

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

**Řídicí systém pro robotickou platformu
Lego Mindstorms 2.0**

Michael Baroň

© 2015 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačního inženýrství

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Michael Baroň

Informatika

Název práce

Řídící systém pro robotickou platformu Lego Mindstorms 2.0

Název anglicky

A Control system for the robotic platform Lego Mindstorms 2.0

Cíle práce

Cílem práce je vytvoření softwareové aplikace pro stolní počítač umožňující ovládání robota Lego Mindstorms 2.0. Komunikace mezi robotem a pracovní stanicí bude probíhat přes komunikační rozhraní Bluetooth.

Metodika

Studium odborné literatury

Návrh vhodného řešení

Implementace řešení

Ověření řešení v laboratorních podmínkách

Zhodnocení výsledků

Doporučení dalších kroků

Závěr

Doporučený rozsah práce

Návrh komunikačního rozhraní lego Mindstorms 2.0

Klíčová slova

Lego Mindstorms, robot, komunikační rozhraní Bluetooth

Doporučené zdroje informací

Alan Cooper: The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity (Feb 24, 2004), ISBN-13: 978-0672326141

Alan Cooper and Robert Reimann: About Face 2.0: The Essentials of Interaction Design (Mar 17, 2003), ISBN-13: 978-0764526411

Alan Cooper, Robert Reimann and David Cronin: About Face 3: The Essentials of Interaction Design (May 7, 2007), ISBN-13: 978-0470084113

Jakob Nielsen's Alertbox, April 14, 2003: Paper Prototyping

Kim Goodwin (Author), Alan Cooper (Foreword): Designing for the Digital Age: How to Create Human-Centered Products and Services, ISBN-13: 978-0470229101

Předběžný termín obhajoby

2015/06 (červen)

Vedoucí práce

Ing. Josef Pavlíček, Ph.D.

Elektronicky schváleno dne 10. 11. 2014

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 10. 11. 2014

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 14. 03. 2015

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Řídící systém pro robotickou platformu Lego Mindstorms 2.0" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 16.3.2015

Poděkování

Rád bych touto cestou poděkoval Ing. Josefu Pavlíčkovi, Ph.D. za jeho rady, jež mě směřovaly k cíli a za úžasné motivující přístup obecně, díky kterému jsem si vypracování této práce velmi užil.

Řídící systém pro robotickou platformu Lego Mindstorms 2.0

A Control system for the robotic platform Lego Mindstorms 2.0

Souhrn

Hlavní náplní práce bylo navrhnout a sestavit robota z produktu Mindstorms společnosti Lego. K němu vznikla aplikace pro počítač, pomocí níž je možné výtvar ovládat, ať už ručně člověkem nebo jakýmkoliv jiným způsobem.

Součástí je i náčrt problematiky okolo robotiky jako vědního oboru, její historie, současnost, budoucí trendy a technologie. Prozkoumán je i stávající trh s komponenty použitelnými, stejně jako Lego Mindstorms 2.0, ke stavbě robotů vlastního návrhu.

Summary

The main objective of the work was to design and build a Mindstorms robot, the product of Lego company. A computer application was created for controlling the creation manually by man or by computer itself.

As a part, there is a reference about robotics as science, its history, present state, future and technological trends. There is a survey about present market with robotic components, which are usable the same way as Lego Mindstorms 2.0.

Klíčová slova: Mindstorms, Lego, robot, software, aplikace, řízení, navigace, orientace

Keywords: Mindstorms, Lego, robot, software, application, control, navigation, orientation

OBSAH

1	ÚVOD	9
2	CÍL PRÁCE A METODIKA	10
3	PŘEHLED ŘEŠENÉ PROBLEMATIKY	11
3.1	HISTORIE ROBOTIKY	11
3.2	SOUČASNOST ROBOTIKY	12
3.2.1	<i>Těžářství</i>	13
3.2.2	<i>Doprava</i>	13
3.2.3	<i>Průzkumnictví</i>	14
3.2.4	<i>Příroda</i>	14
3.2.5	<i>Záchranářství</i>	15
3.2.6	<i>Vojenství</i>	15
3.2.7	<i>Komerční sektor</i>	16
3.2.8	<i>Výzkum</i>	16
3.3	STAVBA FYZICKÉ ČÁSTI ROBOTA	17
3.3.1	<i>Typy strojů</i>	17
3.3.2	<i>Komponenty</i>	18
3.4	TVORBA ŘÍDÍCÍHO SOFTWARE	20
3.4.1	<i>Programování</i>	21
3.4.2	<i>Lego Mindstorms</i>	23
3.4.3	<i>Navigace pomocí cesty</i>	24
3.4.4	<i>Navigace analýzou okolí</i>	25
4	VLASTNÍ PRÁCE	29
4.1	LEGO MINDSTORMS	29

4.1.1	Řídicí jednotka.....	30
4.1.2	Stavba robota	30
4.2	PROGRAMOVACÍ JAZYK	32
4.2.1	Použití v NXT.....	32
4.2.2	Použití s počítačem.....	33
4.3	KOMUNIKACE	33
4.3.1	Příkazy.....	34
4.3.2	Komunikace na straně NXT.....	35
4.3.3	Řešení	36
4.4	NAVIGACE V OKOLÍ	38
4.4.1	Využití senzorů	38
4.4.2	Testovací scéna.....	40
4.4.3	Kamera a navigace.....	42
5	VÝSLEDKY A DISKUZE	45
	ZÁVĚR	47
	SEZNAM POUŽITÝCH ZDROJŮ	48
	PŘÍLOHY.....	52

SEZNAM OBRÁZKŮ

Obrázek 1 Jeden z robotů Williama Greye Waltera.....	12
Obrázek 2 Reakce čidel na polohu na čáře.	24
Obrázek 3 Diagram analýzy a nalezení cesty.....	26
Obrázek 4 Laserová data zachycena průjezdem nejobtížnější pasáže trati.....	27
Obrázek 5 Ukázka rozhodovacích map vytvořených z dat všech senzorů Stanleyho. .	27
Obrázek 6 Balení Lego Mindstorms NXT 2.0.	29
Obrázek 7 Ukázka programu v editoru NXT-G.....	30
Obrázek 8 Podvozek vozítka.....	31
Obrázek 9 Uchycení řídicího modulu NXT.	31
Obrázek 10 Testovací program HelloWorld k otestování funkčnosti leJosu.....	33
Obrázek 11 Displej robota zaznamenávající stavy a příkazy.....	37
Obrázek 12 Počáteční grafické rozhraní aplikace.....	38
Obrázek 13 Pohled na Ultrasonic sensor a zadní nárazník.	40
Obrázek 14 Nachystaná testovací scéna.....	41
Obrázek 15 Webová kamera Trust Primo Webcam.....	42
Obrázek 16 Grafické rozhraní pro odladění kamery.....	43
Obrázek 17 Ukázka použití Canny Edge Detectoru.	44
Obrázek 18 Robot nacházející překážky.....	45
Obrázek 19 Poslední verze uživatelského rozhraní, nalezení překážek.....	46

SEZNAM TABULEK

Tabulka 1 Podpora operačních systémů produkčními řadami společnosti Gumstix. ...	22
Tabulka 2 Příkazy ovládající robota a jejich číselné kódy.....	35

1 ÚVOD

Již od roku 1998, kdy byla veřejnosti představena „inteligentní kostka“ Lego Mindstorms RCX jako vzdělávací zařízení, je produktová řada Mindstorms velmi vyhledávána, jak ukazuje velikost komunity lidí pohybujících se kolem ní. Přesto není samotná společnost Lego nijak známa svými příspěvky do robotiky. Všichni znají produkty pro děti a jejich hračky.

Samotné Lego Mindstorms je stavebnice, ze které lze poskládat programovatelný stroj jakéhokoli tvaru, velikosti a určení. Obsahuje veškeré komponenty, které jsou potřeba k výrobě vlastního plně funkčního stroje. Nejen, že se dodává se všemi základními senzory, ale je také jednoduše programovatelná. Pohled na současné a nebo i historické robotické výtvary nabízí výbornou inspiraci pro všechny, kteří se do podobné činnosti dají.

Je možno dokoupit i mnoho dalších zařízení, které robot dokáže obsluhovat a to nejen od originálního výrobce, ale i od jiných. Už toto dokazuje úspěšnost této série „hraček.“ Navíc se jedná o produkt společnosti Lego, což ukazuje na odolnost a modulárnost.

Jak již název práce napovídá, cílem bude tvorba řídicího systému pro produkt společnosti Lego. Vzniknout by měla počítačová aplikace, pomocí níž bude možné propojit počítač a zmiňovaný produkt. Za cíl je tedy určeno navržení a postavení robota, který se spojí s uživatelským počítačem a pomocí aplikace, vytvořené speciálně pro tento účel, bude ovladatelný, ať už se to týká jeho pohybu nebo nějaké jiné funkce.

Vzhledem k tomu, že s balením přichází také mnoho dalších typu senzorů, nachází se zde velký prostor pro hledání jejich využití. V takovém případě je vhodné experimentovat s více řešeními, aby se našlo to nejvhodnější a nejpoužitelnější.

Otázka ovšem zní: není mnohem výhodnější pro tvorbu robota rovnou využít nabídek jiných společností? I když se Lego stavebnice výborně kompletují, nemusí dostačovat v tak odlišném odvětví, jako je programování a inženýrství.

Tak jako tak, samotný vývoj koncového výtvaru bude směřován k možnosti dalšího rozšíření a bude se tedy jednat o naprostý základ, na kterém lze ovšem vystavět mnohem komplexnější stroje schopné čehokoli.

2 CÍL PRÁCE A METODIKA

Hlavním cílem práce je vytvoření ovládacího softwaru pro platformu Lego Mindstorms 2.0, která umožní řídit robota sestaveného na této platformě. Aplikace poběží na počítači, kdy bude využívat k předávání dat rozhraní Bluetooth. V konstrukci robota samotného budou zakomponovány senzory, jež mu zabrání v narážení do překážek nacházejících se v prostředí, ve kterém se bude pohybovat.

K vytvoření návrhu jsou definovány dílčí cíle:

- 1 Prvním krokem je studium odborné literatury, dále nashromáždění dat z dostupných zdrojů a literatury týkající se programování, robotiky, navigace v prostoru a analýzy obrazu.
- 2 Na základě získaných znalostí a zpracování rešerše bude vybráno vhodné řešení pro sestavení samotného robota a bude vypracován návrh. skladba aplikace pro jeho ovládání.
- 3 Řešení bude současně implementováno na robota Lego Mindstorms 2.0 a funkčnost kódu bude testována v laboratorních podmínkách.
- 4 V průběhu práce bude průběžně řešení zhodnocováno a budou doporučeny další kroky pro optimalizaci kódu a stavby robota.
- 5 Vše bude následně shrnuto v závěru práce, kde budou také nabídnuta případná rozšíření.

3 PŘEHLED ŘEŠENÉ PROBLEMATIKY

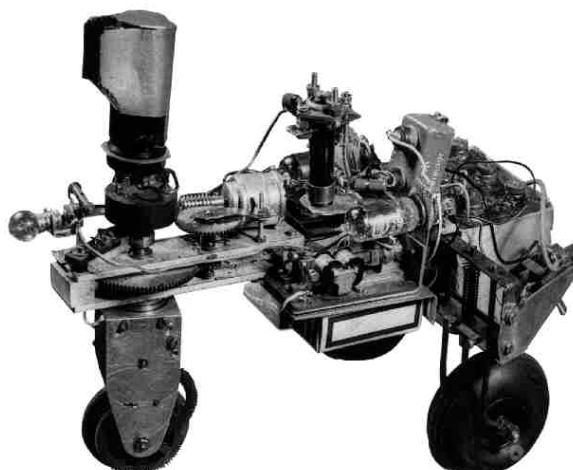
Samotné slovo „Robot“ označuje autonomního pracovníka poslouchajícího lidské příkazy a plnícího jejich přání. Alespoň tak to bylo zamýšleno v románu Karla Čapka R.U.R, kde se tak označovali zjednodušení lidé, kteří fungovali tak, jak je dnes chápán „robot“ v plném významu. Samotné slovo „Robot“ bylo poprvé použito v tomto románu a vymyslel jej Karlův bratr Josef Čapek ze slovanského slova „robota“ označující práci (Zunt, 2007).

3.1 Historie robotiky

Samotná myšlenka ohledně robota je datována již do staré Číny, kdy se vynálezci snažili vytvořit samostatně fungující stroje, často vypadající jako zvířata nebo jako lidé (Needham, 1991). S dálkově ovládanými stroji se nicméně začalo pracovat ve větší míře až ke konci 19. století, kdy jako dálkově ovládané vozidlo bylo prezentováno torpédo roku 1870 Johnem Ericssonem, Johnem Louisem Layem a Victorem von Schelihou (Gray, 2004). Bylo spojeno s mateřskou lodí dlouhým drátem předávajícím elektrické signály kormidlu torpéda a umožňující jej tak dálkově navádět na cíl.

Mezi první humanoidní výtvořky, stroje podobné člověku, lze zařadit robota Gakutensoku, jakožto prvního japonského robota, jehož tvůrce byl biolog Makoto Nishimura. Stroj byl poháněn tlakem vzduchu, mohl otáčet hlavou a pohybovat rukama (Durán, 2012).

Mezi roky 1948 a 1949 sestrojil William Grey Walter dva roboty nazvané „Elmer“ a „Elsie“, na kterých chtěl dokázat, že napodobováním zapojení lidského mozku dokážeme přenést složité struktury chování na roboty. Podařilo se mu vytvořit autonomní jednotky, které byly schopné nalézt cestu na dobíjecí stanici, pokud byly jejich baterie vybité (Hayward, 2001).



Obrázek 1 Jeden z robotů Williama Greya Waltera.

Zdroj: http://www.theoldrobots.com/images48/Elmer_Elsie2.JPG

Prvním digitálně programovatelným robotem určeným do industriální zóny byl Unimate vytvořen Georgem Devolem roku 1954. Roku 1960 byl prodán jeho první kus do General Motors, aby poskládal horké kusy železa a pomáhal tak s výrobou. Nalezením technického využití robotů byl dán vznik moderní robotice. Tak vznikl důvod finančně investovat do tohoto průmyslu a ten se tak začal velmi rychle rozvíjet (Waurzyniak, 2006).

