

Ovládání laboratorního modelu robota Mindstorms NXT (machine robot) pomocí PC

Computer control of laboratory model of the Mindstorm NXT robot

Bc. Bohuslav Tmej

Diplomová práce
2008

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2007/2008

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Bohuslav TMEJ**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Téma práce: **Ovládání laboratorního modelu robota Mindstorms NXT (machine robot) pomocí PC**

Zásady pro vypracování:

1. Provedte literární studii softwarového a hardwarového vybavení robota NXT.
2. Provedte studii programovacího jazyka, možnosti komunikace atd. (Java, Labview, C++, DotNet a další).
3. Provedte návrh ovládacího programu (možno i webové aplikace) s ohledem i na jiné předměty vyučované na FAI.
4. Provedte návrhy laboratorních úloh a zpracujte pro ně dokumentaci.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **PERDUE, David J. THE UNOFFICIAL LEGO MINDSTORMS NXT INVENTOR S GUIDE.** Megan Dunchak; Christina Samuelli. 1st edition. San Francisco : No Starch Press, Inc, c2008. 296 s. ISBN 978-1-59327-154-1.
2. **ASTOLFO, Dave, FERRARI, Mario, FERRARI, Giulio. BUILDING ROBOTS WITH LEGO MINDSTORMS NXT.** Audrey Doyle. 1st edition. Burlington : Syngress Publishing, Inc, c2007. 447 s. ISBN 978-1-59749-152-5.
3. **SCHOLZ, Matthias Paul. Advanced NXT: The Da Vinci Inventions Book.** Jennifer Whipple. Apress, c2007. 369 s. ISBN 978-1-59059-843-6.
4. **GASPERI, Michael, HURBAIN, Philippe, HURBAIN, Isabelle. Extreme NXT : Extending the LEGO MINDSTORMS NXT to the Next Level.** Susannah Davidson Pfalzer. Is.l.I : Apress, c2007. 286 s. ISBN 978-1-59059-818-4.
5. **BOOGAARTS, Martijn, et al. THE LEGO MINDSTORMS NXT IDEA BOOK : design, invent, and built.** Nancy Sixsmith, Megan Dunchak. 1st edition. San Francisco : No Starch Press, Inc, c2007. 344 s. ISBN 978-1-59327-150-3.
6. **BAGNALL, Brian. Maximum LEGO NXT : Building Robots with Java Brains.** Edited by Sylvia Philipps. 1st edition. Canada : Variant Press, c2007. 505 s. ISBN 978-0-9738649-1-5.

Vedoucí diplomové práce:

Ing. Roman Šenkeřík

Ústav aplikované informatiky

Datum zadání diplomové práce:

20. února 2008

Termín odevzdání diplomové práce:

19. května 2008

Ve Zlíně dne 20. února 2008

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Tato diplomová práce se zabývá vytvořením aplikace určené k ovládní laboratorního modelu robota Mindstorm NXT (Machine) a dále návrhem laboratorních úloh využívajících zmíněného robota.

V teoretické části se práce věnuje popisu softwarového a hardwarového vybavení robota NXT. A dále se věnuje možnostem jeho programování a komunikace.

Praktická část se zabývá podrobným popisem vytvořené aplikace ovládající model Mindstorms NXT (Machine) a popisem úloh, vytvořených pro podporu výuky programovacího jazyka C++ a práce v programovém prostředí MATLAB.

Klíčová slova: LEGO, Mindstorms NXT, robot, laboratorní úlohy

ABSTRACT

This master thesis deals with creating an application for computer control of laboratory model of the Mindstorm NXT robot (Machine) and with design of the laboratory exercises which make use of this robot.

The thesis attends to description of software and hardware equipment of the NXT robot in the theoretical part. Next this part discusses possibility of its programming and communication.

The practical part deals with detailed description of the created application to control Mindstorm NXT robot (machine). Next, there is a description of exercises, which were created to support education of C++ programming language and working with Matlab programming environment.

Keywords: LEGO, Mindstorms NXT, robot, exercise

Děkuji vedoucímu práce Ing. Romanu Šenkeříkovi za odbornou pomoc, za věcné připomínky při vedení práce, poskytnuté materiály a ochotu při řešení problémů.

Především bych chtěl ale poděkovat rodičům, za jejich podporu, jež mi umožnila diplomové práce dosáhnout.

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně, dne 1.9.2008

.....
Bohuslav Tmej

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 HARDWAROVÉ A SOFTWAREVÉ VYBAVENÍ ROBOTA NXT	10
1.1 HARDWAROVÁ VYBAVENÍ ROBOTA NXT	11
1.1.1 Ovládací jednotka NXT	11
1.1.2 Senzory a servomotory	12
1.1.2.1 Dotykový senzor	13
1.1.2.2 Zvukový senzor	13
1.1.2.3 Světelný senzor	14
1.1.2.4 Ultrazvukový senzor	15
1.1.2.5 Servomotory	16
1.1.3 Technologie Bluetooth	17
1.2 SOFTWAREVÉ VYBAVENÍ ROBOTA NXT	17
1.2.1 NXT – G	17
1.2.1.1 Systémové požadavky	17
1.2.1.2 Uživatelské rozhraní	18
1.2.2 Firmware NXT	20
2 MOŽNOSTI PROGRAMOVÁNÍ A KOMUNIKACE NXT	23
2.1 PROGRAMOVÁNÍ V PROSTŘEDÍ MATLAB	23
2.1.1 MATLAB	23
2.1.2 RWTH Mindstorms NXT Toolbox	24
2.2 PROGRAMOVACÍ JAZYK C, C++	24
2.2.1 Knihovna NXT++	25
2.3 PROGRAMOVACÍ JAZYK C#	25
2.3.1 Knihovna LEGO NXT.NET	26
2.4 JAVA	27
2.4.1 LeJOS	28
2.5 MICROSOFT ROBOTIC STUDIO	29
2.5.1 Visual Programming Language	29
2.6 ROBOTC	30
2.7 IGR 31	
2.8 MOŽNOSTI KOMUNIKACE	32
2.8.1 Komunikace pomocí USB kabelu	32
2.8.2 Komunikace pomocí Bluetooth	33
II PRAKTICKÁ ČÁST	35
3 NÁVRH OVLADACÍHO PROGRAMU	36
3.1 STRUČNÝ POPIS FUNKCÍ KNIHOVNY NXT.NET	38
3.2 UKÁZKA POUŽITÍ KNIHOVNY LEGO NXT.NET	40
3.2.1 Připojení k NXT a zobrazení systémových informací	40

3.2.2	Získání informací ze senzorů NXT	41
3.2.3	Ovládání motorů.....	41
4	NÁVRH LABORATORNÍCH ÚLOH	43
4.1	LABORATORNÍ ÚLOHY DO PROGRAMOVÉHO SYSTÉMU MATLAB	43
4.1.1	Seznam funkcí knihovny RWHT	43
4.1.2	Úloha č. 1 - Získání a zobrazení údajů ze senzorů NXT.....	49
4.1.3	Úloha č. 2 - Ovládání robota Mindstorm NXT Machine	54
4.2	LABORATORNÍ ÚLOHA DO VÝUKY PROGRAMOVACÍHO JAZYKA C	59
4.2.1	Popis funkcí knihovny NXT++	59
4.2.2	Úloha č.3 - Získání a zobrazení údajů ze senzorů NXT a ovládání motorů robota Mindstorms NXT	64
4.3	LABORATORNÍ ÚLOHA DO VÝUKY PROGRAMOVACÍHO JAZYKA C#	70
	ZÁVĚR	71
	ZÁVĚR V ANGLIČTINĚ	73
	SEZNAM POUŽITÉ LITERATURY	75
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	77
	SEZNAM OBRÁZKŮ	78
	SEZNAM TABULEK.....	80
	SEZNAM PŘÍLOH.....	81

ÚVOD

Lidstvo již odedávna fascinovala myšlenka vytvoření entity s umělou inteligencí, která by za člověka vykonávala těžkou nebo nezáživnou práci. Tohoto úkolu se velice dobře zhostila vědní disciplína nazývaná kybernetika. Jedním jejím odvětvím je i robotika, která se zabývá vývojem strojů, které pomalu začínají nahrazovat člověka tam, kde je lidský organizmus ohrožen, nebo je lidská činnost vyloučena. Tyto stroje se obecně označují jako roboti. Slovo robot bylo poprvé použito v dramatu RUR od Karla Čapka. V jeho pojetí znamená „Robot“ mechanického člověka, neboli dokonalý stroj v přesné lidské podobě. Tento termín se velice rychle rozšířil po celém světě a v dnešní době se používá pro označení samostatně pracujícího stroje vykonávajícího určené úkoly. Roboti například nahrazují lidskou práci v těžkých podmínkách a nedostupném prostředí (pod vodou), nebo v prostředí životu nebezpečném (radiace, vysoké teploty). Slovo robot též označuje automatické zařízení, mající schopnost reagovat na určité podněty ze svého okolí a zároveň na toto okolí zpětně působit.

Robotika je rychle se rozvíjející obor. Nelze se tedy divit, že se stále častěji objevuje i ve studijních osnovách mnohých středních a vysokých škol. Jedním z cenných nástrojů pro podporu výuky je počítačově řízený model LEGO Mindstorms NXT, který lze významnou měrou využít k účinnému vzdělávání mladé generace v tomto odvětví.

Úkolem této diplomové práce je vytvoření aplikace sloužící k ovládní laboratorního modelu robotické ruky sestavené ze stavebnice LEGO Mindstorm NXT a návrh laboratorních úloh využívajících zmíněného robota.

Dalším úkolem je zpracování literární studie softwarového a hardwarového vybavení robota NXT, studie programovacího jazyka a možnosti komunikace zmiňovaného robota. Tímto úkolem se zabývá teoretická část této diplomové práce.

Praktická část se pak zabývá podrobným popisem vytvořené aplikace ovládající model Mindstorms NXT (Machine) a dále popisuje úlohy vytvořené pro podporu výuky programovacího jazyka C, C++, C# a práce v programovém prostředí MATLAB.

I. TEORETICKÁ ČÁST

1 HARDWAROVÉ A SOFTWAREVÉ VYBAVENÍ ROBOTA NXT

Sada MINDSTORMS NXT představuje ukázkou nejnovější robotové techniky. Kombinuje inteligentní kostku s mikropočítačovým mozkiem, důmyslné senzory a programový software s jednoduchým použitím typu „táhni a pusť“. Sada navíc obsahuje 519 LEGO součástek, pomocí kterých můžete postavit z této stavebnice prakticky cokoliv. Můžete tak vytvořit například model některého z vynálezů Leonarda da Vinci.[1] Samotná stavebnice obsahuje návod na postavení čtyř jedinečných robotů. Lze tak postavit například robota pro manipulaci s míčky, štíra nebo Alpha Rexe, který je zobrazen na Obr. 1. a který patří k nejzajímavějším a nejnáročnějším robotům, které lze ze stavebnice postavit. Posledním robotem, který lze ze stavebnice podle přiloženého návodu postavit, je robotická ruka, kterou se bude zabývat tato diplomová práce.



Obr. 1. Robot Mindstorms NXT

1.1 Hardwarová vybavení robota NXT

1.1.1 Ovládací jednotka NXT

Základním prvkem stavebnice a současně „mozkem“ celého robota je centrální řídicí jednotka označovaná NXT, která je řízena 32-bitovým ARM7 procesorem. Disponuje 256 KB flash pamětí a 64 KB RAM pamětí. Dále obsahuje 8-bitový AVR koprocessor.

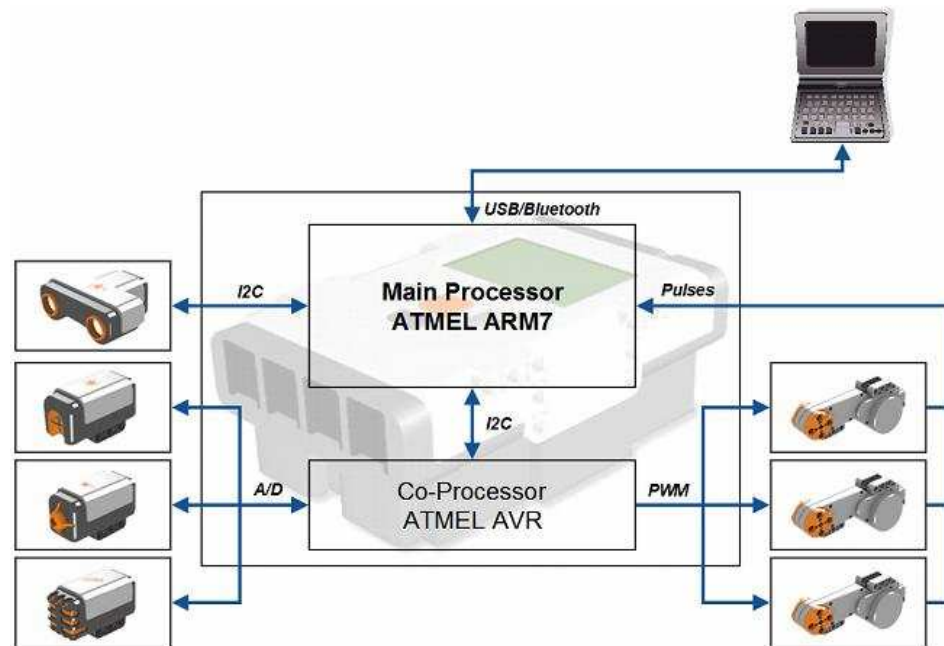
Dále je NXT vybaven grafickým LCD displejem s rozlišením 100 x 64 bodů a reproduktorem s 8 kHz vzorkovací frekvencí.



Obr. 2. NXT

Pro komunikaci s okolím, kvůli možnosti programování, disponuje USB portem a též Bluetooth 2.0 připojením. S tímto bezdrátovým připojením je možné robota, mimo jiného řídit pomocí mobilních telefonů s podporou technologie Bluetooth.

NXT má 4 vstupní porty pro připojení senzorů a 3 výstupní porty pro připojení servomotorů. Jednotka, a s ní i celý robot, je napájena šesti AA bateriemi.



Obr. 3. NXT – komunikace s okolím

1.1.2 Sensory a servomotory

Sada LEGO Mindstorms NXT obsahuje tři servomotory, které lze zároveň využít jako senzory pro měření otáček. Dále pak obsahuje dotykový senzor, světelný senzor, zvukový senzor a ultrazvukový senzor. Způsob jejich připojení je k nahlédnutí na Obr. 4. a popis jednotlivých senzorů následuje hned za ním.



Obr. 4. NXT – připojení senzorů a motorů

1.1.2.1 Dotykový senzor

Dotykový senzor poskytuje robotu schopnost hmatu. V podstatě se jedná o tlačítko, které zaznamená stisknutí a následné uvolnění. Řídící jednotka NXT zobrazuje vždy buď hodnotu 0, pokud tlačítko není stisknuté, a nebo hodnotu 1, pokud stisknuté je.

Dotykový senzor může být využit k tomu, aby robot uchopil nějakou věc. Vybaví se jím paže robota, která dá robotu informaci o tom, zda je v ní něco, co by se dalo uchopit či nikoliv. Dále dotykový senzor může být použit v případě, kdy je vyžadováno, aby robot reagoval na pokyn. Stisknutím dotykového senzoru můžeme robota přimět například k mluvení, chůzi v před, zapnutí televize nebo nějakému složitějšímu úkonu.[2]-[5]



Obr. 5. Dotykový senzor

1.1.2.2 Zvukový senzor

Pomocí tohoto senzoru robot slyší. Zvukový senzor dokáže zaznamenávat akustický tlak ve dvou režimech, a to v decibelech a v tzv. upravených decibelech.

Při zaznamenání upravených decibelů se citlivost senzoru přizpůsobí citlivosti lidského ucha. Jinými slovy se jedná o zvuky, které lidské ucho dokáže zachytit.

Při zaznamenání standardních decibelů jsou všechny zvuky měřeny shodnou citlivostí. Tyto zvuky obsahují i takové zvuky, které jsou pro lidské ucho příliš vysoké nebo nízké.

Zvukový senzor dokáže měřit akustický tlak až do hodnoty 90 dB. Tato úroveň odpovídá hluku, který vydává sekačka na trávu nebo jedoucí vlak. Hladiny akustického tlaku jsou

velmi složité, proto jsou data na displeji NXT uvedena v procentech. Pro ukázkou některé příklady:

- 4-5% odpovídá tichému obývacímu pokoji
- 5-10% odpovídá vzdálenému hovoru
- 10-30% hodnota běžné konverzace v blízkosti senzoru nebo hudba běžné hlasitosti
- 30-100% tato hodnota odpovídá velmi nahlas puštěné hudbě, příp. když lidé křičí[2]-[5]



Obr. 6. Zvukový senzor

1.1.2.3 Světelný senzor

Světelný senzor je jedním ze dvou senzorů, které dávají robotu schopnost vidět (tím druhým je ultrazvukový senzor). Světelný senzor umožňuje robotu například rozlišit světlo a tmu.



