

Středoškolská odborná činnost 2006/2007

Obor 12 – Tvorba učebních pomůcek, didaktická technologie

Robotická Ruka – Robotic Arm

Autoři práce:

Vladimír Minczér

Radek Trunec

Petr Altman

Konzultant práce:

Ing.Petr Hlávka SPŠS Plzeň



Obr. 1. RoboticArm

Prohlašujeme tímto, že jsme soutěžní práci vypracovali samostatně pod vedením ing. Petra Hlávky a uvedli v seznamu literatury veškerou použitou literaturu a další informační zdroje včetně zdrojů z internetu.

V Plzni dne 5. března 2007

Vladimír Minczér

Radek Trunec

Petr Altman

OBSAH

1. Úvod.....	4
2. Charakteristika mechatronického zařízení	5
3. Elektronické obvody	6
3.1. Napájení řídicích modulů a servomechanismů	6
3.2. Použité servomechanismy	6
3.3. Použité součástky	6
3.4. Řídící deska	8
3.5. Použitý LCD displej	8
3.6. Použitý typ mikroprocesoru (ATmegaA16).....	9
3.7. Propojovací kabel PC - servořadič	10
3.8. Popis servomechanismu	11
3.9. Řízení servomechanismu.....	12
3.10. Signalizační LED diody	13
4. Strojní části	13
4.1. Popis konstrukce.....	13
4.2. Technologické postupy.....	13
4.3. Konstrukční řešení.....	15
5. Softwarové řešení	15
5.1. Programování.....	15
5.1.1. Použitý programátor mikroprocesoru:.....	16
5.2. Komunikace s PC.....	16
5.3. Ovládací software	16
5.3.1. Popis programu	16
5.3.2. Ovládání programu	19
5.3.3. Síťová komunikace TCP.....	19
5.3.4. Diagram ovládacího programu a komunikace.....	19
5.4. Popis částí zdrojových kódů pro mikroprocesor	22
6. Závěr	26
7. Anotace	26
Příloha A Seznam použité literatury.....	26
Příloha B Strojírenské výkresy	27
Příloha C Elektronická schémata.....	28
Příloha D Návrhy plošných spojů.....	29

1. Úvod

- Co je to robot
- Úkoly a cíle této práce
- Popis mechatroniky

Roboti se v současné době používají ve stovkách aplikací. Pracují v průmyslu, obsluhují stroje, používají technologické nástroje, ale také přímo pomáhají člověku - ve zdravotnictví, v domácnosti i při zábavě.

Z hlediska funkce se roboty dělí do dvou kategorií: **Servisní**
Průmyslové

Úkolem našeho projektu je zdokonalení v oblasti mechatroniky a využití nejmodernějších mechatronických principů. Toto mechatronické zařízení budeme dále prezentovat na soutěžích, naše ho škola použije při výuce programování a mechatroniky a také pro demonstraci mechatronických systémů při Dnu otevřených dveří.

Mechatronika je kombinace mechaniky, elektroniky, řízení a softwarového inženýrství. Účelem tohoto mezioborového inženýrského oboru je studium automatů z inženýrského pohledu a slouží k řízení vyspělých hybridních systémů.

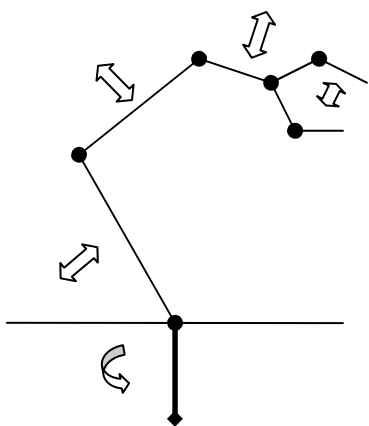
Mechatronika je umístěna mezi mechaniku, elektroniku, řídicí a výpočetní techniku, které dohromady umožňují vývoj jednodušších, ekonomičtějších, spolehlivějších a víceúčelových systémů. Slovo „mechatronika“ poprvé použil Tetsuro Mori, inženýr japonské firmy Yaskawa, v roce 1969. Sousloví „elektromechanické systémy“ a někdy „řízení a automatizace“ také označují mechatroniku.

Inženýrské studium mechatroniky typicky zahrnuje matematiku, mechaniku, návrh součástí strojů, termodynamiku, teorii obvodů, elektroniku, telekomunikace, teorii řízení, zpracování signálu, energetiku a robotiku.

2. Charakteristika mechatronického zařízení

- Princip mechatronického zařízení
- Blokové schéma připojení periférií

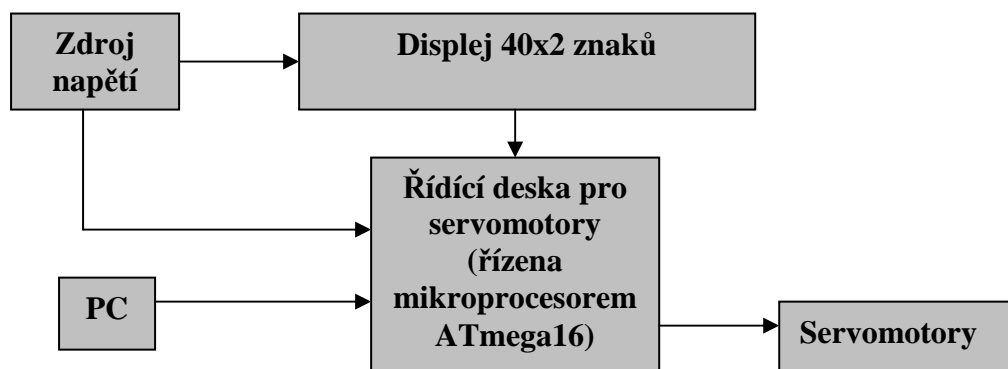
Robotická ruka je sestavena jako jednoduchý model robota s pěti stupni volnosti ovládaného počítačem, schopného manipulovat s tělesem ve třech osách.



Robot se skládá celkem ze čtyř hlavních částí:

1. Pevná základna
2. Otočný portál
3. Kloubové rameno
4. Uchopovací kleště

Blokové schéma řízení:



Z blokového schématu je vidět zapojení mikroprocesoru, který se stará o polohování servomotorů. Druhý mikroprocesor má za úkol obsluhu LCD displeje.

3. Elektronické obvody

3.1. Napájení řídicích modulů a servomechanismů

Napájení celé elektroniky robota je zajišťován spínaným počítačovým zdrojem, který byl speciálně upraven pro tyto účely.

Výstupní hodnoty zdroje: 5V, 16A.

U napájení je nutné, aby bylo rozděleno na napájení řídicí jednotky a na napájení servomechanismu z důvodů potřebného odrušení servomechanismů.

