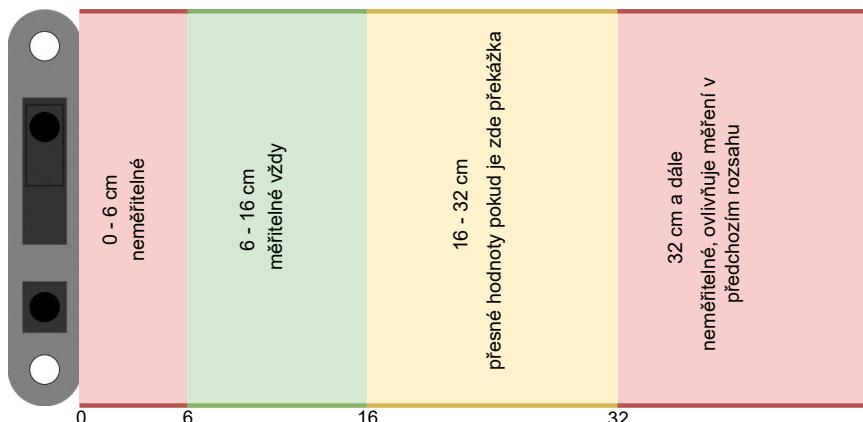
Má výrobcem udávaný rozsah měřící vzdálenosti 4–30 cm, což by byl dostatečný rozsah pro případnou detekci překážky, především v případě, kdy již je překážka očekávána a tomu upravena rychlosť s cílem, aby bylo zastaveno co nejblíže. Při testování se bohužel nepodařilo dosáhnout hodnot v tomto rozsahu. Minimální hodnota, která může být detekována, je okolo 6 cm, což odpovídá hodnotě 4095 při použití 10-ti bitového AD převodníku a tudíž napětí přibližně v hodnotě 3,3 V. Se zvyšující se vzdáleností pak toto napětí a tedy i hodnota na převodníku klesá.

<sup>7</sup>IR – InfraRed (infračervené)

Zajímavý fakt je ten, že ačkoliv je senzor napájen pomocí 5 V a na analogovém výstupu by měly být hodnoty směřující právě k této hodnotě, při měření a testování nikdy nebyla zaznamenána takto vysoká hodnota a maximum bylo právě dříve zmíněných 3,3 V, a proto nebylo třeba toto napětí nijak regulovat vzhledem k napěťové logice, se kterou pracuje mikrokontrolér. Na rozšiřující desce však je možnost osadit odpory a vytvořit tak odporový dělič pro případ, kdy by byl senzor nahrazen jiným analogovým senzorem vzdálenosti, který by na svém výstupu měl vyšší napětí.

Problematičtější však je horní hranice spolehlivé měřitelnosti. Ta byla otestována a stanovena pouze na hodnotě 16 cm (převedená analogová hodnota pak odpovídá hodnotě 2300). Když se vyskytuje překážka do 30 cm, může být tato překážka přesně změřena. Problém však nastává v případě, kdy v tomto rozsahu žádná překážka není, protože senzor pak na výstupu má hodnoty odpovídající tomu, jakoby překážku detekoval a hodnoty se pohybují v rozsahu 16-30 cm. Spolehlivě tak tedy dokážeme detektovat pouze rozsah 6-16 cm. Názorně tento rozsah popisuje obrázek 4.4.

Hodnota 2300 je tedy určena jako hraniční a když dojde k jejímu překročení, s jistotou víme, že se před překážkou vyskytuje překážka. Tohoto může být využito pro finální zastavení před překážkou v takové situaci, kdy již o ní máme informaci z jiných senzorů a díky tomu se model pohybuje malou rychlostí. Mohl by být využit totožný senzor s vyšším rozsahem (do 80 cm, nicméně zde by již bylo pravděpodobně třeba pracovat s úpravou napětí na výstupu senzoru). Ale díky tomu, že se osvědčily ultrazvukové senzory popsáne později, již nebylo potřeba hledat jiný senzor.



Obrázek 4.13: Názorná ukázka toho, v jakých úrovních rozsahu je možné pomocí senzoru vzdálenosti GP2Y0A41SK0F detektovat překážky.

#### 4.5.2 Laserový senzor měření vzdálenosti

Laserový senzor **VL53L0X** je další senzor, který byl testován a zvažován k využití a osazení pro detekci překážek. Je založen na tom, že je vyslán laserový paprsek (jedná se o lidskému oku neviditelný paprsek o vlnové délce 940 nm) a ten je po odrazu od překážky navrácen zpět do senzoru. Z doby mezi vysláním a zpětnou detekcí tohoto paprsku je vypočítána vzdálenost. Jeho výhodou jsou malé rozmezí, menší možnosti rušení tohoto senzoru a hlavně velmi přesné měření. Oproti jiným senzorům je toto měření poměrně dlouhé – při vysoké

přesnosti a při měření v rozsahu do dvou metrů trvá 30–200 ms, což je doba, která není zanedbatelná [14].

Senzor se osvědčil při měření dobře, problém je zde stejný jako u IR senzoru zmíněného dříve – senzor vidí jen před sebe, což není problematické v případě detekce velkých překážek nebo stěn, ale v případě, kde je třeba detektovat i menší překážku, jako je tomu v případě soutěže NXP Cup, to není zrovna praktické. Řešení by bylo osazení více senzorů, aby byla detekční plocha větší, ale později zmíněné ultrazvukové senzory tuto práci zvládly mnohem lépe v jednodušší formě. Kvůli připojení přes I<sup>2</sup>C bylo poměrně složité se senzorem komunikovat a proto tento senzor ve výsledku na modelu nebyl osazen.

Senzor by však perfektně posloužil v případě, pokud by bylo stěžejním úkolem aplikace přesné měření vzdálenosti v prostředí, které se nijak rychle nemění a není riziko, že v případě, kdy senzor překážku nezaznamená, dojde k jeho nárazu a potenciálnímu zničení.

Dalším možným řešením na podobném principu by bylo použití LiDARu<sup>8</sup>, který za pomocí světelných paprsků dokáže přesně monitorovat prostředí před zařízením, nicméně zde se již jedná o poměrně náročné a drahé zařízení, které je pro tento typ projektu zbytečně využít, když potřebnou práci zvládnou mnohonásobně levnější a principiálně jednodušší senzory.

#### 4.5.3 Ultrazvukové senzory měření vzdálenosti

Dalším typem senzorů, které jsou hojně využívány pro detekci překážek či pro detailní měření vzdálenosti, jsou ultrazvukové senzory. Ty jsou založeny na šíření ultrazvukových pulzů, které jsou ze zařízení vyslány směrem k očekávané překážce. Konkrétně se jedná o vyslání periodických pulzů na vysoké frekvenci, v případě obou typů testovaných senzorů se jedná o frekvenci 40 kHz. To je mimo rozsah, který může být slyšitelný lidským uchem, jenž se pohybuje na maximální hranici okolo 20 kHz [22].

Pokud pulz narazí na překážku, dojde k jeho odražení a je zachycen na senzoru a podle délky je určena vzdálenost. Výhodou těchto typů senzorů je to, že je neomezuje světelné podmínky, jejich jednoduchý princip měření a to, že dokážou detektovat i překážku, která se nevyskytuje pouze přímo před senzorem, ale mají jistý detekční úhel, ve kterém ji dokážou zaznamenat. Problémem však naopak můžou být různé materiály, které pohltí vyslaný signál, případně pokud je povrch překážky nerovný. Drobně také měření ovlivňuje teplota a vlhkost, ty mají přímý vliv na rychlosť zvuku, na základě čehož je vzdálenost vypočítávána. Pro tuto práci to však jsou zanedbatelné rozdíly a senzory jsou i bez zohlednění jejich změny velmi přesné.



Obrázek 4.14: Ultrazvukové senzory M5Stack, které byly testovány (vpravo) a použité HY-SRF05 (vlevo).

<sup>8</sup>LiDAR – Light Detection and Ranging (detekce světla a vzdálenosti)

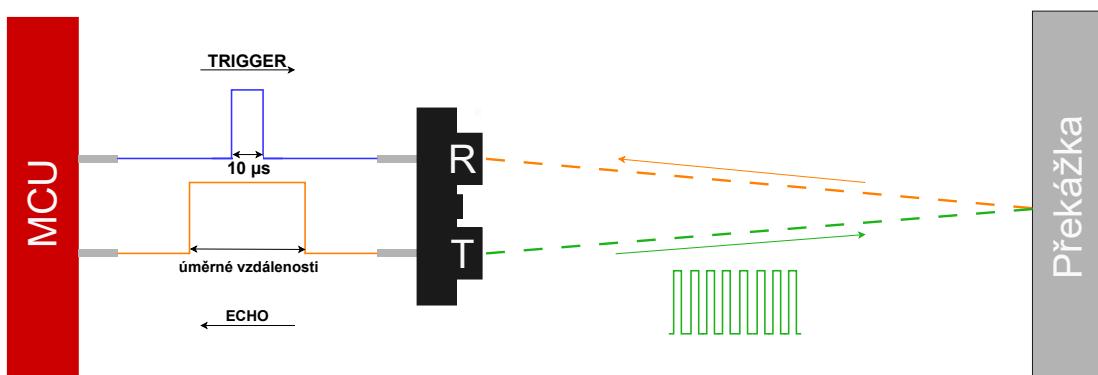
Prvotně byly testovány senzory od **M5Stack**, které se jevily jako ideální z důvodu, že jsou již zasazeny do praktického pouzdra a využívají komunikační sběrnice I<sup>2</sup>C, tudíž by mohly být umístěny dva na jednu sběrnici. Tak by byly ušetřeny konektory na vývojové desce oproti jiným senzorům.

Poskytovaly by dostačující měřící rozsah 2–150 cm, který je uveden na samotném senzoru, nicméně v praxi se dolní hranice značně liší a měřitelná vzdálenost začíná až na 30 cm, což ovšem po dohledání udává i sám výrobce na svém webu a nahradil tyto senzory novějším typem. To byl hlavní důvod, proč nakonec byly použity jiné senzory, které poskytovaly spolehlivější rozsah měření.

Nejvíce ze všech senzorů vzdálenosti se osvědčily senzory **HY-SRF05**. Praktický je jejich měřící rozsah od 2 cm až do vzdálenosti 450 cm, v reálném prostředí spolehlivě fungovaly do 200 cm bez jakýchkoliv problémů, takže pro potřeby soutěže, kde je třeba detektovat překážku ve vzdálenosti řádově desítek centimetrů, jsou ideálním řešením.

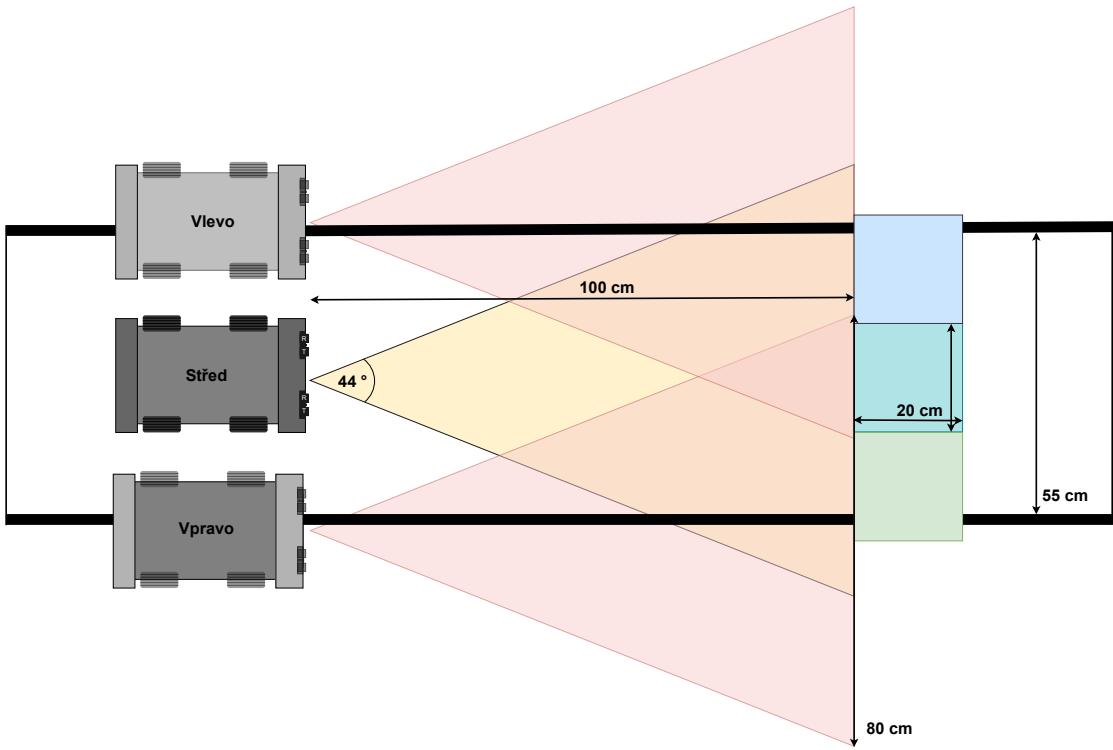
Ke komunikaci se senzorem slouží pouze 2 datové vodiče, kdy jeden slouží k tomu, že je senzor krátkým 10 µs dlouhým pulzem pobídnut ke startu měření a vyslání periodických pulzů. Na dalším pinu je pak následně měřeno, jak dlouho je signál ve stavu logické 1 a na základě délky tohoto pulzu je pak vypočítána vzdálenost. Celý tento měřící a komunikační proces je znázorněn na obrázku 4.15. Díky tomuto jednoduchému principu je tato komunikace praktičtější než využívání I<sup>2</sup>C v případě dříve popsaných senzorů.

Senzory byly usazeny do přípravku vytisknutého na 3D tiskárně, který je drží ve fixní pozici tak, že směřují vodorovně přímo před model. Byly využity dva kusy senzorů, které byly osazeny na vytvořenou přední platformu. Bylo by možné osadit i senzory tří, kde jeden by byl přímo uprostřed a dva na krajních pozicích, ale i při využití právě dvou bylo dosaženo velmi dobrého rozsahu měření, kdy překážka na závodní trati může být detekována v kterékoli pozici (ať už pozici modelu na dráze či překážky). Tyto situace jsou názorně zobrazeny a představeny na obrázku 4.16.



1. Zaslání 10 µs dlouhého pulzu na TRIGGER pin senzoru
2. Senzor vyše 8 40 kHz ultrazvukových pulzů pomocí reproduktoru
3. Signál se odrazí a míří zpět k mikrofonu senzoru, je přenesen na ECHO pin kde je zaznamenána jeho délka

Obrázek 4.15: Ukázka toho, jak probíhá komunikace mezi MCU a ultrazvukovým senzorem HY-SRF05 při měření vzdálenosti od překážky.



Obrázek 4.16: Ukázka toho, jak pomocí dvou ultrazvukových senzorů je možné detektovat překážku ve vzdálenosti 1 metr kdekoliv na dráze.

## 4.6 Magnetické senzory

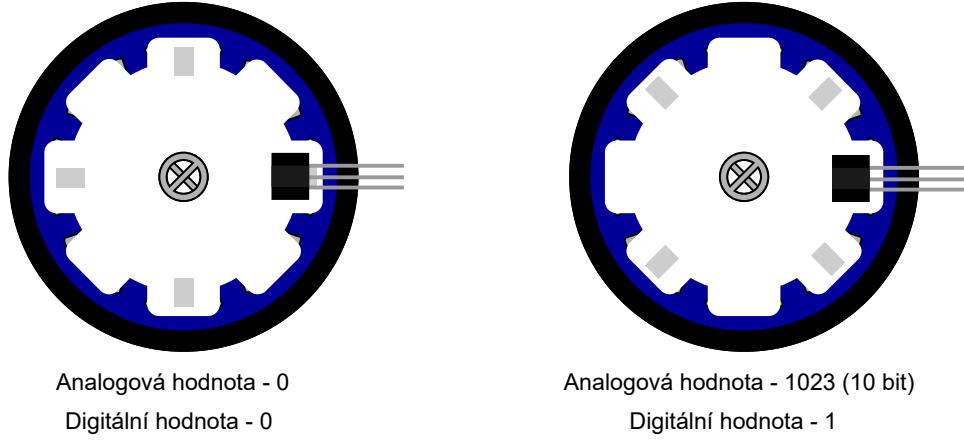
Poslední typ senzorů, který byl využit pro získávání užitečných informací o pohybu modelu, konkrétně o počtu otáček kol a tudíž o přesně ujeté vzdálenosti či aktuální rychlosti, jsou magnetické senzory, přesněji pak Hallův senzor.

Tento senzor je založen na Hallově jevu, který spojuje magnetické a elektrické pole. Protéká-li Hallovým elementem (polovodičovou destičkou) konstantní proud v magnetickém poli (které je na ni kolmé), tak v závislosti na velikosti indukce tohoto magnetického pole je pak generováno rozdílné napětí na bočních stranách elementu, které je přímo úměrné velikosti indukce magnetického pole [21].

