

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

DIPLOMOVÁ PRÁCE

Využití robota Lego MindStorm pro výuku informatiky



2011

Bc. Daniel Krhánek

Anotace

Práce se zabývá vytvořením multimediálního učebního materiálu pro výuku programování s využitím robotické stavebnice Lego Mindstorm. Učební materiál se skládá z výukových projektů odstupňovaných podle obtížnosti. Dále je zde sekce pro úvodní seznámení s robotem, základními programovacími konstrukcemi a výuka technické angličtiny v rozsahu této problematiky. Programy se tvoří současně v ikonografickém prostředí a jazyku Java, aby se materiál mohl využít na základních i středních školách.

Děkuji rodině, vedoucímu práce Mgr. Jiří Zaccpalovi Ph.D. za spolupráci a podporu, vedení obou škol, kde působím, za organizační a materiální pomoc. Kolegům za jejich připomínky a náměty a Mgr. Ondráčkové za gramatickou korekturu.

Obsah

1. Úvod	8
2. Stavebnice Lego Mindstorm NXT	9
2.1. Komponenty stavebnice	9
2.2. Technická specifikace	10
2.3. Komunikace	12
3. Programovací prostředky	13
3.1. Vývojové prostředí LEGO Mindstorm NXT-G	13
3.1.1. Datové vodiče	15
3.1.2. Výstupní programové bloky	15
3.1.3. Vstupní programové bloky	17
3.1.4. Řídící programové bloky	18
3.1.5. Datové bloky	19
3.1.6. Speciální bloky	21
3.1.7. Vlastní bloky	22
3.2. Programování v Javě - firmware LeJOS	22
3.2.1. Klíčové vlastnosti	22
3.2.2. Výuka objektového programování	23
3.2.3. Vícevláknové programování	24
3.2.4. Ukázkový program	24
4. Dostupné učební materiály	28
4.1. Literatura	28
4.2. Webové stránky	28
4.3. Školení a kurzy	28
5. Robotické soutěže	30
5.1. First Lego League	30
5.2. Eurobot	31
5.3. Robot Challenge	32
5.4. Fieldrobot	32
5.5. Robotour	33
6. Způsob realizace výukového materiálu	34
6.1. Zvolené technologie	34
6.2. Příprava multimediálních souborů	34
6.3. Členění materiálu	35
6.4. Mezipředmětové vztahy	35
6.5. Návrh a výroba vlastních hardwarových prvků	35

7. Popis vlastního materiálu	36
7.1. Základy	36
7.1.1. Kostka NXT	36
7.1.2. Motory	36
7.1.3. Senzory	36
7.1.4. Speciální doplňky	36
7.1.5. Software NXT-G	37
7.1.6. Software Java LeJOS	38
7.2. Language corner	39
7.3. Projekty 1	39
7.3.1. Projekt 1.1 - Plnou parou vpřed !	39
7.3.2. Projekt 1.2 - Ovládej tlesknutím	40
7.3.3. Projekt 1.3 - Měření místnosti	42
7.3.4. Projekt 1.4 - Turnikety v muzeu	43
7.3.5. Projekt 1.5 - Střežení objektu	44
7.4. Projekty 2	46
7.4.1. Projekt 2.1 - Přejít pro chodce	46
7.4.2. Projekt 2.2 - Ovládání brány	47
7.4.3. Projekt 2.3 - Udržování teploty čaje	48
7.4.4. Projekt 2.4 - Regulace hladiny v nádrži	50
7.4.5. Projekt 2.5 - Automatické zavlažování rostliny	51
Závěr	53
Conclusions	54
Reference	55
8. Obsah příloženého CD	56

Seznam obrázků

1.	Schéma zapojení výstupního portu NXT	11
2.	Schéma zapojení vstupního portu NXT	11
3.	Okno prostředí LEGO NXT-G s otevřeným programem	14
4.	Hrací plán kategorie Puck collect soutěže Robot Challenge	33
5.	Kategorie Line follower soutěže Robot Challenge	34

Seznam tabulek

1. Barevné značení datových vodičů	15
----------------------------------------------	----

1. Úvod

Robotika je obor, který se neustále dynamicky rozvíjí. U robotů se zdokonalují mechanické vlastnosti, schopnost vnímání okolního světa a zejména jejich inteligence. Využívají se ve všech možných oblastech lidské činnosti. Od zemědělství, přes průmysl, lékařství až po speciální aplikace v kosmickém výzkumu.

Existuje mnoho studijních oborů, které se informatikou a robotikou zabývají. Problémem je ale neustálý pokles zájmu studentů o veškeré technické obory. Důsledkem je nedostatek mladých lidí v oblasti průmyslu, který se již projevuje i na trhu práce. Podle závěru průzkumu ministerstva školství [8] jsou příčinou zejména náročnost přírodovědných předmětů a tím způsobený horší prospěch a strach z neúspěchu při studiu. Když opomeneme důvody genderové, které jsou obecně známé, zůstává nám další závažná příčina, a to nedostatečná motivace na základních i středních školách. Vzhledem ke kurikulu předmětů a vytíženosti vyučujících doporučují přenést těžiště výuky na pomůcky, které zvýší přitažlivost výuky - např. multimédia nebo modely. Navíc se ukazuje zvýšený zájem studentů o obory, u kterých si dokáží představit konkrétní obsah.

Tato práce se zabývá vytvořením multimediálního učebního materiálu pro zatraktivnění výuky informatiky na základních a středních školách. K vlastní výuce používá robotickou stavebnici LEGO Mindstrom NXT education, která nabízí nepřehledné množství možností. První kapitola popisuje tuto stavebnici a její komponenty. Dále jsou popsány možnosti programování a podrobnější popis dvou programovacích prostředků, které jsem vybral pro účely výuky. V dalších kapitolách jsou rozebírány dostupné učební materiály a robotické soutěže, kterých se mohou žáci a studenti zúčastnit. Pak už následuje praktická část, kde je popsáno řešení vlastního učebního materiálu a jednotlivé sekce, ze kterých se materiál skládá. V závěru jsou shrnuty nejdůležitější poznatky celé práce a zhodnoceny zkušenosti při používání vytvořeného materiálu ve výuce.

2. Stavebnice Lego Mindstorm NXT

Stavebnice LEGO Mindstorm NXT je dalším úspěšným produktem společnosti LEGO navazující na starší verzi LEGO RCX. Vychází ze součástek celosvětově rozšířené stavebnice LEGO Technics a doplňuje ji o robotické komponenty. Pořizuje se jako základní sada nezbytná pro stavbu robotů a jejich programování. Tuto sadu je možno snadno rozšiřovat o další součástky i softwarové možnosti.

2.1. Komponenty stavebnice

Základní sada se skládá z následujících komponent:

- řídicí jednotka, tzv. Inteligentní kostka NXT, což je 32-bitový mikropočítač s pamětí typu flash vybavený komunikačními a řídicími porty
- tři motory s integrovanými senzory otáček, které tak umožňují precizní řízení jejich chodu
- tři žárovky s barevnými kryty pro pokusy se světlem
- dva Touch senzory snímají dotyk robota s překážkou a umožňují robotu na ní reagovat
- Light sensor měří intenzitu osvětlení okolí nebo odraženého světla z vlastního zdroje
- Ultrasonic sensor měří vzdálenost od překážky pomocí odrazu ultrazvukového vlnění v rozsahu od 0 do 255 cm

Kostka NXT se s periferiemi propojuje speciálními kabely s RJ konektory a pomocí redukce umožňuje i připojení starších analogových komponent stavebnice LEGO RCX.

Ve většině projektů, které jsem do učebního materiálu navrhl, se využívá pouze tato základní sada, případně jednoduché doplňky vlastní výroby. Ke stavebnici se však dají dokoupit další senzory, které rozšiřují možnosti prováděných experimentů. Z této velké nabídky jsou zajímavé například:

- Temperature sensor pro měření teploty (digitální nebo analogový).
- Acceleration sensor měřící zrychlení
- Gravitation sensor udává polohu robota
- Compass Sensor umožňující robotu orientaci podle světových stran
- Color Sensor, díky kterému robot rozpozná barvu předmětu

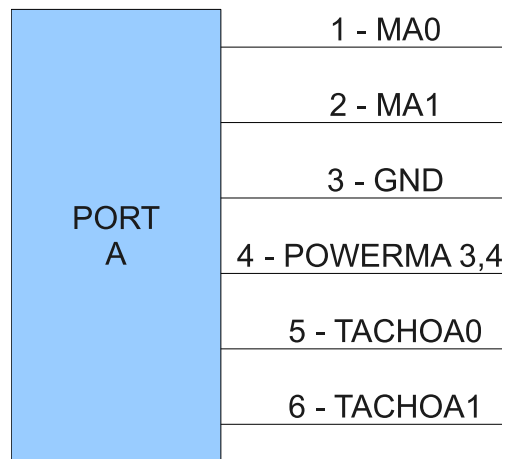
2.2. Technická specifikace

Inteligentní kostka NXT má tyto parametry:

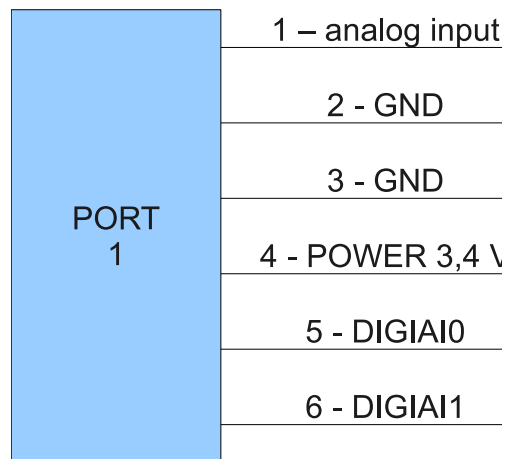
- 32-bitový ARM7 mikroprocesor (Atmel 50 MHz)
- 256 KB kapacita disku typu flash memory
- 64 KB operační paměti typu RAM
- pomocný 8-bitový procesor AVR 4 MHz, 4 KB flash, 512 B RAM (řadič vstupů a výstupů)
- černobílý maticový LCD display s rozlišením 100x64 pixelů
- 8-bitový zvukový systém s reproduktorem
- komunikační port USB 2.0
- komunikační bezdrátové rozhraní Bluetooth Class II, V2.0
- tři napájené porty pro řízení motorů nebo jiných výstupních zařízení
- čtyři napájené vstupní porty pro připojení digitálních nebo analogových senzorů
- Porty jsou založeny na průmyslové sběrnici I²C, takže lze připojovat další periférie jiných výrobců
- napájecí blok pro vložení šesti baterií typu AA nebo dobíjecího akumulátoru

Výstupní porty jsou navrženy jako šestipinové konektory, které poskytují napájení motorů, řízení rychlosti jejich otáčení pulzně šířkovou modulací a zároveň snímání otáček pro potřeby precizní regulace (viz obrázek 1. na straně 11).

Vstupní porty používají stejné konektory, ale jejich zapojení se trochu liší. Pin 1 měří analogový signál (el. napětí), takže je vhodný pro připojení jednoduchých senzorů vlastní výroby, nebo na měření vlhkosti. Můžeme zde například připojit potenciometr 50K Ω s lineární charakteristikou a měřit úhel natočení nebo hladinu plovákem. Frekvence snímání vzorků je 333 Hz, takže se dají snímat i rychle se měnící signály. Vlastní měření provádí 10-bitový AD převodník. Dále porty poskytují napájecí napětí 4.3 V pro aktivní senzory. Komunikace s digitálními senzory probíhá na portech 1 až 3 pomocí standardizované průmyslové sběrnice I²C rychlostí 9600 bps. Na posledním portu je implementována sběrnice RS-485, která podporuje vysokorychlostní komunikaci (921.6 Kbps) mezi mnoha současně připojenými zařízeními. V současné době společnost LEGO nevyrobí žádné zařízení, které by této sběrnice využívalo. Schéma zapojení je na obrázku 2. na straně 11.



Obrázek 1. Schéma zapojení výstupního portu NXT



Obrázek 2. Schéma zapojení vstupního portu NXT

2.3. Komunikace

Kostka NXT má několik možností komunikace. Základním rozhraním je USB. Po spojení s osobním počítačem přes něj můžeme nahrát a nainstalovat nový firmware, přenášet programové i datové soubory, spouštět a zastavovat programy nebo číst stavy vstupů i výstupů a logovat je. Nevýhodou ale je, zejména u experimentů s pohyblivými se roboty, nutnost propojení kabelem.

Pro experimenty s mobilním robotem je lepší využít rozhraní Bluetooth. Pomocí něho dokáže robot komunikovat nejen s počítačem, ale i s dalšími přístroji jako PDA nebo mobily. Velice zajímavá je také možnost spolupráce až čtyř kostek NXT v režimu Master-Slave. Jedna kostka řídí pomocí povelů až tři další a můžeme tak vytvářet rozsáhlejší projekty s mnoha vstupy a výstupy. Nevýhodou je nutnost nejprve navázat spojení a spárovat zařízení pomocí pinu. Rozhraní Bluetooth je použito například v projektu Ovládání brány, popsáném v kapitole 7.4.2. kde druhá kostka NXT slouží jako dálkový ovladač.

3. Programovací prostředky

Další vlastností, která činí tuto stavebnici tak populární a vhodnou pro výukové účely, je obrovská řada možností programování. Jednak s použitím originální verze firmware, nebo po nahrání speciálních verzí, které vznikly po uvolnění zdrojových kódů originálního firmware společností LEGO jako opensource.

