

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ**

# **BAKALÁŘSKÁ PRÁCE**

**Delta-robot řízený z počítače**

**Martin Faktor**

**2013**

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
Fakulta elektrotechnická  
Akademický rok: 2012/2013

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin FAKTOR**  
Osobní číslo: **E10B0293P**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Elektronika a telekomunikace**  
Název tématu: **Delta-robot řízený z počítače**  
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

Navrhněte a realizujte robotickou platformu s "delta" mechanismem.

1. Navrhněte a realizujte HW řešení.
2. Navrhněte a naprogramujte vhodnou řídicí jednotku.
3. Navrhněte a naprogramujte řídicí aplikaci na PC.

Rozsah grafických prací: podle doporučení vedoucího

Rozsah pracovní zprávy: 20 - 30 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

**Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.**

Vedoucí bakalářské práce:

**Ing. Petr Weissar, Ph.D.**

Katedra aplikované elektroniky a telekomunikací

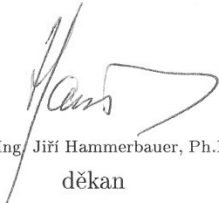
Konzultant bakalářské práce:

**Ing. Petr Weissar, Ph.D.**

Katedra aplikované elektroniky a telekomunikací


Datum zadání bakalářské práce: **15. října 2012**

Termín odevzdání bakalářské práce: **7. června 2013**

  
Doc. Ing. Jiří Hammerbauer, Ph.D.  
děkan



L.S.

  
Doc. Dr. Ing. Vjačeslav Georgiev  
vedoucí katedry

V Plzni dne 15. října 2012

## **Anotace**

Předkládaná bakalářská práce se zabývá návrhem a konstrukcí modelu delta robota ovládaného z počítače. V první části práce je popsán návrh řídicí jednotky a ovládání modelářských serv. Ve druhé části je práce zaměřena na popis kinematiky delta robota a v poslední části se práce zabývá popisem softwarového vybavení řídicí jednotky a aplikace pro osobní počítač.

## **Klíčová slova**

Delta robot, modelářské servo, mikrokontrolér, AVR, RS-232, kinematika.

## **Annotation**

The thesis deals with the proposal and the construction of a delta robot controlled through a computer. In the first part, control unit proposal and controls of the servo motors are described. The second part focuses on delta robot kinematics and the last part describes the control unit software and a personal computer application.

## **Key words**

Delta robot, servo motor, microcontroller, AVR, RS-232, kinematics.

## Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 5.6.2013

Martin Faktor

.....

## Obsah

<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>VI</b>
<b>ÚVOD.....</b>	<b>1</b>
<b>1 HISTORIE.....</b>	<b>2</b>
<b>2 FUNKCE A STRUKTURA MODELU ROBOTA.....</b>	<b>3</b>
<b>3 ŘÍDICÍ JEDNOTKA .....</b>	<b>4</b>
3.1 NAPÁJECÍ ČÁST .....	4
3.2 KOMUNIKAČNÍ ČÁST .....	5
3.2.1 Převodník úrovní RS-232 a TTL.....	6
3.2.2 Převodník USB ↔ RS-232.....	7
3.3 MODELÁŘSKÉ SERVO .....	7
3.3.1 Řídicí elektronika.....	8
3.3.2 Řízení modelářského serva .....	9
3.4 MIKROKONTROLÉR .....	9
3.4.1 Jádro AVR.....	10
3.4.2 Periferie .....	10
<b>4 KINEMATIKA.....</b>	<b>12</b>
4.1 KINEMATICKÉ STRUKTURY .....	12
4.1.1 Kinematická struktura delta robotu.....	13
4.2 INVERZNÍ ÚLOHA KINEMATIKY DELTA ROBOTA.....	14
<b>5 SOFTWARE PRO MIKROKONTROLÉR.....</b>	<b>18</b>
5.1 OVLÁDÁNÍ MODELÁŘSKÝCH SERV .....	18
5.1.1 Konfigurace Čítače/časovače I .....	19
5.2 KOMUNIKACE.....	20
5.2.1 Konfigurace USART jednotky.....	20
5.2.2 Komunikační protokol.....	20
5.2.3 Implementace komunikace v mikrokontroléru.....	21
<b>6 APLIKACE PRO OSOBNÍ POČÍTAČ .....</b>	<b>23</b>
6.1 KRESLENÍ.....	23
6.2 KOMUNIKACE.....	23
6.2.1 Paměť typu FIFO.....	24
6.3 MATEMATICKÝ MODEL .....	25
<b>7 KONSTRUKCE .....</b>	<b>26</b>
<b>8 ZÁVĚR.....</b>	<b>27</b>

<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>28</b>
<b>PŘÍLOHA A – SCHÉMA ŘÍDICÍ JEDNOTKY.....</b>	<b>30</b>
<b>PŘÍLOHA B – OSAZOVACÍ VÝKRES ŘÍDICÍ JEDNOTKY.....</b>	<b>31</b>
<b>PŘÍLOHA C – PARTLIST .....</b>	<b>32</b>
<b>PŘÍLOHA D – FOTOGRAFIE MODELU .....</b>	<b>34</b>



## Seznam použitých symbolů a zkratek

.NET	.....soubor technologií, které tvoří celou platformu
ATmega8	.....8-bitový mikrokontrolér
AVR	.....8-bitové jádro mikrokontroléru
BAUD [Bd]	.....modulační rychlost
C [F]	.....kapacita
$c$ [m]	.....délka vzpěry
C	.....programovací jazyk
C#	.....objektově orientovaný programovací jazyk společnosti Microsoft
CRC	.....cyklický redundantní součet
CTC	.....režim čítače/časovače1
$d$ [m]	.....vzdálenost mezi body <b>K'</b> a <b>Q</b>
$e$ [m]	.....vzdálenost středu platformy od kloubu na platformě
EEPROM	.....nevolatilní elektricky zapisovatelná paměť s libovolným přístupem
FIFO	.....paměť typu fronta
FLASH	.....nevolatilní elektricky zapisovatelná paměť s libovolným přístupem
$f_{osc}$ [Hz]	.....frekvence oscilátoru
GCC	.....sada překladačů programovacích jazyků
$I$ [A]	.....elektrický proud
ISO/OSI	.....referenční model pro standardizaci počítačových sítí
<b>K</b>	.....poloha kloubu na platformě
<b>K'</b>	.....střed kružnice $k_2$
$k_1, k_2$	.....kružnice
$l$ [m]	.....vzdálenost osy serva od středu základny
LED	.....dioda emitující světlo
$m$ [m]	.....délka úsečky <b>SQ</b>
MIPS	.....jednotka, udávající počet zpracovaných instrukcí za sekundu
$n$ [m]	.....délka úsečky <b>K'S</b>
N	.....dělicí poměr děličky
OCR1A	.....komparační registr čítače/časovače1
OCR1B	.....komparační registr čítače/časovače1
<b>P<sub>1</sub>, P<sub>2</sub></b>	.....průsečky kružnic $k_1$ a $k_2$
PC	.....personal computer

PROGMEM .....	makro pro ukládání dat pouze do programové paměti FLASH
PWM.....	pulzně šířková modulace
$q$ .....	rovina definovaná osami $y$ a $z$
<b>Q</b> .....	střed kružnice $k_1$
$r_1$ [m].....	délka ramena
$r_2$ [m].....	poloměr kružnice $k_2$
RISC.....	architektura procesoru s redukovanou instrukční sadou
RS-232, RS-485 .....	standard definující asynchronní sériovou komunikaci
<b>S</b> .....	bod ležící na úsečce <b>K'Q</b>
<i>SERVO</i> .....	proměnná udávající kanál serva
SPI.....	sériové periferní rozhraní
SRAM .....	Static Random Access Memory
$t$ [s] .....	čas
TCNT0 .....	čítací registr čítače/časovače0
TCNT1 .....	čítací registr čítače/časovače1
$t_{per}$ [s] .....	doba trvání jedné periody
TTL .....	tranzistorově-tranzistorová logika
$U$ [V] .....	elektrické napětí
UBRR.....	souhrnný název pro registry UBRRH a UBRRL
UBRRH.....	horní registr programovatelné děličky kmitočtu
UBRRL .....	dolní registr programovatelné děličky kmitočtu
UDR.....	vysílací a přijímací registr USART jednotky
USART .....	univerzální synchronní/asynchronní rozhraní
USB.....	univerzální sériová sběrnice
$v$ [m].....	výška trojúhelníku <b>K'P<sub>1</sub>Q</b>
$x, y, z$ .....	osy souřadného systému
$x', y', z'$ .....	souřadnice bodu <b>K</b> po otočení okolo osy $z$ o úhel $\varphi$
$x_K, y_K, z_K$ .....	souřadnice bodu <b>K</b>
$y_{K'}, z_{K'}$ .....	souřadnice bodu <b>K'</b>
$y_{P1}, z_{P1}$ .....	souřadnice bodu <b>P<sub>1</sub></b>
$y_Q, z_Q$ .....	souřadnice bodu <b>Q</b>
$y_S, z_S$ .....	souřadnice bodu <b>S</b>
$\theta$ [°] .....	kloubové souřadnice
$\varphi$ [°].....	úhel, který svírají serva mezi sebou

## Úvod

V posledních dvaceti letech prošly paralelní manipulátory obrovským vývojem. Mezi hlavní důvod rozvoje patří nesporně lepší parametry, než může nabídnout sériový manipulátor. Mezi hlavní výhody patří dosažitelná síla, přesnost, a velká rychlost. Současné paralelní manipulátory dosahují zrychlení až 50g a pracují s nanometrovou přesností. Díky těmto vlastnostem mají velice široké použití v průmyslu, od balení potravin až po letecké simulátory, kde díky své síle mohou manipulovat s kabinou pilota.

Nicméně i paralelní manipulátory mají své problémy a ne všechny byly za dobu vývoje vyřešeny. V minulých letech byl kladen důraz na vývoj paralelních struktur s šesti stupni volnosti. Tyto struktury nabízí vysokou tuhost celé konstrukce, nízkou setrvačnost a velké užitečné zatížení. Naproti těmto vlastnostem stojí problémy jako relativně malá pracovní plocha, složitá konstrukce a velké nároky na výpočetní výkon řídicího systému.

Proto, aby se vyhnulo zmíněným problémům, došlo v poslední době k zaměření vývoje na paralelní strukturu se třemi stupni volnosti. Tato struktura se vyznačuje vyšší rychlostí, jednodušším řídicím systémem a především pracovní plocha je oproti systému s šesti stupni volnosti větší. Příkladem této struktury je delta robot, který je díky svým vlastnostem ideální pro operace „pick and place“.

Tato bakalářská práce si klade za cíl navrhnout a zkonstruovat malý model delta robota řízeného počítačem.



