

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

NÁVRH ČTYŘNOHÉHO ROBOTA PRO SOUTĚŽ ROBOCUP

FOURLEGGED RACER ROBOT DESIGN

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ADAM BARCAJ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. STANISLAV VĚCHET, PH. D.

BRNO 2007

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2009/2010

Zadání závěrečné práce

student(ka): Adam Barcaj

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh čtyřnohého robota pro soutěž RoboCup

v anglickém jazyce:

Fourlegged racer robot design

Stručná charakteristika problematiky úkolu:

Cílem práce je návrh čtyřnohého robota pro soutěž v robotickém fotbale RoboCup. Jako základ pro sestavení soutěžního robota poslouží stavebnice Bioloid, která umožňuje rychlý vývoj hardwaru a proto je možné se více zaměřit na řešení problému řízení celého robota pro potřeby soutěže.

Cíle bakalářské práce:

Podrobně prostudujte pravidla soutěže RoboCup

Prostudujte v současnosti používané soutěžní roboty

Na základě provedené analýzy proveďte návrh konstrukce

Vybranou konstrukci postavte a oživte.

Seznam odborné literatury:

www.robotika.cz

www.megarobot.net

www.hobbyrobot.cz

Vedoucí bakalářské práce: Ing. Stanislav Věchet, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/2010.
V Brně, dne

L.S.

prof. RNDr. Ing. Miloš Šeda, Ph.D. doc.
Ředitel ústavu

RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

LICENČNÍ SMLOUVA

ABSTRAKT

Tato bakalářská práce se zabývá hlavními součástmi stavebnice BOILOID, vývojovým prostředím RoboPlus a následnou konstrukcí čtyřnohého robota.

Cílem této práce bylo zkonstruovat a oživit čtyřnohého robota, který by prozkoumal možnosti nejen stavebnice jako takové, ale i vývojového prostředí, které bylo pro tuto stavebnici vyvinuto, a tím ověřit použitelnost jednotlivých částí a vývojového prostředí v robotickém fotbale RoboCup.

ABSTRACT

This bachelor's thesis deals with the main components of the BIOLOID kit, the development environment RoboPlus and subsequent construction of a fourlegged robot.

The aim was to construct and vitalize a fourlegged robot, that would explore the potential not only of the kit as such, but also of the development environment that has been created for this kit, and thereby verify applicability of the individual components and the development environment in the robotic football RoboCup.

Klíčová slova

čtyřnohý mobilní robot, RoboCup, stavebnice BIOLOID

KEYWORDS

fourlegged mobile robot, Robocup, BIOLOID kit

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Stanislavu Věchetovi, Ph. D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

Obsah:

	Zadání závěrečné práce.....	3
	Licenční smlouva.....	5
	Abstrakt.....	7
	Poděkování.....	9
1	Úvod.....	13
2	STAVEBNICE BIOLOID.....	15
2.1	Řídící jednotka CM - 510.....	15
2.2	Servo motory AX - 12+.....	16
2.3	Čidla.....	17
2.3.1	IR senzor.....	17
2.3.2	DMS (Distance measurement sensor).....	18
2.4	Gyroskop.....	19
2.5	Bezdrátová komunikace (RC – 100, ZIG – 100/110).....	19
3	VÝVOJOVÉ PROSTŘEDÍ.....	21
3.1	Motion editor.....	21
3.2	Task editor.....	22
4	KONSTRUKCE ČTYŘNOHÉHO ROBOTA.....	25
4.1	Puppy (štěně).....	25
4.2	Vlastní konstrukce robota.....	25
4.3	Konstrukce nohou.....	26
4.4	Kloub pro otáčení robota.....	27
4.5	Osazení periferních zařízení.....	27
4.6	Propojení periferních zařízení s řídící jednotkou.....	28
5	ŘÍZENÍ ROBOTA.....	31
5.1	Řízení pomocí ovladače.....	31
5.2	Autonomní chování.....	33
6	Závěr.....	37
	Seznam použité literatury.....	39

1 ÚVOD

Stavebnice BIOLOID je plastová a v mnoha ohledech připomíná českou stavebnici Merkur, kterou vyrábí korejská firma Robotis, Inc. a je dostupná ve čtyřech verzích. Pro tuto práci byla využita verze BIOLOID Premium kit, která je v mnoha ohledech rozdílná od ostatních verzí stavebnice. Hlavní výhodou tohoto provedení je nová řídicí jednotka CM – 510 postavená na procesoru ATmega 2561. Čidla se v této verzi připojují periferně zvlášť, nemusí být k dispozici dřívější modul senzorů AX – S1, který nebylo možné dále rozšiřovat. Do této verze firma umístila jak čidla na detekci objektu, tak i gyroskop a modul pro bezdrátovou komunikaci. Modul pracuje na standardu ZigBee a v práci je použit pro řízení robota pomocí ovladače.

Vývojové prostředí RoboPlus představila firma v roce 2010. Toto prostředí je nástavbou již dříve využívaných prostředí Motion editor a Task editor. Motion editor slouží k nachystání pohybů, které se pak volají v Task editoru, což je nástroj pro programování robota jako takového. Velké změny prodělal hlavně program Motion editor, který byl změněn od základů. Graficky se program velmi zpřehlednil a zjednodušil, což má za následek efektivnější stavbu a práci s programem. Zásadní inovací pro uživatele bylo umožnění chystání pohybů pro vlastní robota. Absence této možnosti v dřívější verzi programu ztěžovala práci s roboty, kteří nebyli postaveni podle návodu. Jediné, co v programu chybí, je 3D náhled na vlastní robota. Grafická stránka Task editoru se změnila pouze částečně, byla přidána přehledná nabídka příkazů a tím se zvýšila celková přehlednost programu. Programování se velmi podobá jazyku C, ve kterém je možnost robota programovat.

Jedním z cílů práce bylo postavit čtyřnohého robota, který by využil co největšího počtu čidel a odzkoušel tím použitelnost řídicí jednotky a senzorů v robotickém fotbale. Konstrukce robota vychází z dříve postavených robotů, kteří byli postaveni podle návodu. Nejvíce inspirativním robotem byl Puppy (štěně), který sloužil jako předloha pro vlastní konstrukci čtyřnohého robota. Konstrukce je velmi jednoduchá – obsahuje devět servo motorů, tři čidla, ZigBee modul a řídicí jednotku.

Dalším cílem bylo oživení vybrané konstrukce a mělo prozkoumat možnosti a použitelnost vývojového prostředí RoboPlus v robotickém fotbale. Navržený robot vlastní konstrukce má na výběr řízení pomocí ovladače nebo autonomního chování. Řízení pomocí ovladače je realizováno přes ZigBee modul a umožňuje uživateli řídit robota jednoduchými povely. Autonomní chování je realizováno řídicí jednotkou, která zpracovává informace z dostupných senzorů.

