VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

SYSTÉM PRO AUTONOMNÍ ŘÍZENÍ MODELU AUTÍČKA NA ZÁVODNÍ DRÁZE

SYSTEM FOR AUTONOMOUS NAVIGATION OF TOY CAR ON A RACE TRACK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. VIKTOR STEINGART**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. VÁCLAV ŠIMEK**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2019/2020

Zadání diplomové práce

Student: **Steingart Viktor, Bc.**
Program: Informační technologie Obor: Počítačové a vestavěné systémy
Název: **Systém pro autonomní řízení modelu autíčka na závodní dráze**
System for Autonomous Navigation of Toy Car on a Race Track
Kategorie: Vestavěné systémy
Zadání:

1. Nastudujte typické způsoby vytyčení závodní dráhy pro samostatný pohyb vozidla (modelářského autíčka) bez zásahu uživatele.
2. Seznamte se s principy detekce závodní dráhy dle bodu 1) zadání. Připravte krátkou studii, v níž přehledně shrnete získané poznatky.
3. Navrhněte koncepci systému autonomního řízení, který bude schopen regulovat rychlost pohybu vozidla s ohledem na rozpoznání klíčových úseků závodní dráhy (např. rovinku/zatáčku/křížení cest).
4. Zvolte vhodnou implementační platformu dle regulí soutěže NXP Cup. Na obvodové úrovni k ní navrhněte schéma zapojení podpůrného systému pro detekci překážek v dráze modelářského autíčka. Tento detekční systém realizujte v podobě desky plošných spojů a proveďte jeho oživení.
5. Implementujte obslužný firmware, který s využitím zvolených detekčních mechanismů zajistí autonomní navigaci vozidla po vytyčené závodní dráze.
6. Ověřte funkčnost vytvořeného řešení v reálných podmínkách.
7. Zhodnoťte dosažené výsledky a navrhněte případná rozšíření systému autonomního řízení.

Literatura:

- Dle pokynů vedoucího.

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

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 3. června 2020
Datum schválení: 23. dubna 2020

Abstrakt

Práce se zabývá návrhem a realizací vozidla pro závod autonomních modelů autíček s názvem NXP Cup. V práci je popsán celý proces od výběru platformy, přes návrh algoritmu, návrh a realizací desky plošných spojů pro systém detekce překážek, až po experimenty s pohybem.

Abstract

This thesis deals with the design and implementation of a self-driving model race car that is intended to attend an NXP Cup race. The work describes the selection of the platform, design process of algorithm, design process of printed circuit board for obstacle detection system, and various experiments with motion control.

Klíčová slova

vestavěný systém, autonomní vozidlo, model autíčka, závodní dráha, NXP Cup, senzor detekce vzdálenosti

Keywords

embedded system, autonomous car, model race car, race track, NXP Cup, distance detection sensor

Citace

STEINGART, Viktor. *Systém pro autonomní řízení modelu autíčka na závodní dráze*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Systém pro autonomní řízení modelu autíčka na závodní dráze

Prohlášení

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

.....
Viktor Steingart
3. června 2020

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Václavu Šimkovi za vedení mé diplomové práce, za všechny odborné rady, za pomoc s osazením desky plošného spoje a také za přívětivý přístup. Dále bych chtěl poděkovat rodině za podporu během celého studia, zvláště mojí manželce.

Obsah

Kapitola 1

Úvod

Téměř každý člověk již slyšel o autonomních vozidlech nebo alespoň o různých asistenčních systémech v moderních vozidlech. Za naprostou většinou těchto podpůrných systémů stojí vestavěný systém, který zpracovává data o okolí vozidla, rychlosti a směru jízdy, ale také třeba o stavu řidiče. A právě co nejvyšší stupeň autonomního řízení a zajištění bezpečnosti je úkolem, který zaměstnává mnoho inženýrů. Protože řízení reálného vozidla je úlohou velice komplexní, často se provádí experimenty na zmenšených modelech.

V této práci jsem se rozhodl projít si procesem návrhu a realizace modelu vozidla, které bude schopné samostatně jezdit po definované dráze a také detekovat překážky, které se mohou objevit v cestě. Cílem práce je tedy definovat požadavky na vozidlo, potažmo řídící systém, vybrat vhodné komponenty, z nichž bude vozidlo sestaveno, tento návrh zrealizovat a následně ověřit, zda výsledek mé práce splňuje vytyčené cíle.

Protože já osobně jsem velice soutěživý a porovnání mého výkonu s výkony ostatních dokáže být velkou motivací, ocenil jsem možnost spojit zadání práce s účastí v soutěži NXP Cup. Cílem bylo účastnit se kvalifikace, uspět a postoupit do finále pro oblast Evropy, Středního východu a Afriky.

Obsah technické zprávy je rozdělen do následujících kapitol: V druhé kapitole je pozornost věnována soutěži NXP Cup, jejím pravidlům a také je zde zmínka o dalších soutěžích. Třetí kapitola následně popisuje analýzu požadavků na řídící platformu a její výběr. Model vozidla bude poháněn elektromotory, proto jim je věnována čtvrtá kapitola. Pátá kapitola pojednává o způsobech snímání obrazu a detekce čar v obraze. Následující kapitola popisuje návrh a konstrukci celého modelu, jsou zde popsány jednotlivé komponenty a jejich zapojení. V sedmé kapitole je nastíněn návrh a implementace řídícího algoritmu. Následuje kapitola obsahující popis experimentů s pohybem. Celá práce je ukončena závěrem, ve kterém je zhodnocena práce a jsou zde navržena možná budoucí rozšíření.

Kapitola 2

Soutěž autonomních modelů vozidel NXP Cup

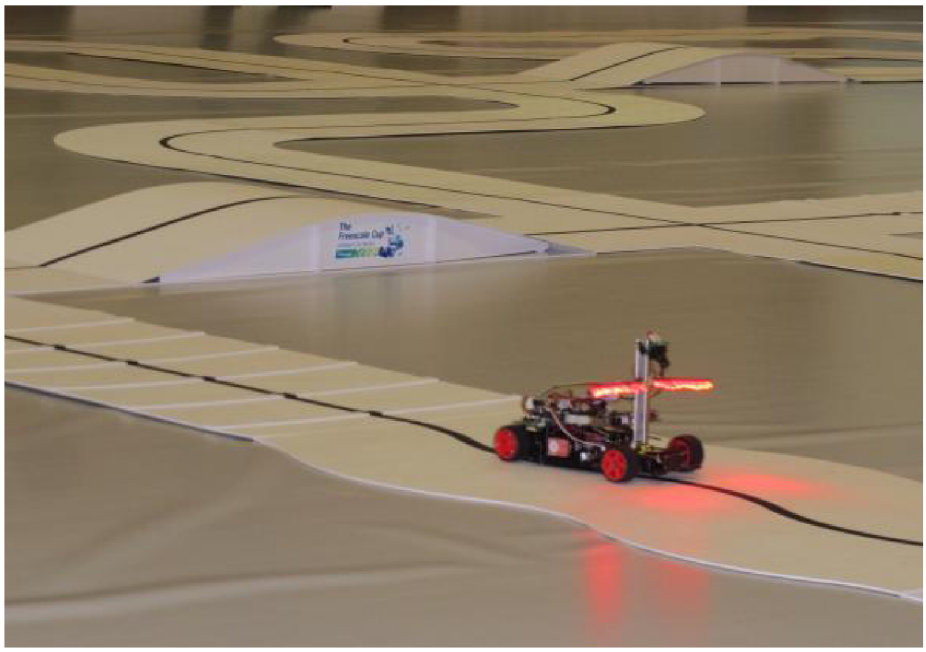
NXP Cup je celosvětová soutěž v autonomním řízení modelů vozidel pro studenty středních a vysokých škol. Hisotrie soutěže sahá do roku 2003, kdy se konala v Korei na Hanyang University soutěž Smart Car Race, které se účastnilo 80 týmů. Od této doby se soutěž rozšířila do Číny, Indie, Malajsie, Severní Ameriky. V roce 2012 se soutěž rozšířila i do Evropy. Od roku 2010 byla soutěž známa jako Freescale Cup, po ročníku soutěže 2015 se přejmenovala znovu na současný název. V roce 2019 se v rámci regionu EMEA (Evropa, Střední východ a Afrika) zúčastnilo soutěže 199 týmů. Tato kapitola vychází z oficiálních stránek soutěže [6], z pravidel [20] a také popisu dráhy [19].

2.1 Pravidla a podmínky soutěže

Pravidla soutěže se každoročně mění a postupně upravují. Předchozí ročníky bylo přesně určeno, z jakého hardware musí účastníci soutěže vycházet. V ročníku 2019 / 2020 v tomto ohledu proběhla změna a bylo povoleno použití libovolného hardwaru, který splňuje pravidla. Nejdůležitějšími omezeními na hardware bylo využít pro ovládání vozidla a vyhodnocování dráhy jednotku obsahující mikrokontroléry nebo mikroprocesory od firmy NXP. Tato podmínka se týká i kamer, které v případě, že dělá nějaké operace s obrazem, také musí byt osazeny čipem od NXP. Autíčko musí být osazeno kamerou pro primární navigaci. Použít kameru jiného výrobce, než NXP je povoleno pouze, pokud je kamera připojena přímo na řídící desku.

Autíčko musí být autonomní a nesmí být ovládáno na dálku a proto nesmí být během závodu osazeno žádným modulem pro bezdrátovou komunikaci. Vozidlo může být vybaveno až 4 koly a maximálně dvěma motory (jsou povoleny komutátorové, i bezkartáčové motory). Pro ovládání motorů je možné použít komerční regulátory (Electronic Speed Control, ESC). Účastníci soutěže smějí k modelu přidávat další senzory, které potřebují ke zvládnutí soutěžních úkolů.

V dřívějších letech byla dráha tvořena černou čárou uprostřed bílé dráhy (Obrázek 2.1). Cílem soutěže tedy bylo projet dráhu podle středové čáry za co nejrychlejší čas. V ročníku 2019 / 2020 byla stejně jako předchozí roky dráha ohraničena dvěma černými čárami, avšak bez středové čáry.



Obrázek 2.1: EMEA Freescale cup 2013. Převzato z [31]

Dráha může mít libovolný tvar, je však limitována tím, že musí být sestavena z předem definovaných dílů, které do sebe přesně zapadají. Základními díly jsou rovinka, zatáčka a křižovatka.

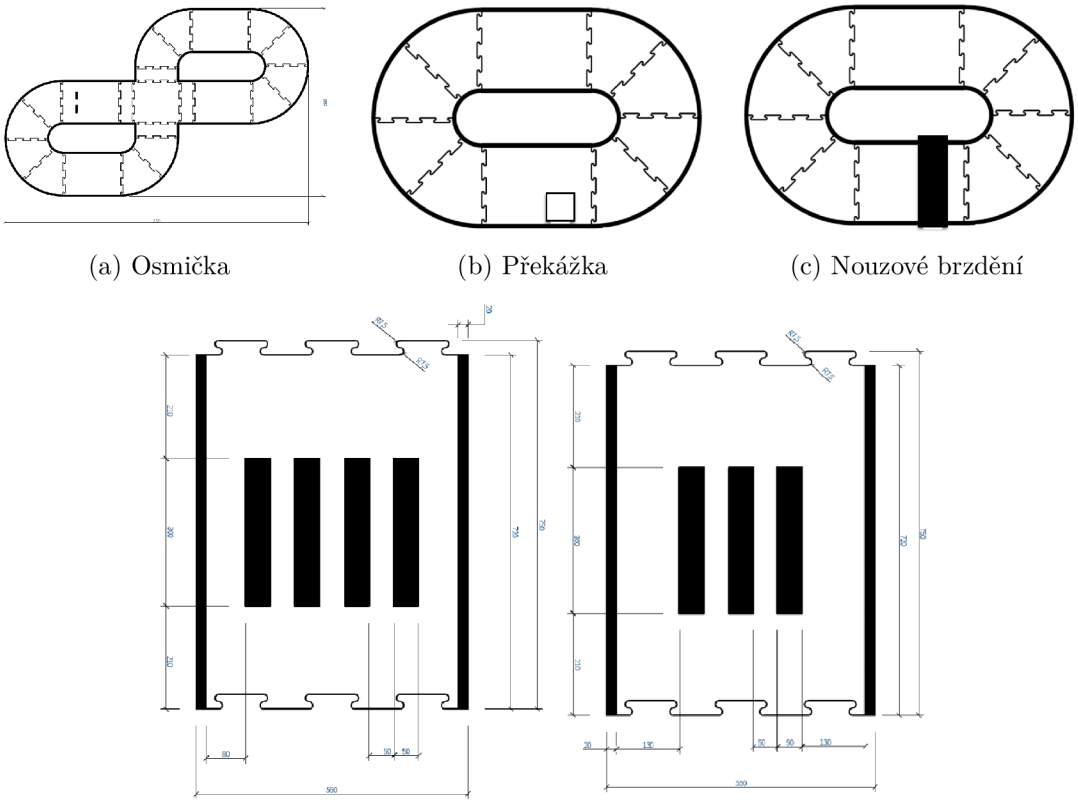
Soutěž se skládá z hlavního závodu a doplňkových disciplín. Za každou disciplínu je možné získat bodový zisk, celkový součet bodů pak určí vítěze.

Hlavní závodní disciplína je závod na čas, který probíhá na dráze, jejíž tvar je neznámý až do samotného závodu. Dráha je stejná pro všechny týmy a nezávisí na vzdělávacím stupni jednotlivých členů týmu ani na použitém vozidle. Každý tým má 3 pokusy pro zvládnutí jednoho kola na dráze a je zaznamenán čas prvního úspěšného pokusu. V případě neúspěchu má jeden člen týmu 2 minuty na úpravy vozidla. Za tuto disciplínu lze získat až 650 bodů za první místo. Každé další umístění až do desátého místa získá vždy o 50 bodů méně.

Ostatní disciplíny jsou volitelné, lze za ně však získat bonusové body. První z nich je Osmička (obrázek 2.2a).V této disciplíně má vozidlo za úkol objet co nejvíce kol během časového limitu 60 vteřin. Tato disciplína je zaměřena na přesnost, což znamená, že z dráhy nesmí vyjet žádné kolo, ani jiná část vozidla.

Další dvě disciplíny jsou zaměřeny na detekci překážek na dráze a za každou lze získat 150 bodů. Obě probíhají na malé dráze, která je složena například ze dvou rovných dílů a osmi zatáček. V případě překážky (obrázek 2.2b) je cílem detekovat polystyrenovou krychli, která může být umístěna kdekoliv na dráze a vyhnout se jí. Žádná část autíčka nesmí vyjet z dráhy, ale ani se dotknout překážky. V případě nouzového brzdění (obrázek 2.2c) je po ujetí dvou kol na oválné dráze umístěna rozhodčím černá překážka, která zcela překryje šířku dráhy. Vozidlo musí překážku detekovat a zastavit před ní.

Poslední disciplínou je měřený úsek, který probíhá opět na malé dráze, například 2 rovné úseky v řadě, doplněné zatáčkami. Jsou zde díly, obsahující 3 a 4 pruhy (obrázek 2.2d). Po detekování 4 pruhů musí vozidlo znatelně zpomalit na přibližně polovinu původní rychlosti. Jakmile detekuje 3 pruhy, opět zrychlí na původní rychlost. Za úspěšné splnění získá tým opět 150 bodů.



