

Návrh a realizace zařízení na zachycení cíle pomocí kamerového systému a laseru polohovaného servomotory s více stupni volnosti

**Design and development of the device for the detection of a target using the multi
degrees of Freedom position adaptivity for the laser and camera system**

Bc. Martin Horák

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. Martin HORÁK**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Návrh a realizace zařízení na zachycení cíle pomocí kamerového systému a laseru polohovaného servomotory s více stupni volnosti**

Zásady pro vypracování:

1. Analýza vlastností pohonných jednotek HITEC HS 322 pro řízení konfigurace stavů zaměřovacího systému.
2. Návrh řídicí jednotky s použitím mikro-kontroléru AT89C2051.
3. Návrh řídicích algoritmů pro manipulaci s robotickým ramenem pomocí programového vybavení WIN SOS.
4. Návrh a realizace funkčního vzoru robotického zaměřovacího systému.
5. Implementace podpůrných algoritmů pro řízení konfigurace stavů zaměřovacího systému.
6. Ověření funkčnosti zaměřovacího systému.

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. ROTTA, J. A BEZSTAROSTI, J. 2001. Robot manipulátor. Praha : Amaro, 2001, 7s. 11s ISSN 1211-328x.
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. Stavební návod: Robotnické rameno se třemi stupni volnosti Ionline! Icit. 2009-01-15! Dostupný z WWW:(<http://hobbyrobot.cz/download.htm>)

Vedoucí diplomové práce:

doc. Ing. Milan Kvasnica, CSc.
Ústav elektrotechniky a měření

Datum zadání diplomové práce:

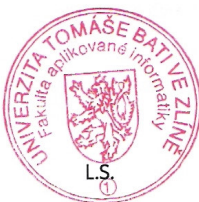
20. února 2009

Termín odevzdání diplomové práce:

22. května 2009

Ve Zlíně dne 20. února 2009


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. RNDr. Vojtěch Křesálek, CSc.
veditel ústavu

ABSTRAKT

Práce popisuje lokomoční systém robotického manipulátoru se třemi stupni volnosti. Je určen jako názorná pomůcka pro studenty při laboratorních pracích v oblasti protiteroristické robotiky. Pohyb ramene je realizován třemi modelářskými servomechanismy. Řídicí jednotka na bázi mikrokontroléru vytváří rozhraní sériový port - servomechanismus. Ovládání lokomočního systému je prováděno pomocí programového vybavení ROB3 implementovaného v osobním počítači. Při návrhu robotického manipulátoru byl kladen důraz na nízké pořizovací náklady s tím, aby byl dostupný studentům středních škol i při jejich mimoškolní tvořivé činnosti. Koncový efektor robotického ramene je opatřen zaměřovacím systémem rotujícího laseru a CCD obrazového snímače pro určení vzdálenosti předmětu.

Klíčová slova: servomechanismus, mobilní robot, pulzní–šířková modulace, mikrokontrolér, WINSOS2, ROB3, laser, CCD obrazový snímač, polohová adaptivita

ABSTRACT

The aim of this thesis is the design and development of locomotive system for robotic manipulator with three degrees of freedom destined for laboratory exercises in the field of antiterrorist robotics. The robotic arm has used three actuators to move. The control unit creates interface between serial port and servomechanism. The control of locomotive system is mediated using software tools ROB 3, which is implemented in the PC. During the design of robotic manipulator were minimized the cost in order to be accessible for education of secondary school students and also for their out-of-school leisure time creative activities. The robotic arm is equipped by the range finder with rotated laser target system and 2-D CCD array.

Keywords: servomechanism, mobile robot, pulse-width modulation, microcontroller, WINSOS2, ROB3, laser, 2-D CCD Array, positioning adaptivity

Poděkování:

Tímto bych chtěl především poděkovat svým rodičům za podporu během studia a vedoucímu diplomové práce doc Ing. Mgr. Milanu Kvasnicovi, CSc. za věcné připomínky a rady k dané problematice.

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	9
I TEORETICKÁ ČÁST	10
1 ROBOTICKÉ MANIPULÁTORY	11
1.1 DRUHY ROBOTŮ	12
1.1.1 Servisní roboty	13
1.1.2 Pick and Place roboty	13
1.1.3 Tele-operátory.....	13
1.2 KINEMATICKÁ KONSTRUKCE ROBOTŮ.....	14
1.2.1 Dělení podle kinematiky	16
1.2.1.1 Kinematika TTT	16
1.2.1.2 Kinematika RTT.....	17
1.2.1.3 Kinematika RRT.....	17
1.2.1.4 Kinematika RRR	18
1.2.2 Rozšíření o další osy.....	18
1.3 PROGRAMOVÁNÍ ROBOTŮ	19
1.3.1 Metoda Play-back.....	19
1.3.2 Metoda Teach-in	20
1.3.3 Metoda Off-line.....	20
2 MODELÁŘSKÝ SERVOMECHANISMUS	21
2.1 SERVOMECHANISMUS HS-322	21
2.1.1 DC motorek	22
2.1.2 Řídicí elektronika	22
2.1.3 Převodové ústrojí	23
2.2 ŘÍZENÍ SERVA	23
2.2.1 Závislost natočení hřídele na šířce PWM.....	24
3 ŘÍDICÍ JEDNOTKA	26
3.1 POPIS JEDNOTKY	26
3.1.1 Propojovací konektory	27
3.2 SCHÉMA ZAPOJENÍ ŘÍDICÍHO MODULU	28
3.3 GENEROVÁNÍ PWM SIGNÁLU.....	28
3.3.1 Generování pulzně šířkově modulovaného signálu na výstupu mikrokontrolér	29
3.4 KOMUNIKAČNÍ FORMÁT MEZI OSOBNÍM POČÍTAČEM A OVLÁDACÍM MODULEM	30
3.5 FREEWARE PROGRAM PRO OVLÁDÁNÍ SERVOMECHANISMŮ	31
3.5.1 Programování metodou Teach-in.....	31
3.6 FREE SERIAL PORT MONITOR	32
4 VÝROBA DESKY PLOŠNÉHO SPOJE ZA POMOCÍ FOTOCESTY	34

4.1	SVĚTLOCITLIVÝ POVRCH DESKY PLOŠNÉHO SPOJE.....	34
5	PROGRAMOVACÍ PROSTŘEDÍ SHARPDEVELOP 3.0	35
5.1	HLAVNÍ OKNO PROGRAMU	35
6	MĚŘENÍ VZDÁLENOSTI.....	38
6.1	MĚŘENÍ VZDÁLENOSTI ZA POMOCÍ LASERU A WEBKAMERY	38
6.1.1	Princip činnosti.....	38
II	PRAKTICKÁ ČÁST	40
7	REALIZACE ROBOTICKÉHO RAMENE.....	41
7.1	KONSTRUKČNÍ PROVEDENÍ.....	41
7.2	REALIZACE ŘÍDICÍHO MODULU	44
7.2.1	Zapojení komunikačního kabelu.....	45
8	PROGRAMOVÉ VYBAVENÍ.....	47
8.1	PROGRAM ROB3	47
8.2	POPIS KÓDU PROGRAMU ROB3	48
8.3	VIDEOOCX	54
8.3.1	Ovládací příkazy.....	55
8.4	REALIZACE LASEROVÉHO ZAMĚŘOVAČE.....	56
8.4.1	Programové vybavení	56
8.5	PROGRAMOVÝ KÓD LASEROVÉHO ZAMĚŘOVÁNÍ.....	57
	ZÁVĚR.....	63
	ZÁVĚR V ANGLIČTINĚ	63
	SEZNAM POUŽITÉ LITERATURY	65
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	67
	SEZNAM OBRÁZKŮ	68

ÚVOD

Pro dnešní dobu, ve které žijeme, je typické využití technických vymožeností při každodenní činnosti jako samozřejmost s tím, že většina vynálezů má za úkol ušetřit lidskou námahu. Lidská pracovní síla je v současnosti ve velké míře nahrazována automatizovanými systémy. Člověk tak pouze systémy programuje a udržuje v provozuschopném stavu. Robotické systémy pracují bez únavy rychle a přesně s vysokou efektivitou a spolehlivostí. Robotické systémy jsou stále více sofistikovanější a proto kladou vysoké nároky na obsluhu. Robotika je v převážné míře předmětem studia na vysokých školách technického směru, přičemž zřízení robotické laboratoře není levnou finanční záležitostí. Střední školy tedy z finančních důvodů nemají možnost poskytovat takovéto vzdělání svým studentům. Možnost rozšíření výuky robotiky na střední škole by zcela jistě vyvolalo u více mladých lidí zvýšený zájem o tuto problematiku, takže by se z nich po absolvování vysoké školy stali renomovaní odborníci stále požadovaní na trhu práce. Na základě této marketingové úvahy předkládám v této práci metodiku vývoje a výroby levného a z hlediska součástkové základny dostupného robotického systému, vhodného pro studenty středních škol jak pro výuku, tak i pro mimoškolní tvůrčí aktivity. Základním požadavkem na výrobu byla co nejnižší pořizovací cena. Popisovaný robotický systém může být zhotoven v praktické části vyučování a přiblížit problematiku polohové adaptivity při výuce robotiky. Jeho výhodou je nízká pořizovací cena jednotlivých komponentů, dovolující jeho rozšíření, což umožní pochopení složité problematiky řízení polohově adaptivních systémů.

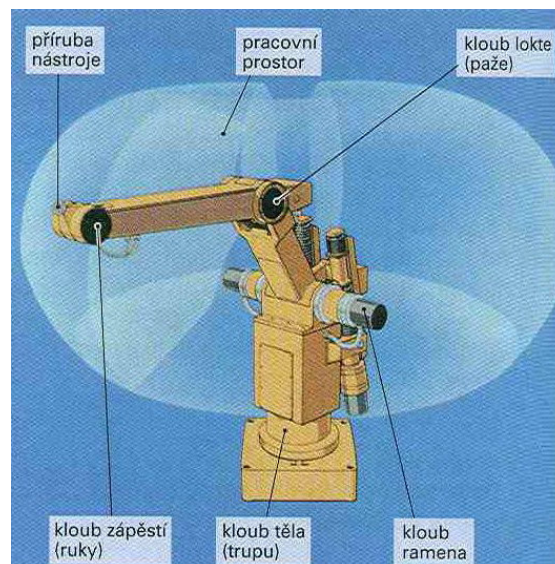
I. TEORETICKÁ ČÁST

1 ROBOTICKÉ MANIPULÁTORY

Pojem robot zavedl český spisovatel Karel Čapek v roce 1920 ve své divadelní hře R.U.R. (Rossums Universal Robots), která byla futuristickou vizí světa ovládaného umělými bytostmi nazývanými roboty, sestavenými původně jen pro vykonání těžké práce.[1]

Dvacátá léta minulého století byla ve znamení nástupu hromadné výroby. Rytmičká pásová výroba, rozdělená na jednotlivé navazující operace, dosud nebyla v plné míře odstraněna. Pracovní činnost člověka je přitom často omezena na rychlé a monotónní úkony podřízené rytmu stroje nebo dopravního pásu a je charakterizována vysokým pracovním zatížením v nepříznivých pracovních podmínkách. To podmínilo vznik nového slova „ROBOT“, odvozeného z českého i slovenského slova „robota“, charakterizujícího monotónní, těžkou, nezajímavou a nedobrovolnou práci nevolníků ve středověku. [1]

V průběhu minulého století došlo k intenzivní robotizaci sériových výrob. Řízení robotů není jednoduchou záležitostí, protože výsledná trajektorie, kterou opisuje rameno, (Obr. 1) je závislá na pohybech ve všech kloubech. Náročnosti těchto operací proto musí odpovídat i výkonný řídicí počítač s dostatečně výkonným procesorem. [1]



Obr. 1 Robotické rameno[1]

Rozvoj robotiky, stejně jako převážná většina věd, byl podmíněn vojenskými aplikacemi, zejména úlohou zaměření pohybujícího se cíle, která přenesla na palebnou

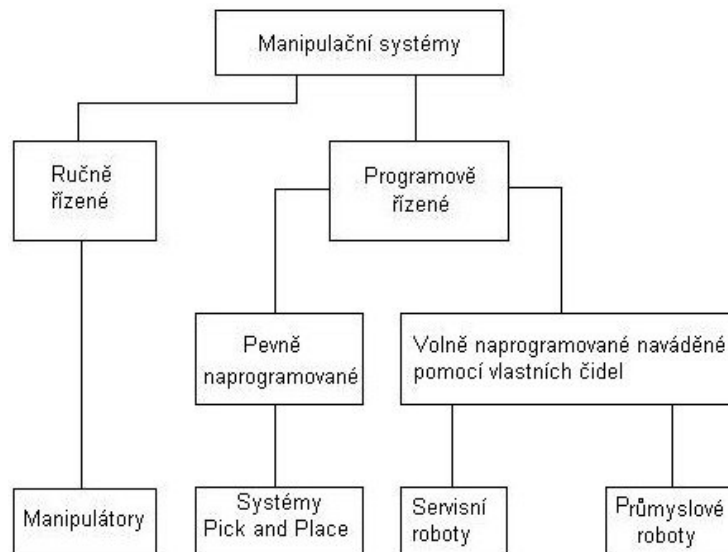
manipulační techniku operující ve dvou stupních volnosti, odměr a náměr. Údaje o odměru a náměru jsou postačující pro přímé zasažení cíle. Při použití granátu s časovanou náloží, po explozi zvyšující pravděpodobnost zásahu, je nezbytné vyhodnotit vzdálenost jako třetí stupeň volnosti, přepočtený při známé rychlosti střely na parametr známý pod názvem časování. Se zvyšující se rychlostí pohyblivých cílů koncem II. světové války, zejména letadel, selhávaly zaměřovací systémy v důsledku omezené rychlosti reakce lidského operátora a časového zpoždění vyhodnocování v elektromechanických analogových výpočetních systémech. Použití vakuových obrazových snímačů i monitorů, známých již od dvacátých let minulého století, umožňovalo on-line sledování vzdušných cílů pohybujících se i nadzvukovou rychlostí, ale chybělo dostatečně rychlé zařízení na výpočet predikce parametrů dráhy letu pro zaměření palebných prostředků. Tím byl podmíněn vývoj prvního elektronkového počítače ENIAC (Electronic Numerical Integrator And Computer), s architekturou v podstatě shodnou s PC, který byl prvotně použitý na řízení palby protiletadlových palebných prostředků. Cílem této práce je robotické rameno se dvěma stupni volnosti, které udržují polohu obrazu sledovaného cíle ve středu plošného obrazového snímače pomocí dvou servomotorů, (perspektivně i na zaměření a sledování cíle, s průběžným vyhodnocováním okamžitých hodnot odměru, náměru a vzdálenosti (časování) pomocí kamery vybavené plošným CCD (CMOS) obrazovým snímačem).

1.1 Druhy robotů

Přestože manipulační systémy jsou podobného vzhledu, liší se od sebe řídicím systémem, způsobem programování a v poslední řadě účelem (Obr. 2).

Rozlišení manipulátoru:

- Pick and Place (uchop a polož)
- Servisní roboty
- Průmyslové roboty [1]



Obr. 2 Rozdělení manipulačních systémů

Pohyby robotů jsou z hlediska sledu pohybů a tvaru dráhy pohybu volně programovatelné. Mechanické vymezení rozsahu pohybů např. pomocí koncových spínačů se používá převážně jen u pevně programovatelných robotů. Adaptivní robotický systém se totiž orientuje při pohybu mezi překážkami pomocí vlastních snímačů.

