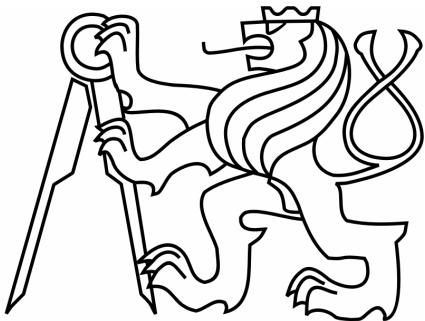


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

Model obráběcího stroje

2007

ALEŠ ZIKMUND

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne
.....
podpis

Poděkování

Rád bych poděkoval vedoucímu své bakalářské práce panu Doc. Ing. Jiřímu Bayerovi, CSc. za důležité připomínky.

Abstrakt

Práce se zabývá propojením modelu tříosého soustruhu s řídicí kartou pomocí níž je řízen. Čtenář se zde seznámí s kompletním popisem modelu, návrhem elektroniky a také s možností programování řídicí karty. Uvedeno je hardwarové a softwarové nastavení celého systému a popis navrženého software.

Abstract:

The work deals with connection of three-axis lathe model and control system. The reader will get acquainted with whole description of model, designed electronics and programming control system. The work contains software and hardware setting system and designed software description.

Katedra řídicí techniky

Školní rok: 2006/2007

Zadání bakalářské práce

Student: Aleš Zikmund

Obor: Kybernetika a měření

Název tématu: Model obráběcího stroje

Zásady pro výpracování:

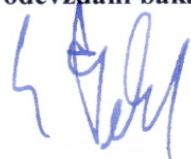
1. Seznamte se se stávajícím modelem obráběcího stroje, ověřte ovládání jeho jednotlivých funkcí, stanovte požadavky na interface.
2. Realizujte propojení modelu stroje s "motion controllerem" PMAC-PC fy Delta Tau s DSP Hitachi HD63140. Ověřte ovládání základních funkcí modelu stroje propojením PMAC-PC Delta Tau s PC prostřednictvím sériové linky RS232.
3. Vypracujte úplnou dokumentaci obvodů obráběcího stroje a jeho propojení s řídicím systémem, interface realizujte na plošném spoji.
4. Pro demonstrační účely vytvořte počítačový model průběhu obrábění na PC, který poběží paralelně s reálným řízením modelu stroje.
5. Výslednou činnost modelu stroje ověřte na vzorovém programu obrábění. Pro výukové účely vypracujte návod na použití modelu obráběcího stroje.

Seznam odborné literatury: Dodá vedoucí práce

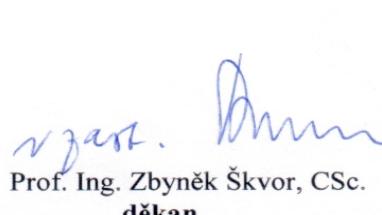
Vedoucí bakalářské práce: Doc. Ing. Jiří Bayer, CSc.

Datum zadání bakalářské práce: zimní semestr 2006/07

Termín odevzdání bakalářské práce: 15. 8. 2007



Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



v jaz. Prof. Ing. Zbyněk Škvor, CSc.
děkan

Obsah

Seznam obrázků	vii
Seznam tabulek	viii
1. Úvod	1
2. Popis modelu obráběcího stroje	2
2.1. Motorky.....	2
2.2. Snímače polohy.....	4
2.3. Koncové snímače	4
3. ISA karta PMAC Lite „motion controller“	5
3.1. Stručný popis karty	5
3.2. Nastavení karty	6
3.3. Vnitřní proměnné PMACu.....	7
3.4. Příkazy	7
3.5. Operátory a funkce.....	8
3.6. Pohybový program (motion program)	8
3.7. Psaní motion programů	11
3.8. PLC programy.....	11
3.9. Psaní PLC programů	11
3.10. Program Pewin32 Pro2	12
3.11. Nastavení karty pro model obráběcího stroje	14
4. Návrh servozesilovače motorků	16
5. Návrh interface pro propojení PMACu s modelem soustruhu	22
5.1. Požadavky	22
5.2. Připojení snímačů polohy	23
5.3. Připojení koncových snímačů	23
5.4. Zapojení povolovacího signálu AENA (amplifier enable)	24
6. Počítacový model obráběcího stroje	24
7. Návod na použití modelu soustruhu	26
7.1. Pomůcky	26
7.2. Spouštění a navázání komunikace	26
7.3. Nastavení PID regulátoru.....	27
7.4. Psaní motion a PLC programů v prostředí PeWin.....	27
8. Počítacový model obráběcího stroje	28
9. Závěr	31
Literatura.....	32
Příloha A. 3D Model obráběcího soustruhu	I
Příloha B. Rozmístění spojek na desce karty PMAC.....	II
Příloha C. Tabulky programových příkazů	III
Příloha D. Schéma zapojení plošného spoje	IV
Příloha E. Plošný spoj	V
Příloha F. Osazení součástek plošného spoje	VI
Příloha G. Zapojení svorkovnic a spojek plošného spoje	VII
Příloha H. Výsledný tvar obrobku z příkladu.....	VIII
Příloha I. Obsah přiloženého CD	IX

Seznam obrázků

2.1 Obvodové schéma motoru s cizím buzením	2
2.2 Blokové schéma stejnosměrného motoru řízeného napětím.....	3
2.3 Princip funkce snímače a průběh výstupů	4
2.4 Zapojení optického snímače polohy SFH 910	4
2.7 Obvodové schéma snímače TCST 1130	4
3.1 Řídicí karta PMAC Lite	5
3.2 Princip mapování vstupů a výstupů do procesoru.	5
3.3 Výpočetní náročnost jednotlivých skupin úloh.....	6
3.4 Příklad lineárních interpolací a)nenavazující b) navazující.....	10
3.5 Příklady kruhových interpolací.....	10
3.6 Vývojové prostředí Pewin32 Pro2	13
3.7 Dialogové okno pro navázání komunikace s PMACem	13
3.8 Okno pro výběr PMACu.....	13
3.9 Okno pro nastavení sériové linky RS232.....	14
3.10 Regulační smyčka PMACu pro jednotlivý motor.....	16
4.1 Princip funkce PWM modulace a H můstek.....	17
4.2 Vnitřní obvodové schéma analogového obvodu L292	17
4.3 Blokové schéma obvodu L292.....	18
4.4 Zapojení obvodu L292 pro malé výkony	20
4.5 Zapojení obvodu s rozšiřujícím H můstkem pro větší proudy.....	20
5.1 Konektor JMACH1	22
5.2 Zapojení optického snímače TCST 1103.....	23
5.3 Zapojení pull up odpor pro AENA	24
6.1 Hlavní okno software	25
6.2 Okno pro nastavení komunikace.....	25
6.3 Okno pro nastavení parametrů obrábění	26
7.1 Terminál pro zadávání příkazů	27
7.2 Zobrazení pozic motorů	27
7.3 Textový editor	28

Seznam tabulek

3.1 Tabulka nastavení jumperů přenosové rychlosti sériové linky.....	15
3.2 Tabulka změn hlavních I-proměnných	15
3.3 Tabulka změn I- proměnných odpovídajících motorům.....	16
3.4 Tabulka změn I- proměnných odpovídajících encodérům.....	16
3.5 Tabulka změn I-proměnných odpovídajících souřadnicovým systémům	16
5.1 Tabulka seznamu signálů pro spojení s PMACem (konektor JMACH1).....	22
5.2 Tabulka důležitých parametrů snímače TCST1103.....	23

1. Úvod

Cílem práce je vytvořit model obráběcího stroje, který by odpovídal reálnému obráběcímu stroji. Tento model představuje soustruh. Soustruh je stroj, na kterém se obrábějí obrobky dle požadovaného tvaru kolem jeho rotační osy.

V základním provedení soustruhu se používají tři osy. Jedna osa specifikuje otáčení obrobku (vřeteno - osa Z) a další dvě, kolmé k této ose, vytvářejí tvar obrobku (posuvy - osa X a Y). Osa Z je převážně nezávislá na ostatních, ale osy X a Y se mohou pohybovat v určité závislosti, aby byly schopny realizovat různé druhy interpolací (např lineární, spline, kruhovou apod.), kde musí být zaručen synchronní pohyb obou os X a Y.

V úvodní kapitole je popsán původní model. Popis je zaměřen na motory, snímače polohy a koncové snímače.

Druhá kapitola se zabývá popisem řídicí karty PMAC Lite motion controller použité pro řízení modelu. Jsou zde uvedeny její vlastnosti a možností interpolací, dále obsahuje postup nastavování karty pohybových a PLC programů a také popis software k řídicí kartě.

Třetí kapitola je věnována návrhu a realizaci servozesilovače motorů, jejichž prostřednictvím se budou řídit motory jednotlivých os. V této části bude také uveden návrh a realizace interface, pomocí jehož se bude soustruh připojovat k PMACu.

Dále je popsán princip funkce a použití navrženého software pro sledování průběhu obrábění.

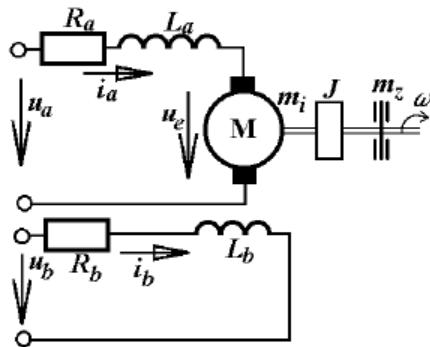
V posledních kapitolách je uveden příklad obrábění s návrhem řešení a popisem programování také návod, jak celý model používat.

2. Popis modelu obráběcího stroje

Model obráběcího stroje je odvozen od výrobku rakouské firmy EMCO, která vyrábí obráběcí stroje. Původně byl tento model řízen pomocí mechanického ovládání jednotlivých posuvů a vřeteno poháněno motorem bez řízení otáček.

2.1. Motorky

Aby byl model kompletně elektronicky řiditelný, jsou posuvy modelu doplněny o stejnosměrné motory. Moment motoru je převáděn na posuv soustruhu pomocí převodovky, která je zpřevodována na velmi malé otáčky a velký moment. Všechny motorky včetně motorku pohánějícího vřeteno modelu jsou stejnosměrné s cizím buzením, kde cizí buzení zaručuje permanentní magnet. Schéma motoru s cizím buzením je na obrázku 2.1.



Obrázek 2.1 Obvodové schéma motoru s cizím buzením

Tyto motory lze popsát jednoduchými diferenciálními rovnicemi.

$$u_a = R_a i_a + L_a \frac{di_a}{dt} + u_e \quad (1) \quad u_e = kf w \quad (2) \quad u_e = kf(i_b)w \quad (3)$$

V nich je u_a napájecí napětí kotvy a u_e je napětí, které se indukuje v kotvě motoru při jejím otáčení v magnetickém poli. Konstanta úměrnosti k je tzv. strojová konstanta a je závislá na konstrukčním uspořádání motoru. Závislost magnetického toku Φ na proudu budicího obvodu i_b vyjadřuje magnetizační charakteristika motoru $\Phi=f(i_b)$. Pro náš případ motoru s buzením permanentním magnetem lze přibližně tuto nelineární funkci nahradit konstantou $x = kf(i_{b0})$, a pak indukované napětí u_e je přímo úměrné otáčkám motoru

$$u_e = xw. \quad (4)$$

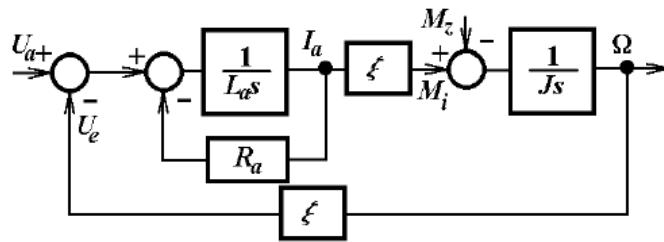
Závislost momentu vytvářeného kotvou motoru na proudu kotvy pak bude rovněž přibližně lineární se stejnou konstantou úměrnosti a pro momentovou rovnováhu na hřídeli motoru můžeme psát

$$M_i = xi_a = J \frac{dw}{dt} + M_z. \quad (5)$$

M_z je zatěžovací moment vyvolaný zátěží a pasivními odpory motoru a J zahrnuje moment setrvačnosti samotného motoru i všechny momenty setrvačnosti pohybujících se částí přepočtené na výstupní hřídel motoru. Pomocí Laplaceovy transformace můžeme nyní celý systém převést do přenosového tvaru a nakreslit blokové schéma.

$$U_a(s) = R_a I_a(s) + sL_a I_a + x \cdot \Omega(s), \quad (6)$$

$$Js\Omega(s) = x \cdot I_a(s) - M_z, \quad (7)$$



Obrázek 2.2 Blokové schéma stejnosměrného motoru řízeného napětím

Důležité pro další pozorování chování stejnosměrného motoru jsou dvě konstanty, které ovlivňují dynamické chování motoru. Jedná se o elektromechanickou konstantu τ_m a o elektrickou konstantu τ_a .

$$\tau_m = R_a \frac{J}{x^2} \quad (8) \qquad \tau_a = \frac{L_a}{R_a} \quad (9)$$

Identifikace motorků

Servozesilovač motorků, navrhovaný ve 4 kapitole, obsahuje vnitřní PI regulátor proudu a jeho návrh vychází ze znalosti konstant motoru R_a , L_a , I_M . I_M je záběrový proud, který vzniká v důsledku překonání suchého tření motorku.

Nejprve jsem změřil přechodovou charakteristiku, tj. odezvu proudu na skokovou změnu napětí kotvy, abych mohl zjistit maximální záběrový proud I_M . Jednotkový skok vstupní veličiny, tj. napětí, jsem simuloval připnutím na napětí 12V, a pak jsem odečetl hodnoty z průběhu zobrazeném na osciloskopu.

Odpor a indukčnost cívky jsem změřil RLCG můstkom.

Na modelu jsou dva typy motorků; hlavní motorek, který pohání přes řemen vřetenou soustruhu (osa Z), a motorek posunů. Tyto dva typy motorků se od sebe liší svými výkony, a proto jsem prováděl měření pro každý druh motorků zvlášť.

Naměřené konstanty motorku vřetene (osa Z):

$$L_a=180\mu H$$

$$R_a=0,5\Omega$$

$$I_M=2,5A$$

Naměřené hodnoty motorků pro ovládání posuvů os X a Y:

$$L_a=240\mu H$$

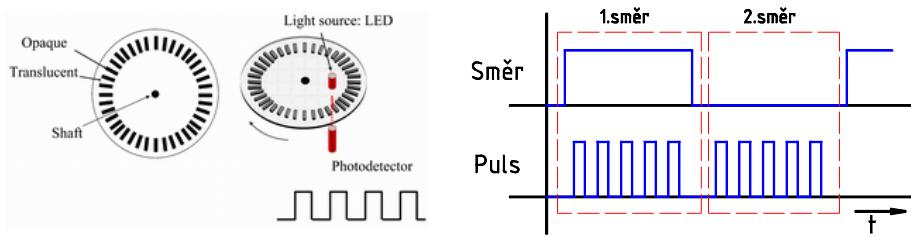
$$R_a=1\Omega$$

$$I_M=1A$$

2.2. Snímače polohy

Každá osa je osazena snímačem polohy (Encoder) SFH 110 od firmy Siemens. Jde o inkrementální optický snímač. Tento snímač polohy nemá jako standardní snímače kvadratické výstupy A-B, ale jeho výstupy jsou přímo ve tvaru pulz a směr (princip viz. obrázek 2.3).

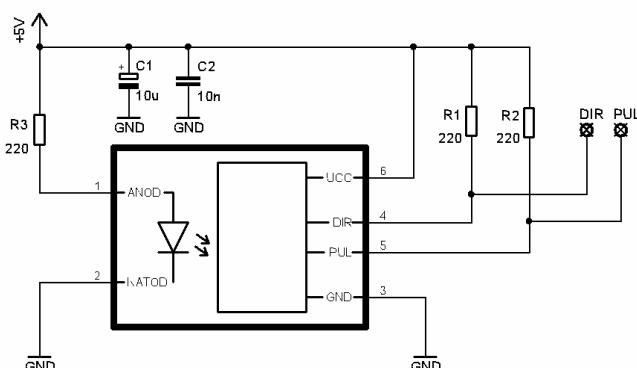
Prostředkem snímače je proložený disk, které provádí přerušování světelného toku zářiče. Disk je spojen s hřídelí motoru (shaft). Světlo vysílané světelným zdrojem (diodou LED) může procházet diskem pouze průhlednými místy (Translucent), zatímco zbývající část disku je neprůhledná (Opaque). Světelné impulsy detekované světelně citlivým detektorem (Photodetector) se převádějí na elektrické.



Obrázek 2.3 Princip funkce snímače a průběh výstupů

Snímač je napájen ze zdroje napětí 5V na svých výstupech potřebuje pull up odpory. Zapojení snímače bylo převzato z původního modelu viz obrázek 2.4Obrázek 2.4.

Měřením jsem zjistil, že snímač má 95 impulsů na otáčku.



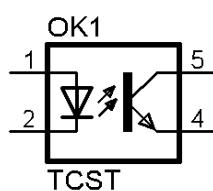
Obrázek 2.4 Zapojení optického snímače polohy SFH 910

2.3. Koncové snímače

Pohybové osy X a Y, které představují posuvy pro tvarování obrobku jsou doplněny o koncové spínače polohy (overtravel limit switch). Každá osa X a Y má dva tyto snímače; jeden pro kladný směr (+LIM) a druhý pro záporný směr (-LIM). Ty zaručují, že při pohybu posuvů nenastane stav, že by se posuvy dostaly do zakázané oblasti, kde by mohly způsobit rozbití stroje.

Koncové snímače jsem zvolil optické, optocouplery, které pracují na principu přerušování světelného toku vyzářeného LED diodou. Plechový jazýček indikující polohu posuvu přeruší světlo, které vyhodnocuje fototranzistor.

Návrh součástek k snímači je uveden v kapitole 5.



Obrázek 2.5 Obvodové schéma snímače TCST 1130

Celý 3D model obráběcího soustruhu s umístěním součástek je obsažen v příloze A.

3. ISA karta PMAC Lite „motion controller“

3.1. Stručný popis karty

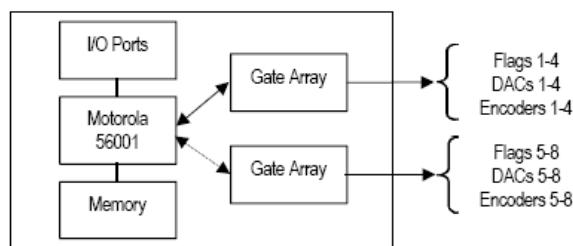
Zkratka PMAC znamená Programmable Mult-Axis Controller od firmy Delta Tau (dále jen PMAC) a jde o řídicí kartu, kterou lze řídit pohyb připojených polohových servomechanismů ve více osách. Konkrétní provedení Lite určuje počet os, možnost připojení k PC i další parametry.



Obrázek 3.1 Řídicí karta PMAC Lite.

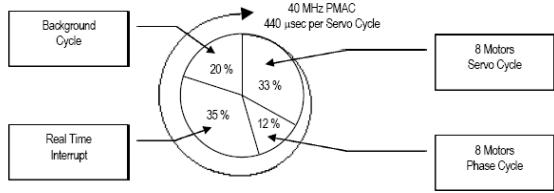
Karta PMAC Lite 40MHz Opt4A použitá v této aplikaci je osazena Motorola DSP Hitachi 56k procesorem umožňujícím přesnou kontrolu nad řízením více os. Lite verze disponuje 8-mi osami. Tato karta může pracovat s nadřazeným PC po ISA sběrnici nebo pracuje autonomně po naprogramování přes sériovou linku RS232.

Všechny registry v PMAC jsou mapovány do paměťového prostoru DSP procesoru, což umožňuje jednoduchý přístup k vstupům i výstupům karty.



Obrázek 3.2 Princip mapování vstupů a výstupů do procesoru.

V kartě PMAC se vykonávají 4 druhy úloh (task). Jednotlivé druhy se od sebe liší prioritou, a tím také výpočetní náročností, která je znázorněna na obrázku 3.3.



Obrázek 3.3 Výpočetní náročnost jednotlivých skupin úloh

Největší prioritu mají Phase Cycle, ve kterých se provádí se aktualizace výstupů. Do nižší priority Servo Cycle mají aktualizace servomotoru nebo spuštění konverzační tabulky pro dekódování signálů od snímačů. Další prioritní skupinou jsou Real-Time, zde jsou spuštěny úlohy s plánováním souřadnicového systému. Nejnižší priorita Background Cycle obsahuje zbylé úlohy jako například PLC, Watchdog, obsluha nadřazeného PC atd.

3.2. Nastavení karty

Pokud se karta používá poprvé musí se nakonfigurovat pro specifickou aplikaci hardwarově i softwarově.

Hardwarové nastavení

PMAC má 120 spojek (jumper), pomocí nichž lze na kartě hardwarově nastavit určité funkce, jako například nastavení rychlosti sériové linky ad. Tyto propojky se nazývají E body (E point). Význam všech spojek viz. [8]. Rozmístění spojek na kartě je uvedeno v příloze Příloha B.

Softwarové nastavení

Softwarově se nastavuje karta přes inicializační proměnné (I-variables), jedná se nastavení motorů, encoderů atd. Tyto proměnné jsou fixovány v paměti procesoru a mají předdefinovaný význam a typ hodnot (integer, float point, ...). V PMACu jich je obsaženo 1024. Proměnné se smí uložit do paměti EPROM (příkaz SAVE) a při spuštění karty nebo restartu se načtou do paměti RAM procesoru. K defaultnímu nastavení proměnné slouží příkaz $I/\{číslo\} = *$ pro jednu proměnnou, nebo $$$$***$ pro všechny proměnné. Organizace I proměnných je následující:

```

I0 -- I79: Hlavní nastavení karty
I80 -- I99: Převodní nastavení
I100 - I185: Nastavení motoru #1
I185 -- I199: Nastavení souřad. systému 1
I200 -- I284: Nastavení motoru #2
I285 -- I299: Nastavení souřad. systému 2
...
I800 -- I884: Nastavení motoru #8
I885 -- I899: Nastavení souřad. systému 8
I900 -- I979: Nastavení snímačů (Encoder)
I980 -- I1023: Reservovány

```

Význam všech I-proměnných je uveden v [7].

3.3. Vnitřní proměnné PMACu

Kromě inicializačních proměnných (I-proměnných), které byly zmíněny v předchozím odstavci má PMAC další tři druhy proměnných. Těmi jsou P-proměné, Q-proměné, M-proměnné.

- P-proměnné jsou uživatelské proměnné. V paměti jich je pevně fixováno 1024 a mají typ 48 bitové hodnoty s plovoucí řádovou čárkou. Všechny souřadnicové systémy mají přístup k těmto proměnným a používají se především k výpočtu pozic, rychlostí, časů atd.
- Q-proměnné jsou také uživatelsky přístupné a jsou opět 48 bitové s plovoucí řádovou čárkou pevně v paměti. Mají stejné použití jako P-proměnné, ale závisí na tom, jaký souřadnicový systém je právě adresován.
- M-proměnné poskytují přístup do paměti, kde jsou mapovány vstupy a výstupy. Ty se do proměnných přiřazují pomocí příkazu $M\{cislo\}->\{adresa\}$. Každé proměnné je přiřazen specifická adresa a formát dat. Příkladem M-proměnných mohou být výstupy pro řízení motorů nebo vstupy ze koncových snímačů.

Důležité je, že proměnné P,Q,M jsou na rozdíl od I-proměnných měnitelné i z programového bufferu viz dále, protože změnu I proměnné je možné provést pouze online příkazem.

3.4. Příkazy

PMACu zadáváme příkazy v ASCII textovém řetězci. Nejprve přijímá znaky příkazu přes port a umisťuje je do fronty. Když přijde přes port ovládací znak (ASCII hodnota 1 - 31 nejpoužívanější je 13 <CR> carriage return) vezmou se znaky z fronty a provedou se jako příkaz. Při odeslání špatného příkazu vrátí PMAC chybovou zprávu. Příkazy můžeme dělit do dvou základních kategorií online příkazy a příkazy programovému bufferu.

§ Online příkazy

Online příkazy dovolují, například pohnout s motorem, změnit proměnné, vrátit proměnnou, spustit q zastavit programy nebo se dotazovat na stavové informace. Online znamená, že jsou spouštěny PMACem okamžitě, a bud' způsobí nějakou akci, změní nějakou proměnnou, nebo oznámí informaci zpět k nadřazenému PC.

Existují tři základní třídy online příkazů:

- Motor specifikující příkazy, které ovlivňují pouze adresovaný motor
Příklady: #xJ+, #xK, #xP....

Motor je adresován příkazem #n, kde n je číslo motoru s rozsahem 1-8. Tento motor zůstává adresován dokud není přijata další adresa #n.

- Příkazy specifikující souřadnicový systém, které mění adresovaný souřadnicový systém

Příklady: &yUNDEFINE, ...

Souřadnicový systém je adresován příkazem &n, kde n je číslo souřadnicového systému s rozsahem 1-8 a platí stejně pravidlo o přepisu adresování jako u motorů.

- globální příkazy, které ovlivňují kartu bez ohledu na adresovaném módu
Příklady: UNDEFINE ALL, \$\$\$, ???, ABS, OPEN BUFFER, I{x}=y, LIST PC, SAVE....

Všechny online příkazy jsou uvedeny v [7] i s jejich významem. Tyto příkazy nelze použít v otevřeném bufferu pouze za použití příkazu *COMMAND*.

§ Příkazy programovému bufferu

Jak název naznačuje, příkazy bufferu nejsou zpracovány okamžitě, ale jsou uloženy do paměti pro pozdější spuštění programu. V podstatě se jedná o příkazy používané pro psaní motion a PLC programů. Většinou se jedná o zadávání souřadnic, interpolací a rychlostí, v případě motion programů, nebo smyček, podmínekových výrazů, časování, změnu proměnných, pro případ PLC.

Příklady: IF, WHILE, ABS, DELAY, DWELL, X10 Y10, SPLINE1, LINEAR, F,

Všechny programové příkazy jsou uvedeny v [7] i s jejich významem.

Před zadáváním programových příkazů, které se budou ukládat do bufferu, musí být buffer nejprve otevřen online příkazem *OPEN PROG n* nebo *OPEN PLC n*. Po otevření je každý programový příkaz přidán na konec bufferu. Když už byly zadány všechny programové povely, tak se použije online příkaz *CLOSE*, který uzavře buffer. Pro výměnu bufferu slouží příkaz *CLEAR*.

3.5. Operátory a funkce

Při programování je možno pracovat se standardními operátory + sčítání, - odčítání, * násobení a / dělení. PMAC podporuje také matematické funkce, například SIN, COS, TAN, ASIN a spoustu dalších. Pro porovnání dvou hodnot proměnných obsahuje karta tyto komparátory:

=	(rovno)
!=	(nerovno)
>	(větší než)
!>	(není větší než; menší než nebo rovno)
<	(menší než)
!<	(není menší než; větší než nebo rovno)
~	(přibližně rovno - kromě jednoho)
!~	(není přibližně rovno - nejméně od jednoho)

3.6. Pohybový program (motion program)

PMAC může mít v paměti až 256 pohybových programů. PMAC motion program lze psát pomocí motion program jazyku nebo standardizovaným jazykem G machine tool jazyk. Motion programový jazyk má výpočtový a logický konstruktor počítačového jazyka a číselné hodnoty v programu můžou být specifikovány jako konstanty nebo

výrazy. Motion program provádí pohyby jednotlivých motorů po souřadnicích, a proto musí každý tento program běžet v nějakém souřadnicovém systému.

3.6.1 Souřadnicové systémy

Souřadnicový systém je seskupení jednoho nebo více motorů za účelem synchronizace polohování a motion program lze spustit pouze tehdy, je-li aktivován (online příkaz `&{číslo}`). V souřadnicovém systému může běžet právě jeden tento program, a proto může být v celém PMACu spuštěno tolik motion programů, kolik je na kartě definováno souřadnicových systémů.

V PMACu lze aktivovat až 8 souřadnicových systémů, adresovaných od `&1` do `&8`. Rozdílné souřadnicové systémy mohou pracovat jako separátní programy v různých časech. Po aktivaci souřadnicového systému se do něj musí vložit motory jako osy, které v něm budou koordinovány. Toto se provádí přiřazením osy k motoru v tzv. definici os. Platí pravidlo, že musí mít nejméně jeden motor přiřazenou nějakou osu uvnitř tohoto systému, jinak nastane chyba a motion program se nespustí.

3.6.2 Definice osy

Osa je součástí souřadnicového systému a je v něm uváděna jako písmeno vybrané z písmen X, Y, Z, A, B, C, U, V a W. Osa je přiřazena k motoru zadáním adresy motoru `(#{číslo})`, jména a měřítka osy (`X1000`), například `#1->X1000` přiřadí osu X v souřadnicovém systému s měřítkem 1000 impulsů snímače).

Osy se dají rozdělit do tří kategorií podle jejich vlastností:

X, Y, Z	- Hlavní lineární osy souřadnicového systému
A, B, C	- Rotační osy
U, V, W	- Vedlejší lineární osy

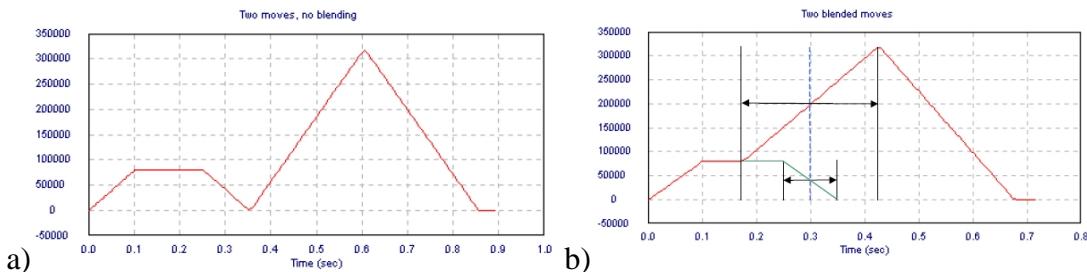
Po nadefinování všech os do souřadnicového systému můžeme pro koordinování pohybu v těchto osách využívat interpolace.

3.6.3 Interpolace

PMAC má pět druhů základních interpolací: LINEAR, RAPID, CIRCLE, SPLINE a speciální PVT. V těchto interpolacích mohou být pohyby specifikovány inkrementálně INC nebo absolutně ABS.

Interpolace LINEAR je lineární interpolací, která a zajišťuje lineárních pohyb na uvedené souřadnice. Navíc je možno nastavit akcelerační doby TA, TM, či nastavení rychlosti pohybu F (feedrate), podle nichž se pohyb provádí. Jestliže navazuje více pohybů na sebe, mohou se provést nezávisle (no blending-bez návaznosti, obrázek 3.4a) užitím příkazu DWELL po zadání souřadnic. Nebo je lze na sebe navázat tak, aby se začal vykonávat další pohyb, ještě před dokončením prvního pohybu (blending). To se provádí nastavením dob TA a TM (obrázek 3.4b). Souřadnice se zadávají programovým příkazem `(osa1){souř.} (osa2){souř.}`, které specifikují koncový bod pohybu.

Příklad lineární interpolace je uveden v příkladu obrábění.



Obrázek 3.4 Příklad lineárních interpolací a)nenavazující b) navazující

Pro použití lineárního pohybu, který se přesunuje na žádané souřadnice maximální rychlosti, je možno použít interpolaci RAPID. Maximální rychlosť je nastavena v I-proměnné $Ix16$ (x číslo motoru) každého motoru. Pro zadávání souřadnic platí stejný příkaz jako u LINEAR.

Kruhová interpolace má dva způsoby vykonávání pohybu v kladném směru (po směru hodinových ručiček) CIRCLE1 nebo v záporném směru CIRCLE2.

Před použitím kruhové interpolace je nejprve nutné specifikovat v jaké rovině se bude kruh vykonávat.

Příkazy pro určení roviny jsou:

<i>NORMAL K-1</i> (defaultní)	rovina X-Y
<i>NORMAL J-1</i>	rovina Z-X
<i>NORMAL I-1</i>	rovina Y-Z

Aby se kruhová interpolace provedla správně, musí být nastavena proměnná I13 na hodnotu mezi 5-10 (nejlépe 7). Tato konstanta specifikuje dobu, ve které se počítají souřadnice pohybu.

Zadávání parametrů kruhové interpolaci:

Koncový bod a rádius:

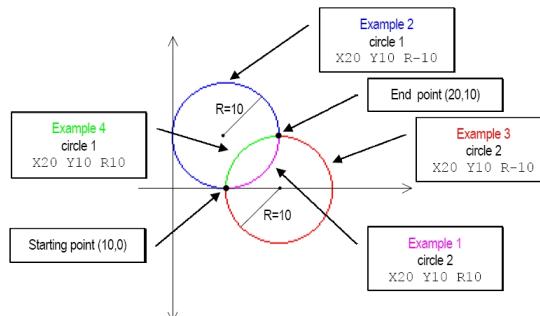
$X\{data\} Y\{data\} R\{data\},$

R(radius) je kladný, pak kruhová výseč bude pod úhlem větší než 180°

R je záporný, pak kruhová výseč bude pod úhlem menším než 180°

Koncový bod a vektor středu kruhu:

$X\{data\} Y\{data\} I\{data\} J\{data\}$



Obrázek 3.5 Příklady kruhových interpolací

Další dvě interpolace jsou vysvětleny v [7].

3.7 Psaní motion programů

Psaní pohybových programů se řídí následujícími pravidly:

- 1) Zavření bufferu *CLOSE* – ošetření v případě nezavřeného bufferu
- 2) Smazání nežádoucích dat – *DELETE GATHER*
- 3) Oddefinování motorů ze souřadnicových systémů – *UNDEFINE ALL*
- 4) Adresování souřad. systému – $\&\{konst\}$, kde *konst* je číslo souř. systému
- 5) Definice os, přiřazení motoru do souř. systému,
 $\#\{motor\} \rightarrow \{měřítko\} \{jméno osy\}$
- 6) Otevření programového bufferu – *OPEN PROG {konst}*, kde *konst* je číslo programu
- 7) Smazání původního obsahu – *CLEAR*..
- 8) V této části se mohou použít všechny programové příkazy.
Příkazy pro pohyb: Zadání interpolace - tabulka interpolací příloha C
Určení módu souřadnic - tabulka módů příloha C
Specifikace průběhu pohyb - tabulka spec. pohybu
příloha C
Souřadnice pohybu - tabulka zadávání souřadnic
příloha C
- 9) Uzavření programového bufferu *CLOSE*

Program se spouští sekvencí online příkazů:

&\{číslo souřadnicového systému\}
B\{číslo programového bufferu\}
R – spuštění programu.

Jestliže chceme program zastavit, můžeme použít online příkazy *CTRL+A*, pro pozastavení, nebo *CTRL+K*, pro úplné ukončení programu.

3.8 PLC programy

PLC programy jsou neustále spouštěny bez ohledu na to, co provádí motion program či jiné PLC. Tyto programy jsou volány asynchronně k pohybu a vykonávají funkci programovatelné logiky. PLC jsou číslovány od 0 do 31 a buď jsou komplikované PLCC nebo nekomplikované PLC. Komplikované jsou rychlejší oproti nekomplikovaným, protože už jsou přeloženy do strojového kódu a také proto, že zabírají menší velikost v paměti. Programy 1-31 jsou spouštěny na pozadí a každý z nich spouští scan nepřerušitelný od jiných úloh na pozadí.

PLC programy mohou obsahovat podmínkové výrazy, smyčky, časovače a dokonce posílat online příkazy.

3.9 Psaní PLC programů

Psaní PLC programů je obdobné jako motion programů skládá se programových příkazů.

Postup:

- 1) Zavření bufferu *CLOSE* – ošetření v případě nezavřeného bufferu
- 2) Smazání nežádoucích dat – *DELETE GATHER, DELETE TRACE*
- 3) Otevření programového bufferu – *OPEN PLC {konst}*, kde *konst* je číslo programu
- 4) Smazání původního obsahu – *CLEAR*
- 5) Vlastní PLC program napsaný z programových příkazů. Typické konstrukce jsou *IF ()-ELS- ENDIF, WHILE()-ENDWHILE*, ad.
- 6) Uzavření bufferu – *CLOSE*

Pro spuštění PLC se musí nastavit proměnná *I5* na 2 , a pak pomocí příkazu *ENABLE PLC {číslo PLC}* se spustí PLC.

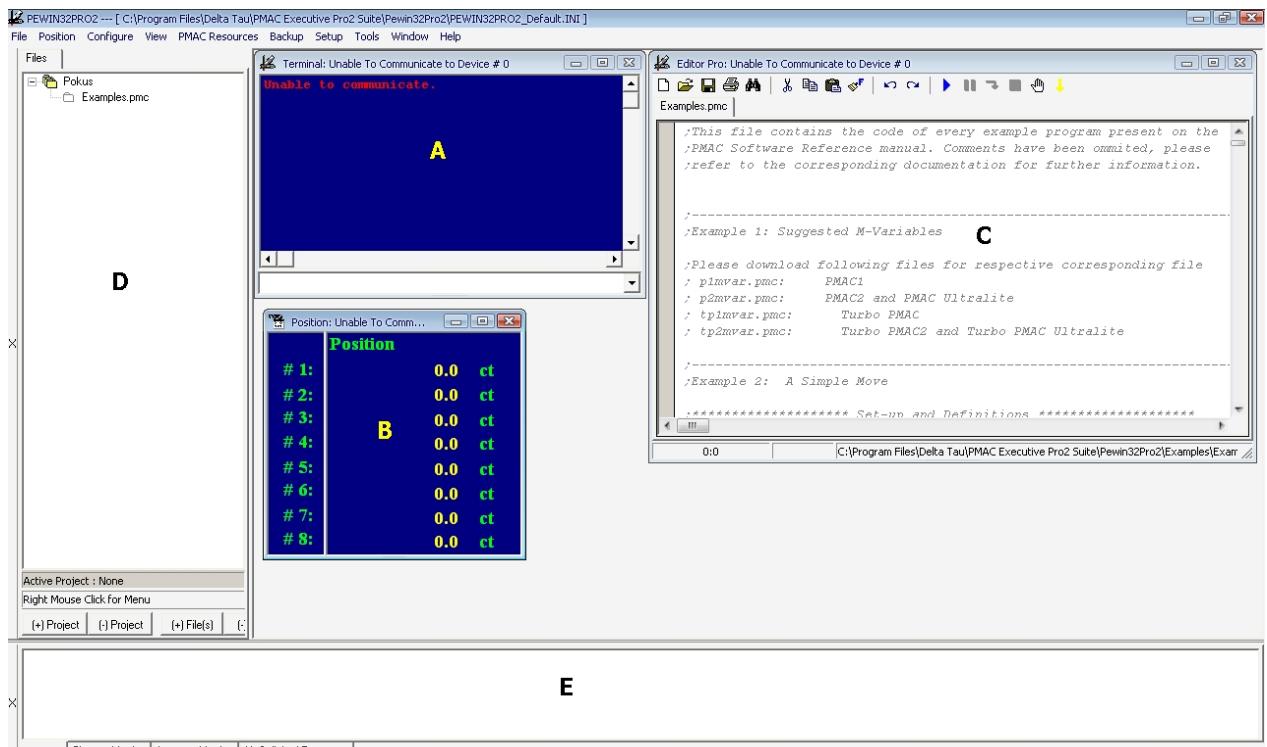
K ukončení PLC programu slouží příkaz *DISABLE PLC {číslo PLC}*.

3.7 Program Pewin32 Pro2

Program Pewin32 Pro2 (PMAC Executive software for Windows 32) je vývojový nástroj od firmy Delta Tau, který umožňuje konfiguraci a programování karet PMAC. Pewin32 Pro poskytuje terminál pro spojení s PMACem (obrázek 3.6A), textový editor (obrázek 3.6C) pro psaní a editaci motion a PLC programů a pracovní prostředí. Navíc obsahuje soubor nástrojů použitelných pro konfiguraci a práci s PMAC.

Pewin32 Pro2 poskytuje vlastnosti a schopnosti:

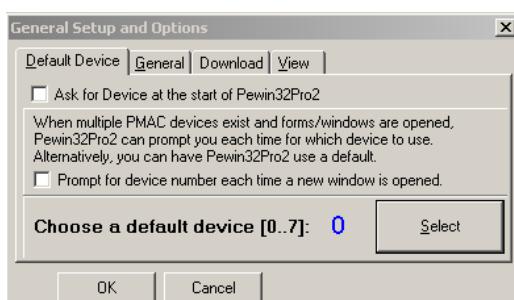
- pracovní prostor řídicích nástrojů pro přizpůsobení rozhraní Pewin32
- nástroj pro organizaci komplexních projektů – obrázek 3.6 D
- terminál pro odesílání příkazů PMAC – obrázek 3.6 A
- okno pro zobrazení pozice, rychlosti a chyb v celém systému – obrázek 3.6 B
- pozorovací okno pro real-timové systémové informace
- real-time PLC překladač.
- kontrola syntaxe kódů programů
- jednoduchá manipulace s vnitřními proměnnými PMACu (I,P,Q a M)
- Real-time stavové zobrazení všech výstupů PMACu



Obrázek 3.6 Vývojové prostředí Pewin32 Pro2

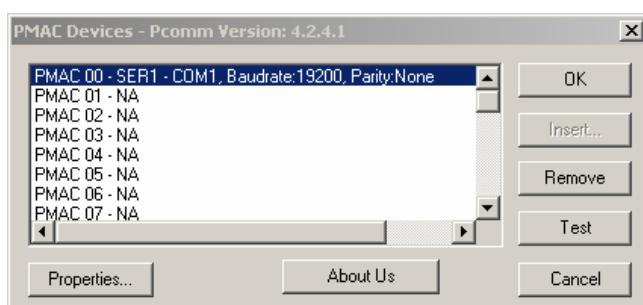
Kromě programu Pewin32 obsahuje instalacní balíček Pewin32 Pro2 Suite doplňkové programy, které slouží k jednoduššímu nastavení karty PMAC, například PMAC Tuning sloužící k nalezení konstant regulátoru, nebo P1Setup, který slouží k nastavení vnitřních proměnných PMACu.

Při prvním spuštění PEWIN je třeba nejprve nastavit komunikaci s PMACem. V menu *Setup->General Setup and Option* se nastavuje komunikace.



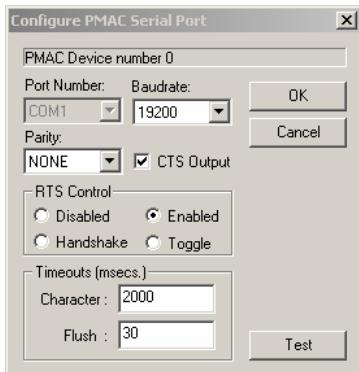
Obrázek 3.7 Dialogové okno pro navázání komunikace s PMACem

V zobrazeném okně po zvolení tlačítka *Select* se otevře další dialogové okno.



Obrázek 3.8 Okno pro výběr PMACu

V tomto okně se objeví všechny nastavené karty. Pokud však není žádný nastaven, musí se příkazem *Insert* vytvořit nové spojení. Pro správné nastavení spojení slouží tlačítko *Properties*, po jehož stisknutí se otevře další dialogové okno, kde se specifikují vlastnosti komunikace. Na obrázku je konfigurace spojení s PMACem přes sériovou linku.



Obrázek 3.9 Okno pro nastavení sériové linky RS232

Po nastavení všech parametrů může uživatel zkusit komunikaci pomocí tlačítka *Test* a pokud se na obrazovku vypíše kladná hláška, pak je spojení navázáno a uživatel může pracovat se všemi funkcemi softwaru.

Pokud se nepodaří navázat komunikaci je třeba zkontrolovat jak nastavení sériové linky, tak kabel na propojení s PMACem. Standardně by měl být umístěn na sériovém portu COM1 přenosová rychlosť 19200Bd, bez parity.

3.8 Nastavení karty pro model obráběcího stroje

V této podkapitole budou uvedeny pouze změny oproti defaultnímu nastavení karty.

Hardware nastavení

Použil pro napájení koncových snímačů 12V, a proto musely být nastaveny spojky následně:

E85 - ON,

E90 - propojit 2 a 3

Jako enable signál jsem použil vstup obvodu /CE. Tento signál aktivuje analogový obvod L299 tehdy, pokud je na něm úroveň L. Proto jsem musel nastavit AENA signálu na opačnou polaritu:

E17A - OFF	motor 1
E17B - OFF	motor 2
E17C - OFF	motor 3

Změna přenosové rychlosti sériové linky pro vyšší rychlosť na 19200 bit/s.

Tabulka 3.1 Tabulka nastavení jumperů přenosové rychlosti sériové linky

E-points				Přenos rychlost [Bd]
E44	E45	E46	E47	
ON	ON	ON	ON	Odpojen
OFF	ON	ON	ON	600
ON	OFF	ON	ON	800
OFF	OFF	ON	ON	1200
ON	ON	OFF	ON	1600
OFF	ON	OFF	ON	2400
ON	OFF	OFF	ON	3200
OFF	OFF	OFF	ON	4800
ON	ON	ON	OFF	6400
OFF	ON	ON	OFF	9600
ON	OFF	ON	OFF	12800
OFF	OFF	ON	OFF	19200
ON	ON	OFF	OFF	25600
OFF	ON	OFF	OFF	38400
ON	OFF	OFF	OFF	51200
OFF	OFF	OFF	OFF	76800

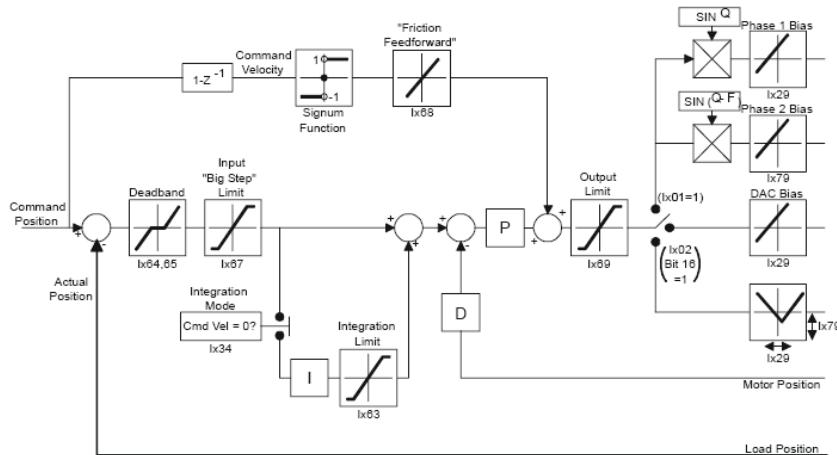
Softwarové nastavení

I-proměnné jsou rozděleny do kategorií hlavní, motory, encodery a souřadnicové systémy. Softwarové nastavení pro model stroje je znázorněno v tabulkách a jsou uvedeny pouze změny od defaultního nastavení.

Tabulka 3.2 Tabulka změn hlavních I-proměnných

Číslo	Význam	Hodnota
I3	I/O Handshance mód	2
I6	Mód hlásící chyby	1
I11	Doba výpočtu programového posunu	10
I13	Doba segmentace programového posuvu	7
I19	Perioda shromažďovaných dat	4

Do softwarového nastavení motorů patří především nastavení regulátorů jednotlivých motorů. Návrh regulátorů jsem prováděl pokusnou metodou. Nejprve jsem nastavil P a I složku na ustálenou odezvu, a pak jsem přidával D složku, aby se odezva zrychlila, ale zůstala stabilní. Jelikož se jedná o regulaci malých motorků, projevují se zde značné nonlinearity, a proto není regulace polohy velmi přesná.



Obrázek 3.10 Regulační smyčka PMACu pro jednotlivý motor

Tabulka 3.3 Tabulka změn I- proměnných odpovídajících motorům

Číslo	Význam	Hodnota M1	Hodnota M2	Hodnota M3
Ix16	Maximální rychlosť	0,5	0,5	10
Ix19	Maximální jog akcelerace	0	0	0
Ix22	Jog rychlosť	0,1	0,1	5
Ix30	Proporcionální složka regulátoru	5000000	5000000	8000000
Ix31	Diferenciální složka regulátoru	35000	35000	1800
Ix33	Integrační složka regulátoru	4000	4000	50
Ix34	Integrační mód	1	1	1
Ix69	Limit výstupu	32767	32767	32767

Nejdůležitější nastavení pro encodery je I-proměnná, která řídí dekódování od snímače. Jelikož máme snímač, který má výstup ve tvaru pulz a směr (kapitola 2), musel jsem tuto proměnnou změnit, protože je defaultně nastavena na A,B kvadratické dekódování.

Tabulka 3.4 Tabulka změn I- proměnných odpovídajících encodérům

Číslo	Význam	Hodnota
I900	Řízení dekodování encoderu 1	4
I905	Řízení dekodování encoderu 2	4
I910	Řízení dekodování encoderu 3	4

Tabulka 3.5 Tabulka změn I-proměnných odpovídajících souřadnicovým systémům

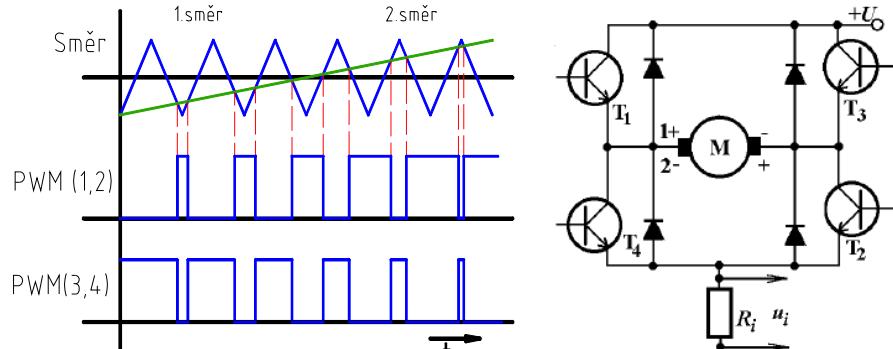
Číslo	Význam	Hodnota K1
Ix87	Defaultní akcelerační doba	100
Ix88	Defaultní doba S-křivky	0
Ix90	Časová jednotka rychlosť posuvu	1000

4 Návrh servozesilovače motorků

Jak již bylo zmíněno v kapitole popisující model obráběcího stroje, jsou osy modelu poháněny stejnosměrnými motorky. PMAC používá pro řízení pohonů výstupy $\pm 10V$. K řízení polohy i rychlosti stejnosměrných motorů se využívá PWM (pulsně šířková modulace) [4]. Tento způsob řízení je výhodný z hlediska efektivnosti, úspory energie,

ale také v plynulosti regulace díky pulsnímu spínání, kdy střední hodnota napětí na motoru je ovlivněna střídou spínání tranzistorů.

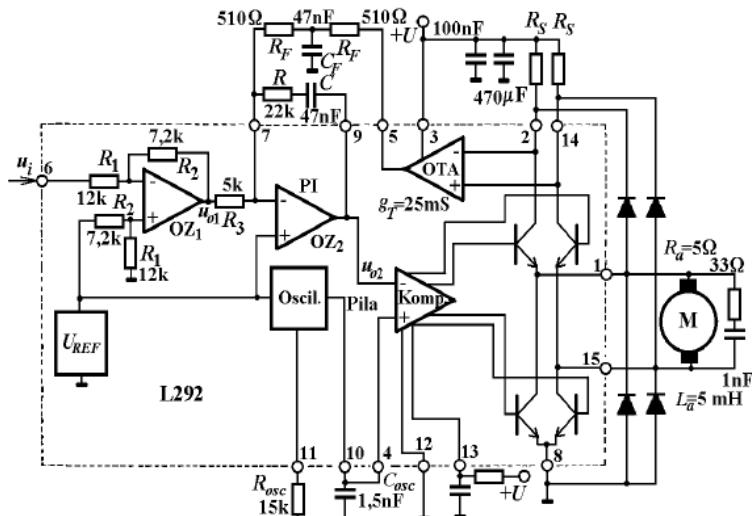
Princip převodu napětí na PWM je znázorněn na obrázku 4.1. Přímka zobrazuje možný rozsah napětí na vstupu (od řídicího modulu PMAC).



Obrázek 4.1 Princip funkce PWM modulace a H můstek

Protože se budou motorky modelu řídit PWM modulací a řídicí karta PMAC řídí pohony analogovým signálem $\pm 10V$, musí se tedy vytvořit obvod, který by převáděl z napětí $\pm 10V$ na vstupu na PWM signál na výstupu.

Vybral analogový obvod L292 od firmy SGS-THOMSON Microelectronics [2], který splňoval požadované podmínky. Strukturu tohoto obvodu je možné popsát blokovým schématem podle obrázku 4.2.



Obrázek 4.2 Vnitřní obvodové schéma analogového obvodu L292

Vstupní obvod s OZ1, které je bipolární, přičítá referenční napětí tak, aby výsledkem byl ovládací signál jedné polarity. Proud protékající můstkem je snímán pomocí dvou shodných odporek R_S . Na těchto odporech je úbytek napětí, jehož polarita záleží na okamžitému směru proudu. Tento úbytek vyhodnocuje zesilovač a jeho výstupem je proud úměrný vstupní napěťové diferenci podle vztahu:

$$i_O = g_T(u_+ - u_-), \quad (10)$$

Výstupní proud z tohoto prvku je filtrován dolní propustí $R_F C_F$. Regulaci v podřazené proudové smyčce zabezpečuje PI regulátor proudu. Nastavení jeho parametrů lze provést některou ze známých metod. Výrobce obvodu ovšem doporučuje tu nejjednodušší: krácení pólů přenosu nulami regulátoru $RC=L_a/R_a$. U PI regulátorů je tento postup použitelný. Volíme-li časovou konstantu tak, aby ke krácení došlo, dostaneme pro přenos otevřené smyčky

$$G_O(s) = \frac{k_m g_T R_s}{sC(1 + sR_F C_F)}. \quad (11)$$

Symbolom k_m je zde označeno statické zesílení mezi napětím u_{O2} na výstupu PI regulátoru a střední hodnotou proudu kotvy motoru. Vyjadřuje tak vlastnosti PWM modulátoru, který lze pokládat za proporcionální člen. Lze je vyjádřit vztahem

$$k_m = \frac{2U}{R_a U_{REF}}, \quad (12)$$

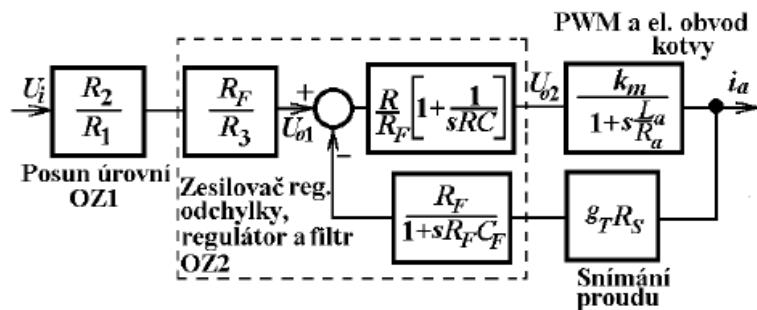
kde U je napájecí napětí a U_{REF} je referenční napětí 8 V. Přenos G_O je druhého rádu a teoreticky by uzavřený regulační obvod měl být vždy stabilní. Je ovšem třeba uvážit, že vzhledem k tolerancím hodnot součástek a změnám některých parametrů za provozu (R_a roste při oteplení motoru) je úplné krácení v přenosu nedosažitelné. Navíc tato analýza některé dynamické vlivy zanedbává. V důsledku zanedbané a nemodelované dynamiky je řád přenosu otevřené smyčky ve skutečnosti vyšší. Jak s ohledem na zaručenou stabilitu, tak s ohledem na kvalitu regulace, je proto vhodné jej navrhnout s dostatečnou fázovou bezpečností. Volíme-li např. fázovou bezpečnost 45° , dostaneme z (21) dodatečnou podmíinku pro hodnoty součástek.

$$k_m g_T R_s R_F C_F < C\sqrt{2}, \quad (13)$$

Celkový statický přenos mezi řídicím napětím u_i a střední hodnotou proudu kotvy je dán vztahem z rovnice

$$\frac{I_a(s)}{U_i(s)} = \frac{R_2 k_m}{R_1 R_2} \frac{sR_F C_F + 1}{s^2 R_F C_F + sC + k_m R_s g_T} \Rightarrow \frac{I_{a0}}{U_{i0}} = \frac{R_2}{R_1 R_3 R_s g_T} = \frac{0,044}{R_s}, \quad (14)$$

Zesílení PI regulátoru $r_O=R/R_F$ lze nastavit tak, abychom dostali vhodný průběh přechodové odezvy. Vzhledem k povaze úlohy by měla přechodová charakteristika být sice rychlá, avšak bez překmitů. Na charakteru regulační úlohy a vlastnostech motoru, především pak na vzájemném vztahu časových konstant τ_m a τ_a záleží, zda při výpočtu nadřazených regulačních obvodů bude nutné uvažovat dynamiku přenosu (24), nebo zda bude možné ji zanedbat a nahradit statickou charakteristikou.



Obrázek 4.3 Blokové schéma obvodu L292

Při analýze otevřené smyčky opět předpokládáme krácení pólů systému, tedy LM/RM=RC, pak přenos otevřené smyčky je

$$Ab = \frac{G_o R_s}{R_4} C \frac{1}{s(1 + sR_F C_F)}, \quad (15)$$

pro dosažení dobré stability se volí fázová bezpečnost větší jak 45° , tím je míněno,

že v $f_F = \frac{1}{2pR_F C_F}$ musí být $|Ab| < 1$ to znamená

$$|Ab|_f = \frac{G_o R_s}{R_4 C} \frac{R_F C_F}{\sqrt{2}} < 1, \quad (16)$$

Pro dobrý výběr útlumu ξ , který určuje kvalitu odezvy systému, se volí hodnota $1/\sqrt{2}$.

V přenosové funkci

$$\frac{I_M}{U_i}(s) = \frac{0,044}{R_s} \frac{1 + sR_F C_F}{1 + 2sR_F R_F + 2s^2 R_F^2 C_F^2}, \quad (17)$$

je absolutní hodnota (modul)

$$\left| \frac{I_M}{U_i} \right|(s) = \frac{0,044}{R_s} 2 \frac{\sqrt{1 + w^2 R_F^2 C_F^2}}{\sqrt{(1 + 2wR_F C_F)^2 + 1} \cdot \sqrt{(1 - 2wR_F C_F)^2 + 1}}, \quad (18)$$

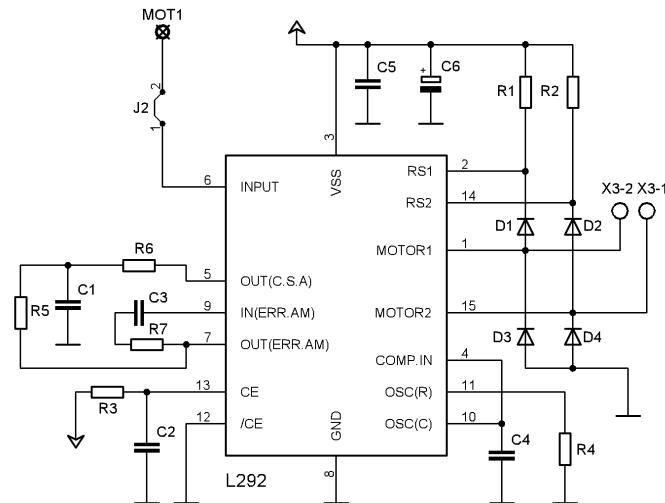
Mezní frekvence je odvozena po dosazení výrazu (18)

$$\left| \frac{I_M}{U_i} \right| = 0,707 \cdot \frac{0,044}{R_s}, \quad (19)$$

Potom mezní frekvence je

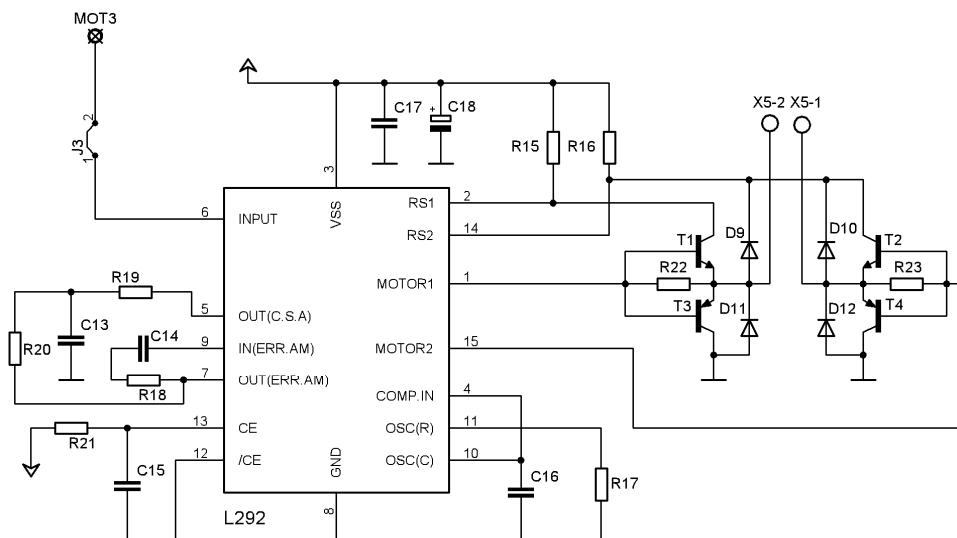
$$f_T = \frac{0,9}{2pR_F C_F}, \quad (20)$$

Jelikož oba motorky pro posun pojazdů jsou malých výkonů, vystačí si obvod L292 s vnitřním H můstkom. Obvod pro řízení motoru pohánějícího vřeteno musí být doplněn o rozšiřující H můstek, aby nedošlo ke zničení obvodu velkým proudem motoru. Zapojení obvodu nastínuje schéma na obrázku 4.4. Tento obvod musí být doplněn externími součástkami (kondenzátory a odpory) tak, aby odpovídal konkrétnímu motoru. Proto je nutné udělat návrh pro jednotlivý typ motoru. (postup návrhu součástek viz [2]).



Obrázek 4.4 Zapojení obvodu L292 pro malé výkony

Obvod L292 pracuje s maximálním napájecím napětím 36 V a zatěžovacím proudem 2A. Z hlediska rozsahu výkonů regulovaných motorů však žádné zlepšení nepřináší. Jeho výkon je ale možné podstatně zvýšit přidáním vnějších výkonových tranzistorů. Všechny funkční schopnosti včetně regulace proudu jsou přitom zachovány a výše uvedená analýza chování obvodu zůstává v platnosti.



Obrázek 4.5 Zapojení obvodu s rozšiřujícím H můstkom pro větší proudy

Návrh součástek pro obvod L292 pro obě zapojení

Návrh vychází z návodu popsáno v manuálu výrobce [2]. Je založen na nastavení vnějších součástek tak, aby vnitřní PI regulátor proudu měl nejideálnější odezvu a zároveň, aby byl stabilní v otevřené smyčce. Jelikož návrh vychází z parametrů řízených motorů, musel jsem nejprve zjistit jejich parametry, které jsou uvedeny v kapitole 2.

Model má dva druhy motorek. Jeden druh poháně vřeteno a druhý poháně posuvy soustruhu, a proto jsem udělal návrh pro oba typy motorek.

Frekvenci PWM modulace jsem nastavil tak, aby byla mimo slyšitelné pásmo.

Tomu odpovídá například frekvence 20kHz a součástky u oscilátoru L292 mají hodnotu $R_{osc} = 15\text{k}\Omega$ a $C_{osc} = 1.5\text{nF}$.

Návrh externích součástek PI regulátoru proudu pro motorky posuvů:

Parametry motoru:	Indukčnost	$L_a = 240 \mu H$
	Odporník	$R_a = 1 \Omega$
	Záběrový proud	$I_M = 1 A$

Napěťové parametry:

Napájecí napětí	$U_s = 24 V$
Maximální vstupní napětí	$U_i = 10 V$

Šířka pásma uzavřené smyčky: $f = 3 kHz$

$$\text{Výpočet: podle rovnice (14)} \quad R_s = \frac{0,044}{I_M} U_{i0} = \frac{0,044}{1} 10 = 0,44 \Omega \text{ zvoleno } 0,47 \Omega$$

$$G_{mo} = \frac{2U_s}{R_a \cdot U_R} = \frac{2 \cdot 24}{0,62 \cdot 8} = 6 \Omega^{-1}$$

$$f_T = \frac{0,143}{R_F C_F} \Rightarrow R_F C_F = \frac{0,143}{5000} = 2,86 \cdot 10^{-5} s$$

$$X^2 = \frac{1}{2} = \frac{400C}{4R_F C_F R_S G_{mo}} \Rightarrow C = \frac{R_F C_F R_S G_{mo}}{200} = \frac{0,47 \cdot 6 \cdot 2,86 \cdot 10^{-5}}{200} = 400 nF$$

$$RC = \frac{L_a}{R_a} = 0,4 \cdot 10^{-3} s \Rightarrow R = \frac{L_a}{R_a C} = \frac{240 \cdot 10^{-6}}{330 \cdot 10^{-9}} = 730 \Omega$$

$$R_F \text{ zvoleno } 510 \Omega \rightarrow C_F = \frac{2,86 \cdot 10^{-5}}{R_F} = \frac{2,86 \cdot 10^{-5}}{510} = 56 nF$$

Návrh externích součástek PI regulátoru proudu pro motorek vřetene soustruhu:

Parametry motoru:	Indukčnost	$L_a = 180 \mu H$
	Odporník	$R_a = 0,55 \Omega$
	Záběrový proud	$I_M = 2,5 A$

Napěťové parametry:

Napájecí napětí	$U_s = 24 V$
Maximální vstupní napětí	$U_i = 10 V$

Šířka pásma uzavřené smyčky: $f = 5 kHz$

$$\text{Výpočet: podle rovnice (14)} \quad R_s = \frac{0,044}{I_M} U_{i0} = \frac{0,044}{2,5} 10 = 0,15 \Omega$$

$$G_{mo} = \frac{2U_s}{R_a \cdot U_R} = \frac{2 \cdot 24}{0,55 \cdot 8} = 11 \Omega^{-1}$$

$$f_T = \frac{0,143}{R_F C_F} \Rightarrow R_F C_F = \frac{0,143}{5000} = 2,86 \cdot 10^{-5} s$$

$$x^2 = \frac{1}{2} = \frac{400C}{4R_F C_F R_S G_{mo}} \Rightarrow C = \frac{R_F C_F R_S G_{mo}}{200} = \frac{0,1 \cdot 11 \cdot 2,86 \cdot 10^{-5}}{200} = 157nF$$

$$RC = \frac{L_a}{R_a} = 1,5 \cdot 10^{-3} s \Rightarrow R = \frac{L_a}{R_a C} = \frac{180 \cdot 10^{-6}}{0,55 \cdot 220 \cdot 10^{-9}} = 1636k\Omega$$

$$R_F \text{ zvoleno } 510 \Omega \quad \rightarrow C_F = \frac{2,86 \cdot 10^{-5}}{R_F} = \frac{2,86 \cdot 10^{-5}}{510} = 56nF$$

Hodnotu odporů R22 a R23 (12Ω) zaručujících otevření tranzistorů vnějšího H můstku, jsem převzal z manuálu výrobce. I ostatní hodnoty součástek jsou také z manuálu [2]. Řídicí modul je sestaven ze tří obvodů L292 i s jeho vnějšími součástkami. Každý obvod bude řídit jednotlivý motor modelu podle signálu z PMACu. Kompletní navržený obvod viz. příloha D.

5 Návrh interface pro propojení PMACu s modelem soustruhu

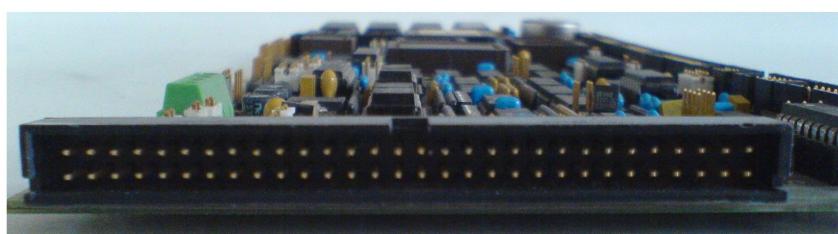
5.6 Požadavky

K řídicí kartě je potřeba připojit vstupní signály ze snímačů polohy, koncových snímačů do řídicí karty a také výstupy, což jsou ovládací signály pro servozesilovač motorů, povolovací signály pro budiče AENA (amplifier enable) a napájecí napětí pro snímače.

Tabulka 5.1 Tabulka seznamu signálů pro spojení s PMACem (konektor JMACH1)

Signál	Význam	Motor	Číslo pinu konektoru JMACH1		
			1	2	3
DAC	Analogový řídicí signál motoru 1		43	44	29
AENA/DIR	Povolovací signál budiče/směr		47	48	33
LIM+	Koncový snímač v kladném směru		53	54	GND
LIM-	Koncový snímač v záporném směru		51	52	GND
CHA	Vstupní kanál encoderu pulsy		25	26	13
CHB	Vstupní kanál encoderu směr		21	22	9
<hr/>					
Společné napájecí svorky					
AGND	Analogová zem		58		
GND	Digitální zem		3,4		
+12V	Napájecí napětí pro koncové snímače		59		
+5V	Napájecí napětí pro snímače polohy		1,2		

Konektor JMACH1 obsahuje vstupní a výstupní piny k PMACu.



Obrázek 5.1 Konektor JMACH1

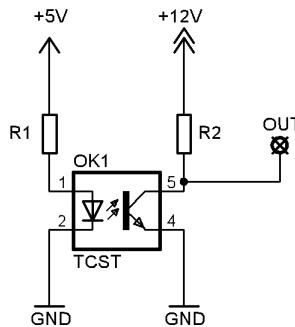
5.7 Připojení snímačů polohy

PMAC disponuje dvěma způsoby připojením snímačů polohy, buď single ended nebo diferenciální. Pro naši aplikaci postačuje zapojení single ended, to znamená, že se signály vyhodnocují proti svorce GND. Signály není třeba v interface nijak upravovat a stačí je přímo připojit k PMACu.

5.8 Připojení koncových snímačů

Ve specifikaci PMACu [5] je popsáno, že při správném zapojení koncových snímačů, by měl trvale držet nízkou úroveň L. Pokud se sepne, nastaví se úroveň H.

Pro koncové snímače jsem využil optický senzor TCST 1103 [3], který pracuje na principu přerušení světelného toku vysílaného od LEDdiody a na výstupu je fototranzistor. Optický snímač je zapojen dle obrázku.



Obrázek 5.2 Zapojení optického snímače TCST 1103

Tabulka 5.2 Tabulka důležitých parametrů snímače TCST1103.

Vstup (emitter)			Výstup (Detector)		
Závěrné napětí	U_{Rmax}	7V	Proud kolektoru	I_{Cmax}	100mA
Propustný proud	I_{Fmax}	60mA	Napětí C-E	U_{CEmax}	70V
Úbytek napětí	U_F	1,25V	Napětí C-E saturační	U_{CEmax}	0,8V

Výpočet odporů: Zvolil jsem proud LEDdiodou na vstupu $I_F=5\text{mA}$

$$R1 = \frac{U_{CC1} - U_F}{I_F} = \frac{5 - 1,25}{5 \cdot 10^{-3}} = 750\Omega, \quad (21)$$

Rozhodovací úrovně vstupů pro koncové snímače řídicí karty jsou odvozeny z napájecího napětí na pinu 59. Jelikož jsem napájal koncové snímače 12V, musel jsem fototranzistor napojit na 12V. Při rozepnutí fototranzistoru se na výstupu OUT snímače drží na úrovni L (0-3V). Při sepnutí se však nastaví do úrovně H (9-12V).

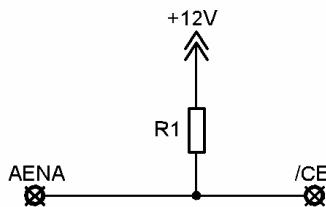
Proud detektorem jsem zvolil proud $I_C=7\text{mA}$ a potom

$$R2 = \frac{U_{CC2} - U_{CEsat}}{I_C} = \frac{12 - 0,8}{7 \cdot 10^{-3}} = 1,6k\Omega, \quad (22)$$

5.9 Zapojení povolovacího signálu AENA (amplifier enable)

Jelikož obvod L292 obsahuje povolovací signál, který okamžitě odpojí servozesilovač, je možno jej připojit k signálu AENA/DIR (amplifier enable, direction). Když není tento signál použit jako Direction pro určení směru otáčení, lze ho použít jako Enable signál. Tento výstupní signál je s otevřeným kolektorem, proto vyžaduje pull up odpor k napětí 12V.

Protože vstup enable signálu je obvod s parametry MOS obvodů, zvolil jsem pull up odpor R1 $10k\Omega$ a zapojení dle obrázku 5.3. Signál /CE se připojí k analogovému obvodu L292 také na pin 12.



Obrázek 5.3 Zapojení pull up odporu pro AENA

Celé zapojení interface je uvedeno v příloze D.

6 Počítačový model obráběcího stroje

Počítačový model jsem vytvořil pro operační systém Windows. Pracoval jsem s poměrně novým prostředím pro aplikace .NET Framework a jazykem C# [10]. NET je nadstavbou operačního systému, která poskytuje uživateli programovat prostředky operačního systému.

Nejdůležitější součástí navržené aplikace je kromě uživatelského prostředí obsluha sériového portu RS232. Aby se mohl zobrazit průběh obrábění modelu, musí se software karty PMAC dotazovat periodicky na pozici osy X, Y a rychlosť pro osu Z. Toto lze zaručit periodickým vláknem, které se opakovaně dotazuje karty PMAC přes sériovou linku na tyto údaje. Tomuto vláknu jsem navíc přiřadil vyšší prioritu.

Jelikož PMAC komunikuje přes RS232 v textové podobě, tak dotaz na pozici os X, Y a rychlosť osy Z vypadá následovně #1P#2P#3V<CR>. Dotaz vrátí pozici motoru 1 a 2 a rychlosť motoru 3. Dotaz se opakuje v intervalech 50 milisekund.

Změna oblasti, ve které probíhá vykreslování obrábění jsem dělal pomocí DoubleBufferingu, zabraňující blikání v oblasti při vykreslování. Metoda spočívá ve vykreslení do bufferu a následně se z něj překreslí na obrazovku jako celek.

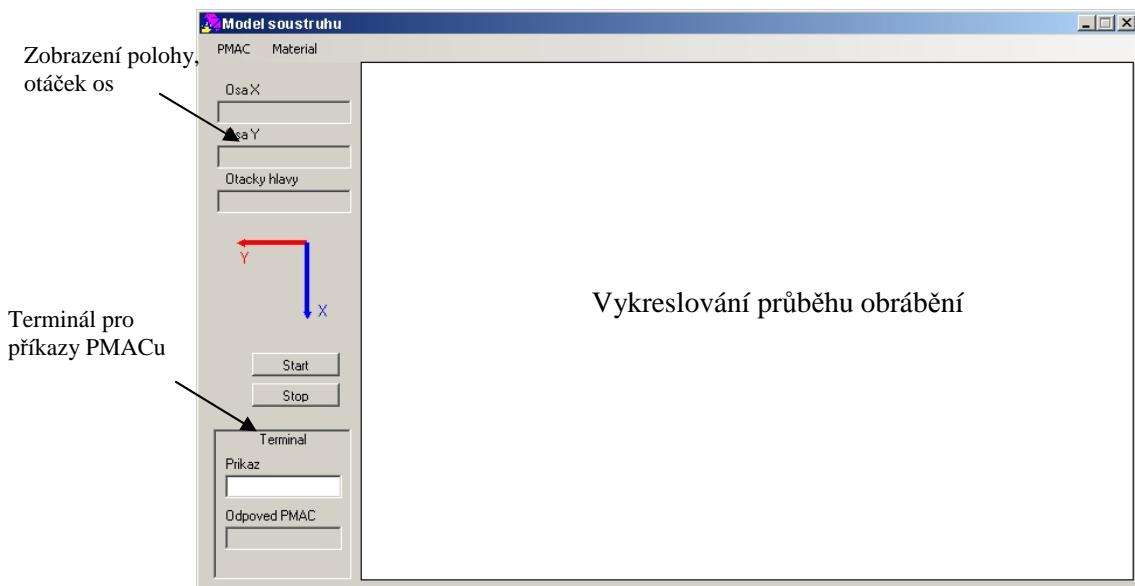
Protože se programy spouští zadáváním specifických znaků (&IB1R) na sériový port, bylo nutné vytvořit terminál k PMACu, přes který se mohou odeslat příkazy PMACu.

Terminál jsem navrhl jako textový box, do kterého se zadávají příkazy.

Po potvrzení klávesou *Enter* se vytvoří obslužné vlákno. Toto vlákno vyšle znaky PMAC a následně přijme výsledek.

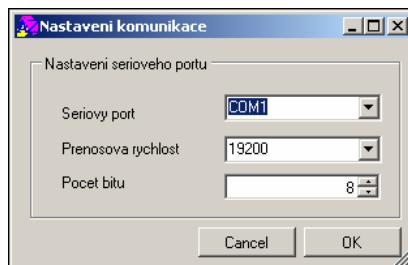
Z důvodu přístupu ke sdílenému prostředku (sériovému portu) jsem musel doplnit aplikaci o mutex, aby nedošlo k situaci, že obě vlákna přistupují k sériovému portu.

Po spuštění software se otevře hlavní okno, které obsahuje vykreslovací oblast, výpis pozic, orientaci os, terminál a také ovládací tlačítka.



Obrázek 6.1 Hlavní okno software

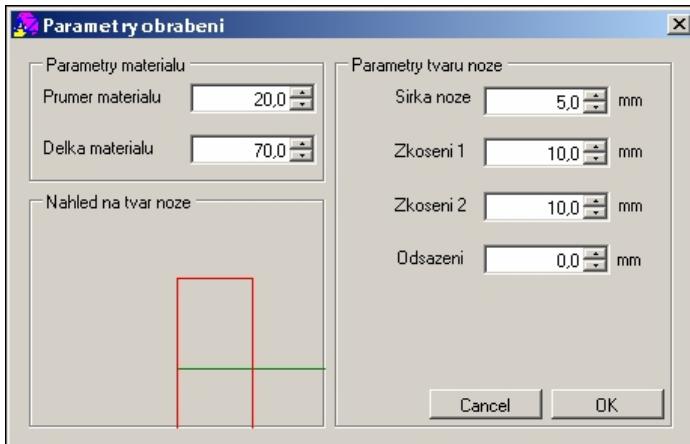
Před navázáním komunikace může uživatel nakonfigurovat vlastnosti sériového portu. Standardní konfigurace při spojení s PMACem při nastavení z kapitoly 3.5 je PMAC připojen na portu COM1, rychlosť 19200 Bd a 8 datových bitů.



Obrázek 6.2 Okno pro nastavení komunikace

Navázání komunikace s PMACem vytvoří uživatel pomocí *Menu->Navazat komunikaci*. Pokud se programu podaří spojit s kartou PMAC, změní se pozadí oblasti terminálu na zelenou barvu a uživatel může zadávat příkazy. Jestliže se však objeví hláška, že nelze navázat komunikaci a zbarví se terminál červeně, je sériový port již používán.

Než uživatel spustí motion program, je třeba zadat parametry obráběného materiálu, tvar a velikost nože, podle kterých se bude vykreslovat průběh obrábění.



Obrázek 6.3 Okno pro nastavení parametrů obrábění

Po zadání všech parametrů můžeme spustit odečítání pozic jednotlivých motorů tlačítkem *Start*, avšak musí být navázána komunikace. Ukončení pohybu a vykreslování obstarává tlačítko *Stop*.

Navržený software je uložen na přiloženém CD.

7 Návod na použití modelu soustruhu

7.6 Pomůcky

Protože není model udělán jako samostatné zařízení, potřebuje pro práci ještě napájecí zdroj, a dockstation pro PMAC.

- Pro napájení je třeba stabilizovaný zdroj 24V s minimální zátěží 4A.
- Dockstation musí obsahovat ISA sběrnici, v které je PMAC zasunut a také zdroj pro jeho napájení.
- PC s nainstalovaným operačním systémem Windows 32bit, .NET Framework 2.0, dále by měl mít sériový port RS232, přes který se komunikuje s PMACom.

7.7 Spouštění a navázání komunikace

Před zapnutím se musí nejprve připojit 60-ti pinový konektor k modelu, kabel na propojení RS232 s PC a napájecí kabely modelu ke zdroji 24V. Po té můžeme zapnout PMAC, nadřízený PC i zdroj 24V.

Po naběhnutí operačního systému může uživatel nainstalovat (pokud již není nainstalován) licencovaný software pro konfiguraci karty PMAC PeWin32Pro z CD, které je přiloženo u bakalářské práce.

Software vyvinutý pro pozorování průběhu obrábění obsahuje přiložené CD. Ten se nemusí instalovat. Stačí pouze zkopírovat složku se softwarem na pevný disk PC a tohoto adresáře jej spustit souborem *seriál.exe* (PC však musí obsahovat .NET Framework 2.0).

Po nainstalování software PeWiN jej můžeme spustit buď přes nabídku Start, nebo pomocí ikony na ploše . Navázání komunikace s PMACom je uvedeno v kapitole 1).

Když se podaří navázat komunikace lze pracovat se všemi komponenty programu PeWin32.

Nejvhodnější pro nastavení proměnných PMACu je přes nástroj, který obsahuje menu

Configure > (I,P,Q,M) Variables. Spuštěním se otevře tabulka proměnných, do které se načtou hodnoty z PMACu. Změny proměnných provedeme tak, že klikneme na proměnu, zadáme novou hodnotu a potvrďme *Enter*. Pokud nelze proměnná změnit nemá do ní uživatel přístup. Je také důležité, že pokud chce uživatel změnu proměnných zachovat i pro další spuštění musí odeslat online příkaz *SAVE*, který uloží všechny proměnné.

Online příkazy se posílají přes Terminál. Ten je umístěn v menu *View>Terminal*.



Obrázek 7.1 Terminál pro zadávání příkazů

Jestliže je nutné sledovat pozici jednotlivých motoru, můžeme použít nástroj z menu.

7.8 Nastavení PID regulátoru

Jestliže uživatel chce nastavit regulátory je nejlepší následný postup.

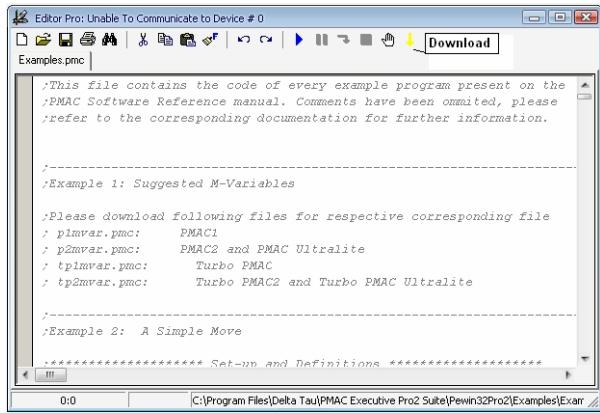
1. Otevřít tabulku s I proměnnými a seřadit ji podle kategorií
2. Otevřít si okno vypisující pozice *View>Position*
3. Vybrat motor, který se bude ladit a zvolit záložku Basic Servo
4. Změnit konstantu PID
5. Otestovat průběh regulace zadáním příkazu `#xJ^{počet impulsů}` do terminálu a v případě nestability stisknout v terminálu *CTRL+K*, který ukončí pohyb
6. Body 3 až 5 se opakují dokud není regulátor správně nastaven

Position		
# 1:	0.0	ct
# 2:	0.0	ct
# 3:	0.0	ct
# 4:	0.0	ct
# 5:	0.0	ct
# 6:	0.0	ct
# 7:	0.0	ct
# 8:	0.0	ct

Obrázek 7.2 Zobrazení pozic motorů

7.9 Psaní motion a PLC programů v prostředí PeWin

Pro psaní programů má PeWin vlastní textový editor. Ten se aktivuje po otevření nebo nové vytvoření souboru v menu *File*. Postup jak napsat a spustit program či PLC je uveden v kapitole 3.7 resp 3.9.



Obrázek 7.3 Textový editor

Po napsání programu se stiskne tlačítko (download) a program se nahraje do PMACu a uživatel ho může spustit příkazy do terminálu

Když je program otestován v prostředí PeWin, může se přejít k softwaru, který zobrazuje průběh obrábění. Ten se spouští souborem *serial.exe*. Postup spuštění motion programu v tomto softwaru je v kapitole 6. Důležité je, aby jiný program včetně PeWin nepřistupoval na sériový port, kde je umístěn PMAC. Z tohoto důvodu je nutné PeWin zavřít.

8 Počítačový model obráběcího stroje

Tento program slouží jako demonstrační příklad, který ukazuje obráběcí možnosti modelu.

Jelikož se jedná model, má omezené možnosti pohybu a přesnosti. Motor 1 se může pohybovat v rozmezí 2000 impulsů a motor 2 asi 5500 impulsů ze snímače. Motor 3 je bez omezení, protože je zařazena rotační rychlostní osa. S ohledem na tyto omezení lze napsat motion program.

M138=0	<i>; Povolení uzavřené smyčky motor 1</i>
M238=0	<i>; Povolení uzavřené smyčky motor 2</i>
CLOSE	<i>; Uzavření bufferu</i>
DELETE GATHER	<i>; Vymazání nežádoucích dat</i>
UNDEFINE ALL	<i>; Vymazání definic všech os souřad. systému</i>
&1	<i>; Aktivování souřadnicového systému 1</i>
#1->X	<i>; Motor #1 definován jako osa X</i>
#2->Y	<i>; Motor #2 definován jako osa Y</i>
OPEN PROG 1 CLEAR	<i>; Otevření bufferu 1 a vymazání obsahu</i>
M114=1	<i>; Povolení servozesilovače motoru 1</i>
M214=1	<i>; Povolení servozesilovače motoru 2</i>
COMMAND"I322=1"	<i>; Otáčky motoru 3 (vřetene)impulsů za msec</i>
COMMAND"#3J+"	<i>; Spuštění motoru 3 (vřetene)</i>
LINEAR	<i>; Lineární interpolace</i>

INC	; Inkrementální mód
TA100	; Nastavení akcelerační doby
F60	; Rychlosť pohybu 60 impulsů/proměnnou I190
X-200 Y200	; Lineární pohyb o 200 impulsů na osách X a Y
DWELL0	; Dokončení pohybu
X200 Y200	; Lineární pohyb o -200 impulsů na osách X a Y
DWELL0	; Dokončení pohybu
NORMAL K-1 ..	; Kruhová interpolace v rovině XY
CIRCLE1	; Aktivace kruhové interpolace (ve směru hod. ručiček)
INC	; Inkrementální mód
INC(R)	; Inkrementální mód rádius
X0 Y1000 R600	; Parametry kruhového pohybu
DWELL0	; Dokončení pohybu
LINEAR	; Lineární interpolace
INC	; Inkrementální mód
TA100	; Akcelerační doba
F70	; Rychlosť pohybu 60 impulsů/proměnnou I190
X0 Y500	; Souřadnice pohybu
DWELL0	; Dokončení pohybu
X-500 Y0	; Souřadnice pohybu
DWELL0	; Dokončení pohybu
SPLINE1	; Spline interpolace
INC	; Inkrementální mód
TA1000	; Akcelerační doba
X50	; Souřadnice pohybu
X250 Y100	; Souřadnice pohybu
X250 Y100	; Souřadnice pohybu
DWELL0	; Dokončení pohybu
RAPID	; Lineární interpolace maximální rychlosťí
INC	; Inkrementální mód
X200	; Souřadnice pohybu
ABS	; Absolutní mód
X1500 Y500	; Souřadnice pohybu
DWELL0	; Dokončení pohybu
COMMAND"<#3J/>"	; Zastavení motoru 3
M114=0	; Odpojení servozesilovače motor 1
M214=0	; Odpojení servozesilovače motor 2
M314=0	; Odpojení servozesilovače motor 3
CLOSE	; Uzavření programového bufferu

Celý program je také na přiloženém CD. Výsledný tvar obrobku převzatý z navrženého software je v příloze H.

Důležité pro správné spuštění programu je nastavení absolutní nuly pozice motorů soustruhu. To se provede tak, že se užitím příkazů $\#xJ^{\{počet impulsů\}}$, kde x je číslo motoru, dojede až na negativní koncové snímače (při dosažení nulové pozice už PMAC nedovolí dojet dále). Zde se musí provést vynulování pozic motorů příkazem \$\$. Následně nastavíme stejným příkazem motory 1 a 2 na pozici, kde se začne provádět

obrábění, pak už stačí spustit příkazem **&IB1R** motion program. Při výskytu chyby je třeba ji dekódovat [7] a odstranit ji. Nejčastější chyba (ERR012) je, že nejsou motory 1 a 2 v uzavřené smyčce, pak se musí proměnné M128 a M138 nastavit na nulu.

Protože při spuštění motion programů je třeba chránit pohyb koncovými snímači musel jsem navrhnout PLC program.

CLOSE	<i>; Uzavření bufferu</i>
OPEN PLC 5 CLEAR	<i>; Otevření PLC bufferu a smazání</i>
IF(M121=1)AND(M122=0)	<i>; Pokud se sepne koncový snímač -LIM1</i>
COMMAND"Q"	<i>; Zastavení motoru 1</i>
ENDIF	
IF(M122=1)AND(M121=0)	<i>; Pokud se sepne koncový snímač +LIM1</i>
COMMAND"Q"	<i>; Ukončení programu</i>
ENDIF	
IF(M221=1)AND(M222=0)	<i>; Pokud se sepne koncový snímač -LIM2</i>
COMMAND"Q"	<i>; Ukončení programu</i>
ENDIF	
IF(M222=1)AND(M221=0)	<i>; Pokud se sepne koncový snímač +LIM2</i>
COMMAND"Q"	<i>; Ukončení programu</i>
ENDIF	
CLOSE	<i>; Uzavření bufferu</i>
ENABLE PLC 5	<i>; Povolení PLC</i>

Pokud nastane případ, že posuvy přejedou hranice vymezené koncovými snímači, PLC ukončí program a zablokuje všechny pohyby.

Jestliže chceme odstranit blokování dalších pohybů, musíme pro jistotu vše ukončit online příkazem **CTRL+K**, následně odpojit PLC, které pohyb blokuje (v našem případě PLC 5) příkazem **DISABLE PLC 5**. Po té se může příkazy **#J^{data}**, odjet mimo vymezenou oblast. Pro správnost je nutné použít právě tento povel, protože si hlídá koncové snímače sám. V případě dosažení koncového snímače v určitém směru nedovolí v tomto směru další pohyb, ale umožní pohyb na druhou stranu.

9 Závěr

V této práci je pospán návrh servozeilovčů k motorkům, návrh interface a propojení s kartou PMAC. Návrh servozesilovače byl ovlivněn tím, že PMAC nemá generování PWM přímo na desce oproti kartám PMAC2.

Propojení všech částí jsem realizoval a úspěšně odzkoušel na příkladech.

Protože jsou v modelu umístěny malé motorky, projevují se značné nonlinearity, jak v převodovce motorků, tak i na šroubových vedeních posuvů. Z toho důvodu se mi nepodařilo přesně navrhnout regulátor polohy. Pohyb motorků se v určitých úsecích přibrzdí, z důvodu zmíněných problémů.

Ostatní části modelu (snímače polohy, koncové snímače, interface) pracovaly spolehlivě.

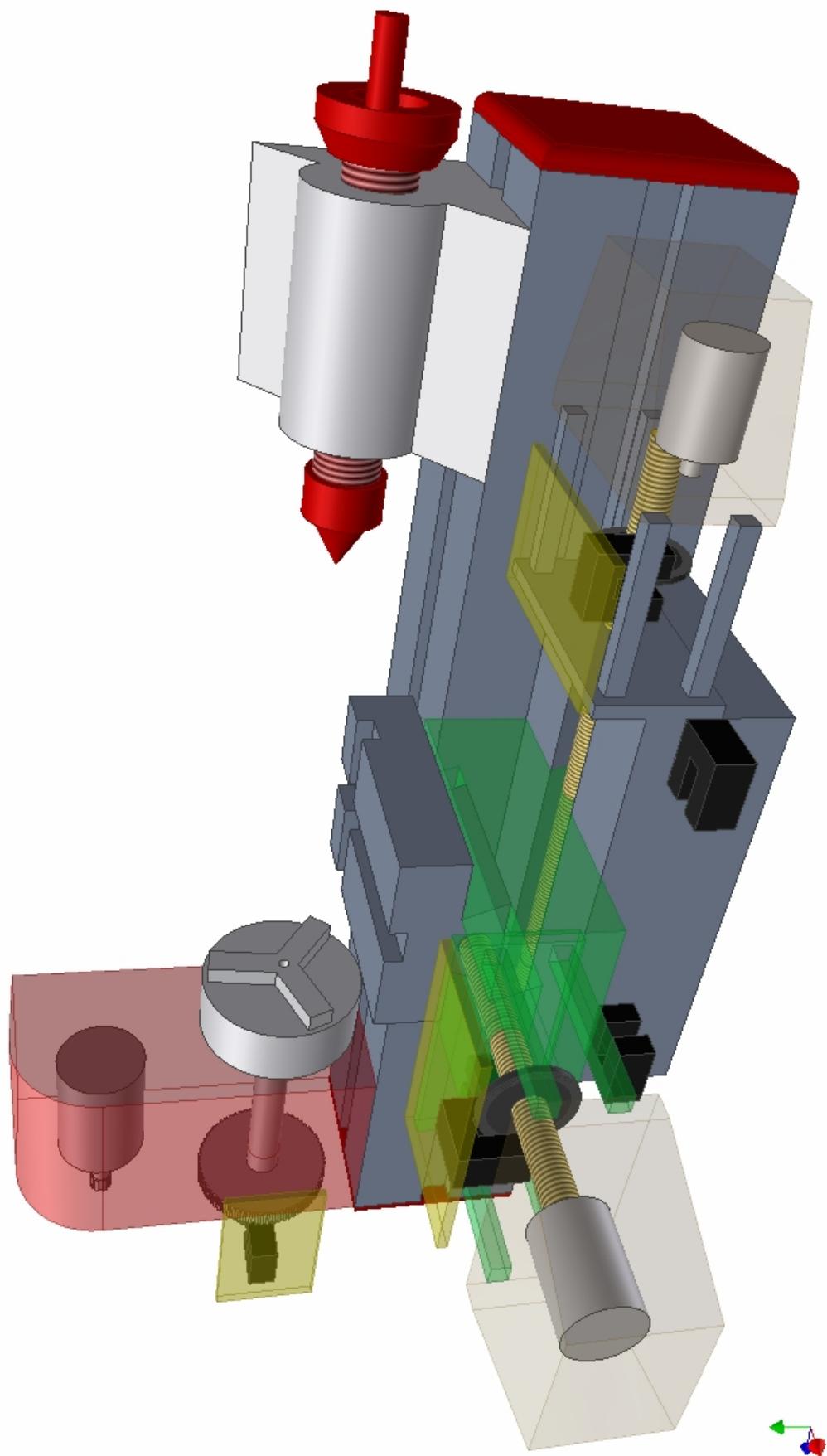
Počítacový model soustruhu pracoval při vykreslování s přesností na 0,2 mm (20 impulsů snímače) i při velkých rychlostech posuvu. Jelikož software nepracuje v real-time operačním systému, může se stát, že vlákno dotazující se na pozici nestihne svoji periodu, tím způsobí, že se nůž vykreslí až v další periodě. Tak se ztratí vykreslovací pozice nože. Proto by se dal software vylepšit naprogramováním v nějakém real-time operačním systému.

Literatura

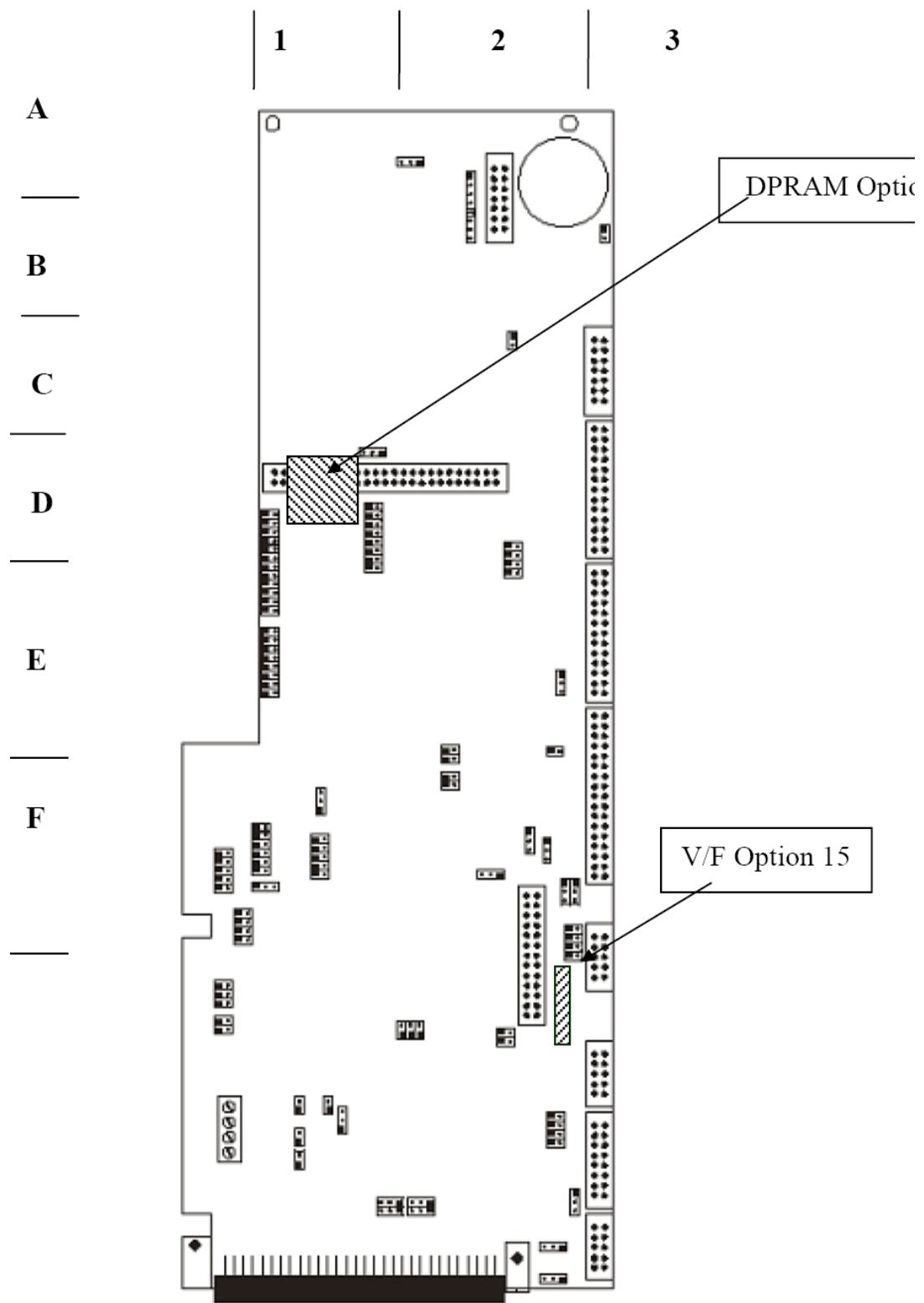
- [1] BAYER J, HANZÁLEK Z, ŠUSTA R: Logické systémy pro řízení, Skripta ČVUT-FEL, Praha 2000
- [2] SGS-THOMSON-Microelectronics: Datasheet L292, Switch-mode driver for DC Motor
- [3] VISHEY: Datasheet TCST 1030, Transmissive Optical Sensor with Phototransistor Output
- [4] T. ŠIMEK, P. BURGET: Elektronické systémy I, Skripta ČVUT-FEL, Praha 2001
- [5] DELTA TAU DATA SYSTEMS: PMAC – User Manual
- [6] DELTA TAU DATA SYSTEMS: PMAC and PMAC2 software reference manual
- [7] DELTA TAU DATA SYSTEMS: PMAC – Quick Reference
- [8] DELTA TAU DATA SYSTEMS: Universal PMAC Lite, Delta Tau Data Systems, 2001
- [9] M. FROHN, W. OBERTHUR, H.-J. SIEDLER, M. WIEMER, P. ZASTROW: Elektronika polovodičové součástky a základní zapojení, BEN- technická literatura, 2006
- [10] J. PROSISE: Programování v Microsoft .NET :webové aplikace v .NET Framework, C# a ASP.NET: Brno Computer Press, 2003

Příloha A.

3D Model obráběcího soustruhu



Příloha B. Rozmístění spojek na desce karty PMAC



E0	E1	E13	F1	E26	H2	E35	E3	E45	C2	E58	C3	E70	D3	E80	F3	E90	G2	E106	A2
E1	E1	E14	F1	E27	H2	E36	F3	E46	D2	E59	C3	E71	D3	E81	F3	E91	D3	E107	F1
E2	E1	E17A	G1	E28	E3	E37	F3	E47	D2	E61	D3	E72	E2	E82	F3	E92	D3	E108	F1
E3	F3	E17B	G1	E29	F3	E38	F3	E48	D1	E62	D3	E73	E2	E83	G3	E93	C3	E109	B1
E4	F3	E17C	G1	E30	F3	E39	D3	E49	D1	E63	D3	E74	E2	E84	G3	E94	C3	E110	C2
E5	F3	E17D	G1	E31	F3	E40	C2	E50	C1	E65	D3	E75	E2	E85	G3	E98	F3	D1	B1
E6	F3	E22	G1	E32	E3	E41	C2	E51	C1	E66	D3	E76	F3	E86	G3	E100	H1	D2	B1
E7	D1	E23	G1	E33	E3	E42	C2	E54	C3	E67	D3	E77	F3	E87	G3	E101	H1	D3	B1
E8	F1	E24	H2	E34A	E3	E43	C2	E55	C3	E68	D3	E78	F3	E88	H3	E102	H1	D21	G1
E9	F1	E25	H2	E34	E3	E44	C2	E57	C3	E69	D3	E79	F3	E89	G2	E103	A1	F1	F1

Příloha C. Tabulky programových příkazů

Interpolce	Programový příkaz
Lineární interpolace	LINEAR
Lineární maximální rychlosť	RAPID
Kruhová ve smere hodinových ručiček	CIRCLE1
Kruhová proti smere hodinových ručiček	CIRCLE2
Spline – jednotná	SPLINE1
Spline - nejednotná	SPLINE2
Speciální Position Velocity Time	PVT

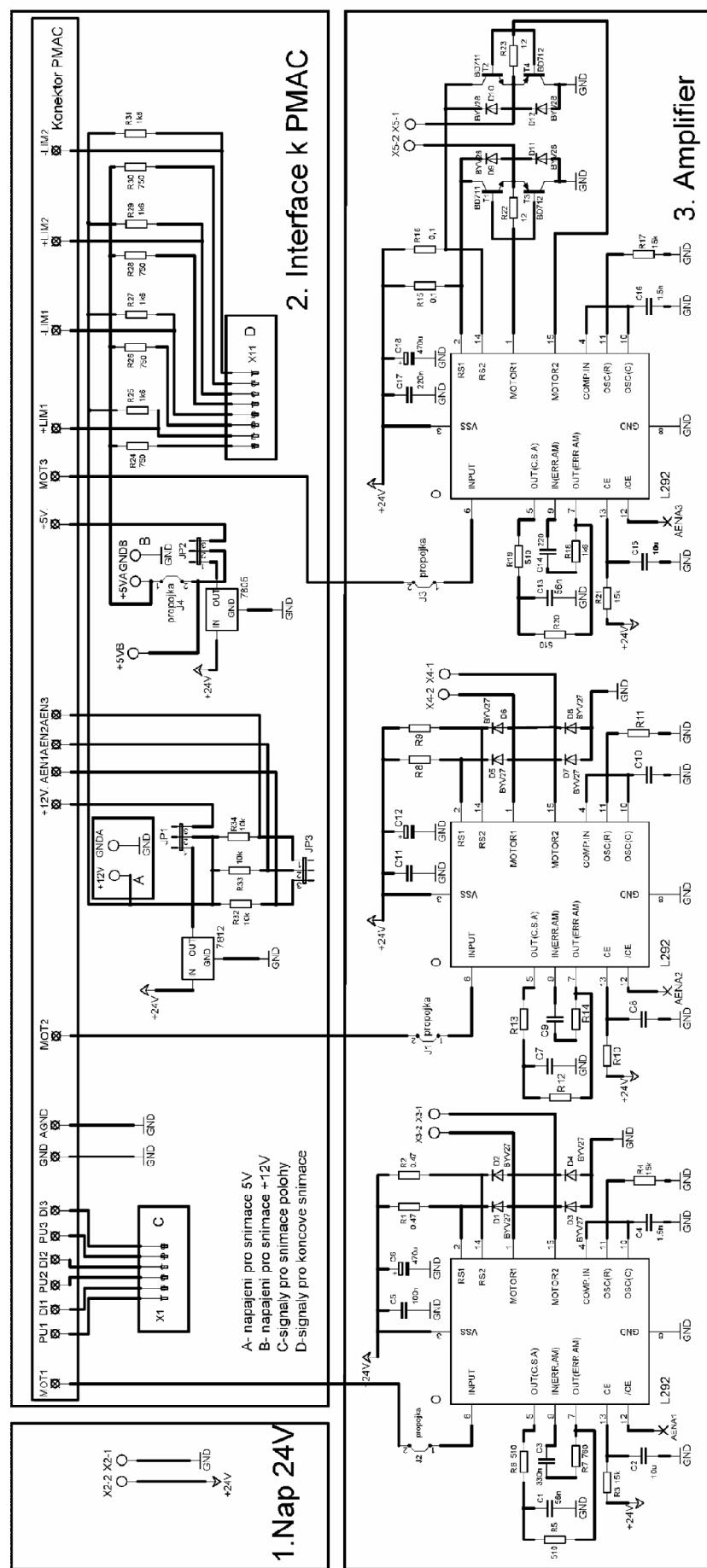
Mód souřadnic	Programový příkaz
Absolutní	ABS
Inkrementální	INC

Specifikace pohyb	Programový příkaz
Akcelerační doba	TA{data}
Doba pohybu	TM{data}
Rychlosť pohybu	F{data}
S-křivka akcelerační doba	TS{data}

Souřadnice pohybu	Programový příkaz
Pohyb v osách	(osa) {data} (osa) {data}
Souřadnice pro kruhovou interpolaci	(osa) {data} (osa) {data} R {data}
Vektor pro určení středu kruh. interpolce	I{data} J{data}

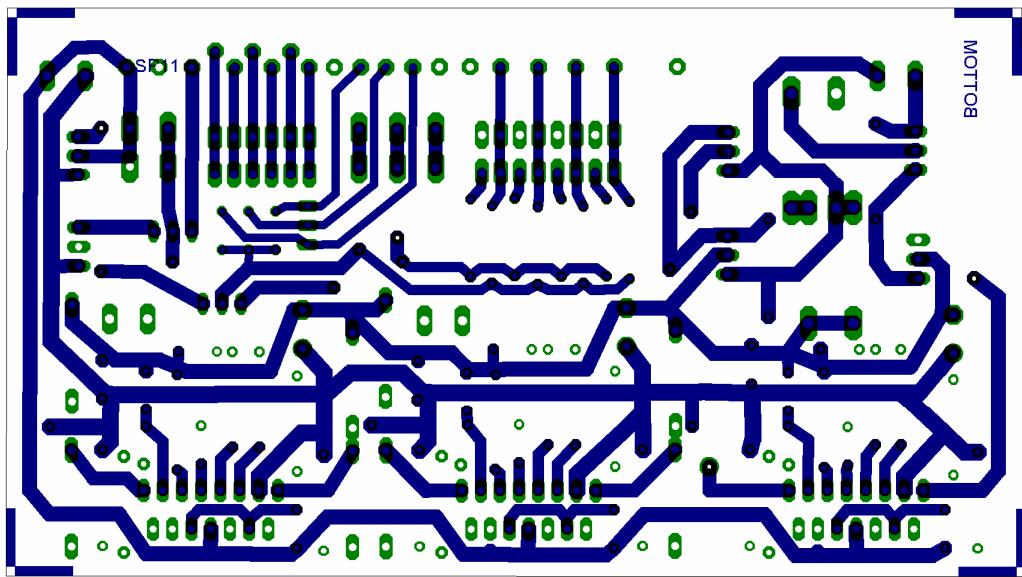
Jiné programové příkazy	Programový příkaz
Podmínka	IF(podmínka)
Nevyhovující podmínce	ELSE
Uzavření podmínky	ENDIF
Podmíněná smyčka	WHILE(podmínka)
Konec smyčky	ENDWHILE
Přiřazení hodnoty do proměnné	{P,Q,M} {číslo}={data,výraz}
Logické spojky	AND, OR
Čekání	DELAY
Zastavení programu	STOP
Poslání online příkazu	COMMAND“příkaz“

Příloha D. Schéma zapojení plošného spoje

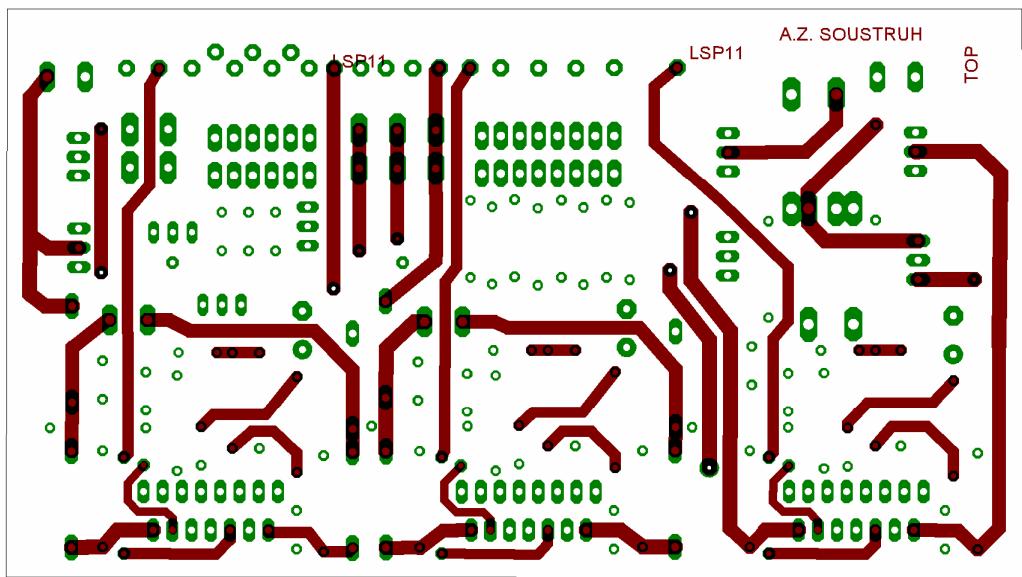


Příloha E. Plošný spoj

Spodní strana plošného spoje (pohled z vrchní strany)

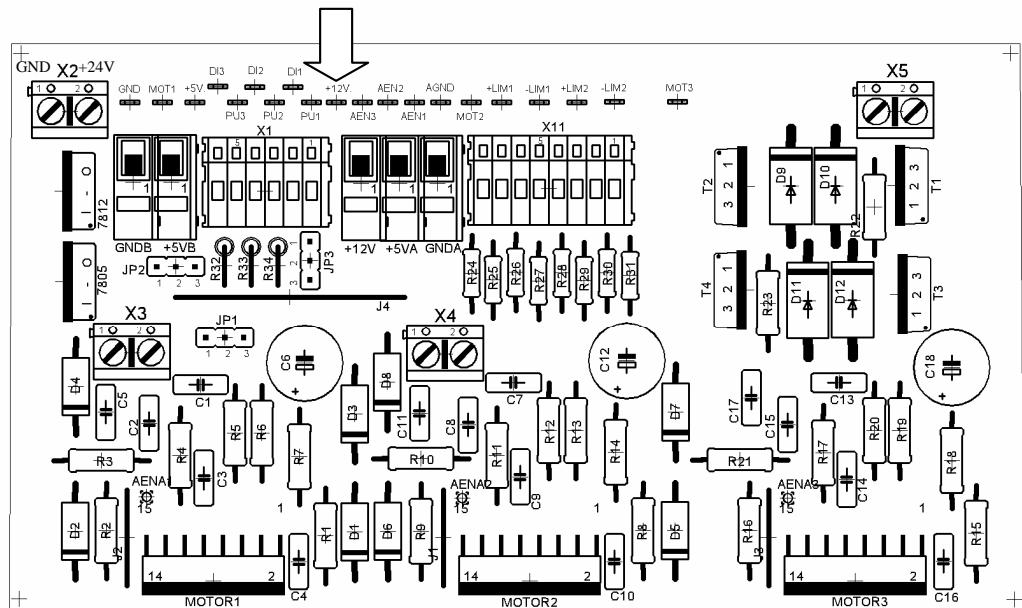


Vrchní strana plošného spoje (pohled z vrchní strany)



Příloha F. Osazení součástek plošného spoje

Osazení součástek na plošném spoji (pohled z horní strany)

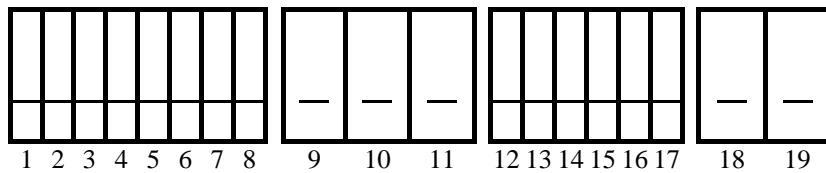


Seznam a parametry součástek:

Stabilizátory		Jumprý
7805	JP1, JP2, JP3	lámací konektorové kolíky
7812		Analogové obvody
Kondenzátory keramické do 50V	MOTOR1	L292
C1,C7	MOTOR2	L292
C3, C9	MOTOR3	L292
C4, C10, C16		Odpory metalizované
C5, C11		Motory 1,2
C13	R1, R2, R8, R9	0.47Ω
C14,C17	R3, R4, R10, R11	15kΩ
	R5, R6, R12, R13	510Ω
Kondenzátory elektrolytické	R7, R14	760Ω
C2, C8, C15		Motor 3
C6, C12, C18	R15, R16	0,1Ω
Diody	R17, R21	15kΩ
D1, D2, D3, D4BYV27/200	R18	1,6kΩ
D5, D6, D7, D8BYV27/200	R19, R20	510Ω
D9, D10, D11, D12	R22, R23	12Ω
Svorkovnice		Interface
+5V	R24, R26, R28, R30	750 Ω
+12V	R25, R27, R29, R31	1,6kΩ
GNDA	R32, R33, R34	10kΩ
GNDB		Tranzistory
X2, X3, X4, X5	T1, T2	H-můstek
kont.	T3, T4	
X1		BD711
X11		BD712

Příloha G. Zapojení svorkovnic a spojek plošného spoje

Zapojení svorkovnic plošného spoje ke kartě PMAC (pohled ze směru šipky z přílohy D (osazení součástek):



Tabulka významů jednotlivých svorek

Číslo	Význam
1	Detektor -LIM2
2	Emitor -LIM2
3	Detektor +LIM2
4	Emitor +LIM2
5	Detektor -LIM1
6	Emitor -LIM1
7	Detektor +LIM1
8	Emitor +LIM1
9	AGND
10	+5V
11	+12V
12	Směr snímač 3
13	Pulsy snímač 3
14	Směr snímač 2
15	Pulsy snímač 2
16	Směr snímač 1
17	Pulsy snímač 1
18	+5V
19	GND

Zapojení propojek na plošném spoji:

JP1 – propojka pro zvolení napájení +12V
 Spojení 1-2 napájení z vlastního zdroje
 Spojení 2-3 napájení z karty PMAC

JP2 – propojka pro zvolení napájení +5V
 Spojení 1-2 napájení z vlastního zdroje
 Spojení 2-3 napájení z karty PMAC

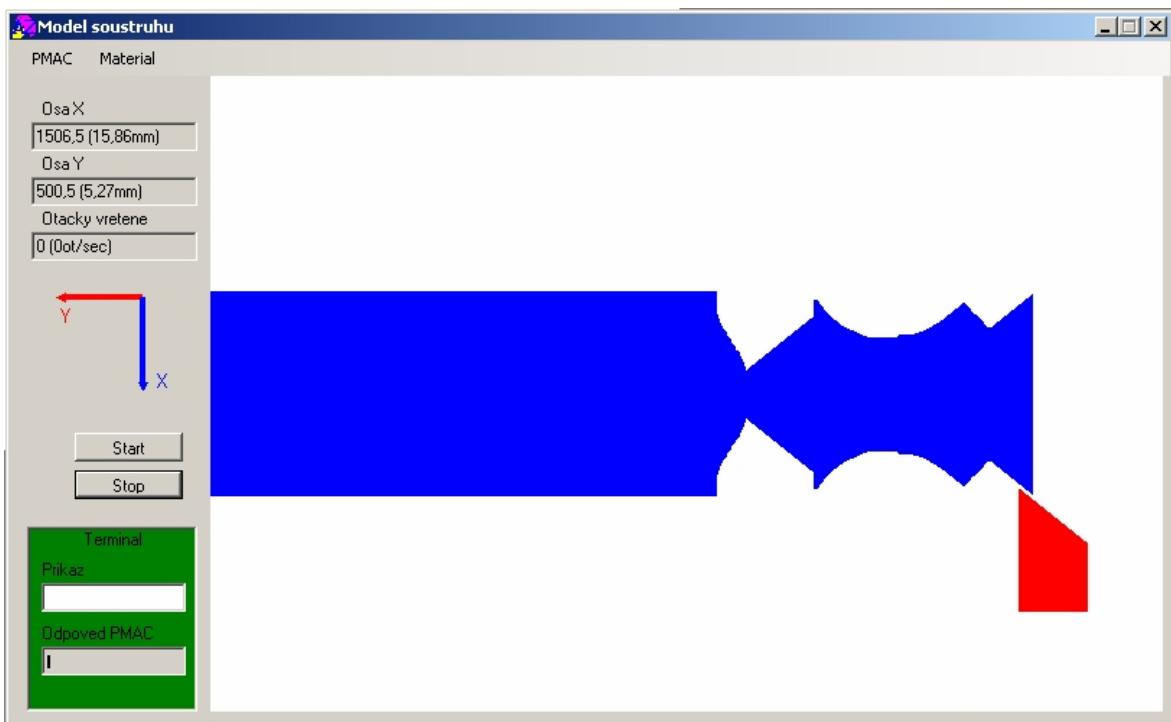
JP2 – konektory pro vnější zapojení ANEA (amplifier enable)
 - pin 1 AENA1 motor 1
 - pin 2 AENA2 motor 2
 - pin 3 AENA3 motor 3

Konektory AENA jednotlivých motorů se vyskytuje v blízkosti obvodu L292.

Konektory pro připojení motorů:

X2- napájení +24V a GND
 X3- motor 1
 X4- motor 2
 X5- motor 3

Příloha H. Výsledek tvar obrobku z příkladu



Příloha I. Obsah přiloženého CD

K této práci je přiloženo CD, které obsahuje materiály používané při práci, software a také příklad programu obrábění.

- Adresář PMAC:
 - § PMACQref.pdf
 - § PMAC AND PMAC2 SOFTWARE.pdf
 - § PMAC USERS.pdf
 - § UNIVERSAL PMAC LITE.pdf
- Adresář Obrázky:
 - § Obrázky schémat a modelu
- Adresář Software:
 - § PMAC Executive Pro2 Suite.zip
 - § ZdrojoveSoubory
 - § SoftModelSoustruh – zobrazní průběhu obrábění
- Adresář ElektDatasheet:
 - § L292 Datasheet.pdf
 - § L292_b.pdf
 - § tcst1030.pdf
 - § SFH 910 datasheet.txt
- Adresář PlosnySpoj:
 - § Soubory z návrhové programu EAGLE
- Adresář PrikladObrabeni:
 - § obrabeni.pmc
- Adresář 3Dmodel:
 - § Soubory z 3D modelovacího programu Invertor
- balalarskaprace.pdf – vlastní bakalářská práce