(a) Osmička (b) Překážka (c) Nouzové brzdění

(d) Měřený úsek

Obrázek 2.2: Typy dráhy v soutěži NXP Cup. Převzato z [19]

2.2 Podobné soutěže

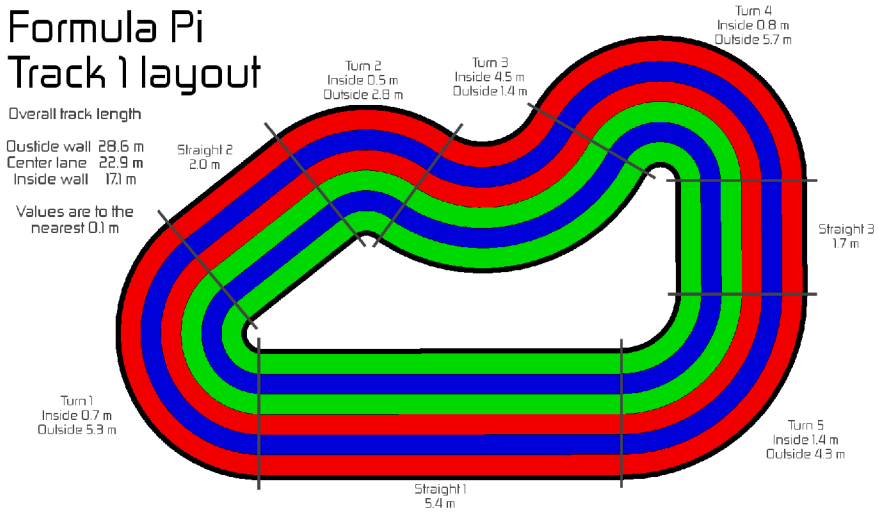
Každý závod autonomních vozidel má vlastní pravidla a vlastní způsob vytyčení závodní dráhy. Přesto lze vysledovat několik společných vlastností.

První velkou kategorií jsou závodní dráhy které jsou zaznačeny barvou, barevnou lepící páskou, případně i jiným způsobem na podkladu. Například soutěž AWS Deepracer [1] , kterou pořádá společnost Amazon, má tmavý podklad dráhy a bílé ohraničení. V příručce pro vývojáře k této soutěži je uvedeno, že vhodný materiál pro domácí výrobu dráhy je jakýkoliv tvrdý tmavý materiál (dřevo, koberec, beton, asfalt . . .) a pro vyznačení dráhy je vhodná bílá lepící páska, která však nesmí být lesklá [25]. Dráha vytištěná na tenký vinyl používaná na závodní události je na obrázku 2.3.

Podobný způsob zaznačení dráhy používá také soutěž NVIDIA's DIY Autonomous Car Race [5]. Kromě rozdílného tvaru dráhy však přidává tato soutěž ještě kužely umístěné podél dráhy v zatáčkách. V této soutěži není nijak penalizováno vyjetí z dráhy za bílou barvu, je však penalizován dotyk kuželu. V případě, že by vozidlo v zatáčce některý z kuželů vynechalo, je ze závodu diskvalifikováno. Prakticky stejná pravidla jako soutěž zaštítěná společností NVIDIA má i soutěž DIY Robocar Races pro vozidla v měřítku 1:10 [27]. DIY Robocar Races má však ještě jednu soutěžní kategorii – modely v měřítku 1:16 [28]. Ty závodí na dráze, která je složena z podélných barevných pruhů. Podobné barevné pruhy využívá pro vytyčení dráhy i soutěž Formula Pi [3], kde je dráha široká 1,83 metru, obsa-

Obrázek 2.3: AWS Deepracer. Převzato z [1]

huje 6 barevných pásů v barevném pořadí červená, modrá, červená, zelená, modrá a zelená (Obrázek . 2.4)



Obrázek 2.4: Formula Pi. Převzato z [3]

Některé soutěže přidávají ještě doprostřed dráhy vodící čáru, která může usnadnit navigaci po dráze. Zástupcem tohoto typu dráhy může být soutěž Freescale Cup (nyní NXP Cup) v letech 2010 - 2014.

Zcela odlišný typ autonomně řízených vozidel představuje Freescale Race Challenge [29], kdy se závodilo na autodráze se středovým vedením, kdy namísto ovládání výkonu pomocí ovladače o rychlosti autíčka rozhodovala integrovaná řídící deska v autíčku.

Kapitola 3

Analýza požadavků a výběr řídící platformy

V této kapitole bych rád provedl analýzu požadavků na řídící platformu a její následný výběr. Z předchozí kapitoly lze odvodit následující specifikaci požadavků na řídící platformu:

- platforma má za cíl vyhodnotit data ze senzorů a na jejich základě upravit rychlost a směr pohybu
- platforma bude umístěna na robotickém podvozku, bude potřeba do ní připojit kameru a podpůrný systém pro detekci překážek a také bude potřeba připojit motory, které zajistí pohyb, a servo které zajistí změnu směru pohybu
- uživatel do procesu řízení nesmí nijak zasahovat, řízení probíhá automaticky

3.1 Požadavky plynoucí ze soutěže

Výše uvedené požadavky lze dále rozvést tak, aby byly splněny všechny požadavky soutěže [20]. Především je nutné dodržet podmínku využít řídící platformu založenou na mikrokontroléru nebo mikročipu od NXP, případně využít desku, která je navržena firmou NXP.

Dále z pravidel soutěže vyplývá, že kamera musí být připojena napřímo přes sběrnici SPI, případně lze využít kameru osazenou s další elektronikou na separátní desce, vyrobené společností NXP. S ohledem na výkonnost celého systému se jeví jako vhodné použít kamerku na separátní desce osazenou vlastní čipem pro zpracování obrazu. U kamery lze očekávat připojení přes sběrnici I^2C nebo přes sběrnici SPI. Dalším zařízením bude podpůrný systém pro detekci překážek. Tento systém bude navržen tak, aby komunikoval přes sběrnici I^2C.

Pravidla soutěže umožňují pouze jednu baterii pro napájení celého vozidla včetně veškeré elektroniky. LiPo baterie jsou povoleny maximálně dvoučlánkové (napájecí napětí 7,4V) o kapacitě 5500mAh, baterie NiCD, NiMH nebo Li-ION pak o kapacitě maximálně 5300mAh.

3.2 Specifikace cílové platformy

Prvním klíčovým parametrem pro výběr platformy je potřebný počet vstupů a výstupů. Jak bylo zmíněno v minulé podkapitole, bude potřeba hardwarová podpora synchronních

sériových rozhraní SPI a I^2C a jejich vyvedení na piny. Dále bude potřeba pro ovládání rychlosti a směru pohybu řídit motory. Pro ovládání motorů (pro každé kolo separátní) a servo motoru řízení bude potřeba generovat signál s různým poměrem logické 0 a logické 1, proto dalším požadavkem je minimálně 3x vstupně-výstupní pin s hardwarovou podporou PWM. Lze očekávat potřebu několikanásobného nahrávání různých verzí firmware, proto je nutné, aby byla platforma vybavena podporou asynchronního sériového rozhraní, případně přímo integrovanou podporou USB.

Dalším důležitým aspektem je paměťová náročnost. Protože lze očekávat, že během vývoje bude nutné několikrát nahradit firmware a současně lze považovat 2MB jako bezpečnou hranici i pro rozsáhlý firmware, jako další podmínku lze položit přepisovatelnou paměť pro program o velikosti alespoň 2MB. Současně bude vhodné mít možnost ukládat data ze senzorů pro pozdější analýzu. Jednou možností je přístupná paměť například přes rozhraní USB, druhou využití řadiče pro SD kartu, kterou bude možné jednoduše vyjmout a prohlížet data na počítači.

Spotřeba řídící platformy pravděpodobně nebude kritická, protože součástí celého systému budou i motory, které svojí spotřebou o několik řádů překročí spotřebu mikrokontroléru. Přesto s ohledem na fakt, že celý systém bude napájen z baterie, nebude nižší spotřeba na škodu.

3.3 Výběr platformy

S ohledem na co nejjednodušší vývoj by bylo vhodné, aby zvolený mikrokontrolér již byl součástí platformy, obsahující další periferie, a nebylo tak nutné navrhovat celou desku od začátku. S ohledem na požadavek výrobce čipu jsem se zabýval výhradně deskami vyrobenými a obsahujícími čipy firmy NXP.

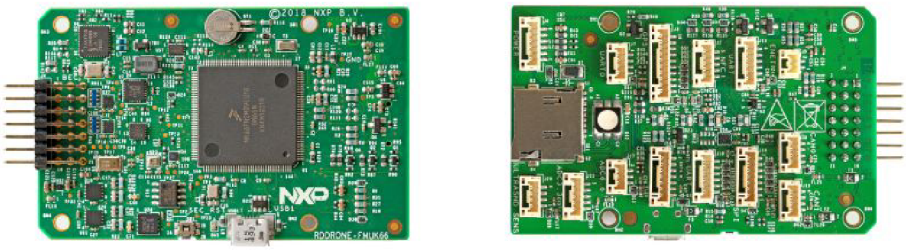
První deskou, kterou bylo možné využít, je vývojová platforma FRDM-KL25Z [22]. Tato platforma je osazena mikrokontrolérem MKL25Z128VLK4 s jádrem Arm® Cortex®-M0+. Mikrokontrolér pracuje na frekvenci 48MHz, obsahuje 128KB flash paměti a 16KB SRAM paměti. Obsahuje full speed USB řadič, rozhraní SPI i I^2C. Další možností je poté deska FRDM-KL46Z [23] s mikrokontrolérem MKL46Z256VLL4MCU, která oproti desce FRDM-KL25Z obsahuje flash paměti i SRAM paměť o dvojnásobné kapacitě. Přidává navíc sedmi-segmentový displej schopný zobrazit 4 číslice, dvě tlačítka a magnetometr.

Druhou možností z portfolia společnosti NXP Cup je platforma RDDRONE-FMUK66. Platforma je založena na mikrokontroléru Kientis K66 [18] založeném na jádru Arm Cortex-M4F. Mikrokontrolér pracuje na frekvenci 180 MHz, obsahuje flash paměť o velikosti 2 MB a 256KB SRAM paměti. Současně má tato platforma velmi nízkou spotřebu, nabízí širokou škálu rozhraní (High-speed USB, UART, I^2S, 3xSPI, 4x I^2S).

Tento mikrokontrolér cílí na aplikace s nízkou spotřebou a větší paměťovou náročností. Jádro běžící na frekvenci 180MHz poskytuje dostatečný výkon pro běžné výpočetní úlohy a 2 MB flash paměti zajišťují dostatečný prostor pro uložení programu.

Platformu RDDRONE-FMUK66 lze vyhledat také pod obchodním názvem PX4 Robotic Drone FMU. Jak již z názvu vyplývá, tato deska je primárně navržena pro řízení letu dronů, lze ji však bez problémů využít i pro řízení pozemních vozidel.

Mezi velké výhody této desky patří fakt, že na ní funguje open-source software PX4 pro řízení bezpilotních dronů i dalších bezpilotních vozidel. Díky tomuto projektu je možné využít mnoho nástrojů, které mohou usnadnit například kalibraci senzorů, či testování funkčnosti periferií. PX4 je součástí neziskové organizace Dronecoode, která je spravována neziskovým technologickým konsorciem Linux Foundation. Platforma PX4 je vysoce mo-



(a) Spodní strana (b) vrchní strana

Obrázek 3.1: PX4 Robotic Drone FMU. Převzato z [24]

dulární, lze tedy libovolně rozšiřovat hardware a software, aniž by rozšířený systém ztrácel robustnost, nebo výkon.

Především s ohledem na výkon a kapacitu paměti jsem se nakonec rozhodl použít právě platformu RDDRONE-FMUK66. Tyto dva parametry pro mne byly klíčové především z důvodu, že v této fázi jsem měl pouze velmi nejasné představy o algoritmu, který bude muset tato platforma zvládnout, a nebyl jsem schopný odhadnout ani velikost programu, který bude nutné do desky nahrát. Zvolil jsem tedy „jistější“ výkonnější verzi.

Kapitola 4

Problematika řízení elektromotorů

V předchozí kapitole byla vybrána řídící platforma. Tato platforma přímo počítá s řízením různých druhů vozidel či letounů. V této kapitole budou rozebrány základní fyzikální principy fungování elektromotorů, typy motorů používané v modelářství a robotice a problematika řízení jednotlivých typů motorů. Kapitola vychází z knih Electric Motors and Drives: Fundamentals, Types and Applications [33], Electric motors and control systems [34], Elektrické motory a pohony [35] a také z repetitoria které vyšlo v časopise Elektro (ročník 2013, čísla 1-7) [9] [10] [11] [12] [13] [14] [15].

4.1 Stejnosměrné motory

Stejnosměrný motor je svojí konstrukcí nejjednodušším typem elektromotoru. Princip funkce vychází ze vztahu (4.1), který obecně platí pro všechny elektomotory. Tento vztah nám říká, že pokud se pohybuje vodič o délce l[m], kterým protéká proud I[A], v homogenním magnetickém poli B[T] v kolmém směru na siločáry, působí na vodič síla F[N]. V případě stejnosměrného motoru se poté využívá toho, že vinutí na rotoru je přitahováno magnetickým polem statoru. Ve správný okamžik je však potřeba obrátit směr proudu, aby se pohyb rotoru nezastavil.

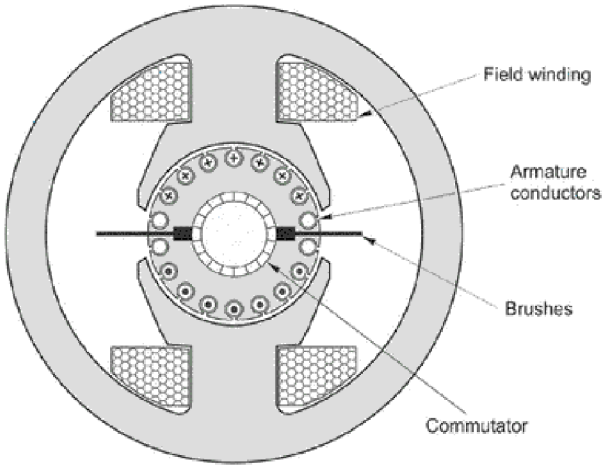
$$F = B \cdot I \cdot l \tag{4.1}$$

Základní struktura stejnosměrného motoru je na obrázku 4.1. Motor se skládá ze dvou oddělených obvodů. Menší z obvodů je připojen na budicí vinutí (Field winding), které vytváří magnetomotorickou sílu pro magnetický tok ve vzduchové mezeře mezi jednotlivými póly. V klidovém stavu je všechna vstupní energie přeměněna na tepelnou energii.

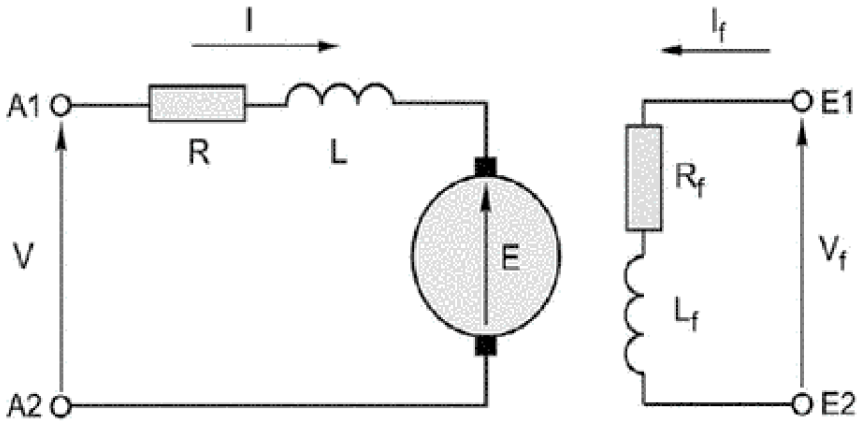
Druhý obvod přivádí elektrický proud vytvářející točivý moment ke kartáčům (brushes), které následně tento proud připojí přes komutátor (commutator) na vinutí kotvy (armature winding). Výše popisovaná změna směru toku proudu vinutím kotvy je se děje právě diký komutátoru a kartáčům. Komutátor má po obvodu obecně několik kontaktů, kdy postupným natáčením komutátoru vůči kartáčům dochází k uzavíraní a rozpojování elektrického obvodu. V závislosti na polaritě jednotlivých kontaktů komutátoru se mění směr proudu. Elektrické schéma zapojení je na obrázku 4.2.

Největší nevýhodou komutátorového motoru je fakt, že při pohybu motorku se kartáče komutátoru opotřebovávají. U větších motorů bývá zpravidla možné kartáče vyměnit, u menších – například modelářských – motorků však výměna nebývá možná.

