

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Řídící program hmCNC

Lukáš Kraicinger

Břeclav 2011

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor: 18. Informatika

Řídící program hmCNC

Autor: Lukáš Kraicinger

Škola: Střední odborná škola průmyslová Edvarda Beneše,
nábřeží Komenského 1
690 25
Břeclav
www.spsbv.cz

Konzultant práce: Ing. Rouček Michal

Břeclav 2011

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně, použil jsem pouze podklady (literaturu, SW atd.) citované v práci a uvedené v příloženém seznamu a postup při zpracování práce je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Břeclavi dne

podpis:

Poděkování

Děkuji Ing. Antonínu Kraicingerovi za obětavou pomoc a podnětné připomínky, které mi během práce poskytoval. Dále kolegovi Liboru Dandovi ze Střední průmyslové školy Břeclav (S4) za testování v praxi.

Anotace

Cílem této práce bylo vytvořit program, který by jednoduchým způsobem zpracoval výkres s navrženými 2D díly na jednoduché 3-osé CNC frézce s vlastním postprocesorem.

Vytvořil jsem program, který zpracuje výkres s díly vytvořenými v prostředí Autocadu. Na tomto výkrese je dále možno provést jednoduché změny a navrhnout postup při řezání jednotlivých dílů. Následně vygeneruje řídicí kód pro řezání na CNC frézce. Před samotným řezáním lze provést otestování tohoto kódu v režimu simulace.

Pro komunikaci s CNC frézku jsem vytvořil vlastní způsob kódování.

Program ve spolupráci s CNC frézku je primárně určen pro užití v amatérské oblasti – řezání modelářských dílů.

Klíčová slova: program, řízení CNC, Autocad, 2D grafika

Anotation

The aim of this work was to create a program which would simply process the drawing with the proposed 2D parts on the simple 3-axis CNC milling machine with own post-processor.

I've created a program that can process drawing with parts made in surrounding of Autocad. It is also possible to make easy changes on this drawing and also propose the process by cutting of individual part. Then it generates the control codes for cutting by CNC milling cutter. It is possible to make the test of this code in simulation mode before cutting.

For communication with the CNC router I created own way of coding.

The program with cooperation of CNC milling cutter is primarily determined for use in amateur sphere – cutting of model's parts.

Keywords: program, operation CNC, Autocad, 2D graphics

1 Obsah

1	Obsah.....	6
2	Úvod	8
2.1	Hlavní vlastnosti.....	8
3	Proces od návrhu k řezání	10
4	Načítání a ukládání	10
4.1	Formát HMC	10
4.2	Formát DXF	10
5	Práce s prvky.....	11
6	Globální proměnné.....	12
6.1.1	Funkce Zpět a Vpřed.....	13
6.1.2	Funkce PŘICHYŤ.....	13
7	Generování kódu	13
7.1	Struktura příkazů v kódu.....	15
7.1.1	Příkazy	15
7.1.2	Pravidla.....	15
8	Algoritmus pro pohyb.....	16
9	Komunikace a řízení	17
10	Aktualizace programu	18
11	Uživatelské prostředí.....	19
12	Ovládací prvky	19
12.1	Hlavní nabídka.....	19
12.1.1	Soubor (výkres).....	19
12.1.2	Úpravy	20
12.1.3	Režim	20
12.1.4	Zobrazit.....	20
12.1.5	Materiál	20
12.1.6	Nástroje	20
12.1.7	Nastavení.....	20
12.1.8	Kód.....	20
12.1.9	Komunikace	21
12.1.10	Nápověda.....	21
12.2	Panel Nástroje.....	21
12.2.1	Výběr	22

12.2.2	Měření	22
12.2.3	Posunout	23
12.2.4	Přečíslovat	23
12.2.5	Odsazení	23
12.2.6	Kopírovat	24
12.2.7	Otočit	24
12.2.8	Změnit měřítko	25
12.2.9	Seskupit, Oddělit	25
12.2.10	Přerušení	25
12.2.11	Kreslení	25
12.3	Panel Zobrazit	26
12.4	Režimy	26
13	Pracovní plocha	27
13.2	Zobrazení	28
14	Použití klávesnice a myši	28
15	Nastavení materiálu	29
16	Nastavení nástroje	30
17	Nastavení aplikace	30
17.1	Nastavení motorů	31
17.2	Pracovní plocha	31
17.2.1	Pravítko	31
17.2.2	Popisky	31
17.2.3	Vlastnosti	31
17.2.4	Zoom faktor	31
17.3	Přerušění čar	32
17.4	Generování kódu	32
17.5	Připojit na port COM při spuštění	32
17.6	Zjišťování aktualizací při spuštění	32
18	Strojová část	33
19	Závěr	33
20	Další vývoj	33
21	Mé další projekty	33
22	Slovník pojmů	33
23	Seznam obrázků	34
24	Seznam příloh	34

2 Úvod

Při studiu oboru „Strojírenství - konstruování s podporou počítače“ jsem se seznámil s oblastí CNC strojů. Tato oblast mě velice zaujala, proto jsem se jí začal věnovat více. S kolegou Liborem Dandou jsme postavili vlastní CNC frézku, kterou bylo potřeba nějakým způsobem řídit. Proto jsme začali zjišťovat, co k tomu bude zapotřebí. Na internetu bylo několik zajímavých řešení, která ale měla vždy nějaké nedostatky.

Vzhledem k tomu, že mám určité znalosti programování z předchozího projektu, rozhodl jsem se, že program vytvořím od základů znovu s vlastním řešením.

Stanovil jsem si tedy požadavky, které by měl program splňovat:

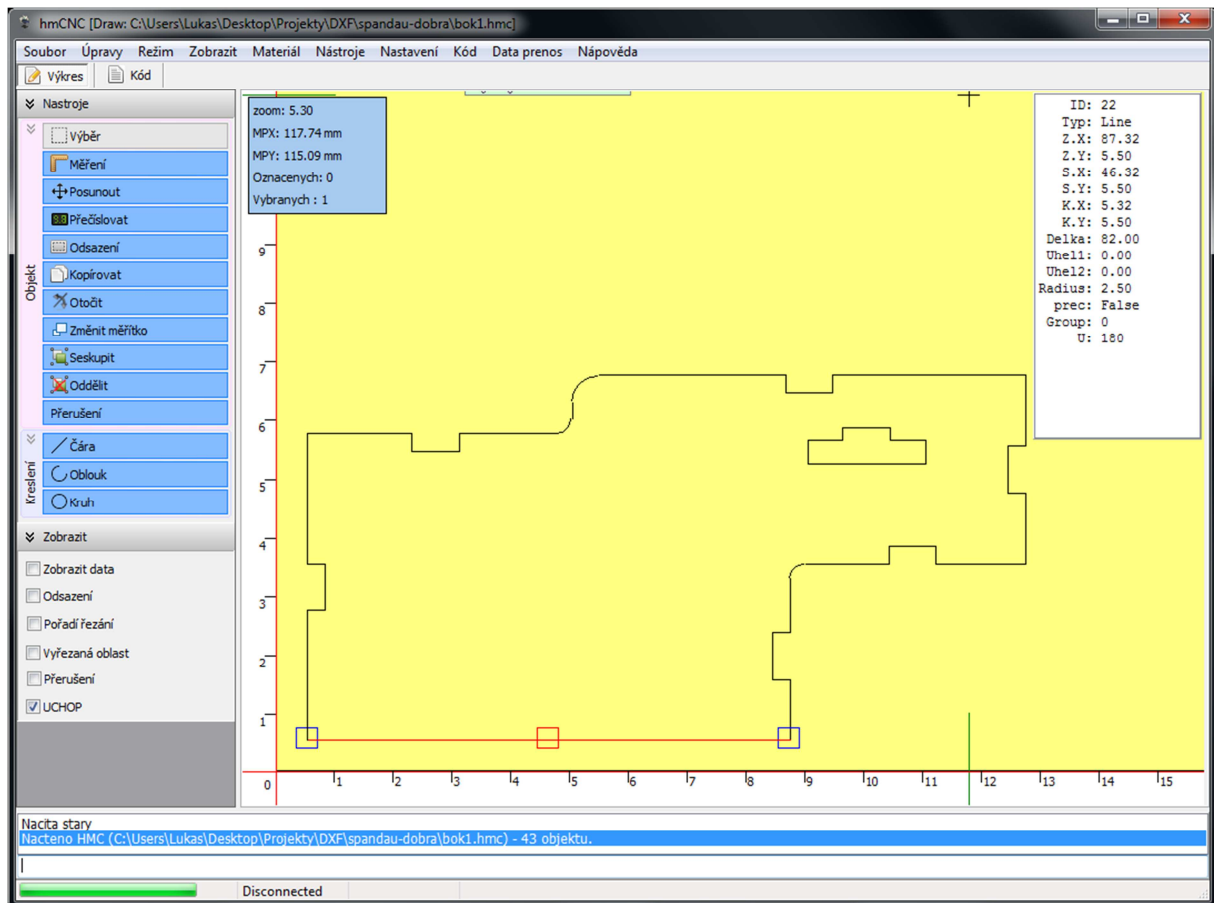
- Přímé načítání výkresů vytvořených v profesionálních programech (AutoCad)
- Žádné ruční psaní kódu pro řízení frézky – proto „Automatické generování kódu“ (vlastní postprocesor)
- Jednoduché ovládání
- Snadná příprava dat pro řezání

Aplikaci, kterou jsem pojmenoval „hmCNC“, jsem vytvořil v Delphi 2010 za použití základní sady komponent, trial verzí komponent společnosti TMS software a bezplatně dostupných komponent DelphiX. Aplikace je typu win32, tím je zajištěna funkčnost na systémech Windows 2000 až Windows 7 (32bitové i 64bitové). Starší systémy (Windows 98) nebyly testovány. Pro chod programu nejsou vyžadovány žádné další součásti typu .net Framework.

V této práci jsem řešil problémy nejen technické, ale také matematické.

