VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Mechanizmy řízení robotického auta NXP

Driving Mechanisms of Robotic Car NXP

2020 Jan Ptáček

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jan Ptáček**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Mechanizmy řízení robotického auta NXP
Driving Mechanizms of Robotic Car NXP

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je vytvořit software pro ovládání robotického auta NXP model „Alamak" s vývojovým kitem FRDM-K64. Software umožní účast na soutěži NXP Cup a ve vybraných dodatečných disciplínách. Konkrétně v disciplínách detekce oblasti se sníženou rychlostí a průjezd trati tvaru „8" na rychlost.

Software umožní:
1. Detekci dráhy a okrajových čar.
2. Připojení a využití senzoru pro měření rychlosti.
3. Dynamické přizpůsobení světelným podmínkám dráhy.
4. Řízení auta po závodní trati.
5. Splnění vybraných dodatečných disciplín ze závodů NXP. A to konkrétně detekce oblasti se sníženou rychlostí a průjezd trati tvaru „8" na rychlost.

Práce bude obsahovat:
1. Přehled používaných metod a algoritmů.
2. Implementaci výše popsané funkcionality.
3. Experimenty a vyhodnocení výsledků.
4. Dokumentaci programového řešení s využitím diagramů jazyka UML.

Seznam doporučené odborné literatury:

[1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (Gang of Four): Návrh programů pomocí vzorů. Grada. Praha 2003. ISBN 8024703025
[2] DARWIN, Ian F. Java cookbook. 2nd ed. Sebastopol, CA: O'Reilly, c2004, xxiv, 829 p. ISBN 05-960-0701-9. Dostupné z: http://it-ebooks.info/book/2249/

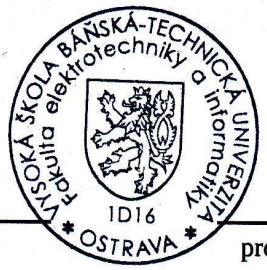
Dále dle pokynů vedoucího práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Ježek, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 13. července 2020 .....

Chtěl bych na tomto místě poděkovat vedoucímu bakalářské práce Ing. Davidu Ježkovi, Ph.D., Ing. Petru Olivkovi, Ph.D. a Bc. Richardu Zvonkovi za jejich spolupráci, pomoc, rady, připomínky a poskytnutí hardwarového i softwarového základu.

Abstrakt

Tato bakalářská práce se zabývá problematikou autonomního řízení modelu auta s účelem účasti na soutěži NXP CUP včetně dodatečných disciplín. Software za pomocí vstupních dat řádkové kamery a přídavných senzorů (AIR) detekuje okrajové čáry závodní dráhy, černé čáry na dráze signalizující cílovou linii, začátek a konec zóny s omezenou rychlostí. Model auta je NXP s podvozkem Alamak, jádrem je vývojová platforma FRDM-K66F rozšířená o shield POLI-TFC-K66 v2.1. Software je psaný v jazyce C++ v IDE MCUXpresso.

Klíčová slova: Autonomní, Řízení, Auto, Autíčko, Čidlo, Senzor, Snímač, Soutěž, Závod, Dráha, Okruh, Disciplína, Kamera, Zpracování, Obraz, Rozpoznávání

Abstract

This bachelor thesis is concerned with autonomous driving of a model car with a purpose of attending car racing competition called NXP CUP including additional challenges. Software, with the help of a line camera and infrared sensors, detects black lines on the edges of the track, the pattern of start and finish line aswell as the pattern signalising start and end of the speed limit zone. The car model is NXP Alamak, core of the project is development platform FRDM-K66F, extended with shield POLI-TFC-K66 v2.1. Software is written in programming language C++ in IDE MCUXpresso.

Keywords: NXP, Alamak, Autonomous, Self-driving, Car, Model, Cup, Competition, Race, Sensor, IR, Track, Circuit, Image, Recognition, FRDM-K66F, TFC, Freescale

Obsah

Seznam použitých zkratek a symbolů

AIR	–	Active Infra Red
MCU	–	Microcontroller Unit
MPU	–	Microprocessor Unit
TFC	–	The Freescale Cup
EMEA	–	Europe, the Middle East and Africa
IR	–	Infrared
IDE	–	Integrated Development Environment
LiPo	–	Lithium-Polymer
NiCd	–	Nikl-kadmio
NiMH	–	Nikl-metal hydrid
Li-ION	–	Lithium-iont
CCD	–	Charge-coupled device
ESC	–	Electronic speed control
IoT	–	Internet of Things
Wi-Fi	–	Wireless Fidelity
SRAM	–	Static Random Access Memory
USB	–	Universal Serial Bus
FEI	–	Fakulta elektrotechniky a informatiky
VŠB-TU	–	Vysoká škola báňská - Technická univerzita
LED	–	Light-Emitting Diode
GPIO	–	General-purpose input/output
RGB	–	Red Green Blue
SD	–	Secure Digital
DIP	–	Dual In-Line Package

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Zájem o problematiku autonomního řízení vozidel v dnešní době rapidně stoupá. Díky novým vyspělým technologiím, digitalizací automobilového průmyslu a pokusům je jasné, že automatizace řízení může výrazně zlepšit bezpečnost silničního provozu redukováním chyb zapříčiněných lidským faktorem. I když jsme v prvopočátcích automatizace a řízení stále vyžaduje výrazný podíl člověka, tak je vidět posun k vyšším stupňům na škále autonomního řízení. Autonomní řízení se rozděluje na šest úrovní automatizace:

Úroveň 0 – vůz bez jakýchkoli asistenčních funkcí

Úroveň 1 – přítomnost asistenčních funkcí jako ABS nebo parkovacího asistenta

Úroveň 2 – částečná automatizace řízení včetně zrychlení a zatáčení

Úroveň 3 – možnost zapnout autopilota nebo převzít řízení, když je potřeba

Úroveň 4 – řízení je automatické, pouze za výjimečných okolností zasáhne řidič

Úroveň 5 – plná automatizace všech funkcí řízení za všech okolností

Cílem dnešní vědecké společnosti je dosáhnout v automatizaci řízení 5. úrovně, tedy plné automatizace všech funkcí řízení za všech okolností. V této práci se dočtete o autonomním řízení modelu auta Alamak, určeného pro soutěž NXP CUP. Pro tento ročník 2019/2020 byl na katedře informatiky FEI VŠB-TU Ostrava vytvořen tříčlenný tým, skládající se z autora této práce, Jana Ptáčka, Frederika Zajace a Bc. Richarda Zvonka, autora stejnojmenné bakalářské práce z minulého roku(Viz. [1]), o jehož práci se tato práce opírá. Cílem práce je vytvořit software pro ovládání robotického auta NXP model „Alamak“ s vývojovým kitem FRDM-K66F. Software umožní účast na soutěži NXP Cup včetně vybraných dodatečných disciplín, konkrétně v disciplínách detekce oblasti se sníženou rychlostí a průjezd trati tvaru číslice 8 na rychlost.

2 NXP CUP

2.1 NXP

NXP Semiconductors N.V. je nizozemsko-americká společnost, vyrábějící polovodičové součástky. Centrála společnosti je v Eindhovenu, Nizozemsko, operující ve více než 33 zemích, s circa 31000 zaměstnanci a více než 100 pobočkami. [2] NXP se zaměřuje na výzkum, rozvoj a inovaci v automobilovém a mobilním průmyslu, komunikační infrastruktuře a IoT. [3]

2.2 Změny oproti minulému ročníku (2018/2019)

- Účastníci si nyní mohou zakoupit vlastní car kit a vybavit si ho elektronickými součástkami, povolenými NXP.
- Nově jsou povoleny bezkartáčové motory a LiPo baterie.
- Soutěžící mají povoleno si vytvořit svoje vlastní autíčko (např. vytisknuté 3D tiskárnou nebo vyřezané ze dřeva pomocí laseru).
- Všechny limitace rozměrů auta byly odstraněny.
- Vozidlo může mít maximálně 4 kola. Tricykly a dvou-kolová sebe sama balancující vozidla jsou povolena.
- Byla přidána nová disciplína - nouzové brzdění, která by měla pomoci studentům připravit svá autíčka tak, aby nebyli penalizováni za nezastavení za cílovou linií v závodě na čas.
- Čas na projíždění tratí v disciplíně „osmičky“ byl snížen z 90 sekund na 60 sekund z důvodu umožnění účasti více týmů na závodech.
- Baterie typu LiPo byly omezeny na 2S 7,4V a 5500mAh.
- Baterie typu NiCd, NiMH a Li-ION jsou nyní omezeny na 5300mAh. [4]

2.3 O soutěži

Cílem soutěže je, aby studenti demonstrovali své dovednosti v oblasti integrace hardware a programování. NPX Cup EMEA (dříve The Freescale Cup) přináší ve školním roce 2019/2020 novou sezónu, zaměřenou na otevření možností původu autíček, dovolující týmům vytvořit si vlastní modely aut. Také jsou povoleny modely aut z minulých let jako je Model-C nebo Alamak. Povolena jsou všechna auta, pokud jsou jejich elektronické součástky v souladu s pravidly. Účastníci soutěže mohou využít široké nabídky technických řešení společnosti NXP, obsahujících mikrokontroléry, mikroprocesory a senzory. Nehledě na elektronickou konfiguraci auta, jeho rozměry nebo typ motoru, budou všechny týmy soutěžit v jedné kategorii. Vítězem NXP Cup EMEA 2019/2020 bude pouze jeden tým. Týmy se skládají ze dvou nebo tří členů, kteří musí

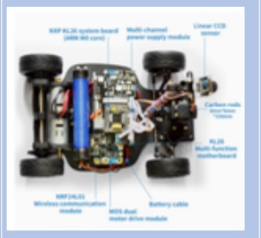
Obrázek 1: Logo a slogan NXP Semiconductors N.V [3]

být studenty jedné z registrovaných škol. Studenti spolu mohou soutěžit v týmu bez ohledu na známky, studijní obor, dosažené vzdělání. Členové týmu mohou být z různých škol, univerzit, asociací nebo klubů. A v neposlední řadě týmy mohou do závodu zaregistrovat více než jedno auto pod jiným názvem týmu, pokud toto auto běží na odlišném MCU nebo MPU. [4] [1]

2.4 Technické požadavky

- Soutěžící mohou:
  1. Využít existující sadu autíčka NXP CUP (DFRobot, Model-C nebo Alamak)
  2. Sehnat komerční sadu (doporučené měřítko modelu je 1/16)
  3. Vytvořit si vlastní autíčko (např. vytisknuté 3D tiskárnou, vyřezávané nebo z lega)
- Auta mohou být poháněna až dvěma bezkartáčovými nebo kartáčovými motory. V případě využití dvou motorů mohou tyto motory být na jedné nápravě (jeden motor pro pravé kolo a jeden pro kolo levé) nebo na rozdílných nápravách (jeden motor pro přední nápravu a jeden pro zadní nápravu).
- Účastníci soutěže mají povoleno využít jakékoliv MCU nebo MPU od NXP nebo dokonce jejich kombinaci. Všechny desky musí nést značku NXP nebo být napájeny MCU/MPU značky NXP. Studenti si mohou vytvořit vlastní desky jako dodatek k deskám poskytnutým ve výchozí sadě (pro Model-C a Alamak).
- Jakákoliv modifikace zakoupené desky musí splňovat níže uvedená pravidla a musí být detailně zdokumentována včetně kusovníku v deníku.