Výhodou tohoto typu motoru však je jeho cena, která je v porovnání s jinými typy motorů nízká. Mezi výhody tohoto typu motoru patří také fakt, že změnou polarity napájecího



Obrázek 4.1: Komutátorový motor. Převzato z [33]



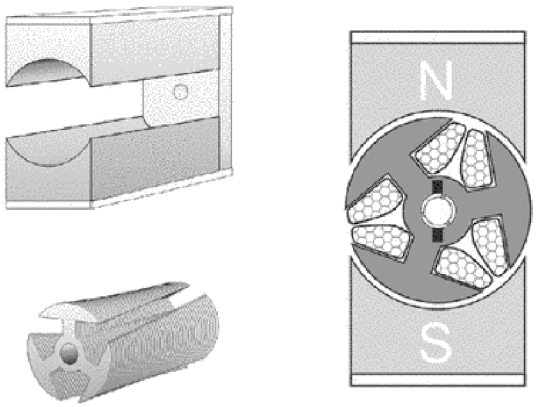
Obrázek 4.2: Elektrické schéma stejnosměrného motoru. Převzato z [33]

napětí lze dosáhnout změny směru rotace. Další užitečnou vlastností je, že zvyšování napětí zvyšuje výkon motoru a díky tomuto může být ovládací elektronika velice jednoduchá.

V modelářství a hračkách se využívá odlišné konstrukce, než je výše popsána, především s ohledem na cenu. Protože tyto motorky pracují vy vysokých otáčkách (v případě nižších otáček se využívá zpřevodování), není zde potřeba řešit plynulý točivý moment. Z tohoto důvodu se pro rotor využívá malý počet cívek (typicky 3-5), což způsobí jednodušší komutátor, který je také levnější na výrobu. Stator je tvořen radiálně zmagnetizovaným keramickým magnetem v kovovém pouzdře pro uzavření magnetického okruhu. Schéma modelářského stejnosměrného motorku je na obrázku 4.3.

4.2 Indukční motory

Podobně jako stejnosměrné motory i indukční motory fungují na principu vzájemného působení magnetického pole na rotoru a statoru. Rozdíl je však ve způsobu napájení – indukční motory využívají pro napájení střídavý proud. Zatímco u velkých výkonných motorů se



Obrázek 4.3: Motor využívaný v modelech. Převzato z [33]

setkáváme s třífázovým proudem, malé motory mohou být i jednofázové. Dalším velkým rozdílem je, že magnetické pole v indukčních motorech se pohybuje relativně vůči statoru, zatímco u stejnosměrných motorů je magnetické pole statické.

Konstrukce rotoru může být dvojího typu – klecový rotor a vinutý rotor. Klecový rotor se skládá z tyčí odlitých do rotorových drážek a na konci spojených vodivými kruhy. Vinutý rotor je tvořen z izolovaných vodičů v izolovaných drážkách, tyto vodiče jsou zapojeny jako třífázové se zapojením do hvězdy. Na vývody rotorového vinutí se připojují (za pomoci kartáčů a tří oddělených kroužků) rezistory, za jichž pomoci lze regulovat otáčky motoru. Otáčky je také možné měnit regulátorem, který je připojen na vývody rotorového vinutí.

4.3 Motory s permanentním magnetem a synchronní motory

Pro tuto skupinu motorů se využívá hned několik různých názvů, které však označují v podstatě stejné zařízení s drobnými úpravami. Jedná se o synchronní stroje, synchronní stroje s permanentními magnety, střídavé bezkartáčové motory, stejnosměrné bezkartáčové motory a servo motory s permanentními magnety.

V synchronním motoru je vinutí statoru v principu stejné, jako v indukčním motoru a platí i zde, že po připojení na třífázové napájení vzniká rotující magnetické pole. Rozdíl však nastává u rotoru, kde místo válcovitého rotoru s vinutím kotvy, které se automaticky přizpůsobuje počtu pólů na statoru, je vinutí rotoru buzeno stejnosměrným proudem, případně je nahrazeno permanentními magnety (odpadá nutnost dvojího napájení - stejnosměrné pro rotor a střídavé pro stator), jejichž počet odpovídá počtu pólů statoru. Rotor je díky tomu synchronizován s otáčejícím se magnetickým polem statoru a má shodnou rychlost i směr, jako magnetické pole.

Pro změnu rychlosti otáčení je třeba měnit frekvenci třífázového signálu, což se děje v regulátoru. Výhodou tohoto motoru je, že díky synchronnímu pohybu rotoru a magnetického pole lze přesně a jednoduše řídit rychlost motoru úpravou generovaného třífázového signálu.

Tohoto typu motoru se využívá i v modelářství, kde je například v případě dronu žádoucí, aby se rotor a na něm připojené vrtule pohybovaly přesně stanovenou rychlostí. V oblasti modelářství je ustálený název BLDC motor (Brushless Direct Current). Pro řízení se nejčastěji využívá elektronický regulátor rychlosti (Electronic Speed Control - ESC). Toto zařízení je připojeno k napájení a jako vstup přijímá PWM signál na frekvenci 50Hz,

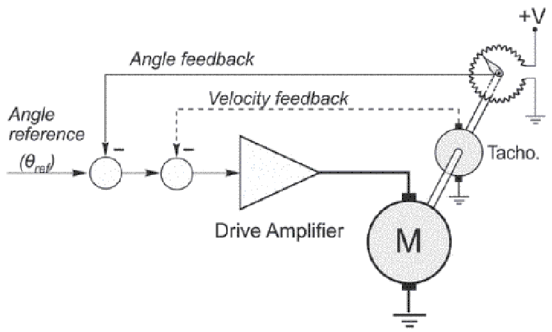
kde se šířka pulzu pohybuje mezi 1 - 2 ms. Nejčastěji se lze setkat s konfigurací, kdy 1500μs znamená nulovou rychlost, 1000μs maximální rychlost jedním směrem, 2000μs druhým směrem. Tyto hodnoty je však potřeba vyčíst v dokumentaci ke konkrétnímu ESC.

4.4 Stejnosměrné servo motory

Jedná se o podskupinu stejnosměrných motorů, které jsou určeny pro vysoký výkon a obsahují zpětnou vazbu tykající se točivého momentu, rychlosti a často také pozice. Tyto motory se také často využívají tam, kde je potřeba prudké zrychlení a zpomalení. Servo motory bývají malé, avšak jejich odpor vinutí kotvy je velký a zkratový proud při plném zatížení několikrát překoná jmenovitý proud motoru.

Z důvodu maximálního zvětšení akcelerace musí být setrvačnost rotoru minimální a proto se volí konstrukce, kdy se pohybuje pouze nezbytný elektrický obvod, magnetická část je statická.

Při využití v aplikacích, které vyžadují přesnou kontrolu pozice se využívá zapojení, kde se kromě vstupního napětí uvažuje aktuální natočení motoru a aktuální rychlost. Schéma takového zapojení je na obrázku 4.4.



Obrázek 4.4: Schéma servo motoru s kontrolou pozice. Převzato z [33]

Servo motorů se využívá v modelářství k řízení natočení kol, nebo například směrovek u letadel. Tyto servo motory jsou vybaveny převodovkou, zpětnovazebním potenciometrem a řídící elektronikou. Nejčastěji se osa modelářského serva dokáže pohybovat v rozsahu 180 stupňů. Podobně jako u řízení modelářských BLDC motorků pro pohon se i zde využívá PWM signálu na frekvenci 50Hz, rozsah 1-2 ms jsou lineárně mapované na rozsah pohybu.

4.5 Motory použité v modelu autíčka

Pro pohon byly využity dva BLDC motory s označením A2212/15T [17]. Parametry motoru jsou uvedeny v tabulce 4.1. Fotografie motoru je na obrázku 4.5. Pro řízení motorků jsou využity regulátory BLS30A.

Pro pohyb s táhly řízení je použit servo motor MG996R [4]. Tento motor lze napájet napětím 5V nebo 6V. Při napájení 6V je servo motor schopný vyvinout kroutící moment 11kg/cm. Servo MG996R je na obrázku 4.6.

Napěťová konstanta	930
Napájecí napětí	7-12V
Příkon	135W
Účinnost	80%
Proud v zátěži	4-10A
Proud v nečinosti	0,5A

Tabulka 4.1: Parametry motoru A2212/15T

Obrázek 4.5: Motor A2212/15T. Převzato z [16]

Obrázek 4.6: Motor A2212/15T. Převzato z [4]

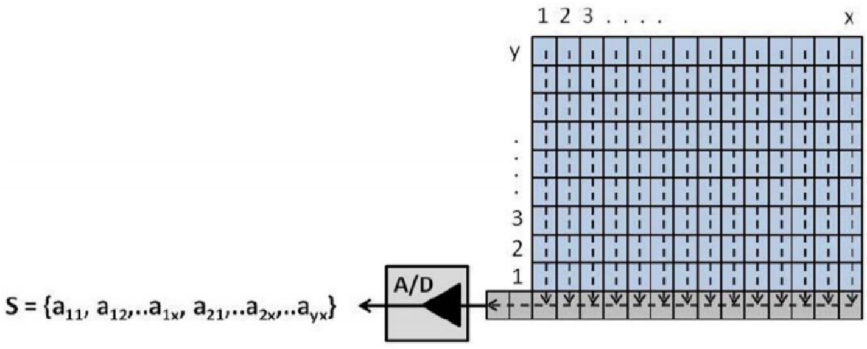
Kapitola 5

Zpracování obrazu a detekce objektu

5.1 Snímání obrazu

Pro snímání obrazu se používají nejčastěji dvě technologie: CCD seznory a CMOS senzory. Následující kapitola je založena na článku TWO REVOLUTIONARY OPTICAL TECHNOLOGIES [8] a na knize Zpracování obrazu a algoritmy v C# [30].

Senzory typu CCD (Charge Coupled Device) využívají technolgie MOS (metal-oxide semiconductor). Povrchová struktura snímače je tvořena z velkého počtu světlocitlivých buněk uspořádaných do řádků a sloupců. Každá z buněk obsahuje MOS kondenzátor, který se nabíjí elektrony, které vznikají díky fotoelektrickému efektu při dopadu fotonů. Množství elektronů je přímo úměrné množství světla, které dopadá na konkrétní buňku. Náboj z kondenzátorů je následně převeden do posuvných registrů, za jejichž pomoci je vždy jeden řádek posouván postupně do převodníku, který převede náboj na digitální hodnotu, která se dále zpracovává. Po přečtení celého řádku je posunut do čtecího registru další řádek a čtení se opakuje. Princip přesunu je na obrázku 5.1.



Obrázek 5.1: Princip vyčítání CCD čipu. Převzato z [8]

Druhým, v současnosti velice rozšířeným typem senzorů jsou senzory typu CMOS. Tyto senzory mají podobné buňky jako snímače CCD, s podstatným rozdílem ve snímání hodnoty náboje. Zatímco v CCD čipu se hodnota náboje posouvá do jednoho A/D převodníku, v CMOS technologii je velikost náboje převáděna na napětí přímo v buňce. Následně je hodnota napětí přenášena soustavou multiplexorů na okraj snímače. Z tohoto konstrukčního

rozdílu plyne fakt, že zatímco v případě CCD snímače je expozice provedena ve všech buňkách snímače současně a následně postupně přenášena, v případě CMOS obrazových snímačů je nutné počátek expozice a snímání jednotlivých buněk provádět postupně. Toto může u rychle se pohybujících objektů způsobit zkreslení. Naopak výhodou tohoto senzoru je, že lze integrovat veškerou řídící elektroniku na jeden čip. Dále má CMOS technologie v porovnání s CCD podstatně menší spotřebu.

5.2 Detekce čar v obraze

Nejjednodušším způsobem nalezení čáry v obraze, avšak ne výpočetně ideálním, je vytvoření seznamu všech dvojic bodů (předpokládá se, že obraz je po prahování, obsahuje tedy pouze hodnoty 0 a 1), které mohou potenciálně tvořit úsečku. Následně by se ověřovalo, že ostatní body leží na této úsečce (jsou dostatečně blízko). Složitost takového přístupu je v nejlepším případě řádu $O(n^3)$, kde n je počet bodů.

Existuje však časově mnohem méně náročné řešení – použití Houghovy transformace. Této transformace lze využít pro hledání téměř libovolných tvarů, které lze parametrizovat. Princip bude demonstrován na přímce v rovině (x, y) danou rovnicí 5.1, kterou lze přepsat do tvaru v rovnici 5.2

$$y = ax + b \tag{5.1}$$

$$b = -ax + y \tag{5.2}$$

Pro další postup uvažujeme rovinu (a, b), kde na vodorovné ose vynášíme parametr a a na svislé parametr b. Díky transformaci tak přímce $y = ax + b$ odpovídá jeden bod v rovině (a, b). Pokud u rovnice 5.2 zvolíme parametr a, pak pro bod (x_1, y_1) lze spočítat parametr b za pomocí rovnice 5.3. Změnou parametru a v rovině (a, b) dostáváme přímku.

$$b = -ax_1 + y_1 \tag{5.3}$$

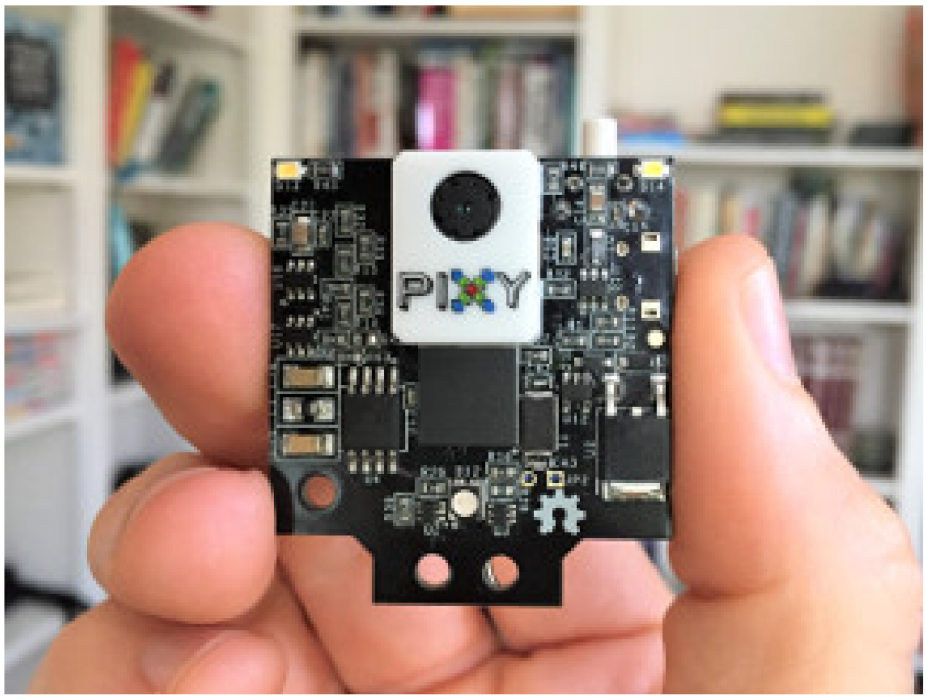
Pokud mám například 3 body v obraze, které leží v přímce v rovině (x, y): (x_1, y_1), $(x_2, y_2), (x_3, y_3)$, po dosazení do rovnice 5.3 a zanesení do roviny (a, b) zjistíme, že v bodě (a', b') se protínají 3 přímky, což znamená že na přímce $y = a'x + b'$ leží 3 body.

Samotný výpočet se provádí tak, že se rovina (a, b) rozdělí na části a pro všechny námi zvolená a se dopočítá parametr b pro všechny body. Každý takto nalezený bod evidujeme, v bodě, kde je evidováno nejvíce průsečíků, odpovídají hodnoty $(a = a', b = b')$ přímce $y = a'x + b'$.

Tato metoda lze využít podobným principem i na další tvary. Detekcí křivek za zabývá například článek A Method of Recognizing Curve Direction Based on Hough Transform [36].

5.3 Kamerka Pixy2 CMUcam5

Tato „chytrá“ kamerka dokáže sama na základě konfiguračních parametrů detekovat čáry a jejich průsečíky. Kromě detekce čar má kamerka podporu pro 16 vestavěných čárových kódů, dokáže detekovat barevné objekty a dvoubarevné kódy. Pro tuto práci je relevantní detekce čar a průsečíků, ostatní funkce využity nejsou. Všechny informace o kamerce jsou čerpány ze stránek výrobce [7]. Kamerka dokáže zpracovávat 60 snímků za vteřinu a díky tomu, že se nepřenáší celý obraz, ale pouze informace o detekovaných čárách, je možné tyto informace přenášet po sběrnici I^2C.