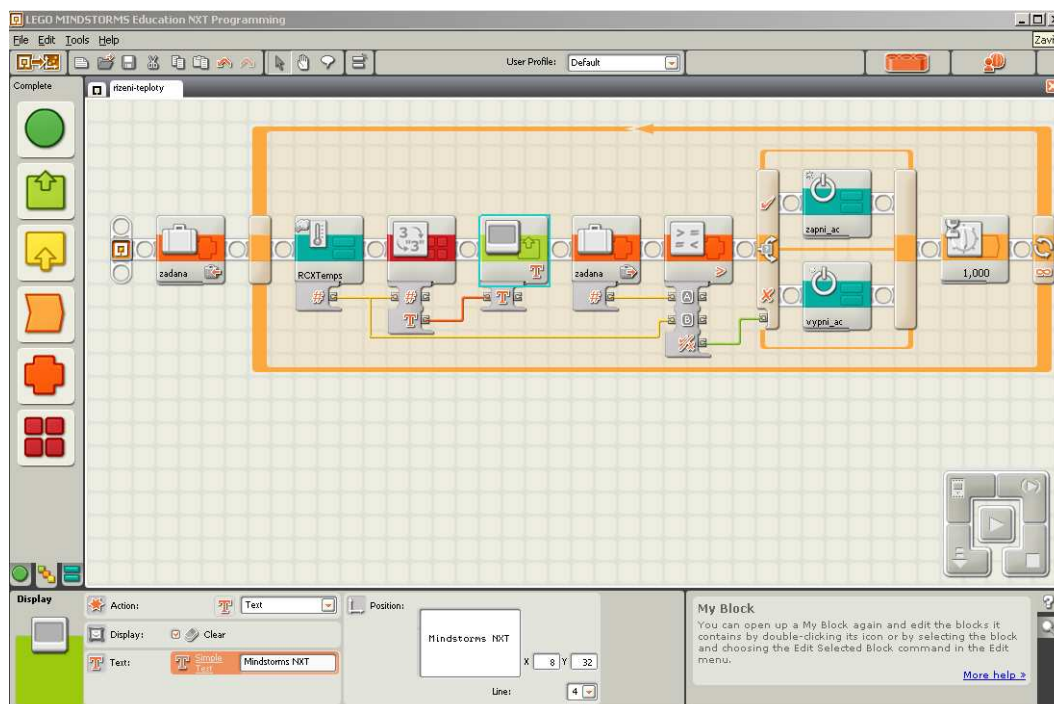
Výčet všech způsobů programování, které jsou momentálně dostupné:

- Základní vývojové prostředí dodávané se stavebnicí NXT-G s ikonografickým programováním
- NBC - projekt umožňující programování NXT v assembleru
- NXC - projekt umožňující programování NXT v jazyce podobném jazyku C
- RobotC - programování v jazyce C
- National Instrument LabView Toolkit - ikonografické programování v prostředí LabView
- LeJOS NXJ - programování robota v jazyce Java
- pbLua - programování v jazyce Lua (obdoba klasického Basicu)
- nxtOSEK - programování v jazyce ANSI C/C++

Po zvážení všech možností jsem nakonec do výukového materiálu použil dva programovací prostředky. Prvním z nich je originální vývojové prostředí dodávané spolu se stavebnicí LEGO Mindstorm NXT. Toto prostředí je jednoduché, intuitivní a díky způsobu programování pomocí ikonografických bloků velmi vhodný pro žáky základních škol. Druhým prostředkem je speciální firmware LeJOS, pro který se programy píšou v jazyce Java. Narozdíl od originálního softwaru je tento projekt multiplatformní, takže se dá použít i v operačním systému Linux. Jazyk Java je čistě objektový a jeho použití se hodí pro studenty středních škol.

3.1. Vývojové prostředí LEGO Mindstorm NXT-G

Základní myšlenkou programovacího prostředí Lego Mindstorm NXT je sestavení programu z grafických bloků. Ty mají určitý význam a vlastnosti. Umí přijímat data z jiných bloků a zase naopak jim data předávat pomocí tzv. datových vodičů. Stejný princip používají profesionální systémy pro řízení technologických procesů. Na první pohled to vypadá jen jako klasické procedurální programování, kde jsou příkazy nahrazeny ikonami a místo proměnných se používají datová propojení. Avšak navíc je zde možné vytvářet paralelní větve, které jsou vykonávány



Obrázek 3. Okno prostředí LEGO NXT-G s otevřeným programem

současně, nezávisle na sobě. Tedy zjednodušená obdoba vícevláknového programování.

Práce probíhá ve třech fázích. Prvním krokem je sestavení robota z řídicí kostky NXT, různých senzorů, servomotorů a dalších kostek stavebnice LEGO Technics. Potom následuje napsání vlastního programu v programovacím prostředí a nakonec se program zkompiluje a nahraje pomocí USB kabelu, případně Bluetooth spojení do řídicí kostky NXT, a zde může být spuštěn.

Program začíná startovním bodem. V jeho vlastnostech můžeme uvést popis programu. Standardně se program odvíjí po jedné hlavní větvi. Ze startovacího bodu můžeme vést další dvě zcela nezávislé větve. Navíc je možné odbočit další větve v libovolném místě programu pomocí klávesy SHIFT. Díky tomu může náš program vykonávat několik nezávislých činností současně. Zároveň mohou tyto programové části komunikovat pomocí proměnných. Jedná se tedy o zjednodušenou formu vícevláknového programování.

Prostředí nabízí i možnost vkládání textových komentářů na libovolné místo v programu, aby i rozsáhlejší projekt zůstal přehledný. Pracovní plochu programu není sice možné zvětšovat ani zmenšovat, ale pomocí nástroje pan lze plochu pohybovat a prohlížet si tak i rozsáhlejší program, který přesahuje okno aplikace.

3.1.1. Datové vodiče

Datové vodiče přenášejí data mezi bloky programu. Řada bloků vyžaduje propojení datových vodičů, aby vůbec fungovaly. Například výstup z bloku Random může být využit jedině přes datový vodič. Datový vodič se vytvoří "vytažením" myši z konektoru na datovém rozbočovači bloku. Skoro všechny bloky mají datové rozbočovače a umožňují propojování datovými vodiči. Datový rozbočovač bloku se otevře kliknutím na zarážku umístěnou v levém dolním rohu bloku. Datové vodiče přenášející informace do bloku (tedy zajišťují vstup dat) se připojují na levé straně datového rozbočovače. Datové vodiče odesílající data ven z bloku do jiného bloku (tedy zajišťují výstup dat) se připojují do konektorů na pravé straně datového rozbočovače. Datový vodič musí být zapojen do konektoru odpovídajícího datového typu, jinak se zabarví šedou barvou a nebude možné program zkompileovat.

Data musí být také v rozsahu, který daný konektor přijímá. Pokud jsou přijatá data mimo možný rozsah konektoru, blok je ignoruje nebo změni jejich hodnotu, aby rozsahu vyhovovala. Konektory, které přijímají jen omezený rozsah několika hodnot (například: pouze 0, 1, nebo 2), ignorují hodnoty mimo tento rozsah. Konektory, které akceptují velký rozsah vstupních hodnot (například: 0 – 100), změni hodnotu na nejbližší možnou hodnotu v povoleném rozsahu. Například když konektor Power bloku Move přijme hodnotu 150, blok ji změni na 100, tedy nejvyšší hodnotu povoleného rozsahu pro výkon motoru v procentech.

Pokud má vstupní konektor odpovídající výstupní konektor, vstupní data se přenášejí ze vstupu na výstup bez jakékoliv změny. V tomto případě můžeme použít výstupní konektor jen tehdy, je-li na vstupní konektor přiveden datový vodič z jiného bloku. Pokud se pokusíme zapojit datový vodič z takového výstupního konektoru, který nemá připojen vstup, do jiného bloku, bude hlásit chybu šedým zbarvením.

barva	popis
žlutá	Datový vodič přenášející číselné hodnoty
zelená	Datový vodič přenášející logické hodnoty
oranžová	Datový vodič přenášející textové informace
šedá	Špatně zapojený datový vodič

Tabulka 1. Barevné značení datových vodičů

3.1.2. Výstupní programové bloky

Tyto bloky jsou umístěny na paletě output. Slouží k ovládání výstupních zařízení:

- Blok Move (servomotor)

- Blok Sound (reproduktor)
- Blok Display
- Blok Výstup přes Bluetooth
- Blok Move starší typ motoru (RCX)
- Blok signalizační světlo (RCX)

Blok Move

Umožňuje řízení motorů připojených na výstupní porty A, B a C kostky NXT. Lze řídit jednotlivé motory, nebo použít režim steering wheel, kdy se nastavují dva motory současně a můžeme přímo řídit zatačení jezdícího robota. Blok umožňuje nastavit, který motor chceme ovládat, jeho výkon a počet otáček, které má vykonat. Dále lze nastavit směr otáčení (direction) , výkon motoru (power), doba otáčení (duration) a chování motoru po ukončení činnosti bloku (next action) : brzda (brake) nebo volnoběh (coast). Můžeme nastavit také pouze zastavení motoru (stop), nebo spustit otáčení na neomezenou dobu (unlimited) a zastavit motor až dalším blokem Move v jiné části programu.

Blok Sound

Tento blok umí generovat tóny přes vestavěný reproduktor kostky NXT, nebo přehrávat zvukové soubory ve formátu wav. Je možné nastavit režim tón (tone) nebo soubor (file), spuštění nebo zastavení přehrávání a hlasitost (volume). Funkce čekat na dokončení (wait for completion) určuje chování programu po spuštění zvuku.

Blok Display

Umožňuje zobrazit na display kostky NXT libovolný text, obrázek, nebo vykreslit tvar. Kombinací více těchto bloků v programové větvi docílíme složitějšího zobrazení. Blok má i volby Reset a Clear pro vymazání obsahu displeje. Obsahuje datový rozbočovač umožňující dynamické zobrazování hodnot z jiných bloků připojením datovými vodiči.

Blok Výstup přes Bluetooth

Tento blok umožňuje odesílat data přes rozhraní Bluetooth. Umí otevřít až 4 spojení s různými Bluetooth zařízeními nebo počítači. Data odesílá přes tzv. mailové schránky (message box). Data se na tento blok napojují přes datové vodiče vedoucí z jiných bloků. Každá mailová schránka může obsahovat až 5 zpráv, pak musí být odeslána. Je to obdoba paketů v nižší úrovni komunikace. Datovými konektory jsou číslo spojení, číslo schránky, textové proměnné, číselné hodnoty a logické proměnné. Můžeme tedy přes datový rozbočovač určit číslo spojení i číslo mailové schránky.

Blok signalizační světlo (RCX)

Tento blok ovládá signalizační lampu připojenou přes port A, B nebo C. Může také sloužit k ovládání silových obvodů přes oddělovací fototranzistor.

3.1.3. Vstupní programové bloky

Tyto bloky čtou data ze vstupních portů, kde se připojují senzory:

- Blok Touch sensor (senzor doteku)
- Blok Sound sensor (senzor hladiny hluku)
- Blok Light sensor (senzor intenzity osvětlení)
- Blok Ultrasonic sensor (ultrazvukové čidlo vzdálenosti)
- Blok NXT buttons (tlačítka na kostce NXT)
- Blok Rotation sensor (senzor otáček vestavěný v motorech NXT)
- Blok Timer (časovač)
- Blok Receive Message (data přijatá přes Bluetooth)

Blok Touch sensor

Tento vstupní blok zjišťuje stav dotykového senzoru připojenému ke vstupnímu portu NXT. Jakmile program dospěje k tomuto bloku, je načten stav senzoru a může být předán datovým vodičem jinému bloku následujícím v programu. Nastavujeme číslo vstupního portu 1-4, sledovaný stav stlačeno (pressed), uvolněno (released) nebo rychlý dotek a uvolnění bumped).

Blok Sound sensor

Tento vstupní blok měří úroveň okolního hluku zvukového senzoru připojenému ke vstupnímu portu NXT. Jakmile program dospěje k tomuto bloku, je načten stav senzoru a může být předán datovým vodičem jinému bloku následujícím v programu. Nastavujeme číslo vstupního portu 1-4. Výstup hodnoty z datového přepínače je buď logický (porovnání s nastavenými mezemi) , nebo číselný (přímo naměřená úroveň).

Blok Light sensor

Tento vstupní blok měří úroveň osvětlení světelného senzoru připojenému ke vstupnímu portu NXT. Jakmile program dospěje k tomuto bloku, je načten stav senzoru a může být předán datovým vodičem jinému bloku následujícím v programu. Umí měřit přirozené světlo v prostředí nebo odražené světlo z jeho vlastního světelného zdroje. Nastavujeme číslo vstupního portu 1-4. Výstup hodnoty

z datového přepínače je buď logický (porovnání s nastavenými mezemi) , nebo číselný (přímo naměřená úroveň).

Blok Ultrasonic sensor

Tento vstupní blok měří vzdálenost od překážky ultrazvukového senzoru připojenému ke vstupnímu portu NXT. Jakmile program dospěje k tomuto bloku, je načten stav senzoru a může být předán datovým vodičem jinému bloku následujícím v programu. Maximální dosah je asi 250 cm. Jako jednotky můžeme zvolit palce nebo centimetry. Nastavujeme číslo vstupního portu 1-4. Výstup hodnoty z datového přepínače je buď logický (porovnání s nastavenými mezemi) , nebo číselný (přímo naměřená úroveň).

Blok NXT buttons

V programu je možné snímat stav i vlastních tlačítek kostky NXT. Tento blok sejme aktuální hodnotu (TRUE nebo FALSE) vybraného tlačítka a odešle ji datovým vodičem jinému bloku programu. Tlačítka na kostce jsou čtyři : Šipka doleva, šipka doprava, Enter a Storno.

