

Řídicí systém CNC872

Doplněk k Návodu pro obsluhu a
programování
pro soustruhy a karusely



1

1. Stavba partprogramu

1.1 Bloky, číslování bloků, začátek a konec partprogramu

Jednotlivé adresy, M-funkce a G-funkce, případně další prvky (např. MAKRA) se v partprogramu sestavují do programových bloků. Programový blok tvoří základní informační jednotku nesoucí údaj o geometrii, technologii obrábění, může obsahovat pomocné výpočty, rozhodovací podmínky apod. Jeden blok partprogramu může být rozepsán na libovolný počet řádků - délka řádku není omezena, nicméně je vhodné psát maximálně 50 znaků na jednu řádku, aby při chodu partprogramu byl vidět celý řádek, bez nutnosti v něm horizontálně rolovat.

Každý programový blok musí začínat adresou N - číslo bloku. Blok končí před následujícím znakem N (nikoli tedy na konci řádku!) nebo koncem souboru. Číslo bloku, uvedené za adresou N, je nepovinné. Číslo bloku slouží pro orientaci v partprogramu, pro volbu bloku a identifikaci chyb, proto se doporučuje je psát. Nezbytné je psát číslo bloku, pokud se na něj provádí skok. Číslo bloku může být i nula, což je to samé jako N bez čísla. V partprogramu se mohou vyskytovat jak číslované bloky, tak bloky N0 nebo N bez čísla. Pokud je blok číslovaný, musí být číslo bloku jedinečné, tj. v partprogramu se nesmí opakovat stejná čísla bloků. Čísla bloků ale nemusí nutně být řazený vzestupně, tj. správný je např. i tento zápis:

```
N PROGRAM
N100 ...
N ...
N20 ...
N0 ...
N0 ...
N5 ...
```

Partprogram začíná klíčovým slovem (makrem) **PROGRAM**. (nepovinně)

Partprogram končí klíčovým slovem (makrem) **ENDPROGRAM** (povinně)

Místo klíčového slova (makra) **ENDPROGRAM** je možné použít přímo funkci M30.

Pozn: Makro je rozvoj textu, uvedeného v definici makra. Např. v definici makra ENDPROGRAM je rozvoj textu M30

Při „ručním“ zápisu partprogramu se doporučuje psát bloky vzestupně s krokem např. po 10, aby se při dodatečném zápisu mohl vložit blok s číslem odpovídajícím vzestupnému číslování (např. 10,20,**25**,30 atd) i když to není nutné ani povinné.

Doporučený formát partprogramu

```
N PROGRAM
N10 ...
N20 ...
N30 ...
.....
N ENDPROGRAM
```

Interpretace takto zapsaného partprogramu:

```
N10 ...
N20 ...
N30 ...
.....
N M30
```

Pozn.: Makro ENDPGRAM neznamena konec souboru, za tímto klíčovým slovem mohou následovat ještě podprogramy.

1.2 Komentáře v partprogramu

V partprogramu se mohou vyskytovat komentáře. KOMENTÁŘ je řetězec libovolných znaků (kromě uvozovek) uzavřený mezi uvozovkami. Pokud je za komentářem uveden znak konce řádku (CR,LF), nemusí být koncové uvozovky uvedeny.

Příklad:

```
N10 X100 Z100      " Toto je komentář"
N20 X200 Z200      " Toto je komentář
" "
"
N30 X100 " Toto je komentář " Z300
N40 X100 " Toto je komentář  Z300
```

Pozn.:

V bloku N40 se do komentáře zahrne i Z300, neboť nejsou uvedeny druhé uvozovky, takže se za konec komentáře považuje konec řádky.

1.3 Poznámky k bloku partprogramu

Systém pracuje s proměnnou délkou bloku. To znamená, že v každém bloku může být libovolný počet slov. Slovem se rozumí adresa a její hodnota (seznam adres viz dále). Každá adresa může být zapsána v jednom bloku pouze jednou, s výjimkou skupinových funkcí (M,G), u kterých může být zapsána jedna hodnota z každé skupiny. Rovněž číslované parametry mohou být v bloku použity vícekrát.

Místo hodnoty adresy je možné u všech adres (kromě adresy N) zapisovat číslo parametru - viz kapitola o parametrickém programování.

Jako oddělovače mezi jednotlivými slovy bloku lze užít libovolný počet mezer, případně tabulátorů. Mezery se mohou navíc užít i uvnitř slova, avšak pouze mezi adresou a hodnotou (nikdy ne uvnitř číselné hodnoty!).

Znaménko je nutné uvádět pouze u záporné hodnoty, kladné znaménko se nemusí uvádět.

Příklad:

```
N10 X-100          „ správně
N10 X -100         „ správně, ale méně vhodné
N10 X- 100         „ špatně (mezera v hodnotě)
N10 X -1 0 0       „ špatně (mezera v hodnotě)
```

Zápis bloku v jednom řádku:

```
N10 G0 X100 Z20 M3 S100 R1=30 R2=-15
N20 ...           „ následující blok
```

Rozepsání bloku do více řádků:

```
N10 G0 X100 Z20 „ dráha rychloposuvem
      M3 S100    „ otáčky
      R1=30      „ parametr 1
      R2=-15     „ parametr 2
N20 ...         „ následující blok
```

Rozepsání bloku do více řádků se často využívá např. pokud chceme do partprogramu zapsat podrobnější komentáře

1.4 Podprogramy, makrocykly a skoky

Složitější partprogram součástí se může skládat nejen z vlastního partprogramu, ale může obsahovat i volání podprogramů, makrocyklů a tzv. pevných cyklů. Podprogramy se vztahují pouze k danému partprogramu, za jehož koncem musí být bezprostředně zapsány. Makrocykly a pevné cykly se nacházejí v samostatných souborech a jsou z těla partprogramu event. podprogramu vyvolány příkazem.

1.4.1 Podprogramy

Podprogramem se rozumí určitá skupina programových bloků, které mají standardní úvodní a závěrečný blok. Podprogram logicky patří pouze k danému partprogramu a může být volán pouze z tohoto partprogramu – je zapsán bezprostředně za logickým koncem partprogramu (za klíčovým slovem ENDPROGRAM, resp. za funkcí M30).

Pro vyvolání podprogramu se používá funkce **CALL(SubNo)**, kde SubNo je číslo podprogramu.

Příklad volání podprogramů funkcí CALL():

```
$VYMENA      20
N PROGRAM      " začátek partprogramu
N ...
N100 CALL(10)   " vyvolání podprogramu číslo 10
N110 CALL(VYMENA) " vyvolání podprogramu číslo 20 (rozvoj makra VYMENA)
N ENDPGRAM
"
N200 BEGIN(10)  " Začátek podprogramu číslo 10
N210 ...
N220 END        " Konec podprogramu číslo 10

N300 BEGIN(VYMENA) " Začátek podprogramu číslo 20
N310 ...
N320 END        " Konec podprogramu číslo 20
```

Druhý podprogram má místo čísla uvedeno VYMENA, což ale není nic jiného než rozvoj makra, které je nadefinované před začátkem partprogramu a jeho rozvojem je číslo 20. Tento způsob lze používat např. pro snadnější zapamatování názvů podprogramů.

V programovém bloku partprogramu, ze kterého je volán podprogram, je přípustné programovat i ostatní posuvové a technologické funkce. Vlastní odskok na zvolený podprogram se provede až v závěru bloku, t.j. po vykonání ostatních programovaných operací. Po provedení podprogramu se řízení vrátí na následující blok partprogramu, ze kterého byl odskok proveden.

1.4.2 Makrocykly

Makrocyklem se rozumí určitá skupina programových bloků, které tvoří partprogram pro typickou součást nebo část součásti vyráběnou na konkrétním stroji. Makrocykly jsou vztaheny ke všem uloženým partprogramům, ze kterých může být volán (může být volán z vlastních partprogramů i jejich podprogramů).

Časté je použití makrocyklů pro činnosti, které jsou společné všem partprogramům, např. nájezd do polohy pro výměnu nástroje a výměna nástroje apod. Je vhodné využít i parametrického programování.

Volání makrocyklu je možno provést z kteréhokoliv bloku libovolného partprogramu. Volání má podobný tvar jako u podprogramů, místo funkce Call(SubNo) se použije **CallMacro(MacNo)**, kde MacNo je číslo

makrocyklu. Protože makrocykly nejsou součástí partprogramu, musí být zřejmé, kde se nacházejí. Na začátku partprogramu musí být uvedena jejich deklarace, která znamená příkaz pro načtení deklarovaného souboru do paměti. Makrocykly se standardně ukládají do adresáře C:\CNC User Files\MAC.

Doporučený způsob používání makrocyklů je následující. Uživatel si vytvoří soubor, nazvaný například MACLATHE.NCP (makrocykly pro soustruh) a do něj postupně zapisuje makrocykly, tak jak je během používání stroje vytváří.

Deklarace makrocyklů (tj. načtení do paměti) v partprogramu je následující :

```
#MAC (MACLATHE.NCP)      "Soubor MACLATHE.NCP je umístěn v adresáři MAC
```

Doporučené volání makrocyklů z partprogramu uvádí následující příklad.

```
#MAC (MACLATHE.NCP)
N PROGRAM      " začátek partprogramu
N ...
N ...
N100 CallMacro(1) " vyvolání makrocyklu číslo 1
N105
N110 CallMacro(2) " vyvolání makrocyklu číslo 2
N ...
N ENDPROGRAM   " konec partprogramu
```

Příklad souboru MACLATHE.NCP s makrocykly:

```
N BEGIN(1)      " začátek makrocyklu číslo 1
N ...           " tělo makrocyklu např. pro výměnu nástroje
N ...
N END           " konec makrocyklu číslo 1

N BEGIN(2)      " začátek makrocyklu číslo 2
N ...           " tělo makrocyklu např. otáčení stolu
N ...
N END           " konec makrocyklu číslo 2

N BEGIN(3)      " začátek makrocyklu číslo 3
N ...           " tělo makrocyklu např. pro proměřování obrobku
N ...
N END           " konec makrocyklu číslo 3
```

1.4.3 Skoky v partprogramu

Skoky jsou povoleny pouze na programové bloky v rámci programového celku, t.j. v rámci partprogramu, nebo podprogramu či makrocyklu ve směru vzad i vpřed (tj. ve směru k začátku nebo konci partprogramu, podprogramu nebo makrocyklu). Nelze tedy použít např. skoku z partprogramu do podprogramu.

Pro zápis skoku se použije funkce **JMP(BlockNo)**, kde BlockNo je číslo bloku, na který se provede skok. Pozor, skok směrem „dozadu“, tj. k začátku partprogramu bez další podmínky, která určí, kdy cyklus opustit, nemá praktický význam, protože by došlo by k „nekonečnému“ zacyklení partprogramu (vyhlásí se chyba „Program obsahuje příliš mnoho bloků“).

Skok směrem „dopředu“, ke konci partprogramu, se použije pokud chceme např. ve fázi odladování partprogramu přeskočit jeho určitý úsek. Většinou se ale funkce skoku používá ve spojení s rozhodovacími podmínkami – viz kapitola o funkcích.

Často je potřeba programovat skok zpět několikrát po sobě (opakování určité části partprogramu vícekrát se změněnými parametry). Můžeme použít systémové předdefinované makro, které provede n-krát skok zpět na

zvolený blok, resp. provede n-krát zvolený úsek partprogramu (první průchod a skok zpět (n-1)krát, t.j. celkem n průchodů). Zápis je mnohem jednodušší. Uvedeme rozvoje tohoto makra, které má tři parametry:

```
$Loop(BlokNo, Count, Counter)      \  
    Counter = Counter + 1           \  
    If(Less(Counter,Count))         \  
        Jmp(BlokNo)                 \  
    Else                             \  
        Counter = 0                 \  
    EndIf
```

Jednotlivé parametry mají tento význam:

BlockNo: Číslo bloku, kterým smyčka začíná (musí být před blokem, kde je programováno Loop)

Count: Počet průchodů

Counter: Čítač počtu průchodů (celočíslný parametr)

Příklad:

Úsek partprogramu mezi bloky N110 a N130 se provede 5x

```
N PROGRAM  
...  
N90 I1=0                „ Případná inicializace čítače průchodů  
...  
N100 X0 G0 G90          „ Najede absolutně na X0  
N110 G91 X1             „ Jede přírůstkově v ose X o 1mm  
N120 ...  
N130 Loop(110,5,I1)     „ Skok na blok N110 se 4x zopakuje  
N140 ...
```

Druhý parametr udává celkový počet průchodů (5), nikoli počet skoků (ten je o jednu menší)! Úsek partprogramu mezi bloky N110 a N130 se tedy provede 5x, v bloku N140 bude míra X5. Pomocný čítač počtu průchodů je v příkladu v parametru I1. Pokud se použitý parametr v partprogramu nepoužije, nemusí se inicializovat (t.j. nulovat). Pokud by se použil v části partprogramu před použitím makra Loop, musí se předem inicializovat, t.j. vynulovat (blok N90). Inicializaci můžeme pro jistotu provést vždy.

Makra Loop() můžeme do sebe vnořovat, pro každé makro ale musíme použít jiný parametr pro čítač průchodů, zde použít I1 a I2. Koncová míra souřadnic v příkladu bude X5 a Y25

```
N PROGRAM  
N10 G90 X0 Y0 G00  
N20 X1 G91 G00          „ přírůstkově X1  
N30 Y1                  „ přírůstkově Y1  
N40 Loop(30,5,I1)       „ skok na blok N30  
N50  
N60 Loop(20,5,I2)       „ skok na blok N20  
N70  
N ENDPROGRAM
```

1.4.4 MAKRA

Makrem se rozumí náhrada **názvu makra** textem, který je uveden v **definici makra**. Makra si může uživatel vytvářet sám nebo využívat předdefinovaná (systémová) makra, umístěná obvykle v hlavičkových souborech (t.j. v adresářích C:\Program Files\MEFI\WinCNC\Include nebo C:\Machine Fines\Include, soubory s příponou *.NCH).

Uživatel si svoje makra rovněž může uložit do hlavičkových souborů (v adresáři D:\CNC User Files\Include) v případě, že je chce používat v různých partprogramech, nebo může makro zapsat přímo do partprogramu (obvykle se MAKRA píšou na začátek), pokud se jedná o jednorázové použití.

Systémová makra jsou uvedena v hlavičkovém souboru StdHdr.nch, který je umístěn v adresáři:

C:\Program Files\MEFI\WinCNC\Include

V něm jsou uvedena např. tato makra (konstanty)

\$IMPERIAL	1	" Palcové zadávání - inch, inch/min, ...
\$METRIC	0	" Metrické zadávání – mm, mm/min
\$PI	3.1415926535897932384626433832795	" Ludolfovo číslo

V následujícím příkladu partprogramu jsou uvedena makra použita, kromě nich jsou vytvořena ještě dvě makra uživatele:

\$Obvod RPARAM

\$Polomer RPARAM

```

N PROGRAM
N X0 G00 G90
N LENGTHUNIT = IMPERIAL      " Programování délek v palcích
N X1 G0                       " Pojede na míru 25,4mm (= 1 palec)
N X0
N LENGTHUNIT = METRIC        " Programování délek v mm
N X1 G0                       " Pojede na míru 1 mm
N X0
N Polomer = 10
N Obvod = 2 * PI * Polomer
N MsgShow(1, 'Obvod kruhu o poloměru \r = \r', Polomer, Obvod)
N ENDPROGRAM

```

Textový řetězec IMPERIAL je v preprocesoru nahrazen 1, textový řetězec METRIC je nahrazen 0. Stejný výsledek bychom tedy dostali, pokud bychom použili tato čísla. Pojmenované konstanty jsou ovšem lépe zapamatovatelné. Makra Obvod a Polomer jsou nahrazena automaticky vygenerovanými reálnými parametry.

Makro může být rozepsáno i do více řádků. V tomto případě se na konci každého řádku, kromě posledního, uvede znak zpětného lomítka (\).

Jako příklad uvádíme naprogramování makra - přímka daná úhlem a jedním koncovým bodem X. Ze systémové proměnné **DIAMETERPROGR** se zjistí, zda se programuje poloměrově nebo průměrově

```

$Angle      RPARAM
$PrX        RPARAM
$Prum       RPARAM
$LineAX(Angle, PrX) if (DIAMETERPROGR) \
                    Prum=2 \
                    else \
                    Prum=1 \
                    endif \
                    if (OR (EQ (Angle, 90), EQ (Angle, 270))) \
                    Err('Nepovolený úhel u makra LineAX()') \
                    else \
                    AxGZ=AxGZ+ ((PrX/ (Prum)) -AxGX) *Tan ((Angle*PI)/180) \
                    AxGX=PrX/ (Prum) \
                    Endif

```

Použití v partprogramu

```

N100 G0 X100
N110 G1 LineAX(25, 155) „ úhel 25 stupňů, koncová míra v X je 155
N120 ...

```



2. Seznam programovatelných adres

V následujících tabulkách je uveden výběr adres a M a G funkcí, které se používají u soustruhů a karuselů. Úplný výčet adres a funkcí je k dispozici v „Návodu k programování“

Poslední sloupec v tabulce udává u některých adres ekvivalentní zápis pomocí systémových parametrů nebo pomocí funkcí.