Obrázek 5.2: Kamerový senzor Pixy2. Převzato z [7]

Kamerka je tvořena samotným obrazovým CMOS senzorem a dvou-jádrovým čipem LPC4330FET100 od firmy NXP. Tento čip patří do skupiny Arm Cortex-M4 a je vybaven koprocesorem Arm Cortex-M0. Jádro Arm Cortex-M4 provádí výpočetní úlohy a koprocesor slouží pro zpracování vstupně výstupních operací.

Pixy2 je možné připojit na libovolnou platformou za pomocí sběrnic SPI, I2C, UART a USB. Současně jsou k dispozici knihovny pro Arduino, LEGO Mindstorms EV3, Raspberry Pi. Knihovny pro práci s kamerkou jsou k dispozici v C/C++ nebo v pythonu. Pro konfiguraci je dodávána utilita PixyMon, ve které lze jednoduše měnit jednotlivé parametry pro algoritmus detekce. Veškerý software i firmware je uvolněn jako open-source pod GNU licencí.

Jak již bylo řečeno, z kamerky se přenáší pouze detekované objekty. Výstupem z kamerky Pixy2 je pole obsahující objekty reprezentující jednotlivé čáry. Každá čára je určena za pomocí dvou bodů v kartézském souřadném systému odpovídajícímu pozici v obraze a také interním indexem. Tento index by měl být stejný přes jednotlivé snímky vstupního obrazu za předpokladu, že čára byla na předchozím snímku a současně pokračuje i na aktuálním snímku. Dalším výstupem je poté pole průsečíků, kdy každý průsečík je definován svojí pozicí a dále polem obsahujícím indexy jednotlivých čar.

Kapitola 6

Konstrukce autonomního modelu vozidla

V této kapitole bude popsán návrh řešení, které cílí na možnost účasti v soutěži NXP Cup 2019/2020. Jedná se o výběr podvozku, na který se následně připevní kamera a senzory pro detekci překážek, hardware, ve kterém poběží navržený řídící algoritmus. Dále bude potřeba zajistit napájení ve formě akumulátoru.

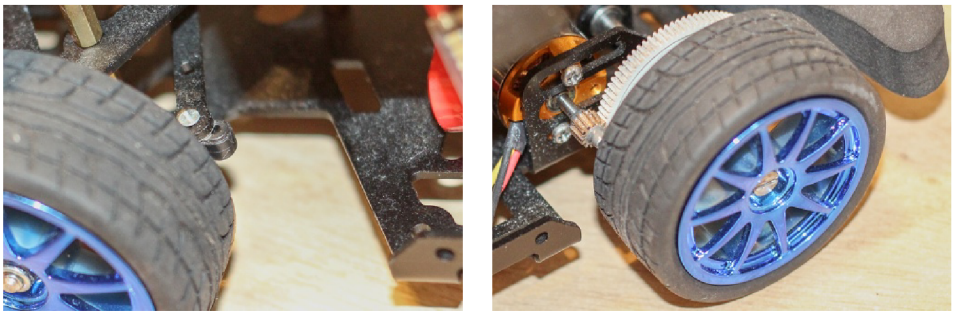
6.1 Podvozek a motory

Dle pravidel soutěže NXP Cup je možné využít téměř libovolný podvozek. Ten může být z modelářského autíčka, ale také to může být samostatný podvozek určený pro stavbu robotů. Podvozky, případně i celé modely vozidel na dálkové ovládání jsou nejčastěji určené pro jízdu v terénu. Z tohoto důvodu jsou podvozky vybaveny pružením. Existují také modely určené do interiéru, které se snaží co nejvíce podobat svým reálným předlohám. S ohledem na fakt, že bude potřeba na podvozek přichytit další součásti, jsem se rozhodl využít podvozek DFRobot ROB0156 [2], který je doporučen pro účast v soutěži přímo organizátorem. Podvozek obsahuje tyčku pro přichycení kamery, přední a zadní nárazník, díly jsou z hliníkové slitiny, což zaručí pevnost. Pro uchycení dalších součástí lze využít děrování po celé ploše dvou hlavních plátů podvozku.

Podvozek je k dispozici ve dvou variantách: s dvojicí stejnosměrných bezkartáčových motorů a s dvojicí stejnosměrných kartáčových motorů. Rozhodl jsem se využít bezkartáčovou variantu, která je letos v soutěži povolena poprvé. Dále je součástí podvozku modelářské servo, které ovládá přední nápravu. Pro řízení motorů bude osazen regulátor.

Při kompletaci podvozku jsem postupoval podle přiloženého návodu. Ve stavebnici však bylo několik problémů, které bylo potřeba vyřešit. Nebyly dodány samořezné šrouby zmiňované v návodu pro přichycení táhla od servomotorku k přední nápravě. Zde jsem tedy použil šroubek v rozměru M3x14mm a odpovídající matku (obrázek 6.2a). U zadní nápravy chyběl šroub, kterým se přichytilo zadní kolo k ose. Dalším problémem bylo ozubené kolečko, které patří na osu motoru. Ozubené kolečko mělo vnitřní průměr větší než je průměr osy a současně chyběl jistící šroubek. Rozměr ozubeného kolečka je nestandardní, v brněnských modelářských obchodech se mi jej sehnat nepodařilo. Pro zajištění kolečka jsem využil dvousložkové epoxidové lepidlo určené k lepení kovů. Jak středový šroub, tak přilepené ozubené kolečko jsou na obrázku 6.2b.

Obrázek 6.1: Podvozek DFRobot. Převzato z [2]



(a) Přední náprava (b) Zadní náprava

Obrázek 6.2: Fotografie řešených problémů

6.2 Řídící deska

Jako bylo popsáno v kapitole 3, byla vybrána platforma PX4 Robotic Drone FMU. Pro vývoj bylo potřeba vytvořit vývojové prostředí sestávající z operačního systému Ubuntu a nainstalované sady nástrojů pro platformu PX4. Zde jsem postupoval dle návodu na stránce [21]. Následně bylo potřeba stáhnout a nahrát za pomoci debuggeru do desky zavaděč a bylo možné nahrát firmware.

Pro konfiguraci desky, na které běží platforma PX4 slouží aplikace QGroundControl. Při začátku práce s novou deskou, případně změnou konfigurace hardware je potřeba nastavit typ vozidla. Na výběr je mnoho různých typů vozidel či letounů. Pro mého robota jsem tedy vybral jako typ vozidla „Rover“ a následně ještě vybral upřesnění „Generic Ground Vehicle“.

Deska je osazena senzory, které nejsou nezbytně nutné, ale při analýze jízdy, případně při chybě, by mohly být užitečné. Jedná se o kompas, gyroskop, akcelerometr. V aplikaci

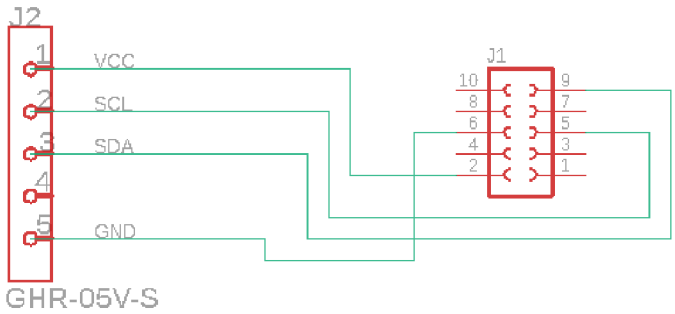
QGroundControl je možné provést kalibraci jednotlivých senzorů. Kalibrací provede průvodce, který podává instrukce, co je potřeba provést za další krok.

Následně bylo potřeba ještě nastavit počet článků akumulátoru a napětí při úplném nabití a také napětí při vybitém článku. Díky těmto informacím může řídící firmware odhadovat procentuální vybití baterie.

6.3 Senzory

Vozidlo bude muset řešit dvě senzorické úlohy. První úlohou je detekce černých vodících čar na okraji dráhy a druhou je detekce překážky, která může být umístěna před vozidlo.

Pro detekci čar jsem se rozhodl využít chytrou kamerku Pixy2 CMUcam5 popsanou v kapitole 5.3. Pro připojení kamerky Pixy2 k řídící desce slouží dodaný kabel s konektorem IDC FC-10P pro připojení kamerky a konektorem JST GHR-05V-S pro připojení do řídící desky. Bohužel se ukázalo, že tento kabel má špatné zapojení a bylo potřeba je opravit. Správné zapojení dle datových listů obou zařízení je zobrazeno na obrázku 6.3.



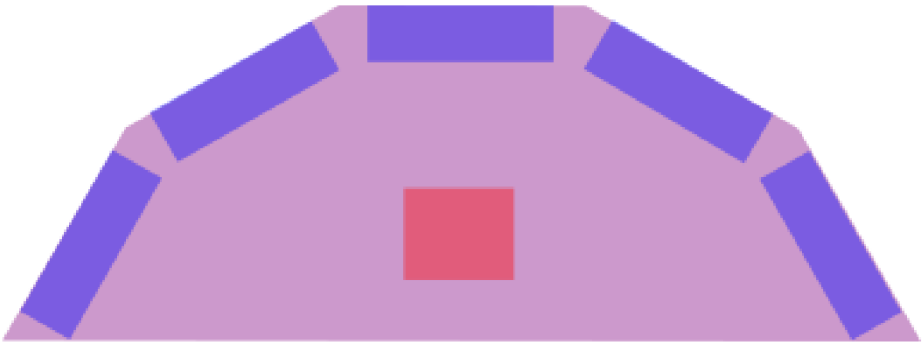
Obrázek 6.3: Schéma správného zapojení kabelu pro propojení Pixy2 a RDDRONE-FMUK66

Druhou úlohou je detekce překážky. Zde jsem se rozhodl využít ultrazvukový senzor. Jeden statický ultrazvukový senzor však poskytne pouze informaci o tom, zda je, nebo není před vozidlem překážka. O směru vyhnutí nepodá žádnou informaci. Uvažoval jsme tedy dvě varianty.

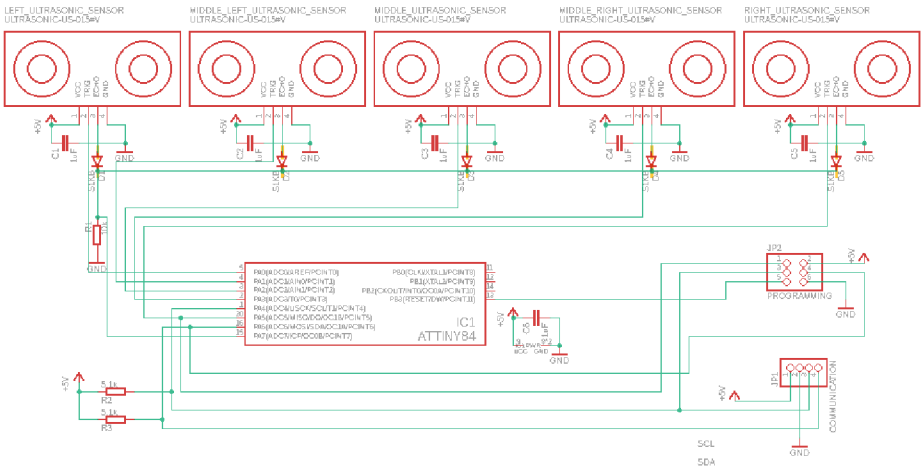
První variantou bylo umístit ultrazvukový senzor na modelářské servo a pohybovat s ním v rozsahu 180 stupňů ze strany na stranu a napodobit tak princip fungování LIDARu (Light Detection And Ranging). Ultrazvukový senzor má však na rozdíl od laserového paprsku širší úhel záběru a proto by informace ohledně úhlu natočení senzoru a vzdálenosti umožnily pouze přibližný odhad toho, jak vypadá okolí. Pro přesnější výsledky by bylo nutné využít lidar, toto řešení je však drahé a pro účast v soutěži nevhodné.

Rozhodl jsem se tedy vytvořit vlastní senzor, který bude obsahovat 5 ultrazvukových senzorů US-015 a připojí se přes sběrnici I2C. Tyto ultrazvukové senzory mají uhel detekce 15-35 stupňů. Těchto 5 senzorů jsem tedy vzájemně natočil vždy o 30 stupňů. Návrh rozložení je na obrázku 6.4. Modře jsou zakresleny jednotlivé senzory US-015, červeně je poté zakreslen mikročip, který bude obsluhovat senzory a naměřené hodnoty zašle do hlavní řídící desky.

V průběhu návrhu této části nastala celosvětová pandemie, soutěž byla odložena na podzim roku 2020 a současně nastalo omezení pohybu osob na území České republiky. Byl jsem tedy omezen dostupnými prostředky a současně jsem věděl, že mé řešení diplomové



Obrázek 6.4: Vizuální návrh senzoru s 5 ultrazvukovými čidly



Obrázek 6.5: Schéma senzoru s 5 ultrazvukovými čidly

práce se na reálnou soutěž nedostane. Z tohoto důvodu jsem zde porušil podmínku použít všechny čipy od firmy NXP. Navržený obvod je založen na mikročipu Attiny84. Schéma navrženého obvodu je na obrázku 6.5.

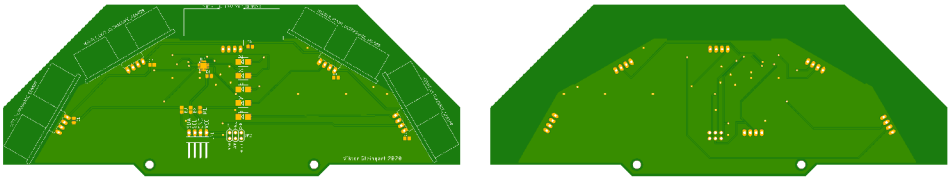
Z mikrokontroléru Attiny84 je vyvedeno komunikační I^2C rozhraní na konektor JP1 a rozhraní ICSP (in circuit serial programming) pro programování čipu na konektoru JP2. Paralelně k napájení jednotlivých senzorů a také paralelně k napájení mikročipu jsou připojeny kondezátory C1 - C6 o kapacitě 1μF. Tyto kondenzátory mají za cíl odstranit případné rušení, které by mohlo vzniknout na jednotlivých vodičích. Dále byly na vodiče SCL a SDA rozhraní I^2C připojeny pull-up rezistory R2 a R3 s hodnotou odporu 5,1 kΩ.

Ultrazvukový senzor US-015 má čtyři piny: kromě dvou napájecích (VCC a GND) jsou zde další dva ECHO a TRIG. Po přivedení pulzu s logickou 1 o délce alespoň 10μs na pin TRIG vyšle ultrazvukový senzor 8 pulsů na frekvenci 40kHZ. Následně očekává že se odražený signál vrátí. Dobu, jakou trvalo, než se signál vrátí následně lze odečíst z pinu ECHO. Zde doba, po jakou je na pinu přivedena logická 1 odpovídá času, který zvuk strávil na cestě k předmětu, od kterého se odrazil a následně se vrátil zpět k senzoru.

Mikročip Attiny84 tedy bude postupně přivádět na TRIG piny jednotlivých senzorů krátké pulsy a následně počítat vzdálenost. Po získání vzdálenosti ze všech 5 senzorů se tato vzdálenost v centimetrech přenese po sběrnici I2C do hlavní řídící desky.

Senzor US-015 detekuje a na ECHO pin zasílá signál pouze po přivedení pulzu na TRIG pin. Za předpokladu že bude mezi přiváděním pulzu na TRIG piny dostatečný čas, je možné

ECHO piny za pomocí OR hradla vytvořeného diodami a rezistorem připojit na jeden pin mikročipu. Pro vytvoření OR logiky slouží diody D1 - D5 a odpor R1.



(a) Vrchní strana (b) Spodní strana

Obrázek 6.6: Náhled na návrh desky plošných spojů

Ze schématu byla následně navržena deska plošných spojů v programu Autodesk Eagle (Obrázek 6.6). Deska byla následně zaslána do výroby a osazena součástkami. Mikročip byl naprogramován a senzor byl připojen k řídící desce.

