



Jak lze vytvořit...

Vítejte v rubrice, která se v jednotlivých volně navazujících kapitolách zabývá NC programováním a obráběním základních konstrukčních prvků. Každý z příkladů je připraven jen pro krátkodobou internetovou prezentaci a následně je nahrazen novějším. Jednotlivé příklady na sebe přímo nenavazují, avšak postupně se zaměřují na hlubší rozbor dílčích "problémů" (např. možnosti při NC programování, problematika volby nástrojového vybavení, optimalizace řezných podmínek, atd.). Pro větší názornost jsou jednotlivé příklady zpracovány a prezentovány s podporou jednoho z nejužívanějších řídicích systémů v ČR, Sinumeriku 840D a výukového programu SinuTrain.

Zpracoval: Ing. Aleš Polzer, Ph.D.

Pokročilé metody programování – č. 1

Úvod

Dnešní informativní příklad věnuje pozornost programovým skokům a smyčkám. Tuto problematiku je možno zařadit do oblastí tzv. pokročilých metod NC programování, které lze úspěšně aplikovat při zápisu formou ISO kódu v textových editorech (v některých případech i v editorech pro dílensky orientované programování). Tvorba NC programů s podmínkami a smyčkami vede k možnosti adaptivního řízení, ale jednotlivé funkce je možno úspěšně využít i při programování běžné výrobní dávky součástí. Využitelnost je i v případech, kdy je běžný automatický chod NC programu operátorem (obsluhou) přerušen např. z důvodu poškození nástroje nebo nevhodné hodnoty korekce. Můžeme vepsat do NC programu nepodmíněný skok a bloky, v nichž je součástka již obrobena, „ignorovat“ (přeskočit). V komplikovanějších případech, kdy je např. NC program určen pro výrobní dávku a pouze jedna ze součástí vyžaduje drobnou úpravu (např. větší předhrubování čela), lze tyto bloky vepsat za funkci M30 a v programu vložit skok nebo podmínku. Po úspěšném obrobení lze tuto skokovou větu vyřadit středníkem nebo hodnotu parametru v rozhodující podmínce změnit tak, aby rozhodnutí vedlo k realizaci jen „standardního“ obrábění.

Pozn.: Více o dílensky orientovaném nebo ISO programování či obecně o pokročilých metodách NC programování se lze dozvědět v řadě specializovaných odborných kurzů a školení. Vybrané termíny jsou vypisovány na

<http://ust.fme.vutbr.cz/obrabeni/?page=kurzy> ,

ale v případě vytvoření skupiny min. 5 účastníků jsou běžně realizovány i kurzy tzv. na zakázku.

1 Nepodmíněné programové skoky

Podstatou nepodmíněných skoků je možnost měnit posloupnost čtených programových bloků. Patrně nejjednodušší použití skoku **GOTOF PRESKOC** je přeskočení několika následujících programovaných vět. Jedná se tedy o skok směrem ke konci NC programu na návěští **PRESKOC**. Skok je však možno realizovat i opačným směrem, tedy k začátku programu - **GOTOB NAVRAT**, na návěští **NAVRAT**. Po skoku vpřed nebo zpět následuje čtení v pořadí následujícího bloku (řádek o jedno níže než je cíl skoku). Zde je nutno upozornit, že v takovýchto případech nemusí funkce ukončení programu (*M30* nebo *M2*) nutně ležet na posledním řádku NC programu. Jméno cíle skoku je volitelné (např. *Preskoc*, *Navrat*), musí však být ukončeno dvojtečkou (slovo bez dvojtečky je řídicím systémem chápáno jako volání podprogramu a v případě absence podprogramu s tímto názvem je aktivováno příslušné chybové hlášení).

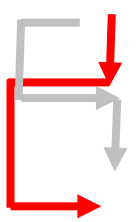
Příklad programování:

N10 G90 G54	;libovolný začátek NC programu
...	
N100 GOTOF SKOK1	;skok směrem ke konci NC ;programu
N110 SKOK2:	;cíl druhého skoku
N120 G0 X60 Z1	;rychlposuv
...	
N200 M30	;konec NC programu
N210 SKOK1 :	;cíl prvního skoku
N220 G1 X50 Z-100 F0,2	;lineární interpolace
...	
N300 GOTOB SKOK2	;skok směrem k začátku NC ;programu

3 Volba mezi dvěmi variantami

Funkci **IF** je možno úspěšně aplikovat i v případech, kdy je nutno realizovat jednu ze dvou variant. Je-li splněna podmínka (posloupnost čtených bloků naznačena červenou barvou), tak se provedou následující bloky. Není-li splněna podmínka (posloupnost čtených bloků naznačena šedou barvou), tak se provedou bloky v druhé větvi (viz krátký příklad). Podmínky typu **IF** uvedené na této straně i v předcházející kapitole mohou pracovat s výpočetními parametry **R**, kterým je možno předepsat „libovolnou“ číselnou hodnotu nebo jejich hodnota může být stanovena matematickým výpočtem parametrů jiných. Přiřazení hodnoty jednotlivých **R** parametrů do tabulky je možno realizovat přímo z NC programu nebo ručním zápisem do systémové oblasti **R-parametry**.