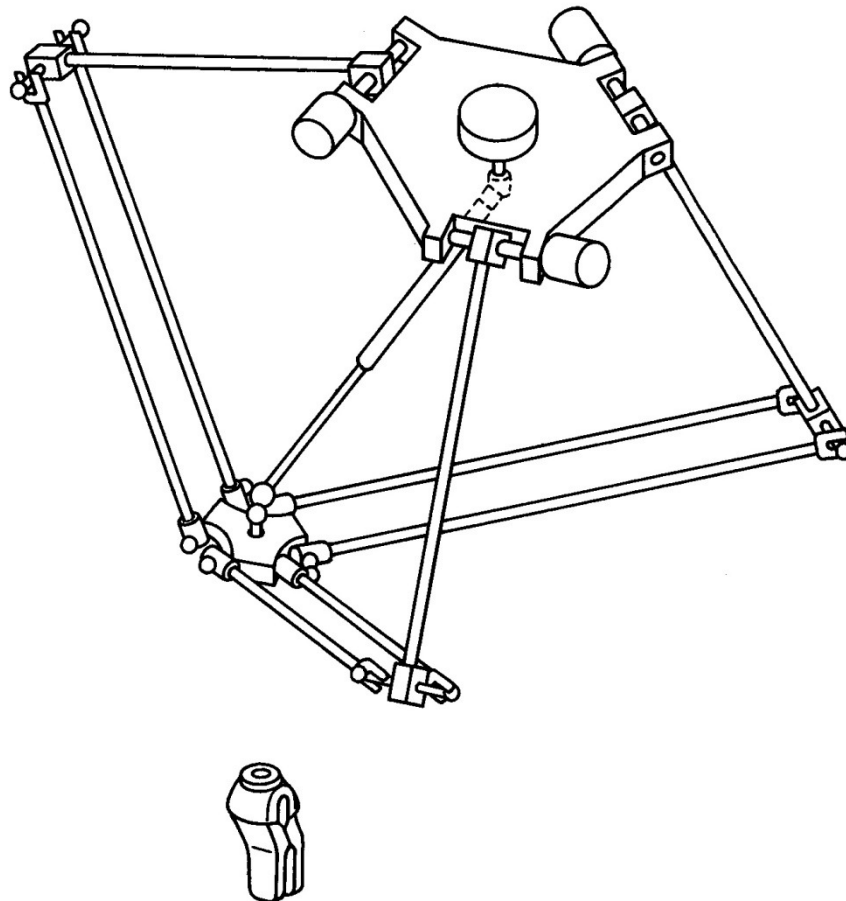
Obrázek 1: Letový simulátor [6]



Obrázek 2: TOSY „pick and place“ robot [7]

# 1 Historie

Počátkem osmdesátých let tým vedený profesorem Reymond Clavel působícím na École Polytechnique Fédérale de Lausanne (EPFL, Switzerland) přichází s geniálním nápadem a to využití paralelní kinematiky k návrhu nového typu robota určeného pro manipulaci s malými a lehkými objekty velmi vysokou rychlostí. Tento robot využíval tři translační stupně volnosti a jeden rotační stupeň volnosti. Tento návrh byl patentován jako (US patent No. 4,976,582). V roce 1987 švýcarská společnost Demareux zakoupila licenci na tohoto robota a zahájila výrobu pro průmysl. V roce 1991 Reymond Clavel představuje disertační práci „Conception d'un robot parallèle rapide à 4 degrés de liberté“ a v roce 1999 získává cenu „The golden robot award“. Následně vznikají různé modifikované varianty, jako například verze se třemi rotačními stupni volnosti vyvinuta firmou FANUC Corporation. Mezi firmy, které se dále zasloužily o rozvoj, patří ABB Flexible Automation a Sigpack Systems. [8].

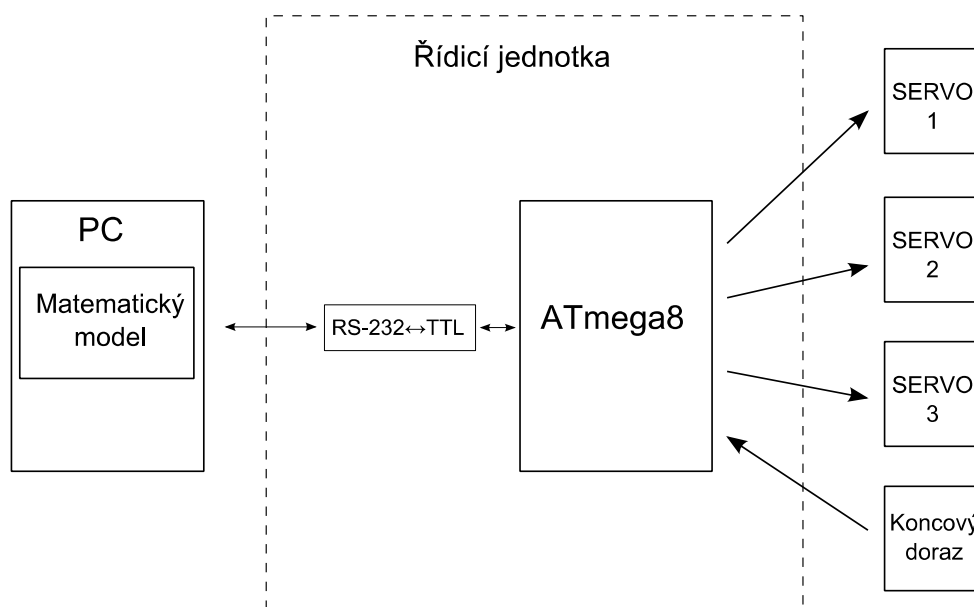


Obrázek 3: US Patent No: 4,976,582 [9]

## 2 Funkce a struktura modelu robota

Pro demonstrační funkci modelu byla vybrána jednoduchá úloha. V aplikaci na počítači je možné kreslit podobně jako v malování a delta robot přenáší nakreslený motiv na papír.

Struktura modelu je navržena s ohledem na řídicí jednotku. Řídicí jednotka je osazena 8-bitovým mikrokontrolérem, který nemá dostatečný výkon pro časově náročný výpočet kinematiky. Z toho důvodu je tento úkol přenechán osobnímu počítači, ve kterém běží aplikace, která se stará o výpočet a dále zajišťuje interaktivní komunikaci s uživatelem. Pomocí sérové linky a definovaného protokolu spolu počítač a řídicí jednotka komunikují, tudíž si předávají pouze velikosti úhlů, které má řídicí jednotka nastavit na jednotlivých akčních prvcích. Celá tato struktura je zobrazena na *obrázku 4*.



Obrázek 4: Struktura modelu

### 3 Řídicí jednotka

Řídicí jednotka je navržena tak, aby zajišťovala tři základní věci. První částí je napájecí obvod, druhou komunikační část a třetí částí je ovládání modelářských serv a koncového dorazu. Jelikož plánuji v budoucnu model značně rozšířit, nachází se ve schématu a na plošném spoji ještě další bloky, jako je integrovaný obvod ULN2803 pro buzení krokového motoru a konektor pro ovládání snímače barev. Celé schéma řídicí jednotky je v příloze A.

#### 3.1 Napájecí část

Napájecí obvod na řídicí jednotce zajišťuje napájení celého modelu. Z tohoto důvodu musí být napájecí obvod dostatečně dimenzován, aby byl schopen napájet tři modelářská serva, která mohou v součtu odebírat proud až 1,5A při napětí 5V. Dále musí být napětí na výstupu dostatečně stabilní a vyfiltrované pro napájení mikrokontroléru a periferních obvodů.

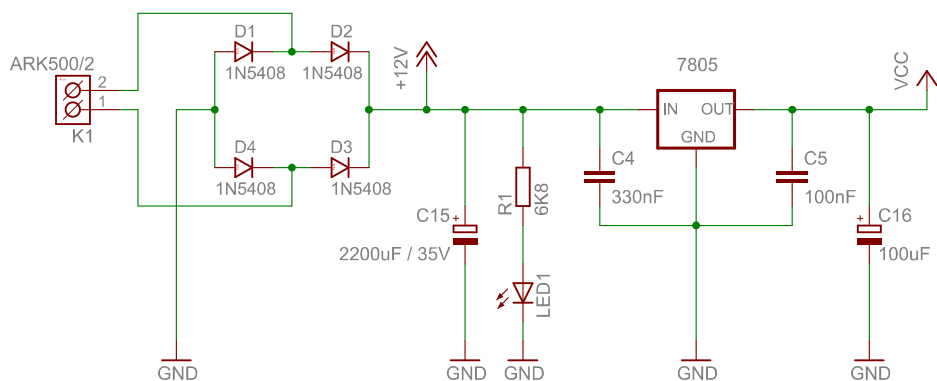
Pro flexibilní použití modelu je požadováno, aby byl schopen pracovat jak na střídavé napětí tak i na stejnosměrné. Z tohoto důvodu je na vstupu vytvořen Graetzův můstek ze čtyř výkonových diod BY299. Za můstkem následuje LED s ochranným rezistorem pro indikaci napájecího napětí.

Pro stabilizaci napětí je použit spojitý regulátor napětí 78S05, který dokáže dodat maximální proud 2A. Regulátor je doplněn o dva blokující keramické kondenzátory C4 a C5 zapojených dle doporučení výrobce.

K vyhlazení tepavého napětí z můstku slouží elektrolytický kondenzátor C15. Kapacitu je možné vypočítat dle vzorce (1) kde  $I$  je maximální odebíraný proud, tedy 1,6A,  $\Delta t$  odpovídá času 0,01s. Pro bezproblémový provoz stabilizátoru je nutné na jeho vstupní straně udržet napětí minimálně 8V. Předpokládané napájecí napětí z transformátoru je 12V efektivních, tedy 17V maximálních. Z toho vyplývá, že pokles napětí  $\Delta U$  je 7,6V. Při těchto parametrech vychází kapacita C15 na 2.1mF. Nejbližší vyráběná hodnota je 2,2mF. Poslední elektrolytický kondenzátor C16 je opět pro vyhlazení.

$$C = \frac{I \cdot \Delta t}{\Delta U} [F] \quad (1)$$

Ve schématu je ještě 12-ti voltový výstup, který bude sloužit pro napájení krokových motorů v plánovaném budoucím rozšíření modelu o pásový dopravník. Celé schéma napájecí části je na obrázku 5.

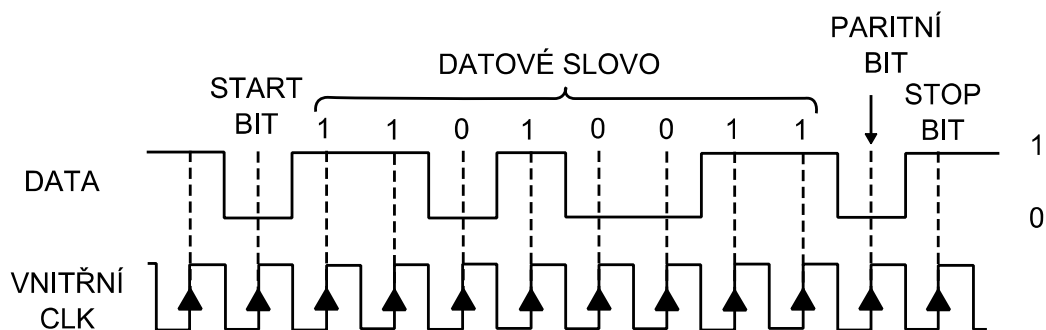


Obrázek 5: Schéma napájecího obvodu

### 3.2 Komunikační část

Přenos dat mezi mikrokontrolérem a počítačem je přes komunikační rozhraní standardu RS-232. Tento standard definuje pouze fyzickou vrstvu v modelu ISO/OSI, to znamená, že se zabývá pouze tím, jakým způsobem přenést data mezi dvěma zařízeními. Přenos probíhá duplexně po dvou párech vodičů asynchronně. Data jsou vysílána sériově od nejméně významného bitu po bit nejvýznamnější. V rámci je možné volit počet datových bitů od 5 do 9. Přenos je možné zabezpečit vložením paritního bitu, ovšem tato kontrola je většinou nedostatečná. Maximální modulační rychlost je 115200Bd. Ostatní rychlosti jsou získány dělením maximální rychlosti.

Klidový stav na sběrnici je v logické „1“. Přenos začíná vysláním start bitu log „0“, který slouží pro zasynchronizování přijímače. Po start bitu následují datové bity, za které je možné připojit paritní bit. Celý rámeček je zakončen jedním nebo dvěma stop bity v log „1“, které uvedou linku do klidového stavu a je možné přenášet další rámeček. Celý rámeček je zobrazen na obrázku 6.



