

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

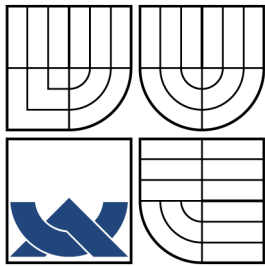
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTEM SLEDOVÁNÍ TRASY PRO AUTONOMNÍ VOZIDLO

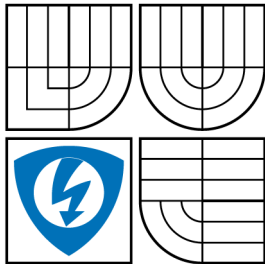
BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETER MOLČÁNY



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTÉM SLEDOVÁNÍ TRASY PRO AUTONOMNÍ VOZIDLO PATH TRACKING SYSTEM FOR AUTONOMOUS CAR MODEL

BAKALÁRSKA PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETER MOLČÁNY

VEDÚCI PRÁCE
SUPERVISOR

Ing. PETR PETYOVSKÝ

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Peter Molčány

ID: 134563

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

System sledování trasy pro autonomní vozidlo

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je sestavit model autonomního vozidla určeného k pohybu po trase určené značkami.

1. Seznamte se s hardwarovými možnostmi modelu vozidla a proveďte základní rozbor úlohy řízení a sledování trasy.
2. Navrhněte vhodné kamerové řešení pro detekci trasy. Zvolte typ osvětlení a umístění kamery.
3. Navrhněte obousměrnou komunikaci mezi vozidlem a PC. Vytvořte aplikaci, umožňující ovládat pohyb vozidla a získávat data ze senzorů v reálném čase.
4. Navrhněte a sestavte algoritmus detekce trasy pohybu vozidla na základě značek na vozovce.
5. Srovnajte dosažené výsledky pro jednotlivé varianty umístění senzorů a parametry detekčního algoritmu.
6. Realizujte řízení natočení předních kol vozidla na základě údajů z kamerové detekce polohy značek na vozovce.
7. Navrhněte a realizujte vhodné řízení rychlosti pohybu vozidla na základě natočení předních kol a údajů z kamerového senzoru.
8. Zhodnoťte dosažené výsledky. Navrhněte další možná vylepšení řízení modelu vozidla.

DOPORUČENÁ LITERATURA:

[1] Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision, 3rd Edition, Thomson 2007, ISBN 049508252X

Termín zadání: 11.2.2013

Termín odevzdání: 27.5.2013

Vedoucí práce: Ing. Petr Petyovský

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

ABSTRAKT

Cieľom tejto bakalárskej práce je navrhnúť a zostrojiť systém sledovania trasy autonóm-
nym vozidlom.

V prvej časti práce sa zaoberáme hardvérovým návrhom a konštrukciou vozidla. Vy-
beráme vhodný typ prísvetlenia scény. Vyberáme vhodný komunikačný prostriedok pre
komunikáciu s PC.

Druhá časť sa zaoberá ovládacím softvérom, ktorý je dostupný pre PC aj pre platformu
Android a tiež softvérom určeným pre vozidlo. Navrhujeme algoritmy detekcie vodiacej
značky a algoritmy pre riadenie vozidla. Meriame rýchlosť vozidla a dostupné informácie
používame pri riadení vozidla.

V záverečnej časti vyhodnotíme dosiahnuté výsledky.

KLÚČOVÉ SLOVÁ

sledovanie trasy, autonómne vozidlo, detekcia čiary, bezdrôtové ovládanie

ABSTRACT

The aim of this thesis is designing and constructing path tracking vehicle. Thesis is
divided into two major parts. In the first, we discuss hardware possibilities of the model.
Additional lighting system is needed for better performance in low-lighting conditions.
Communication module is selected according to needs of this task.

The second part is software design. Control software LineCruiser Monitor for PC was
created and LineCruiser for Android was developed for the needs of efficient parameter
reading and writing. Different path detecting algorithms were tested. Information about
the speed of the car was used for speed regulation.

In the last chapter we evaluate the results.

KEYWORDS

path tracking, autonomous vehicle, wireless control

MOLČÁNY, Peter *Systém sledování trasy pro autonomní vozidlo*: bakalárska práca. Brno:
Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,
Ústav automatizace a měřicí techniky, 2013. 60 s. Vedúci práce bol Ing. Petr Petyovský

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Systém sledování trasy pro autonomní vozidlo“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

(podpis autora)

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Petrovi Petyovskému za odborné vedenie, konzultácie a návrhy k vylepšeniu práce.

Brno

.....

(podpis autora)

OBSAH

1	Úvod	10
2	Požiadavky súťaže Freescale Cup	12
2.1	Popis trate	12
3	Hardvérová časť	15
3.1	Hardvérový popis modelu	15
3.2	Riadenie motorov	16
3.2.1	Servo motor Futaba S3010	16
3.2.2	Riadenie jednosmerných motorov	17
3.3	Meranie otáčok kolies	18
3.3.1	Magnetické snímanie	19
3.3.2	Optické snímanie	20
3.4	Mikroprocesor K40	22
3.5	Snímanie obrazu	23
3.5.1	Riadková kamera s TSL-1401R	23
3.5.2	Požiadavky na prísvetlenie scény	23
3.5.3	Porovnanie LED	24
3.5.4	Vyhodnotenie vhodnosti osvetlenia	26
3.6	Bezdrôtová komunikácia	27
4	Softvérová časť	29
4.1	Komunikácia medzi vozidlom a PC	29
4.1.1	USB debugger	29
4.1.2	Bezdrôtová komunikácia	29
4.1.3	Požiadavky na komunikačný protokol	30
4.1.4	Popis komunikačného protokolu	30
4.1.5	Užívateľské rozhranie ovládacieho programu	31
4.1.6	Popis interného usporiadania programu LineCruiserMonitor	33
4.1.7	Aplikácia pre smartfón	33
4.2	Algoritmy detekcie čiary	35
4.2.1	Minimálna suma na intervale	36
4.2.2	Hranový detektor	36
4.2.3	Zhrnutie detekcie vodiacej čiary a cieľa	40
4.2.4	Porovnanie výsledkov rôzneho umiestnenia senzorov	41
4.2.5	Použitie dvoch senzorov vodiacej čiary	42
4.3	Softvér pre vozidlo	44

4.3.1	Popis potrebných periférií	44
4.3.2	Inicializácia	44
4.3.3	Získavanie obrazu z kamery	44
4.3.4	Meranie otáčok kolies	45
4.4	Riadenie vozidla	47
4.4.1	Natočenie predných kolies	47
4.4.2	Určenie strednej polohy servomotora	47
4.4.3	Natáčanie kolies na základe polohy čiary	47
4.4.4	Regulácia rýchlosti vozidla	47
4.4.5	Jednoduché nastavovanie rýchlosti	48
4.4.6	Pokročilé techniky nastavovania rýchlosti	48
4.4.7	Aktívne brzdenie	49
4.4.8	Elektronický diferenciál	49
5	Vyhodnotenie dosiahnutých výsledkov	51
5.1	Osvetlenie scény	51
5.2	Detekcia čiary	51
5.3	Komunikácia s vozidlom	51
5.4	Meranie rýchlosti	52
5.5	Riadenie	52
5.6	Jazda	52
6	Záver	53
	Literatúra	55
	Zoznam symbolov, veličín a skratiek	56
	Zoznam príloh	57
A	Fotografie trate a vozidla	58
B	CD s programom LineCruiserMonitor pre PC, LineCruiser pre platformu Android, zdrojovým kódom k vozidlu, fotografie vozidla	60

ZOZNAM OBRÁZKOV

2.1	Diel trate štart/cieľ [3]	12
2.2	Diel trate zákruta [3]	13
2.3	Diel trate križovatka [3]	13
2.4	Diel trate vlnovka [3]	14
2.5	Diel trate kopec [3]	14
3.1	Diagram modulov vozidla	15
3.2	Popis jednotlivých častí vozidla	16
3.3	Spojenie predných kolies so servomotorom	17
3.4	Štandardné prepojenie radiča s mikroprocesorom[4]	18
3.5	Zapojenie hallovej sondy TLE4905[6]	20
3.6	Potlač na reflexný kotúč	21
3.7	Pohľad na nainštalovaný kotúč v kombinácii so snímačom CNY70	21
3.8	Relatívna spektrálna charakteristika snímača[5]	24
3.9	Porovnanie intenzity rôznych LED v závislosti od citlivosti kamery	25
3.10	Porovnanie rozloženia pinov Bluetooth modulov s rozdielnym firmvérom	28
3.11	SPI programátor Bluetooth modulov[7]	28
4.1	Bezdrôtové spojenie vozidla s PC	30
4.2	Hlavné okno programu LineCruiserMonitor	32
4.3	Schéma ručného ovládania pomocou smartfónu	34
4.4	Aplikácia LineCruiser: a) základné nastavenie vozidla a zobrazenie potrebných dát, b) graf a šedotónový obraz z kamery prenášaný počas jazdy vozidla	35
4.5	Vodiaca čiara so značkou cieľa	36
4.6	Detekcia vodiacej čiary pomocou hranového detektoru	37
4.7	Detekcia cieľa pomocou hranového detektoru	38
4.8	Detekcia čiary pomocou šablóny	39
4.9	Model koncovej značky-šablóna	39
4.10	Detekcia cieľa pomocou šablóny	40
4.11	Chyba optiky po stranách obrazu	41
4.12	Možnosti nastavovania kamery	42
4.13	Porovnanie použitia jednej kamery a dvoch kamier	43
4.14	Časové priebehy kamery TSL1401 [5]	45
A.1	Vozidlo na trati	58
A.2	Celkový pohľad na trať počas európskeho finále	58
A.3	Zariadenie na meranie času a diváci na európskom finále Freescale Cup 2013	59

ZOZNAM TABULIEK

3.1	Pravdivostná tabuľka MC33931[4]	19
3.2	Parametre LED 515MW7C	25
3.3	Parametre LED 530PR2C	26
3.4	Parametre LED 510E850C	26
3.5	Porovnanie hodnôt integrálov z jednotlivých priebehov	27
3.6	Parametre Luxeon LED 1W	27

1 ÚVOD

Často využívaným automaticky riadeným mobilným robotom v priemysle je AGV (automated guided vehicle). Ide predovšetkým o aplikácie, v ktorých roboty rozvážajú súčiastky, prípadne dopĺňajú tovar v skladoch. Ich trať je vyznačená buď opticky, alebo iným spôsobom, napríklad vodičom v podlahe. Poznáme rôzne typy AGV ako napríklad paletové dopravníky, ťahače, dopravníky súčiastok a iné. V automobilovom priemysle sa používajú predovšetkým dopravníky súčiastok a materiálov, ktoré obsluhujú montážne linky. Tieto dopravníky sa používajú hlavne na aplikácie, kde je potrebné dopravovať pravidelne náklady na väčšiu vzdialenosť, procesy pri ktorých je dôležité sledovanie materiálov a tie, kde je kľúčové včasné dodanie a oneskorenie spôsobuje neefektívnosť.

Snímacím prvkom vodiacej čiary, v prípade použitia optickej značky, môžu byť jednoduché elektronické súčiastky ako fototranzistory, prípadne náročnejšie ako napríklad kamery. V prípade použitia malého počtu snímačov, môže dochádzať k nedostatočnému identifikovaniu polohy. Pre získanie dostatočného množstva informácií sme sa rozhodli vybrať kameru. Použitím kamery môžeme navrhnúť rôzne algoritmy na detekciu značiek na vozovke. Zvyšovaním vzorkovacej frekvencie snímania kamery sa znižuje expozícia, a preto je potrebné myslieť na prípadné dodatočné osvetlenie scény.

Pre ladenie algoritmov a nastavovanie parametrov je potrebné s mobilnou jednotkou komunikovať. Najjednoduchším spôsobom komunikácie je komunikácia po drôte, ktorá dosahuje najvyššie rýchlosti aj spoľahlivosť. Nevýhodou však zostáva nutnosť priameho spojenia s nadriadeným systémom. Pri bezdrôtových moduloch je problém s nízkymi prenosovými rýchlosťami. Bezdrôtové moduly, ktoré dosahujú veľké rýchlosti, sú cenovo náročnejšie, a preto našou úlohou bolo zvoliť cenovo dostupné riešenie s dostatočnou prenosovou rýchlosťou pre obojsmernú komunikáciu. Po nadviazaní spojenia sa uplatní komunikačný protokol, ktorý je potrebné navrhnúť pre efektívne riadenie a získavanie dát z vozidla. Jednoduchosť pre užívateľa zabezpečí zrozumiteľný ovládací program, kde sú umiestnené všetky potrebné komponenty pre zobrazovanie dát z vozidla a jeho ovládanie. Pre ovládanie vozidla počas jazdy je vhodná aplikácia pre smartfóny, ktoré sú v dnešnej dobe ľahko dostupné, pričom významné zastúpenie má platforma Android.

Pre riadenie vozidla sú dôležité jednak informácie z kamerového senzoru a tiež informácie o aktuálnom stave vozidla. Najdôležitejšia je aktuálna rýchlosť a natočenie predných kolies vozidla. Zatáčanie predných kolies môže byť realizované krokovým motorom, šnekovým prevodom, servomotorom prípadne iným spôsobom. Z týchto riešení je najvhodnejší servomotor, ktorý má spätnú väzbu a priamo sníma polohu ramena. Výhodou je tiež kompaktné prevedenie a vďaka prevodovke aj pomerne

veľký krútiaci moment. Pohon vozidla môže byť realizovaný rôznymi spôsobmi, od krokových motorov, po jednosmerné s prevodovkou. Krokové motory majú výhodu v tom, že za určitých podmienok nie je potrebná spätná väzba z odometrie. Jednosmerné s prevodovkou vyžadujú spätnú väzbu, no zvyčajne dosahujú vyššie otáčky. Jednoduchá spätná väzba môže byť magnetická, alebo optická, zabezpečená značkou na motore, prípadne na kolese. Pri použití optického snímania môžeme získať vyššie rozlíšenie. Riadenie rýchlosti motorov môže byť na konštantnú hodnotu, prípadne sa táto konštanta môže meniť v závislosti od režimu, v ktorom sa vozidlo nachádza.

Pri prejazde zákrutou je potrebné spomaliť, aby vozidlo nevybočilo z trate. Na rovnom úseku je výhodné pridať, a tak sa zníži doba prejazdu vyznačenou traťou. Pri zvyšovaní rýchlosti na rovine, je dôležitá predikcia, ktorá včas upozorní vozidlo na blížiacu sa zákrutu. V tomto prípade je výhodné použiť aktívne brzdenie, kedy namiesto odstavenia motorov dôjde k ich prepnutiu do reverzného režimu. Aktívne brzdenie nesmie byť veľmi agresívne, aby sa vozidlo nedostalo do šmyku. Zlepšenie jazdných vlastností môže ešte zabezpečiť elektronický diferenciál, pri ktorom je na vnútorný a vonkajší motor pri prejazde zákrutou aplikovaný rozdielny výkon. Podobne nesmie byť tento rozdiel príliš veľký, lebo by mohlo dôjsť k šmyku vozidla.

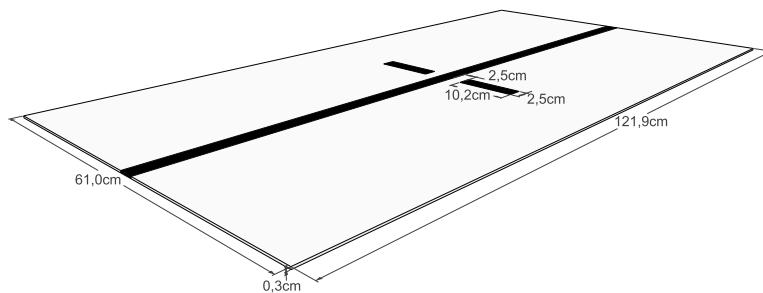
Vozidlo sme koncipovali tak, aby sme sa mohli zúčastniť európskeho finále súťaže Freescale Cup 2013, ktorá sa konala 26.3 - 27.3.2013 v Paríži. Všetky stanovené požiadavky na vozidlo, ako aj trať, budú bližšie popísané a zohľadnili sme ich aj do tejto práce.

2 POŽIADAVKY SÚŤAŽE FREESCALE CUP

Podľa pravidiel súťaže Freescale Cup, bolo potrebné dodržať nasledovné požiadavky. Zostrojiť auto, ktoré je schopné nasledovať čiernu čiaru na bielom podklade z daných dielov a vlastných komponentov. Mikroprocesor môže byť použitý len jeden a musí to byť Kinetis K40. Batéria musí byť NiMH 7,2V 2400 mAh. Konštrukcia celého vozidla nesmie presiahnuť rozmery 250 mm šírka x 400 mm dĺžka. Nie je dovolené zvyšovať napätie na motoroch DC-DC meničom. Celková kapacita všetkých použitých kondenzátorov nesmie presiahnuť 2000 μ F.

