

Merick Calc 3000 - Obrábění (verze 1.1 z 27.11.2012 aplikovaná v programu 1.1.31.1354)

Úvodem

Obrábění je v současnosti jedna z funkcí plošného dílu. Každému plošnému dílu můžete definovat obrábění na jednom nebo více stroji, například obráběcím centru a NC olepovače.

Obrábění funguje na dvou úrovních vytváření řídicích programů, které se mohou vzájemně prolínat.

Jedním z vytváření definice obrábění je definice vrtání. To je určené pro stroje, které program Merick Calc 3000 zná a dokáže je sám řídit (generovat pro ně programové soubory). Uživatel pouze definuje základní parametry vrtání jednotlivých děr a program je ve výrobě přetvoří na kód pro stroj. Tímto způsobem jdou zatím definovat jen vrtání.

Jiné obrábění, nebo například olepování lze definovat pomocí "Volného kódu". Tento způsob generování NC kódu umožňuje definovat i jiné obrábění, než jen vrtání. Stejně tak umožňuje řídit i stroje, které nemají přímou podporu v programu Merick Calc 3000. Mohou to být samozřejmě kromě obráběcích center (OC) i jiné NC stroje, jako jsou například olepovačky, manipulátory, skladovací a balící systémy a podobně.

Programování NC strojů přes Obrábění přináší mnoho výhod, oproti přímého programování přes jednotlivé soubory. Hlavní výhody jsou následující.

- veškerá definice obrábění je uvnitř každého plošného dílu. Tím pádem i případné úpravy se vždy vztahují stále jen na konkrétní plošný díl.
- z toho vyplívá i přesná identifikace plošného dílu během celého výrobního procesu a možnost využívat jednoznačného čárového kódu, který má každý plošný díl pro bezchybné přiřazení s generovanými NC programy.
- možnost využívat základní parametry plošného dílu v NC programu. To pomůže například při záměně materiálu nebo přiřazení materiálu přes materiálového zástupce. Nemusíte tedy hlídat například tloušťku plošného dílu, při změně materiálu.
- u vrtání se současně generuje i schéma, které může být také vytištěno na nálepce/průvodce plošného dílu
- možnost využívat vlastní knihovny, opakujících se definic obrábění
- definice obrábění je také zakomponována do modelářů, takže i zde můžete využít automatického generování NC programů, bez rozpadu parametrického modelu na jednotlivé plošné díly.

Definice základních položek Obrábění

Definici obrábění provádíme ve Vlastnostech plošného dílu, na záložce "Obrábění".

Plošný díl

Jednoduchá skříňka

Název dílu: Dvířka
Plošný materiál: DTDL28-854SU WENGE Otočit rubem nahoru
Počet kusů a Pozn.: 2

Tvar a rozměry plošného dílu: **Obrábění** | Informace o přifezu | Dokumenty

Obdélník

Rozměry: X 396,00 mm Y 720,00 mm

Orientace vzoru (léta): X Y Podélně Příčně Je to jedno

Hrana: Hrana 2,0-buk

Olepení hranou: X Y Z
 Započítat tloušťku hrany do rozměru dílu

Hrana 1 Hrana 2 Hrana 3 Drážka

Uložit jako výchozí OK Storno

Olepení hranou 1
Olepení příslušné strany dílu hranou 1.
Pozn.: Zadané olepení hranou 1 je schématicky zobrazeno na obrázku červenou čarou.

Do seznamu položek obrábění můžeme přidávat tři typy položek.

Parametr

vlastní číselná nebo textová proměnná, která se používá vícekrát v následujících definicích obrábění. Tím, že se deklaruje před vlastním použitím v obrábění, získáváme lepší přehled o její hodnotě (nepočítá se uvnitř ostatních položek). I případné opravy hodnot parametrů se dělají jen jednou a celý zápis obrábění se stává přehlednější a míň náchylný k chybám. Hodnoty parametrů, až na výjimky, jsou číselné hodnoty, přímé, nebo vypočítané pomocí matematických výrazů. V matematických výrazech fungují priority. Na názvy parametrů je vhodné používat znaky anglické abecedy, číslice, nepoužívat mezery. Názvy nejsou case-sensitivní, tedy nerozlišují malá a velká písmena. (Abc = abc = ABC = aBC ...). Název parametru musí vždy začínat písmenem, nikdy ne číslicí! Existují i systémové, implicitní parametry, které se vztahují k obráběnému plošnému dílu. Jsou to:

PX hodnota zadaná jako rozměr X plošného dílu
PX1 hodnota rozměru X uříznutého plošného dílu (rozměr "z pily")
PX2 hodnota rozměru X hotového plošného dílu (uříznutý, obrobený, olepený)
PY hodnota zadaná jako rozměr Y plošného dílu
PY1 hodnota rozměru Y uříznutého plošného dílu (rozměr "z pily")
PY2 hodnota rozměru Y hotového plošného dílu (uříznutý, obrobený, olepený)
PZ tloušťka plošného materiálu

Od verze 1.1.31.1349 jsou k dispozici nové systémové parametry.