1.1.1 Servisní roboty

Jedná se většinou o samostatně pohybující se roboty, které provádějí pracovní úkony, popřípadě přemisťují anebo dopravují břemena. Příkladem může být robot šplhající po hladkých pláštích budovy umývací okna.

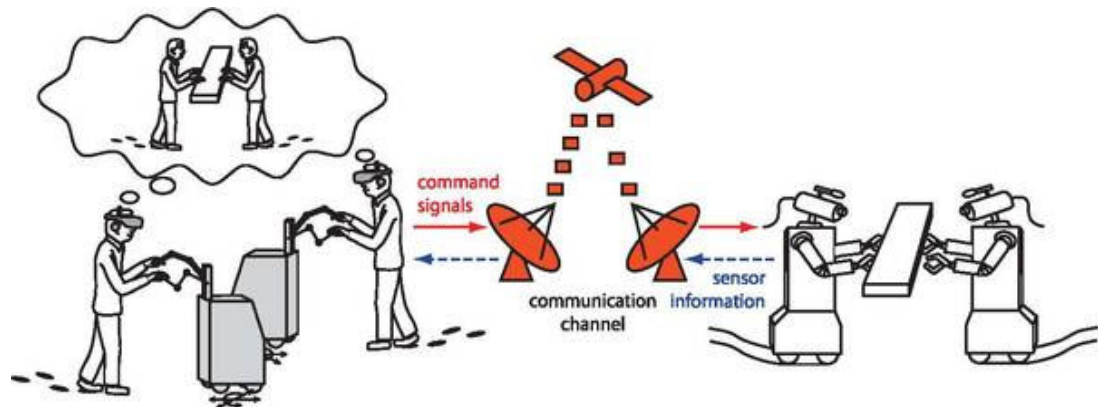
1.1.2 Pick and Place roboty

Tyto robotické systémy se používají pro opakující se přemisťování předmětů o shodném rozměru mezi stejnými místy. Příkladem jsou manipulační zařízení pro přemisťování těžkých kusů materiálu do kovacího lisu nebo těžkých násad a nástrojů pro bourací práce.

1.1.3 Tele-operátory

Dálkově ovládaný tele-operátor (Obr. 3.) je ovládaný ručně v otevřené regulační smyčce s lidským operátorem a pohyb je kontrolovaný za pomoci kamery, která přenáší

dění okolo manipulátoru do zobrazovacího monitoru obsluhy. Tele-operátory jsou využívány v místě, kde by nemohla být lidská obsluha, a proto musí být robot ovládán z bezpečného místa. Jedná se hlavně o manipulaci v nebezpečných prostorách s radioaktivními látkami.

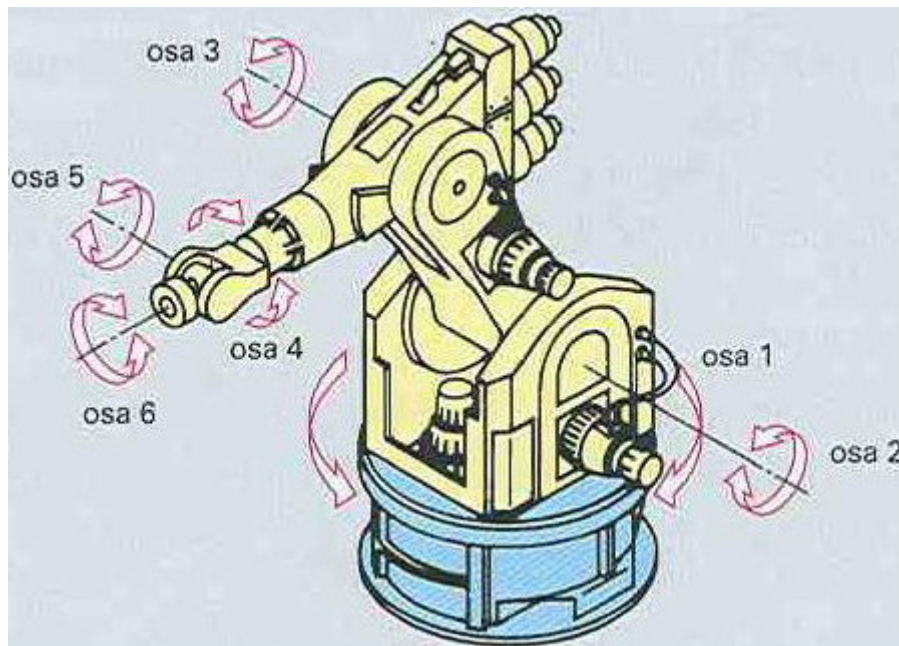


Obr. 3 Princip dálkově ovládaných manipulátorů [1]

1.2 Kinematická konstrukce robotů

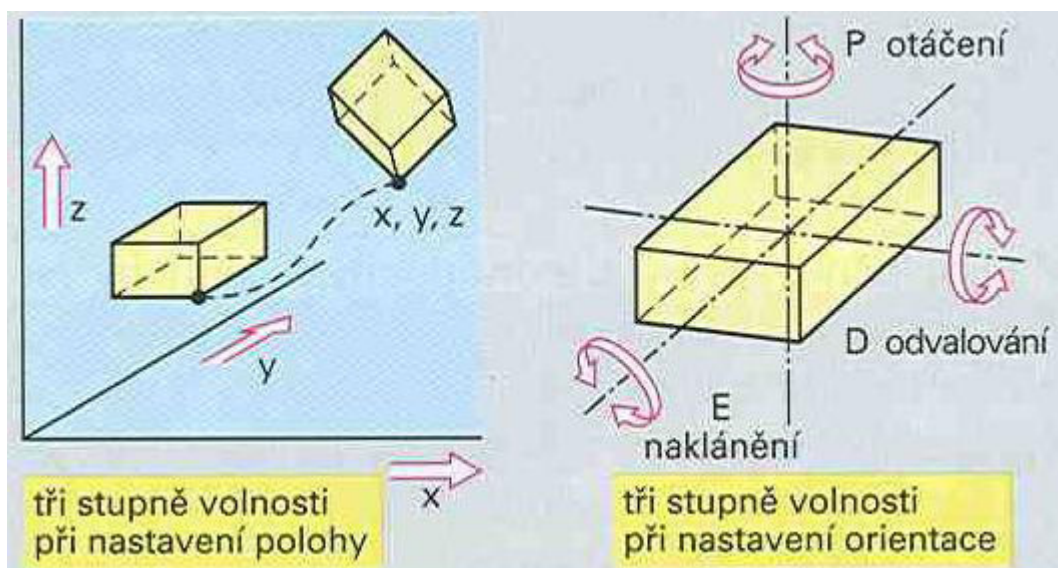
Konstrukce robotu (vnější vzhled), pracovní rozsah, použitelnost a pracovní požadavky jsou určeny vzájemným uspořádáním a počtem jednotek generující pohyb. Pohyblivé vlastnosti jsou dány počtem rotačních os (R – osy) a počtem translačních os (T – osy). [1]

Aby robot mohl dosáhnout kteréhokoliv bodu v dosahu svého manipulačního prostoru, potřebuje k tomu alespoň tři osy (Obr. 4). Tyto osy se jmenují hlavní osy, které jsou součástí ramene robotu. K nastavení úchopu nebo nástroje do libovolné polohy nebo směru jsou potřeba další tři osy, nazývané manipulační osy (Obr. 4). Tyto osy jsou vždy rotační. [1]



Obr. 4 Robot s pohyby v šesti osách [1]

K nastavení pozice v prostoru jsou zapotřebí tři stupně volnosti (translační, rotační, nebo jejich kombinace), např. ve směrech os X, Y, Z pravoúhlého souřadnicového systému a k nastavení orientace (uchopeného předmětu nebo nástroje) jsou potřeba také tři stupně volnosti v otáčení kolem tří nezávislých os. Rotace se někdy rozlišují názvy otáčení kolem svislé osy P, naklánění kolem osy E a odvalování kolem osy D (Obr. 5). [1]



Obr. 5 Pohyb s šesti stupni volnosti [1]

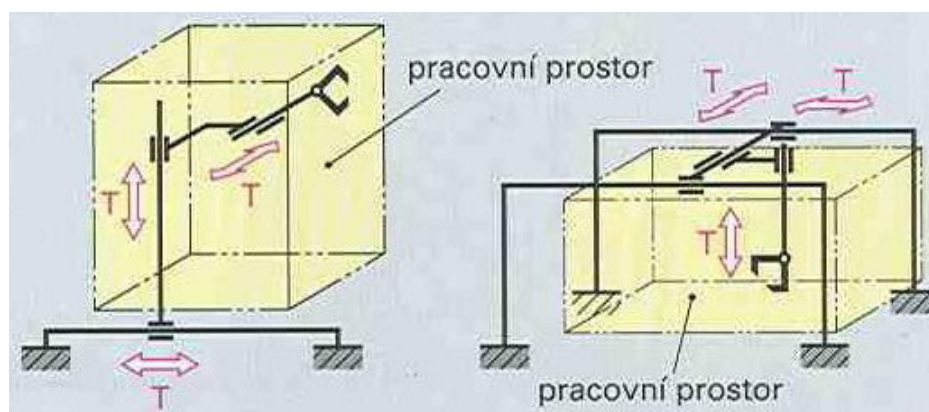
1.2.1 Dělení podle kinematiky

- Kinematika TTT
- Kinematika RTT
- Kinematika RRT
- Kinematika RRR

1.2.1.1 Kinematika TTT

Je řazena do skupiny základní kinematiky robotů. Pohyb se skládá ze tří translačních vzájemně kolmých pohybů (Obr. 6). Tento typ pohybu se uplatňuje u portálových překládacích systémů. Pracovní prostor má tvar kvádra s hranami délek odpovídajících rozsahům pohybu v osách X, Y, Z. Lienární pohyby lze skládat z rovnoměrných pohybů ve třech osách. [1]

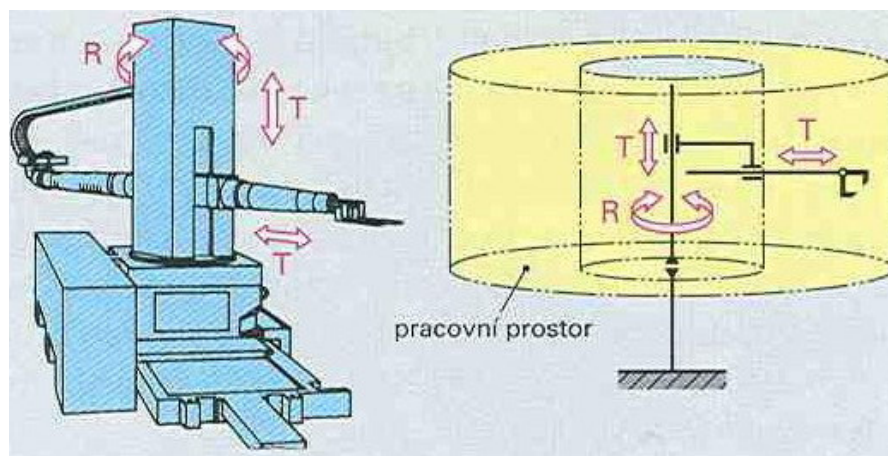
Portálové roboty mohou být konstruovány jak pro malý pracovní prostor, tak pro rozsáhlé pracovní plochy. Kinematika TTT je využita pro konstrukci některých obráběcích NC strojů. [1]



Obr. 6 Robot s kinematikou TTT [1]

1.2.1.2 Kinematika RTT

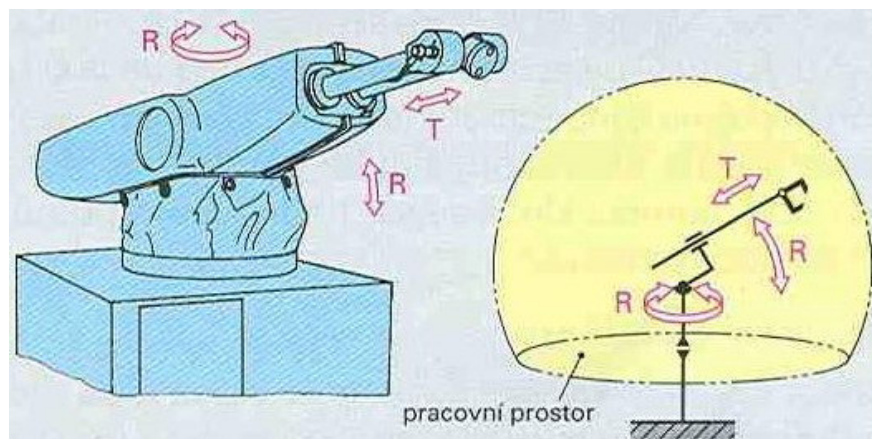
Tato kinematika je složena z jednoho rotačního pohybu a ze dvou translačních pohybů (Obr. 7). Rotační osu konstrukce robota tvoří otočná věž. Nastavení výšky tvoří 2. osa a 3. osa tvoří nastavení radiální vzdálenosti od osy rotace. Oproti kinematice TTT je zde pracovním prostorem válec. Pro správné natočení otočné věže je nutné provést transformaci ze zadaných kartézských souřadnic na cylindrické souřadnice. [1]



Obr. 7 Robot s kinematikou RTT [1]

1.2.1.3 Kinematika RRT

Roboty s kinematikou RRT (Obr. 8) mohou používat 1. svislou osu k náklonu, pro natočení okolo své osy slouží osa, která je kolmou ke svislé ose a jako 3. osu pak přímou translační osu změny vzdálenosti od konce ramena otočného ve dvou osách. Pracovní prostor tohoto systému je ohraničen kulovou plochou a rovinou. Při řízení je zapotřebí transformovat kartézské souřadnice na sférické souřadnice. [1]



Obr. 8 Robot s kinematikou RRT [1]

1.2.1.4 Kinematika RRR

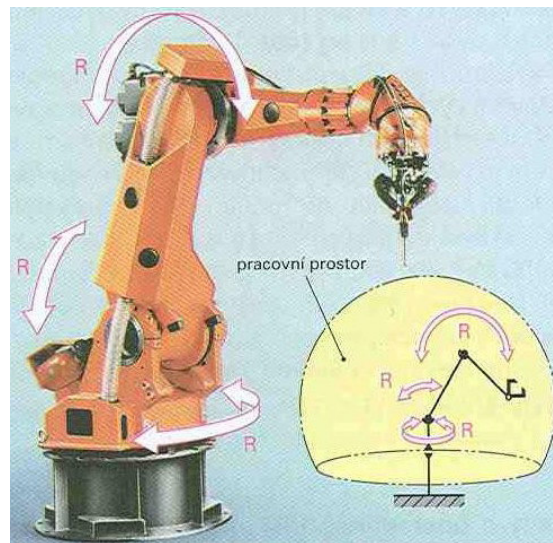
V poslední řadě je uveden poslední typ kinematické konstrukce, který se skládá ze třech rotačních pohybů. Konstrukční řešení (Obr. 9) je provedeno jako trojdílné rameno se třemi otočnými klouby. Většina robotů využívá právě této konstrukce. Kloubové roboty zabírají v poměru k velikosti kulového pracovního prostoru nejméně místa ze všech robotů a potřebují k určitému přemístění za určitou dobu ve srovnání s ostatními roboty nejmenší zrychlující síly. [1]

Kloubové roboty se vyrábějí v různých velikostech s nosností od 1 kg do 200 kg. Pohony os na konci ramene jsou umístěny v prostoru prostředního kloubu a pohyb je přenášen buď pomocí kloubových hřídelí nebo pomocí ozubených řemenů. [1]

1.2.2 Rozšíření o další osy

Roboty je možné rozšířit ještě o další osy pro zvětšení operačního rozsahu. Děje se to přidáním robotu na pojízdný podvozek, pohybující se po kolejkách. Díky tomu je schopen robot upravovat velké obrobky, jako například list vrtule větrné elektrárny. Použitím naklápěcího stolu pro upnutí výrobku získáme další osu. [1]

Získáváním více stupňů volnosti je možné skládat pracovní pohyby z menších počtů dílčích pohybů v jednotlivých osách.



