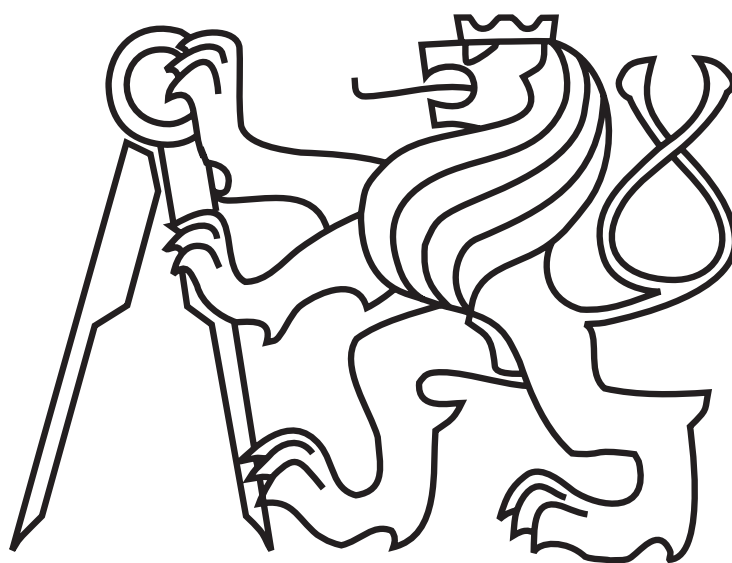


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Bakalářská práce



Monika Svědřihová

Řídicí systém mobilního robotu

Katedra kybernetiky

Vedoucí bakalářské práce: **Ing. Tomáš Krajník**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Monika Svědřihová
Studijní program: Kybernetika a robotika (bakalářský)
Obor: Robotika
Název tématu: Řídicí systém mobilního robotu

Pokyny pro vypracování:

Navrhňte a realizujte řídicí systém mobilního robotu schopného překonávat obtížný terén. Systém by měl odhadovat pozici robotu na základě odometrických dat, implementovat základní algoritmy vyhýbání se překážkám a zprostředkovávat komunikaci s nadřazeným PC. Práce zahrnuje výběr vhodných senzorů a metod zpracování senzorických dat, návrh a realizace desek elektroniky, vývoj řídicího a navigačního software, integrace systému do frameworku ROS a otestování výsledného robotu v reálných podmínkách.

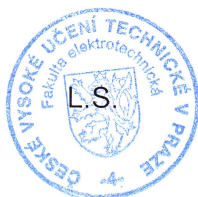
1. Prostudujte stav současné problematiky v řízení mobilních robotů, zpracování senzorických dat a odhadování pozice z proprioceptivních čidel.
2. Vyberte vhodné senzory a algoritmy pro mobilní robot schopný autonomní navigace v obtížném terénu.
3. Navrhňte strukturu řídicího systému pro výše zmíněný robot a systém realizujte.
4. Systém začleňte do frameworku ROS.

Seznam odborné literatury: Dodá vedoucí práce.

Vedoucí bakalářské práce: Ing. Tomáš Krajník

Platnost zadání: do konce zimního semestru 2012/2013


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 9. 1. 2012

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne.....25.05.2012.....

.....Svědirová.....

Poděkování

Přednostně bych chtěla poděkovat vedoucímu bakalářské práce Ing. Tomášovi Krajníkovi a kolegům z práce za cenné rady a připomínky při psaní této bakalářské práce. Dále rodině za podporu a pevné nervy.

Abstrakt

Cílem této bakalářské práce bylo navrhnout a realizovat řídicí systém mobilního robotu schopného překonávat obtížný terén. Práce popisuje výběr vhodných senzorů, metody zpracování dat ze senzorů, návrh a realizaci elektroniky robotu, vývoj řídicího a implementaci navigačního software. Výsledkem práce je řídicí systém schopný ovládat mobilního robot ve vnitřních i venkovních podmínkách, otestovaný v reálném prostředí. Řídicí systém je univerzální, může být součástí výukové či testovací platformy pro robotické aplikace.

Abstract

The aim of the thesis is to design and realize a control system for a mobile robot, which is able to go over difficult terrain. The thesis describes selection of suitable sensors, methods of processing data from sensors, design and realization of robot's electronics, development of control algorithm and implementation of the navigation software. The result of the work is a control system, which is able to control the mobile robot in both indoor and outdoor conditions and is tested in the real environment. The control system is universal - it can be incorporated into educational or testing platforms for robotic applications.

Obsah

1	Úvod	1
2	Senzory pro mobilní robotiku	2
2.1	Interní senzory	2
2.1.1	Inkrementální senzory pro měření relativní polohy	2
2.1.2	Senzor pro měření absolutní polohy	3
2.1.3	Akcelerometr	3
2.1.4	Gyroskop	5
2.1.5	Rozdíl mezi MEMS gyroskopem a akcelerometrem	6
2.2	Externí senzory	7
2.2.1	Kompas	7
2.2.2	Taktilní senzory	8
2.2.3	Infračervený senzor	8
2.2.4	Ultrazvukový dálkoměr	8
2.2.5	Kamera	8
3	Metody zpracování dat ze senzorů	9
3.1	Odhad vzdálenosti překážky	9
3.2	Odometrie	9
4	Hardware	11
4.1	Senzorický systém robotu	11
4.1.1	Vyhýbání se překážkám	11
4.1.2	Výpočet odometrie	12
4.1.3	Interní senzory navigace	12
4.2	Robotická platforma MMP5	14
4.3	Řídicí deska	15
4.3.1	Schéma	15
4.3.2	Návrh plošného spoje	18
5	Software	20
5.1	Ultrazvukový dálkoměr	20
5.2	Senzor CHR-6d	21
5.3	Sledování barevných objektů	22
5.4	Sledování cesty	22
6	ROS	24
6.1	Základní stavební prvky	24
6.1.1	Vysvětlení základních pojmů	24
6.2	Dálkové ovládání	24
6.3	Ukázka závislosti balíčků	25

7	Experimenty	26
7.1	Odometrie	26
7.2	Sledování cesty	26
7.3	Sledování barevných objektů	27
7.4	Test subsumpční architektury	27
8	Závěr	29
	Příloha A	32
	Příloha B	33
	Příloha C	34
	Příloha D	36
	Příloha E	38

Seznam obrázků

1	Princip detekce směru otáčení kotouče enkodérů, převzato [5]	3
2	Senzor pro měření absolutní polohy, převzato [3]	4
3	Princip kapacitního akcelerometru, převzato [3]	5
4	Zjednodušená struktura MEMS Gyroskopu, převzato [8]	6
5	Příklad funkce struktury snímače gyroskopu při rotaci, převzato [8]	6
6	Model ve vědeckém muzeu v Londýně, převzato [9]	7
7	Model diferenciálního robota, převzato [10]	10
8	Ultrazvukový dálkoměr SRF08	11
9	Vnitřní zapojení Hallovy sondy	12
10	Kolo s magnety	13
11	Modul CHR-6D	13
12	Platforma MMP5	14
13	Zapojení mikroprocesoru ATmega128	16
14	Zapojení FTDI	17
15	Vrchní vrstva řídicí desky	18
16	Spodní vrstva	19
17	Kompletně osazená řídicí deska	19
18	Princip interakcí v robotu	21
19	Detekce cesty	23
20	Ukázka závislosti balíčků	25
21	Fotografie z testu na sledování cesty	27
22	Fotografie z testu na sledování barevných objektů	28
23	Kalibrace odometrie v terénu	36
24	Fotografie z testu na sledování objektů	36
25	Fotografie z testu na sledování cesty	37
26	Fotografie z testu na sledování cesty za pouličního osvětlení	37

Seznam tabulek

1	Struktura paketu	13
2	Popis paketu	14
3	Nastavení motorů	15
4	Přesnost odometrie s kalibrací na 10 m	26
5	Přesnost odometrie po kalibraci na 100 m	26
6	Obsah příloženého CD	38

Seznam příloh

Příloha A	32
Příloha B	33
Příloha C	34
Příloha D	36
Příloha E	38

1 Úvod

Již v roce 1920 se v divadelní hře R.U.R. (Rossumovi univerzální roboti) od Karla Čapka zrodila myšlenka stroje, který by pomáhal či postupem času úplně nahradil lidskou práci. Tento stroj byl Josefem Čapkem pojmenován jako robot, jméno bylo nejspíše odvozeno od slova robota (otrocká práce). Dnes je slovo robot již běžně používané a charakterizuje stroj usnadňující práci člověka. Může se jednat o průmyslové manipulátory v továrnách, kuchyňské roboty v domácnostech nebo třeba o inteligentní vozidla na Marsu. Jedno z možných dělení robotů je dle jejich schopnosti se přemisťovat. Stacionární roboty se nemohou pohybovat z místa na místo, plní úlohu průmyslového manipulátoru. Mobilní roboty se mohou přemisťovat, například inteligentní vozidla na Marsu.

Pro zajištění robustnosti, rozšiřitelnosti a možnosti řešení více cílů, využíváme při návrhu robotů metodiku založenou na různých architekturách. U mobilních robotů používáme zejména dvě architektury: Sence-plan-act a Subsumpční. Sence-plan-act architektura, jak již název napovídá, je založena na třech krocích, vnímat (sense) okolí, naplánovat akce (plan) a nakonec provést naplánovanou činnost. Na této architektuře byl postaven první univerzální robot Shakey [1], vyvinutý SRI's Artificial Intelligence Center (dříve Stanford Research Institute), který měl podstatný vliv na vývoj umělé inteligence a robotiky.

Subsumpční architektura byla poprvé představena Rodneyem Brooksem v roce 1986 [2]. Jedná se o vrstvenou architekturu, kde jednotlivé vrstvy reprezentují určité chování. Vrstvy mají vzestupnou prioritu. Například robot může mít ve spodní vrstvě zabudované nouzové zastavení před překážkou a vyšší vrstva řeší, jak danou překážku objede. Jednotlivé vrstvy mohou pracovat na daných cílech souběžně. Obsluha senzorů nemusí být součástí všech vrstev, vyšší vrstvy zpracovávají výsledky ze senzorů, zatímco nižší vrstvy dál senzory používají. Tím je zajištěn neustálý běh procesů a není nutné čekat na zpracování měřených dat. Mobilní robot s touto architekturou je kdykoliv rozšiřitelný přidáním dalších vrstev nebo senzorů.

Tato práce popisuje vytvoření univerzálního řídicího systému založeného na zmiňované subsumpční architektuře. Popisuje návrh jednotlivých vrstev od nejnižší vrstvy zajišťující bezpečnost robota v podobě bezdotykových nárazníků až po vrstvy obsluhující algoritmy na rozpoznávání obrazu. Dále je zde popsán vývoj a realizace elektroniky robota, která bude obsluhovat základní typy senzorů sloužících pro orientaci robota v prostoru.

2 Senzory pro mobilní robotiku

Senzory jsou v současnosti již nezbytnou součástí našeho života, v různých podobách se s nimi denně setkáváme, ať již v samoobsluze v podobě optické závory, nebo třeba při záchranné akci k prozkoumání terénu. Prakticky je nemožné se bez nich obejít. „Senzor je funkční prvek tvořící vstupní blok měřicího řetězce, který je v přímém styku s měřeným prostředím. Senzor snímá sledovanou fyzikální, chemickou nebo biologickou veličinu a transformuje ji na výstupní veličinu – nejčastěji analogový nebo číslicový elektrický signál“ [3]. V robotice senzory nahrazují naše smyslové orgány, jsou to oči a uši robotu. Sensorů existuje nepřeberné množství od obyčejného spínače až po velmi složité systémy integrovaných akcelerometrů a gyroskopů. Každý je založen na jiném principu a způsobu komunikace.

Senzory se dají rozdělit do dvou základních skupin dle vztahu k robotu. První skupinu tvoří interní senzory, které měří stavy robotu, a druhou skupinou jsou senzory externí, měřící parametry okolí robotu.