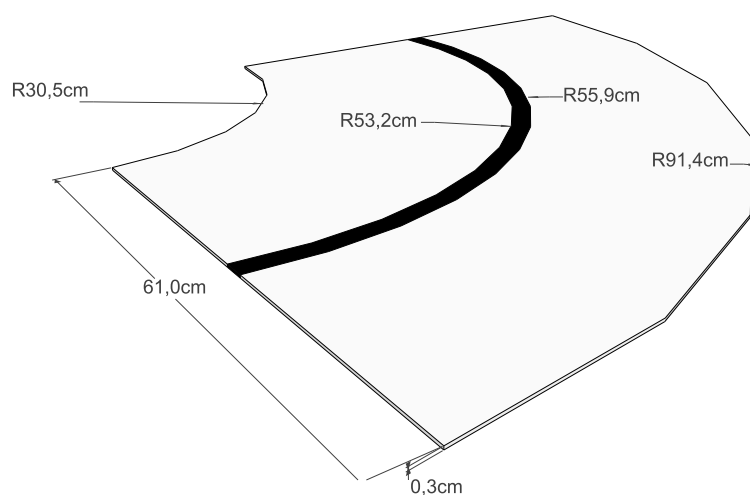
2.1 Popis trate

Trat' je určená súvislou čiernou čiarou šírky 25 mm na bielom podklade. Minimálny polomer zákruty je 500 mm. Vodiaca čiara sa môže križovať pod uhlom 90°. Maximálne stúpanie a klesanie na trati je 15°. Trat' sa skladá z nasledovných dielcov: rovina, štart/cieľ obr. 2.1, zákruta obr. 2.2, križovatka obr. 2.3, vlnovka obr. 2.4 a kopec obr. 2.5, ktoré môžu byť ľubovoľne rozmiestnené. Trat' sa nikdy nerozdvojuje. Na trati sa tiež môže nachádzať tunel, a preto je potrebné auto vybaviť dodatočným osvetlením. Vozidlo nemusí presne kopírovať vodiacu čiaru, ale žiadnym kolesom nesmie vyjsť mimo trate určenej dielcami. Vozidlo musí po prejení cieľa automaticky zastaviť do vzdialenosti 3 m.



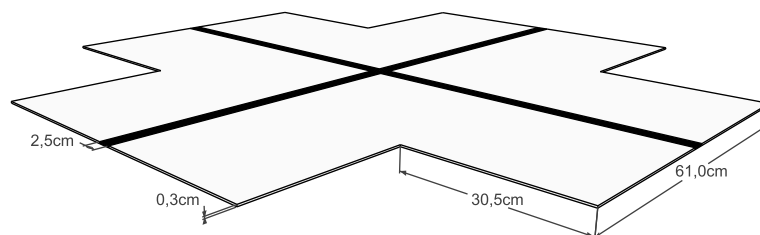
Obr. 2.1: Diel trate štart/cieľ [3]

Štart trate je totožný s cieľom. Čas sa začína počítať keď vozidlo prejde prednými kolesami cez vyznačenú značku a preruší svetelný lúč. Keď preruší tento lúč znovu, čas sa zastaví.



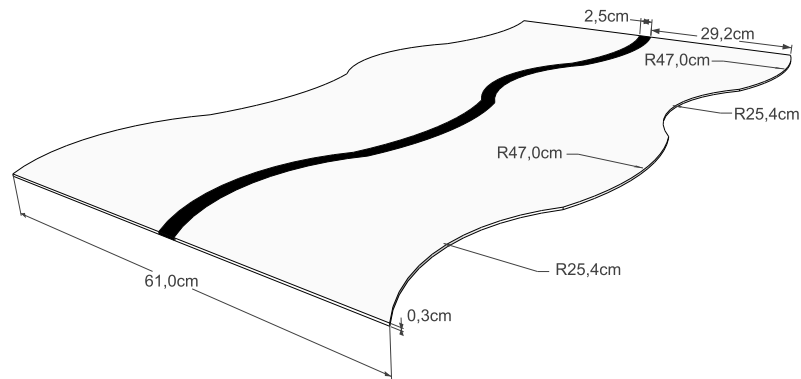
Obr. 2.2: Diel trate zákruta [3]

V zákrute nemôže byť značka cieľu, križovatka ani kopec. Zákruty môžu byť rôzne a nemusia byť presne pod uhlom 90° , ale v našom prípade sme mali k dispozícii iba takéto zákruty.



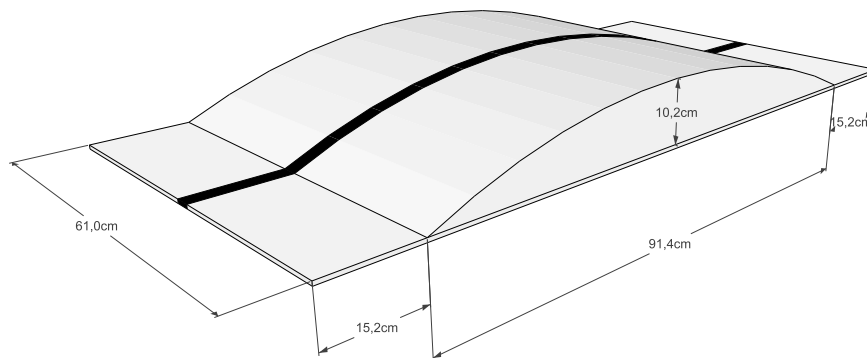
Obr. 2.3: Diel trate križovatka [3]

Križovatky sú vždy pod uhlom 90° , iné tvary nie sú prípustné. Križovatka sa čiastočne podobá na cieľ, no je potrebné vozidlo vybaviť algoritmom, ktoré správne rozlíši, či sa jedná o križovatku alebo cieľ.



Obr. 2.4: Diel trate vlnovka [3]

Vlnovka predstavuje najproblematickejší diel trate. Testuje predovšetkým stabilitu vozidla na trati, pretože pri prejazde hrozí rozkmitanie.



Obr. 2.5: Diel trate kopec [3]

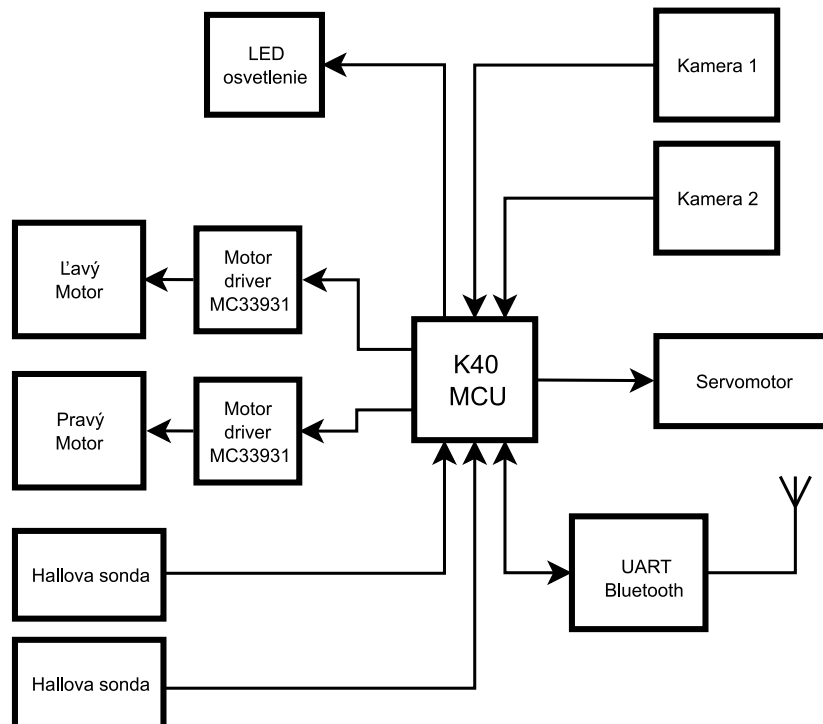
Vozidlo má veľmi nízky podvozok, a preto aj kopec nesmie byť príliš strmý. Pri prejazde cez kopec sme mali problémy a zachytávali sme prednou lištou o trať. Predný dielec sme teda vyradili a pružinu tlmiča zadných kolies sme viac stlačili, čím sme získali vyšší zdvih podvozku.

3 HARDVÉROVÁ ČASŤ

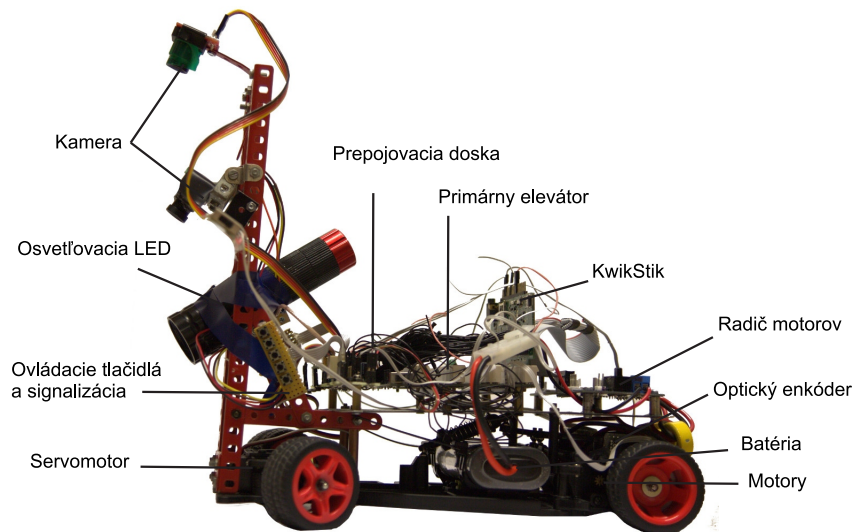
V tejto kapitole budú uvedené požiadavky na zostrojené vozidlo a popísaný návrh jednotlivých modulov vozidla.

3.1 Hardvérový popis modelu

Model sa skladá z plastového podvozku, na ktorom je uchytená zvyšná konštrukcia. Na zadnej časti podvozku sú ukotvené dva jednosmerné motory, ktoré prevodovkou prenášajú krútiaci moment na kolesá. Medzi kolesami je volnobežná oska, ktorá zabezpečuje, aby sa každé koleso mohlo točiť rozdielnou rýchlosťou. V prednej časti podvozku sa nachádza servomotor Futaba S3010, ktorý umožňuje zatáčanie vozidla. Predné kolesá aj zadná náprava sú odpružené. Na podvozku sú prichytené dištančné stĺpiky, na ktorých je umiestnená základná konštrukčná platforma. Na nej sú umiestnené H-mostíky pre ovládanie motorov, základová doska s mikroprocesorom Freescale Kinetis K40 a Bluetooth modul na bezdrôtovú komunikáciu. Na vertikálnej lište sa nachádza kamera s LED osvetlením. Diagram jednotlivých modulov vozidla je na obr. 3.1.



Obr. 3.1: Diagram modulov vozidla



Obr. 3.2: Popis jednotlivých častí vozidla

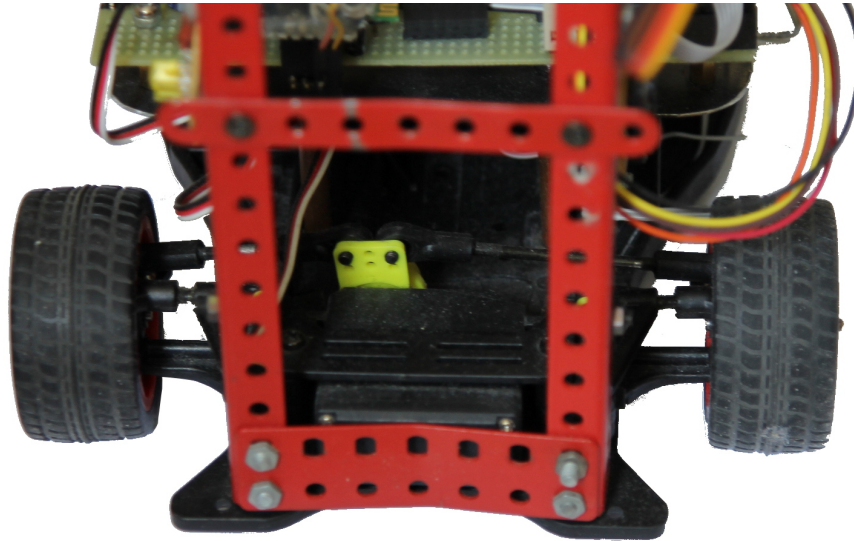
3.2 Riadenie motorov

V našej konštrukcii boli použité tri motory, dva jednosmerné a jeden servomotor. V tejto časti bude popísaný spôsob riadenia oboch typov.

3.2.1 Servo motor Futaba S3010

Na riadenie predných kolies bol použitý servomotor Futaba S3010. Na servomotor sú prichytené ramená, ktoré natáčajú obe predné kolesá obr 3.3.

Servomotor sa môže otáčať v medziach $\pm 90^\circ$. Tento rozsah nie je možné plne využiť, pretože natočenie kolies je limitované konštrukciou. Testovaním sme zistili, aký signál ešte môžeme odoslať do servomotora, bez poškodenia nosného ramena. Táto hodnota sa po čase začala meniť, čo bolo spôsobené preklzovaním krúžku na servomotore. Napájanie servomotora je zabezpečené pomocou 5V stabilizátora, ktorý je umiestnený na doske s H-mostíkom. Riadiaci signál je generovaný mikroprocesorom. Natočenie servomotora určuje strieda signálu PWM, ktorý má frekvenciu 50 Hz, čo zodpovedá času 20 ms. Pre natočenie predných kolies na 0° je potrebné nastaviť signál servomotora na šírku pulzu 1,5 ms. Pre natočenie o 90° voči základnej polohe nastavíme šírku pulzu na 1,25 ms pre natočenie o -90° 1,75 ms.



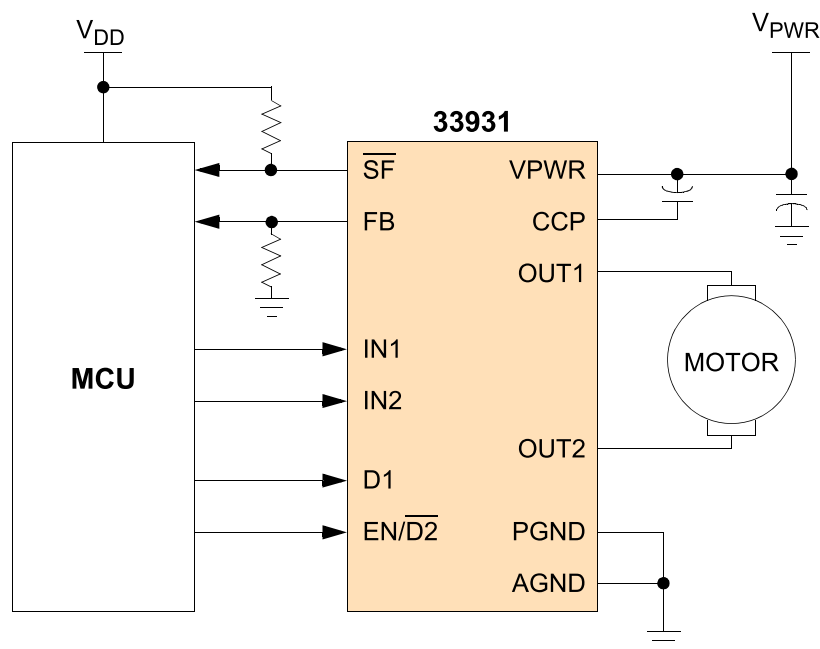
Obr. 3.3: Spojenie predných kolies so servomotorom

3.2.2 Riadenie jednosmerných motorov

V zadnej časti vozidla sa nachádza doska s H-mostíkmi pre každý motor zvlášť. Tiež obsahuje stabilizátory na 5V a konektory pre pripojenie ďalších periférií. Možnosť pripojenia kamery a senzorov k tejto doske sme nevyužili, pretože rušenie od motorov bolo veľmi silné a obraz z kamery skreslený. Preto sme sa rozhodli trasu signálu z kamery a senzorov viesť priamo k základovej doske. Jadrom výkonovej časti je H-mostík Freescale MC33931.

Základné vlastnosti MC33931:

- Napájacie napätie 5,0 až 28V
- Maximálny odpor $R_{DS(ON)} = 235 \text{ m}\Omega$ pri teplote $T_J = 150^\circ\text{C}$ pre každý MOS-FET v H-mostíku
- 3,0 V až 5,0 V TTL/CMOS kompatibilné vstupy
- Nadprúdová ochrana
- Teplotne závislá prúdová poistka
- Režim spánku s odberom prúdu menším ako $50 \mu\text{A}$



Obr. 3.4: Štandardné prepojenie radiča s mikroprocesorom[4]

Ako vyplýva z pravdivostnej tabuľky tab. 3.1, pokiaľ chceme ovládať motor do oboch smerov, musíme meniť signály na pinoch IN1 a IN2. Na doske plošných spojov je však pin IN1 H-mostíku MC33931 priamo pripojený k zemi a tým pádom toto riešenie neumožňuje riadiť motory do oboch smerov. Otáčky motorov nastavujeme striedou signálu PWM.

3.3 Meranie otáčok kolies

Použitie jednosmerné motory nemajú zavedenú spätnú väzbu, a preto sme ju boli nútení z dôvodu lepšieho riadenia rýchlosti zaviesť. Kolesá vozidla majú priemer 50 mm. Pri teoretickej maximálnej rýchlosti vozidla 3 m/s dostávame približne 20 otáčok za sekundu.