3.2. Použité servomechanismy

Do konstrukce robotické ruky bylo umístěno celkem šest servomechanismů Hitec (2 x HS-645MG byl použit ve dvou hlavních ramenech kde je největší namáhání, HS-422 tento servomechanismus byl použit na pohyb úchopových kleští v ose Y , servomechanismus HS-311 je použit na otáčení suportu v ose x, servomechanismus ES-05 je použit pro stisk úchopových kleští).

Servomechanismu HS-645MG má kovové převody a tažnou sílu 7.7Kg na páku 10mm při napájení 4.8V.

Servomechanismu HS-422 má tažnou sílu 3.2Kg na páku 10mm při napájení 4.8V.

Servomechanismu HS-805BB převodová skříň servomechanismu je osazena kuličkovými ložiskami má tažnou sílu 24Kg na páku 10mm při napájení 4.8V.

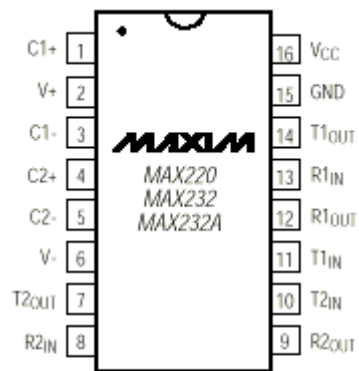
Servomechanismu HS-85MG má tažnou sílu 1Kg na páku 10mm při napájení 4.8V.

3.3. Použité součástky

MAX232 - Převodník

Jedná se o převodník TTL na RS232. Obsahuje dvě dvojice oddělovačů konvertujících napěťové úrovně. Napětí pro RS 232 se získává pomocí nábojové pumpy, a výstupní napětí proto značně závisí na kvalitě použitých kondenzátorů, která u elektrolytických kondenzátorů časem značně klesá. Napětí je možno získat na pinech 2 a 6 a použít pro další obvody.

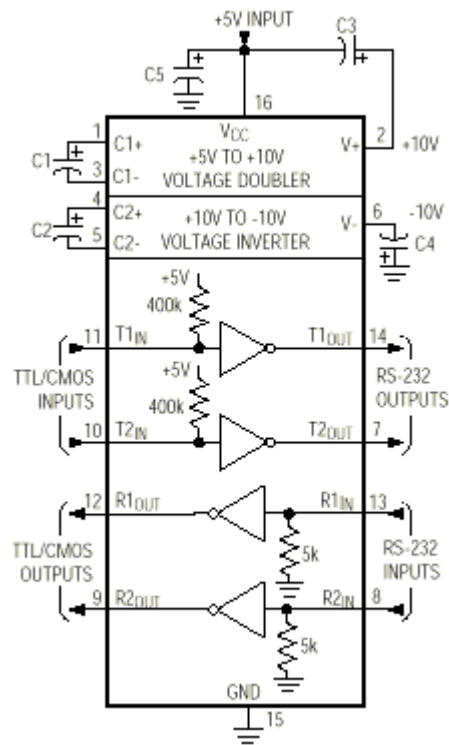
Obvod funguje vždy na první zapojení. Maxim vyrábí i verze s minimální externí kapacitou – (MAX 232A – 0,1 μ F) nebo verze pracující v rozsahu 7,5 – 13 V (určeno pro bateriové aplikace) – MAX 201 a MAX 231. Specialitou firmy MAXIM jsou obvody MAX203 a MAX 233, které dokáží pracovat úplně bez potřeby vnějších kondenzátorů.



DIP/SO

CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Obr. 2. Schéma optonu



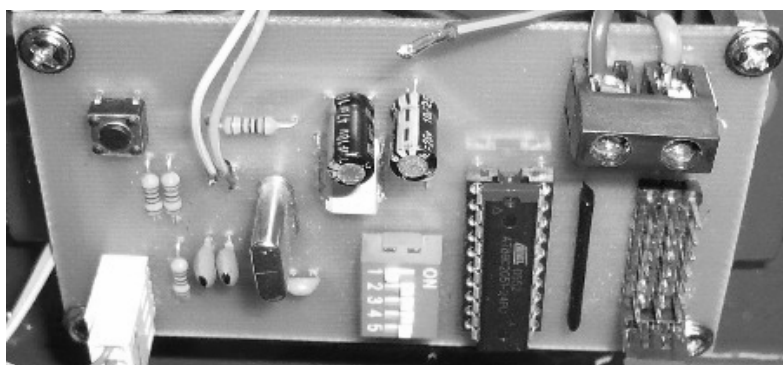
Obr. 3. Aplikační schéma

3.4. Řídicí deska

Pro účel objasnění funkce modulu předpokládáme, že je zdrojem řídicího signálu osobní počítač. Řídicí modul je generátorem řídicího PWM signálu pro šest servomechanismů, ovládaný povely z osobního počítače. Řídicí modul je osazen mikroprocesorem ATmega16 s taktovací frekvencí 16 MHz. Povely jsou posílány přes sériovou linku RS232. Pokud modul žádné povely z nadřazeného zařízení nepřijímá, na všech šesti výstupech neustále generuje poslední řídicí PWM signál. Servomechanismy dostávají stále řídicí signál PWM z řídicího modulu, tím je zajištěno, že se nemůže samovolně pohybovat výstupní osa servomechanismu.

Korektní start programu v mikroprocesoru je zaručen obvodem C2 a R1, nulovací obvod je zapojen podle doporučení výrobce mikroprocesoru. Na vývodech kondenzátoru C2 je zapojeno tlačítko SW1, které nám umožňuje RESTART mikroprocesoru.

Pro převod logické úrovně sériové linky RS232, mezi PC a mikroprocesorem, na logickou úroveň TTL je proveden pomocí integrovaného obvodu MAX232, který je následně připojen na sériové rozhraní mikroprocesoru.



Obr.4. Řídicí deska

3.5. Použitý LCD displej

Displeje z tekutých krystalů (anglicky **Liquid crystal display**, zkratkou **LCD**) se v dnešní době používají v mnoha aplikacích, potkáváme se s nimi na každém kroku, v tramvaji, různé informační tabule, domácí spotřebiče atd. Stávají se již běžným standardem.

Jejich velkou výhodou je malý odběr zobrazovací matice (řádově desítky mikroampérů), malé rozměry, nízká hmotnost, mají vyšší kmitočet obnovení informace oproti klasickým elektronickým obrazovkám a displejům. Naopak nevýhodou LCD displejů je teplotní závislost kapalných krystalů - při nízkých a vysokých teplotách ztrácejí své fyzikální vlastnosti a přestávají správně pracovat.

V naší konstrukci jsme použili LCD displej o velikosti 40x2 znaků. Displej používá nejběžnější řadič HD44780A00, který obsahuje mimo základní anglickou sadu také některé řecké, německé a japonské znaky. Základní znaky odpovídají ASCII kódování.

