



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Model autonomního vozu sledujícího čáru

## Bakalářská práce

*Studijní program:* B 2646 – Informační technologie  
*Studijní obor:* 1802R007 – Informační technologie  
*Autor práce:* **Milan Honců**  
*Vedoucí práce:* Ing. Jan Koprnický, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# Autonomous line following car

## Bachelor thesis

*Study programme:* B 2646 – Information technology  
*Study branch:* 1802R007 – Information technology  
*Author:* **Milan Honců**  
*Supervisor:* Ing. Jan Koprnický, Ph.D.



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Milan Honců**  
Osobní číslo: **M13000102**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Model autonomního vozu sledujícího čáru**  
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

### Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s elektromechanickými částmi podvozku a s řídicí jednotkou.
2. Zprovozněte hardwarovou část systému.
3. Navrhněte algoritmus řízení pro sledování čáry a realizaci parkovacího manévru.
4. Vytvořte odpovídající software pro řídicí jednotku modelu vozu.
5. Funkční model otestujte na laboratorní dráze.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **cca 30–40 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] The Freescale Cup Knowledge Center. FREESCALE. The Freescale Cup Knowledge Center [online]. 2012 [cit. 2015-10-08]. Dostupné z: <https://community.freescale.com/docs/DOC-1284>
- [2] FREESCALE. MPC560xB Controller Board User's Guide. MPC560XBMCBUG, Rev. 0, 08/2012. 2012.

Vedoucí bakalářské práce:

**Ing. Jan Koprnický, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce:

**Ing. Pavel Herajn**

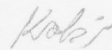
Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: **10. října 2015**

Termín odevzdání bakalářské práce: **16. května 2016**



prof. Ing. Václav Kopecký, CSc.  
děkan



doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu



## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 16.5.2016

Podpis: Milan Honců  


## Abstrakt

Tato bakalářská práce se zabývá vývojem a testováním řídicího algoritmu vyvinutého pro autonomní model vozidla od společnosti Freescale. Dále jsou testovány HW úpravy provedené na vozidle, umožňující reverzaci motorů. Závěrem je popsáno testování a odladění algoritmu na testovací dráze.

**Klíčová slova:** Freescale, EMEA, algoritmus řízení jednořádkové kamery

## Abstract

The thesis deals with the development and testing of a control algorithm developed for an autonomous vehicle model from Freescale company. Further the vehicle HW adjustments, which allows engine reversing, are tested. In conclusion, testing and debugging of the algorithm, that was made on a test track, is described.

**Keywords:** Freescale, EMEA, algorithm of control one-liner camera

## Poděkování

Poděkování za tuto práci patří především Ing. Janu Koprnickému, Ph.D., pod jehož odborným vedením byla práce sepisována a algoritmus laděn.

Dále bych chtěl poděkovat panu Petru Stančíkovi ze společnosti Freescale, který mi pomocí e-mailové komunikace pomohl pochopit principy fungování tohoto vývojového kitu.

Nesmím rozhodně opomenout poděkovat panu docentu Satrapovi za perfektní balík tulthesis, se kterým je sazba textu v systému L<sup>A</sup>T<sub>E</sub>X radostí.

A v neposlední řadě patří mé poděkování rodině a všem, jež mě v mých studijních aktivitách podporují.

# Obsah

Seznam zkratek . . . . .	10
<b>Úvod</b>	<b>11</b>
<b>1 Motivace</b>	<b>12</b>
1.1 EMEA . . . . .	12
1.1.1 Vybraná pravidla EMEA 2016 . . . . .	12
<b>2 Teoretická a rešeršní část</b>	<b>14</b>
2.1 Hardware . . . . .	14
2.1.1 Freescale kit . . . . .	14
2.1.2 Deska TRK-MPC5604B/C . . . . .	15
2.1.3 Mechanické spojky vodičů (jumpery) . . . . .	15
2.1.4 Konektory TRK-MPC5604B/C . . . . .	17
2.1.5 Deska The FC Motor Drive Version A . . . . .	19
2.1.6 Lineární kamera . . . . .	20
2.1.7 Motory, baterie . . . . .	21
2.1.8 Změny HW . . . . .	22
2.2 Software . . . . .	23
2.2.1 Využitý software . . . . .	23
2.2.2 CodeWarrior . . . . .	23
2.2.3 Terminal . . . . .	23
2.2.4 Sample Project . . . . .	24
<b>3 Praktická část</b>	<b>25</b>
3.1 Nástin řešení . . . . .	25
3.1.1 Sledování čáry . . . . .	25
3.1.2 Řízení motorů . . . . .	26
3.2 Možné realizace parkovacího manévru . . . . .	26
3.2.1 Enkodér . . . . .	26
3.2.2 Odometrie . . . . .	27
3.3 Vlastní algoritmus . . . . .	27
3.3.1 Čekací fáze . . . . .	27
3.3.2 Zpracování dat z kamery . . . . .	27
3.3.3 Rychlost . . . . .	28
3.3.4 Prahovací funkce . . . . .	28

3.4	HW úpravy . . . . .	30
3.4.1	Vnější úprava . . . . .	30
3.4.2	Softwarové testování provedených úprav . . . . .	31
3.5	Problémy při realizaci . . . . .	32
3.5.1	32bit OS . . . . .	32
3.5.2	Chybné měření kamery . . . . .	32
3.5.3	Funkce Delay() . . . . .	32
3.5.4	Vybitá baterie . . . . .	32
3.6	Testování . . . . .	33
	<b>Závěr</b>	<b>34</b>
	<b>Přílohy</b>	<b>37</b>
	<b>A Obsah přiloženého CD</b>	<b>37</b>

## Seznam obrázků

1.1	Ukázka tratě podle pravidel v roce 2015 . . . . .	13
2.1	Boční pohled . . . . .	14
2.2	Horní pohled . . . . .	14
2.3	Základní jumpery . . . . .	16
2.4	Řídicí deska . . . . .	18
2.5	Výkonová deska . . . . .	19
2.6	Led diody a tlačítka . . . . .	19
2.7	Motory zadní nápravy . . . . .	21
2.8	Servomotor přední nápravy . . . . .	21
2.9	Propojení prvků vozu . . . . .	22
2.10	Vývojové prostředí CodeWarrior . . . . .	23
2.11	Terminál . . . . .	24
3.1	Nástin řešení . . . . .	25
3.2	Diagram navrhnutého algoritmu . . . . .	29
3.3	Pravdivostní tabulka [15] . . . . .	30
3.4	HW úpravy vozidla . . . . .	31
3.5	Trať v laboratořích TUL . . . . .	33

## Seznam zkratek

**RAM** Random Access Memory — paměť s libovolným přístupem

**HW** Hardware — fyzické vybavení přístroje

**PWM** Pulse Wide Modulation — pulzně šířková modulace

**GND** Ground — uzemnění

**I/O** Input/Output — vstupně/výstupní

**EMEA** Název soutěže, při které se pořádají závody vozů Freescale

**CAN** Controller Area Network — sběrnice

**DC** Direct Current — stejnosměrný proud

**LIN** Local Interconnect Network — sběrnice

**LED** Light Emitting Diode — dioda emitující světlo

# Úvod

Tato práce se zabývá vývojem a testováním algoritmu pro autonomní model vozidla od společnosti Freescale. Dále jsou testovány realizované HW úpravy pro možnost reverzace motorů. V první části se práce zabývá motivací a zpracováním rešeršní části na toto téma.

Popsány jsou základní schopnosti modelu včetně popisu jeho hlavních periférií. Následuje popis a postup vývoje vlastního algoritmu pro pohyb vozidla. Popsány jsou také vybrané realizované stěžejní funkce, které mají pro algoritmus nepostradatelný význam. Jelikož se práce neobešla bez komplikací, v poslední části jsou tyto popsány.



# 1 Motivace

Motivací této práce bylo zrealizovat a následně otestovat algoritmus, kterým by bylo možné splnit požadavky pro kvalifikaci do závodů, jež se s těmito modely organizují. Také bylo třeba otestovat funkčnost a správné provedení HW úprav, které byly realizovány pro letošní rok.

Toto téma jsem si vybral z důvodu mé záliby v robotice a automatizaci. Zpracováním bakalářské práce jsem si osvojil jednoduché zpracování obrazu, inicializaci vývojového procesoru a vytváření jednoduchých řídicích algoritmů. Deska modelu obsahuje také sběrnice typu RS-232, LIN a CAN, které jsou pro mě velmi zajímavé, a proto jsem se s nimi zkusil částečně seznámit.

## 1.1 EMEA

Soutěž, během níž se pořádají závody s těmito modely se jmenuje EMEA [10]. Týká se především studentů technických univerzit. Každý rok jsou specifikovány požadavky pro možnost účasti na této soutěži.

Trať je složena z několika částí. Může obsahovat tunely, mosty a závody mohou probíhat ve tmě, na což jsou ovšem potřebné dodatečné úpravy, podmiňující správné snímání kamery, jako například přídavné led osvětlení.

