

PROGRAMOVANIE ROBOTOV LEGO V LINUXE

Bakalárska práca

IVAN JURIGA

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Katedra informatiky a riadenia procesov

Študijný program: Automatizácia, informatizácia a manažment
v chémii a potravinárstve

Vedúci práce: Ing. Michal Kvasnica, PhD

Dátum odovzdania práce: 21.05.2011

Dátum obhajoby: 11.07.2011

BRATISLAVA, máj 2011

(zadanie bakalárskej práce)

ČESTNÉ PREHLÁSENIE

Čestne prehlasujem, že na diplomovej práci som pracoval samostatne, na základe nadobudnutých poznatkov, skúseností a použitej literatúry.

V Bratislave, 21. mája 2011

Ivan Juriga

POĎAKOVANIE

Ďakujem vedúcemu bakalárskej práce Ing. Michalovi Kvasnicovi, PhD za odborné rady a metodické usmerňovanie. Ďakujem tiež konzultantovi Ing. Richardovi Valovi za konzultácie bakalárskej práce, zapožičanie stavebnice a pomoc pri zostavení modelu.

ABSTRAKT

JURIGA, Ivan. *Programovanie robotov LEGO v Linuxe*. (Bakalárska práca).

Slovenská technická univerzita v Bratislave, Fakulta chemickej a potravinárskej
technológie, Katedra informatiky a riadenia procesov.

Školiteľ: Ing. Michal Kvasnica, PhD.

Študijný odbor: Automatizácia, informatizácia a manažment v chémii a potravinárstve.

Dátum odovzdania práce: 2011-05-21 - Bratislava FCHPT STU 2011 - 42 s.

V bakalárskej práci je uvedený postup inštalácie softvéru pre programovú obsluhu jednotky NXT Intelligent Brick na stavbu robota LEGO Mindstorms v prostredí Linux. Práca na jednoduchom príklade predstavuje možnosti použitia slobodného programového vybavenia pri výuke rozličných technických odborov ako je napríklad, programovanie, algoritmizácia, kybernetika, matematika, fyzika, mechanika a podobne, jednoducho všade tam, kde je možná simulácia naučených teoretických poznatkov na modeli zostrojenom zo stavebnice LEGO. Všeobecné princípy použité v práci je možné aplikovať aj pri použití iných programovateľných jednotiek.

Kľúčové slová: Code::Blocks, LEGO, Linux, Mindstorms, NXB, NXC, NXT,
programovanie, slobodný softvér.

ABSTRACT

In this work describes the installation guide for a program the NXT Intelligent Brick to build LEGO Mindstorms robot in the Linux environment. Publication through a simple example is the possibility of using free software in teaching various technical fields such as programming, algorithms, cybernetics, mathematics, physics, mechanics, etc., in short wherever possible simulation of theoretical knowledge learned in the model constructed of brick LEGO. General principles used in the work can be applied even if other programmable units.

Keywords: Code::Blocks, LEGO, Linux, Mindstorms, NXB, NXC, NXT,
programming, open source.

PREDHovor

Motívom práce je propagácia Linuxu na školách. Linux je voľne šíriteľný operačný systém s otvoreným zdrojovým kódom (open source). Jeho existencia smeruje proti zažitým pravidlám vo svete počítačov: Vývoj zaisťuje voľne organizovaná skupina tisícov dobrovoľníkov na Internete v rámci nezištnej pomoci všetkým, ktorí majú o takúto pomoc záujem. Linux vnáša svetlo a nadšenie z objavovania vlastných schopností do sveta počítačov, navzdory snahe zahaľovať tento svet plášťom tajomstva niekoľkých veľkých korporácií.

Prečo ľudia majú radi Linux?

- *Linux je bezplatné programové vybavenie, je voľne šíriteľný a voľne modifikovateľný;*
- *Linux je rozšírený na mnohých hardwarových platformách od superrýchlych multiprocesorových systémov až po relatívne jednoduché mikročipové zariadenia, ako je napr. sieťový prepínač alebo tlačový server;*
- *Linux má nízke nároky na hardware, je rýchly a výkonný;*
- *Linux má dobrú podporu, vďaka čomu je jeho vylepšovanie pružnejšie ako u proprietárneho systému;*
- *Linux je bezpečný plnohodnotný operačný systém s množstvom užitočných aplikácií;*
- *Linux je prítiažlivý, užívateľovi okrem užitných vlastností pomáha svojou filozofiou formulovať zdravý životný postoj, vychádzajúci z nezištnej pomoci iným, z vďačnosti za bezplatne poskytnuté vedomosti a z radosti zo života, na zlepšovaní ktorého sa môže podieľať. Tieto vlastnosti v žiadnom inom operačnom systéme nenájdeme. Linux znamená oveľa viac, ako len hromadu bitov, ktoré vydolujeme z inštalačného CD.*

(Math Welsh, [1] strana xvii)

Linux vďačí za svoje rozšírenie predovšetkým dostupnosti zdrojového kódu. Pretože sú zdrojové kódy verejne prístupné, môže sa kdokoľvek zúčastňovať na ich vývoji, či už formou odstraňovania chýb alebo tvorbou kompletných aplikácií. Táto možnosť zlákala programátorov na celom svete, aby prispeli k vylepšeniu Linuxu, ktorý sa tak priblížil a v mnohom prekonal komerčné (proprietárne - vlastnené, pozn. autora) operačné systémy. Spolu s tisíckami aplikácií sa tak vytvorila počítačová platforma, použiteľná ako v nekomerčnej, tak aj v komerčnej sfére.

(Mark Mitchel, [2] s. 10)

OBSAH

Zoznam skratiek a symbolov	5
Zoznam ilustrácií a tabuliek	6
Úvod	7
1 Lego Mindstorms NXT	9
1.1 Programové vybavenie LEGO Mindstorms NXT	10
1.1.1 Programy pre operačný systém Windows	11
1.1.2 Programy pre operačný systém Linux	12
2 Programovanie LEGO Mindstorms NXT v Linuxe	14
2.1 Štruktúra programu NXC	14
2.1.1 Pravidlá písania programu	15
2.1.2 Úlohy	16
2.2 Inštalácia prekladača jazyka NXC v prostredí OS Linux	16
2.2.1 Inštalácia NBC z predkompilovaného balíka	17
2.2.2 Inštalácia NBC zo zdrojového kódu	17
2.3 Vytvorenie nového projektu	19
2.4 Prenos súborov medzi počítačom a jednotkou NXT	19
2.4.1 Prenos súborov pomocou LiNXT	20
2.4.2 Prenos súborov pomocou NBC	20
2.4.3 Prenos súborov pomocou T2N	20
2.5 Spustenie programu v jednotke NXT	23
2.6 Preprogramovanie firmvéru v jednotke NXT	23
2.7 Nastavenie NBC kompilátora vo vývojovom prostredí Code::Blocks	25
2.7.1 Postup pre začlenenie prekladača NBC do Code::Blocks IDE	26
2.7.2 Vytvorenie projektu	27
2.7.3 Kompilácia zdrojového súboru	27
2.7.4 Vytvorenie užívateľského nástroja	28
3 Ovládanie lego robota z aplikácie na PC	30
3.1 Komunikačný program pre riadiaci systém	30
3.2 Komunikačný program pre riadený systém	34
3.3 Ovládanie a monitorovanie robota z grafickej aplikácie	36
4 Záver	40
Zoznam bibliografických odkazov	41

ZOZNAM SKRATIEK A SYMBOLOV

- API - Application programming interface - programové rozhranie (knihnice) pre programovanie aplikácii. Obsahuje funkcie a triedy, ktoré užívateľ volá, namiesto programovania obsluhy zariadení na úrovni operačného systému;
- GCC - GNU compiler collection - súbor prekladačov programovacích jazykov zostavený a šírený pod licenciou GNU;
- G++ - GNU C++ compiler - prekladač jazyka C++ pre objektové programovanie;
- GDB - GNU project debugger - štandardný ladiaci program pre GCC. Obsahuje prostriedky pre krokovanie programov s výpisom obsahu premenných v jednotlivých krokoch;
- GNU - GNU not Unix - rekurzívne pomenovanie voľne šíriteľného softvéru Richardom Stallmanom v roku 1984;
- GPL - GNU General public license - licencia voľne šíriteľného programového vybavenia. Užívateľom zaisťuje slobodu pri používaní, kopírovaní, šírení a úprave programov vrátane ich zdrojových kódov;
- GUI - Grafic user interface - všeobecné pomenovanie pre grafické užívateľské rozhranie na rozdiel od textovej konzoly príkazového riadku;
- IDE - Integrated development environment - integrované vývojové prostredie, všeobecné označenie programov slúžiacich pre vývoj iných programov;
- NBC - Next byte codes - názov prekladača zjednodušeného jazyka „C“;
- NXC - Not exactly „C“ - názov zjednodušeného jazyka „C“;
- NXT - Next step from LEGO - ďalší vývojový krok po jednotke RCX;
- OS - Operating system - softvér (programy a dáta), ktorý obsluhuje hardvérové prostriedky počítača, volané funkciami aplikačného programu;
- PC - Personal computer - všeobecné označenie pre osobný počítač, môže to byť stolný počítač (desktop), vežový (tower) alebo notebook či netbook;
- T2N - Talk to NXT - komunikačný program na prenos súborov medzi PC a NXT;
- USB - Universal serial bus - špecifikácia univerzálneho sériového rozhrania na prenos údajov medzi počítačom a prídavnými zariadeniami pomocou kábla.

ZOZNAM ILUSTRÁCIÍ

Obr. 1: Bloková schéma mikropočítača NXT Brick	9
Obr. 2: Príklad použitia NXT Brick na stavbu vozidla	10
Obr. 3: Ukážka grafického jazyka NXT-G spoločnosti LEGO Group	11
Obr. 4: Ukážka programu LabView firmy National Instrument	12
Obr. 5 Nástroje NeXT	13
Obr. 6: Resetovanie jednotky NXT	24
Obr. 7: Vzhľad Code::Blocks IDE	25
Obr. 8: Okno editora nástrojov Code::Blocks	28
Obr. 9: Okno aplikácie bezdrôtového ovládania vozidla	36
Obr.10: Model vozidla	38

ÚVOD

„Len slabosi použijú na zálohovanie lokálne médium - skutoční muži uložia svoje dôležité veci na ftp, aby ich mohli používať celý svet.“

Linus Torvalds¹

V priebehu evolúcie človeka sa ruka v ruke s pochopením prírodných zákonitostí, zbieraním poznatkov a skúseností, rozvíjala aj veda a technika. Vývoj mikroelektroniky sa uberal cez skúmanie polovodičov, ich aplikovanie v diódach a tranzistoroch. Snaha o miniaturizáciu elektronických zariadení viedla k vytvoreniu prvých logických obvodov. Na základe výrokovej logiky vznikali čoraz zložitejšie obvody až po súčasné počítače, servery, ako aj zložité riadiace systémy, spoľahlivo vykonávajúce zložité algoritmy riadenia.

Automatizácia nachádza uplatnenie nielen tam, kde je práca pre človeka fyzicky náročná, nebezpečná alebo stereotipná, ale najmä tam, kde sa vyžaduje vysoká efektivita, presnosť, dôslednosť, citlivosť a rýchlosť, na ktorú človek s jeho zmyslami jednoducho nestačí.

Rozvoj informatizácie, automatizácie a riadenia systémov je stále rýchlejší. Aplikovanie výsledkov vývoja a výskumu v každodennom živote je čoraz potrebnéjšie a žiadanejšie. Rozliční elektronickí pomocníci robia náš život jednoduchším. Dnes si už ani nevieme predstaviť, že by sme museli v zime stále zapínať a vypínať kúrenie, otvárať a zatvárať regulačné ventily, aby sme dosiahli potrebnú teplotu a neplytvali pritom s prírodnými zdrojmi. Riadiace systémy využívajú sofistikované algoritmy šetriace energiu, kým bežný užívateľ väčšinou ani nevie ako dané zariadenie funguje.