Roku 1963 byl představen první balicí stroj společností Fuji Yusoki Kogyo Company (Fuji Yusoki, 2005). Po mnoha dalších výtvorech bylo roku 1976 představeno první plně programovatelné univerzální rameno Victorem Scheinmanem (Roth a spol., 1974).

Nakonec v roce 2000 japonská společnost *Honda* představila světu 130 cm vysokého humanoidního robota pojmenovaného *Asimo* (Advanced Step in Innovative MObility) (Kornblum, 2000). Ten byl navržen jako osobní asistent pro lidi postrádající plnou pohyblivost. Asimovy schopnosti podpořily tehdejší svět vědy a matematiky a napomohly tak urychlení vývoje robotiky (Obringer a spol., 2011).

3.2 Současnost robotiky

Dnes se vývoj robotů posunul dále a ti neslouží pouze v industriální sféře, jako manipulátoři, svářeči a lakýrníci. I když tím prakticky ovládli své odvětví a jsou v některých sférách již nenahraditelní. Například tisknutí plošných obvodů není již kvůli požadované přesnosti a složitost výroby v lidských silách a na řadu tedy přichází stroje (Mølhav, 2006).

V dnešní době zastávají dokonce práci člověka, která by byla pro něj značně repetitivní, složitá, nebezpečná nebo přímo smrtelná.

3.2.1 Těžářství

Firma Komatsu již nějakou dobu staví stroje, které usnadňují práci v těžbařském průmyslu. Jménem společnosti se pyšní obrovské nákladní stroje, které se plně samostatně „prohánějí“ po místě výkopu (Rio Tinto, 2011). Navíc jako první odhalila firma středně velký „inteligentní“ buldozer, jenž tak ještě více automatizuje proces těžení (Komatsu, 2013).

3.2.2 Doprava

Do automobilového průmyslu je robotika a automatizace zapojena asi z největší části, vždyť také jako první vlastnilo výrobní linku s robotem ve výrobním procesu General Motors (Pearce, 2011). Dnes se jedná o jedno z nejvíce automatizovaných odvětví a i když roboti pomáhají stavět vozidla, je tendence vytvářet ze samotných aut plně samostatné jednotky, které jsou označovány jako *Automated guided vehicles* (AGVs) (Muller, 1999).

Na 2015 Consumer Electronics Show (CES) v Las Vegas předvedla společnost BMW plně nezávislého asistenta v elektrovozidle řady I3. Jejich vozidlo samostatně, bez přítomnosti řidiče, zaparkovalo. V praxi by mělo auto samo nalézt parkovací místo a po jeho přivolání přes chytré hodinky by se mělo k vlastníkovi samo dostavit (Pavlůsek, 2014).

Dalším předpokládaným krokem bude úplná samostatnost vozidel. Bez pomoci lidí se budou navigovat v každodenním provozu, vyhýbat se pevným překážkám, sobě navzájem i chodcům (Muller, 1999). Již roku 2005 vyhrálo vozidlo *Stanley* závod *2005 DARPA Grand Challenge*, sestavené týmem *The Stanford Racing Team*, vedený profesorem Sebastianem Thrunem (Svoboda, 2005). Vyhrálo bez pomoci svých konstruktérů a zahájilo tak počátek závodu o plně samostatné auto, kterého se účastní konstruktéři z univerzit celého světa bok po boku s týmy společností Nissan, Renault, Toyota atd. (Russell, 2006). Neopomenutelným je také projekt *Google Self-Driving Car* firmy Google, vedený Sebastianem Thrunem, který stál za úspěchem vozidla *Stanley* v roce 2005 (Thrun, 2010).

3.2.3 Průzkumnictví

Existují místa, na která se lidé nemůžou zatím podívat osobně. Jedním z nich je například mořské dno a propasti, které jsou častým cílem vědců a průzkumníků. Díky jejich iniciativě a zvědavosti pokořilo podvodní vozidlo *Nereus*, sloužící od roku 2009, hloubku 10 000 m (Harlow, 2009).

Jsou ale i místa, co se dají považovat za extrémnější, jako například sopečný vulkán Erebus, který prozkoumával robot *Dante* vytvořený v Carnegie Mellon University (Wettergreen et al., 1993).

Neopomenutelný je ovšem také průzkum vesmíru, kam se lidské pokolení vydalo osobně jen velmi zlehka. Avšak již od roku 1957, kdy Svaz sovětských socialistických republik vypustil první vesmírnou družici *Sputnik 1* (Příbyl, 1997), lidstvu napomáhají automatizované stroje. Roku 1959 to byla *Luna 2*, jež dopadla jako první sonda na měsíc (Toufar, 1976). Prvním vozítkem ovládaným na dálku ze Země a pohybujícím se po cizím vesmírném tělese byl sovětský *Lunochod*. Ten od roku 1970 cestoval po povrchu Měsíce a stal se něčím naprosto nevídaným na poli kosmické robotiky (Rükl, 1991). Prvním vozítkem ovládaným na dálku a jezdícím po povrchu Marsu, pak byl americký rover *Sojourner*, jež úspěšně přistál v roce 1997 (Laubach, 1999). Jedním z posledních velkých počínů lidstva ve vesmírném průzkumnictví je přistání sondy *Rosetta* na kometě, které se podařilo dosednout na její povrch na konci roku 2014 po 10 letech cesty vesmírem (Esa, 2014).

3.2.4 Příroda

Již odnepaměti se snaží přírodovědci pozorovat zvířata ve svém přirozeném prostředí. Od vynálezu kamery mají pozorování značně ulehčené, ačkoli i nadále se musí uchylovat k různým způsobům maskování a ukrývání samotných fotoaparátů a kamer. Díky této potřebě vznikají robotické kamery, které představil světu inovativní fotograf přírody John Downer. Kamery, přestrojené za kousek ledu na vodě nebo za hromádku sněhu, dokázaly samy vyhledat polární medvědy, nenápadně se k nim přiblížit a následně je sledovat. Byly bohužel rozbity samotnými medvědy, což ovšem nebylo vcelku na škodu, jelikož přinesly světu velmi ceněné záběry (Resneck, 2011).

3.2.5 Záchranářství

Ninja robot není nejtýpčtějším příslušníkem této kategorie, ale v konečném důsledku pomáhá chránit lidské životy. Robot totiž šplhá za pomoci přísavky po výškových budovách, aby na nich provedl kontrolu. Lidé, za které robot zaskakuje, byli vystavováni silným poryvům větrů ve velkých výškách a tudíž operovali ve vysoce nebezpečném a proměnlivém prostředí. Robot tak odbourává riziko pro lidské pracovníky (Luk et al., 2007).

Jelikož mají hasiči jedno z nejnebezpečnějších zaměstnání, existuje silná iniciativa jim práci ulehčit. Americké námořnictvo představilo svého humanoidního robota *Saffir*, jenž je určen k hašení požárů lodí, které patří k jedněm z nejhorších (White, 2015). Britští hasiči ovšem již tuto výpomoc mají v podobě čtyřkolého radiem ovládaného robota s ramenem (Palmer, 2009).

I policisté najdou pomoc v robotickém průmyslu. *SuperDroid HD2 SWAT* je určen k zneškodňování bomb a jiných nebezpečných zařízení. Dálkově řízený pásový podvozek s dlouhým ramenem se stal nedílnou součástí policejních jednotek, které bomby likvidují (Abry, 2010).

3.2.6 Vojenství

Armáda patří k těm, které nejvíce dotují robotický průmysl. Také se jedná o nejnebezpečnější sektor vůbec a jak jinak zachraňovat lidské životy, než je rovnou neohrožovat. Příkladem několika takových robotů nahrazujících lidské spolubojovníky může být *MDARS* společnosti *General Dynamics*, s jehož vývojem se započalo již v roce 1990. Jedná se o automatizovaný vůz s velkým množstvím čidel, kamer a dokonce s radarem, který je využíván jako hlídač. Autonomně se pohybuje po určeném perimetru a pokud narazí na vetřelce, informuje operátora, který má možnost dát pokyn k palbě (Mahal, 2013).

Velmi známým zástupcem leteckých „dronů“ je *General Atomics MQ-1 Predator* společnosti *General Atomics*. Předveden byl poprvé roku 1995 a od té doby je stále ve službě. Letoun je dálkově řízen z ústředí a je využíván jako průzkumné letadlo, ale i jako útočný letoun – dokáže nést několik raket *Hellfire* (Whittle, 2013).

Armáda také využívá plně ozbrojené dálkově řízené roboty na pásových podvozcích. K ruce jim ovšem přijde i takzvaný *BigDog*, čtyřnohý robot, který přijme příkaz od operátora, ovšem pak jej plní naprosto samostatně – následuje vojáka, dojde cílového bodu atd. Je určen k přenášení vybavení při přesunech na delší vzdálenosti členitým terénem (Boston Dynamics, 2008).

3.2.7 Komerční sektor

I volná veřejnost si přijde na své. V posledních letech, se stále rychlejším vývojem mobilních zařízení, vznikají firmy, jenž nabízejí volně prodejné roboty a drony.

Takováto zařízení se po propojení s robotem stávají jejich „mozkem“ a uživatel je s jejich pomocí schopen stroje řídit – přijímat obraz z kamer, navigovat je a někdy dokonce i programovat. Příkladem jednoho takového je „kulička“ *Sphero*, toto malé zařízení ve tvaru koule je plně ovladatelné přes mobilní zařízení, reaguje na svého majitele a je prodáváno jako hračka (Sphero, 2015). Na druhou stranu se objevují i rádiem řízené čtyřmotorové vrtulníky (quadrocopter), které mají nejen možnost volně létat a natáčet krajinu (Ungerleider, 2013), ale jako v případě velké obchodní společnosti *Amazon*, doručovat samostatné zásilky až k zákaznickým dveřím (Barilar et al., 2014).

Objevují se i roboti vysavače, které zcela samovolně jezdí po podlaze a vysávají nečistoty. *Roomba* série takovýchto strojů, představených v roce 2002, uklízí již na začátku roku 2014 v 10 mil. domácností. Robot se samovolně pohybuje po daném povrchu, kdy se vyhýbá překážkám i schodům a vrací se sám do nabíjecí jednotky (iRobot, 2015).

3.2.8 Výzkum

Roboti se ovšem dají využít i k výzkumu již existujících věcí, ten může svým naprogramováním simulovat vzorec chování nebo dokonce celého reálného tvora. Tímto způsobem vznikla *Anna Konda* - had hasič. Stroj se pohybuje jako had, jen místo hlavy má hasící zařízení (Liljebäck, 2012). Na podobném principu vznikla i *Robo ryba*, jenž autonomně brázdí přístavy v severním Španělsku a zjišťuje jejich znečištění (Boyle, 2012).

Projekt z roku 2010 s názvem *MorpHex* zkouší úplně nový způsob pohybu, než zvířata napodobující „bratrance“. Robot se totiž pohybuje po šesti nohách, ale k rychlejšímu přemísťování se zabalí do koule a valí se ke svému cíli (Halvorsen, 2010).

Velmi zásadním typem robotů pro aktuální výzkum jsou multirobotické systémy. Neexistuje jen jeden jedinec s vysokou inteligencí, ale velké množství malých s nízkou, ale propojenou inteligencí. Tito jsou řízeni jedním superpočítačem, nebo jsou propojeni do takzvané *hive mind*, jakou oplývají např. mravenci. A i když jedinec nezmůže téměř nic, celek dokáže plnit i komplikované úkoly (Arvin a spol., 2014).

3.3 Stavba fyzické části robota

Pokud navrhuje robota korporátní společnost titánských rozměrů, může si dovolit navrhnout jej úplně od nejmenšího atomu. Vyrobí si na míru procesor, jenž bude „mozkem“ celého výtvaru, vytiskne si základní desku se všemi potřebnými obvody a navrhne jednotlivé servo-motory přesně podle jejich typu určení. Takto například vyvíjí některé své prototypy americký výrobce zbraní *Global Dynamics* (Song a spol., 2010).

Ve chvíli, kdy se ale člověk rozhodne sestavit robota vlastnoručně, narazí na několik problémů. Musí vyřešit, jak jej navrhne, z čeho jej sestaví a čím jej bude instruovat. V každé z těchto sekcí má tvůrce ovšem nepřehledné množství možností výběru.

3.3.1 Typy strojů

Typů robotů k sestavení existuje nepřehledné množství, limitované jen lidskou představivostí, jak dokazuje návrh modulárního robota projektu *MOSS*. Jedná se totiž o malé kostičky obsahující všemožnou elektroniku a jejich spojováním do většího celku může vzniknout dostatečně komplexní výtvar (Modrobotics, 2012).

Ze zásady se roboti dají dělit podle míry autonomie. Za prvé se může jednat o *autonomní* jedince. Ti nepotřebují být vedeni člověkem a vzniklé situace řeší zcela sami. A za druhé *dálkově řízení* roboti – nepodniknou žádnou akci samovolně a vyžadují instrukce od člověka (Ilstu, 2014).

Rozdělení je možné i podle formy pohybu. *Stacionární* jsou často ve formě manipulátoru podobného industriálním robotům. *Pohybující se po zemi* a to, ať už plazící se jako *Dagu Caterpillar* (Dagurobot, 2015), valící jako *Sphero* (Sphero, 2015), jezdící jako rover *Curiosity* (Nasa, 2015) nebo chodící jako *Asimo* (Obringer, 2011). *Plovoucí ve vodě* na hladině jako *Robotic lake lander* (Coxworth, 2012) nebo pod hladinou jako *Amphibot III*. Mohou se ovšem pohybovat i *ve vzduchu* jako *IRIS+*.

