

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

STUDIJNÍ PROGRAM: B2646 - INFORMAČNÍ TECHNOLOGIE

STUDIJNÍ OBOR: 1802R007 - INFORMAČNÍ TECHNOLOGIE

Konvertor CAM dat

Bakalářská práce

Autor:
Vedoucí práce:
Konzultant:

Milan Lochovský
Ing. Leoš Petržílka
prof. Ing. Zdeněk Plíva, Ph.D.

V Liberci 15. 5. 2012

Zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 15. 5. 2012

Podpis: Milan Lochovský

Poděkování

Chtěl bych poděkovat panu profesoru Plívovi a panu inženýru Petržílkovi za umožnění provedení této práce. Cením si jejich přístupu a času, kterým byly ochotni se na této práci podílet. Během práce jsem se seznámil blíže s výrobou DPS a naučil se nové věci při programování a za tuto možnost bych chtěl také poděkovat, protože mi je tento problém blízký a rád se seznámím s něčím novým. Během práce jsem se také obracel na pana Hradeckého, kterému také děkuji za poskytnuté informace ohledně programování, a fungování CNC strojů pro přípravu DPS.

Abstrakt

Cílem této práce je vytvořit funkční program pro převod Excellon dat a HP-GL dat do výstupního formátu, kterým jsou G-kódy. V práci se zabýváme problémem formátu samotných a jejich možnosti převodu. Program najde uplatnění při výrobě desek plošných spojů v laboratoři PCBLab na Technické Univerzitě v Liberci. Bude zde sloužit právě k převodu dat Excellon, které vznikají v programech pro výrobu DPS a slouží jako vstupní data pro vrtací stanici. A formátu HP-GL, který vzniká jako popis kontury desek. Oba tyto formáty je potřeba převést do formátu pro CNC stroj, tedy ISO 6983, aby bylo možno DPS vyrábět za pomoci CNC stroje. Výsledkem práce tedy je program, který tento převod umožní a bude pro uživatele jednoduchý na každodenní využívání. Samotný program byl vyroben pro požadavky laboratoře, takže umožňuje nastavení, které je potřeba při výrobě právě v této laboratoři.

Klíčová slova

Převodník, DPS, G-kódy, HP-GL, Excellon, CNC

Abstract

The aim of this paper is to create a purposive programme which converts excellon data and HP-GL data into the original G-code format. In the paper, we concern with the issue of the formats and the possibilities to convert them. The programme can be used when manufacturing printed circuit boards in a laboratory PCBLab at the Technical University Liberec. It will be used to convert Excellon data which are created in programmes for PCB production and it serves as input data for drilling station and A-format HP-GL which is created as a kontury desek description. Both the formats have to be converted into G-codes to enable DPS production with support of CNC machine. The result of the paper is a programme which enables this conversion and is easy for the user to employ in day-to-day work. The programme itself was created for laboratory purposes, so it makes possible to set it in accordance with the requirements in the production.

Keywords

Converter, PCB, G-code, HP-GL, Excellon, CNC

Obsah

Prohlášení	3
Poděkování.....	4
Abstrakt	5
Klíčová slova.....	5
Abstract	6
Keywords.....	6
Seznam zkratk a termínů	9
1. Úvod	10
1.1 Převod Excellon do ISO 6983	10
1.2 Převod HPGL do ISO 6983.....	10
2. Obdobné aplikace	12
3. Formáty dat.....	15
3.1 ISO 6983	15
3.1.1 Formát dat	16
3.1.2 Použité instrukce v programu	16
3.2 Excellon	18
3.2.1 Formát dat	18
3.2.2 Použité instrukce v programu	19
3.3 HP-GL	20
3.3.1 Formát dat	20
3.3.2 Podporované příkazy	21
4. Implementace.....	24
4.1 Použité technologie	24
4.1.1 Microsoft Visual Studio	24
4.2 Grafické rozhraní programu	25
4.2.1 Hlavní okno.....	25
4.2.2 Záložky „Excellon do G-kódů“ a „HPGL do G-kódů“	26
4.2.3 Nastavení	27
4.2.4 Náhled	30
4.2.5 Náhled obrázku	31

4.3	Programové třídy	32
4.3.1	Třída HPGLtoG.....	32
4.3.2	Třída HPGLfce.....	32
4.3.3	Třída ExcellonToG.....	33
4.3.4	Třída NastaveniStorage	33
4.3.5	Třída NastaveniData	34
4.4	Spuštění programu	34
4.5	Převod Excellon dat do G-kódů	34
4.5.1	Načtení dat	34
4.5.2	Převod mezi formáty	34
4.5.3	Editace dat	35
4.5.4	Uložení dat	35
4.6	Převod HPGL dat do G-kódů	35
4.6.1	Načtení dat	35
4.6.2	Převod mezi formáty	36
4.6.3	Kontrola dat	36
4.6.4	Obrázkový náhled	36
4.6.5	Uložení dat	36
5.	Praktické požití	37
5.1	Převod v programu Eagle	37
5.1.1	Generování Excellon dat.....	37
5.1.2	Generování obrysových dat.....	38
5.2	Převod do G-kódů.....	39
5.2.1	Převod Excellon dat.....	39
5.2.2	Převod HP-GL dat	40
5.3	Náhled výstupních dat	42
6.	Závěr.....	43
	Literatura	44

Seznam zkratek a termínů

DPS – deska plošných spojů

Base64 – datový formát zobrazující binární data pomocí ASCII znaků

Serializace – převod instance do datového formátu, který je možno uložit

Deserializace – opačný proces serializace

CNC – číslicově řízený stroj určený k obrábění

GUI – grafické uživatelské rozhraní

1. | Úvod

Cílem práce je vytvořit program, který bude sloužit k převodu vstupních dat formátu Excellon (data určená k vrtání desek plošných spojů), a nebo HP-GL (vektorová data, využívaná například k popisu kontury desek plošných spojů) do formátu ISO 6983, který slouží k řízení CNC strojů. Jako zdroj dat by měl sloužit program Eagle (program pro návrh desek plošných spojů) tak, aby je bylo možno využít na CNC stroji, který umí zpracovat data v G-kódech. Program tedy bude rozdělen na dvě části, kde každá z částí bude umět převést jeden formát vstupních dat.

Konkrétní využití by tento program měl nalézt v laboratoři „PCB-Lab“ na Technické univerzitě v Liberci, kde by měl sloužit k převodu vrtacích a gravírovacích dat z programu Eagle, který generuje Excellon data pro vrtání a HPGL data pro gravírování desek plošných spojů a následné jejich výrobě na CNC stroji.

1.1 Převod Excellon do ISO 6983

Tato část programu se postará o převod Excellon dat do ISO 6983. Program tedy musí načíst vstupní Excellon data a vytvořit z nich výstup v ISO 6983. V běžné praxi se setkáme spíše s pojmem G-kódy.

Musíme se postarat o načtení vstupních dat, jejich převod do výstupního formátu a možnosti uložení převedených dat do souboru. V případě, že se nepovedlo některý řádek převést, měl by to program uživateli oznámit a umožnit mu případně do převodu zasáhnout a upravit data ručně.

Další neméně důležitou částí je nastavení, kde by uživatel měl mít možnost nastavit hlavičku a patičku, která se vloží na začátek, respektive konec převedených dat. Zároveň je důležité umožnit nastavení, jak se bude program zachovávat v případě instrukcí vrtání, kde G-kódy umožňují více variant. A poslední možnost nastavení se zabývá výměnou nástrojů, kde je také více možností jak se k tomuto problému při převodu postavit.

Tato část již byla řešena v bakalářském projektu, ale doznala významných změn. Byly opraveny chyby, upraven postup převodu a hlavně byla přidána tabulka nástrojů, která umožní uživateli vybrat z konkrétních nástrojů, které má stroj k dispozici.

1.2 Převod HPGL do ISO 6983

Podobně jako první část, tak tato část programu se postará o překlad vstupních dat zadaných v HP-GL do výstupního formátu ISO 6983. Vlastnosti a chování této části se

budeme snažit zachovat co nejvíce podobné, jako to je v případě převodu Excellon do ISO 6983 z důvodu uživatelské přívětivosti.

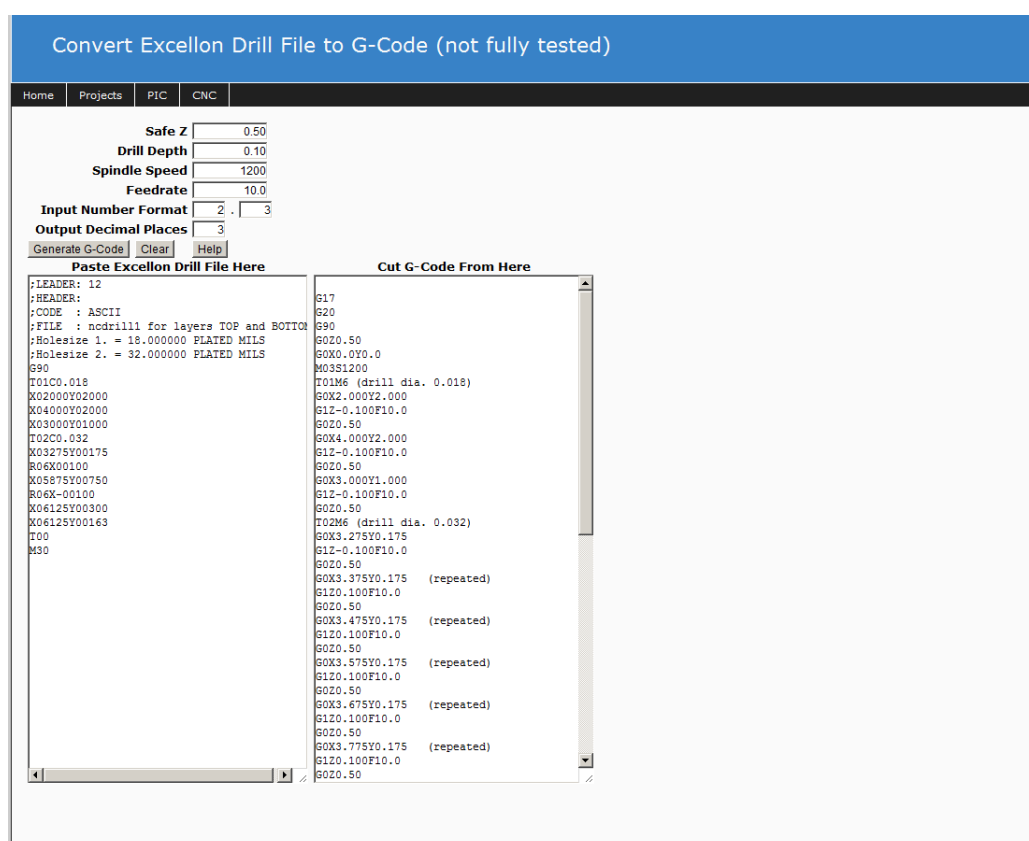
Opět tedy budeme načítat vstupní data, zpracujeme převod mezi formáty a umožníme uživateli jejich uložení a nastavení parametrů převodu.

Oproti převodu dat z formátu Excellon, by bylo dobré v této části umožnit uživateli i optickou kontrolu výstupních dat, tedy pokusit se vykreslit vstupní data. Takže uživatel bude mít možnost před uložením data zkontrolovat jak v textové podobě, tak i grafické.

2. | Obdobné aplikace

Před tím, než byla zahájena práce na aplikaci, bylo úkolem zjistit, jaké jsou aktuální možnosti. Úkolem tedy bylo snažit se na internetu najít informace popřípadě i vyzkoušet různé aplikace, které by schopni plnit stejnou funkcionalitu.

Krom jedné aplikace nebyla nalezena žádná aplikace, která by dokázala převést oba dva vstupní formáty. Nejedná se o nějakou chybu, ale přeci jen je výhodnější, pokud by člověk přišel do styku s jednou aplikací, než se musel naučit logiku dvou či více aplikací.

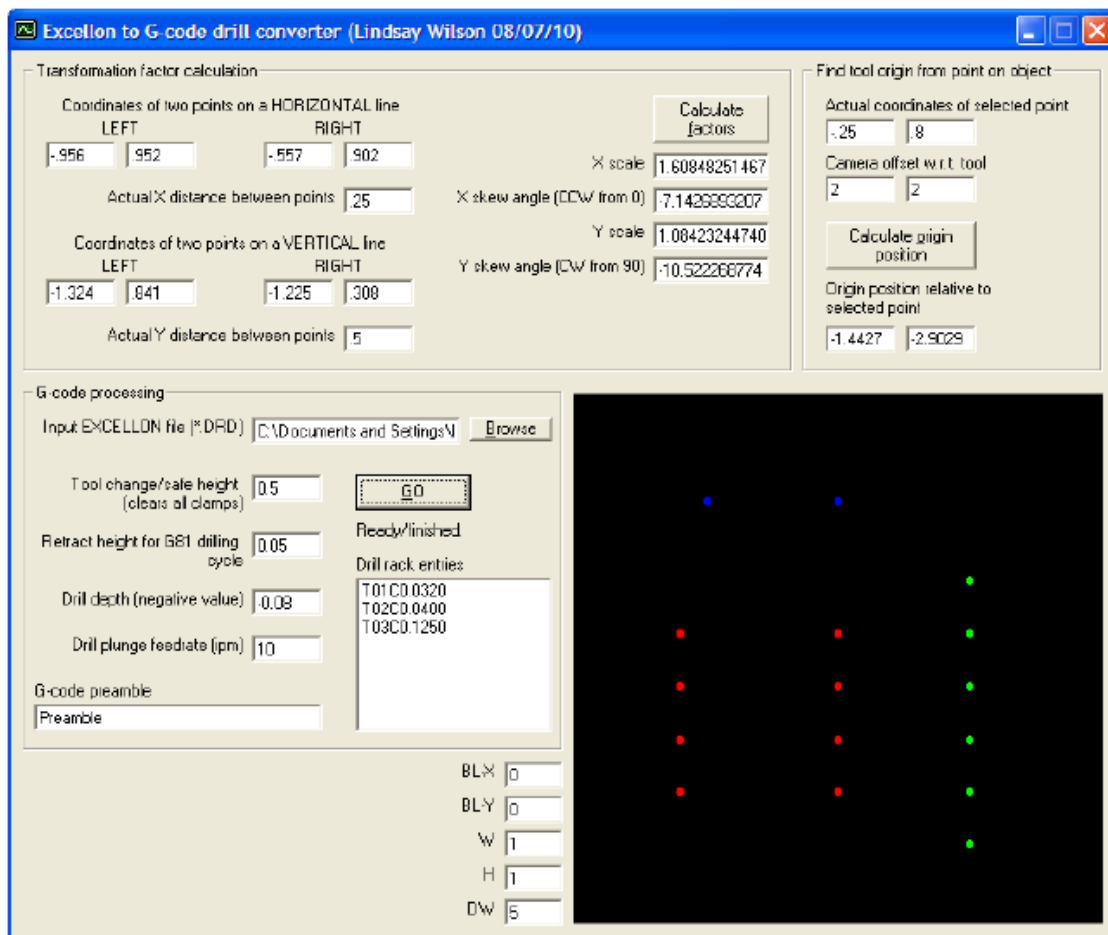


Obrázek 1 - Online nástroj ke konverzi Excellon dat do ISO 6983

Zajímavým nástrojem je online nástroj pro převod Excellon dat do ISO 6983, který se nachází na adrese http://eng.serve.com/cnc/excellon_gcode.html (na Obrázek 1 je aplikace zobrazena). Obsahuje vše, co by pro jednoduchý převod bylo potřeba. Na potíže bychom narazili, pokud bychom chtěli řešit automatickou výměnu nástrojů, nebo pokud bychom potřebovali převést více souborů a museli bychom pořád doplňovat svoji hlavičku či patičku. Jedná se o základní možnosti.

