



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

KOMPLEXNÍ ŘÍDICÍ SYSTÉM MOBILNÍHO ROBOTU HEXOR II

Bakalářská práce

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídicí systémy
Autor práce: **Jan Kredba**
Vedoucí práce: Ing. Miroslav Holada, Ph.D.





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

COMPLEX CONTROL SYSTEM OF THE MOBILE ROBOT HEXOR II

Bachelor thesis

Study programme: B2612 – Electrical Engineering and Informatics
Study branch: 2612R011 – Electronic Information and Control Systems

Author: **Jan Kredba**
Supervisor: Ing. Miroslav Holada, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan Kredba**
Osobní číslo: **M12000058**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Komplexní řídicí systém mobilního robotu Hexor II**
Zadávací katedra: **Ústav informačních technologií a elektroniky**

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

1. Seznamte se s robotem Hexor II na pracovišti školitele, architekturou jeho řídicího systému a možnostmi jeho ovládání.
2. Seznamte se se stávajícím stavem problematiky ovládání mobilních robotů ve světě.
3. Navrhněte komplexní řídicí systém robotu s ohledem na jednoduchost jeho ovládání a informovanost uživatele o aktuálním stavu robotu. Dále navrhněte sofistikovanější způsob autonomního řízení, který by umožnil samostatné navádění robotu v prostoru.
4. Návrh realizujte a ověřte funkčnost navrženého systému.
5. V závěru práce diskutujte přínos nového řídicího systému oproti jeho stávajícímu řešení.

Rozsah grafických prací: Dle potřeby dokumentace
Rozsah pracovní zprávy: cca 30 stran
Forma zpracování bakalářské práce: tištěná/elektronická
Seznam odborné literatury:

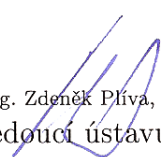
- [1] Petzold, Charles. Programování ve Windows. Praha : Computer Press, 199str. 121ISBN 80-7226-206-8
- [2] Novák, Petr. Mobilní roboty - pohony, senzory, řízení. Praha : BEN - technická literatura, 200str. 24ISBN 80- 7300-141-1.
- [3] STENZEL CORPORATION. HeXor? II The mobile robot for research and education: User's Guide Quick start. V2.0. Gliwice, 200Dostupné na CD dodávaném s robotem Hexor II

Vedoucí bakalářské práce: **Ing. Miroslav Holada, Ph.D.**
Ústav informačních technologií a elektroniky

Datum zadání bakalářské práce: **12. září 2014**
Termín odevzdání bakalářské práce: **15. května 2015**


prof. Ing. Václav Kopecný, CSc.
děkan




prof. Ing. Zdeněk Pliva, Ph.D.
vedoucí ústavu

V Liberci dne 12. září 2014

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: 15.5.2015

Podpis: 

Poděkování

Rád bych na tomto místě poděkoval mému vedoucímu práce Ing. Miroslavu Holadovi, Ph.D. za obětavý přístup při vedení této bakalářské práce. Dále bych chtěl poděkovat panu Ing. Lubomíru Slavíkovi, Ph.D. za zapůjčení laboratorního vybavení při testování zhotovených zařízení a za pořízení fotografií robotu, využitých v této práci.

Abstrakt

Předmětem této bakalářské práce je komplexní řídicí systém mobilního robotu HEXOR[®] II, především jeho rozšíření o nové metody navigace a celkové zvýšení jeho funkcionality. Aby toho bylo možné dosáhnout, byl upraven jak software robotu, tak i jeho hardware. Z hlediska hardwaru se tato práce zabývá především úpravou energetické jednotky a způsobem jejího monitorování. Samotný software se poté skládá z části nízkourovňového řízení, zajištěného mikroprocesorem robotu, a vysokourovňového řízení, jež obstarává osobní počítač. Zde je největší pozornost věnována vzájemné komunikaci těchto částí, odometrickým metodám pro navigaci robotu a samotnému uživatelskému rozhraní sloužícímu k jeho ovládání. Dále bylo provedeno modelování okolního prostředí robotu, za jehož pomoci byl vytvořen systém autonomního řízení.

Klíčová slova:

mobilní roboty, HEXOR[®] II, řídicí systémy, navigace, autonomní řízení

Abstract

The subject of this bachelor thesis is complex control system of the mobile robot HEXOR[®] II, mainly its extension by latest methods of navigation and overall increase in its functionality. To be able to achieve this, the software of the robot and also its hardware were adjusted. Concerning hardware, the thesis primarily handles the adjustment of energy unit and the way of its monitoring. Software itself then consists of parts of low-level control secured by the microprocessor of the robot, and high-level control done by personal computer. Here the thesis focuses on mutual communication between these parts, odometric methods for robot navigation and user interface itself serving for its control. Furthermore the modelling of outer surroundings of the robot was carried out, by means of which the system of autonomous control was created.

Keywords:

mobile robots, HEXOR[®] II, control systems, navigation, automous control

Obsah

Úvod	12
1 Mobilní robotika	13
1.1 Lokomoce mobilních robotů	13
1.2 Senzorika mobilních robotů	14
1.2.1 Mapování prostředí	15
1.2.2 Lokalizace	16
2 Řízení mobilních robotů	17
2.1 Architektury řídicích systémů	18
2.2 Repräsentace okolního prostředí	18
3 Mobilní robot HEXOR[®] II	20
3.1 Architektura řídicího systému	21
3.2 Senzorika robotu	21
4 Energetická jednotka robotu	24
4.1 Zdroj elektrické energie	24
4.2 Integrovaný nabíjecí modul	26
5 Řídicí systém robotu	30
5.1 Stav robotu a okolního prostředí	31
5.1.1 Monitorování stavu baterie	32
5.1.2 Odometrie	35
5.1.3 Monitorování okolního prostředí	37
5.2 Komunikace	38
5.2.1 Komunikace s osobním počítačem	38
5.2.2 Komunikace se senzorickými moduly	42
5.3 Řídicí software pro PC	44
5.3.1 Ovládání robotu	47
5.3.2 Mapa prostředí	48
5.4 Autonomní řízení	49
6 Závěr	52

A	Obsah přiloženého CD	56
B	Schéma zapojení nabíjecího modulu	57
C	Schéma zapojení měřicího modulu	58

Seznam obrázků

3.1	Robot HEXOR [®] II ¹	20
3.2	Periferie robotu HEXOR [®] II ¹	22
3.3	Pracovní rozsah natočení modulu sonaru s kamerou [12]	23
4.1	Umístění starého (nahore) a nového (dole) akumulátoru na robotu . .	25
4.2	Nabíjecí charakteristiky Li-ion článku při nabíjecím proudu 1 C ² [11]	26
4.3	Funkční model integrovaného nabíjecího modulu na univerzální DPS .	27
4.4	Zatěžovací charakteristika integrovaného nabíjecího modulu	28
4.5	Zvlnění výstupního napětí nabíjecího modulu při proudu $I = 1 \text{ A}$. .	28
4.6	Účinnost nabíjecího modulu ve fázi konstantního napětí při napájecím napětí 12 V	29
4.7	Účinnost nabíjecího modulu ve fázi konstantního proudu při napájecím napětí 12 V	29
5.1	Základní struktura softwaru mikroprocesoru robotu (původní řešení) [10]	31
5.2	Závislost napětí Li-ion článku na stupni jeho vybití [2]	33
5.3	Funkční model modulu pro monitorování stavu baterie	34
5.4	Ilustrace volby globálního a lokálního souřadného systému	36
5.5	Kinematický model robotu pro pohyb středních nohou	36
5.6	Původní rámec pro komunikaci směrem od počítače k robotu [10] . .	39
5.7	Původní rámec pro komunikaci směrem od robotu k počítači [10] . .	39
5.8	Formát rámce komunikačního protokolu HB-12/C	40
5.9	Vývojový diagram algoritmu příjmu rámce	41
5.10	Formát rámce proprietárního komunikačního protokolu	43
5.11	Vývojový diagram algoritmu příjmu rámce od senzorických modulů .	44
5.12	Původní program pro ovládání robotu skrze PC [10]	45
5.13	Hlavní okno nového programu pro ovládání robotu skrze PC	45
5.14	Kompletní program pro ovládání robotu skrze PC	47
5.15	Vývojový diagram algoritmu výpočtu trasy robotu	51

Seznam tabulek

4.1	Srovnání parametrů jednotlivých typů akumulátorů [3][8][11]	25
5.1	Srovnání hodnot naměřených modulem s měřicí ústřednou	34
5.2	Zprávy pro vzájemnou komunikaci počítače a robotu	42

Seznam zkratek

AD	Analog Digital Converter, analogově-digitální převodník
C	nízkoúrovňový programovací jazyk
C#	vysokoúrovňový objektově orientovaný programovací jazyk
CCCV	Constant Current followed by Constant Voltage, metoda nabíjení akumulátorů zprvu konstantním proudem, následně konstantním napětím
CCD	Charge-Coupled Device, elektronická součástka pro snímání obrazu
CMOS	Complementary Metal–Oxide–Semiconductor, polovodičová technologie využívající kombinace P-MOS a N-MOS technologií
DC	Direct Current, označení stejnosměrného elektrického proudu
DPS	deska plošných spojů
GPS	Global Positioning System, globální polohovací systém užívaný k určení polohy na Zemi
HB-12/C	komunikační protokol vyvíjeny na Technické univerzitě v Liberci
I ² C	Inter-Integrated Circuit, multi-masterová počítačová sériová sběrnice
ID	identifikátor, zpravidla identifikační číslo nebo kód
IR	Infrared, elektromagnetické vlnění v infračervené oblasti spektra
JTAG	Joint Test Action Group, architektura Boundary-Scan pro testování plošných spojů a programování FLASH paměti
Li-ion	lithium-iontový akumulátor
Li-pol	lithium-polymerový akumulátor
log.	logická hodnota
PC	Personal Computer, osobní počítač
RISC	Reduced Instruction Set Computing, architektura mikrprocesorů s redukovanou instrukční sadou
RS232	sériové komunikační rozhraní
UART	Universal Asynchronous Receiver and Transmitter, asynchronní sériové komunikační rozhraní
USART	Universal Synchronous/Asynchronous Receiver and Transmitter, synchronní/asynchronní sériové komunikační rozhraní
USB	Universal Serial Bus, univerzální sériová sběrnice

Úvod

Mobilní roboty jsou v dnešní době hojně využívány v různých odvětvích lidské činnosti, kde usnadňují a často i nahrazují rutinní či méně významnou práci. Své uplatnění nacházejí nejen ve vojenství, při záchranných operacích, nebo při průzkumu neznámých míst, ale i ve firmách (např. autonomní robotické vozíky) nebo v běžných domácnostech (např. robotické vysavače či sekačky). Z důvodu stále rostoucího počtu těchto zařízení a nepřehledného množství jejich různých aplikací je nutné stále zdokonalovat řídicí systémy těchto robotů tak, aby bylo možné dosáhnout požadovaných výsledků co možná nejefektivněji.

K návrhu a vývoji nových algoritmů a metod řízení jsou využívány různé mobilní roboty, na kterých jsou tyto metody odzkoušeny a odladěny. Jedním z robotů určených k tomuto vývoji je i robot HEXOR[®] II. Od doby jeho vzniku však došlo k poměrně významnému pokroku v oblasti mobilní robotiky, a proto se stal původní řídicí systém nevyhovujícím právě pro tyto účely. Z této skutečnosti vyplývá nutnost aktualizace jeho řídicího systému, a to jak po stránce softwarové, tak i po stránce hardwarové.

Účelem této práce je vytvoření nového řídicího systému mobilního robotu HEXOR[®] II, jenž by reflektoval požadavky pro implementaci nových algoritmů řízení a umožnil maximální využití fyzických možností robotu. Z těchto důvodů je nutná úprava nejen softwaru, ale i hardwaru, aby byla umožněna rozšiřitelnost nového systému (např. senzorického subsystému robotu). Software řídicího systému je rozdělen do dvou částí – software pro nízkoúrovňové řízení (software mikroprocesoru robotu) a pro vysokoúrovňové řízení (software pro řízení robotu skrze osobní počítač). Vzhledem k maximální efektivitě návrhu je nutná úprava obou částí řídicího softwaru, přičemž pro samotné ovládání robotu a implementaci nových algoritmů je určena část vysokoúrovňového řízení.

1 Mobilní robotika

Roboty lze obecně rozdělit podle jejich schopnosti přemísťovat se v prostoru na stacionární (např. průmyslové roboty či manipulátory) a mobilní (např. robotická vozítka či průzkumné roboty). Mobilní roboty jsou schopny měnit svoji polohu v prostoru a čase, na rozdíl od stacionárních, které jsou fixně upevněny ke svému podstavci. Mobilní roboty jsou velmi často nasazovány v prostředích, kde není možná lidská přítomnost (průzkumné roboty), nebo v případech, kdy je samotná přítomnost člověku nebezpečná a ohrožovala by tak jeho zdraví (např. pyrotechnický robot). Další aplikace těchto robotů je pak možné vidět v průmyslu (např. robotická vozítka ve skladech), výzkumu, domácnostech (např. robotické vysavače) nebo v odvětví zábavního průmyslu. [9][10]

1.1 Lokomoce mobilních robotů

Způsob pohybu mobilních robotů lze charakterizovat několika základními parametry, jako jsou stabilita, druh kontaktu pohybového ústrojí s podkladem a typ prostředí, ve kterém se robot pohybuje. Dalším charakteristickým parametrem pohybu robotu je jeho holonomičnost. [9][15]

Z hlediska stability pohybu lze definovat počet a geometrii kontaktních bodů, těžiště robotu, statickou či dynamickou stabilitu nebo sklon terénu. Roboty se statickou stabilitou jsou stabilní v každém okamžiku svého pohybu a je tak možné je v libovolné chvíli zastavit, aniž by hrozil jejich pád. Mezi typické případy lokomočního ústrojí se statickou stabilitou patří například kolové podvozky s minimálně třemi koly, pásové podvozky nebo kráčející platformy s minimálně třibodovou oporou v každém okamžiku kroku. V případě pohybu robotu s dynamickou stabilitou dochází v průběhu pohybu k přesunu těžiště mimo jeho podstavu. Z tohoto důvodu není možné robot zastavit v libovolném okamžiku pohybu, neboť by hrozil jeho pád. Typickými zástupci robotů s tímto způsobem pohybu jsou skákající či kráčející roboty, u nichž chybí třibodová opora v průběhu kroku. [9][15]

Z hlediska druhu kontaktu lokomočního ústrojí s podkladem je možné určit kontaktní body, případně kontaktní plochy – jejich velikost a tvar, kontaktní úhel a tření mezi kontaktními body resp. plochami a podkladem. Počet kontaktních bodů resp. ploch a jejich úhel velmi úzce souvisí s již zmíněnou statickou a dynamickou stabilitou pohybu. Velikost tření poté do značné míry ovlivňuje schopnost pohybu robotu po nakloněné rovině, tedy jaký maximální úhel je robot schopen zdotat, a také má vliv na maximální možné zrychlení robotu. [15]

V případě typu prostředí, ve kterém se robot pohybuje, lze charakterizovat jeho strukturu a médium. Vlastnosti a konstrukce lokomočního ústrojí jsou velmi závislé na zvoleném médiu, a proto je možné mobilní roboty dělit na pohybující se po souši, ve vodě, ve vzduchu, ve vesmírném prostředí a hybridní. Roboty pohybující se po souši lze v závislosti na způsobu, jakým je možné dosáhnout změny polohy, rozdělit na kolové, pásové, kráčející, plazivé, šplhající, skákající a hybridní. V současné době je mezi mobilními roboty největší zastoupení kolových robotů, dále pak pásových a kráčejících. [14][15]

Roboty lze dále dle holonomičnosti dělit na holonomické a neholonomické. Holonomické roboty jsou schopny změnit směr svého pohybu bez nutnosti současného pohybu v ostatních osách. Pro dosažení holonomického pohybu je nutné, aby měl robot stejný nebo vyšší počet stupňů volnosti, než je celkový počet souřadnic nutných pro popis jeho polohy v prostoru. Neholonomický robot je poté takový, u něhož není možná změna směru pohybu bez současného pohybu v ostatních osách. Typickým zástupcem neholonomického typu je automobil, který není schopen měnit svoji orientaci bez současného pohybu vpřed či vzad. [9]

1.2 Senzorika mobilních robotů

Senzorický subsystém patří mezi základní subsystémy mobilních robotů, přičemž volba senzorů, jejich umístění a zpracování senzorických dat přímo předurčují možné použití robotu. Obecně lze senzory rozdělit dle měřených veličin na propioceptivní (interní) a exteroceptivní (externí). Proprioceptivní senzory slouží k diagnostice samotného robotu a jeho jednotlivých subsystémů, jako je např. monitorování stavu baterie, teplota řídicích obvodů či rychlost, zrychlení a poloha jednotlivých motorů. Rovněž je však lze použít i pro lokalizaci robotu v prostoru (např. pomocí odometrie), avšak často je nutné takto získané informace doplnit o data od exteroceptivních senzorů. Exteroceptivní senzory naproti tomu získávají data o okolním prostředí

a umožňují jeho modelování a vlastní lokalizaci robotu vzhledem ke globálnímu souřadnému systému. Mezi typické zástupce těchto senzorů patří taktilní či IR senzory, laserové skenery, sonary nebo kompas. [14][15]

