Ovládání laboratorního modelu robota Mindstorms NXT s využitím umělé inteligence - neuronových sítí

Control laboratory model Mindstorms NXT robot using artificial intelligence - neural networks

Bc. Petr Tihlařík

Diplomová práce
2010

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Petr TIHLAŘÍK**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Ovládání laboratorního modelu robota Mindstorms NXT s využitím umělé inteligence – neuronových sítí**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na téma Ovládání laboratorního modelu robota Mindstorms NXT s využitím umělé inteligence – neuronových sítí.**
2. **Vypracujte rozšiřující studii o možnostech ovládání laboratorních modelů Mindstorms NXT pomocí PC s využitím umělé inteligence – neuronových sítí.**
3. **Vytvořte pomocí literatury novou aplikaci robota a rozšiřte tak stávající databázi laboratorních úloh.**
4. **Proveďte popis úloh a zpracujte dokumentaci k úlohám.**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **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.**
2. **GASPERI, Michael, HURBAIN, Philippe, HURBAIN, Isabelle. Extreme NXT: Extending the LEGO MINDSTORM NXT to the Next Level. Susannah Davidson Pfalzer. [s.l.] : Apress, c2007. 286 s. ISBN 987-1-59059-818-4.**
3. **BOOGAARTS, Martijn, et al. THE LEGO MINDSTORMS NXT IDEA BOOK : design, invent, and build. Nancy Sixsmith, Megan Dunchak. 1st edition. San Francisco : No Starch Press, Inc, C2007. 344 s. ISBN 978-1-59327-150-3.**
4. **BAGNALL, Brian. Maximum LEGO NXT : Building Robots with Java Brains. Editied by Sylvia Philipps. 1st edition. Canada : Variat Press, C2007. 505 s. ISBN 978-0-9738649-1-5.**
5. **GURNEY, Kevin. An Introduction to Neural Networks. 1st edition. CRC Press, 1997. 234 s. ISBN 978-1857285031.**
6. **FREEMAN J. A.: Simulating Neural Networks with Mathematica, Adison-Weslez Publishing Company, 1994, ISBN 0-201-56629-X.**
7. **BOSE N.K., LIANG P.: Neural Network Fundamentals with Graphs, Algorithms, and Applications, McGraw-Hill Series in Electrical and Computer Engineering, ISBN 0-07-006618-3, 1996.**
8. **ZELINKA I.: Umělá inteligence I, VUT Brno, ISBN 80-214-1163-5, 1998.**

Vedoucí diplomové práce: **Ing. Roman Šenkeřík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **19. února 2010**

Termín odevzdání diplomové práce: **8. června 2010**

Ve Zlíně dne 19. února 2010

prof. Ing. Vladimír Vašek, CSc.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Abstrakt česky

Práce se zabývá analýzou programování MINDSTORMS NXT robotů s využitím neuronových sítí. V teoretické části jsou uvedeny možnosti programování robotů pomocí různých programovacích jazyků, v praktické části je probráno řízení konkrétní neuronovou sítí.

Klíčová slova: robot, MIDSTORMS NXT, ovládání, umělá inteligence, neuronové sítě, LeJOS

ABSTRACT

Abstrakt ve světovém jazyce

This master thesis is focused to programming MINDSTORMS NXT robot using neural networks. The theoretical part introduces the possibility of robot programming using various programming languages. The practical part includes a robot control via neural network.

Keywords: robot, MINDSTORMS NXT, control, artificial inteligence, neural networks, LeJOS

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

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně ……

podpis diplomanta

OBSAH

ÚVOD

Lidstvo již odedávna fascinovala myšlenka vytvoření umělé entity s vlastní inteligencí, která by mu usnadnila nebo zcela samostatně vykonávala fyzicky náročné, nebezpečné nebo často se opakující, nezáživné činnosti. V posledních letech dosahuje automatizace velkých pokroků ve všech odvětvích. Roboti mohou mít různý vzhled a schopnosti. Slovo robot se poprvé objevilo v dramatu RUR od Karla Čapka. V jeho pojetí znamená toto slovo mechanického člověka. Nyní takto označujeme zařízení vykonávající určitou činnost v reakci s reálným okolím, na základě určité formy samostatného řízení. Aby byl robot schopný reagovat na okolní prostředí, bývá vybaven velkým množstvím různorodých senzorů. Vyhodnocení jednotlivých situací musí provádět ovládací software.

Robotika se velmi rychle rozvíjí, takže se s ní studenti často setkávají v rámci výuky na technicky zaměřených školách. V roce 2006 uvedla společnost LEGO na trh novou generaci robotů, stavebnici MINDSTORMS NXT, která je vhodná pro první seznámení s programováním jednoduchých robotů. Díky univerzálnosti stavebnice LEGO je možné vytvořit téměř neomezené množství jednotlivých typů robotů. Jako součást stavebnice jsou dodávány základní senzory a 3 motory. Tato základní výbava stačí pro seznámení se s programováním robotů obecně. Otevřenost zdrojových kódů a dostupná kompletní dokumentace nahrává projektům vytvářejícím alternativní firmware a knihovny pro programování v mnoha jazycích.

S rozšiřujícími požadavky na roboty roste také náročnost řídícího software. Cílem je dosáhnout takového stupně chování, které bychom mohli srovnat s inteligencí živých organismů. S vývojem techniky byly postupně navrhovány různé postupy a algoritmy jak takového chování dosáhnout. Vědní disciplína zabývající se těmito problémy se nazývá umělá inteligence. V současné době je jednou z nejrychleji se rozvíjejících vědeckých a technických disciplín.

Neuronové sítě jsou matematické modely napodobující činnost živých organismů. Skládají se z umělých neuronů, jejichž předobrazem je biologický neuron. Neurony jsou navzájem propojeny, předávají si signály a transformují je podle určených přenosových funkcí. Každý neuron má libovolný počet vstupů, ale pouze jeden výstup. Neuronové sítě mají využití i v jiných oblastech než robotika, používají se například pro rozpoznávání a

kompresi obrazu, nebo zvuků, předpovídání vývoje časových řad, v lékařství, nebo dokonce i k filtrování spamu.

Tato práce se zabývá možnostmi programování NXT robotů pomocí neuronových sítí. Praktická část se zabývá konkrétním postupem ovládání robota neuronovou sítí včetně přípravy potřebného software, učení sítě a praktického využití naučené sítě.

I. TEORETICKÁ ČÁST

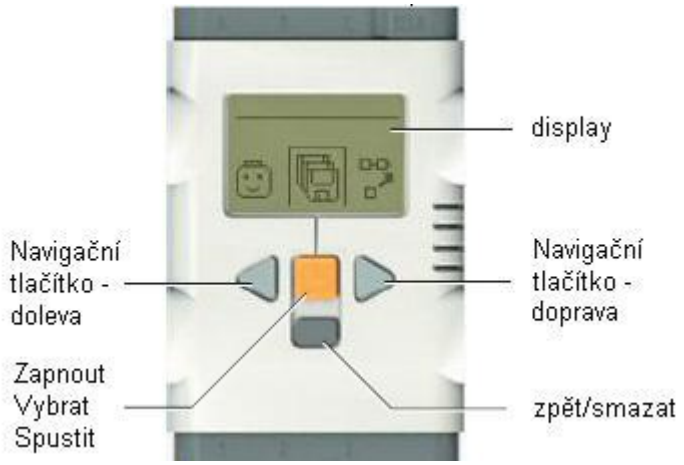
1 VYUŽITÍ A MOŽNOSTI PROGRAMOVÁNÍ ROBOTŮ NXT

1.1 Možnosti NXT robotů

Roboti NXT jsou dodávání jako stavebnice, která tvoří dostupný prvek pro začínající programátory. Jsou také v mnoha případech využíváni pro výuku. Výhodou těchto stavebnic je jejich modularita. Ze stavebnice je možné vytvořit téměř cokoli, co tvůrce napadne. Součástí balení je jednoduchý návod pro sestavení prvního robota a základní software pro programování. Nejčastější konstrukcí bývají různá vozidla, humanoidi a napodobeniny strojů.

Obrázek 1: Lego Mindstorms NXT

Hlavní částí robota je „základní jednotka“ ke které je možné připojit 3 servomotory a 4 různé senzory. Servomotor dosahuje rychlosti až 200 otáček za minutu. Jeho velkou výhodou je možnost řízení otáčení s přesností až 1°



Obrázek 2: Základní jednotka NXT

Základní dodávané senzory jsou

- ultrazvukový – detekuje jak daleko je robot od překážky. Dokáže detekovat překážky do vzdálenosti 255cm. Protože pracuje na principu ultrazvuku, dva a více robotů pohybujících se ve stejném okolí se navzájem ovlivňují
- tlakový – jedná se o jednoduché tlačítko vracející stav stlačeno / nestlačeno
- zvukový – nejčastěji se používá pro spuštění různých příkazů, kdy je možné například tlesknutím spustit, nebo zastavit některou z činností. Senzor vrací procentuální hodnotu hlasitosti zvuku do 90dB
- světelný – rozeznává pouze rozdíl mezi světlým a tmavým podkladem, návratová hodnota je v rozmezí 0 (tmavý) až 100 (světlý)

Robota je možno doplnit i o senzory dodávané jinými výrobci. Může se jednat například o senzor digitální kompas, barevný senzor, infračervený detektor apod.

1.2 Hardwarové možnosti dané konstrukcí NXT

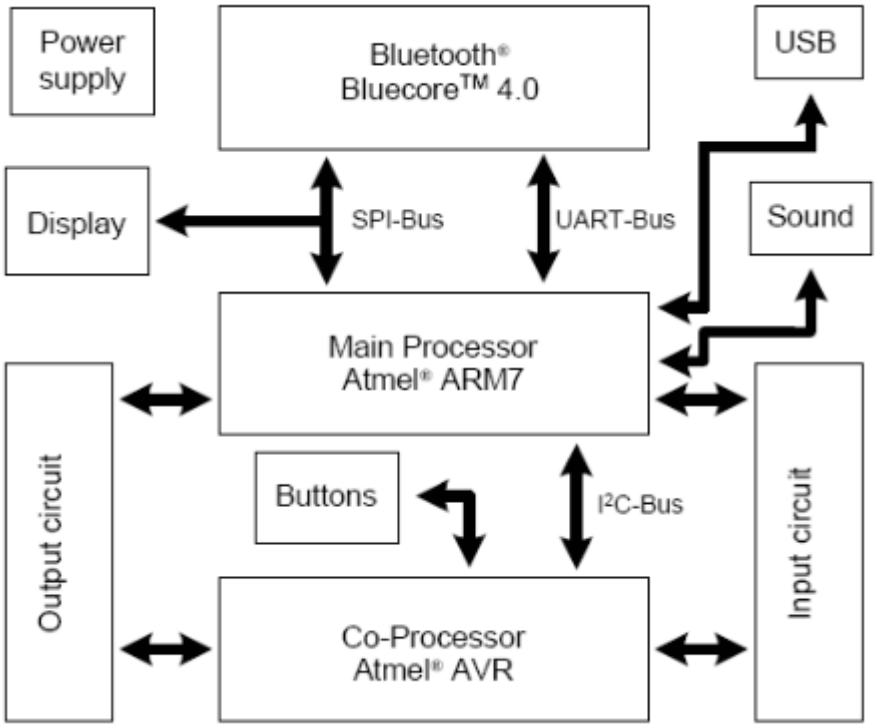
Součástky stavebnice NXT jsou kompatibilní s kostkami LEGO a je možné je kombinovat se stavebnicemi LEGO Technics. Součástky většinou neobsahují kovové části, jejich povrch je vždy tvořen gumou nebo plastem. Kromě standardních kostek je zde několik druhů ozubených převodů, kloubových spojů a os. Komunikace mezi jednotlivými prvky a řídící jednotkou je realizována pomocí 6-ti žilových kabelů s koncovkami podobajícími se RJ11. Rozdíl je v umístění jistícího kolíku na stranu (RJ11 má jistící kolík uprostřed).