2 STAVEBNICE BIOLOID

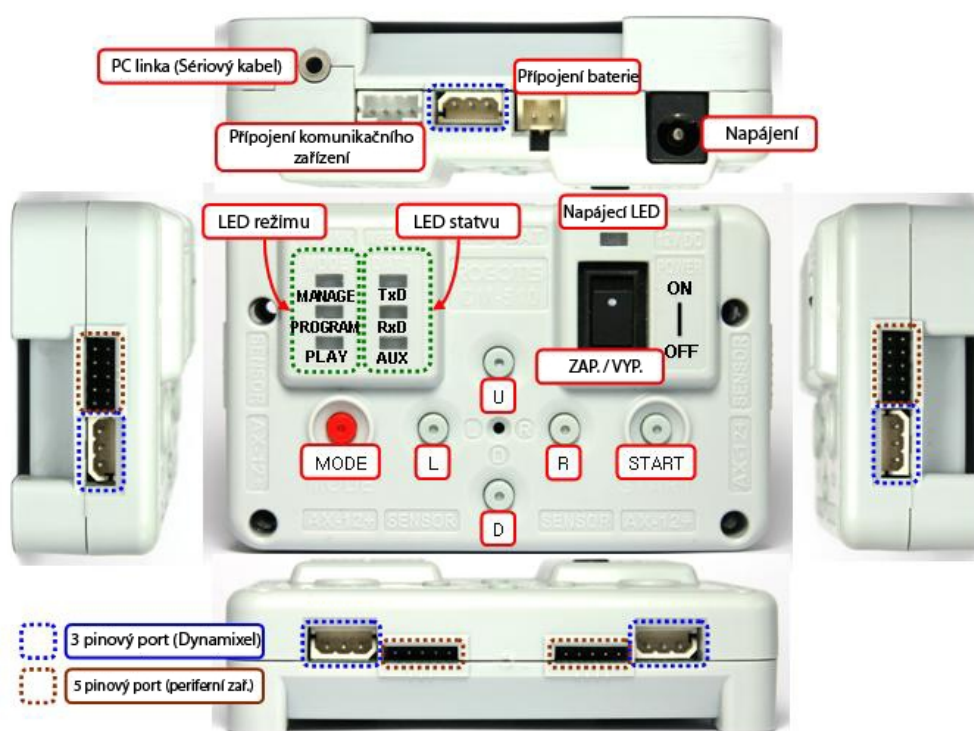
Jednoduchá stavebnice s plastovými součástkami se nápadně podobá české stavebnici Merkur. Na trhu je stavebnice k dostání ve čtyřech variantách. Pro tuto práci byl využit BIOLOID Premium kit obsahující osmnáct servo motorů AX – 12+, řídicí jednotku CM – 510, součástky k sestavení jakéhokoliv robota podle návodu a senzory.

Stavebnice slouží k jednoduchému vývoji hardwaru. Samotného robota je možno postavit buď podle návodu, a nebo navrhnout vlastní konstrukci. Protože jsou návody na postavení robotů psány na jinou verzi stavebnice, než byla k dispozici k této práci, byla montáž robotů poměrně složitá. Hlavní změnou od ostatních verzí stavebnice je řídicí jednotka a periferně připojovaná čidla.

Provedení některých součástek není úplně kvalitní, např. rozchod děr pro přidělení řídicí jednotky má větší rozměr než na samotné jednotce, což má za následek přišroubování pouze dvěma šrouby.

2.1 Řídicí jednotka CM - 510

Řídicí jednotka je postavena na procesoru ATmega 2561 a nabízí dvakrát více paměti, tj. 16 kB, než starší Cm – 5 a deklaraci až 50 proměnných. S počítačem komunikuje přes sériovou linku RS – 232 a s motory po sériové lince TTL.



Obr. 1: Řídicí jednotka CM – 510 [1]

Řídicí jednotka obsahuje: [1]

- PC Linku: propojení CM – 510 a počítače pomocí sériové linky – RS 232
- Konektor pro připojení komunikačních zařízení: používán pro bezdrátovou komunikaci přes ZigBee, infračervený přijímač a pro připojení jiných modulů
- Porty pro připojení periferních zařízení: připojení čidel nebo jiných periferních zařízení

- Sběrnici AX – 12 + : připojení servo motorů
- Tlačítka: START pro zapnutí vybraného režimu, MODE pro změnu pracovního režimu, U/D/L/R sloužící jako vstup při vykonávání programu
- Konektor pro baterii
- Kontrolky LED: kontrola stavu vypnuto/zapnuto, kontrolky pro zobrazení aktuálního provozního režimu CM – 510

Proti přetížení je ve spodní části vyměnitelná pojistka na celkový proud 10 A. Jednotka obsahuje také čidlo pro zachycení zvuku, teplotní čidlo a voltmetr.

2.2 Servo motory AX - 12+



Obr. 2: Servo motor AX – 12+
[12]

Jednou z nejdůležitějších součástí stavebnice jsou digitální servo motory. Do řídicí jednotky se připojují pomocí TTL sériové linky. Tato linka dokáže komunikovat s řídicí jednotkou rychlostí až 1 Mb/s. Řídicí jednotka pošle do motoru informaci na jakou pozici a jak rychle se má otočit. Každý servo motor snímá pozici, teplotu, el. napětí, proud a zátěž. Do řídicí jednotky pošle motor odezvu s úhlem natočení, úhlovou rychlostí a točivým momentem. Při přetížení motoru, např. znemožnění dostat se do požadované pozice kvůli překážce, motor nepošílá paket, který má potvrdit dosažení žádané polohy, a řídicí jednotka tudíž může předejít přetížení, přehřátí a následnému zničení motoru. Každý motor má své identifikační číslo (dále jen ID), podle kterého motor pozná, jestli poslaná instrukce patří jemu, nebo ne. ID motoru lze v případě potřeby změnit.

Každý motor má 1024 možných poloh rozdělených do 300°. Zbývajících 60° potřebných do přetočení motoru dokola je zakázáno (viz Obr. 3). Jediným případem, kdy jsou tyto úhly povoleny, je použití motoru jako kola.

Parametry servo motoru AX – 12+ (hodnoty platné pro vstupní napětí 10 V): [11]

- Převodový stupeň: 1/254
- Točivý moment: 16,5 kg cm
- Rychlost otáčení: 0,196 s/60°



Obr. 3: Rozsah úhlů řízení [1]

2.3 Čidla

Pro autonomní chování robota se využívá různých čidel, která budou popsána v této kapitole. Čidla se zapojují přes sériový kabel do portů v řídicí jednotce. Lze připojit jakékoliv čidlo, které je obsaženo ve stavebnici nebo které si dokoupíme, popř. sami vyrobíme. Čidla, která jsou obsažena přímo ve stavebnici, nejsou pro úplné autonomní chování dostačující – při využití pouze těchto senzorů lze naprogramovat jen nejjednodušší chování robota.

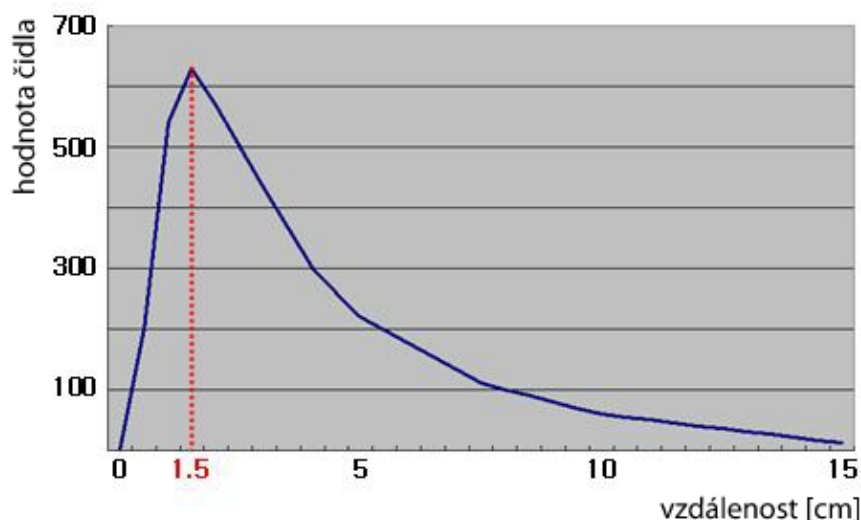
2.3.1 IR senzor



Obr. 4: IR senzor pro detekci objektu [1]

Čidlo slouží k detekci objektu a funguje na principu odrazu infračerveného světla. Nemůže fungovat pro měření vzdálenosti mezi ním a objektem, protože odraz světla od objektu závisí na barvě, lesku a na okolním prostředí.

Čidlo obsahuje vysílač a přijímač. Když je objekt příliš blízko, hodnota v senzoru se prudce zmenšuje, protože úhel, ve kterém se infračervené světlo odráží, není dostatečný. Je těžké určit, kdy již čidlo nerozpozná objekt – může rozpoznat i více než 15 cm.



Obr. 5: Závislost hodnoty čidla na vzdálenosti od objektu [1]

Graf ukazuje závislost hodnoty senzoru na vzdálenosti objektu od čidla. Hodnoty grafu jsou zobrazeny pro bílý papír. Toto čidlo je velmi nepřesné, proto slouží pouze jako detekce objektu na malou vzdálenost.