So zvýšeným dopytom po automatických robotoch narastá aj dopyt po programátoroch a vývojových pracovníkoch, ktorých treba vyškoliť často pomocou softvérových simulátorov alebo modelov ozajstných zariadení. Jedno čínske príslovie hovorí: „Ak mi niečo vysvetlíš - zabudnem, ak mi to ukážeš - pochopím, ale ak to urobím sám - tak sa to naučím“. V súlade s touto pravdou sa na trhu objavujú aj programovateľné „hračky“ ako je robot „LEGO® Mindstorms® NXT“ od dánskej spoločnosti LEGO Group.² Nie je to celkom len hračka. Je to plnohodnotný

¹ Linus Torvalds - autor myšlienky a pôvodného obsahu jadra operačného systému Linux (Fínsko 1991), po ktorom je tento OS aj pomenovaný.

² LEGO a Mindstorms sú registrované obchodné známky spoločnosti LEGO Group Ltd.

programovateľný mikropočítač s bohatým príslušenstvom pre styk s okolím. Určený je pre edukačné účely. Umožňuje pochopenie rôznych riadiacich algoritmov na jednoduchej programovateľnej úrovni, ktorú ovládajú už aj deti na základnej škole.

Práca chce poukázať na možnosť alternatívnej výuky mikroelektroniky a logiky programovaním zariadení pracujúcich v reálnom čase, bez potreby nakupovania často drahých didaktických programov. Získané poznatky môžu byť využité aj pri programovaní iných, aj keď nie takých všestranných zariadení, ktoré možno postaviť svojpomocne podľa návodov dostupných na internete. Môžu to byť rozličné regulátory, programátory, snímacie a zobrazovacie programovateľné jednotky a podobne.

Obsahovo je práca rozdelená na tri kapitoly. V prvej je uvedený popis riadiacej jednotky LEGO Mindstorms NXT „Intelligent Brick“ a jej štandardné programové vybavenie.

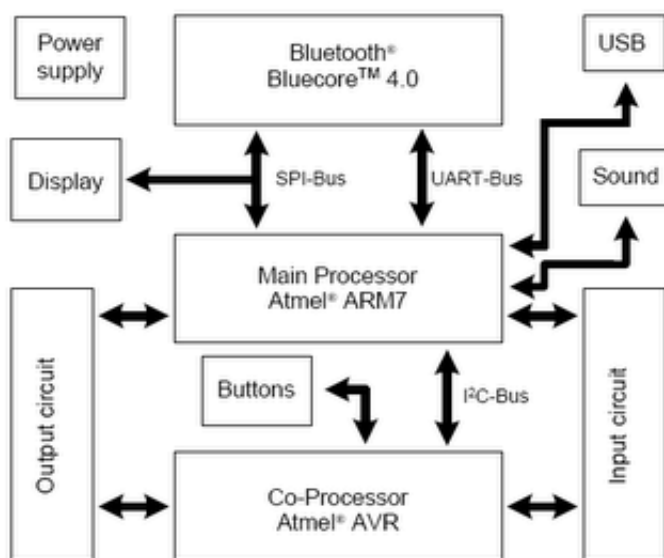
Obsahom druhej kapitoly je programovanie LEGO Mindstorms NXT v prostredí Linux, postup inštalácie NBC prekladača, vytvorenie nového projektu, prenos súborov medzi počítačom a jednotkou NXT. Záujemca tu nájde tiež detailný popis implementácie prekladača NBC do grafického vývojového prostredia Code::Blocks.

V záverečnej tretej kapitole je uvedený príklad grafickej aplikácie riadenia LEGO robota cez bezdrôtové rozhranie Bluetooth. Pomocou jednoduchého programu je vysvetlený princíp obsluhy vysielania a prijímania údajov medzi počítačom a robotom. Program na PC predstavuje virtuálny diaľkový ovládač autíčka, postaveného zo stavebnice LEGO Mindstorms NXT. Všetky v práci popisované programy sú nahraté na priloženom CD, ktoré je súčasťou práce.

Cieľom práce je stimulovať záujem o používanie Linuxu na školách, pretože sú to práve školy, na ktorých je propagácia progresívnych technológií najúčinnějšía.

1 LEGO MINDSTORMS NXT

Lego Mindstorms je nová generácia robotov používaných pri výučbe. Umožňuje študentom oboznámiť sa s využitím programovania v reálnom svete. Študenti sa môžu hravou formou zoznámiť s praktickou realizáciou riadiacich úloh. Základný zdroj informácií je na [3] a [4]. Prvým krokom je mechanické zloženie robota, čo rozvíja manuálne zručnosti. Následne je potrebné vytvoriť program, ktorý je nutné prispôbiť konštrukcii robota a odladiť tak, aby jeho funkcia naplnila predstavy autora.



Obrázok 1: Bloková schéma mikropočítača NXT Brick

Základom stavebnice Lego Mindstorms je 32-bitový mikropočítač NXT Intelligent Brick s dvoma procesormi. Hlavným je 32-bitový procesor ARM7 (48 Mhz) s 256 kB Flash pamäťou a 64 kB datovou pamäťou. Pomocný koprocesor je 8-bitový AVR procesor (8 MHz) so 4 kB Flash pamäťou a 512 B RAM. Základné rozhranie medzi človekom a riadiacou jednotkou tvorí monochromatický maticový displej. Komunikácia s PC je zabezpečená pomocou USB 2.0, prípadne technológiou Bluetooth (Bluetooth Class II V2.0 compliant). Po obvode riadiacej jednotky sú umiestnené 4 vstupné konektory RJ-12 (1,2,3,4), každý na 6-vodičový kábel s číslicovou zbernicou I2C na pripojenie snímačov a 3 výstupné porty (A,B,C), taktiež s konektormi RJ-12 na šesťvodičový kábel s digitálnym riadením na pripojenie akčných členov. Jednotka NXT Brick na svojej frontálnej strane obsahuje monochromatický grafický displej LCD 100 x 64 bodov a štyri tlačítka. Na zadnej strane pod konektorom USB je resetovacie

tlačítko a napájací zdroj 6 nabíjateľných AA batérií. Súčasťou zariadenia je aj programovateľný akustický menič.

1.1 Programové vybavenie LEGO Mindstorms NXT

Robot NXT vyrába komerčná firma a preto aj vývojové prostredie pre NXT je proprietárne, čo v praxi znamená, že za používanie programu musí jeho užívateľ platiť licenciou. Existujú však aj kompilátory z prostredia slobodného softvéru, ktoré sú voľne k dispozícii aj zo zdrojovým kódom potrebným pre prípadné úpravy. Sú šírené zväčša pod licenciou GNU/GPL. Tieto programy je možné kýmkoľvek používať, kopírovať, rozširovať a upravovať za podmienky uvedenia pôvodnej licencie a autora prípadnej úpravy s jej detailným popisom. Tieto programy sú preto vhodné hlavne pre študentov, ktorí ich môžu používať v škole, na pracovisku alebo v súkromí bez obavy z postihu za porušovanie autorských práv.



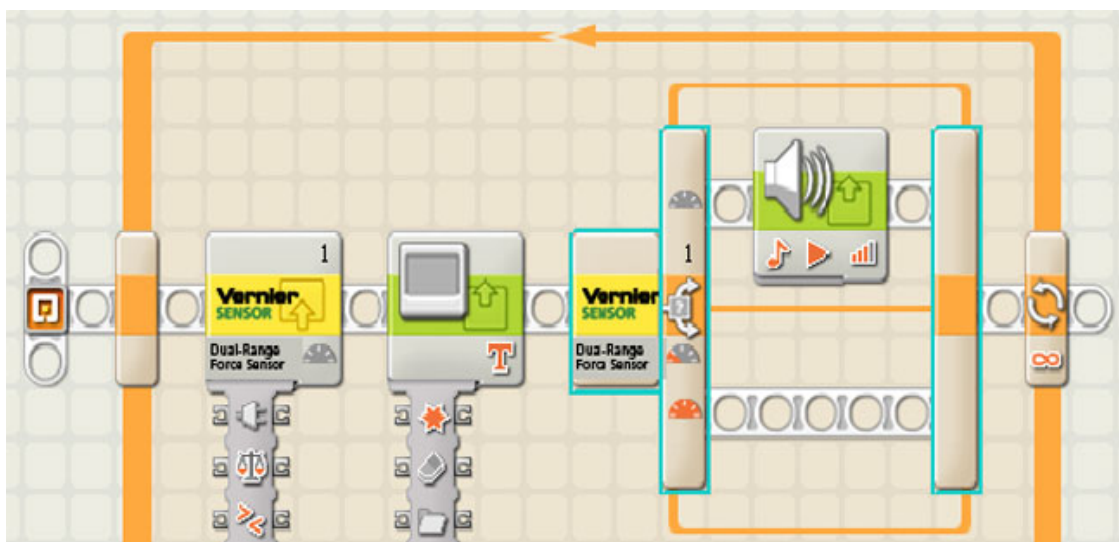
Obrázok 2: Príklad použitia NXT Brick na stavbu vozidla

Ďalšou veľkou výhodou slobodného softvéru je, že vývojové prostredie a používané programy sú aktuálne a ľahko udržiavateľné, pretože bez problémov sa dajú získať najnovšie verzie priamo cez internet, na rozdiel od inštalácie proprietárneho

softvéru z „odležaného“ CD a potreby zháňania rozličných „záplat“ a aktualizácií. Open source programy sú optimalizované na hardvérové prostriedky, na ktorých majú pracovať, takže počítače pod ich velením sú spravidla rýchlejšie a stabilnejšie ako pri použití zastaralého (ťažkopádneho) univerzálneho programu.

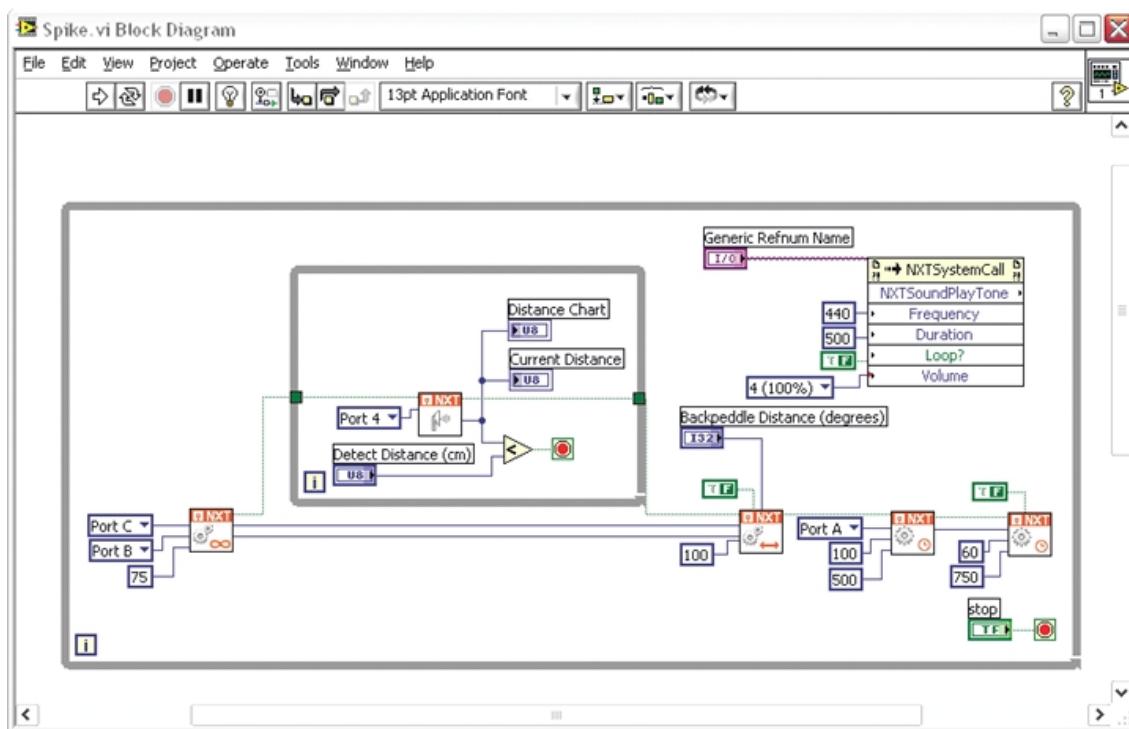
1.1.1 Programy pre operačný systém Windows

- LEGO Mindstorms Software Development Kit s jazykom NXT-G od spoločnosti The LEGO Group;
- MATLAB-Simulink®³ s knižnicou lego_kniznica od spoločnosti MathWorks;
- NI LabView grafický programovací jazyk firmy National Instruments;
- NBC/NXC alternatívny prekladač/jazyk pre LEGO Mindstorms NXT;
- MindScript skriptovací jazyk firmy Lego;
- RCX Code grafický MindScript;
- LASM Lego Assembler;
- PbForth jazyk Forth;
- Visual Basic Mindstorms SDK - ActiveX makrá Spirit.ocx;
- C++ Mindstorms SDK - API knižnica VPBrick;
- NQC IDE BricxCC jednoduchý jazyk so syntaxou podobnou „C“.



Obrázok 3: Ukážka grafického jazyka NXT-G spoločnosti LEGO Group

³ MATLAB a Simulink sú obchodné značky spoločnosti Math Works, Inc.



Obrázok 4: Ukážka programu LabView firmy National Instrument

1.1.2 Programy pre operačný systém Linux

- RWTH - Mindstorms NXT Toolbox for MATLAB z Aachen University, rozšírenie pre MATLAB-Simulink;
- BrickOS IDE (pôvodne LegOS) vývojové prostredie pre písanie programov firmvéru BrickOS pre LEGO jednotky RCX a NXT v jazyku C;
- LeJOS IDE vývojové prostredie pre písanie programov alternatívneho firmvéru leJOS do jednotiek RCX a NXT vytvorené na báze jazyka Java s prekladačom g++;
- NeXT Tools - rozšírenie Mac OS a Linux pre BricxCC;
- NBC/NXC alternatívny prekladač/jazyk pre LEGO Mindstorms NXT.

Stručná charakteristika:

a) RWTH - Mindstorms NXT Toolbox používa niektoré MATLAB® funkcie pre riadenie LEGO® MINDSTORMS® NXT robotov, vrátane obsluhy NXT systémovej jednotky, senzorov a motorov cez Bluetooth alebo USB rozhranie. Implementované má aj blokové funkcie pre ľahšie programovanie. Vývoj RWTH - Mindstorms NXT Toolbox bol motivovaný zapojením 380 študentov z nemeckej RWTH Aachen

University do projektu - MATLAB meets LEGO Mindstorms - v zime 2008, v záujme skvalitnenia edukačného procesu pre budúcih elektroinžinierov. Domovská stránka projektu je: <http://mindstorms.lfb.rwth-aachen.de/> ;

b) brickOS je zabudovaný operačný systém vrátane C a C++ programátorského prostredia pre Lego Mindstorms Robotics Kits, umožňujúce užívateľovi kitu programovať v jazyku C/C++ na rozdiel od štandardného programovacieho jazyka Lego (Lego Programming Language). Pôvodne bolo vyvinuté pre Linux, ale môže bežať aj na počítačoch s Windows. Hodí sa však iba pre staršie Lego jednotky RCX s procesorom Hitachi H8. Viac na: <http://brickos.sourceforge.net/> ;

c) LeJOS (vyslovovaný ako španielske slovo „lejos“ - ďaleko) je malý virtuálny stroj Java (JVM). Pre LEGO NXT brick bol portovaný v roku 2006. LeJOS NXJ definuje všetky potrebné triedy v NXJ API vrátane nástrojov na zavedenie kódu do jednotky NXT a aplikácií pre PC. Jeho základom je objektovo orientovaný jazyk Java; leJOS je dostupný na stránke: <http://lejos.sourceforge.net/> ;

d) NeXT Tools je univerzálna Mac OS X alebo Linux GTK2 aplikácia umožňujúca použitie nástrojov Windows, ktoré sú súčasťou BricxCC (Bricx Command Center) IDE. Stránka na stiahnutie: <http://bricxcc.sourceforge.net/utilities.html> ;



Obrázok 5: Nástroje NeXT

Všetky vyššie spomenuté programy sú šírené pod všeobecnou zverejňovacou licenciou GNU/GPL. Jej slovenské znenie v poslednej 3. verzii je na stránke: <http://www.gpl.sk/>.

2 PROGRAMOVANIE LEGO MINDSTORMS NXT V LINUXE

Po zvážení možností programov pracujúcich v prostredí OS Linux padlo rozhodnutie pre „Next Byte Codes“ (NBC) kompilátor [5], ktorý je voľne stiahnuteľný na stránke <http://bricxcc.sourceforge.net/nbc/> a beží na rôznych operačných systémoch: Windows, Mac OS, UNIX a Linux OS. Je šírený pod licenciou Mozilla Public Licence (MPL).

NBC prekladá zdrojové programy s koncovkou .nxc napísané v programovacom jazyku NXC. Názov je zostavený zo skratiek anglického „Not eXactly C“ - v preklade to znamená „nie presné C“, čím chcel jeho autor John Hansen [6] a [7] ospravedlniť nedodržanie normy ANSI C pri zostavovaní prekladača z dôvodu jeho zjednodušenia, pre lepšie porozumenie neprogramátorov a nadšencov LEGO Mindstorms.

Jazyk NXC bol vytvorený špeciálne pre programovanie robotov LEGO a má prepracované všetky potrebné funkcie na obsluhu senzorov, motorov, displeja, zvuku a komunikácie. Výborné dokumentačné spracovanie v príručke NXC_tutorial [8] a [9], zaručí prípadným záujemcom o programovanie, zvládnutie prvých krokov pri objavovaní zákutí tvorby programu, aj keď ešte nikdy neprogramovali.

NBC je vhodný práve pre štúdijné účely na školách, hlavne z dôvodu dostupnosti príručiek a programov bez potreby nákupu licencie (v škole je to často multilicencia). Použitý jazyk NXC je jednoduchý na pochopenie a má prehľadný zdrojový kód. Navyše pod Linuxom je možné kód písať v ľubovoľnom textovom editore a kompilovať priamo cez konzolu jedným príkazom. Táto vlastnosť sa ukázala ako veľmi praktická hlavne preto, že príkaz na preklad je možné jednoducho integrovať do nástrojov väčšiny grafických vývojových prostredí (GUI, IDE). V našom prípade to bolo prostredie Code::Blocks IDE (open source), ktoré je vytvorené v Jave, vďaka čomu môže bežať na rozličných programových platformách.

2.1 Štruktúra programu NXC

NXC program pozostáva z blokov kódu a premenných. Existujú dva odlišné typy blokov kódu: task (úloha) a function (funkcia). Každý typ má svoje vlastné jedinečné rysy, avšak štruktúra ich popisu v programe je rovnaká. Maximálny počet funkcií a úloh spolu, je 256.

2.1.1 Pravidlá písania programu

Poradie zostavovania kódu má dva aspekty: poradie, v ktorom sa objaví kód v súbore zdrojového programu a poradie, v ktorom budú programové bloky vykonávané:

- a) prvý aspekt je lexikálny, určuje prekladaču NBC, v akom poradí budú úlohy prekladané. Nemá vplyv na poradie vykonávania úloh;
- b) druhý aspekt (runtime order) určuje poradie vykonávania úloh tak, aby bola optimalizovaná doba odozvy aktívnych úloh.

Ad a) Pravidlá pre lexikálne usporiadanie sú:

- identifikátor (pomenovanie) úlohy alebo funkcie, musí byť kompilátoru známy pred prvým použitím v bloku kódu;
- definícia úlohy alebo definícia funkcie používa typy definované pre daný kompilátor;
- deklarácia úlohy alebo deklarácia funkcie taktiež používa definované typy;
- ak je úloha alebo funkcia raz definovaná, nemôže byť znovu predefinovaná alebo deklarovaná;
- ak je úloha alebo funkcia raz deklarovaná, nemôže byť ďalej redefinovaná.

Niekedy nastane situácia, keď nie je možné alebo žiadúce uviesť definíciu úlohy alebo funkcie pred prvým volaním úlohy alebo funkcie. V takom prípade stačí uviesť len deklaráciu úlohy alebo funkcie:

```
task name ();  
return_type name (argument_list);
```

pred programovým blokom, v ktorom sa prvé použitie nachádza. Argumenty uvedené v `argument_list` sa musia zhodovať so zoznamom formálnych argumentov uvedených neskôr v samotnej definícii úlohy alebo funkcie.

Ad b) Pravidlá poradia vykonávania úloh sú:

- program musí obsahovať jednu úlohu pomenovanú `main ()`, ktorá sa bude vždy vykonávať ako prvá;
- poradie v akom budú vykonávané ďalšie úlohy riadi firmvér API. Toto poradie je dokumentované v [6], kapitola Funkcie modulov (Command module functions);
- funkcia sa spustí vždy, keď je volaná z iného bloku kódu.

Posledné pravidlo sa môže zdať samozrejmé. Je však potrebné upozorniť na možné dôsledky vyplývajúce z paralelného vykonávania viacerých úloh. Volanie funkcie, ktorej obsluha sa už začala vykonávať na požiadavku inej úlohy, z ktorej bola predtým volaná, môže vrátiť nepredvídateľné výsledky. Úlohy môžu bezpečne zdieľať funkcie, ak sú zabezpečené volaním návratovej hodnoty `safeCall` alebo vyhradením zdrojov cez premenné typu `mutex`. Mutex je 32-bitová hodnota, použitá na zabezpečenie synchronizovaného prístupu, čím sa zabráni jednej úlohe používať hardvérové prostriedky dočasne používané inou úlohou. Druhou možnosťou bezpečného zdieľania zdrojov je použitie kľúčového slova `safeCall` v definícii funkcie pred jej návratovým typom. Ďalšie volanie funkcie predtypované slovom `safeCall` bude pozdržané, pokiaľ funkcia nevráti návratovú hodnotu predchádzajúcemu volaniu.

2.1.2 Úlohy

Vďaka firmvérovej multi-threading podpore jednotky NXT, každá úloha programu NXC priamo zodpovedá jednému vláknu. Úlohy sú definované pomocou kľúčového slova `task` s uvedením názvu úlohy a predpísanej syntaxe.:

```
task názov_úlohy ()
{
    // Telo úlohy
}
```

Názov úlohy môže byť akýkoľvek dovolený identifikátor. Program musí obsahovať vždy aspoň jednu úlohu s názvom `main`, ktorá sa spustí vždy, keď je spustený program. Telo úlohy obsahuje zoznam príkazov.

2.2 Inštalácia prekladača jazyka NXC v prostredí OS Linux

Všetky postupy [10] uvedené v tejto kapitole boli testované na počítačoch:

32-bit: ASUS Eee PC 1000H s procesorom Intel Atom N270 a OS Linux Fedora 13;

64-bit: ASUS P5K-VM s procesorom Intel Core2Duo E7200 a OS Debian Lenny.

V ďalších kapitolách často používaný výraz „právo roota“ predstavuje príkazy z konzoly terminálu zalogovanej (prihlásenej) s heslom administrátora (root).

2.2.1 Inštalácia NBC z predkompilovaného balíka

Pre 32 bitový systém je najjednoduchšia možnosť nainštalovať preloženú aplikáciu nbc z inštalačného balíka:

- a) najskôr treba stiahnuť zdrojový balík (nbc-x.x.x.rx.tgz) prekladača NBC

do dočasného adresára zo stránky:

http://sourceforge.net/projects/bricxcc/files/NBC_NXC/ alebo z konzoly príkazom:

```
$ wget http://downloads.sourceforge.net/bricxcc/nbc-1.2.1.r4.tgz
```

- b) potom balík rozbalíť do dočasného adresára ./nbc:

```
$ mkdir nbc
```

```
$ tar xzf nbc-1.2.1.r4.tgz -C nbc
```

```
$ cd nbc/NXT
```

- c) takto sa dá odskúšať funkčnosť prekladača:

```
$ ./nbc
```

```
Next Byte Codes Compiler version 1.2 (1.2.1.r4, built Tue Mar 15  
16:10:49 CDT 2011)
```

```
Copyright (c) 2006-2010, John Hansen
```

```
Use "nbc -help" for more information.
```

- d) napokon treba program nbc presunúť do adresára user/local/bin príkazom

s právom roota:

```
$ sudo mv nbc /usr/local/bin/
```

2.2.2 Inštalácia NBC zo zdrojového kódu

Pre ostatné systémy alebo v prípade nefunkčnosti predkompilovanej verzie NBC prekladača je potrebné NBC kompilátor najskôr skompilovať zo zdrojového kódu. Pretože program NBC je písaný v jazyku Pascal, preklad zdrojového kódu vykoná vhodný prekladač, napríklad Free Pascal Compiler od autora Florian Klaempfl:

A. Inštalácia prekladača jazyka Pascal

- a) stiahnuť balík Free Pascal Compiler (fpc-2.4.2.x86_64-linux.tar) zo stránky:

<http://www.freepascal.org> ;

b) rozbalit balík do dočasného (užívateľského) podadresára;

c) prejsť do podadresára s rozbaleným balíkom a spustiť inštaláciu príkazom:

```
$ sh install.sh
```

- pri inštalácii sa inštalačný program pýta na cieľový adresár `/usr/local` alebo `/usr`, - zvolit' `/usr/local`. Na otázky inštalácie `Textmode_IDE` (editor), `FCL` (linker) a `packages` je potrebné odpovedať áno (Y);

d) správnu funkciu programu Free Pascal Compiler možno vyskúšať príkazom:

```
$ fpc
```

- na ktorý program vypíše hlavičku s rozsiahlou nápovedou začínajúcou riadkom:

```
Free Pascal Compiler version 2.4.2 [2010/11/08] for x86_64 Copyright  
(c) 1993-2010 by Florian Klaempfl ...
```

B. Preklad programu NBC

V ďalšom kroku je potrebné pomocou prekladača jazyka Pascal preložiť zdrojové kódy kompilátora NBC v hardvérovom prostredí, na ktorom bude program NBC reálne používaný. Na to je potrebné:

a) stiahnuť zdrojový balík (`nbc-x.x.x.rx.src.tgz`) prekladača NBC do dočasného adresára

zo stránky: http://sourceforge.net/projects/bricxcc/files/NBC_NXC/ ;

b) vytvoriť podadresár `/nbc` a balík doň rozbalit;

c) v súbore `Makefile`, ktorý sa nachádza v podadresári `./nbc/NXT` opraviť položku:

```
PTOOLPREFIX=/usr/bin/
```

na

```
PTOOLPREFIX=/usr/local/bin/
```

(položku nemeniť, ak boli úlohy umiestnené do adresára `/usr/bin`);

d) z podadresára `./nbc/NXT` spustiť príkaz `make` s parametrom `install`:

```
$ cd ./nbc/NXT
```

```
$ make install
```

e) prekladom by sa mal vytvoriť binárny súbor `nbc` v podadresári `./nbc/NXT`. Ďalší postup je zhodný s postupom v kapitole 2.2.1. body c) a d).

2.3 Vytvorenie nového projektu

a) v užívateľskom podadresári `./Projects` vytvoriť adresár projektu, napr.:

```
$ mkdir ./Projects/nxthello
```

b) vo vzniknutom adresári vytvoriť zdrojový súbor `hello.nxc`, ktorý obsahuje krátky program:

```
// textový (zdrojový) súbor hello.nxc
task main ()
{
    TextOut (0,0,"Hello world!");
    wait (1000);
}
```

Jeden zo spôsobov, ako to urobiť, je otvoriť textový editor, napísať v ňom uvedený zdrojový kód a uložiť ako súbor `hello.nxc` do adresára projektu

`./Projects/nxthello`.

c) z adresára projektu `./Projects/nxthello` spustiť preklad príkazom:

```
$nbc hello.nxc -O=hello.rxe
```

d) výsledkom prekladu je spustiteľná aplikácia `hello.rxe`, ktorú treba jedným z nižšie popísaných spôsobov nahráť do NXT Brick.

2.4 Prenos súborov medzi počítačom a jednotkou NXT

Na nahrávanie (put, upload) do NXT a sťahovanie (get, download) súborov z NXT potrebujeme ovládací program pre riadenie komunikácie medzi PC a NXT Brick cez rozhranie USB 2.0.

V prostredí Linux možno prenos programov medzi PC a jednotkou NXT Brick dosiahnuť napríklad pomocou:

1. interpretera jazyka PERL a skriptu LiNXT;
2. prekladača NBC s parametrom `-d <názov_súboru.nxc>`;
3. utility T2N (akronym Talk-to-NXT).

2.4.1 Prenos súborov pomocou LiNXT

LiNXT je aplikácia napísaná pre interpretér jazyka Perl. Jej použitie vyžaduje triedu `Device::USB` (súčasť knižnice `libusb`) a spustenie príkazu s právom `roota`:

```
# perl linxt -u hello.rxe
```

2.4.2 Prenos súborov pomocou NBC

Od verzie programu NBC 1.0.1.350 je možnosť priameho zápisu vykonateľného kódu do jednotky NXT bez potreby súboru `.rxe` a to príkazom s právom `roota`:

```
# nbc -d hello.nxc
```

Podmienkou je existencia odladeného zdrojového súboru `.nxc`. V prípade, že je táto možnosť postačujúca, ďalšie kroky tejto podkapitoly možno preskočiť a pokračovať krokmi v kapitole 2.5.

2.4.3 Prenos súborov pomocou T2N

Talk to NXT (t2n) je jednoduchý nástroj na prenesenie povelov do NXT cez USB port. Domovská stránka projektu je: <http://www-verimag.imag.fr/~raymond/edu/lego/t2n/>

Existujú dva spôsoby inštalácie T2N driveru:

- stiahnutie zdrojového súboru a následnou kompiláciou priamo na jadro OS;
- stiahnutie predkompilovaného balíka pre danú platformu.

1. Stiahnutie zdrojového súboru, s následnou kompiláciou priamo na jadro operačného systému, je univerzálny spôsob. Vyžaduje si však zručnosti a skúsenosti s kompiláciou hlavne v prípade, ak nie sú vyriešené niektoré zo závislostí programových modulov. Kompilátor v takom prípade vypíše chyby, ktoré treba odstraňovať postupným ladením prekladu zdrojového kódu.

a) stiahnuť zdrojový súbor `t2n-0.2.src.tgz` z adresy:

<http://www-verimag.imag.fr/~raymond/edu/lego/t2n/t2n-0.2.src.tgz>

b) rozbaľiť balík v podadresári `./t2n-0.2.src` a preložiť program `t2n` príkazom

make na vytvorenie spustiteľného obrazu úlohy (predtým však treba mať nainštalované balíky g++ a libusb-dev):

```
$ tar xzf t2n-0.2.src.tgz
$ cd t2n-0.2.src
$ make
```

- c) výsledok prekladu - binárny súbor t2n (je uložený v podadresári ./obj) treba ho presunúť do ./usr/local/bin príkazom mv (move) s právom roota:

```
$ cd ./obj
$ sudo mv t2n /usr/local/bin
```

2. Stiahnutie predkompilovaného balíka pre danú distribúciu linuxu a následné presunutie do /usr/local/bin. Tento postup je jednoduchší, nedá sa však použiť pre všetky linuxové distribúcie. Balík obsahuje dve inštancie binárnych úloh: statickú (t2n.static) a dynamickú (t2n). Kým statická zostava obsahuje v sebe všetky potrebné funkcie rozšírenia jadra, dynamická prilinkuje potrebné knižnice až pri každom spustení úlohy. Moduly, ktoré dynamická t2n potrebuje možno zistiť príkazom ldd t2n na overenie závislostí. Pre správnu činnosť dynamickej t2n musia byť moduly v systéme prítomné. Výhody dynamických aplikácií sú, že zaberajú menej miesta a používajú jednotné, udržiavané a najnovšie knižnice:

- a) stiahnuť balík t2n-0.2.tgz a rozbaľiť do pomocného užívateľského adresára:

```
$ wget http://www-verimag.imag.fr/~raymond/edu/lego/t2n/t2n-0.2.tgz
```

- b) aplikáciu t2n premiestniť do adresára /usr/local/bin:

```
$ cd ./t2n-0.2
$ sudo mv t2n /usr/local/bin
```

- c) overiť závislosti aplikácie t2n príkazom ldd:

```
$ cd /usr/local/bin
$ ldd t2n
linux-gate.so.1 => (0xf77af000)
libusb-0.1.so.4 => /lib32/libusb-0.1.so.4 (0xf7788000)
libstdc++.so.6 => /usr/lib32/libstdc++.so.6 (0xf769a000)
libm.so.6 => /lib32/libm.so.6 (0xf7676000)
libgcc_s.so.1 => /usr/lib32/libgcc_s.so.1 (0xf7669000)
libc.so.6 => /lib32/libc.so.6 (0xf7516000)
/lib/ld-linux.so.2 (0xf77b0000)
```


V prípade, ak dynamická aplikácia `t2n` nemá vyriešené závislosti (systém hlási chýbajúce moduly), treba vyskúšať statickú verziu `t2n.static` z toho istého balíka.

- d) zapnúť NXT jednotku, pripojiť ju káblom k USB portu počítača a vyskúšať funkciu ovládača príkazom s právom roota:

```
# t2n -i
```

ak je všetko v poriadku, má sa objaviť výpis informácie z NXT, napr.:

```
#### NXT INFOS #####
protocol version=1.124
firmware version=1.1
NXT Name: NXT
Bluetooth address: 16:53:08:69:ffffffdd:00
Bluetooth signal: 0
Free user flash: 58132
```

- e) ostatné funkcie je možné vypísať príkazom `t2n` s parametrom `-help`:

```
# t2n -help
```

na čo systém vypíše obsah pomocníka:

```
t2n version 0.2
usage: t2n [options] | t2n -help
recognized options:
-b           : check battery level
-i           : print nxt info
-ls          : list files
-put <file.rxe> : upload file
-get <file.rxe> : download file
-v           : set verbose mode
-vv          : even more verbose
-vvv         : gossip
-help        : print help and return
-version     : print version and return
```

- f) zavedenie užívateľskej aplikácie `hello.rxe` z adresára `./Projects/nxthello` do zariadenia NXT pomocou `t2n`:

```
$cd /home/(username)/Projects/nxthello
```

```
# t2n -put hello.rxe
```

2.5 Spustenie programu v jednotke NXT

Ak bol program úspešne zavedený do jednotky robota NXT Brick, jeho názov sa objaví medzi programovými úlohami:

„My Files“ -> „Software files“ -> „hello“

Po stlačení tlačidla „Run“ sa na displeji má zobrazíť reťazec zapísaný v kapitole 2.3 do zdrojového súboru hello.nxc ako operand funkcie TextOut (). V našom prípade je to známy pozdrav z prvých lekcii programovania: „Hello world!“. Je tiež možné, že sa namiesto toho zobrazí iba strohé hlásenie systému NXT „file error“. V takom prípade kompilácia alebo nahrávanie súboru neprebehli správne. Stáva sa to hlavne vtedy, ak je nahrávanie súboru prerušované obsluhou iného aktívneho programu v NXT. Pred nahrávaním preto treba vypnúť aktívne programy. Problém môže byť spôsobený aj zastaralou verziou firmvéru v NXT. V takom prípade je potrebné stiahnuť novšiu verziu a nainštalovať do jednotky NXT, buď pomocou aplikácie „fantom“ v prostredí Windows alebo pomocou aplikácie „fwflash“ v Linuxe.

V čase písania tejto práce bola aktuálna verzia firmvéru 1.29 pod názvom: LEGO_MINDSTORMS_NXT_Firmware_V1.29.rfw dostupná na stránke výrobcu: <http://mindstorms.lego.com/en-us/support/files/Firmware.aspx>

2.6 Preprogramovanie firmvéru v jednotke NXT

1. v prostredí Windows je doporučený postup popísaný na stránke:

<http://mindstorms.lego.com/en-us/Default.aspx>

2. pre fanúšikov Linuxu je postup nasledovný:

a) skontrolovať prítomnosť balíkov (prípadne ich doinštalovať)

scons, libusb-dev, libusb-0.1-4.

Vo Fedore použiť príkaz s právom roota:

```
# yum install scons libusb-dev libusb
```

Pre Debian, Ubuntu použiť príkaz s právom roota:

```
# apt-get install scons libusb-dev libusb
```

b) stiahnuť archív libnxt do dočasného adresára alebo na plochu zo stránky príkazom:

```
$ wget http://libnxt.googlecode.com/files/libnxt-0.3.tar.gz
```

- c) rozbalit balík `libnxt-0.3.tar.gz` a presunúť sa do neho. Pomocou programu `scons` vytvoriť projekt:
- ```
$ tar xzf libnxt-0.3.tar.gz
$ cd ./libnxt-0.3/
$ scons
```
- d) presunúť sa do vytvoreného adresára `fwflash` príkazom:
- ```
$ cd ./fwflash
```
- e) program obsahujúci firmvér `LEGO_MINDSTORMS_NXT_Firmware_V1.29.rfw` stiahnutý zo stránky výrobcu LEGO Mindstorms, treba nakopírovať do adresára s aplikáciou `fwflash`;
- f) pripojiť kocku NXT káblom USB k počítaču a zapnúť ju;
- g) pomocou kancelárskej sponky stlačiť tlačidlo reset na viac ako 4 sekundy, čím sa jednotka prepne do servisného módu (upload firmware mode), teda do režimu, v ktorom zariadenie čaká na nový obsah riadiacej (flash) pamäte. Po prepnutí NXT jednotky vymizne obsah displeja a akustický menič vydáva tichý zvuk;



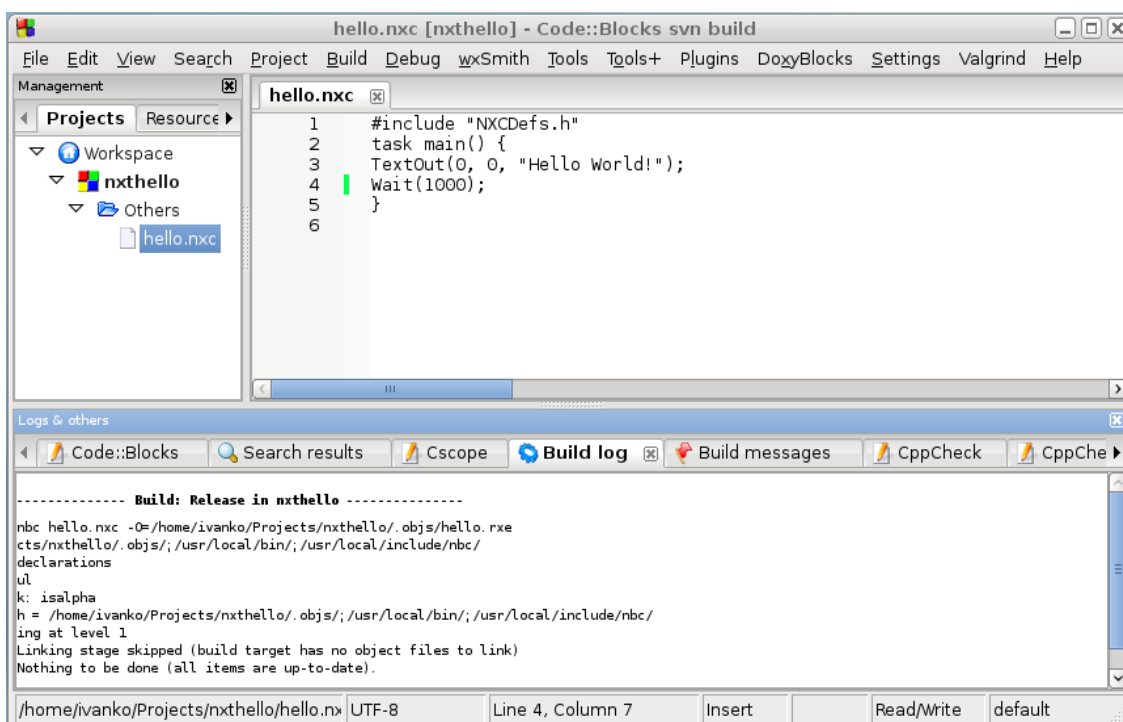
Obrázok 6: Resetovanie jednotky NXT

- h) firmvér zaviesť do jednotky NXT pomocou príkazu s právom roota:
- ```
./fwflash LEGO_MINDSTORMS_NXT_Firmware_V1.29.rfw
```

V prípade zablokovania displeja po skončení uploadu je potrebné ho odblokovať krátkym stlačením tlačidla reset alebo vybraním baterky. Ak NXT vydáva tichý zvuk, je aj naďalej v servisnom móde.

## 2.7 Nastavenie NBC kompilátora vo vývojovom prostredí Code::Blocks

Grafické vývojové prostredie (GUI) bežiacie pod Linuxom umožňuje pohodlne obsluhovať aplikácie, ktoré boli vytvorené pre prácu v textovom režime (konzola), vďaka čomu je do integrovaného vývojového prostredia možné včleniť aj kompilátor NBC. Je potrebné nastaviť požadované parametre a program vytvorený v jazyku NXC pre jednotku LEGO Mindstorms NXT môžeme písať, prekladať, ladiť a „napaľovať“ v elegantnom prostredí, akým je Code::Blocks IDE.



Obrázok 7: Vzhľad Code::Blocks IDE

Code::Blocks IDE je plne konfigurovateľné integrované vývojové prostredie z oblasti slobodného softvéru pod licenciou GNU GPL v 3.0. Je voľne stiahnuteľné a slobodne použiteľné pre každého. Domovská stránka je <http://www.codeblocks.org/>. Stiahnuť sa dá verzia pre operačné systémy Windows, Mac OS a Linux v 32 aj v 64-bitovej verzii. V niektorých distribúciach Linuxu je Code::Blocks IDE možné inštalovať priamo z repozitárov, inštalačným príkazom zadaným z konzoly alebo pomocou grafickej aplikácie.

Prostredie vytvorila skupina nadšencov filozofie GNU v programovacom jazyku C++, pričom boli použité grafické knižnice wxWidgets. Podporuje širokú škálu

kompilátorov pre najrozličnejšie jazyky a platformy od superpočítačov, cez serverové systémy, osobné počítače až po mobilné zariadenia a mikroprocesory rozličných výrobcov. Vďaka flexibilita a plnej užívateľskej rozšíriteľnosti vývojového prostredia nie je problém doplniť množstvo ďalších kompilátorov a nástrojov. V našom prípade sa rozšírenie týkalo integrovania prekladača NBC do prostredia Code::Blocks IDE.

### 2.7.1 Postup pre začlenenie prekladača NBC do Code::Blocks IDE

- a) v aplikácii Code::Blocks v menu „Settings->Compiler and Debugger settings“ pod položkou „Selected compiler“ vybrať voľbu „GNU GCC Compiler“ a skopírovať ju pomocou virtuálneho tlačidla „Copy“. Vzniknutú kópiu premenovať pomocou tlačidla „Rename“ napr. na Lego NXC Compiler. Tým sa dosiahne, že zmeny v nastavení NBC prekladača neovplyvnia pôvodné nastavenia gcc kompilátora;
- b) na karte menu „Settings->Compiler and Debugger settings -> Compiler Flags“ sa z dôvodu prehľadnosti nastavení doporučuje zrušiť všetky aktívne prepínače, pretože NBC kompilátor tieto možnosti nepodporuje;
- c) v tom istom menu „Settings->Compiler and Debugger settings“ vybrať kartu „Other settings“ nachádzajúcu sa úplne vpravo od „Toolchain executables“ a na nej stlačiť tlačidlo „Advanced options“. Zobrazí sa varovné okno upozorňujúce užívateľa na presnú znalosť významu príkazov, ktoré chce meniť. V tomto okne treba zadať „Áno“. V ďalšom okne na karte „Commands“ v položke „Command:“ vybrať „Compile single file to object file“. Pod položku „Command line macro:“ treba napísať výraz „\$compiler \$file -O=\$object“;
- d) v tom istom okne na karte „Others“ v položke „Object file extension“ treba uviesť príponu cieľového súboru (produktu prekladu) „.rx“. Zatvoriť okno „Advanced compiler options“;
- e) na karte „Toolchain executables“ zatiaľ stále otvoreného okna „Global compiler settings“ v zozname „Program files“ v prvých štyroch položkách „C compiler:“, „C++ compiler:“, „Linker for dynamic libs:“, „Linker for static libs:“ zapísať názov kompilátora „nbc“, v posledných troch položkách ponechať pôvodné nastavenie (gdb, „prázdne\_pole“, make). Ladiaci prostriedok (debugger) nie je pre NBC podporovaný.

### 2.7.2 Vytvorenie projektu

Postup pre vytvorenie projektu prekladača NBC v prostredí Code::Blocks je pomerne zložitý, preto je rozčlenený do viacerých krokov:

- a) spustiť program Code::Blocks a v štartovacom okne zvoliť voľbu „Create new project“, z ponuky šablón projektov vybrať „Empty project“, potom stlačiť „Go“;
- b) vyplniť názov projektu (napr. nxthello), stlačiť „Next“ a v ďalšom okne zmeniť typ prekladača na „Lego NXC Compiler“;
- c) v tom istom okne zrušiť zaškrtnutie „Create Debug configuration“ a naopak zaškrtnúť položku „Create Release configuration“, potom stlačiť Finish;
- d) do projektu pridať zdrojový súbor.nxc alebo vytvoriť nový súbor. V prvom prípade je súbor možné pridať cez menu „Project -> Add files“. Súbor je predtým vhodné umiestniť do adresára projektu. Nový súbor možno vytvoriť cez voľbu „File -> New -> Empty file“ a na otázku o pripojení súboru ku projektu odpovedať „Áno“. Do novootvoreného okna napísať názov súboru s príponou .nxc a uložiť;
- e) v zložke „Project -> Properties -> Build targets“ nadefinovať položku „type“ na „Native“;
- f) v poli „Build targets files:“ (okna „Project -> Properties -> Build targets“) musí byť zdrojový súbor zaškrtnutý a označený (vybratý), čím sa sprístupní voľba „Selected file properties“;
- g) po rozkliknutí „Selected file properties“ sa otvorí nové okno, kde treba na karte „Build“ označiť položky „Compile File“ a „Release“. Nakoniec treba otvorené okná zatvoriť tlačidlom „OK“.