2.1 Programovatelné adresy

Adresa	Typ	Popis	Ekvivalent jako systémový parametr
D	INT	Volba položky z tabulky korekcí	Za určitých předpokladů se může použít také ToolLenX, ToolLenY, ToolLenZ, ToolRadius (viz příklady dále v návodu)
F	REAL	- Rychlost posuvu - Časová prodleva pro G04	- Feed - DelayTime
G	INT	Přípravné funkce (viz seznam G-funkcí)	ProgrG(GNo)
I	REAL	Konfigurovatelná adresa, nejčastější použití jako interpolační parametr: Vzdálenost středu kružnice ve směru geometrické osy X	CCX
K	REAL	Konfigurovatelná adresa, nejčastější použití jako interpolační parametr: Vzdálenost středu kružnice ve směru geometrické osy Z	CCZ
L	INT	- Číslo volaného podprogramu pro G71. - Číslo volaného makrociklu pro G72 - Číslo zařazovaného pevného cyklu pro G76. - Číslo cílového bloku skoku pro G73 - Číslo podprogramu, makrociklu či pevného cyklu pro G79.	
M	INT	Pomocná funkce	ProgrM(MNo)
N	INT	Číslo bloku (číselná hodnota za znakem N nemusí být použita)	
R	REAL	Poloměr kruhu pro G2 a G3.	CR
S	REAL	Otáčky vřetena	SPINDLESPEED
T	INT	Volba nástroje	Obvykle se používá parametrizované PLC makro Tool(). Viz dále v návodu
X	REAL	Osa X	

Z	REAL	Osa Z	
---	------	-------	--

2.2 G a M funkce

Přípravné G-funkce a pomocné M-funkce jsou dále rozděleny do skupin. Z hlediska programátora není číslo skupiny důležité, platí pouze, že přípravné G-funkce z jedné skupiny nesmí být (při klasickém programování podle ISO) programovány v témže bloku. Totéž platí pro pomocné M-funkce. Přehled používaných G a M funkcí pro soustruhy a karusely je uveden v následujících tabulkách.

Pozn:

V praxi se u konkrétního stroje může vyskytnout M-funkce, neuvedená v tomto návodu, případně uvedená, ale s modifikovaným významem. V tom případě se jedná o M-funkci, kterou návrhář PLC použije u konkrétního stroje. Seznam použitých M-funkcí by měl být k dispozici od tvůrce PLC.

Seznam G-funkcí

Skupina	Funkce	Popis	Ekvivalent jako systémový parametr nebo parametry, které se ke G-funkci mohou vázat
0	0	Najíždění rychloposuvem	
	1	Lineární interpolace	
	2	Kruhová interpolace ve směru hodinových ručiček (CW)	CR, CREV
	3	Kruhová interpolace proti směru hodinových ručiček (CCW)	CR, CREV
	10	Zkosení rohu se zadanou délkou zkosení CHLEN, nebo se zadanou vzdáleností začátku/konce zkosení od rohu CHLENT.	CHLEN, CHLENT
	11	Zaoblení rohu se zadaným poloměrem ROUND R	ROUND R
	12	Kruhová interpolace, tečné napojení na předchozí blok	
	13	Kruhová interpolace, tečné napojení na následující blok	
	33	Řezání závitů	
1	18	Volba pracovní roviny Z - X	V programech ji není nutné uvádět
2	-	-	U soustruhů a karuselů se nepoužívá
3	40	Bez poloměrové korekce (zrušení poloměrové korekce)	
	41	Poloměrová korekce vlevo	ToolRadius, RAngle, RCMMethod, RCChangeMethod, RCOptions, TNP
	42	Poloměrová korekce vpravo	ToolRadius, RAngle, RCMMethod, RCChangeMethod, RCOptions, TNP
4	23	Plynulá návaznost bloků (mezi bloky se nezastavuje – obálková rychlost)	Funkce je obvykle zadána v prioritním bloku a nemusí se programovat
	24	Napojení bloků bez plynulé návaznosti (mezi bloky klesne rychlost na 0)	
5	50	Posunutí nulového bodu, číslo řádku v tabulce udává adresu L	Translace(), ATranslate(), WTranslate(), WATranslate()
	53	Posunutí 0 nulového bodu	Dtto
	54	Posunutí 1 nulového bodu	Dtto
	55	Posunutí 2 nulového bodu	Dtto
	56	Posunutí 3 nulového bodu	Dtto
	57	Posunutí 4 nulového bodu	Dtto
	58	Posunutí 5 nulového bodu	Dtto

	59	Posunutí 6 nulového bodu	Dtto
6	94	Posuv v mm/min bez konstantní řezné rychlosti	
	95	Posuv v mm/ot bez konstantní řezné rychlosti	
	96	Posuv v mm/ot s konstantní řeznou rychlostí	SLIM,CCS
	97	Posuv v mm/min s konstantní řeznou rychlostí	SLIM,CCS
7	70	Konec podprogramu, makrocycly nebo pevného cyklu	END
	71	Volání podprogramu, číslo podprogramu udává adresa L	Doporučený způsob: Call(SubNo)
	72	Volání makrocycly, číslo makrocycly udává adresa L	Doporučený způsob: CallMacro(MacNo)
	73	Skok na programový blok, jehož číslo udává adresa L	JMP(BlockNo) nebo Loop(BlockNo,Count,I)
	79	Začátek podprogramu, makrocycly či pevného cyklu, číslo udává adresa L	BEGIN
8	76	Zařazení volání pevného cyklu na konci každého bloku, číslo pevného cyklu udává adresa L	
	80	Zrušení opakovaného volání pevného cyklu na konci každého bloku	
	81	Zařazení volání pevného cyklu č. 81 na konci každého bloku	
	82	Zařazení volání pevného cyklu č. 82 na konci každého bloku	
	83	Zařazení volání pevného cyklu č. 83 na konci každého bloku	
	84	Zařazení volání pevného cyklu č. 84 na konci každého bloku	
	85	Zařazení volání pevného cyklu č. 85 na konci každého bloku	
	86	Zařazení volání pevného cyklu č. 86 na konci každého bloku	
	87	Zařazení volání pevného cyklu č. 87 na konci každého bloku	
	88	Zařazení volání pevného cyklu č. 88 na konci každého bloku	
	89	Zařazení volání pevného cyklu č. 89 na konci každého bloku	
9	90	Režim absolutního programování souřadnic	
	91	Režim přírůstkového programování souřadnic	
10	4	Časová prodleva, velikost v sekundách je zadána funkcí F	Doporučený způsob: Delay(Time)
11		REZERVA	

Seznam M-funkcí

M-funkce jsou ve většině případů plně v kompetenci PLC programu pro konkrétní stroj. V seznamu jsou M-funkce podle doporučení ISO. Seznam platných M-funkcí pro konkrétní stroj je součástí dokumentace ke stroji. V jednom bloku nelze psát M-funkce z jedné skupiny. Pokud se ale M-funkce programují pomocí systémových parametrů, toto pravidlo neplatí – používá se to např. v rozhodovacích podmínkách. Podrobněji v dalších částech návodu. Ekvivalenty systémových parametrů v tabulce nejsou, pro příklad uveďme pouze standardní makro pro konec programu: ENDPROGRAM se nahradí funkcí M30.

Skupina	Popis	
1	00	Programový stop
	01	Volitelný stop (viz možnosti jízdy)
	02	Konec partprogramu
	30	Konec partprogramu
2	03	Start vřetena "CW"
	04	Start vřetena "CCW"
	05	Stop vřetena
	19	Stop vřetena v orientovaném bodě
3	40	Rozsah otáček vřetena je vypočten přímo z funkce S
	41	Otáčky vřetena rozsah 1
	42	Otáčky vřetena rozsah 2
	43	Otáčky vřetena rozsah 3
	44	Otáčky vřetena rozsah 4
4		REZERVA
5	7	Zapnutí chlazení 2
	8	Zapnutí chlazení 1
	9	Vypnutí chlazení 1 a 2
	17	Zapnutí chlazení 1 a 2
6	50	Zapnutí chlazení 3
	51	Zapnutí chlazení 4
	52	Vypnutí chlazení 3 a 4
	53	Zapnutí chlazení 3 a 4
7	10	Upnutí obrobku
	11	Uvolnění obrobku
8	49	Překlenutí ručního FEED OVERRIDE
	48	Zrušení překlenutí (zařazení) FEED OVERRIDE
9	06	Výměna nástroje
	60	Výměna obrobku
10	06	Volitelné M-funkce
11	06	Volitelné M-funkce
12	06	Volitelné M-funkce
13	06	Volitelné M-funkce
14	06	Volitelné M-funkce

3

3. Programování

Partprogramy je možno psát v absolutních souřadnicích nebo přírůstkově.

- G90 – absolutní programování
- G91 – přírůstkové programování

Při absolutním programování se koncové míry programují v absolutních hodnotách vzhledem k použitému počátku (obvykle funkce G53 – G59). Při přírůstkovém programování se programuje pouze přírůstek (tj. o kolik mm se má programovaná osa přesunout).

Většina partprogramu se píše absolutně (vzhledem k zvolenému počátku). Absolutní programování umožňuje mimo jiné snadnější úpravy partprogramu. Při vložení bloku do partprogramu, který je napsán přírůstkově, je třeba brát v úvahu, že se ovlivní míry všech následujících bloků!

I v případě přírůstkového programování je potřeba alespoň začátek partprogramu (tj. polohu všech os) naprogramovat absolutně. Teprve v dalších blocích je možné přejít na přírůstkové programování. Absolutní a přírůstkové programování se může v partprogramu libovolně přepínat. V prioritním bloku je obvykle naprogramována funkce G90 – absolutní programování, tj. pokud se programuje výhradně absolutně, nemusí se tato funkce uvádět.

3.1 Stavění souřadnic - funkce G0

Stavěním souřadnic se rozumí přemístění nástroje do koncového (programovaného) bodu rychloposuvem. Koncová poloha se programuje v absolutních nebo inkrementálních mírách (platí obecně pro všechny druhy pohybu). Toto přemístění nástroje je zadáno v bloku funkcí **G0 (možno psát též G00)**, která je současně nositelem informace pro provádění pohybu rychloposuvem. Velikost rychloposuvu (pro každou osu) je v systému zadána pevně jako strojní konstanta (atribut FeedRT v souboru Channel0.ChannelConfig) a v bloku se neprogramuje. Při rychloposuvu je zaručen plynulý rozjezd a dojezd na začátku a konci pohybu. V jednom bloku je možno programovat stavění i více souřadnic najednou.

Příklad

N10 G0 X100 Z50

„ Programování rychloposuvu

3.2 Lineární interpolace - funkce G1

Lineární interpolace se volí funkcí **G1 (možno psát též G01)**. V jednom bloku je možno naprogramovat interpolaci mezi jednou až n-souřadnicemi (n je maximální počet souřadnic podle konfigurace) naprogramováním souřadnic koncového bodu v příslušných osách. Je tedy možná i vzájemná kombinace lineárních a rotačních souřadnic. Pro lineární interpolaci je nutné, aby byla zadána posuvová rychlost (pod adresou **F** nebo parametrem **Feed**). Rychlost F nemusí být uvedena v bloku s G1, ale v kterémkoli předcházejícím bloku. Rychlost je vypočtena pro rovinu X-Z.

Příklad

Absolutní programování

N10 G1 G90 X90 Y50 F100

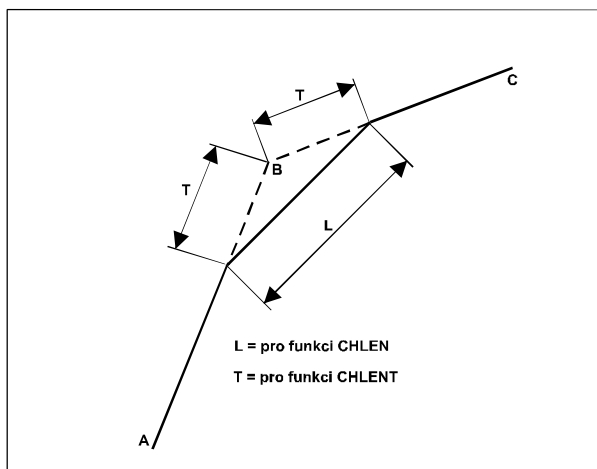
přírůstkové programování

N10 G01 G91 X40 Y30 F100

Je-li výchozím bodem interpolace bod A o souřadnicích X=50, Y=20 a koncový bod B o souřadnicích X=90 Y=50, potom uvedené bloky vykonají stejnou dráhu.

3.2.1 Zkosení rohu – G10

Pomocí funkcí G10 lze programovat zkosení rohu se zadanou délkou zkosení (funkce CHLEN) nebo se zadanou vzdáleností začátku/konce zkosení od rohu (funkce CHLENT). Použití vyplývá z uvedených příkladů:



Pokud se programují se dvě navazující úsečky A-B a B-C (viz obr.), je možné mezi ně vložit zkosení zadané buď jeho délkou L nebo vzdáleností začátku zkosení k průsečíku B.

Zadání zkosení délkou L:

```
N10 G90 G0 X0 Z0
N20 G91 G1 F100 X50 Z150
N30 G10 CHLEN=30 " Délka L
N40 G1 X150 Z50
```

Zadání zkosení vzdáleností T:

```
N10 G90 G0 X0 Z0
N20 G91 G1 F100 X50 Z150
N30 G10 CHLENT=20 " Délka T
N40 G1 X150 Z50
```

3.3 Kruhová interpolace - funkce G2, G3

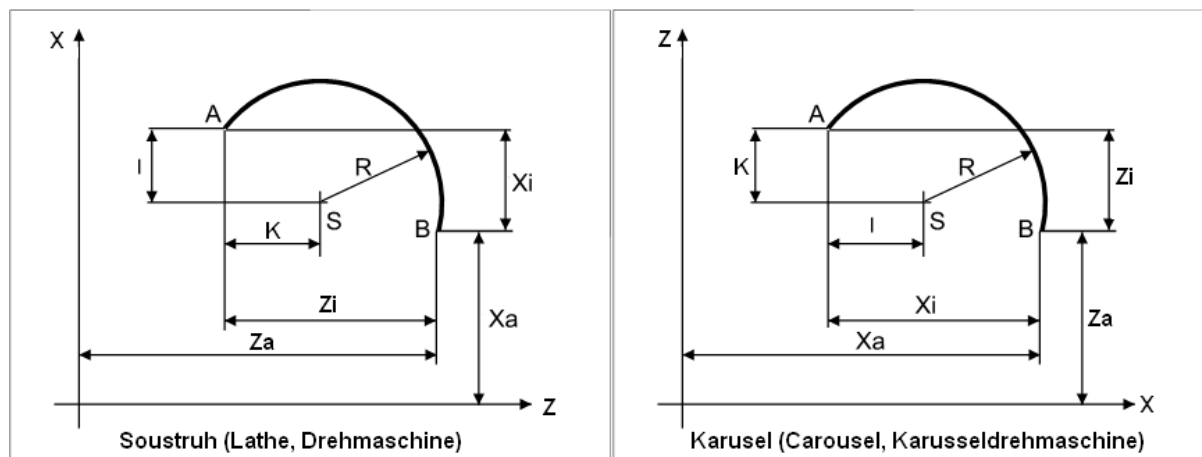
Kruhová interpolace se volí funkcí **G2** (CW - pohyb ve směru hodinových ručiček, možno psát též G02) nebo **G3** (CCW - pohyb proti směru hodinových ručiček, možno psát též G03). Vyjádření směru kruhové interpolace (G2 nebo G3) v rovině X - Z se určuje při pohledu na rovinu kruhové dráhy (viz obr. v kapitole GRAFIKA).

Kruhová interpolace se zadává **koncovým bodem** kružnice (X, Z) a buď **souřadnicemi středu** nebo **poloměrem** kružnice. Pokud se jedna ze souřadnic oproti předešlému bloku nemění, nemusí se uvádět, nicméně doporučuje se u kruhové interpolace vždy uvádět obě souřadnice koncového bodu.

3.3.1 Možnosti zadání kruhové interpolace

Kružnici je možné zadat několika způsoby. Zadáním koncového bodu a středu, zadáním koncového bodu a poloměru. V dalším textu budou uvedeny všechny možnosti.

Na obrázku je uvedeno zadání pro soustruhy a karusely. Prakticky se liší pouze orientací souřadného systému.



- A** - počáteční bod kružnice
- B** - koncový bod kružnice
- S** - střed kružnice
- I** - vzdálenost počátku a středu kružnice ve směru X
- K** - vzdálenost počátku a středu kružnice ve směru Z
- Xa** - X-ová souřadnice koncového bodu kružnice v absolutním programování
- Xi** - X-ová souřadnice koncového bodu kružnice v inkrement. programování
- Za** - Z-ová souřadnice koncového bodu kružnice v absolutním programování
- Zi** - Z-ová souřadnice koncového bodu kružnice v inkrement. programování

3.3.2 Zadání kružnice koncovým bodem a středem

Programuje se koncový bod kružnice (absolutně nebo přírůstkově) a souřadnice středu (vždy jen přírůstkově od počátku kružnice). Souřadnice středu (interpolací parametry) se programují podle ISO adresami I, K nebo systémovými parametry CCX, CCZ.

