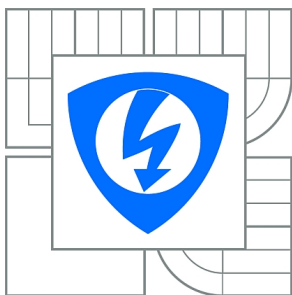


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SKUPINA KRÁČEJÍCÍCH MINIROBOTŮ - SYNCHRONIZACE POHYBŮ

GROUP OF WALKING MINIROBOTS - MOVEMENT SYNCHRONISATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

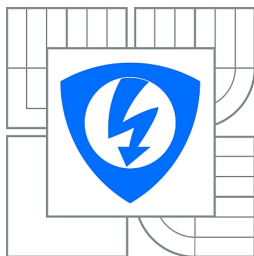
JAKUB MIHÁL

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. LUDĚK ŽALUD, Ph.D.

BRNO 2010



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Jakub Mihál

ID: 106635

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Skupina kráčejších minirobotů - synchronizace pohybů

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte řídicí systém pro miniaturní kráčejší roboty (HiTec Robonova-I, www.hitecrobotics.com). Předpokládá se použití pro prezentační účely (synchronizovaný tanec) či pro robotické hry typu fotbal.

DOPORUČENÁ LITERATURA:

Joseph J. Jones, et. al., Mobile Robots - Inspiration to Implementation, A K Peters, 1999, ISBN 1-56881-097-0

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: doc. Ing. Luděk Žalud, Ph.D.

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Vysoké učení technické v Brně
Fakulta elektrotechniky a komunikačních technologií
Ústav automatizace a měřicí techniky

Skupina kráčejících minirobotů – synchronizace pohybů

Obor: Automatizační a měřicí technika
Student: Jakub Mihál
Vedoucí: Doc. Ing. Luděk Žalud, Ph.D.

Abstrakt:

Úkolem této práce je seznámit se se stavebnicí kráčejícího robota Robonova-I a s diplomovou prací Ing. Tomáše Floriána „Řídicí jednotka pro humanoidní robot“. Stavebnice i diplomová práce byly vyzkoušeny, porovnány a v další části této práce zhodnoceny. Nakonec je vypracováno nové programové řešení řízení robota a bezdrátové komunikace.

Klíčová slova:

Hitec Robotics, Robonova-I, humanoidní robot, řídicí jednotka, mikroprocesor, Freescale, MCF52233, bezdrátové řízení, jednotka XBee-Pro 802.15.4, akcelerometr MMA7260QT

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Control and Instrumentation

Group of walking minirobots - movement synchronisation

Specialisation of study: Automation and Measurement
Student: Jakub Mihál
Supervisor: doc. Ing. Luděk Žalud, Ph.D.

Abstract:

The aim of the work presented here is to get acquainted with the construction of the Robonova-I humanoid robot and with Ing. Florián's dissertation "Control unit for humanoid robot". Both the construction and the dissertation conclusions were tested, compared and, in the next part of the work, evaluated. Finally, a new software solution to the robot control and wireless communication was worked out.

Keywords:

Hitec Robotics, Robonova-I, humanoid robot, control unit, microprocessor, Freescale, MCF52233, wireless control, unit XBee-Pro 802.15.4, accelerometer MMA7260QT

Bibliografická citace díla

MIHÁL, J. *Skupina kráčejících minirobotů - synchronizace pohybů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 49 s. Vedoucí bakalářské práce doc. Ing. Luděk Žalud, Ph.D.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma *Skupina kráčejších minirobotů - synchronizace pohybů* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení §11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení §152 trestního zákona č. 140/1961 Sb.“

V Brně dne:

.....

Jakub Mihál

Poděkování

Děkuji vedoucímu bakalářské práce Doc. Ing. Lud'ku Žaludovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne:

.....

Jakub Mihál

Obsah

1. Úvod	9
1.1 Úkol projektu	9
1.2 Malí kráčeující humanoidní roboti	9
2. Robonova-I	10
2.1 Základní model stavebnice Robonova-I	10
2.1.1 Konstrukce	10
2.1.1.1 Řídící deska (1), (5), (6)	11
2.1.1.2 Servomotory (1), (8)	11
2.1.1.3 Doplnky stavebnice Robonova-I	13
2.1.2 Ovládání a programování (4), (5), (10), (11)	15
2.1.3 Shrnutí	18
2.2 Diplomová práce Ing. Tomáše Floriána (1)	19
2.2.1 Konstrukce	19
2.2.1.1 Mikroprocesor (12)	19
2.2.1.2 Bezdrátová komunikace (13), (14)	20
2.2.1.3 Napájecí část	21
2.2.1.4 Další konektory a příslušenství	21
2.2.2 Programové řešení (1), (8), (15)	21
2.2.3 Popis ovládání robota	24
2.2.4 Shrnutí	25
3. Programové řešení	26
3.1 Komunikace se servopohony	27
3.2 Programování pohybů robota	29
4. Bezdrátová komunikace	31
4.1 Komunikační protokol	31
4.2 Komunikační stavový automat	33
4.2.1 Navázání komunikace	33
4.2.2 Vyhodnocení přijaté zprávy	35

5. Akcelerometr	38
6. Napájecí část	42
7. Závěr	43
8. Seznam použité literatury	44
9. Seznam obrázků	46
10. Seznam tabulek	46
11. Seznam zdrojových textů	47
12. Seznam příloh	47
Přílohy:	48
Příloha A: Redukce programovacího konektoru BDM	48
Příloha B: Tabulka skupin servopohonů	49
Příloha C: Tabulka komunikačního protokolu	50

1. ÚVOD

1.1 ÚKOL PROJEKTU

Hlavním úkolem této bakalářské práce je seznámit se s robotickou stavebnicí Robonova-I a s diplomovou prací Ing. Tomáše Floriána „Řídicí jednotka pro humanoidní robot“ (1). Stavebnice i diplomová práce byly vyzkoušeny, porovnány a v následujících částech této práce zhodnoceny. Původní verze Robonovy-I je popsána v kapitole 2.1 a diplomová práce je popsána v kapitole 2.2.

1.2 MALÍ KRÁČEJÍCÍ HUMANOIDNÍ ROBOTI

Humanoid je obecně bytost (živá či neživá), která se tvarově podobá člověku. Robot, který se velice blíží podobou i chováním člověku je pak nazýván androidem, tuto míru podobnosti nazýváme antropomorfismus (2). Pak v tomto případě z hlediska české mluvnice slovo robot skloňujeme podle vzoru pán (životný rod)(1 str. 10).

Malí kráčejíci humanoidní robot má většinou svoji konstrukci sestavenou ze servomotorů řízených centrálním mikroprocesorem. Nejdůležitějšími vlastnostmi u kráčejících robotů je počet nohou a počet stupňů volnosti na každou nohu. Ten se obvykle pohybuje kolem pěti až sedmi stupni volnosti na jednu nohu. Pokud mluvíme o humanoidním (člověku podobným) robotu, pak robot sestává pouze ze dvou chodidel, které spojuje trup. Na trupu se dále může nacházet další končetiny (ruce) nebo také hlava. U těchto robotů je udržována stabilita spíše na základě dynamické, nežli statické rovnováhy. Tj. pokud by byl náhle zastaven během chůze, je velmi pravděpodobné, že neudrží svoji stabilitu a robot se převrhne. Tato vlastnost je dána počtem končetin (3 str. 118).

2. ROBONOVA-I

2.1 ZÁKLADNÍ MODEL STAVEBNICE ROBONOVA-I

Robonova-I je krácející humanoidní robot vyráběný korejskou firmou Hitec Robotics. Lze ho využít jako výukovou pomůcku, studenti se tak mohou naučit základy robotiky. Zkušenosti se stavbou, ovládáním či programováním ocení nejen studenti, ale i amatéři se zájmem o robotiku (4 str. 3). Robonova-I je dodávána na trh ve dvou variantách - jako rozložený „Kit“ (cena kolem 899,99 USD) nebo sestavený „Ready to Walk“ (cena kolem 1 049,99 USD) (přibližné údaje z prosinec 2009).

2.1.1 Konstrukce

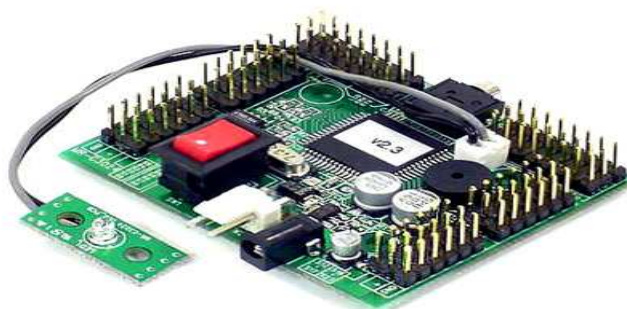
Po mechanické stránce je Robonova-I sestavena z 16-ti digitálních servopohonů Hitec HSR-8498HB s unašeči na obou stranách, spojovaných duralovými díly doplněnými plastovými i kovovými díly trupu. Všechny kovové díly jsou eloxované a zabarvené žlutohnědě. Tyto servopohony jsou řízeny z řídicí desky upevněnou pod pevným plastovým krytem na zadní straně robota. V hlavové části se nachází IR přijímač pro dálkové ovládání a LED dioda pro světelnou signalizaci. Robot je napájen z Ni-MH akumulátorů 6 V/1000 mAh, dobíjených přiloženým nabíječem.



Obrázek 1: Přední a zadní pohled na Robonova-I

2.1.1.1 Řídicí deska (1),(5), (6)

Řídicí jednotkou robota je deska s označením MR-C3024, která byla vyvinutá korejskou společností MiniRobot.



Obrázek 2: Řídicí jednotka

Deska umožňuje připojení až 24 servomotorů, 8 analogových snímačů a 3 konektory slouží pro PWM řízení. Dalšími jsou konektor pro připojení baterie, konektor na nabíječku baterie a pro komunikaci s PC je použito sériové rozhraní RS232, které je vyvedeno na konektor 3,5 mm JACK. Z komunikačních sběrnic je obsažena I2C a sériová linka. Součástí je i piezoelektrický reproduktor pro zvukové signalizace.

Řídicím mikroprocesorem je ATmega 128L od firmy ATMEL (7). Procesor je v 64 pinovém pouzdře TQFP. Paměť je rozdělena na 128 kB Flash, 4 kB EEPROM a 4 kB SRAM. Procesor je 8 bitový s architekturou RISC pracující na taktovací frekvenci 16 MHz. Disponuje 8 kanály 10-ti bitových A/D převodníků, 2x USART, hodiny reálného času RTC, SPI, I2C a 53 dalších univerzálních V/V vývodů.