Obr. 9 Kloubový robot s kinematikou RRR

1.3 Programování robotů

Programování robotů je ve srovnání s programováním NC strojů podstatně složitější, protože je potřeba programovat kromě pozic také orientaci nástroje.

Hlavní metody programování:

- Play-back
- Teach-in
- Off-line

1.3.1 Metoda Play-back

Používá se všude tam, kde není kladen nárok na velkou přesnost pohybu. Použití převážně u robotů v lakovnách. Programování je prováděno manuálním vedením robotického ramene a „ukazováním“ pohybu, co má vykonávat. Řídicí jednotka v určitém časovém intervalu (řádově ms) zapisuje do paměti údaje o pozici stříkací pistole. Uložený cyklus je potom spuštěn a robot pracuje samostatně. Vytvoření programu je rychlé a jednoduché. [1]

1.3.2 Metoda Teach-in

Metoda postupným učením, kdy je rameno robotu postupně naváděno obsluhou za pomoci tlačítek obslužného panelu, popřípadě joysticku programovacího přístroje, do postupných pozic a souřadnice těchto přesně vyladěných poloh i orientace úchopu jsou ukládány do paměti. Při automatické činnosti pak robot využívá zaznamenaných souřadnic a postupných poloh. [1]

1.3.3 Metoda Off-line

Tato metoda patří k nejtěžšímu a nejzdlouhavějšímu programování. Používají se k tomu různá programovací, vývojová a simulační prostředí. Program musí být před uložením do řídicí jednotky ramene odladěn. [1]

2 MODELÁŘSKÝ SERVOMECHANISMUS

Mezi konstrukčně jednoduché a levné polohovací jednotky je možno zařadit právě modelářské servomechanismy. Prioritním určením servomechanismů, jak sám název napovídá, je v modelářském odvětví. Na trhu je nepřeborné množství výrobců. Mezi nejznámější patří firma Futaba nebo firma Hitec. S oblibou jsou servomotory používány i pro stavbu různých robotů, protože pořizovací cena je řádově stovky korun, a řízení motorů není příliš obtížné.

Pro výrobu robotického ramene uvedeného v této práci byly zakoupeny servomechanismy od firmy Hitec, typové označení HS 322 (Obr. 10). V následující kapitole bude pojednáno o vlastnostech tohoto servomechanismu a způsobu řízení.



Obr. 10 Modelářský servomechanismus [6]

2.1 Servomechanismus HS-322

Celý servomechanismus je složen ze čtyřech základních částí (Obr. 11):

- Malý stejnosměrný motorek
- Řídicí elektronika
- Zpětnovazební potenciometr
- Převodové ústrojí

Parametry servomechanismu:

- Napájecí napětí: 4,8 – 6 V
- Síla: 3,7 kg/cm při napájecím napětí 6 V
- Rychlost: 0,15s / 60° při napájecím napětí 6 V
- Rozměry: 40x20x36,5 mm
- Hmotnost 43 g



Obr. 11 Konstrukční části servomechanismu

2.1.1 DC motorek

Hnací sílu pro celý servomechanismus zajišťuje malý stejnosměrný komutátorový motor s permanentními magnety, který by bez použití malé převodovky byl velice slabý.

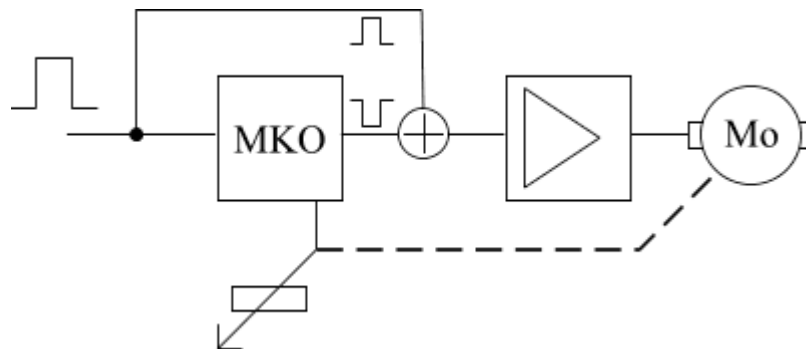
2.1.2 Řídicí elektronika

Řídicí elektronika má za úkol ovládat natáčení hřídele. V podstatě se jedná o monostabilní klopný obvod s potenciometrem pro vytvoření zpětné vazby. Zjednodušené zapojení elektroniky je na blokovém schématu (Obr. 12).

Na vstup přichází řídicí impulz, který spustí monostabilní klopný obvod, ten vygeneruje impulz o délce odpovídající momentální poloze hřídele servomechanismu, ale opačné polarity, než je vstupní řídicí impulz. Tyto dva impulzy se porovnají a výsledkem je rozdílový impulz, který po zesílení přes můstkový spínač způsobí roztočení elektromotoru

jedním nebo druhým směrem. Elektromotor přes převodovku otáčí výstupní hřídeli a současně i potenciometrem, který působí jako zpětná vazba do monostabilního klopného obvodu. Směr otáčení je takový, že impuls generovaný monostabilním klopným obvodem se svojí délkou přibližuje délce vstupního řídicího impulsu a až jsou oba impulsy stejně dlouhé, elektromotor se zastaví. Servo dosáhlo polohy, která odpovídá momentálně přijímanému řídicímu impulsu. [6]

Elektronika servomechanismu je připojena k řídicí desce přes třípinový konektor s roztečí 2,54 mm. Kabeláž je třívodičová s následujícím barevným značením: červená (+ pól), černá (- pól) a žlutá (vstup řídicích impulsů).



Obr. 12 Řídicí elektronika

2.1.3 Převodové ústrojí

Převodové ústrojí servomotoru je tvořeno ze čtyř plastových ozubených koleček, která jsou vhodně uložena v pouzdře. Díky převodovce servomechanismus je schopen ve smyku utáhnout i 3kg při napájecím napětí 6 V.

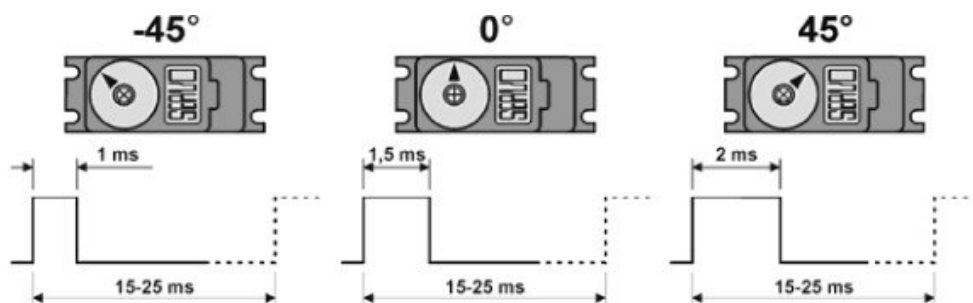
2.2 Řízení serva

Servomotor je řízen za pomoci pulzně - šířkové modulace (PWM). Servo se standardně ovládá 1-2 ms dlouhými pulzy s opakovací frekvencí 50 Hz a s prodlevou 20 ms. Hřídel se po ovládní takto dlouhými pulzy pohybuje v rozmezí 90° úhlových stupňů. Většina servomechanismů však akceptuje pulzy v rozmezí 0,5 – 2,5 ms, což umožňuje natočení hřídele v rozsahu až 180°. Důrazně se doporučuje opatrně odzkoušet, zda je servo

schopné se otáčet v rozsahu 180° , pokud by tomu tak nebylo, může dojít ke zničení elektroniky. Na hřídeli jsou umístěné mechanické dorazy, které vymezují úhel natočení. Bude-li šířka pulzu větší než se může otočit hřídel, dojde ke zvětšení proudu a tím ke zničení elektroniky servomechanismu..

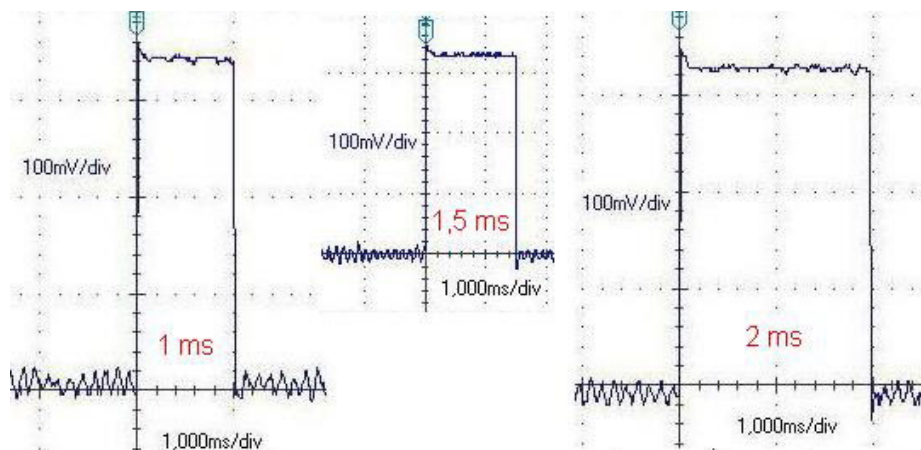
2.2.1 Závislost natočení hřídele na šířce PWM

Zde (Obr. 13) je zobrazena závislost natočení hřídele na délce přivedeného ovládacího signálu. Amplituda pulzu odpovídá velikosti napájecího napětí.



Obr. 13 Závislost natočení hřídele na šířce pulzu [3]

Průběhy zobrazené pod textem (Obr. 14) byli sejmuty z digitálního osciloskopu propojeného s počítačem pomocí softwaru dodávaného k osciloskopu. Jsou na nich zachyceny průběhy odpovídající různým šířkám PWM signálu.



Obr. 14 Průběh PWM na výstupu modulu

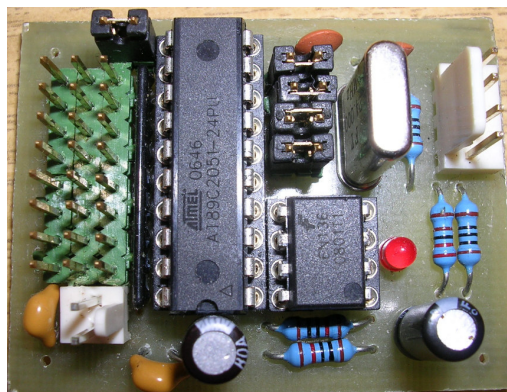
Určitou nevýhodou modelářských servomechanismů je absence zpětnovazební smyčky, která by podávala informace, zda byl daný příkaz proveden. Proto není možné použít tohoto systému pro složitější a náročnější aplikace. Tento nedostatek je ale vykoupen nízkou pořizovací cenou, která je dostupná pro většinu zájemců začínajících s robotikou.

3 ŘÍDICÍ JEDNOTKA

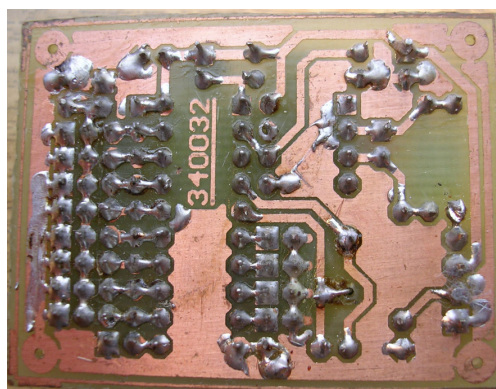
V předchozí kapitole byl popsán způsob řízení servomechanismu, ale dosud nebylo uvedeno jakým způsobem získat potřebný PWM signál. K tomuto účelu slouží řídicí jednotka vyrobená na základě mikrokontroléru od firmy Atmel s typovým označením AT89C2051. Jedná se o často využívaný obvod, který je na trhu již mnoho let a stále se těší velké oblibě.

3.1 Popis jednotky

Jednotka (Obr. 14,15) je připojena k osobnímu počítači přes COM port sériové linky. Vstupní část mikrokontroléru je oddělena od sériového portu počítače dostatečně rychlým optočlenem. Optické oddělení má smysl z důvodu zabránění poškození sériového portu v důsledku zkratu na vedení.



Obr. 15 Řídicí jednotka – pohled ze strany součástek



Obr. 16 Řídicí jednotka – pohled ze strany spojů

Paralelně k výstupu sériového rozhraní je připojena červená LED dioda, která indikuje aktivitu komunikačního kanálu. Při posílání dat bliká v rytmu vysílaných dat. Dioda je indikátorem správné polarizace zapojení vodičů sériové linky, protože musí svítit ihned po připojení konektoru CANON9 do COM portu. Použitý krystal 24 Mhz slouží jako časová základna pro Atmel. K modulu je možné připojit až 8 servomechanismů, které jsou připojeny přes třířadový pinkonektor. Prostřední pin slouží pro + pól napájení elektroniky, pin blíž ke spodnímu okraji DPS přivádí – pól napájení a poslední třetí nejbližší mikrokontroléru slouží pro přívod PWM signálu. Uspořádání přívodních kabelů servomechanismů není jen náhodou takto zapojené. Plusový pól je uprostřed z toho důvodu, kdyby došlo k otočení připojovacího konektoru servomechanismu, tak aby nedošlo k připojení kladného napětí na vstup PWM či záporného pólu a tím ke zničení elektroniky servomechanismu. Napájecí napětí modulu je 4,5 – 6 voltů podobně jako u servomechanismů. Nedoporučuje se používat společný zdroj kvůli rušivým impulzům vznikajících při otáčení servomechanismů. Bude-li použito společného zdroje, je důležité použít filtrační kondenzátor o dostatečné kapacitě. Při použití samostatného zdroje pro napájení servomechanismů lze s výhodou použít pro řídicí modul napájení ze čtyř tužkových baterií, které vydrží v provozu mnoho hodin. [2]