Příklad programování:



```
N10 G54 G90
...
N100 IF R1 == R2
...
N200 ELSE
...
N300 ENDIF
...
N400 M30
```

Proměnné s výpočetními parametry:

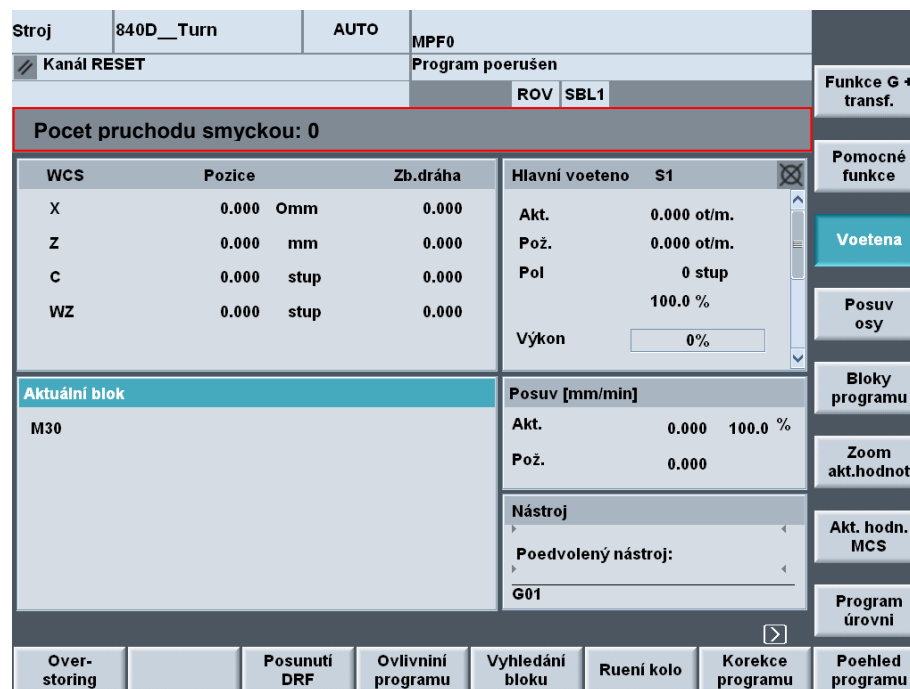
R0 = 3,1415926	R22 = COS (17,3)
R1 = -0,1234567	(cosinus 17,3°)
R2 = 12345678	R24 = SQRT (R25*R25+R26*R26)
R3 = -0,1 EX -8	(druhá odmocnina ze součtu druhých mocnin)
R4 = 12,345 EX 7	
R5 = R0 + 1	...
R8 = R6 + R7	N100 G1 X = R27 Z = R28 F0,2
R9 = R10 - R11	N110 Z = -R29
R14 = R12 * R13	...
R15 = R16 / R17	atd.
R18 = R19 + R20 * R21	

4 Nekonečná smyčka

Nekonečná smyčka se uplatňuje v případech tzv. "nekonečných" NC programů. Při dosažení konce smyčky dochází k opětovnému skoku na její začátek. Využitím možností práce s **R** parametry, v kombinaci s možnostmi textových poznámek, lze realizovat např. sledování počtu opakování smyčky. Matematickým zápisem $R0 = 0$ je do parametru zapsána konkrétní číselná hodnota. Matematickým zápisem $R0 = R0 + 1$ je pak k aktuální hodnotě (libovolné hodnotě) parametru přičtena libovolná číselná hodnota (zde celé číslo 1). Příkaz pro vypsání textové poznámky v poli chybových hlášení (obr. 1) je **MSG**. Textová poznámka typu **MSG** je na obrazovce stroje viditelná do opětovného volání další textové poznámky tohoto typu nebo do okamžiku aktivace chybového hlášení. **MSG** umožňuje vypsát libovolný text vložený do uvozovek a pro volání a vypsání aktuální hodnoty je znakem << připojen příslušný parametr.

Příklad programování:

```
N10 G54 G90 ;volitelné bloky
R0 = 0
...
N100 LOOP ;začátek smyčky
MSG("Počet průchodů smyčkou: " << R0)
... ;volitelné programové bloky
R0 = R0 + 1
N200 ENDLOOP ;konec smyčky
...
N210 M30 ;volitelné ukončení programu
;nebo podprogramu
```



Obr. 1 Pracovní oblast - STROJ

5 Čítací smyčka

Smyčka FOR je zpravidla použita, má-li být pracovní činnost opakována s pevným počtem průběhů. Čítací proměnná typu **INT** přitom počítá směrem nahoru od počáteční (nižší) hodnoty po koncovou (vyšší) hodnotu. U libovolně zvoleného slova pro čítací proměnnou (parametry **R** nejsou vhodné) je na začátku NC programu nutno provést jeho definici na typ **INT** (celočíselná hodnota). S ohledem na nutnost definovat proměnnou není vhodné tento zápis realizovat v prostředí JobShop produktů (ShopTurn a ShopMill). Při dílensky orientovaném programování je vhodnější programovat např. výpočet nebo obecně smyčku v textovém editoru, do kterého je možno z hlavního programu vstoupit voláním souboru podprogramu.