Dále je možné senzory dělit dle principu získávání měřených dat na pasivní a aktivní. Pasivní senzory jsou založeny na získávání informace o měřené veličině pomocí přijaté energie od samotného měřeného systému. Mezi zástupce pasivních senzorů se řadí čidla teploty, CCD/CMOS kamery či dotykové taktilní senzory. Naproti tomu aktivní senzory vyzařují energii do okolního prostředí a zaznamenávají jeho reakci. Zpravidla se jedná o různé druhy vlnění, ať už mechanické (např. ultrazvuk) či elektromagnetické (např. světlo ve viditelné i neviditelné oblasti spektra). Běžně využívané jsou již zmiňované IR senzory, sonary, GPS či senzory pro laserovou triangulaci. [14][15]

1.2.1 Mapování prostředí

Pro mapování okolního prostředí je v robotice hojně využíváno jak senzorů aktivních, tak i pasivních. Nejjednodušším pasivním senzorem, umožňujícím detekci přítomnosti překážky, je dotykový taktilní senzor. Ten je však pouze schopen rozpoznat, zda se překážka v blízkosti robotu vyskytuje, či nikoliv. Navíc je jeho dosah značně omezený, zpravidla se jedná pouze o jednotky až desítky centimetrů, takže je schopen detekovat pouze překážku v těsné blízkosti robotu. Dále se z pasivních senzorů používají různé kamerové systémy poskytující značné množství informací pro modelování okolního světa. Často jsou tyto systémy navíc stereovizní, takže je možné vytvářet trojrozměrný model. [14][15]

Mezi často používané aktivní senzory patří IR senzory, sonary a laserové skenery. IR senzory jsou schopné nejen detekovat přítomnost překážky, ale také podat přibližnou informaci o její vzdálenosti, přičemž jejich dosah se pohybuje řádově v desítkách centimetrů. Tyto senzory jsou založeny na detekci odraženého infračerveného světla od překážky. Z důvodu robustnosti celého senzoru a jeho odolnosti proti rušení od okolního prostředí je vyslaný infračervený signál často amplitudově modulován. Mnohem přesnější informaci o vzdálenosti od překážky poskytují ultrazvukové senzory, neboli sonary. Tyto senzory využívají odrazu mechanického vlnění (ultrazvuku) od překážky, přičemž jejich přesnost se pohybuje řádově v centimetrech až milimetrech a jsou schopny měřit až do vzdálenosti několika metrů. Laserové skenery využívají, na rozdíl od sonarů, odrazu laserového paprsku, přičemž

jsou schopny dosáhnout vyšší přesnosti. Jelikož má laserový paprsek velmi malou divergenci (šíří se tedy pouze jedním směrem), nevzniká zde problém s přesností určení směru, ve kterém se překážka vyskytuje. Díky tomu je tento skener schopen zmapovat okolní prostředí robotu s velkým rozlišením. [14][15][16]

1.2.2 Lokalizace

K určení polohy robotu je možné využít jak senzorů exteroceptivních, tak i proprioceptivních. Proprioceptivní senzory se využívají zejména pro lokalizaci pomocí informací o změnách vnitřního stavu robotu (dead reckoning). Tyto senzory jsou však schopny podávat pouze informace o relativní poloze robotu, a proto je tento způsob vždy zatížen kumulativní chybou, kdy se veškeré chyby vzniklé během přepočtu dat od akčního subsystému sčítají. Nejčastěji užívanou metodou pro určení polohy, která je založená na tomto principu, je odometrie. Odhad polohy je zde prováděn v závislosti na informacích poskytovaných akčním subsystémem a jejich přepočtu dle kinematického modelu robotu. [9][14][16]

Mnohem přesnější informaci o poloze vzhledem ke globálnímu souřadnému systému poskytují senzory exteroceptivní, které jsou schopny podávat data jednak o relativní, ale i o absolutní poloze robotu. Nejčastěji se pro lokalizaci používají elektronické kompas, GPS moduly, trilaterální či triangulační systémy. GPS systémy se využívají při globální navigaci spíše u robotů pohybujících se ve venkovním prostředí, naopak triangulační systémy je možné často vidět ve vnitřním prostředí (průmyslové haly, sklady atd.). [9][14][16]

2 Řízení mobilních robotů

V principu je možné rozdělit metody řízení mobilních robotů do tří základních skupin - dálkově řízené, autonomní a kombinované. U dálkově řízených robotů lze rozlišit mezi teleoperovaným a teleprezenčním způsobem řízení. U obou způsobů robot striktně vykonává příkazy operátora, avšak u teleoperovaného způsobu má operátor přímo vizuální informaci o jeho okolním prostředí. U teleprezenčního způsobu řízení má operátor k dispozici pouze virtuální model okolí robotu. V praxi jsou však často tyto způsoby řízení doplněny o určité autonomní chování robotu, aby byla zajištěna jeho ochrana proti případnému poškození. [6][9][14]

Plně autonomní robot je schopen se pohybovat v prostoru bez jakéhokoliv zásahu operátora nebo jsou operátorem zadány pouze cíle, kterých má dosáhnout. Řídicí systém robotu poté sám vyhodnotí způsob, jakým bude cíle dosaženo. Architekturu těchto řídicích systémů lze rozdělit do tří základních typů - reaktivní architekturu, funkční dekompozici a hybridní architekturu. Oba způsoby řízení (plně řízený a autonomní) spojují různé kombinované formy, které využívají výhod obou dvou předchozích způsobů. [9][14]

Reaktivní architektura je založena na přímé reakci robotu na jeho bezprostřední okolí. Tento způsob je vhodný především pro řešení jednodušších úloh ve vysoce dynamickém prostředí. Robot přímo zpracovává aktuální naměřená data, přičemž si z nich nevytváří žádnou mapu okolí. Takto tedy může reagovat pouze na aktuální situaci a není schopen své jednání jakkoliv plánovat. Z tohoto důvodu není tato architektura vhodná pro složitější úlohy. [7][9]

Naproti tomu funkční dekompozice umožňuje plánování jednotlivých akcí díky vlastnímu modelu prostředí. Tento model může být v čase postupně zpřesňován a upravován získanými senzorickými daty. Díky tomu je možné řešit i ty nejkomplicovanější problémy mobilní robotiky. Takovýto systém však není příliš robustní a v dynamickém prostředí může selhat. Z těchto důvodů se ve většině případů používají architektury hybridní, které spojují výhody obou dvou zmíněných architektur. [7][9]

2.1 Architektury řídicích systémů

Řídicí systém mobilního robotu může mít mnoho podob - od jednoprocessorového systému kompletně umístěném na samotném robotu, až po distribuovaný systém mezi více zařízení. Jelikož je řídicím systémem přímo kontrolován akční a senzorický subsystém, musí být vždy jeho technologická část ovládající hardware umístěna na samotném robotu. Tato část však nemusí být nutně tvořena pouze jedinou výpočetní jednotkou, ale většinou je distribuována mezi více spolupracujících jednotek. Často je navíc společně s touto částí na robotu umístěn systém s dostatečným výpočetním výkonem pro implementaci elementárního chování robotu. Vyšší algoritmy řízení je možné opět implementovat přímo v systému samotného robotu či je distribuovat do jiných zařízení, jež budou s robotem komunikovat. [6][14]

Z důvodu stále rostoucích požadavků na výpočetní výkon při implementaci složitějších algoritmů řízení se ukazuje jako vhodné řešení využít distribuovaného výpočetního výkonu osobního počítače. Ten se může vzhledem k robotu chovat jako master, pokud potřebujeme počítačem kontrolovat chování robotu, či jako slave, pokud osobní počítač slouží pouze jako nástroj pro výpočet algoritmů. [6][14]

Celý systém může být navíc i distribuován mezi více robotů a dohromady tvořit tzv. multirobotický systém. Ty jsou schopné mnohem lépe a často elegantněji řešit složitější úlohy. K celému systému pak může být přistupováno jako k centralizovanému, distribuovanému či hybridnímu. U centralizovaného multirobotického systému je vždy jeden centrální prvek, který řídí komunikaci, koordinaci a kooperaci jednotlivých robotů. Díky tomu nejsou kladeny tak vysoké nároky na zpracování dat samotnými jednotkami. Hlavní nevýhodou tohoto přístupu je nenahraditelnost centrálního prvku, jelikož při jeho výpadku celý systém přestává fungovat. U distribuovaného přístupu je systém rozprostřen mezi všechny účastníky, takže výpadek jednoho člena neohrozí jeho stabilitu. Hybridní systémy opět spojují výhody obou přístupů. [9]

2.2 Repräsentace okolního prostředí

K reprezentaci okolního prostředí může být využito surových dat získaných jednotlivými senzory, nebo je z těchto dat okolní prostředí modelováno např. pomocí mapy. Surových naměřených dat je využíváno v případě reaktivní architektury, kdy je

využit postup „plánuj a jednej“. Z aktuálních naměřených dat se vypočítají jednotlivé akce, které robot následně provádí. Tato data se nikam dále neukládají a nevyužívají k modelování prostředí. Z toho důvodu není možné, aby robot své jednání dopředu plánoval. [9][15]

V případě využití funkční dekompozice, nebo hybridní architektury, je nutné, aby bylo okolní prostředí modelováno. Nejčastěji používaným způsobem, jak toho dosáhnout, je tvorba mapy prostředí ze získaných sensorických dat. Alternativně je možné předat mapu robotu z externího zdroje. Způsobů, jak prostředí pomocí mapy modelovat je mnoho, avšak v principu lze mapy rozdělit na sensorické, geometrické a topologické. [7][9]

Sensorická mapa prostředí je tvořena přímo samotnými sensorickými daty, která jsou pouze ukládána a vhodně interpretována. Nejčastěji se používá mřížka obsazenosti udávající pravděpodobnost výskytu překážky v dané buňce zvoleného rastru. Tento typ mapy je velmi jednoduchý na vlastní tvorbu, avšak je poměrně náročný na paměťový prostor a zpracování dalšími algoritmy. [9]

Geometrická mapa využívá k reprezentaci překážek geometrických primitiv. Při její tvorbě jsou již kladeny vyšší nároky na zpracování sensorických dat, výsledná mapa však využívá již vyšší úroveň abstrakce a snáze se zpracovává dalšími algoritmy. Z těchto map je možná tvorba např. grafu viditelnosti či Voronoiho diagramu využívaných při plánování trasy robotu. [7][9]

Nejvyšší míru abstrakce nad reálným prostředím představují topologické mapy. Ty jsou tvořeny grafem, přičemž uzly označují významná místa v prostoru a hrany způsob, jakým se robot mezi uzly přemísťuje. Tento typ mapy je ideální pro zpracování metodou grafu viditelnosti či grafu tečen, ale jeho tvorba vyžaduje poměrně značné výpočetní nároky. [7][9]

3 Mobilní robot HEXOR[®] II

Robot HEXOR[®] II byl původně navržen na Silesian University of Technology v Gliwici v Polsku, jeho výroby se poté ujala polská firma Stenzel Ltd. [10] Svou konstrukcí se řadí mezi šestinohé kráčející roboty, přičemž jeho základ je tvořen kovovou platformou oválného tvaru. Z hlediska svého pohybu patří mezi roboty holonomické se statickou chůzí, jelikož je schopen měnit svou orientaci v prostoru bez nutnosti současného pohybu vpřed či vzad, a zároveň se v žádném okamžiku kroku nedostává do nestabilní polohy díky neustálé tříbodové opoře. Ačkoliv má robot celkem šest nohou, jejich pohyb zajišťují pouze tři stejnosměrné analogové servomotory od firmy Hitec RCD USA, Inc. Každé servo tedy ovládá najednou dvě nohy. Takto mechanicky svázaný jsou vždy přední a zadní nohy na každé straně a prostřední nohy. Tí je dosaženo pouze jediného stupně volnosti pro každou jeho nohu, což může být v případě pohybu v náročnějším prostředí značně limitující. Proto je tento robot určen pro použití výhradně ve vnitřním prostředí budov. [12]



Obrázek 3.1: Robot HEXOR[®] II ¹

¹Fotografie užita se svolením autora Ing. Lubomíra Slavíka, Ph.D.

3.1 Architektura řídicího systému

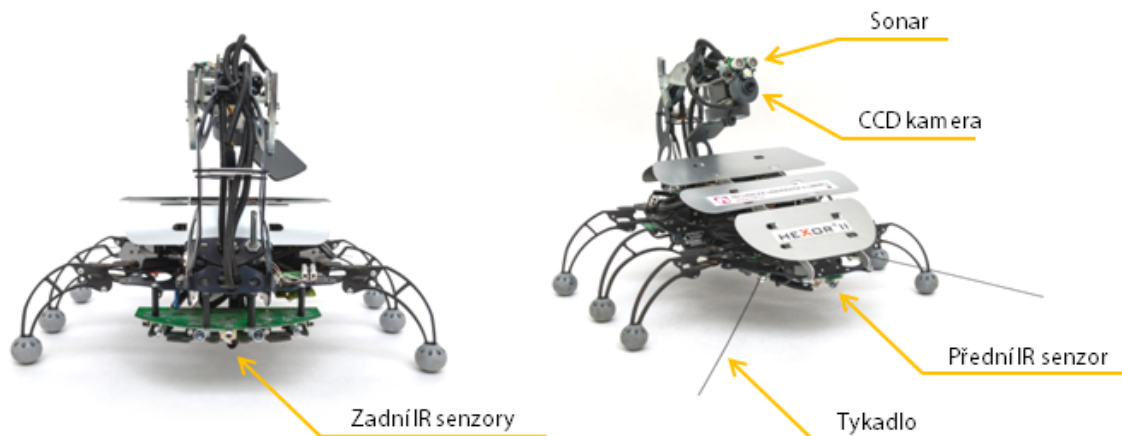
Řídicí systém robotu HEXOR[®] II je rozdělen do dvou základních částí – na část pro nízkoúrovňové řízení, obsluhující akční a senzorický subsystém robotu, a část pro vysokoúrovňové řízení, obstarávající složitější algoritmy řízení. O nízkoúrovňové řízení se stará 8bitový mikroprocesor typu RISC s Harvardskou architekturou ATmega128 od firmy Atmel Corporation, který je umístěn přímo na základní desce robotu. Mikroprocesor je taktován externím krystalem o frekvenci 14,7456 MHz a disponuje dostatečným výkonem pro implementaci základní reaktivní architektury. Mimo jiné je také vybaven dvěma rozhraními USART, podporou rozhraní JTAG či sběrnice I²C. [6][10][12]

Vysokoúrovňové řízení je možné částečně implementovat přímo v mikroprocesoru robotu, nebo jej lze distribuovat do jiných zařízení komunikujících s robotem. Pro implementaci složitějších způsobů řízení, jako je např. funkční dekompozice, již samotný mikroprocesor nedisponuje dostatečným výkonem a je nutné využít distribuovaného výpočetního výkonu. Proto jsou algoritmy vysokoúrovňového řízení implementovány v osobním počítači, jímž je robot ovládán. [6][10][12]

Z hlediska priority tvoří osobní počítač nadřazený řídicí subjekt (master) jemuž je robot podřízen (slave). Komunikace mezi nimi je zajištěna bezdrátovými moduly HM-TR/TTL pracujícími na frekvenci 433 MHz. Přenosová rychlost těchto modulů činí 9600 Bd a komunikace probíhá v poloduplexním režimu. Moduly komunikují s mikroprocesorem skrze rozhraní UART, přičemž na straně počítače je tento modul doplněn o převodník z USB na UART využívající integrovaného obvodu MCP2200. [10]

3.2 Senzorika robotu

Robot HEXOR[®] II je vybaven mnoha různými periferiemi, jež mu umožňují monitorovat své okolí. Jedná se o dva dotykové taktilní senzory, pět infračervených senzorů, jeden ultrazvukový senzor a CCD kameru. Jejich umístění na samotném robotu je uvedeno na obrázku 3.2. [10][12]



Obrázek 3.2: Periferie robotu HEXOR[®] II ¹

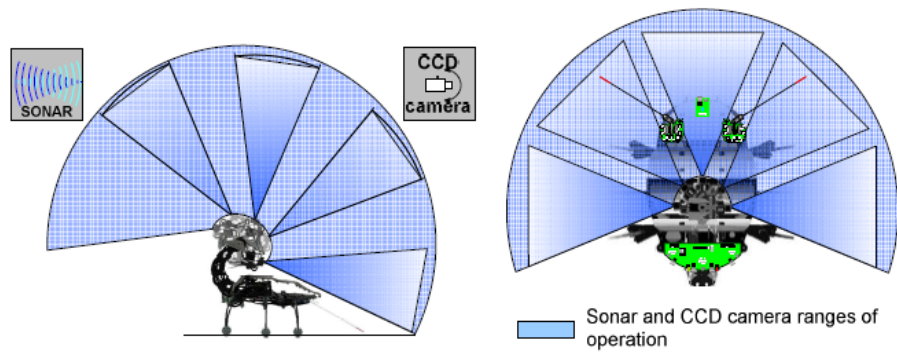
Dotykové taktilní senzory (tykadla), umístěné v přední části, jsou pasivní senzory sloužící k detekci překážek v těsné blízkosti robotu. Při zjištění překážky lze rozlišit, zda se vyskytuje na levé či pravé straně, ale určit, zda došlo k aktivaci senzoru nárazem zepředu či zezadu, již není možné. K pouhému určení přítomnosti překážek v blízkosti robotu dále slouží aktivní IR senzory. V zadní části je robot osazen celkem čtyřmi a v přední části jedním IR senzorem. Jejich dosah se pohybuje mezi 20 až 30 cm a vysílaný signál je z důvodu odolnosti proti rušení amplitudově modulován frekvencí 38 kHz. Oba typy senzorů jsou umístěny na samostatném modulu, který je připojen k základní desce robotu. [10][12]

Ultrazvukový senzor je společně s kamerou umístěn na pohyblivém konci „oca-su“ robotu. Pohyb celého modulu zajišťují celkem dva analogové stejnosměrné servomotory, přičemž pracovní rozsah natočení tohoto modulu je vyobrazen na obrázku 3.3. Ultrazvukový senzor je schopen měřit vzdálenost v rozsahu od 3 cm do 3 m s přesností na 1 cm. V praxi se však s touto přesností nedá zcela počítat, jelikož naměřená vzdálenost velmi závisí na tvaru překážky a na jejím povrchu. Pro samotné měření využívá senzor emitování ultrazvuku o frekvenci 40 kHz, načež měří časový interval, po kterém je zaznamenána odražená vlna. Tento časový interval je reprezentován dobou, po které je na výstupu modulu sonaru log. 1. [10][12]

CCD kamera, na rozdíl od ostatních senzorů, není součástí základního senzorickeho subsystému robotu, a tudíž není ani připojena k základní desce. Pro přenos dat využívá samostatného analogového bezdrátového modulu pracujícího na frekvenci 2,4 GHz. S obrazovými daty tedy není schopen robot sám pracovat, ale musí být využito dalšího zařízení, které daná data zpracuje a výsledek pře pošle zpět robotu.