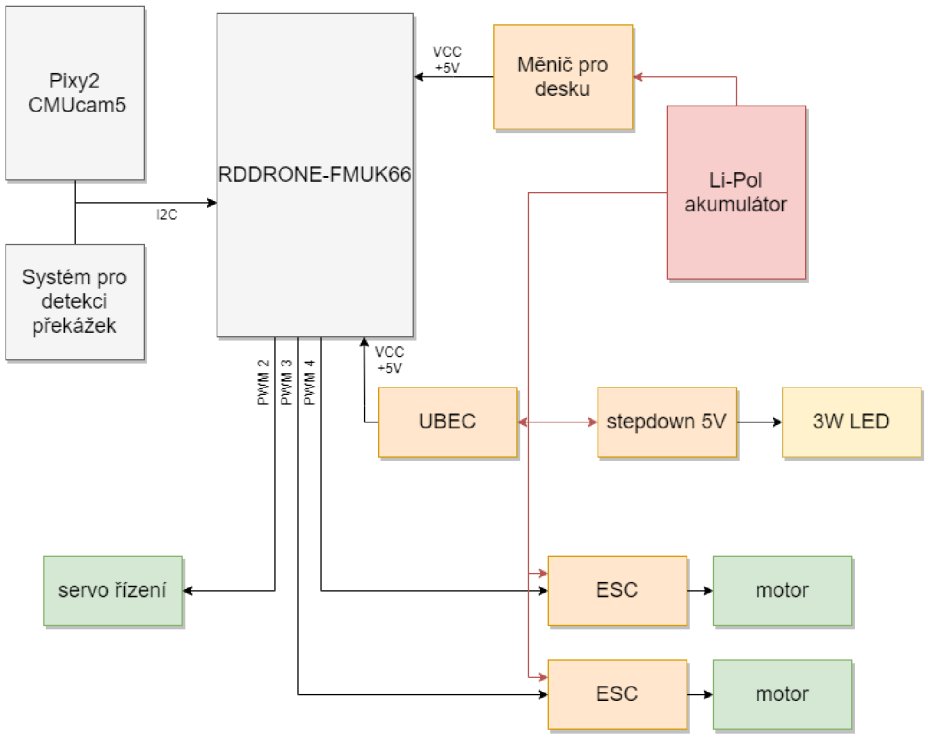
6.4 Finální zapojení komponent

Všechny výše zmíněné komponenty bylo potřeba namontovat na rám podvozku a následně propojit. Na obrázku 6.7 jsou blokově zakresleny všechny komponenty a jejich vzájemné propojení.

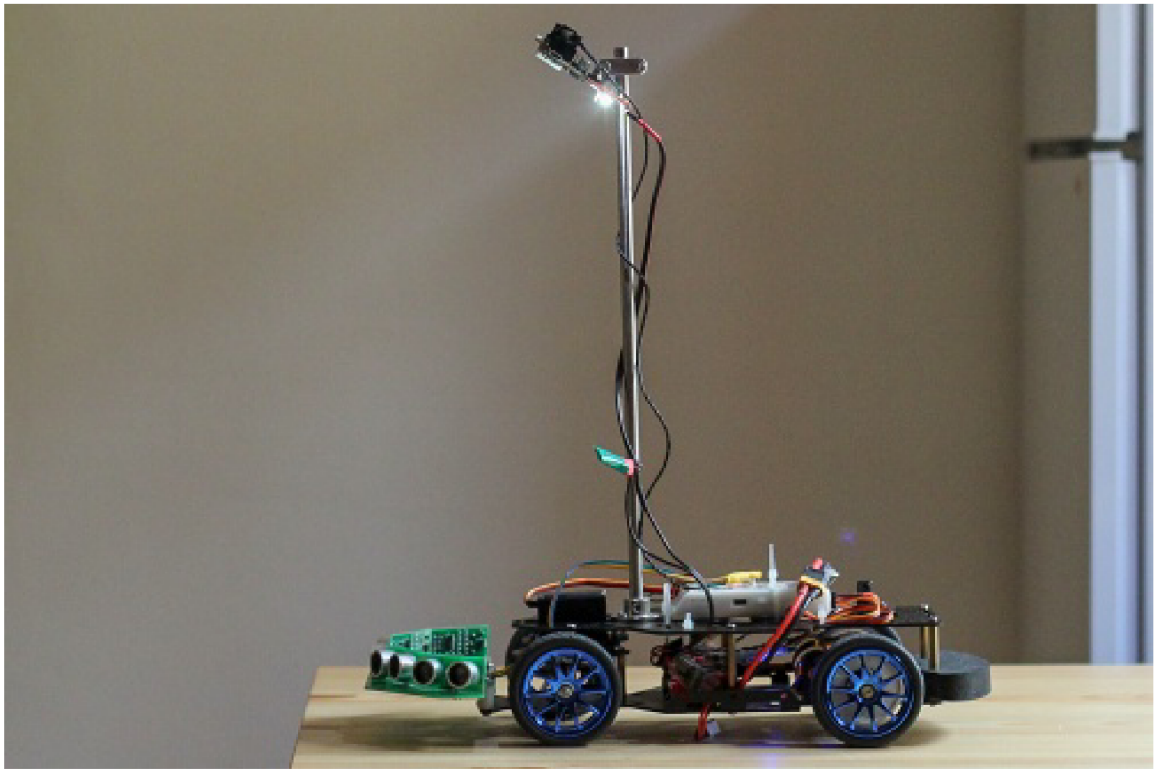
Popis napájení desky začnu u baterie. Byla zvolena lithium - polymerová baterie skládající se ze dvou článků, poskytuje napětí 7,4V a kapacita baterie je 2200mAh a je vybavena konektorem XT60. Na baterii je připojeno paralelně 5 zařízení. Jedná se o napájecí modul řídící desky, který kromě snížení napětí na 5V měří napětí napájecího zdroje a tuto informaci poskytuje řídící desce. Řídící deska sama o sobě nepřivádí napětí na napájecí piny konektorů pro připojení motorů. Je proto třeba připojit do pro tyto účely určených pinů externí zdroj napájení, v tomto případě spínaný stabilizátor napětí (UBEC). Pro napájení BLDC motorů je potřeba připojit na napájení oba regulátory (ESC). Posledním modulem připojeným na napájení je step-down měnič na 5V pro napájení výkonné 3W LED diody. O významu diody bude pojednáno v kapitole 8.7. Zde je uvedena pro úplnost schématu zapojení.

Napájení motorů je vyřešeno, je však potřeba připojit vodiče s řídícím signálem na desku. Pro připojení motorů slouží piny na boční straně desky. Je zde vedle sebe 7 trojic pinů, kde horní řada je připojena na zem, prostřední řada je napájení 5V a spodní řada je signál. Jak bylo zmíněno výše, napájení není deskou poskytováno, je však propojeno s první trojicí pinů, do kterých se připojuje výše zmíněný stabilizátor. Dalších 6 trojic pinů tedy slouží pro připojení motorů. Zde byla pro připojení servo motoru řízení zvolena trojice pinů s PWM výstupem 2, pro připojení ESC řídící BLDC motory pak trojice s PWM výstupem 3 a 4.

Pro připojení kamerky a systému pro detekci překážek je využito rozhraní I^2C vyvedené na horní stranu desky. Zde bylo třeba vytvořit vodič umožňující připojení obou zařízení na jeden I^2C výstup. Na obrázku 6.8 je fotografie finální podoby autíčka včetně zapojení.



Obrázek 6.7: Schéma zapojení všech komponent



Obrázek 6.8: Fotografie hotového autíčka

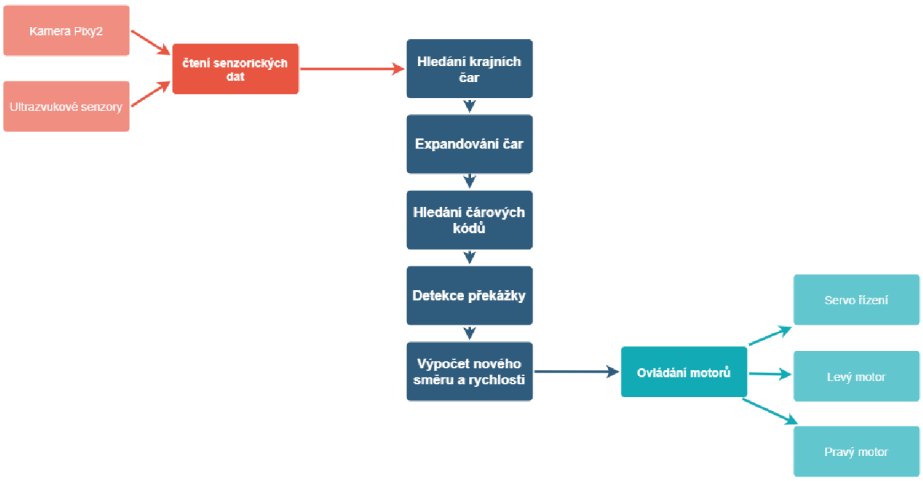
Kapitola 7

Návrh a implementace řídícího algoritmu

V této kapitole bude popsán návrh a implementace řídícího algoritmu. Nejdříve bude představena základní struktura algoritmu a následně podrobněji popsány jednotlivé části.

7.1 Základní návrh

Celý algoritmus je rozdělen do několika bloků. Tyto bloky jsou zobrazeny na obrázku 7.1. Vstupem do algoritmu jsou data z kamery Pixy2 a také ze mnou navrženého modulu s ultrazvukovými senzory. Výstup algoritmu je informace o rychlosti pro pravý a levý motor a natočení kol za pomoci serva řízení. Aplikace, ve které je implementován řídící algoritmus využívá knihovny z frameworku PX4 a je postavena na ukázkové aplikaci dostupné na stránkách NXP Cup [26].



Obrázek 7.1: Struktura řídícího algoritmu

Hlavní programové vlákno implementované funkcí `race_thread_main` při spuštění inicializuje potřebné objekty. Jedná se o strukturu `driveParam`, ve které jsou ukládány získané i vypočítané informace o dráze, či jízdě a také struktura `motorControl`, v níž je uložena rychlost pro oba motory a také natočení kol.

V hlavní programové smyčce je vždy volána funkce z Line tracking API kamerky Pixy2 `getAllFeatures`. Následně je volána funkce `raceTrack`, které jsou předány výše zmíněné struktury a také objekt reprezentující kamerku a její API.

V následujících podkapitolách bude popsána funkčnost a implementace jednotlivých funkčních bloků, které odpovídají samostatným funkcím volaným z funkce `raceTrack`.

7.2 Hledání krajních čar

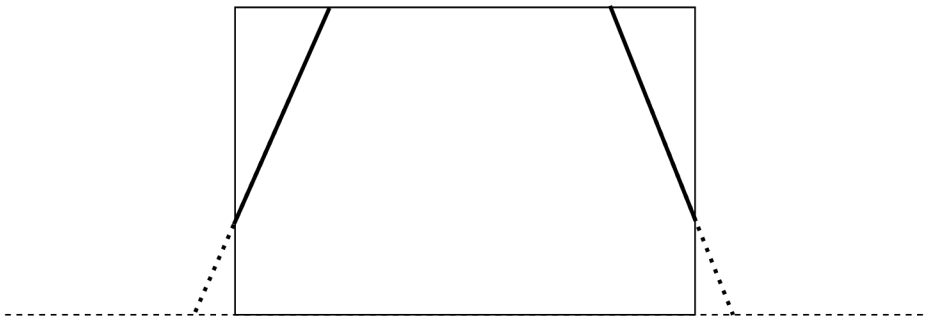
Protože je možné, že se bude v obraze vyskytovat více než dvě čáry, například pokud bude vozidlo projíždět po díle obsahujícím začátek, či konec měřeného úseku (viz obrázek 2.2), je potřeba nalézt okrajové čáry. Tato funkčnost je implementována ve funkci `findLeftRightLine`.

Pokud algoritmus v předchozí iteraci nalezl dvě okrajové čáry (jsou nastaveny), budou jejich indexy uloženy a v aktuální iteraci bude kontrolováno, zda se v obraze nevyskytují opět čáry s těmito indexy (jinými slovy jestli se čára stále vyskytuje na obraze). Toto hledání probíhá tak, že jsou nejdříve na začátku funkce nastaveny parametry `driveParam.validLeftLine` a `driveParam.validRightLine` na hodnotu `false`. Následně se prochází v cyklu všechny nalezené vektory a pokud odpovídá index dříve nalezené čáry s aktuálním, jsou aktualizovány pozice bodů, kterými je definována čára a následně je nastaven parametr `driveParam.validLeftLine` resp. `driveParam.validRightLine`.

Pokud však levá či pravá čára nalezena nebyla, procházejí se všechny nalezené čáry a vybírá se nejlepší kandidát na okrajovou čáru. Přestože by okrajová čára měla vést v ideálním případě po celé výšce obrazu, díky zornému úhlu kamery a natočení vozidla se může stát, že okrajová čára bude viditelná pouze částečně. Algoritmus předpokládá, že nejdelší čára, která má spodní bod na levé straně, je vodící čárou na levé straně, obdobný přístup je poté aplikován i na pravou stranu.

7.3 Expanze čar

Pro detekci čárových kódů, které značí začátek a konec měřeného úseku je potřeba nalézt všechny čáry, které se nacházejí mezi okrajovými čárami. Okrajové čáry však mohou být vlivem zorného úhlu kamery snímány jen částečně. Proto se dopočítá průsečík čáry se spodním okrajem obrazu a následně se tento průsečík použije jako spodní bod obrazu. Toto prodloužení je znázorněno na obrázku 7.2. Spodní hrana obrazu je rozšířena do stran čárkovanou čárou, prodloužení okrajových čar je znázorněno tečkovaně. Stejná operace se provede i s horním bodem.



Obrázek 7.2: Expanze čar

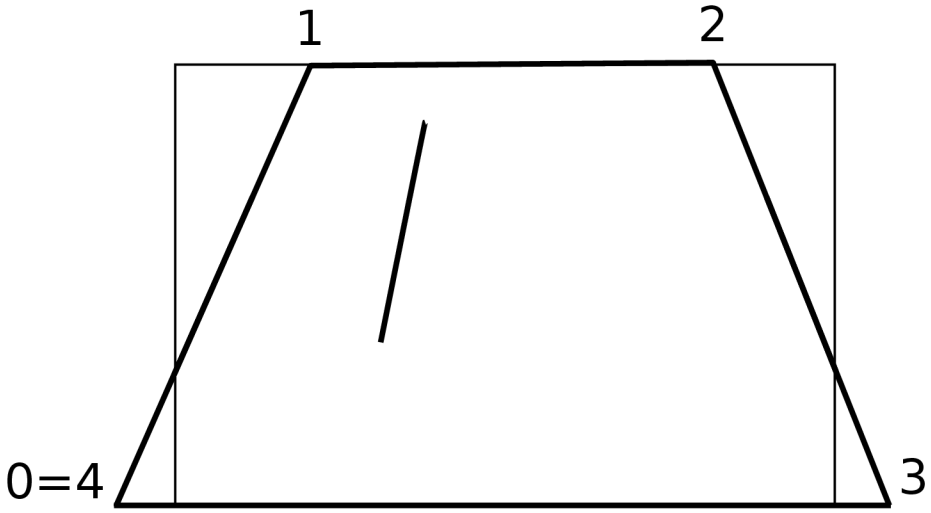
Implementace prodloužení čar je ve funkci `expandLines`. Pro výpočet souřadnice `x` v bodě `y` slouží funkce `pointOfVector`:

```
int pointOfVector(int y, VectorInt *line) {
    int dx = line->m_x1 - line->m_x0;
    int dy = line->m_y1 - line->m_y0;

    float deltay = (float) (y - line->m_y1);

    if (dy == 0) {
        return line->m_x0;
    } else {
        float x = (float) line->m_x1 + (dx / (float) dy) * deltay;
        return (int) round(x);
    }
}
```

Stejně jako na obrázku 7.2 se může stát, že vypočítaný bod leží mimo obraz a tím pádem se může dostat souřadnice `x` do záporné hodnoty. Definice struktury Vector však využívá pro ukládání bodu datový typ `uint8_t`. Proto jsem nadefinoval strukturu VectorInt, která používá větší `int`, kde již není problém ani záporná hodnota.