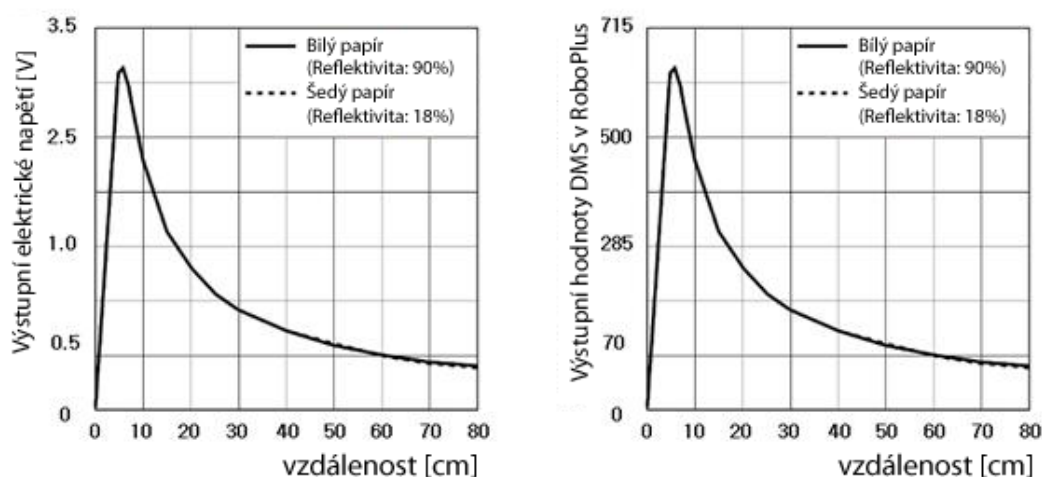
2.3.2 DMS (Distance measurement sensor)



Obr. 6: DMS Sharp [1]

Čidlo slouží pro detekci objektů a pro měření vzdálenosti mezi čidlem a pevně uchyceným objektem. Pracuje na principu odrazu infračerveného světla. Hodnoty napětí na výstupu (a tudíž i hodnoty v programu RoboPlus) se téměř nemění s leskem nebo barvou objektu (viz. Obr. 7), a proto je velmi přesné.

Obsahuje vysílač a přijímač. Měření probíhá každých 38,3 ms. Hodnota napětí se zvyšuje s přiblížením objektu. Při přiblížení objektu blíže než 7 cm se hodnota začne prudce snižovat, protože úhel pro odraz infračerveného světla není dostatečně velký.



Obr. 7: Závislost napětí a hodnot čidla na vzdálenosti [1]

Z grafů je vidět, že hodnoty se pro bílý (reflexivnost 90%) a šedý papír (reflexivnost 18%) téměř neliší. Senzor měří vzdálenost mezi 10 a 80 cm, lze počítat i s přesným měřením lehce pod 10 cm.

2.4 Gyroskop

Pro balancování na nerovných površích je nově ve stavebnici zařazen gyroskop, který měří úhlovou rychlost v osách x a y, tedy v osách, které určují rovinu desky gyroskopu. Vypočítá, na kterou stranu je robot nakloněn nebo kde je větší síla. Dokáže měřit úhlovou rychlost až 300 °/s na každou stranu naklonění. Gyroskop je výhodné použít na dvounohých robotech, na čtyřnohém robotovi, který je součástí této práce, této součástky využito není, protože by vložení tohoto modulu bylo vzhledem ke konstrukci zbytečné.



Obr. 8: Gyroskop [1]

2.5 Bezdrátová komunikace (RC – 100, ZIG – 100/110)

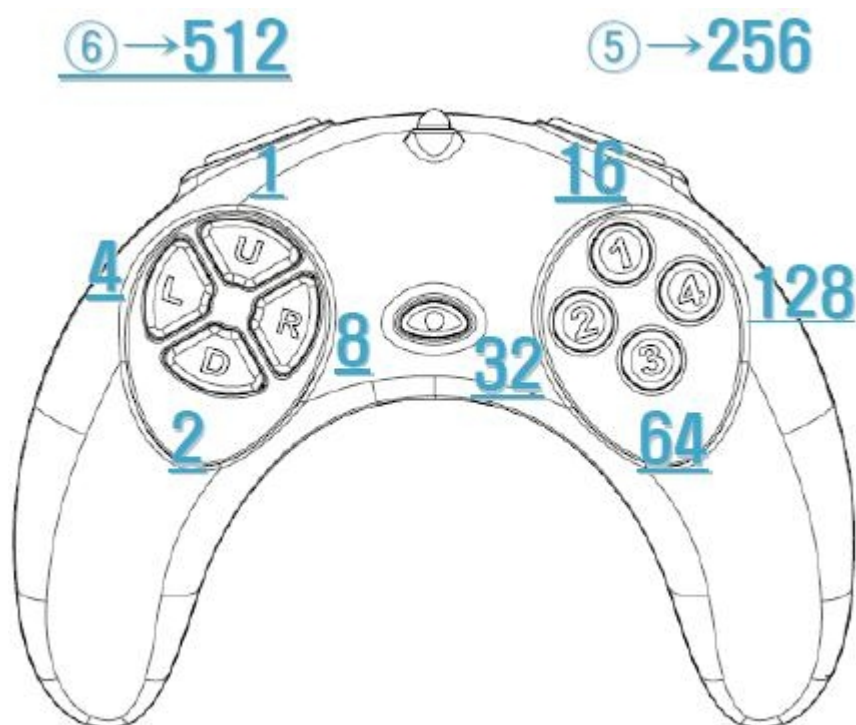
Součástí stavebnice je také modul ZIG – 100/110, využívající standardu ZigBee buď pro bezdrátovou komunikaci s počítačem, nebo pro ovládání robota pomocí ovladače RC – 100. Ovladač lze propojit s robotem také pomocí přijímače infračerveného signálu, ale protože by zařízení musela být v přímé radiové viditelnosti, je takové řešení neefektivní a využívá se ZigBee modulu.

ZigBee pracuje v pásmu 2,4 GHz a přenosová rychlost je maximálně 240 kb/s, běžně je ale přenosová rychlost kolem 50 kb/s. Zařízení nemusí být v přímé radiové viditelnosti.



Obr. 9: Ovladač RC - 100 a ZigBee modul ZIG – 100/110 [13]

Samotný ovladač má deset tlačítek a bezdrátově posílá hodnoty stisknutých tlačítek do řídicí jednotky. Každé tlačítko má svoji hodnotu nastavenou tak, aby při jakékoliv kombinaci nedošlo ke kolizi s jinou kombinací. Tlačítka mají hodnoty od 1 do 512 vždy jako mocniny čísla dvě (viz obr. 10).



Obr. 10: Hodnoty tlačítek ovladače RC – 100 [1]

