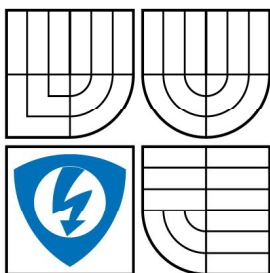


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## ŘÍDICÍ JEDNOTKA PRO HUMANOIDNÍ ROBOT

CONTROL UNIT FOR HUMANOID ROBOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

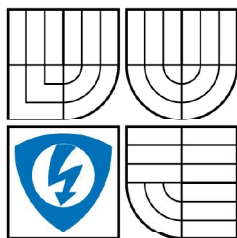
Bc. TOMÁŠ FLORIÁN

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. LUDĚK ŽALUD, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Diplomová práce

magisterský navazující studijní obor  
Kybernetika, automatizace a měření

**Student:** Bc. Tomáš Florián

**ID:** 47465

**Ročník:** 2

**Akademický rok:** 2008/2009

**NÁZEV TÉMATU:**

**Řídicí jednotka pro humanoidní robot**

**POKYNY PRO VYPRACOVÁNÍ:**

Seznamte se s problematikou řízení a stabilizace humanoidních robotů. Seznamte se s předloženou stavebnicí robotu Robonova-I. Navrhněte řídicí jednotku tohoto robotu s výkonnějším řídicím mikrokontrolérem. Navrženou jednotku prakticky realizujte a otestujte.

**DOPORUČENÁ LITERATURA:**

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

**Termín zadání:** 9.2.2009

**Termín odevzdání:** 25.5.2009

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

**prof. Ing. Pavel Jura, CSc.**

*Předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové 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 § 152 trestního zákona č. 140/1961 Sb.

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

## Řídící jednotka pro humanoidní robot

Obor: Kybernetika, automatizace a měření  
Student: Tomáš Florián  
Vedoucí: doc. Ing. Luděk Žalud, Ph.D.

### Abstrakt :

Úkolem práce je seznámit se s komerční stavebnicí krácejícího robota Robonova-I a navrhnout jeho vylepšení. Práce je rozdělena do pěti hlavních kapitol.

První z nich („Robonova-I“) se zabývá stavebnicí Robonova-I, jejím sestavením a možnostmi ovládní. Zabývá se podrobně použitými servomotory, jejich řízením a srovnává je.

Další kapitola se zabývá návrhem řídicí jednotky, popisem architektury procesoru a možnostmi řízení s demonstrační deskou.

Třetí se zabývá softwarovým řešením jednotlivých metod a příkazů, kterými se jednotka s demonstrační deskou ovládá.

Kapitola „Návrh DPS“ popisuje vybavení navržené desky plošných spojů.

Poslední kapitola popisuje koncový program pro řízení robota pomocí počítače a některé algoritmy, kterými robot řídí mikroprocesor.

### Klíčová slova:

Hítec, mikroprocesor, Freescale, Robonova-I, RS232, servo, humanoidní robot, řídicí jednotka

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

## Control Unit for Humanoid Robot

Thesis

Specialisation of study: Cybernetics, Control and Measurement

Student: Tomáš Florián

Supervisor: doc. Ing. Luděk Žalud, Ph.D.

### Abstract :

The main goal of this project is to understand broadband sold Robonova-I humanoid robot and to design new improvements. The thesis is divided into five main chapters:

The first of them (Robonova-I) is concerning the kit of Robonova-I, its completion and control possibilities. In details it deals with used servo motors, their operating and compares them.

The next chapter is concerning the design of the control unit along with description of architecture of the processor and control possibilities with demonstrational board.

In third there is a software solution of particular methods and commands which control the unit with demonstrational board.

Chapter “The Design of Printed Circuit Board” describes facilities of the designed printed circuit board.

The last chapter describes the final program for robot control through the use of computer and some of the algorithms by means of which the robot controls the microprocessor.

### Keywords:

Hitec, microprocessor, Freescale, Robonova-I, RS232, servo, humanoid robot, control unit

### **Bibliografická citace díla**

FLORIÁN, T. Řídicí jednotka pro humanoidní robot. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 82 s. Vedoucí diplomové práce doc. Ing. Luděk Žalud, Ph.D.

## Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Řídicí jednotka pro humanoidní robot jsem vypracoval samostatně pod vedením vedoucího diplomové 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 uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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: **25. května 2009**

.....  
podpis autora

## Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Luděkovi Žaludovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **25. května 2009**

.....  
podpis autora

## OBSAH

<b>1. ÚVOD .....</b>	<b>10</b>
1.1 Terminologie .....	10
1.2 Úkol projektu.....	10
<b>2. ROBONOVA-I.....</b>	<b>11</b>
2.1 Co je Robonova-I.....	11
2.2 Skládání stavebnice.....	11
2.3 Řídící jednotka.....	13
2.4 Digitální robotické servo.....	13
2.4.1 Servomotor.....	13
2.4.2 Specifikace .....	14
2.4.3 Organizace paměti.....	15
2.4.4 HMI protokol [5].....	16
2.5 Softwarová vybavenost a ovládání.....	21
2.5.1 Catch and Play.....	22
2.6 Zhodnocení koncepce a náměty k vylepšení.....	23
2.6.1 Jednoduchost.....	23
2.6.2 Řízení serv.....	23
2.6.3 Uživatelská obsluha.....	23
2.6.4 Napájení .....	24
2.6.5 Úkol projektu .....	24
<b>3. ŘÍDICÍ JEDNOTKA .....</b>	<b>25</b>
3.1 Mikroprocesory Coldfire [1].....	25
3.1.1 Malé, levné a vysoce vybavené.....	25
3.1.2 Strategie vývoje.....	26
3.1.3 Architektura .....	26
3.2 M52233DEMO board [1].....	28
3.2.1 DEMO BOARD .....	28
3.3 MCF52233 .....	29
3.4 Konstrukce .....	31
3.4.1 Standardní a rozšířený mód.....	31

3.4.2 Sériový mód .....	32
<b>4. SOFTWAREVÉ ŘEŠENÍ A OVLÁDÁNÍ TESTOVACÍCH ALGORITMŮ ŘÍZENÍ.....</b>	<b>33</b>
4.1 Standardní a rozšířený mód .....	33
4.1.1 Nastavení sériové linky.....	33
4.1.2 Formát příkazů .....	34
4.1.3 Nastavení polohy, rychlosti a čtení polohy.....	35
4.1.4 Zhodnocení.....	35
4.2 Sériový mód .....	35
4.2.1 nastavení sériové linky .....	35
4.2.2 Formát příkazů .....	35
4.2.3 Zhodnocení.....	39
<b>5. NÁVRH DPS .....</b>	<b>40</b>
5.1 Součástky a konektory .....	41
5.2 Napájecí část.....	41
5.2.1 Stabilizátor LM2576-ADJ D2PAK .....	42
5.2.2 Stabilizátor LF50CDT .....	43
5.2.3 Další konektory .....	43
5.3 Mikroprocesor a okolí.....	44
5.3.1 Programovací konektor.....	44
5.3.2 Sériová linka.....	44
5.3.3 Ethernetový konektor .....	44
5.3.4 Sběrnice I2C [13] .....	44
5.4 Bezdrátová komunikace .....	47
5.5 Analogově digitální převodník (ADC) .....	48
5.5.1 Akcelerometr MMA7260QT .....	48
5.6 Ovládání .....	48
<b>6. PROGRAMY .....</b>	<b>50</b>
6.1 Program pro řízení robota .....	50
6.1.1 Algorismus „MOVE“ .....	50
6.1.2 Algorismus „FAZE“ .....	50

6.1.3 Zhodnocení provedení .....	51
6.2 Program pro ovládání přes počítač .....	51
6.2.1 Hlavní menu .....	52
6.2.2 Obrázek s robotem.....	54
6.2.3 Ovládání robota .....	55
6.2.4 Informační části.....	57
<b>7. ZÁVĚR .....</b>	<b>60</b>
<b>8. SEZNAM PŘÍLOH.....</b>	<b>62</b>
<b>9. POUŽITÁ LITERATURA .....</b>	<b>63</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>65</b>
<b>SEZNAM ROVNIC .....</b>	<b>67</b>
<b>SEZNAM TABULEK.....</b>	<b>68</b>

## 1. ÚVOD

### 1.1 TERMINOLOGIE

Aby nedošlo k mylnému výkladu identifikace předmětu projektu a jeho skloňování, uvádím následující:

V případě, že se jedná o stroj, který vypadá jako člověk (tzv. humanoidní nebo krácející robot), *může* se skloňovat slovo *robot* podle vzoru *pán* (pán, pána,... robot, robota,...). Jinak je skloňování podle vzoru *hrad* (hrad, hradu,... robot, robotu,...). Protože se jedná o první případ, bude v této práci slovo *robot* skloňováno podle vzoru *pán*.

### 1.2 ÚKOL PROJEKTU

Úkolem této diplomové práce je seznámit se s komerční stavebnicí krácejícího robota Robonova-I, prostudovat mechanismus pohonu (robotická serva), navrhnout vylepšený model řídicí jednotky, navrhnout desku plošných spojů (DPS), osadit a oživit DPS, naprogramovat software pro ovládání pomocí PC a sériové linky.

Diplomová práce byla rozdělena do několika částí.

V první fázi sestavit stavebnici humanoidního robota Robonova-I od firmy Hitec. Naučit se ho programovat podle přiloženého software a vyzkoušet, co všechno dovede.

V druhé fázi otestovat vlastní návrh koncepce pro rychlejší a variabilnější řízení tohoto robota (návrh nové řídicí jednotky).

Třetí fáze spočívala v návrhu DPS, vyrobení, osazení součástkami a její oživení.

Čtvrtá fáze byla o návrhu řídicího programu pro PC pomocí RS232.

## 2. ROBONOVA-I

### 2.1 CO JE ROBONOVA-I

Jedná se o 31,8 cm vysokého a 1,33 kg vážícího kráčejícího robota, který je poháněn 16 servomotory. Výrobce je korejská firma Hitec Robotics. Pět servomotorů je pro každou nohu a tři pro každou ruku. Pro pohodlné ovládání je



robot řízen pomocí přiloženého IR ovladače. Při stisku tlačítka se spustí podprogram, který byl tomuto číslu přiřazen v hlavním programu. Robota je možno různě modifikovat (na internetu se nachází spousta neoficiálních vylepšení) a přidávat další komponenty (gyroskop, akcelerometr, snímač vzdálenosti, kamera, rychlejší servo-motor, LCD displej, hlasový modul, bluetooth, radiový modul pro řízení, atd.).

Obr. 1: Robonova-I

### 2.2 SKLÁDÁNÍ STAVEBNICE

Obsah krabice působí dojmem, že z ní nikdy nemůžeme žádného robota složit. Jediné, co nám trochu napovídá o jeho tvaru je hlava a přední a zadní část trupu. Zbytek je do nejmenšího kousku rozmontován. Balení obsahuje stovky šroubků nejrůznějších délek a barev. Aby jich ale nebylo málo, tak se v průběhu



montáže využívají i šroubky, kterými jsou složeny serva, takže celkový počet je ještě větší.

Podle manuálu trvá sestavení robota

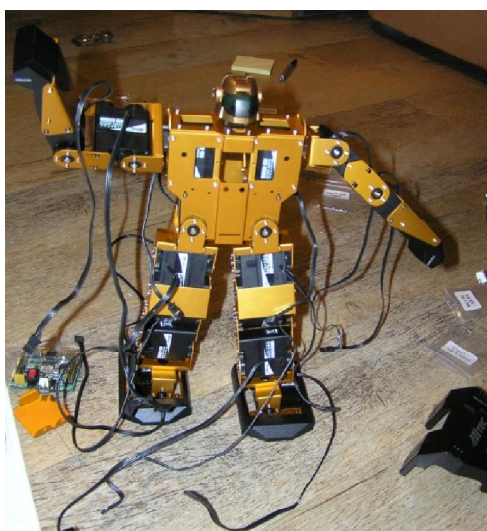
Obr. 2: Obsah balení



šest až osm hodin pouze za pomoci šroubováku. Naše sestavení trvalo čistých 10 hodin práce. Výrobce uváděný čas je spíše pro člověka, který již tuto stavebnici někdy montoval nebo má v tomto ohledu nějaké zkušenosti.

Obr. 3: Stavba robota – nohy a ruce

Montáž probíhá postupně a vše je v manuálu velmi názorně ukázáno. Začíná se stavbou nohou, postupně se přejde ke kompletaci trupu a hlavy a nakonec se připevní ruce. Nutno je dát si pozor na sílu dotahování jednotlivých šroubků. v řadě případů se šroubují do plastu (serva) a je možné jejich přetažení. Při sestavování by

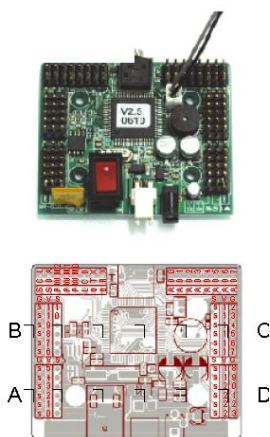


se v některých případech hodilo, aby byl manuál vytištěn barevně (tak, jak je přiložen na CD), protože například finální zapojení serv na příslušnou pozici u řídicí jednotky není, z mého pohledu, úplně logicky uspořádáno. Po zapojení všech kabelů, dotáhnutí posledního šroubku a nabití baterie je robot okamžitě připraven předvést, co umí. Od výrobce je řídicí jednotce nahrán program, který je i na přiloženém CD.

Obr. 4: Stavba robota - tělo

## 2.3 ŘÍDÍCÍ JEDNOTKA

O řízení robota a komunikaci se všemi připojenými zařízeními se stará řídicí jednotka, která je osazena mikroprocesorem ATmega128 od společnosti ATMEL. Základní deska obsahuje 24 signálových pinů pro připojení servomotorů, 8 kanálů A/D převodníku, tři výstupy PWM (pulzní šířková modulace) a I2C konektor pro rozšiřující zařízení. Dále je připojen piezoměnič pro zvukové efekty, konektor pro LED diodu (umístění v hlavě robota), napájení, vypínač, konektor pro připojení k PC přes sériový kabel (RS-232).



Obr. 5: Řídicí jednotka [3]

ATmega128 je 8bitový mikroprocesor architektury RISC, který obsahuje 4kB SRAM, 4kB EEPROM, 8 kanálový A/D převodník (10bit) s taktovací frekvencí 16MHz. [2]

## 2.4 DIGITÁLNÍ ROBOTICKÉ SERVO

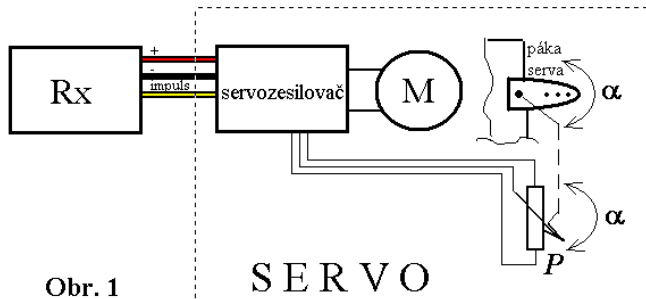
Robot je řízen pomocí robotických servomotorů s označením HSR-8498HB, které vyrobila firma Hitec. Každé servo je osazeno mikroprocesorem od firmy Atmel ATmega8.

### 2.4.1 Servomotor

Servomotor (servo) je motor s integrovanou převodovkou, snímačem polohy (často potenciometrem) a řídicí elektronikou, který má tři vývody – napájení, zem a řídicí signál. Serva se většinou otáčejí pouze v určitém rozsahu (typicky 180° nebo 120°). Natočení se řídí pomocí šířky pulzu, který je přijímán přes řídicí vodič. Jejich nejznámější uplatnění je v modelářském průmyslu. [8]

Servomotory můžeme obecně rozdělit na dva druhy – analogové a digitální. Rozdíl mezi nimi je pouze v použití řídicí elektroniky. Digitální serva řídí mikroprocesor a jsou mnohem přesnější. Nevýhodou je vyšší spotřeba proudu. [9]

Poloha serva je přenášena na potenciometr a jeho jezdec tedy vykazuje

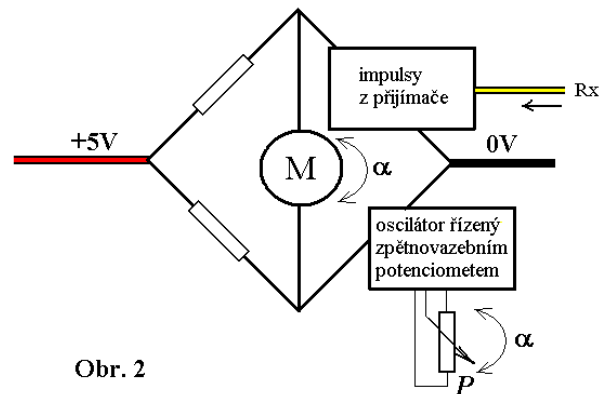


Obr. 1

stejný úhel natočení  $\alpha$  (viz. Obr. 6). Elektronika serva obsahuje generátor impulzů řízený potenciometrem (P).

Obr. 6: Mechanická funkce serva [10]

S natočením motoru se mění šířka pulzů. Tato se porovnává s velikostí žádaného pulzu a podle toho se natáčí motor tak dlouho, dokud nejsou signály totožné. Ve schématu je vyhodnocení znázorněno můstkem. Výše popsané je na Obr. 7. [10]



Obr. 2

Obr. 7: Elektrická funkce serva [10]

#### 2.4.2 Specifikace

HSR-8498HB se vyrábí ve třech modifikacích, tak aby bylo použitelné pro všechny pozice v Robonově. Jednotlivé úpravy jsou znázorněny na obrázku. Jeho rozměry jsou 40 x 20 x 37 mm.

Komunikace probíhá pomocí HMI protokolu, kde je možné využít pulzní šířkovou modulaci (PWM). Toto nám zaručuje kompatibilitu s ostatními servy.

Napájecí napětí je v rozmezí 4,8 až 7,4V, pracovní teplota od -20 do +60°C.



Obr. 8: Druhy servomotorů [3]

Se zvyšujícím se napětím roste rychlost a síla serva. Rychlost otáčení serva je 60° za 0,2 sekundy bez zátěže s kroutícím momentem 7,4 kg na cm při napájení 6.0V (pro 7,4V je doba 0,18 sekund a kroutící moment

9,0 kg na cm).

Rozsah otočení serva je maximálně 190°. Pro běžné používání se doporučuje využít 180° (90° na jednu stranu a 90° na druhou od středové pozice). [4]

### 2.4.3 Organizace paměti

Paměť v rámci serva se dělí na paměť EEPROM a RAM. Obsahy jednotlivých adres paměti jsou uvedeny v tabulkách, které jsou přílohou projektu.

#### 2.4.3.1 Paměť EEPROM

Veškeré změny, které provedeme v této paměti, musí být potvrzeny restartem (vypnutím a opětovným zapnutím) serva, jinak se změny neprojeví. Důležité je po změnách udělat kontrolní součet paměti (tzv. checksum). Pokud by hodnota v něm uložená nebyla správně, servo bude reagovat na všechny příkazy, ale nebude nastavovat polohu. Výpočet kontrolního součtu je podle následujícího vzorce:

výpočet:

$$\text{checksum} = \text{Sum}[0x00:0x2B] \% 256$$

Rov. 1: Výpočet kontrolního součtu pro EEPROM [7]

*pozn.: % je znak pro operaci modulo (zbytek po celočíselném dělení)*

podmínka:

$$\text{Sum}[0:0x2C] \% 256 == 0$$

Rov. 2: Podmínka kontrolního součtu pro EEPROM [7]

#### 2.4.3.2 Paměť RAM

Při zapnutí napájení se provede načtení hodnot z EEPROM od adresy 0x00 po adresu 0x1E do paměti RAM, kde jsou dostupné od adresy 0x80 do adresy 0x9E. Pomocí adresace do paměti RAM můžeme provádět jejich změnu v EEPROM, aniž bychom museli provádět kontrolní součet. Změny v této paměti se okamžitě projeví na chování serva.

