

Návrh a realizace programovatelného mobilního robota

Design and development of programmable mobile robotic system

Bc. Pavel Neckář

Diplomová práce
2009



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav elektrotechniky a měření
akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Pavel NECKÁŘ**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Návrh a realizace programovatelného mobilního robotického systému**

Zásady pro vypracování:

1. Rozbor vlastností pohonných jednotek HITEC HS 322 a ROBOTIS RX 64.
2. Návrh řídicích jednotek na alternativní součástkové základně mikro-kontrolérů AT89C2051 a CM 2+.
3. Návrh řídicích algoritmů na základě programového vybavení WIN SOS a ROBOT-TERMINAL.
4. Návrh a realizace funkčního vzoru pro podvozek mobilního robota.
5. Implementace podpůrných algoritmů podvozku mobilního robota.
6. Ověření funkčnosti podvozku mobilního robota.

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. NOVÁK, Petr. **MOBILNÍ ROBOTY – pohony, senzory, řízení**. 1. vyd. Praha : BEN, 2005. 248 s. ISBN 807300111-1.
2. HLAVÁČ V., SEDLÁČEK M. **Zpracování signálů a obrazů**. Praha: Vydavatelství ČVUT, 2000. ISBN 80-01-03110-1.
3. SONKA M., HLAVAC V., BOYLE R. **Image Processing, Analysis, and Machine Vision**. 2. vyd. PWS Publishing, Pacific Grove, 1999. ISBN 0-534-95393-X.
4. SURÝNEK, Tomáš. **Určení vzdálenosti cíle hloubkoměrným principem se strukturovaným světlem**. Zlín, 2008. 63 s. Vedoucí diplomové práce Ing. Mgr. Milan Kvasnica, CSc.
5. PETR, Ondřej. **Řídicí systém mobilního dvoustopého robota**. Zlín, 2006. 67 s. Vedoucí diplomové práce Mgr. Ing. Milan Kvasnica, CSc .
6. OHAREK, Petr. **Ovládání internetové komunikační kamery se dvěma stupni volnosti** . Zlín, 2006. 74 s. Vedoucí diplomové práce PETR, Ondřej. **Řídicí systém mobilního dvoustopého robota**. Zlín, 2006. 67 s. Vedoucí diplomové práce Mgr. Ing. Milan Kvasnica, CSc .
7. KVASNICA M. **Head Joystick and Interactive Positioning for the Wheelchair** Procding of the ICOST 2003. In 1th IEEE International Conference on Smart Homes and Health Telematics. Paris, 2003, France.
8. **Úprava modelářského servomechanismu [online]** Dostupný z WWW:(http://www.sweb.cz/e78/clanky/uprava_serva_cla.htm)

Vedoucí diplomové práce:

doc. Ing. Milan Kvasnica, CSc.

Ústav elektrotechniky a měření

Datum zadání diplomové práce:

20. února 2009

Termín odevzdání diplomové práce:

22. května 2009

Ve Zlíně dne 20. února 2009



prof. Ing. Vladimír Vašek, CSc.

děkan



doc. RNDr. Vojtěch Křesálek, CSc.

ředitel ústavu

ABSTRAKT

Cílem práce je návrh a realizace dvou rozdílných mobilních robotických systémů s využitím pohonných jednotek HS 322 a RX 64. Obsahuje návrh řídicích modulů, komunikačních bloků a uživatelského rozhraní pro ovládání obou pohonných systémů. Programové vybavení bylo vyvinuto v prostředí SharpDevelop v jazyce C#. Práce je rozdělena do dvou částí teoretické a praktické. První část je věnovaná teoretickému úvodu do problematiky pohonných jednotek a řídicích systémů. V praktické části je uveden postup při vývoji řídicích algoritmů a programového vybavení. Součástí práce jsou dva funkční vzory mobilních robotických systémů.

Klíčová slova: mobilní robot, pohonná jednotka HS 322, pohonná jednotka RX 64, vývojové prostředí SharpDevelop, programovací jazyk C#, řídicí jednotka CM 2+, mikrokontrolér AT89C2051.

ABSTRACT

The aim of this thesis is the design and development of two various mobile robot systems, using drive units HS 322 and RX 64. The thesis contains the description of control units, communication packets and user interface project for both actuators. Software facility was developed in SharpDevelop environment in C# language. The thesis is divided into two parts - theoretical and practical one. The first one is dedicated to theoretical introduction to drive units and control systems. In practical part are introduced control algorithms and supporting software. The parts of diploma thesis are two prototypes of mobile robotic platforms with control equipments.

Keywords: mobile robot, drive unit HS 322, drive unit RX 64, development environment SharpDevelop, C# programming language, control unit CM 2+, micro-controller AT89C2051.

Poděkování, motto

Tímto vyjadřuji poděkování vedoucímu diplomové práce Mgr. Ing. Milanu Kvasnicovi, CSc za svědomité vedení diplomové práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.
V případě publikace výsledků budu uveden jako spoluautor.

Ve

.....

Zlíně

Podpis diplomanta

OBSAH

ÚVOD	12
I TEORETICKÁ ČÁST	14
1 DĚLENÍ MOBILNÍCH ROBOTŮ	15
2 POHONNÉ JEDNOTKY HITEC HS 322	18
2.1 POPIS MOTORŮ HS 322	18
2.2 ŘÍDICÍ JEDNOTKA S MIKRO-KONTROLÉREM AT89C2051 PRO MOTORY HS 322 22	
2.3 PULZNĚ-ŠÍRKOVÁ MODULACE U MOTORŮ HS 322.....	24
2.4 KOMUNIKACE MEZI PC A ŘÍDICÍ JEDNOTKOU	25
3 PROGRAMOVÉ VYBAVENÍ PRO MOTORY HS 322 S NÁZVEM WIN SOS	26
3.1 POPIS PROSTŘEDÍ WIN SOS	27
3.2 OVLÁDÁNÍ A PROGRAMOVÁNÍ MOTORŮ HS 322 VE WIN SOS	28
4 POHONNÉ JEDNOTKY ROBOTIS RX 64	32
4.1 POPIS MOTORŮ RX 64.....	32
4.2 KOMUNIKACE MEZI ŘÍDICÍ JEDNOTKOU A MOTORY	34
4.3 DEFINOVÁNÍ INSTRUKČNÍHO PAKETU	34
4.4 DEFINOVÁNÍ STAVOVÉHO PAKETU.....	34
4.5 ADRESOVÁ TABULKA PAMĚTI V MOTORU RX 64.....	35
4.6 ŘÍDICÍ JEDNOTKA CM 2+	38
4.7 POPIS JEDNOTLIVÝCH SOUČÁSTÍ ŘÍDICÍ JEDNOTKY	39
4.8 PROPOJENÍ MOTORŮ RX 64 Z ŘÍDICÍ JEDNOTKOU CM 2+	41
4.9 POPIS KONEKTORU PRO RX 64	42
5 OVLÁDÁNÍ POMOCÍ SOFTWAREVÉHO VYBAVENÍ ROBOTTERMINAL	44
II PRAKTICKÁ ČÁST	48
6 NÁVRH A REALIZACE MOBILNÍHO ROBOTNICKÉHO SYSTÉMU S MOTORY HS 322	49
6.1 ÚPRAVA SERVOMECHANISMU	49
6.2 SESTAVENÍ A OŽIVENÍ MOBILNÍHO ROBOTNICKÉHO PODVOZKU S HS 322.....	53
6.3 OŽIVENÍ	54
7 DEFINOVÁNÍ SOFTWAREVÉHO VYBAVENÍ PRO ROBOTICKÝ PODVOZEK S HS 322	56

7.1	DESIGN, OTEVŘENÍ A NASTAVENÍ SÉRIOVÉHO PORTU	57
7.2	TVORBA PAKETU A OVLÁDÁNÍ MOBILNÍHO ROBOTNICKÉHO PODVOZKU S HS 322 58	
8	NÁVRH A REALIZACE MOBILNÍHO ROBOTNICKÉHO SYSTÉMU S MOTORY RX 64.....	64
8.1	SESTAVENÍ A OŽIVENÍ MOBILNÍHO ROBOTNICKÉHO PODVOZKU S RX 64	64
8.2	OŽIVENÍ	65
9	DEFINOVÁNÍ SOFTWAREVÉHO VYBAVENÍ PRO ROBOTICKÝ PODVOZEK S RX 64.....	66
9.1	TVORBA PAKETU A OVLÁDÁNÍ MOBILNÍHO ROBOTNICKÉHO PODVOZKU S RX 64 67	
10	ZÁVĚR.....	71
	ZÁVĚR V ANGLIČTINĚ	72
	SEZNAM POUŽITÉ LITERATURY	73
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	75
	SEZNAM OBRÁZKŮ	76
	SEZNAM TABULEK.....	78

ÚVOD

Současná expanze informačních technologií podmínila rozvoj asistenčních zařízení, která usnadňují, popřípadě nahrazují monotónní a vyčerpávající práci lidí. Tato zařízení jsou řazena do kategorie robotických technologií. Koncem minulého století se prosazovala nejvíce v sériových výrobních a v automatizovaných provozech. Podstatný pokles cen paměťových medií a mikroprocesorů umožňuje průnik robotických technologií i do veřejné správy, bezpečnostních složek, do ostrahy veřejných i výrobních objektů. Roboty jsou také neodmyslitelnou součástí v prostorách s výrobními technologiemi ohrožujícími lidský život či zdraví. Na trhu je možno pak vybírat s nepřehledného množství těchto systémů, avšak průnik robotických systémů do průmyslu komerční bezpečnosti zatím zaostává, přičemž jsou očekávány výsledky výzkumu a vývoje v rámci EU. Ceny robotů, které se v současnosti používají ve vojenství, policie a u bezpečnostních složek jsou vysoké, což omezuje jejich větší rozšíření.

Předmětem diplomové práce je výzkum a realizace systému mobilního robota, nejen pro studijní účely a výuku v univerzitních laboratořích pro obor Bezpečnostní technologie systémy a management (BTSM), ale i pro vývoj a zavedení tohoto robota do Průmyslu komerční bezpečnosti (PKB).

Zavedení mobilních robotických systému v PKB, umožňuje snížit náklady na výkon ostrahy a zvýšit efektivitu poskytovaných služeb, zejména pomocí monitorování určitého objektu pomocí mobilního robota vybaveného kamerovým systémem, vlastním napájecím zdrojem a komunikačním blokem. Systém mobilních robotů je řízen centrálním počítačem s inteligentními řídicími algoritmy pro řízení rojových robotických systémů (Swarm Robotic System), které umožňují efektivní využití mobilních robotů pro optimální pokrytí střeženého prostoru.

Rozhodující kritérium pro volbu pohonných jednotek platformy mobilního robota bylo minimalizace nákladů při poměru cena/výkon. První typ byl zvolen s ohledem na minimalizaci nákladů s využitím pohonných jednotek HS 322 od firmy HITEC a byl začleněn do sestavy tříkolového podvozku. Druhý typ mobilního podvozku byl volen z důrazem na vyšší kroutící moment, který umožňuje větší zátěž mobilního robota a jeho pohyb v členitém prostředí. Těmto požadavkům vyhovují pohonné jednotky typu RX 64 od

firmy ROBOTIS, které byly začleněny do sestavy čtyřkolového podvozku mobilního robotu.

I. TEORETICKÁ ČÁST

1 DĚLENÍ MOBILNÍCH ROBOTŮ

Mobilní roboty je možné dělit podle řady kritérií. Mezi základní patří rozdělení na dva typy:

- Autonomní - pracující v uzavřené zpětnovazební smyčce se senzorickým systémem, který zprostředkuje styk s vnějším prostředím.
- Dálkově řízené – pracující v otevřené zpětnovazební smyčce s lidským operátorem, za použití snímačů na rozhraní člověk stroj.

U autonomních robotů se předpokládá schopnost samostatně vykonávat zadanou úlohu. Může se např. jednat o sledování barevné čáry na podlaze a schopnost reagovat na eventuální překážku (zastavit se, vyhnout se) a posléze pokračovat v daném pohybu. Nebo umět se pohybovat v neznámém prostředí, dokázat ho zmapovat, orientovat se pak v něm a dosáhnout např. požadovaného cíle.[1]

Dálkově řízené roboty jsou řízeny operátorem, který má zpravidla vizuální informaci o pracovním okolí robotu. Ale i v tuto chvíli uvažujeme aspoň o částečném autonomním chování. Jako možnost příkladu při ztrátě komunikace s operátorem by tento robot musel uvolnit prostor a přemístit se k nejbližší zdi.

Mezi dálkové řízení lze zařadit též telerobotiku. Jedná se o dálkové řízení za pomoci prvků virtuální reality, kde pomocí stereovize, šesti-složkového ovladače a haptic interface se operátor cítí jako by se nalézal v pracovním prostoru robotu.[1]

Podle prostředí, ve kterém se robot má pohybovat, dělíme mobilní roboty pohybující se:

- Na souši (terrestrial) vnitřní/vnější prostředí (indole/outdoor) ve vodě (aquatic), ve vzduchu (airborne), ve vesmírném prostoru (space), hybridní.[1]

Podle účelu nasazení je lze dělit:

- Manipulační, montážní, servisní, inspekční, průzkumné, vojenství, zdravotnictví, určené pro zábavu a v brzké době doufejme určení pro PKB.[1]

Mobilní roboty pohybující se po souši je možné dále dělit podle typu pohybového subsystému na:

- Kolové, pásové, kráčejší, plazivé, šplhající, skákající a hybridní.
Z nichž jsou nejrozšířenější kolové, pásové a kráčejší.[1]

Mobilní roboty s kolovým podvozkem tvoří nejrozšířenější skupinu, zmíním zde typické platformy.

Kola mohou být aktivní – hnaná nebo pasivní – vlečná. Z konstrukčního hlediska pak je možno použít různé typy kol – standardní, všesměrová, Einsteinovy, článkové, MaxWheel a netradiční.[1]

Dalším důležitým parametrem kol mobilních podvozků je počet stupňů volnosti jejich pohybu. Běžně používaná jich mají jeden nebo dva. Kola s jedním stupněm volnosti se mohou pouze odvalovat po povrchu podél jedné osy (hřídele). V případě kola s více stupni volnosti se kolo pohybuje kolem dvou os, které mohou být rovnoběžné s povrchem, nebo jedna osa je rovnoběžná a druhá kolmá – např. přední kolo auta. [1]

Nejjednodušším typem je tříkolové uspořádání podvozku s diferenčně řízenými koly. Toto uspořádání používá dvě nezávisle poháněná kola a jedno volně otočné nepoháněné směrové kolo vpředu či vzadu mobilního podvozku. Výhodou této koncepce je manévrovatelnost – robot je schopen se otáčet namísto kolem osy, dále snadné řízení – zatáčení změnou rychlosti otáčení kol. Nevýhodou pak je nízká průchodnost terénem.[1]

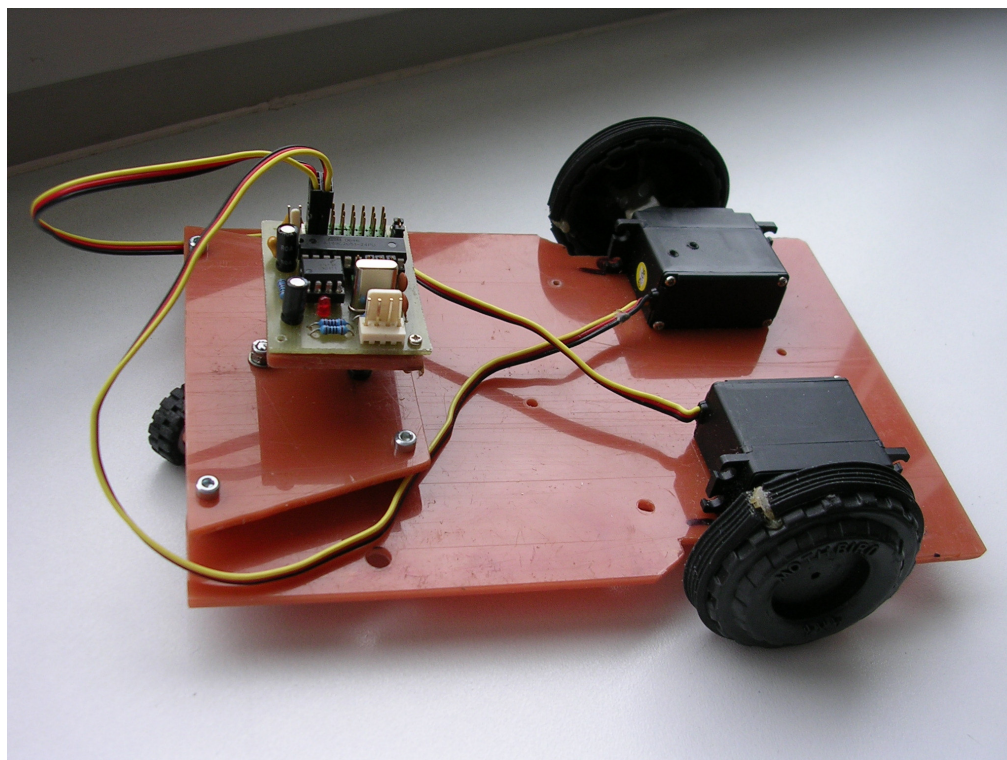
Zvláštní skupinu pohybových subsystémů tvoří nejčastěji tříkolové a čtyřkolové podvozky se všesměrovými koly. Pohybová ústrojí založená na těchto kolech dovolují robotu se volně pohybovat ve všech směrech a zatáčet či rotovat na velmi malém prostoru. Umožňují okamžitou změnu směru s nulovým poloměrem zatáčení. Jejich nevýhodou je vyšší cena, daná požadavkem na precizní výrobu, dále pak malá schopnost zdolávání překážek. Z tohoto důvodu najdou uplatnění prakticky jen v aplikacích pro vnitřní prostředí.[1]

Pásové podvozky, je možné si představit, jako rozšíření kol. Díky tomu se zlepšuje průchodnost terénem a stoupavost. Na druhou stranu je pohyb a zejména otáčení doprovázené smykem, více energeticky náročné. Z tohoto důvodu se používá tento druh podvozku převážně pro soustavy s teleoperátorem a pro speciální aplikace.