2.1 Interní senzory

Interní senzory umožňují robotu zjistit jeho vnitřní stav, například polohu, rychlost, stav nabití baterie apod. Podle měřené veličiny je dále dělíme na diagnostické (informace o stavu baterie či teploty robotu) a navigační (informace o orientaci v prostoru a rychlosti).

2.1.1 Inkrementální senzory pro měření relativní polohy

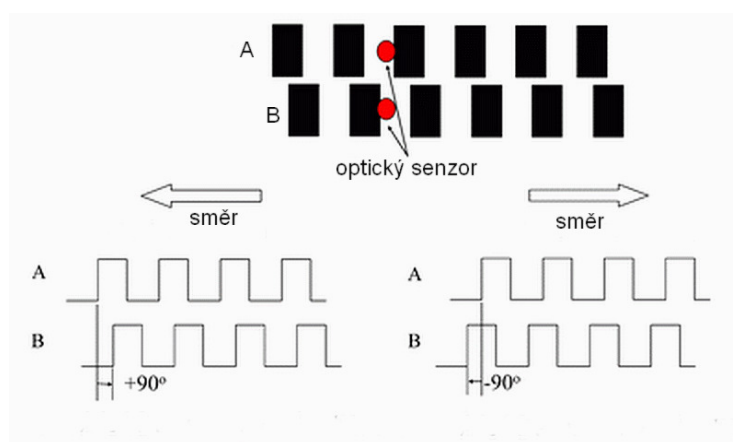
Inkrementální senzory se používají zejména ve zpětnovazebním řízení na určení otáček kola, výpočtu rychlosti a ujeté vzdálenosti. Senzory pracují na různých principech. Zde si představíme základní dva typy sensorů, magnetické, pracující na principu Hallova jevu, a senzory optické.

- Magnetické snímače polohy využívající Hallův jev byly vyvinuty pro zajištění spolehlivé a přitom levné zpětné vazby při regulaci polohy a pohybu. Mohou pracovat i za nepříznivých podmínek a vysokých rychlostí. Hallův jev je fyzikální proces, kdy se za současného působení elektrického a magnetického pole hromadí na jedné straně polovodiče kladný náboj a na druhé straně záporný náboj. Vzhledem k tomu, že póly mají různý potenciál, vzniká tzv. Hallovo napětí [4]. Hallův snímač je nejrozšířenějším a nejjednodušším typem sensorů polohy. „Jedná se o elektronický spínač ovládaný Hallovým napětím, viz rovnice (1),

$$U_H = K_H \cdot B \cdot I, \quad (1)$$

generovaným na elektrodách při působení magnetického pole B na polovodičovou destičku napájenou kolmo na elektrody proudem I “ [3].

- Princip činnosti optických senzorů je založen na otáčivém mezikruží s pravidelně se střídajícími průhlednými a neprůhlednými ploškami, které přerušují světlo od svítivé diody (LED). Průchod světla je detekován senzorem světla, například fototranzistorem, který je umístěn naproti LED. Do optické cesty se většinou umísťuje také maskovací kotouč se stejnou roztečí štěrbin, zaručující maximální tok světla na optický senzor. Získáme obdélníkový signál, kde každý obdélník odpovídá jedné plošce. Při jejich rovnoměrném rozmístění po obvodu kotouče odpovídá jeden obdélník otočení vždy určitému počtu stupňů. Pro zjištění směru otáčení potřebujeme ještě jeden senzor, který je v ideálním případě namontován tak, že jeho signál je s původním signálem fázově posunut o 90° , viz obrázek 1.



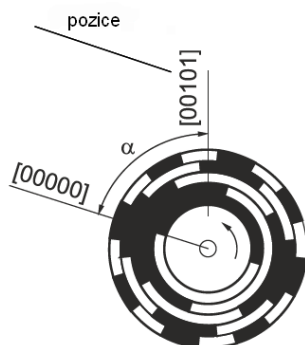
Obrázek 1: Princip detekce směru otáčení kotouče enkodérů, převzato [5]

2.1.2 Senzor pro měření absolutní polohy

Tyto snímače jsou konstruovány na stejném principu jako čidla inkrementální. Tento typ snímače poskytuje informaci o úhlu natočení hřídele motoru bezprostředně po připojení napájení, bez nutnosti předchozího pootočení. Jejich rotující kotouč má stopu s otvory, tvořícími určitý kód, jak je vidět na obrázku 2. S výhodou se používá Grayův kód. Tento kód se při přechodu do sousední polohy mění pouze v jednom bitu, a proto je detekce směru a korekce chyb snadná.

2.1.3 Akcelerometr

Akcelerometr je senzor umožňující měření zrychlení, náklonu a vibrací daného předmětu. V dnešní době je k dispozici několik druhů senzorů fungujících na různých principech kapacitní, piezodporové, piezoelektrické atd. Posledním trendem v oblasti vývoje jsou akcelerometry založené na MEMS (mikro-elektromechanické) technologii. Mají mnohem menší rozměry, nižší energetickou spotřebu a nižší cenu.



Obrázek 2: Senzor pro měření absolutní polohy, převzato [3]

Struktura a funkce MEMS akcelerometru je založena na proměnné kapacitě tříelektrodového vzduchového kondenzátoru. Vychází se ze základního vztahu pro působení síly při zrychlení, viz rovnice (2):

$$F = m \cdot a, \quad (2)$$

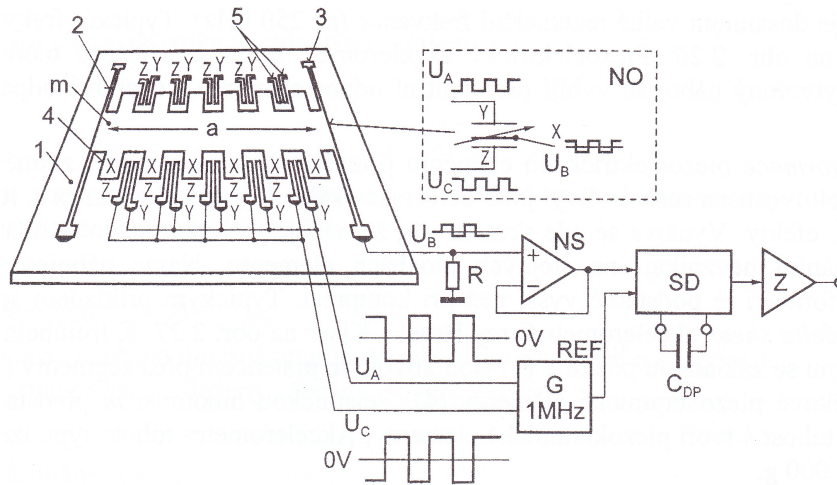
kde F [N] je síla vzniklá působením zrychlení a [m/s^2] na hmotu m [kg]. Síla se pak přes pružiny převádí na posuv tětiny, jejíž některé části tvoří pohyblivé elektrody vzduchového kondenzátoru. Jejich pozice vůči levým pevným elektrodám a pravým pevným elektrodám určuje elektricky měřenou hodnotu kapacity takto vzniklého kondenzátoru [6], dle vzorce 3:

$$C = \varepsilon \cdot \frac{S}{d}, \quad (3)$$

(C - kapacita kondenzátoru [F], ε - permitivita [$\frac{F}{m}$], S - plocha elektrod [m^2], d - vzdálenost elektrod [m]).

Konstrukce, která je zobrazena na obrázku 3, je založená na destičce z polykrystalického křemíku, tvarovaná do dvou pružných tětin (2), zakotvených na monokrystalickém křemíkovém substrátu (1). Obě tětiny, představující tuhost k mechanického oscilátoru, jsou spojeny hřebínkem. Každý z padesáti zubů hřebínku představuje střední pohyblivou elektrodu X (4) soustavy stejného počtu diferenciálních kapacitních senzorů s proměnnou vzduchovou mezerou. Jako pevné elektrody slouží systém nosníků Y, Z (5). Při působení horizontálního zrychlení, například ve směru doprava, se zvýší kapacita mezi elektrodami X, Z a poklesne mezi elektrodami X, Y. Měřící obvod je můstkového typu, elektrody Y a Z jsou napájeny pravoúhlým napětím navzájem invertovaným. Napětí nerovnováhy na odporu R je demodulováno synchronním detektorem. Výstupní napětí demodulátoru je přivedeno na elektrodu X a vyvolá elektrostatický zpětnovazební silový účinek, vyvažující působení měřeného zrychlení [3].

MEMS akcelerometry mají široké využití, nachází se v řadě mobilních telefonů, v automobilovém průmyslu pro zajištění bezpečnosti a například v robotice pro měření náklonu a zrychlení.



Obrázek 3: Princip kapacitního akcelerometru, převzato [3]

2.1.4 Gyroskop

Použití gyroskopů je vedle akcelerometrů další možností, jak měřit pohyb, přesněji řečeno rychlost otáčení. Dříve se používalo zejména mechanické provedení, případně optické s využitím světla nebo optických vláken. Dnes je již lze také najít v MEMS architektuře.

Gyroskopy jsou obecně určené pro měření úhlové rychlosti. Rotaci je možné typicky měřit vzhledem k jedné ze tří os z , y , x . Integrované gyroskopy pracují na principu Coriolisovy síly. Coriolisova síla je setrvačná síla působící na tělesa, která se pohybují v rotující neinerciální vztažné soustavě tak, že se mění jejich vzdálenost od osy otáčení [7]. Vypočteme ji dle vzorce 4:

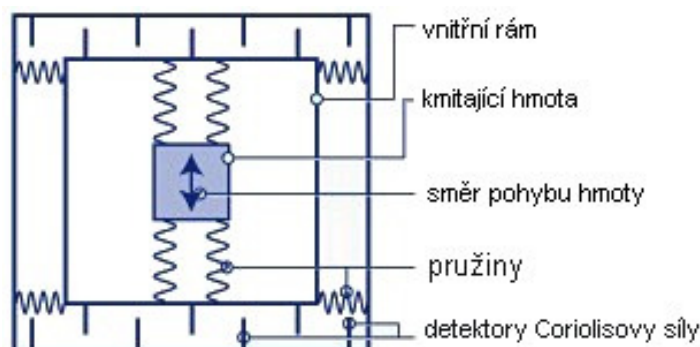
$$F_C = -2m\vec{\omega} \times \vec{v}, \quad (4)$$

kde F_C je Coriolisova síla, m je hmotnost tělesa, \vec{v} je vektor rychlosti tělesa v neinerciální vztažné soustavě, $\vec{\omega}$ je vektor úhlové rychlosti otáčení soustavy a \times označuje vektorový součin.

U mechanických gyroskopů je princip jednoduchý, při pohybu objektu, upevněného na pružinách uvnitř rámu, směrem ven (k okraji rotujícího kotouče) na něj působí Coriolisova síla směrem doleva, při opačném směru pohybu objektu pak doprava. Protože velikost a směr této síly je úměrný i rychlosti a směru otáčení, lze tento systém snadno využít pro měření velikosti úhlové rychlosti.

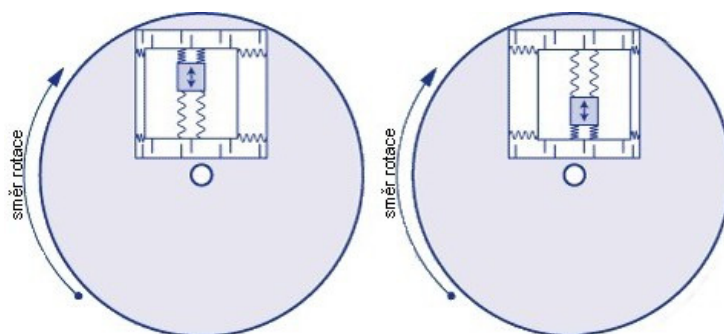
Při praktickém použití Coriolisovy síly v integrovaných gyroskopech se využívá technologie MEMS, kde se vytváří na čipu spolu s elektrickými obvody i mechanické mikrosoučásti, které tvoří samotný snímač. Různí výrobci sice používají trochu odlišné struktury, ale základní princip je vždy podobný. Základem je periodicky se pohybující (mechanicky rezonující) struktura přesně dané hmotnosti, upevněná pomocí pružin v rámu.