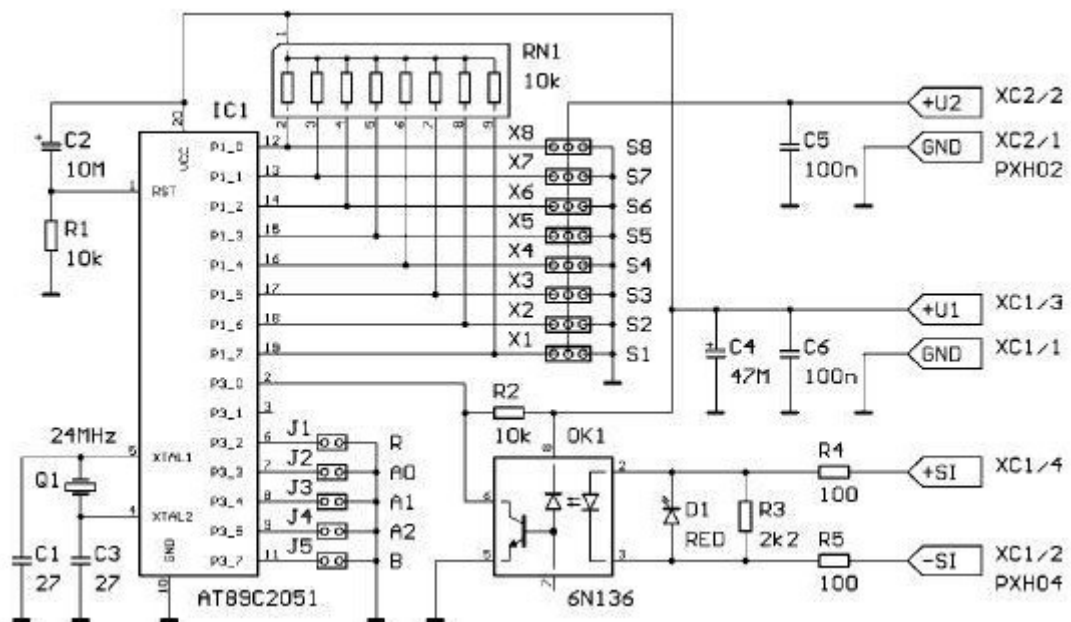
3.1.1 Propojovací konektory

Posledními nepopsanými prvky na desce plošného spoje jsou zkratovací propojky. Je jich použito celkem 5. Každá má svůj význam. První propojka, která je umístěna u připojovacích pinů servomechanismů, slouží pro nastavení komunikační rychlosti sériové linky. Je-li propojka nasazena, komunikační rychlost je 9600 Bd. Při sejmuté spojce systém komunikuje rychlostí 2400 Bd. Snížení rychlosti má oddůvodnění v prostředí se silným rušením. První propojka ve skupině uprostřed desky slouží k nastavení rozsahu PWM signálu. Je-li spojka použita, výstupní rozsah impulzu je od 1 do 2 ms a hřídel servomechanismu se natáčí v rozsahu 90 úhlových stupňů. Je-li spojka sejmuta, výstupní signál PWM je generován již od 0,5 ms až do 2,5 ms. Tímto je možno dosáhnout natočení hřídele v rozpětí až 180 úhlových stupňů. Je potřeba dát si pozor, že každý servomechanismus neumožňuje natáčení v tak velkém rozsahu a mohlo by dojít v horším případě ke zničení elektroniky servomechanismu na základě zvýšeného odebíraného proudu při

snaze překonat mechanický doraz převodového ústrojí. Poslední 3 propojky slouží k nastavení adresy modulu pro komunikaci s počítačem, protože je možné k jedné sériové lince připojit až 8 modulů. Z toho plyne, že systém umožňuje tedy dohromady ovládat až 64 servomechanismů. Standardně jsou všechny tři propojky nastaveny pro komunikaci jednoho modulu.

Aby byla změna nastavení pomocí propojky akceptována, musí být napájecí deska restartována odpojením a znovu připojením od napájecího napětí.

3.2 Schéma zapojení řídicího modulu

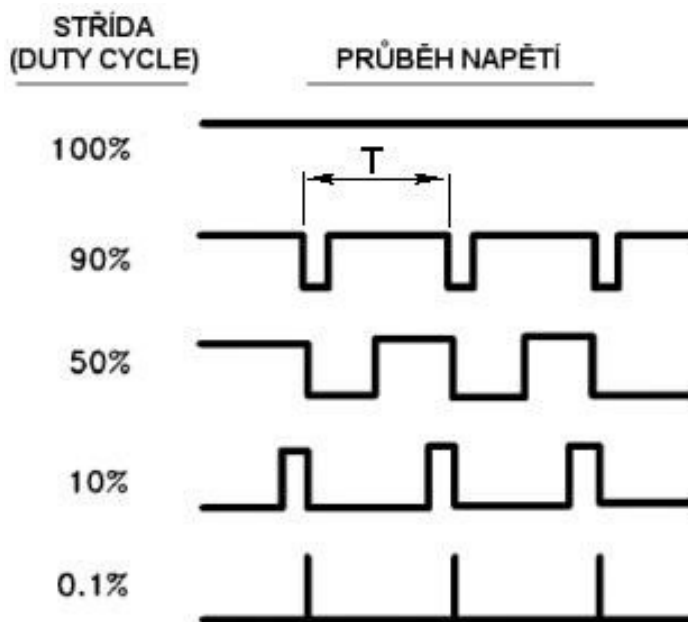


Obr. 17 Schéma zapojení řídicí jednotky [2]

3.3 Generování PWM signálu

Pulzně šířková modulace (PWM) je způsob kódování dat, která se přenášejí z vysílajícího zařízení k přijímacímu zařízení zvolenou přenosovou cestou. Přenosovou cestou může být pevné drátové spojení nebo bezdrátové spojení, kdy se data přenášejí vzduchem např. při IR komunikaci. V kapitole je uvedeno, jak PWM generovat na straně vysílajícího zařízení.

Jde tedy o signál s konstantní periodou T , kde se mění střída napětí. Střída (obr. 18) se uvádí jako poměr (1:1,2:1,1:5 atd.), kdy je nutné uvést, které číslo představuje impuls a které mezeru. Někdy se střída vyjadřuje procentuálně (100%,50%,0.1% atd.), kde 100% představuje ideální poměr 1:0, 50% poměr 1:1 atd.



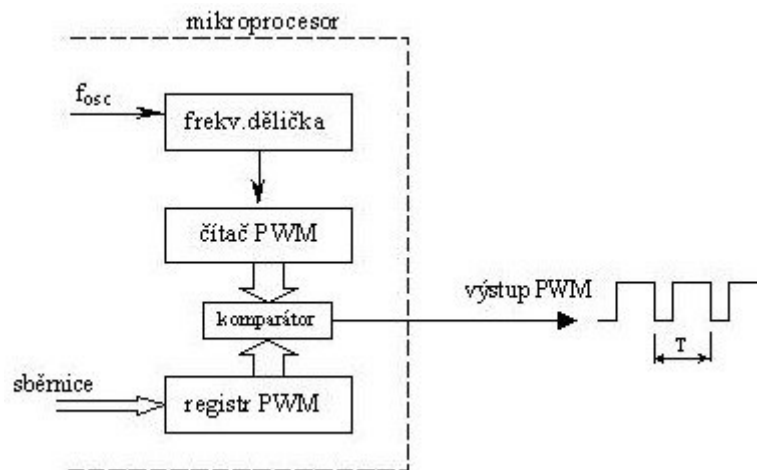
Obr. 18 Střída PWM signálu [2]

3.3.1 Generování pulzně šířkově modulovaného signálu na výstupu mikrokontrolér

V praxi jsou dvě možnosti, jak vytvářet PWM signál. Prvním a jednodušším způsobem je využití mikroprocesoru, který má v sobě integrován obvod pro generování PWM. Druhou možností je vytvářet signál softwarově. V případě Atmelu 89C2051 je zapotřebí generovat PWM softwarově, za pomoci časovače.

Princip generování je schematicky ukázán na následujícím obrázku (Obr. 19). Do frekvenční děličky je přiváděn hodinový signál f_{osc} , který se může ponechat nezměněn nebo se vydělí nějakou dělicí konstantou. Dělicí konstantu lze obvykle zvolit nastavením příslušného řídicího registru. Čítač PWM je volně běžící čítač, který čítá hodinový signál z frekvenční děličky. Obsah tohoto čítače je komparátorem porovnáván s obsahem registru PWM, do kterého programově nastavíme požadovanou hodnotu. Na výstupu komparátoru

dostáváme PWM signál - jsou-li obsahy čítače a registru stejné, je na výstupu log.0; jsou-li různé, je na výstupu log.1. Střída výstupního signálu je tedy úměrná hodnotě zapsané v registru PWM (přibližně podle obr. 18). Perioda výstupního signálu je pevná a je dána čítaným signálem a módem čítače PWM.



Obr. 19 Princip generování PWM

3.4 Komunikační formát mezi osobním počítačem a ovládacím modulem

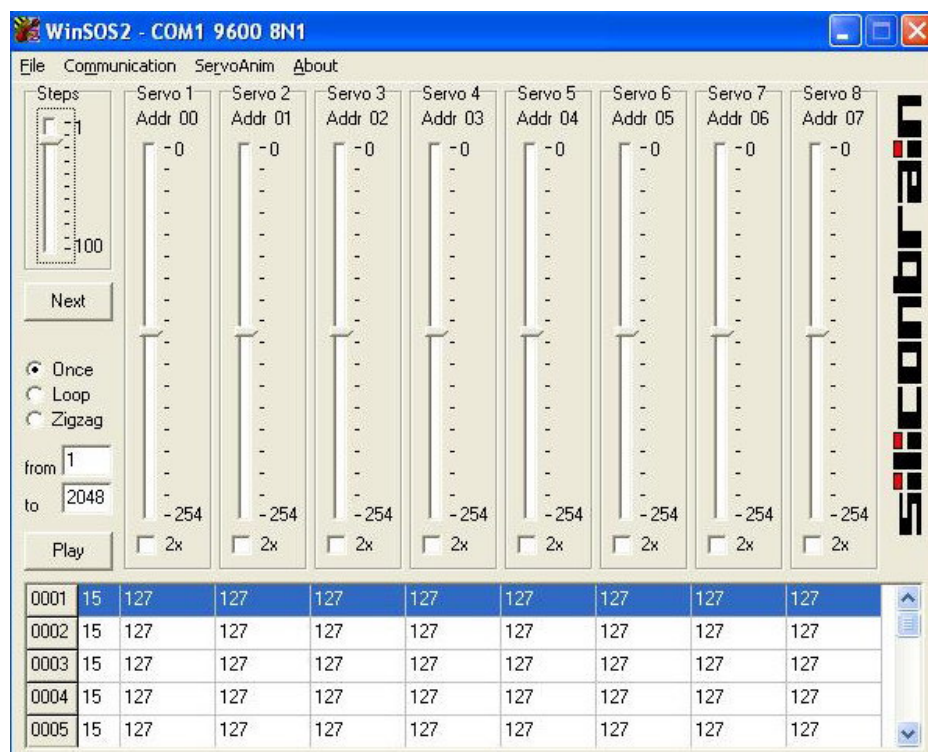
Příkazy posílané z COM portu počítače přes sériovou linku na vstup mikrokontroléru musí mít specifický tvar. Jedná se o 3 bajtové slovo. Ve tvaru <1. byte><2. byte><3. byte>. První bajtové slovo je synchronizační impuls. Ten nabývá vždy hodnoty FFh (255₁₀). Druhý bajt nese číslo označení motoru a velikost je v rozmezí 0-7. Poslední slovo udává pozici, na kterou se má hřídel servomechanismu natočit. Odesílané hodnoty jsou v rozmezí 1 – 254. To znamená, že budeme – li mít servomechanismus, který se natáčí v rozmezí 90 úhlových stupňů, bude tato dráha rozdělená na 254 dílčích kroků.

Konkrétní ukázka: <255><01><149>

Ovládací program musí posílat povely vždy v číselném formátu, nikdy ne v textovém, a pouze tehdy, je – li požadována změna polohy výstupní hřídele některého z ovládaných servomechanismů. Periodické opakování potřebného řídicího signálu PWM zajišťuje řídicí modul automaticky sám. Rychlost otáčení hřídele servomechanismu je dána konstrukcí převodovky a nelze ji nijak ovlivňovat. [2]

3.5 Freeware program pro ovládání servomechanismů

Volně šiřitelný program VŠP (Freeware) program s názvem WINSOS2 (Obr. 20) slouží pro ovládání servomechanismů. Jedná se o jednoduchý a nenáročný program na obsluhu. Tento program byl použit při seznamování s principem ovládání servomechanismů. Později byl vytvořen vlastní ovládací software, který je popsán v praktické části diplomové práce.



Obr. 20 Ovládací program WINSOS2

Program pracuje ve dvou základních režimech. První způsob běhu programu spočívá v ovládání servomechanismů v reálném čase. To znamená, že hřídele servomechanismů reagují na pohyb jezdce a nastavují se do žádané pozice. Druhý režim spočívá v Teach-In programování. Tento způsob programování byl již popsán v úvodní kapitole. Jedná se tedy o „naučení“ jednotlivých kroků robota, a následném spuštění uložených kroků.

3.5.1 Programování metodou Teach-in

1. Vybrat a posunout posuvníkem serva o jeden dílek (hřídel se natočí)

2. Stisknout tlačítko Next (provedený krok se uloží)
3. Pokračovat stejným způsobem až bude docíleno výsledné pohybové sekvence.
4. Do kolonky From zapsat hodnotu 1, do okénka To zapsat hodnotu posledního hodnotu programovaného kroku
5. Spuštění hotové sekvence pomocí tlačítka Play

Před spuštěním programu je možnost nastavit, zda se má program vykonat jednou (Once), nebo se má neustále opakovat (LOOP) anebo zda má být spuštěn od konce (ZigZag)

3.6 Free serial port monitor

Program slouží k jednoduchému a přehlednému monitorování komunikace na sériovém portu osobního počítače. Tento program byl použit k zjištění tvaru komunikačního paketu, který odesílá na sériový port WINSOS2. Teoreticky bylo známo, jak má vypadat daný paket, ale i tak se nedařilo jej správně prezentovat, aby se hřídel servomechanismu pootočila. Výsledkem bylo zjištění, že data odesílána na sériový port jsou převáděna do hexa tvaru.

Na následujícím obrázku (Obr. 21) je ukázáno, jak vypadá řídicí paket pro servomechanismus. Na sériový port bylo odesláno tříbajtové slovo, které provede natočení hřídele do žádané pozice. Povel vypadá takto: 255 05 87.



Obr. 21 Free serial port monitor

4 VÝROBA DESKY PLOŠNÉHO SPOJE ZA POMOCÍ FOTOCESTY

Plošný spoj pro řídicí jednotku byl vyroben pomocí fotocesty. Tento způsob výroby je vhodný pro sériovou výrobu, nebo pro případ, že je zapotřebí dosáhnout profesionálního vzhledu.

4.1 Světlocitlivý povrch desky plošného spoje

Nabízejí se dva způsoby, jak získat kuprexitovou desku plošného spoje se světlocitlivým povrchem. První způsob je nanesení nastříkáním světlocitlivého laku na DPS. Nástřík musí být prováděn v bezprašném prostředí, protože každé zrnko okolo sebe vytváří skvrny, které mají za následek špatně nasvícenou matici. Další nevýhodou je 24 hodinové vytvrzení laku. Druhý a jednodušší způsob je zakoupení desky s již naneseným lakem. Tu stačí ořezat na žádaný rozměr, odlepit černou folii a provést nasvícení předlohy.

Nasvícení předlohy je možné provést nejlépe UV lampou, ale není-li k dispozici, lze využít slunce anebo z vlastní zkušenosti halogenové svítidlo. Osvit se provádí kolmo na předlohu, která je přitisknuta na DPS pomocí skleněné tabulky ve vzdálenosti 30 cm od zdroje světla. Délka osvitu je závislá na zvoleném druhu svítidla. Pro osvit halogenovou lampou 350W postačuje cca 10 minut.

Dalším krokem je vyvolání obrazce na osvícené desce plošného spoje. Vývojka je na bázi zásaditého roztoku. Ponořením a mírným pohupováním desky dojde k odplavení osvíceného světlocitlivého laku. Poté se deska musí řádně opláchnout pod proudem vody. Poslední fází je vyleptání DPS. To je provedeno v leptacím roztoku FeCl_3 .