PX3 výsledný rozměr dílu v ose X po olepení

PY3 výsledný rozměr dílu v ose Y po olepení

Count počet dílů

NoteExt poznámka. Textová hodnota, používá se ve Volném kódu jako <%=NoteExt%>

Příklady Hodnot:

PantOffsetX=16

PantOffsetY=100

PantRoztecY=PY-2*PantOffsetY

UchytkaOffsetX=32

UchytkaOffsetY=100

UchytkaRoztec=96

Přirozeně záleží na pořadí jednotlivých parametrů. Nejprve musíme deklarovat parametr a pak teprve je můžeme používat ve vzorcích a dalších položkách Obrábění.

Vrtání

jednoduché definování jednoho nebo více pravidelně se opakujících stejných otvorů. V parametrech rozměrů a souřadnic můžete kromě číselné hodnoty psát i matematické výrazy a používat implicitní a předem definované explicitní parametry.

Nejlépe to znázorňuje následující obrázek:

Vrtání

Dvířka

Název operace

Umístění, velikost a typ otvoru

Plocha

X

Y

Hloubka

Průměr

Opakování otvoru

Počet otvorů

Posun X

Posun Y

Opakování otvoru

Typ díry a Zahloubení

Posuv a Vyklízení

Volný kód

je to velmi silný nástroj, umožňující kompletní řízení NC stroje přímo z prostředí programu Merick Calc 3000. Umožňuje zadávat kód pro přímé řízení stroje do generovaného NC programu. Využívá se to ve dvou případech. Merick Calc 3000 umí řídit stroj, tedy umí vrtání a my potřebujeme přidat další obrábění na plošném dílu, jako například různé tvarové frézování, drážky a podobně nebo Merick Calc nezná a tím pádem neumí řídit náš NC stroj.

Připravíme si tedy NC program pro další obrábění v NC editoru dodávaným se strojem (CAM) a výsledný kód, pokud je to textový formát vložíme jako Volný kód.

Výhodou je, že v tomto přidaném NC kódu můžeme vkládat parametry stejně jako ve vrtání. Takže ze statického NC kódu, vzniká pružný NC kód, který můžeme beze změny používat v různých plošných dílech stejného typu. Vkládání implicitních/definovaných parametrů se provádí pomocí speciálního formátu podobnému ASP z webových služeb. Nejlepší bude zase příklad:

Příklad modifikace statického NC kódu pomocí vložených parametrů

Takto vypadá statický zápis hlavičky NC programu OC SCM

H DX=396 DY=720 DZ=18 -AB C=0 T=0 R=1 BX=0 BY=0 BZ=0 *MM /"def"

a takto by mohl vypadat pružný zápis hlavičky vložený jako Volný kód

H DX=<%=PX%> DY=<%=PY%> DZ=<%=PZ%> -AB C=0 T=0 R=1 BX=0 BY=0 BZ=0 *MM /"def"

Jak je zde vidět, parametry je možné sem vkládat pomocí "ASP" konvence <%=Parametr%>

Kromě parametrů je zde možné také provést matematické operace stejné jako při definici parametrů a vrtání.

Pro některé stroje je potřeba speciálně formátovat výsledné číselné hodnoty, doplňovat před a za nuly, mezery nebo jiné znaky, zaměnit desetinnou čárku a za desetinnou tečku a podobně. K tomu slouží ve volném kódu formátovací direktivy.

DECIMALSEPARATOR - mění výchozí, systémový desetinný oddělovač, u nás ", " čárka za jiný, nejčastěji za tečku "."

Příklad, hodnota proměnné P1 je 123,45:

<%=P1%> //123,45

<%=DECIMALSEPARATOR="."%> //přepne desetinný oddělovač na tečku

<%=P1%> //123.45 desetinný oddělovač je tečka

```
<%=DECIMALSEPARATOR=""%> //vrátí desetinný oddělovač zpět na výchozí, systémový (u nás na čárku)
<%=P1%> //123,45 a zpět na výchozí systémovou hodnotu, tedy čárku
```

FORMAT - velmi užitečná a programátory známá funkce pro formátování řetězců. Zde se používá hlavně pro ovlivnění numerických formátů. Tato "funkce" má dvě interní číselné proměnné, jedno desetinné číslo a jedno celé, s oříznutou desetinnou částí. Použití proměnné FORMAT si ukážeme dále na příkladech, až po popisu doplňkové formátovací proměnné PREFIXFORMATCHAR.