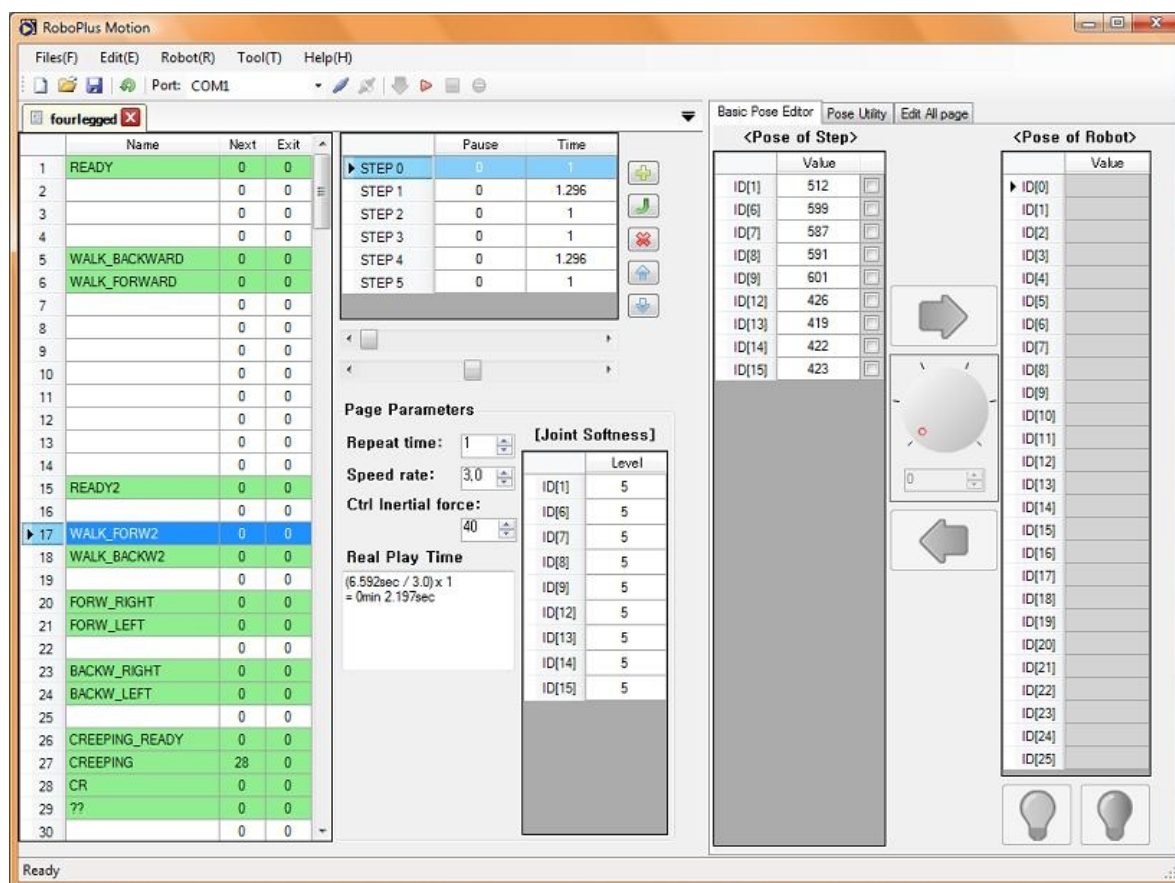
3 VÝVOJOVÉ PROSTŘEDÍ

Pro programování vytvořených robotů vytvořila firma Robotis, Inc. vlastní vývojové prostředí. V roce 2010 představila novou verzi – RoboPlus, kde je možnost nastavit čidla, otestovat funkci servo motorů aj. Hlavními dvěma nástroji pro vytvoření chování robota je Motion editor a Task editor a dále RoboPlus Terminal pro řízení robota bezdrátově přes počítač. RoboPlus Terminal v této práci nebudu popisovat blíže, protože pravidla RoboCupu nedovolují nijak zasahovat do chování robota na hřišti přes počítač.

Motion editor je nástroj pro zrychlení práce s robotem. Při uložení vzniká soubor s příponou .mtn, který je třeba přes sériovou linku nahrát do řídicí jednotky. Po nahrání souboru je možné jednoduše volat jednotlivé pohyby z programu Task editor. Oba dva programy jsou spolu těsně propojené. Z Task editoru není možné bez souboru volat jednotlivé pohyby, naproti tomu programování jednotlivých pohybů pouze v Task editoru by bylo velice složité a zdlouhavé. Místo Task editoru lze jako propojení použít programovacího jazyka C.

3.1 Motion editor

Motion editor je grafické prostředí pro nachystání jednotlivých pohybů. Chystání pohybů pro vlastní roboty je možné teprve s novou verzí programu a balíčkem RoboPlus, který vyšel teprve v roce 2010. Dříve bylo toto předprogramování pouze pro roboty postavené podle návodu.



Obr. 11: RoboPlus Motion editor

V programu lze nachystat celkem 256 pohybů, každý může mít až 6 kroků. Každý krok představuje požadovanou pozici motorů a rychlost, jakou se má provést. Při tvorbě pohybů pro vlastní robot není dostupný 3D model – ten je dostupný pouze pro roboty postavené podle návodu.

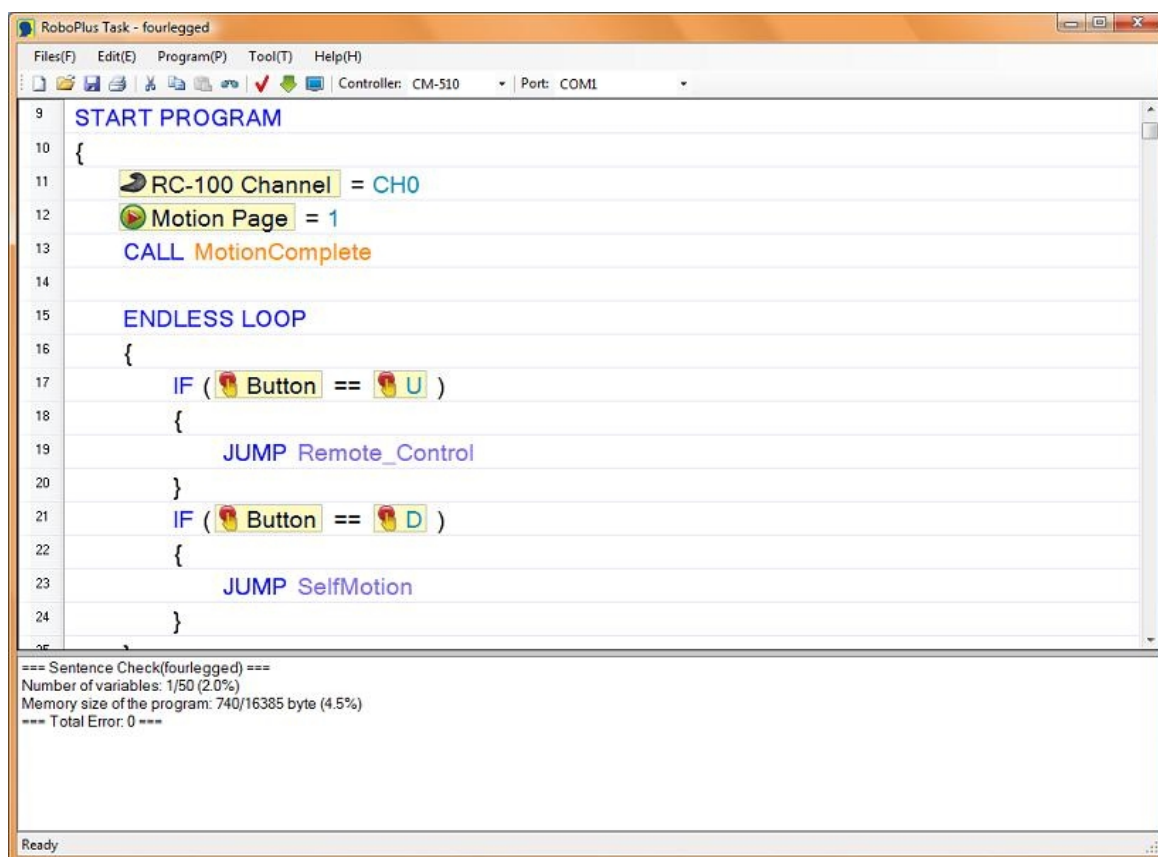
▶ 17	WALK_FORW2	0	0	<Pose of Step>		
		Pause	Time		Value	
STEP 0		0	1	ID[1]	512	☐
▶ STEP 1		0	1.296	ID[6]	670	☐
STEP 2		0	1	ID[7]	587	☐
STEP 3		0	1	ID[8]	691	☐
STEP 4		0	1.296	ID[9]	601	☐
STEP 5		0	1	ID[12]	426	☐
				ID[13]	355	☐
				ID[14]	422	☐
				ID[15]	327	☐

Obr. 12: Výběr kroku pohybu a nastavení servo motorů

Pohyb nemusí být vytvořen pouze ze šesti cílových pozic. Je možné nastavit následující stránku pohybu a tím prodloužit pohyb na libovolný počet stránek. Tímto způsobem lze nastavit i nekonečnou smyčku pohybů, problém nastává v situaci, kdy chceme takový pohyb přerušit.