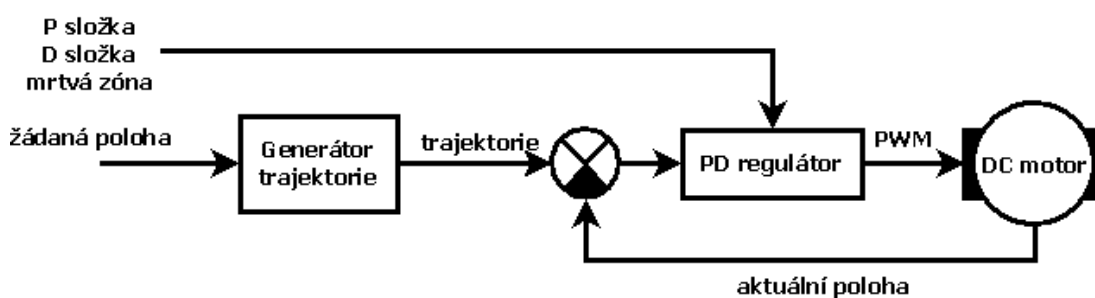
2.1.1.2 Servomotory (1),(8)

Pohyb robota vykonává 16 digitálních servopohonů Hitec HSR-8498HB, který je vyroben ve třech modifikacích pro různé použití na stavebnici Robonova-I



Obrázek 3: Tři varianty tvarů servomotorů HSR-8498HB

HSR-8498HB obsahuje karbonovou převodovku (převodový poměr kolem 300:1), snímačem polohy je zde potenciometr a řídicí elektronikou mikroprocesor ATmega8 firmy ATMEL. Komunikace s procesorem (tedy servopohonem) probíhá pomocí HMI protokolu a řízení motoru je pulzně šířkovou modulací (PWM). Úhlová poloha natočení serva je přenášena na potenciometr, který řídí šířku pulzů generátoru impulzů. Ta se porovná s velikostí žádaného pulzu a elektronika vyhodnocuje tuto odchylku. Servo je řízeno elektronikou pomocí PD (proporcionálně - derivační) regulátoru (viz Obrázek 4), jehož vlastnosti lze programově měnit (P složka, D složka, mrtvá zóna).



Obrázek 4: Řízení DC motoru v servomotoru

Úhlový rozsah natočení serva je 190° (tedy od středové pozice ±95°). Šířka řídicího pulzu odpovídá pak rozsahu 550 až 2450 μs, středová pozice 1500 μs. Výpočet úhlu natočení α v závislosti na šířce pulzu t viz rovnice 1.1.

$$\alpha = \frac{t - 1500}{10} [^\circ; \mu s] \quad (1.1)$$

Pomocí HMI protokolu je možné řídit motor či nastavovat jeho vlastnosti. Komunikace se rozděluje do třech režimů: standardní, rozšířený a sériový. Rozlišení jednotlivých režimů lze pomocí pracovních časových úseků typických pro konkrétní režim.

V normálním režimu (Standard pulse mode) servopohon pracuje s PWM signály v rozsahu 550 až 2450 μ s, na pulzy s šířkou mimo tento rozsah nereaguje. Tento režim slouží pro nastavení vlastností jednotlivých servopohonů.

Pro zjištění polohy slouží tzv. rozšířený režim (Extended pulse mode). V tomto režimu je přerušeno nastavování polohy, snížena rychlost a točivý moment, hrozí tak nebezpečí ztráty stability robota. Tento režim dovoluje volný pohyb servomotoru rukou (nastavení požadované pozice robota) a následné zjištění hodnot při této pozici. Tento režim je používán při programování s názvem „Catch and Play“ (viz manuál k Robonova-I dodávaný ke stavebnici nebo na (9)).

Posledním režimem je sériový režim (Serial mode). Jak již název napovídá, je možné komunikovat se servopohony pomocí sériové linky. Každému servopohonu je přiřazeno identifikační číslo (ID), které nabývá hodnot od 0 do 127 (maximální počet serv je tedy 128). Je však nutné všechny inicializovat např. pomocí programu HMI Servo Programmer. Sériovému režimu je dále věnována kapitola 2.2.2. Více o programování servomotorů pomocí HMI protokolu viz (1 str. 16).

2.1.1.3 Doplnky stavebnice Robonova-I

Výhodou stavebnice Robonova-I je možnost rozšíření. Tyto další části mohou rozšířit vlastnosti samotného robota o řadu nových funkcí (dálkové ovládání či přidání senzorů) nebo i zvýšit samotné pohybové možnosti (zvýšení počtu serv – např. pro pohyb trupu nebo noh ve vertikální ose).

Např. firma Hitec Robotics (výrobce Robonovy-I) nabízí ve svém sortimentu různé následující rozšíření:

- Řídicí desky (Robot controller) MR-C3000E, MR-C3024
- Servopohony (Robot servo) HSR-5498SG, HSR-5990TG, HSR-8498HB
- IR senzor vzdálenosti (Distance sensor)
- Světelný senzor (Light sensor)

- Dotykový senzor (Touch sensor)
- Zvukový senzor (Sound sensor)
- Gyroskopický senzor (Robot gyro HG-R01)
- Snímač náklonu (Tilt sensor HRS-2XA01)
- Mechanické ruce (Grippers) (viz Obrázek 5)
- FM přijímač (HFS-R01 UART FM 27MHz receiver)
- Dálkové ovládání (Zebra 4FM 27MHz Set)



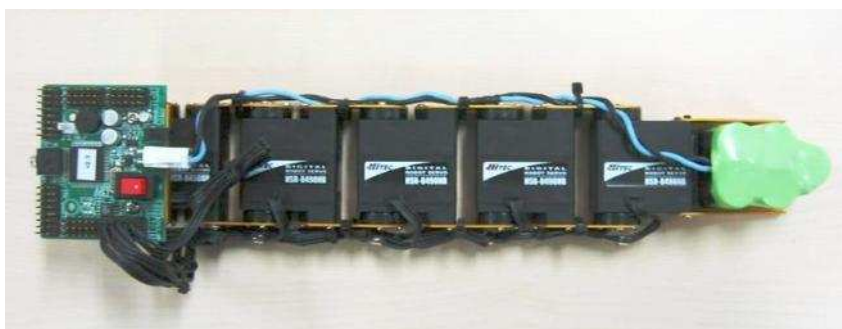
Obrázek 5: Robonova-I s instalovanými mechanickými rukami

Další rozšíření od jiných prodejců mohou být např.: bluetooth modul, bluetooth kontroler, ultrazvukový sonar, různé akcelerometry, bezdrátové Wi-fi moduly a další. Tyto rozšíření nabízí např. dodavatelé:

- Active Robots (<http://www.active-robots.com/>)
- německý dodavatel Hitec Robotics (<http://www.robonova.de/>)
- Trossen Robotics (<http://www.trossenrobotics.com>)

Je zde samozřejmě i možnost přidání vlastního výběru rozšíření (po předcházejícím přizpůsobení).

Robonova-I je zajímavá také tím, že není nutné ji sestavit podle návodu jako humanoidního robota. Fantazii se meze nekladou a tak je možné sestavit ze stavebnice např. tučňáka, pavouka, hada (červa) či jen robotickou paži.



Obrázek 6: Robonova-I sestavená jako robotický červ

2.1.2 Ovládání a programování (4), (5), (10),(11)

Robota lze ovládat pomocí infračerveného dálkového ovládače (součástí základní verze robota), jehož tlačítka spouštějí určité sekvence pohybů uložené v paměti robota. Tyto sekvence (programy) se do robota nahrávají přes sériový port RS-232 z počítače PC. Programy lze spouštět také přímo z PC namísto použití dálkového ovládače. Pro psaní programů slouží speciálně vytvořený jazyk RoboBASIC (založený na programovacím jazyku BASIC), tak jednoduchý, aby ho zvládl i začínající programátor. Lze najít na stránkách <http://www.robobasic.com/>.

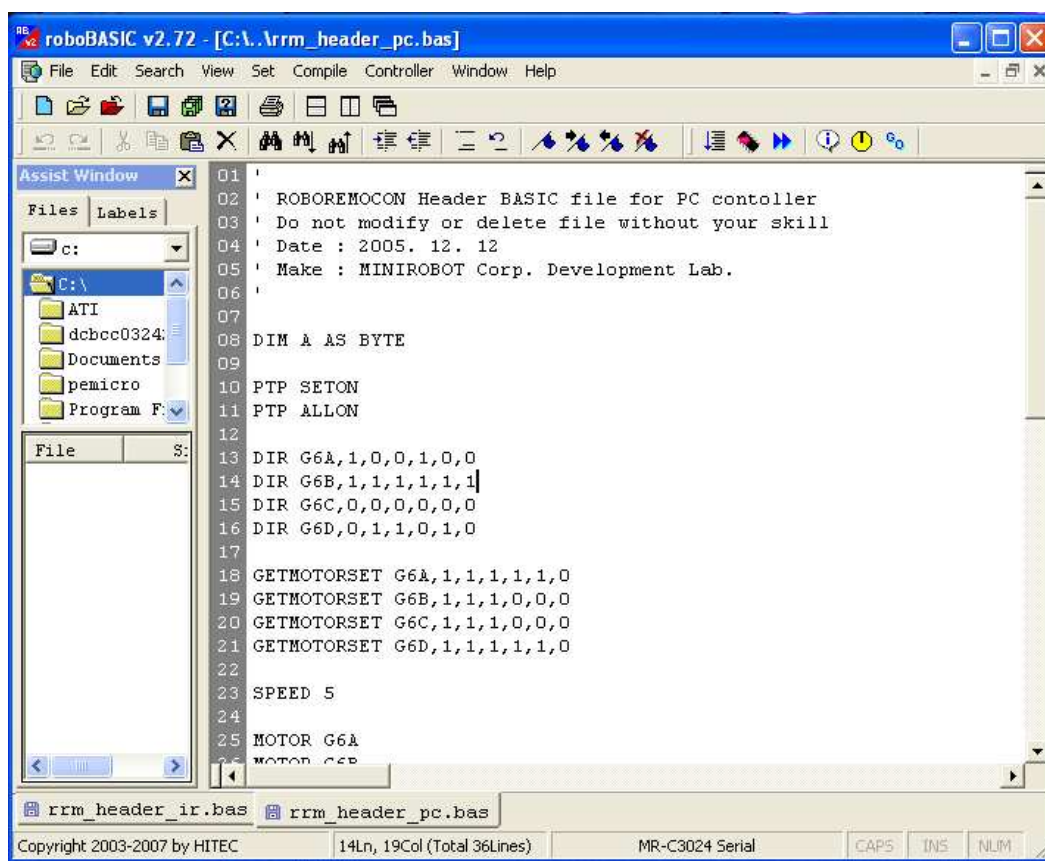
Programování servomotorů lze provést jednotlivě či ve skupinách po 6-ti (G6A – G6D), 8-mi (G8A – G8C) či 24-ti (G24) servomotech. Za příkazem MOVE následuje název programované skupiny a pak jednotlivá čísla udávající natočení servopohonů, oddělená čárkou. Při vypuštění čísla je pozice ponechána v aktuální pozici a nemění se. Čísla reprezentují úhel natočení v rozsahu 10° až 190° (středová poloha je 100°).

Nastavení rychlosti servomotoru se provádí pomocí příkazu SPEED za kterým následuje hodnota odpovídající rychlosti pohybu. Rozmezí rychlostí je od 1 do 15. Běžné nastavení rychlosti je 3 (fyzikální rozměr jednotek nebyl zjištěn).

Mezi dalšími skupinami příkazů jsou příkazy pro paměť, pro ovládání LCD displeje, relační a matematické operátory, příkazy pro piezoelektrický reproduktor,

pro externí komunikaci přes RS-232, pro analogové signály a další. Kompletní příručku pro tento programovací jazyk lze nalézt na stránkách výrobce (9).

Program „roboBASIC“ má implementovanou funkci „Catch and Play“ („Uchop a hraj“). V tomto režimu, po uvolnění kroutícího momentu v jednotlivých servech či rovnou skupinách, lze libovolně pohybovat servy a nastavit požadovanou polohu. Nastavené hodnoty se přečtou a je možné je vložit do programu. Tímto způsobem lze programovat velice jednoduše různé sekvence pohybů.