¹Fotografie užita se svolením autora Ing. Lubomíra Slavíka, Ph.D.

Stejně jako s ultrazvukovým senzorem, je možné s kamerou pohybovat, a to jak v horizontálním, tak i vertikálním směru (viz obrázek 3.3). [10][12]



Obrázek 3.3: Pracovní rozsah natočení modulu sonaru s kamerou [12]

4 Energetická jednotka robotu

Ke svému napájení využívá robot HEXOR[®] II vlastní akumulátor, jenž je umístěn v jeho spodní části [12]. Jelikož se tedy jedná o bateriově napájené mobilní zařízení, je velmi důležité, aby tento akumulátor poskytl dostatek energie pro co nejdelší provoz robotu na jedno nabití. Toho je možné dosáhnout např. výběrem akumulátoru s velkou kapacitou, snížením odběru řídicích obvodů či snížením hmotnosti samotného robotu a tím i snížením zátěžného momentu působícího na jednotlivá serva. Vzhledem k tomuto požadavku byla původní koncepce energetické jednotky značně nevyhovující, neboť robot nebyl na jedno nabití schopen pracovat déle než půl hodiny. Na tomto stavu se pochopitelně podepsalo i stáří robotu, avšak i výrobcem udaný původní rozsah $1,5 \div 3$ h [12] byl při jeho velké aktivitě sotva dosažitelný. Jako poměrně závažný nedostatek původního návrhu se také ukázalo připojení kovové kostry robotu ke kladnému napájecímu potenciálu.

4.1 Zdroj elektrické energie

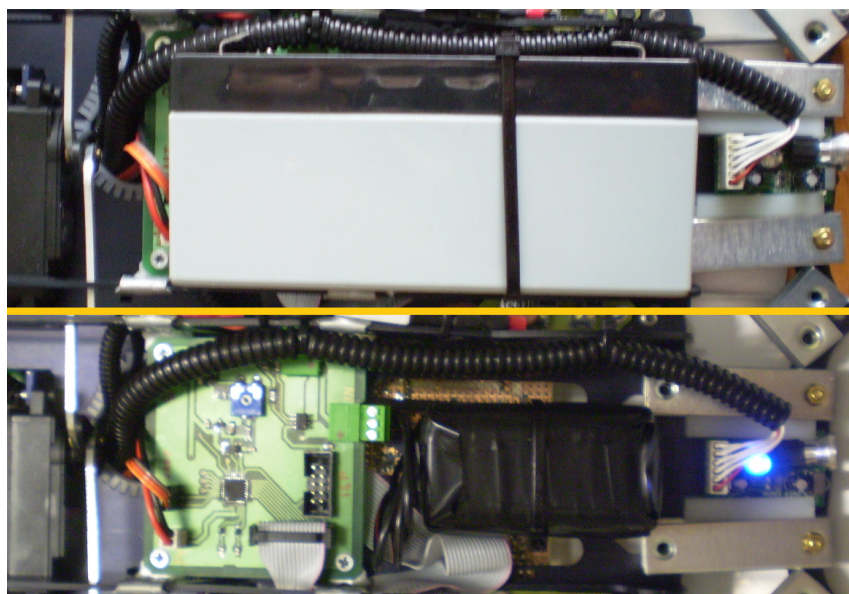
Jako napájecí zdroj byl výrobcem použit gelový olověný akumulátor o jmenovitém napětí 6 V a kapacitě 3,4 Ah [12]. Vzhledem ke svým rozměrům, hmotnosti a nebezpečí jeho zničení při úplném vybití, je však tento typ akumulátoru pro napájení mobilního robotu zcela nevhodný. Z běžně dostupných technologií vykazují pro tyto účely nejlepší vlastnosti (hustotu energie, míru samovybíjení, celkovou životnost) Li-ion a Li-pol akumulátory. Z důvodu vyšší dostupnosti a častého nasazení v mobilních zařízeních byl jako náhrada zvolen Li-ion akumulátor. Oproti původnímu typu disponují tyto akumulátory mnohem vyšší energetickou hustotou, a to jak objemovou, tak i hmotnostní. Navíc vykazují pouze minimální samovybíjení (cca 5 %/měsíc). Srovnání parametrů jednotlivých typů akumulátorů je pro úplnost uvedeno v tabulce 4.1. [3][8][11]

K napájení robotu byl konkrétně použit Li-ion akumulátor o jmenovitém napětí 7,4 V a kapacitě 4,5 Ah. Ten je složen celkem ze čtyř článků CGR18650CG značky

Tabulka 4.1: Srovnání parametrů jednotlivých typů akumulátorů [3][8][11]

Typ akumulátoru	Hustota energie		Životnost		Samovybíjení
	[Wh/kg]	[Wh/l]	cyklů	let	%/měsíc
Olověný	30÷50	70÷120	50÷1000	3÷5	20÷30
Ni-Cd	40÷60	80÷130	500÷2000	3÷10	10÷20
Ni-MH	60÷80	150÷200	300÷600	2÷5	15÷25
Li-ion	90÷120	160÷200	1000	5÷10	5÷8
Li-pol	150	220	1000	5÷10	2

Panasonic se sérioparalelním uspořádáním (dva články v sérii, dva paralelně). Touto výměnou bylo dosaženo snížení celkové hmotnosti robotu o cca 400 g, úspory místa patrného z obrázku 4.1 a zvýšení doby provozuschopnosti na jedno nabití až na cca 3 h při jeho průměrné aktivitě. Veškeré řídicí obvody jsou napájeny z DC/DC měniče, který je schopen pracovat se vstupním napětím v rozsahu 4,5 ÷ 9 V. Díky tomu nebylo nutné napětí akumulátoru nijak upravovat, i když se jeho jmenovitá hodnota zvýšila z původních 6 V na 7,4 V. Mimo samotné výměny akumulátoru byla provedena úprava elektrického zapojení robotu. Kovová kostra byla nově připojena k zápornému pólu akumulátoru tak, aby chránila jeho řídicí obvody proti vnějšímu elektromagnetickému rušení.

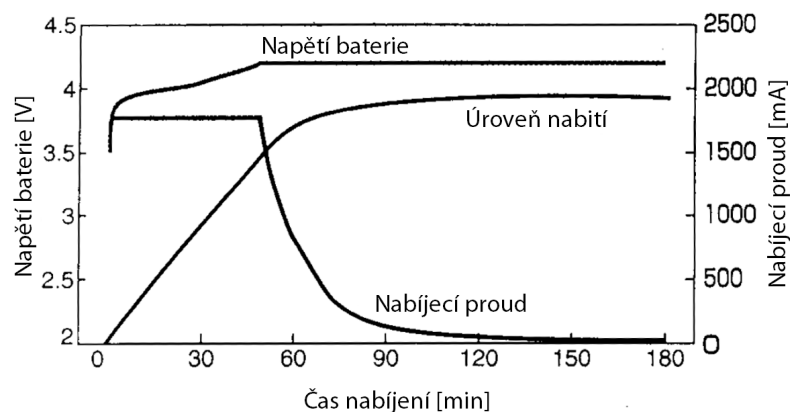


Obrázek 4.1: Umístění starého (nahore) a nového (dole) akumulátoru na robotu

4.2 Integrovaný nabíjecí modul

Jelikož původní nabíječ, primárně určený pro nabíjení olověných akumulátorů, vykazoval nepříznivý průběh nabíjecích charakteristik pro akumulátory typu Li-ion, bylo nutné vytvořit nabíječ nový. Z důvodu častého užití robotu pro prezentační účely je vhodné, aby jej bylo možné nabíjet i z alternativních zdrojů elektrické energie, jako je např. zásuvka osobního automobilu či možné nasazení solárních článků. S ohledem na tyto požadavky byl vyvinut integrovaný nabíjecí modul, jenž je umístěn přímo na samotném robotu, umožňující nabíjení z libovolného zdroje elektrického napětí v rozmezí 12 ÷ 15 V, resp. 12 ÷ 20 V v případě využití součástek s vyšší napětovou odolností. Modul je primárně určen k nabíjení jednoho či dvou sériově řazených Li-ion článků, přičemž je schopen poskytnout nabíjecí proud až 2,5 A.

Pro optimální nabíjení vyžadují články typu Li-ion průběh typu CCCV [11]. Zpočátku jsou články nabíjeny konstantním proudem (zpravidla v rozmezí 0,5 ÷ 1 násobku jmenovité kapacity baterie) a po dosažení maximálního napětí jsou dobíjeny konstantním napětím. Nabíjení je ukončeno při poklesu proudu pod hodnotu 0,05 C² [2]. Průběh nabíjecích charakteristik je uveden na obrázku 4.2.



Obrázek 4.2: Nabíjecí charakteristiky Li-ion článku při nabíjecím proudu 1 C² [11]

Jádro celého modulu je tvořeno integrovaným obvodem LM 2576T-ADJ, jenž společně s dalšími pasivními součástkami realizuje step-down konvertor pracující na frekvenci 52 kHz. Tímto konvertorem je následně přímo nabíjen akumulátor. Samotný integrovaný obvod je tvořen zpětnovazebním regulačním obvodem, regulujícím napětí na svém vstupu označeném jako *FEEDBACK* na hodnotu 1,23 V [5]. Velikost

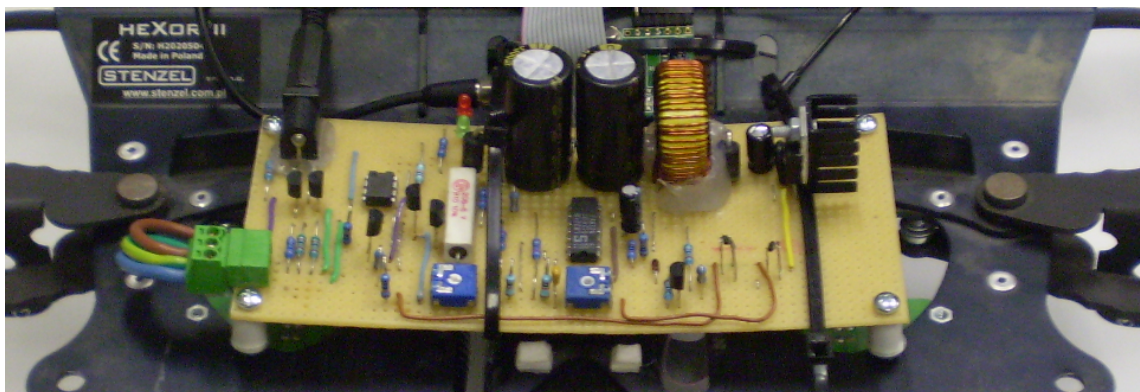
²Jednotka C představuje jmenovitou kapacitu baterie

nabíjecího proudu a konečného napětí je řízeno právě tímto vstupem integrovaného obvodu.

Při režimu konstantního napětí je na vstup regulátoru *FEEDBACK* přiveden pouze výstup napěťového děliče realizovaného mezi výstupem modulu a zemí. Jeho poměr přímo určuje výstupní napětí dle vztahu $U = k \cdot 1,23 V + 1,23 V$, kde k je poměr děliče. Režim konstantního proudu je realizován měřením úbytku napětí na rezistoru vřazeném do série s výstupem modulu. Toto napětí je diferenčním zesilovačem zesíleno a následně zpět pomocí nastavitelného děliče ve vhodném poměru sníženo. Tím je umožněno nastavení velikosti konstantního proudu a zároveň výstup operačního zesilovače není nucen pracovat v krajních polohách. Vstup *FEEDBACK* je poté připojen přímo na výstup tohoto děliče.

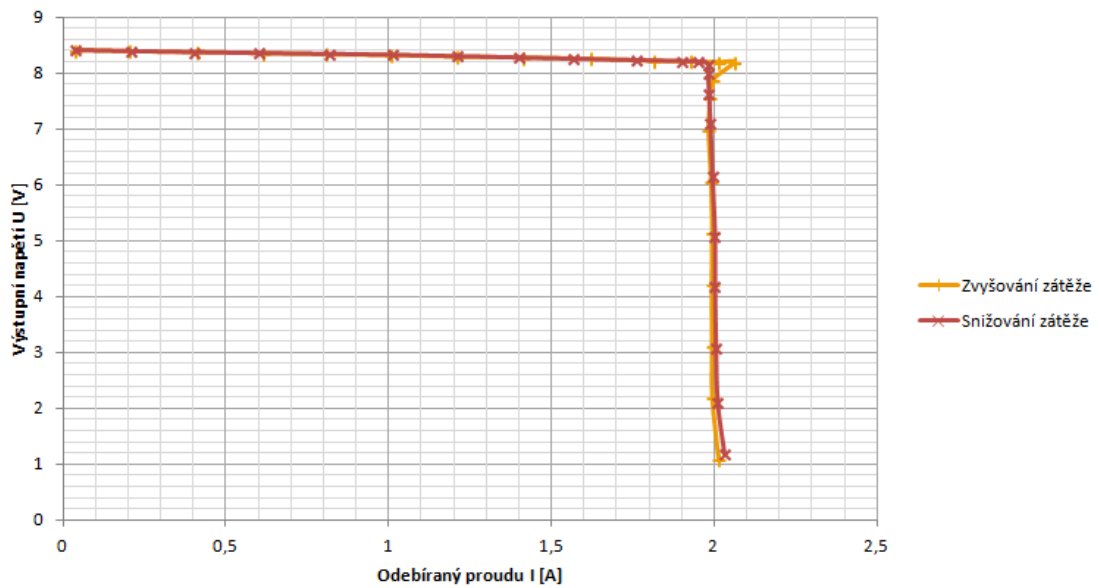
Kromě samotného nabíjení je modul schopen zajistit i balancování článku maximálním proudem 160 mA při jeho nabíjení. Balancér je realizován dvojicí komplementárních bipolárních tranzistorů, které jsou řízeny operačním zesilovačem pracujícím v režimu sledovače napětí. Jeho neinvertující vstup je připojen k děliči napětí, na jehož výstupu je přesně poloviční napětí nabíjeného akumulátoru. Invertující vstup je poté připojen na střední vodič baterie. Pokud má baterie v sobě již balancér zabudovaný, není nutné tento integrovaný během nabíjení využít. Schéma celého zapojení nabíjecího modulu je uvedeno v příloze B.