Bylo využito modulu s Hallovým senzorem 49E a zesilovačem LM393. Ten sice má větší rozměry, než kdyby bylo využito samotného Hallova senzoru, nicméně jeho praktickou výhodou je to, že díky osazenému zesilovači neposkytuje pouze analogový výstup značící sílu naměřeného magnetického pole, ale také digitální vstup, který se sepne na opačnou logickou úroveň napětí v případě, že naměřená analogová hodnota překročí určitou mez. To má vzhledem k využití při běhu programu výhodu tu, že není potřeba využít analogového vstupu a převodníků, ale stačí využít na mikrokontroléru digitálních pinů a vyvolávat přerušení, konkrétně v tomto případě při sestupné hraně, jež značí, že se v dosahu senzoru vyskytl magnet.

Právě magnety byly přilepeny na vnitřní stranu ozubeného kola, které je stejně jako poháněné kolo připojeno k motoru pomocí hřídele. Samotný modul pak byl ve vhodné poloze umístěn v zadní části na vytisknuté destičce tak, aby vystupující Hallova sonda byla

v co nejbližší vzdálenosti od otáčejícího se kola, jak je znázorněno na obrázku 4.17. Pro každé kolo byl osazen jeden senzor a čtyři miniaturní magnety (velikost 2 x 3 mm), díky čemuž bylo dosaženo uspokojivé přesnosti měření (jedna otočka kola značí ujetou vzdálenost 22 cm, tudíž by přesnost měla mít v ideálním případě odchylku maximálně 5,5 cm).



Obrázek 4.17: Ukázka toho, jak jsou přibližně umístěny magnety na poháněném kole a jaké jsou při tom snímány hodnoty pomocí Hallova senzoru.

## 4.7 Ukládání a přenos dat z modelu

Vzhledem k množství informací, které jsou získávány z okolí modelu, bylo třeba navrhnout praktický způsob, jak budou tato data ukládána pro následné zpracování a vyhodnocení.

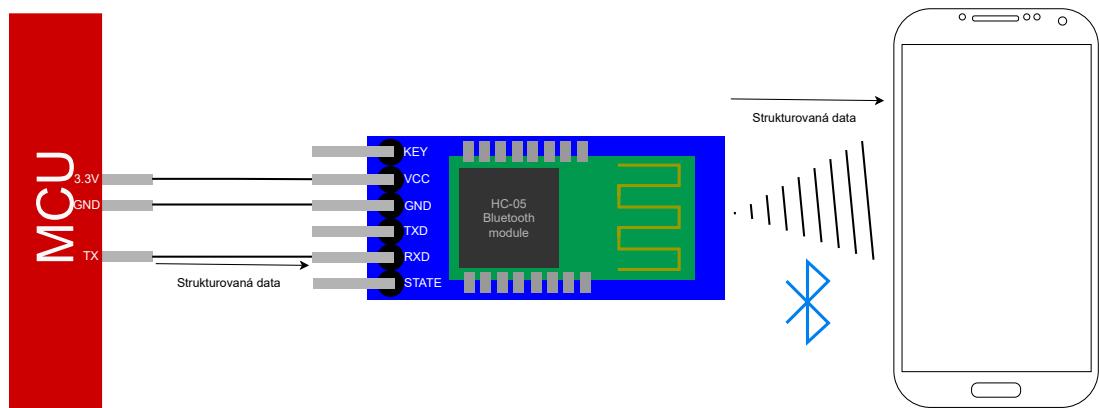
Jako první a logické se jeví využití SD karty, která by mohla být jednoduše připojena přes rozhraní SPI k modelu, ať už v podobě externího modulu, případně by pak mohl být konektor pro ni osazen na vytvořenou rozšiřující desku. Ve vývojovém prostředí bylo docela problematické komunikaci s kartou zprovoznit, ačkoliv byly dostupné drivery, se kterými bylo možné pracovat, nedářilo se je přiřadit správně do výsledného projektu v MCUExpressu. Za větší problém jsem však považoval fakt, že k datům bude přístup až zpětně, když bude model zastaven a karta bude vyjmuta. Díky zkušenosti z vlastní bakalářské práce, v níž bylo ukládání dat realizováno tímto způsobem, došlo ke zvolení jiné alternativy, při které budou zaznamenávaná data moci být sledována okamžitě.

Jedná se o využití Bluetooth modulu **HC-05**, který je pomocí rozhraní UART připojen k vývojové desce a data, která mu přes rozhraní přijde, přeposílá pomocí Bluetooth k připojenému zařízení (znázorněno na obrázku 4.18). To tyto hodnoty okamžitě pomocí sériového terminálu může zobrazit a tento výpis samozřejmě také uložit. Při využití přenosové rychlosti 115200 znaků/s může být posíláno naprostě dostatečné množství informací. Modul slouží pouze pro zasílání dat z modelu do externího zařízení za účelem sledování a ukládání těchto dat.

Nejčastěji bylo využíváno připojení přes telefon s operačním systémem Android, především při testování na dráze to bylo praktické v případech, kdy bylo možné přímo sledovat zaznamenávané hodnoty vektorů z kamery Pixy 2 či aktuální hodnoty ze senzorů. Data byla posílána buď v přehledné podobě s doprovodnými textovými informacemi v případě

testování, nebo pak při pokročilejších pokusech již bylo využíváno pouze strukturovaných dat, která je pak následně možné zpracovávat.

Jedinou nevýhodou tohoto řešení je to, že nemůže být použito během soutěže – poruší pravidla o tom, že se nejedná o mikrokontrolér od NXP a že se jedná o bezdrátové zařízení. Díky tomu tudíž není možné zaznamenávat data ze soutěžních pokusů, ale pouze z testovacích, což však ve výsledku není problém a pro praktičnost tohoto řešení to bylo vyhodnoceno jako nejlepší možná volba pro monitoring dat.



Obrázek 4.18: Ukázka připojení Bluetooth modulu HC-05. Data ze senzorů a další informace jsou zpracovány na mikrokontroléru a přes UART poslány do modulu, jenž je pak do cílového zařízení, kde mohou být uložena, přenese pomocí Bluetooth.

## Kapitola 5

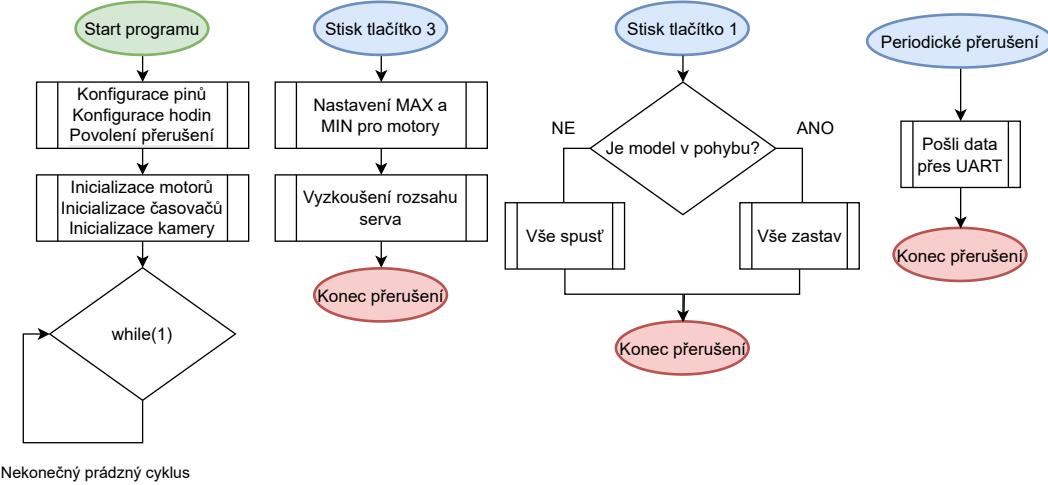
# Řešení a implementace

V této kapitole bude popsána postupně po jednotlivých částech softwarová implementace. Celý běh programu je rozdělen do mnoha souborů a logických funkčních segmentů, spolu provázaných dle potřeby. Stejně tak bude do jednotlivých částí rozdělena tato kapitola, kdy se každá její sekce se bude zabývat popisem implementace dané součásti projektu. Zjednodušeně jsou jednotlivé části popsány ve vývojovém diagramu, který je celý v příloze **B** a jednotlivé části s vysvětlením jsou na obrázcích **5.1**, **5.2** a **5.3** dále.

Všechny popsané kódy, ať už samotný projekt spustitelný v prostředí MCUXpresso, kde probíhala veškerá implementace této práce, či skripty, které byly využity pro generování výstupů, jsou přiloženy v příloze **C**, stejně jako ukázka zaznamenaných dat. Každá sekce je na jejím konci vždy doplněna tabulkou, která popisuje, kam jsou připojeny dané součástky jak na rozšiřující desce, tak na samotném mikrokontroléru, kde je uveden jak modul, tak samotný pin. K implementaci byl zvolen jazyk C, který je pro tyto typy projektů nejpoužívanější. Samotné vývojové prostředí generuje soubory, které jsou automaticky upravovány při každé změně v konfiguraci pinů a jsou využívány pro základní funkcionalitu všech součástí (práce s čítači, přerušeními, periferiemi či moduly). Prostředí také poskytuje vzorové projekty, na základě kterých je možné pochopit, jak s jednotlivými částmi pracovat a postupovat při jejich implementaci.

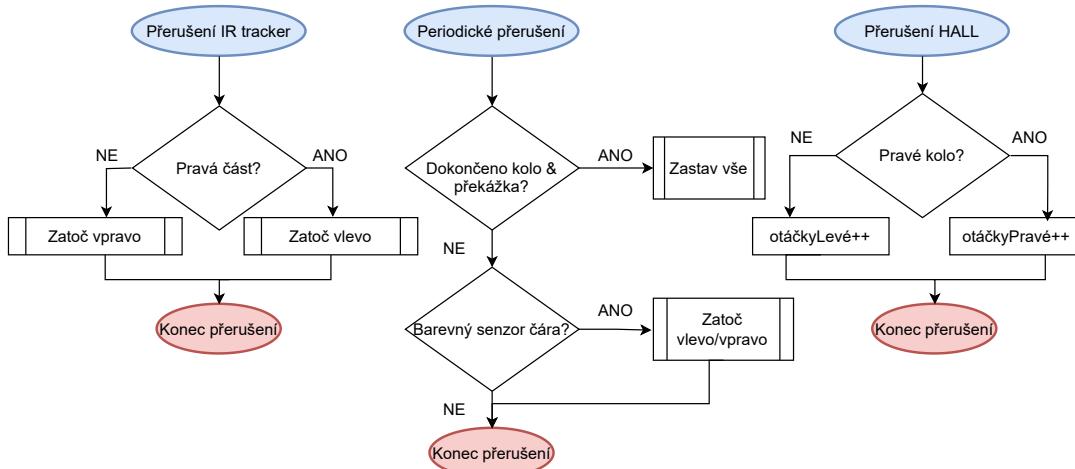
Také jsou příloze **C** přiložena videa ukazující funkcionalitu na oficiální soutěžní dráze, ke kterým jsou vygenerována 1:1 videa zobrazující data ze senzorů. Tato videa byla vytvořena, stejně jako grafy a některé obrázky, pomocí skriptů v jazyce Python.

Základní část programu s vývojovým diagramem na obrázku 5.1 ukazuje inicializační a základní konfigurační funkce, které jsou automaticky spuštěny při startu programu. Stisk tlačítka 1 (SW1) se stará o spuštění či vypnutí (pokud nebylo provedeno samo na základě reakce některého ze senzorů) hlavní funkce modelu – rozjede se a autonomně se snaží pohybovat po dráze. Stisk tlačítka 3 (SW3) slouží k nastavení rozsahu motorů a detailněji je tato část popsána v sekci 5.2. Periodické přerušení pak je vyvoláno za účelem zasílání logovacích dat přes rozhraní UART a zmiňuje ho sekce 5.7.



Obrázek 5.1: Vývojový diagram základní části programu s inicializací, obsluhou tlačítek a odesíláním logovacích dat.

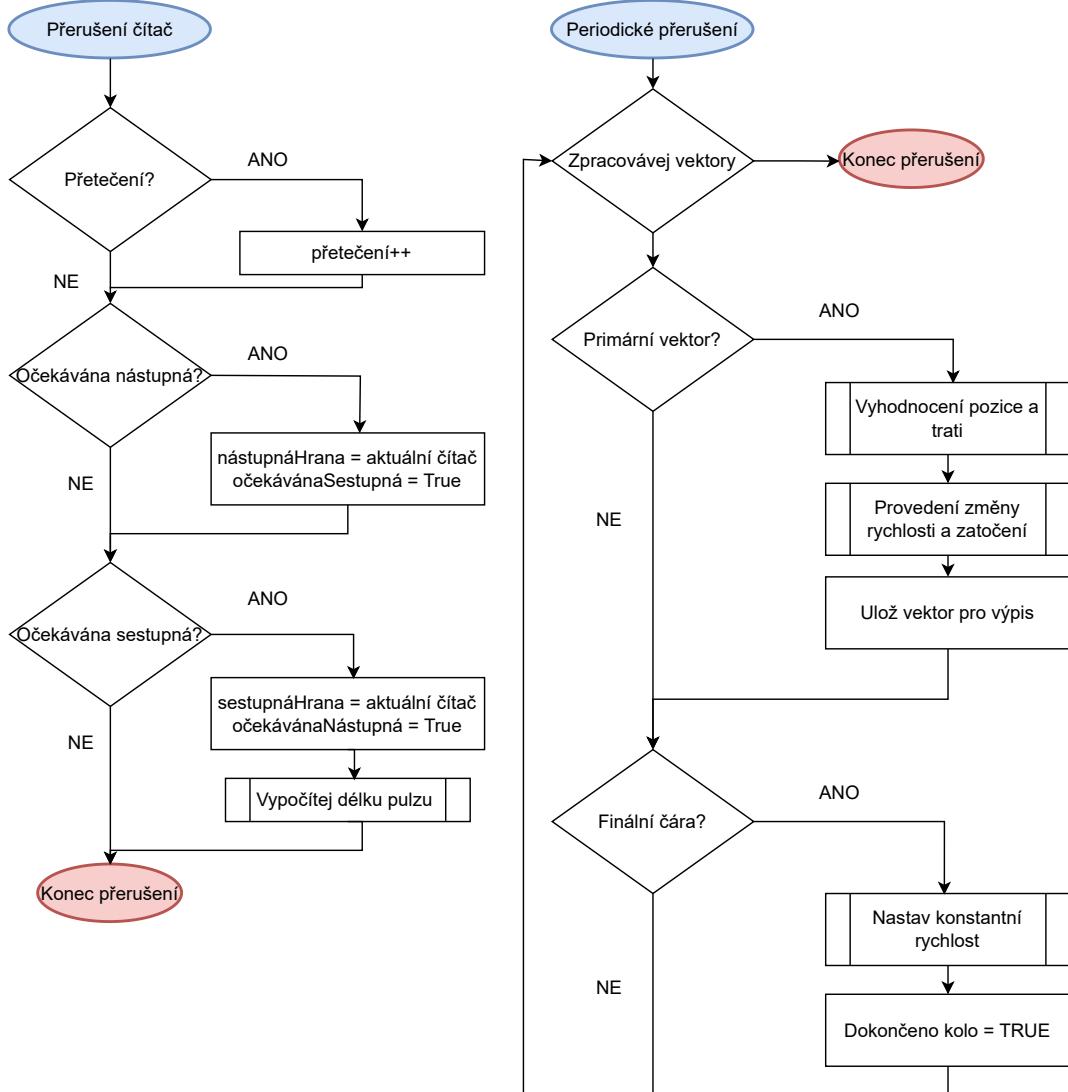
V další části je znázorněna obsluha přerušení, které vyvolá IR senzor a vede k zatočení podle umístění senzoru. Přerušení vyvolané nástupní hranou Hallová senzoru zařídí inkrementaci počtu otáček pro příslušné kolo (viz sekce 5.3). Periodické přerušení pak kontroluje přítomnost překážky a zda barevný senzor nezachytí černou čáru.



Obrázek 5.2: Část vývojového diagramu se zpracováváním přerušení z IR senzorů, Hallových senzorů a obsluhou periodického přerušení pro další senzory.

Poslední část vývojového diagramu ukazuje dvě stěžejní funkce z hlediska implementace. Je zde část algoritmu, která se stará o monitorování délky pulzů na vstupních pinech mikrokontroléru. Jedná se o zjednodušenou variantu, ve skutečnosti vše ukázané probíhá pro každý jeden barevný a ultrazvukový senzor zvlášť, celkem tedy čtyřikrát. Cílem je změřit délku pulzu za účelem dalšího zpracování, které je pak detailněji popsáno v sekci 5.5.

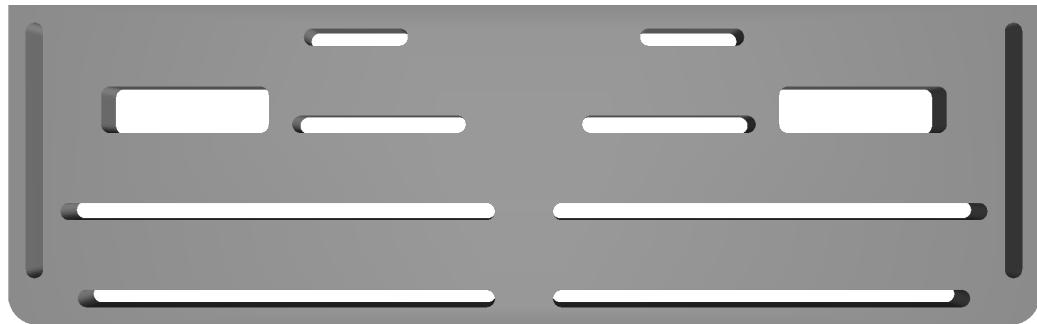
V části s periodickým přerušením je pak možné vidět hlavní část navigačního algoritmu, při kterém jsou postupně zpracovávány všechny vektory, které zachytí kamera. Detailněji je tento celý navigační algoritmus popsán v sekci 5.6.