Obrázek 7: Ukázka programu RoboBasic v2.72

Pro ovládání robota existují i další vývojová prostředí, založené na programovacím jazyku RoboBASIC. Od výrobce Robonovy-I je dodáván např. „roboScript“, který umožňuje bez znalostí programování vytvářet sekvence pohybů serv. Nastavením pozice posuvníků se současně pohybují odpovídající serva. Tyto kroky pohybů se poté uloží a sekvence kroků přiřadí programem „roboRemocon“

určitému tlačítku na dálkovém ovládání. Počet možných naprogramovaných sekvencí odpovídá počtu tlačítek na dálkovém ovládání, kterých je celkem 32.

Na internetu můžeme sehnat další programy i se simulací naprogramovaného pohybu. Výběrem např.:

- RZ1Action (<http://web.mac.com/micono/RZE/RZ1Action.html>)
- SimROBOT 0 for Robonova-I English Version 1.0
(http://www.robonova.com/simrobot/english_download.php)
- MECH Puppeteer v1.01b by Bauer
- (http://bauerindependents.com/SUBMAIN/Robotics_Puppeteer.htm) – ke stažení na (9), sekce download, software



Obrázek 8: Simulační program RZ1Action

2.1.3 Shrnutí

Již se základní verzí lze vytvořit zajímavé sekvence pohybů. Mezi nejzajímavější patří např.: holubice, kliky (i na jedné ruce), hvězda, kung-fu tance nebo přemety (vzad i vpřed). Tyto úkoly jsou napevno dané sekvence programu, které se volají např. při stisku tlačítka na IR ovladači operátorem. Použitím příslušenství lze rozšířit vlastnosti a např. naprogramovat jednoduché autonomní reakce robota nebo třeba i (při použití mechanických rukou) lezení po lanku.

Jako omezení se však ukazuje malý počet připojitelných příslušenství a rychlost procesoru. Také je zde nedostatečná podpora programování jazykem RoboBASIC. Začátečníkovi je velice užitečný, přesto při řešení rozsáhlejších úkolů je již nedostatečný. Můžeme tedy říci, že omezení tohoto jazyka způsobuje i omezení výběru příslušenství.

2.2 DIPLOMOVÁ PRÁCE ING. TOMÁŠE FLORIÁNA (1)

Diplomová práce Ing. Tomáše Floriána se zabývá především návrhem řídicí jednotky pro humanoidního kráčejícího robota Robonova-I. Dále byl navržen jednoduchý komunikační protokol a realizováno softwarové řešení řízení robota pomocí počítače přes bezdrátovou komunikaci.

2.2.1 Konstrukce

Z konstrukčního hlediska se změnila řídicí jednotka, která je zrealizována na oboustranné desce plošných spojů o rozměrech 100 x 60 mm. Přímo na desce plošných spojů se nachází mikroprocesor MCF52233, modul Xbee-Pro, akcelerometr, napájecí část, sběrnice I2C, 3x UART, konektor pro ethernet a další konektory. Na hlavě robota je instalován ultrazvukový snímač vzdálenosti a osvětlovací LED.

2.2.1.1 Mikroprocesor (12)

Řídicí deska obsahuje mikroprocesor MCF52233, který patří do rodiny Coldfire procesorů RISC. Tento 32 bitový procesor je založen na verzi V2 Coldfire procesorových jader pracujících na frekvenci 60 MHz, poskytuje vysoký výkon (57 MIPS) a nízkou výkonovou spotřebu (napájení 3,3 V). Paměť je rozdělena na statickou RAM (SRAM) o velikosti 32 kB a paměť typu Flash o velikosti 256 kB. Procesor obsahuje velké množství komunikačních rozhraní i podpůrných modulů.

Jsou to především a mimo jiné:

- hardwarový kryptografický akcelerátor (CAU)
- 8 kanálový 12 bitový A/D převodník (ADC)
- 4 kanálový řadič DMA (přímý přístup k paměti)
- 3x sériový port UART s DMA (přímý přístup k paměti)
- rozhraní QSPI, I2C, CAN
- 2x programovatelný časovač přerušení
- 8x časovač (PWM/DMA)
- 63 univerzálních v/v
- 10/100 Ethernet řadič s PHY

- PLL, SW Watchdog
- BDM / JTAG port

2.2.1.2 Bezdrátová komunikace (13), (14)

Pro bezdrátovou komunikaci je použit modul XBee-Pro 802.15.4 OEM RF,



**Obrázek 9: Modul
XBee 802.15.4
OEM RF**

který je upevněn v patici 2 x 10 pinů. Tento modul komunikuje na základě mezinárodního standardu pro bezdrátovou komunikaci ZigBee (IEEE 802.15.4). Komunikuje ve frekvenčním pásmu 2,4 GHz rychlostí až 250 kbps s dosahem až 750 m ve venkovních prostorech a 60 m ve vnitřních prostorech (platí pro mezinárodní variantu). Anténa

je keramická (čip), výkonový odběr jen 10 mW (mezinárodní varianta) při napájecím napětí 3,3 V. Modul podporuje tzv. spánkové režimy prodlužující životnost baterie. Další vlastnosti lze nalézt v dokumentaci „90000982_B.pdf“ na (14). Modul naprogramoval Ing. Hynčica (zdroj: Ing. Florián).

Na jiném bezdrátovém kanálu komunikuje miniaturní kamera osazená na hlavové části robota. Miniaturní kamera je připojena k řídicí desce přes LC filtr (filtrace rušení ze servomotorů) za účelem napájení z akumulátoru. Popsaná kamera byla zakoupena v obchodě GM Elecktonic jako zboží číslo 755-110 (viz <http://www.gme.cz/cz/index.php?product=755-110>). Jak je popsáno v (1 str. 43), kamera je od neznámého výrobce a i přes certifikát od společnosti TÜV, kamera ruší přenos modulu XBee.



Obrázek 10: Bezdrátová miniaturní kamera a přijímač

2.2.1.3 Napájecí část

Napájení celého robota je možné z původního Ni-MH akumulátoru 6 V/1000 mAh nebo z externího zdroje 6 V připojeného na napájecí konektor. Napájecí obvod je rozdělen pomocí dvou stabilizátorů na dvě napájecí úrovně: 3,3 V a 5 V. Napájení 5 V je využito pouze pro sběrnici I2C a pro ultrazvukový dálkoměr SRF10. Servopohony a bezdrátová miniaturní kamera (s LC filtrem) jsou napájeny přímo z baterie (popř. externího zdroje).

2.2.1.4 Další konektory a příslušenství

Na desce se nacházejí další konektory pro připojení různých periférií. Je zde programovací konektor BDM, který byl zjednodušen na 2 x 5 pinů. Dále konektory pro napájení bezdrátové miniaturní kamery a osvětlovacích LED diod na hlavě robota, řízených pomocí PWM z mikroprocesoru. Nechybí konektor sběrnice I2C pro připojení ultrazvukového senzoru vzdálenosti a pro budoucí využití je zde konektor RJ45 pro připojení k ethernetu. Pro komunikaci se servomotory slouží 4 konektory, které jsou připojeny na mikroprocesor (UART0) pomocí integrovaného obvodu MAX3232 (převodník mezi TTL a RS232 logikou). Pro připojení na A/D převodník slouží 5 konektorů.

Nevyužitým je prozatím akcelerometr Freescale MMA7260QT. Tento tříosý akcelerometr je připojen na tři 8 bitové analogově-digitální porty procesoru.

2.2.2 Programové řešení (1), (8), (15)

Komunikace mezi servomotory a mikroprocesorem probíhá na základě HMI protokolu v sériovém režimu (Serial mode). V tomto režimu jsou všechna serva zapojena na jedné komunikační lince a jsou jim přiřazena identifikační čísla (ID). Pomocí ID lze nastavovat pozici, rychlost a číst pozici serva. Nelze však číst rychlost, a některá nastavení se týkají všech serv na lince (např. P složka, D složka, mrtvá zóna nebo aktivace/deaktivace). Komunikace pro RS-232 má následující nastavení:

- rychlost 19 200 baud
- datové bity 8 bit

- stop bit 2 bit
- parita bez parity
- handshake nevyužit

Velikost příkazu je 7 Bytů, přičemž poslední dva jsou při odesílání nulové a slouží pro návratové hodnoty. Schéma příkazu viz Tabulka 1.

	0	1	2	3	4	5	6
Mcu:	0x80	Command	Param1 (High)	Param2 (Low)	Checksum	0x00	0x00
Servo:	High Z					Param1 (High)	Param2 (Low)

Tabulka 1: Schéma komunikace HMI protokolu v sériovém režimu

Byte kontrolního součtu (checksum) je určen pro kontrolu na straně servomotoru. Výpočet této hodnoty viz rovnice 1.2.

$$B_4 = checksum = 256 - \left(\left(\sum_{n=0}^3 B_n \right) \bmod 256 \right) \quad (1.2)$$

Podrobnější popis komunikace a použitých příkazů je v (1) a na (8).

Pro komunikaci mezi mikroprocesorem a počítačem slouží modul bezdrátového přenosu XBee 802.15.4, který je připojen na sběrnici UART2 mikroprocesoru. Komunikační nastavení je následující:

- rychlost 38 400 baud
- datové bity 8 bit
- stop bit 1 bit
- parita bez parity
- handshake nevyužit

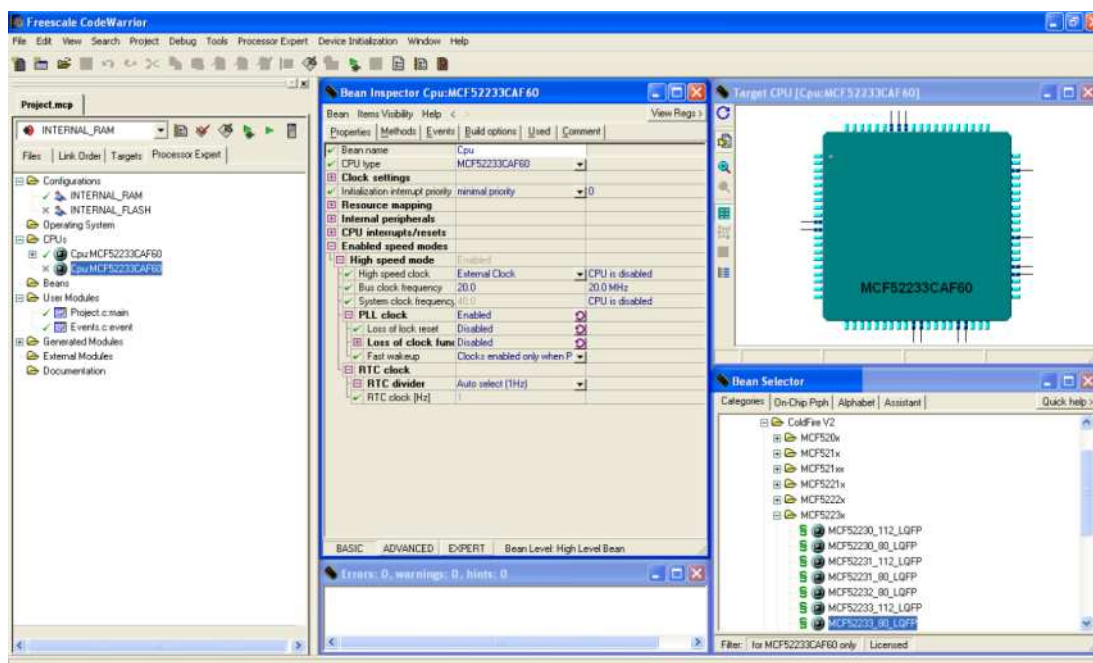
Pro tuto komunikaci byl navržen specifický komunikační protokol. Tento protokol používá nekodované jednotlivé znaky či skupinu znaků (řetězce), které jsou ukončené ASCII kódem 13 (CR) a jejichž velikost nepřesahuje 10 Bytů.