Samotné hodnoty požadovaných pozic nemusíme zadávat ručně, a tedy počítat přesné natočení motorů po dokončení kroku, ale je možné některé motory vypnout a nastavit je ručně, po opětovném zapnutí motorů pouze zapíšeme snímané hodnoty do určitého kroku. V kroku lze nastavit jak rychle se mají motory do požadovaného natočení dostat a čas pozastavení před vykonáním samotného kroku.

3.2 Task editor



Obr. 13: RoboPlus Task editor

Task editor je jednoduchý programovací jazyk sloužící k naprogramování chování robota. Editor je „klikací“ prostředí, které má urychlit programování. Po čase je ale toto prostředí velmi pomalé, protože i pro vložení komentáře se musí myš „proklikat“ přes nabídku příkazů. Slouží nejen jako propojení nachystaných pohybů s robotem, ale lze v něm naprogramovat chování robota na vnější podněty, nejčastěji na detekované objekty z čidel. Po volání nachystaného pohybu musí být vždy celý pohyb dokončen, a proto je obtížné naprogramovat rychlé a okamžité zareagování např. na detekci objektu.

Do řídicí jednotky lze nahrát až 16 kB dat a program umožňuje nadeklarovat až 50 proměnných.

Typ	Příkaz	Vysvětlivka
START & END	Start program	Začátek programu
	Exit program	Předčasné ukončení programu
	{	Začátek bloku programu
	}	Konec bloku programu
	//	Komentář nebo poznámka
EXECUTE	Compute	Aritmetická nebo bitová operace
	Load	Načtení hodnoty
	Label	Pozice pro skákání
	Jump	Skok na pozici
CONDITION	If	Podmínka
	Else if	Když není splněna podmínka zeptá se na podmínku za Else if
	Else	Když není splněna ani jedna z podmínek vykoná program to, co je za Else
LOOP	Endless loop	Nekonečný cyklus (while(1))
	Condition loop	Podmíněný cyklus (while(podmínka))
	Count loop	Cyklus s daným počtem průběhů (For)
	Break loop	Předčasné ukončení smyčky
	Conditional stand	Podmíněné pozastavení programu
FUNCTION	Make function	Vytvoření funkce
	Call function	Volání funkce
	Exit function	Předčasné ukončení funkce (Return)

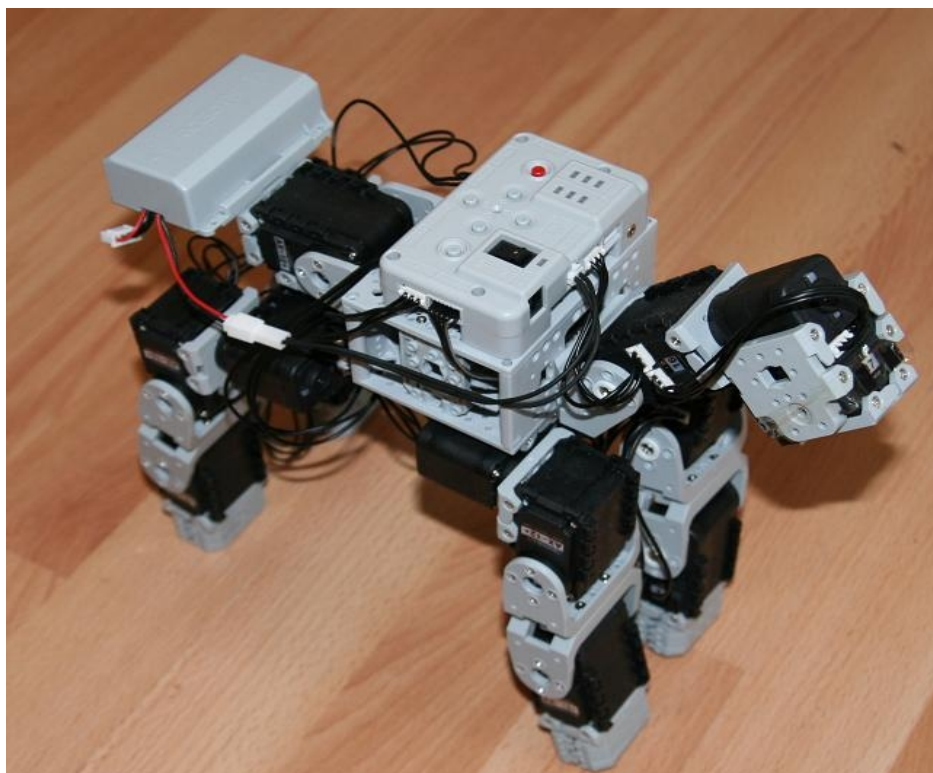
Tabulka 1: Příkazy v prostředí Task editor

4 KONSTRUKCE ČTYŘNOHÉHO ROBOTA

Pro osvojení stavebnice jsem nejdříve postavil několik robotů podle návodů. Nejnáročnějším a nejinspirativnějším byl robot „Puppy“ (štěně), který se stal předlohou pro vlastní konstrukci. Konstrukce většiny robotů je velmi jednoduchá, a proto je lze poměrně rychle sestavit. Přesto stavění provázely určité problémy. Nejdříve byla k dispozici stavebnice BIOLOID Comprehensive kit a teprve poté stavebnice Premium kit, a proto byl velký problém podle návodu postavit celého robota, protože ve stavebnicích jsou určité rozdíly. Největším problémem je usazování čidel, které nemají průměry děr, které jsou používány ve stavebnici, a proto je jejich uchycení na klasickou součástku téměř nemožné.

4.1 Puppy (štěně)

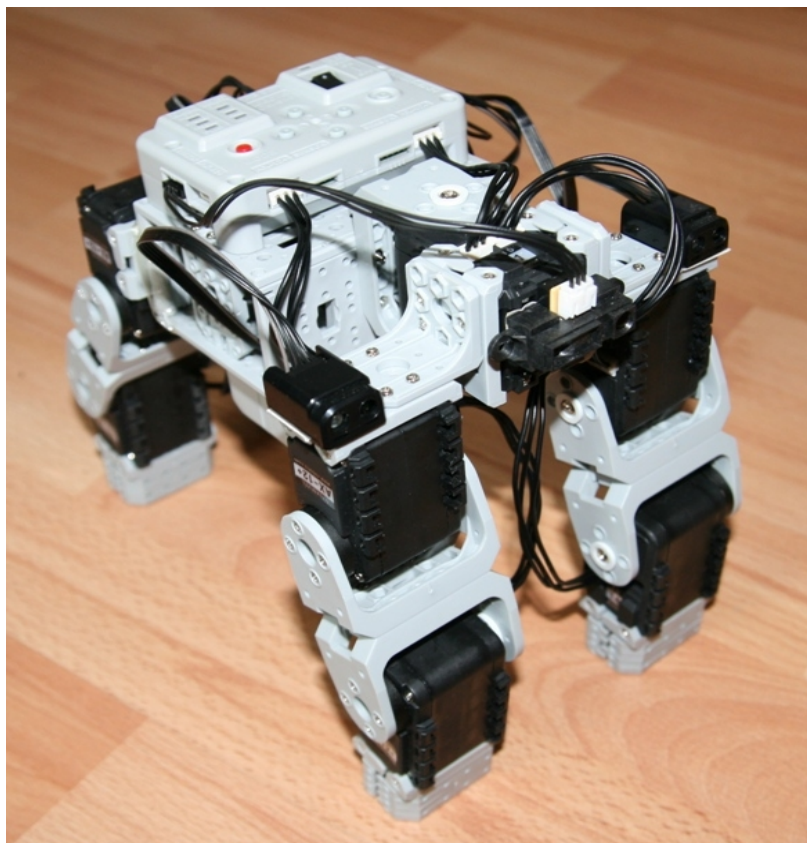
Tento robot je ze série advanced level (tedy pro pokročilé). Je vybaven pouze jedním čidlem na detekci objektu. Celá konstrukce je tvořena patnácti servo motory, kde každou nohu tvoří tři motory, dva tvoří hlavu a jeden servo motor bedra (viz Obr. 15). Vzhledem k rozdílnosti návodu a stavebnice trvalo postavení robota velmi dlouho a celý robot je díky umístění baterie (a tím i posunutí těžiště směrem dozadu) mírně nestabilní.