Povolené úpravy na modelu jsou specifikovány. Některé části kitu je nutné zachovat, jiné je možné demontovat a nahradit jinou součástí. Pravidla jsou velmi striktní a před každým závodem probíhá technická přejímka a kontrola modelu vozu. Týmům, jež jakýmkoliv způsobem poruší podmínky stanovené v pravidlech, není start v soutěži umožněn.

V minulých letech došlo k zásadní změně v úpravě trati. Dříve byla trať opatřena černou čarou uprostřed cesty. V současné době je auto navigováno a řízeno podle krajnic silnice. Novou podobu tratě můžete vidět na obrázku 1.1.

### 1.1.1 Vybraná pravidla EMEA 2016

- Požadavky na tým:
  - maximální velikost týmu jsou 3 členové, minimální počet 2
  - tým smí být složen pouze ze studentů
  - studenti nemusí být z jedné univerzity
  - ...

- Náležitosti a povolené modifikace modelu:
  - nutnost použití Freescale 32-bit Microcontroller
  - procesor může být na modelu pouze jeden
  - model musí být navigován pomocí optického senzoru, žádný jiný způsob není povolen
  - není povolen DC-DC měnič
  - povolena pouze jedna baterie s kapacitou  $\leq 3000$  mAh
  - ...



Obrázek 1.1: Ukázka tratě podle pravidel v roce 2015

Jelikož byl algoritmus psán pro starý typ tratě, nebylo by možné se zúčastnit soutěže podle aktuálních pravidel. Nutnými podmínkami podmiňující start v soutěži by bylo poskládání týmu, výměna kamery a v neposlední řadě přepsání a precizní odladění algoritmu.

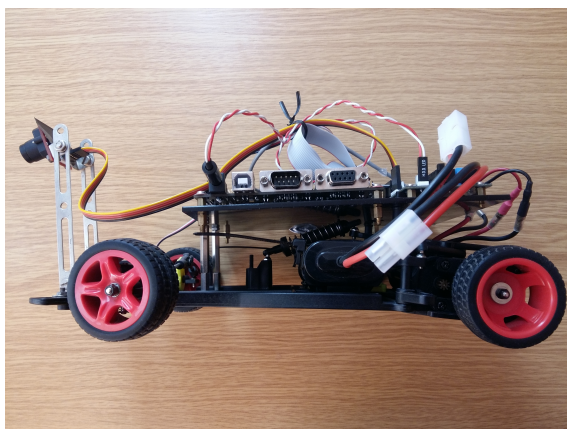
## 2 Teoretická a rešeršní část

### 2.1 Hardware

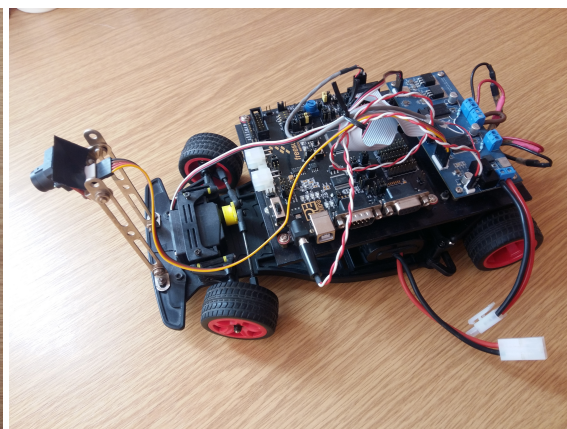
#### 2.1.1 Freescale kit

Základem tohoto projektu je kit od společnosti Freescale [13], který obsahuje základní komponenty pro stavbu vozidla ve velikosti 1/18. Podvozek je tvořen plastovým rámem, na němž jsou uchyceny ostatní komponenty. Vývojový kit je jednoduchý, slouží pouze ke svému účelu výuky programování procesorů. Základní části tvoří tyto komponenty (bude rozebráno později):

- řídicí deska MPC 5604B/C,
- výkonná deska s H-můstky (Motor drive version A),
- baterie pro napájení,
- servomotor pro natáčení předních kol,
- elektromotory pro řízení otáčení zadních kol,
- lineární kamera.



Obrázek 2.1: Boční pohled



Obrázek 2.2: Horní pohled

## 2.1.2 Deska TRK-MPC5604B/C

Tato deska [3] disponující 32-bitovým procesorem se stará o zpracování veškerých instrukcí vstupujících do vývojového kitu. Obsahuje několik konektorů, přes které je možná komunikace s deskou, a tím i celým modelem. Program je možné nahrávat do paměti typu FLASH či SRAM.

### RAM

Jedná se o nejrychlejší typ paměti. Problém je ovšem v jejich nestálosti, což znamená, že po odpojení od zdroje napájení dojde k jejich vymazání. Tomuto jevu se říká volatilita paměti. Paměti RAM se dělí na SRAM a DRAM. SRAM jsou statické paměťové buňky realizovány jako bistabilní klopné obvody. Oproti tomu dynamické RAM jsou na výrobu mnohem jednodušší, a tudíž i značně levnější. Paměťové buňky jsou zde tvořeny pomocí tranzistorů. Nevýhodou je, že se obsah každé paměťové buňky musí pravidelně obnovovat.

Naše deska je opatřena 32 kB paměti typu SRAM.

### FLASH

Jedná se o nevolatilní paměť s libovolným přístupem. Data jsou ukládána v poli unipolárních tranzistorů s plovoucími hradly. V našem případě je deska opatřena FLASH pamětí o velikosti 512 kB.

## 2.1.3 Mechanické spojky vodičů (jumpery)

Na desce se nachází také velké množství propojovacích konektorů neboli jumperů. Pro nás jsou podstatné následující jumpery:

- POWER\_SELECT
- OSJTAG\_IRQ

### OSJTAG\_IRQ

Propojení toho jumperu najde uplatnění v případě, že budeme chtít přehrát firmware desky. Této možnosti bylo využito před začátkem programování algoritmu, kdy došlo k nahrání aktuální verze. Učiněno tak bylo z důvodu nefunkčnosti vývojového softwaru s aktuální verzí OS. Ani tento update ovšem nepomohl s problémy popsány v kapitole 3.5.1 na straně 32.

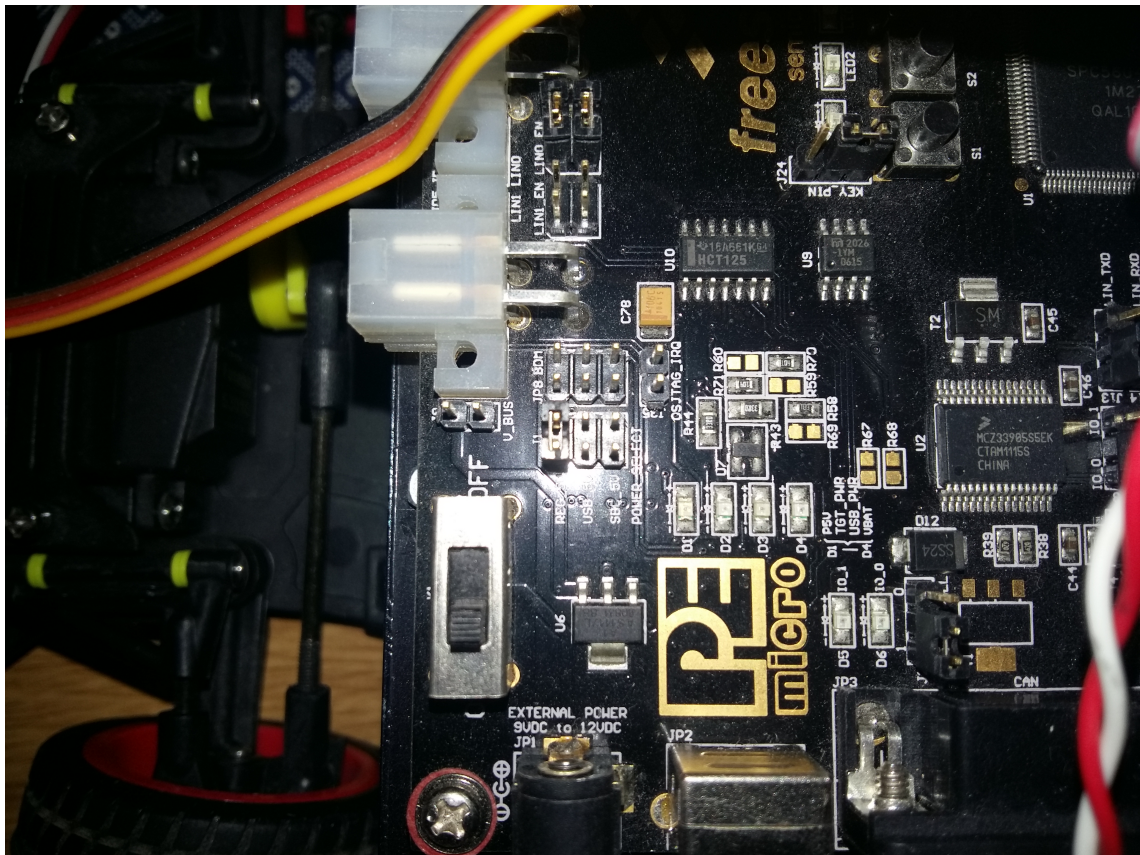


## POWER\_SELECT

Tento jumper slouží pro výběr varianty, jakým způsobem má být deska napájena. Jsou zde tři možnosti propojení:

- REG\_5V
- USB\_5V
- SBC\_5V

Využito bylo možnosti napájení USB\_5V v procesu seznamování s deskou. Toto zapojení postačuje pro obsluhu základních funkcí desky, jakými jsou například rozsvícení LED diod a stisky tlačítek. Pro složitější úlohy jako jsou roztáčení motorů je zapotřebí přepojit jumper na variantu REG\_5V, která nastaví jako výchozí příjem napájení z napájecího konektoru popsaného v kapitole 2.1.4 na straně 18.