Dalším nástrojem pro převod Excellon dat do G-kódů je volně dostupná aplikace pro Windows, která je ke stažení na adrese <http://jdem.cz/un4n8>. Aplikace se jmenuje „Excellon to G-code drill converter“ a na Obrázek 2 je náhled okna aplikace. Nachází se u ní návod na

instalaci a dokonce i návod, jak převést data z programu Eagle, čímž lze aplikaci hodnotit jako přínosnou pro tuto potřebu, protože lze očekávat, že autor zkoušel převod právě na těchto datech. Umožňuje základní nastavení, opět chyběla možnost vkládat vlastní hlavičku a patičku a omezené možnosti práce s nástroji. Jen asi minimální nevýhodou by byl jazyk, aplikace je lokalizována pouze v angličtině.



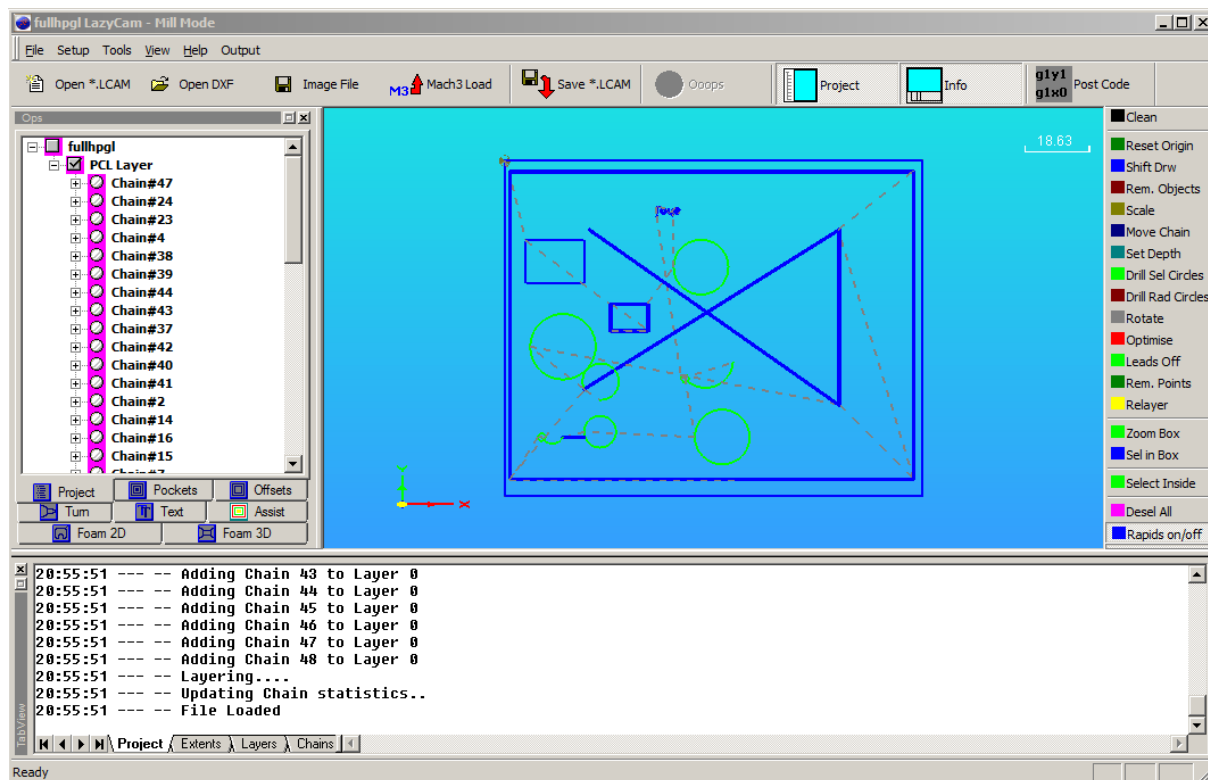
Obrázek 2 – Náhled aplikace „Excellon to G-code drill converter“

Poslední volně dostupná aplikace, která zároveň ale dokáže převést oba dva vstupní formáty je aplikace „LazyCam“ (k nalezení na adrese <http://www.machsupport.com>, náhled na Obrázek 3) od firmy Newfangled Solutions LLC. Stejná firma vyvíjí i program „Mach“, který slouží k ovládání CNC strojů.

Jedná se o CAM aplikaci, tedy aplikaci, která slouží k vytváření programů pro CNC stroje. Jako vstupní data tato aplikace umí většinu vektorových formátů, obrázky a další, která je možné využít na CNC strojích. Jedná se o jedinou aplikaci, která by byla schopna nahradit aplikaci vyvíjenou v této bakalářské práci.

Jak již bylo zmíněno, dokáže převést jak Excellon tak i HP-GL data. Dále nabízí velké možnosti nastavení a úprav. Ale zároveň tím, jak je potřeba složitě vše nastavit, tak se jedná

o horší uživatelskou ovladatelnost. Pro opakovaný převod s jasnými pravidly by se jednalo o zbytečně složité operace.



Obrázek 3 - Náhled programu „LazyCam“

Jedná se o všechny nalezené aplikace. Po pročtení několik fór by se dalo říci, že to jsou snad i všechny existující aplikace. Což ale nevylučuje, že nějaká taková podobná aplikace existuje nebo v budoucnu existovat bude, ale pro základní podvědomí o stavu aplikací můžeme usoudit, že stejná aplikace, jaká vznikne díky této práci, veřejně neexistuje.

3. | Formáty dat

Tato část práce se budeme věnovat formátům dat, které budeme v aplikaci využívat. Vybrány byly tedy formáty Excellon, jako zdroj vrtacích dat při výrobě DPS, HP-GL, jako zdroj konturových dat při výrobě DPS a ISO 6983, které budou sloužit k ovládání CNC stroje. Pro všechny formáty jsou na internetu k dostání referenční příručky, které jsou velice obsáhlé a detailní, ale formáty samotné většinou nabízejí velkou možnost variací zápisů programu. Díky tomu se setkáme s různými zápisy programu, ale pořád dle specifikace zapsané v mezích. Při vývoji programu jsme se snažili specifikaci poslouchat, ale i to nezaručuje, že se objeví nějaká data, která nebude možno převést.

3.1 ISO 6983

Programovací jazyk pro programování CNC strojů je popsán v normě ISO 6983. První vydání této normy je z roku 1982. Norma je již zastaralá a neobsahuje nejnovější potřeby programování CNC strojů a tak se připravuje nová norma ISO 14649. V běžném prostředí se tato norma vžila pod názvem „G-code“, tedy česky G-kódy.

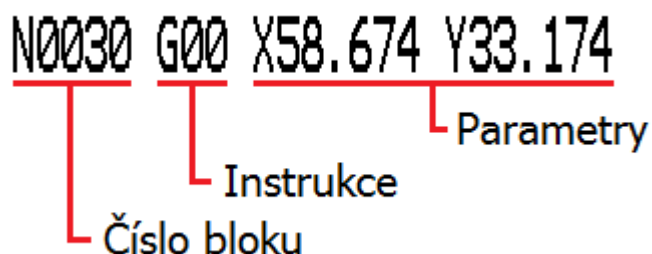
```
1 N0000 G21
2 N0005 (T01C0.0320)
3 N0010 (T02C0.0472)
4 N0015 G00 Z3
5 N0020 T01 M06 (T01)
6 N0025 G00 Z3
7 N0030 G00 X58.674 Y33.274
8 N0035 G81 R2 Z-2 F100
9 N0040 G00 Z3
10 N0045 G00 X61.214 Y33.274
11 N0050 G81 R2 Z-2 F100
12 N0055 G00 Z3
13 N0060 G00 X63.754 Y33.274
14 N0065 G81 R2 Z-2 F100
15 N0070 G00 Z3
16 N0075 G00 X66.294 Y33.274
17 N0080 G81 R2 Z-2 F100
```

Obrázek 4 - Ukázka programu v ISO 6983

Jedná se o programovací jazyk pro NC a CNC stroje. Jazyk, který říká stroji, co a jak má udělat. Vzhledem k univerzálnosti je využívám jak na frézkách, tak i na soustruzích a dalších CNC strojích. Proto jsou některé funkce určeny jen pro určitý druh stroje. Na obrázku Obrázek 4 je zobrazena část programu.

3.1.1 Formát dat

Norma říká přesně, jak mají zapsané data vypadat. Každá instrukce je zapsána na nový řádek, který se nazývá blok. Blok může být číslovaný (řádek začíná písmenem N a číslem řádku), následovaným instrukcí a parametry. Ukázku bloku máme na *Obrázek 5*.



Obrázek 5 - Ukázka bloku instrukce

Jazyk se skládá z G kódů a M kódů. G kódy (instrukce začínající písmenem G) říká stroji, co má dělat. M kódy jsou řídicí kódy programu a nastavují parametry stroje. M kódy jsou také nejvíce závislé na typu stroje, jestli se jedná o frézku nebo soustruh. Mezitím co u G kódů záleží u strojů na dostupných osách.

Vraťme se k *Obrázek 4*, kde jsme měli náhled dat. Popíšeme si řádek 4 a 5. Na řádku 4 máme na první pozici písmeno **N**, které značí, že se jedná o začátek číslování bloku. Číslování řádku se využívá hlavně v případech, že bychom využívali instrukcí, které umožňují přejít na konkrétní řádek. Za označením řádku se nachází instrukce **G00**, který stroji říká, že má rychloposuvem přejít na souřadnice. Souřadnice, zadána jako parametr, je v tomto případě zapsaná jako **Z3**, kde Z značí, že se budeme pohybovat v ose Z a pojedeme na pozici 3. A na pátém řádku najdeme za označením řádku instrukci **M06**, která říká, že má proběhnout výměna nástroje. Jako parametr ji předáváme hodnotu **T1**, což značí, že stroj má vyměnit aktuální nástroj za **T1**.

3.1.2 Použité instrukce v programu

V programu je využita jen malá podmnožina instrukcí. V případě převodu dat z formátu Excellon jsou použity úplně základní, v případě HPGL dat je použito několik příkazů navíc.

3.1.2.1 Inicializační instrukce

Na začátku kódu je vhodné provést inicializaci instrukcemi G40, G49 a G80, které vypnou případné volby (offsety nástrojů). Jedná se o ochranu v případě, že by předtím na stroji byl spouštěn program, který by tyto volby využíval.

Dalším inicializačním kódem je kód G20 respektive G21. V programu je vždy přítomna jedna instrukce a to podle toho, jestli jsou hodnoty v programu zapsány v palcích respektive milimetrech.

Poslední inicializací bude, abychom stroji řekli, jestli jsou příkazy zadávány absolutně, nebo inkrementálně. K tomu nám slouží příkazy G90 pro absolutní a G91 pro inkrementální programování.

3.1.2.2 Instrukce G0 a G1

Instrukce G0 slouží jako rychloposuv stroje. Jedinými parametry jsou souřadnice, na které má stroj přejet. K přejezdu je využita nejvyšší možná rychlost stroje. Tato instrukce je využita v případech, že potřebujeme přejet a neobrábíme materiál.

Instrukce G1 je podobná instrukci G0, ale přesný název je lineární interpolace. Tomuto příkazu kromě souřadnic můžeme také předat rychlost posuvu.

Obě dvě instrukce slouží k přesunu z aktuální souřadnice na novou.

3.1.2.3 Instrukce G02 a G03

Jedná se o kruhové interpolace. Rozdíl mezi instrukcemi G02 a G03 je ve směru interpolace. Instrukce G02 interpoluje po směru hodinových ručiček, zatímco G03 proti hodinovým ručičkám.

Provedení interpolace je možné více způsoby, tím se liší i parametry, konkrétněji se budeme této instrukci věnovat v sekci převodu HP-GL dat, kde ji využijeme.

3.1.2.4 Instrukce G81 a G82

Instrukce G81 a G82 slouží k vrtání. Tyto instrukce využijeme u Excellon dat. Rozdíl mezi instrukcemi G81 a G82 je ve stylu vrtání. G81 vrtá bez čekání, zatímco G82 vrtá mezikroky s čekáním. V případě, že nevyužíváme rozdělení vrtání na mezikroky, tak se instrukce chovají stejně.

3.1.2.5 Instrukce M1 a M30

Jedná se o programové zastavení. Instrukci M01 využijeme, pokud uživatel nastaví, aby výměna nástrojů probíhala pomocí této instrukce. Pokud na ní stroj narazí tak přestane vykonávat další příkazy. V programu je pak možno pokračovat.

V případě M30 dáváme vědět, že program dosáhl svého konce. Vkládáme ji tedy na konec programu.