5 PROGRAMOVACÍ PROSTŘEDÍ SHARPDEVELOP 3.0

SharpDevelop je open source integrované vývojové prostředí, které je volně šiřitelné pod licencí GPL (licence pro svobodný SW) a slouží k vývoji aplikací v programovacích jazycích C#, Visual Basic.NET a Boo na platformě .NET nebo MONO. Usnadňuje psaní, ladění a překlad kódu do výsledných sestavení. Pro kompletaci kódu používá SharpDevelop vlastní C# a VB.NET parser (syntaktická analýza). SharpDevelop 1.1 umí převést projekty z Visual Studia do SharpDevelop projektu, ale až s verzí SharpDevelop 2.0 bylo možné spravovat tyto projekty přímo.

Programovací prostředí Sharp develop bylo použito pro vytvoření ovládacího programu ROB3 pro robotické rameno. Hlavní důvod pro volbu tohoto programu byl dobře pochopitelný způsob tvorby grafické části programu, a to je pro studenta, začínajícího s programováním, podstatné. V této kapitole je popsána základní práce s tímto programem. Dalším důvodem, proč použít Sharp develop, je jeho volná dostupnost na internetu a co se týče instalačního balíčku, ten má pouze 18 Mb.

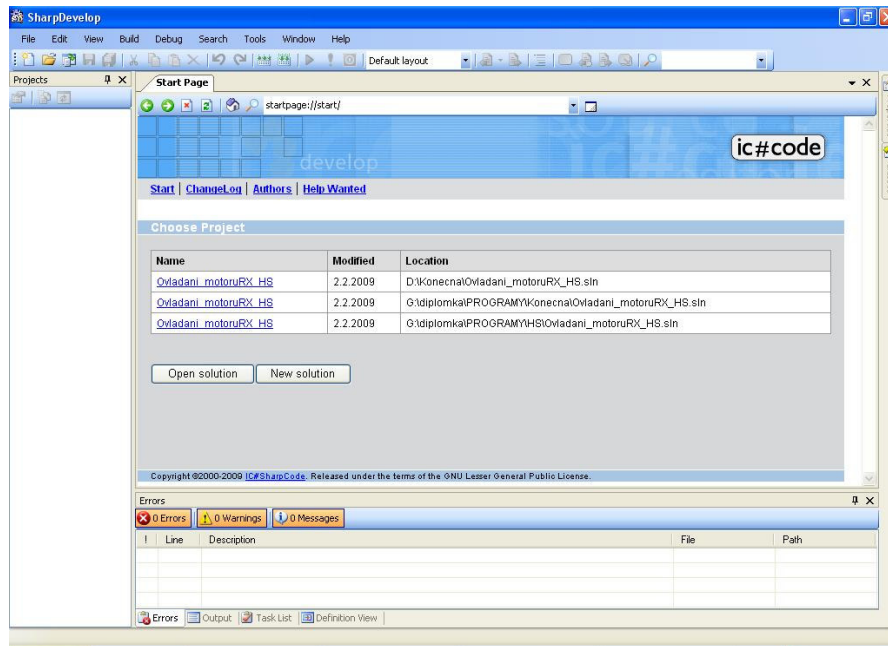
5.1 Hlavní okno programu

Po spuštění programu je zobrazena tabulka (Obr. 22) s vytvořenými projekty. Obsluha má možnost buď otevřít stávající projekt, anebo založit zcela nový.

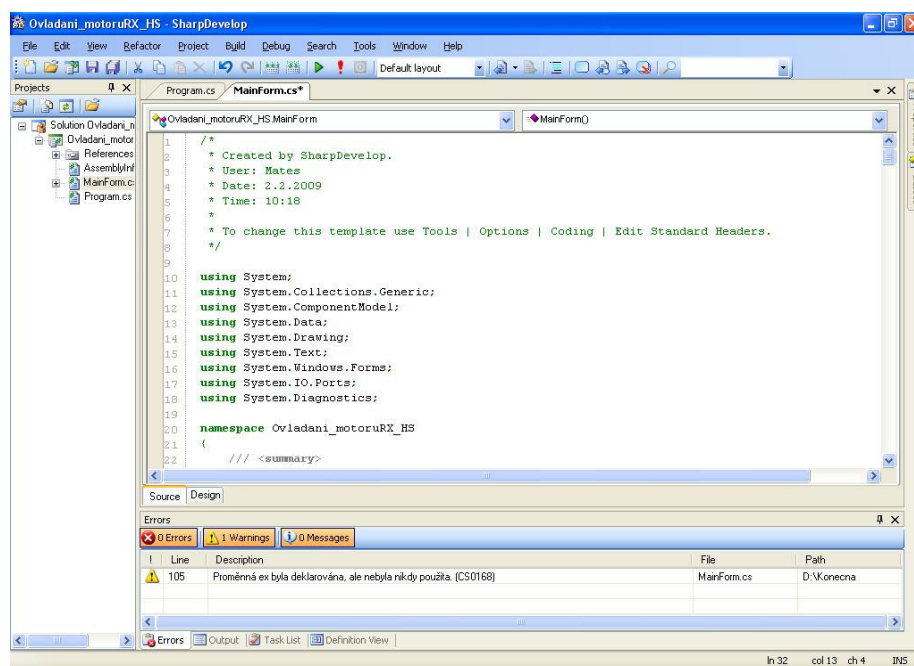
Po načtení projektu je zobrazen v okně tzv. Mainform, ve kterém je zapsán celý zdrojový kód programu. Program je rozdělen do několika logických částí. První důležitou částí je nadefinování knihoven, které budou použity. Potom následuje definice proměnných, v zápětí navazuje tělo programu, které tvoří program. Tělo programu je ohraničeno na začátku a na konci složenými závorkami.

Obrázek 23 ukazuje, jak vypadá programovací prostředí. V levé spodní části okna programu jsou umístěny dvě záložky s názvy Source a Design. Záložky přepínají mezi zdrojovým kódem a grafickým rozhraním návrhu programu. Na obrázku (Obr. 24) je vidět, jak vypadá grafický návrh programu. Pro vytváření grafického okna je použito nástrojů Windows Form. Windows Form nabízejí programátorovi použít předem nadefinované objekty v grafické podobě. Příkladem mohou být tlačítka či různé posuvníky, jak je vidět v levém sloupci na obrázku (Obr. 24). Ty lze potom libovolně umístit do vytvářeného programu. V dalším kroku je nutné přiřadit každému objektu jeho funkci. To se děje

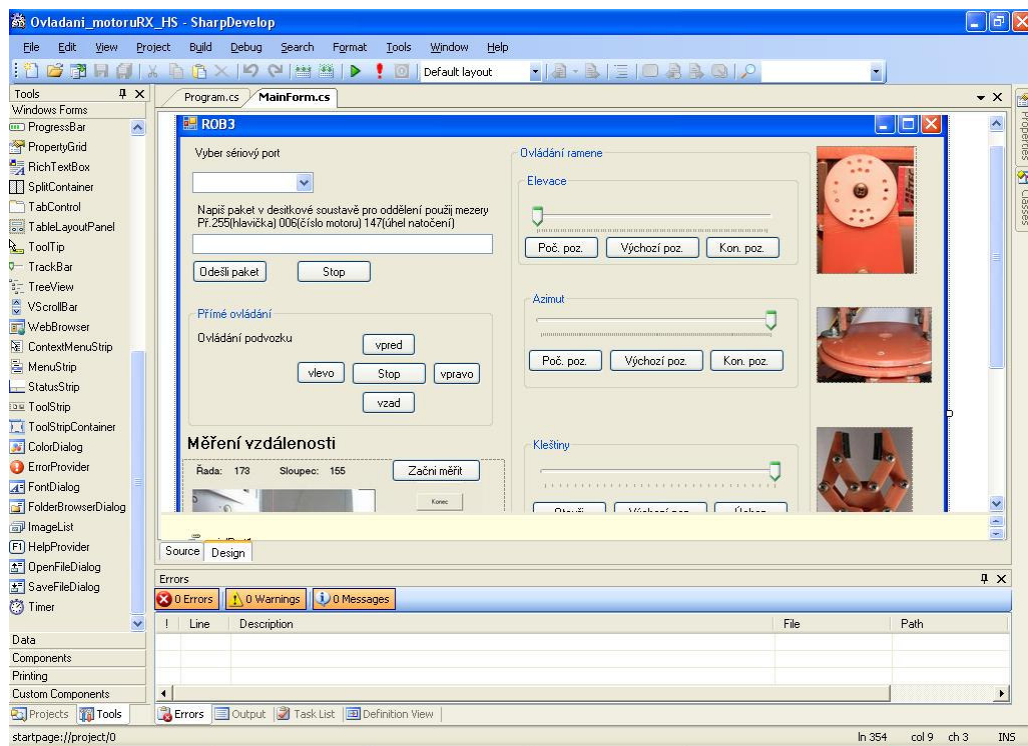
označením objektu dvojklikem. Následně je program přepnut automaticky do režimu Source a programátor vyplní příslušné údaje. Tato operace musí být provedena pro každý objekt. Příklady definic příkazů jsou popsány v praktické části této práce.



Obr. 22 Úvodní obrazovka Sharp Develop



Obr. 23. Hlavní okno programovacího prostředí



Obr. 24 Main Forms

6 MĚŘENÍ VZDÁLENOSTI

6.1 Měření vzdálenosti za pomoci laseru a webkamery

Existuje více možností, jak měřit vzdálenost pomocí ultrazvukových, kapacitních, indukčních a jiných snímačů. Uvedená zařízení pracují při splnění určitých podmínek. Pro použití v protiteroristické robotice se používá měření pomocí triangulačních aktivních, popřípadě pasivních optoelektronických snímačů.

V této kapitole je uveden aktivní způsob měření vzdálenosti použitím laserového zářiče (zde levného laserového ukazovátka) a webkamery. Přesnost systému je závislá na více faktorech, zmíněných na konci kapitoly.

6.1.1 Princip činnosti

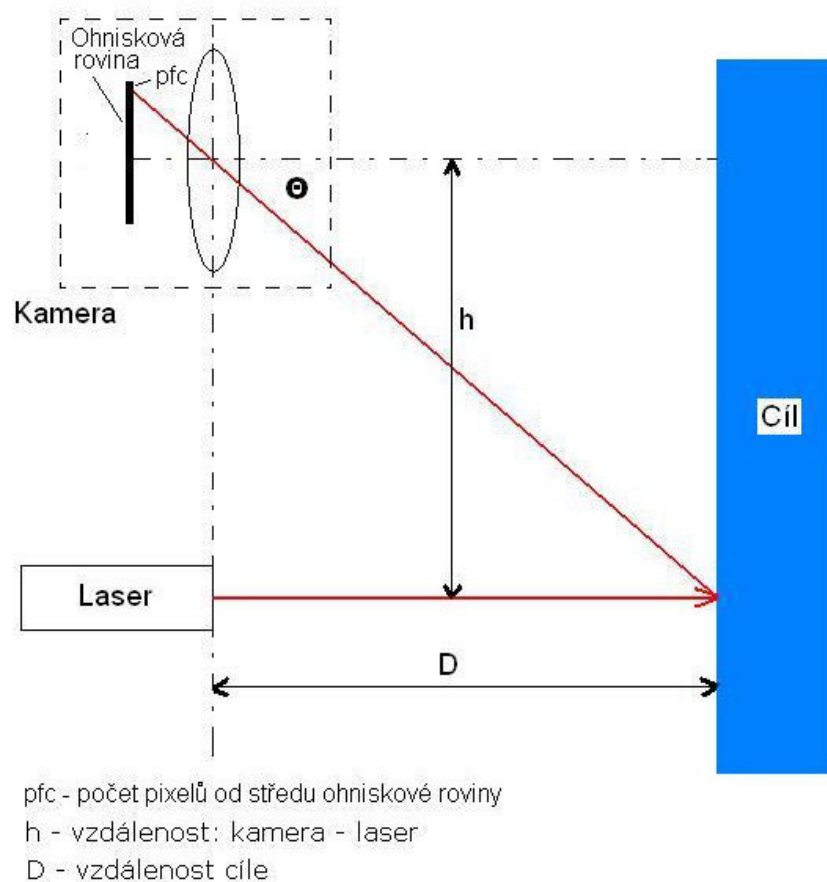
Na obrázku (Obr. 25) je schématické znázornění principu funkce triangulačního laserového zaměřování. Skládá se z laseru a webkamery. Kamera snímá polohu stopy laserového paprsku v jejím zorném poli na obrazovém snímáči a pomocí dalších známých parametrů jako vzdálenost laseru od kamery, ohniskové vzdálenosti, velikosti obrazového elementu a z polohy laserové stopy na obrazovém snímáči se vypočítá vzdálenost předmětu. [4]

Paprsek je promítán na objekt v zorném úhlu kamery. Vysílaný laserový paprsek je rovnoběžný s optickou osou kamery. Světelná stopa laseru je snímána kamerou spolu s ostatními částmi scény. Pozice laserového bodu v obrazovém snímáči je známá. Čím blíže bude laserový bod zachycený na obrazovém snímáči k centru ohniskové roviny, tím bude objekt vzdálenější. Podle obrázku (Obr. 25) se vypočítá vzdálenost D podle vzorce: [4]

$$D = \frac{h}{\tan \Theta} \quad (1)$$

Úhel Θ vypočítáme podle vzorce:

$$\Theta = pfc \cdot rpc \quad (2)$$



Obr. 25 Princip laserového zaměřování

pfc – počet obrazových elementů od středu ohniskové roviny

rpc – délka obrazového elementu

Spojením těchto dvou rovnic získáme výsledný vztah pro určení vzdálenosti objektu:

$$D = \frac{h}{\tan(pfc \cdot rpc)} \quad (3)$$

Přesnost použité metody závisí na dalších faktorech. Relativně přesného měření lze dosáhnout s web kamerou do vzdálenosti 1,5m. Prvním ovlivňujícím faktorem je povrchová úprava měřeného předmětu. Lesklý povrch paprsek odráží a vytváří nepravidelný bod, naopak zase černé materiály pohlcují paprsek. Negativní dopad na měření má příliš vysoká intenzita okolního osvětlení, ideální pro měření vzdálenosti je tmavá místnost.

II. PRAKTICKÁ ČÁST

7 REALIZACE ROBOTICKÉHO RAMENE

Prvotní verze robotického manipulátoru byla vyrobena na základě podkladů z časopisu zabývajícího se elektronikou „Amatérské rádio“. Kamerový 2-D CCD systém s WIFI přenosem obrazové informace je umístěn na robotickém rameni na monitorování okolí. Robotické rameno s podvozkem je ovládáno komerčním softwarem WINSOS2. Pro nastavování polohy robotického ramene tento program vyhovoval, ale pro řízení podvozku byl vytvořen nový program. Součástí tohoto programu se staly ovládací prvky servomechanismů robotu a zaměřovací systém.