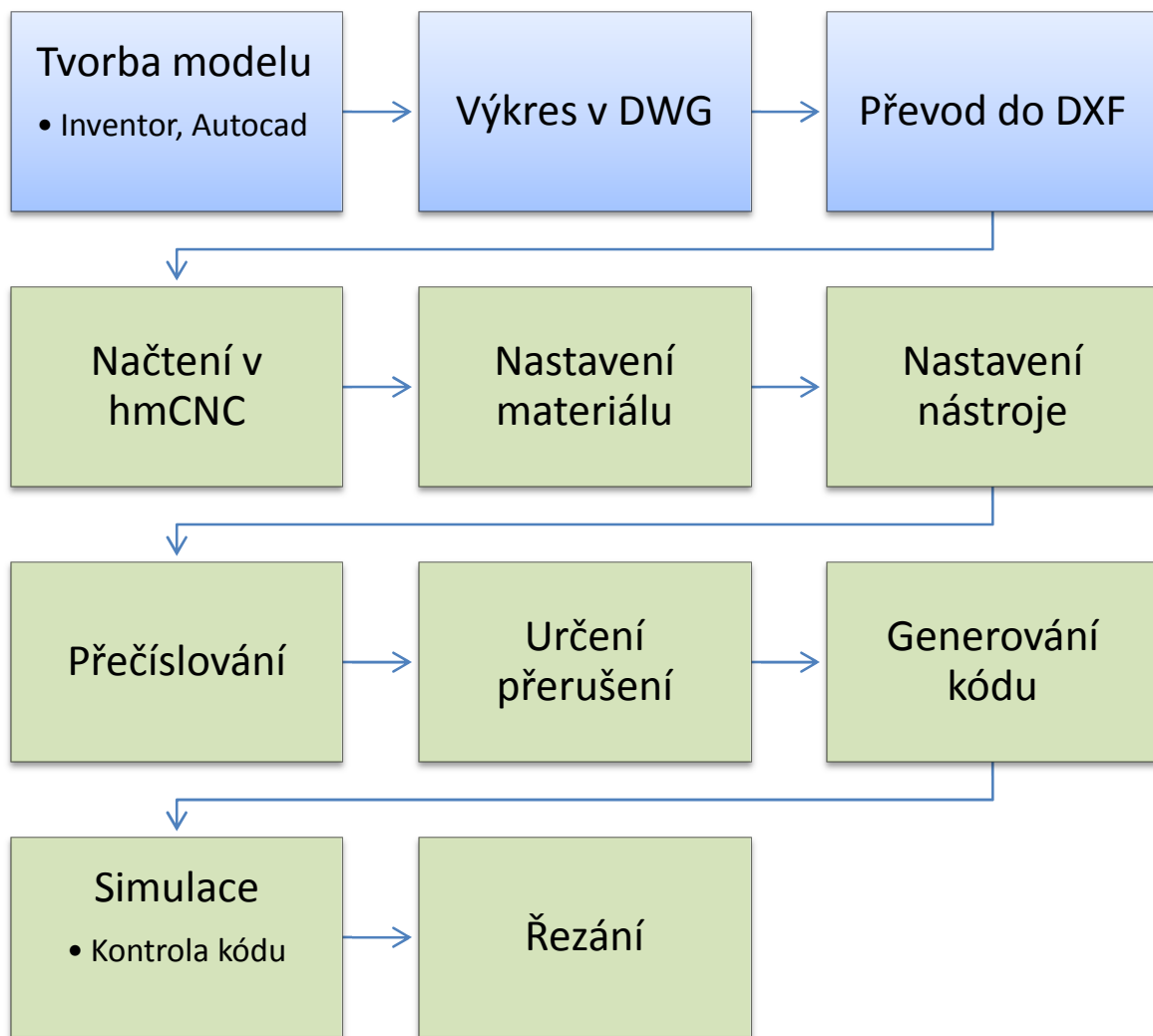
2.1 Hlavní vlastnosti

- Import / Načtení 2D grafiky ze souboru DXF
- Vlastní formát kódu řízení frézky
- Vlastní postprocesor
- Základní kreslení (čáry, oblouky, kruhy)
- Komunikace přes sériovou linku
- Offline programování
- Simulace řezání



Obrázek 1 Náhled programu

3 Proces od návrhu k řezání



Obrázek 2 Schéma procesu

4 Načítání a ukládání

Načítání souboru s výkresem je ve dvou formátech: HMC – vlastní struktura a ve formátu DXF (AutoCAD). Po vybrání se automaticky dle přípony souboru určí algoritmus pro načtení souboru.

4.1 Formát HMC

Formát HMC jsem vytvořil z důvodu jednoduchého ukládání dat jednotlivých prvků. Vzhledem k doplňování dalších vlastností samotné aplikace mají soubory HMC vždy na začátku umístěnou informaci o verzi. Díky tomu je zajištěna zpětná kompatibilita se starší verzí programu. Tento formát je použit jako jediný pro uložení dat. Ukládá se vždy v poslední verzi.

4.2 Formát DXF

Pro přenos výkresů vytvořených v prostředí AutoCAD je použit formát DXF. Formát DXF byl vyvinut firmou Autodesk Inc a specifikaci tohoto formátu DXF zveřejnila na svých

webových stránkách. Vzhledem k objemu importovaných dat je nejlépe využitelný formát verze 2000.

Soubor DXF načítám jako text a procházím po jednotlivých řádcích. Rychlost načítání určuje především verze DXF. Verze 2000 se ukázala pro potřeby načtení dat jako nejefektivnější, neboť získávám jen potřebné informace o jednotlivých prvcích. Například u čar je to počáteční a koncový bod, u kruhů a oblouků středový bod, poloměr, počáteční a koncový úhel.

Proces načítání zobrazuje ukazatel ve stavovém řádku hlavního okna.

5 Práce s prvky

Program umožňuje práci se třemi typy prvků - čára, oblouk a kruh. Všechny tyto prvky jsou uloženy ve stejném datovém poli s názvem *Objekty*. Velikost tohoto pole se mění dynamicky. Používám vlastní datový typ: Ukázka v příloze 3

Hlavní proměnné:

- *Typ* – čára, oblouk, kruh
- *Počáteční, středový a koncový bod*
- *Úhly počáteční a koncový* – používá se u oblouků
- *Poloměr* – u oblouků a kruhů
-

Proměnné vedlejší (počítané nebo doplněné při práci):

- *Pořadí řezání* – určuje, v jakém pořadí se budou prvky řezat
- *Vrchol oblouku* – je bod, který se vypočítá pomocí:

```
Objekty[I].V := GetBodzRAndA2(Objekty[I], Objekty[I].Radius, DegToRad(Objekty[I].U));
```

Funkce *GetBodzRAndA2* vypočítá polohu bodu v 2D prostoru s pomocí počátečního bodu, vzdálenosti a úhlu v radiánech. Funkcí *DegToRad* převedu úhel ve stupních na radiány.

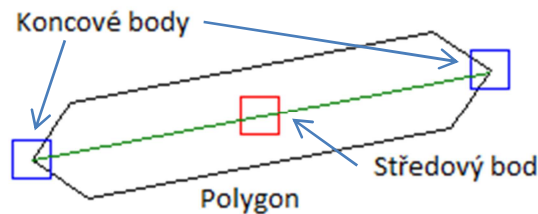
- *Délka*
 - u čar se vypočítá pomocí funkcí *Get2PointRange*, kde zadávám 2 body:

```
Objekty[I].Delka := Get2PointRange(Objekty[I].Z, Objekty[I].K);
```

- u oblouků – úhel × poloměr

```
Objekty[I].Delka := DegToRad(Objekty[I].Angle2-Objekty[I].Angle1)*Objekty[I].Radius
```

- *Skupina* – proměnná typu integer, při hodnotě 0 se nepočítá do skupiny
- *Přerušeni* – je typu boolean, při hodnotě *true* se při generování kódu vypočítají body umístění přerušeni
- Proměnná *B* je pole 6 bodů, které vytváří kolem prvku – čáry polygon. Při najetí myši do tohoto polygonu se prvek označí, viz obrázek 3



Obrázek 3 Výběrový polygon

- *Označený* – typu boolean, nastává hodnoty *true* při najetí myši nad prvek nebo umístěním do výběrového obdélníku, potom je prvek zvýrazněn zeleně
- *Vybraný* – typu boolean, nastává hodnoty *true* při proměnné *Označen* na *true* a kliknutím na prvek, poté se prvek zobrazuje červeně a můžeme s ním dále pracovat
 - Označit a Vybrat můžeme i středový a koncové body
- Další proměnné jsou dodatečné pro výpočty např. pro posouvání, kopírování, odsazení

6 Globální proměnné

Globální proměnnou rozumíme proměnnou, která je dostupná pro všechny funkce v aplikaci.

- *Myš* – typ *TDbIPoint*, určuje skutečnou polohu myši na vykreslovací komponentě *DXDraw*
- *Aktuální načtený soubor* (*AktualniSouborDraw*) – cesta k načtenému souboru výkresu, (*AktualniSouborMat*) cesta k načtenému materiálu
- *SelBox* – typ *TMRect*, obsahuje informace o výběrovém obdélníku, skládá se ze dvou bodů typu *TDbIPoint*
- *Počátek* – *TDbIPoint* – používá se při kopírování, otáčení a změně měřítka, jako výchozí bod
- *Režim* – proměnná typu integer, která nabývá číslo 0 až 2 podle vybraného režimu aplikace
- *BackIndex* – ukazatel na pole změn provedených ve výkresu
- *Proměnné SX a SY* – určují vzdálenost absolutní nuly prostoru vzhledem k nulovému bodu komponenty *DXDraw* (levý horní roh). Proměnná je reálné číslo.
- *MP* – proměnná typu *TDbIPoint*, určuje polohu myši v prostoru pracovní plochy.

```
MP.X := (Mys.X - SX)/N;
```

```
MP.Y := -(Mys.Y - SY)/N;
```

- *N* – určuje měřítko všech zobrazených prvků. Zoomováním se toto číslo mění. Při vykreslování prvků se všechny jeho souřadnice násobí touto proměnnou. Vykreslení čáry z bodu A do B, kde funkcí *round* zaokrouhlím vynásobenou hodnotu na celé číslo a poté k ní přičtu hodnotu proměnné *TSX*, *TSY* (tyto proměnné mají stejnou hodnotu jako *SX*, *SY* jen jsou zaokrouhleny na celé číslo)

```
DXDraw1.Surface.Canvas.MoveTo(round(A.X*N)+TSX, -round(A.Y*N)+TSY);
```

```
DXDraw1.Surface.Canvas.LineTo(round(B.X*N)+TSX, -round(B.Y*N)+TSY);
```

6.1.1 Funkce Zpět a Vpřed

Při práci s prvky bylo nutné vytvořit tzv. „BackBuffer“. To je dynamicky se měnící pole polí prvků (*Multi-dimension array*).

```
BackBuff : array of array of TObjekt;
```

Do tohoto pole se při jakékoli změně prvků (posun, kopie, ...) zapíše celé pole *Objekty* pomocí funkce *WriteBackBuff*. Při potřebě funkce zpět nebo vpřed používám funkci *ReadBackBuff*, která celé pole *Objekty* nahradí vybraným polem z *BackBuff*.

6.1.2 Funkce PŘICHYŤ

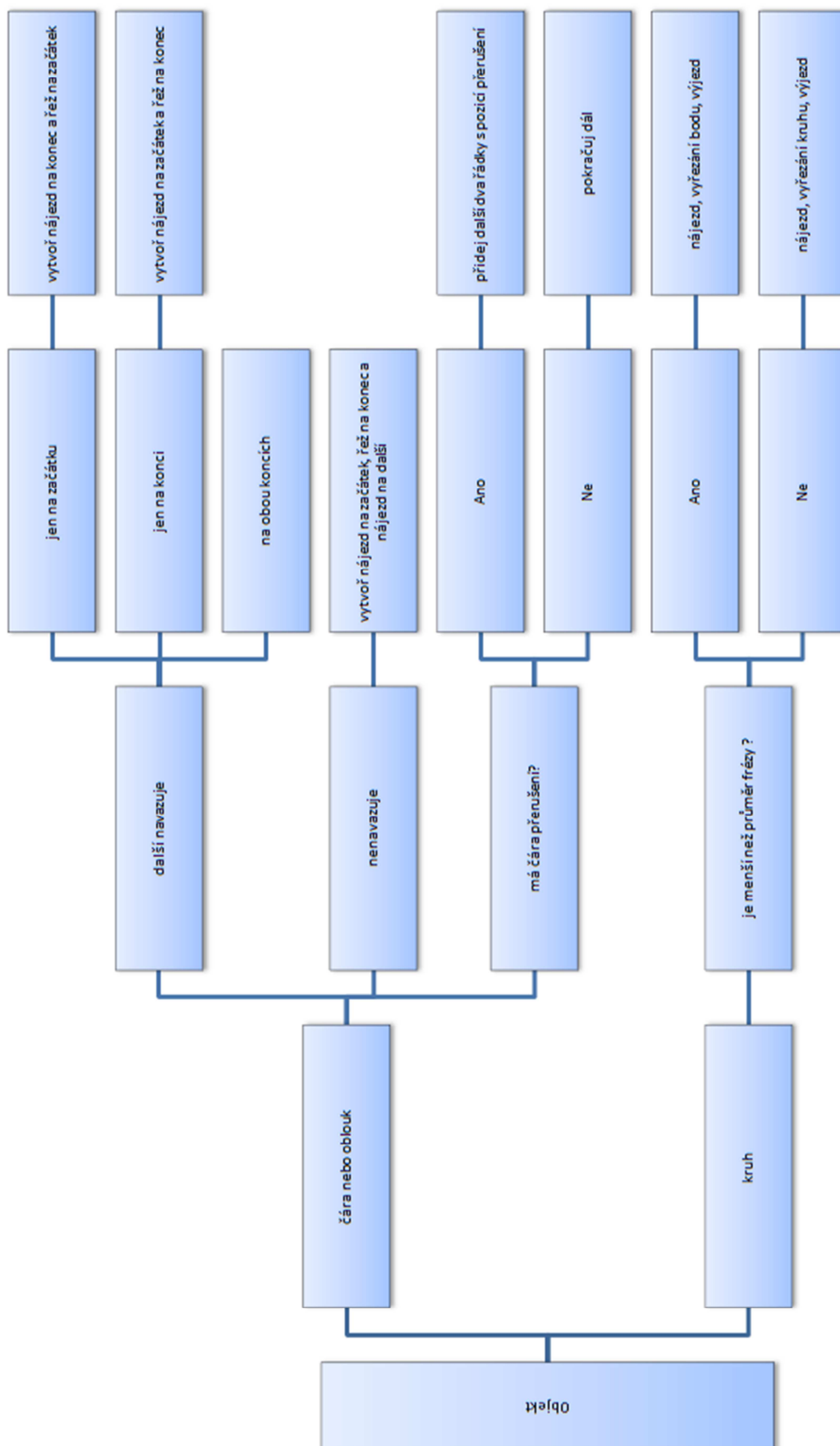
Slouží k přichycení myši (proměnná MP) k nejbližšímu koncovému nebo středovému bodu prvku nebo k absolutní nule pracovního prostoru.