Obr. 7. Světelný senzor

Senzor pracuje ve dvou režimech. V prvním pouze pasivně měří intenzitu světla v okolí a zobrazuje ji v procentech na displeji NXT. Ve druhém režimu se rozsvítí červená dioda, s jejíž pomocí může robot podle množství odraženého světla rozpoznat například různé barvy. Na Obr. 8. je srovnání citlivosti lidského oka a světelného senzoru. [2]-[5]



Obr. 8. Citlivost světelného senzoru při rozeznávání barev

1.1.2.4 Ultrazvukový senzor

Ultrazvukový senzor je dalším senzorem, který robotu umožňuje vidět. Pomocí ultrazvuku se robot dokáže orientovat v prostoru, je schopný nalézat překážky a určit vzdálenost od těchto překážek. Může být tedy využit k tomu, aby se robot dokázal vyhnout překážkám, aby odhadl vzdálenost a zaznamenal pohyb. Zjištěnou vzdálenost zobrazuje NXT na displeji v palcích nebo centimetrech. Dokáže změřit vzdálenost od 0 do 255 cm s přesností +/- 3 cm.

Ultrazvukový senzor využívá stejných vědeckých principů jako netopýři. Měří vzdálenost na základě výpočtu doby, během níž dorazí k předmětu zvuková vlna a znovu se vrátí. Stejně jako ozvěna. Nejlépe se získávají data o předmětech velkých rozměrů. Předměty vyrobené z měkkých materiálů a zaoblených tvarů nebo předměty, které jsou příliš tenké nebo malé, hledá senzor obtížněji.

Nevýhodou je, že jsou-li v jedné místnosti dva či více ultrazvukových senzorů, mohou se vzájemně rušit. [2]-[5]



Obr. 9. Ultrazvukový senzor

1.1.2.5 Servomotory

Pohyb robota umožňují tři servomotory. Výhodou také je, že pokud se v softwarovém programu zvolí možnost „Move block“ (příkaz pro pohyb), dojde k automatické synchronizaci dvou motorů, takže se robot začne pohybovat rovně.

Každý z těchto motorů má navíc zabudovaný rotační senzor otáček, což umožňuje velmi přesné řízení. Rotační senzor měří otáčení motoru ve stupních nebo celkové otáčení (s přesností +/- jeden stupeň).

Vestavěný rotační senzor v každém motoru umožňuje také nastavení různých rychlostí motorů (nastavením různých výkonnostních parametrů v softwaru). [2]-[5]



Obr. 10. Servomotor

1.1.3 Technologie Bluetooth

Bluetooth je technologie, díky níž je možno posílat a získávat data bez použití drátů či kabelů. Díky tomuto zařízení se dají vyměňovat programy mezi jednotlivými NXT nebo lze řídit bezdrátové připojení mezi svým počítačem a robotem a vyzkoušet programy okamžitě, i když bude robot na opačné straně místnosti.

Je-li navíc k dispozici mobilní telefon s možností Bluetooth, může být využit k ovládní robotu. Může být dokonce použit jako kvalitní senzor, jako je senzor fotoaparátu. [6]

Samotný software dodávaný k sadě Mindstorms a firmware nainstalovaný v řídicí kostce NXT mnoho zařízení Bluetooth nepodporuje. Novější verze firmware si však již naštěstí poradí s libovolným Bluetooth adaptérem.

1.2 Softwarové vybavení robota NXT

1.2.1 NXT – G

Grafický programovací jazyk, který používá LEGO MINDSTORM NXT software a je dodáván jako součást sady společně se stavebnicí, se jmenuje NXT-G.

Jedná se o programovací prostředí, které vychází z profesionálního grafického programovacího nástroje LabVIEW od firmy National Instruments.

NXT-G je stejně jako LabVIEW založený na metodice drag and drop, neboli propojování a nastavení funkčních bloků, kde každý blok zastupuje nějakou určitou činnost jako např. řízení motoru.

Důraz je tedy kladen především na intuitivní a snadno a rychle pochopitelný přístup k programování robota.

1.2.1.1 Systémové požadavky

Windows

- Procesor Intel® Pentium® nebo jiný kompatibilní, min. 800 MHz
- Windows XP Professional nebo Home Edition obsahující Service Pack 2
- Minimálně 256MB RAM
- Až 300 MB volné paměti na harddisku
- XGA displej (1024 x 768)
- 1 USB port

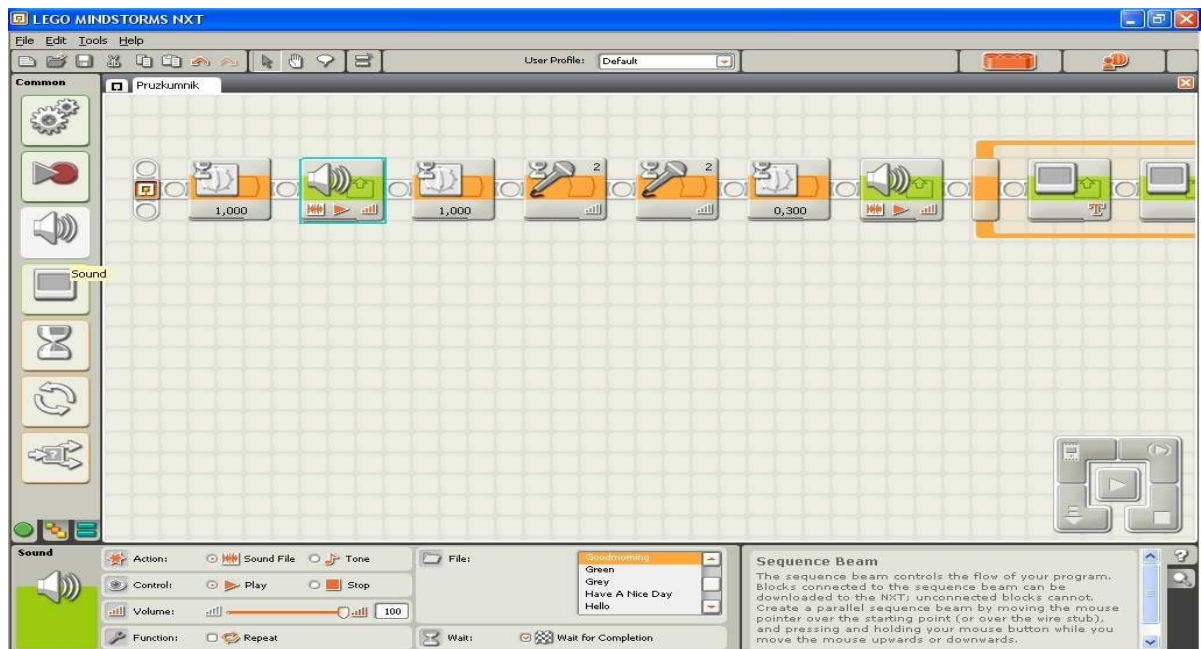
- CD-ROM
- Kompatibilní adaptér Bluetooth (volitelné)*

Macintosh

- PowerPC® G3, G4, G5 procesor, min. 600 MHz
- Apple MacOS X v. 10.3.9. nebo 10.4
- Minimálně 256MB RAM
- Až 300 MB volné paměti na harddisku
- XGA displej (1024 x 768)
- 1 USB port
- CD-ROM
- Kompatibilní adaptér Bluetooth (volitelné)*

1.2.1.2 Uživatelské rozhraní

Uživatelské rozhraní grafického programovacího jazyka NXT-G je velice přehledné a ovládání natolik intuitivní, aby ho po krátké době pochopilo i dítě školního věku. Náhled uživatelského rozhraní je zobrazen na Obr. 11. a popis jeho jednotlivých částí následuje hned za obrázkem.



Obr. 11. NXT - G

- Okno Robo centrum

Zde je k nalezení návod k sestavení a programování čtyř základních modelů.

- Můj portál

Během programování robotů poskytnete přístup k www.MINDSTORMS.com - získání více nápadů a tipů a stahování programů, zvukové efekty a další skvělé možnosti.

- Panel nástrojů

Panel nástrojů obsahuje nejčastěji užívané příkazy z programové nabídky.

- Pracovní plocha

Prostor na obrazovce, kde probíhá programování. Přesuňte programovací příkazy z palety programování na pracovní plochu a připojte je k sekvenčnímu svazku.

- Malé pomocné okno

Poskytnete pomoc, kdykoli je potřeba.

- Mapa pracovní plochy

Nástroj používaný k posouvání, aby se mohlo pohybovat po pracovní ploše. Pro přehled lze využít mapu pracovní plochy (tabulátor v pravém dolním rohu).

- Paleta programování

Paleta programování obsahuje všechny programové příkazy, které budete potřebovat pro vytvoření svých programů. Tabulátory ve spodní části palety vám umožní přepínání mezi obecnou paletou (obsahuje nejčastěji používané příkazy), úplnou paletou (obsahuje všechny příkazy) a upravenou paletou (obsahuje příkazy, které se dají stáhnout nebo si je uživatel může sám vytvořit).

- Konfigurační panel

Každý programovací příkaz má konfigurační panel, který vám umožní upravit tento příkaz z hlediska specifických vstupů a výstupů.

- Řídící jednotka

Pět tlačítek řídicí jednotky umožňujících stahovat programy (nebo jejich části) z počítače do NXT. S pomocí řídicí jednotky lze také změnit nastavení NXT.

- Okno NXT

Toto dialogové okno poskytuje informaci o paměti NXT a komunikačním nastavení.

1.2.2 Firmware NXT

Řídící kostka NXT má v sobě nainstalovaný firmware, který umožňuje mnohem víc, než jen spouštění programů vytvořených v programovacím prostředí NTX-G. Nabízí např. dokonce možnost naprogramovat si vlastní jednoduchý program jen za pomoci NXT. Jednotlivé funkce, které vám firmware NXT nabízí, jsou dostupné pomocí přehledného menu, které je vidět na Obr. 12.



Obr. 12. Ikony v menu NXT

V menu se pohybujete za pomoci čtyř tlačítek řídicí kostky NXT. Na Obr. 13 je znázorněn popis činností jednotlivých tlačítek. A za ním následuje popis jednotlivých ikonek menu a toho, co umožňují, získané z anglického manuálu dodávaného zároveň se stavebnicí.



Obr. 13. Ovládací tlačítka NXT

Moje soubory

Tímto odkazem se zobrazí všechny programy, které byly vytvořeny na NXT nebo se stáhly z počítače.

- Softwarové soubory – programy, které byly stáhnuty z počítače
- NXT soubory – programy, které jste vytvořily na NXT
- Zvukové soubory

Soubory se automaticky ukládají do příslušných složek. Pokud se do NXT stáhne program včetně zvukového souboru, bude program uložen do softwarových souborů, ale zvuková data budou uložena do zvukových souborů. Všechny soubory lze odesílat do ostatních NXT.

Program NXT

K programování robota není potřeba počítač. Pomocí programové nabídky NXT lze vytvořit tisíce různých programů bez počítače.

Vyzkoušej mě (Try me)

Nabídka „Vyzkoušej mě“ umožňuje zábavnou formou vyzkoušet senzory a motory. Je však důležité, aby senzory byly zapojeny do příslušných portů, které jsou uvedeny v manuálu k robotu.

Náhled (View)

Nabídka náhledu umožňuje provést rychlý test senzorů a motorů pozorováním skutečných dat z každé jednotky.

Připojí se senzor nebo motor, který bude potřeba otestovat, zvolí se příslušný port a data ze senzoru či motoru se objeví na displeji.

Nastavení (Setting)

Nastavení nabídky umožňuje upravit nastavení NXT včetně hlasitosti reproduktoru či nastavení klidového režimu. Tato nabídka také umožňuje smazat programy, které byly uloženy v paměti NXT.

- Klidový režim

NXT lze nastavit tak, aby po 2, 5, 10, 30 nebo 60 minutách přešla do klidového režimu, nečinnosti (doba, kdy není používána). Také lze zvolit „Never setting“, což znamená, že zůstane neustále zapnutá (dokud nebude vypnuta).

- Změň hlasitost

Zde se nastavuje hlasitost reproduktorů NXT.

- Vymaž všechny programy

Zde můžete vymazat všechny stažené programy ve všech třech dříve zmiňovaných složkách: softwarové soubory, NXT soubory, zvukové soubory.

Bluetooth

Nabídka Bluetooth slouží k bezdrátovému připojení NXT k dalším zařízením Bluetooth (ostatním NXT, mobilním telefonům a počítači). Bezdrátové připojení Bluetooth lze také využít k posílání programů do ostatních NXT, ke stažení programů z počítače bez použití kabelu USB a ke spoustě dalších skvělých možností, jako je využití mobilního telefonu k ovládání robota.

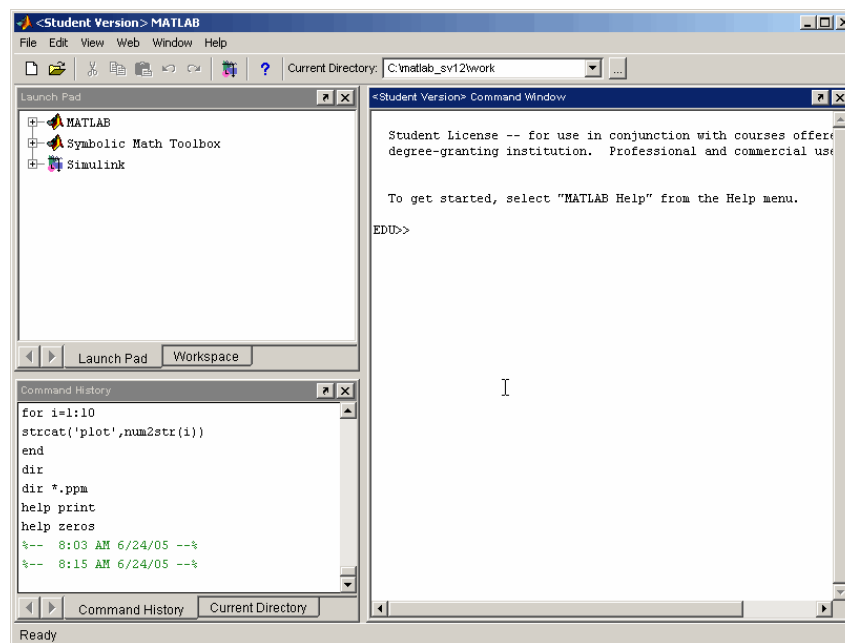
2 MOŽNOSTI PROGRAMOVÁNÍ A KOMUNIKACE NXT

2.1 Programování v prostředí MATLAB

Jednou z možností programování robota LEGO Mindstorms NXT je využití knihovny RWTH – Mindstorms NXT Toolbox vytvořené pro programový systém MATLAB. Další část této diplomové práce se proto věnuje popisu programového prostředí MATLAB a zmíněné knihovny.

2.1.1 MATLAB

MATLAB je integrovaný systém zahrnující nástroje pro symbolické a numerické výpočty, analýzu a vizualizaci dat, modelování a simulace dějů.[7]



Obr. 14. Uživatelské rozhraní programu MATLAB

MATLAB je programové prostředí a skriptovací programovací jazyk pro vědeckotechnické numerické výpočty, modelování, návrhy algoritmů, počítačové simulace, analýzu a prezentaci dat, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. Nástavbou Matlabu je Simulink – program pro simulaci a modelování dynamických systémů, který využívá algoritmy Matlabu pro numerické řešení především nelineárních diferenciálních rovnic.[8]

Název MATLAB vznikl zkrácením dvojice slov MATrix LABoratory (což se dá volně přeložit jako „laboratoř s maticemi“). To odpovídá skutečnosti, že klíčovou datovou strukturou při výpočtech v MATLABu jsou matice. Samotný programovací jazyk MATLABU vychází z jazyka Fortran.

2.1.2 RWTH Mindstorms NXT Toolbox

Otevřená architektura MATLABu vedla ke vzniku knihoven funkcí, nazývaných toolboxy, které rozšiřují použití programu v příslušných vědních a technických oborech. Tyto knihovny, navržené a v jazyce MATLABu napsané nejvýznačnějšími světovými odborníky, nabízejí předzpracované specializované funkce, které je možno rozšiřovat, modifikovat, anebo jen čerpat informace z přehledně dokumentovaných algoritmů.[9]

RWTH Mindstorms NXT Toolbox je knihovna vytvořená pro systémové prostředí MATLAB. Obsahuje funkce sloužící k navázání komunikace s robotem LEGO Mindstorms NXT a posléze k jeho kontrole a ovládní.

Funkce, které knihovna nabízí jsou rozříděné do následujících kategorií:

- Funkce zajišťující komunikaci s NXT
- Funkce pro práci se senzory
- Funkce pro práci s motory
- Funkce pro práci s ovládací kostkou NXT
- Funkce mapující moduly NXT
- Obecné funkce
- Ladící funkce

Seznamem všech funkcí a jejich popisem se zabývá praktická část této diplomové práce.