Směr pohyb musí být vždy kolmý ke směru otáčení. Za těchto podmínek vzniká Coriolisova síla, jejíž velikost je úměrná úhlové rychlosti otáčení, působící na hmotnou pohy-



Obrázek 4: Zjednodušená struktura MEMS Gyroskopu, převzato [8]

bující část snímače. Ta způsobuje stlačení vnějších pružin rámu a způsobí vzájemný posuv měřících plošek, fungujících jako elektrody vzduchových kondenzátorů. Výstupem je změna kapacity úměrná úhlové rychlosti otáčení [8].



Obrázek 5: Příklad funkce struktury snímače gyroskopu při rotaci, převzato [8]

2.1.5 Rozdíl mezi MEMS gyroskopem a akcelerometrem

Z pohledu základního principu je jen jeden rozdíl, zato však významný. Oba využívají podobný pohyblivý rám se snímacími ploškami pracujícími jako elektrody kondenzátorů, u kterých se při pohybu rámu mění kapacita, ale zatímco pohyblivá struktura je u akcelerometru volná a hýbe se jen při působení síly (zrychlení), u gyroskopu je s touto strukturou záměrně periodicky pohybováno (mechanická rezonance), aby se projevil efekt Coriolisovy síly. Zatímco akcelerometr měří hodnotu vychýlení vnitřní MEMS struktury jako důsledek působícího zrychlení, gyroskop zjišťuje rychlost ustáleného rotačního pohybu [8].

2.2 Externí senzory

Externí senzory umožňují zjistit informace z okolního prostředí, například teplotu, vlhkost, elektromagnetické působení, vzdálenost překážky atd. Robotu usnadňují orientaci, pomáhají v navigaci a zabezpečují bezpečnost robota vůči okolí. Není vždy úplně jasné, zda se jedná o externí či interní senzor, některé senzory mohou spadat do obou kategorií.

2.2.1 Kompas

Historicky prvním externím senzorem byl kompas, založený na mechanickém principu. Jeho mechanismus neměl magnety a neuměl automaticky rozpoznat jih. Byl to starověký čínský dvoukolový vůz, na kterém byl umístěn pohyblivý ukazatel v podobě postavičky ukazující směr. Ukazatel byl na začátku cesty nastaven ručně na jih, poté mechanismus za pomoci různých převodů udržel postavičku směrem na jih, i když se vůz obrátil. Ukázkou provedení můžeme vidět na obrázku 6. Typický dnešní kompas obsahuje volně pohyblivou



Obrázek 6: Model ve vědeckém muzeu v Londýně, převzato [9]

magnetickou střílku, která si zachovává směr sever-jih podle zemského magnetického pole. Dnes existují také kompasové senzory založené na mnoha dalších fyzikálních principech, například elektronický a indukční. Máme kompasové senzory speciálně navržené pro oblast robotiky, jako modul CMPS01. Modul používá dvojici vzájemně kolmých senzorů magnetického pole založených na magnetorezistivním jevu, který říká, že elektrický odpor magnetického materiálu je závislý na velikosti a směru externího magnetického pole, v němž je materiál umístěn. Poskytuje informaci o azimutu dvěma způsoby – formou PWM (impulsně šířkové modulace) nebo pomocí I^2C sběrnice. Přesnost měření je uváděna 3° až 4° s rozlišitelností 0,1°.

2.2.2 Taktilní senzory

Nejjednodušším provedením taktilního senzoru je kontaktní spínač. Při kontaktu dojde k sepnutí či rozepnutí spínače a uzavření či otevření elektronického obvodu. Vznikne změna logické hodnoty, která je dále vyhodnocována. V robotu je můžeme vidět v podobě taktilní antény (různě dlouhý drát s malou ohebností, který při vychýlení či stlačení sepne spínač) nebo taktilního nárazníku (při nárazu sepne spínač a detekuje překážku, což bývá poslední ochrana robotu). Implementaci můžeme vidět i u robotických manipulátorů k detekci uchopení předmětu.

2.2.3 Infračervený senzor

Zařízení funguje na principu sledování odraženého infračerveného světelného paprsku od případné překážky. Jako zdroj světla (vysílač) je použita infračervená dioda a jako detektor (přijímač) fototranzistor citlivý v infračervené oblasti, což má rozhodující vliv na vlastnosti a schopnosti senzoru. Při běžném denním osvětlení je spolehlivý dosah senzoru okolo 20 cm. Nevýhodou IR senzoru je, že množství odraženého světla je závislé na barvě překážky a druhu povrchu.

2.2.4 Ultrazvukový dálkoměr

Měření probíhá pomocí ultrazvukového signálu, který vyšle vysílač, signál prochází prostředím, jakmile narazí na překážku, část tohoto signálu se odrazí a putuje zpět. Podle doby prodlevy mezi vysláním a přijmutím signálu se určí, jak daleko je překážka. Velkou výhodou tohoto senzoru je, že ho lze uplatnit i v podmínkách, kde optická detekce selhává, například při detekci průhledných či lesklých předmětů. Nevýhodou může být širší prostorový úhel paprsku.

2.2.5 Kamera

Historie zpracování obrazu v počítači se začíná psát v letech, kdy existující výpočetní technika již umožnila zpracování objemu dat, který je spojen s obrazovou informací. Kamera zachytí obraz zkoumaného předmětu, systém jej vyhodnotí podle předepsaného algoritmu a provede akci na základě výsledku vyhodnocení. Kamer máme nepřeberné množství od speciálně zaměřených průmyslových kamer až po obyčejné domácí kamery. V robotice se velmi často využívají web kamery pro jejich cenovou dostupnost. Bohužel, vzhledem k nízké ceně mají také nevýhody, například danou dobu závěrky apod.

3 Metody zpracování dat ze senzorů

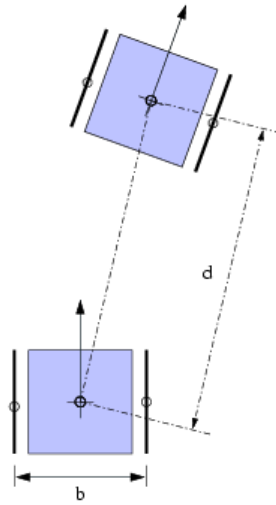
3.1 Odhad vzdálenosti překážky

Pro odhad vzdálenosti překážek nejčastěji využíváme ultrazvukové a infračervené senzory. Oba senzory nám udávají vzdálenost překážek na základě odrazu (princip popsán 2.2.4), ale předání a zpracování informace je odlišné. Infračervené senzory podávají informaci v podobě analogového signálu (napětí), který převedeme A/D převodníkem na digitální signál. Digitální hodnotu dále zpracujeme mikroprocesorem. Ultrazvukové senzory, konkrétně modul SRF08, který je v této platformě použit, komunikuje po sběrnici I^2C a posílá nám digitální zpracovanou informaci (hodnotu). Tu pouze převedeme z hexadecimální soustavy do desítkové a tím získáme vzdálenost překážky v cm. Oba typy senzorů se nejčastěji využívají v nejnižší vrstvě subsumpční architektury. Napomáhají k orientaci v podobě bezdotykových nárazníků (zastav, jakmile je překážka blíže než daný práh) či k mapování překážek a tvoření základních map daného prostoru.

3.2 Odometrie

Odometrie je jedna z nejčastěji používaných metod pro určování polohy, popisuje transformaci dat poskytnutých enkodéry na změnu pozice a orientace robota. Vlastní slovo odometrie je složeno ze dvou řeckých slov, hodos (cestovat, cesta) a metron (měřit). Základem odometrie je znalost geometrického modelu podvozku robota. Tyto modely se liší zejména tím, jakých druhů pohybu jsou roboty schopny. Většinu mobilních podvozků můžeme rozdělit do těchto tří kategorií:

- Ackermanův podvozek, kategorie zahrnuje podvozky, které se nedokáží otáčet na místě, protože změny orientace mohou dosáhnout pouze příslušným natočením kola či nápravy následovaným pohybem vpřed či vzad. Tomuto způsobu pohybu se často říká Ackermanovo řízení a zahrnuje většinu dnešních automobilů.
- Diferenciální „tank“, do této kategorie řadíme všechny roboty, schopné otáčet se na místě kolem své osy. Pohybují se pouze vpřed či vzad. Často je nazýváme diferenciálně řízenými, protože změna orientace závisí na rozdílu rychlosti levého a pravého kola, jak je znázorněno na obrázku 7.
- Všesměrové roboty, se dokáží pohybovat všemi směry bez ohledu na aktuální orientaci, včetně otáčení na místě. S touto kategorií podvozků se nesetkáváme příliš často, i když poslední dobou začíná být populární na rovném povrchu, například v robotickém fotbalu.



Obrázek 7: Model diferenciálního robota, převzato [10]

Dále se budu již zabývat pouze podvozkem typu tank, protože na něm je postaven robot Blackshot.

Základním rysem tohoto modelu jsou dvě nezávisle poháněná kola/nápravy. Pokud se obě kola točí stejně rychle stejným směrem, robot jede rovně. Pokud se naopak točí opačným směrem, otáčí se celý robot na místě kolem své osy. Když je rozdíl rychlostí mezi nápravami nenulový a obě nápravy se točí na stejnou stranu, pohybuje se tento robot po kružnici se středem vně robota. Pokud je rozdíl rychlostí náprav nenulový a obě nápravy se točí různým směrem, leží střed kružnice uvnitř robota. Vzdálenost středu této kružnice je dána poměrem rychlostí pravého a levého kola 5:

$$R = \frac{1}{2} \cdot b \cdot \frac{(v_L + v_R)}{(v_L - v_R)}, \quad (5)$$

kde b je vzájemná vzdálenost kol (rozchod) a v_L a v_R jsou rychlosti. Pokud levé kolo ujede vzdálenost d_L a pravé d_R , změní se orientace o úhel θ (v radiánech)

$$\theta = \frac{d_L - d_R}{b}, \quad (6)$$

kde b je již zmiňovaný rozchod. Celková ujetá vzdálenost d (počítaná pro střed poháněné osy) je pak

$$d = \frac{(d_L + d_R)}{2}. \quad (7)$$

Natočení robota závisí pouze na rozdílu celkově ujeté vzdálenosti pravého a levého kola a nikoliv na průběhu jednotlivých změn. K výpočtu směru tedy stačí i jednoduchý čítač, který lze snadno realizovat v jednočipu (nejsou třeba goniometrické funkce). Při popisu odometrie bylo čerpáno ze stránek [10].

4 Hardware

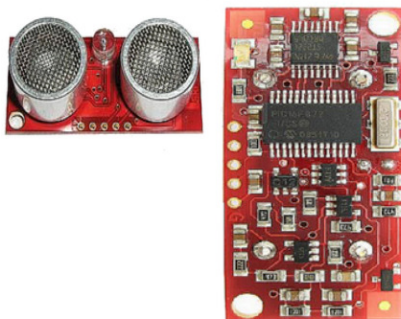
4.1 Senzorický systém robota

Návrh senzorického systému robota je vždy komplexní proces, při kterém musíme brát ohledy na způsob komunikace jednotlivých senzorů, způsob měření dané veličiny, nároky na výpočet a zpracování měřené veličiny. Nejdůležitějším faktorem při výběru senzorů je očekávaná funkce robota a jeho využití. Dále nesmíme zapomenout zohlednit konstrukční provedení robotické platformy, její nosnost a rozměry. Robot Blackshot je postaven na robotické platformě MMP5, která má rozměry 29x28x11 cm a nosnost 1,8 kg. Robot se má pohybovat převážně ve venkovním prostředí a být schopen základní navigace v daném prostředí.