7 Generování kódu

Algoritmus generování kódu je jeden z nejsložitějších v celém programu. Ten jsem musel třikrát vytvořit úplně od začátku, neboť jsem musel měnit například datové typy apod. Na začátku kódu jsou vždy stejné části - test komunikace se strojem, preventivní vypnutí spouštěcího relé frézy, nastavení výšky a nájezd frézy do počáteční pozice. Následně prochází data podle nastaveného pořadí řezání. Zjišťuje typ prvku (objektu), zda je to čára, oblouk, nebo kruh. Je-li to čára nebo oblouk, zjišťuje, zda na tyto objekty navazují další čáry. Dojde k vyloučení duplicity společných bodů. Potom musí zjistit, jakým směrem se bude prvek - čára řezat. Jestli z bodu A do B nebo z B do A, což jsou koncové body čáry. V případě, že žádné další čáry nenavazují, musí vytvořit řádek, ve kterém je nájezd na další polohu. V případě, že je to kruh, zjistí jeho poloměr, ověří, že není menší jak poloměr nastavené frézy a podle toho vytvoří požadovaný řádek kódu. Celý kód se nejprve vytvoří do „bufferu“ a poté se vloží do editačního pole. To zvyšuje rychlost generování. Čas generování se pohybuje v řádu milisekund. Vygenerované hodnoty souřadnic jsou v krocích.

V případě zaškrtnutí políčka *Převádět oblouky a kruhy na čáry* v nastavení aplikace budou veškeré oblouky a kruhy přepočteny na čáry. To se provádí při zapisování do bufferu.

Ukázka v příloze 2



Obrázek 4 Schéma generování kódu

7.1 Struktura příkazů v kódu

Příkazy jsou tvořeny prvním hlavním písmenem a dodatečnými parametry. Název parametru je vždy až za jeho hodnotou. Například:

L1234X5678Y4321Z

Tímto příkazem se fréza přesune na souřadnice $X = 1234$, $Y = 5678$, $Z = 4321$.

7.1.1 Příkazy

- T Odešle znak T a stroj odpoví posláním Test.
- R Zapne relé, nemá parametry.
- E Vypne relé, nemá parametry.
- P Posláním P do čipu mi vrátí aktuální pozici, nemá parametry.
- A Nastaví v čipu pozici $X = 0$, $Y = 0$ a $Z = 0$
- L Přesune frézou na souřadnice zadané v parametrech příkazu, v poslední verzi se přepočty polohy provádějí v PC.
- O Používá se při řezání oblouků a kruhu, při převádění oblouku a kruhů na čáry se v kódu nevyskytuje.
- V K nastavení výšky frézy zadané v parametrech. Vyjede v ose Z na horní spínač, kde nastaví pozici maximálního posuvu nahoru přednastavenou v čipu. Poté sjede na zadanou výšku. Tento příkaz se provádí přímo v čipu.
- W Tento příkaz se používá k vytvoření časové prodlevy. Hodnota parametru je v milisekundách.
- Q Nájezd do výchozí polohy.
- S Říká stroji ukončení řezání. Musí být vždy v kódu.
- 1 Motory na ose Y se otočí o krok vpravo.
- 2 Motory na ose Y se otočí o krok vlevo.
- 3 Motory na ose X se otočí o krok vpravo.
- 4 Motory na ose X se otočí o krok vlevo.
- 5 Motory na ose Z se otočí o krok vpravo.
- 6 Motory na ose Z se otočí o krok vlevo.

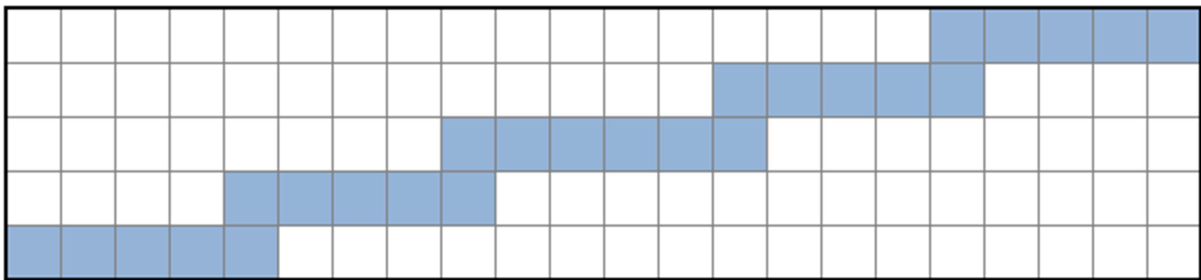
7.1.2 Pravidla

Kód musí splňovat tato pravidla:

- Jeden příkaz je pouze na jednom řádku
- Jeden řádek obsahuje pouze jeden příkaz
- Veškeré poznámky musí být za dvěma lomítky //
- Kód musí začínat příkazem T (Test zařízení)
- Musí být ukončen příkazem S (stop)
- Příkazy nesmí obsahovat mezery
- Příkazy nesmí mít přeházené parametry

8 Algoritmus pro pohyb

Nejprve pohybuje s osou Z (najede na určitou výšku), pak pohybuje současně osou X a Y tak, aby za stejnou dobu byla pozice frézy na zadaných souřadnicích. Tyto výpočty se provádí v PC. Funkcí *StepMotor* se do čipu odešle číslo 1-6 (1 je pohyb o krok na ose Y vpravo, 2 je krok Y vlevo, ...), také změní hodnotu proměnné pX, pY a pZ (aktuální pozice frézy). Přesun frézy do pozice lze přerušit kliknutím na zobrazené tlačítko STOP. Dojde k nastavení proměnné *wstop* na hodnotu *False* a k předčasnému ukončení algoritmu pro pohyb. V případě, že aktuální pozice frézy po nájezdu na výchozí pozici neodpovídá požadované pozici, je tato funkce provedena znovu. Algoritmus vychází z Bresenhamova algoritmu. Typickým případem užití tohoto algoritmu je kreslení přímky na obrazovce monitoru.



Obrázek 5 Kreslení přímky

Ukázka algoritmu v příloze 1

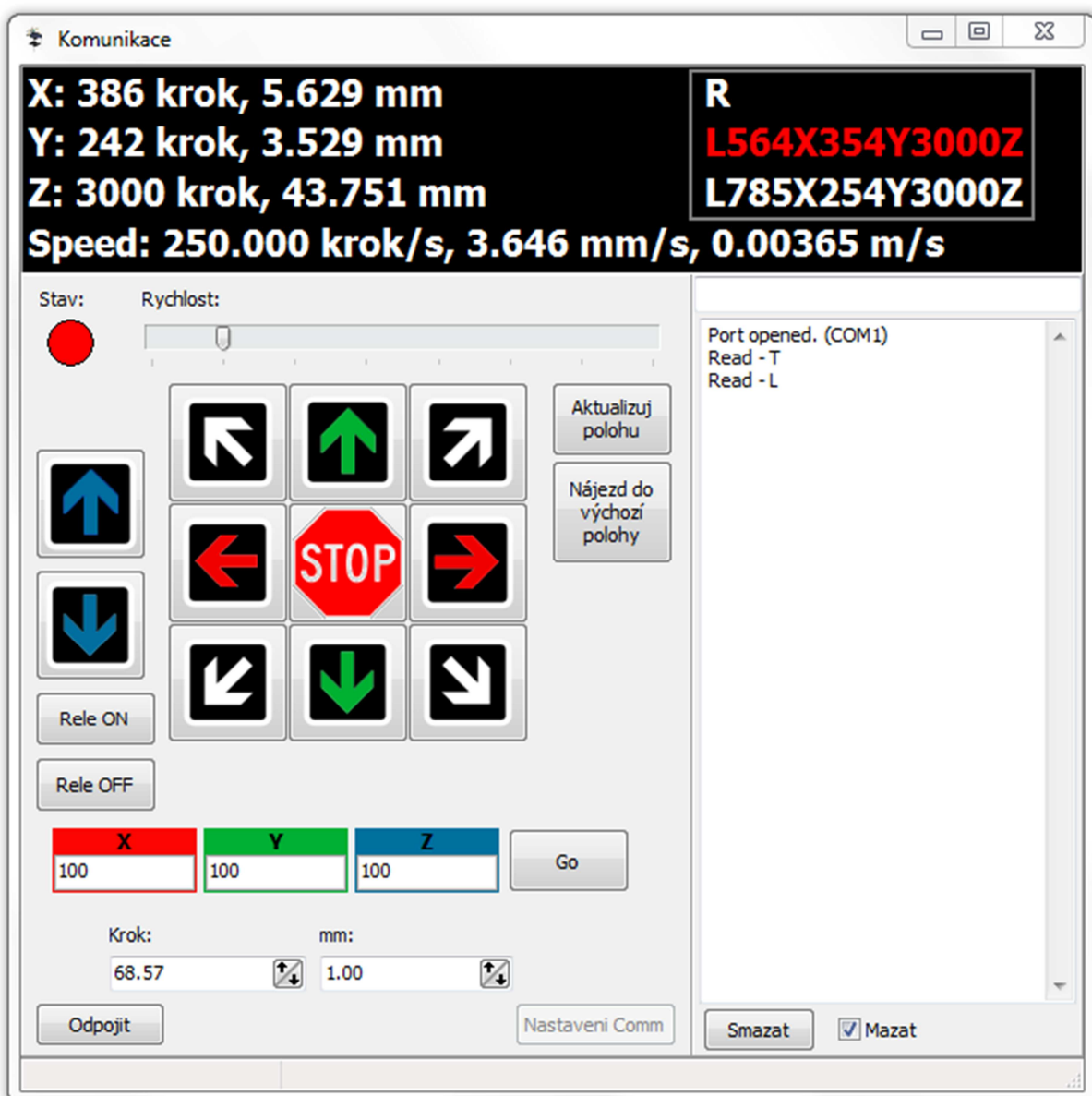
V programu je i funkce pro pohyb frézy ve všech třech osách současně.