Kráčející roboty, zaujímají významné postavení v oblasti mobilních robotů. Uplatnění nacházejí zejména tam, kde se projeví výhody tohoto typu podvozku – průchodnost členitým terénem. Svým vzhledem a způsobem pohybu mohou napodobovat člověka, zvířata nebo hmyz. V současné době existuje celá řada typů kráčejících robotů, které se nejčastěji rozdělují podle počtu noh a jejich stupňů volnosti. Podle tohoto rozdělení je nejrozšířenější šestinohá skupina. Mezi nevýhody těchto robotů patří vyšší počet řízených os, a tedy i akčních členů (pohodny, senzory, převody apod.), složitější řízení, u dvounohých robotů je dále potřeba řídit stabilitu, menší energetická účinnost, konstrukční složitost a výrobní náročnost. [1]

2 POHONNÉ JEDNOTKY HITEC HS 322

V úvodu byl předestřen zájem o vytvoření co nejjednodušší a nejlevnější verze mobilního robota, tento robot byl složen z modelářských pohonných jednotek typu HITEC HS 322, jelikož spadají do cenově velmi přístupné relace a jejich charakteristika splňuje požadavky na vytvoření mobilního robotnického systému.

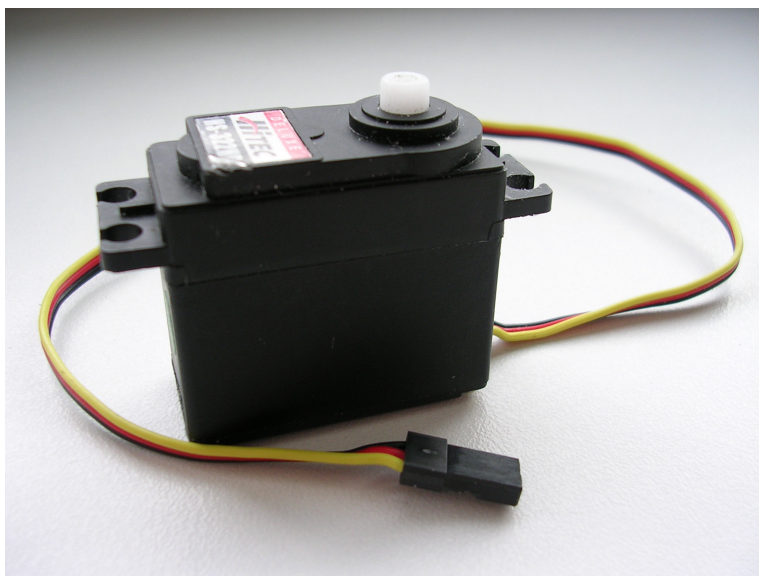


Obrázek 1 Mobilní robotický podvozek s HS 322

2.1 Popis motorů HS 322

Servomotor HS 322 je zařízení, které jako většina jiných servomechanismů obsahuje šest základních částí:

- hnací převody
- motor
- potenciometr
- řídicí obvod
- pouzdro servomechanismu
- výstupní hřídel



Obrázek 2 Motor HS 322

V těchto modelářských servomechanismech se používá motor na stejnosměrné napětí z kterého jsou vyvedeny 3 vodiče (černý, červený, žlutý).[2]

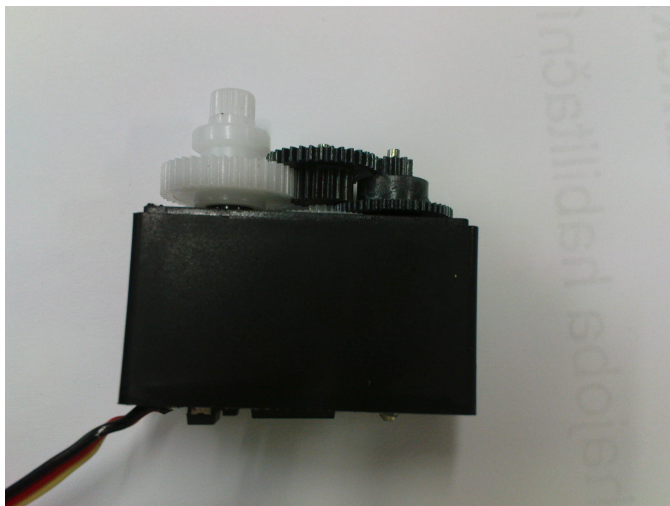
Popis vodičů:

- černý vodič – představuje zem (GND)
- červený vodič – představuje napájení (UCC)
- žlutý vodič – představuje vodič pro přenos signálu PWM

Hřídel motoru HS 322 je možné nastavit do požadované pozice zasíláním signálu pulzně-šířkové modulace (PWM). Po dobu, kdy je signál z řídicí jednotky zasílán, hřídel motoru udržuje aktuální pozici. Při změně PWM signálu dojde také ke změně pozice hřídele. Výhodou těchto modelářských servomechanismů je, že při své velikosti jsou dostatečně výkonné a přivádí výkon rovnoměrně do mechanických částí. Z tohoto důvodu je zřejmé, že málo zátěžový motor nespotřebuje tolik energie. [2]

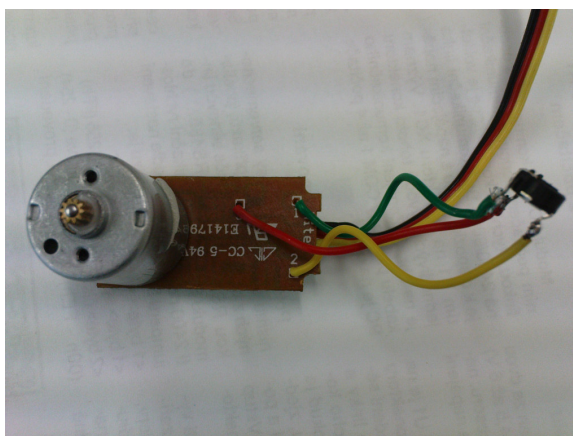
Motory HS 322 můžeme klasifikovat ze dvou hledisek a to rychlosti a kroutícího momentu. HS 322 můžeme přizpůsobit jen na jedno z hledisek týkající se vyšší rychlosti nebo kroutící moment. [2]

Nejdůležitější částí servomechanismu je motor, potenciometr a řídicí obvod. Potenciometr umožňuje řídicímu obvodu sledovat aktuální úhel natočení motoru. Motor otáčí výstupní hřídelí a zároveň se otáčí potenciometr přes sérii ozubených kol [obr. 3].



Obrázek 3 Ozubený převod motoru HS 322

Potenciometr je napojený na řídicí obvod modelářského servomechanismu, ten zabezpečuje zpětnou vazbu pro zastavení motoru v žádané pozici [obr. 4].



Obrázek 4 Zapojení potenciometru do obvodu motoru HS 322

Zpravidla je motor přizpůsobený na otáčení v rozsahu od 0° do 90° , případně od 0° do 180° . Kontinuální otáčení motoru o 360° je možné až v případě demontování mechanických zářezek v převodovém mechanismu.

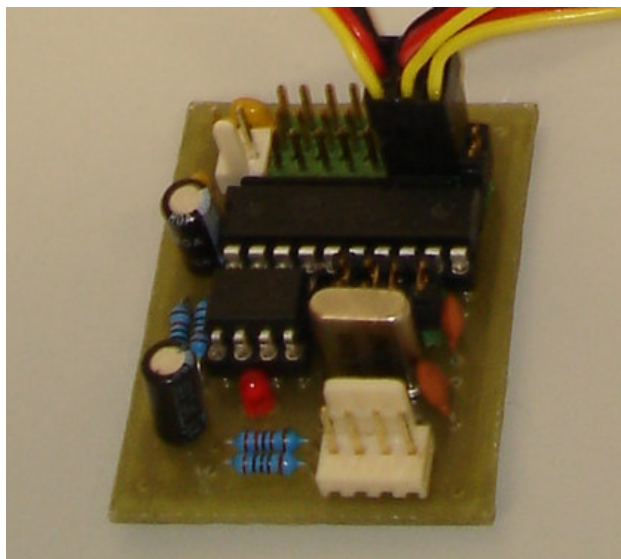
Výkon potřebný na otáčení motoru servomechanismu je rovnoměrný se vzdáleností, kterou musí motor udělat tzn. pokud se hřídel musí otočit o velkou vzdálenost, motor na to využije plnou rychlost, když je zapotřebí pootočení o malou vzdálenost, motor využívá menší rychlosti. [2]

Podrobná specifikace HS 322:

- Řídicí systém: Pulzně-šířkový modulovaný signál – 1500 μ s
- Požadovaný impuls: 3-5 V, obdélníkový signál
- Rozsah vstupního napájení: 4,8 – 6,0 V
- Rozsah teplot: -20 až +60 °C
- Rychlost otočení při nejnižším napájení (4,8 V): 0,19 s/60° (bez zátěže)
- Rychlost otočení při maximálním napájení (6,0 V): 0,15 s/60° (bez zátěže)
- Kroutící moment (4,8 V): 42 oz/in (3,0 kg/cm)
- Kroutící moment (6,0 V): 49 oz/in (4,5 kg/cm)
- Spotřeba elektrického proudu (4,8 V): 7,4 mA (v nečinnosti), 160mA (bez zátěže)
- Spotřeba elektrického proudu (6,0 V): 7,7 mA (v nečinnosti), 180mA (bez zátěže)
- Modifikovatelný na 360°: Ano
- Váha: 1,52 oz (43 g)

2.2 Řídicí jednotka s mikro-kontrolérem AT89C2051 pro motory HS

322



Obrázek 5 Řídicí jednotka s mikro-kontrolérem AT89C2051

Schéma řídicí jednotky je na obrázku níže[obr. 6]. Celkové zapojení je generátorem řídicího signálu PWM pro 8 servomechanismů, ovládaných povely z osobního počítače. V tomto zapojení je využit mikro-kontrolér AT89C2051 s taktovací frekvencí 24MHz. Povely jsou posílány do modulu přes sériovou linku (COM). Pokud řídicí jednotka nezaznamenává žádné povely z nadřazeného systému, tak na všech osmi výstupech se neustále generuje řídicí signál PWM podle naposledy přijatého povelu. Díky tomu, že je na výstupu stále generován signál PWM, nemůže dojít k samovolnému pohybu hřídele servomechanismu (ani při náhodném působení vnější síly).[3]

Parametry ovládacího signálu PWM pro motory a různé možnosti komunikace s počítačem je možné nastavit zkratovacími spojkami (jumpery) R, A0, A1, A2 a B.

Jumper R (rozsah)

Pokud je jumper nasazen (zkratován), je délka kladného impulsu signálu PWM 1 až 2 ms, výstupní hřídel servomechanismu se tak může otáčet v rozsahu maximálně 90°. Toto nastavení je bezpečné pro servomechanismy všech výrobců.

Je-li jumper sejmut, dojde ke zvětšení kladného impulsu PWM signálu na 0,5 až 2 ms a výstupní hřídel servomechanismu se pak může otáčet v rozsahu 180 úhlových stupňů. U tohoto nastavení je zapotřebí dbát velké opatrnosti, aby nedošlo k poškození celého systému servomechanismu.[3]

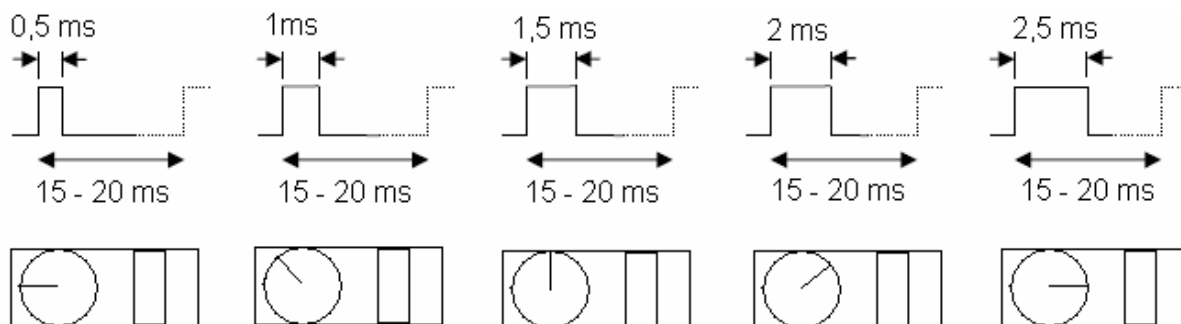
Jumper A (adresa) 0, 1, 2

Nastavením tohoto jumperu pak můžeme propojit více řídicích jednotek, pro ovládání více servomechanismů. [3]

Jumper B (baud)

Nastavování komunikační rychlosti mezi sériové linky se děje právě pomocí tohoto jumperu. Pokud je propojka nasazena, je komunikační rychlost nastavena na 9600Bd. Při sejmutí jumperu klesne komunikační rychlost na 2400Bd. Snížené rychlosti se využívá nejvíce v prostředích se silným rušivým polem a také ke zpětné kompatibilitě se staršími programy. [3]

Stav „spojek“ je čten jen po nulování procesoru, což znamená, že musí dojít k restartu řídicí jednotky nebo k jejímu zapnutí a vypnutí.



Obrázek 7 Nastavení polohy hřídele servomotoru HS 322

2.4 Komunikace mezi PC a řídicí jednotkou

Poloha výstupní osy servomechanismu se řídí sledem příkazů posílaných z osobního počítače sériovou linkou RS 232 s rychlostmi 9600Bd nebo 2400Bd.