[1] NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

Item	Alamak Kit	Model C kit	DFRobot Brush	DFRobot Brushless
Photo				
Body Structure	Unibody	Segmented Body	Unibody	Unibody
Size	28,5 x 16 x 7 cm	28,5 x 16 x 8 cm	32.5c m length	32.5 cm length
Motors	7.2v 380 brush (2 units)	7.2v 260 brush (2 units)	Brush (2 units) ESC 160A	Brushless (2 units) 1000 Rpm/V
Stearing gear	15 kg cm	6.5 kg cm		
Tire diameter	65mm	50mm	64mm	64mm
Wheel base	16 cm	16 cm	17.3 cm	17.3cm

Obrázek 2: Srovnání parametrů autíček [4]

- Výchozí kamera pro Model-C nebo Alamak může být změněna. Použitá kamera musí být buď vybavena mikrokontrolérem nebo mikroprocesorem značky NXP nebo nesmí být vybavena vůbec žádným vestavěným MCU/MPU.
- Autíčko musí primárně využívat optický senzor (kameru) pro navigaci. Týmy mohou využít dodatečné senzory, aby vylepšily navigaci vozidla.
- Model auta musí být autonomní a nemůže být dálkově ovládán. Během závodu a výzev nesmí být auto vybaveno žádným aparátem, schopným bezdrátové komunikace. Ten je povolen pouze během tréninku, aby umožnil monitorování a diagnostiku vozidla, ale musí být odstraněn z vozidla během oficiálního závodu.
- Účastníci mohou přidat libovolný počet senzorů na auto. V případě, že daný senzor je součástí produktové řady společnosti NXP, pak musí být použit senzor značky NXP. Mezi povolené senzory patří:
  - Infračervený vysílač/přijímač
  - CCD senzor
  - Hallova sonda (jedna na kolo)
  - Enkodéry

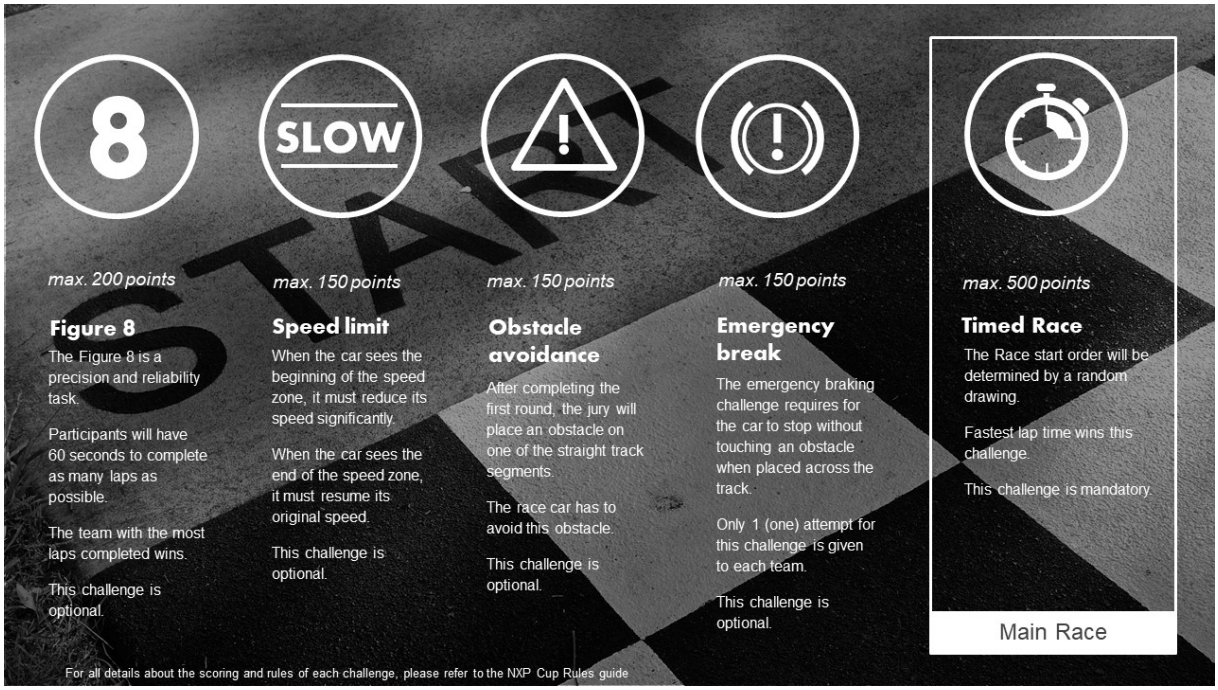
  - Akcelerometr
  - Optické senzory
  - Ultrazvukové senzory
  - Gyroskop
  - Lidar
- Pokud je to nutné, mohou soutěžící přidat komerční elektrický kontrolér rychlosti (ESC). V takovém případě musí být informace a specifikace přidány do logu.
- Požadavky na baterie:
  - Pro napájení vozidla a veškerého připojeného hardware může být využita pouze jedna baterie.
  - Jsou povoleny akumulátory typu NiCd, NiMH a Li-ION s maximální kapacitou 5500mAh.
  - LiPo baterie jsou povoleny, ale limitovány na modely 2S (dva články v sérii), maximální napětí 7,4 V a kapacita 5500mAh.
- Každý tým je povinen poskytnout „log book“ se specifikacemi modelu auta, pokud se nejedná o originální Model-C nebo Alamak. Ten musí obsahovat tyto informace:
  - Značka a model auta, informace o motoru a baterii
  - Informace o desce a elektronice (Model desky s referenčním číslem součástky)
  - Schéma desky, pokud se nejedná o desku s původem v NXP
  - Kusovník
  - Specifické výkonnostní parametry pro danou sadu a elektroniku
- Pro každou disciplínu mohou být na auto namontovány různé senzory a elektronika, pokud to týmy stihnou v časově omezeném okně mezi disciplínami. [4]

2.5 Průběh turnaje

NXP Cup se dělí na kvalifikační kolo a finále. Kvalifikace probíhají ve více zemích z regionu (v tomto roce se kvalifikační kola měla konat v České republice (V Ostravě na FEI-VŠB Ostrava), Francii, Německu, Libanonu, Maroku, Nizozemsku, Rumunsku a Spojeném království v období březen-duben. Finálové kolo se mělo konat v Bukurešti. Prozatím je ale kvůli pandemii koronaviru SARS-CoV-2 (onemocnění COVID-19) celý NXP CUP posunut do druhé poloviny 2020. NXP Cup EMEA 2019/2020 obsahuje jeden hlavní závod na čas a 4 dodatečné nepovinné disciplíny, ve kterých mohou týmy získat další body, které se sčítají. Přehled disciplín můžete vidět na obrázku 4 na straně 18 a bodové ohodnocení disciplín na obrázku 3 na straně 17. Vítězem se pak stává tým s největším počtem nasbíraných bodů. [4]

Figure 8: 90 seconds total completed laps	Figure 8 points	Emergency Braking	Emergency Braking Points	Obstacle Avoidance	Obstacle Avoidance Points	Speed Control	Speed Control Points	Timed Race Best Lap Time	Timed Race Points
1st	200	Success	150	Success	150	Success	150	1st	650
2nd	150							2nd	550
3rd	125							3rd	450
4th	100							4th	400
5th	75							5th	350
6th	50							6th	300
7th	40							7th	250
8th	30							8th	200
9th	20							9th	150
10th	10	No Success	0	No Success	0	No Success	0	10th	100
11th and above	0							11th and above	0

Obrázek 3: Systém hodnocení [4]



Obrázek 4: Disciplíny pro ročník 2019/2020 [5]

2.6 Detaily dráhy

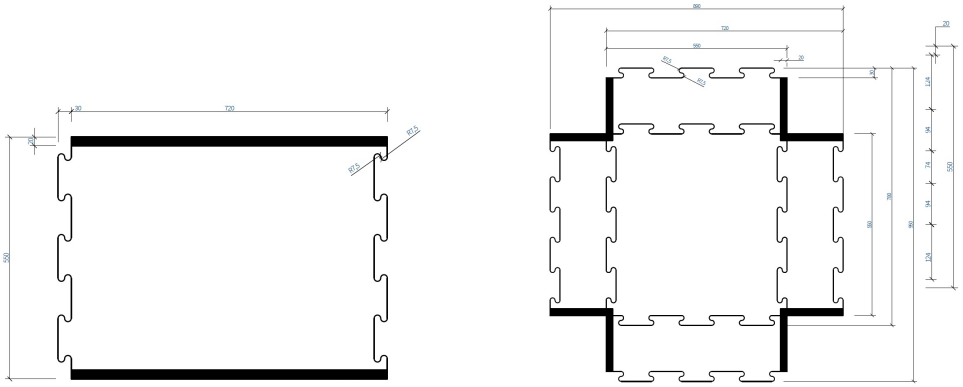
Trať se skládá z rovných úseků (6a), levotočivých zatáček (7b), pravotočivých zatáček (7a), křižovatek (6b) a šikany (8a). Detail startovní a zároveň i cílové linie je na obrázku 8b. Samotné rozložení dráhy je soutěžícím utajeno až do momentu konání disciplín. Šířka tratě je 55 cm, povrch je matně bílý s celistvou dvoucentimetrovou černou linií na každé straně. Trať se může křížit pod úhly 45° a 90°.