Obrázek 2.3: Základní jumpery

## 2.1.4 Konektory TRK-MPC5604B/C

Mezi nejvýznamnější konektor, kterým je tato deska opatřena, je bezpochyby port USB, díky němuž probíhalo připojení k PC a ladění v programu terminál. Dalšími konektory umístěnými na hlavní desce jsou:

- RS-232
- LIN
- CAN
- PORT[A-H]
- napájecí konektor

### RS-232

Tento konektor se používá jako komunikační rozhraní pro spojení osobních počítačů a další elektroniky. Propojení probíhá skrze sériovou komunikaci, což znamená, že bity jsou přenášeny sekvenčně za sebou. Počet přenášených datových bitů je volitelný, nejčastěji se ovšem používá varianta s osmi bity, kdy je pro logickou „1“ napětí - 12V a pro logickou „0“ 12V. Další informace o sběrnici: [14]

### CAN

Jedná se o sběrnici, která se používá především v automobilovém průmyslu. Jako v předchozím případě se jedná o sériovou sběrnici. Parametry přenosu jsou definovány normou ISO 11898. Data jsou odesílána v rámcích, z nichž každý může obsahovat až 8 datových bajtů.

### LIN [9]

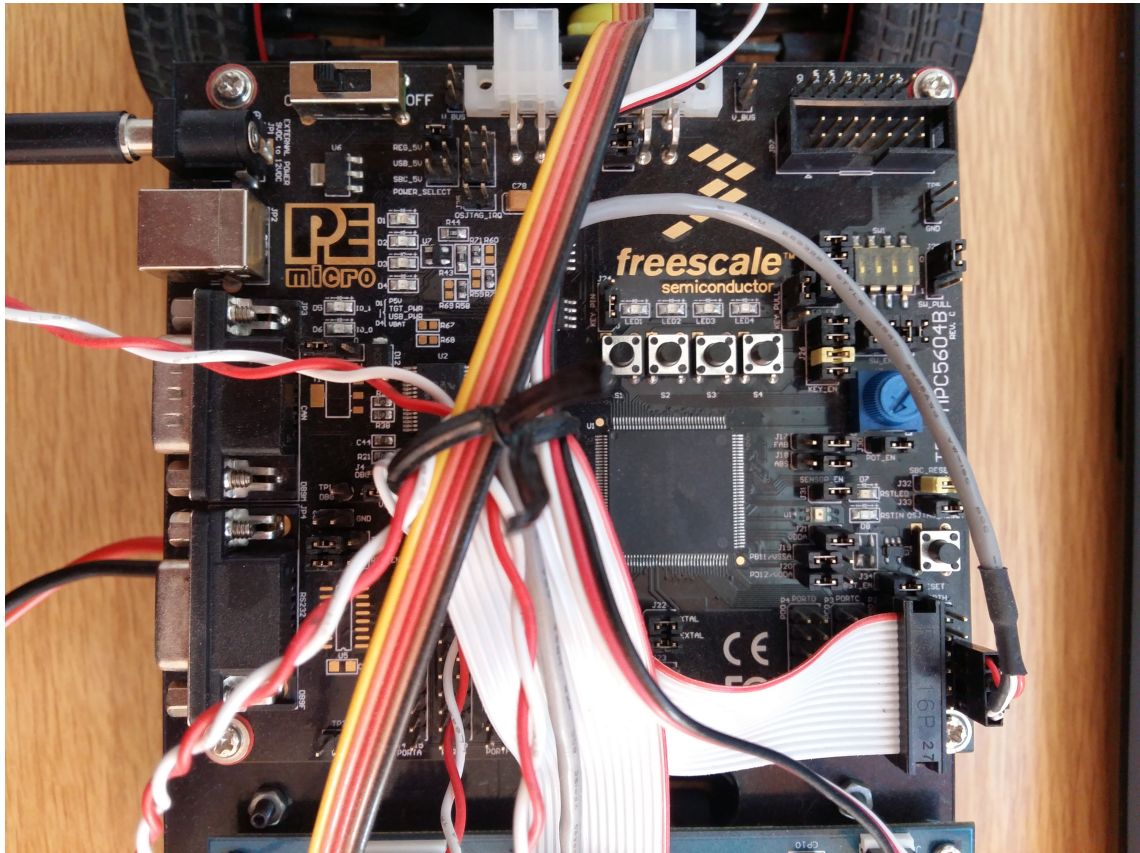
LIN je jednoduchá jednovodičová sběrnice vhodná pro přenos dat mezi řídicí jednotkou, smart senzory a aktuátory. Rychlost komunikace se pohybuje mezi 2400 a 19200 bit/s. Jedná se o poměrně novou sběrnici, jež byla poprvé použita v automobilech roku 2001.

### PORT[A-H]

Na desce se nachází mnoho programovatelných pinů, které jsou rozděleny na PORTY. Portů se zde nachází celkem 8 a to s označením PORT[A-H], podporovaná funkce dostupné na konkrétních pinech lze vyčíst z datasheetu desky. Na každém portu, kromě H, se nachází 15 pinů. Na portu H je 12 programovatelných pinů. Využitý je celý port B, který je spojen s výkonovou deskou. Dále bude využit port E pro schopnost reverzace.

## Napájecí konektor

Konektor kulatého průřezu nacházející se v levé horní části desky slouží jako napájecí. Sem je přiveden vodič z výkonové desky, který se stará o napájení.

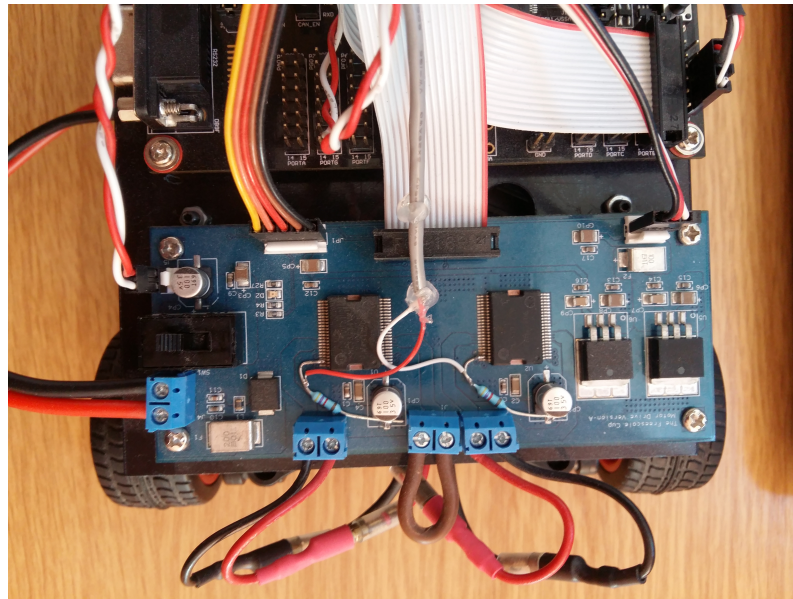


Obrázek 2.4: Řídicí deska



## 2.1.5 Deska The FC Motor Drive Version A

Výraznou součástí modelu je tato deska propojující veškeré komponenty s napájením. Jedná se o starší typ desky (rev. 0), jenž byl prodáván před rokem 2013. Dnes je hojně využívána deska novější (rev. 1). Hlavním rozdílem mezi těmito deskami je fakt, že modernější deska má oproti staršímu typu možnost přímé reverzace motorů bez dodatečných HW úprav. Další výhodou je eliminace elektro-magnetické interference, která způsobovala rušení signálu z lineární kamery a snižování výkonu servomotoru.



Obrázek 2.5: Výkonová deska

### LED diody a tlačítka

Na této desce se nachází 4 programovatelné led diody zelené barvy, přímo sousedící se stejným počtem tlačítek. Na obrázku 2.6 lze vidět také jumpery povolující LED diody a tlačítka.



Obrázek 2.6: Led diody a tlačítka



### 2.1.6 Lineární kamera

Lineární kamera, která je předsazena před vozidlo na kovové rampě smontované ze stavebnice Merkur, se jako jediná stará o získávání vstupních dat z okolí, a tudíž poskytuje rozhodující větvení pro další vykonávání algoritmu. Kamera je spojena propojovacím kabelem přímo s výkonnou deskou. Vodiče, jež jsou potřebné ke správnému fungování kamery jsou ground, power, SI (seriový vstup), AO (analogový výstup), CK (hodinový signál).