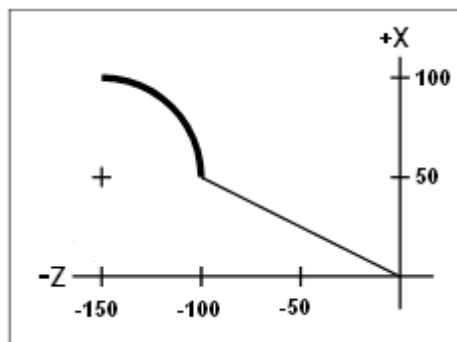
I, K, resp. interpolací parametry jsou přiřazeny osám takto:

- I** resp. CCX - vzdálenost středu od počátku ve směru osy X (s ohledem na znaménko)
- K** resp. CCZ - vzdálenost středu od počátku ve směru osy Z (s ohledem na znaménko)

Důležité upozornění

U soustruhů a karuselů se souřadnice v ose X většinou programují **průměrově** (i když to není pravidlem). V následujících příkladech je použit tento způsob zadání. U příkladu s poloměrovým zadáním je to výslovně uvedeno

Příklad programování kružnice v rovině X-Z s počátečním bodem [50,-100], koncovým bodem [100,-150] a poloměrem kružnice 50 (orientace os je pro soustruh). V **průměrovém** zadání je počáteční bod v ose X na průměru 100 a koncový bod v ose X na průměru 200:



Absolutní programování:

```
N G90 G3 X200 Z-150 I0 K-50 F100
```

Přírůstkové programování:

```
N G91 G3 X100 Z-50 I0 K-50
```

Při přírůstkovém programování (G91) se přírůstky v ose X také zadávají průměrově (pokud programujeme průměrově) !

Pokud se uvedený příklad programoval **poloměrově** (v ose X), vypadaly by bloky následovně:

Absolutní programování:

N G90 G3 X100 Z-150 I0 K-50 F100

Přírůstkové programování:

N G91 G3 X50 Z-50 I0 K-50

Důležité upozornění

Bez ohledu na to, zda programujeme souřadnice X průměrově nebo poloměrově, souřadnice středu se programují **vždy přírůstkově** od počátku kružnice a to s ohledem na znaménko.

Následuje příklad stejné kružnice, ale v opačném směru, tj. pokud bychom ve výše uvedeném příkladu jeli kružnicí z bodu [100,-150] do bodu [50,-100], vypadal by blok následovně (při průměrovém programování):

Absolutně:

N G90 G2 X100 Z-100 I-50 K0

Přírůstkově:

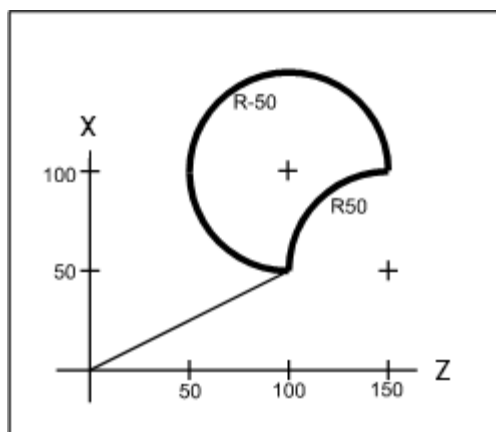
N G91 G2 X-100 Z50 I-50 K0

3.3.3 Zadání kružnice koncovým bodem a poloměrem

Často je jednodušší zadávat kružnici nikoli souřadnicemi středu, ale jejím poloměrem. Programují se opět koncové body kružnice a poloměr kružnice – buď adresou R nebo systémovým parametrem CR (Circle Radius)

Protože při zadání kružnice poloměrem vycházejí matematicky dvě řešení (viz obr.), bere se při programování kladného R kružnice s výsečí menší než 180 stupňů, při programování záporného R se vezme kružnice větší než 180 stupňů. (viz obr.)

Příklad (opět průměrové programování)



Absolutně:

N G90 G2 X200 Z150 R50

Nebo zápisem pomocí systémového parametru CR

N G90 G2 X200 Z150 CR=50

Přírůstkově:

N G91 G2 X100 Z50 R50

Nebo zápisem pomocí systémového parametru CR:

N G91 G2 X100 Z50 CR=50

Pokud bychom chtěl programovat větší z obou kružnic (druhé řešení), programovali bychom blok takto (rozdíl je pouze v záporném znaménku u poloměru) :

Absolutně:

N G90 G2 X200 Z150 R-50

Nebo zápisem pomocí systémového parametru CR

N G90 G2 X200 Z150 CR=-50

Přírůstkově:

N G91 G2 X100 Z50 R-50

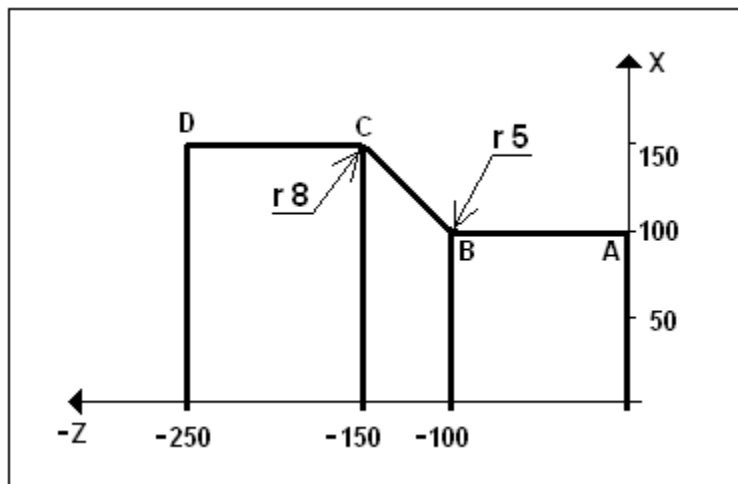
Nebo zápisem pomocí systémového parametru CR:

N G91 G2 X100 Z50 CR=-50

Kruhová interpolace není omezena na jeden kvadrant. Programovaná kružnice může procházet i více kvadranty.

3.3.4 Zaoblení rohu se zadaným poloměrem - G11

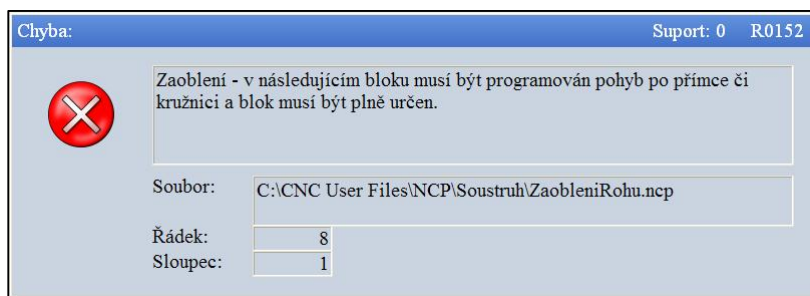
Podobně jako u zkosení lze mezi dvě programované úsečky A-B a B-C a B-C a C-D (viz obr.) vložit zaoblení se zadaným poloměrem, pokud známe souřadnice průsečíku přímk (body B a C, viz obr.). Programuje se funkcí G11 a systémovým parametrem ROUND. Blok se vloží mezi programované úsečky.



Příklad pro **průměrové** programování.
Zaoblení v bodu B je 5mm, zaoblení
v bodu C je 8mm

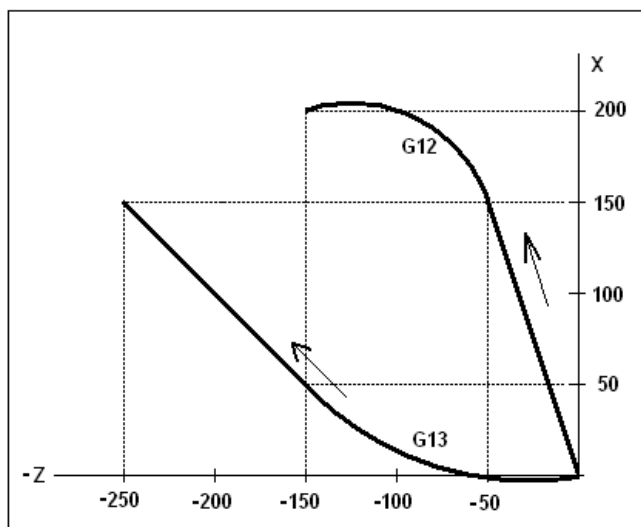
```
N G0 X200 Z0
N G1 X200 Z-100 F100
N G11 Roundr=5
N G1 X300 Z-150
N G11 Roundr=8
N G1 X300 Z-250
```

Upozornění: v bloku, následujícím za blokem, kde je programována funkce G11, nesmíme zapomenout programovat typ interpolace (G0,G1,G2,G3), jinak se vyhlásí chyba:



3.3.5 Kruhová interpolace s tečným napojením

Pro usnadnění programování kruhové interpolace jsou k dispozici dvě funkce



G12 – kruhová interpolace s tečným napojením na **předchozí** blok

G13 – kruhová interpolace s tečným napojením na **následující** blok

Programuje se pouze koncový bod kružnice, ostatní parametry dopočítá systém z podmínky, že kružnice je napojena z předchozího bloku tečně (G12), resp. následující blok je napojen tečně (G13).

Příklad pro funkce G12 a G13 (dle obrázku – průměrové programování):

```
N G90 G0 X0 Z0
N G1 X300 Z-50 F100
N G12 X400 Z-150 „ kružnice tečně napojená na přímku
N G0 X0 Z0
N G13 X100 Z-150 „ kružnice se napojí tečně na následující přímku
N G1 X300 Z-250
N G0 X0 Z0
```

Funkce G12 (a G13) se s výhodou používají při navazování kružnic, jak ukazuje následující příklad.

```
N PROGRAM
N G90 G0 X0 Z0
N G1 X100 Z-50 F100
N G12 X200 Z-200
N G12 X200 Z-300
N G12 X300 Z-400
N G12 X350 Z-450
N ENDPROGRAM
```

3.4 Závítování

Závítování nožem (bez použití závítovacích cyklů) se programuje funkcí G33 a dále popsanými systémovými parametry. Ty jsou deklarovány v hlavičkovém souboru StdHdr.nch, který se standardně připojuje ke každému partprogramu. Uvedené systémové parametry je možné používat v libovolném partprogramu (viz příklady)

Při programování řezání závitu nožem funkcí G33 sváže systém pohyb v souřadnici, pro níž bylo zadáno stoupaní, s pohybem vřetene. Pohyb ostatních interpolačních souřadnic je prováděn tak, aby byl výsledný pohyb v zadaných souřadnicích proveden po zadané trajektorii.

Seznam systémových parametrů pro závitování

TPZ	Stoupání závitu ve směru osy Z (Thread Pitch in Z)
TSPO	Úhlové posunutí začátku závitu pro vícechodé závity (Thread Starting Point Offset)
TRI	Délka nájezdu závitu (Thread Run-In length)
TRO	Délka výjezdu závitu (Thread Run-Out length)
TRIA	Úhel nájezdu závitu (Thread Run-In Angle)
TROA	Úhel výjezdu závitu (Thread Run-Out Angle)

V následujících příkladech je uvedeno závitování na soustruhu. Stoupání se udává ve směru osy vřetene, tj. v ose Z.

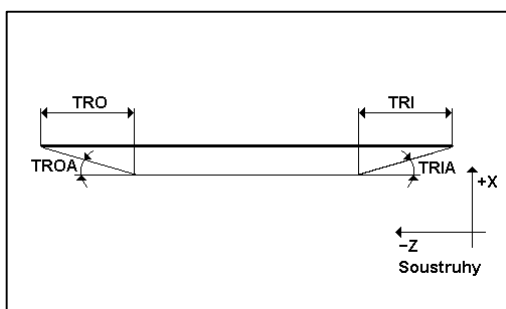
Příklad jednoho bloku se závitem bez výjezdu:

```

N G90 G0 X0 Z100 M3 M44 S400 " Otáčky
N G0 X30 Z2 " Nájezd do výchozího bodu závitování
N G33 Z-100 TPZ=1.5 " Závitování se stoupáním 1.5mm
N G0 X40 " Odskok od závitu
N Z2 " Návrat do výchozího bodu v ose Z
    
```

Funkce G33 (podobně jako G95 - otáčkový posuv mm/ot) sváže pohyb podle programovaných otáček. Oproti funkci G95 zajistí funkce G33 navíc „čekání“ na nulový pulz, kterým se závitování odstartuje. Tím je zajištěn definovaný začátek závitování, nutný pro opakovaný průchod závitem.

- TPZ = Stoupání závitu ve směru osy Z, udává dráhu osy na jednu otáčku. Rychlost posuvu je tedy závislá na otáčkách vřetene.
- TSPO = Úhlové posunutí začátku závitu. Pro jednochodý závit se nemusí uvádět. Pro dvouchodý závit je TSPO = 0 a TSPO = 180, tj. na jedné poloze osy X (u soustruhů) se provede nejprve závit s TSPO=0 a po návratu do Z-tového začátku závitu ve stejné poloze osy X se provede závit s TSPO=180. Podobně by se třikrát na stejné poloze X provedl závit tříchodý (TSPO = 0, TSPO = 120, TSPO = 240). Např. TSPO=240 znamená, že po nulpulzu se vřeteno otočí ještě o 240°, než se posuv rozjede.



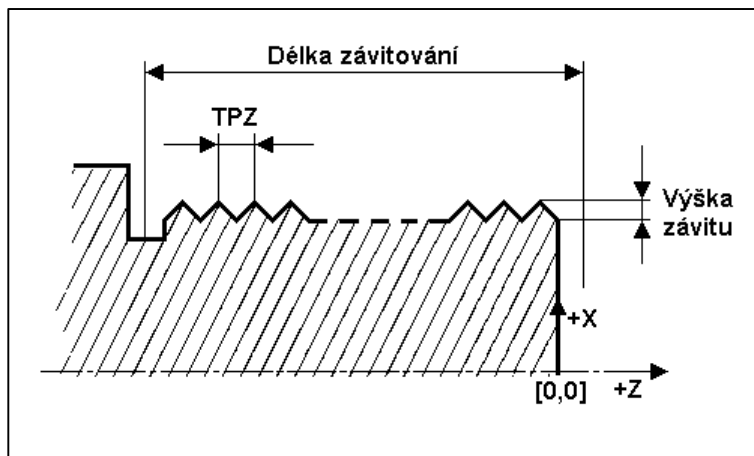
- TRO = délka výjezdu ze závitu – udává délku v ose Z, kdy se závituje, ale současně se v druhé ose (X) „vyjíždí“ ze závitu pod úhlem, který je programován v systémovém parametru TROA. **Programovaná koncová poloha závitu (délka závitu) je v tomto případě včetně délky výjezdu!**

- TROA = úhel výjezdu ze závitu. Pod tímto úhlem se vyjíždí ze závitu po délce, která je programovaná v systémovém parametru TRO. Parametry TRO a TROA by měly být vzájemně smysluplné.
- TRI, TRIA = délka a úhel vjezdu do závitu – mají podobný význam jako u výjezdu, jejich použití ale není tak časté.

Jednochodý závit M10x1.5 bez výjezdu (do zápichu):

V následujícím příkladu je rozepsaný závit M10 se stoupáním 1.5mm. Délka závitu je programována tak, aby závit začínal před materiálem (např. 2mm) a končil v zápichu, ze kterého vyjede v ose X. Pro jednochodý závit bez výběhu je prakticky nutné ze systémových parametrů nastavit pouze stoupání, tj. parametr TPZ.

Pro závit M10x1.5 je výška závitu podle tabulek 0.92mm. Tato výška závitu se obrobí na několik průchodů. Na konečné hloubce se obvykle provedou ještě např. dva průchody kvůli začištění.



Tloušťka třísky je proměnlivá tak, aby průřez odebíraného materiálu byl stejný (koeficient zmenšení, tj. kolikrát je následující třísky menší, je obvykle 0.8).