Displej může komunikovat buď 8-bitovou nebo 4-bitovou sběrnici. My jsme zvolili komunikaci se 4-bitovou sběrnici s procesorem od firmy Atmel ATmega16. Při tomto druhu komunikace jsou využívány jen 4 datové vodiče (DB4 až DB7) a zbytek datových vodičů je uzemněn. Posílaná data a instrukce jsou vyslány nadvakrát. Nejprve se na DB4-DB7 pošle horní polovina bytu, vygeneruje se kladný pulz E (hodinový vstup), na sběrnici se pošle spodní polovina bytu a generuje se druhý pulz na E.



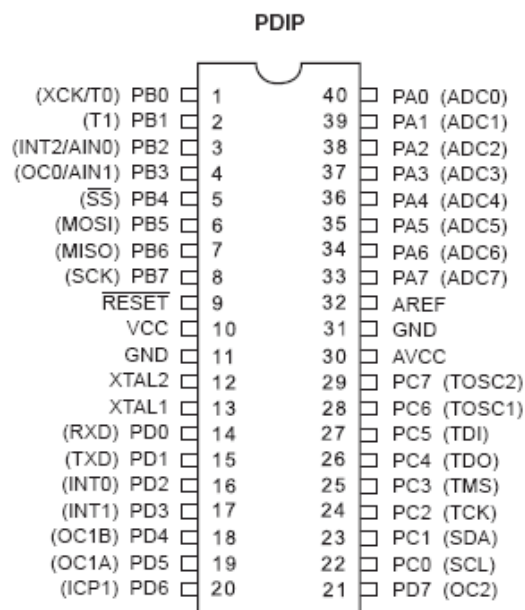
Obr. 5. LCD display

3.6. Použitý typ mikroprocesoru (ATmega16)

ATmega16 je 8-bitový mikroprocesor firmy [ATMEL](http://www.atmel.com). Mezi jeho základní vlastnosti patří:

- Programová Flash paměť velikosti 16 kB, zaručený počet přeprogramování je 10000 cyklů.
- Datová SRAM o kapacitě 1024 B.
- Napájecí napětí v rozsahu 4,5 až 5,5 V.
- 32 vstupně/výstupních linek, které mohou přímo budit LED (zkratový proud jedné linky je 20 mA, součet proudů všech výstupů nesmí překročit 80 mA).
- Dva 8-bitové čítače, jeden 16-bitový čítač, čtyři PWM kanály
- Programovatelný sériový kanál
- Osm A/D převodníků (Převodník analogového signálu na číslo)

Na obrázku obr. 6. je vidět rozmístění vývodů mikroprocesoru ATmega16 v pouzdru DIP40.

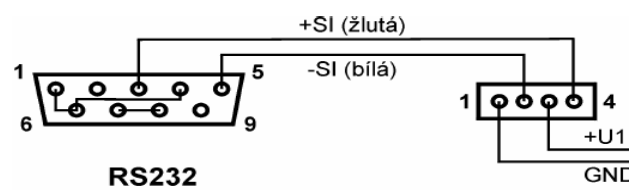


Obr.6 Rozmístění vývodů mikroprocesoru ATmega16

3.7. Propojovací kabel PC - servořadič

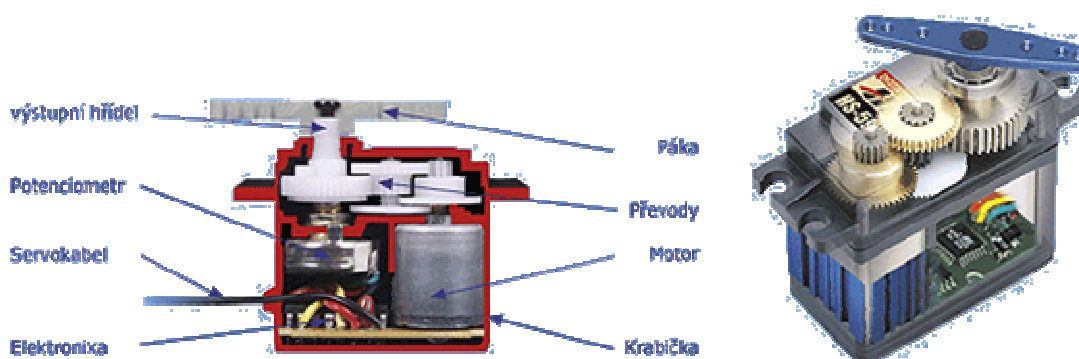
Potřebné signály sériové linky RS 232 (případně RS 422, RS 485) jsou k modulu připojeny sdruženým kabelem na jehož jednom konci je osazen nezáměnný sdružený konektor X1. Na opačném konci kabelu je osazena kabelová konektorová zásuvka Cannon 9, jejíž kontakty jsou propojeny tak, že zajišťují správnou komunikaci se sériovým rozhraním počítače třídy PC.

Pokud je sériová linka správně připojena, řídicí počítač spuštěn a řídicí modul má připojeno napájecí napětí, rozsvítí se na základně zelená signalizační dioda. Pokud probíhá komunikace mezi modulem a řídicím počítačem, signalizační dioda mění mírně jas svitu v rytmu toku přenášených dat.



Obr. 7. Zapojení komunikačního kabelu

3.8. Popis servomechanismu



Motor

tah („síla“) servomechanismu je zhruba úměrný velikosti motoru

Deska elektroniky

Dnes většinou zhotovena technikou plošné montáže (SMD)

Převody

převodový poměr určuje poměr mezi rychlostí a tahem servomechanismu. Běžné je, že se servomechanismy vyrábějí ve dvojicích, které mají stejný motor a elektroniku, ale liší se převody - např. HS-625 a HS-645. Jedno z nich je potom silnější, ale pomalejší, druhé má zvýšenou rychlost na úkor tahu. Převody jsou nejčastěji plastové, pro větší zatížení a větší spolehlivost pak kovové.

Potenciometr

snímá polohu výstupního hřídele a uzavírá tak zpětnou vazbu zajišťující správnou činnost servomechanismu. U menších a levnějších servomechanismu je potenciometr připojen přímo na výstupní hřídel, u servomechanismu vyšších kategorií je připojen přes zvláštní převod (tzv. nepřímý náhon), který velmi účinně chrání před přenosem vibrací. Poškození odporové dráhy potenciometru se může projevit jako zaškubávání servomechanismu v určité poloze nebo chvění v neutrální poloze.

Výstupní hřídel

přenáší pohyb mechanismu serva na ovládací páku. U servomechanismu menších a nižších kategorií je uložen v plastovém pouzdru, servomechanismy vyšších kategorií mají pro přesnější chod a vyšší životnost kuličková ložiska (jedno nebo ještě lépe dvě).