2.2 Programovací jazyk C, C++

Jazyk C pochází z počátku sedmdesátých let, navrhli jej Ken Thompson a Dennis Ritchie a jeho vznik je úzce spjat s Unixem. To do značné míry ovlivnilo i jeho vlastnosti, jedná se o poměrně nízkourovňový a jednoduchý procedurální jazyk určený pro skutečně praktické programování a ne pro kochání se čistotou syntaxe jazyka. C je velmi efektivní

a nevyžaduje náročnou runtime podporu jako např. Java. Céčko se překládá do strojového kódu. Nejprve se spustí preprocesor, který před vlastním překladem upraví zdroják, následuje překlad do objektového kódu a závěrečnou fází je linkování. Jednotlivé konstrukce jazyka jdou velmi jednoduše přeložit do assembleru, kterému se navíc podobá volným přístupem do paměti (ale samozřejmě nikoli syntaxí). Samotný jazyk je poměrně chudý, veškeré složitější věci se řeší voláním funkcí z knihoven. Součástí normy C je i standardní knihovna, která obsahuje alespoň ty nejčastěji používané funkce. Jde především o terminálový vstup a výstup, základní operace se soubory, práce s pamětí, řetězci a matematické funkce.[10]

Ukazatele jsou velmi mocným nástrojem, protože C jazyk povoluje ukazatele nejen na data, ale i na funkce. Současně jsou ukazatele z hlediska přenositelnosti a rizika zhroucení programu při jejich nesprávném použití Achillovou patou jazyka. Na druhou stranu, programátorovi je dána plná zodpovědnost za alokace paměti, není zde tedy závislost na automatickém dealokátoru paměti, jehož činnost nemá v jazycích vyšší úrovně možnost ovlivnit.[11]

Mnoho dalších současných programovacích jazyků přebírá syntaxi (způsob zápisu) z jazyka C. Patří mezi ně například Java či PHP.

2.2.1 Knihovna NXT++

NXT++ je rozhraní napsané v programovacím jazyku C++, které umožňuje kontrolu LEGO MINDSTORMS robotů přímo přes USB nebo Bluetooth připojení. Rozhraní je určeno k jednoduchému a snadnému použití. Tento kód je možné použít v libovolném programu využívajícím programovací jazyk C++.

Mnoho lidí má potíže s kompilací NXT++ při použití Microsoft Visual Studio. Na internetovém odkazu [12] je přehledný návod, s jehož pomocí se kompilace stává bezproblémovou záležitostí.

2.3 Programovací jazyk C#

C# je vysoce úroňový objektově orientovaný programovací jazyk vyvinutý Andersem Hejlsbergem ve společnosti Microsoft. Později byl přijat jako ECMA a ISO standard.

Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi).

C# lze využít k tvorbě webových aplikací a stránek, webových služeb, databázových programů, formulářových aplikací ve Windows, softwaru pro mobilní zařízení (např. mobilní telefony) apod.

Standard ECMA definuje současný design C# takto[13]:

- C# je jednoduchý, moderní, mnohaúčelový a objektivě orientovaný programovací jazyk.
- Jazyk a jeho implementace poskytuje podporu pro principy softwarového inženýrství, jako jsou: hlídání hranic polí, detekce použití neinicializovaných proměnných a automatický garbage collector. Důležité jsou také jeho vlastnosti jako: robustnost, trvanlivost a programátorská produktivita.
- Jazyk je vhodný pro vývoj softwarových komponent distribuovaných v různých prostředích.
- Přenositelnost zdrojového kódu je velmi důležitá, obzvláště pro ty programátory, kteří jsou obeznámeni s C a C++.
- Mezinárodní podpora je též velmi důležitá.
- C# je navržen pro psaní aplikací, jak pro zařízení se sofistikovanými operačními systémy, tak pro zařízení s omezenými možnostmi.
- Přestože by programy psané v C# neměly plýtvat s přiděleným procesorovým časem a pamětí, nemohou se měřit s aplikacemi psanými v C nebo assembleru.

2.3.1 Knihovna LEGO NXT.NET

Jak už bylo dříve zmíněno, aplikace pro robota LEGO NXT Mindstorm lze vyvíjet i za použití technologie .NET Framework od společnosti Microsoft. Pro vývoj těchto aplikací lze použít knihovnu LEGO NXT.NET, která je k dispozici ke stažení na webu s opensource projekty pro vývojová prostředí Microsoft na odkaze[14].

V archivu, který lze nalézt na výše uvedené stránce, jsou celkem dvě knihovny:

- Knihovna NxtNet.DesktopLib.dll, která umožní ovládat robota přes desktopové počítače.
- Knihovna NxtNet.MobileLib.dll, která umožní ovládat robota přes kapesní počítače a smartphony na platformě Windows Mobile

Součástí obou knihoven je i podrobná dokumentace. Vzhledem k tomu, že ukázková aplikace byla vyvíjena pro platformu Desktop PC, budou následující kapitoly věnovány pouze knihovně LEGO NXT.NET pro tuto platformu.

Požadavky na softwarové a hardwarové vybavení

Před samotným popisem knihovny LEGO NXT.NET je vhodné věnovat krátkou kapitolu požadavkům na softwarové a hardwarové vybavení počítače, na kterém budou vyvíjeny aplikace za pomoci této knihovny.

Z hlediska hardwarového vybavení je situace jednoduchá. Aplikace lze vyvíjet na libovolném osobním počítači, na kterém lze provozovat MS .NET viz. [15]. Oproti těmto požadavkům je zde však jedna specifická podmínka navíc. Počítač musí být vybaven Bluetooth modulem. Tato knihovna totiž umožňuje komunikaci s robotem pouze přes Bluetooth rozhraní.

Druhou částí této kapitoly jsou softwarové požadavky. Ty lze shrnout do tří bodů:

- Operační systém MS Windows XP a vyšší
- Knihovna MS .NET Framework 3.0 a vyšší
- Vývojové prostředí MS Visual Studio 2005 nebo 2008 (možno použít i ve verzi express)

2.4 Java

Java je objektově orientovaný jazyk, který vychází z jazyků C a C++ programovací jazyk určený k vývoji aplikací. Programovací jazyk Java, je ale jen jedna část prostředí Javy. To architektura Javy, umožňuje platformovou nezávislost a mnoho dalších výhod.

Celá architektura Javy:

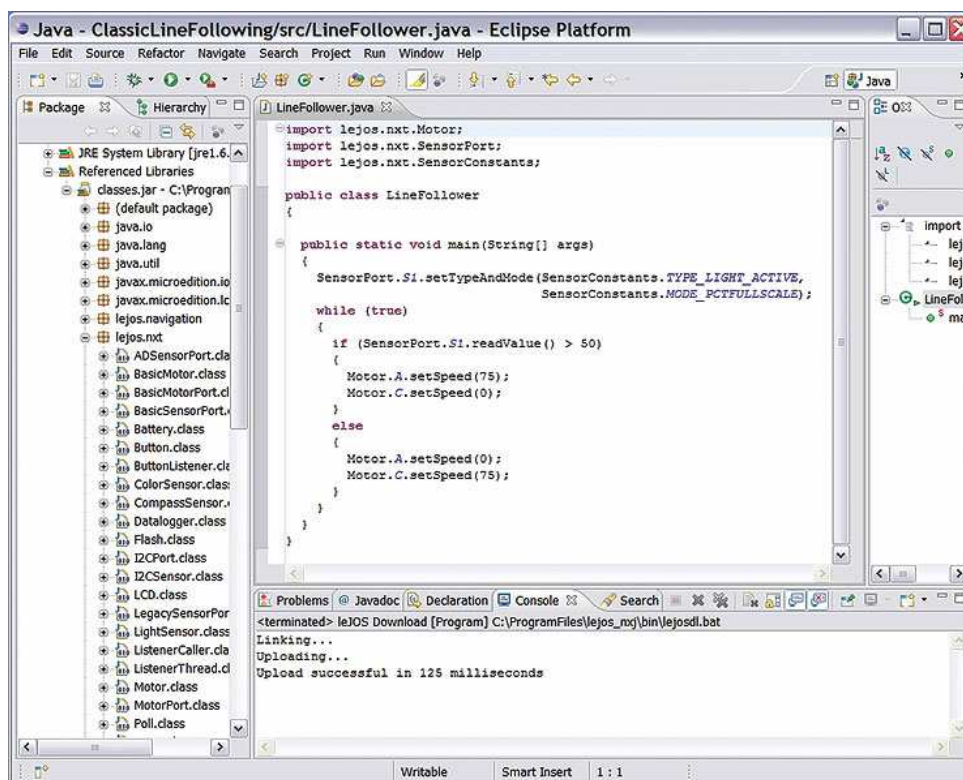
- programovací jazyk Java
- struktura souborů typu .class

- aplikační programové rozhraní (Java CoreAPI)
- abstraktní počítač - virtuální stroj jazyka Java (JVM)

Největším přínosem Javy je bezesporu plná přenositelnost programů na libovolnou platformu bez nutnosti jejich rekompilace. Kód je psán v programovacím jazyce Java. Ten je poté překládán do souborů typu .class. Tyto soubory jsou spouštěny v prostředí virtuálního stroje jazyka Java. Kombinace virtuálního stroje s třídami výkonného jádra jazyka Java je známa jako prostředí pro zpracování jazyka Java (JRE). Toto prostředí lze použít v jakémkoliv operačním systému. API je předem připravený kód, který je uspořádán do tematicky jednotných balíčků.[16][17]

2.4.1 LeJOS

Další možností programování robota Mindstorms NXT je pomocí programovacího jazyka Java. To umožňuje projekt LeJOS, který nabízí knihovnu s funkcemi sloužícími ke komunikaci a ovládání robota NXT do zmíněného jazyka. Zároveň poskytuje nový firmware pro řídicí kostku NXT.



Obr. 15. Ukázka použití funkcí z knihovny LeJOS

Balíčky tříd určených pro Mindstorms NXT:

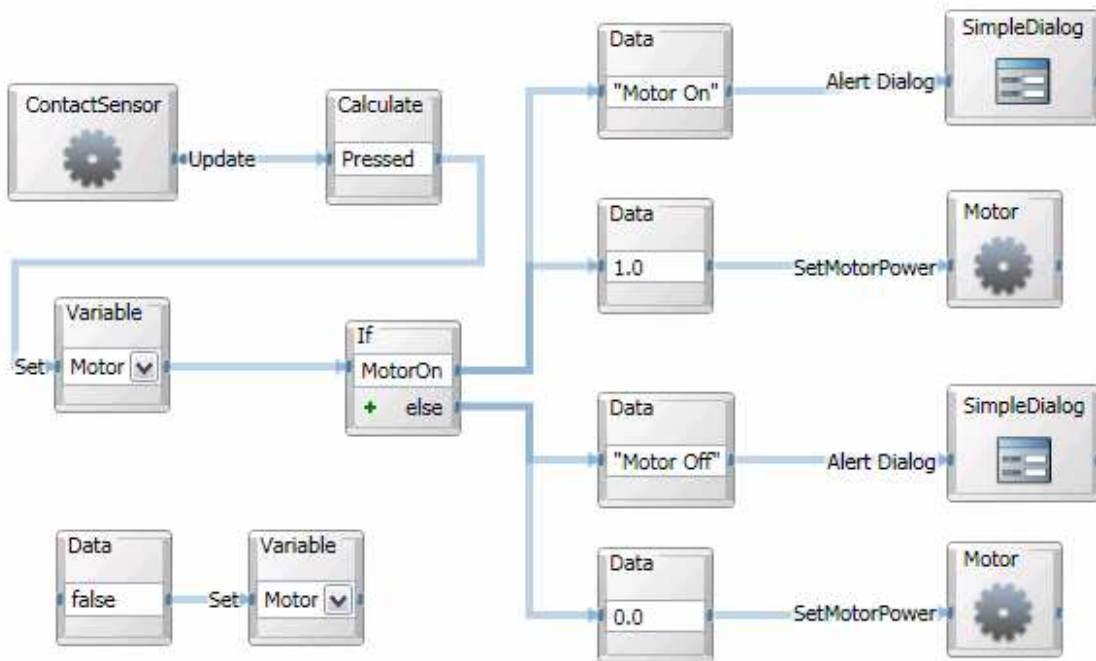
- **lejos.nxt** Zpřístupnění NXT senzorů, motorů, apod.
- **lejos.nxt.comm** Třídy zajišťující komunikaci NXT
- **lejos.nxt.debug** Ladící funkce
- **lejos.nxt.remote** Ovládání NXT pomocí Bluetooth technologie

2.5 Microsoft Robotic Studio

Pokud chcete programovat roboty v .NETu, tak budete potřebovat Microsoft Robotics Studio, které je v současné době uvolňováno v CTP verzích. Vývoj je založen na Concurrency&Coordination runtime (zkratka "CCR"), což je knihovna vyvinutá v Microsoft Research. Tato knihovna řeší u robotů koordinaci mezi jednotlivými aktivitami, které robot může dělat (program je tedy co nejvíce asynchroní). Knihovna je založená na portech (třída Port), pod kterými si lze představit objekt, který může obsahovat frontu zpráv, nebo dat nějakého typu. V případě robotů jsou porty všechny senzory a motory (ze senzorů bude vaše aplikace data číst a do portů pro motory zapisovat).[18]

2.5.1 Visual Programming Language

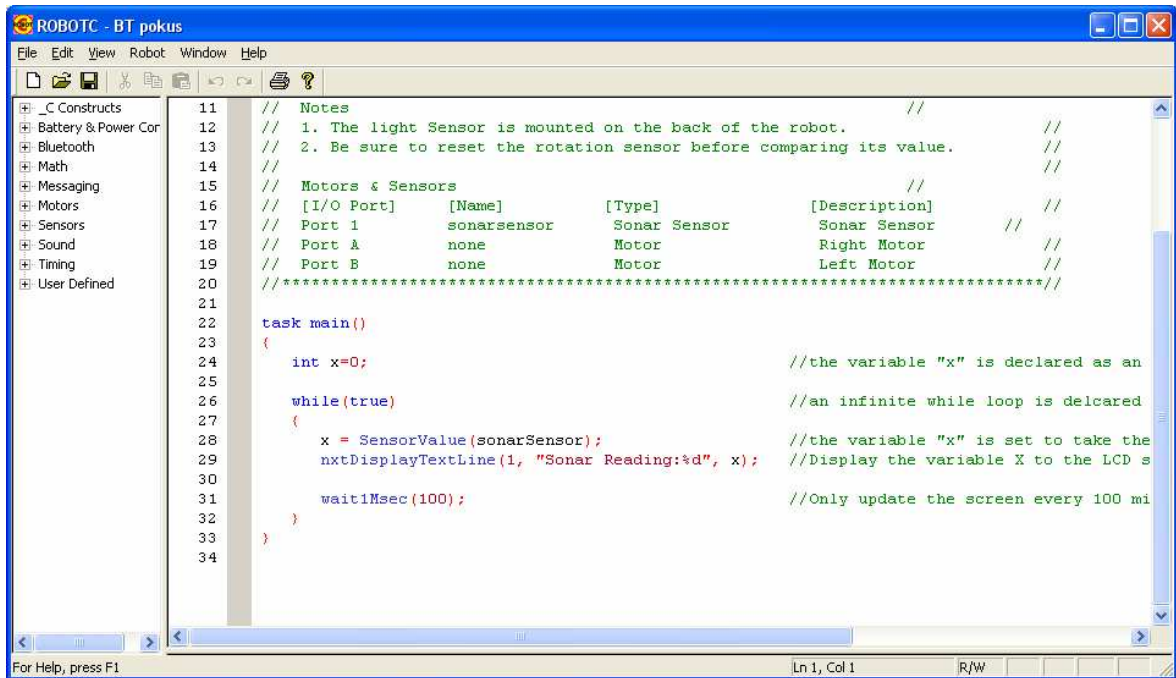
Microsoft Robotics Studio obsahuje nástroj pojmenovaný Visual Programming Language (tedy zkratkou "VPL"). Tento nástroj slouží podobně jako software od LEGA ke grafickému programování, ale narozdíl od LEGA je založený na výše popsaném a celkem propracovaném frameworku. Do VPL lze snadno přidávat vlastní komponenty (říká se jim služby), napsané třeba v C# a ty lze pospojovat do složitějších programů.[18]



Obr. 16. Příklad programu v nástroji Visual Programming Language

2.6 RobotC

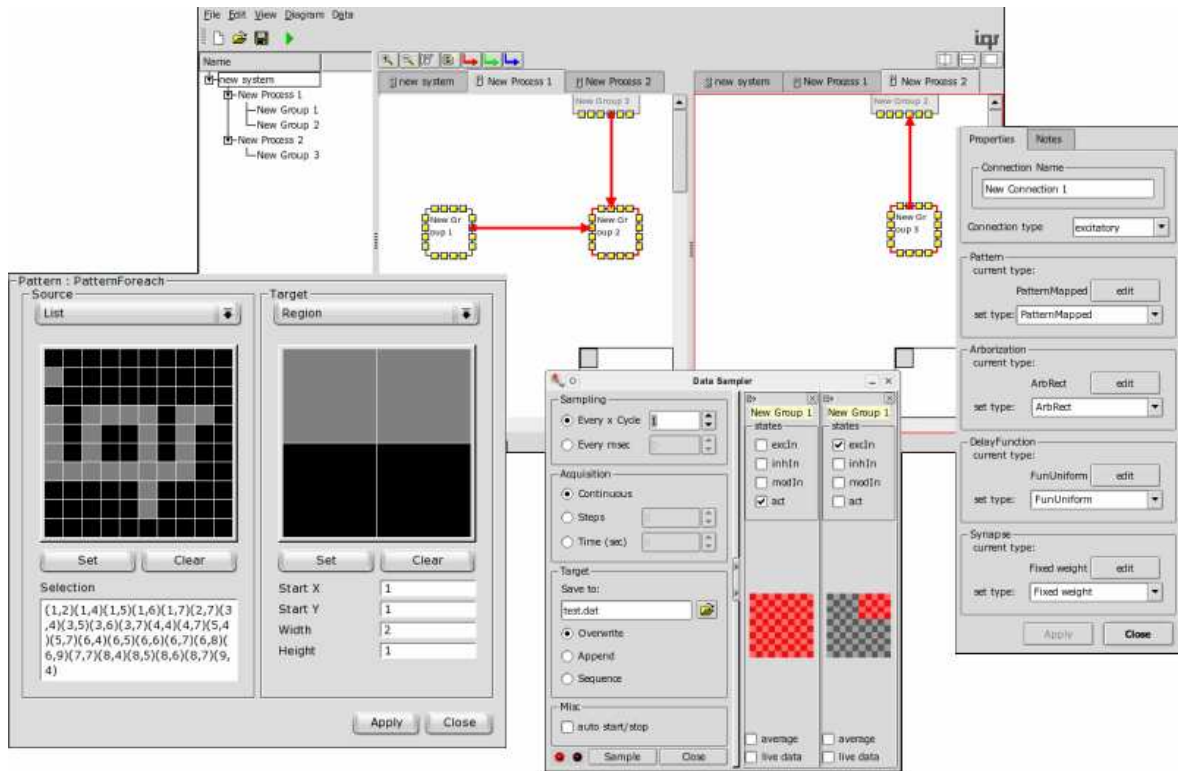
ROBOTC v kombinaci se systémem LEGO Mindstorms je výborný nástroj pro výuku začínajících a pokročilých programátorů na středních a vysokých školách. Jedná se o programovací jazyk založený na syntaxi jazyka C, který je určen k vytváření programů pro roboty VEX a NXT. Je určen pro obory zabývajících se programováním, strojírenstvím nebo například mechatrikou. Náhled uživatelského prostředí je k dispozici na Obr.17.