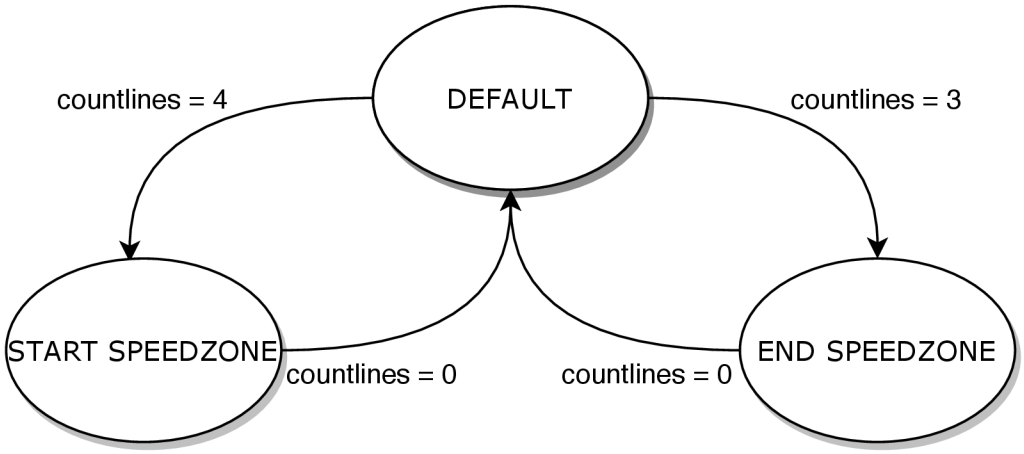
7.4 Hledání čárových kódů

Po prodloužení čar víme, že okrajové čáry vedou od horního po spodní okraj obrazu, s tím že body mohou ležet mimo obraz. Nyní se tedy pro každou čáru ověří, že leží uvnitř polygonu, který vznikne ze 4 bodů tvořící rozšířené čáry. Pro ověření, zda bod leží v polygonu je využita funkce `pointInPolygon`, která je převzata ze zdroje [32]. Této funkci jsou zadána dvě pole, obsahující souřadnice `x` a `y`. Pokud by jedna, nebo případně obě čáry chyběly, použijí se souřadnice bodů odpovídající rohům obrazu. Indexy použité pro jednotlivé body jsou v obrázku 7.3.



Obrázek 7.3: Polygon pro hledání čar v dráze

Ověření zda čára leží v polygonu se provádí tak, že se postupně ověří oba krajní body čáry, zda leží v polygonu. V případě že jsou nalezeny 3 nebo 4 čáry ležící uvnitř polygonu, pravděpodobně to znamená že se jedná o speciální značku 3 nebo 4 pruhů. Protože je ale možné, aby byly detekovány například všechny 4 čáry, ale následně při přejezdu už jedna čára detekována není a počet detekovaných čar je 3. V případě že by se pouze na základě počtu čar určovalo, zda v případě úkolu na měřený úsek zpomalit nebo ne, mohlo by se lehce stát, dříve než se vůbec vyjede ze značky 4 pruhů, že vozidlo zase zrychlí, protože na konci detekovalo jen 3 čáry. Z tohoto důvodu je do algoritmu přidán stavový automat, kdy vstupní symbol je počet detekovaných čar a vnitřní stavy reprezentují stav, kdy není žádný speciální symbol na dráze (`DEFAULT`), kdy je symbol začátku měřeného úseku (`START SPEEDZONE`) a kdy je symbol konce (`END SPEEDZONE`). Automat je znázorněn na obrázku 7.4. V případě, že vstupní symbol je jiný, než jaký je definován u přechodu na obrázku, stavový automat zůstává ve stejném stavu.



Obrázek 7.4: Stavový automat pro detekci speciálních značek na dráze

7.5 Detekce překážek

Pro informace o překážkách na dráze před vozidlem se využívá mnou navržený systém osazený 5 ultrazvukovými senzory. Komunikace se senzorem je implementována v souborech `UltrasonicI2C.h` a `UltrasonicI2C.cpp` za využití třídy I2C z platformy PX4. V souboru `Ultrasonic.h` je pak definována třída, která za využití metody `recv` čte jeden byte ve funkci `readByte`. Pro přečtení všech 5 hodnot ze senzoru pak slouží funkce `readValues`.

```
void readValues(uint8_t *buf){
      while(readByte()<255)
          PX4_INFO("wait for ultrasonic sensor sync");
      for(int i = 0; i<5; i++){
```

```
            buf[i] = readByte();
        }
    }
```

Tato funkce nejdříve čeká než dostane hodnotu 255. tato hodnota je ze senzoru vyslána pouze v případě, kdy senzor začíná vysílat hodnoty. Poté následuje 5 bytů s hodnotami 0-254, které odpovídají detekované vzdálenosti v jednotlivých senzorech.

Přečtené hodnoty se uloží do pole `param.obstacle` a současně se nalezne minimální hodnota vzdálenosti, která se uloží do `param.obstacleDistance`.

7.6 Výpočet nového směru a rychlosti

Poslední fází je určení nového směru a rychlosti ze získaných informací o poloze a směru čar a také o překážce na dráze. Tato funkčnost je implementována ve funkci `controlSpeedSteer`.

Směr se určí tak, že se vezmou obě krajní čáry (v případě že na některé ze stran čára chybí, uvažuje se jako čára patřičná pravá nebo levá hranice obrazu. Následně se z vektoru okrajových čar spočítá poloha bodu vertikálně v bodě dle hodnoty`COMPUTE_POINT_CENTER`. Z těchto dvou bodů se spočítá třetí bod, který leží uprostřed mezi těmito body.

Pokud byla detekována senzorem napravo, nebo senzorem druhým zprava překážka, nastaví se místo pravé okrajové čáry jako okraj hodnota poloviny šířky. Stejná hodnota se nastavuje jako levá překážka v případě, že byla překážka detekována levými senzory.

Pro převod bodu v obraze na směr řízení se použije konstanta upravující převodní poměr mezi bodem v obrazu a hodnotou natočení kol v intervalu $< -1, 1 >$.

Rychlost motorů se v případě že se vozítko nachází v měřeném úseku zmenší na poloviční hodnotu. Při prudším zatáčení je pro větší stabilitu zpomaleno vnitřní kolo. Pokud byla detekována libovolným senzorem překážka ve vzdálenosti menší než `OBSTACLE_AVOID_DISTANCE`, vozidlo zastaví. Pokud byla hodnota zaznamenána 20 iterací po sobě, nejedná se o chybu měření, ale o překážku a proto program skončí.

Kapitola 8

Praktické experimenty s pohybem

V této kapitole bude popsáno vyzkoušení funkčnosti komponent a následně navržený algoritmus na jednotlivých typech dráhy.

8.1 Základní funkčnost komponent a konfigurace

Po zkompletování celého vozidla bylo potřeba vyzkoušet, zda všechny části systému fungují dle očekávání. První celek, který bylo potřeba vyzkoušet je ovládání motorů. Pro vyzkoušení funkčnosti motorů je vhodná utilita `pwm test` ze sady nástrojů PX4. Jako parametry přijímá PWM kanály, na které se bude signál zasílat (např. `-c 23` pro kanály 2 a 3) a délku PWM pulzu v μs (např. `-p 1500`). V případě zapojení dle předchozí kapitoly lze otestovat funkčnost BLDC motorů příkazy:

```
pwm test -c23 -p 1485
pwm test -c23 -p 1600
pwm test -c23 -p 1800
pwm test -c23 -p 1400
pwm test -c23 -p 1200
```

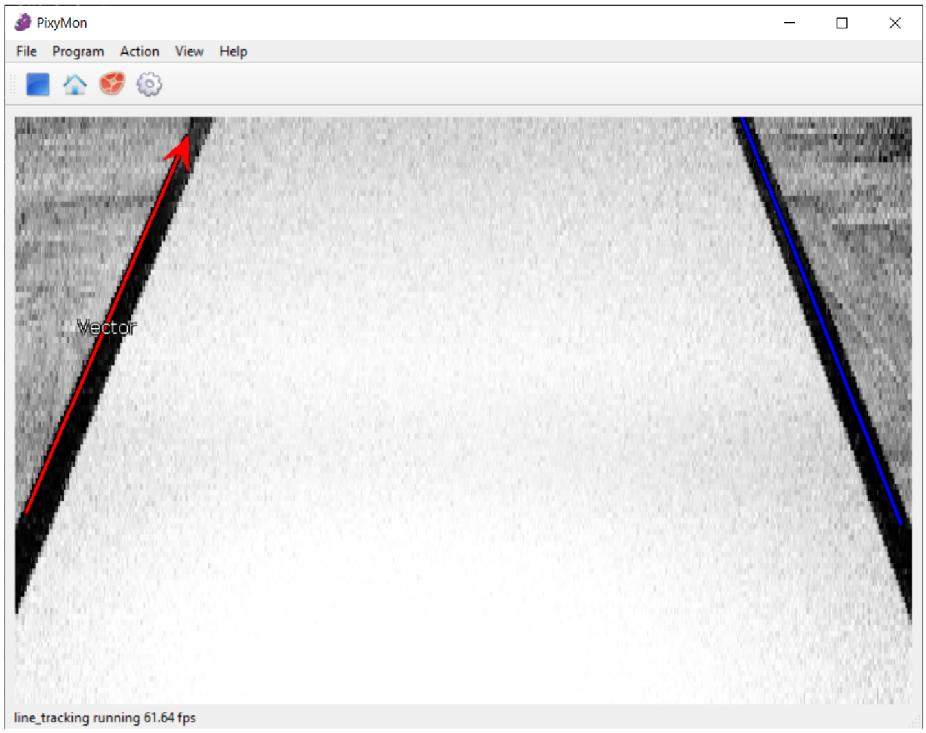
Regulátor BLDC motorů má bezpečnostní funkci, kdy dokud nedostane PWM signál odpovídající nulovému pohybu (v tomto případě délka pulzu 1485μs), nereaguje na žádné jiné hodnoty a motorky se nepohybují. Proto první hodnota, která musí být pro test zaslána je právě hodnota nulového stavu. Dále pro bezproblémový chod je vhodné nastavit v aplikaci QGroundControl konfigurační parametry `PWM_MAIN_DI3` a `PWM_MAIN_DI4` na hodnotu právě 1485μs. Je zde potřeba dávat pozor na fakt, že zatímco u utility `pwm test` jsou PWM kanály indexovány od 0, v konfiguraci QGroundControl je indexace PWM výstupů shodná s popiskem na krabičce k desce. Po spuštění prvního výše uvedeného příkazu motorek vydal melodický zvuk značící odblokování bezpečnostní funkce. Následně další příkazy dle očekávání způsobily rotaci motorků ve směru jízdy pomalu a rychle a následně rotaci v opačném směru opět pomalu a rychle. Následně byl stejným způsobem úspěšně otestován servomotorek pro ovládání řízení v celém rozsahu pohybu:

```
pwm test -c1 -p 1500
pwm test -c1 -p 2000
pwm test -c1 -p 1000
```

Pro otestování funkčnosti kamerky Pixy2 CMUcam5 výborně poslouží aplikace PixyMon dodávaná spolu s kamerkou. Po připojení kamerky do USB rozhraní počítače se zobrazí

obraz z kamery. Pro detekci čar je třeba vybrat `line_tracking` v menu Program. Následně jsou v obraze vykresleny detekované čáry (Obrázek 8.1). V konfiguraci aplikace lze nastavit parametry pro algoritmus detekce. Parametry použité v následujících úkolech jsou v tabulce 8.1.

Pro otestování funkčnosti ultrazvukového senzoru je potřeba jej připojit za pomocí sběrnice I^2C k zařízení, které bude požadovat čtení hodnot. Pro jednoduchost jsem pro vyzkoušení senzoru použil Arduino Uno a jednoduchý prográmek, který čte hodnoty a tyto následně vypisuje na Seriovou linku, kde lze jednoduše zkontrolovat, že hodnoty se mění při vkládání překážky před jednotlivé senzory.



Obrázek 8.1: Detekované čáry v aplikaci PixyMon

8.2 Tvorba dráhy

Pro otestování jízdy po vytyčené dráze jsem v prvních pokusech používal barevnou pásku nalepenou na podlahu. Problémem zde však byla potřeba vytyčit dráhu pokud možno stejných rozměrů, jako je oficiální dráha v soutěži NXP Cup. Současně se jako problematické jevilo použití jiné, než černé barvy, která nevytvářela potřebný kontrast s parketami na podlaze.

Rozhodl jsem se tedy vytvořit si jednotlivé díly dráhy ve stejných rozměrech, jaké jsou uvedeny v pravidlech soutěže NXP Cup. S ohledem na potřebu relativně velkého počtu dílů a současně s ohledem na cenu materiálu a možnosti zpracování jsem se rozhodl za materiál zvolit lepenku. Ve vektorovém editoru jsem překreslil tvary jednotlivých dílů a spoje doplnil výstupky tak, aby se daly jednotlivé díly libovolně spojovat. Tyto navržené díly jsem vyřezal na laserové řezačce v otevřené dílně Fablab.

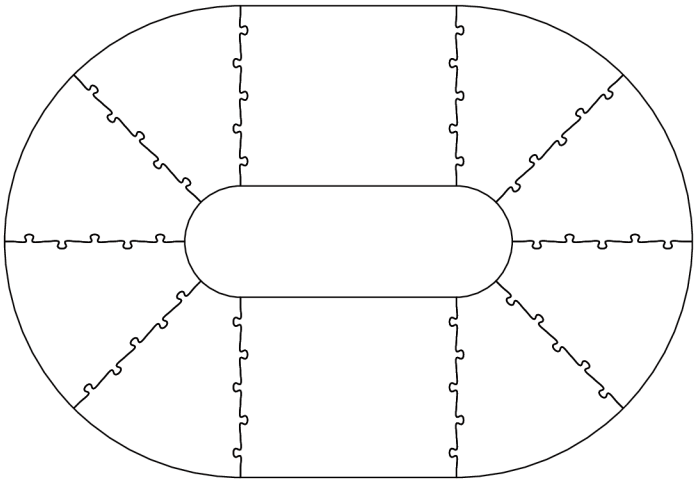
Parametr	Hodnota
Edge threshold	78
Maximum line width	132
Minimum line width	8
Camera brightness	255
Edge distance	2
Line extraction distance	1
Maximum line compare	5000
Maximum merge distance	0
Minimum line length	19
Line filtering	0
Intersection filtering	10
Barcode filtering	0
Default turn angle	0

Tabulka 8.1: Hodnoty nastavené v konfiguraci aplikace PixyMon

8.3 Jednoduchý okruh

Nejjednodušší typ dráhy a současně nejvhodnější pro vyladění jednotlivých parametrů je jednoduchý ovál zobrazený na obrázku 8.2.

Na tomto jednoduchém okruhu jsem experimentoval především s nastavením `COMPUTE_POINT_CENTER` – konstantou pro výpočet bodů ležících na průsečíku vodících čar a této hladiny. Zde jsem došel nakonec k hodnotě rovné polovině výšky obrazu. Když byla hodnota nižší, dostatečně se neprojevilo zatočení čáry.



Obrázek 8.2: Jednoduchý okruh

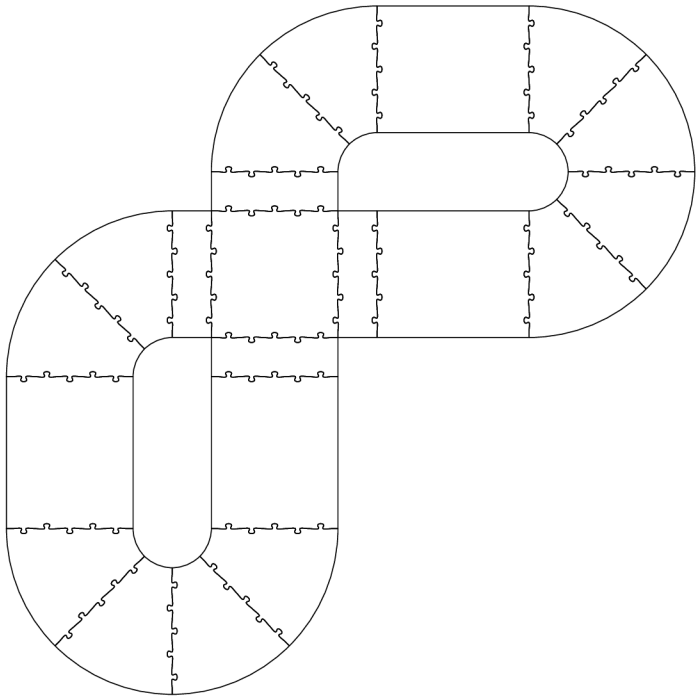
8.4 Křižovatka

Kromě rovného úseku a úseku se zatáčkou je možné dráhu postavit také s křižovatkou. K tomuto účelu slouží 5 dílů, které dohromady utvoří křižovatku. Následně lze tuto křižovatku

napojit na další díly. Především z prostorových důvodů v obýváku, kde jsem převážnou většinu experimentů prováděl, jsem zvolil tvar dráhy zobrazený na obrázku 8.3.