Programování mikroprocesoru probíhá připojením programátoru USB Coldfire Multilink USB-ML-CFE přes redukci na zjednodušený programovací konektor BDM 5x2 pinů na řídicí desce robota. Tento programátor je jednoduchý programovací a ladící nástroj pro řadu procesorů V2/V3/V4 ColdFire. Rozhraní USB zrychluje přenos dat a usnadňuje univerzální použití na různých počítačích. Multilink kontroluje mikrokontrolér prostřednictvím BDM (Backround Debug Mode). Podpora pro tento vývojový nástroj je integrována přímo do programu CodeWarrior, spolupracuje s P&E debuggerem a programátorem flash, případně produktů třetích stran. Cílová napětí jsou od 1,8 V do 5,5 V. Napájení je z USB sběrnice - žádné samostatné napájení tedy není potřeba. Umožňuje synchronní i asynchronní operace. Bližší popis na (16) nebo (17). Redukce viz Příloha A: Redukce programovacího konektoru BDM.

Programovacím nástrojem procesoru v robotovi je vývojový nástroj CodeWarrior Development Studio. Tento program je integrovanou sadou vývojových nástrojů pro architektury ColdFire v7.0. Nástroje CodeWarrior obsahují ANSI C/C++ kompilátor a knihovny speciálně navržené pro zvýšení výkonu a redukci velikosti kódu embedded aplikací na mikroprocesorech a mikrokontrolérech ColdFire. Kompilátor by neměl činit problémy ani uživatelům, kteří se s ním setkávají poprvé a to i díky svému grafickému uživatelskému rozhraní pro Windows.

Inicializační nástroj zařízení vývojového studia umožňuje rychle a snadno konfigurovat on-chip registry a generovat inicializační kód. Kód samotný pak může být přidán přímo do projektu nebo uložen jako samostatný textový soubor.

Další součástí vývojového studia je software Processor Expert, který umožňuje rychle vytvářet návrhy aplikací a pomocí grafického rozhraní definovat potřebnou funkčnost pro svou aplikaci, přičemž Processor Expert se postará o generaci otestovaného a optimalizovaného kódu pro aplikaci a zvolené ColdFire zařízení.



**Obrázek 11: Integrovaný nástroj Processor Expert vývojového studia
CodeWarrior IDE**

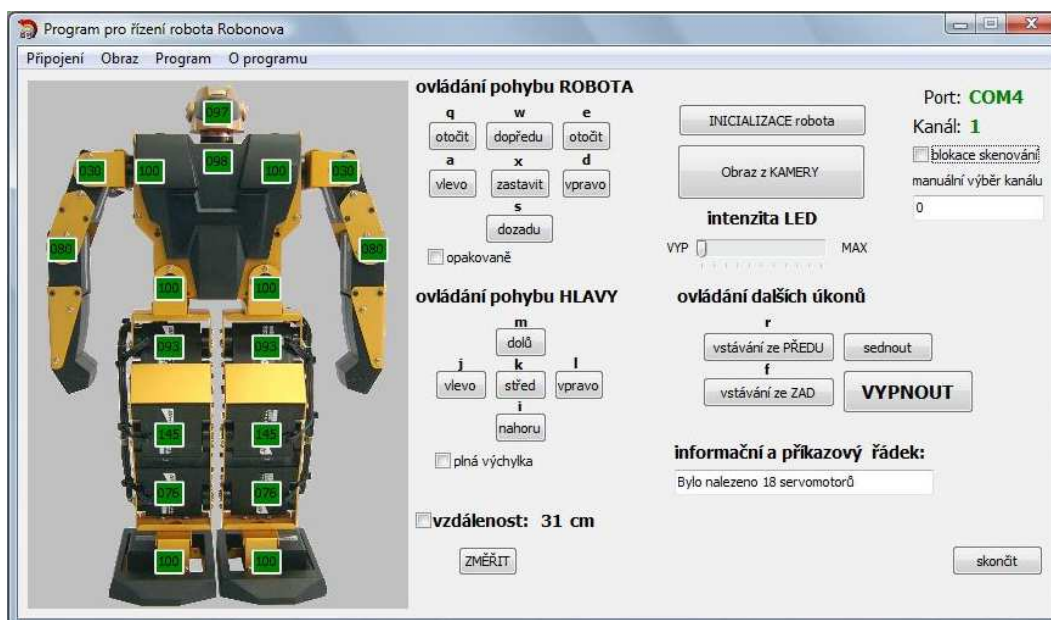
CodeWarrior Development Studio disponuje i příjemným průvodcem tvorby projektů – tzv. Project Wizard, který vývojářům napomůže rychleji vytvořit fungující projekt, a to už i jen několika kliknutími myši.

Blíže viz (17).

2.2.3 Popis ovládání robota

Ovládací program robota byl vytvořen ve vývojovém prostředí C++Builder 2009. Ten slouží jako jednoduchá vizualizace polohy servomotorů (nastavená čísla úhlu natočení), ovládání robota a také z důvodu informovat operátora o aktuálním stavu. Komunikace probíhá bezdrátově přes navržený nekódovaný jednoduchý protokol. Pomocí tohoto protokolu lze robotovi přikazovat předem nadefinované sekvence příkazů jako pohyb vpřed, vzad, vlevo a vpravo. Dále pohyb hlavy (dva servopohony) nahoru, dolů, vlevo, vpravo. Je možné také samostatně ovládat servopohony. Lze ovládat i intenzitu osvětlení LED diod. Protokol umožňuje přikázat robotu dřep či vztyk z hrudi nebo ze zad. Nakonec také posílá data z robota do počítače jako např. počet nalezených servomotorů a naměřená vzdálenost ultrazvukovým senzorem. Program umožňuje také zobrazit obraz z miniaturní

kamery, který není přenášen společně s ostatními daty, ale po vlastním bezdrátovém přenosu. Bližší popis viz (1).



Obrázek 12: Ovládací program modifikované Robonovy-I

2.2.4 Shrnutí

Výsledné řešení diplomové práce byl logický krok při vylepšování vlastností samotného robota. Byla navržena řídicí deska s výkonným procesorem převyšující výkon předchozího. To s sebou přináší možné rozšíření periférií a lepší programovací přístup. Samotné programování procesoru není zcela jednoduché (vyžaduje znalosti z programování procesorů Freescale pomocí C, znalost periférií a protokolu HMI i navrženého protokolu) avšak dává mnohem větší kontrolu nad procesy a dává i prostor pro další tvůrčí činnost. Ovládání robota z počítače je vykonané velice jednoduše s velmi malým rozsahem činností, avšak pro jednoduché úkony zcela dostačuje.

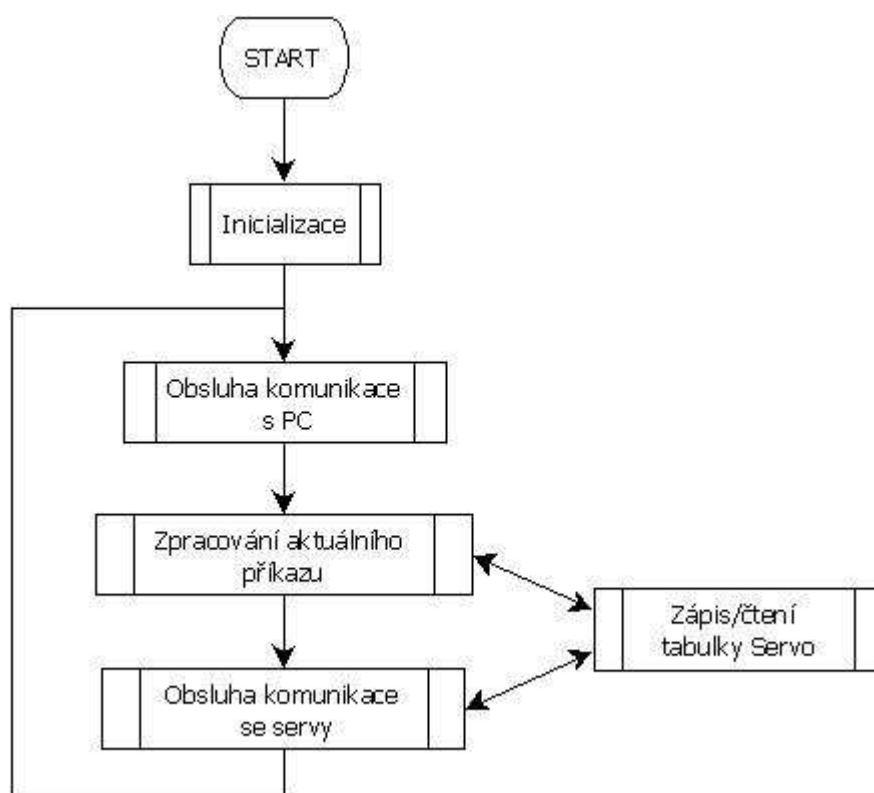
Problémy spojené s touto prací a návrh jejich řešení, případně další připomínky jsou zpracovány v následující kapitole.

3. PROGRAMOVÉ ŘEŠENÍ

Původní ovládací program procesoru byl kompletně přepracován. Pro robota tohoto typu je důležitá rychlá časová odezva a rozdělení programu do samostatných na sobě nezávislých vrstev (z důvodu bezkonfliktního přístupu ke zdrojům mcu).

Program pracuje (po předem dané inicializaci) v nekonečné smyčce (podobně jako v PLC), kde se postupně zpracovávají jednotlivé vrstvy:

- obsluha komunikace s operátorem
- zpracování aktuálního příkazu
- obsluha komunikace se servy



Obrázek 13: Zjednodušený průběh programu v MCU

V hlavním programu je rozeznán přijatý příkaz od PC a provedena daná obsluha. Program musí být neustále průchodný, pracuje v neustálém cyklu. Proto obsahuje spoustu přepínačů (switch – case) na základě kterých se rozhoduje o momentálním stavu programu. Po vykonání daného stavu a případném ověření

splnění podmínek se program přesouvá do následujícího stavu (další case). Pokud je nutné čekat na splnění podmínek, program nečeká (např. v rámci místní smyčky), ale povolí tok programu dál, podmínka bude znovu testována v dalším cyklu.

Po dokončení obsluhy přijatého příkazu se načte další příkaz a znovu se začne zpracovávat.

3.1 KOMUNIKACE SE SERVOPOHONY

Každé servo je reprezentováno strukturou obsahující hodnotu nastavené polohy serva, přečtené polohy, rychlost a vlastní byte pro nastavované flagy. Tyto struktury tvoří tedy tabulku, do které se zapisuje (čte) v průběhu běhu programu požadované hodnoty a nastavují flagy. Samotná komunikace se servy je vytvořena v oddělené vrstvě, kde se zpracovávají hodnoty v této tabulce serv na základě nastavených bitů v bytu Flag.