9 Komunikace a řízení

V okně *Komunikace* se zobrazuje v horní části aktuální pozice frézy, rychlost pohybu, tři řádky z generovaného kódu (aktuální řádek, předchozí a následující).

V levé části je *Řídící panel* pro ruční ovládání CNC frézky. Zobrazuje se zde stav řídicího čipu (zda je například zaneprázdněn). Lze nastavit rychlost posuvu, manuálně ovládat pohyby frézy pomocí tlačítek, aktualizovat pozici v PC z čipu a Nájezd do výchozí polohy, zapnutí a vypnutí relé. Tlačítko *Go* spustí algoritmus pro „3D“ pohyb na zadané souřadnice.

V pravé části je pole pro odeslání dat do čipu (po stisknutí klávesy Enter), dále je pole pro příjem dat z řídicího čipu.

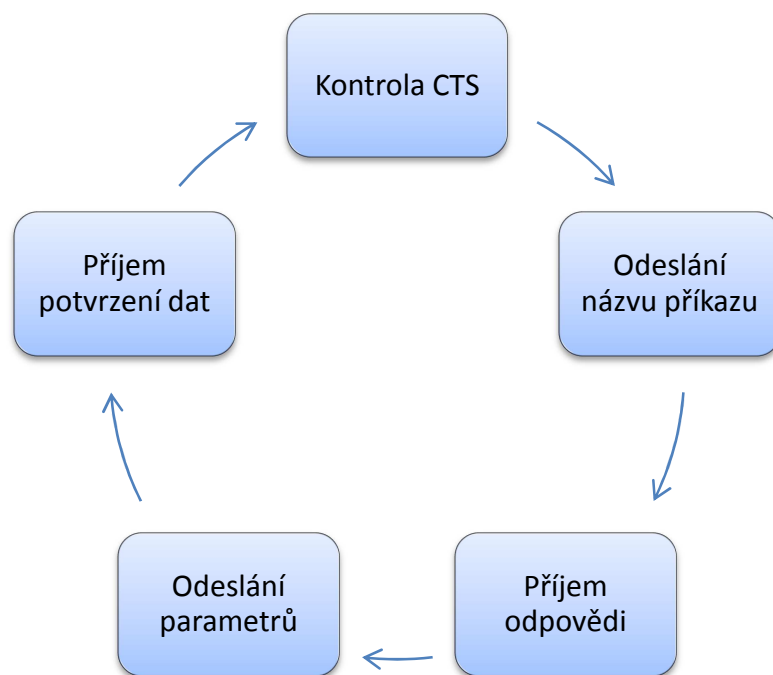


Obrázek 6 Komunikační okno

Pro komunikaci s elektronikou CNC stroje využívám trial verzi komponenty VaComm od společnosti TMS Software¹, která slouží k obsluze sériové linky RS-232. Sériová linka obsahuje 5 vstupních linek a 3 výstupní. Používám linky Receive Data (RXD), Transmit Data (TXD) a Clear to send (CTS). Signálem CTS mi čip oznamuje, že komunikační cesta je volná. Na formuláři jej znázorňuje kolečko s popisem Stav. Zelené ukazuje, že můžu posílat data a červené, že zařízení zpracovává nějaký příkaz a nemůže přijímat data, případně ukazuje, že je zařízení vypnuto nebo odpojeno.

Převodník kroků na milimetry a opačně. Stisknutím klávesy *Enter* se převede.

Stisknutím tlačítka *STOP* se *wstop* nastaví na *True*, aby se ukončil jakýkoli pohyb.



Obrázek 7 Cyklus komunikace

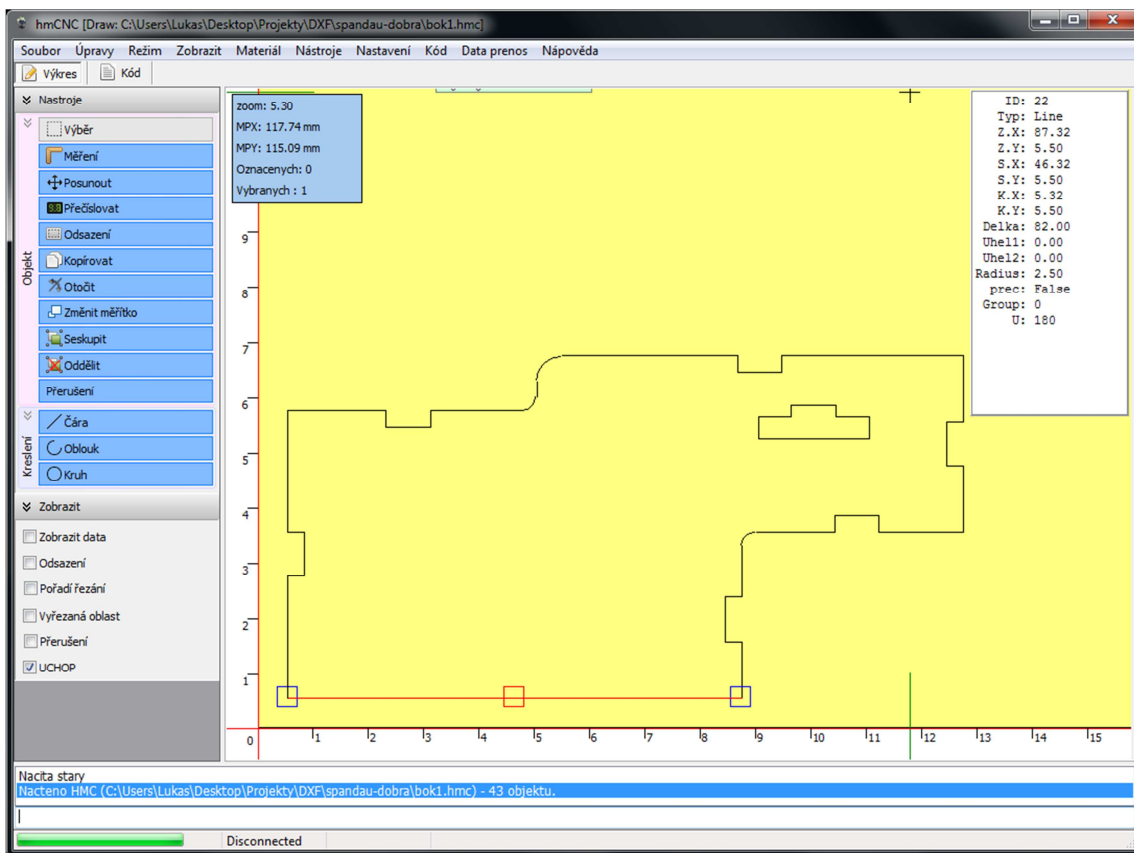
10 Aktualizace programu

Aktualizaci zabezpečuje má vlastní aplikace *update.exe*, která je vždy u samotného programu *hmCNC.exe*. Zjištění nové verze se provádí stažením souboru s verzí umístěnou na internetu, a v případě nové verze program spustí *update.exe* a sám se ukončí. Ten stáhne zabalenou novou verzi. Následně ji rozbalí a spustí ji. Aktualizace se provádí přes klasický http server (server IC.cz). Program *hmCNC* má možnost zjišťovat aktualizace při spuštění.

¹ Webová stránka: <http://www.tmssoftware.com/>

11 Uživatelské prostředí

Popis uživatelského prostředí – pracovní plocha, panely nástrojů, informační okna, hlavní nabídka, stavový řádek. Pracovní plocha představuje řezací plochu na CNC frézce s barevným zvýraznění materiálu.



Obrázek 8 Uživatelské prostředí

12 Ovládací prvky

12.1 Hlavní nabídka

Obsahuje položky *Soubor*, *Úpravy*, *Režim*, *Zobrazit*, *Materiál*, *Nástroje*, *Nastavení*, *Kód*, *Komunikace* a *Nápvěda*.

12.1.1 Soubor (výkres)

Nový

Vytvoří nový výkres a nastaví parametry do výchozích hodnot, kromě materiálu.

Otevřít

Otevře dialogové okno pro výběr souboru s výkresem z disku. Při otvírání nad rozpracovaným výkresem se zobrazí dotaz, zda původní výkres uložit či nikoliv. Podporované typy souborů: DXF (formát AutoCADu), HMC (vlastní formát pro ukládání dat), (MAT) (vlastní formát pro uložení materiálu)

Uložit

Uloží rozpracovaný výkres na disk. Podporované typy souborů: HMC, (MAT)

Uložit jako

Otevře dialogové okno pro výběr názvu a umístění souboru pro uložení na disk. Podporované typy souborů: HMC, (MAT)

Naposledy otevřené Výběr výkresu ze seznamu pěti naposledy otevřených výkresů.

Konec Ukončí celou aplikaci.

12.1.2 Úpravy

Úpravy zpět Vrábí změny ve výkresu nebo v kódu o krok zpět.

Úpravy vpřed Vrábí změny ve výkresu nebo v kódu o krok vpřed.

Vložit výkres Vložit vybraný výkres do aktuálního výkresu.

12.1.3 Režim

Výběr režimu práce

Editace Přepne na režim editace.

Simulace Přepne do prostředí simulace.

Řezání Přepne na režim řezání.

12.1.4 Zobrazit

Materiál Zobrazí na pracovní ploše nastavení materiálu.

Odsazení Zobrazí na pracovní ploše odsazení prvků – čárkovaně.

12.1.5 Materiál

Otevřít materiál Otevře dialogové okno pro výběr souboru z disku. Podporované typy souborů: MAT (vlastní formát)

Uložit materiál Uloží nadefinovaný materiál do souboru na disk.

Uložit materiál jako Otevře dialogové okno pro výběr názvu a umístění souboru pro uložení na disk. Podporované typy souborů: MAT

Naposledy otevřený Otevře naposledy otevřený materiál.

Vlastnosti Zobrazí okno s vlastnostmi materiálu.

12.1.6 Nástroje

Zobrazí se formulář *Nastavení nástrojů*.

12.1.7 Nastavení

Zobrazí okno s nastavením celé aplikace.

12.1.8 Kód

Generování kódu Vygeneruje nový kód.

<i>Otevřít</i>	Otevře dialogové okno pro výběr souboru z disku.
<i>Uložit jako ...</i>	Otevře dialogové okno pro výběr názvu a umístění souboru pro uložení na disk.
<i>Smazat</i>	Vymaže vygenerovaný kód.
<i>Smazat popisky</i>	Vymaže popisku z kódu, vše co je za //

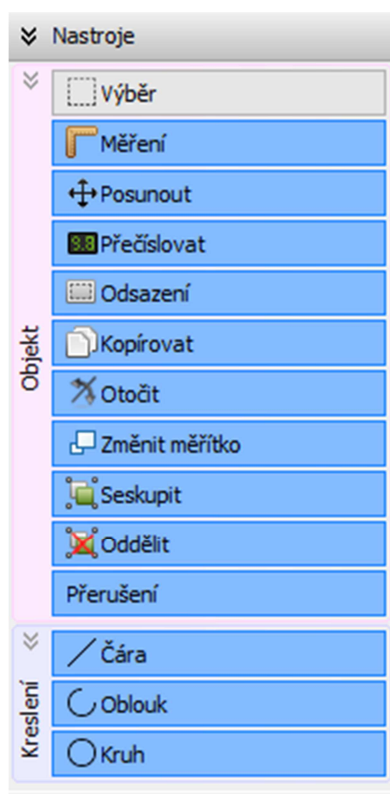
12.1.9 Komunikace

Zobrazí okno pro komunikaci s CNC frézku.

12.1.10 Nápověda

<i>Aktualizovat</i>	Ověří, zda jsou k dispozici nové verze programu. Případně stáhne nejnovější verzi z internetu.
<i>O programu</i>	Zobrazí okno O programu.