4.1.1 Vyhýbání se překážkám

Pro vyhýbání se překážkám byl vybrán ultrazvukový dálkoměr SRF08, zobrazen na obrázku 8. Princip detekce překážek je popsán v kapitole 3.1. Velkou výhodou tohoto senzoru je zpracování informace přímo v modulu SRF08, který posílá vypočítanou vzdálenost překážky převedenou na zvolené jednotky (v palcích, v centimetrech, v mikrosekundách).

Komunikace probíhá po sběrnici I^2C , výchozí nastavení adresy je na 0xE0. Nastavení



Obrázek 8: Ultrazvukový dálkoměr SRF08

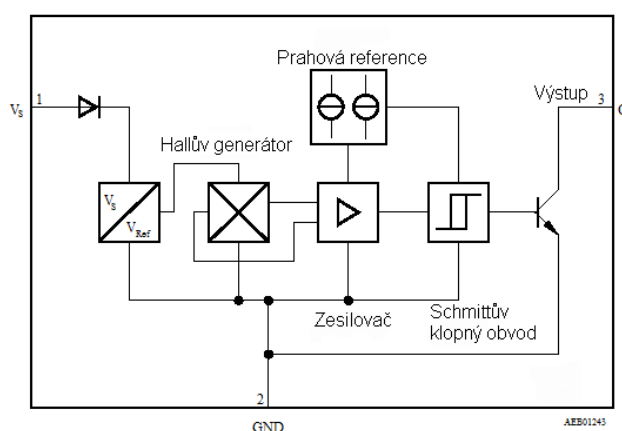
adresy se dá uživatelem změnit na některou z 16 adres. Možnost volby 16 adres nám umožňuje použít až 16 senzorů. Kromě výše uvedených adres senzor reaguje na adresu 0. Zapisovat lze pouze do registrů 0, 1 a 2. Registr 0 je příkazový registr, slouží k zahájení měření. Čtení registru 0 slouží k získání verze firmware. Doba měření je nastavena na 65 ms, ale může být změněna zápisem do registru 2. Registr 1 poskytuje při čtení údaje ze senzoru světla. Tato hodnota je aktualizována při každém měření vzdálenosti. Následující dva registry udávají poslední změřenou vzdálenost. Měření se zahájí zápisem jednoho z příkazů

do příkazového registru, poté je třeba čekat na dokončení měření a přočíst potřebný počet výsledků. Hodnoty všech výsledků jsou nulovány na začátku každého měření. Podrobnější informace a přehled příkazů najdeme v katalogovém listu [11].

4.1.2 Výpočet odometrie

Měření otáček motoru k výpočtu odometrie je zajištěno pomocí Hallovy sondy TLE4905, která využívá Hallova jevu popsáno v kapitole 2.1.1.

Vnitřní zapojení sondy můžeme vidět na obrázku 9. Výstup sondy je připojen na otevřený



Obrázek 9: Vnitřní zapojení Hallovy sondy

kolektor tranzistoru, při detekci magnetického pole dojde k otevření tranzistoru a uzemnění výstupu. Změní se logická hodnota výstupu z logické „1“ na logickou „0“. Počet otáček motoru zjišťujeme sadou magnetů rozmístěných ve stejných vzdálenostech přímo na kole, viz obrázek 10. V blízkosti kola je umístěna Hallova sonda, u které se změnila logická hodnota výstupu při každém přiblížení magnetu a tím zajistí generování obdélníkového signálu. Obdélníkový signál je přiveden na vstup mikroprocesoru, který při každé detekci sestupné hrany inkrementuje vnitřní proměnnou. Hodnota proměnné odpovídá počtu prošlých magnetů. Díky známé vzdálenosti mezi magnety a velikosti kola můžeme určit počet otáček a ujetou vzdálenost. Bohužel tento způsob odečtu otáček kola neurčuje směr otáčení, ten je zjišťován podle stavu motoru. Robot si pamatuje nastavení motorů, a když se mu změnil směr otáčení motoru, aktualizuje směr odometrie.

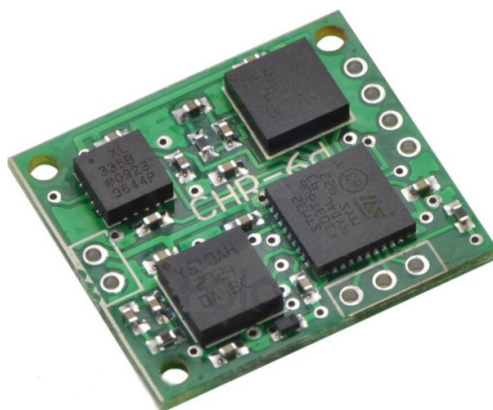
4.1.3 Interní senzory navigace

Zrychlení a úhlová rychlost robotu je měřena senzorem CHR-6d. Jedná se o digitální inerciální měřící jednotku, obsahující 3-osý akcelerometr a gyroskop na založenou MEMS technologie (popsán v kapitole 2.1.3 a 2.1.4) v jednom modulu. Tento modul má rozměry



Obrázek 10: Kolo s magnety

$2 \times 1,8 \times 0,2$ cm a hmotnost pouze 1,5 gramů. Modul je zobrazen na 11, obsahuje 32b procesor s jádrem ARM Cortex, který zajišťuje zpracování a filtrování měřených hodnot. Komunikace s modulem probíhá po sériové lince rychlostí 115200 bd ve formě paketů.



Obrázek 11: Modul CHR-6D

Struktura paketů je zobrazena v tabulce 1 s doprovodným popisem v tabulce 2.

Funkce	's'	'n'	'p'	PT	N	d_1	...	d_N	CHK
Bajt	1	2	3	4	5	6	...	N+5	N+6 N+7

Tabulka 1: Struktura paketu

Struktura vysílacích i přijímacích paketů je stejná. Ve výchozím nastavení modul zasílá data ze šesti kanálů, tří os gyroskopu a akcelerometru. Data jsou ve formě 16b dvojkového doplnku celých čísel. Pokud chceme aktuální data v příslušném měřítku, je nutné je vynásobit danou konstantou k . Data z gyroskopu, po vynásobení konstantou $k = 0,02014$, udávají úhlovou rychlost ve stupních za sekundu a akcelerometr, po vynásobení konstantou $k = 0,0001678$, udává zrychlení v m/s^2 . V roce 2010 byl do jednotky dodán vestavěný

1-3	Posloupnost tří znaků definující začátek paketu
4 PT	Určuje typ paketu
5 N	Počet očekávaných bajtů
6 - (N+5)	Obsahuje N datových bajtů
(N+6) - (N+7)	Dva bajty zajišťující kontrolní součet

Tabulka 2: Popis paketu

Kalmanův filtr pro odhad náklonu a otočení, využívající kombinaci akcelerometru a gyroskopu. Tento filtr odstraňuje negativní vliv přechodných vibrací. Podrobnější informace včetně kalibrace najdeme v katalogovém listu [12]. Tento modul má rozsáhlé využití v robotice, ale také v dalších aplikacích, slouží například ke stabilizaci obrazu, sledování pohybu, GPS navigaci atd.

4.2 Robotická platforma MMP5

Robotická platforma MMP5 zobrazena na obrázku 12 je představitelem robotické platformy pro venkovní prostředí. Její rozměry a terénní kola umožňují rychlou a bezproblémovou jízdu i po nerovném terénu. Další výhodou je její cenová dostupnost oproti konkurenčním výrobkům. Veškerá elektronika k obsluze motorů a základního řízení je součástí platformy, tím dává možnost jízdy i za nepříznivých podmínek, například v prašném prostředí. Součástí platformy jsou 4 stejnosměrné motory se zabudovanou převodovkou



Obrázek 12: Platforma MMP5

s převodovým poměrem 30:1. K napájení slouží sada článků NiCd, skládající se z 10 článků o nominálním napětí 1,2 V a kapacitě 1400 mAh, které zajišťují jmenovité napětí 12 V pro motory a jejich řízení. Řízení motorů zprostředkovává řídicí regulátor, který umožňuje dva způsoby komunikace, dálkové ovládání, pomocí kterého si vygenerujeme

vlastní PWM modulaci, a sériovou komunikaci. Podrobnější informace k platformě MMP5 najdeme v uživatelském manuálu [13], včetně různých režimů nastavení pomocí přepínače na regulátoru. K řízení motorů je použita sériová komunikace. Rychlost sériové komunikace byla zvolena na 19200 bd. Řízení probíhá diferenciallyně, celý podvozek je řízen jako tank, jak je popsáno v kapitole 3.2. Zjednodušený sériový režim umožňuje nastavení motorů pomocí dvou bajtů. Nejvyšší bit udává řízenou stranu, zbývajících 7b nastavuje směr a rychlost otáčení viz tabulka 3.

1 - 64	pravá strana, směr vpřed
65 - 127	pravá strana, směr vzad
128 - 192	levá strana, směr vpřed
193 - 255	levá strana, směr vzad
0	nulová rychlost pro obě strany
1, 127, 128, 255	maximální rychlost
64, 192	nulová rychlost

Tabulka 3: Nastavení motorů

4.3 Řídicí deska

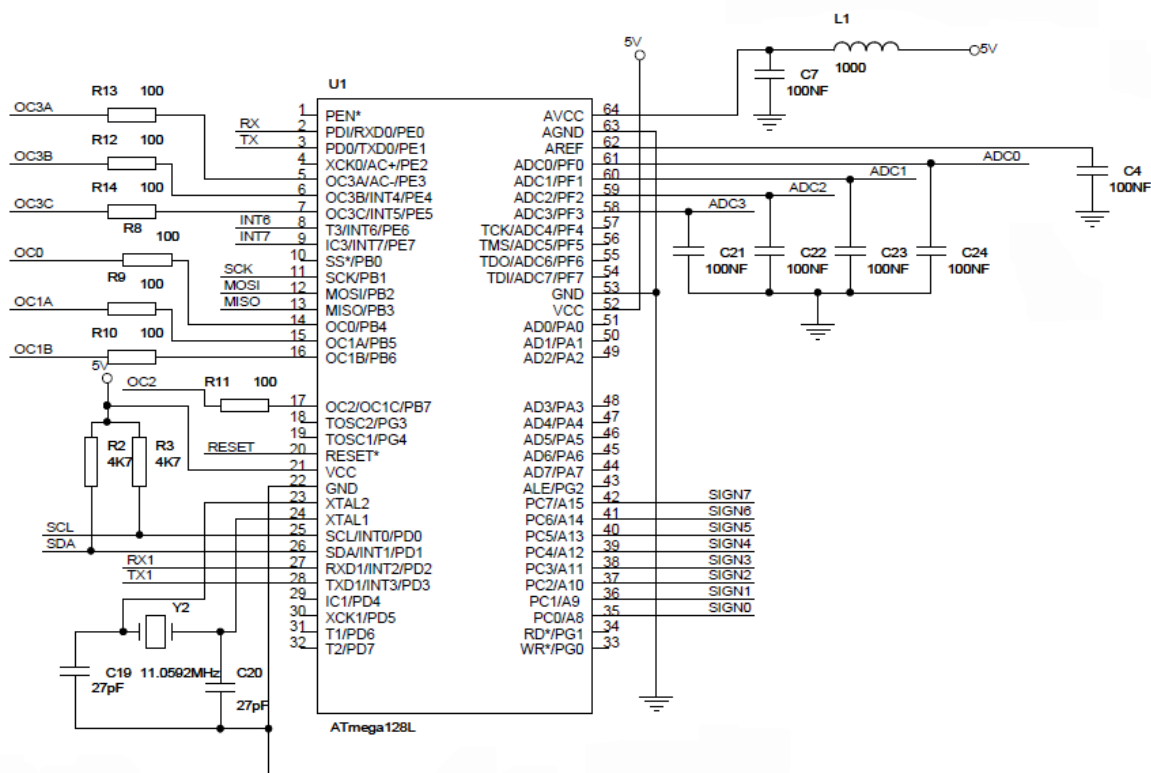
Řídicí deska zajišťuje komunikaci s řídicím počítačem pomocí USB, nastavuje motory a přijímá data ze senzorů. Požadavky na řídicí desku byly následující:

- Podpora UART pro komunikaci s ostatními periferiemi
- Podpora I^2C sběrnice pro komunikaci se senzory
- Podpora A/D převodníku pro měření hodnoty výstupního napětí z infračervených dálkoměrů
- Výpočet a předzpracování odometrie
- Přizpůsobení velikosti desky rozměrům robotu a dobrá dostupnost konektorů

4.3.1 Schéma

Kompletní schéma řídicí desky je součástí přílohy A. Schéma je možné rozdělit do čtyř částí: zapojení MCU, komunikační rozhraní, napájecí a konektorová část.