Formát příkazu musí být specifický a to:

FF – ID – POZICE

FF je definováno jako hlavička paketu, oznamuje začátek nového paketu řídicí elektronice motoru

ID číslo motoru, pro které je paket určen

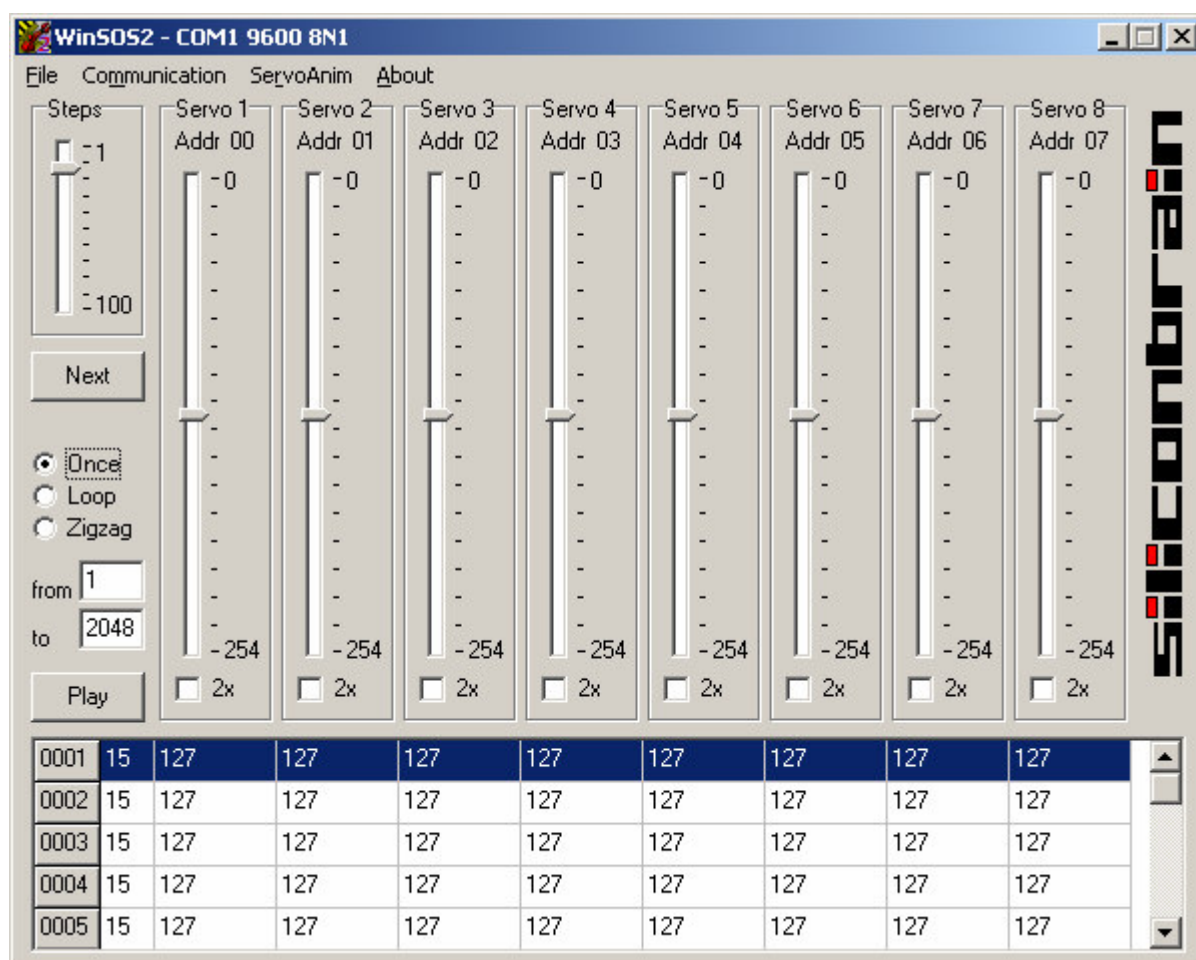
POZICE určuje pozici výstupní hřídele a to v rozsahu (0 – 254)

Povely musí být ovládacím programem posílány vždy v číselném formátu, nikdy ve formátu textovém. Paket odesíláme do řídicí jednotky jen tehdy, požadujeme-li změnu polohy výstupního hřídele některého z ovládaných servomechanismů. Periodické opakování signálu PWM pro udržování hřídele v poloze, kterou jsme si zvolili, pak zajišťuje řídicí jednotka sama. Rychlosti otáčení hřídele servomechanismu mezi koncovými body jsou pak dána pouze konstrukcí, převodovky není možné ji ovlivnit programově.

3 PROGRAMOVÉ VYBAVENÍ PRO MOTORY HS 322 S NÁZVEM WIN SOS

Software WIN SOS [obr.8] je volně šiřitelný program pro ovládání modelářských servomechanismu typu HS. Je vytvořen pro řadu mikro-kontrolérů AT89CXXXX, s podporou osmi slotů pro připojení servomechanismů.

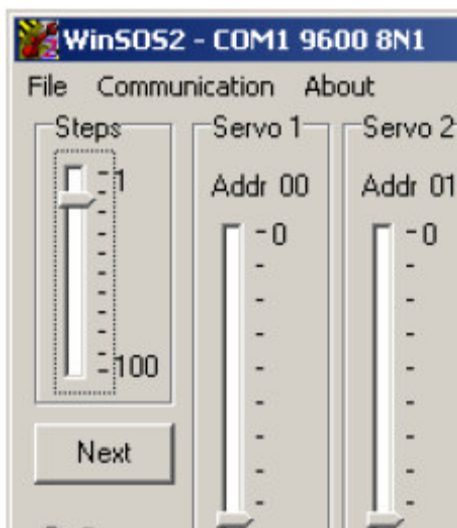
Prostředí WIN SOS je definováno i pro programování sekvencí pomocí playbacku, kdy pohyb jednotlivých motorů je zaznamenáván do souboru a po ukončení je možné tyto sekvence přehrávat v cyklu. WIN SOS slouží jako demonstrativní prostředek pro základy programování průmyslových robotů, kdy jednotlivé pohyby jsou opakovány ku potřebě sériové linky. [4]



Obrázek 8 Prostředí WIN SOS

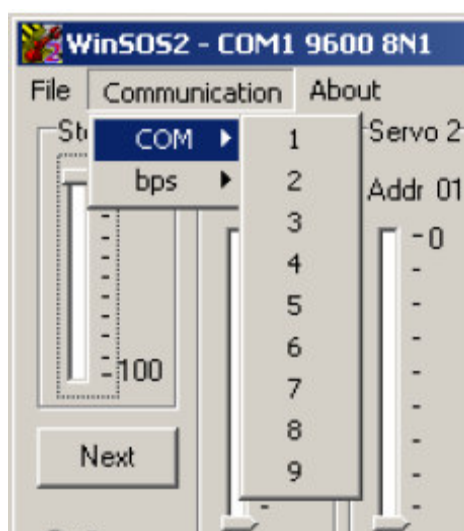
3.1 Popis prostředí WIN SOS

Po spuštění programu nejprve z menu zvolíme parametry komunikačního portu volbou nabídky *Communication* v nástrojové liště. [obr. 9]



Obrázek 9 Program WIN SOS Communication

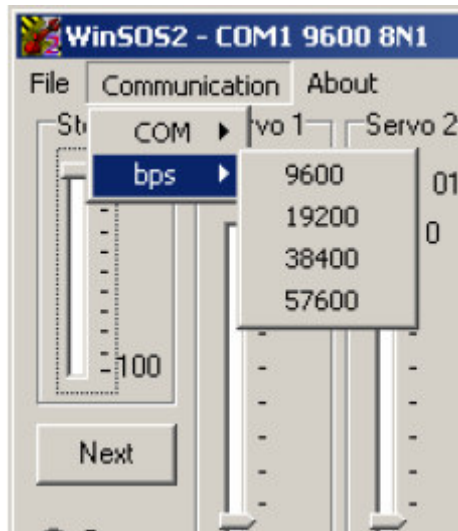
V nabídce nejprve zvolíme položku COM, v níž nastavíme číslo sériového portu, například 1 pro COM1 viz [obr. 10].



Obrázek 10 Program WIN SOS zvolení portu COM

V položce bps viz [obr. 11] pak nastavíme komunikační rychlost sériového portu, která vyhovuje připojenému zařízení. Pro řídicí jednotku s procesorem AT89C2051 zvolíme

komunikační rychlost 9600. Na tuto rychlost bude nastaven vybraný sériový port na PC, kterým budeme ovládat řídicí jednotku. Zvolená komunikační rychlost by měla odpovídat hodnotě nastavené na řídicí jednotce pomocí Jumperu B (musí být zkratován). [4]

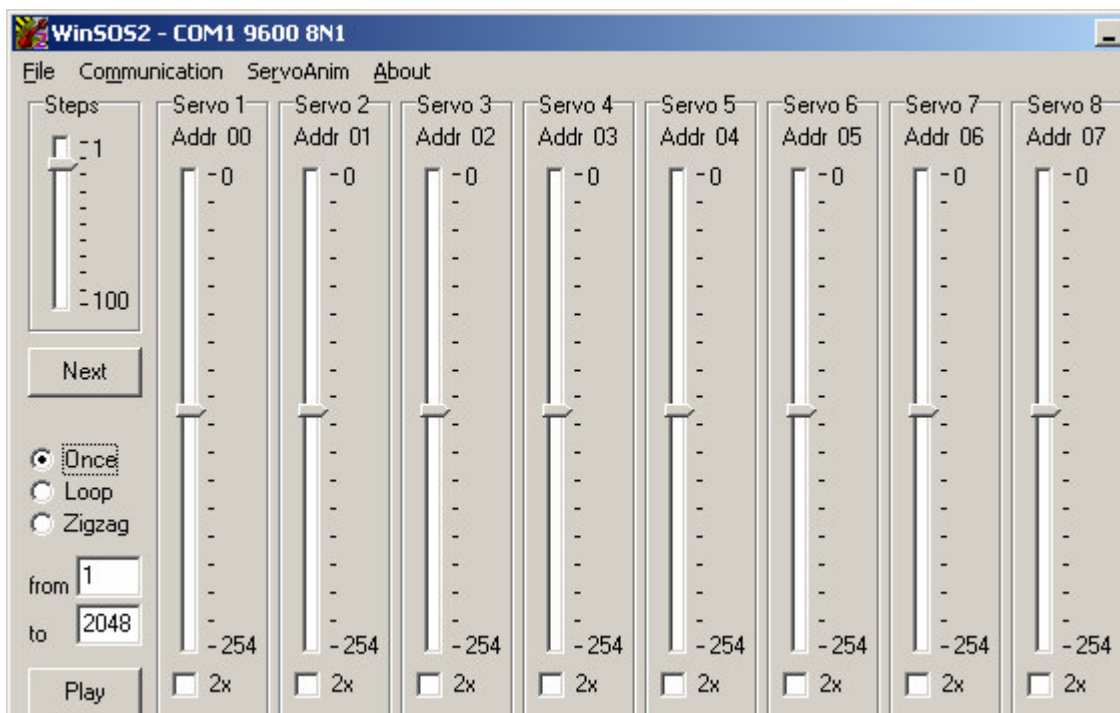


Obrázek 11 Program WIN SOS nastavení komunikační rychlosti portu COM

Zbylé nastavení v této položce je definováno pro řídicí jednotky jiného typu než s mikro-kontrolérem AT89C2051. Námí zvolené nastavení zůstává v programu uloženo. Dokončením nastavení těchto parametrů a propojením pomocí RS 232 je software připraven k použití.[4]

3.2 Ovládání a programování motorů HS 322 ve WIN SOS

V okně programu jsou na panelu ovládací okénka servomechanismů s výchozím názvem „Servo 1“ až „Servo 8“ viz [obr. 12].



Obrázek 12 Program WIN SOS hlavní panel

V těchto oknech jsou umístěny posuvníky, kterými můžeme přímo ovládat pohyb motorů HS 322 a aktuální pozice posuvníku je okamžitě zobrazena v tabulce ve spodní části okna programu[obr.13].

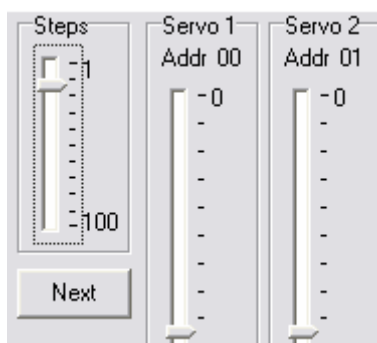
Pokud jsme s nastavenou pozicí servomechanismů spokojeni, stiskneme tlačítko *Next* a aktuální pozice servomechanismů se zapíše do následující pozice tabulky. Proto nedojde při zápisu k pohybu servomechanismu ze zvolené pozice.

0001	15	127	127	127	127	127	127	127	127
0002	15	127	127	127	127	127	127	127	127
0003	15	127	127	127	127	127	127	127	127
0004	15	127	127	127	127	127	127	127	127
0005	15	127	127	127	127	127	127	127	127

Obrázek 13 Tabulka programu WIN SOS se zaznamenanými polohami

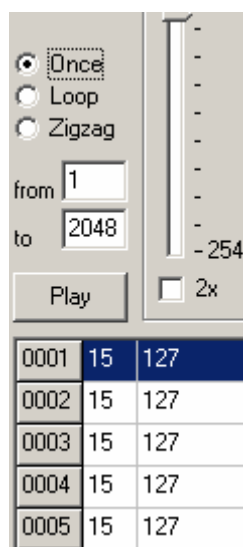
Posuvníky posléze nastavujeme další pozice modelářských servomechanismů a postup opakujeme až do vytvoření celé sekvence pohybu.

Ručně vytváříme pouze pozice klíčové, při přehrávání zaznamenané sekvence je mezi těmito pozicemi prováděna interpolace. Celkový počet takto interpolovaných kroků je při ukládání každé klíčové pozice možno nastavit posuvníkem v oknu „Steps“ [obr. 14]. Tímto posuvníkem zvolíme počet kroků, které budou automaticky vloženy mezi dvě klíčové pozice servomechanismu. Rozestup mezi jednotlivými vkládanými kroky je pevně nastaven na 40ms (1/25s). Pokud chceme vytvořit všechny polohy serv ručně a nevkładat žádnou interpolaci, postačí nechat posuvník *Steps* stále nastaven na 1. Při přehrávání sekvence se pak neprovádějí žádné mezikroky.[4]



Obrázek 14 Zobrazení posuvníku *Steps* a tlačítka *Next* pro ukládání poloh

Pokud máme klíčové pozice nastaveny, můžeme sekvenci přehrát. Nejprve zvolíme od kterého do kterého kroku (řádky tabulky) se má sekvence přehrávat. To nastavíme v okénkách „*from*“ a „*to*“ viz [obr. 15].



Obrázek 15 Nastavení přehrávání

Zvolíme také způsob přehrávání sekvence:

„**Once**“ – pouze jednou

„**Loop**“ – přehrávat stále dokola

„**Zigzag**“ – tam a zpět

Stiskem tlačítka **Play** začne přehrávání sekvence od pozice, zadané v položce „**from**“. Rychlost přesunu ze současné pozice do prvního bodu přehrávané sekvence je omezena, nedojde tedy k prudkému pohybu servomechanismu. Motory nejprve plynule najedou na výchozí pozici sekvence. Tlačítko **Play** se při běhu přehrávání změní na **Stop**. Přehrávání sekvence lze ukončit stiskem tlačítka **Stop**; přehrávaná sekvence se ukončí po dokončení aktuálně prováděné klíčové pozice a tlačítko **Stop** se znovu změní na **Play**. [4]

Je-li hodnota položky „**from**“ rovna položce „**to**“, přehrávání se nespustí, protože rozsah sekvence není korektně nastaven.

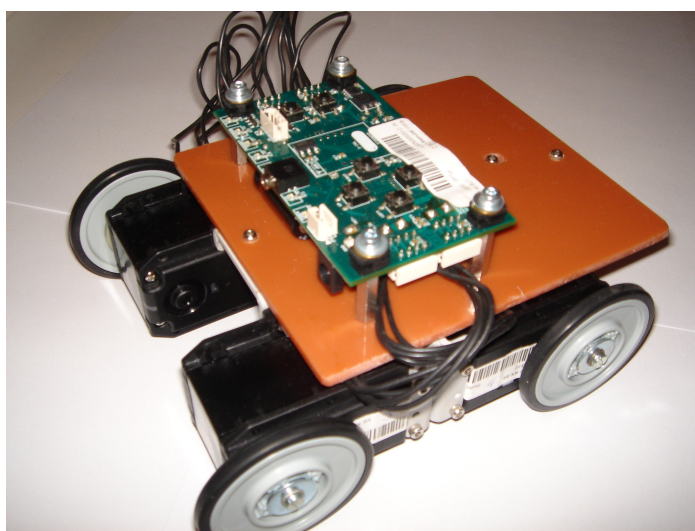
Vyhovuje-li námi nastavená sekvence, můžeme ji uložit volbou položky „**Save**“ z nabídky **File**. Otevřít uloženou sekvenci můžeme z nabídky **File** zvolením položky „**Open**“ viz [obr. 16]. Volbou položky „**Exit**“ program ukončíme.



Obrázek 16 Nabídka File programu WIN SOS

4 POHONNÉ JEDNOTKY ROBOTIS RX 64

Druhý typ mobilního podvozku je složen z motorů typu RX 64, je postaven za účelem dálkového ovládní za pomoci softwarového řešení pomocí vývojového prostředí SharpDevelop, využívající jazyk C#, které je volně šiřitelné a sloužilo k nadefinování příkazů pro dálkové ovládní. Jako řídicí jednotka byla dodána jednotka CM 2+ , která poskytuje možnost propojení motorů RX 64 se sériovým rozhraním počítače a jednodušší typ uživatelského ovládní.