Obr. 14: Puppy (štěně)

4.2 Vlastní konstrukce robota

Vlastní konstrukce robota je složena z devíti servo motorů, kde vždy dva servo motory tvoří jednu nohu a jeden motor slouží jako kloub k otáčení. Robot má kvůli uchycení omezený rádius otáčení, a proto je jeho pohyb v prostoru mnohdy omezen. Řídící jednotka je umístěna na hlavním těle robota tak, aby byla co nejlépe přístupná, baterie potom pod hlavním tělem tak, aby snížila těžiště a tím se zlepšila celková stabilita.



Obr. 15: Konstrukce robota

4.3 Konstrukce nohou



Obr. 16: Konstrukce nohou

Nohy jsou v konstrukci řešeny velice podobně jako u předchozího robota s tím rozdílem, že tato konstrukce představuje pouze dva servo motory na nohu, které umožňují pohyb pouze v jedné ose. Při návrhu konstrukce nohy se vycházelo z výše uvedeného robota odebráním jednoho servo motoru. Zbývá konstrukce samotné nohy zůstala zachována.

Kvůli uchycení servo motorů nelze využít plný rozsah otočení ani jednoho z nich. Každý servo motor má svoje ID (viz Tabulku 2).

	ID vrchního servo motoru	ID spodního servo motoru
Levá přední noha	12	14
Pravá přední noha	13	15
Levá zadní noha	6	8
Pravá Přední noha	7	9

Tabulka 2: ID servo motorů

4.4 Kloub pro otáčení robota



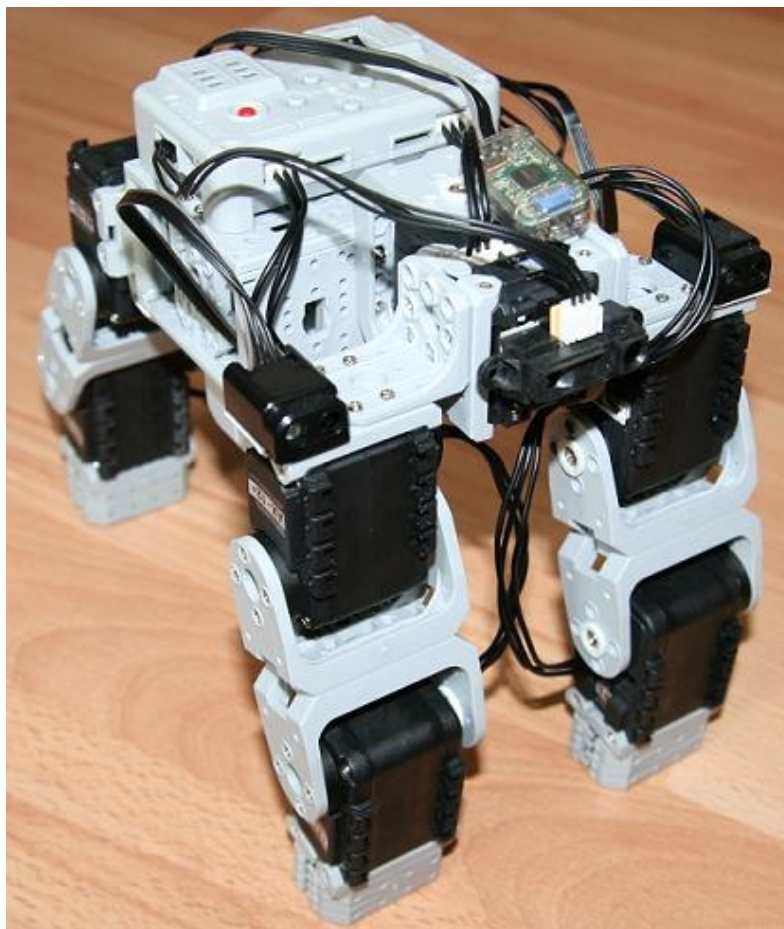
Obr. 17: Usazení kloubu

Samotný kloub je tvořen jedním servo motorem (ID 1), který je umístěn nad předníma nohama (viz Obr. 17). Protože je osa otáčení kloubu směrem k tělu robota, natáčí nejen oběma předníma nohama, ale spolu s nimi je natáčen také samotný kloub, který tvoří další zátěž. Možnost natočení předních nohou je velice omezená konstrukcí robota, a proto je nutné tento fakt zohlednit při natáčení, aby nedošlo k poškození motoru.

4.5 Osazení periferních zařízení

Robot je osazen třemi čidly a ZigBee modulem, tedy jsou využita všechna čidla pro detekování objektu a modul pro bezdrátovou komunikaci.

Čidla slouží pro detekování objektu a následnému vyhnutí objektu. Všechny senzory jsou osazeny v přední části robota, a proto není možné se vyhnout překážce, která je při couvání robota za ním. Jsou využity dva IR senzory pouze pro detekci objektu a jeden senzor pro detekci a pro měření vzdálenosti (DMS). IR senzory jsou umístěny nad oběma předníma nohama, tedy na krajích robota. DMS senzor je umístěn uprostřed robota (viz obr. 18).



Obr. 18: Osazení čidel

ZigBee modul slouží pro bezdrátovou komunikaci ovladače a řídicí jednotky. Tento modul je použit kvůli jeho výhodám oproti obyčejnému IR přijímači. Tento standard má rychlejší přenos a jeho hlavní výhodou je (podobně jako u Bluetooth) nepotřebnost jednotlivých zařízení být v přímé radiové viditelnosti. Modul je vidět ve vrchní části Obr. 18, dále je již skryt v těle robota.

4.6 Propojení periferních zařízení s řídicí jednotkou

Důležité bylo propojit jednotlivá periferní zařízení s řídicí jednotkou. Protože jsou motory propojovány sériovou linkou, nemusí být každý motor propojen s jednotkou zvlášť, ale je důležité, aby jednotlivé kabely nakonec spojily všechny motory (viz Tabulku 3).

Odkud	Kam
ID8	ID6
ID6	CM – 510
ID9	ID7
ID7	CM – 510
ID15	ID13
ID13	CM – 510
ID14	ID12
ID12	CM – 510

Tabulka 3: Propojení servo motorů s řídicí jednotkou

Propojení čidel s řídicí jednotkou je velice jednoduché, každé čidlo se připojuje zvlášť do portu na řídicí jednotce. Jednotka nabízí připojení až šesti senzorů do vyhrazených portů. V robotovi jsou využity pouze tři porty pro výše popsaná čidla (viz Tabulku 4).

čidlo	port
IR senzor levý	5
IR senzor pravý	6
DMS	1

Tabulka 4: Připojení čidel do portů

Ostatní periferní zařízení, jako baterie a ZigBee modul, jsou zapojeny do vyhrazených portů. Do portu pro ZigBee modul je možno zapojit i IR přijímač.

5 ŘÍZENÍ ROBOTA

Důležitou součástí práce bylo naprogramování chování robota. Chování je rozděleno na dvě části: řízení pomocí ovladače a autonomní chování robota. Bylo využito programovacího prostředí RoboPlus.

Nejprve byly nachystány pohyby, se kterými se pracovalo ve vlastním programu. Bylo nachystáno celkem třináct pohybů (viz Tabulku 5). Pohyby pro natočení robota vlevo nebo vpravo nejsou nachystány, protože natočení jednoho motoru se snadněji řeší v samotném programu. Na druhou stranu programování rozsáhlejší soustavy motorů v Task editoru by bylo velmi zdlouhavé a pravděpodobně by došlo k naplnění paměti 16 kB. Stejně jako pouhé natočení robota by bylo možné řešit i chůzi a couvání doleva resp. doprava. Při takovém řešení je problém v rozsahu pohybů, které robot zvládá při chůzi rovně nebo při zatáčení. Při zatáčení je rozsah pohybu servo motorů menší – robot by si při pouhém natočení kloubu podkopával nohy a padal. Z toho důvodu bylo přistoupeno k nachystání čtyř pohybů navíc, aby nebyl některý ze servo motorů při takovém pohybu poškozen.