Základem řídicí desky je ATmega128, která patří do skupiny AVR mikroprocesorů od firmy Atmel. Skupina mikroprocesorů AVR je velice rozšířená, chlubí se zejména velkou pamětí, vysokým stupněm integrace a velmi nízkou spotřebou. ATmega128 byla vybrána



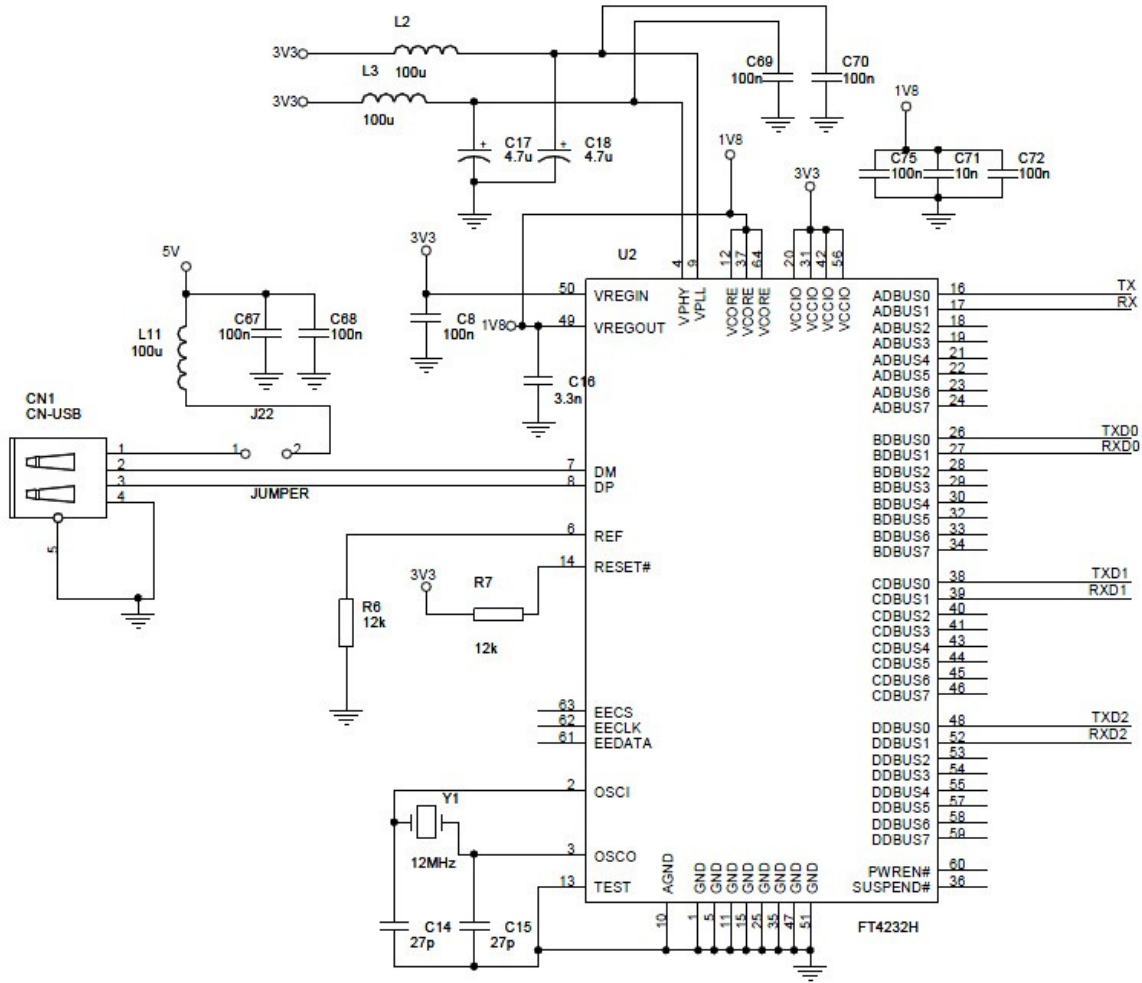
Obrázek 13: Zapojení mikroprocesoru ATmega128

zejména pro svoje vlastnosti, možnosti připojení 2 sériových komunikačních rozhraní, podpoře I^2C sběrnice, 6 časovačům schopných generovat PWM signál, 8 kanálům 10b A/D převodníku a dostatku volných I/O pinů pro připojení různých periférií. Na obrázku 13 je možné vidět část zapojení mikroprocesoru ATmega128 ze schématu řídicí desky. Mikroprocesor byl zapojen dle katalogového listu [14], pro dosažení standardních komunikačních rychlostí na sériové lince byl přidán krystal $Y2 = 11,0592$ MHz. Dalším významným blokem je zapojení FT4232H (převodník USB na sériovou linku) na obrázku 14. Tento typ byl vybrán, protože umožňuje připojení až čtyř sériových portů a ovladače převodníku jsou pro všechny systémy dostupné na stránkách výrobce [15].

Napájení je řešeno dvěma způsoby. Desku je možné napájet přímo z USB nebo externě pomocí konektoru CON10A. K výběru způsobu napájení slouží zkratovací propojka J22. Po připojení zkratovací propojky je řídicí deska napájena přes USB. Napájení mikroprocesoru 5 V a FTDI 3,3 V je řešeno pomocí stabilizátoru LM317, kde kombinací odporů $R5$ a $R4$ určíme stabilizaci napětí (8):

$$V_{OUT} = V_{REF} \left(1 + \frac{R4}{R5}\right) + I_{ADJ} R4 \quad (8)$$

V_{OUT} je výstupní napětí stabilizátoru, V_{REF} je referenční napětí stabilizátoru, které je dáno



Obrázek 14: Zapojení FTDI

dle katalogového listu [16] $V_{REF} = 1,25 V$. V katalogovém listu dále najdeme hodnotu chybového proudu $I_{ADJ} = 100 \mu A$ (ovlivňující zpětnovazební dělič, který způsobuje chybu výstupního napětí) a hodnotu rezistoru $R_5 = 240 \Omega$, kterou standartizujeme dle odporové řady E12 na hodnotu $R_5 = 220 \Omega$. Dosadíme do vzorce 8 a vypočteme odpor R_4 (9):

$$3,3 = 1,25\left(1 + \frac{R_4}{220}\right) + 0,0001R_4 \rightarrow R_4 = 355 \Omega, \quad (9)$$

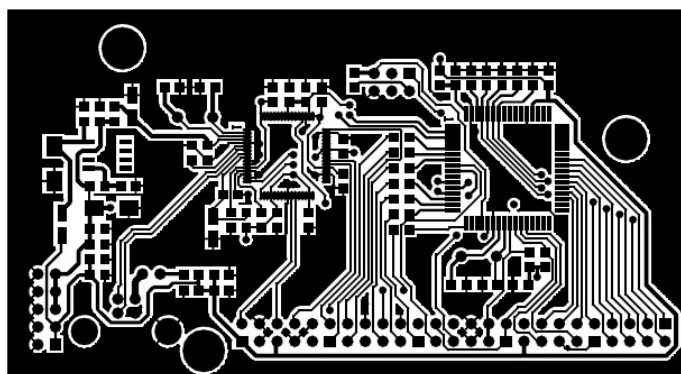
hodnotu $R_4 = 355 \Omega$ upravíme podle odporové řady E12 $R_4 = 390 \Omega$ a ověříme výstupní napětí dosazením do vzorce 8:

$$V_{OUT} = 1,25\left(1 + \frac{390}{220}\right) + 0,0001 \cdot 390 \rightarrow V_{OUT} = 3,5 V, \quad (10)$$

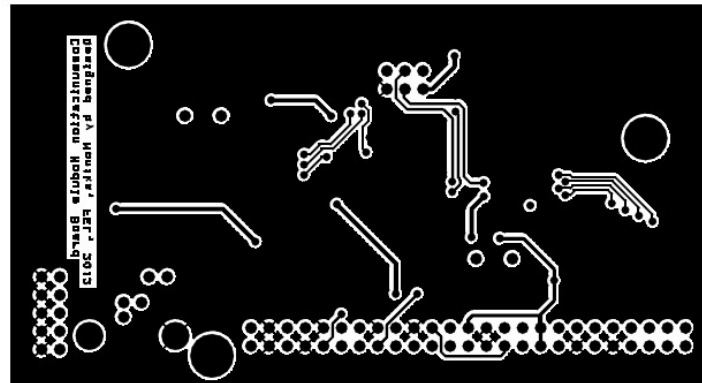
Rozdíl vypočteného a požadovaného napětí je v toleranci všech součástek. Pokud by měla být hodnota výstupního napětí přesnější, musela by být zvolena přesnější odporová řada. Na vstupu stabilizátoru se dále nachází dioda D1, jako ochrana proti přepólování, a filtrační kondenzátory C9 a C13. Poslední konektorová část schématu byla navržena na základě konstrukčních požadavků na desku, podrobněji je popsáno v následující podkapitole návrhu plošného spoje.

4.3.2 Návrh plošného spoje

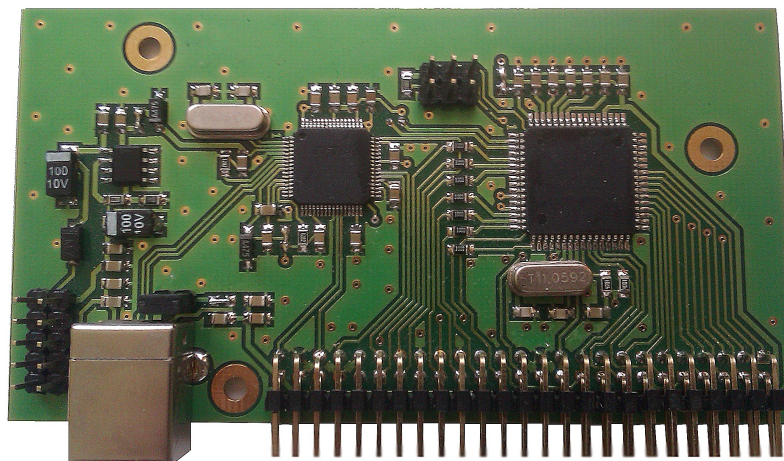
Při návrhu plošného spoje byl kladen důraz na rozměry desky, aby mohla být umístěna mezi motory do těla robota. Dále bylo důležité rozmístění konektorů tak, aby byly dobře dostupné. Následovalo rozmístění součástek s ohledem na funkci a pravidla návrhu plošného spoje. Filtrační kondenzátory a krystaly byly umístěny vždy co nejbližší k součástce, aby nedocházelo k nežádoucímu rušení signálu. Z důvodu dostatečného uzemnění a minimalizování počtu propojek byla vytvořena oboustranná deska s rozlitou zemí. Na obrázku 15 je vidět vrchní vrstva desky plošných spojů a na obrázku 16 spodní vrstva. Vyrobená a plně osazená deska je zobrazena na obrázku 17.