Krabička

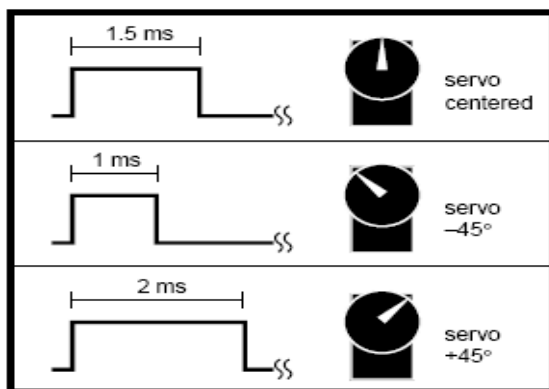
je vyrobena z houževnatého plastu a je opatřena patkami pro montáž.

Přívodní kabel

třížilový kablík se silikonovou izolací s velkou ohebností. U servomechanismu větších rozměrů bývá z krouceného vodiče pro potlačení indukce rušivých napětí při vysokých proudech. Servomechanismy Hitec a Multiplex jsou zapojena následovně: žlutý(oranžový) vodič: řídicí signál, červený: kladný pól napájení („plus“), hnědý: záporný pól napájení („minus“, „zem“).

3.9. Řízení servomechanismu

Žádaná poloha výstupního hřídele je do řídicí elektroniky zadávána jako pulsně šířkově modulovaný (PWM) signál s opakovanou frekvencí - ta je přibližně 50Hz. Nastavováním délky tohoto impulsu v rozsahu 1 až 2 ms je definována poloha výstupního hřídele servomechanismu v rozsahu 90° . Pro střední polohu hřídele je třeba vysílat kladný impuls o délce 1,5ms. PWM signál musí být do servomechanismu vysílán nepřetržitě – pokud ustane, uvolní se zpětná vazba servomechanismu a není nijak zajištěna správná poloha výstupního hřídele. Servomechanismy v základním provedení umožňují otáčení výstupního hřídele o 180 úhlových stupňů tím, že se rozsah kladného pulsu řídicího PWM signálu změní na 0,5 až 2,5 ms. Ovšem řízení takovým způsobem je možné jen u servomechanismů některých výrobců, protože by se mohla tím poškodit převodová skříň.



Obr. 8. Nastavení polohy hřídele servomechanismu

3.10. Signalizační LED diody

V základně robota jsou umístěny tři signalizační diody. První dioda z levé strany nám signalizuje propojení řídicího modulu s počítačem. Druhá dioda z levé strany signalizuje, že elektronika v robotu je pod proudem. A konečně třetí dioda ta signalizuje, že byl uchopen předmět.

4. Strojní části

- Popis konstrukce
 - Technologické postupy
 - Konstruktivní řešení
-

4.1. Popis konstrukce

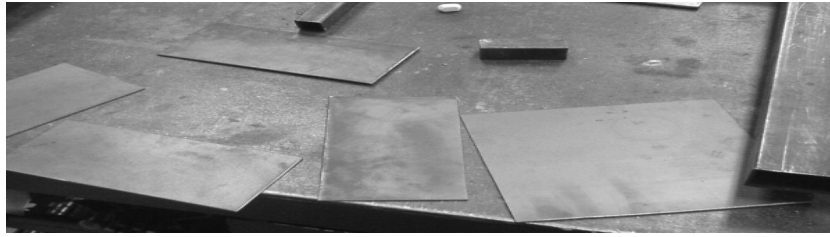
Celková konstrukce robota se skládá celkem ze čtyř základních částí

1. Základna pevná obdélníkového tvaru
 - plechový svařenec o síle plechu 2mm z oceli 11 373
 - rozměr 100x200x200mm (výška x šířka x hloubka)
2. Otočný portál ramene robota
 - plechový svařenec o síle plechu 2mm z oceli 11 373
 - rozměr 60x150x70mm (výška x šířka x hloubka)
2. Kloubové rameno
 - Al. profil: s.2mm, 30x30x400mm
 - s.2mm, 20x50x200mm
4. Uchopovací kleště
 - Teflonová deska s.6mm 200x100x30mm (výška x šířka x hloubka)

4.2. Technologické postupy

1. Základna pevná – výroba:

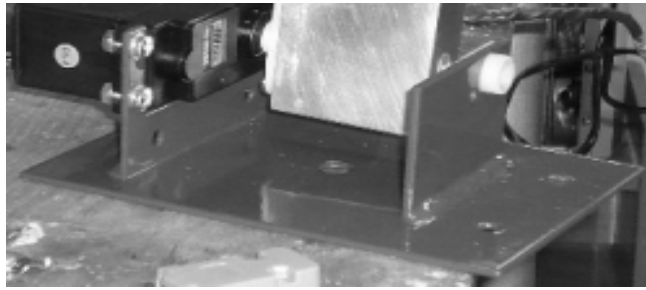
Nastřihání plechu dle výkresové dokumentace a bodové svaření konstrukce. Následné zavaření spojů a přebroušení hran. Vyříznutí otvorů pro LCD display a napájecí konektor, vyvrtání otvorů pro signalizační diody.



Obr. 9. Nastříhaný plech

2. Otočný portál – výroba:

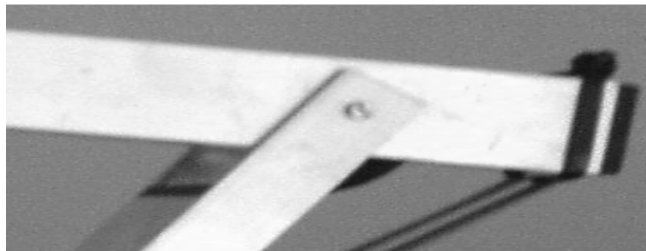
Nastříhání plechu dle výkresové dokumentace a bodové svaření konstrukce. Následné zavaření spojů a přebroušení hran. Montáž úchytů na servopohon a vyvrtání otvoru pro středovou hřídel. Přivaření krytů na servopohon s otvory pro napájení. Vytočení závitu M4 do kotevnicích otvorů. Následné zavaření spojů a přebroušení hran.



Obr.10. Otočný portál

3. Kloubové rameno – výroba:

Nařezání Al profilu dle výkresové dokumentace. Vyřezání tvaru pro kloubový ohyb a vyvrtání spojovacích děr a vytočení závitu M4. Montáž úchytů pro servopohon. Zaslepení profilu se zabroušením hran..



Obr. 11. Kloubové spojení ramen

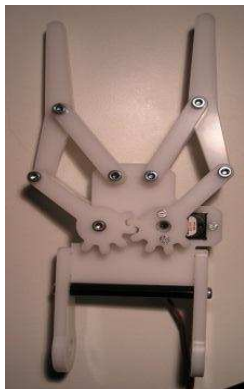
4. Uchopovací kleště – výroba:

Vyfrézování jednotlivých dílů z teflonové desky dle výkresové dokumentace. Vyvrtní díř pro uchycení servopohonu a základní desky k ramenu. Montáž ramen k pevné desce šroubovým spojem a osazení distančního sloupku na pevnou desku. Přilepení pasky pryže vteřinovým lepidlem na dotykové plochy uchopovacích kleští. Přichycení senzoru na koncové plochy ramen kleští.