Obr. 17. Ukázka softwaru RobotC

2.7 Iqr

Iqr je simulační systém, který umí grafické navrhování a ovládání velkých neuronových sítí. Taktéž vám umožní modifikovat parametry sítě za již běžící simulace. Výhodou je rychlost simulací, která je postačující na ovládání robotů v reálném čase. A tak poslední verze implementovala modul, kde můžete pomocí výstupu na vybraných neuronech ovládat externí procesy. Dostupných je několik modulů, včetně modulu pro ovládání Lego MindStorms NXT.[19]



Obr. 18. Simulační systém Igr

2.8 Možnosti komunikace

LEGO Mindstorms NXT nabízí dva způsoby komunikace. Prvním je připojení pomocí kabelu a druhým je využití technologie Bluetooth.

2.8.1 Komunikace pomocí USB kabelu

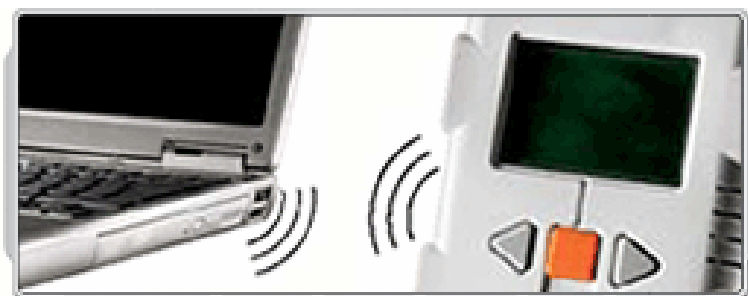
Jednou z možností, jak zajistit komunikaci mezi osobním PC a řídicí kostkou NXT, která ovládá robota, je použití připojovacího USB kabelu typu A-B. Tento kabel je součástí sady LEGO Mindstorms NXT a využívá se k nahrání vytvořeného programu v programovacím jazyce NXT-G do řídicí kostky NXT. Lze jej však využít i k přímému ovládní robota pomocí aplikace spuštěné v osobním počítači. Tento způsob připojení je vhodný především pro statické modely, jakým je například robotická ruka, kterou se zabývá tato diplomová práce.



Obr. 19. NXT – komunikace pomocí USB

2.8.2 Komunikace pomocí Bluetooth

Asi jedním z nejvýraznějších rysů systému Mindstorms NXT je jeho schopnost ovládat robota a komunikovat s ním prostřednictvím Bluetooth technologie. Uvnitř řídicí kostky NXT je CSR Blue Core™ 4 čip, který tuto komunikaci zajišťuje. Tento čip umožňuje navázání spojení se třemi zařízeními současně.



Obr. 20. NXT – Bluetooth komunikace

Bluetooth systém na NXT je určen pro provoz v konfiguraci Master/Slave. Každé zařízení NXT lze spojit až se třemi dalšími. Nicméně komunikovat může jen s jedním zařízením současně. Musí tedy ukončit zasílání nebo přijímání dat od jednoho zařízení, než začne to samé vykonávat s NXT zařízením druhým nebo třetím.

Dále můžeme této technologii využít ke komunikaci s PC pomocí Bluetooth adaptérů. Oficiální firmware od společnosti LEGO bohužel zajišťuje připojení jen s kompatibilními zařízeními (viz. Tab.1), které má málokdo k dispozici. Naštěstí, ale existuje řada neoficiálních verzí firmware, většinou k dispozici zároveň s knihovny do různých programovacích jazyků, které umožňují komunikaci s libovolným Bluetooth adaptérem.

Bluetooth zařízení	Kompatibilita
Abe UB22S	+++
Belkin F8T003 ver. 2 (short range)	+++
BlueFRITZ! AVM BT adapter, BlueFRITZ! USB v2.0	+++
Cables Unlimited USB-1520	+++
Dell TrueMobile Bluetooth Module	+++
Dell Wireless 350 Bluetooth Internal Card	---
Dlink DBT-120	+++
MSI Btoes	+++
MSI StartKey 3X-faster	+++
TDK GoBlue	+++
Qtrek, Bluetooth USB Adapter v2.0	+++

+++...kompatibilní

---...nekompatibilní

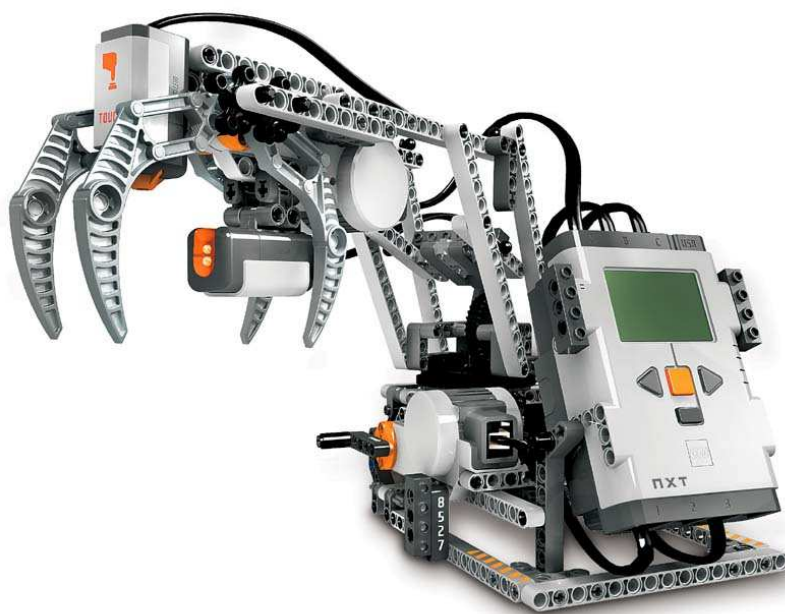
Tab. 1. Kompatibilita Bluetooth zařízení

Poslední možností využití Bluetooth komunikace je připojení mobilního telefonu podporujícího tuto technologii. Mobilní telefon pak lze použít jako dálkové ovládaní pro robota, případně se dá využít jako kamera pro použitého robota.

II. PRAKTICKÁ ČÁST

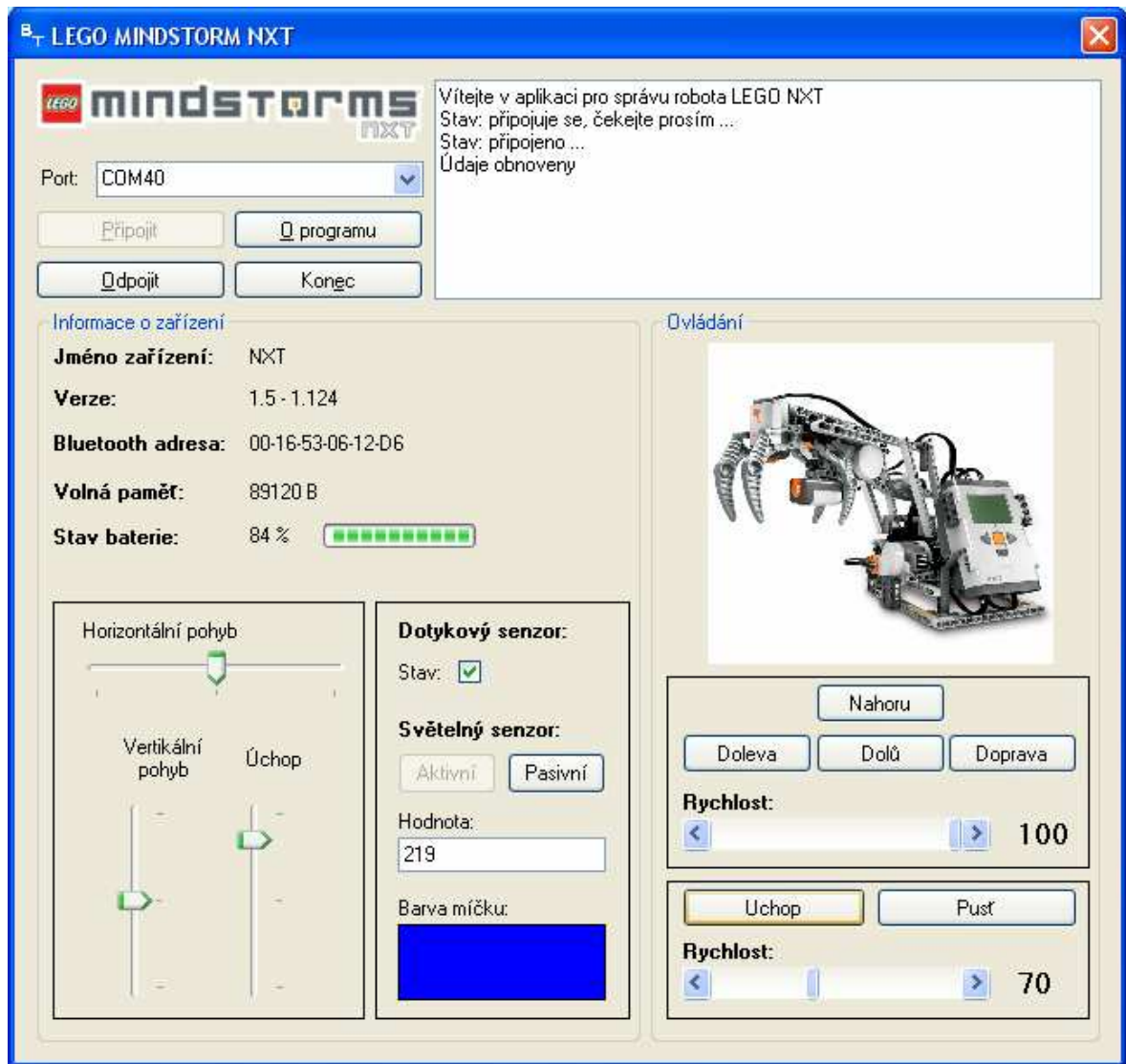
3 NÁVRH OVLADACÍHO PROGRAMU

Jedním z úkolů této diplomové práce, bylo vytvoření ovládacího programu pro robota Mindstorms NXT. Konkrétně pro konstrukční model *RoboArm*, který lze z této LEGO stavebnice postavit.



Obr. 21. LEGO Mindstorms NXT: Machine

Pro vývoj aplikace byla vybrána technologie .NET Framework a je napsána v jazyce C# za použití vývojového prostředí MS Visual C# Express 2008. Pro vytvoření aplikace byl zvolen tento programovací jazyk, neboť se jevil jako nejvhodnější. A to zejména ze dvou důvodů. Za prvé z důvodu existence velmi kvalitní knihovny do tohoto programu podporujícího ovládání robota Mindstorms NXT. Druhým důvodem pak byla možnost vytvoření aplikace jako samostatně spustitelného souboru, nezávislého na softwarovém vybavení uživatele dané aplikace. Proto nebyla aplikace vytvořena např. v programovém prostředí MATLAB, přestože i do tohoto programu existuje volně dostupná kvalitní knihovna. Nicméně této knihovny bylo přece jen využito. A to v další části práce, při vytváření návrhu laboratorních úloh. Náhled výsledné aplikace je na Obr. 22.



Obr. 22. Náhled vytvořené aplikace

Nyní k samotné aplikaci a k tomu, co by aplikace měla splňovat. Tento robot využívá ke svému pohybu trojici motorů. První motor zajišťuje horizontální pohyb ramene. Druhý motor vertikální pohyb a poslední motor slouží k ovládní čelistí na konci ramene, proto základní funkcí této aplikace je ovládní těchto motorů.

Dále pak robot využívá dva senzory. A sice dotykový senzor a světelný senzor. Aplikace proto načítá data z těchto senzorů a dále je vyhodnocuje. Posledním úkolem je zajistit komunikaci.

Výsledný program tedy umožňuje propojení počítače s robotem přes rozhraní Bluetooth v režimu sériový port (je nutné vybrat port – např. COM3). Následně je zobrazen aktuální stav robota a jeho senzorů a motorů. Přes rozhraní programu lze robota i ovládat.

3.1 Stručný popis funkcí knihovny NXT.NET

Knihovna NXT.NET je, stejně jako celá technologie .NET Framework, napsána v duchu objektového programování. Na úvod je tedy vhodné uvést základní třídy této knihovny. Ty shrnuje následující tabulka.

Třída	Popis
DeviceInfo	Struktura popisující kostku NXT.
Extensions	Metody pro převod datových typů.
LightSensor	Třída pro ovládání světelného senzoru.
MotorState	Struktura popisující aktuální stav motoru.
Nxt	Třída zajišťující komunikaci s NXT.
Sensor	Třída pro přístup k základním funkcím senzorů.
SensorEventArgs	Parametry událostí, které souvisí se senzory.
SensorState	Struktura popisující aktuální stav senzoru.
SoundSensor	Třída pro ovládání zvukového senzoru.
TouchSensor	Třída pro ovládání dotykového senzoru.
Version	Aktuální verze protokolu a firmware NXT.

Tab. 2. Základní třídy knihovny NXT.NET

Nejpoužívanější a zároveň nejdůležitější třídou je třída NXT. Tato třída totiž zajišťuje komunikaci s robotem. Proto následuje další tabulka, jejíž obsahem budou nejdůležitější metody této třídy.

Metoda	Popis
Connect	Připojení na daný port.
Disconnect	Odpojení ze všech připojených portů.
GetBatteryLevel	Reads the battery level of the NXT.
GetCurrentProgramName	Vrací jméno programu, který je aktuálně spuštěn na NXT.
GetDeviceInfo	Vrací informace o daném NXT zařízení.
GetInputValues	Vrací aktuální stav senzoru na daném portu.
GetOutputState	Vrací aktuální stav daného motoru.
GetVersion	Vrací aktuální verzi firmware a protokolu NXT.
KeepAlive	Vynuluje časovač přepnutí do úsporného režimu.
PlaySoundFile	Spustí přehrávání zvukového souboru na NXT.
PlayTone	Spustí přehrávání tónu o dané frekvenci, předem stanovenou dobu.
ResetInputScaledValue	Vynuluje hodnotu senzoru na daném portu.
ResetMotorPosition	Vynuluje aktuální pozici motoru.
SetBrickName	Nastaví jméno NXT kostky.
SetInputMode	Nastaví typ senzoru na daném portu.
SetOutputState	Odešle příkaz na daný motor.
StartProgram	Spustí daný program na NXT.
StopProgram	Zastaví aktuálně probíhající program.
StopSoundPlayback	Zastaví přehrávání zvukového souboru.

Tab. 3. NXT.NET - Funkce třídy NXT

Ve výše uvedené tabulce je pouze stručný přehled nejpoužívanějších metod třídy NXT, které je třeba znát pro pochopení kódu v ukázkové aplikaci. Kompletní popis atributů a metod jednotlivých tříd knihovny LEGO NXT.NET

3.2 Ukázka použití knihovny LEGO NXT.NET

Tato kapitola shrnuje teoretické informace z předchozích kapitol a demonstruje je na krátkých příkladech z aplikace, která byla vytvořena s použitím této knihovny. Rozhraní a používání této aplikace je popsáno v následující kapitole.

3.2.1 Připojení k NXT a zobrazení systémových informací

- 1) *Nxt mujrobot = new Nxt();*
- 2) *DeviceInfo infoozarizeni;*
- 3) *mujrobot.Connect("COM3");*
- 4) *infoozarizeni = mujrobot.GetDeviceInfo();*
- 5) *lblJmeno.Text = infoozarizeni.Name;*
- 6) *lblBluetooth.Text = infoozarizeni.BluetoothAddress.ToHexString();*
- 7) *lblPamet.Text = infoozarizeni.FreeUserFlash.ToString()+" B";*
- 8) *lblVerze.Text = mujrobot.GetVersion().Firmware + " - " + mujrobot.GetVersion().Protocol;*
- 9) *mujrobot.Disconnect();*

Před samotným připojením k zařízení je nejprve nutné nadefinovat objekt třídy Nxt, který bude sloužit k samotné komunikaci s robotem (1. řádek). Na druhém řádku je definována instance třídy DeviceInfo, ve které následně budou shromažďovány informace o zařízení. Ty jsou získány pomocí metody mujrobot.GetDeviceInfo (3) a následně vypsány do jednotlivých textových polí formuláře (5-7). Před odpojením od robota (9) jsou ještě získány informace o verzi zařízení. Toho je dosaženo metodou mujrobot.GetVersion (8).