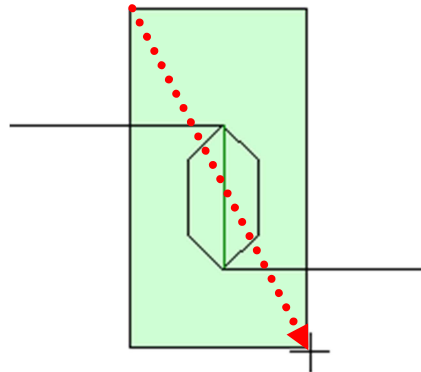
12.2 Panel Nástroje



Obrázek 9 Panel Nástroje

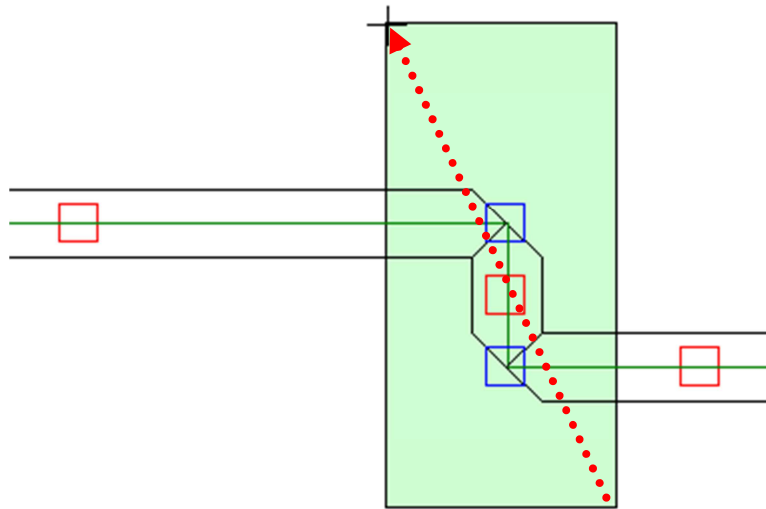
12.2.1 Výběr

Tento nástroj umožňuje výběr prvků (objektů) kliknutím nebo přetažením výběrovým obdélníkem.



Obrázek 10 Výběr objektů tažením

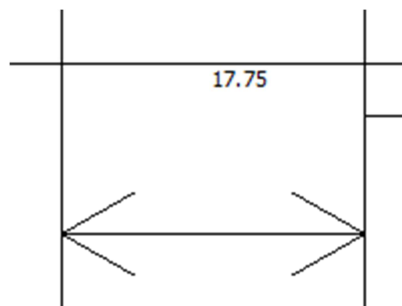
Tažením z levého horního rohu do pravého dolního rohu se označí pouze celé objekty v tomto obdélníku, tažením z pravého dolního rohu do levého horního rohu se označí objekty, které mají koncové body ve výběrovém obdélníku.



Obrázek 11 Výběr objektů za koncové body

12.2.2 Měření

Vybráním dvou bodů ukáže vzdálenost mezi těmito body. Při zapnuté možnosti „PŘICHYŤ“ se myš přichytí k nejbližšímu bodu. Vypočtená vzdálenost je v milimetrech (s přesností na setiny).



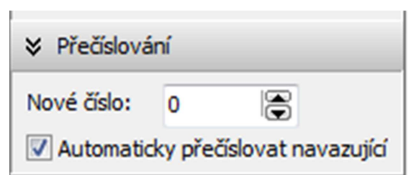
Obrázek 12 Měření

12.2.3 Posunout

Umožňuje posouvat vybrané objekty a vybírat další klinutím na ně. Má možnost „PŘIČHYŤ“.

12.2.4 Přečíslovat

Tato funkce slouží k určení pořadí v jakém se bude provádět řezání jednotlivých dílů. S tímto pořadím souvisí také následně vygenerování kódu pro řezání CNC frézku.

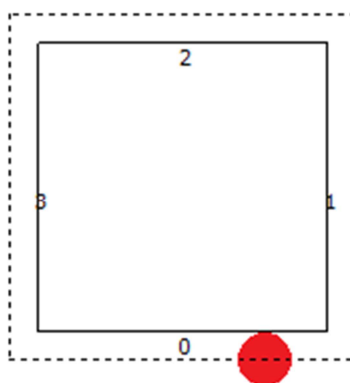


Obrázek 13 Panel Přečíslování

Kliknutím na prvek (čáru, oblouk...) se nastaví hodnota z buňky *Nové číslo*. Toto číslo se automaticky navýší o jedno. Při zapnuté možnosti *Automaticky přečíslovat navazující* se zjišťuje, zda na prvek, který jsme vybrali, navazuje další prvek. Tomu se pak nastaví číslo o jedno větší. A pokud i na tento prvek navazuje další prvek, také jemu se nastaví nové pořadí. To se opakuje do té doby, dokud funkce nedojde na konec smyčky nebo na aktuální prvek již žádný jiný nenavazuje.

12.2.5 Odsazení

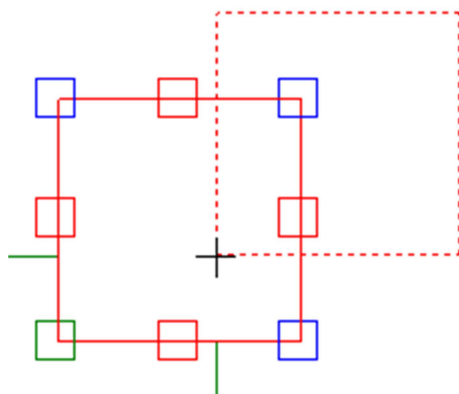
Vzhledem ke složitosti problematiky, je tato funkce v současnosti ve vývoji. Stručně řečeno: do programu načteme výkres s rozměry skutečného výrobku. Při řezání by byl bez tohoto odsazení výsledný díl zmenšen o poloměr frézy. Toto odsazení se počítá automaticky, jen se musí určit, na které straně čáry bude. V dalším vývoji se toto odsazení bude používat při generování kódu. Nyní je potřeba načítat výkres vytvořený již s tímto odsazením.



Obrázek 14 Odsazení

12.2.6 Kopírovat

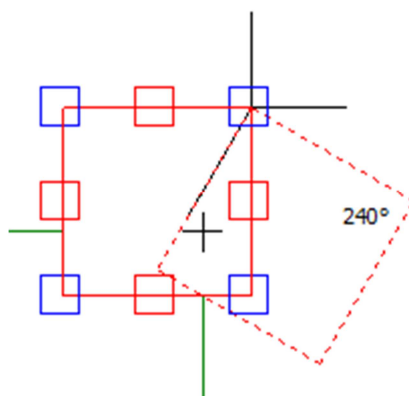
Vytvoření kopie vybraných prvků, objektů. Nejprve vybereme prvky (případně objekty) – označí se červeně. Kliknutím na toto tlačítko se zkopírují vybrané objekty do schránky aplikace (nepoužívá se schránka Windows). Dále kliknutím na ploše určíme nulový bod. Na pracovní ploše se zobrazuje (červeně a čárkovaně) náhled zkopírovaných objektů. Dalším kliknutím na určenou pozici se vytvoří kopírované objekty. Automaticky se jim přiřadí pořadí řezání tak, aby navazovalo na poslední. Je zde dostupná funkce „PŘICHYŤ“.



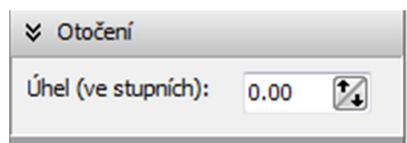
Obrázek 15 Kopírování

12.2.7 Otočit

Otočení vybraných prvků, objektů o určený úhel. Vybereme prvky (objekty) – označí se červeně. Kliknutím na pracovní plochu určíme bod otáčení, dále nastavíme úhel otočení zadáním přesného čísla nebo kliknutím na pracovní plochu. Aktuální úhel se zobrazuje pod bodem otáčení. Při přiblížení myši k tomuto bodu se úhel zarovná na celé číslo po 15 stupních. Dalším kliknutím otočení potvrdíme. Je zobrazován náhled otočení a při výběru bodu otáčení je dostupná funkce „PŘICHYŤ“.



Obrázek 16 Náhled otočení



Obrázek 17 Panel otočení

12.2.8 Změnit měřítko

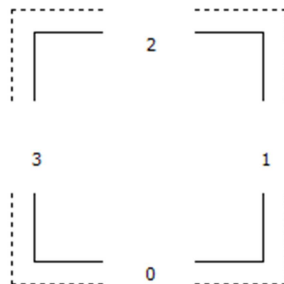
Při kliknutí se vybrané objekty zvětší nebo zmenší podle zadaného měřítka. Velikost se vztahuje vždy k poslední velikosti objektů.

12.2.9 Seskupit, Oddělit

Funkce *Seskupit* slouží k vytvoření skupin. Při výběru objektu ve skupině se vyberou všechny objekty v ní. Nemá vliv na použití přečíslování nebo jiné funkce. *Oddělení* zase zruší skupinu.

12.2.10 Přerušení

Funkce *Přerušení* je určena k tomu, aby vyřezávaný výrobek zůstával stále uchycen během řezání a nedocházelo k jeho uvolnění a případnému poškození. Po skončení řezání je výrobek patřičným nástrojem ručně oddělen.

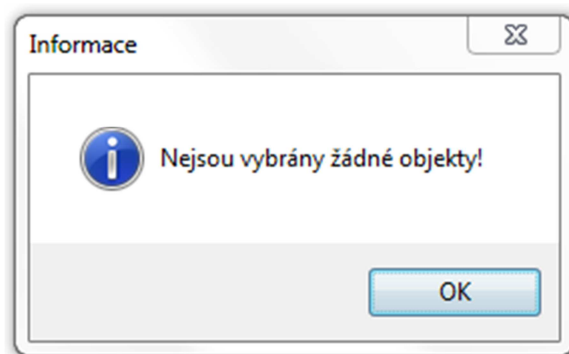


Obrázek 18 Ukázka přerušení 2 mm, s odsazením

12.2.11 Kreslení

Program umožňuje vytvářet objekty pomocí základních prvků - čáry, oblouky a kruhy. Rozměry jednotlivých prvků lze zadávat ručně nebo „kreslením“ na pracovní ploše. Tyto funkce jsou jen základní, protože program není primárně určen ke kreslení složitějších objektů.

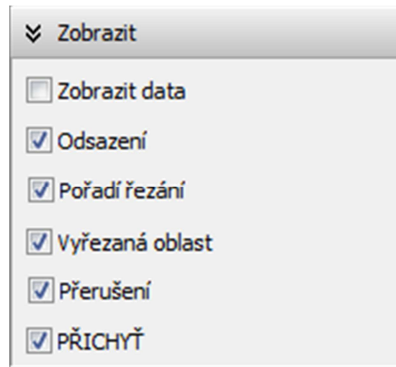
U funkcí *Kopírovat*, *Otočit*, *Změnit měřítko*, *Seskupit*, *Oddělit* musí být objekty vybrány předem, v opačném případě při použití těchto funkcí se zobrazí zpráva „Nejsou vybrány žádné objekty“.



Obrázek 19 Upozornění

Všechny funkce se zruší stisknutím klávesy ESC nebo kliknutím na jinou funkci.

12.3 Panel Zobrazit



Obrázek 20 Panel Zobrazit

Zobrazit data zobrazí na pracovní ploše informační tabulku se souřadnicemi všech objektů. Pohyb v tabulce umožňují klávesy PageUp a PageDown nebo kolečko myši.

Odsazení – zobrazí čárkovaně odsazení.

Pořadí řezání – vedle všech objektů se zobrazuje číslo s pořadím řezání.

Vyřezaná oblast – zobrazí na ploše, jak bude vypadat materiál po vyřezání. Počítá s průměrem nastavené frézy.