Device State	Input Conditions				Status	Outputs	
	EN/D2	D1	IN1	IN2	SF	OUT1	OUT2
Forward	H	L	H	L	H	H	L
Reverse	H	L	L	H	H	L	H
Freewheeling Low	H	L	L	L	H	L	L
Freewheeling High	H	L	H	H	H	H	H
Disable 1 (D1)	H	H	X	X	L	Z	Z
IN1 Disconnected	H	L	Z	X	H	H	X
IN2 Disconnected	H	L	X	Z	H	X	H
D1 Disconnected	H	Z	X	X	L	Z	Z
Under-voltage Lockout	H	X	X	X	L	Z	Z
Over-temperature	H	X	X	X	L	Z	Z
Short-circuit	H	X	X	X	L	Z	Z
Sleep Mode EN/D2	L	X	X	X	H	Z	Z
EN/D2 Disconnected	Z	X	X	X	H	Z	Z

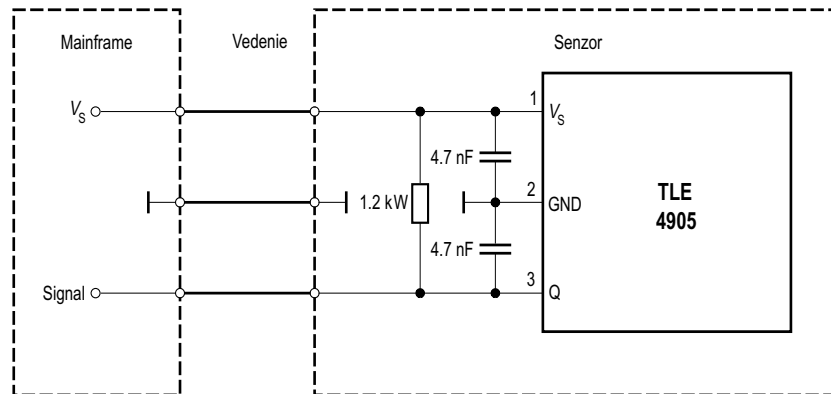
Tab. 3.1: Pravdivostná tabuľka MC33931[4]

3.3.1 Magnetické snímanie

Problematické bolo umiestnenie senzorov, pretože v priestoroch medzi zadnými kolesami je veľmi malý priestor a kolesá majú z vnútornej strany nepravidelný tvar. Zvažovali sme aj umiestnenie snímačov na predné kolesá. Tieto sú však umiestnené na pohyblivom odpruženom ramene a zároveň dochádza k ich natáčaniu. Najjednoduchší spôsob merania bol cez magnetické relé a magnet umiestnený na zadných kolesách. Magnetické relé sme neskôr nahradili hallovou sondou. Zapojenie na obr. 3.5 je konštrukčne jednoduché. Pri kúpe nám dali iný typ, ako sme si objednali, ukázal sa však ako vhodnejší pre túto aplikáciu. Pôvodne sme tam plánovali využiť model, ktorý zmení stav výstupu, pokiaľ na neho pôsobí magnetické pole. Nie je dôležité, či je to južný alebo severný pól magnetu. Pri umiestnení dvoch magnetov na koleso, by sme dostali 2 impulzy teda 4 hrany. Pri približovaní magnetu nábežnú a pri vzdalovaní zostupnú. Pre nás je šírka týchto impulzov nepodstatná, pretože môže závisieť aj od rozdielnej veľkosti a intenzity magnetov. Modely, ktoré nám boli dodané, pracujú ako prepínače. Pri priblížení severného pólu sa dostanú do jedného stavu a zotrývajú v ňom aj po vzdalovaní magnetu. Prepnú sa až po priblížení južného pólu magnetu. Na koleso sme teda upevnili dva magnety. Jeden je natočený južným a druhý severným pólom k snímaču. Pri jednej otáčke dostaneme dve hrany, ktoré sú spracovávané mikroprocesorom. Keďže sa senzory nachádzajú v bezprostrednej blízkosti motorov, na signálnom vodiči sa indukovalo veľké rušenie. Štandardné zapojenie preto odporúča použiť tienené vodiče.

Spracovanie signálu zo senzorov prebieha v mikroprocesore. Na čítač je privedený signál, ktorý nám pri zmene hrany vyvolá prerušenie. Zmeriame čas medzi dvomi hranami a prevrátená hodnota času nám udáva frekvenciu. Pri meraní času si určíme hraničnú hodnotu časovača, po dosiahnutí ktorej sa časovač vynuluje. Týmto

spôsobom môže program rýchlo zistiť, že koleso stojí. Nevýhodou tejto metódy v niektorých aplikáciách môže byť to, že nedokážeme zmerať pomalé otáčanie kolies s otáčkami menšími ako približne 1 s. V našom prípade sa však tieto hodnoty pohybujú v rozmedzí 3-20 otáčok za sekundu.



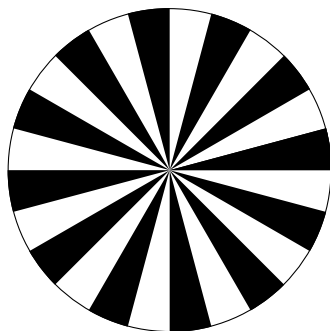
Obr. 3.5: Zapojenie hallovej sondy TLE4905[6]

3.3.2 Optické snímanie

Pre presnejšie meranie otáčok, je vhodné umiestniť na koleso viac značiek, čím získame na otáčku viac hrán. Taktiež rýchlejšie zistíme, že koleso stojí. Táto informácia je dôležitá pre vypnutie motorov. V prípade, že motory bežia na maximálnych otáčkach a náhle sa kolesá zastavia, napríklad pri náraze, začne motormi pretekať veľmi vysoký prúd, ktorý môže viesť až k zničeniu motorov. Preto, pokiaľ nedostávame impulzy zo senzorov, je potrebné vypnúť PWM oboch radičov.

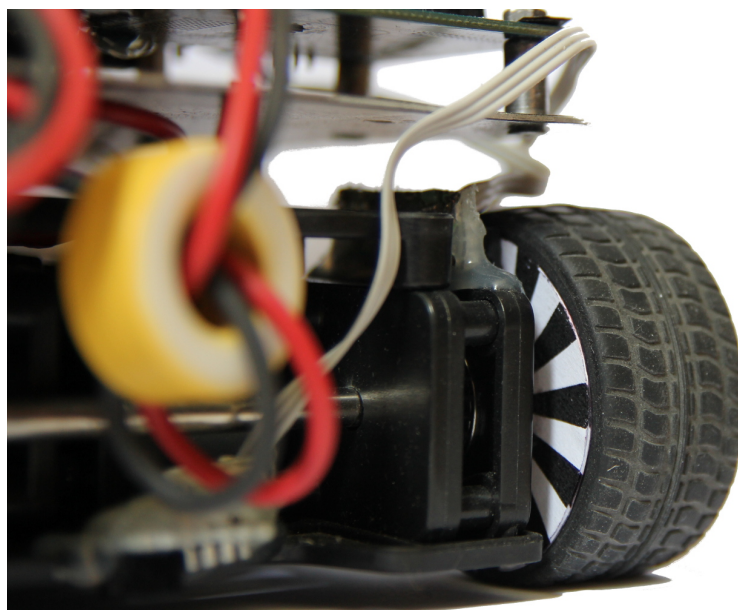
Pre zvýšenie počtu impulzov za otáčku, je vhodné zvoliť inú techniku snímania. V našom prípade je priestor medzi kolesami veľmi malý a nie je možné tam umiestniť precízny inkrementálny enkóder. Ďalšou možnosťou je nalepenie papierového kotúča so značkami po obvode priamo na koleso. Infračervená LED svieti na kotúč a svetlo sa odráža späť. Vhodným reflexným optočlenom, ktorý obsahuje vysielateľ aj prijímač je CNY70. Pracuje na vlnovej dĺžke 950 nm a prúd prechádzajúci vysielacou LED je 50 mA. Výhodou je pomerne malé púzdro, ktoré je možné umiestniť do priestoru pri kolese. Maximálna odporúčaná vzdialenosť optočlenu od reflexného prvku je 5 mm. Skúšali sme kotúče s rozdielnou hustotou značiek, no boli sme limitovaní dvomi požiadavkami. Na jednej strane sme požadovali čo najvyššiu hustotu značiek, aby sme získali čo najvyššie rozlíšenie. Na druhej strane sme potrebovali čo najväčšie reflexné plochy, aby snímač dokázal rozoznať medzi bielou a čiernou farbou na kotúči. Maximálne rozlíšenie, ktoré bol ešte snímač schopný správne detekovať, bolo dvadsať štyri značiek na otáčku. Zvýšiť by sme ho dokázali iba zväčšením kolese, čo

v našom prípade nebolo možné. Nebolo to však ani potrebné, lebo toto rozlíšenie bolo postačujúce.



Obr. 3.6: Potlač na reflexný kotúč

Vysielacia LED vyžaruje svetlo s maximálnou intenzitou pri 950 nm. Toto svetlo sa odráža od kotúča upevneného na kolese, ktorý pozostáva z bielych a čiernych pruhov. Pri dopade svetla na čierny pruh sa väčšina svetla pohltí a nedôjde k odrazu. Na detektore, ktorý tvorí fototranzistor nameriame len zanedbateľný prúd. Avšak po dopade lúča na biely pruh, dôjde k odrazu svetla a detektorom potečie prúd s maximálnou hodnotou 10 mA. Do série k fototranzistoru zapojíme rezistor a meriame napätie na fototranzitore. Môžeme napevno nastaviť komparátor na určitú spínaciu úroveň alebo použiť A/D prevodník procesora a úroveň meniť softvérovo. My sme sa rozhodli pre komparátor, ktorého výstup je pripojený k pinu procesora. Prípadná zmena úrovne sa dá nastaviť trimrom.



Obr. 3.7: Pohľad na nainštalovaný kotúč v kombinácii so snímačom CNY70

3.4 Mikroprocesor K40

Použitý mikroprocesor K40 je od firmy Freescale. Jedná sa o výkonný procesor triedy ARM Cortex-M založený na architektúre ARMv7. Medzi jeho hlavné výsady patria vysoká maximálna frekvencia vnútorných hodín, USB 2.0 v plnej rýchlosti a množstvo analógových, komunikačných, časovacích a riadiacich periférii.

Prehľad základných modulov mikroprocesoru K40

- jadro ARM Cortex-M4
 - 32-bit mikroprocesor s jadrom triedy ARM Cortex-M s DSP inštrukciami, 1.25 DMIPS/MHz, založený na architektúre ARMv7
- Systém
 - manažment napájania s rôznymi módami
 - jednotka ochrany pamäte
 - ostatné riadiace moduly
 - radič DMA s multiplexorom na zvýšenie možných DMA požiadaviek
 - konfigurovateľný Watchdog
- Pamäte
 - programová Flash pamäť
 - 128 kB pamäte RAM
 - SRAM
- Hodiny
 - Rozličné zdroje hodín umožňujúce interné aj externé časovanie
 - Systémový oscilátor na časovanie mikroprocesoru
 - RTC oscilátor ako zdroj hodín pre RTC
- Analógové moduly
 - Vysokorýchlostný A/D prevodník s vnútornou referenciou
 - Komparátor
 - D/A prevodník
- Komunikačné moduly
 - USB OTG radič
 - SPI
 - CAN
 - I²C rozhranie
 - UART

3.5 Snímanie obrazu

Rozhodli sme sa pre snímanie obrazu kamerou, čo prináša viac komplikácii oproti reflexnému snímaniu napríklad senzorom CNY70. Výhodou je viac informácií o scéne, ktoré môžu byť využité rôznymi algoritmi na detekciu vodiacej čiary. Pri použití dvoch kamier je možné zaostriť jednu kameru na vzdialenejší bod, a tak robiť predikciu. Na rovných úsekoch môžeme zrýchliť a pred zákrutou včas spomaliť. Optimalizáciou trasy sa nebudeme zaoberať, pretože je rozpracovaná v samostatnej práci. V tejto časti bude popísaný spôsob získavania obrazu a výber vhodného osvetlenia scény.

3.5.1 Riadková kamera s TSL-1401R

V prednej časti vozidla sa na montážnej lište spolu s osvetlením nachádzajú dve riadkové kamery s čipom TSL-1401R. Priloženú optiku je možné zaostrovať len ručne, a preto je potrebné vždy pred jazdou kameru zaostriť. Jedna z kamier obsahuje na doske operačný zosilňovač, čo výrazne zlepšuje kvalitu výsledného obrazu. Samotný čip tvorí 128 fotodiód s jedným posuvným registrom. Dáta sa vyčítavajú sériovo, a preto je nutné zvoliť vysokú frekvenciu signálu hodín CLK. Zároveň sme však limitovaní expozičiou, ktorá by mala byť čo najdlhšia, aby mal výsledný obraz dostatočný odstup signálu od šumu. Treba nájsť medzi týmito dvoma parametrami vhodný kompromis.

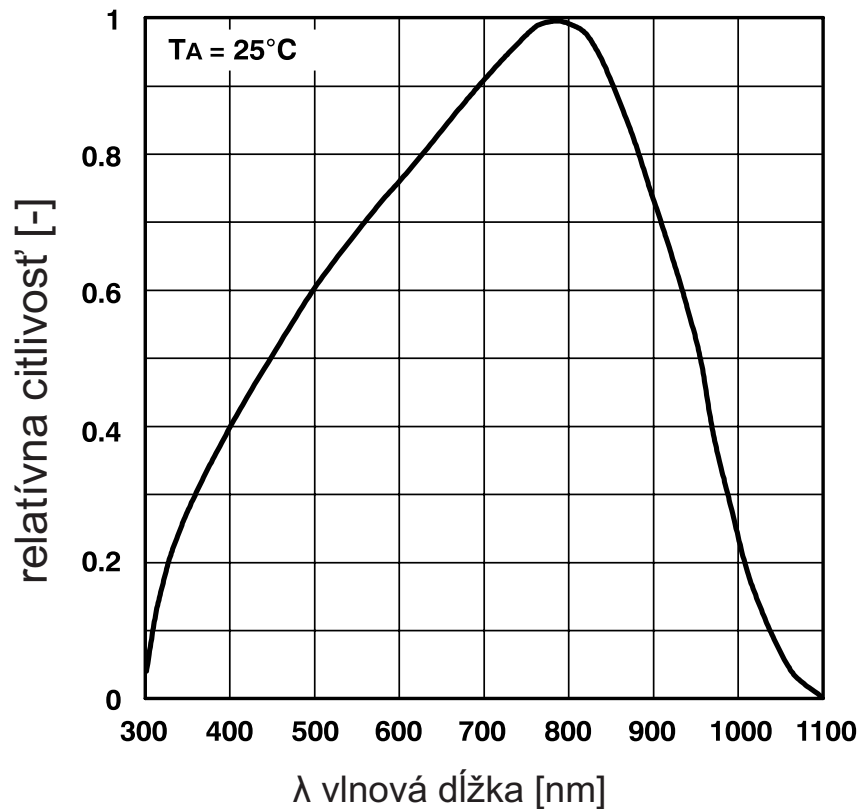
Základné parametre čipu TSL-1401R

- 128x1 snímacích elementov
- Veľkosť pixlu 63,5 μm x 55,5 μm
- Pracovná frekvencia do 8MHz
- Napájanie v rozsahu 3 V až 5 V
- Analógový Rail-to-Rail výstup
- Bez potreby pripojenia externého záťažového rezistora

Priložená optika nerovnomerne sústreďuje svetlo na čip, s týmto obmedzením musíme pri návrhu detekčného algoritmu rátať.

3.5.2 Požiadavky na prisetlenie scény

S cieľom zvýšiť počet získaných snímkov za sekundu v požadovanej kvalite, sme boli nútení pridať vhodné prisetlenie scény. Stanovili sme si požiadavky, ktoré muselo toto osvetlenie spĺňať. Vychádzali sme zo spektrálnej charakteristiky kamery na obr. 3.8.