ID	poloha	ReadPoloha	rychlost	Flags
0	100	98	5	00000011b
1	30	31	5	00000011b
2	80	79	5	00000011b
...	...	...	...	...
max_serv	0	0	6	00000100b

Tabulka 2: Příklad nastavených hodnot v tabulce serv

Příklad nastavené části tabulky serv lze vidět v Tabulka 2. V prvním sloupci je požadovaná poloha serva, v druhém poslední čtená poloha a třetím rychlost serva. Poslední sloupec obsahuje flagy, jejichž význam je vyjádřen v Tabulka 3.

Poslední řádek tabulky (ID zvoleno jako „max_serv“) reprezentuje hodnoty nastavované pro všechny serva na sběrnici. Využito je pouze nastavení společné rychlosti všem servům. Komunikační protokol HMI však umožňuje také nastavit stejnou polohu servopohonům na sběrnici (příkaz 0xE6). Více viz (8).

Název	Hodnota	Význam požadavku
Flag_poloha	0x01 (0. bit)	Nastavení polohy serva
Flag_ReadPoloha	0x02 (1. bit)	Čtení polohy serva
Flag_rych	0x04 (2. bit)	Nastavení rychlosti serva

Tabulka 3: Význam flagů pro jednotlivá serva

Přístup do tabulky serv je z hlavního programu možný pouze z určitých funkcí (obdobu funkcí public z C++, pro soukromé funkce použito static volatile):

- move – zapíše sekvenci polohy pro všechna serva a vrátí 1 po nastavení všech serv cílové pozice (jinak 0)
- speed – zapíše rychlost daného serva
- speed_all – nastaví danou rychlost všem servům na sběrnici
- wait – pokud servo není v požadované poloze vrátí 1 (jinak 0)
- wait_all – stejné jako předchozí, ale pro všechny serva na sběrnici
- request – nastaví požadavek čtení polohy serva
- request_all – nastaví požadavek čtení poloh pro všechny serva na sběrnici
- getPosition – vrátí nastavenou polohu v tabulce serva
- getSpeed – vrátí nastavenou rychlost v tabulce serva
- getReadPosition – vrátí přečtenou polohu serva, pokud je aktuálně nastavený požadavek na čtení polohy, vrátí 0
- isServo – vrátí 1, pokud bylo servo s daným ID zjištěno na sběrnici při inicializaci (jinak 0)

Funkce jsou ošetřeny na případné chybové hodnoty. Inicializaci serv na sériové sběrnici (vyhledání a zapsání aktuální pozic serva do tabulky) je provedeno zavoláním funkce „InitServa()“. Pomocí funkce „SendServa()“ je provedena kompletní komunikace se servy na základě hodnot v tabulce.

Souhrn funkcí a proměnných pro ovládání servopohonů je umístěn ve zdrojovém souboru „Serva.c“.

3.2 PROGRAMOVÁNÍ POHYBŮ ROBOTA

Programování pohybových sestav robota je provedeno v souladu s původními skupiny serv G6. Polohu robota v daném časovém okamžiku je možné vyjádřit jednorozměrným polem 24 hodnot poloh servopohonů v úhlových stupních. Pole lze logicky rozdělit do menších skupin po 6-ti prvcích.

Indexace prvků pole však neodpovídá skutečné identifikaci (ID) jednotlivých servopohonů. Např. pátá hodnota pole (index 4) reprezentuje servopohon s ID číslo 11. ID servopohonu lze změnit pouze v zapojení normálního režimu (Standard pulse mode), kdy je možné měnit vlastnosti jediného serva na sběrnici. Z důvodu nesouhlasné indexace bylo vytvořeno transformační pole pro korektní reprezentaci indexů pole hodnot polohy.

Hodnoty některých servopohonů také nejsou skutečnými polohami. Na robota jsou servopohony umístěny s různým směrem osy otáčení. Aby bylo programování přehlednější, jsou polohy servopohonů přepočteny tak, aby se hodnoty jevíly symetrické podle osy souměrnosti robota. V jazyce RoboBASIC je tato funkce řešena pomocí příkazu DIR. Evivalentem je zde funkce „prepocet()“, která zahrnuje i menší korekce poloh serva (proměnná „korekce“).

Popis skupin servopohonů, indexace, změna směru a korekce viz Příloha B: Tabulka skupin servopohonů.

Příkladem programování pohybů robota je jednoduchá sekvence „hand_shaking()“, viz Zdrojový text 1. Hodnoty poloh serv jsou převzaty ze vzorového podprogramu „Advanced Overall Template Program(Version 1.00 20070307).bas“ pro RoboBASIC ze stejnojmenné funkce.

Stav podprogramu je dán ve statické proměnné „pozice“. V prvním kroku pozice (case 0) se nastaví všem servům rychlost 10 a pomocí „move()“ se zapíše požadované polohy do tabulky serv. Funkci „move()“ jsou předávány pozice jednotlivých skupin servopohonů G6A až G6D. Při hodnotě 0 se poloha skupiny (či jednotlivého serva) nemění. Uvnitř funkce je kontrolováno dosažení této polohy funkcí „wait_all()“. Pokud je dosaženo polohy funkce vrátí 1 a tím inkrementuje proměnnou „pozice“.

```
#include "Serva.h"
static byte pozice = 0;
byte hand_shaking(void)
{
    const byte prikaz1[6] = {190,100,100,0,0,0};
    const byte prikaz2[6] = {190, 45, 45,0,0,0};

    switch(pozice)
    {
        case 0:          speed_all(10); pozice += move(0,prikaz1,prikaz1,0); break;

        case 1:
        case 3:
        case 5:          speed_all(14); pozice += move(0,prikaz1,prikaz2,0); break;
        case 2:
        case 4:
        case 6:          pozice += move(0,prikaz2,prikaz1,0); break;

        case 7:          pozice += standard_pose(9); break;
        default:         pozice = 0; return 1;
    }
    return 0;
}
```

Zdrojový text 1: Příklad sekvence pohybů funkce „hand_shaking()“

Postupně jsou tak provedeny sekvence pohybů a kontrolováno dosažení pozice. Posledním příkazem je volání funkce „standard_pose()“, která uvede robota do výchozí polohy a znovu kontroluje dosažení pozice. Po splnění celé sekvence je nutné vynulovat proměnnou „pozice“ a funkce vrátí 1.

Podobným způsobem je tedy možné robota naprogramovat stejné pozice jako pro původní program vytvořený v jazyce RoboBASIC. Navíc je pro programování použit jazyk ANSI C pro lepší kontrolu nad programem.

Předdefinované pohyby robota jsou uloženy ve zdrojovém souboru „pozy.c“

4. BEZDRÁTOVÁ KOMUNIKACE

4.1 KOMUNIKAČNÍ PROTOKOL

Pro komunikaci mezi robotem a operátorskou stanicí (PC) byl navržen a vytvořen komunikační protokol. Protokol je upravenou verzí předhozího z diplomové práce (1). Všechny zprávy musí být zakončeny znakem CR (Carriage return, ASCII 0x0d) a celková délka (včetně znaku CR) nesmí přesáhnout 12 bytů.

Příkazy pro pohyb soustav serv jsou rozdělena na pohyb hlavy a těla. Pro rozpoznání příkazu pro pohyb hlavy je nutná prefixová znaková hodnota 'H' (z anglického Head) a pro pohyb těla prefix 'B' (z anglického Body). Za tímto znakem následuje daný příkaz. Např. „BW“ značí příkaz pro pohyb těla (tedy robota) vpřed (tj. krok vpřed) atd. atpod.

Servopohony lze také ovládat samostatně. Pro nastavení úhlu polohy serva je příkaz s prefixem 'P' (Position). Za tímto příkazem následuje dvouciferné číslo identifikace (ID) servopohonu (rozsah od 0 do 99), oddělovací znak '.' (tečka) a tříciferné číslo polohy. Při požadavku o čtení polohy servopohonu je zaslán příkaz s nulovou hodnotou polohy.

Nastavení rychlosti konkrétního serva je příkazem s prefixem 'S' (Speed). Zbytek příkazu je stejný jako pro nastavení polohy. Pro čtení nastavené rychlosti serva je zaslán příkaz s nulovou hodnotou rychlosti.

Návratový tvar těchto dvou příkazů je totožný s příkazem nastavení dané hodnoty polohy či rychlosti serva.

Měření vzdálenosti ultrazvukovým senzorem je spuštěno zasláním příkazu 'V'. Robot po dokončení měření odpoví zprávou ve tvaru „V123“, kde číslo za znakem 'V' reprezentuje změřenou vzdálenost v cm.

Intenzita osvětlení LED diod lze měnit pomocí PWM v rozsahu od 0 do 255. Za prefixovou hodnotou příkazu „L“ (Light) následuje tříciferná decimální hodnota intenzity osvětlení.

Celková inicializace robota do počátečního stavu zařídí odeslání příkazu 'I'. Pokud robot přijme tento příkaz, provede se funkce „Init()“, kde jsou inicializovány servopohony na sběrnici (identifikace a zavedení do tabulky serv), robot je nastaven

do standardní pozice („standard_pose()“) a poté je zjišťována poloha pomocí akcelerometru a vykonán vztyk z břicha („StandUpBricho()“), či ze zad („StandUpZada()“).

Celkový stav robota lze zjišťovat zasláním požadavku ‘R’ (jako robot). Odpověď robota je ve stejném tvaru, kde za znakem ‘R’ následují dva hexadecimální znaky stavu robota. Stav robota je poté indikován pomocí flagů nastavujících jednotlivé bity, viz Tabulka 4.

Název	Hodnota	Význam
RobotZap	0x01 (0.bit)	Zapnutý moment v servopohonech (jinak uvolnění)
RobotStoji	0x02 (1.bit)	Robot stojí (jinak sedí)
RobotBricho	0x04 (2.bit)	Robot leží na břiše
RobotZada	0x08 (4.bit)	Robot leží na zádech

Tabulka 4: Význam flagů pro stav robota

Např. pokud robot odpoví zprávou „R05“, pak je robot zapnut (moment v servopohonech) a leží na břiše (nastaveny bity 0x01 a 0x04).

Příkazy popsané výše jsou odesílány s počátečním znakem STX, adresou a na konci s kontrolním součtem, popis viz 4.2.2 Vyhodnocení přijaté zprávy.

Souhrn příkazů popsaných výše viz Příloha C: Tabulka komunikačního protokolu.

4.2 KOMUNIKAČNÍ STAVOVÝ AUTOMAT

Vyhodnocování přijatého znaku od PC (přes UART2) zajišťují dva na sobě nezávislé stavové automaty. Rozhodování o aktuálním stavu je zajištěno pomocí příkazů switch-case. Při přijetí jakéhokoli znaku se zavolá přerušení, kde se nacházejí stavové automaty. Ty vyhodnotí aktuální stav a na základě toho se zavolá příslušná funkce, která vyhodnotí přijatý znak a nastaví následující stav. Stavové automaty jsou tedy synchronní (až na jednu výjimku) s přijetím znaku od PC (přerušení).

4.2.1 Navázání komunikace

Menší ze stavových automatů pouze zajišťuje vyhodnocení synchronizačního příkazu (ekvivalent příkazu PING). Příkaz se skládá z hlavičky netisknutelného znaku SYN (ASCII 0x16) a znaku adresy vysílací stanice. Adresa PC je defaultně nastavena na '0'. Pro robota je vyhrazena pevná adresa v rozsahu od '1' do '9'. Příkaz tedy obecně umožňuje zavolat potenciálně až 9 robotů s různou adresou.