Blok Rotation Sensor

Tento blok načítá počet otáček ze senzorů vestavěných v motorech NXT a odešle je přes datový vodič. Umožňuje také počítadlo otáček ve vhodnou chvíli resetovat a začít tak počítat znovu od nuly. Otáčky je možno získávat velmi přesně ve stupních nebo pouze v počtu otáček. Nastavujeme výstupní port, na který je připojen sledovaný motor, směr otáčení, který chceme sledovat, a volíme stupně nebo počet otáček. Výstup hodnot z datového přepínače je buď logický (porovnání s nastavenými mezemi) , nebo číselný (přímo naměřená úroveň).

Blok Timer

Při spuštění programu jsou automaticky nastartovány tři vestavěné časovače, které můžeme číst a resetovat pomocí tohoto bloku. Jejich hodnoty můžeme využít v jiném bloku pomocí datových vodičů. Volíme číslo vybraného časovače. Výstup hodnot z datového přepínače je buď logický (porovnání s nastavenými mezemi) , nebo číselný (přímo stav časovače).

Blok Receive Message

Tento blok umožňuje číst data z rozhraní Bluetooth a předávat je dalším blokům přes datové vodiče. Nastavuje se typ zprávy (text, logická hodnota, číselná hodnota) a číslo mailové schránky. Na výstupu datového konektoru jsou pak data podle typu zprávy.

3.1.4. Řídící programové bloky

Řídící programové bloky umožňují ovlivňovat vykonávání programu:

- Blok Wait (čekání na určitou hodnotu daného senzoru, nebo jen daný čas)

- Blok Loop (opakování části programu dokud není splněna podmínka)
- Blok Switch (větvení programu podle stavu senzoru nebo proměnné)
- Blok Stop (zastavení běhu programu)

Blok Wait

Tento blok způsobí dočasné zastavení vykonávání programu a čekání na splnění určité podmínky. Tato podmínka může být daná určitou hodnotou vybraného senzoru nebo jen časem.

Blok Loop

Tento blok způsobí opakování sekvence programového kódu. Počet opakování může být neomezený (nekonečný cyklus) , nebo může být dán podmínkou. Dovnitř umisťujeme sekvenci bloků, jejíž vykonávání se má opakovat. Cyklus, daný podmínkou se opakuje tak dlouho, dokud není podmínka splněna. Možností, jak nastavit podmínku, je mnoho. Může to být překročení hodnoty libovolného senzoru, nebo stav proměnné přivedené datovým vodičem, případně časovačem, nebo může být nastaven přesný počet opakování (obdoba cyklu FOR). Vnitřní datový konektor dovoluje použít v programových blocích uvnitř cyklu hodnotu řídicí proměnné.

Blok Switch

Tento blok umožňuje rozvětvit program podle určité podmínky. Vykoná se vždy pouze jedna větev, tak jak to je u konstrukce IF-THEN-ELSE v klasickém programování. Blok se záměrně jmenuje Switch, protože nezahrnuje pouze tuto klasickou programovací konstrukci, ale dovoluje vytvářet libovolný počet větví, tak jak je tomu u příkazu case, nebo switch v klasických programovacích jazycích. Tato možnost je přístupná, pokud se vypne základní režim flat view. Jednotlivé větve se pak zobrazují jako záložky. Hodnota se do bloku přivádí pomocí vstupního datového vodiče.

Blok Stop

Tento blok zastaví program, všechny motory a zhasne všechna signální světla. Nemá žádné parametry, pouze je možné připojit vstupní datový vodič s hodnotami TRUE (zastavit) nebo FALSE (nezastavovat).

3.1.5. Datové bloky

Datové bloky jsou určeny pro manipulaci s daty jak číselnými, tak logickými.

- Blok Logic (logické operace)
- Blok Math (matematické operace)

- Blok Compare (porovnávání hodnot)
- Blok Range (rozsah hodnot)
- Block Random (náhodné číslo)
- Blok Variable (proměnná)
- Blok Logic

Blok Logic

Blok Logic umí provádět logické operace s hodnotami přivedenými datovými vodiči na vstupní straně datového rozbočovače. Výsledek logické operace je předán na výstupní datový vodič. Možné operace jsou OR, AND, OR, XOR a NOT.

Blok Math

Blok Math umí provádět matematické operace s číselnými hodnotami přivedenými datovými vodiči na vstupní straně datového rozbočovače. Výsledek logické operace je předán na výstupní datový vodič. Možné operace jsou - sčítání, odčítání, násobení a dělení.

Blok Compare

Blok Compare umožňuje porovnávání hodnot přivedených datovými vodiči na vstupní straně datového rozbočovače. Výsledek porovnání (logická hodnota TRUE nebo FALSE) je předán na výstupní datový vodič. Možná porovnání jsou větší než, menší než a rovnost.

Blok Range

Tento blok zjišťuje, jestli hodnota přivedená na vstup vyhovuje danému rozsahu, který může být nastaven napevno, nebo jeho meze mohou být přivedené rovněž na vstup vodiči a a B. Výstupem je logická hodnota TRUE – vyhovuje, nebo FALSE – nevyhovuje.

Blok Random

Tento blok vygeneruje náhodné číslo v daném rozsahu, které je pak využitelné v dalších blocích připojených na výstupní datový vodič. Rozsah generovaných hodnot je možno nastavit napevno, nebo pomocí vstupních datových vodičů A a B.

Blok Variable

Tento blok umožňuje zavést do programu proměnnou tak jako u klasického programování. Je zobrazován jako kufr, do něhož si můžeme hodnotu odložit a zase si ji z něho později odebrat. Nejprve se vloží blok Variable s režimem Input (vstup), nastaví se typ proměnné (text, číslo nebo logická hodnota) a její

název. Je také možné nastavit počáteční hodnotu. V dalším bloku Variable je možné stejnou hodnotu (jednoznačně určenou jejím názvem) použít pro jiný blok napojením výstupním datovým vodičem.

3.1.6. Speciální bloky

Speciální bloky slouží k dalším manipulacím se senzory, motory a daty.

- Blok Text (spojování textových řetězců)
- Blok Number to Text (převod čísla na text)
- Blok Keep alive (nastavení času usínání)
- Blok File access (přístup k souborům)
- Blok Calibration (kalibrace světelného a zvukového čidla)
- Blok Reset motor (reset počítadel otáček motorů)

Blok Text

Tento blok umí spojit až tři textové řetězce do jednoho. Využívá k tomu 3 textové vstupní datové konektory (A,B,C) a pro výstup výstupní datový konektor rovněž typu text.

Blok Number to text

Tento blok převádí číselné hodnoty na textové. To je nutné například pro zobrazení číselné hodnoty na displeji NXT nebo jejich odesílání přes Bluetooth. Na vstupní datový konektor přivedeme číselnou hodnotu a na výstupním dostaneme textovou.

Blok Keep alive

Kostka NXT podporuje jako každý moderní počítačový systém režim spánku. V nastavení kostky je hodnota času, po jehož uplynutí v době nečinnosti kostka přechází do režimu spánku. Tento blok slouží k nastavení doby usnutí. Používá se v programech, kde očekáváme delší čekací smyčky, než je standardní doba usnutí kostky NXT.

Blok File access

Pomocí tohoto bloku můžeme ukládat data do souborů v kostce NXT. Po zapsání dat do souboru musíme v dalším bloku File access tento soubor zavřít. Pak je teprve možné tento soubor znovu otevřít pro čtení, případně smazat. Parametry bloku jsou - režim použití souboru, které může být READ, WRITE, CLOSE, nebo DELETE. Dále je to jméno souboru a typ souboru. Rozlišují se pouze dva typy souborů: textový a soubor obsahující číselné hodnoty.

Blok Calibration

Blok Calibration umožňuje přizpůsobit světelné a zvukové senzory okolním podmínkám. Nastavuje minimální a maximální hodnotu rozsahu měření. Používá se na začátku programů, když je potřeba přesně určit rozsah měření prostředí, ve kterém se robot nachází.

Blok Reset motor

Interaktivní motory stavebnice mají vestavěnou automatickou korekci chyb, ale pokud je používáme v režimu volného pohybu po zastavení, měření může vykazovat chyby. Tento blok slouží k resetu motoru a způsobí počítání otáček znovu od nuly. Používá se v místě programu, od kterého vyžadujeme přesné změření otáček. Nastavuje se, které motory se mají resetovat.

3.1.7. Vlastní bloky

Aby byl program přehledný a abychom mohli často používané sekvence příkazů používat snadno opakovaně, můžeme si úsek programu uložit jako vlastní blok. Úsek programu se označí myší a zvolíme v menu Make New My Block. Tento blok pojmenujeme a přiřadíme mu ikonu. Potom ho kdykoliv najdeme v paletě My Blocks. Dokonce je možné vytvořit vlastní blok s datovými vodiči, a to jak vstupními, tak výstupními. Je třeba připravit úsek programu, ze kterého chceme nový blok vytvořit a přidat libovolné další bloky s vhodnými datovými konektory před tento úsek nebo za něj. Tyto konektory propojit s bloky uvnitř připraveného úseku. Pokud se pak vytváří vlastní blok z úseku programu, do kterého vstupují nebo vystupují datové vodiče, stanou se jeho vstupními nebo výstupními konektory.

3.2. Programování v Javě - firmware LeJOS

Název projektu je odvozen od španělského slova lejos, což znamená v překladu daleko, proto se vyslovuje [lechos]. Jde ale spíš o slovní hříčku, neboť zkratka znamená Lego java operating system a jde o malou implementaci Java virtual machine portovanou v roce 2006 na kostku NXT. Má velice dobře propracované API, které popisuje veškeré použitelné třídy při tvorbě programů. Stejně jako samotná Java je multiplatformní, takže můžeme pracovat v libovolném operačním systému. V učebním materiálu je popsána instalace a použití v operačním systému Linux. Tím se také částečně odstraní problém nepoužitelnosti originálního software Lego NXT-G v tomto operačním systému.

3.2.1. Klíčové vlastnosti

- čistě objektově orientovaný programovací jazyk (Java)

- preemptivní multitasking realizovaný pomocí vláken programu
- pole, včetně multi-dimenzionálních
- Rekurze
- Synchronizace
- Výjimky
- datové typy Java např. float, boolean, long a String
- Většinu metod třídy java.lang, java.util a java.io
- Speciální třídy pro precizní řízení robota např. Tachopilot
- výborně dokumentované API

3.2.2. Výuka objektového programování

Použití robota s LeJOS se jeví jako velice vhodné pro výuku objektově orientovaného programování, neboť se zde používají reálné objekty (například motor). Díky tomu je pro studenty mnohem snadnější tuto metodiku programování pochopit. V učebním materiálu studenti formou projektů objeví postupně všechny základní vlastnosti této problematiky.

- Objekty – prvky reality seskupené do logického celku. Například objekt motor, jeho vlastnosti (na kterém portu je připojen, jestli se točí) a metody (dopředu, dozadu, stop, volnoběh).
- Objekt versus třída - v Javě se definují třídy a na jejich základě se pak v programu deklarují objekty jako konkrétní reprezentanti tříd. Například senzor S1 je nový reprezentant třídy Lightsensor.
- Konstruktor - speciální metoda třídy, která se provádí při vzniku objektu. V Javě má stejný název jako celá třída. Například TouchSensor touch1 = TouchSensor(SensorPort.S1) vytvoří objekt touch1, který reprezentuje dotykový senzor připojený na port 1. Toto přiřazení provádí právě metoda Touchsensor, která je konstruktorem této třídy.
- Abstrakce – nepotřebujeme znát způsob, jakým objekt pracuje uvnitř. Můžeme například zavolat metodu Motor.Stop a motor zastaví.
- Zapouzdření – K objektu se přistupuje pouze přes jeho veřejné rozhraní, není možné zasahovat do jeho vnitřních proměnných. Například u motoru nastavovat přímo jeho napájecí napětí.

- Skládání – Objekt může obsahovat jiné objekty. Například třída Regulace obsahuje třídu Zobrazovac.
- Delegování – Objekt využívá služeb jiných objektů. Například třída Regulace volá metodu getCelsius třídy RCXTemperature.
- Dědičnost – objekt může dědit vlastnosti a metody od nadřazeného objektu ve stromové struktuře a přidávat k nim nové. Například třída Motor rozšiřuje třídu BasicMotor, od které dědí základní metody jako dopředu, dozadu a stop a přidává k nim metodu rotace, jež otočí motorem o daný počet stupňů.
- Parametrický polymorfismus – objekt může mít několik metod se stejným názvem, ale různými parametry. Například metoda LCD.drawString(String str, int x, int y, boolean invert) vypíše text na display inverzně a LCD.drawString(String str, int x, int y) normálně.
- Kontextový polymorfismus - stejná metoda může mít u dvou různých podtříd odlišné chování, pokud není definována jako statická. Například třídy USB a Bluetooth jsou obě podtřídy třídy NXTCommDevice a obě mají metodu waitForConnection, která funguje u každé z nich zcela jiným způsobem.

3.2.3. Vícevláknové programování

Další zajímavou vlastností Java virtual machine LeJOS je podpora vícevláknového programování. Ta je velmi důležitá v programech, kde potřebujeme, aby robot vykonával více činností současně. Například sleduje čáru a zároveň přehrává zvuk, ovládá robotické rameno, přijímá a odesílá informace přes Bluetooth rozhraní, nebo jen sleduje stav tlačítka, aby ukončil program.