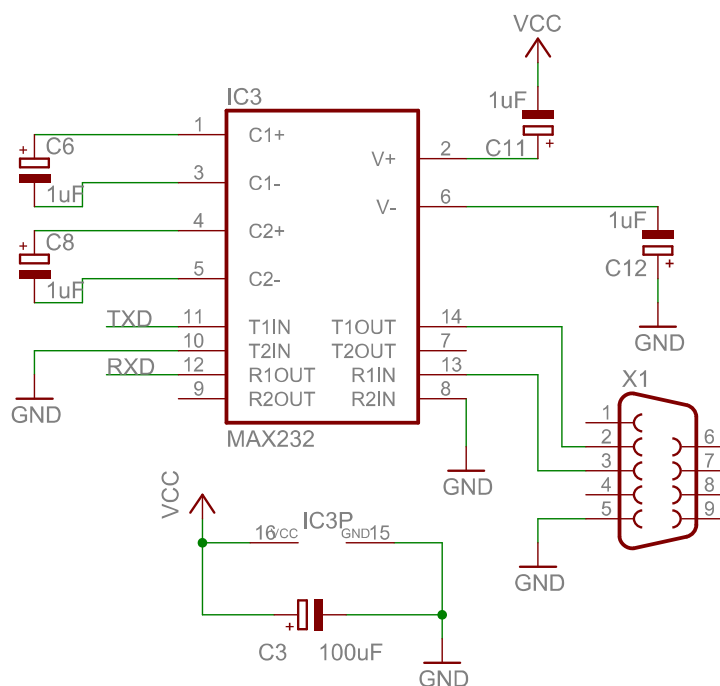
Obrázek 6: Přenosový rámeček

Kromě tvaru rámce jsou definovány také logické úrovně. Logická „0“ na straně vysílače musí být v rozmezí od +5V do +15V a na straně přijímače mezi hodnotami +3V až +15V. Logická „1“ má stejné hladiny ale v záporných hodnotách napětí.

### 3.2.1 Převodník úrovní RS-232 a TTL

Pro řídicí jednotku je použit mikrokontrolér ATmega8, který disponuje jedním sériovým kanálem. Sériový kanál má všechny vlastnosti, které byly popsány výše jako standard RS-232 kromě logických úrovní signálů. Mikrokontrolér pracuje s tzv. TTL logikou, kde logická „0“ představuje napětí 0V až 0,8V a logická „1“ napětí 2V až 5V. Díky těmto nekompatibilním úrovním musí být mezi mikrokontrolér a sériový port vložen převodník úrovní.

Integrovaný obvod MAX232 patentovaný a vyráběný firmou Maxim Integrated Products slouží pro převod úrovní RS-232 a TTL. Tento obvod obsahuje celkem čtyři převodníky. Dva z RS-232 na TTL a dva v opačném směru. Díky tomu je na jednom čipu možné provozovat dvě nezávislé linky bez řídicích signálů, nebo jednu linku s pomocnými řídicími signály. Zapojení převodníku je dle doporučení výrobce, který toto zapojení uvádí v datasheetu [10] a je zobrazeno na *obrázku 7*.



Obrázek 7: Převodník úrovní RS-232 a TTL



### 3.2.2 Převodník USB ↔ RS-232

V dnešní době se na osobních počítačích a notebookách téměř ustoupilo od implementace rozhraní RS-232. To bylo nahrazeno novějším rychlejším a univerzálnějším rozhráním USB. Nicméně sériová linka má stále nezastupitelnou roli v průmyslu, kde se používají modifikace jako např. RS-485. Z toho důvodu výrobci vyrábějí převodníky USB↔RS-232. Tyto převodníky jsou většinou postaveny na čipech FT232 od firmy FTDI. Po nainstalování ovladačů se převodník chová jako virtuální port a je možné k němu přistupovat jako ke standardnímu sériovému portu.

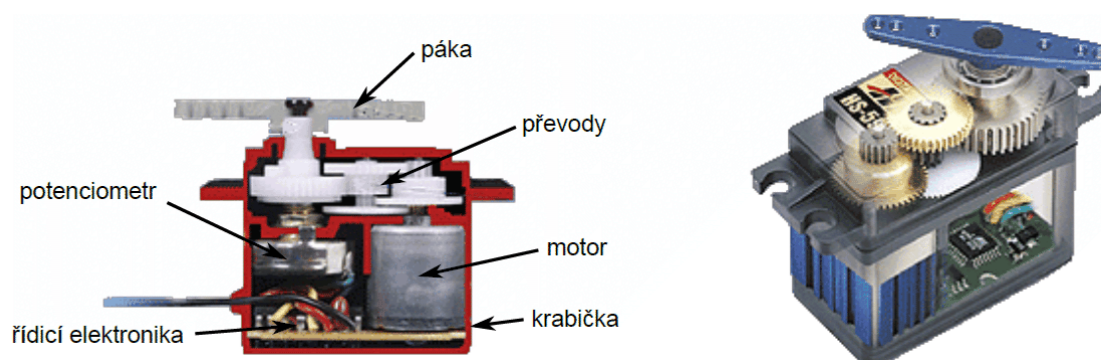
Při použití těchto převodníků platí určitá omezení plynoucí z vlastností rozhraní USB. Převodník netvoří přímé fyzické spojení mezi aplikací a sériovou linkou. Data ze sériové linky jsou ukládána do bufferu v převodníku a odtud si je operační systém vyčítá. V závislosti na operačním systému a dokonce jeho vytížení může být tato doba příliš dlouhá a proměnná. Řádově se jedná o desítky milisekund. Toto chování může způsobovat problém v případech, kdy posíláme z počítače data a od mikrokontroléru očekáváme okamžitou odpověď.



Obrázek 8: Převodník USB ↔ RS-232 [11]

### 3.3 Modelářské servo

Modelářské servo je elektromechanický měnič používaný v modelářství na přeměnu elektrické energie na mechanickou práci. Skládá se z malého stejnosměrného motoru, převodovky, řídicí elektroniky a krabičky. Vyrábějí se v různých velikostech od pikoserv o velikosti 1x2x1cm až po velká maxiserva o velikosti 6x3x5cm. Velikost serva přímo ovlivňuje jeho parametry, které jsou dány vlastnostmi stejnosměrného motoru.



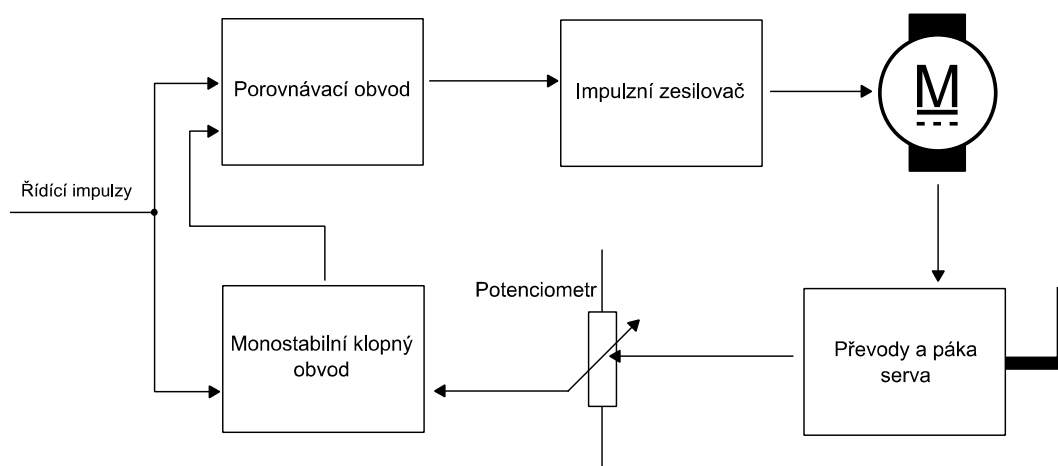
Obrázek 9: Průřez modelářským servem [12]

Parametry zvoleného serva Hitec HS-311:

Napájecí napětí	4,8V	6V
Síla [Kg/cm]	3	3,5
Rychlost [sec/60°]	0.19	0,15
Rozměry [mm]	40x20x36.5	
Hmotnost [g]	43	

### 3.3.1 Řídicí elektronika

Skládá se z porovnávacího obvodu, impulzního zesilovače a monostabilního klopného obvodu. Princip funkce je znázorněn na *obrázku 10*.



Obrázek 10: Schéma řídicí elektroniky modelářského serva

Řídicí impulsy vstupují do monostabilního klopného obvodu. V monostabilním klopném obvodu je s náběžnou hranou řídicích impulsů generován impuls o délce, která je závislá na poloze potenciometru a má opačnou polaritu než řídicí impulsy. Potenciometr je spřažen s

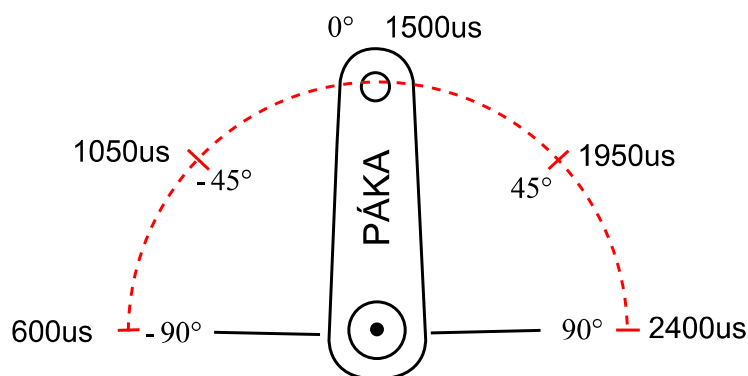
pákou serva a tudíž tvoří zpětnou vazbu o poloze páky serva. Řídicí signál a signál z monostabilního klopného obvodu vstupují do porovnávacího obvodu, kde jsou vyhodnoceny a podle toho, který trvá déle, generuje kladné nebo záporné impulzy. Dále impulzy prochází zesilovačem, který dodá dostatečný výkon pro roztočení motoru. Jakmile je řídicí signál stejný jako signál generovaný z monostabilního klopného obvodu, tak na výstupu porovnávacího obvodu je nulové efektivní napětí a motor se zastaví. Servo dosáhlo požadované polohy.

Z výše uvedeného vyplývá, že čím se signály budou blížit stejné délce, tím na výstupu porovnávacího obvodu bude efektivní hodnota napětí menší. To má za následek nekonstantní sílu serva, která klesá směrem k požadované poloze.

Celý tento systém je velice podobný funkci fázového závěsu s integračním článkem, který nahrazuje potenciometr zůstávající v dané poloze.

### 3.3.2 Řízení modelářského serva

Modelářské servo se řídí krátkými pulzy generovanými s frekvencí 50Hz. Poloha natočení serva je přímo úměrná šířce pulzů. Standardně mají serva výchylku  $\pm 45^\circ$ , ale většina serv zvládá úhel  $\pm 90^\circ$ . Středová poloha odpovídá délce pulzu 1500us a koncové polohy 600us a 2400us.



Obrázek 11: Výchylka modelářského serva

## 3.4 Mikrokontrolér

Pro řídicí jednotku delta robota byl vybrán mikrokontrolér ATmega8 od firmy ATMEL. Výběr padl na tento mikrokontrolér z důvodu nízké ceny a dostupnosti na našem trhu. Další výhodou je jeho snadná programovatelnost a vývojové prostředky, které jsou volně dostupné na stránkách výrobce [13].

### 3.4.1 Jádru AVR

AVR je 8-bitové RISC jádro využívající Harvardské architektury, tedy prostory dat a programu jsou odděleny i s jejich sběrnicemi. K dosažení maximálního výkonu AVR využívá jednoúrovňový pipeline, tedy zatímco jedna instrukce je vykonávána, další je načítána z paměti programu. Toto řešení umožňuje vykonávat instrukci každý hodinový takt, a proto se strojový cyklus rovná hodinovému cyklu. Vzhledem k tomu, že většina instrukcí trvá jeden strojový takt, tak AVR dosahuje výkonu až 1MIPS na 1MHz. Maximální pracovní kmitočet je 16MHz. Jádro dále obsahuje 32 obecně použitelných 8-bitových registrů. Posledních šest registrů lze spojit do dvojic na tři 16-ti bitové registry, pomocí nichž lze adresovat paměť. K jádru jsou připojeny tři druhy pamětí. FLASH (8KB) paměť pro program, EEPROM (512B) pro uložení konfiguračních dat a SRAM (1KB) pro data. Mikrokontrolér nemá vyvedenou adresní ani datovou sběrnici.