Dělení robotů je komplikované a tudíž vznikly i specializované typy, které přesně vymezují funkčnost a podobu strojů v nich obsažených. Existují *Automatic guided vehicle* (AGV), samostatně se pohybující vozidla bez pomoci člověka, *výuční roboti* jako *LEGO Mindstorms* série; *modulární roboti* - seskládání z menších velmi podobných kusů, které ovšem dokáží dohromady spolupracovat a fungovat jako celek (Yim a spol., 2002); *teleroboti*, fungující jako multifunkční ramena ovládaná na dálku (Sheridan, 1992); *disability robot*, neboli robot pomáhající postiženým nebo nemohoucím lidem (Jaeger, 2006); *nanoboti*, mikroskopičtí roboti, pracující ovšem jako celek; *přeskladatelní roboti*, nemající danou podobu a seskládají se ze samostatných částí, které dokáží přemisťovat dle potřeby (Haberzettel, 2002); *soft robots*, bez pevné kostry a sestávající pouze z měkké hmoty, která se ale dokáže dle libosti ohýbat, jak předvedli vědci z Harvardské univerzity (Shepherd a spol., 2011); *swarm bots*, inspirováni koloniemi mravenců, termitů a včel, kde každý jedinec není nijak komplikovaný a inteligentní, ale jako celek dokáže spolupracovat na řešení problémů (Dudek, 1993). Posledními dvěma typy jsou tzv. *dron*, neboli Unmanned Aerial Vehicle (UAV), čili bezpilotní letoun (Krajník a spol., 2011), a samozřejmě *android*, neboli *humanoid*, autonomní robot velmi blízce kopírující člověka, ať už vzhledem nebo chováním (Minato et al., 2004).

3.3.2 Komponenty

Obstarat si díly na stavbu robota není ve skutečnosti až takový problém. Existuje hned několik společností, jenž se na tento obor specializují a je možné si díky nim pořídit klíčové části.

Robot téměř vždy sestává z několika základních komponent. Z *procesoru*, jenž řídí veškeré funkce stroje a z *motorů*, které pohybují ať už samotným robotem, nebo jen nějakou jeho částí kostry, na které jsou jednotlivé komponenty připevněny a tvoří tak tělo celého výtvaru.

Samotná kostra může být vytvořena z čehokoli, od papíru, přes umělou hmotu, až po různé kompozity uhlíku. Je tedy jen na tvůrci, zda využije svých schopností a sestrojí vlastní, nebo zakoupí nějakou již připravenou.

Mikroprocesory jsou něco úplně jiného. Aby vznikly, je potřeba strojové přesnosti a dokonale sterilních podmínek. Něčeho takového není běžný člověk jen tak schopen

dosáhnout. Proto vznikly firmy nabízející tyto zařízení spolu se základními deskami s kompatibilními čipsety (Letsmakerobots, 2008).

Společnost *Atmel* nabízí hned několik produktových sad programovatelných procesorů, které jsou určeny pro automobilový průmysl, ale i do mobilních zařízení díky své nízké spotřebě. Atmel se omezuje jen na procesory a paměti a nenabízí žádné další komponenty, které by byly použitelné pro stavbu robotů. Nicméně jsou poměrně hojně využívány, díky své rozmanitosti v nabízených variantách (Atmel, 2015).

Oproti společnosti Atmel je na trhu i firma *Digilent*, která nenabízí pouze mikroprocesory, ale také zařízení použitelná pro výzkumné účely, měřidla a čidla. Ve svém sortimentu má dokonce i příslušenství pro roboty, od kostry, kol a krytů, až po jednotlivé servo-motory. Pro začátečníky navíc nabízejí základní balení, neboli kity (Digilent, 2015).

Variantou je *Arduino*, společnost zaměřená na řešení pro robotické nadšence a tvůrce. Nabízejí hotové sestavy základních desek a procesorů použitelných jako „mozek“ robota (Arduino, 2015). Podobně jakou Arduino existuje i firma *Axon*, která využívá mikroprocesorů společnosti Atmel. Jedná se o hotové řešení řídicí jednotky s předinstalovaným firmwarem určeným ke zjednodušení práce s jednotkou (Axon, 2015).

Nabídka firmy *Bdmicro* je určena pro širší veřejnost. V portfoliu jsou totiž již hotové řešení řídicích modulů použitelných pro všechny možné zařízení, které se na nich rozhodne člověk sestavit. Základní deska přímo obsahuje procesor, paměti a porty pro vstupní a výstupní zařízení (BDMicro, 2015).

Velkou nabídku všech možných součástí upotřebitelných při stavbě jakéhokoli typu robota má společnost *Active Robots*. Od baterií, přes servo-motory, až po řídicí moduly. V nabídce má i velké množství vstupních a výstupních zařízení. Nabízí ovšem i kompletní sestavy robotů, které jsou určeny pro začínající stavitele (Active-robots, 2015).

Specializující se na kity, nebo-li celé sestavy, je firma *Gumstix*. Jejich balíčky ovšem neobsahují pouze hotové roboty, ale i řešení pro síťové prvky nebo mobilní zařízení (Gumstix, 2015).

Všechny předešlé varianty vyžadují značnou míru dovedností v elektrotechnice a programování. Firma *Lego* nabízí jednodušší řešení určené především pro děti jako výuční pomůcku a hračku. První se objevila na trhu varianta *RCX*, následovala *NXT* a zatím poslední je *EV3*, kdy každá existovala v několika verzích. Nejen, že s postupnými verzemi rostl výkon řídicích „kostek,“ ale rozrůstala se i množina senzorů, která k nim byla dodávána. U *NXT* to byl světelný senzor, zvukový, tlakový a ultrazvukový, k tomu ještě 3 motory. *EV3* ovšem měla ve své výbavě ještě barevný senzor, infračervené dálkové ovládání a výkonnější motory. Veškeré programování se neodehrávalo v typickém programovacím jazyce, ale ve speciálním grafickém rozhraní, kde se skládaly kostičky, jakožto funkce, za sebe, aby vytvořily ve výsledku komplexní program. Nekončilo to ovšem pouze u hraček a jednoduchých a limitovaných řešení. Řídicí kostičky nebyly nijak chráněny a dokonce podporovaly přehrání firmwaru vlastním, jak dokazuje samotná dokumentace k zařízení. Nyní existuje mnoho variant operačních systémů a knihoven pro tyto „hračky“ (Lego, 2014).

Lego ovšem není jediná společnost s touto formou „stavebnic“. *VEX Robotics* čerpá ze stejného nápadu a také nabízí své řešení, jako programovatelnou řídicí jednotku s množstvím různých vstupních a výstupních zařízení. Jenž je za pomoci dalších konstrukčních prvků možné uspořádat do neuvěřitelných výtvorů (Vexrobotics, 2015).

3.4 Tvorba řídicího softwaru

Jakmile je první verze robota sestavena, přichází programátorská část. Chování základní desky a závisle skrz ní i jejích příslušenství, je u všech druhů mikroprocesorů, možné definovat *jazykem symbolických adres* (označovaným *Assembler*). Toto není v mnoha případech žádoucí, jelikož by tato práce byla nesmírně komplikovaná a vznikla náhrada jménem *C*.

Některé společnosti nabízející různá řešení pro vynálezce a robotické nadšence ovšem nezůstaly jen u velmi rozšířeného jazyka *C* a programy určené pro jimi vyprodukované čipy je možné tvořit i za pomoci jiných programovacích jazyků nebo dokonce graficky v nabízeném softwaru.

3.4.1 Programování

Společnost Atmel nabízí kompletní řešení pro vývoj aplikací ke svým výrobkům. *Atmel Studio* nejen dokáže zkompileovat stávající programy v C, C++ a Assembler, ale umožňuje také jejich tvorbu ve svém unikátním vývojovém prostředí, které samozřejmě obsahuje veškeré knihovny a nástroje potřebné ke správě a řízení procesoru a zařízení této firmy. Výsledná řešení rovnou odesílá do paměti konstruktů, jenž využívá výrobků společnosti a uživatel je může rovnou vyzkoušet (Atmel, 2015) (Mann, 2013).

BDMicro využívá čipy společnosti Atmel, proto její řešení lze využít i tomto případě. K použití ovšem nabízí i další kompilátory, jenž dokážou přeložit do strojům srozumitelného jazyka i BASIC, FORTH a Pascal (BDMicro, 2015).

Podobně jako Atmel i čipy Arduino, mají své IDE vývojové prostředí. Se stejným názvem jako společnost, tedy *Arduino*, dokáže kompilovat C, C++ a zasílat čipům instrukce. K tomu nabízí vývojové prostředí pro uživatele, kde naleznou i veškeré knihovny pro své aplikace.

Někteří výrobci nabízejí i hotová řešení operačních systémů pro své zařízení. Jedním z nich je právě Arduino, jehož vlastní verze operačního systému Linux se jménem *Yún* je volně k dostání (Arduino, 2015).

Nezaostává ani společnost Digilent, jenž má vlastní řešení pro podporu stávajících verzí operačního systému Linux. Za pomoci *Digilent Embedded Linux* je možné spolu s zařízeními jmenované firmy využívat plně podporované všechny verze Linuxu. Rozšířené o Linaro Ubuntu, speciální verzi operačního systému firmy Linaro. Jak už název napovídá, je odvozen z verze Linuxu Ubuntu. Jeho přední vlastností je speciální úprava, aby vyžadoval co nejméně výpočetních prostředků. Rozšíření se ještě vztahuje na jednu odnož Linuxu a to BusyBox, velmi skromnou adaptaci operačního systému, kdy obsahuje pouze jádro plného Unixu a několik součástí navíc, aby alespoň z části plnil funkčnost kompletního systému. Výsledkem je operační systém skromný nutné místo v paměti, ale i na výpočetní prostředky. Tyto rozšíření umožňují produktům Digilentu se angažovat v mobilních zařízeních všeho druhu (Digilent, 2013).

Firma Gumstix má plnou podporu různých i méně známých operačních systémů, jako Yocto project nebo Ångström, což jsou opensource projekty odvozené z Linuxu a určeny pro menší zařízení. Vedle nich nezaostává ani u kompatibility se systémy jako Ubuntu a Android. Ne všechny jejich produktové řady ovšem podporují veškeré podporované operační systémy (Gumstix, 2015).

Tabulka 1 Podpora operačních systémů produkčními řadami společnosti Gumstix.

	Overo	DuoVero	Pepper	Verdex pro
Yocto project	✓	✓	✓	
Ubuntu	✓ (bez STORM-P)			
Android			✓	
Ångström	✓			✓

Zdroj: <https://www.gumstix.com/software/supported-software/>

Tvorba aplikací v robotice nezůstala pouze u klasických programovacích jazyků a technik. Rozrostla se i do grafické sféry, přesněji do grafického programování. Uživatel nemusí znát žádnou syntaxi programovacího jazyka, nebo důvěrně znát nějaké hlubší zákonitosti. Postačuje se vyznat v uživatelské rozhraní softwaru umožňujícího vytvářet aplikace právě touto metodou. Jazyk ovšem není limitován množstvím již hotových (předprogramovaných) funkcí, nýbrž si může dle libosti přidávat - již v programovacím jazyce, kterém je prostředí vyhotoveno. Jednou z těchto aplikací je oficiálně dodávaný software se stavebnicí Lego Mindstorms, *ROBOLAB*. Ten je zjednodušen natolik, aby jej bylo schopno obsluhovat i dítě starší deseti let, jak uvádí výrobce (Lego, 2014) (Židek, 2002).

Varianta tohoto programu zprostředkovávající grafické programování je *ROBOTC*. Univerzální programovací jazyk založený na C. Neomezuje se ovšem pouze na roboty firmy Lego, ale pokouší se obsáhnout co největší množinu podobných robotických variant. Pomocí něj se může vytvářet aplikace pro produkty společnosti VEX, Lego, Tetrix, RCX a

některé varianty Arduino. Umožňuje také vytvářet virtuální prostředí, jenž pomáhají ladit programovaný software ještě dříve, než je nahrán do robota (Robo Matter, 2015).

3.4.2 Lego Mindstorms

Produktová řada Mindstorms společnosti Lego začala typem *RCX*. „Programovatelná kostka,“ jak je označován řídicí modul, představovala již hotové hardwarové řešení pro stavitele robotů. Umožňovala zpracovávání jimi vytvořených aplikací zcela bez omezení a vysloužila si tak mezi nimi značnou oblibu. To také vedlo k dalšímu vývoji a zavedení dvou následujících generací – *NXT* a *EV3*. Verze *NXT* se pyšnila zvýšeným výkonem, přidáním víceřádkového displeje, práce se zvukem a mimo jiné rozšířením sortimentu, se kterým se dala „kostka“ přímo propojit (Lego, 2014). *EV3*, vydaná v roce 2013, pak rozšiřovala svoji funkčnost na „chytrá“ mobilní zařízení a přidala několik dalších senzorů a dálkové ovládání, jež byly přímo obsahem balení (Lego, 2014).

K Lego Mindstorms je přikládán i vlastní programovací software, jenž dává možnost vytvářet aplikace pro roboty grafickým modulárním způsobem. Výborný způsob pro začátečníky a děti, na které je také cílena politika firmy Lego. Robotickým nadšencům toto ovšem nedostačuje, protože limituje počet funkcí, se kterými mohou pracovat a nemožnost konstruovat své aplikace i na nestandardní příslušenství, jako webkamery a jiné (Power, 2014). Proto také vzniká velké množství softwarových řešení, jenž lze nahrát přímo do řídicího modulu, nebo pomocí něj zkompilovat vlastní práci. A vzniká tak možnost nahrávat vlastní programy v jiných programovacích jazycích.

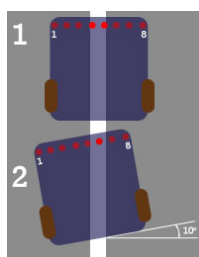
Silně podobný oficiálnímu ROBOLABu je *ROBOTC* (viz. výše), který ovšem rozšiřuje použitelnost o množství funkcí. Mezi populární řešení patří ještě *GNAT GPL* umožňující tvoření aplikací pro Mindstorms v objektově orientovaném jazyce Ada (Adacore, 2014). Rozšířený jazyk C má v své zastoupení v *Not eXactly C* (*NXC*), což je odnož tohoto jazyka – syntaxí připomíná C, ale v jistých ohledech se liší (Hansen, 2007). Využít lze i Visual Basic, kdy aplikace vytvořené v COM+, mohou být nahrány přímo do řídicího modulu. *PbForth* je název firmwaru, jímž lze nahradit ten stávající. Robot pak dokáže zpracovávat software vytvořený v programovacím jazyce Forth (*PbForth*, 2013). Velmi populárním se stalo řešení *LeJos*, operační systém "nahrátelný" přímo do „řídicí kostky“ Lego

Mindstorms, přinášející mnoho knihoven pro správu zařízení dodávaných v balení – motory, senzory (LeJos, 2009).