Možné je také přímo nastavení zaostřovací vzdálenosti na kameře pomocí otočné čočky. Plošný spoj, na němž je kamera uchycena, má ze zadní části spoje nalepenou černou pásku, jež eliminuje světelné paprsky dopadající ze špatné strany panelu. Tímto dochází k lehkému potlačení rušivých informací, které by mohly mít na výsledné snímání kamery neblahý vliv. Kamera produkuje 128 hodnot z fotodiod, jež jsou v rozmezí 0 – 5 V. Zpracování dat z kamery se nejčastěji provádí následujícími způsoby:

#### Prahová funkce

Jedná se o funkci, která na základě nastaveného, či vypočteného vhodného prahu upravuje nasnímané hodnoty. Hodnoty vyšší než je práh nastaví na hodnotu „log. 1“, menší či rovné prahu na „log. 0“.

#### Korelace

V tomto případě je nutné získat kvalitní vzor pro zpracování dat. Algoritmus načte z paměti uložený vzor a porovná s daty získanými z kamery. Na základě vyhodnocení rozdílu mezi těmito dvěma vzorky se vypočítá akční zásah.

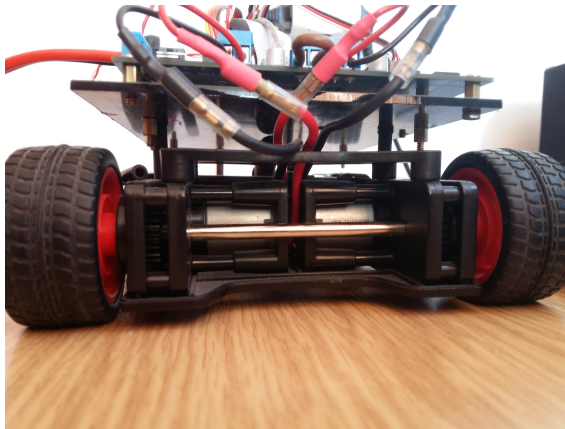
Tento způsob je ovšem poměrně výpočetně náročný, a tudíž je pro naši potřebu rychlého vykonávání algoritmu nevhodný.

## 2.1.7 Motory, baterie

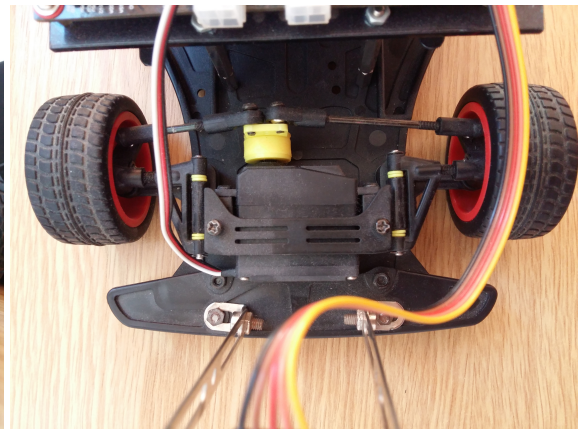
O natáčení předních kol se stará servomotor uchycený k ramenům kol. Vzdálenost a korekce natáčení je možná přes závěsy, které jsou vyrobeny ze závitových tyčí. O pohon zadních kol se starají dva motory, kde každý pohání jedno hnané kolo. Otáčení kol na sobě není závislé, proto je možnost jakési simulace diferenciálu. Rychlost otáčení kol je řízena pomocí střídavy. Baterie které se nachází na tomto modelu disponuje kapacitou 3000 mAh a napětím 7.2 V. Baterie je připojena na výkonovou desku modelu. Aktuální stav akumulátoru není bohužel nijak indikován. V případě nedostatečného nabití se model chová nepředvídatelně a je jen těžko odhadnutelné, že tyto stavy má za následek baterie.

### Servomotor

Rozdíl oproti klasickým motorům je v tom, že u servomotoru lze nastavit přesnou polohu natočení. Řízení probíhá pomocí PWM, pokud je střída nastavená na 50 %, motor bude ve středu svého natočení. Při snižování či zvyšování poměru střídavy dojde k jeho vychýlení buď ve směru vlevo, nebo vpravo.



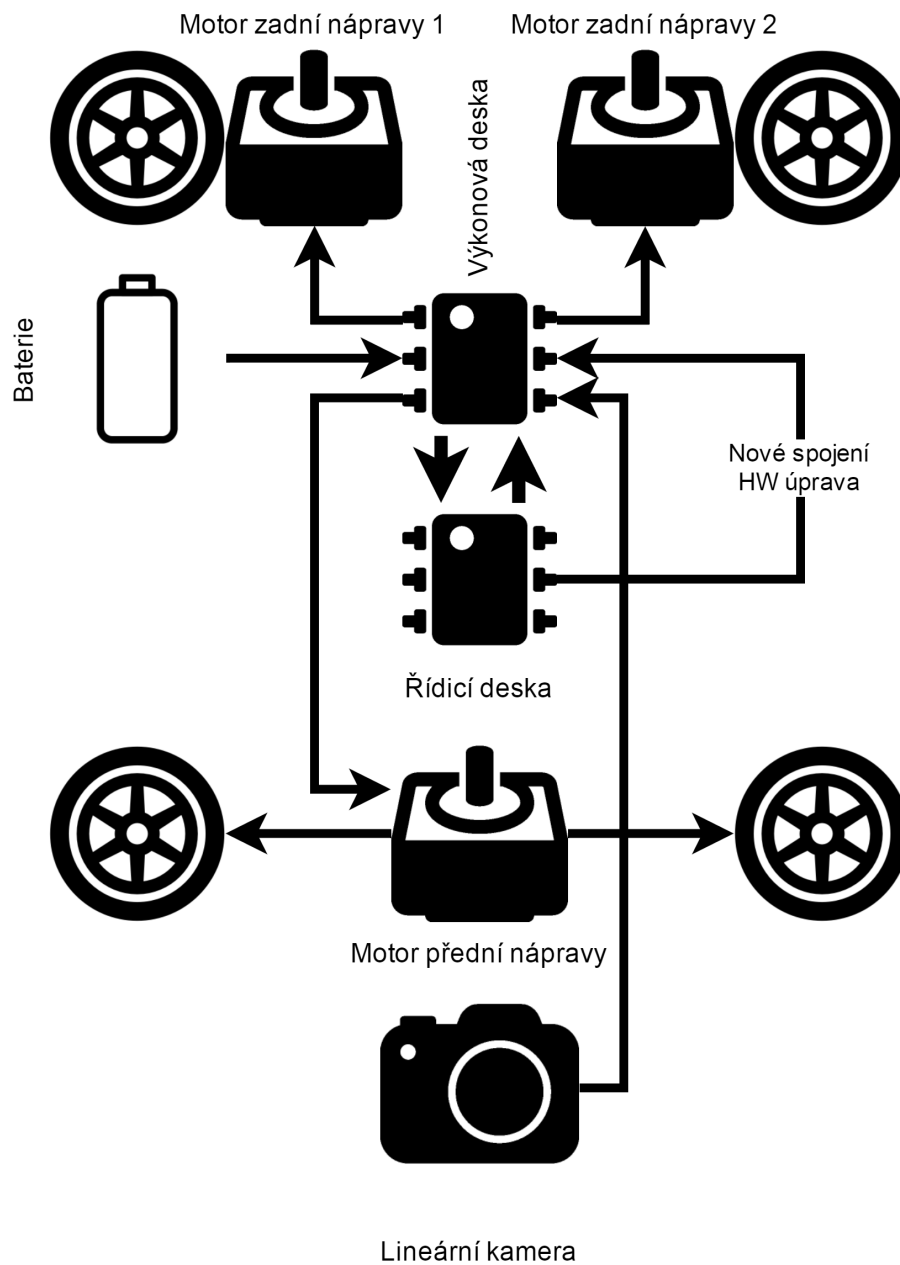
Obrázek 2.7: Motory zadní nápravy



Obrázek 2.8: Servomotor přední nápravy

## 2.1.8 Změny HW

Oproti minulým letům, kdy byl tento model využíván pro BP a PRJ, došlo k výrazné úpravě HW. Jedná se o spojení mezi výkonovou a řídicí deskou, které umožňuje přímou reverzaci motorů, a proto je zajištěna možnost pohybu vzad i u této starší výkonové desky s označením Motor Drive Version A. Předpoklady nutné pro splnění podmínek reverzace najdeme v tabulce 3.3 na straně 30.