### 3.4.2 Periferie

V této části budou stručně popsány jen ty periferie, které byly použity.

#### Čítač/časovač0

Je základní, obecně použitelný, jednokanálový, 8-bitový čítač/časovač. Čítač má na vstupu nastavitelnou 10-ti bitovou děličku. Čítač umí čítat pouze směrem nahoru a přerušení může být vyvoláno pouze při přetečení 8-bitového čítacího registru. Do čítacího registru je přímý přístup pro čtení i zápis.

#### Čítač/časovač1

Tento čítač je 16-ti bitový a mnohem propracovanější. Spolu s Čítačem/časovačem0 sdílejí stejnou 10-ti bitovou děličku, ale každý si jí může individuálně nastavit, proto se chovají nezávisle. Dále tento čítač disponuje dvojicí nezávislých komparačních jednotek OCR1A a OCR1B. Tento čítač může pracovat v několika rozdílných modech.

První je Normal Mode. Při tomto nastavení je funkce stejná jako u Čítače/časovače0, to znamená, že se inkrementuje čítací registr TCNT1 a pokud dosáhne vrcholu, tj. hodnoty 0xffff, je v následujícím kroku vyvoláno přerušení a čítací registr může být znovu přednastaven.

Clear Timer on Compare Match (CTC) Mode už využívá komparační jednotky. Funkce je taková, že při každém inkrementování je porovnávána hodnota čítacího registru TCNT1 s

registry OCR1A a OCR1B a pokud je splněna rovnost, může být vyvoláno přerušení. Ostatní mody jsou různé modifikace PWM generovaných na výstupních pinech mikrokontroléru.

### Sériový kanál

Sériová jednotka může pracovat jak v synchronním tak asynchronním režimu. Může pracovat v plně duplexním režimu, jelikož má nezávislé registry pro příjem a odesílání. Jednotka podporuje přenos 5 až 9 datových bitů plus bit paritní. Pro nastavení přenosové rychlosti slouží dělič kmitočtu, tudíž jednotka není závislá na přesných zdrojích hodinového signálu. Jednotka může generovat tři různé druhy přerušení: dokončení vysílacího cyklu, dokončení přijímacího cyklu a vyprázdnění vysílacího registru.

### I/O porty

Mikrokontrolér ATmega8 má dva osmibitové a jeden sedmibitový port. Většina pinů má alternativní funkce, které je možno aktivovat přes speciální registr. Každý pin je ovládaný trojicí registrů a pokud je nakonfigurován jako vstupní, lze k němu připojit pull-up rezistor. Maximální odebíraný proud z jednoho pinu je 40mA, ale celkový proud nesmí překročit 300mA.

## 4 Kinematika

Pod slovem kinematika robotů se skrývá celý vědní obor, který se zabývá studii geometrie, pohybu a trajektorie, po kterých se pohybují jednotlivé body robota. V kinematice existují dvě základní úlohy. Přímá úloha kinematiky a nepřímá úloha kinematiky.

Přímá úloha kinematiky slouží k získání souřadnic efektoru ze znalosti jednotlivých poloh akčních členů. Tato úloha není nezbytně nutná pro samotný pohyb robota, ale využívá se při inicializaci a kalibraci, kdy známe polohy akčních členů a musíme určit polohu efektoru a následně přemístit efektor do určité pozice. Tato úloha se považuje obecně za jednodušší.

Nepřímá úloha kinematiky transformuje světové souřadnice efektoru na kloubové souřadnice. Tato úloha se vypočítává neustále při změně polohy efektoru a pro řízení robota je nezbytná. Uživateli tedy stačí znát polohu, kam se má efektor přemístit a pomocí této úlohy vypočte polohy jednotlivých pohonů.

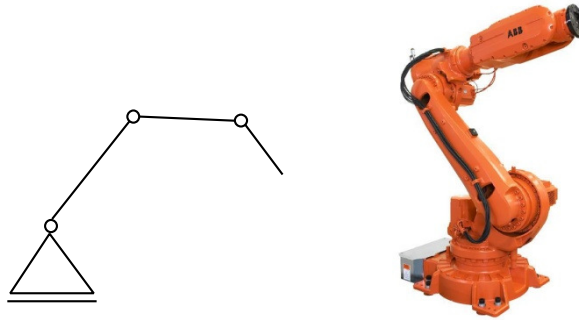
### 4.1 Kinematické struktury

Propojením dvou pevných těles pomocí pohyblivého mechanismu vzniká kinematická dvojice. Spojením kinematických dvojic vzniká kinematický řetězec. Podle topologie kinematických řetězců dělíme roboty do tří základních skupin.

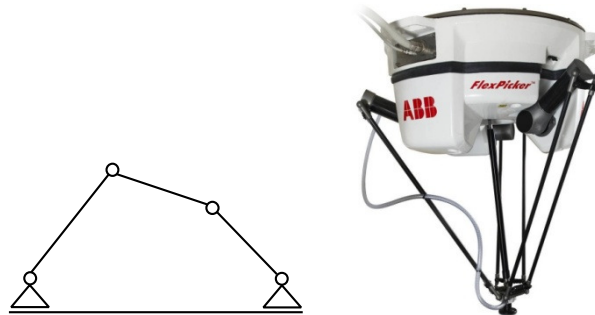
Sériová kinematická struktura. Tato struktura je nejpoužívanější u průmyslových robotů. Jedná se o řazení kinematických dvojic za sebe a tak vzniká otevřený kinematický řetězec. Řazení kinematických dvojic sériově způsobuje největší namáhání konstrukce v krutu a ohybu. Díky tomu musí být konstrukce výrazně masivnější, což má za následek nežádoucí zvýšení hmotnosti a snížení tuhosti robota. S rostoucí hmotností také roste setrvačnost robota. Dále chyba, která vzniká na konci řetězce, je součtem všech chyb akčních prvků.

Paralelní kinematická struktura je tvořena kinematickými dvojicemi, které jsou spojeny paralelně a tvoří uzavřený kinematický řetězec. Jednotlivé kinematické dvojice spolupracují a pro pohyb je většinou nutný pohyb všech akčních členů, což má za následek složitější řízení a tím i nároky na výpočetní výkon. Výhodou této struktury je namáhání konstrukce, které je v tahu a tlaku. Díky těmto vlastnostem mohou mít tyto roboty podstatně menší hmotnost a větší tuhost. Díky menší hmotnosti dosahují i větší rychlosti a přesnosti. Příkladem této struktury je delta robot.

Posledním skupinou je hybridní kinematická struktura, která je složením obou předešlých skupin dohromady. Příkladem může být delta robot, který má na platformě obráběcí hlavu, která se může libovolně natáčet.

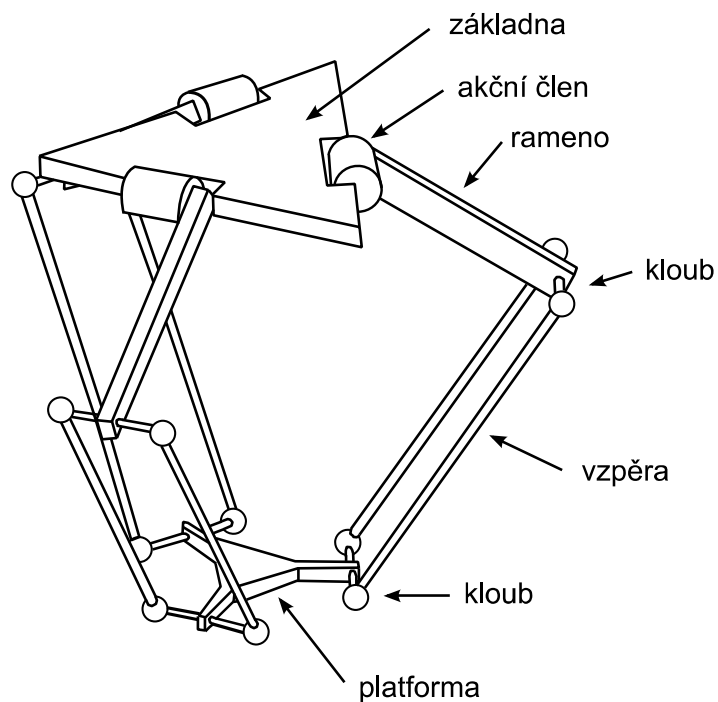


Obrázek 12: Příklad sériového řetězce a průmyslový robot ABB IRB 6620 [14]



Obrázek 13: Příklad paralelního řetězce a průmyslový robot ABB IRB 360 [15]

#### 4.1.1 Kinematická struktura delta robotu



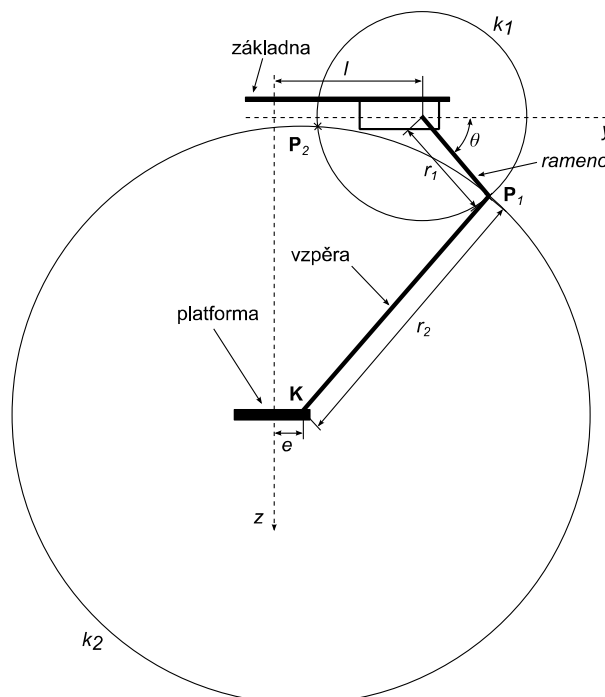
Obrázek 14: kinematická struktura delta robotu

Delta robot je tvořen pevnou základnou tvaru rovnostranného trojúhelníku. Podél základy jsou umístěny akční členy s rameny, které jsou kolmé na hrany základny. Na konci ramenou jsou kulové klouby, které spojují vzpěry do paralelogramu. Druhý konec vzpěr je připojen na platformu opět kulovými klouby. Díky vzpěrám spojených do paralelogramu má delta robot tři stupně volnosti, tudíž platforma nemůže rotovat vůči pevné základně.

## 4.2 Inverzní úloha kinematiky delta robota

Jak již bylo zmíněno výše, inverzní úloha kinematiky spočívá v nalezení kloubových souřadnic při známé poloze efektoru. Řešení tohoto problému v této práci je založeno čistě na využití analytické geometrie.

Pokud se rozpojí kulové klouby mezi rameny a vzpěrami, rozpadne se celá struktura na dvě části. První část se základnou, akčními členy a rameny a na druhou část s platformou a vzpěrami. Akční členy na první části mohou pohybovat rameny, které svými konci opisují kružnice o poloměru, který odpovídá délce ramen. Na druhém dílu jsou pohyblivé vzpěry, které se teoreticky mohou v prostoru pohybovat po kulových plochách. Řešení inverzní úlohy spočívá v nalezení průsečíků kulových ploch a kružnic. Díky této úvaze lze problém výrazně zjednodušit, jelikož konkrétní poloha efektoru odpovídá pouze dvěma průsečíkům a lze řešit každý paralelní řetězec nezávisle na ostatních.