Obr. 3.8: Relatívna spektrálna charakteristika snímača[5]

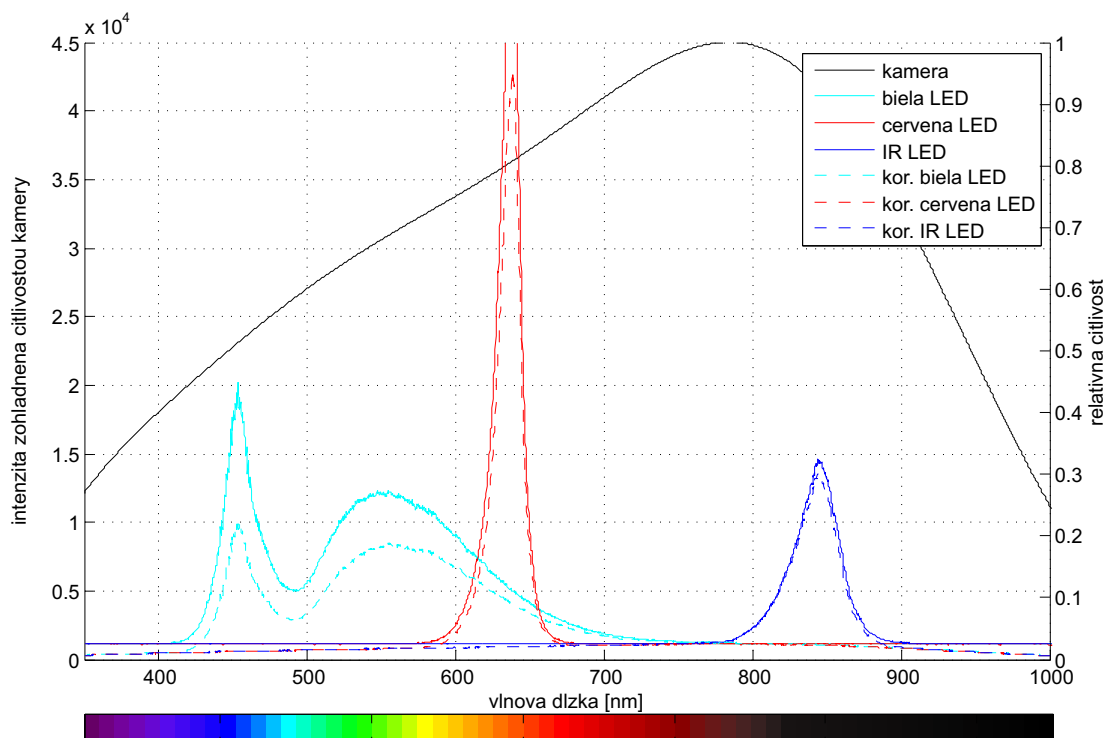
3.5.3 Porovnanie LED

Jednotlivé LED sme merali zo vzdialenosti 200 mm pri prechádzajúcom prúde 20 mA. Zaznamenali sme spektrálne charakteristiky, ktoré sme zohľadnili citlivosťou kamery a vyniesli do grafu na obr. 3.9. Čiernou nepreerušovanou čiarou je vynesena spektrálna charakteristika kamery. Azúrovou, červenou a modrou farbou sú zobrazené merania jednotlivých diód. Tieto priebehy sme utlmili podľa spektrálnej charakteristiky kamery. Pri hodnote 780 nm nedochádzalo k útlmu. Pri zvyšovaní, alebo znižovaní vlnovej dĺžky sa tento útlm zvyšuje.

Zvolené LED boli značky Hebei IT s podobnými parametrami. Dôležitá bola hlavne nominálna svietivosť a malý vyžarovací uhol.

Použité LED

- biela 16000 mcd, 15° 515MW7C tab. 3.2
- červená 5860 mcd, 30° 530PR2C tab. 3.3
- infračervená 1650mW/sr, 13° 510E850C tab. 3.4



Obr. 3.9: Porovnanie intenzity rôznych LED v závislosti od citlivosti kamery

Parameter	Hodnota	Jednotka
Priemer	5	[mm]
Farba Svetla	biela	[-]
Vyžarovací uhol	15	[°]
Farba púzdra	číra	[-]
U_f	3,3	[V]
I_f	20	[mA]
$I_{v(svietivost)}$	16000	[mcd]
Dominantná vlnová dĺžka	460	[nm]

Tab. 3.2: Parametre LED 515MW7C

Parameter	Hodnota	Jednotka
Priemer	5	[mm]
Farba Svetla	červená	[-]
Vyžarovací uhol	30	[°]
Farba púzdra	číra	[-]
U_f	2,6	[V]
I_f	20	[mA]
$I_{v(svietivost)}$	5860	[mcd]
Dominantná vlnová dĺžka	633	[nm]

Tab. 3.3: Parametre LED 530PR2C

Parameter	Hodnota	Jednotka
Priemer	5	[mm]
Farba Svetla	infračervená	[-]
Vyžarovací uhol	13	[°]
Farba púzdra	priehľadná čierna	[-]
U_f	1,55	[V]
I_f	50	[mA]
$I_{e(intezita\ vyzarovania)}$	160	[mW/Sr]
Dominantná vlnová dĺžka	850	[nm]

Tab. 3.4: Parametre LED 510E850C

3.5.4 Vyhodnotenie vhodnosti osvetlenia

Vyhodnotením merania spektra jednotlivých LED, v závislosti od spektrálnej citlivosti kamery, na obr. 3.9 sme určili najvhodnejšie osvetlenie. Z grafu vyplýva, že maximálnu intenzitu dosahuje červená LED. Spočítali sme integrál pre každý typ LED zvlášť. Prehľad spočítaných integrálov je v tab. 3.5. Z tejto tabuľky tiež vyplýva, že kamera je najcitlivejšia na infračervené svetlo, kde je najmenší rozdiel integrálu originálneho priebehu s modifikovaným. Taktiež je vhodné použiť červené svetlo, lebo aj pre túto časť spektra nastáva malý útlm. Kamera je málo citlivá na modrú zložku, kde dochádza k veľkému útlmu signálu, čo sa prejavilo na bielej LED, keďže obsahuje relatívne veľkú časť spektra v tejto oblasti. Napriek tomu, že priebeh je značne utlmený, má biela LED najvyššiu hodnotu integrálu, a preto je najvhodnejším typom osvetlenia.

Farba	Integrál z originálu	Integrál z upravenej charakteristiky
Biela	13,247.10 ⁵	8,7561.10 ⁵
Červená	9,2957.10 ⁵	6,8664.10 ⁵
Infračervená	6,9421.10 ⁵	5,3252.10 ⁵

Tab. 3.5: Porovnanie hodnôt integrálov z jednotlivých priebehov

Vo finálnom prevedení sme použili výkonnejšie LED, keďže sa 5 mm LED prejavili ako nepostačujúce. Vybrali sme LED Luxeon 1W, ktorej parametre sú uvedené v tab. 3.6.

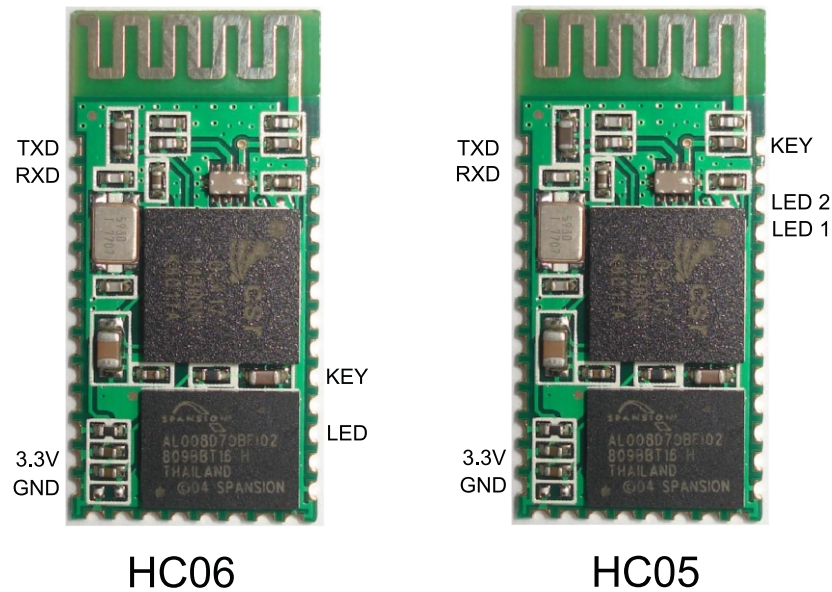
Parameter	Hodnota	Jednotka
Priemer	5	[mm]
Farba Svetla	biela	[-]
Vyžarovací uhol	110	[°]
Farba púzdra	číra	[-]
U_f	3,42	[V]
I_f	350	[mA]
Φ_v	20	[lm]
Teplota	3300	K

Tab. 3.6: Parametre Luxeon LED 1W

Táto LED má veľký vyžarovací uhol, a preto bolo potrebné doplniť optiku s malým vyžarovacím uhlom. Vybrali sme optiku, ktorej vyžarovací uhol bol 30°. Pre dostatočné nasvetlenie vozovky sme umiestnili na lištu 3 LED.

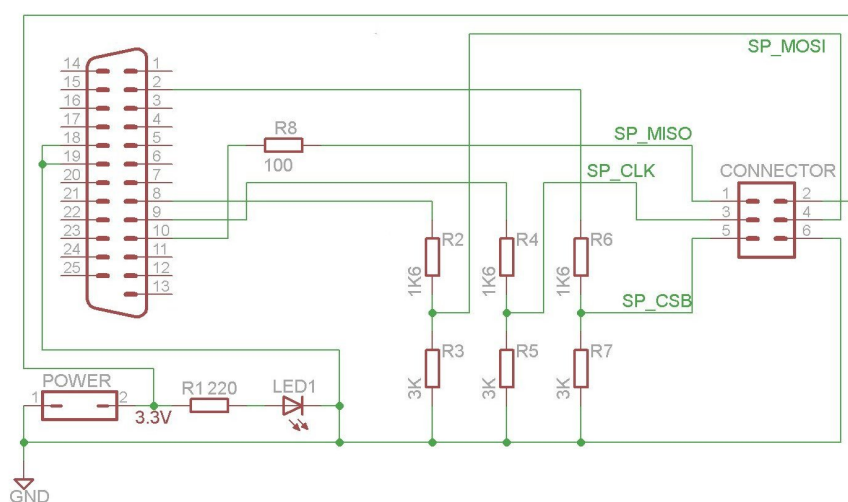
3.6 Bezdrôtová komunikácia

Pre jednoduchšie ladenie algoritmov a pre získavanie aktuálnej informácie o stave vozidla počas jazdy sme zaviedli bezdrôtovú komunikáciu pomocou technológie Bluetooth. Použili sme modul s čipom CSR BC417. Pôvodne sme použili modul s firmvérom HC06, lenže ako sa ukázalo tento firmvér poskytuje len základné nastavenia modulu. Rozhodli sme sa preto ho nahradiť modulom s firmvérom HC05, ktorý umožňuje širokú škálu nastavení a môže pracovať jednak v master, tak aj v slave móde. Novým modulom sa aj mierne zmenilo rozloženie potrebných pinov, a preto sme museli vytvoriť novú dosku s prepojeniami. Názorný rozdiel je uvedený na obr. 3.10.



Obr. 3.10: Porovnanie rozloženia pinov Bluetooth modulov s rozdielnym firmvérom

Tieto Bluetooth moduly je možné programovať pomocou SPI rozhrania. Máme tak k dispozícii modul, ktorý stojí len 5€, ale s nahratím softvéru môže zastávať funkcie špeciálnych Bluetooth modulov v úplne iných cenových hladinách. Schéma SPI programátora je zobrazená na obr. 3.11. Problémom pri programovaní je však podpora paralelného portu, ktorý dokáže emulovať SPI rozhranie. Dnešné moderné PC týmto portom už nedisponujú a aj pri väčšine testovaných počítačov s paralelným portom sa nám nepodarilo komunikáciu nadviazať. Modernou alternatívou je USB SPI adaptér.



Obr. 3.11: SPI programátor Bluetooth modulov[7]

4 SOFTVÉROVÁ ČASŤ

V tejto časti sa budeme zameriavať na softvér pre vozidlo a tiež programom pre vzdialené ovládanie vozidla. Navzájom úzko súvisia, pretože komunikačný protokol musí byť navzájom kompatibilný. Súčasťou softvéru vozidla je algoritmus na vyhodnotenie polohy vodiacej čiary voči vozidlu.

4.1 Komunikácia medzi vozidlom a PC

Pre potreby ladenia algoritmov bolo potrebné zaviesť komunikáciu medzi vozidlom a PC. Porovnali sme rôzne možnosti spojenia a vybrali sme to najvhodnejšie. Navrhli sme komunikačný protokol, ktorý slúži na získavanie dát z vozidla a nastavovanie parametrov jednotlivých algoritmov.

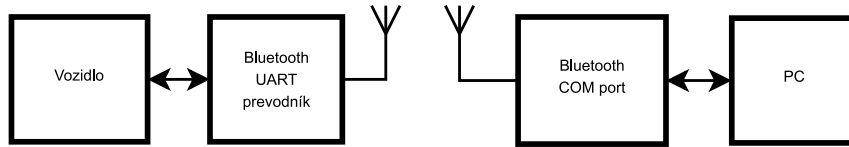
4.1.1 USB debugger

Najjednoduchšou možnosťou ako sledovať prichádzajúce dáta zo snímačov a výstupy algoritmov je pomocou USB J-link debuggeru. Pripája sa priamo na modul Kwix-tik pomocou micro USB rozhrania. V tomto režime je možné program krokovať, prípadne sledovať jednotlivé časti chodu programu. Výhoda spočíva hlavne v tom, že takto vidíme všetky dostupné dáta a môžeme aj zapisovať hodnoty do registrov. Dosahujeme tiež veľké prenosové rýchlosti. Hlavnou nevýhodou je nutnosť priameho spojenia vozidla s PC pomocou káblu, čo je počas jazdy vozidla nerealizovateľná požiadavka. Ďalšou možnosťou je virtuálny sériový port, no ten rovnako potrebuje USB pripojenie.

4.1.2 Bezdrôtová komunikácia

Keďže nebolo v našom prípade možné realizovať priame spojenie, zvolili sme pripojenie pomocou bezdrôtového rozhrania Bluetooth. Emuláciou sériovej linky SPP dokážeme posielat dáta do vozidla a zároveň získať odpoveď. Na hlavnú rozvodnú lištu sme umiestnili Bluetooth modul HC-05. Prenosovú rýchlosť sme chceli zvoliť čo najvyššiu, ale ako sa ukázalo, pri veľmi vysokých rýchlostiach sa dáta strácajú a komunikácia nie je spoľahlivá. Preto sme si vybrali prenosovú rýchlosť (Baud Rate) 57600 odporúčanú výrobcom. Keďže väčšina zariadení od mobilných telefónov až po osobné počítače disponuje zariadením Bluetooth s podporou SPP, nebolo nutné zostrojiť transciever pre PC. Bloková schéma je na obr. 4.1. Potrebné však bolo

vytvoriť aplikáciu, ktorou by sme mohli rýchlo a efektívne získavať hodnoty zo snímačov, zapisovať do registrov a nastavovať režimy jazdy. K tomu bolo treba vytvoriť komunikačný protokol, ktorý by toto zabezpečoval.



Obr. 4.1: Bezdrôtové spojenie vozidla s PC

4.1.3 Požiadavky na komunikačný protokol

Keďže sme mali k dispozícii množstvo informácií, ktoré sme potrebovali zobrazovať v užívateľskej aplikácii, protokol musel byť jednoduchý a nenáročný na dátové toky. Najväčší podiel predstavovali prenášané dáta z dvoch kamier, ktorých 128 bajtov predstavovalo hodnoty 128 pixlov obrazu jednej kamery. Pôvodne sme počítali s priepustnosťou Bluetooth linky 20 zobrazených snímkov za sekundu, ale k tejto hodnote sme sa ani nepriblížili. Nabádalo nás to k zvoleniu riešenia s väčšou prenosovou rýchlosťou pomocou SPI-WIFI mostíkov, ale túto možnosť sme pre náročnosť zavrhlí a pokračovali vo vývoji komunikačného protokolu. Rozhodli sme sa neprenášať celý obraz, ale iba detekovaný stred čiary z oboch kamier, čím sme ušetrili 254 bajtov, teda výsledný dátový rámec predstavoval dĺžku v porovnaní s celým obrazom od oboch kamier menej ako jedno percento.