#### 2.4.4 HMI protokol [5]

HMI protokol obsahuje tři módy:

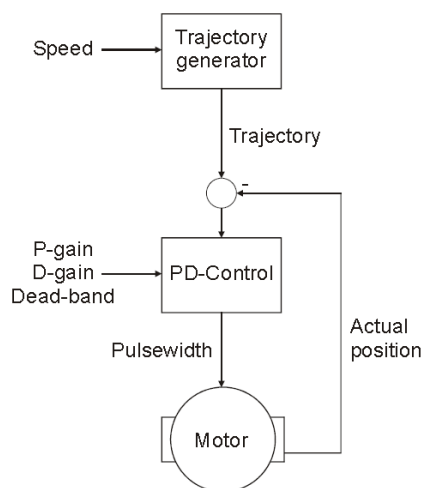
1. Standardní mód (550 až 2450  $\mu\text{s}$ )
2. Rozšířený mód (50  $\mu\text{s}$ )
3. Sériový mód (416  $\mu\text{s}$ )

Protože každý signál pracuje v jiném časovém úseku, není potřeba žádná inicializace nebo jeho identifikace.

Na výběr máme tři předdefinovaná nastavení, která můžeme měnit, a mezi kterými se můžeme přepínat za běhu programu:

- 1) Šířka pulzu 100  $\mu\text{s}$  – použít nastavení 1 (výchozí) [3]
- 2) Šířka pulzu 150  $\mu\text{s}$  – použít nastavení 2 [3]
- 3) Šířka pulzu 200  $\mu\text{s}$  – použít nastavení 3 [3]

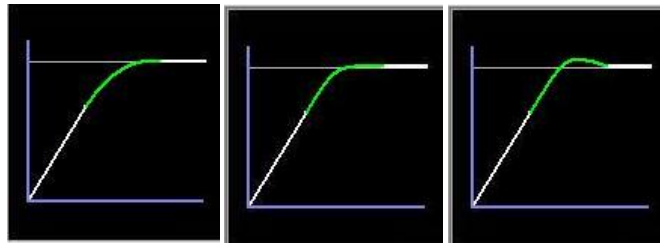
Servo je řízeno PD regulátorem (PD = proporcionálně-derivační) a jeho jednotlivé složky můžeme měnit přepínáním mezi jednotlivými nastaveními.



Obr. 9: Schéma zapojení a řízení serva [7]

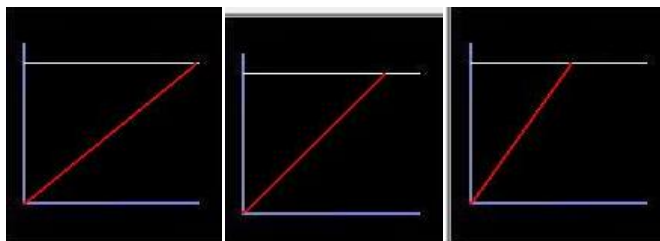
Změnou nastavení měníme následující parametry:

- a) Zesílení D
- b) Zesílení P
- c) Mrtvé pásmo



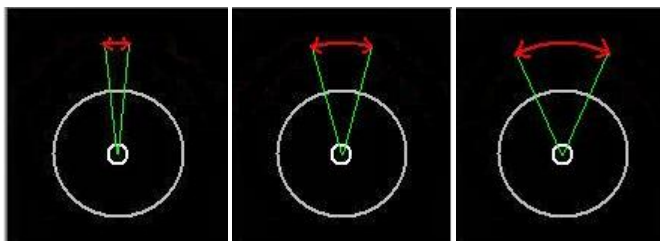
Obr. 10: Vliv parametru D – hodnoty 1, 2, 3 [6]

Tento parametr nám udává, jak plynule servo zastaví (bez překmitu, s překmitem). [6]



Obr. 11: Vliv parametru P – hodnoty 1, 2, 3 [6]

P ovlivňuje rychlost a výkon serva. [6]



Obr. 12: Vliv parametru „mrtvé pásmo“ – 1, 2, 3 [6]

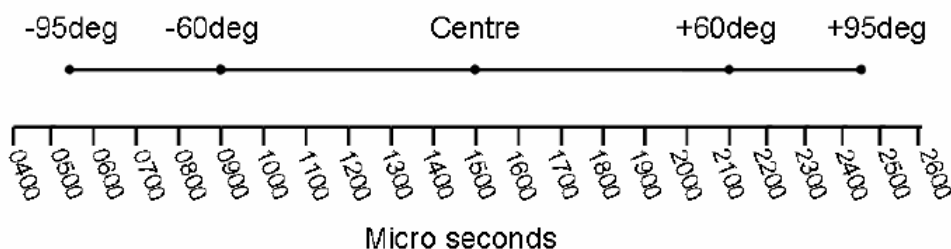
*Mrtvé pásmo* udává velikost nevyužitého rozmezí ke konci trajektorie. [6]

Nejvyšší hodnoty (3) u každého z parametrů mohou způsobit vysoké chvění a velké opotřebení serva. [6]

#### 2.4.4.1 Standardní mód

Servo pracuje se signály o délce 550 až 2450  $\mu$ s, pomocí kterých nastavujeme jeho pozici. Pokud přijme signál mimo toto rozpětí, nebude na něj

reagovat změnou polohy. Střed serva je 1500  $\mu$ s. V tomto módu je nutné neustále posílat servu pozici, kterou má nastavit, jinak dochází ke ztrátě točivého momentu.



Obr. 13: Šířka signálů (ms) a k nim odpovídající poloha serva ve stupních (deg) [5]

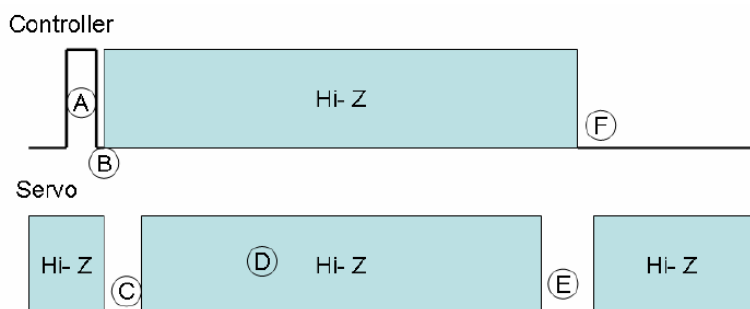
vzorec pro výpočet úhlu natočení serva:

$$\text{úhel} = (\text{šířka} - 1500) / 10 [5]$$

Rov. 3: Vzorec pro výpočet úhlu natočení serva

#### 2.4.4.2 Rozšířený mód

Rozšířený mód je speciálně vytvořen pro potřeby Robonovy. Jedná se o režim, při kterém čteme ze serva jeho polohu. Abychom mohli tento režim využít, je potřeba mít připojený na sběrnici pull-up rezistor (4,7 k $\Omega$  proti napájecímu napětí). Je nutné brát ohled na fakt, že pokud provedeme čtení polohy serva, tak servo zpomalíme (snížíme rychlost pohybu a točivý moment a může se stát, že robot v důsledku toho spadne). V tomto režimu můžeme se servy libovolně pohybovat rukou a nastavovat manuálně žádanou polohu, kterou následně přečteme a uložíme.



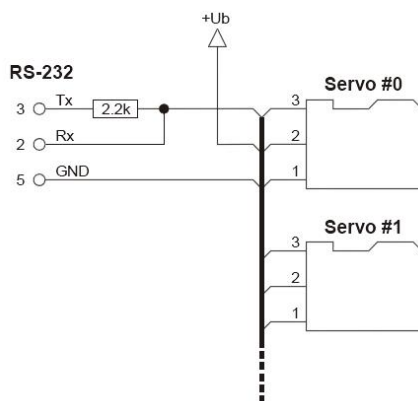
Obr. 14: Průběh rozšířeného módu [5]

pozn.: HI-Z je režim vysoké impedance

- A) Kontrolér vyšle pulz o šířce 50  $\mu$ s [5]
- B) Kontrolér dá na výstup hodnotu 0 po dobu 2 až 20  $\mu$ s [5]
- C) Servo rozpozná polohu a nastaví hodnotu 0 po dobu 125  $\mu$ s [5]
- D) Servo odešle pulz o šířce, která odpovídá poloze (550 až 2450  $\mu$ s) [5]
- E) Servo nastaví hodnotu 0 [5]
- F) Kontrolér identifikuje pulz a nastaví hodnotu 0 [5]

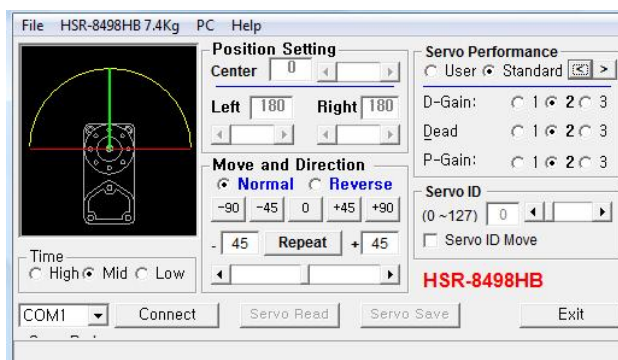
#### 2.4.4.3 Sériový mód

V tomto módu můžeme připojit až 128 serv přes jeden signálový vodič ke



kontroléru a ovládat je. Jako rozhraní je zde využita sériová linka. Jediné, co je nutné udělat před připojením více serv, je jejich inicializace. Nejdůležitější je každému nastavit unikátní ID číslo (0 až 127). Dále je nutné načíst do paměti údaje a spočítat kontrolní součet (checksum), bez kterého by serva reagovala na všechny příkazy, ale nefungovalo by nastavování polohy.

Obr. 15: Zapojení pro sériovou komunikaci [7]



Obr. 16: HMI Servo Programmer [6]

Jako podporu pro inicializaci serv, jejich nastavení a názorné ukázce, co který parametr mění, doporučuji program HMI Servo Programmer, který má firma Hitec k dispozici na svých stránkách ([odkaz do sekce pro stahování](#)).

#### 2.4.4.3.1 Komunikační protokol

Komunikace je pro následující nastavení RS-232: [7]

rychlost: 19,2 kbaudů

datové-bity: 8

stop-bity: 2

parita: žádná

Každý příkaz se skládá ze sedmy bajtů. Poslední dva bajty jsou vždy nulové a promítá se do nich odpověď serva (ze zapojení na Obr. 15 je patrné, že se jedná o zpětnou vazbu).

	0	1	2	3	4	5	6
<i>Host:</i>	0x80	Command	Param1	Param2	Checksum	0x00	0x00
<i>Servo:</i>	High-Z					Return1	Return2

Obr. 17: Komunikační protokol [7]

#### Výpočet kontrolního součtu (checksum):

*podmínka:*

$$\text{Sum}[0:4] \% 256 == 0x00$$

Rov. 4: Podmínka pro kontrolní součet příkazu v sériovém módu [7]

*výpočet:*

$$\text{Checksum} = 256 - (0x80 + \text{Command} + \text{Param1} + \text{Param2}) \% 256$$

Rov. 5: Výpočet kontrolního součtu příkazu v sériovém módu [7]

#### 2.4.4.3.2 Syntaxe příkazů

V následující tabulce jsou uvedeny druhy příkazů a jejich hodnoty pro jednotlivé bajty, které se používají v komunikačním protokolu (viz. Obr. 17).

<i>Command</i>	<i>Param1</i>	<i>Param2</i>	<i>Return1</i>	<i>Return2</i>	<i>Description</i>
0..0x7F (id)	pos_high	pos_low	0x00	0x00	Set servo #id target position
0xE1	addr	0x00	data	0x03	Read EEPROM
0xE2	addr	data	0x03	0x03	Write EEPROM
0xE3	addr	0x00	data	0x03	Read memory
0xE4	addr	data	0x03	0x03	Write memory
0xE5	0x00	0x00	pos_high	pos_low	Read position
0xE6	pos_high	pos_low	0x00	0x00	Set target position
0xE7	0x00	0x00	version	id	Read version & id
0xE8	0x00	0x00	pulsewidth	voltage	Read pulsewidth & voltage
0xE9	0..0x7F (id)	speed	pos_high	pos_low	Set servo #id speed & read position
0xEA	0x00	1 / 2 / 3	0x03	0x06	Select control parameter set
0xEB	0x00	1 / 0	0x03	0x06	Set go / stop
0xEC					
0xED					
0xEE					
0xEF	0x00	0x00	0x03	0x06	Release

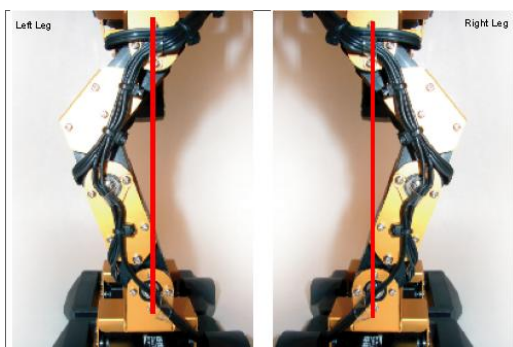
Tab. 1: Seznam příkazů pro sériovou komunikaci [7]

Z tabulky příkazů je patrné, že podle ID čísla můžeme nastavit žádanou pozici serva, nastavit rychlost a přečíst pozici. Nemůžeme, ale přečíst rychlost serva, takže je nutné si ji pamatovat v řídicím programu. Dále nemůžeme provádět jakákoliv další nastavení (čtení nebo zápis paměti), protože jsou to příkazy s platností pro všechna serva na lince. Můžeme tedy u všech serv změnit hodnoty P, D a mrtvého pásma nebo je zapnout či vypnout. Důležité ale je, ve které z pamětí změny provádíme. K těmto změnám můžeme totiž využít pouze standardní paměť RAM. Kdybychom se pokusili změnu zapsat do paměti EEPROM, nejsme schopni udělat správný kontrolní součet a serva by po restartování, které je pro potvrzení hodnot nutné, přestala fungovat.

## 2.5 SOFTWAREVÁ VYBAVENOST A OVLÁDÁNÍ

Robot je vybaven programem RoboBasic, pomocí kterého jsme schopni pohodlně naprogramovat ovládání jednotlivých serv. Vše je graficky znázorněno a programování je velmi intuitivní. Nechybí zde základní příkazy pro cyklus,

podmínku, čekání na provedení příkazu, zpoždění, nastavení rychlosti, polohy a mnoho dalších. Výrobce vytvořil 31 základních pohybů robota. Od úklonu, přes kroky na všechny strany, běh, kop, otáčení, holubičku, stojku, úder až po velmi



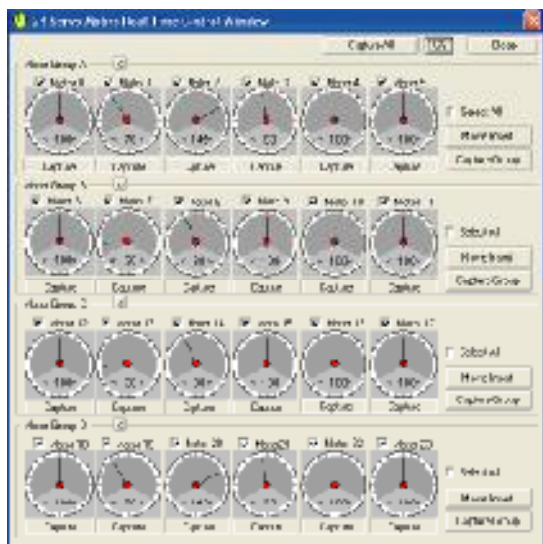
Obr. 18: Nastavení nulových pozic [3]

efektní kotouly všemi směry.

Na začátku je nutné nastavit u všech serv nulovou polohu. Musíme tedy jemnými kroky posunout všechna serva do poloh podle obrázků. Pak teprve může přiložený software fungovat bez problémů.

### 2.5.1 Catch and Play

Pro snazší nastavení nových pozí má program implementovanou funkci Catch and Play (uchop a hraj). Vybereme skupinu serv (ruce, nohy, pravá nebo levá strana) nebo pouze jednotlivá serva, které chceme takto nastavit.



Těmto zaškrtnutím snížíme sílu a můžeme je dát do libovolné polohy rukou. Pak jen dáme načíst pozici a tu uložíme do programu. Takto jsou i úplní začátečníci schopni vytvářet nové pohyby podle vlastních potřeb.

Obr. 19: Nastavení serv na požadovanou polohu [3]

## 2.6 ZHODNOCENÍ KONCEPCE A NÁMĚTY K VYLEPŠENÍ

### 2.6.1 Jednoduchost

Navržená koncepce od firmy Hitec je určená pro sériovou výrobu a také pro jednoduchost zapojování. Uživatel nemusí dlouze přemýšlet, kam které servo umístí a výrobce nemusí každému přiřazovat unikátní ID, které následně umístí i na jeho obal. Navíc by bylo velmi složité hlídat, aby do stavebnice šla pouze serva s různými ID hodnotami (protože se serva nevyrobí jen do těchto stavebnic) a složitěji by se musely řešit opravy a reklamace.

### 2.6.2 Řízení serv

Řízení pomocí šířky pulzů je náročnější pro kontrolér (neustále vysílá pulzy do všech serv), který by se v té chvíli mohl věnovat jiným úkolům.

Také z toho plyne omezení v počtu serv – pro kompatibilní protokol je doporučena obnovovací frekvence 50 Hz. Tedy, že můžeme vyslat signál do 50 serv během jedné sekundy. Samo o sobě je to velké omezení, protože už zpožděná reakce o půl sekundy může mít pro robota katastrofální dopady. Pro případ našeho robota je teoreticky minimální rychlost taková, jaká je minimální doba pro signál polohy. V praxi byla otestování frekvence o hodnotě 200 Hz. Zde jsme již na limitech protokolu, a pokud bychom se chystali řídit nějaký sofistikovanější stroj, museli bychom mít více kontrolérů.

Dalším problémem je i náročnost na počet portů mikroprocesoru – každé servo je jeden bit portu.

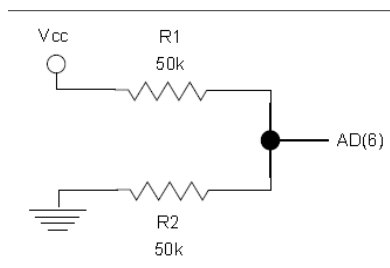
### 2.6.3 Uživatelská obsluha

Ovládání celého robota je na dobré úrovni. Příkazy z IR vysílače zachytává a zpracovává bez větších problémů i z několika metrů ze všech stran (důležité je mít nabitě baterie v ovladači). Pohodlné by mohlo být ovládání robota pomocí radiových vln, ale vzhledem k tomu, že robot nemá kameru a musíme ho mít tedy neustále na očích, není tento nedostatek na závadu. Nehledě na to, že by to mělo za následek větší spotřebu energie.

#### 2.6.4 Napájení

Robot je vybaven baterií Ni-MH, která je složena z pěti článků s celkovým napětím 6,0 V a kapacitě 1.000 mAh (1 Ah). Nabíjení baterie probíhá přes řídicí jednotku, ale pouze v případě, že je robot vypnutý. Při jeho zapnutí dochází k odpojení nabíječky. Nabíjení zcela vybité baterie na maximum se pohybuje okolo jedné hodiny. Na jedno nabití je robot schopen fungovat cca 30 minut. Doba provozu značně závisí od úkonů, které provádí (počet a četnost využívání serv a jejich rychlost).