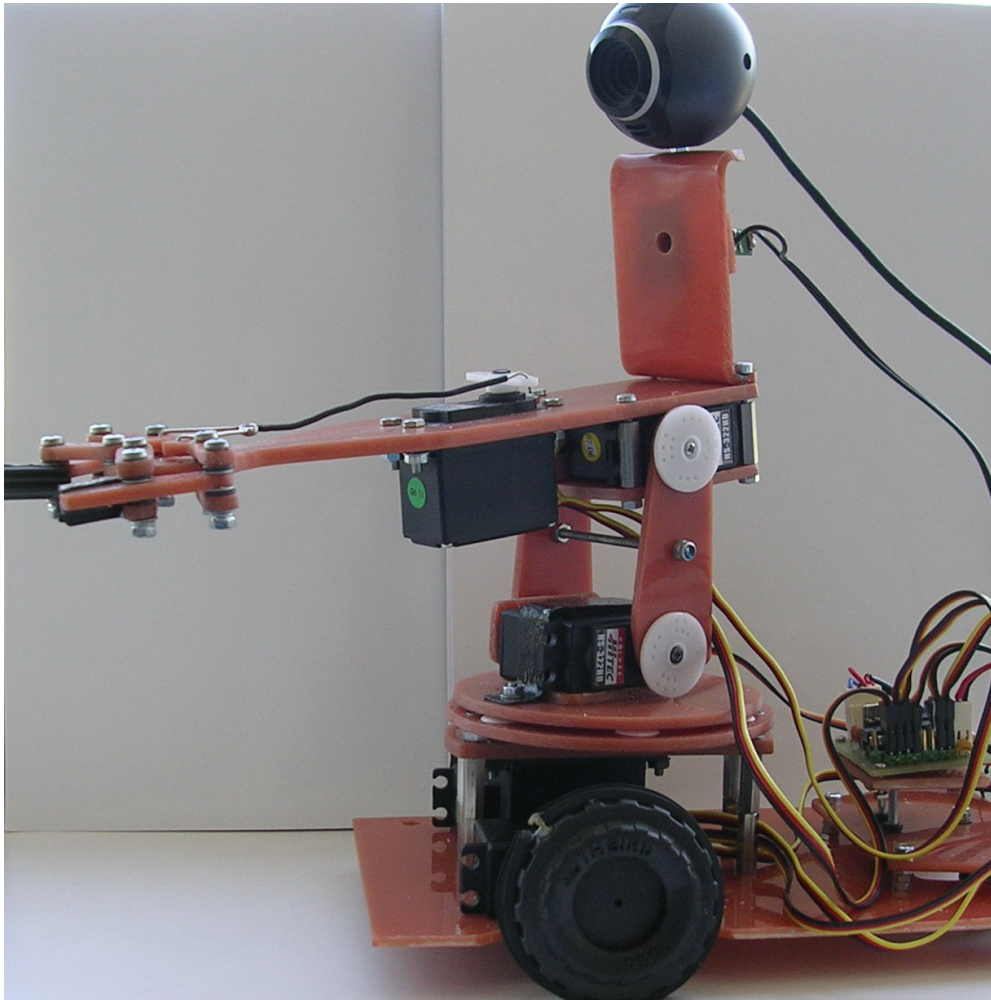
Úprava byla na robotu provedena přidáním dalšího stupně volnosti se servomechanismem, který naklání robotické rameno před sebe, čímž se zvětšil manipulační dosah. Poslední etapou byla montáž zaměřovacího systému, pracujícího na principu snímání laserového bodu kamerovým systémem.

7.1 Konstrukční provedení

Konstrukční díly robotického ramene (Obr. 26) jsou vyřezány z novodurové desky (tvrzeného polyvinylchloridu) o tloušťce 3mm lupénkovou pilkou. Pro zjednodušení práce byla předloha konstrukčních prvků vytištěna v měřítku 1:1 na samolepící papír, který byl nalepen na desku. Vzhledem k tomuto postupu byla dosažena přesnost vyřezaných dílů a vrtaných otvorů. Každá hrana je pečlivě opracována pilníkem a smirkovým papírem. Poté byly vyvrtány díry podle předlohy pro šroubky s metrickým závitem M3, za pomoci kterých byla provedena montáž ramene. Do příslušných výřezů byly umístěny servomechanismy. Dalšími montážními díly byly osmihranné distanční sloupky a šrouby s hlavičkou na imbusový klíč. Vyřezání a složení samotného ramene bez podvozku a zaměřování trvalo při pečlivé práci asi 8 hodin čistého času.

Podstatným konstrukčním prvkem celého ramene jsou upínací kleštiny. Upínací kleština (Obr. 27) je řešena jako paralelogram, skládající se ze šesti shodných ramen jednozvratných a dvou ramen dvojzvratných. Tahem a tlakem serva za dvojzvratná ramena se kleština rozevívá a svírá. Klouby ramen kleštiny jsou vytvořeny provlečením šroubů M3, u zadních ramen jsou na šrouby navlečeny izolační podložky, které slouží jako distanční podložky a umožňují lehké zasunutí táhla od ovládacího servomechanismu. Táhlo serva je

sletováno z kancelářské sponky, popřípadě z tenkého ocelového drátku a potaženo teplem smršťitelnou bužírkou.

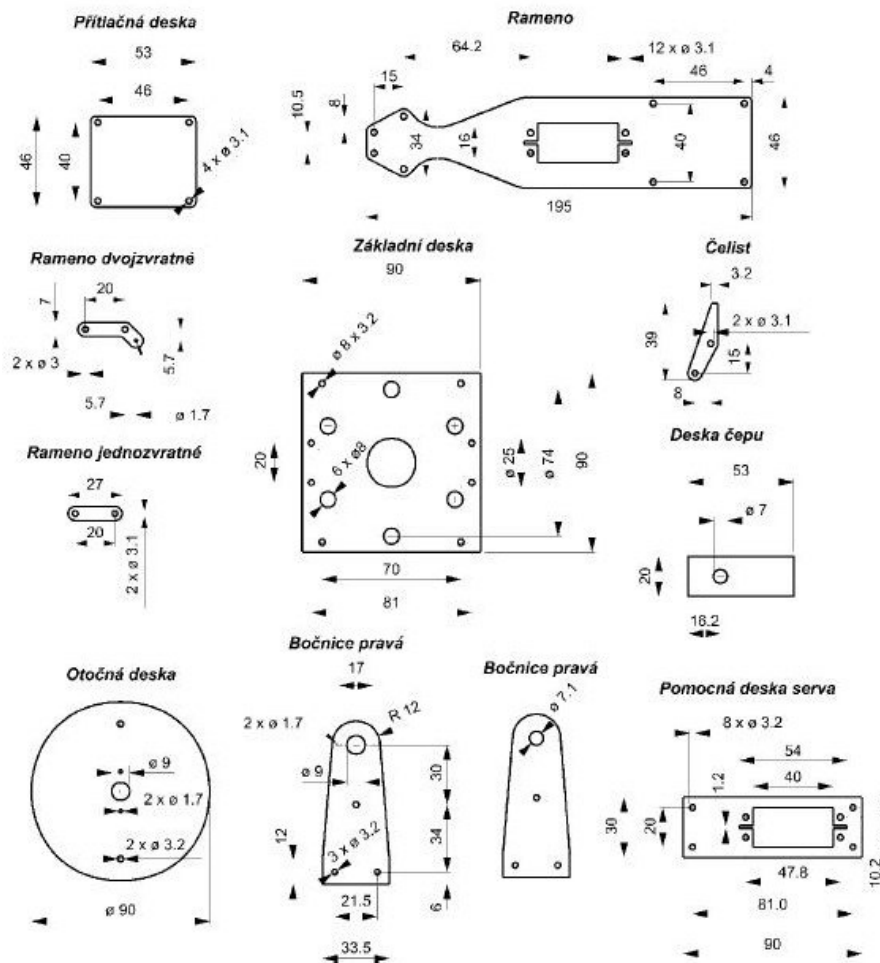


Obr. 26 Robotické rameno



Obr. 27 Konstrukce kleštiny

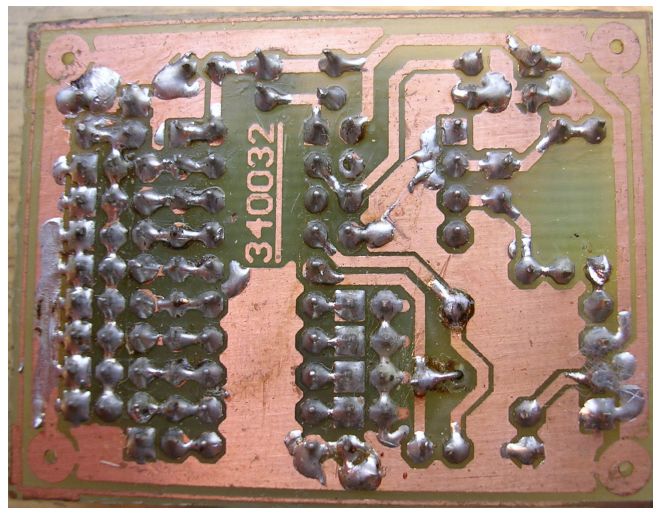
Pro představu, z kolika dílů je robot vyroben, je přiložen rozpis s konstrukčními díly i se zakótováním rozměrů (Obr. 28) .



Obr. 28 Konstrukční díly [2]

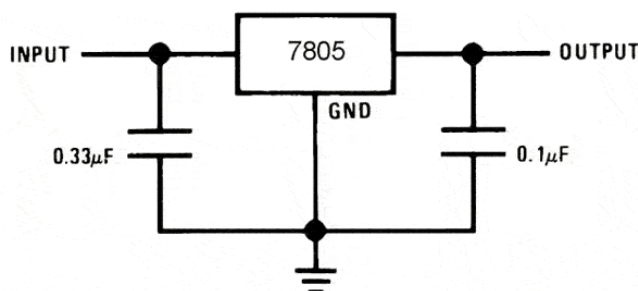
7.2 Realizace řídicího modulu

Řídicí modul je osazen mikrokontrolérem od firmy Atmel s typovým označením AT89C2051. Mikrokontrolér je naprogramován pro řízení servomechanismu pulzně šířkovou modulací. Plošný spoj (Obr. 29) byl vyroben pomocí fotocesty. Hotový plošný spoj byl natřen ochranným lakem, který navíc zlepšuje pájitelnost. Mikrokontrolér je osazen do 20ti pinové patice pro snadnější vložení mikroprocesoru a také, aby bylo zabráněno případnému poškození Atmelu v důsledku pájení.



Obr. 29 Deska plošného spoje

Aby byl modul robotu plně mobilní, je účelné použít bateriového napájení. Pro napájení elektroniky modulu plně dostačuje použít 4 nabíjecí baterie typu AA (tužkové baterie). Protože má jednotka odběr řádově jednotky miliampér, vydrží akumulátory desítky hodin v provozu. Pro napájení servomechanismu je zapotřebí použít zdroj o větší kapacitě, protože servomechanismus při plném zatížení může mít odběr i 400 mA. Zde byl použit akumulátor ze zabezpečovací ústředny o napětí 12 V. Protože servomechanismus je dimenzován na max 6 V, byl použit stabilizátor „srážející“ napětí na 5 V. Zapojení integrovaného stabilizátoru je na obr. 30. Obvod vyrábí firma Tesla pod typovým označením MA 7805. Součástky jsou vyráběny i dalšími firmami jako například LM pod stejným označením 7805.



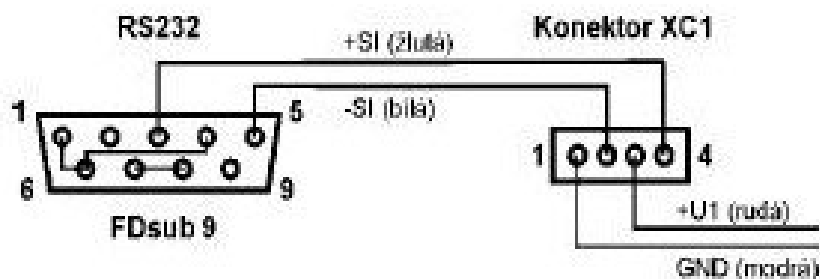
Obr. 30 Zapojení integrovaného stabilizátoru

7.2.1 Zapojení komunikačního kabelu

Komunikace mezi řídicí jednotkou a počítačem je realizována přes sériový port RS232. Tento port byl osazován na každém osobním počítači a notebooku jako standard. V současnosti se od montáže tohoto portu upouští, proto jedinou možností je použít drahý převodník USB na RS232, ne vždy funkční pro danou aplikaci.

Pro spolehlivou funkci je doporučeno použít kvalitní stíněný kabel, který zajistí spolehlivou komunikaci. Technická specifikace sériového rozhraní uvádí jako doporučení použití maximálně 15 metrů dlouhého kabelu. Při splnění určitých požadavků je možné kabel prodloužit.

Schéma zapojení propojovacího vedení je uvedeno na obr. 31. Na straně od PC je použit 9-ti pinový konektor CANON a ze strany od řídicí jednotky 4 pinový konektor se zámkem s obchodním označením PXH04. Součástí konektoru je i napájení jednotky.



Obr. 31 Zapojení komunikačního kabelu [2]

Pro komunikaci stačí pouze jeden datový vodič a zem, protože data nejsou již posílána zpětně z řídicí jednotky do počítače.

Robot by se stal plně mobilním, kdyby přenos byl proveden bezdrátově. Tato alternativa byla zvažována, ale pro nedostatek času k realizaci nedošlo.

Při ožívání systému bezprostředně nenastal žádný výrazný problém. Jeden problém byl řešen až po určité době. Robot byl použit s cílem vyzkoušet nový ovládací program. Přestože program odesílal na port data ve správném tvaru, hřídel servomechanismu se nechtěla pohnout. Nebylo možné přijít na závadu, každá část byla proměřena osciloskopem, vše bylo v pořádku. Napájecí zdroj jednotky a servomechanismů ukazoval požadované hodnoty. Po dlouhé době bylo zjištěno, že i přestože na výstupu stabilizátoru pro napájení servomechanismů je žádaná hodnota napětí, akumulátor nebyl dobíjen a není schopen pokrýt proudové zatížení od servomechanismů.

8 PROGRAMOVÉ VYBAVENÍ

Prvním programem pro ovládání robota byl program WINSOS2. Na základě tohoto programu bylo možné pochopit princip ovládání a inspirovat se pro vývoj vlastního programového vybavení.

8.1 PROGRAM ROB3

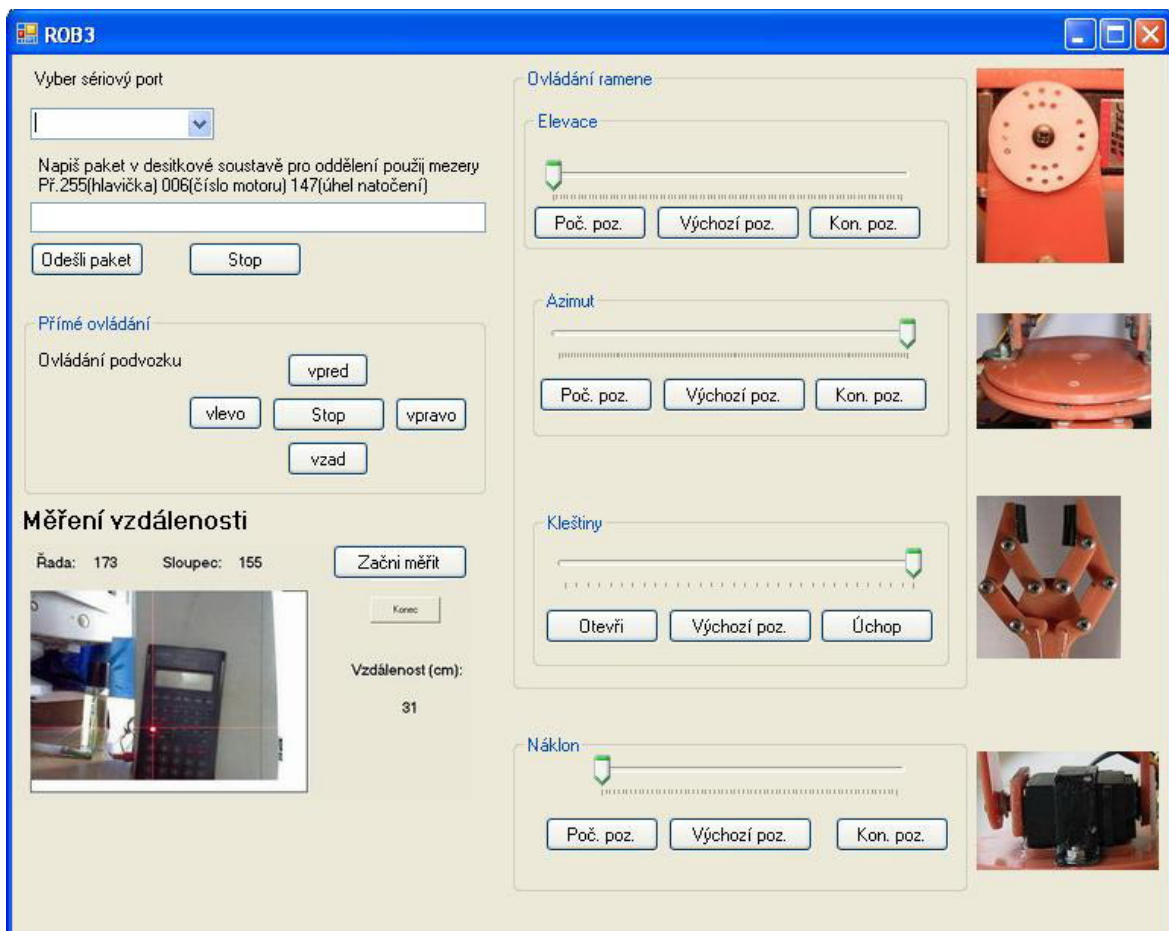
Nový program byl nazván ROB3. Pracovní plocha programu je rozčleněna do několika částí, každá z nich slouží k obsluze robota. Při návrhu byl kladen důraz na jednoduchost a přehlednost obsluhy systému.