4.1.4 Popis komunikačného protokolu

Zariadenie typu *Master* predstavuje PC a zariadenie *Slave* je vozidlo. Počítačový program teda odosiela požiadavky, na ktoré odpovedá vozidlo. Požiadavky môžu byť rôzneho typu, môžu sa týkať jednej konkrétnej periférie, prípadne všetkých dostupných informácií. Všetky príkazy majú rovnakú dĺžku 10 znakov a sú v plain texte a je možné ich odosielať z ľubovoľného terminálu. Boli implementované nasledovné príkazy:

- "?L00000000"
Získanie pozície čiary
- "?C10000000"
Získanie obrazu z prvej kamery
- "?C20000000"
Získanie obrazu z druhej kamery

- "?SP0000000"

Získanie informácií o natočení predných kolies
- "?ALLDATA00"

Získanie všetkých dostupných informácií okrem nespracovaného obrazu z kamier
- "!SSPxx0000"

Manuálne nastavenie polohy kolies 00(vľavo), 42(stred), 84(vpravo)
- "!DCSSxxx00"

Manuálne nastavenie rýchlosti v rozsahu 000 až 400(maximum)
- "!ENABCNT00"

Povolenie manuálneho nastavovania rýchlosti a polohy kolies
- "!DISABCNT0"

Zakázanie manuálneho nastavovania rýchlosti a polohy kolies

Vozidlo tieto príkazy spracuje a odpovie na príslušný príkaz. Po odoslaní príkazu "?L00000000" vozidlo odošle pozíciu čiary ako jeden bajt, ktorý môže nadobúdať hodnoty 0-128. Odpoveďou na príkazy "?CX0000000" je obraz z príslušnej kamery, ktorý pozostáva zo 128 bajtov. Každý bajt predstavuje jeden pixel obrazu. Na príkaz "?SP0000000" odpovie vozidlo dvomi bajtmi, ktoré predstavujú hodnotu v časovači ovládajúcim servomotor. Posledným a najčastejšie používaným príkazom je "?ALLDATA00", ktorý vyžiada všetky dáta vozidla vo formáte: "!XABCCDD", kde :

- X

Počet číselných hodnôt ktorý sa odosiela
- A

Pozícia čiary
- BB

Pozícia servomotora
- CC

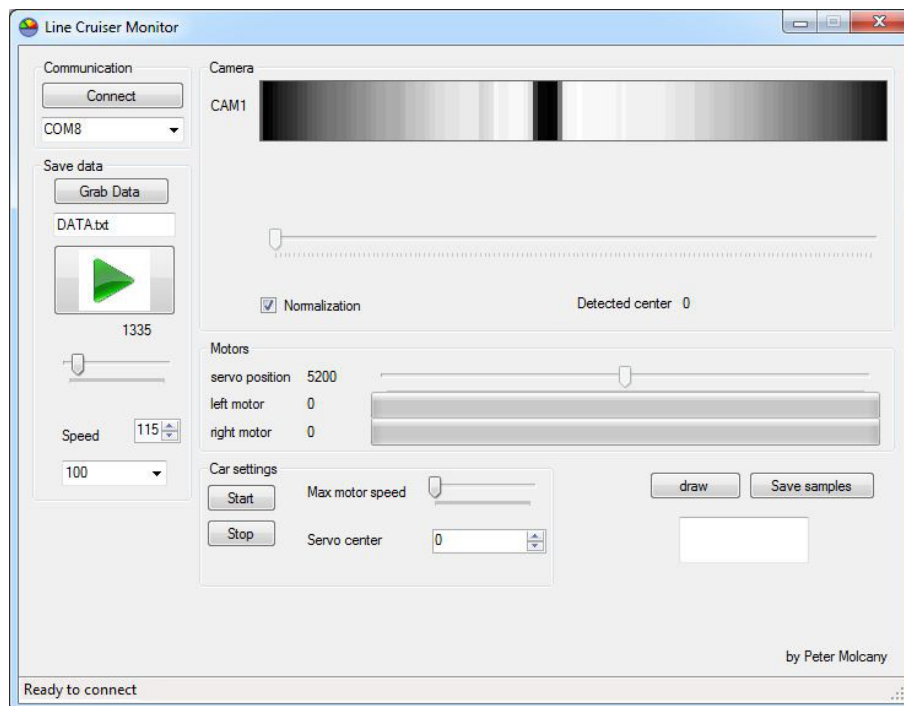
Rýchlosť ľavého motora
- DD

Rýchlosť pravého motora

4.1.5 Uživatelské rozhranie ovládacieho programu

Program LineCruiser Monitor 4.2 bol písaný v C# Microsoft Visual Studio 2010. Po spustení programu sa otvorí hlavné okno, ktoré obsahuje ovládacie a zobrazovacie komponenty. V ľavej hornej časti okna sa nachádza sekcia s názvom *Communication*. Z komponenty typu listbox vyberieme COM port, na ktorom sme sa spárovali s vozidlom. V stavovom riadku, v spodnej časti, sa zobrazí text *Ready to connect*.

Vtedy je možné vytvoriť bezdrôtové spojenie. Pokiaľ sa spojenie úspešne nadviaže, v stavovom riadku sa zobrazí text *Connected*. Od tejto chvíle dostávame dátové rámce z vozidla, ktoré sú spracované. Získané dáta sú zobrazené v hlavnom okne.



Obr. 4.2: Hlavné okno programu LineCruiserMonitor

Základnou informáciou je obraz z kamery. Tvorí ho 128 bodov v rozsahu 0-255. Tieto dáta sú zobrazené v sekcii *Camera*. Obraz z kamery je v programe vyobrazený šedo-tónovým pruhom. Do zobrazenia bola implementovaná funkcia normalizácie obrazu. Tento algoritmus zistí minimálne a maximálne hodnoty v obraze a podľa potreby ekvalizuje histogram. Týmto spôsobom získame väčší kontrast v obraze, ktorý je vhodný pre vizualizáciu hlavne v tmavšom prostredí. Keďže tieto dáta sú prijímané live, je do veľkej miery zaťažená Bluetooth linka. Preto je v programe možné vypnúť žiadanie celého obrazu. Vtedy sa posielajú iba informácie o polohe čiary, serva a otáčkach motorov. Program v pravidelných intervaloch odosiela požiadavku na získanie najnovších dát, ktoré mu po spracovaní vozidlo odošle. Tieto hodnoty sa zobrazia v príslušných komponentoch. V sekcii *Car settings* je možné nastavovať režimy jazdy. Hlavné tlačidlá *Start* a *Stop* slúžia na zásahy do riadenia vozidla. Pri zlyhaní detekčného algoritmu môže tiež obsluha na diaľku deaktivovať vozidlo. V tejto sekcii sa nachádza komponent typu slider, ktorý slúži na nastavenie maximálnej rýchlosti vozidla. Posledným nastavovacím komponentom je *Servo Center*, ktorý slúži na nastavenie strednej polohy predných kolies. Pod sekciov *Communication* sa nachádza sekcia *Save data*, ktorá slúži na ukladanie údajov získaných

z vozidla. Tieto údaje sú potrebné na analýzu detekčných a riadiacich algoritmov. Hodnoty sa pri vyšších rýchlostiach veľmi rýchlo menia a neboli by sme schopní zistiť, prečo vozidlo zlyhalo a zišlo z trate. Z takto uložených dát vieme vyhodnotiť, či stratilo čiaru v zákrute, prípadne či zlyhal detekčný algoritmus. Dáta sú uložené v CSV formáte a je ich možné otvoriť aj v ľubovolnom tabuľkovom procesore. Taktiež je možné urobiť záznam z kamery a následne si ho pozrieť spomalene v rôznych rýchlostiach prehrávania, prípadne si pozerať jednotlivé snímky osobitne. Pri vyšších rýchlostiach však táto možnosť stratila svoje opodstatnenie a tým, že je do veľkej miery zafaržená prenosová linka, získame nedostatočný počet vzoriek za sekundu a dochádza výraznému skresleniu záznamu. V praxi sa to prejaví tak, že na jednej snímke je vozidlo ešte na trati a v ďalšom kroku sa nachádza už mimo trate. Po zavretí programu LineCruiserMonitor sa uzavrie otvorený COM port a uloží záznamový súbor, ktorý obsahuje prijaté dátové rámce z vozidla.

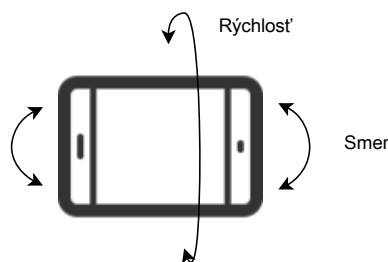
4.1.6 Popis interného usporiadania programu LineCruiser-Monitor

Program LineCruiserMonitor obsahuje dve vlákna, z ktorého jedno zabezpečuje komunikáciu po sériovej linke a druhé celý chod programu a vykresľovanie užívateľského rozhrania. Program pri štarte preskenuje dostupné COM porty systému Windows a zobrazí ich v listboxe. Priradenie portu k Bluetooth SPP zabezpečuje Microsoft Bluetooth Stack, prípadne iný, ktorý výrobca, alebo užívateľ nainštaloval do PC. Užívateľ si vyberie COM port, na ktorom je spárovaný s vozidlom. Po úspešnom spojení sa zasvieti LED na vozidle a zároveň v stavovom riadku sa zobrazí stav Connected. Program obsahuje časovače, ktorými si môžeme pravidelne žiadať dáta z vozidla. Pri kontinuálnom čítaní je dôležité, aby boli spracované všetky požiadavky, a preto sme limitovaní maximálnou frekvenciou dotazovania. V počiatočných štádiách vývoja programu bola implementovaná normalizácia obrazu. Po prenesení dát z vozidla predspracovanie vykonal program LineCruiser Monitor. Odbremenil tak mikropočítač na vozidle od výpočtov, a tým sme získali čas potrebný na ovládanie ostatných periférií. V pokročilejších štádiách sme implementovali do programu vozidla prerušenia, ktorými sme nahradili aktívne čakania.

4.1.7 Aplikácia pre smartfón

V niektorých prípadoch sa ukázal program pre PC nedostačujúci. Pri ladení algoritmov a jazde sme zistili, že aj pri použití notebooku, nie sme schopní včas zareagovať a v prípade, že vozidlo vyjde z trate, môže dôjsť k nárazu. Keďže nemôžeme stáť

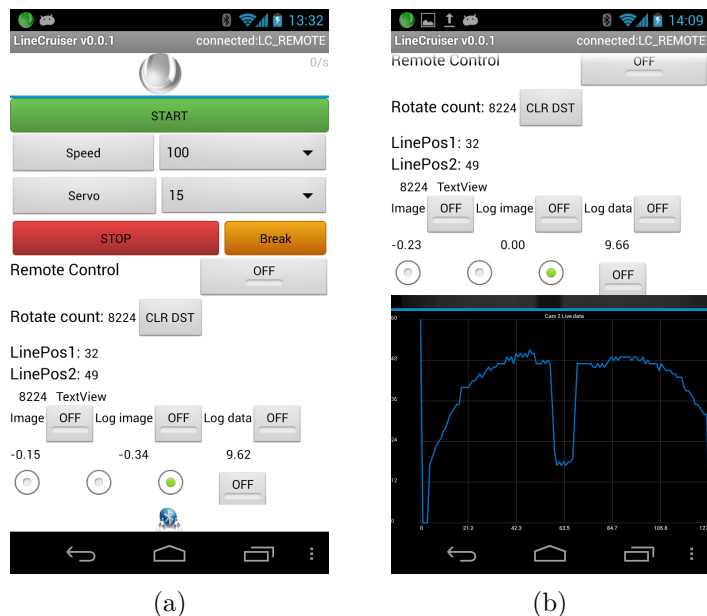
priamo nad traťou a pohotovo zareagovať na prípadné zlyhanie, rozhodli sme sa zaviesť alternatívne riešenie. Pristúpili sme k zámeru vytvoriť aplikáciu pre smartfón, s operačným systémom Android, ktorou by bolo možné uviesť vozidlo do pohybu a zároveň ho zastaviť. Po naštudovaní platformy Android sme vytvorili základnú kosť aplikácie. Nadviazanie spojenia bolo podobne ako pri aplikácii pre PC pomocou Bluetooth. Pridaná bola funkcia *AUTOCONNECT*, pomocou ktorej sa mobilný telefón ihneď po zapnutí aplikácie snaží spojiť s posledným pripojeným zariadením. Po úspešnom spojení sme implementovali už existujúci protokol, ktorý poskytoval všetky potrebné nástroje pre ovládanie vozidla. Najprv sme do programu zahrnuli základné tlačidlá na uvedenie vozidla do pohybu a jeho zastavenie. Postupne sme pridávali ďalšie funkcie podľa potreby. Aplikácia umožňovala zmerať prejdenú dráhu a tiež bolo možné túto hodnotu nulovať. Ďalej sme si nechali posielat polohu čiary z obidvoch kamier, ktorú nám sprostredkovali detekčné algoritmy, ktoré budú popísané v nasledujúcej kapitole. V istých kritických prípadoch bolo potrebné poslať celý obraz a vykresliť ho na displeji smartfónu. Jednalo sa predovšetkým o ostré zákruty, kde vodiaca čiara vybočila z pohľadu kamery a tiež pri zlom natočení kamery. Vďaka tomuto obrazu sme mohli prípadné nedostatky rýchlo a jednoducho odstrániť. Posielanie obrazu z oboch kamier v reálnom čase naraz nebolo možné, pretože to priepustnosť linky nedovoľovala. Zaostrovať a smerovať sme mohli iba jednu kameru naraz, čo v našom prípade nepredstavovalo problém. Pri testovaní, vo veľkej hale na súťaži v Paríži, sa ukázalo vhodné použiť bezdrôtové ovládanie vozidla, pretože na výber bolo viacero tratí a pomocou ručného ovládania bolo možné jednoducho prechádzať z jednej trate priamo na druhú. Preto sme zaviedli do tejto aplikácie získavanie hodnôt z akcelerometra, pričom tieto hodnoty boli použité na výpočet rýchlosti a nastavenie polohy servomotora. Ručné ovládanie je schematicky znázornené na obr. 4.3. Prepnutím tlačidla *Remote Control* je možné uviesť vozidlo znovu do autonómneho módu.



Obr. 4.3: Schéma ručného ovládania pomocou smartfónu

Do aplikácie sme tiež pridali nastavovanie parametrov vozidla, ktoré môžu byť menené aj počas jazdy. Žiadanú hodnotu rýchlosti si môže užívateľ vybrať z kompo-

nenty ListView. Podobne je možné nastaviť aj konštantu servomotora. Aplikácia tiež umožňuje vytvoriť záznam priebehu polohy čiary, prípadne celého obrazu. Vybrané dáta sú ukladané vo formáte CSV na SD kartu smartfónu. V prípade, že smartfón nedisponuje slotom na SD kartu, tieto dáta sú ukladané do internej pamäte. Tieto dáta môžu byť použité pre ďalšiu analýzu priebehu trasy.

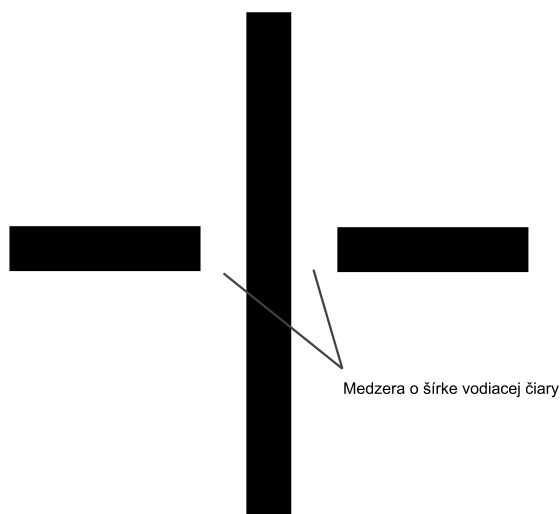


Obr. 4.4: Aplikácia LineCruiser: a) základné nastavenie vozidla a zobrazenie potrebných dát, b) graf a šedotónový obraz z kamery prenášaný počas jazdy vozidla

4.2 Algoritmy detekcie čiary

Trat vozidla je určená čiernou čiarou s šírkou 2,5 cm na bielej podložke. Takto vytvorená značka predstavuje dobrý kontrast a možnosť získať kvalitnú snímku.

Následne spracujeme obraz bežnými metódami spracovania obrazu. Porovnali sme niekoľko algoritmov, z ktorých každý mal svoje prednosti aj nevýhody. Okrem klasického nasledovania čiary bolo potrebné detekovať cieľ ktorý je znázornený na obr. 4.5.



Obr. 4.5: Vodiaca čiara so značkou cieľa

4.2.1 Minimálna suma na intervale

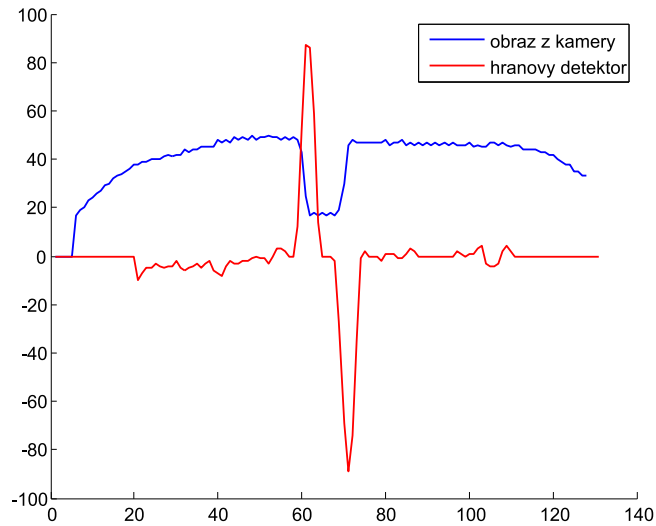
Prvý algoritmus na detekciu čiary predstavoval sumu jasových zložiek jednotlivých pixlov na danom intervale. Vybrali sme prvých 5 prvkov, ktoré sme sčítali a zapísali si súčet do pamäte. Prešli sme celý obraz a následne vybrali najnižšiu z hodnôt, ktorá sa v pamäti nachádzala. Bola to najtmavšia oblasť s najnižšou jasovou úrovňou a považovali sme ju za stred čiary. Tento jednoduchý algoritmus sa ukázal ako veľmi rýchly, spoľahlivý a odolný voči šumu a rušeniu. Dokonca aj na málo kontrastnom povrchu, ktorým bolo hnedé linoleum, bol tento algoritmus schopný detekovať čiaru. Problém však nastával v križovatkách a v cieľi, ktoré mohol nesprávne vyhodnotiť ako stred čiary.

4.2.2 Hranový detektor

Ďalším algoritmom, ktorý sme navrhli, bolo vyhodnocovanie stredu čiary pomocou hranového detektora. Vytvorili sme algoritmus na konvolúciu jednorozmerného obrazu, ktorého vstupom bol náš obraz z kamery a maska na detekciu hrán. Za hrany sme považovali susedné vzorky s veľkou diferenciou jasovej zložky. Masku sme vytvorili ako vektor o dĺžke 5 prvkov s nasledovnými hodnotami $[-2 -1 0 1 2]$. Výsledkom tejto konvolúcie bol signál s kladnými aj zápornými hodnotami. Hodnoty blízke 0 boli susedné prvky, ktoré nemali veľkú diferenciu jasovej zložky. Vysoké hodnoty predstavovali prechod z bielej-vysoká úroveň signálu do čiernej- nízka úroveň signálu. Záporné hodnoty predstavovali prechod z čiernej do bielej. Takto detekované hrany sme považovali za kraj čiary.

Pomocou vzorca 4.1 sme určili pozíciu stredu čiary v obraze.