Podprogram je z hlavního programu (soubor s koncovkou *.mpf) volán zvoleným slovem (blok se skládá ze slova, které není ukončeno dvojtečkou), které odpovídá názvu podprogramu (soubor s koncovkou *.spf). Podprogram je ukončen slovem M17, kterým je realizován skok zpět do hlavního programu.

Příklad programování:

Vytvořte krátký NC program, který bude do pole textových poznámek vypisovat aktuální hodnotu proměnné. Smyčka musí proběhnout přesně 15krát.

DEF INT prom	;definice proměnné jako typ INT – celé číslo
FOR prom = 1 TO 15	;zápis čítací smyčky s podmínkou <i>prom</i> a rozsahem počítání („opakování“) od 1 do 15
...	;volitelné programové bloky
MSG ("Aktuální hodnota proměnné s názvem <i>prom</i> je: " << prom)	;textová poznámka
ENDFOR	;ukončení podmínky
M30	;v hlavním programu nebo M17 v podprogramu




6 Smyčka s podmínkou na začátku

Smyčka proběhne, pokud je splněna podmínka. POZOR - smyčka probíhá stále, dokud je podmínka splněna. Není-li např. v průběhu realizace smyčky provedena změna parametru pro porovnávací operátor, může dojít k nežádoucímu zacyklení.

Příklad programování:

N10 G90 G54	;volitelné programové bloky
R1=1	;přiřazení hodnoty proměnné v NC programu (hodnotu je možno vepsat i do tabulky R parametrů)
...	;volitelné programové bloky
N100 WHILE R1<=10	;programová smyčka s volitelnou podmínkou (zde R1<=10)
...	;volitelné programové bloky
R1=R1+10	;změna hodnoty proměnné
N200 ENDWHILE	;ukončení programové smyčky
...	;volitelné programové bloky
N400 M30	;volitelné ukončení NC programu nebo podprogramu



7 Funkce CASE

Funkce **CASE** umožňuje v závislosti na hodnotě typu INT provádět bloky v různých programových větvích. Podle toho, jaká hodnota je výsledkem výrazu, provede funkce skok na místo s příslušným cílem (např. hodnota výrazu $R1+R2$ je 20, bude realizována pouze větev se zvoleným slovem cíle skoku **label1**). Hledání cílového místa může být směrem ke konci NC programu (**gotof**) nebo k začátku NC programu (**gotob**). To, zda jsou klíčová slova a funkce psána malými nebo velkými písmeny, není rozhodující. V případě, že výsledku výrazu neodpovídá žádná z předem definovaných programových větví, může být pokynem **default** realizována konkrétní věta s programovými bloky.

Příklad programování:

R1 = R1 + R0 ;libovolné stanovení hodnoty parametru

R2 = 20 ;přiřazení libovolné hodnoty parametru

N100 **CASE** (R1+R2) of 20 **gotof** label1 30 **gotof** label2 **default gotof** label3

;20 a 30 jsou možné hodnoty výrazu
;(konstanty) a label je volitelný cíl skoku



M30 ;volitelné ukončení NC programu nebo podprogramu

8 Závěr

Funkce prezentované v dnešním informativním příkladu ani zdaleka nejsou kompletním výčtem možností programových skoků a smyček. Jak naznačuje např. kapitola č. 7 *Smyčka s podmínkou na začátku*, můžeme ve volně navazujícím příkladu očekávat smyčku s podmínkou na konci. Pozornost však bude věnována i dalším „užitečným“ funkcím ISO jazyka, které je však možno úspěšně aplikovat i při dílensky orientovaném programování v ShopTurnu a ShopMillu. Jen namátkou lze již dnes upozornit na běžné užívání funkce G96 (konstantní řezná rychlost). Mezi možnostmi NC programování však můžeme zařadit i programování nekonstantní hodnoty posuvu, ale také funkce pro zápis, čtení a mazání zvoleného textového souboru nebo adaptivní řízení (např. automatická změna hodnoty opotřebení v závislosti na počtu obroběných kusů, atd.).

**Článek vznikl za spolupráce Vysokého učení technického v Brně,
FSI, ÚST, Odboru technologie obrábění
s firmou Siemens a redakcí Technického týdeníku.**

Přehled souvisejících odkazů:

<http://ust.fme.vutbr.cz/obrabeni>

<http://cadcam.fme.vutbr.cz>

<http://cnc.fme.vutbr.cz>

<http://esf.fme.vutbr.cz>

SIEMENS

<http://www.siemens.cz>

<http://www.ad.siemens.de/doconweb>

<http://www.techtydenik.cz>

Kontaktní osoba:

Vysoké učení technické v Brně
Odbor technologie obrábění
Technická 2896/2
616 69 Brno

Ing. Aleš Polzer, Ph.D.

Tel./fax.: 5 4114 2545

E-mail: polzer@fme.vutbr.cz

[Předchozí](#)