4.3. Konstrukční řešení

- Uchopovací kleště :

Pro výrobu kleštiny jsme použili teflonovou desku, z níž jsou jednotlivé díly vyfrézovány. Skládá se ze základní desky, která je pomocí dvou ramen připevněna k servomotoru na koncovém ramenu. Na desce je vyfrézována díra pro uchycení servomotoru ES-05 pro ovládání čelistí. Na servomotoru je uchycena jedna z pák, která pomocí ozubení přenáší sílu na druhou ozubenou páku. Na každé z nich je pohyblivě uchycena jedna z čelistí. Každá čelist je spojena se základní deskou dvěma táhly. Tento systém dovoluje uchopit předmět o šířce 60 až 0 mm. Na konci každé čelisti je pro snazší uchopení předmětu nalepen gumový pásek. Pro spojení čelistí byly použity jako čepy imbusové šrouby M3 pojištěné na konci samojistící matkou. Mezi jednotlivými spoji je vložena fíbrová podložka pro snazší otáčení.



Obr. 12. Uchopovací kleště

5. Softwarové řešení

5.1. Programování

Poloha výstupní osy servomechanismu se řídí sledem příkazů posílaných z počítače sériovou linkou RS232; RS422... do řídicího modulu rychlostí 9600 Bd. Formát příkazu je <1 byte> <2 byte> <3byte>. Význam jednotlivých bytů:

<1 byte> synchronizační, vždy 255 (FFh)

<2 byte> číslo ovládaného servomechanismu (0-255), v modulu je možné použít pouze čísla 0 až 63.

<3byte> pozice výstupního hřídele servomechanismu (0 – 254) (00h – FFh).

Povely musí být ovládacím programem **RoboticArm.exe** posílány vždy v číselném formátu, nikdy ve formátu textovém. Povely je nutné posílat do řídicího modulu

pouze tehdy, požadujeme-li změnu polohy výstupního hřídele některého z ovládaných servomechanismů. Periodické opakování potřebného řídicího signálu PWM zajišťuje program sám automaticky. Rychlost otáčení hřídele servomechanismu mezi koncovými body, zadanými programem, je dána pouze konstrukcí převodové skříně servomechanismu, programově ji nelze ovlivňovat.

5.1.1. Použitý programátor mikroprocesoru:

- PAtmel 3

Podporované obvody : AT89C51, AT89LV51, AT89C52, AT89LV52, AT89C55, AT89LV55, AT89S8252, AT89LS8252, AT89S53, AT89LS53, AT89C1051, AT89C1051U, AT89C2051, AT89C4051, ATMEL AVR,

Programátor PAtmel verze 3 podporuje většinu procesorů řady 51 a AVR firmy Atmel. Software umožňuje všechny potřebné operace s podporovanými obvody, tj. čtení, zápis, mazání a verifikaci datové i programové paměti a nastavování lock i fuse bitů procesoru.



Obr. 13. Programátor PAtmel3

5.2. Komunikace s PC

Povely k otočení výstupní osy servomechanismu jsou do řídicí jednotky robota posílány z nadřazeného zařízení (většinou osobní počítač) oddělenou sériovou linkou. Díky oddělení vstupu řídicí jednotky je lhostejné zda sériová linka je typu TTL, proudová smyčka, RS232, RS422, nebo RS485.

5.3. Ovládací software

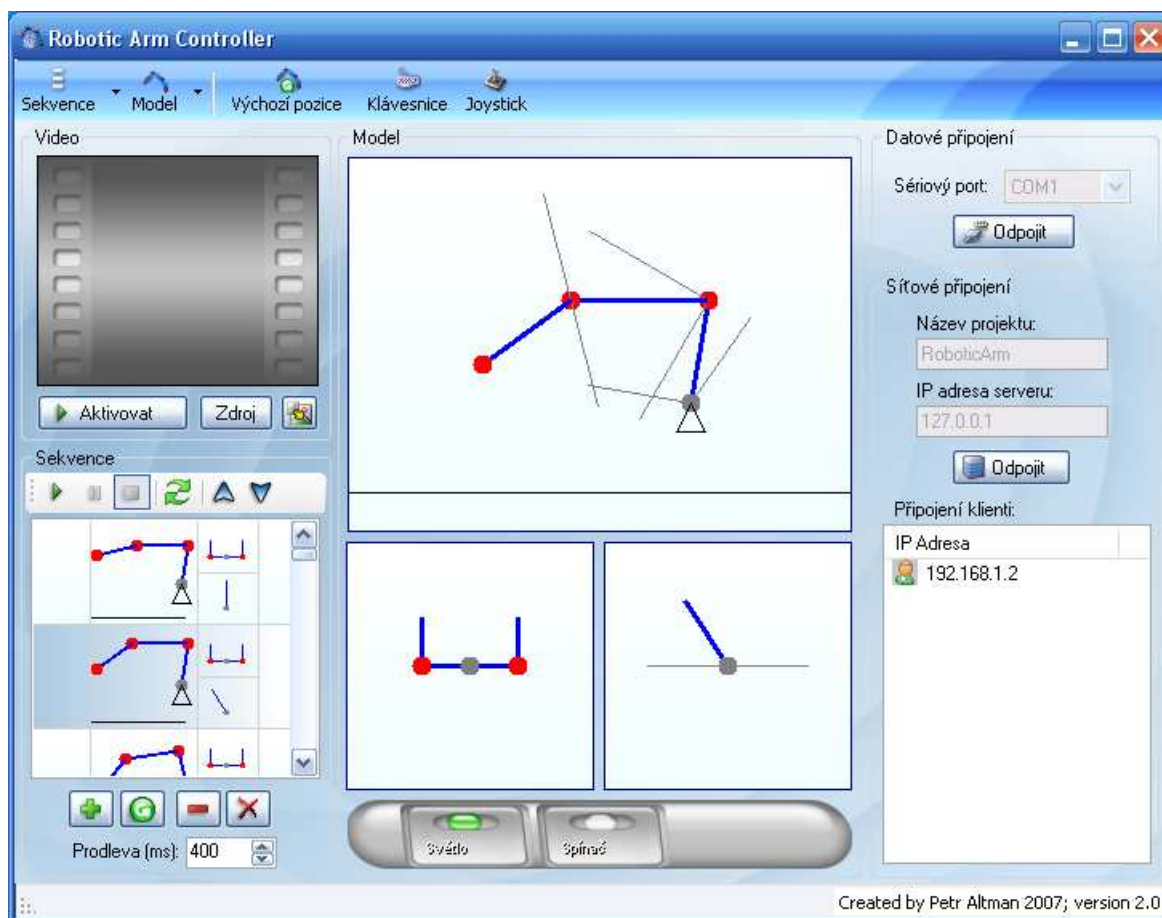
5.3.1. Popis programu

Pro ovládání a komunikaci s řízením manipulátoru jsme vytvořili ovládací program s možností grafického ovládání konstrukce manipulátoru. Grafické zobrazení je řešeno formou tzv. modelu, který lze libovolně nastavit pro případ dalšího vývoje, či využití programu v dalších projektech. Další možností programu je vizuální programování sekvencí pohybů principem point-to-point, které lze následně postupně

přehrávat. Manipulovat s jednotlivými rameny lze pomocí myši, klávesnice nebo joysticku a to jak přímo z řídicího programu, tak přes internet či místní síť pomocí serveru a klientské aplikace.