Obrázek 17 Mobilní robotický podvozek s motory RX 64

4.1 Popis motorů RX 64

Citlivost 0,29°

Podpora kontinuálního otáčení či otáčení v rozmezích 0°- 300°

Napájení 12 – 21V

Maximální odebíraný proud 1,2 A

Rozsah pracovních teplot -5 – 85°C

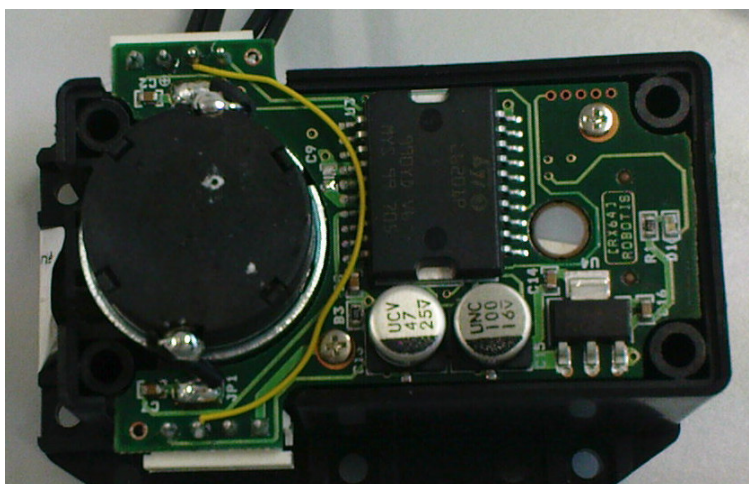
Rozhraní mezi motory – RS 485 asynchronní sériová komunikace (8bitů, 1 stop bit, žádná parita)[5]

Maximální počet připojených motorů je 254

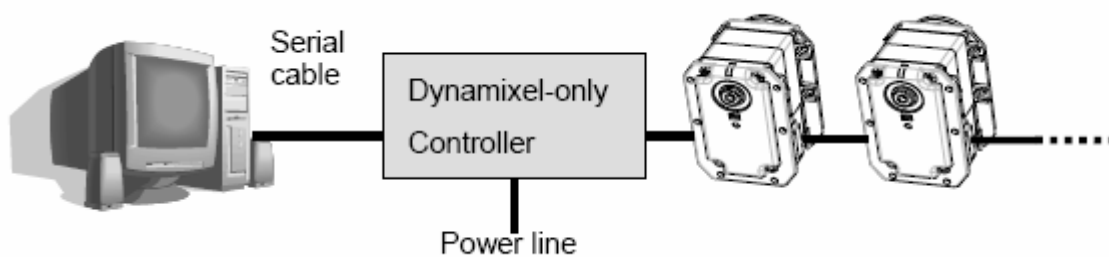
Komunikační rychlost 7343 bps – 1 Mbps

Složení motoru RX 64

- řídicí elektronika
- absolutní snímač polohy otáčení
- stejnosměrný motor
- převodové ústrojí



Obrázek 18 Řídicí elektronika motoru RX 64



Obrázek 19 Zapojení motorů RX 64 s řídicí jednotkou a PC[5]

4.2 Komunikace mezi řídicí jednotkou a motory

Komunikace probíhá pomocí paketů, tyto pakety se rozlišují na instrukční paket a Stavový paket. Instrukční paket je odeslán z řídicí jednotky pro nastavení parametrů motoru. Stavovým paketem odpovídají motory na přijetí paketu a potvrzují nastavení.[5]

4.3 Definování instrukčního paketu

FF-FF-ID-DÉLKA-INSTRUKCE-PARAMETR 1 ... PARAMETR N-KONTROLNÍ
SUMA

FF-FF – je definováno jako hlavička paketu, oznamuje začátek nového paketu řídicí elektronice motoru.

ID – číslo motoru, pro které je paket určen.

DÉLKA – je délka paketu, počítána z parametrů + 2.

INSTRUKCE – definování povelu pro motor.

PARAMETR – je definováno jako zpřesnění instrukce („jak použít paket“).

KONTROLNÍ SUMA – součet id, délky, instrukce a parametrů. Slouží jako kontrola správnosti odeslání paketu jestli nedošlo k chybě.[5]

4.4 Definování stavového paketu

FF-FF-ID-DÉLKA-CHYBA-PARAMETR 1 ... PARAMETR N-KONTROLNÍ SUMA

CHYBA – Pokud dojde k chybě při zpracování nebo odeslání můžeme ze stavovém paketu přesně určit typ chyby.

Ostatní části paketu jsou odpovídající instrukčnímu paketu.[5]

4.5 Adresová tabulka paměti v motoru RX 64

Kontrolní tabulka obsahuje informace o stavu a operacích RX-64. RX-64 je ovládaný zápisem hodnot do kontrolní tabulky a status je kontrolovaný čtením hodnot z kontrolní tabulky.

Data v tabulce ukazují nastavení továrních hodnot motorů RX 64 při připojení ke zdroji elektrické energie.

Tabulka 1 Složení paměti motoru a jednotlivých adres[5]

	Address (hexadecimal)	Name	Description	Access	Initial Value (Hexadecimal)
EEPROM Area	0 (0X00)	Model Number(L)	Lowest byte of model number	R	64 (0X40)
	1 (0X01)	Model Number(H)	Highest byte of model number	R	0 (0X00)
	2 (0X02)	Version of Firmware	Information on the version of firmware	R	-
	3 (0X03)	ID	ID of Dynamixel	RW	1 (0X01)
	4 (0X04)	Baud Rate	Baud Rate of Dynamixel	RW	34 (0X22)
	5 (0X05)	Return Delay Time	Return Delay Time	RW	250 (0XFA)
	6 (0X06)	CW Angle Limit(L)	Lowest byte of clockwise Angle Limit	RW	0 (0X00)
	7 (0X07)	CW Angle Limit(H)	Highest byte of clockwise Angle Limit	RW	0 (0X00)
	8 (0X08)	CCW Angle Limit(L)	Lowest byte of counterclockwise Angle Limit	RW	255 (0XFF)
	9 (0X09)	CCW Angle Limit(H)	Highest byte of counterclockwise Angle Limit	RW	3 (0X03)
	11 (0X0B)	the Highest Limit Temperature	Internal Limit Temperature	RW	80 (0X50)
	12 (0X0C)	the Lowest Limit Voltage	Lowest Limit Voltage	RW	60 (0X3C)
	13 (0X0D)	the Highest Limit Voltage	Highest Limit Voltage	RW	240 (0XF0)
	14 (0X0E)	Max Torque(L)	Lowest byte of Max. Torque	RW	255 (0XFF)
	15 (0X0F)	Max Torque(H)	Highest byte of Max. Torque	RW	3 (0X03)
	16 (0X10)	Status Return Level	Status Return Level	RW	2 (0X02)
	17 (0X11)	Alarm LED	LED for Alarm	RW	36 (0X24)
	18 (0X12)	Alarm Shutdown	Shutdown for Alarm	RW	36 (0X24)
RAM Area	24 (0X18)	Torque Enable	Torque On/Off	RW	0 (0X00)
	25 (0X19)	LED	LED On/Off	RW	0 (0X00)
	26 (0X1A)	CW Compliance Margin	CW Compliance margin	RW	0 (0X00)
	27 (0X1B)	CCW Compliance Margin	CCW Compliance margin	RW	0 (0X00)
	28 (0X1C)	CW Compliance Slope	CW Compliance slope	RW	32 (0X20)
	29 (0X1D)	CCW Compliance Slope	CCW Compliance slope	RW	32 (0X20)
	30 (0X1E)	Goal Position(L)	Lowest byte of Goal Position	RW	-
	31 (0X1F)	Goal Position(H)	Highest byte of Goal Position	RW	-
	32 (0X20)	Moving Speed(L)	Lowest byte of Moving Speed	RW	-
	33 (0X21)	Moving Speed(H)	Highest byte of Moving Speed	RW	-
	34 (0X22)	Torque Limit(L)	Lowest byte of Torque Limit	RW	ADD14
	35 (0X23)	Torque Limit(H)	Highest byte of Torque Limit	RW	ADD15
	36 (0X24)	Present Position(L)	Lowest byte of Current Position	R	-
	37 (0X25)	Present Position(H)	Highest byte of Current Position	R	-
	38 (0X26)	Present Speed(L)	Lowest byte of Current Speed	R	-
	39 (0X27)	Present Speed(H)	Highest byte of Current Speed	R	-
	40 (0X28)	Present Load(L)	Lowest byte of Current Load	R	-
	41 (0X29)	Present Load(H)	Highest byte of Current Load	R	-
	42 (0X2A)	Present Voltage	Current Voltage	R	-
	43 (0X2B)	Present Temperature	Current Temperature	R	-
	44 (0X2C)	Registered Instruction	Means if Instruction is registered	RW	0 (0X00)
	46 (0X2E)	Moving	Means if there is any movement	R	0 (0X00)
47 (0X2F)	Lock	Locking EEPROM	RW	0 (0X00)	
48 (0X30)	Punch(L)	Lowest byte of Punch	RW	32 (0X20)	
49 (0X31)	Punch(H)	Highest byte of Punch	RW	0 (0X00)	

Popis jednotlivých hodnot v kontrolní tabulce:

Address 0x00,0x01 – modelové číslo 0X0040

Address 0x02 – verze firmware

Address 0x03 – unikátní ID číslo motoru, které jej identifikuje

Address 0x04 – přenosová rychlost tzn. komunikační rychlost Speed (BPS) = 2000000 / (Address4 + 1)

Address 0x05 – vrácení zpoždovacího času, je to čas mezi odesláním a přijmutím paketu

Address 0x06,0x07,0x08,0x09 – úhlový operační limit, cíl musí být v limitu

Address 0x0B – nejvyšší teplotní pracovní limit RX-64

Address 0x0C,0x0D – nejvyšší a nejnižší napětí

Address 0x0E,0x0F, 0x22,0x23 – maximální kroutivá síla, pracuje v tzv. volném módu. Pokud připojíme RX-64 ke zdroji je z paměti EEPROM zkopírována hodnota nastavení kroutivého momentu do paměti RAM.

Address 0x10 – navracení paketu. RX motor navrátí stavový paket po příjmu instrukčního paketu

Address 0x14~0x17 – kalibrace- nelze měnit, je nastavena přímo pro daný potenciometr

Address 0x18 – možná kroutivá síla

Address 0x19 – nastavení LED

Address 0x1E,0x1F – požadovaná úhlová pozice pro motor – myšleno počáteční poloha

Address 0x20,0x21 – pohybová rychlost – nastavení úhlové rychlosti výstupního pohybu k cílové pozici. Nastavení této hodnoty je maximálně 0x3ff s výstupní úhlovou rychlostí 114RPM toho je docíleno jen s dostatečným napájením

Address 0x24,0x25 – „stávající pozice“ – aktuální úhlová pozice výstupu motoru

Address 0x26,0x27 - „stávající rychlost“ – aktuální úhlová rychlost výstupu motoru

Address 0x28,0x29 – „stávající zavádění dat do paměti“ – veličina je nahrána do motoru

Address 0x2A – stávající napájení – elektrické napájení aplikované na motoru. Hodnota napětí je 10x za vteřinu aktualizována. Příklad 10V je reprezentováno jako 100 (0x64)

Address 0x2B – stávající teplota – vnitřní teplota je udávána ve stupních Celsia

Address 0x2C – registrační instrukce – nastavení hodnoty 1, kdy instrukce je připsaná do REG_WRITE. Nastavení 0 po tomto dokončení připsá instrukci akčního příkazu.

Address 0x2E – pohyb – nastaví jedničku pokud se motor pohybuje vlastní silou

Address 0x2F – zamknutí – když nastavím 1, tak pouze adresy 0x18 až 0x23 mohou být zapsány. Toto uzamknutí může odemknout pouze vypnutí napájení.

Address 0x30,0x31 – tyto adresy upravují nejnižší možnou dodávku energie do motoru. Počáteční hodnota je nastavena 0x20 a maximální hodnota je nastavena 0x3ff.[5]

4.6 Řídicí jednotka CM 2+

CM 2+ je vyhodnocovací zařízení, používané k řízení motorů RX-64, DX 113, DX 117, RX 28 a senzoru AX S1(infra senzor). Na základě této řídicí jednotky můžeme ovládat systém servomotoru pomocí softwarového vybavení od výrobce či vlastního softwaru.[6]

Procesorové vybavení:

ATmega-128

Manufacturer : Atmel

Flash memory : 128KByte

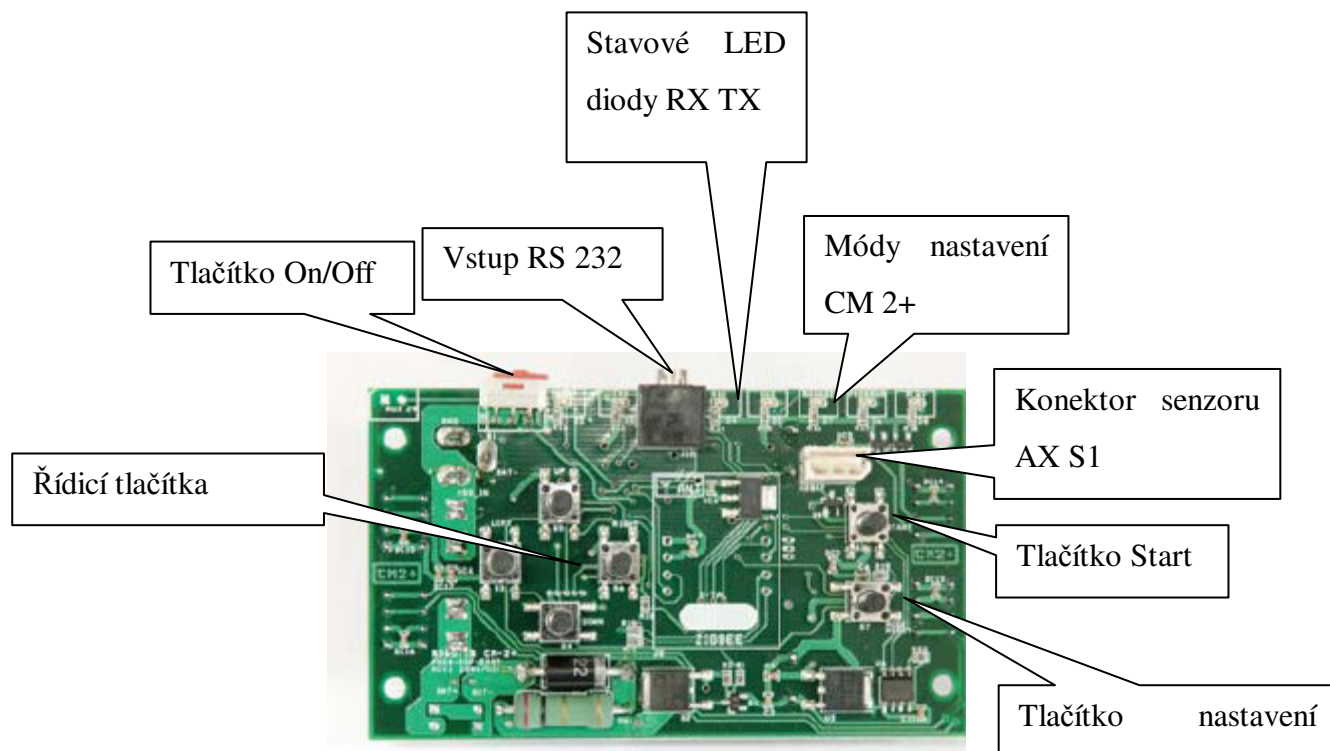
SRAM : 4KByte

EEPROM : 4KByte

UARTs : 2

CLOCK : 16MHz

4.7 Popis jednotlivých součástí řídicí jednotky

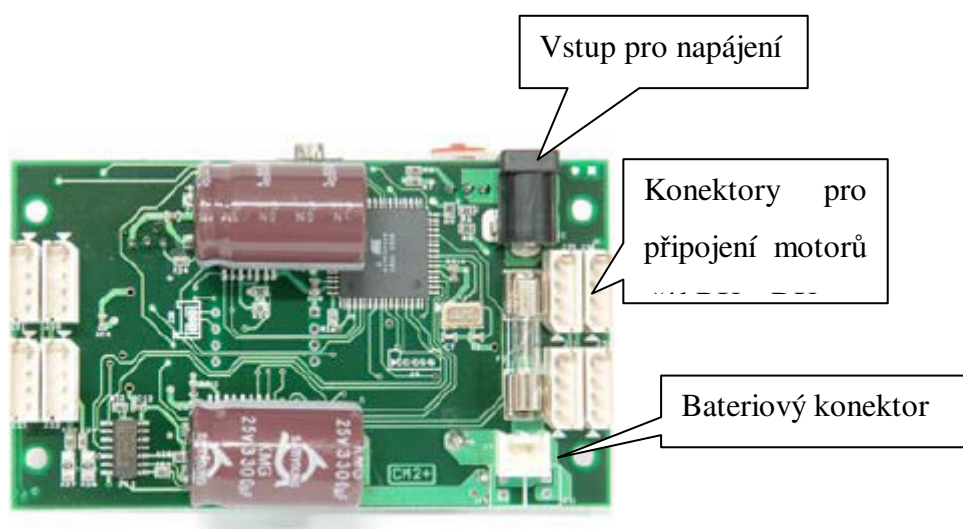


Obrázek 20 Řídicí jednotka CM 2+ (čelní strana)[6]

Tabulka 2 Popis částí řídicí jednotky CM 2+ (čelní strana)

Tlačítko On/Off	při stisknutí tohoto tlačítka dojde k zapnutí či vypnutí řídicí jednotky.
Vstup RS 232	slouží k propojení sériového rozhraní mezi PC a řídicí jednotkou
Stavové LED diody RX, TX	LED nám zobrazují komunikaci mezi PC a řídicí jednotkou a to přijímání(RX) či odesílání(TX) dat.
Módy nastavení CM 2+	lze volit mód, ve kterém chceme, aby řídicí jednotka pracovala, volíme vždy mezi 3 módy(Manage, Program, Play)
Konektor senzoru AX S1	u mobilního podvozku není použit senzor AX S1 – jednalo by se o infračervený senzor, který se využívá v Play módu pro

	vyhýbání se překážkám apod.
Tlačítko start	má potvrzovací funkci, když vybereme jeden z módů je zapotřebí jeho výběr potvrdit tlačítkem start tzn. že po celou dobu práce s řídicí jednotkou budeme pracovat v jednom programovém módu.
Tlačítko nastavení módu	je využíváno k přepínání mezi jednotlivými módy.
Řídicí tlačítka	jsou tlačítka, kterým můžeme přiřadit funkci dle potřeb programátora. Většinou se využívají pro definování směru pohybu v Play módu.