2.7 Závod na čas

Závod na čas je hlavní disciplínou soutěže NXP CUP. Každý tým má 2 minuty na konfiguraci parametrů pomocí přepínačů a tlačítek na desce, nastavení úhlu kamery, výměnu baterie popřípadě vyčištění kol. Připojení závodního autíčka k počítači není povoleno. Každý tým má 3 pokusy na dokončení trati. V případě úspěšného projetí se zbylé pokusy neprovedou. Proto je výhodné mít možnost nastavit agresivnější jízdu na první pokus a v případě neúspěchu nastavit pomalejší, ale jistější režim. Časové penalizace v této disciplíně:

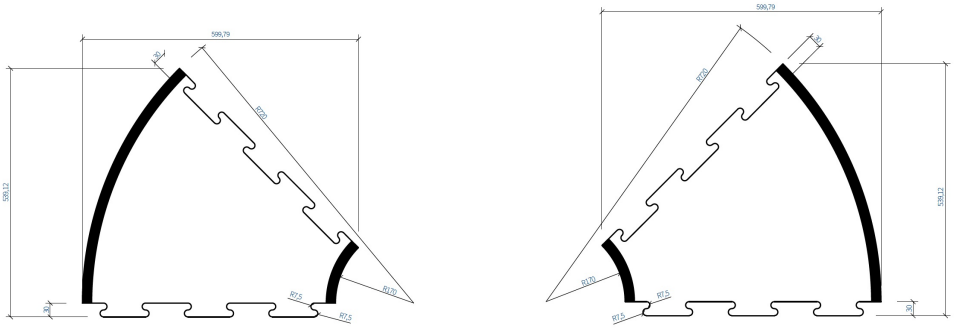
- Auto neopustí startovní lokaci v 30 sekundách po začátku závodu[+1 sekunda]
- Auto nezastaví do 2 metrů za cílovou linií, nebo opustí trať. [+1 sekunda] [4]

Obrázek 5: Rozestavěná dráha u nás na FEI-VŠB v laboratoři (EB428)



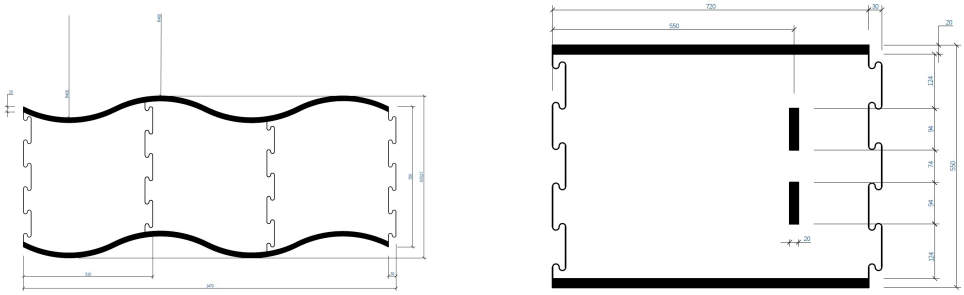
(a) Rovný úsek dráhy (b) Křižovatka

Obrázek 6: Části dráhy 1 [6]



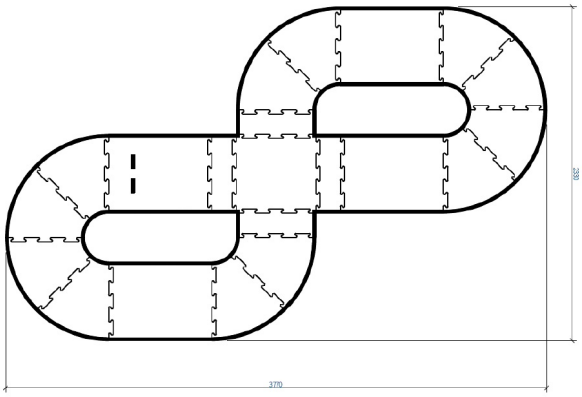
(a) Pravotočivá zatáčka (b) Levotočivá zatáčka

Obrázek 7: Části dráhy 2 [6]



(a) Šikana (b) Startovní/cílová čára

Obrázek 8: Části dráhy 3 [6]



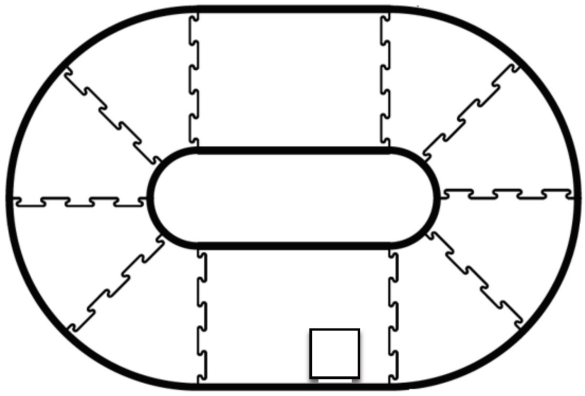
Obrázek 9: Příklad trati pro disciplínu „Trať tvaru čísla 8“ [4]

2.8 Trať tvaru čísla 8

V této disciplíně budou autíčka projíždět tratí tvaru podobnému znaku čísla 8 (Ukázka možné tratě je na obrázku 9 na straně 20). Trať bude obsahovat zatáčky, rovné úseky a křižovatku. Autíčka mají 60 sekund na to, aby projely co nejvíce kol. Žádné kolo nebo část vozidla nesmí opustit dráhu. Soutěžící mají 3 pokusy na splnění této výzvy. Pokud auto opustí trať během těchto 60 sekund, pak je pokus neplatný a tým ztrácí pokus. Pokud je pokus úspěšný, tedy autíčko během 60 sekund nevyjelo z dráhy, pak je počet projetých kol zaznamenán a zbylé pokusy propadají. Tým s největším počtem projetých kol získává nejvíce bodů. Soutěžící mají 30 sekund na nastavení autíčka mezi pokusy. [4]

2.9 Vyhnutí se překážce

Týmy soutěží na malé trati (např.: ovál se dvěma rovnoběžnými rovnými úseky a 180° zatáčkami). Autíčko je postaveno na startovní linii, která se nachází na rovném úseku. Autíčko se rozjede a projede jedno kolo na trati. Během druhého kola rozhodčí umístí překážku na jednu z rovných částí trati. Autíčko se musí vyhnout této překážce. Nesmí se jí dotknout podvozek ani kola. Rychlost v této disciplíně nehraje roli, ale stále platí, že žádná část vozidla nesmí opustit



Obrázek 10: Příklad umístění překážky na trati pro disciplínu „Vyhnutí se překážce“ [4]

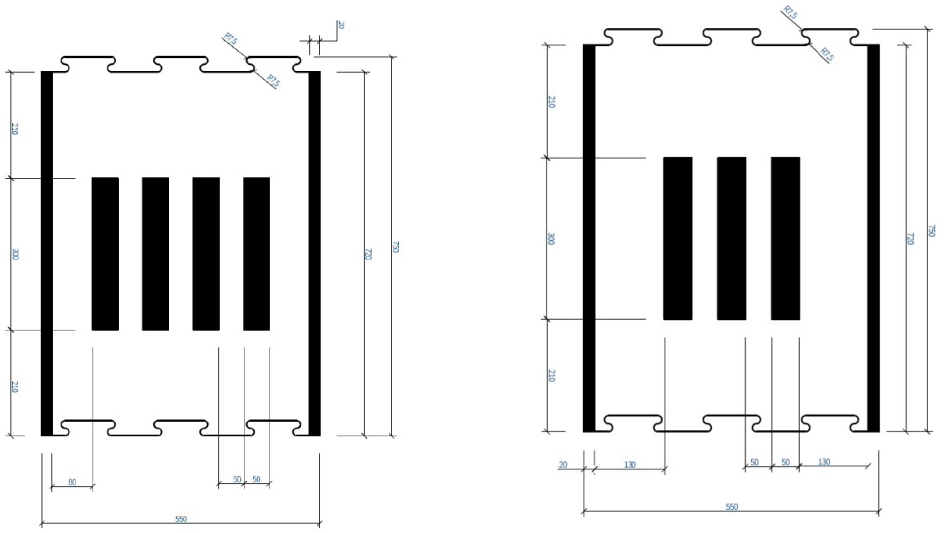
trať. Maximální doba průjezdu tratí je 90 sekund. Překážka je bílá kostka vyrobena z polystyrenu nebo podobného materiálu s rozměry 20 x 20 x 20 cm. Na tuto disciplínu mají týmy pouze jeden pokus. Příklad tratě s možným umístěním překážky viz. obrázek 10 na straně 21. [4]

2.10 Zóna s omezenou rychlostí

Soutěžící soutěží na malé trati (např.: ovál se dvěma rovnoběžnými rovnými úseky a 180° zatáčkami). Auto je umístěno tak, aby začalo těsně za vzorem tří pruhů (viz. obrázek 11b na straně 22). Autíčko musí rychle akcelerovat. Když auto spatří vzor čtyř pruhů (obrázek 11a na straně 22), značící začátek zóny s omezenou rychlostí, musí svou rychlost viditelně zredukovat na zhruba polovinu své původní rychlosti. Když pak auto uvidí vzor tří pruhů, značící konec zóny s omezenou rychlostí, musí zpátky zrychlit na původní rychlost. Opět nesmí auto opustit trať žádnou částí. Rychlost v této disciplíně není podstatná. Auto má až 90 sekund na provedení výše uvedené aktivity. Na tuto disciplínu mají týmy pouze jeden pokus. [4]

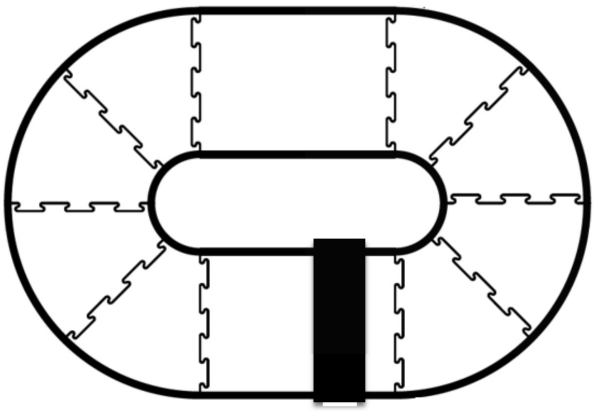
2.11 Nouzové brzdění

Úspěšné splnění disciplíny nouzového brzdění vyžaduje, aby autíčko zastavilo, aniž by se dotklo překážky, umístěné na trať. Auto je umístěno na trať a musí nabrat rychlost. Poté auto musí projet trať (většinou tvaru oválu) dvakrát. Na trati se nenachází žádné značení, jako je např. startovní a cílová linie, nebo rychlostní zóna. Během jízdy auta člen personálu umístí černou překážku přes trať. Rozměry překážky vyrobené z polystyrenu činí 20 x 20 x 60 cm. Tato překážka bude sahat přes celou šířku tratě. Pokud auto zastaví, aniž by se dotklo překážky, je týmu udělena bodová odměna. Pokud se auto dotkne překážky nebo vyjede z trati, pak nejsou uděleny body žádné. Týmy mají na tuto disciplínu pouze jeden pokus.[4]



(a) Začátek zóny s omezenou rychlostí (b) Konec zóny s omezenou rychlostí

Obrázek 11: Vzory na trati při disciplíně „Zóna s omezenou rychlostí“ [4]



Obrázek 12: Příklad trati a překážky pro disciplínu „Nouzové brzdění“ [4]

3 Hardware

3.1 Úvod a model auta

Pro tuto práci a tento ročník soutěže NXP CUP bylo pokračováno na autíčku typu Alamak a software z minulých let. Model Alamak se vyznačuje oproti Modelu-C celistvým tělem, výkonnějšími motory a většími koly. Srovnání modelů je vidět na obrázku 2 na straně 15. Autíčko obsahuje:

1. Vývojovou platformu FRDM-K66F
2. Shield POLI-TFC-K66 v2.1
3. Řádkovou kameru
4. Servomotor
5. 2 stejnosměrné motory
6. Infračervené senzory
7. Akumulátor
8. Wi-Fi modul

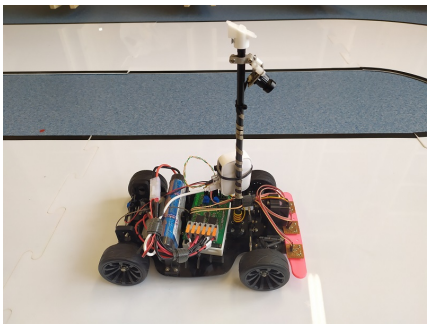
Pro nahrání programu a ladění se využívá Micro-B USB rozhraní pro připojení k počítači. Při jízdě je autíčko napájeno vyměnitelným 7,2 V akumulátorem typu NiMH. Natočení kol přední nápravy ovládá servomotor a zadní kola jsou poháněna dvěma 7,2 V stejnosměrnými kartáčovými motory typu 380.