3.1.2.6 Instrukce M6

Instrukce pro automatickou výměnu nástroje. V případě, že uživatel v nastavení zvolí automatickou výměnu nástroje, tak budeme do kódu vkládat tuto instrukci v místech, kde ve zdrojovém kódu dochází ke změně nástrojů. Instrukci předáváme parametrem číslo nástroje. Pokud chceme tedy zvolit nástroj číslo 5, tak napíšeme M6 T5.

3.2 Excellon

Formát dat Excellon je využíván v elektronice pro řízení CNC strojů, které vyrábějí desky plošných spojů. Název Excellon vznikl podle firmy Excellon Automation Company, která tento formát vznikl v 80 letech a firma jej využívá na svých strojích pro vrtání desek s plošnými spoji a jejichž formát se rozšířil i mimo firmu.

3.2.1 Formát dat

Data jsou uložena v textovém formátu. Program je možno vytvořit i bez pomoci speciálního software v prostém textovém editoru. Velkým problémem formátu je nepřesné dodržování standardu. Proto se setkáme s různými implementacemi. Většinou se jedná však o různé úpravy v hlavičce dat pro konkrétní stroj nebo rozdělením souboru do dvou souborů, kde samotný soubor s vrtacími daty dodržuje specifikaci, ale druhý soubor obsahuje hlavičku, která není specifikována a je vytvořena pro konkrétní aplikaci či stroj. Díky tomu je potřeba ohlídat některé parametry. Pokud je hlavička nestandardní, tak bývá uživatelsky čitelná.

```
1 M48
2 INCH,TZ
3 VER,1
4 FMAT,2
5 T01C0.03937F100B1000S6H2000
6 T02C0.055118F100B1000S6H2000
7 T03C0.125984F100B1000S6H2000
8 T04C0.031496F100B1000S6H2000
9 T05C0.047244F100B1000S6H2000
10 DETECT,ON
11 ATC,ON
12 %
13 T04
14 X23562Y54699
15 X24062Y55199
16 X23562Y55699
17 Y56699
18 X25062
19 Y57949
```

Obrázek 6 - Ukázka dat ve formátu Excellon

Tomu je také tak zčásti třeba u programu Eagle, který také generuje soubor Excellon s vrtacími daty a druhý soubor, který obsahuje informace o nástrojích, jednotky a další. Ale od jiných programů naštěstí ponechává i základní hlavičku, takže se bez druhého souboru obejdeme.

Na rozdíl od HP-GL formátu neobsahuje většina Excellon dat žádné zvláštní příkazy. Důležitou částí Excellon dat je hlavně hlavička, která nám říká vše podstatné, co pro převod potřebujeme. Zbytek programu jsou samotná vrtací data, kde na každé řádce máme souřadnice jednotlivých vrtaných otvorů, popřípadě občas narazíme na příkaz pro změnu nástroje. Každá instrukce je zapsána na nový řádek.

Na *Obrázek 6* je zobrazena část kódu. Na něm se pokusíme vysvětlit, co většinou soubor s Excellon daty obsahuje. Na prvním řádku máme příkaz **M48**, který značí začátek hlavičky. Na dalším řádku nalezneme důležitou informaci, v tomto případě **INCH,TZ**, která nám říká, že všechny měrné jednotky jsou v palcích s koncovými nulami. Řádek 3 nás informuje o verzi rozložení souřadnicového systému, v tomto případě 1. Řádek 4 **FMAT,2**, informuje o formátu zápisu dat, zde tedy formát 2. Mezi řádky 5 a 9 nalézáme informace o použitých nástrojích. Za písmenem **T** je číslo nástroje, za písmenem **C** je průměr nástroje, písmeno **F** značí rychlost posuvu, **B** říká stroji, jaké má použít vrtací otáčky a poslední údaj, za písmenem **H**, říká stroji, kolikrát může být vrták použit. Na řádku 10 je napsána informace, že stroj má kontrolovat, jestli není vrták zlomen. A poslední údaj v hlavičce na řádku 11 je znamení pro stroj, že má automaticky vyměňovat nástroje. Znak **%** na 12 řádku je návěští, kde se má program v případě přetočení zastavit.

Od řádku 13 se nachází samotné jádro dat. První informací je, že budeme používat nástroj **T04**. Na dalším řádku již máme souřadnici. Za písmenem **X** respektive **Y** máme hodnotu souřadnice **X** respektive **Y**. Číslo je udané v palcích, to jsme se dozvěděli v hlavičce. Dalším údajem je, že čísla jsou uvedena s koncovými nulami, tento formát zápisu čísel značí, že třeba hodnota X7500 je 0,75 palce a hodnota X75 je 0,0075 palce. My máme na řádku hodnotu X23562, převedeno na reálnou hodnotu to je 2,3562 palce. V programu vše přepočítáváme na milimetry, pro lepší orientaci ve výsledném programu, takže musíme hodnotu, kterou získáme vynásobit ještě 25,4.

Zbytek programu je již veden v tomto formátu. Důležité je tedy najít v hlavičce důležité údaje a podle nich upravit chování programu aby vždy data správně převedl.

3.2.2 Použité instrukce v programu

Pro převod dat je pro nás důležité jen zjistit, jestli jsou hodnoty zapsané v milimetrech nebo palcích, k čemu nám slouží instrukce **INCH** a **METRIC** v hlavičce programu. Další

důležitou informací je, jak jsou čísla zapsána, jestli s nulami za nebo před číslem. K tomu slouží instrukce TZ respektive LZ. Poté již máme instrukci pro změnu nástrojů, začínající písmenem T a následovanou číslem nástroje. A poslední instrukcí je instrukce vrtání, kterou poznáme podle hodnoty X, Y na samostatném řádku. Na řádku se může nacházet jedna nebo obě souřadnice. V případě, že se nachází jen jedna, tak s druhou osou nepohybujeme.

3.3 HP-GL

Formát dat HP-GL (nebo také HP/GL či jen HPGL) vyvinula firma Hewlett-Packard pro ovládání souřadnicových plotterů. HP-GL je akronym pro Hewlett-Packard Graphics Language. Formát HP-GL nevyužívá jen firma HP, ale i jiné firmy, které vyrábějí plottery jako IBM či Brother.

Rozšířením formátu HPGL vznikl formát HP-GL/2 (často také najdeme zápis HP/GL-2), který doplňuje jazyk HP-GL o šířku čáry a další funkce potřebné pro moderní plottery, které již nepracují s pery, ale obsahují tiskové hlavy.

3.3.1 Formát dat

Zápis dat probíhá v textovém formátu, takže je data možno zapisovat bez nějakého speciální software, stačí jen referenční příručka a textový editor. Formát dat je daný a to konkrétně na dva znaky, které určují, co se má vykonat, následované možnými parametry. Když formát vznikal, komunikovali plottery po sériové lince RS232. Takže formát dat je prostý textový formát s ASCII znaky.

```
1 IN; IP 0,0,100,100; SC 0,100,0,100;  
2 SP 2;  
3 PA 16,10;  
4 PD;  
5 PA 13,11;  
6 PA 11,13;  
7 PU;  
8 PA 2254,1491;  
9 PD;  
10 AA 2273,1636,121.26;  
11 PU; SP 0; PA 0,0;
```

Obrázek 7 - Ukázka dat ve formátu HP-GL

Na *Obrázek 7* máme opět náhled dat v tomto formátu. Při čtení dat je důležité, že každý kreslicí nebo řídící příkaz je zapsána právě dvěma znaky. Za příkazem mohou nebo nemusí být uvedeny parametry. Od sebe jsou odděleny buď mezerou „ “, čárkou „,“ nebo znaménkem plus „+“ nebo mínus „-“. Příkazy mohou být mezi sebou odděleny středníkem „;“, koncem řádku nebo nemusí mít oddělovač žádný, protože lze jednoduše rozeznat, jestli

se jedná o příkaz nebo parametr. Jedná se právě o ušetření znaků v případě komunikace po sériové lince.

3.3.2 Podporované příkazy

V této části si rozebereme příkazy implementované v programu. Nejedná se o celou množinu HP-GL dat. Vynechány jsou příkazy pro práci s textem. Důvodem je, že většina dat bude mít text převeden do křivek, nebo je možno si to zvolit při převodu. Jednalo by se o velký problém, protože by se museli vytvořit fonty a obsáhnout všechna práce s tím spojená. A vzhledem k tomu, že program Eagle text převádí do křivek, by to bylo i zbytečné. Podporované jsou tedy základní funkce, které pro většinu případů stačí.

3.3.2.1 PD – Pero dolu, PU – Pero nahoru

Jedny ze základních HP-GL příkazů. Na plotteru způsobí, že se pero položí na papír, respektive zdvihne z papíru. V HP-GL se instrukce o polohu pera nestarají. Díky tomu se všechny kreslicí instrukce provedou, ať je pero dole nebo ne. Pokud tedy necháme pero nahoře a zavoláme kreslicí příkaz, tak by se příkaz provedl „ve vzduchu“, takže by se na papír nic nenakreslilo.

To samé znamená pro nás. Ať již při kreslení obrázku do náhledu, kde pokud není pero dole, tak se nic nekreslí, nebo po převodu do G-kódů, kde budeme instrukci PD převádět na snížení osy Z do materiálu, nebo naopak instrukci PU přeložíme do G-kódů jako zdvihnutí osy Z. Tím zajistíme stejné chování těchto instrukcí jako v HP-GL.

Z uživatelského hlediska je vhodné, aby uživatel mohl ovlivnit, o kolik se má osa Z při kreslení spustit respektive zdvihnout, proto má uživatel možnost tyto hodnoty změnit v nastavení.

3.3.2.2 PA – Úsečka absolutně, PR – Úsečka relativně

Základní kreslicí příkazy. Účelem příkazu na plotteru je nakreslit úsečku v případě, že je pero dole, nebo v případě že je pero nahoře, tak přejet na pozici.

Jako první bod úsečky se považuje ten bod, kde se hlava plotteru nachází. Druhý bod je parametrem příkazu. Příkazy se liší, v zadání koncového bodu. V případě PA příkazu se koncový bod zadává v absolutních hodnotách. Mezitím co u příkazu PR zadáváme souřadnice koncového bodu v relativních hodnotách.

Převod do G-kódů je u tohoto příkazu jednoduchý. Využijeme G-kódy G00 a G01. Rozdíl v tom, kdy použijeme G00 (rychlé polohování) a G01 (lineární interpolace) je v tom, jestli se pero nachází nahoře nebo dole. V případě, že je pero zdviženo, můžeme použít G-kód

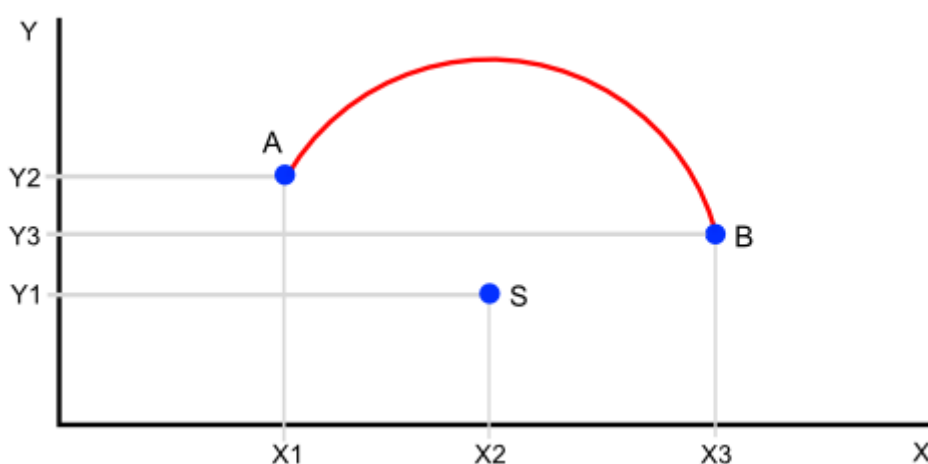
G00, protože se nenacházíme v materiálu a jen přejíždíme. Mezitím co když je pero dole, tak použijeme kód G01, abychom respektovali nastavené rychlosti obrábění.

3.3.2.3 AA – Kruh absolutně, AR – Kruh relativně

Další celkem často využívanou je kreslení kruhové výseče. K tomu v HP-GL slouží příkazy AA a AR. Jako u každého příkazu, pokud je pero plotteru dole, nakreslí se kruhová výseč, pokud je nahoře, tak záleží na implementaci v plotteru, buď se kruh nakreslí „ve vzduchu“, nebo rovnou přejede na koncový bod.

Parametry těchto příkazů jsou střed kružnice (souřadnice X a Y), středový úhel výseče ve stupních a jako nepovinný parametr je úhel tětiny. Středový úhel je možno zadat v kladných číslech, v tom případě se výseč bude vykreslovat proti hodinovým ručičkám, nebo v záporných číslech, kdy se bude výseč vykreslovat po směru hodinových ručiček. Hodnoty větší než 360 jsou děleny modulem 360. Opět máme instrukce zdvojené, podle toho, jestli jsou středové souřadnice zadány absolutně nebo relativně.

Jedná se o nejtěžší příkaz co se převodu do G-kódu v HP-GL nachází. Byť G-kódy obsahují funkce, pro práci s kruhovou interpolací, konkrétně kódy G02 a G03, tak jejich parametry si vzájemně neodpovídají. Proto je potřeba provést pár výpočtů. Příkazy G02 a G03 se liší směrem interpolace kruhu. Příkaz G02 interpoluje kružnici po směru hodinových ručiček, mezitím co příkaz G03 proti hodinovým ručičkám. Parametry mají stejné.



Obrázek 8 - Kruhová výseč

Na *Obrázek 8* máme náčrtek kruhové výseče. V HP-GL máme kruhovou výseč danou bodem **A** a středem **S**. Bod **A** o souřadnicích **X1** a **Y1** je aktuální pozicí pera. A souřadnice bodu **S** předáváme jako parametr příkazu AA respektive AR. Středový úhel mezi bodem **A** a **B** má nulovou hodnotu v bodě **A**.

Mezitím, chceme-li nakreslit kruhovou výseč v G-kódech, tak máme více možností. Příkazy G02 a G03 umožňují nakreslit kruhovou výseč pomocí bodů **A**, **B** a poloměru, nebo pomocí bodů **A**, **B** a středu **S**.

V programu využívám první možnosti, tedy za pomoci bodů **A**, **B** a poloměru. Z HP-GL příkazu známe souřadnice bodu **A**, středu **S** a středového úhlu. Musíme tedy dopočítat souřadnice bodu **B** a poloměr. Je využito jednoduché matematiky a goniometrických funkcí.

