



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**NÁVRH ŘÍDICÍ JEDNOTKY PRO
ROBOTICKÉ VOZIDLO CAR4**

CONTROL UNIT DESIGN FOR ROBOTIC VEHICLE CAR4

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Šustek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Barnabás Dobossy

BRNO 2020

Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Jan Šustek
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	Ing. Barnabás Dobossy
Akademický rok:	2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh řídicí jednotky pro robotické vozidlo Car4

Stručná charakteristika problematiky úkolu:

Projekt Car4 vznikl v roce 2010 v rámci několik závěrečných prací. Od té doby se na něm realizovaly další práce, v rámci kterých byl aktualizován software na řídicích jednotkách, vozidlo bylo rozšířeno o dálkový ovladač a o jednotku sloužící k monitorování stavu nabití baterie. Tato práce je jedna ze dvou prací, jejichž cílem je obnovení a sjednocení původního hardwaru, který se od začátku projektu neměnil. Vzniknou tak podmínky pro další rozvoj projektu, který by se mohl uskutečnit v následujících letech. Práce se bude zabývat návrhem řídicí elektroniky pro robotické vozidlo Car4. Nová řídicí jednotka by měla být schopna obsluhovat enkodéry, výkonové měniče pro motory a servomotory ovládající natočení kol a dále komunikovat s nadřazenými systémy. Součástí práce je i přizpůsobení řídicího algoritmu, který vznikl v rámci dřívějších prací.

Cíle bakalářské práce:

1. Představte základní komunikační sběrnice, které jsou potřebné k obsluze standardních senzorů (UART, I2C, SPI) a běžné prvky komunikačních protokolů.
2. Na základě výchozích požadavků navrhnete novou řídicí jednotku pro vozidlo Car4 založenou na mikrokontroleru dsPIC. Nová řídicí jednotka by měla umožnit komunikaci s následujícími zařízeními:
 - enkodéry,
 - výkonová elektronika,
 - servomotory,
 - dálkový ovladač,
 - PC.
3. Otestujte a implementujte novou řídicí jednotku na vozidlo Car4.
4. Na základě předchozích prací navrhnete bezpečný komunikační protokol pro komunikaci s dálkovým ovládáním a PC pomocí sběrnice UART.
5. Přizpůsobte původní řídicí algoritmus k novým hardwarovým prvkům. Navrhnete regulátor rychlostí kol.

Seznam doporučené literatury:

VALÁŠEK, Michael. Mechatronika. Vydavatelství ČVUT, 1995. ISBN 80-01-01276-X.

VEJLUPEK, Josef. Vývoj elektroniky pro řízení trakce experimentálního vozidla. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2010.

VADLEJCH, Filip. Konstrukce podvozku experimentálního vozidla se čtyřmi řízenými koly. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2009.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato bakalářská práce se zabývá vytvořením nové řídicí jednotky pro experimentální vozidlo CAR4. V první části práci jsou popsány základní komunikační protokoly a algoritmy kontrolních součtů, které se u komunikace používají. V druhé části práce je navrženo elektronické schéma řídicí jednotky, navržena deska plošných spojů a optimalizován program. Je zde také vytvořena komunikace mezi řídicí deskou a počítačem a regulace otáček stejnosměrného motoru.

ABSTRACT

This bachelor thesis deals with creation of new control unit for experimental vehicle CAR4. In the first part there are described basic communication protocols and checksum algorithms. In the second part of the work, the electronic diagram of the new control unit and the printed circuit board are designed and control algorithm is optimized. Also there is established communication between control unit and computer and creation of DC motor speed regulation.

KLÍČOVÁ SLOVA

Car4, UART, I2C, SPI, Řídicí jednotka, Komunikační protokoly, Matlab/Simulink, Kontrolní součet, Stejnosměrný motor

KEYWORDS

Car4, UART, I2C, SPI, Control unit, Communication protocols, Matlab/Simulink, Checksum, DC motor

BIBLIOGRAFICKÁ CITACE

ŠŮSTEK, Jan. *Návrh řídicí jednotky pro robotické vozidlo Car4*. Brno, 2020. 37 s. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/124875>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Barnabás Dobossy.

PODĚKOVÁNÍ

Chtěl bych poděkovat panu Ing. Barnabáši Dobossymu za velmi ochotný přístup, za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Barnabáše Dobosyho a s použitím literatury uvedené v seznamu literatury.

V Brně dne 26. 6. 2020

.....

Jan Šústek

OBSAH

1	ÚVOD.....	10
2	REŠERŠNÍ ČÁST.....	11
2.1	Základní komunikační sběrnice.....	11
2.1.1	UART.....	12
2.1.2	SPI.....	13
2.1.3	I ² C.....	14
2.2	Algoritmus kontrolního součtu.....	16
2.2.1	Kontrolní součet Fletcher.....	17
2.2.2	CRC.....	17
3	EXPERIMENTÁLNÍ VOZIDLO CAR4.....	19
3.1	Aktuální stav vozidla.....	19
3.2	Důvod nového řešení.....	19
4	NÁVRH ŘÍDICÍ JEDNOTKY.....	20
4.1	Požadavky.....	20
4.2	Zvolené řešení.....	20
4.3	Nová řídicí jednotka.....	23
5	ŘÍDICÍ ALGORITMUS.....	25
5.1	Aktualizace řídicího programu.....	25
5.1.1	Konfigurace systému.....	25
5.1.2	Komunikace s dálkovým ovládním.....	26
5.1.3	Data z enkodérů.....	27
5.1.4	Řízení motorů a servopohonů.....	27
5.2	Komunikace s PC.....	28
5.3	Regulátor rychlosti DC motoru.....	30
6	ZÁVĚR.....	31
7	SEZNAM POUŽITÉ LITERATURY.....	32
8	SEZNAM ZKRATEK, OBRÁZKŮ A TABULEK.....	35
8.1	Seznam zkratk.....	35
8.2	Seznam obrázků.....	36
8.3	Seznam tabulek.....	36
9	SEZNAM PŘÍLOH.....	37

1 ÚVOD

Automobilový průmysl byl, je a pravděpodobně vždy bude jedním z nejdůležitějších technologických odvětví. Do vývoje automobilů se investují velké peníze a společnosti se předhánějí v nových technologiích a systémech, které jejich stroje obsahují. V posledních letech je kladen důraz zejména na vývoj hybridních automobilů a elektromobilů a většina výrobců začíná nabízet své vozy i ve verzi s elektromotorem.

S tímhle narůstajícím trendem souvisí i rozvoj elektroniky a softwaru ve vozidle. Nové systémy mají zlepšovat jak bezpečnost, tak komfort cestujících. Příkladem nových technologií může být autopilot, automatické parkování, adaptivní tempomat, detekce chodců nebo systém sledování únavy řidiče.

Z auta se stává dokonalý příklad mechatronického výrobku, kdy spolu musí dokonale kooperovat mechanická složka, elektronická složka a software. I z tohoto důvodu vzniklo v roce 2010 v Mechatronické laboratoři VUT Brno experimentální vozidlo Car4, které je určeno k testování a vývoji systémů, se kterými se můžeme setkat v osobních automobilech. [1] Tomuto projektu se během posledních 10 let věnovalo velké množství bakalářských a diplomových prací, například v roce 2013 vznikl antikolizní systém [2], v roce 2015 systém na lokalizaci vozidla [3] a úprava dálkového ovládání [4], v roce 2016 systém automatického parkování [5] a v roce 2017 jednotka pro monitorování stavu baterie [6].

Tato práce má za cíl vytvořit řídicí jednotku, která bude schopná obsluhovat zařízení, jako jsou enkodéry, výkonové měniče pro motory, servomotory ovládající natočení kol a dálkové ovládání. Tato jednotka bude tvořit nízkouúrovňové ovládání vozidla a bude komunikovat s nadřazenými řídicími systémy. Společně s touto prací vzniká druhá bakalářská práce, která se zabývá návrhem výkonové elektroniky pro toto vozidlo. Společně mají tyto práce za cíl obnovení elektroniky vozidla, tak aby vznikly podmínky pro další rozvoj tohoto vozidla.

V první kapitole této práce budou popsány nejčastěji používané komunikační sběrnice, potřebné k obsluze jednotlivých částí a komunikaci mezi nimi a mikroprocesorem. V závěru rešeršní části jsou uvedeny algoritmy kontrolních součtů, příklady a důvody, proč se využívají.

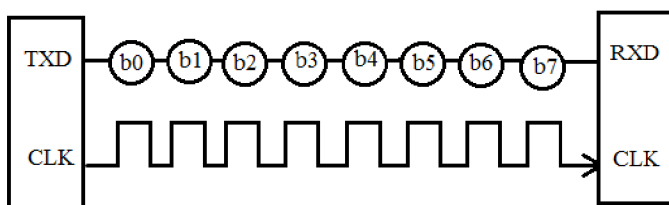
Hlavní část se zabývá návrhem a realizací řídicí jednotky a řídicím algoritmem. Nejdříve se podíváme na výsledné provedení řídicí desky a na její jednotlivé části. V dalších kapitolách je optimalizován program, který byl převzat z předchozích prací. Poté je vytvořena bezpečná komunikace mezi řídicí deskou a PC. V poslední části práce je navržena regulace stejnosměrného motoru pomocí řídicí desky.

2 REŠERŠNÍ ČÁST

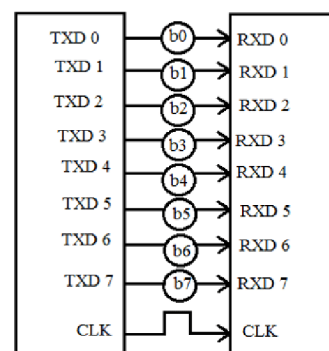
2.1 Základní komunikační sběrnice

Pro komunikaci mezi mikrokontroléry nebo mezi mikrokontrolérem a připojeným zařízením musíme vždy vybrat vhodnou sběrnici. Každá sběrnice má určeny fyzikální, signálové i funkční charakteristiky propojení. Sběrnice můžeme rozdělit na dva typy podle způsobu posílání dat:

- Sériové komunikační protokoly - jednotlivé bity jsou přenášeny po jednom datovém vodiči za sebou. Příkladem sériové komunikace jsou protokoly CAN, ETHERNET, USB, 1-Wire, SATA, I2C, SPI nebo UART. U spojení mikrokontroléru a senzorů se využívá především tohoto způsobu.
- Paralelní komunikační protokoly – bity jsou posílány současně, každý bit po samostatném vodiči. Do této kategorie patří protokoly ISA, ATA, SCSI nebo PCI.