Obrázek 5.3: Ukázka implementace zachytávání hran pro barevné a ultrazvukové senzory (vlevo) a zjednodušený navigační algoritmus obsahující celkové zpracovávání vektorů pomocí kamery Pixy 2.

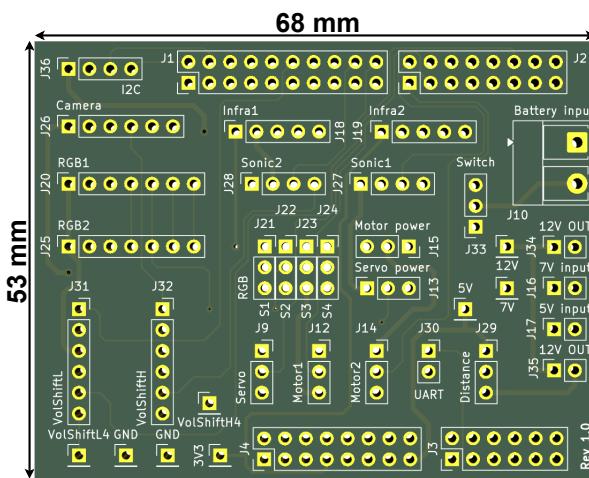
## 5.1 Rozšiřující deska a platforma pro uchycení senzorů

Během práce se senzory bylo třeba vymyslet mechanismus, jak je uchytit na přední stranu modelu. Z tohoto důvodu byla podle rozměrů navržena a na 3D tiskárně vytisknuta platforma (návrh lze vidět na obrázku 5.4). Tu je možné pomocí šroubů uchytit k modelu a na ní jsou umístěny všechny senzory, které je možné možné přesouvat na desce dle potřeby.



Obrázek 5.4: 3D návrh přední platformy, na kterou jsou uchyceny potřebné senzory.

Pro kompaktnější řešení byla navržena rozšiřující deska (i s rozměry je vidět na obrázku 5.5), která poskytla spolehlivější a praktičtější řešení, především z hlediska propojení jednotlivých součástek a napájení. Deska je navržena tak, aby bylo možné ji pohodlně nacvaknout na vývojovou desku FRDM KL27Z, nicméně i přes tyto rozměry na ni mohou být pohodlně osazeny všechny potřebné konektory pro tento projekt.



Obrázek 5.5: Návrh rozšiřující desky s rozměry. Detailněji je možné ji vidět v příloze A.

Nejvíce místa zabírají dvouřadé pinové kolíkové lišty (J1, J2, J3 a J4) sloužící k propojení a nacvaknutí s původní deskou. Podobně se zde vyskytují šestipinové lišty (J31 a J32), do kterých je zacvaknut převodník úrovní napětí. Na pravé straně jsou dvojice pinů, do kterých jsou napájeny výstupy regulátorů napětí a různé úrovně napětí jsou pak na desce rozvedeny k ostatním konektorům.

Velký 2,54 mm konektor na pozici J10 slouží k připojení napájení a jsou do něj přes svorky přivedeny kabely z baterie. Aby nebylo napětí na desce stále, je zde osazen posuvný přepínač na pozici J33, díky kterému může být regulováno, zda je deska napájena z baterie či nikoli.

Trojice pinů (J21-J24) ve středu desky slouží k nastavování rozsahů a filtrů barevných senzorů pomocí zkratovacích propojek, stejně tak je možné pomocí těchto propojek regulovat, jaké je využito napětí pro servomotory (J13 přepíná mezi 5 V a 7 V) a na motorech (J15 mezi 7 V a 11 V).

Další konektory jsou již k přímému připojení senzorů, motorů či dalších zařízení, například Pixy 2 na J26 či HC-05 na pozici J30. Všechny osazené konektory jsou se zámečkem, není tedy možné zapojit konektory opačně, což je praktické řešení především z hlediska jednoduššího používání. Je zde vyveden navíc jeden I<sup>2</sup>C konektor pro připojení dalších zařízení, v práci však tyto piny byly využity jako digitální pro připojení dvou magnetických Hallových senzorů.

## 5.2 Ovládání pohybu modelu

Ovládání pohybu modelu, tedy serva pro natáčení přední nápravy a dvou motorů pro pohánění každého kola zvlášť na zadní nápravě, je založeno na práci s **TPM (Timer/PWM Module)** a je implementováno v souboru *drive\_control.c*, odkud jsou pak jeho funkce volány dle potřeby.

Motory jsou připojeny na TPM2 modul, konkrétně pak každý na jeden z jeho dvou kanálů, stejně tak je připojeno servo na jeden kanál modulu TPM1 (oba moduly jsou pouze dvoukanálové). V inicializační funkci *MotorsInit* dojde ke konfiguraci tak, aby moduly pracovaly v módu pulzně šířkové modulace – nastavena je frekvence 50 Hz, která je striktně daná pro ovládání motorů i servomotorů. Kam přesně jsou motory i servomotor připojeny, shrnuje tabulka 5.1.

Důležitá je funkce *MotorSetRange*, která musí být volána v případě, kdy je třeba zkalirovat a nastavit rozsah pro motory. Je vyvolána stiskem tlačítka SW3 přímo na vývojové desce. Dojde v ní k nastavení horní hranice střídy, což je 9,9 % na dobu 1 sekundy, a pak na stejný časový úsek k nastavení dolní hranice, která je okolo 6 %. Tato dolní hodnota je však jiná pro každý motor z důvodu, aby bylo dosaženo stejných otáček na obou poháněných kolech. Funkci je třeba volat v případě, kdy motory po připojení napájení pípají, čímž dávají najevo, že čekají na kalibraci – většinou je potřeba ji provést, když došlo k odpojení baterie. Po kalibraci přestanou motory vydávat zvukový signál a již je lze ovládat pomocí nastavování střídy v nastaveném rozsahu.

Servo žádnou kalibraci nepotřebuje, nicméně je zde funkce *ServoCheckRange*, která postupně projde celý rozsah serva pro kontrolu, zda se někde nevyskytuje problém a model tedy bude moci pohodlně zatáčet. Funkce je volána také při stisku tlačítka SW3. Střídu má upravenou tak, aby maximální hodnota znamenala co největší úroveň zatáčení, ale zároveň nijak nemohla poškodit servo v případě, že by se snažilo dostat do pozice, kterou ramena s koly už konstrukčně nedovolí.

Funkce, která nastavuje rychlost, je *MotorSetSpeed* a je volána vždy podle potřeby z jiných částí programu. Zadaná hodnota v % se přepočte na hodnotu střídy pro každý motor dle potřeby a dojde k aktualizaci generovaného signálu pomocí PWM. Pro zatáčení je zde pak funkce *SteerStraight*, která bez argumentu nastaví kola do pozice pro jízdu vpřed. *SteerLeft* a *SteerRight* pak slouží pro zatáčení, zadaná hodnota argumentu v procentech (0 % značí jízdu rovně, 100 % maximální zatočení na danou stranu) je přepočtena na procentuální hodnotu střídy. Dochází také k ukládání těchto nastavovaných hodnot do globálních proměnných, aby hodnoty mohly být periodicky odesílány a zpětně analyzovány.

	<b>Modul</b>	<b>Pin KL27Z</b>	<b>Konektor deska</b>	<b>Rozsah střídy</b>
MOTOR 1	TPM2 kanál 0	PTB3	Motor1 (J2)	6,3 % – 9,9 %
MOTOR 2	TPM2 kanál 1	PTB2	Motor2 (J14)	5,9 % – 9,9 %
SERVO	TPM1 kanál 0	PTA12	Servo (J9)	5,6 % – 10,2 %

Tabulka 5.1: Tabulka ukazující rozsahy střídy pro servo a motory s tím, kam jsou na rozšiřovací desce i na mikrokontroléru zapojeny.

### 5.3 Zaznamenávání uražené vzdálenosti

Zaznamenávání ujeté vzdálenosti je založeno na velmi jednoduchém principu a je implementováno v souboru *hall\_sensors.c*. Vstupní piny, na které je připojen výstup ze senzorů (o které se jedná, představuje tabulka 5.2), mají nastaveny přerušení na nástupné hraně. To je pak automaticky zaznamenáno a dojde k inkrementaci počítadla čtvrtotáček pro dané kolo. Každé čtvrté vyvolané přerušení tedy znamená jednu otáčku. Tyto hodnoty jsou zaznamenávány (prvně byly zaznamenávány pouze celé otáčky, později pro větší přesnost každá čtvrtotáčka) při měřeních a pak zpětně byly vyhodnocovány. Díky zaznamenávání těchto hodnot bylo možné provést kalibraci motorů tak, aby jejich procentuální výkon byl vyrovnaný a bylo dosaženo přibližně stejných otáček na každém kole.

V některých případech měření bylo zaznamenáno, že hodnota na vstupu není stálá a dochází k opakování generování nástupné hrany. Toto se děje pravděpodobně díky horším kontaktům, tyto senzory totiž jako jediné nebyly připojeny přímo na rozšiřovací desku, ale byly vedeny ještě přes nepájivé pole. Z tohoto důvodu byla přidána kontrola toho, zda od posledního přerušení uběhl určitý časový úsek (konkrétně 25 ms). Tím je omezena maximální zaznamenávaná rychlosť na  $2,2 \frac{m}{s}$ , nicméně této rychlosťi nebylo dosaženo ani při nejvyšším používaném a testovaném výkonu (maximální zaznamenaná hodnota při pohybu byla okolo  $1,5 \frac{m}{s}$ ). V případě vyšších rychlosťí nebo potřeby měřit výkon motorů na prázdnou jako na obrázku 4.2 představeném dříve je možné tuto hodnotu změnit dle potřeby.

Vyhodnocení správné funkčnosti měření vzdálenosti je vidět například na obrázcích 6.6 a 6.7 v sekci 6.2, kde také jsou shrnutý výsledky dalších měření souvisejících se zaznamenáváním počtu otáček a z toho plynoucích výpočtů veličin popisujících pohyb vozidla na dráze.

	<b>Modul</b>	<b>Pin KL27Z</b>	<b>Konektor deska</b>
Hall senzor 1	GPIO	PTD6	I <sup>2</sup> C (J36) SDA
Hall senzor 2	GPIO	PTD7	I <sup>2</sup> C (J36) SCL

Tabulka 5.2: Představení zapojení Hallových senzorů, které snímají otáčky kola. Byly zapojeny na volný I<sup>2</sup>C konektor, který by byl jinak nevyužit.

## 5.4 Infračervené senzory

V implementaci bylo pracováno se dvěma typy infračervených senzorů – jedním pro měření vzdálenosti a druhým pro detekci černé čáry. Implementace jejich funkčnosti se nachází v souboru *ir\_sensors* a v tabulce 5.3 je popsáno, kam jsou zapojeny.

Senzor vzdálenosti GP2Y0A41SK0F, který byl z teoretického a funkčního hlediska popsán dříve v části 4.5.1, je inicializován v ADC režimu ve funkci *irSensorInit*, kde také dojde k prvnímu vyvolání měření. O jeho dokončení je informováno formou přerušení. Později pak periodicky dochází ke kontrole změřené vzdálenosti a když jsou splněny nastavené podmínky (model se pohybuje a již bylo dokončeno kolo) a hodnota překročí prahovou hodnotu 2300 (přibližně 16 cm, jedná se o nejvyšší hodnotu, která může být bezpečně rozpoznána), dojde k zastavení modelu. Je zde i funkce *irSensorConvert*, která převede zaznamenanou hodnotu z převodníku na centimetry, ale pro vyhodnocování není nutné ji používat.

Infračervené senzory TCRT5000 mají velmi jednoduchou implementaci. Pokud je na jejich výstupu vygenerována nástupná či sestupná hrana, je vyvoláno přerušení. To je obslouženo v souboru *interrupts.c* a zavolá funkci *updateTrackerValues(tracker)*, čímž dojde k uložení do pole se všemi hodnotami. V případě, kdy je senzor používán jako doplněk pro zatáčení, je vyhodnoceno, o jakou stranu modelu se jedná a podle toho dojde k zatočení na druhou stranu.

V případě, že by senzor sloužil k detekování cílové čáry, je zde funkce *getTrackersValuesCount*, která spočítá aktuální počet senzorů detekujících. Když jich je daný počet (pro detekci čáry stačily dvě až tři hodnoty), je tento stav považován za detekování čáry. Jedná se spíše o doplňkovou možnost, kterou by bylo možné využít v případě, kdy by k navigaci sloužily pouze tyto senzory. Kamera Pixy 2 totiž tuto detekci zvládá lépe, především díky tomu, že ji dokáže detektovat mnohem dříve, než model na danou čáru najede.

Z důvodu, že je logika nastavena tak, že jsou detekovány nástupné i sestupné hrany (každá změna hrany vyvolá přerušení) a prvně je očekávána nástupná hrana (značící detekci čáry), je třeba pro správné fungování zajistit, aby při startu běhu programu model již byl umístěn na startovní dráze a mířil na bílou plochu. Pokud by tomu tak nebylo, mohlo by se stát, že bude logika otočena nebo pro každý jednotlivý senzor fungovat jinak.

	Modul	Pin KL27Z	Konektor deska
Deska1 s TCRT5000	GPIO	PTD5, PTD4, PTD3	Infra1 (J18)
Deska2 s TCRT5000	GPIO	PTD1, PTD2, PTD0	Infra2 (J19)
IR Sensor vzdálenosti	ADC0 kanál 14	PTC0	Distance (J29)

Tabulka 5.3: Tabulka přehledně ukazující kam jsou na rozšiřovací desce i na mikrokontroléru zapojeny infračervené senzory.

## 5.5 Barevné a ultrazvukové senzory

Barevné a ultrazvukové senzory jsou k sobě v implementaci umístěny i přes rozdílný typ měřených veličin z důvodu toho, že zaznamenávání jejich výstupu je totožné – je třeba zaznamenat délku pulzu, kterou tyto senzory poskytují. V případě barevných senzorů je zde stálý proud signálu s měnící se frekvencí, která značí, jaká barva je pod senzorem zaznamenávána. U ultrazvukových senzorů je pak poslána pouze jedna hrana, jejíž délka je úměrná vzdálenosti od překážky.

Veškerá práce s těmito senzory je implementována v souboru *sonic\_color.c*. Inicializace zde standardně probíhá ve funkci *TMP0Init*, ve které je provedeno celkové nastavení TMP0 modulu a jednotlivých modulů do módu pro zachytávání hran a jejich délek. Při zaznamenané akci je vyvoláno přerušení a podle statusové flag hodnoty (je v ní zaznamenáno, jaká událost vyvolala přerušení) je rozhodnuto, co bude vykonáno.

Může být detekováno, že došlo k přetečení časovače, který slouží k zaznamenávání časových značek v případě zachycení hran. Časovač je 16bitový, takže k tomuto dojde při hodnotě 65535. Pro každý senzor zvlášť se tedy musí v tomto okamžiku inkrementovat počítadlo přerušení.

V případě, že byla zaznamenána změna požadované hrany na některém z kanálů, je vyhodnoceno, na kterém a pak již je ve všech stejná logika zaznamenávání hodnoty. Je uložena hodnota časovače pro danou hranu a v případě nástupné hrany je vynulován počet přetečení časovače. Modulu na daném kanálu je pak změněno nastavení na zachytávání sestupné hrany. Při její detekci dojde také k uložení času a opět přenastavení zachytávání hrany nástupné. V tomto okamžiku se již také vypočte délka pulzu, logika tohoto výpočtu je znázorněna v pseudokódu 5.1 níže. Hodnoty jsou pak převedeny do  $\mu\text{s}$  a dále je s nimi pracováno již rozdílně podle typu senzorů.

```

1 if (sestupnaHranaHodnotaTimeru >= nastupnaHranaHodnotaTimeru):
2     delkaPulzu = (sestupnaHranaHodnotaTimeru - nastupnaHranaHodnotaTimeru) + pocetPretecení * 65535
3 else:
4     delkaPulzu = (65535 - nastupnaHranaHodnotaTimeru) + sestupnaHranaHodnotaTimeru + (pocetPretecení * 65535)

```

Výpis 5.1: Ukázka principu, na jakém funguje vyhodnocování délky zaznamenaného pulzu.

V případě barevných senzorů se již nemusí s hodnotou v  $\mu\text{s}$  nijak dále pracovat – pouze se zvolí správná hodnota prahu, který odděluje to, kdy ještě senzor snímá bílou barvu a kdy již se jedná o barvu černou. Na základě toho dojde k okamžitému natočení kol směrem od čáry zpět do dráhy. Po jednom dokončeném měřeném pulzu je okamžitě měřen další.