3.2.2 Získání informací ze senzorů NXT

- 1) *String* x;
- 2) *Nxt mujrobot = new Nxt();*
- 3) *DeviceInfo infoozarizeni;*
- 4) *mujrobot.Connect("COM3");*
- 5) *x = mujrobot.GetInputValues(SensorPort.Port3).NormalizedValue.ToString();*
- 6) *mujrobot.Disconnect();*

V této ukázce je získána hodnota ze senzoru. Opět zde lze nalézt definici objektů a funkce pro připojení (řádky 2, 3, 4 a 6). Jediným rozdílem je zde 4. řádek, kde je získána hodnota ze senzoru na portu č. 3, čehož je dosaženo metodou *mujrobot.GetInputValues*.

3.2.3 Ovládání motorů

- 1) *Int32 rychlostuchop = 100;*
- 2) *Nxt mujrobot = new Nxt();*
- 3) *DeviceInfo infoozarizeni;*
- 4) *mujrobot.Connect("COM3");*
- 5) *mujrobot.SetOutputState(MotorPort.PortB, rychlostuchop, MotorModes.On, MotorRegulationMode.Speed, 0, MotorRunState.Running, 0);*
- 6) *mujrobot.Disconnect();*

Opět se jedná o podobnou situaci, jako v předchozích příkladech. Rozdílem je 5. řádek, kde se nachází metoda pro ovládání motoru. U této metody je vhodné se na chvíli zastavit a uvést stručný výčet jejích parametrů:

- *MotorPort motorPort* – port motoru (např. *MotorPort.PortB*),
- *SByte power* – rychlost otáčení motoru v intervalu - 100 až + 100,
- *MotorModes mode* – způsob pohybu motoru (*Coast* - volný pohyb, *On* - řízení pomocí PWM, *Brake* - brždění, *Regulated* - regulace),

- MotorRegulationMode regulation – nastavení regulace motoru (Idle – bez regulace, Speed – regulace rychlosti, Sync – synchronizace s druhým motorem),
- SByte turnRatio – synchronizace motorů v intervalu - 100 až + 100,
- MotorRunState runState – způsob pohybu motoru (Idle – bez napájení, Running – pohyb konstantní rychlostí, RampUp – zrychlení na danou hodnotu, RampDown – zpomalení na danou hodnotu),
- UInt32 tachoLimit – definice doby pohybu motoru pomocí úhlu otáčení.

4 NÁVRH LABORATORNÍCH ÚLOH

Pro návrh laboratorních úloh bylo testováno mnoho vhodných programovacích prostředí. Například bylo vytvořeno pár názorných programů v prostředí RobotC. Nakonec se ale tato část diplomové práce zaměřila pouze na programovací prostředí a jazyky vyučované v předmětech, které jsou součástí studia na Univerzitě Tomáše Bati ve Zlíně. Tyto úlohy využívají stavebnici LEGO Mindstorms NXT. Oživují tak výuku zvolených předmětů a snaží se je podat zábavnější formou.

Laboratorní úlohy byly vytvořeny pro předměty zabývající se výukou programovacích programů C, C#, C++ a programového prostředí MATLAB.

4.1 Laboratorní úlohy do programového systému MATLAB

Do programového systému MATLAB verze 7.0 a vyšší byly vytvořeny dvě úlohy, které si kladou za úkol zábavnou formou procvičit znalosti získané během studia týkajícího se tohoto prostředí. A to především znalosti o tvorbě vlastních aplikací, např. za pomoci nástroje GUI builder.

Pro úlohy byla využita open source knihovna RWHT – Mindstorms NXT Toolbox vytvořená na univerzitě RWTH Aachen.

4.1.1 Seznam funkcí knihovny RWHT

Seznam funkcí knihovny RWHT – Mindstorms NXT Toolbox tématicky rozdělen s přeloženým popisem.

NXT komunikace

- COM_CloseNXT Ukončí a smaže specifický NXT identifikátor nebo smaže všechny existující identifikátory.
- COM_CollectPacket Načítá data z USB nebo sériového/Bluetooth portu a načítá přesně jeden paket.
- COM_CreatePacket Vytváří ověřené pakety připravené k vysílání prostřednictvím Bluetooth .

- `COM_GetDefaultNXT` Navrací globální standardní NXT ukazatel, pokud byla hodnota v minulosti změněna.
- `COM_MakeBTConfigFile` Vytváří konfigurační soubor pro Bluetooth, jenž je požadován pro navázání komunikačního spojení.
- `COM_OpenNXT` Vytváří USB nebo Bluetooth spojení s NXT jednotkou a navrací ukazatel pro případné příští užití.
- `COM_OpenNXTEx` Vytváří Bluetooth nebo USB spojení s NXT jednotkou a navrací ukazatel pro případné příští užití.
- `COM_SendPacket` Posílá inicializační komunikační paket (bytové pole) přes USB nebo Bluetooth komunikační kanál (seriový port).
- `COM_SetDefaultNXT` Nastaví globální standardní NXT ukazatel jenž je využíván ostatními funkcemi, pokud byla hodnota minulosti změněna.

NXT senzory

- `CloseSensor` Uzavře požadovaný komunikační port senzoru (případně. vypne aktivní světelný mód NXT světelného senzoru).
- `GetLight` Načte aktuální hodnotu ze světelného senzoru NXT.
- `GetSound` Načte aktuální hodnotu ze zvukového senzoru NXT.
- `GetSwitch` Načte aktuální hodnotu z dotykového senzoru NXT.
- `GetUltrasonic` Načte aktuální hodnotu z ultrazvukového senzoru NXT.
- `OpenLight` Nastaví parametr režimu světelného senzoru NXT.
- `OpenSound` Nastaví parametr režimu zvukového senzoru NXT.
- `OpenSwitch` Nastaví parametr režimu dotykového senzoru NXT.
- `OpenUltrasonic` Inicializuje a nastaví parametr režimu světelného senzoru NXT.
- `SENSOR_1` Symbolická konstanta `SENSOR_1` (navrací 0).
- `SENSOR_2` Symbolická konstanta `SENSOR_2` (navrací 1).

- `SENSOR_3` Symbolická konstanta `SENSOR_3` (navrací 2).
- `SENSOR_4` Symbolická konstanta `SENSOR_4` (navrací 3).
- `USGetSnapshotResults` Přijímá až osm hodnot (vzdáleností) uchovaných v ultrazvukovém senzoru.
- `USMakeSnapshot` Ultrazvukový senzor vysílá jeden signál ("ping") a zaznamenává odezvu.

NXT motory

- `GetMemoryCount` Zajišťuje vnitřní NXT čítač paměti (manuální mapovací reprodukce).
- `GetMotor` Získává informace o aktuálním nastavení motoru prostřednictvím `SetMotor()`. Zobrazení chyby pokud nebyla hodnota motoru nastavena.
- `GetMotorSettings` Vrací aktuální data o stavu motoru / nastavení (e.g. pozice, rychlost, atd.) z určitého motoru.
- `MOTOR_A` Symbolická konstanta `MOTOR_A` (navrací 0)
- `MOTOR_B` Symbolická konstanta `MOTOR_B` (navrací 1)
- `MOTOR_C` Symbolická konstanta `MOTOR_C` (navrací 2)
- `MotorRotateAbs` Otáčky motoru ve stupních.
- `ResetMotorAngle` Nuluje čítač úhlu pro daný motor.
- `SendMotorSettings` Zasílá předešlé nastavení motoru současně používanému motoru.
- `SetAngleLimit` Nastaví krajní hodnoty úhlu pro port daného motoru.
- `SetMemoryCount` Nastaví vnitřní NXT paměťový čítač (manuální mapovací reprodukce).
- `SetMotor` Zapojí právě používaný motor do užívání prostřednictvím parametrických příkazů.

- SetPower Nastaví výkon aktuálně užívaného motoru.
- SetRampMode Nastaví hodnotu požadovanou pro zprovoznění motoru.
- SetTurnRatio Nastavení rychlosti otáčení motoru.
- SpeedRegulation Zapne/ vypne regulaci rychlosti otáček.
- StopMotor Zastaví/ zabrzdí daný motor.
- SwitchLamp Zhasne nebo rozsvítí kontrolku.
- SyncToMotor Zprovozní funkci synchronizace pro daný motor.
- WaitForMotor Vyčkávací pauza pro konečné zastavení motoru.

Řídící NXT funkce

- NXT_GetBatteryLevel Navrací aktuální stav baterií v milivoltech.
- NXT_GetFirmwareVersion Navrací verzi protokolu a firmwaru NXT.
- NXT_GetInputValues Provádí kompletní čtení hodnoty ze senzoru, i.e. požaduje vstupní hodnotu a navrací odpověď.
- NXT_GetOutputState Požaduje a navrací výstupní hodnotu stavu motoru.
- NXT_LSGetStatus Získá počet přístupných bytů z nízké záznamového senzoru.
- NXT_LSRead Reads Čtení dat z nízké záznamového senzoru.
- NXT_LSWrite Zápis získaných dat do nízké záznamového senzoru.
- NXT_PlaySoundFile Přehrává zvukový soubor získaný skrze NXT kostku.
- NXT_PlayTone Přehrává tón o dané frekvenci a délce.
- NXT_ReadIOMap Čte I/O mapu daného identifikátoru ID
- NXT_ResetInputScaledValue Nuluje měřítkové hodnoty senzoru, v závislosti na současně nastaveném módu (více NXT_SetInputMode)
- NXT_ResetMotorPosition Nuluje NXT vnitřní čítač pro daný motor, relativní i absolutní.

- `NXT_SendKeepAlive` Zasílá udržovací paket. Nastavitelně: Požadovaný časový interval.
- `NXT_SetBrickName` Nastaví název pro NXT kostku (Připojení k danému ukazateli).
- `NXT_SetInputMode` Nastaví mód, konfigurační a inicializační pro stav senzoru připraven k výstupu.
- `NXT_SetOutputState Sends` Nastaví poslední používané parametry aktuálnímu motoru.
- `NXT_StartProgram` Spustí program na NXT kostce.
- `NXT_StopProgram` Zastaví program na NXT kostce.
- `NXT_StopSoundPlayback` Zastaví přehrávání zvuku.
- `NXT_WriteIOMap` Zapiše I/O ID daného modulu.

NXT funkce mapování modulů

- `MAP_GetCommModule` Čte I/O mapu komunikačního modulu.
- `MAP_GetInputModule` Čte I/O mapu vstupního komunikačního modulu.
- `MAP_GetOutputModule` Čte I/O mapu výstupního komunikačního modulu.
- `MAP_GetSoundModule` Čte I/O mapu zvukového modulu.
- `MAP_GetUIModule` Čte I/O mapu modulu komunikačního rozhraní.
- `MAP_SetOutputModule` Zapisuje I/O mapu do výstupního modulu.

Generální funkce

- `checkStatusByte` Interpretuje hodnoty bytů návratového paketu a vytváří chybovou zprávu.
- `readFromIniFile` Čte parametry z konfiguračního souboru (usually *.ini)

- `textOut` Formát pro `fprintf()` jež může volitelně zapsat část výstupu do registračního souboru (logfile).
- `tictic` Podobný MATLAB-ovskému `tic()`, ale je rozšířen o možnost uložení více stavů.
- `toctoc` Podobný MATLAB-ovskému `toc()`, ale je rozšířen o možnost uložení více stavů.

Ladící funkce

- `DebugMode` Získá nebo nastaví ladící mód (i.e. Pokud se jedná o výstup, hodnoty budou zapsány do příkazového okna).

Funkce u panelu nástrojů RWTH - Mindstorms NXT, je možné rozdělit do několika konstrukčních vrstev. Na nejnižší vrstvě (Nízká úroveň a pomocné funkce) jsou k dispozici funkce, které umožňují převést parametry na hodnoty o velikosti 1B dle LEGO dokumentace NXT příkazů. Druhá vrstva obsahuje přímé NXT příkazy, které jsou mapovány v LEGO dokumentaci přímých příkazů bez jakéhokoli omezení a lze je identifikovat pomocí prefixu „NXT_“. V této vrstvě lze nalézt také balíček funkcí souvisejících s Bluetooth připojením. Třetí vrstva obsahuje funkce vysoké úrovně sloužící pro ovládání NXT motorů, senzorů a NXT připojení. Tyto funkce v podstatě využívají příkazů druhé vrstvy k tomu, aby bylo ovládání motorů a senzorů uživatelsky čitelnější a pohodlnější. Čtvrtá vrstva poskytuje funkce nejvyšší úrovně, sloužící pro přesné řízení regulace motorových otáček.

Vrstva	Popis	Výstup/motory	Vstup/senzory	Obecné	Bluetooth
4	High Level Regulation	MotorRotateAbs WaitForMotor			COM_MakeBTConfigFile
3	High Level Functions	SendMotorSettings SetMotor SetPower SetAngleLimit SetRampMode SpeedRegulation SyncToMotor StopMotor GetMotorSettings GetMotor ResetMotorAngle SetMemoryCount GetMemoryCount	OpenLight OpenSound OpenSwitch OpenUltrasonic GetLight GetSound GetSwitch GetUltrasonic CloseSensor	readFromIniFile MAP_GetCommModule MAP_GetInputModule MAP_GetOutputModule MAP_GetSoundModule MAP_GetUIModule MAP_SetOutputModule	COM_OpenNXT COM_OpenNXTEx COM_CloseNXT
2	Direct NXT Commands	NXT_SetOutputState NXT_GetOutputState NXT_ResetMotorPosition	NXT_SetInputMode NXT_GetInputValues NXT_ResetInputScaledValue NXT_LSRead NXT_LSWrite NXT_LSGetStatus	NXT_PlayTone NXT_PlaySoundFile NXT_StopSoundPlayback NXT_StartProgram NXT_StopProgram NXT_SendKeepAlive NXT_GetBatteryLevel NXT_GetFirmwareVersion NXT_SetBrickName NXT_ReadIOMap NXT_WriteIOMap	COM_CreatePacket COM_SendPacket COM_CollectPacket COM_SetDefaultNXT COM_GetDefaultNXT
1	Low Level Functions: Helper, Conversion and Lookup Functions	MOTOR_A MOTOR_B MOTOR_C <i>byte2outputmode</i> <i>byte2regmode</i> <i>byte2runstate</i> <i>outputmode2byte</i> <i>regmode2byte</i> <i>runstate2byte</i> <i>initializeGlobalMotorStateVar</i> <i>resetMotorRegulation</i>	SENSOR_1 SENSOR_2 SENSOR_3 SENSOR_4 <i>byte2sensortype</i> <i>byte2sensormode</i> <i>sensortype2byte</i> <i>sensormode2byte</i>	DebugMode <i>isdebug</i> textOut tictic (o) toctoc (o) <i>dec2wordbytes</i> <i>name2commandbytes</i> <i>commandbyte2name</i> <i>wordbytes2dec</i>	checkStatusByte <i>createHandleStruct</i> <i>checkHandleStruct</i> <i>getLibusbErrorString</i> <i>getVISAErrorString</i> <i>getReplyLengthFromCmdByte</i>

Obr. 23. Konstrukční vrstvení RWTH Mindstorms toolboxu

4.1.2 Úloha č. 1 - Získání a zobrazení údajů ze senzorů NXT

Cíl úlohy

Cílem této úlohy je zábavnou formou procvičit znalosti získané během studia týkajícího se prostředí MATLAB. A to především základy práce v tomto prostředí a vytváření jednoduchých aplikací pomocí GUI nástroje.

Zadání úlohy

- 1) Seznamte se s možnostmi a parametry programovací kostky NXT a jejími senzory.
- 2) Nastudujte možnosti knihovny RWTH - Mindstorms NXT Toolbox.

- 3) Připojte k PC pomocí USB kabelu ovládací kostku NXT se všemi připojenými senzory a motory.
- 4) Vyzkoušejte si v příkazovém okně vytvoření připojení s programovací kostkou NXT a načtení dat z libovolně vybraných senzorů.
- 5) Po seznámení se s prací s Mindstorms NXT Toolboxem vytvořte pomocí GUI builderu jednoduchou aplikaci, která bude po stisku tlačítka zobrazovat aktuální informace z připojených senzorů.
- 6) Doplňte aplikaci o pravidelné automatické načítání těchto dat.



Obr. 24. Úloha č.1 – Mindstorms NXT

Popis úlohy

- 1) Prostudujte teorii o hardwaru robota LEGO Mindstorms NXT, kterou naleznete v příložené diplomové práci.
- 2) Prostudujte dokumentaci k RWTH toolboxu, která je k dispozici na internetové adrese:

<http://svn.lfb.rwth-aachen.de/mindstorms/documents/downloads/doc/version-2.00beta/abc.html>

- 3) Připojte k programovací kostce NXT senzory dle Obrázku 2. Poté připojte kostku NXT přes USB rozhraní k vašemu PC.