Pokud se nepoužije závitování cyklus, musíme X-ové souřadnice podle počtu třísek spočítat „ručně“

" Závit M10x1.5 – průměrové programování

" Výška závitu je 0.92

N1 PROGRAM

N10 X0 Z100 G00

N30 X12 Z2 M3 S300

"

" Nájezd do výchozí polohy, otáčky 300ot/min

N100 X9.459

" X-ová poloha 1. třísky

N110 G33 TPZ=1.5 Z-52

" Závitování 1. třísky

N120 X12 G00

" Výjezd v X do polohy pro přejezd zpět

N130 Z2

" Návrat v Z do výchozí polohy

"

N200 X9.016

" X-ová poloha 2 třísky

N210 G33 TPZ=1.5 Z-52

" Závitování 2. třísky

N220 X12 G00

" Výjezd v X do polohy pro přejezd zpět

N230 Z2

" Návrat v Z do výchozí polohy

"

N300 X8.664

" X-ová poloha 3. třísky

N310 G33 TPZ=1.5 Z-52

" Závitování 3. třísky

N320 X12 G00

" Výjezd v X do polohy pro přejezd zpět

N330 Z2

" Návrat v Z do výchozí polohy

"

" Další třísky

"

N400 Z100 G00 M5

" Odjezd do bezpečné polohy

N500 ENDPROGRAM

Jednochodý závit M10x1.5 s výjezdem:

Ten samý závit s výjezdem o délce 3.75mm a úhlem výjezdu 14° by se programoval podobně, pouze se přidají dva parametry definující délku a úhel výjezdu. Pro ilustraci je uveden pouze jeden průchod:

```
N1 PROGRAM
N10 X0 Z100 G00
N30 X12 Z2 M3 S300           " Nájezd do výchozí polohy
"
N100 X9.459                  " X-ová poloha 1. třísky
N110 G33 TPZ=1.5 TRO=3.75 TROA=14 Z-52 " Závitování 1. třísky s výběhem
N120 X12 G00                  " Výjezd v X
N130 Z2                       " Návrat v Z
...
...
```

Dvouchodý závit M10x1.5 bez výjezdu:

```
N1 PROGRAM
N10 X0 Z100 G00
N30 X12 Z2 M3 S300           " Nájezd do výchozí polohy,
"
N100 X9.459                  " X-ová poloha 1. třísky
N110 G33 TSPO=0 TPZ=1.5 Z-52 " Závitování 1. třísky - první chod
N120 X12 G00                  " Výjezd v X do polohy pro přejezd zpět
N130 Z2                       " Návrat v Z do výchozí polohy
"
N101 X9.459                  " X-ová poloha 1. třísky
N111 G33 TSPO=180 TPZ=1.5 Z-52 " Závitování 1. třísky - druhý chod
N121 X12 G00                  " Výjezd v X do polohy pro přejezd zpět
N131 Z2                       " Návrat v Z do výchozí polohy
"
N200 X9.016                  " X-ová poloha 2 třísky
N210 G33 TSPO=0 TPZ=1.5 Z-52 " Závitování 2. třísky - první chod
N220 X12 G00                  " Výjezd v X do polohy pro přejezd zpět
N230 Z2                       " Návrat v Z do výchozí polohy
"
N201 X9.016                  " X-ová poloha 2 třísky
N211 G33 TSPO=180 TPZ=1.5 Z-52 " Závitování 2. třísky - druhý chod
N221 X12 G00                  " Výjezd v X do polohy pro přejezd zpět
N231 Z2                       " Návrat v Z do výchozí polohy
"
..... atd.
```

4

4. DÉLKOVÉ A POLOMĚROVÉ KOREKCE

4.1 Nastavování délkových a poloměrových korekcí

Korekce, o kterých bude pojednáno v této kapitole, se nastavují pomocí softwarových tlačítek, který se objeví po stisku tlačítka „Technologie“. Příklad menu je na následujícím obrázku.



Menu může být v některých případech pozměněno, neboť jej sestavuje tvůrce PLC programu, v jehož projektu je zahrnuto.

Délkové korekce i poloměrovou korekci je možné nastavit dvěma základními způsoby a to podle způsobu jejich měření – buď na externím měřicím přípravku nebo přímo na stoji.

4.2 Nastavení korekcí v tabulce nástrojů

Délkové korekce i poloměrovou korekci pro jednotlivé nástroje je možné nastavit přímo v tabulce nástrojů. Tento způsob se používá v případě, že délkové korekce nástrojů se měří na externím měřicím přípravku. Hodnoty, které odměříme, se zapíší přímo do tabulky. Dotykem na řádek požadovaného nástroje ho připravíme pro editaci. Do políček „Délka X“, resp. „Délka Z“ zapíšeme naměřené hodnoty délkové korekce. Hodnota délkové korekce pro osu X se do tabulky zadává vždy poloměrově (i v případě průměrového programování)!

 A dialog box titled 'Nástroje' (Tools). It contains a table with two columns: 'Č. nástroje' (Tool number) and 'Nástroj' (Tool). The table lists tools 0 through 5: 0 Nulový nástroj, 1 Hrubovací nůž, 2 Šlichtovací nůž, 3 Vrták 15, 4 Vrták 20, and 5. To the right of the table are up and down arrow buttons. Below the table, there are input fields for 'Nástroj č.: 1', 'Název: Hrubovací nůž', 'Poloměr: 0.800 mm', 'Délka X: 15.400 mm', 'Délka Z: 125.000 mm', 'Typ: 3', and 'Popis: Hr1- 257'. To the right of these fields is a small diagram of a tool with numbered points 1 through 9. At the bottom are buttons for 'Přidat nový', 'Odebrat poslední', and 'OK'.

Pokud technolog využívá i poloměrové korekce, zapíše poloměr špičky nástroje do políčka „Poloměr“ a zároveň uvede orientaci nástroje vzhledem k obrobku podle uvedeného obrázku – příslušnou číslici zapíše do políčka „Typ“. Tvůrce PLC může obrázek modifikovat podle orientace os na skutečném stroji.

Do políčka „Název“ se uvede název nástroje, jeho bližší určení je možné zapsat do políčka „Popis“. Název a Popis jsou nepovinné údaje – slouží pouze pro orientaci.

Stiskem tlačítka „Přidat nový“ se do tabulky doplní řádek pro nový nástroj. Počet nástrojů je omezen pouze vymezenou pamětí, pro většinu aplikací je s velkou rezervou dostatečný. Tlačítkem „Odebrat poslední“ lze tabulku zmenšovat odebráním řádků.

4.3 Nastavení na stroji „škrtnutím“ o obrobek

Pokud se nepoužije externí měřicí zařízení, nebo ho konstrukce stroje, resp. upínací hlavy dokonce neumožňuje využít, mohou se korekce nastavit tzv. „škrtnutím“. Je nutné pouze vědět a následně zadat míry, na kterých se „škrtno“, zápis se provede automaticky.

Postup:

Před měřením se doporučuje nejprve zvolit a odstartovat Centrální anulaci a potom v režimu Ruční předvolby (RUP) zadat potřebné otáčky S a směr otáčení (M3,M4) a odstartovat. Zvolíme manuální režim a najedeme na obrobek ve směru osy X (buď ručně nebo pomocí točítka apod.) a „škrtneme“ si o obrobek. Je možné také obrobit libovolný průměr, např. v režimu RUP. Do doby zápisu korekce nesmíme polohu osy X změnit, tj. lze vyjet z obrobku, pokud je to nutné, pouze v ose Z. Pokud potřebujeme změřit průměr, stopneme vřetenem buď Centrální anulaci nebo v režimu RUP (M5)

Stiskneme tlačítko „Technologie“ a dále „Nastavení korekce X“. Objeví se zadávací dialog (viz obr.). Do příslušných okének zadáme číslo nástroje a hodnotu změřeného **průměru**, který jsme si „škrtnli“ nebo obrobili. Dále musíme zvolit, zda jsem „škrtnání“ prováděli v kladné nebo záporné polorovině. Pozor, záleží pouze na tom, kde se nachází špička nože bez ohledu na orientaci nože. Polorovina se volí stiskem příslušného radiobuttonu.

Po zadání hodnot a poloroviny se stiskne OK. Provede se automatický zápis do tabulky a zvolení měřeného nástroje, tj. na displeji se musí objevit průměr, který jsme zadali.

Podobným způsobem se nastaví délková korekce v ose Z:

Po stisku tlačítka „Nastavení korekce Z“ se objeví dialog (viz obr.). Opět se zadá číslo nástroje a do políčka „Vzdálenost od nuly“ se zadá hodnota, jak daleko jsme od nuly. Pokud si např. škrtneme o obrobené čelo, kde máme nulu, zadáme do políčka nulu. Můžeme zadat přídavek, např. 1 mm. To pak znamená, že se bude po potvrzení OK indikovat hodnota 1. Pojedeme-li v programu na nulu, obrobíme onen zadaný 1 milimetr.

4.4 Změna hodnoty délkové korekce

Po nastavení délkových korekcí podle předešlého postupu a obrobení kusu můžeme po proměření zjistit, že skutečné míry se liší od požadovaných výkresových. Obvykle to bývá maximálně o desítky mikronů. Hodnoty korekcí v tabulce nástrojů můžeme upravit opět dialogovým způsobem, abychom nemuseli „ručně“ odečítat hodnoty a zapisovat je do tabulky nástrojů.

Změna v ose X se provede po stisku tlačítka „Změna korekce X“.

Objeví se dialogové okno (viz obr.), do kterého opět zadáme číslo nástroje, pro který chceme změnu korekce a do druhého políčka zadáme změnu průměru, včetně znaménka, tj. zda chceme korekci zvětšit (znaménko +) nebo zmenšit (znaménko -).

Hodnota se zadává průměrově !

Po potvrzení se změna přičte nebo odečte od hodnoty v tabulce nástrojů.

Opětovným vyvoláním je změna přednastavená na nulu, tj. nepamatuje si předešlé nastavení.

Pro změnu korekce Z se stiskne tlačítko „Změna korekce Z“.

Objeví se dialogové okno pro změnu korekce ose Z (viz obr.). Zadáme číslo nástroje a změnu korekce s ohledem na znaménko.

Po potvrzení se změna přičte nebo odečte od hodnoty v tabulce nástrojů.

Opětovným vyvoláním je změna přednastavená na nulu, tj. nepamatuje si předešlé nastavení.

4.5 Programování korekcí v partprogramu

V partprogramu se požadovaný nástroj vybere příkazem **Tool(Par1,Par2)**. Výběrem nástroje mu současně přiřadíme jeho korekce – poloměrovou na špičku nástroje včetně orientaci špičky a délkové korekce nástroje v ose X a Z.

Příkaz Tool má dva parametry:

Par1 = číslo nástroje (1,2,3,...)

Par2 = zařadit/vyřadit poloměrovou korekci

G-funkce pro poloměrovou korekci jsou:

G40 = zrušení poloměrové korekce

G41= poloměrová korekce vlevo

G42= poloměrová korekce vpravo

V parametru Par2 se však zadává pouze číselná hodnota, tj. 40, 41, nebo 42

Příklad:

N Tool(4,40) „Nástroj číslo 4, bez poloměrové korekce

Poloměrové korekce velmi usnadňují programování šikmých drah. Programátor nemusí pracně přepočítávat souřadnice, programuje pouze výkresové míry. Systém sám přepočítá průsečíky ekvidistant.

Pokud se použije poloměrová korekce, tj. Par2 je 41 nebo 42, musí se příkaz Tool(Par1,Par2) zapsat do pohybového bloku, tj. v bloku, kde je programována osa X nebo osa Z, případně obě osy najednou!

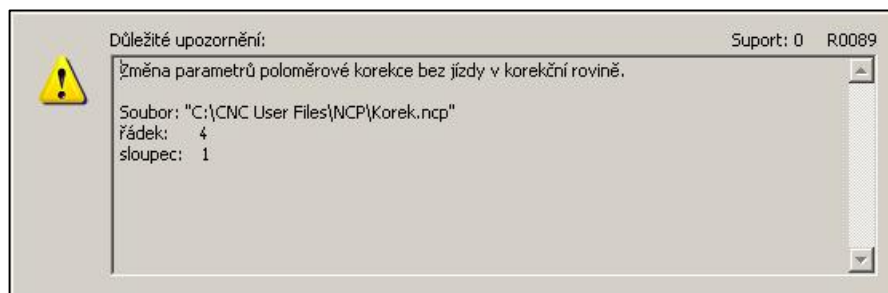
Příklad:

N X20 G01 Tool(2,41) „Nástroj č.2, zařadí se poloměrová „korekce vlevo

Pokud by se v bloku programovalo pouze:

N Tool(2,41)

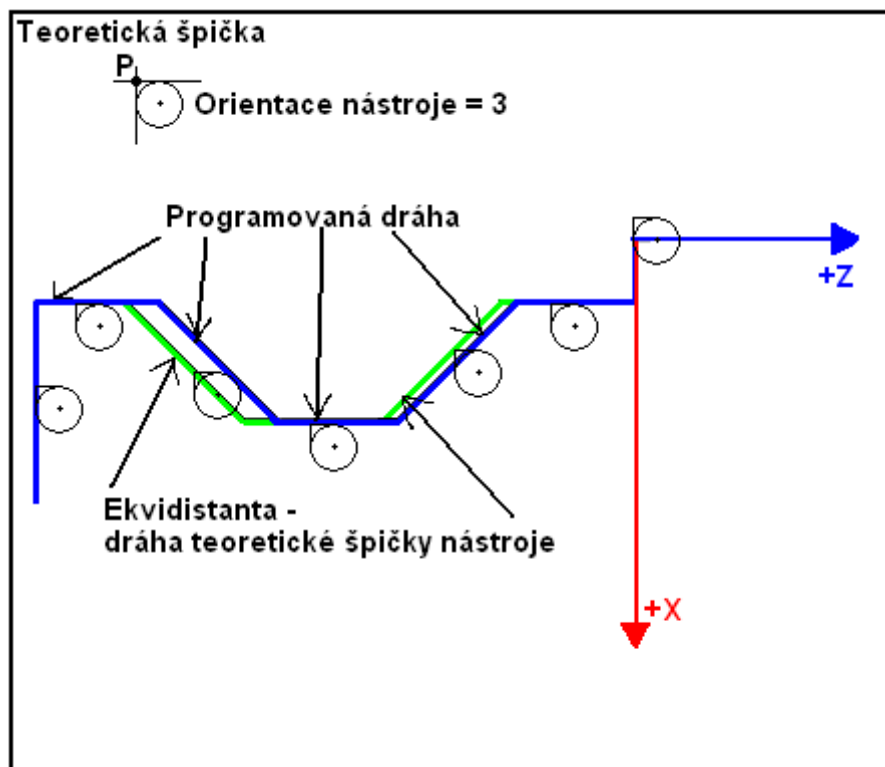
hlásí se „Důležité upozornění“ :



Pokud se korekce zařadí v nepohybovém bloku, nemůže systém jednoznačně spočítat, kam má najet a může nastat případ, že se korekce zařadí správně až v druhém pohybovém bloku po zařazení korekce.

Nemusí to vadit, pokud jsme mimo materiál, nicméně doporučuje se držet se pravidla řadit poloměrovou korekci v pohybovém bloku! Tím jsou nájezdy na korekci jednoznačně dané.

Průběh korigované dráhy je na obrázku:



Pozn.:

Grafické znázornění dráhy teoretické špičky (zelená čára) je správně zobrazeno pouze v případě, že osa X směřuje nahoru. Pokud chceme zobrazení přizpůsobit tak, aby odpovídalo skutečnému pohledu na stroj, jako v našem příkladu (osa X směřuje dolů), je dráha teoretické špičky (zelená čára) zrcadlově převrácená na druhou stranu (jako bychom místo G42 programovali G41 a naopak)

Partprogram by byl např. takto zapsán:

```

N PROGRAM
N10 X0 Z20 G0 M3 S500 Tool(1,40) " Nastroj č.1
N20 Z0 G01 F0.2 Tool(1,42) " Zařadí G42
N30 X10
N40 Z-10
N50 X30 Z-20
N60 Z-30
N70 X10 Z-40
N80 Z-50
N90 X50
N100 Z10 G00 G40 " Vyřazení poloměř. korekce
N110 X0
N ENDPROGRAM

```

Je vidět, že u vodorovné a svislé dráhy se výkresová dráha shoduje s dráhou teoretické špičky nože. Pouze u šikmých drah je ekvidistanta mimo programovanou dráhu. Po ekvidistantě se pohybuje teoretická špička nože, tj. takže nůž (kulatá špička nože) správně jede po výkresové dráze.

5

5. Funkce

Funkce jsou systémová předdefinovaná makra, která usnadňují programování, zpřehledňují tvorbu partprogramu. Přehled všech dostupných systémových funkcí je uveden v Návodu k programování. Zde uvádíme pouze malý výběr některých funkcí a příklad použití.

5.1 Matematické a rozhodovací funkce

Plus(Val1, Val2)	Sčítání
Minus(Val1, Val2)	Odčítání
Multiply(Val1, Val2)	Násobení
Divide(Val1, Val2)	Dělení
Eq(Val1, Val2)	Rovno
NEq(Val1, Val2)	Nerovno
Less(Val1, Val2)	Menší
LE(Val1, Val2)	Menší nebo rovno
Greater(Val1, Val2)	Větší
GE(Val1, Val2)	Větší nebo rovno
If(Val)	Podmínka, podmínka splněna, je-li Val nenulová hodnota
Else	Else
EndIf	Konec podmínky

Funkce pro základní matematické operace sčítání, odčítání, násobení a dělení sečte (odečte, vynásobí, vydělí) dva operandy, výsledek je návratová hodnota funkce, t.j. musí se uložit do nějaké proměnné, jinak se hlásí chyba.

Příklad:

Provedeme tyto operace se dvěma reálnými čísly, které jsou uloženy v proměnných (t.j. v parametrech) CISLO1 a CISLO2, výsledek uložíme do proměnné VYSLEDEK. Při deklaraci proměnných použijeme automatické přidělení čísla reálného parametru pomocí RPARAM (podrobněji viz kapitola Programování parametrů) :

```

$CISLO1      RPARAM
$CISLO2      RPARAM
$VYSLEDEK    RPARAM
.....
N  CISLO1 = 14.56
   CISLO2 = 28.30
N  VYSLEDEK = Plus(CISLO1,CISLO2)
N  VYSLEDEK = Minus(CISLO1,CISLO2)
N  VYSLEDEK = Multiply(CISLO1,CISLO2)
N  VYSLEDEK = Divide(CISLO1,CISLO2)
N  VYSLEDEK = UMinus(CISLO1)
.....