Na této dráze se otestuje schopnost zatáčet doprava i doleva, projet rovný úsek a také schopnost projet křižovatkou. Cílem je, aby vozidlo vždy projelo křižovatku rovně.

Protože celá křižovatka je dlouhá 89 centimetrů. z čehož středový díl na kterém chybí vodicí čáry měří 50cm, byl tento typ dráhy poněkud problematický. Bylo potřeba experimentovat s náklonem a pozicí kamery a také nastavením detekčního algoritmu na kameře Pixy2. Pokud byla kamera připevněná v zadní části vozidla, fungovala detekce spolehlivě, avšak díky tomu, že kamera viděla větší část odbočující cesty a současně měla větší toleranci, pokud čára prudce měnila směr, nebyla ostrá hrana v křižovatce chápána jako průsečík dvou čar, ale jako čára jedna. Díky tomu vozidlo mělo tendenci zatáčet doprava, nebo doleva, v závislosti na tom, kterou vodící čáru sledovalo. Řešením této situace bylo přesunutí kamery do přední části vozidla a také nastavení parametrů dle tabulky 8.1. Po této změně autíčko křižovatku projíždí bez problémů.

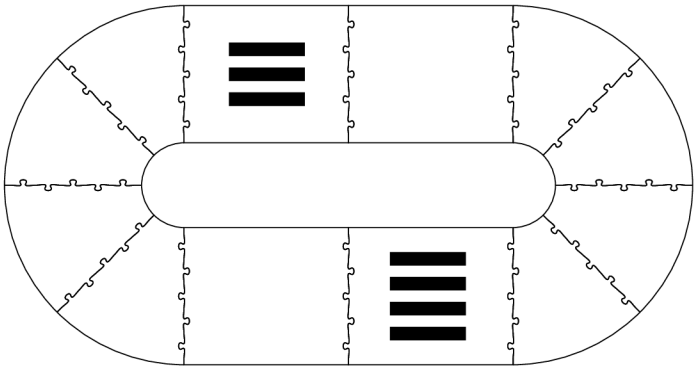


Obrázek 8.3: Okruh s křižovatkou

8.5 Úsek s omezenou rychlostí

Dalším ze specifických dílů dráhy je začátek a konec měřeného úseku značený čtyřmi resp. třemi podélnými pruhy na dráze. Pro vyzkoušení funkčnosti jsem zvolil jednoduchý ovál, do kterého byly přidány tyto dva díly. Výsledná dráha je pak na obrázku 8.4.

Na této dráze se ukázalo, že v některých případech má autíčko problém opět v zorném úhlu kamery. Ne v každém případě totiž autíčko nutně vidělo obě vodící čáry a také se stávalo že sice detekovalo čtyři čáry značící začátek měřeného úseku, následně ale díky úhlu kamery jedna z čar zmizela a autíčko tedy opět zrychlilo. Proto byla doplněna logika popsaná v kapitole 7 pracující na základě stavového automatu 7.4.



Obrázek 8.4: Okruh s omezením rychlosti

8.6 Překážky

Pro detekci překážek jsem použil jednoduchý okruh a na něj položil překážku. V případě hranaté překážky byl problém s detekcí. Ultrazvukový senzor dobře detekuje překážky, které mají plochu kolmou na vlny vycházející ze senzoru, protože se zvuková vlna odrazí zpátky do senzoru. Pokud je však tato překážka natočena v příliš velkém úhlu vůči vlnám, vlny se nevrátí a překážka není detekována. Zkusil jsem tedy využít kulatou překážku, kterou již senzor detekuje dobře.

Během experimentů se ukázalo, že navržené zapojení ultrazvukových senzorů nefunguje úplně dle očekávání, výstupy z echo pinů se částečně překrývají a způsobují neplatné hodnoty. Tento problém se částečně podařilo vyřešit softwarovou cestou - čtení pouze ze tří prostředních senzorů a současně dostatečná pauza mezi jednotlivými čteními.

Senzor má udávaný maximální dosah 4 metry, lze tedy předpokládat že po čase, za který by urazil zvuk tuto vzdálenost a následně odražený zvuk zpátky, přestane obvod čekat na signál. Tento minimální čas lze spočítat:

$$t = d/s = 8/343 = 0,024s = 24ms \tag{8.1}$$

Mezi měřeními byla zvolena pauza 30ms a současně je prováděno měření pouze na 3 senzorech, což způsobuje lepší funkčnost systému. Vyřazení z provozu dvou krajních senzorů nemá příliš vysoký vliv na funkčnost řešení, protože úhel je příliš otevřený a překážka, která je detekována těmito senzory, je zpravidla dříve detekována senzory blíže ke středu, nebo leží úplně mimo dráhu.

8.7 Problém s nedostatečným osvětlením

Pro kamerku se jeví jako problematické příliš dynamické osvětlení. V okamžiku, kdy na dráze byla více osvětlená místa, například sluneční svit pronikající mezi žaluziemi, kamerka nedokázala v těchto místech správně detekovat vodící čáru dráhy. Prvním pokusem o zlepšení situace bylo zapnutí LED na desce kamerky. Světelný výkon diod však nepřinesl přílišné zlepšení, byla instalována ještě jedna 3W LED s udávaným světelným tokem 180-200 lumen. Napájení diody bylo třeba zajistit ze zdroje s napětím 3,3V, který unese větší proudový odběr přibližně 1 ampér. Z tohoto důvodu byl instalován step-down měnič, který napětí na baterii mění na zvolenou hodnotu, v tomto případě 3,3V. Hodnota výstupního napětí se zvolí buď na spodní straně desky plošného spoje propojením kontaktů, nebo lze

zvolit nastavení za pomoci trimru. Kvůli možnosti jemnějšího nastavení napětí jsem zvolil nastavení za pomoci trimru.

Přidání dodatečného osvětlení situaci zlepšilo, avšak je stále potřeba dbát na co nejvyrovnanější světelné podmínky na celé dráze.

Kapitola 9

Závěr

Práce si kladla za cíl sestavit autíčko schopné účasti v soutěži NXP Cup. Po celou dobu práce jsem s účastí v soutěži počítal, byl jsem přihlášen do soutěže a na konci března, kdy měla proběhnout kvalifikace, jsem měl autíčko schopné jízdy po vytyčené dráze a také schopné detekovat úsek, ve kterém by mělo jet pomalu. Jediné, co v této době autíčko nedokázalo, byla detekce překážek.

V této době však vypukla celosvětová pandemie nemoci Covid-19, kvůli které byla celá soutěž přesunuta na podzim roku 2020. Současně jsem byl velice omezen s vybavením, které mám doma. Z tohoto důvodu byl pro systém pro detekci překážek zvolen mikrokontroler Attiny84, který jsem byl doma schopný programovat. V době odevzdání práce je funkční již i detekce překážek, i když zde se ukázalo, že mnou navržený systém detekce není úplně dokonalý a pro účast v soutěži by potřeboval ještě úpravy. Především využít obdobný mikrokontroler vyrobený firmou NXP. Současně by stálo za zvážení využít pro detekci překážek zařízení na principu laseru či mikrovlnného radaru, který by se měl lépe vyrovnat s natočením měřeného předmětu.

Současně by stálo za zvážení využít pro zpracování obrazu modul, který dokáže akcelerovat neuronové sítě (například NXP i.MX 8M), za pomoci kterého by se dalo detekovat nejenom čáru, ale i překážku.

Dalším možným rozšířením této práce by byl modul pro bezdrátovou komunikaci s počítačem. Tento modul by mohl nejenom přenášet informace ze senzorů, ale také obraz snímaný kamerou v případě propojení modulu s kamerou. Současně by mohl umožnit bezdrátové ovládání autíčka. Tento modul by však musel být odnímatelný, aby autíčko stále odpovídalo pravidlům soutěže NXP Cup.

Tato práce se dle mého názoru vydařila, kromě detekce překážek (volitelná bonusová úloha) by všechny ostatní soutěžní úkoly ze soutěže NXP Cup autíčko zvládlo. Pro mne byl velice přínosný návrh a realizace desky plošných spojů pro systém detekce překážek, protože toto byla pro mne úplně nová dovednost a jsem rád, že jsem i v tomto směru mohl rozšířit vědomosti získané při studiu vysoké školy.

Literatura

[1] *AWS DeepRacer* [online]. [cit. 2020-05-29]. Dostupné z: https://aws.amazon.com/deepracer/.

[2] *DFRobot ROB0165 Smart Robot Brushless Motor Racing Car* [online]. [cit. 2020-05-29]. Dostupné z: https://cz.mouser.com/new/dfrobot/dfrobot-rob0165-brushless-motor-racing-car/.

[3] *Formula Pi – Self-driving robot racing with the Raspberry Pi* [online]. [cit. 2020-05-29]. Dostupné z: https://www.formulapi.com/.

[4] *MG996R High Torque Metal Gear Dual Ball Bearing Servo* [online]. [cit. 2020-05-29]. Dostupné z: https://www.gme.cz/data/attachments/dsh.772-293.1.pdf.

[5] *NVIDIA's DIY Autonomous Car Race* [online]. [cit. 2020-05-29]. Dostupné z: https://developer.nvidia.com/embedded/diy-ai-race.

[6] *NXP CUP* [online]. [cit. 2020-05-29]. Dostupné z: https://nxpcup.nxp.com/.

[7] *PIXY* [online]. [cit. 2020-05-29]. Dostupné z: https://pixycam.com/.

[8] *TWO REVOLUTIONARY OPTICAL TECHNOLOGIES* [online]. 2009 [cit. 2020-05-29]. Dostupné z: https://www.nobelprize.org/uploads/2018/06/advanced-physicsprize2009.pdf.

[9] *Elektro: odborný časopis pro elektrotechniku*. Praha: FCC PUBLIC s. r. o.,., 2013-1-7. ISSN 1210-0889.

[10] *Elektro: odborný časopis pro elektrotechniku*. Praha: FCC PUBLIC s. r. o.,., 2013-2-7. ISSN 1210-0889.

[11] *Elektro: odborný časopis pro elektrotechniku*. Praha: FCC PUBLIC s. r. o.,., 2013-3-12. ISSN 1210-0889.

[12] *Elektro: odborný časopis pro elektrotechniku*. Praha: FCC PUBLIC s. r. o.,., 2013-4-4. ISSN 1210-0889.

[13] *Elektro: odborný časopis pro elektrotechniku*. Praha: FCC PUBLIC s. r. o.,., 2013-5-2. ISSN 1210-0889.

[14] *Elektro: odborný časopis pro elektrotechniku*. Praha: FCC PUBLIC s. r. o.,., 2013-6-6. ISSN 1210-0889.

[15] *Elektro: odborný časopis pro elektrotechniku*. Praha: FCC PUBLIC s. r. o.,., 2013-7-8. ISSN 1210-0889.

[16] *930 kV Motor Outrunnerový Bezkartáčový A2212/15T* [online]. 2017 [cit. 2020-05-29]. Dostupné z: https://arduino-shop.cz/arduino/1630-930-kv-motor-outrunnerovy-bezkartacovy-a2212-15t.html.

[17] *BLDC motor outrunnerový 930 kV A2212/15T* [online]. 2017 [cit. 2020-05-29]. Dostupné z: https://arduino-shop.cz/docs/produkty/0/117/1496220893.pdf.

[18] *Kinetis K66 Sub-Family* [online]. 2017 [cit. 2020-05-29]. Dostupné z: https://www.nxp.com/docs/en/data-sheet/K66P144M180SF5V2.pdf.

[19] *NXP CUP – Training tracks* [online]. 2019 [cit. 2020-05-29]. Dostupné z: https://community.nxp.com/servlet/JiveServlet/download/101612-7-431006/NXP+CUP_track_configurations_2018_19.pdf.

[20] *The NXP Cup Official Rules* [online]. 2019 [cit. 2020-05-29]. Dostupné z: https://community.nxp.com/docs/DOC-335269.

[21] *PX4 Toolchain* [online]. 2019 [cit. 2020-05-29]. Dostupné z: https://nxp.gitbook.io/nxp-cup/developer-guide/development-tools/rddrone-fmuk66-development/px4-toolchain.

[22] *FRDM-KL25Z: Freedom Development Platform for Kinetis® KL14, KL15, KL24, KL25 MCUs* [online]. 2020 [cit. 2020-05-29]. Dostupné z: https://www.nxp.com/design/development-boards/freedom-development-boards/mcu-boards/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z.

[23] *FRDM-KL46Z: Freedom Development Platform for Kinetis® KL3x and KL4x MCUs* [online]. 2020 [cit. 2020-05-29]. Dostupné z: https://www.nxp.com/design/development-boards/freedom-development-boards/mcu-boards/freedom-development-platform-for-kinetis-kl3x-and-kl4x-mcus:FRDM-KL46Z.

[24] *PX4 Robotic Drone FMU (RDDRONE-FMUK66)* [online]. 2020 [cit. 2020-05-29]. Dostupné z: https://www.nxp.com/design/designs/px4-robotic-drone-fmu-rddrone-fmuk66:RDDRONE-FMUK66.

[25] *Track Materials and Build Tools* [online]. 2020 [cit. 2020-05-29]. Dostupné z: https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-build-your-track-materials-and-tools.html.

[26] *Working with the Example-Application* [online]. 2020 [cit. 2020-05-29]. Dostupné z: https://nxp.gitbook.io/nxp-cup/developer-guide/development-tools/rddrone-fmuk66-development/commissioning-the-rddrone-fmuk66/the-example-application.

[27] Anderson, C. *1/10th scale racetrack description and rules* [online]. 2017 [cit. 2020-05-29]. Dostupné z: https://diyrobocars.com/110th-scale-race-rules/.

[28] Anderson, C. *1/16th-scale racetrack description and rules* [online]. 2017 [cit. 2020-05-29]. Dostupné z: https://diyrobocars.com/116th-scale-racetrack/.

[29] Brejl, M. *Freescale Race Challenge 2011 - Další ročník soutěže samořídících autíček na autodráhu* [online]. 2010 [cit. 2020-05-29]. Dostupné z: https://vyvoj.hw.cz/teorie-a-praxe/freescale-race-challenge-2011-dalsi-rocnik-souteze-samoridicich-auticek-na-autodrahu.

[30] Dobeš, M. *Zpracování obrazu a algoritmy v C#*. 1. vyd. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-233-6.

[31] Franklin, W. R. *PNPOLY - Point Inclusion in Polygon Test* [online]. 2018 [cit. 2020-05-29]. Dostupné z: https://wrf.ecse.rpi.edu/Research/Short_Notes/pnpoly.html.

[32] Happich, J. *Robotics makes inroads into education* [online]. 2013 [cit. 2020-05-29]. Dostupné z: https://www.eenewseurope.com/news/robotics-makes-inroads-education.

[33] Hughes, A. a Drury, B. *Electric Motors and Drives: Fundamentals, Types and Applications*. 5. vyd. Oxford: Elsevier, 2019. ISBN 978-0-0810-2615-1.

[34] Petruzella, F. D. *Electric motors and control systems*. Second edition. New York, NY: McGraw-Hill Education, [2016]. ISBN 978-0-07-337381-2.

[35] Roubíček, O. *Elektrické motory a pohony: příručka techniky, volby a užití vybraných druhů*. 1. vyd. Praha: BEN - technická literatura, 2004. ISBN 80-7300-092-x.