Obrázek 1 - Sériová komunikace [7]



Obrázek 2 - Paralelní komunikace [7]

Další rozdělení jednotlivých sběrnic je podle možnosti čtení a zapisování dat:

- Simplex (jednosměrná) komunikace – jedno zařízení může data jen odesílat a druhé zařízení může data jen přijímat.
- Half Duplex (poloviční obousměrná) komunikace – komunikace v jednu chvíli probíhá jen jedním směrem, tedy ve směru od řídicího zařízení k připojenému zařízení, nebo opačně.
- Full Duplex (obousměrná) komunikace – zařízení mohou data v jednu chvíli vysílat i přijímat.

Podle synchronizace hodinového signálu dělíme sběrnice na:

- Synchronní - zařízení navzájem sdílí společný hodinový signál, který synchronizuje příjem a vysílání dat.
- Asynchronní - komunikace, kdy přenos dat není řízen společným hodinovým signálem a zařízení musí řídit svoji komunikaci jiným způsobem. [7]

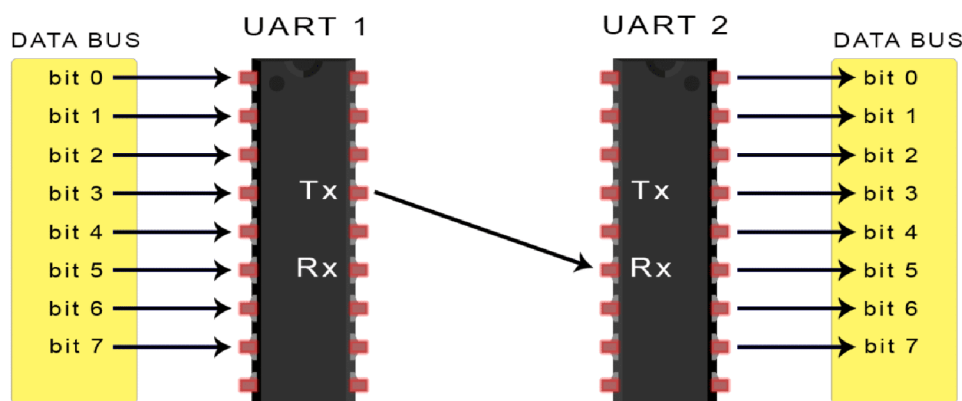
2.1.1 UART

UART je zkratka pro “Universal asynchronous receiver/transmitter“, neboli “Univerzální asynchronní přijímač/vysílač. UART obecně definuje chování komunikačního logického obvodu a jednotlivé protokoly určují dodatečné informace, jako jsou napěťové úrovně, typ konektoru nebo názvy signálů. Těmito protokoly jsou např. RS-232, RS-422 nebo RS-485. [8]

UART je asynchronní full-duplex komunikace. Způsob, který využívá UART pro řízení komunikace, je takový, že používá tzv. start-bit a stop-bit. Tyto bity se vloží před a za vysílaná data a předávají přijímacímu zařízení informaci o tom, kdy začíná a končí přenos dat. U komunikace také definujeme klidový stav, který je většinou v logické 1 a dává informaci zařízením, že neprobíhá žádný přenos. Start-bit překlápí stav do logické 0 a upozorní tím přijímací stranu, že začíná přenos. Naopak stop-bit překlápí přenos do klidového stavu, tj. do logické 1, trvá většinou po dobu dvou bitů a ukončuje komunikaci.

Komunikaci, jako je UART, se říká tzv. point-to-point komunikace, protože si spolu vyměňují informace dvě zařízení. Samotný přenos dat probíhá po dvou linkách, které musí obsahovat obě zařízení. Jsou to linky Tx (vysílací část) a Rx (přijímací část). Obě zařízení musí mít nastavenou přenosovou rychlost (baud-rate), která je u obou zařízení podobná. Tato hodnota určuje frekvenci posílaných dat v bitech za sekundu.

Samotný proces funguje tak, že mikrokontrolér nebo procesor pošle informace v paralelní podobě a tyto informace UART na straně vysílacího zařízení převede do sériové podoby. Následně začíná přenos start-bitem, který převede klidový stav do logické nuly, poté se po jednom bitu přeposílají data druhému zařízení. Po odeslání posledního bitu se může přidat tzv. paritní bit, který slouží ke zkontrolování správnosti přenosu. Paritní bit udává, zda součet daného přenosu je sudý nebo lichý. Pokud se na straně přijímacího zařízení zjistí, že paritní bit má mít jinou hodnotu, je to znamení, že při přenosu nastala chyba. Po paritním bitu se přidá stop-bit, který většinou má délku 1,5–2 bitů. Přijímací UART odebere z přijatých dat start-bit, paritní bit a stop-bit a převede data zpět do paralelní podoby a dále s nimi pracuje. [9]



Obrázek 3 - UART komunikace [9]

2.1.2 SPI

SPI je zkráceně “Serial Peripheral Interface“, tj. “Sériové periferní rozhraní“. Toto rozhraní bylo vytvořeno společností Motorola v 80. letech 20. století. Patří do skupiny nejpoužívanějších forem komunikace, a proto je využíváno mnoha zařízeními, jako jsou senzory, dotykové displeje nebo různé typy pamětí.

SPI je synchronní full-duplex komunikace. Velkou výhodou této komunikace je přenosová rychlost, protože je komunikace synchronní, frekvence se pohybuje v jednotkách MHz, v krajních případech až do 70 MHz. [10]

V případě komunikace UART jsme měli možnost vzájemné komunikace pouze mezi dvěma zařízeními. Rozhraní SPI nám umožňuje komunikaci mezi více uzly. Tato komunikace je typu master-slave, tudíž máme jedno řídicí zařízení, tzv. master, a k němu připojeno teoreticky libovolné množství zařízení, zvaných slave.

V zařízeních, která využívají toto rozhraní, najdeme nejméně čtyři vodiče, které realizují komunikace. Tyto vodiče jsou nazvány:

Označení	Význam
MOSI	Master Output/Slave Input
MISO	Master Input/Slave Output
SCLK nebo SCK	Serial Clock
SS, CS nebo SSEL	Slave select

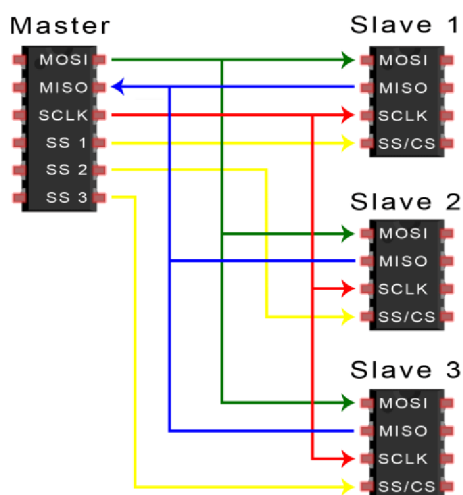
Tabulka 1 - SPI linky

Nejdříve řídicí zařízení přes výstup SCK vyšle hodinový signál do všech připojených zařízení. S hodinovým signálem souvisí konfigurace, kterou si můžeme nastavit. Můžeme si vybrat polaritu hodinového signálu, tím nastavíme jaká úroveň je brána jako klidová, tudíž neprobíhá posílání dat. Dále si můžeme nastavit, jestli chceme číst data při sestupné nebo vzestupné hraně hodinového signálu.

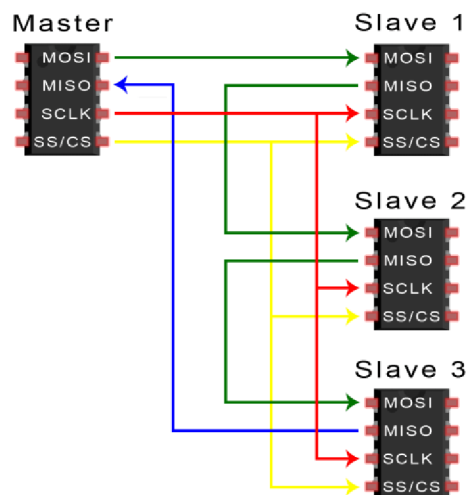
Připojení zařízení k řídicímu členu můžeme realizovat dvěma způsoby:

- První způsob je ten, že pro každé zařízení je vyhrazena z řídicího členu jedna linka “Chip Select“. Tyto výstupy jsou nastaveny na horní úroveň napětí, a když si řídicí člen vybere zařízení, se kterým chce komunikovat, tak konkrétní linku přepne na nízké napětí. Výhoda tohoto způsobu je zejména v rychlosti komunikace, naopak nevýhoda se projeví při větším počtu zařízení, kdy pro každé připojené zařízení musíme vyhradit jednu linku.
- Druhou možností zapojení je tzv. daisy-chain, neboli propojit všechny zařízení do jedné smyčky, kdy data poslaná z řídicího členu projdou všemi zařízeními a vrátí se zpět do hlavního členu. Protože data prochází přes všechna zařízení, tak abychom určili, která data přichází od kterého zařízení, musíme vytvořit ještě doplňující protokol, který určí náležitosti komunikace mezi jednotlivými členy.

Data jsou posílána po lince MOSI – z řídicího členu k připojeným zařízením a lince MISO – data od připojených zařízení do řídicího členu. Při komunikaci pomocí SPI jsou data posílána nepřetržitě bez přerušení, jako je to u jiných typů komunikace. Výhodou je, že data mohou být posílána s vysokou frekvencí a nemusí se řešit adresování zprávy. Nevýhodou je zejména to, že u této komunikace není kontrolována správnost zprávy pomocí doplňujících bitů, jako tomu bylo u komunikace pomocí UART. [11]



Obrázek 5 - SPI komunikace - Zapojení každého zařízení zvlášť [11]



Obrázek 4 - SPI komunikace - Zapojení do smyčky [11]

2.1.3 I²C

Poslední představenou sběrnici bude I²C, tedy “Inter-Integrated Circuit“. Tato sběrnice byla vytvořena firmou Philips v roce 1982. [12] U některých zařízeních se můžete setkat s označením TWI (“Two Wire Serial Interface“), které využívají někteří výrobci. [13]

Komunikace u I²C je typu master-slave, se kterým jsme se setkali i u popisu sběrnice SPI. Oproti SPI je zde jednodušší implementovat více řídicích zařízení typu master. I²C je poloduplexní synchronní komunikace. Rychlost komunikace dosahuje až 5 MHz, ale ve většině případů se využívá rychlosti 100 kHz nebo 400 kHz. Každé zařízení, které využívá sběrnici I²C, má od výroby předem danou adresu. Tato adresa je nastavena výrobcem. U některých zařízeních se můžeme setkat s tím, že výrobce přidal vývody, které po připojení k napájecímu napětí nebo zemi změni adresu zařízení.