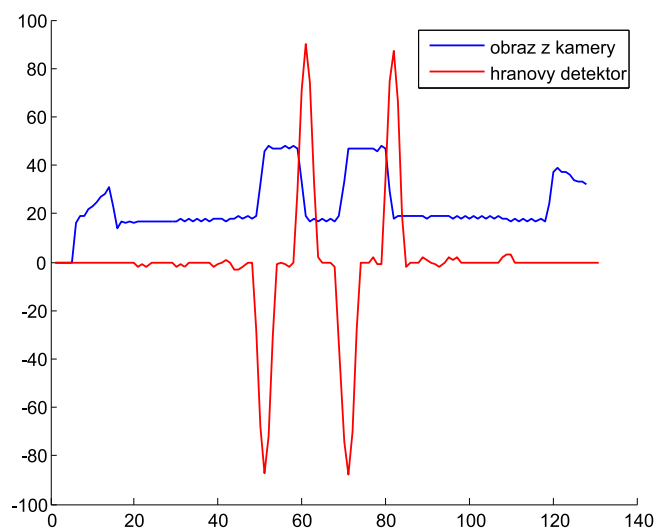
$$Index_{stred} = \frac{Index_{max} + Index_{min}}{2} \quad (4.1)$$



Obr. 4.6: Detekcia vodiacej čiary pomocou hranového detektoru

Ďalej sme si určili minimálnu a maximálnu šírku čiary, ktorými sa odstránil problém pri prechode križovatkou. Tento algoritmus sa tiež ukázal ako celkom vhodný, čo sa týka odolnosti voči šumu a správnej funkcii aj pri veľkej rýchlosti.

Tento algoritmus sme tiež v prvých štádiách používali aj na detekciu cieľa. Definovali sme si, že počet hrán v obraze musí byť rovný štyrom. Keďže dvadsať vzoriek z oboch strán je orezaných, nie je potrebné sa obávať falošnej hrany na krajoch obrazu. Problematické sa však ukázalo nastavenie hodnoty pre rozhodovanie či je v obraze hrana, alebo nie. Niekedy sme detekovali aj falošný cieľ pri spojnicach jednotlivých dielov trate, a preto sme od tohoto riešenia nakoniec upustili a cieľ sme sa rozhodli detekovať iným spôsobom.

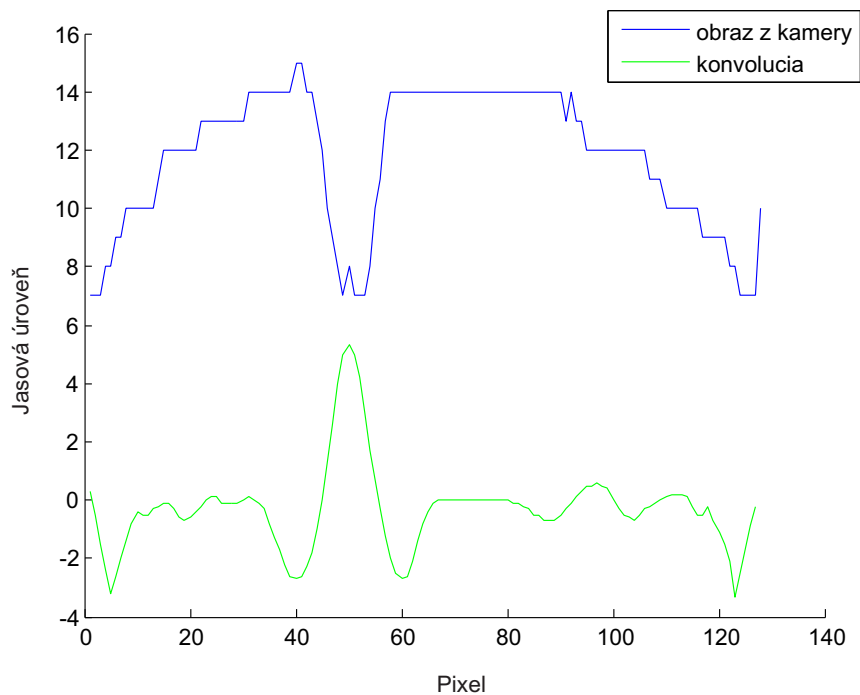


Obr. 4.7: Detekcia cieľa pomocou hranového detektoru

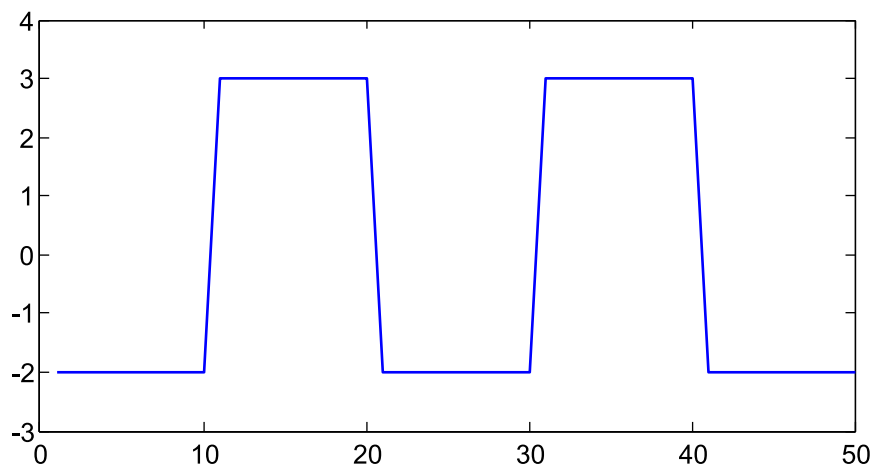
Porovnanie šablóny

Tento algoritmus vychádza z teoretického modelu čiary, s ktorým porovnáваме skutočný obraz. Konvolúciou obrazu a masky získame funkciu, ktorá nám popisuje, do akej miery sa šablóna zhoduje s obrazom v danej časti obrazu. Poloha maxima tejto funkcie nám hovorí, kde v obraze sa šablóna zhoduje s obrazom najviac. Po konvolúcii sme však dostali jednorozmerné pole, ktoré bolo dlhšie ako pôvodný obraz. Tento nežiadúci jav vytváral po okrajoch výrazné lokálne maximá, ktoré častokrát prevýšili skutočné maximum v obraze. Preto bolo potrebné z výsledného poľa odrezat prebytočné body o polovicu šírky masky z oboch strán. Šablónu sme navrhli tak, že sme si zaznamenali kvalitnú snímku čiary so silným osvetlením, ktorá bola umiestnená v strede. Následne sme si spočítali šírku čiary v pixloch. Táto hodnota sa pohybovala v medziach 9-12 pixlov. Konvolučnú masku sme teda navrhli ako dvadsať prvkové pole, ktoré malo v prvkoch 1 až 5 a 16 až 20 hodnotu $1/10$ a v prvkoch 6 až 15 hodnotu $-1/10$. Takáto detekcia čiary sa ukázala ako veľmi spoľahlivá. Výstup z konvolúcie obrazu je na obr. 4.8.

Jednoduchou zmenou masky so zachovaním algoritmu, môžeme navrhnúť aj detekciu cieľa.

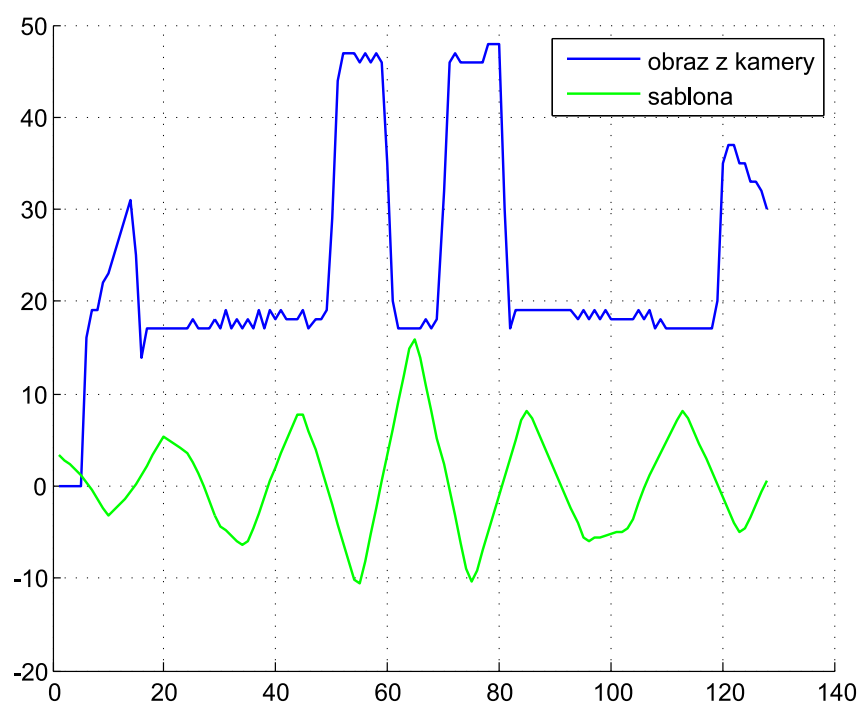


Obr. 4.8: Detekcia čiary pomocou šablóny



Obr. 4.9: Model koncovej značky-šablóna

Ciel má podobu kríža s medzerami o šírke čiary pri vodiacej čiare a je znázornený na obr. 4.5. Podľa toho sme navrhli šablónu, ktorá je graficky zobrazená na obr. 4.9. V tomto prípade rovnako hľadáme maximum z konvolúcie. Rozdiel je však v tom, že táto maska má odlišnú dĺžku, a preto je potrebné s týmto rozdielom rátať. Odčítame dĺžku masky ktorá je 50 bodov. Keďže táto maska je pomerne široká, môže nastať



Obr. 4.10: Detekcia cieľa pomocou šablóny

problém v prípade, že značka cieľa sa bude nachádzať na okraji obrazu. V popise trasy je však uvedené, že značka cieľa sa nikdy nenachádza v zákrute a je vždy na rovnom úseku. Rovný úsek má dĺžku minimálne 1 m. Preto s týmto obmedzením nemusíme rátať. Jediný prípad, kedy by mohlo vozidlo prejsť cez cieľovú značku šikmo, je zlyhanie detekčného algoritmu. Algoritmus však musí pracovať spoľahlivo, lebo vybočenie z trasy nie je v našom prípade tolerované. Ako vidíme na obr. 4.10, okrem maxima v strede sa v jeho okolí vyskytujú aj iné, lokálne maximá. Tieto sú však nežiadúce a preto musíme hladinu určenia cieľa postaviť tak, aby došlo k spoľahlivej detekcii, ale nedostali sme falošnú informáciu o detekovanom cieľi. Preto sme túto hodnotu nastavili na dvanásť, čo spĺňa túto požiadavku.

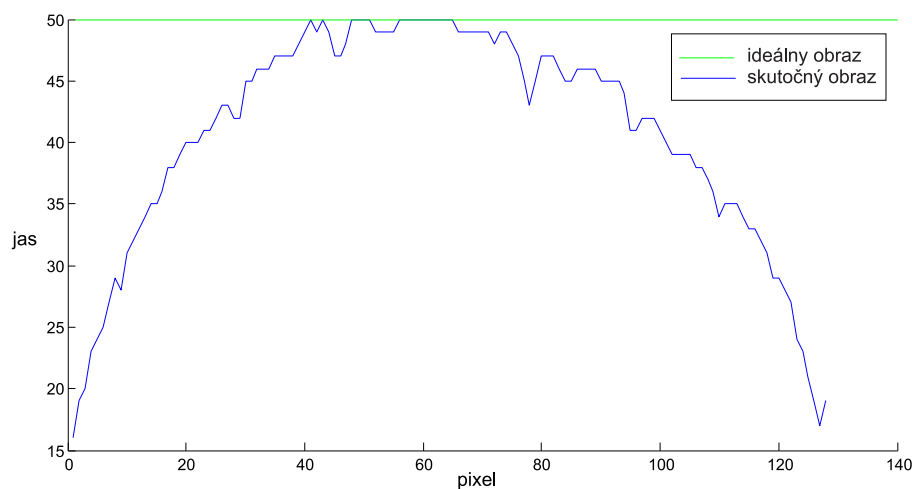
4.2.3 Zhrnutie detekcie vodiacej čiary a cieľa

Vo finálnom riešení sme zvolili nasledovný výber algoritmov. Pre detekciu vodiacej čiary sme vybrali hranový detektor s obmedzením minimálnej a maximálnej šírky vodiacej čiary. Pre detekciu cieľovej značky sme vybrali algoritmus porovnávania šablóny. Bolo potrebné určiť minimálnu prejdenú vzdialenosť, pretože cieľová značka je zároveň aj štartovacou značkou a vozidlo by ihneď po štarte zastalo. Preto sme pristúpili k riešeniu merania vzdialenosti s použitím odometrie. Vozidlo musí dostať

zo snímačov na kolesách aspoň sto impulzov, aby došlo pri prechode cieľovou značkou k zastaveniu vozidla.

4.2.4 Porovnanie výsledkov rôzneho umiestnenia senzorov

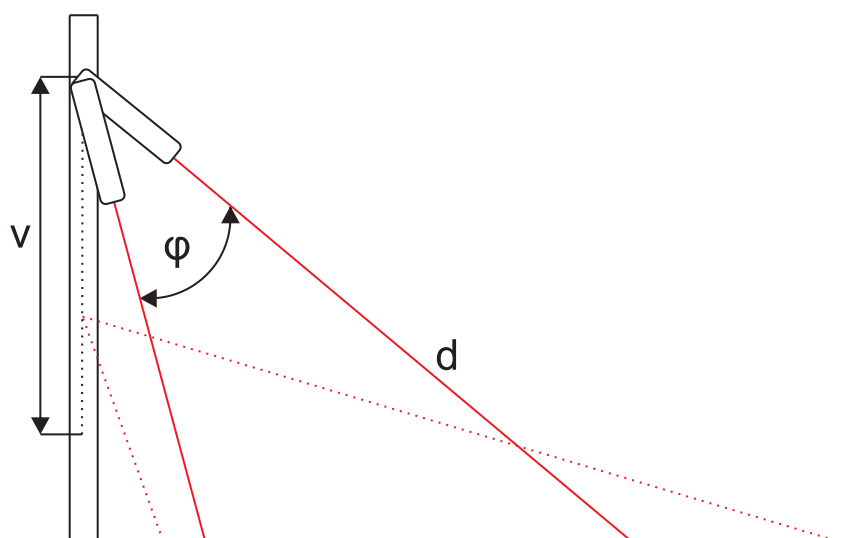
Ku kamerovému senzoru bola dodaná optika, ktorá spôsobuje skreslenie pri krajoch obrazu. Urobili sme snímku s bielym pozadím. Ideálne by sme mali dostať po vynešení do grafu z riadkovej kamery čiaru, vodorovnú s osou x. Namiesto čiary je však v grafe parabola. Názorne je táto skutočnosť zobrazená na obr.4.11.



Obr. 4.11: Chyba optiky po stranách obrazu

Tento útlm bol taký veľký, že na okraje čipu nedopadalo takmer žiadne svetlo. Preto sme kraje obrazu nemohli vôbec použiť na detekciu. Tým nám vznikli požiadavky na umiestnenie kamerového senzoru. Naším cieľom bolo umiestniť kameru čo najvyššie, aby sme dostali dostatočne široký a použiteľný záber. Zároveň sme nemohli umiestniť senzory do veľmi veľkej výšky lebo vozidlo muselo byť schopné prejsť tunelom pri maximálnej výške 32 cm. Preto sme vozidlo vybavili dvomi lištami, ktoré boli schopné niesť aj kamerový senzor aj osvetlenie. Keďže tieto lišty boli zo známej stavebnice Merkúr, mohli sme konštrukciu navrhnuť tak, aby bolo možné kameru nastavovať v dvoch smeroch.

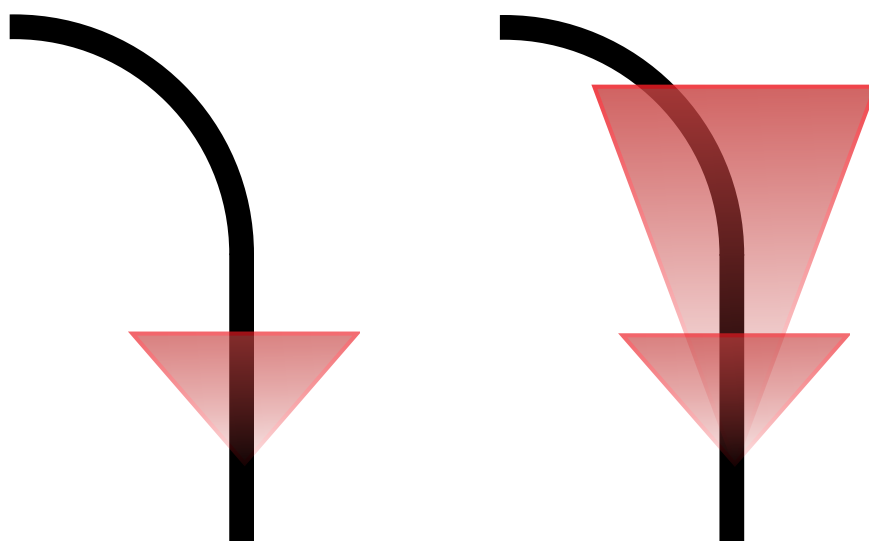
Bolo možné nastavovať výšku snímača(v) a zároveň jeho sklon(φ). Touto kombináciou nastavujeme vzdialenosť kamery od podložky(d). Názorná ukážka je na obr. 4.12. Čím je vyššia vzdialenosť d , tým širší záber z kamery získame. Ak ju však umiestnime tak, že vzdialenosť d bude príliš veľká, znamená to, že pri prejazde zákrutou nastane moment, kedy kamera vodiacu čiaru z obrazu stratí. Kameru sme teda umiestnili na úplný vrchol vodiacej lišty aby sme získali čo najširší záber a pod takým uhlom, aby sme pri prejazde zákrutou nestrácali z obrazu čiaru.