Možnosti propojení s počítačem, případně jiným zařízením jsou dvě. Je možné propojení pomocí standardního USB kabelu, nebo bezdrátově pomocí Bluetooth.

Přes Bluetooth je možné se připojit až na 3 zařízení současně. V jednom okamžiku ale komunikují pouze dvě zařízení mezi sebou.

1.2.1 Technické specifikace

- 32 bitový mikroprocesor ARM7
    - 48 MHz
    - 256 KB flash paměť
    - 64 KB paměti RAM
- 8 bitový co-procesor AVR
    - 4 MHz
    - 4 KB flash paměť
    - 512 B paměti RAM
- LCD display s rozlišením 100 x 64 pixelů
- reproduktor
    - 8 bitový zvukový kanál
    - frekvenční rozsah 2 – 16 kHz
- USB 2.0 port (maximální přenosová rychlost 12Mbit/s)
- Bluetooth třídy II v 2.0
    - podporuje komunikaci po sériovém portu (SSP)
- Napájení je řešeno pomocí 6 AA baterií



Obrázek 3: Blokové schéma řidící jednotky

1.2.2 Porty

- 4 vstupní porty
- 3 výstupní porty

Vstupní i výstupní porty využívají 6-ti vodičovou digitální platformu. Komunikace je možná dvojím způsobem, buď analogově, nebo digitálně.

1.3 Způsoby řízení NXT

Obecně existují dvě možnosti řízení.

1.3.1 Remote řízení

Jak již název napovídá, tento způsob řízení je založen na provádění jednotlivých příkazů, předávaných jiným připojeným zařízením. Může se jednat o zařízení připojené přes USB nebo Bluetooth. Robota NXT tedy můžeme na dálku ovládat i pomocí mobilního telefonu nebo PDA vybaveného Bluetooth a příslušným softwarem.

1.3.2 On-Board řízení

Tento způsob řízení vyžaduje uložit do pamětí robota vytvořený program. Spuštění programu je prováděné pomocí ovládacích prvků na těle robota. Do paměti robota je možné uložit více programů (jsme omezeni vnitřní pamětí robota). Při spouštění programu je nutné respektovat současnou konstrukci robota, aby bylo zabráněno mechanickému poškození. Veškeré další chování robota je ovládáno řídící jednotkou, pokud nepotřebujeme například kooperaci více robotů, není nutné mít v tomto režimu připojené další zařízení.

1.4 Firmware

Řídící jednotka obsahuje základní ovládací program, firmware. Je to v podstatě jednoduchý operační systém robota. Je uložen ve flash paměti, takže nedojde k jeho smazání ani po odpojení napájení.

Základní firmware je v robotovi nahrán přímo od výrobce. Jako u jiných zařízení ale probíhá neustále jeho vývoj a výrobce umožňuje stažení nových verzí ze svého webu.

Obrázek 4: Možnosti originálního firmware

Jsou dostupné také neoficiální firmwary třetích stran, které se můžou hodit při specifickém použití robotů. Tyto firmwary bývají charakterizovány menší velikostí ve srovnání s originálními verzemi, takže je poté možné využít větší část paměti pro své aplikace.

Pokud dojde při nahrávání nového firmware k chybě, nedojde k poruše funkčnosti robota. Řídící jednotka disponuje nouzovým režimem, který umožní opětovné nahrání firmware. Tento nouzový režim se spouští pomocí tlačítka, které je skryto uvnitř první upevňovací díry pod USB konektorem.

2 PROGRAMOVÁNÍ A KOMUNIKACE NXT

Protože společnost Lego uvolnila zdrojové kódy, vytvořilo se mnoho dalších projektů, které se zabývají programováním NXT jednotek. Většina těchto projektů je dostupná jako OpenSource.

2.1 Programové prostředí a dostupné jazyky

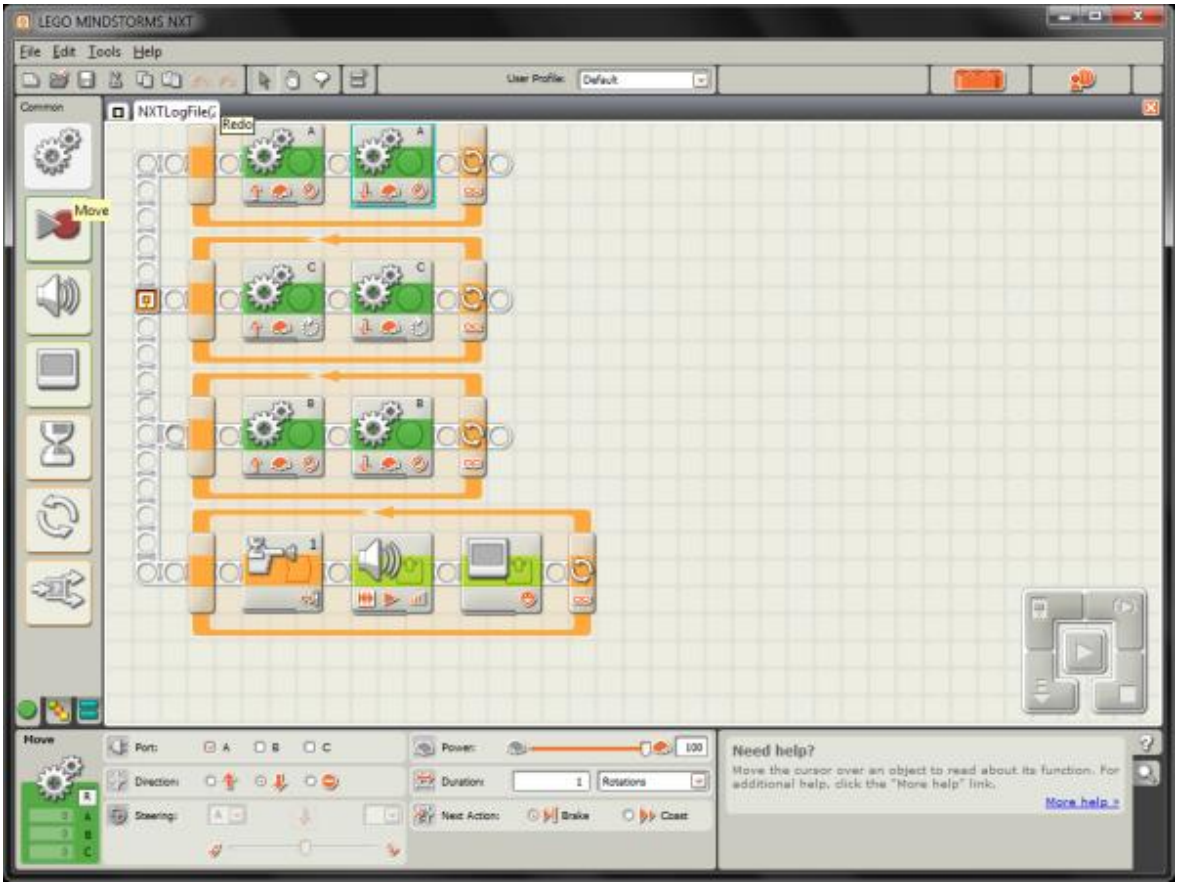
2.1.1 NXT Program

V menu originálního firmwaru je možné vytvořit jednoduchý program pro robota. Samozřejmě možnosti takového programování jsou značně omezené. Prostředí nám nabízí pouze nastavení akce 5 boxů (vstupní akce, výstupní akce a následující akce).

2.1.2 Lego mindstorms NXT: NXT-G

Toto programovací prostředí je dodáváno spolu se stavebnicí. Vývoj aplikace zajišťuje přímo výrobce stavebnice. Je založena na vývojovém nástroji LabView firmy National Instruments, který je využíván pro vývoj elektroniky.

Samotné programování probíhá pomocí skládání a propojování jednotlivých funkčních bloků reprezentovaných grafickými objekty. Tento způsob tvoření programů je vhodný spíše pro začínající programátory.



Obrázek 5: Programování v NXT-G

Součástí prostředí je také rozhraní pro nahrání zkompilovaného programu do řídící jednotky robota, správu paměti a spouštění programů.

NXT-G není vhodné pro tvorbu složitějších programů. Není zde možné pracovat s čísly s plovoucí desetinnou čárkou, chybí datový typ pole a debugging programů je značně omezený.

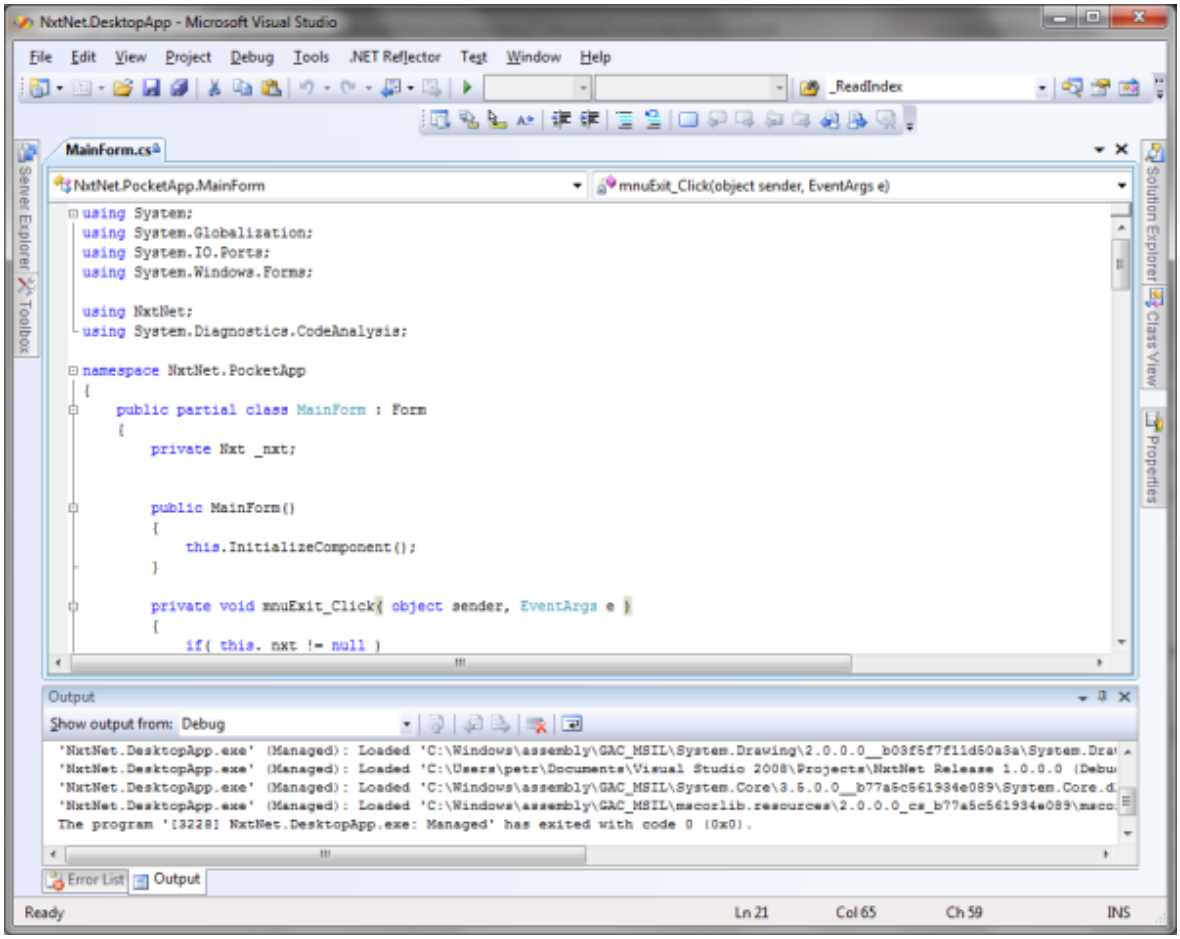
2.1.3 Programovací jazyk C, C++

Vzhledem k všestrannosti a rozšířenosti jazyka C++ jistě nikoho nepřekvapí, že NXT roboty je možné programovat i v něm. Programování je možné v libovolném prostředí umožňující vývoj v C++. Je potřeba pouze doinstalovat knihovnu NXT++.

2.1.4 Programovací jazyk C#

C# je objektově orientovaný programovacím jazykem vyvinutým společností Microsoft. Je založen na principech jazyků C++ a Java (syntaxi čerpá z jazyka C).

Pro vývoj programů v tomto jazyce slouží knihovna NXT.NET. Součástí tohoto projektu je také knihovna *NxtNet.MobileLib.dll*, která umožňuje ovládání robotů pomocí přenosných zařízení s operačním systémem Windows Mobile.



Obrázek 6: Programování NXT v C#

2.1.5 Programování v prostředí MATLAB

Matlab je programové prostředí a skriptovací programovací jazyk pro vědeckotechnické numerické výpočty, modelování, návrhy algoritmu, počítačové simulace, analýzu a prezentaci dat, měření, zpracování signálů, návrhy řídících 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.

Matlab vynikl zkrácením slov MATrix LABoratory. Jak již název napovídá, klíčovou datovou strukturou při výpočtech v Matlabu jsou matice. Syntaxe použitého jazyka vychází z jazyka Fortran. [9]

2.1.6 RWTH Mindstorms NXT Toolbox

Pro Matlab je dostupných několik knihoven funkcí, označovaných jako toolboxy, které rozšiřují použitelnost Matlabu v dalších vědeckých a technických oborech.

RWTH Mindstorms NXT Toolbox je knihovna vytvořena pro systémové prostředí Matlab. Obsahuje funkce sloužící k navázání komunikace s robotem LEGO Mindstorms NXT a posléze k jeho kontrole a ovládání.

Funkce, které knihovna nabízí, jsou roztříděné 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

[10]

2.1.7 Java

Java je objektově orientovaný programovací jazyk, vyvinutý firmou Sun Microsystems. Syntaxe vychází z jazyků C a C++. Velkou výhodou Javy je její robustnost a nezávislost na architektuře. Vytvořená aplikace běží na libovolném operačním systému. Je potřeba pouze to, aby byl na dané platformě nainstalován správný virtuální stroj.

Nevýhodou Javy je pomalejší start aplikací (aplikaci je nutné nejprve přeložit a poté teprve spustit). U menších programů jsou znatelné vyšší paměťové nároky způsobené nutností mít v paměti celé běhové prostředí.

2.1.8 LeJOS

Projekt LeJOS nabízí knihovnu s funkcemi sloužícími ke komunikaci a ovládání robota v jazyce Java. Součástí projektu je také nový firmware pro řídící kostku NXT. Jsou

dostupné jednotlivé třídy pro kompletní práci s NXT (ovládání motorů, senzoru, ladící funkce a komunikace pomocí Bluetooth).

Obrázek 7: Logo leJOS

Výhodou tohoto firmware je jeho menší velikost, takže v řídící jednotce robota je více místa pro vlastní programy, nevýhodou je značně zjednodušené ovládací menu.

Obrázek 8: Menu LeJOS firmware

2.1.9 Přehled vlastností jednotlivých jazyků

Tabulka 1: Přehled prostředí a jejich vlastností

Prostředí		NXT-G	ROBOTC	NXC	NXJ
Jazyk		Graphical	C	C-like	JAVA
Platformy		Windows, MAC OSX	Windows	Windows, MAC OSX,LINUX	Windows
Schopnost uživatele		Nováček, středně pokročilý	Nováček, pokročilý	Středně pokročilý, pokročilý	Pokročilý, zkušený
Použitelnost (1 - 10, 10 nejlepší)		10	8	6	4
Relativní rychlost programu		1X	130X	25X	n/a
IDE		ANO	ANO	ANO	s Eclipse
Vývoj v reálném čase		NE	ANO	částečně	NE
Možnosti přehrávání zvuku		Tones + WAV	Tones + WAV	Tones + WAV	Tones + WAV
MOŽNOSTI JAZYKA	Proměnné	?	ANO	ANO	ANO
	Použití „stringů“	NE	ANO	NE	ANO
	Trigonometrické funkce	NE	ANO	NE	?
	Používání ‘switch’	ANO	ANO	?	ANO
	Použití polí	NE	ANO	ANO	ANO
BLUETOOTH	Podporovaná zařízení	PC, NXT	PC, NXT, GPS, jiné	PC, NXT	NXT, jiné
	Fantom protokol	ANO	ANO	ANO	NE
	Rychlost (duplex)	HALF	FULL	HALF	FULL

3 UMĚLÁ INTELIGENCE

Umělá inteligence je obor informatiky zabývající se tvorbou strojů vykazujících známky inteligentního chování. Definice pojmu „inteligentní chování“ je stále předmětem diskuze, nejčastěji se jako etalon inteligence využívá lidský rozum.

[11]

3.1 Turingův test

Turingův test vytvořil Alan Turing v roce 1950. Jeho cílem je zjistit, jestli se zadaný systém umělé inteligence chová opravdu inteligentně.

Test probíhá tak, že do oddělených místností umístíme jednak testujícího (například člověka), jednak předmět zkoumání (počítač s příslušným programem). Testující poté klade otázky v přirozené řeči a předává je do druhé místnosti, kde je zodpoví buď počítač, nebo druhý člověk (kdo bude odpovídat, se rozhodne náhodně). Odpovědi jsou předávány zpět testujícímu člověku. Pokud testující není schopen rozpoznat, jestli komunikuje se strojem nebo s člověkem, pak program běžící v počítači splňuje tento test.

Turingův test byl dlouho považován za základní měřítko pro posuzování schopností umělé inteligence. Test ale zdaleka nepokrývá všechny aspekty, které od umělé inteligence očekáváme. Jeden z nedostatků tohoto testu ilustruje argument čínského pokoje.

V současné době není znám algoritmus, jehož odpovědi by byly zcela nerozlišitelné od člověka, takže by Turingův test zcela splňoval.

3.2 Argument čínského pokoje

Představme si uzavřenou místnost naplněnou velkým množstvím čínských textů, ve kterých se hypoteticky nalézá každá smysluplná věta tohoto jazyka. Do takového pokoje umístíme člověka, který čínštinu neovládá, ale má znalost, kde případně najít na základě předaného textu odpověď. Tomuto člověku budeme písemně předávat otázky (jako v Turingově testu), tento člověk je teoreticky schopen v této knihovně najít dostatek materiálu na to, aby nalezl výskyt dodané otázky a prostým opsáním části kontextu vytvořil smysluplnou odpověď, kterou předá ven. Vnější tazatel by se mohl domnívat, že člověk uvnitř pokoje čínštině bez problému rozumí, přestože ve skutečnosti tomu tak není, člověk

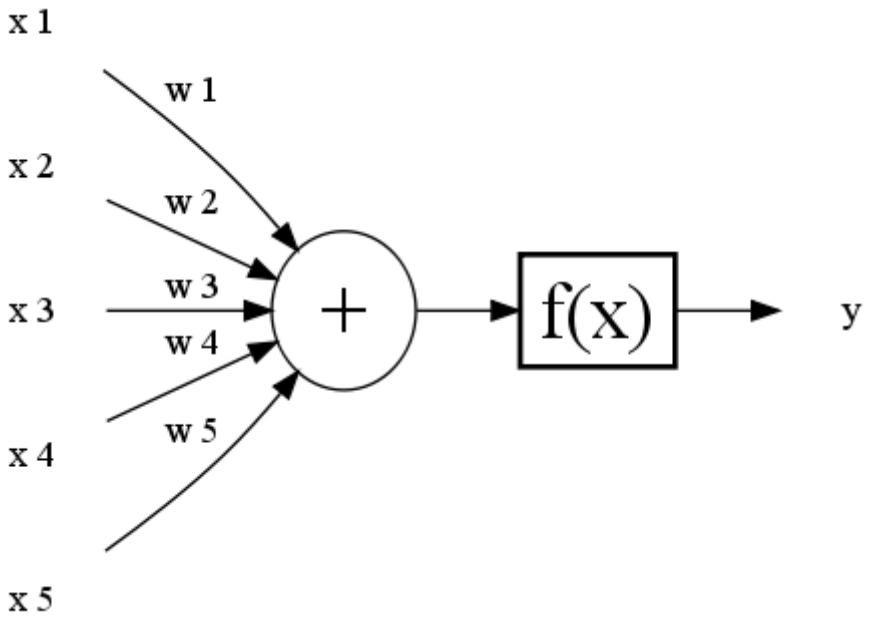
uvnitř pouze mechanicky pracuje s pro něj neznámými symboly, takže by jeho práci mohl zastat i zcela nemyslící stroj.

[12]

3.3 Neuronové sítě

Neuronová síť je jedním z výpočetních modelů používaných v umělé inteligenci. Jejím vzorem je chování odpovídajících biologických struktur. Umělá neuronová síť je struktura určená pro distribuované paralelní zpracování dat.

Skládá se z umělých (nebo také formálních) neuronů, jejichž předobrazem je biologický neuron. Neurony jsou vzájemně propojeny a navzájem si předávají signály a transformují je pomocí určitých přenosových funkcí. Neuron má libovolný počet vstupů, ale pouze jeden výstup.



Obrázek 9: Matematický model neuronu (perceptron)

Neuronová síť má asi jako každý přírodou inspirovaný přístup velmi široké možnosti uplatnění, především při řešení problémů z reálného světa. V této oblasti jsou totiž počítače nejslabší. Dále se přímo nabízí využití v umělé inteligenci. Konkrétní typ a architektura neuronové sítě se většinou silně upravuje na míru danému konkrétnímu problému.

Nejzajímavější vlastností neuronových sítí je jejich odolnost vůči chybám. Jakmile se neuronová síť jednou dobře naučí, je schopná řešit i příbuzné problémy, se kterými se dosud nesetkala. Na druhou stranu, není však garantovaná kvalita ani správnost řešení. Neoddělitelnou součástí výsledku je vždy určitá chyba.

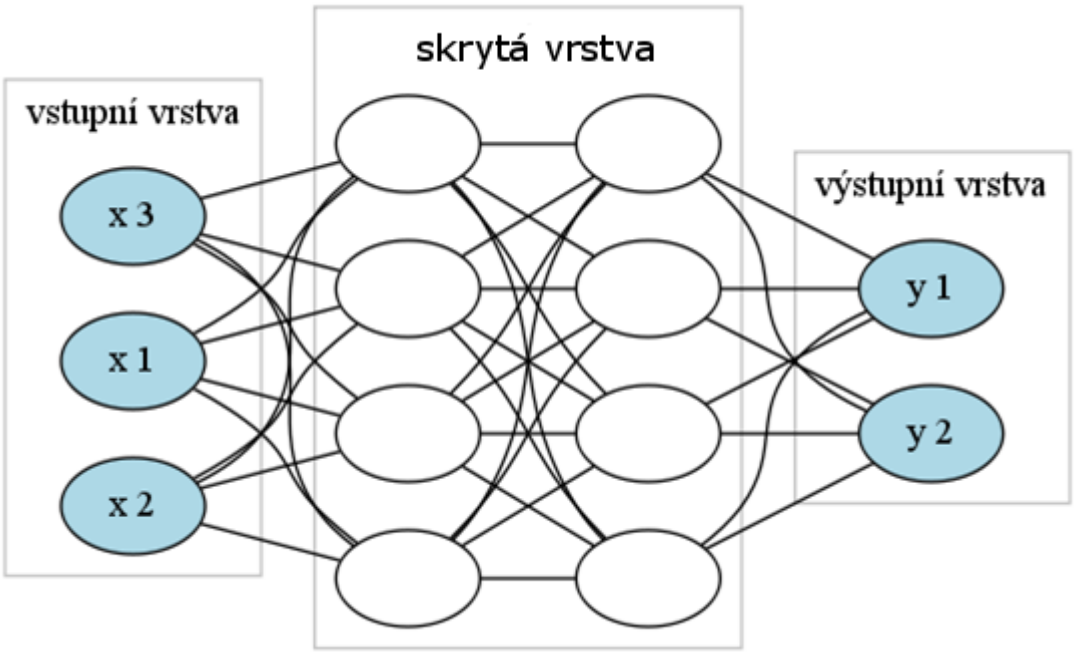
V těchto oblastech se neuronové sítě používají nejčastěji:

- klasifikace
- detekce pravidelnosti
- zpracování řeči
- zpracování obrazu
- optimizační problémy
- ovládání robotů
- predikce časových řad
- simulace

[13]

3.3.1 Vícevrstvý perceptron

Vícevrstvý perceptron (multilayer perceptron) je typ neuronové sítě, která se chová podobně jako jeden perceptron, ale skládá se z většího počtu umělých neuronů. Tyto neurony jsou rozděleny do vrstev. První vrstva, tzv. vstupní vrstva je vstupem neuronové sítě. Počet neuronů ve vstupní vrstvě odpovídá velikosti vstupního vektoru. Za ní následují tzv. skryté vrstvy. Těch může být i více, přičemž jejich počet i velikost je libovolná. Poslední skrytá vrstva je napojena na poslední, tzv. výstupní vrstvu. Počet neuronů ve výstupní vrstvě odpovídá velikosti výstupního vektoru. Všechny neurony jsou spojené pouze dopředně – směrem ze vstupu na výstup.



Obrázek 10: Vícevrstvý perceptron

Při učení se nejprve nastaví vstupní vektor jako výstup vstupní vrstvy. Poté je spuštěn výpočet výstupů všech neuronů, a to postupně od první skryté vrstvy po výstupní vrstvu. Následuje nastavení správného výstupu na výstupní vrstvu. Ta vypočítá chybu a zpětně ji rozšíří až do první skryté vrstvy. Poté jsou podle chyby upraveny jednotlivé váhy všech neuronů.

[14]

3.3.2 Učení neuronové sítě

Cílem učení neuronové sítě je nastavit síť tak, aby dávala přesné výsledky. V biologických sítích jsou zkušenosti uloženy v dendritech. V umělých neuronových sítích jsou zkušenosti uloženy v jejich matematickém ekvivalentu - váhách. Učení neuronové sítě rozlišujeme na učení s učitelem a učení bez učitele. Fáze učení neuronové sítě bývá nazývána adaptivní. Po naučení neuronové sítě je síť ve fázi vybavování.

Učení s učitelem

Podobně jako v biologických sítích je zde využita zpětná vazba. Neuronové síti je předložen vzor. Na základě aktuálního nastavení je zjištěn aktuální výsledek. Ten porovnáme s vyžadovaným výsledkem a určíme chybu. Poté spočítáme nutnou korekci (dle

typu neuronové sítě) a upravíme hodnoty vah či prahů, abychom snížili hodnotu chyby. Toto opakujeme až do dosažení námi stanovené minimální chyby. Poté je síť adaptována.

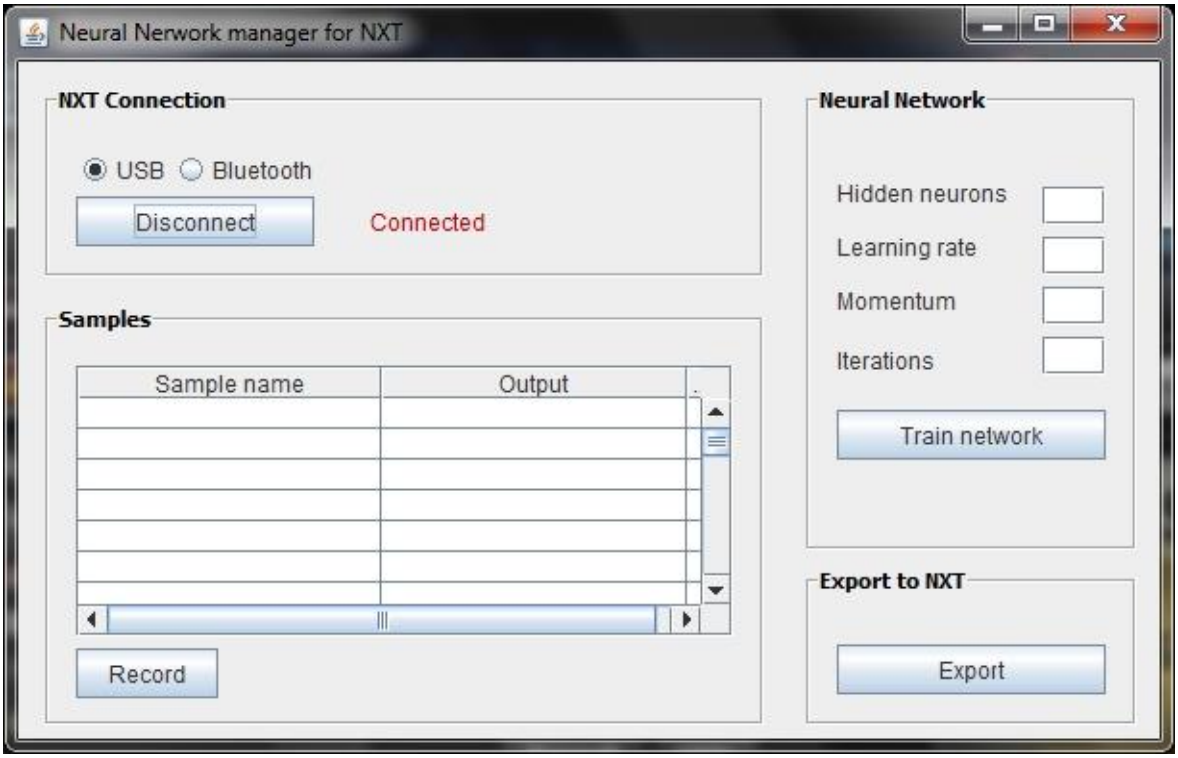
Učení bez učitele

Při učení bez učitele nevyhodnocujeme výstup. Při tomto učení nám výstup není znám. Síť dostává na vstup sadu vzorů, které si sama třídí. Buď si vzory třídí do skupin a reaguje na typického zástupce, nebo si přizpůsobí topologii vlastnostem vstupu. [13]

II. PRAKTICKÁ ČÁST

4 NEURAL NETWORK MANAŽER PRO LEGO MINDSTORMS

Neural Network manažer je grafické uživatelské rozhraní psané v jazyku Java, které umožňuje jednoduché vytváření, správu a trénování neuronových sítí, které můžou být exportovány jako program připravený pro běh v NXT robotech. Tento software vyvinuli Bram van der Vlist a Rick van de Westelaken. Systém se skládá ze dvou částí, rozhraní běžící přímo v řídící jednotce robota a software pro správu neuronových sítí v počítači.



Obrázek 11: Neural network manager

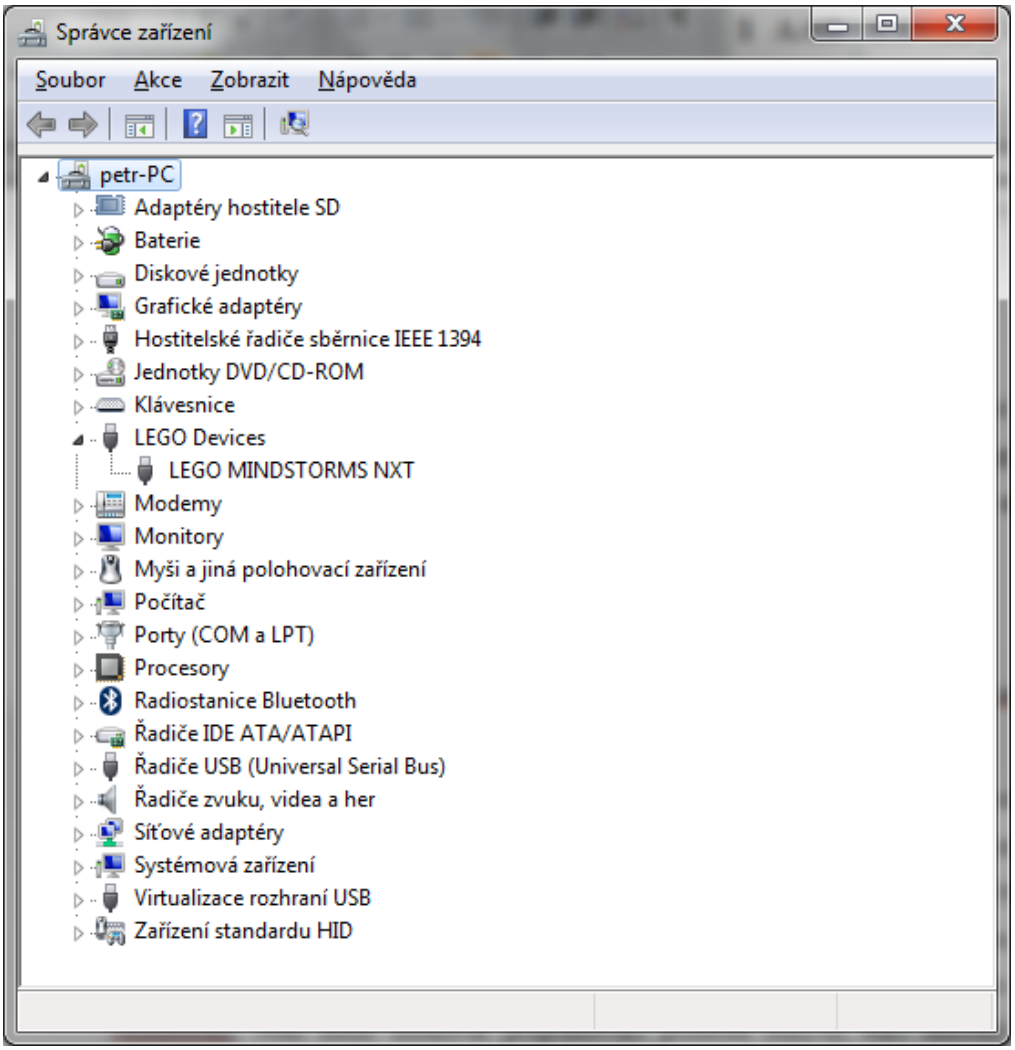
4.1 Instalace potřebného software

Java je více flexibilní než originální NXT-G software. Pro vývoj aplikací v Javě jsem zvolil programovací prostředí Eclipse, které umožňuje snadný vývoj a testování aplikací pro NXT. Eclipse je poskytován jako open source a za pomocí pluginů je možné ho snadno rozšířit o podporu dalších jazyků.

4.1.1 Ovladače a běhové prostředí

Nejprve je nutné nainstalovat Java SE (Standard Edition) JRE (Java Runtume Environment). Není nutné instalovat JDK (Java Developer Kit).

Aby bylo možné připojit NXT robota pomocí USB rozhraní, nebo Bluetooth rozhraní, je nutné nainstalovat ovladače. USB připojení je více stabilní a je jednodušší ho zprovoznit (u Bluetooth komunikace je možné narazit na problémy s kompatibilitou u některých výrobců Bluetooth hardware). Ovladače je nutné nainstalovat ještě před prvním propojením USB kabelu mezi NXT a počítačem. Tyto ovladače jsou dostupné na oficiálním webu Lego Mindstorms. Není nutné instalovat programovací prostředí NXT-G, stačí samotný USB driver. Pokud už máme ovladač nainstalovaný z originálního CD, je dobré zkontrolovat jeho aktuálnost. Po instalaci je nutné restartovat systém. Nyní už je možné připojit NXT robota, pro připojení se ve Správci zařízení zobrazí položka LEGO MINDSTORMS NXT.



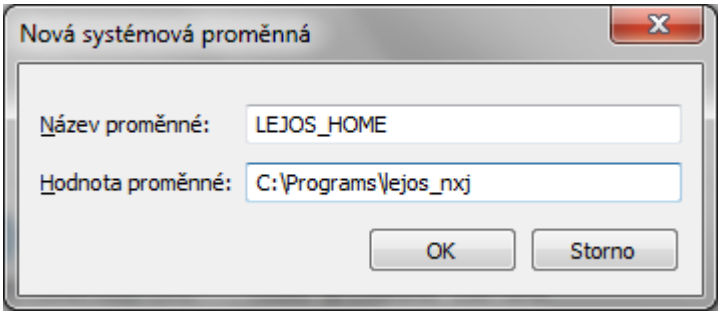
Obrázek 12: Správně nainstalovaný ovladač NXT

4.1.2 Lejos

Po stažení ZIP archivu Lejos ho musíme extrahovat. Je vhodné zvolit název adresáře, který neobsahuje znak mezery (použití adresáře s mezerou v názvu způsobuje v některých případech problémy, proto je nevhodné použít například adresář „Program Files“).

Aby běhové prostředí Java mohlo přistupovat k Lejos knihovnám, je nutné přidat v systému novou systémovou proměnnou. Ve Windows Seven pravým tlačítkem kliknutím na Počítač zvolením Vlastnosti, Upřesnit nastavení systému. Nyní na kartě Upřesnit zvolíme Proměnné prostředí. V sekci Systémové proměnné přidáme novou položku.

Název proměnné je LEJOS_HOME, hodnota proměnné je cesta k Lejos knihovnám (takže například C:\Programs\lejos_nxj).



Obrázek 13: Vytvoření nové systémové proměnné

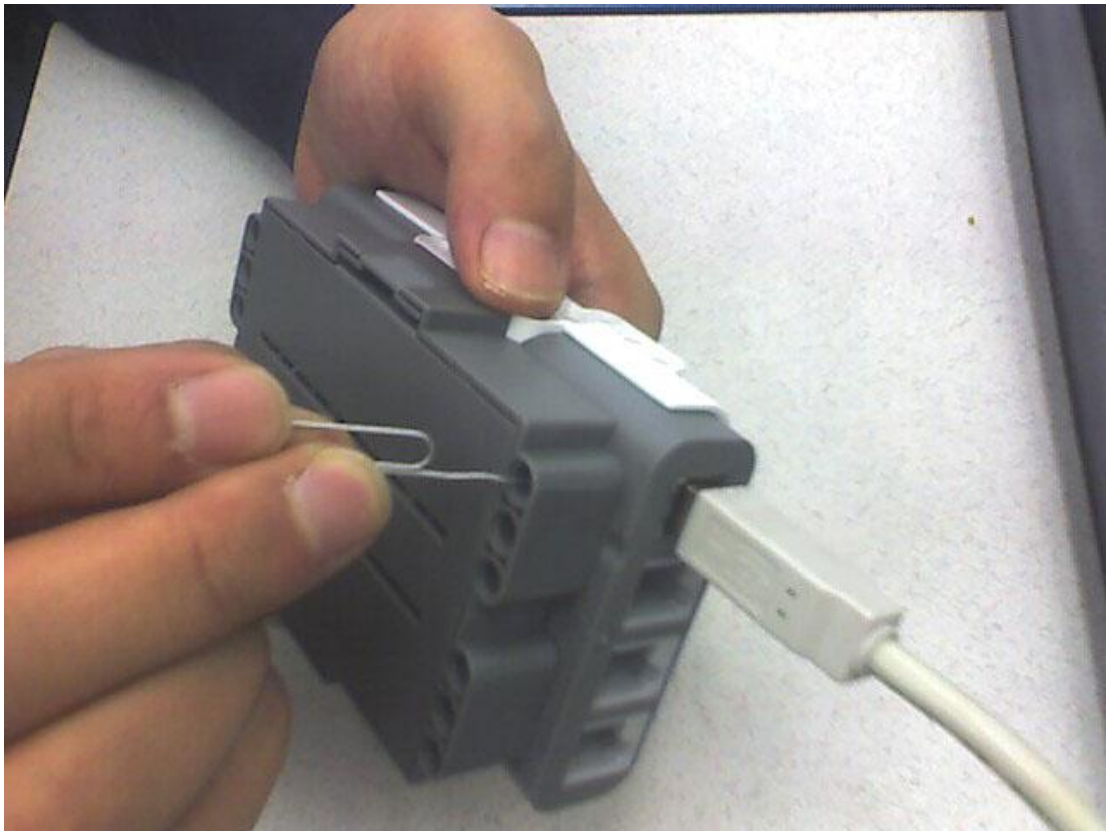
Do systémové proměnné Path se ještě musí přidat hodnota %LEJOS_HOME%\bin (jednotlivé hodnoty se oddělují středníkem).

Správnost nastavení je možné otestovat tím, že do příkazového řádku napíšeme příkaz lejosdl, který by se měl spustit i pokud se nenacházíme v adresáři Lejos. Pokud je vypsána chyba o neexistenci příkazu je něco nastaveno špatně. V tom případě musíme smazat vytvořené proměnné a zkusit je nastavit znovu.

4.1.3 Instalace Lejos firmware do řídící jednotky robota

Originální firmware NXT musíme nahradit Lejos firmwarem. Firmware funguje jako jednoduchý operační systém v NXT jednotce, implementuje Java Virtual Machine pro spoštění programů napsaných v jazyce Java.

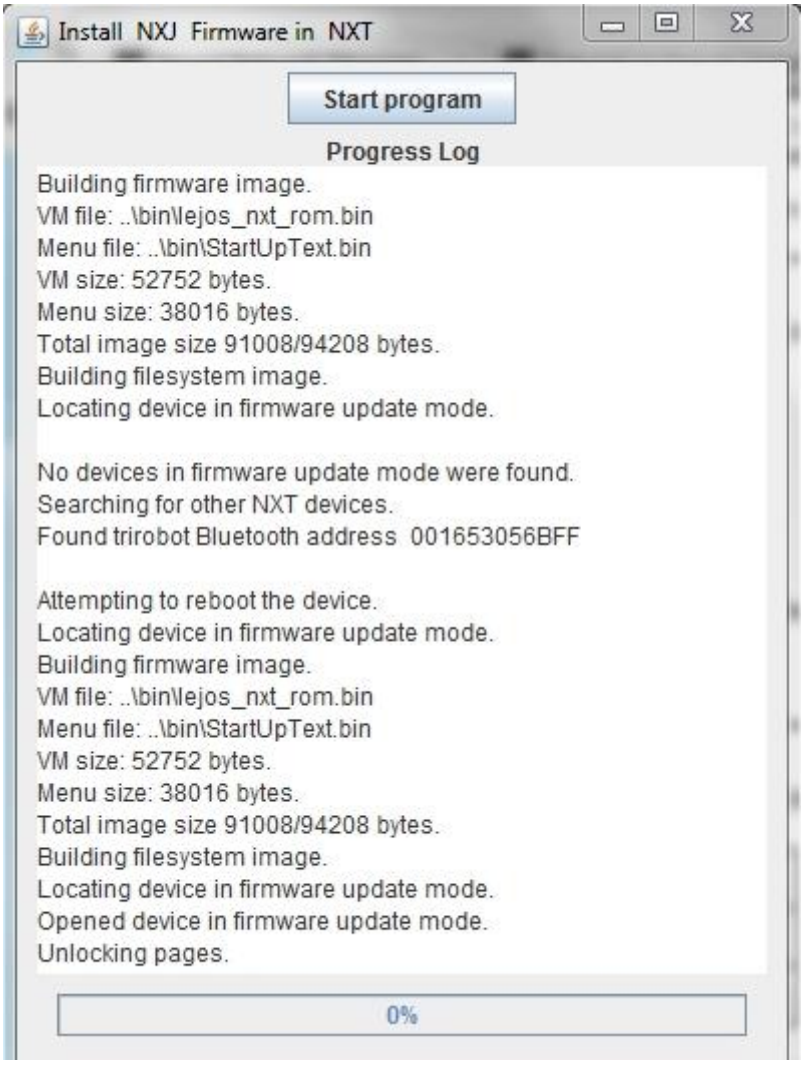
Před instalací zkontrolujeme, jestli je řídící jednotka připojena k PC a správně rozpoznána v operačním systému. Jako první krok instalace musíme přepnout jednotku do upload firmware módu.



Obrázek 14: Upload firmware

Tenkým předmětem podržíme tlačítko skryté uvnitř první upevňovací díry pod USB konektorem. Řídící jednotka by měla signalizovat přepnutí kolísavým přehráváním zvuku.

Nyní můžeme v příkazovém řádku spustit příkaz „lejosfirmdl“.



Obrázek 15: Aktualizace firmware

Po uploadu firmware do řídící jednotky by mělo dojít k jejímu restartu. Při nabíhání systému by se už mělo zobrazit logo Lejos.

Obrázek 16: Nově nainstalovaný Lejos firmware

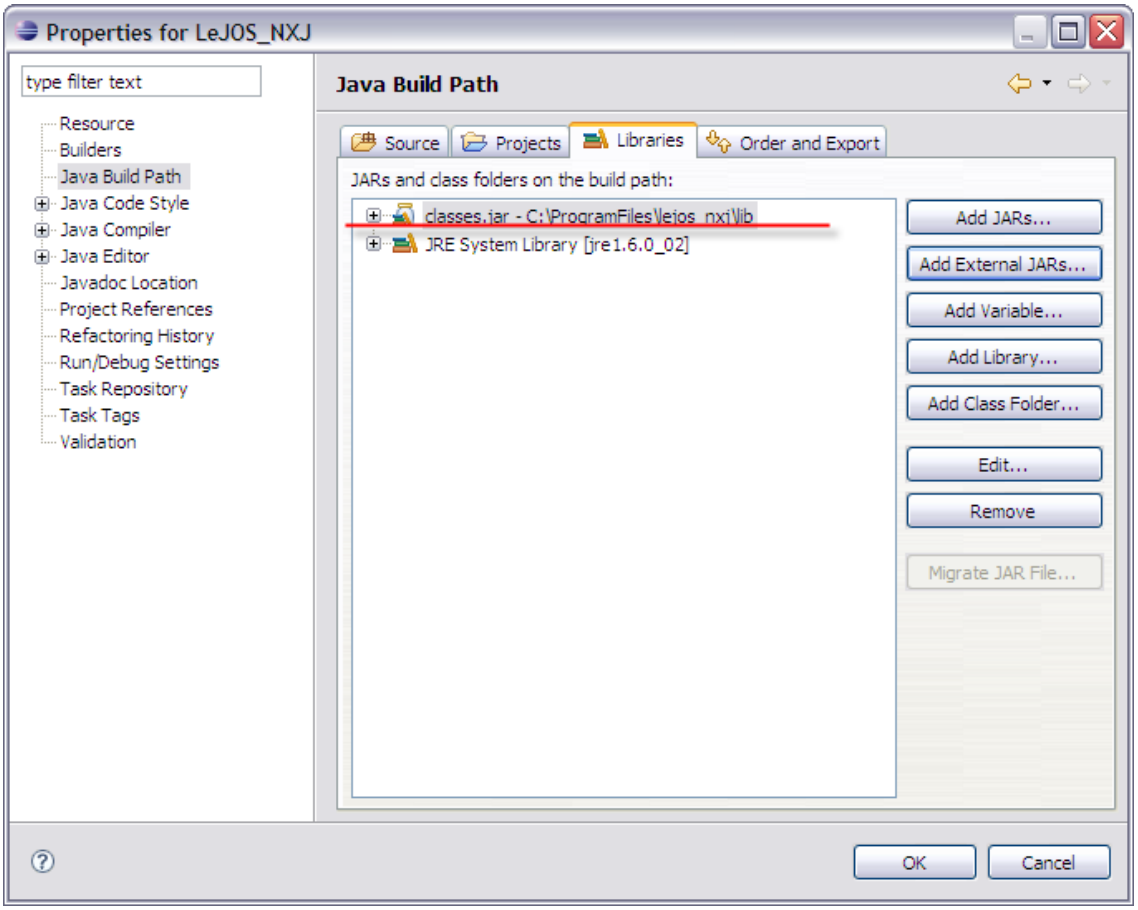
4.1.4 Instalace Eclipse

Jako vývojové prostředí použijeme Eclipse. Tento editor je sám naprogramován v jazyce Java. Balíček neobsahuje instalátor, stačí ho jednoduše extrahovat do vybraného adresáře.

Po spuštění Eclipse musíme nastavit takzvaný workspace. Do workspace se ukládají všechny vytvořené soubory. Jeho název by opět neměl obsahovat znak mezera.

4.1.5 Vytvoření a nastavení projektu

V nastavení nového projektu (File – New – Java Project) musíme přidat Lejos knihovny. V sekci Java Build Path – Libraries – Add External JARs přidáme soubor classes.jar který je v balíčku Lejos v adresáři lib.



Obrázek 17: Nastavení projektu v Eclipse

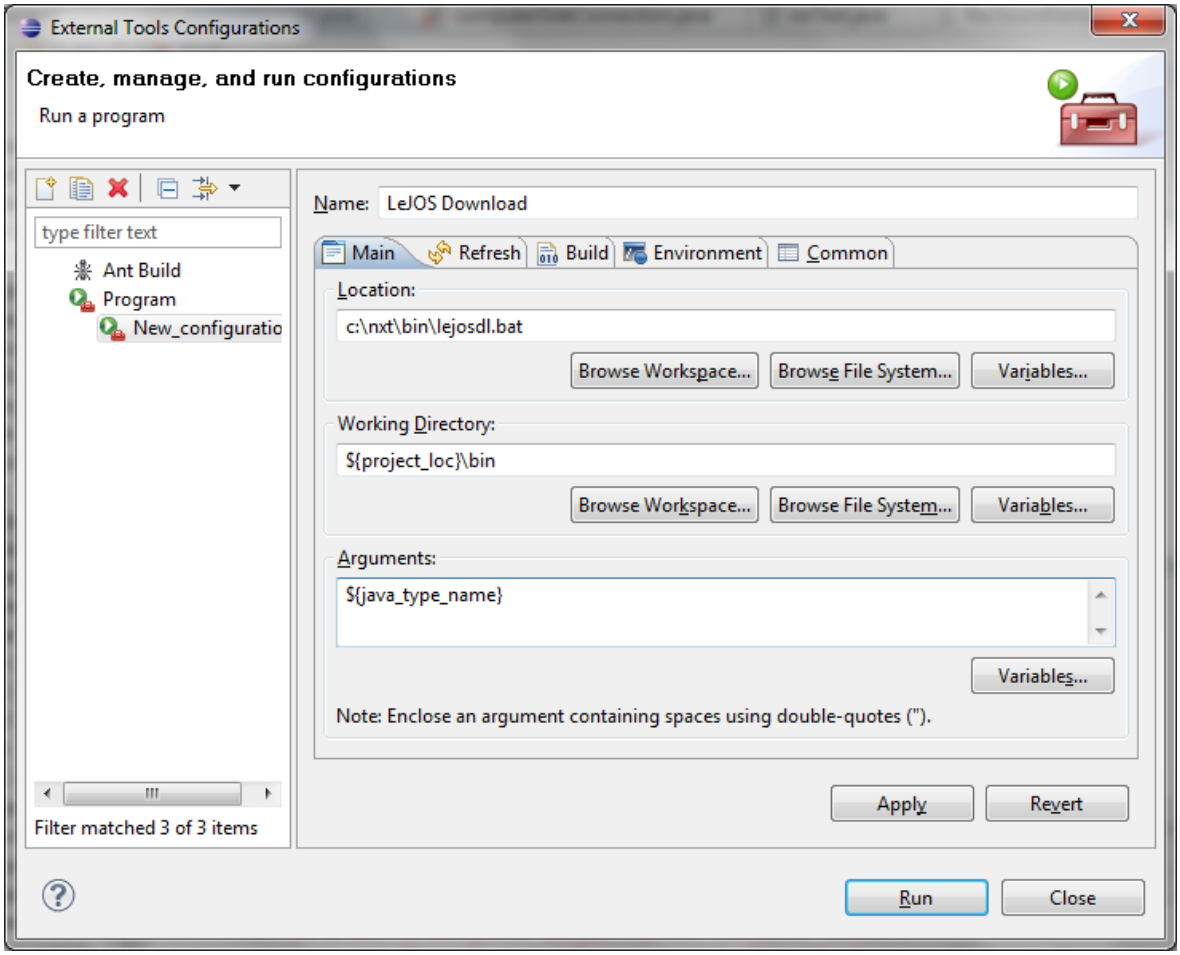
V sekci Java Compiler je nutné zaškrtnout možnost „Enable project specific settings“ a Compiler compliance level nastavit na hodnotu 1.3. Toto nastavení je nutné použít kvůli optimalizaci pro starší verze Javy, které jsou pro programování NXT robotů vhodnější (vyžadují menší množství systémových prostředků).

4.1.6 Nahrávání programů do řídící jednotky

Aby bylo možné nahrávat vytvořené programy do řídící jednotky robota přímo v Eclipse, je nutné nastavit ve vývojovém prostředí externí skript, který nám tuto funkcionalitu nabídne.

V menu Eclipse zvolíme Run – External Tools – Open External Tools Dialog zvolíme možnost New. V tomto dialogu je důležité nastavit čtyři parametry

- Name – identifikační název, například „lejos download“
- Location – cesta k souboru lejosdl.bat (tento soubor je součástí Lejos knihoven)
- Working directory – vložíme hodnotu ${project_loc}\bin
- Arguments – nastavíme ${java_type_name}



Obrázek 18: Nastavení Eclipse

Pro rychlé spouštění této funkce si ještě přidáme zástupce do položky Run. Zvolíme Run – Organize Favorites – Add. V tomto dialogu by měl už být zobrazen název zvolený v předchozím kroku, takže stačí zaškrtnout Checkbox a potvrdit OK.

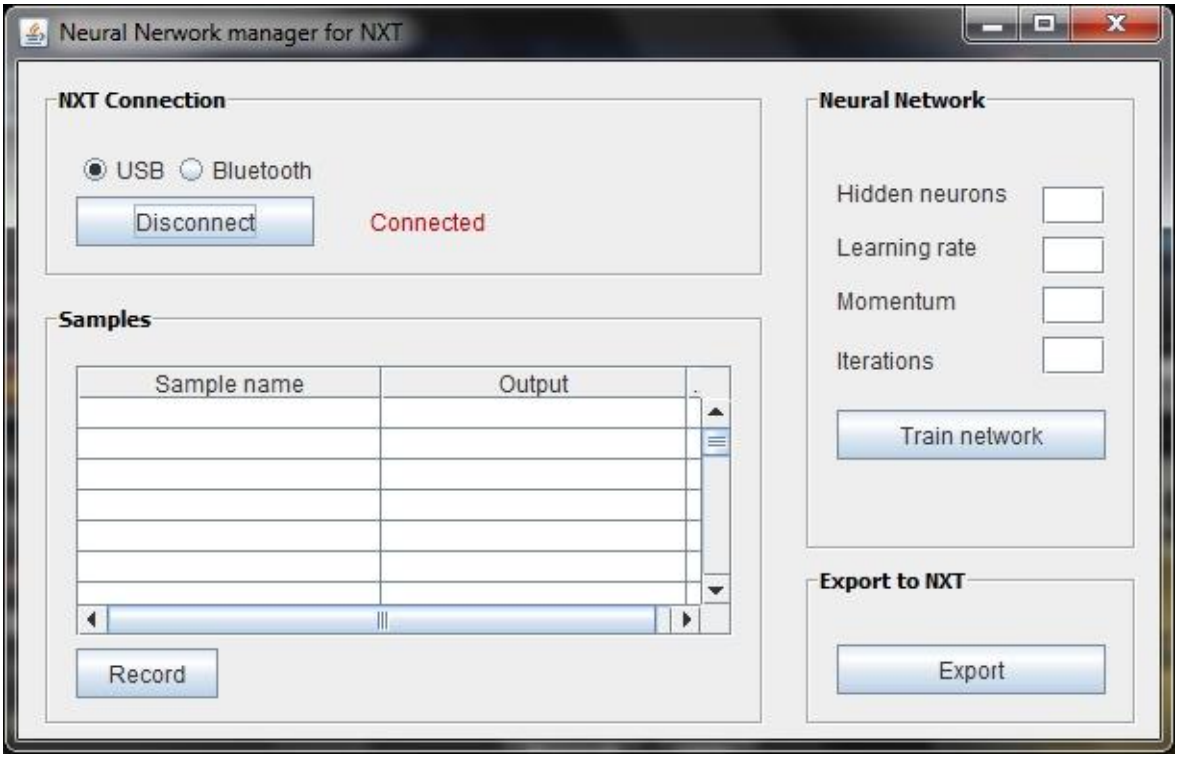
4.2 Balíčky LeJOS tříd určených pro NXT

- lejos.nxt – zpřístupnění senzorů, motorů atd.
- lejos.nxt.addon – ovládání dalších senzorů NXT které nejsou dodávány společně s NXT
- lejos.nxt.comm – třídy zajišťující komunikaci NXT
- lejos.nxt.debug – ladící funkce
- lejos.nxt.rcxcomm – emulace RCX tříd
- lejos.nxt.remote – ovládání pomocí Bluetooth
- lejos.robotics – abstraktní rozhraní pro robotics balíčky
- lejos.robotics.localization – podpora lokalizace
- lejos.robotics.mapping
- lejos.robotics.navigation
- lejos.robotics.proposal
- lejos.robotics.subsumption
- lejos.util – další pomocné třídy
- lejos.io – podpora pro java.io

5 POUŽITÍ NEURAL NETWORK MANAŽERU

Před použitím Neural Network manažeru v řídící jednotce NXT musíme mít zkompilované a nahrané do robota třídy RecSoundSamples a PreProcess (je také možné je přidat jako součásti projektu v Eclipse a poté budou nahrány do NXT současně se samotným projektem).

Řídící jednotka NXT musí být připojena k PC. Na PC spustíme samotný Neural Network manager. V této aplikací musíme zvolit způsob připojení. USB nebo Bluetooth (při použití Bluetooth musíme vyplnit MAC adresu NXT jednotky). Po kliknutí na tlačítko Connect dojde k připojení.



Obrázek 19: Připojená řídící jednotka NXT

5.1 Nahrání audio vzorků

Po úspěšném propojení NXT a PC můžeme začít nahrávat audio vzorky. Zvukový senzor připojíme do portu 1. Nahrávání zvuků je vhodné provádět v prostředí bez nežádoucích rušivých zvuků v pozadí.

5.1.1 Zvukový senzor

Lidské vnímání zvuku je limitováno v intervalu frekvencí 20 Hz až 20 kHz. V průběhu života dochází k jeho zhoršení vlivem opotřebení sluchových receptorů (klesá schopnost vnímat vysoké frekvence). Zvuk, který má frekvenci nižší než 20 Hz označujeme jako infrazvuk, frekvence vyšší než 20 kHz nazýváme ultrazvukem.

Princip snímání zvuku senzorem vychází s přeměny mechanických kmitů akustického vlnění na elektrický signál. Jednotkou pro měření intenzity zvukového signálu je dB (decibel). Decibel je jednotka s logaritmickou mírou (lidský sluch vnímá zvuk logaritmicky k jejich intenzitě).

Obrázek 20: Zvukový senzor NXT

Vztah mezi intenzitou zvuku, vnímání člověkem a hodnotou zobrazovanou senzorem NXT uvádí následující tabulka.

Tabulka 2: Vnímání intenzity zvuku

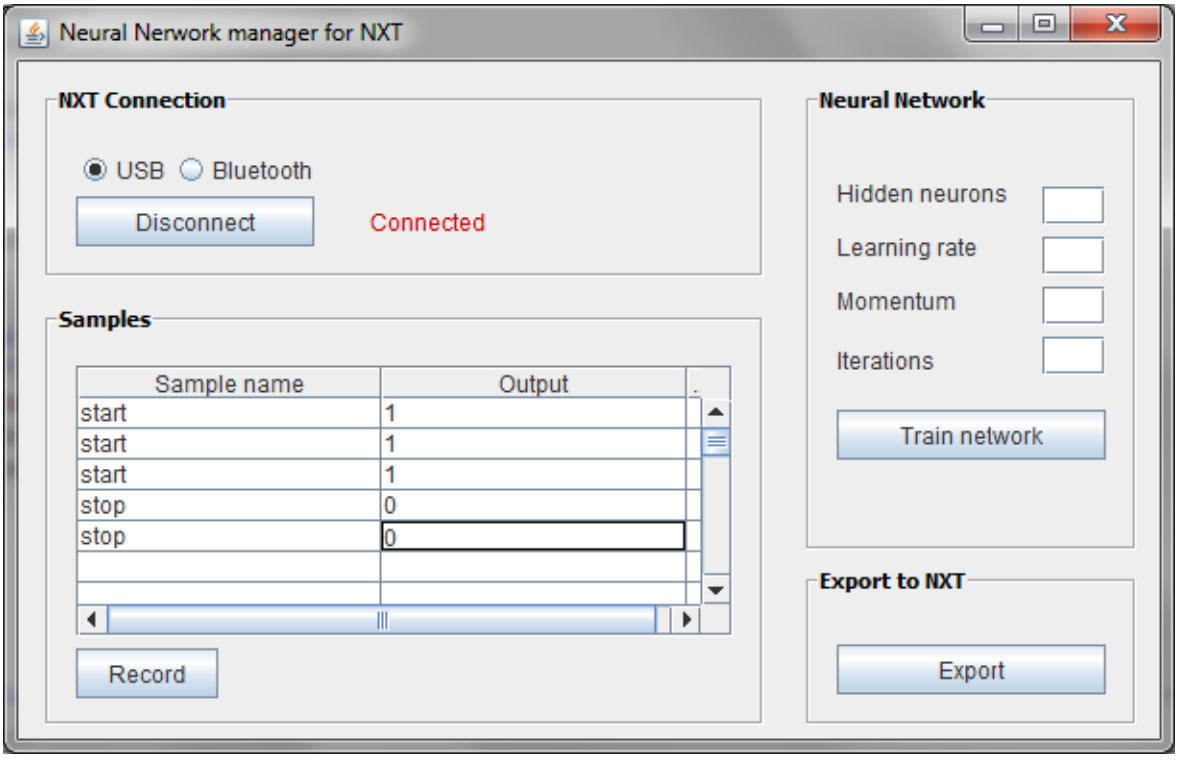
dB	vnímání člověkem	hodnota senzoru
10	práh slyšitelnosti	0 %
20	ticho	
30	bezvětří	4 – 5 %
40	tlumený hovor	
50	tichá pracovna	
60	běžný hovor	10 – 30 %

70	mírný hluk, poslech televize	
80	velmi silná reprodukovaná hudba	30 – 100 %
90	silný hluk, jedoucí vlak	
100	sbíječka	
110	živá rocková hudba	
120	startující proudové letadlo	
130	práh bolestivosti	
140	akustické trauma	

Nahrávání jednotlivých vzorků provedeme tlačítkem Record (k zastavení nahrávky slouží tlačítko Stop).

Pro lepší orientaci je vhodné jednotlivé vzorky výstižně pojmenovat. U jednotlivých povelů nastavíme výstupní hodnoty. Například pokud budeme nahrávat 2 vzorky „start“ a „stop“, pro „start“ nastavíme výstupní hodnotu 1 a pro „stop“ nastavíme hodnotu 0.

Je vhodné pro každý příkaz uložit více vzorků. Výstupní hodnoty nastavujeme u stejných povelu shodné (všechny povely „start“ budou mít výstupní hodnotu 1).



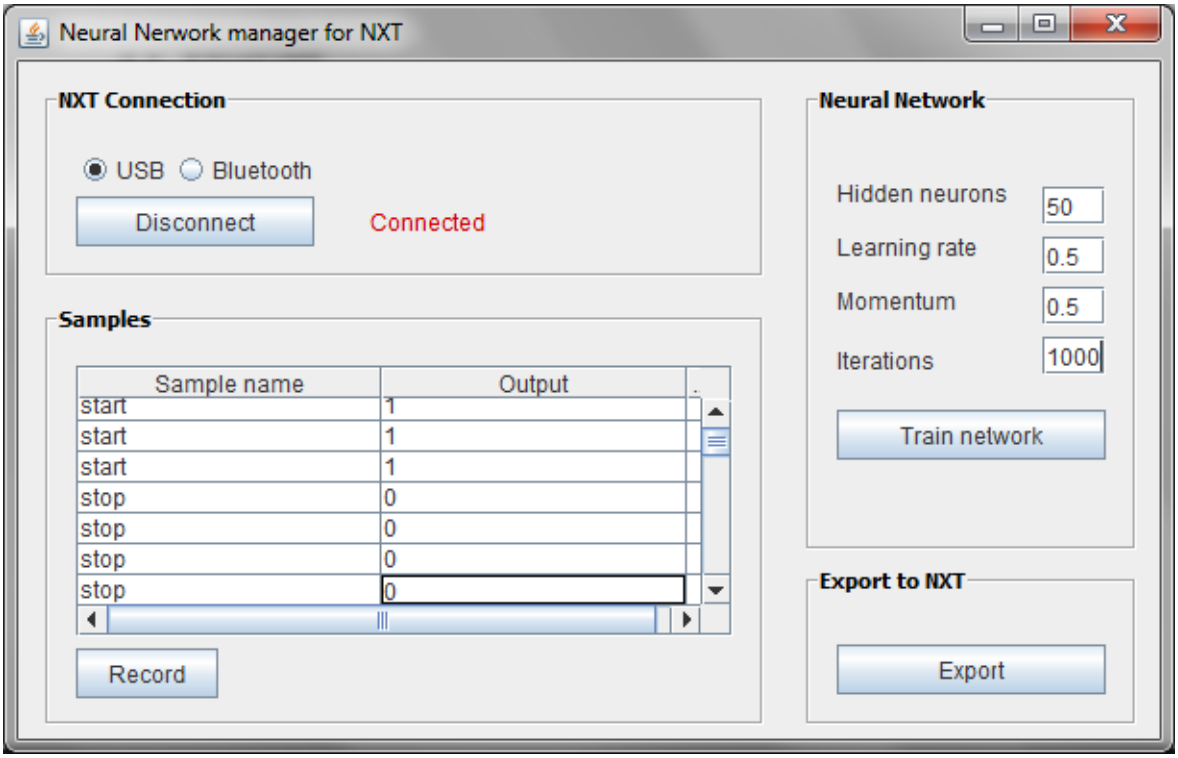
Obrázek 21: Nahrávání zvuků

5.2 Učení sítě

Jakmile máme vstupní vzorky nahrány a uloženy s přiřazenými výstupními hodnotami, můžeme začít s učením sítě.

5.2.1 Nastavení neuronové sítě

- nastavíme počet neuronů ve skryté vrstvě (minimum je počet neuronů ve vstupní vrstvě, v tomto případě je použito 50 neuronů)
- nastavíme hodnotu Learning rate na hodnotu od 0 do 1 (výchozí hodnota je 0.5)
- nastavíme hodnotu Momentum (hybnost)
- nastavíme počet iterací (výchozí hodnota je 1000)



Obrázek 22: Nastavení neuronové sítě

Po stisku tlačítka „Train network“ dojde k učení sítě. Program nám vypíše hodnotu přesnosti. Pokud se tato hodnota nebude blížit k 100, mělo by dojít k novému učení sítě s více iteracemi.

5.3 Export neuronové sítě do NXT

Jakmile je neuronová síť naučená, může být exportována jako Java třída. Pro export slouží tlačítko Export. Neural Network manažer nyní automaticky vygeneruje soubor nn.java.

Tuto třídu už můžeme zkompilovat a nahrát do NXT.

5.4 Metody třídy nn.java

5.4.1 calcHiddenValues

public void calcHiddenValues()

- vypočítá hodnoty neuronů ve skryté vrstvě

5.4.2 calcOutputValues

public void calcOutputValues()

- vypočítá hodnoty výstupních neuronů

5.4.3 outputValue

public int outputValue()

- vrací výstupní hodnotu

5.4.4 getVoiceCommand

public int getVoiceCommand(int[] rawInput)

- vrací hodnotu odpovídající vzorku nastavenou při učení sítě

5.5 Ukázková aplikace

Samotná třída jj.java nahraná v řídící jednotce nezajišťuje žádnou další funkcionalitu. Je nutné ještě napsat aplikaci, která se bude starat o obsluhu senzorů (v tomto případě vstupu ze zvukového senzoru) a vykonávání následné činnosti na základě výstupu neuronové sítě.

```
while (true){
    int i=0;
    while (recording == true){
        recordedSamples[i] = sound.readValue();
        i++;
        try {Thread.sleep(30);}catch (Exception e){}
    }
    if (i > 0)
    {
        int[] toNN = new int[i];
        for (int j =0;j<i;j++){
            toNN[j]=recordedSamples[j];
            LCD.clear();
            LCD.drawInt(netwerk.getVoiceCommand(toNN), 1, 1);
            LCD.refresh();
        }
    }
    try {Thread.sleep(100);}catch (Exception e){}
}
```

Obrázek 23: Ukázka aplikace

V tomto případě je výstupní hodnota neuronové sítě pouze vypsána na display. Podle potřeby už není problém rozšířit funkcionalitu například o řízení motorů robota. V tomto případě je důležité brát ohled na aktuální konstrukci, aby nedošlo k poškození samotného robota.

5.6 Zpracování dat

V případě že budeme chtít zpracovávat vstupní data i z jiných senzorů, než je zvukový, musíme předzpracovat vstupní data pro použití v neuronové síti.

Počet zaznamenaných hodnot v rámci jednoho vzorku musí odpovídat počtu neuronů ve vstupní vrstvě neuronové sítě. Počet vstupních neuronů v tomto případě je 50, ale může být upraven změnou proměnné sampleNumber (třída PreProcess.java).

Prvním krokem předzpracování dat je tedy získat 50 hodnot. Rozdíl mezi počtem hodnot získaných ze senzoru a požadovaným počtem můžeme upravit dvěma způsoby.

- pokud je počet získaných hodnot nižší než požadovaný, nastavíme zbývající hodnoty na 0
- pokud máme více hodnot, nadbytečné hodnoty můžeme smazat, nebo vhodným způsobem zkrátit na požadovanou hodnotu

Dalším krokem pro správné zpracování hodnot neuronovou sítí je jejich znormování v rozmezí od 0 do 1 (nejvyšší hodnota bude reprezentována číslem 1). Toto normování je důležité ze dvou důvodů. Odstraníme rozdíly způsobené rozdílným měřením (například rozdíl hlasitosti při získávání jednotlivých nahrávek), také je díky tomu možné použít stejnou neuronovou síť pro různé typy senzorů.

6 ŘÍZENÍ MINDSTORMS NXT POMOCÍ MATLABU

Matlab se stal celosvětovým standardem v oblasti vědeckých a technických výpočtů. Program samotný poskytuje kvalitní výpočetní a grafické nástroje, ale i rozsáhlé specializované knihovny funkcí.