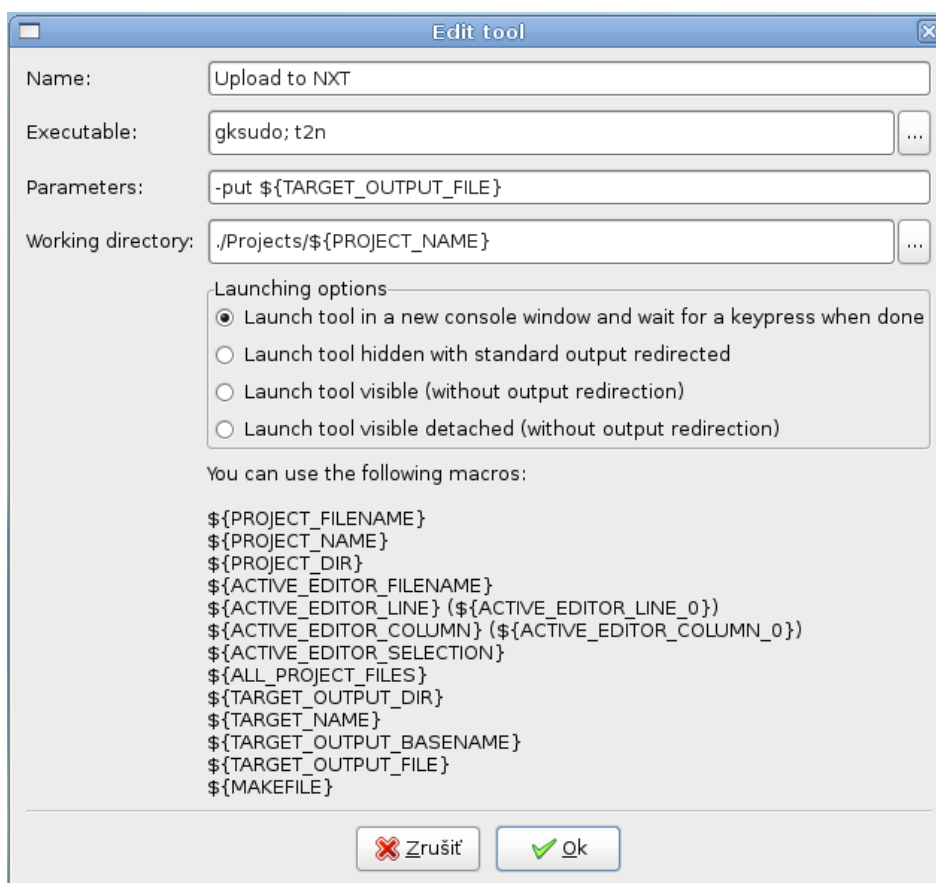
### 2.7.3 Kompilácia zdrojového súboru

Po dopísaní zdrojového kódu je možné zdrojový súbor preložiť (skompilovať) pomocou tlačidla „Build“ alebo stlačením kláves „CTRL + F9“. V procese kompilácie prekladač kontroluje správnosť zdrojového kódu. Výsledok prekladu je zobrazený v dolnom okne „Build log“ aj s popisom riadku programu, v ktorom sa vyskytla chyba. Okno výpisov „Logs & others“ sa dá odpojiť od plochy a presunúť niekde inde. Ak výpis už neobsahuje chyby, bude výsledok prekladu (.rxn súbor) uložený v podzložke projektu ./obj. Tento cieľový súbor treba nahráť cez USB kábel do jednotky LEGO

Mindstorms NXT, napríklad pomocou utility T2N, spôsobom popísaným v kapitole 2.4.3.

## 2.7.4 Vytvorenie užívateľského nástroja

Vývojové prostredie poskytuje možnosť na vytvorenie užívateľom definovaných nástrojov. Tieto nástroje môžu slúžiť na rozličné účely, napríklad spúšťanie rozličných utilít, zmeny konfigurácie, napalovanie procesorov a podobne.



Obrázok 8: Okno editora nástrojov Code::Blocks

Nasledujúci príklad ukáže, ako vytvoriť nástroj, ktorý po spustení odošle preložený program do jednotky LEGO Mindstorms:

- a) otvoriť project v programe Code::Blocks a v menu zvoliť „Tools -> Configure tools -> Add“;

- b) v okne „Edit tool“ napísať meno nástroja, pod ktorým sa bude nástroj zobrazovať, napr.: „Upload to NXT“;
- c) do položky Executable: uviesť príkazy (bez úvodzoviek) oddelené bodkočiarkou: „gksudo; t2n“;
- d) do položky Parameters: „-put \${TARGET\_OUTPUT\_FILE}“;
- e) do Working directory: „../Projects/\${PROJECT\_NAME}“;
- f) voľba Launching options určuje, či bude vykonanie nástroja viditeľné v konzolovom okne alebo nie;
- g) okno zatvoriť tlačidlom „OK“.

Spustenie vytvoreného nástroja je možné cez menu: „Tools -> Upload to NXT“.



### 3 OVLÁDANIE LEGO ROBOTA Z APLIKÁCIE NA PC

Ovládanie robota počítačom bolo realizované z aplikácie spustenej pod operačným systémom Linux (Fedora 13). Aplikácia písaná v jazyku C++ môže byť skompilovaná aj pre Windows. V takom prípade musia byť inštalované knižnice wxWidgets a knižnice ovládania bluetooth (Widcomm Bluetooth 6.1 alebo Windows Socket 2.0).

Využíva sa bezdrôtová komunikácia s riadením protokolu master-slave, čo znamená, že jedno zariadenie (PC) je riadiace - master a druhé zariadenie (NXT) je riadené - slave. Okrem riadiacej grafickej aplikácie v jazyku C++ pre počítač bolo potrebné v jazyku NXC naprogramovať aj obslužný program pre jednotku LEGO Mindstorms NXT.

Komunikácia cez Bluetooth slúži na výmenu informácií jednotky NXT s inými zariadeniami, ktoré sú k NXT Brick pripojené bezdrôtovým rádiovým spojením. Firmware v NXT zabezpečuje uchovávanie správ v schránkach, ku ktorým možno pristupovať pomocou prístupových metód. Zariadenia najskôr musia nadviazať spojenie cez vysokofrekvenčný kanál. Takto vytvoreným kanálom sú odosielané a prijímané správy obsahujúce v [11] predpísané hlavičky a dáta.

Program bežiaci na zariadení v roli master posiela dátové pakety pripojeným zariadeniam v roli slave, pomocou metódy `BluetoothWrite`. Zariadenia v režime slave ukladajú pomocou volania `SendMessage` svoje odpovedné pakety do svojho výstupného frontu správ, kde tieto správy čakajú, kým ich pripojené zariadenie v roli master nevyzdvihne. Pomocou protokolu priamych príkazov (`direct command protocol`), môže zariadenie v roli master posilať správy zariadeniu v roli slave vo forme textových reťazcov, ktoré sú uložené do príslušnej schránky (`mailbox`). Schránok môže byť až desať. Každá schránka na strane zariadenia slave je kruhový front, ktorý môže udržať až päť správ. Maximálna dĺžka správy je 58 bajtov. Správy sú odosielané zariadením master do jednotiek slave spracovaním príkazu `MessageWrite`. Táto správa sa na zariadení slave prečíta zo schránky pomocou volania `ReceiveMessage`. Podmienkou je, aby v čase posielania správy z master zariadenia bežal aj program na zariadení slave. V opačnom prípade bude správa na prijímacej strane stratená.

### 3.1 Komunikačný program pre riadiaci systém

Podobne ako v operačnom systéme Windows, tak aj v OS Linux sú vytvorené ovládacie knižnice, ktoré slúžia na priamy prístup k hardvéru. Pomocou nich možno bez vedomostí o detailoch obsluhy komponentov počítača naplno využívať ich vlastnosti vo svojom programe. Programátorovi stačí zavolať funkciu napísanú v niektorej knižnici a tá sa o zložité operácie postará. Aby mohla byť v programe využívaná komunikácia cez Bluetooth, je potrebné nainštalovať niekoľko knižníc z množiny nazvanej bluez:

a) najskôr je potrebné nainštalovať bluetooth kernel modul bluez, protokol bluez-hcidump, poskytujúci informácie o spojení a knižnicu bluez-libs nasledovne:

- pre debian, ubuntu:

```
(root $su) apt-get install bluez bluez-hcidump bluez-libs
```

- pre fedoru:

```
(root $su) yum install bluez bluez-hcidump bluez-libs
```

- prípadne stiahnuť potrebné balíky zo stránky <http://www.bluez.org/download/>

Inštalačné inštrukcie možno nájsť v súbore README;

b) overenie spojenia: na počítači možno Bluetooth zapnúť cez menu „System -> Preferences -> Bluetooth“ nastavením turn\_on. V menu NXT jednotky je nutné zapnúť položku „Bluetooth“. Existencia spojenia sa dá vyskúšať príkazom na PC:

```
$ hciconfig
```

v prípade, že systém vypíše:

```
hci0:Type: BR/EDR Bus: USB
```

```
BD Address: 00:22:43:AC:DE:8A ACL MTU: 1021:8 SCO MTU: 64:1
```

```
UP RUNNING PSCAN ISCAN
```

```
RX bytes:705 acl:0 sco:0 events:28 errors:0
```

```
TX bytes:1088 acl:0 sco:0 commands:28 errors:0
```

je spojenie nadviazané.

Komunikačný program `master.c` bol vytvorený v jazyku C++ a preložený prekladačom g++. Program má za úlohu prijímanie a odosielanie správ z PC do NXT prostredníctvom rádiového spojenia bluetooth.

Z dôvodu lepšej aplikovateľnosti zdrojového kódu v zložitejších programoch bol program rozdelený do programovej sekcie `master.c` a definičnej sekcie (knižnice)

blnxtcom.h, obsahujúcej vytvorené funkcie. Nasledujúca úkažka je pre lepšiu prehľadnosť zjednodušená tak, aby obsahovala len volania komunikačných funkcií, potrebných k prijímaniu a odosielaniu správ. Kompletný program sa nachádza na priloženom CD.

```
//program master.c
#include "blnxtcom.h"
#define max_message_size 64
int main (void)
{
 nxt_make_connection (); //vytvorenie spojenia
 char message[max_message_size];
 printf ("Aky znak??\n");
 scanf ("%s",&message); //načítanie správy
 nxt_sendmessage (2, message); //odosielanie správy do NXT
 // čakanie na odpoveď od NXT
 while (nxt_readmessage(3, message)){;}
 printf ("Reply: %s\n",message); // zobrazenie správy z NXT
 close (nxtSocket); //zrušenie spojenia
 return 0;
}
//end of program master.c
```

Komunikácia NXT s počítačom alebo iným bluetooth zariadením prebieha posielaním preddefinovanej postupnosti znakov (inštrukcie) do NXT, ktoré sa potom vyhodnotia a vykoná sa príslušný príkaz. Pomocou týchto príkazov nazývaných „Direct commands“ je možné poslať do NXT príkaz na prijatie alebo odoslanie správy, ale aj získanie informácie o stave batérie, zapnutia zvuku, spustenia programu, vypnutia programu a pod. Tieto príkazy sú popísané v programátorskej príručke: LEGO\_MINDSTORMS\_NXT\_Direct\_commands.