Pro potřeby signalizace vybité baterie je v manuálu návod na jednoduchý obvod, který se připojí na libovolný port AD převodníku a ten nám může ukazovat



(blikání LED diody robota v hlavě), že je potřeba jej nabít. Jedná se vlastně o jednoduchý dělič napětí s poměrem 1:1. Pro naše potřeby byl tento obvod vyroben a odzkoušen.

Obr. 20: Schéma obvodu pro signalizaci slabé baterie [3]

#### 2.6.5 Úkol projektu

Úkolem tohoto projektu bylo navrhnout koncepci řízení robota s rychlejším procesorem, větší podporou připojitelných periférií a snazším ovládním připojených serv. Pro testování byl zvolen hotový výrobek M52233DEMO BOARD, na kterém se vyzkoušelo řízení. Na základě těchto zkušeností byla vyvinuta deska plošných spojů pro řízení robota.

### 3. ŘÍDICÍ JEDNOTKA

#### 3.1 MIKROPROCESORY COLDFIRE [1]

První jádro s označením ColdFire se objevilo v roce 1994 a technologicky vycházelo z legendární rodiny procesorů 68K. Hned od počátku se tyto procesory vyznačovaly proměnnou délkou jak instrukcí, tak i dat.

Již od začátku vývoje kladla Motorola (dnes již Freescale) velký důraz na koncové zákazníky a ve snaze vyjít jím vstříc s touto novou architekturou, vyhodnotil zdrojové kódy z mnoha 68K systémů a na základě výsledků této studie, navrhl redukovanou instrukční sadu (která vychází z originální instrukční sady procesorů 68000, jež měly instrukční sadu kompletní) a dále jádro jako vhodné prostředí pro tyto výkonné instrukce. Stejně jako jiné procesory RISC, je většina instrukcí ColdFire provedena v jednom instrukčním cyklu.

RISC architektura jádra ColdFire s proměnnou délkou instrukcí a dat dává uživatelům větší flexibilitu a to při menších nárocích na paměť a systémové prostředky. Protože instrukce mohou být 16, 32 nebo 48 bitů dlouhé, program může být efektivněji uložen v paměti což má za následek lepší hustotu kódu, než mají klasické 32 nebo 64bitové RISC procesory. Efektivnější použití paměti na čipu redukuje šíři sběrnice a nároky na externí paměť, což opět vede ke zmenšení nákladů na systém řízený pomocí těchto procesorů.

##### 3.1.1 Malé, levné a vysoce vybavené

Jádro ColdFire výrazným způsobem redukuje náklady na celkový embedded systém kde je použit. Je to zejména díky tomu, že je kompletně upravitelné (z výrobního hlediska) a jednoduše se do něho dají integrovat systémové moduly, (MAC, FPU atd.) paměti a periférie (porty, rozhraní atd.). Díky jednotce pro správu paměti lze na těchto procesorech provozovat i operační systémy jako je např. Linux či WindowsCE.

Kvůli vzájemné provázanosti obou architektur (68K a ColdFire), mají současní uživatelé 68K procesorů možnost jak zmodernizovat své návrhy bez výraznějších

časových ztrát spojených s přechodem na novou platformu. Jelikož je instrukční sada ColdFire subsadou rodiny procesorů 68K, přechod na novou architekturu není tak náročný.

Díky širokému mixu výkonu, dobré ceny, vysoké integrace obvodů, vyspělé schopnosti debugingu a více jak 20leté historii procesorů 68K, si ColdFire vydobyl velmi slušnou pozici v 32bitovém segmentu trhu. Obrovský je také rozsah nabízených komunikačních rozhraní:

UART; USART; I2C; FEC; QSPI; USB 1.0, 1.1, 2.0; Ethernet 10/100; SII; CAN

### 3.1.2 Strategie vývoje

Strategie vývoje architektury ColdFire je rozdělena do dvou částí - zákaznický specifikované produkty pro velké firmy a standardní produkty pro všeobecné vestavěné systémy. Protože jsou všechna ColdFire jádra 100% slučitelná, je vcelku jednoduché integrovat novou technologii do zákaznického systému. Tyto procesory jsou ve velkém měřítku používány v např. v inkoustových a laserových tiskárnách, audio elektronice, síťových zařízeních, set-top boxech atd.

### 3.1.3 Architektura

Každá verze jádra má několik konkrétních typů lišících se počtem pinů, čítačů, časovačů, taktem jádra, velikostí paměti atd. Následuje základní výčet dělení na jednotlivé verze jádra:

V1 - ač je číslován jako první, jedná se o jádro, které je zjednodušením jádra V2 jedná se o typ, který byl vyvinut jako přechod mezi 8bitovými a 32bitovými systémy. Dokáže zpracovávat 32bitové data a instrukce, které pak převádí na 8bitovou sběrnici a periférie. Proto se skvěle uplatní při modernizaci již existujících aplikací bez výraznějších zásahů do obvodů. Přináší tak výkon 32bitového systému při zachování 8bitové jednoduchosti. Tyto procesory jsou pinově kompatibilní s rodinou mikrokontrolérů S08 (což jsou právě 8bitové MCU).

V2 - toto jádro poskytuje rychlost až 166 MHz, je vyrobeno 0.13  $\mu\text{m}$  technologií a používá 32bitovou sběrnici jak pro data, tak pro instrukce. Dále pak obsahuje integrovanou jednotku pro odlaďování aplikace (debugging) dvou-

stupňovou fetch pipeline pro instrukce a dvou-stupňovou execution pipeline pro operandy matematických a logických operací.

V3 - jádro verze 3 přináší novinky zdokonalená instrukční prefetch pipeline, stochastické schopnosti větvení programu (funkce předpovědi následující instrukce) a vyšší frekvenci prováděných instrukcí. Tato zlepšení umožňují jádru poskytnout až o 300% více výkonu než jádro V2 a to i při navýšení frekvence jádra o méně jak 50%. Dále pak nabízí vylepšení dosavadních funkcí jako je vysoko-rychlostní multiply-accumulate unit (MAC unit), vyhrazená hardwarová dělička a rozšířené ladící schopnosti.

V4 - tato generace mikroarchitektury poskytuje 2.8x větší výkon v porovnání s předcházející verzí V3. I zde se opět modernizovaly funkční bloky jako je MAC jednotka, debugovací jednotka. Zvětšila se velikost jádra na 4.5mm<sup>2</sup> při výrobní technologii 0.25  $\mu$ m. Původní 2fázové pipelines se zvětšili na 4fázovou Instruction Fetch Pipeline a 5fázovou Operand Execution Pipeline. Výhodou tohoto jádra je i to, že naprostá většina instrukcí jsou 1cyklová. Instruction fetch pipeline dále obsahuje 2stupňový akcelerační mechanismus, který nadále zmenšuje dobu potřebnou na provádění změn.

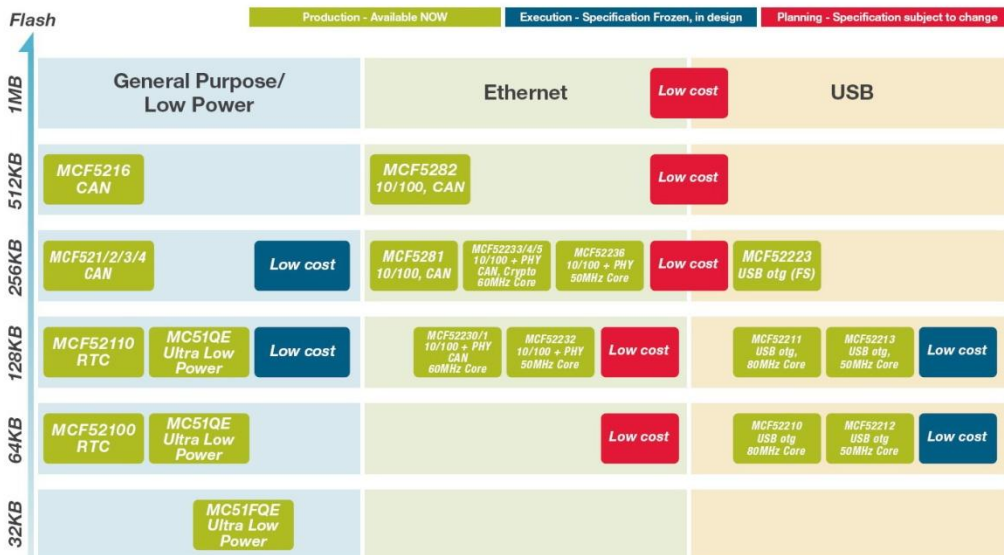
V4e - jádro s tímto označením bylo navrženo pro poskytnutí hardwarové podpory pro multi-processorové aplikace, které vyžadují vysoký výpočetní výkon, který nemůže jednočipové řešení nabídnout. Jako výrobní technologie byla použita 0.18  $\mu$ m a jádro s těmito parametry může běžet až na 225 MHz, jsou i varianty vyráběné 0.13  $\mu$ m technologií, kde rychlost jádra dosahuje hodnoty 333 MHz.

V5 - technologicky minimálně rozdílné od jádra V4e, poskytuje však 1.3x - 1.4x větší výkon při stejné výrobní technologii. Frekvence jádra 300 – 366 MHz.

ColdFire Microcontroller Roadmap



MCF52 – V2 Core  
 MCF51 – V1 Core



Freescale Semiconductor Confidential and Proprietary Information. Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2005.

Obr.

21: ColdFire Roadmap

3.2 M52233DEMO BOARD [1]

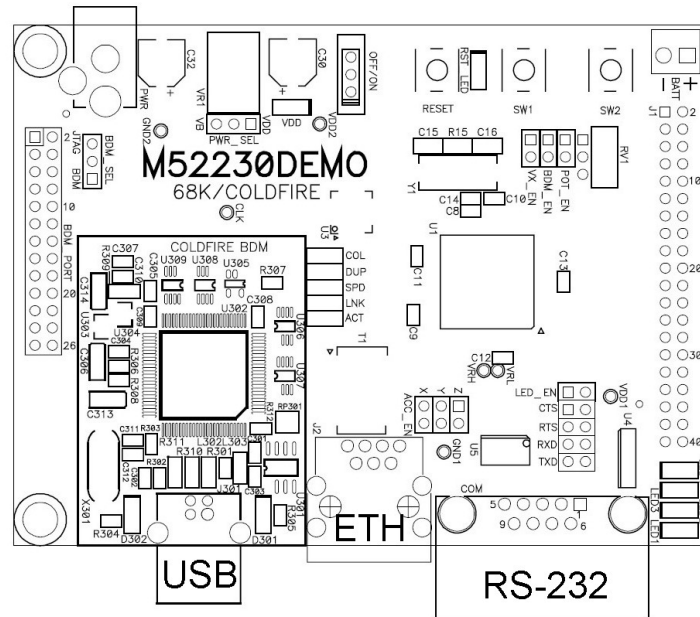
3.2.1 DEMO BOARD

Testování nové koncepce probíhalo demonstrační desce od společnosti Freescale. M52233DEMO BOARD je plně vybavený kit, který obsahuje velké množství periférií. Pro naše potřeby je nejdůležitější mikroprocesor, 256 kB paměti Flash, 32 kB paměti RAM,



40pinový vstupně/výstupní port, USB port (pro programování) a RS-232. Dále možnost napájet desku přes vstup napětím v rozmezí 9-15 V nebo baterii s napětím 3,3 V.

Obr. 22: M52233DEMO BOARD [1]



Obr. 23: Schéma modulu [1]

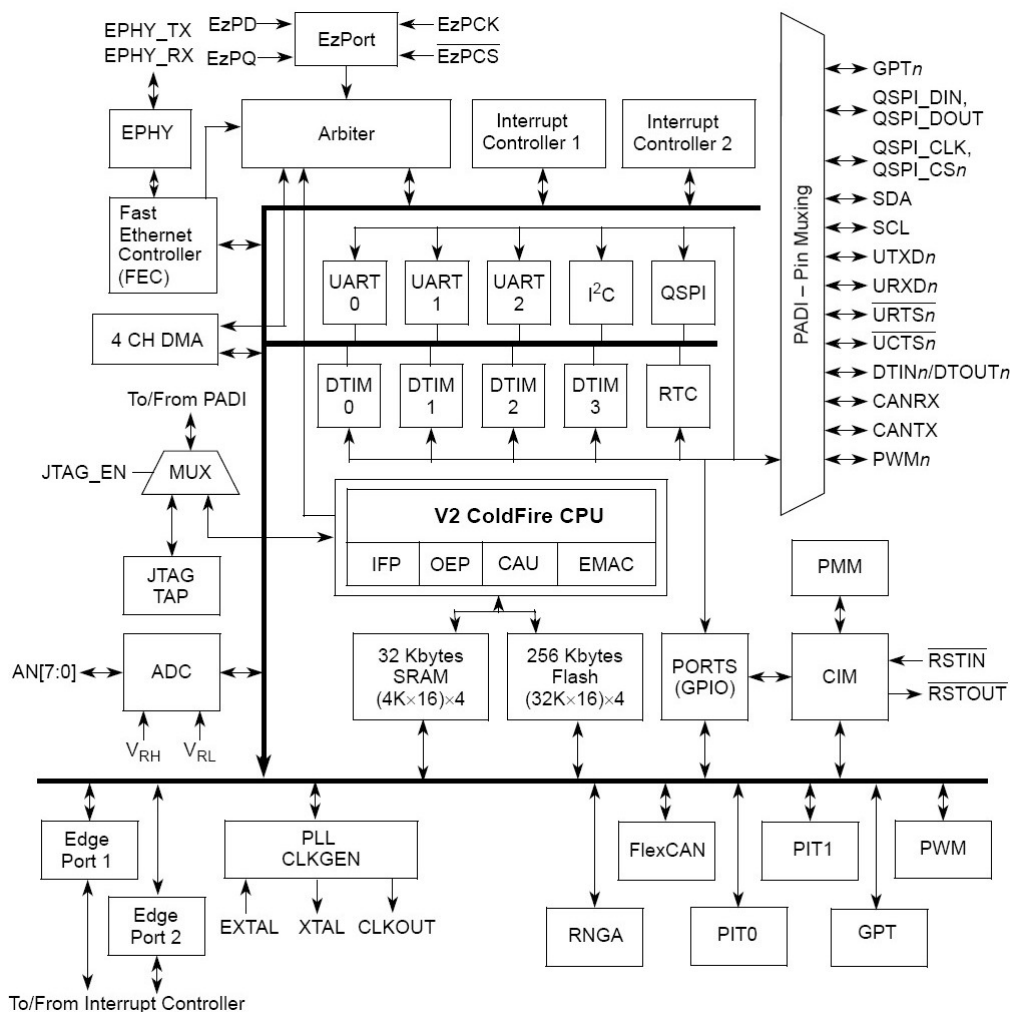
### 3.3 MCF52233

Jak je patrné ze začátku kapitoly, jedná se o 32bitový FLASH mikrokontrolér V2 pro průmyslové řídicí aplikace. Na čipu je osazen Fast Ethernet Controller, Ethernet PHY a ColdFire koprocesor.

Module	52230	52231	52232	52233	52234	52235	52236
Version 2 ColdFire Core with EMAC (Enhanced Multiply-Accumulate Unit)	•	•	•	•	•	•	•
System Clock (MHz)	60	60	50	60	60	60	50
Performance (Dhrystone 2.1 MIPS)	56	56	46	56	56	56	46
Flash / Static RAM (SRAM)	128/32 Kbytes	128/32 Kbytes	128/32 Kbytes	256/32 Kbytes	256/32 Kbytes	256/32 Kbytes	256/32 Kbytes
Interrupt Controllers (INTC0/INTC1)	•	•	•	•	•	•	•
Fast Analog-to-Digital Converter (ADC)	•	•	•	•	•	•	•
Random Number Generator and Crypto Acceleration Unit (CAU)	—	—	—	—	—	•	—
FlexCAN 2.0B Module	—	•	—	—	•	•	—
Fast Ethernet Controller (FEC) with on-chip interface (EPHY)	•	•	•	•	•	•	•
Four-channel Direct-Memory Access (DMA)	•	•	•	•	•	•	•
Software Watchdog Timer (WDT)	•	•	•	•	•	•	•
Programmable Interrupt Timer	2	2	2	2	2	2	2
Four-Channel General Purpose Timer	•	•	•	•	•	•	•
32-bit DMA Timers	4	4	4	4	4	4	4
QSPI	•	•	•	•	•	•	•
UART(s)	3	3	3	3	3	3	3
I <sup>2</sup> C	•	•	•	•	•	•	•
Eight/Four-channel 8/16-bit PWM Timer	•	•	•	•	•	•	•
General Purpose I/O Module (GPIO)	•	•	•	•	•	•	•
Chip Configuration and Reset Controller Module	•	•	•	•	•	•	•
Background Debug Mode (BDM)	•	•	•	•	•	•	•
JTAG - IEEE 1149.1 Test Access Port <sup>1</sup>	•	•	•	•	•	•	•
Package	80 LQFP 112 LQFP	80 LQFP 112 LQFP	80 LQFP	80 LQFP 112 LQFP	112 LQFP 121 MAPBGA	112 LQFP 121 MAPBGA	80 LQFP

<sup>1</sup> The full debug/trace interface is available only on the 112- and 121-pin packages. A reduced debug interface is bonded on the 80-pin package.

Tab. 2: Rodina mikroprocesorů řady MCF5223x



Obr. 24: Blokové schéma mikroprocesorů řady MCF5223x

### 3.4 KONSTRUKCE

Práce spočívala ve dvou metodách řešení. První byla zkusit kombinaci standardního a rozšířeného módu a zjistit, co všechno jsme v tomto režimu schopni dělat a jaké to má výhody a nevýhody. Druhá varianta je návrh koncepce pro sériový mód a opět zjištění jeho výhod a nevýhod.

#### 3.4.1 Standardní a rozšířený mód

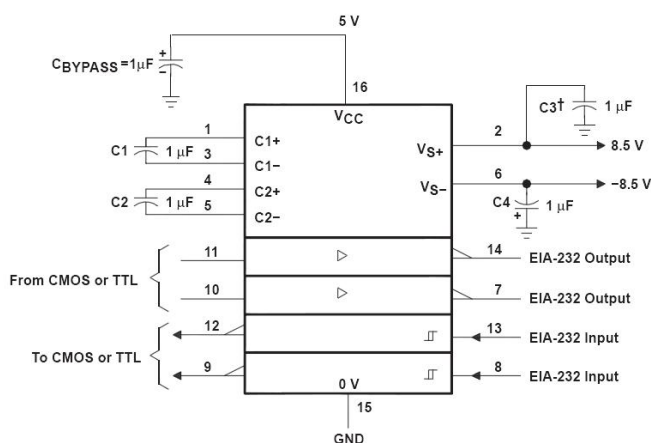
Zapojení bylo provedeno dle doporučení výrobce. Ke každému signálovému vodiči od serva byl připojen pull-up rezistor o velikosti 4,7 kΩ. Vzhledem k tomu, že použitá deska obsahuje pouze jeden plnohodnotný port (PORT AN), bylo

vyzkoušeno 6 serv současně (kompletní řízení rukou). V této konfiguraci program nevykazoval žádná omezení.