Pro samotný robot byl zkonstruován funkční model tohoto modulu, na němž byla ověřena jeho funkčnost, a zároveň byly změřeny jeho základní charakteristiky. Realizace modulu na univerzální desce plošných spojů je vyobrazena na obrázku 4.3. Naměřená zatěžovací charakteristika je uvedena na obrázku 4.4 a zvlnění výstupního napětí při výstupním proudu 1 A je vyobrazeno na obrázku 4.5.

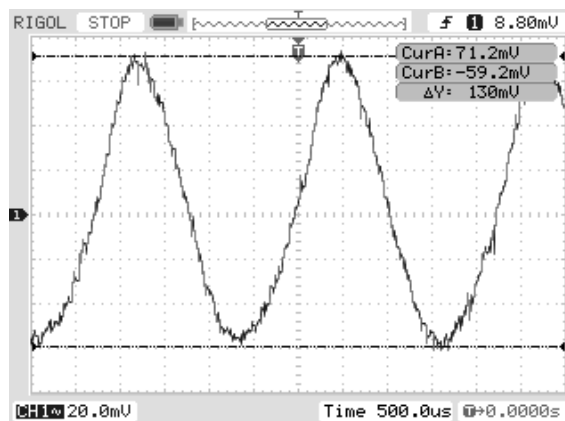


Obrázek 4.3: Funkční model integrovaného nabíjecího modulu na univerzální DPS

Zatěžovací charakteristika nabíjecího modulu



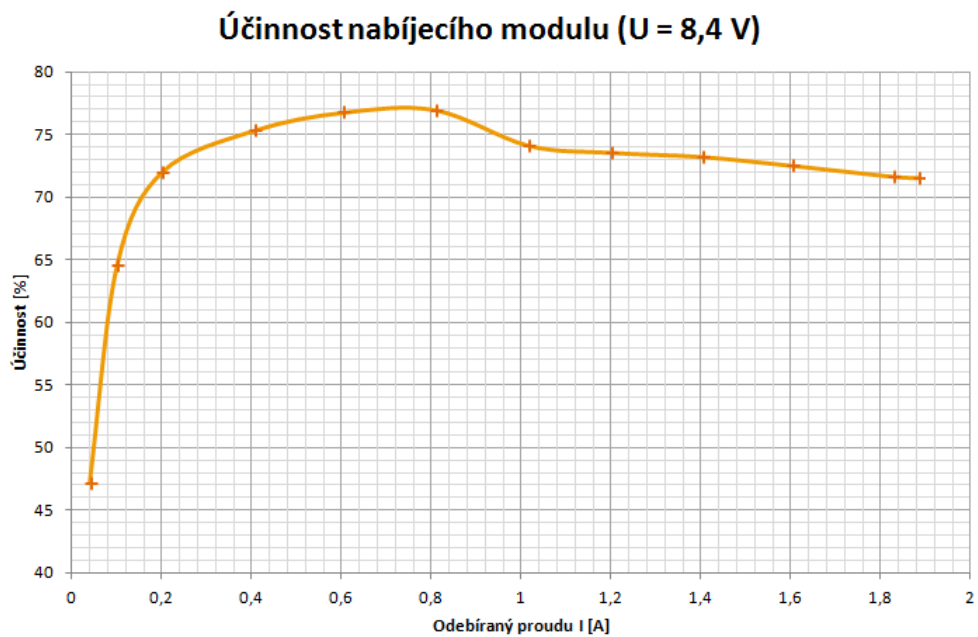
Obrázek 4.4: Zatěžovací charakteristika integrovaného nabíjecího modulu



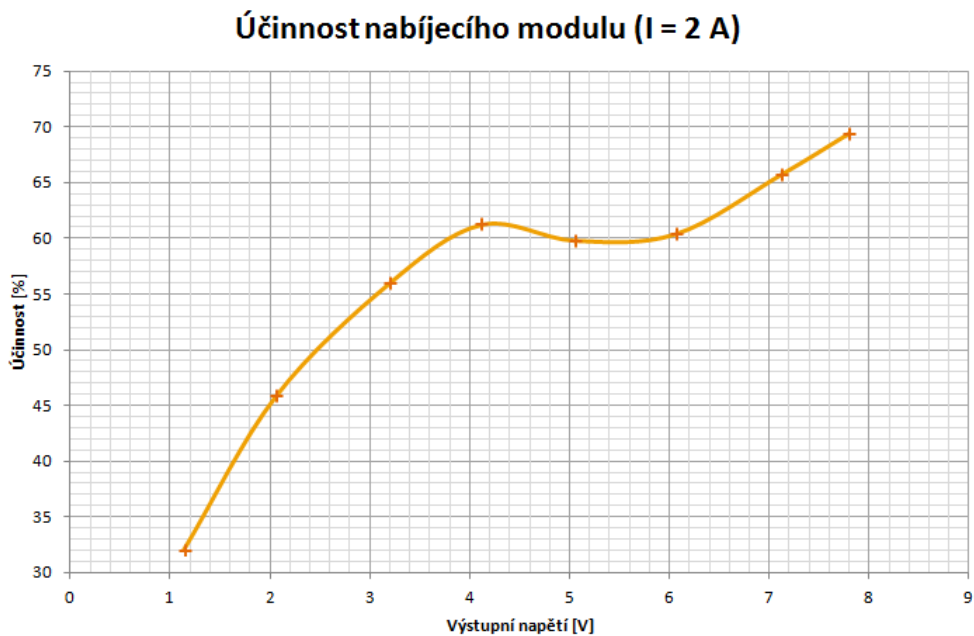
Obrázek 4.5: Zvlnění výstupního napětí nabíjecího modulu při proudu $I = 1 \text{ A}$

Z naměřené zatěžovací charakteristiky byl lineární regresí zjištěn vnitřní odpor nabíjecího modulu v režimu konstantního napětí jako $R_i = (102 \pm 4) \text{ m}\Omega$. Po odečtení systematické chyby měření, tvořené přechodovými odpory a odpory měřicích vodičů, byla tato hodnota přesněji určena jako $R_i = (55 \pm 7) \text{ m}\Omega$. Zmíněný vnitřní odpor má za následek snížení nabíjecího proudu při dosažení napětí blízkého maximálnímu napětí akumulátoru. Ačkoliv tento jev může negativně ovlivňovat dobu nabíjení, má pozitivní vliv na jeho bezpečnost, jelikož již malé překročení konečného nabíjecího napětí může způsobit poškození či zničení akumulátoru [2]. Účinnost tohoto modulu, v případě nabíjení akumulátoru napětím vyšším než 6 V, převyšuje

60 %. Maximální účinnosti 77 % dosahuje modul ve fázi nabíjení konstantním napětím při proudu 0,8 A, jak dokládají naměřené hodnoty uvedené v grafech na obrázcích 4.6 a 4.7.



Obrázek 4.6: Účinnost nabíjecího modulu ve fázi konstantního napětí při napájecím napětí 12 V



Obrázek 4.7: Účinnost nabíjecího modulu ve fázi konstantního proudu při napájecím napětí 12 V

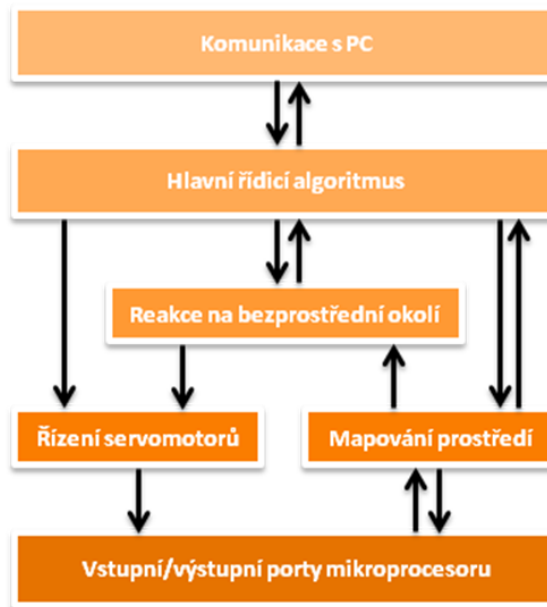
5 Řídicí systém robotu

Jak již bylo řečeno v kapitole 3.1, řídicí systém robotu HEXOR[®] II se skládá ze dvou základních částí – nízkoúrovňového řízení se základní reaktivní architekturou, implementovaného v mikroprocesoru robotu, a vysokoúrovňového řízení, jež je částečně implementováno v osobním počítači, umožňujícího různé způsoby ovládání robotu. Mikroprocesor obstarává samotné řízení servomotorů, sběr a základní zpracování senzorických dat a elementární reakce robotu na detekované překážky v jeho okolí. V osobním počítači je implementováno rozhraní pro ovládání robotu operátorem, zprostředkování informací o jeho stavu a okolním prostředí a probíhají zde výpočty složitějších algoritmů řízení.

Software mikroprocesoru je napsán v jazyce C a během předchozích prací byl již značně inovován. Jeho základní modulární struktura (viz obrázek 5.1) je vhodná pro další vývoj v oblasti řídicího systému, a proto byla z většiny ponechána. Vzhledem k úpravě hardwaru robotu, a potřebě získání více informací o jeho vlastním stavu, byla však i tato část doplněna o nové funkce. Nicméně komunikace s osobním počítačem již uspokojivě vyřešena nebyla, a proto byla navržena její nová koncepce spočívající především v implementaci nového komunikačního protokolu a v moderování vlastní komunikace. Oproti předchozímu návrhu, který explicitně neumožňoval příliš dobře využít distribuovaného výpočetního výkonu, byly navíc některé části hlavního řídicího algoritmu umístěny právě v osobním počítači.

Jednou ze základních činností mikroprocesoru je řízení servomotorů. Jelikož se jedná o analogová polohová serva, je nutné zajistit jejich pravidelné řízení s periodou 20 ms (odpovídá frekvenci 50 Hz). Proto je vlastní vykonávání algoritmů v mikroprocesoru rozděleno do dvou částí – obsluha servomotorů (2 ms) a vykonávání ostatních algoritmů řízení (18 ms), přičemž dochází k jejich pravidelnému střídání. Takto je zajištěno generování řídicích pulzů pro všechny servomotory najednou s potřebnou periodou [10]. Dále mikroprocesor zajišťuje sběr senzorických dat, která jsou vždy v pravidelném intervalu (200 ms) aktualizována a ukládána do globálních proměnných programu. Díky tomu je mohou využívat ostatní řídicí algoritmy, či

mohou být odeslána do osobního počítače. V závislosti na zvoleném režimu řízení jsou dále mikroprocesorem ovládnány reakce robotu na jeho bezprostřední okolí (např. couvnutí před překážkou). Hlavní řídicí algoritmus mikroprocesoru poté pracuje s naměřenými sensorickými daty, provedenými reakcemi na okolí robotu a zadanými cíli, z nichž určuje hodnoty pro řízení jednotlivých servomotorů.



Obrázek 5.1: Základní struktura softwaru mikroprocesoru robotu (původní řešení) [10]

5.1 Stav robotu a okolního prostředí

Jednou ze základních vlastností mobilního robotu je schopnost monitorování vlastního stavu a okolního prostředí. Monitorování okolního prostředí je zajištěno exteroceptivními senzory, jimiž je robot vybaven, konkrétně dvěma dotykovými taktilními senzory, pěti infračervenými a jedním ultrazvukovým senzorem, jak je uvedeno v kapitole 3.2. Původním sensorickým subsystémem však nebyl nijak monitorován stav samotného robotu a jeho vztah k okolnímu prostředí. Největším problémem, z hlediska vzdáleného řízení, byla absence informace o aktuálním stavu baterie a o poloze robotu v prostoru. Při návrhu nového řídicího systému byl tento problém reflektován návrhem modulu pro monitorování stavu baterie a odometrickým určením polohy robotu.

5.1.1 Monitorování stavu baterie

U bateriových zařízení je možné k určení aktuálního stavu baterie využít buď měření jejího napětí nebo z měření proudu vypočítat celkový přenesený náboj. První metoda je univerzálnější, nevyžaduje stálé napájení měřících obvodů, ale zároveň nedosahuje takové přesnosti, jako metoda druhá. U té je však nutné měřicí obvody neustále napájet a v případě výměny baterie je potřeba provést kalibraci. Při vývoji měřícího modulu bylo nutné reflektovat podmínku jeho nulového proudového odběru ve vypnutém stavu robotu, jelikož samotný robot je v některých případech uskladněn i na dobu delší než 2 měsíce, během které není nijak nabíjen. Dále je potřeba, aby celý systém fungoval i v případě výměny baterie a nebylo ho nutné na novou baterii kalibrovat. Z těchto důvodů bylo přistoupeno k metodě kombinující oba předešlé způsoby.

Aby mohlo být využito obou metod, je nutné měřit jak napětí baterie, tak i proud tekoucí z/do baterie. Samotné jádro modulu je tvořeno 8bitovým mikroprocesorem ATmega8 od firmy Atmel Corporation disponujícím 10bitovým AD převodníkem. Měření napětí je realizováno pouhým napěťovým děličem mezi kladným a záporným pólem baterie, přičemž je tento dělič spínán unipolárními tranzistory tak, aby byl zaručen nulový proudový odběr při vypnutém robotu. Výstup děliče je poté skrze dolnoproustní RC filtr přiveden na vstup AD převodníku mikroprocesoru. K měření protékajícího proudu je využit integrovaný obvod ACS712ELCTR-20A-T od firmy Allegro Microsystems, LLC využívající bezkontaktní měření integrovanou Halloovou sondou. Výstupem tohoto obvodu je napětí přímo úměrné měřenému proudu v rozsahu $-20 \div 20$ A s rozlišením 100 mV/A [1]. Pro účely měření proudu tekoucího z/do baterie je však původní rozsah příliš velký a naopak rozlišení pro vstup do převodníku příliš malé. Proto je tento napěťový výstup dále upraven diferenčním zesilovačem a přes dolnoproustní filtr opět přiveden na vstup AD převodníku mikroprocesoru. Komunikace modulu s mikroprocesorem robotu a vlastní přenos naměřených dat je poté zajištěn prostřednictvím UARTu. Schéma kompletního elektrického zapojení modulu je uvedené v příloze C.

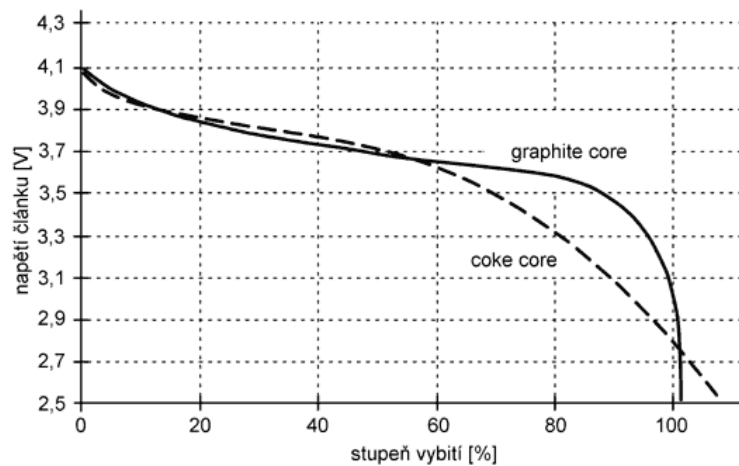
Vzhledem k možnému výskytu rušivých signálů od servomotorů, pracujících na frekvenci 50 Hz, je měření realizováno se vzorkovací frekvencí 100 Hz, čímž je dosaženo jejich maximálního potlačení. Jelikož je měřena pouze střední hodnota daných veličin, jsou naměřená data dále filtrována pomocí klouzavého průměru desátého řádu. Z takto vyfiltrovaných naměřených hodnot napětí a proudu je následně vypočtena úroveň nabití baterie. Pro maximální robustnost celého systému

je vypočtená úroveň nabití baterie opět filtrována, nyní již klouzavým průměrem stého řádu.

Pro základní identifikaci stavu baterie je využito naměřeného napětí společně s proudem tekoucím z/do baterie. Toto napětí je za pomoci známého vnitřního odporu baterie a proudu přepočítáno na ekvivalentní napětí nezatížené baterie dle vztahu 5.1. Z této hodnoty je poté, dle známého průběhu napětí v závislosti na stupni vybití baterie (viz obrázek 5.2), určena úroveň nabití dle vztahu 5.2 aproximujícího tuto závislost v rozsahu napětí $6,0 \div 8,4$ V.

$$U = U_{BAT} + R_{BAT} \cdot I \quad (5.1)$$

$$C_{BAT}[\%] = -18,667 V^{-3} \cdot U^3 + 406,72 V^{-2} \cdot U^2 - 2884,8 V^{-1} \cdot U + 6698,7 \quad (5.2)$$

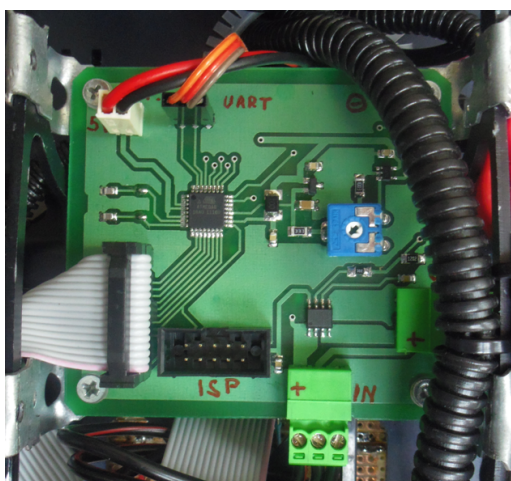


Obrázek 5.2: Závislost napětí Li-ion článku na stupni jeho vybití [2]

Takto zjištěná hodnota však slouží pouze jako výchozí, ze které je následně numerickou integrací protékajícího proudu vypočítávána aktuální úroveň nabití baterie. Aby došlo k ustálení všech měřených veličin, je výchozí hodnota určena až po uplynutí 30 s po zapnutí napájení. Do té doby jsou k dispozici pouze průběžné hodnoty z jednotlivých měření.