Obrázek 2.9: Propojení prvků vozu

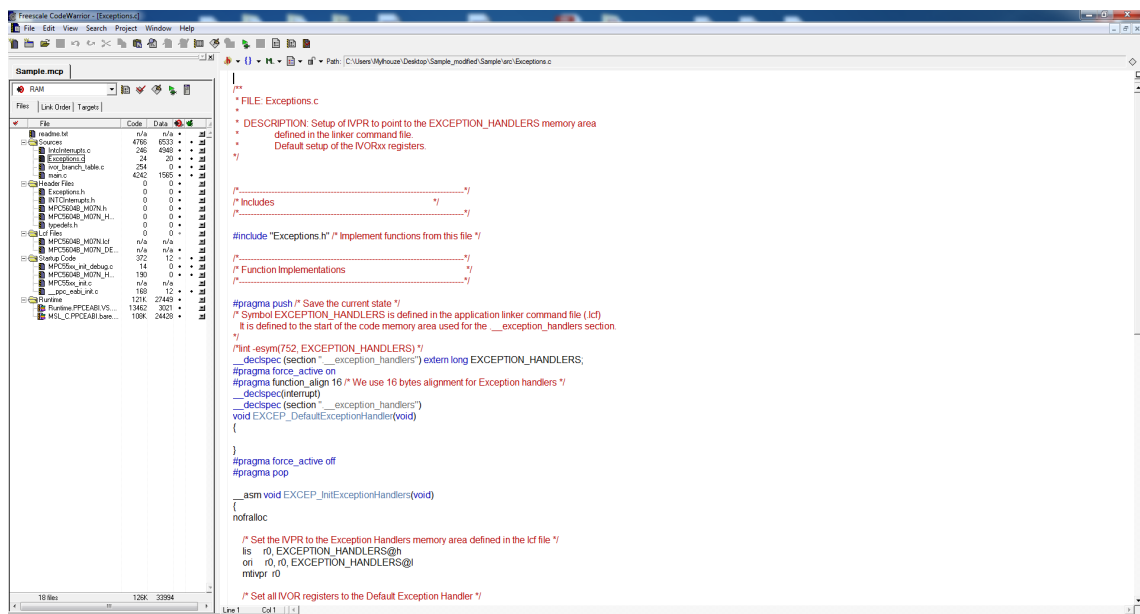
## 2.2 Software

### 2.2.1 Využitý software

Pro napsání funkčního algoritmu pro řízení modelu bylo využito několik programů. Pro napsání kódu byl použit program CodeWarrior MPC55xx/MPC56xx v2.10 (Classic), vhodný právě pro vývoj této desky. Pro odladění a testování napsaného kódu byl využit software od firmy PE micro, jedná se o celý balík programů, které se k odladění využívají. Jeden z nejdůležitějších se jmenuje Terminal Utility a slouží ke komunikaci s vývojovou deskou. Dále zde najdeme kupříkladu Logic Analyzer, Accelerometer, Serial Grapher atd. Použité programy budou zmíněny a popsány níže.

### 2.2.2 CodeWarrior

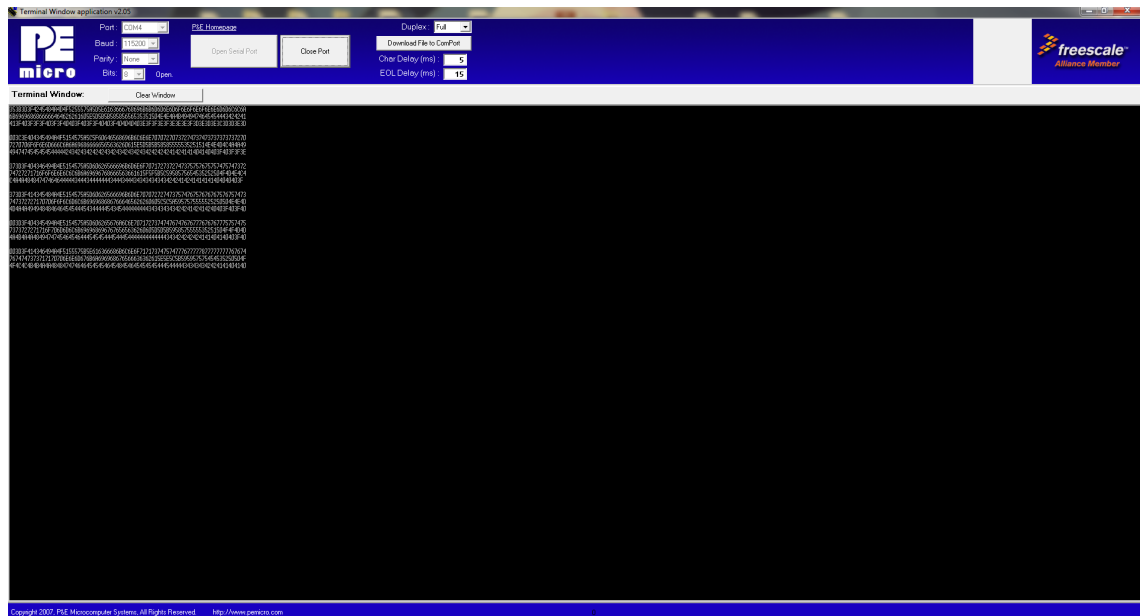
Jedná se o vývojové prostředí pro různé druhy operačních systémů. Hlavním programovacím jazykem je C/C++, ale je možné používat i jiné jazyky, jako například Pascal, Object Pascal, Objective-C či různé Java kompilátory.



Obrázek 2.10: Vývojové prostředí CodeWarrior

## 2.2.3 Terminal

Terminal od firmy PE micro byl cenným pomocníkem při ladění programu. Umožňuje komunikaci s deskou, zadávání vstupních dat a možnost výpisu dat výstupních. Jedná se o velmi intuitivní program, se kterým se dá pracovat bez větších obtíží. Potíž přichází až s 64bit verzí OS, kde se spojení ani za vydatné pomoci vedoucího práce nepodařilo zprovoznit. Proto vývoj probíhal na x86 verzi OS.



Obrázek 2.11: Terminál

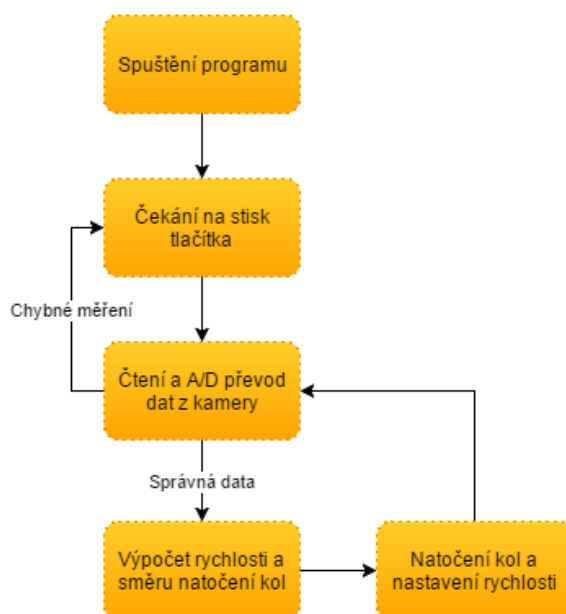
## 2.2.4 Sample Project

Při vývoji algoritmu pro desku byl stěžejní ukázkový program, jenž lze stáhnout na stránkách komunity Freescale. Jedná se o velmi popisný algoritmus, který dává jasnou představu o fungování desky. Je zde vytvořeno několik funkcí, jež nastiňují práci s komponenty desky, jako například diody LED, řízení servomotoru a také samotná práce s registry a povolování konkrétních pinů, které budeme pro náš projekt potřebovat. Je zde realizována také metoda, jež představuje práci s kamerou.

## 3 Praktická část

### 3.1 Nástin řešení

Po seznámení se s HW částmi modelu přišla na řadu programovací část. Jednalo se o vytvoření algoritmu, který bude schopen pomocí kamery sledovat a projíždět vytyčenou dráhu. Dalším úkolem byla realizace parkovacího manévru.



Obrázek 3.1: Nástin řešení

#### 3.1.1 Sledování čáry

Sledování čáry lze provádět různými způsoby. V našem případě se jedná o jednu z nejjednodušších metod, při které se snímá odraz paprsků světla a následně dojde k vyhodnocení. Výhodou tohoto řešení je jednoduchost, nevýhodou naopak to, že si naše řešení vynucuje perfektní světelné podmínky. Při snímání jsou dopadající paprsky na řádkovou kameru pomocí A/D převodníků transformovány a ukládány do struktury. V našem případě se jedná o lineární pole se 128 hodnotami. Opatřebování kamery je již značné, tudíž jsou pro naše řešení neupotřebitelné krajní hodnoty z obou stran. Také uprostřed měření se vyskytuje pár nevalidních hodnot, které ovšem nemají na fungování zásadní vliv.

### 3.1.2 Řízení motorů

Možnost generování šířky střídy nám dává schopnost měnit rychlost otáčení motorů. Toho jistě využijeme při průjezdu tratí, kdy na rovinách budou otáčky vyšší a při jízdě do oblouku nižší za účelem eliminace chybných měření a opuštění tratě. Reverzace motorů je po HW úpravách možná za splnění podmínek z tabulky 3.3. Rozsah parkovacího manévru je ovšem značně eliminován neschopností snímat plochu za vozidlem.