Funkce programu ROB3 (Obr. 32) je rozčleněna na 5 částí. První částí je dialogové okénko s výběrem COM portu. Při spouštění programu jsou z operačního systému detekovány přítomné COM porty počítače. Obvykle jsou porty v rozmezí COM1 až COM3. Je-li tedy zvolen správný port, je možné přistoupit k ovládání robota.

Další položkou v programu je příkazový řádek, který je určen pro vkládání tříbajtových ovládacích paketů. Jak má být prezentován příslušný paket bylo popsáno již v první části této práce. Uživatel si tak může vyzkoušet základní princip ovládání servomechanismů. Na sériový port je možné odeslat celou sérii řídicích povelů najednou. Nevýhodou takového řízení je „otrocké“ vypisování příkazů.

Podvozek se může pohybovat konstantně nastavenou rychlostí vpřed, vzad, doleva a doprava. Natáčení je realizováno protipohybem.

Největší část plochy programu zabírá část pro ovládání jednotlivých servomechanismů pro jednotlivé stupně volnosti. Každému servomechanismu odpovídá jeden posuvný jezdec, který je nastaven v takovém rozmezí, aby nedošlo k přetočení do nežádoucích pozic a k mechanickému poškození konstrukčních dílů či spálení jednotky serva. Tři tlačítka umístěná pod drahou posuvného jezce slouží pro snadnější manipulaci. Tlačítka Počáteční pozice a Koncová pozice mají za úkol provést natočení hřídele do horní a dolní úvrati. Prostřední tlačítko Výchozí pozice zajišťuje natočení do poslední pozice, která byla nastavena před stlačením tlačítek Počáteční a koncová pozice. Použitím tohoto tlačítka má být zabráněno nepříjemnému trhnutí ramene, čímž se předejde poškození robotického manipulátoru.



Obr. 32 Ukázka programu ROB3

Poslední nepopsanou částí programu je levá spodní část okna programu ROB3. Zde je umístěn systém pro měření vzdálenosti objektu od robotického ramene. Po stlačení butonu „Začni měřit“ se spustí externí okno, ve kterém probíhá měření. Zdrojový kód je popsán v další kapitole této práce.

8.2 Popis kódu programu ROB3

Program byl vytvořen v programovacím jazyku C # ve vývojovém prostředí Sharp Develop3. Stručný popis programovacího prostředí byl již uveden v předešlé kapitole.

```
/*
 * Created by SharpDevelop.
 * User: Martin Horak
 * Date: 4.3.2009
 * Time: 13:05
 */
```



```
* V první části programu jsou načteny knihovny a definovány komponenty
zajišťující chod celého programu
*/

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;
using System.Diagnostics;

//název projektu
namespace Ovladani_motoru_HS
{
    /// charakteristika hlavní části programu
    public partial class MainForm : Form
    {
        string InputData = String.Empty;
        delegate void SetTextCallback(string text);
        public MainForm()
        {
            // inicializace komponent volaných z Windows Form (Design část)
            InitializeComponent();
            string[] ports = SerialPort.GetPortNames();

            //přiřazení názvu portu
            foreach (string port in ports)
            {
                comboBox1.Items.Add(port);
            }

            // Volání a nastavení sériového portu
            void ComboBox1SelectedIndexChanged(object sender, EventArgs)
            {
                if (serialPort1.IsOpen) serialPort1.Close();
                serialPort1.PortName = comboBox1.SelectedItem.ToString();
            }

            //zkouška otevření sériového portu
            try
            {
                serialPort1.Open();
            }
            // zpráva o stavu sériového portu
            catch
            {
                MessageBox.Show("Serial port " + serialPort1.PortName + " Port nelze
otevrit!", "RS232 tester", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                comboBox1.SelectedText = "";
            }
        }
    }
}
```

```

//odesílání paketu z příkazového řádku
void Button1Click(object sender, EventArgs e)
{
    /* retezec v textBox1 (příkazovém řádku) musí být ve
    tvaru "ff ff a1 08"
    metoda Split() rozdeli zadnou
    hodnotu podle znaku ' ' do pole*/

    //vlození hodnot z textBoxu (příkazového řádku)
    String[] numbers = textBox1.Text.Split();
    // pole, do kterého se uloží výsledný příkaz
    byte[] cmd = new byte[numbers.Length];
    //definování proměnné respektive pole

try
{
    // kazde hexa cislo je převedeno do desítkového tvaru
    for (int i = 0; i < numbers.Length; i++)
        cmd[i] = byte.Parse(numbers[i],
            System.Globalization.NumberStyles.Integer);
    // metoda Parse detekuje číslo, které není zadáno v hexa tvaru
}
catch (FormatException ex)
{
    MessageBox.Show("Špatně vložený znak", "",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}
// tato část převede obsah příkazového řádku na řetězec
String txt = "";
foreach (byte b in cmd)
{
    txt += b + " ";
}

// zobrazení odesílaných hodnot na port
MessageBox.Show(txt, "Odesílané hodnoty", MessageBoxButtons.OK,
    MessageBoxIcon.Information);

// Zápis hodnot na sériový port
if (serialPort1.IsOpen)
{
    serialPort1.Write(cmd, 0, cmd.Length);
}
else
{
    MessageBox.Show("Sériový port uzavřen!" MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}

textBox1.Clear();

//-----OVLADANI PODVOZKU-----
//jízda dozadu
void Button2Click(object sender, EventArgs e)
{
    /*Na sériový port je odeslána hodnota
    Odpovídající příslušnému směru pohybu*/
}

```

```
        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)
    {
        /* na sériový port je odeslána hodnota
        Která odpovídá zastavení otáčení hřídele*/
        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 (příkazový řádek)
    void Button6Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x06, 0x7F}, 0, 3);
        serialPort1.Write(new byte [] {0xff, 0x07, 0x7F}, 0, 3);
    }

    //-----

    //-----OVLADANI RAMENE-----

    //Jezdec Elevace
    void TrackBar1Scroll(object sender, EventArgs e)
    {
        /*Hodnota z aktuální pozice jezdce je uložena
        do proměnné c. hodnota převedena do hexa
        a odeslána na sériový port*/
        serialPort1.Write(new byte [] {0xff, 0x00}, 0, 2);
        byte[] c = null;
        c = BitConverter.GetBytes(trackBar1.Value);
        serialPort1.Write(c, 0, c.Length);
    }

    //Tlacitko "Pocatecni pozice"
    void Button7Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x00, 0x44}, 0, 3);
    }

    //Tlacitko "Koncová pozice"
    void Button8Click(object sender, EventArgs e)
    {

```

```
        serialPort1.Write(new byte [] {0xff, 0x00, 0xa9}, 0, 3);
    }
    //Tlacitko "Vychozí pozice"
    void Button9Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x00}, 0, 2);
        byte[] c = null;
        c = BitConverter.GetBytes(trackBar1.Value);
        serialPort1.Write(c, 0, c.Length);
    }
    //Popis jezdce "Elevace"
    void Label4Click(object sender, EventArgs e)
    {
    }
    //Vlozeni obrázku
    void PictureBox3Click(object sender, EventArgs e)
    {
    }

    //-----
    //Jezdec azimut
    void TrackBar2Scroll(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x01}, 0, 2);
        byte[] b = null;
        b = BitConverter.GetBytes(trackBar2.Value);
        serialPort1.Write(b, 0, b.Length);
    }
    //Jezdec "Pocateční pozice"
    void Button10Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x01, 0x46}, 0, 3);
    }
    //Tlacitko "Koncová pozice"
    void Button12Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x01, 0xba}, 0, 3);
    }
    //Tlacitko "Výchozí pozice"
    void Button11Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x01}, 0, 2);
        byte[] c = null;
        c = BitConverter.GetBytes(trackBar2.Value);
        serialPort1.Write(c, 0, c.Length);
    }
    //Popis jezdce "Azimut"
    void Label6Click(object sender, EventArgs e)
    {
    }
    //Vlozeni obrázku
    void PictureBox2Click(object sender, EventArgs e)
    {
    }

    //-----

    //Jezdec Kleštiny
    void TrackBar3Scroll(object sender, EventArgs e)
    {
```

```
        serialPort1.Write(new byte [] {0xff, 0x02}, 0, 2);
        byte[] b = null;
        b = BitConverter.GetBytes(trackBar3.Value);
        serialPort1.Write(b, 0, b.Length);
    }
    //Tlacitko "Otevřeni kleštin"
    void Button13Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x02, 0x70}, 0, 3);
    }
    //Tlacitko "Zavřeni kleštin"
    void Button15Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x02, 0x51}, 0, 3);
    }
    //Tlacitko "Výchozí pozice"
    void Button14Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x02}, 0, 2);
        byte[] c = null;
        c = BitConverter.GetBytes(trackBar3.Value);
        serialPort1.Write(c, 0, c.Length);
    }
    //Popis jezdce "Kleštiny"
    void Label5Click(object sender, EventArgs e)
    {
    }
    //Vložení obrázku
    void PictureBox1Click(object sender, EventArgs e)
    {
    }
    //Panel sdružující skupinu ovladacích prvku ramene
    void GroupBox2Enter(object sender, EventArgs e)
    {
    }
    //Panel sdružující skupinu ovladacích prvku podvozku
    void GroupBox1Enter(object sender, EventArgs e)
    {
    }

    //Jezdec náklonu
    void TrackBar4Scroll(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x03}, 0, 2);
        byte[] b = null;
        b = BitConverter.GetBytes(trackBar4.Value);
        serialPort1.Write(b, 0, b.Length);
    }
    //Tlačitko "výchozí pozice"
    void Button17Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x03}, 0, 2);
        byte[] c = null;
        c = BitConverter.GetBytes(trackBar4.Value);
        serialPort1.Write(c, 0, c.Length);
    }

    //Tlačitko "počáteční pozice"
    void Button16Click(object sender, EventArgs e)
```

```
        {
            serialPort1.Write(new byte [] {0xff, 0x03, 0x51}, 0, 3);
        }
//Tlačítko "Koncová pozice"
void Button18Click(object sender, EventArgs e)
    {
        serialPort1.Write(new byte [] {0xff, 0x03, 0x98}, 0, 3);
    }
//Vložení obrázku
void PictureBox4Click(object sender, EventArgs e)
    {
    }
//Panel sdružující skupinu ovládacích prvku ramene
void GroupBox4Enter(object sender, EventArgs e)
    {
    }
//Panel sdružující skupinu ovládacích prvku ramene
void GroupBox3Enter(object sender, EventArgs e)
    {
    }
//Panel sdružující skupinu ovládacích prvku ramene
void GroupBox5Enter(object sender, EventArgs e)
    {
    }

//-----Zaměřování-----
void PictureBox5Click(object sender, EventArgs e)
    {
    }
void Button19Click(object sender, EventArgs e)
    {
        ProcessStartInfo startInfo = new ProcessStartInfo();
        startInfo.WorkingDirectory = @"\\Konecna";
        startInfo.FileName = @"\\Konecna\laser.exe";
        Process.Start(startInfo);
    }
//Vložení popisku
void MainFormLoad(object sender, EventArgs e)
    {
    }
}
}
```

8.3 VideoOCX

Jedná se o programové vybavení, které má za úkol integrovat do vlastních programových aplikací zachytávání videa a zpracování obrazu. Ovládání je kompatibilní s mnoha video zařízeními běžně používanými pod systémem Windows. Mohou to být USB kamery nebo zachytávací karty s CCD kamerami či videokamery. Implementace

VideoOCX do programu spočívá v přidání několika programovacích řádek, které nastaví parametry zachytávání. Mezi vývojová prostředí, která jsou podporována, patří především Microsoft Visual Basic, Microsoft Visual C++, Borland Delphi, Microsoft Access a mnoho dalších prostředí podporující technologii ActiveX.

Prvku VideoOCX, bylo použito pro vytvoření laserového zaměřování. Výhoda použití spočívá v celkovém zjednodušení programu, protože není zapotřebí používat externích knihoven. Komponenta provede automatické načtení obrazu a umožní další operace s obrazovou informací, či videem. Jako příklad možné ukázky práce s OCX je zjištění rozlišení obrazového snímku v obrazových elementech, případně převedení barevného obrazu do stupňů šedi.

8.3.1 Ovládací příkazy

V první řadě je důležité stáhnout z internetu utilitu VideoOCX. Program je aktivní 14 dní po instalaci jako zkušební bez registrace. Po vypršení této doby je potřebná registrace a zaplacení registračního poplatku, který není vysoký se zvýhodněním studentů.. V druhém kroku musí být provedena instalace do operačního systému. Při instalaci není nutné provádět jakékoliv nastavení. Je-li instalace provedena úspěšně, již nebrání nic v práci.

Zde je uvedeno několik základních instrukcí používaných pro běžnou práci s VideoOCX v programovacím prostředí Visual Basic.

VideoOcx.Driver – výběr hardwarového zařízení

VideoOCX.Init – načtení ovladače videozařízení

VideoOcx.GetColorImageHandle – uvolnění paměťového prostoru pro obrazové data

VideoOcx.Start – spuštění zachytávání obrazu

VideoOcx.Stop – přerušení zachytávání obrazu

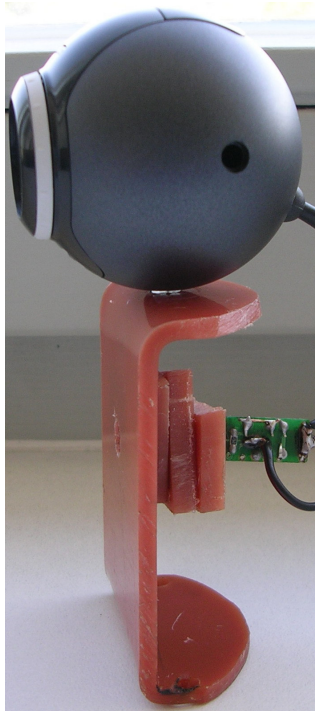
VideoOcx.Show – zobrazí zachycený obraz

VideoOcx.ReleaseImageHandle image – uvolní paměťového prostoru

VideoOcx.Close – ukončí videospojení

8.4 Realizace laserového zaměřovače

Zaměřovací systém (Obr. 33) tvoří 3 základní konstrukční prvky. Prvním základním prvkem je nad teplem ohýbaný plastový pásek vytvarovaný do žádaného tvaru. Na horní plošce plastového držáku je uchycena WEB kamera od firmy Logitech. Kamera byla rozebrána a upravena pro přišroubování. Posledním prvkem je zdroj laserového paprsku. Ten byl získán rozebráním obyčejného laserového ukazovátka. Jako napájecí zdroj laseru je použit stabilizovaný zdroj 3,5 V o jmenovitém proudu 300mA. Vzdálenost mezi optickou osou kamery a laserovým paprskem musí být známa. Tato hodnota musí být vložena do programu pod označením h_{cm} . Protože na základě tohoto údaje je počítána vzdálenost objektu.