V případě, že se pero nenachází ve spodní poloze, tak využijeme příkaz G00 a jako souřadnice použijeme koncový bod výseče. A to z důvodu, že kdyby následoval další příkaz, který by respektoval poslední polohu, tak by neudělal správně to, co by měl.

3.3.2.4 CI – kruh

Jak už název napovídá, jedná se o příkaz, který má za úkol nakreslit kruh. Rozdíl oproti předchozím příkazům je v tom, že se na plotteru pero vždy samo spustí dolu.

Parametry má tento příkaz dva, prvním povinným parametrem je poloměr a druhým nepovinným je úhel tětiny. Jako střed kruhu se bere aktuální pozice pera před provedením tohoto příkazu.

Pro převod do G-kódů využijeme opět příkaz G02 nebo G03, jaký příkaz si vybereme je irelevantní, proto volíme příkaz G02. Využijeme druhou možnost zadávání kruhové interpolace, než kterou jsme využili při kreslení výseče. V této možnosti zadáme poloměr, ale nejdříve se musíme přesunout na kraj kružnice. Proto nejdříve zavoláme příkaz G00 (je-li nástroj dole, tak ho nejdříve zvedneme) a přesuneme se o hodnotu poloměru doleva, poté přemístíme nástroj do pracovní polohy a pomocí příkazu G02 a zadaného poloměru vytvoříme kruh. Z důvodu návaznosti na další příkazy opět nástroj zdvihneme a vrátíme se do středu kružnice.

4. | Implementace

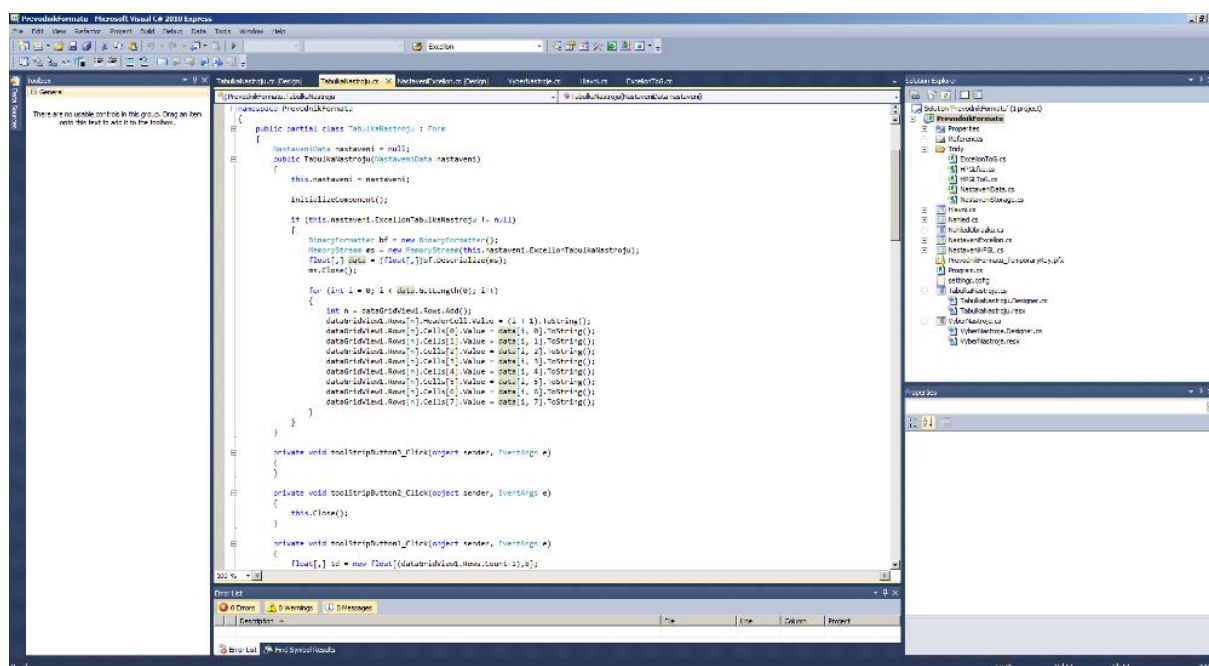
4.1 Použité technologie

K této práci nebyly za potřebí žádné složité implementační technologie. Důležitým faktorem bylo nastudování vstupních a výstupních formátů dat a rozvržení možnosti převoditelnosti.

Z uživatelského pohledu bylo důležité, aby byla použitelnost co nejvíce jednoduchá a intuitivní. Proto jsme se rozhodli pro vytvoření Windows GUI aplikace. Bylo potřeba vybrat nějaký vhodný nástroj pro vývoj této aplikace. Z hlediska znalosti programovacích jazyků a zkušenostech, získaných při výuce, bylo vybráno Microsoft Visual Studio. A jako programovací jazyk byl vybrán Visual C#.

4.1.1 Microsoft Visual Studio

Jedná se o vývojové prostředí (IDE) dodávané firmou Microsoft. Ukázka prostředí je na *Obrázek 9*. V základní nekomerční verzi je k dispozici zdarma na stránkách firmy. Vzhledem k tomu, že aplikace bude využívána v prostředí Windows a také vzhledem k minimálním znalostem vývoje grafických aplikací pro Windows se jedná o nejlepší možný výběr. Prostředí je velice dobře navrženo pro vývojáře a vývoj aplikace probíhá velmi intuitivně. Není potřeba nejdříve nastudovat stohy knih. Stačí si jen několikrát zkusit vytvořit nějakou jednoduchou aplikaci a umět alespoň základní angličtinu.



Obrázek 9 - Ukázka vývojového prostředí Microsoft Visual Studio

Prostředí v sobě obsahuje vše pro vývoj aplikací – editor kódu, překladač, debugger, designer GUI, podporu pro práci s verzemi a další. Díky tomu není potřeba se přepínat mezi různými aplikacemi. Tato vlastnost nám při vývoji ušetří spoustu času.

Microsoft Visual Studio podporuje různé jazyky jako je Microsoft Visual Basic, Visual J#, Visual C# a Visual C++. Pro tuto konkrétní aplikaci byl vybrán jazyk Visual C#. Jedná se o objektově orientovaný jazyk založený na jazyce C++ a Java.

4.2 Grafické rozhraní programu

GUI (grafické uživatelské rozhraní) bude to jediné, co uživatel uvidí, proto je potřeba dát si pozor, aby bylo přehledné, intuitivní a uživatele neodrazovalo od práce s programem.

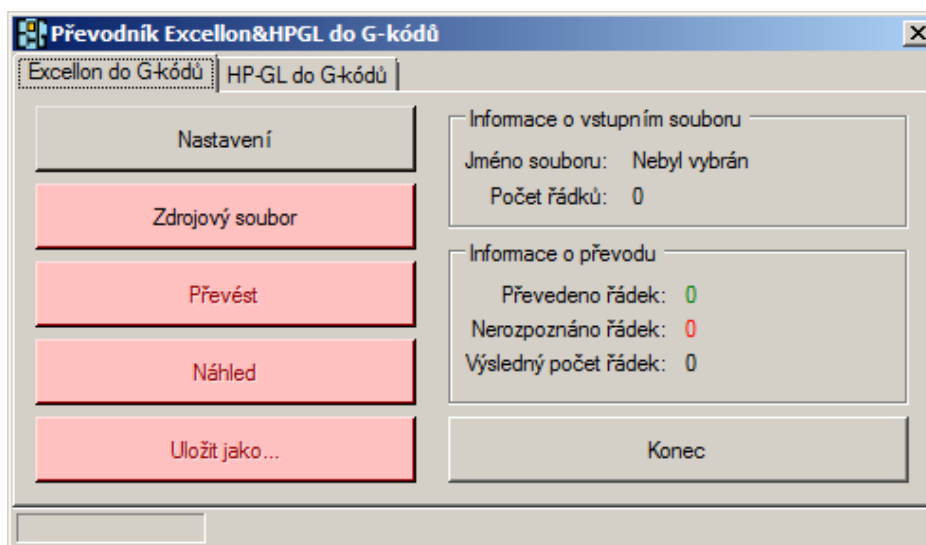
Protože budeme převádět z Excellon a HP-GL data, rozdělíme program na dvě části. Každá bude tedy zajišťovat právě jeden převod. I když by bylo možno vše sjednotit a snažit se rozpoznat vstupní data, jedná se přesto o nejpřehlednější postup. Využijeme k tomu záložky. Takže na jedné záložce budeme převádět Excellon data a na druhé HP-GL data.

Na každé záložce najde uživatel prvky pro ovládání převodu. Načtení vstupního souboru, nastavení převodu, samotný převod a kontrolu dat a poté samozřejmě uložení výsledného souboru.

Zajímavým prvkem je obarvování a zpřístupnění respektive znepřístupnění tlačítek tak, aby uživatel nemusel zdlouhavě přemýšlet, jak se program ovládá.

4.2.1 Hlavní okno

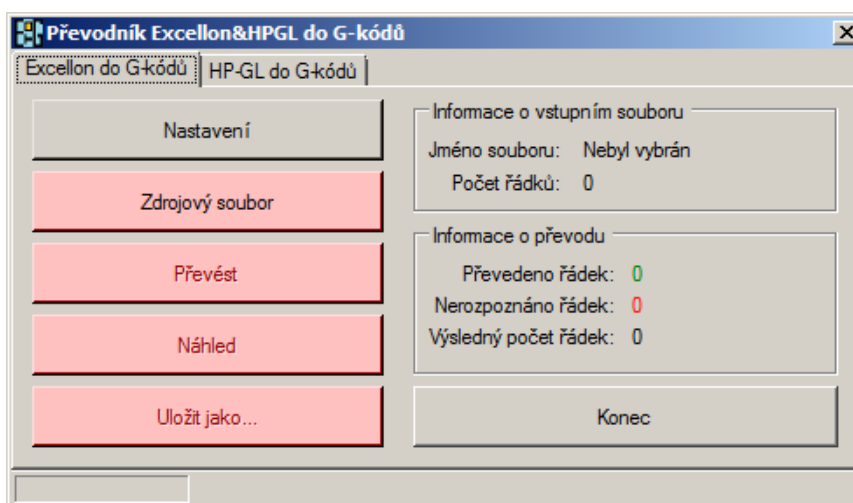
Jedná se o první kontakt uživatele s aplikací po jejím spuštění. Náhled je na *Obrázek 10*.



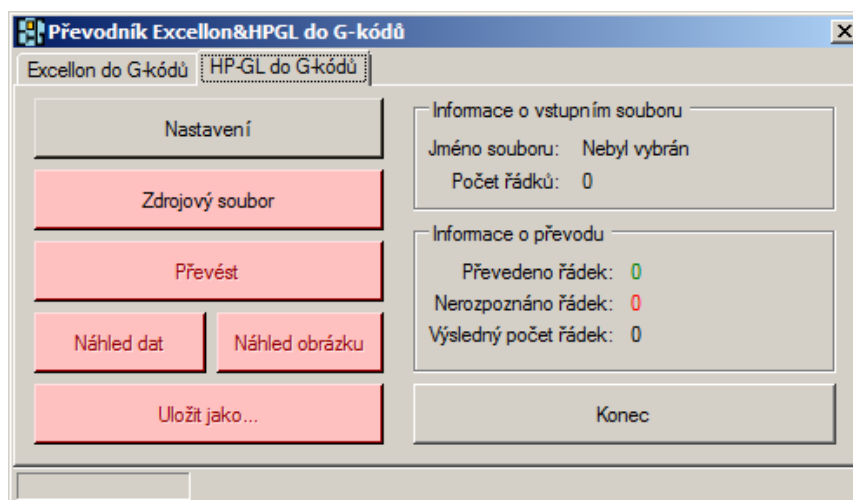
Obrázek 10 - Program po spuštění

4.2.2 Záložky „Excellon do G-kódů“ a „HPGL do G-kódů“

Po spuštění na hlavním okně máme vlevo na nahoře na výběr ze dvou záložek. První záložka se jmenuje „Excellon do G-kódů“ (zobrazena na *Obrázek 11*) a druhá záložka „HP-GL do G-kódů“ (zobrazena na *Obrázek 12*).



Obrázek 11 - Záložka „Excellon do G-kódů“



Obrázek 12 - Záložka "HP-GL do G-kódů"

V levé části ona nalezneme tlačítka pro práci s programem. Prvním je tlačítko „Nastavení“, po kliknutí na toto tlačítko se uživateli zobrazí nové okno s možnostmi nastavení popsané dále. Po kliknutí na tlačítko „Zdrojový soubor“ se zobrazí formulář pro výběr souboru, se kterým bude program pracovat. Další tlačítko „Převést“ slouží k samotnému převodu dat, po kliknutí na toto tlačítko jsou vstupní data v jednom formátu převedena na data ve výstupním formátu. Tlačítko „Náhled dat“ slouží ke kontrole dat. Otevřena nám nové okno s náhledem dat, vysvětlíme si jej také dále. Tlačítko „Uložit jako...“ slouží k uložení souboru s výstupními daty. Po kliknutí se zobrazí standartní formulář pro uložení souboru.

V pravé části najdeme informační oblast, kde se zobrazuje vše, co spojené se vstupním souborem – jméno otevřeného souboru, počet řádků v souboru. A druhá sekce se věnuje informaci o stavu převodu.