Obr. 25. Úloha č.1 – NXT a senzory

- 4) Vyzkoušejte si v příkazovém okně základní příkazy nastudované v dokumentaci o RWTH Mindstorms NXT Toolboxu. Zaměřte se především na následující příkazy.

Syntaxe, kterou budete potřebovat:

- Otevření USB spojení mezi PC a robotem

```
handle = COM_OpenNXT()
```

```
COM_SetDefaultNXT(handle);
```

- Ukončení USB spojení mezi PC a robotem

```
COM_CloseNXT('all');
```

- Připojení jednotlivých senzorů

```
OpenLight(port, mode)
OpenSound(port, mode)
OpenSwitch(port)
OpenUltrasonic(port, mode)
```

- Načtení dat z jednotlivých senzorů

```
light = GetLight(port)
sound = GetSound(port)
switch = GetSwitch(port)
distance = GetUltrasonic(port)
data = GetMotorSettings(port)
```

- Uzavření portů jednotlivých senzorů

```
CloseSensor(port)
```

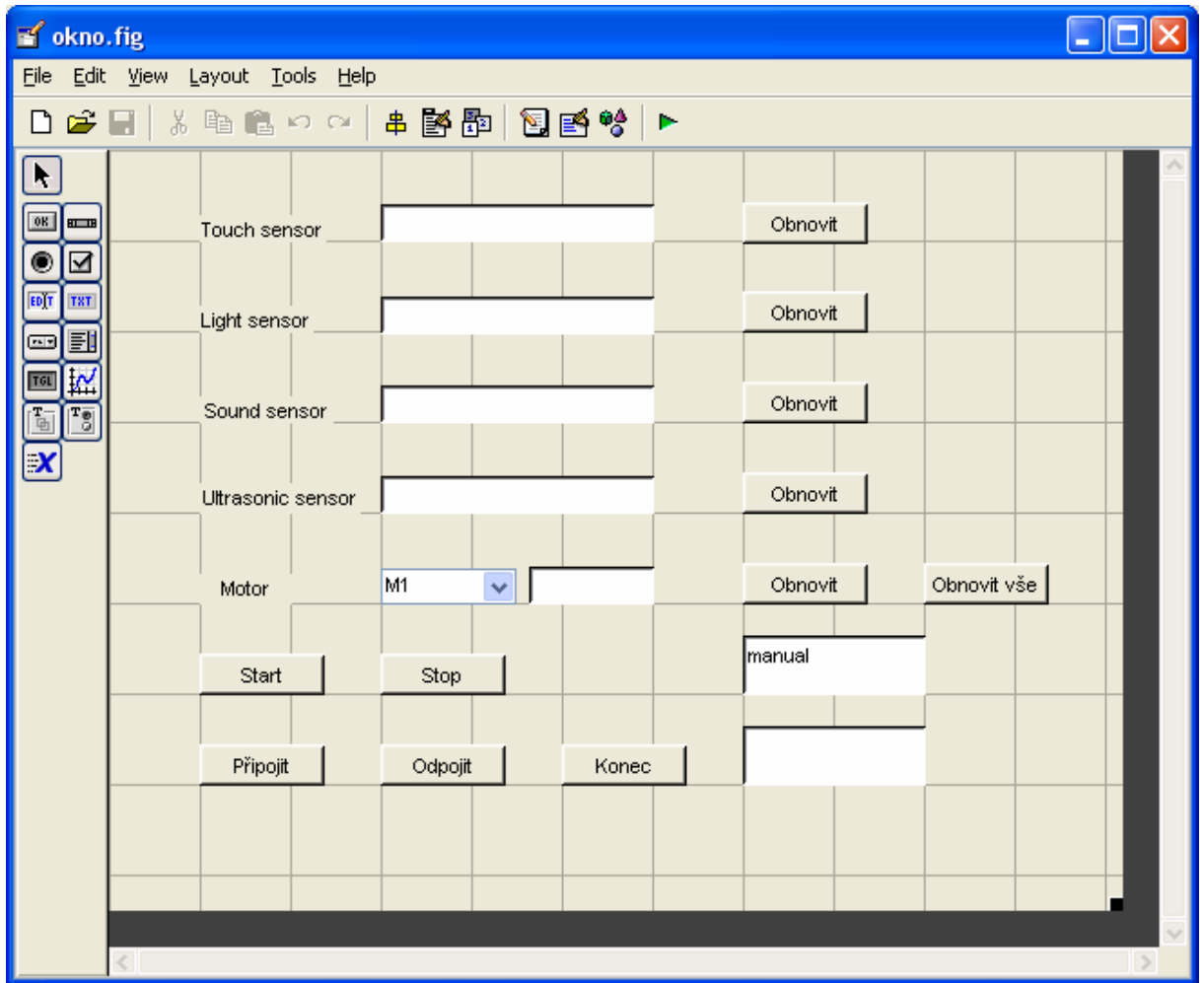
Příklad načtení aktuální hodnoty ze světelného senzoru v pasivním režimu:

```
OpenLight(SENSOR_1, 'INACTIVE');

light = GetLight(SENSOR_1);

CloseSensor(SENSOR_1);
```

- 5) Spust'te nástroj pro tvorbu uživatelského rozhraní GUIDE (Graphical User Interface Development Environment). Pro osvěžení paměti připomínám, že ho naleznete po stisku tlačítka Start->Matlab->GUI builder. V ní vytvořte podobné uživatelské rozhraní jako vidíte na Obr. 26. I když vaší fantazii a kreativnosti se samozřejmě meze nekladou. Důležité je pouze to, aby jste splnili funkci aplikace viz. zadání tohoto bodu úlohy.



Obr. 26. Úloha č.1 – Náhled vzorového uživatelského prostředí

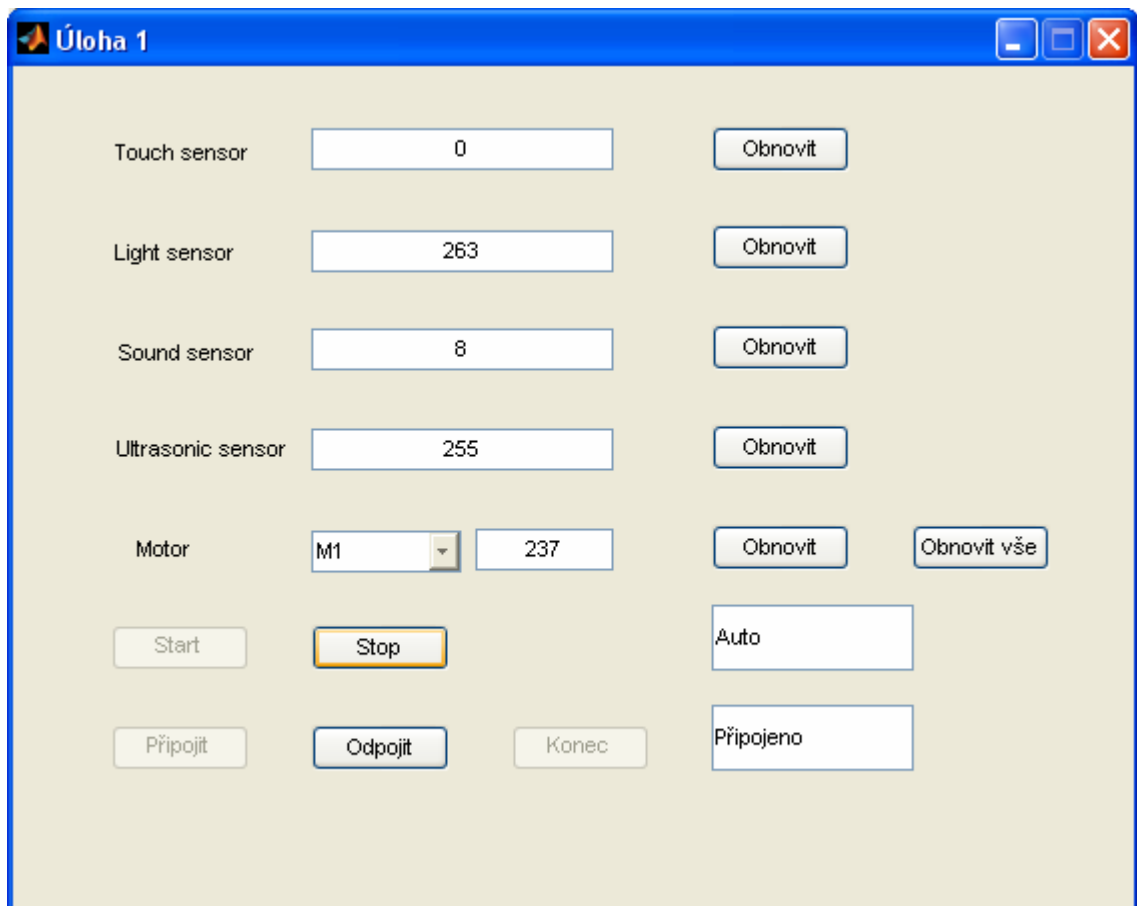
- 6) Doplňte tuto aplikaci o funkci časovače, která bude automaticky aktualizovat a zobrazovat data načtená ze senzorů. Využijte funkci *timer*.

Příklad:

```
t = timer('TimerFcn', @mycallback, 'Period', 10.0);
```

Závěr

Vzorovou úlohu naleznete přiloženou k tomuto zadání. Pokuste se však zadanou aplikaci vytvořit sami. Uvidíte, že programování v prostředí Matlab s využitím robota Mindstorms NXT je docela zábavné. Kdyby jste si však nevěděli s některou částí úlohy rady, tak využijte vzorovou aplikaci k porozumění vašeho problému.



Obr. 27. Úloha č.1 – Náhled vzorové aplikace

4.1.3 Úloha č. 2 - Ovládání robota Mindstorm NXT Machine

Cíl úlohy

Cílem této úlohy je procvičit a prohloubit znalosti získané během studia týkajícího se prostředí Matlab. A to především základy práce v tomto prostředí a vytváření jednoduchých aplikací pomocí GUI nástroje.

Zadání úlohy

- 1) Seznamte se s možnostmi a parametry programovací kostky NXT a jejími senzory.
- 2) Nastudujte možnosti knihovny RWTH - Mindstorms NXT Toolbox.
- 3) Připojte k PC pomocí USB kabelu robota Mindstorms NXT – Machine.
- 4) Vyzkoušejte si v příkazovém okně vytvoření připojení s programovací kostkou NXT a načtení dat z libovolně vybraných senzorů.
- 5) Po seznámení se s prací s Mindstorms NXT Toolboxem vytvořte pomocí GUI builderu jednoduchou aplikaci, která umožní ovládat motory robota Mindstorms NXT Machine.



Obr. 28. Úloha č.2 – Mindstorms NXT

Popis úlohy

- 1) Nastudujte si teorii o hardwaru robota LEGO Mindstorms NXT, která se nachází v příložené diplomové práci.
- 2) Nastudujte si dokumentaci k RWTH toolboxu, která se nachází na internetové adrese:

<http://svn.lfb.rwth-aachen.de/mindstorms/documents/downloads/doc/version-2.00beta/abc.html>

- 3) Připojte přes USB rozhraní k vašemu PC robota LEGO Mindstorms NXT poskládaného jako robotickou ruku (*Machine*) viz Obr. 29.



Obr. 29. Úloha č.2 – NXT Machine

- 4) Vyzkoušejte si v příkazovém okně základní příkazy nastudované v dokumentaci o RWTH Mindstorms NXT Toolboxu. Zaměřte se především na následující příkazy.

Syntaxe, kterou budete potřebovat:

- Otevření USB spojení mezi PC a robotem

```
handle = COM_OpenNXT()  
COM_SetDefaultNXT(handle);
```

- Ukončení USB spojení mezi PC a robotem

```
COM_CloseNXT('all');
```

- Volba motoru, resp. Portu, ke kterému je robot připojený, se kterým chceme dále pracovat

```
SetMotor(port);
```


- Volba rychlosti a směru otáčení motoru (-100,100)

SetPower(power);

- Úhel ve stupních požadovaného natočení motoru

SetAngleLimit(limt);

Příklad ovládání motoru robota připojeného do portu A:

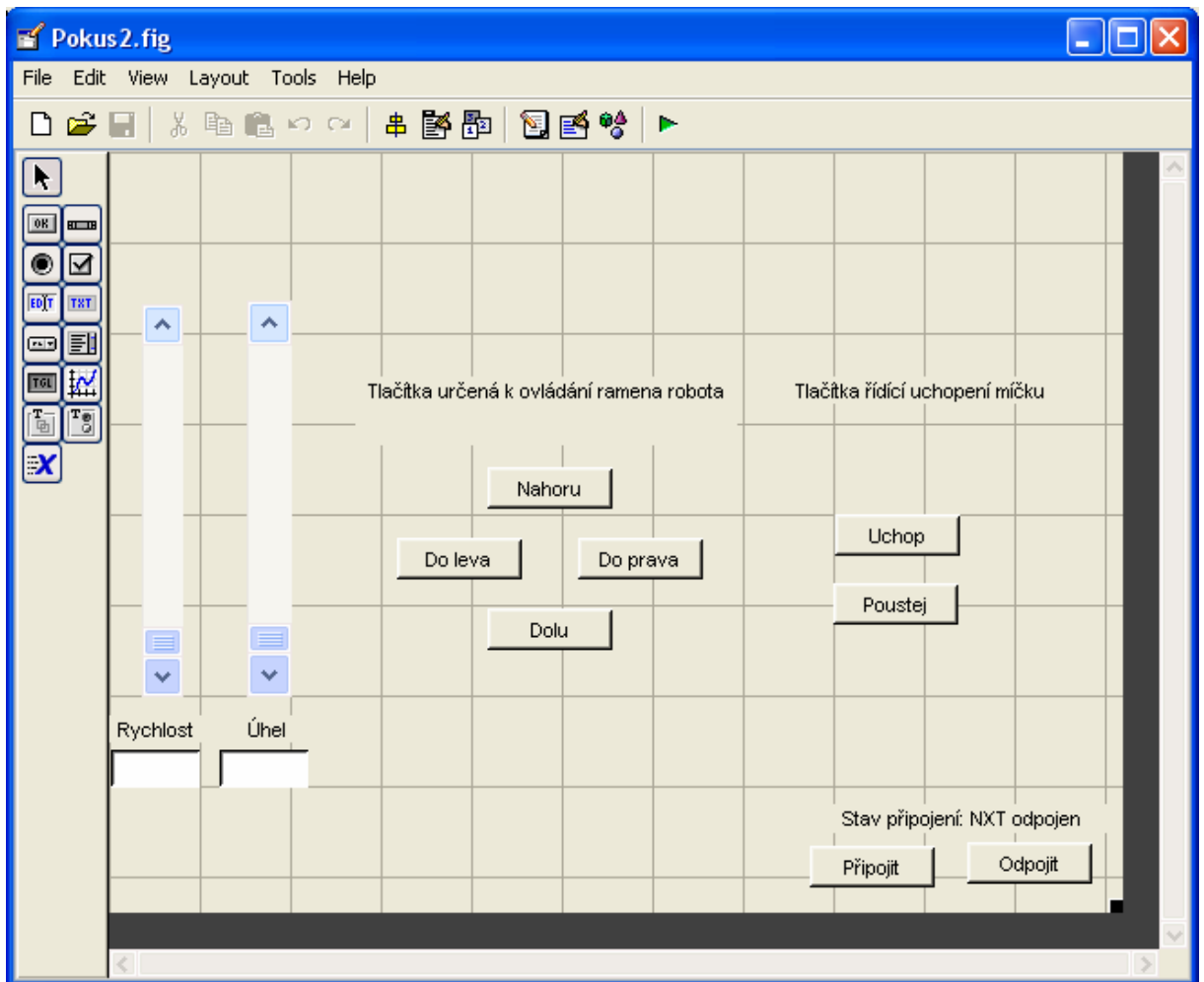
SetMotor(MOTOR_A);

SetPower(100);

SetAngleLimit(720);

SendMotorSettings();

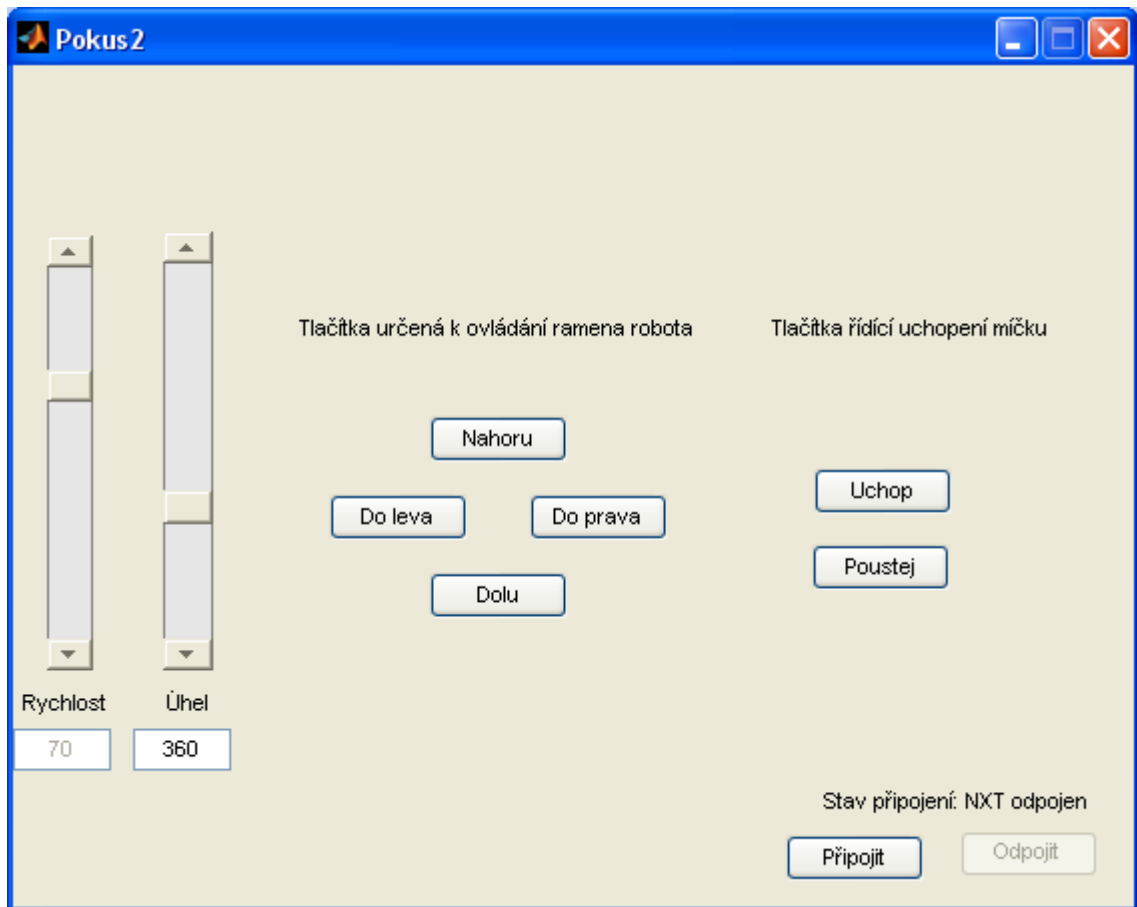
- 5) Spusťte nástroj pro tvorbu uživatelského rozhraní GUIDE (Graphical User Interface Development Environment). Pro osvěžení paměti připomínám, že ho naleznete po stisku tlačítka Start->Matlab->GUI builder. V ní vytvořte podobné uživatelské rozhraní jako vidíte na Obr. 30. I když vaší fantazii a kreativitě se samozřejmě meze nekladou. Důležité je pouze to, aby jste splnili funkci aplikace viz. zadání tohoto bodu úlohy.