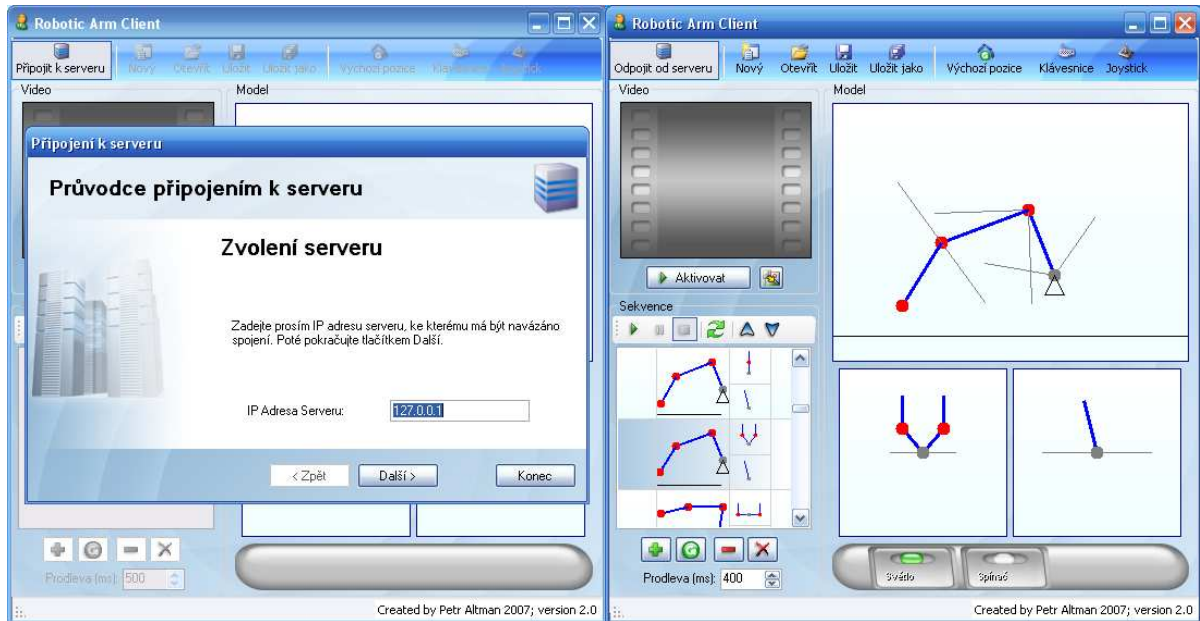
Popis spustitelných souborů:

- **RoboticArmController.exe**
Hlavní řídicí aplikace umožňující datové propojení s řídicí elektronikou manipulátoru, vytvoření projektu pro síťovou komunikaci, editaci a obsluhu sekvencí a modelu.



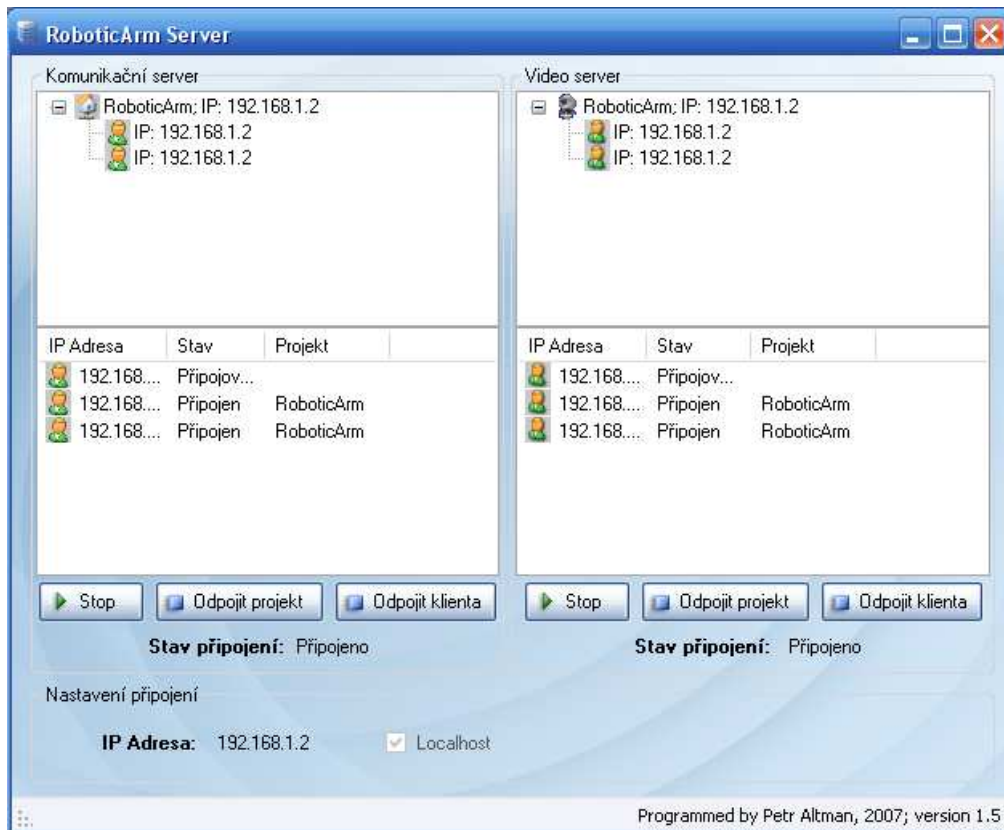
Obr.14 Řídicí aplikace

- **RoboticArmClient.exe**
Klientská aplikace, která slouží ke komunikaci s řídicím projektem. Umožňuje obsluhu modelu a editaci a obsluhu sekvencí.



Obr.15 Klientská aplikace

- **RoboticArmServer.exe**
Aplikace která vytváří síťový server, který zprostředkovává a řídí komunikaci mezi řídicí aplikací a klientskou aplikací.



Obr. 16 Aplikace server

5.3.2. Ovládání programu

Propojení ovládacího programu s řídicí elektronikou manipulátoru je zprostředkováno tlačítkem „Připojit“ ve skupině „Datové připojení“ v řídicí aplikaci. Komunikace probíhá přes sériový port pomocí protokolu RS232.