```

V proměnné VYSLEDEK se postupně objeví hodnoty: 42.86, -13.74, 421.048, 0.514488 a -14.56

Pozn:

Tyto základní matematické operace se mohou v partprogramu psát jednoduše i klasickým zápisem:

```
N CISLO1 = 14.56
  CISLO2 = 28.30
N VYSLEDEK = CISLO1 + CISLO2
N VYSLEDEK = CISLO1 - CISLO2
N VYSLEDEK = CISLO1 * CISLO2
N VYSLEDEK = CISLO1 / CISLO2
N VYSLEDEK = -CISLO1
```

Příklad pro logické operace:

Provedeme vyhodnocení zápisu, ve kterém se porovnávají hodnoty pomocí funkcí **GE()**, **LE()**, **AND()** a **NOT()**, ve kterém pracujeme s celočíselnými parametry I1, I2, I3, I4 které budou nastaveny na hodnoty 10, 20, 30, 40. Podle výsledků se nastaví parametr R5 na hodnotu 100.0 nebo 200.0 a následujícím bloku odjede osa X míru, nastavenou v tomto parametru (v našem zadání se nastaví R5=100.0)

Matematicky by podmínka zapsala:

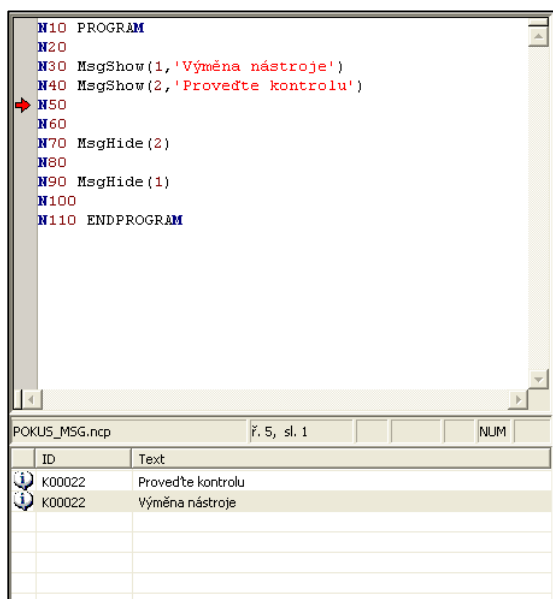
If ((NOT (R1 >= R2)) AND (R3 <= R4))

```
N PROGRAM
N10 I1=10 I2=20 I3=30 I4=40 R5=0
N20 If (AND (NOT (GE (R1,R2)), LE (R3,R4))) R5=100.0 Else R5=200.0 Endif
N30 XR5 G01
N40
N ENDPROGRAM
```

5.2 Funkce pro oznamování chyb, upozornění a informací

Funkce:	MsgShow(ShowNo, MsgNo)
Parametry:	ShowNo – Pořadí zprávy, má význam při nutnosti zobrazovat více zpráv najednou. MsgNo - text zprávy mezi 'apostrofy' nebo identifikační číslo zprávy, uvedené v deklaraci zprávy (deklarace může být kdekoli v partprogramu před jejím vyvoláním nebo libovolném souboru, vloženém do partprogramu pomocí klíčového slova INL (include) .

Funkce:	MsgHide(ShowNo)
Parametry:	ShowNo – pořadí zprávy, která se má smazat. Pokud se uvede 0, smažou se všechny zprávy.



Uvedené funkce slouží pro výpis zpráv, které technolog zapíše do partprogramu.. Zprávy se vypisují při chodu partprogramu v okně „Trvalé chyby a hlášení“.

Po vykonání bloku N40 jsou zobrazeny obě zprávy (viz obr.). V bloku N30 i N40 by mohla být místo textů v apostrofech uvedena čísla zpráv. Obě variant jsou možné. Zprávy se automaticky řadí tak, že naposledy vyslaná zpráva je na prvním řádku.

Po vykonání bloku N70 zmizí zpráva, vyslaná jako druhá v pořadí, po vykonání bloku N90 zmizí zpráva vyslaná jako první v pořadí. Pokud se kdekoli uvede MsgHide(0), zmizí všechny zprávy.

Pokud by se texty deklarovaly např. na začátku partprogramu nebo v libovolném souboru, který by se

připojil pomocí klíčového slova INL, byly by texty deklarovány např. takto:

```
&1 '...'
&2 '...'
&3 ' Výměna nástroje'
&4 ' Proveďte kontrolu'
&5 '...'
```

a v partprogramu by bylo uvedeno:

```
N30 MsgShow(1,3)
N40 MsgShow(2,4)
```

Výpis proměnných v textu zprávy

Kromě statického výpisu zpráv, jak bylo uvedeno v předešlém příkladu, je možné vypisovat v textu i dynamicky měnící se proměnné, což se může s výhodou používat např. pro různé čítače, počítadla kusů apod. Výpis proměnných umožňují tzv. escape sekvence, které umožňují vkládání speciálních znaků a parametrů do řetězce. Tyto sekvence začínají zpětným lomítkem:

\r Místo této sekvence se vloží reálná hodnota, která je uvedena jako parametr funkce MsgShow
\i Místo této sekvence se vloží celočíselná hodnota, která je uvedena jako parametr funkce MsgShow

Kromě proměnných se mohou vkládat i další speciální znaky:

\n Místo této sekvence se vloží nový řádek
\nnn Místo této sekvence se vloží znak s ASCII hodnotou nnn (zapsané dekadicky a povinně tři cifry)
\xhh Místo této sekvence se vloží znak s ASCII hodnotou zapsanou hexadecimálně (vždy dvě cifry)
\' Místo této sekvence se vloží apostrof (apostrof nelze psát do řetězce přímo!)
**** Místo této sekvence se vloží zpětné lomítko. To nelze psát přímo, musí se zdvojit, máli být

Následující příklad uvádí možnost výpisu zprávy, kde chceme zobrazit úhel β a odchylku v mikronech. Úhel bude např. vypočítán a uložen do reálné proměnné UHEL, podobně odchylka bude tentokrát v celočíselné proměnné ODCHYLKA.

Příklad:

```
$UHEL R100 " Přiřazení proměnné reálnému parametru
$ODCHYLKA I100 " Přiřazení proměnné celočíselnému parametru
```

```
N PROGRAM
N UHEL = 45.5 ODCHYLKA = 10
N MsgShow(1,'úhel \223 = \r, povolená odchylka = \i \181m',UHEL,ODCHYLKA)
N ...
N MsgHide(0)
N ENDPROGRAM
```

V okně pro výpis zpráv a chybových hlášení bude vypsáno:

```
'úhel  $\beta$  = 45.50000, povolená odchylka = 10 $\mu$ m
```

Escape sekvence \223 byla nahrazena znakem β

Escape sekvence \r byla nahrazena hodnotou reálné proměnné UHEL

Escape sekvence \i byla nahrazena hodnotou celočíselné proměnné ODCHYLKA

Escape sekvence \181 byla nahrazena znakem μ

Pozn.: kódy znaků lze získat v libovolné ASCII tabulce.

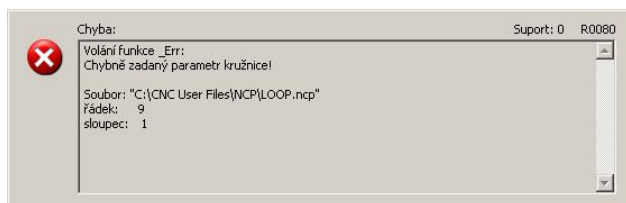
Příklad výpisu čítače cyklů. Čítač „Cit“ se nastaví na začátku programu na hodnotu 10. V cyklu se dekrementuje. Vypisuje se zpráva, kolik cyklů ještě zbývá. Na konci programu v bloku N50 je použit rozhodovací blok, zda se čítač nerovná nule. Pokud ne, jde se znovu do cyklu. Pokud ano, program se ukončí.

```

$Cit I1
N PROGRAM
N10 Cit = 10 X0 G0 G90
N20 MsgShow(1,'Zbývá cyklů: \i',Cit)
N30 AxGx = AxGx + 10 G01 F1000
N40 Cit=Cit-1
N50 if(NEq(Cit,0)) Jmp(20) endif "Není-li čítač nula, skok na N20
N99 ENDPROGRAM

```

Funkce:	Err(MsgNo)
Parametry:	MsgNo – číslo chybového hlášení.



Funkce je podobná předešlé funkci pro výpis zpráv. Rozdíl je pouze v zobrazení. Chyba, jejíž text je deklarován stejným způsobem jako zpráva, se zobrazí v chybovém okně uprostřed obrazovky (viz obr.) po **přípravě bloku**, což je ve většině případů (není to ale podmínka) už po volbě partprogramu. Chyba se musí kvitovat tlačítkem OK. Tímto způsobem může technolog pomoci načtení

systémových parametrů např. zkontrolovat otáčky a eventuálně vyhlásit chybu nebo zkontrolovat parametry kružnice např. na rozsah apod. V partprogramu je zápis možný dvěma způsoby. Buď uvést jako parametr funkce Err číslo chyby (její deklarace je kdekoli v partprogramu, obvykle na začátku uvedena jako &1 a text v apostrofech) a nebo jako parametr uvést přímo text mezi apostrofy.

Příklad:

```

&1 'Chybně zadaný parametr kružnice'
....
N320 if(Greater(Prumer,MaxPrum)
      Err(1)
      Err('Chybně zadaný parametr kružnice 1') " možno zapsat takto
      " nebo takto
    endif
....

```

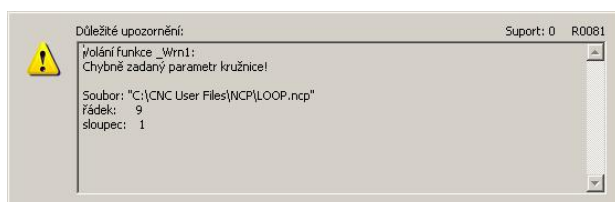
Funkce:	Wrn1(MsgNo) Wrn2(MsgNo) Wrn3(MsgNo)
Parametry:	MsgNo – číslo zprávy upozornění.

Tyto tři funkce jsou podobné předešlé funkci Err pro výpis chyby. Zápis je stejný. Používá se na upozornění obsluhy a lze je programovat ve třech stupních důležitosti.

Wrn1 je Důležité upozornění

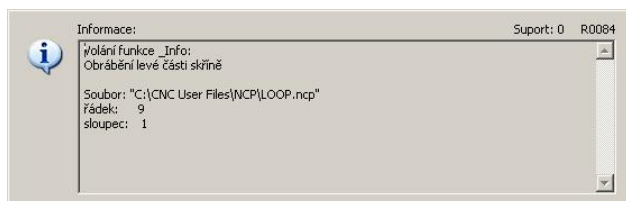
Wrn2 je Upozornění

Wrn3 je Málo důležité upozornění.



Ve všech případech se zobrazuje okno (viz obr.), liší se pouze nápisem nad rámečkem, kde je buď text „Důležité upozornění“ nebo „Upozornění“ nebo „Málo důležité upozornění“. Upozornění se musí kvitovat tlačítkem OK. Tímto způsobem může technolog provádět libovolné kontroly a případně hlásit tato upozornění.

Funkce:	Info(MsgNo)
Parametry:	MsgNo – číslo informačního hlášení.



Funkce podobná jako funkce Err – používá se stejný způsob zápisu, vypisuje se jako informační hlášení. Informační hlášení se musí kvitovat tlačítkem OK

6

6. Posunutí počátku

Posunutí počátku – nulové body programu – umožňují programovat přenesení souřadného systému programu do libovolného bodu podle potřeb programátora. Při přesunutí souřadného systému (nulového bodu) se vztahují všechny další absolutní míry na nový nulový bod. Při inkrementálním programování nemá posunutí nulového bodu vliv na koncový bod programované dráhy. Při absolutním programování je programovaná koncová poloha dráhy automaticky přepočítána na novou koncovou polohu vztaženou k příslušnému nulovému bodu. Indikace absolutní polohy (okamžité polohy) je vztažena vždy vůči některému bodu stroje nebo programu (nulovému bodu programu). Vůči kterému bodu je indikace vztažena, je určeno G funkcí z páté skupiny.

6.1 Tabulka posunutí

Pokud se používá posunutí podle ISO, programované funkcemi G53 – G59 nebo nový způsob, který není omezen počtem posunutí (viz dále), je potřeba nejprve naplnit tabulku posunutí, která je k dispozici (obvykle) v menu TECHNOLOGIE a dále TABULKA POSUNUTÍ.

Tabulka posunutí		
	X	Z
0:	0.000	0.000
1:	0.000	0.000
2:	0.000	0.000
3:	0.000	0.000
4:	0.000	0.000
5:	0.000	0.000
6:	0.000	0.000
7:	0.000	0.000
8:	0.000	0.000
9:	0.000	0.000
10:	0.000	0.000
11:	0.000	0.000
12:	0.000	0.000

Tabulka je uvedena na obr. Je číslována od nuly, položky se do tabulky přidávají tlačítkem „Přidat řádek“ resp. „Ubrat řádek“ – pro vymazání položky. Pokud se programuje podle ISO funkcemi G53 – G59, je přiřazení následující:

G53 = 0

G54 = 1

G55 = 2

G56 = 3

G57 = 4

G58 = 5

G59 = 6

Způsob, který umožní využívat v podstatě libovolný počet posunutí (omezeno pouze konfigurací, resp. vymezenou pamětí) se programuje funkcí G50:

G50 Lxxx,

kde xxx je číslo v tabulce posunutí.

Příklad:

N G50 L22 " Posunutí počátku, hodnota je v tabulce posunutí pod číslem 22

N G50 L0 " Výsledek stejný jako při programování G53 (v tabulce číslo 0)

N G50 L1 " Výsledek stejný jako při programování G54 (v tabulce číslo 1)

atd.

7

7. ZADÁNÍ POSUVU

Pracovní posuv po obráběné křivce se programuje podle ISO pod adresou F nebo systémovým parametrem FEED. Posuv je zadáván způsobem podle G-funkce skupiny 6, nebo-li tato funkce určuje rozměr adresy F.

Posuv		G - funkce	Rozměr rychlosti F nebo FEED
Milimetrový	mm/min	G94	mm_min
	mm/min s KŘR	G97	mm/min
Otáčkový	mm/ot	G95	mm/ot
	mm/ot s KŘR	G96	mm/ot

Pozn.: KŘR = konstantní řezná rychlost

Příklady:

" Milimetrové posuvy:

" Rychlost programována podle ISO adresou F:

```
N G94 F1000      "rychlost posuvu je 1000 mm/min .
N G94 F15000     "rychlost posuvu je 15000 mm/min.
N G94 F1.5       "rychlost posuvu je 1.5 mm/min.
N G94 F0.1       "rychlost posuvu je 0.1 mm/min.
```

" Rychlost programována systémovým parametrem FEED:

```
N G94 FEED=1000  "rychlost posuvu je 1000 mm/min .
N G94 FEED=15000 "rychlost posuvu je 15000 mm/min.
N G94 FEED=1.5   "rychlost posuvu je 1.5 mm/min.
N G94 FEED=0.1   "rychlost posuvu je 0.1 mm/min.
```

" Otáčkové posuvy:

" Rychlost programována podle ISO adresou F:

```
N G95 F10        "rychlost posuvu je 10 mm/ot .
N G95 F1.5       "rychlost posuvu je 1.5 mm/ot.
N G95 F0.1       "rychlost posuvu je 0.1 mm/ot.
N G95 F0.05      "rychlost posuvu je 0.05 mm/ot.
```

" Rychlost programována systémovým parametrem FEED:

```
N G95 FEED=10    "rychlost posuvu je 10 mm/ot .
N G95 FEED=1.5   "rychlost posuvu je 1.5 mm/ot.
N G95 FEED=0.1   "rychlost posuvu je 0.1 mm/ot.
N G95 FEED=0.05  "rychlost posuvu je 0.05 mm/ot.
```

Pozn.:

Stejně rozměry jsou i pro milimetrový nebo otáčkový posuv s konstantní řeznou rychlostí

7.1 Konstantní řezná rychlost

Konstantní řezná rychlost (KŘR) je způsob řízení otáček vřetene, kdy se ze zadané velikosti řezné rychlosti a okamžité polohy řídící souřadnice, neustále vypočítávají požadované otáčky vřetene.

7.2 Ovládání z partprogramu

Systém používá pro KŘR dvě funkce:

G96 – Konstantní řezná rychlost s posuvem mm/ot

G97 – Konstantní řezná rychlost s posuvem mm/min (u soustruhů a karuselů méně časté použití)

Řezná rychlost se zadává funkcí „S“, která má v tomto případě význam nikoli otáček, ale řezné rychlosti v metrech za minutu [m/min]. Například řezná rychlost 50m/min by se zadala hodnotou „S50.0“. Kromě toho je možno při programování využít systémový parametr „CCS“. V tomto případě se naprogramuje: „CCS=50.0“.