Základní použití vícevláknového programování je zde velmi snadné. Nadefinujeme si libovolnou vlastní třídu, která dědí z třídy Thread. V ní nadefinujeme metodu run, ta bude obsahovat programový kód, který má běžet souběžně s hlavním programem. V hlavním programu pak někde zavoláme metodu start naší třídy a spustí se druhé vlákno vykonávající metodu run. Takových tříd si můžeme nadefinovat libovolný počet a všechny je pak současně spustit. Tato vlákna potom mohou mezi sebou komunikovat pomocí nastavování vlastností, nebo lepším způsobem pomocí nezávislé třídy pro výměnu dat.

3.2.4. Ukázkový program

V následující ukázce Java kódu je program, který slouží pro ovládání modelu přechodu pro chodce semaforey. Implementuje dvě vlastní třídy. Třída Lamp abstrahuje ovládání lampy připojené na výstupním portu kostky NXT. Třída Zvuk

dědí od třídy Thread a nabízí metodu run, která přehrává tóny pro potřeby zrakově postižených. Rychlost přehrávání tónů je řízena vlastností setChodci, přes kterou hlavní program komunikuje s tímto vláknem. Ve vlastním programu se objevují všechny programové konstrukce, jako cyklus while, cyklus for i podmínka if. Současně demonstruje způsob čtení hodnot ze senzoru a zobrazování hodnot na displeji NXT.

Listing 1 Řízení přechodu pro chodce semaforey

```
// Program pro rizeni prechodu pro chodce semaforey.
import lejos.nxt.*;

public class semaforey3
{
    // Nadefinujeme si tridu Lamp pro ovladani svetel.
    // Vyuzijeme tridy MotorPort k ovladani lampy.
    // Konkretne metodu controlMotor,
    // ktera ma parametry (power, mode).
    // power - power from 0-100
    // mode - 1=forward, 2=backward, 3=stop, 4=float
    public static class Lamp {
        // soukrome promenne tridy:
        private MotorPort _M = MotorPort.A;
        private boolean _IsLighting;
        // Konstruktor tridy:
        public Lamp(MotorPort M) {
            _M = M;
            _M.controlMotor(0,1);
            _IsLighting = false;
        }
        // Metoda zapni
        public void On() {
            _M.controlMotor(100,1);
            _IsLighting = true;
        }
        // metoda vypni
        public void Off() {
            _M.controlMotor(0,1);
            _IsLighting = false;
        }
        // metoda vracejici stav lampy
        public boolean IsLighting() {
            return _IsLighting;
        }
    } // Konec tridy Lamp
    // Definujeme tridu Zvuk, jejiz metoda run pobezi jako
```

```

// paralelni vlakno s hlavnim programem.
// Tato trida dedi vlastnosti a metody tridy Thread.
public static class Zvuk extends Thread {
    // soukroma promenna tridy:
    private boolean _chodci;
    // Konstruktor tridy:
    public Zvuk() {
        _chodci = false;
    }
    // Metoda run vyuziva metody playTone tridy Sound
    // playTone(int aFreq, int aDurat, int aVol)
    public void run() {
        while (true) {
            Sound.playTone(500,50,100);
            if (_chodci)
                Sound.pause(300);
            else
                Sound.pause(1000);

            } // konec cyklu while
        } // konec metody run
    public void setChodci(boolean chodci) {
        _chodci = chodci;
    } // konec metody setChodci
} // Konec tridy Zvuk
// Vlastni program:
public static void main (String [] aArg)
throws Exception
{
    // Vytvorime tri nove instance tridy Lamp:
    Lamp zelena = new Lamp(MotorPort.B);
    Lamp cervena = new Lamp(MotorPort.A);
    Lamp zluta = new Lamp(MotorPort.C);
    Zvuk zvuk1 = new Zvuk();
    // Vytvorime novou instanci tridy touch sensor
    // pripojenem na vstup 1:
    TouchSensor touch1 = new TouchSensor(SensorPort.S1);
    // Spustime vlakno zvuk1:
    zvuk1.setChodci(false);
    zvuk1.start();

    // Vlastni nekonecny cyklus pro rizeni prechodu:
    while (true) {
        zelena.On();
        // Cekame na dotyk senzoru touch1,

```

```

// pokud je stisknuto tlacitko ESCAPE
// program se ukonci.
while (!touch1.isPressed()) {
    if (Button.ESCAPE.isPressed())
        System.exit(1);
}
// V cyklu, který bezi 10x zobrazujeme i
// (cislo pruchodu):
for (int i = 16; i >= 6; i--)
{
    LCD.drawString(Integer.toString(i), 2, 2);
    Thread.sleep(1000);
}
zelena.Off();
// Prechazime na zlutou
zluta.On();
// V cyklu, který bezi 5x zobrazujeme i
// (cislo pruchodu):
for (int i = 5; i >= 0; i--)
{
    LCD.drawString(Integer.toString(i), 2, 2);
    Thread.sleep(1000);
}
zluta.Off();
// Prechazime na cervenou:
cervena.On();
zvuk1.setChodci(true);
Thread.sleep(10000);
// Prechazime zpět na zlutou:
cervena.Off();
zvuk1.setChodci(false);
zluta.On();
Thread.sleep(5000);
zluta.Off();
} // konec sekvence prikazu nekonecneho cyklu
} // konec metody main
} // konec tridy semafor3

```

4. Dostupné učební materiály

Ke stavebnici Lego Mindstorm NXT existuje celá řada učebních materiálů. Tím nejzákladnějším je dokumentace výrobce k vlastní stavebnici a nápověda k dodávanému vývojovému prostředí. Toto prostředí nabízí i jednoduchého multi-mediálního průvodce pro seznámení se základními možnostmi stavebnice. Proto se základy použití stavebnice ve vlastním materiálu zabývám jen okrajově, spíše se orientuji na praktické ukázky projektů, které využívají i pokročilejší možnosti produktu.

4.1. Literatura

Ke stavebnici vyšlo mnoho knih. Některé jsou jen stručnými příručkami, některé obsahují rozsáhlé řešení problémů a jiné se zaměřují na konkrétní nasazení robota v určité soutěži. Všechny jsou ale dostupné pouze v anglickém jazyce, žádná z nich nebyla přeložena do češtiny.

V seznamu literatury je uvedena kniha Building robots with LEGO Mindstorm NXT [1], která je velmi obsáhlá a zabývá se jak konstrukčními možnostmi stavebnice, tak problematikou softwarovou. Je ovšem spíše pro pokročilejší uživatele, neboť používá jako programovací prostředek jazyk C a v konstrukci modelů zabíhá až do technických detailů a fyzikálních principů jednotlivých problémů.

Například řeší problematiku chůze robotů nebo připojení 16 dotykových senzorů současně. Dále uvádím v seznamu literatury knihu LEGO MINDSTORMS NXT-G Programming Guide [2], která je jakousi základní příručkou pro tvorbu programů v prostředí NXT-G.

4.2. Webové stránky

Na internetu je situace podobná jako u knih. Existuje mnoho stránek zabývajících se touto stavebnicí, všechny jsou v anglickém jazyce.

Společnost Lego provozuje vlastní webové stránky na adrese <http://mindstorms.lego.com>, kde mohou uživatelé sdílet svoje projekty, nalezneme jich tady celou řadu. Stejně tak na dalších stránkách <http://www.nxtprograms.com>, kde jsou dostupné ke stažení náměty, konstrukční návody, programy a další materiály pro práci s roboty. Stránky <http://legoengineering.com> jsou zaměřené na netradiční způsoby řešení pohybu robotů.

4.3. Školení a kurzy

Společnost Eduxe působící na českém trhu se zabývá prodejem a dalšími službami spojenými se stavebnicemi řady LEGO education. Má akreditaci MŠMT ČR k dalšímu vzdělávání pedagogických pracovníků. Nabízí i základní školení

pro uživatele LEGO MINDSTORMS Education NXT. Bližší informace jsou dostupné na webových stránkách společnosti <http://www.eduxe.cz>.

V současné době probíhá i evropský projekt LEGO ve výuce informatiky a fyziky na základní škole Velké Bílovice. Výstupem projektu by měly být nejen e-learningové kurzy pro učitele, ale také učební materiály a montážní návody. Bližší informace jsou postupně zveřejňovány na stránkách <http://www.zsvelkebilovice.cz/lego2010/>.

5. Robotické soutěže

Ve světě existuje nespočet robotických soutěží, které se liší pravidly, obtížností i věkovým omezením účastníků. Vybral jsem zde na ukázkou několik soutěží, které jsou nejen velmi zajímavé a atraktivní pro žáky i studenty, ale také dostupné v našem regionu.

5.1. First Lego League

Soutěž FIRST LEGO LEAGUE vznikla v roce 1998 v USA, od roku 2001 běží v Německu, od roku 2002 v Rakousku, od roku 2003 ve Švýcarsku a od roku 2005 v Maďarsku. V České republice se pořádá regionální kolo od roku 2006. Organizátorem je Asociace mladých debrujárů.

Vizí programu FIRST LEGO LEAGUE je:

- otevřít dětské oči zábavě a vzrušení z vědy a techniky
- rozvíjet u dětí schopnost týmové práce
- stimulovat u dětí užití kreativity při řešení problémů

Podmínkou pro zapojení do soutěže je vytvoření týmu 5 – 10 dětí do 16 let, zaplacení startovního a pořízení stavebnice robota LEGO Mindstorm education. Při soutěži se hodnotí čtyři soutěžní kategorie:

- Robotí hra - zde týmy konstruují a programují robota, aby samostatně splnil co nejvíce úkolů na herním stole v časovém limitu 150 sekund.
- Týmová práce - týmy jsou testovány, jestli v průběhu celé soutěže pracovaly jako tým a zda se děti naučily spolupracovat při řešení problémů.
- Design robota se hodnotí jako samostatná kategorie. Jde o to, jestli při řešení soutěžních úkolů vymysleli členové týmu nějaké zajímavé řešení, ať už při konstrukci robota a jeho příslušenství, tak při jeho programování.
- Výzkumný projekt – podmínkou úspěšného absolvování soutěže je vytvoření inovativního projektu na vyhlášené téma a jeho prezentace před porotou soutěže.

Každý ročník soutěže bývá zahájen v září zveřejněním soutěžních úkolů a témat pro příslušný rok. Na začátku listopadu se pak konají jednotlivá regionální kola. Do soutěže se každoročně zapojuje přes 130 000 dětí z celého světa. Z každého národního kola postupuje vítěz do semifinále pro danou oblast. Vítězové semifinále pak postupují do celosvětového finále, které se koná v USA.

Tématem v roce 2009 bylo Smart move, tedy inovativní řešení dopravních problémů. Tématem loňského ročníku bylo Body forward, neboli řešení projektů

z oblasti biomedicíny. Podle tématu jsou stanoveny i soutěžní úkoly pro robotí hru. Národní kola prvních dvou ročníků se uskutečnila v Praze. Třetí a čtvrtý ročník pořádaný Gymnasiem Jeseník se uskutečnil v Jeseníku. Loni počet týmů už překročil hranici pro vytvoření dvou samostatných regionálních kol, které se konaly v Praze a v Olomouci.

Robotí hra je nejdůležitější část soutěže, jednotlivé týmy konstruují a programují robota tak, aby splnil v čase 150 sekund co nejvíce soutěžních úkolů. Zadání je pro všechny kola na celém světě jednotné, stejně jako pravidla hry. Po zaplacení startovného dostane každý tým mapu herního stolu a kostky na stavbu prvků mapy. Jedním z předchozích úkolů bylo například přemístění ledové bóje z průmyslové do ledové oblasti na herním plánu.

Nevýhodou této soutěže je její velká náročnost na organizaci a udržení motivace žáků. Na přípravě se musí podílet alespoň dva učitelé a věnovat jí spoustu času, neboť úkolů je mnoho a vše se musí připravit během září a října. Na začátku listopadu probíhá vlastní regionální finále, kde soutěží až dvacet týmů. Pokud tým neobsadí první místo, soutěž pro něj skončila. Finančně je celá akce také dosti náročná. Je výhodné sehnat sponzora, který by zaplatil startovné, dopravu a dresy.

5.2. Eurobot

Eurobot je původem francouzská soutěž pro amatérské týmy mladých fanoušků robotiky nebo pro týmy, které pro mladé takovýto projekt organizují. Soutěž Eurobot nechce být nesmiřitelným bojem, naopak vše probíhá v přátelském a sportovním duchu. Tým soutěží se svým autonomním robotem na hřišti o rozměrech cca 3 x 2 metry proti dalšímu týmu (který má opět svého robota). Jeden zápas trvá 90 sekund a roboti při něm samostatně plní svůj úkol bez toho, že by byli řízeni člověkem nebo vzdáleným počítačem; vše potřebné si roboti nesou s sebou a také musí samostatně přemýšlet a rozhodovat. Přesná pravidla se každý rok mění (téma, počet robotů, velikost hřiště atd.), takže i nově vzniklé týmy mají velkou naději na úspěch. Vždy na začátku podzimu jsou vyhlášena pravidla pro další ročník, na jaře probíhají národní kola a následně před létem pak mezinárodní finále.

Soutěž má kategorii Starter, kde jsou pravidla téměř stejná jako pro hlavní kategorii Eurobot. Jediný zásadní rozdíl je v ovládní robota po drátě (pro Eurobot jsou roboti autonomní) a v omezení věku soutěžících této kategorie (do 18 let resp. konce střední školy). Vzhledem k nastavení pravidel v podstatě jen stačí postavit autíčko, které bude schopno před sebou něco tlačit - herní prvky na ploše je možné do cílové zóny dotlačit po zemi a získat tak body! Pro mladší zájemce se tak soutěž otevírá tak, že se mohou snadno zúčastnit a v dalších letech využít své zkušenosti třeba už v hlavní kategorii.