Pro monitorování stavu baterie na robotu HEXOR[®] II byl zhotoven funkční model tohoto modulu (viz obrázek 5.3), na němž byla odzkoušena jeho funkčnost. Výsledný modul je schopen měřit napětí v rozsahu $3 \div 10$ V a proud v rozsahu $-5 \div 5$ A. Aby byla zajištěna co nejlepší přesnost měření, byl zhotovený modul pro každou měřenou veličinu softwarově zkalibrován 10ti bodovou kalibrací. Pro úplnost bylo ještě provedeno srovnávací měření, kdy byly hodnoty naměřené tímto

modulem porovnávají s hodnotami naměřenými měřicí ústřednou Agilent 34410A s přesností o tři řády vyšší, než je teoretická maximální přesnost modulu. Srovnání těchto hodnot je uvedeno v tabulce 5.1.



Obrázek 5.3: Funkční model modulu pro monitorování stavu baterie

Tabulka 5.1: Srovnání hodnot naměřených modulem s měřicí ústřednou

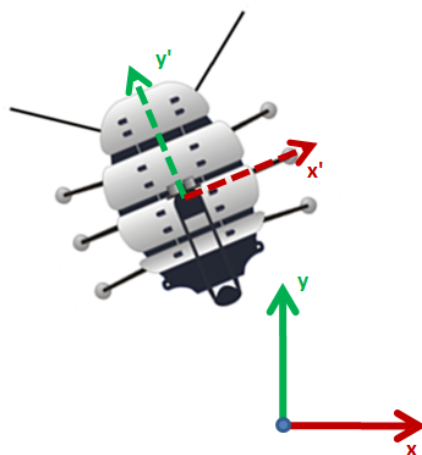
U [V] Agilent	U [V] modul	ΔU [V]	I [A] Agilent	I [A] modul	ΔI [A]
3,0003	3,00	-0,0003	-3,0021	-3,00	0,0021
3,5218	3,52	-0,0018	-2,0000	-2,01	-0,0100
4,0075	4,01	0,0025	-1,8013	-1,81	-0,0087
4,4911	4,49	-0,0011	-1,4031	-1,41	-0,0069
5,0129	5,01	-0,0029	-1,00045	-1,00	0,00045
5,5684	5,57	0,0016	-0,60319	-0,61	-0,00681
6,0384	6,04	0,0016	-0,20064	-0,20	0,00064
6,5395	6,54	0,0005	0,00035	0,00	-0,00035
7,0580	7,05	-0,0080	0,20050	0,20	-0,00050
7,4726	7,47	-0,0026	0,60082	0,60	-0,00082
7,9453	7,95	0,0047	1,00900	1,01	0,00100
8,5610	8,56	-0,0010	1,4021	1,40	-0,0021
9,0464	9,04	-0,0064	1,8023	1,80	-0,0023
9,5144	9,52	0,0056	2,0076	2,00	-0,0076
9,9321	9,93	-0,0021	3,0059	3,00	-0,0059

Z porovnání naměřených hodnot je patrné, že při měření napětí není reálně absolutní chyba vyšší jak 10 mV a v případě měření proudu vyšší jak 20 mA. Při posuzování přesnosti daného modulu je však nutné zohlednit i údaje udané výrobcem použitých součástek. Díky softwarové kalibraci byl odstraněn vliv nepřesnosti odporových děličů, napěťové reference a zesílení operačním zesilovačem. Odstraněna však nebyla kvantizační chyba převodníku a jeho nelinearita. Výrobcem mikroprocesoru je udána absolutní přesnost AD převodníku jako ± 2 LSB [4]. Dále je nutné zohlednit teplotní drift napěťové reference, jež dle výrobce činí ± 50 ppm/ $^{\circ}\text{C}$ [13]. Při provozu zařízení v rozmezí teplot $15 \div 35^{\circ}\text{C}$ pak tento rozdíl může činit až 0,1%, díky čemuž může dojít k ovlivnění naměřené hodnoty až o 0,11%. Pro naměřené napětí tedy bude platit přesnost $\pm(0,11\% \text{ of rdg.} + 2 \text{ dgt.})$. U měření proudu je nutné ještě zohlednit chybu samotného integrovaného obvodu, která je výrobcem udaná jako 1,5% při teplotě 25°C [1]. Z těchto údajů lze výslednou přesnost měření proudu stanovit jako $\pm(1,61\% \text{ of rdg.} + 2 \text{ dgt.})$.

5.1.2 Odometrie

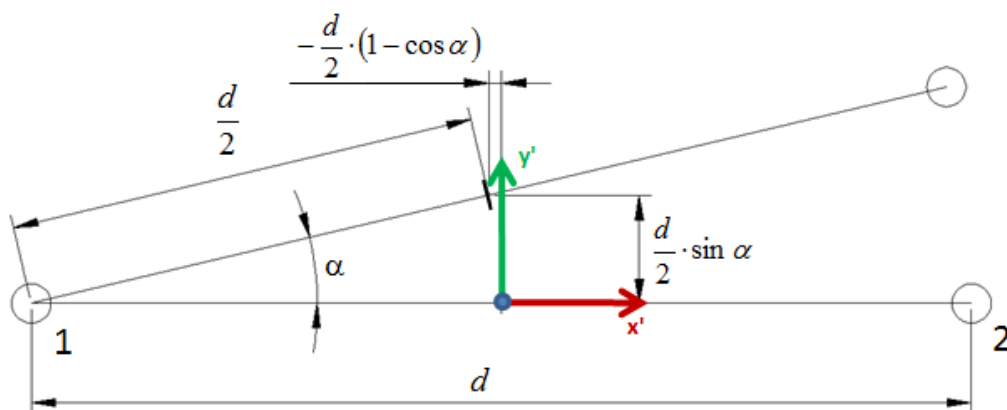
Určení polohy robotu vzhledem ke globálnímu souřadnému systému je jedním ze stěžejních problémů mobilní robotiky. Jak již bylo řečeno v kapitole 1.2.2, k jeho řešení je možné využít odometrické metody, založené na výpočtu polohy dle změn vnitřního stavu robotu, či je možné získávat údaje o poloze z exteroceptivních senzorů. Jelikož robot HEXOR[®] II nedisponuje exteroceptivními senzory schopnými určit jeho polohou v prostoru, byla využita možnost odometrického určení polohy. V případě dalšího rozšíření sensorického subsystému robotu by navíc tato metoda mohla doplňovat data z absolutních senzorů pro získání co nejlepší informace o poloze robotu.

Odometrické metody umožňují pouze relativní navigaci, tedy určení změny polohy vzhledem k předchozí poloze robotu [14]. Z toho důvodu byl jako globální souřadný systém zvolen počáteční lokální souřadný systém robotu. Počátek globálního systému je tedy shodný s počáteční polohou robotu. Lokální souřadný systém je umístěn v jeho těžišti, přičemž osa y vždy směřuje směrem vpřed. Jelikož se robot HEXOR[®] II pohybuje pouze po pevné zemi, lze jeho operační prostor zjednodušit na dvoudimenzionální. Všechny zavedené souřadné systémy jsou pravotočivé a jejich ilustrace je uvedena na obrázku 5.4.



Obrázek 5.4: Ilustrace volby globálního a lokálního souřadného systému

Samotná odometrie je implementována přímo v mikrokontroléru robotu, jelikož je nutné informaci o poloze průběžně aktualizovat i v případě krátkého výpadku komunikace. Navíc by byl komunikační kanál zbytečně zahlcován daty, která je možné zpracovat samotným robotem. Z důvodu absence jakýchkoliv proprioceptivních senzorů, jež by umožňovaly získat informaci o změně polohy, jsou k jejímu určení využita data o robotem vykonaných krocích. Vlastní pohyb je vždy realizován nadzvednutím jedné strany robotu, na které se opírá pouze o svou prostřední nohu, kolem které se následně pohybem nohou na druhé straně pootočí. Samotné těžiště robotu je umístěno přímo nad těmito nohama, což mu zajišťuje dobrou stabilitu. Z těchto údajů byl sestaven kinematický model, jenž je pro prostřední nohy vyobrazen na obrázku 5.5.



Obrázek 5.5: Kinematický model robotu pro pohyb středních nohou

Z tohoto modelu lze určit změnu polohy těžiště v závislosti na úhlu pootočení dle vztahů 5.3 a 5.4, kde n představuje číslo nohy, kolem které se robot otáčí. Transformaci těchto souřadnic do globálního souřadného systému je možné provést dle vztahů 5.5 a 5.6, kde proměnné s indexem 0 označují původní souřadnice, index 1 značí souřadnice nové a φ představuje celkový úhel rotace. Po určení souřadnic polohy robotu je ještě nutné aktualizovat jeho nový směr rotace. Změna úhlu celkové rotace je totožná s úhlem, o který se robot pootočil kolem své nohy, a tudíž je možné tento úhel pouze přičíst k aktuální hodnotě, jak je uvedeno i ve vztahu 5.7.

$$\Delta x' = (-1)^n \cdot \frac{d}{2} \cdot (1 - \cos \alpha) \quad (5.3)$$

$$\Delta y' = (-1)^{n+1} \cdot \frac{d}{2} \cdot \sin \alpha \quad (5.4)$$

$$x_1 = x_0 + \Delta x' \cdot \cos \varphi - \Delta y' \cdot \sin \varphi \quad (5.5)$$

$$y_1 = y_0 + \Delta x' \cdot \sin \varphi + \Delta y' \cdot \cos \varphi \quad (5.6)$$

$$\varphi_1 = \varphi_0 + \alpha \quad (5.7)$$

Tento odometrický způsob určení polohy je možné využít v případě, že je zajištěna dobrá adheze kontaktních ploch robotu s podkladem a nedochází tedy k prokluzu. Při nedodržení těchto předpokladů hrozí vlivem kumulace chyb při relativní navigaci naprosté znehodnocení celé informace. Z těchto důvodů je nutné zajistit robotu co nejkvalitnější povrch. Dále robot není schopen řešit násilné přemístění obsluhou, jelikož nemá žádné prostředky, kterými by tuto změnu polohy zjistil. Proto by bylo vhodné celý systém dále doplnit o absolutní senzory polohy, které by pomáhaly tyto údaje o poloze zpřesňovat.

5.1.3 Monitorování okolního prostředí

K monitorování svého okolního prostředí robot disponuje ultrazvukovým senzorem, pěti infračervenými a dvěma dotykovými taktilními senzory (viz kapitolu 3.2). Z hlediska využití se k tvorbě modelu okolního prostředí jednoznačně nejvíce uplatní ultrazvukový senzor, neboť je na rozdíl od ostatních schopen měřit i vzdálenost od překážek, a to v různých směrech. Poloha detekovaných překážek je robotem určována vždy vzhledem k lokálnímu souřadnému systému, což je výhodné v případě

lokální navigace. Toho je využito např. při jednoduchém autonomním řízení, kdy robot vždy vybírá cestu s nejdelší vzdáleností od překážky. Prostředí je zde modelováno jednorozměrným polem, jehož jednotlivé prvky obsahují informaci o vzdálenosti k nejbližší překážce v daném směru. Takto je vytvořena elementární mapa blízkého okolí, kdy poloha prvku v poli, společně s naměřenou vzdáleností, poskytuje informaci o poloze překážky v polárních souřadnicích (poloha prvku představuje úhel, naměřená hodnota pak vzdálenost od počátku lokálního souřadného systému).

K základní detekci překážek v těsné blízkosti robotu je možné využít infračervených a taktilních sensorů. Ty poskytují pouze informaci o přítomnosti překážky, a proto jsou v celém systému použity spíše k druhotné kontrole průchodnosti prostředím. Díky specifickému umístění na robotu lze rozhodnout, ve které oblasti se pravděpodobně překážka nachází a určit tak adekvátní reakci. Tyto senzory jsou využívány především v základní reaktivní architektuře, která je implementována přímo v mikroprocesoru robotu.

5.2 Komunikace

Jak již bylo řečeno v kapitole 3.1, řídicí systém robotu HEXOR[®] II je složen z části nízkourovňového řízení, implementovaného v mikroprocesoru robotu, a vysokoúrovňového řízení, které je ponecháno nadřazenému řídicímu subjektu, jímž je osobní počítač. Vzájemná komunikace mezi nimi je zajištěna bezdrátovými komunikačními moduly. Její původní způsob byl však nevhodně navržen, jelikož neumožňoval zajistit spolehlivost přenosu dat, nebylo možné odesílat rámce s proměnnou délkou a dokonce ani nebyly řešeny kolize při současném vysílání obou zařízení. Z těchto důvodů byla navržena zcela nová koncepce vzájemné komunikace robotu s osobním počítačem. Kromě toho bylo ale nutné zajistit přenos dat mezi mikroprocesorem robotu a dalšími sensorickými moduly. K tomuto účelu bylo využito UARTu společně s proprietárním komunikačním protokolem, speciálně vyvinutým pro sběr dat od těchto modulů.

5.2.1 Komunikace s osobním počítačem

Komunikace robotu s osobním počítačem je realizována bezdrátovými komunikačními moduly HM-TR/TTL (viz kapitolu 3.1). Na straně robotu je výstup tohoto modulu připojen přímo k UARTu 1 mikroprocesoru ATmega128. Na straně druhé je

tento výstup ještě doplněn o převodník z UARTu na USB, který se poté v počítači chová stejně jako běžná sériová linka s rozhraním RS232. Díky tomu je s robotem možné komunikovat stejně, jako by byl připojen přímo skrze toto rozhraní.

Jelikož je komunikace bezdrátová a probíhá po sdíleném médiu, je umožněn pouze poloduplexní provoz. Proto je nutné celou komunikaci moderovat. K tomu bylo využito oprávnění jednotlivých zařízení vysílat (obdoba klasického tokenu známého např. ze sítí token ring). Primárně má toto oprávnění osobní počítač, který tak funguje jako master řídící celý průběh komunikace. Jakmile je vyžadována odpověď robotu, je mu toto oprávnění na omezenou dobu zapůjčeno. V tu dobu osobní počítač toto oprávnění ztrácí a je tak vyloučena vzájemná kolize. V případě, že robot odvysílá vše potřebné, přeposílá oprávnění zpět osobnímu počítači a ztrácí tak možnost odesílat data. Pokud by robot do stanoveného času neodpověděl, je mu oprávnění odebráno a počítač může opět vysílat.

Aby bylo možné určit příjemce odvysílaného rámce, a zároveň byla zajištěna spolehlivost jeho doručení, bylo nutné také upravit samotný komunikační protokol. Velmi závažným nedostatkem původního protokolu byla navíc blokáce celkem dvou hodnot bajtu, které se nesměly vyskytovat jinde než na začátku a konci celého rámce. Díky tomu nebylo možné přenášet libovolná binární data. Tento protokol nebyl navíc ani vybaven žádným mechanismem, který by kontroloval obdržená data, zda dorazila neporušená. Kromě těchto nedostatků bylo obtížné odesílání většího množství dat, jelikož každý rámeček měl přesně stanovenou délku, kterou nebylo možné měnit. Formát rámečků původního komunikačního protokolu je uveden na obrázcích 5.6 a 5.7.