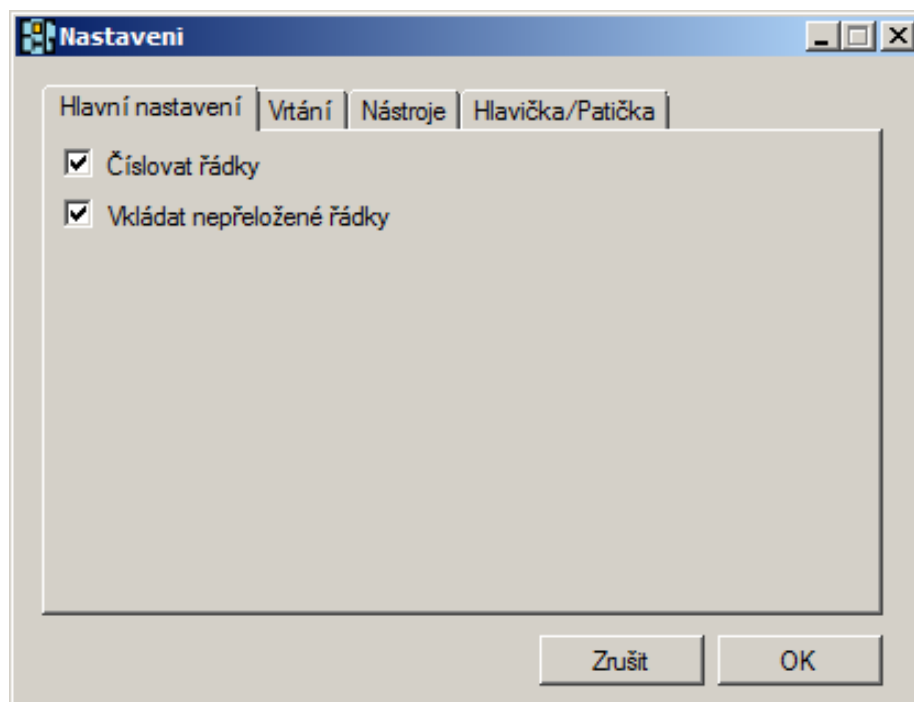
Jediný rozdíl mezi záložkami je, že u převodu HP-GL dat máme vedle tlačítka „Náhled dat“ také tlačítko „Náhled obrázku“, který nám zobrazí náhled obrázku vygenerovaného z HP-GL dat.

4.2.3 Nastavení

Okno s nastavením se liší podle toho, s jakými daty pracujeme, proto si popíšeme tyto okna zvlášť.

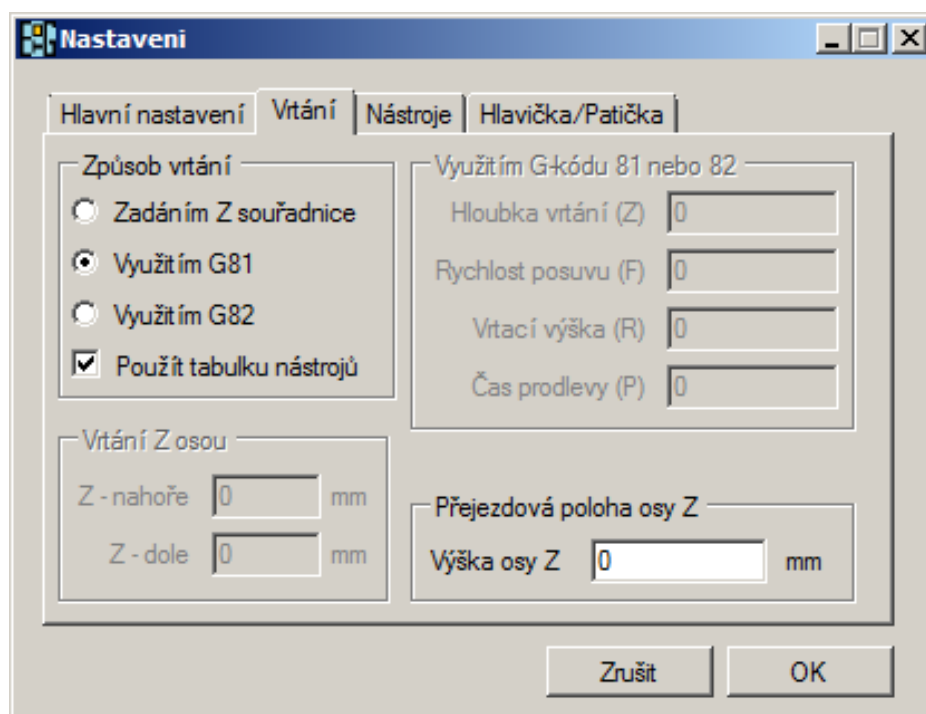
4.2.3.1 Nastavení převodu u Excellon dat

U převodu dat z formátu Excellon máme v okně nastavení čtyři záložky.



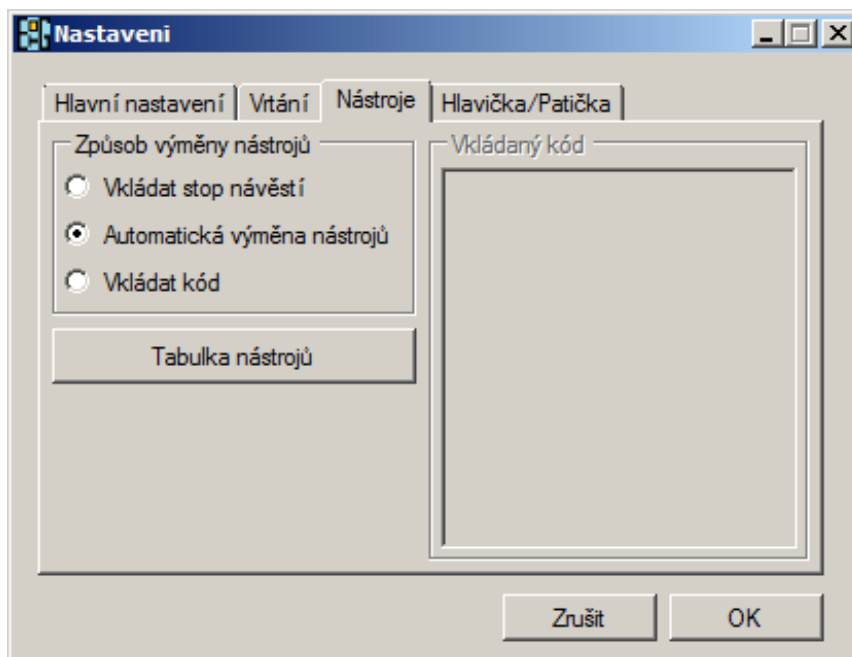
Obrázek 13 - Hlavní nastavení Excellon

První záložka, kterou můžeme vidět na Obrázek 13, se jmenuje „Hlavní nastavení“ obsahuje dvě výběrová pole. První nastavuje, jestli budeme číslovat řádky ve výstupních G-kódech a druhá volba nám určí, jestli budeme vkládat nepřeložené příkazy jako komentář do výstupních dat, což se hodí v případě, že bychom chtěli příkaz převést sami nebo se třeba již jedná o komentář.



Obrázek 14 - Nastavení vrtání

V záložce „Vrtání“, kterou máme vyobrazenou na *Obrázek 14*, můžeme měnit parametry, které se vrtání týkají. Prvním nastavením je určení, jaký způsob se pro vrtání využije. Jestli budeme vrtat posunem Z souřadnice stroje, nebo využijeme G-kódů k tomu určených – G81 nebo G82. V případě, že vybereme vrtání posunem osou Z, tak můžeme nastavit hodnoty pro horní a dolní polohu souřadnice. V případě vybrání jednoho z G-kódů můžeme i zde nastavit parametry které se těmto příkazům při převodu předají, popřípadě můžeme vybrat, že se mají tyto údaje načíst z tabulky nástrojů. Posledním údajem nastavovaným na této záložce je údaj o přejezdové výšce osy Z.



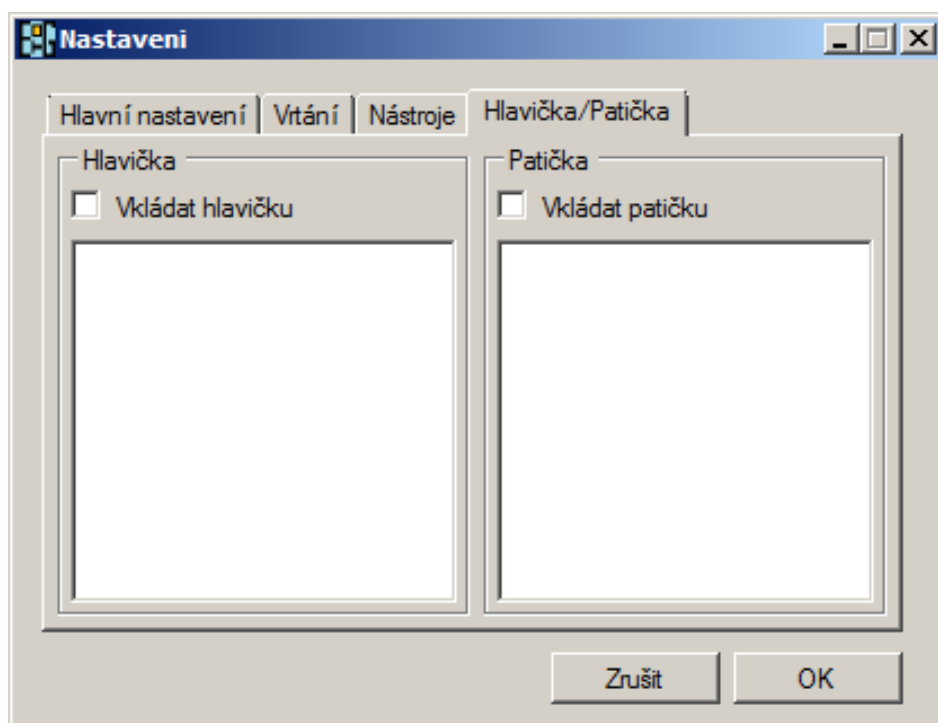
Obrázek 15 - Nastavení nástrojů

Další záložka s názvem „Nástroje“, zobrazenou na *Obrázek 15*, se věnuje způsobu výměny nástrojů. Na výběr jsou tři možnosti. Vkládáním M kódu, který pozastaví stroj a je umožněna výměna nástroje, nebo pomocí automatické výměny nástrojů implementované v G-kódech nebo bude vkládán vlastní kód, který je možno vložit v pravé části. Nachází se zde i tlačítko pro zobrazení tabulky nástrojů.

Tabulka nástrojů						
Uložit Zavřít						
	Průměr vrtáku (mm)	Otáčky vřetene (H)	Rychlost posuvu (F)	Přírutek vrtáku	Počet opakování (L)	Čas prodlevy (P)
▶ 1	0,22	5000	700	0,2	0	0
2	0,2	5000	650	-0,1	0	0
3	0,1	5000	500	0,1	0	0
*						

Obrázek 16 – Tabulka nástrojů

Po kliknutí na tlačítko „Tabulka nástrojů“ se nám zobrazí tabulka, zobrazená na *Obrázek 16*, ve které můžeme nastavit parametry nástrojů, které máme k dispozici a pokud je v záložce „Vrtání“ vybráno vrtání pomocí G-kódu a je vybrána volba „Využít tabulku nástrojů“, tak jsou při převodu data z této tabulky využita pro načtení parametrů konkrétního nástroje.



Obrázek 17 - Nastavení hlavičky a patičky

Na poslední záložce nazvanou „Hlavička/Patička“ (Obrázek 17) můžeme určit, jestli se bude vkládat hlavička a patička do převedených dat. Samotné data, která mají být vložena, se zadají do textových polí.

4.2.3.2 Nastavení převodu u HPGL dat

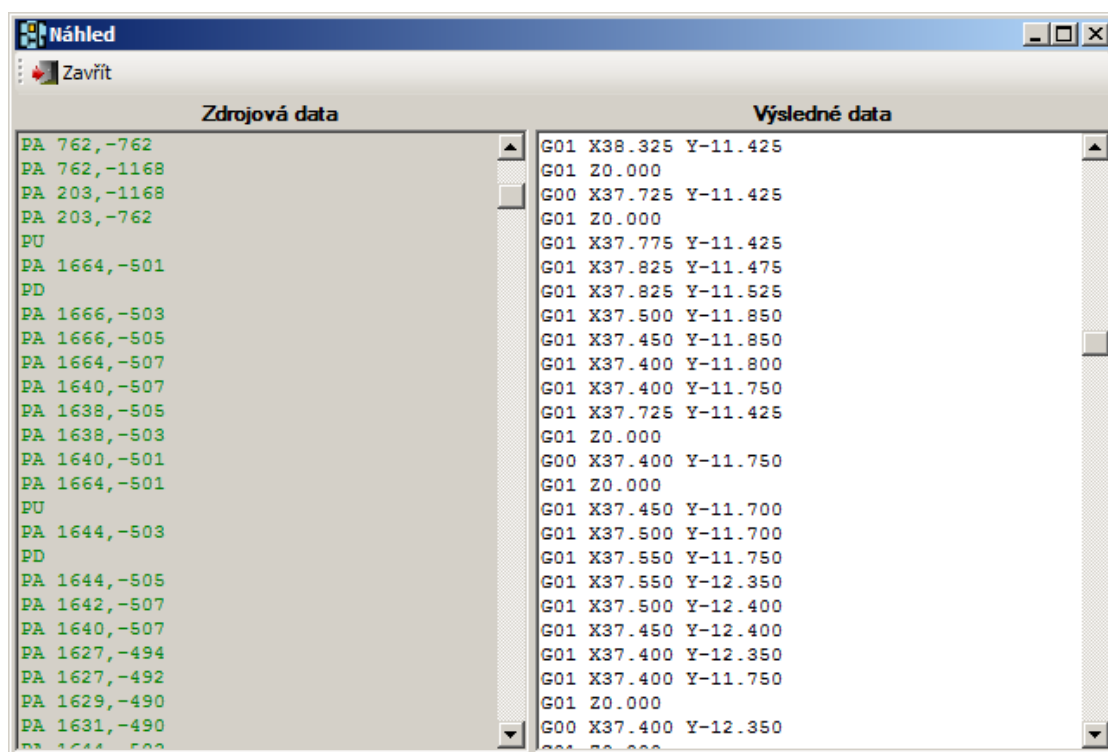
Oproti nastavení u Excellon dat nám zde ubyla záložka „Vrtání“, která je v tomto případě irelevantní. Jediný údaj, který je potřeba nastavovat je poloha Z osy, po které se bude stroj pohybovat, když nebude v materiálu, a toto nastavení se přesunulo do záložky „Nástroje“. A v záložce nastavení ubylo tlačítko pro tabulku nástrojů, protože v případě HP-GL dat bychom ji nevyužili.

4.2.4 Náhled

Toto okno je rozděleno na polovinu. Náhled na toto okno máme na Obrázek 18. V levé polovině se zobrazují vstupní data, v pravé polovině se zobrazují výstupní data, která lze i editovat.