#### Pulzně šířková modulace

Jedná se o jev, kdy je k přenosu analogového signálu nutno použít pouze dvouhodnotový údaj. Signál je přenášen pomocí střídy. Pro demodulaci signálu se používá dolnofrekvenční propust.

#### Střída

U signálů, které během jedné periody přecházejí z jednoho stavu do druhého a naopak, znamená střída poměr časů, v nichž je signál v jednotlivých úrovních. Tudíž například střída, generovaná v poměru 1:1, udává případ, kdy se v polovině doby cyklu nachází výstup na hodnotě vrchní a zbytek cyklu na hraně spodní. Střída se často udává v %, procentuální hodnota udává na kolik % se signál nachází na úrovni 1. Tedy například střída 30 % udává, že 30 % času je na výstupu log. 1 (hodnota HIGH) a 70 % času log. 0 (LOW).

## 3.2 Možné realizace parkovacího manévru

Parkovací manévr lze provést několika způsoby. Jako nejjednodušší se jeví umístění další kamery na zadní vozidla. Tato modifikace by ovšem neumožnila vyjetí mimo stanovenou trať a algoritmizace případných traťových výhybek by byla poměrně náročná. Další možností by bylo umístění enkodérů na kola. Ty by dovolovaly snímat počet otáček kol, a tím poskytovaly informace pro další zpracování. K získání představy o pohybu vozidla by bylo nutné implementovat proces odometrie, který by nám poskytl informace pro řízení. Porovnáním informací z enkodéru by bylo zřejmé, jaké kolo se otočilo kolikrát, a tudíž v jakém úhlu se model vozidla nachází.

### 3.2.1 Enkodér

Základním principem enkodéru je převod rotačního pohybu na sekvence elektrických digitálních impulsů. Enkodéry lze rozdělit na *inkrementální* a *absolutní*. V případě inkrementálního enkodéru získáme signál obdelníkového tvaru nabývající dvou hodnot, který nám ovšem neříká nic o směru otáčení motoru. Pro případ, kdy potřebujeme kontrolu jak nad počtem otočení, tak i směrem, musíme použít enkodér absolutní. V případě absolutního enkodéru máme více stavů, než je tomu u inkrementálního a můžeme tudíž posoudit, dochází-li k inkrementaci či dekrementaci.

### 3.2.2 Odometrie

Odometrie je proces, při kterém lze ze získaných dat z enkodérů určit polohu vozu. Důležitá je znalost prostředí a počet otočení jednotlivých kol. Odometrie je založena na principu, že naměřený rotační pohyb kol je možné přepočítat na konkrétní pohyb v rovině.

## 3.3 Vlastní algoritmus

### 3.3.1 Čekací fáze

Za účelem zamezení okamžitého vykonávání algoritmu je implementovaná funkce `ready()`, která se stará jenom o rozsvícení diody, jež přímo sousedí s tlačítkem. Funkce dále čeká na stisk tohoto tlačítka. Při stlačení je model uveden do provozu. Současně dojde k deaktivaci svitu diody. Při vyhodnocení algoritmem, že se jedná o výjezd z trati, je model uveden opět do této čekací funkce.

### 3.3.2 Zpracování dat z kamery

Jeden z prvních úkolů při realizaci řídicího algoritmu bylo získání a následná úprava dat přicházejících z kamery. Data jsou čtena do pole integerů pomocí funkce `CAMERA()`, pro snadnější použití byly hodnoty v rozsahu 0 – 255 modifikovány do rozsahu 0, 1 (true/false), hodnota jedna pro značení čáry a hodnoty nula pro případ okolního (světlejšího) prostředí. Pole bylo také ořezáno o prvních a posledních deset hodnot, které vykazovaly nejspíše z důvodu opotřebení kamery známky nevalidních hodnot.

Použita byla prahovací funkce popsána níže. Algoritmus funguje tak, že spočítá nejdelší sekvenci hodnot 1 v poli, následně je vypočítán střed této sekvence a na tento index nastavena proměnná pro natočení kol. V případě, že se tento index nachází okolo 30 % či 70 % rozsahu pole, je proměnná pro natočení nastavena na úplně vlevo/vpravo. Při více položkách s hodnotou 1 než je polovina rozsahu pole je proměnná pro natočení kol nastavena do středu. Toto se může stát v případě, že vůz projíždí startovní čarou, která je položena po šířce celé tratě či v případě křížení tratí.

Algoritmus je psán ještě pro starší typ dráhy, kdy byla vodící čára umístěna uprostřed tratě. V současnosti je trať ohraničena čarami z obou stran, obdobně jako je tomu v případě silnic pouze s krajnicemi bez středového pruhu. Takto bylo realizováno z důvodu toho, že veškeré vyrobené testovací tratě jsou ještě přizpůsobeny starším podmínkám. Pro případ výjezdu modelu z tratě, je implementována podmínka, která v případě, že se počet kladných hodnot symbolizujících čáru dostane pod určenou mez, zastaví vozidlo.

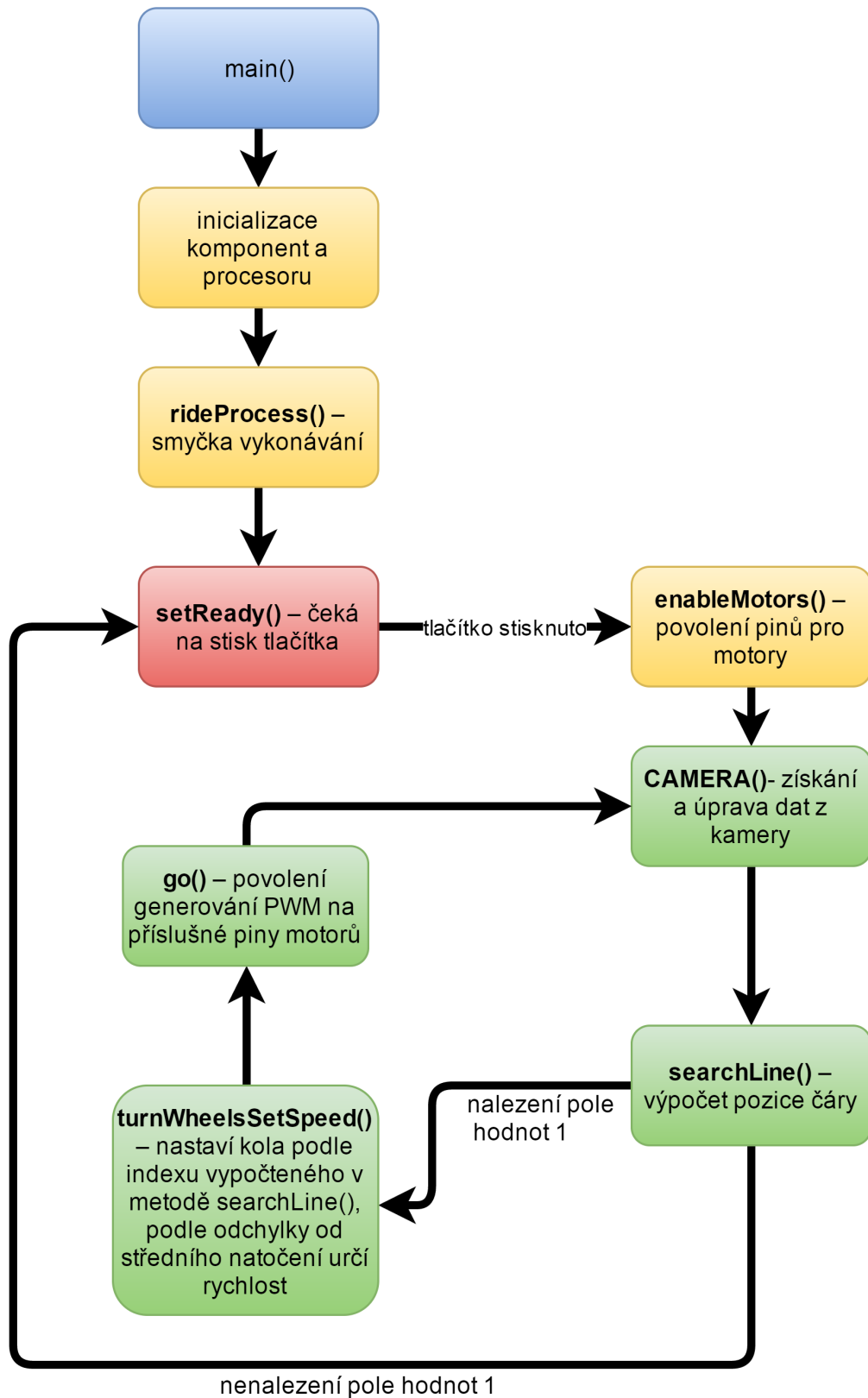


### 3.3.3 Rychlost

Rychlost otáčení motorů je definována pomocí střidy na odpovídajícím výstupu desky. Nastavení rychlosti otáčení se řídí podle dat z kamery. Vypočtena je odchylka od střední hodnoty natočení kol. Rychlost má tři stupně. Při malé odchylce je rychlost vysoká, s přibývajícím odchýlením je snižována. V případě, že jsou hodnoty od kamery vyhodnoceny jako vyjetí z tratě, je algoritmus uveden do stavu čekání.

### 3.3.4 Prahovací funkce

Podle světelných podmínek a schopností kamery lze nastavit prahovou hodnotu, pomocí které je rozeznávána čára od okolního prostředí. Při standardních světelných podmínkách se jevila jako ideální prahová hodnota 250. Nad hodnotu 250 vykázána hodnota true (1) symbolizující čáru, v opačném případě hodnota false (0) pro okolní prostředí. Bylo by také možné použít adaptivní nastavení prahu, kdy by byl nasnímán nejjasnější a nejtemnější bod a byl vypočten průměr těchto hodnot. Od tohoto řešení bylo ale upuštěno a jako vhodná se ukázala hranice 250.



Obrázek 3.2: Diagram navrhnutého algoritmu

## 3.4 HW úpravy

Na vozidle byly provedeny HW úpravy pro možnost reverzace motorů. Provedenou úpravu je možné vidět na obrázku 3.4. Jedná se o spojení mezi výkonovou a řídicí deskou. Jde o dvou vodičový kabel, který je připájen na nožičky H-můstků obou motorů. Na druhé straně lze kabel zasunout na vhodné piny podporující PWM, nezbytnou pro řízení reverzovaného chodu motorů. Více viz. [15].

Device State	Input Conditions				Status	Outputs	
	$EN/\overline{D2}$	D1	IN1	IN2	$\overline{SF}$	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 $EN/\overline{D2}$	L	X	X	X	H	Z	Z
$EN/\overline{D2}$ Disconnected	Z	X	X	X	H	Z	Z

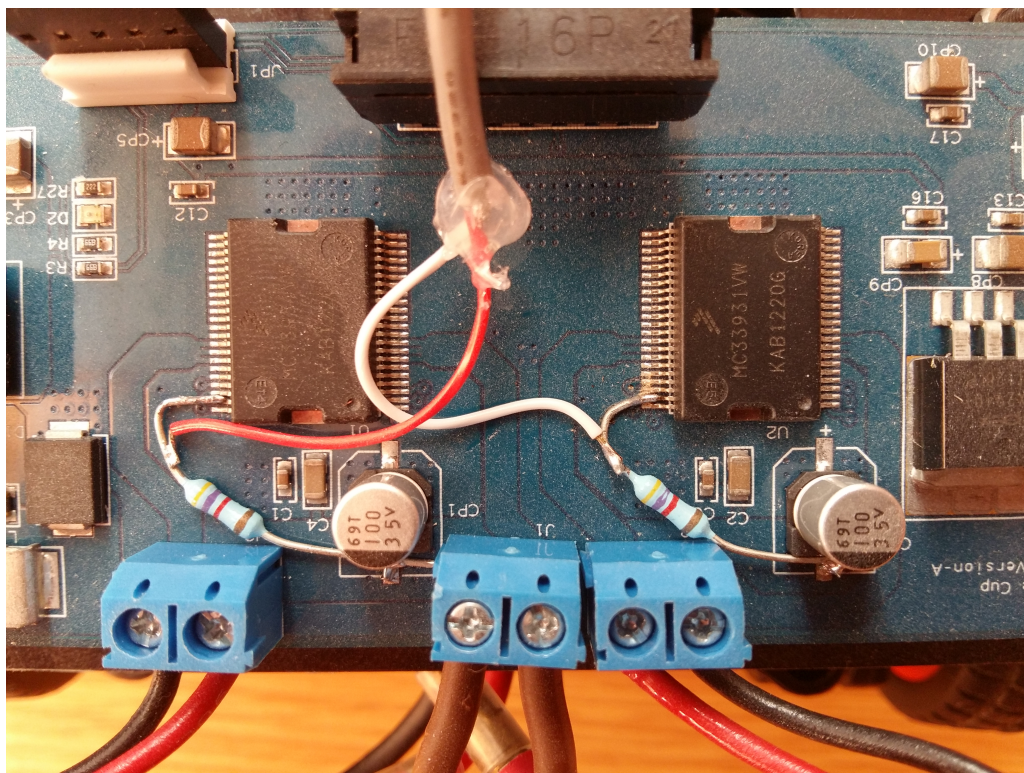
Obrázek 3.3: Pravdivostní tabulka [15]

### 3.4.1 Vnější úprava

Podle tabulky 3.3 bylo třeba pro zpětný chod zajistit možnost řízení vstupujícího signálu IN1 na H-můstcích motorů. Za tímto účelem byl přiveden vodič z řídicí desky. Ve výchozím stavu byl takový, že vstup IN1 byl trvale připojen na GND. Bylo tudíž nutné toto spojení přerušit. Kvůli eliminaci možnosti, že by došlo k ulomení či odtavení nožičky na H-můstku, nebyla odpájena, bylo pouze provedeno přerušení spoje mezi nožičkou a GND.

### 3.4.2 Softwarové testování provedených úprav

Za účelem řízení vybraného pinu na portech řídicí desky bylo třeba zjistit, jaké piny podporují periférii EMIOS pro řízení PWM. Na řídicí desce se nachází několik portů, každý, krom portu H, po 16 pinech. Na portu H se nachází pouze 12 pinů. Porty jsou v rozsahu A-H, proto se zde nachází 124 pinů pro I/O operace. Podporované funkce na každém pinu jsou popsány v datasheetu desky [3]. Jako optimální se zdálo použití pinů, podporujících EMIOS\_0, které byly již nakonfigurovány pro řízení modelu směrem vpřed. Bylo nutné dokonfigurovat kanály 16 a 22, jež odpovídají pinům 64 a 70, na něž bylo spojení s H-můstky napojeno. Konfigurace kanálů byla definována obdobně jako pro kanály 6 a 7, které se starají o vytváření střídy pro pohyb vpřed. Zde následuje ukázková deklarace kanálu č. 16. obdobně je definován a deklarován kanál č. 22. Pro správnou rotaci kol je nutné posílat PWM pouze na IN1 nebo IN2. Tudíž je nutné selektivní povolování výstupů pinů na desce.



Obrázek 3.4: HW úpravy vozidla

## 3.5 Problémy při realizaci

### 3.5.1 32bit OS

Jedním z prvních problémů, který se během vývoje algoritmu vyskytl, byla nutnost použití 32-bitového operačního systému. Terminál i program CodeWarrior odmítal pod 64-bitovým systémem pracovat. Bylo tudíž nutné zajistit operační systém s architekturou x86. Jelikož ani Windows 10 x86 nechtěl s programy spolupracovat, byl nutný návrat k systému Windows 7, jenž se při návrhu algoritmu osvědčil.

### 3.5.2 Chybné měření kamery

Algoritmus získaný ze Sample projektu, který ukládá do pole o rozsahu 128 jednotek hodnoty 0 – 255, vykazoval při testování nevaliditu nasnímaných hodnot na začátku a konci pole. Bylo prokázáno, že hodnoty jsou chybně interpretovány již na čipu kamery. Pole bylo proto ořezáno o deset hodnot zleva a deset zprava, jelikož bylo vypořazováno, že tyto hodnoty vykazují nekonzistenci s nasnímaným prostředím. Měření po tomto kroku začalo vykazovat znaky validity. Nevýhodou je ovšem to, že jsme tak přišli o krajní hodnoty pole, které jsou pro vykonávání algoritmu důležité a umožňují včasnou reakci na nástup čáry do zorného pole kamery. Z tohoto důvodu bylo nutné snížit rychlost pohybu vozidla, aby bylo možné i bez krajních hodnot dostat validní a včasné zásahy do řízení.

### 3.5.3 Funkce Delay()

Po napsání algoritmu a jeho následném testování bylo odhaleno, že program není vykonáván v takové frekvenci, jakou bychom očekávali. To bylo způsobeno tím, že se v kódu nacházelo pár zapomenutých volání funkcí Delay(). Tato funkce při každém zavolání způsobí zpoždění okolo několika milisekund. Nejedná se o zásadní časový úsek, ovšem v našem případě okamžité potřeby vyhodnocení pozice čáry, je každé prodlení na správném fungování znát.

### 3.5.4 Vybitá baterie

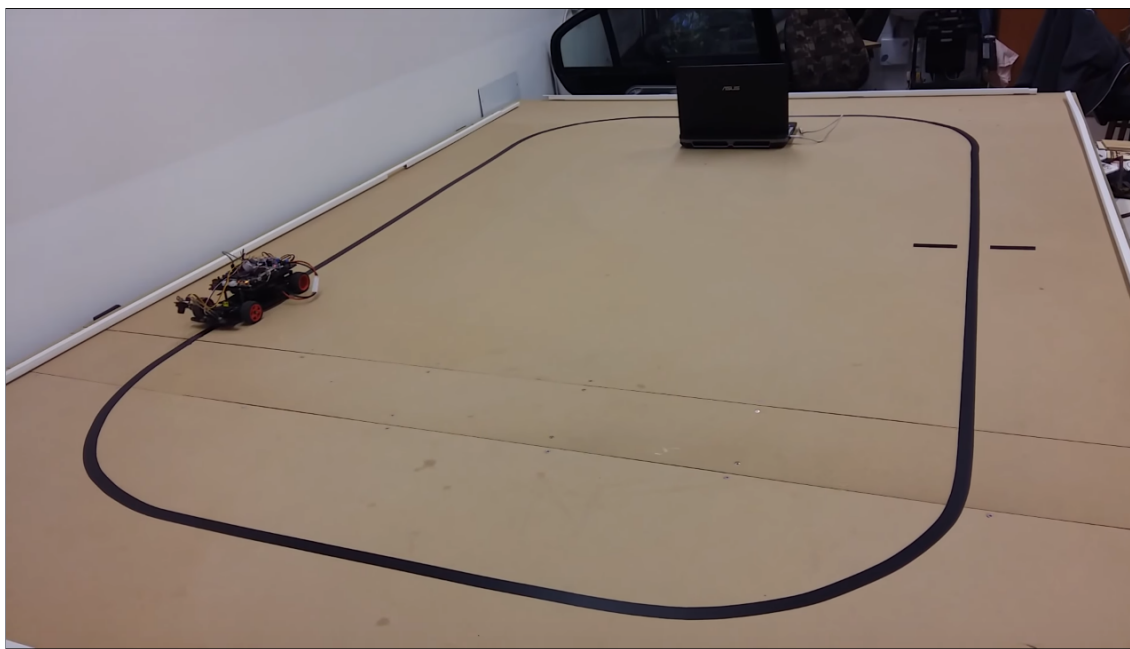
Za tohoto stavu dochází k nepředvídatelnému chování kitu. Problém je jak při připojení k PC, tak při samotném vykonávání programu. Jelikož se na modelu nenachází žádná signalizace stavu nabití baterie, je tento stav jen těžko odlišitelný on případné chyby v algoritmu. Je proto nutné stav nabití baterie pravidelně kontrolovat multimetrem.

## 3.6 Testování

Testování probíhalo nejdříve na zkušebním kousku tratě, který byl realizován pouze pro případ seznámení se s vykonáváním, a správným snímáním kamery. Pro již částečně otestovaný algoritmus byla dále připravená trať vytvořená v laboratoři Technické univerzity v Liberci. Dráha, která je vytvořena na desce stolu za pomoci černé pásky na světlém podkladu. Jedná se o standardní ovál se startovní čarou.

Testy navrhnutého algoritmu probíhaly s nejvyšší obezřetností, aby nedošlo k pádu a poškození vozidla. Trať se podařilo projet s časem pod 20 vteřin bez kolize. Kolem tohoto výsledku se pohybovaly hodnoty, zaznamenané v bakalářských pracích z předešlých let. Testovány byly i větší rychlosti, ovšem omezené snímání kamery, které neposkytuje validní krajní hodnoty, mělo za následek několik kolizí. S ohledem na vyvýšený stůl bylo tudíž od větších rychlostí upuštěno. Byla též otestována možnost reverzace motorů.

Byly testovány jenom základní možnosti couvání. Pro parkovací manévr by byla nutná modifikace tratě a modelu samotného, jenž by musel snímat plochu nejen před, ale i za vozidlem. Algoritmizace takto navrhnutého parkovacího manévru by nebyla příliš složitá, algoritmus vyhodnocování středu čáry by byl dost podobný inverznímu algoritmu směru vpřed. Bylo takto odzkoušeno chvilkovým odpojením kamery od rámu modelu a provizorním připojením na zadní část, aby byl snímán povrch nikoli před, ale za vozidlem.



Obrázek 3.5: Trať v laboratořích TUL

## Závěr

Mnou realizované zadání bakalářské práce se ukázalo jako velice přínosné a zajímavé. Seznámil jsem se za toto období s inicializací a principy fungování mikropočítače, se kterým jsem v dosavadním studiu nebyl příliš mnoho kontaktu. Zajímavým prvkem bylo také vyvíjení řídicího algoritmu, při němž byla osvěžena znalost programovacího jazyka C/C++. Jako většina prací se bohužel ani tato neobešla bez počátečních problémů, které byly naštěstí brzo překonány.

Testování algoritmu na trati dopadlo vcelku podle mých očekávání. Čas, kterého jsem dosáhl, nebyl z těch nejúchvatnějších, ale myslím si, že jsem našel rozumnou hranici mezi rychlostí a rizikem poničení modelu.

Testování reverzace motorů mohlo jistě proběhnout ve větší míře, avšak jsem rád, že se mi přes časový tlak podařilo prokousat datasheety desky a potvrdit správné provedení realizovaných úprav a jejich částečné otestování.

Byl demonstrován částečný parkovací manévry, ovšem jenom po provizorních úpravách modelu. Kamera byla přemístěna za vozidlo a bylo tím umožněno snímat plochu za modelem. Pro složitější parkovací manévry by bylo nutno opatřit model zadní kamerou v případě, že bychom chtěli parkovací manévry realizovat podle vodící čáry. V případě vyšších požadavků na parkování by bylo nutné model opatřit enkodéry a realizovat proces odometrie viz. 3.2.

Touto prací jsem získal mnoho cenných zkušeností, kterých doufám budu moci v budoucnu využít. Ať už se jedná o můj zájem v inicializaci mikropočítačů, či programování v nízkourovňových jazycích.



## Literatura

- [1] PLÍVA, Z., J. DRÁBKOVÁ, J. KOPRNICKÝ a L. PETRŽÍLKA. *Metodika zpracování bakalářských a diplomových prací* 2. upravené vydání. Liberec: Technická univerzita v Liberci, FM, 2014. ISBN 978-80-7494-049-1. Dostupné z: doi:10.15240/tul/002/2014-11-002
- [2] *Pulzně šířková modulace*. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-05]. Dostupné z: [https://cs.wikipedia.org/wiki/Pulzn%C4%9B\\_%C5%A1%C3%AD%C5%99kov%C3%A1\\_modulace](https://cs.wikipedia.org/wiki/Pulzn%C4%9B_%C5%A1%C3%AD%C5%99kov%C3%A1_modulace)
- [3] MPC5604B/C Microcontroller Reference Manual. *MPC5604B/C Microcontroller Reference Manual* [online]. Freescale Semiconductor, 2013 [cit. 2016-05-05]. Dostupné z: [http://cache.freescale.com/files/32bit/doc/ref\\_manual/MPC5604BCRM.pdf](http://cache.freescale.com/files/32bit/doc/ref_manual/MPC5604BCRM.pdf)
- [4] Pravidla EMEA 2016. TFC 2016 *EMEA Challenge Rules 01 June 2015 — NXP Community*[online]. Freescale Semiconductor, 2013 [cit. 2016-05-05]. Dostupné z: <https://community.freescale.com/docs/DOC-105423>
- [5] Sample project *Codewarrior Sample project for MPC55xxB/C and MPC56xxB/C* [online]. [cit. 2016-05-06]. Dostupné z: <https://community.freescale.com/docs/DOC-1019>
- [6] Tvorba diagramů - draw. *Draw.io is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams* [online]. [cit. 2016-05-06]. Dostupné z: <https://www.draw.io/>
- [7] Použité ikon. *Icon Archive* [online]. [cit. 2016-05-06]. Dostupné z: <http://www.iconarchive.com>
- [8] Statické a dynamické paměti. *Root.cz* [online]. [cit. 2016-05-06] Dostupné z: <http://www.root.cz/clanky/staticke-a-dynamicke-pameti/>
- [9] LIN – Local Interconnect Network *Automatizace.hw.cz: rady a poslední novinky z oboru* [online]. 2004 [cit. 2016-05-11]. Dostupné z: <http://automatizace.hw.cz/clanek/2005101501>



- [10] The NXP Cup EMEA. *NXP* [online]. [cit. 2016-05-15]. Dostupné z: <https://community.freescale.com/groups/tfc-emea>
- [11] Enkodéry. *Robotika.cz* [online]. [cit. 2016-05-15]. Dostupné z: <https://robotika.cz/guide/encoders/cs>
- [12] Odometry. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-15]. Dostupné z: <https://en.wikipedia.org/wiki/Odometry>
- [13] TFC-KIT: The NXP Cup Intelligent Car Development System Favorite. *NXP* [online]. [cit. 2016-05-15]. Dostupné z: <http://www.nxp.com/products/microcontrollers-and-processors/power-architecture-processors/mpc5xxx-5xxx-32-bit-mcus/mpc56xx-mcus/the-freescale-cup-intelligent-car-development-system:TFC-KIT>
- [14] RS-232. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-15]. Dostupné z: <https://cs.wikipedia.org/wiki/RS-232>
- [15] Freescale Cup Motor Driver Board. *NXP* [online]. [cit. 2016-05-15]. Dostupné z: <https://community.freescale.com/docs/DOC-1059>

# Přílohy

## A Obsah přiloženého CD

1. Bakalářská práce(BP.pdf)
2. Vývojový algoritmus(BPA.zip)
3. Datasheet řídicí desky(MPC5604BC.pdf)