Obr. 4.12: Možnosti nastavovania kamery

4.2.5 Použitie dvoch senzorov vodiacej čiary

Pri použití jednej kamery potrebujeme snímať polohu vodiacej čiary no zároveň potrebujeme vedieť tendenciu smerovania čiary ďalej od vozidla. Keďže ide o dve protichodné požiadavky rozhodli sme sa vyhovieť obom v plnej miere. Pridali sme ďalšiu kameru, pričom zorné pole prvej kamery bolo tesne pred vozidlom a zorné pole druhej kamery bolo vo vzdialenosti približne 30 cm pre vozidlom. Zorné polia tohto prevedenia sú schematicky znázornené na obr. 4.13. Kamera, ktorá má zorné pole bližšie k vozidlu musí byť umiestnená vo väčšej výške aby sme získali širší záber obrazu a nedošlo k strate vodiacej čiary pri ostrých zákrutách. Z tohto obrázku tiež vyplýva, že v prípade prejazdu zákrutou získame informáciu o blížiacej sa zákrute výrazne skôr a môžeme sa na jej prejazd patrične pripraviť. Pokiaľ umiestnené kamery majú rozdielne vzdialenosti d , potom sa zmení aj šírka čiary v obraze kamery. Kamera, ktorá má vyššiu vzdialenosť d , musí mať tiež rozdielne nastavené hraničné parametre čiary. Definovali sme si ich ako maximálna a minimálna šírka čiary. Minimálnu šírku musíme zmenšiť, lebo pokiaľ bola v bežnom stave šírka čiary 10 pixlov, tak pri naklonení kamery so zorným polom 30 cm od vozidla je to už iba 8 pixlov. S rastúcou vzdialenosťou by táto šírka ešte klesala a už by sme neboli schopní správne detekovať vodiacu značku.



Obr. 4.13: Porovnanie použitia jednej kamery a dvoch kamier

4.3 Softvér pre vozidlo

V tejto časti bude popísaná inicializácia mikroprocesora, spôsob vyčítavania obrazu z kamery a tiež postup merania otáčok vozidla.

4.3.1 Popis potrebných periférií

Pre komunikáciu s rôznymi typmi snímačov a tiež na riadenie sme využili rôzne typy periférií. Podľa toho bolo potrebné navrhnuť jednak hardvérové prepojenie, ale aj softvérové spracovanie prijatých informácií. Na prácu s kamerou bolo potrebné vyhradiť jeden analógový vstup a jeden digitálny výstup. Pre komunikáciu sme vyhradili dva piny procesora sériovej linky TXD a RXD. Ďalšie piny boli využité na ovládanie radiča motorov. Na začiatku to boli len signály PWM, no neskôr sme pridali aj možnosť zmeny smeru každého motora zvlášť. Odlišný pin bol použitý pri ovládaní servomotora. Digitálne vstupy boli použité pri odometrii na snímanie otáčok kolies.

4.3.2 Inicializácia

V prvej fáze sme použili nástroj Processor Expert a celý systém sme uviedli do chodu. Keď sme boli presvedčení o správnej funkcii, došla na rad optimalizácia a kód Processor Expertu sme nahradili vlastným.

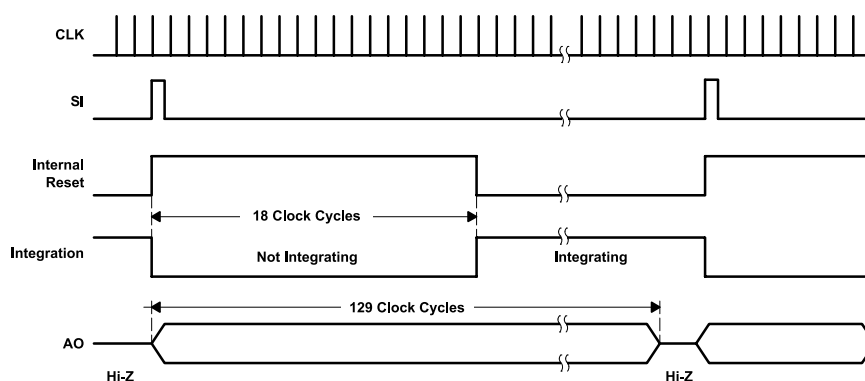
Na začiatku programu sa vykoná inicializácia hodín, pričom sa tento signál privedie na všetky používané periférie. Povolia sa prerušenia na ADC0 a ADC1. Nastavia sa všetky vstupy a výstupy. Následne dochádza k nastaveniu všetkých potrebných časovačov. Tieto časovače slúžia na generovanie signálov CLK pre kamery, meranie rýchlosti vozidla, spúšťanie brzdy a generovanie signálov PWM. Následne dochádza k nastaveniu registrov sériovej linky. Konfigurácia UART je nasledovná. Prenosová rýchlosť BAUD RATE je 115200, počet prenášaných bitov je 8 bez použitia paritných bitov. Stop bit je jeden a Handshaking nie je využívaný.

Celý program sme sa snažili v čo najväčšej miere integrovať do prerušení, aby sme získali čo najviac procesorového času na potrebné výpočty. Najnáročnejšou operáciou je konvolúcia pri detekcii cieľa, pretože maska je veľmi široká a tým je potrebné vykonať veľmi veľa vzájomných celočíselných súčinov.

4.3.3 Získavanie obrazu z kamery

Na to, aby sme mohli získať obraz z kamery umiestnenej na vozidle, bolo potrebné naštudovať spôsob získavania jednotlivých pixlov z kamery. Bližší pohľad je na obr. 4.14. Signál CLK určuje interval vyčítavania jednotlivých pixlov obrazu. Na

spustenia vyčítavania je potrebný signál SI, pričom trvanie tohto signálu na vysokej úrovni musí byť kratšie ako perióda signálu CLK. Následne môžeme vyčítavať jednotlivé pixle obrazu. Na analógovom výstupe môžeme A/D prevodníkom merať príslušné úrovne zodpovedajúce jasovej zložke jednorozmerného signálu. Počas prvých osemnástich cyklov sa neintegruje. Integrácia sa začína až nasledujúcim hodinovým impulzom a integruje sa až do ďalšieho príchodu signálu SI. Po dokončení vyčítavania je analógový výstup v stave vysokej impedancie. V našom prípade sme použili kontinuálne vyčítavane obrazu a ihneď po získaní jedného snímku sa vyšle ďalší signál SI.



Obr. 4.14: Časové priebehy kamery TSL1401 [5]

4.3.4 Meranie otáčok kolies

Rýchlosť otáčajúcich sa kolies vieme v našom prípade zistiť dvomi spôsobmi. Pri prvom z nich zmeriame čas medzi vzostupnou a zostupnou hranou. Tento spôsob je pomerne efektívny, no je veľmi citlivý na šum. Často sme dostávali falošné impulzy, čo sa nepriaznivo prejavilo pri meraní. Tieto falošné impulzy boli pravdepodobne spôsobené osvetlením a odleskami z okolia. Nedokázali sme sa pred nimi efektívne brániť, a preto sme zvolili druhú techniku merania, ktorá nám zabezpečila spoľahlivejšiu informáciu o rýchlosti. Získame ju tak, že počítame vzostupné a zostupné hrany počas určeného časového intervalu. My sme interval stanovili na 100 ms. Počet hrán bol pri maximálnej rýchlosti približne 80. Pri jednom, prípadne dvoch falošných impulzoch v tomto prípade je chyba do 5%, čo je akceptovateľné.

V prípade, že vozidlo narazilo, alebo z iného dôvodu odometer nedostáva žiadne impulzy, je potrebné vozidlo zastaviť. Príčin môže byť niekoľko. Rozpojený prívodný vodič PWM, zlyhanie odometrie, či iné poruchy sa môžu podpísať pod zlyhanie riadenia a uvedenie vozidla do nekontrolovateľného stavu. Preto sme zaviedli kontrolu pomocou odometrie. Ak vozidlo prestane dostávať impulzy zo snímača automaticky do jednej sekundy zastane. Ak sú kolesá blokované, ochráni to tiež motory, ktoré by sa mohli prehriať a nenávratne zničiť. V štádiách ladenia sa ukázala táto funkcionálna ako nutnosť.

4.4 Riadenie vozidla

Riadenie vozidla je uskutočnené pomocou natáčania predných kolies a reguláciou rýchlosti. V nasledovnej časti si popíšeme návrh celkového riadenia.

4.4.1 Natočenie predných kolies

Natočenie predných kolies je zabezpečené privádzaním PWM signálu do riadiaceho servomotora. Tento signál je generovaný pomocou časovača, striedu určuje hodnota v registri *FTM2_COV*.

4.4.2 Určenie strednej polohy servomotora

Na našej konštrukcii sa vyskytol hardvérový problém, ktorý spočíval v natočení kolies aj pri stredovej polohe servomotora. Rameno, ktoré bolo uchytené na servomotor pri silnom zaťažení preklzovalo. Pravdepodobne išlo o ochranu motora pred násilným otáčaním kolies, no v praxi sa to ukázalo ako problematické a museli sme s tým v programe rátať. Najprv sme teda zvolili softvérové riešenie tohto problému. Určili sme si konštantu *SERVO CENTER POSITION*, ktorou sme nastavili hodnotu PWM, pri ktorej sú kolesá vozidla v priamom smere. Túto hodnotu sme podľa potreby menili, no vždy bol potrebný zásah do programu. Nakoniec sme pristúpili k riešeniu hardvérovým spôsobom a to tak, že vždy pred jazdou sme vykonali ručnú kalibráciu kolies a tým sme nastavili servomotor do strednej pozície.

4.4.3 Natáčanie kolies na základe polohy čiary

Po správnom detekovaní polohy vodiacej čiary bolo možné pristúpiť k vytváraniu algoritmu pre natáčanie predných kolies. Poloha čiary je v rozmedzí -42 až 42 pixlov, pričom ak je hodnota nulová, čiara je uprostred obrazu z kamery vozidla. Natáčanie kolies sme realizovali P regulátorom. Odchýlku od strednej polohy sme násobili zosilnením, ktoré sme museli experimentálne určiť. Pre nižšie hodnoty zosilnenia vozidlo prechádzalo traťou veľmi plynule, no pri ostrejších zákrutách už nebolo schopné trať prejsť. Pri vyšších hodnotách zosilnenia sme zaznamenali presné kopírovanie trate, no so zvyšujúcou sa rýchlosťou došlo k silnému kmitaniu vozidla okolo vodiacej čiary, čo spôsobilo vybočenie z trate. Preto sme museli nájsť kompromisné riešenie. Najlepšie jazdné vlastnosti sme dosiahli pri hodnote zosilnenia 22.

4.4.4 Regulácia rýchlosti vozidla

Pohyb vozidla umožňujú dva jednosmerné motory, ktoré sú v zadnej časti vozidla pod doskou radiča. Krútiaci moment je prenášaný na prevodovku a odtiaľ priamo

na kolesá. Kolesá sú vzájomne prepojené oskou. Táto oska je však volnobežná a umožňuje rozdielne otáčky oboch kolies. Strieda signálu PWM vstupujúca do radiča určuje otáčky motora. Ďalším pinom je možné nastavovať smer otáčania. Maximálnu striedu PWM sme nepoužili, pretože pri tejto rýchlosti sme nedokázali vozidlo bezpečne ovládať.

4.4.5 Jednoduché nastavovanie rýchlosti

Na začiatku sme zaviedli jednoduché nastavovanie otáčok motorov nastavením konštantnej hodnoty do registrov *FTM0_C0V* a *FTM0_C3V*. Túto hodnotu sme mohli nastavovať pomocou nášho komunikačného protokolu prostredníctvom bezdrôtového rozhrania. Postupne sme zvyšovali hodnotu rýchlosti a vyladzovali algoritmus natáčania kolies. Následne sme zvyšovali rýchlosť, až do hranice kedy sa vozidlo stabilne držalo trate. Dosiahli sme stav, kedy sme už neboli schopní dosiahnuť vyššiu rýchlosť. Pri jej zvýšení už vozidlo vyšlo z trate.

4.4.6 Pokročilé techniky nastavovania rýchlosti

Keďže sme už jednoduchým spôsobom nedokázali bezpečne zvýšiť rýchlosť, skúšali sme pokročilé metódy nastavovania rýchlosti. Navrhli sme P regulátor rýchlosti, ktorého vstupný parameter bol rozdiel žiadanej hodnoty a hodnoty z enkódera. Na rovine sa vozidlo držalo trate, avšak zákruty prechádzalo rovnako vysokou rýchlosťou. Bolo potrebné správne detekovať zákruty a tie prechádzať nižšou rýchlosťou. Použili sme dve kamery, pričom jednej zorné pole bolo priamo pod vozidlom a druhé bolo 30 cm pred vozidlom 4.13. Algoritmy detekcie čiary boli v oboch prípadoch rovnaké. Jediným rozdielom bolo, že detekcia cieľa prebiehala iba z obrazu prvej kamery. Detekciu zákruty sme vykonávali tak, že bolo potrebné aby sa vozidlo nachádzalo na rovnom úseku a následne poloha čiary na druhej kamere musela byť vyššia ako prednastavená úroveň. Túto úroveň sme nastavili na ± 20 pixlov. Pri prejazde ostrejšími zákrutami sa však stávalo, že predná kamera stratila z dohľadu vodiacu čiaru. Preto sme boli nútení upraviť algoritmus detekcie čiary a zaviesť stav, kedy sa čiara v obraze nenachádza. Problém sa vyskytol pri vychádzaní zo zákruty, kedy sme taktiež detekovali zákrutu. Preto sme vykonali opatrenie, ktorým sme tento problém eliminovali. Pri detekcii zákruty sme znížili rýchlosť a zvýšili sme ju až v štádiu, keď bolo vozidlo vycentrované na stred vodiacej čiary. Pri detekcii zákruty sme nastavili rýchlosť na nulu a vozidlo prešlo zákrutu iba vlastnou zotrvačnosťou.

4.4.7 Aktívne brzdenie

Aby sme dosiahli ešte vyššiu rýchlosť, pri ktorej by vozidlo dokázalo bezpečne prejsť určenú trasu, pristúpili sme k použitiu aktívneho brzdenia. Pri hodnote PWM nula sú motory vo voľnobehu, no sú čiastočne brzdené. Napätie, ktoré sa naindukuje na motoroch voľnobežným otáčaním kolies, je zrážané radičom. Tým dochádza k čiastočnému brzdeniu.

Pre silnejšie brzdenie je potrebné pripojiť na motory napätie opačnej polarita aká bola na motoroch v bežnej prevádzke. Silu brzdenia potom môžeme meniť rôznymi spôsobmi. Dôležitý je čas brzdenia a intenzita. Vhodným nastavením týchto parametrov môžeme na rovine zvýšiť rýchlosť a pred blížiacou sa zákrutou bezpečne zabrzdiť. Ako sa už z prvých jázd s aktívnym brzdením ukázalo, nebolo dobré príliš prudké brzdenie. Vozidlo sa tak dostávalo v zákrutách do šmyku a prejavovalo sa to hlavne agresívnou jazdou. Ak bol čas brzdenia dlhší ako 300 ms a striedou PWM s opačnou polaritou nad 50%, tak dochádzalo k pretáčaniu kolies v opačnom smere. Vtedy už išlo o neriadený šmyk a vozidlo bolo nekontrolovateľné. Preto sme experimentálne nastavovali hodnoty pre brzdiaci algoritmus.

Intenzitu brzdenia však chceme nastavovať podľa rýchlosti vozidla. Ak má vozidlo nízku rýchlosť, je potrebné brzdenie celkom vyradiť. Preto sme ako parameter brzdového algoritmu nastavili aktuálnu rýchlosť vozidla. So zvyšujúcou sa rýchlosťou potom rastie aj doba brzdenia, a tým aj celková intenzita brzdenia. Určili sme dve rýchlostné úrovne. Pokiaľ počet impulzov z odometra presiahne hodnotu 50, nastaví sa vyššia rýchlostná úroveň a zvýši sa strieda PWM v reverznom smere na 75%. V opačnom prípade zostane strieda na 50%.

4.4.8 Elektronický diferenciál

Pri prechode zákrutou dochádza k rozdielnej dĺžke prejdenej dráhy vnútornými a vonkajšími kolesami vozidla. Tým sa zvyšuje požadovaná rýchlosť vonkajšieho, prípadne znížená rýchlosť vnútorného kolesa. Pri iných modeloch nie je možnosť diferenciálu, pretože tieto modely majú iba jeden motor, a tým pádom toto riešenie neumožňuje rozdielne nastavovanie rýchlosti kolies. Náš model však disponoval dvomi motormi, a preto sme naskytla možnosť využiť diferenciál.