Obrázek 15: Geometrické řešení



Rameno opisující kružnici  $k_1$  se může pohybovat pouze v rovině  $q$  definované osami  $y$  a  $z$ . Ve stejné rovině jsou i průsečíky  $\mathbf{P}_1$  a  $\mathbf{P}_2$ , které leží na kružnici  $k_1$ . Kružnice  $k_2$  je získána průnikem kulové plochy a roviny  $q$ , ve které leží kružnice  $k_1$ . Poloměr kružnice  $k_2$  je závislý na vzdálenosti bodu  $\mathbf{K}[x_K; y_K; z_K]$  od roviny  $q$ . Kulová plocha je popsána rovnicí (2), kde  $c$  je délka vzpěry.

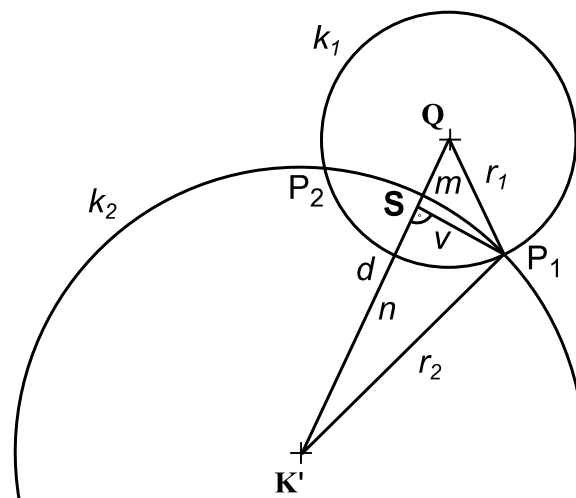
$$(x - x_K)^2 + (y - y_K)^2 + (z - z_K)^2 = c^2 \quad (2)$$

Jelikož rovina  $q$  má nulovou souřadnici  $x$ , lze udělat průnik tak, že v rovnici kulové plochy je vynulována souřadnici  $x$ . Po úpravě vznikne rovnice kružnice, ze které je vyjádřena závislost poloměru kružnice  $k_2$  na vzdálenosti bodu  $\mathbf{K}$  od roviny  $q$ .

$$(y - y_K)^2 + (z - z_K)^2 = v^2 - x_K^2 \Rightarrow r_2 = \sqrt{c^2 - x_K^2} \quad (3)$$

Nyní se celé řešení přesouvá do roviny  $q$ , ve které jsou hledány průsečíky  $\mathbf{P}_1$  a  $\mathbf{P}_2$ . První kružnice  $k_1$  se středem  $\mathbf{Q}[y_Q; z_Q]$  a poloměrem  $r_1$ . Druhá kružnice  $k_2$  o středu  $\mathbf{K}'[y_{K'}; z_{K'}]$  a poloměru  $r_2$ . Tyto dvě kružnice mohou mít vůči sobě různé vzájemné polohy. První poloha je taková, kdy kružnice mají střed ve stejném bodě tzv. soustředné kružnice. V tomto případě kružnice nemají žádný společný bod a této polohy efektoru nelze dosáhnout. Stejně tak kružnice nesoustředné (jedna kružnice leží uvnitř druhé), které nemají společný střed, ale rovněž nemají společný bod. Poslední možnost, která vede k nedosažitelné pozici efektoru, je taková, kdy vzdálenost mezi středy kružnic je větší než součet obou poloměrů kružnic.

Nyní uvažujeme polohu, kde kružnice mají dva společné body a tím pádem i poloha efektoru je dosažitelná.



Obrázek 16: Řešení průniků dvou kružnic [16]

Vzdálenost středů kružnic  $d$  je určena Pythagorovou větou:

$$d = \sqrt{(y_Q - y_{K'})^2 + (z_Q - z_{K'})^2} \quad (4)$$

Při představě, že středy kružnic jsou dva vrcholy trojúhelníka, tak třetí bod je hledaný průsečík. Délky stran trojúhelníka tedy odpovídají poloměrům kružnic  $k_1, k_2$  a délce úsečky  $d$ . Rozdělením úsečky  $d$  na dvě části  $m$  a  $n$  vznikne bod **S**. Bod **S** tvoří vrcholy dvou trojúhelníků  $SP_1Q$  a  $K'P_1S$ . Tyto trojúhelníky mají společnou výšku  $v$  a platí rovnice 5.

$$\begin{aligned} r_1^2 &= v^2 + m^2 \\ r_2^2 &= v^2 + n^2 \end{aligned} \quad (5)$$

Po odečtení obou rovnic:

$$r_1^2 - r_2^2 = m^2 - n^2 = (m+n)(m-n) = d(m-n) \quad (6)$$

Z rovnice je získán rozdíl mezi  $m$  a  $n$  a vyjádřena délka  $m$ :

$$\frac{r_1^2 - r_2^2}{d} = m - n = 2m - d \Rightarrow m = \frac{r_1^2 - r_2^2}{2d} + \frac{d}{2} \quad (7)$$

Pro určení polohy bodu **S** je použita lineární interpolace:

$$\begin{aligned} y_S &= y_Q + \left(\frac{m}{d}\right)(y_{K'} - y_Q) \\ z_S &= z_Q + \left(\frac{m}{d}\right)(z_{K'} - z_Q) \end{aligned} \quad (8)$$

Pomocí Pythagorovy věty je vyjádřena výška  $v$ :

$$v = \sqrt{r_1^2 - m^2} \quad (9)$$

Průsečíky  $P_1$  a  $P_2$  jsou získány záměnou souřadnice  $y$  za  $z$  a následné přičtení výšky  $v$  kolmé na úsečku  $d$

$$\begin{aligned} y_{P1} &= y_S + \left(\frac{v}{d}\right)(z_Q - z_{K'}) & y_{P2} &= y_S - \left(\frac{v}{d}\right)(z_Q - z_{K'}) \\ z_{P1} &= z_S - \left(\frac{v}{d}\right)(y_Q - y_{K'}) & z_{P2} &= z_S + \left(\frac{v}{d}\right)(y_Q - y_{K'}) \end{aligned} \quad (10)$$

Ze dvou možných průsečíků je vybrán ten, který je vzdálenější od osy  $z$ , tedy průsečík  $\mathbf{P}_1$ . Na závěr je vypočten úhel  $\theta$ , který svírá rameno se základnou.

$$\theta = \arcsin\left(\frac{z_{P1}}{r_1}\right) \quad (11)$$

Celý výpočet je zopakován pro všechny tři paralelní řetězce. Pomocí *rotační matice* (12) je bod  $\mathbf{K}$  otočen okolo osy  $z$  o úhel  $\varphi$ , který svírá osa  $y$  s požadovaným řetězcem. Tím je opět získáno řešení v rovině  $q$ .

$$\begin{aligned} x' &= y_K \cdot \sin \varphi + x_K \cdot \cos \varphi \\ y' &= y_K \cdot \cos \varphi - x_K \cdot \sin \varphi \\ z' &= z_K \end{aligned} \quad (12)$$

Souřadnice  $x_K, y_K, z_K$  jsou souřadnice bodu  $\mathbf{K}$  před rotací. Nové souřadnice po rotaci jsou  $x', y', z'$ .

## 5 Software pro mikrokontrolér

Řídicí jednotka je osazena mikrokontrolérem ATmega8. Tento mikrokontrolér je postaven na jádru AVR, které má instrukční sadu optimalizovanou pro použití překladačů jazyka C. Jazyk C byl tedy zvolen pro programování mikrokontroléru. Jako vývojové prostředí bylo zvoleno AVR studio ve verzi 4.19 Build 730, které obsahuje GCC překladač. Toto vývojové prostředí nabízí ATMEL zdarma na svých internetových stránkách [13]. Pro přenos programu do mikrokontroléru byl zvolen programátor STK500, který je integrován do vývojového prostředí. Programátor využívá pro přenos dat SPI rozhraní. Vývojové prostředí dále obsahuje propracovaný simulátor, který významně zjednodušuje ladění programu.

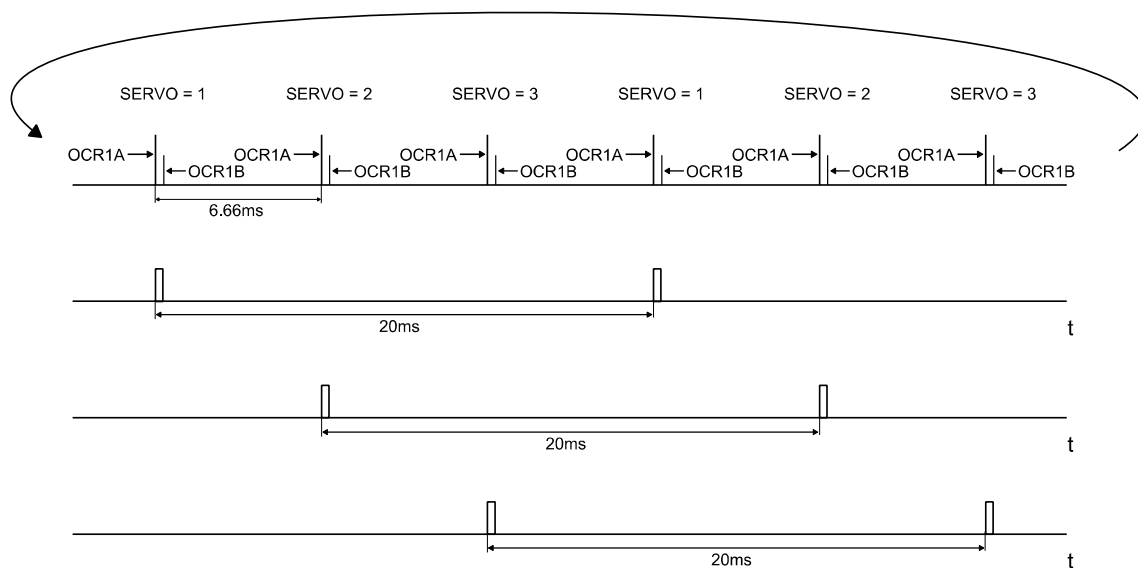
Program v mikrokontroléru implementuje tři základní věci. Ovládání serv, čtení stavu koncového dorazu a komunikaci s počítačem. V následujících kapitolách budou popsány principy jednotlivých částí programu.

### 5.1 Ovládání modelářských serv

Jak již bylo zmíněno v kapitole 4.3.2, modelářské servo se ovládá pulzy generovanými s frekvencí 50Hz. Délka pulzu je od 600us do 2400us. Pro ovládání je použit 16-ti bitový čítač/časovač1, který je nastaven do CTC modu. V tomto modu jsou k dispozici dvě komparační jednotky s čítajícími registry OCR1A a OCR1B.

Pro ovládání tří serv je časový úsek 20ms rozdělen na tři díly. Komparační jednotka OCR1A je tedy nastavena tak, aby generovala přerušení každých 6.666ms. Po dosažení této hodnoty a vyvolání přerušení je čítací registr TCNT1 vynulován. V programu je vytvořena globální proměnná *SERVO* a pomocí konstrukce *switch case* je vybíráno servo, které bude obsluhováno. Tato proměnná je v každém přerušení od komparační jednotky OCR1A inkrementována a testována tak, aby nabývala pouze hodnot 0, 1, 2. Tímto je docíleno toho, že každé servo je obsluhováno s frekvencí 50Hz.

Při vyvolání přerušení od komparační jednotky OCR1A je vybrán pin příslušného serva a nastaven do logické „1“. Dále je do registru OCR1B vložena hodnota, která odpovídá délce pulzu v logické „1“. Následující přerušení je vyvoláno komparační jednotkou OCR1B, ve kterém je příslušný pin nastaven do logické „0“ a do registru OCR1B vložena hodnota 0xffff, aby bylo zabráněno této jednotce generovat přerušení dříve než komparační jednotce OCR1A. Přerušení od komparační jednotky OCR1B nenuluje čítací registr TCNT1. Celý princip je zobrazen na *obrázku 17*.



Obrázek 17: Časový diagram ovládání serv

### 5.1.1 Konfigurace čítače/časovače1

Analogové řídicí obvody serva nepracují s nekonečnou přesností, z toho důvodu je u serv udáváno tzv. pásmo necitlivosti. Tato hodnota udává, jaká nejmenší změna vstupního signálu vede ke změně polohy páky serva. U standardních serv se pohybuje okolo 8 $\mu$ s. Pro natočení serva od 0° do 180° odpovídá změně pulzu o 1800 $\mu$ s. Po uvážení pásma necitlivosti vychází natočení páky serva od 0° do 180° na 225 kroků. Pro uložení a přenos polohy jednoho serva stačí tedy jeden byte.

Mikrokontrolér je osazen 16MHz krystalem pro přesné a stabilní časování. Jeden takt trvá 62.5ns. Dělička čítače je nastavena na dělicí poměr  $N = 64$ . Hodnota registru OCR1A je vypočtena ze vzorce 13 a odpovídá hodnotě 0x683 pro  $t_{per} = 6.666$ ms. Hodnota registru OCR1B se mění v závislosti na požadovaném natočení serva. Nejmenší krok serva odpovídá změně pulzu o 8 $\mu$ s a jeden takt čítače trvá 4 $\mu$ s. Byte, který reprezentuje polohu natočení serva, musí být tedy vynásoben konstantou 2. Vynásobení konstantou 2 je provedeno rotací doleva, což je velice rychlé a efektivní. Jelikož takto vypočtený čas je pouze čas z intervalu 0 $\mu$ s až 1800 $\mu$ s, musí k němu být přičtena konstanta 0x96 odpovídající času 600 $\mu$ s a poloze páky serva 0°. Tato hodnota je při přerušení od komparační jednotky OCR1A vložena do registru OCR1B.

$$OCR1A = \frac{t_{per} \cdot f_{osc}}{N} - 1 \quad (13)$$

## 5.2 Komunikace

Přenos dat mezi osobním počítačem a mikrokontrolérem probíhá po sérové lince v asynchronním režimu s následujícími parametry: přenosová rychlost 9600Bd, 8 datových bitů, jeden stopbit a bez hardwarového řízení toku. Komunikace je založena na paketech přesně definovaného tvaru.

### 5.2.1 Konfigurace USART jednotky

Modulační rychlost se odvíjí od systémových hodin, které jsou dány použitým krystalem. Aby bylo možné modulační rychlost volit dle požadavku aplikace, obsahuje USART jednotka programovatelný dělič kmitočtu. Tento dělič kmitočtu pracuje s dvojicí 8-bitových registrů UBRRL a UBRRH. Z horního registru UBRRH jsou použity pouze čtyři bity, proto je možné volit dělicí poměr od 0 do 4095. Dále jsou za programovatelnou děličkou kmitočtu v kaskádě dvě děličky kmitočtu s pevným dělicím poměrem  $N = 2$  a jedna s poměrem  $N = 4$ . Tyto tři děličky jsou ve *vzorci 14* reprezentovány konstantou 16 ve jmenovateli. Pro výpočet dělicího poměru programovatelné děličky kmitočtu je použit *vzorec 14*, kde BAUD udává požadovanou modulační rychlost v baudech. UBRR je souhrnný název pro registry UBRRL a UBRRH. Hodnota UBRR pro  $f_{osc} = 16\text{MHz}$  a  $\text{BAUD} = 9600\text{Bd}$  vychází  $0x67$ .

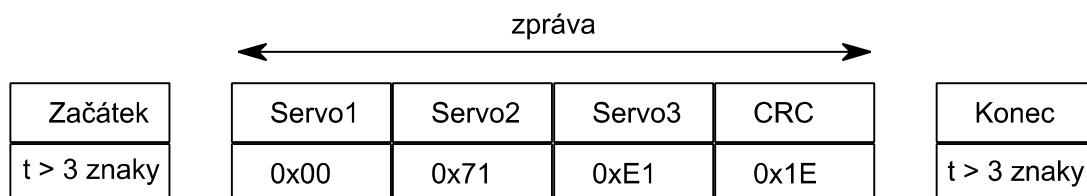
$$UBRR = \frac{f_{osc}}{16 \cdot \text{BAUD}} - 1 \quad (14)$$

### 5.2.2 Komunikační protokol

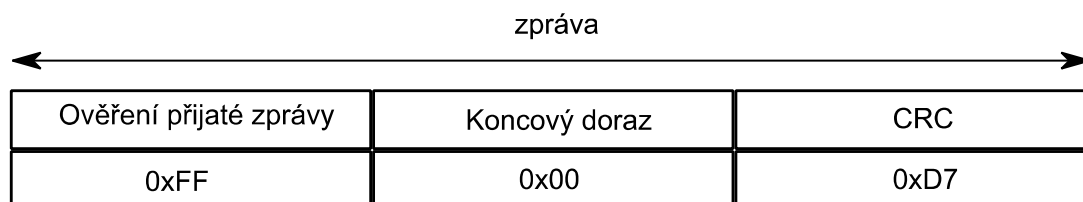
Komunikace je vždy zahájena ze strany počítače, který začíná vysílat zprávu s následující strukturou. První tři byty obsahují polohy jednotlivých serv a čtvrtý byte obsahuje kontrolní součet CRC počítaný přes první tři byty. Vysílání celé zprávy musí být souvislé a konec zprávy je identifikován jako pomlka na sériové lince delší než tři znaky. Po přijetí zprávy řídicí jednotka ověří kontrolní součet CRC a odešle zprávu o třech bytech. První byte indikuje, zda kontrolní součet byl v pořádku. Pokud ano, je první byte zprávy  $0xff$ , pokud kontrolní součet nesouhlasí, je první byte zprávy  $0x00$ . Druhý byte zprávy je stav koncového dorazu. Sepnutý spínač odpovídá hodnotě  $0xff$  a rozepnutý  $0x00$ . Poslední byte je opět kontrolní součet CRC počítaný přes první dva byty.

Jelikož v počítači není možné zajistit detekci konce zprávy jako pomlku na sériové lince, je konec zprávy detekován počtem přenesených znaků, který je pevně dán. Pokud přijatá

zpráva neobsahuje tři znaky, nebo první byte zprávy indikuje chybu v přenosu, je opět vyslána původní zpráva a čeká se na novou odpověď od řídicí jednotky. Stejně tak se opakuje přenos, pokud nesouhlasí kontrolní součet přijaté zprávy. *Obrázek 18* znázorňuje zprávu vyslanou z počítače s požadavkem na natočení prvního serva na 0°, druhého na 90° a třetího na 180°. Odpověď od řídicí jednotky je zobrazena na *obrázku 19* pro případ přenesené zprávy bez chyb a rozepnutým koncovým dorazem.



*Obrázek 18: Zpráva odeslaná z počítače s reálnými daty*



*Obrázek 19: Zpráva od řídicí jednotky s reálnými daty*

### 5.2.3 Implementace komunikace v mikrokontroléru

Pro implementaci komunikace v mikrokontroléru je použita USART jednotka a 8-bitový čítač/časovač0. Čítač/časovač0 se stará o to, aby byl detekován konec zprávy, který je dán pomlčkou na sérové lince delší než tři znaky. Je tedy nakonfigurován tak, aby generoval přerušení za dobu, která odpovídá přenosu tří bytu. Ze *vzorce 13* lze odvodit pro děličku  $N = 256$  a  $t_{\text{per}} = 3.125\text{ms}$  hodnotu čítacího registru jako  $\text{TCNT0} = 256 - \text{OCR1A} = 0x3e$ .

V datové paměti mikrokontroléru je definované čtyřbytové pole typu char. Toto pole slouží jako buffer pro přijatou zprávu. USART jednotka při přijetí každého bytu zprávy generuje přerušení. V přerušení je přijatý byte z registru UDR překopírován do bufferu na místo, kam ukazuje ukazatel vrcholu bufferu.

První přijatý byte zprávy je uložen na nultou pozici bufferu a ukazatel na vrchol bufferu je inkrementován. Následně se do čítacího registru TCNT0 vloží hodnota 0x3e a spustí se čítání. Běh programu se přesouvá z přerušení do hlavního programu. Mezi následujícími třemi byty zprávy nesmí být mezery delší než 3,125ms, aby nebylo vyvoláno přerušení od čítače/časovače0. Po přijetí posledního čtvrtého bytu následuje pomlka, která indikuje konec

zprávy. Během této pomlky není čítač/časovač0 přednastavován a dojde k přetečení a vyvolání přerušení. V přerušení je zastaven čítač/časovač0 a vynulován ukazatel na vrchol bufferu. Dále je ověřen kontrolní součet CRC a odmaskováno přerušení, které generuje prázdný vysílací buffer USART jednotky. Následně je sestavena zpráva a v přerušeních odvysílána do osobního počítače.

Pro zabezpečení dat je použit kontrolní součet CRC. Jako generační polynom je použit polynom CRC-8-CCITT ve tvaru  $x^8 + x^2 + x^1 + x^0$ . Počítání kontrolního součtu je možné dělením informačního slova generačním polynomem. Tento způsob je pomalý, proto byly vyvinuty rychlejší metody založené na generování pomocí tabulky. Tabulka je vytvořena tak, že je pro každý byte od 0x00 do 0xff vypočten kontrolní součet zvlášť. Celá tabulka je uložena v programové paměti FLASH a pomocí makra PROGMEM není při startupu kopírována do datové paměti.



## 6 Aplikace pro osobní počítač

Aplikace je napsána pro platformu Microsoft .NET. Z celé řady programovacích jazyků, která tato platforma nabízí, byl vybrán C#. Jako vývojové prostředí bylo zvoleno Microsoft Visual Studio 2010 ve verzi 10.0.40219.1 SP1Rel

Aplikace zajišťuje interaktivní komunikaci s uživatelem přes plochu, na kterou lze kreslit. Dále aplikace obstarává výpočet inverzní kinematické úlohy a v neposlední řadě komunikuje s řídicí jednotkou. Celá aplikace je rozdělena do tříd, kde každá třída zajišťuje jednu z těchto funkcí.

### 6.1 Kreslení

Ve formuláři je vytvořen objekt třídy *PictureBox*. Tento objekt má implementovanou událost *mouseMove*, která se volá při změně kurzoru na objektu *PictureBox*. Do obsluhy události je předáván objekt s parametry *X* a *Y*, které udávají aktuální polohu kurzoru vůči nulovému bodu *PictureBoxu*. V události je dále vytvořena instance třídy *Graphics*. Tato instance je vytvořit samotným objektem *PictureBox* tak, že je na něm volána metoda *CreateGraphics()*. Pro kreslení je třeba ještě vytvořit pero, tedy instanci třídy *Pen*, ve které je definována barva a tloušťka pera.

Následné malování na objektu *PictureBox* probíhá tak, že v obsluze události *mouseMove* je kreslena čára pomocí metody *DrawLine()* kterou implementuje třída *Graphics*. Metoda *DrawLine()* umí nakreslit čáru danou počátečním a koncovým bodem. Koncovým bod je aktuální poloha kurzoru a tedy parametry *X* a *Y* předané do obsluhy událost. Počáteční bod čáry je minulý stav kurzoru, který musí být při každé obsluze události uložen, aby mohl být v následující obsluze události použit jako počáteční bod čáry. Celá malovaná křivka se tedy skládá z objektů rovných čar, které jsou pospojovány za sebou.

### 6.2 Komunikace

Jak již bylo zmíněno výše, ke komunikaci mezi počítačem a řídicí jednotkou je použit sériový port. Běžové prostředí .NET Framework umožňuje sériovou linku ovládat skrz třídu *SerialPort*. Před prvním použitím portu je třeba nastavit základní vlastnosti, které lze předat rovnou konstruktoru nebo nastavit až na vytvořeném objektu.

Mezi tyto vlastnosti patří *PortName*. Voláním metody *GetPortNames()* lze získat všechna jména portů, které jsou momentálně dostupné. Metoda vrazí jména v poli řetězců. Pro výběr

z dostupných portů je použit ovládací prvek *comboBox*. Další důležitý parametr je modulační rychlost, kterou reprezentuje vlastnost *BaudRate* a je nastavena na 9600Bd. Následující parametr je počet datových bitů daný vlastností *DataBits* nastavený na 8. Předposlední vlastností je počet stopbitů a posledním parametrem je parita, která není využita, proto vlastnost *Parity* je nastavena na *none*.

Po nakonfigurování je možné sériovou linku otevřít metodou *Open()*. Tato metoda může vyhazovat výjimky, například když daný port je zabrán jiným procesem. Tyto výjimky je třeba ošetřit, v tomto případě vrátit se k možnosti výběru jiného portu.

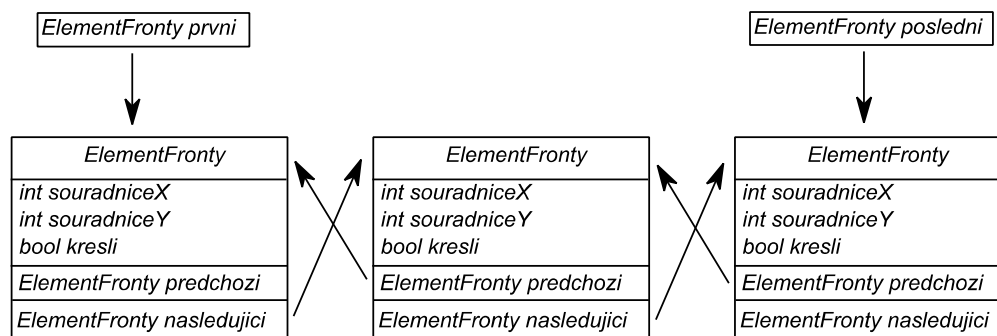
Po úspěšném otevření portu je možné volat metody pro odesílání celých polí jako je metoda *Write()*, která očekává jako parametry: pole odesílaných bytů, od jaké pozice v poli má začít odesílání a jakou pozicí má odesílání skončit. Metoda pro čtení celého pole je *Read()*, význam parametru je shodný s metodou *Write()*. Dále v programu je použita vlastnost *BytesToRead*, která získá počet bytů v přijímacím bufferu. Tato vlastnost je použita pro příjem zprávy od řídicí jednotky, která má přesně definovaný počet bytů.

Program je rozdělen do dvou vláken. Primární vlákno je vytvořeno běhovým prostředím a stará se o ovládání aplikace a kreslení. Druhé pracovní vlákno je vytvořeno za účelem komunikace s řídicí jednotkou. Při kreslení jsou jednotlivé body trajektorie vykreslovány na *pictureBox*, ale také jsou ukládány do struktury, která implementuje paměť typu FIFO. Toto vše se děje v primárním vláknu. Z této paměti jsou následně jednotlivé body trajektorie vybírány pracovním vlákem. Pracovní vlákno následně z těchto bodů vypočte inverzní úlohou kinematiky, sestaví zprávu a odešle řídicí jednotce.

### 6.2.1 Paměť typu FIFO

Tato paměť slouží k oddělení rychlého procesu kreslení od pomalejšího procesu posílání dat řídicí jednotce a následné změně poloh akčních členů. Paměť je tvořena objekty třídy *ElementFronty*, které mají soukromé datové členy udávající polohu kurzoru na *pictureBoxu*, booleovskou proměnnou *kresli* udávající, zda má být pero položené na papíře a dva ukazatele na tentýž objekt, díky kterým je možné objekty řetězit.

Řetězení a samotnou logiku vkládání a vybírání dat implementuje třída *Fronta*, která nabízí funkce *vloz()* pro vložení dat, funkci *vyber()* pro výběr dat a funkci *jePrazdna()* pro zjištění, zda je už paměť prázdná. Dále jsou v této třídě ukazatele na začátek a konec fronty. Celá struktura této paměti je zobrazena na *obrázku 20*, kde je znázorněna fronta o třech prvcích.



Obrázek 20: Implementace struktury FIFO

Vkládání a vybírání dat probíhá ve dvou různých vláknech, to může způsobit problém, pokud ve stejnou dobu budou chtít obě vlákna pracovat s proměnnými ve frontě. Řešení tohoto problému je v tzv. zámčích (locks). Pokud jedno vlákno operuje s proměnnými, zamkne si je pro svoji potřebu a až práci dokončí, opět je odemkne. Pokud chce druhé vlákno pracovat s proměnnými, které jsou dočasně zamknuty, musí počkat, dokud je první vlákno neodemkne. Této konstrukci se říká „thread safe“.

### 6.3 Matematický model

Třída *MatematickyModel* implementuje výpočet inverzní úlohy kinematiky. Objekt této třídy je vytvářen přes konstruktor, který očekává parametry reálného modelu jako jsou: délka ramen, délka vzpěry, vzdálenost středu platformy do kloubů na platformě a vzdálenost osy serva od středu základny.

Tato třída obsahuje pouze jednu metodu, kterou je metoda *vypocti()*. Tato metoda očekává tři parametry, kterými jsou požadované souřadnice efektoru v prostoru. Metoda provede výpočet inverzní úlohy a vrátí pole o třech prvcích, které odpovídají natočení jednotlivých serv.

## 7 Konstrukce

Základna robota je vyrobena z polykarbonátové desky o tloušťce 4mm, která je vyříznuta do tvaru šestiúhelníku. Jednotlivá serva jsou usazena na této základně a vůči sobě jsou pootočená o 120°. Jako ramena jsou použity hliníkové profily typu U o rozměrech 7x7x1mm a délce 60mm. Vzpěry jsou udělány ze závitové tyče M3 o délce 150mm. Platforma je ze stejného materiálu jako základna vyříznuta do tvaru rovnostranného trojúhelníka o délce strany 40mm. Jako klouby jsou použity kulové čepy MP JET MODEL PARTS V1, pr.7, M3.

Kreslicí ustrojí je tvořeno dutou trubičkou, ve které je vsazen fix, který se může pohybovat dolu a nahoru. Na konci trubičky je přilepen koncový spínač P-DM03S1PH. Trubička je podélně naříznuta oválnými otvory. Fix je provrtán a do otvoru vsazen tvarovaný drát, na kterém jsou zavěšeny pružiny, které táhnou fix směrem dolu v oválných otvorech trubičky. Celá tato konstrukce je přidělána na platformu.

Delta robot je za platformu uchycen do klece pomocí tří distančních sloupků DI5M3X40. Klec tvoří kvádr, kde horní strana je opět z polykarbonátové desky, která má po obvodu přišroubované hliníkové profily tvaru L o rozměrech 20x20x2mm a délce 280mm. V rozích jsou přišroubovány hliníkové profily tvaru L o rozměrech 15x15x1,5mm a délce 300mm. Tyto profily tvoří hrany kvádrů a zároveň nohy celé klece. Na těchto nohou jsou ve výšce 20mm nad zemí přišroubovány stejné profily, ze kterých jsou nohy a tvoří první patro společně s dřevotřískovou deskou.

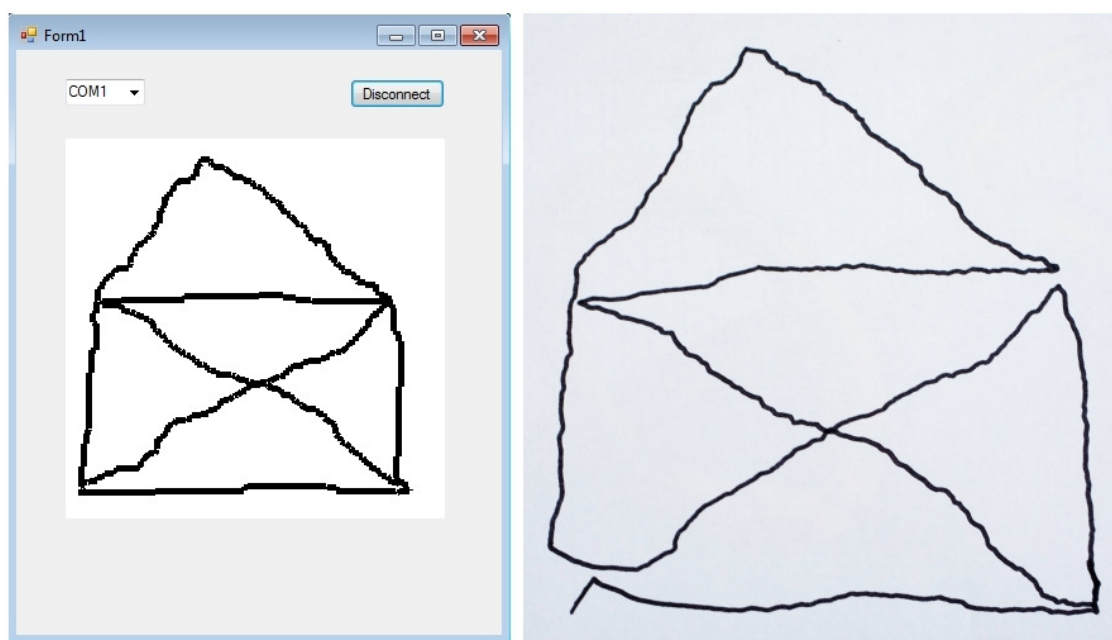
Na polykarbonátové desce, která tvoří vršek klece, je z horní strany přišroubována řídicí jednotka pomocí čtyř distančních sloupků DI5M3X10. Reálná konstrukce je na obrázku v příloze D

## 8 Závěr

Cílem této bakalářské práce bylo navrhnout, zkonstruovat a naprogramovat malý model delta robotu ovládaného z počítače. Pro demonstraci funkčnosti byla vybrána jednoduchá úloha kreslení, která plně ověřila správnost výpočtu kinematické úlohy.

Mechanická konstrukce celého modelu je z dílů, které nemají ideální parametry, a proto se celá konstrukce potýká s pružením a vůlemi. Serva, která jsou použita z finančních důvodů, nemají dostatečnou sílu a vůle, kterými trpí, snižují přesnost celého robota. Z provedených pokusů lze odhadnout přesnost dosažené polohy na  $\pm 5\text{mm}$ . Na *obrázku 21* jsou k porovnání nakresleny dva motivy: vlevo motiv nakreslený v aplikaci na počítači, vpravo reálná kresba provedená robotem.

V kapitole o řídicí jednotce je několikrát zmínka o rozšíření modelu. Už během návrhu současného modelu byl kladen důraz na budoucí rozšíření modelu o pásový dopravník poháněný krokovým motorem. Z tohoto důvodu je na plošném spoji řídicí jednotky tranzistorové pole ULN2803A, které bude sloužit jako budič krokového motoru. Další rozšíření bude o jednotku schopnou rozpoznávat barvy předmětu na pásovém dopravníku. Delta robot bude poté předměty z pásového dopravníku sbírat a podle barvy řadit do kontejnerů. Ke sbírání předmětů z pásu bude sloužit přísavka přes elektricky ovládaný ventil připojená na vývěvu. Pro ovládání ventilu je na řídicí jednotce darlingtonův tranzistor BD679.