Složitější je situace s ultrazvukovými senzory, u kterých před měřením délky pulzu prvně musí dojít k vyslání pulzu směrem k senzoru. K tomu dochází tak, že je na 10  $\mu\text{s}$  nastaven pin na logickou hodnotu 1, což provádí funkce *TriggerPulse1* a *TriggerPulse2*. Pak již je měření pulzu stejné. Po dokončení měření na jednom senzoru je vyslán startující signál měření na senzor druhý, takže se senzory vždy střídají a měří vždy pouze jeden. Tím je eliminováno to, že by se navzájem ovlivňovaly a rušily.

Pro větší přesnost měření také dochází k aritmetickému průměrování naměřených hodnot (konkrétně se využívalo 5 nebo 10 hodnot) a až pak dochází k vyhodnocování měření. Výpočet vzdálenosti probíhá ve funkci *distanceCountF* a zjednodušeně je představen formou pseudokódu 5.2. Vzdálenost musí být dělená dvěma, protože signál putoval od překážky ještě zpět. Hodnota větší než maximální rozsah (nastane v případě, že není detekována žádná překážka a dojde k nastavení sestupné hrany na straně senzoru) pak není brána v potaz a je pouze nastavena maximální hodnota rozsahu.

```

1 rychlosťZvuku = 0.0343 #rychlosť pri 20 stupních Celsia v cm/us
2 vzdalenosť = delkaPulzu * rychlosťZvuku / 2
3 if vzdalenosť >= 450:
4     vzdalenosť = 450

```

Výpis 5.2: Ukázka jak je převedena délka pulzu na vzdálenost v cm.

Po naměření hodnoty (či po dokončení výpočtu průměrné hodnoty) dojde k zavolání funkce *isObstacle*, která zkонтroluje, zda nebyla překročena pevně nastavená prahová hodnota. Pokud ano, je zavolána ukončovací funkce a na motory nastavena rychlosť 0 %. V současném módu ještě musí být splněna podmínka, že již bylo dokončeno kolo a tudíž model se

již pohybuje pomalejší konstantní rychlostí, tudíž není třeba provádět ještě nějaké zpomalení a případné další měření. Tímto zastavením dojde k ukončení veškerého běhu programu i soutěžního pokusu v rámci soutěže NXP Cup.

Tabulka 5.4 představuje, kam jsou zapojeny jednotlivé senzory na rozšiřující desce i na mikrokontroléru. Tabulka pak 5.5 shrnuje, jaké poskytují výstupy z hlediska typu i rozsahu hodnot dosud v implementaci zmíněné senzory.

	<b>Modul</b>	<b>Pin KL27Z</b>	<b>Konektor deska</b>
Barevný senzor 1	TMP0 kanál 0	PTE24	RGB1 (J20)
Barevný senzor 2	TMP0 kanál 1	PTE25	RGB2 (J25)
Ultrazvukový senzor 1	GPIO, TMP0 kanál 5	PTC8, PTC9	Sonic1 (J27)
Ultrazvukový senzor 2	GPIO, TMP0 kanál 4	PTA5, PTE31	Sonic2 (J28)

Tabulka 5.4: Tabulka přehledně ukazující kam jsou na rozšiřovací desce i na mikrokontroléru zapojeny ultrazvukové a barevné senzory.

	<b>Typ vstupu na MCU</b>	<b>Rozsah hodnot</b>	<b>Hodnota</b>
Hallův senzor	Přerušení při nástupné hraně	0/1	ano/ne
Ultrazvukový senzor	Pulz zachytáváný pomocí TPM	0-38 ms	2-450 cm
IR senzor vzdálenosti	Převod analog ->digitál	0-4095	6-32 cm
IR senzor čáry	Přerušení při jakékoliv hraně	0/1	černá/bílá
Senzor barvy	Pulz zachytáváný pomocí TPM	podle frekvence	černá/bílá

Tabulka 5.5: Shrnutí toho, jaké dosud zmíněné senzory poskytují výstupy (na MCU vstupy), jaké mají rozsahy a s jakou finální informací (hodnotou) se může v implementaci pracovat.

## 5.6 Kamera Pixy 2 a navigační algoritmus

Práce s Pixy 2 kamerou a následně celý vyhodnocovací algoritmus, který ovládá pohyb modelu na základě získaných informací, se nachází v souboru *pixySPI.c*. Pro komunikaci s kamerou je využito rozhraní SPI, po němž jsou vždy do kamery poslány podle dokumentace<sup>1</sup> příslušné hodnoty bajtů. Na základě tohoto kamera pošle zpět informace o zaznamenaných vektorech a ty jsou dále zpracovány.

Jsou zde implementovány základní funkce jako *PixySetServos* pro nastavení úrovně serva (jeho rozsah je nastaven v prostředí PixyMon) a funkce *PixySetLamp*, které slouží k vypínání či zapínání přisvětlovacích LED). Servo i nastavení osvětlení je provedeno ve startovací funkci *PixyStart*, inicializace SPI komunikace je tam provedena také.

Každých 50 ms je volána pomocí časovače funkce *ProcessVectors*, ve které dojde ke zpracování přijatého bufferu. Podstatná data začínají až po 20 bajtech (mělo by to být dříve, nicméně kamera vždy poslala určité množství bajtů s konstantní hodnotou 1). První hodnota značí, kolik bajtů s daty poslala (na každý jeden vektor připadá 6 bajtů) a celá struktura je popsána tabulkou 5.6. Posílaní může být omezeno pouze na jeden primární vektor, nicméně toho nebylo využíváno, protože tento primární vektor se vyskytuje na první pozici i v případě, kdy není tato možnost zvolena.

<sup>1</sup>Dokumentace kamery Pixy 2 – [https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:porting\\_guide](https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:porting_guide)

Kolikátý bajt	Hodnota	Poznámka
0 – 12	1	pouze výplň, ignoruje se
13	173	
14	194	
15	49	identifikační bajty
16,17,18	počet dat za hlavičkou	délka + checksum
19	typ dat (vždy 1)	1 značí že data jsou vektory
20	N – délka dat v bajtech	V (počet vektorů) = N/6
21,22	x0,y0	první bod vektoru
23,24	x1,y1	druhý bod vektoru
25,26	index	unikátní index vektoru
další šestice vektorů dokud nejsou přečteny všechny vektory dané počtem V		

Tabulka 5.6: Tabulka ukazující složení přijatého bufferu při získávání vektorů během komunikace s kamerou Pixy 2 přes rozhraní SPI.

Buffer s vektory je postupně procházen a dochází ke kontrole každého kamery zaznamenaného vektoru. Byla připravena struktura pro jejich ukládání, ale nakonec nebylo třeba je nijak využít. Ke kontrole každého vektoru dochází ve funkci *CheckVector*, v níž je prvořádně vypočítána délka vektoru, směr a poměr těchto hodnot, což jsou důležité hodnoty, které budou později použity pro vyhodnocování. Následně jsou zkонтrolovány souřadnice vektorů tak, aby vždy bylo zaručeno, že počáteční souřadnice vektoru je blíže modelu.

Vždy je jako první zpracováván primární vektor, což je ten, který kamera určí při prvním snímání dráhy (většinou se jedná o nejdelší vektor) a na ten se pak snaží v dalších snímcích navazovat. Tento vektor jako jediný slouží k navigaci a vyhodnocování toho, kde se model na dráze nachází a zda má dojít ke směrové či rychlostní korekci. V generovaných obrázcích či videích je zobrazen primární vektor červenou barvou.

Byla testována i implementace, jenž zahrnovala procházení více vektorů a hledání jistých závislostí mezi nimi (například hledání pravých úhlů během křížení či při cílové čáře), nicméně se nepodařilo dosáhnout efektivního řešení, které by poskytovalo dobré výsledky.

Optimální nastavení kamery nakonec bylo nalezeno v tom, že kamera míří spíše na dráhu přímo před modelem a nikoliv do větší vzdálenosti. Toto nastavení bylo stabilnější v případě, kdy podklad pod dráhou nebyl ideální (jednalo se pouze o bílé papíry nebo bílý okraj o šířce pár centimetrů).

Primární vektor je vždy navazován do té doby, než zmizí ze záběru. To se stane v prudších zatáčkách, kdy dojde k přemístění tohoto vektoru na druhou stranu dráhy a navigace dále probíhá podle něj. Vektor projde kontrolou, zda se nachází v předdefinované pozici, která znamená, že je třeba provést korekci. Tento princip bude dále v jednodušší podobě popsán.

Na základě délky vektoru a jeho počátečního bodu je vyhodnoceno, jak velká korekce má být provedena – v případě dlouhého vektoru s malou hodnotou směru (směr je zde vyhodnocován jako absolutní hodnota rozdílu ypsilonových souřadnic vektoru, délka je pak rozdíl v ose x) se jedná o rovný úsek. Model sice jede rovně, ale jeho dráha nevede středem. V závislosti na tom, jak moc je model vychýlen, je provedena korekce ke středu dráhy.

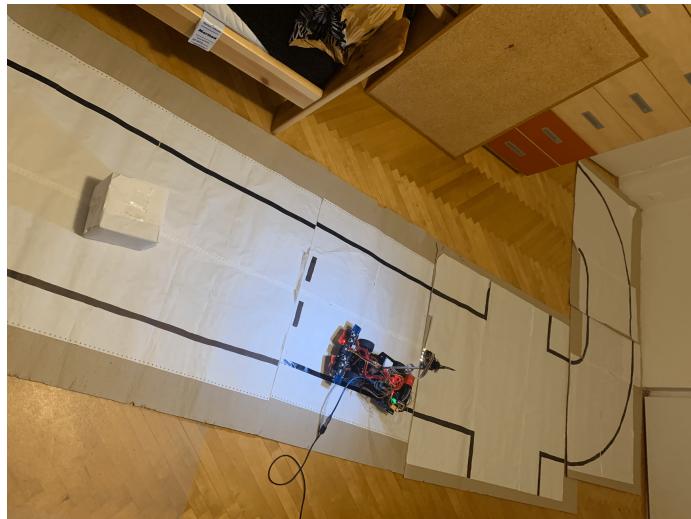
Dalším případem je situace, kdy vektor směruje kolmo ke směru jízdy ven z dráhy. To je detekováno díky delší čáře s větším směrem. Je vyhodnoceno, zda vektor směruje doleva či doprava a je provedena příslušná korekce. Následuje již pouze detekce, zda se model nachází v zatáčce, což je detekováno díky menší délce, ale zato vyšší hodnotě směru i jejich poměru.

Pakliže během kontroly primárního vektoru nedošlo ke zjištění nutnosti korekce, je nastavena na servu hodnota pro jízdu vpřed a je zvýšena rychlosť. Při nutnosti korekce je nastavena hodnota zatáčení a rychlosť snížena podle toho, o jak prudkou korekci se jedná. Všechny tyto hodnoty jsou uloženy do globálních proměnných a později jsou poslány společně s dalšími přes rozhraní UART.

Následuje pouze procházení dalších snímaných vektorů, ve kterých je hledána primárně dvojice horizontálních čar, které detekují cíl. Bylo zde třeba eliminovat falešné detekování, ke kterému by mohlo dojít například na křížení, což bylo vyřešeno tím, že vždy musí být detekován přesně 2 čáry, které nikdy nezačínají v krajích obrazu. Tento princip spolehlivě funguje a když je tato čára detekována v těsné blízkosti modelu (na videu je toto zaznamenáno fialovou barvou), dojde k zpomalení rychlosti modelu (jasně viditelnou změnu vyžadují pravidla soutěže). Model pak dále pokračuje konstantně pomaleji, dokud nedojde k detekci překážky, před kterou má zastavit.

Jednotlivé základní situace, které zde byly popsány, jsou zobrazeny na obrázcích 5.7 – 5.12 na následující straně, které jsou jednotlivými snímky z videa a obsahují i přesné popisky dané situace, ke kterým je možné i vidět nastavené hodnoty na motorech a servu. Obrázek 5.6 ukazuje domácí testovací dráhu, která se skládá z jednotlivých segmentů, na kterých byl testován navigační algoritmus pro kameru. Tabulka 5.7 představuje, jak je zapojena kamera Pixy 2 na rozšiřující desce a na mikrokontrolérku.

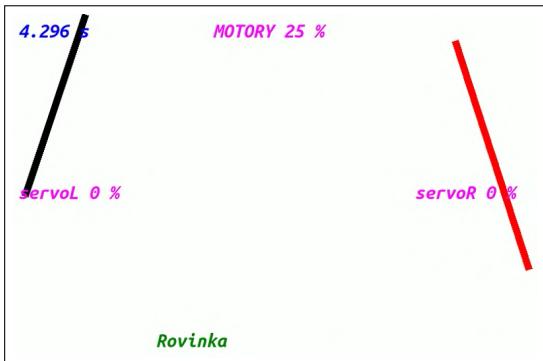
Tento zvolený algoritmus založený na jednoduchém principu, funguje velmi dobře. Je možné ho dále zlepšovat a vyladit některé nedostatky, především by bylo možné naprogramovat jemnější přechod mezi jednotlivými stavami či detektovat, jak dlouho se již model nachází v zatáčce a díky tomu zvýšit úroveň natočení či lépe zvládat prudké zatáčky. Je možné také zvyšovat rychlosť, nicméně to je možné správně v praxi otestovat jen na oficiální trati s většími rozměry.



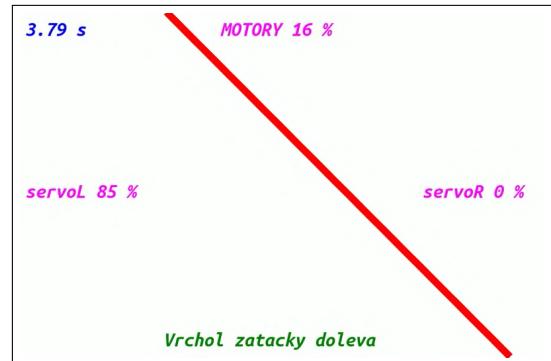
Obrázek 5.6: Ukázka testovací části dráhy, která byla používána v domácích podmínkách pro ověření základní funkcionality autonomního modelu.

	<b>Modul</b>	<b>Pin KL27Z</b>	<b>Konektor deska</b>
Pixy 2 kamera	SPI0	PTC4-7 (CS, SCK, MISO, MOSI)	Camera (J26)

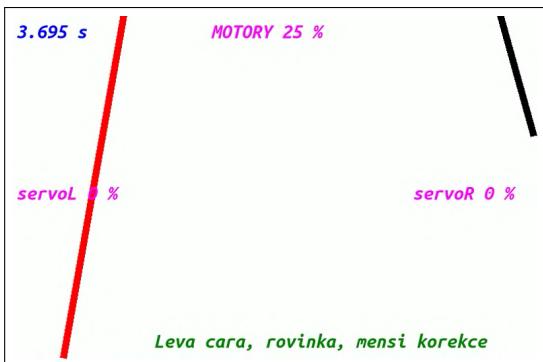
Tabulka 5.7: Představení toho, jak byla připojena na desku a mikrokontrolér kamera Pixy 2.



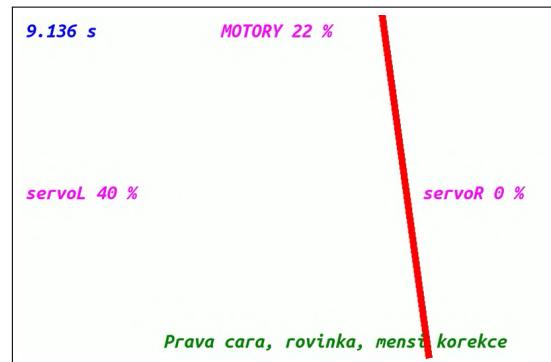
Obrázek 5.7: Ukázka detekované rovinky, není provedena korekce



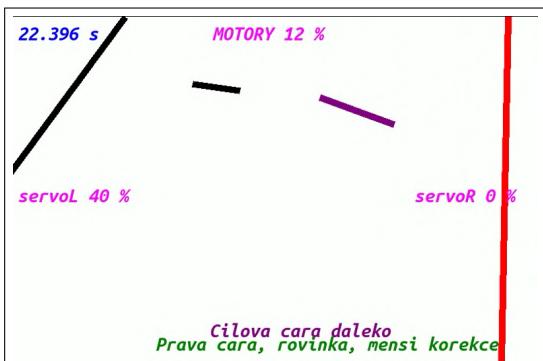
Obrázek 5.8: Ukázka detekce zatáčky doleva.



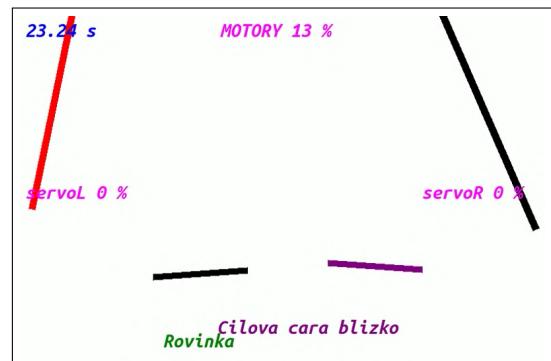
Obrázek 5.9: Detekce, kdy model jede rovně, ale je třeba provést korekci doprava ke středu dráhy.



Obrázek 5.10: Detekce, kdy model jede rovně, ale je třeba provést korekci doleva ke středu dráhy.