U zdrojových dat jsou řádky rozděleny barevně, a to z důvodu, že řádky obarvené zeleně se nám povedlo převést, u řádků zabarvených červeně nastala chyba (nebyla rozpoznán příkaz, špatný zápis).

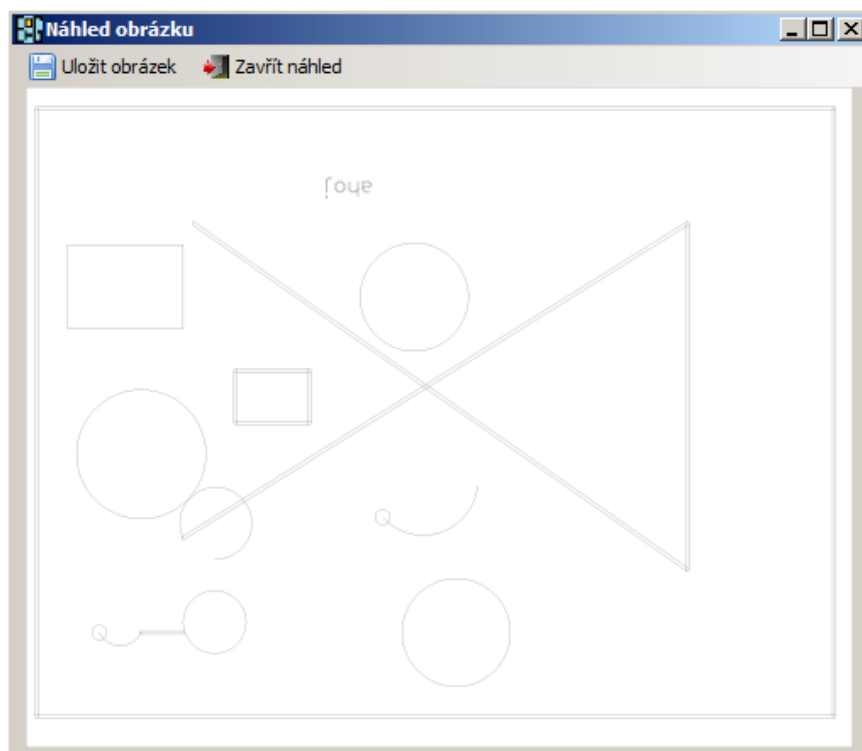
Na horní liště nalezneme tlačítko pro uzavření náhledu.



Obrázek 18 - Náhled dat

4.2.5 Náhled obrázku

U HP-GL dat máme možnost zobrazit náhled obrázku, kde si můžeme zobrazit obrázek, který se generuje ze vstupních dat.



Obrázek 19 - Náhled obrázku

V levé horní části máme lištu, na které se nachází tlačítko pro uložení obrázku a tlačítko pro zavření náhledu.

4.3 Programové třídy

Pro práci programu bylo vytvořeno několik tříd. Protože se na ně budeme v dalším textu odkazovat, tak je zde uvedeme a stručně popíšeme.

4.3.1 Třída HPGLtoG

Hlavní třída pro převod dat z HP-GL do G-kódů. Obsahuje metody pro načtení, uložení a práci s HP-GL daty.

V konstruktoru třídy předáváme nastavení programu, které obsahuje i nastavení ovlivňující převod.

Veřejné metody *otevriSoubor* a *ulozSoubor*, kterým předáváme název souboru, se starají o otevření respektive uložení souboru a načtení respektive zapsání dat z respektive do souboru. Důležitým prvkem této metody jsou kontrola možných chyb při otevírání či ukládání souboru. V případě, že nějaká chyba nastane, je o tom uživatel informován výstražným oknem.

Privátní metoda *rozdělRadky* se nám postará o rozdělení příkazů do řádků i v případě, že by byly zapsané za sebou. Díky tomu si usnadníme další práci s daty.

Privátní metoda *rozparsuj* se nám postará o rozdělení jednotlivých řádek na příkaz a parametry. Takto rozdělené řádky načítáme do spojového seznamu, který se skládá z tříd *HPGLfce*.

Privátní metoda *proved* slouží k samotnému převodu HP-GL dat do G-kódů. V této metodě procházíme spojový seznam vytvořený metodou *rozparsuj* a k funkci v HP-GL formátu najdeme ekvivalent v G-kódech a provedeme případné operace, abychom dosáhli stejného výsledku.

Další veřejnou metodou je metoda *zpracujData*. Tato metoda volá další metody. Nejprve zavolá metodu *rozdělRadky*, poté zavoláme metodu *rozparsuj* a jako poslední *proved*.

Zbylé metody slouží k přístupu k datům třídy.

4.3.2 Třída HPGLfce

Třída, kterou využijeme při převodu HP-GL dat, využijeme ve spojovém seznamu pro uchování funkcí a parametrů. Obsahuje tedy konstruktor a metody pro přístup k datům.

4.3.3 Třída *ExcellonToG*

Tato třída mý na starosti převod Excellon dat do G-kódů. Obsahuje metody pro převod dat, přístup k převedeným datům a informacích o převodu.

Stejně jako třída *HPGLtoG* obsahuje tato třída konstruktor, ve kterém předáváme nastavení programu. Velice podobné jsou i metody *otevriSoubor* a *zavriSoubor*.

Veřejná metoda *zpracujData* se nám stejně jako v případě převodu HP-GL dat postará o celý proces převodu dat mezi formáty. Oproti převodu z HP-GL v této metodě procházíme načtené řádky ze souboru a voláme na ně metodu *zmenRadek*. A výsledné řádky uloží do pole.

Privátní metoda *zmenRadek* se stará o samotný převod Excellon dat do G-kódů. Kontrolujeme přítomnost příkazů, které jsou pro nás důležité, a upravuje popřípadě chování převodu.

Metoda *reset* maže všechny nastavení a zpracovaná data, abychom mohli třídu využít pro další nebo opakovaný převod dat.

4.3.4 Třída *NastaveniStorage*

Jedná se o třídu, která pracuje s nastavením programu, konkrétně o ukládání a načítání nastavení. Nastavení máme uložené ve třídě *NastaveniData* a tuto třídu serializujeme a ukládáme respektive deserializujeme a načítáme, tím se staráme o persistenci nastavení. Třída obsahuje dvě metody.

První metoda *loadData* je volána po spuštění programu a postará se o načtení souboru s nastavením. Soubor s nastavením se ukládá do stejné složky, jako je spustitelný soubor programu. Po otevření je potřeba data deserializovat, tím získáme třídu *NastaveniData*, kterou využíváme ve zbytku programu pro získávání nastavení, které si uživatel nastavil.

Druhou metodou je *saveNastaveni*, tato metoda se volá při ukončení programu a má na starosti pokusit se uložit třídu s nastavením. Abychom mohli třídu uchovat, potřebujeme ji převést do formátu, který se dá uložit. K tomu nám slouží proces serializace, který se postará, aby byl aktuální stav třídy převeden do formátu, díky kterému bude možné v budoucnu třídu opět obnovit. V programu využíváme konkrétně serializaci do XML formátu.

Během práce na programu jsme narazili na problém, že nelze serializovat třídy, které obsahují vícerozměrné pole. V nastavení využíváme vícerozměrné pole u tabulky nástrojů. Proto tuto proměnou nejdříve převádíme do bytového pole a to poté pomocí funkce *base64* konvertujeme do textového formátu.

4.3.5 Třída *NastaveniData*

Tato třída obsahuje samotné data nastavení. Obsahuje metody pro přístup k datům. Pokud v programu pracujeme s nastavením, pracujeme s touto třídou.

4.4 Spuštění programu

Po spuštění programu načteme nastavení programu. Využijeme k tomu třídu *NastaveniStorage*. V případě, že aplikace soubor nenalezne, tak se jej pokusí vytvořit. Nastane-li taková situace, že nebude možno soubor vytvořit, tak aplikaci bude moct spustit, ale bude zamezena persistence dat. V případě, že není nalezen soubor nebo nebude nalezen, tak bude program nastaven do výchozích hodnot.

4.5 Převod Excellon dat do G-kódů

4.5.1 Načtení dat

Načtení dat probíhá po kliknutí na tlačítko „Zdrojový soubor“, po kliknutí se uživateli zobrazí standardní Windows formulář pro otevření souboru a uživatel může vybrat soubor který chce otevřít.

V případě, že během otevírání nastane chyba, je program ošetřen tak, aby uživatele upozornil a umožnil mu další pokus o otevření.

Zároveň si inicializujeme třídu *ExcellonToG*, které předáme načtené řádky a nastavení aplikace, které jsme si načetli při spuštění aplikace.

4.5.2 Převod mezi formáty

Jakmile uživatel stiskne tlačítko „Převést“, je v třídě *HPGLtoG* zavolána metoda *Convert*, která se postará o převod dat.

V případě, že máme v nastavení vybráno vrtání pomocí kódů G81 nebo G82 a zároveň využíváme tabulku nástrojů, tak budeme dotázáni, které nástroje využijeme a kde na stroji se fyzicky nachází.

Protože by převod větších souborů mohl způsobit „zaseknutí“ programu, tak tuto funkci voláme v *BackgroundWorkeru*, který se nám postará o to, že se převod přesune jako samotná úloha, samotný program tedy bude plně funkční. O průběhu převodu informujeme uživatele ve stavové řádce pomocí prvku progress bar v liště, který graficky znázorňuje stav převodu.

Nejdříve je potřeba správně načíst vstupní data na funkce a parametry. K tomu využijeme třídu *HPGLfce*. Vytvoříme spojový seznam a vytvoříme instance této třídy a postupně vkládáme načtené funkce a jejich parametry.

4.5.3 Editace dat

Pokud uživatel načte a převede vstupní data, tak má možnost po kliknutí na tlačítko „Náhled“ zobrazit si vstupní a výstupní data, uživateli se zobrazí okno rozdělené na dvě poloviny.

V levé polovině jsou vstupní data, která program načetl, zeleně jsou zobrazeny příkazy, které program převedl a červeně ty, které se programu nepodařilo převést. V pravé polovině se zobrazují převedená data. V této polovině může také uživatel provádět editaci, takže může doplnit své příkazy nebo naopak příkazy odebrat. Provedené změny se projeví ve výsledném souboru. Pokud však uživatel provede opětovný převod, tak tyto data budou ztracena.

Pokud má uživatel v nastavení vybranou volbu, že se mají nepřevedené příkazy vkládat jako komentáře, tak jsou takovéto příkazy v pravém okně vloženy jako komentáře, pokud tuto volbu vybranou nemá, tak řádky do převedených dat vkládány nejsou.

4.5.4 Uložení dat

Pokud je uživatel se vším spokojen, nezbyvá než mu umožnit data uložit. K tomu mu slouží tlačítko „Uložit jako...“, po stisknutí se zobrazí standardní Windows formulář pro uložení souboru. Uživatel zadá jméno a místo kam chce soubor uložit. Poté program uloží převedená data.

Samozřejmostí je ošetření pokud možno všech chyb a možných problémů při ukládání, jako je nezadané jméno souboru, nebo plný disk a další možné chyby. O chybách je uživatel informován a je mu tedy umožněna náprava a možnost opětovného pokusu o uložení souboru.

4.6 Převod HPGL dat do G-kódů

4.6.1 Načtení dat

Pro načtení dat zobrazíme uživateli standardní Windows formulář pro výběr souboru. V tomto formuláři si uživatel vybere soubor, implicitně jsou soubory filtrovány jen podle koncovky **.plt*, což je koncovka pro soubory s obsahem HPGL dat, ale protože by to omezovalo uživatele při otvírání nestandardních souborů, které nejsou uloženy pod touto koncovkou, je možnost zobrazit všechny soubory.

Po výběru souborů, které je ošetřeno pro různé výjimky jako je nečitelnost souboru či problému se čtením ze souboru následuje načtení souboru po řádcích do pole, se kterým budeme pracovat dále.

Zároveň si inicializujeme třídu *HPGLtoG*, které předáme načtené řádky a nastavení aplikace, které jsme si načetli při spuštění aplikace.

4.6.2 Převod mezi formáty

Jakmile uživatel stiskne tlačítko „Převést“ na záložce „HP-GL do G-kódů“, je v třídě *HPGLtoG* zavolána metoda *ZpracujData*.

Protože by převod větších souborů mohl způsobit „zaseknutí“ programu, opět voláme tuto funkci v *BackgroundWorkeru*, stejně jako u Excellon dat.

Nejdříve je potřeba správně načíst vstupní data na funkce a parametry. K tomu využijeme třídu *HPGLfce*. Vytvoříme spojový seznam a vytvoříme instance této třídy a postupně vkládáme načtené funkce a jejich parametry.

4.6.3 Kontrola dat

Po převodu má uživatel možnost překontrolovat převod. U HP-GL dat má možnost překontrolovat výstupní data pomocí integrovaného prohlížeče, kde má možnost vidět jak vstupní tak výstupní data. V případě, že nebylo možné nějaký příkaz převést, tak je tento řádek obarven. Uživatel také může do převodu zasáhnout a opravit chybný řádek, nebo dopsat své příkazy.

Pro další kontrolu má uživatel možnost náhledu i v grafické formě. Může si tedy zobrazit obrázek, který znázorňuje vstupní data.

4.6.4 Obrázkový náhled

Obrázkový náhled se generuje z Excellon dat, ale obsahuje všechny funkce, které jsou implementovány i pro převod do G-kódů, pokud by tedy byl ve vstupních datech využit příkaz, který program neumí převést, tak se nezobrazí ani v tomto náhledu, proto by měl obrázek věrně odpovídat tomu, co bude převedeno do G-kódů.

Pro kreslení obrázku využíváme předpřipravené data ze třídy *HPGLToG*. Výsledný obrázek je možno uložit. Díky tomu se dá program využít pro převod HPGL dat do formy obrázku.