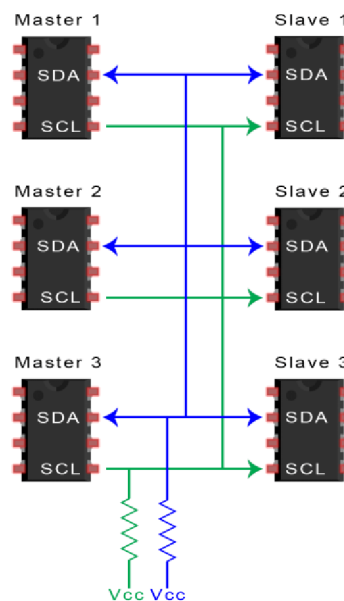
Pro vytvoření I²C sběrnice potřebujeme linky:

Označení	Význam
SDA	Serial Data Line
SCL	Serial Clock Line

Tabulka 2 - I²C linky

Na tyto dvě linky jsou připojena všechna zařízení. Dále se obě linky musí připojit přes pull-up rezistory na napájecí napětí, což zaručí, že na nich v klidovém stavu bude horní

úroveň napětí, tedy logická 1. V některých situacích se přidává i společná signálová zem.



Obrázek 6 - I2C komunikace [14]

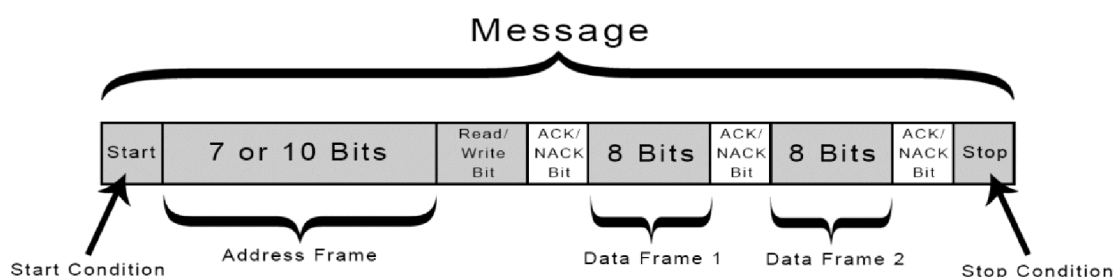
Přenos mezi zařízeními začíná vždy vysláním tzv. start-bitu. Tento bit se vyšle tak, že linku SCL necháme v logické 1 a linku SDA přepneme do nuly. Tímto dáme najevo, že začíná přenos. Hodinový signál vždy vysílá řídicí zařízení (master). Pokud máme zapojeno více řídicích zařízení, tak to, které bude řídit komunikaci a tedy i hodinový signál, se určí tím, které dříve vyšle start-bit.

Po vyslání start-bitu následuje poslání adresy zařízení. Tato adresa může být sedmibitová nebo desetibitová.

- V případě, že máme sedmibitovou adresu, se po start-bitu pošle číslo adresy postupně od nejvýznamnějšího bitu a za něj se přidá bit, který určuje, jestli chceme z příslušného zařízení číst data, v tomto případě zapíšeme logickou 1, nebo do něj data posílat, zapíšeme logickou nulu. Poslaná data přijala všechna připojená zařízení, a pokud se shodují s adresou některého z nich, tak to zařízení vyšle na linku SDA tzv. ACK (Acknowledge) bit. Pokud se na SDA lince objeví logická nula, znamená to, že zařízení úspěšně přijalo adresu a můžeme s ním začít komunikaci, v opačném případě se zápis nevydařil a řídicí zařízení musí určit další postup.
- V případě, že máme desetibitovou adresu zařízení, tak v prvním bloku pošleme bity 11110, následují dva nejvýznamnější bity adresy zařízení a bit, kterým určíme, jestli chceme zapisovat nebo číst data. Zařízení, které mají stejné dva bity, odešlou ACK bit. V dalším bloku se odešle zbytek adresy zařízení, také s údajem, jestli čteme nebo zapisujeme data. ACK bit nyní již odešle zařízení, které má příslušnou adresu.

Dále následuje posílání dat. Řídicí zařízení pokračuje ve vysílání hodinového signálu a zapisuje nebo čte hodnoty z příslušného zařízení. Devátý bit je vždy ACK bit, který určí, jestli se data správně přenesla. Pokud se stane, že zařízení nestihá data zpracovávat nebo posílat s frekvencí, kterou určuje hodinový signál řídicího zařízení, má zařízení možnost převzít kontrolu nad linkou SCL a pozdržet ji, dokud nebude připraveno znovu komunikovat.

Pro ukončení komunikace vyšleme tzv. stop-bit. Ten vyšleme tak, že linku SCL přepneme do logické 1, a poté přepneme do logické 0 i linku SDA. Speciální případ ukončení komunikace s jedním zařízením je vyslání start-bitu místo stop-bitu. Tím začneme komunikaci s jiným zařízením a zajistíme, že kontrolu nad komunikací nepřevzme jiné řídicí zařízení. [14]



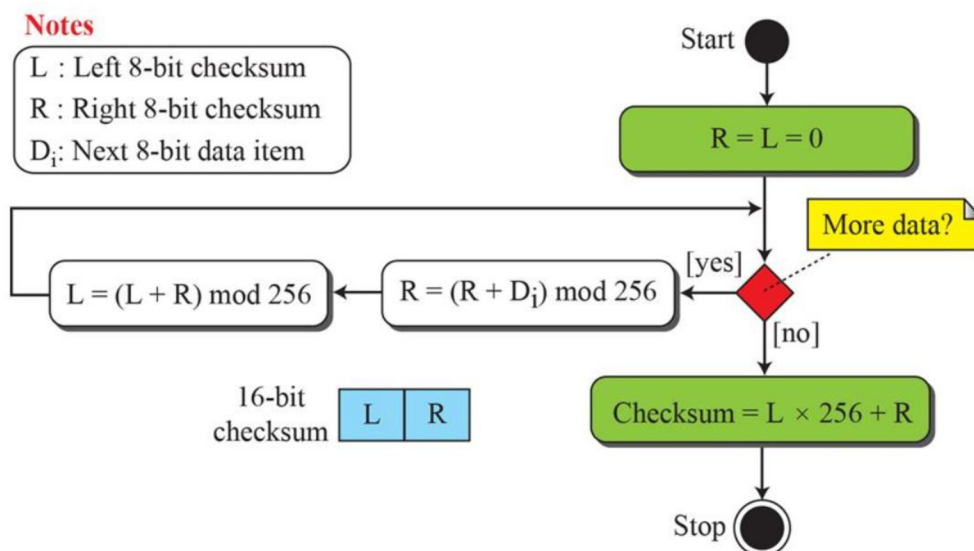
Obrázek 7 - Struktura zprávy u I²C komunikace [14]

2.2 Algoritmus kontrolního součtu

Algoritmy kontrolního součtu využíváme k ověření správnosti přenesení dat. Je to metoda, kdy se k posílané informaci nebo souboru vygeneruje dodatečná informace, tzv. kontrolní součet. Tuto metodu provedeme před posláním informace a při přijetí. Pokud se hodnoty vygenerované touto metodou shodují, zpráva je v pořádku. V opačném případě zpráva v pořádku není nebo se vyskytla chyba při přenosu dat. Jako jednoduchý příklad kontrolního součtu můžeme uvést paritu, kde se sečtou posílané hodnoty a podle toho, jestli je součet kladný nebo záporný, se přidá dodatečné číslo. Při stahování souborů z internetu se můžeme setkat s kontrolními součty typu MD5, SHA-1 nebo SHA-2. Při posílání dat z mikrokontrolérů se využívají kontrolní součty Fletcher, Adler nebo CRC.

2.2.1 Kontrolní součet Fletcher

Algoritmus vytvořil John G. Fletcher v 70. letech 20. století. Tento algoritmus závisí na pořadí poslaných hodnot, čímž se liší od nejjednodušších kontrolních součtů, které neupozorní na chybu při přehození poslaných hodnot. U tohoto kontrolního součtu se nejdříve celá zpráva rozdělí do bloků. Tyto bloky jsou osmibitové (typ Fletcher-16), šestnáctibitové (typ Fletcher-32) nebo třiceti dvoubitové (typ Fletcher-64). Nyní se zaměříme na typ Fletcher-16. Zpráva je rozdělena do osmibitových bloků. Metoda spočívá ve vypočítání dvou konstant. První konstanta je vypočítána tak, že sčítáme aktuální hodnotu této konstanty s posílanou hodnotou a poté tento součet vydělíme číslem 256 nebo 255. První konstanta je poté zbytek po tomto dělení. Druhá konstanta je získána stejným způsobem, jen zde sčítáme aktuální hodnotu druhé konstanty s hodnotou první. Druhá konstanta je zbytek po dělení tohoto součtu číslem 256 nebo 255. Tímto způsobem projdeme celou posílanou zprávu a vytvoříme dvě konstanty. Hodnota, kterou přidáme k posílané zprávě, bude součet druhé konstanty vynásobené hodnotou 256 a první konstanty. Názorně je operace ukázána v příloženém diagramu.[15]

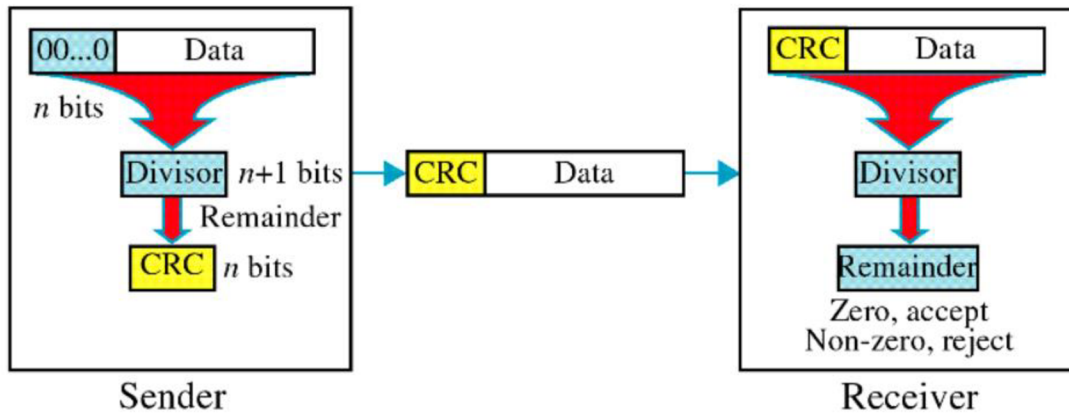


Obrázek 8 - Algoritmus kontrolního součtu Fletcher [16]