Obrázek 5.11: Ukázka detekce cílové čáry ve větší vzdálenosti s nutností ještě provést korekci.



Obrázek 5.12: Ukázka, kdy je na rovnici detekována cílová čára v blízkosti a model tímto zaznamená dokončené kolo.

## 5.7 Sběr, vyhodnocení a zpracování dat

Jak již bylo zmíněno v předchozí kapitole v sekci 4.7, k zasílání dat na externí zařízení je použito posílání dat přes UART na zařízení HC-05, které je pak dále přepošle přes rozhraní Bluetooth.

V programové sekci je tedy implementováno právě posílání dat přes UART, konkrétně se pak vše odehrává v souboru *uart\_control.c*, kde v inicializaci je nastaveno, že bude komunikace probíhat pouze v jednom směru (nebudou přijímána data) a je nastaven baudrate na 115200. Díky tomu je možné přenášet 11520 bajtů dat za sekundu (UART posílá vždy 8 bitů dat a k nim start a stop bit), což je pro potřeby projektu naprosto dostačující, protože data jsou posílána pro uložení v maximální frekvenci 20 Hz.

Pro sběr dat bylo implementováno více funkcí, které byly využívány během testování podle potřeby – někdy byly posílány pouze vektory, které posílá kamera, někdy pouze zaznamenávané hodnoty ze jednotlivých senzorů, které pak mohly být vyhodnocovány dle potřeby. Primární pak bylo sesbírat data během testovacího a soutěžního dne, kdy bylo možné, aby model jezdil na trati větších rozměrů, což nebylo v domácích podmírkách možné. V tom případě bylo využito funkce *UART2\_sendToHC05All*, jenž posílala data vždy, když docházelo ke zpracovávání vektorů, tedy právě s frekvencí 20 Hz. Díky tomu pak bylo možné data zpětně zpracovat a zanalyzovat a generovat video, které má stejnou frekvenci snímků jako má čtení dat z kamery.

K datům není potřeba nijak dodávat informace o čase, ty tam jsou přidány v aplikaci v mobilním telefonu, která tam vždy při výpisu jednoho řádku dodá aktuální čas přesný na tisíce sekundy.

Data byla později zpracována a vizualizována pomocí skriptů v jazyce Python, ve kterém časové údaje byly upraveny a následně byly vytvořeny grafy, se kterými je možné se setkávat v textu této práce. Dále také byla vytvořena videa, jenž 1:1 zpětně rekonstruuje a zobrazuje to, co vidí kamera a další senzory. Díky zobrazení všech vektorů je možné v datech jasně vidět trať a to, jaký vektor byl stěžejní pro navigaci. Tato videa, stejně tak jako všechna další naměřená data či grafy, které byly zpracovány či vytvořeny pomocí skriptů, je možné shlednout v příloze této diplomové práce, jejíž struktura je popsána v příloze C a jednotlivé snímky, které jsou generovány skriptem, se vyskytly v práci dříve například na obrázcích 5.7 – 5.12. Tabulka 5.8 pak opět shrnuje, kam je modul zapojen.

	Modul	Pin KL27Z	Konektor deska
HC-05 UART	UART2	PTE16	UART (J30) + 5V

Tabulka 5.8: Představení zapojení Bluetooth modulu HC-05 na rozšiřovací desku a na konkrétní pin mikrokontroléru.

# Kapitola 6

## Zhodnocení a náměty na rozšíření

V této kapitole dojde ke zhodnocení toho, jak vytvořené řešení obstálo z hlediska výsledné funkčnosti v reálných podmínkách i vzhledem k účasti na soutěži NXP Cup. Následně pak v sekci 6.2 dojde k představení námětů na možné rozšíření této práce v budoucnu, přičemž náměty budou svým charakterem primárně směřovat k dosažení lepších výsledků ve dříve zmíněné soutěži.

### 6.1 Zhodnocení řešení a dosažených výsledků

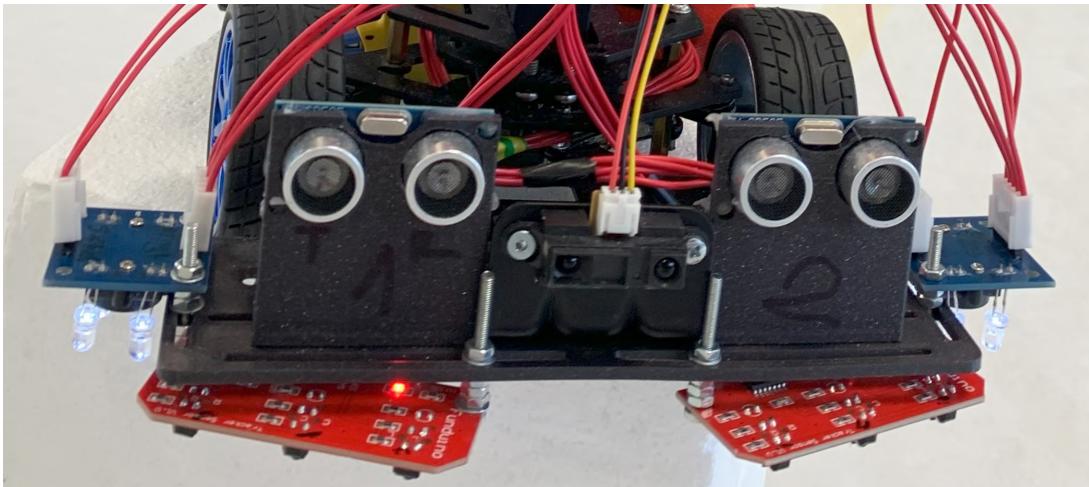
I přes prvotní potíže s motory se podařilo vytvořit funkční model, který se mohl úspěšně zúčastnit soutěže NXP Cup s ambicí na dosažení dobrého výsledku. Ačkoliv se vyskytovaly problémy se zprovozněním jednotlivých součástí, vždy se je podařilo vyřešit, případně najít jinou alternativu. Byla také vytvořena rozšiřující deska, což výrazně přispělo ke spolehlivosti řešení z hlediska napájení a odpadla nutnost kontroly toho, zda jsou jednotlivé senzory správně zapojeny. To umožnilo, především ve finální části práce v období před soutěží, zaměřit se plně na vývoj a testování samotného algoritmu a nikoliv na řešení technických problémů. Stále je zde velké množství vodičů, nicméně tomu se při použití senzorů v přední části modelu není možné vyhnout.

Positivně lze hodnotit to, že je možné model bez problémů ovládat a regulovat jeho rychlosť dle potřeby, stejně tak jako úhel natáčení. Konstrukce modelu nedovoluje dělat moc změn z hlediska typu motoru či serva, tudíž na nich byla založena celá funkcionalita projektu. Za drobný nedostatek je možné označit, že u motorů nebylo možné používat u regulátorů motoru módu brzdy, což by v některých případech mohlo vést ke zlepšení výsledků v soutěži díky možnosti využívat větších rychlostí a následného brzdění, nicméně z hlediska funkčnosti algoritmu a povahy zadání této diplomové práce se jedná pouze o detail, kterým je možné se zabývat v budoucnu v podobě postupného vylepšování tohoto řešení.

Stejně tak tomu je u zatáčení pomocí serva, kdy byly součásti ramen směřujících od serva ke kolu napevno zlepšeny, aby nedocházelo k jejich vyšroubování (při otřesech) a tím ke změně směřování kol při jízdě rovně. Díky tomu je zmenšena možnost nastavení maximální úrovni zatáčení, nicméně tím je vždy zaručena stejná pozice pro jízdu rovně. Díky tomu je řešení stabilní a spolehlivé, což bylo vzhledem k cíli zvládnout bez problémů projíždět dráhu v soutěžním prostředí základem k tomu, aby mohl být implementován funkční algoritmus pro další zpracovávání.

Vhodně byla zvolena i skladba senzorů sloužících k monitorování čar. Byly vybrány spíše technicky jednodušší typy a dopředu byly zváženy možnosti toho, jak budou použí-

vány z hlediska implementace a kde na modelu budou umístěny. To se promítlo do návrhu platformy (osazená na modelu lze vidět na obrázku 6.1), vyrobené na míru prostřednictvím 3D tisku a umístěné na přední část modelu.



Obrázek 6.1: Ukázka přední platformy, na které jsou osazeny všechny používané senzory.

Barevné senzory fungují dobře, mohou být ve větší výšce od podložky a čáru zvládnou spolehlivě a rychle vyhodnotit, tudíž na ni je možné okamžitě reagovat. Drobou nevýhodou je velký počet výstupních pinů a tedy potřeba dvou svazků přivedených vodičů s konektory. Jeden senzor měl právě na těchto pinech horší kontakt, takže se občas stalo, že senzor vůbec neměřil a bylo vždy třeba provést kontrolu po spuštění, zda vše funguje tak, jak má. Nabízelo se přívodní kabely trvale připájet, z důvodu snahy o co nejvíce modulární řešení však toto nebylo realizováno.

Jako problematické a ne úplně optimální řešení bylo vyhodnoceno použití IR senzorů pro sledování čar. V domácích podmínkách a na hladké dráze fungují velmi dobře – dokázaly by zvládnout reagovat i na dráhu, která by neměla bílý povrch. Pro správnou funkčnost musejí být tyto senzory velmi nízko, aby zvládly správnou detekci, což je problém v případě soutěžní dráhy, protože je složena z jednotlivých částí a na jejich spojích někdy vznikají nerovnosti (lze vidět na obrázku 6.2). Při větší nerovnosti se stane, že tyto senzory na desce zavadí o zem a může díky tomu dojít k jejich poškození. Stejně tak tomu je při návratu modelu na trať v případě jejího opuštění (desky, ze kterých je trať složena, končí černým ohrazením a za ním je již pouze bílý podklad, který však není součástí trati samotné).



Obrázek 6.2: Nerovnosti, které vznikají na trati mezi jednotlivými segmenty dráhy.

Řešením tohoto problému s infračervenými senzory by bylo vytvoření vlastní desky, na které by byly osazeny senzory TCRT5000 a byly by zapuštěny tak, aby bylo co nejvíce eliminováno riziko poškození.

Barevné senzory a IR senzory byly používány spíše jako doplňkové a nestojí na nich primární navigační algoritmus. Díky jejich principu, kdy na jejich zaznamenané hodnoty je třeba okamžitě reagovat v podobě zatočení, nebylo nijak přínosné data z těchto senzorů nějak více analyzovat a zobrazovat v podobě grafů či snímků jako u ostatních senzorů, byla data zaznamenávána periodicky. Výstup v podobě frekvence byl u barevných senzorů zobrazen dříve na obrázku 4.11.

Z hlediska primární navigace, ke které slouží kamera Pixy 2, bylo dosaženo dobrého výsledku s ohledem na to, jaká data je kamera schopna poskytnout. Tato kamera se potýká s poměrně velkým množstvím drobných nedostatků, asi nejvážnější je nižší rychlosť poskytování dat pro následné zpracování. Nepodařilo se data číst rychleji než s frekvencí 20 Hz, tedy každých 50 ms. Nicméně i toto zpracování je pro základní navigaci dostatečné, minimálně při použitých rychlostech nebylo shledáno, že by mělo být problematické.

Ve stejně frekvenci pak byla data logována a zpětně z nich docházelo ke generování videa (tedy se snímkovou frekvencí 20 FPS<sup>1</sup>). Tato videa je možné shlédnout v příloze C, kde jsou společně s pořízeným videozáznamem spojena do jednoho a slouží tak jako demonstrace celkové funkčnosti navigačního algoritmu. Stejně tak jsou jednotlivé snímky z videa představeny dříve na obrázcích 5.7 – 5.12, které slouží k představení jednotlivých základních situací, ve kterých dochází k vyhodnocování toho, kde se model na dráze nachází.

Dalším faktorem, který ovlivňuje chování kamery, je osvětlení místnosti. Pro každé prostředí musí být kamera přenastavena pro dané světelné podmínky. Zvláštností je, že v domácích podmínkách, které jsou z hlediska osvětlení horší, nebylo nastavení tolik problematické, protože si kamera je schopna pomocí LED obstojně přisvítit. Horší, především pak v soutěžním prostředí, je situace, kdy je v okolí dráhy více zdrojů světla, které mohou ovlivňovat snímání. Konkrétně v místě, kde se konala soutěž, byly po obou stranách velká okna, díky kterým byly na jedné části díky dennímu světlu jiné světelné podmínky než na části druhé (lze vidět na obrázku 6.3 dále). To v některých případech ovlivnilo chování modelu na dráze, protože některé čáry nebyly zachyceny celé (problematické to může být především v zatáčkách). Pro kameru je také nutné mít pod dráhou bílý podklad bez znatelných přechodů, které kamera v některých případech také může falešně detektovat.



Obrázek 6.3: Ukázka situace, kdy denní světlo svítící na trať může negativně ovlivnit, co snímá kamera (červený vektor by v tomto případě měl být detekován až do horní části obrázku stejně tak jako zelený).

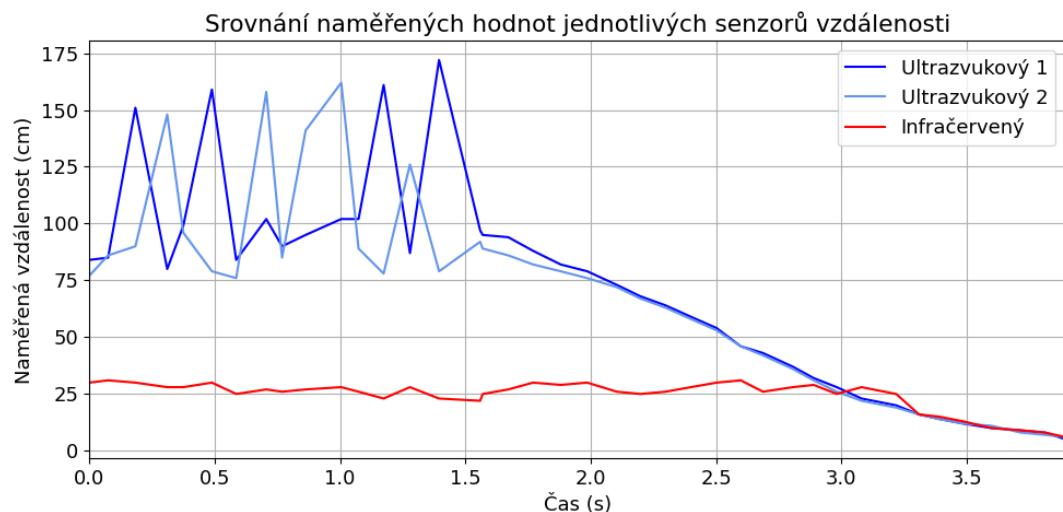
<sup>1</sup>FPS – Frames Per Second (snímky za sekundu)

Algoritmus, který byl vytvořen na základě kamerou poskytnutých vektorů, funguje dobře – zvládá pohyb v domácí podmírkách na menších segmentech dráhy, stejně tak kroužit na menších okruzích bez křížení, ale také komplexní dlouhou trať se všemi částmi, které se mohou v soutěži NXP Cup vyskytnout. Problematické může být v jistých situacích křížení (v případě, že těsně před ním je prudší zatáčka). Vytvořené řešení zvládá projet všechny typy zatáček, problém nastane pouze v případě, když jsou zkombinovány za sebou. Tento problém je více popsán později v části textu, který se týká přímo účasti na soutěži, kde tato konkrétní situace nastala.

Alternativou pro dosažení lepších výsledků by bylo použití lineární kamery či ekvivalentního 1D snímače, jenž by poskytl možnost zpracování zaznamenaného obrazu s výrazně vyšší frekvencí. Implementace tohoto algoritmu by však byla pravděpodobně složitější a bylo by třeba zvolit jiný přístup vyhodnocování, než který byl využíván na základě dat o vektorech, které poskytla kamera Pixy 2.

Dobře fungují také senzory pro detekci překážky, zde však bylo řešení s použitím dvou ultrazvukových senzorů a jednoho IR snímače poměrně předimenzované. Všechny ostatní týmy používaly pouze jeden senzor a i takové řešení by bylo pro detekci překážky dostatečné. Použité řešení však stabilně funguje a kdyby byl použit IR senzor s větší detekční vzdáleností, mohlo by být řešení flexibilnější. Stejně tak by bylo možné použít dříve zmínovaný laserový senzor či jinou alternativu, ultrazvukové senzory však byly nejflexibilnější a osvědčily se.

Problém, který však v konečném výsledku mají všechny testované senzory pro detekci překážky, je to, že nejlépe pracují v prostředí, kdy detekují překážku, která je vždy umístěna v jejich rozsahu. Jakmile koukají "do prázdná" či je překážka mimo detekční rozsah, zaznamenané hodnoty jsou rozptýlené v části jejich rozsahu se skoky v datech, což lze vidět na obrázku 6.4. Na něm je možné vidět měření ze všech senzorů v situaci, kdy se model pohyboval konstantní rychlostí po dráze, na které byla umístěna překážka přes celou dráhu tak, aby všechny senzory měly možnost ji zachytit. Klíčová však je detekce překážky v prostoru maximálně 50 cm před modelem, což ultrazvukový senzor spolehlivě poskytuje. IR senzor má omezený rozsah (detailněji byl popsán na obrázku 4.4) a aby mohl být použit jako spolehlivý hlavní senzor, je třeba použít obdobný typ senzoru s větším rozsahem.