Obrázek 21 Řídicí jednotka CM 2+ (zadní strana)[6]

Tabulka 3 Popis částí řídicí jednotky CM 2+ (zadní strana)

Vstup napájení	zde přivádíme elektrickou energii pro napájení řídicí jednotky a motorů
Konektory pro připojení	řídicí jednotka je dimenzována na ovládání silnějších motorů tříd RX a DX tříd RX A DX, proto jsou na ní zastoupeny pouze

	konektory pro tyto typy
Bateriový konektor	jelikož se řídicí jednotka CM 2+ využívá i pro autonomní systémy, je v ní zabudován již speciální konektor pro připojení baterie.

4.8 Propojení motorů RX 64 z řídicí jednotkou CM 2+

K řídicí jednotce lze připojit jen určitý typ zařízení a to i od specifického výrobce jako je ROBOTIS. Řídicí jednotka dokáže ovládat pouze tyto typy zařízení: RX-64, DX 113, DX 117, RX 28 a senzor AX S1. Pro tato zařízení jsou připraveny jak konektory, tak firmware umístěný uvnitř řídicího procesoru. Jelikož si řídicí jednotka provádí samostatnou detekci připojených zařízení, není možné připojení jiných součástí než předepsaných. V případě, že dojde k přepsání firmwaru je možné obejít systém. To se však nedoporučuje, jelikož je možné zničit, jak zařízení řídicí jednotky, tak připojených servomechanismů.

S každým novým připojeným zařízením k řídicí jednotce je zapotřebí zjistit nastavení jeho ID (identifikačního čísla). U všech zařízení, které lze připojit k řídicí jednotce CM 2+ je nutné toto nastavení provést po jednotlivých kusech. Pokud bychom provedli zapojení všech servomechanismů do řídicí jednotky bez nastavení ID jednotka nám detekuje pouze jedno zařízení, které je připojeno. Důvodem je však to, že továrně jsou nastaveny všechny servomotory na ID 1. Je na programátorovi, aby jednotlivé identifikační čísla změnil a nastavil dle svých potřeb. Za pomoci softwaru RobotTerminál probíhá nastavení ID motorů funkcí ID [číslo motoru]. Číslo motoru představuje hodnotu, která slouží pro identifikaci mezi jednotlivými motory. Při komunikaci motorů s řídicí jednotkou CM 2+, pak nemůže dojít ke kolizi instrukčních paketů.

Systém CM 2+ je vyzdvihován v rámci vlastností, které splňuje a to hlavně ve smyslu připojených zařízení a jejich obsluhy. CM 2+ dokáže zpravovat na 254 servomotorů. Je to však podmíněno dostatečnou elektrickou stabilitou.

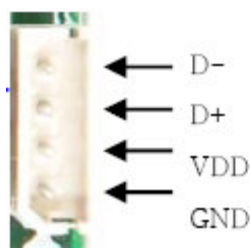
Struktura zapojení servomotorů do systému s CM 2+ využívá rozhraní RS 485. Na této struktuře probíhá komunikace mezi motory a řídicí jednotkou ve formě polo-duplexu.

Požadované napájení je v rozmezí 12 – 16V. Energie dodávaná do CM 2+ je dodávaná i do motorů z důvodu jednoho zdroje elektrické energie. Požadovaný proud jen pro CM 2+ je 50mA. Ale příjem této energie musí být upraven i pro ostatní motory.



Obrázek 22 Schématické naznačení zapojení a identifikace motorů RX 64[6]

4.9 Popis konektoru pro RX 64



Obrázek 23 Konektor RX 64[6]

Konektor obsahuje 4 piny, ve kterých je obsaženo napájení VDD, GND a komunikace D- , D+. Pro snadnější orientaci je v konektoru také veden klíč pro správné rozpoznání jednotlivých pinů. [6]

Popis zkratek:

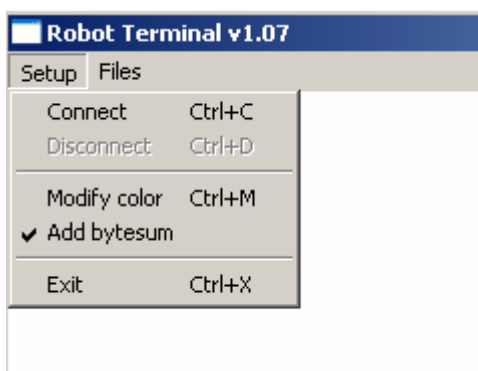
- D- invertovaný vodič
- D+ neinvertovaný vodič
- VDD napájecí napětí
- GND zem

Jednotlivé vodiče jsou označeny D- /D +, kde D (-) označuje tzv. invertovaný vodič a D (+) jako neinvertovaný vodič. Logický stav 1 (někdy označený jako OFF), reprezentuje napěťový rozdíl $D(-) - D(+)$ < - 0.3 V, zatímco logický stav 0 (ON) reprezentuje rozdíl $D(-) - D(+)$ > + 0.3 V. Přenos pomocí rozdílového napětí eliminuje vliv naindukovaného rušivého napětí vztaženého k nulovému potenciálu (zemi), protože se na obou vodičích naindukují stejná velikost napětí. Správný vysílač by měl generovat na výstupu úroveň + 2 V a - 2 V a přijímač by měl být ještě schopen rozlišit úroveň + 200 mV a - 200 mV jako platný signál.

5 OVLÁDÁNÍ POMOCÍ SOFTWAREVÉHO VYBAVENÍ ROBOTTERMINAL

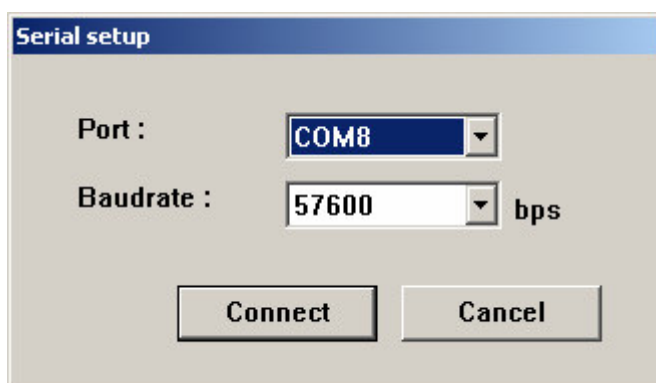
Nastavování a ovládání motorů je prováděno za pomoci softwarového vybavení dodávaného výrobcem a to pod názvem RobotTerminal. Používání RobotTerminálu lze pouze v módu Manage nastaveném na řídicí jednotce. Po tomto správném nastavení můžeme spustit tento software a provést další nastavení.

Spuštěním RobotTerminálu musíme nastavit připojení. Kliknutím na Setup a posléze na Connect můžeme nastavit připojení jednotky CM 2+.[obr.24] Zložka files slouží k přenosu souborů do paměti řídicí jednotky.[6]



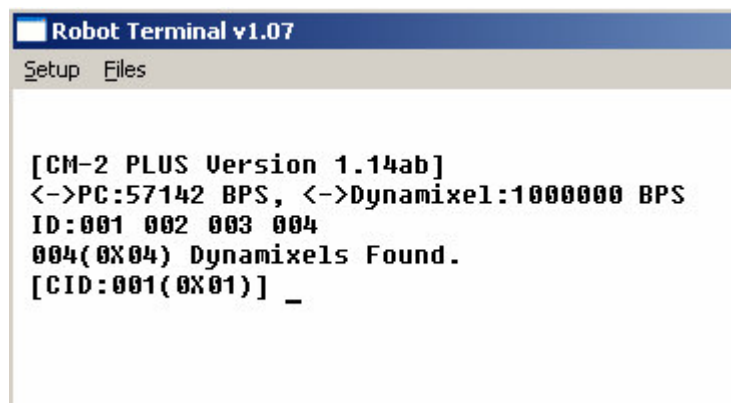
Obrázek 24 Nastavení připojení CM 2+

Kliknutím na Connect se zobrazí tabulka s výběrem COM portu, na kterém máme zařízení připojeno s nastavením přenosové rychlosti. [obr. 25]



Obrázek 25 Tabulka nastavení portu COM a přenosové rychlosti

Vyplněním údajů můžeme přejít na tlačítko Connect a propojit jím PC s řídicí jednotkou CM 2+. Posléze by se nám mělo objevit definování verze firmwaru instalovaného v řídicí jednotce. Dále rychlost připojení mezi řídicí jednotkou a PC a na stejném řádku i nastavení rychlosti mezi motory RX 64. Hodnoty rychlosti jsou vyjádřeny v Baudech za vteřinu. Na následujícím řádku jsou vypsány identifikace jednotlivých motorů a pod nimi počet zařízení, které řídicí jednotka našla celkem. V dalším řádku je možné již přímo ovládat zvolený motor 001. [obr. 26][6]

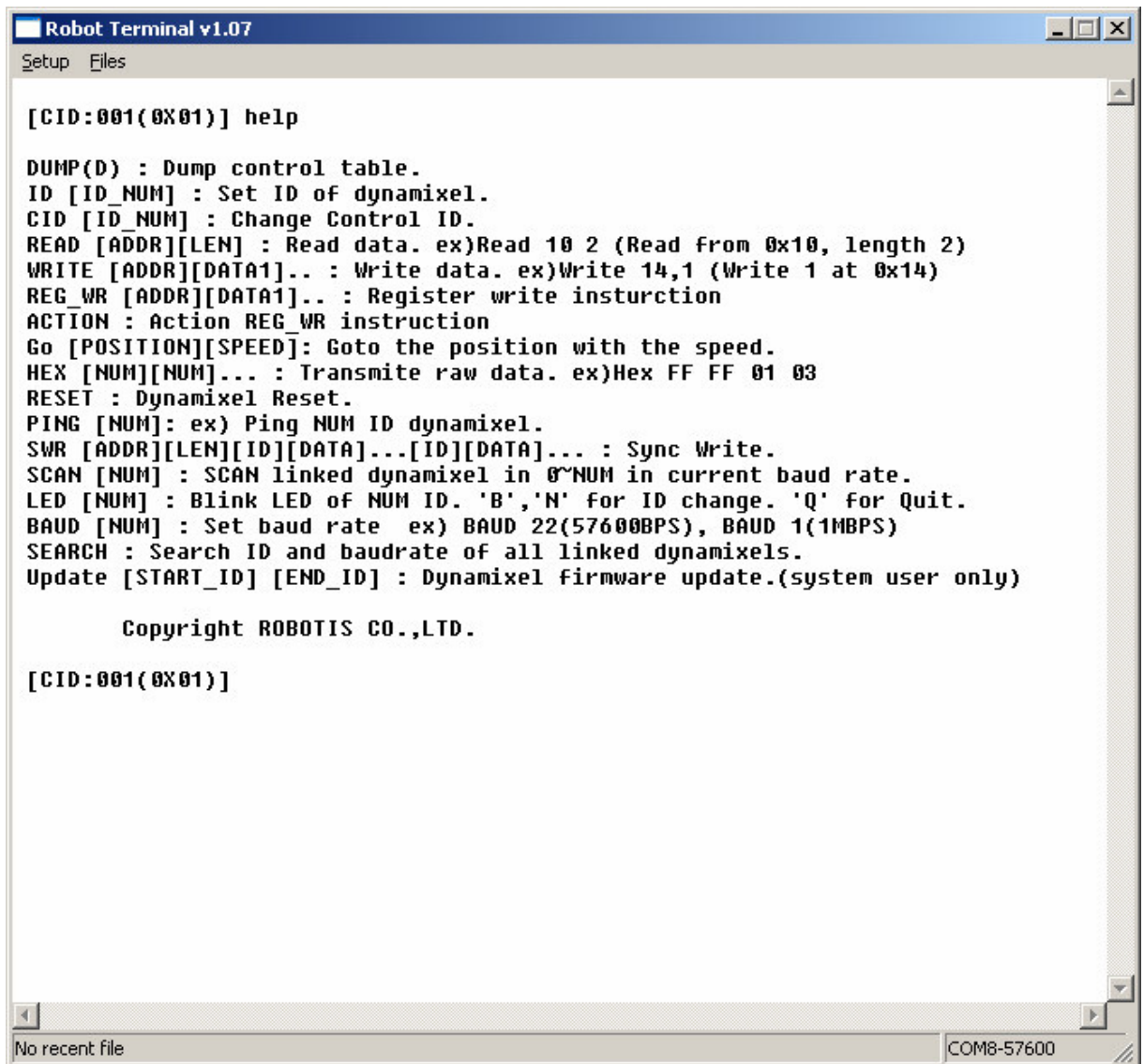


```
Robot Terminal v1.07
Setup Files

[CM-2 PLUS Version 1.14ab]
<->PC:57142 BPS, <->Dynamixel:1000000 BPS
ID:001 002 003 004
004(0X04) Dynamixels Found.
[ CID:001(0X01) ] _
```

Obrázek 26 Detekce firmwaru, rychlostí a zařízení při spuštění RobotTerminálu

Program je specifickým příkazovým řádkem, ve kterém definujeme příkazy pro jednotlivé motory, příklady příkazů můžeme vidět na [obr.27]



```
Robot Terminal v1.07
Setup Files

[CID:001(0X01)] help

DUMP(D) : Dump control table.
ID [ID_NUM] : Set ID of dynamixel.
CID [ID_NUM] : Change Control ID.
READ [ADDR][LEN] : Read data. ex)Read 10 2 (Read from 0x10, length 2)
WRITE [ADDR][DATA1].. : Write data. ex)Write 14,1 (Write 1 at 0x14)
REG_WR [ADDR][DATA1].. : Register write instruction
ACTION : Action REG_WR instruction
Go [POSITION][SPEED]: Goto the position with the speed.
HEX [NUM][NUM]... : Transmite raw data. ex)Hex FF FF 01 03
RESET : Dynamixel Reset.
PING [NUM]: ex) Ping NUM ID dynamixel.
SWR [ADDR][LEN][ID][DATA]...[ID][DATA]... : Sync Write.
SCAN [NUM] : SCAN linked dynamixel in 0~NUM in current baud rate.
LED [NUM] : Blink LED of NUM ID. 'B','N' for ID change. 'Q' for Quit.
BAUD [NUM] : Set baud rate ex) BAUD 22(57600BPS), BAUD 1(1MBPS)
SEARCH : Search ID and baudrate of all linked dynamixels.
Update [START_ID] [END_ID] : Dynamixel firmware update.(system user only)

      Copyright ROBOTIS CO.,LTD.

[CID:001(0X01)]
```

No recent file COM8-57600

Obrázek 27 Použité funkce pro ovládání motorů přes RobotTerminal

DUMP – vytiskne kontrolní tabulku námi zvoleného motoru

ID – pomocí tohoto příkazu nastavujeme ID motoru

CID – přepnutí na jiný motor

READ – při vypsání adresy a velikosti se zobrazí nastavení dané adresy

WRITE – zápis resp. změna nastavení adresy

REG_WR – zápis instrukce pomocí adresy a dat (využívá tzv. safe módu) zvolená adresa a data se do této adresy zapíšou, ale provedou se až po volání funkce ACTION

Go – instrukce pro otáčení motoru ve smyslu nastavení pozice a rychlosti

HEX – funkce využívaná pro převod hodnot z desítkové do hexadecimální a naopak

RESET – provede reset motoru a nastavení všech jeho parametrů do továrního nastavení

PING – instrukce pro ověřování správné komunikace mezi motorem a řídicí jednotkou

SWR – funkce využívající nastavování více motorů najednou

SCAN – zjišťuje počet zapojených motorů k řídicí jednotce

LED – pomocí této funkce může nastavovat LED diody na motoru, tovární nastavení je definováno na reakci při vzniku chyby