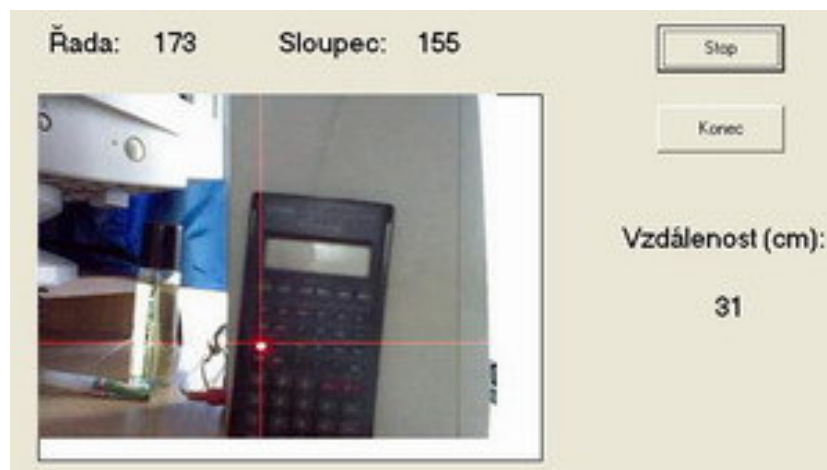
Obr. 33 Konstrukce zaměřovacího systému

8.4.1 Programové vybavení

Pro vytvoření programu bylo použito programovacího prostředí Visual Basic. Práce s Visual Basicem je založena na podobném principu jako u vývojového prostředí Sharp Develop.

Dominantní částí programu (Obr. 34) je okno s obrazovou informací z WEB kamery připojené k počítači. V obrazu je umístěn červený osový kříž, který je tvořen dvěma osami.

Průsečík os udává pozici nejjasnějšího bodu v obraze, tedy svítivý bod laseru. Obsluha programu je velice jednoduchá. Po kliknutí na tlačítko Start dojde k načtení připojené WEB kamery k počítači a automaticky se spustí měření vzdálenosti. Po namíření laserového paprsku na předmět je zobrazena vzdálenost zaměřovače od objektu. Pro ukončení měření slouží tlačítko Stop.



Obr. 34 Programové okno laserového zaměřovače

8.5 Programový kód laserového zaměřování

'definice popisku Sloupec

```
Private Sub col_lbl_Click()
```

```
End Sub
```

'Zobrazí pozici SLOUPCE

```
Private Sub col_val_Click()
```

```
End Sub
```

'Definice tlačítka EXIT

```
Private Sub exit_Click()
```

```
    If (Timer1.Enabled) Then
```

```
        Timer1.Enabled = False
```

```
VideoOCX.Stop
```

```
VideoOCX.Close
```

```
End If
```

```
End
```

```
End Sub
```

```
'Definice popisku ŘÁDEK
```

```
Private Sub row_lbl_Click()
```

```
End Sub
```

```
'Definice popisku VZDÁLENOST
```

```
Private Sub range_lbl_Click()
```

```
End Sub
```

```
'Zobrazí pozici ŘÁDKU
```

```
Private Sub row_val_Click()
```

```
End Sub
```

```
'Definice tlačítka START
```

```
Private Sub Start_Click()
```

```
    'Zaved' VideoOCX
```

```
    If (Not Timer1.Enabled) Then
```

```
    Start.Caption = "Stop"
```

```
    ' Zavedení ovladačů WEB kamery
```

```
    If (Not VideoOCX.Init) Then
```

```
        ' Dojde-li k chybě zobraz zprávu
```

```
        MsgBox VideoOCX.GetLastErrorString, vbOKOnly, "Neni pripojena kamera"
```

```
    End
```

Else

'Přidělení paměťového prostoru pro zachycené obrázky

capture_image = VideoOCX.GetColorImageHandle

'Spuštění časovače zachytávání obrazu

Timer1.Enabled = True

If (Not VideoOCX.Start) Then

' Nebude-li detekována kamera zobraz příslušnou chybovou hlášku

MsgBox VideoOCX.GetLastErrorString, vbOKOnly, "Cbyba VideoOCX"

End

End If

End If

Else

Start.Caption = "Start"

'Vypni časovač

Timer1.Enabled = False

VideoOCX.Stop

VideoOCX.Close

End If

End Sub

Private Sub Timer1_Timer()

'Deklarace proměnných

Dim matrix As Variant

Dim height, width As Integer

Dim r, c As Integer

Dim max_r, max_c As Integer

```
Dim max_red As Integer
```

```
Dim gain, offset As Variant
```

```
Dim h_cm As Variant
```

```
Dim range As Integer
```

```
Dim pixels_from_center As Integer
```

```
'přiřazení hodnot získané kalibrací
```

```
'zisk
```

```
gain = 0.0024259348
```

```
'kompenzace
```

```
offset = -0.056514344
```

```
'Vzdálenost laseru od kamery
```

```
h_cm = 5.842
```

```
max_red = 0
```

```
'Zachytávání obrazu
```

```
If (VideoOCX.Capture(capture_image)) Then
```

```
    'Načtení matice
```

```
        matrix = VideoOCX.GetMatrix(capture_image)
```

```
    'Vrátí výšku obrazu
```

```
        height = VideoOCX.GetHeight
```

```
    'Vrátí šířku obrazu
```

```
        width = VideoOCX.GetWidth
```

```
' Zpracování obrazu
```

```
' Vymezení zorné měřící oblasti
```

```
For r = height / 2 - 20 To height - 1
```

```
    For c = width / 2 - 25 To width / 2 + 24
```

'Hledá nejvyšší hodnotu pixelu v obraze

If (matrix(c, r, 2) > max_red) Then

max_red = matrix(c, r, 2)

max_r = r

max_c = c

End If

Next c

Next r

'Vypočet vzdálenosti laserového bodu od středu obrazu

pixels_from_center = max_r - 120

'Výpočet vzdálenosti objektu v cm na základě kalibračních hodnot

range = h_cm / Tan(pixels_from_center * gain + offset)

'zobrazení pozice laserového bodu v obraze

row_val.Caption = max_r

col_val.Caption = max_c

'Vykreslí červenou svislou čáru

For r = 0 To height - 1

matrix(max_c, r, 2) = 255

Next r

'Vykreslí červenou vodorovnou čáru

For c = 0 To width - 1

matrix(c, max_r, 2) = 255

Next c

VideoOCX.ReleaseMatrixToImageHandle (capture_image)

End If

VideoOCX.Show capture_image

End Sub

ZÁVĚR

Cílem této práce bylo vytvoření funkčního, levného a dostupného modelu robotického manipulátoru včetně programování polohové adaptivity motorů pro výuku robotiky na středních školách jako názorná pomůcka pro studenty při laboratorních pracích z robotiky. Při návrhu robotického manipulátoru byl kladen důraz na nízké pořizovací náklady s tím, aby byl dostupný studentům středních škol i při jejich mimoškolní tvořivé činnosti. Částka, za kterou je možné pořídit jednotlivé komponenty robotického ramene s podvozkem se pohybuje v cenové relaci do 2500 Kč. Servomechanismy je možné zakoupit v každém větším obchodě s modelářskými potřebami. Servomechanismus lze upravit i pro kontinuální otáčení, což umožňuje implementaci polohové adaptivity, takže robotický manipulátor byl vyroben i s podvozkem. Jako součást laboratorní výuky je i programování polohové adaptivity motorů.

Během diplomové práce jsem měl možnost prezentovat své výsledky na konferenci pod názvem Perspektivy elektroniky PEL 2009 v Rožnově pod Radhoštěm. Cílem konference bylo řešení metodické návaznosti výuky na středních školách na vysokoškolské studijní předměty. Při praktické ukázce řada účastníků konference projevila zájem o zařazení tohoto robotického systému včetně programového vybavení do laboratorních úloh.

ZÁVĚR V ANGLIČTINĚ

The aim of the thesis was design and development of purposive, reasonable and available robotic manipulator including the development of software environment for position adaptivity of actuators for education of robotics on secondary school as didactic tool for students at laboratory exercises from robotics. During the design of robotic manipulator were minimized the cost in order to be accessible for education of secondary school students and also for their out-of-school leisure time creative activities. The price of all parts needed for the assembly of robotic arm with mobile robotic platform is approximately 2500Kč. The parts of servomechanism are available in the hobby shop. Described servomechanism is possible adapt for continuous rotation suitable for the implementation of position adaptivity, what mediated to develop robotic arm together with mobile robotic platform. The part of laboratory exercises is the development of software for the position adaptivity of actuators.

The result of diploma thesis was presented at the conference “Perspective electronic“ PEL 2009 in Rožnov pod Radhoštěm. The topic of the conference was the solution of methodical continuity between the secondary school with universities. At the exhibition of robotic arm with mobile robotic platform many participants of the conference expressed the interest include this device and supporting software for the laboratory exercises

SEZNAM POUŽITÉ LITERATURY

- [1] Schmid D.: Řízení a regulace pro strojírenství a mechatroniku. *Nakladatelství Europa-Sobotales cz, Praha, 2005*. ISBN 80-86706-10-9.
- [2] Rotta, J. Bezstarosti, J. 2001. Robot manipulátor. *Praha : Amatérské rádio 2001, 7s - 11s, ISSN 1211-328x*.
- [3] Novák, Petr. MOBILNÍ ROBOTY - pohony, senzory, řízení. 1. vyd. *Praha : BEN, 2005*. 248 s. ISBN 807300111-1.
- [4] Webcam Based DIY Laser Rangefinder [online]. Version 1.5. Buffalo : Mark Tjersland, 2003, 19.3.2006 [cit.2009-04-21]. Dostupný z WWW:
<http://www.eng.buffalo.edu/ubr/ff03laser.php>
- [5] SURÝNEK, Tomáš. *Určení vzdálenosti cíle hloubkoměrným principem se strukturovaným světlem*. [s.l.], 2008. 63 s. UTB Zlín. Vedoucí diplomové práce Doc. Ing. Mgr. Milan Kvasnica, CSc.
- [6] Bastlení Atari Portfolio [online] Dostupný z WWW:<<http://vlastikd.webz.cz/bastl/servocontrol.htm>>
- [7] KOPLÍK, Petr. *Návrh a realizace nestabilní soustavy* . [s.l.], 2006. 55 s. Vysoké učení technické v Brně. Vedoucí diplomové práce Ing. Jiří Krejsa PhD. Dostupný z WWW: <www.umt.fme.vutbr.cz/umtmb/diplomky6/Koplik.pdf>.
- [8] BEZSTAROSTI, Jiří. WinSOS 2. Hobbyrobot [online]. 2006 [cit. 2008-04-2], s. 1-4. Dostupný z WWW: <www.hobbyrobot.cz>.
- [9] Kvasnica M.: Six-DOF Sensory System for Interactive Positioning and Motion Control in Rehabilitation Robotics. *International Journal of Human-Friendly Welfare Robotic Systems, Volume 4, Number 3, September 2003, Korea*. ISSN 1598-3250.
- [10] HORÁK , Martin. *Řízení motorů HS322 pro polohovou adaptivitu robotického ramene se zaměřovacím systémem pro určení vzdálenosti cíle*. In STOČ 2009 Ostrava. [s.l.] : [s.n.], 2009. s. 1-10.
- [11] HORÁK , Martin. *Řízení motorů HS322 pro polohovou adaptivitu robotického ramene se zaměřovacím systémem pro určení vzdálenosti cíle* In Perspektivy elektroniky 2009 : Moderní technologie v elektrotechnice předmětem výuky. SŠIEŘ Rožnov pod

Radhoštěm, : [s.n.], 2009. s. 35-39. Dostupný z WWW:
<<http://www.roznovskastredni.cz/dwnl/pel2009/PEL09.pdf>>. ISBN 978-80-254-40.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PWM Pulzně šířková modulace

DPS Deska plošného spoje

MKO Monostabilní klopný obvod.

PFC Počet obrazových pixelů od středu ohniskové roviny

Bd Baud - jednotka pro měření rychlosti přenesených dat

LED Light-Emitting Diode

CCD Charge-Coupled Device

SEZNAM OBRÁZKŮ

<i>Obr. 1 Robotické rameno [1]</i>	11
<i>Obr. 2 Rozdělení manipulačních systémů</i>	13
<i>Obr. 3 Princip dálkově ovládaných manipulátorů [1]</i>	14
<i>Obr. 4 Robot s pohyby v šesti osách [1]</i>	15
<i>Obr. 5 Pohyb s šesti stupni volnosti [1]</i>	15
<i>Obr. 6 Robot s kinematikou TTT [1]</i>	16
<i>Obr. 7 Robot s kinematikou RTT [1]</i>	17
<i>Obr. 8 Robot s kinematikou RRT [1]</i>	18
<i>Obr. 9 Kloubový robot s kinematikou RRR</i>	19
<i>Obr. 10 Modelářský servomechanismus [6]</i>	21
<i>Obr. 11 Konstrukční části servomechanismu</i>	22
<i>Obr. 12 Řídicí elektronika</i>	23
<i>Obr. 13 Závislost natočení hřídele na šířce pulzu [3]</i>	24
<i>Obr. 14 Průběh PWM na výstupu modulu</i>	24
<i>Obr. 15 Řídicí jednotka – pohled ze strany součástek</i>	26
<i>Obr. 16 Řídicí jednotka – pohled ze strany spojů</i>	26
<i>Obr. 17 Schéma zapojení řídicí jednotky [2]</i>	28
<i>Obr. 18 Sřídka PWM signálu [2]</i>	29
<i>Obr. 19 Princip generování PWM</i>	30
<i>Obr. 20 Ovládací program WINSOS2</i>	31
<i>Obr. 21 Free serial port monitor</i>	33
<i>Obr. 22 Úvodní obrazovka Sharp Develop</i>	36
<i>Obr. 23. Hlavní okno programovacího prostředí</i>	36
<i>Obr. 24 Main Forms</i>	37
<i>Obr. 25 Princip laserového zaměrování</i>	39
<i>Obr. 26 Robotické rameno</i>	42
<i>Obr. 27 Konstrukce kleštiny</i>	43
<i>Obr. 28 Konstrukční díly [2]</i>	43
<i>Obr. 29 Deska plošného spoje</i>	44
<i>Obr. 30 Zapojení integrovaného stabilizátoru</i>	45
<i>Obr. 31 Zapojení komunikačního kabelu [2]</i>	45

<i>Obr. 32 Ukázka programu ROB3</i>	48
<i>Obr. 33 Konstrukce zaměřovacího systému</i>	56
<i>Obr. 34 Programové okno laserového zaměřovače</i>	57