3.2 FRDM-K66F

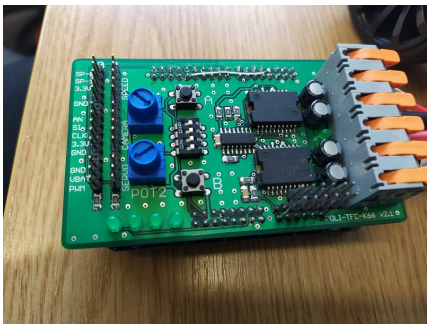
FRDM-K66F je levná vývojová platforma pro Kinetis® K66, K65 a K26 MCU. Taktovací frekvence je 180 MHz, obsahuje 2 MB flash paměti, 256 KB paměti SRAM, dva Micro-B USB konektory, Ethernet port, 54 GPIO pinů, které jsou kompatibilní s Arduinem, akcelerometr a magnetometr (FXOS8700CQ), gyroskop (FXAS21002), slot na Micro SD kartu, dvě tlačítka, set RGB LED, mikrofon, možnost přídavného rádiového a Bluetooth modulu a programovatelné ladící rozhraní JLink OpenSDAv2.1. [7] [8]

3.3 TFC SHIELD

Shield POLI-TFC-K66 v2.1 je rozšiřující deska pro desku FRDM-K66F odvozená z oficiálního TFC Shield pro FRDM-KL25Z [1]. Ta nám unifikuje a sjednocuje rozhraní pro periferie. TFC shield obsahuje dva H-můstky MC33887APVW určené k ovládání motorů, 2 rozhraní pro servomotor, 2 rozhraní pro kameru a dvě rozhraní pro senzory rychlosti. Dále na desce najdeme 2

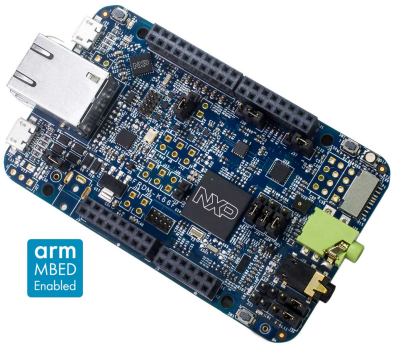


(a) Fotka modelu auta typu Alamak

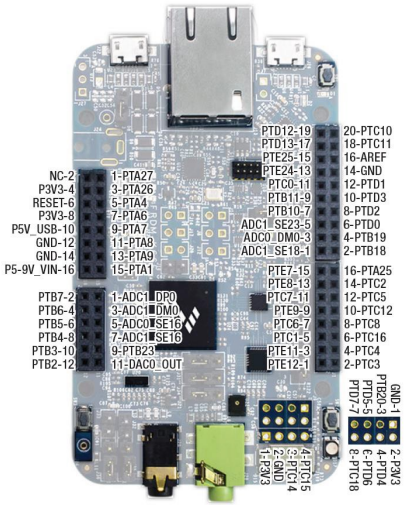


(b) Shield POLI-TFC-K66 v2.1

Obrázek 13: Hardwarový základ



(a) Detail desky FRDM-K66F



(b) Rozložení pinů na vývojové platformě FRDM-K66F

Obrázek 14: MCU Freedom K66F

potenciometry, kterými můžeme nastavovat parametry, 4 DIP přepínače, kterými nastavujeme a přepínáme mezi módy jízdy a jednotlivými disciplínami, 2 tlačítka a 4 LED pro signalizaci.

3.4 Řádková kamera

Řádková kamera, připevněná na stožár ve středu auta, disponuje rozlišením 1x128 bodů. S obrazem této kamery se dá pracovat stejně, jako s obrazem konvenčních 2D kamer, pomocí historie záznamu. Zpracování obrazu probíhá ve čtyřech krocích:

1. Získání obrazu z kamery
2. Filtrování mediánem
3. Normalizace obrazu
4. Prahování průměrem [1]

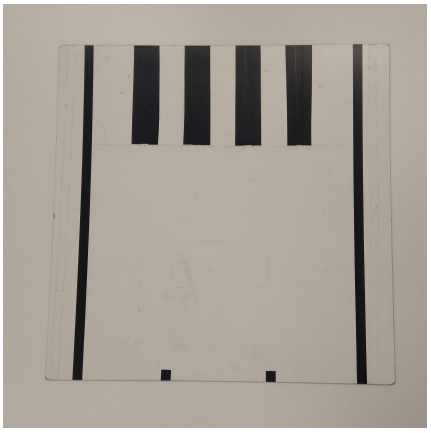
3.5 Kalibrace kamery

Data z kamery jsou náchylná na externí faktory jako například osvětlení místnosti nebo nepořádek na dráze. Z toho důvodu je velice důležité kameru přizpůsobit okolním podmínkám, abychom minimalizovali chyby a dosáhli co nejlepších výsledků. Obraz z kamery je ovlivněn těmito parametry:

1. Náklon kamery
2. Zaostření kamery
3. Natočení kamery

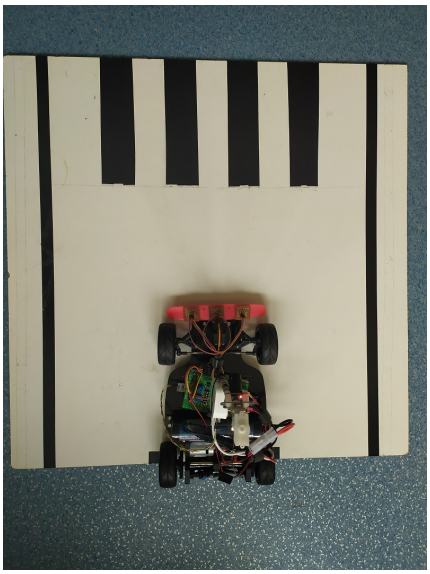
Pro kalibraci kamery byla vytvořena kalibrační deska, na které jsou vyznačeny černou páskou místo pro postavení autíčka, okrajové čáry a vzor pro začátek zóny s omezenou rychlostí. Pro správnou kalibraci, která je nezbytná pro správné fungování programu auta, umístíme auto na vyznačenou pozici na desce (Viz. obrázek 16a, kola jsou mezi značkami, přesahují z poloviny za desku a auto stojí rovnoběžně s čarami) a zaostříme obraz kamery šroubováním objektivu do závitu kamery tak, aby byly všechny pruhy jasně viditelné a co nejširší. Nesmíme zapomenout na náklon kamery - kamera není pevně uchycena a můžeme ji natočit tak, aby snímala prostor blíže nebo dále od auta. Poté autíčko přesuneme za desku tak, aby se jeho kola dotýkala desky, autíčko bylo mezi značkami a rovnoběžně s čarami (Viz. obrázek 17a). V této pozici chceme, aby byl náklon kamery takový, abychom viděli pouze 2 okrajové čáry, kameru již neostříme, ale zabýváme se natočením kamery (Kamera se dá na sloupku otočit doprava nebo doleva, základem je podle oka kameru nasměrovat na střed). Poté se snažíme kameru natočit tak, aby chybovost nalezených regionů zobrazená ve vizualizační aplikaci byla ideálně nulová.

(a) Detail řádkové kamery

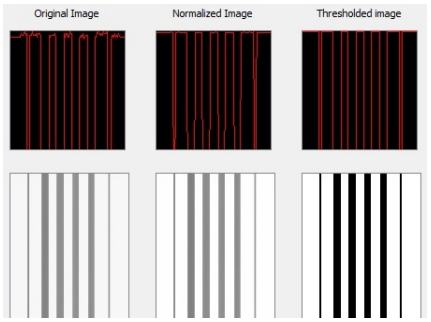


(b) Kalibrační deska [1]

Obrázek 15: Kalibrace kamery

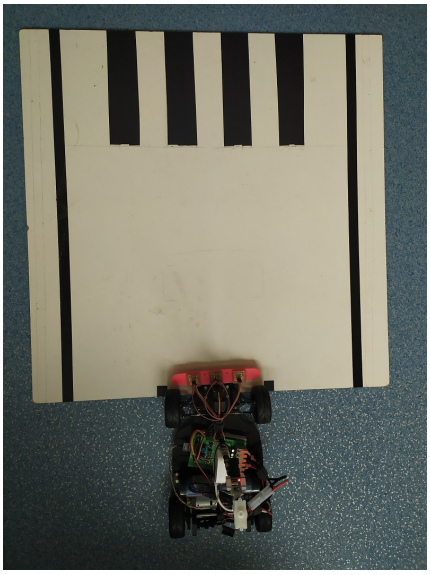


(a) Umístění autíčka na podložce

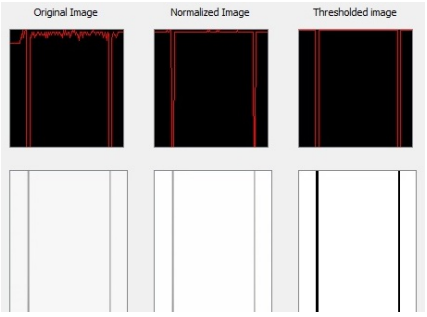


(b) Výsledek po kalibraci

Obrázek 16: Kalibrace kamery - krok první

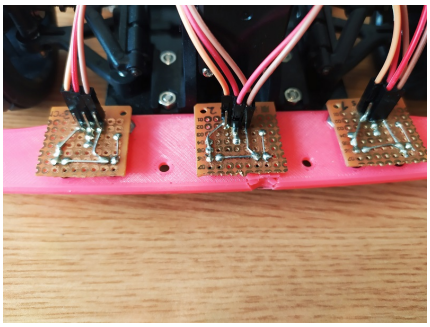


(a) Umístění autíčka na podložce

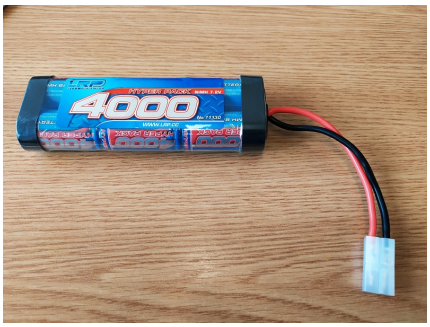


(b) Výsledek po kalibraci

Obrázek 17: Kalibrace kamery - krok druhý



(a) Trojice infračervených senzorů detekujících černé linie na trati



(b) Akumulátor

Obrázek 18: Senzory a napájení

3.6 Infračervené senzory

Pro vyřešení problému nezastavení auta za cílovou linií, který je v soutěži penalizován postihem jedné sekundy, bylo autíčko rozšířeno o tři infračervené senzory v oblasti předního nárazníku. Jejich jádrem je infračervená LED, která vyzařuje infračervené záření. Světlé povrchy toto záření reflektují a tmavé pohlcují. Měřením napětí na kolektoru jsme potom schopni zjistit barvu povrchu pod senzorem. Konkrétně jsou tyto senzory přidány do struktury analogových dat v kódu tfc. Data prochází ADC převodníkem, a poté jsou prahována, protože nás zajímají pouze dva stavy, černá a bílá, tedy čára se nachází nebo nenachází pod senzorem.