České kolo pořádá Katedra softwarového inženýrství MFF UK a občanské sdružení Robonika. Z tohoto kola postupují vždy tři týmy do mezinárodního

kola. Loňským tématem byla sklizeň úrody. Roboti pomáhají lidem se simulovanou sklizní ovoce, zeleniny a obilovin. Ten robot, který sklídí největší množství úrody, vyhraje. Do soutěže se přihlásilo 35 týmů pocházejících nejen z celé České republiky, ale i ze Slovenska, Polska a Francie.

5.3. Robot Challenge

Robot Challenge je rakouská soutěž autonomních robotů, která se koná každoročně začátkem jara ve Vídni. Soutěží se v mnoha kategoriích od jízdy po čáře (obrázek 5. na straně 34), přes sběr barevných puků (obrázek 4. na straně 33), sumo různých velikostí, až po volnou jízdu a sprint humanoidních robotů. Registrace je za mírný poplatek a není zde žádné věkové omezení (přesněji 9 až 99 let).

Soutěž Robot Challenge exponenciálně roste a snaží se o zápis do Guinnessovy knihy rekordů. Stejně jako minulé ročníky ji pořádá rakouská společnost pro inovativní počítačové vědy (InnoC.at). Loni už zabrala dva velké sály ministerstva pro vědu v centru Vídně a soutěž musela být dvoudenní. Česká účast byla velmi dobrá, dařilo se ve volné jízdě a opět v kategorii Puck Collect.

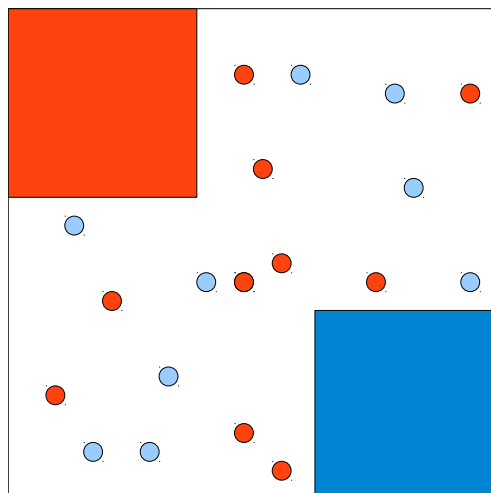
Letošního ročníku jsem se zúčastnil spolu s týmem žáků ze Základní školy 8. května 63 v Šumperku. Náš robot se jmenoval Bobik a byl nasazen v kategorii Puck Collect. Je to soutěž, kde v každém zápase dva roboti sbírají puky vylosované barvy a dopravují je do své základny. Celkem se zúčastnilo 15 týmů z celé Evropy, které bojovaly v sobotu ve dvou skupinách. V naší skupině se nám velice dařilo a po 7 utkáních skončil Bobik na 1. místě. V neděli se 4 nejlepší týmy z každé skupiny utkaly ve finále. Po těžkém boji v nelehké konkurenci se náš robot probojoval na čtvrtou pozici.

Celkové pořadí ve finále:

- 1 - J2MP - Vysoké učení technické Bratislava - Slovensko
- 2 - R-BOT Sarmatic - Střední škola technická Gdyně - Polsko
- 3 - Speedy - Hornorakouská univerzita - Rakousko
- 4 - Bobik - Základní škola Šumperk, 8. května - Česká republika
- 5 - FemmeFatale - Hornorakouská univerzita - Rakousko
- 6 - R-BOT Polonic - Střední škola technická Gdyně - Polsko
- 7 - Umornik - Karlova univerzita - Česká republika
- 8 - GATTACA11 - Gattaca team - Německo

5.4. Fieldrobot

Field Robot Event je soutěž autonomních robotů, která se odehrává ve venkovním prostředí. Motivace pochází ze zemědělství a původ je spíše akademický. Úkolem robotů je rychle se navigovat v kukuřičném poli, aniž by rostliny poničili. Soutěž je mezinárodní a putuje po Evropě — poslední kolo se konalo 11. - 13.



Obrázek 4. Hrací plán kategorie Puck collect soutěže Robot Challenge

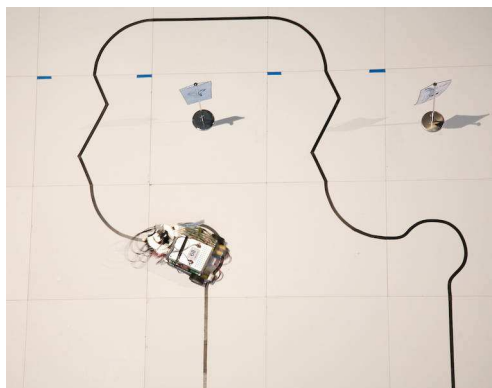
června 2010 v Braunschweig/Německo. Tato soutěž patří již do kategorie náročnějších a zúčastňují se jich zejména týmy studentů technických oborů vysokých škol.

5.5. Robotour

Robotour — robotika.cz outdoor delivery challenge, je původem česká soutěž autonomních robotů pohybujících se po parkových cestičkách. Roboti vezoucí náklad (pětilitrový soudek) jsou bodově zvýhodněni. Soutěž se koná každoročně na podzim. První ročník byl v roce 2006 v Praze v parku Stromovka.

Pátý ročník soutěže autonomních outdoor robotů se bude konat 18. září 2010 na Slovensku v jednom ze tří předvybraných parků v Bratislavě. Na rozdíl od předešlých ročníků roboti dostanou pouze mapu a souřadnice cíle. Roboti nebudou přesně znát svoji startovní polohu a interakce s operátorem se omezí na zadání cíle. Robot úspěšně řešící tuto úlohu by měl být schopen demonstrovat své schopnosti v jakémkoli parku s odpovídající mapou.

Tuto soutěž zde uvádím pouze pro zajímavost, neboť je úplně z jiné kategorie náročnosti. Jak z hlediska programování, tak i konstrukce robota. Je určena spíše pro studenty vysokých škol. Co se týká hardwarového vybavení, tak se zde používají roboti vlastní konstrukce vybavené senzory typu kompas, GPS a kamery a řízené obvykle notebookem. Pro začátečníky je spíše určena soutěž pořádaná Radioklubem Písek od roku 2009 s názvem Robotem rovně, kde se soutěží, který robot dojede po rovné cestě nejdále.



Obrázek 5. Kategorie Line follower soutěže Robot Challenge

6. Způsob realizace výukového materiálu

6.1. Zvolené technologie

Pro výukový materiál jsem zvolil řešení pomocí statických webových stránek s jednotným vzhledem řízeným kaskádovými styly. Důvodem byla zejména nezávislost na technologii webového serveru a možnost vytvoření offline verze na CD nebo šířené přes USB flash disk. Toto řešení trochu omezuje interaktivní možnosti výukového materiálu, jako jsou rejstříky, vyhledávání a automatické vyhodnocování testů. Vzhledem k povaze tohoto materiálu a jeho členění však tyto funkce nebyly potřebné. Zaměřil jsem se spíše na stránku multimediální. Většina stránek je tvořena fotogaleriemi z přípravy materiálu a stavby robota, nebo screencasty z programování a video ukázky z praktického provedení.

Pro fotogalerie jsem použil technologii Ajax založenou na Javascriptu, konkrétně engine VisualLightBox, který je distribuován jako software zdarma pro nekomerční využití. Pro přehrávání video souborů přímo na webové stránce jsem zvolil dnes nejpoužívanější technologii Adobe Flash plugin s přehrávačem JW player. Ten je šířen s licencí Creative Commons, která dovoluje použití, modifikaci a libovolnou následnou distribuci, ale opět pouze pro nekomerční účely. Pro přehrávání zvukových souborů jsem použil 1pixelout audio přehrávač, který má licenci GNU General Public Licence a jeho použití, modifikace a následná distribuce jsou tedy neomezené.

Ke správnému zobrazení by měl tedy stačit libovolný webový prohlížeč s podporou Javascriptu a instalovaným Adobe flash pluginem verze alespoň 9.0.

6.2. Příprava multimediálních souborů

Všechny podklady a multimediální soubory byly zpracovány v aplikacích s licencí GNU General Public Licence v operačním systému Linux. Fotografie poří-

zené digitálním fotoaparátem byly upravovány v aplikaci Gimp ve formátu JPG. Video z ukázky činnosti projektovaného zařízení jsou pořízena digitální videokamerou na kazety mini DV. Následně staženy do počítače přes rozhraní Firewire (IEEE 1394) pomocí utility dvgrab. Střih byl proveden v aplikaci Kino a výsledné video exportováno pomocí utility ffmpeg do formátu flv (flash video kontejner) s kodekem obrazu Sorenson Video. Soubory s výslovností anglických slov byly vygenerovány softwarovým řečovým syntetizérem na stránkách <http://www.text2speech.org/> a uloženy ve formátu mp3.

6.3. Členění materiálu

Materiál je rozdělen do čtyř částí: Základy, Projekty 1, Projekty 2 a English corner. V sekci Základy jsou popsány hardwarové části stavby, senzory, motory a další speciální doplňky, které je třeba připravit. V softwarové části jsou video-návody pro použití ikonografického programování NXT-G a postup instalace a tvorby programů v LeJOS API v operačním systému Linux.

6.4. Mezipředmětové vztahy

Při práci na projektech využívají žáci znalostí nejen z oblasti Informatiky, ale i z dalších předmětů. Zejména Fyziky (měření veličin, mechanika, elektrické obvody), Matematiky (měření úhlů, obvod kružnice, převody jednotek) a Anglického jazyka.

6.5. Návrh a výroba vlastních hardwarových prvků

V druhé sadě projektů jsou navrženy hardwarové prvky vlastní výroby. Je potřeba je vyrobit, nebo jejich výrobu zajistit, nebo je nahradit dostupnými profesionálními komponenty. Jedná se o akční člen, přes který může robot ovládat elektrospotřebiče na síťové napětí, senzor pro měření vlhkosti půdy založený na principu měření elektrického odporu přes analogový vstup. Dále pak laserová závora tvořená laserovým ukazovátkem ve spojení se senzorem osvětlení a úprava NXT kabelů, aby bylo možné připojit současně čtyři analogové a čtyři digitální senzory.

7. Popis vlastního materiálu

7.1. Základy

V této sekci se žáci seznámí s tím, jak začít používat stavebnici a programovací prostředí. Pokud už mají s robotem nějaké zkušenosti, mohou ji vynechat a přejít rovnou k projektům. Užitečná je také kapitola o instalaci LeJOS v operačním systému Linux a popis jeho programování.

7.1.1. Kostka NXT

Kostka NXT je základní součástka stavebnice. V této sekci se čtenář dozví, jaké má vstupy a výstupy, jak se připojí k počítači a jakým způsobem je možné ji napájet elektrickým proudem.

7.1.2. Motory

Motory se připojují do výstupních portů A, B nebo C. Jejich programování je vysvětleno přímo v projektech. Podrobně se tomu věnuje první projekt Plnou parou vpřed ! V projektu Měření místnosti je využit i jejich zabudovaný tachometr k měření ujeté vzdálenosti. Vyzkoušet mohou žáci také ozubené převody, které znají z Fyziky.

7.1.3. Senzory

Žáci se seznámí se všemi čtyřmi senzory základní stavebnice. Je to senzor dotyku, osvětlení, zvuku a senzor ultrazvukového odrazu. Jsou zde ukázky použití, způsob propojení s kostkou NXT a ukázkové programy v NXT-G i Javě. Tyto programy jsou tvořeny nekonečným cyklem, kde se čte hodnota naměřená senzorem a zobrazuje se na displeji kostky NXT. U senzoru dotyku, který poskytuje pouze logickou hodnotu, je využit blok switch nebo konstrukce if v Javě.

7.1.4. Speciální doplňky

Tato sekce poskytuje návod pro výrobu dalších hardwarových prvků, které jsou potřebné v projektech nebo mohou rozšířit možnosti stavebnice při tvorbě vlastních návrhů.

Akční člen

Akční člen slouží k bezpečnému zapínání a vypínání elektrospotřebičů na nebezpečné síťové napětí. Přes akční člen může robot ovládat například lampu, ponorné topení pro akvária nebo fontánové čerpadlo. Využijeme k tomu běžně dostupnou zásuvku na dálkové ovládání bez stmívače. Dálkové ovládání spíná robot motorem s dvojitým klikovým mechanismem. Aby poznal, že je akční člen

zapnutý nebo vypnutý, poskytuje mu zpětnou vazbu lampa přes světelný senzor. Potřebný software k ovládání tohoto mechanismu naleznou studenti přímo v projektech, které akční člen využívají.

Čidlo vlhkosti

Analogové vstupy obsažené v portech 1 až 4 dokáží měřit elektrický odpor až do $50k\Omega$. Toho se dá využít při různých pokusech, například tzv. detektor lží, kdy měříme elektrický odpor lidské kůže, nebo k měření vlhkosti půdy. Využijeme k tomu převodník z kabelu NXT na RCX a koncovku ustříhneme. Místo ní připájíme měděné vodiče. Software je jednoduchý, využijeme bloku Touch sensor, který má výstup datového vodiče s tzv. raw value (surovou hodnotu), která odpovídá naměřenému odporu.

Laserová závora

V průmyslových aplikacích se často používá tzv. laserová závora pro detekci průchodu osob nebo pohybu zboží. Skládá se ze dvou částí. První je vysílač laserového paprsku. Ten si můžeme vytvořit jednoduše z laserového ukazovátka. Druhou část tvoří přijímač, který vytvoříme pomocí senzoru světla. Důležité je přesné zamíření optického senzoru a nastavení optimální rozhodovací úrovně v programu. Jakmile se paprsek přeruší průchodem osoby nebo průjezdem předmětu, úroveň osvětlení na senzoru klesne a program zareaguje. Tuto závoru využijeme v projektu Otevírání brány.