Příklad formátu příkazu počítače lze vidět v Tabulka 5.

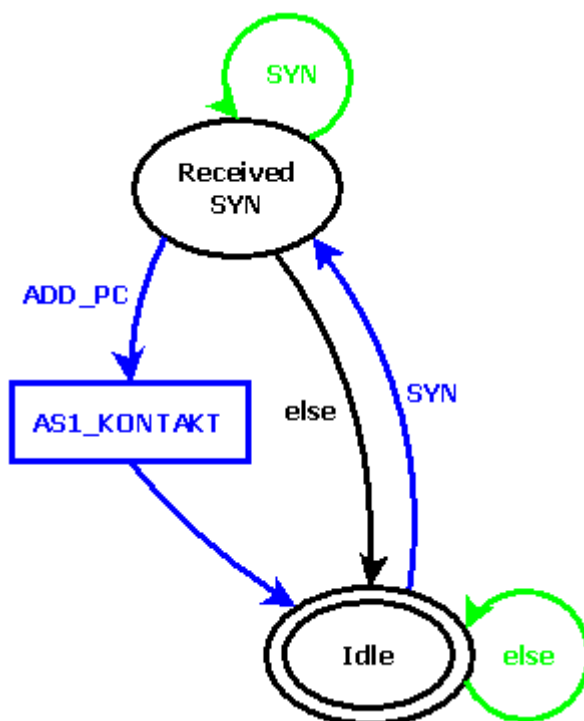
	B ₀	B ₁
Význam	SYN	adresa
Hodnota	0x16	'0'

Tabulka 5: Formát synchronizačního příkazu počítače

Ztráta signálu je v mikroprocesoru vyřešena pomocí vnitřních hodin reálného času. Pokud je přijat synchronizační příkaz od počítače, je nastaven flag o navázané komunikaci. Na konci obsluhy přerušení je testováno nastavení tohoto flagu a vynulovány hodiny reálného času. Pokud je v hlavním cyklu programu detekováno překročení tohoto času o 30 sekund, je to považováno za ztrátu signálu. Spustí se tedy cyklické vysílání synchronizačního příkazu a zároveň se procházejí frekvenční kanály modulu XBee. Hledání kanálu je ukončeno při přijetí odpovědi od PC.

Je tedy nutné, aby se PC pravidelně ozývalo, než uběhne překročení nastaveného časového limitu.

Vyhodnocení přijatého synchronizačního příkazu stavovým automatem lze vidět na Obrázek 14.



Obrázek 14: Stavový diagram automatu vyhodnocujícího přijetí synchronizační zprávy

4.2.2 Vyhodnocení přijaté zprávy

Rozpoznání přijaté validní zprávy zajišťuje větší ze stavových automatů. V tomto případě je nutné rozeznat začátek zprávy, adresu přijímací stanice (robota), přenášená data a konec zprávy. Pro začátek a konec zprávy jsou zvoleny standardní netisknutelné znaky STX (ASCII 0x02) a CR (ASCII 0x0D). Před samotnými daty se nachází adresa stanice, pro kterou je tato zpráva určena. Adresy jsou přiděleny tak, jak bylo popsáno v předchozí kapitole 4.2.1.

Adresa a samotná data nesmí obsahovat žádné vyhrazené systémové znaky. Proto všechny přenášené číselné hodnoty musí být převedeny na jejich znakové ekvivalenty (včetně hexadecimálních či decimálních čísel). Před ukončujícím znakem CR jsou vyhrazeny dva byty pro znakově zakódovaný kontrolní součet (Checksum) pro kontrolu validity přenášené zprávy.

Struktura zprávy je zobrazena v Tabulka 6.

B_0	B_1	$B_2 \dots B_{(n-3)}$	$B_{(n-2)}$	$B_{(n-1)}$	B_n
STX	adresa	Data	Hi CS	Lo CS	CR

Tabulka 6: Formát přenášené zprávy bezdrátovou komunikací (n max 12)

Výpočet kontrolního součtu CS (Checksum) je založen na funkci modulo (zbytek po dělení), kdy sčítáme samotná data zprávy. Výpočet viz rovnice 1.3.

$$CS = 256 - \left[\left(\sum_{i=2}^{n-3} B_i \right) \bmod 256 \right] \quad (1.3)$$

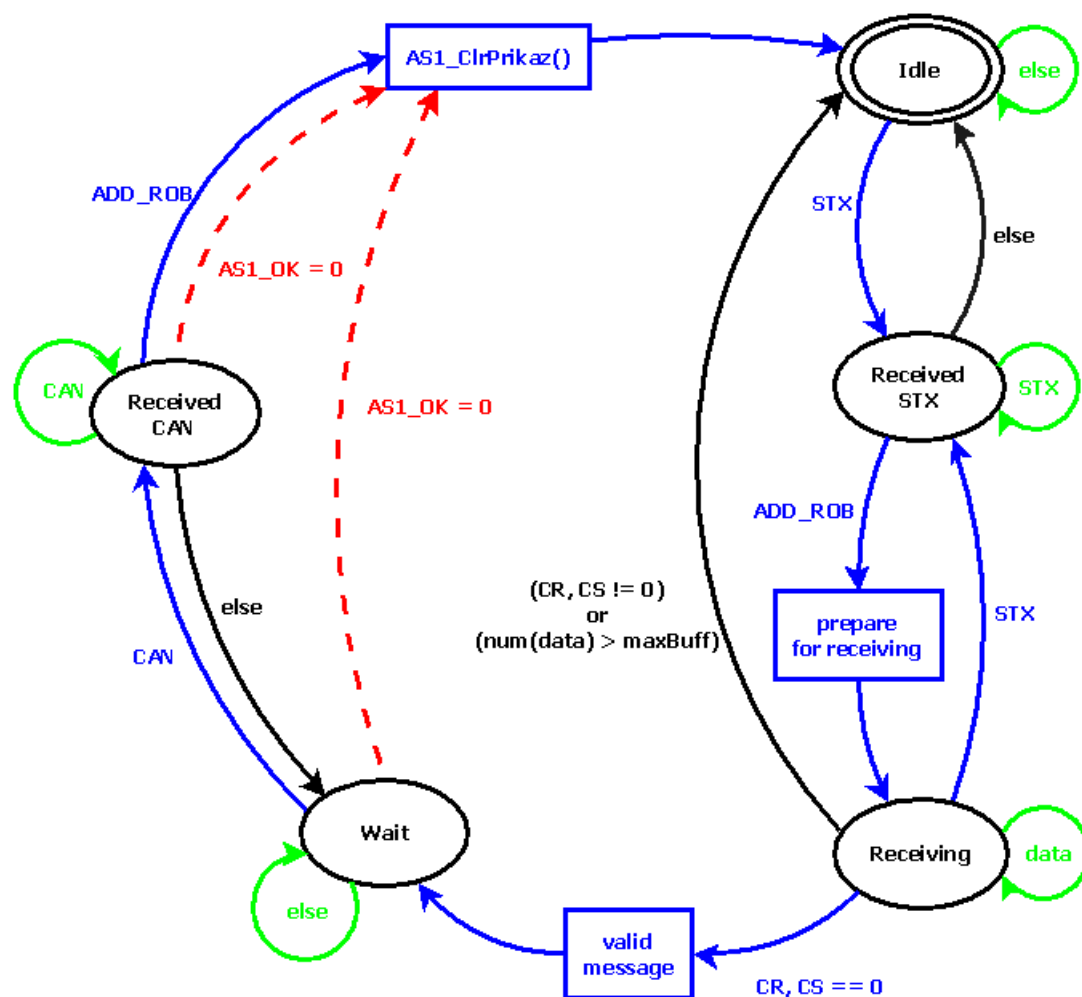
Výsledek CS je rozdělen do dvou bytů a v hexadecimální podobě převeden do znaků. První je přenášen vyšší byte, následovaný nižším bytem.

Při kontrole správně přenesené zprávy musí platit rovnice 1.4.

$$\left(\sum_{i=2}^{n-1} B_i \right) \bmod 256 = 0 \quad (1.4)$$

Větší ze stavových automatů má celkem 5 stavů: Počátečním stavem je stav „Idle“, který čeká na přijetí startovního bytu STX (ASCII 0x02), poté přejde do stavu „Received STX“. Pokud je přijata správná adresa robota, stav se přesune do

„Receiving“. Jinak se vrací do stavu „Idle“, tj. zpráva není adresována tomuto robotu. Ve stavu „Receiving“ se zapisují přijatá data do zásobníku, dokud není detekován poslední znak CR (ASCII 0x0D). V případě dosažení počtu dat maximální velikosti Bufferu, je vyhodnocena tato zpráva jako neplatná. Zásobník má dostatečnou velikost pro jakoukoli zprávu odpovídající komunikačnímu protokolu popsanou v 4.1.



Obrázek 15: Stavový diagram automatu vyhodnocující validní zprávy

Po přijetí konce zprávy je vyhodnocena validita zprávy pomocí kontrolního součtu a pokud je zpráva v pořádku, zkopíruje se do jiného zásobníku a nastaví flag o přijaté validní zprávě. Automat se přesune do stavu „Wait“.

V tomto stavu automat čeká na zpracování aktuálního příkazu. Pokud přijme znak od PC, tak vyšle odpověď o zaneprázdněnosti XOFF (ASCII 0x13). Automat se vrací po zpracování aktuální zprávy do stavu „Idle“. Tento přechod je nastaven asynchronně z hlavního cyklu programu.

Ve stavu „Wait“ je možné také stornovat aktuální příkaz pomocí zprávy CAN (ASCII 0x18) následovaným adresou přijímacího robota. Pokud detekovaná adresa není shodná s adresou robota, stav se vrací do „Wait“.

Při ukončení zpracování aktuálního příkazu či jeho stornování se provede vynulování flagu o přijaté validní zprávě, tj. aktuální příkaz již pozbude platnosti.

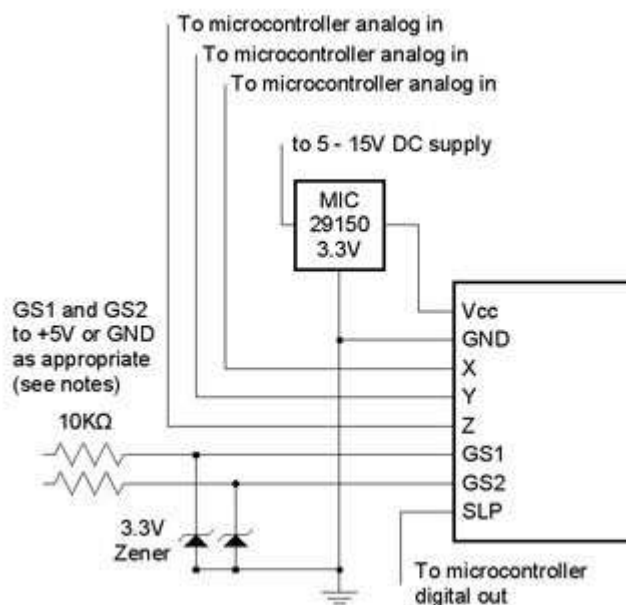
Stavové schéma tohoto automatu je na Obrázek 15.