Z NC programu možno kdykoli zadat omezení otáček pro KŘR pomocí systémového parametru „SLIM“. Systém omezuje otáčky vzhledem k menší hodnotě z maximálních otáček převodového stupně a ze zadaného omezení.

Příklad:

N10 G0 X300 Z100 M44 M3	“rychloposuv na průměr 300, otáčky 100ot/min
SLIM=1000 S100	“omezení maximálních otáček na 1000ot/min
N20 G96 G1 F0.2 CCS=90 X50	“zařadí KŘR 90m/min, sjetí na průměr 50, posuv
	“F = 0.2mm/ot

Při programování rychloposuvu se mění otáčky rovněž podle průměru, pokud není funkce G96 odvolána programováním G94.

KŘR není omezena jen na pohybové bloky NC programu. Zařazení KŘR může proto být i v nepohybovém bloku. Také změna délkové korekce a změna posunutí počátku se uplatní okamžitě a nemusí se čekat na pohybový blok. Korekce a posunutí mají vliv na KŘR, protože KŘR se počítá na špičku nástroje a ne na suport. KŘR se může také uplatňovat v ručních pojezdech nebo v jiných druzích pohybu, například vlečení nebo pro pohyby řízené z PLC programu.

KŘR nikdy nepřepíná převodový stupeň, ale omezí otáčky na maximální otáčky daného převodového stupně (pokud není ještě jiné omezení otáček).

8

8. TECHNOLOGICKÉ FUNKCE

Význam některých funkcí uvedených v této kapitole může být modifikován tvůrcem interfejsu pro daný stroj. Přesný význam a seznam použitých M-funkcí musí dodat tvůrce interfejsu stroje.

8.1 Časová prodleva

Časovou prodlevu je možné programovat systémovou funkcí DELAY(t), kde t je čas v sekundách nebo podle ISO funkcí G04 ze skupiny 10 a adresou F, která v tomto případě neznámá rychlost, ale čas v sekundách!

Příklad:

N	DELAY (1)	"	časová prodleva 1 sec.
N	DELAY (3.5)	"	časová prodleva 3,5 sec.
N	DELAY (0.2)	"	časová prodleva 0.2 sec.
N	DELAY (20)	"	časová prodleva 20 sec.

Nebo ty samé hodnoty programované podle ISO

N	G04 F1	"	časová prodleva 1 sec.
N	G04 F3.5	"	časová prodleva 3,5 sec.
N	G04 F0.2	"	časová prodleva 0.2 sec.
N	G04 F20	"	časová prodleva 20 sec.

8.2 Otáčky vřetene

Otáčky vřetene se programují adresou S. Velikost otáček je reálné číslo. Rozsah je daný konfigurací stroje.

Příklad:

N10	S55	"	55 ot/min
N20	S0.5	"	0,5 ot/min

Zařazení otáčkového rozsahu převodového stupně je podle doporučení programováno funkcemi M41, M42, M43, M44.

Start otáčení ve směru CW je zadán funkcí M3, start ve směru CCW je zadán funkcí M4. Po odstartování pohybu těmito funkcemi trvá otáčení tak dlouho, dokud není programována funkce M5 (STOP vřetene) nebo M19 (STOP vřetene v orientovaném bodě).

Příklad:

N10	S55	M3 M41	"	55ot/min, směr CW, první převodový stupeň
N20	M4		"	reverzace otáček na směr CCW

N20 M5 " stop otáček

8.3 Chlazení nástroje

Pro ovládání chlazení jsou v systému vyčleněny 2 skupiny M funkcí 5 a 6, které mohou ovládat 4 samostatné chladicí okruhy. Čísla M funkcí jsou vedena v souhrnném přehledu adres návrháře PLC.

8.4 Přerušení a konec partprogramu

V systému lze programovat čtyři M-funkce (M00,M01,M02,M30), kterými lze ukončit nebo přerušit zpracováváný partprogram:

M0	Nepodmíněný stop Zpracováváný partprogram program se přeruší po vykonání všech operací v bloku. Jaké funkce se provedou musí být součástí návodu PLC
M1	Podmíněný stop Systém se zachová shodně jako při M00 v případě, že pracuje v modifikaci M01 (možnosti jízdy)
M2	Konec partprogramu
M30	Konec partprogramu.

Pozn.:

Tvůrce rozhraní pro konkrétní stroj může chování těchto funkcí do jisté míry modifikovat.

8.5 Upnutí a uvolnění obrobku

Funkce jsou v režii PLC.

8.6 Výměna nástroje a obrobku

Funkce jsou v režii PLC.

9

9. PEVNÉ CYKLY PRO SOUSTRUHY

Pro dílenské programování se mohou využívat vestavěné pevné cykly (tlačítko „Tvary“).

V revizi 550 jsou k dispozici následující pevné cykly:

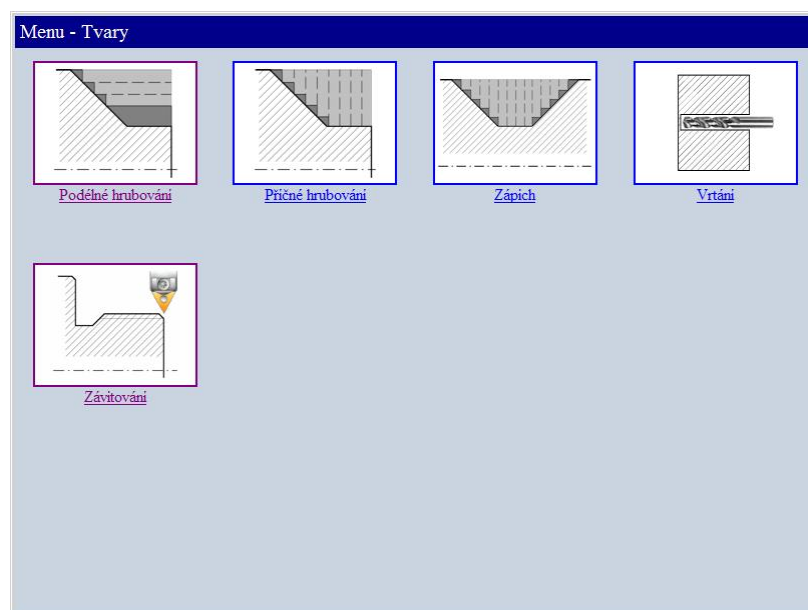
- Hrubování podélné
- Hrubování příčné
- Zápich
- Vrtání
- Závítování

Pevné cykly se mohou používat samostatně, t.j. odjedou se po vyplnění parametrů, potvrzení tlačítkem O.K a stiskem tlačítka START.

Častěji se pevné cykly používají tak, že se z jednotlivých cyklů sestavuje klasický partprogram. Vytvořený cyklus je nutné (stav k revizi 552) vložit do souboru partprogramu v editoru ze schránky (clipboardu), kam se zkopíruje v editoru pomocí menu „Kopírovat“. V připravovaných verzích se budou cykly vkládat do editoru automaticky.

Podrobný postup je uveden dále.

Do menu pevných cyklů – tvarů - se dostaneme stiskem tlačítka „Ruční“, které je obvykle v základním menu, a dále stiskem tlačítka „Tvary“.



Objeví se okno (viz. obr) s nabídkou cyklů, které jsou k dispozici.

Požadovaný cyklus se vybere dotykem příslušného obrázku.

Následně se obecně může objevit další okno, které blíže specifikuje zvolený cyklus. Např. u vrtání to může být nabídka různých druhů vrtání: normální vrtání, vrtání s časovou prodlevou, vrtání s výplachem apod. Pokud není potřeba dále specifikovat typ pevného cyklu objeví se již okno pro zadávání vstupních parametrů.

V dalším textu jsou jednotlivé pevné cykly popsány podrobně.

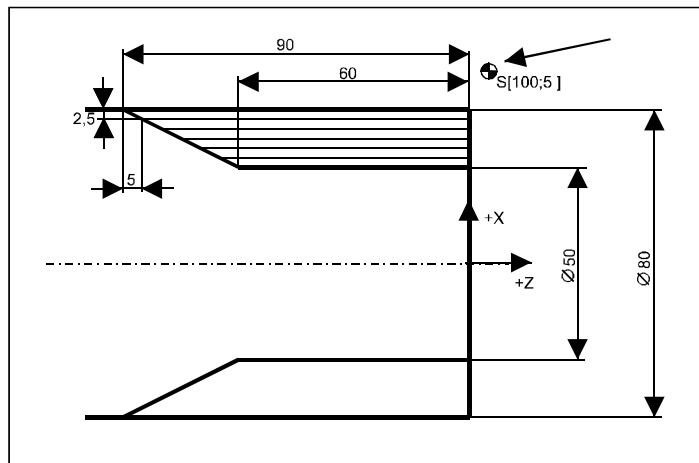
Postup, jakým způsobem vložit konkrétní pevný cyklus do vytvářené partprogramu je popsán podrobně u prvního cyklu, t.j.

podélného hrubování. Postup uvedený u podélného hrubování se používá i u ostatních cyklů – principy jsou podobné.

podélného hrubování. Postup uvedený u podélného hrubování se používá i u ostatních cyklů – principy jsou podobné.

9.1 Podélné hrubování

Předpokládejme vyhrubování polotovaru z tyče o průměru 80mm (viz obr.). Hrubovat budeme z průměru 80mm na průměr 50mm. Celková délka hrubování bude 90mm, Zkosení začíná na 60 mm a je dáno v tomto případě délkou (5mm) na šířku jedné třísky (2,5mm). Zkosení lze zadávat také úhlem a délkou zkosení, případně i jinými způsoby.



Hrubování začíná v bodě S, do kterého se najíždí rychloposuvem.

Výchozí bod musí být zadán o bezpečnou vzdálenost mimo materiál, ale neměl by být zbytečně daleko, neboť z výchozího bodu se jede již pracovním posuvem.

X-ová souřadnice bodu S se zadává obvykle průměrově, jak je u soustruhů zvykem, je možné ji ale zadat případně i poloměrově.

Typ programování v ose

X (průměrově/poloměrově) je dán nastavením systémového parametru DIAMETERPROG

a DIAMETERINCPROGR. Tyto parametry jsou zadány v prioritním bloku (možno zkontrolovat např. při centrální anulaci). Pokud by nebyly uvedeny, programovalo by se poloměrově !

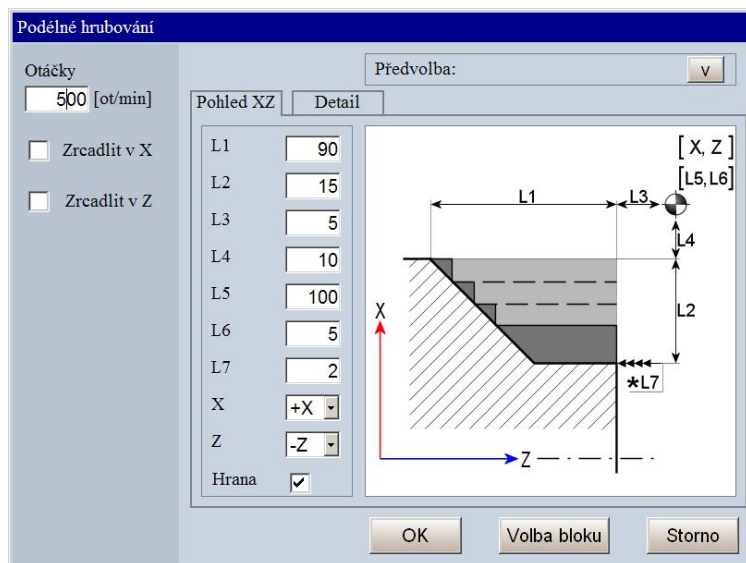
DIAMETERPROG = 1 (absolutní míry v X jsou programovány průměrově)

DIAMETERINCPROGR = 1 (přírůstkové míry v X jsou programovány průměrově)

Pozn.: Místo jedničky může být uvedeno TRUE

V dialogové okně může být obecně několik záložek, pro podélné hrubování je to záložka „PohledXZ“ a záložka „Detail“.

Záložka Pohled XZ



Parametry pro uvedený příklad by se v záložce „Pohled XZ“ vyplnily následovně:

L1 – Celková délka hrubování v ose Z, včetně eventuálního zkosení.

L2 – Hloubka hrubování. Uvádí se míra tak, jak je kótovaná na obrázku parametrem L2. Není to tedy rozdíl průměrů 80 a 50!

L3 – Bezpečná vzdálenost v ose Z. Je to vzdálenost startovacího bodu S od materiálu v ose Z.

Tuto vzdálenost jede cyklus již pracovní rychlostí, proto by neměla být zbytečně daleko.

L4 – Bezpečná vzdálenost v ose X. Je to vzdálenost startovacího bodu S od materiálu v ose X.

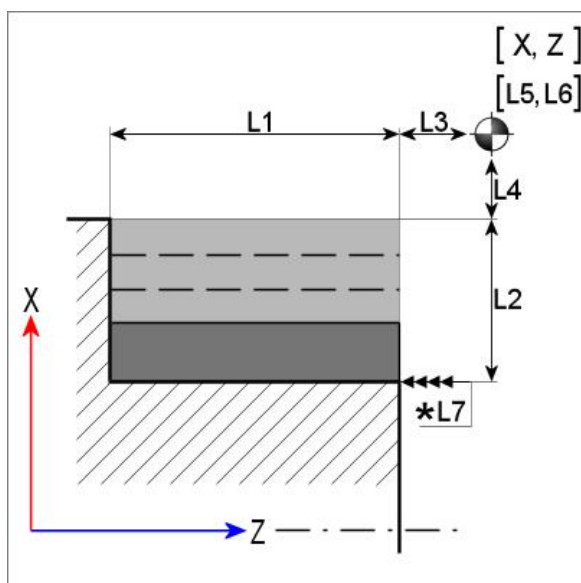
Míra se opět zadává tak, jak je kótovaná na obrázku (není to rozdíl průměrů!)

L5, L6 – Absolutní souřadnice startovacího bodu hrubovacího cyklu. L5 je hodnota osy X, L6 je hodnota osy Z. Souřadnice X se zadává průměrově! (samozřejmě pouze v případě, že se používá průměrové programování)
 L7 – počet začišťovacích průchodů. Pokud je zadána 0, začišťovací průchody se neprovedou, tj. na eventuelní šikmé ploše zůstanou neobrobené „schody“. Vodorovná část (rovnoběžná s osou Z) ale bude obrobena na zadanou míru (podle L2). Začišťovací průchod tedy L7-krát objíždí konečný tvar hrubování.

Parametry X a Z určují eventuelní změnu znaménka v příslušné ose - zrcadlení. V běžných případech je X kladné a Z záporné. Praktické využití má u soustruhů a karuselů pouze změna znaménka u X pro případ, že by se hrubování provádělo v záporných souřadnicích.

Prakticky stejný význam má zaškrtnutí políčka „Zrcadlit v X“ resp. „Zrcadlit v Z“. Rozdíl je pouze v použitých prostředcích. Zaškrtnutím políčka „Zrcadlit v ...“ se zrcadlení provádí pomocí systémové funkce MIRROR.

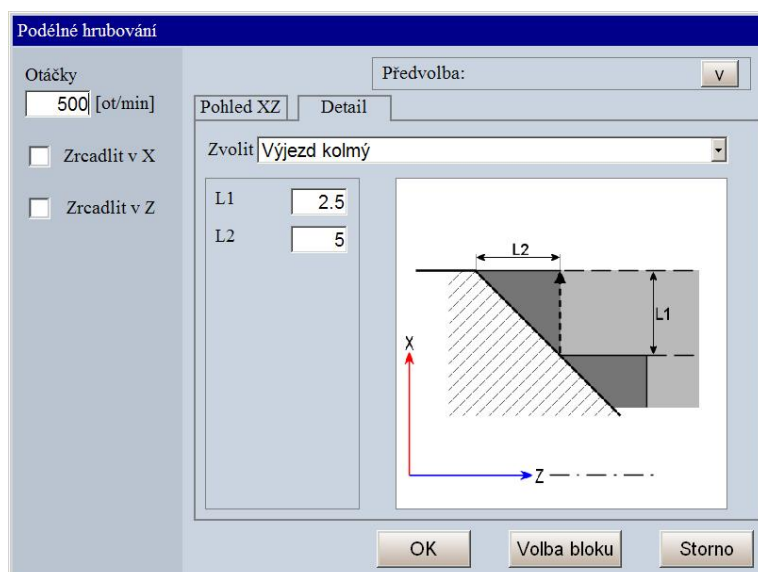
V dialogu se dále volí požadované otáčky pro hrubování a rychlost.



Hrana - zaškrťovací políčko specifikuje blíže hrubovací cyklus. Je-li zaškrtnuté, má hrubovací cyklus šikmý výjezd, jak je použito v našem příkladu. Není-li zaškrtnuté, nemá hrubovací cyklus šikmý výjezd, jedná se pouze o válcové vyhrubování (viz obr.).