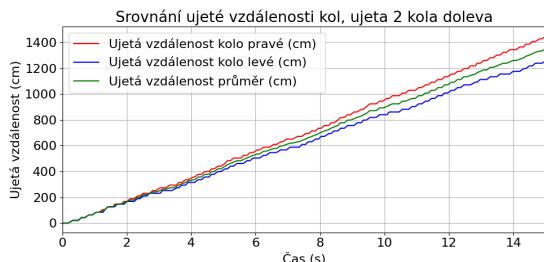


Obrázek 6.4: Ukázka co naměřily jednotlivé senzory při přibližování k překážce. Ultrazvukové spolehlivě měří přibližně od vzdálenosti 1 m, infračervený senzor pak až 16 cm a méně.

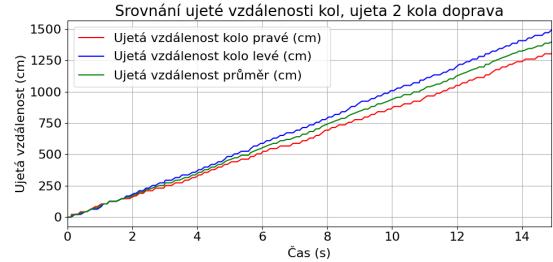
Hallový senzory, které během pohybu zaznamenávaly počet otáček jednotlivých kol, spolehlivě poskytovaly důležitá data pro analýzu. Prvotně posloužily ke kalibraci výkonů jednotlivých motorů tak, aby bylo dosaženo stejných otáček (výsledná data po kalibraci je možné vidět v grafu na obrázku 4.2 v sekci týkající se motorů). Také díky nim bylo možné získávat hodnoty důležitých veličin (ujetá vzdálenost, aktuální rychlosť), které slouží jako informační v generovaných videích, viz obrázek 6.8. Grafy na obrázcích 6.6 a 6.7 ukazují rozdíl v počtu otáček u jednotlivých kol modelu při ujeté vzdálenosti dvou kol na trati z obrázku 6.5. Na grafech je vidět, že vnitřní kolo ujede během jednoho okruhu dlouhého 660 cm přibližně o 100 cm méně než kolo vnější, které opisuje delší křivky. Celková ujetá vzdálenost modelu je průměrem z otáček levého i pravého kola.



Obrázek 6.5: Dráha složená do okruhu, na které bylo testováno měření otáček. Grafy s vyhodnocením jsou na následujících obrázcích.



Obrázek 6.6: Zaznamenaná data při kroužení doleva.



Obrázek 6.7: Zaznamenaná data při kroužení doprava.

Na té samé trati také proběhla testovací měření, jejichž výsledek vypovídá o jedné z nevýhod tohoto algoritmu. Model na dráze ujede delší vzdálenost než je nutné – nevolí totiž pro průjezd zataček nejkratší možnou dráhu. Podle reálných výsledků je ujetá vzdálenost vyšší i oproti situaci, kdy by jel celou dobu středem. Krom průjezdů zataček toto způsobuje i následné chování na rovině, kdy se k návratu do středu používá prudké zatočení ke středu, které však znamená nutnost ujetí delší vzdálenosti než kdyby korekce byla plynulejší a na delším úseku. Tabulka 6.1 představuje naměřené hodnoty (průměr obou kol) z kroužení na tomto okruhu. Délka jednoho kola při teoretickém průjezdu středem dráhy je 660 cm.

Hodnoty se tomuto číslu přiblížují, zajímavé je, že při kroužení doprava (po směru hodinových ručiček) model průměrně ujede přibližně o 20 cm více (důvodem bude pozdější reakce navigačního algoritmu). První měření je vždy nejkratší, pravděpodobně z důvodu, že model byl při startu umístěn na středu a do první zatačky vždy jel rovně bez korekce,

tedy nejkratší cestou. Jedná se o odchylky, které sice mají v cíli vliv na výsledný čas, ale z hlediska celkového řešení se jedná spíše o detail.

	<b>Kolo 1</b>	<b>Kolo 2</b>	<b>Kolo 3</b>	<b>Kolo 4</b>	<b>Kolo 5</b>	<b>Průměr</b>	<b>Délka kola</b>
<b>Doprava</b>	670 cm	710 cm	700 cm	690 cm	710 cm	696 cm	
<b>Doleva</b>	660 cm	670 cm	670 cm	690 cm	680 cm	674 cm	660 cm

Tabulka 6.1: Ukázka naměřených hodnot při testovacích průjezdech na okruhu.

Tabulka 6.2 ukazuje testovací měření provedená za účelem ověření přesnosti snímání otáček jednotlivých kol. Měření bylo provedeno na rovném úseku, kdy model vždy vyjel ze stejné pozice směrem k překážce, kterou detekoval ve vzdálenosti 50 cm od ní, po ujetí vzdálenosti 275 cm. Využito bylo napájení 7 V, stejně jako po celou dobu testování modelu, přičemž maximálně bylo využito 30 % z maximálního možného výkonu při tomto napájení. Při výšších hodnotách již model jede moc rychle a nebyly využity ani pro soutěž.

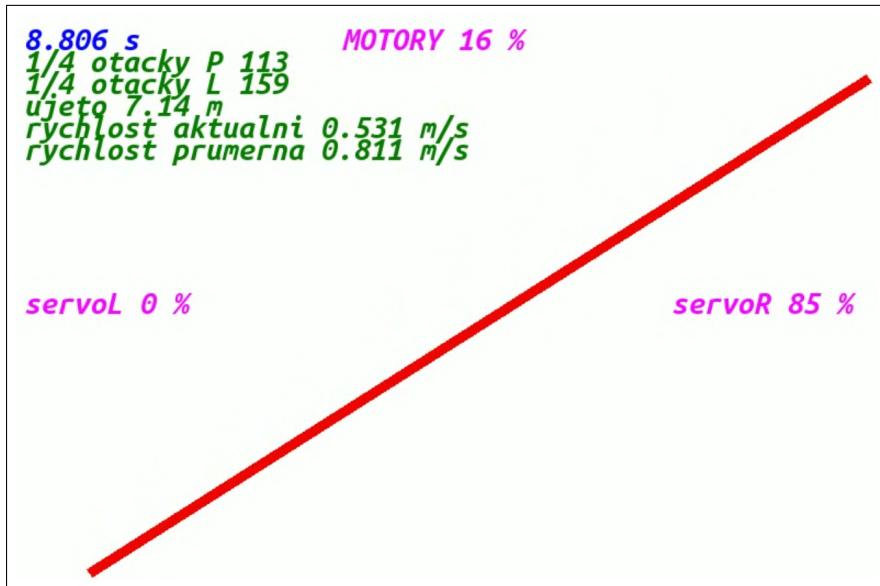
Měření má díky principu zaznamenávání jistou odchylku – v maximálním případě až  $\pm 5,5$  cm (díky rozmístění 4 magnetů tak, že detekce každého znamená přičtení této hodnoty k výsledné vzdálenosti). Nejpřesnějších výsledků bylo dosaženo při výkonu 15 % a 20 %, které jsou ekvivalentem pro plynulou jízdu modelu a byly využívány i v soutěži.

Při výkonu 10 % bylo dosaženo drobné odchylky pramenící z toho, že při tomto výkonu model jede pomalu a v některých případech téměř dochází k zastavení. Bylo vyzozorováno, že během tohoto pohybu dochází k protáčení ozubeného kola, na kterém jsou připojeny magnety. Díky tomu jsou zaznamenané vyšší hodnoty, ve výsledném měření se však jedná o páár centimetrů, přičemž se jedná o výkon, který není téměř vůbec využíván.

Hodnota 30 % pak naopak už pro model znamená velký výkon – v soutěži by tyto hodnoty znamenaly takovou rychlosť, na kterou by již bylo problematické reagovat (maximálně bylo využíváno 25 % výkonu při jízdě na rovince). Větší hodnota je zaznamenána z důvodu protočení kol při rozjezdu, nicméně na testovaném povrchu (parkety) se jedná pouze o drobnou nesrovonalost, ke které dochází pouze při rozjezdu. Pro každý povrch by byly hodnoty lehce odlišné, soutěžní dráha je z materiálu, na kterém snáze dochází k protáčení kol. Tato série měření potvrdila, že výsledné řešení dokáže zaznamenávat vzdálenost velmi přesně.

<b>Výkon motoru při 7 V</b>		<b>10 %</b>		<b>15 %</b>		<b>20 %</b>		<b>30 %</b>		
<b>Délka úseku 275 cm</b>	<b>1/4 otáčky kol</b>	<b>1. měření</b>	51	50	49	50	51	50	52	52
		<b>2. měření</b>	52	51	50	50	49	49	52	53
		<b>3. měření</b>	49	52	50	49	50	50	52	53
		<b>4. měření</b>	52	52	51	50	51	50	53	51
		<b>5. měření</b>	52	51	50	51	50	51	52	52
		<b>6. měření</b>	52	51	49	49	49	50	53	53
		<b>7. měření</b>	52	52	51	50	51	50	51	53
		<b>8. měření</b>	52	51	50	51	50	49	51	52
		<b>9. měření</b>	50	50	49	48	50	50	52	53
		<b>10. měření</b>	50	51	50	49	50	49	53	51
<b>Průměrně 1/4 otáček</b>		51,15		49,8		49,95		52,2		
<b>Vzdálenost (cm)</b>		281,325		273,9		274,725		287,1		
<b>Průměrný čas (s)</b>		9,5		4,6		3,3		2		
<b>Průměrná rychlosť (<math>\frac{m}{s}</math>)</b>		0,3		0,6		0,83		1,44		

Tabulka 6.2: Výsledků měření otáček obou kol na rovném úseku o délce 275 cm.



Obrázek 6.8: Jeden snímek z videa, kde na základě detekce magnetů pomocí Hallových senzorů je z počtu otáček počítána ujetá vzdálenost, aktuální a průměrná rychlosť.

Spolehlivě a efektivně se také podařilo vyřešit sběr a vyhodnocení dat, díky čemuž je možné v reálném čase sledovat hodnoty jednotlivých senzorů či zpětně vytvořit video, které zrekonstruuje zaznamenané hodnoty z kamery a při srovnání s videozáznamem je možné analyzovat, kde se model choval nestandardně a jaké hodnoty k tomuto chování vedly. Data jsou zasílána v datově co nejúspornějším strukturovaném formátu, zobrazena v aplikaci (snímek obrazovky je pro představu možné vidět na obrázku 6.9 a popis na 6.1) a v téže aplikaci jsou data uložena do souboru. Řešení je flexibilní především díky tomu, že ke sledování dat stačí pouze mobilní telefon (v případě testování konkrétních částí byla posílána pouze data týkající se daného problému). Ze sesbíraných dat byla vytvořena již dříve zmíněná videa, ale také veškeré grafy či obrázky ukazující vektory, které se vyskytly v průběhu této práce.

```

Terminal
1 02:52:56.421 0;339;29;20;0;0;30;25;0;0;67;51;58;0;124;75;15;78;6;226;76;20;75;15;245;75;15;74;10;246;
1 02:52:56.471 0;406;20;23;0;0;28;22;40;0;67;51;57;0;124;75;15;77;7;226;
1 02:52:56.532 0;412;20;23;0;0;30;25;0;0;67;51;58;0;124;74;14;78;6;226;
1 02:52:56.596 0;478;20;23;0;0;28;22;40;0;67;51;57;0;124;76;19;75;13;226;75;13;78;6;249;
1 02:52:56.647 0;480;20;23;0;0;29;25;0;0;67;51;58;0;124;74;15;78;5;226;
1 02:52:56.697 0;516;20;23;0;0;27;22;40;0;67;51;58;0;124;74;15;75;10;226;
1 02:52:56.741 0;518;20;23;0;0;30;22;40;0;67;51;57;0;124;74;14;78;6;226;
1 02:52:56.839 0;518;20;23;0;0;28;25;0;0;67;51;58;0;124;75;16;75;10;226;75;10;78;6;250;
1 02:52:56.846 0;524;20;23;0;0;28;22;40;0;67;51;58;0;124;74;13;78;6;226;
1 02:52:56.866 0;529;20;23;0;0;28;22;40;0;67;51;58;0;124;74;10;78;6;226;
1 02:52:56.916 0;529;20;23;0;0;29;22;40;0;67;51;58;0;124;75;16;73;10;226;

```

Obrázek 6.9: Ukázka strukturovaného formátu ve kterém jsou zasílána data.

```
1 cas otacky1;otacky2;cm1;cm2;barva1;barva2;cm3;motory%;servoP%;servoL%;x0,y0,x1,y1,index;...vektorN-1;
```

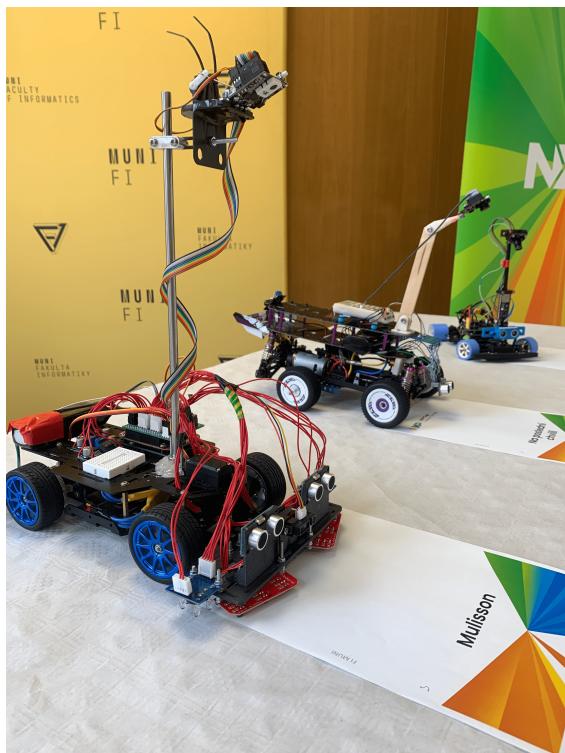
Výpis 6.1: Ukázka co reprezentují jednotlivé hodnoty z obrázku 6.9

### 6.1.1 Účast na soutěži NXP Cup

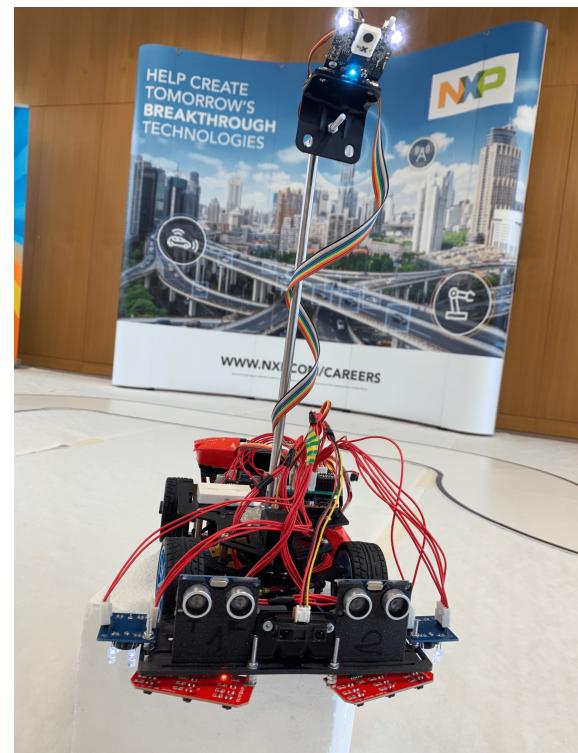
Účast na soutěži byla jedním z hlavních cílů, ke kterým bylo směrováno úsilí v průběhu této diplomové práce. Postupně se řešení rozšiřovalo a navigace byla testována v domácích podmínkách na jednotlivých segmentech. K hlavnímu testování na dráze pak docházelo na dvou testovacích dnech pořádaných firmou NXP na stejné trati, na které se později závodilo v rámci soutěžního dne.

Jeden testovací den nemohl být naplno využit kvůli selhání serva pro ovládání směru jízdy modelu, na druhém se již povedlo otestovat vše potřebné a potvrdilo se, že je model implementačně připraven na soutěž.

Na tu bylo celkem přihlášeno 17 týmů, ale na soutěžní den dorazilo pouze 11 a pouze 6 bylo schopných nastoupit se svými modely k soutěžním pokusům. Díky tomu, že soupeři byly již zkušené týmy, které se v minulých letech účastnily i celoevropských finále, bylo hlavní ambicí soutěžní kolo dokončit a co nejvíce se těmto týmům v rámci možností přiblížit. Proto bylo pro finálové pokusy zvoleno konzervativnější nastavení týkající se rychlostí, které bylo otestováno jako funkční během testovací části dne. Cílem bylo s jistotou projet trať v prvním okruhu a poté bezpečně zastavit před překážkou. Právě zastavení před překážkou bylo problematické pro většinu týmů, nakonec se to podařilo pouze jednomu.



Obrázek 6.10: Modely před soutěží během technické kontroly.