Obr. 30. Úloha č.2 – Náhled vzorového uživatelského prostředí

Závěr

Pokuste se vytvořit zadanou aplikaci sami a bez cizí pomoci. Zjistíte, že programování v prostředí Matlab, s využitím robota Mindstorms NXT, se stane docela zábavným. Pokud si nebudete s některou částí úlohy vědět rady, jako nápovědu využijte přiloženou vzorovou aplikaci, s jejíž pomocí se pokuste váš problém vyřešit.



Obr. 31. Úloha č.2 – Náhled vzorové aplikace

4.2 Laboratorní úloha do výuky programovacího jazyka C

Pro výuku zabývající se programovacím jazykem C byly vytvořeny tři úlohy. První úloha, velice snadná, má za úkol jen vyzkoušet připojení programovací kostky NXT a vyzkoušení pár základních příkazů a využití podmínek a cyklů v programovacím jazyku C. Druhá úloha pak navíc procvičuje kontrolu stisku vybraných kláves, které mají za úkol ovládnutí motorů robota. A konečně třetí úloha, která má za úkol kompletní načtení dat ze senzorů robota, a jejich ukládání do souboru a řízení motorů robota.

4.2.1 Popis funkcí knihovny NXT++

- **Comm**

Pomocné funkce NXT++ pro komunikaci. Tyto funkce většinou běžní uživatelé nepoužijí.

```
void SearchBT(std::vector< std::string > *names)
```

- **NXT**

Obsahuje všechny funkce, které pracují s řídicí kostkou NXT

bool *Open* (*Comm::NXTComm *comm*)

Zahajuje NXT připojení.

bool *OpenBT* (*Comm::NXTComm *comm*)

Zahajuje NXT připojení prostřednictvím technologie Bluetooth.

void *Close* (*Comm::NXTComm *comm*)

Ukončuje NXT připojení.

void *PlayTone* (*Comm::NXTComm *comm, int frequency, int duration*)

Přehraje tón.

int *BatteryLevel* (*Comm::NXTComm *comm*)

Vrací aktuální stav baterie.

std::string *GetName* (*Comm::NXTComm *comm*)

Zjistí jméno NXT.

double *GetAvailableFlash* (*Comm::NXTComm *comm*)

Zjistí aktuální velikost volné flash paměti NXT.

void *StopProgram* (*Comm::NXTComm *comm*)

Zastaví právě probíhající program, pokud je spuštěn.

double *GetProtocolVersion* (*Comm::NXTComm *comm*)

Zjistí verzi protokolu daného NXT

double *GetFirmwareVersion* (*Comm::NXTComm *comm*)

Zjistí verzi firmware daného NXT.

void *StartProgram* (*Comm::NXTComm *comm, std::string name*)

Spustí na NXT program s konkrétním jménem.

void *PlaySoundFile* (*Comm::NXTComm *comm, std::string name, bool loop*)

Přehraje na NXT zvukový soubor s konkrétním jménem

*void StopSound (Comm::NXTComm *comm)*

Zastaví aktuálně přehrávaný zvuk.

*void KeepAlive (Comm::NXTComm *comm)*

Vynuluje vnitřní časovač automatického vypnutí.

*void SendDirectCommand (Comm::NXTComm *comm, ViBoolean response, ViByte *dc_buf, int dc_buf_size, ViByte *re_buf, int re_buf_size)*

Umožní zadat přímý příkaz k NXT.

- **NXT:: Soubor**

Souborové funkce NXT

*int getNextFile (Comm::NXTComm *comm, std::string filename, NXTFile &file)*

Získá konkrétní NXT souborový ukazatel podle jména.

*int getNextFilesByName (Comm::NXTComm *comm, std::string pattern, NXTFileIterator &iter)*

Získá NXT souborový ukazatel podle jména.

*int getNextFilesByType (Comm::NXTComm *comm, int fileType, NXTFileIterator &iter)*

Získá NXT souborový ukazatel podle typu.

*void destroyFileIterator (Comm::NXTComm *comm, NXTFileIterator &iter)*

Zruší souborový iterátor.

*void destroyFile (Comm::NXTComm *comm, NXTFile &file)*

Zruší souborový ukazatel.

- **NXT:: Modul**

Modulové funkce NXT

*int getNextModules (Comm::NXTComm *comm, std::string pattern, NXTModuleIterator &iter)*

Navrací ukazatel na NXT modul dle jména.

*int getNXTModule (Comm::NXTComm *comm, std::string pattern, int moduleID, int moduleSize, int ioMapSize, int &status, NXTModule &module)*

Navrací ukazatel na konkrétní NXT modul dle jména.

- **NXT:: Motor**

Obsahuje všechny funkce, týkající se motorů

*int GetRotationCount (Comm::NXTComm *comm, int port)*

Získává informace o otáčkách motoru.

*void ResetRotationCount (Comm::NXTComm *comm, int port, bool relative)*

Vynuluje počítadlo otáček u konkrétního motoru.

*void SetForward (Comm::NXTComm *comm, int port, int power)*

Nastavuje pohyb motoru vpřed konkrétní rychlostí.

*void SetReverse (Comm::NXTComm *comm, int port, int power)*

Nastavuje pohyb motoru vzad konkrétní rychlostí.

*void Stop (Comm::NXTComm *comm, int port, bool brake)*

Nařídí motoru zastavit pohyb.

*void BrakeOn (Comm::NXTComm *comm, int port)*

Nařídí motoru, ať začne brzdit.

*void BrakeOff (Comm::NXTComm *comm, int port)*

Nařídí motoru ať přestane brzdit.

*void GoTo (Comm::NXTComm *comm, int port, int power, int tacho, bool brake)*

Nařídí motoru otočení o zvolený úhel a zastavit.

- **NXT:: Sensor**

Obsahuje všechny funkce, které mají něco do činění se senzory

*void SetTouch (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako dotykový senzor.

*void SetSound (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako zvukový senzor

*void SetLight (Comm::NXTComm *comm, int port, bool active)*

Nastavuje senzor na specifickém portu jako světelný senzor

*void SetSonar (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako ultrazvukový senzor

*void SetCmpsNx (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako senzor kompas.

*void SetDistNx (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako senzor pro měření vzdáleností.

*void SetRaw (Comm::NXTComm *comm, int port)*

Nastaví senzor k návratu hrubé hodnoty (jakkoliv neformátované).

*void Set (Comm::NXTComm *comm, int port, SensorType type)*

Nastaví senzor jako specifický typ senzoru.

*int GetValue (Comm::NXTComm *comm, int port)*

Navrací hodnotu ze senzoru.

*int GetAvgCmpsNxValue (Comm::NXTComm *comm, int port, int numValues, int waitTime)*

Získává průměr z více hodnot ze senzoru kompas.

*int GetCmpsNxValue (Comm::NXTComm *comm, int port)*

Získává hodnotu ze senzoru kompas.

*int GetDistNxValue (Comm::NXTComm *comm, int port)*

Získává hodnotu ze senzoru měřícího vzdálenosti.

*int GetCleanDistNxValue (Comm::NXTComm *comm, int port)*

Získává průměr z více hodnot ze senzoru měřícího vzdálenosti.

*int LSGetStatus (Comm::NXTComm *comm, int port)*

Získá stav nízkého rychlostního portu..

```
int GetSonarValue (Comm::NXTComm *comm, int port)
```

Získává hodnotu z ultrazvukového senzoru.

```
void SetSonarOff (Comm::NXTComm *comm, int port)
```

Nařídí ultrazvukovému senzoru zastavení vysílání impulzů.

```
void SetSonarSingleShot (Comm::NXTComm *comm, int port)
```

Nařídí ultrazvukovému senzoru, aby vysílal impulzy, jen když je požadována aktuální hodnota ze senzoru.

```
void SetSonarContinuous (Comm::NXTComm *comm, int port)
```

Nařídí ultrazvukovému senzoru, aby pokračoval ve vysílání impulzů.

```
void SetSonarContinuousInterval (Comm::NXTComm *comm, int port, int interval)
```

Nastaví interval, ve kterém ultrazvukový senzor vysílá impulzy.

```
void writeI2C (Comm::NXTComm *comm, int port, ViUInt8 command[], int replySz, int tx_length)
```

Odešle příkaz k I2C senzoru.

```
ViUInt8 * readI2C (Comm::NXTComm *comm, int port)
```

Čte hodnotu z registru I2C senzoru.

4.2.2 Úloha č.3 - Získání a zobrazení údajů ze senzorů NXT a ovládání motorů robota Mindstorms NXT

Cíl úlohy

Cílem této úlohy je procvičit znalosti získané během studia týkajícího se programovacího jazyka C a programování v konzole. Práci by vám mělo příjemnit využití robota Mindstorms NXT.

Zadání úlohy

- 1) Seznamte se s možnostmi a parametry programovací kostky NXT a jejími senzory.
- 2) Nastudujte možnosti knihovny NXT++.

- 3) Připojte k PC pomocí USB kabelu ovládací kostku NXT se všemi připojenými senzory a motory.
- 4) Za pomocí jazyka C++ a knihovny NXT++ vytvořte následující konzolové aplikace:
 - a. konzolová aplikace, která zajistí ovládání motoru pomocí dotykového senzoru
 - b. konzolová aplikace, která zajistí ovládání všech motorů pomocí klávesnice počítače
 - c. konzolová aplikace, která zajistí zobrazování hodnot ze sensorů na monitoru počítače a ovládání jednotlivých motorů pomocí klávesnice počítače

Popis aplikací

a. konzolová aplikace, která zajistí ovládání motoru pomocí dotykového senzoru

Konzolová aplikace vytvořená v jazyce C++ za použití knihovny NXT++. Po spuštění aplikace bude možno spustit motor aktivací dotykového senzoru. Po puštění dotykového senzoru se motor zastaví.

b. konzolová aplikace, která zajistí ovládání všech motorů pomocí klávesnice počítače

Konzolová aplikace vytvořená v jazyce C++ za použití knihovny NXT++. Po spuštění aplikace bude možno spustit všechny motory stisknutím klávesy „M“. Po stisknutí klávesy „N“ se motory zastaví.

c. konzolová aplikace, která zajistí zobrazování hodnot ze sensorů na monitoru počítače a ovládání jednotlivých motorů pomocí klávesnice počítače

Konzolová aplikace vytvořená v jazyce C++ za použití knihovny NXT++. Po spuštění aplikace se zobrazí informace o stavu jednotlivých sensorů a motorů. Dále budou nadefinovány klávesy, jejichž stisknutím se spustí vybraný motor. Opětovným stisknutím klávesy se daný motor zastaví. Aktuální hodnoty bude možno uložit do textového souboru po stisku vybrané klávesy. S hodnotami se do souboru uloží i aktuální datum a čas.

Teorie

Pro vytvoření výše uvedených aplikací je třeba mít k dispozici počítač s nainstalovaným vývojovým prostředím MS Visual Studio ve verzi 2005 nebo 2008 (možno použít i express verzi).

Dále je nutné použít knihovnu NXT++, která je k dispozici na stránce <http://nxtpp.sourceforge.net/index.php>. Součástí knihovny je i kompletní dokumentace,

proto jsou zde uvedeny jen nejdůležitější funkce. Přehled všech funkcí včetně českého popisu je připojen jako soubor `nxt++.doc`.

Funkce pro práci s robotem NXT (hlavičkový soubor `NXT++.h`)

- **NXT**

Obsahuje všechny funkce, které pracují s řídicí kostkou NXT

bool *Open* (*Comm::NXTComm *comm*)

Zahajuje NXT připojení.

bool *OpenBT* (*Comm::NXTComm *comm*)

Zahajuje NXT připojení prostřednictvím technologie Bluetooth.

void *Close* (*Comm::NXTComm *comm*)

Ukončuje NXT připojení.

void *PlayTone* (*Comm::NXTComm *comm*, *int frequency*, *int duration*)

Přehraje tón.

int *BatteryLevel* (*Comm::NXTComm *comm*)

Vrací aktuální stav baterie.

std::string *GetName* (*Comm::NXTComm *comm*)

Zjistí jméno NXT.

double *GetAvailableFlash* (*Comm::NXTComm *comm*)

Zjistí aktuální velikost volné flash paměti NXT.

double *GetProtocolVersion* (*Comm::NXTComm *comm*)

Zjistí verzi protokolu daného NXT.

double *GetFirmwareVersion* (*Comm::NXTComm *comm*)

Zjistí verzi firmware daného NXT.

- **NXT:: Motor**

Obsahuje všechny funkce, týkající se motorů

*void SetForward (Comm::NXTComm *comm, int port, int power)*

Nastavuje pohyb motoru vpřed konkrétní rychlostí.

*void SetReverse (Comm::NXTComm *comm, int port, int power)*

Nastavuje pohyb motoru vzad konkrétní rychlostí.

*void Stop (Comm::NXTComm *comm, int port, bool brake)*

Nařídí motoru zastavit pohyb.

- **NXT:: Sensor**

Obsahuje všechny funkce, které mají něco do činění se senzory.

*void SetTouch (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako dotykový senzor.

*void SetSound (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako zvukový senzor.

*void SetLight (Comm::NXTComm *comm, int port, bool active)*

Nastavuje senzor na specifickém portu jako světelný senzor.

*void SetSonar (Comm::NXTComm *comm, int port)*

Nastavuje senzor na specifickém portu jako ultrazvukový senzor.

*int GetValue (Comm::NXTComm *comm, int port)*

Navrací hodnotu ze senzoru.

*int GetSonarValue (Comm::NXTComm *comm, int port)*

Získává hodnotu z ultrazvukového senzoru.

Funkce pro ovládání programu pomocí klávesnice (hlavičkový soubor windows.h)

`GetAsyncKeyState(kódklávesy)` Zjištění, zda je stisknuta klávesa definovaná kódem klávesy.