Záložka Detail

Parametry v této záložce umožňují zadat hloubku třísky a eventuelní sklon. Jiný způsob zadání je zadat úhel šikmého výjezdu a šířku třísky. U prvního způsobu je nutné z úhlu spočítat parametry L1 a L2.



V příkladu je použita hloubka třísky 2.5mm (L1) a parametr L2=5mm. (úhel sklonu vychází 26.565°).

Většinou je znám úhel a tloušťka třísky, tj. např. pro úhel 25° a tloušťku třísky 2.5mm by se pro tento způsob zadání vypočetla hodnota L2=5.361.

Kromě parametrů, které určují úhel se volí parametry výjezdu. Je možno zvolit tyto typy výjezdu:

- Výjezd kolmý
- Výjezd zadaný úhlem
- Výjezd zadaný vzdáleností

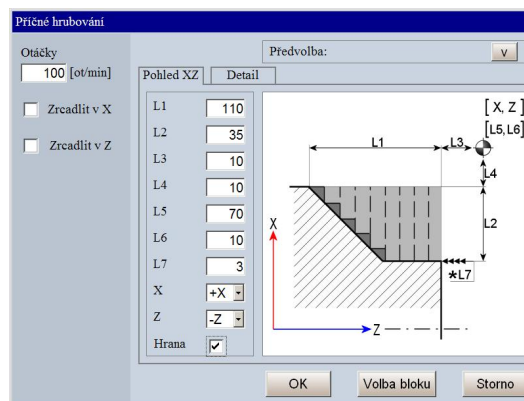
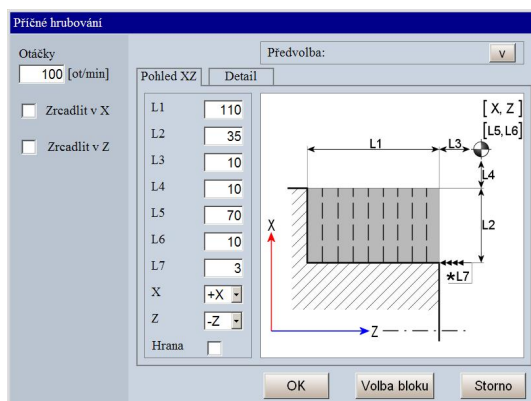
Pokud je hrubování se šikmou hranou, volí, resp. může se volit výjezd kolmý. U hrubování bez šikmé hrany se volí obvykle šikmý výjezd, který můžeme zadat většinou úhlem, případně vzdáleností L3. Šikmý výjezd se zadává toho důvodu, aby nůž nejezdil opakovaně po již obrobené ploše.

Předvolba

Systém umožňuje uložení jednou nastavených parametrů do paměti tak, aby bylo možné je kdykoli znovu vyvolat a tím si ušetřit práci s jejich opětovným vyplňováním. Je možné si vytvořit hrubovací cykly pro různé průměry a uložit si je do paměti, například pod názvem „Hrubování80“. Může to např. znamenat, že se bude hrubovat z tyčoviny průměru 80mm. Budeme mít přeplněny všechny parametry, přičemž se bude měnit např. pouze délka hrubování. Praktičtější případ úschovy parametrů bude spíše u závitovacích cyklů (viz dále), kde si můžeme přednastavit parametry všech potřebných závitů. Tím se podstatně zjednoduší zadávání, neboť u všech cyklů je nutné nastavit poměrně dost parametrů, které se ale většinou nemění – zůstávají stále stejné. Mimo úschovy parametrů do paměti se při volbě cyklů vždy předvolí naposledy zadané parametry, takže i pokud nepoužijeme předvolbu, zůstanou parametry nastavené.

9.2 Příčné hrubování

Příčné hrubování je velmi podobné podélnému hrubování, jediný rozdíl je, že se hrubuje ve směru osy X. Příčné hrubování může být i se šikmou stěnou. Použité parametry vyplývají z obrázku a nebudeme je zde podrobněji popisovat.



Příčné
é

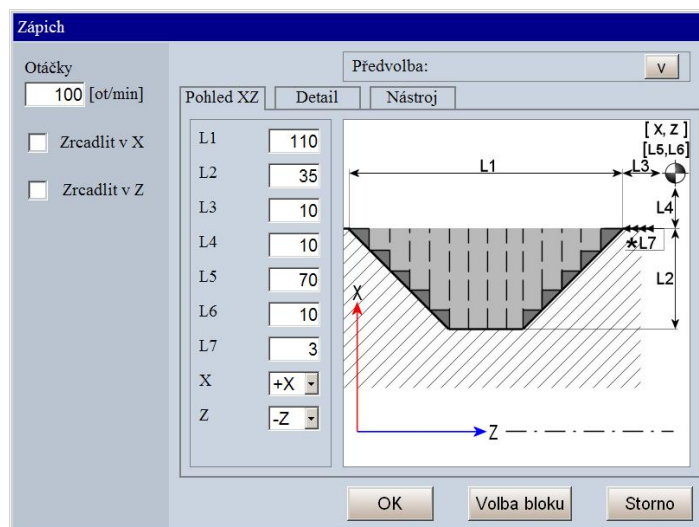
hrubování bez šikmé stěny a se šikmou stěnou – ovlivňuje to zaškrtnutí políčka „Hrana“. Ostatní rozměrové parametry se zadají podle obrázku.

V záložce „Detail“ se volí typy výjezdů:

- Výjezd kolmý
- Výjezd zadaný úhlem
- Výjezd zadaný vzdáleností

9.3 Zápich

Cyklus zápichu může mít rovné nebo šikmé stěny (určuje se to v záložce „Detail“)



L1- Celková délka zápichu

L2 – Hloubka zápichu

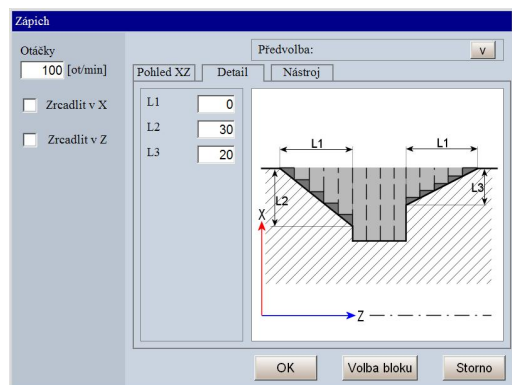
L3 – Bezpečná vzdálenost v ose Z. Je to vzdálenost startovacího bodu S od materiálu v ose Z.

L4 – Bezpečná vzdálenost v ose X. Je to vzdálenost startovacího bodu S od materiálu v ose X.

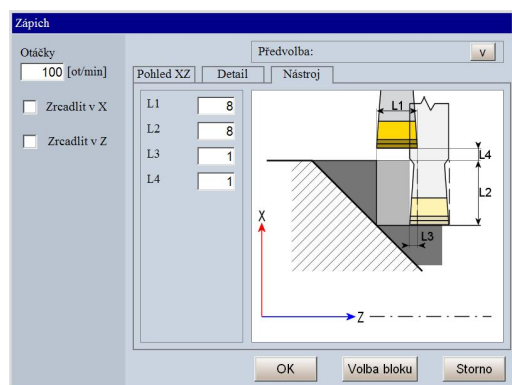
L5, L6 – Absolutní souřadnice startovacího bodu hrubovacího cyklu. L5 je hodnota osy X, L6 je hodnota osy Z. Při průměrovém programování se souřadnice X zadává průměrově!

L7 – Počet začišťovacích průchodů

Pozn. jednou nastavené parametry X a Z, kde se určují směry, se pro daný stroj nemění. Možnosti zrcadlení se u běžných soustruhů obvykle nevyužívají.



V záložce „Detail“ se udávají rozměry, pokud se jedná o nestandardní zápich, t.j. zápich se šikmými stěnami. Zadávané parametry jsou patrné z obrázku:



V záložce „Nástroj“ se zadávají tyto parametry:

L1 - Šířka zapichovacího nástroje

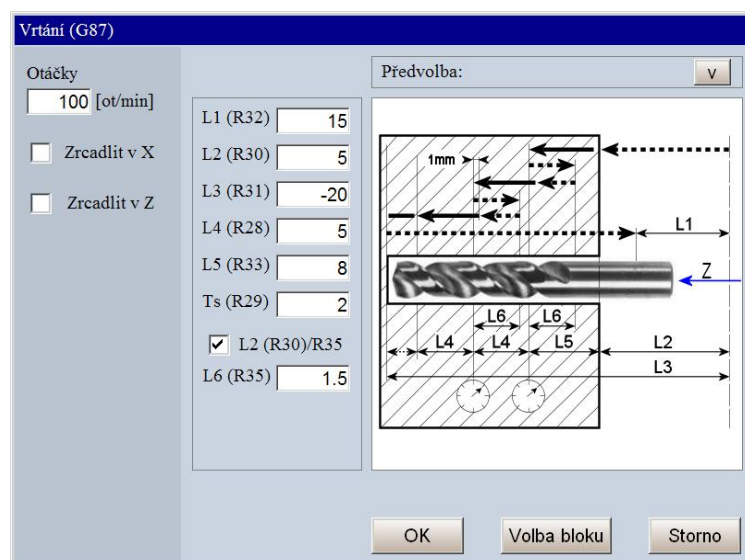
L2 - Hloubka jedné třísky

L3 - Přesah třísky

L4 – Bezpečná vzdálenost nad zápichem

9.4 Vrtání

Vrtací cyklus s výplachem. Zadávají se následující parametry:



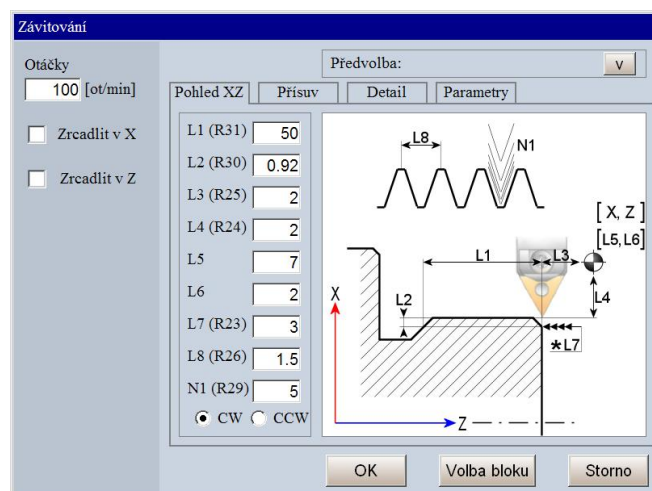
- L1 – Kam se vyjede po vrtání
- L2 – Odkud vrtat (absolutně)
- L3 – Kam vrtat (absolutně)
- L4 – Přírůstek hloubky
- L5 – První hloubka
- L6 – Kolik povyjet (výplach)

Parametr L6 se vyplní pouze, je-li zaškrtnuto zaškrťovací políčko. Pokud není zaškrtnuté, provádí se výplach až ven z díry. Po výplachu se vrták vrací rychloposuvem do díry, 1 mm před poslední vyvrtanou hloubku a pak vrtá dál pracovním posuvem.

9.5 Závítování vnější

Parametry pro závítování se zadávají ve čtyřech záložkách. Pro většinu praktických případů stačí nastavit pouze záložku „Pohled XZ“. Parametry v ostatních záložkách zůstávají většinou stejné.

Záložka Pohled XZ

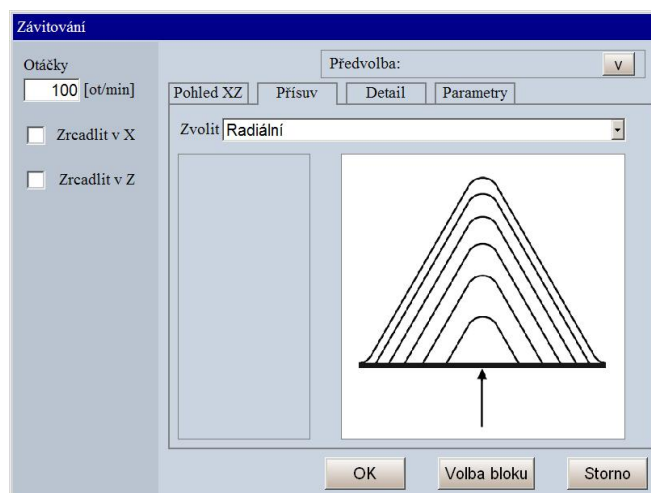


- L1 – celková délka závitu
- L2 – hloubka závitu
- L3 – bezpečná vzdálenost startovacího bodu od materiálu v ose X
- L4 – bezpečná vzdálenost startovacího bodu od materiálu v ose Z
- L5 – absolutní hodnota startovacího bodu závitu v ose X
- L6 – absolutní hodnota startovacího bodu závitu v ose Z
- L7 – počet začišťovacích průchodů (obrábí se na stejném průměru)
- L8 – stoupání závitu
- N1 – počet třísek, kterými má být závit vyříznut
- CW, CCW – směr otáčení vřetena (ve směru hodinových ručiček a proti směru hodinových ručiček)

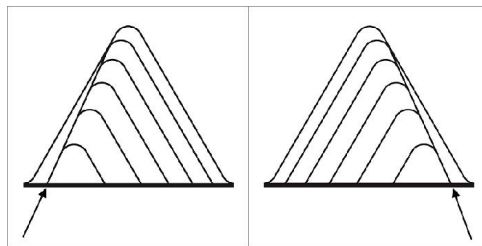
Otáčky – otáčky vřetena

Pozn.: zaškrťovací políčko „Zrcadlit“ se u závítování prakticky nepoužívá

Záložka Přísuv

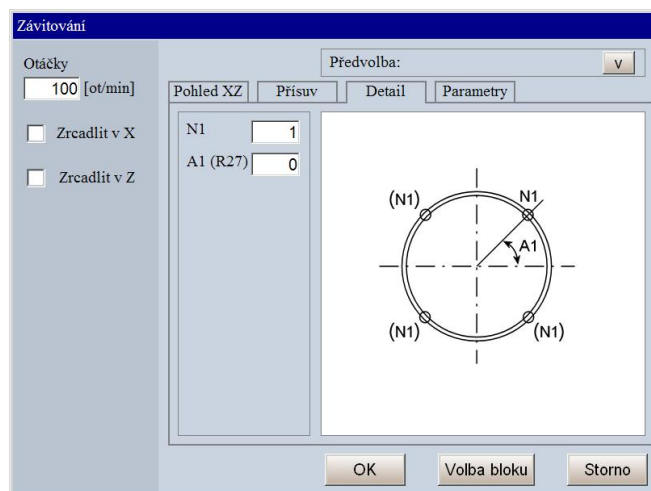


V záložce se nastavuje typ přísluvu:



K dispozici je přísluv radiální, pravý boční a levý boční. Přísluv se volí v políčku „Zvolit“

Záložka Detail



Záložka „Detail“ se používá pro vícechodé závity. Pro jednochodé závity se oba parametry (N1 a A1) nastavují na nulu.

Pro nejběžnější, t.j. jednochodé závity je $N1 = 1$

Pro dvouchodé závity se nastaví $N1 = 2$

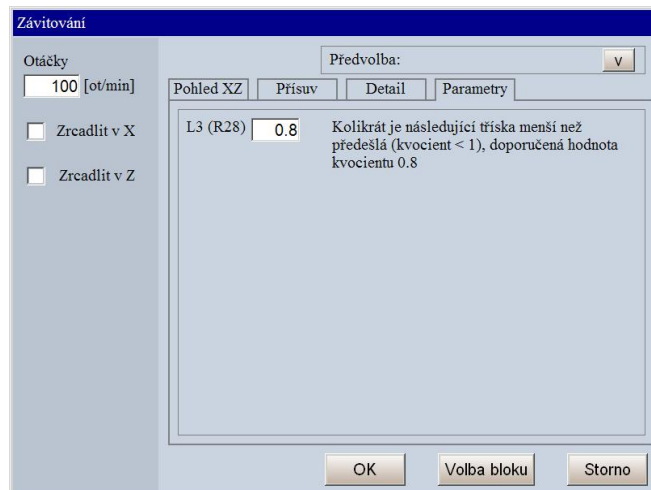
Pro tříchodé závity se nastaví $N1 = 3$

atd.

A1 – úhel počátku závitu, t.j. úhel, o který se pootočí vřeteno od nulového pulzu než začne přísluv v ose Z. U běžných závitů není nutné úhel zadávat.

Zadal by se, pokud by bylo z nějakých důvodů nutné začít závit v přesné poloze vzhledem k obrobku (např. mazací drážky s velkým stoupáním apod.)