Začátek rámce (255)	ID příkazu	Rozkaz 1	Parametr 1	Rozkaz 2	Parametr 2	Konec rámce (254)
------------------------------------	-------------------	-----------------	-------------------	-----------------	-------------------	----------------------------------

Obrázek 5.6: Původní rámeček pro komunikaci směrem od počítače k robotu [10]

Začátek rámce (99)	ID příkazu	Parametr	Konec rámce
-----------------------------------	-------------------	-----------------	------------------------

Obrázek 5.7: Původní rámeček pro komunikaci směrem od robotu k počítači [10]

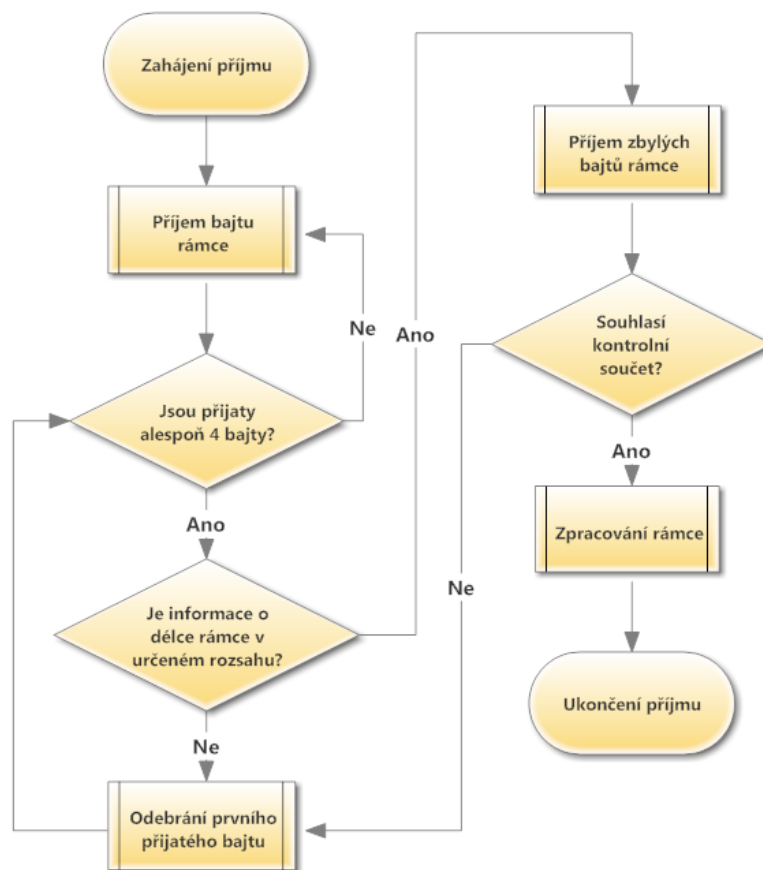
Pro splnění všech těchto požadavků na vlastní přenos dat byl implementován komunikační protokol HB-12/C. Základní rámeček tohoto protokolu ve své hlavičce

udává celkovou délku rámce včetně hlavičky (4 B), odesílaný příkaz (2 B), kontrolní součet (1 B), unikátní ID každého rámce (2 B) a adresu cílového zařízení (2 B). Po hlavičce následují data, která jsou však pouze nepovinnou součástí celého rámce. Tímto uspořádáním bylo dosaženo možnosti odesílání rámce dlouhého 12 B ÷ 4 GB, jednoznačného určení příjemce a kontroly přijatých dat. Díky specifickému ID každého rámce je také možné zajistit spolehlivou komunikaci odesláním potvrzení o doručení jednotlivých rámců. U tohoto komunikačního protokolu mají rámce jdoucí oběma směry shodný formát, jenž je uveden na obrázku 5.8. Při vlastní komunikaci je využito kódování Little-endian, nejnížší bajt je tedy posílán jako první.

Délka rámce (4 B)				Příkaz (2 B)		Verze	CRC	ID rámce (2 B)		Adresa příjemce (2 B)		Data (0B ÷ 4GB)

Obrázek 5.8: Formát rámce komunikačního protokolu HB-12/C

Oproti předchozímu komunikačnímu protokolu již v rámci není obsažena definice jeho začátku, což umožňuje posílat libovolná binární data. Na druhou stranu je nutné v implementaci protokolu vyřešit jeho detekci. Algoritmus detekce počátku rámce je založen na kontrole správnosti údaje o délce rámce a shodě přijatého kontrolního součtu s vypočteným kontrolním součtem přijatého rámce. Zprvu čeká příjemce, dokud neobdrží alespoň první 4 bajty rámce. Po jejich obdržení je provedena kontrola, zda délka rámce, udaná v prvních čtyřech bajtech, není příliš malá (menší než 12 B, což odpovídá délce hlavičky), nebo příliš velká (v případě omezení maximální velikosti). Pokud je zjištěna chybná hodnota, dojde k zahození prvního přijatého bajtu a příjemce opět čeká, než bude mít k dispozici první čtyři bajty rámce. Pro ty je opět provedena kontrola správnosti obsaženého údaje. Pakliže délka odpovídá určenému rozsahu, čeká příjemce do obdržení všech bajtů rámce. Poté je proveden přepočítání kontrolního součtu. V případě jeho shody je celý rámeček uznán jako platný a je dále zpracován. Pokud se však kontrolní součet neshoduje, je odebrán první bajt celého rámce a celý algoritmus se opakuje. Vývojový diagram tohoto algoritmu je uveden na obrázku 5.9.



Obrázek 5.9: Vývojový diagram algoritmu příjmu rámce

Díky tomuto algoritmu je možné komunikovat mezi zařízeními komunikačním protokolem, který neobsahuje definici začátku rámce a to i při nezabezpečeném přenosu dat, jaký probíhá při komunikaci přes UART. Při samotné implementaci tohoto algoritmu je využíváno kruhového bufferu, jenž je schopen pojmout až dva rámce s maximální určenou délkou. Maximální možnou délku rámce je nutné omezit na vhodnou hodnotu především z důvodu omezené paměti mikroprocesoru. Při komunikaci s robotem byla tato velikost určena jako 64 B. Pro vzájemnou komunikaci osobního počítače a robotu byly definovány zprávy, pomocí nichž si obě zařízení předávají informace. Přehled všech těchto zpráv je uveden v tabulce 5.2, včetně informace, zda je s jejich odesláním předáno i oprávnění promluvit, či nikoliv.

Tabulka 5.2: Zprávy pro vzájemnou komunikaci počítače a robotu

Zpráva	Číslo příkazu	Data	Ztráta oprávnění vysílat	Očekávaná odpověď
Zprávy směřované k robotu				
Připojení k robotu	10	Adresa počítače	ANO (0,2 s)	Potvrzení
Volba módu řízení	20	Zvolený mód	ANO (0,2 s)	Potvrzení
Rychlost robotu	30	Zvolená rychlost	ANO (0,2 s)	Potvrzení
Začátek trasy	40	Sekvence kroků	ANO (0,2 s)	Potvrzení
Data trasy	41	Sekvence kroků	ANO (0,2 s)	Potvrzení
Konec trasy	42	Sekvence kroků	ANO (0,2 s)	Potvrzení
Pohyb robotu	110	Směr pohybu	NE	Žádná
Pohyb kamery	120	Směr pohybu	NE	Žádná
Vých. poloha kamery	121	-	NE	Žádná
Skenování prostředí	130	Měřené směry	NE	Žádná
Senzorická data	210	-	ANO (0,5 s)	Senzorická data
Oprávnění vysílat	500	-	ANO (1 s)	Vysílání robotu
Zprávy směřované k osobnímu počítači				
Senzorická data	310	Naměřená data	NE	Žádná
Stav baterie	311	Stav baterie	NE	Žádná
Poloha robotu	312	Globální souřadnice	NE	Žádná
Zjištěné překážky	320	Naměřená data	NE	Žádná
Přepočet trasy	330	-	NE	Žádná
Oprávnění vysílat	500	-	ANO	Žádná
Potvrzení	999	ID zprávy	NE	Žádná

5.2.2 Komunikace se senzorickými moduly

K přenosu dat od senzorických modulů k mikroprocesoru robotu je možné využít mnoho různých způsobů komunikace. Z důvodu obsazenosti jednotlivých komunikačních linek robotu však byla zvolena možnost jejich vzájemného spojení přes UART,

jelikož byl dosud využit pouze jeden ze dvou UARTů, jimiž mikroprocesor disponuje. Díky tomu je možné druhý vyhradit čistě pro přenos dat od senzorických modulů.

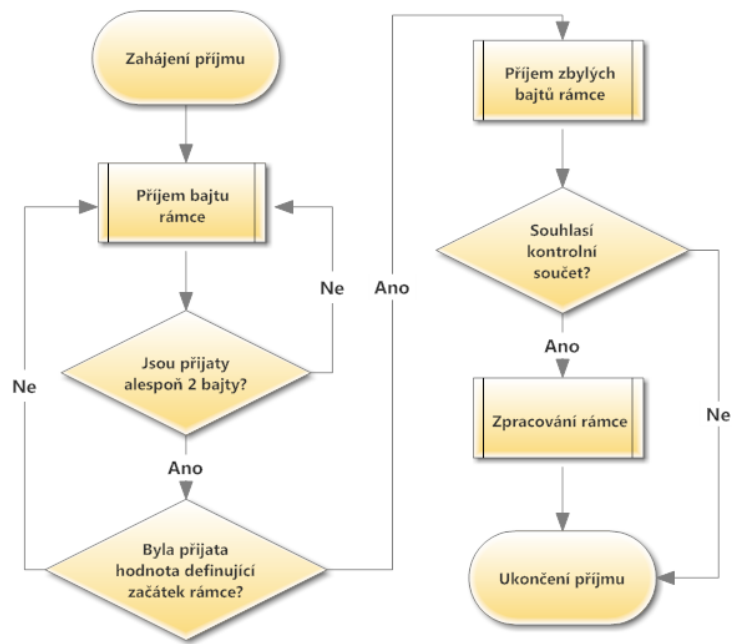
Vzhledem k povaze komunikace, kdy jsou většinu času pouze přeposílána naměřená data, je zbytečné opět nasazovat protokol HB-12/C. Namísto něj byl k tomuto účelu vyvinut proprietární protokol, jenž poskytuje dostatečnou kontrolu přenášených dat, a zároveň jeho implementace není náročná na výpočetní výkon. Hlavička rámce tohoto protokolu (viz obrázek 5.10) se skládá ze 2 bajtů o přesně definované hodnotě, jež označuje jeho začátek, délky rámce včetně hlavičky (1 B), kontrolního součtu (1 B) a identifikátoru příkazu (1 B). Za hlavičkou následují odesílaná data, která mohou mít délku 0 ÷ 250 B. Tento velmi jednoduchý protokol vyniká svou minimální režii, zároveň však poskytuje dostatečnou kontrolu přijatých dat díky kontrolnímu součtu.

Začátek rámce (2 B)		Délka	CRC	Příkaz	Data (0 ÷ 250 B)
255	255				

Obrázek 5.10: Formát rámce proprietárního komunikačního protokolu

Jelikož i zde převyšuje potenciál komunikačního protokolu, co se týče velikosti přenášených dat, jeho reálné využití, byla i v tomto případě omezena maximální velikost celého rámce, tentokrát na 32 B. Při výsledné implementaci protokolu byl opět z důvodu bezpečnosti při přetečení využit kruhový buffer, jenž je schopen pojmout až 64 B dat, tedy až dva rámce o maximální délce. Algoritmus využitý při implementaci tohoto protokolu je uveden na obrázku 5.11. Výsledná komunikace byla poté nastavena na rychlost 115200 Bd.

Ačkoliv v současné době komunikaci využívá pouze jediný modul (modul pro monitorování stavu baterie), je možné v budoucnu robot rozšířit i o další moduly, jež budou komunikovat po téže sběrnici. V té bude mikroprocesor robotu plnit úlohu mastera, kdy bude pouze vyzývat jednotlivé moduly k zaslání naměřených dat. Adresaci zařízení je poté možné realizovat díky 9bitovému režimu, kdy každá odeslaná informace s hodnotou vyšší než 255 obsahuje adresu zařízení, se kterým bude komunikováno.

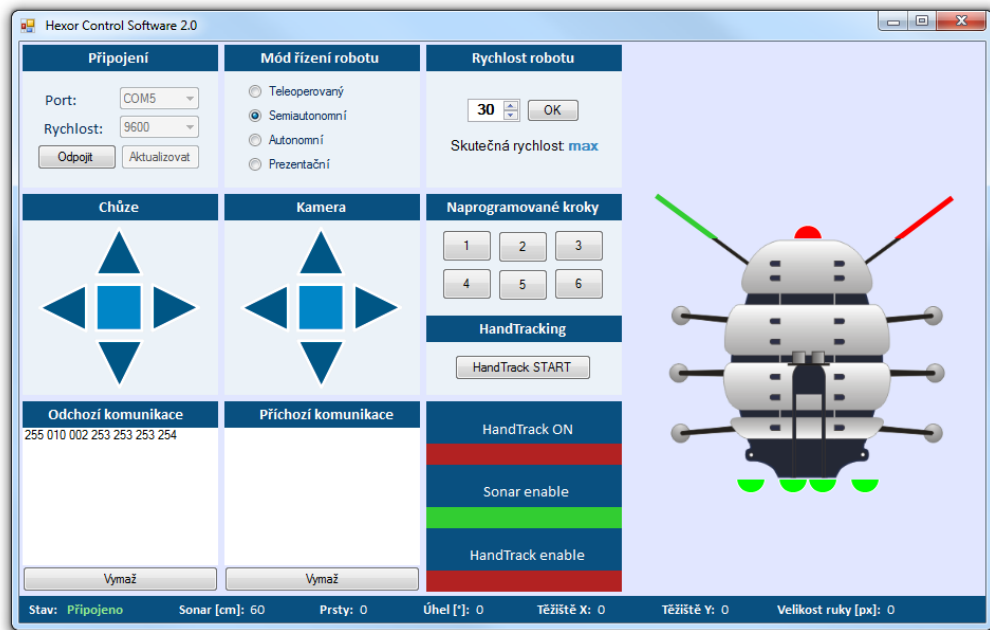


Obrázek 5.11: Vývojový diagram algoritmu příjmu rámce od sensorických modulů

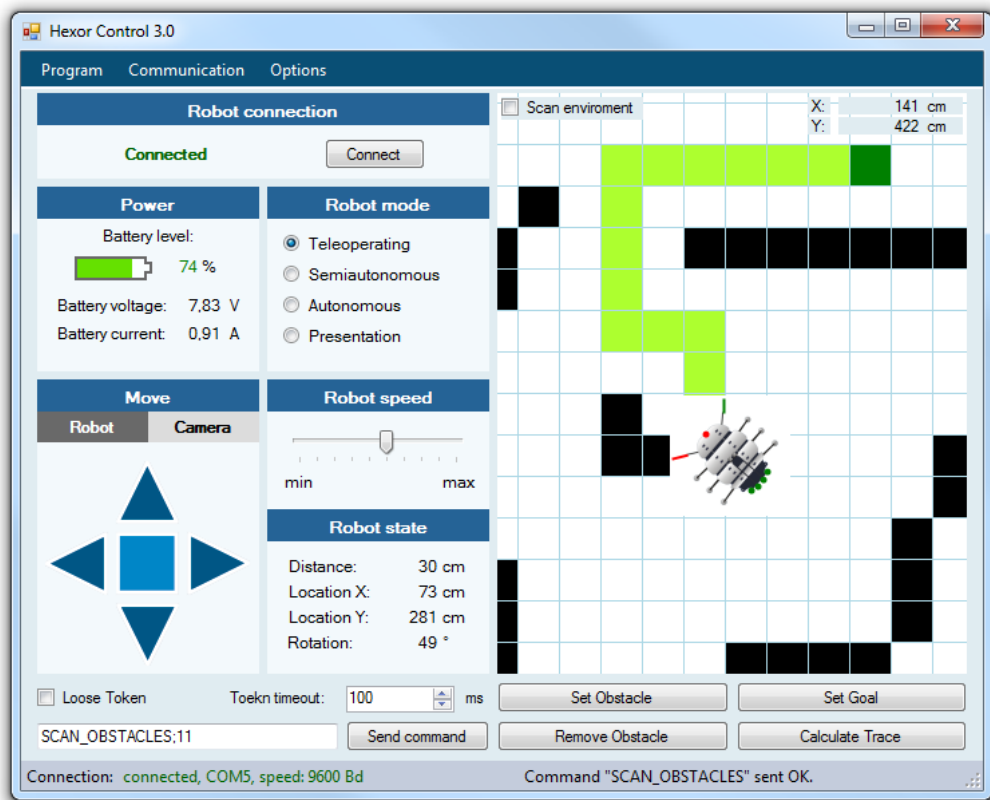
5.3 Řídicí software pro PC

Řídicí software pro osobní počítač hraje velmi důležitou roli v celém řídicím systému, jelikož zprostředkovává uživateli interface pro ovládání robotu a nastavení jeho parametrů. V historii robotu byl tento software již dvakrát přepracován, avšak ani jeho poslední verze (viz obrázek 5.12) není schopna využít celý potenciál robotu. Navíc byl robot rozšířen i o nové funkce, ke kterým není možné prostřednictvím původního programu přistupovat. Z těchto důvodů bylo přistoupeno k vývoji tohoto programu nového.

Nový program je napsán v jazyce C#, přičemž jeho jádro je tvořeno globálním nastavením programu, komunikací s robotem a dvěma singletony, v nichž jsou uloženy veškeré údaje o stavu robotu a aktuálním stavu programu. Nad tímto jádrem je vytvořena grafická nástavba (viz obrázek 5.13), která vytváří grafické uživatelské rozhraní, prostřednictvím kterého může uživatel měnit nastavení samotného programu a přímo robot ovládat. Díky tomuto oddělení jádra programu od jeho grafické části lze velmi snadno přizpůsobit grafickou reprezentaci dat získaných od robotu konkrétnímu účelu, či ji dále rozšířit. Oproti původnímu řešení je navíc celá aplikace koncipována jako vícevláknová, což umožňuje lépe využít softwarové prostředky osobního počítače.