Jednotlivé kódy kláves jsou uvedeny v dokumentaci MSDN. Níže jsou uvedeny kódy pro jednotlivá písmena:

(0x41)	A	(0x4A)	J	(0x53)	S
(0x42)	B	(0x4B)	K	(0x54)	T
(0x43)	C	(0x4C)	L	(0x55)	U
(0x44)	D	(0x4D)	M	(0x56)	V
(0x45)	E	(0x4E)	N	(0x57)	W
(0x46)	F	(0x4F)	O	(0x58)	X
(0x47)	G	(0x50)	P	(0x59)	Y
(0x48)	H	(0x51)	Q	(0x5A)	Z
(0x49)	I	(0x52)	R		

Tab. 4. Tabulka kódů příslušných kláves

Zápis do souboru

Přidání dat do souboru:

```
file *f
f=fopen("data.txt", "ab");
fprintf(f, "text");
fclose(f);
```

Přepsání souboru:

```
file *f
f=fopen("data.txt", "wb");
fprintf(f, "text");
```

```
fclose(f);
```

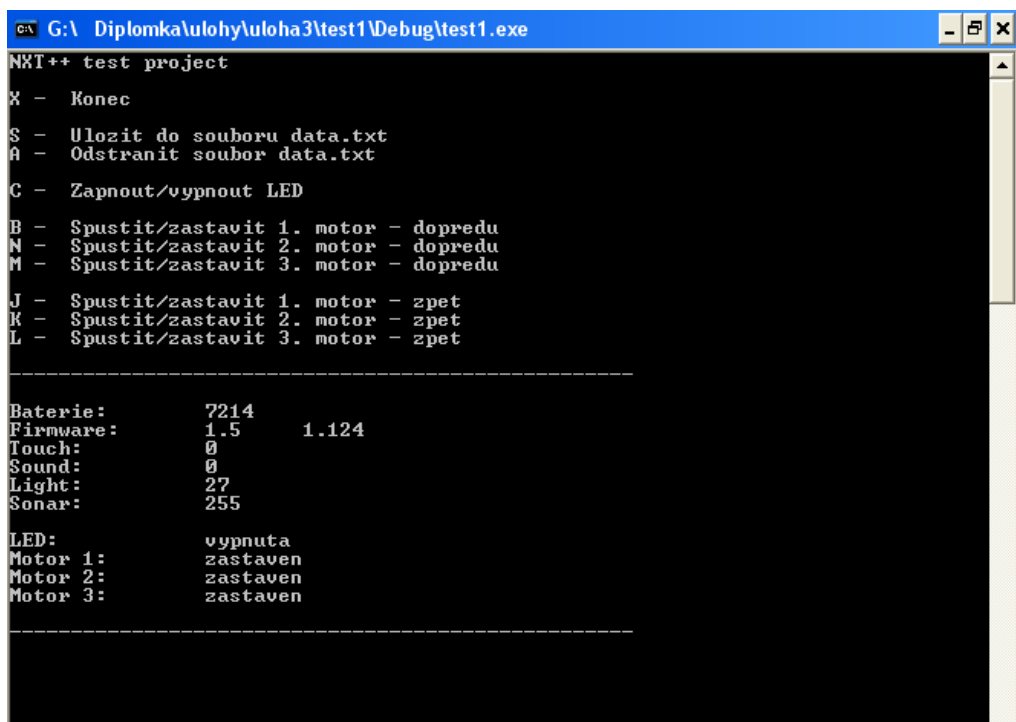
Datum a čas (knihovna time.h)

Získání aktuálního data a času:

```
time_t cas;  
  
struct tm *lmcas;  
  
cas = time(NULL);  
  
lmcas = localtime(&cas);  
  
f=fopen("data.txt", "ab");  
  
// zapsání času do aktuálně otevřeného souboru:  
  
fprintf(f, "%s\n", asctime(lmcas));
```

Závěr

Pokuste se vytvořit zadanou aplikaci a vyzkoušejte si všechny funkce sestaveného robota. Shledáte, že programování v jazyce C++ v prostředí MS Visual Studio se díky využití robota Mindstorms NXT stane mnohem zábavnější záležitostí. V případě nějakého problému využijte jako nápovědu vzorovou aplikaci.



```
G:\Diplomka\ulohy\uloha3\test1\Debug\test1.exe  
NXT++ test project  
X - Konec  
S - Ulozit do souboru data.txt  
A - Odstranit soubor data.txt  
C - Zapnout/vypnout LED  
B - Spustit/zastavit 1. motor - dopredu  
N - Spustit/zastavit 2. motor - dopredu  
M - Spustit/zastavit 3. motor - dopredu  
J - Spustit/zastavit 1. motor - zpet  
K - Spustit/zastavit 2. motor - zpet  
L - Spustit/zastavit 3. motor - zpet  
-----  
Baterie:      7214  
Firmware:    1.5    1.124  
Touch:       0  
Sound:       0  
Light:       27  
Sonar:       255  
-----  
LED:         vypnuta  
Motor 1:     zastaven  
Motor 2:     zastaven  
Motor 3:     zastaven  
-----
```

Obr. 32. Náhled aplikace v programovacím jazyce C

4.3 Laboratorní úloha do výuky programovacího jazyka C#

Cílem této úlohy je praktické vyzkoušení znalosti programovacího jazyka C#. Motivací pro vytvoření programu je lákavá odměna na závěr v podobě ovládní modelu robota Mindstorms NXT (RoboArm). V podstatě je úkolem této úlohy naprogramovat aplikaci podobnou té, která byla vytvořena jako součást této diplomové práce. Student má za úkol se podobně jako v předcházejících úlohách, seznámit s možnostmi robota Mindstorms NXT. Proto zde, z důvodu nadbytečnosti, nebude zadání úlohy vypisováno. Je však k dispozici na přiloženém CD-ROM disku.

ZÁVĚR

Cílem této diplomové práce bylo splnění čtyř požadovaných úkolů stanovených v zadání.

Dva ze čtyř úkolů se staly náplní teoretické části. První polovina teoretické části se proto zabývá zpracováním literární studie softwarového a hardwarového vybavení robota NXT. Věnuje se popisu řídicí kostky NXT, jejím parametrům a podrobnému popisu jednotlivých senzorů a servomotorů. Dále se pak zabývá popisem grafického programovacího jazyka NXT – G, který je dodáván společně se systémem LEGO Mindstorms NXT. A na závěr se věnuje popisu uživatelského rozhraní firmwaru nainstalovaného v řídicí kostce NXT.

Druhá polovina teoretické části se zabývá studií programovacích jazyků, které lze použít pro naprogramování robota NXT, a možnostem komunikace zmiňovaného robota s okolím. Věnuje se např. popisu programovacích prostředí a jazyků využitých v praktické části, tedy systému MATLAB a knihovně RWTH-Mindstorms NXT toolbox, programovacím jazykům C a C# a příslušným knihovnám NXT++ a NXT.NET. Mimoto je zde popsáno i pár dalších vybraných programovacích prostředí. Závěr této části diplomové práce popisuje možnosti komunikace NXT s okolím pomocí USB rozhraní a technologii bluetooth.

Praktická část diplomové práce si vzala za úkol splnění zbývajících dvou bodů zadání. Prvním z nich je vytvoření aplikace určené k ovládní robota NXT (Machine). Tato část se proto zabývá popisem vytvořené aplikace. K vývoji aplikace byla zvolena moderní platforma .NET, která se stává v poslední době velmi populární. Samotný zdrojový kód byl napsán v programovacím jazyce C#. K výběru tohoto jazyka značnou měrou přispěla existence kvalitní knihovny NXT.NET a možnost vytvoření aplikace spustitelné bez potřeby využití některého programového prostředí, jako je např. MATLAB. Zmíněná aplikace umožňuje dálkové ovládní robota pomocí technologie bluetooth a zobrazení aktuálních informací o stavu řídicí kostky NXT, včetně zobrazení dostupných dat ze všech připojených senzorů.

A konečně poslední část diplomové práce se zabývá návrhem čtyř laboratorních úloh využívajících zmíněného robota. Vytvořené úlohy jsou určené pro podporu výuky programovacího jazyka C respektive C++, C# a práce v programovém prostředí MATLAB. Dvě z úloh jsou určeny pro programové prostředí MATLAB. První z nich má za úkol načtení a přehledné zobrazení dat ze všech senzorů robota NXT. Druhá pak vytvoření jednoduché aplikace pro řízení tohoto robota. Dále byl vytvořen návrh laboratorní úlohy

určené pro výuku programovacího jazyka C. Úkolem úlohy je mimo jiné vytvoření jednoduché konzolové aplikace sloužící k ovládní servomotorů robota pomocí klávesnice PC přes USB rozhraní, načítání dat ze senzorů, případně jejich uložení do textového souboru spolu s aktuálním časem. Poslední úlohou je vytvoření aplikace v jazyce C#, podobné té, vytvořené jako součást této diplomové práce.

Závěrem by stálo za zmínku, že potenciál robota NXT tato práce ani zdaleka nenaplnila. Tento robot by se dal využít nejen jako podpora výuky některých vybraných předmětů, u kterých napomáhá k oživení probírané látky a umožňuje její podání zábavnější formou. Ale dal by se úspěšně použít i pro výuku samostatného předmětu týkajícího se robotiky, kde by se tento robot stal názornou pomůckou při výuce základů tohoto oboru.

ZÁVĚR V ANGLIČTINĚ

The aim of this master thesis was fulfilment of four required tasks, which were determined in the assignment.

Two of four parts came to be a contents of the theoretical part. That is why the first half of the theoretical part deals with elaboration of literary study of software and hardware equipment of the NXT robot. It attends to the description of the NXT Brick, its parameters and detailed description of individual sensors and servomotors. Next it is engaged in description of graphical programming language NXT-G, which is provided with NXT LEGO Mindstorm system. And the end of this part is devoted to description of user interface of the firmware, which is installed in the NXT brick.

The second half of the theoretical part deals with the study of programming languages, which could be used to programming of NXT robot, and with possibility of communication of this robot with the surrounding area. For example it attends to the description of programming environment and languages, which are used in the practical part. It attends to the following topics:

- MATLAB system and RWTH-Mindstorms NXT toolbox library.
- Programming languages C++ and C# and its relevant libraries NXT and NXT.NET.
- Others programming environments.

Conclusion of this part of the master thesis describes abilities of the communication of the NXT with surrounding area by the USB or Bluetooth.

The practical part of this master thesis takes the issue of fulfilment of the last two points of the assignment. The first task is creation of the application, which is intended to control of the NXT robot (Machine). That is why this part deals with description of this application. For the development of the application it was chosen modern platform .NET, which is very popular now. The source code was written in the C# programming language. For the selection of this programming language it was helpful the fact of existence of quality library NXT.NET and the ability of creation of the application, which could be executed without requirement of any programming environment such as MATLAB. The mentioned application allows remote control of the robot by the way of the Bluetooth technology and

display of the actual information about state of the NXT control brick including display of the available values from all connected sensors.

And finally, the last part of this master thesis is concerned with design of four laboratory exercises, which make use of this robot. Created tasks are intended to support education of C (or C++) programming language, C# programming language and working with MATLAB programming environment. Two exercises are intended for MATLAB programming environment. The first of it was called to do reading and showing values of all sensors of the NXT robot. The second exercise is creating of simple application to control this robot. Next it was created project of the laboratory exercise for education of the C language. The task of this exercise is creating of the simple console application to control servomotors of the NXT robot (connected over USB) by the PC keyboard. The application allows to save values to a text file with actual date and time. Assignment of the last (fourth) exercise is creation of the similar application in C#, such as the application created as part of this master thesis.

Finally, it would be worth to a mention, that this thesis does not fill the potential of the NXT robot wide of the mark. This robot could be used not only as support of the education of some selected subject, where it assists to animation of the subject matter and allows to its presentation by funnier way. But it could be successfully used to education of the robotics subject, where this robot could be used as illustration tool in the education of basics of this branch.

SEZNAM POUŽITÉ LITERATURY

Monografie:

- [1] SCHOLZ, Matthias Paul. Advanced NXT: The Da Vinci Inventions Book. Jennifer Whipple. Apress, c2007. 369 s. ISBN 978-1-59059-843-6.
- [2] PERDUE, David J. THE UNOFFICIAL LEGO MINDSTORMS NXT INVENTOR'S GUIDE. Megan Dunchak; Christina Samuell. 1st edition. San Francisco : No Starch Press, Inc, c2008. 296 s. ISBN 978-1-59327-154-1.
- [3] ASTOLFO, Dave, FERRARI, Mario, FERRARI, Giulio. BUILDING ROBOTS WITH LEGO MINDSTORMS NXT. Audrey Doyle. 1st edition. Burlington : Syngress Publishing, Inc, c2007. 447 s. ISBN 978-1-59749-152-5.
- [4] GASPERI, Michael, HURBAIN, Philippe, HURBAIN, Isabelle. Extreme NXT : Extending the LEGO MINDSTORMS NXT to the Next Level. Susannah Davidson Pfalzer. [s.l.] : Apress, c2007. 286 s. ISBN 978-1-59059-818-4.
- [5] BOOGAARTS, Martijn, et al. THE LEGO MINDSTORMS NXT IDEA BOOK : design, invent, and built. Nancy Sixsmith, Megan Dunchak. 1st edition. San Francisco : No Starch Press, Inc, c2007. 344 s. ISBN 978-1-59327-150-3.
- [6] BAGNALL, Brian. Maximum LEGO NXT : Building Robots with Java Brains. Edited by Sylvia Philipps. 1st edition. Canada : Variant Press, c2007. 505 s. ISBN 978-0-9738649-1-5.

Internetové zdroje:

- [7] *MATLAB* [online]. MetaCentrum. [cit. 2007-02-01]. Dostupný z WWW: <<http://meta.cesnet.cz/cs/docs/software/matlab.html>>
- [8] *MATLAB* [online]. Wikipedia, the free encyclopedia. [cit. 2008-08-04]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/MATLAB>>
- [9] *MATLAB* [online]. HUMUSOFT. [cit. 2007-09-06]. Dostupný z WWW: <<http://www.humusoft.cz/matlab/matlab.htm>>
- [10] *C/C++* [online]. LinuxSoft.cz. [cit. 2004-08-30]. Dostupný z WWW: <http://www.linuxsoft.cz/article.php?id_article=370>

- [11] C (programovací jazyk) [online]. Wikipedia, the free encyclopedia. [cit. 2007-07-29]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/C_\(programovac%C3%AD_jazyk\)](http://cs.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk))>
- [12] NXT++ [online]. NXT Wiki. [cit. 2008-01-24]. Dostupný z WWW: http://nxtpp.wetter61169.de/mediawiki-1.6.9/index.php/Visual_Studio_Tutorials:_Setting_up_Visual_Studio
- [13] C# [online]. Wikipedia, the free encyclopedia. [cit. 2001-02-04]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/C_Sharp>
- [14] Java [online]. Java Mgr. Pavel Dohnal [cit. 2008-08-24]. Dostupný z WWW: <<http://java.euweb.cz/index.php>>
- [15] Java [online]. Dioné (studentský informační server) [cit. 2001-02-04]. Dostupný z WWW: <<http://dione.zcu.cz/java/sbornik/toc.html>>
- [16] Microsoft .NET Framework 3.5 [online]. Microsoft [cit. 2001-02-04]. Dostupný z WWW: <<http://www.microsoft.com/downloads/details.aspx?familyid=333325FD-AE52-4E35-B531-508D977D32A6&displaylang=cs>>
- [17] NXT .NET [online]. CodePlex. [cit. 2008-08-12]. Dostupný z WWW: <<http://www.codeplex.com/NxtNet>>
- [18] Microsoft Robotics Studio [online]. Vyvojar.cz [cit. 2006-11-06]. Dostupný z WWW: <<http://www.vyvojar.cz/Articles/429-roboti-od-lega-a-microsoft-robotics-studio.aspx>>
- [19] Igr [online]. Root.cz. [cit. 2001-02-04]. Dostupný z WWW: <<http://www.root.cz/clanky/softwareva-sklizen-25-6-2008/>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MATLAB Matrix laboratory

NXT Řídící jednotka robota Mindstorm NXT

PC Personal Computer

RWTH Rheinisch-Westfaelische Technische Hochschule

USB Universal Serial Bus

SEZNAM OBRÁZKŮ

Obr. 1. Robot Mindstorms NXT	10
Obr. 2. NXT	11
Obr. 3. NXT – komunikace s okolím.....	12
Obr. 4. NXT – připojení senzorů a motorů.....	12
Obr. 5. Dotykový senzor	13
Obr. 6. Zvukový senzor.....	14
Obr. 7. Světelný senzor	14
Obr. 8. Citlivost světelného senzoru při rozeznávání barev	15
Obr. 9. Ultrazvukový senzor	16
Obr. 10. Servomotor	16
Obr. 11. NXT - G.....	18
Obr. 12. Ikony v menu NXT	20
Obr. 13. Ovládací tlačítka NXT	20
Obr. 14. Uživatelské rozhraní programu MATLAB.....	23
Obr. 15. Ukázka použití funkcí z knihovny LeJOS	28
Obr. 16. Příklad programu v nástroji Visual Programming Language	30
Obr. 17. Ukázka softwaru RobotC.....	31
Obr. 18. Simulační systém Igr	32
Obr. 19. NXT – komunikace pomocí USB.....	33
Obr. 20. NXT – Bluetooth komunikace.....	33
Obr. 21. LEGO Mindstorms NXT: Machine	36
Obr. 22. Náhled vytvořené aplikace.....	37
Obr. 23. Konstrukční vrstvení RWTH Mindstorms toolboxu	49
Obr. 24. Úloha č.1 – Mindstorms NXT	50
Obr. 25. Úloha č.1 – NXT a senzory	51
Obr. 26. Úloha č.1 – Náhled vzorového uživatelského prostředí	53
Obr. 27. Úloha č.1 – Náhled vzorové aplikace	54
Obr. 28. Úloha č.2 – Mindstorms NXT	55
Obr. 29. Úloha č.2 – NXT Machine.....	56
Obr. 30. Úloha č.2 – Náhled vzorového uživatelského prostředí	58
Obr. 31. Úloha č.2 – Náhled vzorové aplikace	59

Obr. 32. Náhled aplikace v programovacím jazyce C 69

SEZNAM TABULEK

Tab. 1. Kompatibilita Bluetooth zařízení.....	34
Tab. 2. Základní třídy knihovny NXT.NET.....	38
Tab. 3. NXT.NET - Funkce třídy NXT.....	39
Tab. 4. Tabulka kódů příslušných kláves.....	68

SEZNAM PŘÍLOH

Příloha P I: Disk CD-ROM