Název pohybu	Popis pohybu	Strana	Počet kroků
READY	Základní pozice	1	1
READY2	Základní pozice 2	15	1
WALK_FORWARD	Chůze vpřed	17	6
WALK_BACKWARD	Chůze vzad	18	6
FWD_RIGHT	Chůze dopředu a doprava	20	6
FWD_LEFT	Chůze dopředu a doleva	21	6
BWD_RIGHT	Chůze dozadu a doprava	23	6
BWD_LEFT	Chůze dozadu a doleva	24	6
CREEPING_READY	Lehnutí	26	1
WU_EXERCISE	Rozcvička	27,28	7
CREEPING_BACK	Plazení vzad	29	3
STAND_UP_R	Vstávání z pravého boku	34	3
STAND_UP_L	Vstávání z levého boku	35	3

Tabulka 5: Pohyby v Motion editoru

Pro autonomní chování není využit pohyb plazení a rozcvičky, tyto pohyby byly vloženy jen pro ukázky při řízení pomocí ovladače. Taktéž není naprogramováno vstávání robota při pádu na bok, protože za normálních podmínek k pádu nedojde, a díky postavení čidel není možno pád detekovat.

Po zmáčknutí tlačítka START se robot nastaví do základní polohy 1 (Ready) a jsou na výběr dvě možnosti chování. Při zmáčknutí tlačítka U na základní jednotce nebo při zmáčknutí tlačítka U na ovladači je vybrána možnost řízení pomocí ovladače. Při stisknutí tlačítka D na řídicí jednotce je vybrána možnost autonomního chování robota.

5.1 Řízení pomocí ovladače

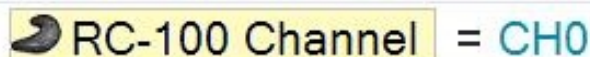
Ovladač s řídicí jednotkou komunikuje bezdrátově přes standard ZigBee. Protože je reálná přenosová rychlost přes tento standard do 50 kb/s, má každé tlačítko ovladače svou

hodnotu. Hodnoty tlačítek jsou voleny tak, aby při jakékoliv kombinaci nebyla možná kolize (viz Tabulku 6).

Kombinace tlačítek	Hodnota vysílaného signálu	Pohyb
U	1	Vpřed
D	2	Vzad
U+4	129	Vpřed a doprava
U+2	33	Vpřed a doleva
D+4	130	Vzad a doprava
D+2	24	Vzad a doleva
6	512	Lehnutí
U+6	513	Rozcvička
D+6	514	Plazení vzad
5	256	Základní pozice 2
L	4	Natočení vlevo
R	8	Natočení vpravo
L+4+5+6	900	Skok do nabídky chování
4+6	516	Vstávání z pravého boku
2+6	520	Vstávání z levého boku

Tabulka 6: Reakce na kombinaci tlačítek ovladače

Program pro řízení pomocí ovladače je velice jednoduchý, nejdříve je zapnut kanál pro přenos mezi ovladačem a ZigBee modulem připojeného do řídicí jednotky (viz Obr. 19).



Obr. 19: Propojení ovladače a řídicí jednotky

Po otevření kanálu a zvolení řízení program čeká na příchozí signál. Při shodě přijímaného signálu s hodnotou některé kombinace tlačítek robot zareaguje. Při řízení jsou všechna čidla aktivní a kontrolují, jestli uživatel neposílá robota záměrně proti překážce. V tomto případě robot zareaguje, vrací se do základní polohy 2 a nedovolí uživateli jakýkoliv pohyb vpřed. To znamená, že robot by při jakémkoli pohybu vpřed neměl narazit do překážky, která je stejně velká jako on sám a je z materiálu, který odráží infračervené světlo a je tedy hlavně pro boční senzory detekovatelným objektem.

Na Obr. 20 je vidět různé řešení pohybů. První pohyb je pohyb doleva-vpřed za podmínky, že ani jeden ze senzorů nedetekuje překážku. Každý port představuje jedno čidlo (porty 5 a 6 IR čidla a port 1 DMS). Pohyb je poté volán nadefinovanou funkcí, která stojí samostatně mimo hlavní blok programu. Druhý pohyb je po stisknutí správné kombinace tlačítek řešen voláním přímo na stránku nachystaného pohybu a třetí pouze natočením vrchního kloubu, tedy pouze pohybem jednoho z motorů do cílové pozice (*Goal position*).

```

LOOP WHILE ( Remocon RXD == U+2 && PORT[5] <= 100 && PORT[6] <= 100 && PORT[1] <= 250 )
{
    CALL FWD_Left
}
LOOP WHILE ( Remocon RXD == D+6 )
{
    Motion Page = 29
    CALL MotionComplete
}
IF ( Remocon RXD == L )
{
    ID[1]: Goal position = 604
    CALL MotionComplete
IF ( Remocon RXD == L+4+5+6 )
{
    JUMP Mode_selection
}
}

```

Obr. 20: Různá řešení pohybů

Z řízení pomocí ovladače lze při stisknutí správné kombinace (viz Tabulku 6) „skočit“ (*JUMP*) v programu zpět do nabídky a zde si vybrat jinou možnost chování.

5.2 Autonomní chování

Při zvolení autonomního chování robot ihned zahájí chůzi vpřed. Při detekování objektu robot dokončí započatý pohyb a začne situaci řešit. Otáčení robota je kvůli velkému poloměru otáčení vždy řešeno jako chůze do boku-vzad a až poté chůze do opačného boku- vpřed. Tímto způsobem je otáčení řešeno hlavně kvůli bočním čidlům, protože robot by stíhal zatočit pouze v případě bílé barvy překážky.

```

FUNCTION Turn_right
{
    LOOP FOR ( i = 1 ~ 3 )
    {
        CALL BWD_Left
    }
    Left = TRUE
}

IF ( Left == TRUE )
{
    LOOP FOR ( i = 1 ~ 3 )
    {
        CALL FWD_Right
        IF ( PORT[6] >= 50 || PORT[5] >= 50 || PORT[1] >= 300 )
        {
            BREAK LOOP
        }
    }
    Left = FALSE
}
}

```

Obr. 21: Otáčení robota vpravo

Protože otáčení proběhne postupně ve funkci a poté v hlavním programu, je třeba nadeklarovat proměnné, které jsou nastavovány podle směru otáčení (viz Obr. 21). Při volání funkce *Turn_right* zahájí robot chůzi doleva-vzad. Díky cyklu *COUNT LOOP*, což je cyklus s daným počtem průběhů (obdoba cyklu *for* v jazyce C), se pohyb provede třikrát. Poté se nastaví proměnná *Left* na hodnotu *TRUE* (pravda) a program se vrátí do hlavního bloku programu. Zde díky nastavení proměnné *Left* na hodnotu *TRUE* se provede druhá část otáčení, tedy chůze doprava-vpřed. Tento pohyb je opět uzavřen do cyklu *COUNT LOOP*, a proto je nutno ošetřit reakci na detekování objektu. Aby robot pohyb doprava-vpřed vykonal nesmí ani jedno z čidel detekovat objekt, v opačném případě je cyklus předčasně ukončen příkazem *BREAK LOOP*. Při přerušení tohoto cyklu program pokračuje v běhu programu. Pokud čidla dále nerozpoznají žádný objekt, zahájí chůzi vpřed, při detekci překážky je volána funkce *Wall*. Obdobně je řešeno otáčení robota vpravo (proměnnou *Right*).