PREFIXFORMATCHAR - určuje znak, kterým je doplněno číslo zepředu, pomocí proměnné FORMAT. Standarně je číslo doplněno mezerami, pokud ale potřebujete nuly"0", tak použijte tuto proměnnou. Příklad použití bude společně s proměnnou FORMAT.

Příklady formátování pomocí proměnné FORMAT:

```
<%=P1%> //123,45
<%=DECIMALSEPARATOR="."%>
<%=P1%> //123.45
<%=FORMAT="%12.3f"%>
<%=P1%> // 123.450
<%=FORMAT="%d"%>
<%=P1%> //123
<%=FORMAT="%8d"%>
<%=P1%> // 123
<%=PREFIXFORMATCHAR="0"%>
<%=P1%> //00000123
<%=FORMAT="%12.3f"%>
<%=P1%> //00000123.450
```

#CharCode - vrací znak uvedeného čísla. Používá se pro některé znaky, které editor nezobrazuje, jako znaky 1-31. Touto funkcí nelze přímo vložit znak #0. To by způsobilo ukončení textového řetězce, takže tento znak je nahrazen postprocess funkcí, která je vysvětlena dál, v tomto dokumentu.

Poznámky ve Volném kódu - pro vyšší srozumitelnost je občas potřeba přidat do kódu poznámky, které se nepřenesou dále do NC programu a přitom nejsou vyhodnoceny jako neznámý příkaz nebo direktiva. Poznámky mohou být i na více řádcích. Více ukážou následující příklady:

Příklad poznámek:

```
<%= 'a tady si můžete psát co chcete, překladač to ignoruje%>
```

```
<%= 'a tady si můžete psát co chcete, překladač to ignoruje
a může to být i na více řádcích%>
```

```
<%= "kdo neví, jak se píše apostrof, může použít i uvozovky, funguje to stejně...%>
```

```
H DX=396 DY=720 DZ=18 -AB <%= "poznámka může být kdekoliv, i uprostřed NC kódu%>C=0 T=0 R=1 BX=0 BY=0
BZ=0 *MM /"def"
```

INCLUDECODE("FileName") - vloží kód ze souboru FileName do tohoto Volného kódu. Možno použít na stále stejné, opakující se části NC programu. Pokud momentálně není soubor k dispozici (může být vytvořen později, nebo může být přístupný až z PC, které připravuje výrobu), vypíše se řádek ve formě postprocess poznámky, vy ho tedy uvidíte, ale výsledný kód to neovlivní.

Ve vloženém kódu samozřejmě mohou být další funkce a proměnné. Pozor na rekurzi (cyklické volání funkce INCLUDE), není ošetřena!

Název souboru může být i relativní, vzhledem k implicití složce "ToolingDir", která je standarně "%ProgramDir%\Obrábění". Což se může hodit hlavně v síti. Více o standarních složkách a jejich explicitnímu určení najdete v dokumentaci k mc3k.ini

Příklad:

```
<%=INCLUDECODE("C:\NCSTROJ\LIB\TOOLSET01.TXT")%> //absolutní cesta
<%=INCLUDECODE("\NCCSCM\TOOLSET01.TXT")%> //relativní cesta, lepší pro práci v síti
```

INCLUDEFILE("FileName") - vloží obsah souboru do výsledného souboru, ovšem bez dalšího překladu a až při vytváření definitivního souboru. Slouží pro sestavování netextových NC programů. V náhledu bude nahrazen opět formou postprocess kódu. Název souboru může být také relativní

Příklad:

```
<%=INCLUDEFILE("C:\NCSTROJ\LIB\TOOLSET01.BIN")%> //absolutní cesta
<%=INCLUDEFILE("\NCCSCM\TOOLSET01.BIN")%> //relativní cesta, lepší pro práci v síti
```

MACHINE - proměnná, která umožňuje dělit jeden kód pro více strojů. Vztahuje se na generovaný kód pro celý plošný díl, tedy jen v rámci jednoho volného kódu. Jako hodnota může být použit Název stroje nebo jeho Zkratka.