Zajímavou možností ovládání NXT robotů je Matlab vybavený RWTH toolboxem a Neural Network toolboxem. RWTH toolbox slouží k získání informací ze senzorů NXT robota, pomocí funkcí matlabu tato data zpracujeme do požadovaného formátu. Tato data je možné použít jako vstupní data pro některou z neuronových sítí poskytovanou Neural Network toolboxem.

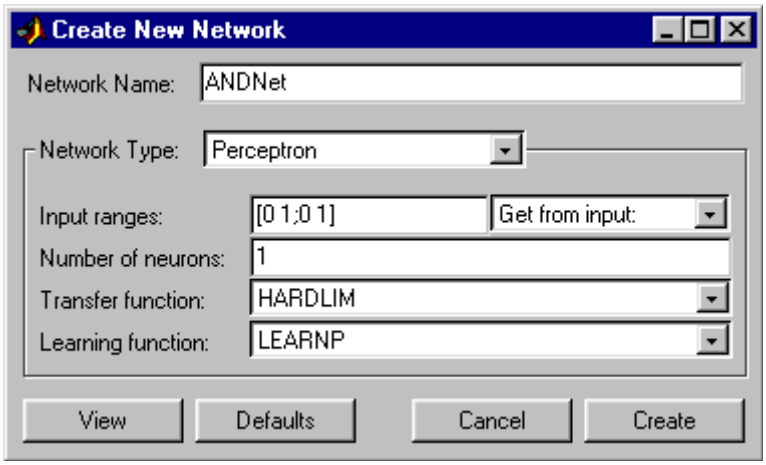
6.1 RWTH toolbox

RWTH toolbox poskytuje nástroje pro komunikaci s NXT roboty, řízení motorů, snímání dat ze senzorů ale má několik odlišností ve srovnání s ostatními jazyky použitelnými pro programování NXT.

- možnost ovládání několika robotů současně pomocí Bluetooth (je nutné použít více Bluetooth adaptérů)
- zvýšení funkcionality a prakticky neomezená velikost programu
- využití periferií připojovaných k PC (joysticky, webové kamery)
- pokročilá podpora ladění
- jednoduchá možnost vizualizace
- jednoduché propojení s dalšími toolboxy (nejenom neuronové sítě, ale i další knihovny jako jsou databázove komponenty, nebo knihovny pro zpracování obrazu)
- možnost návrhu aplikací a formulářů metodou drag and drop
- stabilita a ověřený běh toolboxů v Matlabu
- velká uživatelská základna (výhoda při řešení problémů)

6.2 Neural Network toolbox

Tento toolbox poskytuje nástroje pro návrh, tvorbu, trénování a simulaci neuronových sítí. Nabízí jednoduché uživatelské rozhraní pro návrh a práci s neuronovými sítěmi. Obsahuje také funkce pro automatické vytváření simulačních bloků neuronových sítí do Simulinku.



Obrázek 24: Vytvoření nové sítě

ZÁVĚR

Problematika řízení MINDSTORMS NXT robotů je dnes realizovatelná ve velkém množství programovacích jazyků. Implementace řízení ve všech jazycích není nijak obtížná, především díky dostupnosti vývojových prostředí a také díky různým komunitám, které neustále pracují na vývoji dalších verzí knihoven, rozšiřování podpory pro různé periferní zařízení, případně opravách objevených chyb.

Teoretická část popisuje hardwarové možnosti NXT robotů a možnosti komunikace s okolím. V druhé polovině teoretické části jsem se věnoval možnostem programování v různých vývojových prostředích a programovacích jazycích. Uvedený výčet možností programování není kompletní, díky otevřenosti NXT architektury jsou dostupné mnohé další možnosti programování. Každý ať už začínající, nebo pokročilý programátor si určitě najde jazyk, který mu bude nejvíce vyhovovat.

V praktické části jsem se věnoval programování robotů pomocí neuronových sítí, napsaných v jazyce Java s podporou knihoven a firmware LeJOS. Je uveden popis potřebného software, instalace, nastavení a potřebná konfigurace. Popsány jsou jednotlivé kroky při získávání vzorků, nastavení neuronové sítě, fáze učení a poté využití v ovládacím programu běžícím v řídící jednotce NXT.

Po zkušenostech získaných při programování NXT robotů je vidím jako vhodný prostředek pro vstup do světa robotiky. Při programování těchto robotů jsme omezeni jejich hardwarovou konstrukcí a omezeným možnostem počtu připojených periferií. Naopak z pohledu programování se nabízí velký počet možností vývoje, nezávisle na použité platformě.

ZÁVĚR V ANGLIČTINĚ

Problems of Mindstorms NXT robot is now feasible in a large number of programming languages. Implementation of management in all languages is not difficult, especially with the availability of development environments and also because of the different communities are constantly working to develop additional versions of libraries, expansion of support for various peripherals, or corrections of errors discovered.

The theoretical part describes the hardware options NXT robot and posibilities of comunication. In the second half of the theoretical part, I worked in a variety of programming opportunities for development environments and programming languages. That list is not complete programming choices, thanks to the openness of the NXT architecture are available to many more programming options. Everyone either beginners or advanced programmer can find a language that is most suited for him.

The practical section is devoted to programming robots using neural networks, written in Java with support for libraries and firmware LeJOS. A description of the required software, installation, setup and configuration required. Described are the steps of obtaining samples, set the neural network learning phase and then use the control program running in the NXT control unit.

Following the experience gained in programming NXT robots I see as an expedient to enter the world of robotics. When programming these robots are limited by their hardware design and the limited possibilities of connected peripherals. In contrast, the view programming offers a large number of development options, regardless of the platform used.

SEZNAM POUŽITÉ LITERATURY

[1] 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.

[2] 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.

[3] 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.

[4] BAGNALL, Brian. Maximum LEGO NXT : Building Robots with Java Brains. Editied by Sylvia Philipps. 1st edition. Canada : Variant Press, c2007. 505 s. ISBN 978-0-9738649-1-5.

[5] GURNEY, Kevin. An Introduction to Neural Networks. 1st edition. CRC Press, 1997. 234s. ISBN 978-1857285031.

[6] FREEMAN J. A.: Simulating Neural Networks with Mathematica, Adison-Weslez Publishing Company, 1994, ISBN 0-201-56629-X

[7] BOSE N.K., LIANG P.: Neural Network Fundamentals with Graphs, Algorithms, and Applications, McGraw-Hill Series in Electrical and Computer Engineering, ISBN 0-07-006618-3, 1996.

[8] ZELINKA I.: Umělá inteligence I. VUT Brno, ISBN 80-214-1163-5, 1998.

[9] Wikipedie [online]. 2010 [cit. 2010-06-03]. Matlab. Dostupné z WWW: <http://cs.wikipedia.org/wiki/MATLAB>.

[10] TMEJ, Bohuslav. Ovládání laboratorního modelu robota MindstormsNXT (machine robot) pomocí PC. Zlín, 2008. 81 s. Diplomová práce. UTB Zlín.

[11] Wikipedie [online]. 2010 [cit. 2010-06-03]. Umělá inteligence. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Umělá_inteligence>.

[12] Wikipedie [online]. 2010 [cit. 2010-06-03]. Argument čínského pokoje. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Argument_čínského_pokoje>.

[13] Wikipedie [online]. 2010 [cit. 2010-06-03]. Neuronová síť. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Neuronová_síť>.

[14] Digitální wiki [online]. 2010 [cit. 2010-06-03]. Neuronové sítě. Dostupné z WWW: <http://voho.cz/wiki/neuronove-site/>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

GPL General Public Licence.

JRE Java Runtime Environment.

JDK Java Developer Kit

LCD Liquid Crystal Display

PC Personal Computer

USB Universal Serial Bus

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

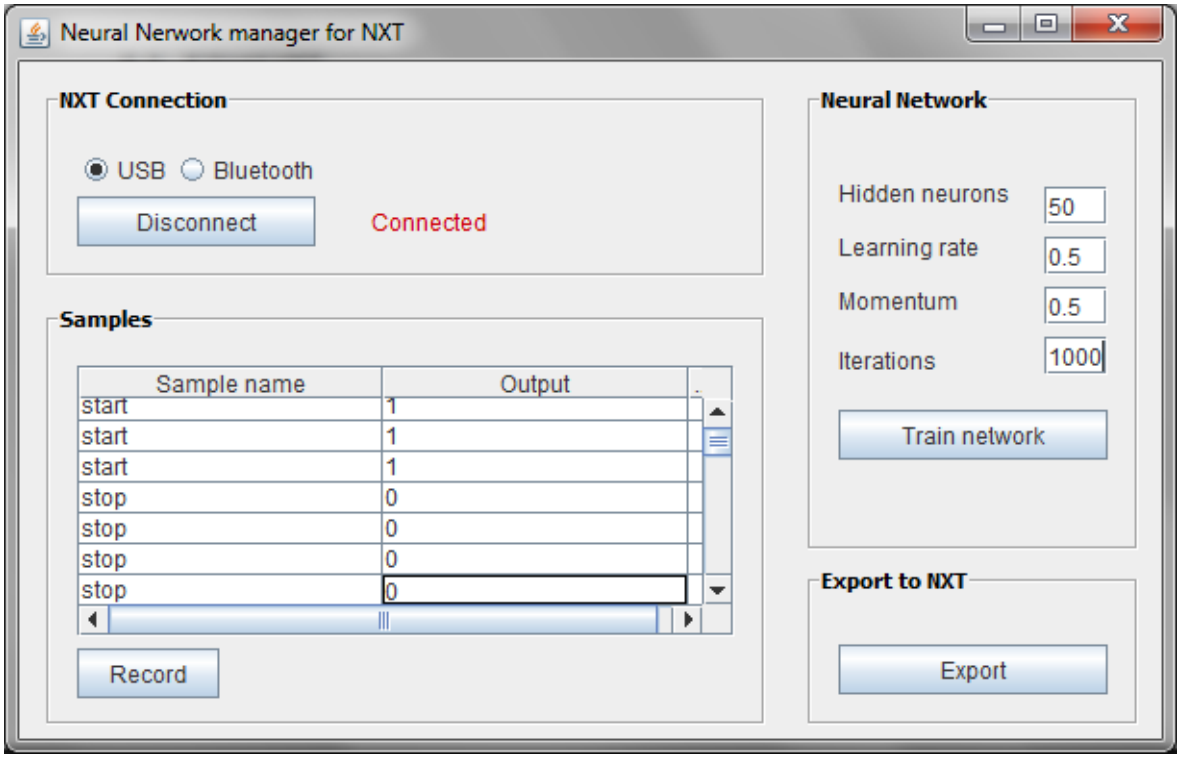
1. CD s elektronickou podobou diplomové práce
2. Zadání laboratorní úlohy 1

PŘÍLOHA P I: ZADÁNÍ LABORATORNÍ ÚLOHY 1

Úkol měření:

Ovládání robota NXT pomocí Neural Network manažeru.

- seznamte se s hardwarovými vlastnostmi robota NXT, funkcemi dodávaných senzorů a motorů
- seznamte se s fungováním neuronových sítí
- seznamte se s možnostmi použití alternativních firmware pro NXT roboty
- proveďte upgrade firmware, do řídící jednotky nahrejte LeJOS firmware
- seznamte se s ovládáním Neural Network Manageru



- pomocí Neural Network manageru proveďte propojení NXT robota a PC pomocí USB kabelu
- přes připojený zvukový senzor nahrajte několik vzorků
- proveďte učení sítě s vhodným nastavením parametrů
- výslednou síť exportujte pro spuštění v NXT jednotce
- vytvořte ovládací program pro NXT robota podle dokumentace k třídě nn.java

Seznam použitých přístrojů a součástek:

- inteligentní programovatelná kostka NXT
- zvukový senzor
- propojovací kabely
- počítač s vývojovým prostředím Eclipse