4.6.5 Uložení dat

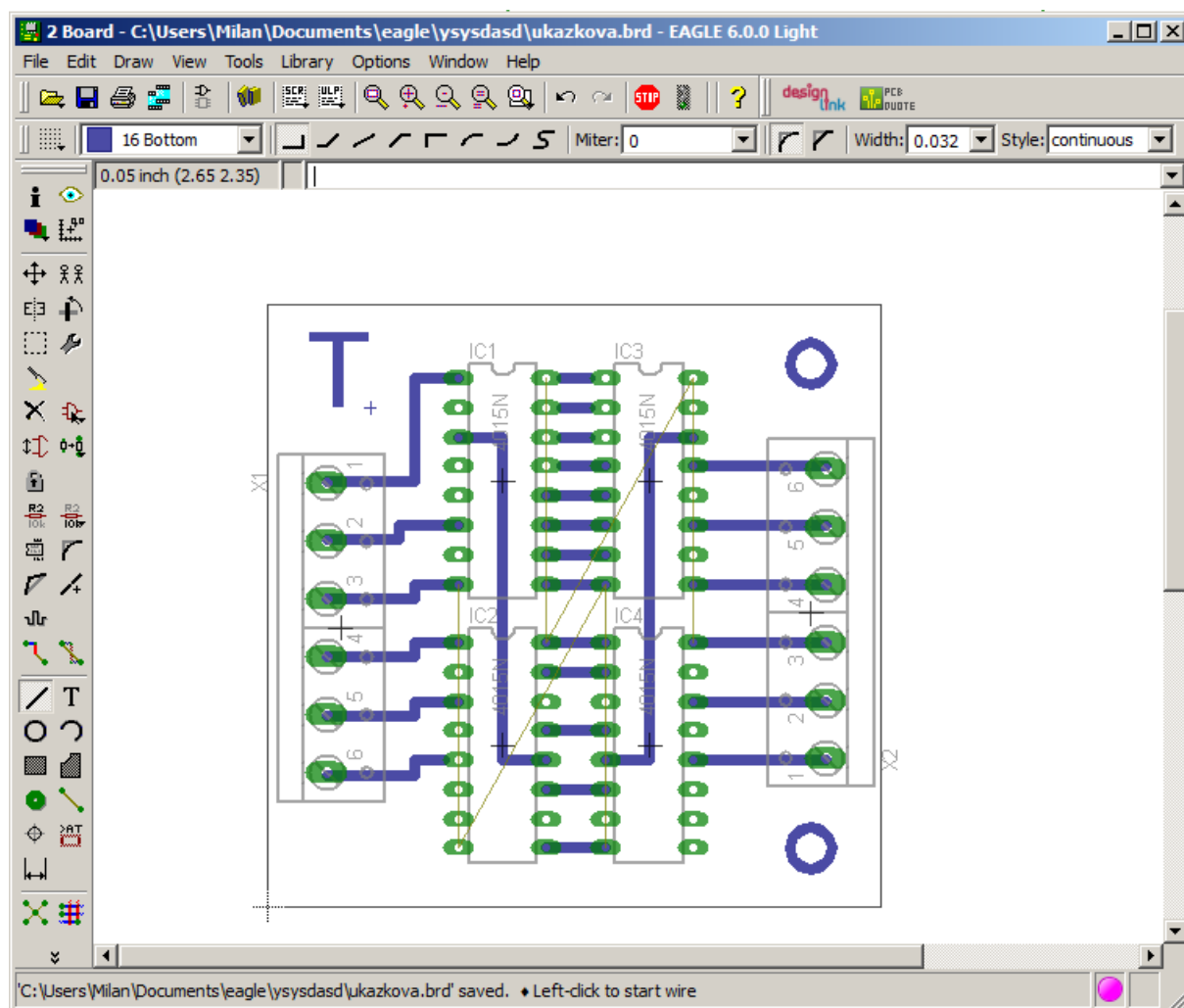
Uložení dat probíhá stejně jako u převodu z Excellon dat.

5. | Praktické požití

V této části bude ukázána práce z pohledu uživatele. Nejdříve vytvoříme vstupní data pro převod a poté je převedeme. Vstupní data budou vytvořena v programu Eagle.

5.1 Převod v programu Eagle

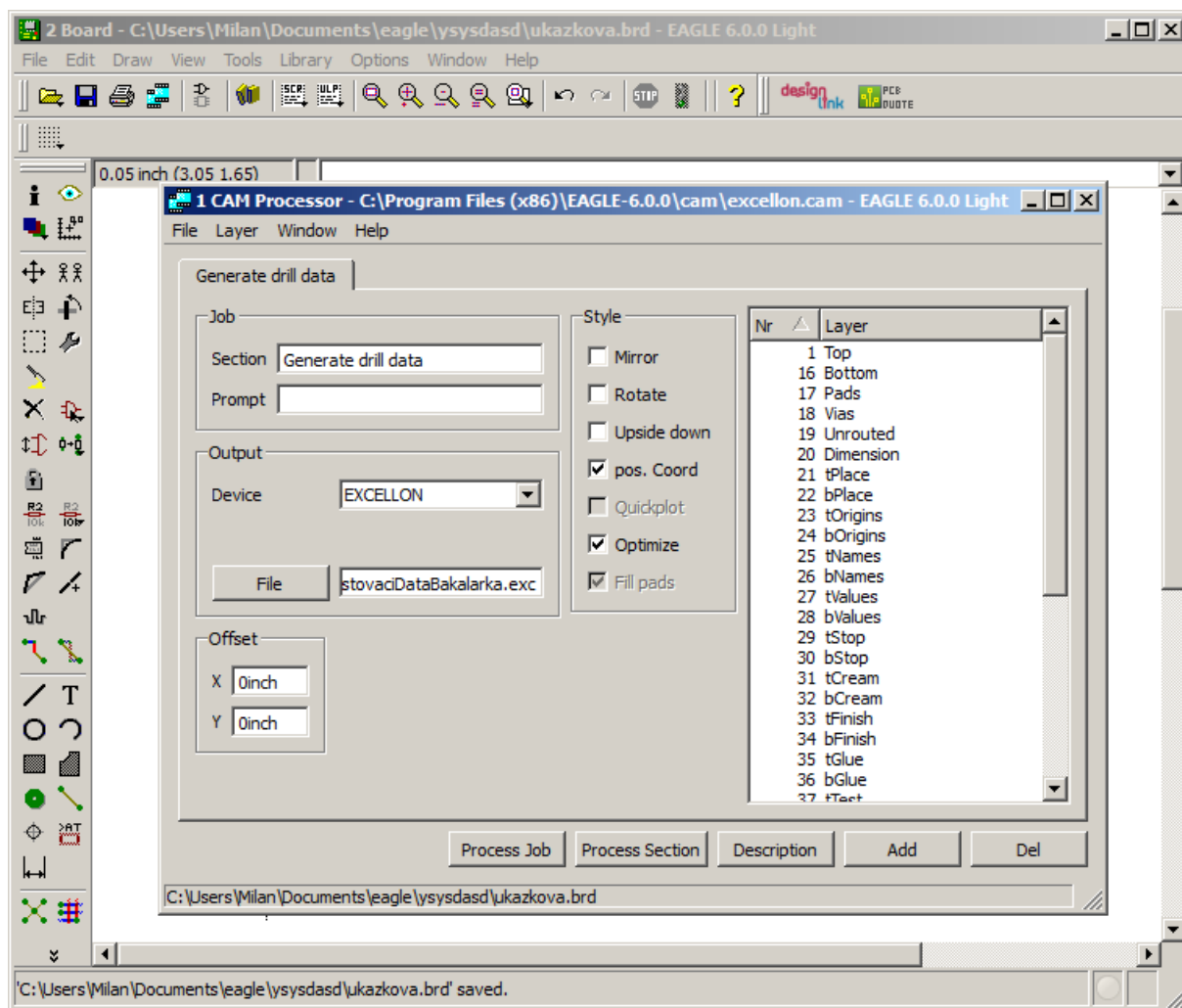
Nejdříve je potřeba vytvořit vstupní data. V programu Eagle zvlášť vytvoříme vrtací data a obrysová data. Nejdříve byla navržena jednoduchá testovací deska, kterou je možné vidět na *Obrázek 20*.



Obrázek 20 - Vytvořená DPS

5.1.1 Generování Excellon dat

Dalším krokem je vytvoření Excellon dat. K tomu využijeme CAM procesor integrovaný v programu Eagle. V menu *File* vybereme položku *CAM Processor* a zobrazí se nám okno, které můžeme vidět na *Obrázek 21*.

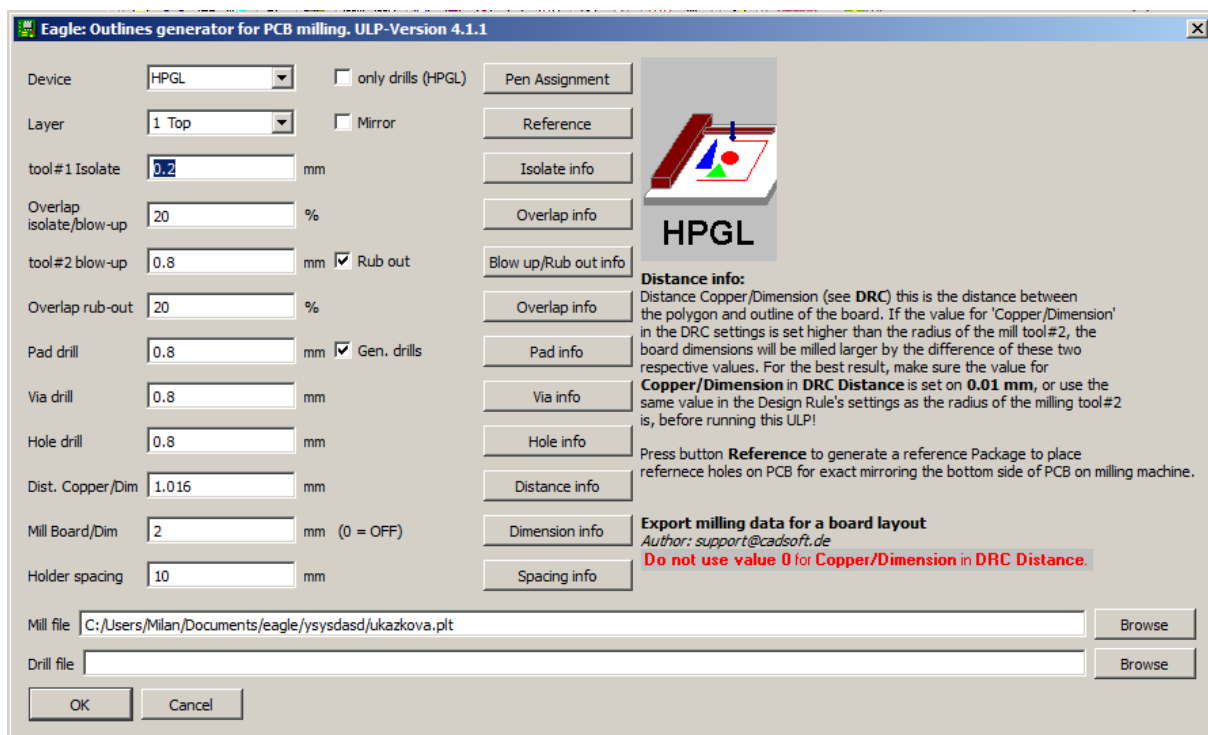


Obrázek 21 - Příprava Excellon dat

Na tomto okně je potřeba nejdříve načíst soubor, podle kterého se bude DPS zpracovávat. K tomu potřebujeme zvolit položku *Job* v menu *File – Open*. Budeme vyzváni k vybrání souboru. Vybereme soubor *excellon.cam*. Tím načteme soubor, který nám zajistí převod DPS do Excellon dat. Nyní už jen zkontrolujeme, jestli je v rozbalovacím položkovém menu *Device* vybráno *EXCELLON*. Dále klikneme na tlačítko *File* a vybereme, pod jakým názvem a kam má být uložen soubor. Poté již klikneme na tlačítko *Process Job*. Budeme informováni o průběhu převodu a až bude operace dokončena, tak máme Excellon data naší DPS. Nyní můžeme toto okno zavřít.

5.1.2 Generování obrysových dat

Dalším krokem je vytvoření obrysových dat. K tomu využijeme skript *mill-outlines*. V menu *File* vybereme položku *Run...*, zobrazí se nám okno, kde jsou dotázáni na soubor. Vybereme tedy *mill-outlines.ulp*. Otevře se nám okno, které můžeme vidět na *Obrázek 22*.



Zde máme už větší možnost nastavení. Hlavními údaji, které je potřeba nastavit, je hned první (*Device*), který musí být nastaven na *HPGL* a druhý údaj (*Layer*) musí být nastaven na tu stranu DPS, kterou chceme vygravírovat. A také je třeba v položce *Mill file* vybrat soubor, kam se mají vytvořená data uložit.

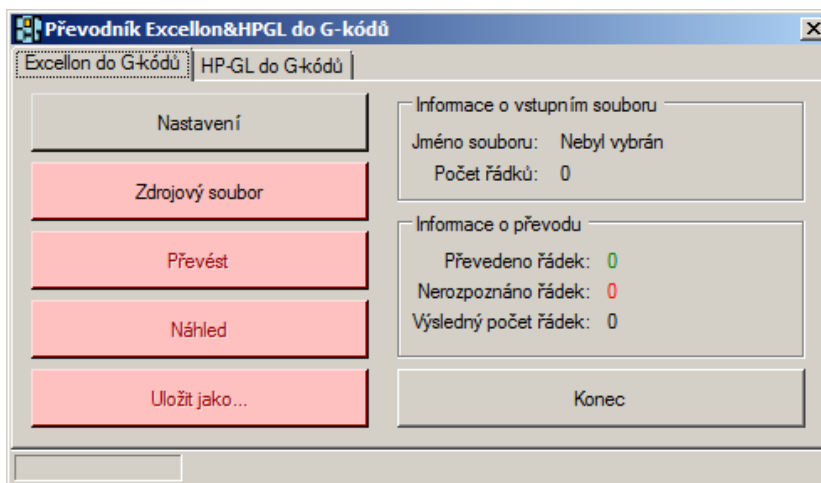
Ostatní volby můžeme nechat nastavené na nabízené hodnoty. Stačí tedy kliknout na tlačítko *OK* a do vybraného souboru budou vygenerovány HP-GL data.

5.2 Převod do G-kódů

Vstupní data pro nás program máme vytvořená, nyní již využijeme programu pro vytvoření dat pro CNC stroj.