Postupně se objevuje stále více řešení pro tvorbu vlastních aplikací v různých jazycích, jako *Mindstorms SDK*, *Robotics.NXT*, *Simulink* a další.

3.4.3 Navigace pomocí cesty

V kontextu s robotikou se často řeší problém orientace robota v reálném prostředí. Jedním z řešení je vymezení cesty, po které se může pohybovat a nesmí z ní nijak vybočovat. Díky tomuto řešení není potřeba programovat do myslících cyklů nějaké složité analyzační procedury okolí. Robot musí nanejvýš reagovat na překážky v cestě, ale nemusí je nijak zvlášť zkoumat, aby se jim vyhnul. Toho lze docílit pouhým sledováním vzdáleností od okolních překážek.



Obrázek 2 Reakce čidel na polohu na čáře.

Zdroj: http://ikalogic.cluster006.ovh.net/wp-content/uploads/cell_spacing.jpg

Tato cesta bývá v reálném světě fyzicky přítomna. Je možné ji zadat do algoritmu počítače řídicího robota virtuálně, ale můžou pak vznikat jisté problémy, které se následně špatně ladí. Nejen, že vývoj takového řídicího softwaru je mnohem složitější a tudíž dražší, ale variabilita je na tom ještě poněkud hůř. Aby se vůbec mohla taková dráha vytvořit, muselo by se důkladně zmapovat prostředí, ve kterém se bude pohybovat, navíc jej složitě převést do počítačové mluvy. Řešení by bylo náchylné na proměnlivost prostředí, pro představu, je jednodušší posunout „koleje“ ve fyzickém světě, než je přetvořit v počítači. Výhodou tohoto provedení je, že když je stroj vychýlen, ať už nedopatřením nebo v rámci nějakého navigačního prvku z kurzu, mnohem lépe nachází svou původní dráhu (Kamal, 2008).

Mnohonásobně jednodušší formou je pak následování fyzicky existující cesty vytyčené například černou čarou na zemi. Algoritmus ovládající robota je v kontrastu s virtuální

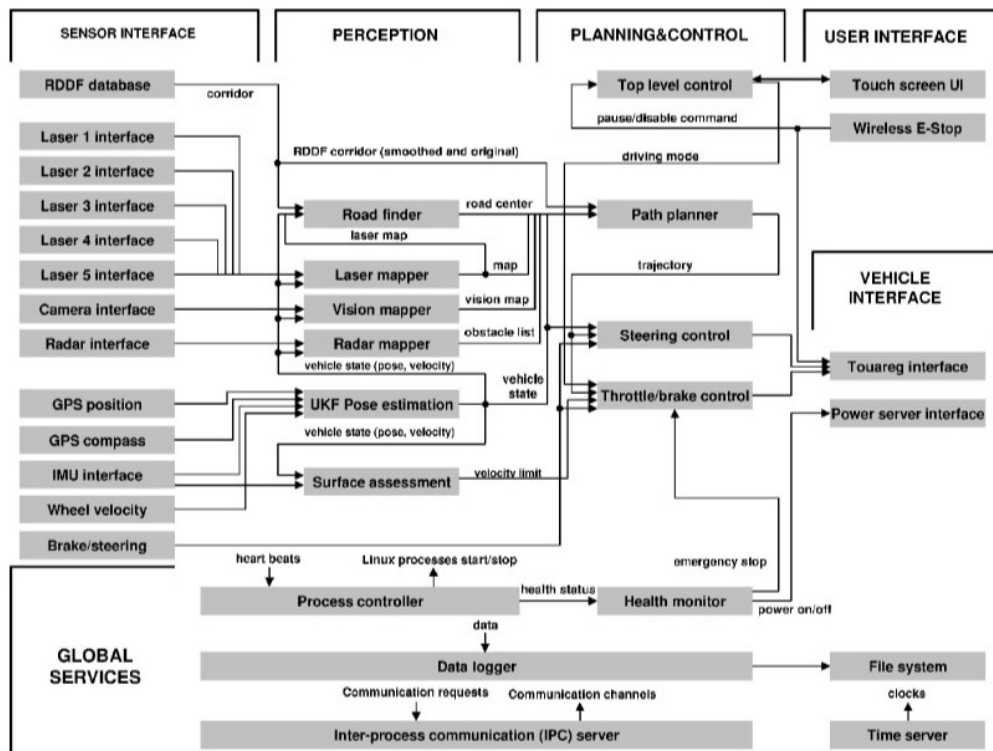
cestou značně zjednodušen. Kamera nebo jiná forma senzoru je natočena kolmo k zemi a přímo sleduje stopu na ni vyznačenou. Vychýlený stroj z kurzu ovšem jen těžko znovu nachází původní cestu. A navíc, sledování referenční stopy je náchylné na podmínky viditelnosti, pokud je použita optika. To se ovšem dá vyřešit nahrazením optické metody sledování viditelné čáry za například magnetickou. V podlaze komplexu, ve kterém se robot pohybuje, je nainstalován magnetický pásek, který lze sledovat i když není plně viditelný, čímž se zvyšuje durabilita navigace (Mathas, 2013).

Tento způsob orientace robotů je často využíván v industriální sféře, kde se mohou volně pohybovat např. po montážní hale a přemisťovat těžké materiály.

3.4.4 Navigace analýzou okolí

Definování dráhy předem není vždy žádoucí. Pokud je prostor proměnlivý, nebo je stroj určen do otevřeného prostředí, ve kterém se nacházejí překážky, pohybující se jiné stroje a lidé, pak je mnohem vhodnější určovat cestu v reálném čase, v daném okamžiku. Stavba vodící linky tedy nepřipadá v úvahu, robotovi se jen řekne kam se má dostat a on sám pak nalezne vlastní cestu k tomuto cíli. Aby měl tuto schopnost, musí se mu dodat oči a uši, které nalézáme u živočichů. On ovšem může oplývat i mnohem větší sestavou senzorů, pomocí nichž bude prozkoumávat fyzický svět. Termální a infračervenou kameru pro noční vidění, sonar pro hloubkové snímání, skener pro detailní analyzování atd. Výsledkem by měla být výborná orientace v prostoru, ale samotný hardware nemusí stačit, pokud mozek nevyužívá dostatečně komplexní algoritmus k vyhodnocování vstupů z těchto „smyslů“ (Rao a spol., 1990) a (Zhang a spol., 2013).

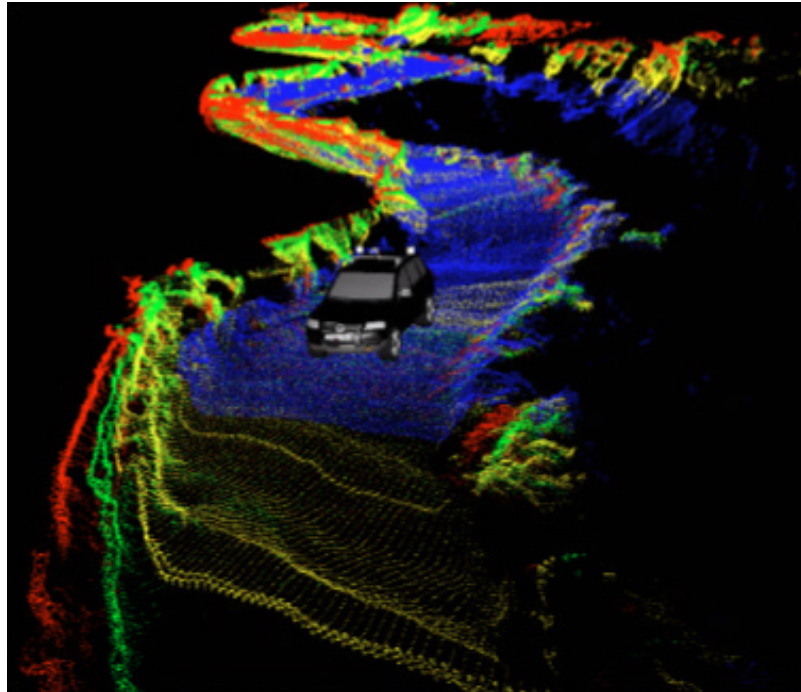
Autonomní vozidlo *Stanley*, výherce 2005 DARPA Grand Challenge (série závodů autonomních vozidle pořádaných společností Red Bull), ukazuje způsob komplexního „myšlení“ takového robota, jenž má zadaný pouze cíl a aby k němu dostal, musí si sám nalézt cestu. K „vidění“ využíval GPS navigaci k určení hrubé polohy, radar pro vzdálenější okolí, kameru k obrazové analýze, sadu laserů, jimiž prohledával své blízké okolí, gyroskop pro udržení náklonu a mnoho dalších senzorů. Výstupní data senzorů rozčleňoval do jednotlivých vrstev, které následně skládal na sebe a od nejzákladnější se rozhodoval jak prostředí okolo něj vypadá a kudy se nejlépe vydat.



Obrázek 3 Diagram analýzy a nalezení cesty.

Zdroj: <http://robots.stanford.edu/papers/thrun.stanley05.pdf>

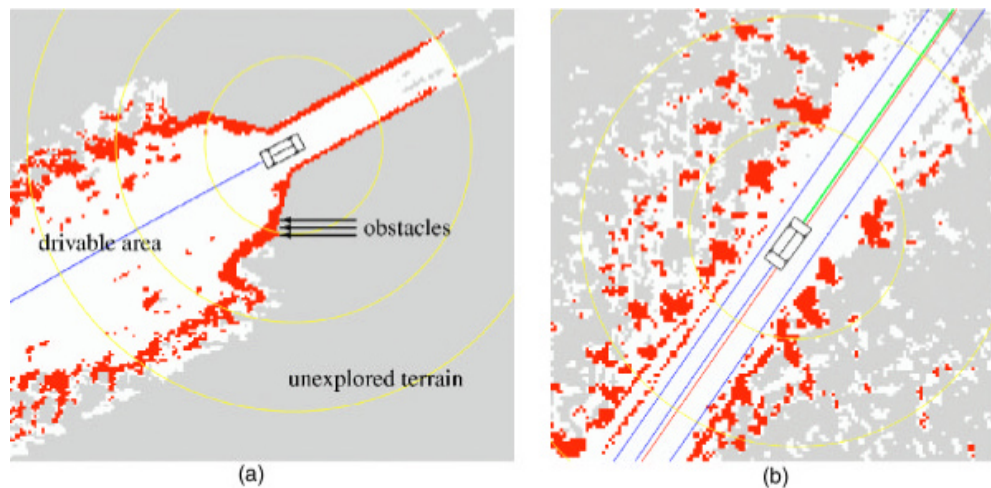
Díky takovému množství „očí“ mohl Stanley zkoumat své okolí velmi důsledně. Obzvláště laserové senzory se zasadily o jeho vítězství v tomto závodě. Dokázaly velmi přesně a pohotově získávat data popisující vzdálenosti, výšky a velikosti objektů a krajiny.



Obrázek 4 Laserová data zachycena průjezdem nejobtížnější pasáže trati.

Zdroj: <http://robots.stanford.edu/papers/thrun.stanley05.pdf>

V propojení s dalšími daty se vytvářely logické mapy, za jejichž pomoci se vozidlo rozhodovalo, jakou cestu zvolí. Ne vždy vše fungovalo bez problému, jelikož čas od času vznikali tzv. „duchové“ – záznamy neexistujících překážek – jimž se Stanley vyhýbal.



Obrázek 5 Ukázka rozhodovacích map vytvořených z dat všech senzorů Stanleyho.

Zdroj: <http://robots.stanford.edu/papers/thrun.stanley05.pdf>

Potřeba většího množství senzorických zařízení je zřejmá. Díky propojování dat z více typově různých zdrojů klesá pravděpodobnost tvorby chyb. Robot se může mnohem přesněji a sebevědoměji pohybovat v reálném světě.

Projekt tohoto autonomního vozidla ovšem nepatřil mezi ty nejlevnější, jak dokazuje cena systému, jenž by řešil problém s výskytem „duchů,“ odhadovaný na stovky tisíc dolarů. Systém právě kvůli své ceně nebyl do Stanleyho nainstalován (Thrun a spol., 2006) a (Kato a spol., 1996).

Samotný závod DARPA Grand Challenge se od té doby konal již několikrát a stále posouvá svými výzvami hranice robotiky dál (Urmson a spol., 2007).

4 VLASTNÍ PRÁCE

Existuje mnoho způsobů, jak sestavit vlastního robota, aby nejen vypadal dle vašich představ, ale i se tak choval. Velmi zručný inženýr si může k ruce vzít pájku a sestavit si jej úplně od píky. Jedná se ovšem o velmi náročný proces, jak časově, tak intelektuálně. Naštěstí mnoho firem vychází těmto lidem vstříc a nabízí hotové díly, jako motory, senzory a procesory. Existuje ale i mnoho stavebnic, pomocí nichž lze takových výtvorů dosáhnout, aniž by člověk musel spájet jediný obvod, nebo řešit do jaké „krabice“ to celé uzavře.

Tato práce se zabývá právě tou druhou variantou, kdy si člověk může zakoupit již hotové jisté aspekty projektu. Nemusí se tak s nimi trápit a může se soustředit na jiné. Sestaven bude robot poslouchající příkazy z počítače, sledující cestu a vidící objekty na ní.

4.1 Lego Mindstorms

Pro práci byl využit robot verze Lego Mindstorms NXT 2.0, druhá generace těchto zařízení. V prodeji se objevila roku 2009 s balením o 619 dílkách, obsahem s inteligentní kostkou, jak bývá označována programovatelná řídicí jednotka, softwarem pro nahrávání zvuků, jejich úpravu a opětovné nahrání do NXT. Dále s editorem, ve kterém lze upravovat obrázky a následně zobrazovat na displeji zařízení. A s programem, ve kterém se tvoří aplikace pro Lego Mindstorms, jednoduchým skládáním procesů za sebe, v grafickém rozhraní.