Obrázek 5.12: Původní program pro ovládání robotu skrze PC [10]



Obrázek 5.13: Hlavní okno nového programu pro ovládání robotu skrze PC

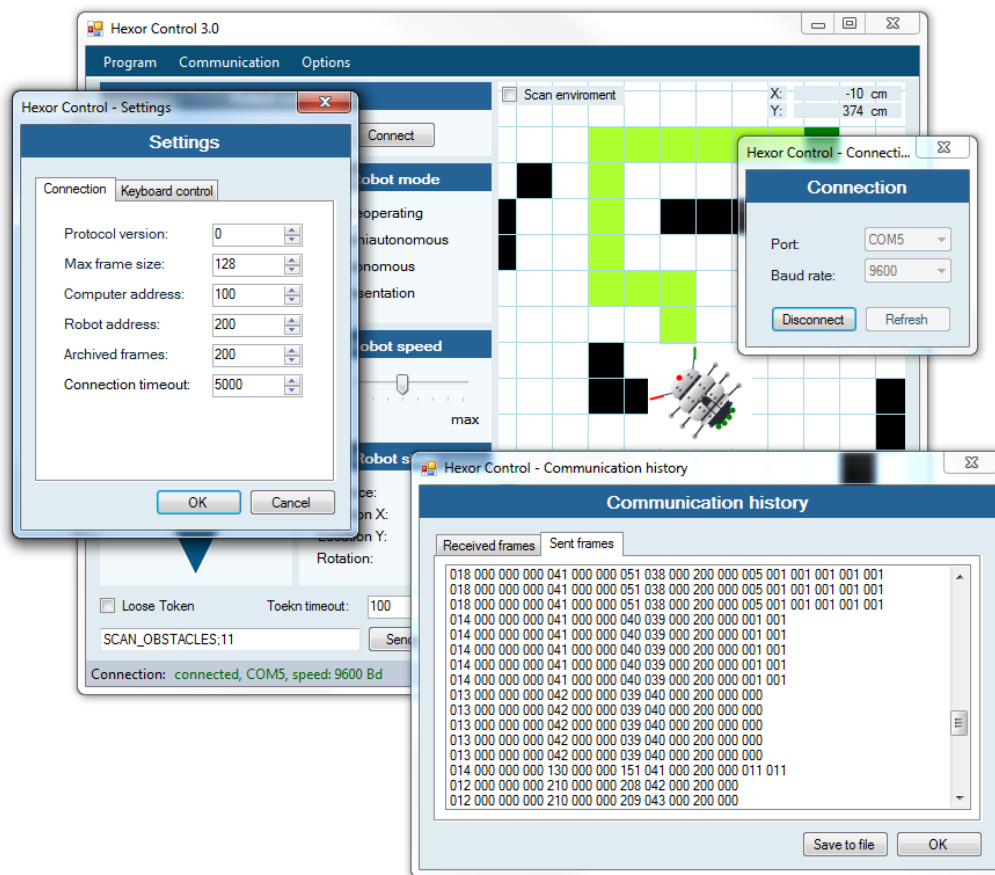
Veškerá nastavení programu jsou uložena v inicializačním souboru *setting.ini*, který je při jeho spuštění načten do paměti. O toto načítání se stará třída *IniFile*, která provádí základní rozdělení jednotlivých proměnných do skupin. Tyto skupiny jsou následně přetypovány na příslušný datový typ, odpovídající charakteru daných proměnných. Takto přetypované skupiny proměnných jsou dále předány třídě *Setting*, ze které je již možné proměnné číst, nebo do nich zapisovat. Všechna jednotlivá nastavení poté zastřešuje třída *ProgramSettings*, která obsahuje kompletní nastavení programu. Po ukončení programu je soubor *setting.ini* vždy přehrán poslední nastavenou konfigurací.

Pro komunikaci s robotem je implementován protokol HB-12/C, přičemž algoritmus příjmu rámce je shodný s algoritmem uvedeným v kapitole 5.2.1. Celá implementace tohoto protokolu byla zapouzdřena do dynamické knihovny *hb12ccs.dll*, aby v případě změny komunikačního protokolu nebylo nutné nijak zasahovat do zbytku programu. Základ knihovny tvoří třídy *ReceiverHB12C* a *FrameBuilderHB12C* sloužící k příjmu a sestavení odchozích rámců. Předání dat ze třídy *ReceiverHB12C* je následně zajištěno skrze interface *IDataReceiver*. Knihovna mimo jiné obsahuje i implementaci této komunikace přímo určenou pro sériovou linku, která je ve třídě *SerialCommunication*.

Samotný program je dále ještě doplněn o vlastní moderování komunikace, jež zajišťuje třída *CommunicationTokenDriver*. Ta si udržuje informaci o tom, zda má oprávnění vysílat, či nikoliv, případně za jak dlouho bude obdrženo zpět. Pokud je třídě zadán požadavek na odeslání rámce, ale nemá oprávnění promluvit, pozdrží odeslání rámce do té doby, než zpět nabude tohoto oprávnění. Program dále disponuje třídami *CommandsRepeater* a *AcknowledgeCommansRepeater* zajišťujícími opakování zadaných příkazů, resp. opakování příkazu do potvrzení o jeho doručení robotem. Zpracování přijatých rámců má poté na starosti třída *DataProcessor*, implementující interface *IDataReceiver*, která na základě rozpoznání příkazu provede uložení přijatých dat do sigletonu vyhrazeného pro stav robotu. Celá příchozí komunikace tak může probíhat v samostatném vlákne a není závislá na zbytku programu.

Jak již bylo řečeno, k ukládání stavu robotu a aktuálního stavu programu slouží celkem dva sigletony. Ty jsou tzv. thread-safe, je k nim tedy možné přistupovat z více vláken najednou, aniž by hrozil pád celého programu. Díky tomu mohou tvořit společné úložiště programu a umožňují přenos dat mezi jednotlivými vlákny. K předávání zpráv v rámci jednoho vlákna je poté využit návrhový vzor *Observer*.

Tato celková koncepce programu umožnila vytvořit uživatelsky přívětivé prostředí, jež je pro maximální přehlednost rozděleno do několika samostatných oken. Při vlastním návrhu bylo využito především standardních ovládacích prvků, aby bylo ovládání celého programu co nejintuitivnější. Kompletní program, včetně všech dalších oken, je vyobrazen na obrázku 5.14.



Obrázek 5.14: Kompletní program pro ovládání robotu skrze PC

5.3.1 Ovládání robotu

Program pro osobní počítač umožňuje uživateli nastavit režim řízení robotu, jeho rychlost, zadat pokyn k jeho pohybu, či určit směr natočení kamery. Kromě těchto základních funkcí však zprostředkovává i zpětnou vazbu směrem od robotu k uživateli. Ten díky němu získává informaci o stavu baterie, aktuálních senzoričkových datech, poloze robotu a dokonce i základní představu o okolním prostředí, v němž se robot pohybuje. Z režimů řízení je možné vybrat mezi teleoperovaným, semiautonomním, autonomním a prezentačním.

Při teleoperovaném režimu robot striktně vykonává příkazy obsluhy a není mu umožněna jakákoliv reakce na jeho okolí. Při semiautonomním je robot opět řízen obsluhou, avšak je již vybaven základní reaktivní architekturou, jež mu umožňuje couvnout před překážkou. Při základním autonomním řízení je prostředí velmi hrubě modelováno samotným robotem, který vždy vybírá směr s největší vzdáleností od překážky. Posledním režimem řízení je prezentační, kdy robot následuje osobu, která stojí před ním. Dále je program doplněn o způsob autonomního řízení, kdy je robotu zadán pouze cíl, kterého má dosáhnout, načež je vyhodnocena cesta k tomuto cíli, kterou robot následně realizuje. Tento režim bude podrobněji popsán v kapitole 5.4. Při všech těchto režimech jsou od robotu neustále získávána data o jeho okolním prostředí, která jsou následně využívána pro jeho modelování.

Při teleoperovaném a semiautonomním režimu je možné primárně ovládat robot pomocí myši, kliknutím na šipku udávající směr jeho pohybu. Pro maximální komfort byl však program dále doplněn o možnost ovládání pohybu pomocí klávesnice. Nastavení kláves, sloužících k tomuto účelu, je možné změnit v grafickém rozhraní programu. Stejným způsobem lze ovládat i pohyb modulu kamery s ultrazvukovým senzorem. Veškeré příkazy, které jsou zasílány robotu, jsou načítány ze zvláštního souboru *commands.ini*. Tím je umožněno alternativní ovládání robotu pomocí příkazového řádku. Každý příkaz lze doplnit o další data, která jsou přidána na konec rámce. Tato data se od příkazu oddělují středníkem a zapisují se po jednotlivých bajtech. U každého příkazu je navíc možné zvolit, zda po jeho odeslání dojde ke ztrátě oprávnění vysílat, či nikoliv.

5.3.2 Mapa prostředí

Okolní prostředí robotu je možné reprezentovat různými způsoby (viz kapitolu 2.2). Jelikož robot HEXOR[®] II disponuje pouze základními exteroceptivními senzory, jež jsou schopné překážky detekovat, byla k vlastnímu modelování využita sensorická mapa. Konkrétně se jedná o binární mapu obsazenosti, kde je celý pracovní prostor rozčleněn do rastru. Každé pole tohoto rastru poté obsahuje údaj o přítomnosti překážky. Vzhledem k nepřilíživé přesnosti senzorů robotu se osvědčila volba rastru s menším rozlišením, konkrétně se jednalo o velikost jednoho pole 20×20 cm. Jedno pole rastru poté přibližně odpovídá $1/6$ plochy obsazené samotným robotem. Grafická reprezentace této mapy je patrná z obrázku 5.13.

K určení překážek v pracovním prostoru je primárně využito ultrazvukového senzoru, jelikož je jako jediný z dostupných senzorů schopen určit vzdálenost od překážky. Ostatní senzory (infračervené a taktilní) jsou schopny určit pouze jejich přítomnost v těsné blízkosti robotu. Díky možnosti nastavení směru natočení ultrazvukového senzoru, je možné jím skenovat okolní prostředí v úrovni výšky konce „ocasů“ robotu v horizontálním rozsahu $-90 \div 90^\circ$. K vlastnímu skenování prostředí je robot vždy vyzván osobním počítačem, který mu udá počet měřených bodů připadající na plný úhlový rozsah 180° . Výsledek skenování je poté v robotu uložen do jednodimenzionálního pole, které je počítači při jeho následném dotazu na sensorická data odeslán.

Získaná sensorická data udávají vždy pouze polohu překážek vzhledem k lokálnímu souřadnému systému robotu. Ze známé polohy a úhlu jeho rotace je však možné je přepočítat na polohu v globálním souřadném systému. Takto získané informace o poloze překážek je možné přímo zanést do výsledné mapy. Všechny překážky, které byly v mapě evidovány mezi robotem a novou polohou překážky, jsou zároveň vymazány, jelikož senzor vždy detekuje nejbližší překážku v daném směru. Zbylé senzory jsou následně využity pouze pro doplnění překážek, které nebyly ultrazvukovým senzorem detekovány (např. byly moc nízké nebo tenké).

5.4 Autonomní řízení

Autonomních způsobů řízení existuje nepřehledné množství. Některé roboty jsou navigovány pomocí optických či magnetických čar, jiné zase hledají cestu ve složitém prostředí na základě jeho modelu [14]. Robot HEXOR[®] II byl v původním řídicím systému vybaven velmi jednoduchým autonomním řízením, kdy na základě elementární mapy prostředí určoval směr svého pohybu. Mapa byla tvořena pouze jednorozměrným polem, ve kterém byly uloženy vzdálenosti robotu od překážek v jednotlivých směrech. Z těchto údajů robot vybral směr s největší vzdáleností od překážky. Tento způsob autonomního řízení byl natolik jednoduchý, že byl implementován v samotném mikroprocesoru robotu. Z hlediska využití je však vhodný pouze pro předváděcí účely, kdy je schopen např. najít cestu ve velmi jednoduchém bludišti.

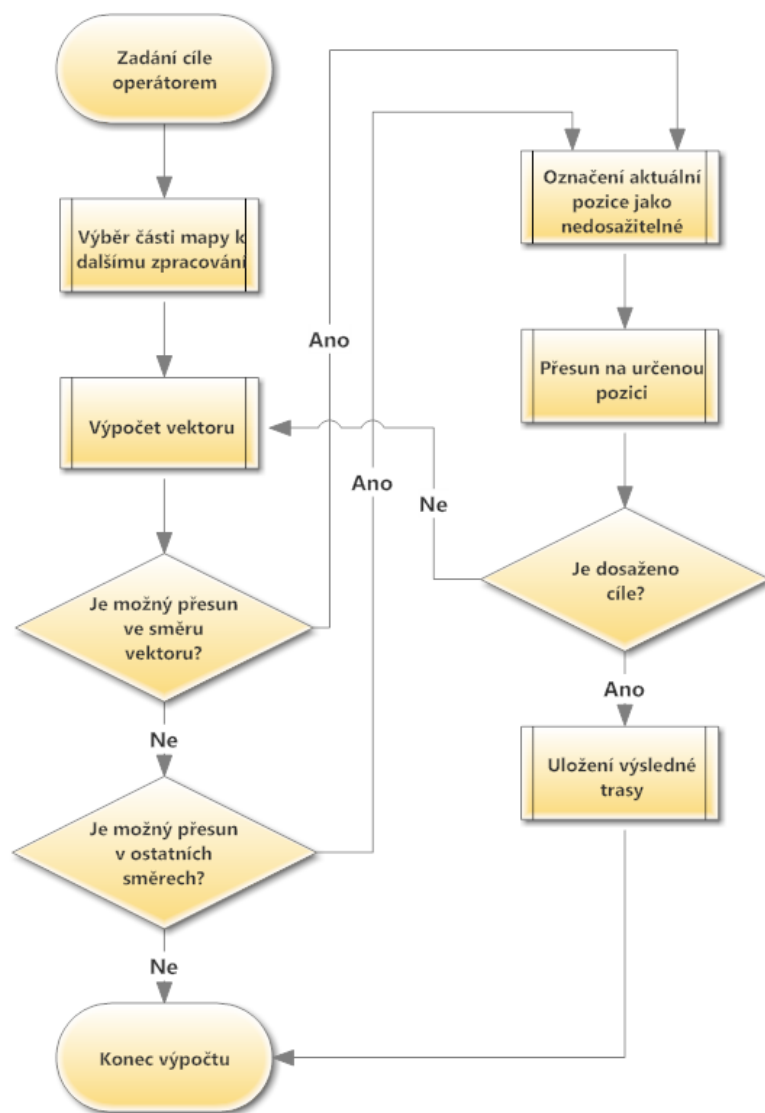
Nový řídicí systém však poskytuje mnohem dokonalejší model okolního prostředí, vyjádřený mapou obsazenosti. Díky tomu lze implementovat v osobním počítači sofistikovanější způsob autonomního řízení, než jakým robot doposud disponoval.

V tomto novém autonomním režimu by měl být robot schopen samostatně dosáhnout cílové polohy zadané operátorem.

K zadání cílové polohy slouží grafické uživatelské rozhraní programu pro osobní počítač. V něm je možné pouhým kliknutím do rastru mapy určit cíl, kterého má robot dosáhnout. Po jeho určení je od jeho současné polohy proveden výpočet trasy, k němuž je využito vektoru určujícího přímý směr od robotu ke stanovenému cíli. Samotný algoritmus je rekurzivní a pracuje s jednotlivými posuny v rastru.

Primárně je testována možnost pohybu robotu ve směru určeného vektorem, tedy ve směru kterým se přiblíží k požadovanému cíli. Tento vektor je vždy před každým krokem aktualizován. Pokud není možné pokračovat v cestě ve směru vektoru, jsou alternativně zkušeny všechny ostatní směry. Jakmile je zjištěn možný přesun na další pozici, je předchozí pozice označena jako nedosažitelná a výpočet probíhá dále z nově nalezené pozice. Opět je poté přepočítán vektor a testována možnost pohybu robotu v jeho směru. Takto pokračuje celý výpočet až do chvíle, kdy robot dosáhne cíle, či uvázne ve slepé uličce. Jakmile je dosaženo cíle, je celá trasa uložena do seznamu realizovatelných cest. V opačném případě je celá trasa zahozena. Ze všech takto vypočtených cest je následně vybrána vždy ta nejkratší. Celý algoritmus je uveden v diagramu na obrázku 5.15.