Příklad, jeden Volný kód obsahuje data pro OC a NC olepovačku

```
<%=MACHINE="NCCSCM"%>
toto je kod generoovany pouze pro SCM OC
toto je kod generoovany pouze pro SCM OC
toto je kod generoovany pouze pro SCM OC
<%=MACHINE="NCTORNADO"%>
toto je kod generoovany pouze pro NC olepovačku Tornado
toto je kod generoovany pouze pro NC olepovačku Tornado
toto je kod generoovany pouze pro NC olepovačku Tornado
<%=MACHINE=""%> //toto vypne určení stroje, další kód by byl společný pro oba stroje=blbost
```

ASSIGN(SourceStr | DestStr | CaseSensitive) - funkce, pro vytvoření assignační tabulky, která se uplatní při načítání pomocí funkce INCLUDECODE. Pokud potřebujete zaměnit některé statické řetězce v souboru například za dynamické proměnné, potom si můžete před načtením vytvořit assignační tabulku, ve které máte páry pro záměnu řetězců. Vzhledem k tomu, že pomocí funkce Assign lze zaměnit libovolné řetězce, má tato funkce poněkud jinou syntaxi, než ostatní funkce. Jediný povinný znak mezi závorkami je separátor, oddělující zdrojový a výsledný řetězec. Všechny ostatní znaky mezi závorkami funkce se považují za platné znaky parametrů.

Pokud má při hledání SourceStr rozlišovat malá a velká písmena, pak CaseSensitive=1 jinak 0 nebo není třeba neuvádět.

Pokud není DestStr uveden, dojde pouze k vymazání SourceStr

Poznámka: Pokud potřebujeme v DestStr použít kombinaci znaků "%>", která jinak znamená konec dynamické proměnné nebo funkce v systému MC3K, je potřeba assignaci provést přes jiný, nepoužívaný znak, například "⌘" prasátko (levé Alt+164), podle následujícího příkladu:

```
<%=assign(ANZ="1"|ANZ="<⌘=Count⌘>")%>
<%=assign(⌘|%)%>
<%=includecode("9920010001.mpr")%>
```

Ukázka jednoduchého kompletního příkladu použití funkcí ASSIGN pro OC WEEKE a změnou výstupního formátování čísla:*

```
<%=assign(ANZ="1"|ANZ="<⌘=Count⌘>")%>
<%=assign(_BSX=600.000000|<⌘=DECIMALSEPARATOR="."⌘><⌘=FORMAT="% .6f"⌘>_BSX=<⌘=PX3⌘>)%>
<%=assign(_BSY=328.500000|_BSY=<⌘=PY3⌘>)%>
<%=assign(_BSZ=18.000000|_BSZ=<⌘=PZ⌘>)%>
<%=assign(_RX=600.000000|_BSX=<⌘=PX3⌘>)%>
<%=assign(_RY=328.500000|_BSY=<⌘=PY3⌘>)%>
<%=assign(l="600"|<⌘=FORMAT=" "⌘>l="<⌘=PX3⌘>")%>
<%=assign(b="328.5"|b="<⌘=PY3⌘>")%>
<%=assign(d="18"|d="<⌘=PZ⌘>")%>
<%=assign(⌘|%)%>
<%=includecode("9920010001.mpr")%>
```

Poznámka:

Samozřejmě, nemusíte mít mít všechny příkazy ASSIGN zapsané přímo do volného kódu, ale můžete využít funkci INCLUDECODE a k němu vytvořit soubor WEEKE.txt ve kterém jsou příkazy ASSIGN uvedeny, pokud se stále opakují.

```
<%=includecode("WEEKE.txt")%>
<%=includecode("9920010001.mpr")%>
```

ASSIGNSEPARATOR="|" - standardně je separátor tvořen znakem "pipe" "|" (pravé Alt+W). Pokud se tento znak používá i v parametrech funkce ASSIGN, pak je možné si zvolit jiný, nepoužívaný znak.

Příklad, když je potřeba použít jiný znak jako separátor mezi řetězci SourceStr a DestStr:

```
<%=ASSIGNSEPARATOR="@"%>
```

způsobí, že místo znaku "|" se jako separátor považuje znak "@". Funkce ASSIGN pak vypadá následovně:

```
<%=assign(ANZ="1"@ANZ="<#=<Count#>")%>
```

Zrušení uživatelského separátoru a nastavení do výchozí hodnoty "|" lze provést například příkazem:

```
<%=ASSIGNSEPARATOR=""%>
```

ASSIGNCLEAR - funkce bez parametrů, která vymaže dosavadní tabulku assignací. Není nutno použít, pokud nepotřebujeme načítat více souborů funkcí INCLUDECODE s různými asignačními tabulkami.