Obrázek 6 Balení Lego Mindstorms NXT 2.0.

Zdroj: Vlastní



Obrázek 7 Ukázka programu v editoru NXT-G.

Zdroj: <http://thetechnicgear.com/wp-content/uploads/2014/03/light-callibration.png>

Balení také obsahuje sérii několika senzorů a tří elektromotorů. Dva tlakové senzory, barevný, který dokáže detekovat šest barev (modrou, zelenou, červenou, žlutou, černou, bílou) a jejich světelnou intenzitu. Ultrazvukový senzor, pro měření vzdálenosti od objektů, zvukový senzor (mikrofon), který ovšem neumí nahrávat zvuky, ale jen rozpoznat jejich hlasitost.

4.1.1 Řídicí jednotka

Samotné NXT má na sobě několik programovatelných tlačítek, mikrofon, reproduktor a LCD displej s rozlišením 100x64 pixelů. Pro komunikaci s okolím využívá čtyři vstupní porty a tři výstupní, jeden USB pro propojení s počítačem a Bluetooth v2.0. Uvnitř se ukrývá 32-bitový Atmel AT91SAM7S256 procesor s 256 KB paměti flash a 64 KB RAM. Vše pak funguje na šesti AA bateriích.

Pokud se bude jevit programování v dodávaném softwaru některým programátorům poněkud limitující, není nutnost zůstat u něj. Dá se tedy využít mnoho jiných programovacích jazyků díky komunitě vývojářů, jenž vytvořila mnohé rozhraní umožňující jejich využití.

4.1.2 Stavba robota

Prvním důležitým krokem bylo určit podobu a typ robota, který se bude vznikat. Vybrán byl pozemní, pohybující se na pásovém podvozku, jelikož umožňuje mnohem jednodušší testování různých funkcí a jejich realizaci vůbec. Pásový podvozek navíc umožňuje variabilnější možnosti pohybu a i když existují i jiné možnosti, tento byl pro zamýšlený výsledek nejvýhodnější.

Vznikl podvozek, s jakým se lze setkat u tanků. Ne, že by stejně fungovalo zavěšení kol, rozložení a převodování výkonu na jednotlivé pásy atd., ale byla převzata funkčnost tankového podvozku. Aby se mohlo vozidlo, mimo obvyklé pohyby, otáčet na místě a jezdit po jakkoli složitém povrchu. Robot má dvě pásové nápravy poháněné dvěma nezávislými elektromotory. Třetí motor byl zasazen do konstrukce podvozku, jako příprava na budoucí rozšíření.



Obrázek 8 Podvozek vozítka.

Zdroj: Vlastní

Řídicí modul je ukotven nad tím vším tak, aby byl lehce odnímatelný a aby mohl sloužit pro případné rozšiřování konstrukce do výšky.



Obrázek 9 Uchycení řídicího modulu NXT.

Zdroj: Vlastní

4.2 Programovací jazyk

Pro projekt je vybrán programovací jazyk leJos NXJ založený na Javě. Svě rivaly porazil díky své propracovanosti. Obsahuje velké množství knihoven pro práci se samotným NXT modulem, ale i všemi jeho součástmi, včetně senzorů a motorů. V základu se jedná o firmware do *inteligentní kostky* NXT zpracovávající aplikace napsané v jazyce Java a knihovnu tříd pro práci se všemi náležitostmi Lego Mindstorms.

Výběr programovacího jazyku Java má ještě jednu výhodu. Díky propracovanosti knihovny leJos a jejím postavením na zmiňované Javě, umožňuje jednoduchý vývoj aplikací propojující funkčnost robota a mobilních zařízení, jako jsou „chytré“ telefony a tablety. Ty pak mohou sloužit například jako dálkové ovládání. Umožněno je ale i propojení robotů a *Java application serverů* ovládající najednou větší množství strojů obsahující NXT kostky.

4.2.1 Použití v NXT

V první řadě je nutné přehrát originální firmware v řídicím modulu leJosem. Jen tak je možné nahrávat programy vytvořené v Javě do NXT.

Zařízení se připojí přes USB kabel do počítače a za pomoci instalačního softwaru se nainstaluje leJos do počítače, ale i do zařízení. Do Mindstorms se jen nahraje leJos verze firmwaru a v počítači se rozbalí do „instalační“ složky knihovna s dokumentací, kompilér a ukázkové projekty, na kterých lze pochopit samotné používání jazyku. Nejedná se tedy o instalaci jako takovou a celý proces lze provést i ručně, bez nabízeného instalátoru, přes příkazový řádek.

Jakmile je toto hotové, grafické rozhraní na displeji NXT se kompletně změní. USB kabel již není esenciální a lze nahrávat své programy přes Bluetooth. Je vhodné vyzkoušet kompilaci a nahrání první aplikace pomocí bezdrátového připojení, jelikož samotný leJos využívá vlastní kompilátor a s robotem se bude komunikovat na dálku.

```

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
        Button.waitForAnyPress();
    }
}

```

Obrázek 10 Testovací program HelloWorld k otestování funkčnosti leJosu.

Zdroj: Vlastní

Ke spuštění programu je nutné zadat několik příkazů do příkazové řádky operačního systému. Napřed se aplikace zkompiluje příkazem `nxc <název souboru>.java` do souboru typu *class*. Dalším příkazem se přeloží do formátu *nxt*, kterému firmware leJos již rozumí, nahraje do NXT a rovnou i spustí (což provádí přívlástek `-r`). Tímto příkazem je `nxc -r <vzniklý soubor class>`.

Aplikaci není nutné takhle pokaždé nahrávat. Je možné ji znovu spustit přes grafické rozhraní přímo v řídicím modulu, kde zůstává uložena.

4.2.2 Použití s počítačem

Kompilovat a nahrávat projekty lze i přímo v IDE prostředí, ve kterém je psán kód programu. Pro Microsoft Visual Studio existuje samostatný script navržený tvůrci, který IDE nastaví tak, aby to bylo umožněno. NetBeans na druhou stranu tímto neoplyvají a musí se nastavit ručně.

Speciálně kompilovat je potřeba pro aplikace určené do NXT, část běžící například na počítači, který s ním pouze komunikuje, tato nutnost nevzniká. Avšak i tak existuje příkaz pro příkazový řádek na kompilaci `nxcpc <název souboru>.java` a spuštění `nxcpc <název vzniklého class>`.

4.3 Komunikace

Součástí projektu je řízení robota na dálku pomocí počítače. Samotná komunikace obou zařízení není nijak jednoduchá. Nejen, že si mezi sebou musí předávat pakety naplněné daty, kterým obě strany rozumí, musí to dělat i dost rychle na to, aby bylo ovládaní aspoň trochu použitelné. Reagovat by měli i na různé náhlé případy, jako výpadek spojení, ztráta paketu nebo zasláná špatná data.

Samotná vysoká rychlost spojení by se dala zajistit využitím USB kabelu k propojení obou zařízení. To by ovšem nemělo moc velký význam, jelikož by se tak ovlivnila pohyblivost a variabilita robota. Ke komunikaci je tedy využito rozhraní Bluetooth, jenž obsahuje samotné NXT, počítač v podobě notebooku, i případné jiné mobilní zařízení. Rozšíření tohoto způsobu komunikace, přes všechny možné druhy mobilních zařízení, zvyšuje využitelnost vytvořeného komunikačního protokolu v této práci. Zároveň ovšem naráží na otázku dosahu, jelikož vysílače Bluetooth nejsou konstruovány na takový dosah, jako třeba u technologie Wi-Fi. Naměřený dosah při komunikaci s NXT byl dvacet metrů. Mezi oběma vysílači se ovšem nenacházeli žádné překážky, jenž by signálu bránily v dosažení cíle.

Než mohl začít samotný proces navrhování komunikačního protokolu, museli se obě zařízení „spárovat“, předat si své adresy, aby si mohli začít vyměňovat data mezi sebou. To bohužel nebylo možné, dokud nebyl na počítači přítomen *Fantom Driver* (ovladač pro komunikaci s Lego Mindstorms, volně dostupný na stránkách výrobce). Pak již vše proběhlo v pořádku.

4.3.1 Příkazy

S ohledem na rychlost bylo navrženo, že každý příkaz, který bude odeslán z počítače do NXT, bude mít číselné označení a bude se odesílat jen tento číselný kód. Pakety pak mohou být pouze jeden bajt velké. Sníží se ztrátovost informace, jelikož je odeslán právě jeden pro celý jeden příkaz. A samotné zpracování příjemcem probíhá velice rychle, jelikož stačí nalézt příslušnou metodu k provedení označenou právě tímto kódem.

Tabulka 2 Příkazy ovládající robota a jejich číselné kódy.

Klávesa	Číselný kód	Operace
Dopředu	1	
Dopředu + Vpravo	2	
Vpravo	3	
Dolů + Vpravo	4	
Dolů	5	
Dolů + Vlevo	6	
Vlevo	7	
Dopředu + Vlevo	8	
Mezerník	10	Zastaví NXT
Enter	11	Vyrovná NXT podél cesty

Zdroj: Vlastní

Data přenesená pomocí paketů, nejsou limitována jen na čísla o délce jednoho bajtu, nýbrž lze posílat i jiná datová typy, String, float atd.

Jak je vidět v tabulce, program zvládá i dvě stisknuté klávesy naráz, což je vhodné pro kombinování více příkazů. Počet není ovšem limitován na dvě, ale může být navýšen na libovolné číslo. V tomto případě bylo vhodnější omezit se jen na dvě.

Důležité je ještě zmínit speciální příkaz s kódovým označením *1000*. Ten totiž robotovi řekne, že se má spojení ukončit. To nikdy neuzavírá NXT, ale vždy počítač, jelikož je zamýšleno, aby robot vždy poslouchal „svého pána a nešel spát, pokud mu není dovoleno.“

4.3.2 Komunikace na straně NXT

Dříve, než se propojí robot a počítač, musí se nastartovat naslouchání na straně NXT a to ať pro komunikaci přes Bluetooth nebo USB. Pak se ze strany řídicího počítače odešle

příkaz pro navázání spojení. K zařízení a udržování spojení přes bezdrátovou variantu se využívá knihovna *BlueCove*, určena právě k tomuto účelu v jazyce Java. Ta je obsažena mezi balíčky projektu leJos a není jí tedy nutné nijak složitě shánět a implementovat.

V případě, že je signál mezi oběma zařízeními rušen, nebo se z nějakého jiného důvodu spojení ztratí, pak NXT přechází na naslouchací režim. Zastaví všechny motory, znovu se pokouší o spojení s počítačem a naslouchá, zda jej o toto propojení někdo nepožádá. Čeká ovšem jen na tu variantu, která byla použita prvně. Pokud tedy existovalo spojení pomocí USB, pak se bude očekávat dotaz přes USB. To samé pak platí i v případě Bluetooth.

Kontrola příkazů, jenž mu přicházejí, probíhá hned po přijetí příslušného paketu obsahujícího data. Pokud byl očekáván kód příkazu a místo něj došly naprosto jiná data, pak jsou ignorována a robot pokračuje v čekání na ty správné. Stejně to platí, pokud došla informace, která byla očekávána, ale například o špatném číselném označení – číslo příkazu, který není definován.

Kvůli případně připojeným skenerům a jiným vstupním zařízením, může stroj odesílat data počítači a ne jen naopak. V tomto případě je datový přenos poněkud větší, než jen nějaké číselné kódy zasílané počítačem. Toho je docíleno především nezpracováváním dat ze senzorů. Pokud jsou totiž informace získané ze vstupních zařízení příliš velké, pak „mozek“ robota je není schopen dostatečně rychle zpracovávat a vhodnější je zasílat je mnohem výkonnějšímu stroji. Pokud se ovšem jedná o jednoduché soubory dat, pak tato možnost není potřeba, nicméně i tak může být implementována do softwaru NXT pro budoucí případy.

4.3.3 Řešení

Aplikace v robotovi je složena ze dvou esenciálních tříd. Jednou z nich je komunikační třída *Comm*, ta se stará o spojení s počítačem a o přijímání a odesílání paketů. Velmi důležitou třídou ovšem je i *Main*, která se chová jako spouštěcí. Navíc ale obsahuje i základní logiku, jenž robot využívá k plnění jemu zaslanych příkazů. Ve třídě *Main* jsou definovány všechny třídy a zároveň jsou zde propojeny se všemi ostatními. Kdyby se program stal složitějším, pak by bylo nutné tuto logiku přenést do vlastní třídy a nezaplňovat s ní spouštěcí *Main*, to v tomto projektu ovšem nebylo nutné.

Samotný robot musí mít v sobě naprogramovanou i jistou pojistku proti „zaseknutí se“ v posledním příkazu. Toto by mohlo nastat, pokud by byl poslední příkaz například *jed' dopředu* a přitom by se přerušilo spojení. Stroj by pokračoval v cestě a snižoval by šance na obnovení signálu, za předpokladu, že na vinně byla vzdálenost obou zařízení, ovladače a ovládaného, od sebe. Při výpadku se pojistka sepne, dá pokyn motorům, ať se zastaví a začne se pokoušet o nové navázání spojení.

Na straně počítače je také esenciální třída *Comm*, jenž zodpovídá za propojení a komunikaci s NXT. Data která má odesílat, ale získává z *KeyboardListener*. Ten naslouchá klávesnici a ukládá si stisknuté klávesy do seznamu, který následně vyhodnocuje a určuje tak jaký číselný příkaz má třídě *Comm* předat k odeslání.

Aby se celá aplikace dala nějak odladit, veškeré příkazy a stavy NXT se zobrazují na jeho displeji. Jen i při nastartování aplikace nabízí, zda komunikace bude probíhat pomocí USB kabelu nebo Bluetooth.