Vytvorenie matematického modelu pre tento účel sa zdalo príliš komplikované, a preto sme sa rozhodli meniť rozdiel rýchlostí vnútorného a vonkajšieho kolesa experimentálne. Zistili sme, že v našom prípade je dôležitejší algoritmus natáčania kolies a regulácie rýchlosti, pričom diferenciál sa veľmi na jazdných vlastnostiach neprejavoval. Častejšie dochádzalo k zhoršeniu jazdných vlastností a stáčaniu vozidla v zákrute. Mierne vyššiu stabilitu pri prechode zákrutou sme zaznamenali v prípade nižšej hodnoty diferenciálneho koeficientu.

Elektronický diferenciál sme realizovali tak, že sme pri zápise do registra PWM zohľadnili natočenie kolies. Tento koeficient mohol nadobúdať hodnoty ± 21 . Ak vozidlo zatáčalo doľava tak sa k registru PWM pravého kolesa pričítala kladná hodnota a k PWM ľavého záporná a ak vpravo tak naopak. Pri jazde po rovine bola hodnota koeficientu nulová. Keďže sa hodnota koeficientu pri prechode zákrutou zvyšuje lineárne podľa natočenia kolies, nedochádza k prudkej zmene rýchlosti kolies.

5 VYHODNOTENIE DOSIAHNUTÝCH VÝSLEDKOV

V tejto kapitole zhrnieme všetky dosiahnuté výsledky z jednotlivých oblastí návrhu vozidla tak, ako boli použité vo finálnom prevedení.

5.1 Osvetlenie scény

Vozidlo bolo schopné jazdiť aj pri zníženom osvetlení bez použitia dodatočného osvetlenia scény. Pri prejazde tunelom, prípadne pri použití nevhodného blikajúceho svetla bolo potrebné scénu dostatočne osvetliť, na čo sa nám hodila podľa meraní najviac biela LED.

5.2 Detekcia čiary

Na detekciu sme použili dve jednoriadkové kamery s 200 snímkami za sekundu. Táto frekvencia snímania poskytovala dostatok snímkov aj na detekciu cieľa pri vyššej rýchlosti. Na detekciu vodiacej značky sme použili hranový detektor. Cieľ sme detekovali konvolúciou obrazu a šablóny, ktorá bola modelom cieľovej značky. Po konvolúcii sme hľadali maximum.

5.3 Komunikácia s vozidlom

Navrhli sme protokol na komunikáciu s vozidlom s použitím Bluetooth modulu. Takto sme dokázali získavať dáta z vozidla v reálnom čase. Vytvorili sme si niekoľko štandardných paketov, ktoré sme po prenose analyzovali. Pakety boli rozdelené na príkazové a dotazovacie. Príkazovými sme mohli nastavovať režimy jazdy vozidla, prípadne premenné do všetkých algoritmov a nulovať počítadlo celkovej prejdenej dráhy. Dotazovacími sme získavali dáta o aktuálnej polohe čiary, rýchlosti, natočení servomotora, prípadne celý obraz z kamery. Ak sme však prenášali obraz, dokázali sme získať iba 1-5 fps, za čo mohla nízka priepustnosť sériovej linky. Pre nastavovanie kamier to bolo postačujúce. Vytvorili sme aplikáciu pre PC a neskôr aj pre platformu Android. Obe aplikácie umožňujú ovládanie vozidla a zároveň sledovanie požadovaných parametrov.

5.4 Meranie rýchlosti

Meranie otáčok zadných kolies bolo zabezpečené opticky, pomocou reflexného snímača CNY70 a disku s 24 čierno-bielymi plochami. Signál zo snímača vyhodnocoval operačný zosilňovač v zapojení ako Schmittov preklápací obvod, ktorého výstup bol pripojený priamo k procesoru. Náhodné falošné impulzy boli vyfiltrované tak, že procesor počítal počet impulzov za 100 ms. Novú hodnotu rýchlosti sme teda dostávali 10-krát za sekundu.

5.5 Riadenie

Nastavovanie žiadanej rýchlosti bolo dvojstupňové v závislosti na tom, či išlo vozidlo po rovine alebo v zákrute. V prípade, že program vyhodnotí obraz z kamery, ktorej zorné pole sa nachádza ďalej pred vozidlom, že sa jedná o zákrutu a aktuálna rýchlosť prekračuje nastavenú hodnotu, spustí sa brzdenie vozidla. Intenzita brzdenia je lineárne závislá na rýchlosti vozidla. Ak má vozidlo pri prejazde zákrutou len malú rýchlosť brzdiaci režim sa neaktivuje. Pri prejazde zákrutou sa tiež uplatní elektronický diferenciál, ktorý k hodnote registrov PWM pričíta aktuálnu polohu čiar.

5.6 Jazda

Pri optimálnom nastavení všetkých konštánt a plne nabitej batérii, sme zistili najkratší možný čas pri prejazde malým okruhom. Tento okruh pozostával z dielcov 2.1, 2.2, 2.4 a rovných dielcov, v celkovej dĺžke 6,5 m. Najnižší dosiahnutý čas bol 5,9 s. Tým dostávame priemernú rýchlosť 1,1 m/s. Aktuálny európsky rekord trate dĺžky 44.1 m je čas 25,8 s z čoho plynie priemerná rýchlosť 1,7 m/s. Celkový pohľad na trať je na obr. A.2 v prílohe A.

6 ZÁVER

Zoznámili sme sa s hardvérovými možnosťami modelu a riadenia jednotlivých motorov umiestnených vo vozidle. Návrh sme rozdelili na hardvérovú a softvérovú časť.

Po prvotnom uvedení vozidla do chodu sme zvolili vhodné umiestnenie kamery, aby bol obraz čo najmenej skreslený a kamera nebola rušená nežiadúcim bočným svetlom. Rozhodli sme sa pre použitie dvoch kamier, pričom obraz z jednej bol snímaný priamo pod vozidlom a obraz z druhej 30 cm pred vozidlom. Tiež sme zistili, že DPS kamier zo zadnej strany presvitá, a preto je potrebné ich zakryť tmavou nepresvitajúcou fóliou, v našom prípade elektrikárskou izolačnou páskou. Pre nízku expozičnú dobu pri slabom osvetlení, sme tiež boli nútení zaviesť osvetlenie scény pomocou LED. Zmerali sme jednotlivé spektrálne charakteristiky rôznych LED, z čoho sme vybrali najvhodnejšieho kandidáta. Napriek tomu, že najväčšiu intenzitu dosahovala červená LED, cez závislosť snímača na danej vlnovej dĺžke obr. 3.9 sme zistili, že biela LED je najvhodnejším riešením. Predpokladali sme, že vhodným typom bude infračervená LED, pretože v tejto oblasti dochádza k najmenšiemu útlmu signálu. Tento predpoklad sa však nenaplnil.

Vybrali sme vhodné komunikačné rozhranie, ktorým bol Bluetooth. Tento výber bol kompromisom medzi prenosovou rýchlosťou, jednoduchosťou implementácie a cenovou dostupnosťou. Navrhli sme tiež komunikačný protokol, ktorý mal byť šetrný k množstvu prenášaných dát. Napriek úsiliu o optimalizáciu protokolu sa nám nepodarilo prenášať plynulý obraz z kamery v reálnom čase, získali sme iba 1-5 snímok za sekundu pri dobrej kvalite signálu. Túto možnosť sme využili iba pri nastavovaní kamier pred jazdou. Počas jazdy nebola táto funkcionálna potrebná. Počas jazdy sme však prenášali spracované dáta, ktoré boli potrebné prevažne pri ladení algoritmov, ale aj pri sledovaní parametrov vozidla počas jazdy. Sledovanými parametrami bola prejdená rýchlosť, otáčky kolies, akčný zásah do motorov, poloha servomotora a poloha stredu sledovanej čiary od oboch kamier.

Navrhli sme ovládací program, ktorým je možné sledovať stav uvedených parametrov a zároveň zasiahnuť do jeho riadenia. Taktiež je možné nastavovať rôzne režimy jazdy, čo sa uplatnilo predovšetkým pri ladení riadiacich algoritmov. Nastavovali sme predovšetkým polohu stredu servomotora, požadovanú rýchlosť a konštantu pre servomotor. Vytvorili sme tiež aplikáciu pre smartfón, postavenú na platforme Android. Aplikácia mala rovnakú funkcionálnu ako pre PC, pridali sme do nej však možnosť prepnutia medzi automatickým a manuálnym ovládaním. Výhodou tejto aplikácie v smartfóne bola hlavne mobilnosť, pričom sme mohli spúšťať vozidlo a nastavovať jeho parametre priamo pri trati bez použitia notebooku.

Pre spoľahlivosť algoritmu detekcie vodiacej čiary sme porovnali viac rozdielnych variant. Postupovali sme od tých najjednoduchších k zložitejším. Najspoľahlivejším

algoritmom pri jednoduchej detekcii sa stal ten, ktorý bol zároveň najjednoduchším. Hľadaním minimálnej sumy v obraze sme získali polohu čiary. Problém však nastal pri komplikovanejších miestach trate, a preto sme boli nútení aplikovať zložitejšie varianty. Hranový detektor na obr. 4.6 sa osvedčil ako najlepší pri detekovaní vodiacej značky a metóda porovnávania šablóny na obr. 4.10 tiež vyriešila problém s detekciou cieľa. Algoritmus porovnávania šablóny sa na detekciu ukázal ako veľmi spoľahlivý a tiež odolný voči šumu.

Po spoľahlivej detekcii vodiacej značky sme mohli pristúpiť k návrhu algoritmu na zatáčanie predných kolies. Poloha stredu vodiacej čiary bola v intervale ± 42 pixlov. Na nastavovanie polohy servomotora sme použili iba obraz z kamery, ktorej obraz bol snímaný v tesnej blízkosti pred vozidlom. Vyfiltrovanú polohu čiary sme mohli priamo použiť na nastavovanie polohy servomotora. Táto filtrácia čiastočne zvýšila odozvu, no bola nutná pre spoľahlivý prechod vozidla cez trať.

Navrhli sme riadenie rýchlosti vozidla, ktoré je poháňané dvomi jednosmernými motormi s prevodovkou. Zaviedli sme snímanie otáčok, pretože motory, ktoré poháňajú kolesá, nemajú zavedenú spätnú väzbu. Pre minimalizovanie rušenia sme na prírodné vodiče motorov umiestnili feritové jadrá a na kontakty motorov sme priadli keramické kondenzátory. Žiadaná rýchlosť motorov sa mení v závislosti na tom, či sa vozidlo pohybuje po rovine, prípadne prechádza cez zákrutu. K tomu nám slúži obraz z kamery, ktorá sníma scénu vo vzdialenosti 30 cm pred vozidlom. Pri blížiac sa zákrute vozidlo včas spomalíme a zabránime vybočeniu vozidla zo zákruty. Na mierne zlepšenie jazdných vlastností sme zakomponovali do vozidla elektronický diferenciál, ktorý pri prechode zákrutou na vonkajšom kolese mierne pridá a na vnútornom čiastočne uberie výkon a tak získame stabilnejší prejazd zákrutou.

Priestor pre pokračovanie tejto práce vidím v možnosti identifikácii modelu vozidla a vylepšení algoritmu nastavovania rýchlosti, napríklad regulátorom PSD. Detekcia tendencie smerovania vodiacej čiary by mohla byť realizovaná tak, že vozidlo by informáciu z dvoch kamerových snímačov používalo ako jeden vektor, ktorého poloha a orientácia by slúžila k nastavovaniu rýchlosti a tiež by slúžila pre nastavovanie servomotora. Zlepšila by sa tým stabilita vozidla predovšetkým na trati typu vlnka.

LITERATÚRA

- [1] Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis, and Machine Vision, 3rd Edition*, Thomson 2007, ISBN 049508252X
- [2] Everett, H., R.: *Sensors for Mobile Robots theory and application*, CRC Press 1995, ISBN 1568810482
- [3] The Freescale Cup Track Components. In: *The Freescale Cup Track Components od autora fuadamjala – Galéria 3D* [online]. 2011-04-07 [cit. 2013-05-12]. Dostupné z URL: <<http://sketchup.google.com/3dwarehouse/cldetails?mid=1b8f905e694b06b850f087be02267415&ct=mdcc>>.
- [4] Freescale Semiconductor, Inc.: *MC33931, 5.0 A Throttle Control H-Bridge - Data Sheet* [online]. 2012-10 [cit. 2013-05-12]. Dostupné z URL: <http://cache.freescale.com/files/analog/doc/data_sheet/MC33931.pdf>.
- [5] Taos, Inc: *TSL1401R-LF 128 × 1 linear sensor array with hold* [online]. 2009-04 [cit. 2013-05-12]. Dostupné z URL: <<http://www.dese.iisc.ernet.in/extra/tfcindia2012/Datasheet/TSL1401R-LF-B.pdf>>.
- [6] Siemens: *Uni- and Bipolar Hall IC Switches for Magnetic Field Applications* [online]. 1997-01 [cit. 2013-05-12]. Dostupné z URL: <<http://www.gme.sk/dokumentace/533/533-439/dsh.533-439.1.pdf>>.
- [7] SLABPROG: *HC-0X Bluetooth-UART modules* [online]. 2013 [cit. 2013-05-12]. Dostupné z URL: <<http://silabs.org.ua>>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

AGV automaticky riadené dopravné zariadenie - automated guided vehicle

COM Sériový port

SPP protokol sériovej linky - Serial Port Protocol

SPI Sériové periférne rozhranie - Serial Peripheral Interface Bus

UART Asynchrónne sériové rozhranie - Universal Asynchronous Receiver/Transmitter

RXD Vodič na prijímanie dát sériovou linkou - Receive Data

TXD Vodič na vysielanie dát sériovou linkou - Transmit Data

PWM pulzná šírková modulácia - Pulse-width modulation

TTL tranzistorovo tranzistorová logika - transistor-transistor logic

CMOS Complementary metal-oxide-semiconductor

MCU Jednočipový mikropočítač - Microcontroller unit

DMA priamy prístup do pamäte - Direct memory access

RAM Pamäť s voľným prístupom - Random Access Memory

SRAM Statická pamäť s voľným prístupom - Static Random Access Memory

RTC Hodiny reálneho času - Real Time Clock

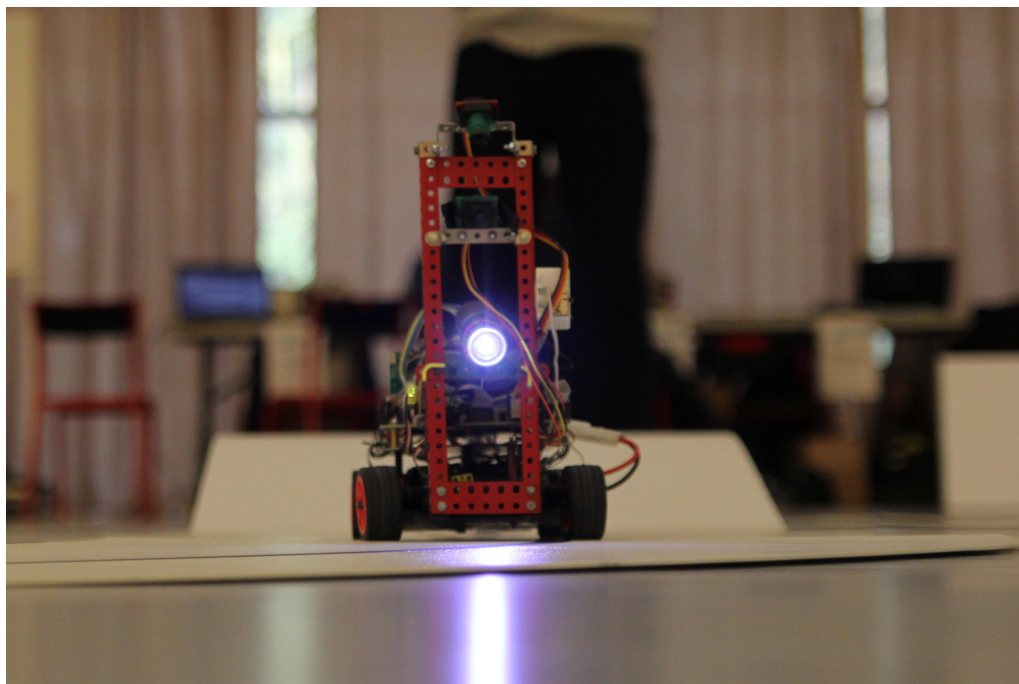
CLK Signál hodinového pulzu- Clock

LED Luminiscenčná dióda - light-emitting diode

ZOZNAM PRÍLOH

A	Fotografie trate a vozidla	58
B	CD s programom LineCruiserMonitor pre PC, LineCruiser pre platformu Android, zdrojovým kódom k vozidlu, fotografie vozidla	60

A FOTOGRAFIE TRATE A VOZIDLA



Obr. A.1: Vozidlo na trati



Obr. A.2: Celkový pohľad na trať počas európskeho finále



Obr. A.3: Zariadenie na meranie času a diváci na európskom finále Freescale Cup 2013

**B CD S PROGRAMOM LINECRUISERMONI-
TOR PRE PC, LINECRUISER PRE PLATO-
FRMU ANDROID, ZDROJOVÝM KÓDOM
K VOZIDLU, FOTOGRAFIE VOZIDLA**