5. AKCELEROMETR

Senzor MMA7260QT je levný kapacitní 3D MEMS akcelerometr s integrovaným zpracováním signálů, jednopólovým filtrem typu dolní propust, teplotní kompenzací a externí volbou měřicího rozsahu a citlivosti. Každý senzor má již z výroby nastavenou nulovou hladinu a zlomový kmitočet filtru a tedy nejsou zde vyžadovány žádné externí součástky. Nízkou spotřebu v běžném provozu podtrhuje nepatrná spotřeba ve sleep módu. Akcelerometr je tedy vhodný i pro bateriově napájená zařízení. Blíže viz (18).

Mezi jeho základní vlastnosti patří:

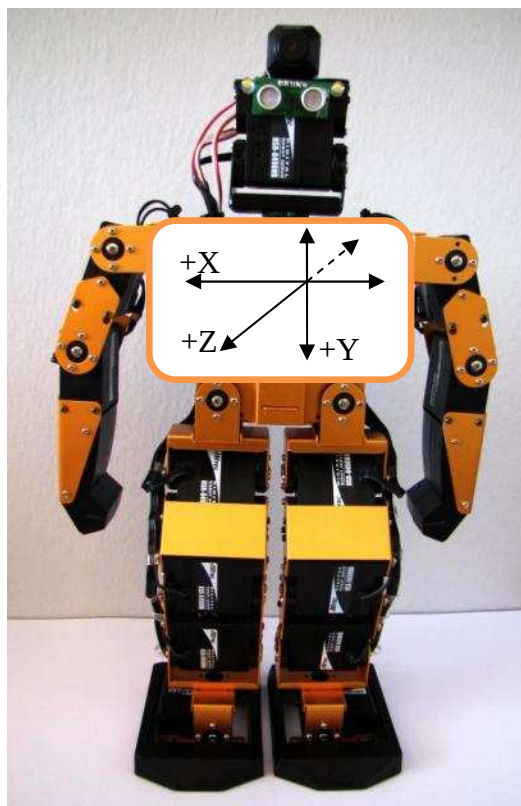
- Měřicí rozsah: 1.5 g / 2 g / 4 g / 6 g volitelný 2 vstupy g-Select
- Maximální citlivost: 800 mV/g v rozsahu 1.5g
- Nelinearita: +/- 1 % z rozsahu
- Křížová citlivost: 5 %
- Spotřeba: 500 μ A, v sleep módu jen 3 μ A
- Napájecí napětí: 2.2 V až 3.6 V
- Pracovní teplota: -40 až 105 °C
- Pouzdro: QFN 6 x 6 x 1.45 mm



Obrázek 16: Typické připojení akcelerometru MAA7260 k mikroprocesoru

3 analogové výstupy poskytující napěťový signál úměrný působícímu zrychlení v ose X, Y a Z jsou již připojeny na 3 kanály A/D převodníku MCU. Vstupy g-Select, kterými se volí jeden ze čtyř možných měřících rozsahů, jsou vyvedeny pouze na prokovy. Nastavení těchto vstupů by muselo být dáno pevným připojením přes ochranné rezistory a zenerovy diody na napájecí napětí (logická 1) nebo zem (logická 0). Viz Obrázek 16.

Umístění integrovaného obvodu na řídicí desce je takové, že kladná osa X směřuje k pravé ruce, kladná osa Y dolů a kladná osa Z směrem vpřed. Směr těchto os z je vyobrazena na Obrázek 17.



Obrázek 17: Směr os akcelerometru instalovaného na robotovi

Pro nulové dynamické zrychlení je na výstupu senzoru v konkrétním směru osy nulová hodnota 1,65 V (polovina napájecího napětí 3,3 V). Záporné směry zrychlení snižují tuto hodnotu a kladné naopak zvyšují. Citlivost k_g lze nastavit pomocí digitálních vstupů g-Select. Při natočení akcelerometru se výstupní napětí chová podle rovnice 1.5.

$$V_{OUT} = (k_g * \sin \alpha * 1 g) + 1,65 V \quad (1.5)$$

Table 3. g-Select Pin Descriptions

g-Select2	g-Select1	g-Range	Sensitivity
0	0	1.5g	800 mV/g
0	1	2g	600 mV/g
1	0	4g	300 mV/g
1	1	6g	200 mV/g

Obrázek 18: Nastavení rozsahu a citlivosti akcelerometru

Akcelerometr je využit v robotovi při citlivosti 800 mV/g pro detekci pádu vpřed či vzad. Je důležité zaklonit v těchto případech hlavu a tím ochránit kameru před poškozením. Při pádu vpřed se zakloní hlava dozadu, při pádu vzad dopředu.

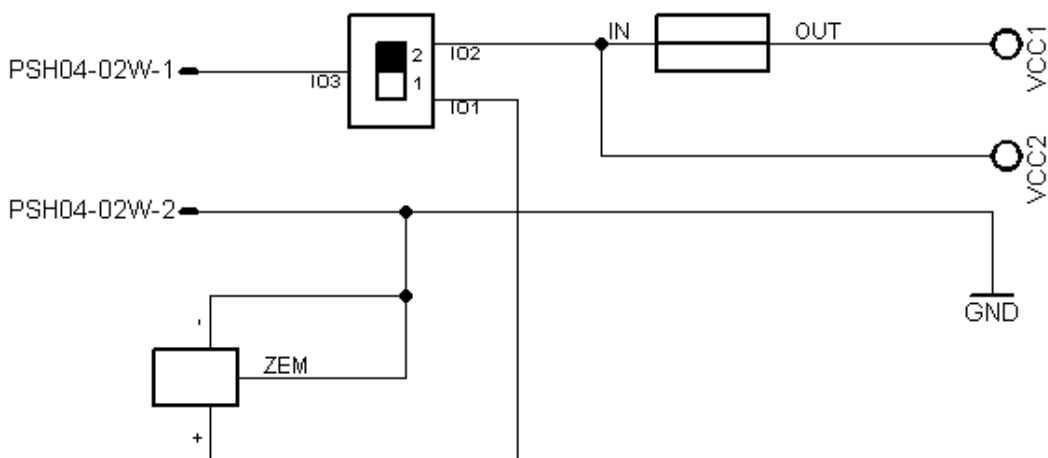
Pokud napětí z akcelerometru překročí určitou hranici, A/D převodník vyvolá přerušení a zde se nastaví flag o kritické situaci a směru pádu. Pokud je zjištěn v hlavním programu nastavení tohoto flagu, přeskočí se zpracovávání aktuálního příkazu a obnoví se tabulka serv ze zálohy. Zároveň se nastaví požadovaná poloha hlavy. Operátor je informován o aktuálním stavu robota a vyčkává se na další příkaz.

V případě, že operátor zašle příkaz o inicializaci robota, ten zjistí v jaké poloze se nachází (pomocí akcelerometru) a provede danou sekvenci pohybů. Robot tak automaticky zajistí vztyk na nohy z dané pozice.

6. NAPÁJECÍ ČÁST

Rídící deska robota obsahuje napájecí konektor 2,5/5,5 mm. V diplomové práci Ing. Floriána sloužil tento konektor jako alternativní napájení robota (např. z externího zdroje). Přepínání napájecího zdroje mezi akumulátorem a napájecím konektorem bylo zajištěno dvoupólovým přepínačem.

Aby bylo možné nabíjet akumulátor, aniž by bylo nutné ho vyjmout z robota, byly prohozeny dva kontakty na přepínači. Přepínač tak slouží k přepínání akumulátoru mezi nabíječkou (připojenou na napájecí konektor) a napájení řídicí desky robota. Pokud není připojena nabíječka, slouží tento přepínač jako vypínač. Schéma viz Obrázek 19.



Obrázek 19: Schéma zapojení vstupu napájecí části řídicí desky

Konektor pro akumulátor PSH02-02WG (rozteč 2,54 mm) byl vyměněn za PSH04-02W (rozteč 3,96 mm). Původním záměrem bylo také vyměnit napájecí zdířku 2,5/5,5 mm za velikost 1,3/3,5 mm, jejíž velikost odpovídá nabíječce ze stavebnice Robonova-1. Ze záměru však bylo upuštěno, z důvodu nízké dostupnosti tohoto konektoru.

Nakonec byl vyměněn také akumulátor za skupinu pěti akumulátorových baterií GP 130AFHR s následujícími parametry: typ NiMH, 1,2 V / 1300mAh, velikost 2/3AF, viz (19).

7. ZÁVĚR

Humanoidní robot Robonova-I umožňuje různé využití ve výuce robotiky. Je možné připojit různé příslušenství a rozšířit tak jeho funkce či vlastnosti. O oblíbenosti tohoto robota mluví častá účast na různých robotických soutěžích (robotický fotbal apod.) a také oblíbenost u amatérů zabývajících se robotikou je bezesporná. Robot má však také své nedostatky: Největší problém je zřejmě nedostatečný programovací nástroj RoboBASIC, který velice omezuje použití při pokročilem programování. Výkon mikroprocesoru se zdá být také neuspokojující.

Tyto nedostatky byly částečně odstraněny v diplomové práci (1): Jako programovací nástroj je použit časem ověřený jazyk C a výkon procesoru je jistě postačující. Řídicí deska je navržena velice dobře a obsahuje i nevyužité periferie (akcelerátor, různé konektory, ethernet). V diplomové práci jsou zrealizovány ovládací programy bezdrátovou komunikací z počítače. Negativní stránkou je však složitější programování a nepřehlednost kódu mikroprocesoru. Problémy spojené s miniaturní kamerou jsou již méně podstatné, ale řešitelné.

V této bakalářské práci byl ovládací program mikroprocesoru zcela přepracován tak, aby byla co nejrychlejší časová odezva. Program je také rozdělen do jednotlivých funkčních vrstev obsluhujících zvláště bezdrátovou komunikaci, řízení toku hlavního programu a komunikaci se servopohony. Celý program byl navržen tak, aby nebylo problémem programovat již vyřešené sekvence pohybů vypracovaných pro robota Robonova-1 z původního jazyka RoboBasic.

Problémem při učení pohybů je řešení inverzní úlohy kinematiky, kdy jsou programovány přímo polohy kloubů robota. Lze programovat rychlost a natočení jednotlivých kloubů. Tento způsob učení je však velice náročný a výsledky pohybu nejsou vždy uspokojující.

Pro komunikaci s počítačem (PC) byl navržen komunikační protokol. Předpokládá se, že by bylo možné ovládat několik stejných robotů najednou, takže bylo nutné zavést adresaci robotů i operátorské stanice. Protokol je navržen pouze ke komunikaci mezi PC a robotem, nikoli mezi roboty navzájem.