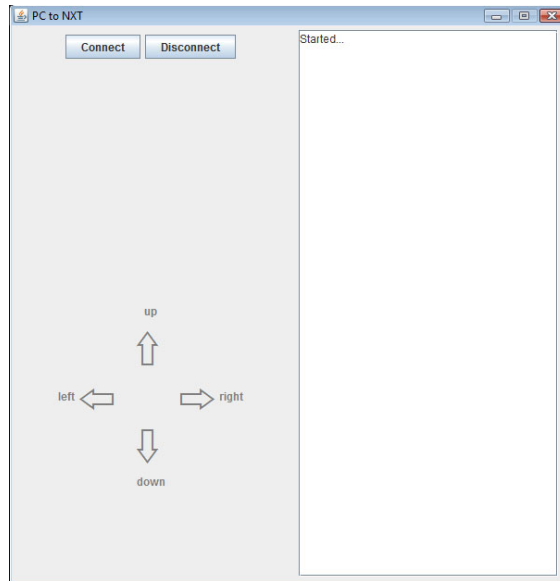


Obrázek 11 Displej robota zaznamenávající stavy a příkazy.

Zdroj: Vlastní

Na počítači ovšem je také nutné nějak zaznamenávat různé skutečnosti. Nabízelo se zapisovat tyto logy do konzole, ze které by se aplikace spouštěla. Problém ovšem byl, že aby fungovalo naslouchání stisknutým klávesám, musí být určeno, na čem má probíhat. Toto může fungovat jen na tzv. „zaměřitelném“ objektu, jako například grafické okno aplikace.

Proto vzniklo grafické uživatelské rozhraní pro aplikaci na straně počítače. V okně se nachází textové pole, do kterého se zaznamenávají stisknuté klávesy, různé stavy procesů atd. Je zde také graficky znázorněno, do jakých směrů by měl robot zamířit. A nemůžou chybět také tlačítka *připojit* a *odpojit* s NXT.



Obrázek 12 Počáteční grafické rozhraní aplikace.

Zdroj: Vlastní

4.4 Navigace v okolí

V tomto okamžiku nebylo vozítko příliš blízko autonomitě. Veškeré příkazy mu musel udílet člověk prostřednictvím počítače. To sice bylo jedním z cílů projektu, nicméně robot by mohl být poněkud inteligentnější, než dálkově ovládané autíčko. Přišlo tedy na přidání senzorů, jež mu umožní analyzovat své okolí a dají mu tak možnost alespoň omezeně reagovat na vzniklé situace. V nejhorším případě mohou alespoň informovat samotného uživatele a on ať rozhodne, co udělá. Tímto krokem by odpadla nutnost zvýšení komplexnosti softwaru, pomocí kterého robot operuje, ale zároveň by se zlepšila ovladatelnost robota. Člověk u pomyslného kormidla by získal další sadu informací o okolí, v němž se stroj vyskytuje a rozšířila by se mu tak pravděpodobnost, že správně vyhodnotí vzniklé situace.

4.4.1 Využití senzorů

V základním balení Lego Mindstorms NXT jsou obsaženy různé typy senzorů, jenž se dají využít ke zkoumání blízkého okolí vozítka. Z nabídky nejvíce vyhovoval *Ultrasonic sensor*, který funguje na principu echolokace. Jistém intervalu vždy vyšle zvukový signál o vysoké frekvenci a čeká, jestli se mu s časovým posunem navrátí zpět. Díky zpoždění může odhadovat vzdálenost překážek, od kterých se zvuk odrazil zpět. Přijímač dokáže

rozlišit, až osm různých odrazů a tudíž i osm potencionálních předmětů, na vzdálenostech mezi pěti a osmdesáti centimetry.

Tento senzor ovšem není naprosto bezchybný. Pokud totiž jsou překážky natočeny tak, že zvukový impuls odrazí úplně mimo přijímač, pak jsou pro stroj naprosto neviditelné, což není nijak žádoucí. Problém také nastává se schopnostmi samotného odrazu signálu. Některé materiály, jako dřevo, pěnová hmota a jiné „měkké“ materiály, špatně odrážejí zvuk a dokonce jej i pohlcují. Existují i takové, které dokonce nepohlcují, ale nechají jej projít skrz sebe a na druhé straně rezonují ve směru, kterém dopadli na objekt. Takto se například chovají kapaliny nebo sklo.

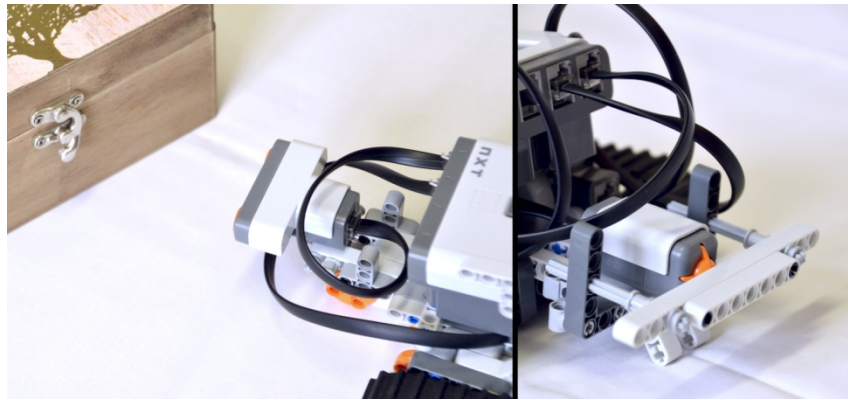
Při zahrnutí jisté tolerance je senzor schopen nakonec zachytit objekty rozmístěné před ním. Tato benevolence může být časová, neboli delší časový úsek pro čekání na výsledky, kdy se někdy stává, že i když při prvním vyslání zvukové vlny nic nezachytí, při druhém či třetím již ano. Toto se dá přisuzovat proměnlivosti prostředí, pohyb vzduchu, interference s jinou zvukovou vlnou atd. Další ústupek může být v prostém přijmutí, že nějaké procento překážek nedokáže robot zachytit, teda aspoň ne touto metodou.

I přes jisté klopýtání, se jeví Ultrasonic sensor, jako vhodný příspěvek pro orientaci robota v reálném prostředí. První myšlenka by mohla být, umístit senzor jako sonar, na otočnou plošinu na „krku,“ nad tělo vozítka. Stroj vždy popojel, zastavil se, pětkrát otočil „hlavou,“ kdy postupně naskenoval své okolí před sebou, vyhodnotil cestu a posunul se o kousek dál.

Z dat bylo možné zjistit, kde se nachází stěna a kde volný prostor. Byl samostatný a neboural do překážek, pokud je tedy dokázal zachytit, což se mu nakonec dařilo na výbornou. Mělo to ovšem jeden zásadní problém, byl neuvěřitelně pomalý. Při každém skenování bylo nutné čekat zhruba tři sta milisekund, aby byl schopen zachytit alespoň čtyři objekty a toto prováděl pětkrát, během jedné zastávky. Nepomohlo ani vynechání zastávky a provádění skenování za pomalejší jízdy. Chybovitost vzrostla mnohonásobně a tudíž i procento nárazů do překážek. Přes veškerou snahu se tato metoda nepovedla zdokonalit a byla zamítnuta s verdiktem, že echolokace se pro tento způsob nehodí.

Senzor ovšem zůstal na vozítku umístěn a to v předu, jako nárazník. Vozítko, oproti klasickému tlakovému čidlu sloužícímu často jako nárazník, dokáže detekovat překážky

v cestě s předstihem a předejít tak případným incidentům dříve, než nastanou. Bohužel tak nemůže nic „vidět“ za sebou, na řadu tedy přišlo tlakové čidlo sloužící jako nárazník.



Obrázek 13 Pohled na Ultrasonic sensor a zadní nárazník.

Zdroj: Vlastní

Jakmile je detekována překážka, ať už před robotem nebo za ním, robot se automaticky zastaví a do počítače odešle zprávu, že narazil na překážku. Samotné vyhodnocení dat probíhá přímo v řídicím modulu a ulehčuje to komunikaci mezi oběma zařízeními.

Nicméně stále nebyla splněna podmínka pro samostatné navádění, jak to ukázal Stanley v závodě DARPA. Bylo nutné přidat další zařízení, jež v kombinaci se stávajícími dokážou úspěšně navigovat stavěného robota.

4.4.2 Testovací scéna

Dříve než se přešlo k výběru nového senzoru zpřesňujícího navigaci vozidla, bylo třeba vymyslet a vytvořit testovací scénu, na které by se dal vyvinout a odladit. Muselo se jednat o téměř dokonalé prostředí, které by se eventuálně dalo přestavovat do stále komplexnějšího a tak bližšího reálnému.

Pozadí muselo být laděno v jednom barevném odstínu i intenzitě. Dokonce i materiálově mělo být co nejuniformnější, aby se předešlo zkreslování měřených dat pro robota nedůležitým a ignorovatelným prostředím. Pro tento případ byla zvolena bílá mírně lesklá plocha. S ohledem na variabilitu, přenositelnost i cenu nakonec byla vybrána role papíru, která se dala využít, jako pokrývka podlahy, ale rovnou i jako stěna znázorňující pozadí v dálce. Navíc se díky své měkkosti a ohebnosti dala různě vlnit, pro potřeby simulování nerovného terénu.

Dalším bodem bylo, jak vyznačit dráhu, po které by mohl robot jezdit. Jednou variantou může být, že se nakreslí přímo na plátno nebo papír, které slouží, jako podklad. Druhou pak, že bude cesta tvořena trochu proměnlivějším způsobem a to něčím, co je možné na podlahu jen položit a vytvarovat do chtěného tvaru. Černá matná stuha zastala tuto funkci na výbornou. Ne, že je dostatečně kontrastní k lesklému bílému papíru, ale lze ji bez problému nastavit dle požadavků.

Celá scéna se na konci pokusů může velmi lehce složit a uschovat, dokud nebude znovu potřeba, což umožňuje výbornou přenositelnost.

I když je mezi mnoha pokusy a roboty sledování nějaké předem vyznačené cesty, jedná se vždy pouze o jednu vodící linku. Tu kontroluje čidlo a snaží se na ní udržet robota, za každou cenu, což ovšem ukazuje na nepružnost této metody. Jakmile je totiž linka ztracena, vozidlo neví, kam pokračovat. Existuje mnoho řešení, jak je tento případ řešen. Otočí se a vrací se na začátek, pokračuje v cestě, dokud nenarazí na pokračování, začne kroužit okolo posledního známého bodu a hledat začátek další linky atd. Vždy se ale jedná o poměrně nemotorná a „umělá“ řešení, pokud je naprostým cílem autonomie pohybu robota.

V tomto projektu se nebude stroj orientovat podle jedné vodící linky, ale hned dvou. Tím, že se bude snažit držet mezi oběma a nenarazit do žádné z nich, se připraví algoritmus na orientaci v prostoru, kde jsou referenční čáry nahrazeny například zdmi. Díky stužce, která je naprosto volná, se dá navrhovat všelijaké uspořádání vodících čar. Nemusí tedy zůstat jen u rovnoběžných tvarů, ale je možné je různě „kroutit“ a ohýbat.



Obrázek 14 Nachystaná testovací scéna.

Zdroj: Vlastní

Překážky v této vykonstruované scéně mohou být jakékoli běžné předměty, se kterými se teoreticky může robot setkat.

4.4.3 Kamera a navigace

Vzhledem k tomu, že *Ultrasonic sensor* zklamal a testovací prostředí bylo připravené, zůstala problematika autonomní navigace robota nevyřešená. K dispozici nebyly žádné lasery a radary, ale využít se dala obyčejná webová kamera. V tomto případě *Trust Primo Webcam*.



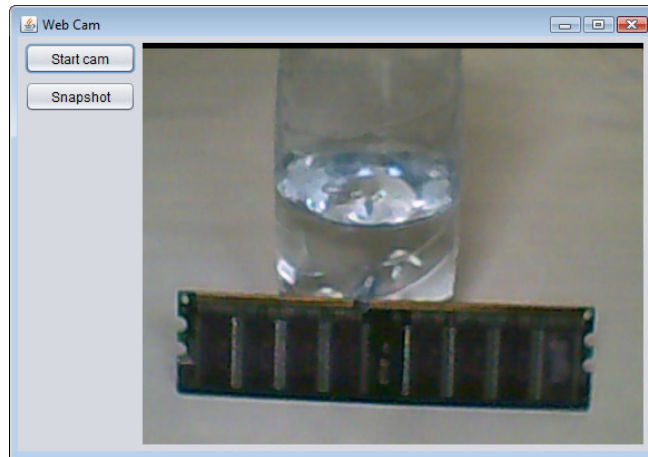
Obrázek 15 Webová kamera Trust Primo Webcam.

Zdroj: Vlastní

Díky ní se naskytla možnost přidělat vozítku „oči.“ Umožňovala totiž přenášet obraz uživateli u počítače. Ten tak nemusel mít vizuální kontakt se strojem a přesto věděl, kde je a co se okolo něj děje. Toto sice byl přínos pro ruční ovládání robota, ale nijak to zatím neřešilo otázku autonomie.

Obraz z kamery ovšem mohl být i ukládán a zpětně analyzován, což také bylo tím důvodem v tomto případě. Do paměti počítače, na který byl obraz vysílán, se neukládal celý záznam, nýbrž pouze jednotlivé snímky, v určitých intervalech stanovených na jednu sekundu.

K samotné práci s kamerou byla využita knihovna *sarxos.webcam*, umožňující využívat všemožné funkce webových kamer pomocí jazyka Java. Zpřístupnila přístup k datům přicházejícím z tohoto vstupního rozhraní, tvorbu celistvých snímků a jejich ukládání. K odladování chyb musel vyniknout grafický interfece, jinak by nebylo vůbec možné vyřešit různé problémy s rozlišením, zaostřením a světelností.