Záložka Parametry



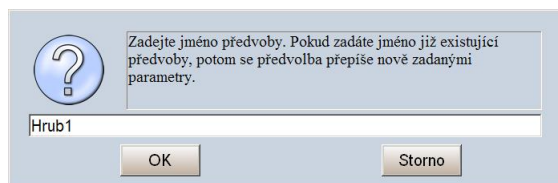
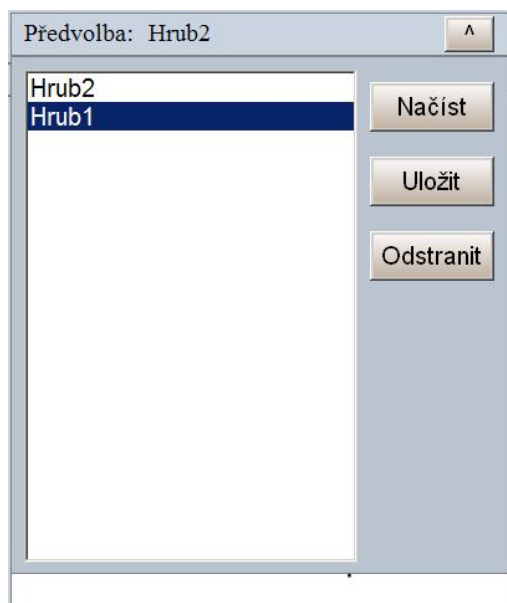
V této záložce se zadává pouze koeficient, kolikrát je následující tříska menší než předešlá. Při přísluvu do závitu v ose X se tříska zmenšuje z toho důvodu, aby průřez odebíraného materiálu zůstal stejný. Doporučená hodnota je 0.8 a není důvod ji měnit.

9.6 Předvolba

Každý pevný tvar má k dispozici tzv. „Předvolbu“. Pomocí předvolby je možné uschovat (do souboru) a kdykoli později načíst již jednou použité parametry. Ušetří se tím jejich opětovné zadávání

Stiskem tlačítka  v poli „Předvolba“ se objeví (a opětovným stiskem zmizí) dialog pro načtení, uložení a případně odstranění předvolby.

Pokud jsou již předvolby k dispozici, objeví se jejich seznam. Na obrázku jsou to předvolby Hrub1 a Hrub2. Zvolíme požadovanou a stiskneme tlačítko „Načíst“. Dialogové okno zmizí a parametry se objeví v zadávacích okénkách.



Pokud naopak chceme parametry uschovat, stiskneme tlačítko „Uložit“ a do dialogového okénka zadáme název předvolby, např. „Hrub1“ (viz obr) a potvrdíme O.K. Pokud chceme předvolbu odstranit, nastavíme na ni kurzor (modré zvýraznění) a stiskneme „Odstranit“.

Pozn.:

Parametry jsou uschovány v textových souborech v adresáři CNC User Files\NCP\MACRO a dále v podadresáři konkrétního tvaru. Z hlediska uživatele to ale není důležité.

9.7 Vložení pevného cyklu do partprogramu

Pokud vytváříme partprogram ručně, nebo potřebujeme do již hotového partprogramu vložit libovolný pevný cyklus, postupujeme následujícím způsobem:

- Zvolíme si požadovaný cyklus (menu „Ruční“ a „Tvary“) a v dialogu vyplníme potřebné parametry.
- Stiskneme tlačítko OK
- Tím je jednotlivý cyklus připraven k vykonání – to ale v tomto případě nepotřebujeme. V levém okně se ale objeví jeho rozvoj, tj. výpis, který potřebujeme vložit do našeho partprogramu.
- Označíme se celý výpis (měl by mít modrý podklad) tímto způsobem: vynoříme se z menu tak abychom mohli zvolit „Editor“. Tím se pevný cyklus dostane do editoru. Pomocí menu (pravé svislé) Editace – Blokové operace zvolíme „Režim označování“. Pokud nyní stiskneme kurzorovou šipku dolů, začnou se postupně jednotlivé řádky označovat (zmodrají). Dojedeme až na konec souboru (za blok N ENDPGRAM) a stiskneme tlačítko „Kopírovat“. Tím se dostane označený blok do schránky (clipboard). Nyní pomocí tlačítka „Otevřít“ (spodní menu) otevřeme požadovaný soubor, případně zvolme nový, prázdný soubor. Opět zvolíme (pravé svislé menu) Editace – Blokové operace a nyní zvolíme tlačítko „Vložit“. Obsah schránky, tj. náš pevný cyklus, se vloží do editoru na místo, kde je kurzor. Před vložením je tedy nutné si kurzor správně umístit. Obsah editoru nyní můžeme „Uložit“ nebo „Uložit jako...“. Tím máme pevný cyklus včetně nastavených parametrů vložený do požadovaného souboru.
- Protože po volbě pevného cyklu stiskem tlačítka OK byl cyklus připraven k vykonání, svítí kontrolka „Systém připraven“. Zrušíme ji navolením a odstartováním centrální anulace.
- Můžeme vkládat stejným způsobem další pevný cyklus.

10

10. RUP – ruční předvolba bloku

Ruční předvolba bloku slouží k zadání a provedení jednoho bloku v automatickém režimu. Po zadání požadovaných hodnot a po stisku tlačítka START se blok vykoná. Zadávat se může jak pohybový blok, tak blok s technologií nebo kombinaci obou najednou. Ruční předvolba se používá např. pro nájezd do přesné polohy, pro roztočení vřetene, změnu otáček vřetene a další podobné technologické i systémové funkce.

Jednotlivá zadávací políčka mají tento význam:

Volba posuvu – volí se jedna ze dvou možností – rychloposuv G00 nebo lineární interpolace G01. Při volbě lineární interpolace je nutné v jednom z následujících políček zadat i rychlost (F).

Zadání souřadnic – volí se jedna ze dvou možností, t.j. absolutní programování (G90) nebo přírůstkové programování (G91)

Osa X, Osa Z – pokud se políčko zaškrtně, provede se posuv na míry, zadané v okénkách pod nimi. I když bude zadaná nějaká hodnota, posuv se neprovede, pokud nebude políčko zaškrtnuté.

Rychlost (F) – musí být zadaná nenulová hodnota, pokud je ve „Volbě posuvu“ zvolena lineární interpolace

Otáčky – pokud chceme roztočit vřeteno, zadají se požadované otáčky (a zvolíme dále směr točení).

Technologie – políčko pro zadání dalších M-funkcí, které nejsou explicitně uvedeny dále (např. M06, T12 atd.). Je možné zde zadat i G-funkce (např. funkce pro počátky G53, G54, atd.). Dále je možné zadat transformační funkce (např. TRANSLATE(x,y,z)). Je třeba si uvědomit, zda zadaná funkce má trvalou platnost (t.j. platila by eventuálně i v následně zvoleném partprogramu), nebo zda ji přepíše funkce v prioritním bloku. V běžných případech se ale RUP pro zadávání funkcí platných i následně nepoužívá.

Vřeteno - pomocí radiobutonů se volí jedna z možností M3, M4 a M5, t.j. směr točení vřeteno nebo stop vřetene. Předpokládá se rovněž zadání otáček v políčku „Otáčky“.

Chlazení - pomocí radiobutonů se volí zapnutí nebo vypnutí chlazení (funkce musí být příslušným způsobem implementovány v PLC).

OK – potvrzení zadaných hodnot, ale blok se vykoná až po stisknutí tlačítka (zeleného) START na panelu obsluhy.

Start - okamžité vykonání zadaného bloku.

Storno – zadaný blok se nevykoná a okno RUP se zavře (zmizí)

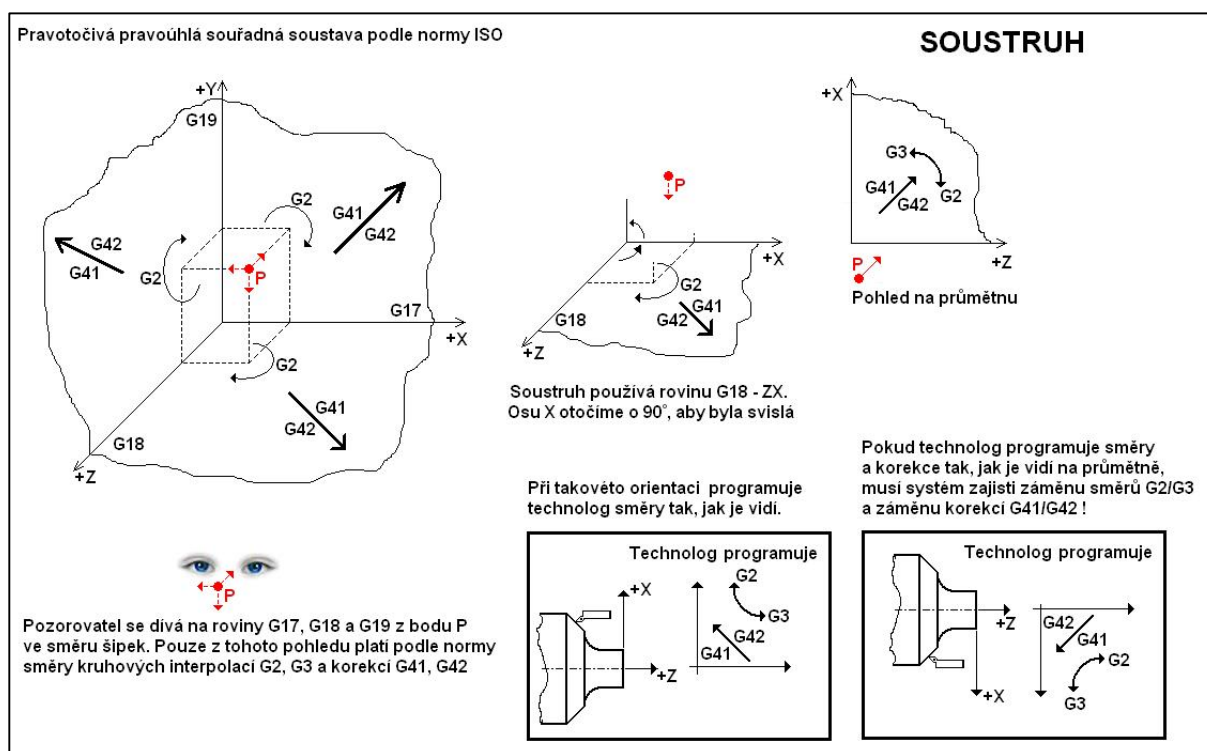
Pozn.:

Zadané hodnoty ve všech políčkách zůstanou zapamatované i při příštím otevření dialogu RUP (pokud se nedá Storno). Z bezpečnostních důvodů se ale zruší požadavek na pohyb (zruší případné zaškrtnutí) a radiobuttony se nastaví na Stop vřetene (M5) a Vypnutí chlazení (M9)

11

11. GRAFIKA – souřadný systém

Systémy CNC872 pracují (z hlediska programování) v pravoúhlém souřadném systému. V závislosti na orientaci os u konkrétních strojů (soustruh, karusel, orientace os) je třeba počítat se správným programováním korekce a směru kružnic. Problematika je patrná z následujícího obrázku



Protože obvykle technolog programuje směry tak, jak je „vidí na papíru“, doporučuje se místo programování G2 a G3 programovat CW a CCW:

CW – pohyb ve směru hodinových ručiček

CCW - pohyb proti směru hodinových ručiček

Systém podle skutečné orientace na stroji nahradí tato makra (CW a CCW) správnými funkcemi G2 a G3.

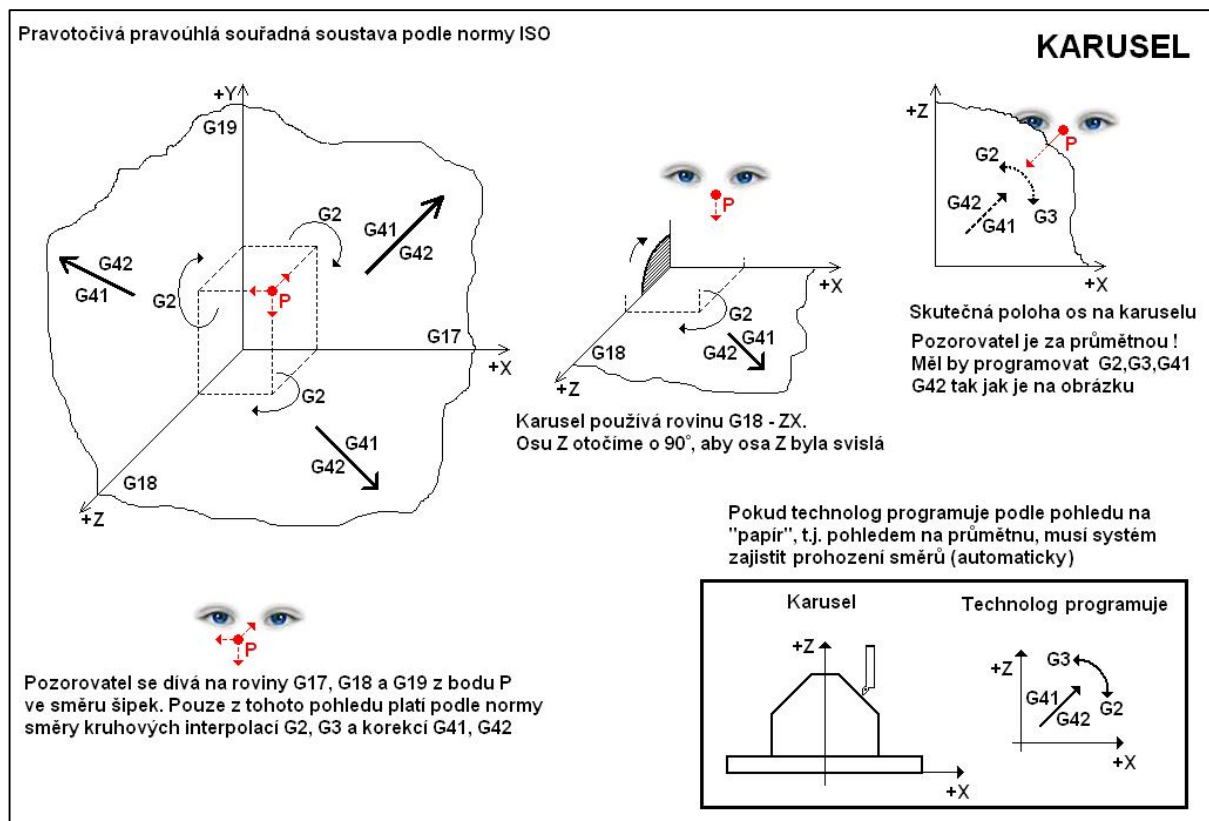
Podobně pro programování poloměrových korekcí se používá makro:

Tool(Nastroj,Korekce)

kde:

Nastroj - číslo nástroje

Korekce – hodnota 40 (bez korekce), 41 (korekce vlevo), 42 (korekce vpravo)



OBSAH

1.	Stavba partprogramu	1-2
1.1	Bloky, číslování bloků, začátek a konec partprogramu	1-2
1.2	Komentáře v partprogramu	1-3
1.3	Poznámky k bloku partprogramu	1-3
1.4	Podprogramy, makrocikly a skoky	1-4
1.4.1	Podprogramy	1-4
1.4.2	Makrocikly	1-4
1.4.3	Skoky v partprogramu	1-5
1.4.4	MAKRA	1-6
2.	Seznam programovatelných adres	2-8
2.1	Programovatelné adresy	2-8
2.2	G a M funkce	2-9
3.	Programování	3-12
3.1	Stavění souřadnic - funkce G0	3-12
3.2	Lineární interpolace - funkce G1	3-12
3.2.1	Zkosení rohu – G10	3-13
3.3	Kruhová interpolace - funkce G2, G3	3-13
3.3.1	Možnosti zadání kruhové interpolace	3-13
3.3.2	Zadání kružnice koncovým bodem a středem	3-14
3.3.3	Zadání kružnice koncovým bodem a poloměrem	3-15
3.3.4	Zaoblení rohu se zadaným poloměrem - G11	3-16
3.3.5	Kruhová interpolace s tečným napojením	3-17
3.4	Závitování	3-17
4.	DÉLKOVÉ A POLOMĚROVÉ KOREKCE	4-21
4.1	Nastavování délkových a poloměrových korekcí	4-21
4.2	Nastavení korekcí v tabulce nástrojů	4-21
4.3	Nastavení na stroji „škrtnutím“ o obrobek	4-22
4.4	Změna hodnoty délkové korekce	4-23
4.5	Programování korekcí v partprogramu	4-24
5.	Funkce	5-26
5.1	Matematické a rozhodovací funkce	5-26
5.2	Funkce pro oznamování chyb, upozornění a informací	5-27
6.	Posunutí počátku	6-31
6.1	Tabulka posunutí	6-31
7.	ZADÁNÍ POSUVU	7-33
7.1	Konstantní řezná rychlost	7-34
7.2	Ovládání z partprogramu	7-34
8.	TECHNOLOGICKÉ FUNKCE	8-35
8.1	Časová prodleva	8-35
8.2	Otáčky vřetene	8-35
8.3	Chlazení nástroje	8-36
8.4	Přerušení a konec partprogramu	8-36
8.5	Upnutí a uvolnění obrobku	8-36
8.6	Výměna nástroje a obrobku	8-36
9.	PEVNÉ CYKLY PRO SOUSTRUHY	9-37
9.1	Podélné hrubování	9-38
9.2	Příčné hrubování	9-40
9.3	Zápich	9-41
9.4	Vrtání	9-42
9.5	Závitování vnější	9-42
9.6	Předvolba	9-44
9.7	Vložení pevného cyklu do partprogramu	9-44
10.	RUP – ruční předvolba bloku	10-45
11.	GRAFIKA – souřadný systém	11-47