BAUD – měníme rychlost přenosu dat

SEARCH – vyhledá připojené motory

Update - slouží pro přehrání firmwaru motorů

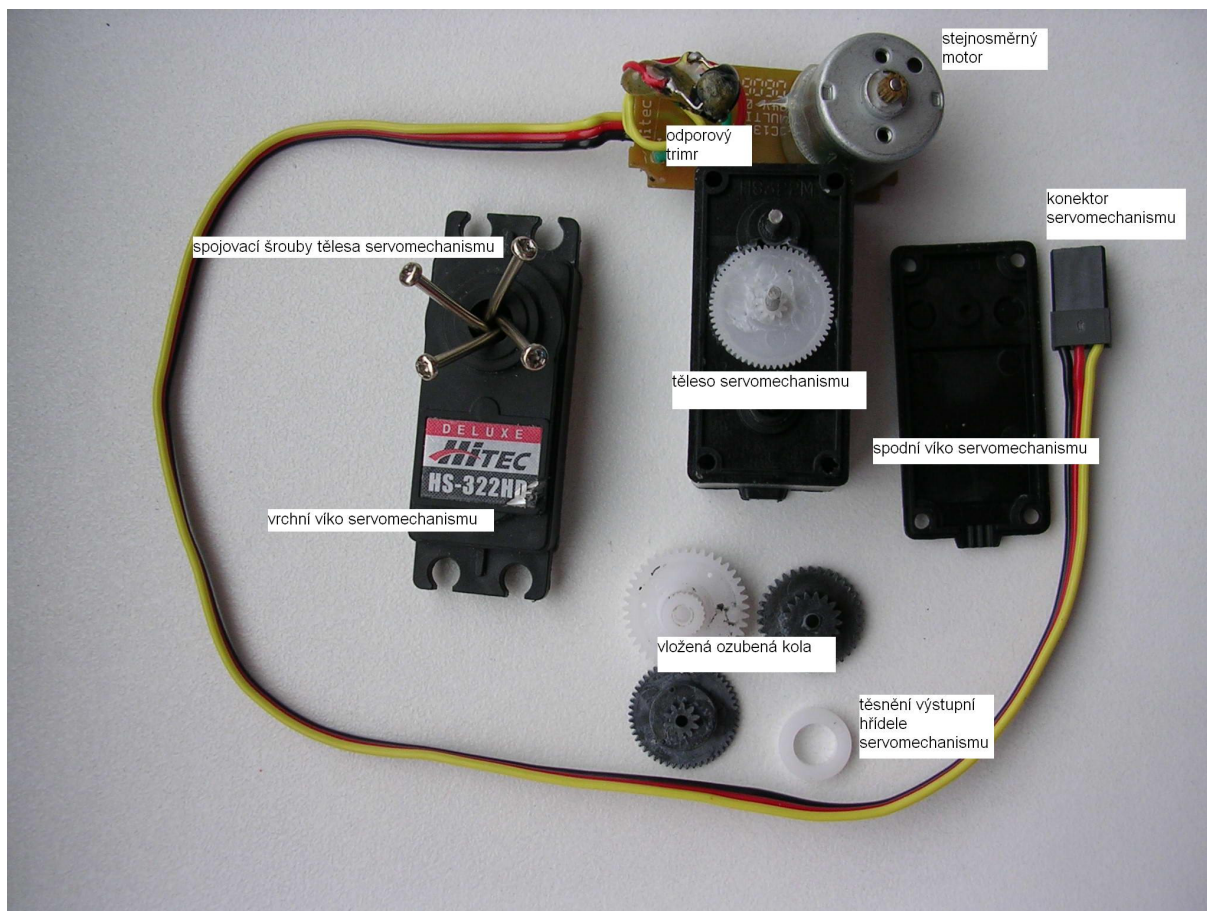
II. PRAKTICKÁ ČÁST

6 NÁVRH A REALIZACE MOBILNÍHO ROBOTNICKÉHO SYSTÉMU S MOTORY HS 322

Mobilní robotický systém s motory HS 322 byl vyvinut za účelem, co nejnižších finančních nároků. Pro sestavení podvozku bylo použito dvou modelářských servomechanismů. U mobilního podvozku bylo potřebné, aby se servomechanismy otáčeli kontinuálně, bohužel levnější verze servomechanismu HS 322, která byla použita, disponuje mechanickými dorazy. Mechanické dorazy brání kontinuálnímu otáčení a možný rozsah otočení je 0° až 180°. Na základě těchto informací bylo zapotřebí demontovat servomechanismus a tyto mechanické bloky odstranit.

6.1 Úprava servomechanismu

Úprava servomechanismu HS 322 je snadná a nevyžaduje žádné speciální vybavení. Úprava spočívá v demontáži zpětnovazebního potenciometru, odstranění mechanického rotačního dorazu výstupní osy servomechanismu a nahrazení zpětnovazebního potenciometru dvojicí rezistorů nebo odporovým trimrem.[7]



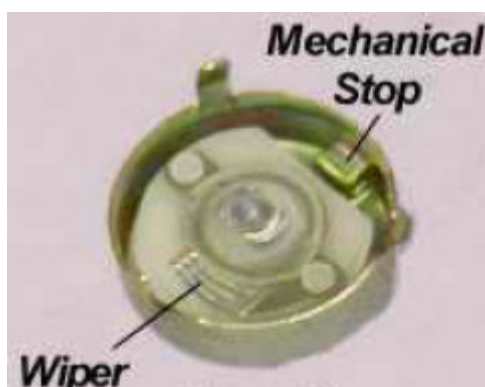
Obrázek 28 Rozložený servomechanismus HS 322

Po rozložení motoru HITEC HS 322 podle [obr. 28] byla provedena demontáž zpětnovazebního potenciometru z tělesa servomechanismu. Po odpájení přívodních vodičů a rozebrání potenciometru byla vyjmuta odporová dráha potenciometru, která byla zajištěna pomocí zahnutých plechových jazýčků [obr. 29]. Odporová dráha se vyjmula, jelikož pro kontinuální otáčení servomechanismu nespĺňuje žádnou funkci. [7]



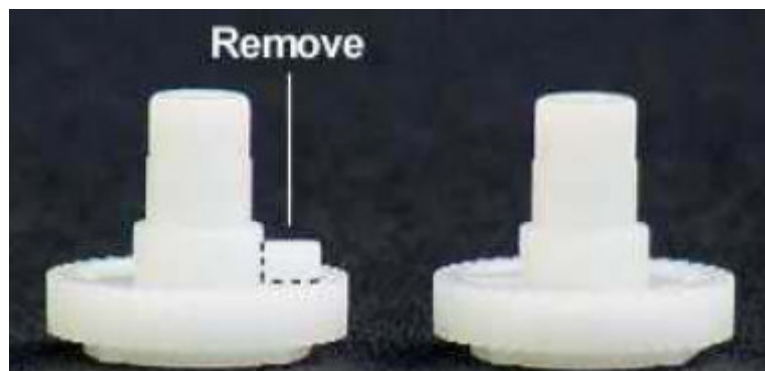
Obrázek 29 Demontáž potenciometru

Následně se odstraňoval sběrač (wiper) z plastového unašeče. Posléze byl odstraněn mechanický doraz (mechanical stop) v tělese potenciometru [obr. 30], který je tvořen prolisem v plechu tělesa potenciometru. Doraz se odstranil obroušením zubařskou frézou upnutou ve vrtačce.



Obrázek 30 Těleso potenciometru

Odříznutí nožem se odstranil výstupek na výstupním ozubeném kole servomechanismu, který tvoří mechanický doraz převodovky proti násilnému přetočení o více než 180°.



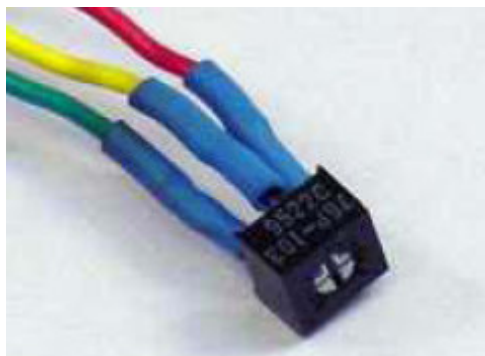
Obrázek 31 Výstupní ozubené kolo servomechanismu

Ze dvou rezistorů 2k7 lze vytvořit odporový dělič nahrazující zpětnovazební potenciometr, který po sléze je připájen na vodiče, na nichž byl původně potenciometr připojen. Rudý a zelený vodič jsou zapojeny na koncích děliče, žlutý vodič uprostřed děliče. Odporový dělič se zaizoluje pomocí lepicí pásky.

Místo odporového děliče se více používá miniaturní odporový trimr hodnoty 5k. Vývody trimru je nutno zaizolovat, v tomto případě nejlépe smršťovací bužírkou. Tímto trimrem lze jemně nastavit nulovou rychlost otáčení servomechanismu při šířce řídicího impulsu 1,5 ms. Poloha trimru po nastavení byla zajištěna zakápnutím nastavovacího šroubu speciálním voskem. Pro skutečně jemné nastavení lze použít trimr víceotáčkový.[7]



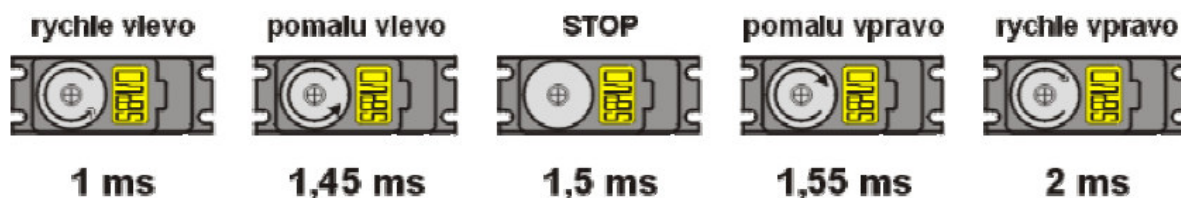
Obrázek 32 Odporový dělič 2x 2k7



Obrázek 33 Odporový trimr hodnoty 5K

Těleso potenciometru s odstraněným dorazem, odporovou dráhou a sběračem byl umístěn zpět do tělesa servomechanismu. Ozubená kola převodovky byly pečlivě složeny zpět v opačném pořadí, než se vyjímalý a dbalo se na jejich správné usazení a promazání. Servomechanismus se skládá bez použití násilí a zajišťuje se čtyřmi dlouhými šrouby. [7]

Posléze došlo k otestování funkcí servomechanismů. Pohyb hřídelí reagoval bez problémů a rychlost otáčení byla odvozena pomocí měření osciloskopu pulzně-šířkové modulace.[obr.34]



Obrázek 34 Způsob řízení servomechanismu[7]

6.2 Sestavení a oživení mobilního robotnického podvozku s HS 322

Správnost oživení a sestavení vycházelo s předešlé kapitoly o úpravě servomechanismu. Po úpravě se provedla montáž motorů na plastovou desku, která byla upravena výřezy pro nasazení kol na hřídel motoru. Po montáži motorů byl proveden ještě jeden výřez a to v zadní části podvozku. Výřez byl proveden za účelem nízkého profilu podvozku od země a montáži třetího vyvažovacího kola. Nasazením překlenovací plastové destičky se docílilo uchycení vyvažovacího kola. Nad touto plastovou destičkou bylo vyrobeno uchycení pro řídicí jednotku s procesorem AT89C2051. Dle obrázku. XX(v teoretické části). Rozložení se volilo, co nejoptimálněji pro budoucí další vývoj. Jelikož na

tento mobilní podvozek bude vestavováno ještě rameno a to v kooperaci s další diplomovou prací. Podvozek pak bude dále vybaven bateriemi, které budou napájet celou strukturu motorů a řídicí jednotku.

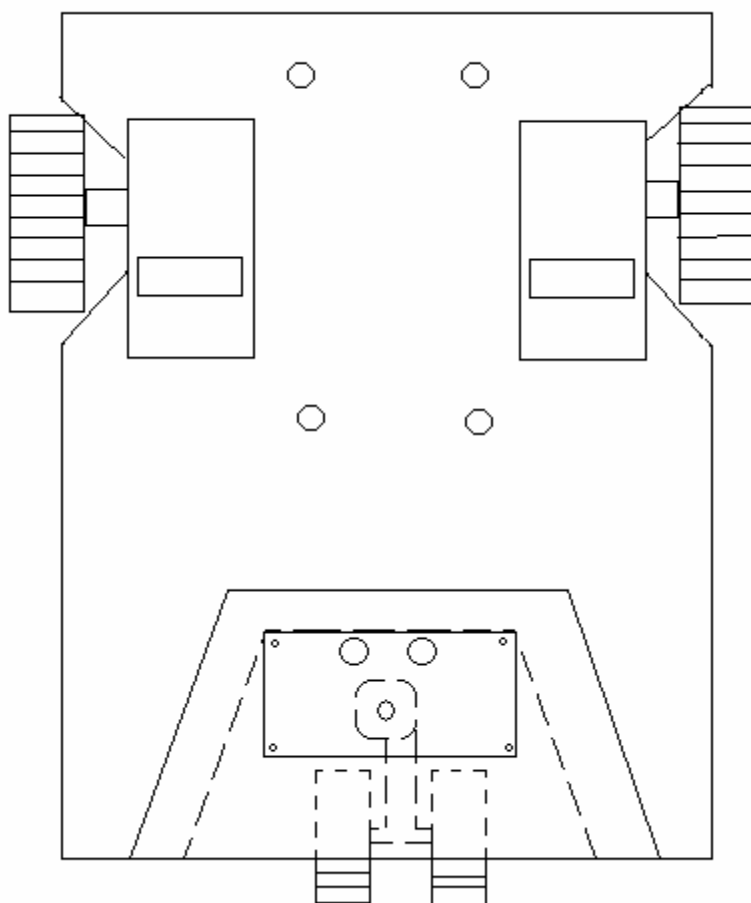
Navrhovaný systém podvozku je znázorněn na obrázku 35. Tento plán, byl použit jako vzor pro vývoj celého nízkonákladového mobilního robota. Verze tohoto podvozku i s výřezy a veškerými úpravami byla volena, co nejjednodušší a pro potřebu tvorby kopie lehce kopírovatelného modelu mobilního robota.

6.3 Oživení

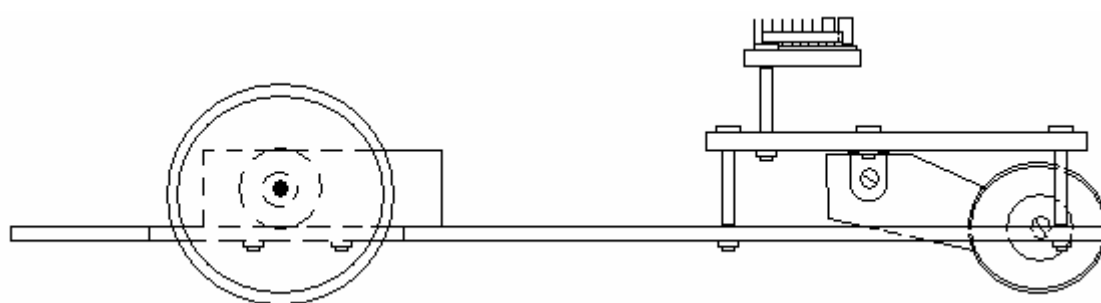
Oživení tohoto mobilního podvozku bylo bez problému. Důraz byl kladen na správné nastavení a osazením propojek po celé řídicí jednotce. Jednotka byla následně připojena k sériovému rozhraní. Jiné nastavení řídicí jednotky není zapotřebí. Pouze na straně počítače je nutné nastavit správný COM a přenosovou rychlost, ke kterému je připojena řídicí jednotka. Předešlá procedura je kompletní nastavení přenosových rychlostí a komunikace.

Dále bylo připojeno napájení pro řídicí jednotku a motory. Zde se použil nastavitelný zdroj napájení. Z důvodu nízkých napětí a proudového odběru je možné nerozběhnutí motorů, je velmi důležité zkontrolovat nastavení zdroje a to tak, aby nepřesahovalo rozsah povolený výrobcem. A to ani poddimenzování celého napájení. Motory při větším zatížení zvedají odběr proudu a to může zavinit selhání systému ve směru nefunkčnosti.

Dodržení veškerých výše popsanych bodů postupu zaručuje správnost funkce celého robotického systému s motory HS 322.



Obrázek 35 Návrh konstrukce mobilního podvozku pro motory HS 322 (nárys)



*Obrázek 36 Návrh konstrukce mobilního podvozku pro motory HS 322
(bokorys)*

7 DEFINOVÁNÍ SOFTWAREVÉHO VYBAVENÍ PRO ROBOTICKÝ PODVOZEK S HS 322

Ovládání mobilních podvozků s motory HS 322 a RX 64 bylo voleno za pomoci programovacího jazyku C# ve studiu SharpDevelop. Studio SharpDevelop je volně šiřitelné a poskytuje dostatečné vybavení programových komponent.

Programové vybavení vznikalo ve dvou částech. První částí bylo definování formy okna (design), otevření a nastavení sériového portu. Druhá část se zabývala tvorbou paketu a ovládáním mobilního podvozku.