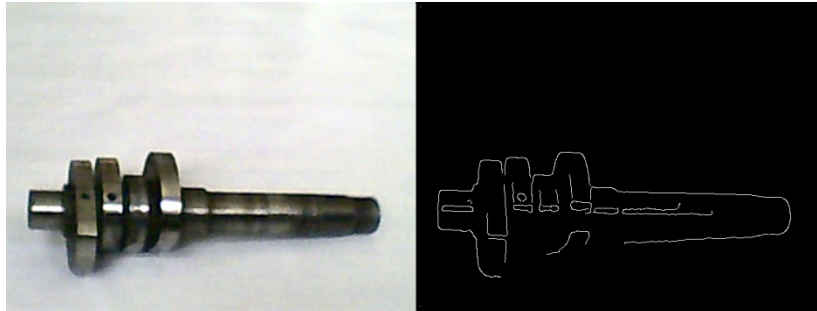
Obrázek 16 Grafické rozhraní pro odladění kamery.

Zdroj: Vlastní

Jelikož se jedná o levnější model příslušenství, neobsahuje automatické ostření a kdykoli je s kamerou manipulováno, musí být ručně doostřena. Samotná velikost obrazu, která byla přenášena do počítače, se ukázala také problematickou, jelikož je čip zachytávající obraz o nějakém rozlišení, základní nastavení v konstantách knihovny jsou naprosto odlišné. Optická soustava také není jedna z nejlepších možných, tím se projeví problémy s nasvícením prostředí, jež je snímáno. Muselo být dodán dostatek světla, jinak nebyly snímky nijak použitelné.

Jelikož byla kamera náhradou za zvukový senzor, byla umístěna na stejné místo jako byl původně on. Nad tělo robota, odkud mohla sledovat, co se děje před ním. Tato oblast byla již prozkoumávána čidlem na předku vozidla, nicméně i Stanley měl těchto zařízení více, aby docílil takové přesnosti, jakou předvedl. Nyní ovšem vedl od robota USB kabel k počítači, jelikož nebyl jiný způsob, jak tento určitý typ propojit. Poněkud se tak limitoval pohyb, projekt to ovšem posouvalo dále.

Na řadu přišla samotná detekce objektů a cesty. Nutné bylo přetransformovat obraz z kamery do nějakého použitelnějšího stavu, jelikož obsahoval příliš mnoho detailů, které se museli prakticky odfiltrovat. Využilo se metody hledání hran v obraze (*Canny Edge Detector*) – procházejí se jednotlivé pixely a nalézají se kontrasty a zbarvení jejich sousedů, přesahují-li tyto vlastnosti určenou hranici (*threshold*), pak je mezi nimi definována hrana. Složitá barevná fotografie se převedla na černobílý obrázek, kde bílé byly jen hrany jednotlivých předmětů.



Obrázek 17 Ukázka použití Canny Edge Detectoru.

Zdroj: Vlastní

Přišla na řadu analýza těchto snímků. Ty byly dále oříznuty podle přednastaveného obdélníku. Snížila se tak doba potřebná na zpracování obrazu a odfiltrovali se i objekty v pozadí, jelikož nebyly pro robota nijak zajímavé – důležité je jeho bezprostřední okolí.

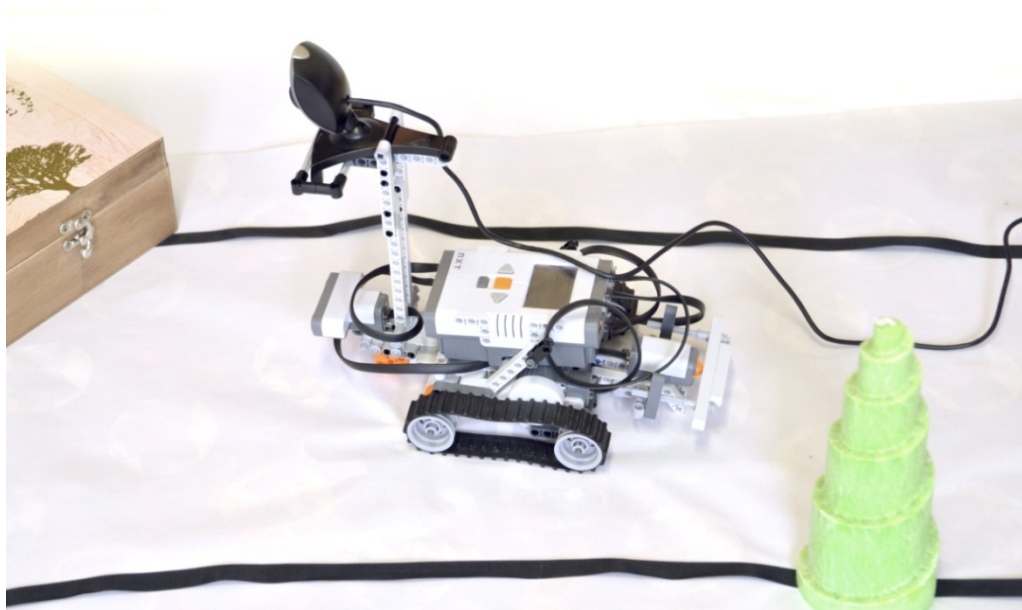
Výřezy se znovu procházeli obrazový bod po bodu a vždy se řadily k jednotlivým prozatím nalezeným „objektům.“ Každý z nich byl definován úsečkami, které byly postupně rozšiřovány, jak prozkoumané pixely přibývaly. Jakmile se prošel celý obraz, nejmenší úsečky byly „zahozeny,“ jelikož se daly považovat za chybu nebo šum. Zbylé se nakonec mohly seskupit do tvarů, pokud platilo, že vždy dvě byly svými sousedy.

Obraz zachycený kamerou byl zanalyzován a transformován na mnohonásobně jednodušší verzi. Tu už je možné poměrně rychle přenášet jakýmkoli datovým přenosem, jelikož se nejedná o velká množství pixelů, nýbrž několik logicky seřazených čísel.

5 VÝSLEDKY A DISKUZE

Výsledkem celé práce je robot na pásovém podvozku, který se dokáže spojit s aplikací v počítači, která jej umožňuje ovládat na dálku. Vozidlo je navigováno pomocí šipek na klávesnici a volně se přemísťuje po zemi. Díky pásovému podvozku nemá problém s jakýmkoli povrchem, i když se nepohybuje obzvlášť rychle.

Na přední a zadní straně jsou připevněna čidla sloužící jako nárazníky. V zadní části je umístěn tlakový senzor, který se chová jako typický nárazník. Dokud robot nenarazí při couvání do nějaké překážky, čidlo o ní neví a robot se nezastaví. Přední část ve srovnání se zadní funguje jako „intuitivní nárazník“, jelikož předvídá náraz. Sleduje prostor před vozidlem na vzdálenost osmdesáti centimetrů, přičemž zastaví motory, pokud se ve dvaceti centimetrech od něj objeví jakákoli překážka.

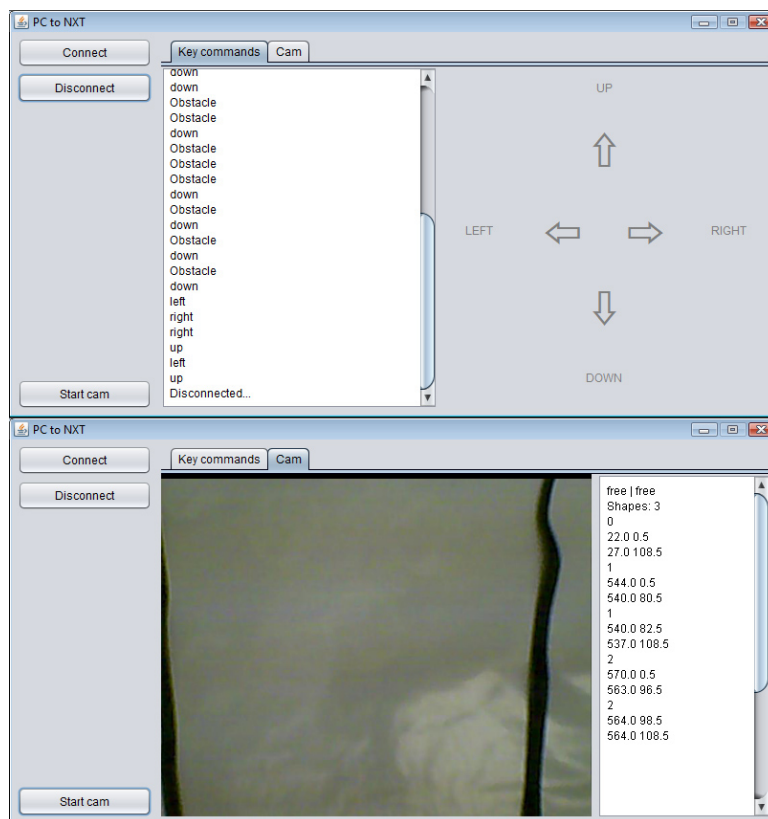


Obrázek 18 Robot nacházející překážky.

Zdroj: Vlastní

Grafická aplikace, která vznikla za účelem ovládání vozítka a ladění samotných principů, umožňuje člověku volně pohybovat robotem. Dále jej také informuje o stavech robota, ve kterých se momentálně nachází ať už robot samotný, nebo ve vztahu k ovládání. Například může stroj narazit na překážku nebo naprosto ztratit spojení s počítačem, přičemž o této skutečnosti aplikace vždy informuje.

K vozidlu je navíc přidána webová kamera, která rychle zvládá předávat obrázky počítači k dalšímu zpracování. Ve snímcích jsou programem rozpoznány jednotlivé předměty před robotem, které jsou následně převáděny do souřadnicového systému. Každý objekt je definován sérií úseček znázorňujících jejich viděný obrys.



Obrázek 19 Poslední verze uživatelského rozhraní, nalezení překážek.

Zdroj: Vlastní

Díky této analýze je možné, aby robot rozpoznal, zda se nachází stále na dráze, nebo již vyjíždí mimo ni. Tuto skutečnost také ohlašuje uživateli a ten se již sám může rozhodnout, zda bude ve směru pokračovat nebo ne.

Použitá webová kamera sama o sobě není nejvhodnějším zařízením pro tuto činnost, její pozorovací úhel činí pouze třicet stupňů, což není příliš vhodné, pokud má by měla být nakloněna směrem k zemi, aby „viděla“ těsně před robota. Chybovost této metody by se dala do jisté míry snížit, pokud by se přidala ještě jedna kamera a zavedlo by se tak *stereoskopické vidění*. Objekty by tak mohly být analyzovány, ne jako rovinné útvary, ale ve všech třech dimenzích. Výměnou kamer za kvalitnější by se také dosáhlo lepších rozpoznávacích schopností při horších světelných podmínkách a chybovost by dále klesla.

ZÁVĚR

Cílem práce bylo vytvořit řídicí systém pro robotickou stavebnici Lego Mindstorms NXT 2.0. Tohoto cíle bylo dosaženo a dále byl rozšířen tím, že k robotovi byla nainstalována kamera pro rozpoznávání okolí a dala mu tak možnost se samovolně orientovat v prostředí, jež bude v budoucnu možno využít při komplexnějších operacích.

K tomuto kroku inspiroval samotný průzkum vývoje robotiky jako vědy a jejího současného stavu ve světě. Cílem ani tak nebylo naučit tohoto specifického jedince být autonomní, ale připravit jej na propojení s nějakým výkonnějším strojem, například serverem. Ten by mohl být „mozkem“ ve vzájemné spolupráci a jeho „svěřeneč“ by mu jen zpřístupňoval zjednodušenou variantu dat o svém stavu a okolí, samozřejmě natolik komplexní, aby se mohl rozhodnout, co s robotem dál. Tento směr uvažování by s nejvyšší pravděpodobností vedl k dalšímu stupni práce, jelikož by se dal rozvinout do nepřeberných možností.

Samotný robot byl úspěšně sestaven ve formě vozítka a byla pro něj vytvořena aplikace do počítače v programovacím jazyce Java. S tou dokázal úspěšně komunikovat přes rozhraní Bluetooth, kdy mu byly předávány příkazy a on hlásil, pokud narazil na nějakou překážku, jež mu znemožňovala cestu. Uživatel měl plnou kontrolu nad vozítkem, kdy si případné nárazy hlídalo samo a zastavilo, pokud nějaký hrozil.

Součástí práce byl i průzkum aktuálního trhu s nabídkami pro robotické nadšence. Ten je poměrně široký a není problém nalézt potřebné součástky ke stavbě jakéhokoli druhu robota. Po porovnání nabízených produktů je ovšem jasné, že i když se Lego Mindstorms výborně používá a dobře se na něm vyvíjí, nepatří mezi nejlevnější sortiment. Jeho variantou, jen mnohem levnější, je proto nabídka společnosti Arduino, jež by mohla být využita k dalšímu rozvoji problematiky.