### 3.4.2 Sériový mód

V sériovém módu byl signálový vodič společný, a také byl doplněn o pull-up rezistor, tentokrát o velikosti 2,2 k $\Omega$ . V tomto režimu bylo otestováno řízení všech 16 serv. Vše probíhalo bez komplikací. Jediný problém se vyskytl při zapínání robota a jeho inicializaci. V tento moment se totiž většinou hýbe všemi servy, takže celkový proud, který může u jednoho serva nabývat hodnoty přes 1 A, je opravdu veliký. Pro přiloženou baterii by to nebyl problém (je schopna špičkově dodávat vysoké proudy, které mnohonásobně převyšují její kapacitu), ale laboratorní zdroj s omezením 2 A mnohokrát indikoval přetížení. Z tohoto důvodu se někdy nepodařilo všechna serva inicializovat a musel být proveden softwarový reset. Nakonec bylo tomuto problému předcházeno snížením rychlosti, kterou se serva do výchozí polohy dostávají.

#### 3.4.2.1 Převodník napěťových úrovní



Pro propojení serv v sériovém módu s námi navrženou řídicí deskou je nutné použít převodník napěťových úrovní z RS232 na TTL logiku a naopak. Bylo použito standardní zapojení, které je vidět na obrázku.

Obr. 25: Zapojení převodníku úrovní [11]

## 4. SOFTWAREVÉ ŘEŠENÍ A OVLÁDÁNÍ TESTOVACÍCH ALGORITMŮ ŘÍZENÍ

Program pro mikroprocesor byl napsán ve vývojovém prostředí CodeWarrior ColdFire Development Studio IDE version 5.9.0 build 3024 v jazyce C.

U obou módů je nutné dodržet přesnou syntaxi příkazů (vše malými písmeny, dodržovat čárky a uvádět všechny pozice), jinak se může stát, že příkaz nebude rozpoznán.

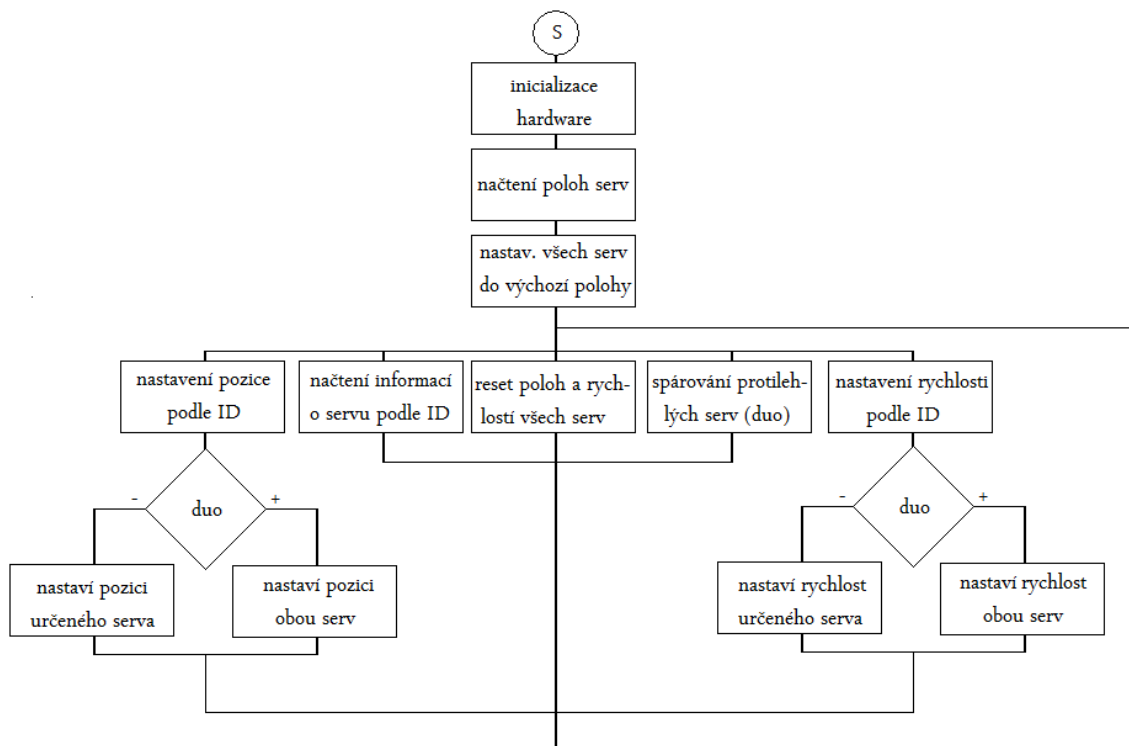
### 4.1 STANDARDNÍ A ROZŠÍŘENÝ MÓD

#### 4.1.1 Nastavení sériové linky

Komunikace probíhá přes sériovou linku (např. pomocí programu Bray++) s těmito parametry:

rychlost: 115,2 kbaudů  
šířka: 8 bitů  
stop bity: 1  
parita: žádná

#### 4.1.2 Formát příkazů



Obr. 26: Zjednodušený vývojový diagram pro první variantu

#### Příkazy pro nastavení jednoho serva (podle ID):

*sxy,a,hodnota*

*s* – nutná identifikace příkazu (*s* - servo)

*xy* – číslo serva (00 až 99)

*a* – událost: *p* - poloha, *r* - rychlost, *cti* - načtení polohy serva (čárka a parametr hodnota se poté neuvádějí)

*hodnota* – hodnota v rozsahu 001-255 pro rychlost a 010 až 190 pro úhel, je nutná brát ohled na to, že hodnotu 1 je nutné napsat jako 001. Hodnoty jsou uváděny dekadicky. Pokud je zadán hodnota mimo rozsah, příkaz se neprovede.

#### Doplňkové příkazy:

*res* - resetuje nastavení všech serv do výchozí polohy

*duo* - spáruje nadefinovaná serva (u dvou serv provádí stejný pohyb se stejnou rychlostí)

#### 4.1.3 Nastavení polohy, rychlosti a čtení polohy

Koncepce hodnot byla převzata od tvůrce robota. Serva se tedy pohybují od 0 do 180° a střed je v hodnotě 90°. Má to výhodu v tom, že nám odpadnou záporná čísla (-90 až +90°).

Rychlost se reguluje také stejně. Měníme šířku pulzu s různou rychlostí. Serva v tomto režimu jedou plnou rychlostí a rychlost proto řídíme pouze pomocí programu.

Čtení polohy dělá programu největší problémy. Pokud pustíme opakované čtení polohy, budeme se pohybovat v rozpětí cca  $\pm 3^\circ$ .

#### 4.1.4 Zhodnocení

Program se přes některé problémy podařilo „odladit“ tak, aby vykonával všechny potřebné funkce – čtení a nastavování polohy a nastavování rychlosti. Díky některým nepřesnostem a zbytečnému zatěžování mikroprocesoru však není doporučen pro řízení robota. Z tohoto důvodu také nebyl příliš rozvíjen a není tak obsáhlý jako druhá varianta.

## 4.2 SÉRIOVÝ MÓD

### 4.2.1 nastavení sériové linky

Komunikace je stejná jako pro první řešení, tedy:

rychlost: 115,2 kbaudů

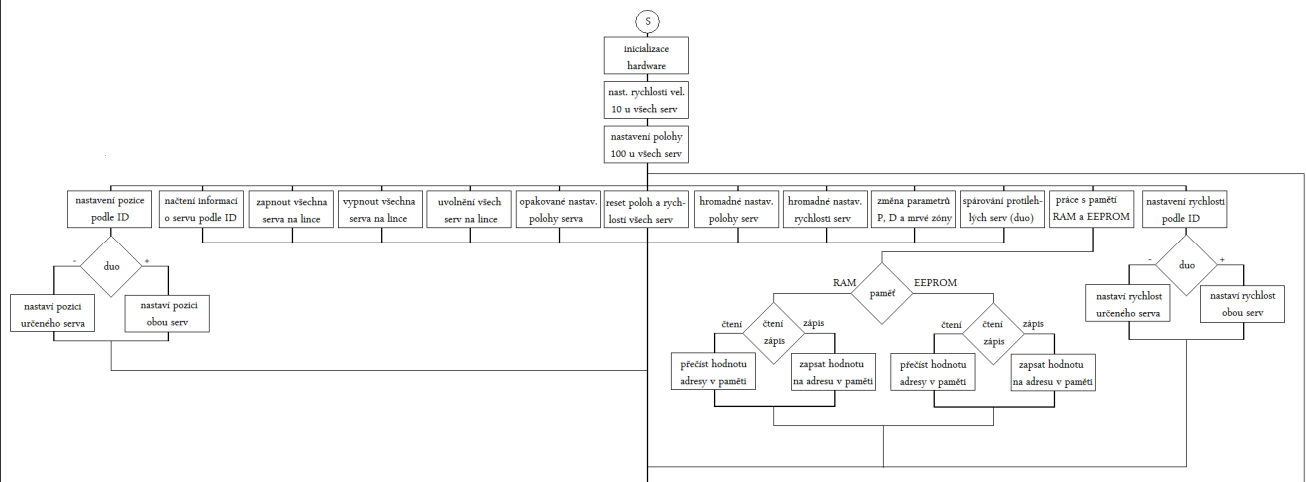
šířka: 8 bitů

stop bity: 1

parita: žádná

### 4.2.2 Formát příkazů

Příkazy pro sériový mód vycházejí z první varianty řešení, aby bylo ovládání velmi podobné. Vyskytuje se zde navíc celá řada příkazů a program mnohem lépe komunikuje s obsluhou. V případě výskytu chyby při provádění příkazu (není připojeno žádné servo nebo příkaz nebyl proveden) je na terminál odesláno hlášení „Error“.



Obr. 27: Zjednodušený vývojový diagram pro druhou variantu

**Příkazy pro nastavení hodnot jednoho serva (podle ID):**

*sxy,a,hodnota*

s – nutná identifikace příkazu (s - servo)

xy – číslo serva (00 až 99)

a – událost: p - poloha, r - rychlost, cti - načtení polohy serva (čárka a parametr hodnota se poté neuvádějí)

hodnota – hodnota v rozsahu 001-255 pro rychlost a 010 až 190 pro úhel, je nutné brát ohled na to, že hodnotu 1 je nutné napsat jako 001. Hodnoty jsou uváděny dekadicky. Pokud je zadané hodnota mimo rozsah, příkaz se neprovede.

V případě vyčítání informací o pozici a rychlosti serva se odpověď odesílá stejným komunikačním prostředkem (terminálem). Tvar odpovědi je:

*sab,pcderfgh*

*s* – identifikace (servo)

*ab* – číslo serva

*p* – identifikátor polohy

*cde* – hodnota polohy (úhel natočení)

*r* – identifikátor rychlosti

*fgh* – velikost rychlosti

př.:

*s01,p149,r010*

Interpretace: servo číslo 01 má polohu 149° a rychlost je 10.

*tst,ab,cde*

*tst* – režim testování (opakování)

*ab* – číslo serva

*cde* – počet opakování

Tento příkaz začne provádět automatické hýbání se servem od polohy 010 do 190 v předem určeném počtu opakování. Rychlost je potřeba nastavit před zadáním tohoto příkazu. Tento příkaz nelze přerušit a procesor nebude reagovat na žádný další, dokud jej nedokončí.

**Hromadné příkazy:**

*p,xyz*

*p* – hromadné nastavení polohy

*xyz* – úhel natočení

*r,xyz*

*r* – hromadné nastavení rychlosti

*xyz* – hodnota rychlosti

*on* – zapne pohyby všech serv a vykoná případné nevykonané nastavení polohy

*off* – vypne pohyby všech serv (po tomto příkazu můžeme provést řadu nastavení u jednotlivých serv a příkazem *on* je hromadně provedeme), i když serva nereagují na povel ke změně polohy, stále drží svoji pozici

*res* - resetuje nastavení všech serv do výchozí polohy a rychlosti

### Speciální příkazy:

#### VAROVÁNÍ:

Většina těchto příkazů platí pro všechna serva připojená na sběrnici. Zásahem do paměti EEPROM, zapíšeme tuto informaci do všech připojených serv. Pokud máme více zařízení na sběrnici, nejsme schopni provést správný kontrolní součet a po restartu nebudou nastavovat polohu a bude nutná jejich opětovná inicializace. Ze stejného důvodu se nesmí provádět inicializace, pokud máme více serv připojených na sběrnici.

Pokud chceme provádět změny v paměti EEPROM při více připojených zařízeních, můžeme tak činit pouze hromadně přes paměť RAM (adresa 0x80 až 0x9E). Zde si již nemusíme hlídat kontrolní součty, tyto za nás automaticky dopočítá řídicí jednotka serva.

*duo* - spáruje nadefinovaná serva (u dvou serv provádí stejný pohyb se stejnou rychlostí)

*er*, *ew* – čtení (*er*) a zápis (*ew*) do paměti EEPROM, zde se posílají čísla v šestnáctkové soustavě; př.: *er,00* vyčte hodnotu na pozici 0x00 (pro standardní nastavení je to hodnota 0x50), *ew,00,50* na pozici 0x00 zapíše hodnotu 0x50.  
POZOR: Pro *ew* je nutné, aby bylo na sběrnici připojeno jen jedno servo!

*mr, mw* – čtení (*mr*) a zápis (*mw*) do paměti RAM, stejná syntaxe příkazů jako pro *er* a *ew*.

*parx* – mění kontrolní parametry (P, D a mrtvou zónu), podle tabulky v příloze Parametr Set 1,2 nebo 3, *x* je číslo parametru (1,2 nebo 3) – zde není čárka v příkazu

*init,xzy* – provede inicializaci serva připojeného na sběrnici. O průběhu informuje pomocí terminálu a po provedení příkazu je možnost inicializovat další (ID se inkrementuje automaticky) nebo skončit. *xzy* – je startovní ID inicializace. POZOR: Je nutné, aby bylo na sběrnici připojeno jen jedno servo!

*chck* – provede výpočet kontrolního součtu paměti EEPROM a jeho uložení na příslušnou pozici. POZOR: Je nutné, aby bylo na sběrnici připojeno jen jedno servo!

#### 4.2.3 Zhodnocení

Sériový mód se jeví jako jeden nejvhodnějších kandidátů na řízení robota. Poskytuje nám dostatečný komfort ovládání, neplýtvá časem procesoru a je jednoduchý na rozšíření. Jeho nevýhodou oproti první variantě je nutná samostatná inicializace každého serva a správné spárování ID a čísla serva v hlavním programu (jinak hrozí kolize a špatná funkce programu).

Inicializace může být například vyřešena samostatným pinem pro inicializaci serv na řídicí jednotce.

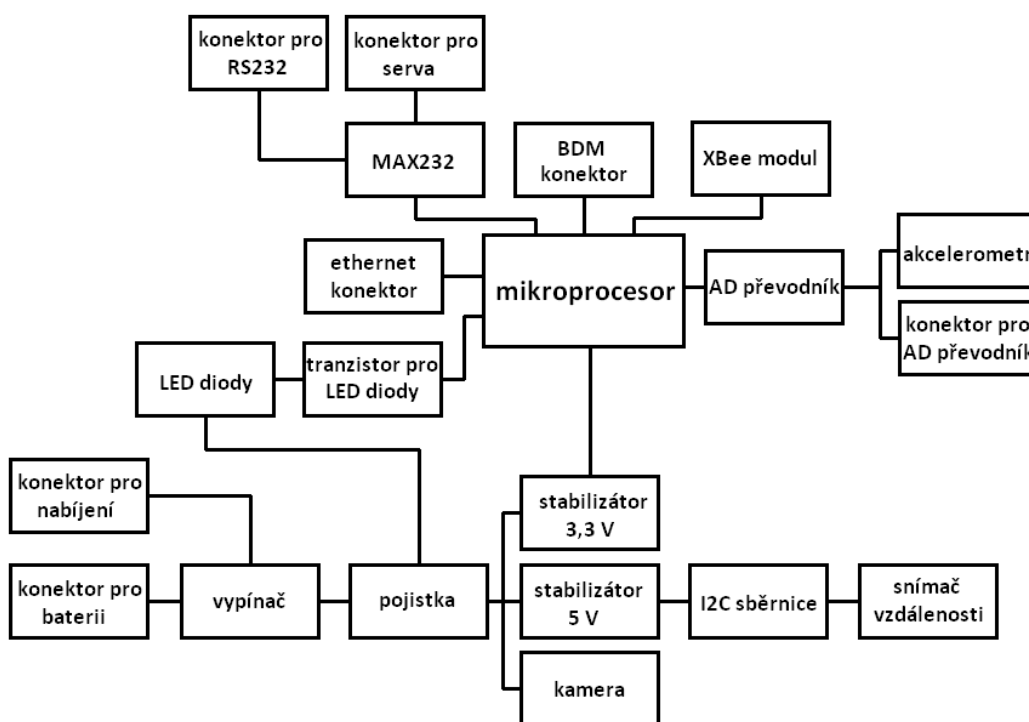
Správné spárování je možné vyřešit v těle hlavního programu – buď uživatel zadá číslo serva a jeho pozici nebo bude programem veden, aby danému servu přiřadil konkrétní ID. Vše může být vizuálně ověřeno jeho pohybem.

## 5. NÁVRH DPS

DPS byla navržena v programu EAGLE 4.16r2, Light Edition.

Aby mohla jednotka komunikovat s okolím a získávat z něj informace, obsahuje DPS následující prvky: sběrnici I2C, 3xUART (jeden pro serva, jeden pro řízení a jeden volný), ethernet a modul XBee pro bezdrátovou komunikaci; dále je připojen akcelerometr, ultrazvukový snímač vzdálenosti, LED diody pro osvětlení scény, bezdrátová kamera pro přenos obrazu a zvuku k operátorovi.

Při návrhu desky plošných spojů byl limit velikosti určen rozměry robota. Bylo nutné, aby DPS v žádném pohybu robota nemohla být poškozena. Dále aby byly připojitelné periferie ze spodní strany DPS. Celá deska byla rozdělena do čtyř základních segmentů, které byly ze spodní strany vylity mědi (zejména kvůli rušení). Jedná se o napájecí část, část s mikroprocesorem (konektory, programovací konektor, I2C, ethernet port a RS232), část bezdrátového modulu XBee a část analogově-číslicového převodníku s porty pro rozšíření o další prvky a akcelerometr. Velikost celé desky je 10 x 6 cm [š x v].



Obr. 28: Blokové schéma navržené DPS

výbava jednotek	původní	navržená
procesor	ATmega128	ColdFire V2
frekvence	16 MHz	60 MHz
architektura	8bit, RISC	32bit, RISC
počet ADC	8	8
ovládání	IR, ovladač	XBee, 2,4 GHz
max. počet serv	24	128
další sběrnice	I2C	I2C, UART, ethernet
připojená zařízení	piezoměnič	2x LED, akcelerometr, ultrazvukový snímač vzdálenosti, bezdrátová kamera

Tab. 3: Srovnání původní a navržené koncepce

### 5.1 SOUČÁSTKY A KONEKTORY

Velikosti odporů a kondenzátorů se pohybují od 0603 do 1205. Velikost 0603 byla použita pro odpory a je asi na samé hraně pájitelnosti v „amatérských“ podmínkách.

Na desce jsou umístěny konektory pro připojení baterie, napájecí konektor ze sítě (slouží pouze pro nabíjení baterie – v tomto režimu je napájení robota odpojeno), konektor RJ45 (ethernet), stereo jack 2,1 mm pro připojení na sériovou linku (zde je nutné provést drátovou propojku a tou zvolit, jestli bude na výstupu logická úroveň TTL nebo RS232), I2C sběrnice s napájením a pět volných portů s analogově-digitálním převodníkem.

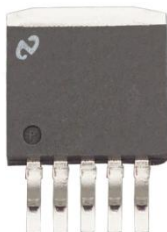
### 5.2 NAPÁJECÍ ČÁST

Protože se na desce vyskytují součástky rozdílných napěťových úrovní, byly použity dva stabilizátory napětí. Jeden na 3,3 V (na který je připojena většina zařízení) a na 5V (pro zařízení na I2C sběrnici).