Po volání funkce *Wall* je otestováno, která čidla detekují objekt a která ne. Při detekci jen jednoho z krajních čidel se robot otočí do strany, kde objekt podle aktuálních hodnot není. Při detekci překážky všech čidel zároveň je volána funkce *All*.

```

look = 0
// doprava
ID[1]: Goal position = 420
CALL MotionComplete
Timer = 1.920sec
WAIT WHILE ( Timer > 0 )
// levý, pravý ne
IF ( PORT[5] >= 50 && PORT[6] < 50 )
{
    look = 1
}
// doleva
ID[1]: Goal position = 604
CALL MotionComplete
Timer = 1.920sec
WAIT WHILE ( Timer > 0 )
// levý ne, pravý
IF ( PORT[5] < 50 && PORT[6] >= 50 )
{
    look = 2
}

```

Obr. 22: Rozhodnutí strany (část funkce *All*)

Po započetí funkce *All* se pracuje s časovačem, natáčením jednoho motoru, proměnnými a čidly. Na Obr. 22 je pouze hlavní část funkce. Nejdříve je nadeklarována proměnná *look*, která se bude měnit níže podle hodnot čidel. Nejdříve se provede natočení

kloubu doprava ($ID[1]:Goal\ position = 420$). V této pozici robot čeká necelé dvě sekundy a poté vyhodnotí podmínku. Když pravé čidlo objekt nedetekuje a levé ano, nastaví se hodnota proměnné *look* na 1. Poté se provede natočení kloubu doleva, opět robot čeká a poté vyhodnotí podmínku. Když levé čidlo detekuje objekt a pravé ano, nastaví se hodnota proměnné *look* na 2. Když nebyla splněna ani jedna podmínka zůstává v proměnné stále 0 (viz Obr. 22). Po provedení této části programu se kontroluje na jakou hodnotu je proměnná *look* nastavena. V případě hodnoty 1 se robot začne otáčet směrem doprava, v případě 2 doleva a v případě 0 je volána funkce *Change*. Tato funkce zajistí, že se robot vždy otočí do strany, kde nedetekuje objekt a tedy bude mít místo na otočení (například v rohu místnosti).

V případě volání funkce *Change* se robot rozhoduje do jaké strany se otočí vždy jinak. V programu Task editor není příkaz pro náhodná čísla (*Randomize*), a proto je zajištěno pouze střídání stran vždy, když dojde k volání této funkce. Při prvním volání funkce půjde robot vždy doleva a při každém dalším průběhu se bude střídát otáčení vlevo a vpravo.

Robot po místnosti chodí zcela sám, problém nastává u překážek, které jsou menší než je výška samotného robota, které jsou úzké nebo které mají jinou barvu než bílou. Robot si nepamatuje místa, kterými již prošel, a proto chodí od překážky k překážce, kde se vždy otočí.

Návrat do nabídky chování je možný pomocí stisknutí kombinace tlačítek L,4,5 a 6, stejně jako při řízení pomocí ovladače. Návrat je možný pouze na začátku hlavního cyklu, tzn. že tlačítka je potřeba mít stisknutá poměrně dlouho. Po navrácení do nabídky dojde k vynulování proměnných tak, aby robot při opětovném zvolení autonomního chování nedokončil otáčení, které před skokem nazpět do nabídky započal.

6 ZÁVĚR

Cílem práce bylo navrhnout a oživit čtyřnohého robota a otestovat použitelnost jednotlivých součástí stavebnice BIOLOID a vývojového prostředí RoboPlus v robotickém fotbale RoboCup.

Navržený čtyřnohý robot byl tedy postaven ze stavebnice BIOLOID a postavená konstrukce byla oživena pomocí vývojového prostředí RoboPlus, resp. Task a Motion editoru. Robot má dvě možnosti řízení, a to pomocí ovladače, nebo autonomní chování robota.

Robot při chůzi dopředu nenaráží do překážek, které jsou mu postaveny do cesty. V případě řízení pomocí ovladače se robot při detekci překážky zastaví a nepovolí žádný další pohyb, který by znamenal náraz do překážky, v případě autonomního chování se robot otočí a vyrazí jiným směrem. Robot není schopen detekovat objekt, který je menší než on sám. Je také obtížné detekovat překážky, které nejsou bílé nebo mají malou reflektivitu materiálu. Takové překážky jsou těžce detekovatelné zejména pro boční IR čidla.

Konstrukce robota není dobrá z hlediska pohyblivosti. Vzhledem ke stavbě nohou, kde na každou nohu byly použity pouze dva servo motory, a tím byl umožněn pohyb pouze v jedné ose, je jeho pohyb velmi omezen. Robot potřebuje ke svému otáčení poměrně velký prostor.

Byla použita všechna čidla pro detekci objektu obsažena ve stavebnici BIOLOID Premium kit. Pro robotický fotbal je potřeba přikoupit nebo si vyrobit čidla vlastní, čidla obsažená ve stavebnici jsou zcela nedostatečná. Taktéž vývojové prostředí je pro programování pro robotický fotbal nevyhovující. Programování v něm je zdlouhavé a napsaný kód je dlouhý. Pro programování by bylo vhodnější využít programovacího jazyka C.

SEZNAM POUŽITÉ LITERATURY

- [1] Robotis: Tech Support, [online] 28.01.2010 [cit. 2010-04-25]. Dostupný z WWW:
< <http://support.robotis.com/en/> >
- [2] Sharp: DMS GP2Y0A21YK0F, [online] 09.02.2007 [cit. 2010-04-28] Dostupný z WWW:
< http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf >
- [3] Sharp: Sensor Box Cover – Smart Opto Sensors, [online]. 2007 [cit. 2010-04-28]
< http://www.silicalighting.eu/fileadmin/data_sheets/Sharp/DMS_ProductsBrief.pdf >
- [4] Wikipedie : ZigBee. [online]. 28.04.2010 [cit. 2010-05-01]. Dostupný z WWW:
< <http://cs.wikipedia.org/wiki/ZigBee> >
- [5] Wikipedie : TTL (logika) [online]. 27.04.2010 [cit. 2010-05-01]. Dostupný z WWW:
< [http://cs.wikipedia.org/wiki/TTL_\(logika\)](http://cs.wikipedia.org/wiki/TTL_(logika)) >
- [6] Robotis: User's Manual Dynamixel AX – 12 [online]. 14.06.2006 [cit. 2010-05-02].
Dostupný z WWW: < http://www.megarobot.net/cj/manualy/robotis/biolod/AX-12_aj.pdf >
- [7] Robotis: User's Manual Zig – 100 [online]. 06.07.2006 [cit. 2010-05-03]. Dostupný z
WWW: < http://www.megarobot.net/cj/manualy/robotis/bioloid/Zig100_aj.pdf >
- [8] Robotis: User's manual BIOLOID ver. 1.0 [online]. 13.09.2006 [cit. 2010-05-03].
Dostupný z WWW:
< http://www.megarobot.net/cj/manualy/robotis/bioloid/UsersGuide_aj.pdf >
- [9] Robotis: BIOLOID QuickStart Comprehensive Kit Robot Series ver. 1.0 [online].
13.09.2006 [cit. 2010-05-03]. Dostupný z WWW: < http://www.megarobot.net/cj/manualy/robotis/bioloid/QuickStart_ComprehensiveKit_aj.pdf >
- [10] Robotis: BIOLOID QuickStart Premium Kit Robot Series ver. 1.0 [online]. 18.10.2008
[cit. 2010-05-03]. Dostupný z WWW:
< http://www.megarobot.net/cj/manualy/robotis/bioloid/BIOLOID_Premium_Quickstart_en.pdf >
- [11] Megarobot: Stavebnice BIOLOID [online]. 06.12.2008 [cit. 2010-05-04]. Dostupný z
WWW: < <http://www.megarobot.net/cj/manualy/robotis/bioloid/Stavebnice%20BIOLOID.pdf> >
- [12] AX – 12 [online]. 05.02.2008 [cit. 2010-05-02]. Dostupný z WWW:
< <http://www.antratek.co.uk/images/dynamixel-AX-12.jpg> >
- [13] RoboSavvy: Bioloid Premium [online]. 12.09.2009 [cit. 2010-05-03]. Dostupný z WWW:
< <http://robosavvy.com/Builders/limor/BioloidPremiumBlog/Bioloid%2520Premium%2520Construction%2520Blog.html> >