V ďalšom texte je uvedený návod na použitie „priamych“ povelov, konkrétne vo funkcii `nxt_send_message()`. Zdrojový kód `blnxtcom.h`, obsahujúci túto funkciu je dosť rozsiahly preto je na tomto mieste uvedený len fragment z tohoto programu. Kompletný program je opäť na priloženom CD.

```

//program blnxtcom.h
...
int nxt_sendmessage (int mbox, char *message)
{
 unsigned char btlength[2]={0x00,0x00};
 char rep[max_message_size]={0x00};
 char cmd[max_message_size]={0x00};
 int result, msgsize, replylength, error = 0;
 //posielanie žiadosti
 /* direct command format:
 * {length/lsb, length/msb, byte 0, byte 1 ... , byte n}
 * pre zápis správy (manual direct commands):
 * bajt 0: 0x00 or 0x80
 * bajt 1: 0x09
 * bajt 2: číslo schránky (0-9) v našom prípade 3
 * bajt 3: veľkosť správy
 * bajt 4-n: správa, kde n = veľkosť správy + 3
 * length/lsb: message size + 4
 * max message size: 58 (max. veľkosť správy)
 */
 // kontrola dĺžky správy
 msgsize = strlen(message) + 1; // jeden pre nulový znak
 if (msgsize > (max_message_size - 4))
 {
 fprintf(stderr, "messagewrite : message is to long");
 return -1;
 }
 // zostavenie povelu
 cmd[0] = 0x00;
 cmd[1] = 0x09;
 cmd[2] = mbox;
 cmd[3] = msgsize;
 memcpy(&cmd[4], message, msgsize);
 // veľkosť bluetooth správy
 btlength[0]= 4 + msgsize;
 ...
} //end of program blnxtcom.h

```

Funkcia `nxt_send_message()` má dva parametre. Prvý je typu `int` a predstavuje schránku (`front`), cez ktorú bude správa odoslaná. Jej hodnota musí byť zhodná s hodnotou vo funkcii `ReadMessage()` na prijímacej strane v NXT. Druhý parameter je smerník typu `char`, ktorý ukazuje na premennú `message` vytvorenú v `main()`. Jej hodnota sa skopíruje na adresu `&cmd[4]`. Od riadku 10 až po riadok 20 sú v komentári zobrazené hodnoty `direct commands`. Po doručení `cmd` správy a následnej kontrole formátu, jednotka NXT odošle počítaču potvrdenie prijatia správy. Kontrolná správa prichádzajúca späť do počítača obsahuje `direct commands` príkazy v poli `cdm` typu `char`. Jednotlivé bajty by mali obsahovať:

- bajt 0: `cdm[0]=0x02`
- bajt 1: `cdm[1]=0x09`
- bajt 2: `cdm[2]<>0x00` (statusový bajt)
- bajt 3: `cdm[3]=3` (číslo schránky, v našom prípade 3)
- bajt 4: `cdm[4]=veľkosť správy`
- bajt 5: `cdm[5]` (a ďalšie) =posielaná správa

V programe sa prvky poľa `cdm[ ]` porovnávajú s očakávanou hodnotou. Ak sa nezhodujú, správa bola prenesená chybné a program vráti hodnotu 1. Naopak ak sa zhodujú, správa bola prenesená úspešne. V praxi stačí porovnávať prvé tri bajty.

### 3.2 Komunikačný program pre riadený systém

Program `slave.nxc` bol napísaný v jazyku NXC a preložený prekladačom NBC. Predstavuje zjednodušený modelový program, bežiaci v NXT jednotke, ktorá prijíma správy z master zariadenia a následne ich odošle späť. Prijaté aj odosielané správy sú zároveň zobrazované na displeji.

```
//program slave.nxc
#define pixels_per_char 6
task main ()
{
 int loopnum = 0;
 string buffer="", message="";
```

```

ClearScreen(); // inicializácia displeja
TextOut (0, LCD_LINE1, "display mailbox"); // úvodná správa
Wait(1000);
while(1) //nekonečná slučka -program treba manuálne vypnúť
{
 ClearScreen();
 TextOut (0, LCD_LINE1, "loop number:");
 NumOut ((StrLen("loop number:") * pixels_per_char),\
 LCD_LINE1, loopnum);
 if (!ReceiveMessage(2, true, buffer))
 {
 message=buffer; // kópia správy
 TextOut (0, LCD_LINE2, buffer); //zobrazí správu
 SendMessage (3, message); // vráti správu do PC
 }
 Wait (1000);
 loopnum++;
}
}
//end of program slave

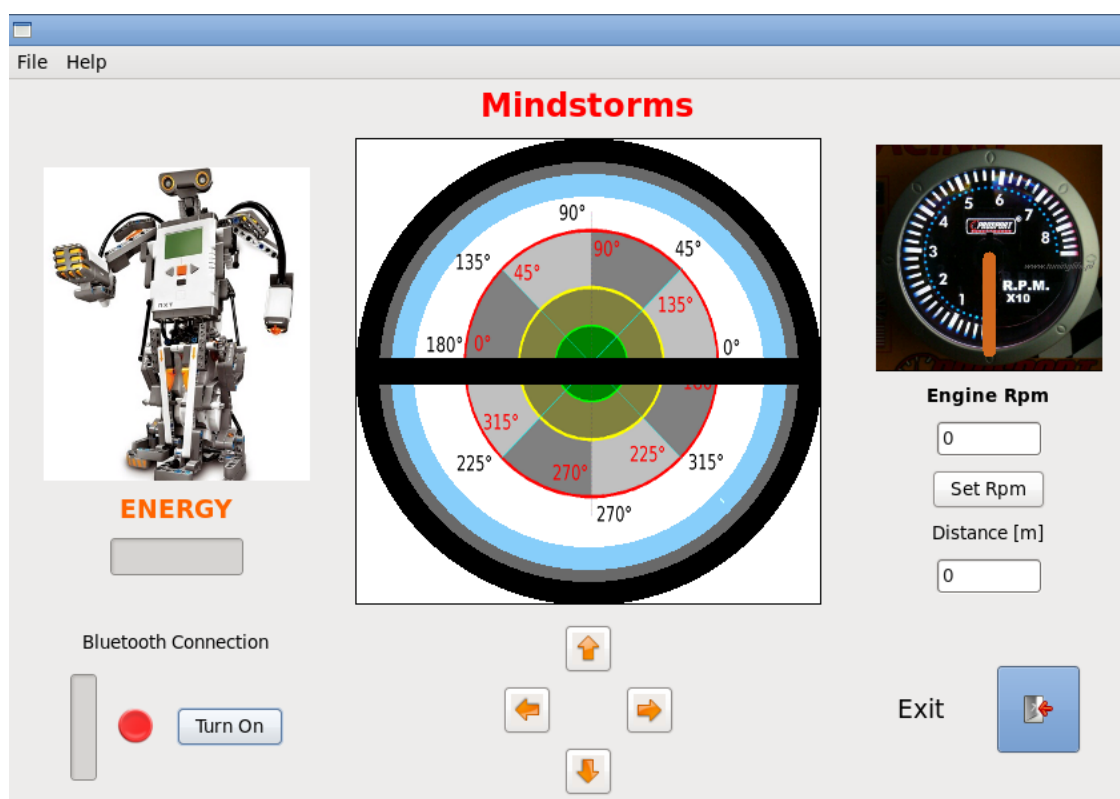
```

Po inicializácii displeja, celý program beží v nekonečnej slučke `while(1)`. Reťazcová premenná `buffer` slúži na príjem správy z PC. Komunikácia medzi Bluetooth a NXT je riadená operačným systémom jednotky LEGO Mindstorms. Obsluha zariadenia Bluetooth prebieha na inom vlákne paralelne s užívateľskou aplikáciou v jej vykonateľnej (cieľovej) podobe `slave.rxe`. Obsluha Bluetooth kanála zabezpečuje napĺňanie prijímacieho buffera znak po znaku nezávisle od programu `slave`. Priebežne prijímanú správu vyhodnocuje funkcia `ReceiveMessage`, ktorej návratová hodnota je nulová iba v prípade kompletne prijatej správy. Správa uložená v premennej `buffer` sa následne skopíruje do premennej `message`, aby sa predišlo jej možnému prepísaniu novou správou, prijatou pred spracovaním správy predchádzajúcej. Prvý parameter funkcie `ReceiveMessage (int mailbox, bool vymazat, string buffer)` (`mailbox`) určuje, cez ktorú schránku (`front`) bude správa prijímaná. V odosielacej funkcii `SendMessage` na PC musí byť nastavená tá istá hodnota (v našom prípade 2). Druhý parameter (`vymazat`) je typu `bool`. Ak je jeho hodnota `true`, schránka bude

po načítaní vymazaná. Tretí parameter (buffer) je premenná typu string, do ktorej sa ukladá správa prijatá z externého zdroja.

### 3.3 Ovládanie a monitorovanie robota z grafickej aplikácie

V predchádzajúcich dvoch kapitolách bol vysvetlený spôsob výmeny informácií medzi robotom a počítačom. Vytvorená užívateľská knižnica `blnxtcom.h` je taktiež volaná z grafickej aplikácie, programovanej pomocou grafickej knižnice `wxWidgets`. Umožňuje zostavovanie prenositeľných grafických aplikácií písaných v jazyku C++, ale aj v jazyku Python, Perl, Ruby a ďalších. Knižnica `wxWidgets` bola použitá aj pri vývoji Code::Blocks IDE, ktorý ju zároveň podporuje. Zvláda aj automatické generovanie blokov kódu, čo jednoduchým klikaním v menu umožňuje rýchle vytvorenie a formátovanie vzhľadu aplikácie a definovanie udalostí. Tento postup výrazne uľahčuje programovanie, pretože kvalitný vzhľad aplikácie môžeme získať v krátkom čase z preddefinovaných šablón.



Obrázok 9: Okno aplikácie bezdrôtového ovládania vozidla

Ďalšou fázou návrhu je dopĺňovanie vlastného kódu do vygenerovanej programovej kostry. To si vyžaduje pochopenie vygenerovaného kódu. Generátor prácu síce zjednoduší, ale rozhodne ju za programátora celú nespraví. Pre pochopenie grafickej knižnice treba preštudovať dokumentáciu. Netreba sa ju učiť naspamäť, stačí vedieť s ňou pracovať. Vytváranie grafických aplikácii predpokladá ovládanie objektovo orientovaného programovania.

Od grafickej aplikácie sa vyžaduje, aby okrem odosielania a prijímania správ, vytvorených na základe vstupných informácií, stlačením kláves alebo klikaním na príslušné tlačidlo, poskytovala aj užívateľsky prívetivé zobrazenie dostupných hodnôt nielen v textovom ale aj v grafickom vyjadrení. Grafické zobrazenie prijatých, nameraných alebo vypočítaných hodnôt napomáha k lepšej predstavivosti operátora o stave systému, ako pri alfanumerickom vyjadrení. Stĺpec s premenlivou šírkou predstavuje stav vybitia batérie, farebný terčik červená/zelená signalizuje existenciu Bluetooth spojenia, natočenie volantu určuje smer pohybu vozidla a ručička tachometra jeho rýchlosť.

Zobrazenie hodnôt v textových poliach je pomerne jednoduché. Stačí nastaviť hodnotu premennej textového poľa, ktorá je typu `wxstring` a následne bude zobrazená. Problém nastáva pri vykresľovaní obrázkov. Tie sa zvyčajne kreslia do panelu. Funkcia na vykreslenie využíva inštanciu triedy `wxPaintDC` alebo `wxBufferedPaintDC` – optimalizovanú na prekresľovanie len tých častí obrázku, ktoré sa zmenili a je ich potrebné prekresliť. Po vytvorení inštancie ju použijeme ako vstupný parameter do funkcie `Render(wxDC &dc)`. Jej obsahom sú všetky príkazy určujúce vzhľad, napr. nastavenie hrúbky pera, vykreslenie pozadia a vykreslenie čiar.