2.2.2 CRC

CRC, neboli “Cyklický redundantní součet“, je výpočetně náročnější algoritmus než předešlý algoritmus Fletcher. S tímto typem algoritmu se můžeme setkat také u některých pevných disků, kdy detekuje chybu při přenášení souborů. Hlavním rozdílem oproti jiným algoritmům kontrolních součtů je to, že za pomoci CRC dokážeme některé chyby, často opakující se při přenosu dat, i opravovat. Posílané data, které vyjádříme ve dvojkové soustavě, můžeme taky brát jako koeficienty polynomu. Pro výpočet přidané hodnoty CRC vydělíme datový polynom generujícím polynomem. Generující polynomy

jsou primitivní polynomy, které jsou schopné detekovat dvoubitovou chybu i ve velmi dlouhých polynomech. Pro většinu aplikací známe generující polynomy, které jsou vhodné a hojně využívané. Tím, že jsou vhodné, myslíme, že jejich použití zajišťuje téměř bezchybnou kontrolu. Hodnota CRC se přidá k posílanému datovému polynomu a přijímací zařízení vydělí hodnotu znovu generujícím polynomem, a pokud dělení vyjde beze zbytku, tak přenos proběhl bez chyb. [17]



Obrázek 9 - Cyklický redundantní součet[18]

3 EXPERIMENTÁLNÍ VOZIDLO CAR4

Projekt Car4 vznikl v roce 2010 v rámci dvou bakalářských a tří diplomových prací. Tento projekt má sloužit především k vývoji a testování algoritmů jízdní stability a k edukativním účelům.

3.1 Aktuální stav vozidla

- Řídící část vozidla tvoří tři mikrokontroléry typu dsPIC33FJ128MC804. Tyto mikrokontroléry obsluhují všechny připojené periferie.
- Elektrickou energii vozidlu dodávají sérioparalelně zapojené lithium-polymerové baterie. Nominální napětí je 22,2 V a celková kapacita 13200 mAh.
- Ve vozidle je také jednotka pro monitorování stavu baterie. [6]
- Vozidlo má 4 nezávisle poháněná kola. Každé kolo je poháněno stejnosměrným motorem s enkodérem. Ke každému kolu je navíc připojen RC servomotor, který řídí natočení kola.
- Ovládání vozidla je možné přes dálkové ovládání nebo přes rozhraní UART.

3.2 Důvod nového řešení

Hardware vozidla vychází zejména z diplomové práce Ing. Josefa Vejlupek, Ph.D. z roku 2010. [19] Vytvořená řídicí jednotka má zajistit nízko úrovně ovládání vozidla a bude komunikovat s nadřazenými řídicími systémy. Společným cílem této a druhé vznikající bakalářské práce je obnovení a sjednocení hardwaru, který se od počátku projektu neměnil.

4 NÁVRH ŘÍDICÍ JEDNOTKY

Tato kapitola se zabývá návrhem a vytvořením nové řídicí jednotky. Nová řídicí jednotka musí být schopná spolehlivě obsluhovat všechny zařízení, které vozidlo obsahuje, a také musí dokázat komunikovat s nadřazeným řídicím systémem. Návrh řídicí jednotky začíná sepsáním požadavků, které od nové desky požadujeme. Poté bylo vytvořeno elektronické schéma řídicí jednotky a navržena deska plošných spojů.

4.1 Požadavky

Po provedení rešerše o aktuálním stavu vozidla byly sepsány požadované prvky, aby nová jednotka mohla nahradit současné řešení. Nová jednotka musí mít tyto prvky:

- Dvě sběrnice I²C
 - pro čtení hodnot z enkodérů
 - pro snímání stavu baterie
- Dvě sběrnice UART
 - pro připojení modulu pro dálkové ovládání
 - pro komunikaci s nadřazeným řídicím systémem
- Sběrnici SPI
 - Pro příjem informací ze senzorů
- Výstupy pro řízení motorů
 - 4 výstupy PWM signálu pro řízení DC motorů pohánějících kola
 - 4 výstupy PWM signálu pro řízení RC servomotorů, které ovládají natočení kol
 - Výstupy DIR (Direction) a DIS (Disable) pro každý motor
- USB konektor – možnost připojení řídicí jednotky k počítači
- Linky GPIO (General purpose input/output)
- Napájení ze zdroje napětí 3.3 V

4.2 Zvolené řešení

Řídicí jednotka bude rozdělena do dvou desek. První deska bude typu plug-in a bude na ní jen mikrokontrolér s nezbytnými prvky pro jeho funkci (krystal, tlačítko pro restartování a JTAG konektor pro programování mikrokontroléru) a notifikační LED diody. Tato deska bude připojena na druhou desku, kde budou příslušné obvody a výstupy k jednotlivým periferiím. Řešení s plug-in deskou umožňuje jednodušší údržbu, popřípadě opravu, a také v případě potřeby změny řídicí desky stačí vyměnit desku s výstupy k periferiím a k ní připojit plug-in desku s mikrokontrolérem.

Mikrokontrolér

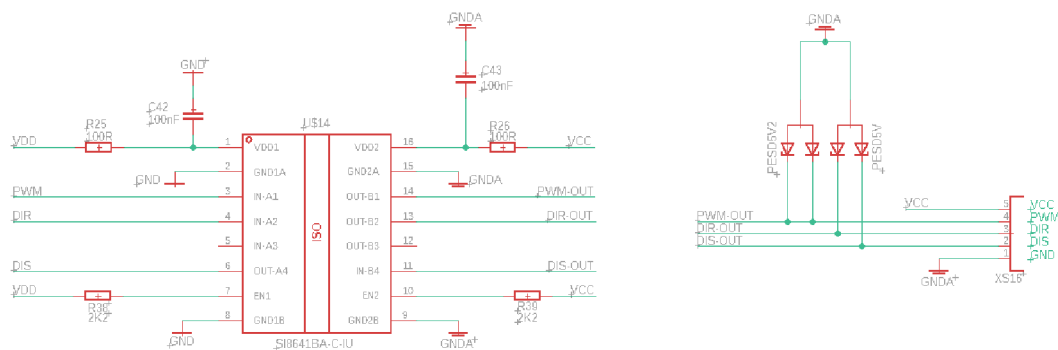
Při výběru mikrokontroléru jsme vybírali především z produktů firmy Microchip Technology, protože i původní řešení bylo založeno na mikrokontrolérech této firmy. Podle sepsaných požadavků byl vybrán šestnáctibitový mikrokontrolér dsPIC33EP512MU810. Hlavní vlastnosti tohoto mikrokontroléru:

- Modifikovaná harvardská architektura
- Pouzdro TQFP se 100 piny
- Až 28 PWM výstupů
- Možnost až 4 komunikačních sběrnic UART, 4 sběrnic SPI a 2 sběrnic I²C
- Velké množství přemapovatelných pinů, a tedy možnost využít jich co nejvíce, zejména pro zařazení většího množství komunikačních sběrnic

Všechny vlastnosti tohoto mikrokontroléru jsou k nalezení v katalogovém listu. [20]

Galvanické oddělení mezi řídicí částí a výkonovou částí

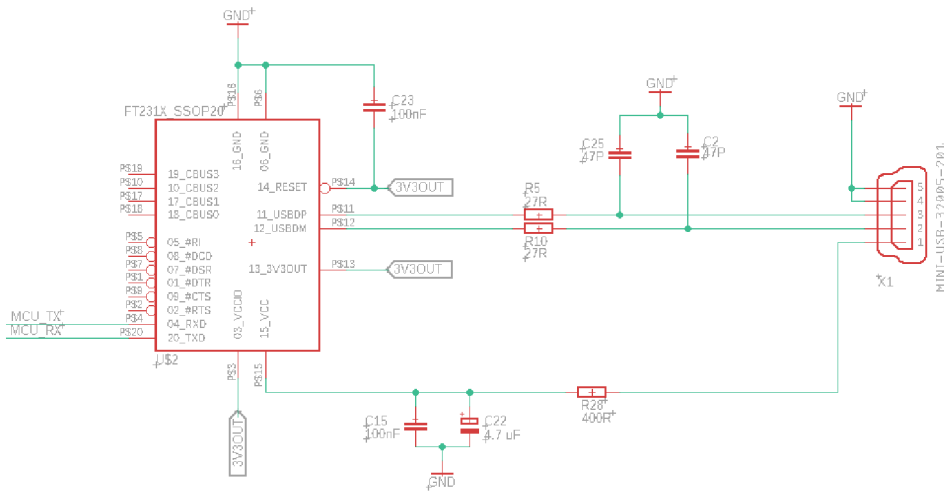
Z nové řídicí desky vedou ke každému DC motoru výstupy s PWM signálem pro řízení motoru, signálem DIR (Direction) pro určení směru otáčení a signálem DIS (Disable) pro možnost odstavení motoru. Protože řídicí část je napájena menším napětím než výkonové měniče pro motory, umístíme mezi výstupní signály a obvody s mikrokontrolérem galvanické oddělení. Toto řešení používáme proto, že při spínání tranzistorů vzniká velká strmost napětí a veliké rušení, které by mohlo zničit řídicí jednotku. Konkrétně jsme využili magnetické izolátory typu Si864x od firmy Silicon Labs. [21] Zapojení je na následujícím obrázku.



Obrázek 10 - Galvanické oddělení obvodů

Převodník UART-USB

Dalším požadavkem u nové řídicí jednotky byl USB konektor s možností připojení desky k počítači. Zde byl využit integrovaný obvod FT231X od firmy FTDI Chip, který umožňuje jednoduché připojení USB konektoru přes rozhraní UART. Zapojení USB konektoru je na následujícím schématu.



Obrázek 11 - Převodník UART-USB

Komunikační sběrnice

Ve výsledném řešení jsou zabudovány tyto komunikační sběrnice:

- 3 sběrnice UART
 - Pro komunikaci s nadřazeným řídicím systémem
 - Pro připojení modulu pro dálkové ovládání
 - Pro připojení USB konektoru přes UART-USB převodník
- 2 sběrnice I²C
 - Pro obsluhu enkodérů
 - Pro připojení jednotky k monitorování stavu baterie
- 2 sběrnice SPI
 - Pro čtení dat ze senzorů
 - Druhá sběrnice pro možnost připojení dalších prvků

Ostatní prvky řídicí jednotky

Kromě již zmíněných prvků nová řídicí deska obsahuje:

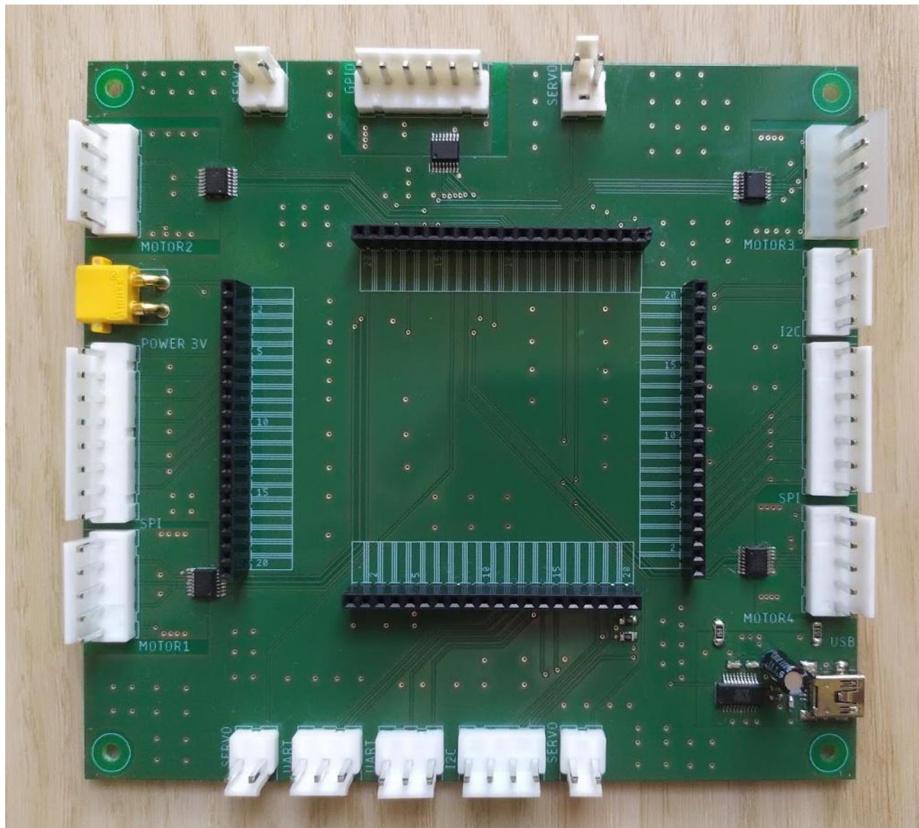
- Řízení RC servopohonů – 4 výstupy PWM signálu pro ovládání natočení kol pomocí servopohonů
- Notifikační LED diody
- Ochrana proti ESD – u všech výstupů z řídicí desky byly dány součástky na ochranu proti elektrostatickým výbojům, aby se zabránilo zničení celé jednotky

4.3 Nová řídicí jednotka

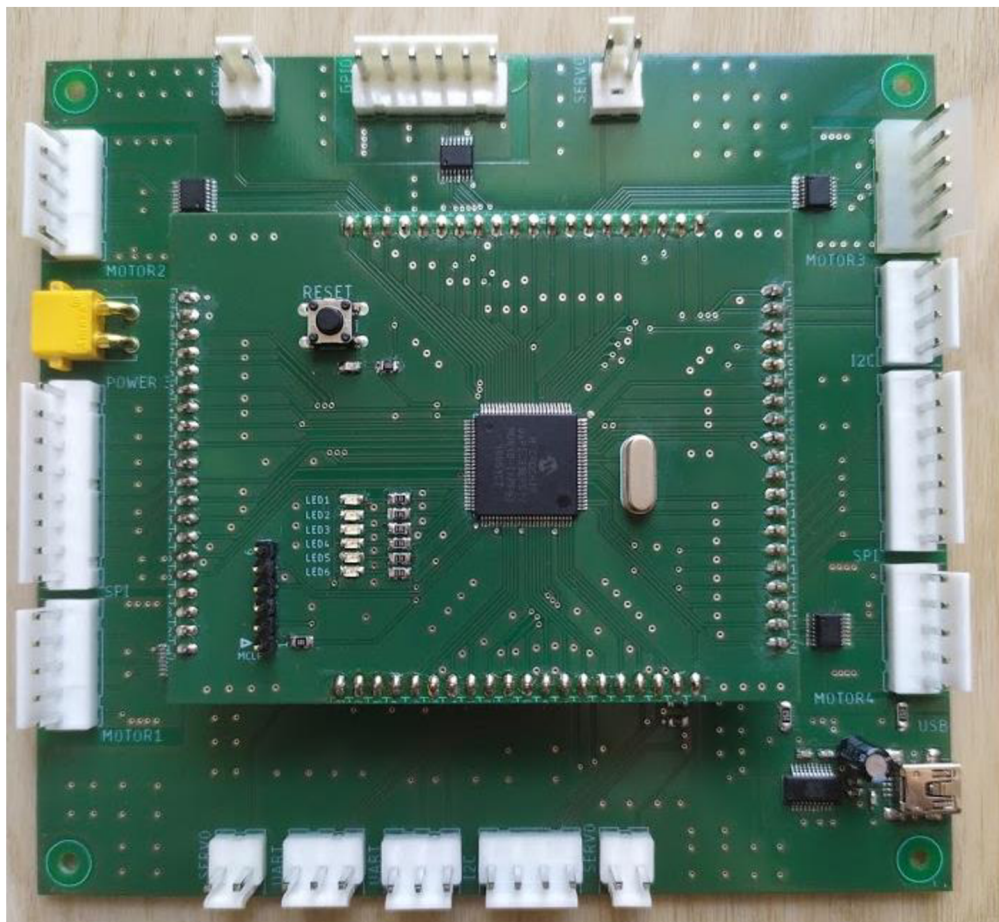
Na přiložených obrázcích jsou ukázány části nové řídicí jednotky, tedy plug-in deska s mikrokontrolérem a spodní deska s výstupy k jednotlivým perifériím. Na poslední fotce je vyfocena finální podoba řídicí jednotky.



Obrázek 12 - Plug-in deska s mikrokontrolérem



Obrázek 13 - Deska s výstupy k jednotlivým perifériím



Obrázek 14 - Finální podoba řídicí jednotky

5 ŘÍDICÍ ALGORITMUS

Řídicí algoritmus byl v rámci předchozích prací mnohokrát upravován, proto bylo nutné ho přizpůsobit novému zařízení. Kvůli situaci s koronavirem v tomto roce nebyla nová řídicí jednotka implementována do vozidla, a proto bylo pozměněno zadání bakalářské práce. Nejdříve bude přizpůsoben řídicí software novému hardwaru. Funkčnost bude ukázána na komunikaci mezi řídicí jednotkou a počítačem. Otáčkový regulátor nebude proveden na vozidle Car4, ale na samostatném DC motoru.

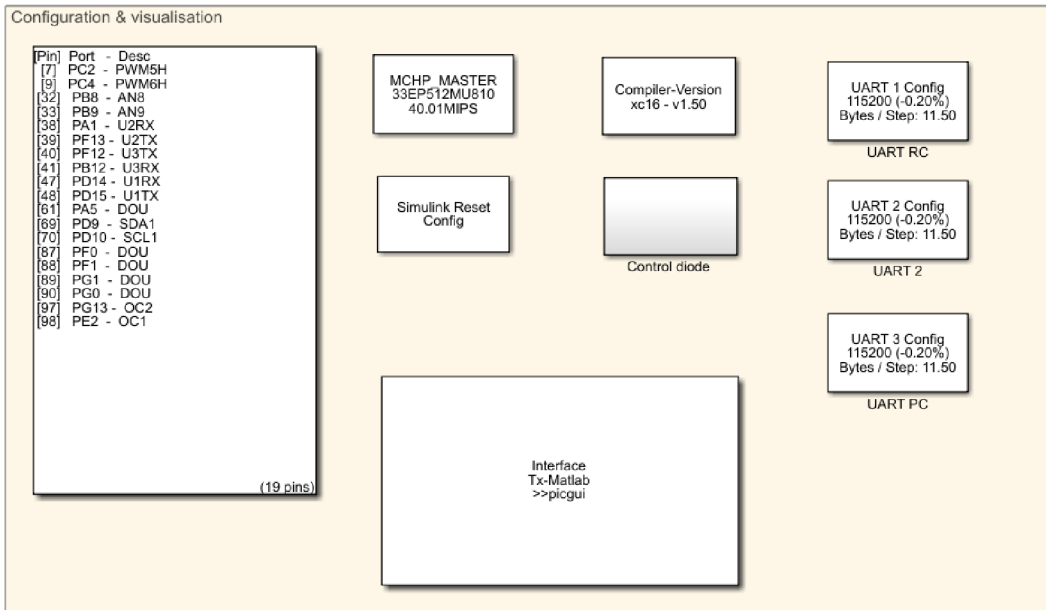
Program pro vozidlo Car4 je vytvořen pomocí programu Matlab a jeho nástavby Simulink od firmy MathWorks. Zde poté využíváme toolbox *“MPLAB Device Blocks for Simulink“*. Uživatel s pomocí tohoto toolboxu vytvoří blokové schéma, z kterého se poté automaticky vygeneruje C kód a nahraje ho do mikrokontroléru.

5.1 Aktualizace řídicího programu

Kvůli tomu, že je vytvořena nová řídicí jednotka, musíme aktualizovat program tak, abychom mohli tuto řídicí desku nasadit do vozidla. Zde se projevila výhoda využití toolboxu *“MPLAB Device Blocks for Simulink“*, protože ve většině případů stačí překonfigurovat bloky jednotlivých periférií, tak aby program fungoval na nově vytvořené řídicí desce. Blokové schéma je rozděleno do několika sekcí. Jedna sekce se týká konfigurace mikrokontroléru a konfigurace komunikačního rozhraní UART. Ve druhé sekci je vytvořena komunikace mezi dálkovým ovládním a řídicí jednotkou. Další sekce obsluhuje čtení dat z enkodérů a ovládní motorů.

5.1.1 Konfigurace systému

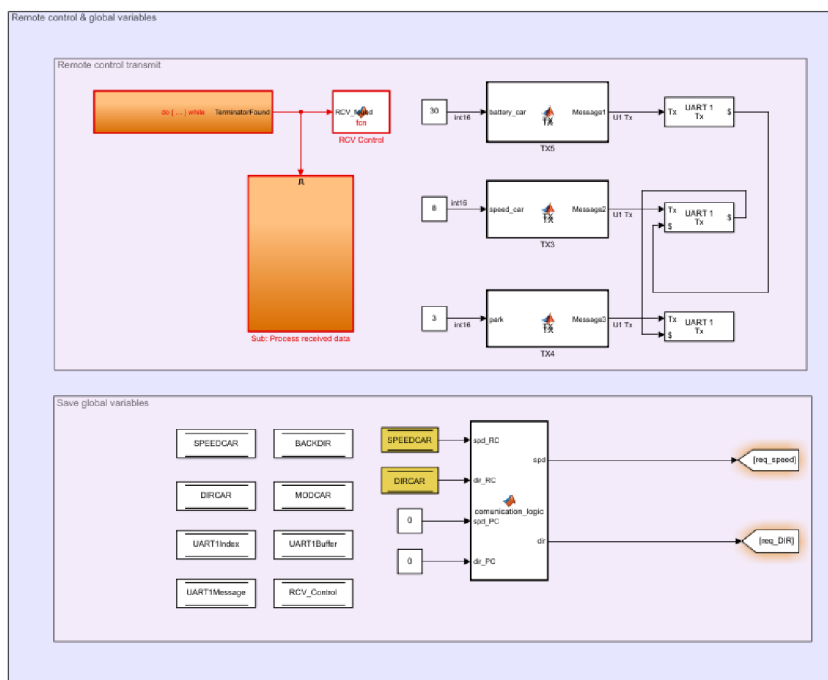
V této sekci blokového schématu jsou nakonfigurovány základní prvky systému. Jsou zde bloky *„Microchip Master“*, *„Compiler Options“*, ve kterých vybereme použitý mikrokontrolér, kompilér a jejich nastavení. Blok *„Simulink Reset Config“* slouží k obnovení modelu. V této sekci máme taky nakonfigurované sběrnice UART a blok grafického rozhraní, přes který můžeme posílat hodnoty z počítače a v reálném čase vyčítat hodnoty veličin do grafu. Pro možnost notifikace je zde subsystém *„Control diode“*.



Obrázek 15 - Konfigurace systému

5.1.2 Komunikace s dálkovým ovládáním

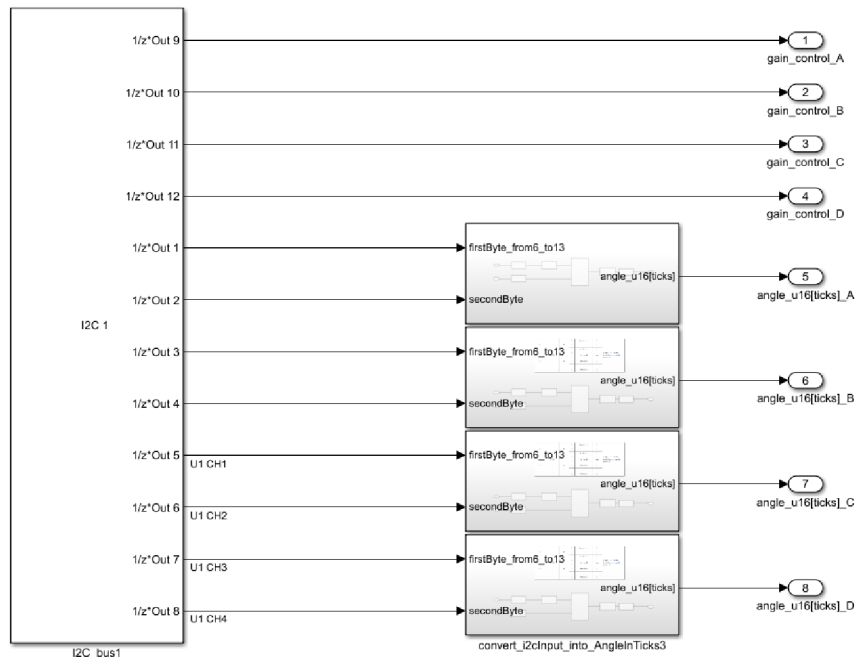
Druhá sekce se týká komunikace mezi řídicí jednotkou a dálkovým ovládáním. Bloky „UART Rx“ a „UART Tx“ obstarávají příjem a odesílání informací po sběrnici UART. Tyto bloky pracují jen s osmibitovými hodnotami, proto musíme všechny hodnoty nejprve převést z šestnáctibitového datového typu na osmibitový nebo opačně v případě příjmu dat. Pro kontrolu správnosti poslaných a přijatých dat se zde využívá algoritmu kontrolního součtu Fletcher, který byl popsán v rešeršní části. O tuto operaci se starají bloky „Matlab Function“, které také obstarávají zapisování přijatých hodnot do vytvořených globálních proměnných.



Obrázek 16 - Komunikace s dálkovým ovládáním

5.1.3 Data z enkodérů

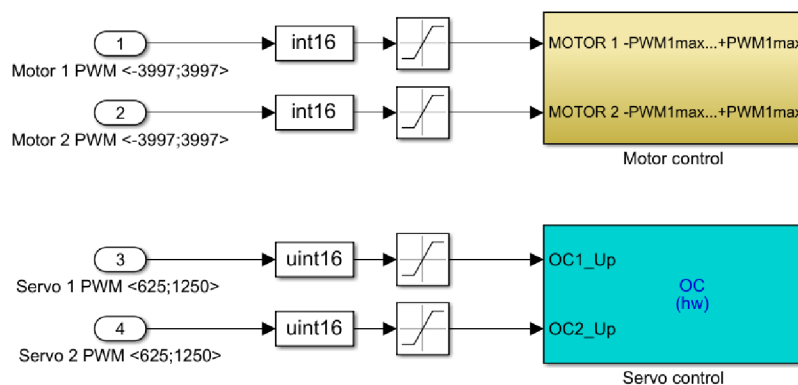
Další částí programu je vyčítání dat z enkodérů. Všechny 4 enkodéry jsou připojeny ke společné sběrnici I²C. Zde využíváme bloku „BUS I2C MASTER“. V tomto bloku nastavíme používané piny, přenosovou rychlost a pomocí přednastavených příkazů nastavíme funkci sběrnice. Data, která přijmeme, poté zpracováváme, abychom zjistili rychlost vozidla. Z informací, které přijmeme, můžeme také zjistit, jestli jednotlivé enkodéry pracují v pořádku.



Obrázek 17 - Příjem dat z enkodérů

5.1.4 Řízení motorů a servopohonů

V subsystému „Motor control“ najdeme blok „PWM High Speed“, který slouží k posílání PWM signálů do motorů. Dále jsou zde bloky „Digital Output“, které posílají do motoru informaci o směru otáčení, příp. o vypnutí motoru. Druhou částí této sekce je blok „Output Compare – HW“, který vysílá PWM signál do servopohonů, které ovládají natočení kol.

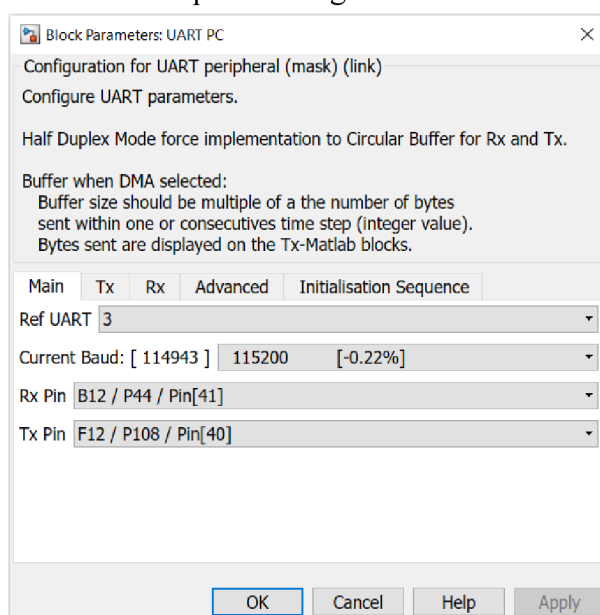


Obrázek 18 - Řízení motorů a servopohonů

5.2 Komunikace s PC

Při vytváření komunikace mezi počítačem a mikrokontrolérem jsem se inspiroval prací Ing. Jany Šumpelové z roku 2015, která se věnovala dálkovému ovládní vozidla.[4] V kapitole 3.4 byl popsán převodník UART-USB, díky kterému můžeme připojit řídicí desku k počítači.

Díky integrovanému obvodu FT231X probíhá komunikace stejně jako komunikace po sběrnici UART. Nejdříve tedy musíme nakonfigurovat blok „*UART Configuration*“, ve kterém nastavíme hodnotu přenosové rychlosti a piny, které pro komunikaci využíváme. V tomto bloku můžeme také nastavit, jestli komunikace bude oboustranná, paritu a stop-bit. Stejně nastavení, zejména přenosovou rychlost, musíme nastavit i v počítači, aby komunikace správně fungovala.

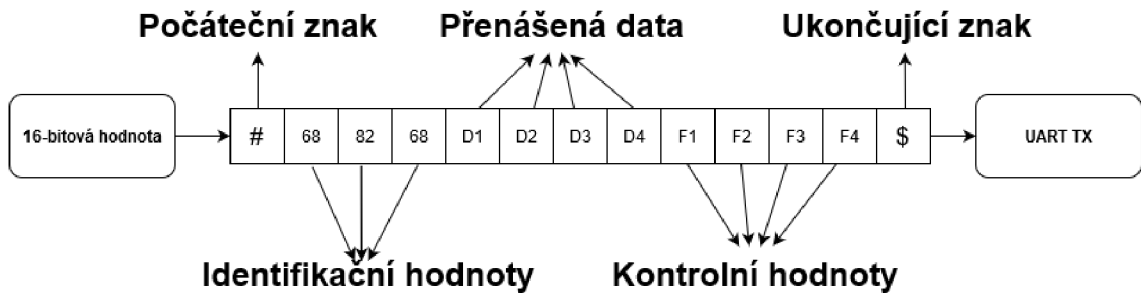


Obrázek 19 - Nastavení UART komunikace

Komunikace po rozhraní UART pracuje s osmibitovými neznaménkovými hodnotami (datový typ uint8), ale my používáme v celém programu šestnáctibitové hodnoty. V prvním kroku tedy musíme vytvořit funkci, která převede používané hodnoty na zprávu, kterou budeme schopni poslat. Funkci můžeme do programu implementovat pomocí bloku „*Matlab Function*“. Nejdříve funkce převede pomocí příkazu šestnáctibitovou hodnotu na dvě osmibitové hodnoty. Následně jsou tyto hodnoty převedeny z dekadického zápisu do zápisu pomocí znaků ASCII tabulky. Posledním krokem této funkce je výpočet algoritmu kontrolního součtu Fletcher, který byl popsán v kapitole 2.3.1. Tento algoritmus nám ke zprávě přidá 4 znaky ASCII, které budou sloužit ke zkontrolování správnosti přenosu.