Rozšíření na 8 vstupů

Nevýhodou kostky NXT je malý počet vstupů. Pro složitější projekty budeme potřebovat toto omezení nějak obejít. Na rozšíření počtu motorů lze dokoupit rozbočovače a získat tak další motorové porty. U vstupů je situace složitější, rozbočovač podporují pouze digitální senzory, které umí změnit svou adresu. Můžeme ale využít vlastnosti, že všechny 4 vstupy se skládají z analogové a digitální části. Takže po vhodné úpravě kabelů můžeme na každý port připojit jeden analogový a jeden digitální senzor. Například tlačítko a Ultrazvuk. Jeden kabel přestříhneme, druhý opatrně nařízneme nožem a připájíme ho pouze na vodiče 1 a 2 (analogová vstupní linka).

7.1.5. Software NXT-G

Seznámení s ikonografickým programovacím prostředím je provedeno pomocí devíti video sekvencí popisujících použití jednotlivých částí aplikace a programovacích bloků.

- Nový program - seznamuje nás, jak vytvořit nový program a jak jej uložit na pevný disk.
- Přehled bloků - uvádí přehled všech dostupných bloků použitelných pro programování stavebnice

- Komunikace - znázorňuje, jak posílat programy do kostky NXT a jak je spouštět.
- Nekonečný cyklus - ukazuje, jak vytvořit část programu, která se neustále opakuje.
- Cyklus s podmínkou - cyklus, který se opakuje až do splnění předem dané podmínky, zejména změny hodnoty libovolného senzoru.
- Cyklus řízený hodnotou - programová smyčka opakující se po dobu přesně daného času, nebo smyčka s přesným počtem opakování.
- Nepodmíněné větvení - program se může rozdělit do více paralelních větví, které se vykonávají souběžně.
- Větvení s podmínkou - příkazy se vykonají jen v případě, že je splněna daná podmínka. Opět může jít o hodnotu senzoru.
- Vícenásobné větvení - Na základě příchozí hodnoty se provede jedna z mnoha možností, pro tuto volbu se musí u blok switch vypnout plochý pohled a přejít tak k pohledu se záložkami.

7.1.6. Software Java LeJOS

Sekce popisuje, jak začít s programováním robota v jazyce Java za pomoci speciálního firmware LeJOS. Je rozdělena do několika částí.

Instalace

Praktický návod, jak v operačním systému nainstalovat potřebné softwarové prostředky a nastavit systém, aby komunikoval s kostkou NXT přes USB port i Bluetooth rozhraní. Také popisuje způsob nahrávání firmware LeJOS do paměti kostky NXT. Vybrána byla distribuce Ubuntu 10.04, která je v současné době nejpoužívanější, ale s drobnou obměnou je možné návod použít i v ostatních distribucích.

První program

Programování začíná klasickým programem Ahoj světe ! Navíc je zde popsána cesta, jak se dostat k ukázkovým programům k využití různých tříd celého API a naučit se tak co nejvíce z praktických příkladů. Stejný způsob učení je použit i v samotných projektech.

Proměnné

Přehled a ukázky použití proměnných v programech psaných v jazyce Java. Různé možnosti proměnných pro číselné hodnoty v celých číslech i číslech s plovoucí desetinou čárkou, znaky, řetězce a logické hodnoty.

Větvení programu

Programová konstrukce if-else a různé způsoby tvoření podmínek. Spojování podmínek logickými spojkami a vícenásobné větvení pomocí konstrukce switch s praktickou ukázkou částí programu.

Cykly

Vysvětlení použití opakování sekvence příkazů v cyklu while, poznámka o způsobu vytvoření nekonečného cyklu, který se hodí v aplikacích pro regulaci veličin a speciální cyklus for.

Motory

Popis řízení motorů začíná prostým ovládáním jednotlivých výstupních portů A,B a C. U motorů můžeme nastavovat rychlost, výkon, rozbíhat je na obě strany, nebo nastavit přesný úhel otočení. API obsahuje navíc speciální třídy pro precizní řízení jízdy celého robota. Popsána je třída Pilot, která po nastavení parametrů nápravy robota v konstruktoru dokáže vykonávat metody pro jízdu na přesně danou vzdálenost nebo zatáčení celého robota na místě o přesně daný úhel.

Senzory

Zde je jen zmínka o tom, kde najít potřebné informace k programování speciálních senzorů. Standardní senzory stavebnice jsou podrobně popsány v sekci Hardware včetně zapojení a ukázek programů.

7.2. Language corner

V informatice a robotice se používá spousta anglických slov. Nejen dokumentace, která bývá dostupná pouze v angličtině, ale také příkazy programovacích jazyků a rozhraní aplikací se většinou do češtiny nepřekládá. Stejně tak i prostředí Lego Mindstorm je dostupné pouze v několika světových jazycích. Proto jsem se rozhodl zařadit do výukového materiálu i tuto sekci, kde si žáci osvojí potřebnou slovní zásobu. Je zde zpracováno asi 250 anglických slov, které se vyskytují při práci s robotem. Jsou rozdělena do deseti lekcí. V každé lekci najdou žáci slovníček včetně výslovnosti a několik cvičení na daná slova. Aby mohli cvičení vypracovávat, je možné stáhnout pracovní sešit ve formátu pdf.

7.3. Projekty 1

Tyto projekty slouží k prvotnímu seznámení s konstrukcí robota a jeho programováním. Vyžadují k provedení pouze stavebnici Lego Mindstorm NXT.

7.3.1. Projekt 1.1 - Plnou parou vpřed !

Základní projekt, kde se žáci naučí sestavit robota podle návodu a připojit k němu motory, senzory dotyku a ultrazvukové detekce překážek. Tak jako další projekty je podrobně rozpracován do 11 kroků. V úvodu je popsán problém,

který budeme řešit a motivační video zobrazuje robotický vysavač iRobot, který je běžně v prodeji na našem trhu.

V prvním programu popojede robot o určitou přesně danou vzdálenost. Potom programujeme robota, aby objel překážku, o které předem víme. Jedná se o tzv. jízdu podle mapy. Tato strategie se velice často používá v robotických soutěžích s předem daným herním plánem bez náhodných prvků. V dalším kroku se robot snaží překážku detekovat pomocí senzoru dotyku. Jak žáci zjistí pokusem, není tento způsob příliš efektivní, protože se musí vracet, aby mohl překážku objet. Proto vyměníme později senzor dotyku za senzor ultrazvukového odrazu, který umožní detekci překážky s předstihem. Protože může být překážek několik za sebou uplatní se zde i nekonečný cyklus.

Na závěr tohoto projektu budou žáci vyzváni, aby celou konstrukci i program zdokonalili kombinací více senzorů pro detekci délky překážky. Využijí tak dovedností, které během projektu získali k vlastní samostatné práci.

Použité aktivní prvky stavebnice:

- dva motory
- senzor dotyku
- senzor ultrazvukového odrazu

Použité programové konstrukce:

- ovládání motorů
- čtení hodnoty senzorů
- nekonečný cyklus

Další potřebné dovednosti:

- porovnávání čísel
- měření vzdálenosti
- číslo Π a výpočet obvodu kružnice
- měření úhlů ve stupních

7.3.2. Projekt 1.2 - Ovládej tlesknutím

V tomto projektu se pracuje se senzorem zvuku. Jako motivace je použito video z japonské výstavy, kde robot poslouchá příkazy člověka. Tak složitý cíl si náš projekt neklade, naučíme robota rozeznávat pět povelů ve formě tlesknutí. Využijeme měření úrovně hluku v okolí. V prvním kroku opět sestavíme jezdícího robota, jehož směr se řídí dvěma nezávislými motory. Mezi potřebnými

součástkami jsou zejména dva zvukové senzory, ale pro základní zadání postačí i jeden. Nejprve se naučíme měřit intenzitu zvuku a zobrazovat ji na display. V dalším kroku navážeme měřenou úroveň na zapínání a vypínání lampy. V rozšiřujícím úkolu mají žáci vytvořit rampu se třemi barevnými lampami reagujícími na různou hladinu hluku.

Po zkušební jízdě robota reagujícího na tlesknutí jízdou a následným zastavením vyzkoušíme další zajímavý experiment. Připojíme dva zvukové senzory, tak jako se používají mikrofony při nahrávání stereo. Naprogramujeme robota, aby zatačel směrem, ze kterého vychází silnější zvuk. Výsledkem je robot, který jede směrem, kde se nachází zdroj zvuku.

Další část projektu je již trochu složitější. Naučíme robota rozeznávat počet tlesknutí. Ve zkušebním programu robot bliká při každém tlesknutí a po skončení potlesku zazní tolik zvukových signálů, kolik tlesknutí slyšel. Nakonec žáci sami sestaví výsledný program, kde robot podle počtu tlesknutí vykonává různé příkazy. Jedno tlesknutí vyvolá jízdu dopředu o dvě otáčky motoru, dvě tlesknutí jízdu dozadu, tři tlesknutí otočku doleva a čtyři tlesknutí otočku doprava.

V závěru je opět zhodnocení celého projektu a rozšiřující úkol, tentokrát zaměřený na zpřesnění zatačení, tak aby robot skutečně pomocí povelů dojel na určené místo.

Použité aktivní prvky stavebnice:

- 2 motory NXT
- 2 senzory zvuku
- 3 světelné lampy RCX s kabely

Použité programové konstrukce:

- ovládání motorů
- ovládání světel
- nekonečný cyklus
- čtení hodnot ze sensorů
- větvení programu (podmínka IF nebo blok Switch)
- časování pomocí bloku Timer
- použití proměnné pro uchování hodnoty
- použití matematického bloku pro zvýšení počtu tlesknutí v proměnné
- tvorba vlastního bloku

Další potřebné dovednosti:

- sčítání a porovnávání čísel
- sestavení jezdícího robota
- pochopení použití proměnné
- měření úhlů a času

7.3.3. Projekt 1.3 - Měření místnosti

Další projekt opět využívá jezdícího robota, tentokrát k přesnému změření rozměrů místnosti. Motivací je video, které si žáci sami natočili. Znázorňuje, jak je pracné změřit velkou místnost pásmem a spočítat podlahovou plochu. Robot tuto práci vykoná hravě. Jen ho musíme sestavit a naprogramovat.

Sestavíme jezdícího robota, který bude mít vepředu i vzadu senzory dotyku a zvukový senzor. Naprogramujeme ho tak, aby jel rovně tak dlouho, dokud nenarazí na zeď. Až se bude vracet zpět, tak můžeme ve chvíli, kdy bude kolem něj volný prostor tlesknout a robot se otočí o 90 stupňů. Pak pojedou opět až ke zdi.

Až bude robot spolehlivě jezdit, můžeme přejít k dalšímu kroku, a to měření vzdálenosti pomocí tachometru. Robot dokáže měřit otáčky motoru ve stupních. My ale potřebujeme měřit ujetou vzdálenost. Zde využijeme znalosti z matematiky a spočítáme si obvod kola. Potom pomocí přímé úměrnosti vypočteme, kolik centimetrů připadá na jeden stupeň. Robot musí ve vhodnou chvíli vynulovat počítadlo. V NXT-G je k tomu speciální blok na paletě vstupních bloků. Stejný blok se používá i pro čtení stavu tachometru. V Javě použijeme metodu `Motor.B.resetTachoCount()` a pro čtení metodu `Motor.B.getTachoCount()`.

Další problém, na který narazíme, je potřeba uchovat naměřené hodnoty obou směrů v proměnných. Uchované hodnoty délky a šířky se nakonec využijí na výpočet podlahové plochy. Tu by měli žáci přepočítat na m^2 a zobrazit na display kostky NXT.

V posledním kroku je ukázková jízda se zobrazením naměřených a vyčtených hodnot. V závěru se po vyhodnocení projektu nabízí možnost další aplikace měření pomocí tachometru motoru. Jde o kontrolu maximálního objemu balíků určených pro přepravu.

Použité aktivní prvky stavebnice:

- 2 motory NXT
- 1 senzor zvuku
- 2 senzory dotyku

Použité programové konstrukce:

- ovládání motorů

- reakce na dotykový senzor
- čtení stavu tachometru v motoru
- použití proměnných pro uchování hodnoty
- použití matematického bloku

Další potřebné dovednosti:

- násobení čísel
- měření úhlů
- výpočet obvodu kružnice
- výpočet obsahu obdélníka
- výpočet objemu kvádra
- sestavení jezdícího robota
- pochopení použití proměnné

7.3.4. Projekt 1.4 - Turnikety v muzeu

V tomto projektu se pokusíme vytvořit automatický systém turniketů pro příchod a odchod z venkovních výstavních prostor. Naše zařízení bude zjišťovat počet návštěvníků v prostorách muzejní expozice. Hlídač pak bude před zavírací dobou vědět, kolik lidí se ještě v muzeu pohybuje.

K tomu budeme potřebovat vytvořit model dané situace a vlastní turnikety. Na obrázcích v projektu je model, který vytvořili žáci sami při provádění tohoto projektu. Každý turniket je doplněný dotykovým senzorem, aby bylo možné projít. Druhý turniket se točí v opačném směru a umožňuje návštěvníkům z výstaviště odejít.

Samotný program zajišťuje otáčení turniketů správným směrem po stisku příslušného senzoru a v proměnné si uchovává aktuální počet návštěvníků. Když návštěvník vstoupí, počet se zvýší o jedničku, a naopak když odejde, počet se o jedničku sníží. Aktuální počet se zobrazuje na display kostky NXT, takže má personál neustálý přehled o počtu návštěvníků.