Přerušení – na čárách ukáže mezeru přerušení podle nastavené vzdálenosti.

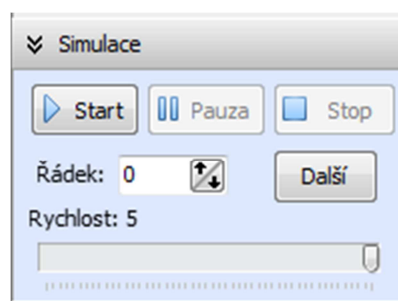
PŘICHYŤ – zapne / vypne tuto funkci.

12.4 Režimy

Aplikace pracuje ve třech režimech – editace, simulace a řezání.

V režimu *Editace* můžeme upravovat objekty, generovat nový kód pro řízení CNC frézky a úpravu tohoto kódu.

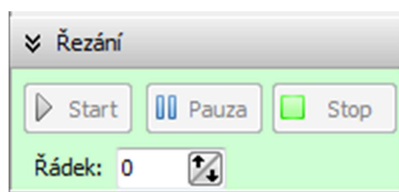
V režimu *Simulace* se skryje panel *Nástroje* a zobrazí se panel *Simulace*. Režim slouží k otestování a kontrole vygenerovaného kódu. Tlačítkem *Start* začne simulace na řádku, který se určí v buňce *Řádek*. Tlačítka *Pauza* a *Stop* slouží k pozastavení a zastavení simulace. Tlačítko *Další* je určeno k posunu na další řádek vygenerovaného kódu (skočí na další prvek). Na panelu je posuvník k nastavení rychlosti simulace.



Obrázek 21 Panel Simulace

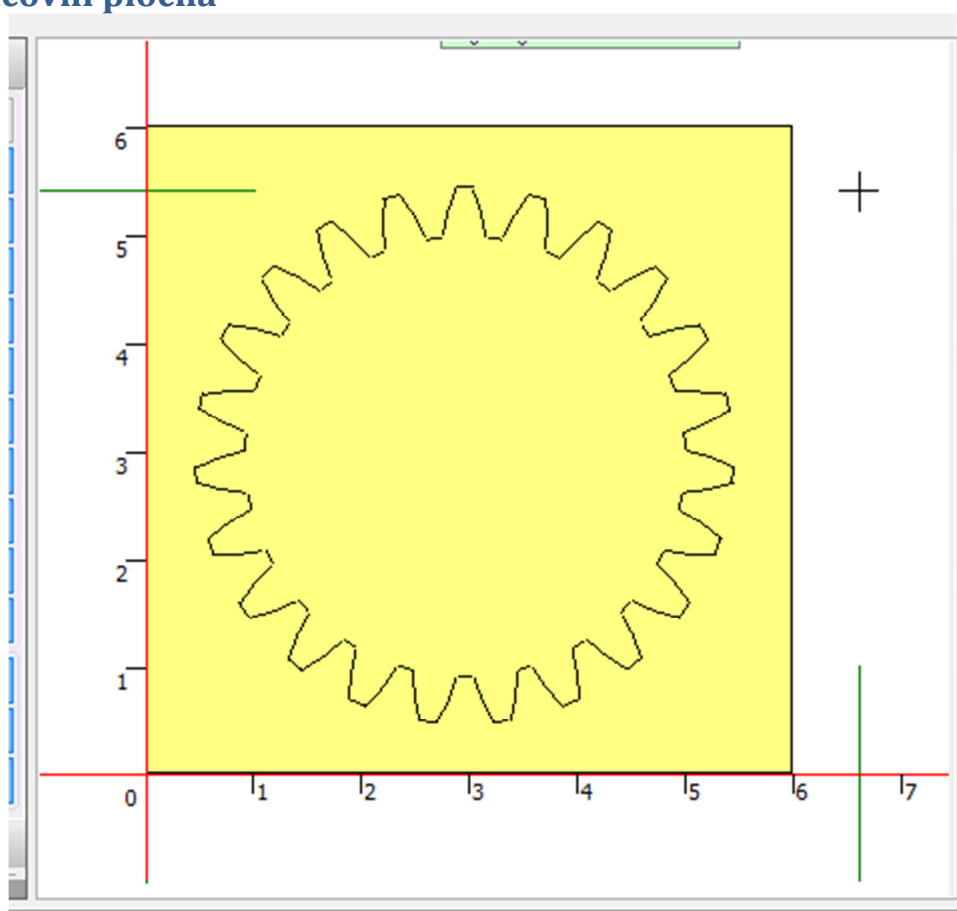
V tomto režimu je zablokována většina dostupných funkcí. Není možné vybírat objekty a vyvolat kontextovou nabídku. Držení levého tlačítka umožňuje pohyb pohledu.

Řezání je stejné jako *simulace*, jen s tím rozdílem, že už řídí samotný stroj CNC. Zobrazí se podobný panel s tlačítky Start, Pauza, Stop a buňka na určení řádku.



Obrázek 22 Panel Řezání

13 Pracovní plocha

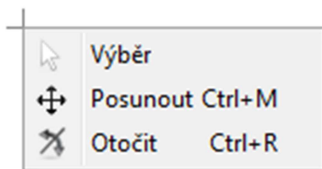


Obrázek 23 Pracovní plocha

Pracovní plocha slouží k zobrazení a úpravě řezaného předmětu. Zobrazuje se zde pravítka, typ vybraného materiálu, kurzor myši, řezaný předmět tvořený z čar a oblouků. Pak dále dodatečné informace jako je přerušení čar a čísla pořadí, podle kterých se generuje výsledný kód.

Kontextová nabídka

Kliknutím pravého tlačítka myši se otevře nabídka pro rychlé úpravy.



Obrázek 24 Kontextová nabídka

Zobrazuje se jen při režimu editace.

13.1 Zobrazení

Pro zobrazování veškerých prvků využívám komponent DelphiX, které využívají DirectX.

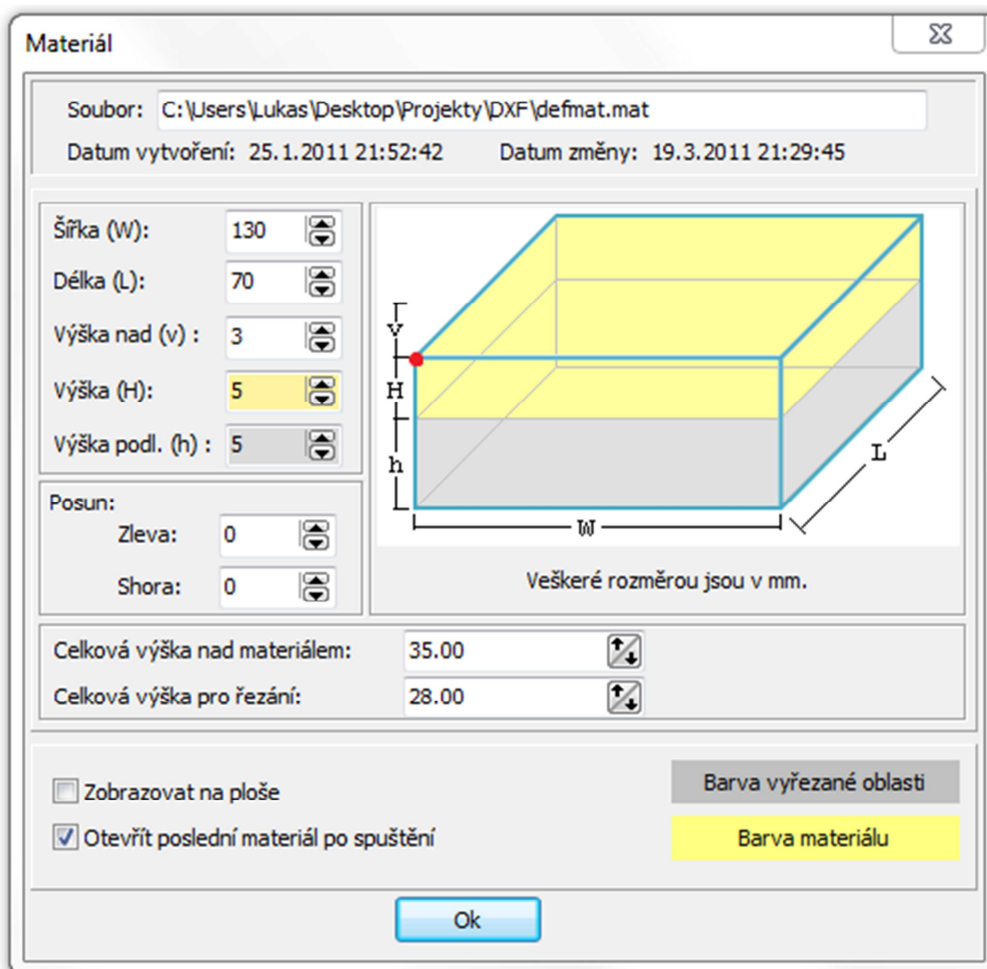
14 Použití klávesnice a myši

Ovládání pomocí klávesových zkratk urychluje práci s programem.

Klávesová zkratka	Popis
CTRL + N	Nový – Smaže všechny objekty.
CTRL + O	Otevřít – Vyvolá dialogové okno pro výběr souboru k otevření.
CTRL + S	Uložit – Uloží načtený výkres, v případě, že ještě nebyl uložen do souboru, vyvolá okno pro uložení.
CTRL + ALT + S	Uložit jako - Vyvolá okno pro uložení.
CTRL + Q	Ukončí celou aplikaci.
CTRL + Z	Vrátí změny zpět, na záložce Výkres vrátí změny ve výkresu a na záložce Kód změny v textu.
CTRL + Y	Vrátí změny vpřed, funkčnost jako při vrácení zpět.
F9	Přepne na režim Editace.
F10	Přepne na režim Simulace.
F11	Přepne na režim Řezání.
F12	Přepíná mezi záložkami Výkres a Kód.
F4	Otevře nastavení materiálu.
F5	Zobrazí materiál.
F6	Zobrazí odsazení.
CTRL + F9	Vygeneruje nový kód.

Kolečko myši je k zoomování na pracovní ploše, dvojklikem se pohled automaticky přiblíží nebo oddálí tak, aby byly vidět všechny prvky umístěné na pracovní ploše.

15 Nastavení materiálu



Materiál

Soubor: C:\Users\Lukas\Desktop\Projekty\DXF\defmat.mat
Datum vytvoření: 25.1.2011 21:52:42 Datum změny: 19.3.2011 21:29:45

Šířka (W): 130
Délka (L): 70
Výška nad (v): 3
Výška (H): 5
Výška podl. (h): 5

Posun:
Zleva: 0
Shora: 0

Celková výška nad materiálem: 35.00
Celková výška pro řezání: 28.00

Zobrazovat na ploše
 Otevřít poslední materiál po spuštění

Barva vyřezané oblasti
Barva materiálu

Ok

Veškeré rozměry jsou v mm.

Obrázek 25 Materiál - nastavení

Formulář slouží k nastavení rozměrů řezaného materiálu, podložky², umístění na řezací ploše a určení výjezdové a řezací výšky. Na formuláři se zobrazuje umístění souboru s aktuálně načteným materiálem, datum vytvoření a datum změny.

Výjezdová výška (*Celková výška nad materiálem*) se automaticky počítá sečtením výšky materiálu (H), výšky nad (v) a výšky podložky (h). Výška nad (v) je výška nad materiálem, kdy je vrchol frézy od horní strany materiálu přesně v této vzdálenosti.