POSTPROCESS kód <!-- ABCD -->

Některý kód nelze definovat přímo v textovém editoru, protože se může jednat o obecná, netextová data. Proto Merick Calc 3000 umožňuje vkládat kód, který se bude vykonávat až při vytváření výsledného souboru s NC kódem.

Postprocess kód začíná kombinací znaků: <!-- a končí kombinací znaků -->, tedy zápis vypadá nějak takto:

```
<!-- nějaký text nebo postprocess funkce -->
```

Jak již bylo výše uvedeno, některý postprocess kódy vkládá přímo program (INCLUDEFILE, CHR...), ale můžete je stejně tak vložit do kódu ručně. Protože funkce vychází z výše uvedených funkcí a proměnných, uvedu jen jejich výslednou podobu na příkladech, bez velkého vysvětlování. Většinou není nutné vkládat přímo postprocess kódy do Volného kódu, ale je to možné.

Příklady postprocess kódů

```
<!-- poznámka, poznámka, poznámka... libovolný text,  
může být i na více řádcích -->
```

```
<!-- poznámka je všechno, co nelze přeložit podle níže uvedených pravidel -->
```

```
<!-- CHR(0) --> //vloží do výsledného souboru znak #0. povolený rozsah je 0..255
```

```
<!-- CHR(9) --> //vloží do výsledného souboru znak TAB
```

```
<!-- CHR(13) --><!-- CHR(10) --> //vloží do výsledného souboru znaky LF a CR
```

```
<!-- MACHINE="NCCSCM" --> //další kód je generovaný jen pro stroj s touto Zkratkou
```

```
<!-- MACHINE="" --> //další kód je generovaný opět pro všechny NC stroje bez rozdílu
```

```
<!-- INCLUDEFILE("C:\NCSTROJ\LIB\TOOLSET01.BIN") --> //vloží do výstupního souboru požadovaný  
soubor. Pozor, tady už je možná jen kompletní, absolutní cesta. Pokud soubor nenajde, je to Váš problém, chová se jako  
by to byla poznámka. Proto je lepší využívat funkci <%=INCLUDECODE("FNAME")%> a když k tomu není nějaký  
rozumný důvod, tak nepsat INCLUDECODE pomocí postprocess kódu
```

Výchozí obrábění pro všechny plošné díly v MC3K (od verze 1.1)

V Rozšířeném nastavení programu (Výroba - Výchozí obrábění neobráběného dílu) lze nastavit seznam položek Obrábění, které jsou doplněny do výsledného plošného dílu ve Zpracování nářezových plánů. Pokud plošný díl při vstupu do Zpracování NP nemá vlastní definici Obrábění, pak dostane přidělenou společnou definici z Nastavení.

Používá se pro vytvoření prázdných šablon NC kódu, pojmenovaných a uspořádaných podle čárového kódu. Do prázdných šablon potom programátor doplňuje konkrétní obrábění.

Seznam je tvořen textovými řádky, kde v každém řádku je definice jednoho typu položky.

Řádek typu "Parametr"

```
ToolingValue|ParameterName|Definition
```

Příklad:

```
ToolingValue|VertCenter|PY3/2
```

Řádek typu "Vrtání"

```
Drilling|DrillingName|SideIndex|XDef|YDef|ZDef|DiameterDef|CountDef|DXDef|DYDef|DrillingKind|Recess*10000  
|Movement*10000|Clearance [
```

Řádek typu "Volný kód"

ToolingCode|CodeName|Definition

pokud je Definition na více řádků, použije se místo CRLF znak ¶ (0182).
Znak "|" (pipe) není dovolen.

Příklad:

```
ToolingCode|PRG1|<%=INCLUDECODE("DEFAULTPRG.TXT")%>
```

Makra položek Obrábění (zatím v MC3K vypnuto)

Při programování NC strojů pro výrobu nábytku se většina kódu často opakuje, byť s drobnou modifikací. Protože Merick Calc 3000 umožňuje silně parametrizovat připravovaný kód, bylo by škoda si často opakované sekvence neuložit pro další použití. Například vrtání na dvířkách je stejné, ať jsou dvířka 720x400 nebo 680x350. Při správném využití parametrů bude sekvence obrábění úplně stejná pro obě velikosti dvířek (a mnoho dalších také).

Ve standardní instalaci programu Merick Calc 3000 jsou dodáváná základní makra pro nábytkářskou výrobu, ale vy si můžete samozřejmě vytvářet a přidávat další, svoje, podle potřeby.