[36] Zhang, W., Li, H., Yan, X. a Liu, Z. A Method of Recognizing Curve Direction Based on Hough Transform. In: *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*. IEEE, 2016, sv. 2, s. 3–6. DOI: 10.1109/ISCID.2016.2010.

Příloha A

Obsah přiloženého paměťového média

Struktura paměťového media je následující:

```
src
    pcb
    RDDRONE-FMUK66 firmware
    text
    Ultrasonic sensor firmware
bin
    nxp_fmuk66_v3_default.px4
video
xstein08.pdf
```

Složka `src` obsahuje soubory se zdrojovým kódem. V podložce `pcb` je schéma a návrh desky plošných spojů systému pro detekci plošných spojů. Dále jsou přiložena gerber data a CAM soubor, kterým byla gerber data pro výrobu ve firmě Gatema generována. Pro rychlý náhled desky je zde vyrenderovaný obrázek horní i spodní desky. V podsložce `RDDRONE-FMUK66 firmware` jsou zdrojové soubory pro desku RDDRONE-FMUK66. Jedná se o repositář z Githubu [1] doplněný o můj kód. Můj kód lze nalézt v adresáři `/src/RDDRONE-FMUK66 firmware/src/examples/nxpcup`. V podsložce `text` jsou zdrojové kódy k této textové práci. Podložka `Ultrasonic sensor firmware` obsahuje zdrojové kódy pro firmware mnou navrženého senzoru.

Ve složce `bin` v kořenovém adresáři se nachází zkompilovaný firmware pro desku RDDRONE-FMUK66.

Složka `video` obsahuje video ukázky autíčka jezdícího po různých drahách. Soubor `okruh.mp4` demonstruje jízdu po jednoduchém ovále. Soubor `prekazka.mp4` ukazuje jízdu po jednoduchém okruhu, kdy na dráze leží překážka. Demnonstrace překážky, kterou nelze objet je v souboru `velka_prekazka.mp4`. Jízda na okruhu, ve kterém je úsek se sníženou rychlostí je v souboru `rychlost.mp4`. Nejsložitější dráha obsahující i křižovatku je v souboru `krizovatka.mp4`.

V kořenovém adresáři se nachází pdf verze textové práce.

[1] https://github.com/PX4/Firmware

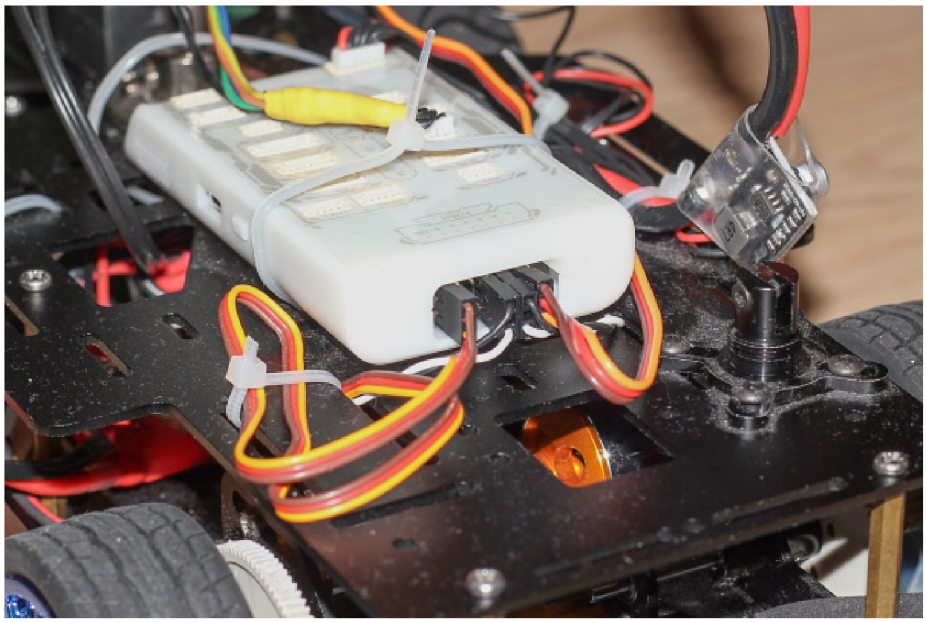
Příloha B

Zprovoznění modelu autíčka

B.1 Zapojení jednotlivých komponent

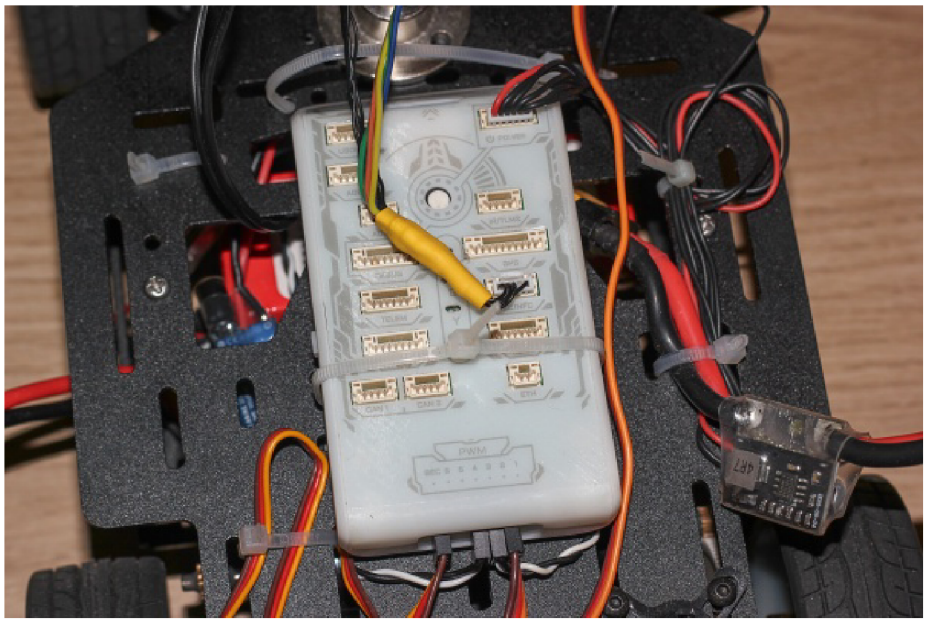
Pro zprovoznění modelu je potřeba provést správné zapojení. Popis zapojení je v kapitole 6, zde bude popsáno praktické zapojení.

Do desky RDDRONE-FMUK66 se do řady pinů na boční straně (obrázek B.1) přípojí zleva přídavné napájení pro servo motor, poté levý ESC, pravý ESC a nakonec servo motor. Napajení je již připraveno, všechny spoje jsou připájeny a zaizolovány.

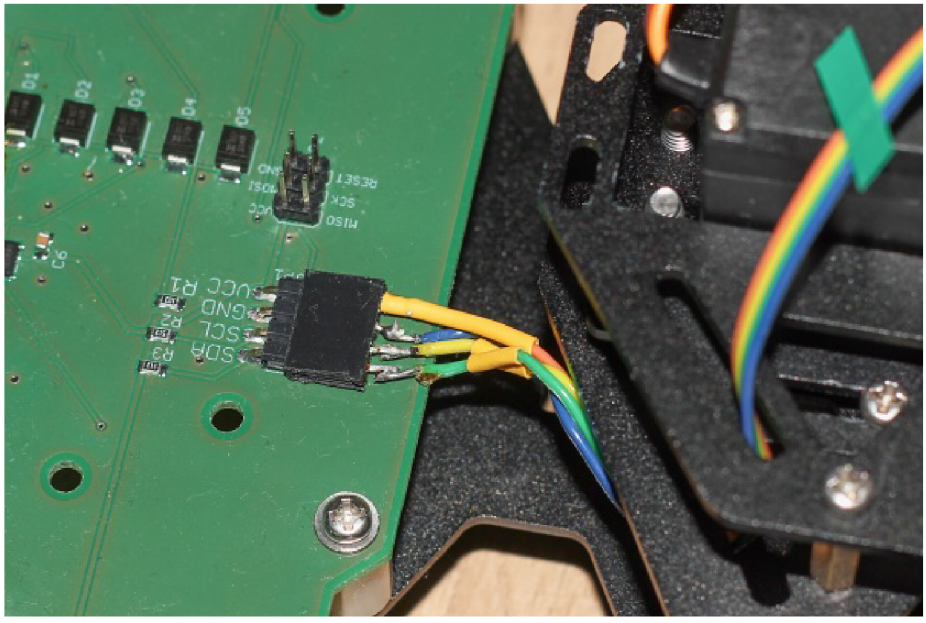


Obrázek B.1: Fotografie zapojení do řady pinů na boku desky RDDRONE-FMUK66.

Na horní stranu desky je připojena kamerka a podpůrný systém pro detekci překážek. Tyto dva senzory jsou připojeny kabelem upraveným pro připojení dvou I2C zařízení do jednoho konektoru s popiskem I2C/NFC. Zapojení je na obrázku B.2. Opačné konce kabelu jsou připojeny do kamerky Pixy2 a také do senzoru pro detekci překážek (obrázek B.3).

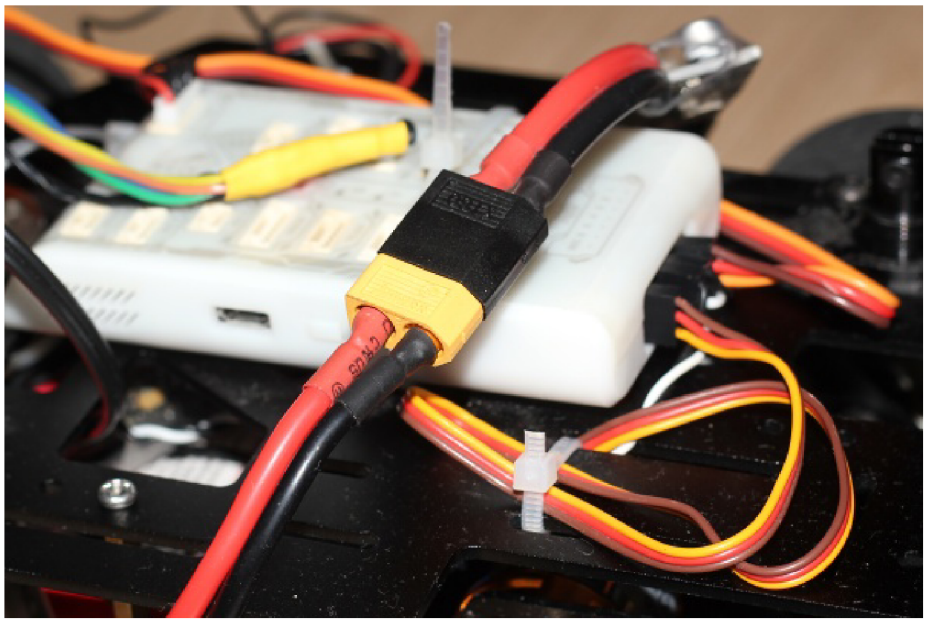


Obrázek B.2: Fotografie zapojení horní strany desky RDDRONE-FMUK66



Obrázek B.3: Fotografie připojení navrženého podpůrného systému.

Pro připojení celého autíčka na napájení je potřeba propojit černý konektor, který vede od regulátoru pro řídící desku a žlutý konektor, který je na vodiči vedoucím od akumulátoru. Takto propojené napájení je na obrázku B.4



Obrázek B.4: Fotografie připojeného napájení

B.2 Nahrání firmware

Pro nahrání firmware do řídící desky je potřeba nainstalovat PX4 toolchain. Návod je dostupný na webu NXP Cup[21]. Je předpokládána instalace na linuxovou distribuci Ubuntu. Dále jsou uvedeny jednotlivé příkazy:

```
sudo usermod -a -G dialout $USER
sudo apt install screen
wget https://raw.githubusercontent.com/PX4/Devguide/master/build_scripts/ubuntu_sim_nut
source ubuntu_sim_nuttx.sh
```

Následně je možné přeložit obsah adresáře `/src/RDDRONE-FMUK66/` příkazem

```
make nxp_fmuk66-v3_default
```

Pro nahrání binárního souboru do desky lze použít příkaz, který kód přeloží a nahraje:

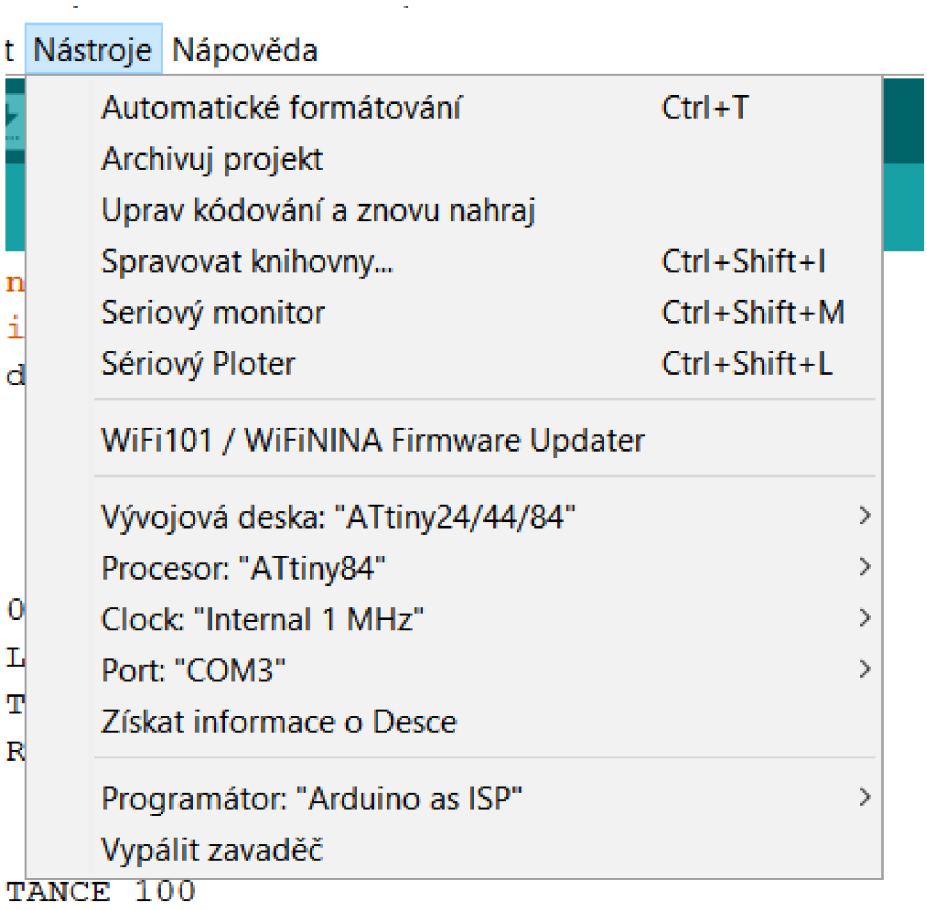
```
make nxp_fmuk66-v3_default upload
```

Pro překlad a nahrání firmware podpůrného systému pro detekci překážek je využito Arduino IDE. Pro přidání podpory mikrokontroléru ATtiny84 je třeba otevřít kontextové menu Soubor a vybrat Vlastnosti. Zde do položky „Správce dalších desek URL“ přidat následující adresu:

```
https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_dame
```

Následně lze ve správci desek nainstalovat desku s názvem `attiny`. Protože nevlastním programátor, kterým bych mohl do Attiny84 nahrávat firmware, používal jsem Arduino Uno v režimu „Arduino as ISP“. Součástí příkladů v Arduino IDE je `ArduinoISP`. Tento kód byl nahrán do Arduina Uno a následně bylo možné k němu připojit Attiny84. K tomuto účelu lze využít ISCP piny na Arduinu. Na mnou navržené desce je tento konektor vyveden

ve stejném rozložení. Pouze reset signál se na Arduinu Uno musí připojit místo do ISCP konektoru do pinu D10. V Arduino IDE se nastaví následující konfigurace desky:



Obrázek B.5: Nastavení Arduino IDE pro nahrávání firmware do Attiny84

Pro instalaci použitých knihoven je potřeba obsah složky

`/src/Ultrasonic sensor firmware/libraries/`

zkopírovat do domovského adresáře Arduino do složky `libraries`. Následně lze přeložit a nahrát firmware do mikrokontroléru.