Na hlavní stabilizátor byly kladeny velké nároky. Výstupní napětí 3,3 V, malý úbytek napětí (napětí na baterii je od 7,5 V do 4,5 V), velký výstupní proud

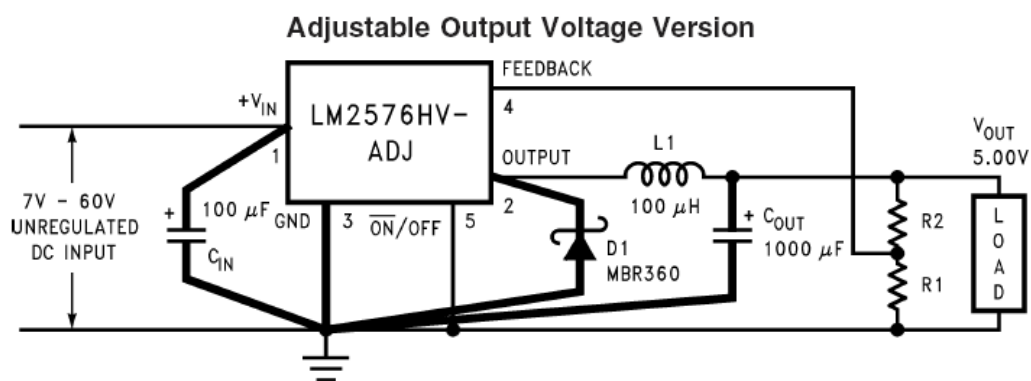
(spotřeba samotné elektroniky bez serv může být špičkově až 0,8 A). Největší problém byl s nízkým úbytkem napětí.

### 5.2.1 Stabilizátor LM2576-ADJ D2PAK



Jedná se o stabilizátor s volitelným výstupním napětím a maximálním výstupním proudem 3 A. Výstupní napětí se volí pomocí zpětnovazebního pinu (feedback), který se připojí mezi odpor R1 a R2. Odpor R1 se volí v rozmezí 1 kΩ až 5 kΩ

Obr. 29: LM2576



Obr. 30: Schéma zapojení LM2576-ADJ D2PAK [12]

$$V_{OUT} = V_{REF} \left( 1 + \frac{R_2}{R_1} \right)$$

$$R_2 = R_1 \left( \frac{V_{OUT}}{V_{REF}} - 1 \right)$$

Rovnice 1: Výpočet hodnot odporu pro požadované výstupní napětí [12]

kde  $V_{REF} = 1,23 \text{ V}$

V našem případě byla zvolena hodnota  $R_1 = 1,2 \text{ k}\Omega$ , výpočtem nám vyšel druhý odpor na hodnotu  $R_2 = 2 \text{ k}\Omega$ .

### 5.2.2 Stabilizátor LF50CDT



LF50CDT je „nízkoubytkový“ stabilizátor na 5 V s výstupním proudem až 1 A. Na DPS je určen výhradně pro napájení sběrnice I2C, takže výstupní proud je dostatečný.

Obr. 31: LF50CDT

### 5.2.3 Další konektory

V napájecí části se dále nachází konektor pro napájení bezdrátové kamery. Vzhledem k tomu, že kamera obsahuje vlastní stabilizátor napětí, je připojena rovnou na baterii (za pojistkou a vypínačem). Pro odfiltrování rušivého signálu ze servomotorů je na vstupu připojen LC filtr.

Kamera slouží operátorovi, aby mohl sledovat prostředí, ve kterém se robot pohybuje a správně ho navigovat. V našem případě byl důraz kladen malou



hmotnost a velikost. Kamera byla zakoupena v GM Electronic od neznámého výrobce za 1.550,- Kč. I přesto, že podle obalu má výrobce certifikát od společnosti TÚV a je certifikován dle ISO 9001:2000, dochází ke vzájemnému rušení s modulem XBee a dosah tohoto modulu je snížen na zlomek vzdálenosti, kterou dosáhne, když je kamera vypnutá.

Obr. 32: Kit bezdrátové kamery a přijímače

Druhým konektorem je napájení a ovládání osvětlení LED diod. Jedná se o dvě sériově zapojené led diody, které jsou řízeny MOSFET tranzistorem LR120N. Tento tranzistor je připojen na PWM výstup mikroprocesoru a intenzita LED se ovládá pomocí šířky pulzu napětí.

### 5.3 MIKROPROCESOR A OKOLÍ

#### 5.3.1 Programovací konektor

Pro programování mikroprocesoru je na DPS osazen BDM konektor, který byl pro potřeby UAMT FEKT VUTB zjednodušen na 2x 10 pinů.

#### 5.3.2 Sériová linka

Jako převodník napěťových úrovní mezi TTL a RS232 logikou je použit MAX3232CSE. Jedná se o dvoukanálový převodník, u kterého je využit jen jeden kanál (pro řízení servomotorů) a druhý je volný. Na ten je možné připojit procesor a konektor jack 2,1 mm pro další zařízení.

#### 5.3.3 Ethernetový konektor

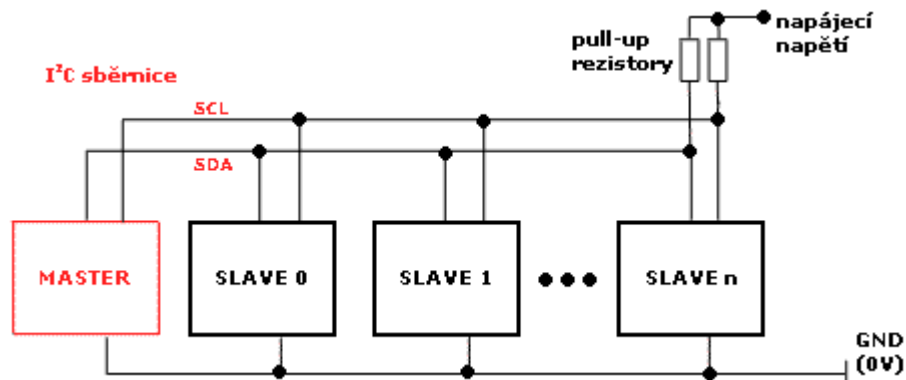
Pro další rozšíření a komunikaci je na DPS osazen konektor RJ45 pro připojení robota pomocí síťové karty nebo jiného prvku do sítě LAN. Komunikace prostřednictvím tohoto konektoru není předmětem diplomové práce, takže v rámci programu není kód pro jeho obsluhu.

#### 5.3.4 Sběrnice I2C [13]

I2C nebo IIC (Inter Integrated Circuit) je komunikační sběrnice, kterou navrhl pro svá zařízení Philips. Z důvodu patentu někteří výrobci nehovoří o I2C, ale např. o TWI (Two Wire Serial Interface – dvoudrátové sériové rozhraní).

Firma Philips řešila rozvod datových kabelů ve svých zařízeních (televize, video, atd.) a ukázalo se, že je mnohem ekonomičtější propojit všechny moduly pomocí jedné sběrnice a individuálně s nimi komunikovat.

Sběrnice IIC je dvou vodičové datové propojení mezi jedním nebo několika procesory (MASTER) a speciálními periferními součástkami (SLAVES). Všechny součástky jsou připojeny na téže sběrnici a jsou cíleně vybírány svými adresami. Adresy i data se přenášejí stejnými vodiči. Sběrnice umožňuje velmi jednoduché propojení mezi několika integrovanými obvody a bezproblémové dodatečné rozšiřování.



Obr. 33: Ukázka zapojení sběrnice I2C

Sběrnice využívá datovou linku SDA (serial data line) a linku hodinového vstupu SCL (seriál clock line). Obě linky se používají jako obousměrné. Jsou vybaveny pull-up odpory, takže sběrnice je v klidovém režimu v log. 1.

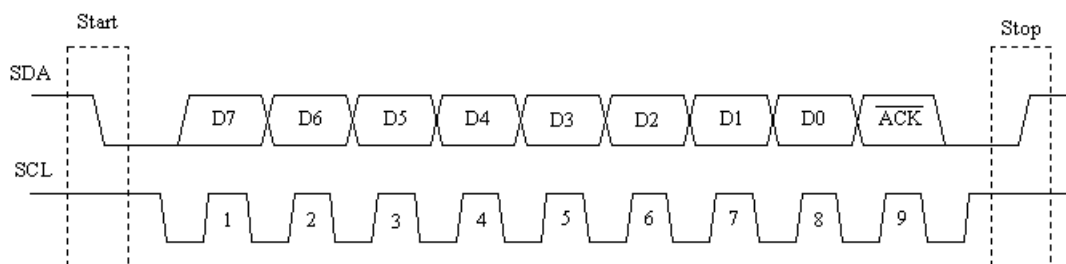
V základní verzi lze na sběrnici připojit teoreticky až 128 zařízení (7bitové adresování), rozšířené verzi až 1024 zařízení (10bitové adresování). Při připojování zařízení na sběrnici jsme ale omezeni nastavitelností jeho adresy, která se dělí na statickou a dynamickou. Statickou část je stejná pro daný typ součástky a dynamickou si můžeme zvolit (zapojením pinů, jumperů, atd.). V konečném důsledku máme k dispozici třeba jen 3 bity na adresování a tím pádem k dispozici 8 zařízení stejného typu. V některých případech je adresa dána napevno od výroby a může mít jen jeden takový obvod na každé sběrnici.

Přenosová rychlost je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100 kHz. V novějších verzích sběrnice to může být 400 kHz až 1 MHz. Při konstrukci musíme vzít v potaz, že rychlost sběrnice bude vždy omezena nejpomalejší součástkou na sběrnici. Čím vyšší komunikační frekvence, tím nižší hodnoty pull-up rezistorů musí být. Pro základní verzi (100 kHz) dostačují hodnoty 2k2, které byly použity v tomto návrhu.

*předdefinované stavy sběrnice:*

- *klidový stav:* SDA i SCL jsou v log. 1

- *start bit*: SDA je MASTERem stažena na log. 0, SCL je v log. 1
- *stop bit*: SDA jde z log. 0 do log. 1, SCL zůstává v log. 1
- *přenos dat*: Vysílač přivede na linku SDA osm datových bitů v taktu podle impulsů SCL, které vysílá MASTER. Přenos začíná nejvyšším bitem.
- *potvrzení*: Příjímač potvrdí příjem bytu nízkou úrovní SDA a čeká dokud MASTER nevyšle devátý impuls na SCL. Poté se může přijímat další byte. Ukončení přenosu se indikuje neexistencí potvrzení. Ukončení přenosu se dosahuje podmínkou stop.



Popis k obrázku 33 : Obr. 34: Příklad nejjednoduššího přenosu dat

- MASTER vyrobí start bit
- v bitech 7 až 1 přeneše adresu součástky a v bitu 0 zvolí směr dat (0 - zápis, 1 - čtení)
- SLAVE potvrdí příjem bitem ACK
- MASTER vyšle datový byte, který SLAVE opět potvrdí
- nyní může MASTER spojení ukončit (stop bit) nebo bude se stejným zařízením v komunikaci pokračovat

#### 5.3.4.1 Měření vzdálenosti

Pro měření vzdálenosti byl na hlavu robota implementován ultrazvukový snímač vzdálenosti SRF10. Jedná se o destičku s dvojicí ultrazvukového přijímače a vysílače, který je připojen k procesoru přes I2C sběrnici.



Obr. 35: Snímač vzdálenosti

Měřicí rozsah je 6 cm až 6 m. Hodnoty naměřené vzdálenosti můžeme ze zařízení obdržet v centimetrech, palcích nebo mikrosekundách. V případě, že nepotřebujeme využít celý rozsah měření, je možné snížit zesílení modulu (a tím i zkrátit dobu měření).

#### 5.4 BEZDRÁTOVÁ KOMUNIKACE

Na desce je osazena patice pro připojení bezdrátového modulu XBee, který s mikroprocesorem komunikuje pomocí rozhraní RS232 na úrovni TTL logiky (není nutný konvertor). Modul komunikuje s procesorem rychlostí 38.400 baudů, 8 datových bitů, 1 stop bit a žádná parita.

Modul XBee pracuje v bezlicenčním pásmu 2,4 GHz a používá se zejména pro průmyslové aplikace. Jeho výhodou je velikost, cena a podpora od výrobce.



XBee se vyrábí ve dvou variantách – XBee a XBee-PRO. Podle údajů výrobce má XBee dosah až 120 m na volném prostranství a XBee-PRO až 1,6 km. Odráží se ale ve spotřebě; XBee má 40 mA a XBee-PRO 295 mA špičkově při vysílání dat (přijímání je na stejné úrovni 40 mA).

**Obr. 36: Modul XBee**

Modul je osazen procesorem HCS08GT60 od Freescale, a proto neslouží pouze jako modul pro bezdrátovou komunikaci, ale výrobce jej rozšířil o řadu funkcí, které procesor podporuje. Nalezneme na něm tedy analogově-digitální převodník a výstupy portů, které můžeme programovat. Pro naše potřeby máme vše potřebné v hlavním mikroprocesoru, a proto nejsou tyto piny připojeny.

Protože vytváří výrobce dobrou podporu pro tento modul, je k dispozici celá řada ukázkových programů a jsou jen s drobnými úpravami (vyrovnávací paměť pro příchozí znaky) použity pro tuto práci.

## 5.5 ANALOGOVĚ DIGITÁLNÍ PŘEVODNÍK (ADC)

Procesor obsahuje jeden port (PORT AN), který může sloužit jako analogově digitální převodník. V našem zapojení je pět pinů volně přístupných pro další rozšíření robota a tři jsou připojeny na akcelerometr.

### 5.5.1 Akcelerometr MMA7260QT

MMA7260QT je tři-osý akcelerometr od společnosti Freescale. Pomocí něj jsme schopni měřit náklon v každé ze tří os souřadného systému.



Měřicí rozsah si můžeme vybrat ze čtyř variant (volba se provádí dvěma piny) – 1,5 g / 2 g / 4 g nebo 6 g. Maximální citlivost je 800 mV při rozsahu 1,5 g.

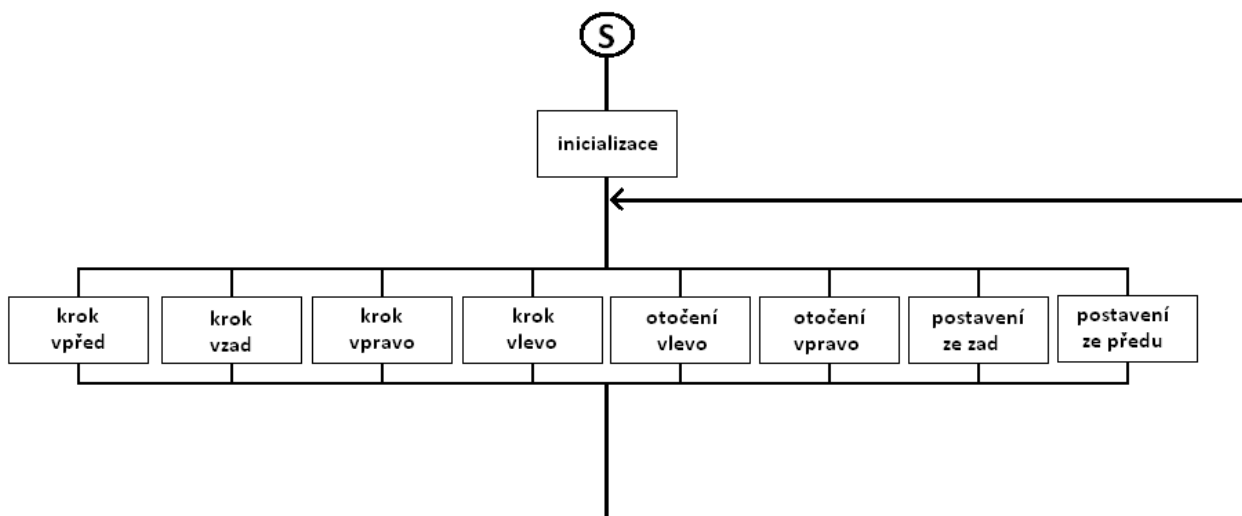
Obr. 37: MMA7260QT [14]

Hlavní využití modulu je k ochraně robota, a aby operátor věděl, po jak rovném terénu robot kráčí. Ochrana robota může být provedena algoritmem, který v případě nežádaného náklonu sníží těžiště robota dolů (dřepem) a zvýší tak jeho stabilitu. Dalším prvkem ochrany může být automatické sklápění hlavy robota, které obsahuje elektroniku, jež by se vlivem nárazu mohla poškodit. Pokud robot padá na zem je automaticky protisměrem sklopena jeho hlava.

## 5.6 OVLÁDÁNÍ

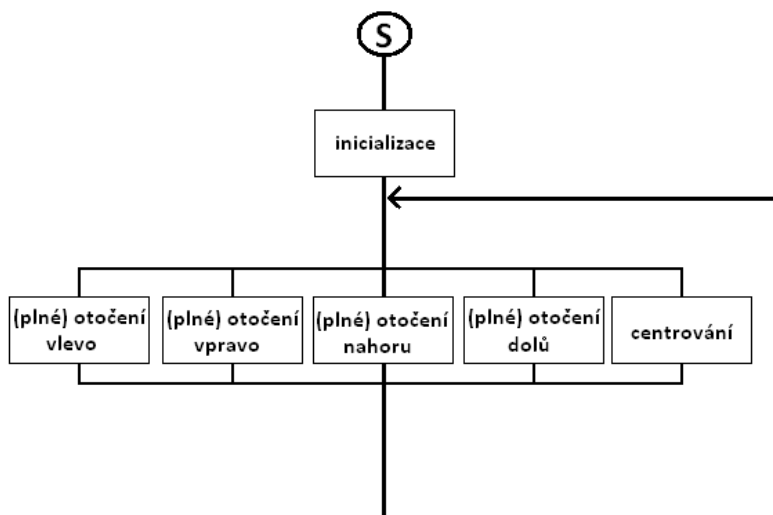
Robot se ovládá pomocí nekódovaných příkazů přímo ze sériové linky. Každý znak nebo řetězec znaků musí být ukončen kódem 13 (ENTER) a nesmí přesáhnout 10 bajtů včetně ukončovacího znaku. Ovládání je rozděleno do čtyř základních kategorií – ovládání těla (pohyb), hlavy (kamery), intenzity osvětlení a snímače vzdálenosti.

V rámci ovládání těla umí robot vykonat osm příkazů – krok dopředu (klávesová zkratka „w“), dozadu (s), vlevo (a), vpravo (d), otočení vlevo (q), otočení vpravo (e), vstávání ze zad (f) a vstávání zepředu (r). Opakování pohybů se aktivuje, když programu zašleme znak velkým písmenem. Následuje blokový diagram příkazů.



Obr. 38: Blokovaný diagram pohybu těla

Ovládání hlavy má pět příkazů – otočení vlevo (*j*), vpravo (*l*), nahoru (*i*), dolů (*m*), vycentrování (*k*). Dále jsou v kódu obsaženy jejich maximální ekvivalenty – plná výchylka vpravo (*L*), vlevo (*J*), atd. (písemné zkratky jsou velkými písmeny). Blokovaný diagram je uveden na obrázku.



Obr. 39: Blokovaný diagram ovládání hlavy

Intenzita osvětlení se ovládá v jedenácti krocích. Od „0“ až po „9“ a nejvyšší hodnota je označena znakem „-“.

## 6. PROGRAMY

### 6.1 PROGRAM PRO ŘÍZENÍ ROBOTA

Program pro mikroprocesor byl napsán ve vývojovém prostředí CodeWarrior ColdFire Development Studio IDE version 5.9.0 build 3024 v jazyce C.