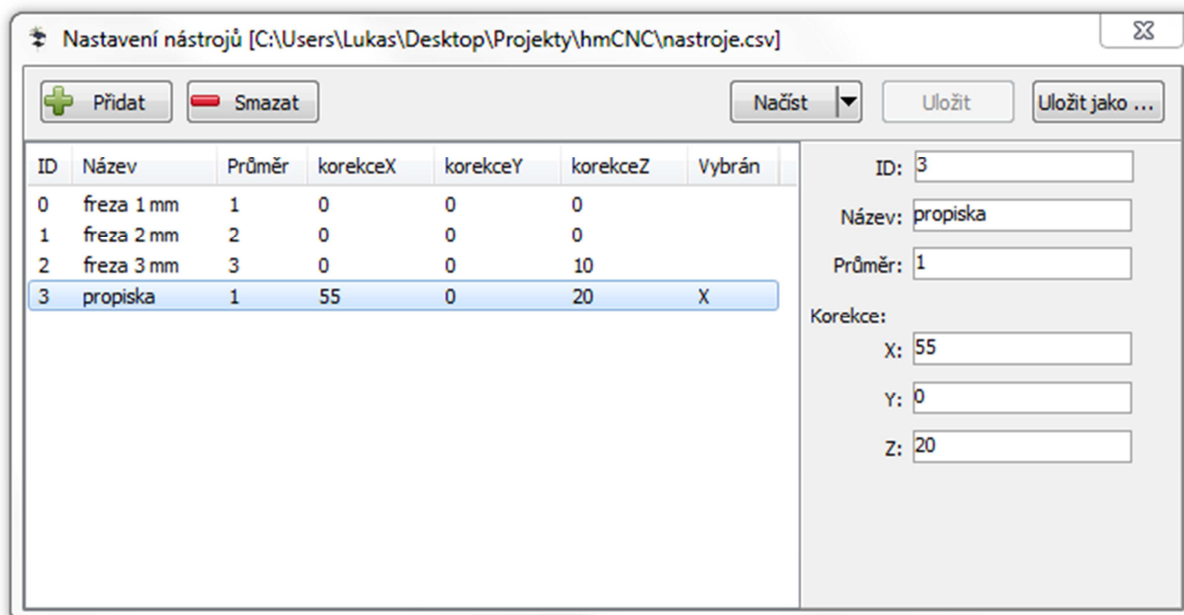
Řezací výška (*Celková výška pro řezání*) se automaticky počítá při změně výšky podložky. Je to Výška podložky vydělená dvěma.

Všechny tyto hodnoty se dají ručně přepsat na vlastní rozměr. Veškeré délky jsou v milimetrech.

Zaškrtnutím políčka *Zobrazovat na ploše* se bude na pracovní ploše zobrazovat materiál v barvě nastavené vpravo a podle rozměrů nahoře.

² Podložka je zde proto, aby se materiál při řezání prořízl v celé tloušťce.

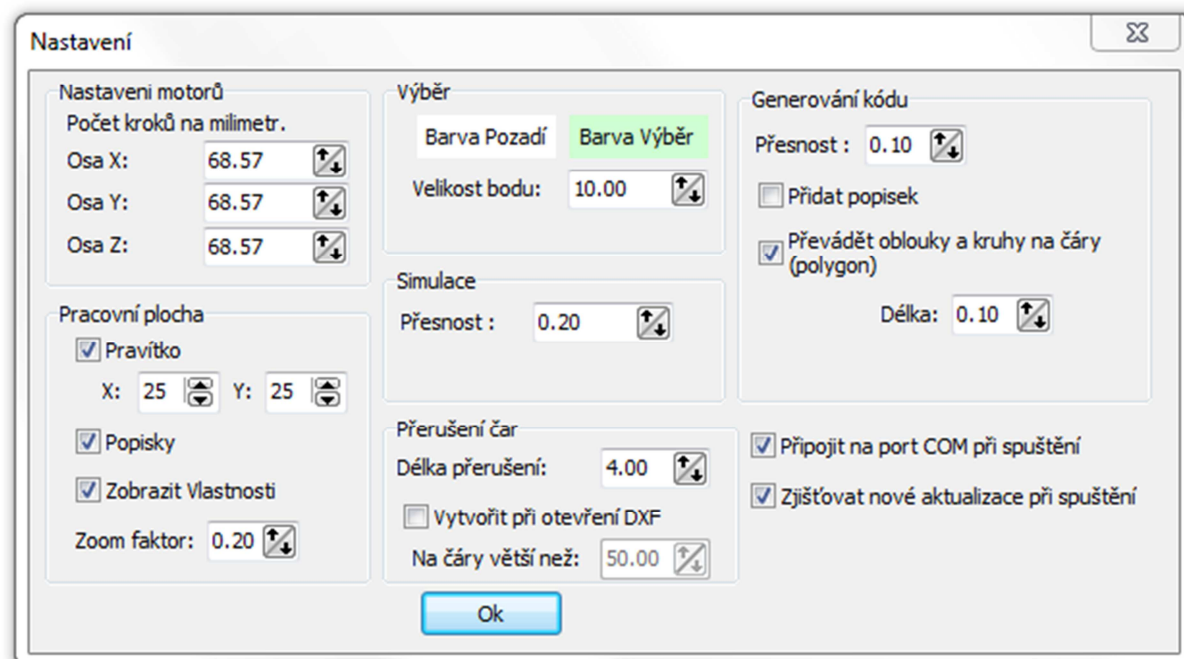
16 Nastavení nástroje



Obrázek 26 Nastavení nástrojů

Korekce vybraného nástroje se používají při generování kódu a výpočtu odsazení. Název je pouze informativní. Průměr a korekce jsou v milimetrech.

17 Nastavení aplikace



Obrázek 27 Nastavení aplikace

17.1 Nastavení motorů

Zadáva se počet kroků motoru na milimetr ve 3 osách.

17.2 Pracovní plocha

17.2.1 Právítko

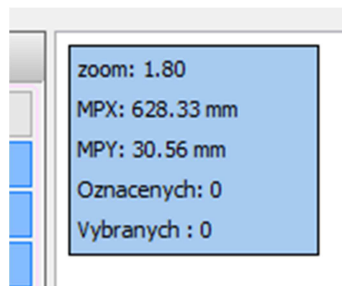
Zaškrtnutím se bude na ploše zobrazovat pravítko začínající v nule. Právítko je po centimetrech do vzdálenosti zadané v buňkách X a Y.



Obrázek 28 Právítko - nastavení

17.2.2 Popisky

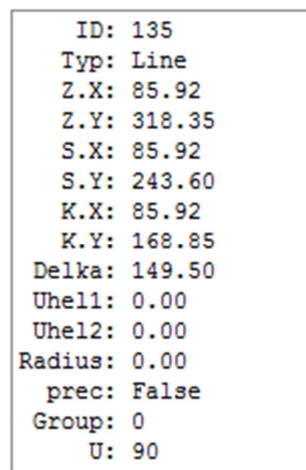
Zaškrtnutím se bude na ploše zobrazovat v levém horním rohu informační tabulka s velikostí aktuálního zoomu a pozicí myši na ploše.



Obrázek 29 Informace o zobrazení

17.2.3 Zobrazit vlastnosti

V pravém horním rohu se zobrazí tabulka s vlastnostmi vybraného prvku.



Obrázek 30 Vlastnosti prvku

17.2.4 Zoom faktor

Zadaná hodnota udává změnu přiblížení při otočení kolečka myši.

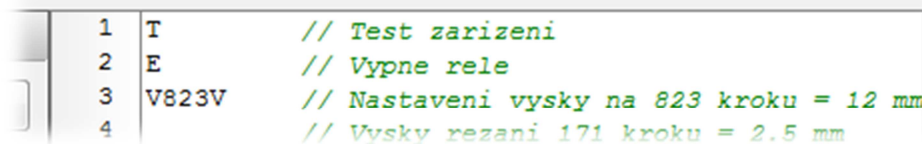
17.3 Přerušení čar

Délka přerušení určuje, jak velká bude mezera přerušení. Zaškrtnutím *Vytvořit při otevření DXF* se povolí automatické vytvoření přerušení na čáry delší, než je hodnota zadaná v buňce *Na čáry větší než*. Funkce funguje jen při otevření souboru DXF.

17.4 Generování kódu

Přesnost generování kódu se používá při velmi krátkých čarách.

Přidat popisek – k vygenerovanému kódu se vytvoří popisek každého řádku. Skládá se ze dvou pravých lomítek a následného textu. V kódu se zobrazuje zeleně kurzívou.



```
1 T // Test zařízení
2 E // Vypne rele
3 V823V // Nastavení vysky na 823 kroku = 12 mm
4 // Vysky rezání 171 kroku = 2.5 mm
```

Obrázek 31 Ukázka popisků v kódu

Převádět oblouky a kruhy na čáry (polygon) – při generování kódu se oblouky a kruhy budou převádět na čáry. Délka těchto čar je určena v buňce *Délka*. Výsledný kód je vždy větší.

17.5 Připojit na port COM při spuštění

Automaticky se připojí na sériovou linku při otevření okna *Komunikace*.

17.6 Zjišťování aktualizací při spuštění

Při spuštění aplikace zjistí, zda jsou dostupné aktualizace.

18 Strojová část

Samotný stroj CNC jsme zkonstruovali s kolegou Liborem Dandou. Práci jsme si rozdělili, já jsem vymyslel a vyrobil elektroniku, vytvořil software a on postavil samotný stroj. Stroj pracuje s přesností jedné desetiny milimetru. Řídící elektronika je založená na čipu ATmega32, který řídí krokové motory. Celý kód v čipu řídící elektroniky jsem naprogramoval v jazyce C. Protože nestuduji obor informatika, ale strojařinu, musel jsem se vše učit z internetu.

Nutné změny provedené v závěru práce

Původní řešení počítalo s tím, že veškeré výpočty na ovládání motorů se budou provádět přímo v čipu stroje, čehož bylo také dosaženo. Dalším vývojem se ovšem toto řešení ukázalo jako nepraktické – problémy nastaly při řezání oblouků. Kód v čipu jsem tedy doplnil o přímé řízení motorů.

19 Závěr

Aplikace je použitelná na **amatérské CNC stroje**. Veškeré algoritmy použité v programu jsou mé vlastní. V této práci jsem se naučil používat matematiku v programování, použití dynamických polí, použití Memorystreamu (způsob načítání, ukládání dat), tvořit vlastní algoritmy, práci s grafikou a velkou část o programování mikročipů Atmel a jejich zapojení.

20 Další vývoj

Další vývoj programu bude zaměřen především na dokončení funkce „Odsazení“. Dále bych chtěl rozšířit import o načítání prvku typu křivka (*spline*).

Neméně důležitým krokem bude přechod z vlastního kódu na již zavedený programovací jazyk G-kód³, pro širší využití programu při řízení jiných zařízení.

V další verzi programu by měla být možnost práce s jednoduchými 3D objekty.

21 Mé další projekty

- Robotické rameno – účast na 32. ročníku SOČ, obor 9. strojírenství, postup do celostátního kola
- Quadcopter – vývoj softwaru a elektroniky (další programovací jazyk) – ve vývoji
- Domácí CNC frézka – předchozí verze

22 Slovník pojmů

- **CNC** (computer numerical controlled) je stroj plně řízený počítačem.
- **Postprocesor** zajišťuje překlad souřadnicových dat do formátu, který stroj zpracuje
- **Fréza** je vícebřitý obráběcí nástroj používaný k frézování, třískovému obrábění kovů i jiných materiálů. Obráběcí stroj, který ho používá, nazýváme **frézka**.