(a) Přední strana

(b) Zadní strana

Obrázek 19: Směrovač Nexx wt3020f [6]

3.7 Wi-Fi modul a bezdrátová komunikace

Při vyvíjení software pro autíčko se ukázalo, že by bylo vhodné vidět informace o autíčku za jízdy, což by ulehčilo ladící proces. Proto byl k desce připojen nízkopříkonový směrovač *Nexx wt3020f*, který do bezdrátové Wi-Fi sítě, vytvořené směrovačem v NXP laboratoři, vysílá broadcast na pevně daném portu. Dále byla vytvořena desktopová aplikace „NXP Car Interface“, která poslouchá na daném portu a vizualizuje přijatá data.

4 Software

4.1 MCUXpresso

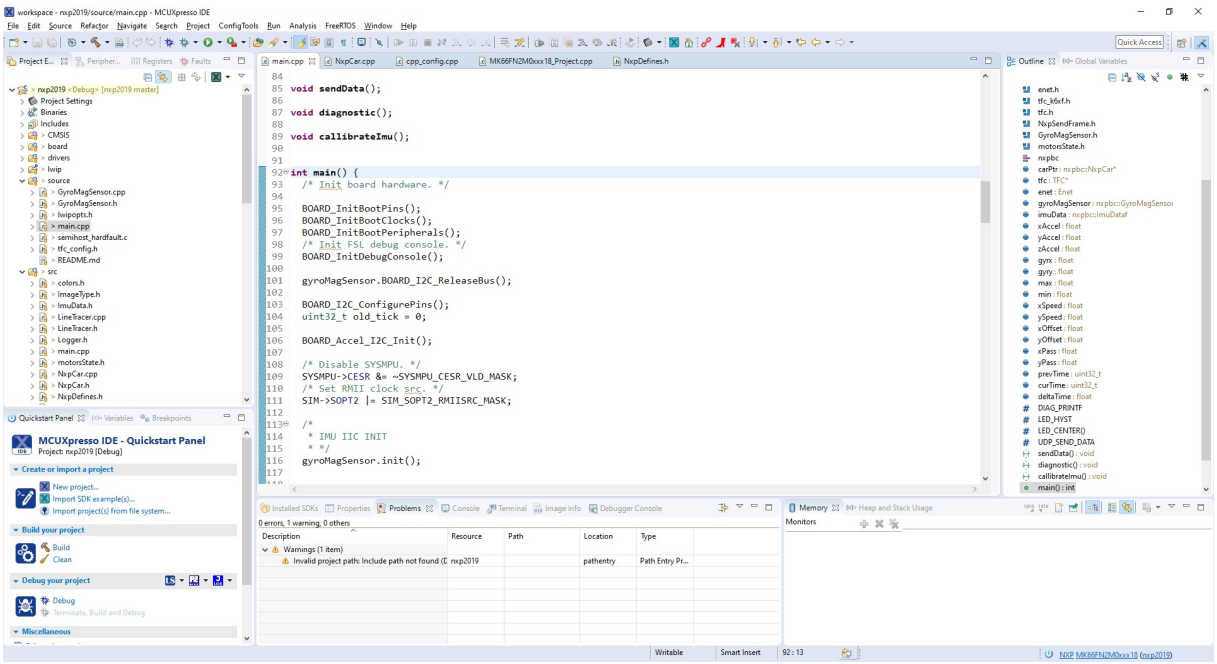
MCUXpresso je integrované vývojové prostředí (IDE), založené na rozšířením IDE „Eclipse“, určené k vývoji software pro mikrokontroléry od NXP. MCUXpresso nabízí rozšířené funkce editování, kompilování a ladění, trasování kódu, profilování, vícejaderné ladění a integrované konfigurační nástroje.

4.2 Vizualizační software

Desktopová aplikace „NXP Car Interface“ slouží k zobrazení dat v reálném čase. Data jsou jednoduše rozšiřitelná. V levé části aplikace je zobrazen originální obraz, normalizovaný obraz a prahovaný obraz, a to i s historií. V pravé části můžeme vidět informace o regionech, ze senzorů - akcelerometru, magnetometru, gyroskopu, infračervených senzorech na nárazníku nebo senzoru rychlosti, o obrazu, stav vozidla, nastavený mód jízdy a informace o motorech. Aplikace byla napsána v jazyce C++ za pomocí knihoven Qt a OpenCV. Pro omezení bezdrátového provozu se přenáší pouze surové data z kamery (Po aplikaci mediánového filtru) a ty se pak v desktopové aplikaci upravují stejnými algoritmy jako v MCU na autíčku. Důležitou vlastností aplikace je její schopnost ukládat hodnoty do .csv souboru, který jsem pak dále využil pro vytvoření grafů. Aplikace tyto hodnoty začne ukládat po zmáčknutí tlačítka B, tedy po rozjezdu autíčka a přestává po zastavení autíčka - tedy opětovné stisknutí tlačítka B anebo samostatné zastavení auta zaznamenáním cílové čáry nebo vyjetím z dráhy. Při každém novém rozjetí je soubor přepsán.

```
enum NxpModes {
  modeDiagBtns = 0b0001,
  modeDiagServo = 0b0011,
  modeDiagMotors = 0b0101,
  modeDiagCam = 0b1001,
  modeSpeedZone = 0b0010,
  modeFigEight = 0b0100,
  modeObstacle = 0b1000,
  modeRideOne = 0b0000,
  modeRideTwo = 0b0110,
  modeRideThree = 0b1010,
  modeRideSetting = 0b1110
};
```

Výpis 1: Výčet módů jízdy a jejich kódů (Nastavených na DIP přepínači)



Obrázek 20: Ukázka vývojového prostředí MCUXpresso

Obrázek 21: Desktopová vizualizační aplikace „Nxp Car Interface“

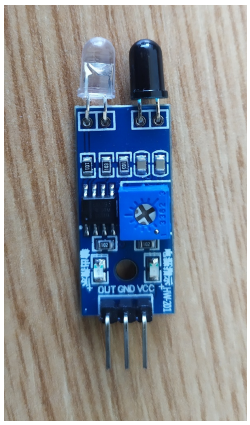
```
/**
* @brief Struktura pro posílání dat přes UDP
*/
struct SendData {
uint16_t image[TFC_CAMERA_LINE_LENGTH] = {0};
uint16_t adc[anLast];
//float voltage = 0.0f;
int16_t leftPwm = 0;
int16_t rightPwm = 0;
int16_t motorSpeed = 0;
int16_t servo = 0;
float error = 0.0f;
uint8_t whiteRegionsCount = 0;
uint8_t blackRegionsCount = 0;
Region regions[SEND_REGIONS_NUM] = {Region()};
Region biggestRegion = Region();
uint8_t bits = 0x00;
/* 0b0000 0001 computed region
 * 0b0000 0010 turning
 */
uint8_t regionAverage[2] = {0};
uint8_t regionMedian[2] = {0};
int16_t imageDiversity = 0;
float accelValues[3];
float gyroValues[3];
float magnetoValues[3];
float angle;
float my_speed;
uint8_t currentState = 0;
NxpModeSetting modeSetting_ = constRide1Setting;
};
```

Výpis 2: Struktura dat posílaných bezdrátově

5 Měření rychlosti auta

5.1 IR Senzor pro měření rychlosti

Jeden z problémů projektu byla nemožnost zjistit aktuální rychlost autíčka. Pro vyřešení problému s průběžným měřením rychlosti autíčka byl využit infračervený senzor FC-51 (Viz. obrázek 22a na straně 32). Základní prvek tohoto senzoru je LED emitor, vyzařující infračervené záření (na obrázku průhledná baňka) a přijímač (černá baňka). Tmavé povrchy potom infračervené záření pohlcují, zatímco světlé záření reflektují. Tento senzor je umístěn k ozubenému kolu na zadní nápravě autíčka, aby v něm detekoval díry. Senzor dále obsahuje potenciometr, kterým se nastavuje vzdálenost, na kterou bude schopný senzor detekovat objekt. Dále LED signalizující napájení senzoru a LED detekce překážky.



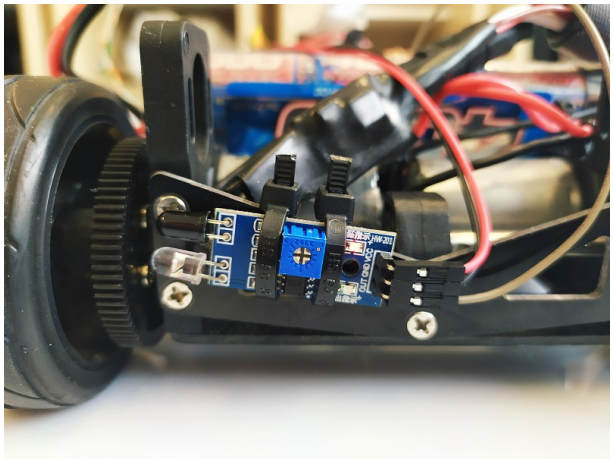
(a) IR senzor FC-51

(b) Ilustrační fotografie ozubeného kola [9]

Obrázek 22: Řešení měření rychlosti

5.1.1 Přerušení

Jelikož procesorový čas je drahý a procesor je neustále zaneprázdněn výpočty, nejlepším možným způsobem řešení problému obsluhy senzoru je přerušení. Do projektu byl implementován kód, který definuje vyvolání přerušení při sestupné hraně na GPIO pinu, na který je připojen digitální výstup senzoru. Ve funkci obsluhující přerušení se zkontroluje, že přerušení bylo vyvoláno správně a poté se inkrementuje počítadlo děr. Výstup senzoru je připojen na GPIO pin PTB3, proto je jeho obsluha řešena přerušením na GPIOB. Měření času a spouštění samotného výpočtu rychlosti se řeší přes přerušení na „Periodic Interrupt Timer (PIT)“, kde je vyvoláno přerušení každých 10 mikrosekund.