Při návrhu řídicího algoritmu pro mikroprocesor umístění na DPS byl kladen důraz na maximální kompatibilitu s původním programem. Bylo to hlavně z důvodu zjednodušení řídicích algoritmů. Při řízení pohybů se využívá stejného zápisu pozic jednotlivých serv, jako používá výrobce ve svých skriptech. Stačí tedy řadu čísel zkopírovat, vložit do algoritmu a můžeme tak robota rychle naučit pohyby, které byly vytvořeny pro původní Robonovu. Jediný problém, který nastal, bylo jiné přiřazení ID čísel servům podle nového systému. Toto řeší převodní tabulka, kterou program automaticky využívá při zasílání polohy a není nutné na rozdíl od ID čísla myslet.

V rámci programu je korekční tabulka, která má za cíl upravovat výchozí polohy servomotorů, které mohou být způsobeny jejich vůlí, tolerancí výrobního postupu, atd. Tato tabulka se automaticky aplikuje na požadované pohyby.

Největší problém, který se vyskytl, bylo posílání poloh jednotlivým servům. Nakonec byly vytvořeny dva algoritmy, které mají za cíl postupně serva natočit do požadované pozice:

#### 6.1.1 Algorismus „MOVE“

Sevu se pošle hodnota cílové pozice a servo se automaticky nastavenou rychlostí začne natáčet. Nevýhodou v tomto případě je, že algoritmus neřeší vzdálenosti do cíle vzhledem k jednotlivým servům a to může způsobit nestabilitu robota.

#### 6.1.2 Algorismus „FAZE“

V případě tohoto algoritmu posíláme informaci o počáteční pozici, koncové pozici, rychlosti a počtů „fází“. Algorismus provede rozdělení vzdáleností pohybu

každého serva na daný počet fází a posílá servům postupně cílovou pozici navýšenou o přírůstek tak dlouho, dokud serva nedorazí do cíle (zde se průběžně využívá algoritmu „*move*“). Celý pohyb serv potom vypadá jako „pumpování“. Tento algoritmus byl vytvořen hlavně pro vstávání robota ze země.

### 6.1.3 Zhodnocení provedení

Ač byla maximální snaha o kompatibilitu s původním programem, výsledek se nezdařil na 100%. Nepodařilo se rozklíčovat, kdy je dobré použít který algoritmus a je tedy nutné při přejímání nového pohybu sledovat jeho průběh a korigovat řídicí algoritmy. V několika případech bylo dokonce nutné příkazy ručně upravit.

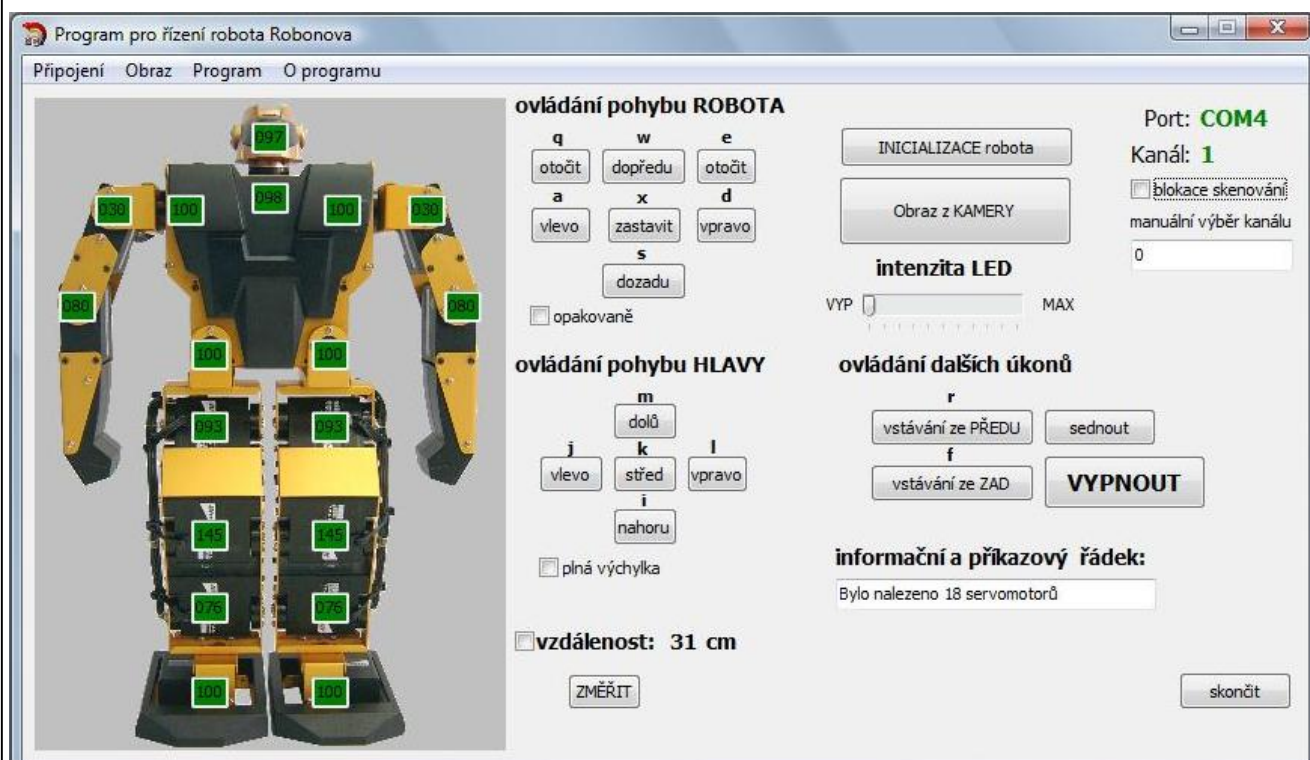
Celý program obsahuje zhruba 1.000 řádků kódu, který se stará o řízení celého robota a jednotlivé procedury jsou komentovány přímo v programu. V programu se již pohybujeme na hranici omezení pro bezlicenční verzi programu CodeWarrior, a proto byly navrženy spíše základní pohyby robota. Celý problém by se mohl vyřešit přesunutím jednotlivých příkazů do paměti flash, odkud by se podle potřeby vyčítaly.

## 6.2 PROGRAM PRO OVLÁDÁNÍ PŘES POČÍTAČ

Protože komunikace pomocí sériové linky není dostatečně komfortní a operátor by se ve velkém množství informací o polohách jednotlivých serv nevyznal, byl vytvořen program, který má za úkol stav a údaje od robota vizualizovat a zároveň umožnit jeho ovládání.

Program pro ovládání přes počítač byl napsán v programu CodeGear C++ Builder 2009 version 12.0.3170.16989 v jazyce C++. Ovládání COM portu včetně přerušení využívá napsanou knihovnu od Thierryho Schneidera z roku 2002, která je volně šiřitelná a použitelná. Pro získávání obrazu z kamery je použita knihovna VideoLab od společnosti Mitov Software, která je pro nekomerční použití zdarma.

Program byl navržen tak, aby umožňoval přehledné a pohodlné ovládání celého robota a zařízení k němu připojených. Komunikační protokol mezi robotem a mikroprocesorem je bez kódování. Vše se posílá formou prostého textu s minimálním počtem znaků – často jedním. U řady prvků se zobrazuje nápovědný text, který aktivujeme najetím kurzoru od myši.



Obr. 40: Hlavní nabídka programu pro ovládání robota

Hlavní nabídka se skládá z několika částí.

### 6.2.1 Hlavní menu

V hlavním menu nalezneme čtyři nabídky:

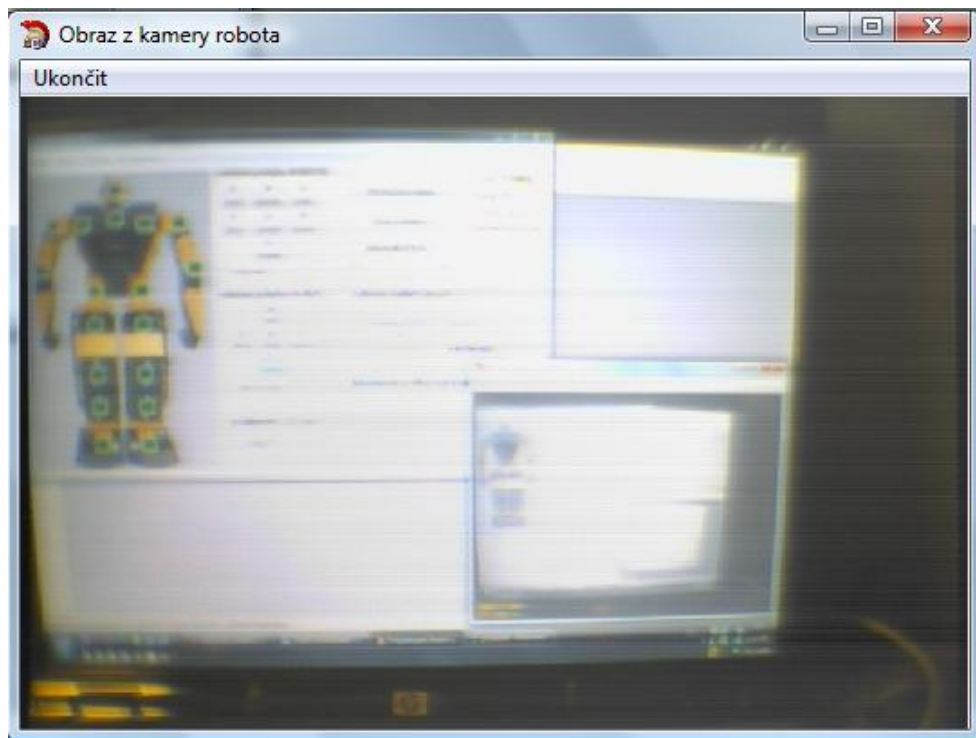
#### 6.2.1.1 Připojení

Pokud stiskneme možnost „Nastavení portu“, volíme v nabídce COM port (jedna až osm), ke kterému je připojen bezdrátový modul XBee. Po výběru portu a stisknutí tlačítka OK začne automatické vyhledávání robota. V případě, že vybereme COM port, který v počítači není obsažen, program se na něj nepřipojí

a nezačne vyhledávání. Pomocí nabídky se můžeme kdykoliv z portu odpojit nebo se s ním opětovně spojit.

#### 6.2.1.2 *Obraz*

V této nabídce máme možnost zvolit buď „Spustit“ nebo „Zavřít“. Pomocí těchto příkazů buď otevřeme, nebo uzavřeme nové okno s náhledem z kamery. Otevření náhledu provedeme i tlačítkem „Obraz z KAMERY“, které je umístěno na ovládací ploše. O stavu nás informuje „Informační a příkazový řádek“.



Obr. 41: Okno s náhledem z kamery

#### 6.2.1.3 *Program*

Nabídka slouží k ukončení programu. Stejnou volbu provedeme i tlačítkem „Skončit“ z ovládací plochy.

#### 6.2.1.4 *O programu*

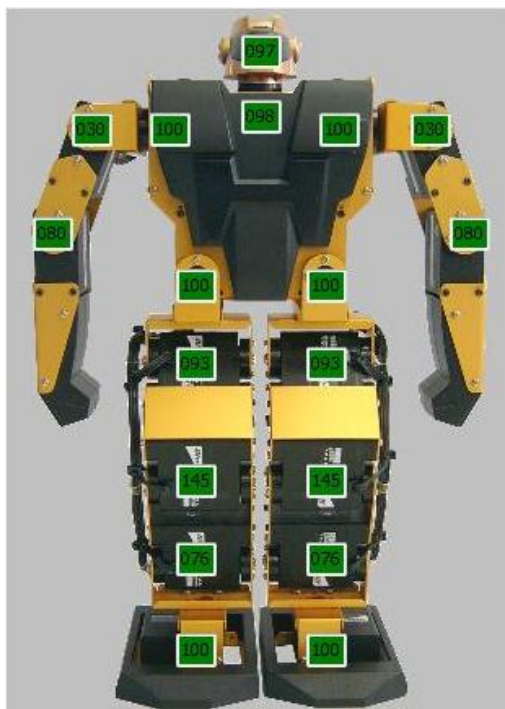
Volba otevře okno s informacemi o práci a verzi programu.



Obr. 42: Informace o práci a verzi

### 6.2.2 Obrázek s robotem

Obrázek slouží k představě o nastavení poloh jednotlivých serv. Na každém místě, kde se nachází servo, je umístěna buňka s číslem, které reprezentuje jeho natočení. V případě pohybu robota jsou buňky průběžně aktualizovány a probíhá přepisování jejich aktuální polohy. Kliknutím na buňku můžeme měnit natočení daného serva. Do buňky můžeme zadávat pouze čísla. Po vystoupení z buňky nebo stisknutím klávesy enter se provede odeslání pohybu na servo, které vykoná požadovaný pohyb. V případě, že se bude požadavek mimo povolený rozsah serva (10 - 190), příkaz se neprovede a do buňky se vepíše původní hodnota.



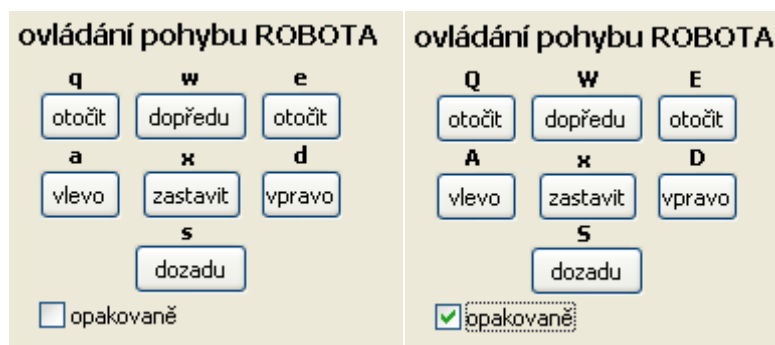
Obr. 43: Reprezentace natočení serv v programu

Při spuštění programu nebo při ztrátě spojení s robotem je pozadí buněk červené. Označuje to, že servo není připojeno. Po připojení k robotu se provede inicializace všech serv a ty, které byly nalezeny, změní pozadí barvy na zelenou.

### 6.2.3 Ovládání robota

#### 6.2.3.1 Ovládání pohybu

Pomocí tlačítek můžeme robotovi posílat předem nadefinované příkazy. Jedná se o pohyb dopředu, dozadu, vlevo, vpravo, otočit vlevo a otočit vpravo. Všechny tyto příkazy mají i cyklické varianty, které se opakují, dokud nepřijde příkaz pro ukončení. Všechna tlačítka mají klávesovou zkratku, které je nad nimi znázorněna a v případě stisku tohoto tlačítka na klávesnici se příkaz také provede. Cyklické varianty se od normálního příkazu liší vždy velkým písmenem.



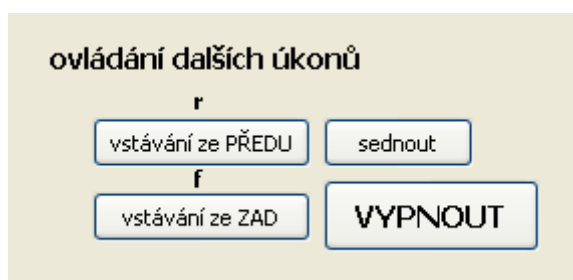
Obr. 44: Při stisku zaškrťovacího tlačítka se u cyklických příkazů změni popisky na velká písmena

### 6.2.3.2 Ovládání natočení hlavy

Ovládání je stejné jako u pohybu těla. Opět zde nalezneme tlačítka s ekvivalentem v podobě písmen. Rozdíl je v tom, že v případě zaslání velkého písmena nebo zaškrtnutí políčka „plná výchylka“ dojde k pohybu do maximálního otočení serva.

### 6.2.3.3 Ostatní úkony

Ostatní úkony v sobě zahrnují tlačítka pro vstávání robota z přední a zadní části, sednutí (stoupnutí) a vypnutí (zapnutí) robota.



Obr. 45: Ovládání dalších úkonů

Popisy a příkazy, které tlačítka „sednout“ a „VYPNOUT“ provedou po jejich stisku, se měni v závislosti na aktuálním stavu. V případě, že je robot již vypnut, změni se popis na „ZAPNOUT“. Tlačítko „sednout“ se změni na „vstát“. Pokud je robot vypnut, nevykovává žádné příkazy mimo zapnutí. Serva jsou v tomto módu

v „uvolněném“ režimu a můžeme s nimi rukou libovolně pohybovat. Po zapnutí se plnou rychlostí vrátí do původní polohy.

Tlačítko „*INICIALIZACE robota*“ slouží pro obnovení výchozí polohy všech serv a zjištění, která serva jsou připojena a která ne. Inicializace se provede vždy, když program vyhledá robota.

#### 6.2.3.4 Ovládání osvětlení

Intenzitu led diod je možné ovládat pomocí posuvného pole v jedenácti krocích (0 až 10). V případě výpadku signálu se při připojení provede automatické nastavení intenzity poosvětlení podle polohy jezdce.



Obr. 46: Ovládání intenzity osvětlení

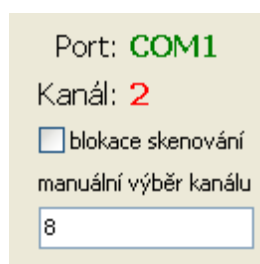
#### 6.2.4 Informační části

Informační části jsou podle potřeby rozmístěny na ovládacím panelu. První z nich se nachází v pravém horním rohu a informuje nás o připojení na port COM a kanál XBee modulu. V případě, že nápis je červené barvy, nejsme připojeni. Zelená nás naopak informuje, že připojeni jsme. Číslo COM portu se vybírá v hlavním menu, po jeho připojení se automaticky začne prohledávat všech 16 kanálů XBee (00 - 15).

Prohledávání probíhá tak, že v každém kanálu se vyšle desetkrát *obnovovací* řetězec „*Rob*“ a očekává se odpověď ve tvaru „*h\_ack*“. Pokud dorazí, je spojení v pořádku a dalších deset znaků se vyšle za 10 sekund. *Obnovovací* řetězec se vyšle každých deset sekund v případě, že nedošlo k jiné komunikaci mezi programem a robotem (pokud ano, vysílá se deset sekund od poslední komunikace, aby nedocházelo ke zbytečnému vytěžování linky). Pokud je robot ve vyhledávacím režimu, přestane vykonávat všechny funkce, čeká na *obnovovací* řetězec a prohledává všechny kanály. Robot o tomto režimu upozorňuje blikajícími LED diodami. Aby se eliminovala situace, kdy se vyhledáváním programy nenajdou, tak

ovládací program v počítači kanály inkrementuje a program v robotu dekrementuje. Navíc má každý program jinou dobu pro prohledávání jednoho kanálu.

V případě potřeby můžeme zablokovat zaškrtnutím políčkem prohledávání kanálů. To má za následek, že program bude robotu vyhledávat jen na aktuálním kanálu. Tento můžeme změnit okénkem „*manuální výběr kanálu*“.



Obr. 47: Informace o připojeném portu a kanálu Xbee

Mezi další informační prvky patří informační a příkazový řádek. Pomocí něj nám program oznamuje prováděné úkony nebo výsledky jednotlivých požadavků (vyhledávání robota, inicializace kamery, počet nalezených serv, ztráty signálu robota, provádění pohybu atd.).

informační a příkazový řádek:

Program připraven

Obr. 48: Informační řádek

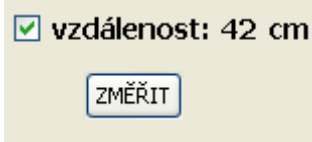
Důležitou funkcí tohoto řádku je, že pomocí něj můžeme poslat jakýkoliv text na sériovou linku. Jediné omezení je, že text nesmí překročit devět znaků. Je to dáno vyrovnávací pamětí, kterou má v sobě vytvořenu Xbee modul. V případě, že se pokusíme odeslat více znaků, budeme na tuto skutečnost upozorněni a odeslání se neprovede.

informační a příkazový řádek:

Nemůže být odesláno víc než 9 znaků!!!