³ <http://cs.wikipedia.org/wiki/G-kód>

23 Seznam obrázků

Obrázek 1 Náhled programu	9
Obrázek 2 Schéma procesu	10
Obrázek 3 Výběrový polygon	12
Obrázek 4 Schéma generování kódu	14
Obrázek 5 Kreslení přímky.....	16
Obrázek 6 Komunikační okno.....	17
Obrázek 7 Cyklus komunikace.....	18
Obrázek 8 Uživatelské prostředí	19
Obrázek 9 Panel Nástroje	21
Obrázek 10 Výběr objektů tažením.....	22
Obrázek 11 Výběr objektů za koncové body.....	22
Obrázek 12 Měření.....	23
Obrázek 13 Panel Přechíslování	23
Obrázek 14 Odsazení.....	23
Obrázek 15 Kopírování	24
Obrázek 16 Náhled otočení.....	24
Obrázek 17 Panel otočení	24
Obrázek 18 Ukázka přerušování 2 mm, s odsazením	25
Obrázek 19 Upozornění.....	25
Obrázek 20 Panel Zobrazit	26
Obrázek 21 Panel Simulace	26
Obrázek 22 Panel Řezání	27
Obrázek 23 Pracovní plocha	27
Obrázek 24 Kontextová nabídka	28
Obrázek 25 Materiál - nastavení	29
Obrázek 26 Nastavení nástrojů	30
Obrázek 27 Nastavení aplikace	30
Obrázek 28 Pravítka - nastavení.....	31
Obrázek 29 Informace o zobrazení	31
Obrázek 30 Vlastnosti prvku	31
Obrázek 31 Ukázka popisků v kódu.....	32

24 Seznam příloh

Příloha 1 Ukázka funkce pro pohyb v souřadnicích XY

Příloha 2 Část algoritmu pro generování

Příloha 3 Struktura datového pole

Příloha 1 Ukázka funkce pro pohyb v souřadnicích XY

```
procedure Tf_data_prenos.NewMove2D(aX,aY,aZ : Integer);
var dx, dy, x, y, xp, yp      : Integer;
    m,b : Double;
    Z : TPoint;
begin
  Z.X := pX;
  Z.Y := pY;
  dx := (aX) - (Z.x);
  dy := (aY) - (Z.y);
  x := pX;
  y := pY;
  xp := pX;
  yp := pY;

  while pZ <> aZ do
  begin
    if pZ < az then
      StepMotor(6, False); // motor v ose Z - dolů
    if pZ > az then
      StepMotor(5, False); // motor v ose z + nahorů

    sleep(trckbr1.Position);
    Application.ProcessMessages;
    if wstop then Break; // zastavení pohybu
  end;

  if not wstop then
  if (pZ <> aZ) then // kontrola se zpětnou smyčkou
    NewMove2D(aX, aY, aZ);

  PosliChar('0'); // vypne všechny porty (s motory) na čipu
  Ready; // čekej na CTS = True

  if (abs(dx) > abs(dy)) then // zjištění řídicí osy - v ose X
  begin
    m := dy / dx;
    b := Z.y - m*Z.x;

    if dx < 0 then dx := -1 else dx := 1;

    while px <> ax do
    begin
      if wstop then Break;

      x := x + dx;
      y := Round(m*x + b);

      if dx = 1 then
        StepMotor(4, False); // motor v ose x +
      if dx = -1 then
        StepMotor(3, False); // motor v ose x -

      if yp <> y then
      begin
        if y > yp then
          StepMotor(2, False); // motor v ose y -
        if y < yp then
          StepMotor(1, False); // motor v ose y +
        yp := y;
      end;
    end;
  end;
```

```

        end;
        sleep(trckbr1.Position);
        Application.ProcessMessages;
    end;
end
else
if (dy <> 0) then // v ose Y
begin
    ..
end;

if not wstop then
if (pX <> aX) or (pY <> aY) then // kontrola se zpětnou smyčkou
    NewMove2D(aX, aY, aZ);

if Cutting = Stop then // při skončení řezání vypni motory a vrať polohu
begin
    PosliChar('P');
    PosliChar('0');
end;
wstop := False;
end;
end;

```

Příloha 2 Část algoritmu pro generování

```
procedure TfMain.btn_gen_codeClick(Sender: TObject);
var I, I2 : integer;
    osaZ : integer;
    rozsah : Double;
begin
    if not KontorlaPrecislovani then
    begin
        Application.MessageBox('Zkontroluj přečíslování! Generování
zastaveno.', 'hmCNC - Error', MB_ICONERROR + MB_OK);
        Exit;
    end;

    if Length(Objekty) = 0 then
    begin
        Application.MessageBox('Nejsou žádné prvky na výkrese! Generování
zastaveno.', 'hmCNC - Error', MB_ICONERROR + MB_OK);
        Exit;
    end;

    if (VysNajezd = 0) {or (VysRezani = 0)} then
    begin
        Application.MessageBox('Výška nájezdová je 0! Generování zastaveno.',
'hmCNC - Error', MB_ICONWARNING + MB_OK);
        Exit;
    end;

    if (VysRezani = 0) then
    begin
        if Application.MessageBox('Výška řezací je 0, pokračovat?', 'hmCNC -
Varování', MB_ICONWARNING + MB_YESNO) = 7 then
            Exit;
    end;

    MemoCodeBuffer := TStringList.Create;
    MemoCodeBuffer.Clear;

    rozsah := fnastaveni.jvsp_gencode_presnost.value;
    ProgressBar1.Position := 0;
    ProgressBar1.Max := Length(Objekty);
    MemoCode.Clear;
    AddLineToMemoCode('T','', 'Test zarizeni');
    AddLineToMemoCode('P','', 'Zjistí pozici');
    AddLineToMemoCode('E','', 'Vypne rele');

    AddLineToMemoCode('V',ITS(VysNajezd)+'V', 'Nastaveni vysky na
'+ITS(VysNajezd)+' kroku =
'+FloatToStr(fMaterial.jvsp_nadMaterialem.Value)+' mm');
    AddLineToMemoCode('','', 'Vysky rezani '+ITS(VysRezani)+' kroku =
'+FloatToStr(fMaterial.jvsp_Rez.Value)+' mm');

    I2 := 0;
    while I2 <> Length(Objekty) do
    begin
        I := bc2bc(I2);
        ProgressBar1.Position := ProgressBar1.Position + 1;

        if Objekty[I].typ = ttLine then
        begin
```

```

        if not ( dblSeRovna( Objekty[I].K , VratObj(I2-1).Z, rozsah) or
dblSeRovna( Objekty[I].K , VratObj(I2-1).K, rozsah)) and
        not ( dblSeRovna( Objekty[I].Z , VratObj(I2-1).Z, rozsah) or
dblSeRovna( Objekty[I].Z , VratObj(I2-1).K, rozsah)) then
        begin
            osaZ := VysNajezd;//fnastaveni.se_z_up.Value;
            if dblSeRovna( Objekty[I].Z , VratObj(I2+1).Z, rozsah) or
dblSeRovna( Objekty[I].Z , VratObj(I2+1).K, rozsah) then
                AddLine(Objekty[I].K.X, Objekty[I].K.Y, osaZ , ITS(I2)+' Line
najezd na Konec')
            else
                AddLine(Objekty[I].Z.X, Objekty[I].Z.Y, osaZ , ITS(I2)+' Line
najezd na Zacatek');
            end;

        if I2 = 0 then
            AddLineToMemoCode('R','',' 'zapne rele');

        if ( dblSeRovna( Objekty[I].Z , VratObj(I2+1).Z, rozsah) or
dblSeRovna( Objekty[I].Z , VratObj(I2+1).K, rozsah) ) or
        ( dblSeRovna( Objekty[I].K , VratObj(I2-1).Z, rozsah) or
dblSeRovna( Objekty[I].K , VratObj(I2-1).K, rozsah) ) then
        begin
            if Objekty[I].Preruseni then
                begin
                    Objekty[I].PreruseniBod1 := GetBodzRAndA2(Objekty[I].S,
fnastaveni.jvsp_velikostPreruseni.Value, -DegToRad(Objekty[I].U));
                    Objekty[I].PreruseniBod2 := GetBodzRAndA2(Objekty[I].S, -
fnastaveni.jvsp_velikostPreruseni.Value, -DegToRad(Objekty[I].U));

                    osaZ := VysRezani;
                    AddLine(Objekty[I].PreruseniBod1.X, Objekty[I].PreruseniBod1.Y,
osaZ , ITS(I2)+' Line na preruseni 1');

                    osaZ := VysNajezd;
                    AddLine(Objekty[I].PreruseniBod2.X, Objekty[I].PreruseniBod2.Y,
osaZ , ITS(I2)+' Line na preruseni 2');

                    osaZ := VysRezani;
                    AddLine(Objekty[I].Z.X, Objekty[I].Z.Y, osaZ , ITS(I2)+' Line na
Zacatek');
                end
            else
                begin
                    osaZ := VysRezani;
                    AddLine(Objekty[I].Z.X, Objekty[I].Z.Y, osaZ , ITS(I2)+' Line na
Zacatek');
                end;
            end
        else
            begin
                if Objekty[I].Preruseni then
                    begin
                        Objekty[I].PreruseniBod1 := GetBodzRAndA2(Objekty[I].S, -
fnastaveni.jvsp_velikostPreruseni.Value, -DegToRad(Objekty[I].U));
                        Objekty[I].PreruseniBod2 := GetBodzRAndA2(Objekty[I].S, -
fnastaveni.jvsp_velikostPreruseni.Value, -DegToRad(Objekty[I].U));

                        osaZ := VysRezani;
                        AddLine(Objekty[I].PreruseniBod1.X, Objekty[I].PreruseniBod1.Y,
osaZ , ITS(I2)+' Line na preruseni 1');
                    end
                end
            end
        end
    end

```

```

        osaZ := VysNajezd;
        AddLine(Objekty[I].PreruseniBod2.X, Objekty[I].PreruseniBod2.Y,
osaZ , ITS(I2)+' Line na preruseni 2');

        osaZ := VysRezani;
        AddLine(Objekty[I].K.X, Objekty[I].K.Y, osaZ , ITS(I2)+' Line na
Konec');
    end
    else
    begin
        osaZ := VysRezani;
        AddLine(Objekty[I].K.X, Objekty[I].K.Y, osaZ , ITS(I2)+' Line na
Konec');
    end;
    end;
    end;
    ...
    end;
End;

```

Příloha 3 Struktura datového pole

```
TObjekt = record
  PozRez   : integer;           // pozice pro rezani;
  Typ      : TTyp;
  V,      // vrchol oblouku
  Z,      // zacatecni bod
  S,      // stredovy bod (pocita se)
  K       // koncovy bod
  : TDbfPoint;
  Zmar,   // je oznaceny zacatecni bod
  Smar,   //          stredovy bod
  Kmar,   //          koncovy bod
  Zsel,   // je vybrany zacatecni bod
  Ssel,   //          stredovy bod
  Ksel    //          koncovy bod
  : Boolean;
  Delka,  // delka cary nebo oblouku
  OsaZ1,OsaZ2, // osa Z
  Radius,
  Angle1,
  Angle2  : Double;
  Marked,
  Selected : boolean;
  Group    : Integer;
  DX1,DY1, // pri vyberu je to rozdil mezi mysi a bodem,
  DX2,DY2, // to se pak pricte k bodu aby
  DX3,DY3:Double; // mela mys porad stejnou vzdalenost od bodu,
  // nastavuje se v MouseDown
  Precislovane : boolean;
  Color : TColor;
  Color2 : TColor;
  B : array [0..5] of TDbfPoint;
  U : integer;
  OVZ, OVK : Double;//odsazeni vzdalenost
  OS, OZ, // odsazeni zacatek
  OK : TDbfPoint; // odsazeni konec}
  copy : boolean;
  Rezat : boolean; // rika jestli se pouzije pri generovani kodu
  Freza : integer; // ID frezy z nastroju
  Preruseni : Boolean; // preruseni cary presne uprostred
  PreruseniBod1,
  PreruseniBod2 : TDbfPoint;
  Priorita: integer;
  Locked : Boolean;
end;
```