8. SEZNAM POUŽITÉ LITERATURY

1. **FLORIÁN, T.** *Řídící jednotka pro humanoidní robot*. Brno : Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, 2009. str. 82. Vedoucí diplomové práce Doc. Ing. Luděk Žalud, Ph.D..
2. Humanoid. *Wikipedia: the free encyclopedia*. [Online] 5. 8 2010. [Citace: 22. 5 2010.] <<http://en.wikipedia.org/wiki/Humanoid>>.
3. **ŠOLC, František a ŽALUD, Luděk.** *Robotika*. Brno : Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, 1. 11. 2006. str. 144. Skripta do předmětu Základy robotiky.
4. **HANZAL, Josef.** Roboti už jdou, už tady jsou. *ROBOT REVUE - magazín ze světa robotiky*. měsíčník, září 2009, Sv. 1, START, str. 3.
5. **STAROSTA, Michal, et al.** *Humanoidní robot Robonova-I*. Brno : Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, 23. 4. 2009. str. 19. Semestrální projekt do předmětu BROB. Vedoucí semestrálního projektu Doc. Ing. Luděk Žalud, Ph.D..
6. HitecRobonova-I. *Robosavvy.com*. [Online] 20. 10 2009. [Citace: 20. 12 2009.] <http://robosavvy.com/site/index.php?option=com_openwiki&id=wiki:robonova-i_faq>.
7. ATmega128(L). *Atmel.com*. [Online] 2467S-AVR-07/09, 07 2009. [Citace: 20. 12 2009.] Datasheet.
<http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf>.
8. **IBBOTSON, Richard.** Robotics Files. [Online] 24. 7 2006. [Citace: 20. 12 2009.] <<http://web.ukonline.co.uk/r.ibbotson/files.htm>>.
9. *Hitec robotics*. [Online] [Citace: 20. 12 2009.] <<http://www.robonova.com/>>.
10. **BEZSTAROSTI, Jiří.** Roboti už jdou, už tady jsou: Robonova-I - oživení. *ROBOT REVUE - magazín ze světa robotiky*. měsíčník, listopad 2009, Sv. 1, 2, stránky 8-9.
11. —. Roboti už jdou, už tady jsou: Robonova-I - sestavení. *ROBOT REVUE - magazín ze světa robotiky*. měsíčník, říjen 2009, Sv. 1, 1, stránky 11-13.

12. MCF5223X: Integrated ColdFire V2 Ethernet Microcontrollers. *Freescale semiconductor*. [Online] [Citace: 20. 12 2009.]
<http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MCF5223X&fsrc=1>.
13. **HYNČICA, Ondřej**. Bezdrátové sítě typu mesh. *odbornecasopisy.cz*. [Online] [Citace: 20. 12 2009.] Online publikovaný článek z časopisu Automa, číslo 12 ročník 2005. <http://www.odbornecasopisy.cz/index.php?id_document=30826>.
14. XBee® & XBee-PRO® 802.15.4 OEM RF Modules. *digi.com*. [Online] [Citace: 20. 12 2009.] <<http://www.digi.com/products/wireless/point-multipoint/xbec-series1-module.jsp>>.
15. **BUCKLEY, David**. Robonova. *davidbuckley.net*. [Online] 18. 11 2008. [Citace: 20. 12 2009.] <<http://davidbuckley.net/DB/RoboNova/RoboNova.htm>>.
16. USB-ML-CFE. *P&E Microcomputer Systems*. [Online] [Citace: 27. 12 2009.] <http://www.pemicro.com/products/product_view.cfm?product_ID=163>.
17. **VOJÁČEK, Antonín**. Processor Expert - snadné nastavení MCU a periférií jen klikáním myši. *hw.cz*. [Online] 25. Březen 2008. [Citace: 20. 12 2009.] <<http://hw.cz/teorie-praxe/programovani/art2269-processor-expert-snadne-nastaveni-mcu-periferii-jen-klikanim-mysi>>.
18. MMA7260QT: 3-Axis Acceleration Sensor. *Freescale semiconductor*. [Online] [Citace: 20. 12 2009.] <http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMA7260QT>.
19. *GP batteries*. [Online] [Citace: 10. 5 2010.] Datasheet. <<http://www.gpbatteries.com/pic/GP130AFHR.pdf>>.

9. SEZNAM OBRÁZKŮ

<i>Obrázek 1: Přední a zadní pohled na Robonova-I</i>	10
<i>Obrázek 2: Řídicí jednotka</i>	11
<i>Obrázek 3: Tři varianty tvarů servomotorů HSR-8498HB</i>	12
<i>Obrázek 4: Řízení DC motoru v servomotoru</i>	12
<i>Obrázek 5: Robonova-I s instalovanými mechanickými rukami</i>	14
<i>Obrázek 6: Robonova-I sestavená jako robotický červ</i>	15
<i>Obrázek 7: Ukázka programu RoboBasic v2.72</i>	16
<i>Obrázek 8: Simulační program RZ1Action</i>	17
<i>Obrázek 9: Modul XBee 802.15.4 OEM RF</i>	20
<i>Obrázek 10: Bezdrátová miniaturní kamera a přijímač</i>	20
<i>Obrázek 11: Integrovaný nástroj Processor Expert vývojového studia CodeWarrior IDE</i>	24
<i>Obrázek 12: Ovládací program modifikované Robonovy-I</i>	25
<i>Obrázek 13: Zjednodušený průběh programu v MCU</i>	26
<i>Obrázek 14: Stavový diagram automatu vyhodnocujícího přijetí synchronizační zprávy</i>	34
<i>Obrázek 15: Stavový diagram automatu vyhodnocující validní zprávy</i>	36
<i>Obrázek 16: Typické připojení akcelerometru MAA7260 k mikroprocesoru</i>	39
<i>Obrázek 17: Směr os akcelerometru instalovaného na robotovi</i>	40
<i>Obrázek 18: Nastavení rozsahu a citlivosti akcelerometru</i>	41
<i>Obrázek 19: Schéma zapojení vstupu napájecí části řídicí desky</i>	42

10. SEZNAM TABULEK

<i>Tabulka 1: Schéma komunikace HMI protokolu v sériovém režimu</i>	22
<i>Tabulka 2: Příklad nastavených hodnot v tabulce serv</i>	27
<i>Tabulka 3: Význam flagů pro jednotlivá serva</i>	28
<i>Tabulka 4: Význam flagů pro stav robota</i>	32
<i>Tabulka 5: Formát synchronizačního příkazu počítače</i>	33
<i>Tabulka 6: Formát přenášené zprávy bezdrátovou komunikací (n max 12)</i>	35

11. SEZNAM ZDROJOVÝCH TEXTŮ

Zdrojový text 1: Příklad sekvence pohybů funkce „hand_shaking()“..... 30

12. SEZNAM PŘÍLOH

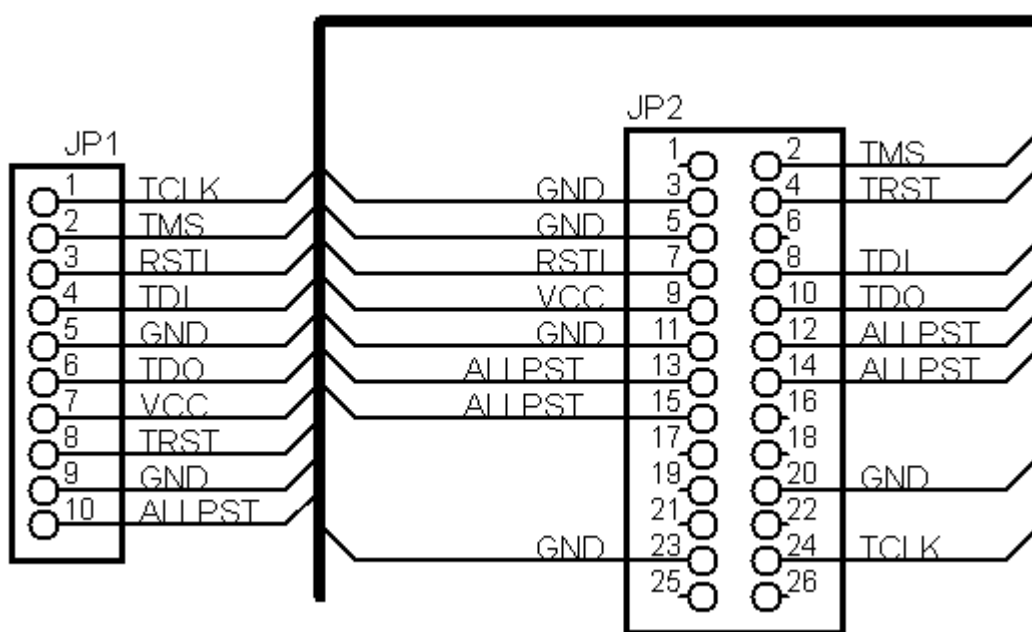
Příloha A: Redukce programovacího konektoru BDM..... 48

Příloha B: Tabulka skupin servopohonů 49

Příloha C: Tabulka komunikačního protokolu..... 50

PŘÍLOHY:

PŘÍLOHA A: REDUKCE PROGRAMOVACÍHO KONEKTORU BDM



PŘÍLOHA B: TABULKA SKUPIN SERVOPOHONŮ

index	ID serva	skupina G6	popis	změna směru	standard_pose	korekce
0	15	G6A	levá noha	ne	100	0
1	14			ano	76	-2
2	13			ano	145	0
3	12			ne	93	0
4	11			ano	100	2
5			nevyužito	ne	100	0
6	3	G6B	levá ruka	ne	100	0
7	4			ne	30	0
8	5			ne	80	4
9			nevyužito	ne	100	0
10			nevyužito	ne	100	0
11			nevyužito	ne	100	0
12	0	G6C	pravá ruka	ano	100	-3
13	1			ano	30	0
14	2			ano	80	0
15			nevyužito	ne	100	0
16	16		hlava	ne	100	2
17	17			ne	100	3
18	10	G6D	pravá noha	ano	100	0
19	9			ne	76	-2
20	8			ne	145	0
21	7			ano	93	4
22	6			ne	100	-2
23			nevyužito	ne	100	0

$$G24 = (G8A, G8B, G8C) = (G6A, G6B, G6C, G6D)$$

PŘÍLOHA C: TABULKA KOMUNIKAČNÍHO PROTOKOLU

Komunikační příkazy:

Příkaz	Parametr	Popis	
'H'	'j'	Ovládání hlavy	vlevo o 1°
	'J'		max vlevo
	'l'		vpravo o 1°
	'L'		max vpravo
	'i'		nahoru o 1°
	'I'		max nahoru
	'm'		dolu o 1°
	'M'		max dolu
	'k'		centrování
'B'	'w'	Ovládání těla	krok vpřed
	's'		krok vzad
	'a'		krok vlevo
	'd'		krok vpravo
	'q'		otočení vlevo
	'e'		otočení vpravo
'I'			inicializace
'P'	"ab.abc"	Servo	(position) zápis nebo čtení polohy
'S'	"ab.abc"		(speed) zápis nebo čtení rychlosti
'L'	"abc"	LED	(light) ovládání osvětlení
'V'		Ultrazvuk	spuštění měření vzdálenosti (odpověď „Vabc“)
'R'			stav robota

Systémové příkazy:

Příkaz	Parametr	Popis	
SYN	add_rob/add_pc	Synchronizace	komunikace
CAN	add_rob	Cancel	zrušení aktuálního příkazu
ACK	add_rob	Acknowledge	potvrzení
NAK	add_rob	Neg. Acknowledge	zamítnutí
XON	add_rob	Transmit on	Povolení komunikace
XOFF	add_rob	Transmit off	Zakázání komunikace

Poznámky:

- Komunikační příkazy jsou dále zpracovány viz kapitola 4.2.2.
- délka komunikačního příkazu je max 7 bytů
- „abc“ – řetězec třiciferného decimálního čísla, např. „123“
- „ab“ – řetězec dvouciferného decimálního čísla, např. „12“
- add_rob – adresa robota, tj. znak v rozsahu od ‘1’ do ‘9’
- add_pc – pevná adresa počítače znak ‘0’
- celková délka zprávy je max 12 bytů
- ACK = ASCII 0x06, Acknowledgment
- XON = ASCII 0x11, Resume transmission
- XOFF = ASCII 0x13, Pause transmission
- NAK = ASCII 0x15, Negative Acknowledgment
- SYN = ASCII 0x16, Synchronous Idle
- CAN = ASCII 0x18, Cancel
- CR = ASCII 0x0D, Carriage Return