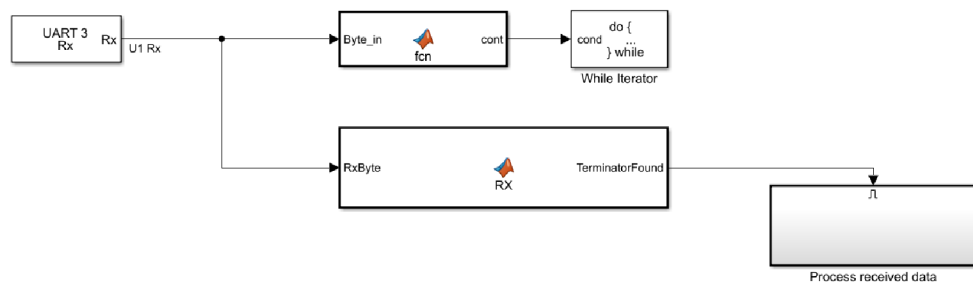
Konečná zpráva bude mít tedy tvar: nejdříve funkce přidá ke zprávě počáteční znak „#“ (V ASCII tabulce jako 35), následně pošleme 3 identifikační čísla, která určují, k jaké proměnné preposílaná hodnota patří. Poté následuje poslání naší hodnoty ve formě 4 znaků ASCII, za nimi jsou přiloženy 4 kontrolní znaky, které jsme získali

z algoritmu kontrolního součtu Fletcher. Na závěr zprávy funkce přidá ukončující znak „\$“ (V ASCII tabulce jako 36). Tato zpráva tedy obsahuje 13 osmibitových bloků, které vstupují do bloku „UART Tx“, která provede přenos po sběrnici UART.



Obrázek 20 - Tvar zprávy při komunikaci mezi PC a mikrokontrolérem

K přijímání dat slouží blok „UART Rx“, který příjem dat obsluží. Výstupem tohoto bloku jsou přijaté data. Tyto data poté pošleme do bloku „Matlab Function“, který čeká na znak „#“, protože víme, že posílané zprávy začínají tímto znakem. Po přijetí tohoto znaku subsystém převede poslanou zprávu na jednotlivé bloky – počáteční znak „#“, identifikační čísla, posílané hodnoty, kontrolní hodnoty a ukončující znak „\$“. Hodnoty jsou převedeny zpět na dekadická čísla. Funkce opět spočítá algoritmus kontrolního součtu Fletcher, a pokud se hodnoty shodují s přijatými kontrolními hodnotami, tak přenos proběhl správně. Podle identifikačních čísel systém rozhodne do které globální proměnné přijaté data zapsat.

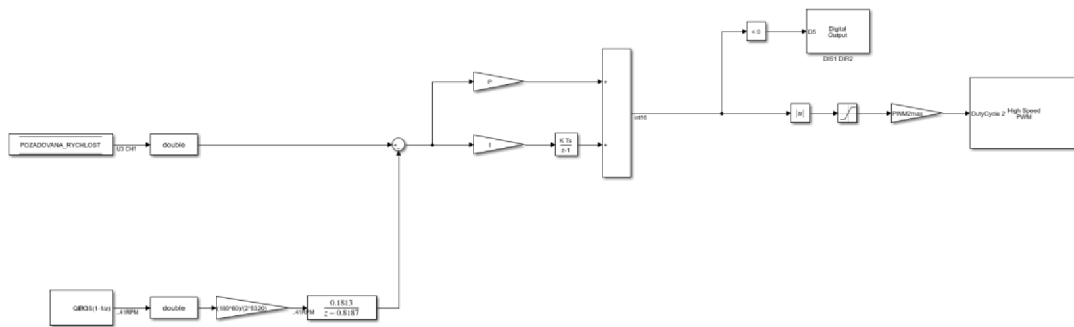


Obrázek 21 - Příjem dat z počítače

Čtení a posílání dat z počítače do mikrokontroléru je vyřešeno přes skript v aplikaci Matlab. Nejdříve musíme vytvořit virtuální sériový port a nakonfigurovat jeho vlastnosti. Poté pomocí funkce “read“ můžeme číst data, která přes tento sériový port přijdou a naopak funkcí “write“ hodnoty zapisujeme. Struktura zprávy, kterou posíláme do mikrokontroléru, je stejná jako ta, kterou posíláme do počítače.

5.3 Regulátor rychlosti DC motoru

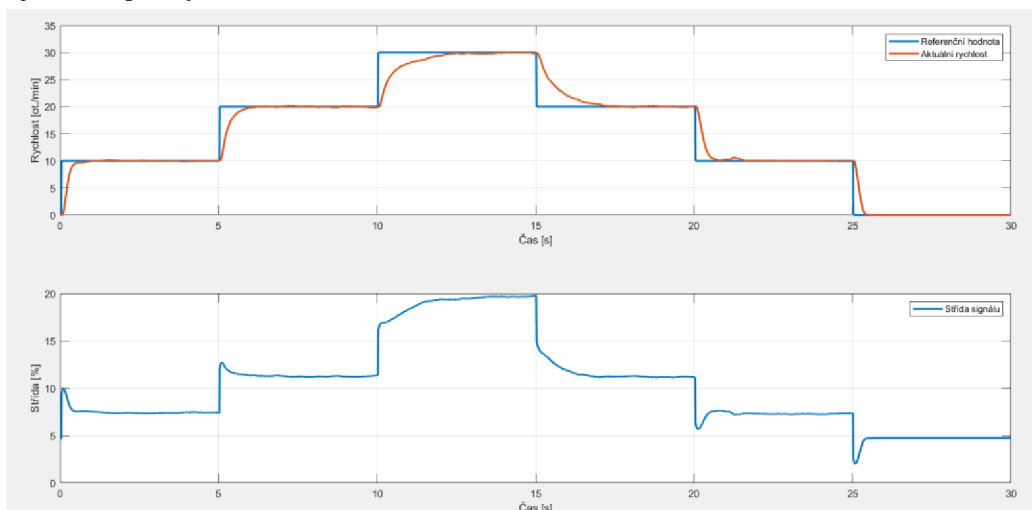
Model regulátoru rychlosti otáčení DC motoru vypadá tak, že požadovanou hodnotu rychlosti posíláme z počítače pomocí komunikace, kterou jsme vytvořili v předcházející kapitole. K motoru máme připojený kvadrurní enkodér, proto využijeme bloku „Quadrature Encoder *Interface*“ ze kterého získáme aktuální hodnotu rychlosti. Tuto informaci vyfiltrujeme pomocí bloku „Discrete Transfer Fnc“, který vytváří dolní propust 1. řádu. Regulační odchylku posíláme do PI regulátoru. Nastavení parametrů regulátoru bylo laděno podle metody Ziegler-Nichols. Konečné koeficienty regulátoru jsou: $K_P = 0.0048$ a $K_I = 0.0138$. Z regulátoru vystupuje hodnota střídý, která poté vstupuje do bloku PWM, který generuje PWM signál jdoucí z řídicí desky do motoru.



Obrázek 22 - Blokové schéma regulace DC motoru

Řídicí desku jsme propojili s modulem H-můstku L9110S a tento modul poté se stejnosměrným motorem FIT0487 od firmy DFROBOT. Tento motor pracuje při napětí 6 V a má jmenovité otáčky 41 otáček za minutu. Motor má zabudovaný enkodér, pomocí kterého můžeme vyčítat rychlost otáčení. [22]

Na prvním grafu je možné vidět průběh regulované veličiny, tedy rychlosti, a na druhém průběh hodnoty střídý. Grafy byly pořízeny při testování, kdy vždy po 5 vteřinách zvýšíme požadované otáčky o 10 otáček za minutu až do hodnoty 30 otáček za minutu, poté jdeme opačným směrem.



Obrázek 23 - Grafy regulované veličiny a akční veličiny

6 ZÁVĚR

Cílem této bakalářské práce bylo vytvořit novou řídicí jednotku, která bude základem pro budoucí rozvoj projektu CAR4. Tato jednotka bude tvořit nízkourovňové ovládání vozidla a bude komunikovat s nadřazenými řídicími systémy.

Nejdříve byly v rešeršní části popsány komunikační sběrnice, které se využívají při komunikaci mezi mikrokontroléry a připojenými perifériemi a které jsme poté využili při tvorbě nové řídicí jednotky. V rešeršní části bylo také seznámení s algoritmy kontrolního součtu, používané kvůli bezpečnosti přenosu dat.

Hlavní část práce se zabývala vytvořením nové řídicí jednotky a práce s ní. V první kapitole bylo podle sepsaných požadavků navrženo elektronické schéma řídicí jednotky a navržena deska plošných spojů. Nová řídicí jednotka se skládá ze dvou desek. První deska obsahuje příslušné obvody a výstupy k perifériím a k ní je připojena deska s mikrokontrolérem.

Následně byl optimalizován program, který vznikl v rámci předchozích prací. Tento program řídí komunikaci se všemi připojenými perifériemi a řídí funkci vozidla.

Další kapitoly se týkali vytvoření komunikace mezi počítačem a mikrokontrolérem, a také vytvoření regulace otáček stejnosměrného motoru. Komunikace s počítačem probíhá přes rozhraní UART, konkrétně připojením přes USB konektor. Zde je nejdůležitější částí převedení šestnáctibitových hodnot do tvaru, který jsme schopni po lince UART posílat. Při přenosu se zde využívá také metoda kontrolního součtu Fletcher, která určuje, zda přenos proběhl správně. V poslední kapitole byl využit regulátor typu PI, který nastavuje rychlost otáčení DC motoru na požadovanou hodnotu.