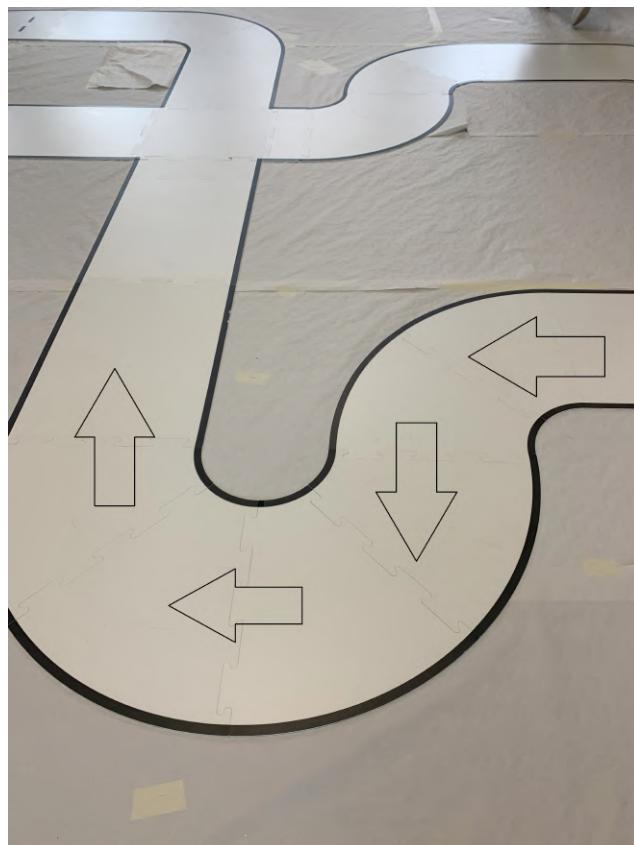
Obrázek 6.11: Čelní pohled na model během soutěžního dne.

Bohužel finálová trať obsahovala takovou sekvenci zatáček, která nebyla nikdy dříve testována. Jednalo se o zatáčku o  $180^\circ$ , které přímo předcházela opačná zatáčka v úhlu  $90^\circ$  (viz obrázek 6.12) a nemile to překvapilo i zkušené týmy. Algoritmus byl implementován společně s nastavením zorného úhlu kamery tak, aby reagoval na vjezd do zatáčky poz-

ději a volil spíše vnější stopu na jejím výjezdu (při dřívějším vjezdu někdy byly zatáčky detekovány moc brzy a model díky tomu vyjel z trati ještě před zatáčkou).

Toto nastavení fungovalo dobře na všech testovaných tratích nebo jejich úsecích, kdy sekvence opačných zatáček nedělala problém, případně i jeden rovný segment trati za zatáčkou většinou stačil pro vyrovnání modelu ke středu trati. V případě, kdy však následovala otočka o  $180^\circ$ , tak model během soutěžních pokusů tento úsek nedokázal projet. Příčiny jsou dvě – samotná velikost modelu, kdy je zde těžké tuto otočku vytočit v případě, kdy na začátku této dlouhé prudké zatáčky je model v její vnitřní straně. Další je pak fakt, že tato situace a kombinace zatáček nebyla brána v úvahu v samotné implementaci. Řešením bylo zvýšit stupeň zatáčení v případě detekce čáry pomocí barevných senzorů v takovém případě, kdy model má informaci o tom, že se nachází v prudké zatáčce, případně upravit detekci zatáček tak, aby na ně model reagoval dříve.

Tímto inkriminovaným úsekem se bohužel nepodařilo v soutěžním pokusu 2x projet, na třetí pokus došlo k úpravě pohledu kamery, nicméně jelikož nebylo možné zkонтrolovat, co snímá, tak se modelu nepodařilo projet křížením na trati. Po soutěži se však pouze s jiným nakloněním úhlu kamery soutěžní dráhou podařilo bez problémů projet. Videozáznam z těchto pokusů je stejně jako další soubory v příloze C. Díky vyjetí z dráhy nebylo dokončeno soutěžní měřené kolo, tudíž se model umístil mezi poraženými, nicméně výsledek samotný může být příslibem do budoucna a je na něm možné dále budovat efektivnější a spolehlivější řešení, které by již mohlo být konkurencí i těm nejlepším týmům.



Obrázek 6.12: Ukázka problematického úseku dráhy, který se objevil jako soutěžní během NXP Cupu.

## 6.2 Možnosti rozšíření řešení

V této sekci budou představeny návrhy a možnosti toho, jak může na řešení zpracované v této diplomové práci být v budoucnosti navázáno. Primárně by se mohlo jednat o rozšiřování a zlepšování implementace v podobě navigačního algoritmu tak, aby se podařilo v příštích ročnících soutěže dosáhnout co nejlepších výsledků.

Rozšíření stávajícího algoritmu není doporučeno v hledání jiných alternativních řešení z hlediska senzorů vzdálenosti (ty fungují dobře a spolehlivě) či přímé detekce čáry (jedná se pouze o doplněk). Úpravy by se měly primárně zaměřit na zlepšování stávající implementace navigačního algoritmu zpracovávajícího data z kamery Pixy 2 či vytvoření podobného algoritmu, který by však byl založen na použití lineárního senzoru. Ten by musel být vybudován od základů, nicméně je zde potenciál pro dosažení lepších výsledků v důsledku možnosti rychlejšího sběru dat či lepší možnosti detekce toho, kde se model nachází a jak má reagovat na vymezující čáry. Na soutěži s tímto typem snímače fungovala téměř všechna konkurenční řešení od ostatních týmů, tudíž to je řešení, které by mohlo být efektivní a mohlo by vést k posunu jak z hlediska funkčnosti, tak zlepšení dosažených výsledků na soutěži.

Další možností, která by mohla být předmětem rozšíření funkčnosti autonomního modelu, je rozšíření komunikace přes rozhraní UART. V aktuálním řešení slouží primárně pouze v režimu, kdy model odesílá data a ta jsou přijímána na externím zařízení, což je řešení, které mělo především zastoupit úlohu SD karty. To bylo splněno a navíc byla možnost sledovat hodnoty zaznamenávané modelem v reálném čase, což šetřilo při testování na oficiální soutěžní dráze cenný čas. Rozšíření by mohlo spočívat v přidání komunikace i opačným směrem, k čemuž by mohly být přidány jednoduché pokyny, jako například okamžité zastavení modelu. Tato funkcionalita by se uplatnila při testování na velké trati v případě, kdy model opustí trať a začne se nekontrolovaně pohybovat po okolí.

Komunikaci by pak bylo možné rozšířit o jednoduchou samostatnou aplikaci, která by sloužila k přímému ovládání modelu (šipky, jak je známe z počítačových klávesnic, nastavení úrovní motorů či úhlů natočení kamery). Díky tomu by mohly být lépe testovány například rychlosti, kterými se model může pohybovat.

Posledním užitečným nástrojem, který by mohl pomoci při implementaci a vylepšování navigačního algoritmu, by mohlo být zobrazovací zařízení, ať už pro zobrazování důležitých dat ze senzorů v reálném čase, či pro okamžitou rekonstrukci toho, co vidí kamera. To by mohlo suplovat funkci aplikace PixyMon, ke které je nutné, aby byla kamera připojena přes USB kabel do PC. Možnosti by bylo osadit externí displej ovládaný pomocí mikrokontroléru, ten by však musel zůstat na modelu, praktičtější by proto bylo zabudování přímo do dříve zmíněné ovládací aplikace.

### 6.2.1 Dokument sloužící jako průvodce implementací obsahující úlohy pro zvládnutí soutěže NXP Cup

Model ve stavu, ve kterém byl dokončen společně s touto prací, je schopen se zúčastnit soutěže téměř bez jakýchkoliv úprav a je možné ho tedy používat a v příštích ročnících soutěže NXP Cup navázat na aktuální řešení a výsledky. Budoucí student a účastník soutěže se však musí detailně seznámit s fungováním všech součástí modelu, a to jak po hardwarové, tak po softwarové stránce.

Za tímto účelem byl vytvořen dokument přiložený v příloze C, který může sloužit jako seznamovací průvodce k tomu, aby bylo možné rychle a efektivně zprovoznit všechny zá-

kladní funkce modelu a pak se již primárně soustředit na vylepšování algoritmu, který se stará o co nejrychlejší průjezd soutěžní dráhy.

Jedná se spíše o neformální dokument doplněný vlastními zkušenostmi s jednotlivými součástmi modelu a jejich zprovozňováním, které výrazně mohou pomoci pochopit danou problematiku. Jsou zde také představeny konkrétní problematické části, na které je se třeba zaměřit při vylepšování implementace. Zmiňuje také jednotlivé součástky, jejich výhody a nevýhody, případně také jejich alternativy, které by je mohly teoreticky nahradit či doplnit.

Podrobně je také představena rozšiřovací deska, která byla v rámci této práce vytvořena a její použití značně usnadňuje práci při připojování jednotlivých součástí, především díky tomu, že jsou již správně rozvedeny jednotlivé úrovňě napětí. Díky tomu není třeba se touto problematikou dále zaobírat a stačí pouze zapojit ty konektory, které chce uživatel zrovna využít.

Dokument na konci také představuje sadu úloh, jejichž postupné plnění od základů ověřuje splnění základních požadavků, které jsou kladený na model. V základních úlohách dojde k ověření funkčnosti základního vybavení modelu v podobě motorů a serva, stejně tak je ověřena funkčnost a schopnost reagovat na překážky na trati. Následují pak úlohy na základní navigační algoritmus na rovných úsecích, kdy je cílem udržet model na dráze a vrátit ho vždy do středu dráhy.

V dalších úlohách je již testována adaptace i na jiné segmenty dráhy – důležité je detekce různých druhů zatáček a adekvátní reakce na ně či detekce cílové čáry, na kterou je třeba adekvátně zareagovat.

Další úloha do řešení vnáší možnost zapojit další senzory, které mohou algoritmus vylepšit a snížit tím riziko vyjetí z trati. Je tím ověřena základní funkčnost těchto senzorů a jsou tím představena možnosti, jak je využít jako doplňky pro navigační algoritmus.

Poslední úloha již slouží ke komplexnímu otestování modelu na kompletních úsecích trati, nejlépe během testovacích dní či tréninku před samotnou soutěží. Jsou zde představeny ty části dráhy, které pro model vytvořený v této diplomové práci představovaly problémy a jejich vyřešení by mělo stávající model vylepšit tak, že bude mít potenciál úspěch v dalších ročnících soutěže.

Tato sada úloh byla vytvořena na základě vlastních zkušeností během vývoje autonomního modelu pro soutěž NXP Cup a skládá se ze úloh, kterými model postupně prošel a díky jejich posloupnosti tím došlo k otestování důležitých požadavků na funkčnost modelu, se bez problémů mohl této soutěže zúčastnit.

# Kapitola 7

## Závěr

Cílem této diplomové práce bylo seznámit se s možnostmi návrhu samořiditelných modelů a úskalími, se kterými se toto odvětví potýká. Na základě získaných poznatků pak bylo stanoveno za cíl navrhnut model tak, aby se mohl zúčastnit soutěže NXP Cup a vytvořené řešení bylo příslibem i pro další ročníky.

V úvodní části byl shrnut aktuální stav a popsána samotná soutěž z hlediska pravidel a již vytvořených řešení. Popsán byl také použitý model autíčka spolu se základním vybavením v podobě motorů a serva.

Následuje část práce týkající se teoretické rešerše dané problematiky. Zde byly představeny vestavěné systémy, autonomní mobilita a její úrovně či základní fyzikální veličiny, které mají vliv na pohyb modelu. V neposlední řadě došlo k představení základních stavebních prvků budoucí implementace, jako jsou například pulzně šířková modulace, převodníky a vybraná komunikační rozhraní, která byla v projektu dále využita.

Další část se již zabývala návrhem výsledného řešení autonomního modelu a byly po logických celcích představeny vybrané součástky. Prvně ty týkající se napájení, pohonu a základního ovládání modelu, následně senzory, jež je třeba osadit z důvodu implementace navigačního algoritmu či detekce překážek.

Podle finálního výběru součástek pak došlo k vytvoření samotné implementace. Ta byla budována od základů a postupně rozšiřovala funkčnost modelu. Na základě poznatků získaných během vývoje byla sestavena sada úloh, které jsou popsány ve vytvořeném dokumentu. Ten slouží jako průvodce implementací i jako celkový návod na zprovoznění řešení pro budoucí účastníky soutěže. Byla také vytvořena rozšiřující deska umožňující pohodlné a spolehlivé zapojení jednotlivých součástek do jednoho kompaktního celku.

Řešení bylo podrobeno testování na jednotlivých segmentech domácí dráhy a poté na oficiální dráze během testovacích dnů a samotné soutěže. Zde také došlo ke sběru dat, která posloužila k vytvoření výstupů v podobě grafů či videí, na kterých je díky zpětné rekonstrukci možné vidět, co zaznamenávala kamera a porovnávat to s videozáznamem, který byl během těchto pokusů pořízen.

Hlavní cíl diplomové práce byl splněn. Podařilo se vytvořit takový samořiditelný model, jenž byl schopen se zúčastnit soutěže NXP Cup. I přes to, že zde model neuspěl podle představ, řešení může být příslibem do budoucna. Kromě implementace navigačního algoritmu se také efektivně podařilo vyřešit sběr a zpětnou prezentaci dat. Jsou představeny také možnosti rozšíření řešení v podobě zdokonalení ovládání či rozšířené prezentace dat, ale také v podobě zlepšení samotné implementace a funkčnosti modelu. Díky vytvořenému průvodci je také možné na řešení jednoduše navázat a pokračovat tak v příštích ročnících soutěže již s funkčním základem, který je možné vylepšovat.

# Literatura

- [1] BALOGH, R. a LÁSZLO, M. *Model-Based Design of a Competition Car* online. 10. ledna 2017. Dostupné z: [https://doi.org/10.1007/978-3-319-42975-5\\_20](https://doi.org/10.1007/978-3-319-42975-5_20). [cit. 2025-11-01].
- [2] BEKR, J. *Spínané DC-DC měniče pro použití v elektronice* online. Duben 2023. Dostupné z:  
<https://www.elektrolab.eu/blog/spinane-dc-dc-menice-pro-pouziti-v-elektronice>. [cit. 2024-16-12].
- [3] BREJL, M. *Freescale Race Challenge 2011 – Další ročník soutěže samovídících autíček na autodráhu* online. 25. října 2010. Dostupné z:  
<https://vyvoj.hw.cz/theorie-a-praxe/freescale-race-challenge-2011-dalsi-rocnik-souteze-samovidicich-auticek-na-autodrahu>. [cit. 2025-10-01].
- [4] CAMPBELL, S. *Basics of UART Communication* online. 2016. Dostupné z:  
<https://www.circuitbasics.com/basics-uart-communication/>. [cit. 2025-06-04].
- [5] CHRÁSKA, J. *Architektury počítačů, procesory, desky* online. Duben 2023. Dostupné z: <https://sspvc.lixis.cz/hardware/arch/arch.html>. [cit. 2025-12-01].
- [6] DUDKA, M. *Detekce překážky pomocí IR?* online. Únor 2019. Dostupné z:  
[http://www.elektromys.eu/clanky/avr1\\_adc2/clanek.html](http://www.elektromys.eu/clanky/avr1_adc2/clanek.html). [cit. 2024-25-12].
- [7] ELECTRONIC, G. *Regulátor otáček pro bezkartáčové motory 30A* online. 2024. Dostupné z:  
[https://img.gme.cz/files/eshop\\_data/eshop\\_data/10/773-013/dsh.773-013.1.pdf](https://img.gme.cz/files/eshop_data/eshop_data/10/773-013/dsh.773-013.1.pdf). [cit. 2025-13-03].
- [8] FIGIEL, M. *Co je to signál PWM?* online. 5. května 2022. Dostupné z:  
<https://botland.cz/blog/co-je-to-signal-pwm/>. [cit. 2024-27-12].
- [9] GILLIS, A. S. *What is an embedded system?* online. Srpen 2024. Dostupné z:  
<https://www.techtarget.com/iotagenda/definition/embedded-system>. [cit. 2024-21-12].
- [10] HASHEMI POUR, C. *What is a microcontroller (MCU)?* online. Říjen 2024. Dostupné z: <https://www.techtarget.com/iotagenda/definition/microcontroller>. [cit. 2025-12-01].
- [11] HUERECA, A. *OpenSDAv2* online. Leden 2014. Dostupné z:  
<https://community.nxp.com/t5/Kinetis-Microcontrollers/OpenSDAv2/ta-p/1121569>. [cit. 2025-06-04].