Jak je vidět na výsledném videozáznamu, po odchodu posledního návštěvníka je stav počítadla 0. Po vyhodnocení projektu mohou žáci program vylepšit tak, že bude na druhém řádku zobrazovat celkový počet návštěvníků, kteří daný den muzeum navštívili, případně po stanovení jednotné ceny vstupného i celkovou vybranou sumu.

Použité aktivní prvky stavebnice:

- 2 motory NXT
- 2 senzory doteku

Použité programové konstrukce:

- ovládání motorů
- reakce na dotykový senzor
- použití proměnných pro uchování hodnoty
- použití matematického bloku
-

Další potřebné dovednosti:

- sestavení modelu podle návrhu
- sčítání, odčítání a násobení čísel
- výpočet celkové ceny

7.3.5. Projekt 1.5 - Střežení objektu

Poslední projekt z první sady je z programátorského hlediska nejnáročnější. Pokusíme se vytvořit model elektronického zabezpečovacího zařízení, stejného jako bývá v obchodech, firmách, na úřadech nebo rodinných domech. Sestavení modelu je velmi snadné, stačí nám k tomu jeden ultrazvukový senzor a jeden motor. Pro zadávání kódu použijeme motor a jeho schopnost odečítat otáčky, protože připojení numerické klávesnice ke kostce NXT by bylo příliš náročné.

Program bude pracovat následovně. Pokud někdo naruší hlídanou zónu, začne znít výstražný signál a odpočítávání času. Uživatel musí do uplynutí časového limitu zadat čtyřmístný kód. Čísla se zadávají otáčením motoru. Jakmile robot zaznamená pootočení motoru větší než 30 stupňů, zvýší hodnotu čísla. Zde musíme ošetřit situaci, že hodnota je vyšší než 9, pak by se měla nastavit opět 0. Můžeme přidat i detekci otáčení v záporném smyslu a hodnotu čísla odečítat. Po stisknutí tlačítka ENTER přejde robot na zadávání další pozice.

Po zadání čtyř číslic dojde k vyhodnocení, jestli jsou všechny vloženy správně. To se provede následovně. Na začátku nadefinujeme proměnnou spravne typu boolean a nastavíme ji na true (pravda). Potom postupně testujeme jednotlivá čísla, jestli se rovnají přednastavené hodnotě. Když některé nesouhlasí, nastavíme hodnotu spravne na false (nepravda). Pokud alespoň jedno s čísel nesouhlasilo, zůstane nakonec v proměnné spravne logická hodnota false.

Zadávání čísla se může opakovat, dokud neuplyne časový limit. Tady se dostáváme do zajímavé situace, ve které potřebujeme ukončení cyklu řídit dvěma

hodnotami současně. Použijeme tedy cyklus v režimu Value a obě logické hodnoty porovnáme v bloku logic.

Celá situace v cyklu bude vypadat následovně, blok Timer ověří, jestli nevypršel čas. Náš vlastní blok Zadavani vrací, jestli byl kód zadán správně. Obě hodnoty se přivedou do logického bloku v režimu AND. Výsledná logická hodnota se datovým vodičem přivede do ukončovací podmínky cyklu.

Stejnou konstrukci můžeme použít i v případě, že chceme ukončit cyklus po detekci změny stavu dvou různých senzorů, zde bychom použili logický blok v režimu OR.

Na konci projektu program vyzkoušíme a doladíme časy a detekční vzdálenosti ultrazvukového senzoru. Žáci mohou projekt vylepšit přidáním dalších senzorů na detekci narušení objektu. Například senzory doteku v obráceném režimu nebo laserovou závoru sestavenou s laserového ukazovátka a světelného senzoru. Senzory musí být čteny v cyklu a jejich logický výstup přiveden do logických bloku OR, výslednou hodnotou se pak řídí ukončovací podmínka cyklu.

Použité aktivní prvky stavebnice:

- 1 motor NXT
- 1 senzor ultrazvukového odrazu
- 2 senzory doteku

Použité programové konstrukce:

- čtení tachometru motoru
- čtení hodnoty senzorů
- použití proměnných pro uchování hodnoty
- použití matematického bloku
- použití logického bloku
- použití vlastního bloku s výstupním datovým vodičem

Další potřebné dovednosti:

- logická proměnná
- logické operátory AND a OR
- měření času
- sčítání a odčítání čísel
- porovnávání čísel
- ovládání zvukového výstupu

7.4. Projekty 2

Druhá sada projektů vyžaduje k provedení nejen stavebnici Lego Mindstorm NXT, ale také další součásti. Některé se dají snadno vyrobit, jiné je třeba zakoupit. Na začátku každého projektu je uveden seznam potřebných dílů.

7.4.1. Projekt 2.1 - Přechod pro chodce

Tento projekt má za cíl seznámit žáky s problematikou řízení světelných křižovatek. Jelikož kostka NXT má jen tři výstupy, je zredukovaný pouze na přechod pro chodce, kde s použitím drobné úpravy zapojení lamp je tento počet výstupů dostatečný. Nejde tedy o dokonalé napodobení reálného systému, ale o vytvoření funkčního modelu, na kterém je jasný princip celého zařízení. Na začátku projektu je jako motivace použito video zobrazující přechod pro chodce před hlavním nádražím v Přerově, kde mají speciální display, který zobrazuje chodcům zbývající čas do rozsvícení zelené na semaforu. Toto zobrazení je pak i součástí projektu.

V průběhu projektu si žáci připraví potřebný materiál pro vytvoření modelu. Stačí použít jen papír a barvy, nebo mohou využít fantazii a sestavit věrohodný model včetně domů a zeleně. V ukázce je jednoduchý model s černé tapety a kostek stavebnice Lego. V dalším kroku je třeba vyrobit semaforey. Ty je možno realizovat pomocí stavebnice Lego a lamp dodávaných spolu s robotem, nebo vytvořit vlastní s využitím LED diod opatřených rezistory a napojených přes redukci NXT-RCX. Potom je třeba Semaforey propojit, zde nastane problém, že kostka NXT má pouze tři výstupy, a musíme lampy vhodně propojit pomocí diod. Takže se žáci seznámí i s funkcí této elektronické součástky.

Po sestavení modelu a přidání tlačítka pro přijetí požadavku chodců následuje testovací program na ovládání semaforů. V prostředí NXT-G je přímo blok Lamp s akcemi On a Off, takže ovládání světel je velmi snadné. V Javě musíme nadefinovat třídu Lamp s metodami zapni a vypni sami. Využijeme třídu MotorPort a její metodu controlMotor, která umožňuje nízkourovňové řízení výstupních portů.

Vlastní program pro řízení přechodu je tvořen nekonečným cyklem obsahujícím nastavení základního stavu a čekání na stisknutí tlačítka u přechodu. Pak probíhá postupné přepínání semaforů, aby mohli chodci přejít. V dalších krocích se program postupně vylepšuje o zobrazení zbývajícího času chodců a zvukové signalizace pro zrakově postižené. Na závěr mají žáci program upravit, aby reagoval na dálkové ovládání z vozidel záchranné služby. To mohou provést přes druhý dotykový senzor nebo použít rozhraní Bluetooth.

Použité aktivní prvky stavebnice:

- 5 lamp s barevnými kryty
- 2 senzory doteku
- 9 redukcí NXT-RXC

Použité programové konstrukce:

- ovládání lamp (v Javě vlastní třída)
- nekonečný cyklus
- posloupnost příkazů s časovou prodlevou
- větvení programu (podmínka IF nebo blok Switch)
- paralelní vlákno (zvuková signalizace)

Další potřebné dovednosti:

- letování el. obvodů
- funkce polovodičové diody
- výroba modelu silnice a přechodu

7.4.2. Projekt 2.2 - Ovládání brány

Tento projekt opět přináší možnost seznámit se s prostředky automatizace používanými v praxi. Kromě běžně dostupného laserového ukazovátka nevyžaduje žádné doplňující hardwarové prvky, a tak ho celý mohou provádět žáci samostatně. Na úvod je ukázka automatické brány ve skutečném provozu a její profesionální montáž odbornou firmou. Potom se provádí konstrukce modelu z kostek stavebnice Lego. Pohon křídlové brány je navržen místo obvyklé šroubovice výklopný, který se v praxi používá méně často, ale jeho sestavení je jednodušší. Přesto by bylo možné použít i jiný typ pohonu, například sestavit celou bránu jako posuvnou se zubovým převodem.

Dalším problémem, který projekt řeší, je vlastní ovládání. Nejprve se použijí jen senzory ultrazvukového odrazu (detekce příjezdějícího vozidla) a laserová světelná závora tvořená ukazovátkem a světelným senzorem (detekce vozidla opouštějícího bránu). Na závěr projektu se ovládání upraví pro dálkové řízení. Vyčijeme druhou kostku NXT a komunikaci přes Bluetooth rozhraní. Navážeme spojení a odesíláme textové příkazy. Existuje i možnost ovládat kostku z mobilního telefonu s vestavěným Java virtual machine, touto problematikou se ale projekt nezabývá.

V závěru projektu je nastíněna situace, kdy dochází k zavírání brány, ve které zůstalo vozidlo stát. Žáci mají za úkol provést úpravu zařízení, aby k tomuto problému nedocházelo.

Použité aktivní prvky stavebnice:

- 2 motory NXT
- senzor ultrazvukového odrazu a světelný senzor

- druhá kostka NXT pro řízení přes Bluetooth rozhraní

Použité programové konstrukce:

- ovládání motorů
- nekonečný cyklus
- čtení hodnot ze senzorů
- větvení programu (podmínka IF nebo blok Switch)
- komunikace přes Bluetooth

Další potřebné dovednosti:

- měření úhlů
- mechanické sestavení brány a pohonu
- sestavení laserové závory

7.4.3. Projekt 2.3 - Udržování teploty čaje

Tento projekt se zabývá problematikou dvoustavové regulace. S tou se v praxi setkáváme velice často, zejména u tepelných spotřebičů všeho druhu. Pro regulaci jsem zvolil konkrétní použití u ohřevu čaje na žádanou teplotu, aby bylo využití tohoto projektu ihned zřejmé.

Základním problémem je měření teploty. Ke stavebnici NXT je možné dokoupit dva druhy teplotních senzorů. Starší analogový typ připojovaný přes redukci NXT-RCX nebo nový digitální senzor teploty NXT. Analogový senzor má rozsah od -20 do $+50$ °C s rozlišením 0.2 °C, digitální od -40 do 125 °C s přesností řádově vyšší. Také rychlost měření je u nového typu větší. Pro účely tohoto experimentu je ale analogová verze dostačující. V programovém prostředí NXT-G však pro něj není senzorový blok. Tento nedostatek jsem obešel použitím bloku Touch sensor, který má výstup datovým vodičem tzv. raw value, což je přímo naměřená hodnota analogového vstupu. Teplota t se z hodnoty senzoru r vypočítá jednoduchým vztahem:

$$t = \frac{765 - r}{8.4}$$

Druhý problém, který se zde řeší, je ovládání akčního členu pro řízení silového obvodu topení. Kostka NXT má ale výstup pouze pro řízení motorů nebo světel. Tento problém jsem vyřešil vytvořením akčního členu, který se skládá z motoru NXT a ojnice, přes niž motor zapíná a vypíná tlačítko v provedení do vlhka. Aby byla zajištěna zpětná vazba, je přidána ještě stolní lampa a světelný senzor. Toto řešení má jednu nevýhodu, musí se vyrobit silový obvod pro napájení 230 V

složený z tlačítka, spojovací krabice a zásuvky. Tuto práci musí provést elektro-mechanik s patřičnou odbornou kvalifikací. Proto jsem navrhl ještě jedno řešení sestavené z dálkově ovládané zásuvky bateriovým ovladačem, která je na trhu běžně dostupná a zde motor pomocí ojnice spíná střídavě zapínací a vypínací tlačítko. Výhodou je, že se nemusí žádné speciální zařízení vyrábět a žáci je mohou samostatně používat jako běžné elektrospotřebiče v domácnosti. Co se týká řídicího programu, je pro oba typy akčních členů stejný. Otáčí motorem NXT až do té doby, než přijde odezva ze světelného senzoru, která detekuje zapnutí nebo vypnutí spínače.

Na silový obvod je pak napojena stolní lampa poskytující zpětnou vazbu a vlastní topení pro akvária, které má nízký příkon (25 W) a integrovaný termostat, takže je jeho použití při experimentu bezpečné. Celý experiment se tak může uskutečnit v libovolné dostatečně velké nádobě.

Samotná regulace spočívá v pravidelném měření teploty a její porovnávání s žádanou hodnotou uloženou v proměnné. Na základě tohoto porovnání se spíná akční člen. Diference mezi zapnutím a vypnutím se neřeší, neboť se teplota měří v delších intervalech a zakmitání způsobené nepřesností čidla je minimální.

Po zprovoznění celého zařízení se přidá logování teploty do souboru. Žáci mají za úkol nechat regulaci běžet po delší dobu a potom uložené hodnoty znázornit grafem v tabulkovém kalkulátoru. Potom se porovná skutečný průběh regulace s průběhem ideálním.

V posledním kroku se program rozšiřuje o paralelní vlákno, které umožňuje uživateli nastavovat žádanou teplotu pomocí tlačítek na kostce NXT. V závěru jsou žáci vyzváni, aby celý projekt zhodnotili. Námětem na možné rozšíření je úprava, kde program zvukovou signalizací upozorní uživatele, že čaj má žádanou teplotu.