Obrázek 15: Vrchní vrstva řídicí desky



Obrázek 16: Spodní vrstva



Obrázek 17: Kompletně osazená řídicí deska

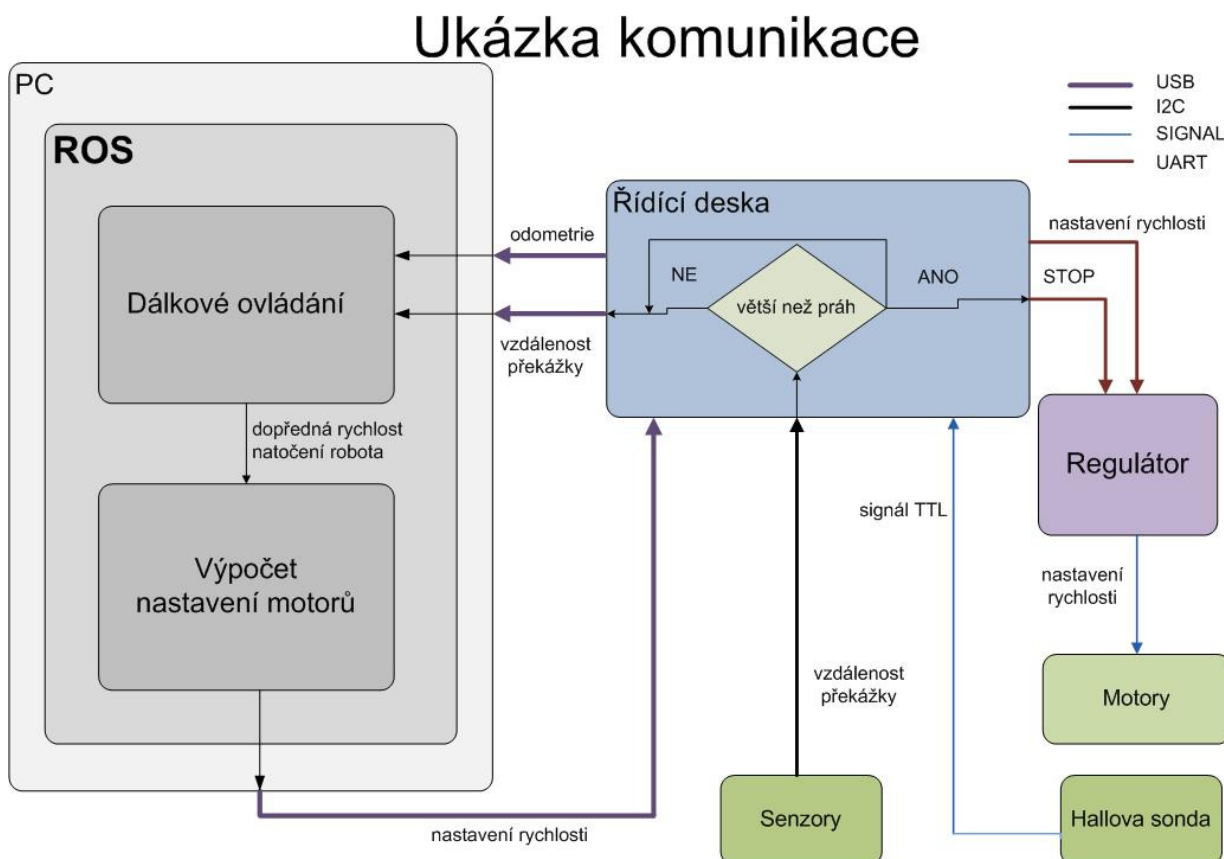
5 Software

Tato kapitola popisuje vývoj a návrh softwaru, obsluhu jednotlivých senzorů, vzájemnou interakci, zpracování dat ze senzorů, robotický operační systém ROS, návrh a implementaci algoritmů na rozpoznávání obrazu. Celý software byl navržen v podobě subsumpční architektury již zmiňované v úvodní kapitole 1. Vrstvy mají vzestupnou prioritu. Vyšší vrstvy jsou postavené nad nižšími vrstvami z důvodu vytvoření komplexnějšího chování. Fungování celého systému je výsledkem interakce mezi vrstvami. Princip interakcí v robotu je znázorněn ve schématu na obrázku 18. Nejspodnější vrstvu zajišťuje řídicí deska robota, kde je naprogramovaný bezpečnostní práh vzdálenosti překážek. Jakmile se překročí bezpečnostní práh, robot zastaví. Nad touto vrstvou je naimplementována vrstva obsluhy, která po zastavení motorů rozhodne podle režimu, ve kterém se daný robot nachází, co se bude dít dál. V režimu dálkového ovládání bude čekat na vyjádření obsluhy, v režimu vyhýbání se překážkám sama rozhodne, kam se vyhnout, a v režimu navigace dle kamery předá informaci nadřazenému softwaru a čeká na vyhodnocení dané situace.

Pro každý senzor byl napsán ukázkový software, který zprostředkuje komunikaci a přehledně zobrazí zpracovaná data v konzoli. Díky interakci v reálném čase je možné bezproblémově vyzkoušet funkce senzoru, jeho rozsah a chování v daném prostředí.

5.1 Ultrazvukový dálkoměr

Byl vybrán modul SRF08, popsán v kapitole 4.1.1, který komunikuje po sběrnici I^2C . Obousměrná komunikace na sběrnici I^2C probíhá po dvou vodičích, SDA (datový signál) a SCL (hodinový signál). Z elektrického hlediska jsou oba signály zapojeny jako otevřený kolektor. Maximální délka vodičů je dána jejich nejvyšší přípustnou kapacitou 400 pF. Každý vodič musí být připojen jedním pull-up rezistorem na 5 V, tímto způsobem je zajištěna logická úroveň „1“ v klidovém stavu. Při probíhajícím přenosu jsou na SDA vysílány jednotlivé datové bity, přičemž platí pravidlo, že logická úroveň na SDA se smí měnit pouze, je-li SCL v úrovni L. Toto pravidlo je porušeno ve dvou speciálních případech, a to při vysílání podmínek START a STOP, které se používají k zahájení komunikace a k ukončení přenosu. Maximální frekvence signálu SCL je podle verze I^2C 100 kHz nebo 400 kHz. Umožňuje propojení až 128 různých zařízení. Sběrnice rozděluje připojená zařízení na řídicí (master – zahajuje a ukončuje komunikaci, generuje hodinový signál SCL) a řízené (slave – zařízení adresované masterem)[17]. Pro komunikaci se senzory byla využita volně dostupná knihovna ze stránek [18]. Pomocí této knihovny byl napsán obslužný program pro senzor SRF08, který zajišťuje vyčtení dat z registru a jejich zpracování. Ukázka vyčtení dat ze senzoru je v příloze B.



Obrázek 18: Princip interakcí v robotu

5.2 Senzor CHR-6d

Digitální inerciální jednotka CHR-6d komunikuje po sériové lince rychlostí 115200 bd. Popis a funkce jednotky je popsán v kapitole 4.1.3. Pořadí přenosu datových bitů po sériové lince je od nejméně významného bitu (LSB) po bit nejvýznamnější (MSB). Počet datových bitů je volitelný, obvykle se používá 8 bitů. Logický stav „0“/„1“ přenášených dat je reprezentován pomocí dvou možných úrovní napětí, u jednotky CHR - 6d je napěťová úroveň 3,3 V, ale je 5 V tolerantní. Rozhraní sériové linky je třívodičové (příjem RxD, vysílání TxD a společná zem GND). K obsluze tohoto senzoru byl napsán obslužný program, který umožňuje kalibraci, detekuje přijatý paket a vyčte jednotlivé hodnoty, které přehledně i s popisem zobrazí na konzoli. Ke komunikaci se sériovou linkou a pro zobrazení dat na konzoli využívá volně dostupné knihovny pro AVR ze stránek [19]. Kompletní program je přiložen na CD.

5.3 Sledování barevných objektů

Rozpoznávání objektů v obraze se věnuje celý vědní obor a v literatuře existuje řada algoritmů. Na robot Blackshot byla implementována metoda založená na rozpoznávání objektů na základě barevné odlišnosti od okolí. Metoda je založena na odhadu pozice objektu, analogickému výpočtu těžiště tělesa. K výpočtu těžiště je potřeba vyhodnotit podobnost pixelu se vzorovou barvou. Nejdříve zadáme vzorovou barvu, danou hodnotami jednotlivých barevných složek r, g, b , a barvu hodnoceného pixelu na pozici x, y danou hodnotami r_{xy}, g_{xy}, b_{xy} . Výsledkem vzorce, který ohodnocuje pixel s obrazovými souřadnicemi x, y , je míra jeho podobnosti se vzorem, tedy číslo $p_{xy} = f(r, g, b, r_{xy}, g_{xy}, b_{xy})$. Výpočet "těžiště" provedeme podle vzorců 11:

$$t = \sum_{x=1}^w \sum_{y=1}^h p_{xy}, \quad t_x = \frac{1}{t} \sum_{x=1}^w \sum_{y=1}^h p_{xy}x, \quad t_y = \frac{1}{t} \sum_{x=1}^w \sum_{y=1}^h p_{xy}y, \quad (11)$$

kde t_x a t_y přibližně udávají pozici středu hledaného objektu v obraze. Velkou výhodou tohoto algoritmu je rychlost. Nevýhodou je, že pokud robot uvidí více objektů stejné barvy, budou výsledné souřadnice mezi nimi. To odstraníme vyloučením pixelů, které jsou příliš daleko. Pokud vypočtená vzdálenost bude větší než daná hodnota t , bude výsledkem f hodnota 0, jinak bude výsledkem 1. Pro potlačení vlivu osvětlení celý výpočet provedeme v modelu HSV barevného prostoru. Tento barevný prostor charakterizuje jednotlivé pixely třemi parametry: barevným tónem (Hue, h), sytostí (Saturation, s) a jasnem (Value, v). Změna intenzity osvětlení se projeví zejména ve složce Value. Před porovnáním podobnosti pixelů převedeme oba pixely do HSV modelu a potlačíme vliv jasové složky 12:

$$v_{xy} = \sqrt{\frac{(h_{xy} - h)^2}{1} + \frac{(s_{xy} - s)^2}{4} + \frac{(v_{xy} - v)^2}{16}} \\ v_{xy} \leq t \rightarrow p_{xy} = 1, v_{xy} > t \rightarrow p_{xy} = 0. \quad (12)$$

Výsledný objekt bude zobrazen jako oblast pixelů s podobnou barvou. Robot sleduje daný objekt tak, že se svým pohybem snaží umístit těžiště objektu do středu obrazu. K řízení rychlosti je použit princip zpětné vazby. K pohybu robota je použita úhlová a dopředná rychlost, vypočteny dle vzorců 13. Celý program včetně videí z testování je přiložen na CD. Tato metoda byla popsána v cyklu článků [20], [21], [22] a [23].

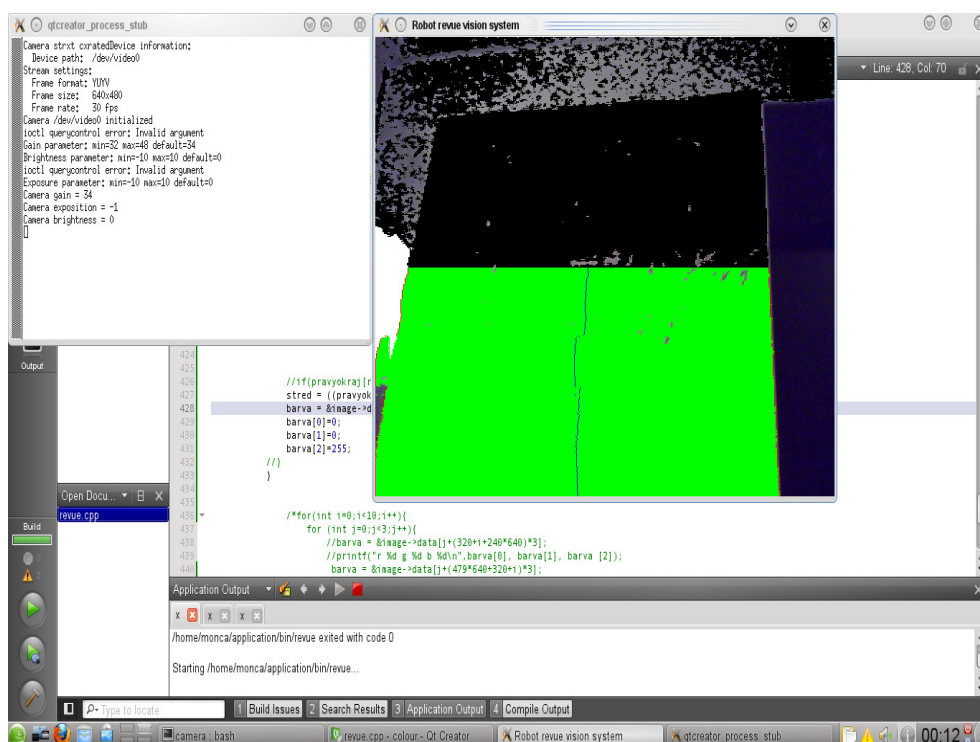
$$v = k_v \cdot d_y \quad (13)$$

$$\omega = k_\omega \cdot d_x \quad (14)$$

5.4 Sledování cesty

Algoritmus na hledání cest funguje obdobně jako algoritmus na sledování objektů. Využívá předpokladu, že barva cesty je odlišná od okolí. Robot se naučí danou barvu cesty v modelu HSV pro potlačení vlivu osvětlení. Určíme minima a maxima jednotlivých složek.

Vytvoříme nový pixel o složkách (min_h, min_s, min_v) a pixel o složkách (max_h, max_s, max_v) . Podle těchto pixelů určíme minimum a maximum v RGB. Následně projdeme celý obrázek, ohodnotíme jednotlivé pixely p , zjistíme cestu v obraze a určíme okraje cesty (poslední pixely s podobnou barvou). Z okrajů cesty vypočítáme střed cesty a podle toho, o kolik se liší střed obrazu od středu cesty, určíme úhlovou rychlost robotu. Dopředná rychlost robotu je vypočtena podle vzdálenosti cesty v obraze (podle zbývajících počtu řádků pixelů cesty). Zároveň určíme minimální šířku cesty, odpovídající konstrukčním rozměrům robotu. Pokud je menší než daná šířka pixelů, robot ji nevidí. Celý program včetně videí z testování je přiložen na CD.



Obrázek 19: Detekce cesty

6 ROS

Tato kapitola popisuje základní funkce robotického operačního systému, vývoj a návrh dálkového ovládání robotu s využitím tohoto systému a jeho modulů. Je zde popsán souborový systém, jeho členění a vzájemné vazby. Dále způsob propojení modulů a jejich vzájemná interakce.

Robotický operační systém ROS je nově vznikající platforma, která slouží pro usnadnění vývoje robotických aplikací. Její hlavní výhodou jsou standardizovaná rozhraní pro komunikaci mezi moduly. Moduly představují nezávislé programy s definovanými vstupními a výstupními událostmi. Událostí je například publikování odometrie a její zanesení do map. Poskytuje také abstrakci hardware a ovladače různých zařízení. Je distribuován pod otevřenou licencí BSD. Většina funkcí je přehledně zdokumentována v ROS wiki [24], kde najdeme také spoustu návodů a cyklus článků „krok za krokem“ pro začátečníky, kde jsou přehledně na praktických příkladech ukázány základní funkce důležité pro práci.

6.1 Základní stavební prvky

6.1.1 Vysvětlení základních pojmů

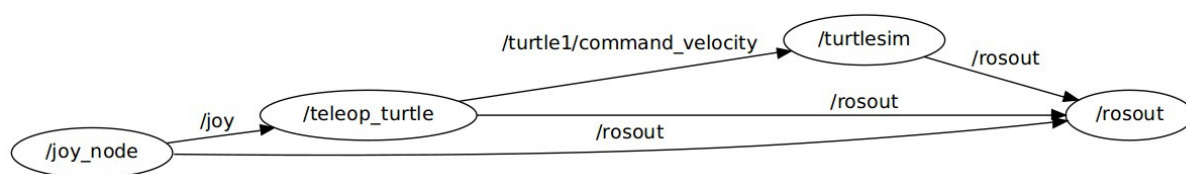
- Package - balíček, nejnižší úroveň souborového systému. Obsahuje například knihovny, nástroje, konfigurační soubory atd.
- Manifest - popis balíčku, definuje závislosti mezi balíčky (který balíček musí být zároveň nainstalován), dále obsahuje informace o autorovi a licenci.
- Stack - soubor balíčků, které poskytují společné funkce. Například soubor balíčků pro navigaci v souborovém systému.
- Stack Manifest - popis Stack, stejné jako Manifest akorát pro Stack.

6.2 Dálkové ovládání

Dálkové ovládání robotu je realizované pomocí joysticku, jehož parametry najdeme na stránkách [25]. Model robotu je s joystickem spojen pomocí systému ROS. Při implementaci do systému ROS bylo využito návodů pro začátečníky, dostupné na stránkách [24]. Při postupném plnění pokynů v návodu si vytvoříme model robotu v podobě želvy ovládané klávesnicí. Ovládání pomocí klávesnice poté nahradíme ovládáním joystickem. Pro tento účel se využívá ovladače joystick drivers, který plní funkci zpřístupnění joysticků kompatibilních s operačním systémem Linux. Dále potřebujeme vyčíst informace o aktuálním stavu tlačítek. K tomu slouží balíček „joy“, který již publikuje zprávy „Joy“ obsahující momentální stav všech tlačítek joysticku. Bohužel naše želva poslouchá jen jednoduché

zprávy říkající pootočit se, jed' rovně, dozadu atd. Z tohoto důvodu musíme vytvořit uzel, překládající zprávy typu „Joy“ na příkazy pro želvu. Celý kód balíčku včetně vysvětlení jednotlivých příkazů je přiložen v příloze C. Postupovat lze podle návodu na stránkách [26]. Hierarchie komunikace je lépe znázorněná na obrázku 20.

6.3 Ukázka závislosti balíčků



Obrázek 20: Ukázka závislosti balíčků

7 Experimenty

7.1 Odometrie

Při testech odometrie byla měřena přesnost výpočtu ujeté vzdálenosti robotu. Byla odzkoušena jízda vpřed bez zatáčení, protože otáčky jsou měřeny pouze na jednom kole robotu. První testy probíhaly ve vnitřním prostředí na hladkém povrchu, odometrie byla zkalibrována na 1 m. Výsledky prvních testů přesnosti odometrie jsou zobrazeny v tabulce 4.

Id	Vzdálenost [m]	chyba [cm]
1	1	0
2	15	20
3	64	100

Tabulka 4: Přesnost odometrie s kalibrací na 10 m

Z výsledků je vidět, že na 64 m je chyba odometrie 1 m. Procentuální chyba je sice pouze 1,56 %, ale přesto byla nepřesnost příliš velká, protože se jednalo o jízdu na hladkém povrchu s neustále stejným zatížením kol, bez skluzu a prokluzu kol. Z tohoto důvodu byla provedena kalibrace ve venkovním prostředí na asfaltovém chodníku při jízdě na 100 m vzdálenost. Výsledky dalších testů jsou ukázány v tabulce 5.

Id	Vzdálenost [m]	chyba [cm]
1	10	1
2	50	3
3	80	4
4	100	16

Tabulka 5: Přesnost odometrie po kalibraci na 100 m

Z výsledků je vidět podstatné zlepšení. Procentuální chyba odometrie je pouze 0,16 %. Protože se jedná pouze o 1,5 otáčky kola, mohla být tato chyba způsobena nepřesností měření, pozdním odečtem pozorovatele či pozdním zastavením robotu.

7.2 Sledování cesty

První testy na sledování cesty proběhly v noci za pouličního osvětlení na cestě široké 1,5 m, robot ujel 20 m po cestě aniž by sjel z cesty. Bohužel po 20 m bylo osvětlení tak slabé,

že robot nerozpoznal cestu. Ale přesto dosáhl výborného výsledku za daných podmínek, světelné podmínky jsou vidět na obrázku 26.



Obrázek 21: Fotografie z testu na sledování cesty

Další testy probíhaly ve dne, robot dokázal dojet až na konec cesty dlouhé 60 m a široké 1,5 m.

7.3 Sledování barevných objektů

Pro odzkoušení správné implementace algoritmu byl proveden test ve vnitřním i venkovním prostředí na sledování barevných objektů. Sledování barevných objektů probíhalo bez problému, robot dokáže sledovat míček o průměru 7 cm na vzdálenost 10 m a jet za ním. Pokud objekt ztratí, robot se točí na místě kolem své osy a hledá objekty podobné barvy.

7.4 Test subsumpční architektury

V rámci testování algoritmů na sledování cesty a barevných objektů byla ověřena bezpečnost robota a správný návrh subsumpční architektury. Test byl proveden za pomoci objektu, který měl stejnou barvu jako cesta. Robot ho kamerou nerozpoznal a jel vpřed v tu chvíli zareagovala nižší vrstva na vzdálenost překážky přijatou od ultrazvukového dálkoměru a robot zastavil. Podobný test byl proveden i při sledování objektů. Objekt byl umístěn za překážku, ale tak, aby ho robot viděl a při tom nesměl nabourat. Oba testy



Obrázek 22: Fotografie z testu na sledování barevných objektů

byly úspěšné. Testy byly řádně zdokumentovány, fotografie jsou přiloženy v příloze D, další fotografie a videa na CD.

8 Závěr

Cílem této bakalářské práce bylo navrhnout a realizovat řídicí systém mobilního robotu schopného překonávat obtížný terén. Pro kompletní realizaci bylo nutné vybrat vhodné senzory, navrhnout a realizovat elektroniku robotu, napsat obslužný firmware a software. Řídicí systém, popsáný v této práci, je značně univerzální a snadno přenositelný na jinou robotickou platformu.

Před samotným návrhem bylo potřeba se seznámit s problematikou řízení mobilních robotů, zpracování sensorických dat a odhadování pozice z proprioceptivních sensorů. Tato problematika je popsána v kapitolách 2 a 3. Následně byly vybrány vhodné senzory, umožňující orientaci robotu v prostoru. Výběrem těchto sensorů byly definovány požadavky na řídicí elektroniku robotu. Od toho se vyvíjel výběr řídicího mikroprocesoru a způsob komunikace. Celý návrh a samotná realizace řídicí elektroniky robotu jsou popsány v kapitole 4.

Kapitola 5 pojednává o vytvoření ukázkového programu, který zajišťuje komunikaci se senzory a přehledně zobrazuje sejmутá sensorická data. Tyto obslužné rutiny umožňují otestování a zkalibrování daného senzoru před jeho připojením k robotu. Dále pojednává o implementaci metod pro sledování objektů a navigaci po cestě.

Celý systém byl začleněn do frameworku ROS, který mimo jiné umožnil dálkové ovládání robotu. Nedílnou součástí této práce byly experimenty v reálném prostředí, kde byla ověřena spolehlivost řídicího systému, přesnost výpočtu odometrie a otestována schopnost autonomní navigace. Z výsledků experimentů je vidět, že robot splňuje dané požadavky a je schopný pohybu ve vnitřních i venkovních prostorech. Výsledný řídicí systém je využitelný i pro jiné robotické platformy.

Reference

- [1] SRI's Artificial Intelligence Center. Shakey. [Online; navštíveno 7-4-2012].
- [2] Rodney A. Brooks. A robust layered control system for a mobile robot, 1985. [Online; navštíveno 7-4-2012].
- [3] P. Ripka, S. Ďaďo, M. Kreidl, and J. Novák. *Senzory a převodníky*, 2005.
- [4] Wikipedia. Hallův jev— Wikipedia, the free encyclopedia, 2011. [Online; navštíveno 13-3-2012].
- [5] Antonín Vojáček. Princip optických enkodérů polohy pro řízení motorů, 2006. [Online; navštíveno 10-4-2012].
- [6] Antonín Vojáček. Jak pracují nové 3D MEMS akcelerometry Freescale ?, 2007. [Online; navštíveno 9-5-2012].
- [7] Wikipedia. Coriolisova síla— Wikipedia,, 2012. [Online; navštíveno 12-4-2012].
- [8] Antonín Vojáček. Integrované MEMS GYROSKOPY, 2009. [Online; navštíveno 12-4-2012].
- [9] Wikipedia. South-pointing chariot— Wikipedia, the free encyclopedia, 2012. [Online; navštíveno 14-4-2012].
- [10] Zbyněk Winkler. *Odometrie*, 2005. [Online; navštíveno 15-4-2012].
- [11] srf08 katalogový list.
- [12] CHR-6D katalogový list.
- [13] MMP - 5 uživatelský manuál.
- [14] ATmega128 katalogový list.
- [15] Future Technology Devices International Limited. FTDI chip, 2009. [Online; navštíveno 5-3-2012].
- [16] LM317 katalogový list.
- [17] Wikipedia. *i²c*— Wikipedia,, 2012. [Online; navštíveno 7-5-2012].
- [18] Pascal Stang. *Procyon AVRlib*, 2005. [Online; navštíveno 5-3-2012].
- [19] Peter Fleury. *AVR Software*, 2006. [Online; navštíveno 5-3-2012].
- [20] T. Krajník, V. Vonásek, J. Sova, and H. Szücsová. Jak to vidí roboti - III. *ROBOT REVUE*, 5/10(1):36–38, květen 2010.

-
- [21] T. Krajník and V. Vonásek. Jak to vidí roboti - IV. *ROBOT REVUE*, 6/10(1):37–38, červen 2010.
- [22] T. Krajník, H. Szücssová, M. Svědihrová, J. Chudoba, and V. Vonásek. Jak to vidí roboti - V. *ROBOT REVUE*, 8/10(1):31–35, srpen 2010.
- [23] J. Sova, T. Krajník, and M. Svědihrová. Jak to vidí roboti - VI. *ROBOT REVUE*, 9/10(1):34–37, září 2010 2010.
- [24] Ken Conley a kolektiv. ROS/Tutorials, 2012. [Online; navštíveno 10-3-2012].
- [25] Logitech. Rumblepad 2, 2012. [Online; navštíveno 20-2-2012].
- [26] Erica Kane a kolektiv. Writing a teleoperation node for a linux-supported joystick, 2011. [Online; navštíveno 10-3-2012].

Ukázka programu pro komunikaci se senzory pomocí sběrnice I^2C

```
int main(void) {
unsigned int nBuff; // deklarace proměnné
uartInit(); // inicializace sériové linky
rprintfInit(uart0AddToTxBuffer); // definuje výstup funkce rprintf na uart0
// inicializace funkce, která zprostředkovává zobrazení výstupu na konzoli
vt100Init();
vt100SetCursorMode(0); // zapnutí kurzoru
// nastavení kurzoru v konzoli na pozici 1,1 (první řádek, první znak)
vt100SetCursorPos(1,1);
rprintf("Vzdalenost:"); // popis výsledku, vytisknutí na zásobník
uartSendTxBuffer(0); // odeslání zásobníku na UART0
uartFlushReceiveBuffer(0); // vymazání obsahu zásobníku
i2c_init(); // inicializace I2C sběrnice
_delay_ms(20); // čekací cyklus 20 ms
while(1) { // začátek nekonečné smyčky
i2c_start(0xE0); // zapne měření, vyšle start bit na adresu E0
i2c_write(0x00); // nastavení registru
i2c_write(0x51); // zapíšeme do registru, volba jednotek cm
i2c_stop(); // vyšle stopbit
_delay_ms(200); // počká na výsledek měření
nBuff = 0; // vynuluje proměnnou
i2c_start(0xE0); // start bit
i2c_write(0x02); // nastavení registru
i2c_rep_start(0xE1); // povolí čtení
// přečte hodnotu z registru a přičte ji do proměnné
nBuff += i2c_readAck();
// posune o 8 znaků a tím provede převod z hexadecimální do desítkové soustavy
nBuff <<= 8;
// přečte hodnotu z registru a přičte ji do proměnné (přičte spodní bajt)
nBuff += i2c_readNak();
i2c_stop(); // vyšle stopbit
// nastaví kurzor na pozici 1,13, 1. Řádek 13 znak,
// tak aby se nepřepsal popis měřené veličiny
vt100SetCursorPos(1,13);
rprintfNum(10, 4, 0, ' ', nBuff); // vytiskne proměnnou na zásobník
uartSendTxBuffer(0); // odeslání zásobníku na UART0
uartFlushReceiveBuffer(0); // vymazání obsahu zásobníku
} //konec nekonečné smyčky
}
```

Ukázka balíčku pro komunikaci s joystickem

```
//implementace potřebných knihoven
#include <ros/ros.h>
#include <turtlesim/Velocity.h> // pro generování příkazů pro želvu
#include <sensor_msgs/Joy.h> // abychom mohli přijímat zprávy joy

// vytvoření třídy TeleopTurtle, která definuje funkci joyCallback,
// pro práci se zprávami typu joy
//vytvoření uzlů handle, publisher a subscriber
class TeleopTurtle
{
public:
    TeleopTurtle();

private:
    void joyCallback(const sensor_msgs::Joy::ConstPtr& joy);

    ros::NodeHandle nh_;

    int linear_, angular_;
    double l_scale_, a_scale_;
    ros::Publisher vel_pub_;
    ros::Subscriber joy_sub_;

};

//inicializace parametrů, proměnných linear a angular,
//které bude poslouchat želva
TeleopTurtle::TeleopTurtle():
    linear_(1),
    angular_(2)
{

    nh_.param("axis_linear", linear_, linear_);
    nh_.param("axis_angular", angular_, angular_);
    nh_.param("scale_angular", a_scale_, a_scale_);
    nh_.param("scale_linear", l_scale_, l_scale_);

//vytvoření generování zpráv pro želvu
    vel_pub_ = nh_.advertise<turtlesim::Velocity>("turtle1/command_velocity", 1);
```

```
// uzel na zpracování příchozích zpráv joy pro želvu
joy_sub_ = nh_.subscribe<sensor_msgs::Joy>
("joy", 10, &TeleopTurtle::joyCallback, this);

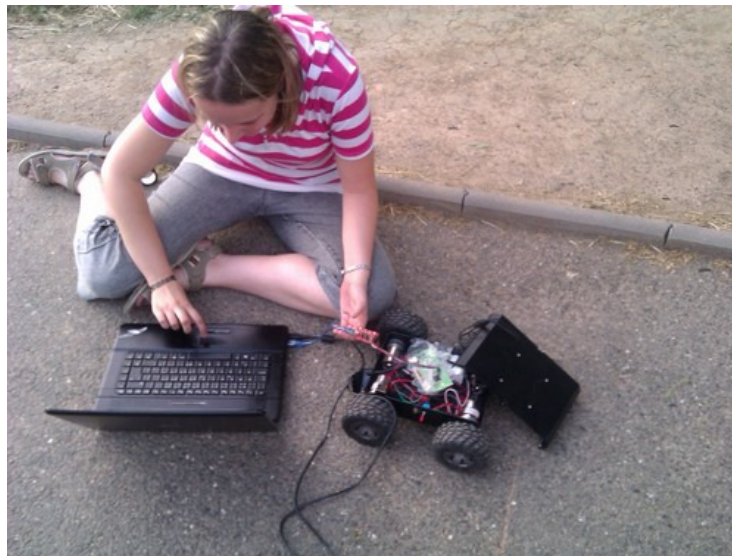
}
//přepočet zpráv z joysticku na zprávy pro želvu

void TeleopTurtle::joyCallback(const sensor_msgs::Joy::ConstPtr& joy)
{
    turtlesim::Velocity vel;
    vel.angular = a_scale_*joy->axes[angular_];
    vel.linear = l_scale_*joy->axes[linear_];
    vel_pub_.publish(vel);
}

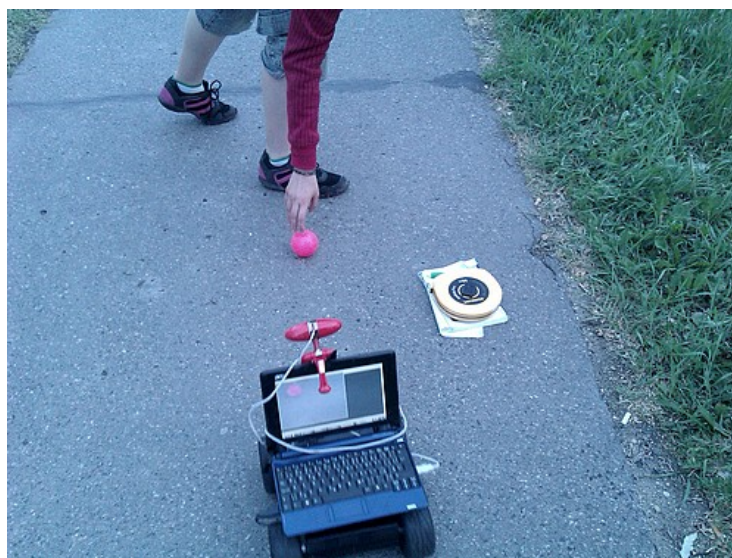
//inicializace ros uzlu
int main(int argc, char** argv)
{
    ros::init(argc, argv, "teleop_turtle");
    TeleopTurtle teleop_turtle;

    ros::spin();
}
```

Fotografie z testů



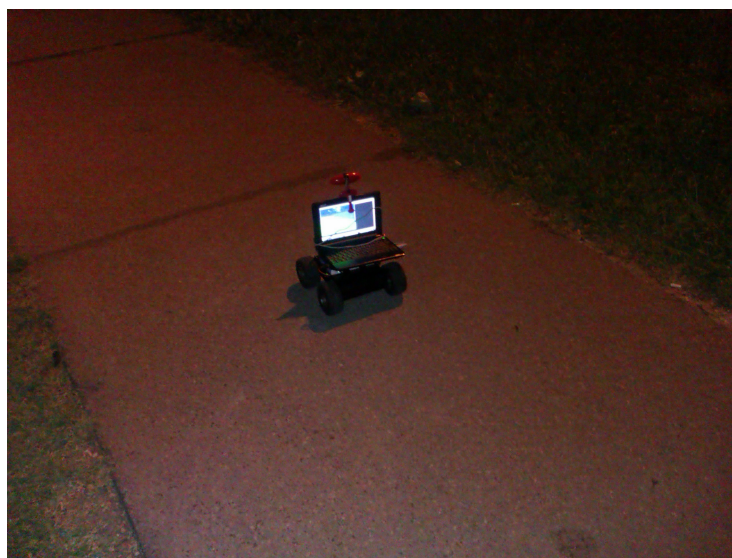
Obrázek 23: Kalibrace odometrie v terénu



Obrázek 24: Fotografie z testu na sledování objektů



Obrázek 25: Fotografie z testu na sledování cesty



Obrázek 26: Fotografie z testu na sledování cesty za pouličního osvětlení

Obsah přiloženého CD

V tabulce 6 jsou uvedena jména všech kořenových adresářů přiloženého CD s popisem obsahu.

Jméno adresáře	Popis obsahu
bp	bakalářská práce v pdf formátu
mcu	zdrojové kódy pro mikroprocesor
katalog	katalogové listy k použitým součástkám
elektronika	schéma zapojení a deska plošných spojů řídicí desky
navigace	algoritmy na sledování cesty a objektů
videa	videa z testování
foto	fotografie z testování

Tabulka 6: Obsah přiloženého CD