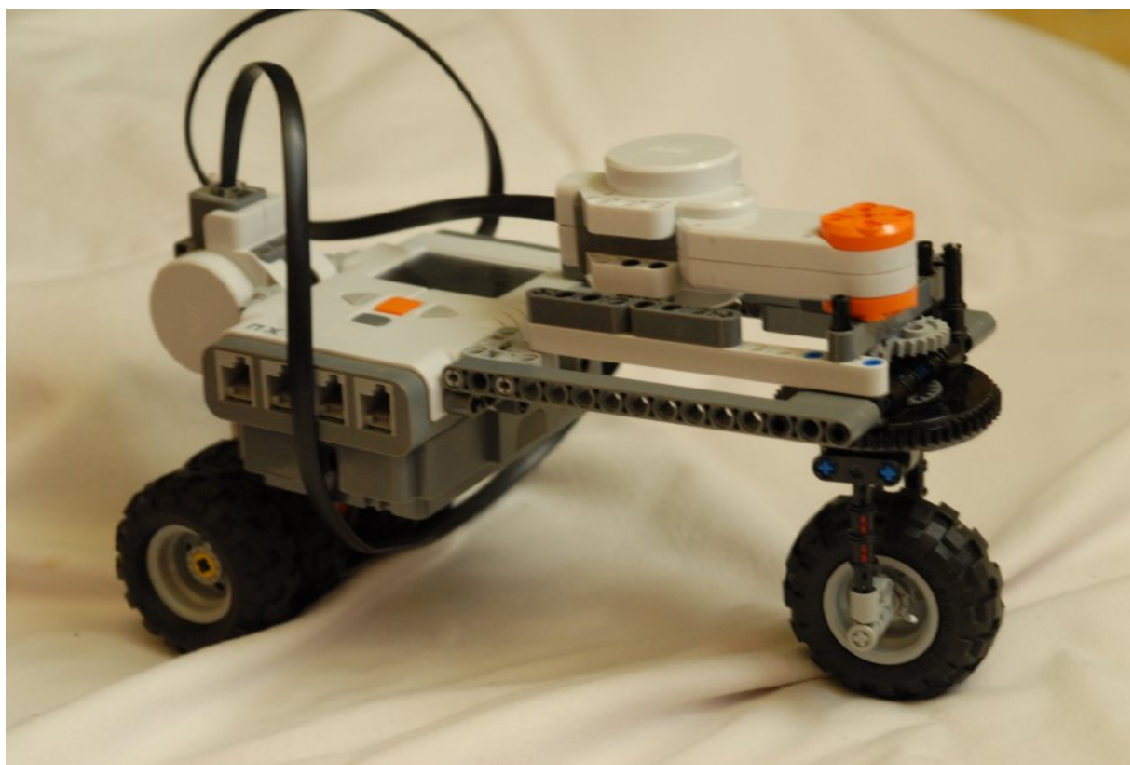
Dôležité je spomenúť, že funkcia `Render()` používa globálne premenné, ktorých hodnota sa mení po stlačení niektorej klávesy alebo tlačidla v inej časti programu. Programový blok na vykreslenie scény, ktorý obsahuje už spomenutú inštanciu triedy `wxBufferedDC` a funkciu `Render()`, je volaný prostredníctvom `wxeventov`, ktoré sa pripájajú do rámca cez službu `connect` a môžu byť volané aj z inej časti programu.

Nepříjemnou vlastnosťou použitej komunikácie je dlhá doba odozvy ovládaného zariadenia na povel z počítača. Na obidvoch zariadeniach sa nachádza operačný systém. Ten pri komunikácii používa zabezpečovacie prvky a potrebuje pred každým odoslaním a prijatím správy zariadenie identifikovať - overiť pin kód, pod ktorým bolo spojenie so zariadením vytvorené. Počas komunikácie sú medzi zariadeniami taktiež



prenášané potvrdzovacie správy, čím sa predlžuje odozva na vykonanie príkazu. Preto nie je možné odosielať informácie do NXT príliš často, napríklad po viacnásobnom stlačení klávesy.

Na monitore sa po každom stlačení šípky doľava alebo doprava posunie volant o  $4^\circ$ . Ak by sme držali šípku na klávesnici dlhšie, klávesnica sa prepne do režimu rýchleho opakovania, čím sa uhol mení po  $4^\circ$  v krátkych časových intervaloch. V aplikácii to vyvolá dojem súvislého pohybu volantu, avšak v prijímači nastane problém so zahltením prijímacieho buffera častým odosielaním správy o zmene uhla do robota. Jeho prijímacia schránka sa zaplní a informácie sa buď stratia alebo v lepšom prípade budú vykonávané s veľkým oneskorením. Pohyb sa stane príliš nepružný, polohovanie nepresné a aplikácia nepoužiteľná. Pri komunikácii cez bluetooth je preto vhodné, viacnásobne sa opakujúce informácie nejakú dobu (napr. 1 sekundu) zbierať a následne odosielať ich kumulovanú hodnotu v jednej správe. Takéto ovládanie nie je veľmi pružné, ale je použiteľné.



Obrázok 10: Model vozidla

Na vytvorenie rýchleho modelu (auto na diaľkové ovládanie) by bolo vhodnejšie použiť infračervený port, ktorý umožňuje rýchlejšiu reakciu v reálnom čase. Kým by

bolo stlačené tlačidlo, vysielal by sa signál z diaľkového ovládača. Pokiaľ by daný signál autíčko prijímalo, motor by bol v pohybe. Po prerušení signálu by sa motor ihneď zastavil.

Po prijatí informácie o veľkosti uhla sa robot snaží nastaviť motor do požadovanej polohy. Pootočí motorom, následne však musí overiť, či má motor správny uhol. Aj tento dej si vyžaduje čas. Vzniká nežiadúce oneskorenie. Preto je vhodnejšie odosielať naraz súčet niekoľkých zmien od posledného vysielania s väčším časovým odstupom.

Pravidelné odosielanie správ je zabezpečené časovaním nekonečnej slučky, v ktorej sa odošle správa a následne sa čaká určený interval. Ostatné časti programu sa prestanú vykonávať. Výsledkom je zamrzenie aplikácie a odosielanie tej istej správy. Preto je potrebné zabezpečiť, aby proces pravidelného odosielania správy neobmedzoval činnosť zvyšku aplikácie. Na tento účel sa používajú vlákna. Moderné počítačové operačné systémy umožňujú funkčnosť a chod viacerých procesov súčasne. Procesy sa uchádzajú o pracovný čas procesora podľa priority. Procesy s vyššou prioritou, akými sú služby operačného systému, získavajú viac času a vykonávajú sa prednostne. Naopak procesy s nižšou prioritou sa vykonávajú, až keď sú obslužené všetky procesy s vyššou prioritou, ktoré o obsluhu požiadali. V programovacom jazyku C/C++ sa na prácu s viacerými vláknami používajú knižnice s názvom `POSIX_thread`. Umožňujú súčasné vykonávanie viacerých úloh, ktoré by však nemali súčasne pristupovať k rovnakým premenným, pretože by sa hodnoty navzájom prepisovali. To by viedlo k nesprávnym výsledkom a mohlo spôsobiť nestabilitu programu. Ak je napriek tomu nevyhnutné niektoré premenné zdieľať, musí sa zabezpečiť ich uzamknutie pred prístupom z programu bežiaceho na hlavnom vlákne v čase, keď sú premenné používané procesom bežiacim na vedľajšom vlákne. Je žiadúce, aby premenné boli uzamknuté iba na nevyhnutnú dobu, najlepšie iba počas ich kopírovania z/do lokálnych premenných. Pri spracovaní načítaných hodnôt a pri výpočte nových hodnôt ako aj počas vysielania správy sú už premenné odomknuté, čím sú sprístupnené hlavnému procesu.

Príručka používania knižnice `POSIX_thread` sa nachádza na stránke:

<http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html> .

## 4 ZÁVER

Práca na zvolenom príklade simulácie diaľkového ovládania vozidla LEGO Mindstorms NXT preukázala použiteľnosť nasledovných programov z oblasti slobodného softvéru:

1. grafické prostredie Code::Blocks na implementáciu prekladača NBC pre programovanie stavebnice LEGO Mindstorms NXT;
2. program T2N na prenos súborov medzi PC a NXT Brick cez rozhranie USB;
3. program FWFLASH na upload firmvéru do NXT Brick;
4. knižnice wxWidgets, Bluez a POSIX\_thread pri programovaní grafickej aplikácie na ovládanie robotov LEGO v prostredí Code::Blocks IDE za použitia prekladača g++.

Postavený model vozidla ako aj vytvorená grafická aplikácia majú slúžiť ako príklad možnosti vytvorenia master-slave komunikácie pomocou Bluetooth. Okrem toho slúžia ako názorná ukážka využitia získaných vedomostí pri štúdiu programovania v jazyku C++. Použitie Bluetooth na riadenie autíčka sa ukázalo ako nevhodné z dôvodu dlhšej odozvy. Toto spojenie je použiteľné napríklad na nastavovanie parametrov NXT zariadení, poprípade odosielanie riadiacich inštrukcií alebo stavových dát v prípadoch, keď sa nevyžaduje okamžitá odozva systému.

Pri súčasnom ladení obidvoch programov, pre robot aj pre počítač, nastáva potreba častého prepínania medzi dvoma rôznymi projektami. Pretože program Code::Blocks môže byť spustený iba raz a pre jeden projekt súčasne, bolo potrebné prácu s jedným projektom zakaždým ukončiť a následne otvoriť projekt druhý. Z časových dôvodov je preto lepšie ladiť aplikácie z konzoly, pretože naraz môže byť na jednom systéme otvorených viac konzolových okien. V jednej konzole môže byť pripravený prekladač nbc a v druhej napr. g++. Inou možnosťou je vytvorenie nástroja tools pre nbc preklad priamo v Code::Blocks so súčasným zavedením aktívneho projektu používajúcim prekladač g++.

## ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

[1] WELSH, Math et al.: Používame Linux. Podrobný príručka Linuxem. Computer Press Brno, 2003. Tretie aktualizované vydanie. ISBN 80-7226-698-5 (659s).

[2] MITCHELL, Mark et al.: Pokročilé programovanie v operačnom systéme Linux. SoftPress Praha, 2002. ISBN 80-86497-29-1 (320s).

[3] Lego\_Mindstorms na internetovom portáli Wikipedia. Dostupné na internete [cit. 2011-01-18]: <[http://en.wikipedia.org/wiki/Lego\\_Mindstorms](http://en.wikipedia.org/wiki/Lego_Mindstorms)>.

[4] Lego\_Mindstorms na stránkach výrobcu LEGO Group, Ltd. Dostupné na internete [cit. 2011-01-18]: <<http://mindstorms.lego.com/en-us/Default.aspx>>.

[5] HANSEN, John: NBC Programmer's Guide. Verzia 1.2.1 r4, 6.okt.2010. Dostupné na internete [cit. 2011-04-21]: <[http://bricxcc.sourceforge.net/nbc/doc/NBC\\_Guide.pdf](http://bricxcc.sourceforge.net/nbc/doc/NBC_Guide.pdf)>.

[6] HANSEN, John: NXC. Verzia 1.2.1 r4, 5.okt 2010. Dostupné na internete [cit. 2011-04-21]: <[http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC\\_Guide.pdf](http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_Guide.pdf)>.

[7] HANSEN, John: NXC -Príručka programátora. Manuál k verzii 1.0.1 b33 zo dňa 10.okt 2007, slovenský preklad Pavel Petrovič, 3.aug 2009. Dostupné na internete [cit. 2011-04-21]: <[http://webcvs.robotika.sk/cgi-bin/cvswb/~checkout~/robotika/lego/doc/nxc/guide/NXC\\_Guide\\_sk.pdf](http://webcvs.robotika.sk/cgi-bin/cvswb/~checkout~/robotika/lego/doc/nxc/guide/NXC_Guide_sk.pdf)>.

[8] BENEDETTELLI, Daniele: Programming LEGO NXT Robots using NXC. NXC Tutorial, verzia 2.2, 7.jún 2007. Dostupné na internete [cit. 2011-04-21]: <[http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC\\_tutorial.pdf](http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf)>.

[9] BENEDETTELLI, Daniele: Programovanie robotov LEGO NXT pomocou NXC. Verzia 2.2, 7.jún 2007, slovenský preklad Ing. Slavko Fedorik, 24. marca 2010. Dostupné na internete [cit. 2011-04-21]:

<[http://virtuallab.kar.elf.stuba.sk/robowiki/images/9/99/NXT\\_tutorial\\_sk.pdf](http://virtuallab.kar.elf.stuba.sk/robowiki/images/9/99/NXT_tutorial_sk.pdf)>.

[10] LARSEN, J.C.: Using NXC on Linux, 04.sept 2008. Dostupné na internete [cit. 2011-04-21]:

<<http://www.jclarsen.dk/index.php/guides/21-programming/52-using-nxc-on-linux>>.

[11] SCHULTZ, Jeremy: Lego NXT, Bluetooth and Linux. Prezentácia, 23.jan 2008.

Dostupné na internete [cit. 2011-04-21]: <[http://www.cs.uleth.ca/~benkoczi/3720/data/NXT\\_Bluetooth\\_handout-jeremy.pdf](http://www.cs.uleth.ca/~benkoczi/3720/data/NXT_Bluetooth_handout-jeremy.pdf)>.