Obr. 49: Program neodešle víc než devět znaků

Posledním informačním prvkem je blok, který nás informuje o naměřené vzdálenosti od hlavy robota. Změření vzdálenosti provedeme tlačítkem „ZMĚŘIT“. Pokud potřebujeme měřit vzdálenost kontinuálně, je možné zaškrtnout políčko u vzdálenosti. Měření v tomto režimu se provádí dvakrát za sekundu.



Obr. 50: Ovládání měření vzdálenosti

## 7. ZÁVĚR

Stavebnice Robonova-I je velmi povedená a praktická ukázka možností současné techniky. Určitě bych ji doporučil každému, kdo se o robotiku zajímá, na vyzkoušení a zjištění, co všechno lze udělat. Obsah balení postačí pro základní uvedení do problematiky řízení humanoidních robotů (stabilita, chování na různých površích, atd.).

Hlavním problémem, který zhoršuje rozšíření je cena. V současné době se stavebnice prodává za cenu kolem 1.000 USD, což je pro většinu lidí v ČR víc než jeden měsíční příjem. Další faktor ovlivňující prodej je špatná podpora a dostupnost ze stran prodejců (a možná i samotného výrobce) na území ČR.

Hlavní důvod vlastní koncepce elektroniky je větší podpora připojitelných zařízení, rychlejší procesor a efektivnější program pro řízení robota. Navrhovaný procesor má řádově vyšší výkon, několiknásobně větší paměť RAM i EEPROM (Flash) a větší počet integrovaných obvodů (UART, ethernet, časovač, AD převodník).

Byly otestovány dva druhy řízení. Jeden pro základní a rozšířený mód a druhý pro sériový mód. První varianta byla ve finále zavržena z důvodu veliké zátěže pro procesor a nemožnost jednoduchého rozšíření celé desky. Druhá varianta, i přes některé drobné nedostatky, byla vyhodnocena jako perspektivní pro toto řízení. Všechny nedostatky jsme schopni ošetřit v rámci hlavního programu (čtení nastavené rychlosti serv) nebo pomocí externího obvodu pro inicializaci serv (či vytvoření pinu pro jejich inicializaci přímo na řídicí jednotce).

Problém, který se nepodařilo vyřešit, je vzájemné rušení bezdrátové kamery a modulu XBee. Pravděpodobně bude nutné najít na trhu jiný výrobek, u kterého tento problém nenastane.

Podle zadání bylo cílem práce navrhnout výkonnější řídicí jednotku. V tomto ohledu práce splnila a i překročila zadání. Nad rámec byly přidány LED diody pro osvětlení scény, akcelerometr, ultrazvukový snímač vzdálenosti a program, pomocí kterého můžeme robota ovládat přes počítač.

V rámci šíření jména fakulty, podpory zájmu dětí o robotiku a názorný příklad spojení studia s praxí byl robot oficiálním maskotem Olympijských her na ZŠ Otnice, které se konaly na podzim roku 2008. Aby to nebyla jen formální záležitost, byla dětem celé školy předvedena robotem nachystaná sestava. Poté mělo několik žáků možnost si ovládání vyzkoušet. Ohlas si je možné přečíst v příloze.

Dne 23.4.2009 se na Vysoké škole báňské – Technické univerzitě Ostrava konal 14. ročník mezinárodní soutěže pro studenty bakalářského a magisterského studia vysokých škol. Soutěže se zúčastnilo 44 prací z celkem 10 fakult z České republiky, Slovenska a Polska.

Tato práce, která v té době nebyla doplněna o program pro ovládání přes počítač a další prvky, obsadila v sekci S6 - Mecha & auto-tronika druhé místo.



Obr. 51: Autoři tří nejlepších prací v sekci Mecha & auto-tronika na STOČ 2009

## 8. SEZNAM PŘÍLOH

- [A] CD s programem pro mikroprocesor a ovládání pomocí počítače, elektrické schéma pro program Eagle, fotky robota, manuály k servomotrům a touto prací, níže uvedené přílohy v elektronické podobě
- [B] Rozložení paměti EEPROM servomotoru [7]
- [C] Rozložení paměti servomotoru [7]
- [D] Detailní popis příkazů pro servomotor [7]
- [E] Diplom – robot jako maskot
- [F] Robot Robonova – zhodnocení organizátora Olympijských her
- [G] Diplom ze STOČ 2009
- [H] Schéma elektrického zapojení obvodu
- [I] Deska plošných spojů – všechny vrstvy
- [J] Deska plošných spojů – spodní vrstva
- [K] Deska plošných spojů – horní vrstva
- [L] Foto robota zepředu
- [M] Foto robota zezadu

## 9. POUŽITÁ LITERATURA

- [1] Florián, T.: EL displej, Semestrální projekt 1, 2007/2008
- [2] Atmel, Product Card, 23.12.2008  
[on-line] [http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=2018](http://www.atmel.com/dyn/products/product_card.asp?part_id=2018)
- [3] Hitec, RoboNova Manual, 26.12.2008  
RoboNova Manual-Eng-V1.50.pdf
- [4] Hitec, HSR-8498HB, 26.12.2008  
[on-line] <http://www.hitecrcd.com/servos/show?name=HSR-8498HB>
- [5] Ibbotson, R.: <http://web.ukonline.co.uk/r.ibbotson>  
[on-line] [web.ukonline.co.uk/r.ibbotson/files/HSR8498HB%20Servo.pdf](http://web.ukonline.co.uk/r.ibbotson/files/HSR8498HB%20Servo.pdf)
- [6] Hitec, HMI Servo Programmer, 26.12.2008  
[on-line]  
[http://www.robonova.com/manager/control/board.php?go=3&tableName=board\\_14&behavior=read&seqid=12&field=&keyword=&openWin=](http://www.robonova.com/manager/control/board.php?go=3&tableName=board_14&behavior=read&seqid=12&field=&keyword=&openWin=)
- [7] neznámý autor, HSR Serial Communication, 26.12.2008  
[on-line]  
[www.staff.uni-bayreuth.de/~btp918/cmt2007/geraete/hitec\\_hsr/HSR\\_Serial\\_communication.pdf](http://www.staff.uni-bayreuth.de/~btp918/cmt2007/geraete/hitec_hsr/HSR_Serial_communication.pdf)
- [8] Žalud, L.: Součásti mobilních robotů, prezentace z předmětu MRBT, 5.5.2008
- [9] HOŘEJŠÍ MODEL, s.r.o., Serva, 28.12.2008  
[on-line] <http://www.horejsi.cz/clanky/serva/serva.htm>
- [10] RCmania, Jak a proč prodloužit život zpětnovazebním potenciometrům v servech, 28.12.2008  
[on-line] <http://www.rcmania.cz/2002/hankey/potak.htm>
- [11] MAX232 Manual, 29.12.2008  
[on-line] [focus.ti.com/lit/ds/symlink/max232.pdf](http://focus.ti.com/lit/ds/symlink/max232.pdf)
- [12] LM2576 Datasheet, 12.5.2009  
[on-line] <http://www.national.com/mpf/LM/LM2576.html#Datasheet>

- [13] Florián, T.: Návrh ZigBee zařízení – přístupový terminál/alfanumerický display s tlačítky, Bakalářská práce, FEKT VUT v BRNĚ 2007
- [14] MMA7260QT Datasheet, 13.5.2009  
[on-line]  
[http://www.freescale.com/files/sensors/doc/data\\_sheet/MMA7260QT.pdf](http://www.freescale.com/files/sensors/doc/data_sheet/MMA7260QT.pdf)

## SEZNAM OBRÁZKŮ

Obr. 1: Robonova-I.....	11
Obr. 2: Obsah balení.....	11
Obr. 3: Stavba robota – nohy a ruce .....	12
Obr. 4: Stavba robota - tělo .....	12
Obr. 5: Řídící jednotka [3] .....	13
Obr. 6: Mechanická funkce serva [10] .....	14
Obr. 7: Elektrická funkce serva [10] .....	14
Obr. 8: Druhy servomotorů [3] .....	14
Obr. 9: Schéma zapojení a řízení serva [7] .....	16
Obr. 10: Vliv parametru D – hodnoty 1, 2, 3 [6].....	17
Obr. 11: Vliv parametru P – hodnoty 1, 2, 3 [6] .....	17
Obr. 12: Vliv parametru „mrtvé pásmo“ – 1, 2, 3 [6].....	17
Obr. 13: Šířka signálů ( $\mu\text{s}$ ) a k nim odpovídající poloha serva ve stupních (deg) [5]	18
Obr. 14: Průběh rozšířeného módu [5] .....	18
Obr. 15: Zapojení pro sériovou komunikaci [7].....	19
Obr. 16: HMI Servo Programmer [6] .....	19
Obr. 17: Komunikační protokol [7].....	20
Obr. 18: Nastavení nulových pozic [3].....	22
Obr. 19: Nastavení serv na požadovanou polohu [3] .....	22
Obr. 20: Schéma obvodu pro signalizaci slabé baterie [3] .....	24
Obr. 21: ColdFire Roadmap .....	28
Obr. 22: M52233DEMO BOARD [1].....	28
Obr. 23: Schéma modulu [1].....	29
Obr. 24: Blokové schéma mikroprocesorů řady MCF5223x .....	31
Obr. 25: Zapojení převodníku úrovní [11].....	32
Obr. 26: Zjednodušený vývojový diagram pro první variantu.....	34
Obr. 27: Zjednodušený vývojový diagram pro druhou variantu .....	36
Obr. 28: Blokové schéma navržené DPS .....	40
Obr. 29: LM2576.....	42

Obr. 30: Schéma zapojení LM2576-ADJ D2PAK [12].....	42
Obr. 31: LF50CDT .....	43
Obr. 32: Kit bezdrátové kamery a přijímače .....	43
Obr. 33: Ukázka zapojení sběrnice I2C .....	45
Obr. 34: Příklad nejjednoduššího přenosu dat .....	46
Obr. 35: Snímač vzdálenosti .....	46
Obr. 36: Modul XBee .....	47
Obr. 37: MMA7260QT [14] .....	48
Obr. 38: Blokový diagram pohybu těla .....	49
Obr. 39: Blokový diagram ovládání hlavy .....	49
Obr. 40: Hlavní nabídka programu pro ovládání robota.....	52
Obr. 41: Okno s náhledem z kamery .....	53
Obr. 42: Informace o práci a verzi.....	54
Obr. 43: Reprezentace natočení serv v programu .....	55
Obr. 44: Při stisku zaškrtačacího tlačítka se u cyklických příkazů změni popisky na velká písmena .....	56
Obr. 45: Ovládání dalších úkonů.....	56
Obr. 46: Ovládání intenzity osvětlení.....	57
Obr. 47: Informace o připojeném portu a kanálu XBee .....	58
Obr. 48: Informační řádek.....	58
Obr. 49: Program neodešle víc než devět znaků .....	58
Obr. 50: Ovládání měření vzdálenosti.....	59
Obr. 51: Autoři tří nejlepších prací v sekci Mecha & auto-tronika na STOČ 2009...	61

## SEZNAM ROVNIC

Rov. 1: Výpočet kontrolního součtu pro EEPROM [7].....	15
Rov. 2: Podmínka kontrolního součtu pro EEPROM [7] .....	15
Rov. 3: Vzorec pro výpočet úhlu natočení serva.....	18
Rov. 4: Podmínka pro kontrolní součet příkazu v sériovém módu [7].....	20
Rov. 5: Výpočet kontrolního součtu příkazu v sériovém módu [7] .....	20

## SEZNAM TABULEK

Tab. 1: Seznam příkazů pro sériovou komunikaci [7].....	21
Tab. 2: Rodina mikroprocesorů řady MCF5223x .....	30
Tab. 3: Srovnání původní a navržené koncepce.....	41

# Příloha B

## Rozložení paměti EEPROM servomotoru

<i>Default settings (**)</i>			
Address	8498SG	5498SG	5980SG / 5990TG
0x00	0x50	0x50	0x14
0x01	0xB4	0xB4	0x64
0x02	0x03	0x04	0x04
0x03	0x1E	0x0A	0x05
0x04	0x01	0x02	0x02
0x05	0x01	0x01	0x01
0x06	0xFF	0xFF	0xFF
0x07	0x00	0xFF	0x00
0x08	0x00	0xB4	0x00
0x09	0x02	0x02	0x02
0x0A	0x26	0x26	0x26
0x0B	0x09	0x09	0x09
0x0C	0x92	0x92	0x92
0x0D	0x00	0x00	0x00
0x0E	0x10	0x10	0x10
0x0F	0x03	0x03	0x03
0x10	0xF0	0xF0	0xF0
0x11	0x05	0x05	0x05
0x12	0x1C	0xDC	0xD0
0x13	0xB4	0xB2	0xB4
0x14	0xB4	0xB9	0xB4
0x15	0x13	0x13	0x13
0x16	0x88	0x88	0x88
0x17	0x00	0x00	0x00
0x18	0x00	0x00	0x00
0x19	0x05	0x05	0x05
0x1A	0xDC	0xDC	0xDC
0x1B	0x29	0x29	0x29
0x1C	0x28	0x28	0x28
0x1D	0xD2	0xD2	0xD2
0x1E	0x05	0x0A	0x0A
0x1F	0x54	0x34	0x1E
0x20	0x78	0x78	0x78
0x21	0x04	0x05	0x04
0x22	0x32	0x0F	0x07
0x23	0x01	0x02	0x02
0x24	0x3C	0x3C	0x0A
0x25	0xA0	0xA0	0x50
0x26	0x02	0x03	0x04
0x27	0x0A	0x05	0x03
0x28	0x01	0x02	0x02
0x29	0x00	0x00	0x00
0x2A	0x0A	0x0A	0x0A
0x2B	0x02	0x02	0x02
0x2C	0x2F	0x03	0xB3

P-gain

Parameter set 1

Dead-zone

D-gain

D-gain ? (not reloaded at parameter change)

Speed

Center position (\*\*)

(550) Minimum position limit.

(2450) Maximum position limit.

(1500) Center position ?

Left limit (\*\*)

Right limit (\*\*)

(1500) Center position ?

Direction: 0x29: forward, 0x28: reverse (\*\*)

P-gain

Parameter set 2

Dead-zone

D-gain

D-gain ? (not reloaded at parameter change)

P-gain

Parameter set 3

Dead-zone

D-gain

D-gain ? (not reloaded at parameter change)

ID

EEPROM Checksum (\*)

Není-li uvedeno jinak, dvoubajtové hodnoty mají nejvýznamnější bajt uložen jako první.

\* *výpočet:*

checksum = Sum[0x00:0x2B] % 256

*podmínka:*

Sum[0:0x2C] % 256 == 0

\*\* Nebylo zkoušeno. Převzato z HMI-Servo programmer v 1.0.2

## Příloha C

### Rozložení paměti servomotoru

<i>Address</i>	
0x06	(low/high) Current trajectory set-point
0x07	
0x80	Correspond to EEPROM addresses 0x00..0x1E.
...	Cells are loaded from EEPROM at power-on.
0x9E	
0xC3	Speed (sometimes updated from 0x86)
0xA5	Target position
0xA6	
0xA7	Actual position (Not reliable if motor is released ### ? ###)
0xA8	
0xA9	Target position – actual position
0xAA	
0xC9	Go / stop

Není-li uvedeno jinak, dvoubajtové hodnoty mají nejvýznamnější bajt uložen jako první.

# Příloha D

## Detailní popis příkazů pro ovládání serva

Command	0..0x7F (id)	Set servo #id target position
Param1	pos_high	This allows to set the position of a single servo.
Param2	pos_low	
Return1	0x00	<b>Note:</b> This command re-enables ALL released motors on the bus (see command 0xFE).
Return2	0x00	

Command	0xE1	Read EEPROM
Param1	addr	Read one byte from the EEPROM.
Param2	0x00	
Return1	data	Readable addresses are 0..0xFF. Addresses >0x2C always read as 0x00. Known EEPROM addresses are tabulated below.
Return2	0x00	

Command	0xE2	Write EEPROM
Param1	addr	Write one byte to the EEPROM.
Param2	data	
Return1	0x03	Writable addresses are 0..0x2C. When writing addresses > 0x2C every 3 <sup>rd</sup> write attempt will fail, and return 0x00,0x00 ### ? ###
Return2	0x03	

Command	0xE3	Read memory
Param1	addr	Read one byte from the controllers volatile memory.
Param2	0x00	
Return1	data	Readable addresses are 0..0xFF. Known memory addresses are tabulated below.
Return2	0x03	

Command	0xE4	Write memory
Param1	addr	Write one byte to the controllers volatile memory.
Param2	data	
Return1	0x03	The addressable memory range is 0..0xFF, but not all addresses are writable. When a write attempt fails the controller returns 0x00,0x00. ### <i>TODO: Tabulate writable addresses.</i> ###
Return2	0x03	

Command	0xE5	Read position
Param1	0x00	Read the current position.
Param2	0x00	
Return1	pos_high	Use command 0xE9 to address multiple motors on one bus.
Return2	pos_low	

Command	0xE6	Set target position
Param1	pos_high	This sets the target position for all motors on the bus.
Param2	pos_low	
Return1	0x00	Use command 0..0x7F to address multiple motors.
Return2	0x00	

Command	0xE7	Read version & id
Param1	0x00	
Param2	0x00	
Return1	version	
Return2	id	

Command	0xE8	<b>Read pulsewidth &amp; voltage</b>
Param1	0x00	Reads the driving pulsewidth applied to the motor and the supply voltage.
Param2	0x00	
Return1	pulsewidth	Voltage scale = [0.03522 V]
Return2	voltage	

Command	0xE9	<b>Set servo #id speed &amp; read position</b>
Param1	0..0x7F (id)	This command sets the speed of a single servo on the bus and reads its current position. Valid speed range is 1..255.
Param2	speed	
Return1	pos_high	
Return2	pos_low	

Command	0xEA	<b>Select control parameter set</b>
Param1	0x00	This command sets the control parameters (P-gain,D-gain,Dead-band) according to the parameter sets 1-3 stored in the EEPROM. Values other than 1-3 always select parameter set 3.
Param2	1 / 2 / 3	
Return1	0x03	Parameter set 1 is loaded at power-on.
Return2	0x06	

Command	0xEB	<b>Set go/stop</b>
Param1	1 / 0	Sending a 0 will suspend subsequent changes of the target position until sending a value >0. This command does NOT disable the position control.
Param2	0x00	
Return1	0x03	To disable the position control mechanism use command 0xEF.
Return2	0x06	

Command	0xEF	<b>Release</b>
Param1	0x00	This command stops the servos position control mechanism. The joint can be moved by hand afterwards.
Param2	0x00	
Return1	0x03	<b>Note:</b> Any following move command on ANY servo on the bus will re-enable all servos. A re-enabled servo will move to the last set target position with highest possible speed.
Return2	0x06	

# Příloha E

Diplom – robot jako maskot

Základní škola Otnice, okres Vyškov

## NÁŠ MASKOT

V úterý 30. 9. 2008 jsme uspořádali první ročník Otnických olympijských her. Naším maskotem byl ROBOT.  
Uměl cvičit lépe než mnozí z nás.



# Příloha F

## Robot Robonova

- maskot Olympijských her Otnice 2008

Když mně můj syn Tomáš poprvé ukázal svého nového „kamaráda“ robota Robonovu, hned mně, jako správnou učitelku, napadlo jeho využití na naší základní škole. V době školních vzdělávacích programů a zavádění alternativních forem výuky bylo jasné, že by se mohl stát vhodnou pomůckou při motivaci a následně pak také při výuce dětí.