5.2.1 Převod Excellon dat

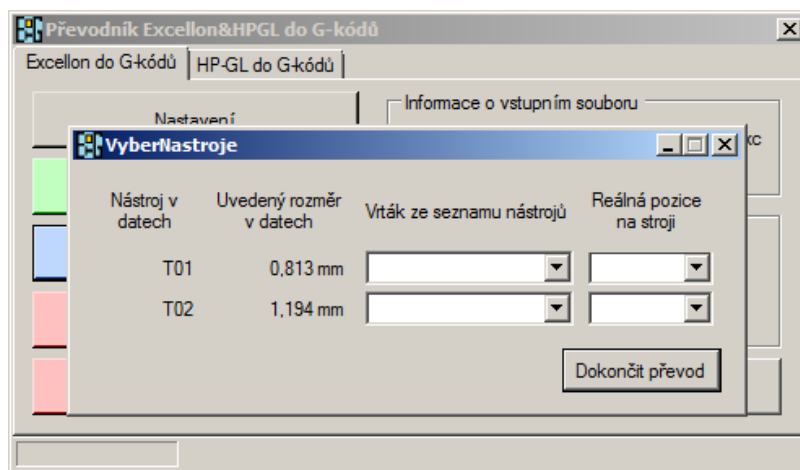
Nejprve převedeme Excellon data. Spustíme program. Na záložce „Excellon to G-kódy“ (*Obrázek 22*) nejdříve můžeme kliknout na tlačítko „Nastavení“ a nastavit parametry převodu.



Obrázek 22 - Převod Excellon dat

Prvním úkonem je otevření vstupního souboru, klikneme tedy na tlačítko „Zdrojový soubor“ a najdeme vygenerovaný Excellon soubor.

Druhým úkonem je převedení vstupních dat. Klikneme tedy na tlačítko „Převést“. Pokud jsme v nastavení vybrali vrtání pomocí G-kódu G81 nebo G82 a zároveň jsme zaškrtnuli volbu, že budeme pracovat s tabulkou nástrojů, tak se nám zobrazí tabulka jako na *Obrázek 23*, kde je potřeba vybrat, jaké vrtáky z tabulky nástrojů jsou použity a jakou pozici bude mít vrták ve výstupních datech.



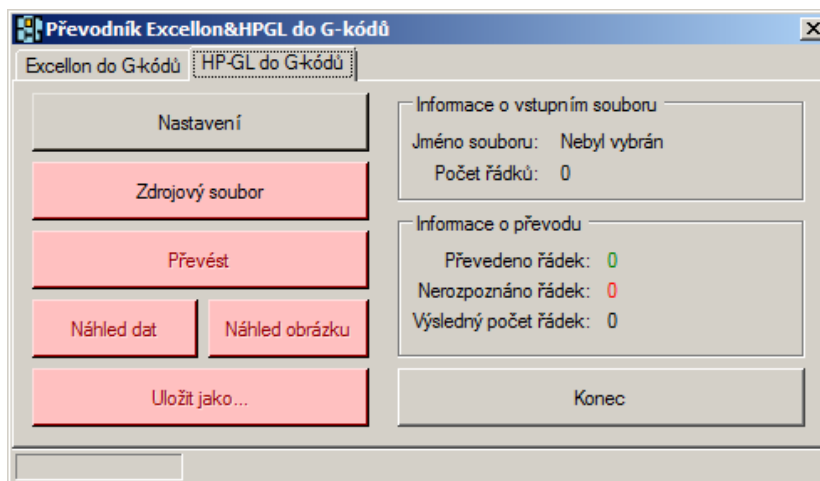
Obrázek 23 - Tabulka pro výběr vrtáků

Pokud klikneme na tlačítko „Náhled“, tak se nám zobrazí náhled dat, kde můžeme provést úpravy.

A jako poslední krok klikneme na tlačítko „Uložit jako...“ a výstupní data uložíme.

5.2.2 Převod HP-GL dat

Po spuštění programu se přepneme na záložku „HP-GL do G-kódů“ (*Obrázek 24*). Pokud chceme provést nastavení ovlivňující průběh převodu, tak klikneme na tlačítko „Nastavení“.



Obrázek 24 - Převod HP-GL dat

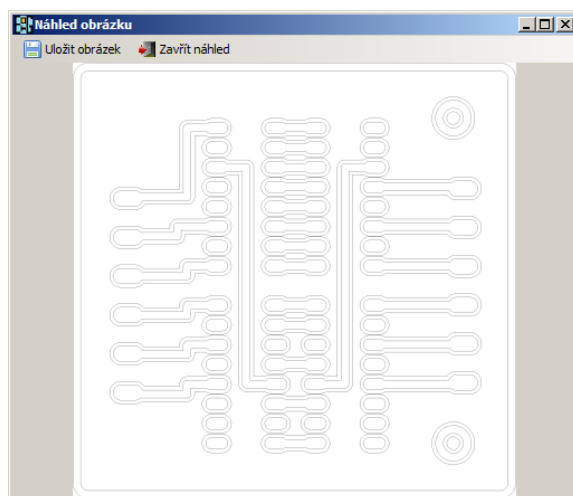
Prvním úkolem pro úspěšný převod je otevření vstupního souboru. K tomu použijeme tlačítko „Zdrojový soubor“ a vybereme soubor, který jsme si dříve vygenerovali v Eagle.

Po načtení zdrojového souboru se nám povolí tlačítko „Převést“ na které klikneme. Po chvilce bude převod dokončen a zároveň se nám umožní kliknout na tlačítka „Náhled dat“ a „Náhled obrázku“.

Pokud chceme nahlédnout na data, klikneme na „Náhled dat“ stejně jako u Excellon převodu.

Chceme-li se kouknout na obrázkový náhled, klikneme na tlačítko „Náhled obrázku“ a zobrazí se nám okno s náhledem jako na *Obrázek 25*.

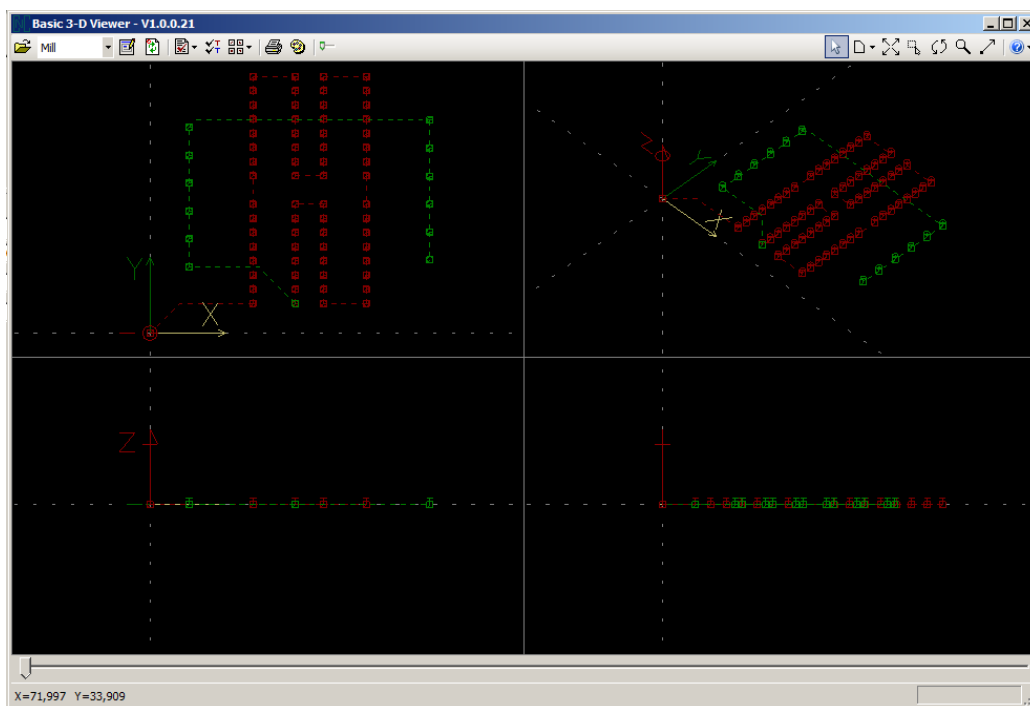
Pokud jsme spokojeni, tak zbývá už jen data uložit. Klikneme na tlačítko „Uložit jako...“ a data uložíme.



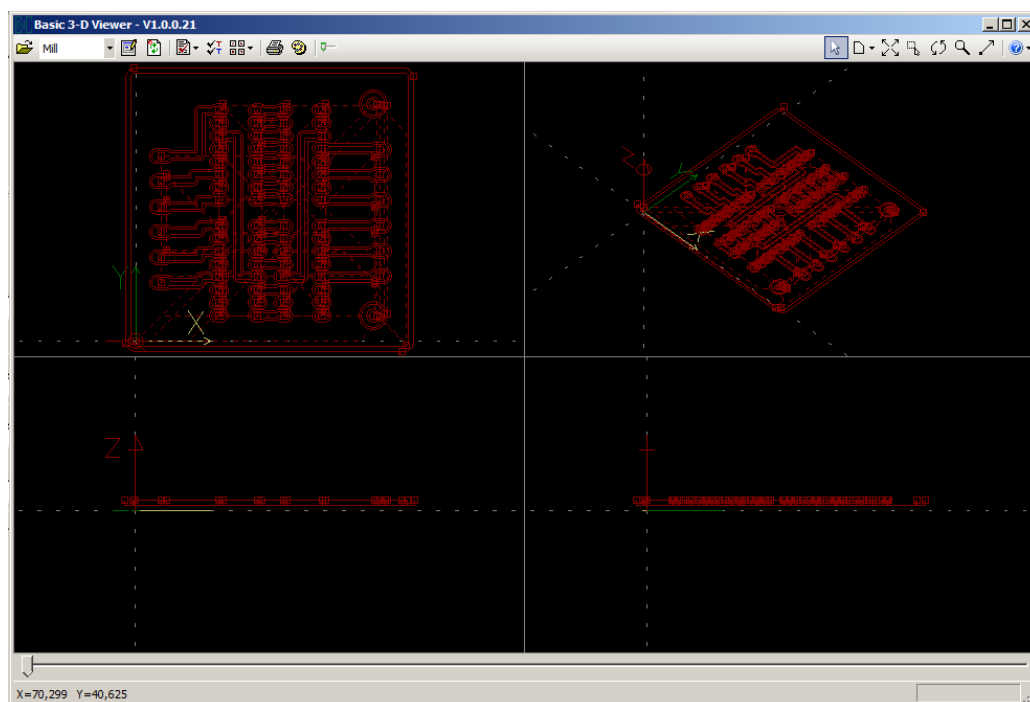
Obrázek 25 - Náhled obrázku

5.3 Náhled výstupních dat

Výstupní data v G-kódech máme připraveny a stačilo by je načíst do stroje a nechat stroj vykonat svoji práci. Na obrázku *Obrázek 26* a *Obrázek 27* můžeme vidět výstupní data v programu „Basic CNC Viewer“, který umí vizualizovat data zapsané v G-kódech.



Obrázek 26 - Převedená Excellon data



Obrázek 27 - Převedená HP-GL data

6. | Závěr

Program byl vyzkoušen na datech vygenerovaných v programu Eagle, kde se také očekává jeho využití. I když se nejednalo o složitý vývoj na algoritmus, tak i přes to některé převody bylo potřeba velice dobře rozmyslet (hlavně u HP-GL dat, kde se pracuje třeba s kružnicemi), aby převedená data odpovídala vstupním datům. Naopak zase u Excellon dat bylo potřeba dát pozor na možnost uvádění hodnot v milimetrech a palcích. Největší problém se nacházel při načítání dat a různých možnostech zápisu.

Cíle projektu byly naplněny. Převod Excellon dat a HP-GL dat je plně funkční, až na omezené možnosti převodu, protože jsme neobsáhly všechny možné instrukce. Ale to není vzhledem k plánovanému využití potřeba, kdyby bylo potřeba nějaké příkazy přidat, tak je určitě možnost program rozšířit. Vzhledem k nalezeným datům na internetu, které byly na program vyzkoušeny, by i tak měl program být schopen převést většinu vstupních dat.

Během vývoje byly použity nástroje k vizualizaci dat a v několika případech v této vyvinuté aplikaci docházelo k lepším výsledkům, než v nástrojích k tomu určených.

Určitá budoucnost programu je v jeho optimalizaci, ne všechny převody jsou provedeny ideálně a daly by se optimalizovat.

Výstupní data byla úspěšně vyzkoušena na CNC stroji, který laboratoř vlastní. Jednalo se o typ bez automatické výměny nástrojů, jinak byl převod úspěšný a výsledek odpovídal zadání.

Literatura

- [1] Miroslav Štulpa – CNC obráběcí stroje a jejich programování, BEN. 2005, ISBN 978-80-73002-07-7
- [2] *Excellon manual* [online]. 1. verze. 2005 [cit. 2012-05-14]. Excellon manuál. Dostupné z WWW: <<http://www.excellon.com/manuals/program.htm>>.
- [3] *G Code Basics* [online]. 2001 [cit. 2012-05-14]. RS274NGC G-CODE PROGRAMMING BASICS. Dostupné z WWW: <<http://www.linuxcnc.org/handbook/gcode/g-code.html> >.
- [4] HEWLETT-PACKARD. *The HP-GL/2 and HP RTL Reference Guide: A handbook for Program Developers* [online]. 2. vyd. 1996 [cit. 2012-05-14]. ISBN 0-201-63325-6. Dostupné z: <http://www.hp-museum.net/document.php?catfile=213>
- [5] CNCEZPRO™ SOFTWARE. *CNC G Codes Definitions Examples Programs Programming Learning Training* [online]. [cit. 2012-05-14]. Dostupné z: <http://www.cncezpro.com/gcodes.cfm>
- [6] Vektorový grafický formát HPGL. In: *Root.cz* [online]. 2007 [cit. 2012-05-14]. Dostupné z: <http://www.root.cz/clanky/vektorovy-graficky-format-hpgl/>
- [7] *HP-GL Reference Guide*. ISOPLOTEC CORPORATION. [online]. [cit. 2012-05-14]. Dostupné z: <http://www.isoplotec.co.jp/HPGL/eHPGL.htm>
- [8] ARISTO. *HPGL Database Description* [online]. [cit. 2012-05-14]. Dostupné z: http://web.me.com/europlotter/europlotter/Machines_files/hpgl.pdf
- [9] Machine Code Reference. *Tormach* [online]. 2012 [cit. 2012-05-14]. Dostupné z: http://www.tormach.com/machine_codes.html