Obrázek 23: Provizorně připevněný IR senzor snímající kolo.

5.2 Výpočet rychlosti

Pokud zaznamenáme 9 děr, můžeme předpokládat, že autíčko urazilo dráhu rovnou obvodu kola. Tedy když zaznamenáme jednu díru, můžeme z toho odvodit, že autíčko se posunulo o jednu devítinu obvodu kola. Dále víme ze specifikace modelu Alamak, že průměr kola je 65 mm (hodnotu jsem ověřoval měřením). Z průměru si vypočteme obvod kola a po dělení obvodu počtem děr získáme vzdálenost uraženou za každou zaznamenanou díru. Čas získáváme pomocí přerušení, známe tak všechny potřebné údaje k výpočtu rychlosti. Výpočet rychlosti je pak proveden dle vzorce 1, kde:

d – Průměr kola v metrech.

i – Počet zaznamenaných objektů

t – Naměřený čas v sekundách.

c – Počet inkrementací počítadla během jednoho otočení kola.

$$\frac{\pi \cdot d \cdot i}{t \cdot c} \tag{1}$$

NxpDefines.h

```
/ * @brief Pi
 */
/**
 * @brief How many times an interrupt will be triggered per every rotation of
    the wheel.
 */
#define NUMBER_OF_INTERRUPTS_PER_ROTATION 9

/**
 * @brief Time in milisecond of each PIT Time interupt.
 */
#define PIT_TIME_CONST 100

/**
 * @brief Wheel diameter in milimeters.
 */
#define WHEEL_DIAMETER_M 65

/**
 * @brief How often will the program calculate and update speed.
 */
#define SPEED_CALCULATION_PERIOD 100000

/**
 * @brief Minimal time that the sensor must continuously output logical one in
    order to increment the counter;
 */
#define MINIMUM_TIME_IN_LOGICAL_ONE 1600
/**
 * @brief Maximální hodnota rychlosti namerene na senzory rychlosti
 */
#define SENSOR_SPEED_MAX 3
/**
 * @brief Hodnota PWM kroku pro senzor rychlosti
 */
#define CONTROL_PWM_STEP_SENSOR 5
```

Výpis 3: Definice konstant

```
int time_us = 0;
int counter = 0;
int time_spent_in_blackzone = 0;
float my_speed = 0;

extern "C" {

   void PIT0_IRQHandler(void){
       PIT_ClearStatusFlags(PIT, kPIT_Chnl_0, kPIT_TimerFlag);
      time_us += PIT_TIME_CONST;
      time_spent_in_blackzone += PIT_TIME_CONST;
      if(carPtr->getSpeedFlag())
      {
         if(carPtr->getDebounceTimer() <= 0)
         {
            carPtr->setSpeedFlag(false);
         }
         carPtr->lowerDebounceTimer(PIT_TIME_CONST);
      }
      if(time_us == SPEED_CALCULATION_PERIOD)
      {
         my_speed = ((((PI * (WHEEL_DIAMETER_M / 1000.0)) /
             NUMBER_OF_INTERRUPTS_PER_ROTATION ) * counter) / (time_us /
             1000000.0));
         carPtr->setSensorSpeed(my_speed);
         counter = 0;
         time_us = 0;
      }
   }

   void GPIOB_IRQHANDLER(void)
   {
      uint32_t flags = GPIO_PortGetInterruptFlags(GPIOB);
      if((flags & (1U << 3)) > 0)
      {
          GPIO_PortClearInterruptFlags(GPIOB, 1U << 3);
          if(GPIO_PinRead(GPIOB, 3) == 0)
          {
```

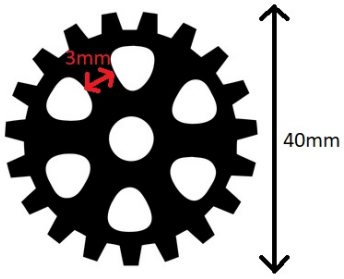
```
            if(time_spent_in_blackzone >= MINIMUM_TIME_IN_LOGICAL_ONE)
            {
               counter++;
            }
            time_spent_in_blackzone = 0;
         }
      }
   }
}
```

Výpis 4: Obsluha přerušení, řešení zákmitů a výpočet rychlosti

5.3 Problém se zákmity

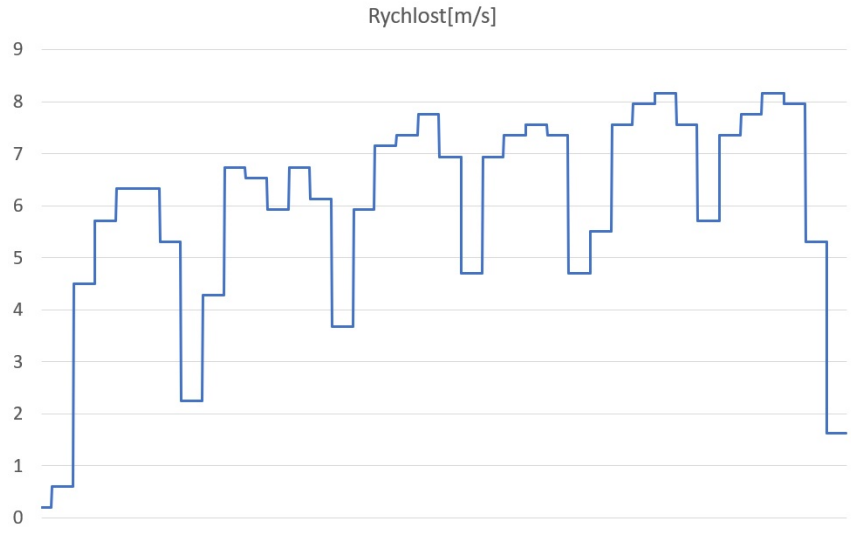
Zapojení senzoru proběhlo bez problémů. Přerušení funguje jak má. Struktura posílaných dat (viz. 2 na straně 31) byla rozšířena tak, aby obsahovala průběžnou rychlost. Můžeme tedy za jízdy kontrolovat naměřenou rychlost a provést experimenty. Problémy s infračerveným senzorem jsem odhalil během testování funkčnosti přerušení. GPIO pin PTB3, na který jsem připojil digitální výstup senzoru, je nastaven tak, aby vyvolal přerušení. Nezáleží však, jestli je přerušení nastaveno, aby se vyvolalo na náběžné hraně, sestupné hraně nebo u obou. Testy byly provedeny se stojícím autíčkem v naprostém klidu, tedy očekáváme, že přerušení nebudou vyvolána. Při všech nastaveních výstup senzoru kmitá v případě, že není objekt detekován, tedy před senzorem se nachází díra a digitální výstup je 0. V případě opačném, tedy úspěšném zaznamenání objektu, odrazu záření, digitální výstup senzoru je logická jedna, zůstává výstup stále stejný, kmitání jsem nezaznamenal a žádná přerušení nejsou vyvolána - tedy očekávané chování. Proto nám nestačí v každém vyvolaném přerušení inkrementovat počítadlo, ale musíme detekovat, zda se jedná o zákmit nebo ne. Musíme místo děr zaznamenávat „sloupky“, které jsou mezi nimi. Podle frekvence senzoru a maximální rychlosti autíčka můžeme definovat čas, po který musí senzor stále vracet logickou jedničku. Pokud je tento čas překročen, tak je inkrementováno počítadlo. Dalším problémem, který byl zaznamenán s využitím infračerveného senzoru, je zákmit při změně stavu z díry na sloupek a opačně. Původní plán byl tento problém odstranit definicí „timeoutu“, kdy po úspěšném zaznamenání objektu a inkrementace počítadla by byla odfiltrována a ignorována všechna přerušení po daný krátký čas. Tento druhý problém s kmitáním je ale vyřešen implementací řešení prvního problému. Problém s měřením se neprojevoval pouze v klidovém stavu, ale také za jízdy, kdy jsem vytvořil krátký oválný okruh dlouhý cca 3 metry. Provedl jsem měření doby projetí okruhu autíčkem pomocí stopek. Průměrná doba jednoho kola byla cca. 6 sekund. Což by mělo znamenat průměrnou rychlost kolem půl metru za sekundu. Avšak naměřená, spočítaná a bezdrátově přenesená rychlost se pohybovala kolem 6 metrů za sekundu. V grafu na obrázku 25) můžeme vidět naměřené hodnoty ze senzoru. Jak stoupá při rozjezdu, klesá při zastavení a také jak auto zpomaluje v zatáčkách a následně se rozjíždí ze za-

(a) Zákmit - naměřená rychlost, když bylo auto v klidu



(b) Rozměry ozubeného kola

Obrázek 24: Řešení problému se zákmity



Obrázek 25: Graf naměřené rychlosti v závislosti na čase(se zákmity)

táček. Problém je v nepřesných hodnotách. Průměrná rychlost se měla pohybovat zhruba kolem jednoho metru za sekundu.

5.4 Implementace řešení zákmitů

Dalším krokem tedy bylo inkrementovat počítadlo pouze v situaci, kdy senzor kontinuálně zůstává v logické jedničce po dobu určitého času. Tento čas musíme zjistit výpočtem tak, aby správně zaznamenával sloupky i v případě vyšších rychlostí. Měřením jsem zjistil, že průměr ozubeného kola je 40 mm. Z čehož pomocí vzorečku pro obvod kružnice můžeme zjisti obvod kola, který je cca. 125 mm. Dále tloušťka sloupku u obvodu, kde jej zaznamenáváme, je 3 mm (Viz. obrázek 24b). Jako maximální rychlost při aktuálním nastavení jsem si zvolil 2 metry za sekundu. Průměr kola je 65 mm, obvod tedy cca. 204 mm, z čehož lze odvodit, že se za jednu sekundu může kolo otočit až 10 krát. Na jedno otočení kola tedy mám čas 100 milisekund a když budu pro jistotu počítat, že mi stačí zaznamenat 2 mm sloupku, tak:

$$\frac{2}{125} \cdot 100 = 1,6 \tag{2}$$

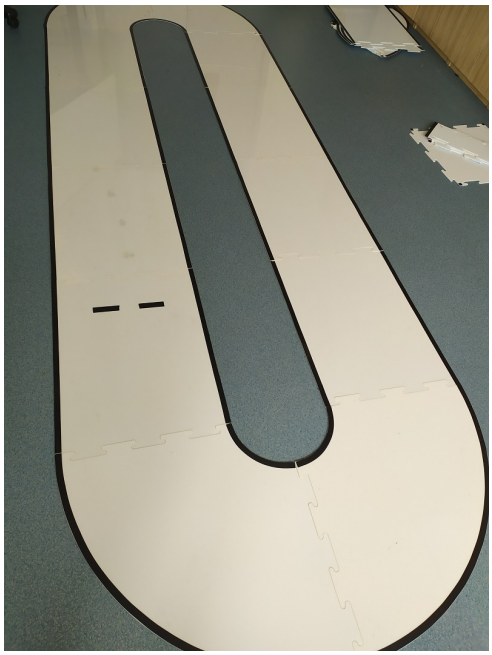
Musím tedy zaznamenávat logickou jedničku v kuse po dobu jedné a třech pětin milisekundy, abych inkrementoval počet zaznamenaných sloupků. Testování zákmitů:

- Auto je v klidu, senzor je v logické jedničce – bez vyvolání přerušení
- Auto je v klidu, senzor je v logické nule – bez vyvolání přerušení

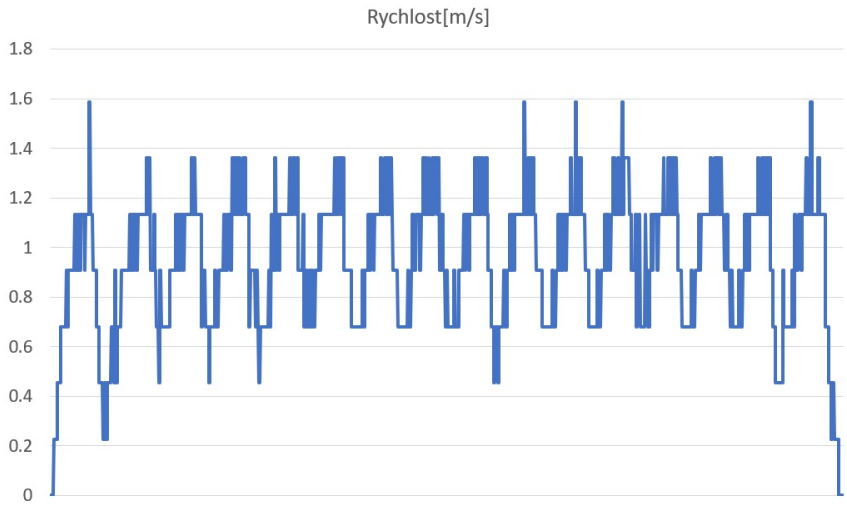
Následuje nové měření. Rozestavěl jsem nový oválný okruh s délkou cca 7 metrů, který autíčko projíždělo v průměru za cca. 7 sekund. Průměrná naměřená rychlost by se tedy měla pohybovat kolem jednoho metru za sekundu. Na obrázku 27 můžeme vidět v grafu naměřené hodnoty rychlosti v závislosti na čase po úpravách kódu a vyřešení zákmitů. Z grafu lze opět vyčíst rozjezd, kolísání rychlosti mezi zatáčkami a rovinkami i samotné zastavení autíčka. Průměrnou rychlost můžeme stanovit kolem 1 m/s, což odpovídá výpočtu z naměřené délky dráhy a stopnutého času. Můžeme tedy tímto experimentem považovat senzor rychlosti a následní výpočet za funkční a správný.

5.5 Úprava dráhy

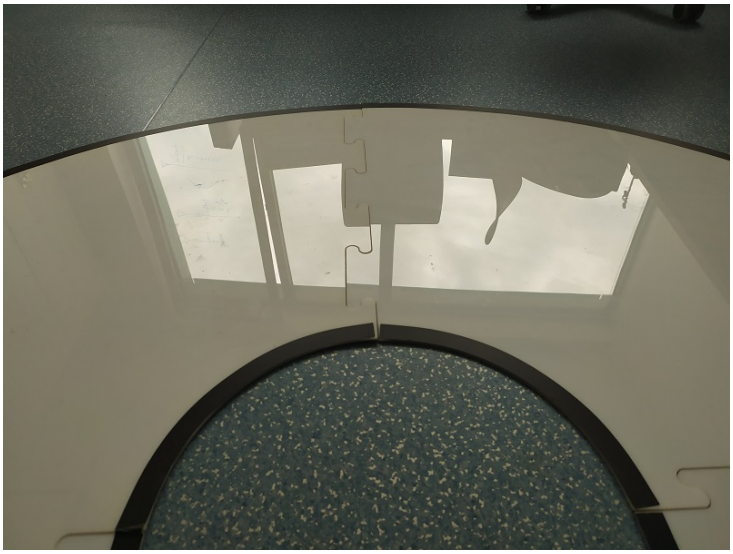
Během testování v laboratoři se ukázalo, že ovládání vozidla, které je z valné většiny závislé na datech z kamery, je náchylné na externí faktory. Hlavním z nich je osvětlení. Pokud například dopadají sluneční paprsky pouze na určitou část dráhy, autíčko má tendenci ztratit v takové oblasti okrajové čáry. Samotná dráha má potom lesklý vysoce reflektivní povrch, takže sluneční paprsky



Obrázek 26: Dráha pokusu se senzorem rychlosti



Obrázek 27: Graf naměřené rychlosti v závislosti na čase

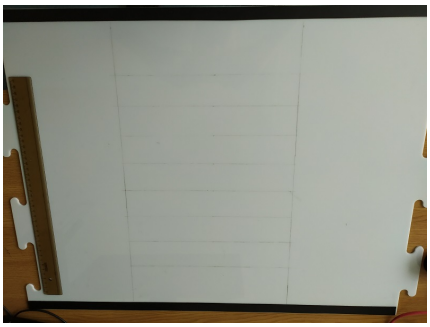


Obrázek 28: Detail odlesku na dráze

nemusí nutně dopadat přímo na dráhu, ale stačí např. pouze odraz od okna nebo odražené světlo při zapnutém osvětlení v laboratoři. Ukázalo se tedy, že abychom zajistili neproměnlivé jízdní vlastnosti, je žádoucí odstranit nebo omezit zdroje světla. V laboratoři lze podmínky vylepšit vypnutým umělým osvětlením a sluneční záření omezit roletami. Dalším méně značným externím faktorem je nepořádek na trati – ať už prach, šmouhy nebo špína, které činí bílý povrch dráhy tmavější a černý světlejší. Odlesk můžeme vidět na obrázku 28 na straně 40, nepořádek na trati na obrázku 26 na straně 39. Utřenou a uklizenou dráhu, ale také různé odlesky na fotografiích 31 na straně 43 a 34 na straně 44. Pro provedení měření a pokusů souvisejících se zadanými disciplínami jsem vyrobil jeden díl tratě se startovní/cílovou linií (Viz. obr. 29b), druhý se vzorem pro začátek zóny s omezenou rychlosti (4 rovnoběžné pruhy)(Viz. obr. 30a) a třetí se vzorem pro konec zóny s omezenou rychlostí (3 rovnoběžné pruhy)(Viz. obr. 30b). K tomu jsem využil stávajících částí dráhy – konkrétně rovinek, na které jsem dle rozměrů z pravidel NXP prvně načrtl tužkou hranice(Viz. obr. 29a) a ty pak přelepil černou páskou.

5.6 Disciplína zóny s omezenou rychlostí

Po nastavení autíčka do režimu určeného pro disciplínu zóny s omezenou rychlostí autíčko přestává komunikovat, zatáčet a regulovat rychlost. Rozhodl jsem se tedy namísto rozpoznávání vzoru pomocí kamery a regionů využít infračervených senzorů umístěných v nárazníku. A tedy využít kód pro běžnou jízdu, ale jelikož se u disciplíny s omezenou rychlostí nezastavuje, tak při zaznamenání čáry senzorem se namísto zastavení alternuje mezi nastavením vyšší a nižší rychlosti. Rychlost byla z globálních proměnných přemístěna do atributů třídy NxpCar a je k nim přistupováno z kódu pro přerušení pomocí metod NxpCar. Kód vytvoření instance třídy NxpCar byl přesunut na začátek funkce main, kvůli volání metod této třídy v obsluhách přerušení. Obsluha infračervených čidel byla změněna tak, aby zareagovala i když je zaznamenána černá