Obrázek 37 Softwarové okno pro ovládání mobilního podvozku s motory HS

322

Popis softwaru:

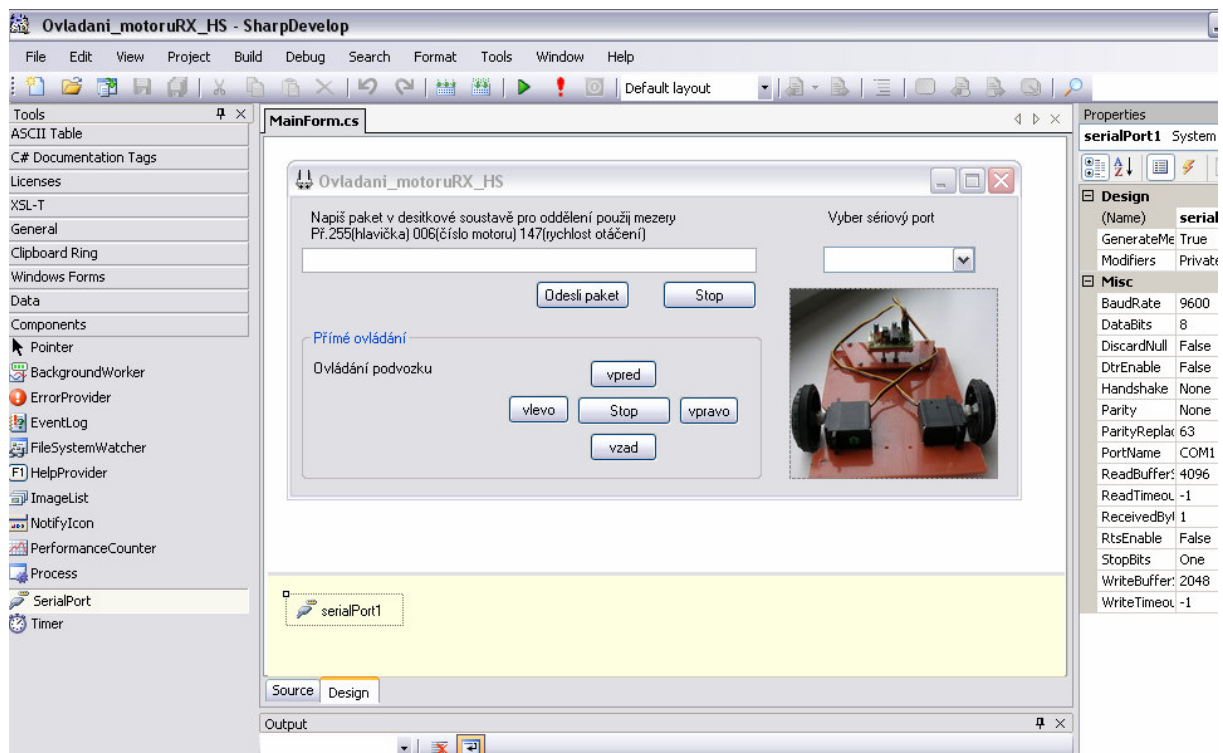
Vyber sériový port – funkce detekuje porty připojené k počítači a student si zvolí ten port, ke kterému má připojenu řídicí jednotku.

Příkazový řádek – zde je možnost vypsání paketu pro ovládání motoru ručně podle příkladu uvedeného nad příkazovým řádkem

Přímé ovládání – slouží k ovládání robotického podvozku, stiskem tlačítka se vykoná pohyb dle specifik tlačítka

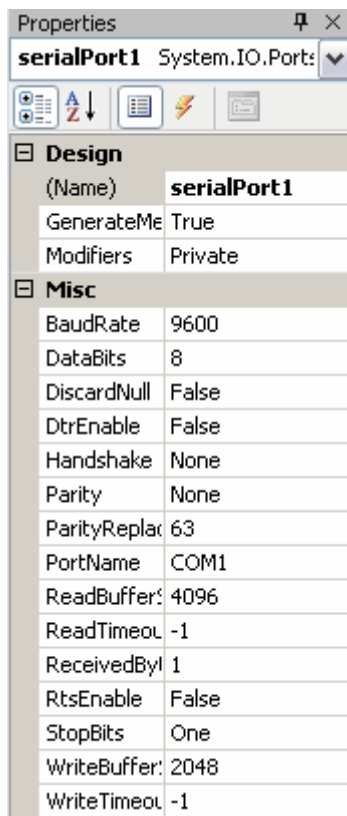
7.1 Design, otevření a nastavení sériového portu

První část softwaru byla vytvořena za pomoci designového přístupu, kdy jednotlivé komponenty byly vkládány do hlavního formu. Tímto postupem vzniklo „hlavní okno“ [obr.38], které se zobrazí při spuštění celého softwaru.



Obrázek 38 Okno Design v softwaru SharpDevelop

Dále došlo k definování sériového portu, který byl určen pro komunikaci mezi PC a řídicí jednotkou vybavenou procesorem AT89C2031. Nastavení se provedlo v oknu Design po spuštění vlastností sériového portu. [obr. 39]. Celkové nastavení bylo provedeno podle požadavků, které si kladli výrobci na správnou komunikaci mezi řídicí jednotkou a PC.



Obrázek 39 Nastavení sériového portu v SW SharpDevelop

V části „Misc“ se nastavovaly pouze proměnné BaudRate, DataBits a StopBits. Zbylé části nebylo nutné nastavovat. Použití bylo defaultní. Správná funkčnost nastavení funkce serialPort1 byla ověřena testováním.

7.2 Tvorba balíku a ovládání mobilního robotnického podvozku s HS

322

```

/*
 * Created by SharpDevelop.
 * User: Paja
 * Date: 2.2.2009
 * Time: 10:18
 *
 * */

```

```
// Hlavní část je zaměřena na definování jednotlivých komponent, které
program vyžaduje pro správnou funkčnost. Jednotlivé komponenty se
vytvořily při definování Designu okna//
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
namespace Ovladani_motoruRX_HS
```

```
// název celého Projektu s nastavením a vytvořením hlavního Formu
```

```
{
    public partial class MainForm : Form
    {
        string InputData = String.Empty;
        delegate void SetTextCallback(string text);
        public MainForm()
        {
            // Část zabývající se inicializací komponent a nastavením sériového portu
            // pro comboBox, ve kterém jsou zobrazeny jednotlivé sériové porty, které
            // počítač obsahuje.

            InitializeComponent();

            string[] ports = SerialPort.GetPortNames();

            foreach (string port in ports)
```

```

        {
            comboBox1.Items.Add(port);
        }
    }

//volání funkce comboBox, která odpovídá designu s nastavením a otevřením
sériového portu

    void ComboBox1SelectedIndexChanged(object sender, EventArgs e)
    {
        if (serialPort1.IsOpen) serialPort1.Close();

        serialPort1.PortName = comboBox1.SelectedItem.ToString();

//Testování, zda je sériový port otevřen

        try
        {
            serialPort1.Open();
        }

//Zaslání zprávy o nenavázání spojení se sériovým portem

        catch
        {
            MessageBox.Show("Sériový port " + serialPort1.PortName + " nemohl být
otevřen!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

            comboBox1.SelectedText = "";
        }
    }

    private string hex2binary(string hexvalue)
    {
        string binaryval;

        binaryval = textBox1.Text;

        hexvalue = Convert.ToString(Convert.ToInt32(binaryval, 16),
2);

        return hexvalue;

//Funkce pro převádění hodnot do hexa soustavy

```

```

        }

        void TextBox1TextChanged(object sender, EventArgs e)
        {
        }

//odesílání paketu

        void Button1Click(object sender, EventArgs e)
        {

//řetězec v textBox1 má tvar třeba "ff ff a1 05"
//metodou Split() se rozdělí podle znaku ' ' do pole
                String[] numbers = textBox1.Text.Split();

//vložení hodnot z textBoxu

//pole ve kterém bude výsledný příkaz

                byte[] cmd = new byte[numbers.Length];

//definování proměnné respektive pole

                try {

// každé hexa číslo převedu do desítkové soustavy

                        for (int i = 0; i < numbers.Length; i++)

cmd[i] =
byte.Parse(numbers[i], System.Globalization.NumberStyles.Integer);

//zadávání v šestnáctkové soustavě
//byte.Parse(numbers[i], System.Globalization.NumberStyles.HexNumber);

                }

// metoda Parse může zveřejnit výjimku, když řetězec není hexa
číslo, popř. desítkové číslo

                } catch (FormatException ex) {

MessageBox.Show("Špatně vložený znak či číslo", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);

                return;

        }

// pole s příkazem je převedeno na řetězec, čísla jsou oddělena mezerou

                String txt = "";

```

```

        foreach (byte b in cmd) {

            txt += b + " ";

        }

//Zobrazení hodnot, které jsou odesílány

MessageBox.Show(txt, "Odesílané hodnoty", MessageBoxButtons.OK,
MessageBoxIcon.Information);

//Zápis na port

        if (serialPort1.IsOpen) {

            serialPort1.Write(cmd, 0, cmd.Length);

        } else {

MessageBox.Show("Sériový port je zavřen!", MessageBoxButtons.OK,
MessageBoxIcon.Error);

        }

        textBox1.Clear();

    }

//jízda dozadu

void Button2Click(object sender, EventArgs e)

{

serialPort1.Write(new byte [] {0xff, 0x06, 0xA7}, 0, 3);

serialPort1.Write(new byte [] {0xff, 0x07, 0x57}, 0, 3);

}

//zastavení

void StopClick(object sender, EventArgs e)

{

serialPort1.Write(new byte [] {0xff, 0x06, 0x7F}, 0, 3);

serialPort1.Write(new byte [] {0xff, 0x07, 0x7F}, 0, 3);

}

//jízda vpřed

void Button3Click(object sender, EventArgs e)

{

serialPort1.Write(new byte [] {0xff, 0x07, 0xA7}, 0, 3);

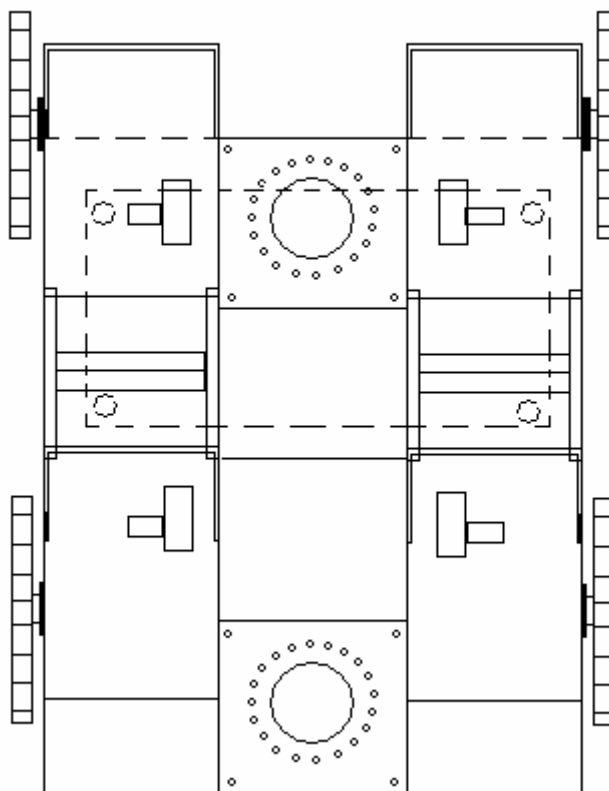
```

```
serialPort1.Write(new byte [] {0xff, 0x06, 0x57}, 0, 3);  
}  
  
//jízda doprava  
void Button4Click(object sender, EventArgs e)  
{  
    serialPort1.Write(new byte [] {0xff, 0x06, 0x57}, 0, 3);  
    serialPort1.Write(new byte [] {0xff, 0x07, 0x7F}, 0, 3);  
}  
  
//jízda doleva  
void Button5Click(object sender, EventArgs e)  
{  
    serialPort1.Write(new byte [] {0xff, 0x07, 0xA7}, 0, 3);  
    serialPort1.Write(new byte [] {0xff, 0x06, 0x7F}, 0, 3);  
}  
  
//stop  
void Button6Click(object sender, EventArgs e)  
{  
    serialPort1.Write(new byte [] {0xff, 0x06, 0x7F}, 0, 3);  
    serialPort1.Write(new byte [] {0xff, 0x07, 0x7F}, 0, 3);  
}  
}
```

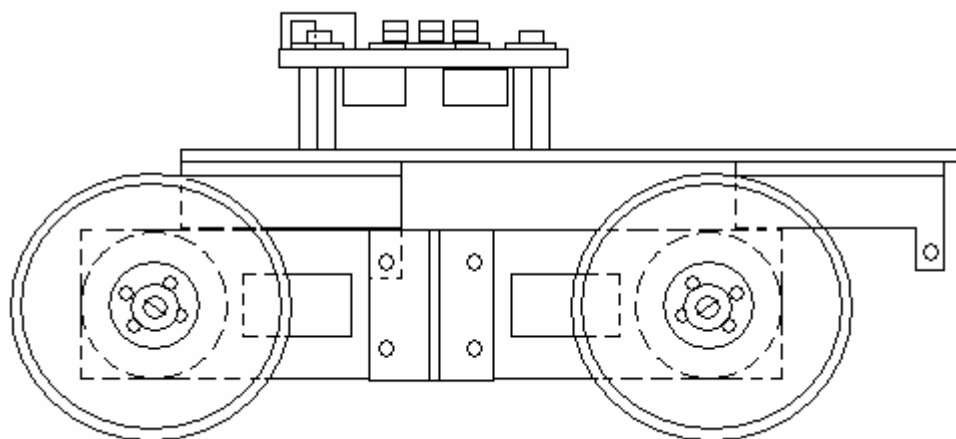
8 NÁVRH A REALIZACE MOBILNÍHO ROBOTNICKÉHO SYSTÉMU S MOTORY RX 64

8.1 Sestavení a oživení mobilního robotnického podvozku s RX 64

Mobilní robotický podvozek složený s motorů RX 64 byl sestaven za pomoci modulárních komponent, které byly dodány přímo s pohonnými jednotkami. Následně, s využitím plastové desky a jejímu uchycení k celé kostře motorů, vznikl prostor na upevnění řídicí jednotky. Sestavením všech těchto komponent dle nákresů, které byly voleny v prvotní fázi, byl sestaven kompletní mobilní podvozek s motory typu RX 64.



Obrázek 40 Návrh konstrukce mobilního podvozku pro motory RX 64 (nárys)



Obrázek 41 Návrh konstrukce mobilního podvozku pro motory RX 64 (bokorys)

8.2 Oživení

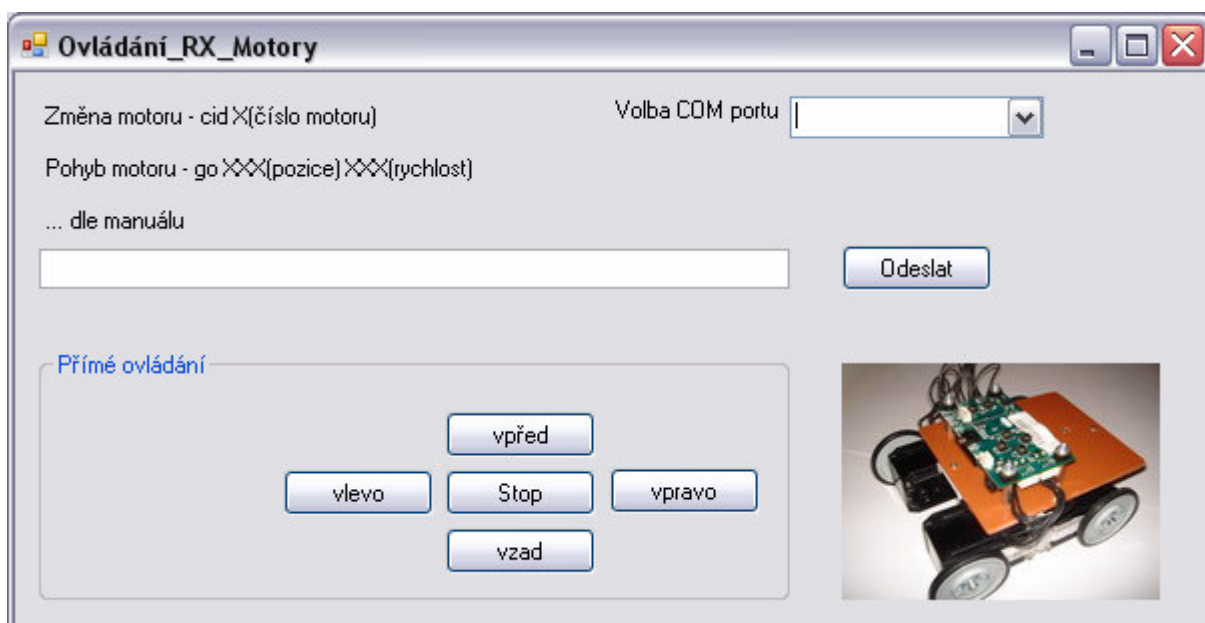
Testováním schopnosti motorů kontinuálního otáčení pro potřeby podvozku došlo k několika rozporům s doporučeními výrobce. Minimální napájecí napětí uváděné výrobcem uskuteční bezproblémový styk pouze mezi PC a řídicí jednotkou. Minimální napájení použité pro ovládání motorů zapříčiňuje nestabilitu celého systému. Tento nedostatek se projevil náhodným otáčením motoru. Po analýze problému bylo stanoveno minimální napájení na hodnotu 14V, které zabezpečuje správný chod celého pohonného systému.