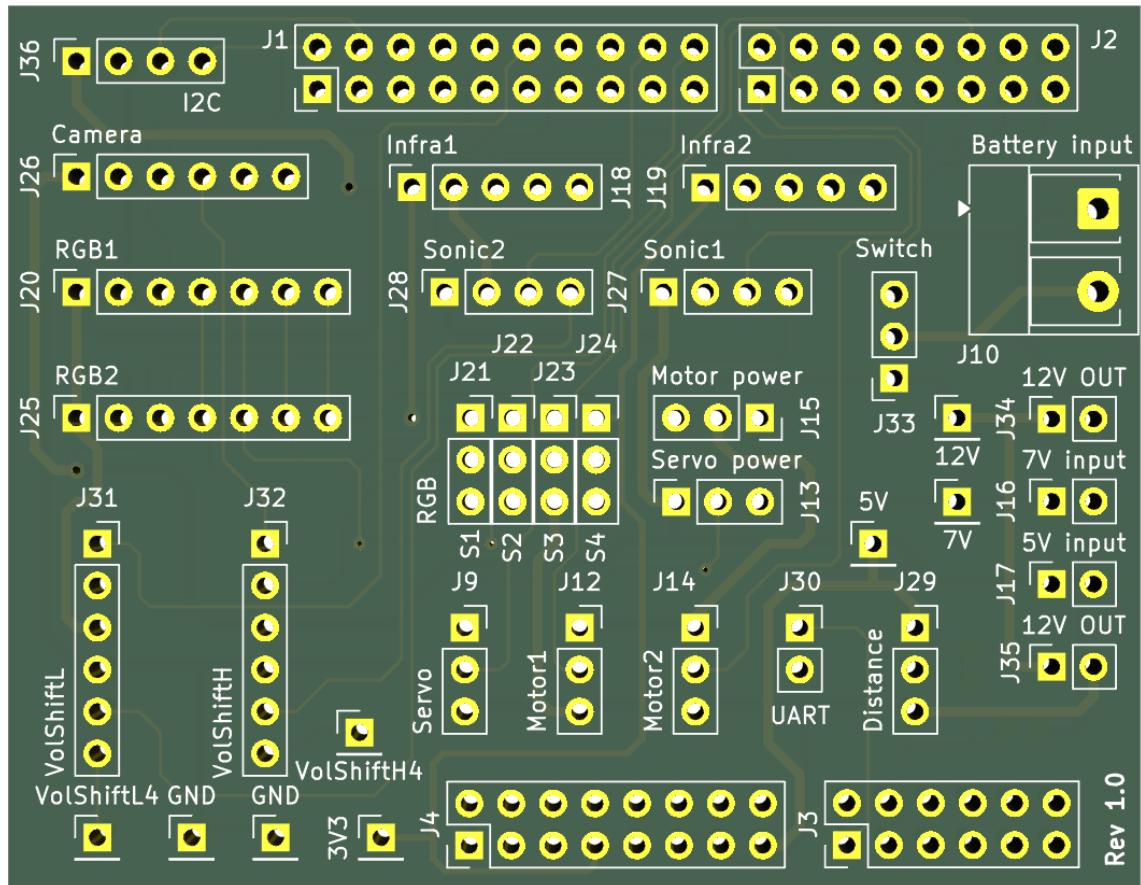
- [12] JOUAV.COM. *Brushed vs. Brushless DC Motor: What's the Difference?* Červenec 2024. Dostupné z: <https://www.jouav.com/blog/brushed-vs-brushless-motor.html>.
- [13] JÍŠA, K. *Vše o Lithium-polymerových akumulátorech* online. Listopad 2023. Dostupné z: <https://www.rcprofi.cz/poradna/vse-o-lithium-polymerovech-akumulatorech>. [cit. 2024-15-12].
- [14] LLAMAS, L. *Measuring distance accurately with Arduino and VL53L0X* online. 2024. Dostupné z: <https://www.luisllamas.es/en/arduino-distance-sensor-vl53l0x/>. [cit. 2025-05-02].
- [15] MAWAKANA, T. a DOLGOV, D. *Doubling down on Waymo One* online. 26. července 2023. Dostupné z: <https://waymo.com/blog/2023/07/doubling-down-on-waymo-one>. [cit. 2024-18-12].
- [16] MRÓZ, M. *Bldc motor – Co to je a jak funguje?* online. botland blog, březen 2023. Dostupné z: <https://botland.cz/blog/bldc-motor-co-to-je-a-jak-funguje/#Pouziti-motoru-BLDC>. [cit. 2024-11-04].
- [17] NXP.COM. *The NXP Cup Official Rules* online. Říjen 2024. Dostupné z: <https://www.nxp.com/docs/en/supporting-information/NXP-Cup-2025-rules.pdf>. [cit. 2024-11-11].
- [18] REICHL, J. *Rychlosť hmotného bodu* online. Listopad 2011. Dostupné z: <http://fyzika.jreichl.com/main.article/view/6-rychlosť-hmotneho-bodu>. [cit. 2025-13-01].
- [19] SANTOS, R. *Guide for TCS230/TCS3200 Color Sensor*. 2017. Dostupné z: <https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/>.
- [20] TME.EU. *Oblíbená komunikační rozhraní a protokoly* online. Červenec 2024. Dostupné z: <https://www.tme.eu/cz/news/library-articles/page/60826/oblibena-komunikacni-rozhrani-a-protokoly/#roznice>. [cit. 2025-06-04].
- [21] VOJÁČEK, A. *Magnetické senzory s Hallovým efektem – 1. princip* online. Listopad 2007. Dostupné z: <https://automatizace.hw.cz/magneticke-senzory-s-hallovym-efektem-1-princip>. [cit. 2025-28-03].
- [22] VOJÁČEK, A. *Ultrazvukové senzory přiblížení – funkce, provedení, použití* online. Červenec 2017. Dostupné z: <https://automatizace.hw.cz/ultrazvukove-senzory-priblizeni-funkce-provedeni-pouziti.html>. [cit. 2025-31-03].
- [23] ŠÍMA, V. *Měření pohybu auta na autodráze Carrera*. Brno, CZ, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/24774/>.
- [24] WINTERS, R. *How Servo Motors Work*. 2016. Dostupné z: <https://www.jameco.com/Jameco/workshop/Howitworks/how-servo-motors-work.html>.

- [25] WWW.ARROW.COM. *Analog-to-digital converters basics* online. Duben 2023.  
Dostupné z: <https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters>. [cit. 2025-07-04].
- [26] ČR, M. dopravy. *VIZE ROZVOJE AUTONOMNÍ MOBILITY* online. 2017.  
Dostupné z: [https://md.gov.cz/getattachment/Uzitecne-odkazy/Autonomni-mobilita/vize\\_rozvoje\\_autonomni\\_mobility.pdf.aspx](https://md.gov.cz/getattachment/Uzitecne-odkazy/Autonomni-mobilita/vize_rozvoje_autonomni_mobility.pdf.aspx). [cit. 2024-17-12].
- [27] ČR, M. dopravy. *Novela o autonomních vozidlech: krok k širšímu využití autonomních technologií na našich silnicích* online. 30. října 2024. Dostupné z: <https://md.gov.cz/Media/Media-a-tiskove-zpravy/Novela-o-autonomnich-vozidlech-krok-k-sirsimu-vy>. [cit. 2024-17-12].

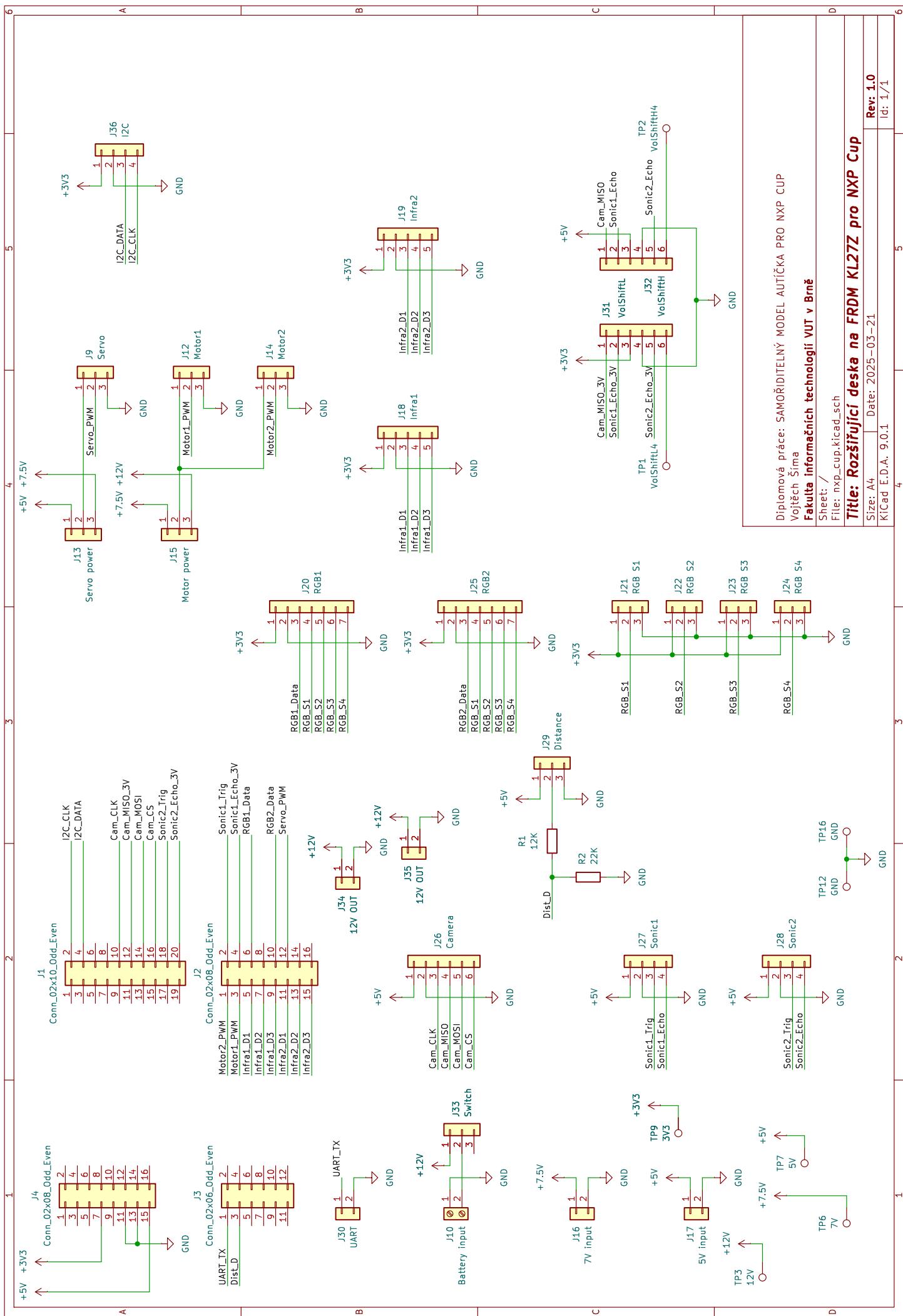
## Příloha A

# Schéma desky plošného spoje

V této příloze je vložen schématický návrh a další soubory týkající se rozšiřující desky plošného spoje, který byl vytvořen v programu KiCad<sup>1</sup>. Deska slouží k efektivnímu a spolehlivému propojení jednotlivých součástek s mikrokontrolérem a s napájením.



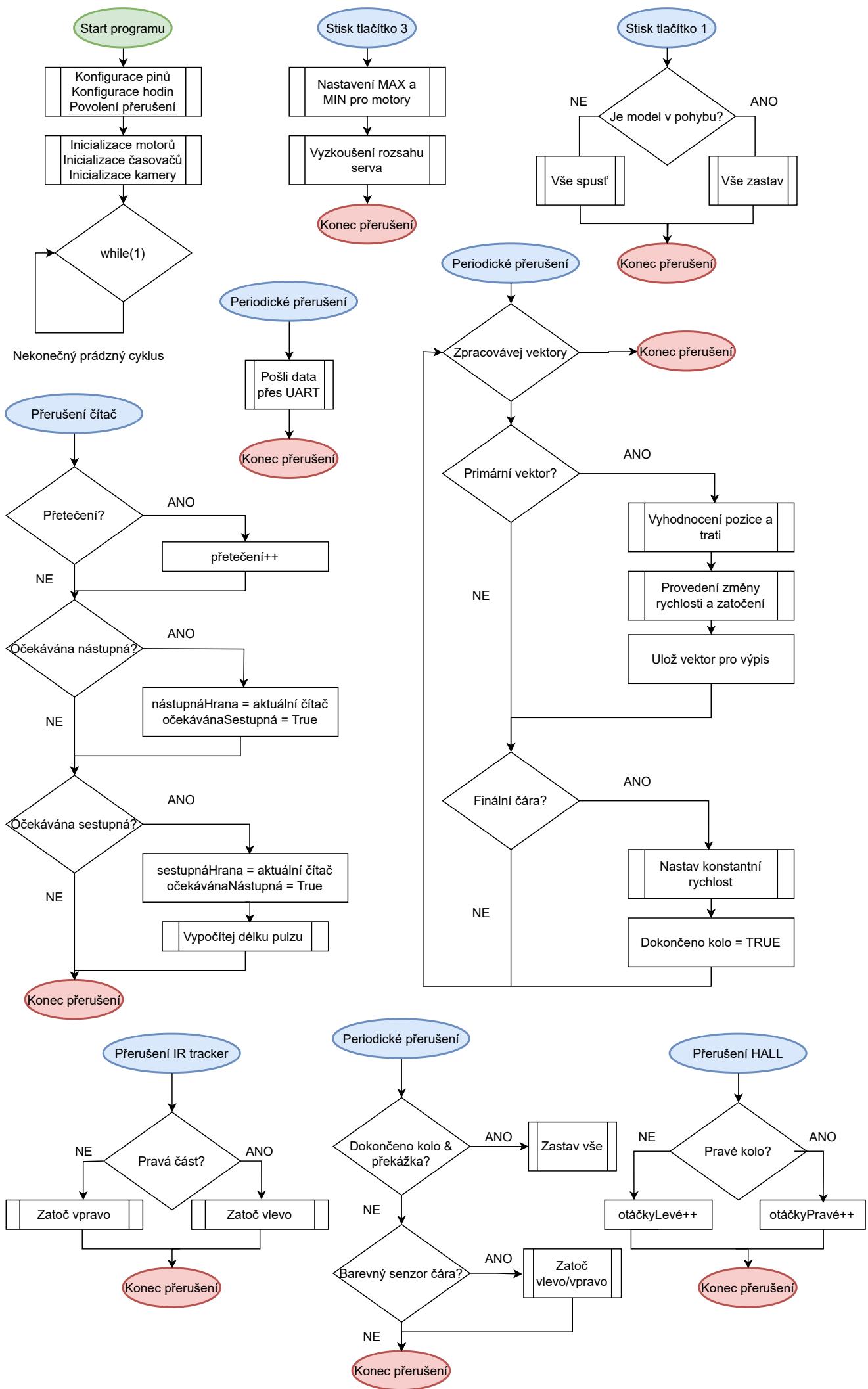
<sup>1</sup>KiCad – <https://www.kicad.org/>



## **Příloha B**

### **Vývojový diagram**

V této příloze je možné vidět celý vývojový diagram, který zjednodušeně představuje funkcionality celého programu rozdělenou do logických částí podle toho, jak bývají (nezávisle na sobě) vyvolávány za běhu programu. Jednotlivé části byly zobrazeny i popsány v textové části práce i s popisem.



## Příloha C

# Struktura přiloženého datového obsahu

```
xsimav01_dp
├── pdf
├── latex_overleaf
├── skripty
│   ├── data
│   ├── videa
│   ├── grafy
│   └── vygenerovane_obrazky
├── projekt_MCUXPresso
└── videa
```

```
├── rozsirujici_deska
├── platforma3D
├── pixy2_nastaveni
├── pruvodce_pro_NXPCup
└── readme.md
```

Přiložené datové médium obsahuje všechny potřebné soubory a dokumenty pro odevzdání diplomové práce. Zde je popsána jeho struktura a to, co jednotlivé složky či soubory obsahují.

- **pdf**
  - dvě verze výsledné práce v pdf (pro tisk a pro elektronické odevzdání)
  - zadání diplomové práce v pdf
- **latex\_overleaf**
  - obsahuje všechny zdrojové soubory, ze kterých dříve zmíněné pdf soubory mohou být vygenerovány
- **skripty**
  - obsahuje skripty, které byly použity pro zpracování naměřených dat či pro jejich následnou reprezentaci v podobě videa nebo grafů

- **data** obsahuje data naměřená při pohybu modelu které jsou vstupy skriptu
  - **videa** složka pro generovaná videa
  - **grafy** složka pro generované grafy
  - **vygenerovane\_obrazky** složka pro další generované obrázky
- **projekt\_MCUXPresso**
    - obsahuje celý projekt, který je možné otevřít v MCUXpresso IDE a skrze něj pak zkompilovaný kód může být nahrán do FRDM KL27Z
    - soubory s implementací jsou ve složce *source*
  - **videa**
    - obsahuje videa prezentující funkčnost modelu – jsou zde videa z testovacích pokusů na dráze i ze soutěžního pokusu
    - videa jsou zde samostatně a také spojená 1:1 s vygenerovanými videi, kde je zpětně ukázáno, jaké hodnoty zrovna zaznamenávala kamera a jiné senzory
  - **rozsirujici\_deska**
    - složka s projektem pro program KiCad, ve kterém byl vytvořen návrh rozšiřující desky
  - **platofrma3D**
    - složka obsahuje soubory pro 3D tisk přední platformy pro senzory
  - **pixy2\_nastaveni**
    - soubor s nastavením, které je potřeba nahrát do kamery Pixy 2 aby fungoval správně navigační algoritmus
  - **pruvodce\_pro\_NXPCup**
    - soubor obsahující průvodce pro úspěšné zprovoznění modelů a seznámení se s celkovou implementací, na kterou může být navázáno v příštích ročnících soutěže
  - **readme.md**
    - soubor popisující strukturu odevzdávaného média podobně jako tato příloha