Takto určená trasa je následně převedena na sekvenci kroků, jež má robot vykonat. Ta je robotu odeslána a následně jím samotným zcela autonomně realizována. Tímto způsobem bylo využito distribuovaného výpočetního výkonu osobního počítače k řešení poměrně složitého problému výpočtu cesty ve zmapovaném prostředí. Během vykonávání předem určené sekvence kroků je robotem prováděna neustálá kontrola přítomnosti překážek v jeho bezprostředním okolí. V případě její detekce je vlastní vykonávání sekvence kroků pozastaveno a robotem je zvolena adekvátní reakce k jeho uvolnění. Poté je počítači odeslán požadavek na přepočet trasy k cíli z jeho aktuální polohy. Tímto způsobem bylo dosaženo spojení funkční dekompozice a reaktivní architektury pro zajištění maximální robustnosti celého systému autonomního řízení.



Obrázek 5.15: Vývojový diagram algoritmu výpočtu trasy robotu

6 Závěr

Nový řídicí systém robotu HEXOR[®] II zachovává jeho původní architekturu, tedy jeho základní rozdělení na nízkoúrovňové a vysokoúrovňové řízení. Nízkoúrovňové řízení je zajištěno samotným mikroprocesorem robotu, který obstarává obsluhu akčního a senzorického subsystému. Dále je na něm implementována základní reaktivní architektura umožňující robotu reagovat na jeho okolní prostředí. Tato část řídicího systému byla do značné míry ponechána původní, jelikož její modulární struktura vyhovovala požadavkům nové koncepce řídicího systému.

I zde však byly učiněny změny, především v oblasti senzorického subsystému. Ten byl rozšířen o nový senzorický modul, jež má za úkol monitorovat stav baterie, konkrétně její úroveň nabití, napětí a odebíraný proud. S tím však bylo nutné zajistit i výměnu energetické jednotky robotu, jelikož původní řešení vykazovalo poměrně závažné nedostatky. Těmi byly především příliš velká hmotnost a rozměry akumulátoru, nízká výdrž na jedno nabití a jeho nevhodné připojení ke kostře robotu. Z těchto důvodů byl původní olověný akumulátor nahrazen Li-ion akumulátorem, s výrazně lepšími parametry pro napájení mobilního bateriového zařízení. Mimo samotnou výměnu akumulátoru byla tato jednotka doplněna o integrovaný nabíjecí modul, jež umožňuje jeho nabíjení z libovolného zdroje elektrického napětí v rozmezí 12 ÷ 20 V. Kromě této úpravy bylo do této části řídicího systému přidáno i určení aktuální polohy robotu, vzhledem ke globálnímu souřadnému systému, pomocí odometrie.

V části vysokoúrovňového řízení byl vyvinut zcela nový řídicí program pro osobní počítač, který mnohem lépe využívá dostupné možnosti mobilního robotu. Oproti jeho předchozí verzi je zde zajištěno modelování okolního prostředí senzorickou mapou obsazenosti a jednotlivá senzorická data jsou zde mnohem přehledněji graficky reprezentována. V programu je dále implementováno autonomní řízení, které umožňuje uživateli zadat pouze cíl, kterého má robot dosáhnout. Způsob dosažení tohoto cíle je poté vyhodnocen samotným programem a celé řízení probíhá

již zcela autonomně. Celý program tak tvoří kompaktní funkční celek vhodný nejen pro ovládání robotu, ale i pro další vývoj a ladění nových algoritmů jeho řízení.

Komunikace mezi těmito částmi řídicího systému zajišťují bezdrátové komunikační moduly, s nimiž je komunikováno po standardní sériové lince. Oproti jeho původnímu řešení však došlo ke změně komunikačního protokolu, jelikož původní řešení neumožňovalo zajistit přenos dostatečného množství dat s potřebnou spolehlivostí. Nově navržený systém tak zajišťuje moderování komunikace, kontrolu přenášených dat a zpětné informování o doručení odeslaných dat.

Oproti původnímu řešení přináší nová koncepce řídicího systému mnohem vyšší přehled obsluhy o stavu robotu a zprostředkovává informaci okolním prostředí jeho modelováním. Díky tomu je ovládání robotu mnohem snazší a je možné lépe využít distribuovaného výpočetního výkonu osobního počítače k náročnějším a pokročilejším způsobům jeho řízení. Tento systém tak tvoří robustní základ pro různé aplikace tohoto mobilního robotu.

Literatura

- [1] ALLEGRO MICROSYSTEMS, Inc. *ACS712: Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor* [on-line]. USA: Massachusetts, 2006 [cit. 2015-05-04]. Dostupné z: <http://www.tme.eu/cz/Document/75af175f80c090e9b8f9078a0b0b2409/ACS712.PDF>
- [2] BELZA, Jaroslav. Akumulátory Li-ion a jejich nabíjení. *Časopis Praktická elektronika*, č. 4, 2001.
- [3] CENEK, Miroslav. *Akumulátory od principu k praxi*. Praha: FCC Public, 2003, 248 s. ISBN 80-86534-03-0.
- [4] ATMEL CORPORATION. *8-bit Atmel with 8KBytes In-System Programmable Flash: ATmega8, ATmega8L* [on-line]. USA: San Jose, 2013 [cit. 2015-05-04]. Dostupné z: http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_datasheet.pdf
- [5] NATIONAL SEMICONDUCTOR CORPORATION. *LM2576/LM2576HV Series SIMPLE SWITCHER® 3A Step-Down Voltage Regulator* [on-line]. 1999 [cit. 2015-05-04]. Dostupné z: <http://www.gme.cz/img/cache/doc/332/049/lm2576t-adj-datasheet-1.pdf>
- [6] DURCHÁNEK, Martin. *Interaktivní metody ovládání robota Hexor II*. Liberec, 2013. Diplomová práce. Technická univerzita v Liberci. Vedoucí práce Miroslav Holada.
- [7] JANOVEC, Aleš. *Navigace mobilního robota pomocí fuzzy logiky*. Brno, 2010. Dostupné z: http://autnt.fme.vutbr.cz/szz/2010/DP_Janovec.pdf Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Jiří Dvořák.
- [8] KAMEŠ, Josef. *Alternativní pohon automobilů*, 1. vyd. Praha: BEN - technická literatura, 2004, 231 s. ISBN 80-730-0127-6.
- [9] KOŠNAR, Karel. Mobilní robotika, *Pandatron.cz Elektrotechnický magazín* [on-line]. Praha: České Vysoké Učení Technické v Praze, 2010 [cit. 2015-04-18]. Dostupné z: http://pandatron.cz/?1745&mobilni_robotika

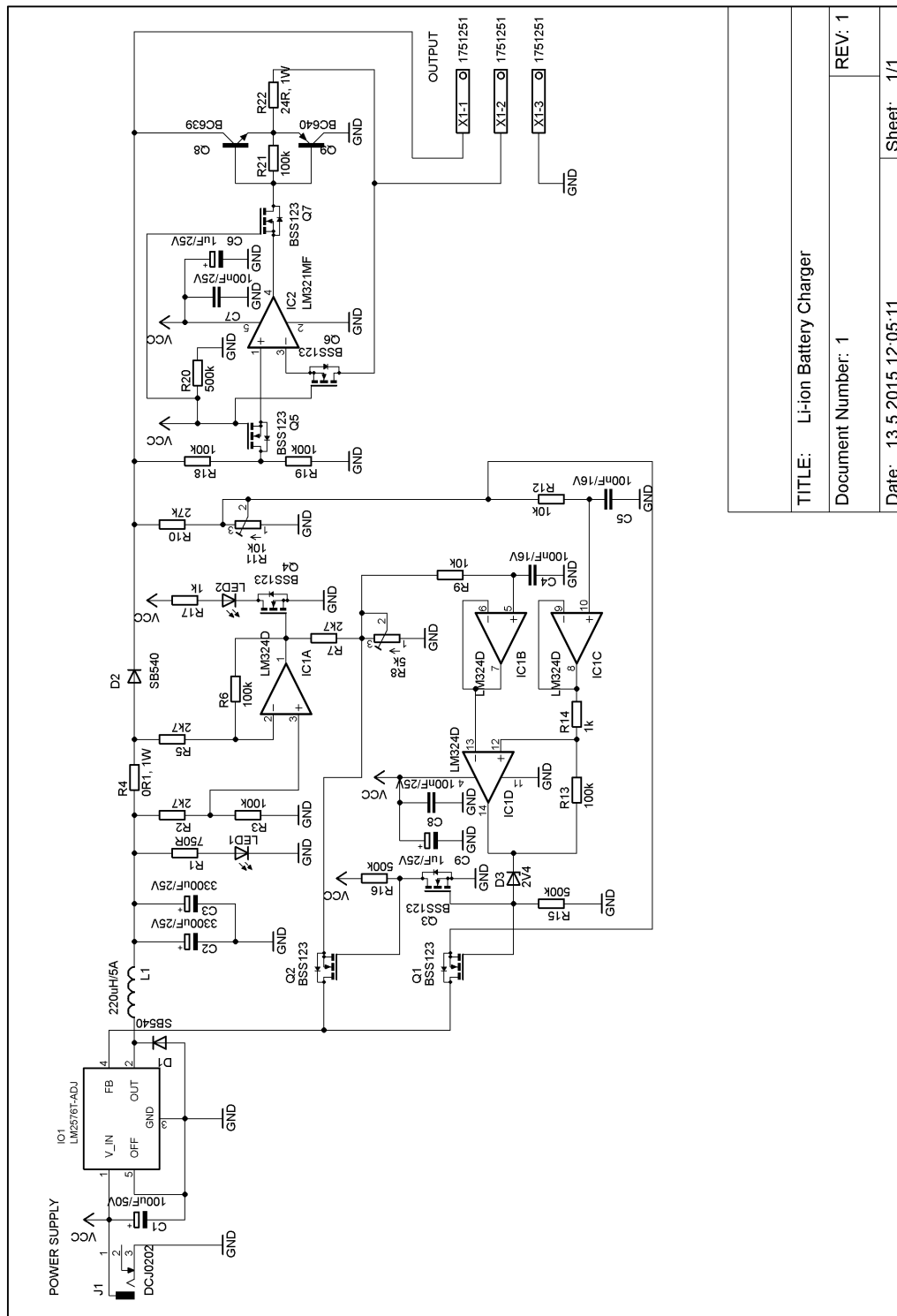
- [10] KREDBA, Jan. *Návrh vylepšení řídicího systému robota Hexor II*. Liberec, 2014. Ročníkový projekt. Technická univerzita v Liberci. Vedoucí práce Miroslav Holada.
- [11] LINDEN, David a Thomas BREDDY. *Handbook of batteries*, 3. vyd. New York: McGraw-Hill, 2002. ISBN 00-713-5978-8.
- [12] STENZEL LTD. *HeXor® II The mobile robot for research and education: User's Guide - Quick start v 2.0a*. Poland: Gliwice, 2007, dostupné na CD dodávaném s robotem Hexor® II.
- [13] MICROCHIP TECHNOLOGY, Inc. *MCP1525/41: 2.5V and 4.096V Voltage References* [on-line]. 2012 [cit. 2015-05-04]. Dostupné z: http://www.tme.eu/cz/Document/099a0ddd51530999c29263c97c74c06b/MCP1525-I_T0.pdf
- [14] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*, 1. vyd. Praha: BEN - technická literatura, 2004, 248 s. ISBN 80-730-0141-1.
- [15] SIEGWART, Roland. *Introduction to autonomous mobile robots*, 1. vyd. Massachusetts: MIT Press, 2004, 321 s. ISBN 02-621-9502-X. Dostupné z: http://www.robotee.com/EBooks/Introduction_to_Autonomous_Mobile_Robots.pdf
- [16] TOPIČ, Pavel. *Provedení ověřovacích experimentů lokalizace mobilního robota*. Brno, 2007. Dostupné z: http://autnt.fme.vutbr.cz/szz/2007/BP_Topic.pdf Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Stanislav Věchet.

A Obsah přiloženého CD

Přiložené CD obsahuje:

- text bakalářské práce
 - bakalarska_prace_2015_Jan_Kredba.pdf
- program pro ovládání robotu skrze PC
 - spustitelná verze programu
 - zdrojový kód programu (v programovacím jazyce C#, IDE MS Visual Studio 2013 Ultimate)
- zdrojový kód programu mikrokontroléru robotu (v programovacím jazyce C, IDE Atmel Studio 6.2)
- zdrojový kód programu mikrokontroléru měřicího modulu (v programovacím jazyce C, IDE Atmel Studio 6.2)
- výkresovou dokumentaci navržených modulů
 - schéma elektrického zapojení (ve formátu .sch, EAGLE 7.2.0)
 - návrh desek plošných spojů (ve formátu .brd, EAGLE 7.2.0)
- katalogové listy použitých součástek

B Schéma zapojení nabíjecího modulu



TITLE: Li-ion Battery Charger

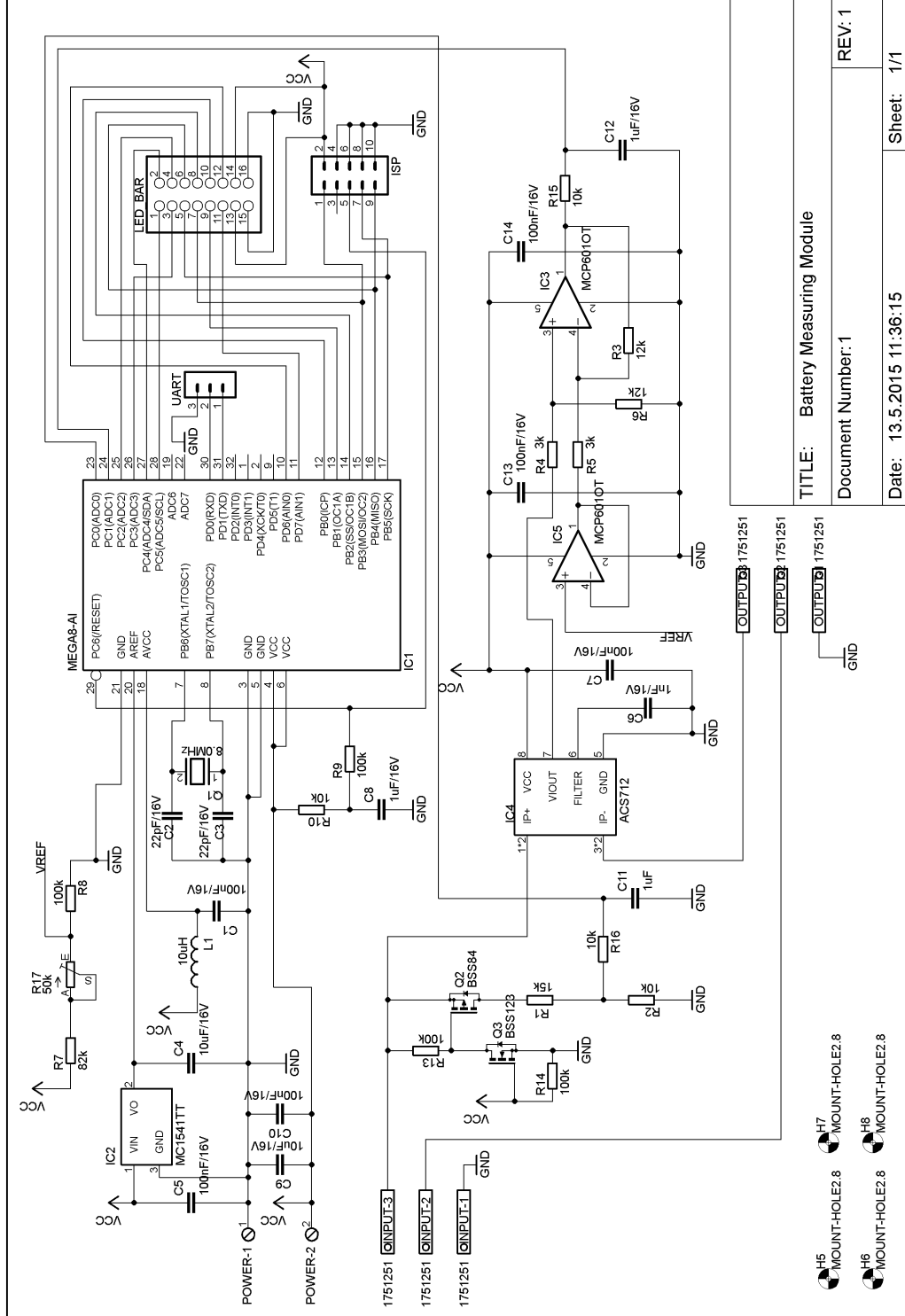
Document Number: 1

Date: 13.5.2015 12:05:11

REV: 1

Sheet: 1/1

C Schéma zapojení měřicího modulu



TITLE: Battery Measuring Module	
Document Number:1	
Date: 13.5.2015 11:36:15	
H5 MOUNT-HOLE2.8	REV: 1
H6 MOUNT-HOLE2.8	Document Number:1
H7 MOUNT-HOLE2.8	Date: 13.5.2015 11:36:15
H8 MOUNT-HOLE2.8	Sheet: 1/1