V dalších částech testování mobilního robotického podvozku nedošlo k nalezení žádných závad nebo problémů zapříčiněných chybou výrobce.

9 DEFINOVÁNÍ SOFTWARE VYBAVENÍ PRO ROBOTICKÝ PODVOZEK S RX 64

K otestování programového vybavení na motorech HS 322 bylo s výhodou využito jazyku C# i pro nastavení ovládání motorů typu RX 64. Programové vybavení bylo přepracováno podle potřeb řady motorů RX a následně testován pro odstranění poruch v komunikaci. Použito bylo vývojové studio SharpDevelop. Postupným zpracováním podobného ovládacího okna a nastavením sériového portu, došlo k definování ovládacích příkazů pro motory typové řady RX.

Nezbytné bylo zachování příkazového řádku, který byl specifikován v softwaru výrobce a jeho základy byly použity i v softwaru definovaném pro podvozek. Byly také zvoleny jednoduchá tlačítka pro ovládání pohybu vpřed, vzad, vlevo a vpravo. Na základě jednoduchého ovládání si mohou studenti zkoušet snadné řízení pomocí tlačítek a porovnat s obtížností příkazového řádku a správně pochopit ovládání motorů RX 64.



Obrázek 42 Software pro mobilní podvozek s využitím motorů RX 64

Popis softwaru:

Zvol COM port – funkce detekuje porty připojené k počítači a student si zvolí ten port, ke kterému má připojenu řídicí jednotku.

Příkazový řádek – je obdobný s RobotTerminalem a využívá stejné příkazy

Přímé ovládání – slouží k ovládání robotického podvozku, stiskem tlačítka se vykoná pohyb dle specifik tlačítka

9.1 Tvorba paketu a ovládání mobilního robotického podvozku s RX 64

Použitý zdrojový kód vychází ze softwaru vytvořeného pro mobilní podvozek s motory HS 322. Zásadní rozdíl je v zasílání dat na sériový port. Data musí přímo odpovídat vloženým hodnotám přepsaným do ASCII znaků. Každému znaku, který je vepsán do příkazového řádku, odpovídá příslušná hodnota v ASCII kódu. Potvrzením je daný paket odeslán řídicí jednotce. Řídicí jednotka přijatý paket zpracuje a dle instrukcí v něm obsažených, odešle jednotlivé příkazy motorům RX.

```
/*  
 * Created by SharpDevelop.  
 * User: Paja  
 * Date: 4.2.2009  
 * Time: 13:08  
 *  
 */  
  
using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel;  
  
using System.Data;  
  
using System.Drawing;  
  
using System.Text;
```

```

using System.Windows.Forms;

using System.IO.Ports;

namespace Ovládání_RX_Motory
{
    /// <summary>
    /// Description of MainForm.
    /// </summary>

    public partial class MainForm : Form
    {
        string inputData = String.Empty;

        delegate void SetTextCallback(string text);

        public MainForm()
        {
            //

            InitializeComponent();

            string[] ports = SerialPort.GetPortNames();

            foreach (string port in ports)
            {
                comboBox1.Items.Add(port);
            }
        }

        void ComboBox1SelectedIndexChanged(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen) serialPort1.Close();

            serialPort1.PortName = comboBox1.SelectedItem.ToString();

            try
            {
                serialPort1.Open();
            }
        }
    }
}

```

```

        catch
        {
            MessageBox.Show("Sériový port " + serialPort1.PortName + " nemohl být
otevřen!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

            comboBox1.SelectedText = "";
        }
    }

    void Button1Click(object sender, EventArgs e)
    {
        if(serialPort1.IsOpen) serialPort1.WriteLine(textBox1.Text);

        else MessageBox.Show("Seriový port je zavřený!", "",
        MessageBoxButtons.OK, MessageBoxIcon.Error);

        textBox1.Clear();
    }

    void TextBox1TextChanged(object sender, EventArgs e)
    {
    }

    void Button2Click(object sender, EventArgs e)
    {

        if (serialPort1.IsOpen) serialPort1.Write("swr 30 4 1 0 0 0 0 2 0 0 0 0 3
0 0 0 0 4 0 0 0 0");

        serialPort1.Write(new byte [] {0x0D}, 0, 1);

    }

    void Button3Click(object sender, EventArgs e)
    {

        if (serialPort1.IsOpen) serialPort1.Write("swr 30 4 1 0 2 0 3 2 0 2 0 3 3
0 2 0 7 4 0 2 0 7");

        serialPort1.Write(new byte [] {0x0D}, 0, 1);

    }

    void Button4Click(object sender, EventArgs e)
    {

```

```
if (serialPort1.IsOpen) serialPort1.Write("swr 30 4 1 0 2 0 7 2 0 2 0 7 3  
0 2 0 3 4 0 2 0 3");  
  
serialPort1.Write(new byte [] {0x0D}, 0, 1);  
  
    }  
  
    void Button5Click(object sender, EventArgs e)  
  
    {  
  
if (serialPort1.IsOpen) serialPort1.Write("swr 30 4 1 0 2 0 7 2 0 2 0 7 3  
0 2 0 7 4 0 2 0 7");  
  
serialPort1.Write(new byte [] {0x0D}, 0, 1);  
  
    }  
  
    void Button6Click(object sender, EventArgs e)  
  
    {  
  
if (serialPort1.IsOpen) serialPort1.Write("swr 30 4 1 0 2 0 3 2 0 2 0 3 3  
0 2 0 3 4 0 2 0 3");  
  
serialPort1.Write(new byte [] {0x0D}, 0, 1);  
  
    }  
  
    }  
  
    }
```

10 ZÁVĚR

Práce předkládá řešení a realizaci dvou typů mobilních robotických podvozků, určených nejen pro pedagogickou a výzkumnou činnost Univerzity Tomáše Bati, ale i pro použití v Průmyslu komerční bezpečnosti. Základními moduly těchto robotických systémů jsou řídicí jednotky CM 2+ pro motory typu RX 64 a řídicí jednotka využívající mikrokontrolér AT89C2051 pro ovládání motorů HS 322.

Vývoj mobilního robotického podvozku s motory HS 322 byl zaměřen na snížení pořizovacích nákladů. Účelem bylo vytvořit univerzální funkční a modulární nosný podvozek pro bezpečnostní technologie i pro průmyslové využití. Projekt byl vypracován se zřetelem na rychlou sestavitelnost a jednoduché ovládání. Z hlediska potřeb PKB byla zvolena rychlost podvozku 5cm/s, která umožňuje využití optimální síly kroutícího momentu motorů (3Kg/cm), což dovoluje v daném uspořádání uvést baterie, řídicí jednotku a protiteroristické vybavení.

Mobilní robotický podvozek s pohonnými jednotkami typu RX 64 ovládaný řídicí jednotkou CM 2+ byl navržen se zřetelem na výukové účely v Laboratoři procesní a asistenční robotiky Fakulty aplikované informatiky. Perspektivně bude mobilní podvozek používán pro doktorské práce v oblasti technické kybernetiky. Jednotlivé komponenty, zejména pohonné jednotky RX 64 byly vybrány s ohledem na poměr cena/výkon v porovnání s jinými komerčními systémy, takže dosahují až 7x nižších pořizovacích nákladů oproti komerčním mobilním robotickým podvozkům, například Pioneer 3AT od výrobce AktiveMedia Robotic, USA.

Programové vybavení sestává ze souboru instrukcí pro komunikaci mezi PC a řídicími moduly robotických systémů a vlastním algoritmem pro řízení polohové adaptivity mobilních robotů. Programové vybavení bylo vytvořeno v prostředí SharpDevelop při použití jazyku C#. Software, který byl použitý na vývoj obou mobilních systémů, splňoval veškeré licenční smlouvy a dohody.

ZÁVĚR V ANGLIČTINĚ

This thesis brings design and development of two types of drive units for mobile robotic platform designed not only for educational and research activity of Thomas Bata University, but for use in commercial security industry as well. Basic modules of those robot systems are control units CM 2+ for actuators RX 64 and control units using micro-controller AT89C2051 for actuators HS 322 control.

Development of mobile robotic platform with HS 322 actuators was focused on decrease of purchase costs. The purpose was to create universal functional and modular base drive units for security technologies as well as for industrial applications. Project was focused in consideration of fast assembly and simple control. Speed of mobile robotic platform was set to 5 cm/s enabling to use optimal actuators torque force (3 Kg/cm) in term of commercial security industry needs, which allows in certain form to carry batteries, control unit and antiterrorist equipment.

Mobile robotic platform with RX 64 actuators controlled by control unit CM 2+ was developed in consideration of educational purposes in Process and assistive robotics laboratory on Faculty of applied informatics. The mobile robotic platform will be perspectively used for post-graduate thesis in the field of technical cybernetics. Individual components, mainly RX 64 driving units, was chosen with regard to the proportion of prize/effectivity in comparison to other commercial systems, so that sevenfold reduction of purchase costs against commercial mobile robotic platform can thus be achieved, for example Pioneer 3AT by AktiveMedia Robotic from the USA.

Software facility consists of instruction set for communication between computer and command modules of robotic system and self algorithm for controlling mobile robots' positional adaptability. Software facility was developed in SharpDevelop environment in C# language. Software used for both mobile systems development fulfils all licence contracts and agreements.

SEZNAM POUŽITÉ LITERATURY

- [1] NOVÁK, Petr. MOBILNÍ ROBOTY - pohony, senzory, řízení. 1. vyd. Praha : BEN, 2005. 248 s. ISBN 807300111-1.
- [2] LENGYEL, Kristyán. Ovládání robotického manipulátora mikrokontrolérem. Brno, 2007. 51 s. Vysoké učení technické Brno. Vedoucí bakalářské práce Richard Ružička.
- [3] ROTTA, Jiří. Robot manipulátor ROB 1-3 řízený z PC. A Radio. 1.1.2002, č. 12, s. 7-10.
- [4] BEZSTAROSTI, Jiří. WinSOS 2. Hobbyrobot [online]. 2006 [cit. 2008-05-12], s. 1-4. Dostupný z WWW: <www.hobbyrobot.cz>.
- [5] ROBOTIS: RX-64 MANUAL(ENGLISH) [online]. [cit.:neuveveno] dostupný z WWW :
<http://www.robotis.com/zbxe/?mid=software_en&category=7471&document_srl=543>
- [6] ROBOTIS: CM-2+ MANUAL(ENGLISH) [online]. [cit.:neuveveno] dostupný z WWW:
<http://www.robotis.com/zbxe/?mid=software_en&category=7471&document_srl=544>
- [7] Úprava modelářského servomechanismu [online] Dostupný z WWW:
<http://www.sweb.cz/e78/clanky/uprava_serva_cla.htm>
- [8] Bastlení Atari Portfolio [online] Dostupný z WWW:
<<http://vlastikd.webz.cz/bastl/servocontrol.htm>>
- [9] Robotics resources [online] 2005 roč. 8 č.3 str. 16-20. Dostupný z WWW:
<<http://hobbyrobot.cz/PDF/serva.pdf>>
- [10] HLAVÁČ V., SEDLÁČEK M. *Zpracování signálů a obrazů*. Praha: Vydavatelství ČVUT, 2000. ISBN 80-01-03110-1.
- [11] SONKA M., HLAVAC V., BOYLE R. *Image Processing, Analysis, and Machine Vision*. 2. vyd. PWS Publishing, Pacific Grove, 1999. ISBN 0-534-95393-X.

- [12] KVASNICA M. *Head Joystick and Interactive Positioning for the Wheelchair Proceedings of the ICOST 2003*. In 1th IEEE International Conference on Smart Homes and Health Telematics. Paris, 2003, France.
- [13] SURÝNEK, Tomáš. Určení vzdálenosti cíle hloubkoměrným principem se strukturovaným světlem. Zlín, 2008. 63 s. Vedoucí diplomové práce Ing. Mgr. Milan Kvasnica, CSc.
- [14] PETR, Ondřej. Řídicí systém mobilního dvoustopého robota. Zlín, 2006. 67s. Vedoucí diplomové práce Mgr. Ing. Milan Kvasnica, CSc .
- [15] NECKÁŘ, Pavel. Řízení motorů RX 64 pro polohovou adaptivitu mobilního robotického systému. In *Perspektivy elektroniky 2009 : 6. Celostátní seminář učitelů středních škol. Rožnov pod Radhoštěm : [s.n.], 2009. s. 1-7. ISBN 978-80-254-40.*
- [16] NECKÁŘ, Pavel. Řízení motorů RX64 pro polohovou adaptivitu mobilního robota. In *STOČ 2009. Ostrava : [s.n.], 2009. s. 1-7.*

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PKB Průmysl komerční bezpečnosti

BTSM Bezpečnostní Technologie Systémy a Management

PWM Pulsně šířková modulace

SEZNAM OBRÁZKŮ

<i>Obrázek 1 Mobilní robotický podvozek s HS 322</i>	18
<i>Obrázek 2 Motor HS 322</i>	19
<i>Obrázek 3 Ozubený převod motoru HS 322</i>	20
<i>Obrázek 4 Zapojení potenciometru do obvodu motoru HS 322</i>	20
<i>Obrázek 5 Řídicí jednotka s mikro-kontrolérem AT89C2051</i>	22
<i>Obrázek 6 Schéma zapojení řídicí jednotky pro ovládání motoru HS 322[3]</i>	24
<i>Obrázek 7 Nastavení polohy hřídele servomotoru HS 322</i>	25
<i>Obrázek 8 Prostředí WIN SOS</i>	26
<i>Obrázek 9 Program WIN SOS Communication</i>	27
<i>Obrázek 10 Program WIN SOS zvolení portu COM</i>	27
<i>Obrázek 11 Program WIN SOS nastavení komunikační rychlosti portu COM</i>	28
<i>Obrázek 12 Program WIN SOS hlavní panel</i>	29
<i>Obrázek 13 Tabulka programu WIN SOS se zaznamenanými polohami</i>	29
<i>Obrázek 14 Zobrazení posuvníku Steps a tlačítka Next pro ukládání poloh</i>	30
<i>Obrázek 15 Nastavení přehrávání</i>	30
<i>Obrázek 16 Nabídka File programu WIN SOS</i>	31
<i>Obrázek 17 Mobilní robotický podvozek s motory RX 64</i>	32
<i>Obrázek 18 Řídicí elektronika motoru RX 64</i>	33
<i>Obrázek 19 Zapojení motorů RX 64 s řídicí jednotkou a PC[5]</i>	33
<i>Obrázek 20 Řídicí jednotka CM 2+ (čelní strana)[6]</i>	39
<i>Obrázek 21 Řídicí jednotka CM 2+ (zadní strana)[6]</i>	40
<i>Obrázek 22 Schématické naznačení zapojení a identifikace motorů RX 64[6]</i>	42
<i>Obrázek 23 Konektor RX 64[6]</i>	42
<i>Obrázek 24 Nastavení připojení CM 2+</i>	44
<i>Obrázek 25 Tabulka nastavení portu COM a přenosové rychlosti</i>	44
<i>Obrázek 26 Detekce firmwaru, rychlostí a zařízení při spuštění RobotTerminálu</i>	45
<i>Obrázek 27 Použité funkce pro ovládání motorů přes RobotTerminal</i>	46
<i>Obrázek 28 Rozložený servomechanismus HS 322</i>	50
<i>Obrázek 29 Demontáž potenciometru</i>	51
<i>Obrázek 30 Těleso potenciometru</i>	51
<i>Obrázek 31 Výstupní ozubené kolo servomechanismu</i>	52

<i>Obrázek 32 Odporový dělič 2x 2k7</i>	52
<i>Obrázek 33 Odporový trimr hodnoty 5K</i>	53
<i>Obrázek 34 Způsob řízení servomechanismu[7]</i>	53
<i>Obrázek 35 Návrh konstrukce mobilního podvozku pro motory HS 322 (nárýs)</i>	55
<i>Obrázek 36 Návrh konstrukce mobilního podvozku pro motory HS 322 (bokorys)</i>	55
<i>Obrázek 37 Softwarové okno pro ovládání mobilního podvozku s motory HS 322</i>	56
<i>Obrázek 38 Okno Design v softwaru SharpDevelop</i>	57
<i>Obrázek 39 Nastavení sériového portu v SW SharpDevelop</i>	58
<i>Obrázek 40 Návrh konstrukce mobilního podvozku pro motory RX 64 (nárýs)</i>	64
<i>Obrázek 41 Návrh konstrukce mobilního podvozku pro motory RX 64 (bokorys)</i>	65
<i>Obrázek 42 Software pro mobilní podvozek s využitím motorů RX 64</i>	66

SEZNAM TABULEK

Tabulka 1 Složení paměti motoru a jednotlivých adres[5]	36
Tabulka 2 Popis částí řídicí jednotky CM 2+ (čelní strana).....	39
Tabulka 3 Popis částí řídicí jednotky CM 2+ (zadní strana).....	40