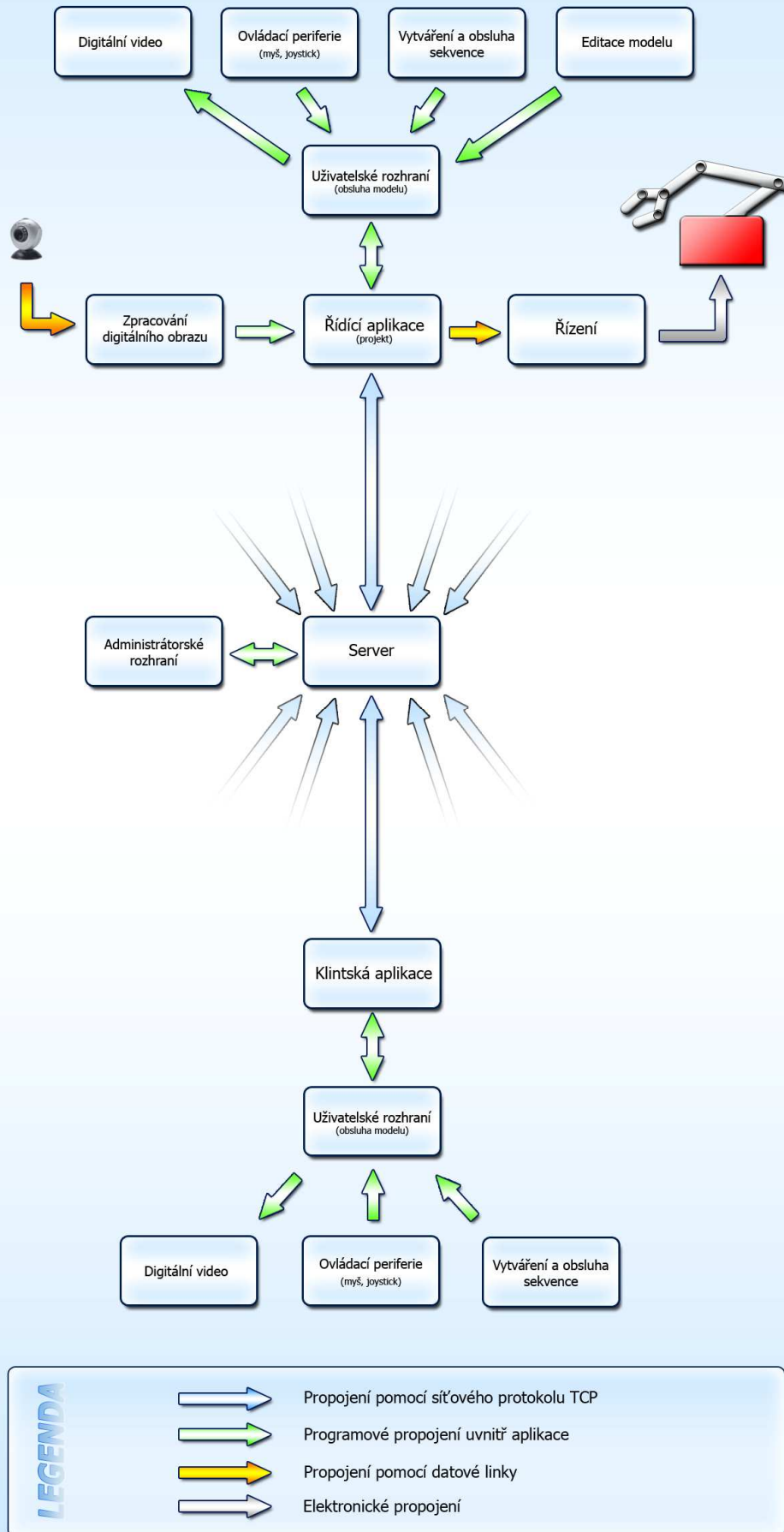
Pro manipulaci s rameny modelu je k dispozici grafické zobrazení jednotlivých kloubů, kterými lze pomocí myši pohybovat. Pokud je řídicí aplikace propojená s řídicí elektronikou manipulátoru, je reakce manipulátoru na změnu polohy ramena okamžitá. K pohybu rameny je také možno použít jakýkoli joystick a klávesnici. Pro vytváření programových sekvencí je k dispozici vizuální editor, kde jsou graficky zobrazeny jednotlivé kroky. Pokud je ovládání manipulátoru prováděné pomocí síťové komunikace, je k dispozici i vizuální kontakt s manipulátorem prostřednictvím **kamery**.

5.3.3. Síťová komunikace TCP

Síťová komunikace probíhá prostřednictvím protokolu TCP tzn. že lze manipulátor ovládat například přes **internet**. Použili jsme síťovou topologii **klient-server**. Jako centrum komunikace je v tomto případě server, který řídí a usměrňuje datovou komunikaci mezi jednotlivými klienty. Tito klienti se dále rozlišují na *projekty (řídicí aplikace)* a *klienty projektu (klientská aplikace)*. Server byl naprogramován tak, aby k němu bylo možno připojit několik řídicích aplikací a k těm se bylo možno připojit klientskou aplikací na základě jména projektu dané řídicí aplikace.

5.3.4. Diagram ovládacího programu a komunikace

Blokové schéma komunikace řídicího programu



Požadavky programu

Ovládací program je naprogramován v jazyce C# na moderní platformě Microsoft .NET Framework verze 2.0. Ke spuštění je zapotřebí operační systém Microsoft Windows 98/2000/XP, platforma MS.NET Framework verze 2.0 (možno stáhnout z www.microsoft.com) a DirectX 9.c. Pro využití síťové komunikace je dále zapotřebí podpora síťového připojení.