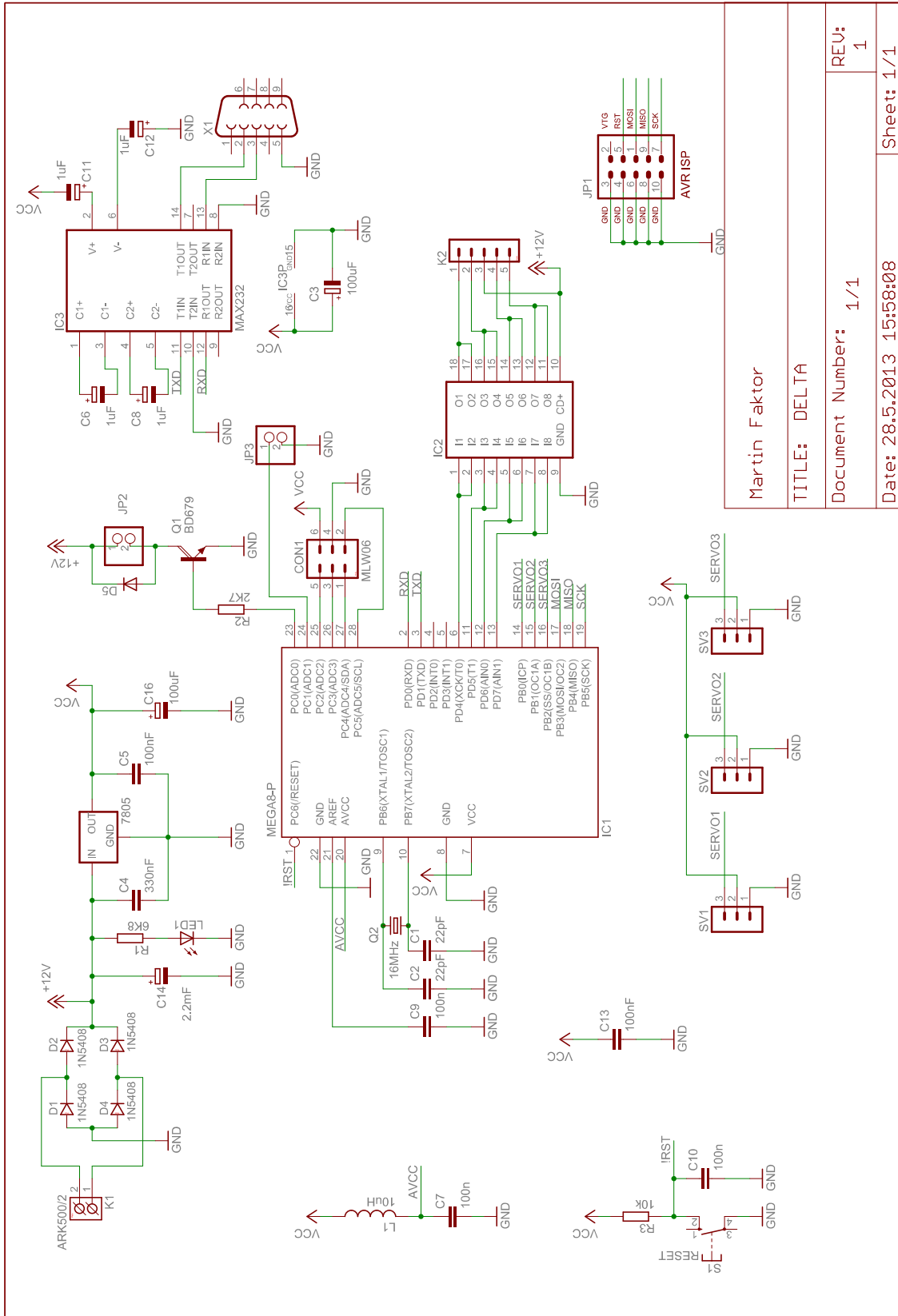
Obrázek 21: Vlevo obrázek namalovaný na počítači, vpravo reálný obrázek namalovaný delta robotem

## Seznam použité literatury

- [1] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR: [měření, řízení a regulace pomocí několika jednoduchých přípravků]*. 1. vyd. Praha: BEN - technická literatura, 2006, 319 s. ISBN 80-730-0174-8.
- [2] PINKER, Jiří. *Mikroprocesory a mikropočítače*. 1. vyd., 1. dot. Praha: BEN - technická literatura, 2008, 159 s. ISBN 978-80-7300-110-0.
- [3] VÁŇA, Vladimír. *Mikrokontroléry ATMEL AVR: programování v jazyce C : popis a práce ve vývojovém prostředí CodeVisionAVR C*. 1. vyd. Praha: BEN - technická literatura, 2003, 215 s. ISBN 80-730-0102-0.
- [4] MANN, Burkhard. *C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky*. Vyd. 1. Praha: BEN, 2003, 279 s. ISBN 80-730-0077-6.
- [5] HEROUT, Pavel. *Učebnice jazyka C*. 6. vyd. České Budějovice: Kopp, 2009, 271, viii s. ISBN 978-80-7232-383-8.
- [6] Full flight simulator. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-06-03]. Dostupné z: [http://en.wikipedia.org/wiki/Full\\_flight\\_simulator](http://en.wikipedia.org/wiki/Full_flight_simulator)
- [7] File:TOSY Parallel Robot.JPG - Wikipedia, the free encyclopedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-06-04]. Dostupné z: [http://en.wikipedia.org/wiki/File:TOSY\\_Parallel\\_Robot.JPG](http://en.wikipedia.org/wiki/File:TOSY_Parallel_Robot.JPG)
- [8] Delta robot. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-28]. Dostupné z: [http://en.wikipedia.org/wiki/Delta\\_robot](http://en.wikipedia.org/wiki/Delta_robot)
- [9] Parallel robot provided with wrist section having three degrees of freedom - Patent #08109173. *Patent Information Database | Free Patent Search Online* [online]. 1999 [cit. 2013-06-04]. Dostupné z: <http://www.strutpatent.com/patent/08109173/parallel-robot-provided-with-wrist-section-having-three-degrees-of-freedom>
- [10] MAXIM INTEGRATED. *MAX220–MAX249*. 19-4323; Rev 16;. 160 Rio Robles, San Jose, CA 95134 USA, 2010. Dostupné z: <http://datasheets.maximintegrated.com/en/ds/MAX220-MAX249.pdf>

- [11] I-tec USB to Serial RS232 Adapter. *I-TEC - vše, co můžete k počítači potřebovat* [online]. [cit. 2013-05-28]. Dostupné z: <http://www.i-tec.cz/?t=3&v=31>
- [12] Serva. *RC.PELIKANDANIEL.COM* [online]. [cit. 2013-05-28]. Dostupné z: <http://rc.pelikandaniel.com/?sec=page&id=22>
- [13] Atmel Studio. *Atmel Corporation - Microcontrollers, 32-bit, and touch solutions* [online]. [cit. 2013-05-28]. Dostupné z: <http://www.atmel.com/tools/ATMELSTUDIO.aspx>
- [14] ABB IRB 6620. *Used Robots - Motoman, Fanuc, Other Used Industrial Robots* [online]. [cit. 2013-05-28]. Dostupné z: <http://www.used-robots.com/used-abb-robots.php?robot=irb+6620>
- [15] ABB Secures Agreement to Provide Robotic Automation to A123 Systems for Advanced Ba. *ThomasNet News - The complete source for the latest industrial news* [online]. [cit. 2013-05-28]. Dostupné z: <http://news.thomasnet.com/companystory/ABB-Secures-Agreement-to-Provide-Robotic-Automation-to-A123-Systems-for-Advanced-Battery-Applications-831668>
- [16] Geometrie/Numerický výpočet průniku dvou kružnic. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-28]. Dostupné z: [http://cs.wikibooks.org/wiki/Geometrie/Numerický\\_výpočet\\_průniku\\_dvou\\_kružnic](http://cs.wikibooks.org/wiki/Geometrie/Numerický_výpočet_průniku_dvou_kružnic)
- [17] Řízení serva - teorie :: Serva a jejich ovládání. *Serva a jejich ovládání* [online]. [cit. 2013-05-28]. Dostupné z: <http://www.serva.cz/rizeni-serva-teorie/>
- [18] ATMEL CORPORATION. *8-bit Atmel Microcontroller with 8KB In-System Programmable Flash*. Rev. 2486M–AVR–12/03. San Jose, CA 95131, USA, 2003. Dostupné z: [http://www.atmel.com/Images/Atmel-8159-8-bit-AVR-microcontroller-ATmega8A\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-8159-8-bit-AVR-microcontroller-ATmega8A_datasheet.pdf)
- [19] Descriptive Geometric Kinematic Analysis of Clavel's "Delta" Robot. April 1, 2004, s. 15. Dostupné z: <http://www.cim.mcgill.ca/~paul/clavdelt.pdf>

# Příloha A – Schéma řídicí jednotky



Martin Faktor

TITLE: DELTA

Document Number: 1/1

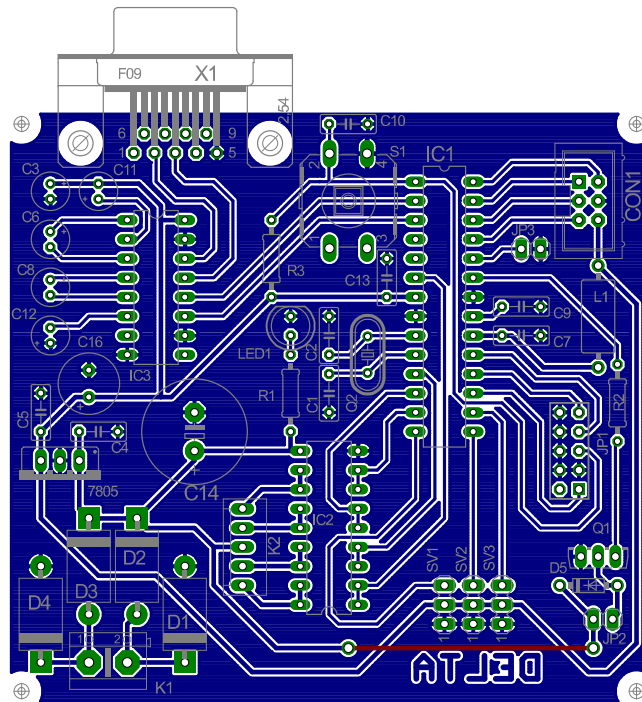
Date: 28.5.2013 15:58:08

REV: 1

Sheet: 1/1



## Příloha B – Osazovací výkres řídicí jednotky



## Příloha C – Partlist

Part	Value	Device	Package
7805		78XXS	78XXS
C1	22pF	C-EU050-025X075	C050-025X075
C2	22pF	C-EU050-025X075	C050-025X075
C3	100uF	CPOL-EUE2-5	E2-5
C4	330nF	C-EU050-025X075	C050-025X075
C5	100nF	C-EU050-025X075	C050-025X075
C6	1uF	CPOL-EUE2-5	E2-5
C7	100n	C-EU050-025X075	C050-025X075
C8	1uF	CPOL-EUE2-5	E2-5
C9	100n	C-EU050-025X075	C050-025X075
C10	100n	C-EU050-025X075	C050-025X075
C11	1uF	CPOL-EUE2-5	E2-5
C12	1uF	CPOL-EUE2-5	E2-5
C13	100nF	C-EU050-025X075	C050-025X075
C14	2.2mF	CPOL-EUE5-13	E5-13
C16	100uF	CPOL-EUE3.5-8	E3,5-8
CON1	MLW06	MLW06	MLW06G
D1	1N5408	1N5408	DO201
D2	1N5408	1N5408	DO201
D3	1N5408	1N5408	DO201
D4	1N5408	1N5408	DO201
D5		DIODE-DO41-7.6	DO41-7.6
IC1	MEGA8-P	MEGA8-P	DIL28-3
IC2		ULN2803A	DIL18
IC3	MAX232	MAX232	DIL16
JP1	AVR-ISP-10	AVR-ISP-10	AVR-ISP-10
JP2		PINHD-1X2	1X02
JP3		PINHD-1X2	1X02
K1	ARK500/2	ARK500/2	ARK500/2
K2		PSH02-05P	PSH02-05P
L1	10uH	L-USIR-4	IR-4
1.01		LED5MM	LED5MM
Q1	BD679	BD679	TO126AV
Q2	16MHz	CRYSTALHC49S	HC49/S
R1	6K8	R-EU_0207/10	0207/10
R2	2K7	R-EU_0207/10	0207/10
R3	10k	R-EU_0207/10	0207/10
S1	RESET	RACON12	RACON12
SV1		MA03-1	MA03-1
SV2		MA03-1	MA03-1

SV3		MA03-1	MA03-1
X1		F09H	F09H

## Příloha D – Fotografie modelu