Protože učím tělesnou výchovu a o prázdninách byly v Pekingu letní olympijské hry, tak jsem po jejich skončení přemýšlela, jak je uspořádat na naší škole. Jelikož jsem měla představu, že se do olympiády zapojí celá škola, hledala jsem inspiraci ve starověkém Řecku, kdy se závodilo nejenom ve sportu, ale také ve výtvarné výchově a literatuře.

Když jsem přemýšlela, jak děti do akce vtáhnout a co nejlépe je motivovat, což je v současné době vlastně to nejsložitější, napadl mě robot jako náš olympijský maskot.

Ihned po začátku školního roku jsme se s ostatními vyučujícími na škole dohodli na společném postupu a motivaci dětí ve svých předmětech. Já jsem zpracovala oznámení, kdy se bude olympiáda konat, za jakých podmínek a přesný plán olympijských her. Tam se děti poprvé dověděly, že maskotem olympijských her bude robot. Všechny děti byly hrozně zvědavé a neuměly si robota představit. Pak jsem přinesla jeho fotografie a plakátky. Některé děti se ho pokoušely nakreslit do olympijské soutěže, jiné ho v rámci informatiky vzaly

jako podklad pro soutěžní diplomy, ze kterých jsme pak vybírali ty nejlepší. Ty byly později velmi zdařilou odměnou pro vítěze jednotlivých soutěží.

Ovšem děti se ani zdaleka nechtěly spokojit s plakátkem maskota našich olympijských her. Byla jsem tedy nucena jim ho v rámci olympiády ukázat. Tomáš ho v den olympijských her přijel dětem ukázat. Žáci všech tříd a ročníků s úžasem sledovali, co všechno robot dokáže. Už tady jsme si s mnohými slíbili, že to nebudu poslední setkání s robotem, ale že se s ním uvidíme při některé hodině gymnastiky, kdy nám robot bude své nejpovedenější prvky předvádět a my se mu budeme snažit ve cvičení vyrovnat.

Po skončení Olympijských her Otnice 2008 žáci naší školy průběh celé akce hodnotili. Byla jsem ráda, že se všem olympiáda líbila a kritických připomínek bylo pomálu. Ovšem kdo získal absolutorium, byl náš maskot – robot Robonova. U dětí to vyhrál na celé čáře. A nejen u dětí. Paní učitelka fyziky si už taky zamlouvala robota k prezentaci.

Mgr. Drahomíra Floriánová

# Příloha G

Diplom ze STOČ 2009



The image shows a diploma certificate for the 2009 Student Technical Olympiad (STOČ). The certificate is issued by the Faculty of Electrical and Communication Technologies of the Brno University of Technology. It recognizes Tomáš Florián for his 2nd place in the Mechatronics and Robotics section. The certificate features logos of various sponsors including CID International, ControlTech, ELVAC, Freescale, Ingeteam, Microsys, Pantek, and HETU. It also includes the logos of the four participating universities: VSB-TECHNICAL UNIVERSITY OF OSTRAVA, UNIVERSITY OF SOUTH BOHEMIA, UNIVERSITY OF JYVÄSKYLÄ, and UNIVERSITY OF ŽILINA. The certificate is signed by Prof. Ing. Radim Farana, CSc., Dean of the Faculty, and Prof. Dr. RNDr. Lubomír Smutný, STOČ FS Guarantor.

**Vysoká škola báňská – Technická univerzita Ostrava**  
**Studentská tvůrčí a odborná činnost**

**Tomáš Florián**  
Řídící jednotka pro humanoidní robot

**DIPLOM**  
za **2. místo** v sekci  
**Mecha & auto – tronika**

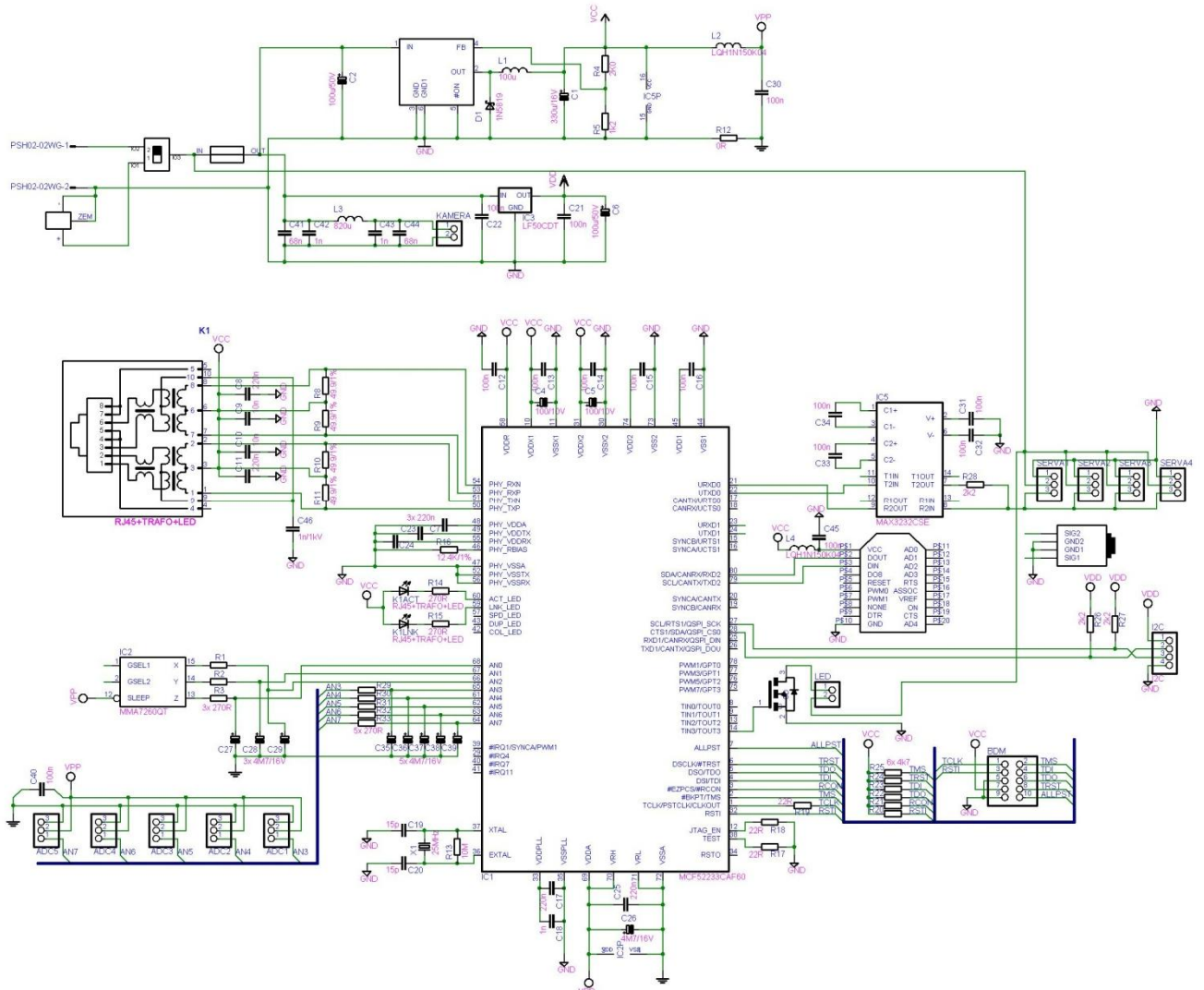
*Radim Farana*  
prof. Ing. Radim Farana, CSc.  
děkan FS VŠB-TUO

**STOČ '09**

*Lubomír Smutný*  
prof. Dr. RNDr. Lubomír Smutný  
garant STOČ FS

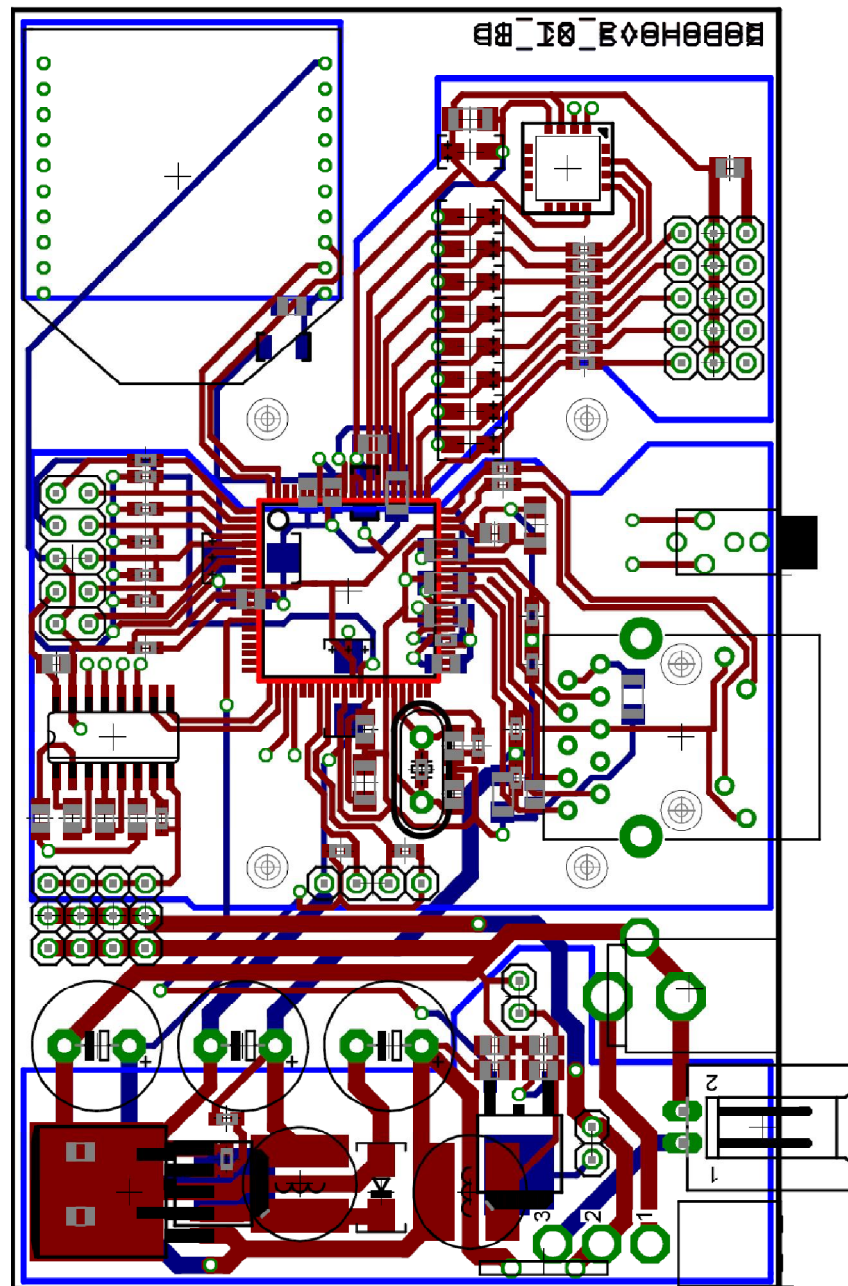
# Příloha H

## Schéma elektrického zapojení obvodu



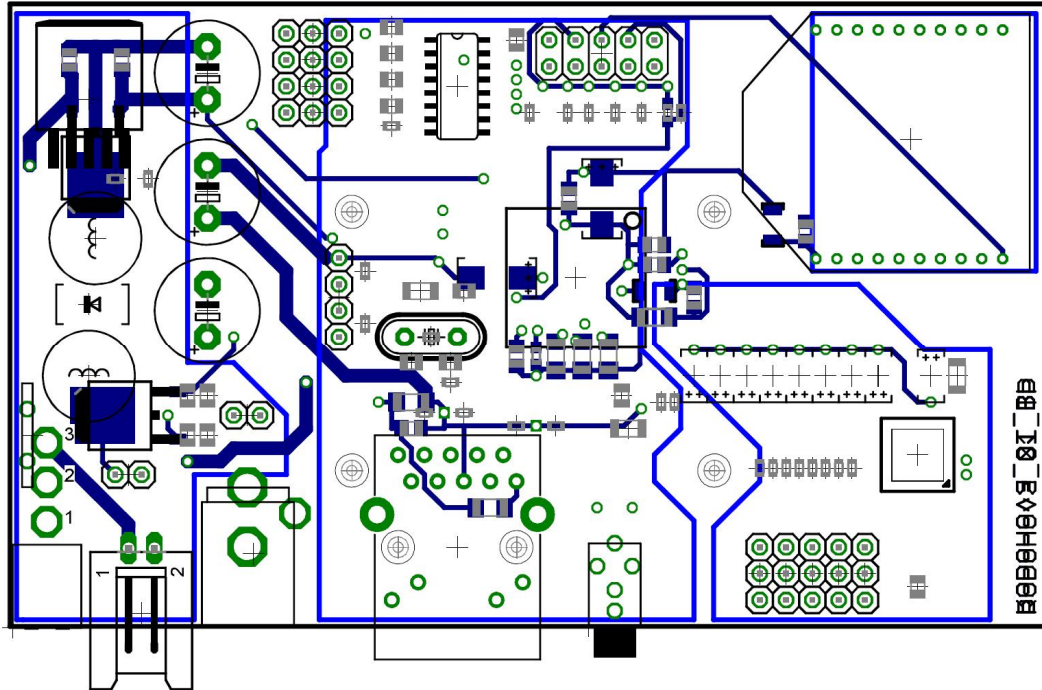
# Příloha I

Deska plošných spojů – všechny vrstvy



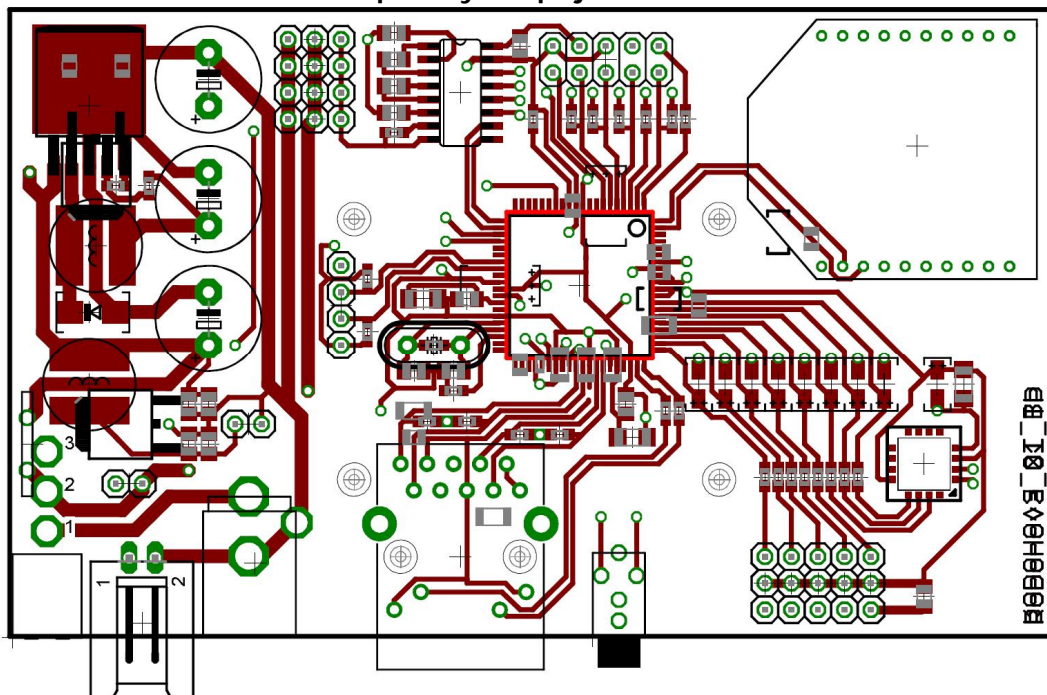
# Příloha J

Deska plošných spojů – spodní vrstva



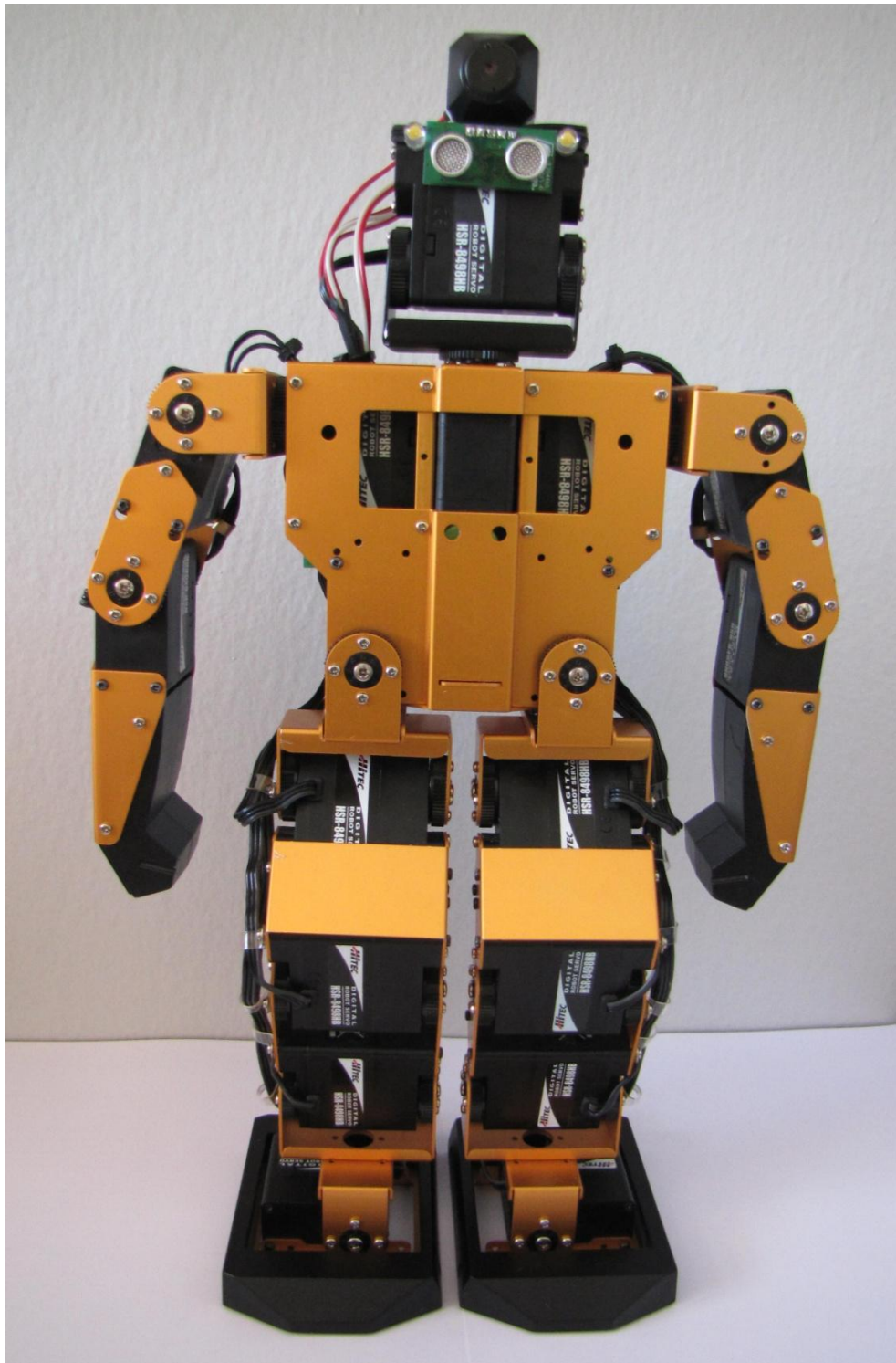
# Příloha K

Deska plošných spojů – horní vrstva



# Příloha L

Foto robota zepředu



# Příloha N

Foto robota zezadu