5.4. Popis částí zdrojových kódů pro mikroprocesor

Zdrojový kód pro mikroprocesor byl napsán v jazyce ANSI-C.

```
#define F_CPU 16000000UL // 16 MHz

#include <avr/signal.h>
#include <avr/interrupt.h>
#include "../api/api.c"
#include "defs.h"
#include "usart.c"
#include "lcd.c"
#include "servo.c"
#include "cmds.c"

// citac kroku behajiciho textu
u16 tp = 0;
// casovac pro behajici text
u32 textTimer = 0;
// pole aktualnich hodnot vstupu/vystupu
u08 iostates[] = {0, 0, 0, 0, 0, 0, 0, 0};
// logicka promenna urcujici, zda je pozadovana zpetna vazba od servomotoru
bool requestNotify = false;
// deklarace prikazu zpetne vazby servomotoru
command_struct cmd_notify;

// metoda pro nastaveni vystupu
void setIO(u08 address, bool state)
{
    iostates[address] = state; // ulozeni stavu vystupu do pameti
    write(&PORTC, address, state); // zapsani nove hodnoty na port
}

// metoda zajistujici kontrolu vstupu/vystupu
void checkIO()
{
    u08 value;
    int i;
    for(i = 0; i < 8; i++)
    {
        // nacteni aktualni hodnoty na vstupu
        value = getBit(PINC, i);

        // kontrola zmeny hodnoty na vstupu
        if(value != iostates[i])
        {
            // ulozeni aktualniho stavu do pameti
            iostates[i] = value;

            // hodnota byla zmenena, odeslani zpetne vazby
            command_struct cmd; // deklarace prikazu
            cmd.command = COMMAND_SETIO; // hlavicka prikazu
            cmd.paramsCount = 2; // pocet parametru prikazu
            cmd.params[0] = i; // index vstupu
            cmd.params[1] = value; // hodnota vstupu
            transmitCommand(cmd); // odeslani prikazu
        }
    }
}

// metoda vyvolana vzdy, je-li prijat prikaz po seriove lince
// metoda obsahuje parametr typu command_struct obsahujici informace o
// prijatem prikazu
```

```

void executeCommand(command_struct cmd)
{
    switch(cmd.command)
    {
        // prijmuti prikazu s pozadavkem o zpetnou vazbu pozice servomotoru
        case COMMAND_CALLBACK_REQUEST:
            requestNotify = true;
            break;
        // prijmuti prikazu s novou hodnotou natoceni hridele servomotoru
        case COMMAND_SERVOAD:
            ;
            // prijmuti uhlu natoceni
            s16 angle = cmd.params[1] | (cmd.params[2] << 8);
            // prijmuti rychlosti otaceni
            servo_setSpeed(cmd.params[0], cmd.params[3]);
            // nastaveni uhlu natoceni
            servo_setAngle(cmd.params[0], angle);
            break;
        // prijmuti prikazu s novou hodnotou daneho vystupu
        case COMMAND_SETIO:
            setIO(cmd.params[0], cmd.params[1]);
            break;
    }
}

void main(void)
{
    // rozsviceni inicializacni LED
    openPin(LED_INIC_DDR, LED_INIC_PIN);
    write(LED_INIC_PORT, LED_INIC_PIN, 1);

    // inicializace rozhrani serioveho USART
    usart_init(103);

    // nastavení pinů portu C jako vstupy/vystupy
    DDRC = 0xFF;

    // inicializace vystupu pro servomotory
    DDRB = 0xFF;
    servo_add(&PORTB, PB0);
    servo_add(&PORTB, PB1);
    servo_add(&PORTB, PB2);
    servo_add(&PORTB, PB3);
    servo_add(&PORTB, PB4);
    servo_add(&PORTB, PB5);
    servo_add(&PORTB, PB6);
    servo_add(&PORTB, PB7);
    servo_init();

    // inicializace vystupu pro LCD display
    LCD_init();
    fdevopen(LCD_data, 0, 0);

    // definovani prikazu pro zpetnou vazbu
    cmd_notify.command = COMMAND_CALLBACK;
    cmd_notify.paramsCount = 0;

    // povoleni globalniho preruseni
    sei();

    // prodleva
    _delay_ms(100);

    // zhasnuti inicializacni LED

```

```

write(LED_INIC_PORT, LED_INIC_PIN, 0);

u08 i;
bool notify = false;
while(true)
{
    // kontrola zda po seriove lince neprisel nejaky prikaz
    checkUSARTCommands();

    // zpetna vazba po ukoncení pohybu vsech serv
    if(requestNotify == true)
    {
        notify = true;
        // kontrola vystupu vsech serv, zda dosahly
        // pozadovane pozice
        for(i = 0; i < servo_count; i++)
        {
            if(servo[i].tick != servo[i].atick)
            {
                notify = false;
                break;
            }
        }

        // pokud vsechna serva dosahla pozadovane pozice,
        // odesli PC zpetnou vazbu
        if(notify == true)
        {
            // odeslani zpetne vazby
            transmitCommand(cmd_notify);
            notify = false;
            requestNotify = false;
        }
    }

    // kontrola vstupu
    checkIO();

    // behaci text
    if(textTimer > 7500)
    {
        // vytvoreni useku aktualne zobrazovaneho textu
        int j = tp;
        char* showText = "                ";
        for(i = 0; i < 20; i++)
        {
            showText[i] = TEXT[j];
            j++;

            if(j >= sizeof(TEXT)-1)
                j = 0;
        }

        // nastaveni kurzoru displeje na nulovou pozici
        LCD_moveTo(0, 0);

        // poslani textu na display
        printf("%s", showText);

        tp++;
        if(tp >= sizeof(TEXT)-1)
            tp = 0;
        textTimer = 0;
    } else {
        textTimer++;
    }
}

```


} }

5. Závěr

Naše práce nám pomohla se zdokonalit ve strojírenství, v elektronice a zejména v programování mikroprocesoru řady ATmega16. Tento robot má široké spektrum užití, např: při výuce programování, při demonstraci možností počítače, v reklamě, pro jednoduchá polohovací zařízení, nebo jen pro zábavu.

6. Anotace

RoboticArm je robot manipulátor schopný přemísťovat předměty ve třech osách. Celého robota řídíme pomocí počítače.. K pohybu celé konstrukce byly použity následující modelářské servomechanismy (HS-805BB,HS-422,ES-05,HS-645MG). Robot byl vytvořen pro výuku mechatronických principů a postupů . Celý projekt byl realizován v domácích podmínkách.

Příloha A Seznam použité literatury

Zdroje: <http://measure.feld.cvut.cz/groups/edu/osv/6N136.pdf>
<http://elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatice.html>
<http://rubick.wz.cz/default.php?url=cist&sekce=cit&idclanku=2>
<http://www.rentron.com/8051.htm>
http://www.sweb.cz/e78/clanky/uprava_serva_cla.htm
<http://rs232.hw.cz/>
<http://vlastikd.webz.cz/>
<http://www.hobbyrobot.cz/electronic.htm>
<http://robotika.cz/guide/servo/cs>
<http://hugo.xf.cz/datash/at89c2051.php>
<http://abcdimenze.wz.cz/linky/odkazy01.html>
http://prochazka.d2.cz/ovladac_lcd.php
http://tvorbawebu.wz.cz/faiirfe/help/ad_a_kvantovani.htm
<http://www.volny.cz/d72/>
<http://www.elektroworld.info/index.php>
<http://en.wikipedia.org/wiki/RS-232>
<http://rs232.hw.cz/#max232zapojeni>

Příloha B Strojírenské výkresy

Kompletní strojírenské výkresy jsou v tištěné verzi přidané na konci dokumentu, před výkresy elektronické. Na doprovodném CD jsou ve složce Strojní konstrukce

Příloha C Elektronická schémata

Kompletní elektronická schémata jsou v tištěné verzi přidané na konci dokumentu. Na doprovodném CD jsou ve složce Elektronické schémata

Příloha D Návrhy plošných spojů

Kompletní návrhy plošných spojů jsou v tištěné verzi přidány na konci dokumentu. Na doprovodném CD jsou ve složce Plošné spoje