. Kvůli karanténním opatřením, spojených s koronavirem, nebyla jednotka implementována do vozidla. Funkčnost byla otestována při komunikaci s počítačem a při řízení samostatného DC motoru. Řídicí jednotka má vytvořit základ pro budoucí rozvoj vozidla.

7 SEZNAM POUŽITÉ LITERATURY

- [1] GREPL, R.; VEJLUPEK, J.; LAMBERSKY, V.; JASANSKY, M.; VADLEJCH, F.; COUPEK, P. *Development of 4WS/4WD Experimental Vehicle: platform for research and education in mechatronics*. Istanbul: IEEE International Conference on Mechatronics, 2011. ISBN 978-1-61284-982-9.
- [2] NAJMAN, Jan. *Rozšíření robotu Car4 o palubní počítač a snímače Kinect a Hokuyo*. Brno, 2013. 41 s. Bakalářská práce na Fakultě strojního inženýrství Vysokého učení technického na Ústavu mechaniky těles, mechatroniky a biomechaniky. Vedoucí bakalářské práce Ing. Josef Vejlupek.
- [3] NAJMAN, Jan. *Aplikace SLAM algoritmů pro vozidlo s čtyřmi řízenými koly*. Brno, 2015. 75 s. Diplomová práce na Fakultě strojního inženýrství Vysokého učení technického na Ústavu mechaniky těles, mechatroniky a biomechaniky. Vedoucí diplomové práce doc. Ing. Robert Grepl, Ph.D..
- [4] ŠUMPELOVÁ, Jana. *Úprava dálkového ovládní experimentálního vozidla Car4*. Brno, 2015. 43 s. Bakalářská práce na Fakultě strojního inženýrství Vysokého učení technického na Ústavu mechaniky těles, mechatroniky a biomechaniky. Vedoucí bakalářské práce Ing. Jan Chalupa.
- [5] VRBKOVÁ, Michaela. *Návrh algoritmu automatického parkování pro experimentální vozidlo se čtyřmi řízenými koly*. Brno, 2016. 32 s. Bakalářská práce na Fakultě strojního inženýrství Vysokého učení technického na Ústavu mechaniky těles, mechatroniky a biomechaniky. Vedoucí bakalářské práce doc. Ing. Robert Grepl, Ph.D.
- [6] DOBOSSY, Barnabás. *Energy managment pro robotické vozidlo Car4*. Brno, 2017. 58 s. Bakalářská práce na Fakultě strojního inženýrství Vysokého učení technického na Ústavu mechaniky těles, mechatroniky a biomechaniky. Vedoucí bakalářské práce Ing. Jan Najman.
- [7] PANDIT, Abhiemanyu. *Serial Communication Protocols* [online]. 29. 4. 2019 [cit. 2020-05-10]. Dostupné z: <<https://circuitdigest.com/tutorial/serial-communication-protocols>>.
- [8] *MCU: protokol UART* [online]. 22. 3. 2016 [cit. 2020-05-12]. Dostupné z: <http://src.athaj.cz/teaching/rev/arch_uart>.
- [9] *Basics of UART communication* [online]. 2016 [cit. 2020-05-13]. Dostupné z: <<https://www.circuitbasics.com/basics-uart-communication>>.

- [10] TIŠNOVSKÝ, Pavel. *Externí sériové sběrnice SPI a I²C* [online]. 30. 12. 2008 [cit. 2020-05-15]. Dostupné z: <<https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c>>.
- [11] *Basics of the SPI communication protocol* [online]. 2016 [cit. 2020-05-20]. Dostupné z: <<https://www.circuitbasics.com/basics-of-the-spi-communication-protocol>>.
- [12] *I²C* [online]. 2015 [cit. 2020-05-26]. Dostupné z: <<https://learn.sparkfun.com/tutorials/i2c/all>>.
- [13] TIŠNOVSKÝ, Pavel. *Komunikace po sériové sběrnici I²C* [online]. 8. 1. 2009 [cit. 2020-05-26]. Dostupné z: <<https://www.root.cz/clanky/komunikace-po-seriove-sbernici-isup2supc>>.
- [14] *Basics of the I²C communication protocol* [online]. 2016 [cit. 2020-05-26]. Dostupné z: <<https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol>>
- [15] *Fletcher's checksum* [online]. 19. 6. 2020 [cit. 2020-06-20]. Dostupné z: <https://en.wikipedia.org/wiki/Fletcher%27s_checksum?fbclid=IwAR2bZdKaxi93veLPdc1hTTmAh2kTBPP-17vOyH-17ayiGEiGMtt2rWEV6Zs>.
- [16] FOROUZAN, Behrouz A. *Data communications and networking*. 5th ed. New York: McGraw-Hill, 2013. ISBN 978-0-07-337622-6.
- [17] HORČÍK, Rostislav. *CRC a pseudonáhodná čísla* [online]. 20. 4. 2010 [cit. 2020-05-29]. Dostupné z: <<https://docplayer.cz/17120355-Cyklicke-redundantni-soucty-a-generatory.html>>.
- [18] CHOUBE, Sumit. *Cyclic Redundancy check and Checksum* [online]. 19. 1. 2020 [cit. 2020-06-20]. Dostupné z: <<https://cppsecrets.com/users/147811511710910511646991041111179810149556411810511646101100117/Cyclic-Redundancy-check-and-Checksum.php>>.
- [19] VEJLUPEK, Josef. *Vývoj elektroniky pro řízení trakce experimentálního vozidla*. Brno, 2010. 73 s. Diplomová práce na Fakultě strojního inženýrství Vysokého učení technického na Ústavu mechaniky těles, mechatroniky a biomechaniky. Vedoucí diplomové práce Ing. Robert Grepl, Ph.D..
- [20] *16-Bit Microcontrollers and Digital Signal Controllers with High-Speed PWM, USB and Advanced Analog*. Microchip Technology Inc. 2009-2012. 622 s.

[21] *Si864x Data Sheet*. Silicon Laboratories Inc. 2010-2019. 45 s.

[22] DFROBOT. *Micro Metal Geared motor w/Encoder - 6V 41RPM 380:1* [online].
[cit. 20-6-2020]. Dostupné z: <<https://www.dfrobot.com/product-1437.html>>.

8 SEZNAM ZKRATEK, OBRÁZKŮ A TABULEK

8.1 Seznam zkratek

CAN: Controller Area Network – sběrnice
USB: Universal Serial Bus – univerzální sériová sběrnice
I2C: Inter-Integrated Circuit – sériová sběrnice
SPI: Serial Peripheral Interface – sériové periferní rozhraní
UART: Universal asynchronous receiver-transmitter – druh komunikačního protokolu
SCSI: Small Computer System Interface – počítačové systémové rozhraní
PCI: Peripheral Component Interconnect – počítačová sběrnice
TX: Transmit – vysílací pin
RX: Receive – přijímací pin
MOSI: Master Output/Slave Input – linka SPI
MISO: Master Input /Slave Output – linka SPI
SCLK: Serial Clock – linka SPI
SCK: Serial Clock – linka SPI
SS: Slave Select – linka SPI
CS: Chip Select – linka SPI
SSEL: Slave Select – linka SPI
SDA: Serial Data Line – linka I2C
SCL: Serial Clock Line – linka I2C
ACK: Acknowledge – bit pro potvrzení přenosu
MD5: Message-Digest algorithm – rodina hašovacích funkcí
SHA: Secure Hash Algorithm – hašovací funkce
CRC: Cyclic Redundancy Check - Cyklický redundantní součet – hašovací funkce
TWI: Two Wire Interface – dvoudrátové rozhraní
PWM: Pulse Width Modulation – pulzně šířková modulace
DC: Direct Current - Stejnoseměrný elektrický proud
DIR: Direction – Směr
DIS: Disable – pin k zablokování motoru
GPIO: General Purpose Input/Output - Univerzální vstupní/výstupní pin
JTAG: Joint Test Action Group – standart pro testování plošných spojů
TQFP: A thin quad flat pack – pouzdro mikrokontroléru
ESD: Electrostatic discharge - elektrostatický výboj
ASCII: American Standard Code for Information Interchange – kódová tabulka
QEI: Quadrature Encoder Input – Data z rotačního enkodéru
PI: Proporcionálně-integrační regulátor

8.2 Seznam obrázků

Obrázek 1 - Sériová komunikace [7].....	11
Obrázek 2 - Paralelní komunikace [7].....	11
Obrázek 3 - UART komunikace [9].....	12
Obrázek 4 - SPI komunikace - Zapojení do smyčky [11].....	14
Obrázek 5 - SPI komunikace - Zapojení každého zařízení zvlášť [11].....	14
Obrázek 6 - I2C komunikace [14].....	15
Obrázek 7 - Struktura zprávy u I ² C komunikace [14].....	16
Obrázek 8 - Algoritmus kontrolního součtu Fletcher [16].....	17
Obrázek 9 - Cyklický redundantní součet[18].....	18
Obrázek 10 - Galvanické oddělení obvodů.....	21
Obrázek 11 - Převodník UART-USB.....	22
Obrázek 12 - Plug-in deska s mikrokontrolérem.....	23
Obrázek 13 - Deska s výstupy k jednotlivým perifériím.....	24
Obrázek 14 - Finální podoba řídicí jednotky.....	24
Obrázek 15 - Konfigurace systému.....	26
Obrázek 16 - Komunikace s dálkovým ovládním.....	26
Obrázek 17 - Příjem dat z enkodérů.....	27
Obrázek 18 - Řízení motorů a servopohonů.....	27
Obrázek 19 - Nastavení UART komunikace.....	28
Obrázek 20 - Tvar zprávy při komunikaci mezi PC a mikrokontrolérem.....	29
Obrázek 21 - Příjem dat z počítače.....	29
Obrázek 22 - Blokové schéma regulace DC motoru.....	30
Obrázek 23 - Grafy regulované veličiny a akční veličiny.....	30

8.3 Seznam tabulek

Tabulka 1 - SPI linky.....	13
Tabulka 2 - I2C linky.....	14

9 SEZNAM PŘÍLOH

Elektronická příloha obsahuje:

- Elektronické schéma desky s mikrokontrolérem a návrh desky plošných spojů
- Schéma desky s výstupními obvody a návrh desky plošných spojů
- Řídicí program pro vozidlo Car4
- Program s UART komunikací mezi PC a mikrokontrolérem a regulací DC motoru