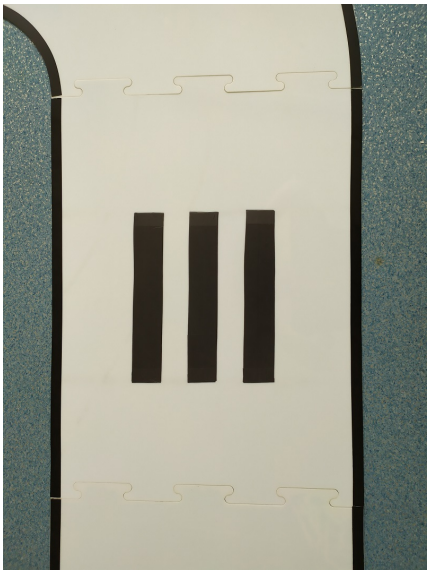


(a) Mezikrok - vyznačení oblastí tužkou

(b) Výsledný díl dráhy s cílovou linií

Obrázek 29: Úprava dílů dráhy 1

(a) Výsledný díl dráhy se vzorem značící začátek zóny s omezenou rychlostí



(b) Výsledný díl dráhy se vzorem značící konec zóny s omezenou rychlostí

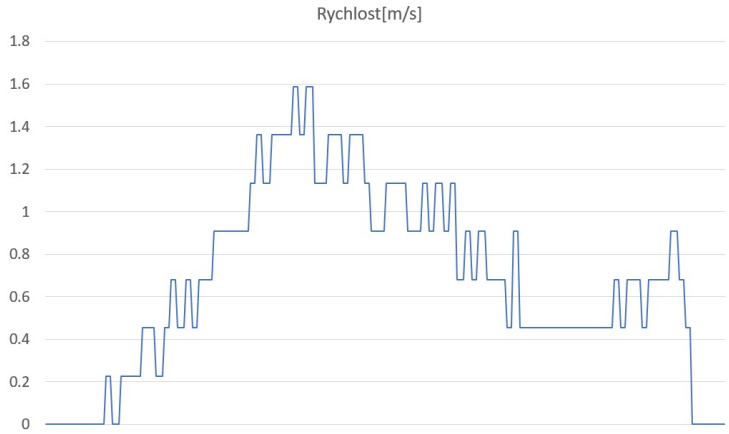
Obrázek 30: Úprava dílů dráhy 2

čára pouze pod jedním z nich, kvůli pruhům, které jsou mnohem užší než cílová linie. Aby se autíčko nepřepínalo rapidně mezi nižší a vyšší rychlostí při průjezdu vzorem, byla implementována proměnná typu boolean kontrolující, zda uplynul čas uložený v konstantě, po který je zakázáno přepínat mezi režimy. Pro samotnou kontrolu rychlosti bylo využito nového senzoru rychlosti. Definujeme si konstantu pro každý režim (u režimu pro disciplínu zóny s omezenou rychlostí budou konstanty dvě – jedna pro situaci, když jsme v zóně zpomalení a druhou, když jsme mimo ni). Tato konstanta určuje maximální povolenou rychlost v metrech za sekundu, ke které bude porovnávána naměřená rychlost ze senzoru. Při zaznamenání vzoru pro začátek nebo konec zóny s omezenou rychlostí se tedy alternuje mezi dvěma hodnotami maximální rychlosti. Po porovnání aktuální naměřené rychlosti s maximální povolenou rychlostí poté inkrementujeme nebo dekrementujeme PWM motoru o 1 krok daný parametrem uloženým v konstantě. V první verzi implementace bylo toto porovnávání a změna PWM v každém cyklu programu, což se ukázalo jako vysoce nepraktické, protože sinusoida rychlosti pak díky nízké frekvenci výpočtu rychlosti měla zbytečně velkou amplitudu. Perioda měření, tedy výpočtu rychlosti byla původně 1 sekunda, poté 100 milisekund, na kterých zůstala. Ukázalo se totiž testováním, že čím kratší periodu nastavíme, tím více znehodnotíme měření, protože se nestihne inkrementovat počítadlo. Obzvlášť u pomalejších rychlostí. Zároveň však tato perioda nemůže být příliš vysoká, abychom ji mohli efektivně využít k řízení auta. Je proto ideální najít zlatý střed, který byl prozatím určen na 100 milisekund. Abych tedy vyhladil zpomalování a zrychlování auta, implementoval jsem porovnávání naměřené rychlosti s maximální povolenou rychlostí a následnou změnu PWM motoru do setteru rychlosti. 10 krát za vteřinu se tedy provádí měření a na základě toho se hned po něm mění rychlost auta. V tomto pokusu bylo autíčko postaveno dle pravidel disciplíny hned za vzor 3 pruhů na dráhu 31. Jak můžeme vidět v grafu na obrázku 32, autíčko po zmáčknutí tlačítka B začalo akcelerovat, jakmile zaznamenalo vzor 4 pruhů, tedy začátku zóny, tak byla maximální rychlost nastavena na 0,5 metrů za sekundu a autíčko začalo zpomalovat. Pak si chvilku snažilo udržovat rychlost kolem 0,5 metrů za sekundu, než bylo zastaveno.

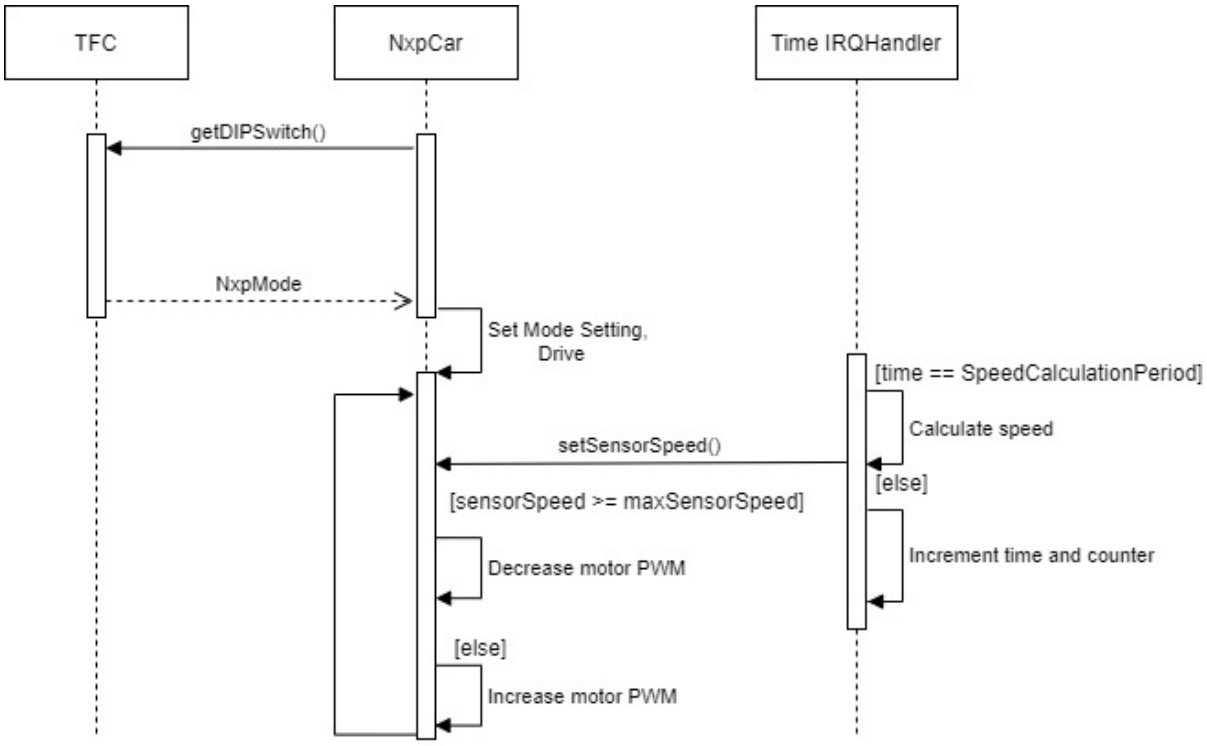
5.7 Disciplína průjezdu osmičkou

Testy na drahách tvaru číslice osm bylo již provedeno více, ale na závěr jsem ještě postavil dráhu uvedenou v pravidlech soutěže jako příklad rozložení dráhy pro disciplínu průjezdu osmičkou. Ta je vyfocena na obrázku 34. Nově v tomto režimu autíčko dostalo parametr maximální rychlosti, který byl nastaven na 1 metr za sekundu. Graf naměřených hodnot je na obrázku 35. Na tomto grafu můžeme vidět, jak se změnila sinusoida oproti grafu na obrázku 27 na straně 39. Průměrná rychlost zůstává stejná, ale amplituda rychlosti je menší a součet odchylek je také menší. V soutěži by se pak tato hodnota nastavila vyšší a při neúspěchu pak snižovala, jelikož na ni mají týmy 3 pokusy.

Obrázek 31: Dráha se zónou s omezenou rychlostí

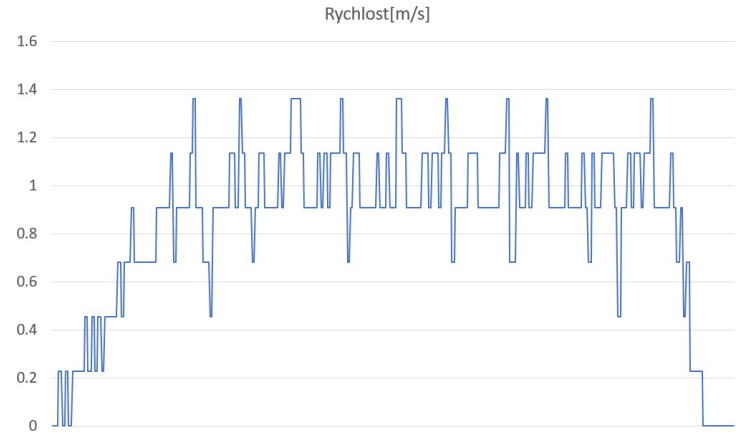


Obrázek 32: Závislost rychlosti na čase v režimu zóny s omezenou rychlostí



Obrázek 33: Sekvenční diagram zobrazující komunikaci mezi objekty a regulaci rychlosti

Obrázek 34: Dráha pro disciplínu osmičky (viz. pravidla)



Obrázek 35: Graf závislosti rychlosti na čase u dráhy tvaru 8

6 Závěr

Autonomní řízení bude možné využít v každém segmentu dopravy, automobilovém, železničním nebo leteckém. Před zavedením takové technologie na špičkové úrovní do provozu je ale nutná maximální jistota, že zajistí maximální komfort a spolehlivost, zároveň ale musí být bez bezpečnostních rizik. Tato práce se zaměřila na svět autonomní technologie z pohledu využití autíčka, které mělo kvalitně reprezentovat na soutěži NXP CUP. Cílem této práce byla tvorba software pro ovládání robotického auta, včetně detekce dráhy a okrajových čar, připojení a využití senzoru pro měření rychlosti, což se podařilo, ale bohužel nebylo možné předvést v praxi na soutěži. Dynamické přizpůsobení světelným podmínkám dráhy je zajímavé téma, na které mi bohužel nezbyl čas. Jak jistě všichni víte, v roce odevzdávání této bakalářské práce, tedy roce 2020, byl vyhlášen nouzový stav skoro po celém světe, nevyjímaje Českou republiku. Svět byl a stále je zasažen pandemií koronaviru *SARS-CoV-2*, souvisejícím s vysoce infekčním onemocněním horních cest dýchacích *Covid-19*. V důsledku této situace byly a stále jsou omezeny přístupy do škol. Také ročník soutěže NXP CUP 2019/2020 byl prvotně přesunut do druhé poloviny roku 2020 a následně zrušen.

Literatura

1. ZVONEK, Richard. *Mechanizmy řízení robotického auta NXP (FREESCALE)* [online]. Ostrava, 2019-04-30 [cit. 2020-05-02]. Dostupné z: http://hdl.handle.net/10084/136273. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava.
2. NXP SEMICONDUCTORS N.V. *NXP CUP EMEA 2018/2019 - About NXP Cup* [online]. 2018 [cit. 2020-05-05]. Dostupné z: https://community.nxp.com/servlet/JiveServlet/downloadBody/340794-102-3-285762/European%20NXP%20Cup%202018_19.pdf.
3. NXP SEMICONDUCTORS N.V. *NXP Company corporate overview* [online]. 2020-04-01 [cit. 2020-05-05]. Dostupné z: https://www.nxp.com/docs/en/supporting-information/NXP-CORPORATE-OVERVIEW.pdf.
4. NXP SEMICONDUCTORS N.V. *The NXP Cup Official Rules* [online]. 2019-09-28. Version 1.3 [cit. 2020-05-05]. Dostupné z: https://community.nxp.com/servlet/JiveServlet/downloadBody/335269-102-6-293715/NXP%20Cup%202019_20_rules%20V1_3%2027Sep19.pdf.
5. NXP SEMICONDUCTORS N.V. *The NXP Cup EMEA Community* [online]. 2019-09-27 [cit. 2020-05-05]. Dostupné z: https://community.nxp.com/groups/tfc-emea.
6. NXP SEMICONDUCTORS N.V. *NXP CUP Track Configurations* [online] [cit. 2020-05-05]. Dostupné z: https://community.nxp.com/servlet/JiveServlet/download/101612-7-431006/NXP+CUP_track_configurations_2018_19.pdf.
7. NXP SEMICONDUCTORS N.V. *FRDM-K66F* [online] [cit. 2020-05-12]. Dostupné z: https://www.nxp.com/design/development-boards/freedom-development-boards/mcu-boards/freedom-development-platform-for-kinetis-k66-k65-and-k26-mcus:FRDM-K66F.
8. NXP SEMICONDUCTORS N.V. *The NXP Cup Official Rules* [online]. 2016 [cit. 2020-05-12]. Dostupné z: https://www.nxp.com/docs/en/user-guide/FRDMK66FUG.pdf.
9. *Ilustrační fotografie ozubuného kola* [online] [cit. 2020-05-12]. Dostupné z: https://www.astramodel.cz/images/arrma/800x600/ARA310920.jpg?v=nITv.