SEZNAM POUŽITÝCH ZDROJŮ

- ABRY, Vincent. *RobotShop Swat SuperDroid in the Best of Deals Woot!*. [online]. 27.4.2010 [cit. 2015-03-15]. Dostupné z: <http://www.robotshop.com/blog/en/robotshop-swat-superdroid-in-the-best-of-deals-woot-507>
- ACTIVE-ROBOTS. *Active-robots* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.active-robots.com/>
- ADACORE. *GNAT GPL Edition* [online]. 2014 [cit. 2015-03-15]. Dostupné z: <http://libre.adacore.com/tools/gnat-gpl-edition/>
- ARDUINO. *Arduino* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://arduino.cc>
- ARVIN, Farshad, et al. *Development of an autonomous micro robot for swarm robotics*. In: Mechatronics and Automation (ICMA), 2014 IEEE International Conference on. IEEE, 2014. p. 635-640.
- ATMEL. *Atmel* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.atmel.com>
- ATMEL. *Atmel Studio* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.atmel.com/tools/atmelstudio.aspx>
- AXON. *Axon* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.societyofrobots.com/axon/>
- BARILAR, Chase; MALCOLM, Matthew. *AMAZON PRIME AIR: COMMERCIAL DRONE USE AS THE FUTURE OF PRODUCT DELIVERY*. 2014.
- BDMICRO. *BDMicro* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.bdmicro.com/>
- BOSTON DYNAMICS. *BigDog Overview*. 22.11.2008. Dostupné z: http://www.bostondynamics.com/img/BigDog_Overview.pdf
- BOYLE, Rebecca. *Giant European Robofish Sniff Out Ocean Pollutants Autonomously*. [online]. 22.5.2012 [cit. 2015-03-15]. Dostupné z: <http://www.popsci.com/technology/article/2012-05/giant-robofish-sniff-out-ocean-pollutants-autonomously>
- COXWORTH, Ben. *Robotic lake lander could explore bodies of water on other planets*. [online]. 21.3.2012 [cit. 2015-03-15]. Dostupné z: <http://www.gizmag.com/tex-ii-robotic-lake-lander/21903/>
- DAGUROBOT. *Dagurobot* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.dagurobot.com/>
- DIGILENT. *Digilent* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.digilentinc.com>
- DIGILENT. *Digilent Embedded Linux* [online]. 16.1.2013 [cit. 2015-03-15]. Dostupné z: <http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,66,1143&Prod=EMBEDDED-LINUX>
- DUDEK, Gregory, et al. *A taxonomy for swarm robots*. In: Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on. IEEE, 1993. p. 441-447.
- DURÁN, Boris; THILL, Serge. *Rob's robot: current and future challenges for humanoid robots*. INTECH Open Access Publisher, 2012.
- ESA. *Touchdown! Rosetta's Philae probe lands on comet*. [online]. 12.11.2014 [cit. 2015-03-15]. Dostupné z: http://www.esa.int/Our_Activities/Space_Science/Rosetta/Touchdown!_Rosetta_s_Philae_probe_lands_on_comet
- FUJI YUSOKI. *Fuji yusoki 2005*. [online]. [cit. 2015-03-01]. Dostupné z: <http://www.fujiyusoki.com/company/history/>
- GRAY, Edwyn. *Nineteenth-century Torpedoes and their Inventors*. Naval Institute Press, 2004.
- GUMSTIX. *Gumstix* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <https://www.gumstix.com/>
- HABERZETTL, C. A. *Nanomedicine: destination or journey?*. *Nanotechnology*, 2002, 13.4: R9.
- HALVORSEN, Kåre. *MorpHex Project – Morphing Hexapod*. [online]. 19.11.2010 [cit. 2015-03-15]. Dostupné z: <http://blog.robotpark.com/designproject/morphex-project-morphing-hexapod-kare-halvorsen-81020/>

- HANSEN, John. Not eXactly C (NXC) Programmer's Guide. [online]. 10.10.2007 [cit. 2015-03-15]. Dostupné z: http://www.robosoutez.cz/files/nxc/NXC_Guide.pdf
- HARLOW, John. *Old rivalries surface as US races to sea's deepest spot*. [online]. 22.2.2009 [cit. 2015-03-15]. Dostupné z: http://www.thesundaytimes.co.uk/sto/news/world_news/article151892.ece
- HAYWARD, Rhodri. *The tortoise and the love-machine: Grey Walter and the politics of electroencephalography*. *Science in Context*, 2001, 14.04: 615-641.
- ILSTU. *KIND OF ROBOTS* [online]. 2014 [cit. 2015-03-15]. Dostupné z: http://www.mind.ilstu.edu/curriculum/medical_robotics/kinds_of_robots.php
- IROBOT. *IRobot: Our History* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.irobot.com/About-iRobot/Company-Information/History.aspx>
- JAEGER, Robert J. Rehabilitation robotics research at the National Institute on Disability and Rehabilitation Research. *Journal of Rehabilitation Research & Development*, 2006, 43.5: xvii.
- KAMAL, Ibrahim. Line tracking sensors and algorithms. [online]. 28.2.2008 [cit. 2015-03-15]. Dostupné z: <http://www.ikalogic.com/line-tracking-sensors-and-algorithms/>
- KATO, Shin; TOMITA, Kohji; TSUGAWA, Sadayuki. Visual navigation along reference lines and collision avoidance for autonomous vehicles. In: *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*. IEEE, 1996. p. 385-390.
- KOMATSU, Corporate Communications. *Launching intelligent Machine Control Medium-sized Bulldozer Equipped with World's First Fully Automatic Blade Control*. In: [online]. 16.4.2013 [cit. 2015-03-15]. Dostupné z: <http://www.komatsu.com/CompanyInfo/press/2013041615252928343.html>
- KRAJNÍK, Tomáš, et al. AR-drone as a platform for robotic research and education. In: *Research and Education in Robotics-EUROBOT 2011*. Springer Berlin Heidelberg, 2011. p. 172-186.
- LAUBACH, Sharon L.; BURDICK, Joel W. *An autonomous sensor-based path-planner for planetary microrovers*. In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*. IEEE, 1999. p. 347-354.
- LEGO. *31313 MINDSTORMS EV3 - Mindstorms LEGO.com* [online]. 2014 [cit. 2015-03-15]. Dostupné z: <http://www.lego.com/en-us/mindstorms/products/31313-mindstorms-ev3>
- LEGO. *History - Mindstorms LEGO.com* [online]. 2014 [cit. 2015-03-15]. Dostupné z: <http://www.lego.com/en-us/mindstorms/history>
- LEGO. *Nauč se programovat* [online]. 2014 [cit. 2015-03-15]. Dostupné z: <http://www.lego.com/cs-cz/mindstorms/learn-to-program>
- LEJOS. *LeJos* [online]. 2009 [cit. 2015-03-15]. Dostupné z: <http://www.lejos.org/>
- LETSMAKEROBOTS. *How to make your first robot* [online]. 30.1.2008 [cit. 2015-03-15]. Dostupné z: <http://letsmakerobots.com/start>
- LILJEBÄCK, Pål. Anna Konda – The fire fighting snake robot. [online]. 2012 [cit. 2015-03-15]. Dostupné z: <http://robotnor.no/research/anna-konda-the-fire-fighting-snake-robot/>
- LUK, Bing L.; LIU, Louis KP; COLLIE, Arthur A. *Climbing service robots for improving safety in building maintenance industry*. INTECH Open Access Publisher, 2007.
- MAHAL, Igor. *Strážce na čtyřech kolech: Robot hlídá nebezpečný arzenál*. [online]. 28.9.2013 [cit. 2015-03-15]. Dostupné z: <http://www.stoplusjednicka.cz/strazce-na-ctyrech-kolech-robot-hlida-nebezpecny-arzenal>
- MANN, Richard. *How to Program an 8-bit Microcontroller Using C language*. 2013. Dostupné z: http://www.atmel.com/images/avr_3_04.pdf
- MATHAS, Carolyn. Robots follow invisible magnetic tracks. [online]. 9.1.2013 [cit. 2015-03-15]. Dostupné z: <http://www.edn.com/electronics-products/other/4404638/Robots-follow-invisible-magnetic-tracks>

- MINATO, Takashi, et al. Development of an android robot for studying human-robot interaction. In: *Innovations in applied artificial intelligence*. Springer Berlin Heidelberg, 2004. p. 424-434.
- MODROBOTICS. *Modrobotics* [online]. 2012 [cit. 2015-03-15]. Dostupné z: <http://www.modrobotics.com/>
- MØLHAVE, Kristian, et al. *Pick-and-place nanomanipulation using microfabricated grippers*. *Nanotechnology*, 2006, 17.10: 2434.
- MULLER, Thomas. *AUTOMATED GUIDED VEHICLES*. 17.11.1999. ISBN 00771850.
- NASA. *Nasa* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.nasa.gov/>
- NEEDHAM, Joseph (1991). *Science and Civilisation in China: Volume 2, History of Scientific Thought*. Cambridge University Press. ISBN 0-521-05800-7.
- OBRINGER, Lee Ann a Jonathan STRICKLAND. *Honda ASIMO Robot*. [online]. 2011 [cit. 2015-03-14]. Dostupné z: <http://science.howstuffworks.com/asimo.htm>
- PALMER, Jason. *Robotic firefighting team debuts*. [online]. 28.2.2009 [cit. 2015-03-15]. Dostupné z: <http://news.bbc.co.uk/2/hi/8172739.stm>
- PAVLŮSEK, Ondřej. *BMW představí automatické parkování bez řidiče*. [online]. 16.12.2014 [cit. 2015-03-15]. Dostupné z: <http://www.auto.cz/bmw-predstavi-automaticke-parkovani-ridice-84658>
- PBFORTH. *PbForth* [online]. 2013 [cit. 2015-03-15]. Dostupné z: <http://www.tldp.org/HOWTO/Lego/pbforth.html>
- PEARCE, Jeremy. *George C. Devol, Inventor of Robot Arm, Dies at 99*. [online]. 15.8.2011 [cit. 2015-03-15]. Dostupné z: www.nytimes.com/2011/08/16/business/george-devol-developer-of-robot-arm-dies-at-99.html?_r=0
- POWER, Miguel. *HOWTO create a Line Following Robot using Mindstorms*. [online]. 13.3.2014 [cit. 2015-03-15]. Dostupné z: <http://thetechnicgear.com/2014/03/howto-create-line-following-robot-using-mindstorms/>
- PŘIBYL, Tomáš. *Rudé hvězdy ve vesmíru*. 1997. ISBN 80-902352-1-2.
- RAO, Nageswara SV; IYENGAR, S. Satharama. Autonomous robot navigation in unknown terrains: Incidental learning and environmental exploration. *Systems, Man and Cybernetics, IEEE Transactions on*, 1990, 20.6: 1443-1449.
- RESNECK. *Polar Bear: Spy On The Ice*, 8.3.2011. [online]. [cit. 2015-03-02]. Dostupné z: <http://www.coolhunting.com/tech/polar-bear-spy>
- RIO TINTO. *Rio Tinto boosts driverless truck fleet to 150 under Mine of the Future TM programme*. [online]. 2.11.2011 [cit. 2015-03-15]. Dostupné z: http://www.riotinto.com/documents/111102_Rio_Tinto_boosts_driverless_truck_fleet_to_150_under_Mine_of_the_Future_programme.pdf
- ROBO MATTER. *Robotc* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.robotc.net/download/>
- ROTH, Bernard; RASTEGAR, Jahangir; SCHEINMAN, Victor. *On the design of computer controlled manipulators*. In: *On Theory and Practice of Robots and Manipulators*. Springer Vienna, 1974. p. 93-113.
- RÜKL, Antonín. *Atlas Mésice*. 1991. ISBN 80-85277-10-7.
- RUSSELL, Steve. *DARPA Grand Challenge Winner: Stanley the Robot!*. [online]. 8.1.2006 [cit. 2015-03-15]. Dostupné z: <http://www.popularmechanics.com/technology/robots/a393/2169012/>
- SHEPHERD, Robert F., et al. Multigait soft robot. *Proceedings of the National Academy of Sciences*, 2011, 108.51: 20400-20403.
- SHERIDAN, Thomas B. *Telerobotics, automation, and human supervisory control*. MIT press, 1992.
- SONG, Sang-Eun, et al. *Development of a pneumatic robot for MRI-guided transperineal prostate biopsy and brachytherapy: New approaches*. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010. p. 2580-2585.
- SPHERO. *GoSphero* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.gosphero.com/sphero/>

- SVOBODA, Elizabeth. *PopSci's Darpa Grand Challenge Preview: Update #4*. [online]. 6.10.2005 [cit. 2015-03-15]. Dostupné z: <http://www.popsci.com/scitech/article/2005-10/popscis-darpa-grand-challenge-preview-update-4>
- THRUN, Sebastian, et al. Stanley: The robot that won the DARPA Grand Challenge. *Journal of field Robotics*, 2006, 23.9: 661-692.
- THRUN, Sebastian. *What we're driving at*. [online]. 9.11.2010 [cit. 2015-03-15]. Dostupné z: <http://googleblog.blogspot.cz/2010/10/what-were-driving-at.html>
- TOUFAR, Pavel a Miroslav BALOUS. Miroslav. *Cesty ke hvězdám*. 1976. ISBN 13-767-76.
- UNGERLEIDER, Neal. See What You Can Do With Drone Filmmaking. [online]. 31.1.2013 [cit. 2015-03-15]. Dostupné z: <http://www.fastcocrete.com/1682320/see-what-you-can-do-with-drone-filmmaking>
- URMSON, Chris, et al. Tartan racing: A multi-modal approach to the darpa urban challenge. 2007.
- VEXROBOTICS. *Vexrobotics* [online]. 2015 [cit. 2015-03-15]. Dostupné z: <http://www.vexrobotics.com>
- WAURZYNIAK, Patrick. *Masters of Manufacturing: Joseph F. Engelberger*. 2006.
- WETTERGREEN, David; THORPE, Chuck; WHITTAKER, Red. *Exploring Mount Erebus by walking robot*. Robotics and Autonomous Systems, 1993, 11.3: 171-185.
- WHITE, Tammy. *Making Sailors 'SAFFiR' - Navy Unveils Firefighting Robot Prototype at Naval Tech EXPO*. [online]. 4.2.2015 [cit. 2015-03-15]. Dostupné z: http://www.navy.mil/submit/display.asp?story_id=85459
- WHITTLE, Richard. *Predator: the secret origins of the drone revolution*. First edition. 2013, 353 pages, 8 pages of unnumbered plates. ISBN 978-0805099645.
- YIM, Mark; ZHANG, Ying; DUFF, David. Modular robots. *Spectrum, IEEE*, 2002, 39.2: 30-34.
- ZHANG, Zheng, et al. Path planning and navigation for mobile robots in a hybrid sensor network without prior location information. *Int J Adv Robotic Sy*, 2013, 10.172: 2013.
- ZUNT, Dominik. *Who did actually invent the word "robot" and what does it mean?*. The Karel Čapek website. Retrieved 2007-09-11.
- ŽÍDEK, Jan. *GRAFICKÉ PROGRAMOVÁNÍ VE VÝVOJOVÉM PROSTŘEDÍ LabVIEW*. 2002. Výuková skripta. VŠB-TU Ostrava.

PŘÍLOHY

Součástí práce jsou elektronické soubory, které byly využity v této práci, nebo jsou jejím produktem. Ty jsou umístěny na přiloženém CD.

Seznam příloh:

1. *aplikace-NXT.zip*

Aplikace pro leJos v NXT.

2. *aplikace-PC.zip*

Aplikace pro počítač, k ovládání NXT.

3. *lib.zip*

Knihovna potřebná pro spuštění aplikací, musí se zavést do složek „lib.“