Použité aktivní prvky stavebnice:

- motor NXT
- světelný senzor a senzor teploty
- akční člen (tlačítko nebo dálkově ovládaná zásuvka)
- topení pro akvária 25 W
- stolní lampa

Použité programové konstrukce:

- ovládání motorů
- nekonečný cyklus
- čtení hodnot ze senzorů a tlačítek

- větvení programu (podmínka IF nebo blok Switch)
- paralelní vlákno programu

Další potřebné dovednosti:

- měření teploty ve stupních Celsia
- sestavení akčního členu řízeného motorem s ojnící
- tvorba grafů z externích dat v tabulkovém kalkulátoru

7.4.4. Projekt 2.4 - Regulace hladiny v nádrži

Další projekt se zabývá opět regulací, tentokrát je to hladina kapaliny v nádrži. Pro motivaci je uvedeno schéma tříokruhové jaderné elektrárny, kde je udržení správné hladiny vody ve všech okruzích životně důležité pro chod celé elektrárny. Rovněž se zde využívá akční člen, který již máme vyrobený, ale tentokrát na něj napojíme fontánové čerpadlo.

Na začátku projektu je animace činnosti regulační soustavy. Potom jsou uvedeny příklady různých možností snímání hladiny kapaliny v nádrži. V projektu se využívá senzor ultrazvukového odrazu, který měří hladinu s přesností asi 2 cm, což je pro tento experiment dostačující. K sestavení regulační soustavy jsou použity dva plastové kanystry s objemem 10 litrů. Jeden z nich představuje zdroj vody a druhý samotnou nádrž s regulovanou hladinou. Má vypust s kohoutkem pro demonstraci chyby regulace. Horní část s rukojetí je odstraněna, aby bylo možné nasadit senzor hladiny.

Návrh programu je rozdělen do 4 kroků. Nejprve se naprogramuje měření vzdálenosti a její zobrazování na display. Potom se vytvoří samostatné programy na zapínání a vypínání akčního členu. Nakonec se sestaví celý program s předšlých částí. Přitom musíme vyřešit problém, aby nám čerpadlo necyklilo, takže určíme maximální a minimální žádanou hladinu s určitou diferencí. Jak velká bude potřeba mohou žáci vyzkoušet sami experimentem.

Na závěr máme zrychlené video s ukázkou provozu regulační soustavy. Nechybí obvyklé zhodnocení projektu a návrhem na rozšíření je tentokrát doplnění zvukové a světelné signalizace havarijního stavu hladiny v nádrži.

Použité aktivní prvky stavebnice:

- motor NXT
- světlo RCX s kabelem
- světelný senzor a ultrazvukový senzor
- akční člen (tlačítko nebo dálkově ovládaná zásuvka)
- fontánové čerpadlo

- stolní lampa
- 2 kanystry s obsahem 10 litrů
- vhodné hadičky

Použité programové konstrukce:

- ovládání motorů
- ovládání lampy
- nekonečný cyklus
- čtení hodnot ze senzorů
- větvení programu (podmínka IF nebo blok Switch)
- zvuková signalizace

Další potřebné dovednosti:

- měření vzdálenosti
- sestavení akčního členu řízeného motorem s ojnící
- sestavení čerpadla a hadic

7.4.5. Projekt 2.5 - Automatické zavlažování rostliny

Poslední projekt využívá stejné vybavení jako ten předchozí k další praktické aplikaci. Tentokrát bude robot řídit automatické zavlažování rostliny. Motivace je jednoduchá. Každý si vybaví pohled na pokojové rostliny, když na ně pár dní zapomeneme. Nejprve si ukážeme profesionální čidla pro měření vlhkosti půdy a pak se pustíme do výroby vlastního.

Čerpadlo a akční člen sestavíme stejně jako v projektu Regulace hladiny v nádrži. Opět je naprogramujeme a vyzkoušíme. Potom vyrobíme podle návodu čidlo vlhkosti půdy. Je založené na měření elektrického odporu přes deseti bitový AD převodník obsaženém v každém vstupním portu kostky NXT. Pro připojení využijeme kabelovou redukci NXT-RCX, u které odstraníme vstupní konektor. Místo něj napojíme měděné vodiče přiměřené délky. Program pro měření vlhkosti vytvoříme z bloku pro tlačítkový senzor, který má datový vodič s tzv. raw value. Ta odpovídá naměřenému odporu. Necháme si ji zobrazit na display a prozkoumáme, jak se odpor mění v závislosti na vlhkosti půdy.

Připravíme si několik misek se zeminou, každou s jinou intenzitou zálivky. Zjistíme, jaká hodnota odpovídá ideální zálivce a při které hodnotě by měl robot spustit zalévání.

Celý program bude vypadat následovně. Změří se vlhkost půdy a vyhodnotí se, jestli je potřeba zalévat. Pokud ano, spustí se akční člen na krátkou dobu a pak zase čerpadlo zastaví. Potom musí následovat dostatečná časová prodleva, aby se voda vsákla do zeminy. Celý proces se stále opakuje. Aby zařízení fungovalo správně, musíme dobře odhadnout čas běhu čerpadla i prodlevu mezi jednotlivými měřeními. Kontrolu provedeme tak, že zařízení necháme v provozu delší dobu.

Na závěr můžeme projekt vylepšit o kompletní péči o rostlinu pěstovanou v umělém prostředí, tedy spínání osvětlení simulující den a noc, udržování ideální teploty a sypaní hnojiva ve formě granulí ze zásobníku ovládaného motorem.

Použité aktivní prvky stavebnice:

- motor NXT
- světlo RCX s kabelem
- světelný senzor
- akční člen (tlačítko nebo dálkově ovládaná zásuvka)
- fontánové čerpadlo
- stolní lampa
- kanystr s vodou
- vhodné hadičky

Použité programové konstrukce:

- ovládání motorů
- ovládání lampy
- nekonečný cyklus
- čtení hodnot ze senzorů
- čtení surové hodnoty z AD převodníku
- větvení programu (podmínka IF nebo blok Switch)
- časovač

Další potřebné dovednosti:

- sestavení akčního členu řízeného motorem s ojnicí
- sestavení čerpadla a hadic
- výpočet nepřímé úměrnosti

Závěr

Výuka programování může mít různé podoby. Jednou z nich je používání programovacích jazyků určených speciálně pro výuku. Například želví grafika v prostředí Comenius LOGO nebo jeho obdobách. Tyto programovací jazyky ale neprodukují klasické spustitelné aplikace. Motivace žáků pak není příliš vysoká. Na druhé straně se k výuce používají komerční programovací prostředí a jazyky, které tento problém nemají. Musíme se ale při práci s nimi zabývat věcmi, jenž jsou pro výuku nepodstatné. Já jsem se rozhodl přistoupit k výuce jiným způsobem. Učit se principům programování pomocí úspěšné robotické stavebnice Lego Mindstorm NXT.

Podařilo se vytvořit multimediální výukový materiál, který může sloužit učitelům jako inspirace nebo přímo žákům ve výuce. Všechny programy jsou realizovány ve dvou verzích. První v ikonografickém prostředí NXT-G. To je vhodné pro žáky čtvrtých až osmých tříd základních škol. Druhé v objektově orientovaném jazyce Java se speciálním rozhraním LeJOS, vhodném pro žáky devátých tříd a studenty středních a vysokých škol. Výuka je realizována pomocí projektů, jejichž témata řeší problematiku známou z běžného života. Například světelné křižovatky, otevírání brány, regulace teploty, hladiny vody atd. Díky tomu se žáci mohou v problematice snadno orientovat a dokáží sami navrhnout řešení. Projekty jsou většinou rozsáhlejší, a tak vyžadují týmovou práci. Tím rozvíjí u studentů také kompetence sociální, personální a komunikativní. Důraz byl kladen rovněž na mezipředmětové vztahy. Studenti využívají v projektech znalosti a dovednosti z matematiky, fyziky a anglického jazyka.

Projekty realizovali žáci ze Základní školy 8. května Šumperk, Základní školy Karla staršího ze Žerotína v Bludově a studenti Vyšší odborné a Střední průmyslové školy v Šumperku. Práce na projektech je bavila a s většinou konstrukcí a programů si poradili. Pomohli mi také s úpravou vlastních materiálů, aby byly srozumitelnější a přehlednější. Své znalosti zúročili i na mezinárodní soutěži Robot Challenge 2011, kde obsadili čtvrté místo v těžké konkurenci zejména vysokých škol z celé Evropy.

Samotná práce je tvořena sedmi kapitolami. Prvních pět kapitol popisuje stavebnici Lego Mindstorm NXT a možnosti jejího programování, dále dostupné učební materiály a soutěže. Poslední dvě kapitoly se zabývají tvorbou vlastního výukového materiálu, zejména rozбором jednotlivých projektů.

Conclusions

Teaching of programming can take various forms. One of them is a use of programming languages specifically designed for teaching, for example turtle graphics in Comenius LOGO environment or its variations. These programming languages do not produce classic executable applications. The students' motivation is not too high then. On the other hand there are some commercial programming environments and languages which do not have this problem. But when we use them we have to deal with things that are irrelevant to education. I've decided to teach a differential way. My students can learn principles of programming with the help of a successful robotic kit Lego Mindstorm NXT.

I managed to create multimedia teaching materials that can serve as an inspiration for teachers or directly to students in their learning. All the programmes are implemented in two versions. The first one in iconographic NXT-G environment is suitable for pupils of elementary schools (grade 4 to 8). The second one in object-oriented language Java with a special interface LeJOS is suitable for ninth grade students and for high school and university students. Teaching is carried out through projects and their topics solve everyday life problems, for example traffic lights, automatic gate openers, control of the temperature and water level etc. As a result, students can easily understand the problems and are able to propose their own solutions. The projects are usually larger so they need teamwork. Thanks to it we can develop students' social, communicative and personal competences. The emphasis was also placed on relations between IT and other school subjects. Students use Maths, Physics and English skills and knowledge in the projects.

The projects were realized by pupils of two elementary schools (ZŠ 8.května Šumperk and ZŠ Karla staršího ze Žerotína Bludov) and students of Higher and High Industrial School in Šumperk. They enjoyed their work and managed with most of structures and problems by themselves. They also helped me with the improvement of the materials and made them clearer and more understandable. They used their knowledge in an international competition Robot Challenge 2011 in Vienna where they took the 4th place in a tough competition, especially from universities throughout Europe.

The thesis consists of seven chapters. The first five chapters describe the kit Lego Mindstorm NXT and possibilities of its programming. Further more there are the available instructional materials and competitions. The last two chapters deal with the creation of my own teaching materials, mainly with the analysis of individual projects.

Reference

- [1] Ferrari, Mario, Ferrari, Giulio *Building robots with LEGO Mindstorm NXT* Syngress, 2007. 448 s. ISBN 15-9749-152-7.
- [2] Kelly, James Floyd *Lego Mindstorms NXT-G Programming Guide* Apress, 2007. 196 s. ISBN 15-9059-871-7.
- [3] Carnegie Mellon University - Robotics Academy *Introduction to robotics* Multimediální DVD, 2006.
- [4] Staníček, Petr *CSS Kaskádové styly* Computer press, 2003. 192 s. ISBN: 80-7226-872-4.
- [5] Rinkens, Tim a kolektiv autorů *Java for Lego Mindstorm* - <http://lejos.sourceforge.net/> Elektronická publikace, 2010.
- [6] Dlouhý, Martin, Winkler, Zbyněk <http://robotika.cz> Elektronická publikace, 2010.
- [7] Krhánek, Daniel <http://www.hluchak.cz/?q=robotchallenge2011vysledky> Elektronická publikace, 2011.
- [8] White wolf consulting *VÝZKUMNÁ ZPRÁVA - Důvody nezájmu žáků o přírodovědné a technické obory* MŠMT, 2009. 84 s.

8. Obsah příloženého CD

`index.html`

Hlavní stránka multimediálního učebního materiálu vytvořeném ve formátu html. Lze ji otevřít v libovolném webovém prohlížeči, který má podporu Javascriptu a instalovaný Adobe Flash plugin verze 9 nebo vyšší.

`autorun.inf`

Soubor s nastavením automatického načtení a spuštění materiálu v operačním systému Windows.

`autorun`

Skript s nastavením automatického načtení a spuštění materiálu v operačním systému Linux.

`AutorunPro.EXE`

Program pro operační systém Windows pro spuštění výchozího webového prohlížeče a zobrazení titulní stránky výukového materiálu.

`readme.txt`

Instrukce pro spuštění výukového materiálu, včetně požadavků pro jeho provoz.

`doc/`

Dokumentace práce ve formátu pdf, vytvořená dle závazného stylu KI PŘF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování pdf souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

Navíc CD obsahuje:

`audio/`

Audio přehrávač včetně licenčního souboru a složky se soubory ve formátu MP3 obsahující slovíčka ze sekce english corner vygenerované řečovým syntetizérem.

`dp.SyntaxHighlighter/`

Javascript sloužící ke zvýrazňování syntaxe zdrojových kódů různých programovacích jazyků na webových stránkách.

`engine/`

Javascriptový projektor obrázků Visuallightbox založený na technologii Ajax.

`english/`

Webové stránky a dokumenty pdf tvořící sekci english corner.

images/

Obrázky ve formátu jpg a png použité v grafickém vzhledu stránek.

projekt1_X/

Složky s projekty v první sadě.

projekt2_X/

Složky s projekty v druhé sadě.

styles/

Kaskádové styly řídící vzhled webových stránek.

video/

Video soubory použité v projektech a přehrávač JW FLV Player.

video2/

Video soubory použité v sekci základy a přehrávač JW FLV Player s playlisty.

zaklady/

Webové stránky obsahující úvodní seznámení s robotem a programováním.

projekty1.html

Webová stránka obsahující rozcestník první sady projektů.

projekty2.html

Webová stránka obsahující rozcestník druhé sady projektů.

U veškerých odjinud převzatých materiálů obsažených na CD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu.