

Rexroth IndraMotion MTX

Příručka programování

R911311168
Vydání 01

Popis aplikace



Název	Rexroth IndraMotion MTX Příručka programování
Druh dokumentace	Popis aplikace
Typ-dokumentace	DOK-MTX***-NC**PRO*V02-AW01-CS-P
Interní poznámka	Číslo dokumentace 120-2500-B301-01/DE
Účel dokumentace	Tato příručka informuje o: <ul style="list-style-type: none"> • standardní-programování IndraMotion MTX-řídící jednotky.

Průběh změn

Dosavadní vydání	Stav	Pozn.
DOK-MTX***-NC**PRO*V02-AW01-CS-P	09.2004	Platí pro 02VRS

Ochranná doložka © Bosch Rexroth AG, 2004

Předávání stejně jako reprodukce těchto podkladů, zhodnocování či sdělování jejich obsahu je nepřipustné, pokud není vysloveně povoleno. Překročení tohoto zákazu zavazuje k náhradě škod. Všechna práva pro případ udělení patentu nebo registrace užitého vzoru vyhrazena. (DIN 34-1)

Závaznost Uvedené informace slouží pouze k popsání výrobku a nesmí být chápány jako záruka jeho vlastností v právním smyslu. Změny v obsahu dokumentace a dodacích možnostech výrobku jsou vyhrazeny.

Vydavatel Bosch Rexroth AG
Postfach 11 62
D-64701 Erbach
Berliner Straße 25
D-64711 Erbach, Germany
Tel.: +49 (0) 60 62/78-0
Fax: +49 (0) 60 62/78-4 28
Odd.: BRC/ESM11 (WE)

Obsah

Obsah

Strana

1	Bezpečnostní pokyny	1-1
1.1	Použití k určenému účelu	1-1
1.2	Kvalifikovaný personál	1-3
1.3	Bezpečnostní pokyny pro výrobky	1-4
1.4	Bezpečnostní pokyny v této příručce	1-5
1.5	Bezpečnostní pokyny pro popsany výrobek	1-6
2	Základy NC-programování	2-1
2.1	Základní informace o standardním programování a programování CPL	2-2
2.2	Spojování NC-programů	2-5
2.3	Základní součásti NC-programu	2-7
2.4	Působení programových slov	2-18
2.5	Speciální prvky pro vytváření programů	2-20
2.6	Podprogramy	2-25
2.7	Programování návěští a instrukce skoku	2-32
2.8	Rozhodovací a rozvětvovací instrukce	2-38
2.9	Opakovací instrukce	2-42
2.10	Programování proměnných	2-45
2.11	Instrukce CPL	2-59
2.12	Další základní prvky CPL	2-64
2.13	Příkazy pro synchronizaci NC-vět	2-67
3	NC-funkce se syntaxí podle DIN 66025 (vč. rozšíření)	3-1
3.1	Přehled	3-1
3.2	G-kódy	3-3
3.3	M-kódy	3-87
3.4	Programování posuvu a počtu otáček	3-97
3.5	Oprava nástroje	3-101
4	NC-funkce s jazykovou syntaxí	4-1
	Přehled všech G-instrukcí a funkcí naleznete v příloze.	
5	Funkce CPL	5-1
5.1	Hodnoty souřadnic a os	5-1
5.2	Posunutí nulových bodů	5-19
5.3	Opravy nástroje	5-21
5.4	Databáze nástrojů	5-23
5.5	Umístění (šikmá rovina)	5-26
5.6	Všeobecný přístup do tabulek XML	5-29
5.7	Měřítka	5-30
5.8	Systémová data	5-31

Obsah

5.9	Variabilní adresa osy	5-44
5.10	Rozhraní SPS	5-45
5.11	Zaznamenávání času	5-47
5.12	Chyby a kategorie chyb	5-48
5.13	Připojení NCS	5-51
5.14	Zpracování znakových řetězců	5-94
5.15	Zpracování souborů	5-111
5.16	Komunikace	5-136
A	Příloha	A-1
A.1	Zkratky	A-1
A.2	Tabulkové přehledy NC-funkcí	A-2
A.3	Sada znaků ASCII	A-58
A.4	Přídavné kódy kláves	A-58
A.5	Rejstřík	A-59

Bezpečnostní pokyny

1 Bezpečnostní pokyny

- ★ Přečtěte si tuto příručku předtím, než začnete programovat IndraMotion MTX nebo měnit stávající programy!
- ★ Uchovávejte tuto příručku na místě vždy přístupném pro všechny uživatele!


1.1 Použití k určenému účelu


Tato příručka obsahuje údaje o používání k určenému účelu. Z důvodů přehlednosti však neobsahuje všechny podrobnosti o všech možných kombinacích funkcí. Rovněž není možné vzít v úvahu každý myslitelný případ integrace nebo provozu.

IndraMotion MTX-Řídicí jednotka slouží k

- programování obrysů a technologie obrábění (posuv, počet otáček včetně, výměna nástrojů) obrobku.
- vedení obráběcího nástroje podél naprogramované dráhy.
Posuvové pohony, včetně a pomocné osy obráběcího stroje jsou řízeny pomocí rozhraní SERCOS.

Jakékoli použití, které sahá nad tento rámeček, není použití k určenému účelu!

 **Kromě toho jsou zapotřebí vstupní/výstupní komponenty pro integrovanou jednotku, která - v komunikaci s vlastní jednotkou - kompletně řídí obráběcí proces na stroji a také ho monitoruje z bezpečnostně technického hlediska.**

 **Bezchybný a bezpečný provoz výrobku předpokládá správný transport, správné skladování, instalaci a montáž stejně jako pečlivou obsluhu.**

Popsané výrobky

- byly vyvinuty, vyrobeny, zkontrolovány a zdokumentovány při dodržování bezpečnostních norem. Při dodržování předpisů, platných pro projektování, montáž a používání k určenému účelu, a bezpečnostně technických pokynů nepředstavuje výrobek v normálním případě žádné nebezpečí pro lidi nebo věci.
- splňují požadavky
 - směrnic EMV (89/336/EEC, 93/68/EEC a 93/44/EEC)
 - nízkonapěťové směrnice (73/23/EEC)
 - harmonizovaných norem EN 50081-2 a EN 50082-2
- jsou určeny pro provoz v průmyslovém prostředí, tzn.
 - bez přímého připojení k veřejnému nízkonapěťovému elektrickému napájení,
 - připojení pomocí transformátoru ke středně-, resp. vysokonapěťové síti.

Bezpečnostní pokyny

V obytných oblastech, v obchodních a komerčních prostorech stejně jako v malých podnicích smí být přístroje třídy -A- používány s ohledem na následující pokyn:

-  **Toto je zařízení třídy A. Toto zařízení může v obytných oblastech způsobit rádiové rušení. V takovém případě může být nezbytné, aby provozovatel provedl přiměřená opatření pro odstranění tohoto rušení.**

Bezpečnostní pokyny

1.2 Kvalifikovaný personál

Požadavky na kvalifikovaný personál vyplývají z profilů požadavků popsaných v ZVEI a VDMA, viz:

Další vzdělávání v automatizační technice

Vydal: ZVEI a VDMA

MaschinenbauVerlag

Postfach 71 08 64

60498 Frankfurt

Tato příručka je určena pro **NC-programátory** a **NC-projektanty**. Tento okruh osob potřebuje speciální znalosti o

- fungování, syntaxi a příkazech IndraMotion MTX-standardního programování.

Programování, spouštění a obsluhu stejně jako změny programových parametrů smí provádět pouze vhodně vyškolený odborný personál! Tento personál musí být schopen identifikovat možná nebezpečí, která mohou být způsobena programováním, změnami programů a obecně mechanickým, elektrickým nebo elektronickým vybavením.

Zásahy do hardwaru a softwaru našich výrobků, které nejsou popsány v této příručce, smí provádět pouze náš odborný personál.

Při nekvalifikovaných zásazích do hardwaru či softwaru nebo při nedodržování výstražných pokynů uvedených v této příručce a rozmístěných na stroji může dojít k těžkým zraněním nebo věcným škodám.

Popsané výrobky smí instalovat a udržovat pouze odborní elektrikáři podle IEC 60364-4-41 (změněno), kteří znají obsah této příručky.

Jedná se o osoby, které

- na základě svého odborného vzdělání mají znalosti a zkušenosti a na základě svých znalostí platných norem dokážou posoudit vykonávané práce a identifikovat případná nebezpečí.
- díky mnohaleté činnosti ve srovnatelném oboru mají stejné znalosti jako po odborném vzdělání.

Věnujte v této souvislosti pozornost naší bohaté nabídce školení. Nejaktuálnější informace o nabídce školení a školicích systémech naleznete na <http://www.boschrexroth.com>.

Informace vám sdělí také:

- naše středisko Didactic Center Erbach,
Telefon: (+49) (0 60 62) 78-600
- naše školicí centrum Lohr-Wombach,
Telefon: (+49) (0 93 52) 40 43 25.

Bezpečnostní pokyny

1.3 Bezpečnostní pokyny pro výroby



Varování před nebezpečným elektrickým napětím!



Varování před nebezpečím z baterií!



Elektrostaticky ohrožené součástky!



Varování před škodlivým světelným zářením (optické vysílače)



Před otevřením vytáhněte síovou zástrčku!



Ochranný vodič PE



Uzemnění všeobecně

Bezpečnostní pokyny

1.4 Bezpečnostní pokyny v této příručce



NEBEZPEČNÉ ELEKTRICKÉ NAPĚTÍ

Tento symbol varuje před **nebezpečným elektrickým napětím**. Při nepřesném dodržování nebo nedodržování tohoto pokynu může dojít ke **zranění osob**.



NEBEZPEČÍ

Tento symbol se používá, když může při nepřesném dodržování nebo nedodržování pokynů dojít ke **zranění osob**.



POZOR

Tento symbol se používá, když může při nepřesném dodržování nebo nedodržování pokynů dojít k **poškození přístrojů nebo souborů**.

☞ Tento symbol se používá, když je nutné vás upozornit na něco zvláštního.

★ Tento znak signalizuje, že se jedná o činnost, kterou máte provést.

Bezpečnostní pokyny

1.5 Bezpečnostní pokyny pro popsany výrobek



NEBEZPEČÍ

Nebezpečí života kvůli nedostatečným zařízením nouzového vypnutí!

Zařízení nouzového vypnutí musí být ve všech provozních režimech systému účinná a dosažitelná. Odblokování zařízení nouzového vypnutí nesmí mít za následek nekontrolované opětovné spuštění systému!

Nejprve zkontrolujte řetězec nouzového vypnutí a pak ho teprve zapněte!



NEBEZPEČÍ

Nesprávné nebo nechtěné pohyby os!

Vyzkoušejte nové programy nejprve pečlivě bez pohybu os! Řídící jednotka nabízí k tomuto účelu v obslužné oblasti 'Zpracování' možnost zablokovat pomocí soft kláves pohyby os, resp. výstupy pomocných funkcí.



NEBEZPEČÍ

Nesprávné nebo nežádoucí reakce řídicí jednotky!

Rexroth neručí za následné škody, které vyplynou ze zpracování NC-programu, jednotlivé NC-věty nebo z ručního pojezdu os. Rexroth rovněž neručí za následné škody, kterým bylo možné zabránit odpovídajícím naprogramováním jednotky SPS!



NEBEZPEČÍ

Dodatečné vybavení nebo změny mohou mít nepříznivý vliv na bezpečnost popsanych výrobků!

Následkem mohou být těžká zranění a věcné nebo ekologické škody. Případné dodatečné vybavení nebo změny zařízení s použitím součástek od cizích výrobců proto musí být schváleny firmou Rexroth.



NEBEZPEČÍ

Vysoký výkon svítivých diod v optických kabelech může při přímém pohledu ohrozit zrak.

Při zapnutém měniči se nedívejte do svítivé diody nebo do otevřeného konce krátkého připojeného vedení.



NEBEZPEČNÉ ELEKTRICKÉ NAPĚTÍ

Údržbové práce se musí provádět, není-li předepsáno jinak, zásadně jen při vypnutém zařízení! Přitom musí být zařízení zajištěno proti nepovolanému nebo neúmyslnému opětovnému zapnutí.

Pokud je nutné provádět měřicí nebo kontrolní práce na zapnutém zařízení, musí je vykonávat odborní elektrikáři.

Bezpečnostní pokyny

**NEBEZPEČÍ****Pohyby nástroje a os!**

Posuvové motory a motory vřeten generují velmi vysoké mechanické síly a mohou s velkou dynamikou velice rychle akcelerovat.

- Při zapnutém zařízení se nikdy nezdržujte v nebezpečné zóně stroje!
- Nikdy nevyřazujte z provozu bezpečnostní funkce zařízení!
- Poruchy na zařízení ohlaste okamžitě údržbovému, resp. servisnímu oddělení!

**POZOR**

Smíte používat jen námi schválené náhradní díly!

**POZOR**

Při manipulaci s konstrukčními skupinami a prvky dodržujte všechna opatření pro ochranu proti elektrostatickým výbojům! Eliminujte elektrostatické výboje!

Dodržujte následující ochranná opatření pro elektrostaticky ohrožené moduly a konstrukční prvky (EGB)!

- Personál odpovědný za skladování, transport a manipulaci musí být vyškolen v ochraně před elektrostatickým výbojem.
- Součástky ohrožené elektrostatickým výbojem musí být skladovány a transportovány v předepsaných ochranných obalech.
- Se součástkami ohroženými elektrostatickým výbojem se smí manipulovat zásadně jen na k tomu určených pracovištích.
- Personál, pracovní desky a všechny přístroje a nástroje, které mohou přijít do kontaktu se součástkami ohroženými elektrostatickým výbojem, musí být na stejném potenciálu (např. uzemněné).
- Vytvořte uzemňovací pásek. Uzemňovací pásek musí být spojen pomocí kabelu s integrovaným 1-M Ω odporem s pracovní deskou.
- Součástky ohrožené elektrostatickým výbojem nesmí v žádném případě přijít do kontaktu s předměty, které mohou být nabitě; mezi ně patří většina plastů.
- Při montáži součástek ohrožených elektrostatickým výbojem do přístrojů a při jejich demontáži musíte odpojit přístroje od elektrického napájení.

Bezpečnostní pokyny

Ochranné známky

Všechny ochranné známky softwaru, které jsou při expedici nainstalovány na našich výrobcích, jsou vlastnictvím příslušných výrobců.

Při dodání existuje na každý nainstalovaný software autorské právo. Software se smí rozmnožovat pouze s naším svolením, resp. v souladu s platnými licenčními podmínkami příslušného výrobce.

PROFIBUS® je registrovaná ochranná známka společnosti PROFIBUS Nutzerorganisation e. V.

SERCOS interface™ je registrovaná ochranná známka zájmové společnosti SERCOS interface e. V.

Základy NC-programování

2 Základy NC-programování

NC-řízení dostává prostřednictvím NC-programu (dílčího programu) všechny informace potřebné pro zpracování obrobku na obráběcím stroji.

Struktura takového NC-programu je variabilní, takže lze obrábět téměř libovolné obrobky s nejrůznějšími technologiemi (frézování, soustružení, broušení atd.). V dílčím programu jsou uloženy jak pohybové informace, které popisují dráhu nástroje vzhledem k obrobku, tak i technologické údaje.

Pohybové informace jsou rozděleny na jednotlivé elementární prvky kontur (přímky, kruhy, spirály, křivky spline, křivky nurb atd.). Pohyby pro každý z těchto geometricky jednoduchých konturových prvků pak může provádět řídicí jednotka vždy v jednom obráběcím kroku, když jsou v NC-programu stanoveny všechny obráběcí kroky ve správném pořadí a se všemi nezbytnými okrajovými podmínkami. Nezbytné okrajové podmínky se mimo jiné skládají z technologických funkcí (rychlosti, počty otáček atd.) a z pomocných funkcí stroje (např. pro chlazení kapalínou či upínání os).

 **Základní směrnice o struktuře NC-programu jsou uvedeny v DIN 66025.**

Norma DIN 66025 "Struktura programů pro číslicově řízené výrobní stroje" (část 1 a 2) obsahově souhlasí s mezinárodní normou: ISO/DIS 6983 a ISO/DP 6983 "Numerical control of machines".

IndraMotion MTX spravuje NC-programy v "systému souborů" řídicí jednotky. Kromě toho existuje možnost připojit externí jednotky a zpracovávat programy přímo z nich.

Další údaje o **systému souborů a přístupových oprávněních**, stejně jako informace o sestavování a editaci dílčích programů naleznete v návodu k obsluze MTX.

Základy NC-programování

2.1 Základní informace o standardním programování a programování CPL

Řídicí jednotka nabízí dvě možnosti programování:

- Standardní programování nebo programování DIN
- Programování CPL (CPL: Customer Programming Language).

Standardní programování nebo programování DIN popisuje sled pohybů a jejich okrajové podmínky (geometrie, kinematika, dynamika, opravy atd.). Standardní programování je čistý příkazový jazyk pro řízení pohybů na stroji a ovládání určitých konkrétních funkcí stroje.

Syntaxe IndraMotion MTX se skládá z příkazů, které jsou stanoveny v rámci DIN 66025 (G- a M-kódy), a z podstatného rozšíření v oblasti G-kódů a přidavných jazykových syntaktických prvků.

Základními prvky standardního programování jsou tzv. NC-funkce, kterým je vždy přiřazena programovací syntaxe.

NC-funkci mohou být přiřazeny další parametry, pomocí nichž je funkce parametrizována.

Příklad:

NC-funkce:	G2	Kruhová interpolace ve směru hodinových ručiček
Parametry:	I, J, K, R	Souřadnice středu, poloměr

Programování CPL (Customer Programming Language) se orientuje na standard jazyka BASIC, ale obsahuje také prvky struktury podobné Pascalu. Proto se dá snadno naučit.

CPL je skutečný programovací jazyk a představuje tedy rozšíření možností programování. Obsahuje programové prvky, které jsou mimo ohnisko řízení stroje, přičemž speciální systémové funkce umožňují přístup k systémovým datům řídicí jednotky.

Programování CPL nabízí následující možnosti:

- Symbolické programování s proměnnými
- Zpracování znakových řetězců
- Zpracování souborů
- Matematické operátory: +, -, *, /, a trigonometrické funkce, ...
- Relační operátory: =, <, >, ...
- logické operace: NOT, AND, OR, ...
- Kontrolní struktury pro řízení průběhu programu: REPEAT, WHILE, FOR, IF, CASE, GOTO, ...
- Systémové funkce pro zjišťování interních systémových stavů: polohy, aktivní funkce, data nástrojů, signály rozhraní, ...
- Procesové služby: volba programu, základní poloha, spuštění programu, zadávání provozních režimů

Tak je možné vytvářet a ukládat libovolné obráběcí procedury ve variabilním způsobu zápisu.

Základy NC-programování

Instrukce CPL se zapisují s ohledem na formální předpisy zásadně velkými písmeny.

Používání CPL vede:

- ke zkrácení NC-programů u opakujících se postupů a stejných částí programu
- k variantám programu závislým na stavu díky přístupu k systémovým stavům řídicí jednotky.

Podstatný rozdíl mezi standardním programováním a programováním CPL spočívá v tom, že všechny části CPL jsou spuštěny již v okamžiku přípravy věty, přímo při čtení příslušného řádku programu. Při dalším zpracování věty a v době interpolace NC-věty již tedy součásti CPL neexistují.

Prvky CPL uvnitř dílčího programu musí být označeny!

V rámci dílčího programu lze používat programování CPL stejně jako standardní programování. Z důvodů jednoznačnosti a v neposlední řadě také kvůli efektivní interpretaci programu musí být součásti programu CPL speciálně označeny:

- Věty programu, které obsahují výhradně prvky CPL, se označují přímo **na začátku věty celočíselnou hodnotou** bez znaménka (číslo řádku).

Příklad:

20 Ax1\$="U"

V CPL-větě 20 proměnné String "Ax1\$" přiřadit hodnotu "U".

- Jsou-li prvky CPL naprogramovány **uvnitř standardní NC-věty** (např. kvůli programování symbolických proměnných), musí být tyto součásti CPL vždy uzavřeny **v lomených závorkách** ("[" a "]"). To slouží k tomu, aby byla NC-funkcím a parametrům NC-funkcí přiřazena parametrizovatelná hodnota. Proto jsou zde uvnitř lomených závorek dovoleny jen takové výrazy, které se mohou v CPL nacházet na pravé straně znaku "=" (proměnné, matematické výrazy, systémové funkce, které dávají odpovídající hodnotu).

Příklad:

N10 G1 Y[CPOS(X)]

V NC-větě N10 má osa Y pojet na stejnou pozici obrobku, která byla naposledy naprogramována pro osu X.

N20
ROT([360/A+SIN(B)])

V NC-větě N20 aktivujte vstupní pomůcku otáčení. Úhel otočení je definován výrazem CPL.

30 XPOS=100.5

V CPL-větě 30 bude proměnné "XPOS" přiřazena nejprve hodnota 100.5.

N40 G1 X[XPOS] Y10

V NC-větě N40 je obsah proměnné "XPOS" předán jako hodnota souřadnice ose X.

Základy NC-programování

**Mějte na paměti okamžiky interpretace mezi
součástmi CPL a standardního jazyka!**

Díky speciálnímu označování jazykových prvků CPL může programátor okamžitě poznat, které části programu již byly spuštěny během přípravy věty.

Prvky standardního programování se navenek projevují až v okamžiku provádění věty na stroji.

Příklad:

N523 KvProg(X[@KVX])

V okamžiku provedení věty N523 je pro osu X použita jako KV ta hodnota, kterou **byla** osazena permanentní proměnná @KVX v okamžiku přípravy věty.

Až do okamžiku provedení věty N523 však může být proměnná @KVX osazena úplně jinou hodnotou (která byla např. zapsána z jiného kanálu)!

Abyste se zde vyhnuli nejasnostem, musíte použít funkci "WAIT". Příkaz Wait blokuje přípravu věty, dokud není kompletně zpracována NC-věta naprogramovaná bezprostředně před touto větou. Příprava věty je tak synchronizována s aktivním stavem (okamžik interpolace) řídicí jednotky. Zařízení následně pokračuje přípravou další naprogramované věty. V tomto okamžiku neexistují vůbec žádné připravené věty, protože byly předtím všechny zpracovány.

Příklad:

WAI T

N523 KvProg(X[@KVX])

Synchronizace přípravy věty

V okamžiku provedení věty N523 je pro osu X použita jako KV ta hodnota, kterou byla osazena permanentní proměnná @KVX v okamžiku přípravy věty. Protože však nyní byla příprava věty synchronizována, jedná se o aktivní hodnotu proměnné.

Základy NC-programování

2.2 Spojování NC-programů

Po volbě programu se program nejprve kontroluje s ohledem na syntaxi a možné cíle skoků a volání podprogramů. Kromě toho se pro proměnné CPL vytvářejí odpovídající správné struktury, ke kterým má řídicí jednotka během chodu programu přístup. Tento proces se označuje jako **spojování** (nebo příprava).

Výsledkem úspěšné procedury spojování je vytvoření **tabulky spojení** pro příslušný NC-program. Všechny tabulky spojení IndraMotion MTX jsou uloženy ve speciálním adresáři, který je určen v parametru stroje 3080 00004. Název tabulky spojení programu se skládá z názvu dílčího programu, k němuž je připojena přípona ".l" (l: Link).

Během náběhu řízení hledá řídicí jednotka ke všem existujícím tabulkám spojení příslušný NC-program. Hledání přitom probíhá podle vyhledávací cesty nastavené v parametru stroje 3080 00001. Tabulky spojení, pro které není nalezen žádný dílčí program, jsou vymazány.

Při nové volbě již spojeného NC-programu používá IndraMotion MTX již existující tabulku spojení, pokud se mezitím nezměnil dílčí program. Pro změně programu probíhá nové spojení.

Spojování podprogramů

Pokud jsou ve spojovaném programu volány podprogramy, kontroluje IndraMotion MTX, jestli pro ně existují platné tabulky spojení. Pokud ano, nejsou takové podprogramy znovu spojovány. Tím se dá podstatně zkrátit proces spojování.

Jestliže se v dílčím programu nevyskytují žádné prvky CPL (skoky, proměnné CPL, výrazy CPL atd.), ale pouze NC-věty (DIN) a volání podprogramů, nemusí být program před provedením výslovně spojen. V takovém případě je možné při volbě programu přepnout soft klávesu "CPL Prog / DIN Prog" na nastavení "DIN Prog". Podprogramy jsou pak v průběhu programu připojovány podle potřeby, což může za určitých okolností vést ke zpožděním v obráběcím procesu.

Je-li volání podprogramu naprogramováno pomocí proměnných CPL (např. P[UP\$]), proběhne spojovací procedura pro tento podprogram až v průběhu programu, protože teprve tehdy může být uvolněn název proměnné. Podprogram vyvolaný tímto způsobem je tedy, pokud ještě neexistuje žádná tabulka spojení, vždy dodatečně spojen.

Ovlivňování spojovacího procesu pomocí identifikace DIN/CPL

Spojování se dá ovlivnit za předem popsanych předpokladů také přímo, instrukcemi v dílčím programu, nezávisle na aktuální poloze soft klávesy "CPL Prog / DIN Prog".

Přitom je třeba naprogramovat na začátku prvního řádku programu klíčové slovo "(DIN)". Také v tomto případě se podprogramy spojují podle potřeby v průběhu programu.

Základy NC-programování

Instrukce (DIN) může být naprogramována také v prvním řádku každého podprogramu, což naopak znamená, že takto označený podprogram nebude spojen.

Alternativně může být klíčové slovo "**DIN**" (bez kulatých závorek) zapsáno také přímo za voláním podprogramu ve volajícím programu (viz též kapitola 2.6 Podprogramy).

Analogicky k instrukci (DIN) existuje také klíčové slovo "**(CPL)**", které se rovněž programuje na začátek prvního řádku programu. To si vynucuje vytvoření tabulky spojení pro příslušný program i v případě, že bylo pod nastavením vybráno "DIN Prog" nebo byl vyvolán jako podprogram s identifikací DIN.

Jednotky spojení

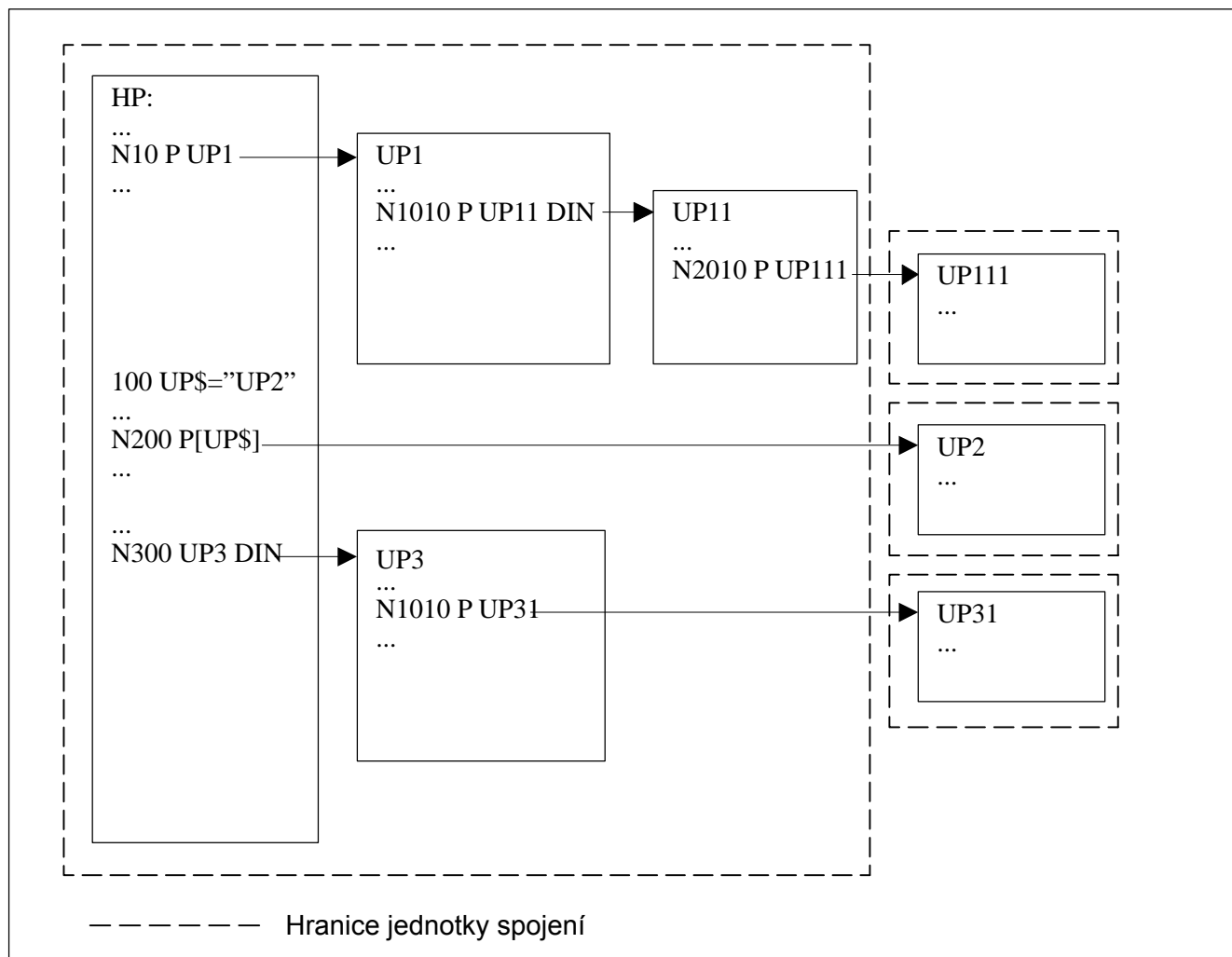
Dodatečným spojováním v průběhu programu vznikají vždy samostatné jednotky spojení:

- Pokud je volání podprogramu naprogramováno pomocí proměnné CPL (např. P[UP\$]), patří volající program a volaný podprogram k **různým** jednotkám spojení.
- Je-li podprogram naprogramován přímo, ale dodatečně spojen (např. kvůli identifikaci DIN při volání podprogramu: P UP DIN), patří volající program a volaný podprogram sice ke **stejně** jednotce spojení, ale všechny další podprogramy vyvolané podprogramem náležejí do **jiné** jednotky spojení.

Pokud jde o platnost proměnných CPL mezi programy různých jednotek spojení, platí:

- Globální proměnné jsou pro každou jednotku spojení nově definovány. Jestliže tedy existují globální proměnné se stejným názvem v (pod)programech různých jednotek spojení, jedná se o navzájem nezávislé proměnné.
- Pokud mají být vyměňovány informace o pevných proměnných mezi programy různých úrovní spojení, může to probíhat pouze pomocí permanentních nebo strukturovaných proměnných CPL.

Základy NC-programování



2.3 Základní součásti NC-programu

NC-program se skládá nejméně z 1 **programové věty**.

Pro programové věty platí:

- V každé programové větě smí být naprogramován nejvýš 1 konturový prvek (např. přímka, kruhový oblouk).
- Prázdné řádky před nebo za programovou větou jsou dovoleny pro lepší členění, resp. čitelnost programového kódu.
- Programová věta nesmí být delší než 512 znaků.
- Programová věta je ukončena ASCII-znakem <LI NEFEED>.
- Programová věta se skládá nejméně z jednoho **programového slova** (např. NC-funkce, funkční parametr).

Základy NC-programování

Instrukce

Instrukcemi se rozumějí programová slova, která přímo nebo nepřímo ovlivňují dráhu nástroje, průběh programu, stav nebo způsob reakce řídicí jednotky. Např. všechny NC-funkce jsou typickými instrukcemi. Dostupné NC-funkce naleznete s příslušným syntaktickým předpisem v kapitole 4.

Zvláštní místo mezi instrukcemi zauímají **dráhové funkce**:

Dráhové funkce popisují, jakým způsobem má proběhnout najetí na pozici (např. přímka, kruh, s interpolací zúčastněných os nebo bez ní, nájezdový pohyb v posuvu nebo rychloběhu atd.).

Příklady: G0, G1, G2

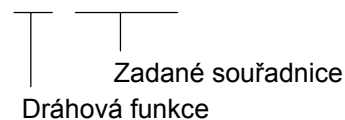
Dráhové funkce se často programují ve větě s údajem o poloze, dráze, resp. poloměru. V takových případech spouštějí dráhové funkce vždy také pojezdové pohyby.

Rovněž polohové nebo dráhové údaje, které jsou naprogramovány v programové větě bez dráhové funkce, spouštějí vždy pojezdové pohyby, protože nějaká dráhová funkce je vždy aktivní.

Příklad:

Dráhová funkce se zadanými souřadnicemi

G01 X40 Y50 Najetí v posuvu do X40/Y50

**Doplňující podmínky**

Doplňujícími podmínkami se rozumějí programová slova, pomocí kterých se na stroji nastavují okrajové podmínky potřebné pro obrábění nebo technologii.

Důležitá programová slova, která fungují jako doplňující podmínky:

F<číslo> ovlivňuje posuv synchronních os

FA<číslo> ovlivňuje posuv asynchronních os

S<číslo> ovlivňuje počet otáček vřetena

M<číslo> aktivuje M-funkce (např. volba převodového stupně, směr otáčení vřetena, vyvolání podprogramu). Také pomocné funkce se často programují jako M-funkce.

T<číslo> vybírá nástroje

Programová slova jsou podrobněji vysvětlena v kapitole 4.

Základy NC-programování

Příklad: Dráhová informace s doplňujícími podmínkami

G01 G71 X40 Y50 F250 S500 T05 M03

_____ |
Doplňující podmínky

Najet v posuvu do X40/Y50; s naprogramovanou hodnotu F (posuv = 250 mm/min) a hodnotou S (počet otáček = 500 mm/min); s vřetenem otáčejícím se doprava a připravit nástroj T05 v zásobníku nástrojů.

- ☞ **Ačkoli takové doplňující podmínky neslouží bezprostředně k popisu kontury nebo průběhu dráhy, mohou přesto ovlivňovat pohyby na stroji nebo je také spouštět (např. pohyb zásobníku nástrojů)!**

Základy NC-programování

2.3.1 Programová slova

Programové slovo je:

1. **NC-funkce**, nebo
2. **parametr s hodnotou nebo seznamem parametru**. Parametr je v dílčím programu identifikován svou syntaxí (adresou).

☞ **Synchronizační příkazy NC věty a volání podprogramů jsou speciální případy NC-funkcí, protože mohou být naprogramovány společně se seznamem parametrů CPL.**

Každé programové slovo se vždy skládá z jednoho nebo dvou **dílčích slov**, která mohou být kombinována následujícím způsobem:

Programové slovo:	Dílčí slovo 1:	Dílčí slovo 2:	Příklady:
NC-funkce	Syntaxe funkce	-	AxAcc G0 G52.1
	Syntaxe funkce	Seznam parametrů	AxAcc(...) G0(NIPS)
	Syntaxe funkce	Hodnota	D5 M777
Příkaz synchronizace	Syntaxe příkazu	-	OFFSTOPA
	Syntaxe příkazu	Seznam parametrů CPL	WPV[@9=10]
Podprogram	UP-syntaxe, UP-název	-	P UP1 G4711
	UP-syntaxe UP-názvy	Seznam parametrů CPL	P UP2[7,@25] G81[Z,R1]
Parametr	Syntaxe parametru	Hodnota	X-23.45 Y=AC(40) S250 F5000
	Syntaxe parametru	Seznam parametrů	O(0,0,1) ROTAX(0,45)

- ☞ **G- a M-kódy se mohou vyskytovat s dílčím slovem 2 i bez něho:**
- U **G- a M-kódů s pevnou vnitřní funkcí** je číselná hodnota součástí syntaxe, např. **G0, G17, G54, M0, M3, M19, ...**
 - U **uživatelsky definovaných funkcí (volání podprogramů a pomocné funkce)** není číselná hodnota součástí syntaxe, ale představuje hodnotu pro funkci. Tyto funkce jsou zkonfigurovány v rámci parametrů stroje.

Základy NC-programování

Programová slova z NC-funkcí


Zde obecně platí:

- NC-funkce se mohou syntakticky skládat jak z G- a M-kódů, tak i z jazykových prvků.

Příklady: "G0", "G41", "G141", "G52. 0", "M30", "M r r o r (...)"

Všechny dostupné NC-funkce naleznete s příslušným syntaktickým předpisem v kapitole 4.

- NC-funkce mohou mít další parametry, s nimiž je možné řídit způsob práce NC-funkce. Přitom rozlišujeme 3 případy:
 - Parametry, které jsou naprogramovány jako samostatné programové slovo v NC-větě, např. "G02 X10 I1.3 J2.5 G94 F1000". Jedná se přitom převážně o takové parametry, které jsou definovány v DIN 66 025.
 - Parametry, které jsou sice naprogramovány jako programové slovo, avšak uvnitř **seznamu parametrů se specifickými syntaktickými prvky** uzavřeného v kulatých závorkách, např. "KvProg(X1.2,Y1.2)".
 - Parametry uvnitř **seznamu parametrů bez specifických syntaktických prvků** uzavřeného v kulatých závorkách, pro které je naprogramována pouze hodnota na definovaném místě, např. "Coord(0,1)", "Rotate(45)", "GetAxis(Z1,Z,W,REV)".
- Seznamy parametrů mohou být volitelné a obsahují nejméně jeden možný parametr. Jestliže existuje pro seznam parametru více parametrů, jsou tyto parametry navzájem oddělené čárkou.

 **Existující, ale ne současně použitelné parametry (alternativní parametry) označujeme v této příručce znakem "I" mezi oběma alternativami. Znak "I" se neprogramuje.**

- Parametry NC-funkcí mohou být volitelné. Jestliže nejsou naprogramovány, používají se zpravidla standardní nastavení, která mohou být pevně kódována nebo uložena v parametrech stroje.

 **V této příručce označujeme**

- **volitelné parametry složenými závorkami a**
 - **symbolsy pro naprogramované hodnoty úhlovými závorkami.**
- Složené a úhlové závorky se neprogramují.**

Příklad:

Syntaktický předpis: TangTool Ori ({SYM<s>}, {ANG<a>})

Na základě syntaktického předpisu rozpoznáte, že jsou parametry SYM a ANG volitelné. K SYM se místo symbolu <s>, resp. k ANG se místo symbolu <a> musí naprogramovat vhodná hodnota. Které hodnoty pro <s> a <a> jsou dovoleny, je zpravidla uvedeno v syntaktickém předpisu.

Programování možné k výše uvedenému syntaktickému předpisu by proto bylo "TangTool Ori (SYM4)".

Základy NC-programování

Seznamy parametrů bez specifických syntaktických prvků:

Všechny parametry v tomto typu seznamu jsou konstanty a předávají se přímo jako číselná hodnota nebo název. Musí být naprogramovány v přesně definovaném pořadí, protože význam každého parametru je dán pouze jeho pozicí v seznamu.

V důsledku záměny pořadí parametrů může vzniknout úplně jiné působení:

Příklad:

GetAxis(X1, X, Y3, Y)	Osy se systémovými názvy X1 a Y3 jsou převzaty do kanálu a jsou jim přiřazeny názvy kanálů "X" a "Y".
GetAxis(X1, Y3, X, Y)	Osy se systémovými názvy X1 a X jsou převzaty do kanálu a jsou jim přiřazeny názvy kanálů "Y3" a "Y".

Parametry v seznamu parametrů bez specifických syntaktických prvků mohou být volitelné. V takových případech platí:

- Volitelné parametry na začátku a uvnitř seznamu mohou být vynechány, ale musí být naprogramována příslušná čárka. Jedině tak zůstane pozice parametrů v seznamu jednoznačná.
- Volitelné parametry na konci seznamu mohou být vynechány a seznam může být ukončen znakem ")".

Příklad:

GetAxis(<Phy.Ax1> {,<Log.Ax1>} {,<Phy.Ax2>} {,<Log.Ax2>} {,...})
 naprogramováno bez parametrů uprostřed: GetAxis(X,,Z)
 naprogramováno bez parametrů na konci: GetAxis(X)

Používání výrazů **CPL v seznamech parametrů** bez specifických syntaktických prvků:

- Pokud jsou jednotlivé prvky převzaty jako výraz CPL, musí být vždy uzavřeny v lomených závorkách.

Příklad:

Názvy proměnných jako předávané parametry:

10 AX1\$="U"	Proměnná String "AX1\$" má obsah "U".
20 AX2\$="V"	Proměnná String "AX2\$" má obsah "V".
:	
N100 G17([AX1\$], [AX2\$])	G17 upíná s osami U a V pracovní rovinu.

Seznamy parametrů se specifickými syntaktickými prvky:

Pořadí všech parametrů v tomto typu seznamu je libovolné, protože význam parametru je určen jeho naprogramovanou syntaxí.

Základy NC-programování

Příklad:

`"KvProg(X1. 2, Y1. 4, Z1. 6)" a``"KvProg(Z1. 6, Y1. 4, X1. 2)"`

mají stejný účinek, protože jednotlivé hodnoty KV mohou být díky adresám os X, Y a Z jednoznačně přiřazeny.

Pro syntaktické prvky stejně jako pro číselné hodnoty se smí používat výrazy CPL. Ty přitom musí být uzavřeny v lomených závorkách "[" a "]".

Příklad:

`20 WERT=1. 2`

Přiřadit proměnné "WERT" hodnotu 1.2.

`N30 KvProg(Y[WERT], X1. 2)`

Naprogramovat hodnoty Kv os Y a X. Osa Y obdrží hodnotu, která je v proměnné "WERT", osa X hodnotu 1.2 (konstanta).

Příklad:

`40 ACHSB$="X": FACT%=2`

Přiřadit proměnné String "ACHSB\$" hodnotu "X" a proměnné Integer "FACT%" hodnotu 2.

`N50 Scale([ACHSB$][FACT%])`

Zapnout pro osu "X" měřítko s faktorem 2.

Odpovídá naprogramování: Scale(X2)

Příklad:

`60 DIM PARAMETERS(10)`

Vytvořit znakové pole pro řetězec s max. 10 znaky.

`70 PARAMETER$="X2"``N80 AxAcc([Parameter$])`

Přiřadit ose "X" zrychlení osy 2 m/s².

Programová slova jako parametry

Parametry s následující hodnotou:

Pro taková programová slova obecně platí:

- Adresa začíná vždy písmenem a může se skládat z několika znaků.
- Programová slova s adresou a čísly se používají např. k programování
 - označení os a souřadnic (např. X..., Y..., Z..., B...)
 - poloměrů (R...) a parametrů interpolace (I..., J..., K...)
 - hodnot posuvu, resp. času (F...)
 - počtu otáček vřetena a řezných rychlostí (S..., S_i=...)
 - externích oprav nástrojů (ED...)
 - D-korektur (D...)
 - pomocných funkcí (M..., T...)

Základy NC-programování

- Nevýznamné nuly se nemusí programovat.
- Desetinná čísla se zapisují s desetinnou tečkou; nuly na konci se mohou vynechávat (např. "X100.500" odpovídá "X100.5")
- Pokud není naprogramováno žádné znaménko nebo je naprogramováno znaménko plus, je následující hodnota vždy interpretována jako kladná. Znaménko minus deklaruje zápornou hodnotu.

Příklad:

Programové slovo, skládající se z písmena adresy a čísla
(zde: hodnota souřadnice osy X)

X-2407.0458

				Hodnota za desetinnou tečkou
				Hodnota před desetinnou tečkou
				Znaménko
				Písmeno adresy

Parametry s následujícím seznamem parametrů:

Pro taková programová slova obecně platí:

- Platí v zásadě stejné údaje jako pro programová slova, která se skládají z adresy a čísla
- Seznam parametrů je naprogramován za adresou a uzavřen v kulatých závorkách. Jednotlivé prvky musí být navzájem odděleny čárkou.
- Prvky v seznamu parametrů jsou konstanty a předávají se přímo jako číselná hodnota nebo název. Musí být naprogramovány v přesně definovaném pořadí, protože význam každého parametru je dán pouze jeho pozicí v seznamu.
- Pokud jsou jednotlivé prvky převzaty jako výrazy CPL, musí být uzavřeny v lomených závorkách.
- Programová slova s adresou a seznamem parametrů se používají např. k programování
 - spline koeficientů (z.B. X(...)..., Y(...)..., Z(...)..., B(...)...)
 - orientace vektorů (O(...), ROTAX(...))

Programovací atributy

Při programování poloh os a souřadnic existuje volitelná možnost specifikovat pro naprogramovanou hodnotu polohy programovací atribut.

Bez programovacího atributu je hodnota polohy interpretována podle současného modálního stavu stroje.

S programovacím atributem je možné lokálně potlačit modální stav osy, resp. souřadnice.

Základy NC-programování

IndraMotion MTX zná následující atributy:

- AC(...): Naprogramovaný polohový údaj je nezávisle na G90/G91 interpretován jako absolutní.
- IC(...): Naprogramovaný polohový údaj je nezávisle na G90/G91 interpretován jako inkrementální.
- DC(...): Poloha naprogramovaná pro nekonečnou osu je najížděna po nejkratší dráze, nezávisle na nastaveních v parametrech stroje a nezávisle na naprogramování funkce "PosMode".
- ACP(...): Poloha naprogramovaná pro nekonečnou osu je najížděna s kladným směrem otáčení, nezávisle na nastaveních v parametrech stroje a nezávisle na naprogramování funkce "PosMode".
- ACN(...): Poloha naprogramovaná pro nekonečnou osu je najížděna se záporným směrem otáčení, nezávisle na nastaveních v parametrech stroje a nezávisle na naprogramování funkce "PosMode".

Příklady:

- | | |
|-------------------------------|---|
| N10 G90 G1 F1000 X10 Y=IC(15) | Osa X pojíždí absolutně do polohy 10, osa Y inkrementálně o 15 mm dál. |
| N20 G91 X=AC(15) Y5 | Navzdory G91 pojíždí osa X do absolutní polohy 15, osa Y pojíždí inkrementálně o 5 mm dál. |
| N30 B=DC(90) | Osa B (nekonečná osa) má najet po nejkratší dráze do polohy 90 stupňů. |
| N40 B=ACP(350) | Osa B má najet s kladným směrem otáčení do polohy 350 stupňů (dráha pojezdu: 260 stupňů). |
| N50 B=ACN(0) | Osa B má být polohována se záporným směrem otáčení na 0 stupňů (dráha pojezdu: 350 stupňů). |

Používání oddělovacích znaků mezi 2 dílčími slovy

Každé dílčí slovo se může skládat z jednoho nebo několika (řetězce) znaků. Každý programovatelný znak se dá klasifikovat podle následujících skupin:

- Písmena: "A" - "Z", "a" - "z"
- Číslice (včetně desetinné tečky!): "0" - "9", ".", "
- Znak, které fungují jako oddělovací: " ", "=", "+", "-", "(", ")"
- Jiné speciální znaky (dále nedůležité)

Mezi 2 sousedními dílčími slovy musí být naprogramován oddělovací znak, když:

- dílčí slovo uvedené vpředu končí písmenem **nebo** číslicí a
- dílčí slovo uvedené za ním začíná písmenem, **resp.** číslicí.

Základy NC-programování

To je např. v případě,

- když za NC-funkcí bez hodnoty a seznamu parametrů následuje další syntaxe, nebo
- když syntaxe parametru končí číslem a parametru má být přiřazena číselná hodnota.

Příklady:

1. N10 OVE FeedForward(. . .) vhodný oddělovací znak: " ".

2. N20 X2=2 vhodný oddělovací znak: "="

Ve 2. případě je možné alternativně použít jako oddělovací znak také " " nebo "+", při záporné přiřazené hodnotě je možné též "-".

Používání oddělovacích znaků může mít význam také v seznamech parametrů s prvky souvisejícími se syntaxí!

Zvláštnosti při označování os a souřadnic

Který identifikátor os a souřadnic je zapotřebí v řídicí jednotce, musíte stanovit v parametrech stroje:

- MP 1003 00001 Systémové označení os
- MP 7010 00010 Kanálové označení os
- MP 7010 00020 Volitelné názvy os kanálu
- MP 7080 00010 Názvy souřadnic kanálu
- MP 7080 00020 Kartézské názvy souřadnic kanálu

Označení os a souřadnic začínají vždy písmenem a smí:

- se skládat z jednoho nebo více písmen, přičemž výsledný znakový řetězec nesmí být stejný jako NC-funkce.
Příklady: "X", "PALETTE"
- se smí skládat z jednoho nebo více písmen a končit číslem.
Příklady: "X1", "PALETTE1"
- Přitom je pro oddělení mezi identifikátorem a následující hodnotou nutné naprogramovat vhodný oddělovací znak: "=", "+", "-" nebo mezeru.

Příklad:

Jsou definovány identifikátory os "X" a "X2".

N40	G1	X10	Osa X pojíždí do polohy 10
N50	G1	X20	Osa X pojíždí do polohy 20
N60	G1	X2. 5	Osa X pojíždí do polohy 2.5
N70	G1	X=2. 2	Osa X pojíždí do polohy 2.2
N80	G1	X 2. 8	Osa X pojíždí do polohy 2.8
N90	G1	X2	Osa X pojíždí do polohy 2
N100	G1	X2 1	Osa X2 pojíždí do polohy 1
N110	G1	X2=2. 8	Osa X2 pojíždí do polohy 2.8
N120	G1	X2+3	Osa X2 pojíždí do polohy 3
N130	G1	X2- 2. 4	Osa X2 pojíždí do polohy -2.4

Základy NC-programování

2.3.2 Konec programu

Konec programu, resp. podprogramu je dosažen v následujících případech

- na konci souboru nebo
- na programovém řádku, který obsahuje "M2", "M02" nebo "M30".
Bližší informace o uvedených M-funkcích naleznete na straně 3-88.

Příklad:

```
:  
N250 . . .           N250 je poslední programová věta.  
M30                 Konec programu.
```

Na konci podprogramu následuje zpětný skok do volajícího programu. Všechny modální stavy ("modální" viz strana 2-18) přitom zůstávají zachovány.

Po skončení hlavního programu následuje skok na jeho začátek a čekání na nový příkaz "NC-Start". Pokud byla pro ukončení hlavního programu použita funkce "M2", "M02" nebo "M30", jsou modální stavy nastaveny na složku M30 Initstring.


2.4 Působení programových slov

Programová slova mohou působit "modálně" nebo "nemodeálně".

modální

"modální" znamená, že programové slovo zůstává v každé další programové větě účinné tak dlouho, dokud

- není naprogramováno stejné programové slovo s jinou hodnotou,
- není naprogramováno jiné programové slovo, které zruší jeho působení, nebo
- není cíleně vypnuta funkce programového slova.

 **Jako synonymum pro "modální" se někdy používá také výraz "samodržné".**


Příklad dílčího NC-programu:

N10 F1000	Nastavte rychlost posuvu na 1000 mm/min. F1000 působí modálně.
N20 G0 X0 Y0	Lineární interpolace v rychloběhu do polohy X0/Y0. G0 působí modálně.
N30 Z100	Lineární interpolace v rychloběhu do polohy Z100.
N40 G1(IPS1) X10 Y10	Lineární interpolace (v posuvu; s jemným polohovacím oknem) s 1000 mm/min do polohy X10/Y10. G1 ruší působení G0. G1(IPS1) působí modálně.
N50 X20	Lineární interpolace (v posuvu; s jemným polohovacím oknem) s 1000 mm/min do polohy X20/Y10.
N60 G1(IPS2) X30 Y30	Lineární interpolace (v posuvu; s hrubým polohovacím oknem) s 1000 mm/min do polohy X30/Y30. G1 bylo naprogramováno s jinou hodnotou (IPS2). G1(IPS2) působí modálně.
N70 X40 Y40 F500	Lineární interpolace (v posuvu; s hrubým polohovacím oknem) s 500 mm/min do polohy X40/Y40. F bylo naprogramováno s jinou hodnotou (500). F500 působí modálně.
N80 G0 X0 Y0	Lineární interpolace v rychloběhu do polohy X0/Y0. G0 ruší působení G1(...). G0 působí modálně.
N90 Scal e(X2, Y2)	Zapnout funkci měřítka. Funkce měřítka působí modálně.
:	Funkce měřítka je nadále účinná.
N200 Scal e()	Vypnout funkci měřítka.

Základy NC-programování

nemodální

"nemodální" znamená, že programové slovo působí pouze v programové větě, ve které bylo naprogramováno.

 **Jako synonymum pro "nemodální" se někdy používají také pojmy "nesamodržné" nebo "lokální".**

Příklad dílčího NC-programu:

N10 G1 F1000	Aktivovat lineární interpolaci. G1 působí modálně. Nastavit rychlost posuvu na 1000 mm/min. F1000 působí modálně.
N20 G75 X100 Y100	Měřicí dotyk G75 působí nemodálně. Funkce G75 potlačuje v této větě modální funkci G1.
N30 Z100	Lineární interpolace v posuvu do Z100. G1 dál působí modálně.

2.5 Speciální prvky pro vytváření programů

2.5.1 Označení kanálů

Tím určujete, ve kterém kanálu smí být příslušný program výhradně používán. Spuštění v jiném kanálu generuje běhovou chybu. Označení kanálu je zapsáno na začátku programu.

Syntaxe:

$\$$ <číslo kanálu>

Příklad:

N10	$\$$ 2	Tento program může běžet pouze v kanálu 2.
:		Programové instrukce
M30		Konec programu

2.5.2 Číslo vět

- Programové věty **můžete** označit čísly, abyste zlepšili čitelnost programového kódu.
Jestliže zadáte jednoznačná čísla vět, existuje možnost naprogramovat skoky na tato čísla.
- Programové věty, které obsahují výhradně prvky CPL, **musí** být označeny čísly. Takové věty se označují také jako **věty CPL**.

Platí:

- Číslo vět musí být vždy naprogramována jako první NC-slovo v programovém řádku.
- Číslo **standardních NC-vět** se skládají z písmena adresy "N" a přímo následujícího kladného čísla (Příklad: "N10", "N10.2").
Pamatujte si, že výrazy CPL ve standardních NC-větách musí být uzavřeny v lomených závorkách.
- Číslo **čistých vět CPL** se skládají výhradně z kladného čísla bez znaménka (Příklad: "10", "11.9").
Za číslem věty je naprogramována instrukce CPL nebo CPL-deklarační věty.
- Je-li věta CPL ukončena znakem ":", musí k ní být připojena další věta CPL, která však nemá číslo řádku.

2.5.3 Prázdné řádky v programovém kódu

Pomocí prázdných řádků je možné strukturovat program a tím zvýšit jeho přehlednost. Prázdné řádky jsou řídicí jednotkou přeskakovány.

2.5.4 Komentáře v dílčím programu

Komentáře jsou v průběhu programu přeskakovány řídicí jednotkou. Komentáře se používají pro:

- zdokumentování programového kódu nebo jeho opatření vysvětlivkami
- komentování celých programových řádků nebo jednotlivých částí.

Dobře komentované programy usnadňují a urychlují pozdější přepracování, když jsou např. zapotřebí změny v programu. Programový soubor se ovšem s každým znakem komentáře zvětšuje o 1 bajt.

Komentování celých programových řádků

Můžete úplně skrýt každý programový řádek, nezávisle na tom, jestli se jedná o standardní NC-větu nebo o větu CPL. Při komentování:

- naprogramujete na začátku věty středník ";," nebo
- uzavřete celý řádek do kulatých závorek "(" a ")".

Komentáře ve standardní NC-větě

Uvnitř standardní NC-věty můžete naprogramovat komentář na libovolném místě:

- Naprogramujte středník ";," na místě, kde začíná komentář. Řídicí jednotka interpretuje programový řádek od středníku až do konce řádku jako komentář.
- Uzavřete samostatné komentáře uvnitř standardní NC-věty do kulatých závorek "(" a ")". Tak je také možné programovat "vnořené komentáře", např. když chcete uvnitř programového řádku skrýt sekvenci, ve které je již obsažen jeden komentář s kulatými závorkami.

Kulaté závorky se nesmí používat za funkcemi, pro které může být volitelně naprogramován seznam parametrů s kulatými závorkami (např. G0, G1, G61), přičemž ale tento seznam parametrů nebyl naprogramován.

Příklad:

N10 GO (<text komentáře>)	Neplatné naprogramování!
N20 GO (NI PS) (<text komentáře>)	Platné naprogramování!

- Za funkcí s možným volitelným seznamem parametrů můžete v případě nepoužití seznamu parametrů uvést komentář pomocí znaků "//". Takový komentář se dá alternativně ukončit s použitím znaků "\\".

Základy NC-programování

Komentáře ve větě CPL

Uvnitř vět CPL uvádějte komentáře označením "REM". Řídicí jednotka interpretuje programový řádek od instrukce REM až do konce řádku jako komentář.

Syntaxe:

REM <text komentáře>

Příklad:

```
...
10 REM ***UP pro demaskování stavového slova***
...
```

Příklady komentářů v dílčím programu:

:	Programový kód
; <Komentář>	Řádek komentáře
;N10 <standardní NC-věta>	Skrýt standardní NC-větu
;20 <CPL>	Skrýt větu CPL
(<Komentář>)	Řádek komentáře
(N30 <standardní NC-věta>)	Skrýt standardní NC-větu
(40 <CPL>)	Skrýt větu CPL
N50 G1 X0 Y0 ; <komentář>	Komentář ve standardní NC-větě
N60 G1(IPS) (<komentář>) F1000	Celistvý komentář ve standardní NC-větě
N70 X10 (Y10 (<komentář>))	"Vnořený komentář" ve standardní NC-větě.
N80 G0 //<komentář>	Komentář za funkcí, pro kterou mohou být naprogramovány volitelné parametry
N90 G0 //<komentář>\\ X0 Y0	Celistvý komentář za funkcí, pro kterou mohou být naprogramovány volitelné parametry
100 REM <komentář>	Komentáře ve větě CPL.

2.5.5 Pokyny v uživatelském rozhraní

Odkazové programování se používá pro zobrazení instruktážních textů na NC uživatelském rozhraní (max. 80 znaků). Tak můžete obsluhu stroje např. v průběhu programu

- informovat o momentálním stavu programu nebo
- vydat pokyny pro práci.

Rozlišují se dva typy pokynů:

- **pokyny specifické pro kanál:**
Ty se mažou při zrušení výběru programu nebo v základním nastavení kanálu.
Syntaxe: MSG (<text pokynu>)
- **pokyny přesahující kanály:**
Dají se vymazat celkovým základním nastavením.
Syntaxe: GMSG (<text pokynu>)

Pokyn můžete naprogramovat také proto, abyste vydali obsluze stroje instrukci pro další postup. Přitom naprogramujte ve stejném nebo následujícím řádku např. "M0". Tím zajistíte, že se program bezprostředně po vydání textu pokynu přeruší. Program se znovu rozběhne až po stisknutí "NC-Start" .

Příklad:

:	Programový kód
N60 (MSG Změřte obrobek!)	Vydání pokynu specifického pro kanál
N70 M0	Počkat na tlačítko NC-Start.
:	Programový kód

Z důvodů kompatibility existuje pro odkazové programování několik alternativních variant syntaxe, jejich funkce jsou ovšem ekvivalentní.

- Varianty syntaxe pro pokyny specifické pro kanál:
(MSG <text pokynu>)
(*MSG <text pokynu>)
(MSG, <text pokynu>)
(*MSG, <text pokynu>)
- Varianty syntaxe pro pokyny přesahující kanály:
(GMSG <text pokynu>)
(GMSG, <text pokynu>)

2.5.6 Skoky v průběhu programu

Čím větší jsou programy, tím důležitější je "čisté programování". Pod tím se především rozumí:

- strukturované programování,
- chybová tolerance a
- softwarová ergonomie.

Strukturované programy jsou obecně přehlednější. Seskupení smysluplných úseků nebo často používaných funkcí do (parametrizovaných) podprogramů nebo pod jeden cíl skoku, který je opatřen srozumitelným identifikátorem (Label), vede kromě lepší čitelnosti také k efektivnější práci, protože tyto programy mohou být znovu využity i v jiných programech.

K dispozici jsou následující možnosti:

- **Vyvolání podprogramu**

(viz kapitola 2.6 od strany 2-25)

Podprogramy se používají, když se během obrábění několikrát vyskytuje určitý obráběcí postup, který se provádí stejným nebo podobným způsobem.

Tento úsek naprogramujete pouze jednou (případně s možností předání parametrů), uložíte ho jako program a v případě potřeby jednoduše vyvoláte.

Tím ušetříte programový kód a místo v paměti. Kromě toho budou vaše programy přehlednější a jednodušší na údržbu.

- **Instrukce skoku**

(viz kapitola 2.7 od strany 2-32)

Instrukce skoku se používají, chcete-li v rámci aktuálního programu pokračovat ve zpracovávání programu v závislosti na definovaných podmínkách (např. výsledky výpočtů) na jiných místech.

- **Rozhodovací/rozvětovací instrukce**

(viz kapitola 2.8 od strany 2-38)

Tyto příkazy se používají, když mají být jednotlivé programové věty, různé úseky programu nebo celé podprogramy provedeny v závislosti na určitých podmínkách.

- **Opakovací instrukce**

(viz kapitola 2.9 od strany 2-42)

Opakovací instrukce se používají, když mají úseky programu nebo celé podprogramy probíhat opakovaně. Přitom je zpravidla také možné zadat, kolikrát má opakování proběhnout.

2.6 Podprogramy

Podprogramy (UP) se nazývají programy, které jsou volány voláním podprogramu. Po provedení podprogramu běží volající program dál, počínaje za vyvoláním podprogramu. Jestliže program volá sám sebe, nazývá se to "rekurzivní" volání podprogramu.

Hlavní program (HP) se nazývá program, ze kterého se přeskakuje na první úroveň podprogramů (úroveň UP).

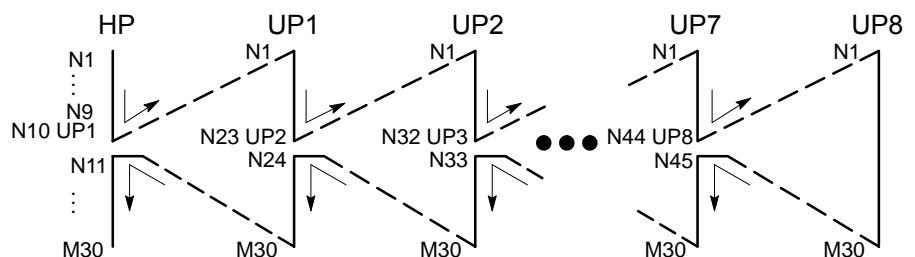
Formálně se hlavní programy a podprogramy neliší, avšak pouze podprogramy mohou být vybaveny předávacími parametry. Platí:

- Podprogramy mohou obsahovat standardní NC-věty a věty CPL.
- Každý dílčí program může být vyvolán jinými programy jako podprogram. Program ovšem nemůže vyvolat sám sebe jako podprogram (není možné rekurzivní vyvolání).
- Volající program může předat podprogramu parametry.
- Maximální hloubka vnoření je 8, tzn. řídicí jednotka dokáže udržovat maximálně 8 současně otevřených úrovní podprogramů.
- Řídicí jednotka rozlišuje také u názvů podprogramů mezi velkými a malými písmeny.

 **Informace o ukončení podprogramu naleznete v kapitole 2.3.2 na straně 2-17.**

Příklad:

Vnoření podprogramů. UPx: Názvy podprogramů



1. úroveň podprogramů 2. úroveň podprogramů 7. úroveň podprogramů 8. úroveň podprogramů

Rozlišují se lokální a modální podprogramy:

- Podprogramy jsou zpravidla **lokální**. V tom případě je podprogram na pozici vyvolání vyvolán jednou.
- Je-li naproti tomu aktivován **modální** podprogram, je znovu vyvolán s každým následujícím naprogramovaným pojezdovým pohybem, dokud není jeho volba opět zrušena. To se využívá např. ve vrtacích cyklech.

IndraMotion MTX nabízí následující varianty pro vyvolání podprogramů:

- Vyvolání s P-adresou a názvem podprogramu ze standardní NC-věty (možno se zadáním cesty).

Základy NC-programování

- Vyvolání pouze pomocí názvu podprogramu bez zadání cesty a bez P-adresy ze standardní NC-věty.
- Podprogram jako uživatelsky definovaná G- nebo M-funkce ve standardní NC-větě.
- Aktivace modálního podprogramu s uživatelsky definovanou syntaxí ze standardní NC-věty.
- Vyvolání pomocí funkce CALL z věty CPL.

Vyvolání podprogramu s P-adresou

- Za adresou P je přímo naprogramován název podprogramu. Může být naprogramován také adresář, ve kterém se podprogram nachází.
- Pro lepší čitelnost může být mezi adresou P a názvem podprogramu naprogramována mezera jako oddělovací znak.
- Vyvolání podprogramu musí být naprogramováno na konci věty. Pojezdové pohyby, které jsou naprogramovány ve stejné větě, jsou provedeny ještě před vyvoláním podprogramu (viz příklad).
- Ve větě smí být naprogramováno nejvýš 1 volání podprogramu.
- Vyvolání podprogramu je lokální (ne modální).

Syntaxe:

P{ <cesta> } <název> { DI N }

kde

<cesta> Adresář, ve kterém se nachází podprogram.

<název> Název volaného programu.

DIN Volitelné. Brání spojení podprogramu.

Tento parametr se používá, pouze když podprogram nevolá žádné věty CPL a žádné další podprogramy. Jinak se v průběhu programu objeví chybové hlášení.

Další informace naleznete v Příkaz CALL, strana 2-29.

Příklad:

:

N40 P Schéma rozložení Vyvolání programu "Schéma rozložení vrtaných
vrtaných otvorů" otvorů".

N50 X100 Po skončení podprogramu následuje věta N50.

:

N140 G0 X10 Y0 PUP1 Nejprve polohování na X10/Y0 v rychloběhu.

N150 Z0 Následně vyvolání programu "UP1".

:

N150. Po skončení podprogramu následuje věta N150.

Základy NC-programování

Vyvolání podprogramu bez P-adresy

Podprogramy mohou být vyvolány také **bez** zadané P-adresy.

- Přímo se programuje pouze název podprogramu.
- Zadání cesty není možné.
- Vyvolání podprogramu musí být naprogramováno na konci věty. Pojezdové pohyby, které jsou naprogramovány ve stejné větě, jsou provedeny ještě před vyvoláním podprogramu (viz příklad).
- Ve větě smí být naprogramováno nejvýš 1 volání podprogramu.
- Vyvolání podprogramu je lokální (ne modální).

Syntaxe:

<Název>

kde

<název> Název volaného programu.

 **Dejte pozor, aby tato programovací varianta nevedla k záměnám s normální syntaxí!**

Používejte proto pro své podprogramy vždy jednoznačné názvy, abyste zabránili chybnému výkladu v překladači řídicí jednotky.

Příklady:

:

N40 XUP

Vyvolání programu "XUP".

N50 X100

Po skončení podprogramu následuje věta N50.

:

N100 X1UP

Pozor! Zde naprogramovaný název podprogramu "X1UP" vede k syntaktické chybě, protože "X1" bude interpretováno jako souřadnice osy s názvem "X" a program s názvem "UP" neexistuje.

:

:

:

:

:

:

N140 G0 X10 Y0 XUP

Nejprve polohování na X10/Y0 v rychloběhu.

N150 Z0

Následné vyvolání programu "XUP".

:

Po skončení podprogramu následuje věta N150.

Základy NC-programování

Uživatelsky definované vyvolání podprogramu s G- a M-kódy

Řízení nabízí možnost kromě již uvedených volání podprogramů

- "nemodální" volání podprogramů pomocí M-adresy (viz MP 3090 00003 a MP 3090 00004), a
- "nemodální" volání podprogramů pomocí G-adresy (viz MP 3090 00001 a MP 3090 00002)

uživatelsky definovat.

Je možné přiřadit podprogramy maximálně pro 16 uživatelsky definovatelných G-kódů a maximálně 8 uživatelsky definovatelných M-kódů; tyto podprogramy jsou při naprogramování příslušného G- nebo M-kódu vyvolány jako lokální (nemodální) podprogramy.

G- a M-kódy zkonfigurované jako volání podprogramů nesmí kolidovat s pevně definovanými G- a M-kódy.

 **Která volání podprogramů jsou zkonfigurována speciálně na vašem stroji, zjistíte u pracovníka odpovědného za systém.**

Při **programování** uživatelsky definovaných volání podprogramů platí:

- V dílčím programu je naprogramován pouze příslušný G- nebo M-kód. Který podprogram je jím volán, vyplývá z konfigurace v parametrech stroje.
- Vyvolání podprogramu musí být naprogramováno na konci věty. Pojezdové pohyby, které jsou naprogramovány ve stejné větě, jsou provedeny ještě před vyvoláním podprogramu.
- Ve větě smí být naprogramováno nejvýš 1 volání podprogramu.
- Vyvolání podprogramu je lokální (ne modální).


Uživatelsky definované volání modálních podprogramů

Modálně účinné podprogramy jsou po prvním vyvolání automaticky provedeny po každém pojezdovém pohybu, který byl předepsán standardní NC-větou. To platí, dokud jejich výběr není zrušen speciální NC-funkcí.

Modální vyvolání podprogramů je zkonfigurováno včetně syntaxe vypínací funkce v parametrech stroje (viz MP 3090 00005 ff.).

Je možné nastavit maximálně 15 modálních podprogramů, které se vzájemně ruší. Volně definovatelným syntaxím jsou přitom přiřazeny názvy podprogramů. Kromě toho je třeba zadat, kolik parametrů se dá maximálně předat do příslušného podprogramu.

Syntaxe modálního volání podprogramů nesmí kolidovat s pevně definovanými NC-funkcemi.

 **Která volání podprogramů jsou zkonfigurována speciálně na vašem stroji, zjistíte u pracovníka odpovědného za systém. Standardně jsou zkonfigurovány vrtací cykly G80, G81-G86 a G184.**

Základy NC-programování

Při **programování** modálního volání podprogramů platí:


- V dílčím programu je naprogramována pouze příslušná zkonfigurovaná syntaxe modálního podprogramu. Který podprogram je jím volán, vyplývá z konfigurace v parametrech stroje.
- Ve větě smí být naprogramováno pouze 1 volání podprogramu.
- Volání podprogramu je modální, tzn. podprogram bude až do zrušení výběru znovu vyvolán po každém naprogramovaném pojezdovém pohybu.

Vyvolání podprogramu v CPL pomocí příkazu CALL

Příkaz CALL nabízí možnost volat podprogramy také z čistých CPL-programů. Při programování platí:

- Podprogram naprogramovaný za příkazem CALL je přímo vyvolán.
- Příkaz CALL musí být naprogramován v samostatné větě CPL.
- Vyvolání podprogramu je lokální.

Za klíčovým slovem CALL je uveden název programu, za nímž mohou následovat předávací parametry v lomených závorkách a na posledním místě identifikace "DIN" (pro ovlivňování procesu Link).

 **Ve větě CPL s instrukcí CALL se nesmí používat znak ":". Následující instrukce CPL musí být naprogramovány v nové větě CPL.**

Příklad:

```
50 IF A% = 1 THEN
%! CALL P999
%" ENDI F
:
```

Ovlivňování procesu Link ("příprava") pomocí identifikace "DIN"

Pokud při vyvolání podprogramu pomocí příkazu CALL naprogramujete jako ukončení identifikaci "DIN", nespojí řízení vyvolaný podprogram. Tak můžete výrazně urychlit např. proces Link hlavního programu, který vyvolává více podprogramů.

Příklad:

```
50 IF A% = 1 THEN
51 CALL P999 DIN           Podprogram "P999" nebude spojen
52 ENDI F
M30
```

Doporučujeme programovat identifikaci "DIN" pouze tehdy, když vyvolaný podprogram

- se skládá výhradně z DIN-vět a
- nevolá žádné další podprogramy.

Základy NC-programování

Pokud podprogram kvůli identifikaci "DIN" nebyl spojen a obsahuje prvky CPL, vydá řízení v průběhu programu odpovídající chybové hlášení.

Alternativně můžete vložit identifikaci "DIN" také **jako komentář** do prvního řádku vyvolaného podprogramu. Řízení pak nespojí program.

Příklad: Identifikace "DIN" ve volajícím programu

```
N10 (DIN)                Podprogram "P999" nebude spojen
N20 . . .
:
```

Předávání parametrů podprogramům

Pomocí **seznamu parametrů CPL** mohou být podprogramu předány parametry. Parametry jsou přitom naprogramovány v jednom ze seznamů uzavřených v lomených závorkách CPL "[" a "]" bezprostředně za voláním podprogramu. Jednotlivé parametry jsou navzájem odděleny čárkami.

Jako parametry jsou přípustné:

- Čísla
- Řetězcové konstanty CPL (v apostrofech: "<řetězcová konstanta CPL>")
- Proměnné CPL
- Aritmetické výrazy CPL

Parametry předané při vyvolání podprogramu jsou v podprogramu vždy volány s použitím proměnných P1, P2, P3 atd. podle pořadí předání parametru.

Parametry mohou být volány také např. s použitím proměnných P1TEST, P2XYZ atd., velká písmena následující po P1, P2 atd. jsou ovšem ignorována (P1 = P1TEST = P1XYZ).

Příklad:

- P1 má v UP P999 hodnotu 2.75,
- P2 má hodnotu proměnné X% v okamžiku předání parametrů
- P3 má hodnotu 0.

Jestliže má P2 i v podprogramu reprezentovat hodnotu INTEGER, dá se to provést připojením znaku % k P2. Toto označení typu proměnné může být odpovídajícím způsobem provedeno také u ostatních typů proměnných.

Hodnota jednotlivých parametrů může být v podprogramu přiřazena dalším proměnným.

Základy NC-programování

Hlavní program:

```
50 IF A% = 1 THEN
51 CALL P999 [2.75, X%, 0]  Vvolání podprogramu s předáním
                             parametrů
52 ENDIF
```

M30

Podprogram P999:

```
1 FAKTOR=P1 : XWERT%=P2% : KORRTAB%=P3%
N1 G1 X[XWERT%*FAKTOR]
N2 G22 K[KORRTAB%]
```

Pokud podprogram

- má být vyvolán s konstantami String jako předávacími parametry a
 - volající program je zvolen bez spojení,
- musí se použít příkaz PDIM.

Syntaxe:

PDIM <název parametru>(<velikost pole>)

Pokud není naprogramována žádná nebo příliš malá velikost pole, hlásí řízení chybu dílčího programu "Unzulässige Variable".

Příklad:

Hlavní program:

```
N10 (DIN)
:
N50 P UP["Test"]
M30
```

Podprogram:

```
10 PDIM P1$(4)
M30
```


Proměnná String P1\$ má hodnotu "TEST".


2.7 Programování návěští a instrukce skoku

S použitím instrukcí skoku je možné pokračovat v průběhu programu v určitých vstupních bodech.

IndraMotion MTX nabízí s ohledem na instrukce skoku následující funkce:

- Programování návěští pro standardní NC-věty (DIN)
- Programování návěští pro věty CPL
- GoAhead (GOA): Skok dopředu na standardní NC-větu
- GoBack (GOB): Skok dozadu na standardní NC-větu
- GoCond (GOC): Podmíněný skok na standardní NC-větu
- GoTo: Nepodmíněný skok na standardní NC-větu
- Skok CPL (GOTO): Skok na libovolnou programovou větu

 **Skok CPL GOTO se spouští již v okamžiku spojení. U standardních NC-příkazů se naproti tomu hledá cíl skoku až v průběhu programu, což může mít nepříznivé důsledky pro chování programu, obzvláště v případě, že cíl skoku leží daleko od místa vyvolání skoku.**

 **Pomocí standardních NC-příkazů skoku GoAhead, GoBack, GoCond a GoTo se nesmí skákat do opakovací instrukce CPL nebo rozvětovací instrukce CPL, resp. z těchto instrukcí CPL vyskakovat. To platí pro instrukce CPL:**

- REPEAT – UNTIL
- WHILE – DO – END
- FOR – STEP – TO – NEXT
- IF – THEN – ELSE – ENDIF
- CASE – LABEL ... LABEL – OTHERWISE – ENDCASE.

Návěští ve standardních NC-větách a větách CPL

Návěští je značka pro příkaz skoku. IndraMotion MTX rozlišuje mezi návěštími ve standardní NC-větě (DIN) a ve větě CPL.

Programování návěští ve standardní NC-větě (DIN):

- Cíl skoku musí být vždy naprogramován přímo na začátku věty.
- U vět s číslem věty je cíl skoku uveden přímo za číslem věty, oddělený mezerou.
- Název návěští se smí skládat ze 2 až 32 znaků. Jsou povolena písmena, podtržítka a číslice, přičemž první dva znaky nesmí být číslice. Rozlišují se velká a malá písmena.
- U cíle skoku musí být za názvem návěští naprogramována dvojtečka.

Programování návěští ve větě CPL:

- Cíl skoku se programuje přímo za číslem věty, oddělený mezerou.

Základy NC-programování

- Název návěští se skládá z desetinné tečky a následujících znaků ASCII, začínajících velkým písmenem.
- Návěštím nesmí být proměnná.

GoAhead (GOA)**Skok dopředu na standardní NC-větu**

Průběh programu pokračuje bez podmínek v cíli skoku (návěští). Přitom platí:

- Cíl skoku musí být definován.
- Cíl skoku se musí nacházet ve směru konce souboru, vzhledem k aktuální programové větě.

Potřebné programování návěští viz strana 2-32.

Syntaxe:

GoAhead <návěští>


Zkratka: **GOA**

kde

<Návěští> Název cíle skoku.
2 až 32 znaků. Jsou dovolena písmena, podtržítka a číslice, přičemž první 2 znaky nesmí být číslice.

Příklad:

:		
N40	GoAhead LABEL1	Skok dopředu na cíl skoku "LABEL1".
:		
N80	LABEL1:	Programování návěští cíle skoku "LABEL1".
:		

 **Pomocí standardního NC-příkazu skoku GoAhead se nesmí skákat do opakovací instrukce CPL nebo rozvětovací instrukce CPL, resp. z těchto instrukcí CPL vyskakovat. Příslušné instrukce CPL viz strana 2-32.**

Základy NC-programování

GoBack (GOB)**Skok dozadu na standardní NC-větu**

Průběh programu pokračuje bez podmínek v cíli skoku (návěští). Přitom platí:

- Cíl skoku musí být definován.
- Cíl skoku se musí nacházet ve směru začátku souboru, vzhledem k aktuální programové větě.

Potřebné programování návěstí viz strana 2-32.

Syntaxe:

GoBack <návěští>

Zkratka: **GOB**


kde

<Návěští> Název cíle skoku.
2 až 32 znaků. Jsou povolena písmena, podtržítka a číslice, přičemž první 2 znaky nesmí být číslice.

 **Pamatujte si, že při programování skoků dozadu mohou snadno vzniknout nežádoucí nekonečné smyčky!**

Příklad:

:	
N40 LABEL1:	Programování návěstí cíle skoku "LABEL1".
:	
N80 GoBack LABEL1	Skok dozadu na cíl skoku "LABEL1".
:	
	Pokud mezi N40 a N80 není naprogramován žádný další příkaz skoku, běží program mezi N40 a N80 v nekonečné smyčce!

 **Pomocí standardního NC-příkazu skoku GoBack se nesmí skákat do opakovací instrukce CPL nebo rozvětvací instrukce CPL, resp. z těchto instrukcí CPL vyskakovat. Příslušné instrukce CPL viz strana 2-32.**

Základy NC-programování

GoCond (GOC)**Podmíněný skok na standardní NC-větu**

Průběh programu pokračuje na uvedeném čísle věty, pokud byl na rozhraní kanálu v okamžiku přípravy věty aktivní vstupní signál "Podmíněný skok". Přitom platí:

- Uvedené číslo věty musí existovat.
- Poloha uvedeného čísla věty uvnitř programového souboru je vzhledem k aktuální programové větě libovolná.

Syntaxe:

GoCond N<číslo>

Zkratka: **GOC**

kde

<Číslo> Číslo věty.

- Číslo ve formátu Integer nebo Real max. s 15 číslicemi.
- Pokud je číslo cílové věty naprogramováno s nulami na začátku (např. "N0020 . . ."), musí zde být rovněž naprogramovány nuly na začátku ("GoCond N0020").

Zvláštnosti a omezení:


- Kromě případně naprogramovaného čísla věty nejsou ve stejné větě dovolena žádná další programová slova.
- Nejsou brány v úvahu změny signálu rozhraní kanálu "Podmíněný skok" během časového intervalu mezi přípravou věty a jejím provedením.
Je-li takové chování pro vaši aplikaci nepřijatelné, musíte naprogramovat v předchozím programovém řádku funkci "WAIT".

 **Pamatujte si, že při skocích ve směru k začátku souboru mohou snadno vzniknout nežádoucí nekonečné smyčky!**

Příklad:

```
10 WAIT
N20 GoCond N090
:
:
:
:
N090 . . .
:
```

Zastavení přípravy věty, dokud nejsou zpracovány všechny věty před N20.
Následuje skok na větu N090, pokud je v okamžiku přípravy věty N20 aktivní signál rozhraní "Podmíněný skok".

 **Pomocí standardního NC-příkazu skoku GoCond se nesmí skákat do opakovací instrukce CPL nebo rozvětvací instrukce CPL, resp. z těchto instrukcí CPL vyskakovat. Příslušné instrukce CPL viz strana 2-32.**

Základy NC-programování

GoTo**Nepodmíněný skok na standardní NC-větu**

Průběh programu pokračuje bez podmínek na libovolném čísle věty. Přitom platí:

- Uvedené číslo věty musí existovat.
- Poloha uvedeného čísla věty uvnitř programového souboru je vzhledem k aktuální programové větě libovolná.

Syntaxe:

GoTo N<číslo>

kde

<Číslo> Číslo věty.

- Číslo ve formátu Integer nebo Real max. s 15 číslicemi.
- Pokud je číslo cílové věty naprogramováno s nulami na začátku (např. "N0020 . . ."), musí zde být rovněž naprogramovány nuly na začátku ("GoTo N0020").


Zvláštnosti a omezení:

- Kromě případně naprogramovaného čísla věty nejsou ve stejné větě dovolena žádná další programová slova.

 **Pamatujte si, že při skocích ve směru k začátku souboru mohou snadno vzniknout nežádoucí nekonečné smyčky!**

Příklad:

```
:  
N40 GoTo N080           Skok dopředu na větu N080.  
:  
N080 GoTo N40           Skok dozadu na větu N40.  
:  
                        Pokud mezi N40 a N080 není naprogramován žádný další příkaz skoku, běží program mezi N40 a N080 v nekonečné smyčce!
```

 **Pomocí standardního NC-příkazu skoku GoTo se nesmí skákat do opakovací instrukce CPL nebo rozvětovací instrukce CPL, resp. z těchto instrukcí CPL vyskakovat. Příslušné instrukce CPL viz strana 2-32.**

Základy NC-programování

Skok CPL (GOTO)**Skok na libovolnou programovou větu**

Průběh programu pokračuje bez podmínek v cíli skoku. Přitom platí:

- Jako cíl skoku může být zadáno číslo věty CPL, číslo standardní NC-věty nebo "návěští" (značka skoku).
- Poloha cíle skoku uvnitř programového souboru je vzhledem k aktuální programové větě libovolná.

Syntaxe:

GOTO < cíl >

Příklad:

10 GOTO N20	Skok na větu N20
N20 X100	
30 GOTO 120	Skok na větu CPL 120
...	
120 GOTO .ZI EL1	Skok na návěští .ZIEL1
...	
150 .ZI EL1	

Podmínky programování návěští viz strana 2-32.

2.8 Rozhodovací a rozvětovací instrukce

Rozhodovací a rozvětovací instrukce slouží k tomu, aby byly jednotlivé programové věty a úseky programu nebo kompletní podprogramy prováděny v závislosti na určitých událostech.

IndraMotion MTX nabízí k tomuto účelu následující možnosti:

- Funkce "vynechání věty" pro standardní NC-věty (DIN)
- Instrukce CPL IF-THEN-ELSE-ENDIF
- Instrukce CPL CASE-LABEL...LABEL-OTHERWISE-ENDCASE


Funkce "vynechání věty"

Pomocí této funkce mohou být jednotlivé standardní NC-věty (DIN) vynechány řízením. Přitom musíte naprogramovat na začátku příslušných programových řádků znak "/".

Označené programové věty jsou vynechány, pouze když je aktivován signál rozhraní qCh_BlockSlash (vynechat větu) pro bitové rozhraní příslušného kanálu.

Příklad:

```
:                               Signál rozhraní "Vynechat větu" je aktivován.  
:  
/N100 ...                       Věta N100 bude ignorována.  
:                               Signál rozhraní "Vynechat větu" je deaktivován.  
:  
/N300 ...                       Věta N300 bude zpracována.  
:
```

 **Funkce "vynechání věty" se dá používat jen pro standardní NC-věty (DIN)!**

Základy NC-programování

Instrukce CPL: IF-THEN-ELSE-ENDIF

Tato funkce je jednoduchou podmíněnou rozvětvovací instrukcí:
"Když je (IF) splněna určitá podmínka, pak (THEN) je provedena určitá procedura, jinak (ELSE) je provedena jiná procedura!".

Syntaxe:

```
IF <podmínka> THEN <procedura> [ELSE <alternativní procedura>]
ENDIF
```

Přitom platí:


- Podmínka je uvedena v tomtéž řádku jako příkaz "IF" a je ukončena příkazem "THEN" na stejném řádku.
- Procedura THEN a procedura ELSE jsou větve programu, které nemusí za každou cenu proběhnout.
- Jestliže vynecháte větev ELSE, pokračuje program při nesplnění podmínky rovnou na instrukci ENDIF.

Podobně jako v podmínkách přerušení u smyčkových instrukcí mohou být v podmínce příkazu IF používány aritmetické, trigonometrické a logické operace. I zde je možné vnořování.

Příkaz IF musí být vždy ukončen instrukcí ENDIF, protože jinak není rozpoznán konec procedury nebo alternativní procedury. Protože umístění instrukce ENDIF závisí na logice průběhu programu, nemůže řízení po každé jednoznačně rozpoznat chybějící instrukci ENDIF. Následkem jsou pak zavádějící chybová hlášení. Programátor tedy musí zásadně kontrolovat úplnost příkazu IF.

Příklad:

```
...
10 X = 1
20 . START
30 IF X>=100 THEN
40     GOTO . ENDE
50     ELSE X=X+2. 75
60     GOTO . START
70 ENDI F
...
90 ENDE
...
```

 **Pomocí standardních NC-příkazů skoku GoAhead, GoBack, GoCond a GoTo se nesmí skákat do instrukce IF-THEN-ELSE-ENDIF ani z ní vyskakovat.**

Základy NC-programování

Instrukce CPL: CASE-LABEL...LABEL-OTHERWISE-ENDCASE

Uvnitř programu je často nezbytné dotazovat se na **více než 2 stavy** výrazu Integer nebo proměnné Integer. Dotazování pomocí instrukce IF je v takových případech možné pouze pomocí několika do sebe vnořených příkazů IF. To stojí operační čas navíc a zhoršuje čitelnost a udržovatelnost programu.

Tyto nevýhody můžete odstranit s použitím struktury CASE:

CASE <výraz Integer> **OF**

```

LABEL <konstanta Int.>[, <konstanta Int.>]    [: <instrukce>]
    <Instrukce>
    :
LABEL ...
    :
[ OTHERWISE <instrukce>
    <Instrukce>
    : ]

```

ENDCASE

Průběh programu se rozvětňuje za instrukcí CASE na tu instrukci LABEL, u které je jedna z <konstant Int.> stejná jako hodnota <výrazu Integer>. Nyní jsou provedeny všechny instrukce až po další instrukci LABEL nebo OTHERWISE. Potom program odbočí rovnou k instrukci ENDCASE.

Pokud neexistuje žádná instrukce LABEL, která by splňovala tuto podmínku, odbočí program na instrukci OTHERWISE nebo (pokud větev OTHERWISE nebyla naprogramována) přejde přímo na instrukci ENDCASE.

V oblasti <instrukce> struktury CASE mohou být používány všechny instrukce CPL. Je možné vnoření max. 10 struktur CASE.

Příklady:

```


10 CASE A% OF
20 LABEL 0 : Y=1
30 LABEL 2
40           Y=Y*Y
50 LABEL 4 : Z=Y*Y
60           Y=Z*Z
70 OTHERWI SE Y=0
80 ENDCASE

```

Základy NC-programování

```
10 CASE (INT(X/Y)+C%) OF
20 LABEL 1, 2 : X=1 : Y=2
30 LABEL 4, 8
40           X=2 : Y=4
50 LABEL 0
60           X=0 : Y=1
70 OTHERWISE X=0 : Y=0
80 ENDCASE
```

```
10 CASE INTFELD%(1, 2) OF
20 LABEL 1, 2, 3 : GOTO . MARKE1
30 LABEL 4, 5, 6 : GOTO . MARKE2
40 OTHERWISE GOTO . ENDE
50 ENDCASE
```

-  **Pomocí standardních NC-příkazů skoku GoAhead, GoBack, GoCond a GoTo se nesmí skákat do instrukce CASE-LABEL...LABEL-OTHERWISE-ENDCASE ani z ní vyskakovat.**

2.9 Opakovací instrukce

Chcete-li zpracovat jednu nebo několik programových vět v závislosti na určitých podmínkách opakovaně, existuje možnost naprogramovat to s použitím **opakovacích instrukcí CPL**. Vícenásobný průchod programu se označuje také jako smyčka.

IndraMotion MTX nabízí k tomuto účelu následující možnosti:

- Instrukce CPL FOR-STEP-TO_NEXT
- Instrukce CPL REPEAT-UNTIL
- Instrukce CPL WHILE_DO_END

Instrukce CPL: FOR-STEP-TO-NEXT

Pokud má podmínka přerušení pro opakovací instrukci vyplývat přímo ze zpracování procedury, je např. zapotřebí připojené počítadlo. To nemusí být u smyčky FOR-NEXT nijak zvlášť naprogramované. Je stanovena číselná proměnná (INTEGER) a musí být zadán její počáteční a koncový stav na počítadle. Je-li velikost kroku počítadla různá od 1, je možné určit tuto velikost kroku (STEP) samostatně.

Syntaxe:

```
FOR <číselná prom.>=<počáteční hodnota> [STEP <velikost kroku>] TO <koncová hodnota><procedura>
NEXT [<číselná proměnná>]
```

Příklad:

```
10 FOR I%=0 TO 18
20 XSINUS(I%)=SIN(I%*10)
30 NEXT I%
```

Po skončení smyčky má číselná proměnná hodnotu, která je větší než koncová hodnota (max. velikost kroku).

Zde se do pole XSINUS zapisují hodnoty funkce sinus pro 0 až 180 stupňů. V řádku 30 je k příkazu "NEXT" připojeno "I %"; to slouží pouze ke zvýraznění a dá se případně vynechat.

Smyčky FOR-NEXT mohou být naprogramovány také s **variabilní velikostí kroku**. Proměnná velikosti kroku pak musí mít stejný typ jako číselná proměnná.

Příklad:

```
10 OPENW(1, "P222", 130)
20 SCHRI TT%=2 : ANFANG%=1 : ENDE%=3500 : NJUST
30 FOR ZAEHLER%=ANFANG% STEP SCHRI TT% TO ENDE%
40   SCHRI TT%=ROUND(SCHRI TT%*SQRT(SCHRI TT%))
50   PRN#(1, "POČÍTADLO: ",ZAEHLER%,"VELIKOST KROKU: "
      , SCHRI TT%)
60 NEXT
70 CLOSE(1)
```

Základy NC-programování

Po proběhnutí tohoto programu je v souboru "P222":

POČÍTADLO: 1	VELIKOST KROKU: 3
POČÍTADLO: 4	VELIKOST KROKU: 5
POČÍTADLO: 9	VELIKOST KROKU: 11
POČÍTADLO: 20	VELIKOST KROKU: 36
POČÍTADLO: 56	VELIKOST KROKU: 216
POČÍTADLO: 272	VELIKOST KROKU: 3175
POČÍTADLO: 3447	VELIKOST KROKU: 178902

 **Pomocí standardních NC-příkazů skoku GoAhead, GoBack, GoCond a GoTo se nesmí skákat do instrukce FOR-STEP-TO-NEXT ani z ní vyskakovat.**

Instrukce CPL: REPEAT-UNTIL

Má-li být podmínka přerušení pro opakovací instrukci zkoumána až po prvním zpracování procedury, je možné použít smyčku REPEAT.

Syntaxe:

REPEAT <procedura> **UNTIL** <podmínka>

Příklad:

```
:  
30 REPEAT  
40 X=X+1  
50 UNTIL X=100  
:
```

Smyčka do X = 100

 **Pomocí standardních NC-příkazů skoku GoAhead, GoBack, GoCond a GoTo se nesmí skákat do instrukce REPEAT-UNTIL ani z ní vyskakovat.**

Základy NC-programování

Instrukce CPL: WHILE-DO-END

Pokud má být podmínka přerušení pro opakovací instrukci dotazována před prvním proběhnutím smyčky, můžeme to zformulovat takto: "Dokud (☞ *angl.* while) je splněna podmínka, provádí se (☞ *angl.* to do) procedura!". Smyčka WHILE je strukturována následujícím způsobem:

Syntaxe:

WHILE <podmínka> **DO** <procedura> **END**

Příklad:

```
:  
30 WHILE SD(9) = 0           Čekací smyčka, až SD(9) nabude hodnoty 0  
40 I = I + 1  
50 END  
:
```

☞ **Pomocí standardních NC-příkazů skoku GoAhead, GoBack, GoCond a GoTo se nesmí skákat do instrukce WHILE-DO-END ani z ní vyskakovat.**

2.10 Programování proměnných

2.10.1 Názvy proměnných

Programování proměnných je součástí jazyka CPL!

Programování proměnných v CPL slouží k tomu, aby byly vytvářené programy parametrizovatelné a průběh programu tak mohl být přizpůsoben aktuálnímu okolnostem.

- Proměnné jsou libovolné symbolické názvy, pro které však platí některé zvláštní okrajové podmínky:
 - Názvy proměnných musí být jednoznačné.
 - Názvy proměnných nesmí být stejné jako rezervovaná příkazová slova CPL.
Díky formálnímu oddělení mezi standardním NC-programováním a programováním CPL však mohou názvy proměnných teoreticky souhlasit s názvy NC-funkcí nebo NC funkčních parametrů, např. je možné definovat proměnnou 'X', ačkoli v systému zároveň existuje osa s označením 'X'.
 - Název proměnné se skládá z libovolné posloupnosti velkých písmen a číslic, přičemž prvním znakem musí být velké písmeno.

Pouze prvních 8 znaků názvu proměnné je významných, tzn. jen prvních 8 znaků se používá k rozlišení názvu.

- Existují celkem 3 skupiny proměnných, které určují jejich obor platnosti. Skupina proměnných je dána označením na začátku názvu proměnné. Tento znak se vždy počítá mezi významná místa názvu! Existují následující **skupiny proměnných** s příslušným označením:
 - Lokální proměnné: žádné speciální označení
 - Globální proměnné: '#'
 - Permanentní proměnné: '@'
- **Typ proměnné** se vždy určuje označením na konci názvu proměnné. To platí, i když název proměnné překročí počet významných míst. Existují následující typy proměnných s příslušným označením:
 - INTEGER: '%'
 - DOUBLE: '!'
 - BOOLEAN: '?'
 - CHARACTER: '\$'
 - REAL: žádné zvláštní označení

Základy NC-programování

Příklady lokálních, globálních a permanentních proměnných:

10 ANZAHL1% = 1	lokální proměnná INTEGER
20 #ANZAHL2% = 2	globální proměnná INTEGER
30 @36% = 3	permanentní proměnná INTEGER
40 @ABCD% = 4	definovaná permanentní proměnná INTE- GER

2.10.2 Skupiny proměnných


Kvůli možnosti používat podprogramy a eventuální nutnosti dočasně uchovávat hodnoty proměnných nezávisle na příslušném programu jsou zapotřebí ujednání o oboru působnosti proměnných. Přitom rozlišujeme mezi následujícími skupinami proměnných:

Lokální proměnné

působí jen uvnitř programu, ve kterém jsou dohodnuty. Po skončení programu jsou tyto proměnné vymazány a obsazené paměťové místo se uvolní. Při vyvolání podprogramu je název lokální proměnné volajícího programu "neviditelný" pro podprogram. Název proměnné tam tedy může být taktéž lokálně dohodnut, aniž by se obě proměnné navzájem ovlivňovaly. Při návratu do volajícího programu je opět k dispozici původní lokální proměnná s hodnotou, která byla obsazena bezprostředně před vyvoláním podprogramu.

Globální proměnné

jsou označeny znakem # na začátku. Když je globální proměnné poprvé přiřazena hodnota, mohou ji po zbytek doby trvání celého programu číst nebo měnit všechny součásti programu. Po skončení programu jsou globální proměnné vymazány.

-  **Platnost globálních proměnných je vždy omezena na jednotku spojení!**
(Vysvětlení k jednotkám spojení naleznete v kapitole 2.2 Spojování NC-programů, na straně 2-6 "Jednotky spojení").

Základy NC-programování

Permanentní proměnné

jsou označeny znakem @ na začátku, za nímž následuje název proměnné. Mohou být používány každým aktivním programem. Proměnné zůstávají i po skončení programu trvale zachovány. Vymazání je možné jedině cíleným přepsáním. Permanentní proměnné jsou uloženy v samostatné oblasti paměti. Vymazání celé paměti tedy na permanentní proměnné neúčinkuje.

Pod označením @1 až @100 je možné používat permanentní proměnné typu INTEGER (význam typu INTEGER naleznete v typech proměnných na straně 2-52). Pro lepší čitelnost programu je možné doplnit označení takových permanentních proměnných připojením písmen k číslům.

Kromě toho je možné používat permanentní jednorozměrné pole proměnných @_R se 100 prvky typu "Double". Dvě permanentní proměnné @_RES_DOUBLE a @_RES_DWORD jsou vyhrazeny pro interní používání a nesmí se používat.

Definovatelné permanentní proměnné

jsou rovněž označeny znakem @ na začátku, za nímž následuje název proměnné.

Rozdíl oproti "permanentním proměnným" jsou:

1. Nejsou automatickou součástí systémového softwaru, ale musí být **ručně** jednotlivě **deklarovány** v souborech "wmhperm.dat" (pro WMH-specifická data) a "anwperm.dat" (pro data koncových uživatelů). Syntaxi pro deklaraci najdete ve struktuře souborů "wmhperm.dat" a "anwperm.dat".

Při spuštění hledá řízení tyto soubory nejprve v kořenovém adresáři, pak v uživatelské FEPRM a potom ve FEPRM.

První nalezený soubor od každého názvu pak řízení vyhodnotí a vytvoří ze zde existujících záznamů "definovatelné permanentní proměnné", pokud ještě neexistují. Stávající "definovatelné permanentní proměnné", které nejsou deklarovány v žádném z obou souborů, jsou vymazány.

Maximální možný počet definovatelných permanentních proměnných je omezen velikostí dostupného paměťového místa. Pokud již není volné žádné paměťové místo pro vytváření proměnných, vydá Typ3 osa/PNC odpovídající chybové hlášení.

2. Názvy "definovatelných permanentních proměnných" začínají vždy **znakem @ a znakovým řetězcem**. Tento znakový řetězec se skládá z velkého písmena, za nímž následují libovolná velká písmena nebo číslice.

U "definovatelných permanentních proměnných" je významných prvních **16 míst** názvu proměnné. Jestliže se liší až od 17-tého místa, interpretuje je CPL jako jedinou proměnnou!

Základy NC-programování

3. Definovatelné permanentní proměnné smí být typu **INTEGER, REAL, DOUBLE, BOOLEAN** nebo **CHARACTER**.

Typ proměnné se určuje označením na konci názvu proměnné. Toto označení musí být uvedeno v dílčím programu:

@ABCD% def. perm. proměnná typu INTEGER
 @EFGH def. perm. proměnná typu REAL (bez %, !, \$ nebo ?)
 @IJKL! def. perm. proměnná typu DOUBLE
 @MNOP? def. perm. proměnná typu BOOLEAN
 @QRST\$ def. perm. proměnná typu CHARACTER

4. Smí se používat jedno- a dvojrozměrná **pole**. Maximální **index pole** činí u polí proměnných typu INTEGER, REAL, DOUBLE nebo BOOLEAN **65535**. U polí proměnných typu CHARACTER **1024**.

Příklady:

@WZNR(1) = 4 První proměnné (s indexem 1) jedno-rozměrného pole @WZNR typu INTEGER je přiřazena hodnota 4.
 @WZKOR(2, 2) = 0.2 Proměnné (s indexy 2,2) ve dvojrozměrném poli @WZKOR typu REAL je přiřazena hodnota 0.2.

5. Hodnocení **počtu** nově definovatelných permanentních proměnných, které jsou k dispozici:

- Celkové paměové místo pro permanentní proměnné:
100 kByte (102400 Byte)

Č.	Rezervováno pro	Paměové místo v bajtech	Poznámka
1	všechny permanentní proměnné	102400	Celková pamě
z toho je rezervováno pro			
2	@1 - @100 (permanentní proměnné)	800	
3	Administrativní informace	24	
4	všechny definovatelné permanentní proměnné	101576	(4) = (1) - (2) - (3)

Základy NC-programování

Č.	Rezervováno pro	Paměové místo v bajtech	Poznámka
4	všechny definovatelné permanentní proměnné	101576	(4) = (1) - (2) - (3)
z toho je rezervováno pro			
5	@_R	823	Permanentní pole proměnných se 100 prvky typu DOUBLE
6	@_RES_DOUBLE	40	Permanentní proměnná typu DOUBLE, rezervovaná pro interní použití
7	@_RES_DWORD	35	Permanentní proměnná typu INTEGER, rezervovaná pro interní použití
8	nově definovatelné permanentní proměnné	100678	(8) = (4) - (5) - (6) - (7)

Každá definovatelná permanentní proměnná zabírá následující paměové místo:

Č.	Rezervováno pro	Paměové místo v bajtech	Poznámka
9	název definovatelné permanentní proměnné	max. 16	1 Byte na znak
10	hodnotu definovatelné permanentní proměnné	1, 4 nebo 8	Integer: 4 Byte Double: 8 Byte Real: 4 Byte Boolean: 1 Byte
11	Administrativní informace	20	
12	definovatelnou permanentní proměnnou typu DOUBLE s délkou názvu 16 znaků	44	např.: maximální obsazení paměti (9) + (10) + (11)

Základy NC-programování

Počet "definovatelných permanentních proměnných" typu DOUBLE a INTEGER:

Typ proměnné	Počet proměnných	Poznámka
Typ DOUBLE s délkou názvu max. 16 znaků	2288	100678/44=2288
Typ INTEGER s délkou názvu max. 16 znaků	2516	100678/(16+4+20)=2516
Typ INTEGER s délkou názvu max. 8 znaků	3146	100678/(8+4+20)=3146
Pole proměnných s délkou názvu max. 16 znaků, typ INTEGER	25160	(100678-16-20)/4=25160
Pole proměnných s délkou názvu max. 16 znaků, typ DOUBLE	12580	(100678-16-20)/8=12580

Struktura souborů "wmhperm.dat" a "anwperm.dat":

Soubory smí obsahovat pouze deklarace "definovatelných permanentních proměnných". Každá deklarace se provádí v samostatném řádku a ukončuje klávesou Return.

Deklační řádek má vždy následující strukturu:

DEF <typ proměnné> @<název proměnné>; [<komentář>]

Příklady "wmhperm.dat" a "anwperm.dat":

```
DEF INT @ABCD           ;jednoduchá proměnná INTEGER
DEF REAL @EFGH          ;jednoduchá proměnná REAL
DEF DOUBLE @IJKL        ;jednoduchá proměnná DOUBLE
DEF BOOL @MNOP          ;jednoduchá proměnná BOOLEAN
DEF CHAR @PSTR1(3)      ;proměnná CHARACTER s délkou 3
DEF INT @WZNR(9)        ;1-rozměrné pole INTEGER s 9 proměnnými
DEF INT @WZKOR(9, 2)    ;2-rozměrné pole REAL s 18 proměnnými
DEF CHAR @PSTR2(9, 2)   ;2-rozměrné pole CHARACTER s 9 dílčími
                        ;stringy po 2 znacích
```

Příklady použití permanentních proměnných:

```
10 @1 = 1
15 @2_ZAEHLER = 2
20 @ABCD% = 3
25 @EFGH = 4. 1
30 @IJKL! = 5. 12345
35 @MNOP? = TRUE
40 @PSTR1$ = "ABC"
45 @WZNR%(2) = 6
50 @WZKOR(3, 2) = 7. 6
55 @PSTR2$(3) = "DE"
```

Základy NC-programování

Strukturované proměnné

mohou být nahrazeny systémovými daty (SD), ale z důvodů kompatibility zůstávají zachovány (viz kapitola 4.8.2 Systémová data strukturovaných typů).

Strukturované proměnné jsou označeny různými úrovněmi struktury, které se při zapisování oddělují tečkou ("."). Strukturované proměnné začínají vždy "SV."


Všechny strukturované proměnné, které mají být použity v instrukci CPL, musí být **ručně** jednotlivě **deklarovány** v souboru "machdef.dat". Tento soubor se při spuštění NC-řízení vyhodnocuje a vytvářejí se administrativní data a pamě pro uživatelská data strukturovaných proměnných. Při každém spuštění se znovu vytváří pamě a všechny hodnoty jsou obsazeny 0.

Soubor "machdef.dat" slouží pouze k definování strukturovaných proměnných. Každá definice je uvedena v samostatném řádku a ukončena středníkem.

Definiční řádek má vždy následující strukturu:

DEF <prostor názvu> : <typ proměnné> **SV.**<název proměnné>;
[<komentář>]

<Prostor názvu>, <typ proměnné> a <název proměnné> jsou přitom stringy, které se mohou skládat z malých nebo velkých písmen, číslic, pomlčky a podtržítka.

 **U všech názvů proměnných je významných prvních 16 míst názvu proměnné. Jestliže se liší až od 17-tého místa, interpretuje je CPL jako jedinou proměnnou!**

Příklad:

```
DEF BRCSystem: DbHeader_t SV. A;      Definice strukturované
                                     proměnné typu DbHeader_t
```

Instrukce CPL mohou vyžadovat přístup jak k SV.A, tak i ke všem složkám struktury SV.A. Různé složky struktury jsou navzájem oddělené znakem ".".

Příklad:

```
10 SV. A=DBSEA("/dbt 1/Rec", - 1, - 1, "Key1=1", I%)
```

2.10.3 Typy proměnných

Celočíselná proměnná (INTEGER)

Proměnná INTEGER potřebuje 32 bitů paměti. Je označena znakem "%" připojeným za názvem proměnné. Rozsah hodnot sahá od -2.147.483.647 do +2.147.483.647.

```
10 ANZAHL% = 4
    |
    |_____ Proměnná INTEGER
```

Proměnná s pohyblivou řádovou čárkou (REAL)

Pokud za názvem proměnné není uvedeno žádné zvláštní označení, je proměnná interpretována jako proměnná REAL s jednoduchou přesností.

V takovém případě potřebuje proměnná 32 bitů paměti. Rozsah hodnot činí +/-10³⁸. To odpovídá 7 platným místům.

```
10 PI = 3. 141593
    |
    |_____ Proměnná REAL s jednoduchou přesností
```

Proměnná s pohyblivou řádovou čárkou (DOUBLE)

Je-li za názvem proměnné uveden znak "!", je proměnná interpretována jako proměnná REAL s dvojnásobnou přesností.

V takovém případě potřebuje proměnná 64 bitů paměti. Rozsah hodnot činí +/- 10³⁰⁸. To odpovídá 15 platným místům.

```
10 PI! = 3. 141592653589793
    |
    |_____ Proměnná REAL s dvojnásobnou přesností
```

Logická proměnná (BOOLEAN)

Označení se provádí znakem "?" za názvem proměnné. Logické proměnné (BOOL) mohou nabývat pouze hodnoty **TRUE** (= pravda) nebo **FALSE** (= nepravda). Ukládají se v nich logické stavy nebo podmínky, které jsou zapotřebí v dalším průběhu programu.

```
10 START? = FALSE
    |
    |_____ Proměnná BOOLEAN
```

Základy NC-programování

Pole proměnných (ARRAY)

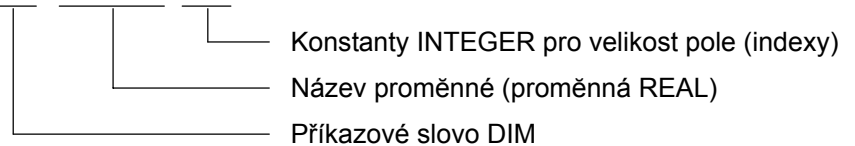
Při použití proměnných ARRAY je možné rezervovat v paměťové oblasti pod jediným názvem proměnné jedno- nebo dvojrozměrné pole (Array), které se skládá z více proměnných téhož typu.

Definice polí jsou možné pro proměnné typu INTEGER, REAL, DOUBLE, BOOLEAN a CHARACTER. Aby bylo možné získat přístup k jednotlivým prvkům pole, zadává se kromě názvu pole proměnných také jejich index, resp. indexy.

Příklad:

Dimenzování proměnné ARRAY

```
10 DIM FELDVAR(2, 3)
```



Příklad:


Přístup k proměnné Array

```
100 FELDVAR(1, 1) = MPOS(1)
110 FELDVAR(2, 1) = CPOS(1)
120 FELDVAR(1, 2) = MPOS(2)
130 FELDVAR(2, 2) = CPOS(2)
140 FELDVAR(1, 3) = MPOS(3)
150 FELDVAR(2, 3) = CPOS(3)
```

Rozsahy indexů, resp. velikost pole musí být před prvním použitím pole proměnných dimenzovány s použitím konstant INTEGER:

- Velikost pole proměnných typu INTEGER a REAL:
max. 65536
- Velikost pole proměnných typu CHARACTER:
max. 1024

DIM <název proměnné>(<velikost pole1>[,<velikost pole2>])

 **Dimenzování pomocí DIM se nesmí používat na "definovatelné permanentní proměnné". Místo toho se provádí dimenzování těchto proměnných v souboru wmhperm.dat, resp. anwpwerm.dat.**

Základy NC-programování

Proměnné CHARACTER a STRING

Proměnná CHARACTER je označena znakem "\$" za názvem. V proměnných tohoto typu můžete uložit jediný znak, ale také úplný znakový řetězec (String).

Instrukce pro znakové řetězce (viz odstavec "Zpracování znakových řetězců") jsou ovšem možné jen tehdy, když je znakový řetězec uložen v jedno- nebo dvojrozměrném poli (Array) z proměnných CHARACTER. Přitom musí být pole deklarováno instrukcí DIM.

Každá proměnná CHARACTER v tomto poli pak obsahuje pouze 1 znak znakového řetězce.

Jednorozměrné pole z proměnných typu CHARACTER nazýváme proměnná STRING. Při přístupu k jednorozměrným proměnným CHARACTER se nezadáva žádný index. Naproti tomu při přístupu ke dvojrozměrným proměnným CHARACTER je nutné zadávat index.

Příklad:

```
1 REM Stringvariable AB (délka 10)
2 DIM ABS(10)
3 REM 3 Stringvariablen CD (vždy délka 5)
4 DIM CDS(3, 5)
5 ABS = "Z"
6 CDS(2) = "ABC"
```

Přehled proměnných

Skupina proměnných	Název proměnné	Typ proměnné	Pole (Arrays) možná (X=ano)
Lokální	max. 8 významných znaků	% INTEGER REAL ! DOUBLE ? BOOLEAN \$ CHARACTER	X X X X X
Globální#	vč. "#" -znaku, max. 8 významných znaků	% INTEGER REAL ! DOUBLE ? BOOLEAN \$ CHARACTER	X X X X X
Permanentní@	1 - 100		
Definovatelné permanentní@	max. 16 významných znaků	% INTEGER REAL ! DOUBLE ? BOOLEAN \$ CHARACTER	X X X X X

Základy NC-programování

2.10.4 Proměnné ve standardním NC-programování

Často chceme v dílčím programu v rámci standardního NC-programování (DIN) parametrizovat NC-funkce nebo parametry NC-funkcí pomocí proměnných.

Při tomto přiřazování hodnot je třeba mít na paměti, že výrazy CPL ve standardním NC-programování musí být uzavřeny v lomených závorkách "[" a "]".

Příklady parametrizace standardních NC-vět:

```
10 ANGLE = 45
20 VAL1 = 1.5
30 VAL2 = 1.5
40 XPOS = 10.2
50 YPOS = 5.73
60 FEEDRATE = 1000
N70 Rotate([ANGLE])
N80 Scale(X[VAL1], Y[VAL2])
N90 G1 X[XPOS] Y[YPOS] F[FEEDRATE]
```

 **Číslo věty se nedá parametrizovat pomocí proměnných CPL!**

 **Všechny adresy, které vyvolávají podprogram, nejsou určeny pro variabilní způsob zápisu!**

2.10.5 Variabilní ERRNO pro vyhodnocení chyb funkcí CPL

Funkci CPL, která v případě chyby generuje běhovou chybu, může být jako předávací parametr předána proměnná CPL ERRNO, ve které volaná funkce hlásí chyby.

ERRNO

Proměnná CPL ERRNO je volitelný parametr, který může být naprogramován ve funkcích CPL, u nichž je uveden jako parametr. Tam se dá naprogramovat na libovolné pozici.

Pokud parametr ERRNO není naprogramován, generuje funkce CPL v případě chyby běhovou chybu nebo varování.

Při naprogramovaném parametru ERRNO negeneruje funkce CPL **žádné** běhové chyby ani varování. V takovém případě musí být proměnná CPL ERRNO vyhodnocena v programu CPL a explicitně generována pomocí SETERR běhová chyba nebo pomocí SETWARN výstražné hlášení.

Při správném provedení funkce CPL má ERRNO hodnotu 0. Chyby jsou hlášeny zápornými hodnotami. Význam záporných hodnot je všeobecně platný, ale ne všechny hodnoty jsou použity pro každou funkci CPL. Možné hodnoty chyb pro funkci CPL jsou uvedeny u příslušné funkce.

Seznam všeobecně platných hodnot chyb:

- 0 : Přístup OK
- 1 : Chyba parametrů
- 2 : Souřadnice/osa neexistuje.
- 3 : Nepřípustná souřadnice/osa v kanálu.
- 4 : Osa není žádná pseudosouřadnice.
- 5 : Kanál neexistuje.
- 6 : Funkce smí být vyvolána pouze ve vlastním kanálu.
- 7 : Nelze načíst data.
- 8 : Název zdrojového souboru včetně cesty je příliš dlouhý
- 9 : Přístup ke zdrojovému souboru není možný
- 10 : Název cílového souboru včetně cesty je příliš dlouhý
- 11 : Název souboru (zdrojového nebo cílového) nepřípustný
- 12 : Kopírování není možné

Příklad:

Načíst polohu osy kanálu X v prvním kanálu.

Přitom je možné při stejném účinku alternativně naprogramovat jeden z následujících čtyř řádků:

```
10 POS = ACS("X", 1, 1, ERRNO)
10 POS = ACS("X", 1, ERRNO, 1)
10 POS = ACS("X", ERRNO, 1, 1)
10 POS = ACS(ERRNO, "X", 1, 1)
```

Základy NC-programování

SETERR

Po vyhodnocení parametru ERRNO je možné pomocí příkazu CPL SETERR generovat běhovou chybu. Tím se dílčí program na tomto místě přeruší.

Přitom je vždy vydán text uvedený v parametru *<chybový řetězec>*, nezávisle na jinak nastaveném jazyce.

Syntaxe:

SETERR(*<chybový řetězec>*[, *<kanál>*])

kde

<Chybový řetězec> Text, který je vydán jako chyba.
(pod číslem chyby Wera 3371)

<Kanal> Číslo kanálu, ve kterém má být vydána chyba.
Pokud není uveden *<kanál>*, použije se jako standardní hodnota aktuální kanál.

ERRNO Proměnná CPL, programovatelná na libovolném místě v závorce.

Při použití ERRNO se při chybě negeneruje běhová chyba; vrácené hodnoty jsou:

0: Přístup OK

-1: Chyba parametrů

-5: Kanál neexistuje.

Příklad:

Načíst polohu osy kanálu X v prvním kanálu

```
10 POS = ACS("X", 1, 1, ERRNO)
20 IF ERRNO <> 0 THEN
30 SETERR("Nelze načíst polohu")
40 ENDIF
```

Základy NC-programování

SETWARN

Po vyhodnocení ERRNO je možné pomocí příkazu CPL SETWARN generovat varování, přičemž dílčí program běží dál. Přitom je vždy vydán text uvedený v parametru <výstražný řetězec>, nezávisle na jinak nastaveném jazyce.

Syntaxe:

SETWARN(<výstražný řetězec>[, <kanál>])

kde

<Výstražný řetězec>	Text, který je vydán jako varování. (pod číslem chyby Wera 3372)
<Kanal>	Číslo kanálu, ve kterém má být vydáno varování. Pokud není uveden <kanál>, použije se jako standardní hodnota aktuální kanál.
ERRNO	Proměnná CPL, programovatelná na libovolném místě v závorce. Při použití ERRNO se při chybě negeneruje běhová chyba; vrácené hodnoty jsou: 0: Přístup OK -1: Chyba parametrů -5: Kanál neexistuje.

CLRWARN

S použitím příkazu CLRWARN je možné vymazat všechna varování kanálu, generovaná příkazem SETWARN.

Syntaxe:

CLRWARN([<kanál>])

kde

<Kanal>	Číslo kanálu, ve kterém mají být vymazána všechna výstražná hlášení generovaná příkazem SETWARN. Pokud není uveden <kanál>, použije se jako standardní hodnota aktuální kanál.
ERRNO	Proměnná CPL, programovatelná na libovolném místě v závorce. Při použití ERRNO se při chybě negeneruje běhová chyba; vrácené hodnoty jsou: 0: Přístup OK -1: Chyba parametrů -5: Kanál neexistuje.

Příklad:

```
10 CLRWARN(, ERRNO)
20 IF ERRNO <> 0 THEN
30   SETWARN("Varování nej sou vymazána")
40 ENDIF
```

Základy NC-programování

2.11 Instrukce CPL

Lokálním a globálním proměnným mohou být přiřazeny hodnoty. To se provádí s použitím rovnítko "=".

Příklad: Přiřazení hodnoty pro proměnnou BOOLEAN

```
10 START? = FALSE
```

Hodnota
Přiřazovací symbol
(logická) proměnná

Příklad: Přiřazení hodnoty pro proměnnou REAL

```
1 X1MIN! = 2097.876
```

Hodnota (max. 7-místná)
Přiřazovací symbol
Proměnná REAL s dvojnásobnou přesností

Příklad: Přiřazení hodnot mezi proměnnými

```
1 XSOLL = X1MIN!
```

Hodnota (proměnná REAL s dvojnásobnou pře:
Přiřazovací symbol
Proměnná REAL s jednoduchou přesností

Proměnná, které má být přiřazena hodnota, musí být umístěna vlevo od přiřazovacího symbolu, příslušná hodnota pak vpravo od něj. Tato konvence se musí dodržovat obzvláště tehdy, když má být hodnota proměnné přiřazena jiné proměnné.

NUL

Jestliže není proměnné přiřazena žádná hodnota, má tato proměnná hodnotu NUL, tzn. výrok $\langle \text{proměnná} \rangle = \text{NUL}$ je pravdivý. Zde je vidět, že rovnítko může být umístěno také v relacích nebo podmínkách.

Pokud má být lokální nebo globální proměnná cíleně vymazána, je možné to provést přiřazením hodnoty NUL. Permanentní proměnná se nedá vymazat, ale pouze přepsat.

Příklad: Vymazání proměnné

```
1 XSOLL = NUL
2 IF XSOLL = NUL THEN
3     PRN#(0, "Proměnná není obsazena.")
4 ENDIF
```

Základy NC-programování

Matematické operace

Kromě přiřazení hodnoty jako konstantního výrazu (číslo) nebo jako proměnné může být proměnné přiřazena také hodnota výrazu CPL. Ve výrazu CPL mohou být funkce s konstantami a proměnnými.

K nejjednodušším funkcím patří základní aritmetické operace:

Sčítání	» + «
Odčítání	» - «
Násobení	» * «
Dělení	» / «

Platí, že násobení a dělení se provádí před sčítáním a odčítáním. Kromě toho lze používat závorky, u nichž je pro jednoduché výrazy (bez volání funkcí) možné 7-násobné vnoření.

Příklad:

```
1 I% = 25: XHABEN = 10
2 XSOLL = 150/(100-I%)+XHABEN      XSOLL má hodnotu 12
```

Dále je rovněž možné volat matematické funkce působící na proměnné, konstanty nebo výrazy CPL, které musí být uvedeny v kulatých závorkách bezprostředně za příslušným příkazovým slovem. Funkce se vždy vztahuje k internímu číselnému zobrazení vstupní hodnoty. To je možné při provádění programu zkontrolovat pod "Kontrolou programu". U vnořených výrazů, obzvláště s voláním funkcí, je třeba brát v úvahu maximální možnou hloubku vnoření, kterou potřebují výrazy v závorkách během provádění a která je závislá na paměťovém místě.

ABS

Vrací absolutní hodnotu vstupní hodnoty, tzn. záporné hodnoty se mění na kladné, kladné zůstávají kladnými.

Příklad:

```
1 I% = -125
2 XWERT = 2*SQRT(ABS(100+I%))      XWERT má hodnotu 10
```

INT

převádí vstupní hodnotu (REAL) odříznutím míst za desetinnou tečkou (zaokrouhlením dolů) na celé číslo (INTEGER). Vstupní hodnotou může být konstanta nebo proměnná.

Příklad:

```
1 XWERT% = INT(10.9)              XWERT má hodnotu 10
```

Základy NC-programování

ROUND

převádí vstupní hodnotu zaokrouhlením nahoru, resp. dolů na celé číslo (INTEGER). Vstupní hodnotou může být výraz REAL.

Příklad:

1	XWERT% = Round(10.9)	XWERT má hodnotu 11
2	XWERT% = Round(5.5)	XWERT má hodnotu 6
3	XWERT% = Round(5.49)	XWERT má hodnotu 5

SQRT

Vytváří druhou odmocninu vstupní hodnoty. Vstupní hodnota přitom nesmí být záporná, protože by pak výsledek nebyl definován.

Příklad:

1	I% = 44	
2	XSOLL = 4*SQRT(100+I%)	XSOLL má hodnotu 48

SIN, COS, TAN, ASIN, ACOS, ATAN

U trigonometrických funkcí, které zpracovávají úhly v šedesátiných stupních, má smysl označovat úhly jako proměnné REAL s dvojnásobnou přesností. Mohou být používány následující trigonometrické funkce:

- SIN Funkce sinus
- COS Funkce kosinus
- TAN Funkce tangens
- ASIN Funkce arcussinus
- ACOS Funkce arcuskosinus
- ATAN Funkce arcustangens

Příklad:

1	WINKEL = 30	
2	XWERT = SIN(WINKEL)	XWERT má hodnotu 0.5
3	YWERT = ASIN(XWERT)	YWERT má hodnotu 30

Logické operace

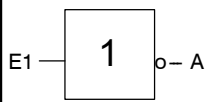
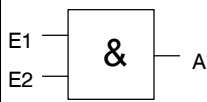
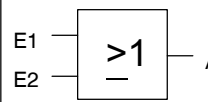
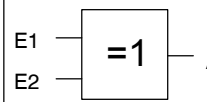
Logické operace mohou být prováděny binárně s logickými proměnnými a decimálně s proměnnými INTEGER. Mohou být také zobrazovány, jak je znázorněno na následujícím schématu binárních operací, s obvyklými operačními symboly, znaménky »·« a »+«, (-ne v CPL-). I zde platí, že »násobení má přednost před sčítáním«, tedy operace AND před operací OR. Je možné až 7-násobné vnoření závorek.

Základy NC-programování

NOT, AND, OR, XOR

CPL nabízí čtyři logické funkce:

- Funktion negace **NOT**
- Funktion A **AND**
- Funktion NEBO **OR**
- Funkce EXKLUZIVNÍ NEBO **XOR**

 <p>Člen NOT</p> $E1 = \bar{A}$	 <p>Člen AND</p> $E1 \cdot E2 = A$	 <p>Člen OR</p> $E1 + E2 = A$	 <p>Člen XOR</p> $E1 \cdot \bar{E2} + \bar{E1} \cdot E2 = A$	
	NOT	AND	OR	XOR
E1	0 L	0 0 L L	0 0 L L	0 0 L L
E2	- -	0 L 0 L	0 L 0 L	0 L 0 L
A	L 0	0 0 0 L	0 L L L	0 L L 0

Logické operace se mohou používat k maskování bitů.

Příklad: Je v @20 nastaven bit 0?

```

...
20 IF @20 AND 1 <> 0 THEN GOTO . GESETZT
30 ELSE GOTO . UNGESETZT ENDI F
...

```

Konverze mezi číselnými systémy

BCD

Konvertovat binární hodnotu na formát BCD:

$$\langle \text{Hodnota BCD} \rangle = \mathbf{BCD}(\langle \text{binární hodnota} \rangle)$$

Příklad:

$$1 \text{ BCD_WERT} = \text{BCD}(49) \qquad \text{BCD_WERT má hodnotu 73}$$

BIN

Konverze BCD-kódovaných čísel na binární hodnotu:

$$\langle \text{Binární hodnota} \rangle = \mathbf{BIN}(\langle \text{hodnota BCD} \rangle)$$

Příklad:

$$1 \text{ BIN_WERT} = \text{BCD}(49) \qquad \text{BIN_WERT má hodnotu 31}$$

Základy NC-programování

Relační operace

=, >=, >, <>, <=, <

Jsou dovoleny následující relační operátory:

» = «	rovno
» >= «	větší nebo rovno
» > «	větší
» <> «	různo
» <= «	menší nebo rovno
» < «	menší

Relační operace se používají pro popis **relace** ("splněno" nebo "nesplněno") **podmínky** (např. při instrukcích REPEAT - UNTIL, WHILE - DO - END, IF - THEN - ELSE - ENDIF).

Základy NC-programování

2.12 Další základní prvky CPL

Konstanty

Jsou-li pro průběh programu stanoveny číselné hodnoty, které mají zůstat beze změny (konstantní), je možné používat tyto hodnoty v instrukcích přímo v číslicovém zápisu.

Celočíselná konstanta (INTEGER)

Celá čísla se zapisují bez desetinné tečky.

Příklad:

ANZAHL% = 4

Konstanta INTEGER

Konstanta s pohyblivou řádovou čárkou (REAL)

Reálná čísla (desetinná čísla nebo zlomky) jsou označena desetinnou tečkou (pohyblivou řádovou čárkou).

Příklad:

PI = 3. 141593

Konstanta REAL

Konstanta s dvojnásobnou přesností a operace s dvojnásobnou přesností

Konstanty, které jsou přiřazeny proměnné REAL s dvojnásobnou přesností nebo jsou s takovou proměnnou porovnávány, se zobrazují s dvojnásobnou přesností (tzn. s přesností na 15 míst).

Příklad:

Přiřazení konstant REAL s dvojnásobnou přesností a porovnávání proměnných s konstantami REAL s dvojnásobnou přesností:

```
4 D5! = -1234. 123456 + 12345 + 1234. 234567
20 D0! = 123456789. 123456
22 D1! = 1. 12345678901234
24 D2! = -123456789012345
26 D3! = -1234. 123456
```

Základy NC-programování

Následující dotazy vedou k výsledku: E? = TRUE

```

28 IF D0!=123456789.123456 THEN E?=TRUE ELSE E?=FALSE ENDIF
29 IF D1!=1.12345678901234 THEN E?=TRUE ELSE E?=FALSE ENDIF
30 IF D2!=-123456789012345 THEN E?=TRUE ELSE E?=FALSE ENDIF
31 IF D3!=-1234.123456 THEN E?=TRUE ELSE E?=FALSE ENDIF
32 IF D0! + 2.1 + 3.1 = 123456789.123456 + 2.1 + 3.1 THEN
33 E? = TRUE
34 ELSE
35 E? = FALSE
36 ENDIF
37 IF (D0! + 2.1) + 3.1 = 123456789.123456 + 2.1 + 3.1 THEN
38 E? = TRUE
39 ELSE
40 E? = FALSE
41 ENDIF

```

Znaková řetězcová konstanta

Znaková řetězcová konstanta (konstanta String) je omezena uvozovkami.

Příklad:

BEI SPI ELS = "Toto je znakový řetězec" Konstanta STRING

Klíčové znaky

V CPL se používají následující klíčové znaky:

#	!	?	,	"	[]	<	-	/	&
@	%	\$:	()	=	>	+	*	

Čárka se normálně používá jako oddělovací znak. Pouze uvnitř znakových řetězců se zpracovává jako interpunkční znaménko. Tečka se používá v desetinných číslech jako desetinná tečka a u cílů skoku jako označení návěští (Label). Tečka uvnitř znakových řetězců je považována za interpunkční znaménko.

Základy NC-programování

Rezervovaná příkazová slova

Zde uvedené klíčové pojmy musí být zapsány samostatně nebo být omezeny speciálními znaky. Tak se dají identifikovat jako příkazová slova. Při výběru názvů proměnných se nesmí používat žádná rezervovaná příkazová slova!

Příklad:

GOTO 10

Skok na řádek 10

GOTO10

libovolný symbolický název (proměnná); samostatně uvedeno vede k chybovému hlášení "Laufzeitfehler 2167 = fehlt", protože je očekáváno přiřazení hodnoty pro proměnnou "GOTO10".

Klíčové pojmy:

A: ABS ACOS AND APOS ASC ASIN ATAN	AXO AXP	B: BCD BIN BITIF	C: CALL CASE CHR\$ CLOCK CLOSE CLRWARN	COF COS CPOS CPROBE	D: DATE DBSEA DBTAB DCT DIM DIRCR	DIRDEL DIRINF DO DPC
E: ELSE END ENDIF ENDCASE EOF ERASE	ERRNO	F: FALSE FILEACCESS FILECOPY FILEDATE FILENO FILEPOS	FILESIZE FOR FXC FXCR FXDEL FXINS	G: GETERR GOTO	I: IF INP# INSTR INT	
L: LABEL LEN LJUST		M: MCODS MCOPI MID\$ MMC MPOS	N: NCF NEXT NJUST NOT NUL		O: OF OPENR OPENW OTHERWISE OR	
P: PDIM PMT PMV PPOS PRN# PROBE		R: REM REPEAT REWRITE ROUND	S: SCL SCS SCSL SD SDR SEEK	SETERR SETWARN SIN SPOS SQRT STEP STR\$	T: TAN TCV THEN TIME TO TRIM\$	TRUE
U: UNTIL		V: VAL VERSINF\$	W: WAIT WHILE WPOS		X: XOR XTAB	
Z: ZOT ZOV						

2.13 Příkazy pro synchronizaci NC-vět

2.13.1 Přehled

Pomocí příkazů pro synchronizaci NC-vět je možné:

- Synchronizovat NC-programy s určitými událostmi.
- Synchronizovat přípravu vět s prováděním programu.
- Omezit náhled vět (Look-Ahead) na určitý počet vět.
- Vzájemně synchronizovat NC-programy v různých kanálech.

Synchronizační funkce působí buď v okamžiku přípravy vět, nebo v aktivním okamžiku provádění vět.

Synchronizační funkce **přípravy vět**:

- WAIT (bez parametru): Zastavuje přípravu vět, dokud nejsou zpracovány všechny předchozí vět.
- Funkce CPL WAIT(<čekací doba>): Příprava vět se zastaví na zadanou dobu.
- Funkce CPL WAIT(BITIF(...)): Příprava vět čeká na určitý signál na bitovém rozhraní SPS-NC.
- BlkNmb: Omezuje náhled vět na naprogramovaný počet vět.

Synchronizační funkce v **aktivním okamžiku**:

- WAITA / WAITO: Čekání na určitý stav na bitovém rozhraní SPS-NC.
- WPV / WPVE: Čekání na hodnotu permanentní proměnné CPL.
- SPV / SPVE: Zápis permanentní proměnné CPL.
- ASTOPA / ASTOPO: Zastavení pohybu, dokud nejsou dosaženy určité polohy os.
- BSTOPA / BSTOPO: Zastavení pohybu, dokud nejsou dosaženy určité polohy v základní soustavě souřadnic obrobku (BCS).
- WSTOPA / WSTOPO: Zastavení pohybu, dokud nejsou dosaženy určité polohy v aktuální soustavě souřadnic obrobku (WCS).
- OFFSTOPA / OFFSTOPO: Zrušení podmínek zastavení v řídicím kanálu.

2.13.2 Synchronizační funkce přípravy vět

WAIT(bez parametru)


Funkce WAIT **zastavuje přípravu vět**, dokud nejsou úplně zpracovány všechny větý, které byly naprogramovány před větou WAIT. Je nezbytně nutná, jestliže mají být později v programu používána aktuální data související se strojem nebo s procesem.

Příprava větý, ve které jsou analyzovány a interpretovány jednotlivé programové řádky, probíhá –z časového hlediska - vždy před provedením na stroji. Časový interval mezi přípravou a provedením není konstantní, ale závisí na různých parametrech (posuv, dráha pojezdu, Look-Ahead atd.).

Pokud má nyní program reagovat na skutečný stav stroje nebo procesu (např. aktuální skutečná poloha, signál na bitovém rozhraní atd.), musí funkce WAIT zajistit, aby tyto časové intervaly byly přesně v okamžiku vyhodnocení rovné "0".

Příklad: WAIT(bez parametru)

```
N10 X0
N100 (MSG, Ještě běží)
N20 X150
30 WAIT Zastavit zpracování vět
40 XPOS = MPOS(1) - 150
50 IF XPOS < 0.0001 THEN Na X=150 je vydáno hlášení
    (MSG, Poloha dosažena) "Poloha dosažena"
70 ENDIF
```

 **Funkce WAIT (bez parametru) může být naprogramována jak ve standardních NC-větách (DIN) tak i ve větách CPL. Ve větě CPL s instrukcí WAIT nesmí být naprogramován znak ":". Následující instrukce CPL musí být zapsány v nové větě CPL.**

Základy NC-programování

Funkce CPL: WAIT(<čekací doba>)

Zastavuje přípravu vět, dokud neuplyne naprogramovaná čekací doba.

Syntaxe:

WAIT(, <čekací doba>[, <výsledná proměnná>])

kde

<Čekací doba> Čekací doba v milisekundách, bez desetinných míst. Čekací doba smí být naprogramována také jako celočíselný aritmetický výraz.

<Výsledná proměnná> Volitelná proměnná Integer. Po uplynutí čekací doby je do <výsledné proměnné> přiřazena 1.

Příklady funkce WAIT s čekací dobou:

10 WAIT(, 1000, E%)	Zpracování vět se zastaví na 1000ms. Potom je proměnná E% obsazena hodnotou Integer "1".
10 WAIT(, ZEIT%)	Jak dlouho zůstane příprava vět zastavena, závisí na obsahu proměnné Integer ZEIT%. Nevrací se žádná hodnota.

Funkce CPL: WAIT(BITIF(...))

Zastavuje přípravu vět, dokud nenastane určitý stav na bitovém rozhraní SPS-NC.

Volitelně je možné naprogramovat také časový interval. Příprava vět je pak zastavena tak dlouho, dokud nenastane příslušný stav na bitovém rozhraní nebo neuplyne zadaný časový interval.

Syntaxe:

WAIT(<BITIF-podmínka>[, [<timeout>] [, <výsledná proměnná>]])

kde

<BITIF-podmínka> Specifikuje podmínku, která se má kontrolovat na bitovém rozhraní SPS-NC. Přitom je nutné dodržovat následující syntaxi:

[NOT() BI TF(<parametr>)] [=<stav>]

<Parametr> Předávací parametr funkce BITIF (popis viz funkce BITIF, strana 5-44).

<Stav> Výraz BOOL, se kterým se porovnává výsledek funkce BITIF. Pokud není <stav> naprogramován, provádí se porovnání s hodnotou TRUE.

Když je splněna podmínka, rozběhne se opět příprava vět.

<Timeout> Příprava vět čeká, až bude splněna <podmínka BITIF>, nejdéle však, dokud neuplyne zadaná doba <Timeout>.

Základy NC-programování

<Výsledná proměnná> Volitelná proměnná Integer. Po uplynutí čekací doby je do <výsledné proměnné> přiřazena 1.

Příklady WAIT(BITIF(...)):

```
10 WAIT(BITIF(1, 1, 1)=TRUE)
```

Čekání, až bude nastaven 2. osový vstupní signál 1. osy.

```
10 WAIT(BITIF(2, 0, 2)=(E1? OR E2?))
```

Čekání, až bude mít 3. kanálový vstupní signál 2. kanálu hodnotu logického výrazu (E1? OR E2?).

```
10 WAIT(NOT BITIF(3, 2, 1), , C%)
```

Čekání, až bude mít 4. vřetenový vstupní signál 1. vřetena hodnotu FALSE. Proměnná C% dává buď hodnotu "0", když je podmínka splněna již při vyvolání funkce WAIT, nebo hodnotu "2", je-li podmínka splněna až během čekání.

```
20 IF C%=0 THEN
```

```
30   DSP(10, 10, "PODMÍNKA JIŽ OK")
```

```
40 ENDIF
```

```
10 WAIT(BITIF(4, 4, 1)=E7?, 250, ERG%)
```

Čekání, až 5. osový výstupní signál 1. osy nabude hodnoty proměnné E7?, nebo uplyne 250 ms. Proměnná ERG% dává buď hodnotu "0", když je podmínka splněna již při vyvolání funkce WAIT, nebo hodnotu "1", když uplyne časový interval, nebo hodnotu "2", když je podmínka splněna během čekání.

```
20 IF ERG%=0 THEN
```

```
30   DSP(10, 10, "NEČEKÁ SE")
```

```
40 ENDIF
```

```
50 IF ERG%=2 THEN
```

```
60   DSP(10, 10, "ČEKÁ SE >250ms")
```

```
70 ENDIF
```

Základy NC-programování

BlkNmb (BNB)

Pomocí standardní NC-funkce BlkNmb je možné omezit maximální počet vět v přípravě vět.

S použitím funkce BlkNmb lze například řídit další zpracování výsledků měření zjištěných v průběhu programu v dalším dílčím programu.

Syntaxe:

BlkNmb(<počet>)

Omezení zapnuto.

BlkNmb() nebo

Omezení vypnuto.

BlkNmb(0)

Max. povolený počet vět závisí na MP 7060 00110.

Zkratka: **BNB(...)**

kde

<Počet>

Maximální požadovaný počet vět.

Integer.

Rozsah zadání: větší/rovno 0.

0: Počet vět je závislý na MP 7060 00110.

Zvláštnosti a omezení:

- Pokud je v okamžiku vyvolání funkce připraveno již více než <počet> vět, zastaví se příprava vět na tak dlouho, dokud počet připravených vět neklesne pod <počet>.
- Je-li parametrem <počet> zadáno více vět, než se předpokládá v rámci konfigurace řízení, je efektivní počet vět automaticky omezen na hodnotu nastavenou v MP 7060 00110.

2.13.3 Synchronizační funkce v okamžiku provádění věty

V IndraMotion MTX může v každém kanálu běžet jeden program. Jsou-li jednotlivé úseky obrábění rozděleny do různých jednotlivých programů a tyto programy běží v různých kanálech, je možné řídit průběh obrábění všech jednotlivých programů synchronizačními funkcemi v okamžiku provádění věty.

Všechny synchronizační funkce v okamžiku provádění věty mají **seznamy parametrů CPL** (podobně jako podprogramy). V nich jsou naprogramovány seznamy výrazů CPL, které jsou uzavřeny v lomených závorkách "[" a "]".

Všeobecně platí:

- Permanentní proměnné CPL používané ve funkcích WPV, WPVE, SPV a SPVE jsou platné v celém systému. Programátor proto musí zajistit, aby nedošlo k žádným nezamýšleným následkům kvůli nesprávnému použití.
- V rámci nabízených NC-funkcí jsou dovoleny pouze následující jednoduché typy permanentních proměnných CPL:
 - INT
 - BOOL
 - REAL
 - DOUBLE

U polí je možné používat jen jednotlivé prvky!



Funkce WAITA, WAITO, WVP a WVPE působí implicitně pokles (Downslope) na konci věty. Nesprávně nastavené body synchronizace mohou vést k poškození stroje.

Vyzkoušejte průběh programu před vlastním obráběním s ohledem na možné problémy synchronizace během programu.

Základy NC-programování

Čekání na stavy na bitovém rozhraní SPS-NC: WAITA / WAITO

S použitím funkce WAITA / WAITO se čeká, až jeden nebo několik signálů rozhraní z maximálně 16 signálů nabude předem zadané hodnoty.

V závislosti na propojení několika signálů rozhraní je možné naprogramovat:

- **WAITA:** "Spojení A" jednotlivých signálů
Čekání, až **všechny** signály rozhraní nabudou předem zadané hodnoty.
- **WAITO:** "Spojení NEBO" jednotlivých signálů
Čekání, až **jeden** ze signálů rozhraní nabude předem zadané hodnoty.

Příklad:

Program 1 v kanálu 1 obrábí čelní plochu soustruženého dílu. Program 2 v kanálu 2 má frézovat drážku na této čelní ploše a musí **počkat**, až program 1 uvolní soustruženou součástku pro program 2. Uvolnění pro program 2 se provádí **nastavením** určitých signálů rozhraní. Když signály rozhraní dosáhnou tohoto stavu, předá kanál 1 uvolnění kanálu 2. Během obrábění podle programu 2 čeká program 1 na program 2, aby mohl pokračovat ve svém obrábění.

Syntaxe:

Čekání na **každý** z uvedených signálů:

WAITA[BITIF (<parametr>) {=<stav>}, BITIF (<parametr>) {=<stav>}, ..., {<timeout>}]

Čekání na **jeden** z uvedených signálů:

WAITO[BITIF (<parametr>) {=<stav>}, BITIF (<parametr>) {=<stav>}, ...{, <timeout>}]

kde:

BITIF	Funkce BITIF se dotazuje rozhraní SPS-NC. Volitelně je možné se dotazovat na 2 až 16 signálů rozhraní současně.
<Parametr>	Předávací parametr funkce BITIF (popis viz funkce BITIF, strana 5-44).
<Stav>	Výraz BOOL, se kterým se porovnává výsledek funkce BITIF. Pokud není <stav> naprogramován, provádí se porovnání s hodnotou TRUE. Když je splněna podmínka, rozběhne se opět příprava vět.
<Timeout>	Volně nastavitelný čas v ms, standardně = 0. Když uplyne <timeout> předtím, než je splněna příslušná podmínka, je generováno varování a čekání pokračuje. Není-li naprogramován timeout nebo je roven 0, negeneruje se žádné varování.

Základy NC-programování

Příklady:

```
N10 WAITO[ BITIF(10, 1, 1) =FALSE,  
          BITIF(11, 1, 2) ]
```

Čekání je aktivní, dokud BITIF(10,1,1) nenabude hodnoty 0 nebo BITIF(11,1,2) hodnoty 1.

```
N10 WAITA[ BITIF(10, 1, 1) =FALSE,  
          BITIF(11, 1, 2) ]
```

Čekání je aktivní, dokud BITIF(10,1,1) nenabude hodnoty 0 a BITIF(11,1,2) hodnoty 1.

Pro WAITA, WAITO platí:

- Jsou-li funkce WAITA a WAITO naprogramovány v jednom NC-bloku, zastaví se provádění vět, dokud nebudou splněny obě podmínky. Nejprve se vyhodnocuje podmínka WAITO.



Funkce WAITA, WAITO, WVP a WVPE působí implicitně pokles (Downslope) na konci věty. Nesprávně nastavené body synchronizace mohou vést k poškození stroje. Vyzkoušejte průběh programu před vlastním obráběním s ohledem na možné problémy synchronizace během programu.

Základy NC-programování

Čekání na hodnotu permanentní proměnné CPL: WPV / WPVE

S použitím funkce WPV / WPVE se čeká, až permanentní proměnná CPL nabude určité srovnávací hodnoty.

Srovnávací hodnota může být zjišována v různých časech:

- **WPV:** Srovnávací hodnotou je výraz CPL, který se určuje v průběhu programu a porovnává s hodnotou permanentní proměnné. Kvůli vyhodnocení v průběhu programu je přípustný pouze jednoduchý výraz CPL.
- **WPVE:** Srovnávací hodnotou je výraz CPL, který se určuje v okamžiku přípravy, ale teprve v průběhu programu porovnává s hodnotou permanentní proměnné.

Syntaxe:

Srovnávací hodnota se určuje v **průběhu programu**:

WPV[<perm. proměnná CPL><relační operátor><jednoduchý výraz CPL>{ , <timeout> }]

Srovnávací hodnota se určuje již v **okamžiku přípravy**:

WPVE[<perm. proměnná CPL><relační operátor><výraz CPL>{ , <timeout> }]

kde:

<Perm. proměnná CPL>

Permanentní proměnná , označená znakem "@" a následovaná názvem proměnné.

<Relační operátor>

Jsou možné následující relační operátory:

- = Permanentní proměnná CPL je **rovná** hodnotě výrazu CPL. Má smysl pouze pro hodnoty Integer nebo Bool.
- < > Permanentní proměnná CPL se **nerovná** hodnotě výrazu CPL. Má smysl pouze pro hodnoty Integer nebo Bool.
- < Permanentní proměnná CPL je **menší** než hodnota výrazu CPL.
- ≤ Permanentní proměnná CPL je **menší nebo rovná** hodnotě výrazu CPL.
- > Permanentní proměnná CPL je **větší** než hodnota výrazu CPL.
- ≥ Permanentní proměnná CPL je **větší nebo rovná** hodnotě výrazu CPL.

Základy NC-programování

*<jednoduchý
výraz CPL>*

Aby nebylo ovlivněno generování pohybu, smí se v průběhu programu vyhodnocovat pouze jednoduché výrazy CPL.

Jednoduchý výraz CPL je matematický výraz, skládající se z permanentních proměnných CPL, konstant a matematických operací možných v CPL.

<Výraz CPL>

Libovolný matematický výraz v programovacím jazyce CPL.

<Uplynutí stanoveného času>

Volně nastavitelný čas v ms, standardně = 0.

Když uplyne <timeout> předtím, než je splněna příslušná podmínka, je generováno varování a čekání pokračuje.

Není-li naprogramován timeout nebo je roven 0, negeneruje se žádné varování.

Příklady:

N10 WPV[@9=10]

Program čeká v aktivním okamžiku, až permanentní proměnná @9 nabude hodnoty 10.

N10 WPVE[@8=(5*#VAR2%)]

Výraz "5 * #VAR2%" se vyhodnocuje v okamžiku přípravy. Přitom zjištěná hodnota se v průběhu programu porovnává s permanentní proměnnou @8. Pokud @8 neodpovídá určené hodnotě, není aktivní žádná nová NC-věta.

Základy NC-programování

Zápis permanentní proměnné CPL: SPV / SPVE

Pomocí funkce SPV / SPVE se v průběhu programu přiřazuje permanentní proměnné CPL zápisem hodnota.

Hodnota může být zjišťována v různých časech:

- **SPV:** přiřazovaná hodnota pro permanentní proměnnou se určuje v průběhu programu.
Kvůli vyhodnocení v průběhu programu je přípustný pouze jednoduchý výraz CPL.
- **SPVE:** přiřazovaná hodnota permanentní proměnné se určuje v okamžiku přípravy (čas interpretace CPL), ale teprve v průběhu programu přiřazuje permanentní proměnné CPL.

Syntaxe:

Přiřazovaná hodnota se určuje v **průběhu programu**:

SPV[<perm. proměnná CPL> = <jednoduchý výraz CPL>]

Přiřazovaná hodnota se určuje v **době přípravy**:

SPVE[<perm. proměnná CPL> = <výraz CPL>]

kde:

<Perm. proměnná
CPL>

Permanentní proměnná , označená znakem
"@" a následovaná názvem proměnné.

<jednoduchý
Výraz CPL>

Aby nebylo ovlivněno generování pohybu, smí
se v průběhu programu vyhodnocovat pouze
jednoduché výrazy CPL.
Jednoduchý výraz CPL je matematický výraz,
skládající se z permanentních proměnných
CPL, konstant a matematických operací
možných v CPL.

<Výraz CPL>

Libovolný matematický výraz v programo-
vacím jazyce CPL.

Příklady:

N10 SPV[@6=1]

Permanentní proměnné "@6"
je v průběhu programu
přiřazena hodnota 1.

N10 SPV[@5=(7*(@PERMVAR1% +5))]

Hodnota výrazu (7*(@PERM-
VAR1% + 5)) se určuje v
průběhu programu a pak
přiřazuje proměnné @6.

N10 SPVE[@5=(7*#VAR1%)]

Hodnota výrazu (7 * #VAR1%)
se určuje v době přípravy a v
průběhu programu přiřazuje
proměnné @5.

Základy NC-programování

Zastavení pohybu, dokud není dosažena poloha osy: ASTOPA / ASTOPO

S použitím funkce ASTOPA / ASTOPO je možné synchronizovat pohyby mezi kanály. V závislosti na **poloze** jedné nebo několika os **osové soustavy souřadnic ACS** v kanálu se zastavuje a znovu spouští synchronní pohyb v jiném kanálu.

Omezení:

- Osy, které se používají k synchronizaci, musí patřit jinému než řídicímu kanálu, jinak může dojít k samozádržnému zablokování.
- Řídicí kanál se musí nacházet v provozním režimu automatika nebo ruční zadávání.
- Jsou-li pro kanál současně zadány podmínky AND a OR, dojde k zastavení kanálu, když je splněna příslušná podmínka alespoň pro jednu z obou funkcí.

Pro každý řídicí kanál je možné zadat jednu jedinou nebo několik podmínek pro zastavení kanálu.

- **ASTOPA:** Když jsou splněny **všechny podmínky**, zastaví se synchronní pohyb řídicího kanálu (logická operace AND).
- **ASTOPO:** Když je splněna **alespoň jedna podmínka**, zastaví se synchronní pohyb řídicího kanálu (logická operace OR).

 **Zadáním nových podmínek AND a OR se ruší všechny dosavadní podmínky v tomto kanálu.**

Syntaxe:

Podmínky AND:

ASTOPA[<číslo kanálu>,
<podmínka1>{ , <podmínka2> } { . . { , <podmínka8> } . }]

Podmínky OR:

ASTOPO[<číslo kanálu>,
<podmínka1>{ , <podmínka2> } { . . { , <podmínka8> } . }]

kde:

<Číslo kanálu> Číslo řídicího kanálu (1..n). Hodnota Integer nebo proměnná Integer.

<Podmínka1,
podmínka2...
podmínka8> Zadání 1 až 8 podmínek ve formě:
<osa><relační operátor><srovnávací hodnota>

kde:

<Osa> Název systémové nebo kanálové osy, naprogramovaný jako řetězcová konstanta CPL nebo jako řetězcová proměnná CPL.

Základy NC-programování

<i><Relační operátor></i>	povolené operátory: < , ≤, > , ≥
<i><Srovnávací hodnota></i>	Hodnota Real nebo výraz CPL Real. Hodnota se určuje v době přípravy a zůstává modálně účinná.

 **Z jednoho kanálu lze zastavit maximálně 4 jiné kanály s podmínkami AND/OR.**

Příklad:

Používání názvů os a čísel

10 ACHSNR% = 2

20 ACHSNAMES = "X"

30 STOPCHAN% = 2

:

N40 ASTOPO[STOPCHAN%, ACHSNR%<10]

:

N90 ASTOPO[STOPCHAN%, "Z">20. 3]

:

N150 ASTOPO[STOPCHAN%, ACHSNAMES<1. 5]

Definice:

- číslo osy,

- název osy

- číslo kanálu

Základy NC-programování

Zastavení pohybu, dokud není dosažena základní poloha obrobku: BSTOPA / BSTOPO

S použitím funkce BSTOPA/BSTOPO je možné synchronizovat pohyby mezi kanály. V závislosti na **poloze** jedné nebo několika souřadnic **základní soustavy souřadnic obrobku BCS** v kanálu se zastavuje a znovu spouští synchronní pohyb v jiném kanálu.

Omezení:

- Souřadnice, které se používají k synchronizaci, musí patřit jinému než řídicímu kanálu, jinak může dojít k samozadržnému zablokování.
- Řídicí kanál se musí nacházet v provozním režimu automatika nebo ruční zadávání.
- Jsou-li pro kanál současně zadány podmínky AND a OR, dojde k zastavení kanálu, když je splněna příslušná podmínka alespoň pro jednu z obou funkcí.

Pro každý řídicí kanál je možné zadat jednu jedinou nebo několik podmínek pro zastavení kanálu.

- **BSTOPA:** Když jsou splněny **všechny podmínky**, zastaví se synchronní pohyb řídicího kanálu (logická operace AND).
- **BSTOPO:** Když je splněna **alespoň jedna podmínka**, zastaví se synchronní pohyb řídicího kanálu (logická operace OR).

 **Zadáním nových podmínek AND a OR se ruší všechny dosavadní podmínky v tomto kanálu.**

Syntaxe:

Podmínky AND:

BSTOPA[<číslo kanálu>,
<podmínka1>{ , <podmínka2> } { . . { , <podmínka8> } . }]

Podmínky OR:

BSTOPO[<číslo kanálu>,
<podmínka1>{ , <podmínka2> } { . . { , <podmínka8> } . }]

kde:

<Číslo kanálu> Číslo řídicího kanálu (1..n). Hodnota Integer nebo proměnná Integer.

<Podmínka1,
podmínka2...
podmínka8> Zadání 1 až 8 podmínek ve formě:
<souřadnice><relační
operátor><srovnávací hodnota>

kde:

<Souřadnice> Vzhledem k BCS:
Název nebo index souřadnice,
naprogramovaný jako řetězcová
konstanta CPL nebo jako
řetězcová proměnná CPL.

<Relační
operátor> povolené operátory:
< , ≤ , > , ≥

Základy NC-programování

<i><Srovnávací hodnota></i>	Hodnota Real nebo výraz CPL Real. Hodnota se určuje v době přípravy a zůstává modálně účinná.
-----------------------------------	---

 **Z jednoho kanálu lze zastavit maximálně 4 jiné kanály s podmínkami AND/OR.**

Příklad:

Aktivovat podmínku AND pro základní souřadnice obrobku

N10 BSTOPA[3, "z"<12. 0, "x">15]

Kanál 3 se zastaví na tak dlouho, dokud v řídicím kanálu bude platit:

Poloha základní souřadnice obrobku $z < 12$ mm a poloha základní souřadnice obrobku $x > 15$ mm

Zastavení pohybu, dokud není dosažena poloha obrobku: WSTOPA / WSTOPO

S použitím funkce WSTOPA / WSTOPO je možné synchronizovat pohyby mezi kanály. V závislosti na **poloze** jedné nebo několika souřadnic **soustavy souřadnic obrobku WCS** v kanálu se zastavuje a znovu spouští synchronní pohyb v jiném kanálu.

Omezení:

- Souřadnice, které se používají k synchronizaci, musí patřit jinému než řídicímu kanálu, jinak může dojít k samozadržnému zablokování.
- Řídicí kanál se musí nacházet v provozním režimu automatika nebo ruční zadávání.
- Jsou-li pro kanál současně zadány podmínky AND a OR, dojde k zastavení kanálu, když je splněna příslušná podmínka alespoň pro jednu z obou funkcí.

Pro každý řídicí kanál je možné zadat jednu jedinou nebo několik podmínek pro zastavení kanálu.

- **WSTOPA:** Když jsou splněny **všechny podmínky**, zastaví se synchronní pohyb řídicího kanálu (logická operace AND).
- **WSTOPO:** Když je splněna **alespoň jedna podmínka**, zastaví se synchronní pohyb řídicího kanálu (logická operace OR).

 **Zadáním nových podmínek AND a OR se ruší všechny dosavadní podmínky v tomto kanálu.**

Základy NC-programování

Syntaxe:

Podmínky AND:

WSTOPA [<číslo kanálu>,
<podmínka1>{ , <podmínka2> } { . . { , <podmínka8> } . }]

Podmínky OR:

WSTOP0 [<číslo kanálu>,
<podmínka1>{ , <podmínka2> } { . . { , <podmínka8> } . }]

kde:

<Číslo kanálu> Číslo řídicího kanálu (1..n). Hodnota Integer nebo proměnná Integer.

<Podmínka1,
podmínka2...
podmínka8> Zadání 1 až 8 podmínek ve formě:
<souřadnice><relační
operátor><srovnávací hodnota>

kde:

<Souřadnice> Vzhledem k WCS:
Název nebo index souřadnice,
naprogramovaný jako řetězcová
konstanta CPL nebo jako
řetězcová proměnná CPL.

<Relační
operátor> povolené operátory:
< , ≤ , > , ≥

<Srovnávací
hodnota> Hodnota Real nebo výraz CPL
Real. Hodnota se určuje v době
přípravy a zůstává modálně
účinná.

 **Z jednoho kanálu lze zastavit maximálně 4 jiné kanály s podmínkami AND/OR.**

Příklad:

Aktivovat podmínku AND pro souřadnice obrobku

N10 WSTOPA[3, "Z" <12. 0, "X" >15]

Kanál 3 se zastaví na tak dlouho, dokud v řídicím kanálu bude platit:
Poloha souřadnice obrobku Z kanálu je menší než 12 mm a poloha souřadnice obrobku X větší než 15 mm

Základy NC-programování

Zrušení podmínek zastavení: OFFSTOPA / OFFSTOPO

Vymazání podmínek zastavení pro řídicí kanál.

Syntaxe:

OFFSTOPA Maže všechny podmínky zastavení AND

OFFSTOPO Maže všechny podmínky zastavení OR

Pro synchronizační funkce platí:

- ASTOPO, BSTOPO, WSTOPO a OFFSTOPO jsou modálně účinné a vzájemně se ruší.
- ASTOPA, BSTOPA, WSTOPA a OFFSTOPA jsou modálně účinné a vzájemně se ruší.

**POZOR****Neúmyslný pojezdový pohyb!**

Některé NC-funkce očekávají naprogramování adres os, resp. souřadnic. Zpravidla zde smí být kromě adres souřadnic naprogramovány pouze adresy příslušných os kanálu, tedy synchronních os.

Naprogramování adresy asynchronní osy v seznamu parametrů vede obecně k syntaktické chybě, naproti tomu naprogramování mimo seznam parametrů vede k synchronnímu pojezdovému pohybu!

Výjimky:

G74(Home) VA1: Asynchronní osa VA najede do referenčního bodu.

GAX(VA): Asynchronní osa VA je do kanálu převzata

FsMove, FsTorque, FsReset: Funkce najetí na pevný doraz působí také na asynchronní osy.

Základy NC-programování

Notizen:

3 NC-funkce se syntaxí podle DIN 66025 (vč. rozšíření)

3.1 Přehled

Řízení je vybaveno různými NC-funkcemi. Přitom jsou zde kromě příkazů, které jsou stanoveny v DIN 66025, obsažena také podstatná rozšíření v oblasti G-kódů a doplňujících jazykových syntaktických prvků.

Mezi **NC-funkce** řízení patří

- **1- a 2-místné G-kódy:**

orientují se do značné míry na normu DIN 66025 a účelně doplňují její "slovní zásobu G-kódů", čímž

- rozšiřují funkční skupiny, které jsou v DIN k dispozici (např. **G52**, G53, G54, ...), nebo
- zavádějí nové skupiny funkcí, které souvisejí s již existujícími.

1-místné G-kódy mohou být vždy zapsány jako 2-místné s nulou na začátku (např. G0=G00, G1=G01, ...).

- **3-místné G-kódy** (výjimečné případy):

takové G-kódy se používají, když je použita již existující funkce z oboru 2D podobným způsobem v oboru 3D (např. G41, G42 -> G141, G142).

- **G-kódy s příponou ".":**

používají se ve spojení s posunutími nulových bodů nebo transformacemi souřadnic.

Tyto funkce mají až 6 řad, které mohou být pomocí přípony "." volány či programovány.

Příklady:

- G54.4 1. Posunutí nulového bodu řady 4
- G59.4 6. Posunutí nulového bodu řady 4

- **M-funkce s definovanou NC-funkčností**

např. M0, M30

- **NC-funkce s jazykovou syntaxí:**

- (viz kapitola 4)

 **Tabulkový přehled všech NC-funkcí naleznete v příloze od strany A-2.**

Kromě toho jsou všechny NC-funkce uvedeny v rejstříku od strany A-52.

Používané způsoby zápisu

Pro syntaxi NC-funkcí se v příručce používají následující způsoby zápisu:

Typ písma "Courier tučné" nebo "Courier":

Znakové řetězce v tomto typu písma musí být naprogramovány, jak jsou vytištěny.

Příklad: `GO(POL)`

Úhlové závorky < >

označují symbol pro programovaný výraz, resp. parametr. Symbol je vytištěn kurzívou.

Příklad: `<osa1>`

Složené závorky { }

označují volitelný výraz, resp. parametr.

Takové syntaktické prvky mohou, ale nemusí být naprogramovány.

Příklad: `GO({POL, }{<Par1>})`

Znak "|"

vzájemně odděluje možné, ale ne současně použitelné parametry (alternativní parametry).

Příklad: `GO({POL, }{NIPS|IPS1|IPS2|IPS3})`

G-kódy G00

3.2 G-kódy

3.2.1 Lineární interpolace v rychlém chodu

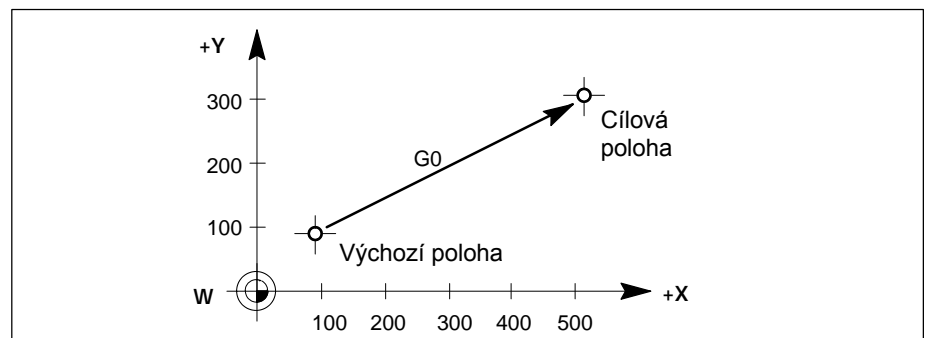
G00

Následek

Najíždí do naprogramované polohy pomocí interpolace přímkou s maximální rychlostí posuvu (= v rychlém chodu).

Platí:

- Nejméně jedna osa pojíždí s maximální rychlostí nebo zrychlením. Rychlost os je řízena tak, aby všechny dosáhly ve stejném okamžiku cílového bodu.
- Při aktivované funkci je vydán signál rozhraní kanálu "Rychlý chod aktivní".
- Vlečná vzdálenost (posunutí mezi požadovanou a skutečnou hodnotou) na konci věty se dá eliminovat na velikost určených cílových oken (= funkce "přesné zastavení").
- Funkce působí modálně a maže G1, G2, G3, G5, G6, G33.



Programování

Základní funkce Syntaxe:

G0 Rychlý chod s přesným zastavením (jemné polohovací okno) a naprogramování koncové polohy v kartézských souřadnicích.

Příklad:

```
:
N40 X100 Y100
N50 G0 X500 Y300
```

Výchozí poloha.

Najetí do cílové polohy v rychlém chodu.

G-kódy G00

Volitelné parametry Syntaxe:**G0**({NI PS | IPS | IPS1 | IPS2 | IPS3, }{POL})

kde

NI PS	Vypnutí přesného zastavení. Žádné zabrzdění na $v=0$ na konci věty.
IPS	Zapnutí přesného zastavení. Přitom je bráno v úvahu poslední pro rychlý chod nastavené okno přesného zastavení (standardně = jemné polohovací okno). Na konci věty zabrzdí řízení nejprve rychlost posuvu na $v=0$. Teprve když je toto polohovací okno dosaženo všemi zúčastněnými osami, zpracovává se další věta. Okno přesného zastavení pro posuvový provoz není nijak ovlivňováno polohovacím oknem rychlého chodu.
IPS1	Jako IPS, ale vždy s jemným polohovacím oknem .
IPS2	Jako IPS, ale vždy s hrubým polohovacím oknem .
IPS3	Jako IPS, ale bez kontroly polohovacího okna, pouze na konci věty proběhne zabrzdění na rychlost $v=0$. Bližší informace o přesném zastavení viz funkce G61/G62.
POL	Aktivuje programování polárních souřadnic a nastavuje polární úhel 1 na 0 a polární úhel 2 na 90 stupňů. Jak se programuje v polárních souřadnicích, naleznete v kap. 4.53.2 od strany 4-17.

Zvláštnosti a omezení:

- Funkce se dá programovat s osovými adresami nebo bez nich ve stejné větě.
- Příslušná rychlost se určuje pomocí parametrů stroje pro maximální osovou rychlost.
- Rychlost rychlého chodu se dá omezit pomocí signálu rozhraní kanálu "Redukovaný rychlý chod" (qCh_RedRap) na redukovanou hodnotu nastavenou v parametrech stroje.
- Rychlost rychlého chodu se dá omezit také s použitím funkce "zkušební rychlý chod", která se řídí pomocí signálu rozhraní kanálu "zkušební rychlý chod" (qCh_TestRap).
- Rychlost se dá ovlivňovat potenciometrem.

G-kódy G01

3.2.2 Lineární interpolace v posuvu

G01

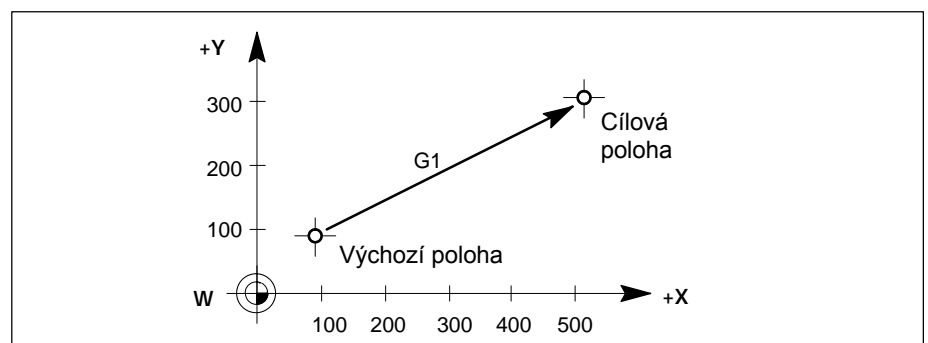
Následek

Najíždí do naprogramované polohy pomocí interpolace přímkou s efektivním posuvem (programovatelným s použitím F-adresy).

Pohyb je koordinován tak, aby všechny zúčastněné osy dosáhly zároveň naprogramovaného koncového bodu.

Přitom platí:

- Naprogramovaná hodnota posuvu (F) působí jako posuv dráhy; takže je při pohybech několika os podíl každé jednotlivé osy menší než F.
- Funkce působí modálně a maže G0, G2, G3, G5, G6, G33.



Programování

Základní funkce Syntaxe:

G1 Pohyb podle aktivního posuvu a programování koncové polohy v kartézských souřadnicích.

Příklad:

```

:
N40 X100 Y100 Výchozí poloha.
N50 G0 X500 Y300 F100 Najetí do cílové polohy s posuvem F100.

```

Volitelné parametry Syntaxe:

G1 ({IPS | IPS1 | IPS2 | IPS3, }{POL})

kde

IPS Zapnutí přesného zastavení, nezávislé na G61/G62. Přitom je bráno v úvahu poslední pro **posuvový provoz nastavené okno přesného zastavení** (standardně = jemné polohovací okno).

Na konci věty zabrzdí řízení nejprve rychlost posuvu na $v=0$. Teprve když je toto polohovací okno dosaženo všemi zúčastněnými osami, zpracovává se další věta.

IPS1 Jako IPS, ale vždy s **jemným polohovacím oknem**.

IPS2 Jako IPS, ale vždy s **hrubým polohovacím oknem**.

G-kódy G01

- I PS3 Jako IPS, ale bez kontroly polohovacího okna, pouze na konci věty proběhne zabrzdění na rychlost $v=0$.
Bližší informace o přesném zastavení viz funkce G61/G62.
- POL Aktivuje **programování polárních souřadnic** a nastavuje polární úhel 1 na 0 a polární úhel 2 na 90 stupňů.
Jak se programuje v polárních souřadnicích, naleznete v kap. 4.53.2 od strany 4-17.

Zvláštnosti a omezení:

- Funkce se dá programovat s osovými adresami nebo bez nich ve stejné větě.
- Pokud ještě není efektivní žádný posuv, musí být ve stejném řádku naprogramován posuv pomocí F-adresy.
Tento posuv pak zůstává efektivní, dokud není přepsán novou hodnotou posuvu.
- Naprogramovaná rychlost posuvu se dá omezit pomocí parametrů stroje.
- Rychlost posuvu se dá ovlivňovat potenciometrem posuvu.

G-kódy G02, G03

3.2.3 Kruhová-/spirálová-/spirálová-N interpolace

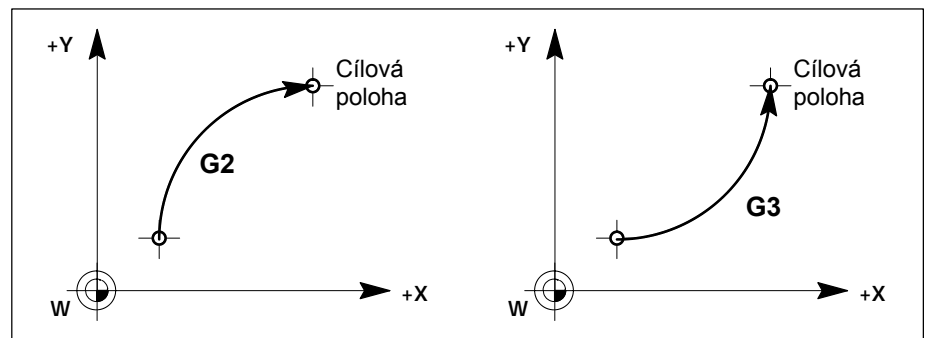
G02, G03

Následek

Najíždí do cílové polohy naprogramované v aktivní pracovní rovině pomocí interpolace po kruhové dráze s aktuálním posuvem (programovatelným s použitím F-adresy).

- G2: Kruhová dráha ve směru hodinových ručiček (pravotočivá)
- G3: Kruhová dráha proti směru hodinových ručiček (levotočivá)

Kromě toho mohou být mimo pracovní rovinu naprogramovány další osy, jejichž pohyby jsou současně lineárně interpolovány.



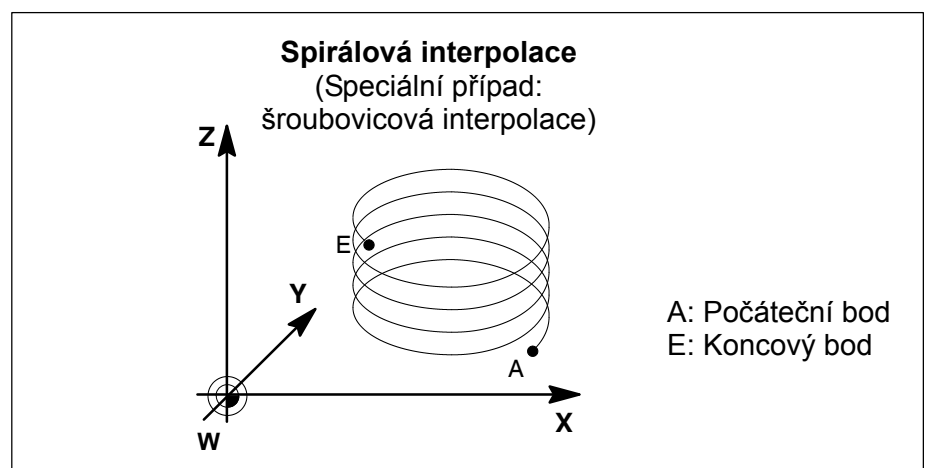
Kruhová interpolace:

Zúčastňují se výhradně osy aktivní pracovní roviny.

Spirálová interpolace:

Kromě os aktivní pracovní roviny se zúčastňuje ještě **jedna další** synchronní osa.

Pokud je tato osa kolmá k aktivní pracovní rovině (např. zbývající osa soustavy souřadnic obrobku), vytváří se šroubovicová dráha s konstantním stoupáním (spirálová interpolace).

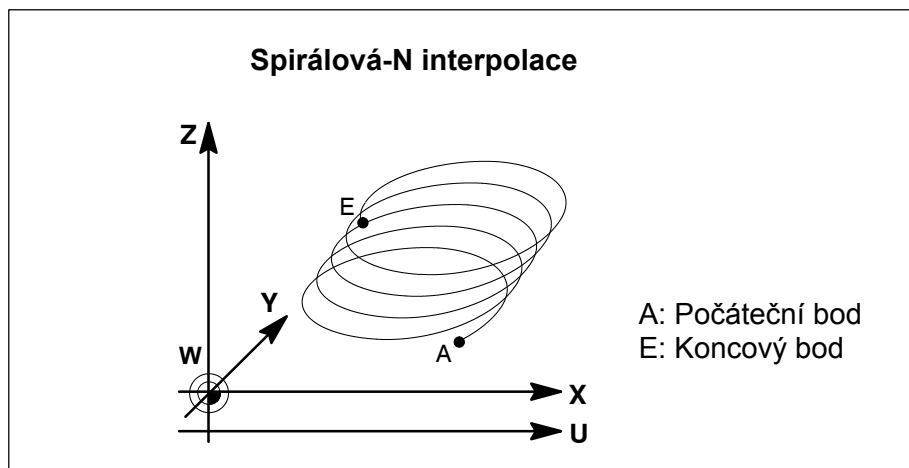


Spirálová-N interpolace:

Kromě os aktivní pracovní roviny se programuje **více než jedna další** (max. 6) synchronní osa.

G-kódy

G02, G03

**Platí:**

- Osy, které pojíždějí po kruhovém oblouku, jsou jednoznačně určeny vybranou pracovní rovinou (G17, G18, G19, G20).
- Naprogramovaná hodnota posuvu (F) působí jako posuv dráhy a vztahuje se zpravidla ke všem pojíždějícím osám ve větě; tak je při pohybech několika os podíl každé jednotlivé osy menší než F.
- Reálný posuv může být omezen maximálním možným radiálním zrychlením.
- Funkce G0, G1, G2, G3, G5, G6 tvoří modální skupinu a navzájem se tedy ruší.
- Kruhová dráha může být naprogramována pomocí:
 - programování poloměru nebo
 - programování středu.

G-kódy G02, G03

Programování poloměru

Na základě aktuální polohy jako počátečního bodu určete kruhový pohyb naprogramováním

- koncového bodu kruhu a
- poloměru kruhu.

Koncový bod kruhu přitom může být zadán jako absolutní nebo inkrementální hodnota polohy.

Základní funkce Syntaxe:

G2 <EP> R<hodnota> Kruhová dráha pravotočivá

G3 <EP> R<hodnota> Kruhová dráha levotočivá

kde

<EP> Souřadnice koncového bodu.

<Hodnota> Poloměr kruhové dráhy.

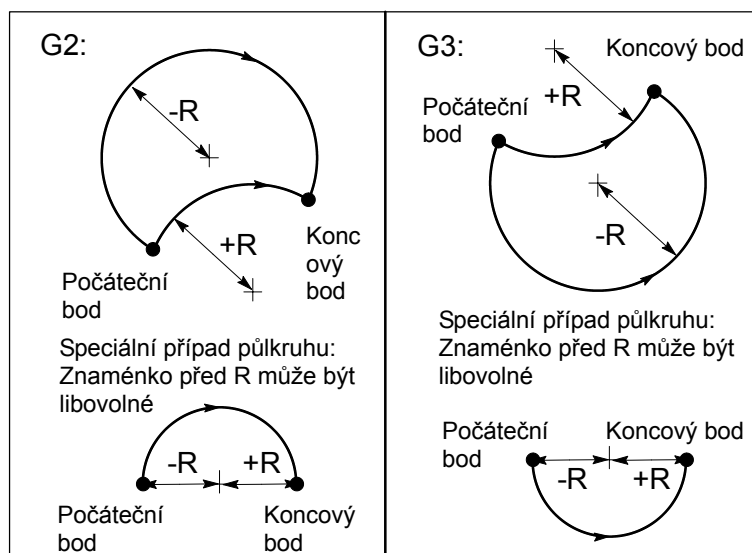
Poloměr musí být nejméně tak velký, jako poloviční vzdálenost mezi počátečním a koncovým bodem.

Pomocí znaménka určete, jestli při dvou možných kruhových drahách chcete pojíždět po dráze s menším nebo větším kruhovým obloukem:

kladná hodnota poloměru: kruhový oblouk ≤ 180 stupňů

záporná hodnota poloměru: kruhový oblouk > 180 stupňů.

Je-li poloměr roven přesně polovině vzdálenosti mezi počátečním a koncovým bodem, určuje **půlkruh** a znaménko hodnoty poloměru může být jakékoli.

**Zvláštnosti a omezení:**

- Není možné vytvářet celé kruhy.
- Nepřesně zadané poloměry opravuje řízení automaticky podle tolerancí nastavených v parametrech stroje. Jinak je následkem běhová chyba.

G-kódy G02, G03

Programování středu

Na základě aktuální polohy jako počátečního bodu určete kruhový pohyb naprogramováním

- koncového bodu kruhu a
- Střed kruhu.

Střed kruhu se programuje inkrementálně vzhledem k příslušnému počátečnímu bodu kruhu.

Základní funkce Syntaxe:

G2 <EP> <IP> Kruhová dráha pravotočivá

G3 <EP> <IP> Kruhová dráha levotočivá

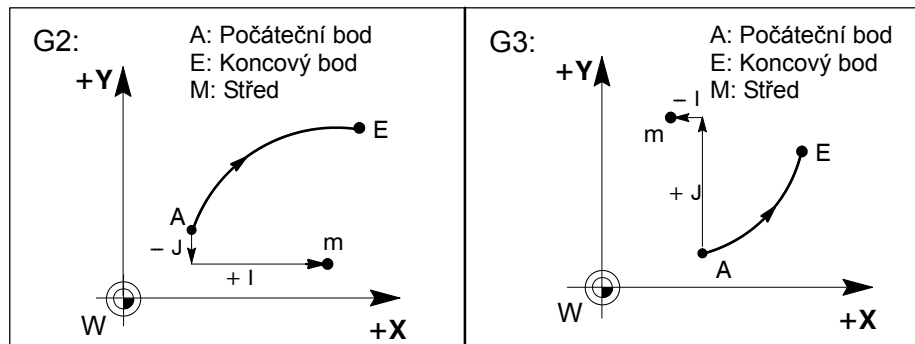
kde

<EP> Souřadnice koncového bodu.
Pokud jsou počáteční a koncový bod v rovině kruhu totožné, generuje se automaticky **plný kruh**.

<IP> Parametry interpolace I, J a K.
Definují pro každou osu vzdálenost počátečního bodu kruhu A od středu kruhu M. Znaménko je dáno v závislosti na směru vektoru z A do M.

Který interpolační parametr bude přiřazen které ose, se odvozuje z významu souřadnic v parametrech stroje.

Standardně platí: $I = M(X) - A(X)$ pro souřadnici X
 $J = M(Y) - A(Y)$ pro souřadnici Y
 $K = M(Z) - A(Z)$ pro souřadnici Z

**Zvláštnosti a omezení:**

- Jsou-li počáteční a koncový bod totožné, generuje řízení automaticky plný kruh.
- Nepřesně zadané středy opravuje řízení automaticky podle tolerancí nastavených v parametrech stroje.
Přitom se odpovídajícím způsobem posunuje poloha středu.
- Pokud mají být ve stejné větě naprogramovány parametry interpolace a navíc poloměr kruhu, je brán v úvahu výhradně poloměr kruhu (= programování poloměru).
- Jsou-li naprogramovány parametry interpolace, které se nehodí k vybrané rovině, hlásí řízení běhovou chybu.

Příklad: G17 G2 X5 I9 K7 (chyba: K se nehodí k rovině X/Y)

G-kódy G02, G03

Volitelné parametry

Syntaxe:

G2|G3(POL) <EP> R<hodnota>

Programování poloměru v
polárních souřadnicích

G2|G3(POL) <EP> <IP>

Programování středu v polárních
souřadnicích

kde

POL

Aktivuje programování polárních souřadnic pro souřadnice
koncového bodu a nastavuje polární úhel 1 na 0 a polární
úhel 2 na 90 stupňů.Jak se programuje v polárních souřadnicích, naleznete v
kap. 4.53.2 od strany 4-17.

Příklad: Programování poloměru (kartézské)

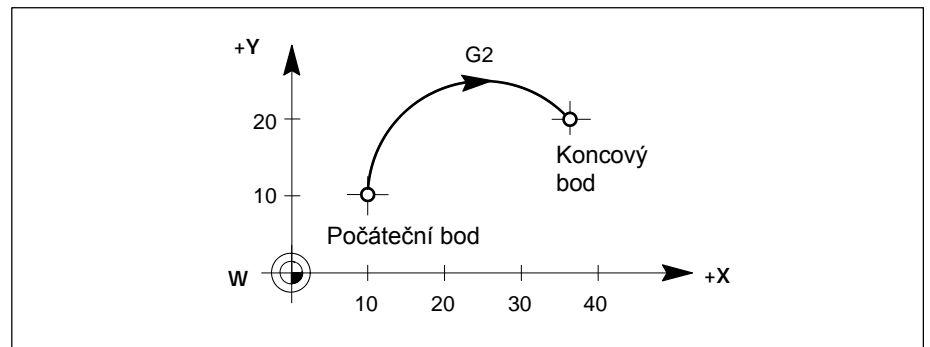
:

N40 G1 X10 Y10 F100

Najetí do výchozího bodu

N50 G2 X38 Y20 R15

Kruhová interpolace.

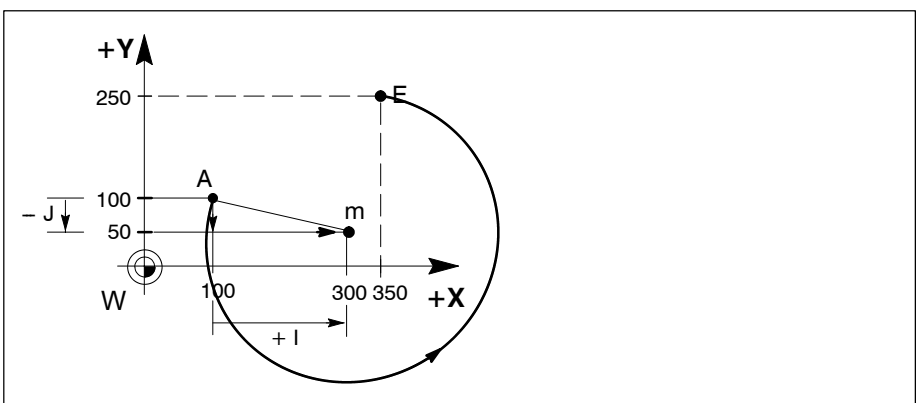


Příklad: Programování středu (kartézské)

:

N80 G1 X100 Y100 F100

Najetí do výchozí polohy.

N90 G90 G17 G3 X350 Y250
I 200 J- 50Kruhová interpolace proti směru hodi-
nových ručiček v rovině X/Y. Progra-
mování absolutních rozměrů pro
souřadnice koncového bodu je ak-
tivní.

G-kódy G04

3.2.4 Doba prodlevy

G04

Následek

Zastavuje provádění programu. Následující naprogramovaná věta je zpracována až po uplynutí naprogramované doby prodlevy.

Doba prodlevy začíná, když je kompletně zpracována předchozí NC-věta.

Otáčející se vřetena nebo pojíždějící pomocné osy se nezastavují. Synchronní osy mohou případně eliminovat doběh.

Doba prodlevy se dá naprogramovat v:

- v sekundách nebo
- v otáčkách vřetena.

Pro určení otáček vřetena se cyklicky zjišuje aktuální skutečný počet otáček hlavního vřetena a z něho vypočítávají absolvované otáčky. U vysoce dynamických vřeten proto může ve fázích zrychlení nebo brzdění docházet k určitým odchylkám mezi naprogramovanými a skutečně očekávanými otáčkami vřetena.

Pokud je zkonfigurovaným hlavním vřetenem analogové vřeteno (bez zpětné vazby počtu otáček), používá se pro výpočet místo hodnoty ze seznamu počtů otáček požadovaná hodnota počtu otáček.

Programování

Syntaxe:

G4(F<hodnota>) Doba prodlevy v sekundách.

G4(S<hodnota>) Doba prodlevy v otáčkách vřetena.

kde

<Hodnota> Zadané sekundy, resp. otáčky vřetena.

Při zadání "0" je věta G4 interně zrušena.

Zvláštnosti a omezení:

- Funkce musí být naprogramována v samostatné větě bez informací o dráze. V této větě jsou ještě přípustné pouze pomocné a přídavné funkce.
- Naprogramované otáčky vřetena se vztahují k hlavnímu vřetenu zkonfigurovanému v parametrech stroje nebo pomocí funkce MainSp (viz strana 4-70).


G-kódy G05

3.2.5 Tangenciální vstup kruhu

G05

Následek

Na základě naposledy naprogramovaného pojezdového pohybu vypočítává řízení automaticky pro větu G5 tangenciální vstup kruhu. Jako tangenciální přitom platí pouze přechod, který nevykazuje žádnou změnu směru.

 **Při spojení několika pohybů G5 ovlivňuje 1. vstupní tečna všechny následující konturové prvky G5.**

Funkce G0, G1, G2, G3, G5, G6 tvoří modální skupinu a navzájem se tedy ruší.



POZOR

Obráběcí značky na přechodu vět při spirálové-/spirálové-N interpolaci možné!

Vypočítaný tangenciální přechod se vztahuje pouze k rovině kruhu! Prostorová tečna může na přechodu vět skočit!

Programování

Syntaxe:

G5 <EP>

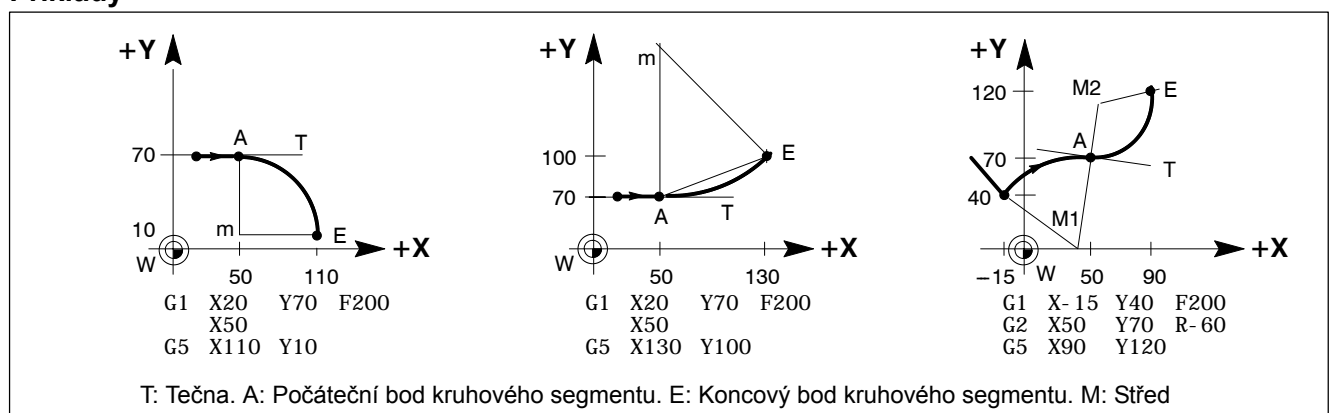
kde

<EP> Souřadnice koncového bodu kruhu.
Pokud jsou počáteční a koncový bod v rovině kruhu totožné, generuje se automaticky plný kruh.

Zvláštnosti a omezení:

- Programování G5 v provozním režimu "ruční zadávání" není možné.
- Před G5 musí být ve stejném programu naprogramována věta s pojezdovým pohybem.
- Bezprostředně před a za aktivní funkcí G5 nesmí být přepnuta aktuální rovina.

Příklady



G-kódy G06

3.2.6 Spline programování


G06

Ve srovnání s lineární interpolací nabízí spline interpolace s menším počtem opěrných bodů téměř stejnou kvalitu povrchu a přesnost kontur, protože se mezi body vypočítávají spojitě křivky.

IndraMotion MTX podporuje následující typy spline křivek:

- **Spline typu 0:** Spline křivka s programováním koeficientů (koeficienty polynomů ze systému CAD/CAM)
- **Spline typu 1:** C^1 -spojité kubické spline křivky s programováním opěrných bodů (tangenciální přechody v opěrných bodech)
- **Spline typu 2:** C^2 -spojité kubické spline křivky s programováním opěrných bodů (obloukové spojitě přechody v opěrných bodech)
- **Spline typu 3:** B-spline křivka s programováním kontrolních bodů (křivkový průběh v blízkosti opěrných bodů).

Požadovaný typ spline křivek se vybírá a inicializuje pomocí funkce "SplineDef" (SDF, viz strana 4-56). Pak je možné aktivovat spline programování s použitím funkce G6.

 **Podrobné informace o jednotlivých typech spline křivek naleznete v příručce "Funkční popis".**

Zvláštnosti a omezení:

Následující funkce **nemohou** být naprogramovány společně se spline funkcemi:

- Tenzorová orientace
- 2D oprava dráhy G41/G42
- Děrování/vysekávání s rozdělením dráhy
- Fasety a zaoblení
- Tangenciální vedení nástroje
- Přesné programování
- G5 s připojením na spline křivku
- Vymazat zbývající dráhu

Programování

Syntaxe:

G6

Aktivace typu dráhy "spline".

V závislosti na požadovaném typu programování je možné naprogramovat G6 pro různé typy spline funkcí s různými, modálně působícími parametry:

G-kódy G06

3.2.7 Spline s programováním koeficientů (spline typu 0)

Programování souřadnic/os:

Každá souřadnice kanálu se může podle potřeby pohybovat

- jako spline funkce zadáním koeficientů polynomu:
 $\langle \text{Název souřadnice} \rangle (\langle c_0 \rangle, \langle c_1 \rangle, \dots, \langle c_n \rangle)$ Programování jednotlivé souřadnice s koeficienty polynomu.
- nebo lineárně zadáním koncové polohy:
 $\langle \text{Název souřadnice} \rangle (\langle \text{Koncová poloha} \rangle)$ Programování koncové polohy jednotlivých souřadnic/os.

kde

$\langle \text{Název souřadnice} \rangle$: Název souřadnice/osy.
 $\langle c_0 \rangle, \langle c_1 \rangle, \dots, \langle c_n \rangle$: Koeficienty polynomu souřadnice. n odpovídá spline stupni nastavenému v "SplineDef".
 $\langle \text{Koncová poloha} \rangle$: Koncová poloha souřadnice.

Příklad:

Spl i neDef (3)
 G6 X(0. 1, 1. 25, 0. 5, 0. 73) Y30 B(0. 0, - 1. 0, 0. 1, - 0. 2)

Programování polynomu jmenovatele:

$DN(\langle g_0 \rangle, \langle g_1 \rangle, \dots, \langle g_n \rangle)$ Společný polynom jmenovatele pro všechny spline souřadnice. Přesný popis pro racionální Bezierovy spline křivky, racionální B-spline křivky (NURBS) a všechny kuželosečky.

kde

$\langle g_0 \rangle, \langle g_1 \rangle, \dots, \langle g_n \rangle$: Koeficienty polynomu jmenovatele. n odpovídá spline stupni nastavenému v "SplineDef".

Příklad:

Spl i neDef (3)
 G6 X(0. 1, 1. 25, 0. 5, 0. 73) B(0. 0, - 1. 0, 0. 1, - 0. 2) DN(1, 0, 1)

Programování orientačního vektoru:

Tento typ programování předpokládá aktivní transformaci os s vektorovou orientací ("Coord(..)").

$O1(\langle o_{10} \rangle, \langle o_{11} \rangle, \dots, \langle o_{1n} \rangle)$ Složka x orientačního vektoru
 $O2(\langle o_{20} \rangle, \langle o_{21} \rangle, \dots, \langle o_{2n} \rangle)$ Složka y orientačního vektoru
 $O3(\langle o_{30} \rangle, \langle o_{31} \rangle, \dots, \langle o_{3n} \rangle)$ Složka z orientačního vektoru

G-kódy G06

kde

$\langle o_{10} \rangle, \langle o_{11} \rangle, \dots, \langle o_{1n} \rangle$: Spline koeficienty složky x orientačního vektoru.

$\langle o_{20} \rangle, \langle o_{21} \rangle, \dots, \langle o_{2n} \rangle$: Spline koeficienty složky y orientačního vektoru.

$\langle o_{30} \rangle, \langle o_{31} \rangle, \dots, \langle o_{3n} \rangle$: Spline koeficienty složky z orientačního vektoru.

n vždy odpovídá spline stupni nastavenému v "SplineDef".

Příklad:

```

N00 ; Spline koeficienty pro orientaci vektoru
001 PI=3. 14159: PIH=PI/2: PIHQ=PIH*PIH: PIHC=PIHQ*PIH
N10 G1 F30000 X0 Y0 Z0 B90 C0
N20 SplineDef(3)
N30 Coord(1); Zapnutí 5-osového transformátoru s vektorovou orientací
N40 G6 PL[PIH]; PL viz programování délek spline parametrů
N50 O1(1, 0, -3/PIHQ, 2/PIHC) O3(0, 1, (3-PI)/PIHQ, (-2+PIH)/PIHC)
N60 O1(0, 0, 3/PIHQ, -2/PIHC) O3(1, 0, (-3+PIH)/PIHQ, (2-PIH)/PIHC)
N70 O(0, 1, 0); Normální vektorová orientace
N80 G1
N90 Coord(0)

```

Programování délek spline parametrů:

Délka spline parametru je délka definičního intervalu w . w nabývá hodnot $0 \dots w_e$. Hodnota w_e je modální a zůstává platná pro všechny NC-věty až do zrušení G6.

V první pojezdové větě po G6 musí být naprogramováno PL, jinak dojde k běhové chybě.

{PL< w_e >} Volitelné programování délek spline parametrů.

kde

$\langle w_e \rangle$: libovolná hodnota > 0

Příklad:

```
G6 X(0.1, 1.25, 0.5, 0.73) B(0.0, -1.0, 0.1, -0.2) PL0.6
```

($X = 0.1 + 1.25 w + 0.5 w^2 + 0.75 w^3$ a

$B = 0.0 - 1.0 w + 0.1 w^2 - 0.2 w^3$ kde

w nabývá hodnot $0 \dots 0.6$)

G-kódy G06

3.2.8 C¹- a C²-spojité kubické spline funkce (spline typu 1 a 2)

Programování souřadnic/os:

Programují se koncové body kanálových souřadnic. Všechny členy *<Members>* uvedené ve "SplineDef" se pohybují po spline křivce, zbývající souřadnice neuvedené ve "SplineDef" se pohybují lineárně.

<Název souřadnice>(*<Koncová poloha>*) a/nebo *<Název osy>*(*<Koncová poloha>*) a/nebo *<Souřadnice orientace>*(*<Koncová orientace>*) Programování jednotlivých souřadnic / os / souřadnic orientace a jejich hodnot.

kde

<Název souřadnice>: Název souřadnice.

<Název osy>: Název osy.

<Koncová poloha>: Koncová poloha souřadnice/osy.

<Souřadnice orientace>: Souřadnice orientace "O" nebo "phi" a "theta".

<Koncová orientace>: Koncová orientace v polárním úhlu nebo kartézských souřadnicích.

Příklad:

Souřadnice x, y, z a souřadnice orientace phi, theta.

SplineDef (2203, x, y, z, phi, theta)

G6 x10 y20 phi 20 theta30

Příklad:

Osy X,Y,U

SplineDef (1213, X, Y)

G6 X10 Y10 U20

lineárně)

(X a Y se pohybují jako spline křivky, U

Počáteční a koncové podmínky:

Syntaxe:

SBC(*<typ>*{,*<hodnoty>*})

Okrajové podmínky pro počáteční bod spline sekvence

C¹ s 3 počátečními podmínkami

C² s 5 počátečními podmínkami

EBC(*<typ>*{,*<hodnoty>*})

Okrajové podmínky pro koncový bod spline sekvence

C¹ s 3 počátečními podmínkami

C² s 5 počátečními podmínkami

G-kódy

G06

kde

<Typ>

Standardní: 2

1: (platí pro C^1 a C^2)

Zadání tečného směru v počátečním, resp. koncovém bodě spline sekvence. Pro každý spline člen se zadává hodnota v seznamu <Hodnoty>.

2: (platí pro C^1 a C^2)

Zadání druhé derivace v počátečním, resp. koncovém bodě spline sekvence. Pro každý spline člen se zadává hodnota v seznamu <Hodnoty>.

3: (platí pro C^2)

De-Boorova okrajová podmínka spojuje druhé derivace na prvních, resp. posledních dvou opěrných bodech. <Hodnoty> je zde většinou 1.

4: (platí pro C^2)

Periodická okrajová podmínka: poslední a první bod spline sekvence jsou stejné. SBC(4) vyžaduje nezbytně EBC(4), resp. EBC(4) vyžaduje nezbytně SBC(4). Celá spline sekvence musí ležet v oblasti náhledu, jinak je generována běhová chyba.

11: (platí pro C^1 a C^2)

Pro počáteční podmínku: první spline křivka začíná tečně k předchozí lineární větě.

Pro koncovou podmínku: poslední spline křivka ústí tečně do následující lineární věty.

<Hodnoty>

Standardní: 0, ..., 0

Všechny údaje v <Hodnoty> dohromady udávají směr a velikost počáteční, resp. koncové tečny nebo druhé derivace v počátečním, resp. koncovém bodě.

Pro každý spline člen může být nastavena kladná nebo záporná hodnota.

Typ 11 nepotřebuje žádné <Hodnoty>.

Příklad:

SBC(1, 1. 0, 1. 0, 0. 2)

když SplineDef(1213,X,Y,B)

Délka spline parametrů:

Délku spline parametrů vypočítává NC ze zadaných opěrných bodů. Přitom se používá postup zadaný ve Spline-Id (parametrizace). Délka spline parametrů může být v případě potřeby taktéž naprogramována:

{PL< w_e >}

Volitelné programování délky spline parametrů, když má být přepsán výběr <parametrizace> (viz strana 4-56 "SplineDef", parametr <Id>).

kde

< w_e >:

libovolná hodnota > 0

G-kódy G06

3.2.9 B-spline křivky (NURBS) (spline typu 3)

Programování souřadnic/os:

Programují se koncové body kanálových souřadnic (kontrolní body). Všechny <členy> uvedené ve "SplineDef" se pohybují po spline křivce, zbývající souřadnice neuvedené ve "SplineDef" se pohybují lineárně.

<Název souřadnice>(<Koncová poloha>) a/nebo Programování jednotlivých kontrolních bodů
<Název osy>(<Koncová poloha>) a/nebo (souřadnic/os) a jejich hodnot.
<Souřadnice orientace>(<Koncová orientace>)

kde

<Název souřadnice>: Název souřadnice.

<Název osy>: Název osy.

<Koncová poloha>: Koncová poloha souřadnice/osy.

<Souřadnice orientace>: Souřadnice orientace "O" nebo "phi" a "theta".

<Koncová orientace>: Koncová orientace v polárním úhlu nebo kartézských souřadnicích.

Příklad: Souřadnice x, y, z a souřadnice orientace

SplineDef (3103, x, y, z, 0) G6 x10 y20 z30 0(0.1, 0, 1.0)

Příklad: Osy X, Y, U

SplineDef (3102, X, Y) G6 X10 Y10 U20 (X a Y se pohybují jako spline křivky, U lineárně)

Délka spline parametrů:

Délku spline parametrů vypočítává automaticky NC interně ze zadaných kontrolních opěrných bodů. Přitom se používá postup zadaný ve SplineId (parametrizace = 1). Délka spline parametrů může být v případě potřeby taktéž naprogramována:

{PL< w_e >} Volitelné programování délky spline parametrů, když má být přepsán výběr <parametrizace> (viz strana 4-56 "SplineDef", parametr <ld>).

kde

< w_e >: libovolná hodnota > 0

Váha spline bodu pro kontrolní bod u B-spline křivek:

{PW< w_e >} Volitelné programování vah bodů. Tak je možné měnit spline křivku v okolí kontrolního bodu.

kde

< w_e >: Standardní: 10 < w_e < 1: odtlačuje spline křivku od kontrolního bodu w_e > 1: přitahuje spline křivku ke kontrolnímu bodu

G-kódy

G06

Příklad: Souřadnice x, y, z a souřadnice orientace

Spl i neDef (3103, x, y, z, 0)

G6 x10 y20 z30 0(0. 1, 0, 1. 0) PW2. 3

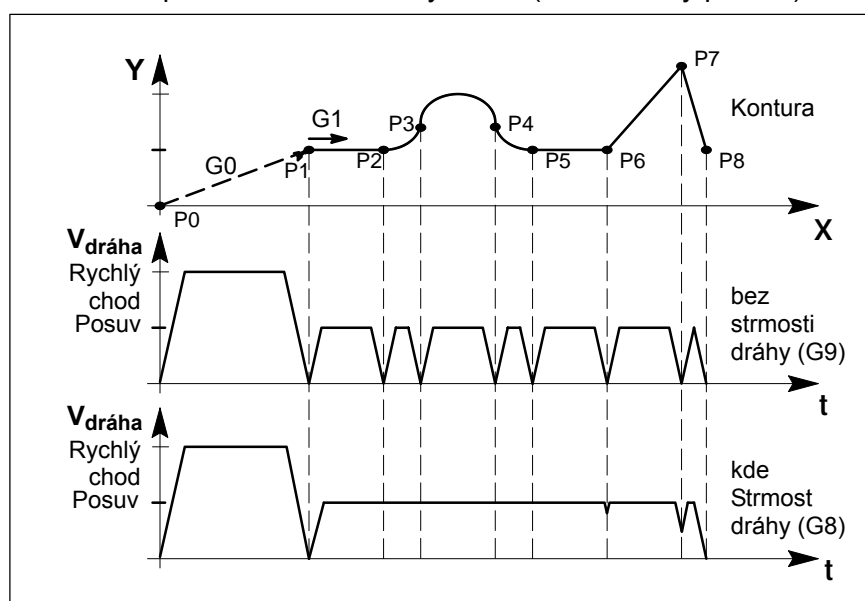
G-kódy G08 G09

3.2.10 Zapnutí strmosti dráhy Vypnutí strmosti dráhy

G08
G09

Následek

- Bez "strmosti" provádí řízení na začátku a na konci pojezdové větý kompletní Up- a Down-Slope (lineární změna rychlosti) na $v=0$. Tím se zmenšuje odchylka kontury na přechodu vět, ale zároveň to vyžaduje delší dobu obrábění.
- Se "strmostí dráhy" se řízení pokouší i na přechodu vět generovat pokud možno konstantní rychlost ve velikosti naprogramovaného posuvu. Tím se zkracuje doba obrábění. Že se přitom kontura v rozích poněkud "obrousí", může při určitých obráběcích procedurách navíc výhodné (rovnoměrný povrch).



Při určování optimálního rychlostního profilu bere řízení v úvahu kromě hodnoty pro kapacitu osového skoku (MP 1010 00011) také počet následujících programových vět (náhled vět; MP 7060 00110 - 7060 00130), protože musí být kdykoli zaručeno zabrzdění v rámci dynamické vůle stroje.

Pro omezení vznikajícího zkreslení kontur na skutečných rozích nesmí být nastavena příliš velká kapacita osového skoku. Na druhé straně vede příliš malá kapacita osového skoku k nežádoucímu zabrzdění při malých ohybech kontur (kvazikontinuální přechody).

Vyhlazení zrychlení dráhy se dá pro G8 zadat jako standardní nastavení v parametru stroje 7050 00320 (funkce SHAPE, viz též strana 3-23).

G-kódy G08 G09

Programování**Základní funkce** Syntaxe:

- G8 Zapnutí strmosti dráhy.
G9 Vypnutí strmosti dráhy.

Volitelné parametry Syntaxe:

G8({DTT<Úhel>})

kde

<Úhel> Přechodový úhel mezi dvěma úseky kontury.
Ohyby kontur s menšími přechodovými úhly, než jsou zde naprogramovány, přejíždí řízení s konstantní rychlostí dráhy, pokud to dovoluje brzdná dráha, která je k dispozici v rámci náhledu vět. Vliv kapacity osového skoku se tedy omezuje na velké přechodové úhly.
Rozsah hodnot: 0 až 50 stupňů.

Zvláštnosti a omezení:

- Pokud je aktivováno přesné zastavení, proběhne navzdory aktivní funkci G8 po každé větě zabrzdění na $v=0$.
- Pomocné funkce mohou redukovat účinnost funkce G8, pokud doba jejich provádění včetně potvrzení není podstatně kratší než doba interpolace vět. V případě potřeby je nutné prodloužit dráhu pojezdu vět nebo zmenšit posuv.
- Strmost dráhy působí jen na synchronní (obráběcí) osy.
- Funkce G8 a G9 tvoří spolu s lineárními změnovými funkcemi modální skupinu a vzájemně se ruší.

G-kódy G8(SHAPE...) G9(SHAPE...) G9(ASHAPE...) G9(X... , Y... , ...)

3.2.11 Řízení rychlosti s omezením škubání

G8(SHAPE...)
G9(SHAPE...)
G9(ASHAPE...)
G9(X... , Y... , ...)

Následek

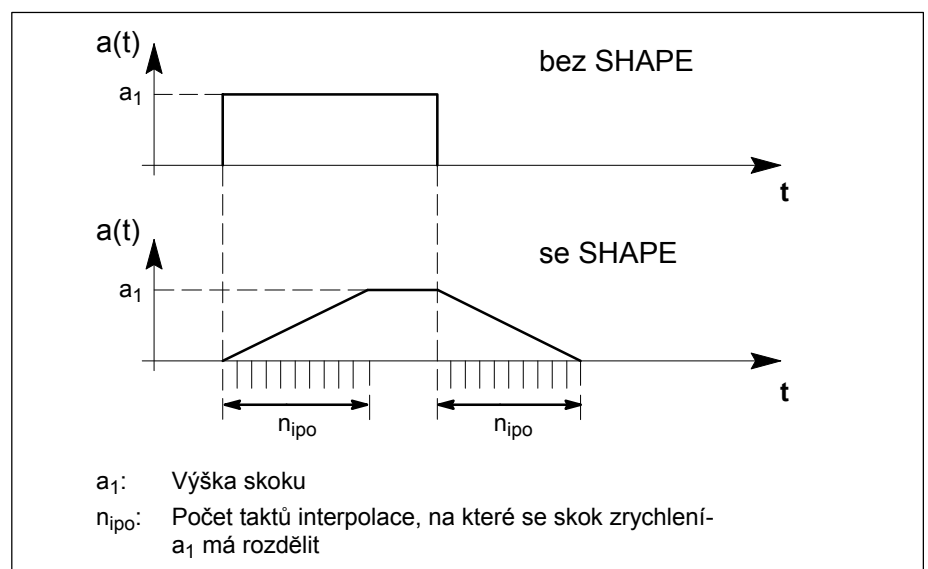
Řízení rychlosti s omezením škubání vyhlazuje skoky v průběhu zrychlení dráhy během několika taktů interpolace. Počet taktů interpolace je programovatelný.

Tímto způsobem je možné docílit hladkých přechodů rychlosti (omezení škubání).

Řízení rychlosti s omezením škubání je možné v

- "posuvovém provozu" (při aktivní funkci G8), stejně jako v
- "polohovacím provozu" (při aktivní funkci G9).

Řízení rychlosti s omezením škubání se také označuje jako funkce SHAPE.



Rozdělení zrychlení s/bez SHAPE

G-kódy G8(SHAPE...) G9(SHAPE...) G9(ASHAPE...) G9(X... , Y... , ...)

Shape pro posuvový provoz

Syntaxe:

G8(SHAPE<Hodnota>, {DTT})

kde

<Hodnota> 0: Zapnutí Shape pro posuvový provoz s hodnotou uloženou v parametrech stroje (standardní).

1: Vypnutí Shape pro posuvový provoz.

2 až 100 (celočíslné):

Počet taktů interpolace, na které má řízení **lineárně** rozdělit skok (řád SHAPE).

{DTT} Viz strana 3-21.

 **Změna řádu SHAPE vede vždy k rychlosti přechodu vět v = 0.**

Příklad:

N30 G8(SHAPE10) Zapnout SHAPE v posuvovém provozu a rozdělit skoky zrychlení na 10 IPO-taktů.

:

:

N50 G8(SHAPE0) Zapnout SHAPE v posuvovém provozu a rozdělit skoky zrychlení podle MP 7050 00320.

:

:

N70 G8(SHAPE1) Vypnout SHAPE v posuvovém provozu.

Shape pro polohovací provoz

Syntaxe:

G9(SHAPE<hodnota>) Zapnout lineární SHAPE pro polohovací provoz.

kde

<Hodnota> Počet taktů interpolace, na které má řízení **lineárně** rozdělit skok zrychlení.

Rozsah hodnot: 2 ... 100 taktů; celočíselné.

G9(SIN<hodnota>) Zapnout SHAPE tvaru \sin^2 pro polohovací provoz.

kde

<Hodnota> Počet taktů interpolace, na které má řízení rozdělit skok zrychlení ve tvaru **\sin^2** .

Jsou přípustné následující hodnoty:

5: 5 IPO-taktů

10: 10 IPO-taktů

15: 15 IPO-taktů

20: 20 IPO-taktů

40: 40 IPO-taktů

G-kódy	G8(SHAPE...) G9(SHAPE...) G9(ASHAPE...)	G9(X... , Y... , ...)
	G9(ASHAPE)	Přiřazuje každé ose kanálu příslušný osový řád SHAPE (počet IPO-taktů) z MP 1003 00008 a vypočítává vždy po větách výsledný řád SHAPE pro jednotlivé větvy dráhy.
	G9(<Par ₁ > {,<Par ₂ >,...})	Programování řádu SHAPE po osách. Pro nenaprogramované osy je řád SHAPE převzat z MP 1003 00008.
	kde <Par _x >	Logický název osy s řádem SHAPE, který má být přiřazen této ose (maximálně 100).
	Příklad: N30 G9(X4, Y6, Z10)	Řád SHAPE (osa X) = 4 Řád SHAPE (osa Y) = 6 Řád SHAPE (osa Z) = 10

Výsledný řád Shape

Výsledný řád Shape dráhy S_b je maximum z efektivních řádů Shape S_i^{eff} všech os zúčastněných na interpolaci:

$$S_b = \max \{S_1^{eff}, \dots, S_n^{eff}\}$$

Efektivní řády Shape os S_i^{eff} se vypočítají z naprogramovaných řádů Shape podle vzorce:

$$S_i^{eff} = S_i^p \frac{a_i^{eff}}{a_i^{max.}}$$

Zde je

S_i^p řád Shape osy naprogramovaný podle G9(...).

a_i^{eff} efektivní osové zrychlení v aktuální NC-větě.

Hodnota závisí při lineární interpolaci na aktuální části dráhy osy. Při jiných typech interpolace (např. kruhová či spirálová) odpovídá osovému zrychlení naprogramovanému obvykle pomocí "AxAcc". U funkcí "šikmá rovina" nebo "spojení os" se efektivní osové zrychlení obecně ještě zmenšuje oproti hodnotě "AxAcc"!

$a_i^{max.}$ maximální osové zrychlení (z MP 1010 00001).

 **"AxAcc" nemění maximální osové zrychlení z MP 1010 00001!**

G-kódy G8(SHAPE...) G9(SHAPE...) G9(ASHAPE...) G9(X... , Y... , ...)

Souvislost mezi řádem Shape a přírůstkem zrychlení

Pomocí řádů Shape os S_i^{eff} se pro každou osu definuje maximální přírůstek zrychlení $r_i^{\text{max.}}$ (derivace zrychlení podle času), které není při žádném pohybu překročeno.

Přírůstek zrychlení je definován jako:

$$r_i^{\text{max.}} = \frac{a_i^{\text{max.}}}{S_i^p T_{\text{ipo}}} \quad T_{\text{ipo}}: \text{Doba taktu interpolátoru}$$

Příklad:

Nech má osa X maximální zrychlení (MP 1010 00001) 10 m/s^2 . Naprogramovaný řád Shape os nech je 5 a takt interpolace 4 ms.

Podle výše uvedeného vzorce je tedy pro osu X definován maximální přírůstek zrychlení 500 m/s^3 .

G-kódy G16

3.2.12 Žádná rovina

G16

Následek

Deaktivuje případně aktivní rovinu.

Platí:

- Pokud je hlavní nebo vedlejší osa vyjmuta z kanálu (např. v souvislosti s funkcí "předání os"), vypne řízení automaticky vybranou rovinu a aktivuje funkci G16.
Kruhová/spirálová interpolace není v tomto kanálu možná tak dlouho, dokud není opět zvolena platná rovina.
- Pro některé aplikace, typy strojů nebo obráběcí jednotky není konfigurace roviny nezbytná, když není zapotřebí kruhová ani spirálová interpolace (např. kanály jen s jednou obráběcí osou).
V MP 7010 00030 (význam souřadnice) je pak možné zadat pro každou osu "999" (bez pracovní technického významu).
- Není-li v zapnutém stavu kanálu (MP 7060 00010 a 7060 00020) zadána žádná funkce roviny (G17, G18, G19, G20), aktivuje se automaticky funkce G16 pro příslušný kanál.
- Funkce G16, G17, G18, G19 a G20 tvoří modální skupinu a navzájem se tedy ruší.

Programování

Syntaxe:

G16

Zvláštnosti a omezení:

- Při aktivní opravě dráhy frézy (G41/G42) nesmí být aktivní rovina přepnuta ani deaktivována.

G-kódy G17, G18, G19

3.2.13 Přepnutí roviny

G17, G18, G19

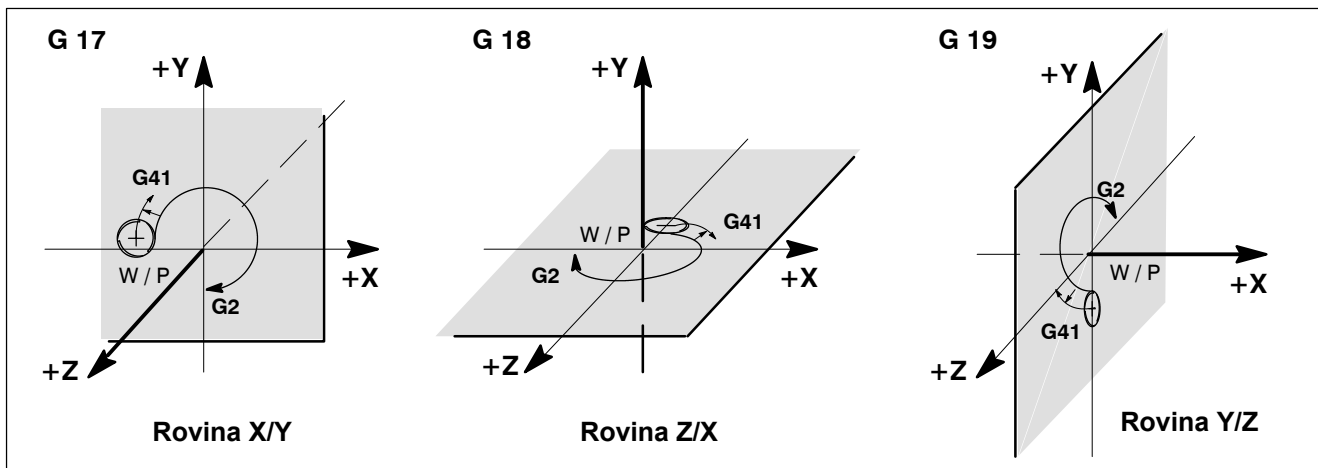
Následek

Pro **běžný kartézský standardní obráběcí stroj** je osami X, Y a Z vymezena kartézská soustava souřadnic obrobku (WCS). S použitím vždy 2 souřadnicových os soustavy souřadnic obrobku je možné definovat celkem 3 různé základní roviny.

S použitím funkce přepnutí roviny se vybírá aktivní rovina. Ta je definována hlavní a vedlejší souřadnicí, stejně jako příslušnou přísuvovou osou, která je kolmá k této rovině.

Přitom se programují funkce G17, G18, resp. G19 bez parametrů.

Pomocí G17, G18 a G19 se určují tyto souřadnice a aktivní rovina následujícím způsobem:



	Hlavní souřadnice	Vedlejší souřadnice	Přísuvová souřadnice	Aktivní rovina
G17	X	Y	Z	XY
G18	Z	X	Y	ZX
G19	Y	Z	X	YZ

Programování

Syntaxe:

G17

G18

G19

G-kódy

G17, G18, G19

Zvláštnosti a omezení:

- Při aktivní opravě dráhy frézy (G41/G42) nesmí být aktivní rovina přepnuta ani deaktivována.
- Pro parametry interpolace I, J, K při G02/G03 a G33 platí:
 - I se vztahuje k ose X
 - J se vztahuje k ose Y
 - K se vztahuje k ose Z
- Oprava délky nástroje se přiřazuje pomocí G47(ActPlane) aktivní rovině následujícím způsobem:
 - L1 hlavní souřadnici
 - L2 vedlejší souřadnici
 - L3 přísuvové souřadnici.
 Toto přiřazení se po přepnutí aktivní roviny aktualizuje.
- Funkce G16, G17, G18, G19 a G20 tvoří modální skupinu a navzájem se tedy ruší.

Přiřazení nezávislé na názvech os

Názvy os se dají v IndraMotion MTX libovolně nastavovat. Výběr os, které mají definovat soustavu souřadnic obrobku WCS, se proto provádí pomocí parametru stroje 7010 00030 (význam souřadnice). Ten stanovuje **nezávisle** na zkonfigurovaných názvech os, které osy vždy definují souřadnici X, Y a Z soustavy souřadnic obrobku:

Osa s "významem osy X": souřadnice X ve WCS

Osa s "významem osy Y": souřadnice Y ve WCS

Osa s "významem osy Z": souřadnice Z ve WCS

Všeobecně tedy platí pro G17, G18 a G19 následující přiřazení:

	Hlavní souřadnice	Vedlejší souřadnice	Přísuvové souřadnice	Aktivní rovina
G17	Souřadnice X WCS	Souřadnice Y WCS	Souřadnice Z WCS	Souřadnice XY WCS
G18	Souřadnice Z WCS	Souřadnice X WCS	Souřadnice Y WCS	Souřadnice ZX WCS
G19	Souřadnice Y WCS	Souřadnice Z WCS	Souřadnice X WCS	Souřadnice YZ WCS

G-kódy G17(...), G18(...), G19(...)

3.2.14 Rozšíření přepínání rovin**G17(...), G18(...), G19(...)****Následek**

Při rozšířeném přepínání rovin se kromě výběru aktivní roviny ještě určuje, které osy mají aktuálně definovat soustavu souřadnic obrobku WCS.

Tím vzniká v průběhu programu možnost definovat soustavu souřadnic obrobku odlišnou od nastavení v parametrech stroje a na jejím základě vybrat aktuální rovinu.

Rovina se vybírá podle následujícího přehledu:

	Hlavní souřadnice	Vedlejší souřadnice	Přísuvová souřadnice	Aktivní rovina
G17	Souřadnice X WCS	Souřadnice Y WCS	Souřadnice Z WCS	Souřadnice XY WCS
G18	Souřadnice Z WCS	Souřadnice X WCS	Souřadnice Y WCS	Souřadnice ZX WCS
G19	Souřadnice Y WCS	Souřadnice Z WCS	Souřadnice X WCS	Souřadnice YZ WCS

Programování

Syntaxe:

G17(<osa1>, <osa2>, <osa3>)

G18(<osa1>, <osa2>, <osa3>)

G19(<osa1>, <osa2>, <osa3>)

kde

<Osa1> "Souřadnice X" soustavy souřadnic obrobku WCS

<Osa2> "Souřadnice Y" soustavy souřadnic obrobku WCS

<Osa3> "Souřadnice Z" soustavy souřadnic obrobku WCS

Zvláštnosti a omezení:

- Osy mohou být naprogramovány s logickým (vztaheným ke kanálu) nebo fyzickým (pro celý systém) osovým označením a musí být vzájemně různé. Zadávání je možné přímo nebo jako proměnná CPL typu String.
- Naprogramované osy musí být aktuálně přiřazeny kanálu.
- Při programování stačí zadat jako parametry pouze ty osy, které mají být nově zkonfigurovány. Pro všechny vynechané osy zůstávají příslušné souřadnice WCS beze změny.

Příklad:

G17(, , W)

Souřadnice Z ve WCS je definována osou W. Souřadnice X a Y ve WCS zůstávají nezměněny.

Aktivní rovina je definována souřadnicemi X a Y ve WCS.

G-kódy

G17(...), G18(...), G19(...)

- Naprogramování prázdného výrazu v závorce obnovuje soustavu souřadnic WCS zkonfigurovanou v MP 7010 00030 a potom aktivuje vybranou rovinu.

Příklad:

G17()

- Funkce nesmí být naprogramována při aktivní opravě dráhy frézy (G41/G42).
Při aktivní opravě dráhy frézy nesmí být rovina obecně přepnuta ani deaktivována.
- Při aktivní transformaci os nesmí být naprogramováno žádné rozšířené přepínání roviny, protože soustava souřadnic WCS je definována prostorovými souřadnicemi.
- Funkce G16, G17, G18, G19 a G20 tvoří modální skupinu a navzájem se tedy ruší.

G-kódy G20

3.2.15 Volný výběr roviny (nezávislý na WCS)

G20

Následek

Volný výběr roviny je nezbytný pro aplikace, při kterých musí být aktivní rovina odpojena od aktuální soustavy souřadnic obrobku WCS. Umožňuje definování s použitím libovolných os nezávisle na tom, jestli tyto osy aktuálně definují také WCS.

Díky tomu vzniká mimo jiné možnost vypočítávat opravy nástroje (G41, G42, G47) v hlavní, vedlejší a přísluvové souřadnici aktuální roviny, zatímco transformace obrobku (umístění, opravy upnutí) se zároveň vztahují k soustavě souřadnic obrobku, která je definována jinými osami.

Programování

Syntaxe:

G20(<osa1>, <osa2>{, <osa3>})

kde

- <Osa1> Hlavní souřadnice definované roviny; bude přiřazena interpolačnímu parametru "I".
- <Osa2> Vedlejší souřadnice definované roviny; bude přiřazena interpolačnímu parametru "J".
- <Osa3> Přísluvová souřadnice definované roviny.
Může být naprogramována, když je pro opravu nástroje "G47(ActPlane)" nezbytné stanovit přísluvovou souřadnici.

Zvláštnosti a omezení:

- Osy mohou být naprogramovány s logickým (vztáženým ke kanálu) nebo fyzickým (pro celý systém) osovým označením.
- Oprava dráhy frézy G41/G42 se vztahuje k vybrané rovině. Proto se smí aktivní rovina přepínat pouze při deaktivované opravě dráhy frézy (G40). Jinak je ohlášena chyba.
- Oprava délky nástroje se přiřazuje pomocí "G47(ActPlane)" aktivní rovině následujícím způsobem:
 - L1 hlavní souřadnici
 - L2 vedlejší souřadnici
 - L3 přísluvové souřadnici.Toto přiřazení se po přepnutí aktivní roviny aktualizuje.
- Funkce G16, G17, G18, G19 a G20 tvoří modální skupinu a navzájem se tedy ruší.

G-kódy G33

3.2.16 Řezání závitů

G33

Následek

Umožňuje řezání

- podélných závitů (řezný pohyb paralelní s hlavní osou aktivní roviny),
- spirálních závitů (řezný pohyb paralelní s vedlejší osou aktivní roviny),
- kónických závitů (na řezném pohybu se podílí hlavní stejně jako vedlejší osa aktivní roviny).

Funkce G33 je možná s vřeteny s regulovanými otáčkami stejně jako s regulovanou polohou.

Řezný pohyb je vždy spojen s hlavním vřetenem aktivním v odpovídajícím kanálu (viz strana 4-70).

Rychlost posuvu řezného pohybu závisí na aktuálním počtu otáček vřetena a příslušném naprogramovaném stoupání (pevné, variabilní; viz pod "Programování").

Zvláštnosti:

- možnost výroby jedno- i vícechodého závitu
- možnost naprogramování konstantního a proměnlivého stoupání závitu
- speciálně nastavitelná dynamika během řezání
- programovatelný rychlý zpětný pohyb
- možnost výroby zřetězeného závitu.

☞ **Potenciometr posuvu nemá během funkce G33 žádný efekt.**

☞ **Řezání závitů je jako kruhová interpolace (G2, G3) funkcí závislou na aktivní rovině (G17...G20).**

Chování funkce "řezání závitů" se normálně určuje pomocí parametrů stroje 7050 006xx.

V jednotlivých případech nebo během prvního uvedení do provozu je výhodné, když lze rychle přizpůsobit jednotlivé dílčí oblasti. Proto je k dispozici funkce "ThreadSet" (popis viz od strany 4-70).

"ThreadSet" umožňuje

- Přizpůsobení dynamiky a zpětného pohybu
- Přepínání provozních režimů vřetena (regulace počtu otáček, regulace polohy)
- Nastavení signálu na rozhraní kanálu (možnost konfigurace v rámci signálů "aktivní funkce" podle parametrů stroje).

G-kódy G33

Programování

Syntaxe:

G33 <EP>, <pevné stoupání>{, <var. stoupání>}{, <po áte ní
úhel>}

kde

<EP>

Osové souřadnice hlavní a vedlejší osy aktivní roviny.
Aktivní rovina je určena pomocí G17, G18, G19 nebo
G20.

Příklad:

Pro G18 se definuje aktivní rovina obvykle pomocí os
Z (hlavní osa) a X (vedlejší
osa).<Pevné stoupání> Určuje dráhu (v mm), která má být uražena ve směru
hlavní nebo vedlejší osy na otáčku vřetena. Hodnota
se programuje s použitím interpolačního parametru,
který je platný v aktivní rovině (I, J nebo K).U kónických závitů se musí zadané stoupání závitu
vždy vztahovat ke směru hlavního břítu.

Příklad:

Pro G18 je parametr K přiřazen hlavní ose a I vedlejší
ose. Pro podélný závit (stoupání ve směru hlavní osy)
se programuje pevné stoupání závitu s adresou K.

<var. stoupání> Volitelný parametr s adresou DF.

Definuje přírůstek/úbytek stoupání na otáčku vřetena
v mm.Programování: "**DF**<hodnota>" s <hodnotou> v mm.

<Počáteční úhel> Volitelný parametr.

Pokud <počáteční úhel> není naprogramován, má
hodnotu 0 stupňů.Počáteční úhel (posunutí) je zapotřebí pro vícechodé
závity. Jako adresa slouží interpolační parametr, který
není přiřazen aktivní rovině.

Příklad:

Pro G18 jsou adresy I a K přiřazeny rovině. Adresa
počátečního úhlu je tedy J.

G-kódy

G33

Příklad: Podélný závit

G91 G18 G8 M3 S1000

Aktivovat programování relativních rozměrů.
Aktivovat rovinu Z/X.

G0 X-10

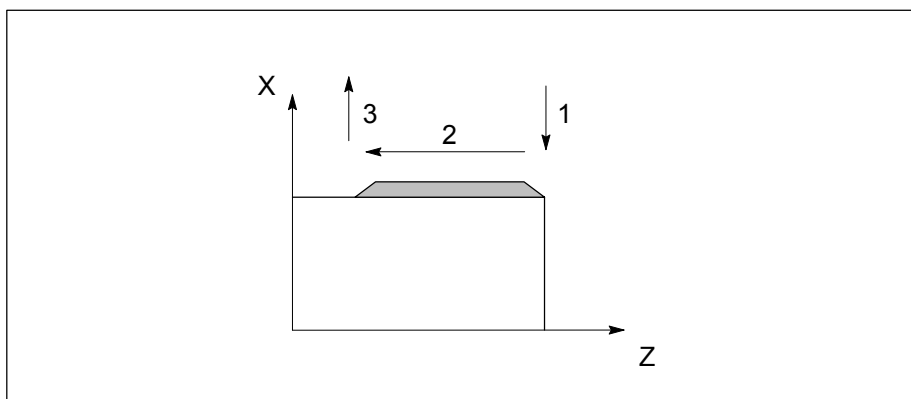
Přisuvový pohyb řezného nástroje (1).

G33 Z-50, K2

Řezání závitů (2). Koncový bod: inkrementální
o -50 mm ve směru Z.Pevné stoupání závitů: 2 mm/ot. Parametry in-
terpolace: zde K.

G0 X10

Volný pojezd řezného nástroje (3).



Příklad: Spirálový závit

G91 G18 G8 M3 S1000

Aktivovat programování relativních rozměrů.
Aktivovat rovinu Z/X.

G0 Z-10

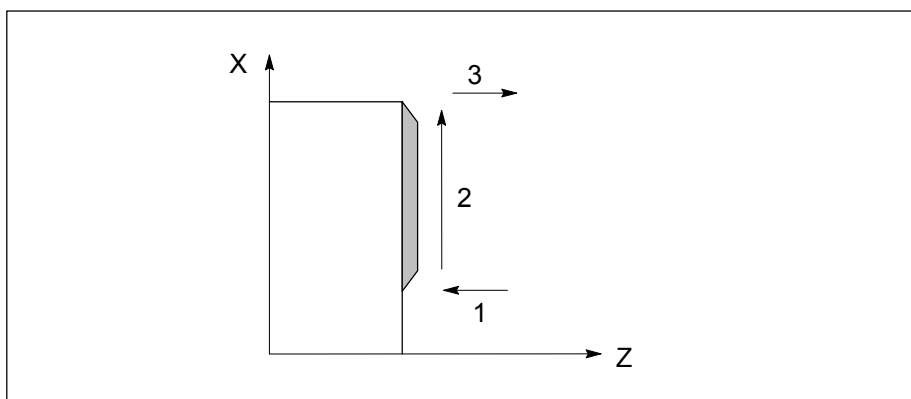
Přisuvový pohyb řezného nástroje (1).

G33 X40, I2

Řezání závitů (2). Koncový bod: inkrementální
o +40 mm ve směru X.Pevné stoupání závitů: 2 mm/ot. Parametry in-
terpolace: zde I.

G0 Z10

Volný pojezd řezného nástroje (3).



G-kódy

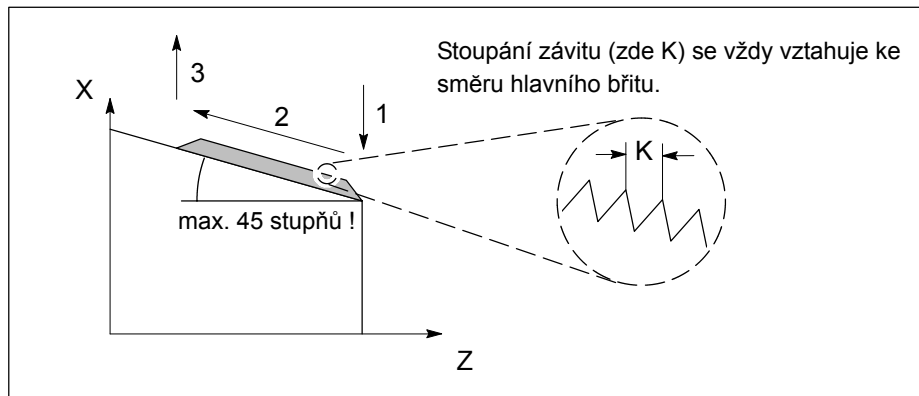
G33

Příklad: Kónický podélný závit

G91 G18 G8 M3 S1000 Aktivovat programování relativních rozměrů.
Aktivovat rovinu Z/X.

G0 X-20 Přisuvový pohyb řezného nástroje (1).
G33 Z-50 X15, K2 Řezání závitů (2). Koncový bod: inkrementální o -50 mm ve směru Z a +15 mm ve směru X. Pevné stoupání závitu: 2 mm/ot. Parametry interpolace: zde I.

G0 X5 Volný pojezd řezného nástroje (3).

**Zřetězený závit**

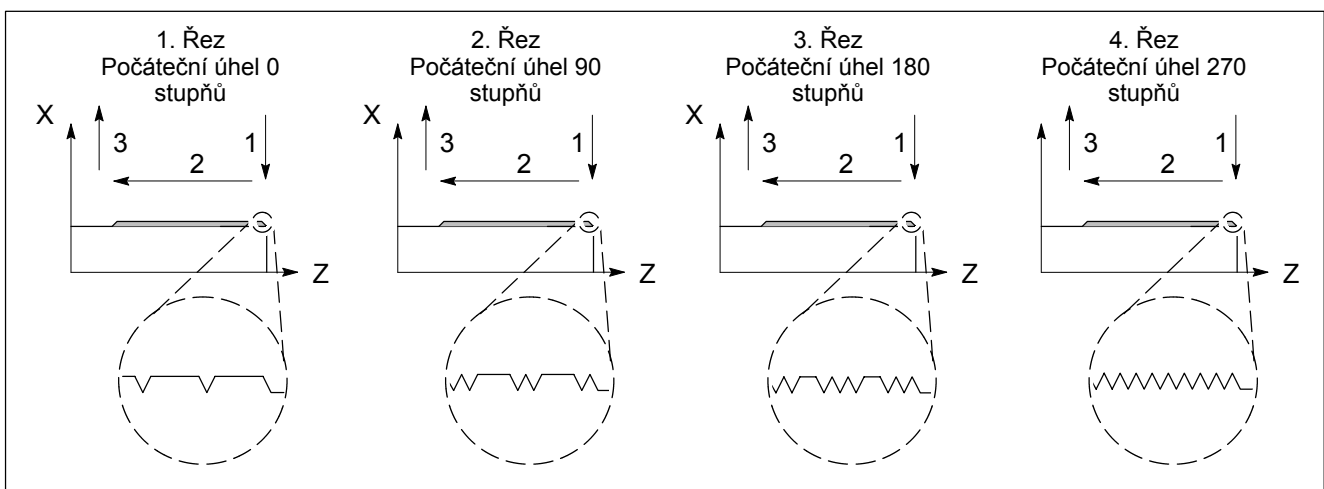
- Dají se vyrobit ze všech výše uvedených typů závitů.
- Programují se s použitím několika vět G33 následujících za sebou. NC kontroluje u každé naprogramované věty G33, jestli existuje následující věta G33 s dráhou. Pokud ano, proběhne přechod na další větu bez zastavení os.

Vícechodý závit

Vícechodý závit vyrobíte posunutím počátečního úhlu (počáteční úhel viz strana 3-34).

Příklad:

Čtyřchodý závit se vyrábí pomocí čtyř řezů posunutých vždy o 90 stupňů (0, 90, 180, 270).



G-kódy G33

Dynamické chování

Na začátku a na konci obrábění závitu musí být zúčastněné osy zrychleny, resp. zastaveny.

- ★ Naplánujte si proto vždy dostatečně velkou náběhovou dráhu (pro zrychlení řezných os) a doběhovou dráhu (pro zastavení).

V zásadě rozlišujeme mezi 2 možnostmi obrábění:

- "tvrdé" spuštění a "tvrdý" konec řezného pohybu:
Na začátku pohybu G33 skočí osy při dosažení počátečního úhlu na řeznou rychlost (počet otáček vřetena * pevné stoupání). Na konci pohybu G33 skočí rychlost na 0.
- Spuštění/konec řezného pohybu s individuálně nastavitelnou dynamikou:
Protože "tvrdé" řešení není vždy žádoucí nebo kvůli existujícím omezením v oblasti osové dynamiky není proveditelné, můžete individuálně nastavit dynamické chování s ohledem na skok rychlosti, a zrychlení, resp. brzdění:
 - staticky podle parametrů stroje (7050 00610, 7050 00615 a 7050 00620)
 - dynamicky podle dílčího programu s "TreadSet (DYN)" (viz strana 4-70).

Řízení vypočítává z naprogramovaného počátečního úhlu v závislosti na strmosti lineární změny zrychlení posunutý počáteční úhel. Tím je zaručeno, že bude nezávisle na velikosti zrychlení vždy řezán stejný chod závitu.

Na konci závitu se řezné osy odpojí od vřetena, a zabrzdí v závislosti na nastaveném brzděném zrychlení nejprve na rychlost skoku a potom **za normálních okolností až do zastavení**.

Pokud ovšem při aktivní funkci G8 následuje po větě G33 přímo další pojezdová věta, spustí se tento pohyb s rychlostí, která by byla nastavena, kdyby větou řezání závitu byla věta G1.

Rychlý zpětný pohyb

Ve spojení s G33 je užitečný "rychlý zpětný pohyb".

Pokud jsou data zpětného pohybu

- zkonfigurována
 - (- staticky podle parametrů stroje (7050 00645, 7050 00650) nebo
 - dynamicky v programu s "TreadSet (RD...)" (strana 4-70))


- aktivována
 - (- staticky podle parametrů stroje (7050 00640) nebo
 - dynamicky v programu s "TreadSet (RON1)" (strana 4-70)),

spouští pozitivní impuls v signálu rozhraní kanálu "rychlý zpětný pohyb" zpětný pohyb s následujícím průběhem:

1. Řezný pohyb se změní na pohyb orientovaný kolmo ke směru hlavního břítu.
2. Je-li uraženo více než 70% zpětného pohybu, odpojí se řezné osy od vřetena a zastaví s nastaveným brzděným zrychlením (parametr stroje 7050 00620).

G-kódy

G33

 **Pokud byl spuštěn zpětný pohyb, dá se tento stav opustit pouze "základním nastavením" nebo "odjetím od kontury".**

Zpětné pohyby probíhají vždy kolmo ke směru hlavního břitu ve směru vedlejší řezné osy.

Zpětné pohyby jsou spuštěny z NC automaticky, když dojde k událostem "základní nastavení kanálu", "systému" a "vřetena"!

Příklad: Zpětný pohyb z podélného závitu

G18	Aktivovat rovinu Z/X (G18).
TST(RO1)	Aktivovat rychlý zpětný pohyb (RON1).
TST(RD(0, 5))	Zpětný pohyb (RD...) o +5 mm ve směru vedlejšího břitu (zde X).
:	
G91 G33 Z- 20, K1	Zapnout inkrementální programování (G91). Řezání závitů (G33). Koncový bod: inkrementální o -20 mm ve směru Z. Pevné stoupání závitu: 1 mm/ot. Parametry interpolace: zde K.

G-kódy G40, G41, G42

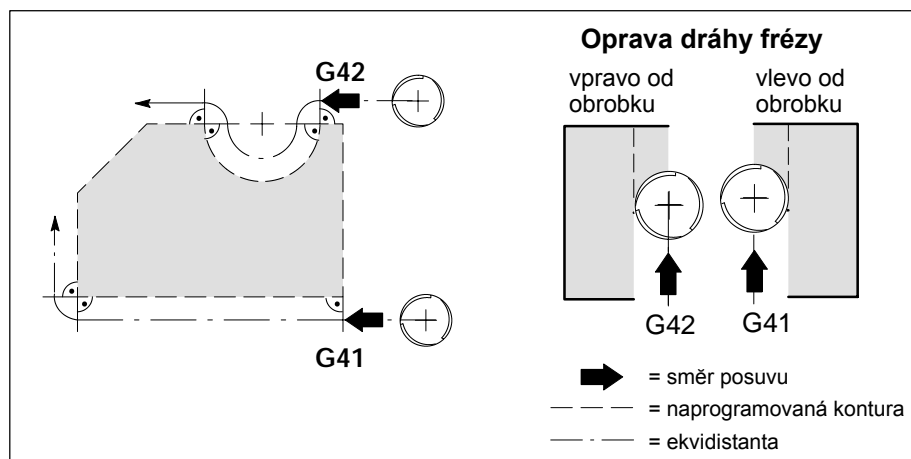
3.2.17 Oprava dráhy frézy

G40, G41, G42

Následek

Oprava dráhy frézy vede nástroj při zpracování programu obrobku po ekvidistantě paralelní s naprogramovanou dráhou (ekvidistanta = dráha s konstantní kolmou vzdáleností od naprogramované kontury). Vzdálenost ekvidistanty od naprogramované dráhy závisí na aktivní hodnotě opravy poloměru.

Následující obrázek znázorňuje princip:



Programování

Syntaxe:

- G40** Vypnutí opravy dráhy frézy (zapnutý stav).
Pokud ve větě G40 není naprogramován žádný pojezdový pohyb, vytváří řízení opravu okamžitě a kolmo k předchozí pojezdové větě.
Je-li ve větě G40 naprogramován pojezdový pohyb, vytváří řízení opravu lineárně na dráze do koncového bodu pojezdového pohybu.
Účinná oprava délky nástroje není funkcí G40 ovlivněna.
- G41** Zapnutí opravy dráhy frézy **vlevo od obrobku** (při kladných hodnotách opravy z pohledu ve směru obrábění).
Ve stejné větě jako G41 smí být kromě D-adresy s požadovanou hodnotou opravy poloměru navíc naprogramován lineární pojezdový pohyb pro osy aktivní roviny. Tak najede oprava na dráhu do koncového bodu pojezdového pohybu.
Pokud ve větě G41 není naprogramován žádný pojezdový pohyb, vytváří řízení opravu okamžitě a kolmo k další pojezdové větě.
- G42** Zapnutí opravy dráhy frézy **vpravo od obrobku**
Jinak jako G41.

G-kódy

G40, G41, G42

**POZOR**

Hodnoty oprav jsou za určitých okolností okamžitě najížděny nebo odjížděny bez naprogramování zvláštního pojezdového pohybu. To může vést k poškození obrobku nebo nástroje. Věnujte proto pozornost všem informacím v této kapitole!

Zvláštnosti a omezení:

- Funkce G40, G41 a G42 působí modálně a vzájemně se ruší.
- Funkce G40, G41 nebo G42 smí být při aktivní funkci G2, G3 nebo G5 naprogramovány pouze bez pojezdového pohybu.
- Při aktivní funkci G41 nebo G42 **nejsou dovoleny**:
 - G17 ... G20 (přepínání rovin)
 - G70, G71 (přepínání Inch/metrické)
 - G63 (řezání vnitřních závitů bez vyrovnávacího sklíčidla)
 - G74 (njetí do souřadnic referenčního bodu)
 - G75 (vstup měřicího dotyku)
 - G76 (njetí do pevné osové polohy stroje)
 - G54.x ... G59.x (posunutí nulových bodů)
 - G154.x ... G159.x (šikmá rovina)

G-kódy G43 G44

3.2.18 Konturové přechody pro opravu dráhy frézy: Kruhový oblouk Průsečík

G43
G44

Následek

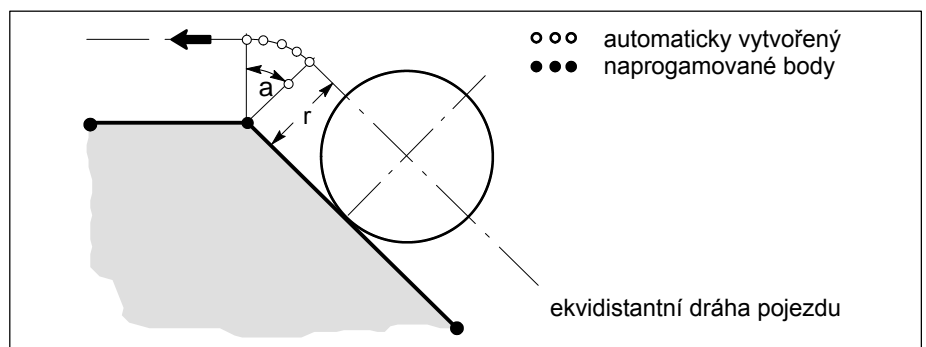
Funkce pro aktivní opravu dráhy frézy (G41, G42).

Řízení realizuje konturový přechod **na vnějších okrajích** podle potřeby jako automaticky vytvořený

- kruhový oblouk (G43) nebo jako
- průsečík ekvidistant (G44).

G43: Kruhový oblouk

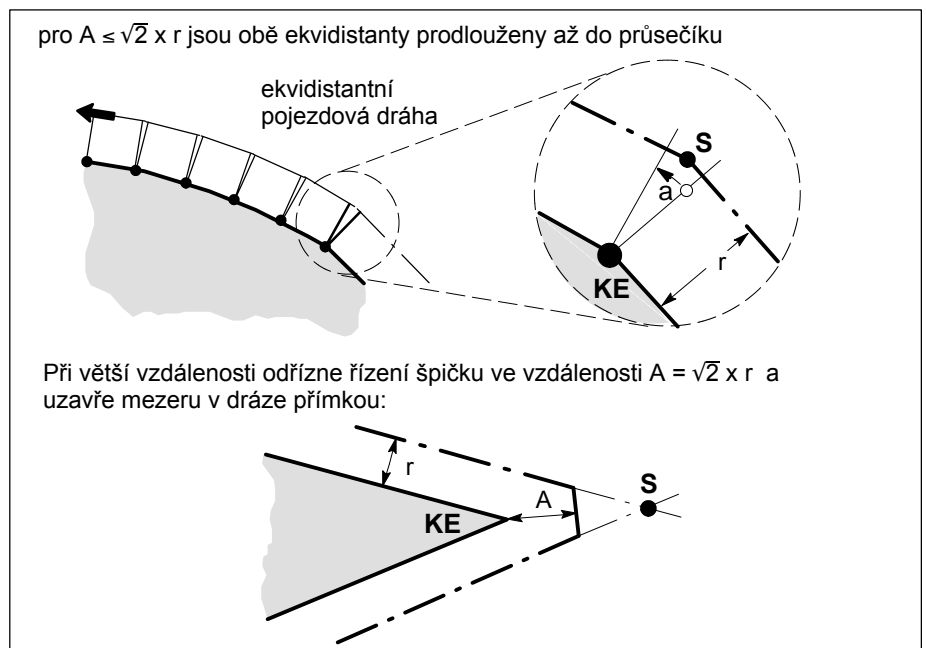
Mezera v dráze se uzavře tangenciálním kruhovým obloukem o poloměru "r":



G44: Průsečík

Řízení se pokouší uzavřít mezeru v dráze určením průsečíku obou ekvidistant.

V závislosti na vzdálenosti "A" mezi okrajem kontury "KE" a průsečíkem "S" postupuje řízení následujícím způsobem:



G-kódy G43 G44

 **Jesliže neexistuje žádný průsečík, je linie uzavřena jako u G43 pomocí kruhového oblouku.**

Programování

Syntaxe:

G43 Konturový přechod jako kruhový oblouk

G44 Konturový přechod jako průsečík ekvidistant.

Platí:

- Funkce G43 a G44 působí modálně a vzájemně se ruší.
- Zapnutý stav se dá určit pomocí parametru stroje.

Zvláštnosti a omezení:

- Funkce G43, resp. G44 se programují bez přípravných funkcí.

G-kódy G45 G46

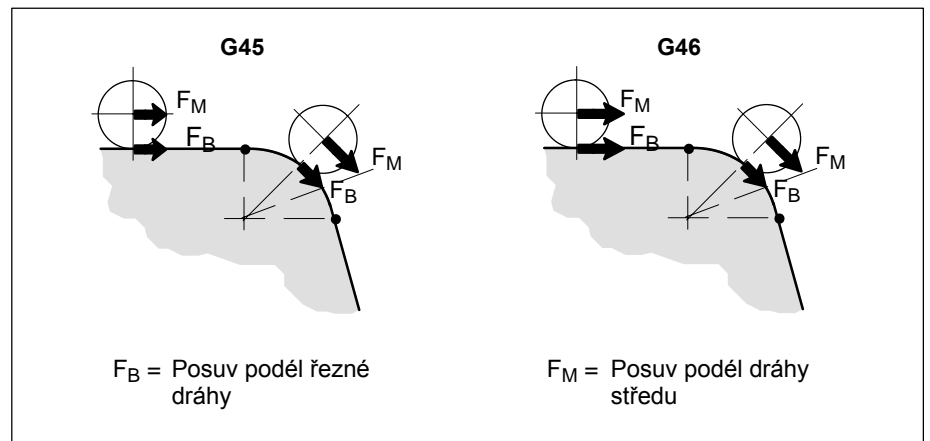
3.2.19 Oprava posuvu: Bod záběru frézy Střed frézy

G45
G46

Následek

Funkce pro aktivní opravu dráhy frézy (G41, G42).
Určují, jestli má řízení při kruhové interpolaci udržovat naprogramovaný posuv

- v bodě záběru frézy (řezná dráha frézy) nebo
 - na dráze středu frézy
- konstantní.



Programování

Syntaxe:

G45 Udržovat posuv F_B podél řezné dráhy konstantní.

G46 Udržovat posuv F_M podél dráhy středu frézy konstantní.

Platí:

- Funkce G45 a G46 působí modálně a vzájemně se ruší.
- Zapnutý stav se dá určit pomocí parametru stroje.

Zvláštnosti a omezení:

- Používejte G45 jen při frézování načisto, protože se může silně zvýšit rychlost posuvu na kruhových konturách.

G-kódy G47, G48

3.2.20 Oprava délky nástroje

G47, G48

Následek

Funkce

- aktivuje/deaktivuje opravu délky nástroje
- přepíná podle potřeby přiřazení hodnot opravy délky L_1 , L_2 a L_3 k jednotlivým souřadnicím (působí jako G78; viz strana 3-60).

Při aktivované opravě délky nástroje působí

- hodnoty opravy délky L_1 , L_2 a L_3 aktuálně vybrané sady D-korektur (Dxx)
- aditivně k tomu hodnoty opravy délky L_1 , L_2 a L_3 externí opravy nástroje (EDxx)
- funkce "statická orientace nástroje" (parametrizovatelná pomocí STO; viz strana 4-60)
- funkce "oprava polohy břitů", pokud je v aktuálně vybrané sadě D-korektur (Dxx) zadána poloha břitů a je aktivní oprava dráhy frézy G41/G42 (viz strana 3-39).

Programování

Syntaxe:

G47	Zapnutí opravy délky nástroje.
G47(ActPlane)	Přiřazení hodnot opravy délky L_1 , L_2 a L_3 hlavní, vedlejší a normálové souřadnici aktivní roviny a pak zapnutí opravy délky nástroje. Funkce zůstává modálně účinná a při každém dalším přepnutí roviny se automaticky znovu zkonfiguruje.
G47({{-}<L1-Koord> {,{{-}<L2-Koord> {,{{-}<L3-Koord>}}})	Změna přiřazení hodnot oprav délky L_1 , L_2 a L_3 k jednotlivým souřadnicím a pak zapnutí opravy délky nástroje.
G47()	Přiřazení hodnot opravy délky L_1 , L_2 a L_3 podle nastavení v parametrech stroje 7050 01300 a 7050 01310 a pak zapnutí opravy délky nástroje.
G48	Vypnutí opravy délky nástroje.
kde	
<Li-Koord>	Název souřadnice WCS, které má být přiřazena oprava L_i ($s = 1, 2, 3$) v aktivní soustavě souřadnic obrobku. Má-li oprava L_i ($s = 1, 2, 3$) působit na souřadnice v nástrojové soustavě souřadnic (TCS), musí se jako identifikátory souřadnic používat XTR, YTR a ZTR a musí být aktivní odpovídající transformace os. Volitelné záporné znaménko: Oprava bude vy počítána v záporném směru. L_i -Hodnoty oprav, jimž není přiřazena žádná

G-kódy

G47, G48

souřadnice, zůstávají beze změny.

Zvláštnosti a omezení:

- Souřadnice, které nepatří ke standardnímu nastavení kanálu, jsou při syntaxi "G47()" vždy ignorovány.
- Funkce G47.. a G48 jsou modálně účinné a vzájemně se ruší.
- Jestli se souřadnicové adresy naprogramované v G47 vztahují k souřadnicím obrobku nebo nástroje, závisí na použitých identifikátorech souřadnic (viz <Souřadnice *i*> výše).
- Funkce G47/G48 mohou být naprogramovány společně s jinými přípravnými funkcemi, pojezdovými informacemi nebo pomocnými funkcemi.

Příklad:

G47(X, , ZTR)

Oprava L_1 je přiřazena souřadnici X soustavy souřadnic obrobku (WCS), oprava L_3 souřadnici Z nástrojové soustavy souřadnic (TCS). Obě opravy se vypočítávají v kladném směru. Potom se zapne oprava délky nástroje. Oprava L_2 se nevypočítává, protože není přiřazena žádné souřadnici.

G-kódy G53 G53.1-G59.1 G53.5-G59.5

3.2.21 Posunutí nulových bodů (NPV)

**G53,
G53.1-G59.1
až G53.5-G59.5**

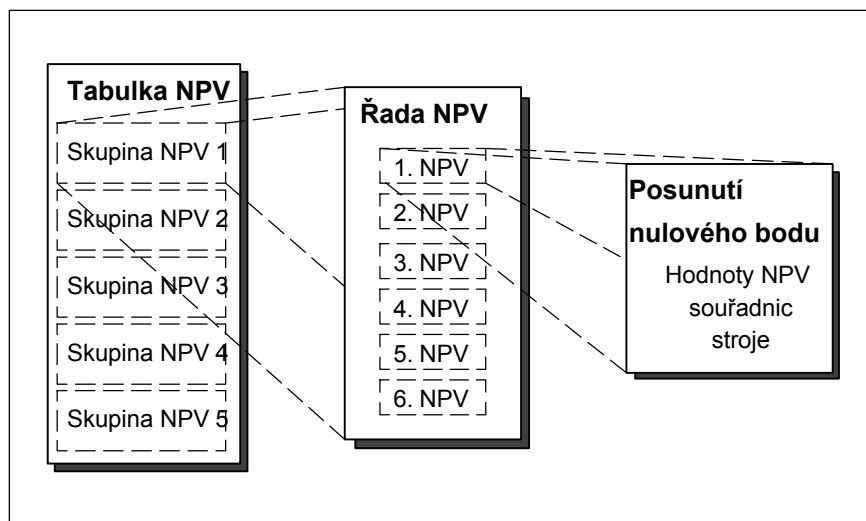
Následek

Pomocí NPV lze **posouvat** soustavu souřadnic stroje v prostoru.

☞ **Pokud je zapotřebí oprava upnutí, viz funkce "BcsCorr" (transformace obrobku: oprava polohy obrobku, viz strana 4-23).**

V "tabulkách posunutí nulových bodů" jsou uloženy vzdálenosti posunutí pro souřadnice stroje v kanálu:

- Tabulka NPV obsahuje 5 řad NPV (skupin) po 6 posunutích nulových bodů (NPV).
- Posunutí z různých řad NPV účinkují vždy aditivně.
- Posunutí v rámci jedné řady NPV se vzájemně přepisují.



☞ **Pokyny pro úpravy tabulek NPV naleznete v návodu k obsluze.**

G-kódy G53 G53.1–G59.1 G53.5–G59.5

Programování

1. Aktivujte požadovanou tabulku NPV pomocí "ZoTSEL" (viz strana 4-79).
2. Naprogramujte požadovanou funkci:

Syntaxe:

G53	Vypnutí všech NPV pro všechny řady oprav
G53.<Řada NPV>	Vypnutí všech NPV <řady NPV>
G54.<Řada NPV>	1. Zapnutí NPV <řady NPV>
G55.<Řada NPV>	2. Zapnutí NPV <řady NPV>
G56.<Řada NPV>	3. Zapnutí NPV <řady NPV>
G57.<Řada NPV>	4. Zapnutí NPV <řady NPV>
G58.<Řada NPV>	5. Zapnutí NPV <řady NPV>
G59.<Řada NPV>	6. Zapnutí NPV <řady NPV>
Zkratka: G54 až G59	Pro příslušné NPV řady 1

kde

<Řada NPV> 1...5, Integer

Platí:

- G53 působí modálně a ruší každé aktivní NPV.
G53.1 až G59.1 působí modálně a vzájemně se ruší.
G53.2 až G59.2 působí modálně a vzájemně se ruší.
G53.3 až G59.3 působí modálně a vzájemně se ruší.
G53.4 až G59.4 působí modálně a vzájemně se ruší.
G53.5 až G59.5 působí modálně a vzájemně se ruší.
- Samostatně naprogramované nezpůsobují tyto funkce žádný pojezdový pohyb. Pouze se posunuje aktuální soustava souřadnic stroje.
- Funkce mohou být naprogramovány také společně s přípravnými funkcemi ve stejné větě. V takových případech se nejprve aktivuje odpovídající posunutí nulového bodu a potom se najíždí do naprogramované polohy.

G-kódy

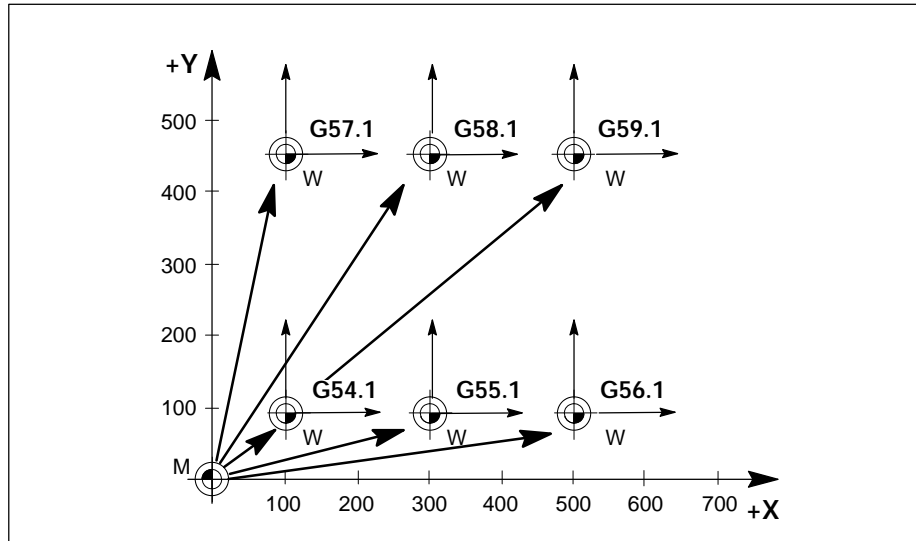
G53 G53.1-G59.1 G53.5-G59.5

Příklad:

V tabulce NPV "Z01" jsou obsaženy následující hodnoty posunutí v řadě NPV 1:

- 1.NPV: X100, Y100
- 2.NPV: X300, Y100
- 3.NPV: X500, Y100
- 4.NPV: X100, Y450
- 5.NPV: X300, Y450
- 6.NPV: X500, Y450

Účinky funkcí G54.1 až G59.1:

**Příklad:**

Aditivní působení NPV z různých řad NPV.

V tabulce NPV "Z01" jsou obsaženy následující hodnoty posunutí:

- Řada NPV 1, 1. NPV: X100, Y100
- Řada NPV 1, 2. NPV: X300, Y100
- Řada NPV 2, 4. NPV: X200, Y350

:	Aktuální poloha stroje 0,0. Žádné NPV.
:	Všechny následující souřadnicové údaje jsou souřadnice stroje!
:	Aktivovat tabulku NPV "Z01".
N40 Z0S(Z01)	Žádný pojezdový pohyb.
N50 G54. 1	Aktivováno posunutí na X100, Y100.
N60 G55. 1 X. . . Y. . .	Aktivováno posunutí na X300, Y100.
N70 G57. 2 X. . . Y. . .	Následně pojezdový pohyb.
N80 G53	Aktivováno posunutí na X500, Y450.
	Následně pojezdový pohyb.
	Vypnutí všech ještě aktivních posunutí.

G-kódy G61, G62

3.2.22 Zapnutí/vypnutí přesného zastavení

G61, G62

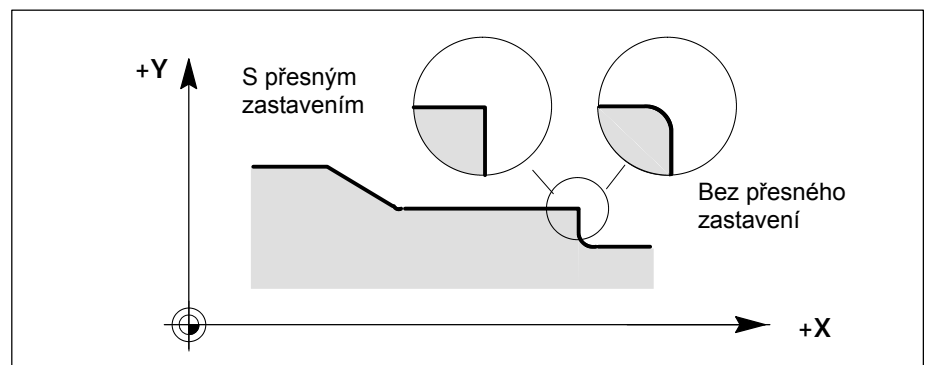
Následek

Minimalizuje vlečnou chybu na konci věty.

Při pohybech nástroje dochází v důsledku dynamiky stroje k časovému posunutí mezi požadovanými a skutečnými hodnotami jednotlivých os. V důsledku tohoto "doběhového efektu" vzniká při třískovém obrábění vlečná chyba, jejíž velikost závisí na rychlosti posuvu a faktoru KV (osová dynamika). Při nespojitých konturových přechodech (rozích) se tato vlečná chyba obzvlášť zvýrazňuje.

Přitom platí:

- Vlečná vzdálenost (posunutí mezi požadovanou a skutečnou hodnotou) na konci věty se dá eliminovat na velikost určených cílových oken.
- G61 působí pouze na pohyby v posuvu. Nepůsobí při pohybech v rychlém chodu.
- Na rozdíl od funkce přesného zastavení u G1 působí G61 s výjimkou G0 na všechny interpolační funkce (také při kruhové/spirálové interpolaci a spline interpolaci).
- G61 a G62 působí modálně a vzájemně se ruší.



☞ Při aktivní funkci G61 (zapnuté přesné zastavení) brzdí řízení na konci věty vždy na $v = 0$.

Programování

Základní funkce Syntaxe:

G61 Zapnutí přesného zastavení.

G62 Vypnutí přesného zastavení.

Při zapnutí přesného zastavení platí nadále poslední aktivované polohovací okno pro posuvový provoz. Standardní: Polohovací okno jemné.

G-kódy G61, G62


Volitelné parametry Pro výběr polohovacího okna pro posuvový provoz.

Syntaxe:

G61(IPS1 | IPS2 | IPS3)

kde

IPS1	Zapnutí přesného zastavení. Čekání na jemné polohovací okno . Na konci věty zabrzdí řízení nejprve rychlost posuvu na $v=0$. Teprve když je toto polohovací okno dosaženo všemi zúčastněnými osami, zpracovává se další věta.
IPS2	Zapnutí přesného zastavení. Čekání na hrubé polohovací okno . Na rozhraní kanálu se objeví "Oblast Inpos 2 aktivní" (viz též příručka "Rozhraní SPS"). Na konci věty zabrzdí řízení nejprve rychlost posuvu na $v=0$. Teprve když je toto polohovací okno dosaženo všemi zúčastněnými osami, zpracovává se další věta.
IPS3	Zapnutí přesného zastavení. Zabrzdění na $v=0$ na konci věty. Na konci věty zabrzdí řízení rychlost posuvu na $v=0$. Potom proběhne najetí na další větu bez kontroly polohovacího okna.

 **Parametry "Polohovací okno jemné" a "Polohovací okno hrubé" se dají určit v souborech SERCOS pro fázi 3. Podrobnosti o souborech SERCOS naleznete v příručce "Parametry stroje" v části "Inicializace SERCOS".**

Zvláštnosti a omezení:

- Funkce G61 a G62 musí být naprogramovány nejpozději ve větě, ve které mají působit.

Příklad:

:	
N40 G61	Zapnutí přesného zastavení.
:	
N50 Y200	Polohovat.
:	
N60 G62	Vypnutí přesného zastavení.
:	
N70 Y0	Polohovat.
:	
N80 G61 Y200	Polohovat se zapnutým přesným zastavením.
:	

G-kódy G63

3.2.23 Řezání vnitřních závitů bez vyrovnávacího sklíčidla G63

Následek

- Synchronizuje lineární interpolaci osy vrtání s vřetenem přepnutým na provoz osy C. Díky tomu je vyrovnávací sklíčidlo, které jinak musí kompenzovat rozdíl v počtu otáček mezi osou vrtání a vřetenem, zbytečné.
- G63 působí pouze v naprogramované větě.
- Po dobu řezání vnitřních závitů může být na rozhraní kanálu vydáván signál (konfigurovatelný v rámci signálů "Aktivní funkce" pomocí parametrů stroje).
- Během řezání vnitřních závitů funguje pouze potenciometr posuvu.

 **Pro řezání vnitřních závitů s více vřeteny viz též kapitola 4.89. Pokud je zapotřebí skrýt osy pro vytvoření posuvu, viz kap. 4.28.**

Programování

Syntaxe:

G63(M<3|4>, S<Počet otáček>|H<Stoupání závitu>) <Osa vrtání><Hloubka přísluvu> {F<Posuv>},

Stoupání závitu vyplývá z poměru posuvu k počtu otáček (F/S).



POZOR

Různá stoupání závitů při zavrtávání a při zpětném pohybu poškozují obrobek a nástroj!

Proto naprogramujte zavrtávání a zpětný pohyb vždy tak, aby výsledné stoupání závitu bylo v obou případech stejné!

Zvláštnosti a omezení:

- Věty vrtání a zpětného pohybu musí být naprogramovány hned za sebou.
- Při vrtání hlubokých děr je třeba naprogramovat mezi jednotlivými dílčími otvory prázdný řádek.
- Pokud ve větě G63 není zadána žádná jiná hodnota posuvu (adresa F), používá řízení aktivní posuv dráhy.
- Adresy M a S působí pouze v naprogramované větě G63.
- Před G63 není nezbytné ani "seřízení vřetena" ani "zastavení vřetena".
- Přepínání na provoz osy C se provádí automaticky. Před spuštěním přitom čeká řízení na interní hlášení "INPOS" ze všech zúčastněných os. Jestliže se některá osa vychýlí z oblasti INPOS, nespustí se G63.
- Po větě zpětného pohybu přejde vřeteno automaticky opět do vřetenového provozu.

G-kódy

G63

Příklad:

N20 G0 X20 Y15 Z10 F1000
N30 G63(M3, S500) Z- 20 F500
N40 G63(M4, S500) Z5 F500

:

N120 G0 X20 Y15 Z10 F1000

N130 G63(M3, S500) Z- 20 F500

N140 G63(M4, S500) Z5 F500

<Prázdný řádek>

N150 G63(M3, S500) Z- 40 F500

N160 G63(M4, S500) Z5 F500

:

Polohovat.

Zavrtávat (osa vrtání Z).

Zpětný pohyb (osa vrtání Z).

Polohovat pro vrtání hlubokých
děr.

Zavrtávat úsek součástky 1.

Zpětný pohyb z úseku součástky
1.

Naprogramujte prázdný řádek.

Zavrtávat úsek součástky 2.

Zpětný pohyb z úseku součástky
2.

G-kódy G70

3.2.24 Coulové programování

G70

Následek

Umožňuje zadávat pojezdové a posuvové informace a zrychlení v **coulových mírách**.

Programování

Syntaxe:

G70 Pojezdové a posuvové informace a zrychlení jsou interpretovány v coulech.

Platí:

- G70 působí modálně a ruší G71.
- Zapnutý stav se dá určit pomocí parametru stroje.
- Funkce G70 může být naprogramována také společně s dalšími přípravnými funkcemi ve stejné větě.
- G70 se vztahuje k
 - drahám pojezdu
 - posuvům
 - zrychlením
 - jiným geometrickým pomocným veličinám, jako jsou např. interpolační parametry I, J, K.

Příklad:

N40 G70 : Od N40 včetně jsou všechny pojezdové a posuvové informace interpretovány v coulech.

G-kódy G71

3.2.25 Metrické programování

G71

Následek

Umožňuje zadávat pojezdové a posuvové informace a zrychlení v **metrických mírách**.

Programování

Syntaxe:

G71 Pojezdové a posuvové informace a zrychlení jsou interpretovány metricky.

Platí:

- G71 působí modálně a ruší G70.
- Zapnutý stav se dá určit pomocí parametru stroje.
- Funkce G71 může být naprogramována také společně s dalšími přípravnými funkcemi ve stejné větě.
- G71 se vztahuje k
 - drahám pojezdu
 - posuvům
 - zrychlením
 - jiným geometrickým pomocným veličinám, jako jsou např. interpolační parametry I, J, K.

Příklad:

N40 G71 : Od N40 včetně jsou všechny pojezdové a posuvové informace interpretovány metricky.

G-kódy G74

3.2.26 Najetí na souřadnice referenčního bodu

G74

Následek

Osy naprogramované ve stejné větě s G74 pojíždějí současně do svých referenčních poloh.

Rychlost pojezdu přitom závisí na G0/G1 v rychlém chodu / posuvu.

U funkce G74 nejsou brány v úvahu zarážky ani značky referenčních bodů. G74 je **čistá polohovací procedura na absolutní osové polo-
lohy**, platí tedy také pro osy se snímači s kódováním vzdálenosti.



POZOR

Případně aktivní opravy zůstávají při této polohovací proceduře bez povšimnutí!

Platí:

- G74 působí po větách a ruší se, když všechny osy stroje naprogramované ve větě G74 dosáhnou svých referenčních bodů.
- Při najetí do referenčního bodu pomocí G74 se nenastavují nově skutečné hodnoty os.
- Případně ještě aktivní opravy, NPV atd. zůstávají ve větě G74 pro naprogramované osy ignorovány.

Programování

Syntaxe:

G74 <Souřadnice os>

Spuštění "njetí do referenčních bodů souřadnic".

kde

<Souřadnice os>:

Osové adresy musí být naprogramovány spolu s číselnou hodnotou (např. X1 Y1 Z1). Číselná hodnota přitom nemá žádný vliv na polohu referenčního bodu. Slouží pouze k doplnění.

Zvláštnosti a omezení:

- G74 se programuje ve zvláštní větě s pojíždějícími osami. Pomocné a přidavné funkce mohou být naprogramovány ve stejné větě.

Příklad:

N100 G74 X1 Y1 Z1
:

Osy X, Y a Z začínají současně pojezd do svých referenčních poloh a současně jich dosáhnou.

G-kódy G74(HOME)

3.2.27 Najetí do referenčního bodu

G74(HOME)

 **Funkce G74(HOME) je použitelná i pro asynchronní osy!**

Následek

Spouští v obráběcím programu referencování naprogramovaných os.

Funkce ruší pro pohony naprogramovaných os příslušný příkaz SERCOS "Referencování řízené pohonem" (S-0-0148).

Pohon se pak odpojí od řízení a sám si vytvoří polohové údaje pro referencování. Přitom používá parametry SERCOS S-0-0147 (parametr referenčního pojezdu), S-0-0041 (rychlost referenčního pojezdu) a S-0-0042 (zrychlení referenčního pojezdu).

 **Podrobný popis naleznete v příručce "Funkční popis".**

Platí:

- Je-li ve větě G74(HOME) naprogramováno několik os, najíždějí vzájemně nezávisle do svých referenčních bodů (ne v posuvovém provozu). Referenční body tedy nejsou dosaženy zároveň.
- Je možné programovat **synchronní a asynchronní** osy. Zpracování vět zůstává zastavené, dokud všechny pohony řízení nepotvrdí úspěšné dosažení svých referenčních bodů.
- Tato funkce se neliší od "njetí do referenčního bodu" v provozním režimu "Seřízení".

Programování

Syntaxe:

G74(HOME) <Souřadnice os>

Spustit příkaz SERCOS "Referencování řízené pohonem" pro naprogramované osy.

kde

<Souřadnice os>:

Osové adresy musí být naprogramovány spolu s číselnou hodnotou (např. X1 Y1 Z1). Číselná hodnota přitom nemá žádný vliv na polohu referenčního bodu. Slouží pouze k doplnění.

Příklad:

N1 G74(HOME) X1 Y1 Z1
:

Odeslat do pohonů os X, Y a Z příkaz SERCOS "Referencování řízené pohonem".

G-kódy G75

3.2.28 Měřicí dotyk


G75

Následek

Řízení vede jednu nebo několik měřených os v posuvu do polohy naprogramované funkcí G75 a kontroluje přitom, jestli se spustí měřicí dotyk. Jakmile je rozpoznán impuls definovaný pomocí parametru stroje, reaguje řízení takto:

- Uložení skutečné polohy
- Zabrzdění na $v=0$ s maximálním dovoleným zrychlením
- Vymazání G75 a zbývající dráhy
- Přepnutí na další větu

 **Používejte G75 jen v souvislosti s programem CPL pro vyhodnocení.**

 **Funkce měřicího dotyku se dá parametrizovat pomocí parametrů stroje.**

Programování

Syntaxe:

G75 <Souřadnice osy> Aktivovat měřicí vstup a najet s měřenou osou na <souřadnici osy>.

Zvláštnosti a omezení:

- G75 působí po větách.
- Ve větě G75 nesmí být naprogramovány žádné pomocné funkce. Jiné přípravné funkce jsou však povoleny.
- Funkce G75 musí být naprogramována společně nejméně s jednou <souřadnicí osy>. Její hodnota představuje maximální hloubku hledání, do které se musí dotyk nejpozději sepnout.
- Zastavení přípravy vět po větě G75 není nutné (není zapotřebí funkce WAIT).
- Vyhodnocení osových informací, monitorování bezpečnosti, generování chybových hlášení atd. musí být realizováno programem CPL.

Příklad:

N100 G75 Y250 F500	Najet s F500 do polohy Y250.
110 IF SD(9)=0 THEN	Dotaz, jestli se vychýlil měřicí dotyk.
120 YPOS=PP0S(2)	Uložit polohu ve spínacím momentu 2. osy
:	(osa Y) do proměnné YPOS.
N130 (MSG, KONTAKT)	
140 GOTO N180	
150 ENDIF	
N160 (MSG, ŽÁDNÝ KON-	
TAKT)	
N170 MO	Zastavení programu
N180 . . .	

G-kódy G76

3.2.29 Najetí do pevné osové polohy stroje

G76

Následek

Umožňuje programování a lineární najíždění do poloh v soustavě souřadnic stroje, aniž by bylo nutné zvlášť deaktivovat případně aktivní opravy a transformace v obráběcím programu.

To může být nutné např. při výměně nástroje, při kontrolách zlomení nástrojů, měřicích cyklech nebo výměnách palet.

G76 působí po větách v rychlém chodu (G0) nebo v posuvu (G1) a také ve spojení s G93 (časové programování), G94/G95 (programování posuvu) a se slovem F.

Následující funkce jsou při G76 ignorovány:

- Opravy nástroje (G41, G42, G47 Dxx, EDxx)
- Pomůcky pro zadávání (PoleSet, Mirror, Scale, Rotate)
- Posunutí nulových bodů (G54.1...G59.5)
- Umístění (BcsCorr, G154.1...G159.5)
- Transformace os na bázi souřadnic stroje, např. 5-osová transformace
- Programování relativních rozměrů (G91)
- Nastavení programové polohy (SetPos)

Programování

Syntaxe:

G76 <Souřadnice stroje>

kde

<Souřadnice stroje>: Najížděné souřadnice stroje.

Zvláštnosti a omezení:

- Funkce G76 může být zapsána společně s dalšími přípravnými funkcemi (např. G0, G1, G93, G94, G95, F-slovo).
- Lokální programování relativních rozměrů ("IC(...)"; viz strana 3-69) není v souvislosti s G76 dovoleno a vede k běhové chybě.

G-kódy G77

3.2.30 Asynchronní podprogramy: Změna polohy jednotlivých souřadnic

G77

Následek

Pojíždí s definovanými souřadnicemi na bod opětovného najetí po skončení asynchronního podprogramu, ve kterém je funkce G77 naprogramována.

Do bodu opětovného najetí se vždy najíždí po přímce.

 **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

G77 <Souřadnice 1><Režim> <Souřadnice n><Režim> ... **F**<Hodnota>

kde

<Souřadnice i> Název souřadnice (např. "x")

<Režim> <Souřadnice i> při opětovném najetí

1 brát v úvahu.

0 nebrat v úvahu. Souřadnice zůstane na aktuální hodnotě.

<Hodnota> Posuv, se kterým má proběhnout polohování na bod opětovného najetí.

Zvláštnosti a omezení:

- Funkce je určena k používání v asynchronním podprogramu.
- Všechny souřadnice nenaprogramované funkcí G77 najedou na konci asynchronního podprogramu automaticky po přímce do bodu opětovného najetí.
- Mezitím provedené změny oprav v asynchronním podprogramu jsou při interním výpočtu požadovaného bodu opětovného najetí automaticky brány v úvahu.
- Jako alternativní syntaxi lze místo G77 naprogramovat také REPOS.

G-kódy G78 G79

3.2.31 Zapnutí oprav Vypnutí oprav

G78
G79

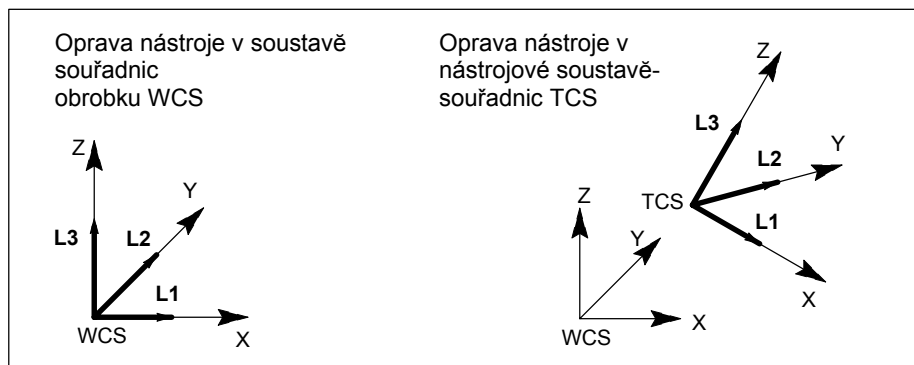
Následek

Přiřazuje **délkové opravy** geometrických korekčních funkcí jednotlivým souřadnicím. To jsou souřadnice:

- aktuální soustavy souřadnic obrobku (WCS) nebo
- nástrojové soustavy souřadnic (TCS).

Přiřazení souřadnic **soustavy souřadnic obrobku (WCS)** je možné vždy, když je nástroj vyrovnaný kolmo k aktuální pracovní rovině a jeho orientace vzhledem k pracovní rovině během obrábění zůstává konstantní.

Přiřazení souřadnic **nástrojové soustavy souřadnic (TCS)** je nutné, když se orientace nástroje během obrábění mění, např. při frézování volných ploch. Pro tento výpočet opravy je nezbytná aktivní transformace os (např. 5-osová nebo 6-osová). Výpočet hodnot opravy se provádí v transformaci os.



☞ **Pro aktivaci aktuálních hodnot oprav pro obrábění viz G47 od strany 3-44.**

Programování

Syntaxe:

G78({{-}<Souřadnice1>},{{-}<Souřadnice2>},{{-}< Souřadnice3>}) Zapnutí oprav.

G-kódy

G80 G79

kde

<Souřadnice i> Název souřadnice WCS (logický/fyzický osový název), které má být přiřazena oprava L_i (kde $i = 1, 2, 3$).

Pokud má oprava L_i působit na souřadnice v nástrojové soustavě souřadnic TCS, musí být jako identifikátory souřadnic použity XTR, YTR a ZTR.

Volitelné záporné znaménko: Oprava bude vypočítána v záporném směru.

L_i -Opravy, jimž není přiřazena žádná souřadnice, jsou ignorovány.

G78(ActPlane) Opravy L_1 , L_2 a L_3 jsou přiřazeny hlavní, vedlejší a normálové souřadnici příslušné aktivní roviny.

G79 Vypnutí oprav.
Platí opět nastavení v parametrech stroje.

Zvláštnosti a omezení:

- Funkce G78 a G79 jsou modálně účinné a vzájemně se ruší.
- Souřadnice, které nepatří ke standardnímu nastavení kanálu, jsou při funkci G79 vždy ignorovány!
- Souřadnicové adresy naprogramované v G78 se mohou vztahovat v souřadnicím obrobku nebo nástroje (viz výše <Souřadnice i>).
- Funkce G78/G79 mohou být naprogramovány společně s jinými přípravnými funkcemi, pojezdovými informacemi nebo pomocnými funkcemi.

Příklady:

G78(X, ,ZTR) Oprava L_1 je přiřazena ose X soustavy souřadnic obrobku (WCS), oprava L_3 ose Z nástrojové soustavy souřadnic (TCS). Obě opravy se vypočítávají v kladném směru.

G78(, , -Y) Oprava L_3 je přiřazena ose Y soustavy souřadnic obrobku (WCS) a vypočítána v záporném směru.

G78(YA,YB) Oprava L_1 je přiřazena ose YA a oprava L_2 ose YB soustavy souřadnic obrobku (WCS). Obě opravy se vypočítávají v kladném směru.

G79 Pro všechny standardní souřadnice kanálu jsou vypnuty opravy.

3.2.32 Vypnutí vrtacího cyklu

G80

Následek

Vypíná aktivní vrtací cyklus (G81 - G86, G184).

G-kódy G82 G81

Programování

Syntaxe:

G80

Zvláštnosti a omezení:

- Ve větě G80 nesmí být naprogramován žádný výraz se závorkami.

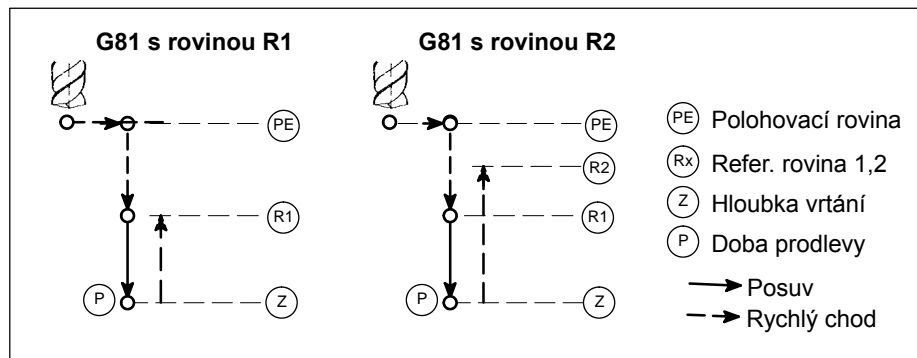
3.2.33 Vrtací cyklus: Vrtání se zpětným pohybem v rychlém chodu G81

Používání

- Středicí a jednoduché vrtací obrábění, čelní zahlubování, vyvrtávání.

Následek

Po dosažení hloubky vrtání Z může být podle potřeby účinná doba prodlevy. Potom následuje zpětný pohyb v rychlém chodu.

**Programování**

Syntaxe:

G81 [<Z>, <R1>, {<P>}, {<R2>}]

Zvláštnosti a omezení:

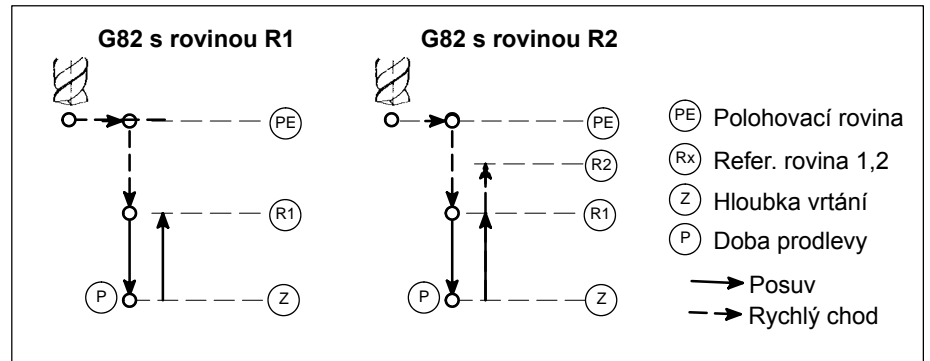
- Tento vrtací cyklus se provádí v každé nové pojezdové větě.
- Zrušení s G81.
- Před změnou cyklu je třeba zrušit aktuální cyklus pomocí G80.

3.2.34 Vrtací cyklus: Vrtání se zpětným pohybem v posuvu G82

Jako G81. Zpětný pohyb do R1 však probíhá v posuvu.

G-kódy

G82



Programování

Syntaxe:

G82[<Z>, <R1>, {<P>}, {<R2>}]

Zvláštnosti a omezení:

- Tento vrtací cyklus se provádí v každé nové pojezdové větě.
- Zrušení s G81.
- Před změnou cyklu je třeba zrušit aktuální cyklus pomocí G80.

G-kódy

G83

3.2.35 Vrtací cyklus: Vrtání hlubokých děr

G83

Používání

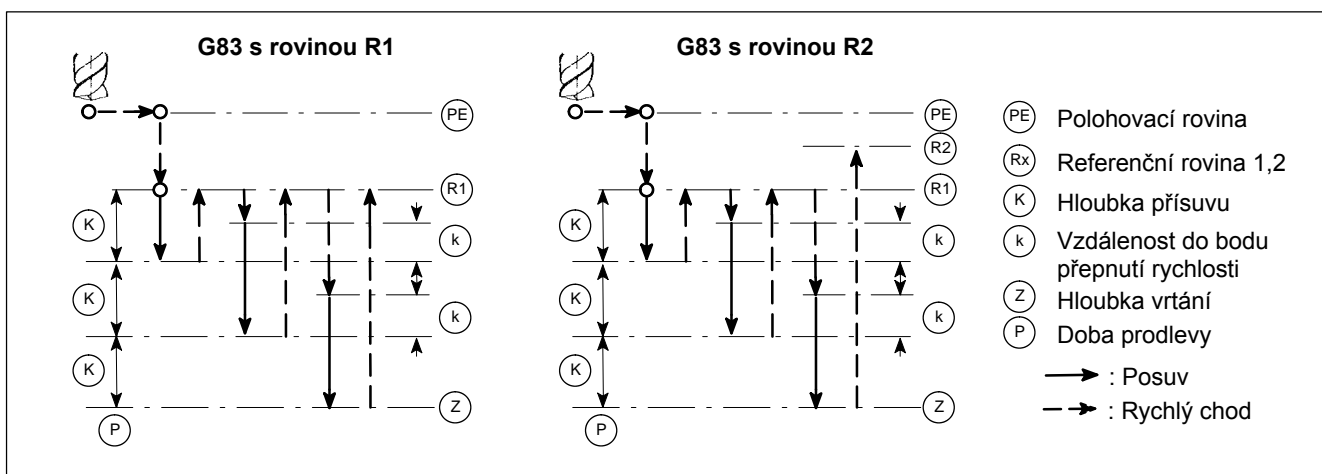
- Vrtání hlubokých děr s odstraňováním třísek.

Následek

Po dosažení naprogramované hloubky přísluvu K následuje zpětný pohyb do referenční roviny R1 v rychlém chodu.

Nový přísluv se až do naprogramované vzdálenosti k (bod přepnutí rychlosti) provádí rovněž v rychlém chodu. Potom se řízení opět přepne na rychlost posuvu.

Krokový přísluv s příslušným zpětným pohybem do referenční roviny se provádí tak dlouho, dokud není dosažena naprogramovaná celková hloubka vrtání Z.



Programování

Syntaxe:

G83[<Z>, <R1>, <K>, <k>, {<P>}, {<R2>}]

- ★ Naprogramujte hloubku přísluvu K nezávisle na směru vrtání v přírůstku souřadnic a bez znaménka.
 Když je maximální hloubka vrtání Z kvůli chybnému naprogramování hloubky přísluvu K překročena, přeruší řízení nejprve vrtací cyklus funkcí M0 a zobrazí chybové hlášení "BOHRTIEFE K ZU GROSS".
 Po opakovaném spuštění bude vrtací cyklus ukončen (M30).

Zvláštnosti a omezení:

- Tento vrtací cyklus se provádí v každé nové pojezdové větě.
- Zrušení s G81.
- Před změnou cyklu je třeba zrušit aktuální cyklus pomocí G80.

G-kódy G84

3.2.36 Vrtací cyklus: Řezání vnitřních závitů s vyrovnávacím sklíčidlem G84

Používání

- Řezání vnitřních závitů (levých a pravých) s vyrovnávacím sklíčidlem.

Předpoklad:

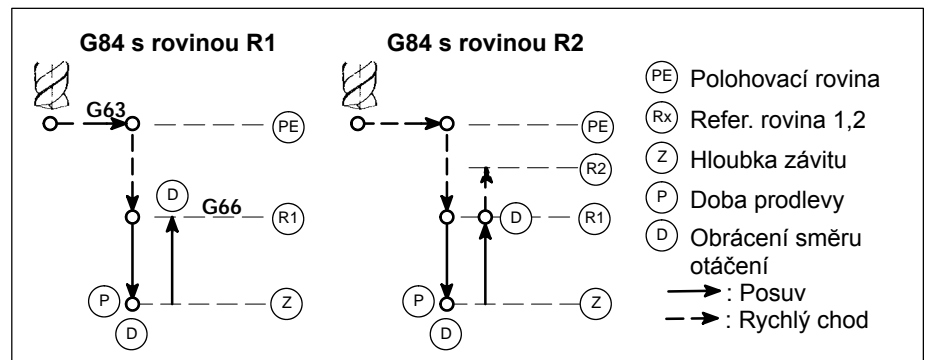
- **vnitřní** vřeteno jako osa vrtání. Vnější vřetena nejsou dovolena.

Následek

Přísuv nástroje probíhá s naprogramovaným pravotočivým M3 nebo levotočivým M4 chodem vřetena (pravý nebo levý závit).

Je-li dosažena hloubka vrtání Z (hloubka závitů), obrátí se směr otáčení a začne běžet doba prodlevy P (pokud je naprogramována).

Potom proběhne zpětný pohyb v posuvu do referenční roviny. Když je dosažena, zruší se opět obrácení směru otáčení vřetena.



Programování

Syntaxe:

G84[<Z>, <R1>, {<P>}, {<R2>}]



POZOR

Možnost poškození nástrojů nebo obrobků!

Během cyklu není potlačeno eventuálně aktivní zpracování jednotlivých vět!

Díky tomu běží vřeteno po polohovací proceduře v rámci cyklu dál. Tím se může poškodit obrobek a nástroj.

Dávejte proto pozor, aby řízení zpracovávalo cyklus jedině v provozním režimu "Následná věta"!

Zvláštnosti a omezení:

- Tento vrtací cyklus se provádí v každé nové pojzdové větě.
- Zrušení s G81.
- Před změnou cyklu je třeba zrušit aktuální cyklus pomocí G80.

G-kódy G85

3.2.37 Vrtací cyklus: Vyvrtávání se zpětným pohybem v rychlém chodu

G85

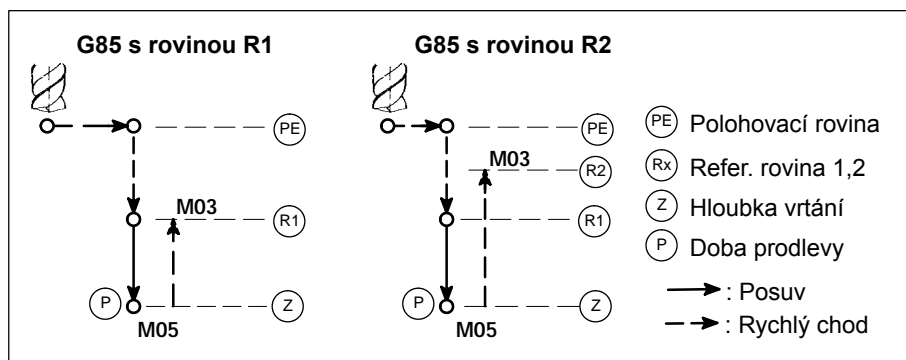
Používání

- Vyvrtávání

Následek

Po dosažení hloubky vrtání Z

- zastaví se vřeteno a
- začne běžet doba prodlevy (je-li naprogramována).
- Potom následuje zpětný pohyb v rychlém chodu.



Programování

Syntaxe:

G85 [<Z>, <R1>, {<P>}, {<R2>}]

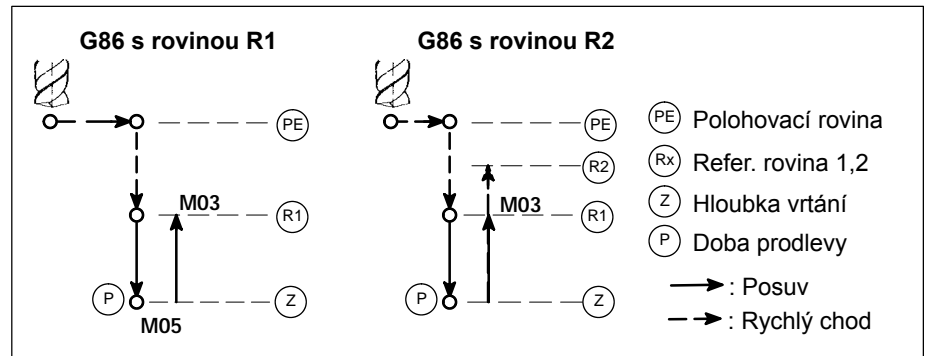
Zvláštnosti a omezení:

- Tento vrtací cyklus se provádí v každé nové pojezdové větě.
- Zrušení s G81.
- Před změnou cyklu je třeba zrušit aktuální cyklus pomocí G80.

G-kódy G86

3.2.38 Vrtací cyklus: Vyvrátání se zpětným pohybem v posuvu G86

Jako G85. Zpětný pohyb do R1 však probíhá v posuvu.

**Programování**

Syntaxe:

G86[<Z>, <R1>, {<P>}, {<R2>}]

Zvláštnosti a omezení:

- Tento vrtací cyklus se provádí v každé nové pojzdové větě.
- Zrušení s G81.
- Před změnou cyklu je třeba zrušit aktuální cyklus pomocí G80.

G-kódy G184

3.2.39 Vrtací cyklus: Řezání vnitřních závitů bez vyrovnávacího sklíčidla G184

Používání

- Řezání vnitřních závitů (levých a pravých) **bez** vyrovnávacího sklíčidla.

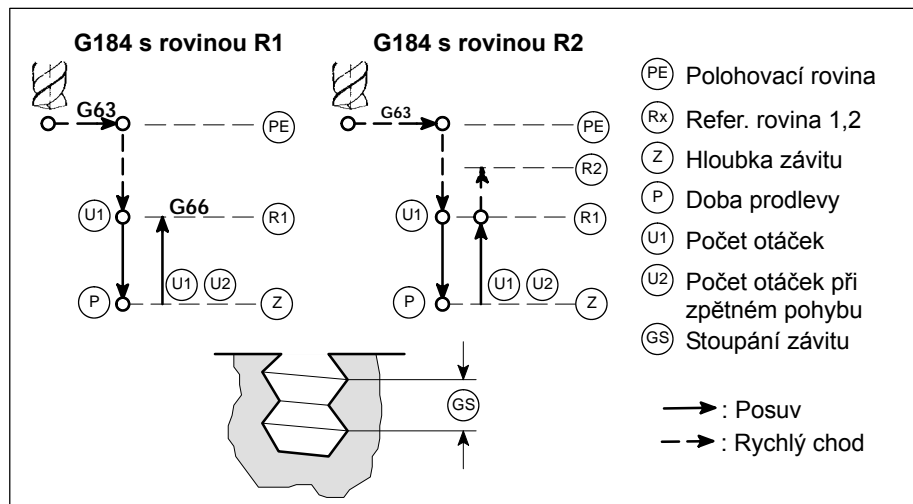
Předpoklady:

- regulované vřeteno
- G63 (řezání vnitřních závitů bez vyrovnávacího sklíčidla).

Následek

Přísuv nástroje se vypočítává interně pomocí součinu "počet otáček x stoupání závitu" ($U1 \cdot GS$). Směr otáčení (pravý nebo levý závit) se vybírá pomocí znaménka parametru GS (stoupání závitu).

Když je dosažena hloubka vrtání Z (hloubka závitu), obrátí se směr otáčení. Potom proběhne zpětný pohyb v posuvu do referenční roviny. Směr otáčení vřetena zůstane zachován, dokud nenaprogramujete nový cyklus řezání vnitřních závitů.



Programování

Syntaxe:

Pravý závit:

G184 [<Z>, <R1>, {<P>}, {<R2>}, <GS>, <U1>, {<U2>}, {<RP*>}]

Levý závit:

G184 [<Z>, <R1>, {<P>}, {<R2>}, - <GS>, <U1>, {<U2>}, {<RP*>}]

*) RP určuje polohu vřetena.

 **P smí být z důvodů kompatibility naprogramováno, ale nevyhodnocuje se!**

Zvláštnosti a omezení:

- Tento vrtací cyklus se provádí v každé nové pojezdové větě.
- Zrušení s G81.
- Před změnou cyklu je třeba zrušit aktuální cyklus pomocí G80.

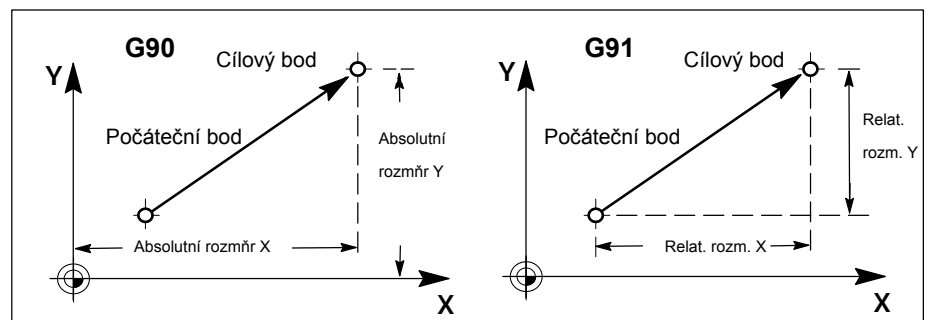
G-kódy G90 G91 AC(...) IC(...)

3.2.40	Programování absolutních rozměrů	G90
	Programování relativních rozměrů	G91
	Lokální programování absolutních rozměrů	AC(...)
	Lokální programování relativních rozměrů	IC(...)

Následek

Určuje, jestli má řízení interpretovat rozměrové údaje pro osy a souřadnice jako absolutní nebo relativní (inkrementální) hodnoty.

- Absolutní rozměrové údaje se vztahují k aktuálnímu nulovému bodu (v programové soustavě souřadnic).
- Relativní rozměrové údaje se vztahují k poslední najeté poloze.



Platí:

- Funkce G90 a G91 působí modálně a vzájemně se ruší.
- AC(...) a IC(...) působí nezávisle na aktivní funkci G90/G91 pouze pro informace o osách, jejichž poloha je s použitím AC(...), resp. IC(...) naprogramována.

Programování

Syntaxe:

G90	Programování absolutních rozměrů
G91	Programování relativních rozměrů
<Osa>=AC(<Hodnota>)	Lokální programování absolutních rozměrů.
<Osa>=IC(<Hodnota>)	Lokální programování relativních rozměrů.

kde

<Osa>: Logická osová adresa.

<Hodnota>: Informace o dráze pro <osu>.

Příklad:

N10 G1 G90	Zapnout programování absolutních rozměrů.
N20 X100 Y100	Najet na souřadnice X100, Y100.
N30 G91	Zapnout programování relativních rozměrů.
N40 X100 Y100	Najet na souřadnice X200, Y200.
N50 X=AC(50) Y50	Lokální programování absolutních rozměrů pro osu X. Pojezdový pohyb stroje na X50, Y250.

G-kódy G93

3.2.41 Časové programování


G93

Následek

Řízení interpretuje F-slova (popis viz kapitola "Pomocné a přídavné funkce") jako dobu obrábění v sekundách pro naprogramovanou dráhu (trvání věty).

Platí:

- Funkce G93, G94 a G95 působí modálně a vzájemně se ruší.

 **Požadovaný zapnutý stav (G93, G94 nebo G95) se dá určit v parametrech stroje (standardně: G94).**

Programování

Syntaxe:

G93

Přepnutí na časové programování.

Příklad:

N5 **G93** G1 X30 Y20 **F20** Naprogramovaná lineární interpolace trvá
: 20 sekund.

Zvláštnosti a omezení:

- Během funkce G93 naprogramované F-slovo zůstává při přepnutí na G94 nebo G95 interně uloženo a při opětovném přepnutí na G93 se znovu aktivuje.
- Po spuštění nebo základním nastavení je aktivní F-slovo stanovené v parametrech stroje (standardně: F0).
- Řízení interně vypočítává potřebný posuv z délky dráhy příslušné pojezdové věty a naprogramované doby obrábění. Definitivně účinný posuv však může být řízením omezen podle naprogramované dráhy a maximálních hodnot zúčastněných os, takže se trvání věty prodlouží.
- Override posuvu zůstává i při funkci G93 účinný v závislosti na funkci OvrEna / OvrDis.

G-kódy G94

3.2.42 Programování posuvu (za min)

G94


Následek

Řízení interpretuje F-slova (viz strana 3-96) nebo Omega-slova (viz strana 3-97) jako posuv pro naprogramovanou dráhu.

Platí

- Funkce G93, G94 a G95 působí modálně a vzájemně se ruší.
- Naprogramovaný posuv je interpretován
 - jako mm/min při aktivní funkci G71
 - jako Inch/min při aktivní funkci G70
 - jako stupně/min ve spojení s kruhovými osami.

 **Podle parametrů stroje se dá nastavit vážení posuvu pro G70/G71.**

 **Požadovaný zapnutý stav (G93, G94 nebo G95) se dá určit v parametrech stroje (standardně: G94).**

Programování

Syntaxe:

G94

Přepnutí na programování posuvu.

Příklad:

N10 G71

Zapnout metrické programování.

N20 G1 **G94** X20 Y30 **F200**

Pojíždět s posuvem 200 mm/min.

N30 G4(F40)

Doba prodlevy 40 sekund.

N40 G70

Zapnout programování Inch.

N60 X300 Y400

Znovu působí posuv F200 (v mm/min).

N70 **F100**

Nová hodnota posuvu: 100 Inch/min.

:

Zvláštnosti a omezení:

- Během funkce G94 naprogramované F-slovo nebo Omega-slovo zůstává při přepnutí na G93 nebo G95 interně uloženo a při opětovném přepnutí na G94 se znovu aktivuje.
- Po spuštění nebo základním nastavení je aktivní F-slovo stanovené v parametrech stroje (standardně: F0). Omega-slovo se po spuštění vždy automaticky inicializuje na "0".
- Účinný posuv může být řízením omezen v závislosti na zkonfigurovaných maximálních rychlostech zúčastněných os.
- Posuv je v závislosti na funkci OvrEna / OvrDis ovlivňován hodnotou Override posuvu.

G-kódy G94(...)

3.2.43 Inkrementální programování rychlosti s přizpůsobením zrychlení

G94(...)

Následek

Mění posuv nebo počet otáček s ohledem na poslední aktivně působící hodnotu. Přitom se zrychlení ve větě G94(...) přizpůsobuje tak, že jsou výsledná hodnota rychlosti, resp. počtu otáček dosažena až na konci věty. To vede k velmi měkkému chování zrychlení.

Platí:

- Funkce G94 je modální, ale z ní vyplývající posuv nemožný pro následující věty.
- Jednotka inkrementálního posuvu je podle G70/G71 Inch/min nebo mm/min.
- Posuv je v závislosti na funkci OvrEna / OvrDis ovlivňován hodnotou Override posuvu.

Programování

Syntaxe:

G94(DF<F-hodnota>)

Inkrementální programování
rychlosti posuvu

G94(DS1<S-hodnota>)

Inkrementální programování
počtu otáček vřetena 1

G94(DF<F-hodnota>,DS7<S-hodnota>)

Inkrementální programování
rychlosti posuvu a počtu otáček
vřetena 7

kde

<F-hodnota>: Inkrementální rychlost posuvu. Kladné hodnoty zvyšují a záporné snižují aktuálně účinnou rychlost posuvu.

<S-hodnota>: Inkrementální počet otáček vřetena. Kladné hodnoty zvyšují a záporné snižují aktuálně účinný počet otáček vřetena.

Příklad:

N30 G94(DF100) X250 Y300

:

N50 G94(DF- 50) X300 Y200

:

N70 G94(DS1=100) X25 Y30

:

N90 G94(DF100, DS7=150) X2 Y2

:

Lineárně zvyšovat posuv dráhy až
do konce věty o 100 mm/min.Lineárně snižovat posuv dráhy až do
konce věty o 50 mm/min.Lineárně zvyšovat počet otáček
vřetena 1 do konce věty o 100 ot./
min.Do konce věty lineárně zvyšovat po-
suv dráhy o 100 mm/min a počet
otáček vřetena 2 o 150 ot./min.

G-kódy

G94(...)

Zvláštnosti a omezení:

- Stávající mezní hodnoty zrychlení nebo brzdění jsou monitorovány. Kvůli tomu může být případně možné dosáhnout výsledné koncové rychlosti až v další větě.
- Vypočítané zrychlení je účinné pouze ve větě G94(...). Při přerušení věty probíhá brzdění s vypočítaným zrychlením.
- Absolutní a inkrementální posuv nesmí být naprogramovány současně v jedné větě.

G-kódy G95

3.2.44 Programování posuvu (na ot.)

G95

Následek

Řízení interpretuje F-slova (viz strana 3-96) jako posuv/otáčku. To je nutné v souvislosti s hlavním vřetenem. Platí

- Funkce G93, G94 a G95 působí modálně a vzájemně se ruší.
- Naprogramovaný posuv je interpretován
 - jako mm/ot. při aktivní funkci G71
 - jako Inch/ot. při aktivní funkci G70

☞ **Podle parametrů stroje se dá nastavit vážení posuvu pro G70/G71.**

☞ **Požadovaný zapnutý stav (G93, G94 nebo G95) se dá určit v parametrech stroje (standardně: G94).**

☞ **Definice hlavního vřetena viz MP 7020 00010, resp. funkce "MainSp".**

Programování

Syntaxe:

G95

Přepnutí na rotační programování posuvu.

Příklad:

N05 G71

N10 S200 M4

N20 G1 G95 X20 Z30 F0. 2

N30 G4(S20)

N40 G70

N60 X300 Z40

N70 F0. 1

:

Zapnout metrické programování.

Počet otáček vřetena 200 ot./min, levotočivý chod.

Pojíždět s posuvem 0,2 mm/ot.

Doba prodlevy 20 otáček.

Zapnout programování Inch.

Znovu působí posuv F0.2 (v mm/ot.).

Nový posuv 0,1 Inch/ot.

Zvláštnosti a omezení:

- G95 vyžaduje otáčející se hlavní vřeteno.
- Účinný posuv je ovlivňován jak potenciometrem vřetena, tak i potenciometrem posuvu.
- Účinný posuv může být řízením omezen v závislosti na zkonfigurovaných maximálních rychlostech zúčastněných os.
- Během funkce G95 naprogramované F-slovo zůstává při přepnutí na G93 nebo G94 interně zachováno a při opětovném přepnutí na G95 se znovu aktivuje.
- Po spuštění nebo základním nastavení je aktivní F-slovo stanovené v parametrech stroje (standardně: F0).

G-kódy G96 G97

3.2.45 Konstantní řezná rychlost Přímé programování počtu otáček

G96
G97

Následek

Řízení během obrábění soustružením interpretuje S-slova jako

- požadovanou řeznou rychlost nástroje (G96) nebo
- počet otáček rotační osy obrobku (G97).

Řezná rychlost závisí při daném počtu otáček vřetena na vzdálenosti řezného nástroje od rotační osy obrobku.

Pro kompenzaci této skutečnosti mění funkce G96 v závislosti na vzdálenosti mezi řezným nástrojem a rotační osou obrobku automaticky počet otáček rotační osy:

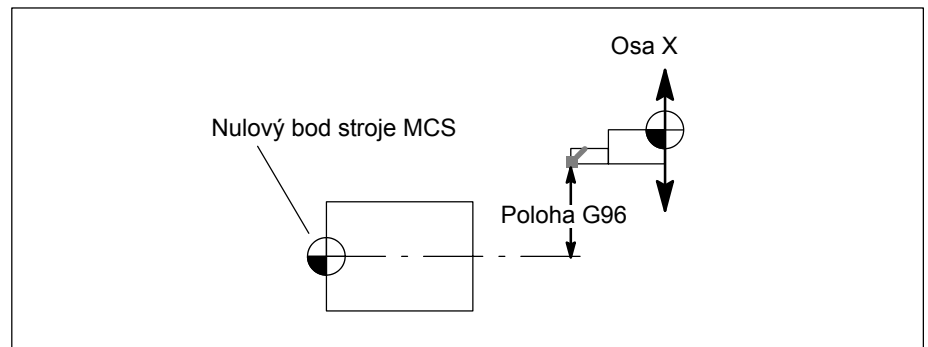
- při G71:

$$S_{Akt}[min^{-1}] = \frac{S_{G96} \times 1000}{2 \times \pi \times |Poloha(souradnice, vztažnýsystém) - opravanástroje|} \left[\frac{m \times 1000}{min \times mm} \right]$$

- při G70:

$$\left[\frac{feet \times 12}{min \times inch} \right]$$

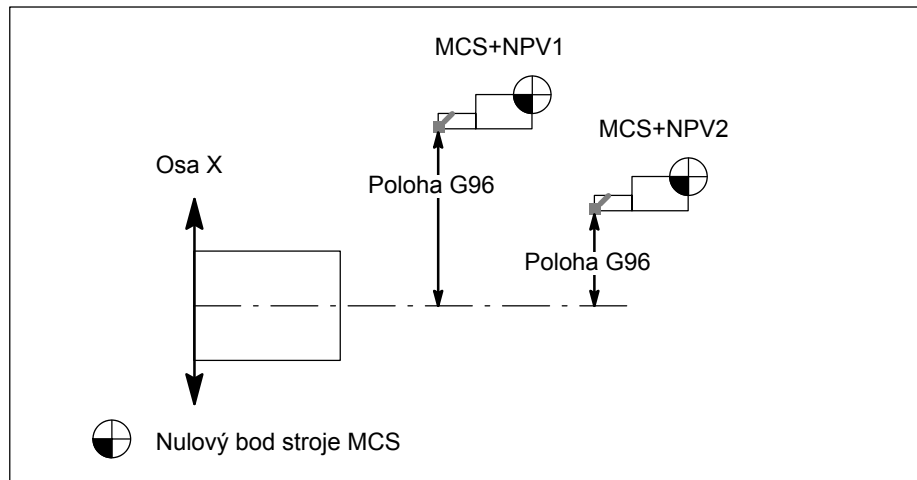
Standardně vypočítává NC vzdálenost v soustavě souřadnic stroje (vztažná soustava MCS):



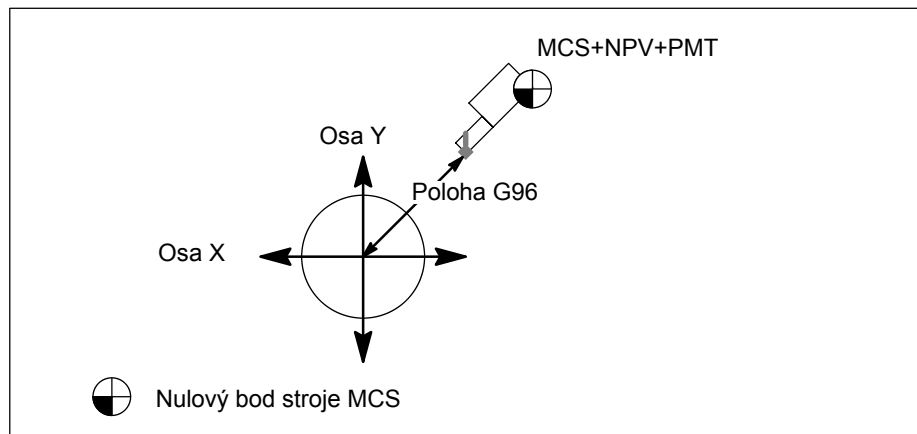
G-kódy

G96 G97

Pokud musí být vřetenem polohováno mezi různé body upnutí nástroje, může G96 zahrnout odpovídající posunutí nulových bodů, která definují lokální soustavu souřadnic (vztažný systém LCS):



Pro nástroje, které nejsou uspořádány osově paralelně s vřetenem, mohou být navíc brána v úvahu posunutí a otočení (umístění) nulového bodu obrobku (vztažná soustava PCS):



Řezná rychlost na obrobku pak již není ovlivňována vzdáleností mezi řezným nástrojem a rotační osou obrobku.

Pokud toto chování není potřebné nebo žádoucí, použijte G97. Počet otáček rotační osy obrobku je pak určen výhradně naprogramovaným S-slovem.

Platí:

- Vztažná osa pro vzdálenost mezi řezným hrotem a rotační osou obrobku je definována pomocí parametru stroje 7010 00110 (standardní hodnota)
- Vztažná osa se dá v průběhu programu přeprogramovat.
- Vztažná soustava souřadnic (působíště) pro vztažnou osu je nastavena parametrem stroje 7010 00120. Na výběr jsou:
 - PCS: Poloha v programové soustavě souřadnic

G-kódy

G96 G97

- LCS: Poloha v lokální soustavě souřadnic stroje (posunutě o NPV)
- MCS: Poloha v soustavě souřadnic stroje (standardní)
- Působíště vztažné osy se dá v průběhu programu přepnout.
- Naprogramovaná řezná rychlost je interpretována jako
 - m/min při aktivní funkci G71
 - feet/min při aktivní funkci G70 (1 feet = 12 Inch).
- Kromě Override vřeten jsou účinné také funkce pro omezení počtu otáček (SMin, SMax; viz strana 4-45).

Programování

Syntaxe:

G96{({<Vztažná osa>{<Působíště>}})} ... S<i>=<V>

S-slova vřeten, která jsou naprogramována ve větě G96, jsou při obrábění soustružením interpretována jako požadovaná řezná rychlost na nástroji. Všechna ostatní vřeten se vrátí k přímému programování počtu otáček!

G96

Jsou aktivní poslední naprogramovaná nastavení; pokud žádná nebyla provedena, platí standardní hodnoty.

G96() ...

Znovu se aktivují standardní hodnoty pro vztažnou osu a působíště.

G97 ... S<i>=<Počet otáček>

K přímému programování počtu otáček se vrátí pouze vřeten, jejichž S-slova jsou naprogramována ve větě G97. Naprogramovaná S-slova jsou pro tato vřeten interpretována jako počet otáček rotační osy obrobku.

G97

Všechna vřeten se vrátí k přímému programování počtu otáček.

kde

<Vztažná osa> Logický nebo fyzický název vztažné osy. Naprogramování zůstává zachováno, dokud není naprogramována nová hodnota nebo aktivována standardní hodnota.

<Působíště>

Soustava souřadnic vztažné osy:
PCS, LCS nebo MCS

Naprogramování zůstává zachováno, dokud není naprogramována nová hodnota nebo aktivována standardní hodnota.

<i>

Index vřeten.

1: první vřeten; 2: druhé vřeten atd.

<V>

Řezná rychlost příslušného vřeten v m/min nebo feet/min.

G-kódy G96 G97

<Počet otáček> Počet otáček příslušného vřetena v ot./min.

Příklad (v systému jsou zkonfigurována 2 vřetena):

N10	G71	S1=500	S2=500	Poprvé v programu definovat požadované otáčky obou vřeten.
:				
N60	G96	S1=50		Aktivovat G96 pro 1. vřeteno s řeznou rychlostí 50 m/min.
:				
:				2. vřeteno běží ještě s programováním počtu otáček. G96 je aktivní.
:				
N100		S1=30		Snižte řeznou rychlost 1. vřetena na 30 m/min.
:				
:				2. vřeteno běží ještě s programováním počtu otáček. G96 je aktivní.
:				
N140	G96(Y, LCS)		S2=100	Vztažnou osou je Y, působíštěm lokální soustava souřadnic stroje LCS.
:				
:				Aktivovat řeznou rychlost 100 m/min pro 2. vřeteno.
:				
:				1. vřeteno se vrátí k programování počtu otáček. G96 je aktivní.
:				
N180	G97			Všechna vřetena se vrátí - pokud se to ještě nestalo - k programování počtu otáček. G97 je aktivní.
:				
:				

Zvláštnosti a omezení:

- Ve stejné větě se dá společně přepínat více vřeten tak, že se odpovídající S-slova naprogramují za sebou (příklad: G96 S1=100 S2=1000).
- Chcete-li změnit řeznou rychlost vřetena již přepnutého na G96 v dalším průběhu programu, musíte pouze nově naprogramovat S-slovo příslušného vřetena.
- Aktuálně působící řezná rychlost vřetena zůstává po přepnutí na G97 interně uložena. Znovu se aktivuje, jakmile přepnete příslušné vřeteno zpět na G96.
- Při přepnutí z G96 na G97 přebírá řízení pro všechna vřetena, jejichž S-slovo není naprogramováno ve větě G97, jejich aktuální počet otáček jako novou požadovanou hodnotu počtu otáček.
- Případně požadovaná změna převodového stupně musí být provedena před aktivací G96.
- Při aktivní funkci G96 nemění funkce "automatické přepínání převodových stupňů" žádné převodové stupně.
- Při aktivní funkci G96 kontroluje řízení, jestli existuje "průnik" vztažné osy s aktivním působíštěm. Pokud tomu tak není (např. když osa opustí kanál), je ohlášena běhová chyba.

G-kódy G140, G141, G142

3.2.46 3-D oprava poloměru nástroje

G140, G141, G142

Následek

Tato funkce pohybuje s rotačně symetrickým nástrojem doleva nebo doprava po naprogramované dráze a vede přitom jeho bod záběru v aktivním posuvu po naprogramované dráze.

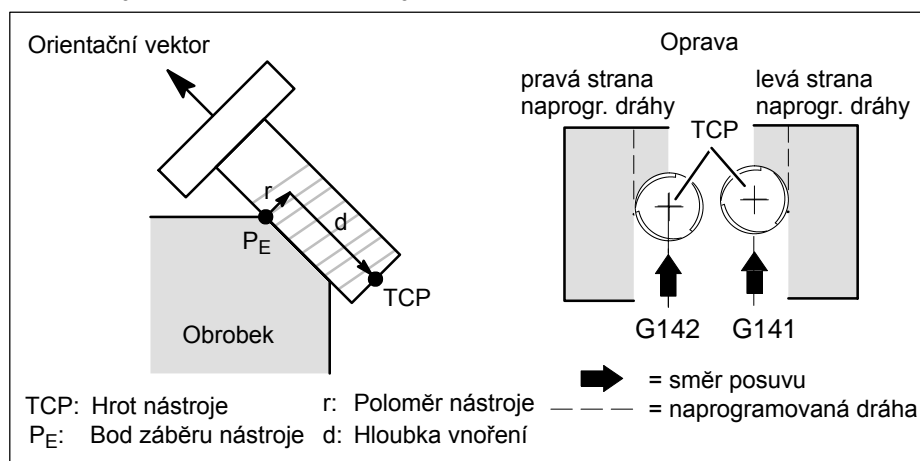
Při čistě orientačních pohybech zůstává aktuální poloha bodu záběru pevná a pohybuje se TCP.

Na vnějších okrajích se průběh dráhy uzavírá automaticky vytvořenou vloženou větou (kruhový oblouk). Na vnitřních okrajích probíhá výpočet průřezu.

Při opravě bere řízení v úvahu

- poloměr nástroje r (je automaticky převzat z aktivních dat nástroje D, G146 až G846 nebo G147 až G847)
- hloubku vnoření d (viz syntaxe **INSDEP**)
- volitelný přídavek k dráze, který působí aditivně na poloměr nástroje (viz syntaxe **COFFS**).

Následující obrázek znázorňuje princip:



Změny hloubky vnoření (**INSDEP**), přídavku (**COFFS**), délky nástroje nebo poloměru frézy se při aktivní funkci G141/G142 provádějí okamžitě a hladce (pomocí spline křivek).

☞ **Podrobný popis funkce naleznete v příručce "Funkční popis".**



POZOR

Hodnoty oprav jsou za určitých okolností okamžitě najížděny nebo odjížděny bez naprogramování zvláštního pojezdového pohybu. To může vést k poškození obrobku nebo nástroje.

Věnujte proto pozornost všem informacím v této kapitole a v příručce "Funkční popis".

G-kódy G140, G141, G142

Programování**Syntaxe:**

G140	Vypnutí 3-D opravy poloměru nástroje (Zapnutý stav po spuštění řízení). Oprava (poloměr, hloubka vnoření) se vytváří pomocí interpolace s případně naprogramovaným pojezdovým pohybem (pomocí spline křivek 3. stupně). Účinná oprava délky nástroje není funkcí G140 ovlivněna.
G141	Zapnutí 3-D opravy poloměru nástroje na levé straně dráhy (z pohledu ve směru obrábění při kladných hodnotách opravy). Pojezdové nebo orientační pohyby smí být naprogramovány ve stejné větě. Poloměr a hloubka vnoření jsou najety, když je ve stejné větě naprogramována jedna z aktivních prostorových souřadnic. Jinak vytvoří řízení opravu <ul style="list-style-type: none"> • teprve s následující pojezdovou větou, pokud byla předtím aktivní funkce G140. • okamžitě (v aktivním posuvu), pokud byla předtím aktivní funkce G141/G142.
G142	Zapnutí 3-D opravy poloměru nástroje na pravé straně dráhy . Jinak jako u G141.
INSDEP<ET>	Definuje hloubku vnoření <ET>. Působí modálně. Zapnutý stav po spuštění řízení: 0
COFFS<KA>	Definuje přídavek <KA>. Působí modálně a aditivně k poloměru nástroje r. Zapnutý stav po spuštění řízení: 0

Zvláštnosti a omezení:

- Funkce G140, G141 a G142 působí modálně a vzájemně se ruší.
- Vytvoření opravy na libovolné kontuře (G2, G6 atd.).
- Při aktivní funkci G141 nebo G142 **nejsou dovoleny**
 - G17 ... G20 (přepínání rovin)
 - G70, G71 (přepínání Inch/metrické)
 - G63 (řezání vnitřních závitů bez vyrovnávacího sklíčidla)
 - G74 (njetí na souřadnice referenčního bodu)
 - G75 (vstup měřicího dotyku)
 - G76 (njetí do pevné osové polohy stroje)
 - SetPos (nastavení programové polohy)
 - G54.x ... G59.x (posunutí nulových bodů)
 - G154.x ... G159.x (šikmá rovina)
 - G40 ... G42 (oprava poloměru nástroje)
 - Coord() (zapnutí/vypnutí/přepnutí transformace os)
 - Předávání os, které jsou díky transformaci os/souřadnic zúčastněny na aktuálních prostorových souřadnicích.

G-kódy

G140, G141, G142

- Při aktivní funkci G141 nebo G142 neprobíhá kontrola kolizí.
- Skončí-li program bez M30, musí být v tomto okamžiku aktivní funkce G140. Aktuální hodnoty pro hloubku vnoření a přídavek zůstávají zachovány.
- Po základním nastavení, základním nastavení systému nebo M30 se automaticky aktivuje funkce G140. Aktuální hodnoty pro hloubku vnoření a přídavek jsou nastaveny na 0.

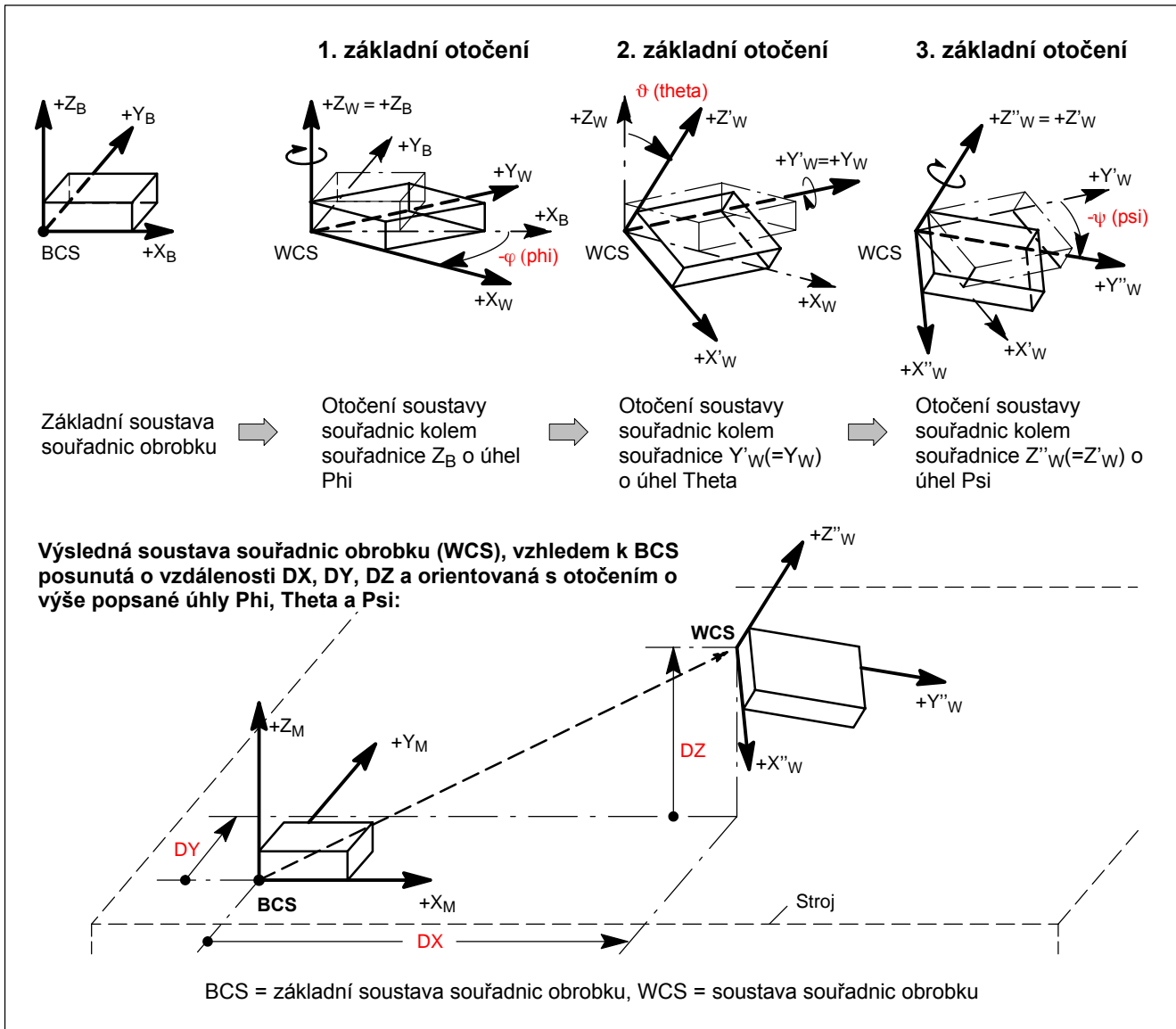
G-kódy G152 - G159.5

3.2.47 Umístění: Šikmá rovina

G152 - G159.5

Následek

Umístění "šikmá rovina" může libovolně posouvat a orientovat soustavu souřadnic obrobku v prostoru. Šikmá rovina působí v příslušném kanálu na souřadnice s významy "x", "y" a "z".
Protože pro orientaci existují 3 stupně volnosti, dá se každá orientace zobrazit 3 po sobě následujícími základními otočeními:



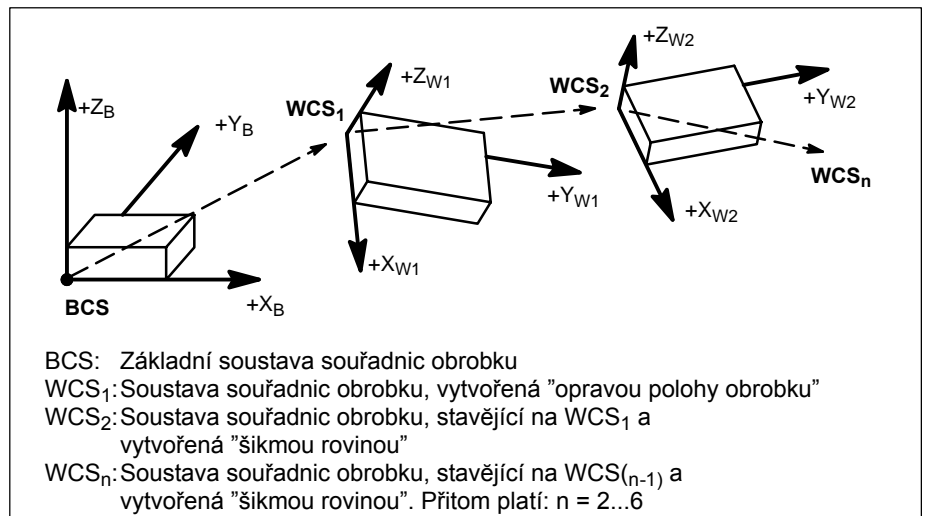
G-kódy

G152 - G159.5

Výsledná funkce "šikmé roviny" je ovlivněna

- 5 řadami (1 až 5), které na sebe navzájem působí aditivně.
Tak lze vytvořit až 5 soustav souřadnic obrobku, které se skládají na sebe.
- 7 alternativních sad na řadu.
Každá sada obsahuje všechna potřebná data posunutí a orientace pro soustavu souřadnic obrobku.
V řadě může být v daném okamžiku aktivní vždy pouze jediná sada, tzn. sady jedné řady se vzájemně ruší.

Šikmá rovina působí aditivně k umístění "oprava polohy obrobku" a nachází se tedy ve "výpočetním řetězci" za opravou polohy obrobku:



Programování ...

Všechna data potřebná pro posunutí a orientaci soustavy souřadnic obrobku jsou zaznamenána v "sadě".

Sady mohou být buď

- naprogramovány přímo v obráběcím programu jako funkční parametry, nebo
- zadány v tabulce umístění (viz strana 4-15).
Tabulka umístění může obsahovat max. 30 sad (5 řad, 6 sad na řadu).

G-kódy

G152 – G159.5

... přímo v programovém řádku jako parametry:

Syntaxe:

G152.<Řada>({<X_W-Offset>},{
{<Y_W-Offset>},{<Z_W-Offset>},{
{<Úhel1>},{<Úhel2>},{
{<Úhel3>}}}}))

Zapnutí šikmé roviny požadované řady, s naprogramovanými daty posunutí/orientace.

Zkratka: G152

odpovídá G152.1

G153.<Stupeň>

Vypnutí šikmé roviny naprogr. řady.

G153

Vypnutí šikmé roviny (vypnutí všech řad).

kde

<Řada>: Číslo požadované řady.
Rozsah hodnot: 1 až 5

<X_W-offset>: Hodnota posunutí ve směru hlavní souřadnice

<Y_W-offset>: Hodnota posunutí ve směru vedlejší souřadnice

<Z_W-offset>: Hodnota posunutí ve směru normálové souřadnice

<Úhel1>: Úhel otočení kolem souřadnice Z.
Rozsah hodnot: $0 \leq \langle \text{úhel1} \rangle < 360$ stupňů

<Úhel2>: Úhel otočení kolem nové souřadnice Y'.
Rozsah hodnot: $0 \leq \langle \text{úhel2} \rangle < 180$ stupňů

<Úhel3>: Úhel otočení kolem nové souřadnice Z''.
Rozsah hodnot: $0 \leq \langle \text{úhel3} \rangle < 360$ stupňů

Příklad:

:	
N40 G152. 1(100, 0, 0, 90)	Šikmá rovina, řada 1 zapnout.
:	Žádný pojezdový pohyb.
:	Nulový bod je posunutý o 100 mm ve
:	směru X a soustava WCS otočená o 90
:	stupňů kolem souřadnice Z.
:	
N180 G153	Vypnout všechny řady (úplně vypnout šikmou rovinu).

... ve spojení s tabulkami umístění:

1. Aktivujte požadovanou tabulku umístění (viz strana 4-79).
2. Naprogramujte požadovanou funkci:

Syntaxe:

G154.<Řada>

Zapnutí šikmé roviny naprogr. řady, sada 1

G155.<Řada>

Zapnutí šikmé roviny naprogr. řady, sada 2

G156.<Řada>

Zapnutí šikmé roviny naprogr. řady, sada 3

G157.<Řada>

Zapnutí šikmé roviny naprogr. řady, sada 4

G158.<Řada>

Zapnutí šikmé roviny naprogr. řady, sada 5

G-kódy

G152 - G159.5

G159.<Řada>	Zapnutí šikmé roviny naprogr. řady, sada 6
Zkratka: G152 - G159	odpovídá G154.1 - G159.1
G153.<Řada>	Vypnutí šikmé roviny naprogr. řady.
G153	Vypnutí šikmé roviny (vypnutí všech řad).
kde	
<Řada>:	Číslo požadované řady. Rozsah hodnot: 1 až 5

 **Pokyny pro vytváření nebo úpravy tabulek umístění naleznete v návodu k obsluze.**

Příklad:

N40 PMS(Tab1)	Aktivovat tabulku umístění "Tab1".
N50 G154. 1	Zapnout řadu 1 se sadou 1. Žádný pojezdový pohyb.
:	
N90 G154. 2 X1 Y2 Z3	Zapnout řadu 2 se sadou 1 (působí aditivně k řadě 1). Ve výsledné soustavě WCS najet do naprogramované polohy P(1,2,3).
:	
:	
N120 G153. 2	Vypnout řadu 2 (řada 1 zůstává zapnutá).
:	
N180 G153	Vypnout všechny řady (úplně vypnout šikmou rovinu).

Zvláštnosti a omezení:

- Pokud je aktivní transformace os, nesmí být šikmá rovina naprogramována společně s pojezdovými pohyby ve stejné větě.
- Zapnutý stav a chování při základním nastavení jsou zkonfigurovány v parametrech stroje 7060 00010 a 7060 00020.
- Všechny funkce působí v každé řadě (1...5) modálně a vzájemně se ruší.
- Aktivace a deaktivace šikmé roviny přerušuje náhled vět a nesmí být proto naprogramována při aktivní opravě dráhy frézy (G41/G42, viz strana 3-39).

M-kódy M0, M00 M1, M01

3.3 M-kódy

3.3.1 Přerušení programu (zastavení programu)

M0, M00

Následek

- Přerušuje NC-program,
- zastavuje pohyby stroje po provedení věty a
- vydává kanálový signál rozhraní "Zastavení programu M0".

Aktuální stav kanálu se změní na "NC připraveno".

Při opětovném použití "NC-Start" pokračuje provádění programu.

 **Informace o přerušení programu v závislosti na kanálovém signálu rozhraní "Volitelné zastavení" viz M1/M01.**

Programování

Syntaxe:

M0 nebo **M00**

Zvláštnosti a omezení:

- "Zastavení programu" můžete naprogramovat společně s dalšími NC-funkcemi ve stejné větě. Teprve po provedení všech ostatních naprogramovaných funkcí se aktivuje "Zastavení programu".

3.3.2 Podmíněné přerušení programu (podm. zastavení programu)

M1, M01

Následek

- Přerušuje NC-program a
- zastavuje pohyby stroje po provedení věty, pokud je kanálový signál rozhraní nastaven na "Volitelné zastavení".

Aktuální stav kanálu se změní na "NC připraveno".

Při opětovném použití "NC-Start" pokračuje provádění programu.

Programování

Syntaxe:

M1 nebo **M01**

Zvláštnosti a omezení:

- "Podmíněné zastavení programu" můžete naprogramovat společně s dalšími NC-funkcemi ve stejné větě. Teprve po provedení všech ostatních naprogramovaných funkcí se aktivuje "Podmíněné zastavení programu".

M-kódy M2, M02, M30

3.3.3 Ukončení programu (konec programu)

M2, M02, M30

Následek

Ukončuje program.

Je-li program podprogramem,

- vydá NC pomocnou funkci (M2, M02, resp. M30),
- skočí zpět do volajícího programu a
- dál zpracovává volající program.

Modální stavy změněné v podprogramu se přitom neresetují!

Je-li program hlavním programem,

- nastaví NC kanálový signál rozhraní na "Konec programu M30",
- resetuje kanálový signál rozhraní "Program běží",
- deaktivuje příp. aktivní "automatickou volbu převodových stupňů" (aktuální převodový stupeň ale zůstane zvolen)
- přepne na "přímé programování počtu otáček" (G97)
- aktivuje všechny stavy, které jsou definovány v parametru stroje 7060 00020 "Zapnuté stavy" pro událost "M30",
- skočí na začátek hlavního programu a
- čeká na nový "NC-Start".

Nový "NC-Start" znovu spustí provádění programu.

Programování

Syntaxe:

M2 nebo **M02** nebo **M30**

Zvláštnosti a omezení:

- Naprogramujte tuto funkci v samostatném programovém řádku.



POZOR

Možné poškození kvůli nedefinovaným zapnutým stavům!

Pokud jsou po skončení hlavního programu zapotřebí určité stavy nebo funkce, musí být zaručeno, aby byl správně parametrizován Initstring události "M30" v MP 7060 00020! Ten musí obsahovat všechny funkce, které po skončení hlavního programu uvedou NC do požadovaného/žádoucího stavu.

Pamatujte si v této souvislosti, že modálně účinné funkce zůstávají aktivní ještě i po skončení programu!

 **Další informace o zapnutých stavech naleznete v příručce "Funkční popis".**

M-kódy M3, M103, M203 M13, M113, M213

3.3.4 Pravotočivý chod vřetena Pravotočivý chod vřetena a zapnutí chladicí kapaliny

M3, M103, M203

M13, M113, M213



POZOR

Popsaná syntaxe platí pouze pro standardní nastavení oblastí parametrů stroje 1040 001xx a 1040 002xx.


Je volně konfigurovatelná a může proto být na vašem stroji jiná!

Přednost má proto vždy příslušná dokumentace WMH.

Zeptejte se pracovníka odpovědného za systém, jestli zde popsané funkce platí i pro váš stroj!

Následek

- Spouští - při pohledu ve směru "od nástroje k obrobku" - pravotočivý chod vřetena.
- Ruší regulaci polohy aktivovanou příkazem "Seřízení/polohování vřetena".
- Rezervuje příslušná vřetena pro aktuální kanál.

 **Pohyb vřetena se spustí, pouze když bylo předtím nebo ve stejné větě naprogramováno S-, resp. SSPG-slovo větší než 0 pro příslušné vřeteno či skupinu vřeten (viz strana 3-98).**

Programování

Syntaxe:

M3 Působí na 1. skupinu vřeten.

M103 Působí na 1. vřeteno.

M203 Působí na 2. vřeteno.

M13 Působí na 1. skupinu vřeten. Navíc zapnutí chladicí kapaliny.

M113 Působí na 1. vřeteno. Navíc zapnutí chladicí kapaliny.

M213 Působí na 2. vřeteno. Navíc zapnutí chladicí kapaliny.

Zvláštnosti a omezení:

- Příslušná funkce automaticky rezervuje odpovídající vřetena pro aktuální kanál. Využití rezervovaných vřeten jiným kanálem je možné až poté, co je pro příslušná vřetena použita funkce "Zastavení vřetena" (viz strana 3-90) nebo "SpAdmin" (SPA, viz strana 4-46).
- Příslušná funkce působí tak dlouho, dokud není pro stejná vřetena naprogramován nový pohybový stav (např. jiný směr otáčení, s chladicí kapalinou / bez, "zastavení vřetena" nebo "seřízení vřetena").
- Konkuroující si úlohy mezi jednotlivými vřeteny a skupinami vřeten, naprogramované ve stejné větě, vyvolávají běhovou chybu. (Příklad: Funkce M3 a M104 nejsou povoleny ve stejné větě).
- Po změně převodového stupně se automaticky obnoví dříve naprogramovaný směr otáčení vřetena.

M-kódy M4, M104, M204 M14, M114, M214

3.3.5 Levotočivý chod vřetena M4, M104, M204

Levotočivý chod vřetena a M14, M114, M214 zapnutí chladicí kapaliny



POZOR

Popsaná syntaxe platí pouze pro standardní nastavení oblastí parametrů stroje 1040 001xx a 1040 002xx.

Je volně konfigurovatelná a může proto být na vašem stroji jiná! Přednost má proto vždy příslušná dokumentace WMH.

Zptejte se pracovníka odpovědného za systém, jestli zde popsané funkce platí i pro váš stroj!

Následek

- Spouští - při pohledu ve směru "od nástroje k obrobku" - levotočivý chod vřetena.
- Ruší regulaci polohy aktivovanou příkazem "Seřízení/polohování vřetena".
- Rezervuje příslušná vřetena pro aktuální kanál.

Pohyb vřetena se spustí, pouze když bylo předtím nebo ve stejné větě naprogramováno S-, resp. SSPG-slovo větší než 0 pro příslušné vřeteno či skupinu vřeten (viz strana 3-98).

Programování

Syntaxe:

M4 Působí na 1. skupinu vřeten.

M104 Působí na 1. vřeteno.

M204 Působí na 2. vřeteno.

M14 Působí na 1. skupinu vřeten. Navíc zapnutí chladicí kapaliny.

M114 Působí na 1. vřeteno. Navíc zapnutí chladicí kapaliny.

M214 Působí na 2. vřeteno. Navíc zapnutí chladicí kapaliny.

Zvláštnosti a omezení:

- Příslušná funkce automaticky rezervuje odpovídající vřetena pro aktuální kanál. Využití rezervovaných vřeten jiným kanálem je možné až poté, co je pro příslušná vřetena použita funkce "Zastavení vřetena" (viz strana 3-90) nebo "SpAdmin" (SPA, viz strana 4-46).
- Příslušná funkce působí tak dlouho, dokud není pro stejná vřetena naprogramován nový pohybový stav (např. jiný směr otáčení, s chladicí kapalinou / bez, "zastavení vřetena" nebo "seřízení vřetena").
- Konkurojící si úlohy mezi jednotlivými vřeteny a skupinami vřeten, naprogramované ve stejné větě, vyvolávají běhovou chybu. (Příklad: Funkce M3 a M104 nejsou povoleny ve stejné větě).
- Po změně převodového stupně se automaticky obnoví dříve naprogramovaný směr otáčení vřetena.

M-kódy M5, M105, M205

3.3.6 Zastavení vřetena

M5, M105, M205

**POZOR**

Popsaná syntaxe platí pouze pro standardní nastavení oblastí parametrů stroje 1040 001xx a 1040 002xx.

Je volně konfigurovatelná a může proto být na vašem stroji jiná! Přednost má proto vždy příslušná dokumentace WMH.

Zeptejte se pracovníka odpovědného za systém, pokud si nebudete jisti, jestli zde popsané funkce platí i pro váš stroj!

Následek

- Zastavuje vřetena.
- Ruší případnou rezervaci uvedených vřeten v aktivním kanálu.
- Ruší regulaci polohy aktivovanou příkazem "Seřízení/polohování vřetena".

Programování

Syntaxe:

M5	Působí na skupinu vřeten 1.
M105	Působí na 1. vřeteno.
M205	Působí na 2. vřeteno.

Zvláštnosti a omezení:

- Příslušná funkce působí tak dlouho, dokud není pro stejná vřetena naprogramován nový pohybový stav (např. "levotočivý/pravotočivý chod vřetena" nebo "seřízení vřetena").
- Konkurující si úlohy mezi jednotlivými vřeteny a skupinami vřeten, naprogramované ve stejné větě, vyvolávají běhovou chybu. (Příklad: Funkce M3 a M105 nejsou povoleny ve stejné větě).

M-kódy M19, M119, M219

3.3.7 Seřízení vřetena / polohování vřetena

M19, M119, M219

**POZOR**

Popsaná syntaxe platí pouze pro standardní nastavení oblastí parametrů stroje 1040 001xx a 1040 002xx.

Je volně konfigurovatelná a může proto být na vašem stroji jiná! Přednost má proto vždy příslušná dokumentace WMH.

Zptejte se pracovníka odpovědného za systém, pokud si nebudete jisti, jestli zde popsané funkce platí i pro váš stroj!

Následek

Vřeteno je pro aktuální kanál rezervováno a polohováno v regulaci polohy na definované místo.

Polohováno je

- při zastavení: jak je určeno v parametru S-0-0154 pohonu
- při aktivním rotačním pohybu: při zachování směru otáčení.

Programování

Syntaxe:

M19	Skupina vřeten 1: polohovat všechna zúčastněná vřetena do jejich referenčních úhlů (parametr pohonu S-0-0153).
M119	1. Polohovat vřeteno do jeho referenčního úhlu (S-0-0153).
M219	2. Polohovat vřeteno do jeho referenčního úhlu (S-0-0153).
M19 S<Úhel>	Skupina vřeten 1: polohovat všechna zúčastněná vřetena do <úhlu>.
M119 S1=<Úhel>	1. Polohovat vřeteno do <úhlu>.
M219 S2=<Úhel>	2. Polohovat vřeteno do <úhlu>.
<Úhel>	Požadovaná absolutní poloha vřetena ve stupních. Rozsah hodnot: $0^\circ \leq \text{Poloha vřetena} < 360^\circ$. Pokud je jako poloha naprogramována jiná hodnota, je automaticky přepočítána do zadaného intervalu. Jestliže se vřeteno již nachází v zadané poloze, neproběhne žádný nový pohyb.

Zvláštnosti a omezení:

- Příslušná funkce nesmí být naprogramována ve stejné větě s konkurenční funkcí vřetena ("pravotočivý/levotočivý chod vřetena", "zastavení vřetena").
- Je-li zapotřebí S-slovo, musí být naprogramováno ve stejné větě.
- Po polohovací proceduře zůstává vřeteno v regulaci polohy. Regulace polohy se automaticky ruší teprve příkazem "pravotočivý/levotočivý chod vřetena" nebo "zastavení vřetena".

M-kódy M19, M119, M219

- Konkuruující si úlohy mezi jednotlivými vřeteny a skupinami vřeten, naprogramované ve stejné větě, vyvolávají běhovou chybu. (Příklad: Funkce M3 a M119 nejsou povoleny ve stejné větě).

Příklady:

N60	M19	Polohovat všechna vřetena 1. skupiny vřeten do jejich referenční úhlu.
:		
N70	M219	Pouze 2. vřeteno polohováno do svého referenčního úhlu.
:		
N80	M19 S180	Polohovat všechna vřetena 1. skupiny vřeten na 180 stupňů.
:		
N90	M119 S1=370	1. Vřeteno polohováno na 10 stupňů.
:		
N95	M19 S1=10 S2=20	Pokud je 1. a 2. vřeteno přiřazeno 1. skupině vřeten: 1. vřeteno polohováno na 10, 2. vřeteno na 20 stupňů. Polohovat všechna další vřetena 1. skupiny vřeten do jejich referenční úhlu.
:		

M-kódy M40, M140, M240

3.3.8 Automatická volba převodových stupňů

M40, M140, M240

**POZOR**

Popsaná syntaxe platí pouze pro standardní nastavení oblastí parametrů stroje 1040 001xx a 1040 002xx.

Je volně konfigurovatelná a může proto být na vašem stroji jiná! Přednost má proto vždy příslušná dokumentace WMH.

Zeptejte se pracovníka odpovědného za systém, pokud si nebudete jisti, jestli zde popsané funkce platí i pro váš stroj!

Následek

Řízení automaticky vybírá podle právě aktivního počtu otáček vhodný stupeň z převodových stupňů, které jsou k dispozici.

Jestliže se rozsahy otáček jednotlivých stupňů překrývají, vybírá řízení vždy nejbližší nižší stupeň (s vyššími otáčkami motoru).

Programování

Syntaxe:

M40	Zapnout automatickou volbu převodových stupňů pro skupinu vřeten 1.
M140	Zapnout automatickou volbu převodových stupňů pro 1. vřeteno.
M240	Zapnout automatickou volbu převodových stupňů pro 2. vřeteno.

Zvláštnosti a omezení:

- Naprogramování počtu otáček "0" nezpůsobuje žádnou změnu aktuálního převodového stupně.
- Funkce M40, M41-M44, M48, M140, M141-M144 a M148 působí modálně a vzájemně se ruší.
- Funkce M240, M241-M244 a M248 působí modálně a vzájemně se ruší.
- M30 deaktivuje příp. aktivní automatickou volbu převodových stupňů (aktuální převodový stupeň ale zůstane zvolen).
- Automatická volba převodových stupňů neúčinkuje na analogová vřetena.

M-kódy M41...44, M141...144, M241...244

3.3.9 Ruční volba převodových stupňů M41...44, M141...144, M241...244

**POZOR**

Popsaná syntaxe platí pouze pro standardní nastavení oblastí parametrů stroje 1040 001xx a 1040 002xx.

Je volně konfigurovatelná a může proto být na vašem stroji jiná! Přednost má proto vždy příslušná dokumentace WMH.

Zeptejte se pracovníka odpovědného za systém, pokud si nebudete jisti, jestli zde popsané funkce platí i pro váš stroj!

Následek

Vybírá odpovídající převodový stupeň.

Je-li počet otáček naprogramován mimo rozsah otáček příslušného převodového stupně, vydá řízení minimální, resp. maximální počet otáček pro příslušný převodový stupeň.

Programování

Syntaxe:

M4<Stupeň> Zapnout převodový stupeň <Stupeň> pro skupinu vřeten 1.

M14<Stupeň> Zapnout převodový stupeň <Stupeň> pro 1. vřeten.

M24<Stupeň> Zapnout převodový stupeň <Stupeň> pro 2. vřeten.

kde

<Stupeň> Rozsah zadání: 1 ... 4.

Zvláštnosti a omezení:

- Funkce M40, M41-M44, M48, M140, M141-M144 a M148 působí modálně a vzájemně se ruší.
- Funkce M240, M241-M244 a M248 působí modálně a vzájemně se ruší.
- Ruční volba převodových stupňů neúčinkuje na analogová vřeten.

M-kódy M48, M148, M248

3.3.10 Vyřazení převodového stupně

M48, M148, M248

**POZOR**

Popsaná syntaxe platí pouze pro standardní nastavení oblastí parametrů stroje 1040 001xx a 1040 002xx.

Je volně konfigurovatelná a může proto být na vašem stroji jiná! Přednost má proto vždy příslušná dokumentace WMH.

Zeptejte se pracovníka odpovědného za systém, pokud si nebudete jisti, jestli zde popsané funkce platí i pro váš stroj!

Následek

Vyřazuje převodový stupeň.
Převodovka se pak nachází v neutrálu.

Programování

Syntaxe:

M48 Vyřadit převodový stupeň pro skupinu vřeten 1.

M148 Vyřadit převodový stupeň pro 1. vřeteno.

M248 Vyřadit převodový stupeň pro 2. vřeteno.

Zvláštnosti a omezení:

- Funkce M40, M41-M44, M48, M140, M141-M144 a M148 působí modálně a vzájemně se ruší.
- Funkce M240, M241-M244 a M248 působí modálně a vzájemně se ruší.
- Funkce nepůsobí na analogová vřetena.

Vorschub/Drehzahl F

3.4 Programování posuvu a počtu otáček

3.4.1 F-adresa

Následek

Řízení interpretuje F-adresy v závislosti na právě aktivní G-funkci G93, G94, G95 jako

- dobu interpolace v sekundách (viz G93, strana 3-70)
- posuv v mm/min nebo Inch/min (viz G94, strana 3-71)
- posuv v mm/ot. (viz G95, strana 3-74)

Platí:

- U funkcí G94 a G95 účinkuje F modálně.



POZOR

Po spuštění, základním nastavení nebo resetu se případně změní poslední aktuální F-hodnota!

Po výše uvedených událostech působí F-hodnota definovaná v parametrech stroje 7060 00020 nebo 7060 00010 (standardní hodnota: F0).

Zde je také uloženo, zda má být potom aktivní G93, G94 nebo G95 (standardní hodnota: G94).

Proto se přesvědčte, že je před obráběním vždy naprogramován potřebný posuv!

Programování

Syntaxe:

F<Hodnota>

kde

<Hodnota> Je interpretována v závislosti na aktivní G-funkci jako doba interpolace, posuv nebo doba prodlevy.

Zvláštnosti a omezení:

- Naprogramovaný posuv dráhy se dá potlačit pomocí funkce zkušební posuv. Ta je řízena signálem rozhraní kanálu qCh_Test-Feed (zkušební posuv).

Vorschub/Drehzahl FA Omega

3.4.2 Rychlost asynchronních os

FA

Následek

Všechny pojezdové pohyby asynchronních os naprogramované ve větě FA neprobíhají v rychlém chodu, ale s naprogramovanou rychlostí.

**POZOR****Možnost poškození stroje v důsledku chybného naprogramování!****Zadaná rychlost působí pouze v aktuální FA-větě!****Jestliže naprogramujete v další větě asynchronní osy bez nového FA-slova, pojezdějí osy opět v rychlém chodu!**

Programování

Syntaxe:

FA<Hodnota>

kde

<Hodnota> Požadovaná rychlost.

Příklad:

N10 G1 G94 X200 Z300 F200

Posuv synchronních os:
200 mm/min

N11 UA400 VA140 FA250

Asynchronní osy UA a VA pojezdějí s
250mm/min.

N12 UA0 WA10

Asynchronní osy UA a WA pojezdějí
opět v rychlém chodu.

3.4.3 Omega-adresa (posuv)

Omega

Následek

Pokud se mají ve větě pohybovat výhradně osy, které jsou při vytváření posuvu skryté (viz "FeedAd", strana 4-41), dá se jejich **posuv** nastavit pomocí adresy "Omega".

Programování

Syntaxe:

Omega<Hodnota>

kde

<Hodnota> Požadovaný posuv.

Vorschub/Drehzahl S, SSPG


3.4.4 Programování počtu otáček vřetena

S, SSPG

Následek

Určuje při aktivní funkci G97 požadovaný počet otáček

- jednotlivého vřetena (S...), resp.
- kompletní skupiny vřeten (SSPG...).

 **Vřeteno dosahuje naprogramovaného počtu otáček, pouze když je vydán příkaz otáčení vřetena (např. pravotočivý chod: M3, levotočivý chod: M4).**

Programování

Syntaxe:

S<číslo>=<hodnota>

Programování počtu otáček pro jednotlivé vřeteno.

SSPG<Skupina>=<Hodnota>

Programování počtu otáček pro kompletní skupinu vřeten.

S<Hodnota>

Zkrácený zápis pro naprogramování počtu otáček 1. vřetena.

Funguje výhradně pro 1. vřeteno, pokud není pomocí MP 1040 00002 **přiřazeno žádné** skupině vřeten.

Jinak se tímto způsobem programuje kompletní skupina vřeten, ve které je 1. vřeteno obsaženo.

kde

<Číslo>

Číslo (index) vřetena.

Rozsah zadání:

1 až počet definovaných vřeten (definováno pomocí MP 1040 00001). Integer.

<Skupina>

Číslo skupiny vřeten.

Rozsah zadání: 1... 4

<Hodnota>

Požadovaný počet otáček vřetena (standardní jednotka: ot./min).

Zadaná hodnota: ≥ 0 .

Zvláštnosti a omezení:

- Při aktivní funkci G96 je S-slovo interpretováno jako řezná rychlost. Syntaxe viz strana 3-75.
- V souvislosti s funkcí "Seřízení vřetena" je S-slovo interpretováno jako úhel polohování. Syntaxe viz strana 3-91.
- Ve stejné větě smí být naprogramovány počty otáček několika vřeten, resp. skupin vřeten.
- Výsledný počet otáček je závislý na hodnotě Override vřetena.

Vorschub/Drehzahl

S, SSPG

- Výsledný počet otáček je podle potřeby omezen
 - funkcemi SMin, resp. SMax (viz strana 4-45)
 - mezními hodnotami převodových stupňů (MP 1040 00011, 1040 00012)
- Maximální povolený počet otáček svazku spojení vřeten (synchronně běžící vřetena) závisí na době NC-cyklu. Přitom platí: $S_{max} [U/min] = 14400 / (MP\ 9030\ 00001\ [ms])$
- Nastavený počet otáček platí tak dlouho, dokud není přepsán nově zadanou hodnotou počtu otáček pro stejná vřetena. Po spuštění řízení platí $S=0$.

Příklad:

N10 G97

Aktivovat programování počtu otáček.

N20 SSPG1=1000

Počet otáček všech vřeten 1. skupiny vřeten na 1000 ot./min.

:

N50 S1=2000 S2=60

Počet otáček 1. vřetena na 2000 ot./min.

:

Počet otáček 2. vřetena na 60 ot./min.

N80 S3=2000

Počet otáček 3. vřetena na 2000 ot./min.

:

N90 S1500

Počet otáček 1. vřetena na 1500 ot./min.

:

Je-li 1. vřeteno přiřazeno skupině vřeten, platí zadání počtu otáček pro celou skupinu vřeten.

Oprava nástroje D

3.5 Oprava nástroje

3.5.1 D-korektura

D

Následek

D-korektura vyvolává hodnoty oprav nástroje, které jsou uloženy v IndraMotion MTX v tabulkách XML D-korektur. Tabulka oprav přitom může obsahovat maximálně 99 datových záznamů.

Každý datový záznam obsahuje následující hodnoty oprav:

- 3 délky nástroje L1, L2, L3,
- poloměr břitu RAD,
- poloha břitu ORI.

D-korektura je stejně vhodná pro vrtací, frézovací a soustružnické nástroje i pro nástroje s úhlovou hlavou. Díky celkem 3 hodnotám posunutí L1, L2 a L3 je možné realizovat jak prostorově konstantní posunutí nástroje, tak i paralelní opravy délky maximálně pro 3 různé nástroje.

Tabulka D-korektur se aktivuje funkcí "DcTsel" (DCS) a pak se pomocí NC-příkazu "D" vybírá z max. 99 datových záznamů korekční věta.

Hodnoty oprav jsou účinné při aktivní opravě nástroje G47 (délka nástroje a poloha břitu), resp. při aktivní opravě dráhy frézy G41/G42 nebo G141/G142 (poloměr nástroje). Přitom se aditivně skládají s aktivně vybranými korekčními hodnotami externí opravy nástroje ED.

Platí:

- Zvolená sada nástrojů je modálně účinná. Nové naprogramování maže dosud zvolenou sadu nástrojů.
- D-korektura může být naprogramována ve větě spolu s jinými přípravnými funkcemi, pojezdovými pohyby a pomocnými funkcemi.
- Oprava nástroje se vypočítává, teprve když je aktivována odpovídající NC-funkce: G47, G41, G42, G141, G142.

Programování

Syntaxe:

D<Č. sady nástrojů>

Zvolit sadu oprav nástroje z aktivní tabulky D-korektur

D0

Zrušit volbu sady oprav nástroje, aniž by byla zvolena nová.

kde

<Č. sady nástroje>

Číslo korekčního datového záznamu.
Rozsah zadání: 1... 99.

Příklad:

N10 D7

Zvolit korekční hodnoty pro sadu nástrojů 7.

:

Oprava nástroje ED

3.5.2 ED-korektura

ED

Následek

Externí oprava nástroje (ED-korektura) vyvolává korekční hodnoty maximálně pro 16 nástrojových břitů. Hodnoty oprav mohou být popsány pomocí programového modulu MT_TCorr v SPS nebo pomocí obráběcího programu s příkazem CPL "DCT".

Každý ze 16 datových záznamů obsahuje následující hodnoty oprav:

- 3 délky nástroje L1, L2, L3,
- poloměr břitu RAD,
- poloha břitu ORI.

ED-korektura je stejně vhodná pro vrtací, frézovací a soustružnické nástroje i pro nástroje s úhlovou hlavou. Díky celkem 3 hodnotám posunutí L1, L2 a L3 je možné realizovat jak prostorově konstantní posunutí nástroje, tak i paralelní opravy délky maximálně pro 3 různé nástroje.

Pomocí NC-příkazu "ED" se vybírá z max. 16 datových záznamů sada oprav.

Hodnoty oprav jsou účinné při aktivní opravě nástroje G47 (délka nástroje a poloha břitu), resp. při aktivní opravě dráhy frézy G41/G42 nebo G141/G142 (poloměr nástroje). Přitom se aditivně skládají s aktivně vybranými korekčními hodnotami D-korektury.

Platí:

- Zvolený nástrojový břit je modálně účinný. Nové naprogramování maže dosud zvolený nástrojový břit.
- ED-korektura může být naprogramována ve větě spolu s jinými přípravnými funkcemi, pojezdovými pohyby a pomocnými funkcemi.
- Oprava nástroje se vypočítává, teprve když je aktivována odpovídající NC-funkce: G47, G41, G42, G141, G142.
- Zvolený nástrojový břit může být vydán na SPS-rozhraní příslušného kanálu: iCh_ActFunc1..24

Programování

Syntaxe:

ED<Č. nástrojového břitu>

Zvolit sadu oprav

ED0

Zrušit volbu sady oprav, aniž by byla zvolena nová.

kde

<Č. nástrojového břitu>

Číslo nástrojového břitu.
Rozsah zadání: 1... 16.

Příklad:

N10 ED7

Zvolit korekční hodnoty pro břit 7.

:

Oprava nástroje ED

Notizen:

NC-Funktionen

4 NC-funkce s jazykovou syntaxí

4.1 Přehled


Řízení je vybaveno různými NC-funkcemi. Přitom jsou zde kromě příkazů, které jsou stanoveny v DIN 66025, obsažena také podstatná rozšíření v oblasti G-kódů a doplňujících jazykových syntaktických prvků, které jsou popsány v této kapitole.

- **NC-funkce s jazykovou syntaxí:**

U NC-funkcí s jazykovou syntaxí je zpravidla možné odvodit význam funkce již z názvu (např. "Scale": Měřítka). Většinou existují následující rovnocenné varianty:

- Dlouhý tvar s libovolným zápisem velkých a malých písmen.
V této příručce je pro lepší přehlednost vždy první písmeno spojených dílčích slov velké. Příklad: "KvProg"
- Zkratka, skládající se ze 3 velkých písmen (ve výjimečných případech také ze 4 velkých písmen).
Příklad: Zkratka z KvProg: KVP
Příslušná zkratka se objevuje také v zobrazení procesových dat aktivních modálních NC-funkcí.

V následujících popisech syntaxe jsou uvedeny obě varianty.

 **Tabulkový přehled všech NC-funkcí naleznete v příloze od strany A-2.**
Kromě toho jsou všechny NC-funkce uvedeny v rejstříku od strany A-52.

Používané způsoby zápisu

Pro syntaxi NC-funkcí se v příručce používají následující způsoby zápisu:

Typ písma "Courier tučné" nebo "Courier":

Znakové řetězce v tomto typu písma musí být naprogramovány, jak jsou vytištěny.

Příklad: G0 (POL)

Úhlové závorky < >

označují symbol pro programovaný výraz, resp. parametr. Symbol je vytištěn kurzívou.

Příklad: <osa1>

Složené závorky { }

označují volitelný výraz, resp. parametr.

Takové syntaktické prvky mohou, ale nemusí být naprogramovány.

Příklad: G0({POL},{<Par1>})

Znak "|"

vzájemně odděluje možné, ale ne současně použitelné parametry (alternativní parametry).

Příklad: G0({POL},{NIPS|IPS1|IPS2|IPS3})

NC-Funktionen Area, ARA

4.2 Monitorování oblastí

Area, ARA

Následek

Definuje, aktivuje nebo deaktivuje až 10 obdélníkových dvojrozměrných mrtvých nebo pracovních oblastí s osově paralelními mezemi.

- **Mrtvá oblast**
Pojezdový pohyb ji nesmí křížit ani se jí dotknout (včetně jejích hranic).
- **Pracovní oblasti:**
Pojezdový pohyb je nesmí opustit (včetně jejich hranic).

 **Standardní data všech oblastí jsou definována pomocí skupiny parametrů stroje MP 8002.**

Programování: Aktivovat jednu jednotlivou nebo všechny monitorované oblasti najednou.

Syntaxe:

Area (<BNr>, <Sta>)

Zkratka: **ARA** (...)

kde

<BNr>	Číslo oblasti. Integer. Rozsah zadání: -1, 1...10. -1: Deaktivovat/aktivovat všechny oblasti.
<Sta>	Požadovaný stav monitorování. 0: Deaktivovat monitorování. 1: Aktivovat monitorování.

Programování: Definovat, aktivovat, deaktivovat jednu jednotlivou oblast monitorování.

Syntaxe:

Area (<BNr>, <Sta>{, <Mod>, {<P1>}, {<P2>}, {<D1>}, {<D2>}})


Zkratka: **ARA** (...)

kde

<BNr>	Číslo oblasti. Integer. Rozsah zadání: 1...10.
<Sta>	Požadovaný stav monitorování. 0: Deaktivovat monitorování oblasti <BNr>. 1: Aktivovat monitorování oblasti <BNr>.
<Mod>	0: Oblast <BNr> se nepoužívá. 1: Oblast <BNr> je mrtvá oblast. 2: Oblast <BNr> je pracovní oblast.

NC-Funktionen Area, ARA

<P1>	<p>Poloha v soustavě souřadnic stroje. Určuje hodnotu polohy středu oblasti vzhledem k první systémové ose zúčastněné na oblasti (definováno pomocí MP 8002 00001). Viz následující příklad. Programovací jednotka jako pro souřadnice os.</p>
<P2>	<p>Poloha v soustavě souřadnic stroje. Určuje hodnotu polohy středu oblasti vzhledem k druhé systémové ose zúčastněné na oblasti (definováno pomocí MP 8002 00002). Viz následující příklad. Programovací jednotka jako pro souřadnice os.</p>
<D1>	<p>Určuje délku oblasti vzhledem k první systémové ose zúčastněné na oblasti (definováno pomocí MP 8002 00001). Viz následující příklad. Programovací jednotka jako pro souřadnice os.</p>
<D2>	<p>Určuje délku oblasti vzhledem k druhé systémové ose zúčastněné na oblasti (definováno pomocí MP 8002 00002). Viz následující příklad. Programovací jednotka jako pro souřadnice os.</p>

 **Nenaprogramované hodnoty zůstávají zachovány, pokud již byly v průběhu programu zadány. Pokud ještě nebyly v průběhu programu zadány, používá řízení odpovídající hodnoty ze skupiny parametrů stroje 8002.**

Zvláštnosti a omezení:

- Všechny systémové osy, které jsou zúčastněny na jednotlivých oblastech, musí být definovány v MP 8002 00001 a MP 8002 00002.
- Aby bylo možné ovlivňovat oblasti pomocí funkce "Area", je zapotřebí odpovídající nastavení MP 8002 00032.
- Funkce podmiňuje referencované osy.
- Při aktivaci oblasti musí příslušné osy ležet v aktuálním kanálu.
- Ovlivňování oblasti pomocí funkce "Area" působí pouze v kanálu, ve kterém je funkce "Area" naprogramována.
Jsou-li osy oblasti předány jinému kanálu, má oblast v cílovém kanálu standardní hodnoty přednastavené ve skupině parametrů stroje 8002 a je neaktivní.
Hodnoty naprogramované případně funkcí "Area" ve zdrojovém kanálu se nepředávají.
- V krokovacím provozu: Osy, které pojíždějí ručně, nejsou monitorovány.
- V krokovacím provozu: z os, které definují mrtvou oblast, smí být krokována vždy pouze jedna osa.
- Porušení oblasti v krokovacím provozu generuje varování. Příslušná osa zůstane stát a může být krokována pouze v opačném směru.

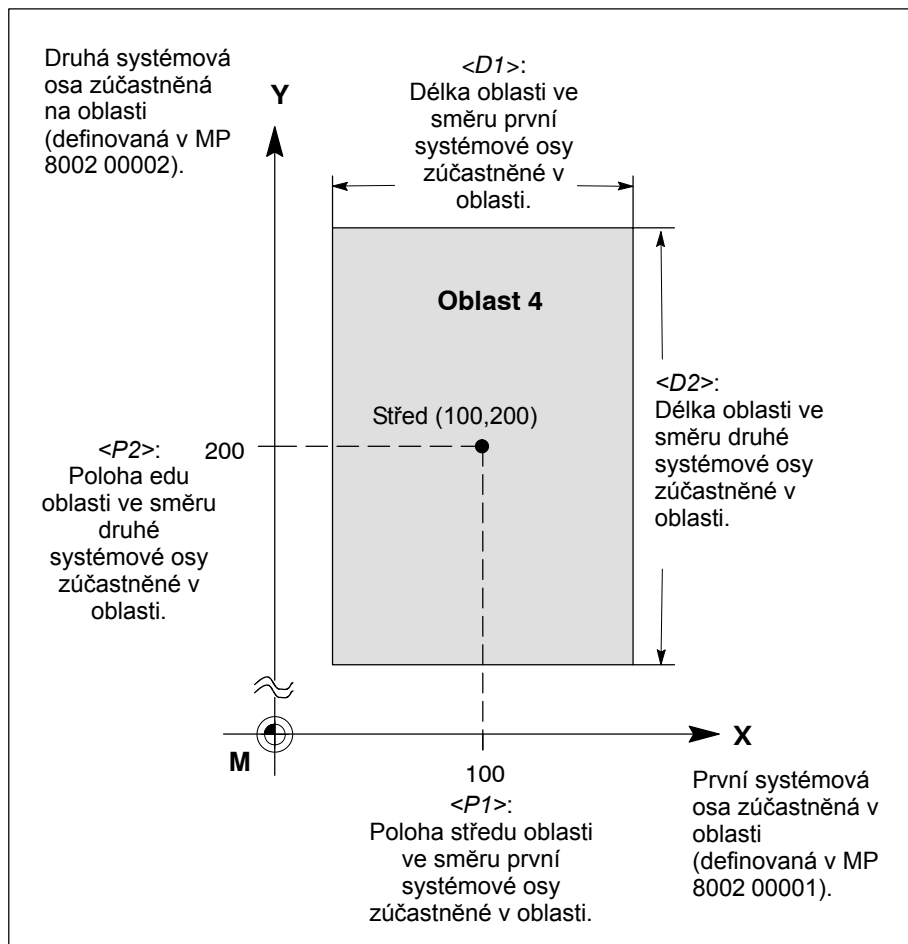
NC-Funktionen Area, ARA

Příklad:

N100 Area(4,0,,100,200)

Deaktivovat oblast 4 a nastavit střed oblasti na zadané souřadnice stroje (100,200).

Délky oblasti zůstanou beze změny.



NC-Funktionen ASPCLR

4.3 Asynchronní podprogramy: odhlášení

ASPCLR

Následek

Odhlašuje asynchronní podprogram v aktuálním kanálu.

Odhlášené podprogramy nemohou být aktivovány ani deaktivovány.

☞ **Chcete-li znovu přihlásit odhlášené asynchronní podprogramy, viz funkce ASPSET na straně 4-8.**

☞ **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete ve funkční příručce.**

Programování

Syntaxe:

ASPCLR(<Č. UP>)

kde

<Č. UP> Číslo podprogramu.
Rozsah zadání: 1...8. Integer.

Zvláštnosti a omezení:

- Příslušný podprogram musí být přihlášen v cílovém kanálu (viz ASPSET na straně 4-8).
- Pro dočasnou deaktivaci podprogramu viz funkce ASPDIS na straně 4-6.

NC-Funktionen ASPDIS ASPENA

4.4 Asynchronní podprogramy: vypnutí

ASPDIS

Následek

Deaktivuje asynchronní podprogram v aktuálním kanálu.

Deaktivovaný podprogram není vyvolán, když nastane předepsaná událost.

 **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete ve funkční příručce.**

Programování

Syntaxe:

ASPDIS (<Č. UP>)

kde

<Č. UP> Číslo podprogramu.

Rozsah zadání: 1...8. Integer.

Zvláštnosti a omezení:

- Příslušný podprogram musí být přihlášen v cílovém kanálu (viz ASP-SET na straně 4-8).
- Pro aktivaci deaktivovaného podprogramu viz funkce ASPENA na straně 4-6.

4.5 Asynchronní podprogramy: zapnutí

ASPENA

Následek

Aktivuje asynchronní podprogram v aktuálním kanálu.

Pouze aktivované podprogramy mohou být vyvolány, když nastane předepsaná událost.

 **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete ve funkční příručce.**

Programování

Syntaxe:

ASPENA (<Č. UP>)

kde

<Č. UP> Číslo podprogramu.

Rozsah zadání: 1...8. Integer.

Zvláštnosti a omezení:

- Příslušný podprogram musí být přihlášen v cílovém kanálu (viz ASP-SET na straně 4-8).
- Pro deaktivaci podprogramu viz funkce ASPDIS na straně 4-6.

NC-Funktionen ASPRTP

4.6 Asynchronní podprogramy: definování bodu opětovného najetí

ASPRTP

Následek

Definuje, zda má být řízení po skončení asynchronního podprogramu polohováno na

- počáteční bod
- koncový bod nebo
- bod přerušení

případně přerušené pojezdové věty.

Pokud v okamžiku přerušení nebyla aktivní žádná pojezdová věta, polohuje se řízení vždy na poslední aktivní souřadnice.

 **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete ve funkční příručce.**

Programování

Syntaxe:

ASPRTP (<Č. UP>, <bod>)

kde

<Č. UP>

Číslo podprogramu.

Rozsah zadání: -1; 1...8. Integer.

-1: Definovat požadovaný bod opětovného najetí pro všechny asynchronní podprogramy aktuálního kanálu.

<Bod>

Požadovaný bod opětovného najetí:

- 1: Počáteční bod
- 2: Koncový bod
- 3: Bod přerušení

Zvláštnosti a omezení:

- Nastavený požadovaný bod opětovného najetí se maže základním nastavením nebo funkcí M30.
- Mezitím provedené změny oprav v asynchronním podprogramu jsou při interním výpočtu požadovaného bodu opětovného najetí automaticky brány v úvahu.
- Pomocí funkce REPOSTP (viz strana 4-35) je možné definovaný bod opětovného najetí v asynchronním podprogramu dočasně skrýt.

NC-Funktionen ASPSET

4.7 Asynchronní podprogramy: přihlášení

ASPSET

Následek

Přihlašuje asynchronní podprogram v právě aktuálním kanálu a zapíná ho (zapnutí viz též funkce ASPENA na straně 4-6).

Je možné využívat pouze přihlášené a zapnuté podprogramy.

- ☞ **Chcete-li znovu odhlásit přihlášené asynchronní podprogramy, viz funkce ASPCLR na straně 4-5.**
- ☞ **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete ve funkční příručce.**

Programování

Syntaxe:

ASPSET (<Č. UP>, <Název UP>{, <Příznaky>})

kde

- <Č. UP> Číslo podprogramu.
Rozsah zadání: 1...8. Integer.
- <Název UP> Název podprogramu; se zadáním cesty nebo bez. Jestliže chybí cesta, hledá se soubor podle MP 3080 00001 (vyhledávací cesta pro podprogramy).
- <Příznaky> Spínače, s nimiž je možné ovlivňovat chování po vyvolání:
 - 00 Ani spojení ani automatické spuštění.
 - 10 Spojení, ale bez automatického spuštění.
 - 01 Žádné spojení, ale automatické spuštění.
 - 11 Spojení a automatické spuštění.

Příklad:

```

:
N30 ASPSET(1, ASUP1, 10)
:

```

Přihlášení programu ASUP1 jako prvního asynchronního podprogramu v aktivním kanálu. Po vyvolání (např. signálem rozhraní) bude v případě potřeby automaticky spojen a explicitně spuštěn s použitím NC-Start.

NC-Funktionen ASPSTA

4.8 Asynchronní podprogramy: spuštění programem ASPSTA

Následek

Vyvolává asynchronní podprogram s programovým řízením v libovolném kanálu.

 **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete ve funkční příručce.**

Programování

Syntaxe:

ASPSTA (<č. *UP*>{,<č. *kanálu*>})

kde

<č. *UP*> Číslo podprogramu.

Rozsah zadání: 1...8. Integer.

<č. *kanálu*> Cílový kanál, ve kterém má být vyvolán <č. *UP*>. Pokud není naprogramován, je <č. *UP*> vyvolán v aktuálním kanálu.

Zvláštnosti a omezení:

- Asynchronní podprogramy se nesmí vnořovat.
- Vyvolaný podprogram musí být přihlášen v cílovém kanálu (viz ASP-SET na straně 4-8).
- Vyvolaný podprogram nesmí být v cílovém kanálu vypnutý (viz ASP-DIS na straně 4-6, resp. ASPENA na straně 4-6).

NC-Funktionen AssLogName, ALN

4.9 Zadání logického názvu osy

AssLogName, ALN

Následek

Přiřazuje synchronní ose ve volajícím kanálu nový logický název. Starý logický název osy pak již nebude platný.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

AssLogName (<PAN> | <PAI> | <LAN>, <LANneu>
{, <PAN> | <PAI> | <LAN>, <LANneu>}...)

Zkratka: **ALN** (. .)

kde

<PAN>	Fyzický název osy. Určuje osu, která má být v aktuálním kanálu přejmenována.
<PAI>	Fyzický index osy. Účinky jako <PAN>.
<LAN>	Logický název osy. Účinky jako <PAN>.
<LANneu>	Nový logický název osy. Osa zadaná pomocí <PAN>, <PAI> nebo <LAN> obdrží v aktuálním kanálu logický název <LANneu>. <LANneu> musí být definováno v MP 7010 00010 (logické označení os) nebo MP 7010 00020 (volitelné označení os).

Zvláštnosti a omezení:

- Přejmenovávaná osa musí být zastavená. Pokud tomu tak není, generuje řízení chybové hlášení a přeruší program.
- Osové polohy ve stejné větě musí být vždy naprogramovány za výrazem AssLogName(...).

Příklad:

```

:
N030 ALN(YP,X,3,Y,B,Z)
:

```

Fyzická osa YP dostává logický název X, 3.
fyzická osa dostává logický název Y a lo-
gická osa B dostává logický název Z.
Následné naprogramování B vede k běhové
chybě.

NC-Funktionen ATCAL

4.10 Kalibrace osových kinematik: optimalizace parametrů ATCAL

Následek

Slouží v souvislosti s funkcí "Kalibrace osových kinematik" k optimalizaci délkových a úhlových parametrů vztahujících se k osovým kinematikám. Tato data jsou pro každý typ transformace os odlišná a pro každou transformaci os jsou obsažena v MP 1030 00140. Musí být zpravidla před optimalizací načtena funkcí ATGET (viz strana 4-14) a po optimalizaci znovu uložena pomocí funkce ATPUT (viz strana 4-15) do NC.

 **Podrobné informace o kalibraci osových kinematik naleznete ve funkční příručce.**

Programování

Syntaxe:

ATCAL (<Soubor>, <OptData>, <Maska>{, <Info>}{, <Anzlt>})

kde

<Soubor> Název kalibračního souboru; se zadáním cesty nebo bez. Jestliže chybí cesta, hledá se soubor podle MP 3080 00001 (vyhledávací cesta pro podprogramy). <Soubor> obsahuje data, která jsou zapotřebí pro optimalizaci. Příslušné informace naleznete ve funkční příručce.

<OptData> Permanentní, globální nebo lokální pole CPL. Typ: Double. Rozměr: min. 16.

Pole obsahuje po výpočtu sadu parametrů s optimalizovanými délkovými a úhlovými parametry.

Pořadí jednotlivých proměnných (index 1 až 16) přitom odpovídá indexu prvků MP 1030 00140. Hodnoty mají stejnou jednotku jako jednotlivé parametry v MP 1030 00140.

<Maska> Proměnná CPL. Typ: Integer.

Bitová maska, pomocí které se určí optimalizované jednotlivé parametry. Příklad:

	← Pole <OptData> před optimalizací →															
Index:	1	2	3	4	5	6	7	8	...	15	16					
Bitová maska: 1 = optimalizovat jednotlivé parametry	1	1	1	0	1	0	0	0	...	0	0					
Hodnota:	1	2	4	8	16	32	64	128	...							
požad. hodnota v <masce>:	1	2	4		16											
	1	+	2	+	4		+	16								= 23
	← Pole <OptData> po optimalizaci →															
Index:	1	2	3	4	5	6	7	8	...	15	16					
	■ : optimalizovaný jednotlivý parametr															

<Info> Permanentní, globální nebo lokální pole CPL. Typ: Double. Rozměr: min. 4.

NC-Funktionen ATCAL

Obsahuje po optimalizaci následující data:

- <Info>[1] Hodnota funkce jakosti (kvadratická odchylka) před optimalizací. Jednotka: mm.
- <Info>[2] Hodnota funkce jakosti (kvadratická odchylka) po optimalizaci. Jednotka: mm.
- <Info>[3] Max. odchylka před optimalizací (v mm).
- <Info>[4] Max. odchylka po optimalizaci (v mm).
- <AnzIt> Maximální počet iteračních kroků pro optimalizaci.
Pokud není zadáno nebo je zadáno "-1", skončí iterace teprve tehdy, až odchylka mezi dvěma po sobě následujícími vypočítanými sadami parametrů klesne pod dostatečně malou interní prahovou hodnotu NC (přestane existovat nezanedbatelná odchylka).
Je-li zadáno "1", jedná se o případ lineárního vyrovnávacího počtu.

Příklad:

01 DIM PAR! (16)

Dimenzovat lokální pole CPL s 16 poli typu Double (pro optimalizované délkové a úhlové parametry).

02 DIM GA! (4)

Dimenzovat lokální pole CPL s 4 poli typu Double (pro jakost a odchylku).

03 MASK%=2+4+32

Optimalizace jednotlivých parametrů s indexy pole 2, 3 a 6.

N4 ATCAL (CL.TXT, [PAR!],
[MASK%], [GA!], -1)

Spustit optimalizaci.

NC-Funktionen ATFW

4.11 Kalibrace osových kinematik: přepočítání parametrů ATFW

Následek

Slouží v souvislosti s funkcí "Kalibrace osových kinematik" k přepočítávání reálných poloh os na souřadnice soustavy souřadnic, vytvořené transformací os. Tato procedura se nazývá také "transformace dopředu".

 **Podrobné informace o kalibraci osových kinematik naleznete ve funkční příručce.**

Programování

Syntaxe:

ATFW (<Koord>, <AxKoord>{, <ParData>})

kde

<Koord> Permanentní, globální nebo lokální pole CPL.
Typ: Double. Rozměr: nejméně tak velký jako počet kanálových souřadnic v soustavě souřadnic, vytvořené transformací os.

Jednotlivé proměnné pole obsahují po přepočítání výsledné kanálové souřadnice v transformované soustavě souřadnic.

<AxKoord> Permanentní, globální nebo lokální pole CPL.
Typ: Double. Rozměr: nejméně tak velký jako počet os v aktuálním kanálu.

Proměnné pole musí obsahovat reálné osové polohy všech kanálových os zúčastněných na transformaci os.

<ParData> Permanentní, globální nebo lokální pole CPL.
Typ: Double. Rozměr: min. 16.
Pole musí obsahovat sadu parametrů se všemi délkovými a úhlovými parametry pro transformaci os.
Pořadí jednotlivých proměnných (index 1 až 16) odpovídá indexu prvků MP 1030 00140. Hodnoty musí mít stejnou jednotku jako příslušné jednotlivé parametry v MP 1030 00140.

Pokud není naprogramováno, použije se sada parametrů právě aktivní transformace os.

Pokud jsou v řízení současně aktivní dvě transformace os (AT1, AT2; viz funkce "Coord" od strany 4-30), použijí se data z AT2.

Zvláštnosti a omezení:

- Jestliže nebylo naprogramováno <ParData> a není aktivní žádná transformace os, objeví se běhová chyba.

NC-Funktionen ATGET

4.12 Kalibrace osových kinematik: načtení parametrů z NC ATGET

Následek

Slouží v souvislosti s funkcí "Kalibrace osových kinematik" ke čtení délkových a úhlových parametrů vztahujících se k osovým kinematikám. Tato data jsou pro každý typ transformace os odlišná a pro každou transformaci os jsou obsažena v MP 1030 00140. Musí být zpravidla načtena před optimalizací (viz funkce ATCAL na straně 4-11) a po optimalizaci znovu uložena pomocí funkce ATPUT (viz strana 4-15) do NC.

 **Podrobné informace o kalibraci osových kinematik naleznete ve funkční příručce.**

Programování

Syntaxe:

ATGET (<ParData>{, <AxTrafoNr>})

kde

<ParData> Permanentní, globální nebo lokální pole CPL.
Typ: Double. Rozměr: min. 16.

Pole obsahuje po provedení funkce sadu parametrů se všemi délkovými a úhlovými parametry transformace os definované pomocí <ATrafNr>.

Pořadí jednotlivých proměnných (index 1 až 16) přitom odpovídá indexu prvků MP 1030 00140. Hodnoty mají stejnou jednotku jako jednotlivé parametry v MP 1030 00140.

<AxTrafoNr> Číslo načítané transformace os.

Pokud není naprogramováno, načtou se data právě aktivní transformace os.

Pokud jsou v řízení současně aktivní dvě transformace os (AT1, AT2; viz funkce "Coord" od strany 4-30), použijí se data z AT2.

Zvláštnosti a omezení:

- Jestliže nebylo naprogramováno <ATrafNr> a není aktivní žádná transformace os, vyvolá to běhovou chybu.

NC-Funktionen ATPUT

4.13 Kalibrace osových kinematik: zápis parametrů do NC ATPUT

Následek

Slouží v souvislosti s funkcí "Kalibrace osových kinematik" k přepsání všech délkových a úhlových parametrů transformace os vztahujících se k osovým kinematikám.

Tato data jsou pro každý typ transformace os odlišná a pro každou transformaci os jsou obsažena v MP 1030 00140. Musí být zpravidla před optimalizací (viz funkce ATCAL, strana 4-11) načtena (viz funkce ATGET, strana 4-14) a po optimalizaci znovu přenesena pomocí funkce ATPUT do NC.

 **Podrobné informace o kalibraci osových kinematik naleznete ve funkční příručce.**

Programování

Syntaxe:

ATPUT (<ParData>{, <AxTrafoNr>})

kde

<ParData> Permanentní, globální nebo lokální pole CPL.
Typ: Double. Rozměr: min. 16.

Pole musí obsahovat sadu parametrů se všemi délkovými a úhlovými parametry transformace os definované pomocí <AxTrafoNr>.

Pořadí jednotlivých proměnných (index 1 až 16) odpovídá indexu prvků MP 1030 00140. Hodnoty musí mít stejnou jednotku jako příslušné jednotlivé parametry v MP 1030 00140.

<AxTrafoNr> Číslo požadované transformace os, jejíž sada parametrů má být zapsána do MP 1030 00140.

Pokud není naprogramováno, budou přepsána data aktivní transformace os.

Pokud jsou v řízení současně aktivní dvě transformace os (AT1, AT2; viz funkce "Coord" od strany 4-30), budou přitom změněna data AT2.

Zvláštnosti a omezení:

- Jestliže nebylo naprogramováno <ATrafNr> a není aktivní žádná transformace os, vyvolá to běhovou chybu.
- Pro provádění funkce ATPUT je zapotřebí uživatelská úroveň NC, která dovoluje přepisovat parametry stroje.
- Zapsaná sada parametrů je aktivní teprve
 - po základním nastavení systému a
 - novém naprogramování COORD (<ATrafNr>).

NC-Funktionen AUXFUNC

4.14 Provedení aktivních pomocných funkcí všech skupin AUXFUNC

Následek

Provádí právě aktivní pomocné funkce, kanálově specifické a přesahující kanály, všech existujících skupin pomocných funkcí.

Funkce je důležitá ve spojení s procházením vět (např. po přerušení obrábění) pro obnovení stavů všech pomocných funkcí na určité pozici dílčího programu.

Pokud je např. dílčí program během obrábění přerušen, je pomocí procházení vět mimo jiné možné opět obnovit obrábění od věty, která byla zpracovávána v okamžiku přerušení.

Přitom se sice dílčí program spustí od začátku s procházením vět, ale obrábění na stroji začne až od první definované programové věty.

Protože řízení během procházení vět neinterpoluje ani nevydává naprogramované pomocné funkce, je na konci procházení vět nezbytné obnovit stavy všech pomocných funkcí.

 **Podrobné informace o používání a parametrizaci pomocných funkcí naleznete ve funkční příručce.**

Programování

Syntaxe:

AUXFUNC

Příklad:

N100 T102	Vybrat nástroj T102.
N110 M6	Vyměnit nástroj T102.
:	
N150 M3	Zapnout 1. vřeteno / skupinu vřeten. Pravotočivý chod.
:	

... v N160 přerušení programu...

- aktivní byly T102 a M3.
- nástroj T102 je vyměněný. Poloha nástroje se nezmění.

Program bude následně znovu vybrán a poběží s procházením vět až po N150 včetně bez pojezdových pohybů. Naprogramované pomocné funkce jsou přitom sice aktivovány, ale ne vydávány.

Než bude zpracována věta N160, je nyní třeba pomocí AUXFUNC provést právě aktivní pomocné funkce, kanálově specifické i přesahující kanály, všech existujících skupin pomocných funkcí.

V příkladu bude mít AUXFUNC tyto účinky:

- Aktivace čísla nástroje T102 (nástroj je ještě vložený).
- Zapnutí 1. vřetena / skupiny vřeten s pravotočivým chodem.

NC-Funktionen AxAcc, AAC

4.15 Změna maximálního zrychlení os Dočasné uložení max. zrychlení os

AxAcc, AAC
AxAccSave, AAS

Následek

- **AxAcc:**
mění dočasně horní meze maximálního zrychlení os.
Funkce přitom překrývá hodnoty maximálního zrychlení os z parametrů stroje naprogramovanými hodnotami.
- **AxAccSave:**
ukládá dočasně aktuální maximální hodnoty zrychlení všech os v interní paměti.
Při každé volbě programu se tato interní paměť inicializuje s hodnotami z parametrů stroje.

Programování

Syntaxe:

AxAccSave

Dočasně uložit všechna aktuální maximální zrychlení os.

Zkratka: AAS

AxAcc(<hodnoty>)

Změnit maximální zrychlení os.

AxAcc nebo

AxAcc(1)

Znovu aktivovat zrychlení os, která byla předtím uložena pomocí "AxAccSave".

AxAcc() nebo

AxAcc(0)

Znovu aktivovat zrychlení os z parametrů stroje.

Zkratka: AAC(..)

kde

<Hodnoty> Název osy a hodnota zrychlení.

Údaje pro několik os jsou odděleny čárkou.

Podle aktivní měrné jednotky (G71/G70) interpretuje řízení naprogramovaná data jako "1000 Inch/s²" nebo "m/s²".

Zvláštnosti a omezení:

- Pokud v systému existuje pod stejným názvem fyzická i logická osa aktivního kanálu, je vždy ovlivňováno zrychlení logické osy.

NC-Funktionen AxAcc, AAC

Příklad:

Výchozí situace: osy X až Z jsou v parametrech stroje obsazeny hodnotou 8.0 m/s².

:		
N40	AAC (X1.0, Z2.1)	Max. zrychlení pro osu X: 1.0 m/s ² , Max. zrychlení pro osu Z: 2.1 m/s ² .
:		
:		Max. zrychlení osy Y zůstává beze změny (8.0 m/s ²).
:		
N80	AxAccSave	Dočasně uložit všechny právě aktivní maximální hodnoty zrychlení os.
:		
N90	AxAcc ()	Znovu aktivovat hodnoty z parametrů stroje.
:		max. zrychlení pro osu X: 8.0 m/s ²
:		max. zrychlení pro osu Y: 8.0 m/s ²
:		max. zrychlení pro osu Z: 8.0 m/s ²
:		
N150	AxAcc (Y5)	Max. zrychlení pro osu Y: 5.0 m/s ² .
:		Max. zrychlení pro osy X a Z zůstávají beze změny.
:		
N200	AxAcc	Znovu aktivovat hodnoty uložené pomocí "AxAccSave":
:		max. zrychlení pro osu X: 1.0 m/s ²
:		max. zrychlení pro osu Y: 8.0 m/s ²
:		max. zrychlení pro osu Z: 2.1 m/s ²

NC-Funktionen AxCouple, AXC

4.16 Spojení os

AxCouple, AXC

Následek

Pomocí spojení os se vytváří určitý vztah mezi pohybem synchronní "hlavní osy" (řídící osa) a jednou nebo více (max. 7) synchronními "podřízenými osami" (následné osy).

Jestliže pojíždí hlavní osa, pohybují se všechny podřízené osy automaticky v závislosti na svém definovaném vztahu k hlavní ose. Hlavní osa a všechny zúčastněné podřízené osy se proto nazývají také "svazek os".

Možné vztahy mezi hlavní a podřízenou osou:

- Požadované polohy hlavní osy jsou na základě **konstantního offsetu** transformovány na odpovídající požadované polohy podřízené osy (viz vzorec 1).
Podřízená osa proto může být posunuta oproti hlavní ose o libovolnou, vždy konstantní dráhu v kladném nebo záporném směru pojezdu.
- Požadované polohy hlavní osy jsou na základě **konstantního faktoru spojení** transformovány na odpovídající požadované polohy podřízené osy (viz vzorec 1).
Podřízená osa se tak může pohybovat vzhledem k hlavní ose v určitém poměru.
- Požadované hodnoty hlavní osy jsou na základě tabulky (spojení) transformovány na **libovolné** odpovídající požadované polohy podřízené osy. Přitom jsou v tabulce uloženy páry opěrných bodů, které pro polohu hlavní osy a případně potřebné posunutí hlavní osy vždy udávají odpovídající polohu podřízené osy (viz vzorec 2).
Polohy mezi jednotlivými opěrnými body může určovat řízení lineární interpolací nebo pomocí kubické spline funkce.

Vzorec 1 (pro lineární spojení):

$$p_s = p_m * k + o$$

Offset (posunutí)
Faktor spojení

Vzorec 2 (pro libovolné spojení):

$$p_s = f(p_m - p_m^o) * k + o$$

Offset (posunutí)
Faktor spojení
Posunutí hlavní osy
Funkce spojení (jako tabulka spojení)


Všechny uvedené vztahy se dají libovolně kombinovat.

Tímto způsobem mohou být velmi jednoduše realizovány paralelní osy (např. pro paralelně uspořádané obráběcí stoly) nebo také elektronické převodovky (1 otáčka hlavní osy dává např. 10 otáček podřízené osy).

NC-Funktionen AxCouple, AXC

Platí:

- všechny osy svazku os musí ležet ve stejném kanálu.
- je dovoleno několik svazků os v jednom kanálu.

 **Podrobné informace o funkci "Spojení os" naleznete v příručce "Funkční popis".**
Zde najdete také informace o tabulce spojení potřebné při "libovolných spojeníh".

Programování

Syntaxe:

$$\text{AxCouple}(\langle M \rangle \langle \text{varianta} \rangle, \langle S \rangle \{ \langle SO \rangle \}, \{ \langle SF \rangle \}, \{ \langle MO \rangle \}, \langle \text{Tab} \rangle \} \{ \dots \})$$

AxCouple() nebo
AxCouple(0)

Vymazat všechny svazky os v aktuálním kanálu.

Zkratka: **AXC (. .)**

kde

$\langle M \rangle$	Logická adresa hlavní osy.
$\langle \text{Varianta} \rangle$	0 Znovu vytvořit svazek os 1 Změnit svazek os (přidat nové osy nebo změnit označení spojení) -1 Vymazat podřízenou osu nebo úplně zrušit svazek
$\langle S \rangle$	Logická adresa podřízené osy.
$\langle SO \rangle$	Offset posunutí podřízené osy. Pokud není naprogramováno, platí $\langle SO \rangle = 0$.
$\langle SF \rangle$	Faktor spojení podřízené osy. Pokud není naprogramováno, platí $\langle SF \rangle = 1$.
$\langle MO \rangle$	Posunutí hlavní osy. Používá se jen u "libovolného spojení" (viz vzorec 2 výše). Pokud není naprogramováno, platí $\langle MO \rangle = 0$.
$\langle \text{Tab} \rangle$	Název tabulky spojení příslušné podřízené osy. Používá se jen u "libovolného spojení" (viz vzorec 2 výše).

Příklady:

N100 AXC(Z0,A(4,2),B(2,1))	Vytvořit svazek os.
:	Z: hlavní osa,
:	A/B: podřízené osy.
:	Obě podřízené osy běží s "lineárním" spojením.
N200 AXC(X0,A(4,2),B(,0.5,,T_B))	Vytvořit svazek os.
:	X: hlavní osa,
:	A/B: podřízené osy.
	Podřízená osa A běží s "lineárním", podřízená osa B s "libovolným" spojením.

NC-Funktionen AxCouple, AXC

:	Podřízené osy A a B budou vymazány ze svazku os Z.
N100 AXC(Z-1,A(),B())	
:	
N200 AXC(Z-1)	Bude vymazán celý svazek os Z.
:	

**POZOR**

Funkce spouští pojezdový pohyb všech podřízených os naprogramovaných v této větě!

Ty přitom vždy pojíždějí do svého specifického bodu spojení, který je dán polohou hlavní osy a spojovacím vztahem.

Zvláštnosti a omezení:

- Všechny osy zúčastněné ve svazku os musí být přinejmenším během spojení os synchronní.
Asynchronní nebo Hirthovy osy nejsou dovoleny.
- Programování pojezdového pohybu pro podřízené osy není dovoleno a vede k chybovému hlášení.
- Podřízená osa nemůže být zároveň hlavní osou v jiném svazku os.
- Konec programu nevolňuje automaticky existující spojení os.
- Je-li hlavní osou modulo osa, musí být podřízenou osou při lineárním spojovacím vztahu rovněž modulo osa.
- Aby bylo možné najíždění os do referenčních bodů, je zapotřebí otevření spojení os.
- Povolný rozsah pojezdu hlavní osy může být připojenými podřízenými osami zredukován (např. když podřízená osa dosahuje svých koncových oblastí rychleji než hlavní osa, nebo je rozsah pojezdu podřízené osy menší než rozsah pojezdu hlavní osy).
- Pokud jsou zakryty koncové vypínače pro hlavní, resp. podřízenou osu, není pro celý svazek os účinný žádný koncový vypínač.
- Maximální dynamika "nejslabší" osy určuje maximální dynamiku celého svazku os.
- Zablokování os u aktivního svazku os je zakázáno.
- Osy spojené ve zkušebním provozu musí být před vypnutím zkušebního provozu opět odpojeny.

NC-Funktionen AxisToSpindle, ATS

4.17 Vypnutí provozu osy C pro vřetena

AxisToSpindle, ATS

Následek

Přepíná vřeteno, které se nachází v provozu osy C (viz strana 4-55), na vřetenový provoz.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

AxisToSpindle (<PAN> | <PAI>{, <PAN> | <PAI>}...)

Zkratka: **ATS** (..)

kde

<PAN>	Fyzický název osy. Určuje osu, která má být přepnuta z provozu osy C zpět na provoz vřetena.
<PAI>	Fyzický index osy. Účinky jako <PAN>.

Zvláštnosti a omezení:

- Zadaná osa musí být zastavená a nesmí patřit do žádné skupiny os. Pokud tomu tak není, generuje řízení chybové hlášení a přeruší program.
- Osové polohy ve stejné větě musí být vždy naprogramovány za výrazem AxisToSpindle(...).

Příklad:

:		
N030	ATS(CH)	Fyzická osa CH (tzn. vřeteno, které má v osovém provozu název CH) je přepnuta na vřetenový provoz.
:		

NC-Funktionen BcsCorr, BCR

4.18 Umístění: Oprava polohy obrobku

BcsCorr, BCR

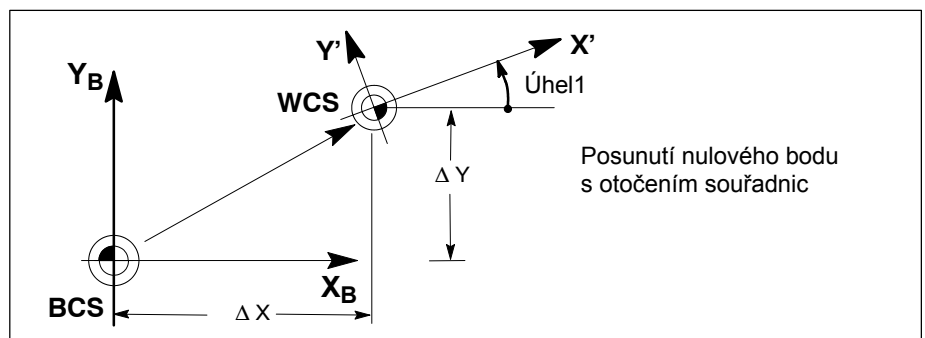
Následek

Slouží jako oprava upnutí.

Umístění "opravy polohy obrobku" může libovolně posouvat a orientovat soustavu souřadnic obrobku v prostoru. Oprava polohy obrobku působí v příslušném kanálu na souřadnice s významy "x", "y" a "z".

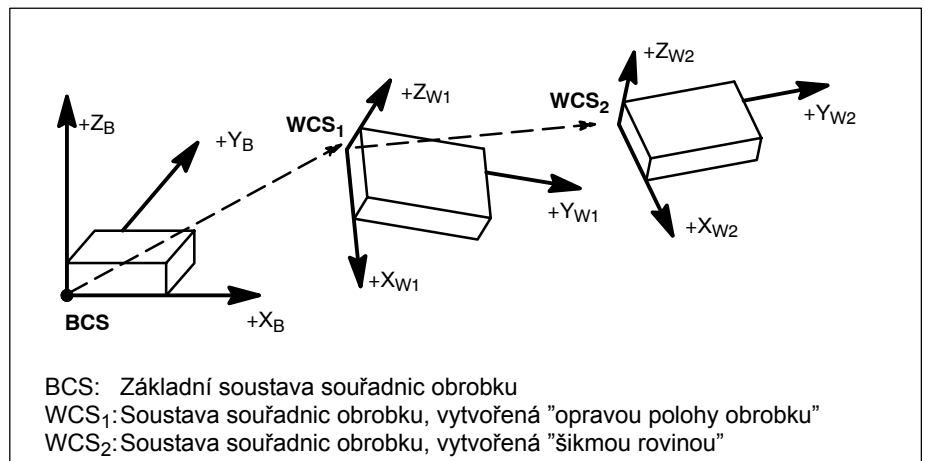
Protože je poloha obrobku po upnutí vyměřena a brána v úvahu opravou polohy obrobku, zmenšují se výrazně nároky na seřizování.

Protože pro orientaci existují 3 stupně volnosti, dá se každá orientace zobrazit 3 po sobě následujícími základními otočeními. Na následujícím obrázku je znázorněna jednoduchost díky pouhému základnímu otočení kolem osy Z.



☞ Funkčně identické je umístění "šikmé roviny" (viz strana 3-82). Zde naleznete také znázornění všech 3 základních otočení.

Další umístění (např. šikmá rovina) účinkují aditivně. Ve "výpočetním řetězci" je oprava polohy obrobku ještě před šikmou rovinou:



NC-Funktionen BcsCorr, BCR

Programování

Syntaxe:

$$\text{BcsCorr}(\{\langle X_W\text{-offset} \rangle\}, \{\langle Y_W\text{-offset} \rangle\}, \{\langle Z_W\text{-offset} \rangle\}, \{\langle \text{úhel1} \rangle\}, \{\langle \text{úhel2} \rangle\}, \{\langle \text{úhel3} \rangle\})$$
Zapnutí opravy polohy
obrobku.BcsCorr() nebo
BcsCorr(0)Vypnutí opravy polohy
obrobku.

Zkratka: BCR(..)

kde

- $\langle X_W\text{-offset} \rangle$: Hodnota posunutí ve směru hlavní souřadnice
 $\langle Y_W\text{-offset} \rangle$: Hodnota posunutí ve směru vedlejší souřadnice
 $\langle Z_W\text{-offset} \rangle$: Hodnota posunutí ve směru normálové souřadnice
 $\langle \text{úhel1} \rangle$: Úhel otočení kolem souřadnice Z.
 Rozsah hodnot: $0 \leq \langle \text{úhel1} \rangle < 360$ stupňů
 $\langle \text{úhel2} \rangle$: Úhel otočení kolem nové souřadnice Y'.
 Rozsah hodnot: $0 \leq \langle \text{úhel2} \rangle < 180$ stupňů
 $\langle \text{úhel3} \rangle$: Úhel otočení kolem nové souřadnice Z".
 Rozsah hodnot: $0 \leq \langle \text{úhel3} \rangle < 360$ stupňů

Příklad:

N70 G40	Vypnutí opravy dráhy frézy.
N80 BCR(50,300,10,1.23)	Zapnutí opravy polohy obrobku.
:	Nulový bod nové soustavy souřadnic
:	obrobku leží v BCS na X50 Y300 a Z10.
:	Souřadnicové osy X a Y nové soustavy
:	souřadnic obrobku jsou vzhledem k BCS
:	otočeny o 1,23 stupně proti směru
:	hodinových ručiček kolem nové
:	souřadnicové osy Z.
:	
N200 BCR()	Vypnutí opravy polohy obrobku.

Zvláštnosti a omezení:

- Aktivace a deaktivace opravy polohy obrobku přerušuje náhled vět a nesmí být proto naprogramována při aktivní opravě dráhy frézy (G41/G42, viz strana 3-39).
- Zapnutý stav a chování při základním nastavení jsou zkonfigurovány v parametrech stroje 7060 00010 a 7060 00020.

NC-Funktionen ChLength, CHL ChSection, CHS

4.19 Programování faset

ChLength, CHL
ChSection, CHS

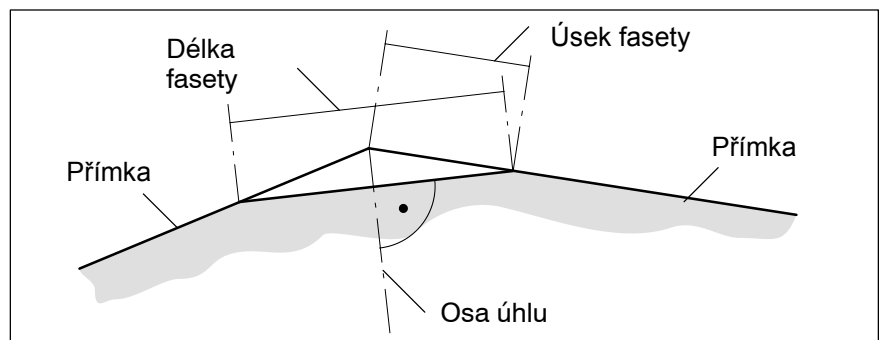
Následek

Funkce "programování faset" vkládá mezi dvě po sobě následující NC-věty typu přímka nebo kruh přechodovou fází, jejíž délka může být zadána buď jako absolutní **délka fasety** nebo jako **délka úseku fasety**. Faseta je generována v aktivní pracovní rovině.

Jsou možné následující fasetové přechody:

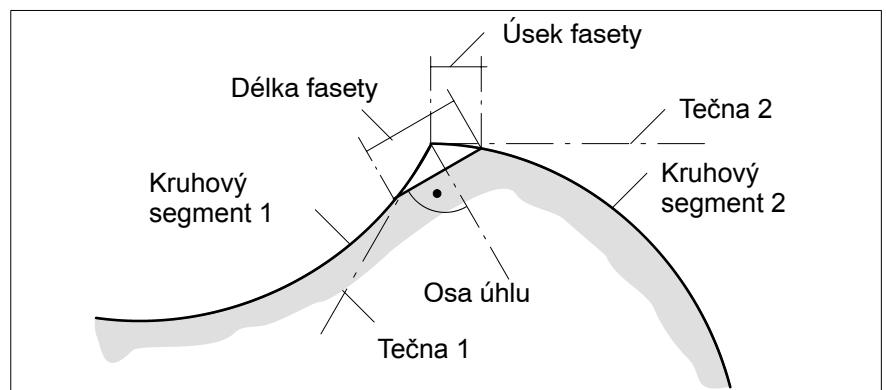
- **Faseta mezi dvěma stýkajícími se přímkami**

Faseta probíhá v pravém úhlu k ose úhlu mezi sousedními segmenty dráhy. Délka fasety se automaticky opravuje (zmenšuje), když neexistuje žádný průsečík se sousedními naprogramovanými segmenty dráhy.



- **Faseta mezi dvěma stýkajícími se kruhovými segmenty**

Při konturových přechodech s kruhovými segmenty se rozměrové údaje faset vztahují k příslušné koncové, resp. počáteční tečně segmentů dráhy zúčastněných na konturovém přechodu. Skutečně existující délka fasety je mimo jiné značně závislá na poloměrech zúčastněných kruhů a liší se tedy více nebo méně od naprogramovaných rozměrů.



NC-Funktionen ChLength, CHL ChSection, CHS

Programování**Syntaxe:**ChLength(<*délka fasety*>)Zapnutí "programování faset".
Zadat požadovanou <*délku fasety*> v jednotkách mm (G71), resp. Inch (G70).ChLength() nebo
ChLength(0)

Vypnutí "programování faset".

Zkratka: CHL(..)

ChSection(<*úsek fasety*>)Zapnutí "programování faset".
Zadat požadovaný <*úsek fasety*> v jednotkách mm (G71), resp. Inch (G70).ChSection() nebo
ChSection(0)

Vypnutí "programování faset".

Zkratka: CHS(..)

Zvláštnosti a omezení:

- Funkce "ChLength", "ChSection", "RoundEps" a "Rounding" působí modálně a vzájemně se ruší.
- Faseta se vztahuje vždy pouze k aktivní pracovní rovině (G17, G18, G19, G20). Jsou-li na pohybu zúčastněny další osy, nemají žádný vliv na fasety.
Protože souřadnice naprogramovaných pojezdových vět os v pracovní rovině jsou ovlivňovány fasetami, ale hodnoty pro osy mimo pracovní rovinu zůstávají beze změny, může se např. u přímek změnit směr v prostoru.
- Funkce působí pouze v oblasti "zpracování" pod "následnou větou", "jednotlivou větou" nebo "jednotlivým krokem".
Protože se "programová věta" chová jako ruční zadání, není zde programování faset účinné.
- Chování při zapnutí/vypnutí stejně jako při základním nastavení se určuje pomocí parametrů Initstring v parametrech stroje 7060 00010 a 7060 00020.

NC-Funktionen Collision, CLN

4.20 Monitorování kolizí

Collision, CLN

Následek

Nabízí možnost, monitorování kolizí opravy dráhy frézy G41/G42

- zapnout nebo vypnout a
- přizpůsobit ohledně oblasti náhledu a chování v případě kolize aplikace.

Platí:

- Monitorování kolizí funguje pouze při aktivní opravě dráhy frézy G41/G42, i když má poloměr opravy hodnotu "0".
- Jestliže aktuální hodnota opravy poloměru nepřípouští obrábění jednotlivých konturových prvků, pokouší se řízení upravit příslušný průběh dráhy tak, aby kontura nebyla porušena.
- Monitorování kolizí bere v průběhu kontury v úvahu pouze souřadnice aktivní pracovní roviny. Jsou-li kolize programově technicky eliminovány, např. změnou hloubky přísuvu nástroje, vede to v aktuální oblasti náhledu přesto k aktivaci monitorování kolizí. V takových případech je možné dočasně vypnout monitorování kolizí v příslušném úseku obrábění.

 **Podrobné informace o funkci "Monitorování kolizí" naleznete v příručce "Funkční popis".**

 **Doporučujeme zadat požadované chování monitorování kolizí při zapnutí do MP 7060 00010 / MP 7060 00020.**

Programování

Syntaxe:

Collision(1)

Zapnutí monitorování kolizí.
Zachovat aktuální chování v případě kolize.
Pokud chování dosud ještě nebylo naprogramováno nebo zadáno v MP 7060 00010/MP 7060 00020, odpovídá chování podle Collision(CollErr 0).

Collision() nebo
Collision(0)

Vypnutí monitorování kolizí.

Collision(DEF)

Určit přednastavení pro oblast náhledu monitorování kolizí na 2 věty.

Collision(CollErr <typ>)

Zapnutí monitorování kolizí a stanovení chování v případě kolize.

NC-Funktionen Collision, CLN

Collision(DLA <v[ty]>)	Určení přednastavení pro oblast náhledu monitorování kolizí. Je účinné při dalším naprogramování G41/G42.
Collision(LA <v[ty]>)	Dočasná změna oblasti náhledu až do dalšího naprogramování G41/G42.

Zkratka: CLN(...)

kde

<Typ>	Určuje chování řízení při zjištění kolize: 0: Není vydána běhová chyba ani varování. Obrábění není přerušeno, ale konturové smyčky jsou vynechány. 1: Je vydána běhová chyba. Obrábění se přeruší. 2: Je vydáno varování. Obrábění se nepřeruší.
<Věty>	Určuje velikost (počet vět) oblasti náhledu. Hodnota Integer. Doporučená oblast náhledu: 1 až 10 vět.

Zvláštnosti a omezení:

- Přednastavení pro oblast náhledu jsou 2 věty.
- Maximální možná oblast náhledu je závislá na parametrech stroje 7060 00110 až 7060 00130.
- Aby při zapnuté opravě dráhy frézy bylo možné najet dozadu na konturu, aniž by se aktivovalo monitorování kolizí, musí se od bodu obratu změnit aktivní směr opravy (při aktivní funkci G41 se programuje G42, při aktivní funkci G42 se programuje G41). Řízení přitom automaticky ukončuje oblast náhledu monitorování kolizí ve větě G41 nebo G42 a bezprostředně poté znovu spouští náhled.

NC-Funktionen Collision, CLN

Příklad:

:	
N100 CLN(DLA 5)	Přednastavení pro oblast náhledu monitorování kolizí od další funkce G41/G42: 5 vět.
:	
:	
N110 G41 D10	Oprava dráhy frézy vlevo od obrobku.
N120 X10	Pojezd dopředu.
N130 X20	
N140 X30	
N150 G42	Přepnutí na opravu dráhy frézy vpravo od obrobku. Náhled monitorování kolizí je ukončen od větě 150 a poté znovu spuštěn.
:	
:	
:	
N160 X20	Pojezd dozadu.
N170 X10	
N180 X0	
N190 G40	Vypnutí opravy dráhy frézy. Monitorování kolizí je nyní neaktivní, ale ne deaktivované!
:	

NC-Funktionen Coord, CRD

4.21 Volba transformace os

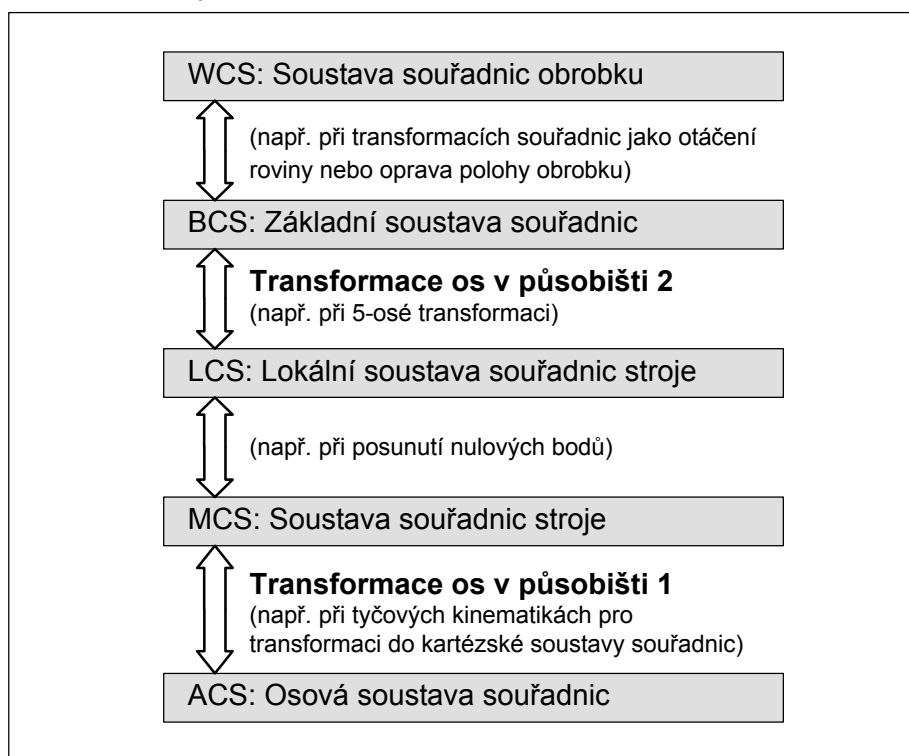
Coord, CRD

Následek

Aktivuje nebo deaktivuje transformace os, které jsou zkonfigurovány ve skupině parametrů stroje 1030.

- ☞ **Funkce je zapotřebí např. v souvislosti s programováním prostorových souřadnic (viz příručka " Funkční popis") nebo podle okolností při kalibraci osových kinematik (viz strany 4-14, 4-15 a 4-13).**

V řízení smí být transformace os účinná maximálně ve 2 působíštích:



- ☞ **Pro každou dostupnou transformaci os je již interně stanoveno, ve kterém působíšti tato transformace funguje.**

NC-Funktionen Coord, CRD

Programování

Syntaxe:

Coord(<AxTrafoNr>)

Zapnutí transformace os

Coord() nebo

Coord(0)

Vypnutí transformace os v
působišti 2

Coord(0{,<působíště>})

Vypnutí transformace os ve
vybraném působištiZkratka: **CRD (...)**

kde


<AxTrafoNr>

Číslo transformace os.
Rozsah zadání: 1... 20. Integer.

<Působíště>

Působíště vypínané transformace os.
Zadání: 1 nebo 2**Zvláštnosti a omezení:**

- Je možné přímo přepínat mezi různými transformacemi os v působišti 2. Není zapotřebí předchozí deaktivace.
- Transformace os v působišti 1 se smí zapnout, pouze když v působišti 2 není aktivní žádná transformace os.

 **Přiřazení mezi typem a číslem transformace os je uloženo v MP 1030 00110.**
Informace o dostupných typech transformací os naleznete v příručce "Funkční popis".

NC-Funktionen CoupleSplineTab, CST

4.22 Spline tabulka spojení os

CoupleSplineTab, CST

Následek

Při spojení os je funkce spojení uložena ve formě párů opěrných bodů v tabulce spojení. Pro výpočet poloh podřízené osy mezi opěrnými body vytváří přípravy věty spline tabulku.


Spline tabulka se vytváří při překladu syntaxe spojení a ukládá jako soubor v adresáři tabulek spojení.

Spline tabulky se vytvářejí automaticky. Pomocí CoupleSplineTab(...) je možné vytvořit spline tabulku také podle definice:

- CST(STAB(<TabName>,1)) vyžaduje explicitně nové vytvoření spline tabulky.
- CST(STAB(<TabName>,0)) vyžaduje nové vytvoření, když neexistuje žádná spline tabulka nebo je starší než tabulka spojení.

Pomocí CoupleSplineTab(...) je také možné vytvořit spline tabulku, aniž by musel existovat svazek os (např. při ručním zadávání).

Název spline tabulky se vytváří z názvu právě aktivní tabulky spojení připojením přípony ".s", např. z názvu tabulky spojení **curve.fct** se stane název spline tabulky **curve.fct.s**.

 **Tabulka spojení se hledá v aktuální vyhledávací cestě. Vyhledávací cesta je nastavena v parametru stroje 3080 00001. Adresářem tabulek spojení je standardně /usr/lnk. Ten může být volně definován pomocí parametru stroje 3080 00004.**

Programování

Syntaxe:

CoupleSplineTab(STAB(<TabName>{,<1|0>}))

Vytvořit spline tabulku

Zkratka: **CST (...)**

kde

<TabName> Název tabulky spojení, která se hledá v aktuální vyhledávací cestě a vytváří pro spline tabulku.

<1|0> Podle volby:
 0: Spline tabulka se vytváří, pouze když neexistuje nebo je starší než tabulka spojení. (standardní)
 1: Spline tabulka se vytváří nově.

Příklad:

CST(STAB(curve.fct)) Vytváří v případě potřeby spline tabulku /<adresář spojení>/curve.fct.s

CST(STAB(curve.fct,1)) Vytváří nezávisle na datu nebo existenci spline tabulku /<adresář spojení>/curve.fct.s

NC-Funktionen DcTsel, DCS

4.23 Aktivace tabulek D-korektur

DcTsel, DCS

Následek

Aktivuje tabulku D-korektur (tabulka geometrických oprav, tabulka GEO). Tabulky D-korektur jsou uloženy jako soubory XML v systému souborů řízení.

Programování

Syntaxe:

DcTsel ({<cesta>}<název souboru>)

Zkratka: **DCS** (. .)

kde

<Cesta> Volitelné zadání cesty k adresáři, v němž je uložen <název souboru>.

Bez zadání probíhá hledání v cestě "/database".

Pokud tam <název souboru> neexistuje, použije řízení vyhledávací cestu pro podprogramy a hledá <název souboru> také v jiných adresářích.

<Název souboru> Název souboru tabulky D-korektur včetně přípony souboru.

Tabulky se standardním názvem (DC<číslo>.dct) mohou být aktivovány přímo pomocí čísla, např.

DcTsel(7) aktivuje tabulku DC7.dct.

 **Pokyny pro vytváření a úpravy tabulek D-korektur naleznete v návodu k obsluze řízení!**

Příklad:

:		
N030	DCS(geotab.dct)	Hledá tabulku D-korektur "geotab.dct" nejprve v adresáři "/database", a když tam neexistuje, ve vyhledávací cestě pro podprogramy. První nalezená tabulka D-korektur s názvem "geotab.dct" je aktivována.
:		
N130	DCS(/mnt/ge.dct)	Hledá a aktivuje tabulku D-korektur "ge.dct" v adresáři "/mnt". Jestliže tam není nalezena, objeví se chybové hlášení.
:		

NC-Funktionen DefAxis, DAX

4.24 Převzetí nastavení os z MP

DefAxis, DAX

Následek

Aktivuje kanálově přesahující standardní konfiguraci os podle MP 1003 00002.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování


Syntaxe:

DefAxis

Zkratka: **DAX**

Zvláštnosti a omezení:

- Vede k běhové chybě, pokud zúčastněná osa nebyla uvolněna.

 **Protože se na předávání os často podílí více kanálů, doporučujeme zadat funkci v MP 7060 00020 za klíčovým slovem "#SysRes". Tímto způsobem je možné vždy obnovit standardní konfiguraci os v souvislosti se "základním nastavením systému".**

NC-Funktionen DiaProg, DIA RadProg, RAD

4.25 Programování průměru Programování poloměru

DiaProg, DIA
RadProg, RAD

Následek

Souřadnicové údaje pro rovinné osy (na soustruzích; většinou souřadnice X) mohou být alternativně interpretovány jako průměr nebo poloměr. Tak je možné převzít příslušné rozměrové údaje bez přepočítávání přímo do dílčího programu.

Je-li zapnuto "programování průměru", nastavuje se na osových zobrazeních rovinné osy pro polohu obrobku, zbývající dráhu, koncovou polohu a programovou hodnotu symbol průměru.

Poloha stroje, skutečná hodnota osy a doběh se vždy zobrazují jako hodnoty poloměru.



POZOR

Možná nesprávná interpretace rozměrových údajů!

"DIA" působí pouze na zadané/zkonfigurované souřadnice průměru. Programování průměru nemá žádný vliv na parametry kruhové interpolace I, J, K.

Dejte pozor, aby byly vždy naprogramovány pouze vhodné rozměrové údaje.

Programování

Syntaxe:

DIA{({<Coord1>{,...,<Coord8>}})}

Zapnutí programování průměru až pro 8 lineárních os, resp. souřadnic. Pro všechny nezadané souřadnice je programování průměru vypnuté.

kde

<Coordi>

Max. 8 lineárních os, resp. souřadnic (i = 1...8), jejichž dráhy mají být vyhodnoceny jako údaje průměru.

DIA

Obnovuje poslední stav před RAD. Po spuštění řízení platí standardní hodnoty parametrů stroje.

DIA()

Aktivují se standardní hodnoty. Pokud v parametrech stroje nejsou definovány žádné souřadnice průměru, je osou průměru osa klasifikace X. Jestliže rovněž neexistuje, hlásí řízení běhovou chybu.

RAD

Zapnutí programování poloměru pro všechny souřadnice.

Zvláštnosti a omezení:

- Funkce působí modálně a vzájemně se ruší.

NC-Funktionen DiaProg, DIA RadProg, RAD

- Je-li souřadnice průměru zahrnuta do transformace os, vypne NC pro tuto osu programování průměru. Při zapnuté transformaci os však mohou být přepínány souřadnice na programování průměru.
- Pro provozní režimy "ruční zadávání" a "zpracování" platí:
U rozměrových údajů pro souřadnice středu kruhu, délky nástrojů a posunutí nulových bodů jsou souřadnice rovinné osy vždy interpretovány jako hodnota poloměru.
- Pro provozní režimy "provoz s ručním kolečkem" a "krokovací provoz" platí:
Je možné přepínat mezi programováním průměru a poloměru pomocí signálu osového rozhraní "Rozměr kroku průměru" (qAx_Jog-Dia).

Příklad:

N10 DIA ()	Zapnout zkonfigurované souřadnice průměru.
N20 DIA (Y1, W2)	Uvedené souřadnice Y1 a W1 se stanou souřadnicemi průměru, všechny nenaprogramované souřadnice souřadnicemi poloměru.
N30 RAD	Programování průměru se pro všechny souřadnice vypne.
N40 DIA :	Souřadnice Y1 a W1 se stanou souřadnicemi průměru (poslední stav před RAD).

NC-Funktionen DistCtrl, DCR

4.26 Regulace výšky pro digitalizaci

DistCtrl, DCR

Následek

Udržuje při digitalizaci konstantní vzdálenost mezi snímaným povrchem a měřicím zařízením (např. laser). Tím je zaručeno, že nebude opuštěna dostupná pracovní oblast měřicího zařízení.

 **Podrobný popis funkce naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

DistCtrl(1) nebo DistCtrl
Spouští regulaci výšky a přebírá aktuální vzdálenost mezi měřicím zařízením a povrchem jako vztažnou hodnotu.

Platí konfigurační data definovaná pomocí parametrů stroje 7050 007xx.

DistCtrl() nebo DistCtrl(0)
Ukončuje regulaci výšky, přebírá aktuální hodnotu opravy a zastavuje pohyb os.

Je-li regulace výšky naprogramována ve stejné větě s pojezdovým pohybem, vypíná se až po provedení pohybu.

DistCtrl(<Fkt>)
Překrývá některá funkčně specifická konfigurační data v parametrech stroje.

Zkratka: **DCR (...)**

kde

<Fkt> **DcAxis (<osa>, <oprava>)**

Překrývá MP 7050 00702.

Zkratka: **DCA (...)**

<Osa> Název nebo číslo kanálové osy, která má být výškově regulována.

<Oprava> Směr pohybu, do kterého mají být započítány hodnoty opravy:

+1 nebo 1: v kladném směru pohybu

-1: v záporném směru pohybu

DcFilter (<čas>)

Překrývá MP 7050 00730.

Zkratka: **DCF (...)**

<Čas> Parametrizace filtru pro vyhlazení hodnot senzoru.

0: Vypnutý filtr

>0: Zapnutý filtr; doba vyhlazení v ms

NC-Funktionen DistCtrl, DCR

DcLimit ({<Geschw>},{<Beschl>})

Překrývá MP 7050 00740, resp. MP 7050 00741.

Zkratka: **DCL**(...)<Geschw> Maximální rychlost změny hodnoty opravy.
Překrývá MP 7050 00740.

Zadaná hodnota v závislosti na aktivní měrné soustavě (G71,G70) v jednotkách mm/min, resp. Inch/min.

<Beschl> Maximální strmost (zrychlení) hodnoty opravy.
Překrývá MP 7050 00741.Zadaná hodnota v závislosti na aktivní měrné soustavě (G71,G70) v jednotkách m/s², resp. 1000 Inch/s².**DcMon** ({<kolize>},{<otvor>})

Překrývá MP 7050 00750, resp. MP 7050 00751.

Zkratka: **DCM**(...)

<Kolize> Pásmo tolerance pro detekci kolize.

Překrývá MP 7050 00750.

Zadaná hodnota v závislosti na aktivní měrné soustavě (G71,G70) v jednotkách mm, resp. Inch.

0: Vypnutá detekce kolizí.

<Otvor> Pásmo tolerance pro identifikaci otvorů.

Překrývá MP 7050 00751.

Zadaná hodnota v závislosti na aktivní měrné soustavě (G71,G70) v jednotkách mm, resp. Inch.

0: Vypnutá identifikace otvorů.

DcBreak Přerušuje regulaci výšky. Aktuální hodnota opravy zůstává aktivní.

Zkratka: **DCB**

DcCont Obnovuje opět regulaci výšky přerušenou předtím pomocí DCB. NC co možná nejrychleji reguluje odchylku od vztažné hodnoty.

Zkratka: **DCC**

NC-Funktionen EndPosCouple, EPC

4.27 Spojení koncových poloh

EndPosCouple, EPC

Následek

Spojuje dvě aktuálně synchronní souřadnice v kanálu podle následujícího vztahu:

- Δ podřízená souřadnice = faktor spojení * Δ hlavní souřadnice.

Při výpočtu výsledné podřízené souřadnice se polohové údaje hlavní souřadnice vždy automaticky převádějí na inkrementální dráhy.

Pro každou NC-větu, ve které je naprogramován polohový údaj hlavní souřadnice, vypočítává řízení podle výše uvedeného vztahu požadovanou koncovou polohu podřízené souřadnice.

Podřízení souřadnice se přitom chová - vzhledem k ostatním NC-funkcím - tak, jako by byla její dráha pojezdu zapsána v dílčím programu s použitím lokálního programování relativních rozměrů (IC). To např. znamená, že aktuální posuv ovlivňuje **výsledný posuvový pohyb**.

Programování

Syntaxe:

EndPosCouple(<M-Koord>, <S-Koord>, <Faktor>)

Zapnutí spojení koncových poloh.

EndPosCouple() nebo EndPosCouple(0)

Vypnutí spojení koncových poloh.

Zkratka: **EPC (...)**

kde

<M-Koord> Název hlavní souřadnice.

<S-Koord> Název podřízené souřadnice.

<Faktor> Poměr podřízené dráhy pojezdu k hlavní dráze pojezdu.

Zvláštnosti a omezení:

- Spojování koncových poloh není možné v souvislosti s nekonečnými souřadnicemi (souřadnice, jejichž hodnoty jsou při překročení mezí rozsahu automaticky transformovány zpět do definovaného rozsahu hodnot).
- Při aktivovaném spojení koncových poloh není dovoleno programování polárních souřadnic.
- Hlavní a podřízená souřadnice nesmí být při aktivním spojení koncových poloh naprogramovány společně ve stejné NC-větě. Výhradní naprogramování podřízené souřadnice v NC-větě je ovšem dovoleno.
- Hlavní ani podřízená souřadnice nesmí při aktivním spojení koncových poloh opustit kanál. Změna osy nebo změna aktivní transformace os nejsou dovoleny.
- I když je pro hlavní souřadnici zapnuto programování průměru (viz strana 4-35), používá se pro spojení odpovídající hodnota poloměru.

NC-Funktionen EndPosCouple, EPC

Příklad:

N10 G18 G0 Z0

Aktivní rovina: z,x. Pomocí rychlého chodu
na z=0.

N20 G1 F1000 Z3 X1

Polohování na P1.

N30 EPC(Z,X,1)

Zapnout spojení koncových poloh.

:

Hlavní souřadnice: Z; podřízená

:

souřadnice: X

Podřízená dráha pojezdu = hlavní dráha
pojezdu.

N40 Z4

Polohování na P2.

N50 X1

Polohování na P3.

N60 Z5

Polohování na P4.

N70 X3

Polohování na P5.

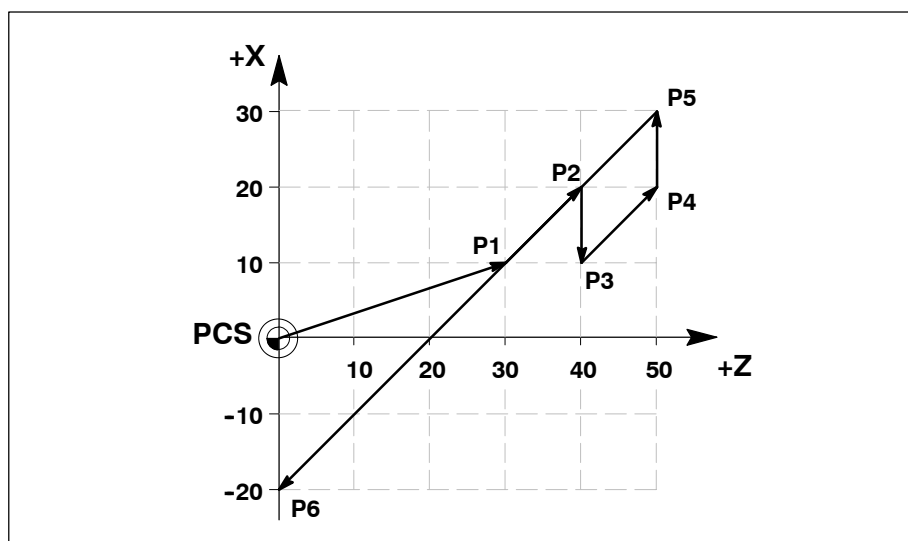
N80 Z0

Polohování na P6.

N90 EPC()

Vypnutí spojení koncových poloh.

:



NC-Funktionen FeedAd, FAD

4.28 Vytváření posuvu: skrytí os

FeedAd, FAD

Následek

Vyjímá všechny osy, které jsou definovány v MP 1003 00020, společně z vytváření posuvu.

Vyjmuté osy se pak pohybují synchronně.

Následkem toho se může zvětšit reálný posuv dráhy oproti naprogramované F-hodnotě.

Programování

Syntaxe:

FeedAd(1) nebo
FeedAd

Vyjmout osy z vytváření posuvu

FeedAd(0) nebo
FeedAd()

Brát v úvahu osy při vytváření posuvu

Zkratka: **FAD** (. .)

Zvláštnosti a omezení:

- Pokud se mají ve větě pohybovat výhradně osy, které jsou při vytváření posuvu skryté, dá se jejich posuv místo adresy "F" nastavit pomocí adresy "Omega" (viz strana 3-97).

Příklad: (v MP 1003 00020 je zapsána osa Y)

:	
N100 G94 G0 X0 Y0	Polohování na P(0;0) v rychlém chodu.
N110 FeedAd()	
N120 X100 Y100 F100	Polohování na P(100;100).
:	Napogr. posuv dráhy: 100 mm/min
:	Reálný posuv dráhy: 100 mm/min
N150 FeedAd	
N160 X200 Y200 F100	Polohování na P(200;200).
:	Napogr. posuv dráhy: 100 mm/min
:	Reálný posuv dráhy: 141,42 mm/min


NC-Funktionen FeedForward, FFW

4.29 Servo řízení

FeedForward, FFW

Následek

Funkce zkracuje doběh daný systémem vhodnou opravou hodnot interpolátoru zadaných v pohonu. Také je možné docílit vyšší věrnosti kontur nebo "provozu bez vlečné chyby".

 **Funkce "servo řízení" je implementována výrobcem v pohonu a aktivuje se, resp. deaktivuje pouze na základě správné syntaxe příkazu z dílčího programu.**

Přesný popis funkce "servo řízení" naleznete v dokumentaci pohonu.

 **Možnost aktivovat servo řízení musí být povolena parametrem stroje 1003 00009 pro příslušné osy.**

Programování

Syntaxe:

FeedForward(1)nebo
FeedForward

Aktivovat servo řízení pro všechny osy povolené podle MP 1003 00009 (příslušné pohony se přepnou na vedlejší provozní režim 1).

FeedForward(<Adr><hodnota>,...)

Aktivovat/deaktivovat servo řízení pro naprogramované osy (účinek závisí na <hodnotě>).

FeedForward() nebo
FeedForward(0)

Deaktivovat servo řízení pro všechny osy.
(Všechny příslušné osy se přepnou na hlavní provozní režim.)

Zkratka: FFW(..)

kde

<Adr>: Fyzická nebo logická adresa osy.

<Hodnota>: Informace o dráze pro <Adr>.

0: Deaktivovat servo řízení.
nerovno 0: Aktivovat servo řízení.

Příklad:

N10 FFW

Aktivovat servo řízení pro všechny osy kanálu povolené podle MP 1003 00009.

:

N50 FFW(Z0)

Deaktivovat servo řízení pro Z (přepnout Z na hlavní provozní režim).

:

N90 FFW()

Deaktivovat servo řízení všech os kanálu (přepnout na hlavní provozní režim).

:

:

Zvláštnosti a omezení:

- Při deaktivaci servo řízení se všechny osy kanálu přepnou na hlavní provozní režim.

NC-Funktionen FeedForward, FFW

- Parametrizace funkce servo řízení v pohonu je možná pouze zapsáním příslušných parametrů pohonu. Jako funkce pro zapisování parametrů SERCOS pomocí dílčího programu je k dispozici "WritelD" (viz strana 4-76).

NC-Funktionen FlyMeas, FME

4.30 Průběžné měření

FlyMeas, FME

Následek

Funkce "průběžné měření" slouží k doprovodnému měření během obrábění. Pojezdový pohyb se při zapnutí měřicího dotyku nepřerušuje.

Naměřená poloha je sdělena pohonem do NC. Může být zjištěna příkazem CPL PPOS.

Programování

Syntaxe:

FlyMeas(MpiAxis<*i*>)<Souřadnice os> Spustit měřicí cyklus.

Zkratka: FME(..)

kde

<*i*>: Index fyzické osy, se kterou se má měřit.
<Souřadnice os>: Poloha, na kterou se má během měřicí procedury najet.

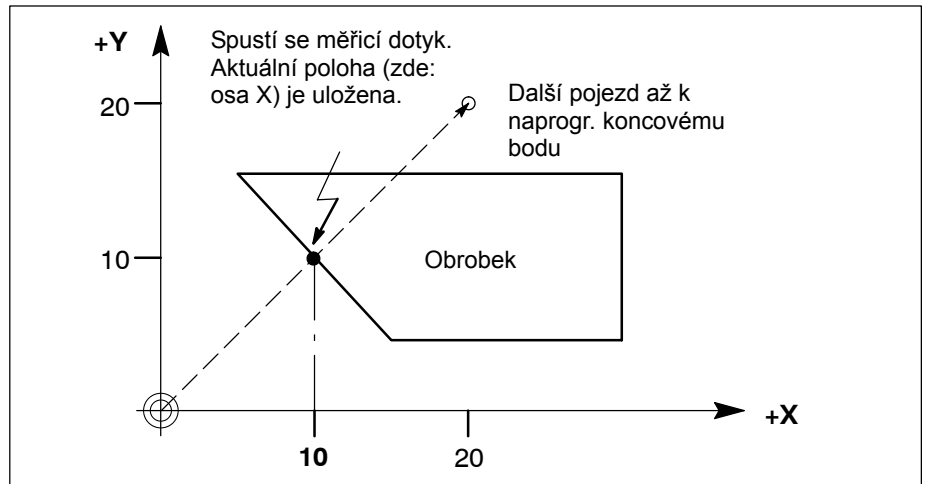
Zvláštnosti a omezení:

- Před prvním měřením je nutné inicializovat měřicí dotyk pomocí funkce "InitMeas".
- Funkce pracuje po větách.
- Funkce může být naprogramována spolu se všemi typy interpolace a pracuje paralelně s aktivní interpolací.
- Jestliže se měřicí dotyk nezapne, čeká NC na konec věty tak dlouho, dokud nedojde k měření.
- Když mají být informace měřicího dotyku dále zpracovány v následujících programových větách, je třeba věnovat pozornost náhledu vět. V případě potřeby je nutné naprogramovat funkci WAIT nebo omezit náhled příkazem BlkNmb.
- Poloha měření se dá zjišťovat pomocí příkazu PPOS.
- Funkce může být používána ve spojení s lineárními nekonečnými osami, pokud mají naprogramované polohy kladné znaménko. Pojezd s měřicím dotykem dozadu (naprogramování záporných poloh) nedává jednoznačné hodnoty!

NC-Funktionen FlyMeas, FME

Příklad:

```
:  
N100 IME(MpiAxis 1)           Inicializace logiky měřicího do-  
:                               tyku fyzické osy 1 (zde: osa X).  
  
N110 G0 X0 Y0  
N120 FME(MpiAxis 1) G1 X10 Y10  Spuštění měřicího cyklu a polo-  
:                               hování v posuvu na X10,Y10.
```



NC-Funktionen FsProbe, FSP

4.31 Měření na pevném dorazu

FsProbe, FSP

Následek

Zatímco řízení pojíždí se všemi naprogramovanými synchronními osami za použití lineární interpolace v zadaném posuvu do naprogramovaného koncového bodu, je u vybrané osy monitorován aktuální točivý moment.

Jestliže během pohybu překročí točivý moment této osy zkonfigurovanou mezní hodnotu, vede to v řízení k následujícím krokům:

- Nastavení signálu osového rozhraní "Pevný doraz dosažen"
- Uložení skutečné polohy
- Zabrzdění posuvového pohybu na $v=0$ s maximálním dovoleným zrychlením
- Vymazání zbývající dráhy
- Vymazání "FsProbe" ("FsProbe" pracuje po větách).

Pokud na konci dráhy ještě nebyl dosažen "pevný doraz" (překročení zadaného prahového točivého momentu), generuje řízení chybové hlášení.

 **Používejte "FsProbe" jen v souvislosti s programem CPL pro vyhodnocení.**

Programování

Syntaxe:

<Souřadnice> *<posuv>*

Zapnutí "měření na pevném dorazu"
pro osu *<i>* a naprogramované souřadnice s lineárně interpolovaným pojezdem.

FsProbeFsProbe(MfsAxis*<i>*)
(MfsAxis
(*<i>*,*<Prahová hodnota>*))
<Souřadnice> *<posuv>*

Zapnutí "měření na pevném dorazu"
s prahovým točivým momentem pro osu *<i>* a naprogramované souřadnice s lineárně interpolovaným pojezdem.

Zkratka: FSP(..)

kde

<i>

Index fyzické osy, se kterou se má měřit.

<Prahová hodnota> Prahový točivý moment.

Zadaná hodnota: v % maximálního točivého momentu.

Pokud není naprogramována *<prahová hodnota>*, působí parametr stroje 1003 00031 (mezní hodnota točivého momentu pevného dorazu).

<Souřadnice>

Poloha, na kterou se má během měřicí procedury najet.

NC-Funktionen FsProbe, FSP

<Posuv>: Požadovaný posuv dráhy.
Je omezen parametry MP 1005 00030 (maximální posuv pro najetí na pevný doraz) a MP 1005 00002 (maximální rychlost osy a rychlého chodu).

Příklad:

```

N100 FSP(MfsAxis(1,30)) X100 F500
110 IF SD(9)=0 THEN
120 XPOS=PPOS(1)
:
:
N130 (MSG, KONTAKT)
140 GOTO N180
150 ENDIF
N160 (MSG, ŽÁDNÝ KONTAKT)
N170 M0
N180 ...

```

Aktivovat "měření na pevném dorazu" pro první fyzickou osu a najet do polohy X100 s F500. Jako prahový ročivý moment nastavit 30% maximálního momentu.
Dotaz, jestli byl překročen prahový točivý moment.
Uložit polohu ve spínacím momentu 1. osy (osa X) do proměnné XPOS.
Zastavení programu

Zvláštnosti a omezení:

- Ve větě "FsProbe" nejsou povoleny následující funkce:
 - G75 (měřicí dotyk),
 - InitMeas/FlyMeas (průběžné měření),
 - RedTorque (redukce momentu),
 - FsMove/FsReset/FsTorque (najíetí na pevný doraz).
- Funkce "FsProbe" musí být naprogramována společně nejméně s jednou souřadnicí. Její hodnota představuje maximální hloubku hledání, do které musí být nejpozději dosažen "pevný doraz".
- "FsProbe" zastavuje implicitně přípravu následujících vět. Proto není nutné naprogramovat funkci "WAIT".
- Vyhodnocení, zda byl dosažen "pevný doraz", pokračování v programu (po chybě), bezpečnostní monitorování, generování chybových hlášení atd. musí být realizováno s použitím programu CPL.
- Pomocí SD(9) lze zjistit, jestli byl dosažen pevný doraz.
- Pomocí PPOS je možné zjistit polohu pevného dorazu.

NC-Funktionen FsMove, FSM FsTorque, FST FsReset, FSR

4.32 Najetí na pevný doraz

**FsMove, FSM
FsTorque, FST
FsReset, FSR** **Funkce najetí na pevný doraz je použitelná i pro asynchronní osy!**

Následek

Celková funkce "njetí na pevný doraz" obsahuje následující dílčí funkce:

1. **"Redukce momentu pevného dorazu":** **FsTorque, FST**
Nastavení točivého momentu, který smí pohon po zapnutí funkce "njetí na pevný doraz" maximálně generovat.
Není-li použita funkce "FsTorque", působí MP 1003 00031.

2. **"Njetí na pevný doraz":** **FsMove, FSM**
Spuštění pohybu ve směru pevného dorazu s ohledem na maximální dovolený točivý moment.

Dosažení max. povoleného točivého momentu (viz bod 1.) v průběhu tohoto pohybu spouští v řízení následující řetězec kroků:

- Vydání signálu osového rozhraní "Pevný doraz dosažen"
- Zabrzdění posuvového pohybu na $v=0$ s maximálním dovoleným zrychlením.
- Nastavení požadované polohy na:
skutečná poloha + 0,1 mm (resp. skutečná poloha + 0,1 stupně).
- Monitorování polohy osy s ohledem na:
poloha pevného dorazu + MP 1003 00032 ("monitorovací okno pevného dorazu v mm, resp. stupních").
- Udržování specifikovaného točivého momentu na příslušném pohonu.
- Další zpracovávání dílčího programu.


Pokud do naprogramovaného konce dráhy nebyl dosažen pevný doraz (dosažení zadaného prahového točivého momentu; viz "FsTorque"), generuje řízení chybové hlášení.

Funkce "FsMove" zůstává podle věty "FsMove" účinná a vypíná se pouze příkazem "FsReset".

3. **"Zrušení pevného dorazu":** **FsReset, FSR**
Uvolnění os a případné odjetí od pevného dorazu.
 - Pokud jsou ve větě "FsReset" naprogramovány synchronní, resp. asynchronní osy, pojíždí řízení se všemi osami v příslušném zadaném posuvu do naprogramovaných koncových bodů. Pro pohyb při odjíždění platí maximální povolený točivý moment jednotlivých os.

NC-Funktionen FsTorque, FSTFsMove, FSM FsReset, FSR

- Nejsou-li ve větě "FsReset" naprogramovány žádné osy, uvolní se výhradně všechny synchronní osy. Asynchronní osy, pro které je "njetí na pevný doraz" ještě aktivní, mohou být v tomto případě uvolněny pouze signálem rozhraní "Zrušit pevný doraz".

 **Pokud je to nutné a věta "FsReset" (viz bod 3.) ještě není naprogramována, je možné v dalším průběhu dílčího programu změnit aktivní točivý moment působící na pevném dorazu pomocí "FsTorque" (viz bod 1.).**

Programování

Syntaxe:

FsTorque(<Adr><Mom>)

Pro osu <Adr> aktivovat maximální povolený točivý moment <Mom>. Jsou povoleny synchronní a asynchronní osy.

Zkratka: FST(..)

FsMove

<Koord-Syn> <Posuv>
<Koord-Asy> <Posuv-Asy>

Aktivovat "njetí na pevnou zarážku". Najet se synchronními a asynchronními osami do naprogramovaných koncových poloh.

Zkratka: FSM(..)

FsReset

<Koord-Syn> <Posuv>
<Koord-Asy> <Posuv-Asy>

Vypnout "njetí na pevnou zarážku". Najet se synchronními a asynchronními osami do naprogramovaných koncových poloh.

Zkratka: FSR(..)

kde

<Adr>:

Fyzická nebo logická adresa osy, jsou možné také asynchronní osy.

<Mom>:

Maximální moment v % příslušného klidového momentu osy. Rozsah hodnot: 0 až 500 %.

<Koord-Syn>:

Požadované souřadnice koncového bodu synchronních os (např. "X100 Y100 Z100"). Jsou najížděny pomocí lineární interpolace všech zúčastněných os s ohledem na <posuv> a MP 1010 00030 (maximální zrychlení "njetí na pevný doraz").

<Posuv>:

Požadovaný posuv dráhy. Programuje se pomocí adresy F a je omezen parametry MP 1005 00030 (maximální posuv "njetí na pevný doraz") a MP 1005 00002 (maximální rychlost osy a rychlého chodu).

NC-Funktionen FsTorque, FSTFsMove, FSM FsReset, FSR

<Koord-Asy>: Požadované souřadnice koncového bodu asynchronních os. Jsou najížděny s ohledem na <posuv-Asy> a MP 1010 00030 (maximální zrychlení "njetí na pevný doraz").

<Posuv-Asy>: Požadovaný posuv asynchronních os. Programuje se pomocí adresy "FA" a je omezen parametry MP 1005 00030 (maximální posuv "njetí na pevný doraz") a MP 1005 00002 (maximální rychlost osy a rychlého chodu).

Příklad:

:	
N100 FST(X20)	Omezit točivý moment pro osu s označením "X" na 20% klidového momentu osy.
N110 FSM X100 F200	Aktivovat "njetí na pevnou zářku". Spustit pohyb s posuvem 200 mm/min do polohy X=100.
:	
:	Dále zpracovávat dílčí program.
N500 FSR	Vypnout "njetí na pevný doraz" a uvolnit všechny synchronní osy.
:	

Zvláštnosti a omezení:

- Během aktivního "njetí na pevný doraz" a ve větě "FsReset" nejsou dovoleny následující funkce:
 - G75 (měřicí dotyk)
 - InitMeas/FlyMeas (průběžné měření)
 - Redukce momentu přes rozhraní SPS
 - FsProbe (měření na pevném dorazu).
- Funkce FsMove musí být naprogramována společně nejméně s jednou souřadnicí koncového bodu. Její hodnota představuje maximální hloubku hledání, do které musí být nejpozději dosažen "pevný doraz".

NC-Funktionen GetAxis, GAX

4.33 Převzetí osy

GetAxis, GAX

Následek

Přebírá asynchronní osu do volajícího kanálu. Z asynchronní osy se tak stane synchronní.

Osa se pak dá naprogramovat v aktuálním kanálu pomocí svého fyzického nebo logického osového názvu.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

GetAxis (<PAN> | <PAI>, {<LAN>}, {<PAN> | <PAI>, {<LAN>}}...)

Zkratka: **GAX** (. .)

kde

<PAN>	Fyzický název osy. Určuje osu, která má být převzata do aktuálního kanálu.
<PAI>	Fyzický index osy. Účinky jako <PAN>.
<LAN>	Logický název osy. Je-li naprogramován, dostává integrovaná osa v aktuálním kanálu logický název <LAN>. <LAN> musí být definováno v MP 7010 00010 (logické označení os) nebo MP 7010 00020 (volitelné označení os).

Zvláštnosti a omezení:

- Přebíraná osa musí být zastavená.
Pokud tomu tak není, generuje řízení na rozdíl od funkce "WaitAxis" (viz strana 4-75) chybové hlášení a přeruší program.
- Osové polohy ve stejné větě musí být vždy naprogramovány za GetAxis(...) a smí být naprogramovány jedině tehdy, když není aktivní žádná transformace os.
- Přebírané osy nesmí být zúčastněny v žádné aktivní oblasti monitorování (viz strana 4-2).

Příklad:

```

:
N030 GAX (YP, , ZP, Z)
:

```

Fyzické osy YP a ZP jsou převzaty do volajícího kanálu.
Zatímco osa YP je i ve volajícím kanálu označena jako adresa YP, ZP dostává ve volajícím kanálu adresu Z.

NC-Funktionen HsBlkSwitch, HSB

4.34 Průběžná změna věty pomocí signálu HighSpeed

HsBlkSwitch, HSB

Následek

Umožňuje předčasnou změnu věty pomocí vstupů Highspeed v řízení. Tímto způsobem je možné předčasně ukončit **lineární** pojezdový pohyb v závislosti na externích událostech.



POZOR

Změna naprogramované dráhy!

Jestliže naprogramované koncové body věty "HsBlkSwitch" a její následné věty neleží na jedné přímce, má výskyt vnější události vždy za následek ne přesně předvídatelnou změnu průběhu dráhy!

 **Signály Highspeed mohou být zkonfigurovány pomocí parametrů stroje.**

Pro průběžnou změnu vět platí:

- vymazání bez zbývající dráhy v provozních režimech "následná věta", "jednotlivá věta" a "jednotlivý krok".
- vymazání se zbývající dráhou v provozních režimech "programová věta" a "ruční zadávání".
- **většinou bez** zastavení os. Věta je opuštěna s aktuální rychlostí (výjimky viz "Zvláštnosti a omezení").
- žádná kontrola maximální možné kapacity osového skoku
- působení závislé na aktuálním provozním režimu (následná/jednotlivá/programová věta, jednotlivý krok, ruční zadávání). Viz "Zvláštnosti a omezení".

Programování

Syntaxe:

HsBlkSwitch(HS<x>=<y>)

Průběžná změna věty

Zkratka: HSB(..)

kde

<x>: Číslo signálu Highspeed.
Rozsah hodnot: 1...8, Integer.

<y>: Logický stav signálu potřebný pro změnu věty.
0: Low
1: High

NC-Funktionen HsBlkSwitch, HSB

Zvláštnosti a omezení:

- Funkce vyžaduje lineární pojezdový pohyb jak ve větě "HsBlkSwitch", tak i v její následné větě.
- Funkce musí být zapsána s informacemi o dráze a může být zapsána spolu s dalšími podmínkami pro dráhu.
- V provozních režimech následná věta, jednotlivá věta a jednotlivý krok jsou koncové body os, které nejsou naprogramovány v následující větě, převzaty z předčasně ukončené věty.
- "Průběžná změna věty" se zastavením os se provádí v následujících případech:
 - Ohyb kontury >90 stupňů mezi větou "HsBlkSwitch" a její následnou větou.
 - Přesné zastavení je aktivní, tzn. G0(IPS...) nebo G1(IPS...).
 - Následná věta začíná díky doplňujícímu programování s $v=0$ (např. při aktivním programování KV nebo servo řízení).
 - Je aktivní interpolace po větách s G9(SHAPE...) nebo G9(ASHAPE).
 - Je aktivní provozní režim "jednotlivá věta", "jednotlivý krok", "programová věta" nebo "ruční zadávání".


Příklad: Posuv v závislosti na událostech.

Na přímkové dráze má být omezen posuv pojezdu v závislosti na vnější události.

K tomu jsou zapotřebí 3 NC-věty, v nichž jsou definovány vždy různé posuvové úlohy. Protože na přechodu vět nemá proběhnout zabrzdění na $v=0$, je zapotřebí průběžná změna věty.

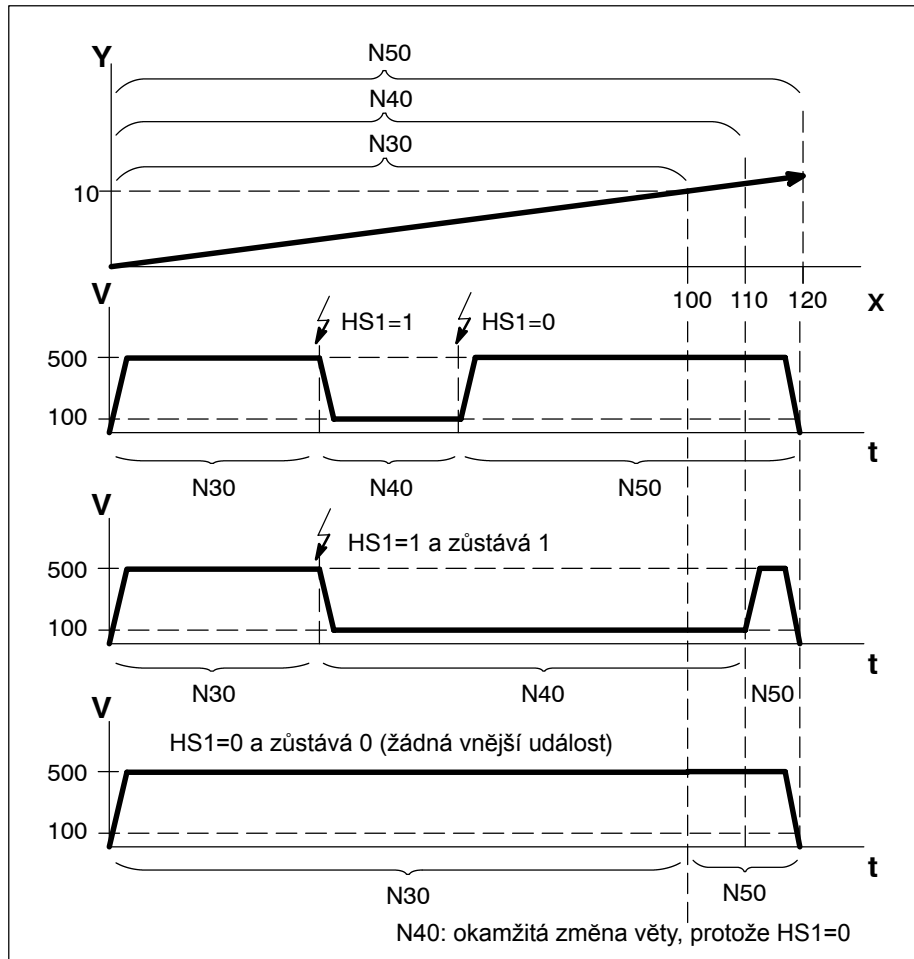
Všimněte si, že kvůli zadání úlohy (... "na přímkové dráze" ...) musí ležet všechny naprogramované koncové body na stejné přímce.

 **Koncové body následné věty nesmí být totožné s koncovými body předcházející věty. Jinak se v následné větě nespustí žádný pojezdový pohyb!**

 **Naprogramovaná pojezdová dráha následné věty ovlivňuje maximální možnou rychlost posuvu na přechodu vět. Příliš krátké pojezdové dráhy mohou způsobit automatické snížení rychlosti posuvu!**

:		
N20	G0 X0 Y0	Najetí do výchozí polohy.
N30	HSB(HS1=1) G1 X100 Y10 F500	Pojezd s F500, dokud signál Highspeed 1 není roven "High" nebo není dosažena poloha X100 Y10.
N40	HSB(HS1=0) X110 Y11 F100	Pojezd s F100, dokud signál Highspeed 1 není roven "Low" nebo není dosažena poloha X110 Y11.
N50	X120 Y12 F500	Pojezd po zbývající dráze do X120 Y12 s F500.
:		

NC-Funktionen HsBlkSwitch, HSB



NC-Funktionen HsBlkSwitch(..,HSSTOP=..), HSB(..,HSSTOP=..)

4.35 Změna věty s přerušením pomocí signálu HighSpeed HsBlkSwitch(..,HSSTOP=..), HSB(..,HSSTOP=..)

Následek

Umožňuje předčasnou změnu věty pomocí vstupů Highspeed v řízení. Tímto způsobem je možné předčasně ukončit **lineární** pojezdový pohyb v závislosti na externích událostech.



POZOR

Změna naprogramované dráhy!

Jestliže naprogramované koncové body věty "HsBlkSwitch" a její následné věty neleží na jedné přímce, má výskyt vnější události vždy za následek ne přesně předvídatelnou změnu průběhu dráhy!

 **Signály Highspeed mohou být zkonfigurovány pomocí parametrů stroje.**

Pro změnu věty s přerušením platí:

- ve všech provozních režimech vymazání se zbývající dráhou.
- Zabrzdění vždy na $v=0$; volitelně pomocí skoku rychlosti nebo Downslope.
Na konci věty proběhne zabrzdění na $v=0$ i tehdy, kdy nenastane vnější událost.

Programování

Syntaxe:

HsBlkSwitch(HS<x>=<y>,HSSTOP=<z>)

Změna věty s
přerušením

Zkratka: HSB(..,HSSTOP=..)

kde

<x>: Číslo signálu Highspeed.

Rozsah hodnot: 1...8, Integer.

<y>: Logický stav signálu potřebný pro změnu věty.

0: Low

1: High

<z>: Typ zabrzdění při výskytu události:

0: Lineární zpomalení na $v=0$ s max. brzdným zrychlením

-1: Skok rychlosti na $V=0$.

Zvláštnosti a omezení:

- Funkce vyžaduje lineární pojezdový pohyb jak ve větě "HsBlkSwitch", tak i v její následné větě.
- Funkce musí být zapsána s informacemi o dráze a může být zapsána spolu s dalšími podmínkami pro dráhu.

NC-Funktionen HsBlkSwitch(..,HSSTOP=..), HSB(..,HSSTOP=..)

Příklad:

Přerušení pojezdového pohybu v závislosti na události:

:

N20	G1	X0	Y0	F1000	Najetí do výchozí polohy.
N30	HSB(HS1=1,HSSTOP=-1)				Pojezd osy X s F10, dokud signál High-speed 1 není roven "High" nebo není dosažena poloha X10.
		X10	F10		Výskyt události spouští skok na v=0 a maže zbývající dráhu.
N40	HSB(HS2=1,HSSTOP=0)				Pojezd osy Y s F200, dokud signál High-speed 2 není roven "High" nebo není dosažena poloha Y100.
		Y100	F200		Výskyt události spouští lineární zpomalení na v=0 s max. brzdným zrychlením a maže zbývající dráhu.

NC-Funktionen HWOC HWOCDIS

4.36 Online oprava v souřadnicích obrobku

**HWOC
HWOCDIS**

Následek

S online opravou v souřadnicích obrobku se aktuální

- polohy nebo orientace v soustavě souřadnic obrobku kanálu, nebo
- polohy podélné osy nástroje ve směru TCS-z-

”přemísují” v důsledku opravy. Velikost opravy odvozuje řízení z připojeného ručního kolečka (normální případ), pohonu nebo z hodnoty proměnné CPL.

Podrobný popis funkce naleznete v příručce ”Funkční popis”.

Programování

Syntaxe:

HWOC({CHAN<č. kanálu>},CRDNO
<č. souřadnice>,{STEP<Inkr>})

Zapnutí online opravy

HWOCDIS({CHAN<č. kanálu>})

Vypnutí online opravy,
aktuální velikost opravy
zůstává interně uložena.

HWOC() nebo
HWOC(0)

Vypnutí online opravy a
vymazání příslušných hodnot
oprav.

kde

<Č. kanálu> Číslo kanálu, ve kterém je aktivována/deaktivována on-line oprava.

Pokud není naprogramováno **CHAN**<č. kanálu>, používá se číslo aktuálního kanálu.

<Č. souřadnice> Souřadnice, na kterou má účinkovat online oprava.

Rozsah hodnot: 1 až 9 a 103. Integer.

1 až 8: . Číslo souřadnice v zadaném kanálu

9 nebo 103: Souřadnice TCS.

<Inkr> Zadáání požadované velikosti kroku na 1 přírůstek.

Funguje, pouze když MP 7050 00926 = 0.

Zvláštnosti a omezení:

- Poloha souřadnice vytvořená online opravou se nekontroluje s ohledem na překročení softwarových koncových vypínačů. Proto musí být při použití online opravy aktivováno monitorování koncových vypínačů na straně pohonu.
- Online oprava není možná v provozních režimech ”Ruční seřízení” (Krokovací provoz) a ”Seřízení najetí do referenčního bodu”.
- G76 najíždí do polohy posunuté o aktuální hodnotu opravy.
- Funkce CPL PPOS a PCSPROBE neberou v úvahu hodnotu online opravy.
- G75 měří reálnou skutečnou polohu (vč. online opravy).
- ”FsProbe” měří reálnou skutečnou polohu (včetně online opravy).

NC-Funktionen InitMeas, IME HWOCDIS

4.37 Inicializace průběžného měření

InitMeas, IME

Následek

Funkce InitMeas slouží k inicializaci měřicího dotyku pro funkci "průběžné měření" (FlyMeas). Inicializace musí být vyvolána před první měřicí procedurou osy.

Programování

Syntaxe:

InitMeas(MpiAxis<*i*>)

Inicializace logiky měřicího dotyku

Zkratka: IME(..)

kde

<*i*>: Index fyzické osy, se kterou se má měřit.

Zvláštnosti a omezení:

- Funkce pracuje po větách.
- Funkce může být naprogramována spolu se všemi typy interpolace a pracuje paralelně s aktivní interpolací.
- Funkce musí být vyvolána před prvním vyvoláním funkce "FlyMeas" pro příslušnou osu.

NC-Funktionen JogWCSSelectHWOCDis

4.38 Krokovací provoz v souřadnicích obrobku

JogWCSSelect

Následek

Definuje pro provozní režim "Seřizovací provoz, krokování v souřadnicích obrobku", která souřadnice má být krokována nebo pojíždět pomocí ručního kolečka.

Pro krokování lze vybrat následující souřadnice:

- všechny souřadnice jednotlivých os stroje (pseudosouřadnice), pokud není aktivní žádná transformace os
- všechny lineární a orientační souřadnice - vzhledem k aktivní soustavě souřadnic WCS - pokud není aktivní transformace os s orientací
- souřadnice TCS z, pokud není aktivní transformace os s orientací.

Zda jsou rychlost posuvu a velikost kroku (při inkrementálním krokování) odvozeny od nastavení existující osy stroje nebo musí být explicitně naprogramovány, závisí na nastavení v MP 7050 01010.

 **Podrobný popis funkce naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

```
JogWCSSelect ({JWSCHAN<Č. kanálu>,}JWSCOORD<Č. souřadnice>,
{JWSFEED<F-hodnota>{, JWSSTEP<Inkr>}})
```

kde

<Č. kanálu> Číslo kanálu, ve kterém má být krokována souřadnice.
Standardní: Číslo kanálu, ve kterém je naprogramována funkce.

<Č. souřadnice> Výběr souřadnice.
Rozsah hodnot: 1 až 8 a 103. Integer.
1 až 8: Číslo krokované souřadnice
103: Má být krokována souřadnice TCS.

<F-hodnota> Rychlost krokování.
Zadaná hodnota v závislosti na aktivní měrné soustavě (G71,G70) v jednotkách mm/min, Inch/min, resp. stupně/min.

Funguje, pouze když MP 7050 01010 = 0.

<Inkr> Volba inkrementálního krokování a zadání požadované velikosti kroku v přírůstcích.
Funguje, pouze když MP 7050 01010 = 0.

Zvláštnosti a omezení:

- Funkce je určena pro použití ve speciálním NC-programu pro volbu souřadnic (viz též MP 705001110).
Podrobné informace naleznete v příručce "Funkční popis".

NC-Funktionen KvProg, KVP

4.39 Programování KV

KvProg, KVP

Následek

Funkce umožňuje programem řízené změny hodnot KV jednotlivých os. Tak lze dočasně zvýšit tuhost os (např. při frézování otvoru).

Po spuštění řízení načte NC mimo jiné aktuální hodnoty KV všech připojených pohonů a uloží je.

Během programování KV načítá řízení naprogramované hodnoty KV do příslušných pohonů (parametr S-0-0104).

Při vypnutí programování KV předá řízení dříve uložené hodnoty KV zpět do pohonů a obnoví tak znovu původní stav.

Platí:

- $KV = (\text{posuv dráhy v m/min}) / (\text{doběh v mm})$
- Před větou s přepnutím KV proběhne vždy zabrzdění na $v=0$, protože hodnota KV v pohonu může být přepnuta jedině v klidovém stavu.
- Po příkazu přepnutí KV čeká řízení interně vždy na potvrzení od všech zúčastněných pohonů.
- Přepnutí KV se provádí bezprostředně před případným pojezdovým pohybem naprogramovaným ve stejné větě.

Příklad:

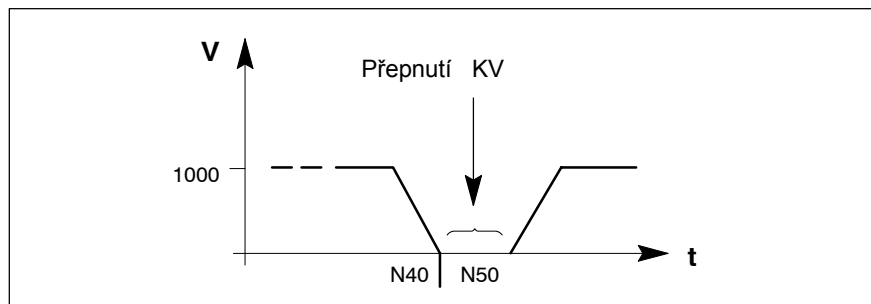
```

:
N10 G1 F1000
:
N40 X40
:
N50 KVP(X2) X50
:

```

Polohování se standardní hodnotou KV.

Předat hodnotu KV 2 ose X a potom pojíždět.



Programování

Syntaxe:

`KvProg(<osa1>{,<osa2>}{,...})`

Zapnutí programování KV.

`KvProg({0})`

Vypnutí programování KV.

Zkratka: `KVP(..)`

kde

<Osa> Fyzické (systémové) nebo logické (vztahované ke kanálu) označení osy včetně požadované hodnoty KV.
Max. programovatelná hodnota KV: 655.35

NC-Funktionen KvProg, KVP

Zvláštnosti a omezení:

- Smí být programovány pouze osy, které jsou přiřazeny aktuálnímu kanálu.

Příklad:

```
:  
:  
N10 G0 X0 Y0 Z100  
N20 KVP(Z2.1)  
:  
N30 G1 Z0  
N40 KVP() Z100  
:  
:
```

Výchozí bod: ve všech pohonech je aktivní hodnota KV 1.0.
Polohování s hodnotou KV = 1.0
Předat pohonu fyzické osy Z hodnotu KV 2.1.
Polohování s hodnotou KV = 2.1
Vypnutí programování KV.
NC automaticky načte do všech pohonů opět hodnotu KV 1.0.
Potom pojezd osy Z.


NC-Funktionen LEN

4.40 Rozdělení pojezdové věty: délka dílčí dráhy

LEN

Následek

Rozděluje naprogramovanou pojezdovou větu na několik stejně dlouhých dílčích drah.

 **Působí pouze v souvislosti s funkcemi "Děrování" (viz strana 4-30) a "Vysekávání" (viz strana 4-74).**

Programování

Syntaxe:

LEN=<Hodnota>

kde

<Hodnota> U lineárních vět: délka dílčí dráhy.
U kruhových vět: délka oblouku.
Programovací jednotka jako pro souřadnice os.
<Hodnota> nemusí být celým dělitelem naprogramované dráhy. V NC se automaticky interně určuje efektivní hodnota LEN menší nebo rovná naprogramované hodnotě LEN, takže je efektivní délka dílčí dráhy vždy celým dělitelem naprogramované dráhy.

Zvláštnosti a omezení:

- V rozdělených kruhových větách probíhá pohyb od zdvihu ke zdvihu vždy lineárně!
- LEN působí modálně, dokud je aktivní funkce děrování/vysekávání. Může však být po větách překryto parametrem NUM (viz strana 4-76).
- Programování parametru LEN je možné kdykoli při uvolněném děrování/vysekávání (MP 8001 00010). Rozdělení věty ale začne až po zapnutí děrování/vysekávání.

Příklad:

- G90 je aktivní (programování absolutních rozměrů)
- Aktivní rovina: X/Y
- Aktuální poloha: X=0, Y=0, C=0
- Děrování/vysekávání je vypnuté.

:

N20 X100 Y100 LEN=15

Rozdělit následující pojezdové věty na stejné dílčí dráhy maximálně po 15 mm. LEN ještě nepůsobí, protože je děrování/vysekávání vypnuté.

N30 X200 Y200 C180 Punch(1)

Zapnout děrování. LEN působí. Délka dráhy je rozdělena na 10 dílčích vět.

Výsledné polohy zdvihů (X,Y,C):

P1 (110,110,18)

P2 (120,120,36)

:

P10 (200,200,180).

NC-Funktionen LEN

N40 Y290 C210

Délka dráhy (90 mm) je rozdělena na
6 dílčích vět. Výsledné polohy zdvihů
(X,Y,C):

P11 (200,215,185)

P12 (200,230,190)

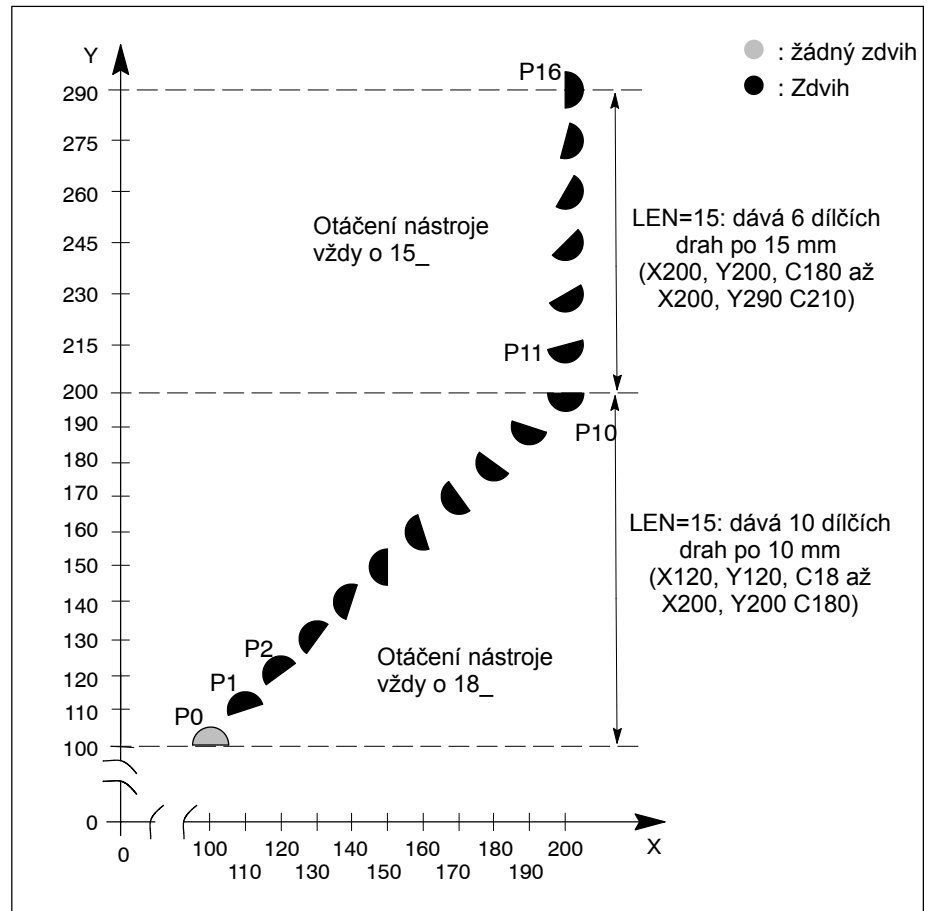
:

P16 (200,290,210).

Vypnout děrování.

N50 Punch()

:



NC-Funktionen LFP LFCConf, LFC

4.41 Řízení výkonu laseru v závislosti na rychlosti posuvu

LFP
LFCConf, LFC

Následek

Řídí výkon laseru pomocí analogového napěového signálu (0...10 V) v závislosti na aktuálním posuvu dráhy v_{Bahn} .

 **Podrobný popis funkce naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

LFP(1) nebo
LFP

Zapnutí řízení výkonu laseru s aktuální parametrizací.

LFP({LL(...)})

Zapnutí řízení výkonu laseru s doplňkovou parametrizací, jak je popsáno pod **LFCConf**.

LFP(0)

Vypnutí řízení výkonu laseru.

LFCConf({LL(<Spg>,<Vmin>)},
{UL(<Spg>,<Vmax>)},
{PL(<Režim>)},
{CD(<Souřadnice 1>{,<Souřadnice n>
...}),})

Parametrizace řízení laserového výkonu.

Zkratka: LFC(..)

kde

<Spg> Napětí v % maximálního výstupního napětí (10 V).
V souvislosti s <Vmin>: vydává se při rychlostech posuvu menších než <Vmin>.
V souvislosti s <Vmax>: vydává se při rychlostech posuvu větších než <Vmax>.

<Vmin> Dolní mez rychlosti posuvu.
Zadaná hodnota v závislosti na aktivní měrné soustavě (G71,G70) v jednotkách mm/min, Inch/min, resp. stupně/min.

<Vmax> Horní mez rychlosti posuvu.
Zadaná hodnota v závislosti na aktivní měrné soustavě (G71,G70) v jednotkách mm/min, Inch/min, resp. stupně/min.

<Režim> Definuje, které souřadnice mají být použity k určení v_{Bahn} :
APL Všechny souřadnice aktuální roviny.
ASP Všechny souřadnice aktuálního prostoru.
CFD Souřadnice podle MP 7050 00820.

NC-Funktionen LFP LFCConf, LFC

<Souřadnice x> Definuje, které souřadnice mají být použity k určení V_{Bahn} .
Zadané hodnoty: Názvy zúčastněných prostorových souřadnic (při aktivní transformaci os) nebo logické názvy os (pseudosouřadnice; při deaktivované transformaci os).

Příklady:

LFPON LL(10,100)

Zapnout řízení laserového výkonu. Doplnující parametrizace: Výstupní napětí na 10% (=1V), když rychlost posuvu klesne pod 100 mm/min.

LFP LL(10,100)

Pouze parametrizace: Výstupní napětí na 10% (=1V), když rychlost posuvu klesne pod 100 mm/min.

LFP UL(80,500)

Pouze parametrizace: Výstupní napětí na 80% (=8V), když rychlost posuvu vzroste nad 500 mm/min.

Zvláštnosti a omezení:

- Analogové výstupy, které jsou k dispozici, omezují počet kanálů, které mohou používat funkci.
- Napěový signál pro řízení laseru klesá v následujících případech na 0 V:
 - Došlo k chybě (běhová chyba, chyba stavové třídy 1)
 - Je deaktivován signál "Pohon v provozu" pohonu zúčastněného na dráze (žádné FG, pohon vypnutý)
 - Je aktivní signál "Zastavení posuvu".

NC-Funktionen LinUpFeed, LNU

4.42 Rychlostní profily (lineární změnové funkce)

Funkce umožňují definovat úseky obrábění, které mají být pojížděny s vlastními rychlostními profily. Jako jednotlivé "stavební kameny" jsou zde k dispozici

- 1 interpolátor konstantního pojezdu.
- 3 akcelerační a 3 brzdné interpolátory (vždy s lineárním a sinusovým průběhem rychlosti a ve tvaru \sin^2),

Platí:

- Všechny níže popsané funkce působí modálně, tvoří s G8 a G9 modální skupinu a vzájemně se ruší.

Působení akceleračních interpolátorů

LinUpFeed, LNU
SinUpFeed, SNU
Sin2UpFeed, S2U

Řízení zrychluje počínaje rychlostí V_0 (rychlost na začátku věty) po celé naprogramované délce dráhy na cílovou rychlost V_1 .

V závislosti na naprogramované funkci to probíhá s nárůstem rychlosti lineárně nebo podle funkce sinus či \sin^2 .

Cílová rychlost V_1 je dosažena v naprogramovaném koncovém bodě a vyplývá z naprogramovaného posuvu, váženého aktuální hodnotou Override. Je omezena

- maximálním zrychlením posuvu a
- maximální přípustnou rychlostí posuvu.

Obě veličiny vypočítává řízení po segmentech dráhy pro každou NC-větu a provádí v souvislosti s maximální přípustnou rychlostí náhled 1 věty. To brání překročení maximálních osových rychlostí v příslušné následné větě.

 **Pokud V_1 není větší než V_0 , je volání interpolátoru zrychlení v aktuální větě ignorováno.**

Chování při změnách Override:

- Zvýšení Override vede k novému výpočtu akcelerační lineární změny.
- Zmenšení Override na výsledné hodnoty menší než počáteční rychlost V_0
 - vede u "LinUpFeed" k novému výpočtu brzdné lineární změny, která bude trvat do naprogramovaného koncového bodu.
 - je u "SinUpFeed" a "Sin2UpFeed" ignorováno.

Působení interpolátoru konstantního pojezdu

ConstFeed, CFD

Řízení se snaží dosáhnout naprogramované požadované rychlosti s ohledem na maximální přípustnou rychlost posuvu a aktuální nastavení Override.

Změny rychlosti při změnách Override se provádějí vždy s přípustným zrychlením.

NC-Funktionen LinUpFeed, LNU LinDownFeed, LND

Působení brzdných interpolátorů**LinDownFeed, LND**
SinDownFeed, SND
Sin2DownFeed, S2D

Řízení brzdí počínaje rychlostí V_0 (rychlost na začátku věty) po celé naprogramované délce dráhy až do zastavení ($V_1=0$).

V závislosti na naprogramované funkci to probíhá s poklesem rychlosti lineárně nebo podle funkce sinus či \sin^2 .

Změny Override zůstávají bez účinku, až na následující výjimku:

Pokud je v předchozí větě nastaven Override na 0% a pak je dosažena požadovaná rychlost 0 přesně při změně věty pro brzdný interpolátor, zastavuje řízení brzdný interpolátor tak dlouho, dokud se Override nezvýší na hodnotu >0 !

Rychlost přeskočí o jeden krok zrychlení (podle přípustného zrychlení posuvu) nahoru.

Na základě takto získané hodnoty rychlosti vypočítává řízení potřebnou brzdnu lineární změnu. Skutečná hodnota Override pak zůstává až do konce věty bez účinku.

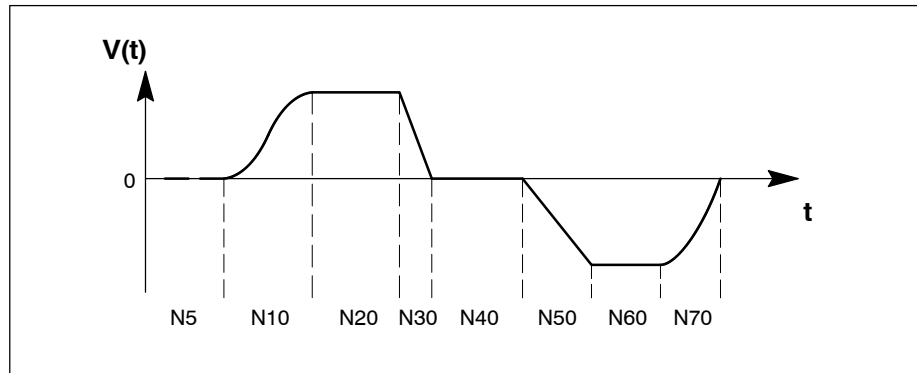
Programování

Syntaxe:

LinUpFeed Zkratka: LNU	Aktivovat akcelerační interpolátor s lineárním nárůstem rychlosti.
SinUpFeed Zkratka: SNU	Aktivovat akcelerační interpolátor se sinusovým nárůstem rychlosti.
Sin2UpFeed Zkratka: S2U	Aktivovat akcelerační interpolátor s nárůstem rychlosti tvaru \sin^2 .
ConstFeed Zkratka: CFD	Aktivovat interpolátor konstantního pojezdu.
LinDownFeed Zkratka: LND	Aktivovat akcelerační interpolátor s lineárním poklesem rychlosti.
SinDownFeed Zkratka: SND	Aktivovat akcelerační interpolátor se sinusovým poklesem rychlosti.
Sin2DownFeed Zkratka: S2D	Aktivovat akcelerační interpolátor s poklesem rychlosti tvaru \sin^2 .

NC-Funktionen LinUpFeed, LNU LinDownFeed, LND

Příklad: Rychlostní profil pro kyvadlový cyklus osy U



N5 G0 U10	Najet s osou U do počáteční polohy (U=10mm)
N10 S2U U17 F500	Zrychlovat s průběhem \sin^2 až do polohy U=17. Požadovaný posuv v koncovém bodě: F=500 mm/min.
N20 CFD U23	Konstantní pojezd do polohy U=23.
N30 LND U29	Lineární brzdění do polohy U=29. Koncová rychlost: 0 mm/min.
N40 G4 F0.5	Doba prodlevy v bodě obratu.
N50 LNU U20	Lineární zrychlení do polohy U=20.
N60 CFD U17	Konstantní pojezd do polohy U=17.
N70 SND U10	Sinusové brzdění do polohy U=10. Koncová rychlost: 0 mm/min.

Zvláštnosti a omezení:

- V naprogramované větě je vždy zapotřebí zadání požadované souřadnice koncového bodu.
- Všechny lineární změnové funkce jsou použitelné jen v provozním režimu automatika / následná věta. Jiné provozní režimy (ruční zadávání, jednotlivá věta, jednotlivý krok nebo programová věta) vedou k běhové chybě.
- Při používání brzdících interpolátorů ve spojení s velmi krátkými pojezdovými drahami je možné překročení zrychlení, které může spustit chybu servo pohonu. Dodržujte proto při vytváření dílčích programů maximální možnou dynamiku stroje.
- Když jsou aktivní interpolátory konstantního pojezdu a akcelerační interpolátory, není dovoleno používat pomocné funkce nebo funkce jako např. "přesné zastavení" (mohou vést k překročení rychlosti). Zakázané funkce jsou např.: G0, G4, "KvProg", G63, G33, G61, "G1(IPS)", G75, "G74(HOME)", "HsBlkSwitch", "Writeld".

NC-Funktionen LinModZp, LMZ

4.43 Vynulování modulu osy (lineární nekonečné osy) LinModZp, LMZ

Následek

Pomocí "LinModZp" můžete definovat aktuální polohu lineární nekonečné osy jako nový programový nulový bod.

Osové zobrazení pak přeskočí na hodnotu "0". Vzniklý offset zůstává uložený v řízení a interně se přičítá ke všem dalším souřadnicovým údajům příslušné osy.

Po základním nastavení je znovu vypočítán ze zobrazené hodnoty a vymazán.

 **Informace o lineárních nekonečných osách viz parametr stroje 1003 00004.**

Programování

Syntaxe:

LinModZp

Definovat aktuální polohy všech lineárních nekonečných os zkonfigurovaných v MP 1003 00004 jako nový programový nulový bod.

LinModZp(LinModAxis<*i*>)

Definovat aktuální polohu lineární nekonečné osy s fyzickým osovým indexem, zkonfigurované v MP 1003 00004, <*i*> jako novou programovou souřadnici nulového bodu této osy.

Zkratka: LMZ(..)

kde

<*i*>: Fyzický index osy.

Příklad:

N10 LMZ

:

:

Definovat aktuální polohy všech zkonfigurovaných lineárních nekonečných os jako nový programový nulový bod.

N50 LMZ(LinModAxis1) X-20


:

:

Definovat aktuální polohu lineární nekonečné osy s osovým indexem 1 jako novou programovou souřadnici nulového bodu. Potom najet na X-20.

Zvláštnosti a omezení:

- Zadání poloh větších než modulo hodnota není dovoleno.
- Záporné zadané polohy jsou možné, ale jejich absolutní hodnota musí být menší než modulo hodnota.

 **Modulo hodnota je v pohonu definována pomocí parametru SERCOS S-0-0103 při spuštění SERCOS. Změněná modulo hodnota je účinná až po novém spuštění SERCOS!**

NC-Funktionen MainSp, MSP

4.44 Přepnutí hlavního vřetena

MainSp, MSP

Následek

Definuje - na rozdíl od nastavení v MP 7020 00010 - zadané vřeteno jako hlavní vřeteno v aktuálním kanálu.

Tímto způsobem je možné v dílčím programu dynamicky určit, na které vřeteno mají funkce

- G33 (řezání závitů)
 - G95 (programování posuvu v mm/ot.) a
 - G4 (doba prodlevy)
- působit.

Programování

Syntaxe:

MainSp(*<Num>*) nebo
MainSp(*<Vřeteno>*)

Zkratka: **MSP...**

kde

<Num> Číslo (index) vřetena.
Rozsah zadání: -1; 1...8. Integer.
-1: Hlavní vřeteno podle MP 7020 00010.

<Vřeteno> jako *<Num>*
nebo
název vřetena (např. S1)

NC-Funktionen Mirror(...), MIR(...)

4.45 Pomůcka: zrcadlové převrácení


Mirror(...), MIR(...)

Následek

Funkce zrcadlové převrácení patří k pomůckám.

Řízení obrátí naprogramovanou konturu nebo např. vrtné schéma zrcadlově převrácené.

Zrcadlové převrácení se vždy vztahuje k aktuálnímu bodu zrcadlového převrácení (viz PoleSet na straně 4-21). Pokud nebyl bod zrcadlového převrácení explicitně naprogramován, je jím aktuální programový nulový bod.

-  **Zrcadlové převrácení je pomůcka, a proto nemění aktuální programovou soustavu souřadnic. Pomůcka pouze představuje jinou možnost zadávání programových souřadnic. Zrcadlové převrácení se může používat také společně s nastavením měřítka a otáčením.**

Platí:

- Funkce působí modálně. Zůstává účinná tak dlouho, dokud není vypnuta.
- Smí být naprogramována s jinými podmínkami a pomocnými funkcemi ve stejné větě.

Programování

Syntaxe:

Mirror(<Osa1>1{,<Osy>1})

Zapnutí zrcadlového převrácení pro zadané osy.

Mirror(<Osa1>0{,<Osy>0})

Vypnutí zrcadlového převrácení pro zadané osy.

Mirror(0) nebo
Mirror()

Vypnutí zrcadlového převrácení pro všechny osy v kanálu. Najeté osové polohy zůstávají zachovány, dokud nejsou nově naprogramovány.

Zkratka: MIR(..)

kde

<Osa1>, <Osy>:

Osové adresy (např. X) ve spojení s hodnotou "1" aktivují funkci: všechny další naprogramované dráhové příkazy příslušných os (např. X100) jsou interně vynásobeny hodnotou "-1".

Zrcadlové převrácení je účinné až s další pojezdovou informací.

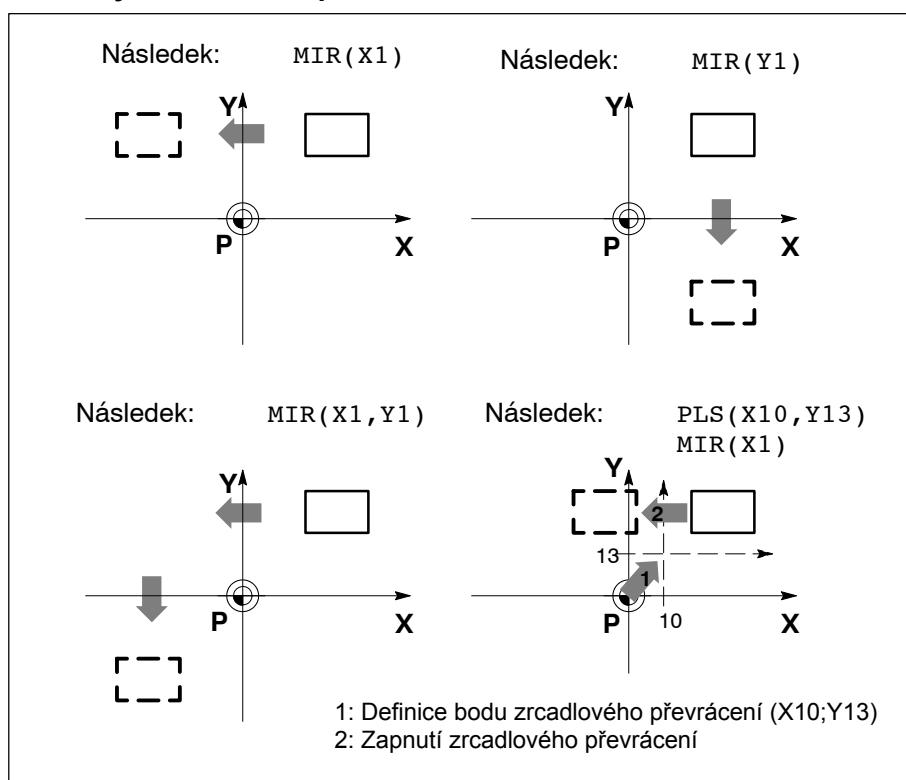
Zvláštnosti a omezení:

- Funkce bere v úvahu interpolační parametry při kruhové interpolaci.

NC-Funktionen Mirror(...), MIR(...)

- Má vliv na programovatelné posunutí kontur (Shift). Viz strana 4-44.
- **Nemá** vliv na:
 - Posunutí nulových bodů (G54-G59.5; viz strana 3-46),
 - Posunutí programových souřadnic (Trans nebo ATrans; viz strana 4-72),
 - Nastavení programových poloh ("SetPos"; viz strana 4-43),
 - Najetí na souřadnice referenčního bodu (G74 ; viz strana 3-55),
 - Najetí do pevné osové polohy stroje (G76 ; viz strana 3-58),
 - Hodnoty oprav poloměru frézy a délky nástroje.

Příklady zrcadlového převrácení:



Zrcadlové převrácení orientačního vektoru

Orientační vektor se zrcadlově převrací výhradně po složkách podle následující funkční syntaxe.

Případně aktivní nastavení měřítka nebo bodu zrcadlového převrácení / otáčení nemá žádný vliv na výsledek.

Syntaxe:

Mirror(O(<Sx>,<Sy>,<Sz>))

Zapnout/vypnout zrcadlové převrácení pro zadané komponenty orientačního vektoru.

Zkratka: MIR(..)

NC-Funktionen Mirror(...), MIR(...)

kde

 $\langle S_x \rangle, \langle S_y \rangle, \langle S_z \rangle$

Zrcadlové faktory pro jednotlivé složky vektoru.

0: Žádné zrcadlové převrácení

1: Zrcadlové převrácení

Pro orientační vektor platí:

- Polární souřadnice φ a θ nemohou být zrcadlově převráceny.
- Naprogramování $\text{MIR}(\text{phi1}, \text{theta1})$ není dovoleno.

NC-Funktionen Nibble, NIB

4.46 Vysekávání

Nibble, NIB

Následek

Zapíná nebo vypíná funkci "vysekávání".

Při aktivovaném vysekávání se spouští zdvih v následujících případech:

- na konci každé naprogramované, resp. funkcí "NUM" (viz strana 4-76) nebo "LEN" (viz strana 4-62) vytvořené dílčí dráhy a
- na začátku první dílčí dráhy, pokud v předchozí větě nebyl naprogramován žádný pojezdový pohyb v aktivní rovině nebo vypnuto vysekávání.

Výsledný pojezdový pohyb začíná vždy teprve po skončení zdvihu.



Funkce pro ovlivnění času spuštění zdvihu:

PtDefault (viz strana 4-28)

PtBlkEnd (viz strana 4-26)

PtInpos (viz strana 4-29).

Programování

Syntaxe:

Nibble(1) nebo
Nibble

Zapnout vysekávání.

Nibble(0) nebo
Nibble()

Vypnout vysekávání.

Zkratka: **NIB (. .)**

Zvláštnosti a omezení:

- Vysekávání musí být uvolněno parametrem MP 8001 00010.
- Je nezbytné naprogramovat funkci "NUM" nebo "LEN". Tak generuje řízení z naprogramované pojezdové dráhy automaticky dílčí dráhy, na jejichž konci je vždy spuštěn zdvih.
- Funkce působí modálně a již na pojezdové pohyby, které jsou naprogramovány ve stejné větě.
- Je-li zapnuto děrování (viz strana 4-30), ruší se volba funkce vysekávání.
- Věty, které neobsahují žádnou osovou souřadnici z aktivní roviny, nespouštějí také žádný zdvih.
- Pokud SPS potlačí spuštění zdvihů, zůstane obrábění stát v poloze spuštění zdvihu, dokud SPS spuštění zdvihů znovu neuvolní.

NC-Funktionen Nibble, NIB

Příklad:

- G90 je aktivní (programování absolutních rozměrů)
- Aktivní rovina: X/Y
- Aktuální poloha: X=0, Y=0, C=0

:

N10 LEN=12

Rozdělit pojezdové větvy na stejné dílčí dráhy maximálně po 12 mm.

Hodnota LEN musí být naprogramována před zapnutím vysekávání!

N20 C10 Nibble(1)

Zapnout vysekávání. Modální LEN působí. Osa C se otočí na 10 stupňů.

Žádný zdvih, protože osy X a Y nejsou naprogramovány.

N30 X0

Žádný pojezdový pohyb, protože osa X je již v poloze 0.

N40 X110

Zdvih, protože osa X leží v aktivní rovině.

Pojezdová věta se rozdělí na 10 stejných dílčích drah po 11 mm. Zdvih v polohách X11, X22, X33 ...X99, X110.

N50 Y30 NUM=3

Přídavný zdvih v poloze X0, protože v N30 neproběhl žádný pojezdový pohyb.

Překrývá modálně působící hodnotu LEN (N10) pro aktuální větu.

Rozdělit pojezdové větvy na 3 stejné dílčí dráhy. Zdvih při Y10, Y20, Y30.

N60 Y90

LEN působí opět z N10.

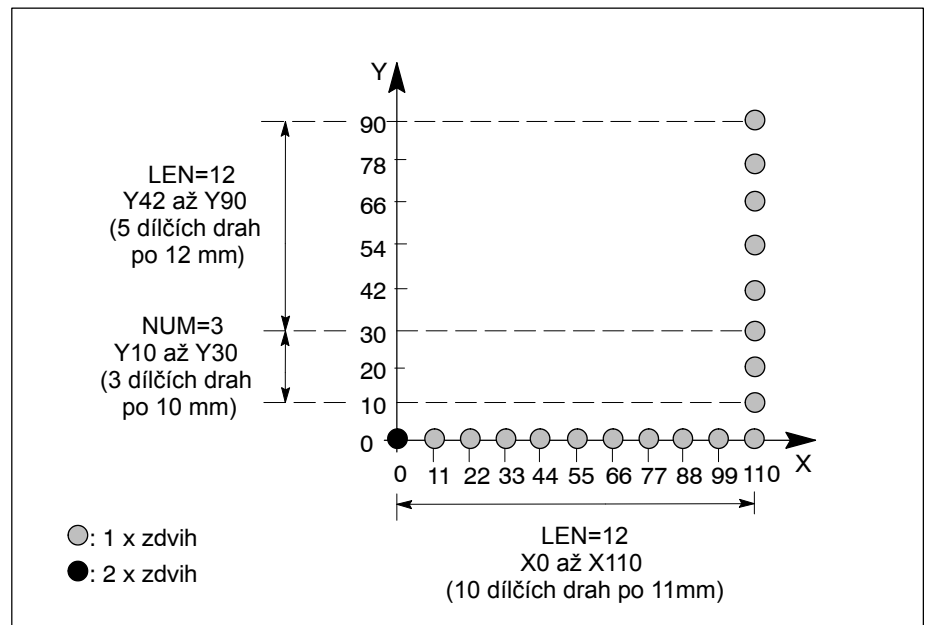
Zdvih při Y42, Y54, Y66, Y78, Y90.

N70 X50 Y50 Nibble()

Vypnout vysekávání.

:

Pojezdový pohyb na X=50, Y=50.




NC-Funktionen NUM

4.47 Rozdělení pojzdové věty: Počet dílčích drah

NUM

Následek

Rozděluje naprogramovanou pojzdovou větu na definovaný počet stejně dlouhých dílčích drah.

 **Působí pouze v souvislosti s funkcemi "Děrování" (viz strana 4-30) a "Vysekávání" (viz strana 4-74).**

Programování

Syntaxe:

NUM=<Hodnota>

kde

<Hodnota> počet dílčích drah.
Zadaná hodnota: Integer, větší než 0.
NUM=1 nevede k žádnému rozdělení.

Zvláštnosti a omezení:

- V rozdělených kruhových větách probíhá pohyb od zdvihu ke zdvihu vždy lineárně!
- NUM působí pouze v naprogramované větě a překrývá aktivní LEN (viz strana 4-62).
- Programování parametru NUM je možné pouze při uvolněném (MP 8001 00010) a aktivním děrování/vysekávání.

Příklad:

Viz příklad v kap. 4.40 od strany 4-62.

O(), ROTAX() phi, theta, psi

4.48 Orientační programování

O(), ROTAX()
phi, theta, psi

Orientuje nástroj (fréza, vrták, laser, podavač) vzhledem k aktuální programové soustavě souřadnic (PCS), libovolně v prostoru.

- ☞ **Pokud se má kromě orientačního pohybu nástroje pohybovat také TCP, viz doplňující funkce "Programování prostorových souřadnic" v příručce "Funkční popis".**

Je-li programován orientační pohyb nástroje bez současného pohybu TCP, působí aktivní posuv (F) výhradně na orientační pohyb. V případě potřeby jsou přitom synchronně prováděny naprogramované pohyby pseudosouřadnic (pohyby pseudosouřadnic: pohyby, které jsou vyvolány naprogramováním přímých osových poloh). Alternativně existuje při aktivní funkci G94 možnost programování "Omega".

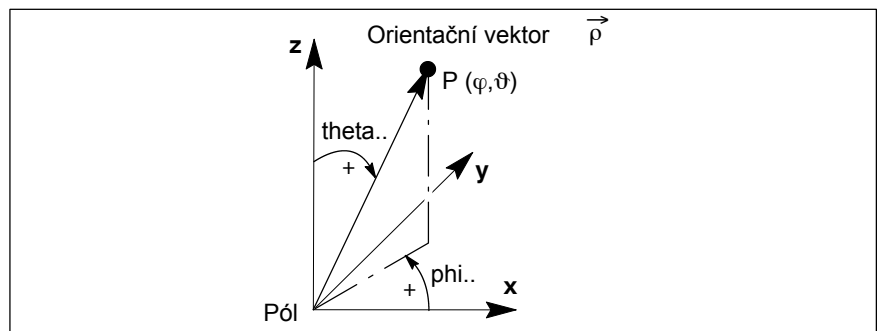
Orientační vektor leží podél osy souměrnosti nástroje a ukazuje k upínadlu nástroje. Jsou k dispozici různé typy orientačního programování:

4.48.1 Vektorová orientace

Následek

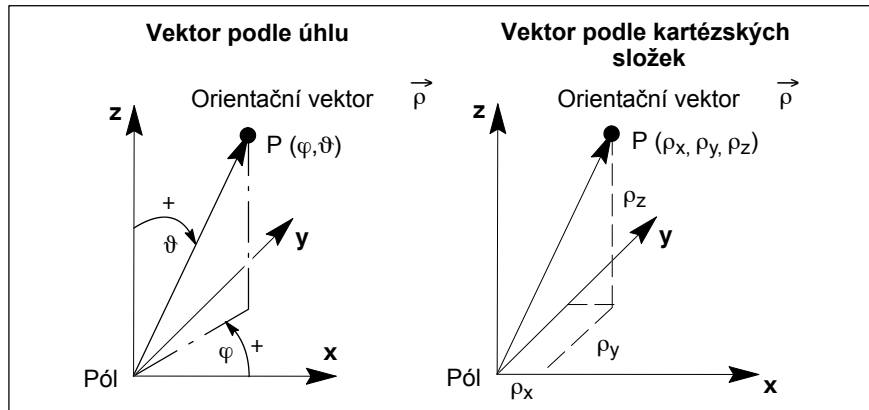
Programování orientace nástroje je alternativně možné pomocí

- orientačních souřadnic orientačního vektoru (např. phi a theta).

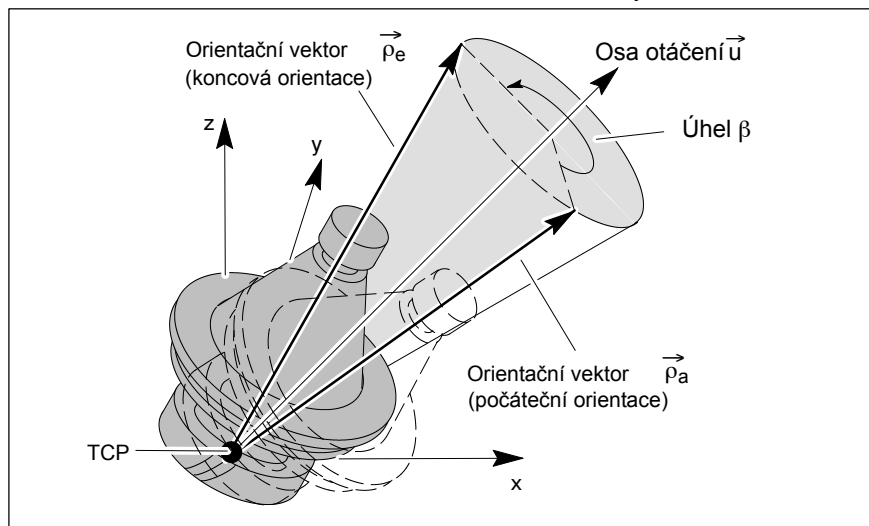


O(), ROTAX() phi, theta, psi

- Funkce O(..) s polárními úhly (φ, ϑ) nebo kartézskými složkami (ρ_x, ρ_y, ρ_z) orientačního vektoru.



- Funkce ROTAX(..) pro definici osy otáčení, orientované libovolně v prostoru, okolo které lze otáčet orientační vektor pomocí funkce O(..) o určitý úhel β .
Naprogramování osy otáčení je přitom možné s polárními úhly (φ_u, ϑ_u) stejně jako pomocí kartézských složek (u_x, u_y, u_z).



Platí:

- Vektorová orientace může být naprogramována pouze společně s transformací os, jejíž 100. místo v typu = 2.
- Orientační pohyb probíhá jako pohyb otočného ukazatele orientačního vektoru z počáteční do naprogramované koncové orientace.
- Vhodné pro rotačně symetrické nástroje.

O(), ROTAX() phi, theta, psi

Programování

1. Aktivujte přípustný typ transformace os (100. místo v typu = 2) pomocí "Coord(...)" (viz strana 4-30).
2. Dále použijte uvedenou syntaxi.

Syntaxe:

$\{<Nr>\} \{<TCP-Bew>\} \{<phi><\varphi>\} \{<theta><\vartheta>\}$ nebo
 $\{<Nr>\} \{<TCP-Bew>\} \mathbf{O}(<\varphi>, <\vartheta>)$ nebo
 $\{<Nr>\} \{<TCP-Bew>\} \mathbf{O}(<\rho_x>, <\rho_y>, <\rho_z>)$ nebo
 $\{<Nr>\} \{<TCP-Bew>\} \mathbf{ROTAX}(<\varphi_u>, <\vartheta_u>) \mathbf{O}(<\beta>)$ nebo
 $\{<Nr>\} \{<TCP-Bew>\} \mathbf{ROTAX}(<u_x>, <u_y>, <u_z>) \mathbf{O}(<\beta>)$

kde

$<Nr>$	Číslo věty. Viz strana 2-20.
$<TCP-Bew>$	Přídavný pohyb TCP. Syntaxe viz příručka "Funkční popis" pod programováním prostorových souřadnic.
$<phi>$	Název úhlu, který je zadán v MP 7080 00010[4]. Standardní: phi
$<theta>$	Název úhlu, který je zadán v MP 7080 00010[5]. Standardní: theta
$<\varphi>, <\vartheta>, <\varphi_u>, <\vartheta_u>$	Absolutní úhlové hodnoty ve stupních. Rozsah hodnot: $0^\circ \leq \varphi < 360^\circ$; $0^\circ \leq \vartheta \leq 180^\circ$. Pokud je jako ϑ naprogramována jiná hodnota, je automaticky přepočítána do zadaného intervalu.
$\mathbf{O}(<\varphi>, <\vartheta>)$	Orientace podle funkce O(..) a polárních úhlů $<\varphi>$ a $<\vartheta>$ orientačního vektoru.
$\mathbf{O}(<\rho_x>, <\rho_y>, <\rho_z>)$	Orientace podle funkce O(..) a kartézských složek $<\rho_x>, <\rho_y>, <\rho_z>$ orientačního vektoru v absolutních rozměrech. Automatické normování na 1. Proto dávají následující příklady údajů totožnou orientaci: O(1,2,4), O(2,4,8)
$\mathbf{ROTAX}(<\varphi_u>, <\vartheta_u>)$	Definice osy otáčení pomocí polárních úhlů (φ_u, ϑ_u).
$\mathbf{ROTAX}(<u_x>, <u_y>, <u_z>)$	Definice osy otáčení pomocí kartézských složek $<u_x>, <u_y>, <u_z>$ v absolutním rozměru. Automatické normování na 1.
$\mathbf{O}(<\beta>)$	Udává inkrementální úhel ve stupních $<\beta>$, o který se má otočit orientační vektor kolem osy otáčení. Jsou povoleny hodnoty větší než 360 stupňů. Směr otáčení se vybírá pomocí znaménka.

Zvláštnosti a omezení:

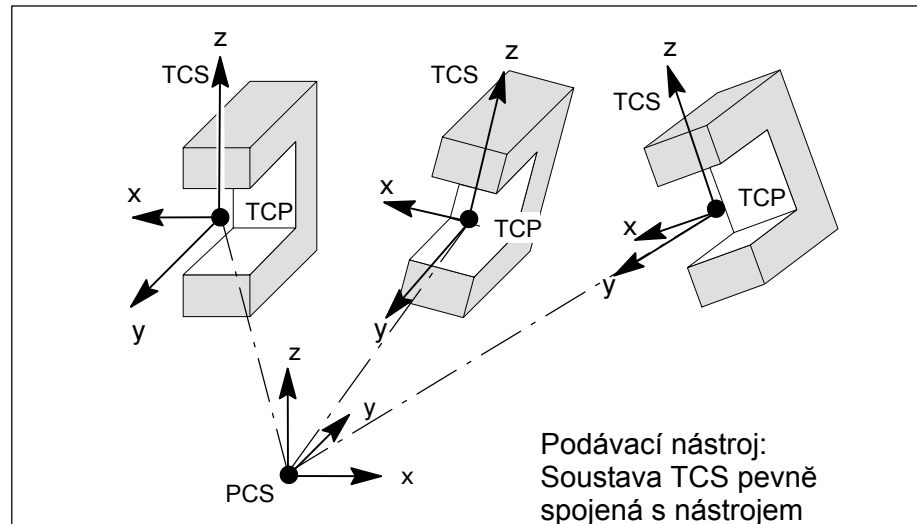
- Počáteční a koncová orientace orientačního vektoru nesmí být kromě programování ROTAX(..) probíhat paralelně ani antiparalelně.

O(), ROTAX() phi, theta, psi

4.48.2 Tenzorová orientace

Následek

Ovlivňování orientace nástroje se provádí orientací celé nástrojové soustavy souřadnic TCS, vzhledem k aktuální soustavě souřadnic PCS.



- Programování je alternativně možné pomocí
 - Eulerových úhlů phi φ , theta ϑ a psi ψ .
Postupným otáčením s Eulerovými úhly získává TCS požadovanou orientaci.
 - Orientační tenzor 3x3 (matice otáčení, která nově vyrovnává celou soustavu TCS kolem TCP). Složky sloupcových vektorů (O_x , O_y , O_z) přesně určují orientaci TCS a jsou programovatelné jako polární úhly nebo kartézské složky.
 - Funkce ROTAX(..) pro definici osy otáčení, orientované libovolně v prostoru, okolo které lze otáčet orientační tenzor pomocí funkce O(..) o určitý úhel β .
Naprogramování osy otáčení je přitom možné s polárními úhly (φ_u , ϑ_u) stejně jako pomocí kartézských složek (u_x , u_y , u_z).

Platí:

- Tenzorová orientace může být naprogramována pouze společně s transformací os, jejíž 100. místo v typu = 3.
- Orientační pohyb probíhá jako pohyb otočného ukazatele orientačního tenzoru z jeho počáteční do naprogramované koncové orientace.
- Vhodné pouze pro **ne** rotačně symetrické nástroje, protože je zde soustava TCS pevně spojena s nástrojem.

Programování

1. Aktivujte přípustný typ transformace os (100. místo v typu = 3) pomocí "Coord(...)" (viz strana 4-30).
2. Dále použijte uvedenou syntaxi.

O(), ROTAX() phi, theta, psi

Syntaxe:

{<Nr>} {<TCP>} {<phi><φ>} {<theta><θ>} {<psi><ψ>} nebo

{<Nr>} {<TCP>} **Ox**(<φ_{xxOy(<φ_{yy}}

{<Nr>} {<TCP>} **Ox**(<φ_{xxOz(<φ_{zz}}

{<Nr>} {<TCP>} **Oy**(<φ_{yyOz(<φ_{zz}}

{<Nr>} {<TCP>} **Ox**(<O11>,<O21>,<O31>) **Oy**(<O12>,<O22>,<O32>)

nebo

{<Nr>} {<TCP>} **Ox**(<O11>,<O21>,<O31>) **Oz**(<O13>,<O23>,<O33>)

nebo

{<Nr>} {<TCP>} **Oy**(<O12>,<O22>,<O32>) **Oz**(<O13>,<O23>,<O33>)

nebo

{<Nr>} {<TCP>} **ROTAX**(<φ_{uuO(<β>) nebo}

{<Nr>} {<TCP>} **ROTAX**(<u_{xyzO(<β>)}

kde

<Nr>

Číslo věty. Viz strana 2-20.

<TCP-Bew>

Přídavný pohyb TCP.

Syntaxe viz příručka "Funkční popis" pod programováním prostorových souřadnic.

<phi>

Název úhlu, který je zadán v MP 7080 00010[4].
Standardní: phi

<theta>

Název úhlu, který je zadán v MP 7080 00010[5].
Standardní: theta

<psi>

Název úhlu, který je zadán v MP 7080 00010[6].
Standardní: psi

<φ>,<θ>,<ψ>

Absolutní Eulerovy úhly ve stupních.
Programování AC/IC je dovoleno.

Rozsah hodnot: $0^\circ \leq \varphi < 360^\circ$;

$0^\circ \leq \vartheta \leq 180^\circ$;

$0^\circ \leq \psi \leq 360^\circ$.

Hodnoty mimo rozsah se automaticky přepočítávají do příslušného intervalu.

<φ.>,<θ.>

Absolutní úhlové hodnoty ve stupních.

Rozsah hodnot: $0^\circ \leq \varphi. < 360^\circ$;

$0^\circ \leq \vartheta. \leq 180^\circ$.

Pokud je jako θ. naprogramována hodnota mimo rozsah, je automaticky přepočítána do zadaného intervalu.

Ox(<O11>,<O21>,<O31>)

Oy(<O12>,<O22>,<O32>)

Oz(<O13>,<O23>,<O33>)

Ox(<φ_{xx}

Oy(<φ_{yy}

Oz(<φ_{zz}

Orientierung per Funktion Ox(..), Oy(..), Oz(..).
Ox(..) například definuje směr souřadnice x soustavy TCS ve vztažné soustavě souřadnic. Analogické úvahy platí pro Oy(..) a Oz(..).

O(), ROTAX() phi, theta, psi

Směr může být zadán buď odpovídajícími polárními úhly $\langle\varphi_{..}\rangle$ a $\langle\vartheta_{..}\rangle$, nebo pomocí kartézských složek sloupcových vektorů orientačního tenzoru.

Jsou dovoleny pouze absolutní rozměrové údaje.

Hodnoty složek sloupcových vektorů (o..) jsou automaticky normovány na 1.

ROTAX ($\langle\varphi_u\rangle, \langle\vartheta_u\rangle$) Definice osy otáčení pomocí polárních úhlů (φ_u, ϑ_u).

ROTAX ($\langle u_x\rangle, \langle u_y\rangle, \langle u_z\rangle$) Definice osy otáčení pomocí kartézských složek $\langle u_x\rangle, \langle u_y\rangle, \langle u_z\rangle$ v absolutním rozměru. Automatické normování na 1.

O ($\langle\beta\rangle$) Udává inkrementální úhel ve stupních $\langle\beta\rangle$, o který se má otočit orientační tenzor kolem osy otáčení.

Jsou povoleny hodnoty větší než 360 stupňů. Směr otáčení se vybírá pomocí znaménka.

Příklad:

Příklady nastavení:

MP 1030 00110[1] pro transformaci os 2:3333301

MP 7080 00010[1]: x

MP 7080 00010[2]: y

MP 7080 00010[3]: z

MP 7080 00010[4]: phi

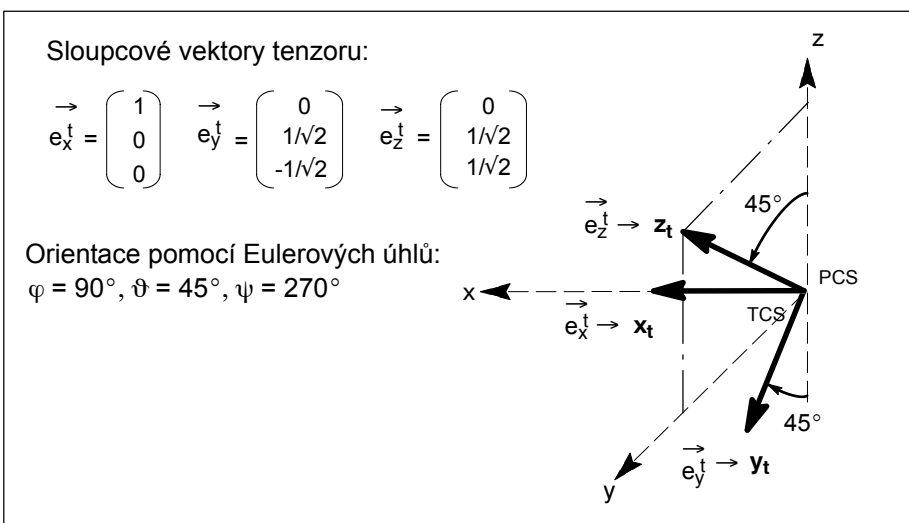
MP 7080 00010[5]: theta

MP 7080 00010[6]: psi

N100 G1 X10 Y20 Z30 Ox(1,0,0) Oy(0,0.707,-0.707) nebo

N100 G1 X10 Y20 Z30 Ox(1,0,0) Oz(0,0.707,0.707) nebo

N100 G1 X10 Y20 Z30 Oy(0,0.707,-0.707) Oz(0,0.707,0.707)



O(), ROTAX() phi, theta, psi

Zvláštnosti a omezení:

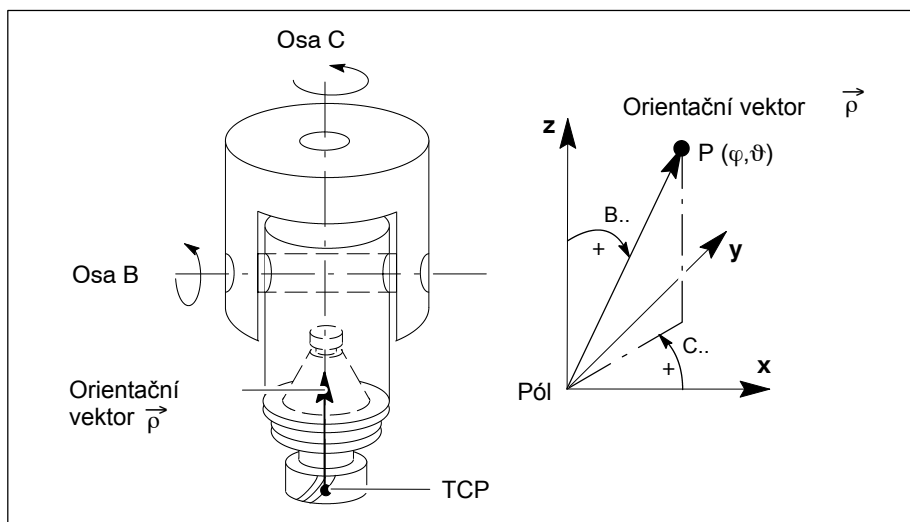
- Orientační pohyb probíhá - s výjimkou programování ROTAX(..) - vždy po nejkratší dráze do koncové orientace.
U ROTAX(..) závisí směr otáčení na znaménku $\langle \beta \rangle$.
- Pro jednoznačné vytvoření orientačního tenzoru jsou nezbytná následující omezení:
Při $\vartheta = 0^\circ$: k určení orientace je zapotřebí součet φ a ψ .
Při $\vartheta = 180^\circ$: k určení orientace je zapotřebí rozdíl φ a ψ .
- Pokud se při programování sloupcových vektorů tenzoru ukáže, že jsou dva sloupcové vektory paralelní nebo antiparalelní, není možné vypočítat orientační tenzor.
Objeví se běhová chyba.

4.48.3 Lineární orientační pohyb s programováním os

 Platí pouze pro transformaci os typu 3032101.

Následek

Při programování orientace nástroje jsou naprogramovány polohové hodnoty dvou kruhových os zasahujících do nástroje (např. B, C).



Platí:

- Jsou dovoleny pouze takové osové kinematiky, u kterých je možné vyobrazit polohové hodnoty obou kruhových os jedna ku jedné v souřadnicích orientace orientačního vektoru (např. phi a theta).
- Orientační pohyb se provádí jako lineární pohyb v kruhových osách.
- Vhodné pro rotačně symetrické nástroje.

O(), ROTAX() phi, theta, psi

Programování

1. Aktivujte transformaci os typu **3032101** pomocí "Coord(...)" (viz strana 4-30).
2. Dále použijte uvedenou syntaxi.

Syntaxe:

```
{<Nr>} {<TCP-Bew>} {<B><Pos>} {<C><Pos>}
```

kde

<Nr>	Číslo věty. Viz strana 2-20.
<TCP-Bew>	Přídavný pohyb TCP. Syntaxe viz příručka "Funkční popis" pod programováním prostorových souřadnic.
, <C>	Osové adresy kruhových os nacházejících se na nástroji
<Pos>	Absolutní osová poloha ve stupních.

Příklad:

Příklady nastavení:

```
MP 1030 00110[1] pro transformaci os 2:3032101
MP 7080 00010[1]: x
MP 7080 00010[2]: y
MP 7080 00010[3]: z
```

N10 G1 X0 Y0 Z0 B0 C0	Naprogramovat logické/fyzické názvy os.
N20 Coord(2)	Zapnout transformaci os 2. Programovatelné jsou nyní lineární souřadnice x, y, z a orientační souřadnice B a C.
N30 x100 y200 z300 B20 C60	Lineární interpolace souřadnic s přídavným orientačním pohybem.
N40 G2 x.. y.. z.. I.. J.. B20 C60	Spirálový pohyb TCP s přídavným orientačním pohybem.
N50 G1 B20 C10	Čistý orientační pohyb. TCP zůstává konstantní.
N60 Coord(0)	Vypnout transformaci os.
:	

Zvláštnosti a omezení:

- Je zapotřebí aktivní transformace os typu 3032101.
- Pomocí speciální logiky vyhledávání dráhy jsou eliminovány rotace kruhových os větší než 180 stupňů.

O(), ROTAX() phi, theta, psi

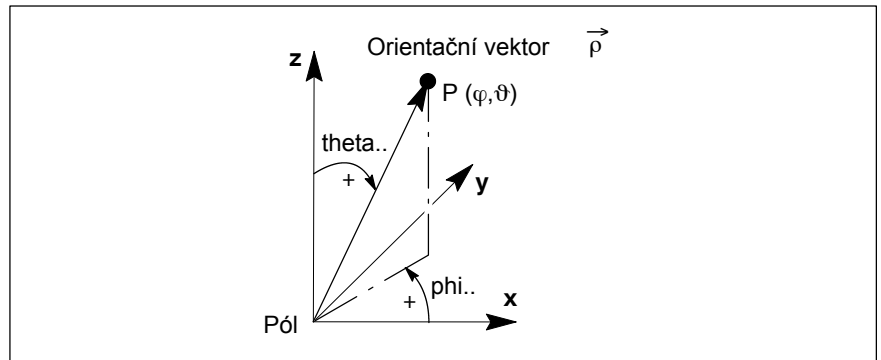
4.48.4 Lineární orientační pohyb s programováním souřadnic

 Platí pouze pro transformaci os typu 3232101.

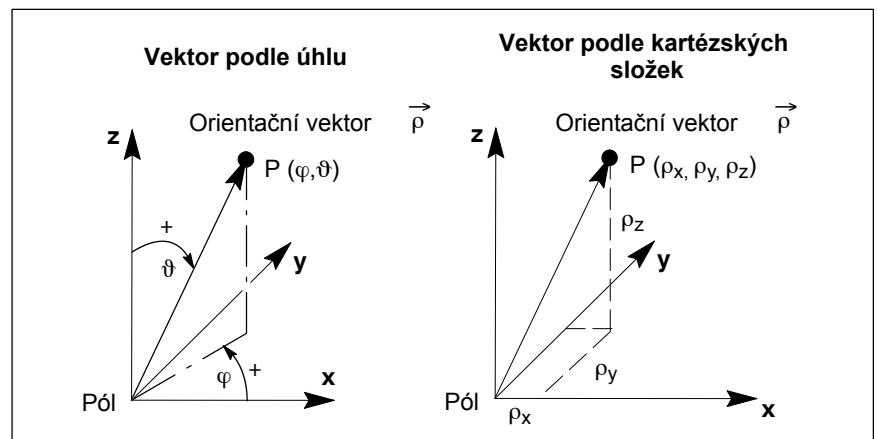
Následek

Programování orientace nástroje je alternativně možné pomocí:

- orientačních souřadnic orientačního vektoru (např. phi a theta).



- Funkce O(..) s polárními úhly (φ, ϑ) nebo kartézskými složkami (ρ_x, ρ_y, ρ_z) orientačního vektoru.



- Základní osová kinematika je libovolná.
- Orientační pohyb probíhá jako lineární interpolace v φ a ϑ , tzn. jako přímka v zamýšlené φ/ϑ -rovině.
- Vhodné pro rotačně symetrické nástroje.

Programování

1. Aktivujte transformaci os typu **3232101** pomocí "Coord(...)" (viz strana 4-30).
2. Dále použijte uvedenou syntaxi.

O(), ROTAX() phi, theta, psi

Syntaxe:

{<Nr>} {<TCP-Bew>} {<phi><φ>} {<theta><θ>} nebo
 {<Nr>} {<TCP-Bew>} O(<φ>, <θ>) nebo
 {<Nr>} {<TCP-Bew>} O(<ρ_{xyz}

kde

<Nr>	Číslo věty. Viz strana 2-20.
<TCP-Bew>	Přídavný pohyb TCP. Syntaxe viz příručka "Funkční popis" pod programováním prostorových souřadnic.
<phi>	Název úhlu, který je zadán v MP 7080 00010[4]. Standardní: phi
<theta>	Název úhlu, který je zadán v MP 7080 00010[5]. Standardní: theta
<φ>, <θ>	Absolutní úhlové hodnoty ve stupních. Rozsah hodnot: $0^\circ \leq \varphi < 360^\circ$; $0^\circ \leq \vartheta \leq 180^\circ$
O(<φ>, <θ>)	Orientace podle funkce O(..) a polárních úhlů <φ> a <θ> orientačního vektoru.
O(<ρ _{xyz <td>Orientace podle funkce O(..) a kartézských složek <ρ_{xyzAutomatické normování na 1. Proto dávají následující příklady údajů totožnou orientaci: O(1,2,4), O(2,4,8)}</td>}	Orientace podle funkce O(..) a kartézských složek <ρ _{xyzAutomatické normování na 1. Proto dávají následující příklady údajů totožnou orientaci: O(1,2,4), O(2,4,8)}

Příklad:

Příklady nastavení:

MP 1030 00110[1] pro transformaci os 2:3232101

MP 7080 00010[1]: x

MP 7080 00010[2]: y

MP 7080 00010[3]: z

MP 7080 00010[4]: phi

MP 7080 00010[5]: theta

N10 G1 X0 Y0 Z0 B0 C0

N20 Coord(2)

Naprogramovat logické/fyzické názvy os.

Zapnout transformaci os 2.

Programovatelné jsou nyní

lineární souřadnice x, y, z a

Souřadnice orientace phi, theta.

N30 x1 y2 z3 phi5 theta5

Lineární interpolace souřadnic s
přídavným orientačním pohybem.

N40 Coord(0)

Vypnout transformaci os.

Zvláštnosti a omezení:

- Je zapotřebí aktivní transformace os typu 3232101.
- Pomocí speciální logiky vyhledávání dráhy jsou eliminovány rotace kruhových os větší než 180 stupňů.
- Programování "ROTAX(... | O(<β>)" není možné.

OvrDis, OVD OvrEna, OVE

4.49 Posuv 100%

**OvrDis, OVD,
OvrEna, OVE**

Následek

Ovlivňuje s programovým řízením účinky potenciometru posuvu na posuv a rychlý chod.
Funkce pracují v provozních režimech "Ruční zadávání" a "Zpracování".

Programování

Syntaxe:

OvrDis Vypnutí potenciometru posuvu.
Posuv se nezávisle na potenciometru posuvu nastaví na 100% naprogramované hodnoty.

Zkratka: OVD

OvrEna Zapnutí potenciometru posuvu.
Posuv je závislý na nastavení potenciometru posuvu.

Zkratka: OVE

Platí:

- Obě funkce působí modálně a vzájemně se ruší.
- Zapnutý stav se dá určit pomocí parametru stroje.
- Obě funkce mohou být naprogramovány také společně s dalšími podmínkami ve stejné větě.

Příklad:

:	Potenciometr posuvu je nastaven na 100%.
N40 OVD G1 X5 Z-2 F200 S100 M4	Deaktivovat potenciometr posuvu.
:	Změnit potenciometr posuvu na 40%, posuv zůstane na 100%.
N80 OVE X100 Y50	Aktivovat potenciometr posuvu. Posuv se změní na 40%.

PathAcc, PAC

4.50 Změna maximálního zrychlení posuvu

PathAcc, PAC

Následek

Snižuje v dílčím programu horní meze pro

- zrychlení posuvu a
- brzdné zrychlení posuvu.

Obě hodnoty zrychlení se přednastavují pomocí parametrů stroje a dají se v dílčím programu přepínat samostatně nebo dohromady.

 **Naprogramované nebo přednastavené zrychlení posuvu může být omezeno maximálním povoleným zrychlením os účastnících se posuvu.**

Programování

Syntaxe:

PathAcc(ACC<Hodnota>)

Společné nastavení zrychlení posuvu a brzdného zrychlení posuvu.

PathAcc({UP<Hodnota1>},{DOWN<Hodnota2>})

Samostatné nastavení zrychlení posuvu a brzdného zrychlení posuvu.

PathAcc()nebo
PathAcc(0)

Znovu aktivovat hodnoty zrychlení z MP 7030 00210 a 7030 00220.

Zkratka: PAC(..)

kde

<Hodnota> Hodnota zrychlení. Podle aktivní měrné jednotky (G71/G70) interpretuje řízení naprogramovanou hodnotu jako "1000 Inch/s₂" nebo "m/s₂".

<Hodnota1> Hodnota pro zrychlení posuvu. Jinak jako <Hodnota>.

<Hodnota2> Hodnota pro brzdné zrychlení posuvu. Jinak jako <Hodnota>.

Zvláštnosti a omezení:

- Maximální programovatelné hodnoty zrychlení jsou vždy omezeny hodnotami parametrů stroje.
- Doporučujeme programovat tuto funkci v samostatné větě.

Příklad:

N30 G71

Zapnout metrické programování.

N40 PAC(UP1.5)

Nastavit zrychlení posuvu na 1,5 m/s².

:

N140 PAC(ACC5)

Nastavit zrychlení posuvu a brzdné zrychlení posuvu na 5 m/s².

:

N200 PAC(UP3.5,DOWN2)

Nastavit zrychlení posuvu na 3,5 m/s² a brzdné zrychlení posuvu na 2 m/s².

:

N240 PAC()

Nastavit hodnoty zrychlení opět na nastavené hodnoty parametrů stroje.

:

PosDepHSOut, PHS

4.51 Výstup Highspeed závislý na poloze

PosDepHSOut, PHS

Následek

Ovlivňuje stav výstupu Highspeed (výstup HS) v závislosti na naprogramovaném pojezdovém pohybu.

Když **požadovaná poloha NC** vzhledem k začátku nebo konci aktuální věty dosáhne naprogramované hodnoty, výstup HS se v závislosti na naprogramování nastaví nebo resetuje.

Jak dlouho má výstup po výskytu specifikované události zůstat nastavený, se dá naprogramovat.

 **Parametrizace dostupných výstupů HS: pomocí MP 4075 00102.**

Programování

Syntaxe:

PosDepHSOut (<Režim>{,<Vzdálenost>}{,<Trvání>})

Zkratka: **PHS** (. .)

kde

- <Režim> Požadovaný způsob působení funkce.
Zadané hodnoty: 0, 1 nebo -1.
- 0 Ukládá modálně data naprogramovaná pomocí <vzdálenosti>, resp. <trvání>. Aktuální stav výstupu HS se přitom nemění.
 - 1 Ukládá modálně data naprogramovaná pomocí <vzdálenosti>, resp. <trvání>. a nastavuje výstup HS. Pro nenaprogramované parametry platí příslušné modální datové hodnoty.
 - 1 Resetuje výstup HS. Případně naprogramovaná <vzdálenost> působí pouze lokálně pro tuto větu. Případně naprogramované <trvání> je ignorováno.
- <Vzdálenost> Vzdálenost od začátku/konce věty, při jejímž dosažení má být nastaven signál (v mm, resp. Inch).
Standardní hodnota: 0
- 0: Na konci věty
 - Větší než 0: Vzdálenost od začátku věty
 - Menší než 0: Vzdálenost od konce věty.
- <Trvání> Maximální doba zapnutí výstupu HS (v ms).
Rozsah hodnot: 0.5 ... 10000.0
Naprogramované hodnoty jsou interně zaokrouhlovány na nejbližší vyšší celočíselné násobky doby NC-cyklu.


PosDepHSOut, PHS

Příklady:

:	
N05 G71	
N10 PHS(0,-1.2,40)	Zkonfigurovat funkci.
:	Při následném vyvolání pomocí syntaxe
:	PHS(1) bude výstup HS nastaven cca
:	1.2 mm před dosažením koncového
:	bodu cca na 40 ms.
N210 G1 G91 F1000	
N220 X10 Y23 PHS(1)	Nastavit výstup HS podle konfigurace v
:	N10.
:	
N330 X10 PHS(1,0.1,900)	0.1 mm za počáteční polohou nastavit
:	výstup HS max. na 900 ms.
N350 X20 PHS(-1,-0.3)	0.3 mm před koncovou polohou reseto-
:	vat výstup HS.

Zvláštnosti a omezení:

- V každém kanálu je podporován 1 výstup HS.
- Funkce působí pouze na pojezdový pohyb, který je naprogramován ve stejné větě.
- Nastavený výstup HS se resetuje teprve po uplynutí doby zapnutí. Základní nastavení nemá na tento okamžik vliv.
- <Vzdálenost> se vztahuje vždy k aktuální požadované poloze z hlediska NC.
Systémově podmíněná zpoždění (např. vyvolaná funkcemi pohonu nebo doběhem os) přitom nejsou brána v úvahu.
- Pokud se během doby zapnutí vyskytne nová úloha PHS, vymaže NC ještě aktivní úlohu a provede novou.

 **Jestliže při dvou po sobě následujících úlohách PHS nedojde ke změně signálu na výstupu HS, nemůže být nová úloha zpravidla rozpoznána externím hardwarem.**

PmTSEL, PMS

4.52 Aktivace tabulek umístění

PmTSEL, PMS

Následek

Aktivuje tabulky umístění (tabulky oprav umístění "šikmé roviny"). Tyto tabulky oprav jsou uloženy jako soubory XML v systému souborů řízení.

Programování

Syntaxe:

PmTSEL ({<Cesta>}<Název souboru>)

Zkratka: **PMS** (. .)

kde

<Cesta> Volitelné zadání cesty k adresáři, v němž je uložen <název souboru>.

Bez zadání probíhá hledání v cestě "/database".

Pokud tam <název souboru> neexistuje, použije řízení vyhledávací cestu pro podprogramy a hledá <název souboru> také v jiných adresářích.

<Název souboru> Název souboru tabulky umístění včetně přípony souboru.

Tabulky se standardním názvem (PM<číslo>.pmt) mohou být aktivovány přímo pomocí čísla, např.

PmTSEL(3) aktivuje tabulku PM3.pmt.

 **Pokyny pro vytváření a úpravy tabulek umístění naleznete v návodu k obsluze řízení!**

PolarPol, POP

4.53 Programování polárních souřadnic

4.53.1 Definice pólu

PolarPol, POP

Následek

- Definuje počátek polární soustavy souřadnic vzhledem k aktivní rovině aktivní programové soustavy souřadnic.
- Nastavuje polární úhel 1 pro programování polárních souřadnic (standardní adresa: A; definovatelná v MP 8005 00001) na 0 stupňů.
- Nastavuje polární úhel 2 pro programování polárních souřadnic (standardní adresa: B; definovatelná v MP 8005 00002) na 90 stupňů.

Programování

Syntaxe:

PolarPol(<HKWert>,<NKWert>)

Určení pólu.

PolarPol(ACTPOS)

Nastavení pólu do aktuální polohy.

PolarPol()nebo
PolarPol(0)

Nastavení pólu na souřadnici
0,0.

Zkratka: POP(..)

kde

<HKWert> Hodnota pólu pro hlavní souřadnici.

<NKWert> Hodnota pólu pro vedlejší souřadnici.

Zvláštnosti a omezení:

- Pokud nepoužijete "PolarPol", používá NC jako pól počátek aktivní programové soustavy souřadnic.
- Naprogramovaný pól zůstává účinný pro aktuální rovinu pouze do dalšího přepnutí roviny (viz strana 3-28).

Příklad:

:

N10 G18

Přepnutí na rovinu ZX se Z jako hlavní a X jako vedlejší souřadnicí.

:

N30 G0 X10 Z25

Polohovací pohyb v rychlém chodu na kartézské souřadnice X10 a Z25.

:

N40 POP(ACTPOS)

Pro definici pólu jsou převzaty aktuální polohové hodnoty (HKWert=25 NKWert=10).

:

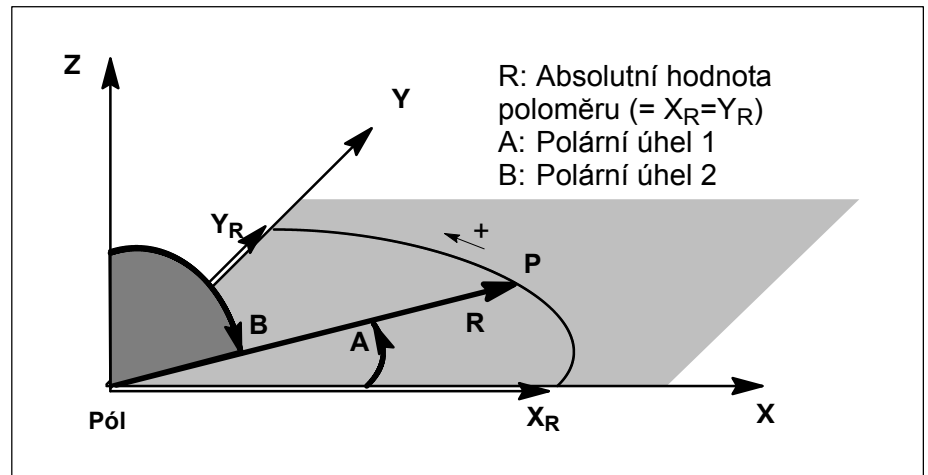
PolarPol, POP

4.53.2 Programování polárních souřadnic

IndraMotion MTX umožňuje programovat polární souřadnice jak v rovině (2D), tak i v prostoru (3D).

Požadovaný koncový bod je zadán pomocí

- absolutní hodnoty poloměru (vzdálenost cílového bodu P od pólu)
- Polární úhel 1 a
- polární úhel 2.



Příklad: Polární souřadnice v rovině při aktivní funkci G17

Absolutní hodnota poloměru:

Programuje se vždy pomocí osové adresy aktuální pracovní roviny (např. "X" **nebo** "Y" při aktivní funkci G17). Tato naprogramovaná souřadnice se označuje také jako souřadnice poloměru.

Udává při programování absolutních rozměrů vzdálenost cílového bodu od pólu a při inkrementálním programování vzdálenost cílového bodu od počátečního.

Jsou-li obě možné osové adresy naprogramovány v jediné větě, objeví se běhová chyba.

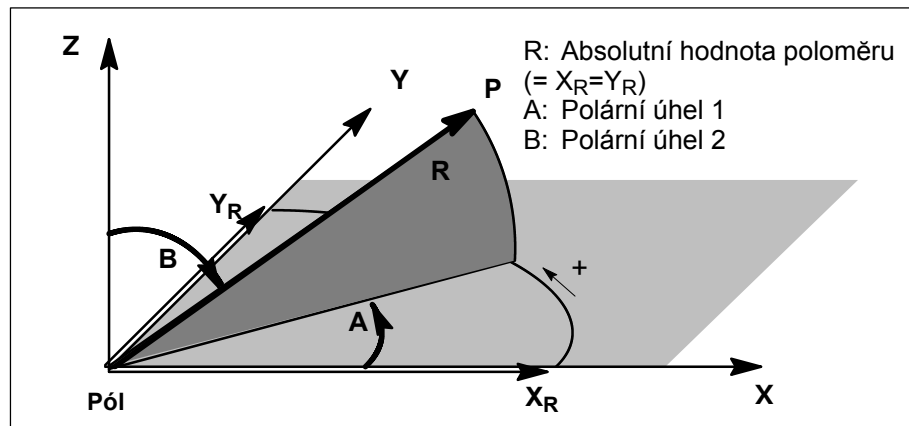
Polární úhel 1:

- Standardně se programuje jako adresa "A" a leží vždy v aktuální pracovní rovině.
- Naprogramovaný polární úhel 1 se vždy vztahuje k **naposledy naprogramované** souřadnicové ose poloměru. Kladná hodnota znamená otáčení v matematicky kladném směru a zůstává účinná do té doby, než je naprogramována nová hodnota. Při aktivním inkrementálním programování se naprogramovaný úhel inkrementálně vypočítává.
- Při aktivaci programování polárních souřadnic s POL se polární úhel 1 nastavuje na hodnotu 0 stupňů.

PolarPol, POP

Polární úhel 2:

- Standardně se programuje jako adresa "B" a při aktivním programování absolutních rozměrů popisuje úhel mezi normálovou osou (k aktuální pracovní rovině) a vektorem z aktuálního pólu do cílového bodu. Zůstává účinný tak dlouho, dokud není naprogramována nová hodnota.
Při aktivním inkrementálním programování se naprogramovaný úhel inkrementálně vypočítává.
- Při aktivaci programování polárních souřadnic s POL se polární úhel 2 nastavuje na hodnotu 90 stupňů.
V takovém případě se cílový bod nachází přesně v aktivní pracovní rovině. Nyní působí polární souřadnice pouze v rovině. Všechny ostatní souřadnice mimo pracovní rovinu jsou interpretovány jako kartézské. To platí obzvlášť také pro naprogramované hodnoty souřadnic aktuální normálové osy!
- "Skutečné" prostorové polární souřadnice (sférické souřadnice) vzniknou, když je polární úhel 2 různý od 90 stupňů. Naprogramování hodnot souřadnic aktuální normálové osy pak vede k běhové chybě. Všechny souřadnice vně 3D prostoru však smí být současně naprogramovány. Jako ve 2D případě jsou i zde interpretovány jako kartézské.
- Při polárním úhlu 2 rovném 0 stupňů nehraje hodnota polárního úhlu 1 žádnou roli, protože naprogramovaný bod leží přesně kolmo nad pólem.



Příklad: Polární souřadnice v prostoru při aktivní funkci G17

- ☞ Standardní adresy pro polární úhly 1 a 2 se dají změnit pomocí parametrů stroje 8005 0001 a 8005 0002. Mohou se proto ve vašem zařízení nazývat jinak.

PolarPol, POP

Příklad: Programování polárních souřadnic v absolutních rozměrech

N10 G18

Přepnutí roviny. Zde: Přepnutí na rovinu ZX se Z jako hlavní a X jako vedlejší souřadnicí.

:

:

N40 POP(25,10)

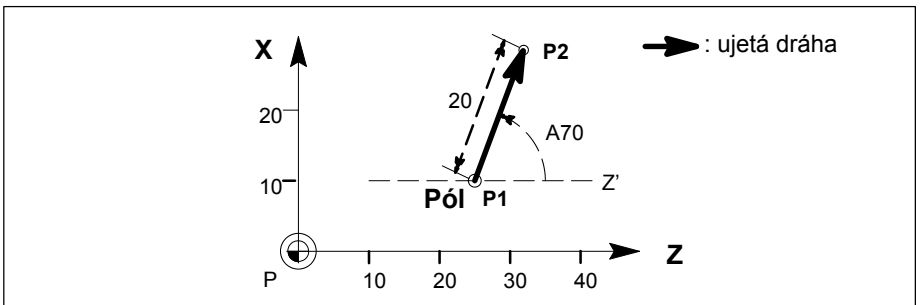
Definice pólu v aktuální rovině (zde: s kartézskými souřadnicemi Z=25 a X=10).

:

:

N50 G1(POL) Z20 A70

Posuvový pohyb v polárních souřadnicích do P2 (viz obr.): Bod P2 je definován absolutní vzdáleností od pólu (zde: 20 mm) a polárním úhlem 1 (zde: A=70 stupňů) vzhledem k naprogramované ose poloměru (zde: osa Z).



N60 G1(POL) X20 A90

Posuvový pohyb v polárních souřadnicích do P3 (viz obr.):

:

:

:

:

:

:

:

:

:

:

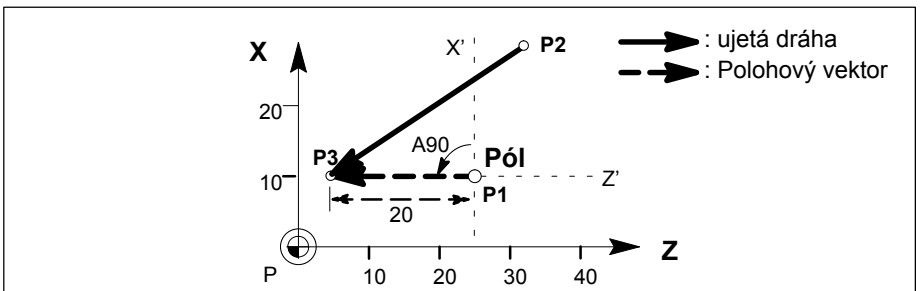
Bod P3 je definován absolutní vzdáleností od pólu (zde: 20 mm) a polárním úhlem 1 (zde: A=90 stupňů) vzhledem k naprogramované ose poloměru (zde: osa X).

Alternativně by bylo možné také následující naprogramování:

N60 G1(POL) Z20 A180

nebo

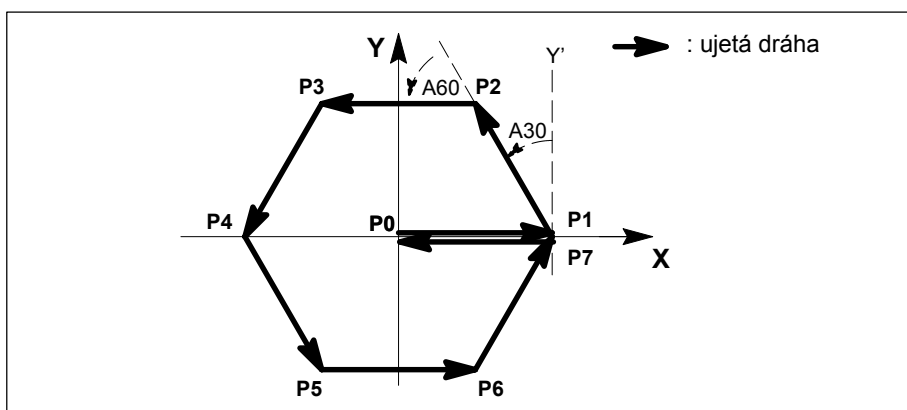
N60 Z20 A180



PolarPol, POP

Příklad: Programování polárních souřadnic v inkrementálních rozměrech

```
N10 G17 G90 G0 X0 Y0 Počáteční poloha: P0
N20 G1(POL) X200 A0 P1
N30 G91 Y200 A30 P2
N40 A60 P3
N50 A60 P4
N60 A60 P5
N70 A60 P6
N80 A60 P7
N90 X0 P0
N100 M30
```




PoleSet, PLS

4.54 Definice bodu zrcadlového převrácení / otáčení PoleSet, PLS

Následek

Tato funkce určuje absolutní polohu pólu pro pomůcky zrcadlové převrácení, nastavení měřítka a otáčení. Definovaný pól je pak vztažným bodem těchto pomůcek a představuje tedy bod zrcadlového převrácení / otáčení.

 **Funkce není zapotřebí, když má být zrcadlové převrácení nebo otočení provedeno vzhledem k programovému nulovému bodu.**

Platí:

- Funkce PoleSet je modálně účinná. Poloha pólu se vztahuje k aktuální programové soustavě souřadnic a zůstává zachována tak dlouho, dokud není resetována na počátek programové soustavy souřadnic nebo nově definována.
- Funkce nezpůsobuje žádný pojezd os.
- Funkce smí být naprogramována s jinými podmínkami a pomocnými funkcemi ve stejné větě.
- Naprogramovaný pól se vztahuje k funkcím zrcadlového převrácení (Mirror), nastavení měřítka (Scale) a otáčení (Rotate).

Programování

Syntaxe:

PoleSet(<Souřadnice>)

Nastavení pólu na zadané souřadnice. Zadané souřadnice musí být odděleny čárkami
např.: PoleSet (X5 , Y2).

PoleSet(0) nebo
PoleSet() nebo

Reset pólu do počátku programové soustavy souřadnic.

Zkratka: PLS(..)

Příklad:

```
:
N30 PLS ( X5 , Y2 )
:
N130 PLS ( )
:
```

Nastavit bod zrcadlového převrácení / otáčení do polohy X=5 a Y=2.
Resetovat bod zrcadlového převrácení / otáčení do počátku programové soustavy souřadnic.

Zvláštnosti a omezení:

- Poloha pólu musí být zadána jako absolutní, tzn. vzhledem k aktuálnímu programovému nulovému bodu.

PosMode, PMD DC ACP ACN

4.55 Typ polohování pro nekonečné osy Lokální typ polohování pro nekonečné osy

PosMode, PMD
DC, ACP, ACN

Následek

Určuje, ve kterém směru se mají otáčet osy typu "Endlos" při polohovací proceduře.

Platí:

- DC(...), ACP(...) a ACN(...) působí na rozdíl od "PosMode" po větách a překrývají v aktuální větě aktivní typ polohování příslušné naprogramované osy.
- Polohové údaje, které jsou naprogramovány ve spojení s funkcemi DC, ACP, resp. ACN, jsou vždy interpretovány jako absolutní polohové hodnoty.

Programování

Syntaxe:

PosMode(<Adr><Typ>) Aktivovat typ polohování pro osu s adresou <Adr> podle <typu>. V závorkách smí být naprogramováno několik os. Pro nenaprogramované kruhové/nekonečné osy v aktuálním kanálu aktivovat typ polohování podle MP 1003 00005.

PosMode() nebo PosMode(0) Aktivovat typ polohování všech os v aktuálním kanálu podle MP 1003 00005.

Zkratka: PMD(..)

<Adr>=DC(<Hodnota>) Osa <Adr> pojíždí do absolutní polohy <Hodnota> po nejkratší dráze.

<Adr>=ACP(<Hodnota>) Osa <Adr> pojíždí do absolutní polohy <Hodnota> v matematicky kladném směru (viz následující upozornění).

<Adr>=ACN(<Hodnota>) Osa <Adr> pojíždí do absolutní polohy <Hodnota> v matematicky záporném směru.

kde

<Adr>: Osová adresa.
Jsou povoleny nekonečné osy (typu "Endlos").

<Hodnota>: Informace o dráze pro <Adr>.

<Typ>: Typ polohování

- 0: Žádná polohovací logika.
Směr pojezdu vyplývá vždy výhradně z rozdílu staré a nové polohy.
- 1: Nejkratší dráha.
Maximální dráha pojezdu osy je nejvýš polovina příslušné modulu hodnoty.
- 2: Podle naprogramované znaménka:
"+": otáčení doprava; "-": otáčení doleva.

PosMode, PMD DC ACP ACN

 **Matematicky kladný směr: Směr otáčení proti směru hodinových ručiček s ohledem na souřadnicovou osu ve směru počátku soustavy souřadnic.**

Příklad:

N40 B=ACP(-258)

:

:

:

N80 PMD(A1C2)

:

:

:

Nezávisle na "PosMode" pojíždí fyzická osa s označením "B" v matematicky kladném směru do polohy 258 stupňů. Znaménko je ignorováno.

Typ polohování

osy A: nejkratší dráha

osy B: podle MP 1003 00005

osy C: znaménko

Zvláštnosti a omezení:

- Ve stejné větě může být naprogramováno několik funkcí DC, ACP, resp. ACN (vždy pro odlišné osy).
- Funkce DC, ACP, resp. ACN jsou účinné pouze pro synchronní osy typu "Rund" (kruhové) nebo "Endlos" (nekonečné). Pokud je naprogramujete ve spojení s jinými typy os, jsou řízením ignorovány.
- Záporné znaménko v polohových údajích funkcí DC, ACP, resp. ACN je ignorováno.

PrecProg, PRP

4.56 Přesné programování

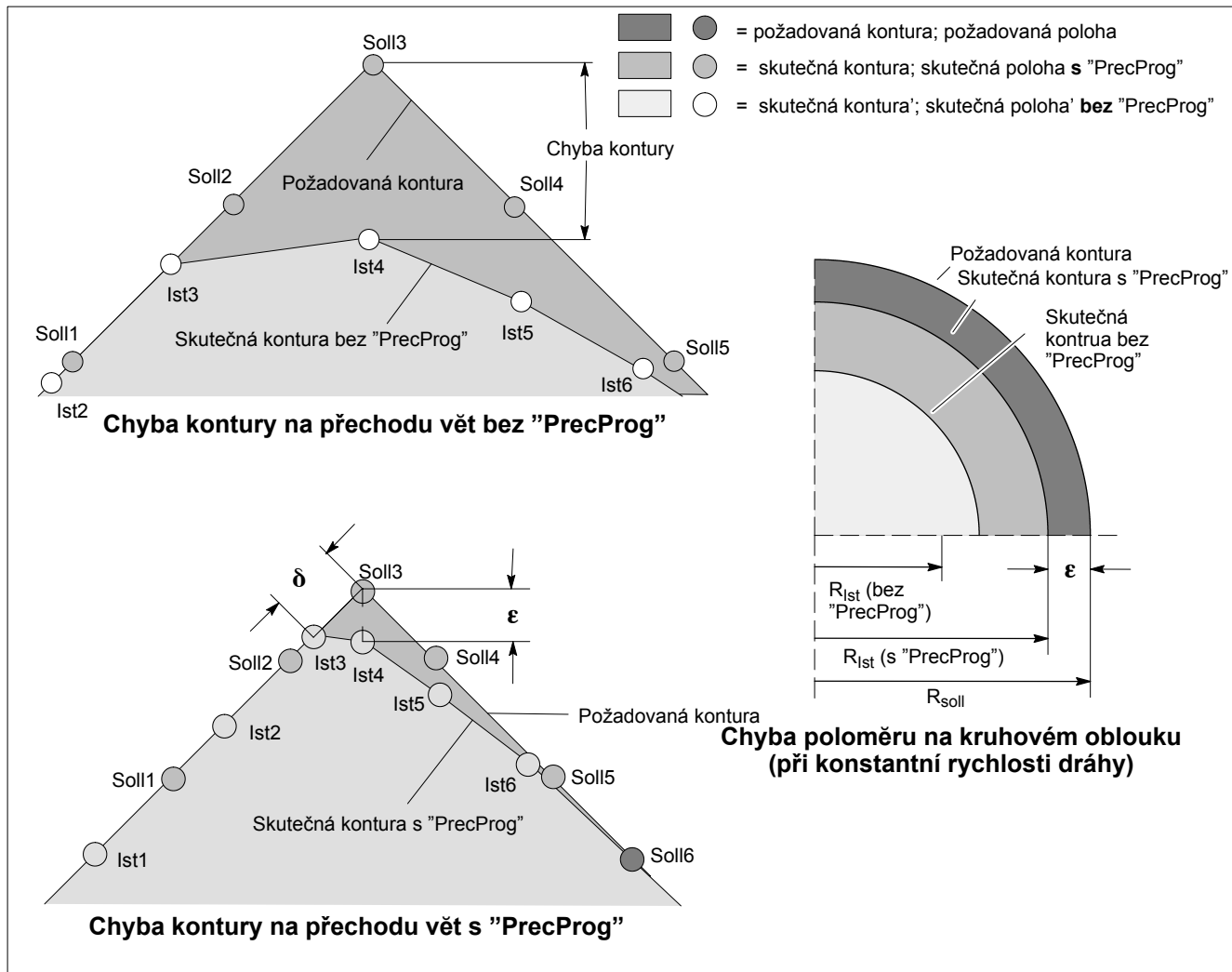
PrecProg, PRP

Následek

Snižuje posuv na konturových přechodech a na kruhových segmentech dráhy tak, aby byla dodržena předepsaná přesnost.

Požadovaná přesnost se dá alternativně ovlivňovat zadáním

- maximální přípustné chyby **kontury nebo poloměru** ϵ na konturovém přechodu či u kruhového oblouku, nebo
- maximálního přípustného **doběhu dráhy** δ (vzdálenost rohů), který nesmí být při projíždění konturového přechodu překročen.



☞ Na rozdíl od funkce "Přesné zastavení" (G61/G62; viz strana 3-49) neprobíhá na přechodu vět nutně zabrzdění na rychlost 0. Viz "Zvláštnosti a omezení".

PrecProg, PRP

Programování

Syntaxe:

PrecProg nebo
PrecProg(1)Zapnutí přesného programování.
Max. přípustná chyba kontury/poloměru
 ε podle MP 8003 00001.PrecProg(EPS< ε >)Zapnutí přesného programování.
Maximální přípustná chyba kontury/po-
loměru: < ε >.PrecProg(DIST< δ >)Zapnutí přesného programování.
Maximální přípustný doběh dráhy: < δ >.
Pro kruhové segmenty dráhy: vezměte v
úvahu hodnotu ε z MP 8003 00001.PrecProg() nebo
PrecProg(0)

Vypnutí přesného programování.

Zkratka: PRP(..)

kde

< ε >: Maximální přípustná chyba kontury/poloměru v mm, resp..
Inch (v závislosti na G70/G71).< δ >: Maximální přípustný doběh dráhy v mm, resp. Inch (v závis-
losti na G70/G71).**Zvláštnosti a omezení:**

- Všechny příslušné osy musí být nastaveny se stejnou dynamikou.
- Pokud funkce G8 a G62 nejsou aktivní, proběhne na každém přechodu vět zabrzdění na $v=0$.
- Programování přesného zastavení dává rozumné výsledky jedině tehdy, když se zanedbá vliv zrychlení na doběh.

PtBlkEnd, PTE

4.57 Čas spuštění zdvihu (koncový bod interpolace) PtBlkEnd, PTE

Následek

- ☞ **Působí v souvislosti s funkcemi "děrování" (viz strana 4-106) a "vysekávání" (viz strana 4-74).**

Funkce

- definuje pro zadané osy časově referenční událost "NC-interpolátor dosahuje koncového bodu pojezdového pohybu" a
- určuje časový interval mezi časově referenční událostí a spuštěním zdvihu.

Tímto způsobem je možné libovolně posouvat okamžik spuštění zdvihu vzhledem k časové referenci. Možnosti jsou

- urychlené spuštění zdvihu
(např. pro kompenzaci konstantní čekací doby závislé na aplikaci, která je způsobena zpracováním signálu) a
- zpožděné spuštění zdvihu
(např. pro zvýšení přesnosti polohování u os s nízkou dynamikou, nebo při děrování s přidržovači).

- ☞ **Další funkce, které mají vliv na čas spuštění zdvihu:
PtBlkEnd (viz strana 4-102)
PtInpos (viz strana 4-105).**

Programování

Syntaxe:

PtBlkEnd (<Osa><Čas>{,<Osa><Čas>}...)

Zkratka: **PTE** (. .)

kde

<Osa> Logický název osy.

<Čas> Požadovaný časový interval (v ms) mezi časově referenční událostí a spuštěním zdvihu.

0: Spuštění zdvihu v okamžiku události.

Záporná hodnota: urychlené spuštění zdvihu

Kladná hodnota: zpožděné spuštění zdvihu

Zvláštnosti a omezení:

- Funkce PTD, PTE a PTI jsou modálně účinné a vzájemně se ruší.
- Děrování nebo vysekávání musí být aplikováno v MP 8001 00010.
- Všechny naprogramované časy jsou zaokrouhleny na násobky doby cyklu SERCOS.

PtBlkEnd, PTE

- Při pojezdových pohybech, na kterých je zúčastněno několik os s různě definovanými časovými referencemi a čekacími dobami, určuje "nejslabší" osa skutečné chování při spuštění zdvihu. Přitom platí:
 - Osy s časovou referencí "okno Inpos" (viz strana 4-105) jsou "slabší" než osy s časovou referencí "koncový bod interpolace".
 - Při stejné časové referenci je "nejslabší" osa s nejdelší čekací dobou.
- Pomocí funkce PtDefault (viz strana 4-104) se dá spuštění zdvihu všech os v systému nastavit opět na standardní hodnoty.

Příklad:

- Osová konfigurace: X, Y, C
- Doba cyklu SERCOS: 3 ms

Nastavení časů spuštění zdvihu:

```

:
N10 PTE(X-10) Časově referenční událost pro osu X:
: "NC-interpolátor dosahuje koncový bod pojezd-
: dového pohybu".
: Urychlené spuštění zdvihu o 10 ms.
N20 PTI(Y10,C2) Časově referenční událost pro osy Y a C: "Do-
: saženo okno Inpos"
: (funkce viz strana 4-105).
: Osa Y: Zpožděné spuštění zdvihu o 10 ms.
: Osa C: Zpožděné spuštění zdvihu o 2 ms.

```

Účinky během pojezdových pohybů:

```

:
N100 G1 Y20 C10 Y určuje chování při spuštění zdvihu, protože při
: stejné časové referenci byla naprogramována delší
: čekací doba (viz N20).
: Protože se čekací doby zaokrouhlují na násobky
: doby cyklu SERCOS, dostáváme skutečné
: zpožděné spuštění zdvihu o 12 ms.
:
N110 X20 C20 C určuje chování při spuštění zdvihu, protože osy s
: časovou referencí "okno Inpos" jsou "slabší" než
: osy s časovou referencí "koncový bod interpolace".
: Protože se čekací doby zaokrouhlují na násobky
: doby cyklu SERCOS, dostáváme skutečné
: zpožděné spuštění zdvihu o 3 ms.
:
:
N120 X30 X určuje chování při spuštění zdvihu, protože
: pouze osa X se pohybuje.
: Skutečné urychlené spuštění zdvihu: -9 ms.
N130 PTD Nastavit spuštění zdvihu všech os v systému podle
: MP 8001 00020 a MP 8001 00021
: (funkce viz strana 4-104).
:

```

PtDefault, PTD

4.58 Čas spuštění zdvihu (nastavení na standardní hodnotu) PtDefault, PTD

Následek

- ☞ **Působí v souvislosti s funkcemi "děrování" (viz strana 4-106) a "vysekávání" (viz strana 4-74).**

Nastavuje okamžik spuštění zdvihu všech os v systému na hodnoty definované v MP 8001 00020 a MP 8001 00021.

Programování

Syntaxe:

PtDefault

Zkratka: **PTD**

Zvláštnosti a omezení:

- Funkce PTD, PTE a PTI jsou modálně účinné a vzájemně se ruší.
- Děrování nebo vysekávání musí být aplikováno v MP 8001 00010.

Příklad:

viz strana 4-103.

- ☞ **Další funkce, které mají vliv na čas spuštění zdvihu:
PtBlkEnd (viz strana 4-102)
PtInpos (viz strana 4-105).**

PtInpos, PTI

4.59 Čas spuštění zdvihu (okno Inpos)

PtInpos, PTI

Následek

- ☞ **Působí v souvislosti s funkcemi "děrování" (viz strana 4-106) a "vysekávání" (viz strana 4-74).**

Funkce

- definuje pro zadané osy časově referenční událost "Okno Inpos dosaženo" a
- určuje časový interval mezi časově referenční událostí a spuštěním zdvihu.

Tímto způsobem je možné libovolně zpozdít okamžik spuštění zdvihu vzhledem k časové referenci (např. pro zvýšení přesnosti polohování u os s nízkou dynamikou nebo při děrování s přidržovači).

- ☞ **Další funkce, které mají vliv na čas spuštění zdvihu: PtBlkEnd (viz strana 4-102) PtInpos (viz strana 4-105).**

Programování

Syntaxe:

PtInpos (<Osa><Čas>{,<Osa><Čas>}...)

Zkratka: **PTI** (. .)

kde

<Osa> Logický název osy.

<Čas> Požadovaný časový interval (v ms) mezi časově referenční událostí a spuštěním zdvihu.

Zvláštnosti a omezení:

- Funkce PTD, PTE a PTI jsou modálně účinné a vzájemně se ruší.
- Děrování nebo vysekávání musí být aplikováno v MP 8001 00010.
- Všechny naprogramované časy jsou zaokrouhleny na násobky doby cyklu SERCOS.
- Při pojezdových pohybech, na kterých je zúčastněno několik os s různě definovanými časovými referencemi a čekacími dobami, určuje "nejslabší" osa skutečné chování při spuštění zdvihu. Přitom platí:
 - Osy s časovou referencí "okno Inpos" (viz strana 4-105) jsou "slabší" než osy s časovou referencí "koncový bod interpolace".
 - Při stejné časové referenci je "nejslabší" osa s nejdelší čekací dobou.
- Pomocí funkce PtDefault (viz strana 4-104) se dá spuštění zdvihu všech os v systému nastavit opět na standardní hodnoty.

Příklad viz kap. 4.57.

Punch, PUN

4.60 Děrování

Punch, PUN


Následek

Zapíná nebo vypíná funkci "děrování".

Při aktivovaném děrování se

- na konci každého naprogramovaného pojezdového pohybu (provoz s jednotlivými zdvihy), nebo
- na konci každé dílčí dráhy vytvořené funkcí "NUM" (viz strana 4-76) nebo "LEN" (viz strana 4-62)

spouští zdvih. Výsledný pojezdový pohyb začíná vždy teprve po skončení zdvihu.

 **Funkce pro ovlivnění času spuštění zdvihu:**
PtDefault (viz strana 4-104)
PtBlkEnd (viz strana 4-102)
PtInpos (viz strana 4-105).

Programování

Syntaxe:

Punch(1) nebo
Punch

Zapnout děrování.

Punch(0) nebo
Punch()

Vypnout děrování.

Zkratka: **PUN (. .)**

Zvláštnosti a omezení:

- Děrování musí být uvolněno parametrem MP 8001 00010.
- Funkce působí modálně a již na pojezdové pohyby, které jsou naprogramovány ve stejné větě.
- Pomocí funkce "NUM" nebo "LEN" může řízení z naprogramované pojezdové dráhy automaticky generovat dílčí dráhy, na jejichž konci je vždy spuštěn zdvih.
- Je-li zapnuto vysekávání (viz strana 4-74), ruší se volba funkce děrování.
- Věty, které neobsahují žádnou osovou souřadnici z aktivní roviny, nespouštějí také žádný zdvih.
- Pokud SPS potlačí spuštění zdvihů, zůstane obrábění stát v poloze spuštění zdvihu, dokud SPS spuštění zdvihů znovu neuvolní.

Punch, PUN

Příklad:

- G90 je aktivní (programování absolutních rozměrů)
- Aktivní rovina: X/Y
- Aktuální poloha: X=0, Y=0, C=0

:

N10 C10 Punch(1)

Zapnout děrování. Osa C se otočí na 10 stupňů. Žádný zdvih, protože osy X a Y nejsou naprogramovány.

N20 C60

Osa C se otočí na 60 stupňů. Žádný zdvih, protože osy X a Y nejsou naprogramovány.

N30 X0

Žádný pojezdový pohyb, protože osa X je již v poloze 0.

N40 LEN=12

Zdvih, protože osa X leží v aktivní rovině.

Rozdělit následující pojezdové věty na stejné dílčí dráhy maximálně po 12 mm.

N50 X110

Pojezdová věta se rozdělí na 10 stejných dílčích drah po 11 mm. Zdvih v polohách X11, X22, X33 ...X99, X110.

N60 Y30 NUM=3

Překrývá modálně působící hodnotu LEN (N40) pro aktuální větu.

Rozdělit pojezdové věty na 3 stejné dílčí dráhy. Zdvih při Y10, Y20, Y30.

N70 Y90

LEN působí opět z N40.

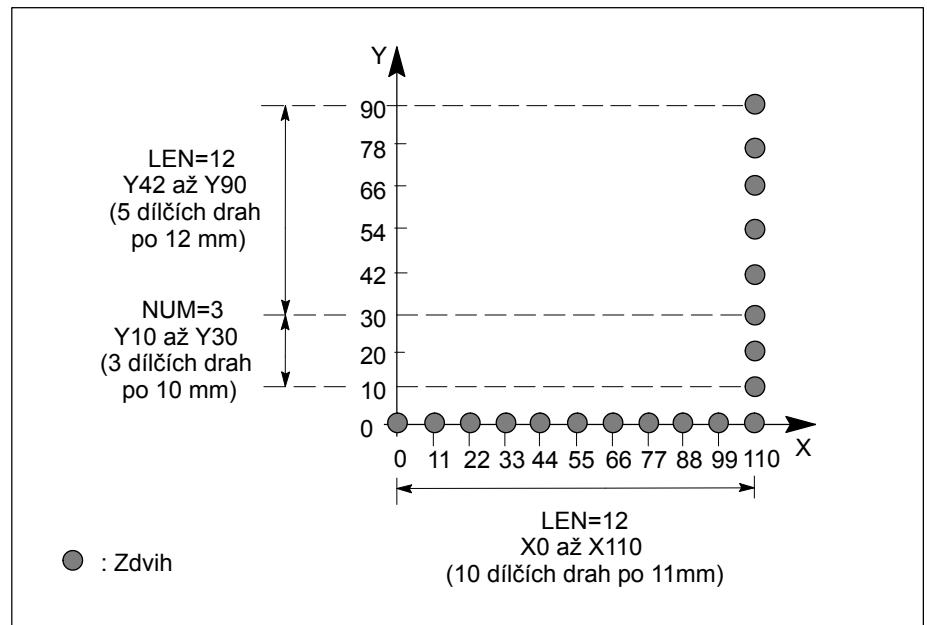
Zdvih při Y42, Y54, Y66, Y78, Y90.

N80 X50 Y50 Punch()

Vypnout děrování.

:

Pojezdový pohyb na X=50, Y=50.



RedTorque, RDT

4.61 Omezení momentu

RedTorque, RDT

Následek

"RedTorque" překrývá pro jednotlivé osy hodnotu parametru stroje 1003 00010 naprogramovanou hodnotou.

Samotná hodnota parametru stroje se tím nemění.

Tímto způsobem se dá snížený maximální moment osy, který může být účinný po kladném impulsu signálu osového rozhraní "Omezení momentu" (qAx_TrqLim), nejen pevně omezit na hodnotu parametru stroje, ale také "dynamicky" nastavovat.

Programování

Syntaxe:

RedTorque(<Osa1>
{, <Osa2>{, ...}})

Nastavit hodnoty pro snížený maximální moment na naprogramované hodnoty.

RedTorque(0) nebo
RedTorque()

Nastavit hodnoty pro snížený maximální moment na hodnoty parametru stroje (MP 1003 00010).

Zkratka: RDT(..)

kde

<Osa i>: Fyzické nebo logické označení osy včetně příslušného maximálního momentu jako % z klidového momentu osy.
Rozsah hodnot: 0 až 500 %.

Příklad:

N8 RDT(X5) S dalším kladným impulsem příslušného osového signálu
: "Omezení momentu" se omezuje maximální moment osy
: (zde X) na 5 % svého klidového momentu.

RemAxis, RAX

4.62 Odstranění osy ze skupiny os

RemAxis, RAX

Následek

Odstraňuje synchronní osu z volajícího kanálu. Ze synchronní osy se přitom stane asynchronní.

Asynchronní osa je potom programovatelná v každém kanálu pomocí svého fyzického osového názvu.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

RemAxis (<PAN> | <PAI> | <LAN>{,<PAN> | <PAI> | <LAN>}...)

Zkratka: **RAX** (. .)

kde

<PAN>	Fyzický název osy. Určuje osu, která má být odstraněna z aktuálního kanálu.
<PAI>	Fyzický index osy. Účinky jako <PAN>.
<LAN>	Logický název osy. Účinky jako <PAN>.

Zvláštnosti a omezení:

- Příprava vět se při odstranění osy nezastavuje.
- Neplatné názvy os způsobují chybové hlášení.
- Je-li v systému definována osa pro odstranění, která ale již v aktuálním kanálu neexistuje, není vydáno žádné chybové hlášení.
- Osové polohy ve stejné větě musí být vždy naprogramovány za výrazem RemAxis(...).

Příklad:

```

:
N030 RAX(XP,2,Z)           Fyzická osa XP, fyzická osa s indexem 2 a
:                           logická osa Z budou odstraněny z kanálu.

```

RemLogName, RLN

4.63 Vymazání logického názvu osy

RemLogName, RLN

Následek

Vymaže logický název synchronní osy ve volajícím kanálu. Osa zůstává v kanálu, ale může zde být programována pouze pomocí svého fyzického názvu nebo fyzického indexu.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

RemLogName (<PAN> | <PAI> | <LAN>{, <PAN> | <PAI> | <LAN>}...)

Zkratka: **RLN** (. .)

kde

<PAN>	Fyzický název osy. Určuje osu, jejíž logický název má být v aktuálním kanálu vymazán.
<PAI>	Fyzický index osy. Účinky jako <PAN>.
<LAN>	Logický název osy. Účinky jako <PAN>.

Zvláštnosti a omezení:

- Zadaná osa musí být zastavená. Pokud tomu tak není, generuje řízení chybové hlášení a přeruší program.
- Osové polohy ve stejné větě musí být vždy naprogramovány za výrazem RemLogName(...).

Příklad:

:
N030 RLN(YP, 3, Z) Logické názvy fyzické osy YP, 3. fyzické osy
: a logické osy Z budou ve volajícím kanálu
 odstraněny.

REPOSTP

4.64 Asynchronní podprogramy: Definice bodu opětovného najetí v asynchronním podprogramu

REPOSTP

Následek

Definuje, jestli má být řízení po skončení asynchronního podprogramu, ve kterém je naprogramována funkce REPOSTP, polohováno na

- počáteční bod
- koncový bod nebo
- bod přerušení

případně přerušené pojezdové věty.

Pokud v okamžiku přerušení nebyla aktivní žádná pojezdová věta, polohuje se řízení vždy na poslední aktivní souřadnice.

REPOSTP přitom překrývá bod opětovného najetí pro právě běžící asynchronní podprogram, nastavený dříve pomocí ASPRTP (viz strana 4-7).

 **Podrobné informace o používání a parametrizaci asynchronních podprogramů naleznete ve funkční příručce.**

Programování

Syntaxe:

REPOSTP (<Bod>)

kde

<Bod>	Požadovaný bod opětovného najetí:
	1: Počáteční bod
	2: Koncový bod
	3: Bod přerušení

Zvláštnosti a omezení:

- Funkce je určena k používání v asynchronním podprogramu.
- Bod opětovného najetí pro běžící asynchronní podprogram, nastavený dříve pomocí ASPRTP (viz strana 4-7), je po skončení podprogramu automaticky opět účinný.
- Mezitím provedené změny oprav v asynchronním podprogramu jsou při interním výpočtu požadovaného bodu opětovného najetí automaticky brány v úvahu.

Rotate(...), ROT(...)

4.65 Pomůcka: otáčení

Rotate(...), ROT(...)

Následek


Funkce otáčení patří k pomůckám.

Řízení otáčí naprogramovanou konturu v aktivní rovině (informace o rovinách viz G17, G18, G19 nebo G20).

Otáčení se vždy vztahuje k aktuálnímu bodu otáčení (viz funkce PoleSet na straně 4-97). Pokud nebyl bod otáčení explicitně naprogramován, je jím aktuální programový nulový bod.

Při použití této funkce musí být opakující se úseky programu, které jsou otočeny o určitý úhel, naprogramovány pouze jednou.

Kromě toho již nemusíte např. přepočítávat rozměry úhlových obrobků na souřadnice základní soustavy souřadnic obrobku; převezměte jednoduše rozměry přímo z výrobního výkresu a zadejte potřebný úhel otočení. O zbytek se postará řízení.

-  **Otáčení je pomůcka, a proto nemění aktuální programovou soustavu souřadnic. Pomůcka pouze představuje jinou možnost zadávání programových souřadnic. Otáčení se může používat také společně se zrcadlovým převrácením a nastavením měřítká.**

Platí:

- Funkce působí modálně. Zůstává účinná tak dlouho, dokud není vypnuta.
- Smí být naprogramována s jinými podmínkami a pomocnými funkcemi ve stejné větě.

Programování

Syntaxe:

Rotate(<Úhel otočení>)

Zapnutí otočení o požadovaný <úhel otočení>.

Rotate(0) nebo
Rotate()

Vypnutí otáčení pro všechny osy v kanálu. Všechny úhly otáčení jsou nastaveny na "0". Najeté osové polohy zůstávají zachovány, dokud nejsou nově naprogramovány.

Zkratka: ROT(..)

kde

<Úhel otáčení> > 0: Otáčení proti směru hodinových ručiček
< 0: Otáčení ve směru hodinových ručiček
= 0: Vypnutí otáčení.

Všechny další naprogramované souřadnice aktivní roviny se otočí kolem bodu otáčení (viz PoleSet na straně 4-97).

Otáčení je účinné až s další pojezdovou informací.

Rotate(...), ROT(...)

Zvláštnosti a omezení:

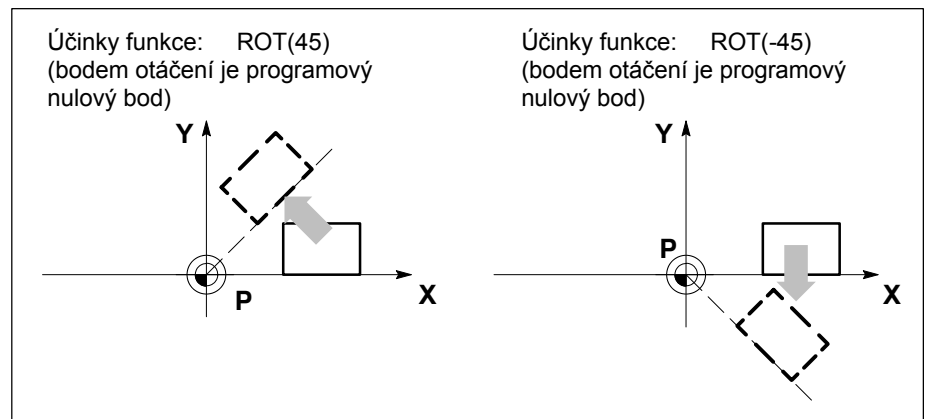
- Funkce bere v úvahu interpolační parametry při kruhové interpolaci.
- Má vliv na programovatelné posunutí kontur (Shift). Viz strana 4-120.
- **Nemá** vliv na:
 - Posunutí nulových bodů (G54-G59.5; viz strana 3-46),
 - Posunutí programových souřadnic (Trans nebo ATrans; viz strana 4-148),
 - Nastavení programových poloh ("SetPos"; viz strana 4-119),
 - Najetí na souřadnice referenčního bodu (G74 ; viz strana 3-55),
 - Najetí do pevné osové polohy stroje (G76 ; viz strana 3-58),
 - Hodnoty oprav poloměru frézy a délky nástroje.

Příklad:

```

:
N30 ROT(45)   Zapnout otáčení. Všechny další naprogramované
:             souřadnice aktivní roviny se otočí o 45 stupňů proti
:             směru hodinových ručiček kolem případně
:             naprogramovaného bodu otáčení.

```

**Otáčení orientačního vektoru**

Orientační vektor se otáčí kolem normály aktivní roviny. Případně aktivní nastavení měřítka nebo bod zrcadlového převrácení / otáčení nemá žádný vliv na výsledek.

Syntaxe je popsána v odstavci "Programování".

RoundEps, RNE

4.66 Zaoblení rohů se zadáním odchyšky

RoundEps, RNE

Následek

Tato funkce vkládá mezi 2 lineární větvy v hlavní rovině tečné přechodové oblouky.

Tím se sice poněkud změní naprogramovaná kontura na těchto rozích, ale pro interpolaci jsou získány spojitě průběhy rychlosti a zrychlení.

Programování

Syntaxe:

RoundEps(<Odchyška>) Zapnout zaoblení rohů mezi 2 lineárními větami.

RoundEps(DEF) Zapnout zaoblení rohů mezi 2 lineárními větami se standardní odchyškou z MP 7050 00110.

RoundEps(0) nebo
RoundEps() Vypnout zaoblení rohů mezi 2 lineárními větami.

Zkratka: RNE(..)

kde

<Odchyška> Maximální přípustná odchyška (v mm) mezi změněnou a naprogramovanou konturou. Desetinná místa jsou povolena.
Řízení vypočítává vhodný tečný přechodový oblouk.

Zvláštnosti a omezení:

- Funkce "ChLength", "ChSection", "RoundEps" a "Rounding" působí modálně a vzájemně se ruší.
- Řízení **neprovede** funkci "zaoblení rohů", když
 - alespoň jedna z obou sousedních vět není lineární věta.
 - alespoň jedna z obou sousedních vět má část dráhy mimo vybranou hlavní rovinu, nebo
 - alespoň jedna z obou sousedních vět má dráhu pojezdu, která je menší než dráha nastavená v MP 7050 00120 (2 až 90 μm, standardní hodnota: 10 μm), nebo
 - přechod vět podle parametrů stroje je považován za spojitý, tzn. úhel mezi oběma větami je menší než maximální úhel zadaný v MP 7050 00130 (standardní hodnota = 1 stupeň).

Rounding, RND

4.67 Zaoblení rohů se zadáním poloměru

Rounding, RND

Následek

Tato funkce vkládá mezi 2 lineární, kruhové, resp. spirálové větvy v hlavní rovině tečné přechodové oblouky.

Tím se sice poněkud změní naprogramovaná kontura na těchto rozích, ale pro interpolaci jsou získány spojitě průběhy rychlosti a zrychlení.

Programování

Syntaxe:

Rounding(<Polom[r >)

Zapnout zaoblení rohů mezi 2 lineárními/kruhovými/spirálovými větami.

Rounding(0) nebo
Rounding()

Vypnout zaoblení rohů mezi 2 lineárními/kruhovými/spirálovými větami.

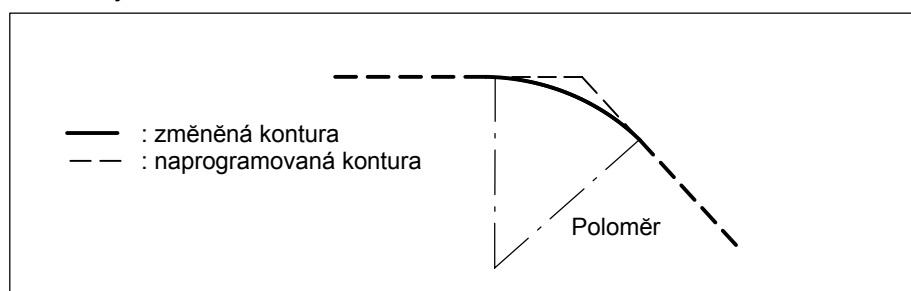
Zkratka: RND(..)

kde

<Poloměr> Požadovaný poloměr přechodového oblouku, desetinná místa jsou dovolena.

Zvláštnosti a omezení:

- Funkce "ChLength", "ChSection", "RoundEps" a "Rounding" působí modálně a vzájemně se ruší.
- Řízení **neprovede** funkci "zaoblení rohů", když
 - alespoň jedna z obou sousedních vět má dráhu pojezdu, která je menší než dráha nastavená v MP 7050 00120 (2 až 90 μm , standardní hodnota: 10 μm), nebo
 - přechod vět podle parametrů stroje je považován za spojitý, tzn. úhel mezi oběma větami je menší než maximální úhel zadáný v MP 7050 00130 (standardní hodnota = 1 stupeň).
- Pro zaoblení jsou brány v úvahu pouze složky aktivní pracovní roviny. To vede u prostorových přímek k tomu, že se mění jejich směr v prostoru. Něco podobného platí analogicky pro segmenty spirálové dráhy.



Scale(...), SCL(...)

4.68 Pomůcka: Nastavení měřítka

Scale(...), SCL(...)

Následek

Funkce nastavení měřítka patří k pomůckám.

Řízení zvětšuje nebo zmenšuje naprogramovanou konturu vzhledem k programovému nulovému bodu.

Nastavení měřítka se vždy vztahuje k aktuálnímu bodu zrcadlového převrácení (viz funkce PoleSet na straně 4-97). Pokud nebyl bod zrcadlového převrácení explicitně naprogramován, je jím aktuální programový nulový bod.

S použitím funkce nastavení měřítka je možné programovat kontury v dílčích programech vždy se stanovenou (normovanou) velikostí. Před vyvoláním takového "normovaného" dílčího programu (např. jako podprogramu) nastavte pomocí faktorů měřítka pro každou osu měřítka naprogramované kontury.

Při vytváření forem odlitků a výkovek tak mohou být např. snadno kompenzovány součinitele smrštění obrobků.

 **Nastavení měřítka je pomůcka, a proto nemění aktuální programovou soustavu souřadnic. Pomůcka pouze představuje jinou možnost zadávání programových souřadnic.**

Nastavení měřítka se může používat také společně se zrcadlovým převrácením a otáčením.

Platí:

- Funkce působí modálně. Zůstává účinná tak dlouho, dokud není vypnuta.
- Smí být naprogramována s jinými podmínkami a pomocnými funkcemi ve stejné větě.

Programování

Syntaxe:

Scale(<Osa1><Faktor>{,...})

Zapnout nastavení měřítka pro zadané osy s naprogramovaným faktorem.

Scale(0) nebo
Scale()

Vypnutí nastavení měřítka pro všechny osy v kanálu. Všechny měřítkové faktory jsou nastaveny na hodnotu "1".

Najeté osové polohy zůstávají zachovány, dokud nejsou nově naprogramovány.

Zkratka: SCL(..)

kde

<Osa1> Osové adresy (např. X), jejichž měřítka má být nastaveno.

Scale(...), SCL(...)

- <Faktor> Naprogramování osové adresy s kladným faktorem aktivuje funkci.
- Následkem toho jsou všechny další naprogramované dráhové příkazy příslušné osy (např. X10) interně vynásobeny tímto faktorem:
- Faktor > 1 : kontura se zvětšuje
 - Faktor < 1 : kontura se zmenšuje
 - Faktor = 1 : kontura zůstává beze změny.
- Nastavení měřítka samo o sobě nespouští žádný pojezdový pohyb a je účinné teprve s další pojezdovou informací.

Zvláštnosti a omezení:

- Záporné faktory měřítka nejsou dovoleny.
- Při kruhové/spirálové/spirálové-N interpolaci musí být měřítkové faktory všech zúčastněných souřadnic roviny kruhu stejné! Jinak je generováno chybové hlášení.
- Funkce působí také na parametry interpolace I, J, K a na absolutní hodnotu adresy R (při programování poloměru).
- Má vliv na programovatelné posunutí kontur (Shift). Viz strana 4-120.
- **Nemá** vliv na:
 - Programování posuvu nebo aktivní posuv,
 - Posunutí nulových bodů (G54-G59.5; viz strana 3-46),
 - Posunutí programových souřadnic (Trans nebo ATrans; viz strana 4-148),
 - Nastavení programových poloh ("SetPos"; viz strana 4-119),
 - Najetí na souřadnice referenčního bodu (G74 ; viz strana 3-55),
 - Najetí do pevné osové polohy stroje (G76 ; viz strana 3-58),
 - Hodnoty oprav poloměru frézy a délky nástroje.

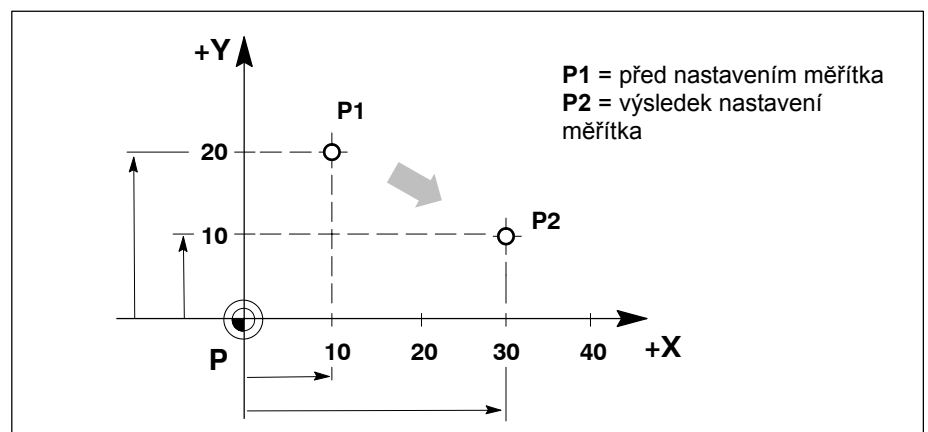
Příklad:

```

:
N30 SCL(X3, Y0.5)
:
:

```

Zapnout nastavení měřítka. Všechny další naprogramované souřadnice X jsou vynásobeny hodnotou "3" a souřadnice Y hodnotou "0.5".



SelCrdCouple, SCC

4.69 Selektivní aditivní spojení souřadnic

SelCrdCouple, SCC

Následek

Připojuje souřadnici obrobku aktuálního kanálu (cíl) k souřadnici obrobku jiného kanálu (zdroj). Hodnota souřadnice zdroje má tedy aditivní účinek na hodnotu souřadnice cíle.

Tímto způsobem je možné skládat pohyby v aktuálním kanálu s pohyby, které jsou naprogramovány v jiném kanálu.

Programování

Syntaxe:

SelCrdCouple(SC<Kanál>,
CL(<Q1>,<Z1>{{,<Qn>,<Zn>}...}))

Zapnutí selektivního
aditivního spojení souřadnic.

SelCrdCouple(0) nebo
SelCrdCouple()

Vypnutí všech aktivních
spojení souřadnic.

Zkratka: SCC(..)

kde

<Kanál> Číslo kanálu, ve kterém se nachází <Qn>.

Zadaná hodnota: Integer.

<Kanál> musí být menší než číslo aktuálního kanálu.

<Qn> Zdroj. Logický název nebo logické číslo souřadnice v
<kanálu>.

<Zn> Cíl. Logický název nebo logické číslo souřadnice v
aktuálním kanálu.

Příklady:

SCC(SC1,CL(YA,YB))

Připojuje souřadnici YB v aktuálním kanálu k
souřadnici YA z kanálu 1.

SCC(SC1,CL(1,1,2,2))

Připojuje souřadnice s logickými čísly 1 a 2 v
aktuálním kanálu ke všem souřadnicím s
logickými čísly 1, resp. 2 z kanálu 1.

SetPos, SPS

4.70 Nastavení programové polohy

SetPos, SPS

Následek

Nastavuje aktuální programový nulový bod (vztažený k aktuální programové soustavě souřadnic a aktivnímu nulovému bodu) na příslušnou naprogramovanou hodnotu, aniž by se spustily osové pohyby. Následně se automaticky zobrazí nové polohové hodnoty.

Programování

Syntaxe:

SetPos(<Souřadnice>)

Nastavit programový nulový bod pro osy naprogramované v <souřadnicích>.

SetPos

Zrušit všechna posunutí spuštěná funkcí SetPos.

Zkratka: SPS(..)

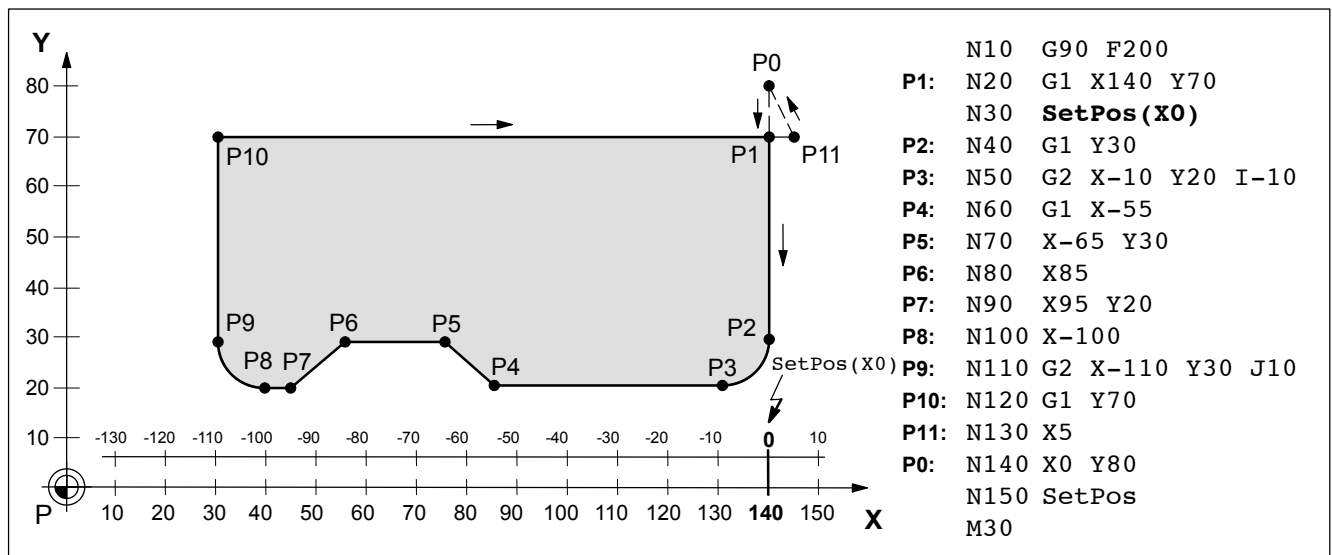
kde

<Souřadnice> Nastavení osových hodnot na uvedené souřadnice. Zadané souřadnice musí být odděleny čárkou (např.: (X0, Y0)).

Zvláštnosti a omezení:

- Jestli budou posunutí SetPos po základním nastavení vymazána nebo zůstanou zachována, se dá nastavit pro jednotlivé kanály pomocí parametrů stroje.

Příklad:



- ☞ Alternativně je možné naprogramovat také příkaz DIN "G92". Příkaz G92 však nesmí být naprogramován společně s pojzdovými informacemi v jedné větě.

Shift, SHT

4.71 Naprogramované posunutí kontur

Shift, SHT

Následek

Funkce Shift patří k pomůckám.

Řízení posunuje naprogramovanou konturu paralelně s osami programové soustavy souřadnic.

 **Shift je pomůcka, a proto nemění aktuální programovou soustavu souřadnic. Pomůcka pouze představuje jinou možnost zadávání programových souřadnic.**

Platí:

- Naprogramované posunutí souřadnice zůstávají účinné tak dlouho, dokud nejsou přepsány novou větou Shift nebo vypnuty.
- Naprogramovaná funkce Shift sama o sobě nevede k žádnému pojezdovému pohybu, ale ve stejné větě mohou být naprogramovány také pojezdové informace.



POZOR

Možnost poškození obrobku a stroje při chybném naprogramování!

Funkce Shift je ovlivňována zrcadlovým převrácením, nastavením měřítka a otáčením, tzn. souřadnice nového konturového nulového bodu, naprogramované ve větě Shift, se rovněž zrcadlově převrací, otáčejí, resp. je nastavováno jejich měřítko!

Programování

Syntaxe:

Shift(<Souřadnice>)

Zapnout posunutí kontur.

Shift(0) nebo
Shift()

Vypnout posunutí kontur.

Zkratka: SHT(..)

kde

<Souřadnice> Programové souřadnice posunutého konturového nulového bodu. Několik souřadnicových údajů se odděluje čárkami (např.: SHT (X5 , Y2)).

Příklad:

N10 SHT(X10, Y10, Z50)

Nový konturový nulový bod v X10, Y10, Z50. Žádný pojezd os.

:

N100 G1 X...Y...Z...

Pojezd os s ohledem na posunutí.

:

N110 SHT(X20, Y20)

Nový konturový nulový bod v X20, Y20, Z50 (posunutí Z zůstává!). Žádný pojezd os.

:

:

N210 SHT() X...Y...Z...

Vypnout posunutí, najet s osami do naprogramované polohy.

:

SMin, SMN SMax, SMX

4.72 Omezení počtu otáček

**SMin, SMN,
SMax, SMX**

Následek

Definuje rozsah otáček, ve kterém se smí pohybovat počet otáček vřetena při přímém programování počtu otáček G97 a konstantní řezné rychlosti G96 (viz strana 3-75) během úseku obrábění. Definovaný rozsah otáček platí pro všechny převodové stupně. Při aktivním omezení počtu otáček jsou všechny zadávané počty otáček omezeny na naprogramované mezní hodnoty.

Změny počtu otáček (i takové, které jsou způsobeny potenciometrem vřetena) provádí systém jedině tehdy, když se pohybují uvnitř předepsaného rozsahu počtu otáček.

Programování

Syntaxe:

SMin(<Počet otáček1>

Aktivuje jako dolní mez povoleného rozsahu počtu otáček <počet otáček1>.

SMin(0)nebo
SMin()Vypíná dolní mez.
Aktuální počet otáček pak není zdola omezen.

Zkratka: SMN(..)

SMax(S<Počet otáček2>)

Aktivuje jako horní mez povoleného rozsahu počtu otáček <počet otáček2>.

SMax(0)nebo
SMax()Vypíná horní mez.
Aktuální počet otáček pak není shora omezen.

Zkratka: SMX(..)

kde

<Počet otáček1>: Nejmenší přípustný počet otáček. Rozsah hodnot: > 0.

Musí být menší než <počet otáček2>.

<Počet otáček2>: Největší přípustný počet otáček. Rozsah hodnot: > 0.

Musí být větší než <počet otáček1>.

Příklad:

N50 X.. Y.. SMN(S1500)

Počet otáček vřetena musí být v rozsahu 1500 až 2500 ot./min.

N60 X.. Y.. SMX(S2500)

:

N90 X.. Y.. SMN() SMX()

Vypnout dolní a horní mez (= vypnutí omezení počtu otáček).

Zvláštnosti a omezení:

- Omezení počtu otáček působí, pouze když naprogramované mezní hodnoty počtu otáček leží v mezích převodových stupňů.

SpAdmin, SPA

4.73 Uvolnění/převzetí rezervovaného vřetena

SpAdmin, SPA

Následek

Umožňuje

- uvolnit vřeteno rezervované právě v aktuálním kanálu, aniž by se rotující vřeteno zastavilo
- následně ho převzít libovolným kanálem.

 **Vřeteno se rovněž uvolňuje pomocí funkce "Zastavení vřetena" (viz strana 3-90).**

Programování

Syntaxe:

SpAdmin (s<Num>=<Režim>{, s<Num>=<Režim>}...)Zkratka: **SPA (...)**

kde

<Num> Číslo (index) vřetena.
Rozsah zadání: 1...8. Integer.

<Režim> 0: Uvolnit vřeteno
 1: Rezervovat a převzít uvolněné vřeteno

Zvláštnosti a omezení:

- Základní nastavení nebo funkce M30 uvolňuje vřetena rezervovaná aktuálním kanálem pouze tehdy, když je na příslušném místě v MP7060 00020 zadán příkaz pro "zastavení vřetena".

Příklad:

N60 SPA(S1=0, S2=0) 1. a 2. vřeteno uvolnit.
:

SpCoupleConfig, SPCC

4.74 Definice (aktivace) svazku spojení Rozpojení (deaktivace) svazku spojení SpCoupleConfig, SPCC

Následek

- Definuje svazek spojení a aktivuje jeho spojení vřetena. Přitom přepíná řízení automaticky zúčastněné pohony vřeten na rozhraní polohy.
- Přidává následná vřetena k existujícímu svazku spojení nebo je odebírá z existujícího svazku spojení. Přidané pohony vřeten se automaticky přepnou na rozhraní polohy, vyjmuté pohony vřeten na rozhraní počtu otáček (pokud bylo na příslušných vřetenech před spojením aktivováno rozhraní počtu otáček).
- Deaktivuje spojení vřetena svazku spojení a rozpojuje celý svazek spojení. Všechny zúčastněné pohony vřeten se přepnou na rozhraní počtu otáček, pokud bylo na příslušných vřetenech před spojením aktivováno rozhraní počtu otáček.

Programování

Syntaxe:

SpCoupleConfig (CP=<Svazek>,MA=<Master>, S<Pořadí>=1{{,S<Pořadí>=1}...})	Definovat svazek spojení.
SpCoupleConfig (CP=<Svazek>,MA=0)	Rozpojit svazek spojení.
SpCoupleConfig (CP=<Svazek>, S<Pořadí>=<Režim> {{,S<Pořadí>=<Režim>}...})	Přidat/odebrat následná vřetena ke/ze svazku spojení.

Zkratka: **SPCC (...)**

kde

<Svazek>	Číslo svazku spojení. Rozsah zadání: 1...4. Integer.
<Master>	Číslo řídicího vřetena (index vřetena). Rozsah zadání: 1...8. Integer.
<Pořadí>	Číslo následného vřetena (index vřetena). Rozsah zadání: 1...8. Integer.
<Režim>	0: Odstranit následné vřeteno ze <svazku>. 1: Přidat následné vřeteno ke <svazku>.


SpCoupleDist, SPCD

4.75 Vzdálenost spojení následného vřetena SpCoupleDist, SPCD

Následek

Konfiguruje pro funkci "spojení vřeten" požadovaný rozdíl poloh mezi řídicím a následným vřetenem v okamžiku spojení (při vytvoření spojení).

Není-li funkce pro požadované následné vřeteno použita, činí jeho polohový rozdíl vůči řídicímu vřetenu 0 stupňů (v okamžiku spojení).

 **Informace o najetí na aditivní úhlové posunutí mezi řídicím a následným vřetenem během aktivního spojení vřeten viz funkce "SpCouplePosOffs" na straně 4-126.**

Programování

Syntaxe:

```
SpCoupleDist (S<Num>=<Vzdálenost>{, S<Num>=  
<Vzdálenost>}...)
```

Zkratka: **SPCD** (...)

kde

<Num> Číslo následného vřetena (index vřetena).
Rozsah zadání: 1...8. Integer.

<Vzdálenost> Rozdíl poloh mezi řídicím a následným vřetenem ve stupních.

Rozsah hodnot: $0^\circ \leq \text{rozdíl poloh} < 360^\circ$.

Pokud je naprogramována jiná hodnota, je automaticky přepočítána do zadaného intervalu.

Zvláštnosti a omezení:

- Funkce smí být naprogramována pouze s vřeteny, která momentálně nepatří k žádnému svazku spojení.

SpCoupleErrWin, SPCE

4.76 Okno chyby synchronního chodu

SpCoupleErrWin, SPCE

Následek

Konfiguruje pro funkci "spojení vřeten" maximální povolenou odchylku polohy mezi požadovanou a skutečnou hodnotou následného vřeten. Pokud odchylka polohy leží během aktivního spojení v definovaném intervalu, je vydán výstupní signál spojený s vřetenem "**Synchronní chod 2**".

Jestliže není použita tato funkce pro požadované následné vřeten, činí jeho okno chyby synchronního chodu +/-10 stupňů.

 **Viz též funkce "SpCoupleSyncWin" na straně 4-128.**

Programování

Syntaxe:

SpCoupleErrWin (S<Num>=<Okno>{, S<Num>=<Okno>}...)

Zkratka: **SPCE (...)**

kde

<Num> Číslo následného vřeten (index vřeten).
Rozsah zadání: 1...8. Integer.

<Okno> Maximální povolená odchylka polohy od požadované hodnoty ve stupních.
Rozsah zadání: 0 až 359.9999.

Zvláštnosti a omezení:

- Funkce smí být naprogramována pouze s vřeteny, která momentálně nepatří k žádnému svazku spojení.

SpCouplePosOffs, SPCP

4.77 Úhlové posunutí při aktivním spojení **SpCouplePosOffs, SPCP**

Následek

Najíždí během aktivního spojení vřeten do úhlového posunutí mezi řídicím a následným vřetenem. Relativní počet otáček mezi řídicím a následným vřetenem, s nímž má proběhnout najetí do úhlového posunutí, se dá podle potřeby naprogramovat.

- ☞ **Po dobu trvání otáčení resetuje řízení výstupní signál pro vřeteno "Synchronní chod 1".**

Programování

Syntaxe:

```
SpCouplePosOffs (S<Num>=<Posunutí>{{, S<Num>=<Posu-  
nutí>}...}  
{, POSVEL<Počet otáček>)
```

Zkratka: **SPCP** (...)

kde

- <Num> Číslo následného vřeten (index vřeten).
Rozsah zadání: 1...8. Integer.
- <Posunutí> Absolutní úhel otočení mezi řídicím a následným vřetenem ve stupních.
Rozsah hodnot: -3600° ... +3600°
- <Počet otáček> Relativní počet otáček mezi řídicím a následným vřetenem, s nímž má proběhnout najetí do úhlového posunutí.
Jednotka a standardní hodnota podle parametru pohonu S-0-0222.
Po zadání zůstává <počet otáček> interně uložený, dokud se nezmění opětovným vyvoláním funkce.

Zvláštnosti a omezení:

- Úhlové posunutí působí aditivně k případně zkonfigurované vzdálenosti spojení (viz strana 4-124).

SpCouplePosOffs_Wait, SPCP_WAIT

4.78 Čekání na úhlové posunutí SpCouplePosOffs_Wait, SPCP_WAIT

Následek

Zastavuje dílčí program, dokud na stroji skutečně neproběhne najetí do naprogramovaného úhlového posunutí svazku spojení (SPCP; viz strana 4-126).

Programování

Syntaxe:

SpCouplePosOffs_Wait (CP=<Svazek>)

Zkratka: **SPCP_WAIT**(...)

kde

<Svazek> Číslo svazku spojení.
Rozsah zadání: 1...4. Integer.

SpCoupleSyncWin, SPCS

4.79 Okno synchronního chodu

SpCoupleSyncWin, SPCS

Následek

Konfiguruje pro funkci "spojení vřeten" maximální povolenou odchylku polohy mezi požadovanou a skutečnou hodnotou následného vřetena.

- Působí při vytvoření či změně spojení (na začátku synchronizační fáze).
Dílčí program čeká tak dlouho, dokud odchylka polohy neleží uvnitř definovaného intervalu.
- Působení během spojení:
Pokud odchylka polohy leží v definovaném intervalu, je vydán výstupní signál spojený s vřetenem "**Synchronní chod 1**".

Jestliže není použita tato funkce pro požadované následné vřeteno, činí jeho okno synchronního chodu +/-1 stupeň.

 **Informace o přídavném monitorování během aktivního spojení viz funkce "SpCoupleErrWin" na straně 4-125.**

Programování

Syntaxe:

SpCoupleSyncWin (S<Num>=<Okno>{, S<Num>=<Okno>}...)

Zkratka: **SPCS (...)**

kde

<Num> Číslo následného vřetena (index vřetena).
Rozsah zadání: 1...8. Integer.

<Okno> Maximální povolená odchylka polohy od požadované hodnoty ve stupních.
Rozsah zadání: 0 až 20.

Zvláštnosti a omezení:

- Funkce smí být naprogramována pouze s vřeteny, která momentálně nepatří k žádnému svazku spojení.

SpCouplePosOffs_Wait, SPCP_WAIT

4.80 Čekání na synchronní provozSpCouplePosOffs_Wait, SPCP_WAIT

Následek

Zastavuje dílčí program, dokud není úspěšně vytvořen, nově zkonfigurován nebo rozpojen svazek spojení (SPCC; viz strana 4-123).

Programování

Syntaxe:

SpCouple_wait(CP=<Svazek>)

Zkratka: **SPC_WAIT**(...)

kde

<Svazek> Číslo svazku spojení.
Rozsah zadání: 1...4. Integer.

SPG..., SPGALL

4.81 Definice/rozpojení skupin vřeten

SPG..., SPGALL

Následek

Skupiny vřeten (nazývané též paralelní vřetena) umožňují zjednodušené programování několika vřeten v souvislosti s funkcemi pro převodové stupně, pravotočivý/levotočivý chod, zastavení vřetena a seřízení vřetena. Tím se ušetří oddělené programování jednotlivých vřeten.

☞ **Statické přiřazení všech vřeten k jednotlivým skupinám vřeten je možné pomocí MP 1040 00002.**

☞ **Funkce "Skupina vřeten" se nesmí zaměňovat se "svazkem spojení" dvou nebo více polohově regulovaných vřeten. Zatímco vřetena skupiny vřeten mohou běžet s různými otáčkami, všechna vřetena svazku spojení běží v principu synchronně.**

Programování

Syntaxe:

SPG<Skupina>(<Číslo>)	Určuje, která vřetena v aktuálním kanálu mají být sdružena jako skupina vřeten.
SPG<Skupina>(0)	Obnovuje v aktuálním kanálu pro odpovídající skupinu vřeten nastavení podle MP 1040 00002.
SPG<Skupina>(-1)	Rozpojuje v aktuálním kanálu příslušnou skupinu vřeten.
SPGALL(0)	Obnovuje v aktuálním kanálu pro všechny skupiny vřeten nastavení podle MP 1040 00002.

kde

<Skupina>	Číslo skupiny vřeten. Rozsah zadávání: 1 až 4. Integer.
<Číslo>	Číslo všech vřeten (index vřeten) oddělená čárkou, která mají být přiřazena k příslušné skupině vřeten. Rozsah zadání: 1...8. Integer.

Zvláštnosti a omezení:

- Konkurojící si úlohy mezi jednotlivými vřeteny a skupinami vřeten, naprogramované ve stejné větě, vyvolávají běhovou chybu. (Příklad: N10 M3 M104).

SpindleToAxis, STA

4.82 Zapnutí provozu osy C pro vřetena

SpindleToAxis, STA

Následek

Zapíná vřeteno, které je zaznamenáno

- v MP 1001 00001 (typ funkce pohonu) jako vřeteno / osa C a
- v MP 1040 00001 (výběr typu vřetena) jako vřeteno SERCOS, do provozu osy C. Z vřetena se přitom z hlediska obráběcí techniky stane asynchronní osa.

Na displeji se objeví jako asynchronní osa, která je nejprve v libovolné poloze mezi 0 a 359.9999 stupni.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

SpindleToAxis (<PAN> | <PAI>{, <PAN> | <PAI>}...)

Zkratka: **STA**(..)

kde

<PAN>	Fyzický název osy. Určuje vřeteno, které má být přepnuto do provozu osy C.
<PAI>	Fyzický index osy. Účinky jako <PAN>.

Zvláštnosti a omezení:

- Zadané vřeteno musí být zastavené. Pokud tomu tak není, generuje řízení chybové hlášení a přeruší program.
- Osové polohy ve stejné větě musí být vždy naprogramovány za výrazem SpindleToAxis(...).

Příklad:

:	
N030 STA(CH)	Fyzická osa CH (tzn. vřeteno, které má v osovém provozu název CH) je přepnuta na asynchronní osu.
:	

SplineDef, SDF

4.83 Definice typu spline křivky

SplineDef, SDF

Následek

IndraMotion MTX podporuje následující typy spline křivek:

- **Spline typu 0:** Spline křivka s programováním koeficientů (koeficienty polynomů ze systému CAD/CAM)
- **Spline typu 1:** C¹-spojité kubické spline křivky s programováním opěrných bodů (tangenciální přechody v opěrných bodech)
- **Spline typu 2:** C²-spojité kubické spline křivky s programováním opěrných bodů (obloukové spojité přechody v opěrných bodech)
- **Spline typu 3:** B-spline křivky s programováním kontrolních bodů (křivkový průběh v blízkosti opěrných bodů).

Požadovaný typ spline křivek se vybírá a inicializuje pomocí funkce "SplineDef" (SDF). Pak je možné aktivovat spline programování s použitím funkce G6.

Programování

Syntaxe:

SplineDef(<Id>,{<Členy>})

Zkratka: SDF(..)

kde

<Id>:

Inicializace typu spline křivky.

maximálně 4-místné celé číslo pro variantu spline křivek:

Typ spline křivky	Místo			
	1000.	100.	10.	1.
Typ spline křivky 0...3		Parametrizace 0: Žádná 1: Ekvidistantní 2: Chordální 3: Centripetální	Výpočet tečen 0: Žádný 1: Bessel 2: Akima 3: Sehnen	Stupeň spline křivky 1...5
Typ spline křivky 0	0	0	0	1...5
Typ spline křivky 1	1	1...3	0...3	1...5
Typ spline křivky 2	2	1...3	0...3	1...5
Typ spline křivky 3	3	1...3	0	1...5

SplineDef, SDF

<Členy>:

Ne pro spline křivky typu 0!

Seznam s názvy souřadnic/os, které se mají zúčastnit spline pohybu.

Ve <členech> nenaprogramované souřadnice/osy se pohybují lineárně.

Orientační souřadnice se dají naprogramovat také jako spline funkce. Přitom se ve <členech> zadává "O" pro orientaci nebo polární souřadnice "phi" a "theta".

Příklady:

N10 SplineDef(5)	Inicializovat spline funkci typu 0 se stupněm spline křivek 5.
:	
N20 SplineDef(2203,X,Y,Z)	Inicializovat spline funkci typu 2 se stupněm spline křivek 3 a chordální parametrizací. Zúčastněny jsou souřadnice X, Y a Z.
:	
N30 SplineDef(3103,x,y,z,O)	Inicializovat spline funkci typu 3 se stupněm spline křivek 3 a ekvidistantní parametrizací. Zúčastněny jsou souřadnice x, y, z a orientační souřadnice.
:	

Split, SPT

4.84 Programovatelné rozdělení dráhy

Split, SPT

Následek

Rozděluje naprogramované pojezdové věty na několik dílčích drah, pokud překročí určitou délku.

Programování

Syntaxe:

Split ({<Režim>}{, <Dílčí délka>})

Zkratka: **SPT** (...)

kde

<Režim> 0: Funkce nepracuje.
1: Funkce pracuje modálně.
2: Funkce působí pouze v naprogramované větě.
Není naprogramováno: Působení jako při 0.

<Dílčí délka> U lineárních vět: délka dílčí dráhy.
U kruhových vět: délka oblouku.
Programovací jednotka jako pro souřadnice os.
Pokud není <dílčí délka> naprogramována, platí:
Účinná <dílčí délka> odpovídá přibližně maximální dráze, která může být uražena ve 2 po sobě následujících interpolačních taktech v závislosti na aktuálním posuvu.

Zvláštnosti a omezení:

- Naprogramované rozdělení dráhy dělí pojezdovou větu pouze tehdy, když je její délka větší než <dílčí délka>.
- Jestliže <dílčí délka> není celým dělitelem pojezdové dráhy, je také zbývající část považována za samostatnou dílčí délku.
- Pokud programovatelné rozdělení dráhy působí po větách (<Režim>=2), musí být ve stejné větě naprogramován pojezdový pohyb.

Příklad:

:	
N20 Split(2,10) X100	Zapnout rozdělení dráhy účinné po větách pro N20. Pojezdový pohyb v N20 se rozdělí na dílčí dráhy po 10 mm.
:	
:	
N120 Split(1)	Zapnout modálně účinné rozdělení dráhy od N120. Pojezdové pohyby od N120 se rozdělí na dílčí dráhy, které mohou být pojížděny cca ve 2 interpolačních taktech.
:	
:	
N220 Split(1,3.0)	Zapnout modálně účinné rozdělení dráhy od N220. Pojezdové pohyby od N220 se rozdělí na dílčí dráhy po 3 mm.
:	
:	
N320 Split()	Vypnutí programovatelného rozdělení dráhy.
:	

SpMode, SPM

4.85 Vřeteno: Přepnutí rozhraní polohy / počtu otáček SpMode, SPM**Následek**

Přepíná pohon vřetena mezi otáčkovým a polohovým provozem.

Programování

Syntaxe:

SpMode (S<Num>=<Režim>{, S<Num>=<Režim>}...)Zkratka: **SPM (...)**

kde

<Num> Číslo (index) vřetena.
Rozsah zadání: 1...8. Integer.**<Režim>** 0: Otáčkový provoz
1: Polohový provoz**Zvláštnosti a omezení:**

- Seřízení vřetena je přerušeno
- otáčející se vřeteno se při přepnutí na polohový provoz krátce zastaví.

Příklad:N60 SPM(S1=1, S2=1)
:

1. a 2. vřeteno přepnout na polohový provoz.

StatToolOri, STO

4.86 Parametrizace statické orientace nástroje

StatToolOri, STO

Následek

Pomocí funkce G47(...) nebo G78 se přiřazují opravy délky nástroje L_1 , L_2 a L_3 souřadnicovým směřům L_1 , L_2 a L_3 . Tyto souřadnicové směry tvoří soustavu souřadnic a vektor opravy délky nástroje L má v této soustavě souřadnic složky L_1 , L_2 a L_3 .

Příkazem StatToolOri se L otáčí v této soustavě souřadnic o Eulerovy úhly ϕ , θ a ψ .

Při této orientaci nástroje se vektor L otáčí následujícím způsobem:

- nejprve o úhel ϕ kolem souřadnicového směru L_3 ,
- pak o úhel θ kolem **otočeného souřadnicového směru** L_2 ,
- a nakonec o úhel ψ (ψ) kolem **dvakrát otočeného souřadnicového směru** L_3 .

Programování

Syntaxe:

StatToolOri ({<phi>{=}< φ >} {<theta>{=}< ϑ >} {<psi>{=}< ψ >})

Zkratka: STO(..)

kde

<phi>

Při aktivní transformaci os:

Název úhlu, který je zadán v MP 7080 00010[4].

Standardní: **phi**.

Při neaktivní transformaci os: **PHI**

<theta>

Při aktivní transformaci os:

Název úhlu, který je zadán v MP 7080 00010[5].

Standardní: **theta**.

Při neaktivní transformaci os: **THETA**

<psi>

Při aktivní transformaci os:

Název úhlu, který je zadán v MP 7080 00010[6].

Standardní: **psi**.

Při neaktivní transformaci os: **PSI**

< φ >, < ϑ >, < ψ >

Absolutní Eulerovy úhly ve stupních.

Programování AC/IC je dovoleno.

Rozsah hodnot:

$0^\circ \leq \varphi < 360^\circ$; $0^\circ \leq \vartheta \leq 180^\circ$; $0^\circ \leq \psi \leq 360^\circ$.

Hodnoty mimo rozsah se automaticky přepočítávají do příslušného intervalu.

Zvláštnosti a omezení:

- Statická orientace nástroje se parametrizuje pomocí funkce STO a aktivuje pomocí funkce G47 (viz strana 3-44).
- Pokud je G47 při provádění funkce STO již aktivní, je změna orientace nástroje okamžitě účinná.
- Jestliže nebyly všem 3 opravám délky nástroje L_1 , L_2 a L_3 přiřazeny funkcí G47(...) nebo G78 identifikátory souřadnic, bere řízení při orientaci nástroje v úvahu pouze přiřazené souřadnice a generuje varování.

TangTool, TTL

4.87 Tangenciální vedení nástroje

TangTool, TTL

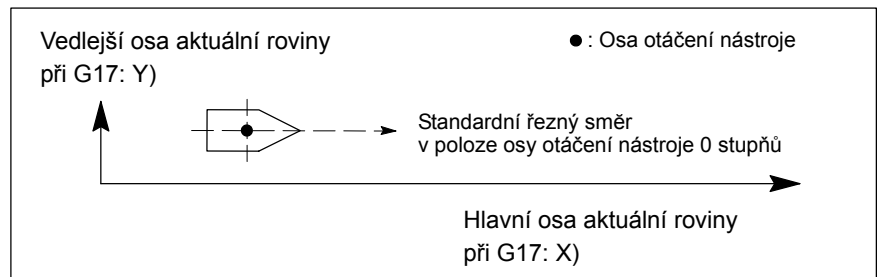
Následek

Vede kruhovou nebo nekonečnou osu pod nastavitelným úhlem podél naprogramované dráhy v aktivní rovině. Tímto způsobem se může nástroj během záběru kdykoli pohybovat pod požadovaným úhlem nastavení k naprogramované dráze.

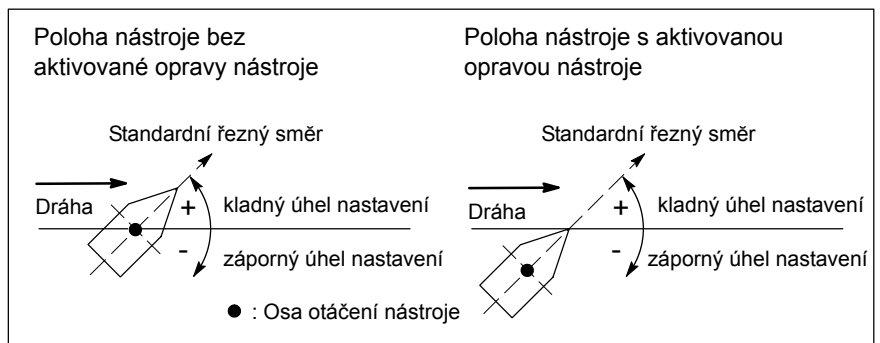
Aby se nástrojová osa otočila na příslušný požadovaný úhel nastavení, bere funkce u nástrojů s více břity v úvahu případně existující symetrii nástroje. Kromě toho může řízení při ohybu kontury na přechodu vět automaticky vložit další větu, která otočí nástrojovou osu o požadovaný úhel.

Platí

- V poloze osy otáčení nástroje 0 stupňů leží standardní řezný směr nástroje paralelně s kladným směrem pojezdu hlavní osy aktuální roviny.



- Úhel nastavení udává rozdíl úhlů mezi dráhou a standardním řezným směrem nástroje. Je 0 stupňů, když standardní řezný směr nástroje prochází tečně k naprogramované dráze.



- "TangTool" nevede při zapnutí k žádnému pojezdovému pohybu. Naprogramovaná nástrojová osa se nastavuje teprve s dalším pojezdovým pohybem. Přitom se v závislosti na parametrizovaném úhlu ve vložené větě buď
 - nejprve prochází vložená věta pro otočení nástroje, nebo
 - přeskakuje na začátek věty na polohu nastavení.

TangTool, TTL

- Výpočet pro požadované otáčení nástrojové osy při daném úhlu nastavení během kruhové dráhy se provádí v taktu interpolátoru. Osa nástroje se proto při takových úsecích kontur otáčí v taktu interpolátoru o aktuálně vypočítaný úhel nastavení dál.

Programování

Syntaxe:

```
TangTool({TAX{=}<Osa>
{,SYM{=}<s>},{,ANG{=}<a>
{,IA{=}<zsw>},{,PLC{=}<p>}})
```

Zapnutí tangenciálního vedení nástroje.

```
TangTool() nebo
TangTool(0)
```

Vypnutí tangenciálního vedení nástroje.

Zkratka: TTL(..)

kde

<Osa>: Označení osy, na kterou má působit tangenciální vedení nástroje.

Je dovolen logický či fyzický název osy nebo logické číslo osy.

Pokud není naprogramováno, platí MP 7050 00210.

<s>: Symetrie nástroje (zpravidla počet břitů).

Zadaná hodnota: Integer, různá od 0.

Nástroj se symetrií <s> dosahuje při otočení o 360 stupňů / <s> technologicky ekvivalentní polohy.

Příklady: Obdélníkový nástroj: <s> = 2, čtvercový nástroj: <s> = 4.

1: Nesymetrický nástroj nebo existuje pouze jeden břit.

>0: Symetrický nástroj, který má několik břitů se stejnými vzájemnými rozestupy.

Při ohybu kontury se nástroj otáčí jen do té míry, aby byl další břit v úhlu nastavení ke kontuře.

<0: Při změně směru (ohyb 180 stupňů) neprobíhá nezávisle na úhlu nastavení žádné otočení nástroje.

Jinak jako při ">0".

Pokud není naprogramováno, platí MP 7050 00220.

<a>: Úhel nastavení. Rozsah hodnot: -180 stupňů až +180 stupňů.

Udává rozdíl úhlů mezi dráhou a standardním řezným směrem nástroje.

Pokud není naprogramováno, platí MP 7050 00250. Zde se určuje, jestli se aktuální úhel osy otáčení nástroje použije jako úhel nastavení nebo má být použita hodnota z MP 7050 002540.

<zsw>: Úhel ve vložené větě. Rozsah hodnot: 0 až 180 stupňů.

Určuje, od jakého úhlu ohybu mezi dvěma větami se má vkládat další věta pro otočení nástrojové osy.

Při úhlech ohybu menších než <zsw> se pro otočení nevkládá žádná zvláštní věta. Nástroj přeskočí na začátku další věty do nové polohy.

Pokud není naprogramováno, platí MP 7050 00230.

TangTool, TTL

- <p>: Zapnout nebo vypnout komunikaci NC-SPS během provádění vložené věty.
- 0: Vypnout komunikaci NC-SPS.
NC provádí vloženou větu bez podmínek.
- 1: Zapnout komunikaci NC-SPS.
Provádění otáčecí věty je řízeno komunikací mezi NC a SPS.
- Pokud není naprogramováno, platí MP 7050 00260.

Příklady:

TTL	Pro osu zadanou v
:	MP 7050 00210 se aktivuje tan-
:	genciální vedení nástroje s od-
:	povídajícími hodnotami
:	parametrů stroje.
TTL (TAX=C , SYM1 , ANG90 , IA20 , PLC0)	Programování s logickým
:	osovým názvem.
TTL (TAX3 , SYM1 , ANG90 , IA20 , PLC0)	Programování s logickým
:	osovým číslem.
TTL (TAX [NAME\$] , SYM1 , ANG90 , IA20)	Programování s proměnnou
:	CPL.

Zvláštnosti a omezení:


- "Tangenciální vedení nástroje" nesmí být aktivní zároveň s "tangenciální orientací nástroje" (viz strana 4-140).
- "Tangenciální vedení nástroje" nesmí být naprogramováno společně s pojezdovým pohybem.

TangToolOri, TTO

4.88 Tangenciální orientace nástroje

TangToolOri, TTO

Následek

 **Působí pouze v souvislosti s funkcemi "Děrování" (viz strana 4-106) a "Vysekávání" (viz strana 4-74).**

Při zapnutém děrování/vysekávání zajišuje, aby byl děrovací/vysekávací nástroj při každém zdvihu nastaven v definovaném úhlu k naprogramované dráze.

Při zdvihu na začátku věty se přitom - je-li to nutné - automaticky vkládá věta pro otočení nástroje.

Optimální směr otáčení určuje řízení automaticky (nejkratší dráha).

Osa otáčení je pevně dána parametrem MP 7050 00210.

Programování

Syntaxe:

TangToolOri({SYM<s>},{ANG<a>}) nebo Zapnutí tangenciální orientace nástroje
TangToolOri(1)

TangToolOri

TangToolOri() nebo Vypnutí tangenciální orientace nástroje
TangToolOri(0)

Zkratka: TTO...

kde

<s>: Symetrie nástroje (zpravidla počet břitů).
Zadaná hodnota: Integer, větší než 0.
Nástroj se symetrií <s> dosahuje při otočení o 360 stupňů / <s> technologicky ekvivalentní polohy.
Příklady: Obdélníkový nástroj: <s> = 2, čtvercový nástroj: <s> = 4.
1: Nesymetrický nástroj nebo existuje pouze jeden břit.
>1: Symetrický nástroj, který má několik břitů se stejnými vzájemnými rozestupy.
SYM není naprogramováno: Fungování jako SYM1.

<a>: Úhel nastavení k naprogramované dráze.
Rozsah hodnot: -180 stupňů až +180 stupňů.
0: Nástrojová osa je nastavena v úhlu 0 stupňů k naprogramované dráze.
ANG není naprogramováno: Fungování jako ANG0.

Zvláštnosti a omezení:

- Funkce nesmí být naprogramována společně s funkcí "TangTool" (tangenciální vedení nástroje, TTL).
- Při aktivované funkci není možné "tangenciální vedení nástroje" (TTL, viz strana 4-137).

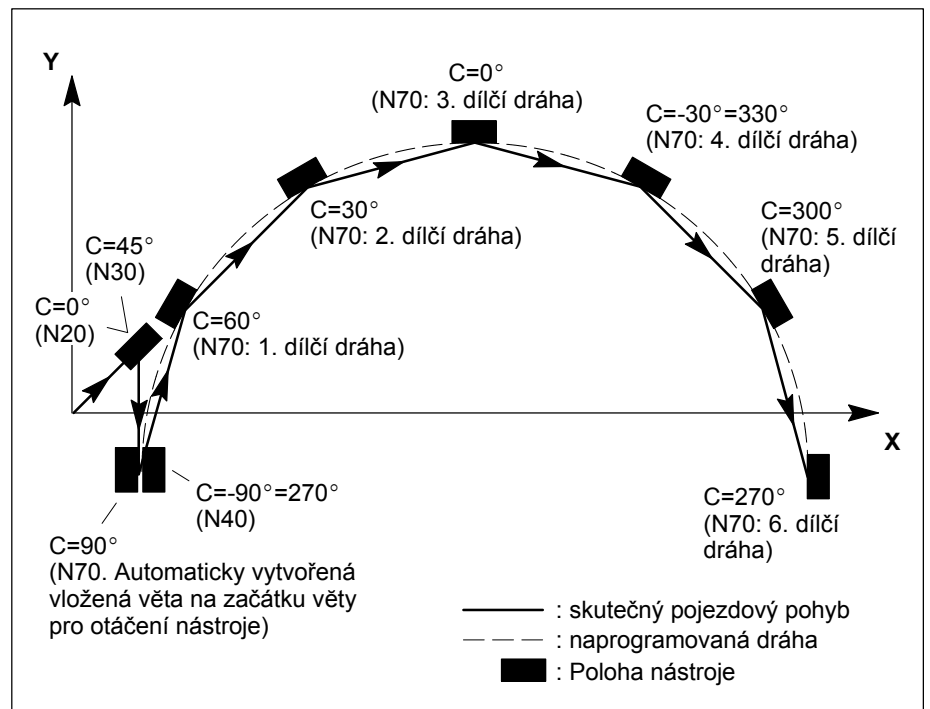
Příklad:

- G90 je aktivní (programování absolutních rozměrů)
- Aktivní rovina: X/Y
- Nástrojová osa: C (modulo 360)

TangToolOri, TTO

- Aktuální poloha: $X=0$, $Y=0$, $C=0$
- Děrování/vysekávání je vypnuté.

:		
N10	TangToolOri(1)	Zapnout tangenciální orientaci nástroje s působením jako SYN1 a ANG0.
N20	G1 G91 X10 Y10	Zatím žádná orientace osy C, protože je vypnuté děrování/vysekávání.
N30	X10 Y10 Punch(1)	Zapnout děrování. Osa C se otočí na 45° .
N40	Y-10	Osa C se otočí na -90° . Po modulo výpočtu: $C=270^\circ$.
N50	Punch(0)	Vypnout děrování.
N60	LEN=30 Nibble(1)	Zapnutí vysekávání.
N70	G2 X114.6 I57.3 J0	Půlkruh s délkou oblouku 180 mm. Věta je rozdělena na 6 dílčích vět s délkou oblouku po 30 mm.
N80	TTO()	Vypnout tangenciální orientaci nástroje.
:		



TappSp, TSP

4.89 Výběr vřetena pro řezání vnitřních závitů bez vyrovnávacího sklíčidla TappSp, TSP

Následek

- Určuje, ke kterým vřetenům se má vztahovat G63. Funkce "TappSp" působí tak dlouho, dokud ji znovu nenaprogramujete. Působí také po "základním nastavení"!
- Bez naprogramování funkce "TappSp" se G63 vztahuje vždy k 1. vřetenu.

Programování

Syntaxe:

TappSp(CAX< <i>i</i> >)	Aktivovat řezání vnitřních závitů pro jednotlivá vřetena pomocí čísel vřeten.
TappSp(GRP< <i>j</i> >)	Aktivovat řezání vnitřních závitů pro všechna vřetena ze skupiny vřeten.
TappSp(GRP< <i>j</i> >{,CAX< <i>i</i> >})	Aktivovat řezání vnitřních závitů pro všechna vřetena ze skupiny vřeten a další vřetena pomocí čísel vřeten.

Zkratka: TSP(..)

kde

< <i>i</i> >	Číslo vřetena , ke kterému se vztahuje G63. Rozsah hodnot: 1 ... n, přičemž n znamená nejvyšší číslo vřetena v systému (max. 8). Celočíselné. Více vřeten se programuje s použitím několika příkazů CAX< <i>i</i> >, oddělených čárkou.
< <i>j</i> >	Číslo skupiny vřeten , ke které se vztahuje G63. Rozsah hodnot: 1 ... n, přičemž n znamená nejvyšší číslo skupiny vřeten (max. 4). Celočíselné.

Zvláštnosti a omezení:

- Může být aktivována pouze jediná skupina vřeten pro G63.
- Čísla vřeten a případně navíc 1 skupina vřeten se mohou libovolně vzájemně kombinovat.

Příklad:

N20 TappSp(CAX2)	Vybrat pro G63 vřeteno s číslem vřetena 2.
:	
N120 TappSp(CAX2, CAX4, CAX7)	Vybrat pro G63 vřetena s čísly vřetena 2, 4 a 7.
:	
N220 TappSp(GRP2)	Vybrat pro G63 skupinu vřeten s číslem 2.
:	
N320 TappSp(GRP3, CAX4)	Vybrat pro G63 skupinu vřeten s číslem 3 a kromě toho ještě vřeteno s číslem vřetena 4.
:	

TcsDef, TCS

4.90 Definice TCS v programových souřadnicích

TcsDef, TCS

Následek

Vytváří při aktivní 6-osové transformaci os nástrojovou soustavu souřadnic TCS_p , která může být oproti poslední aktuální soustavě TCS posunutá, resp. otočená.

Hodnoty souřadnic zadané pro TCS_p se interně přepočítávají v NC a ukládají v paměti oprav nástroje.

Vytvořená soustava TCS_p se dá vypnutím funkce opět vymazat. Přitom se znovu aktivuje poslední aktuální TCS.

Programování

1. Pokud jste to ještě neudělali, aktivujte transformaci os typu **3333301** (viz funkce "Coord(...)", strana 4-30).
2. Použijte níže uvedenou syntaxi.

Syntaxe:

$TcsDef(\{<Poloha>\},\{<Orientace>\})$

Definovat a aktivovat polohu a orientaci soustavy TCS_p . Detailní syntaxe je uvedena níže.

$TcsDef()$ nebo
 $TcsDef(0)$

Vymazat TCS_p a znovu aktivovat poslední aktuální TCS.

Zkratka: **TCS (...)**

Detailní syntaxe:

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{<phi><\varphi>\},\{<theta><\vartheta>\},\{<psi><\psi>\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{O(\langle\varphi\rangle,\langle\vartheta\rangle,\langle\psi\rangle)\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{Ox(\langle\varphi_x\rangle,\langle\vartheta_x\rangle) Oy(\langle\varphi_y\rangle,\langle\vartheta_y\rangle)\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{Ox(\langle\varphi_x\rangle,\langle\vartheta_x\rangle) Oz(\langle\varphi_z\rangle,\langle\vartheta_z\rangle)\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{Oy(\langle\varphi_y\rangle,\langle\vartheta_y\rangle) Oz(\langle\varphi_z\rangle,\langle\vartheta_z\rangle)\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{Ox(\langle o11\rangle,\langle o21\rangle,\langle o31\rangle) Oy(\langle o12\rangle,\langle o22\rangle,\langle o32\rangle)\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{Ox(\langle o11\rangle,\langle o21\rangle,\langle o31\rangle) Oz(\langle o13\rangle,\langle o23\rangle,\langle o33\rangle)\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{Oy(\langle o12\rangle,\langle o22\rangle,\langle o32\rangle) Oz(\langle o13\rangle,\langle o23\rangle,\langle o33\rangle)\})$

nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{ROTAX(\langle q_u\rangle,\langle \vartheta_u\rangle) O(\langle \beta\rangle)\})$ nebo

$TcsDef(\{<x><px>\},\{<y><py>\},\{<z><pz>\},\{ROTAX(\langle u_x\rangle,\langle u_y\rangle,\langle u_z\rangle) O(\langle \beta\rangle)\})$

kde

$<x>$

Název souřadnice, která je zadána v MP 7080 00010[1]. Standardní: x

$<y>$

Název souřadnice, která je zadána v MP 7080 00010[2]. Standardní: y

$<z>$

Název souřadnice, která je zadána v MP 7080 00010[3]. Standardní: z

TcsDef, TCS

$\langle px \rangle, \langle py \rangle, \langle pz \rangle$	Absolutní kartézské souřadnic osových adres $\langle x \rangle, \langle y \rangle$ a $\langle z \rangle$, vztažené k aktuální soustavě souřadnic PCS. Hodnoty definují počátek nové soustavy souřadnic TCS _p . Programování AC/IC je dovoleno.
$\langle phi \rangle$	Název úhlu, který je zadán v MP 7080 00010[4]. Standardní: phi
$\langle theta \rangle$	Název úhlu, který je zadán v MP 7080 00010[5]. Standardní: theta
$\langle psi \rangle$	Název úhlu, který je zadán v MP 7080 00010[6]. Standardní: psi
$\langle \varphi \rangle, \langle \vartheta \rangle, \langle \psi \rangle$	Absolutní Eulerovy úhly ve stupních, vztažené k aktuální soustavě souřadnic PCS. Hodnoty definují orientaci nové soustavy souřadnic TCS _p . Programování AC/IC je dovoleno. Rozsah hodnot: $0^\circ \leq \varphi < 360^\circ$; $0^\circ \leq \vartheta \leq 180^\circ$; $0^\circ \leq \psi \leq 360^\circ$. Hodnoty mimo rozsah se automaticky přepočítávají do příslušného definičního intervalu.
$\langle \varphi.. \rangle, \langle \vartheta.. \rangle$	Absolutní úhlové hodnoty ve stupních. Rozsah hodnot: $0^\circ \leq \varphi.. < 360^\circ$; $0^\circ \leq \vartheta.. \leq 180^\circ$. Pokud je jako $\vartheta..$ naprogramována hodnota mimo rozsah, je automaticky přepočítána do zadaného intervalu.
Ox ($\langle o11 \rangle, \langle o21 \rangle, \langle o31 \rangle$) Oy ($\langle o12 \rangle, \langle o22 \rangle, \langle o32 \rangle$) Oz ($\langle o13 \rangle, \langle o23 \rangle, \langle o33 \rangle$) Ox ($\langle \varphi_x \rangle, \langle \vartheta_x \rangle$) Oy ($\langle \varphi_y \rangle, \langle \vartheta_y \rangle$) Oz ($\langle \varphi_z \rangle, \langle \vartheta_z \rangle$)	Orientierung per Funktion Ox(..), Oy(..), Oz(..). Ox(..) například definuje směr souřadnice x soustavy TCS _p ve vztažené soustavě souřadnic PCS. Analogické úvahy platí pro Oy(..) a Oz(..). Směr může být zadán buď odpovídajícími polárními úhly $\langle \varphi.. \rangle$ und $\langle \vartheta.. \rangle$, nebo pomocí kartézských složek sloupcových vektorů orientačního tenzoru TCS _p . Jsou dovoleny pouze absolutní rozměrové údaje. Hodnoty složek sloupcových vektorů (o..) jsou automaticky normovány na 1.
ROTAX ($\langle \varphi_u \rangle, \langle \vartheta_u \rangle$)	Definice osy otáčení pomocí polárních úhlů (φ_u, ϑ_u).
ROTAX ($\langle u_x \rangle, \langle u_y \rangle, \langle u_z \rangle$)	Definice osy otáčení pomocí kartézských složek $\langle u_x \rangle, \langle u_y \rangle, \langle u_z \rangle$ v absolutním rozměru. Automatické normování na 1.

TcsDef, TCS

o (< β >)

Udává inkrementální úhel < β > ve stupních, o který se má otočit orientační tenzor TCS_p , vycházející z orientačního tenzoru poslední aktuální soustavy TCS, kolem osy otáčení. Jsou povoleny hodnoty větší než 360 stupňů. Směr otáčení se vybírá pomocí znaménka.

Zvláštnosti a omezení:

- Je zapotřebí aktivní transformace os typu 3333301.
- Při novém programování funkce "Coord(...)" (viz strana 4-30) se dříve vytvořená soustava TCS_p automaticky vymaže a znovu se aktivuje poslední aktuální TCS.
- Pokud se při programování sloupcových vektorů tenzoru ukáže, že jsou dva sloupcové vektory paralelní nebo antiparalelní, není možné vypočítat orientační tenzor. Objeví se běhová chyba.

ThreadSet, TST

4.91 Přídavné funkce k řezání závitů

ThreadSet, TST

Následek

Pomocí funkce "ThreadSet" můžete dočasně přizpůsobit jednotlivé dílčí oblasti funkce G33.


Přitom potlačí řízení statické hodnoty uložené v parametrech stroje.

"ThreadSet" umožňuje

- Přizpůsobení dynamiky a zpětného pohybu
- Přepínání provozních režimů vřetena (regulace počtu otáček, regulace polohy)
- Nastavení signálu na rozhraní kanálu (možnost konfigurace v rámci signálů "aktivní funkce" podle parametrů stroje).

Základní nastavení nebo M30

- opět ruší nastavení daná funkcí "ThreadSet"
- maže signál rozhraní nastavený funkcí "ThreadSet"
- přepíná hlavní vřeteno zpět do provozu s regulací otáček, pokud bylo předtím přepnuto pomocí funkce "ThreadSet(SPC1)" do provozu s regulací polohy.

 **Všechny níže popsané dílčí funkce se dají naprogramovat také společně ve větě "ThreadSet", oddělené čárkou.**

Programování: konfigurace dat zpětného pohybu

Syntaxe:

ThreadSet (RD (<Hodnota HA> , <Hodnota NA>{, -1}))

Zkratka: **TST** (. .)

kde

<Hodnota HA> Dráha zpětného pohybu (inkrementální v mm) ve směru hlavní osy aktuálně vybrané roviny (G17, G18, G19, G20).

Hodnota musí být vždy naprogramována, ovšem význam má pouze u podélných a kónických závitů.

<Hodnota NA> Dráha zpětného pohybu (inkrementální v mm) ve směru vedlejší osy aktuálně vybrané roviny.

Hodnota musí být vždy naprogramována, ovšem význam má pouze u spirálových a kónických závitů.

-1

"-1" může být podle potřeby zadáno jako třetí parametr. V takovém případě jsou opět účinná data zpětného pohybu z MP 7050 00645 a MP 7050 00650.

ThreadSet, TST

Programování: uvolnění zpětného pohybu

Syntaxe:

ThreadSet (RON<Stav>) nebo**TST (..)**

kde

<Stav> 0: Vypnutí rychlého zpětného pohybu.
1: Zapnutí rychlého zpětného pohybu.

Programování: konfigurace dynamiky

Syntaxe:

ThreadSet (DYN({<Skok>},{<Zrychlení>}{,<Brzdné zrychlení>}))

nebo

TST (..)

kde

<Skok> max. povolená rychlost skoku v mm/min.
Při zadání "-1" platí opět MP 7050 00610.

<Zrychlení> Zrychlení v m/s².
Při zadání "-1" platí opět MP 7050 00615.

<Brzdné zrychlení> Brzdné zrychlení v m/s².
Při zadání "-1" platí opět MP 7050 00620.

Programování: přepnutí provozního režimu vřetena

Syntaxe:

ThreadSet (SPC<Stav>) nebo**TST (..)**

kde

<Stav> 0:
Zapnutí hlavního vřetena v provozu s regulací počtu otáček.

1:
Zapnutí hlavního vřetena podle nastavení 7050 00600 [3] do provozu s regulací polohy.

Informace o hlavních vřetenech viz strana 4-70.

Programování: ovlivňování signálu rozhraní kanálu

Syntaxe:

ThreadSet (TCI<Stav>) nebo**TST (..)**

kde

<Stav> 0: Maže signál rozhraní kanálu
1: Nastavuje signál rozhraní kanálu

 **Který signál na rozhraní kanálu bude ovlivňován, se dá zkonfigurovat v rámci signálů "Aktivní funkce" pomocí parametrů stroje.**

Trans, TRS ATrans, ATR

4.92 Posunutí programových souřadnic Aditivní posunutí programových souřadnic

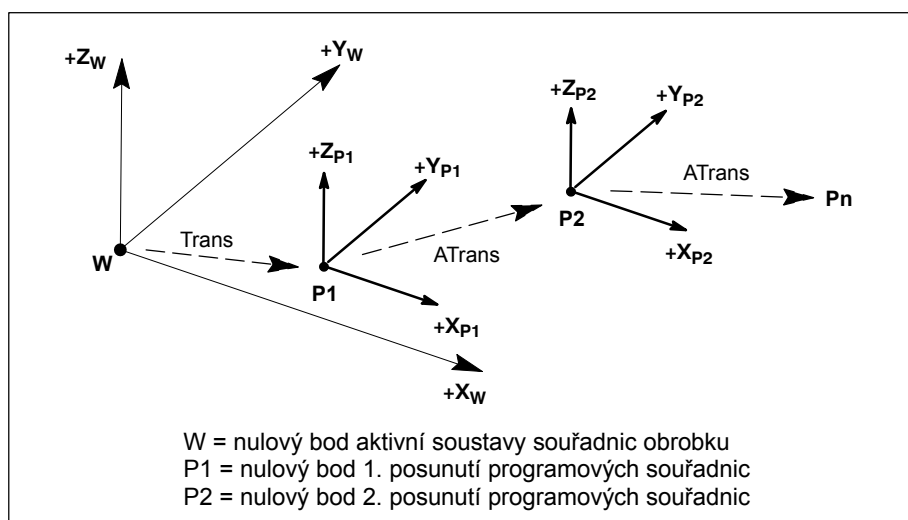
Trans, TRS
ATrans, ATR

Následek

"Trans" posouvá aktivní programovou soustavu souřadnic paralelně se souřadnicovými osami aktuální soustavy souřadnic obrobku.

"ATrans" umožňuje další paralelní, vždy aditivně působící posunutí programové soustavy souřadnic.

Tak lze provést dílčí program beze změny naprogramované kontury na libovolných místech v soustavě souřadnic obrobku. Pomůcky pro zadávání, které jsou založeny staví na aktivní programové soustavě souřadnic (např. naprogramované posunutí kontur, nastavení měřítka, zrcadlové převrácení, otáčení), jsou nezávislé na posunutí programových souřadnic a nemusí se proto také přizpůsobovat.



Platí:

- Naprogramování posunutí programových souřadnic nespouští žádný pojezdový pohyb.

Trans, TRS ATrans, ATR

Programování**Syntaxe:**

Trans(<Souřadnice>)

Zapnutí posunutí programových souřadnic.

Nulový bod programové soustavy souřadnic je nastaven na naprogramované <souřadnice> aktivní soustavy souřadnic obrobku.

Smí být naprogramováno několik souřadnicových os oddělených čárkou v závorce

(Příklad: TRS (X100 , Y50 , Z50)).

Trans() nebo
Trans(0)

Vypnutí všech posunutí programových souřadnic.

Zkratka: TRS(..)

ATrans(<Souřadnice>)

Zapnutí aditivního posunutí programových souřadnic.

Nulový bod výsledné programové soustavy souřadnic je nastaven na naprogramované <souřadnice> aktivní programové soustavy souřadnic.

Smí být naprogramováno několik souřadnicových os oddělených čárkou v závorce.

ATrans() nebo
ATrans(0)

Vypnutí aditivních posunutí programových souřadnic.

Zkratka: ATR(..)

Příklad:

N10 TRS (X10 , Y10 , Z50)

:

:

:

N50 ATR (X20 , Y10)

:

:

:

N80 ATR ()

:

:

:

:

N180 TRS ()

Nastavit nulový bod programové soustavy souřadnic do polohy X10 Y10 Z50 aktuální soustavy souřadnic obrobku.

Nulový bod výsledné programové soustavy souřadnic je v poloze X30 Y20 Z50 vzhledem k původní soustavě souřadnic obrobku.

Vypnutí aditivního posunutí programových souřadnic. Nulový bod programové soustavy souřadnic je opět v poloze X10 Y10 Z50 vzhledem k původní soustavě souřadnic obrobku.

Vypnutí všech posunutí programových souřadnic.

VirtAxisPos, VAP

4.93 Virtuální pohony

VirtAxisPos, VAP

Následek

Nastavuje polohu virtuálních synchronních os v aktuálním kanálu.

☞ **Další informace o virtuálních pohonech naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

VirtAxisPos (<Osa 1><Hodnota>,<Osa n><Hodnota>, ...)

Zkratka: **VAP** (...)

kde

<Osa x> Adresa virtuální osy.

<Hodnota> Polohová informace o <ose x>.

Příklad:

VAP(VX150) Polohovat osu VX na 150mm.

Zvláštnosti a omezení:

- Lokální inkrementální programování (IC...) není dovoleno.

WaitAxis, WAX

4.94 Převzetí osy, příp. čekání

WaitAxis, WAX

Následek

Přebírá asynchronní osu do volajícího kanálu. Z asynchronní osy se přitom stane synchronní.

Osa se pak dá naprogramovat v aktuálním kanálu pomocí svého fyzického nebo logického osového názvu.

 **Podrobné informace o funkci "Předávání os" naleznete v příručce "Funkční popis".**

Programování

Syntaxe:

WaitAxis(*<PAN>* | *<PAI>*, {*<LAN>*}{, *<PAN>* | *<PAI>*, {*<LAN>*}}...)

Zkratka: **WAX**(..)

kde

<i><PAN></i>	Fyzický název osy. Určuje osu, která má být převzata do aktuálního kanálu.
<i><PAI></i>	Fyzický index osy. Účinky jako <i><PAN></i> .
<i><LAN></i>	Logický název osy. Je-li naprogramován, dostává integrovaná osa v aktuálním kanálu logický název <i><LAN></i> . <i><LAN></i> musí být definováno v MP 7010 00010 (logické označení os) nebo MP 7010 00020 (volitelné označení os).

Zvláštnosti a omezení:

- Pokud přebíraná osa ještě není zastavená, čeká příprava vět, až se osa zastaví. Potom je osa převzata.
Na rozdíl od funkce "GetAxis" (viz strana 4-51) se neobjeví žádné chybové hlášení a nedojde k přerušení programu.
- Osové polohy ve stejné větě musí být vždy naprogramovány za WaitAxis(...) a smí být naprogramovány jedině tehdy, když není aktivní žádná transformace os.
- Přebírané osy nesmí být zúčastněny v žádné aktivní oblasti monitorování (viz strana 4-2).

Příklad:

```
N030 WAX(YP,,ZP,Z)
:
```

Příprava vět čeká případně na zastavení fyzických os YP a ZP. Potom jsou osy převzaty do volajícího kanálu. Zatímco osa YP je i ve volajícím kanálu označena jako adresa YP, ZP dostává ve volajícím kanálu adresu Z.

Writeld, WID

4.95 Zápis parametrů SERCOS

Writeld, WID

Následek

Zapíše naprogramované parametry pohonu SERCOS (S-x-xxxx, P-x-xxxx) do jednoho nebo více pohonů.

Pro parametry pohonu SERCOS, specifické pro určitý výrobek (P-x-xxxx), je možné zapisovat také celé seznamy hodnot maximálně se 16 hodnotami nebo seznamy ID maximálně s 8 identifikačními čísly.

Předpoklady:

- Parametr musí být platný a změnitelný ve fázi SERCOS 4.
- Hodnota parametru leží v povoleném rozsahu hodnot.
- Pohon je pomocí rozhraní SERCOS připojen k řízení a probíhá cyklická komunikace mezi řízením a pohonem.

 Pro čtení parametrů pohonu SERCOS je k dispozici příkaz CPL SCS.



NEBEZPEČÍ

Nesprávná nebo neodborná změna parametrů pohonu SERCOS může vést k poškození obrobku, resp. stroje a k nebezpečným či nepředvídatelným reakcím stroje.

Pro správné používání jsou zapotřebí informace o parametrech pohonu SERCOS, které jsou k dispozici v pohonu. Prostudujte si proto dokumentaci pohonu.

Programování

Zápis parametru do jednoho nebo několika pohonů (pro synchronní osy):

Syntaxe:

writeId(<Par>, <SA1><W1>{, <SAn><Wn>}...)

Zkratka: **WID(...)**

kde

<Par> Standardní parametr (S-x-xxxx) nebo parametr specifický pro výrobek (P-x-xxxx).

<SA1>...<SAn> Logická osová adresa pohonu SERCOS.

<W1>...<Wn> Hodnota, která má být zapsána do <Par>.

Příklad:

N100 WID(P-0-0500, X10, Y20, Z30) Zapsat P-0-0500:
v pohonu osy X: 10
v pohonu osy Y: 20
v pohonu osy Z: 30

WritelD, WID

Programování**Zápis parametru do jednoho nebo několika pohonů (pro vřetena a asynchronní osy):**

Syntaxe:

WriteId(*<Par>* **DRIVE**(*<AA1>*, *<W1>*{, *<AA*n*, *<W*n*

Zkratka: **WID**(...)

kde

<Par> Standardní parametr (S-x-xxxx) nebo parametr specifický pro výrobek (P-x-xxxx).

<AA1>...*<AA*n**> Fyzický osový index nebo fyzický osový název pohonu SERCOS. Pomocí syntaxe DRIVE(...) smí být definováno max. 8 pohonů.

<W1>...*<W*n**> Hodnota, která má být zapsána do *<Par>*.

Programování

Zápis hodnot seznamu parametrů do pohonu:

Syntaxe:

WriteId(*<Par>* **LIST**(*<A1>*, *<W1>*{, *<W*n**>}...))

Zkratka: **WID**(...)

kde

<Par> Parametr specifický pro výrobek (P-x-xxxx), který obsahuje seznam hodnot. (standardní parametry (S-x-xxxx) nejsou dovoleny!)

<A1>...*<A*n**> Fyzický osový index nebo fyzický osový název pohonu SERCOS.

<W1>...*<W*n**> Maximálně 16 hodnot, které mají být zapsány do *<Par>*.

Příklad:

N100 WID(P-0-0515 LIST(1,0.5,0.2) P-0-0515 v pohonu 1
popsat 2 hodnotami

Programování

Zápis identifikačních čísel ze seznamu ID do pohonu:

Syntaxe:

WriteId(*<Par>* **ID_LIST**(*<A1>*, *<Id1>*{, *<Id*n**>}...))

Zkratka: **WID**(...)

kde

<Par> Parametr specifický pro výrobek (P-x-xxxx), který obsahuje seznam ID. (standardní parametry (S-x-xxxx) nejsou dovoleny!)

<A1>...*<A*n**> Fyzický osový index nebo fyzický osový název pohonu SERCOS.

<Id1>...*<Id*n**> Maximálně 8 identifikačních čísel, která mají být zapsána do *<Par>*.**

Writel, WID

Příklad:

N100 WID(P-0-0417 ID_LIST(X,S-0-47,S-0-51) P-0-0417 v pohonu osy X popsat 2 identifikačními čísly.

Zvláštnosti a omezení:

- Funkce nesmí být naprogramována během obrábění kontur.
- Naprogramované parametry se zapisují až po zastavení příslušného pohonu.

ZoTSel, ZOS

4.96 Aktivace tabulek posunutí nulových bodů

ZoTSel, ZOS

Následek

Aktivuje tabulku posunutí nulových bodů (tabulka NPV). Tabulky NPV jsou uloženy jako soubory XML v systému souborů řízení.

Programování

Syntaxe:

zoTsel ({<Cesta>}<Název souboru>)

Zkratka: **ZOS** (. .)

kde

<Cesta> Volitelné zadání cesty k adresáři, v němž je uložen <název souboru>.

Bez zadání probíhá hledání v cestě "/database".

Pokud tam <název souboru> neexistuje, použije řízení vyhledávací cestu pro podprogramy a hledá <název souboru> také v jiných adresářích.

<Název souboru> Název souboru tabulky NPV včetně přípony souboru.

Tabulky se standardním názvem (ZO<číslo>.zot) mohou být aktivovány přímo pomocí čísla, např. ZoTSel(5) aktivuje tabulku ZO5.zot.

Zvláštnosti a omezení:

- Řízení může v případě potřeby při aktivaci tabulky kontrolovat, jestli se aktuální osová konfigurace kanálu hodí k obsahu tabulky (funkce "pevné přiřazení"; nastavuje se s použitím editoru tabulek nebo při prvním generování tabulky).
Pokud se při aktivované funkci "pevné přiřazení" nehodí, je program přerušen s chybovým hlášením.

 **Pokyny pro vytváření a úpravy tabulek NPV naleznete v návodu k obsluze řízení!**

Příklad:

:	
N030	ZOS(npvtab.zot)
:	
N130	ZOS(/mnt/np.zot)
:	

Hledá tabulku NPV "npvtab.zot" nejprve v adresáři "/database", a když tam neexistuje, ve vyhledávací cestě pro podprogramy. První nalezená tabulka NPV s názvem "npvtab.zot" je aktivována.

Hledá a aktivuje tabulku NPV "np.zot" v adresáři "/mnt". Jestliže tam není nalezena, objeví se chybové hlášení.

Hodnoty souřadnic a os

5 Funkce CPL


5.1 Hodnoty souřadnic a os

V CPL jsou k dispozici funkce, pomocí kterých je možné zjišťovat různé hodnoty souřadnic a os.

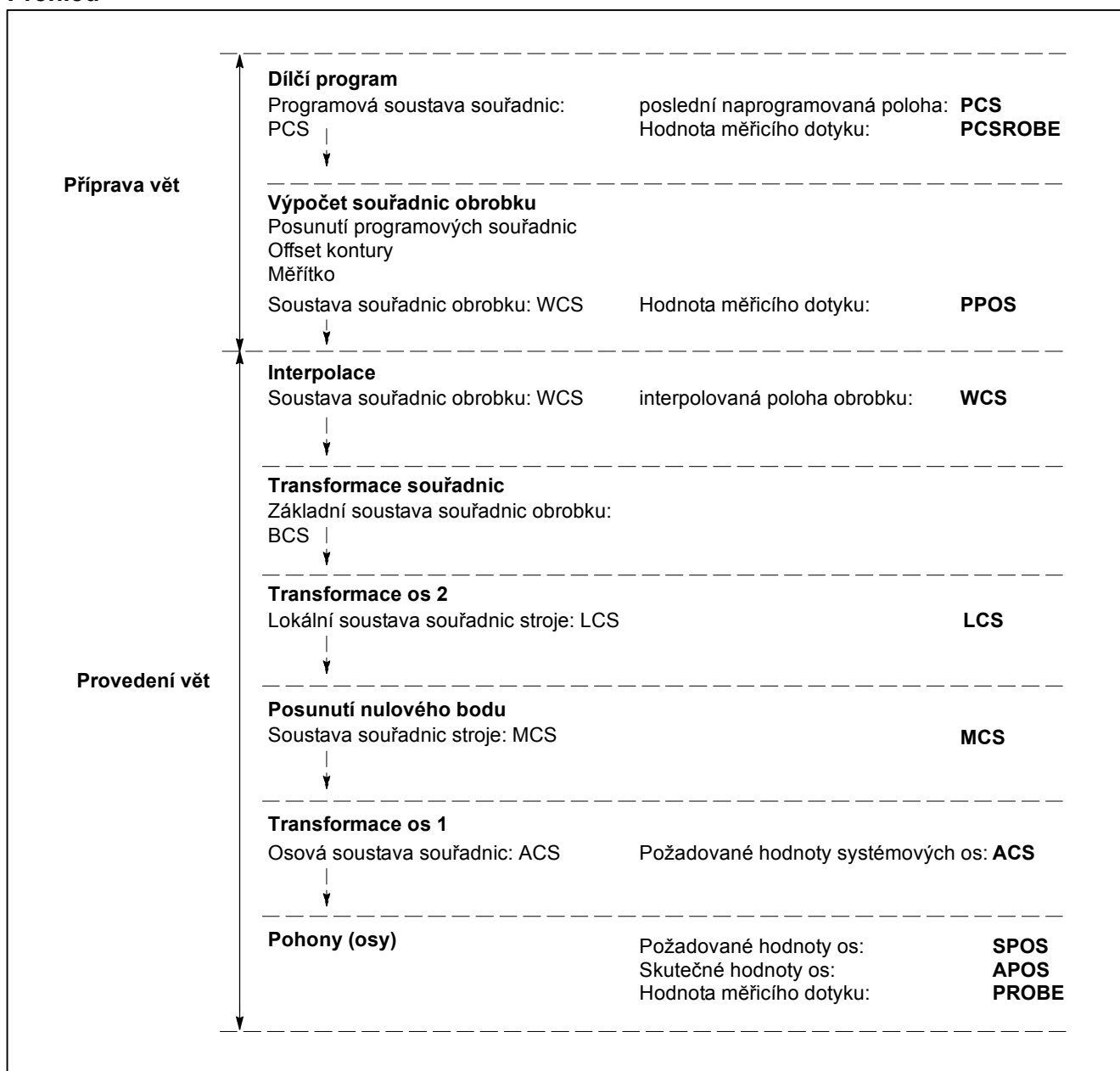
Přitom rozlišujeme:

- Funkce pro čtení poloh souřadnic a os (PCS, WCS, MCS, ACS, SPOS, APOS)
- Funkce pro čtení měřicího dotyku (PCSPROBE, PROBE, PPOS)
- Funkce pro čtení posunutí a umístění (AXO, COF, DPC).

V dílčím programu jsou vždy naprogramovány souřadnice, které jsou při zpracování programu interpolovány. Transformace os vypočítávají z aktuálních hodnot souřadnic požadované hodnoty pro příslušné osy (prostorové souřadnice a souřadnice stroje).

 **Podrobné informace k tématům "souřadnice, osy a transformace" naleznete v příručce "Funkční popis".**

Hodnoty souřadnic a os

Přehled**Měrné jednotky pro dodávané polohy os a souřadnic**

Synchronní lineární osy a
translační prostorové souřadnice:: "mm" nebo "Inch";
v závislosti na aktuálním
nastavení (G71, G70) ve vo-
laném kanálu.

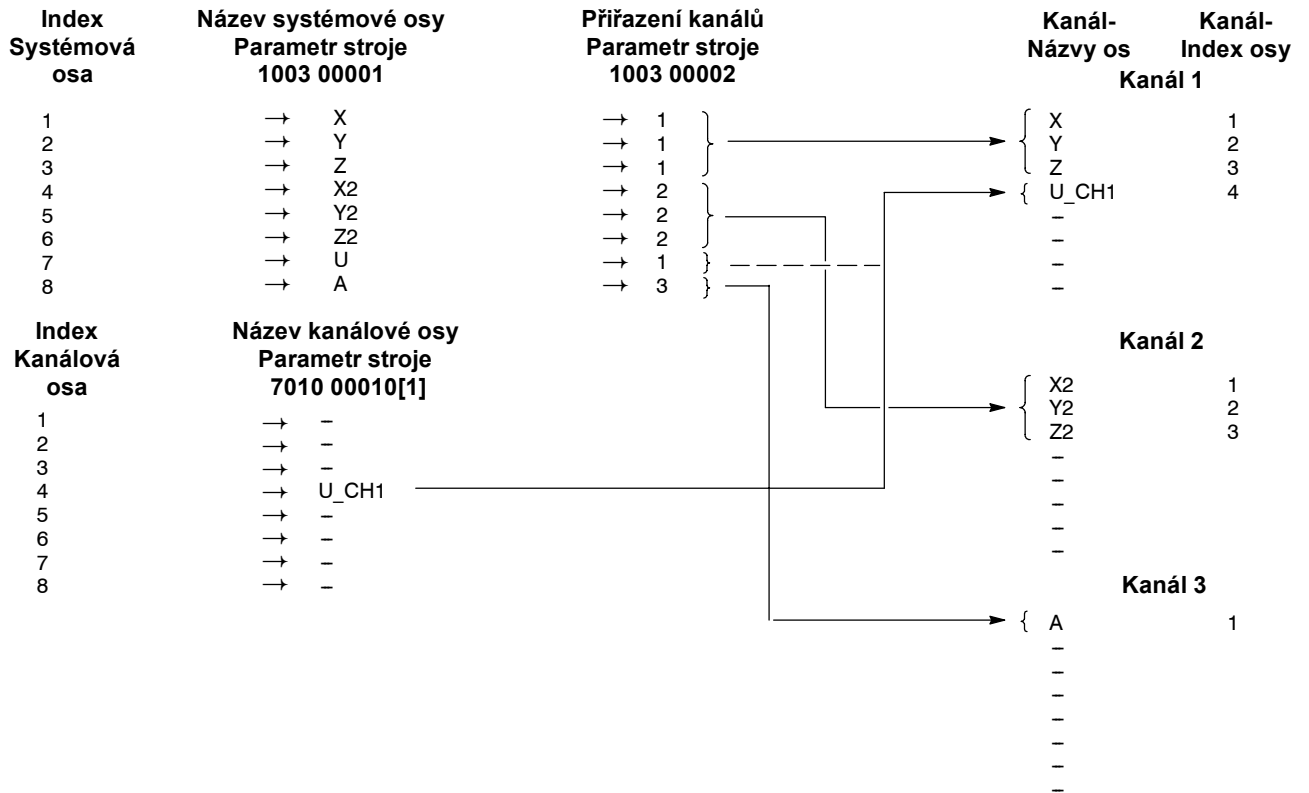
Synchronní kruhové osy a
rota D;ní prostorové souřadnice: "Stupně"

Asynchronní lineární osy: "mm"

Asynchronní kruhové osy: "Stupně"

Hodnoty souřadnic a os

Příklad: Konfigurace a přiřazení názvů os ke kanálům



Hodnoty souřadnic a os

5.1.1 Funkce pro čtení poloh souřadnic a os

Přitom si musíte pamatovat:

- Pro prostorové souřadnice a souřadnice stroje je index souřadnic v kanálu vždy pevný.
- Souřadnice mohou být naproti tomu přijímány nebo vydávány kanálem pomocí funkcí předávání os. Tak je možné změnit index jiných souřadnic v kanálu.
Díky možnosti předem zadat index systémových os lze pracovat také s pevnými indexy.
- Při zadání nezkonfigurované systémové osy je hlášena běhová chyba.

Popsané funkce pracují s podobnými parametry, které jsou proto vysvětleny v následujícím textu.

- <Souřadnice>** Index nebo název souřadnice:
Případný **název** je interpretován jako název souřadnice. Pouze když neexistuje žádný odpovídající název souřadnice, je interpretován jako název kanálové nebo systémové osy.
Případný **index** je interpretován podle zadaného *<typu výběru>*.
Naprogramování nezkonfigurované souřadnice, resp. osy vede k běhové chybě, pokud není naprogramována volitelná proměnná ERRNO.
- <Osa>** Index nebo název osy:
Případný **název** je interpretován jako název kanálové osy. Pokud žádný neexistuje, je přijat název systémové osy.
Případný **index** je interpretován podle zadaného *<typu výběru>*.
Naprogramování nezkonfigurované osy vede k běhové chybě, pokud není naprogramována volitelná proměnná ERRNO.
- <Typ výběru>** **volitelný:**
Určuje, jak bude interpretován index naprogramovaný pod *<souřadnicí>* nebo *<osou>*:
0: Index systémové osy
1: Index souřadnice nebo index kanálové osy (standardní)
Bez explicitního naprogramování se používá *<typ výběru>* = 1.

Hodnoty souřadnic a os

<Kanál>

volitelný:

Číslo kanálu, povoleno pouze ve spojení s <typem výběru> = 1.

Pokud jsou souřadnice/osy načítány cizími kanály a volány pomocí svého indexu nebo názvu, je v <kanálu> uvedeno číslo kanálu, kterému je souřadnice/osa aktuálně přiřazena.

Není-li zadán žádný kanál, používají se souřadnice/osy aktuálního kanálu. Pokud je adresována systémová osa (názvem nebo indexem) a současně je zadán kanál, je generováno chybové hlášení, i když je naprogramována proměnná ERRNO.

ERRNO

Proměnná CPL, programovatelná na libovolném místě.

Při použití ERRNO se při chybě negeneruje běhová chyba; vrácené hodnoty jsou:

0: Přístup OK

-1 : Chyba parametrů

-2 : Souřadnice/osa neexistuje.

-3 : Nepřípustná souřadnice/osa v kanálu.

-4 : Osa není žádná pseudosouřadnice.

-5 : Kanál neexistuje.

-6 : Funkce smí být vyvolána pouze ve vlastním kanálu.

-7 : Nelze načíst data.

Hodnoty souřadnic a os

PCS

Udává poslední naprogramovanou hodnotu pro souřadnici.

Platí:

- Jestliže mezi programováním souřadnice a zjišováním polohy dojde ke změně soustavy souřadnic obrobku, bere udaná hodnota v úvahu již novou soustavu souřadnic obrobku.
- Je možné zjišovat pouze souřadnice vlastního kanálu.

Syntaxe:

PCS (<souřadnice> [, <typ výběru>])

(Parametry viz strana 5-4,
programování ERRNO viz strana 2-56)

Příklady:

PCS ("X" , ERRNO)

PCS ("X" , 1)

Kanál 2 podle příkladu konfigurace na straně 5-3:

```
N1 G0 G90 X2=150 Y2=100
02 X2WERT=PCS(1,ERRNO)
03 IF ERRNO <>0 THEN
04 PRN#(0,"Chyba:",ERRNO)
05 ENDIF
```

Proměnné X2WERT je přiřazena naprogramovaná absolutní poloha 1. souřadnice v aktuálním kanálu (X2WERT = 150)

```
N10 G91 X2=10
011 X2WERT=PCS(1,1)
```

Proměnné X2WERT je přiřazena naprogramovaná absolutní poloha 1. souřadnice v aktuálním kanálu (X2WERT = 160)

```
N12 X2=5 Y2=10
13 Y2WERT=PCS("Y2",1)
```

Proměnné Y2WERT je přiřazena naprogramovaná absolutní poloha systémové osy Y2 (Y2WERT = 110)

```
14 X2WERT=PCS("X2")
```

Proměnné X2WERT je přiřazena naprogramovaná absolutní poloha souřadnice X2 v aktuálním kanálu (X2WERT = 165)

```
15 XWERT=PCS(1,0)
```

Běhová chyba: Přístup k 1. systémové ose v kanálu 2 není povolen (osa je přiřazena ke kanálu 1).

Hodnoty souřadnic a os

WCS

Udává aktuální polohu obrobku bez hodnot online opravy pro souřadnici.

Platí:

- Má-li být udaná hodnota určena v okamžiku provedení věty, musí být nejprve ve vlastní větě naprogramována funkce "WAIT" (viz též kapitola 2.13.2).
Bez funkce WAIT nejsou dodávány jednoznačně předvídatelné hodnoty, protože není přesně známo, o kolik se provádění vět "zpožiuje" za přípravou vět.
- Při přístupu k hodnotám souřadnic cizího kanálu je případně nutné provést synchronizační opatření pro změření definované polohy.

Syntaxe:

WCS (<souřadnice> [, <typ výběru> [, <kanál>]])

(Parametry viz strana 5-4
programování ERRNO viz strana 2-56)

Příklady:

WCS ("X", ERRNO)

WCS ("X", 1)

WCS ("X", 1, 1)

Kanál 2 podle příkladu konfigurace na straně 5-3:

```

10  WAIT
20  Z2POS=WCS(3,1,2,ERRNO)
21  IF ERRNO <>0 THEN
22  PRN#(0,"Chyba:",ERRNO)
23  ENDIF

30  WAIT
110 YPOS = WCS("Y")

120 XPOS = WCS(1)

```

Proměnné Z2POS je přiřazena aktuální interpolovaná poloha obrobku souřadnice 2. kanálu (osa Z2)

Proměnné YPOS je přiřazena aktuální interpolovaná poloha obrobku souřadnice Y aktuálního kanálu

Proměnné XPOS je přiřazena aktuální interpolovaná poloha obrobku 1. souřadnice aktuálního kanálu.

Hodnoty souřadnic a os

MCS

Udává aktuální polohu stroje (MCS) pro souřadnici stroje bez opravy chyby přímosti a úhlu.

Je možné používat také souřadnice cizích kanálů.

Platí:

- Má-li být udaná hodnota určena v okamžiku provedení věty, musí být nejprve ve vlastní větě naprogramována funkce "WAIT" (viz též kapitola 2.13.2).
Bez funkce WAIT nejsou dodávány jednoznačně předvídatelné hodnoty, protože není přesně známo, o kolik se provádění vět "zpožiuje" za přípravou vět.
- Při přístupu k hodnotám souřadnic cizího kanálu je případně nutné provést synchronizační opatření pro změření definované polohy.

Syntaxe:

MCS (<souřadnice>[, <typ výběru>[, <kanál>]])

(Parametry viz strana 5-4
programování ERRNO viz strana 2-56)

Příklady:

```
MCS ("X", ERRNO)
```

```
MCS ("X", 1)
```

```
MCS ("X", 1, 1)
```

Kanál 2 podle příkladu konfigurace na straně 5-3:

```
N10 G0 G90 X2=150 Y2=100
```

```
20 WAIT
```

```
30 X2WERT=MCS ("X2", ERRNO)
```

```
31 IF ERRNO <>0 THEN
```

```
32 PRN#(0, "Chyba:", ERRNO)
```

```
33 ENDIF
```

Proměnné X2WERT je přiřazena aktuálně interpolovaná poloha stroje souřadnice X2

```
N40 G91 X2=10 Y2=10
```

```
50 WAIT
```

```
60 X2WERT=MCS (1, 1)
```

Proměnné X2WERT je přiřazena aktuálně interpolovaná poloha stroje 1. souřadnice stroje aktuálního kanálu (souřadnice X2).

```
70 Y2WERT=MCS ("Y2", 1, 2)
```

Proměnné Y2WERT je přiřazena aktuálně interpolovaná poloha souřadnice Y2 2. kanálu.

```
80 XWERT=MCS (1, 0)
```

Proměnné XWERT je přiřazena aktuálně interpolovaná osová poloha 1. systémové osy.

Tento přístup je povolen, pouze když je systémová osa totožná se souřadnicí stroje.

Hodnoty souřadnic a os

ACS

Udává aktuální požadovanou polohu osy.
Osy mohou být volány pomocí názvu nebo indexu kanálové osy.
Alternativně může být osa volána také názvem nebo indexem systémové osy.

Platí:

- Výsledek ACS odpovídá vždy funkci SPOS.
- Má-li být udaná hodnota určena v okamžiku provedení věty, musí být nejprve ve vlastní větě naprogramována funkce "WAIT" (viz též kapitola 2.13.2).
Bez funkce WAIT nejsou dodávány jednoznačně předvídatelné hodnoty, protože není přesně známo, o kolik se provádění vět "zpožiuje" za přípravou vět.
- Při přístupu k hodnotám os cizího kanálu je případně nutné provést synchronizační opatření pro změření definované polohy.

Syntaxe:

ACS (<osa> [, <typ výběru> [, <kanál>]])

(Parametry viz strana 5-4
programování ERRNO viz strana 2-56)

Příklady:

ACS ("X", ERRNO)

ACS ("X", 1)

ACS ("X", 1, 1)

Kanál 2 podle příkladu konfigurace na straně 5-3:

```
N10 G0 G90 X2=150 Y2=100
20 WAIT
30 X2WERT=ACS("X2",ERRNO)
31 IF ERRNO <>0 THEN
32 PRN#(0,"Chyba:",ERRNO)
33 ENDIF
```

Proměnné X2WERT je přiřazena aktuálně interpolovaná osová poloha kanálové osy X2

```
N40 G91 X2=10 Y2=10
50 WAIT
60 X2WERT=ACS(1,1)
```

Proměnné X2WERT je přiřazena aktuálně interpolovaná osová poloha 1. kanálové osy aktuálního kanálu (osa X2).

```
70 Y2WERT=ACS("Y2",1,2)
```

Proměnné Y2WERT je přiřazena aktuálně interpolovaná osová poloha kanálové osy Y2 2. kanálu.

```
80 XWERT=ACS(1,0)
```

Proměnné XWERT je přiřazena aktuálně interpolovaná osová poloha 1. systémové osy.

Hodnoty souřadnic a os

SPOS

Udává aktuální **požadovanou polohu** osy vzhledem k osově soustavě souřadnic.

Je možné volat všechny systémové osy pomocí jejich názvu nebo indexu systémové osy.

Platí:

- SPOS a ACS dávají vždy stejné hodnoty.
- Má-li být udaná hodnota určena v okamžiku provedení věty, musí být nejprve ve vlastní větě naprogramována funkce "WAIT" (viz též kapitola 2.13.2).
Bez funkce WAIT nejsou dodávány jednoznačně předvídatelné hodnoty, protože není přesně známo, o kolik se provádění vět "zpožďuje" za přípravou vět.
- Při přístupu k hodnotám os cizího kanálu je případně nutné provést synchronizační opatření pro změření definované polohy.

Syntaxe:

SPOS (<osa>)

<Osa> Index nebo název systémové osy
 Naprogramování nezkonfigurované osy vede k
 běhové chybě.

ERRNO se nedá naprogramovat.

Příklad:

Kanály podle příkladu konfigurace na straně 5-3.

30 POS1=SPOS(1)	Proměnné POS1 je přiřazena aktuální
:	osová požadovaná hodnota 1. systémové
	osy (osa X v kanálu 1).
50 POS5=SPOS("Y2")	Proměnné POS5 je přiřazena aktuální
:	osová požadovaná hodnota 5. systémové
	osy (osa Y2 v kanálu 2).

Hodnoty souřadnic a os

APOS

Udává aktuální **skutečnou polohu** osy vzhledem k osově soustavě souřadnic.

Je možné volat všechny systémové osy pomocí jejich názvu nebo indexu systémové osy.

Platí:

- Má-li být udaná hodnota určena v okamžiku provedení věty, musí být nejprve ve vlastní větě naprogramována funkce "WAIT" (viz též kapitola 2.13.2).
Bez funkce WAIT nejsou dodávány jednoznačně předvídatelné hodnoty, protože není přesně známo, o kolik se provádění vět "zpožiuje" za přípravou vět.
- Při přístupu k hodnotám os cizího kanálu je případně nutné provést synchronizační opatření pro změření definované polohy.

Syntaxe:

APOS (<osa>)

<Osa> Index nebo název systémové osy
Naprogramování nezkonfigurované osy vede k
běhové chybě.

ERRNO se nedá naprogramovat.

Příklad:

Kanály podle příkladu konfigurace na straně 5-3.

30 AKT4=APOS (4)	Proměnné AKT4 je přiřazena aktuální osová skutečná hodnota 4. systémové osy (osa X2 v kanálu 2).
:	
50 AKT8=APOS ("A")	Proměnné AKT8 je přiřazena aktuální osová skutečná hodnota 8. systémové osy (osa A v kanálu 3).
:	

Hodnoty souřadnic a os

5.1.2 Funkce pro čtení měřicího dotyku

PCSPROBE


Pokud je k osám kanálu připojen spínací měřicí dotyk a bylo spuštěno měření, může PCSPROBE přečíst naměřenou hodnotu vždy pro jednu souřadnici.

Předpokladem pro PCSPROBE je, aby byl měřicí dotyk blokován všemi osami kanálu. Zablokované osově hodnoty se přepočítávají pomocí posledního aktivovaného transformačního řetězce na programové souřadnice. PCSPROBE udává hodnotu pro souřadnici.

Syntaxe:

PCSPROBE (<souřadnice> [, <typ výběru>])

(Parametry viz strana 5-4
programování ERRNO viz strana 2-56)

 **PCSPROBE se nedá používat společně s funkcí "průběžné měření" "FlyMeas" (FME), protože při "průběžném měření" je měřena vždy jen jediná osa.**

Příklad:

Kanál 1 podle příkladu konfigurace na straně 5-3:

```
N10 G75 X100 Y100 Z50
20 IF SD(9)=1 THEN
N30 (MSG, Měřicí dotyk se nevychýlil!)
40 GOTO .FEHLER
50 ELSE
60 ZMESS=PCSPROBE(3,ERRNO) Proměnné ZMESS je přiřazena hod-
61 IF ERRNO <>0 THEN nota 3. souřadnice naměřené polohy.
62 GOTO .Fehler
63 ENDIF
70 ENDIF
```

Hodnoty souřadnic a os

PROBE

Udává osově hodnoty, které se vztahují k osově soustavě souřadnic ASC.

Platí:

- PROBE umožňuje přístup pouze k osám vlastního kanálu. Proto nelze zjišťovat polohy asynchronních os.
- Jsou brány v úvahu pouze chyby stoupání vřetena a křížové kompenzace.
- Spuštění měřicího dotyku v kanálu se dá kontrolovat pomocí funkce SD(9).
- U funkce "Měření na pevném dorazu" FsProbe jsou naměřené hodnoty načítány funkcemi PPOS nebo PROBE.

Syntaxe:

PROBE (<osa> [, <typ výběru>])

(Parametry viz strana 5-4; ERRNO se nedá naprogramovat)

Příklad:

Kanál 2 podle příkladu konfigurace na straně 5-3:

```
N70 G75 Y2 250
```

```
80 IF SD(9)=1 THEN
```

```
N90 (MSG, Měřicí dotyk se nevychýlí!)
```

```
100 GOTO .FEHLER
```

```
110 ELSE
```

```
120 Y2MESS=PROBE(2)
```

Proměnné Y2MESS je přiřazena naměřená skutečná poloha 2. kanálové osy (zde: osa Y2 kanálu 2).

```
130 ENDIF
```

Hodnoty souřadnic a os

PPOS

Udává aktuální osovou skutečnou polohu synchronní osy ve spínacím bodě měřicího dotyku.

PPOS bere v úvahu následující opravy:

- Osová posunutí nulových bodů (G54...G59)
- Opravy nástroje (G48, ED)
- Posunutí programových souřadnic (Trans, ATrans)
- Chyba stoupání vřetena a křížová kompenzace

Bez povšimnutí zůstávají:

- Transformace os (Coord)
- Transformace souřadnic (BcsCorr, G152...G159)
- Nastavení měřítka (PoleSet, Mirror, Scale, Rotate)

Platí:

- PPOS umožňuje přístup pouze k osám vlastního kanálu. Proto nelze zjišovat polohy asynchronních os.
- Bez transformace os nebo transformace souřadnic se udaná hodnota vztahuje k poslední naprogramované soustavě souřadnic obrobku WCS.
- U funkce "Měření na pevném dorazu" FSB jsou naměřené hodnoty načítány funkcemi PPOS nebo PROBE.
- Spuštění měřicího dotyku v kanálu můžete kontrolovat pomocí funkce SD(9).

Syntaxe:

PPOS (<osa> [, <typ výběru>])

(Parametry viz strana 5-4; ERRNO se nedá naprogramovat)

Příklad:

Kanál 3 podle příkladu konfigurace na straně 5-3:

```
N10 G1 G75 A250 F500
20 IF SD(9)=1 THEN
N30 (MSG, Měřicí dotyk se nevychýlil!)
40 GOTO .FEHLER
50 ELSE
60 AMESS = PPOS(1,1)   Proměnné AMESS je přiřazena naměřená
                        hodnota 1. kanálové osy kanálu.
70 ENDIF
```

Hodnoty souřadnic a os

5.1.3 Funkce pro čtení posunutí a umístění


AXO

AXO udává aktuální posunutí G92 pro souřadnici v okamžiku přípravy věty, tzn. v okamžiku interpretace programu je dodáno poslední aktivované posunutí.

Syntaxe:

AXO (<souřadnice> [, <typ výběru>])

(Parametry viz strana 5-4; ERRNO se nedá naprogramovat)

 **AXO umožňuje přístup pouze k hodnotám posunutí vlastního kanálu. Asynchronní osy nemají žádné posunutí G92, a proto je funkce AXO pro asynchronní osy nepřipustná.**

Příklad:

Kanál 1 podle příkladu konfigurace na straně 5-3:

N10 G1 G90 X100 F1000

N20 G92 X75 Y125

30 XD = AXO("X")

40 YD = AXO(2,0)

50 X2D = AXO(4,0)

Proměnné XD je přiřazeno poslední aktivované posunutí G92 souřadnice X aktuálního kanálu (XD=100-75=25)

Proměnné YD je přiřazeno poslední aktivované posunutí G92 2. systémové osy (YD=200-125=75)

Běhová chyba, protože je 4. systémové ose přiřazen kanál 2.

Hodnoty souřadnic a os

COF

Udává poslední naprogramované konturové posunutí (Shift) souřadnice.

Protože naprogramované konturové posunutí působí pouze na souřadnice v aktuálním kanálu, dává volba souřadnice, která v aktuálním kanálu neexistuje, chybové hlášení.

Hodnoty opravy jsou udávány v aktivních měrných jednotkách aktuálního kanálu, tzn. u G70 v "Inch" a u G71 v "mm". U kruhových os, resp. rotačních prostorových souřadnic je jednotkou vždy "stupeň".

Syntaxe:

COF (<souřadnice> [, <typ výběru>])

(Parametry viz strana 5-4; ERRNO se nedá naprogramovat)

Příklady:

10	A=COF(3)	Udává poslední naprogramované konturové posunutí souřadnice s indexem 3. souřadnice v aktivním kanálu.
:		
20	B=COF("X")	Udává poslední naprogramované konturové posunutí souřadnice X v aktivním kanálu.
:		
30	C=COF(2,0)	Udává poslední naprogramované konturové posunutí 2. systémové osy v aktivním kanálu.
:		Tento přístup je povolen, pouze když je systémová osa totožná se souřadnicí WCS.
100	C=COF(0)	Běhová chyba, protože 0 není platný index souřadnice.
:		

Hodnoty souřadnic a os

DPC

Udává poslední naprogramované parametry opravy polohy obrobku (BCR) souřadnice (hodnoty posunutí a úhel otočení).

Protože oprava polohy obrobku působí pouze na souřadnice v aktuálním kanálu, dává volba souřadnice, která v aktuálním kanálu neexistuje, chybové hlášení.

Hodnoty opravy jsou udávány v aktivních měrných jednotkách aktuálního kanálu, tzn. u G70 v "Inch" a u G71 v "mm". U kruhových os, resp. rotačních prostorových souřadnic je jednotkou vždy "stupeň".

Syntaxe:

DPC (<souřadnice> [, <typ výběru>])

(Parametry viz strana 5-4; ERRNO se nedá naprogramovat)

<Souřadnice> Navíc působení zadaných údajů:
 "1"..."n" nebo "název": udává hodnotu posunutí
 "0" udává úhel otočení Phi.

Příklady:

10	A=DPC (1)	Udává poslední naprogramovanou opravu polohy obrobku souřadnice s indexem 1. souřadnice v kanálu.
:		
15	B=DPC (" X ")	Udává poslední naprogramovanou opravu polohy obrobku osy/souřadnice X v kanálu.
:		
20	B=DPC (2)	Udává poslední naprogramovanou opravu polohy obrobku souřadnice s indexem 2. souřadnice v kanálu.
:		
25	B=DPC (2 , 0)	Udává poslední naprogramovanou opravu polohy obrobku 2. systémové osy v aktivním kanálu.
:		
30	WINKEL=DPC (0)	Udává poslední naprogramovaný úhel otočení Phi.
:		
100	C=DPC (9)	Běžová chyba, protože při 8 osách v systému není 9 platný index souřadnice.
:		

Posunutí nulových bodů

5.2 Posunutí nulových bodů

Následující příkazy CPL umožňují číst, vytvářet a upravovat tabulky posunutí nulových bodů (NPV).

ZOV (ZeroOffsetValue)

Udává součet naposledy naprogramovaných a tedy efektivních hodnot NPV souřadnice stroje (osy) nebo alternativně efektivní hodnotu jednotlivé řady NPV.

Syntaxe:

ZOV (<výběr osy> [, <řada NPV>])

<Výběr osy> Název souřadnice stroje (osy), naprogramovaný v apostrofech nebo index souřadnice stroje (osy), hodnoty 1..8

<Řada NPV> Index řady NPV, hodnoty 1..5

Příklady:

ZOV ("X") Udává součet všech efektivních (naposledy naprogramovaných) NPV pro souřadnici stroje (osu) X.

ZOV (2, 3) Udává efektivní NPV 3. řady NPV pro 2. souřadnici stroje (osu).

ZOT ZeroOffsetTable

Čtecí a zápisový přístup do libovolné tabulky posunutí nulových bodů XML v rámci systému souborů IndraMotion MTX. Přístup je možný pouze k jednotlivým prvkům. Při zápisu je možné provádět také inkrementální změny.

Syntaxe:

ZOT (<výběr sloupce>, <kód NPV> [, <řada NPV>] [, <tabulka>] [, <jednotka>]]]])

<Výběr sloupce> Název souřadnice stroje (osy), naprogramovaný v apostrofech nebo index sloupce v tabulce.

Při zápisovém přístupu je možné:

- přepsat tabulkovou hodnotu, nebo
- přičíst hodnotu k tabulkové hodnotě, když <výběr sloupce> začíná znaménkem minus.

<Kód NPV> 54: 1. posunutí nulového bodu
55: 2. posunutí nulového bodu
56: 3. posunutí nulového bodu
57: 4. posunutí nulového bodu
58: 5. posunutí nulového bodu
59: 6. posunutí nulového bodu

Posunutí nulových bodů

<Řada NPV>	Index řady NPV (1...5). Standardní hodnota: 1
<Tabulka>	Název tabulky NPV, příp. s absolutním nebo inkrementálním zadáním cesty (není nutné programovat příponu tabulky ".zot"). Bez zadání cesty se prohledává zkonfigurovaná vyhledávací cesta. Pro tabulky se standardními názvy ZO1, ZO2 atd. je třeba zadat pouze číselnou hodnotu. Bez zadání názvu tabulky se používá poslední aktivovaná tabulka.
<Jednotka>	0 nebo "MM": mm 1 nebo "INCH": Inch Při zápisovém přístupu je přiřazená hodnota interpretována v zadaných jednotkách. Při čtecím přístupu je hodnota konvertována na zadanou jednotku. Standardní hodnota: mm

Příklady:

ZOT("X", 54)	Přístup k posunutí G54 osy / souřadnice stroje "X" v poslední aktivované tabulce NPV kanálu.
ZOT(2, 55, 2)	Přístup k posunutí G55.2 2. zadané osy / souřadnice stroje v poslední aktivované tabulce NPV kanálu.
ZOT(3, 57, , "V1")	Přístup k posunutí G57 3. zadané osy / souřadnice stroje v tabulce NPV "V1.zot".
ZOT("Z", 58, 4, 21, 1)	Přístup typu Inch k posunutí G58.4 osy / souřadnice stroje "Z" v tabulce NPV "ZO21.zot".
ZOT("Y", 59, 5, "/mnt/esmuser/de/HHGENIUS", "MM")	Metrický přístup k posunutí G59.5 osy / souřadnice stroje "Y" v tabulce "HHGENIUS" v adresáři "/mnt/esmuser/de".

Opravy nástroje

5.3 Opravy nástroje

Následující příkazy CPL umožňují přístup k interním datům oprav nástroje (D-korektury) a k externím datům oprav nástroje (ED-korektury), k nimž má přístup SPS.

TCV ToolCorrectionValue

Udává poslední naprogramované hodnoty oprav nástroje buď jako hodnotu součtu (D-korektura + externí pam oprav) nebo jako jednotlivou hodnotu.

Syntaxe:

TCV (<výběr hodnoty>[,<výběr opravy>])

<Výběr hodnoty>	Číslo D-/ED-korektury:
	1 nebo "L1": L1-Délka opravy
	2 nebo "L2": L2-Délka opravy
	3 nebo "L3": L3-Délka opravy
	4 nebo "RAD": Poloměr nástroje/břitů
	5 nebo "ORI": Poloha břitů
	6 nebo "PHI": Eulerův úhel φ (pouze ED-korektura)
	7 nebo "THE": Eulerův úhel θ (pouze ED-korektura)
	8 nebo "PSI": Eulerův úhel ψ (pouze ED-korektura)
<Výběr opravy>	1 nebo "D": D-korektura (interní)
	2 nebo "E": ED-korektura (externí)
	Standardní: Součet D- a ED- opravy

Příklady:

TCV ("L1")	Čtení součtu hodnot L1 poslední naprogramované D- a ED-korektury.
TCV (4, 1)	Čtení poloměru nástroje poslední naprogramované D-korektury
TCV (2, "E")	Čtení délky nástroje L2 poslední naprogramované ED-korektury.

Opravy nástroje

DCT D-CorrectionTable

Čtecí a zápisový přístup k hodnotám oprav nástroje v libovolných tabulkách D-korektur stejně jako k hodnotám externích oprav (ED-korektury). Při zápisu je možné provádět také inkrementální změny.

Syntaxe:

DCT (<výběr hodnoty>, <datový záznam>[, <tabulka>][, <jednotka>])

<Výběr hodnoty> Číslo D-/ED-korektury:

1 nebo "L1":	L1-Délka opravy
2 nebo "L2":	L2-Délka opravy
3 nebo "L3":	L3-Délka opravy
4 nebo "RAD":	Poloměr nástroje/břitů
5 nebo "ORI":	Poloha břitů
6 nebo "PHI":	Eulerův úhel φ (pouze ED-korektura)
7 nebo "THE":	Eulerův úhel ϑ (pouze ED-korektura)
8 nebo "PSI":	Eulerův úhel ψ (pouze ED-korektura)

Hodnota může být načtena nebo zapsána.

Při zápisovém přístupu je možné:

- přepsat tabulkovou hodnotu, nebo
- přičíst hodnotu k tabulkové hodnotě, když <Výběr hodnoty> začíná znaménkem minus.

Pro polohu břitů (5 nebo "ORI") není možný inkrementální zápis.

<Datový záznam> 1..99: při D-korektuře
1..16: při ED-korektuře

<Tabulka> Název tabulky D-korektur, příp. s absolutním nebo inkrementálním zadáním cesty (bez přípony tabulky ".dct"). Bez zadání cesty se prohledává zkonfigurovaná vyhledávací cesta. Pro tabulky se standardními názvy DC1, DC2 atd. je třeba zadat pouze číselnou hodnotu. Bez zadání <tabulky> se používá poslední vybraná tabulka D-korektur.

0: Přístup k ED-korektuře.

<Jednotka> 0 nebo "MM": mm (standardní)
1 nebo "INCH": Inch
Při zápisovém přístupu je přiřazená hodnota interpretována v zadaných jednotkách. Při čtecím přístupu je hodnota konvertována na zadanou jednotku.

Příklady:

DCT(1, 10, "K4")	Přístup k hodnotě opravy "L1" z datového záznamu 10 v tabulce D-korektur "K4".
DCT("RAD", 7)	Inkrementální zápis do poloměru nástroje datového záznamu 7 v poslední aktivované tabulce.
DCT("L2", 16, 0, "INCH")	Inchový přístup k opravě "L2" datového záznamu 16 externí opravy nástroje.

Databáze nástrojů

5.4 Databáze nástrojů

DBTAB

Je možné načíst celý datový záznam nebo dílčí strukturu databázové tabulky nástrojů do proměnné CPL, resp. je z proměnné opět zapsat do datového záznamu.

Syntaxe:

DBTAB (<DB tabulka>,<Klíč1>,<Klíč2>[,<ResVar>])

<DB tabulka> Název databázové tabulky s dílčími strukturami (Xpath-String). Jako oddělovací znaky mezi složkami struktury je možné používat "." nebo "/".

<Klíč1> Databázový klíč 1

<Klíč2> Databázový klíč 2

<ResVar> Proměnná typu Integer.
Když je zadána hodnota <ResVar>, negeneruje se při chybě přístupu běhová chyba. Jsou možné následující vrácené hodnoty:

0: Přístup OK

1: Chyba při přístupu

2: Nedostatek paměti

3: Nepřípustný typ proměnné

Pokud není zadána hodnota <ResVar>, generuje se při chybě přístupu běhová chyba.

Příklad:

```
10 SV.A=DBTAB("DBT1.Rec",1,1)
20 D!=DBTAB("DBT1.Rec.UD.Ed.Geo.
   L1",1,1)
30 SV.A.UD.Ed.Geo.L1 = 10
40 DBTAB("DBT1.Rec",1,1) = SV.A
```

DBSEA

Vyhledává datový záznam v databázové tabulce nástrojů.

Syntaxe:

DBSEA (<DB tabulka>, <Klíč1>, <Klíč2>, <vyhledávací podmínka>, <výsledek hledání>[, <ResVar>])

<DB tabulka> Název databázové tabulky.

<Klíč1> Počáteční klíč 1

<Klíč2> Počáteční klíč 2

<Vyhledávací podmínka> Vyhledávací kritérium jako String (viz níže)

<Výsledek hledání> Výsledek hledání, proměnná typu Integer.

0: Nebyl nalezen žádný datový záznam odpovídající vyhledávacímu kritériu

1: Nalezen datový záznam odpovídající vyhledávacímu kritériu.

<ResVar>: Proměnná typu Integer.

Když je zadána hodnota <ResVar>, negeneruje se při chybě přístupu běhová chyba. Jsou možné následující vrácené hodnoty:

0: Přístup OK

1: Chyba při přístupu

2: Nedostatek paměti

3: Nepřípustný typ proměnné

4: Chybné vyhledávací kritérium

Pokud není zadána hodnota <ResVar>, generuje se při chybě přístupu běhová chyba.

Vyhledávací kritérium (<vyhledávací podmínka>)

Je třeba dodržovat následující syntaxi:

- Je zadán seznam jednotlivých podmínek, které musí být všechny splněny.
- Jednotlivé podmínky jsou navzájem odděleny znakem ', '.
- Normální jednotlivá podmínka má tvar: <název příznaku><relační operátor><hodnota> kde:
 - <relační operátor>: "=" rovno
 - ">" větší
 - ">=" větší nebo rovno
 - "<" menší
 - "<=" menší nebo rovno
 - "<>" různě
- Pro SKQ jako relační operátor jsou přípustné pouze: "=" a "<>", např. SKQ=4711

Databáze nástrojů

- Jednotlivá podmínka bitové masky má tvar:
 - <název příznaku>=<hodnota>:<maska>
 - např. BQ1=0x307000:0xF0FF00
 - kde:

- <Hodnota> : Hodnota, kterou musí mít příslušné bity
- <Maska> : Stanovení příslušných bitů

Hodnoty a masky jsou interpretovány:

- jako hexadecimální číslo, když začínají "0x" nebo "0X"
- jako osmičkové číslo, když začínají "0"
- jinak jako desítkové číslo.

V <hodnotě> se nevyhodnocují mezery vepředu a vzadu. Pokud je zadán string, ve kterém mají tyto mezery význam, musí být dílčí string uzavřen v apostrofech (např.: SKQ=' ').

Chcete-li v dílčím stringu uzavřeném v apostrofech použít apostrof, musíte před něj umístit znak "\" (např.: SKQ='Achim\'s')

Okrajové podmínky:

- Smí se používat jen předem definované názvy příznaků
- Jsou možné pouze dotazy na rovnost.

Příklad:

```
K1=1
IKQ2=3
BQ2=0x1:0x1
```

Funkce CPL vrací první datový záznam, který odpovídá vyhledávacímu kritériu. V tomto případě vrací proměnná <výsledek hledání> hodnotu 1. Přitom začíná hledání u datového záznamu, který je definován oběma počátečními klíči. Když má jeden z počátečních klíčů hodnotu -1, začíná hledání u prvního datového záznamu v databázové tabulce.

Příklad:

```
10 SV.A.Hd=DBSEA("DBT1",-1,-1,
  "K1=1",FOUND%)
20 WHILE FOUND%=1 DO
30 SV.A.Hd=DBSEA("DBT1",SV.A.Hd.K1,
  SV.A.Hd.K2,"K1=1",FOUND%)
40 END
```

Umístění (šikmá rovina)

5.5 Umístění (šikmá rovina)

Následující příkazy CPL umožňují přístup k tabulkám umístění.

PMV **PlaceMentValue**

Udává součet posledních naprogramovaných a tedy efektivních hodnot umístění pro souřadnici. Alternativně lze načíst také efektivní hodnotu jednotlivé řady umístění.

Syntaxe:

PMV (<výběr souřadnice>[,<řada umístění>])

<Výběr souřadnice> Index souřadnice (1...6) nebo pevný identifikátor "X", "Y", "Z", "PHI", "THE", "PSI" pro souřadnice WCS.

<Řada umístění> Index řady umístění (1...5)

Příklady:

PMV ("Y")

Udává součet všech efektivních (naposledy naprogramovaných) umístění pro souřadnici Y. Kvůli možnému otočení se přitom nejedná jen o součet jednotlivých řad, ale také o výsledné posunutí ve směru "Y" základní soustavy souřadnic obrobku.

PMV (4, 4)

Udává Eulerův úhel PHI aktivního umístění 4. řady.

Umístění (šikmá rovina)

PMT PlaceMentTable

Čtecí a zápisový přístup do libovolné tabulky umístění XML v IndraMotion MTX, přístup pouze k jednotlivým prvkům. Při zápisu je možné provádět také inkrementální změny.

Syntaxe:

PMT (<výběr souřadnice>,<kód umístění>[,<řada umístění>][,<tabulka>]
[,<jednotka>]]])

<Výběr souřadnice> Index souřadnice (1...6) nebo pevný identifikátor "X", "Y", "Z", "PHI", "THE", "PSI" pro souřadnice WCS

Hodnota může být načtena nebo zapsána. Při zápisovém přístupu je možné:

- přepsat tabulkovou hodnotu, nebo
- přičíst hodnotu k tabulkové hodnotě, když <Výběr souřadnice> se znaménkem minus začíná.

<Kód umístění> 154: Umístění 1
155: Umístění 2
156: Umístění 3
157: Umístění 4
158: Umístění 5
159: Umístění 6

<Řada umístění> Index řady umístění (1...5)
Standardní hodnota: 1

<Tabulka>Název tabulky umístění, příp. s absolutním nebo inkrementálním zadáním cesty (není nutné programovat příponu tabulky ".pmt"). Bez zadání cesty se prohledává zkonfigurovaná vyhledávací cesta.

Pro tabulky se standardními názvy PM1, PM2 atd. je třeba zadat pouze číselnou hodnotu. Bez zadání názvu tabulky se používá poslední aktivovaná tabulka.

<Jednotka> 0 nebo "MM": mm
1 nebo "INCH": Inch

Pro 3 translační souřadnice se při zápisovém přístupu interpretuje přiřazená hodnota v zadané jednotce, při čtecím přístupu se hodnota převádí na zadanou jednotku.

Standardní hodnota: mm

Umístění (šikmá rovina)

Příklady:

PMT("Z",155)	Přístup k posunutí G155 souřadnice "Z" v poslední aktivované tabulce umístění kanálu.
PMT(5,157,3)	Přístup k Eulerovu úhlu θ pro G157.3 v poslední aktivované tabulce umístění kanálu.
PMT("X",154,, "P1")	Přístup k posunutí G154.1 souřadnice "X" v tabulce umístění P1.
PMT(3,158,4,,1)	Inchový přístup k posunutí G158.4 souřadnice "Z" v poslední aktivované tabulce umístění kanálu.
PMT("Y",159,5,"/mnt/esmuser/de/PL5","MM")	Metrický přístup k posunutí G159.5 souřadnice "Y" v tabulce "PL5" v adresáři "/mnt/esmuser/de".

Měřítka

5.7 Měřítka

SCL

Üdává pro aktuální kanál (zde: kanál, ve kterém běží program s příkazem SCL) poslední naprogramované parametry funkcí PLS, SCL, MIR a ROT(...) (souřadnice pólů, faktory měřítka a úhly otočení).

Protože funkce PLS, SCL, MIR a ROT(...) působí pouze na souřadnice v aktuálním kanálu, dává volba souřadnic, které v aktuálním kanálu neexistují, chybové hlášení.

Polohové hodnoty jsou udávány v aktivních měrných jednotkách aktuálního kanálu, tzn. u G70 v "Inch" a u G71 v "mm". U kruhových os, resp. u rotačních prostorových souřadnic je jednotkou vždy "stupeň".

Syntaxe:

SCL (<výběř>[, <souřadnic>[, <typ výběru>]])

<Výběř> 0: naposledy naprogramovaný úhel otáčení hlavní roviny
 1: naposledy naprogramovaný pól kanálové osy
 2: naposledy naprogramovaný faktor měřítka kanálové osy

<Souřadnice> Index nebo název souřadnice:
Případný **název** je interpretován jako název souřadnice. Pouze když neexistuje žádný odpovídající název souřadnice, je interpretován jako název kanálové nebo systémové osy.
Případný **index** je interpretován podle zadaného <typu výběru>.

Naprogramování nezkonfigurované souřadnice, resp. osy vede k běhové chybě, pokud není naprogramována volitelná proměnná ERRNO.

<Typ výběru> **volitelný:**
Určuje, jak bude interpretován index naprogramovaný pod <souřadnicí>:

0: Index systémové osy

1: index souřadnice (standardní)

Bez zadání <typu výběru> je index interpretován jako index souřadnice!

Příklady:

10 W=SCL(0)	Zapíše do proměnné W poslední naprogramovaný úhel otočení.
:	
20 P=SCL(1,2)	Zapíše do proměnné P pól souřadnice s 2. indexem souřadnice v kanálu.
:	
30 F=SCL(2,2,1)	Zapíše do proměnné F faktor měřítka souřadnice s 2. indexem souřadnice v kanálu.
:	
40 D=SCL(2,"X")	Zapíše do proměnné D faktor měřítka souřadnice X v aktivním kanálu.

Systémová data

5.8 Systémová data

5.8.1 Systémová data jednoduchých typů

NCF

Udává syntaxi NC-funkce, která byla naprogramována jako poslední v modální skupině, resp. skupině pomocných funkcí *<NC-funkce>*. Je možné dotazovat všechny modální skupiny řízení, stejně jako pomocné funkce, které jsou definovány ve skupinách pomocných funkcí

Proměnná, do které má být uložen výsledek, musí být typu "dimenzované znakové pole".

Typové konflikty jsou detekovány v průběhu programu a potvrzeny chybovým hlášením.

Syntaxe:

NCF (*<NC-funkce>*)

<NC-funkce>

Syntaxe libovolné NC-funkce nebo pomocné funkce.

Pokud je naprogramována neexistující syntaxe, objeví se běhová chyba.

Příklady:

10 DIM A\$(4)

:

20 A\$=NCF("G1")

:

N80 [A\$]

Dimenzovat znakové pole pro string o délce max. 4 znaky.

Proměnné String A\$ je přiřazena syntaxe poslední naprogramované NC-funkce ze skupiny, která obsahuje jako syntaxi "G1".

Dříve dotazovaná NC-funkce je nově naprogramována.

N10 M3 S1234 T2345

123 DIM A\$(4)

20 A\$=NCF("M3")

N30 [A\$]

Dříve dotazovaná NC-funkce je nově naprogramována.

50 PRN#(0,"Ve skupině M3
je ",A\$," ak-

tivní")

N30 M5

33 A\$=NCF("M3")

34 PRN#(0,"Ve skupině M3
je ",A\$," ak-

tivní")

N70 M30

Systémová data

- ☞ **Syntaxe vřetena IndraMotion MTX je volně konfigurovatelná a nastavuje se v parametrech stroje 1040 00101 ff.**

SCS

Umožňuje čtecí přístup k parametrům pohonu SERCOS z aktivní sady parametrů.

Syntaxe:

scs (<výběr osy>, <typ ID>, <ID-č.>[, <výsledná proměnná>])

<Výběr osy> Index nebo název systémové osy

<Typ ID> Výraz String.
"S": Parametr S
"P": Parametr P

<ID-č.> Číslo parametru SERCOS

<Výsledná proměnná>: Když je zadána <výsledná hodnota>, není při chybě přístupu generována běhová chyba:

Je možná následující vrácená hodnota:

0: Přístup OK

1: Přístup v tomto okamžiku není možný

Pokud není zadána <výsledná proměnná>, generuje se při chybě přístupu běhová chyba.

<Výsledná proměnná> je proměnná Integer.

Obsah parametru je udáván bez jednotky a vážení.

- ☞ **Parametry, které obsahují seznam (více hodnot oddělených čárkou), se nedají číst. Řízení v takových případech generuje chybové hlášení.**

Pokud se v telegramu pohonu SERCOS nacházejí data pohonu, jsou z něho přečtena (viz příručka parametrů Servodyn-D). Jinak se data pohonu načítají přímo v pohonu.

Jsou-li data pohonu používána jinými aplikacemi, není přístup k datům pohonu v **tomto** okamžiku možný. Na tento případ chyby je možné reagovat v dílčím programu parametrem <výsledná proměnná>. Opakovaný přístup může poskytnout požadovaná data pohonu.

- ☞ **Trvalý přístup k datům pohonu může zabránit přístupu pro jiné aplikace!**

Systémová data

Přístup k datům pohonu	LAGE%	FEHLER%	Chybové hlášení
ano	nová skutečná hodnota polohy i-té osy	0	žádné
ne	stará skutečná hodnota polohy i-té osy zůstane zachována	1	SERCOS SERVICE-KANAL IST BELEGT

Na základě vyhodnocení proměnné Integer FEHLER% je možné v dílčím programu reagovat na chybový případ.

Příklad:

```

10 LAGE%=SCS(1,"S",51,FEHLER%)
12 IF FEHLER% = 0 THEN
13   REM***Lze správně přečíst skutečnou hodnotu polohy***
14   ELSE
15   REM***Nelze přečíst skutečnou hodnotu polohy***
16 ENDIF

```

Proměnné Integer LAGE% je přiřazena skutečná hodnota polohy 1. osy.
Vyhodnocení chyby

SCSL

Některé parametry SERCOS jsou uloženy v pohonu jako seznamy, které mohou být přečteny příkazem SCSL. Protože délka (potřebná velikost paměti) seznamu není známá, ukládají se přečtené prvky seznamů do souborů ASCII. Následně mohou být načtená data zpracována pomocí příkazů pro soubory CPL.

Příkazem SCSL se nově vytváří soubor uvedený v příkazu, pokud ještě neexistuje. Obsah již existujícího souboru se přepíše.

Syntaxe:

SCSL (<index osy>, <typ ID>, <ID-č.>, <název souboru>[, <výsledná proměnná>])

<Index osy> Index nebo název systémové osy

<Typ ID> Výraz String.
"S": Parametr S
"P": Parametr P

<ID-č.> Číslo parametru SERCOS

<Název souboru> Název souboru ASCII, ve kterém má být uložen přečtený seznam.

<Výsledná proměnná> Proměnná Integer

Když je zadána hodnota <výsledná proměnná>, ne-generuje se při chybě přístupu běhová chyba.


Systémová data

Je možná následující vrácená hodnota:

- 0: Přístup OK
- 1: Přístup do SERCOS v tomto okamžiku není možný
- 2: Přístup do souboru je chybný.

Pokud není zadána <výsledná proměnná>, generuje se při chybě přístupu běhová chyba.

Přístup k datům pohonu nemusí být v určitých okamžicích možný, pokud jsou právě používána jinými aplikacemi. Pomocí parametru <výsledná proměnná> reaguje dílčí program na tento chybový případ. Opakovaný přístup může poskytnout požadovaná data pohonu.

 **Trvalý přístup k datům pohonu může zabránit přístupu pro jiné aplikace.**

Systémová data

SD

Čte ze souboru systémových dat IndraMotion MTX **ve formátu Integer**.

Syntaxe:

SD (<skupina>[,<index1>[,<index2>[,<index3>]]])

Funkce SD dává hodnoty INTEGER.

Skupina	Index1	Index2	Index3	Týká se funkce	Popis
2	1				Aktivní poloha Override pro příslušný potenciometr v procentech:
2	2				• Posuv
2	3				• Rychlý chod
2	4				• Vřetenno ($\pm \mp \leq . \approx \pm SD(202,1)$; SD(2,3)=0, pokud není aplikováno žádné vřetenno)
2					• 2. vřetenno ($\pm \mp \leq . \approx \pm SD(202,2)$; (SD(2,4)=0, pokud není aplikováno žádné 2. vřetenno)
5	1	1			Aktivní rychlosti, zaokrouhlené na celočíselnou hodnotu:
5	2	1			• Posuv v zadané jednotce za minutu; hodnoceno potenciometrem (při OvrEna dává SD hodnotu 100%)
5	3	1			• Rychloběh v mm/min, resp. Inch/min (hodnota 100%)
5	4	1			• Počet otáček vřetenno v ot./min; hodnocený potenciometrem ($\pm \mp \leq . \approx \pm SD(205,1,1)$; SD(5,3,1)=0, pokud není aplikováno žádné vřetenno)
5					• Počet otáček 2. vřetenno v ot./min; hodnocený potenciometrem ($\pm \mp \leq . \approx \pm SD(205,1,2)$; SD(5,4,1)=0, pokud není aplikováno žádné 2. vřetenno)
5	1	2			Poslední naprogramované rychlosti:
5	3	2			• Posuv v zadané jednotce za minutu
5	4	2			• Počet otáček vřetenno v ot./min; hodnocený potenciometrem ($\pm \mp \leq . \approx \pm SD(205,2,1)$; SD(5,3,2)=0, pokud není aplikováno žádné vřetenno)
5					• Počet otáček 2. vřetenno v ot./min; hodnocený potenciometrem ($\pm \mp \leq . \approx \pm SD(205,2,2)$; SD(5,4,2)=0, pokud není aplikováno žádné 2. vřetenno)
5	3	3			Skutečný počet otáček ($\pm \mp \leq . \approx \pm SD(205,3,1)$)
5	4	3			Skutečný počet otáček 2. vřetenno ($\pm \mp \leq . \approx \pm SD(205,3,2)$)
8					Udává číslo volaného kanálu.

* : Číslo kanálu:

Pokud zadaný kanál není aktivní, mohou být osy z tohoto kanálu již zapůjčeny, tzn. jsou v tomto okamžiku aktivní v jiném kanálu. Přesto patří zapůjčené osy k uvedenému kanálu.

Příklad: Osa X2 patří ke kanálu 2 (neaktivnímu) a X2 v tomto okamžiku synchronně pojíždí v kanálu 1. V obou instrukcích SD "Počet synchronních os kanálu" SD(21,2,...) a SD(21,1,...) je brána v úvahu osa X2.

Systémová data

Skupina	Index1	Index2	Index3	Týká se funkce	Popis
9				G75 FsProbe	Měřicí dotyk zapnutý: SD(9)=0 (G75 neaktivní) Měřicí dotyk není zapnutý: SD(9)=1 (G75 aktivní) Měření na pevném dorazu provedeno: SD(9)=0 (FSB neaktivní) Měření na pevném dorazu ještě neprovedeno: SD(9)=1 (FSB aktivní)
10	1,2	1,2,3			Index 1: 1= č. naposledy naprogramované osy vrtání 2= č. aktivní osy vrtání Index 2: 1= osa, na kterou působí oprava L1 2= osa, na kterou působí oprava L2 3= osa, na kterou působí oprava L3 SD(10, i) = SD(10, i, 3)
11	1	1			Hlavní osa posledního naprogramovaného přepnutí roviny
11	2	1			Vedlejší osa posledního naprogramovaného přepnutí roviny
11	1	2			Hlavní osa aktivní roviny
11	2	2			Vedlejší osa aktivní roviny
12	1				Aktivní směr otáčení vřetena ($\pm \mp \leq \cdot \approx \pm$ SD(212,1,1): SD(12,1)= 3 pravotočivý chod vřetena SD(12,1)= 4 levotočivý chod vřetena SD(12,1)= 0 zastavení vřetena SD(12,1)= -1 vřeteno není aplikováno SD(12,1)= 19 seřízení vřetena
12	2				Poslední naprogramovaný směr otáčení vřetena ($\pm \mp \leq \cdot \approx \pm$ SD(212,2,1); Funkce jako u "aktivního směru otáčení vřetena") Není bráno v úvahu přepnutí aktivního směru otáčení vřetena signálem rozhraní!
12	3				Aktivní směr otáčení vřetena (2. vřeteno) ($\pm \mp \leq \cdot \approx \pm$ SD(212,1,2); Funkce jako u "aktivního směru otáčení vřetena")
12	4				Poslední naprogramovaný směr otáčení vřetena (2. vřeteno) ($\pm \mp \leq \cdot \approx \pm$ SD(212,2,2); Funkce jako u "aktivního směru otáčení vřetena")
13					Typy obrábění při zpracování v okamžiku interpretace: SD(13)= 0 Jednotlivá věta, jednotlivý krok SD(13)= 1 Následná věta SD(13)= 2 Programová věta SD(13)= 11 Průběh po větách s následnou větou (vybraná věta ještě není interpretována).

* : Číslo kanálu:

Pokud zadaný kanál není aktivní, mohou být osy z tohoto kanálu již zapůjčeny, tzn. jsou v tomto okamžiku aktivní v jiném kanálu. Přesto patří zapůjčené osy k uvedenému kanálu.

Příklad: Osa X2 patří ke kanálu 2 (neaktivnímu) a X2 v tomto okamžiku synchronně pojíždí v kanálu 1. V obou instrukcích SD "Počet synchronních os kanálu" SD(21,2,...) a SD(21,1,...) je brána v úvahu osa X2.

Systémová data

Skupina	Index1	Index2	Index3	Týká se funkce	Popis
14					Číslo aktivního národního jazyka (parametr stroje 6010 00010)
15					Test bez pohybu SD(15)=0 : ne SD(15)=1 : ano
20	1, 2				Udává počet synchronních os volajícího kanálu: SD(20,1) = Hodnota (standardní hodnota) v okamžiku přípravy vět SD(20,2) = Hodnota v aktivním okamžiku.
21	1...n*	1, 2			Udává počet synchronních os kanálu: SD(21, <1..n> ,1) = Hodnota (standardní hodnota) v okamžiku přípravy vět SD(21, <1..n> ,2) = Hodnota v aktivním okamžiku. 1...n = číslo kanálu, n = max. počet kanálů
22	1..m nebo označení kanálové osy	1, 2			Udává číslo kanálové osy systémové osy volajícího kanálu nebo -1: SD(22, <1..m> String ,1) = Hodnota (standardní hodnota) v okamžiku přípravy vět. SD(22, <m> String ,2) = Hodnota v aktivním okamžiku. 1..m = číslo systémové osy; m = max. počet systémových os String = název systémové osy
23	1..m nebo označení kanálové osy	1, 2			Udává číslo systémové osy kanálové osy volajícího kanálu nebo -1: SD(23, <1..m> String ,1) = Hodnota (standardní hodnota) v okamžiku přípravy vět. SD(23, <1..m> String ,2) = Hodnota v aktivním okamžiku. 1..m = číslo kanálové osy; m = max. počet kanálových os String = název kanálové osy

* : Číslo kanálu:

Pokud zadaný kanál není aktivní, mohou být osy z tohoto kanálu již zapůjčeny, tzn. jsou v tomto okamžiku aktivní v jiném kanálu. Přesto patří zapůjčené osy k uvedenému kanálu.

Příklad: Osa X2 patří ke kanálu 2 (neaktivnímu) a X2 v tomto okamžiku synchronně pojíždí v kanálu 1. V obou instrukcích SD "Počet synchronních os kanálu" SD(21,2,...) a SD(21,1,...) je brána v úvahu osa X2.

Systémová data

Skupina	Index1	Index2	Index3	Týká se funkce	Popis
24	1...m nebo označ# x10D;en í kanálové osy	1...n*	1, 2		<p>Udává číslo systémové osy kanálové osy nebo -1: SD(24, <1..m> String, <1..n>, 1) = hodnota (standardní hodnota) v okamžiku přípravy věty. SD(24, <1..m> String, <1..n>, 2) = hodnota v aktivním okamžiku.</p> <p>1...n = číslo kanálu, n = max. počet kanálů 1..m = číslo kanálové osy; m = max. počet kanálových os String = název kanálové osy</p>
25	1...m nebo ozna- čení kanálové osy				<p>Udává kanál systémové osy: SD(25, <1..m> String) = Hodnota v aktivním okamžiku. 1..m = číslo systémové osy; m = max. počet systémových os String = název systémové osy</p>
68	1	1...8		Trans, ATrans	<ul style="list-style-type: none"> • Součet naposledy naprogramovaných posunutí programových souřadnic (Trans + ATrans) pro uvedenou osu (Index2). • Součet aktivních posunutí programových souřadnic (Trans + ATrans) pro uvedenou osu (Index2). • Hodnota naposledy naprogramovaného posunutí programových souřadnic (Trans) pro uvedenou osu (Index2). • Hodnota aktivního posunutí programových souřadnic (Trans) pro uvedenou osu (Index2). • Hodnota naposledy naprogramovaného aditivního posunutí programových souřadnic (ATrans) pro uvedenou osu (index2). • Hodnota aktivního aditivního posunutí programových souřadnic (Arans) pro uvedenou osu (index2).
68	2	1...8			
168	1	1...8			
168	2	1...8			
268	1	1...8			
268	2	1...8			
77					<p>Udává aktuálně nastavený bod najetí asynchronního programu:</p> <p>1: Počáteční bod 2: Koncový bod 3: Bod přerušení</p>
131	1			TangTool	Číslo kanálové osy rotační osy
131	2				Symetrie

* : Číslo kanálu:

Pokud zadaný kanál není aktivní, mohou být osy z tohoto kanálu již zapůjčeny, tzn. jsou v tomto okamžiku aktivní v jiném kanálu. Přesto patří zapůjčené osy k uvedenému kanálu.

Příklad: Osa X2 patří ke kanálu 2 (neaktivnímu) a X2 v tomto okamžiku synchronně pojíždí v kanálu 1. V obou instrukcích SD "Počet synchronních os kanálu" SD(21,2,...) a SD(21,1,...) je brána v úvahu osa X2.

Systémová data

Skupina	Index1	Index2	Index3	Týká se funkce	Popis	
200	1			Pracovní oblast, mrtvá oblast	<ul style="list-style-type: none"> Počet aktivních oblastí v kanálu. Udává, zda je oblast i (index2) aktivní: 0: Oblast i (index2) není aktivní Typ oblasti i: 0: Typ není definován 1: Mrtvá oblast 2: Pracovní oblast Poloha středu oblasti i (index2) v programovacích jednotkách (pro 1. osu oblasti). Poloha středu oblasti i (index2) v programovacích jednotkách (pro 2. osu oblasti). Expanze oblasti i (index2) v programovacích jednotkách (pro 1. osu oblasti). Expanze oblasti i (index2) v programovacích jednotkách (pro 2. osu oblasti). 	
200	1	1...10				
200	3	1...10				
200	11	1...10				
200	12	1...10				
200	21	1...10				
200	22	1...10				
				Vřetena	1...8 = Číslo vřetena Aktuální hodnota potenciometru Aktivní požadovaný počet otáček (vč. potenciometru) Poslední naprogramovaný požadovaný počet otáček Skutečný počet otáček Aktivní pohybová funkce Naposledy naprogramovaná pohybová funkce	Staré, nadále platné SD-funkce: SD(2,3), SD(2,4) SD(5,3,1), SD(5,4,1) SD(5,3,2), SD(5,4,2) SD(5,3,3), SD(5,4,3) SD(12,1), SD(12,3) SD(12,2), SD(12,4)
202	1...8					
205	1	1...8				
205	2	1...8				
205	3	1...8				
212	1	1...8				
212	2	1...8				

* : Číslo kanálu:

Pokud zadaný kanál není aktivní, mohou být osy z tohoto kanálu již zapůjčeny, tzn. jsou v tomto okamžiku aktivní v jiném kanálu. Přesto patří zapůjčené osy k uvedenému kanálu.

Příklad: Osa X2 patří ke kanálu 2 (neaktivnímu) a X2 v tomto okamžiku synchronně pojíždí v kanálu 1. V obou instrukcích SD "Počet synchronních os kanálu" SD(21,2,...) a SD(21,1,...) je brána v úvahu osa X2.

Systémová data

Skupina	Index1	Index2	Index3	Týká se funkce	Popis
328				PrecProg	Naposledy naprogramovaná mez přesnosti PrecProg
328	1				Naposledy naprogramovaná mez přesnosti PrecProg
328	2				Naposledy naprogramovaná rohová vzdálenost PrecProg
				Spojení os	m: Číslo kanálové osy hlavního zařízení v aktuálním kanálu s: Číslo kanálové osy podřízeného zařízení v aktuálním kanálu
581	0	1...8(m)			<ul style="list-style-type: none"> • Číslo kanálové osy m, když je osa m řídicí 0: když neexistuje žádná řídicí osa
581	1...8(s)	0			<ul style="list-style-type: none"> • Číslo řídicí osy, ke které je osa s následnou osou 0: když s není následná osa
581	1...8(s)	1			<ul style="list-style-type: none"> • Naprogramované posunutí následné osy v programovacích jednotkách. 0: když s není následná osa
581	1...8(s)	2			<ul style="list-style-type: none"> • Naprogramovaný faktor spojení 0: když s není následná osa
581	1...8(s)	3			<ul style="list-style-type: none"> • Naprogramované posunutí řídicí osy v programovacích jednotkách 0: když s není následná osa

* : Číslo kanálu:

Pokud zadáný kanál není aktivní, mohou být osy z tohoto kanálu již zapůjčeny, tzn. jsou v tomto okamžiku aktivní v jiném kanálu. Přesto patří zapůjčené osy k uvedenému kanálu.

Příklad: Osa X2 patří ke kanálu 2 (neaktivnímu) a X2 v tomto okamžiku synchronně pojíždí v kanálu 1. V obou instrukcích SD "Počet synchronních os kanálu" SD(21,2,...) a SD(21,1...) je brána v úvahu osa X2.

Příklady:

30 A% = SD(2,1) A% obsahuje aktivní polohu posuvového
: potenciometru v procentech
40 B% = SD(5,1,1) B% obsahuje aktivní rychlost posuvu
:

Příklad programu: SD (dotaz měřicího dotyku)

```

N4  G75 X120
60  IF SD(9)=1 THEN
N7    (MSG, Měřicí dotyk se nevychýlí.)
80  GOTO .FEHLER
90  ELSE
100 XMESS = PPOS(1)
110 ENDIF

```

V příkladu SD (dotaz měřicího dotyku) pojíždí osa X ve směru zadané polohy. Pokud je dosažena poloha a měřicí dotyk se nevychýlí, je vydáno hlášení (řádek N7) a proběhne skok na návěští .FEHLER. Jestliže se měřicí dotyk vychýlí, je možné uložit aktuální polohu vzhledem k programové soustavě souřadnic v XMESS.

Systémová data

SDR

Čte ze souboru systémových dat IndraMotion MTX **ve formátu REAL**.
Formát příkazu a používání odpovídají SD-funkci.

Syntaxe:

SDR (<skupina> [, <index1> [, <index2>]])

Skupina	Index1	Index2	Týká se funkce	Popis
1	1...8			Polohy všech obráběcích os, které byly vypočítány při zavádění, resp. opakovaném vstupu vět (index1 = číslo osy). Bez zavádění vět je vrácena 0. Pokud je volána neaplikovaná nebo pomocná osa, vede to k běhové chybě.
2	1			Aktivní poloha Override pro příslušný potenciometr v procentech: <ul style="list-style-type: none"> • Posuv • Vřeteno (SD(2,3)=0, pokud není aplikováno žádné vřeteno) • 2. Vřeteno (SD(2,4)=0, pokud není aplikováno žádné 2. vřeteno)
2	3			
2	4			
202	1...8			Aktuální hodnota potenciometru (stará nadále platná funkce SDR: SDR(2,3), SDR(2,4))
68	1	1...8	Trans, ATrans	<ul style="list-style-type: none"> • Součet naposledy naprogramovaných posunutí programových souřadnic (Trans+ATrans) pro uvedenou osu (index2). • Součet aktivních posunutí programových souřadnic (Trans + ATrans) pro uvedenou osu (Index2). • Hodnota naposledy naprogramovaného posunutí programových souřadnic (Trans) pro uvedenou osu (index1). • Hodnota aktivního posunutí programových souřadnic (Trans) pro uvedenou osu (Index2). • Hodnota naposledy naprogramovaného aditivního posunutí programových souřadnic (ATrans) pro uvedenou osu (index1). • Hodnota aktivního aditivního posunutí programových souřadnic (Arans) pro uvedenou osu (index2).
68	2	1...8		
168	1	1...8		
168	2	1...8		
268	1	1...8		
268	2	1...8		
77	Název souřadnice stroje	0 1 2 3		V asynchronním podprogramu: <ul style="list-style-type: none"> • Udává aktuální bod najetí podle ASPRTP/REPOSTP souřadnice stroje ve větě přerušení. • Udává počáteční bod souřadnice stroje ve větě přerušení. • Udává koncový bod souřadnice stroje ve větě přerušení. • Udává bod přerušení souřadnice stroje ve větě přerušení. Udaná poloha umožňuje v asynchronním podprogramu pomocí G76 zpětné polohování souřadnic do blízkosti požadovaného bodu najetí.
131	3		Tangenciální vedení nástroje	<ul style="list-style-type: none"> • Udává úhel nastavení ve stupních (index2). • Udává mezilehlý úhel ve stupních (index2).
131	4			

Systémová data

Skupina	Index1	Index2	Týká se funkce	Popis
328			PrecProg	<ul style="list-style-type: none"> • Naposledy naprogramovaná mez přesnosti PrecProg
328	1			<ul style="list-style-type: none"> • Naposledy naprogramovaná mez přesnosti PrecProg
328	2			<ul style="list-style-type: none"> • Naposledy naprogramovaná rohová vzdálenost PrecProg
			Spojení os	m: Číslo kanálové osy hlavního zařízení v aktuálním kanálu s: Číslo kanálové osy podřízeného zařízení v aktuálním kanálu
581	0	1...8(m)		<ul style="list-style-type: none"> • Číslo kanálové osy m, když je osa m řídicí 0: když neexistuje žádná řídicí osa
581	1...8(s)	0		<ul style="list-style-type: none"> • Číslo řídicí osy, ke které je osa s následnou osou 0: když s není následná osa
581	1...8(s)	1		<ul style="list-style-type: none"> • Naprogramované posunutí následné osy v programovacích jednotkách. 0: když s není následná osa
581	1...8(s)	2		<ul style="list-style-type: none"> • Naprogramovaný faktor spojení 0: když s není následná osa
581	1...8(s)	3		<ul style="list-style-type: none"> • Naprogramované posunutí řídicí osy v programovacích jednotkách 0: když s není následná osa

5.8.2 Systémová data strukturovaných typů

Strukturovaná systémová data (krátce SD) tvoří třídu dat v celém systému s následujícími vlastnostmi:

- Jednotlivá strukturovaná data jsou uložena podle potřeby v dočasné nebo trvalé paměti (permanентní SD).
- Počet, velikost a struktura jsou v rámci existující paměti volně definovatelné.

Podrobné informace o používání a definicích systémových dat naleznete ve funkční příručce.

V CPL se získává přístup k SD s použitím klíčového slova. Na rozdíl od adresování XPath (standardizovaný přístup ke strukturovaným datům) se místo oddělovacího znaku "/" používá bodový operátor ".". Pole jsou adresována s použitím [], např.

Xpath: /MyArrVar[2,3,4]/dílní složka

CPL: SD.MyArrVar[2,3,4].dílní složka

Příklad:

```
10 SD.MyChanVar=DBSEA("/dbt1/Rec",-1,-1,"Key1=1",I%)
```

U kanálových SD je možné vynechávat index kanálu, např.:

SD.MyArrVar[3,4] nebo

SD.MyChanVar bez [] u jednorozměrných kanálových SD.

Systémová data

V lomených závorkách mohou být naprogramovány libovolné výrazy CPL, které dávají hodnotu Integer.

Data SD se stejnými typy struktury mohou být navzájem přiřazována. U základních typů platí následující pravidla přiřazení:

Název	Délka v bajtech	Popis	Typ C	Typ IndraLogic	CPL
Řetězec	2 * max. délka + 1	String (formát UTF-8)	char []	STRING()	STRING
IsoLatin1String	max. délka + 1	String (formát Latin1)	char []	STRING()	STRING
Byte_t	1	Signed 8-bitový integer	char	SINT	INT
Short_t	2	Signed 16-bitový integer	short	INT	INT
Int_t	4	Signed 32-bitový integer	int	DINT	INT
UnsignedByte_t	1	Unsigned 8-bitový integer	unsigned char	USINT	INT
UnsignedShort_t	2	Unsigned 16-bitový integer	unsigned short	UINT	INT
UnsignedInt_t	4	Unsigned 32-bitový integer	unsigned int	UDINT	INT
Float_t	4	32-bitový real	float	REAL	REAL
Double_t	8	64-bitový real	double	LREAL	DOUBLE
Boolean_t	1	true, false, 1, 0	char	BOOL	BOOLEAN

Permanentní systémová data se inicializují jednou, dočasná systémová data při každém spuštění znovu. Hodnoty mohou být stanoveny v inicializačním souboru. Pokud zde není nic stanoveno, jsou obsazeny 0.

Strukturované proměnné SV mohou být nahrazeny daty SD, ale z důvodů kompatibility zůstávají zachovány.

Systémová data

5.9 Variabilní adresa osy

AXP

Tato funkce umožňuje zapisovat dílčí a měřicí programy nezávisle na rovinách.

AXP (<číslo osy>,<informace o dráze>[,<typ osy>])

Funkce se používá v NC-větě. Musí být uvedena v lomených závorkách "[]" a programuje se na místě hodnot adresy.

<Číslo osy>	Index souřadnice obrobku nebo systémové osy
<Informace o dráze>	Proměnná nebo hodnota s informací o dráze
<Typ osy>	volitelný: Určuje, jak bude interpretován index naprogramovaný pod <číslem osy>: "0": Index systémové osy. "1": Index souřadnice obrobku kanálu, ve kterém je program aktuálně zpracováván. Bez explicitního naprogramování je nastaven <typ osy> = 1.

Příklad:

Podprogram:

10 A%=P1% : B%=P2%	Č. osy z P1% a P2% předáno do A% a B%
20 C=P3 : D=P4 : RA=P5	Předány požadované hodnoty pro G2
30 E=0	Konstanta pro pól u G20
N40 G20 [AXP(A%,E)][AXP(B%,E)]	Přepnutí roviny s G20; pól na 0,0
N50 G2 [AXP(A%,C)][AXP(B%,D)] R[RA] F1000	Programování poloměru pomocí G2

Definice roviny pomocí A% a B%. Potom přepnutí roviny pomocí G20. Nakonec pojíždějí osy s F1000 po kruhovém oblouku, který je určen proměnnými C a D (koncový bod) a RA (poloměr).

Systémová data

5.10 Rozhraní SPS

BITIF

Tato funkce umožňuje přístup k digitálnímu rozhraní mezi NC a SPS.

- Je možné číst všechny vstupy a výstupy.
- Navíc je možné zapisovat do zákaznických výstupů CPL Ch_Cpl01...16 v kanálovém rozhraní.

BITIF (<bitový signál>[,<index>[,<jednotka IF>]])

kde

<Bitový signál> Bitový offset nebo symbolická adresa.

Bitový offset: 0...31 pro globální rozhraní
 0...111 pro kanálové rozhraní
 0...95 pro osové rozhraní
 0...95 pro vřetenové rozhraní
 0...7 pro rozhraní High Speed

Symbolické adresy naleznete v příručce k rozhraní SPS.

<Index>

Index osy, vřetena nebo kanálu.

Není-li žádný index naprogramován, platí pro dotazování:

- aktivní kanál pro kanálové rozhraní
- osa 1 pro osové rozhraní
- vřeteno 1 pro vřetenové rozhraní

<Jednotka IF> 0 nebo QCH: Výstupní signály SPS, vztažené ke kanálu

1 nebo QAX: Výstupní signály SPS, vztažené k ose

2 nebo QSP: Výstupní signály SPS, vztažené k vřetenu

3 nebo ICH: Vstupní signály SPS, vztažené ke kanálu

(vč. zákaznických výstupů CPL z NC)

4 nebo IAX: Vstupní signály SPS, vztažené k ose

5 nebo ISP: Vstupní signály SPS, vztažené k vřetenu

6 nebo QGEN: Výstupní signály SPS, globální

7 nebo IGEN: Vstupní signály SPS, globální

8 nebo QHS: Výstupní signály SPS, high speed

9 nebo IHS: Vstupní signály SPS, high speed

Systémová data

Příklady:

I?=BITIF("QGEN_RESET")	Čtení globálního výstupního signálu "Základní poloha systému".
I?=BITIF("ICH_RESET",1)	Čtení kanálového vstupního signálu "Kanál v základní poloze" kanálu 1
I?=BITIF(0,1,"IAX")	Čtení osového vstupního signálu "Referenční bod známý" pro osu 1
I?=BITIF(88,2,5)	Čtení vřetenového vstupního signálu "Chyba stavové třídy 1" pro vřeteno 2
BITIF("ICH_CPL05")=1	Popsání zákaznického výstupu CPL 5 hodnotou 1
BITIF(50,1,3)=TRUE	
BITIF("ICH_CPL02")=FALSE	

PLC

Tato funkce umožňuje přístup k operandům SPS.

PLC (<Typ>,<Prázdný parametr>,<Adresa>,<Velikost>)

kde

<Typ>	1: Vstup (E)
	2: Výstup (A)
	3: Označovač (M)

Pro všechny typy je dovolen čtecí přístup.
Zápisový přístup do označovače je dovolený, když je parametr MP 2060 00200 nastaven na hodnotu 5 (= IndraLogic).

<Prázdný parametr> (neobsazeno)

<Adresa> Relevantní bajtová adresa od začátku oblasti.
Řízení kontroluje parametr podle aktivní jednotky SPS.

<Velikost> Velikost datového typu:

1:	Byte
2:	Slovo
3:	Dvojité slovo

Příklad:

```

:
30 REM Číst 2 bajty od vstupu 10
40 I% = PLC(1,,10,2)
:

```

Systémová data

5.11 Zaznamenávání času

CLOCK

Čte interní systémový čas řízení v milisekundách.

Příklad:

```
:  
20 WAIT  
30 STARTZEIT% = CLOCK  
N4 G1 X50 Y70  
40 WAIT  
50 ENDZEIT%=Clock : DIFF%=ENDZEIT%-STARTZEIT%  
:
```

Před a po provedení věty N4 se aktuální stav počítadla času přiřazuje proměnným "STARTZEIT%", resp. "ENDZEIT%". Z rozdílu obsahů obou proměnných je pak možné určit dobu provádění věty N4 v milisekundách.

Pamatujte si, že je zde nezbytně nutný příkaz WAIT!

DATE

Udává aktuální hodnotu pro datum.

Příklad:

```
:  
30 A$ = DATE  
:
```

Proměnné STRING A\$ je přiřazeno datum ve formě DD.MM


TIME

Udává aktuální hodnotu pro čas.

Příklad:

```
:  
40 B$ = TIME  
:
```

Proměnné STRING B\$ je přiřazen čas ve formě HH.MM.SS

 Pokud je funkce DATE/TIME použita v programu vícekrát, musí být dimenzovány odpovídající výsledné proměnné. Jinak bude napsledy načtená hodnota funkce DATE/TIME přiřazena všem nedimenzovaným proměnným, které obsahují výsledek přiřazení DATE/TIME.

Systémová data

5.12 Chyby a kategorie chyb

GETERR

Tato funkce vyvolává aktuální chybu v CPL-programu. Obsaženy jsou prvky č. aktuální chyby, č. kanálu chyby a příslušná kategorie chyby. Každá chyba, která se vyskytne, je se svými prvky uložena v poli. Max. počet chyb, které se mají vejít do pole, je omezen dimenzováním (DIM) parametru <č. chyby>.


Funkce GETERR vrací následující hodnoty:

- -1: Nelze provést funkci.
- ≥ 0 : Počet chyb vyskytujících se v <kanálu>.

GETERR(<kanál>,[<kategorie>],[<č. chyby>],[<počet>])

<Kaná<	Č. dotazovaného kanálu -1 : Všechny kanály > 0: Č. kanálu
<Kategorie>	0: Všechna varování a chyby (standardní) 1: Lehké systémové chyby 2: Chyby regulace nebo pohonu 3: Chyby interpolátoru 4: Hardwarové chyby 5: Chyby ICL 6: Chyby dílčího programu 7: Běhová varování 8: Hlášení MZA: Chyba 9: Hlášení MZA: Varování 10: Hlášení MZA: Upozornění
<Č. chyby>	Výsledná proměnná: Dvozměrné pole Integer nejméně se 3 prvky ve druhém rozměru (DIM <č. chyby>% (x,3)), standardní hodnota: 0. Funkce dává v časově vzestupném pořadí čísla vyskytujících se chyb z <kanálu>. Význam 3 prvků 2. rozměru: <Č. chyby>(x,1): Č. chyby <Č. chyby>(x,2): Kanál chyby (-1 = přesahující kanály) <Č. chyby>(x,3): Kategorie chyby (pokud je pomocí příkazu DIM deklarována). 0 = neznámá kategorie jiné hodnoty jako <kategorie>

Příklad: DIM ERRNO% (100,3).

 **Smí být zadán pouze název proměnné bez rozměru, resp. indexu!**

Systémová data

<Počet> Proměnná Integer (standardní hodnota: 1)
Určuje počet chyb k přečtení.
1: Standardní hodnota
> 0: Hodnoty parametrů se nekontrolují s ohledem na platnost,
tzn. když je počet dat ke čtení větší než rozměr pole,
není generována žádná chyba dílčího programu.

Příklad:

<Počet> =120, ale DIM ERRNO% (100,3).
20 chyb v tomto případě není přečteno.

Příklad:

Zeptat se na chyby v posledním dílčím programu v kanálu 2

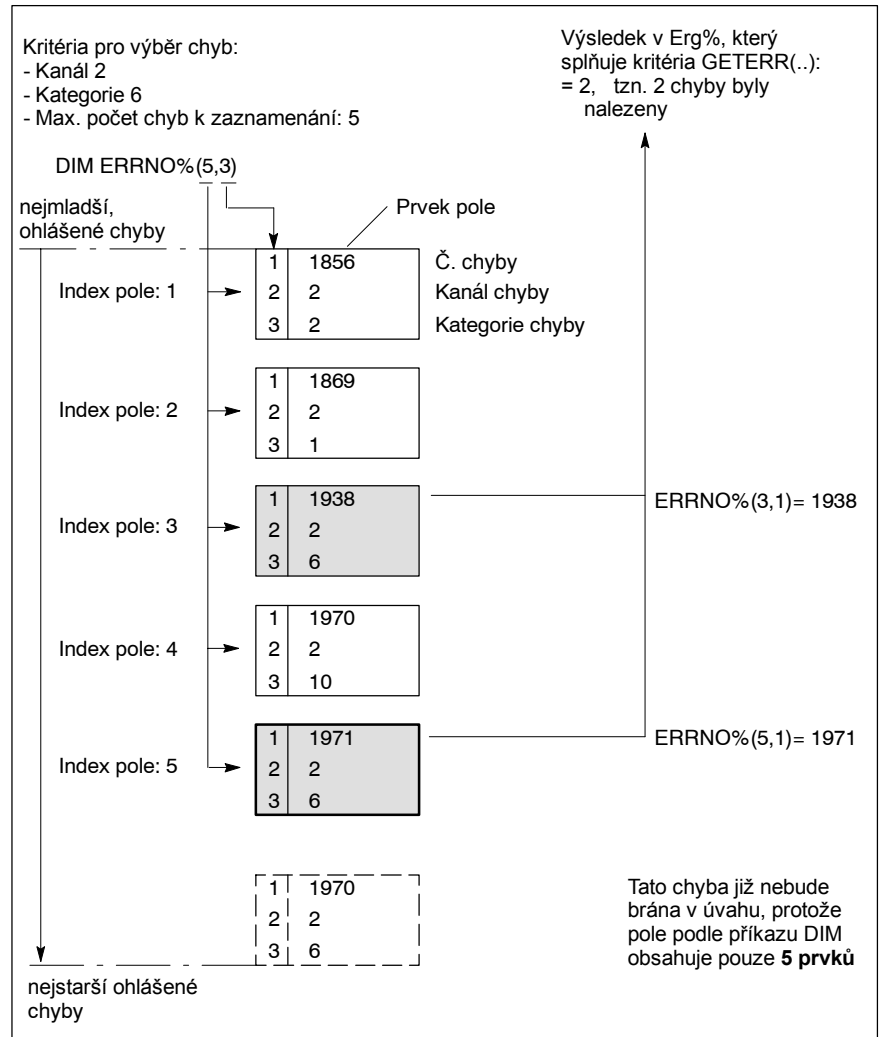
```
10 DIM ERRN=(5,3) : REM Integer-Array s 5 prvky
20 REM Zeptat se na chyby v posledním dílčím programu v
kanálu 2
30 CHAN%=2:CATEGORY%=6
40 ERG%=GETERR(CHAN%,CATEGORY%,ERRNO%,1)
:
```

Příklad:

Výsledek-Vyhodnocení pro 5 prvků pole.

```
10 DIM ERRNO%(5,3):REM Integer-Array s 5 prvky
20 CHAN%=2 : CATEGORY%=0
25 REM Zeptat se na chyby v posledním dílčím programu v kanálu 2
30 ERG%= GETERR(CHAN%, CATEGORY%, ERRNO%, 5)
40 FOR I%= 1 TO ERG%
50     IF ERRNO%(I%,3)=6 THEN
60         PRN#(0, "Chyby dílčího programu: ",ERRNO%(I%,1))
70     ENDIF
80 NEXT I%
```

Systémová data



Jako výsledek jsou v okně MSG vydány chyby dílčího programu 1938 a 1971. Proměnná Erg% obsahuje hodnotu 2.

 **S použitím funkce GETERR můžete mimo jiné zaznamenat chronologický výskyt jedné nebo několika chyb, a na základě toho zjistit vlastní příčinu chyby.**

Systémová data

5.13 Připojení NCS

Pomocí funkcí pro připojení NCS je možné s použitím CPL získat přístup k procesovým a datovým službám interního rozhraní NCS.

5.13.1 Možné vrácené chybové hodnoty funkcí

Všechna volání funkcí vracejí hodnotu pro kontrolu a zpracování chyb. Tato hodnota může být přiřazena proměnné Integer nebo Real.

Příklad:

```
ERR_VAR% = MCOPS (...)
```

```
ERR_VAR% = MCODES (...)
```

**POZOR**

Jsou možné nesprávné reakce programu!

Když vyvolané funkce vracejí chybový kód, nebyly provedeny operace důležité pro další průběh programu nebo byly provedeny neúplně.

Proto důrazně doporučujeme po vyvolání funkce zkontrolovat s použitím programové techniky (např. pomocí CASE), jestli mohla být funkce bez chyb provedena.


Další chování programu závisí na typu a závažnosti chyby, která se vyskytla.

Jsou definovány následující vrácené hodnoty:

- 0: Nedošlo k žádné chybě
- 1: Zadaný kanál neexistuje
- 2: Nelze provést funkci, protože zadaný kanál je právě obsazený (okamžitý stav nedovoluje provést operaci)
- 3: V zadaném kanálu ještě neskončilo spuštěné základní nastavení.
- 4: Zadaný název programu je příliš dlouhý (v současné době se nepoužívá)
- 5: Funkce vyžaduje najetí do referenčních bodů
- 6: Zadaný program neexistuje, nebo není proveditelný
- 7: Při zadávání NC-vět do vyrovnávací paměti byl přerušen zápis. Druhá instance se zároveň pokusila zapisovat do vyrovnávací paměti.
- 8: Nelze provést funkci v aktuálním provozním režimu
- 9: Nelze spustit kanál, protože jeho stav není PŘIPRAVEN
- 10: Nelze provést funkci, protože není zvolen žádný program
- 11: Nelze zvolit zadaný program, protože to nedovoluje stav kanálu (např. stav přípravy vět a interpolátoru je BĚŽÍ)
- 12: V současné době se nepoužívá
- 13: Nelze změnit provozní režim, protože to nedovoluje stav kanálu
- 14: Nebyl nalezen cíl příkazu "hledání věty"

Systémová data

- 15: "Hledání věty" není možné, protože stav kanálu je sice PŘIPRAVEN, ale začalo již zpracování hlavního programu (program je např. na M0)
- 16: Při zadání externích posunutí nulových bodů je počet os příliš vysoký
- 17: Při zadání externích posunutí nulových bodů je počet skupin osových NPV příliš vysoký
- 18: Zadaná syntaxe je neznámá
- 19: Nepřípustný index při zadání externí opravy nástroje
- 20: Počet oprav při zadání externí opravy nástroje je příliš vysoký (příp. ve spojení s indexem opravy)
- 21: Nepřípustný zadaný formát při zadání externí opravy nástroje
- 22: Nepřípustná poloha břitů při zadání externí opravy nástroje
- 23: Nepřípustná skupina oprav
- 24: Volaná osa neexistuje
- 25: Při zadávání NC-vět s automatickým spuštěním byla zjištěna běhová chyba, např. chyba syntaxe.
- 26: Při zadávání NC-vět do vyrovnávací paměti přetekla paměť.
- 27: Zadání pro filtr souřadnic není správné.
- 100: Číslo Magic telegramu je nesprávné
- 101: Komunikace NCS je rušená
- 102: Zadaná funkce není v této softwarové verzi k dispozici
- 103: Došlo k interní chybě (v současné době se nepoužívá)

 **Jsou-li ve vyvolaném programu obsaženy také NC-věty, předbíhá příprava vět zpravidla obrábění na stroji. Pokud pak v okamžiku přípravy věty požaduje funkce MCOPS nebo MCODS procesovou službu nebo se ptá na stav stroje, není na stroji ještě splněn případný nezbytný předpoklad.**

Tento problém ovšem nastává pouze u funkcí, které používají přesně ten kanál, v němž jsou samy prováděny.


V takovém případě použijte v řádku před vyvoláním funkce příkaz WAIT. Tím se zastaví příprava vět na tak dlouho, dokud nebudou skutečně zpracovány všechny věty před funkcí WAIT.

Systémová data

5.13.2 Dostupné funkce

MCODS

Vyvolává datové služby Motion Control v NCS pomocí CPL. Tak je možné načítat data a stavy z CNC.

-  **Všechny načtené hodnoty se vztahují k okamžiku, ve kterém byla věta CPL zpracována přípravou vět.**
Jsou-li ve vyvolaném programu obsaženy také NC-věty, předbíhá příprava vět zpravidla obrábění na stroji. Jestliže chcete ovlivnit průběh programů funkcemi, které zjišťují aktuální data nebo stavy stroje, musíte eliminovat "časový posun" mezi přípravou vět a aktuálním stavem stroje. To ovšem platí jen při použití funkcí, které používají přesně ten kanál, v němž jsou samy prováděny. V takovém případě použijte v řádku před vyvoláním funkce příkaz WAIT. Tím se zastaví příprava vět na tak dlouho, dokud nebudou skutečně zpracovány všechny věty před funkcí WAIT.

Funkce dávají vrácenou hodnotu (viz kapitola 5.13.1).


Všeobecná syntaxe:

MCODS (<typ>,<kanál>,<verze>,<vyrovnávací paměť>,<velikost>,<číslo osy>,<identifikační číslo>[,<P1>])

- | | |
|---------------------|--|
| <Typ> | Výraz Integer. Udává prováděnou funkci. V následující tabulce jsou uvedeny všechny dostupné funkce. |
| <Kanál> | Výraz Integer. Udává kanál, na který má funkce působit. |
| <Verze> | Inicializovaná proměnná Integer nebo Real (ne konstanta!).
Pokud je obsah proměnné při vyvolání funkce = 0, zapíše funkce daná <typem> požadovaná data okamžitě do <vyrovnávací paměti>.
Kromě toho vrací funkce ve <verzi> označení verze dodaných dat. Je-li toto označení verze při příštím vyvolání funkce ještě obsaženo v proměnné, nezapíše funkce požadovaná data okamžitě, ale až po další změně dat ve <vyrovnávací paměti>.
Tímto způsobem je např. možné procházet programovou smyčku tak dlouho, až dosáhne kanál určitého stavu. Přitom ovšem musíte zahrnout do smyčky podmínku timeout (např. počítadlo nebo uplynulý časový interval), aby nevznikla nekonečná smyčka! |
| <Vyrovnávací paměť> | Ve <vyrovnávací paměti> vrací funkce požadované hodnoty dat. Podle typu dat musí být <vyrovnávací paměť>: |

Systémová data

- jednoduchá proměnná typu Integer, Real, Double
- pole proměnných typu Integer, Real, Double
- proměnná String (1-rozměrné pole Character).


 **U polí proměnných nebo proměnných String se smí zadávat pouze název proměnné bez rozměru, resp. indexu!**

<Velikost> Výraz Integer. Určuje velikost pole <vyrovnávací paměti>.

Pokud <vyrovnávací paměť> není pole proměnných, ale jednoduchá proměnná typu Integer, Real nebo Double, zadejte pro <velikost> hodnotu 1.

<Číslo osy> Výraz Integer. Udává číslo systémové osy.

<Identifikační číslo> Výraz Integer. Udává hodnotu <identifikačního čísla> z cyklického osového telegramu pro všechny osy.

 **Velikost použitého pole proměnných musí být předem definována příkazem DIM a nesmí být v parametru <velikost> překročena!**

<P1>: Volitelný parametr, který je závislý na <typu>.

Přehled funkce MCODS(...)


Polohy	
Požadovaná poloha osy	viz MCODS(1..)
Požadovaná poloha osy	viz MCODS(2..)
Skutečná poloha osy	viz MCODS(35..)
Skutečné hodnoty os (soustava souřadnic stroje)	viz MCODS(38..)
Doběh	viz MCODS(4..)
Programová hodnota osy (programová soustava souřadnic)	viz MCODS(37..)
Naprogramované koncové polohy vč. posunutí	viz MCODS(16..)
Naprogramované koncové polohy bez posunutí	viz MCODS(23..)
Rychlost a počet otáček	
Posuv dráhy	viz MCODS(3..)
naprogramovaný posuv dráhy	viz MCODS(24..)
Rychlosti krokování	viz MCODS(27..)
Požadovaný počet otáček vřetena, řezná rychlost	viz MCODS(5..)
Skutečný počet otáček vřetena	viz MCODS(36..)
Maximální počet otáček vřetena	viz MCODS(19..)
Minimální počet otáček vřetena	viz MCODS(20..)

Systémová data

naprogramované počty otáček vřetena	viz MCODES(25..)
Stavy	
Stav "InPos"	viz MCODES(6..)
Stav "Zkušební provoz"	viz MCODES(29..)
Stav "Najetí do referenčního bodu"	viz MCODES(26..)
Stav "Doba prodlevy aktivní"	viz MCODES(39..)
Stav "Pomocná funkce s povinným potvrzením aktivní"	viz MCODES(40..)
Stav "Uvolnění zadávání"	viz MCODES(41..)
Stav "Příkaz pojezdu"	viz MCODES(47..)
Stav "Blokování posuvu"	viz MCODES(49..)
Stav SAV a IPO	viz MCODES(32..)
Potenciometr	
Hodnota potenciometru posuvu	viz MCODES(7..)
Hodnoty potenciometru vřetena	viz MCODES(8..)
Hodnota potenciometrů os	viz MCODES(50..)
Opravy	
Číslo aktivní opravy délky	viz MCODES(9..)
Aktivní oprava délky	viz MCODES(10..)
Číslo aktivní opravy poloměru nástroje	viz MCODES(11..)
Aktivní oprava poloměru nástroje	viz MCODES(12..)
Název aktivní tabulky oprav nástroje	viz MCODES(13..)
Název aktivní tabulky osových posunutí nulových bodů	viz MCODES(14..)
Aktivní hodnoty osových posunutí nulových bodů	viz MCODES(15..)
Aktivní hodnoty externí opravy nástroje	viz MCODES(51..)
Aktivní hodnoty externích osových posunutí nulových bodů	viz MCODES(52..)
Aktivní univerzální oprava nástroje	viz MCODES(54..)
Provozní režimy	
Provozní režim kanálu	viz MCODES(31..)
Provozní režim osy	viz MCODES(48..)
Struktura systému	
Počet posuvových os, pomocných os, vřeten; typy pohybu, typy pohonu	viz MCODES(34..)
Počet kanálů	viz MCODES(44..)
Počet os	viz MCODES(45..)
Názvy os	viz MCODES(33..)
Aktivní názvy kanálových os	viz MCODES(59..)

Systémová data

Názvy kanálových os (standardní nastavení)	viz MCODES(60..)
Přiřazení osa - kanál	viz MCODES(43..)
Standardní přiřazení osa - kanál	viz MCODES(58..)
Identifikační číslo z cyklického osového telegramu	viz MCODES(62..)
Měrné jednotky	
Měrné jednotky os (standardní nastavení)	viz MCODES(61..)
Měrná jednotka os	viz MCODES(53..)
Typ programování (Inch/metrické)	viz MCODES(18..)
Programování průměru	viz MCODES(91..)
Pomocné funkce	
Čísla skupin pomocných funkcí	viz MCODES(94..)
Aktivní syntaxe skupin pomocných funkcí	viz MCODES(65..)
Vřetena	
Pohybové funkce vřeten	viz MCODES(63..)
Převodové stupně vřeten	viz MCODES(64..)
Automatická, resp. ruční volba převodových stupňů	viz MCODES(66..)
Informace, zda je přepínání převodů aktivní	viz MCODES(67..)
Pohon	
Verze výrobce	viz MCODES(55..)
Typ regulátoru	viz MCODES(56..)
Typ motoru	viz MCODES(57..)
Různé	
Hlášení v dílčím programu	viz MCODES(28..)
Cesta a název hlavního programu	viz MCODES(30..)
Strategie opětovného najetí a zaznamenávání krokových pohybů	viz MCODES(46..)
Zákaznická data	viz MCODES(42..)
Volitelné zastavení (aktivované)	viz MCODES(68..)
Vynechat větu (aktivovat)	viz MCODES(69..)
Automatická opakovaná volba programu aktivní	viz MCODES(70..)

 **V následující tabulce jsou u syntaxe uvedeny jako parametry zčásti konstanty Integer. Místo těchto konstant můžete naprogramovat také proměnné Integer, které ovšem musí být v okamžiku vyvolání funkce obsazeny zadanou hodnotou.**

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Požadovaná poloha osy	Double, Array	MCODS(1,-1,<verze>,<vyrovňovací pamě>,<velikost>)
Z		Udává ve <vyrovňovací paměti> ve vzestupném pořadí, nezávislém na kanálech, požadované polohy všech posuvových a pomocných os v systému : u lineárních os v mm u kruhových os ve stupních V hodnotách je započítáno "nastavení skutečné hodnoty" (např. G92).
Požadovaná poloha osy	Integer, Array	MCODS(2,-1,<verze>,<vyrovňovací pamě>,<velikost>)
Z		Udává ve <vyrovňovací paměti> ve vzestupném pořadí, nezávislém na kanálech, požadované polohy všech posuvových a pomocných os v systému : u lineárních os v 0,0001 mm u kruhových os v 0,0001 stupně V hodnotách je započítáno "nastavení skutečné hodnoty" (např. G92).
Posuv dráhy	Real, Array	MCODS(3,<kanál>,<verze>,<vyrovňovací pamě>,3)
Z		Udává ve <vyrovňovací paměti> ve vzestupném pořadí 3 hodnoty s aktuálními posuvy drah <kanálu> (vč. potenciometru posuvu) v jednotkách mm/min: 1. Požadovaná rychlost, která je externě zadána interpolátoru. 2. Skutečná rychlost interpolátoru (= okamžitá rychlost dráhy). 3. Interní požadovaná rychlost interpolátoru. Dá se změnit oproti externě zadané pomocí aplikace (např. funkce Feed-Adapt). Při programování posuvu v mm/U (G95) dává posuv dráhy v mm/min.
Doběh	Real, Array	MCODS(4,-1,<verze>,<vyrovňovací pamě>,<velikost>)
Z		Udává ve <vyrovňovací paměti> ve vzestupném pořadí doběh všech systémových os: u lineárních os v mm u kruhových os ve stupních Není-li přenos doběhu podporován pohony (pomocí parametrů SERCOS), vrací se hodnota 0.0.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Požadovaný počet otáček vřetena, řezná rychlost-	Real, Array	MCODS(5,-1,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> ve vzestupném pořadí (S, resp. S1, S2, S3 atd.) požadované počty otáček vřeten nebo řezné rychlosti všech vřeten existujících v systému . Při aktivní funkci G196 jsou udávány řezné rychlosti v m/min, jinak aktuální požadované počty otáček vřeten v ot./min. Započítány jsou potenciometr, omezení počtu otáček (SMin, SMax) a omezení převodovým stupněm. Pokud vřeteno neexistuje, je ve <vyrovnávací paměti> vrácena na příslušném místě hodnota 0.0.
Stav "InPos"	Integer, Array	MCODS(6,-1,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> ve vzestupném pořadí, nezávislém na kanálech, hodnotu 0 nebo 1 jako signál InPos pro každou posuvovou a pomocnou osu : Osa je v poloze: 1 Osa není v poloze: 0 Osa je v poloze, když se nachází v parametrizovaném okně InPos (MP 1015 00100) a neexistuje žádný příkaz pojezdu (viz též MCODES(47...)).
Hodnota potenciometru posuvu	Real	MCODS(7,<kanál>,<verze>,<vyrovnávací pamě>,1)
Z		Udává ve <vyrovnávací paměti> aktuální hodnotu potenciometru posuvu <kanálu> v 1/100 procenta.
Hodnoty potenciometru vřetena	Real, Array	MCODS(8,-1,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> ve vzestupném pořadí pro všechna vřetena existující v systému (S, resp. S1, S2, S3 atd.) aktuální hodnoty potenciometru posuvu v 1/100 procenta.
Číslo aktivní opravy délky	Integer	MCODS(9,<kanál>,<verze>,<vyrovnávací pamě>,1)
S		Udává ve <vyrovnávací paměti> číslo opravy délky aktivní v <kanálu>. Není-li aktivní žádná oprava délky, vrací se hodnota -1.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Aktivní oprava délky	Real	MCODS(10,<kanál>,<verze>,<vyrovňovací pamě>,1)
S		Udává ve <vyrovňovací paměti> opravu délky aktivní v <kanálu> v mm. Není-li aktivní žádná oprava délky, vrací se hodnota 0.0.
Číslo aktivní opravy poloměru nástroje	Integer	MCODS(11,<kanál>,<verze>,<vyrovňovací pamě>,1)
S		Udává ve <vyrovňovací paměti> číslo opravy poloměru nástroje aktivní v <kanálu>. Není-li aktivní žádná oprava poloměru, vrací se hodnota -1.
Aktivní oprava poloměru nástroje	Real	MCODS(12,<kanál>,<verze>,<vyrovňovací pamě>,1)
S		Udává ve <vyrovňovací paměti> opravu poloměru nástroje aktivní v <kanálu> v mm. Není-li aktivní žádná oprava poloměru nástroje, vrací se hodnota 0.0.
Název aktivní tabulky oprav nástroje	Character, Array	MCODS(13,<kanál>,<verze>,<vyrovňovací pamě>,<velikost>)
S		Udává ve <vyrovňovací paměti> název tabulky oprav nástroje aktivní v <kanálu>. Není-li žádná aktivní, jsou vráceny 3 mezery (prázdné znaky) jako String.
Název aktivní tabulky osových posunutí nulových bodů	Character, Array	MCODS(14,<kanál>,<verze>,<vyrovňovací pamě>,<velikost>)
S		Udává ve <vyrovňovací paměti> název tabulky osových posunutí nulových bodů aktivní v <kanálu>. Není-li žádná aktivní, jsou vráceny 3 mezery (prázdné znaky) jako String.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Aktivní hodnoty osových posunutí nulových bodů	Real, Array	MCODS(15,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
S		<p>Udává ve <vyrovnávací paměti> hodnoty osových posunutí nulových bodů všech řad, aktivních v <kanálu>, pro všechny posuvové osy v mm. Není-li aktivní žádné posunutí, vrací se vždy hodnota 0.0.</p> <p>Platí následující pořadí:</p> <ul style="list-style-type: none"> ● Posunutí 1. osy v řadě 1 ● Posunutí 2. osy v řadě 1 ● Posunutí n-té osy v řadě 1 ● Posunutí 1. osy v řadě 2 : ● Posunutí n-té osy v řadě 2 : ● Posunutí n-té osy v řadě 3
Naprogramované koncové polohy včetně posunutí	Real, Array	MCODS(16,0,<verze>,<vyrovnávací pamě>,<velikost>)
S		<p>Udává ve <vyrovnávací paměti> ve vzestupném pořadí, nezávislém na pořadí, koncové polohy aktivních vět všech posuvových a pomocných os vztahených k souřadnicím obrobku:</p> <ul style="list-style-type: none"> ● u lineárních os v mm ● u kruhových os ve stupních <p>Jsou započítány všechny hodnoty posunutí. "Nastavení skutečné hodnoty" (např. G92) není v hodnotách započítáno.</p>
Typ programování (Inch/metrické)	Integer	MCODS(18,<kanál>,<verze>,<vyrovnávací pamě>,1)
S		<p>Udává ve <vyrovnávací paměti> typ programování os existujících v <kanálu>:</p> <p>0: Inch 1: metrické 2: stupně 3: neexistují žádné osy</p>

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Maximální počet otáček vřetena	Real, Array	MCODS(19,0,<verze>,<vyrovnávací pamě>,<velikost>)
S		Udává ve <vyrovnávací paměti> maximální přípustné počty otáček všech vřeten v systému v ot./min. Pořadí: S nebo S1, S2 atd. Omezení počtu otáček jsou započítána.
Minimální počet otáček vřetena	Real, Array	MCODS(20,0,<verze>,<vyrovnávací pamě>,<velikost>)
S		Udává ve <vyrovnávací paměti> minimální přípustné počty otáček všech vřeten v systému v ot./min. Pořadí: S nebo S1, S2 atd. Omezení počtu otáček jsou započítána.
Naprogramované koncové polohy bez posunutí	Real, Array	MCODS(23,0,<verze>,<vyrovnávací pamě>,<velikost>)
S		jako MCODES(16...), ovšem bez posunutí.
naprogramovaný posuv dráhy	Real	MCODS(24,<kanál>,<verze>,<vyrovnávací pamě>,1)
S		Udává ve <vyrovnávací paměti> naprogramovaný posuv dráhy <kanálu> v jednotkách mm/min.
naprogramované počty otáček vřetena	Real, Array	MCODS(25,0,<verze>,<vyrovnávací pamě>,<velikost>)
S		Udává ve <vyrovnávací paměti> naprogramované počty otáček všech vřetenových os v systému v ot./min. Pořadí: S nebo S1, S2 atd.
Stav "Najetí do referenčního bodu"	Integer, Array	MCODS(26,-1,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> ve vzestupném pořadí, nezávislém na kanálech, hodnotu 0 nebo 1 jako signál "Najetí do referenčního bodu" pro každou posuvovou a pomocnou osu : Najeto do referenčního bodu: 1 Nenajeto do referenčního bodu: 0
Rychlosti krokování	Real, Array	MCODS(27,-1,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> ve vzestupném pořadí, nezávislém na kanálech, aktuální rychlosti krokování všech posuvových a pomocných os v systému : u lineárních os v mm/min u kruhových os v ot./min

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Hlášení v dílčím programu	Character, Array	MCODS(28,<kanál>,<verze>,<vyrovnávací pamě>,80)
E		Udává ve <vyrovnávací paměti> pokyny naprogramované příkazem MSG v <kanálu>.
Stav "Zkušební provoz"	Integer	MCODS(29,0,<verze>,<vyrovnávací pamě>,1)
E		Udává ve <vyrovnávací paměti> hodnotu 1, když je aktivován zkušební provoz. Jinak 0.
Cesta a název hlavního programu	Character, Array	MCODS(30,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> cestu vč. názvu hlavního programu zvoleného v <kanálu> jako String. Hodnota existující ve <verzi> je při vyvolání funkce ignorována. U souborů, které jsou uloženy v NC-interním systému souborů, musí být ve <velikosti> zadána hodnota 31 (cesta vč. názvu souboru zde může mít délku max. 30 znaků). U souborů, které jsou uloženy v namontovaném systému souborů, závisí hodnota <velikosti> na maximálním možném počtu znaků, které jsou podporovány externím systémem souborů pro cestu a název souboru.

- 1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

- 2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Provozní režim kanálu	Integer	MCODS(31,<kanál>,<verze>,<vyrovňovací pamě>,1)
E		<p>Udává ve <vyrovňovací paměti> provozní režim aktivní v <kanálu>:</p> <p>0: Není aktivní žádný provozní režim a tedy také žádný proces.</p> <p>1: Krokovací provoz. Osy se dají krokovat (+/-).</p> <p>2: Najetí do referenčního bodu. Osy se dají spouštět pomocí signálů Ruční + / Ruční -.</p> <p>3: Rezervováno.</p> <p>4: Ruční zadávání. Jednotlivé NC-věty mohou být předávány ke zpracování.</p> <p>5: Automatika (následná věta). Dílčí programy se kompletně zpracovávají.</p> <p>6: Automatika (programová věta). Jednotlivé věty dílčího programu se zpracovávají jedna po druhé. Každá jednotlivá věta je připravena a spuštěna pomocí NC-Start.</p> <p>7: Automatika (jednotlivý krok). Z jednotlivé NC-věty v dílčím programu generuje a připravuje NC několik bloků. V tomto provozním režimu se pomocí NC-Start vždy předává jednotlivý blok do interpolátoru ke zpracování.</p> <p>8: Rezervováno.</p> <p>9: Rezervováno.</p> <p>10: Automatika (jednotlivá věta). Pomocí NC-Start se předávají všechny bloky, které byly generovány a připraveny z jednotlivé NC-věty v dílčím programu, do interpolátoru ke zpracování.</p> <p>11: Opětné najetí. Je možné ručně odjízdit s osami od kontur a automaticky nebo ručně k nim opět najíždět.</p> <p>12: CPL-Debugger (provoz s programovými větami): Zpracovávají se jednotlivé věty, jak jsou uvedeny v dílčím programu.</p> <p>13: CPL-Debugger (provoz s následnými větami): Zpracovávají se všechny věty až k dalšímu bodu přerušení.</p>
<p>1) Data, která dává řízení cyklicky k dispozici, jsou označena "Z". Data, která jsou k dispozici po každé změně věty, jsou označena "S". Data, která jsou dodávána v nepravidelných intervalech po změně, jsou označena "E". Data, která jsou dodávána okamžitě po požadavku, jsou označena "I". Data, která se nikdy nemění (stačí si je vyžádat pouze jednou), jsou označena "R".</p>		
<p>2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character). Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".</p>		

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Stav SAV a IPO	Integer, Array	MCODS(32,<kanál>,<verze>,<vyrovnávací pamě>,2)
E		<p>Udává ve <vyrovnávací paměti> z <kanálu></p> <ul style="list-style-type: none"> - stav SAV (příprava vět) a - stav interpolátoru. <p>Jako stav SAV jsou definovány následující hodnoty:</p> <ol style="list-style-type: none"> 1: Provozní režim není aktivní. Je možné zvolit proces. 2: Provozní režim je připravený. Je možné spustit proces. 3: Provozní režim je aktivní. Zpracovává se program nebo NC-věta. 4: Rezervováno 5: Rezervováno 6: IV provozním režimu došlo k chybě, která může být odstraněna pouze základním nastavením nebo zrušením volby programu. 7: Rezervováno 8: Právě se provádí základní nastavení. 9: Je zvolen program a právě se připravuje (např. spojuje). 10: Byl spuštěn příkaz "Vymazat zbylou dráhu" a ještě není ukončen. 11: Provozní režim je aktivní a znovu zpracovává stávající vyrovnávací pamě. 12: Provozní režim je připravený. Proces je na začátku programu a může být spuštěn. 13: Při zadávání NC-vět do vyrovnávací paměti jsou zpracovány všechny věty. Čeká se na další zadání. <p>Jako stav IPO jsou definovány následující hodnoty:</p> <ol style="list-style-type: none"> 1: Interpolátor běží. 2: Interpolátor se kvůli zastavení posuvu deaktivuje. 3: Interpolátor zastavil osy.
<p>1) Data, která dává řízení cyklicky k dispozici, jsou označena "Z". Data, která jsou k dispozici po každé změně věty, jsou označena "S". Data, která jsou dodávána v nepravidelných intervalech po změně, jsou označena "E". Data, která jsou dodávána okamžitě po požadavku, jsou označena "I". Data, která se nikdy nemění (stačí si je vyžádat pouze jednou), jsou označena "R".</p> <p>2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character). Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".</p>		

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Názvy os	Character, Array	MCODS(33,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
R		Pro <kanál> = -1 udává <vyrovnávací pamě> názvy všech systémových os, oddělené znakem "0" (0 bajtů), ve vzestupném pořadí po 9 bajtech. Pro <kanál> = existující číslo kanálu udává <vyrovnávací pamě> názvy všech os v uvedeném kanálu , oddělené znakem "0" (0 bajtů), ve vzestupném pořadí. Názvy, které jsou kratší než 8 znaků, jsou až po 8. znak doplněny mezerami. Ve <velikosti> se zadává velikost <vyrovnávací paměti>. Ta může při 16 osách činit maximálně (9*16) 144 bajtů. Příklad viz kap. 5.13.3 strana 5-90.
Počet posuvových os, pomocných os, vřeten; typy pohybu, typy pohonu	Integer, Array	MCODS(34,-1,<verze>,<vyrovnávací pamě>,<velikost>)
R		Rezervováno Používejte místo toho MCODES(45...)
Skutečná poloha osy	Real, Array	MCODS(35,-1,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> ve vzestupném pořadí, nezávislém na kanálech, skutečné polohy všech posuvových a pomocných os , které jsou sdělovány z pohonů s použitím parametrů SERCOS do CNC: u lineárních os v mm u kruhových os ve stupních "Nastavení skutečné hodnoty" (např. G92) není v hodnotách započítáno.
Skutečný počet otáček vřetena	Real, Array	MCODS(36,-1,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> ve vzestupném pořadí (S, resp. S1, S2, S3 atd.) skutečné počty otáček všech vřeten existujících v systému . Započítány jsou potenciometr, omezení počtu otáček (SMin, SMax) a omezení převodovým stupněm. Pokud vřeteno neexistuje, pak je ve <vyrovnávací paměti> vrácena na příslušném místě hodnota 0.0.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Programová hodnota osy (programová soustava souřadnic)	Double, Array	MCODS(37,-1,<verze>,<vyrovňovací pamě>,<velikost>)
Z		Udává ve <vyrovňovací paměti> ve vzestupném pořadí, nezávislém na kanálech, požadované polohy interpolátoru vzhledem k obrobku pro všechny posuvové osy v systému : u lineárních os v mm u kruhových os ve stupních "Nastavení skutečné hodnoty" (např. G92) a osově posunutí nulových bodů (G54.x ... G59.x) není v hodnotách započítáno.
Skutečné hodnoty os (soustava souřadnic stroje)	Double, Array	MCODS(38,-1,<verze>,<vyrovňovací pamě>,<velikost>)
Z		Udává ve <vyrovňovací paměti> ve vzestupném pořadí, nezávislém na kanálech, "skutečné hodnoty" vzhledem k příslušné (kartézské) soustavě souřadnic stroje. Hodnoty se vypočítávají ze skutečných poloh os za použití kinematických transformací (transformace os), specifikovaných pro kanály a stroje. Pokud není aktivována žádná kinematická transformace os, dává MCODES(38) stejné hodnoty jako MCODES(35). Předpokladem pro použití funkce MCODES(38) je odpovídající nastavení parametru stroje 9030 00002, s jehož použitím se konfiguruje, jestli a jak často se vypočítávají skutečné hodnoty v soustavě souřadnic stroje.
Stav "Doba prodlevy aktivní"	Integer	MCODS(39,<kanál>,<verze>,<vyrovňovací pamě>,1)
E		Udává ve <vyrovňovací paměti> hodnotu 1, když je v <kanálu> aktivována doba prodlevy. Jinak 0.
Stav "Pomocná funkce s povinným potvrzením aktivní"	Integer	MCODS(40,<kanál>,<verze>,<vyrovňovací pamě>,1)
E		Udává ve <vyrovňovací paměti> hodnotu 1, když v <kanálu> čeká pomocná funkce na potvrzení. Jinak 0.
<p>1) Data, která dává řízení cyklicky k dispozici, jsou označena "Z". Data, která jsou k dispozici po každé změně věty, jsou označena "S". Data, která jsou dodávána v nepravidelných intervalech po změně, jsou označena "E". Data, která jsou dodávána okamžitě po požadavku, jsou označena "I". Data, která se nikdy nemění (stačí si je vyžádat pouze jednou), jsou označena "R".</p> <p>2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character). Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".</p>		

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Stav "Uvolnění zadávání"	Integer	MCODS(41,<kanál>,<verze>,<vyrovnávací pamě>,1)
E		Udává ve <vyrovnávací paměti> hodnotu 1, když je pro <kanál> nastaven vstupní signál NC "Blokování načítání". Jinak 0.
Zákaznická data		MCODS(42,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>,<P1>)
E		Udává ve <vyrovnávací paměti> zákaznická data z <kanálu>. V <P1> může být při vyvolání funkce na zákaznickém serveru předána hodnota Integer v rozsahu 0 až 65535 pro selekci určitých dat. Funkce je určena pro vlastní vývoj zákazníků v oblasti "NC-jádra".
Přiřazení osa - kanál	Integer, Array	MCODS(43,-1,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> pro každou systémovou osu následující informace: ≥ 0 : Číslo kanálu synchronní osy -1: Osa je asynchronní -2: Osa je vřetená -3: Osa není definovaná
Počet kanálů	Integer, Array	MCODS(44,-1,<verze>,<vyrovnávací pamě>,3)
R		Udává ve <vyrovnávací paměti> ve vzestupném pořadí <ul style="list-style-type: none"> počet využitelných uživatelských kanálů počet kanálů na rozhraní počet všech interních a externích kanálů

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Počet os	Integer, Array	MCODS(45,-1,<verze>,<vyrovňovací pamě>,3)
R		Udává ve <vyrovňovací paměti> ve vzestupném pořadí <ul style="list-style-type: none"> ● maximální index v systémových osách. Je zapotřebí při vytváření smyček pro všechny osy. Pokud neexistují žádné proluky, jedná se o počet pohonů, které existují v systému. ● maximální index os v systému. Je zapotřebí např. pro velikost osového rozhraní. Pokud neexistují žádné proluky, jedná se o počet os, které existují v systému. ● maximální počet vřeten v systému. Je zapotřebí např. pro velikost vřetenového rozhraní.
Strategie opětovného najetí a zaznamenávání krokových pohybů	Integer, Array	MCODS(46,<kanál>,<verze>,<vyrovňovací pamě>,3)
E		Udává ve <vyrovňovací paměti> pro zadaný <kanál> ve vzestupném pořadí <ul style="list-style-type: none"> ● provozní režim opětovného najetí ● bod opětovného najetí ● stav zaznamenávání krokových pohybů. Jako provozní režim opětovného najetí jsou možné následující hodnoty: <ol style="list-style-type: none"> 1 Automatické opětovné najetí 2 Opětovné najetí s jednotlivou větou 3 Ruční opětovné najetí Pro bod opětovného najetí : <ol style="list-style-type: none"> 1 Opětovné najetí do počátečního bodu 2 Opětovné najetí do koncového bodu 3 Opětovné najetí do bodu přerušení Pro stav zaznamenávání : <ol style="list-style-type: none"> 0 Zaznamenávání není aktivní 1 Zaznamenávání aktivní
<p>1) Data, která dává řízení cyklicky k dispozici, jsou označena "Z". Data, která jsou k dispozici po každé změně věty, jsou označena "S". Data, která jsou dodávána v nepravidelných intervalech po změně, jsou označena "E". Data, která jsou dodávána okamžitě po požadavku, jsou označena "I". Data, která se nikdy nemění (stačí si je vyžádat pouze jednou), jsou označena "R".</p> <p>2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character). Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".</p>		

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Stav "Příkaz pojezdu"	Integer, Array	MCODS(47,<kanál>,<verze>,<vyrovňovací pamě>,<velikost>)
Z		<p>Pro <kanál> = -1 udává <vyrovňovací pamě> signály příkazu pojezdu všech systémových os ve vzestupném pořadí.</p> <p>Pro <kanál> = existující číslo kanálu udává <vyrovňovací pamě> signály příkazu pojezdu všech os v zadaném kanálu a za nimi asynchronních os ve vzestupném pořadí.</p> <p>Příkaz pojezdu je vydán: 1 Příkaz pojezdu není vydán: 0</p> <p>Příkaz pojezdu je nastaven vždy, když má osa na základě ručního zadání nebo zadání v dílčím programu provést pojezdový pohyb.</p>

- 1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
 Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
 Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
 Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
 Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

- 2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character).
 Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dáva / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Provozní režim osy	Integer, Array	MCODS(48,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		<p>Pro <kanál> = -1 udává <vyrovnávací pamě> provozní režimy všech systémových os ve vzestupném pořadí. Pro <kanál> = existující číslo kanálu udává <vyrovnávací pamě> provozní režimy všech os v zadaném kanálu a za nimi asynchronních os ve vzestupném pořadí.</p> <p>Možné vrácené hodnoty pro provozní režimy:</p> <p>0: Není aktivní žádný provozní režim a tedy také žádný proces.</p> <p>1: Krokovací provoz. Osy se dají krokovat (+/-).</p> <p>2: Najetí do referenčního bodu. Osy se dají spouštět pomocí signálů Ruční + / Ruční -.</p> <p>3: Rezervováno.</p> <p>4: Ruční zadávání. Jednotlivé NC-věty mohou být předávány ke zpracování.</p> <p>5: Automatika (následná věta). Dílčí programy se kompletně zpracovávají.</p> <p>6: Automatika (programová věta). Jednotlivé věty dílčího programu se zpracovávají jedna po druhé. Každá jednotlivá věta je připravena a spuštěna pomocí NC-Start.</p> <p>7: Automatika (jednotlivý krok). Z jednotlivé NC-věty v dílčím programu generuje a připravuje NC několik bloků. V tomto provozním režimu se pomocí NC-Start vždy předává jednotlivý blok do interpolátoru ke zpracování.</p> <p>8: Rezervováno.</p> <p>9: Rezervováno.</p> <p>10:Automatika (jednotlivá věta). Pomocí NC-Start se předávají všechny bloky, které byly generovány a připraveny z jednotlivé NC-věty v dílčím programu, do interpolátoru ke zpracování.</p> <p>11: Opětovné najetí. Je možné ručně odjíždět s osami od kontur a automaticky nebo ručně k nim opět najíždět.</p>

- 1) Data, která dáva řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

- 2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Stav "Blokování posuvu"	Integer, Array	MCODS(49,<kanál>,<verze>,<vyrovňovací pamě>,<velikost>)
E		Pro <kanál> = -1 udává <vyrovňovací pamě> signály blokování posuvu všech systémových os ve vzestupném pořadí. Pro <kanál> = existující číslo kanálu udává <vyrovňovací pamě> blokování posuvu všech os v zadaném kanálu a za nimi asynchronních os ve vzestupném pořadí. 1: Blokování posuvu je aktivováno 0: Blokování posuvu není aktivováno
Hodnota potenciometrů os	Real, Array	MCODS(50,<kanál>,<verze>,<vyrovňovací pamě>,<velikost>)
E		Pro <kanál> = -1 udává <vyrovňovací pamě> hodnoty potenciometrů všech systémových os ve vzestupném pořadí (v 0.01 procenta). Pro <kanál> = existující číslo kanálu udává <vyrovňovací pamě> pro každou osu v zadaném kanálu hodnotu potenciometru kanálu a následně hodnoty potenciometrů všech asynchronních os ve vzestupném pořadí (v 0.01 procenta).
Aktivní hodnoty externí opravy nástroje	Real, Array	MCODS(51,<kanál>,<verze>,<vyrovňovací pamě>,<velikost>)
S		Udává ve <vyrovňovací paměti> hodnoty externí opravy nástroje aktivní v <kanálu>. Pořadí: Oprava poloměru, oprava délky Není-li aktivní žádná externí oprava nástroje, vrací se vždy hodnota 0.0.
Aktivní hodnoty externích osových posunutí nulových bodů	Real, Array	MCODS(52,<kanál>,<verze>,<vyrovňovací pamě>,<velikost>)
S		Udává ve <vyrovňovací paměti> hodnoty externích osových posunutí nulových bodů, aktivní v <kanálu>. Pořadí: 1. log. osa, ... 8. log. osa Není-li aktivní žádné externí posunutí, vrací se vždy hodnota 0.0.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Měrná jednotka os	Integer, Array	MCODS(53,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		<p>Pro <kanál> = -1 udává <vyrovnávací pamě> měrné jednotky (metrické, inch, stupně) všech systémových os ve vzestupném pořadí.</p> <p>Pro <kanál> = existující číslo kanálu udává <vyrovnávací pamě> měrnou jednotku pro každou osu v zadaném kanálu a následně měrnou jednotku všech asynchronních os ve vzestupném pořadí.</p> <p>U asynchronních lineárních os v osových provozních režimech "krokování" a "njetí do referenčního bodu" určuje měrnou jednotku osové rozhraní. Pokud není zadán žádný osový provozní režim, udává se hodnota "metrická".</p> <p>U synchronních lineárních os v kanálových provozních režimech "krokování" a "njetí do referenčního bodu" určuje měrnou jednotku osové rozhraní. V ostatních provozních režimech závisí na kanálové měrné jednotce, metrické/Inch (G70/G71).</p> <p>U kruhových os a vřeten se udává měrná jednotka stupeň, u Hirthových os s programováním místa zase odpovídající měrná jednotka.</p> <p>Možné vrácené hodnoty pro měrné jednotky:</p> <p>0: inch 1: metrické 2: stupně 3: Osa neexistuje 4: Hirthova osa s programováním místa</p>
<p>1) Data, která dává řízení cyklicky k dispozici, jsou označena "Z". Data, která jsou k dispozici po každé změně věty, jsou označena "S". Data, která jsou dodávána v nepravidelných intervalech po změně, jsou označena "E". Data, která jsou dodávána okamžitě po požadavku, jsou označena "I". Data, která se nikdy nemění (stačí si je vyžádat pouze jednou), jsou označena "R".</p> <p>2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character). Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".</p>		

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Aktivní univerzální oprava nástroje	Real, Array	MCODS(54,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
S		<p>Udává ve <vyrovnávací paměti> hodnoty univerzální opravy nástroje, aktivní v <kanálu>.</p> <p>Pořadí:</p> <ul style="list-style-type: none"> ● Oprava poloměru ● Oprava délky L3 ● Oprava délky L1 ● Oprava délky L2 ● Poloha břitů ● Typ opravy <p>Jsou definovány následující typy oprav:</p> <p>0: Žádná oprava 1: Vrtací nástroj 2: Frézovací nástroj 3: Soustružnický nástroj 4: Nástroj s úhlovou hlavou</p> <p>Není-li aktivní žádná univerzální oprava nástroje, vrací se vždy hodnota 0.0.</p>
Verze výrobce	Character, Array	MCODS(55,-1,<verze>,<vyrovnávací pamě>,<velikost>,<číslo osy>)
R		<p>Udává verzi výrobce pohonu. Výběr osy se provádí v parametru <číslo osy> zadáním systémové osy. (Č. 0 dává rovněž 1. osu.) Verze výrobce odpovídá parametru SERCOS S-0-0030. V parametru <vyrovnávací pamě> se udává pole s max. 40 znaky.</p>
Typ regulátoru	Character, Array	MCODS(56,-1,<verze>,<vyrovnávací pamě>,<velikost>,<číslo osy>)
R		<p>Udává typ regulátoru pohonu. Výběr osy se provádí v parametru <číslo osy> zadáním systémové osy. (Č. 0 dává rovněž 1. osu.) Typ regulátoru odpovídá parametru SERCOS S-0-0140. V parametru <vyrovnávací pamě> se udává pole s max. 40 znaky.</p>
<p>1) Data, která dává řízení cyklicky k dispozici, jsou označena "Z". Data, která jsou k dispozici po každé změně věty, jsou označena "S". Data, která jsou dodávána v nepravidelných intervalech po změně, jsou označena "E". Data, která jsou dodávána okamžitě po požadavku, jsou označena "I". Data, která se nikdy nemění (stačí si je vyžádat pouze jednou), jsou označena "R".</p> <p>2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character). Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".</p>		

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Typ motoru	Character, Array	MCODS(57,-1,<verze>,<vyrovnávací pamě>,<velikost>,<číslo osy>)
R		Udává typ motoru pohonu. Výběr osy se provádí v parametru <číslo osy> zadáním systémové osy. (Č. 0 dává rovněž 1. osu.) Typ motoru odpovídá parametru SERCOS S-0-0141. V parametru <vyrovnávací pamě> se udává pole s max. 40 znaky.
Standardní přiřazení osa - kanál	Integer	MCODS(58,-1,<verze>,<vyrovnávací pamě>,<velikost>)
R		Udává ve <vyrovnávací paměti> pro každou systémovou osu následující standardní přiřazení: ≥0: Číslo kanálu synchronní osy -1: Osa je asynchronní -2: Osa je vřeteno -3: Osa není definovaná Při 16 osách musí <vyrovnávací pamě> mít <velikost> 16 (Integer).
Aktivní názvy kanálových os	Character, Array	MCODS(59,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		Pro <kanál> = -1 udává <vyrovnávací pamě> názvy všech aktivních kanálových os, oddělené znakem "0" (0 bajtů), ve vzestupném pořadí po 9 bajtech. Pro <kanál> = existující číslo kanálu udává <vyrovnávací pamě> názvy všech os v uvedeném kanálu , oddělené znakem "0" (0 bajtů), ve vzestupném pořadí. Názvy, které jsou kratší než 8 znaků, jsou až po 8. znak doplněny mezerami. Ve <velikosti> se zadává velikost <vyrovnávací paměti>. Ta může při 16 osách činit maximálně (9*16) 144 bajtů. Příklad viz kap. 5.13.3 strana 5-90.

- 1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
 Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
 Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
 Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
 Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

- 2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
 Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Standardní nastavení názvů kanálových os	Character, Array	MCODS(60,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
R		<p>Pro <kanál> = -1 udává <vyrovnávací pamě> názvy všech kanálových os (ve standardním nastavení), oddělené znakem "0" (0 bajtů), ve vzestupném pořadí po 9 bajtech.</p> <p>Pro <kanál> = existující číslo kanálu udává <vyrovnávací pamě> názvy všech os v uvedeném kanálu, oddělené znakem "0" (0 bajtů), ve vzestupném pořadí.</p> <p>Názvy, které jsou kratší než 8 znaků, jsou až po 8. znak doplněny mezerami.</p> <p>Ve <velikosti> se zadává velikost <vyrovnávací paměti>. Ta může při 16 osách činit maximálně (9*16) 144 bajtů.</p> <p>Příklad viz kap. 5.13.3 strana 5-90.</p>
Měrné jednotky standardního nastavení os	Integer, Array	MCODS(61,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
R		<p>Pro <kanál> = -1 udává <vyrovnávací pamě> měrné jednotky (metrické, inch, stupně) všech systémových os (ve standardním nastavení), oddělené znakem "0" (0 bajtů), ve vzestupném pořadí po 9 bajtech.</p> <p>Pro <kanál> = existující číslo kanálu udává <vyrovnávací pamě> názvy všech os v uvedeném kanálu, oddělené znakem "0" (0 bajtů), ve vzestupném pořadí.</p> <ul style="list-style-type: none"> ● U asynchronních lineárních os se udávají "metrické". ● U synchronních lineárních os závisí měrná jednotka na stavu zapnutí po spuštění (parametr stroje 7060 00010): "metrické/Inch" (G70/G71) ● U kruhových os a vřeten se udává měrná jednotka stupeň, u Hirthových os s programováním místa zase odpovídající měrná jednotka. <p>Možné vrácené hodnoty pro měrné jednotky:</p> <p>0: inch 1: metrické 2: stupně 3: Osa neexistuje 4: Hirthova osa s programováním místa</p> <p>Při 16 osách musí <vyrovnávací pamě> mít <velikost> 16 (Integer)</p>

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Identifikační číslo z cyklického osového telegramu	Integer, Array	MCODS(62,-1,<verze>,<vyrovnávací pamě>,<velikost>,<identifikační číslo>)
Z		<p>Udává hodnotu <identifikačního čísla> z cyklického osového telegramu pro všechny osy. Pokud <identifikační číslo> není v cyklickém telegramu, vrací se hodnota NCS_MCO_NOT_IN_CYCL_AT_C (-2147483648). Hodnota je "Integer" ve vážení SERCOS.</p> <p>Při 16 osách musí <vyrovnávací pamě> mít <velikost> 16 (Integer)</p>
Pohybové funkce vřeten	Integer, Array	MCODS(63,-1,<verze>,<vyrovnávací pamě>,<velikost>)
E		<p>Udává pohybové funkce všech vřeten. Kódování pohybových funkcí: 0: Vřeteno není definováno 1: Pravotočivý chod bez chladicí kapaliny 2: Pravotočivý chod s chladicí kapalinou 3: Levotočivý chod bez chladicí kapaliny 4: Levotočivý chod s chladicí kapalinou 5: Zastavení vřetena 6: Seřízení vřetena</p> <p>Při 8 vřetenech musí <vyrovnávací pamě> mít <velikost> 8 (Integer)</p>
Převodové stupně vřeten	Integer, Array	MCODS(64,-1,<verze>,<vyrovnávací pamě>,<velikost>)
E		<p>Udává převodové stupně vřeten. Kódování převodových stupňů: 40:Automatická volba převodových stupňů 41:Převodový stupeň 1 42:Převodový stupeň 2 43:Převodový stupeň 3 44:Převodový stupeň 4 48:Vyřazení převodového stupně</p> <p>Při 8 vřetenech musí <vyrovnávací pamě> mít <velikost> 8 (Integer)</p>

- 1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

- 2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Automatická, resp. ruční volba převodových stupňů	Integer, Array	MCODS(66,-1,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává automatickou nebo ruční volbu převodových stupňů: 0: Ruční 1: Automatická Při 8 vřetenech musí <vyrovnávací pamě> mít <velikost> 8 (Integer)
Informace, zda je přepínání převodů aktivní	Integer, Array	MCODS(67,-1,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává informaci, zda je přepínání převodů aktivní: 0: Přepínání převodů není aktivní 1: Přepínání převodů je aktivní Při 8 vřetenech musí <vyrovnávací pamě> mít <velikost> 8 (Integer)
Vynechat větu (aktivovat)	Integer, Array	MCODS(68,<kanál>,<verze>,<vyrovnávací pamě>,2)
E		Udává ve <vyrovnávací paměti> stav výstupního signálu NC Aktivovat vynechání věty a vstupního signálu Vynechat větu z <kanálu>.
Volitelné zastavení (aktivované)	Integer, Array	MCODS(69,<kanál>,<verze>,<vyrovnávací pamě>,2)
E		Udává ve <vyrovnávací paměti> stav výstupního signálu NC Volitelné zastavení aktivováno a vstupního signálu Volitelné zastavení <kanálu>.
Automatická opakovaná volba programu aktivní	Integer	MCODS(70,<kanál>,<verze>,<vyrovnávací pamě>,1)
I		Udává ve <vyrovnávací paměti>, zda je v zadaném kanálu aplikována automatická opětná volba programu : 0: Funkce není aplikována 1: Funkce je aplikována
Souřadnice obrobku	Real, Array	MCODS(71,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> hodnoty souřadnic obrobku (WCS) zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Požadované hodnoty základních souřadnic	Real, Array	MCODS(72,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> požadované hodnoty základních souřadnic (BCS) zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.
Osové souřadnice	Real, Array	MCODS(73,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> hodnoty osových souřadnic (ACS) zadaného kanálu: Kanál = -1: Všechny osové souřadnice 0 < Kanál ≤ max. kanál: Data zadaného kanálu
Souřadnice stroje	Real, Array	MCODS(74,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> hodnoty souřadnic stroje (MCS) zadaného kanálu: Kanál = -1: Všechny osové souřadnice 0 < Kanál ≤ max. kanál: Data zadaného kanálu
Skutečné hodnoty základních souřadnic	Real, Array	MCODS(75,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> skutečné hodnoty základních souřadnic (BCS) zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.
Naprogramované koncové body souřadnic	Real, Array	MCODS(76,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
S		Udává ve <vyrovnávací paměti> naprogramované koncové body souřadnic zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.
Koncové body souřadnic	Real, Array	MCODS(77,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
S		Udává ve <vyrovnávací paměti> koncové body souřadnic zadaného kanálu včetně posunutí: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Názvy souřadnic	Character, Array	MCODS(78,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> názvy aktivních souřadnic zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.
Stav INPOS souřadnic	Integer, Array	MCODS(79,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> stav INPOS souřadnic zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu. Pro prostorovou souřadnici se vytváří stav z logické operace AND osových signálů.
Referenční stav souřadnic	Integer, Array	MCODS(80,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> stav referenčních bodů souřadnic zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu. Pro prostorovou souřadnici se vytváří stav z logické operace AND osových signálů.
Měrná jednotka souřadnic	Integer, Array	MCODS(81,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> měrné jednotky souřadnic zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu. Možné vrácené hodnoty pro měrné jednotky: 0: inch 1: metrické 2: stupně 3: Souřadnice neexistuje
Počet souřadnic	Integer, Array	MCODS(82,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
E		Udává ve <vyrovnávací paměti> ve 3 prvcích počet souřadnic/os zadaného kanálu: 1. hodnota: Celkový počet os kanálu 2. hodnota: Počet prostorových souřadnic + počet pseudosouřadnic kanálu. 3. hodnota: Počet pseudosouřadnic kanálu.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Zbývající dráha souřadnic obrobku	Real, Array	MCODS(83,<kanál>,<verze>,<vyrovnávací pamě>,<velikost>)
Z		Udává ve <vyrovnávací paměti> zbývající dráhy souřadnic obrobku (WCS) zadaného kanálu: nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.
Stavy čekání kanálu	Integer	MCODS(87,<kanál>,<verze>,<vyrovnávací pamě>,1)
E		Udává ve <vyrovnávací paměti> stavy čekání kanálu . Jestliže kanál přejde do stavu čekání, sděluje tato funkce příčiny. Aktivní stavy čekání jsou bitově kódovány. Následující konstanty definují odpovídající bity první hodnoty Integer, počínaje nejnižší hodnotou: 0: Doba prodlevy 1: Pomocná funkce s povinným potvrzením 2: Blokování načítání 3: Posuv v kanálu roven 0 4: Zastavení programu s M0/M1 5: Zastavení posuvu v kanálu 6: Blokování posuvu v kanálu nebo kanálové ose 7: Blokování načítání zadané uživatelem 8: Synchronizované zastavení pohybu mezi kanály (ASTOP, ...) 9: Čekání na osu při změně osy (G511) 10: Čekání na permanentní proměnnou (WPV) 11: Čekání na signál rozhraní v aktivním okamžiku (WAITA, ...) 12: Čekání na signál rozhraní (WAIT(BITIF...)) nebo na stanovený časový interval (WAIT(,ZEIT%)) v přípravě vět 13: Datová služba Motion Control (MCODS(...)) Viz příklad pod 5.13.3
Hodnoty online oprav (WCS)	Real, Array	MCODS(89, <kanál>, <verze>, <vyrovnávací pamě>, <velikost>)
E		Udává ve <vyrovnávací paměti> hodnoty online opravy (WCS) zadaného kanálu. Nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovňovací pamě> je typu ²⁾	Syntaxe / popis
Stav online opravy (WCS)	Integer, Array	MCODS(90, <kanál>, <verze>, <vyrovňovací pamě>, <velikost>)
E		Udává ve <vyrovňovací paměti> aktuální stav online opravy (WCS) zadaného kanálu. Nejprve všechny prostorové souřadnice a pak pseudosouřadnice kanálu. 0: Neaktivní 1: Aktivní
Programování průměru	Integer, Array	MCODS(91, <kanál>, <verze>, <vyrovňovací pamě>, <velikost>)
S		Udává ve <vyrovňovací paměti> hodnotu 1 pro osy s programováním průměru. Pokud je jako kanál zadána hodnota Ncs_MCoNoChannel_C (-1), jsou udány hodnoty všech systémových os. Je-li zadáno číslo kanálu, vracejí se hodnoty kanálových os.
Základní souřadnice hrotu nástroje (BCS-Tcp)	Real, Array	MCODS(92, <kanál>, <verze>, <vyrovňovací pamě>, <velikost>)
Z		Udává při aktivní 3D opravě poloměru ve <vyrovňovací paměti> požadované hodnoty hrotu nástroje v základní soustavě souřadnic (BCS-Tcp), nejprve prostorové souřadnice a pak pseudosouřadnice zadaného kanálu. Při vypnuté 3D opravě poloměru jsou hodnoty stejné jako u MCODES(72,...).
Čísla skupin pomocných funkcí	Character, Array	MCODS(94, <kanál>, <verze>, <skupina>, <velikost>, <syntaxe>)
Z		Udává ve <skupině> čísla skupin k pomocným funkcím zadaného kanálu uvedeným v <syntaxi>. <Syntaxe> obsahuje pomocné funkce, oddělené mezerami a ukončené CHR\$(0), viz příklad na straně 5-81.

1) Data, která dává řízení **cyklicky** k dispozici, jsou označena "Z".
Data, která jsou k dispozici **po každé změně věty**, jsou označena "S".
Data, která jsou dodávána v **nepravidelných intervalech** po změně, jsou označena "E".
Data, která jsou dodávána **okamžitě** po požadavku, jsou označena "I".
Data, která se **nikdy** nemění (stačí si je vyžádat pouze jednou), jsou označena "R".

2) Udává typ proměnné potřebný pro <vyrovňovací pamě> (Integer, Real, Double, Character).
Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".

Systémová data

Funkce dává / Refresh ¹⁾	<Vyrovnávací pamě> je typu ²⁾	Syntaxe / popis
Aktivní syntaxe skupin pomocných funkcí	Character, Array	MCODS(95, <kanál>, <verze>, <syntaxe>, <velikost>, <skupina>)
Z		Udává v <syntaxi> aktivní syntaxi k číslům skupin pomocných funkcí zadaného kanálu uvedených ve <skupině>. Ve <skupině> se předává pole požadovaných čísel skupin pomocných funkcí, uzavřené příkazem CHR\$(0), viz příklad níže.
<p>1) Data, která dává řízení cyklicky k dispozici, jsou označena "Z". Data, která jsou k dispozici po každé změně věty, jsou označena "S". Data, která jsou dodávána v nepravidelných intervalech po změně, jsou označena "E". Data, která jsou dodávána okamžitě po požadavku, jsou označena "I". Data, která se nikdy nemění (stačí si je vyžádat pouze jednou), jsou označena "R".</p> <p>2) Udává typ proměnné potřebný pro <vyrovnávací pamě> (Integer, Real, Double, Character). Pokud je zapotřebí nikoli jednoduchá proměnná, ale pole proměnných, je za typem proměnné uvedeno ještě "Array".</p>		

Příklad k pomocným funkcím:

```

:
N10 M3 S1234 T5678
WAIT
10 DIM GROUPID$(64)
15 DIM SYNTAX$(256)
17 VERSION=0
18 CHAN%=1
20 REM Najít čísla skupin M3 a T
25 SYNTAX$="T M3"+CHR$(0)
30 ERR_VAR%=MCODS(94,CHAN%,VERSION,GROUPID$,64,SYNTAX$)
31 PRN#(0,"94 chyba:",ERR_VAR%)
32 PRN#(0,"T M3:",ASC(MID$(GROUPID$,1,1)),
ASC(MID$(GROUPID$,2,1)))
33 REM Přečíst skupinu 2 a 1
36 GROUPID$=CHR$(2)+CHR$(1)+CHR$(0) :
37 REM Zeptat se na skupinu 2 a 1
50 VERSION=0
58 CHAN%=1
60 ERR_VAR%=MCODS(95,CHAN%,VERSION,SYNTAX$,256,GROUPID$)
65 PRN#(0,"95 chyba:",ERR_VAR%)
70 PRN#(0,"Skupina 2 1 : ",SYNTAX$)
71 REM SYNTAX$ obsahuje např. "S1234 M3"

```

Systémová data

MCOPS

Vyvolává procesové služby Motion Control v NCS pomocí CPL. Tak lze řídit kanály v CNC.


Všeobecná syntaxe:


MCOPS (<funkce>, <kanál> [[, [<P1>] [, [<P2>] , [<P3>]]] , <P4>])

<Funkce> Výraz Integer. Udává prováděnou funkci. V následující tabulce jsou popsány všechny dostupné funkce.

<Kanál> Výraz Integer. Udává kanál, na který má funkce působit.

<P1> ... <P4> Volitelné parametry, které jsou závislé na <funkci>. Je přípustných několik čárek za sebou, ale žádná čárka před uzavírající závorkou.

 **V následující tabulce jsou u syntaxe uvedeny jako parametry zčásti konstanty Integer. Místo těchto konstant můžete naprogramovat také proměnné Integer, které ovšem musí být v okamžiku vyvolání funkce obsazeny zadanou hodnotou.**

Následek	Syntaxe / popis
Vymazat zbývající dráhu	MCOPS(1,<kanál>)
	<p>Spouští funkci "Vymazat zbývající dráhu" v naprogramovaném <kanálu>:</p> <ul style="list-style-type: none"> Po spuštění vymazání zbývající dráhy jsou odmítnuty všechny připravené NC-věty včetně zbytku aktuální věty a znovu připraveny. <p> Věty CPL nebo součásti CPL zůstávají beze změny:</p> <p>Příklad: Proměnná CPL POS měla při přípravě hodnotu 10. NC-slovo X[POS] je po "vymazání zbývající dráhy" interpretováno jako X10, ačkoli v tomto okamžiku má POS pravděpodobně úplně jinou hodnotu. Případně změněné hodnoty oprav jsou brány v úvahu.</p> <ul style="list-style-type: none"> V zobrazení je zobrazený koncový bod nastaven do aktuální polohy, čímž je zároveň vymazána zobrazená zbývající dráha. <Kanál> pak přejde do stavu NC-připraveno (vstupní signál SPS iCh_NCRReady). Po aktivaci NC-Start (výstupní signál SPS qCh_NCStart) pokračuje program na místě přerušení, s ohledem na nové hodnoty oprav. <p>Příklad použití MCOPS(1,<kanál>): po změně tabulek oprav, když mají nové hodnoty platit i pro již připravené věty.</p>


Systémová data

Následek	Syntaxe / popis
Základní nastavení	MCOPS(2,<kanál>[,<typ základního nastavení>])
	<p>Spouští funkci "Základní nastavení" v naprogramovaném <kanálu>. Chcete-li spustit celkové základní nastavení: <kanál> = -2</p> <ul style="list-style-type: none"> • Kanál nejprve nepřijímá žádné další úlohy, jako např. volba programu nebo přepnutí provozního režimu. • Interpolátor se zastaví. • Dosud nezpracované úlohy v kanálu jsou zrušeny. • Je zrušena volba hlavního programu. • Změněné parametry stroje, které nevyžadují inicializaci, jsou převzaty, např. parametr stroje 1020 00001 (softwarový koncový vypínač). • Chyby a varování, které tento kanál generoval, jsou vymazány. • Interpolátor se opět spustí. • Stav zapnutí při základním nastavení (parametr stroje 7060 00020) je převzat, tzn. aktivují se příslušné modální stavy. • Kanál vydává signál rozhraní 0.2 'Základní nastavení provedeno' a je opět připraven pro nové úlohy. • <Typ základního nastavení>: Výraz Integer. Určuje chování funkce. Následující seznam obsahuje všechny definované vzorce chování. Každý vzorec je označen identifikačním číslem. Chcete-li nastavit určité chování, musíte v <typu základního nastavení> zadat odpovídající identifikační číslo. Pokud chcete zkombinovat několik vzorců chování, musíte v <typu základního nastavení> předat funkci součet všech odpovídajících identifikačních čísel. Zatím obsahuje seznam pouze jeden prvek: Identifikační číslo: 2: Automatická opakovaná volba programu je potlačena, pokud je aktivní. <p>Příklad: ERR_VAR=MCOPS(2,2,2) Základní nastavení ve 2. kanálu bez automatické opakované volby programu</p>

Systémová data

Následek	Syntaxe / popis
Hledat větu	<p>MCOPS(3,<kanál>[,<počáteční věta>],[<koncová věta>]])</p> <p>Spouští funkci "Hledat větu" ve zvoleném, ale ještě nespouštěném hlavním programu naprogramovaného <kanálu>:</p> <ul style="list-style-type: none"> ● <Počáteční věta> a <koncová věta> jsou předávány jako výrazy String. Při hledání <počáteční věty> a <koncové věty> platí následující konvence: <ul style="list-style-type: none"> ● Mezera, <Tab> a <LF> na začátku NC-věty jsou ignorovány. ● Jestliže <počáteční věta>, resp. <koncová věta> začíná číslicí a výraz není v prohledávaném programu nalezen, hledá systém výraz znovu; tentokrát ale se znakem "N" vpředu. Tímto způsobem nachází např. "50" také NC-větu "N50X100". ● Pokud <počáteční věta>, resp. <koncová věta> končí číslicí, je výraz nalezen v prohledávaném programu jedině tehdy, když bezprostředně za ním nenásleduje žádná další číslice. Např. "G1X10" nenalezne NC-větu "G1X100" .. ● Pokud <počáteční věta>, resp. <koncová věta> končí písmenem, je výraz nalezen v prohledávaném programu jedině tehdy, když bezprostředně za ním následuje mezera. Např. "50A" nenalezne NC-větu "50A =1", ale "50A=1". ● Zpracování začíná <počáteční větou> a končí <koncovou větou>. Pokud chybí <počáteční věta> nebo není nalezena, začíná zpracování na začátku programu. Pokud chybí <koncová věta> nebo není nalezena, zastavuje se zpracování na konci programu. ● NC-stav se změní na PŘIPRAVENO. <p>Příklad: ERR_VAR=MCOPS(3,2,"N50","N100") Spouští v kanálu 2 funkci "Hledat větu". Hlavní program má být zpracován počínaje větou N50 po větu N100 včetně.</p>
Zvolte program nebo vyberte string pro ruční zadání	<p>MCOPS(4,<kanál>[,<string>],[<počáteční věta>],[<koncová věta>]],<typ volby>)]</p> <p>Vybírá v naprogramovaném <kanálu> program pro zpracování nebo string pro zpracování v provozním režimu ručního zadávání. Pro <typ volby> = 32768 je možné zadávat asynchronní podprogramy. Můžete je přihlašovat, vypínat, znovu zapínat, mazat a spouštět.</p> <ul style="list-style-type: none"> ● <String>: Výraz String V závislosti na <typu volby> interpretuje systém parametr jako <ul style="list-style-type: none"> ● název cesty (vč. názvu dílčího programu) vybraného dílčího programu (maximálně 100 znaků), nebo ● pokud je v <typu volby> zadáno 32: jako NC-větu (velikost max. 512 bajtů vč. ukončujícího 0 bajtu), která má být zpracována v provozním režimu ručního zadávání, nebo

Systémová data

Následek	Syntaxe / popis
	<ul style="list-style-type: none"> ● pokud je v <i><typu volby></i> zadáno 32+4096: jako několik NC-vět, které mají být zpracovány v provozním režimu ručního zadávání. Několik NC-vět je odděleno pomocí NewLine ("\\n", Hex 0x0A). Max. velikost všech NC-vět nesmí překročit 4096 bajtů včetně ukončujícího 0 bajtu. ● pokud je v <i><typu volby></i> zadáno 32768: V úloze "SETINT" pro asynchronní podprogramy je zadán název nebo název cesty. <p>● <i><Počáteční věta></i> a <i><koncová věta></i> určují počáteční a koncovou větu v dílčím programu pro obrábění. Postup jako u MCOPS(3,...). Když systém interpretuje <i><string></i> jako ruční zadávání, jsou <i><počáteční věta></i> a <i><koncová věta></i> ignorovány. Při zadávání pro asynchronní podprogramy (<i><typ volby></i>= 32768) se v <i><počáteční větě></i> zadává číslo ($1 \leq \text{číslo} \leq 8$) jako string (např. "1") a v <i><koncové větě></i> úloha.</p> <p>Úlohy pro asynchronní podprogramy jsou:</p> <ul style="list-style-type: none"> "SETINT" přihlásit "DISABLE" vypnout "ENABLE" znovu zapnout "CLRINT" vymazat "START" spustit <p>● <i><Typ volby></i>: Výraz Integer. Určuje chování funkce. Následující seznam obsahuje všechny definované vzorce chování. Každý vzorec je označen identifikačním číslem. Chcete-li nastavit určité chování, musíte do <i><typu volby></i> předat toto identifikační číslo. Pokud chcete zkombinovat několik vzorců chování, musíte v <i><typu volby></i> předat funkci součet všech odpovídajících identifikačních čísel.</p> <p> Je-li zadán <i><typ volby></i> 2 (čekání, až se NC-stav změní na PŘIPRAVENO) a vybraný program neexistuje nebo není proveditelný, vrací se chybové hlášení 6. Ve všech ostatních případech se nekontroluje, zda je program proveditelný. Funkce dává 0 (nedošlo k žádné chybě).</p> <p>Teprve následné spojení generuje odpovídající běhovou chybu.</p>

Systémová data

Následek	Syntaxe / popis
	1 Během volby provádí systém spojení. Pokud ještě neexistuje, je pro zvolený hlavní program vytvořena tabulka spojení. Tabulky spojení jsou nezbytné, když v programu existují volání podprogramů nebo instrukce CPL.
	2 Systém potvrzuje přípustnost volby až poté, co se NC-stav změní na PŘIPRAVENO. Normálně se volba potvrzuje bez čekání na NC-stav PŘIPRAVENO.
	32 Systém interpretuje <string> jako větu ručního zadávání. Viz též identifikační číslo 128.
	64 Než funkce vybere zadaný program nebo zadanou větu ručního zadávání, zruší se nejprve volba aktivního programu či aktivního ručního zadávání.
	128 Je okamžitě spuštěna věta ručního zadávání. Rozlišují se 2 případy: <Kanál> není aktivní: věta je okamžitě zpracována jako normální věta ručního zadávání. <Kanál> je již aktivní: věta je okamžitě zpracována jako spínací funkce. Omezení viz identifikační číslo 1024.
	256 Předpoklad pro pojezd os v provozním režimu "krokování" nebo když mají probíhat pohyby v provozním režimu "krokování v souřadnicích obrobku".
	512 Předpoklad pro spuštění os v provozním režimu "njetí do referenčního bodu".
	1024 Spínací funkce. Působí ve spojení s identifikačním číslem 128. Věta ručního zadávání se zpracovává paralelně se zadaným <kanálem>. Ve větě ručního zadávání jsou ovšem přípustné pouze pomocné funkce a asynchronní pohyby os.
	2048 Již aktivní program je nahrazen nově zvoleným. Přitom zůstávají zachovány všechny modální stavy. Při ručním zadávání se starý znakový řetězec nahrazuje novým.
	4096 Zadávání NC-vět do vyrovnávací paměti působí ve spojení s identifikačním číslem 32. Zatímco se ještě zpracovávají předcházející věty, mohou již být zadávány další.
	32768 Pomocí tohoto příznaku je možné řídit asynchronní podprogramy (viz <počáteční věta> a <koncová věta>).
	Příklady: ERR_VAR=MCOPS(4,1,"sekt.cnc","N50","N100",1) Volba programu "sekt.cnc" v kanálu 1 vč. hledání vět a spojení.
	ERR_VAR=MCOPS(4,1,"usr/user/p1.cnc") Volba programu "p1.cnc" v kanálu 1 bez hledání vět a spojení.
	ERR_VAR=MCOPS(4,1,"F1000G1X500",,,32) Vybírá v kanálu 1 větu "F1000G1X500" při ručním zadávání.

Systémová data

Následek	Syntaxe / popis
Zrušení volby programu	<p data-bbox="643 288 1126 322">MCOPS(5,<kanál>,<typ zrušení volby>)</p> <p data-bbox="643 338 1465 398">Ruší v naprogramovaném <kanálu> zvolený program nebo zvolenou větu ručního zadávání.</p> <ul data-bbox="643 434 1516 685" style="list-style-type: none"> • <Typ zrušení volby>: Výraz Integer. Určuje chování funkce. Následující seznam obsahuje všechny definované vzorce chování. Každý vzorec je označen identifikačním číslem. Chcete-li nastavit určité chování, musíte v <typu zrušení volby> zadat odpovídající identifikační číslo. Pokud chcete zkombinovat několik vzorců chování, musíte v <typu zrušení volby> předat součet všech odpovídajících identifikačních čísel. Zatím obsahuje seznam pouze jedno identifikační číslo: <p data-bbox="683 696 1497 786">2: Pokud je aktivována automatická opakovaná volba programu, může být hodnotou 2 v tomto zrušení volby programu potlačena.</p> <p data-bbox="643 824 1474 920">Příklad: ERR_VAR=MCOPS(5,2,2) Zrušení volby programu v kanálu 2 bez automatické opakované volby programu.</p>
Spuštění programu	<p data-bbox="643 965 884 999">MCOPS(6,<kanál>)</p>
	<p data-bbox="643 1010 1501 1070">Spouští v naprogramovaném <kanálu> zvolený program nebo zvolenou větu ručního zadávání.</p>

Systémová data

Následek	Syntaxe / popis
Zadání provozního režimu	MCOPS(7,<kanál>,<provozní režim>)
	<p>Zadává v naprogramovaném <kanálu> provozní režim.</p> <ul style="list-style-type: none"> ● <Provozní režim> : Výraz Integer. Určuje provozní režim, na který se má přepnout. <ol style="list-style-type: none"> 1 Krokovací provoz. Osy se dají krokovat (+/-). Viz též MCOPS(4..) v <typu volby>: Identifikační číslo 256. 2 Najetí do referenčního bodu. Osy se dají spouštět pomocí signálů Ruční + / Ruční -. Viz též MCOPS(4..) v <typu volby>: Identifikační číslo 512. 4 Ruční zadávání. Jednotlivé NC-věty mohou být předávány ke zpracování. 5 Automatika (následná věta). Dílčí programy se kompletně zpracovávají. 6 Automatika (programová věta). Jednotlivé věty dílčího programu se zpracovávají jedna po druhé. Každá jednotlivá věta je připravena a spuštěna pomocí NC-Start. 7 Automatika (jednotlivý krok). Z jednotlivé NC-věty v dílčím programu generuje a připravuje NC několik bloků. V tomto provozním režimu se pomocí NC-Start vždy předává jednotlivý blok do interpolátoru ke zpracování. 10 Automatika (jednotlivá věta). Pomocí NC-Start se předávají všechny bloky, které byly generovány a připraveny z jednotlivé NC-věty v dílčím programu, do interpolátoru ke zpracování. 11 Opětné najetí. Je možné ručně odjíždět s osami od kontur a automaticky nebo ručně k nim opět najíždět. 12 CPL-Debugger: Zpracovávají se jednotlivé věty, jak jsou uvedeny v dílčím programu. 13 CPL-Debugger: Zpracovávají se všechny věty až k dalšímu bodu přerušení. 14 Krokovací provoz: Pohyb v souřadnicích obrobku <p>Změna provozního režimu je možná pouze za následujících předpokladů:</p> <ul style="list-style-type: none"> ● signál rozhraní kanálu qCh_OpModePlc (provozní režim SPS) nesmí být nastaven ● v NC není zvolen žádný program nebo věta - nebo - je nutné přepínat výhradně mezi automatickými provozními režimy následná věta, programová věta, jednotlivý krok nebo jednotlivá věta. <p>Příklad: ERR_VAR=MCOPS(7,2,5) Změna provozního režimu v 2. kanálu na automatiku (následná věta).</p>

Systémová data

Následek	Syntaxe / popis
Změna strategie opětovného najetí	MCOPS(8,<kanál>,<jak>,<kam>)
	<p>Určuje v naprogramovaném <kanálu> strategii opětovného najetí.</p> <ul style="list-style-type: none"> ● <Jak> : Výraz Integer. Udává, zda je požadováno <ol style="list-style-type: none"> 1 Automatické opětovné najetí 2 Opětovné najetí s jednotlivou větou, nebo 3 Ruční opětovné najetí. ● <Kam> : Výraz Integer. Udává, jestli se při opětovném najetí najíždí <ol style="list-style-type: none"> 1 do počátečního bodu 2 do koncového bodu, nebo 3 do bodu přerušení.
Ukončení zaznamenávání opětovného najetí	MCOPS(9,<kanál>)
	Ukončuje v naprogramovaném <kanálu> zaznamenávání opětovného najetí. Křakovací pohyby pak již nejsou zaznamenávány.

Systémová data

5.13.3 Příklady programování

Příklad 1: Okamžitě změnit stav SAV a interpolátoru kanálu 2

```

10 DIM BUF%(2)           Vytvořit pole
20 VERSION=0             Okamžitě dodávat data
30 ERR_VAR%=MCOVS(32,2,VERSION,
    BUF%,2)              Vyvolání funkce

```

V BUF%(1) je stav SAV, v BUF%(2) je stav IPO.
Ve VERZI je uvedeno aktuální číslo verze dat
(důležité pro příklad 2).

Příklad 2: Čekání, až se stav SAV kanálu 2 změní na "neaktivní"

<Kód z příkladu 1>

```

:
10 INAKTIV = 1
20 WHILE BUF%(1) <> INAKTIV DO
30   ERR_VAR% = MCOVS(32,2,VERSION,BUF%,2)
40 END

```

Funkce se po vyvolání vrátí zpět do volajícího programu, teprve až se změní stav SAV (VERZE obsahuje ještě hodnotu různou od 0; číslo verze, viz příklad 1). Smyčka bude opuštěna až poté, co BUF%(1) bude obsahovat hodnotu 1.

Příklad 3: Výpis názvů os v okně MSG

```

:
30 VERSION=0
40 DIM AXNAME$(512)
50 REM Požadovat názvy všech os
60 ERR=MCOVS(33,-1,VERSION,AXNAME$,512)
70 IF ERR=0 THEN
80   REM Určit počet os
90   DIM AXNMB%(3)
100  VERSION=0
110  ERR=MCOVS(45,-1,VERSION,AXNMB%,3)
120  ANZ=AXNMB%(2)
130  ENDIF
140 IF ERR<>0 THEN
150  PRN#(0,"Došlo k chybě: ",ERR)
160 ELSE
170  REM zobrazení názvů os
180  FOR I%=0 TO (ANZ-1)
190    NAME$=MID$(AXNAME$,I%*9+1,8)
200    IF ASC(NAME$)<>0 THEN
210      REM Název osy je definován
220      PRN#(0,I%+1,". Název osy: ",NAME$)
230    ENDIF
240  NEXT
250  ENDIF
N260 M30

```

Systémová data

Příklad 4: Kanál / stav čekání

```
10 CHAN%=1
20 VERSION%=0
30 STATES%=0
40 ERR=MCODS(87,CHAN%,VERSION%,STATES%,1)
50 IF ERR=0 THEN
60     MASKE%=1
70     WHILE MASKE% <= STATES% DO
80         CASE (STATES% AND MASKE%) OF
90             LABEL 1:PRN#(0,"Doba prodlevy")
100            LABEL 2:PRN#(0,"Pomocná funkce s povinným potvrzením")
110            LABEL 4:PRN#(0,"Blokování načítání")
120            LABEL 8:PRN#(0,"Posuv v kanálu roven 0")
130            LABEL 16:PRN#(0,"Zastavení programu s M0/M1")
140            LABEL 32:PRN#(0,"Zastavení posuvu v kanálu")
150            LABEL 64:PRN#(0,"Blokování posuvu v kanálu nebo kanálové ose")
160            LABEL 128:PRN#(0,"Blokování načítání zadané zákazníkem")
170            LABEL 256:PRN#(0,"Synchronizované zastavení pohybů
                mezi kanály (ASTOP, ...)")
180            LABEL 512:PRN#(0,"Čekání na osu při změně osy (G511)")
190            LABEL 1024:PRN#(0,"Čekání na permanentní proměnnou (WPV)")
200            LABEL 2048:PRN#(0,"Čekání na signál rozhraní v
                aktivním okamžiku (WAITA, ...)")
210            LABEL 4096:PRN#(0,"Čekání na signál rozhraní (WAIT(BITIF(...))) nebo")
220                PRN#(0," (WAIT(,ZEIT%)) v přípravě vět")
230            LABEL 8192:PRN#(0,"Datová služba Motion Control (MCODS(...))")
240        ENDCASE
250        MASKE%=MASKE%*2
260    END
270    ENDIF
M30
```

Systémová data

Příklad 5: Přirazení osa - kanál

```
10 REM Program se ptá na počet os v systému
15 REM a čísla kanálových os. S těmito
20 REM informace kanálové osy nejprve na
30 REM pozici 0 a potom na pozici <čísla kanálu> 40 REM najedou
50 CHAN%=SD(8) : REM Vlastní číslo kanálu
60 IDCHAX%=43 : REM Funkční typ pro přiřazení
      osa - kanál
70 IDMAXAX%=45 : REM Funkční typ pro počet os
80 DIM BUF%(16) : REM Vyrovnávací pamě pro přiřazení osa - kanál
90 SIZE%=16
100 ANZ%=0 : REM Max. index systémových os
120 REM Určit kanálové osy aktivního kanálu
130 VERSION=0
140 ERR=MCODS(IDCHAX%,CHAN%,VERSION,BUF%,SIZE%)
150 IF ERR=0 THEN
160 VERSION=0
170 ERR=MCODS(IDMAXAX%,CHAN%,VERSION,ANZ%,1)
180 IF ERR=0 THEN
190 FOR I%=1 TO ANZ%
200 IF BUF%(I%) = CHAN% THEN
N210 F1000 [AXP(I%,0,0)]; Kanálové osy na 0
      ; najet
220 ENDIF
230 NEXT
240 FOR I%=1 TO ANZ%
250 IF BUF%(I%) = CHAN% THEN
N260 M0
N270 WAIT
N280 F1000 [AXP(I%,CHAN%,0)]; Kanálové osy na
      ; CHAN% najet
290 ENDIF
300 NEXT
310 ENDIF
330 ENDIF
N310 M30
```

Zpracování znakových řetězců

5.14 Zpracování znakových řetězců

Aby bylo možné zpracovávat v CPL znakové řetězce (stringy), musí být uloženy v jednorozměrném poli (pole: Array) z indikované proměnné Character. Každá proměnná Character v tomto poli je adresována pomocí indexu a může obsahovat přesně 1 znak.

Pro zpracování stringů jsou k dispozici příkazy CPL: MID\$, LEN, INSTR, ASC, STR\$, VAL a TRIM\$.

5.14.1 Dimenzování znakových polí

DIM

Chcete-li vytvořit znakové pole, musíte pomocí instrukce DIM indikovat proměnnou Character.

Tímto způsobem lze vytvářet znaková pole s max. kapacitou 1024 znaků (rozsah hodnot indexu: 1 až 1024).

Pokud nedodržíte rozsah hodnot, objeví se chybové hlášení UNZULÄSSIGE FELDGRENZE.

Příklad:

```
1 DIM VWX$(14)
```

V tomto příkladu je vytvořeno znakové pole `vwx$`, které se skládá ze 14 jednotlivých proměnných Character. Je tedy možné ukládat do `vwx$` stringy o délce do 14 znaků.

Příklady:

- 1 DIM ABC\$(1) Znakové pole pro string o délce max. 1 znak.
- 2 DIM BCDE\$(10) Znakové pole pro string o délce max. 10 znaků.

Zpracování znakových řetězců

5.14.2 Načtení znaků do znakového řetězce

MID\$

Tato funkce vyjímá části z výrazu String.

Výsledek může být předán do dimenzované nebo nedimenzované proměnné Character:

- **Dimenzovaná** proměnná Character dostává celý dílčí string, určený příkazem MID\$.
- **Nedimenzovaná** proměnná Character dostává pouze počáteční adresu a délku příslušného dílčího stringu. Když se změní výraz String, ze kterého byl vyjmut dílčí string, změní se tím odpovídajícím způsobem také nedimenzovaná proměnná Character.

Pokud uvnitř příkazu MID dojde ke zřetězení (např. MID\$(A\$+B\$, 2, 3)), smí být výsledek přiřazen pouze znakovému poli.

MID\$ (<výraz STR>, <počáteční bod>[, <počet znaků>])

<Výraz STR> Výraz String, z něhož mají být vyjmuty části.
 <Počáteční bod> určuje pozici uvnitř znakového pole <výraz STRING>, od které mají být vyjmuty znaky.
 <Počet znaků> určuje počet znaků, které budou vyjmuty. Není-li <počet znaků> naprogramován, budou vyjmuty všechny znaky až do konce znakového pole.

Rozsah hodnot pro 2. a 3. parametr obsahuje hodnoty Integer mezi 1 a 1024. Pokud není rozsah hodnot dodržen, objeví se chybové hlášení "Unzulässiger Parameter".

Při zásahu do části znakového pole, která ještě není obsazena, se vrátí 'NUL'.

Příklad:

```

1 DIM A$(10)
2 DIM B$(5)
3 A$="ABCDEFGH IJ"
4 B$=MID$(A$, 2, 5)
5 C$=MID$(A$, 2, 5)
6 REM   Obě proměnné B$ a C$ mají
        obsah: BCDEF
7 A$="QRSTUVWXYZ"
8 REM   Proměnná B$ má obsah: BCDEF
        Proměnná C$ má obsah: RSTUV
  
```

Zpracování znakových řetězců

5.14.3 Změny znakových řetězců

MID\$

Instrukce MID\$ přepisuje části znakového pole.

MID\$ (<znakové pole>, <počáteční bod>[, <počet znaků>])

<Znakové pole>	Znakové pole, jehož části mají být přepsány.
<Počáteční bod>	určuje pozici uvnitř <znakového pole>, od které mají být přepsány znaky. Hodnota <počátečního bodu> smí překročit počet dosud obsazených složek (délku) maximálně o 1.
<Počet znaků>	určuje počet znaků, které budou přepsány. Není-li <počet znaků> naprogramován, zapíší se do <znakového pole> všechny přiřazené znaky, pokud to dimenzování znakového pole dovolí.

Rozsah hodnot pro 2. a 3. parametr leží mezi 1 a 1024. Pokud není rozsah hodnot dodržen, objeví se chybové hlášení UNZULÄSSIGER PARAMETER.

Příklad:

```
1 DIM A$(10)
2 A$="ABC"           Délka A$ je 3.
3 MID$(A$, 4, 3)="DEF"
```

Bude popsána 4.-6. složka znakového pole. To je dovoleno, protože první tři složky jsou již obsazeny.

Příklad:

```
1 DIM A$(10)
2 A$="ABC"           Délka A$ je 3.
3 MID$(A$, 5, 3)="DEF"
```

Bude učiněn pokus o popsání 5. až 7. složky znakového pole. To však povede k chybovému hlášení ZEICHENFELD NICHT BELEGT, protože 4. složka ještě nebyla obsazena.

Je-li přiřazeno více znaků, než umožňuje maximální délka znakového pole, jsou tyto znaky odmítnuty.

Zpracování znakových řetězců

5.14.4 Délka znakového řetězce

LEN

LEN (<výraz *String*>)

Udává počet znaků <výrazu *STRING*>. Výsledkem je hodnota INTEGER.

Je-li <výraz *String*> prázdný, vrací funkce LEN hodnotu 0.

Pokud <výraz *String*> není definovaný, vrací funkce LEN hodnotu -1.

Příklad:

1 DIM XYZ\$(10)	
2 XYZ\$="ABC"	
3 I%=LEN(XYZ\$)	Proměnná INTEGER I% má hodnotu 3
4 XYZ\$=" "	
5 J%=LEN(XYZ\$)	Proměnná INTEGER J% má hodnotu 0
6 XYZ\$=NUL	
7 K%=LEN(XYZ\$)	Proměnná INTEGER K% má hodnotu -1

5.14.5 Hledání znakového řetězce

INSTR

INSTR (<znakový řetězec>, <výraz *String*>[, <počáteční bod>])

Prohledává <výraz *String*> od <počátečního bodu> s ohledem na <znakový řetězec> a vydává pozici prvního znaku nalezeného <znakového řetězce> ve <výrazu *String*> jako hodnotu INTEGER.

Pokud <znakový řetězec> není nalezen, je vydána hodnota 0.

Tento <znakový řetězec> může být naprogramován jako výraz STRING.

Rozsah hodnot pro 3. parametr leží mezi 1 a 1024. Pokud není rozsah hodnot dodržen, objeví se chybové hlášení UNZULÄSSIGER PARAMETER .

Příklad:

1 DIM A\$(8)	
2 DIM B\$(16)	
3 A\$="A" : MID\$(A\$,2)="UVWXYZ"	
4 B\$="ABCDEF UVWXYZ GH"	
5 POS1%=INSTR(MID\$(A\$,2),B\$,4)	Proměnná INTEGER POS1% má hodnotu 8
6 POS2%=INSTR(MID\$(A\$,2,4),B\$,10)	Proměnná INTEGER POS2% má hodnotu 0
7 POS3%=INSTR(MID\$(A\$,2),B\$)	Proměnná INTEGER POS3% má hodnotu 8

Zpracování znakových řetězců

5.14.6 Znakové řetězce a čísla

ASC

ASC (<znakový řetězec>)

Vydává pořadové číslo prvního znaku (ASCII-kód) ze <znakového řetězce> jako hodnotu INTEGER.

Pokud je <znakový řetězec> prázdný nebo nedefinovaný, vrací funkce ASC hodnotu -1.

<Znakový řetězec> musí být výraz STRING.

ASC je inverzní funkcí k CHR\$.

Příklad:

10 DIM A\$ (1)	
20 A\$ = "ABC"	
30 B\$ = "BCD"	
40 I% = ASC(A\$)	Proměnná INTEGER I% má hodnotu 65
50 J% = ASC(B\$)	Proměnná INTEGER J% má hodnotu 66
60 A\$ = ""	
70 K% = ASC(A\$)	Proměnná INTEGER K% má hodnotu -1
80 A\$ = NUL	
90 L% = ASC(A\$)	Proměnná INTEGER L% má hodnotu -1

CHR\$

CHR\$ je inverzní funkcí k ASC.

CHR\$ (<výraz Integer>)

Konvertuje <výraz Integer> na odpovídající znak ASCII.

V příloze této příručky naleznete v tabulce "Sada znaků ASCII" seznam všech desítkových hodnot znaků ASCII.

Příklad:

10 DIM A\$ (1)	
20 I% = 65	
40 A\$ = CHR\$(I%)	Proměnná STRING A\$ má hodnotu 65

STR\$

STR\$ ([<formátový string>], <hodnota>)

Konvertuje číselný výraz <hodnota> na znakový řetězec, který může být přiřazen výhradně znakovému poli. Přiřazení k proměnné STRING vede k běhové chybě.

Zpracování znakových řetězců

<Hodnota> může být výraz INTEGER nebo REAL s jednoduchou či dvojnásobnou přesností.

Je-li naprogramován <formátový string> může být znakový řetězec vydán zformátovaný. Přitom se číslice zobrazují jako "#" a desetinné tečky jako ".". Bez <formátového stringu> má výstup standardní formát.

Standardní formáty:

číslo INTEGER: 9 míst

číslo REAL s jednoduchou přesností: 4 místa před a 3 za desetinnou tečkou.

číslo REAL s dvojnásobnou přesností: 9 míst před a 6 za desetinnou tečkou.

Příklad:

```
10 DIM A$(50)
```

```
20 DIM B$(21)
```

```
30 A$=STR$( "Zahl=##.###" , (37/3) )
```

Obsah znakového pole A\$:
"Číslo = 12.333"

```
40 B$=STR$(2.5)
```

Obsah znakového pole B\$:
" 2.500"

Zpracování znakových řetězců

VAL**VAL** (<výraz *STRING*>)

Vrací číselnou hodnotu pro <výraz *STRING*>. Jestliže se ve znakovém řetězci vyskytuje jiný znak než vodící mezera, vodící znak '+' nebo '-', číslice 0 až 9 a desetinná tečka '.', je provedena konverze až po tento (jiný) znak. Vodící mezery jsou při vytváření hodnoty stejně jako nevýznamné nuly ignorovány. Pokud se neobjeví žádný z výše uvedených znaků, je vrácen výsledek 'NUL'. Jestliže se ve znakovém řetězci vyskytuje desetinná tečka, smí být výsledek přiřazen pouze proměnné REAL nebo REAL s dvojnásobnou přesností. V takovém případě vede přiřazení k proměnné INTEGER k chybovému hlášení UNZULÄSSIGE ZUWEISUNG

Příklad:

```
1 I% = VAL("1.23DE")
2 K% = VAL("123DE")
3 J% = VAL("ABC")
4 R = VAL("-1.23DE")
5 Z = VAL("+ 000001234TEST4365")
6 X = VAL("ABC1.23DE")
7 D! = VAL("1234567.234567")
```

Řádek 1 vede k chybovému hlášení, protože má proběhnout přiřazení k proměnné INTEGER.

Hodnota proměnné INTEGER K% je 123. Číslice 1, 2, 3 jsou konvertovány na číslo INTEGER. Znak "D" ukončuje konverzi, protože nemůže patřit k číslu INTEGER. Následující znaky nejsou dále zpracovány.

Hodnota proměnné INTEGER J% je NUL, proměnná tedy není obsazena. Znak "A" ukončuje zpracování <výrazu *STRING*>.

Hodnota proměnné REAL R je -1.23. Znak "-" je identifikován jako znaménko pro číslo REAL. Číslice 1, znak "." a číslice 2 a 3 jsou konvertovány na číslo REAL. Znak "D" ukončuje konverzi, protože nemůže patřit k číslu REAL. Znak "E" již není zpracován.

Hodnota proměnné REAL Z je 1234. Znak "+" je identifikován jako znaménko pro číslo REAL. Následující mezery jsou při vytváření hodnoty stejně jako nevýznamné nuly ignorovány. Číslice 1, 2, 3 a 4 jsou konvertovány na číslo REAL. Znak "T" ukončuje konverzi, protože nemůže patřit k číslu REAL. Zbývající znaky nejsou dále zpracovány.

Proměnná REAL X je NUL, tzn. není obsazena. V důsledku identifikace znaku "A" je konverze ukončena.

Hodnota proměnné REAL D! s dvojnásobnou přesností je 1234567.234567.

Zpracování znakových řetězců

5.14.7 Odstraňování vodicích, resp. následujících mezer

TRIM\$

TRIM\$ (<řetězec znaků>)

TRIM\$ (<řetězec znaků>, "L")

TRIM\$ (<řetězec znaků>, "R")

TRIM\$() dává při přiřazení oblasti znakového pole k proměnné STRING nebo ke znakovému poli znakový řetězec bez mezer vepředu (→ index L), resp. vzadu (→ index R).

Funkce TRIM bez indexu odstraňuje mezery vepředu i vzadu.

Pokud uvnitř příkazu TRIM dojde ke zřetězení (např. TRIM\$(A\$+B\$)), smí být výsledek přiřazen výhradně znakovému poli.

Příklad:

```
1 A$ = " ABCDEF "  
2 B$ = TRIM$(A$, "L")  
3 C$ = TRIM$(A$, "R")  
4 D$ = TRIM$(A$)  
5 PRN#(1, ">", A$, "<")  
6 PRN#(1, ">", B$, "<")  
7 PRN#(1, ">", C$, "<")  
8 PRN#(1, ">", D$, "<")
```

vede v souboru s logickým číslem 1, otevřeném pro zápis, k následujícím řádkům:

```
> ABCDEF <  
>ABCDEF <  
> ABCDEF<  
>ABCDEF<
```

Zpracování znakových řetězců

5.14.8 Příklady programování

Proměnné STRING může být přiřazen výraz STRING.

Příklad:

Programování proměnné STRING (bez předchozího dimenzování)

```
1 A$="ABCDE"
2 B$=CHR$(10)
```

Při čtecím přístupu je možné pomocí příkazu MID\$ získat přístup k částem proměnné STRING:

```
1 A$="ABCDEFGHIJKLMN"
2 B$=MID$(A$,2,1)
3 C$=MID$(A$,4,4)
```

Následující naprogramování vede k chybám:

```
4 MID$(A$,1,4)="ABCD"
4 A$=MID$(A$,1,3) + MID$(A$,4,1)
4 A$=B$ + A$
```

Při dalším zpracování dimenzovaného znakového pole je nutné cíleně zasáhnout do jednoho nebo několika souvisejících znaků. Jedině tak je možné přiřadit znakové pole nebo část znakového pole k proměnné STRING či k jinému znakovému poli.

Čtecí a zápisový přístup k části znakového pole lze získat pomocí příkazu MID\$. Je-li uveden pouze název znakového pole, používá se celé znakové pole.

Čtení znakového pole

Chcete-li se dostat k <n>-tému znaku znakového pole, postupujte následujícím způsobem (n je menší nebo rovno délce znakového pole a počtu znaků v poli).

Příklad: Čtení znakového pole

```
1 DIM VWX$(13)
2 VWX$="TEST TEST TES"
3 A$ = MID$(VWX$,12,1)
4 I%=12
5 A$=MID$(VWX$,I%,1)
12. znak ("E") znakového pole VWX$ je přiřazen proměnné STRING A$.
```

Zápis do znakového pole

Chcete-li předat obsah proměnné STRING do znakového pole nebo části znakového pole, musíte zaměnit přiřazení.

Příklad: Zápis do části znakového pole

```
1 DIM XYZ$(15)
2 B$="ABCDE"
3 MID$(XYZ$,1,5)=B$
4 MID$(XYZ$,6,5)=B$
1. až 10. znak znakového pole XYZ$ jsou obsazeny obsahem proměnné STRING B$.
```

Zpracování znakových řetězců

Následující naprogramování by vedlo k chybovému hlášení ZEICHEN-FELD NICHT BELEGT, protože znaky 1 až 5 znakového pole ještě nejsou obsazené:

```
1 DIM XYZ$(15)
2 B$="ABCDE"
4 MID$(XYZ$,6,5)=B$
```

Příklad: Zápis do části znakového pole

```
1 DIM XYZ$(100)
2 B$="ABCDE"
3 MID$(XYZ$,1,10)=B$
```

Obsah proměnné STRING B\$: "ABCDE"
Obsah proměnné pole XYZ\$: "ABCDE". Proměnná pole má délku 5.
Zbývajících 95 znaků není obsazeno.

Pokud je délka proměnné STRING menší než znakové pole, popíše se znakové pole XYZ\$ pouze v délce proměnné STRING. Při přiřazení tohoto znakového pole k proměnné STRING se nepřihadí celé znakové pole, které bylo určeno instrukcí DIM, ale pouze oblast, která byla předem popsána (→ délka znakového pole).

Příklad:

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,10)="ABCDE"
4 MID$(XYZ$,6,3)="T"
```

Po větě 3 je obsah proměnné pole XYZ\$: "ABCDE".
Proměnná pole má délku 5. Zbývajících 95 znaků není obsazeno a nezapočítávají se proto do délky.
Po větě 4 je obsah proměnné pole XYZ\$: "ABCDET".
Proměnná pole má nyní délku 6. Zbývajících 94 znaků není obsazeno a nezapočítávají se proto do délky.

Příklad: Přepsání znakového pole

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,10)="1234567890"
4 MID$(XYZ$,3,3)="T"
```

Po větě 3 je obsah proměnné pole XYZ\$ "1234567890".
Proměnná pole má délku 10.
Po větě 4 má proměnná pole XYZ\$ obsah "12T4567890".
Proměnná pole má délku 10. Znak "3" je přepsán znakem "T". Znaky "4" a "5" zůstávají zachovány.

Příklad: Zakázaný přístup do znakového pole

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,6)="ABCDEF"
5 MID$(XYZ$,9,5)="TESTE"
```

Zpracování znakových řetězců

Po větě 3 má proměnná pole XYZ\$ obsah "ABCDEF".

Proměnná pole má délku 6.

Po větě 5 je učiněn pokus přiřadit konstantu do 9. až 13. složky znakového pole. To však povede k chybovému hlášení ZEICHENFELD NICHT BELEGT, protože 7. a 8. složka nejsou obsazeny.

Chcete-li získat přístup k celému znakovému poli, stačí uvést název proměnné.

Zpracování znakových řetězců

7 MID\$(XYZ\$,5,12)=B\$

OBSAH ZNAKOVÉHO POLE

1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16												

D	A	S		I	S	T		E	I	N		T	E
S	T												

Délka = 16; Oblast je zcela obsazená

8 MID\$(XYZ\$,9,8)=C\$

OBSAH ZNAKOVÉHO POLE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

D	A	S		I	S	T		E	I	N		E	I	S	T
---	---	---	--	---	---	---	--	---	---	---	--	---	---	---	---

Délka = 16; Oblast je zcela obsazená

9 XYZ\$=MID\$(XYZ\$,1,4)

OBSAH ZNAKOVÉHO POLE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

D	A	S													
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--

Délka = 4

Oblast není obsazená

Následuje ořezání znakového pole.

Zpracování znakových řetězců

5.14.10 Porovnávání výrazů STRING

Je také možné vzájemně porovnávat výrazy STRING, tedy konstanty STRING, proměnné STRING a znaková pole.

Příklad:

```
1 DIM A$(10)
2 DIM B$(15)
3 A$="ANTON"
4 B$="WILLI"
5 C$="ABCDE"
6 D$="VWXYZ"
7 IF A$ < B$ THEN ...
8 IF MID$(A$,2,3) = MID$(B$,1,3) THEN ...
9 Z?=A$ <> "TESTE"
10 IF "A" <= "C" THEN ...
11 IF C$ > D$ THEN ...
12 IF A$ = C$ THEN ...
13 IF "TE" < MID$(D$,2,2) THEN ...
```

Obsahy výrazů STRING jsou zkontrolovány s ohledem na abecední pořadí podle kódů ASCII jednotlivých znaků.

5.14.11 Zřetězení výrazů STRING

Pomocí znaku '+' je možné provést zřetězení několika výrazů STRING. Výsledek musí být přiřazen znakovému poli. Hloubka vnoření při řetězení výrazů STRING je 3. Při jejím překročení se objeví chybové hlášení LAUFZEITFEHLER 2153 - SCHACHTELUNG ZU TIEF.

Příklad: Zřetězení také uvnitř příkazů CPL

```
1 DIM A$(3)
2 DIM B$(3)
3 A$ = "ABC"
4 B$ = "DEF"
5 C$ = "GH"
6 D$ = "JKL"
7 OPENW(1,"P2",130,"TEST VERKETTUNG",10)
8 PRN#(1,A$+B$)
9 PRN#(1,A$+C$)
10 PRN#(1,C$+D$)
11 PRN#(1,A$+C$+"TEST")
12 PRN#(1,"UVW"+"XYZ")
13 CLOSE(1)
```

Obsah souboru P2:

```
ABCDEF <LF>
ABCGH <LF>
GHJKL <LF>
ABCGHTEST <LF>
UVWXYZ <LF><ETX><LF>
```

Příklad: Zřetězení textů pomocí výrazů STRING

Zpracování znakových řetězců

```

10 DIM A$(100)
20 DIM B$(100)
30 DIM C$(10)
40 DIM D$(20)
51 DIM E$(30)
52 DIM F$(30)
53 DIM G$(30)
54 DIM H$(30)
55 DIM I$(30)
60 A$="DAS "
70 B$="IST EIN TEST"
80 MID$(C$,1,6)=A$ + B$
90 MID$(D$,1,10)=MID$(A$,1,1) + MID$(B$,1,2)
92 E$=A$ + MID$(B$,1)
93 X$="ABC"
94 Y$="DE"
95 F$=X$ + Y$
96 G$=X$ + A$
97 H$=X$ + A$ + "TEST"
98 I$="TES" + "T1"

```

Obsah A\$:	"DAS "	délka 4
Obsah B\$:	"IST EIN TEST"	délka 12
Obsah C\$:	"DAS IS"	délka 6
Obsah D\$:	"DIS"	délka 3
Obsah E\$:	"DAS IST EIN TEST"	délka 16
Obsah F\$:	"ABCDE"	délka 5
Obsah G\$:	"ABCDAS "	délka 7
Obsah H\$:	"ABCDAS TEST"	délka 11
Obsah I\$:	"TEST1"	délka 5

Následující naprogramování vede k chybám:

```

1 DIM A$(3):A$ = "ABC":B$ = "CD":C$ = "EF"
2 D$ = A$ + B$
3 D$ = B$ + C$
4 D$ = A$ + B$ + "TEST"
5 D$ = "TEST" + "TEST1"

```

} nedovolené přiřazení k nedimenzované proměnné STRING

Příklad: STR\$

```

1 DIM A$(50) : DIM B$(21)
2 A$ = STR$("A$ = ##.###", (37/3)) : B$ = STR$(2.5)

```

Obsah znakového pole A\$: "A\$ = 12.333"; znakového pole B\$: "2.500"

Příklad:VAL

```

1 DIM FOLGE$(20) : FOLGE$="X-WERT -0001.234 MM"
2 XR=VAL(MID$(FOLGE$,7)) : Z%=VAL(MID$(FOLGE$,7,6))
3 Y%=VAL(MID$(FOLGE$,15,5)) : X%=VAL(MID$(FOLGE$,18))

```

Obsah proměnné REAL XR:	-1.234
Obsah proměnné INTEGER Z%:	-1
Obsah proměnné INTEGER Y%:	34
Obsah proměnné INTEGER X%:	NUL

Zpracování znakových řetězců

Příklad: LEN

```
1 DIM Z$(10)
2 Z$ = "TEST"
3 S$ = "TEST"
4 A% = LEN("TEST")
5 B% = LEN(Z$)
6 C% = LEN(S$)
7 D% = LEN("TEST"+Z$+S$)
```

Obsah proměnné INTEGER A% : 4
Obsah proměnné INTEGER B% : 4
Obsah proměnné INTEGER C% : 4
Obsah proměnné INTEGER D% : 12

Příklad: Příkaz MID\$ se čtecím přístupem

```
10 DIM A$(4)
20 DIM B$(10)
30 DIM C$(10)
40 DIM D$(10)
50 DIM E$(10)
55 DIM F$(10)
60 A$ = "ABCD"

70 B$ = MID$(A$,2,2)    --> B$ = "BC"
80 C$ = MID$(A$,2,5)    --> C$ = "BCD"
95 E$ = MID$(A$,5,1)    --> E$ = NUL
97 F$ = MID$(A$,2)      --> F$ = "BCD"
98 F$ = MID$(F$,1,1)    --> F$ = "B"
```

Příklad: Příkaz MID\$ se zápisovým přístupem

```
10 DIM A$(4)
20 DIM B$(10)
30 DIM C$(10)
40 DIM D$(10)
60 A$ = "ABCD"
70 B$ = "1234567890"
80 C$ = "EFGHIJKLMN"
85 D$ = A$              --> D$ = "ABCD"
90 MID$(D$,2,3) = B$    --> D$ = "A123"
95 MID$(D$,5,1) = C$    --> D$ = "A123E"
97 MID$(D$,4) = B$      --> D$ = "A121234567"
```

Příklad:TRIM\$

```
1 DIM XYZ$(16)
2 XYZ$ = "XWERT = 0.123 "
3 A$ = MID$(XYZ$,8)
4 B$ = TRIM$(MID$(XYZ$,8))
5 C$ = TRIM$(MID$(XYZ$,8),"L")
6 D$ = TRIM$(MID$(XYZ$,8),"R")
```

Obsah proměnné STRING A\$: " 0.123 "
Obsah proměnné STRING B\$: "0.123"
Obsah proměnné STRING C\$: "0.123 "
Obsah proměnné STRING D\$: " 0.123"

Zpracování souborů

5.15 Zpracování souborů

Soubory jsou kontejnery, které obsahují data. Data mohou být v průběhu programu CPL načítána ze souborů nebo ukládána do souborů.

Tak je možné např. naměřené hodnoty nejprve uložit a později zobrazit nebo vytisknout na tiskárně.

Soubory v systému souborů jsou interně spravovány v CNC. Jsou uloženy organizované v hierarchické struktuře. K jednotlivým souborům se lze dostat pomocí cest.


Soubory mohou být spravovány v různých provozních režimech.

Abyste získali čtecí nebo zápisový přístup k datům, musíte vždy nejprve otevřít příslušný soubor (viz příkazy OPENW, OPENR); když už přístup k datům nepotřebujete, soubor opět zavřete (viz příkaz CLOSE).

5.15.1 Názvy souborů

U názvů souborů musíte dodržovat následující konvence:

- Maximální délka 30 znaků. Nerozlišuje se mezi názvem souboru a případnou příponou názvu. Přípustná jsou všechna písmena, číslice a speciální znaky '.' a '_'.
- Názvy souborů dílčích programů smí mít délku max. 28 znaků, protože CNC při spojování vytváří soubor, jehož název se skládá z původního názvu souboru plus 2 dalších znaků.

 **Speciální znak '\$' se používá jen pro interně vytvořené soubory. Názvy souborů přicházejících zvenčí (přes uživatelské rozhraní, DNC) nesmí obsahovat znak '\$'.**

- Rozlišují se velká a malá písmena.

Příklady: Názvy souborů

P123456789.PRG

P12_Daten_Dial

P12_DATEN_DIAL

- Názvy souborů '.' a '..' nejsou dovolené, protože jsou již používány interně.
- Názvy souborů musí být v rámci adresáře jednoznačné. V různých adresářích ale mohou existovat soubory se stejným názvem.

Zpracování souborů

5.15.2 Struktura sekvenčních souborů

Sekvenční soubor obsahuje posloupnost **složek** (záznamů), které mohou mít **variabilní délku**. Při hledání určitého záznamu v sekvenčním souboru musí být soubor prohledán od začátku s ohledem na tento záznam. Přímý přístup není možný. Při změně délky záznamu v sekvenčním souboru se musí všechny následující záznamy posunout.

Na rozdíl od souborů Random mají záznamy v sekvenčních souborech různou délku (max. délka 1024 znaků). Konec záznamu je označen symbolem <LF>, který se nezapočítává do délky. Po posledním záznamu v souboru je vloženo <ETX><LF>, což představuje pointer EOF. Pointer EOF je ukazatel na konec využitelných dat (<ETX>) v souboru.

5.15.3 Struktura souborů Random

Soubor Random obsahuje **složky** (záznamy) s **pevnou, definovatelnou délkou**. Proto je možný přímý přístup k libovolné složce souboru. Rozdělení souboru Random na záznamy pevné délky umožňuje přímý přístup k určitému záznamu. Data v sekvenčních souborech jsou uložena jako znaky ASCII. To umožňuje obvyklý přístup s použitím editoru stejně jako načítání a čtení souborů Random.

Soubor Random má tu výhodu, že je možné se rychleji dostat k potřebným datům. Kromě toho lze zpracovávat či měnit data záznamu, aniž by se změnila struktura zbytku souboru. Záznamy, které nejsou zcela zaplněny daty, jsou doplněny až po definovanou délku prázdnými znaky (→ mezery).

Jestliže se pokusíte vložit do souboru Random proměnnou STRING, jejíž délka je větší než délka záznamu, naplní se záznam až po definovanou délku prvními znaky proměnné STRING a zbývající znaky jsou odmítnuty.

Při čtení souboru je konec souboru rozpoznán podle symbolu EOF.

Instrukce REWRITE a CLOSE se používají stejně jako u sekvenčních souborů.

Rovněž je možný sekvenční přístup do souboru Random.

5.15.4 Otevření souboru

Aby mohl CPL-program zasahovat do souboru s příkazy zpracování souboru, musí být tento soubor pro CPL-program nejprve otevřený. K tomu slouží následující příkazy.

Zpracování souborů

OPENW, OPENR

Příkaz pro otevření souboru je závislý na požadovaném typu přístupu:

- zápisový přístup: OPENW
- čtecí přístup: OPENR

Pokud otevíraný soubor ještě neexistuje, vytvoří se při otevírání a rezervuje si stanovené místo v paměti.

Je možné otevřít libovolné soubory, které jsou již otevřeny pro zápis, pomocí příkazu OPENR navíc také pro čtecí přístup. Otevřený soubor se však nedá znovu otevřít pro zápis.

Při otevření souboru Random se zavádí další parametr, který udává délku záznamů souboru v bajtech (1 bajt = délka jednoho znaku). Jinak odpovídá struktura příkazů sekvenčním souborům.

Po použití příkazu OPENR je ukazatel souboru na prvním záznamu, ke kterému lze získat čtecí přístup.

Po použití příkazu OPENW je ukazatel souboru na pointeru EOF, tedy za posledním záznamem souboru.

Syntaxe:

OPENW (<n>, <název programu>, <délka>[, <komentář programu>]
[, <délka záznamu>])

OPENR (<n>, <název programu>[, <délka záznamu>])

<n>: Logické číslo, pod kterým je možné volat soubor. Je možné používat hodnoty od 1 do 9. Logické číslo musí být naprogramováno jako výraz INTEGER. Logické číslo nesmí být zadáno současně pro čtení i zápis do souboru. Je tedy možné otevřít nejvýš 9 souborů zároveň.

Pokud nedodržíte rozsah hodnot, objeví se chybové hlášení UNZULÄSSIGE LOGISCHE NUMMER.

<Název programu>: Musí být naprogramován jako výraz STRING. String musí obsahovat přinejmenším název souboru (max. 30 znaků včetně přípony názvu souboru). Je dovoleno zadání názvu souboru včetně úplné nastavené cesty.

<Délka>: Rezervovaná délka při vytvoření souboru v bajtech. Je zapotřebí minimální délka 130 bajtů, protože při zápisu do souboru je vytvořen a uložen nejméně 1 záznam (= 130 znaků). Při nedodržení se objeví chybové hlášení DATEILÄNGE UNZULÄSSIG.

<Komentář programu>: Pro programování parametru komentáře programu je povolen pouze výraz STRING.

<Délka záznamu>: Počet bajtů záznamu; rozsah hodnot: 1..1024. Jestliže nedodržíte rozsah hodnot, objeví se chybové hlášení UNZULÄSSIGE KOMPONENTENLÄNGE

Zpracování souborů

Příklady:

```
50 OPENW(1,"P500",1024,"Dies ist mein schoenstes  
Programm")
```

```
40 A$="P500" : B1$="Dies ist mein schönstes Programm"  
50 OPENW(9,A$,1024,B1$)
```

```
50 OPENW(7,"PDaten_Mes.DAT",1024,"Messdaten ablegen")
```

Při otevření souboru pro zápis se kontroluje, zda je ještě zachována struktura Random.

Pokud byla struktura zničena editorem, objeví se chybové hlášení UNZULÄSSIGE KOMPONENTENLÄNGE.

Příklad:

```
10 OPENW(2,"P200",1024,10)  
20 FOR I% = 1 TO 3  
30 PRN#(2,"TESTE")  
40 NEXT I%  
50 CLOSE(2)
```

Výsledek: "P2"

TESTE <LF>

TESTE <LF>

TESTE <LF>

<ETX><LF>

Při otevření souboru pro čtení se kontroluje, zda je ještě zachována struktura Random. Všechny složky musí mít stejnou délku uvedenou v příkazu OPENR.

Příklad:

```
1 OPENW(2,"P200",130,"TEST",10)  
2 PRN#(2,"ABC")  
3 CLOSE(2)  
4 OPENR(1,"P2",5)  
5 CLOSE(1)
```

Kontroluje se, jestli délka záznamu souboru "P2" je 5. Délka záznamu tohoto souboru je však 10.

Zpracování souborů

Příklad:

```
P1:
N10 G1F10000X1000Y1000Z1000
1 A$="01234567890123456789"
2 B$="TEST"
N20 X0
M30
```

P2:

```
1 OPENW(1,"PMess_PRG",500,"RANDOMDATEI",10)
2 OPENR(2,"P1")
3 DIM A$(30)
4 FOR I% = 1 TO 5
5 INP$(2,A$)
6 PRN$(1,A$)
7 NEXT
8 CLOSE(2)
9 CLOSE(1)
```

VÝSLEDEK: PMess_PRG:

```
N10 G1F100<LF>
1 A$="0123<LF>
2 B$="TEST<LF>
N20 X0 <LF>
M30 <LF>
<ETX><LF><LF>
```

Pokud byla struktura zničena editorem, objeví se chybové hlášení UNZULÄSSIGE KOMPONENTENLÄNGE.

FILENO

Pro přístup do souborů jsou zapotřebí logická čísla souborů, která smí nabývat hodnot mezi 1 a 9, tzn. s použitím CPL je možné otevřít maximálně 9 souborů současně.

Příkaz FILENO umožňuje zeptat se na další platné logické číslo souboru. Pokud funkce vrátí hodnotu "-1", není k dispozici žádné volné logické číslo souboru. Funkční hodnota je typu Integer.

Příklad:

```
10 LOG_NR%= FILENO
20 IF (LOG_NR% <> (-1)) THEN
30 OPENW(LOG_NR%,"/user/usr/Test",130)
40 PRN$(LOG_NR%,"Datum: ",DATE)
50 CLOSE(LOG_NR%)
60 ELSE
70 PRN$(0,"Keine freie log. Datei-Nummer vorhanden!")
80 ENDIF
M30
```

Zpracování souborů

VERSINF\$

S použitím funkce VERSINF\$ je možné dotazovat se v CPL-programu na administrativní data IndraMotion MTX.

Funkce dává hodnotu typu String.

Syntaxe:

VERSINF\$ (<index1>[,<index2>])

<Index1>: Typ konstanta Integer.

Funkční režim:

1: Čtení softwarové verze

2: Čtení hardwarové verze.

<Index2>: Typ Integer.

Přídavný režim funkce, závislý na hodnotě parametru <Index1>.

<Index1> = 1: Není k dispozici žádný přídavný režim

<Index1> = 2: Přídavný režim:

1: Typ hardwaru (standardní)

2: Č. skupin zapojení

3: Č. indexu skupin zapojení

Příklad:

```
10 DIM SYS_INFO$(50)
20 SYS_INFO$ = VERSINF$(1)
30 PRN#(0,"Software-Version: ", SYS_INFO$)
M30
```

DIRINF

S použitím funkce DIRINF je možné dotazovat se v CPL-programu na administrativní data systému souborů IndraMotion MTX. Funkce dává hodnotu typu Integer (při záporných výsledcích je reálný výsledek větší než maximální hodnota Integer, tzn. > 2.147.483.647).

Syntaxe se liší v závislosti na hodnotě parametru <index1>.

Syntaxe pro <index1> = 1, 2 nebo 3:

DIRINF (<index1>[,<index2>])

Syntaxe pro <index1> = 4:

DIRINF (4, [<index2>], <výsledná proměnná>[, <číslo souboru>])

<Index1>: Typ konstanta Integer. Funkční režim:

1: Volné paměové místo

2: Obsazené paměové místo

3: Počet existujících souborů v adresáři

4: Název souboru v adresáři

<Index2>: Typ String. Přídavný režim:

Název adresáře (standardní: aktuální adresář)

Zpracování souborů

<Výsledná proměnná>: Typ dimenzovaná proměnná String, pouze pro funkční režim 4.

Dává po provedení příkazu název souboru v uvedeném adresáři.

<Číslo souboru>: Typ Integer, pouze pro funkční režim 4.

V <čísle souboru> je uvedeno pořadové číslo souboru v adresáři, jehož název má být vydán ve <výsledné proměnné>.

Při nepřipustném pořadovém čísle je ve <výsledné proměnné> vydána hodnota NUL.

Vrácená hodnota:

Funkční režim 1:	Volné paměové místo v bajtech
Funkční režim 2:	Obsazené paměové místo v bajtech
Funkční režim 3:	Počet existujících souborů v adresáři
Funkční režim 4:	0 (<výsledná proměnná> obsahuje název souboru) nebo 1 (<výsledná proměnná> je NUL)

Příklad:

```
; Zobrazit názvy všech souborů z adresáře "/database" v
; okně MSG
10 DIM FILENAME$(30)
20 LJUST
30 DIR$= "/database"
40 ANZ_FILES%= DIRINF(3,DIR$)
50 FOR LNR%= 1 TO ANZ_FILES%
60  ERG%= DIRINF(4,DIR$,FILENAME$,LNR%)
70  PRN#(0,LNR%," : ",FILENAME$)
80 NEXT LNR%
M30
```

DIRCR

S použitím příkazu DIRCR je možné vytvořit v CPL-programu nový adresář. Funkce dává hodnotu typu Integer.

Syntaxe:

DIRCR(<adresář>)

<Adresář>: Název adresáře s úplnou cestou jako výraz String. Bez udání cesty je výrazem String nastaven aktuální adresář.

Vrácená hodnota:

- 0: Adresář byl vytvořen
- 1: Nelze vytvořit adresář

Zpracování souborů

Příklad:

```
10 I% = DIRCR("/usr/user/test")  
M30
```

DIRDEL

S použitím příkazu DIRDEL je možné vymazat v CPL-programu prázdný adresář. Funkce dává hodnotu typu Integer.

Syntaxe:

DIRDEL (<adresář>)

<Adresář>: Název adresáře s úplnou cestou jako výraz String.

Vrácená hodnota:

0: Adresář byl vymazán

1: Nelze vymazat adresář

Příklad:

```
10 I% = DIRDEL("/usr/user/test")  
M30
```

Zpracování souborů

5.15.5 Zapisování do souboru

LJUST, NJUST

Příkazem LJUST (= Left JUSTify) se přepíná na datový výstup zarovnaný doleva. To platí pro všechny výstupy ze souboru až do konce průběhu programu. Příkazem NJUST (No JUSTify) je možné předčasně přepnout zpět na formátovaný výstup.

Při výstupu dat ze souborů je k dispozici pro datový typ REAL maximálně 7 míst (4 před desetinnou čárkou a 3 za ní) a pro datový typ INTEGER maximálně 9 míst. Nevýznamné nuly před a za hodnotou jsou přitom potlačeny. To platí i pro výstup zarovnaný doleva.

S použitím příkazu LJUST je možné pomocí CPL přímo vytvářet NC-programy, které mohou být uvedeny v provozním režimu ABARBEITEN (Zpracování), protože jsou potlačeny mezery mezi NC-adresou a hodnotou.

PRN#

Syntaxe:

PRN# (<n>,[<výraz>][,<výraz>][,<výraz>][,...];:)

- <n>: 1 až 9: Logické číslo souboru, do kterého chcete zapisovat.
0: Výstup je přenesen na obrazovku (jako při odkazovém programování příkazem MSG).
- <Výraz>: Libovolné alfanumerické znaky (text v uvozovkách), formátové řetězce nebo proměnné, jejichž obsah má být uložen, resp. zobrazen.
- ;
Potlačuje automatické připojování <CR><LF>. Pokud je příkazem PRN# přepsán záznam, platí:
Příkaz PRN# se středníkem:
Pokud je délka nově zapisovaných dat menší než délka starých dat, jsou vložena nová data a zbytek starých dat zůstane zachován.
Příkaz PRN# bez středníku:
Pokud je délka nově zapisovaných dat menší než délka starých dat, jsou vložena nová data a zbytek starých dat je přepsán mezerami.

Typ proměnné je volně volitelný. Je možné používat také indikované proměnné a znaková pole. Rovněž je možné programovat výrazy REAL s dvojnásobnou přesností jako libovolné výrazy CPL.

Zpracování souborů

Má-li být výsledek výrazu vydán s uvedením formátu, musí být alespoň jeden z výrazů typu STRING. V tomto formátovém řetězci můžete pomocí znaků '#' a '.' předepsat formát. Výsledky se zapíší na místo formátové instrukce zadané znakem '#'. Přitom se první formátová instrukce obsažená ve výrazu STRING vztahuje k prvnímu následujícímu výrazu, který může být vydán s uvedením formátu. Výrazy BOOL nemohou být formátovány. Počet všech naprogramovaných formátových údajů musí být menší nebo roven počtu vydávaných výrazů. Pokud tato podmínka není splněna, zobrazí se přebytečné znaky '#'. Bez uvedení formátu je výraz vydán ve standardním formátu.

Je-li výstup výrazu větší než 1024 znaků, objeví se chybové hlášení SATZ GRÖßER ALS 1024 BYTE.

Pokud se výsledek nedá zobrazit v uvedeném formátu, je vydáno varování PRN-FORMAT FALSCH a místo chybného formátu jsou vydány znaky "**".

Jsou-li znaky # vytvořeny v souboru, nesmí za stringem následovat žádný formátovatelný výraz v rámci instrukce PRN#.

Výstup znaku # se dá provést také pomocí funkce CHR\$(35).

S použitím CHR\$(13) je možné zajistit během výstupu posuv řádků, tzn. další výstup příkazu PRN# pokračuje na dalším řádku, tedy v příštím záznamu.

S použitím funkce CHR\$() je možné, např. při výstupu přes sériové rozhraní, předávat také jiné řídicí znaky.

Příklad:Příkaz PRN# se středníkem

```
1 OPENW(2,"PProg123.PRG",200,35)
2 PRN#(2,"TEST1 FUER PRN-BEFEHL MIT SEMIKOLON")
3 PRN#(2,"TEST2 FUER PRN-BEFEHL MIT SEMIKOLON")
4 PRN#(2,"TEST3 FUER PRN-BEFEHL MIT SEMIKOLON")
6 SEEK(2,1)
7 PRN#(2,"UEBERSCHREIBEN";)
8 CLOSE(2)
```

VÝSLEDEK v PProg123.PRG:

```
UEBERSCHREIBEN-BEFEHL MIT SEMIKOLON<LF>
TEST2 FUER PRN-BEFEHL MIT SEMIKOLON<LF>
TEST3 FUER PRN-BEFEHL MIT SEMIKOLON<LF>
<ETX> <LF>
```

Zpracování souborů

Příklad:Příkaz PRN# bez středníku

```

1 OPENW(2,"P2",1000,36)
1 REWRITE(2)
2 PRN#(2,"TEST1 FUER PRN-BEFEHL OHNE SEMIKOLON")
3 PRN#(2,"TEST2 FUER PRN-BEFEHL OHNE SEMIKOLON")
4 PRN#(2,"TEST3 FUER PRN-BEFEHL OHNE SEMIKOLON")
6 SEEK(2,1)
7 PRN#(2,"UEBERSCHREIBEN")
8 CLOSE(2)

```

VÝSLEDEK v P2:

```

UEBERSCHREIBEN <LF>
TEST2 FUER PRN-BEFEHL OHNE SEMIKOLON<LF>
TEST3 FUER PRN-BEFEHL OHNE SEMIKOLON<LF>
<ETX><LF>

```

Za posledním záznamem souboru je vložen symbol <ETX><LF>.

Jestliže délka záznamu překročí 1024 znaků, objeví se chybové hlášení SATZ GRÖßER ALS 1024 BYTE .

Je-li při zápisu do sekvenčního souboru dosažen konec souboru, je soubor automaticky okopírován a rezervovaná oblast se zvětší o obsazenou délku, pokud je k dispozici dostatek místa v paměti dílčího programu.

Protože se přitom velmi rychle spotřebovává velké množství paměťového místa, doporučujeme při vytvoření souboru příkazem OPENW rezervovat dostatečně velkou délku souboru.

Příklad:

```

1 OPENW(1,"P2",300,"TEST PRN-BEFEHL")
2 A$="TEST"
3 B$="FUER"
4 C$="PRN-BEFEHL"
5 PRN#(1,A$)
6 PRN#(1,B$)
7 PRN#(1,C$)
8 PRN#(1,A$;)
9 PRN#(1,B$;)
10 PRN#(1,C$;)
11 CLOSE(1)

```

Výsledek:

```

P2:
TEST<LF>
FUER<LF>
PRN-BEFEHL<LF>
TESTFUERPRN-BEFEHL<LF><ETX><LF>

```

Zpracování souborů

Příklad:

```
10 DIM E$(50)
20 OPENW(1,"P2",300,"TEST2")
30 A% = 5000
40 R = 1.231
50 B! = 4/3
60 D$ = "ABCDE"
70 E$ = "CDEFGHI"
80 PRN#(1,"10";)
90 PRN#(1,"#####", "###.###", "#.#####", A%, R, B!, D$, E$)
95 CLOSE(1)
```

P2:

```
10 5000 1.2311.33333ABCDECDEFghi <LF>
    |-----|-----|-----|-----|
    A%      R      B!      D$      E$
```

REWRITE

Pokud jsou v otevřeném souboru již k dispozici data, připojují se nová data při zapisování za normálních okolností k již existujícím. Existující soubor ale může být přepsán příkazem REWRITE, aniž by bylo předem nutné zvlášť vymazat již nepotřebný obsah. Při přepisování zůstává v paměti dílčího programu k dispozici oblast rezervovaná v příkazu OPENW.

Syntaxe:

REWRITE (<n>)

<n>: Logické číslo souboru (rozsah hodnot 1 ... 9)

Před přepsáním souboru musí být tento soubor otevřený.

Zpracování souborů

5.15.6 Čtení souboru

INP#

Pomocí instrukce INP# můžete číst data ASCII z otevřeného souboru ve formě záznamů a přiřadit je jedné nebo několika proměnným. Příkaz účinkuje pouze na soubory, které byly otevřeny příkazem "OPENR(..)".

INP# (<n> , <proměnná>[,<proměnná>] [, ...] [;])

<n>: 1 bis 9: Logické číslo souboru, ze kterého chcete číst.

<Proměnná>: Proměnná, do které budou uložena přečtená data.

; Je-li naprogramován středník, zůstává ukazatel souboru v záznamu tak dlouho, dokud není dosažen konec záznamu. Potom následuje přepnutí na další záznam. Tam však nepokračuje automaticky čtení. Jestliže není naprogramován středník, proběhne automatické přepnutí na další záznam.

Typ proměnné je volně volitelný. Je možné používat také indikované proměnné a znaková pole. Pokud je logické proměnné přiřazena jiná hodnota než TRUE či FALSE, je tato proměnná obsazena hodnotou NUL.

U proměnných typu INTEGER nebo REAL (s jednoduchou či dvojnásobnou přesností) jsou znaky '0' .. '9', znaménka '-', '+' a nevýznamné nuly nebo mezery převáděny na hodnoty INTEGER nebo REAL. Pokud je proměnné INTEGER nebo REAL přiřazen jiný znak, je tato proměnná obsazena hodnotou NUL. Je-li proměnná obsazena hodnotou NUL, nemění se její pozice uvnitř souboru.

Pokud je proměnné INTEGER nebo REAL přiřazena příliš velká hodnota, objeví se odpovídající chybové hlášení:

UNZULÄSSIGER INTEGER-WERT

UNZULÄSSIGER FLOAT-WERT

Příklad: Instrukce INP#

P2:
ABC 123456789 ABC

P3:
1 OPENR(2, "P2")
2 DIM C\$(3)
3 DIM D\$(3)
4 INP#(2, I%, J, L?, C\$, K%, D\$)
5 CLOSE(2)

VÝSLEDEK:
I% = NUL
J = NUL
L? = NUL
C\$ = "ABC"
K% = 123456789
D\$ = "ABC"

Zpracování souborů

Příklad: Čtení záznamu ze souboru

```
1 OPENW(1,"P2",200,"Test",22)
2 PRN#(1,"-12TEST1.23V12ABCD2.4A")
3 PRN#(1,"-12TEST1.23V12ABCD2.4A")
4 PRN#(1,"-12TEST1.23V12ABCD2.4A")
5 CLOSE(1)
6 DIM A$(3)
7 DIM C$(5)
8 DIM D$(4)
9 DIM E$(4)
10 DIM G$(25)
11 DIM H$(7)
12 DIM I$(7)
13 DIM J$(25)
14 DIM R(1,2)
15 OPENR(2,"P2",22)
16 INP#(2,B%,D$,R(1,1),MID$(E$,1,1),R(1,2),A$,C$)
17 INP#(2,G$)
18 INP#(2,H$;)
19 INP#(2,I$;)
20 INP#(2,J$)
21 CLOSE(2)
```

Výsledek:

```
B% = -12
D$ = "TEST" , protože max. délka znakového pole = 4
R(1,1) = 1.230
E$ = "V"
R(1,2) = 12.000
A$ = "ABC" , protože max. délka znakového pole = 3
C$ = "D2.4A"
G$ = "-12TEST1.23V12ABCD2.4A"
H$ = "-12TEST" , protože max. délka znakového pole = 7
I$ = "1.23V12" , protože max. délka znakového pole = 7
J$ = "ABCD2.4A"
```

Zpracování souborů

5.15.7 Identifikace konce souboru

EOF

S použitím funkce EOF lze zjišovat, zda je dosažen konec souboru (EOF = **e**nd **o**f **f**ile).

Funkce EOF dává logickou hodnotu TRUE, když je při čtecím přístupu dosažen konec souboru. Jinak je vydána hodnota FALSE.

Příklad:

```
:
9 DIM A$(10)
10 OPENR(1,"P",444) : I%=0
11 WHILE NOT (EOF(1))DO
12 INP#(1,A$)
13 I%=I%+1
14 END
15 CLOSE(1)
M30
```

5.15.8 Zavření souboru

CLOSE

Zavírá soubor.

Celkem může být současně otevřeno max. 9 souborů. Pokud je při 9 otevřených souborech zapotřebí přístup do dalšího souboru, je nutné nejprve nějaký soubor zavřít.

Otevřené soubory se proto musí zpravidla znovu zavírat okamžitě po skončení čtecích nebo zápisových operací.

Syntaxe:

CLOSE (<n>)

<n>: 1 až 9: Logické číslo souboru,
který chcete zavřít.

Příklad:

```
:
90 DIM A$(35)
100 XPOS = MPOS(1)
110 YPOS = MPOS(2)
120 OPENW(1,"P5",500,"ACHSPOS")
130 REWRITE(1)
140 PRN#(1,"X-Achse",XPOS,YPOS,"Y-Achse",YPOS)
150 CLOSE(1)
160 OPENR(1,"P5")
170 INP#(1,A$)
180 CLOSE(1)
:
```

Zpracování souborů

Ve výše uvedeném příkladu jsou aktuální polohy os X a Y předány do proměnných (řádek 90 až 110). Potom se otevře soubor 1 a uloží jako dílčí program P5 (řádek 120).

Následně je soubor popsán, resp. přepsán a pak zavřen (řádek 140 až 150). Soubor se pak otevře pro čtení a jeho obsah je přiřazen proměnné A\$. Po čtecím přístupu se soubor opět zavře (řádek 160 až 180).

5.15.9 Čtení pozice ukazatele souboru

FILEPOS

Funkce FILEPOS() udává číslo aktuálního záznamu v souboru Random, k němuž je pak možné získat přístup. Kromě toho je možné určit offset záznamu v rámci aktuálního záznamu souboru Random nebo vůči aktuálnímu bajtu u sekvenčního souboru, k němuž je možné získat přístup. Soubor přitom může být sekvenční stejně jako Random.

Offsetem se rozumí počet bajtů od začátku souboru k aktuálnímu bajtu v souboru. Offset záznamu udává, na který bajt v záznamu se má polohovat. Offset záznamu začíná u hodnoty 1 (= 1. bajt v záznamu) a může nabýt maximální hodnoty délka záznamu + 1 (poslední bajt v tomto záznamu je <LF>). Jestliže se nacházíte na pointeru EOF, je vrácena hodnota 1.

Syntaxe:

FILEPOS (<n>[,<režim>])

<n>: 1 až 9: Logické číslo souboru, ve kterém má být přečtena pozice ukazatele souboru.
Pokud nedodržíte rozsah hodnot, objeví se chybové hlášení UNZULÄSSIGE DATEINUMMER.

<Režim>: **U souborů Random:** rozsah hodnot 1 až 3

<Režim> = 1:

Udává offset na aktuální bajt, který může být přečten nebo popsán.

<Režim> = 2:

Udává číslo aktuálního záznamu, který může být přečten, resp. popsán. Jestliže se nacházíte na pointeru EOF, je výsledek: počet záznamů + 1.

<Režim> = 3:

Udává offset záznamu v aktuálním záznamu, který může být přečten, resp. popsán. Offset záznamu začíná u hodnoty 1 (1. bajt v tomto záznamu) a může nabýt maximální hodnoty délka záznamu + 1 (poslední bajt v tomto záznamu je <LF>).

Jestliže se nacházíte na pointeru EOF, je vrácena hodnota 1 a čtení ze souboru je nepřípustné.

Zpracování souborů

<Režim> není naprogramován:

Udává číslo aktuálního záznamu, který může být přečten, resp. popsán. Jestliže se nacházíte na pointeru EOF, je výsledek: počet záznamů + 1.

U sekvenčních souborů: rozsah hodnot 1

<Režim> = 1 nebo není naprogramován:

Udává offset na aktuální bajt, který může být přečten nebo popsán.

 **Pokud nedodržíte rozsah hodnot <režimu>, objeví se chybové hlášení UNZULÄSSIGER PARAMETER.**

Příklad: FILEPOS a sekvenční soubor

```
1 OPENW(1,"P2",200,"TEST")
2 FOR I% = 1 TO 10
3 PRN#(1,"TEST FUER FILEPOS")
4 NEXT
5 CLOSE(1)
6 OPENR(1,"P2")
7 SEEK(1,3)
8 POS% = FILEPOS(1)
9 POS1% = FILEPOS(1,1)
11 SEEK(1,0) : REM AUF DATEIENDE POSITIONIERT
12 POS2% = FILEPOS(1)
13 POS3% = FILEPOS(1,1)
14 CLOSE(1)
```

Výsledek:

POS% = 3 -> bajt-číslo

POS1% = 3 -> bajt-číslo

POS2% = 171 -> bajt-číslo

POS3% = 171 -> bajt-číslo

Zpracování souborů

Příklad: FILEPOS a soubor Random

```

1 OPENW(1,"P2",200,"TEST",1024)
2 FOR I% = 1 TO 10
3 PRN#(1,"TEST FUER FILEPOS")
4 NEXT
5 SEEK(1,3,2)
6 POS% = FILEPOS(1)
7 POS1% = FILEPOS(1,1)
8 POS2% = FILEPOS(1,2)
9 POS3% = FILEPOS(1,3)
10 PRN#(1,"Přepsání 3. záznamu od bajtu 2
    tímto textem")
11 SEEK(1,0) : REM Umístěno na konci souboru
6 POS% = FILEPOS(1)
7 POS1% = FILEPOS(1,1)
8 POS2% = FILEPOS(1,2)
9 POS3% = FILEPOS(1,3)
11 CLOSE(1)

```

Výsledek:

```

POS% = 3 -> Číslo záznamu, ve kterém se nacházíte
POS1% = 258 -> bajt-číslo
POS2% = 3 -> Číslo záznamu, ve kterém se nacházíte
POS3% = 2 -> Pozice uvnitř 3. záznamu
POS% = 11 -> Číslo záznamu, ve kterém se nacházíte
POS1% = 1281 -> bajt-číslo
POS2% = 11 -> Číslo záznamu, ve kterém se nacházíte
POS3% = 1 -> Pozice uvnitř 3. záznamu

```

5.15.10 Nastavení ukazatele souboru**SEEK**

Umísuje ukazatel souboru na určité místo v otevřeném souboru. Soubor přitom může být sekvenční stejně jako Random.

V případě sekvenčního souboru musí být soubor otevřen příkazem "OPENR(..)". U souborů Random je kromě toho přípustný také příkaz "OPENW(..)".

Syntaxe:

SEEK (<n>, <k> [, <o>])

<n>: Logické číslo souboru, ve kterém chcete polohovat ukazatel souboru.

Rozsah hodnot: 1 až 9

Pokud nedodržíte rozsah hodnot, objeví se chybové hlášení UNZULÄSSIGE DATEINUMMER.

Zpracování souborů

- <k>**: Číslo záznamu v souboru Random nebo číslo bajtu v sekvenčním souboru. Ukazatel souboru je umístěn na pozici <k>.
Wertebereich: 0 až poslední existující záznam, resp. 0 až poslední existující bajt.
Za poslední existující záznam platí záznam s pointerem EOF. Při hodnotě 0 proběhne polohování na pointer EOF. Pokud není dodržen rozsah hodnot nebo uvedený záznam neexistuje, objeví se chybové hlášení UNZULÄSSIGE KOMPONENTE.
- <o>**: Offset záznamu. Udává, na který bajt v záznamu se má polohovat.
Rozsah hodnot: 1 ... délka záznamu + 1.
Není-li naprogramován offset záznamu u souboru Random, je ukazatel souboru polohován na 1. bajt záznamu <k>.
Pokud nedodržíte rozsah hodnot, objeví se chybové hlášení UNZULÄSSIGER PARAMETER.
Tento parametr je přípustný pouze u souborů Random. Pokud je ovšem naprogramován, přestože se jedná o sekvenční soubor (otevřený ke čtení), objeví se chybové hlášení UNZULÄSSIGER PARAMETER.

Příklad: SEEK a sekvenční soubor

```
1 DIM A$(1):LJUST:OPENW(1,"P271",130,"TEST"):FOR I%=1 TO 10: PRN#(1,"!/-/!/-/!/-/!/-/!/-/!/-/!/-/!/"):NEXT: CLOSE(1):OPENR(2,"P271"):FOR I%=1 TO FILESIZE(2,2)-28: IF NOT (EOF(2)) THEN SEEK(2,I%):INP#(2,A$) ENDIF: IF (EOF(2)) THEN PRN#(0,"###",I%,". BYTE: <EOF>"): ELSE PRN#(0,"###",I%,". BYTE: <"A$,">") ENDIF: NEXT I%:CLOSE(2)
```

M30

Příklad: SEEK a soubor Random

```
1 OPENW(1,27272,200,"TEST",1024):LJUST
2 FOR I% = 1 TO 10
3 PRN#(1,I%,". Record")
4 NEXT
5 SEEK(1,3,4) : REM na 4. bajt ve 3. záznamu
umístěno
6 PRN#(1,"Přepsání 3. záznamu od bajtu 4
tímto textem")
7 SEEK(1,11):PRN#(1,"11. Record")
8 SEEK(1,11,5):PRN#(1,"@@")
9 SEEK(1,0):PRN#(1,"<EOF>")
10 SEEK(1,0,1):PRN#(1,"nový <EOF>")
11 CLOSE(1)
```

Zpracování souborů

5.15.11 Určení velikosti souboru

FILESIZE

Udává velikost souboru nebo mez, po kterou je již soubor zapsán. Soubor přitom může být sekvenční stejně jako Random. Příkaz účinkuje pouze na soubory, které byly otevřeny příkazem "OPENR(..)".

Syntaxe:

FILESIZE (<n> [, <k>])

<n>: 1 až 9

Logické číslo souboru, jehož velikost má být určena. Při nesprávném rozsahu hodnot se objeví chybové hlášení UNZULÄSSIGE DATEINUMMER.

<k>: **u souborů Random:** rozsah hodnot 1 až 4
u sekvenčních souborů: rozsah hodnot 1 až 2

<k> = 1:

Celková velikost paměového místa obsazeného souborem v bajtech.

<k> = 2:

Velikost obsazeného paměového místa od začátku datové oblasti po pointer EOF v bajtech (kromě velikosti pointeru EOF).

<k> = 3:

Maximální počet záznamů v souboru. Tento výsledek závisí na tom, s jakou délkou záznamu byl soubor otevřen.

<k> = 4:

Počet záznamů od začátku souboru po pointer EOF. Tento výsledek závisí na tom, s jakou délkou záznamu byl soubor otevřen.

<k> není naprogramováno:

jako <k> = 1.

Při nesprávném rozsahu hodnot pro <k> se objeví chybové hlášení UNZULÄSSIGER PARAMETER.

Příklad: FILESIZE a sekvenční soubor

```
1 OPENW(1,2,1000)
2 FOR I% = 1 TO 20
3 PRN#(1,"TESTE FILESIZE")
4 NEXT
5 CLOSE(1)
6 OPENR(2,2)
7 A%=FILESIZE(2)
9 B%=FILESIZE(2,1)
10 C%=FILESIZE(2,2)
11 CLOSE(2)
```

Proměnná INTEGER A% obsahuje hodnotu: 302

Proměnná INTEGER B% obsahuje hodnotu: 302

Proměnná INTEGER C% obsahuje hodnotu: 300

Zpracování souborů

Příklad: FILESIZE a soubor Random

```
1 OPENW(1,"P2",1000,10)
2 FOR I% = 1 TO 20
3 PRN#(1,"TESTE FILESIZE")
4 NEXT
5 CLOSE(1)
6 OPENR(2,2,10)
7 A%=FILESIZE(2)
9 B%=FILESIZE(2,1)
10 C%=FILESIZE(2,2)
10 D=FILESIZE(2,3)
10 E%=FILESIZE(2,4)
11 CLOSE(2)
```

Proměnná INTEGER A% obsahuje hodnotu: 222

Proměnná INTEGER B% obsahuje hodnotu: 222

Proměnná INTEGER C% obsahuje hodnotu: 220

Proměnná INTEGER D% obsahuje hodnotu: 20

Proměnná INTEGER E% obsahuje hodnotu: 20

Zpracování souborů

5.15.12 Vymazání souboru

ERASE

Maže soubory v aktuálním adresáři.

Syntaxe:

ERASE (<identifikace programu>)

<Identifikace programu> Výraz STRING; maximálně 30 znaků.
Jinak se objeví chybové hlášení FILENAME UNZU-
LÄSSIG.

Funkce ERASE může vracet do přiřazené proměnné INTEGER nebo do smyček a dotazů (WHILE, IF atd.) následující výsledky jako hodnotu Integer:

- 0: Soubor byl vymazán.
- 1: Soubor nebyl vymazán, protože neexistuje.
- 2: Soubor nebyl vymazán, protože je chráněný proti zápisu.
- 3: Soubor nebyl vymazán, protože je aktivní.

Není-li možné vymazat soubor, je vydáno příslušné varování a program je dál zpracováván.

Příklady:

```
10 IF ERASE("P1") <> 0 THEN ...  
  
10 I% = ERASE("P1")  
  
10 WHILE ERASE("P1") <> 0 DO ...
```

Příklad:

```
10 OPENW(1,"P2",200)  
11 OPENW(2,"P3",200)  
20 PRN#(1,"TEST1 FUER ERASE")  
21 PRN#(2,"TEST2 FUER ERASE")  
31 CLOSE(1)  
32 CLOSE(2)  
40 ERASE("P2")  
43 A$="P3"  
44 ERASE(A$)
```

Zpracování souborů

5.15.13 Určení přístupových oprávnění pro soubor

FILEACCESS

S použitím příkazu FILEACCESS je možné zjistit v CPL-programu, jestli existuje soubor a jaká má přístupová oprávnění.

Syntaxe:

FILEACCESS (<název souboru>)

<Název souboru> Název souboru s úplnou cestou jako výraz String. Bez udání cesty je soubor hledán v aktuálním adresáři. Funkce CPL vrací hodnotu Integer:

- 1 : Soubor neexistuje
- 0 : Soubor bez přístupových oprávnění
- jinak : Binárně kódovaná přístupová oprávnění:
 - Bit1: Provedení možné (X)
 - Bit2: Zápis povolen (W)
 - Bit3: Čtení povoleno (R)
 - Bit4: Soubor je adresář (D)
 - Bit5: Soubor je aktivní program (A)

Aktivní program je soubor, který je v kanálu

- zpracováván jako program
- zpracováván jako podprogram programu
- otevřen příkazem CPL.

Pokud jsou **přístupová oprávnění** tabulky posunutí nulových bodů nebo tabulky oprav nástroje, která se právě používá v probíhající dílčí programudie , dotazována příkazem CPL FILEACCESS, **není** bit5 nastaven.

Příklad:

```
10 I% = FILEACCESS("/usrfep/test.cnc")
```

Zpracování souborů

5.15.14 Určení data souboru

FILEDATE

Příkazem FILEDATE je možné v CPL-programu určit datum souboru. Chyba přístupu negeneruje chybu dílčího programu a místo toho dává funkce prázdný string.

Syntaxe:

FILEDATE (<název souboru>[,<režim>])

kde:

<Název souboru> Název souboru s úplnou cestou jako výraz String.
Bez udání cesty je soubor hledán v aktuálním adresáři. Funkce CPL vrací výraz String:

<Režim> Proměnná Integer pro funkční režim
(standardní = 1):
1 = datum souboru, formát: dd.mm.rr
2 = čas souboru, formát: hh.mm.ss

Příklad:

```
10 DIM DATUM$(10)
20 DATUM$ = FILEDATE("/usr/user/Test.txt",1)
30 IF LEN(DATUM$)>0 THEN
40 PRN#(0,"datum souboru: ",DATUM$)
50 ENDIF
```

Zpracování souborů

5.15.15 Kopírování souboru

FILECOPYY

Příkazem FILECOPYY se v CPL-programu kopíruje soubor.

Syntaxe:

FILECOPYY (<zdrojový soubor>, <cílový soubor>)

kde:

<Zdrojový soubor> Název zdrojového souboru s uvedením cesty jako výraz String. Bez udání cesty je soubor hledán v aktuálním adresáři.

<Cílový soubor> Název cílového souboru s uvedením cesty jako výraz String. Bez udání cesty je soubor vytvořen v aktuálním adresáři.

ERRNO Proměnná CPL, programovatelná na libovolném místě.

Při použití ERRNO se při chybě negeneruje běhová chyba; vrácené hodnoty jsou:

- 0: Přístup OK
- 8: Název zdrojového souboru včetně cesty příliš dlouhý
- 9: Přístup ke zdrojovému souboru není možný
- 10: Název cílového souboru včetně cesty příliš dlouhý
- 11: Název souboru (zdrojového nebo cílového) nepřijatelný
- 12: Kopírování není možné.

Příklad:

```
10 FILECOPYY("/usr/user/Test.txt", "/usr/user/Test.bak",  
ERRNO)  
20 IF ERNO = 0 THEN  
30 PRN#(0, "Kopírování OK")  
40 ENDIF
```

5.16 Komunikace

MMC

Odesílá informace o chodu programu z dílčího programu do klienta a čeká na výsledek z tohoto klienta.

Probíhá pomocí proměnných CPL, které mohou posílat hodnoty z dílčího programu stejně jako doručovat hodnoty zpět do dílčího programu.

Dílčí program se během zpracovávání zastaví na místě, kde je dosažen příkaz MMC.

Existují následující možnosti zpracování:

- Pokud se ještě nepřihlásil žádný klient, který dokáže zpracovat data příkazu MMC, je nastavena odpovídající vrácená hodnota (=1) a pokračuje zpracování dílčího programu.
- Jestliže existuje vhodný klient pro zpracování dat příkazu MMC, proběhne přiřazení mezi dílčím programem a klientem. Když klient odešle odpověď, je nastavena odpovídající vrácená hodnota a pokračuje zpracování dílčího programu.

Příkaz MMC může mít jako parametry maximálně 20 proměnných CPL. Klientu jsou sdělovány názvy i hodnoty těchto proměnných.

Syntaxe:

MMC (<CPL-Var1> [, <Cpl-Var2>.... [, <Cpl-VarN>]])

<CPL-Var1>... Proměnné CPL, N = max. 20

<CPL-VarN> Klient může do proměnných CPL uvedených v příkazu MMC zapisovat nové hodnoty. Tyto proměnné CPL uvedené v příkazu MMC se mohou používat v dílčím programu.

Jako výsledek dává příkaz MMC následující vrácené hodnoty:

- 0: OK
- 1: Neexistuje žádný klient
- 2: Chyba v klientu
- 9: Klient ukončen.

Komunikace

Příklad:

```
10 DIM PROGNAMES$ (50)
20 PROGNAMES$="WinProg"
30 INTPAR%=1
40 REALPAR=1.1
50 I%=MMC (PROGNAMES$,
    INTPAR%,REALPAR)

60 IF I%=0 THEN
70     IF INTPAR%=2 THEN
80         ...
90     ELSE
100        ...
110    ENDIF
120 ENDIF
```

Proměnné CPL PROGNAMES\$, INTPAR% a REALPAR s hodnotami jsou k dispozici klientu.

Příprava vět dílčího programu pokračuje na řádce 60 až poté, co je obdrženo odpovídající hlášení o dokončení.

Komunikace

Notizen:

Příloha

A Příloha

A.1 Zkratky

Zkratka	Význam
BA	Provozní režim
BOF	Uživatelské rozhraní
EGB	Elektrostaticky ohrožené moduly
ESD	electro static discharge Zkratka pro všechna označení, která se týkají elektrostatických výbojů, např. ochrana proti ESD, ohrožení ESD.
Fx	Funkční klávesa s číslem x
HP	Hlavní program
LSEC	Kompenzace chyby stoupání vřetena (Lead Screw Error Compensation)
MDI	Provozní režim "ruční zadávání" (manual data input)
MP	Parametry stroje
MZA	Stavová indikace stroje
NC, CNC	Číslicové řízení
PE	Protective Earth, ochranný zemnicí vodič
SK	Soft klávesa
SPS	Řízení programovatelné z paměti
UP	Podprogram
WMH	Výrobce obráběcího stroje
WZ	Nástroj

Příloha

A.2 Tabulkové přehledy NC-funkcí

Alfanumericky seřazené podle úplného označení: od strany A-2

Alfanumericky seřazené podle skupin: od strany A-27

Alfanumericky seřazené podle úplného označení:

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("..." zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
;	-	-	Komentář: vynechat zbytek řádku	2-21		
()	-	-	Komentář: vynechat obsah závorky	2-21		
(MSG)	-	-	vydat text hlášení	2-23		
/	-	-	Vynechat větu	2-38		
//	-	-	Komentář	2-21		
ABS	-	CPL	ABS(<vstupní hodnota>) Vrací vstupní hodnotu jako absolutní hodnotu.	2-60	ABS	-
AC(...)	-	-	Lokální programování absolutního rozměru, např. X=AC(10)	3-69	AC(...)	-
ACOS	-	CPL	<Funkční hodnota> = ACOS(<vstupní hod- nota>) Použití funkce arcuscosinus na <vstupní hod- notu>.	2-61	ACOS	-
AND	-	CPL	<Výtisk1> AND <Výtisk2> Binární operace se dvěma výrazy typu BOOL, resp. INTEGER	2-62	AND	-
APOS	-	CPL	APOS(<volba osy>) Předává aktuální skutečnou hodnotu osy vzhledem k nulovému bodu stroje.	5-11	APOS	-
Area	ARA	-	Area(<BNr>,<Sta>{,<Mod>,{<P1>},{<P2>}, {<D1>},{<D2>}}) Definuje, aktivuje nebo deaktivuje až 10 ob- délníkových dvojrozměrných mrtvých nebo pracovních oblastí s osově paralelními me- zemi.	4-2	-	-
ASC	-	CPL	ASC(<znakový řetězec>) Vydává pořadové číslo prvního znaku (ASCII- kód) ze <znakového řetězce> jako hodnotu INTEGER.	5-97	ASC	-
ASIN	-	CPL	<Funkční hodnota> = ASIN(<vstupní hod- nota>) Použití funkce arcussinus na <vstupní hod- notu>.	2-61	ASIN	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
ASPCLR	-	-	ASPCLR(<Up-č.>) Asynchronní podprogramy: odhlásit	4-5	-	-
ASPDIS	-	-	ASPDIS(<Up-č.>) Asynchronní podprogramy: vypnout	4-6	-	-
ASPENA	-	-	ASPENA(<Up-č.>) Asynchronní podprogramy: zapnout	4-6	-	-
ASPRTP	-	-	ASPRTP(<Up-č.>,<bod>) Asynchronní podprogramy: definovat bod op- ětovného najetí	4-7	-	-
ASPSET	-	-	ASPSET(<Up-č.>,<Up-název>{,<příznaky>}) Asynchronní podprogramy: přihlásit	4-8	-	-
ASPSTA	-	-	ASPSTA(<Up-č.>{,<kanál-č.>}) Asynchronní podprogramy: spustit	4-9	-	-
AssLogName	ALN	-	ALN(...) Přiřadit logický název osy	4-10	G515	-
ASTOPA	-	-	ASTOPA[<č. kanálu>, <podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile jsou splněny všechny podmínky, zastaví se řídicí kanál. Podmínky se vztahují k polohám os, např. "Z" > 20	2-78	-	-
ASTOPO	-	-	ASTOPO[<č. kanálu>, <podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile je splněna jedna podmínka, zastaví se řídicí kanál. Podmínky se vztahují k polohám os, např. "Z" > 20	2-78	-	-
ATAN	-	CPL	<Funkční hodnota> = ATAN(<vstupní hod- nota>) Použití funkce arcustangens na <vstupní hod- notu>.	2-61	ATAN	-
ATCAL	-	-	ATCAL(<soubor>,<OptData>,<maska> {,<Info>}{,<AnzIt>}) Kalibrovat kinematiky os: Optimalizovat para- metry:	4-11		
ATFWD	-	-	ATFWD(<Koord>,<AxKoord>{,<ParData>}) Kalibrovat kinematiky os: Přepočítat parametry	4-13		
ATGET	-	-	ATGET(<ParData>{,<ATrafNr>}) Kalibrovat kinematiky os: Načíst parametry z NC	4-14		
ATPUT	-	-	ATPUT(<ParData>{,<ATrafNr>}) Kalibrovat kinematiky os: Zapsat parametry do NC	4-15		

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
ATrans	ATR	Posunutí programových souřadnic	ATR(X..., Y..., Z..., ...): Zapnutí aditivního posunutí programových souřadnic ATR(): Vypnutí aditivního posunutí programových souřadnic	4-148		
AUXFUNC	-	-	Provést aktivní pomocné funkce všech skupin	4-16		
AxAcc	AAC	Zrychlení os	AAC(X..., Y..., Z..., ...): Zapnout AAC(1): Zapnout s uloženým nastavením AAC(): Vypnout Programovat zrychlení os	4-17	G6/G7	-
AxAccSave	AAS	-	Uložit aktuální zrychlení os.	4-17	-	-
AxCouple	AXC	Spojení os	AXC(<Master>,<Slave1>(…),<Slave2>(…),…): Aktivovat spojení os AXC(): Deaktivovat spojení os	4-19	G580/G581	
AxisToSpindle	ATS	-	ATS(<název osy>) Přepnout osu na provoz vřetena.	4-22		
AXO	-	CPL	AXO((<volba osy>[,<typ volby>]) Předává aktivní posunutí G92 pro souřadnici.	5-15	AXO	-
AXP	-	CPL	AXP(číslo osy>,<informace o dráze>) Tato funkce umožňuje dílčí a měřicí programy nezávislé na rovině (naprogramujte AXP místo hodnoty adresy).	5-43	AXP	-
BCD	-	CPL	<Hodnota BCD> = BCD(<binární hodnota>) Konvertovat formát BCD na binární formát.	2-62	BCD	-
BcsCorr	BCR	Oprava polohy obrobku	BCR({<X _W -offset>},{<Y _W -offset>}{,<Z _W -offset>},{<úhel1>}{,<úhel2>}{,<úhel3>}}): Zapnout BCR(): Vypnout Umístění: Oprava polohy obrobku.	4-23	G138/G139	-
BIN	-	CPL	<Binární hodnota> = BIN(<hodnota BCD>) Konvertovat binární formát na formát BCD.	2-62	BIN	-
BITIF	-	CPL	BITIF(<bitový signál>[,<index>[,<jednotka IF>]]) Přístup k digitálnímu rozhraní mezi NC a SPS.	5-44	IC	-
BlkNmb	BNB	-	BlkNmb(<Anz>) Omezuje počet programových vět, které mohou být načteny a brány v úvahu přípravou vět.	2-71	PREPNUM	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
BSTOPA	-	-	BSTOPA[<č. kanálu>, <podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile jsou splněny všechny podmínky, zastaví se řídicí kanál. Podmínky se vztahují k základním polohám obrobku, např. "Z" > 20	2-80	-	-
BSTOPO	-	-	BSTOPO[<č. kanálu>, <podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile je splněna jedna podmínka, zastaví se řídicí kanál. Podmínky se vztahují k základním polohám obrobku, např. "Z" > 20	2-80	-	-
CALL	-	CPL	CALL <číslo programu> [<předávací parametr1>,...] [DIN] Vyvolání podprogramu z programu CPL	2-29	CALL	-
CASE	-	CPL	CASE <Integer-výraz> OF LABEL <Int.-konstanta>[,<další Int.-konstanta>] [: <instrukce>] <Instrukce> LABEL ... OTHERWISE <instrukce> <Instrukce> ENDCASE Podmíněný výběr z několika alternativ.	2-40	CASE	-
ChLength	CHL	Fasety/zao- blení	CHL(<délka fasety>) : Zapnout CHL () : Vypnout Vložit přechodové fasety s definovanou délkou.	4-25	G234/G35	-
CHR\$	-	CPL	CHR\$(<Integer-výraz>) Dává znak, jehož pořadové číslo v tabulce ASCII je stejné jako hodnota vydaná parametrem <INTEGER-výraz>.	5-97	CHR\$	-
ChSection	CHS	Fasety/zao- blení	CHS(<úsek fasety>) : Zapnout CHS () : Vypnout Vložit přechodové fasety s definovaným úsekem.	4-25	G234/G35	-
CLOCK	-	CPL	<Funkční hodnota> = CLOCK Dotaz počítadla času v milisekundách.	5-46	CLOCK	-
CLOSE	-	CPL	CLOSE(<n>) Zavírá otevřený soubor po ukončení čtecích nebo zápisových operací.	5-123	CLOSE	-
CLRWARN	-	CPL	Vymazat všechna výstražná hlášení kanálu generovaná pomocí SETWARN.	2-58	-	-
COF	-	CPL	COF(<volba osy>[,<typ volby>]) Dává pro aktuální kanál poslední naprogramované posunutí kontury (Shift) souřadnice.	5-16	COF	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
COFFS	-	-	Offset kontury pro opravu 3D poloměru	3-79	-	-
Collision	CLN	Monitorování kolizí	CLN(1): Zapnout CLN(CollErr.. LA... DLA... DEF) : Zapnout s parametrizací CLN() : Vypnout Monitorování kolizí pro opravu 2D dráhy frézy.	4-27		
ConstFeed	CFD	V-Profil	Konstantní rychlost	4-66	G310	-
Coord	CRD	Transformace os	CRD(x): Zapnout transformaci os CRD() : Vypnout transformaci os např. transformace čelní plochy	4-30	Coord	G30ff.
COS	-	CPL	<Funkční hodnota> = COS(<vstupní hodnota>) Funkce kosinus pro <vstupní hodnotu>.	2-61	COS	-
CoupleSpline-Tab	CST	Spojení os	CST(STAB(...)) Vytvoření tabulky spline spojení	4-32	G582	
CPROBE	-	CPL	CPROBE(<výběr>[, <typ výběru>]) Načítá naměřenou hodnotu vždy pro jednu souřadnici	5-12	CPROBE	-
D	D	D-korektura	Dx: Volba D-korektury (1-99 nástrojů) D0: Zrušení volby D-korektury.	3-100	G147 - G847,G148	
DATE	-	CPL	<Proměnná String> = DATE Přiřazuje proměnné <STRING> datum ve formě DD.MM.	5-46	DATE	-
DBSEA	-	CPL	DBSEA(<DbTab>,<Key1>,<Key2>,<Search-Cond>,<SeachRes>[,<ResVar>]) Vyhledává datové záznamy v databázové tabulce.	5-23		
DBTAB	-	CPL	DBTAB(<DbTabelle>,<Key1>,<Key2>[,<Res-Var>]) Načítá kompletní datový záznam nebo dílčí strukturu databázové tabulky do proměnné CPL nebo ji z ní vypisuje.	5-22		
DCT	-	CPL	DCT(<výběr hodnoty>, <datový záznam>[, <tabulka>] [, <jednotka>]) Čtecí a zápisový přístup do libovolné tabulky D-korektur nebo k externím hodnotám oprav nástroje.	5-21	TC	-
DcTsel	DCS	-	DCS({<cesta>}<název souboru>) Volba tabulky D-korektur.	4-33	G22 K	-
DefAxis	DAX	-	Vytvořit standardní konfiguraci os	4-34	G513	-
DefTangTrans	DTT	Přechodový úhel	DTT(<přechodový úhel>): Aktivovat DTT() : Deaktivovat Definice tangenciálního přechodu kontury		G228	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
DiaProg	DIA	Průměr/ poloměr	Programování průměru	4-35	DIA	G16
DIM	-	CPL	DIM <název proměnné>(<velikost pole1>[,<ve- likost pole2>]) Stanovení velikosti pole (dimenzování) proměnných ARRAY s konstantami INTEGER.	2-53, 5-93	DIM	-
DIRCR	-	CPL	DIRCR(<adresář>) Vytváří nový adresář.	5-115		
DIRDEL	-	CPL	DIRDEL(<adresář>) Maže prázdný adresář.	5-116		
DIRINF	-	CPL	DIRINF(4,[<index2>],<Ergebnisvar>[,<číslo souboru]) Vývolává data správy systému souborů.	5-114		
DistCtrl	DCR	-	DCR(1) Regulace výšky pro digitalizaci, zapnutí podle parametrů stroje DCR(<funkce>) zapnutí s vlastními konfigu- račními daty DCR() vypnutí regulace výšky	4-37	DistCtrl	-
DPC	-	CPL	DPC(<výběr>[,<typ výběru>]) Dává pro aktuální kanál poslední naprogramo- vané parametry opravy polohy obrobku BcsCorr souřadnice (hodnoty posunutí a úhlu otočení).	5-17	DPC	-
ED	ED	ED-korektura	ED x: Volba externí opravy nástroje (1..16 nástrojů/břitů) ED0: Zrušení volby externí opravy nástroje	3-101	G145 - G845,G146	
EndPosCouple	EPC	-	EPC(<M-koord>,<S-koord>,<faktor>): Zapnutí spojení koncové polohy EPC(): Vypnutí spojení koncové polohy	4-39	-	-
EOF	-	CPL	EOF(<n>) Kontrola konce souboru.	5-123	EOF	-
ERASE	-	CPL	ERASE(<identifikace programu>) Vymaže soubory.	5-130	ERASE	-
ERRNO	-	CPL	Chybové hlášení různých funkcí CPL.	2-56	-	-
F	F	-	F-adresa pro G93, G94, G95	3-96	F	
FA	FA	-	Rychlost pro asynchronní osy.	3-97	FA	
FALSE	-	CPL	<Proměnná BOOL> = FALSE Pravdivostní hodnota proměnné BOOL	2-52	FALSE	-
FeedAd	FAD	Přizpůsobení posuvu	FAD(1) : Osu z vytváření posuvu vyjmout FAD() : Zachovat osu ve vytváření posuvu	4-41	G594/G595	

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
FeedForward	FFW	Servo řízení	FFW(X..., Y..., Z..., ...): Zapnout FFW(): Vypnout Aktivovat servo řízení	4-42	G114/G115	G6/G7
FILEACCESS	-	CPL	FILEACCESS(<název souboru> Určuje, zda existuje soubor a jaká má přístupová oprávnění.	5-131	FILEAC- CESS	-
FILECOPY	-	CPL	Kopírovat soubory.	5-133	-	-
FILEDATE	-	CPL	FILEDATE(<název souboru>[, <režim>]) Zjišťuje datum/čas souboru.	5-132	FILEDATE	-
FILENO	-	CPL	Dává další platné logické číslo souboru.	5-113		
FILEPOS	-	CPL	FILEPOS(<n>[, <režim>]) Dává číslo aktuálního záznamu a offset záznamu souboru Random. Dává u sekvenčních souborů aktuální bajtovou pozici ukazatele souboru.	5-124	FILEPOS	-
FILESIZE	-	CPL	FILESIZE(<n>[, <k>]) Dává velikost souboru nebo mez, po kterou je již soubor zapsán.	5-128	FILESIZE	-
FlyMeas	FME	Průběžné měření	FME(MpiAxis< i>) X... Y... Z... Pojezd s průběžným měřením	4-44	G275	-
FOR NEXT	-	CPL	FOR <číselná prom.>=<počáteční hodnota> [STEP <velikost kroku>] TO <koncová hod- nota><procedura> NEXT [<číselná proměnná>] Smyčková konstrukce s automaticky běžícím počítadlem.	2-42	FOR NEXT	-
FsMove	FSM	-	Pojezd do pevného dorazu (Fs = fixed stop)	4-48	G475	G75
FsProbe	FSP	-	Měření na pevném dorazu	4-46	G475	G75
FsReset	FSR	-	Odstranění pevného dorazu	4-48	G477	G76
FsTorque	FST	-	Moment pro pevný doraz	4-48	G476	AXD
G	-	-	Uživatelsky definované lokální podprogramy			
G00	G0	Interpolace	Lineární interpolace (rychl oběh)	3-3	G00/G10	G00

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G00(..)	G0(..)	Interpolace	Rychloběh s možnostmi Volitelné parametry: NIPS: bez přesného zastavení IPS1: Okno přesného zastavení 1 IPS2: Okno přesného zastavení 2 IPS3: Okno přesného zastavení 3 (pouze brzdění na V=0) POL/POLAR: s programováním polárních souřadnic, např. G0(POL) X50 A45 B10, s A, B = polární úhel 1/2	3-3		
G01	G1	Interpolace	Lineární interpolace (posuv)	3-5	G01/G11	G01
G02	G2	Interpolace	Kruhová interpolace ve směru hodinových ručiček (vč. Helical-N)	3-7	G02/G12	G02
G02(POL)	G2(..)	Interpolace	Kruhová interpolace ve směru hodinových ručiček (vč. Helical-N) s programováním polárních souřadnic	3-11	G02/G12	G02
G03	G3	Interpolace	Kruhová interpolace proti směru hodinových ručiček (vč. Helical-N)	3-7	G03/G13	G03
G03(POL)	G3(..)	Interpolace	Kruhová interpolace proti směru hodinových ručiček (vč. Helical-N) s programováním polárních souřadnic	3-11	G02/G12	G02
G04	G4	-	G4(F..) Doba prodlevy v sekundách G4(S...) Doba prodlevy v otáčkách vřetena, vztažným vřetenem je hlavní vřeteno (MainSp)	3-12	G04/G104	G04
G05	G5	Interpolace	Tangenciální vstup kruhu (vč. Helical-N)	3-13	G05	-
G06	G6	Interpolace	Spline křivky, nurb křivky	3-14	G99	-
G08	G8	Strmost dráhy	Zapnutí strmosti dráhy se SHAPE dráhy podle parametrů stroje	3-21	G08	G08
G08(SHAPE...)	-	Strmost dráhy	Zapnutí strmosti dráhy s naprogramovaným SHAPE dráhy	3-23	G108	G08
G09	G9	Strmost dráhy	Vypnutí strmosti dráhy (bez SHAPE)	3-21	G09	G09

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina (".." zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G09(..)	G9(..)	Strmost dráhy	G9(X..., Y..., ...) Vypnutí strmosti dráhy s osově naprogramo- vaným pořadím SHAPE G9(ASHAPE) Vypnutí strmosti dráhy s osovým pořadím SHAPE podle parametrů stroje G9(SHAPE ...) Vypnutí strmosti dráhy s naprogramovaným SHAPE dráhy G9(SIN ...) Vypnutí strmosti dráhy se SHAPE tvaru sin ² (možné pořadí SHAPE: 5, 10, 15, 20, 40)	3-23	G408, G608	
G140	-	Oprava 3D poloměru	Vypnutí opravy 3D poloměru	3-79	G141	
G141	-	Oprava 3D poloměru	Oprava 3D poloměru vlevo od kontury	3-79	G141	
G142	-	Oprava 3D poloměru	Oprava 3D poloměru vpravo od kontury	3-79	G142	
G152.1	G152	SE řada 1	Umístění: Šikmá rovina, programovatelná, řada 1	3-84	G352	
G152.2-G152.5	-	SE řada 2-5	Umístění: Šikmá rovina, programovatelná, řada 2-5	3-84	G452	
G153	-	SE	Umístění: Šikmá rovina, vše vypnout	3-84, 3-84	-	
G153.1-G153.5	-	SE řada 1-5	Umístění: Šikmá rovina, řada 1-5 vypnout	3-84, 3-84	G353	
G154.1	G154	SE řada 1	Umístění: 1. Šikmá rovina, řada 1 zapnout	3-84	G354	
G154.2-G154.5	-	SE řada 2-5	Umístění: 1. Šikmá rovina, řada 2-5 zapnout	3-84	G454/G554	
G155.1	G155	SE řada 1	Umístění: 2. Šikmá rovina, řada 1 zapnout	3-84	G355	
G155.2-G155.5	-	SE řada 2-5	Umístění: 2. Šikmá rovina, řada 2-5 zapnout	3-84	G455/G555	
G156.1	G156	SE řada 1	Umístění: 3. Šikmá rovina, řada 1 zapnout	3-84	G356	
G156.2-G156.5	-	SE řada 2-5	Umístění: 3. Šikmá rovina, řada 2-5 zapnout	3-84	G456/G556	
G157.1	G157	SE řada 1	Umístění: 4. Šikmá rovina, řada 1 zapnout	3-84	G357	
G157.2-G157.5	-	SE řada 2-5	Umístění: 4. Šikmá rovina, řada 2-5 zapnout	3-84	G457/G557	
G158.1	G158	SE řada 1	Umístění: 5. Šikmá rovina, řada 1 zapnout	3-84	G358	
G158.2-G158.5	-	SE řada 2-5	Umístění: 5. Šikmá rovina, řada 2-5 zapnout	3-84	G458/G558	
G159.1	G159	SE řada 1	Umístění: 6. Šikmá rovina, řada 1 zapnout	3-84	G359	
G159.2-G159.5	-	SE řada 2-5	Umístění: 5. Šikmá rovina, řada 2-5 zapnout	3-84	G459/G559	
G16	-	Volba roviny	žádná rovina	3-27	G16	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G17	-	Volba roviny	Výběr roviny XY	3-28	G17/G20	G17/G20
G17(...), G18(...), G19(...)	-	Volba roviny	Rozšířené přepínání rovin G17/18/19(<osa1>,<osa2>,<osa3>) Osy v závorkách upínají WCS a dostávají význam X, Y a Z. Následně se vybírá naprogramovaná rovina. G17/18/19() Nastavte soustavu souřadnic obrobku opět na standardní hodnoty a potom vyberte naprogra- movanou rovinu.	3-30	-	-
G18	-	Volba roviny	Výběr roviny XZ	3-28	G18/G20	G18/G21
G184	-	Vrtací cyklus	Vrtací cyklus: Řezání vnitřních závitů bez vy- rovnávacího sklíčidla	3-68	G184	
G19	-	Volba roviny	Výběr roviny YZ	3-28	G19/G20	G19/G22
G20	-	Volba roviny	Volný výběr roviny (nezávislý na WCS) Umístění má vliv na WCS, rovina je definována nezávisle na WCS.	3-32	-	-
G33	-	Závit	Řezání závitů	3-33	G33	G33
G40	-	Oprava dráhy	Vypnutí opravy dráhy frézy	3-39	G40	G40
G41	-	Oprava dráhy	Oprava dráhy frézy vlevo od obrobku	3-39	G41	G41
G42	-	Oprava dráhy	Oprava dráhy frézy vpravo od obrobku	3-39	G42	G42
G43	-	Strategie vkládání	Strategie vkládání kruhového oblouku	3-41	G68	G43
G44	-	Strategie vkládání	Strategie vkládání průsečíku	3-41	G69	G44
G45	-	Rychlost frézy	Rychlost bodu záběru frézy	3-43	G64	G99
G46	-	Rychlost frézy	Rychlost středu frézy	3-43	G65	G98
G47	-	Oprava nástroje	Zapnutí opravy délky nástroje	3-44	G146/8	G47
G47(..)	-	Oprava nástroje	G47(<souřadnice L1>,<souřadnice L2>,<souřadnice L3>) G47(ActPlane) Zapnutí opravy délky nástroje s přepnutím přiřazení opravy. G47(): Zapnutí opravy délky nástroje s přiřazením opravy podle parametrů stroje.	3-44	G146/8	G47
G48	-	Oprava nástroje	Vypnutí opravy délky nástroje	3-44	G145/7ff.	G48/G49
G53	-	Posunutí nu- lového bodu	Vypnutí posunutí všech nulových bodů	3-46	-	

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G53.1-G53.5	-	Posunutí nulového bodu, řada 1-5	Vypnutí posunutí nulového bodu, řada 1-5	3-46	G53-G253	
G54.1	G54	Posunutí nulového bodu, řada 1	1. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G54.2-G54.5	-	Posunutí nulového bodu, řada 2-5	1. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G54-G254	
G55.1	G55	Posunutí nulového bodu, řada 1	2. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G55.2-G55.5	-	Posunutí nulového bodu, řada 2-5	2. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G55-G255	
G56.1	G56	Posunutí nulového bodu, řada 1	3. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G56.2-G56.5	-	Posunutí nulového bodu, řada 2-5	3. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G56-G256	
G57.1	G57	Posunutí nulového bodu, řada 1	4. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G57.2-G57.5	-	Posunutí nulového bodu, řada 2-5	4. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G57-G257	
G58.1	G58	Posunutí nulového bodu, řada 1	5. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G58.2-G58.5	-	Posunutí nulového bodu, řada 2-5	5. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G58-G258	
G59.1	G59	Posunutí nulového bodu, řada 1	6. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G59.2-G59.5	-	Posunutí nulového bodu, řada 2-5	6. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G59-G259	
G61		Přesné zastavení	Zapnutí přesného zastavení	3-49	G61/G161	G61
G61(IPS...)	-	Přesné zastavení	Zapnutí přesného zastavení s oknem přesného zastavení IPS1, IPS2 nebo IPS3	3-49	G61/G161	G61
G62	-	Přesné zastavení	Vypnutí přesného zastavení	3-49	G62/G162	G62

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G63	-	Závit	G63(M3/M4, S.../H...) F... Z... ... Řezání vnitřních závitů bez vyrovnávacího sklíčidla	3-51	G32	G63/G64
G70	-	Inch/metrické	Coulové programování, působí na naprogramované dráhové a délkové rozměry, posuvy a zrychlení.	3-53	G70	G70
G71	-	Inch/metrické	Metrické programování	3-54	G71	G71
G74	-	-	G74 X1 Y1 Z1 ... Najetí na souřadnice referenčního bodu	3-55	G74/G374	G74
G74(Home)	-	-	G74(HOME) X1 Y1 Z1 Najetí do referenčního bodu (skutečné referen- cování, také pro asynchronní osy)	3-55	G74/G374	G74
G75	-	Měřicí dotyk	Najetí k měřicímu dotyku (přerušení pohybu)	3-57	G75	-
G76	-	-	Najetí do pevné polohy stroje (souřadnice stroje)	3-58	G76	-
G77	-	-	G77 <souřadnice 1><režim> <souřadnice n><režim> ... F<hodnota> Alternativní syntaxe: REPOS Asynchronní podprogramy: Změna polohy jednotlivých souřadnic	3-59	-	G77
G80	-	Vrtací cyklus	Vypnout vrtací cyklus	3-61	G80	-
G81	-	Vrtací cyklus	Vrtací cyklus: Vrtání se zpětným pohybem v rychloběhu	3-62	G81	DEFINE
G82	-	Vrtací cyklus	Vrtací cyklus: Vrtání se zpětným pohybem v posuvu	3-62	G82	DEFINE
G83	-	Vrtací cyklus	Vrtací cyklus: Vrtání hlubokých děr	3-64	G83	DEFINE
G84	-	Vrtací cyklus	Vrtací cyklus: Řezání vnitřních závitů s vyrovnávacím sklíčidlem	3-65	G84	DEFINE
G85	-	Vrtací cyklus	Vrtací cyklus: Vyvrtávání se zpětným pohybem v rychloběhu	3-66	G85	DEFINE
G86	-	Vrtací cyklus	Vrtací cyklus: Vyvrtávání se zpětným pohybem v posuvu	3-67	G86	DEFINE
G90	-	Abs/Rel	Programování absolutních rozměrů	3-69	G90	G90
G91	-	Abs/Rel	Programování relativních rozměrů	3-69	G91	G91
G93	-	Programování posuvu	Časové programování	3-70	G93	G93
G94	-	Programování posuvu	Programování posuvu (za min)	3-71	G94	G94

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G94(...)	-	Programování posuvu	G94({DF <hodnota>}, { DS1 <hodnota>, DS2 <hodnota>, ...}) Inkrementální programování rychlosti s přízpusobením zrychlení.	3-72	G94	G94
G95	-	Prog. posuvu	Programování posuvu (na ot.)	3-74	G95	G95
G96	-	Prog. vřetena	G96({<vztažná osa>{<působíště>}}) Konstantní řezná rychlost	3-75	G96/G196	G96
G97	-	Prog. vřetena	Přímé programování počtu otáček	3-75	G97	G97
GetAxis	GAX	-	GAX(...) Převzít volnou osu do kanálu	4-51	G510	
GETERR	-	CPL	GETERR(<kanál> [,<kategorie>],<chyba>-č.> [,<počet>]) Udává pro aktuální chybu č. chyby, č. kanálu a kategorii chyby.	5-47	GETERR	-
GMSG	-	-	GMSG(<text pokynu>) Programování pokynů na uživatelském roz- hraní.	2-23	(GMSG	
GoAhead	GOA	-	Skok dopředu	2-33		
GoBack	GOB	-	Skok dozadu	2-34		
GoCond	GOC	-	Podmíněný skok	2-35		
GOTO	-	CPL	GOTO <cíl skoku> Nepodmíněné programové skoky na číslo řádku, číslo věty nebo návěští.	2-37	GOTO	-
GoTo	-	-	Nepodmíněný skok	2-36		
HsBlkSwitch	HSB	-	HSB(HS<X>=<Y>) Průběžná změna vět pomocí signálu High- Speed	4-52	G575	-
HsBlkSwitch(. .HSSTOP=..)	HSB	-	HSB(...,HSSTOP=..) Změna vět s přerušením pomocí signálu High- Speed	4-55	G575	-
HWOC	-	-	HWOC({CHAN<č. kanálu>},CRDNO <č. souřadnice>,{STEP<Inkr>}) Zapnutí online opravy v souřadnicích obrobku HWOCDIS({CHAN<č. kanálu>}) Vypnutí online opravy, uložit hodnoty HWO() Vypnutí online opravy, vymazat hodnoty	4-57	HWO- CON, HWO- COFF	-
IC(...)	-	-	Lokální programování relativních rozměrů, např. X=IC(5)	3-69	IC(...)	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
IF ENDIF	-	CPL	IF <podmínka> THEN <procedura> [ELSE <alternativní procedura>] ENDIF Podmíněný skok na proceduru nebo alternativní proceduru.	2-39	IF ENDIF	-
InitMeas	IME	-	IME(MpiAxis i) Inicializovat průběžné měření.	4-44	G175	-
INP#	-	CPL	INP#(<n>, <proměnná>[, <proměnná>] [...] [:]) Čtecí přístup do souboru.	5-121	INP#	-
INSDEP	-	-	Hloubka ponoru, oprava 3D poloměru.	3-79	-	-
INSTR	-	CPL	INSTR(<znakový řetězec>, <String-výraz>[, <počáteční bod>]) Prohledává výraz <String> od <počátečního bodu> s ohledem na <znakový řetězec> a udává jeho počáteční polohu jako hodnotu INTEGER.	5-96	INSTR	-
INT	-	CPL	<Číslo Integer>=INT(<výraz Real>) Konvertuje <výraz Real> na <číslo Integer> odříznutím míst za desetinnou tečkou.	2-60	INT	-
JogWCSSelect	-	-	Vybírá souřadnice pro krokování v souřadnicích obrobku (seřizovací provoz).	4-59	JogWCS-Select	-
KvProg	KVP	Progr. KV	KVP(Y..., Y..., Z..., ...): Zapnutí programování KV KVP(): Vypnutí programování KV	4-60	G14/G15	AXD
LEN	-	CPL	LEN(<výraz String>) Udává počet znaků výrazu STRING.	5-96	LEN	-
LEN	-	-	LEN=<hodnota> Rozděluje pojedzovou větu na několik stejně dlouhých dílčích drah.	4-62	-	-
LFCConf	LFC	-	LFC({LL(...)}) Parametrizovat řízení laserového výkonu	4-64	LFPON	-
LFP	LFP	-	LFP, LFP(1), LFP({LL(...)}) Zapnout řízení laserového výkonu LFP(0) Vypnout řízení laserového výkonu	4-64	LFPON, LFPOFF	-
LinDownFeed	LND	V-Profil	Lineární brzdění	4-67	G312	-
LinModZp	LMZ	-	LMZ{(LinModAxis i)} Vynulování lineární modulu osy (Zp = zero point)	4-69	G105	-
LinUpFeed	LNU	V-Profil	Lineární zrychlení	4-66	G311	-
LJUST	-	CPL	Přepnutí na datový výstup zarovnaný doleva, účinný až do konce běhu programu.	5-117	LJUST	-
m	m	-	Uživatelsky definované lokální podprogramy			

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
M0	M0	-	Zastavení programu	3-86	M0	
M1	M1	-	Podmíněné zastavení programu	3-86	M1	
M19, M119, M219	-	-	Seřízení vřetena.	3-91	M19-M219	
M2	M2	-	Konec hlavního programu	3-87	M2	
M3, M103, M203, M13, M113, M213	-	-	Pravotočivý chod vřetena, Pravotočivý chod vřetena se zapnutou chladicí kapalinou	3-88	M3 - M203, M13-M213	
M30	M30	-	Konec hlavního programu	3-87	M30	
M4, M104, M204, M14, M114, M214	-	-	Levotočivý chod vřetena, Levotočivý chod vřetena se zapnutou chladicí kapalinou	3-89	M4 - M204, M14-M214	
M40, M140, M240	-	-	automatická volba převodových stupňů	3-93	M40-M240	
M41 ... M44, M141 ... M144, M241 ... M244	-	-	Ruční volba převodových stupňů	3-94	M41 - M44, M141-M14 4 M241-M24 4	
M48, M148, M248	-	-	Vyřadit převodový stupeň.	3-95	M48-M248	
M5, M105, M205	-	-	Zastavení vřetena	3-90	M5 - M205	
MainSp	MSP	-	Výběr hlavního vřetena např. pro G33 a G95	4-70	MAINSP	SPF
MCODS	-	CPL	MCODS(<typ>,<kanál>,<verze>,<vyrovnávací paměť>,<velikost>[,<P1>]) Vyvolává datové služby Motion Control v NCS. Tak je možné načítat data a stavy z NC.	5-52	MCODS	-
MCOPS	-	CPL	MCOPS(<funkce>,<kanál>[[, [<P1>][, [<P2>], [<P3>]]], <P4>]) Vyvolává procesové služby Motion Control v NCS. Tak lze řídit kanály v NC.	5-82	MCOPS	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
MID\$	-	CPL	MID\$(<výraz String>,<počáteční bod> [,<počet znaků>]) Vyjímá část výrazu STRING a vydává ji jako text. Výsledek může být předán do proměnné STRING nebo do vhodně dimenzovaného zna- kového pole. MID\$(<znakové pole>,<počáteční bod>[,<počet znaků>]) Přepisuje části znakového pole.	5-94	MID\$	-
Mirror	MIR	Zrcadlové převrácení	MIR(X1, Y1,Z1,...): Zrcadlové převrácení souřadnic zapnout MIR(): Vypnout zrcadlové převrácení souřadnic	4-71	G38/G39	
MMC	-	CPL	MMC(<CPL-Var1>[,<CPL-Var2>...[,<CPL- VarN>]...]) Odesílá informace o chodu programu z dílčího programu do klienta a čeká na výsledek z to- hoto klienta.	5-134	MMC	-
MPOS	-	CPL	MPOS(<výběr osy>[,<typ osy>[,<kanál>]]) Předává aktuálně interpolovanou požadovanou polohu, vztahenou k nulovému bodu soustavy souřadnic stroje MCS.	5-9	MPOS	-
MSG	-	-	MSG(<text pokynu>) Programování pokynů na uživatelském roz- hraní.	2-23	(MSG	
NCF	-	CPL	NCF(<NC-funkce>) Předává syntaxi aktivní NC-funkce uvnitř modální NC-skupiny <NC-funkce>.	5-30	NCF	-
Nibble	NIB	Děrování	NIB(NUM...): Zapnutí obrábění děrováním NIB(): Vypnutí obrábění děrováním	4-74	G662/G660	-
NJUST	-	CPL	Předčasné přepnutí datového výstupu zarov- naného doleva na formátovaný výstup.	5-117	NJUST	-
NOT	-	CPL	NOT <výraz> Negace výrazu BOOL, resp. bitová negace výrazu INTEGER.	2-62	NOT	-
NUL	-	CPL	<proměnná> = NUL Vymazání proměnné.	2-59	NUL	-
NUM	-	-	NUM=<hodnota> Rozděluje pojzdovou větu na definovaný počet stejně dlouhých dílčích drah.	4-76	-	-
O	O	-	O(...) Orientační pohyb pro nástroj	4-77	O(...)	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("_" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
OFFSTOPA	-	-	Příkaz synchronizace kanálu: Maže naprogramované "podmínky zastavení A" v řídicím kanálu (ASTOPA, BSTOPA, WSTOPA).	2-83		
OFFSTOPO	-	-	Příkaz synchronizace kanálu: Maže naprogramované "podmínky zastavení NEBO" v řídicím kanálu (ASTOPO, BСТОPO, WСТОPO).	2-83		
Omega	-	-	Adresa Omega	3-97		
OPENR	-	CPL	OPENR(<n>, <název prog.>[, <délka záznamu>]) Otevírá soubor pro následný čtecí přístup.	5-111	OPENR	-
OPENW	-	CPL	OPENW(<n>, <název prog.>[, <délka>][, <komentář k prog.>][, <délka záznamu>]) Otevírá soubor pro následný zapisovací přístup.	5-111	OPENW	-
OR	-	CPL	<výraz> OR <výraz> Binární operace se dvěma výrazy typu BOOL, resp. INTEGER s funkcí NEBO.	2-62	OR	-
OvrDis	OVD	Posuv 100%	Vypnutí potenciometru posuvu	4-87	G63	-
OvrEna	OVE	Posuv 100%	Zapnutí potenciometru posuvu	4-87	G66	-
P	P	-	Vyvolání podprogramu (možné také přímo bez P)	2-26		
PathAcc	PAC	Zrychl. dráhy	PAC(ACC... UP..., DOWN ...) : Zapnout PAC() : Vypnout Programování zrychlení dráhy, volitelně oddělené pro zrychlení a brzdění nebo přesahující.	4-88	G106/G107	ACC
PCS	-	CPL	PCS(<souřadnice>[, <typ výběru>]) Předává poslední naprogramovanou absolutní polohu souřadnice.	5-6	CPOS	-
PDIM	-	CPL	PDIM <název parametru>(<velikost pole>) Musí se používat, když podprogram: <ul style="list-style-type: none"> • má být vyvolán s konstantou String jako předávacím parametrem a • volající program je zvolen bez spojení. 	2-31	PDIM	-
PMT	-	CPL	PMT(<výběr souřadnice>, <kód PI>[, [<řada PI>][, [<tabulka>][, <jednotka>]]]) Čtecí a zapisový přístup do tabulky umístění XML.	5-26	-	
PmTsel	PMS	-	PMS(<název tabulky umístění>) Volba tabulky umístění	4-91	G22 ID	-
PMV	-	CPL	PMV(<výběr souřadnice>[, <řada PI>]) Udává účinné hodnoty umístění pro souřadnici.	5-25	-	

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
PolarPol	POP	PolarPol	POP(X...,Y...,Z...): Nastavení pólu POP(): Pól v počátku Pól pro programování polárních souřadnic.	4-92	-	
PoleSet	PLS	Nastavení pólu	PLS(X..., Y..., Z...): Nastavení pólu PLS(): Pól v počátku Pól pro pomůcky zadávání zrcadlového převrácení, otočení.	4-97	G37/G39	
PosDepHSOut	PHS	-	PHS(<režim>{,<vzdálenost>},{<trvání>}) Programovatelný rychloběžný výstup závislý na poloze	4-89	PDHSO	-
PosMode	PMD	Typ polohování	PMD(A..., B...) : Přepnutí typu polohování PMD(): Typ polohování podle parametrů stroje Typ polohování pro nekonečné osy	4-98	G151/G150	G36ff.
PPOS	-	CPL	PPOS(<výběr osy>[,<typ osy>]) Dotaz na skutečnou polohu osy v okamžiku sepnutí měřicího dotyku.	5-14	PPOS	-
PrecProg	PRP	Přesné programování	PRP(DIST... EPS...) : Přesné programování zapnout PRP(): Přesné programování vypnout	4-100	G328/G329	-
PRN#	-	CPL	PRN#(<n>[,<výraz>][,<výraz>][,<výraz>][,...][:]) Zápisový přístup do souboru.	5-117	PRN#	-
PROBE	-	CPL	PROBE(<výběr osy>[,<typ osy>]) Dotaz na polohu osy v okamžiku sepnutí měřicího dotyku, vzhledem k souřadnicím nulového bodu osy v MCS.	5-13	PROBE	-
PtBikEnd	PTE	Zdvih děrování	PTE(X..., Y..., Z..., ...) Spuštění zdvihu na konci věty	4-102	G612	-
PtDefault	PTD	Zdvih děrování	Spuštění zdvihu podle parametrizace	4-104	G610	-
PtInpos	PTI	Zdvih děrování	PTI(X..., Y..., Z..., ...) Spuštění zdvihu při Inpos (Pt = punch time)	4-105	G611	-
Punch	PUN	Děrování	PUN(NUM...): Zapnutí obrábění děrováním PUN(): Vypnutí obrábění děrováním	4-106	G661/G660	-
RadProg	RAD	Průměr/ poloměr	Programování poloměru	4-35	RAD	G15
RedTorque	RDT	-	RDT(<osa1>{,<osa2>{, ...}}) Stanovení momentu pro redukci momentu	4-108	G177	-
REM	-	CPL	REM <text komentáře> Komentování programu.	2-22	REM	-
RemAxis	RAX	-	RAX(...) Odstranit osu z kanálu	4-109	G512	FAX

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
RemLogName	RLN	-	RLN(...) Odstranit logický název osy	4-110	G516	-
REPEAT	-	CPL	REPEAT <procedura> UNTIL <podmínka> Smyčková konstrukce s dotazem na podmínku ukončení po prvním zpracování procedury.	2-43	REPEAT	-
REPOSTP	-	-	REPOSTP(<bod>) Asynchronní podprogramy: Definovat bod opětovného najetí v asynchronním UP	4-111		
REWRITE	-	CPL	REWRITE(<n>) Přepsání existujícího souboru.	5-120	REWRITE	-
Rotate	ROT	Otáčení	ROT(<úhel>): Zapnutí otáčení souřadnic ROT(): Vypnutí otáčení souřadnic	4-112	G38/G39	
ROTAX	-	-	Definování osy otáčení pro vektorovou orientaci	4-77	ROTAX	-
ROUND	-	CPL	<Číslo Integer> = ROUND(<výraz Real>) Konverze výrazu REAL zaokrouhlením nahoru, resp. dolů na číslo INTEGER.	2-61	Round	-
RoundEps	RNE	Fasety/zaoblení	RNE(<hodnota>): Zapnutí zaoblení RNE(): Vypnutí zaoblení Zaoblení s definovanou odchylkou kontur.	4-114	G34/G35	-
Rounding	RND	Fasety/zaoblení	RND(<hodnota>): Zapnutí zaoblení RND(): Vypnutí zaoblení Zaoblení s definovaným poloměrem zaoblení	4-115	G134/G35	-
S	S	-	S<číslo>=<hodnota> S<hodnota> Programování počtu otáček vřetena	3-98	S	
Scale	SCL	Měřítka	SCL(X..., Y..., Z..., ...): nastavení měřítka souřadnic zapnout SCL(): Vypnutí nastavení měřítka souřadnic	4-116	G38/G39	G78/G79
SCL	-	CPL	SCL(<SCL-výběr>[,<výběr>[,<typ výběru>]]) Dává pro aktuální kanál poslední naprogramované parametry funkcí PLS a ROT (souřadnice pólu, faktory měřítka a úhly otočení).	5-29	SCL	-
SCS	-	CPL	SCS(<index osy>,<typ ID>,<ID-č.>[,<výsledná proměnná>]) Čtecí přístup k parametrům pohonu SERCOS z aktivní sady parametrů.	5-31	SCS	-
SCSL	-	CPL	SCSL(<index osy>,<typ ID>,<ID-č.>,<název souboru>[,<výsledná proměnná>]) Vytvoření souboru pro seznamy parametrů SERCOS.	5-32	SCSL	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
SD	-	CPL	SD(<skupina>[,<index1>[,<index2>[,<index3>]]]) Čtení aktivních systémových dat NC-řízení.	5-34	SD	-
SDR	-	CPL	SDR(<skupina>[,<index1>[,<index2>]]]) Čtení aktivních systémových dat NC-řízení ve formátu REAL.	5-40	SDR	-
SEEK	-	CPL	SEEK(<n>,<k>[,<o>]) Umísuje ukazatel souboru na <k>-tý záznam souboru Random nebo na <k>-tý bajt sekvenčního souboru.	5-126	SEEK	-
SelCrdCouple	SCC	-	SCC(SC<kanál>, CL(Q1>,<Z1>,{...})): Zapnutí selektivního aditivního spojení souřadnic SCC(): Vypnutí všech spojení souřadnic	4-118	-	-
SETERR	-	CPL	Po vyhodnocení ERRNO generovat běhovou chybu.	2-57	-	-
SetPos	SPS	-	Nastavit pozici programu	4-119	G92	G52
SETWARN	-	CPL	Po vyhodnocení ERRNO generovat výstražné hlášení.	2-58	-	-
Shift	SHT	-	SHT(X..., Y ..., Z ..., ...): Posunutí kontur zapnout SHT(): Vypnout posunutí kontur	4-120	G60/G67	-
SIN	-	CPL	<Funkční hodnota> = SIN(<vstupní hodnota>) Použití funkce sinus na vstupní hodnotu.	2-61	SIN	-
Sin2DownFeed	S2D	V-Profil	Brzdění tvaru Sin2	4-67	G316	-
Sin2UpFeed	S2U	V-Profil	Zrychlení tvaru Sin2	4-66	G315	-
SinDownFeed	SND	V-Profil	Brzdění sinusového tvaru	4-67	G314	-
SinUpFeed	SNU	V-Profil	Zrychlení sinusového tvaru	4-66	G313	-
Smax	SMX	-	SMX(<hodnota>) Maximální počet otáček pro vřeteno.	4-121	G192	G92
Smin	SMN	-	SMN(<hodnota>) Minimální počet otáček pro vřeteno.	4-121	G292	-
SpAdmin	SPA	-	SPA(Si=0 1) Podmíněně uvolnit rezervované vřeteno nebo ho převzít jiným kanálem.	4-122	-	-
SpCouple _Wait	SPC_ WAIT	-	SPC_WAIT(CP=1..4) Čekání na synchronní provoz uvedeného svazku spojek.	4-129	-	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
SpCouple Config	SPCC	-	SPCC(CP=1..4, MA=<Master>, Si=1, Sj=1, ...) Definovat svazek spojek, vymazat nebo přidat následná větена, zrušit svazek spojek.	4-123	-	-
SpCouple PosOffs	SPCP	-	SPCP(S1=<posunutí>, S2=<posunutí>, ... {POSVEL<počet otáček>}) Definovat úhlové posunutí pro aktivní svazek spojek větена.	4-126	-	-
SpCouple PosOffs_Wait	SPCP _WAIT	-	SPCP_WAIT(CP=1..4) Čekání na úhlové posunutí pro naprogramo- vaný svazek spojek.	4-127	-	-
SpCoupleDist	SPCD	-	SPCD(S1=<vzdálenost>, S2=<vzdálenost>, ...) Definovat vzdálenost spojek pro spojky větена.	4-124	-	-
SpCoupleErr Win	SPCE	-	SPCE(S1=<okno>, S2=<okno>, ...) Definovat chybové okno synchronního chodu pro spojky větена.	4-125	-	-
SpCouple- SyncWin	SPCS	-	SPCS(S1=<okno>, S2=<okno>, ...) Definovat okno synchronního chodu pro spojky větена.	4-128	-	-
SPG	-	-	SPG<skupina>(<čísla>) SPGALL(0) Definovat/rozdělit skupiny větен	4-130		
SpindleToA- xis	STA	-	STA(...) Přepnout větена do osového provozu.	4-131		
SplineDef	SDF	-	Stanovení spline varianty.	4-132		
Split	SPT	-	Split({<režim>},{<dílčí délka>}) Rozděluje pojezdové větы na několik dílčích drah, pokud překročí určitou délku.	4-134		
SpMode	SPM	-	SPM(S1=0 1, S2=0 1, ...) Ručně přepnout rozhraní pohonu větена mezi otáčkovým a polohovým provozem	4-135		
SPOS	-	CPL	SPOS(<výběr osy>) Předává aktuální požadovanou hodnotu fy- zické osy.	5-10	SPOS	-
SPV	-	-	SPV [<perm. prom. CPL> = <jednoduchý výraz CPL>] Příkaz synchronizace kanálu: V průběhu pro- gramu je zapsána permanentní proměnná CPL.	2-77		

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
SPVE	-	-	SPV E[<perm. prom. CPL> = <výraz CPL>] Příkaz synchronizace kanálu: V průběhu programu je zapsána permanentní proměnná CPL. Výraz CPL může obsahovat příkazy CPL, ale spouští se již v okamžiku přípravy věty.	2-77		
SQRT	-	CPL	<Funkční hodnota> = SQRT(<vstupní hodnota>) Použití funkce druhé odmocniny na <vstupní hodnotu>.	2-61	SQRT	-
SSPG	-	-	SSPG<skupina>=<hodnota> Zadání počtu otáček pro skupinu vřeten, viz též "S<hodnota>"	3-98	SSPG	
StatToolOri	STO	-	STO(PHI ..., THETA ..., PSI ...) Statická orientace nástroje	4-136		
STR\$	-	CPL	STR\$([<formátový řetězec> ,] <hodnota>) Konvertuje číselný výraz <hodnota> na znakový řetězec, který může být přiřazen výhradně znakovému poli. S <formátovým řetězcem> může být znakový řetězec vydán zformátovaný.	5-97	STR\$	-
TAN	-	CPL	<Funkční hodnota> = TAN(<vstupní hodnota>) Použití funkce tangens na <vstupní hodnotu>.	2-61	TAN	-
TangTool	TTL	Orient. WZ	TTL(TAX..., SYM..., ANG..., IA..., PLC...): Zapnutí tangenciálního vedení nástroje TTL(): Vypnutí tangenciálního vedení nástroje	4-137	G131/G130	-
TangToolOri	TTO	Tangenciální orient. WZ	TTO(SYM..., ANG...): Zapnutí tangenciální orientace nástroje TTO(): Vypnutí tangenciální orientace nástroje	4-140	G630/G631	-
TappSp	TSP	-	TSP(CAXi, ..., GRPj, ...) Výběr vřetena pro řezání vnitřních závitů bez vyrovnávacího sklíčidla.	4-142	G532	SPF
TcsDef	TCS	-	Explicitně stanovit polohu soustavy souřadnic nástroje (implicitní oprava délky nástroje ve spojení s odpovídající transformací os).	4-143	TcsDef	-
TCV	-	CPL	TCV(<výběr hodnot>[,<výběr opravy>]) Dává poslední naprogramované hodnoty opravy nástroje.	5-20	TC	-
ThreadSet	TST	Závit	Řezání závitů, přídavné funkce	4-146		
TIME	-	CPL	<Proměnná String> = TIME Přiřazuje proměnné STRING čas ve formě HH.MM.SS.	5-46	TIME	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
Trans	TRS	Posunutí programových souřadnic	TRS(X ..., Y ..., Z ..., ...) : Zapnutí posunutí programových souřadnic TRS() : Vypnutí posunutí programových souřadnic	4-148	G60/G67	
TRIM\$	-	CPL	TRIM\$(<řetězec znaků>) TRIM\$(<řetězec znaků>,"L") TRIM\$(<řetězec znaků>,"R") Dává při přiřazení oblasti znakového pole k proměnné STRING nebo ke znakovému poli znakový řetězec bez mezer vepředu (index "L"), resp. vzadu (index "R"). Bez indexu se vynechávají mezery vepředu i vzadu.	5-100	TRIM\$	-
TRUE	-	CPL	<Proměnná Boolean> = TRUE Pravdivostní hodnota proměnné BOOL	2-52	TRUE	-
VAL	-	CPL	VAL(<výraz String>) Vrací číselnou hodnotu výrazu STRING.	5-99	VAL	-
VERSINF\$	-	CPL	VERSINF\$(<index1>[,<index2>]) Vyvolává administrativní data, např. SW verzi.	5-114		
VirtAxisPos	VAP	-	Nastavuje polohu virtuálních synchronních os v aktuálním kanálu.	4-150	-	-
WAIT	-	-	WAIT v NC-větě (jako CPL WAIT)	2-68		
WAIT	-	CPL	WAIT(bez parametru) Zastavuje zpracování vět, dokud nejsou zpracovány všechny věty naprogramované před příkazem WAIT. WAIT(<čekací doba>) Zastavuje zpracování vět, dokud neuplyne stanovený časový interval. WAIT(BITIF(...)) Zastavuje zpracování vět, dokud na bitovém rozhraní SPS-NC nenastane určitý stav.	2-68	WAIT	-
WAITA	-	-	WAITA[BITIF(<parametr>){=<stav>} {,...},{<timeout>}] Příkaz synchronizace kanálu: Čeká se, až budou vydány všechny dotazované signály rozhraní nebo uplyne timeout.	2-73	WAITA	
WaitAxis	WAX	-	WAX(...) Čekání, až bude osa uvolněna a následně převzata do kanálu.	4-151	G511	GAX ?
WAITO	-	-	WAITO[BITIF(<parametr>){=<stav>} {,...},{<timeout>}] Příkaz synchronizace kanálu: Čeká se, až bude vydán jeden z dotazovaných signálů rozhraní nebo uplyne timeout.	2-73	WAITO	

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
WHILE	-	CPL	WHILE <podmínka> DO <procedura> END Smyčková konstrukce s dotazem na podmínku ukončení před prvním proběhnutím smyčky.	2-44	WHILE	-
WPOS	-	CPL	WPOS(<výběr osy>[,<typ výběru>[,<kanál>]]) Předává interpolovanou požadovanou polohu, vzhledem k nulovému bodu obrobku v aktuální WCS.	5-8	WPOS	-
WPV	-	-	WPV[<perm. prom. CPL><relační op- erátor><jednoduchý výraz CPL>[,<timeout>] Příkaz synchronizace kanálu: Čeká se, až bude splněn výraz nebo uplyne timeout.	2-75		
WPVE	-	-	WPVE[<perm. prom. CPL><relační op- erátor><výraz CPL>[,<timeout>]] Příkaz synchronizace kanálu: Čeká se, až bude splněn výraz nebo uplyne timeout. Výraz CPL může obsahovat příkazy CPL, ale spouští se již v okamžiku přípravy věty.	2-75		
Writeld	WID	-	WID(S-0-0104, X..., Y..., Z..., ...) Zápis parametrů SERCOS	4-152	G900	AXD
WSTOPA	-	-	WSTOPA[<č. kanálu>,<podmínka1>{,<podmínka2> {...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Zastavení řídicího kanálu, dokud nejsou splněny všechny podmínky. Podmínky se vztahují k polohám obrobku, např. "Z" > 20.	2-81		
WSTOPO	-	-	WSTOPO[<č. kanálu>,<podmínka1>{,<podmínka2> {...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Řídicí kanál je zastaven, dokud není splněna jedna z podmínek. Podmínky se vztahují k polohám obrobku, např. "Z" > 20.	2-81		
XOR	-	CPL	<výraz> XOR <výraz> Binární operace se dvěma výrazy typu BOOL, resp. INTEGER s funkcí EXKLUZIVNÍ NEBO.	2-62	XOR	-
XTAB	-	CPL	XTAB(<tabulka>,<dílčí struktura> Čtecí a zápisový přístup do libovolné tabulky XML, která může být definována také uživatelem.	5-28	TDA	-
ZOT	-	CPL	ZOT(<výběr sloupce>,<kód NPV>[,<řada NPV>][,<tabulka>][,<jednotka>]]) Čtecí a zápisový přístup do libovolné tabulky posunutí nulových bodů XML.	5-18	FXC	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
ZoTsel	ZOS	-	ZOS(<název NPV> Volba tabulky posunutí nulových bodů	4-155	G22 V	O
ZOV	-	CPL	ZOV(<výběr osy>[.<řada NPV>]) Udává účinné hodnoty posunutí nulového bodu pro osu / souřadnici stroje.	5-18	FXC	-

Příloha

Alfanumericky seřazené podle skupin:

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Strana	Funkce v PNC MTC	
;	-	-	Komentář: vynechat zbytek řádku	2-21		
()	-	-	Komentář: vynechat obsah závorky	2-21		
(MSG	-	-	vydat text hlášení	2-23		
/	-	-	Vynechat větu	2-38		
//	-	-	Komentář	2-21		
AC(...)	-	-	Lokální programování absolutního rozměru, např. X=AC(10)	3-69	AC(...)	-
Area	ARA	-	Area(<BNr>,<Sta>{,<Mod>,{<P1>},{<P2>},{<D 1>},{<D2>}}) Definuje, aktivuje nebo deaktivuje až 10 ob- délníkových dvojrozměrných mrtvých nebo pracovních oblastí s osově paralelními me- zemi.	4-2	-	-
ASPCLR	-	-	ASPCLR(<Up-č.>) Asynchronní podprogramy: odhlásit	4-5	-	-
ASPDIS	-	-	ASPDIS(<Up-č.>) Asynchronní podprogramy: vypnout	4-6	-	-
ASPENA	-	-	ASPENA(<Up-č.>) Asynchronní podprogramy: zapnout	4-6	-	-
ASPRTP	-	-	ASPRTP(<Up-č.>,<bod>) Asynchronní podprogramy: definovat bod op- ětovného najetí	4-7	-	-
ASPSET	-	-	ASPSET(<Up-č.>,<Up-název>{,<příznaky>}) Asynchronní podprogramy: přihlásit	4-8	-	-
ASPSTA	-	-	ASPSTA(<Up-č.>,<kanál-č.>) Asynchronní podprogramy: spustit	4-9	-	-
AssLogName	ALN	-	ALN(...) Přiřadit logický název osy	4-10	G515	-
ASTOPA	-	-	ASTOPA[<č. kanálu>,<podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile jsou splněny všechny podmínky, zastaví se řídicí kanál. Podmínky se vztahují k polohám os, např. "Z" > 20	2-78	-	-
ASTOPO	-	-	ASTOPO[<č. kanálu>,<podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile je splněna jedna podmínka, zastaví se řídicí kanál. Podmínky se vztahují k polohám os, např. "Z" > 20	2-78	-	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
ATCAL	-	-	ATCAL(<soubor>,<OptData>,<maska> {,<Info>},{,<AnzIt>}) Kalibrovat kinematiky os: Optimalizovat para- metry:	4-11		
ATFWD	-	-	ATFWD(<Koord>,<AxKoord>{,<ParData>}) Kalibrovat kinematiky os: Přepočítat parametry	4-13		
ATGET	-	-	ATGET(<ParData>{,<ATrafNr>}) Kalibrovat kinematiky os: Načíst parametry z NC	4-14		
ATPUT	-	-	ATPUT(<ParData>{,<ATrafNr>}) Kalibrovat kinematiky os: Zapsat parametry do NC	4-15		
AUXFUNC	-	-	Provést aktivní pomocné funkce všech skupin	4-16		
AxAccSave	AAS	-	Uložit aktuální zrychlení os.	4-17	-	-
AxisToS- pindle	ATS	-	ATS(<název osy>) Přepnout osu na provoz vřetena.	4-22		
BlkNmb	BNB	-	BlkNmb(<Anz>) Omezuje počet programových vět, které mo- hou být načteny a brány v úvahu přípravou vět.	2-71	PREPNUM	-
BSTOPA	-	-	BSTOPA[<č. kanálu>, <podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile jsou splněny všechny podmínky, zastaví se řídicí kanál. Podmínky se vztahují k základním po- lohám obrobku, např. "Z" > 20	2-80	-	-
BSTOPO	-	-	BSTOPO[<č. kanálu>, <podmínka1> {,<podmínka2>}{...{,<podmínka8>}...}] Příkaz synchronizace kanálu: Jakmile je splněna jedna podmínka, zastaví se řídicí kanál. Podmínky se vztahují k základním po- lohám obrobku, např. "Z" > 20	2-80	-	-
COFFS	-	-	Offset kontury pro opravu 3D poloměru	3-79	-	-
DcTsel	DCS	-	DCS({<cesta>}<název souboru>) Volba tabulky D-korektur.	4-33	G22 K	-
DefAxis	DAX	-	Vytvořit standardní konfiguraci os	4-34	G513	-
DistCtrl	DCR	-	DCR(1) Regulace výšky pro digitalizaci, zapnutí podle parametrů stroje DCR(<funkce>) zapnutí s vlastními konfigu- račními daty DCR() vypnutí regulace výšky	4-37	DistCtrl	-
EndPosCouple	EPC	-	EPC(<M-koord>,<S-koord>,<faktor>): Zapnutí spojení koncové polohy EPC(): Vypnutí spojení koncové polohy	4-39	-	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
F	F	-	F-adresa pro G93, G94, G95	3-96	F	
FA	FA	-	Rychlost pro asynchronní osy.	3-97	FA	
FsMove	FSM	-	Pojezd do pevného dorazu (Fs = fixed stop)	4-48	G475	G75
FsProbe	FSP	-	Měření na pevném dorazu	4-46	G475	G75
FsReset	FSR	-	Odstranění pevného dorazu	4-48	G477	G76
FsTorque	FST	-	Moment pro pevný doraz	4-48	G476	AXD
G	-	-	Uživatelsky definované lokální podprogramy			
G04	G4	-	G4(F..) Doba prodlevy v sekundách G4(S...) Doba prodlevy v otáčkách vřetena, vztažným vřetenem je hlavní vřeteno (MainSp)	3-12	G04/G104	G04
G74	-	-	G74 X1 Y1 Z1 ... Najetí na souřadnice referenčního bodu	3-55	G74/G374	G74
G74(Home)	-	-	G74(HOME) X1 Y1 Z1 Najetí do referenčního bodu (skutečné referen- cování, také pro asynchronní osy)	3-55	G74/G374	G74
G76	-	-	Najetí do pevné polohy stroje (souřadnice stroje)	3-58	G76	-
G77	-	-	G77 <souřadnice 1><režim> <souřadnice n><režim> ... F<hodnota> Alternativní syntaxe: REPOS Asynchronní podprogramy: Změna polohy jednotlivých souřadnic	3-59	-	G77
GetAxis	GAX	-	GAX(...) Převzít volnou osu do kanálu	4-51	G510	
GMSG	-	-	GMSG(<text pokynu>) Programování pokynů na uživatelském roz- hraní.	2-23	(GMSG	
GoAhead	GOA	-	Skok dopředu	2-33		
GoBack	GOB	-	Skok dozadu	2-34		
GoCond	GOC	-	Podmíněný skok	2-35		
GoTo	-	-	Nepodmíněný skok	2-36		
HsBlkSwitch	HSB	-	HSB(HS<X>=<Y>) Průběžná změna vět pomocí signálu High- Speed	4-52	G575	-
HsBlkSwitch(. .HSSTOP=..)	HSB	-	HSB(...,HSSTOP=..) Změna vět s přerušením pomocí signálu High- Speed	4-55	G575	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
HWOC	-	-	HWOC{{CHAN<č. kanálu>},CRDNO <č. souřadnice>,{STEP<Inkr>}} Zapnutí online opravy v souřadnicích obrobku HWOCDIS{{CHAN<č. kanálu>}} Vypnutí online opravy, uložit hodnoty HWO() Vypnutí online opravy, vymazat hodnoty	4-57	HWO- CON, HWO- COFF	-
IC(...)	-	-	Lokální programování relativních rozměrů, např. X=IC(5)	3-69	IC(...)	-
InitMeas	IME	-	IME(MpiAxis i) Inicializovat průběžné měření.	4-44	G175	-
INSDEP	-	-	Hloubka ponoru, oprava 3D poloměru.	3-79	-	-
JogWCSselect	-	-	Vybírá souřadnice pro krokování v souřadnicích obrobku (seřizovací provoz).	4-59	JogWCS- Select	-
LEN	-	-	LEN=<hodnota> Rozděluje pojezdovou větu na několik stejně dlouhých dílčích drah.	4-62	-	-
LFCConf	LFC	-	LFC{{LL(...)}} Parametrizovat řízení laserového výkonu	4-64	LFPON	-
LFP	LFP	-	LFP, LFP(1), LFP{{LL(...)}} Zapnout řízení laserového výkonu LFP(0) Vypnout řízení laserového výkonu	4-64	LFPON, LFPOFF	-
LinModZp	LMZ	-	LMZ{{LinModAxis i}} Vynulování lineární modulu osy (Zp = zero point)	4-69	G105	-
m	m	-	Uživatelsky definované lokální podprogramy			
M0	M0	-	Zastavení programu	3-86	M0	
M1	M1	-	Podmíněné zastavení programu	3-86	M1	
M19, M119, M219	-	-	Seřízení vřetena.	3-91	M19-M219	
M2	M2	-	Konec hlavního programu	3-87	M2	
M3, M103, M203, M13, M113, M213	-	-	Pravotočivý chod vřetena, Pravotočivý chod vřetena se zapnutou chladicí kapalinou	3-88	M3 - M203, M13-M213	
M30	M30	-	Konec hlavního programu	3-87	M30	
M4, M104, M204, M14, M114, M214	-	-	Levotočivý chod vřetena, Levotočivý chod vřetena se zapnutou chladicí kapalinou	3-89	M4 - M204, M14-M214	

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
M40, M140, M240	-	-	automatická volba převodových stupňů	3-93	M40-M240	
M41 ... M44, M141 ... M144, M241 ... M244	-	-	Ruční volba převodových stupňů	3-94	M41 - M44, M141-M14 4 M241-M24 4	
M48, M148, M248	-	-	Vyřadit převodový stupeň.	3-95	M48-M248	
M5, M105, M205	-	-	Zastavení vřetena	3-90	M5 - M205	
MainSp	MSP	-	Výběr hlavního vřetena např. pro G33 a G95	4-70	MAINSP	SPF
MSG	-	-	MSG(<text pokynu> Programování pokynů na uživatelském roz- hraní.	2-23	(MSG	
NUM	-	-	NUM=<hodnota> Rozděluje pojezdovou větu na definovaný počet stejně dlouhých dílčích drah.	4-76	-	-
O	O	-	O(...) Orientační pohyb pro nástroj	4-77	O(...)	-
OFFSTOPA	-	-	Příkaz synchronizace kanálu: Maže naprogramované "podmínky zastavení A" v řídicím kanálu (ASTOPA, BSTOPA, WSTOPA).	2-83		
OFFSTOPO	-	-	Příkaz synchronizace kanálu: Maže naprogramované "podmínky zastavení NEBO" v řídicím kanálu (ASTOPO, BSTOPO, WSTOPO).	2-83		
Omega	-	-	Adresa Omega	3-97		
P	P	-	Vyvolání podprogramu (možné také přímo bez P)	2-26		
PmTsel	PMS	-	PMS(<název tabulky umístění> Volba tabulky umístění	4-91	G22 ID	-
PosDepHSOut	PHS	-	PHS(<režim>{,<vzdálenost>}{,<trvání>}) Programovatelný rychloběžný výstup závislý na poloze	4-89	PDHSO	-
RedTorque	RDT	-	RDT(<osa1>{,<osa2>{, ...}}) Stanovení momentu pro redukci momentu	4-108	G177	-
RemAxis	RAX	-	RAX(...) Odstranit osu z kanálu	4-109	G512	FAX
RemLogName	RLN	-	RLN(...) Odstranit logický název osy	4-110	G516	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("_" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
REPOSTP	-	-	REPOSTP(<bod>) Asynchronní podprogramy: Definovat bod opětovného najetí v asynch- ronním UP	4-111		
ROTAX	-	-	Definování osy otáčení pro vektorovou orien- taci	4-77	ROTAX	-
S	S	-	S<číslo>=<hodnota> S<hodnota> Programování počtu otáček vřetena	3-98	S	
SeICrdCouple	SCC	-	SCC(SC<kanál>, CL(Q1>,<Z1>,{...})): Zapnutí selektivního aditivního spojení souřadnic SCC(): Vypnutí všech spojení souřadnic	4-118	-	-
SetPos	SPS	-	Nastavit pozici programu	4-119	G92	G52
Shift	SHT	-	SHT(X..., Y ..., Z ..., ...): Posunutí kontur zapnout SHT(): Vypnout posunutí kontur	4-120	G60/G67	-
Smax	SMX	-	SMX(<hodnota>) Maximální počet otáček pro vřeteno.	4-121	G192	G92
Smin	SMN	-	SMN(<hodnota>) Minimální počet otáček pro vřeteno.	4-121	G292	-
SpAdmin	SPA	-	SPA(Si=0 1) Podmíněně uvolnit rezervované vřeteno nebo ho převzít jiným kanálem.	4-122	-	-
SpCouple _Wait	SPC_ WAIT	-	SPC_WAIT(CP=1..4) Čekání na synchronní provoz uvedeného svazku spojek.	4-129	-	-
SpCouple Config	SPCC	-	SPCC(CP=1..4, MA=<Master>, Si=1, Sj=1, ...) Definovat svazek spojek, vymazat nebo přidat následná vřetena, zrušit svazek spojek.	4-123	-	-
SpCouple PosOffs	SPCP	-	SPCP(S1=<posunutí>, S2=<posunutí>, ... {POSVEL<počet otáček>}) Definovat úhlové posunutí pro aktivní svazek spojek vřetena.	4-126	-	-
SpCouple PosOffs_Wait	SPCP_ _WAIT	-	SPCP_WAIT(CP=1..4) Čekání na úhlové posunutí pro naprogramo- vaný svazek spojek.	4-127	-	-
SpCoupleDist	SPCD	-	SPCD(S1=<vzdálenost>, S2=<vzdálenost>, ...) Definovat vzdálenost spojek pro spojky vřetena.	4-124	-	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
SpCoupleErr Win	SPCE	-	SPCE(S1=<okno>, S2=<okno>, ...) Definovat chybové okno synchronního chodu pro spojky vřeten.	4-125	-	-
SpCouple- SyncWin	SPCS	-	SPCS(S1=<okno>, S2=<okno>, ...) Definovat okno synchronního chodu pro spojky vřeten.	4-128	-	-
SPG	-	-	SPG<skupina>(<čísla>) SPGALL(0) Definovat/rozdělit skupiny vřeten	4-130		
SpindleToA- xis	STA	-	STA(...) Přepnout vřeten do osového provozu.	4-131		
SplineDef	SDF	-	Stanovení spline varianty.	4-132		
Split	SPT	-	Split({<režim>},{<dílčí délka>}) Rozděluje pojezdové věty na několik dílčích drah, pokud překročí určitou délku.	4-134		
SpMode	SPM	-	SPM(S1=0 1, S2=0 1, ...) Ručně přepnout rozhraní pohonu vřeten mezi otáčkovým a polohovým provozem	4-135		
SPV	-	-	SPV [<perm. prom. CPL> = <jednoduchý výraz CPL>] Příkaz synchronizace kanálu: V průběhu programu je zapsána permanentní proměnná CPL.	2-77		
SPVE	-	-	SPV E[<perm. prom. CPL> = <výraz CPL>] Příkaz synchronizace kanálu: V průběhu programu je zapsána permanentní proměnná CPL. Výraz CPL může obsahovat příkazy CPL, ale spouští se již v okamžiku přípravy věty.	2-77		
SSPG	-	-	SSPG<skupina>=<hodnota> Zadání počtu otáček pro skupinu vřeten, viz též "S<hodnota>"	3-98	SSPG	
StatToolOri	STO	-	STO(PHI ..., THETA ..., PSI ...) Statická orientace nástroje	4-136		
TappSp	TSP	-	TSP(CAXi, ..., GRPj, ...) Výběr vřeten pro řezání vnitřních závitů bez vyrovnávacího sklíčidla.	4-142	G532	SPF
TcsDef	TCS	-	Explicitně stanovit polohu soustavy souřadnic nástroje (implicitní oprava délky nástroje ve spojení s odpovídající transformací os).	4-143	TcsDef	-
VirtAxisPos	VAP	-	Nastavuje polohu virtuálních synchronních os v aktuálním kanálu.	4-150	-	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
WAIT	-	-	WAIT v NC-větě (jako CPL WAIT)	2-68		
WAITA	-	-	WAITA[BITIF(<parametr>){=<stav>} {...}{, <timeout>}] Příkaz synchronizace kanálu: Čeká se, až bu- dou vydány všechny dotazované signály roz- hraní nebo uplyne timeout.	2-73	WAITA	
WaitAxis	WAX	-	WAX(...) Čekání, až bude osa uvolněna a následně převzata do kanálu.	4-151	G511	GAX ?
WAITO	-	-	WAITO[BITIF(<parametr>){=<stav>} {...}{, <timeout>}] Příkaz synchronizace kanálu: Čeká se, až bude vydán jeden z dotazovaných signálů roz- hraní nebo uplyne timeout.	2-73	WAITO	
WPV	-	-	WPV[<perm. prom. CPL><relační op- erátor><jednoduchý výraz CPL>{, <timeout>} Příkaz synchronizace kanálu: Čeká se, až bude splněn výraz nebo uplyne timeout.	2-75		
WPVE	-	-	WPVE[<perm. prom. CPL><relační op- erátor><výraz CPL>{, <timeout>}] Příkaz synchronizace kanálu: Čeká se, až bude splněn výraz nebo uplyne timeout. Výraz CPL může obsahovat příkazy CPL, ale spouští se již v okamžiku přípravy věty.	2-75		
Writeld	WID	-	WID(S-0-0104, X..., Y..., Z..., ...) Zápis parametrů SERCOS	4-152	G900	AXD
WSTOPA	-	-	WSTOPA[<č. kanálu>, <podmínka1>{, <podmínka2>} {...{, <podmínka8>}...}] Příkaz synchronizace kanálu: Zastavení řídícího kanálu, dokud nejsou splněny všechny podmínky. Podmínky se vztahují k polohám obrobku, např. "Z" > 20.	2-81		
WSTOPO	-	-	WSTOPO[<č. kanálu>, <podmínka1>{, <podmínka2>} {...{, <podmínka8>}...}] Příkaz synchronizace kanálu: Řídící kanál je zastaven, dokud není splněna jedna z podmínek. Podmínky se vztahují k polohám obrobku, např. "Z" > 20.	2-81		
ZoTSel	ZOS	-	ZOS(<název NPV>) Volba tabulky posunutí nulových bodů	4-155	G22 V	O
G140	-	Oprava 3D po- loměru	Vypnutí opravy 3D poloměru	3-79	G141	

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G141	-	Oprava 3D poloměru	Oprava 3D poloměru vlevo od kontury	3-79	G141	
G142	-	Oprava 3D poloměru	Oprava 3D poloměru vpravo od kontury	3-79	G142	
G90	-	Abs/Rel	Programování absolutních rozměrů	3-69	G90	G90
G91	-	Abs/Rel	Programování relativních rozměrů	3-69	G91	G91
AxAcc	AAC	Zrychlení os	AAC(X..., Y..., Z..., ...): Zapnout AAC(1): Zapnout s uloženým nastavením AAC(): Vypnout Programovat zrychlení os	4-17	G6/G7	-
AxCouple	AXC	Spojení os	AXC(<Master>,<Slave1>(…),<Slave2>(…),…): Aktivovat spojení os AXC(): Deaktivovat spojení os	4-19	G590/G591	
CoupleSpline- Tab	CST	Spojení os	CST(STAB(...)) Vytvoření tabulky spline spojení	4-32	G582	
Coord	CRD	Transformace os	CRD(x): Zapnout transformaci os CRD(): Vypnout transformaci os např. transformace čelní plochy	4-30	Coord	G30ff.
PathAcc	PAC	Zrychl. dráhy	PAC(ACC... UP..., DOWN ...): Zapnout PAC(): Vypnout Programování zrychlení dráhy, volitelně oddělené pro zrychlení a brzdění nebo přesahující.	4-88	G106/G107	ACC
G40	-	Oprava dráhy	Vypnutí opravy dráhy frézy	3-39	G40	G40
G41	-	Oprava dráhy	Oprava dráhy frézy vlevo od obrobku	3-39	G41	G41
G42	-	Oprava dráhy	Oprava dráhy frézy vpravo od obrobku	3-39	G42	G42
G08	G8	Strmost dráhy	Zapnutí strmosti dráhy se SHAPE dráhy podle parametrů stroje	3-21	G08	G08
G08(SHAPE...)	-	Strmost dráhy	Zapnutí strmosti dráhy s naprogramovaným SHAPE dráhy	3-23	G108	G08
G09	G9	Strmost dráhy	Vypnutí strmosti dráhy (bez SHAPE)	3-21	G09	G09

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G09(..)	G9(..)	Strmost dráhy	G9(X..., Y..., ...) Vypnutí strmosti dráhy s osově naprogramovaným pořadím SHAPE G9(ASHAPE) Vypnutí strmosti dráhy s osovým pořadím SHAPE podle parametrů stroje G9(SHAPE ...) Vypnutí strmosti dráhy s naprogramovaným SHAPE dráhy G9(SIN ...) Vypnutí strmosti dráhy se SHAPE tvaru \sin^2 (možné pořadí SHAPE: 5, 10, 15, 20, 40)	3-23	G408, G608	
G184	-	Vrtací cyklus	Vrtací cyklus: Řezání vnitřních závitů bez vyrovnávacího sklíčidla	3-68	G184	
G80	-	Vrtací cyklus	Vypnout vrtací cyklus	3-61	G80	-
G81	-	Vrtací cyklus	Vrtací cyklus: Vrtání se zpětným pohybem v rychloběhu	3-62	G81	DEFINE
G82	-	Vrtací cyklus	Vrtací cyklus: Vrtání se zpětným pohybem v posuvu	3-62	G82	DEFINE
G83	-	Vrtací cyklus	Vrtací cyklus: Vrtání hlubokých děr	3-64	G83	DEFINE
G84	-	Vrtací cyklus	Vrtací cyklus: Řezání vnitřních závitů s vyrovnávacím sklíčidlem	3-65	G84	DEFINE
G85	-	Vrtací cyklus	Vrtací cyklus: Vyvrtávání se zpětným pohybem v rychloběhu	3-66	G85	DEFINE
G86	-	Vrtací cyklus	Vrtací cyklus: Vyvrtávání se zpětným pohybem v posuvu	3-67	G86	DEFINE
ABS	-	CPL	ABS(<vstupní hodnota>) Vrací vstupní hodnotu jako absolutní hodnotu.	2-60	ABS	-
ACOS	-	CPL	<Funkční hodnota> = ACOS(<vstupní hodnota>) Použití funkce arcuscosinus na <vstupní hodnotu>.	2-61	ACOS	-
AND	-	CPL	<Výtisk1> AND <Výtisk2> Binární operace se dvěma výrazy typu BOOL, resp. INTEGER	2-62	AND	-
APOS	-	CPL	APOS(<volba osy>) Předává aktuální skutečnou hodnotu osy vzhledem k nulovému bodu stroje.	5-11	APOS	-
ASC	-	CPL	ASC(<znakový řetězec>) Vydává pořadové číslo prvního znaku (ASCII-kód) ze <znakového řetězce> jako hodnotu INTEGER.	5-97	ASC	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
ASIN	-	CPL	<Funkční hodnota> = ASIN(<vstupní hodnota>) Použití funkce arcussinus na <vstupní hodnotu>.	2-61	ASIN	-
ATAN	-	CPL	<Funkční hodnota> = ATAN(<vstupní hodnota>) Použití funkce arcustangens na <vstupní hodnotu>.	2-61	ATAN	-
AXO	-	CPL	AXO(<volba osy>[,<typ volby>]) Předává aktivní posunutí G92 pro souřadnici.	5-15	AXO	-
AXP	-	CPL	AXP(číslo osy>,<informace o dráze>) Tato funkce umožňuje dílčí a měřicí programy nezávislé na rovině (naprogramujte AXP místo hodnoty adresy).	5-43	AXP	-
BCD	-	CPL	<Hodnota BCD> = BCD(<binární hodnota>) Konvertovat formát BCD na binární formát.	2-62	BCD	-
BIN	-	CPL	<Binární hodnota> = BIN(<hodnota BCD>) Konvertovat binární formát na formát BCD.	2-62	BIN	-
BITIF	-	CPL	BITIF(<bitový signál>[,<index>[,<jednotka IF>]]) Přístup k digitálnímu rozhraní mezi NC a SPS.	5-44	IC	-
CALL	-	CPL	CALL <číslo programu> [<předávací parametr1>,...] [DIN] Vyvolání podprogramu z programu CPL	2-29	CALL	-
CASE	-	CPL	CASE <Integer-výraz> OF LABEL <Int.-konstanta>[,<další Int.-konstanta>] [: <instrukce>] <Instrukce> LABEL ... OTHERWISE <instrukce> <Instrukce> ENDCASE Podmíněný výběr z několika alternativ.	2-40	CASE	-
CHR\$	-	CPL	CHR\$(<Integer-výraz>) Dává znak, jehož pořadové číslo v tabulce ASCII je stejné jako hodnota vydaná parametrem <INTEGER-výraz>.	5-97	CHR\$	-
CLOCK	-	CPL	<Funkční hodnota> = CLOCK Dotaz počítadla času v milisekundách.	5-46	CLOCK	-
CLOSE	-	CPL	CLOSE(<n>) Zavírá otevřený soubor po ukončení čtecích nebo zápisových operací.	5-123	CLOSE	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
CLRWARN	-	CPL	Vymazat všechna výstražná hlášení kanálu generovaná pomocí SETWARN.	2-58	-	-
COF	-	CPL	COF(<volba osy>[,<typ volby>]) Dává pro aktuální kanál poslední naprogramované posunutí kontury (Shift) souřadnice.	5-16	COF	-
COS	-	CPL	<Funkční hodnota> = COS(<vstupní hodnota>) Funkce kosinus pro <vstupní hodnotu>.	2-61	COS	-
CPROBE	-	CPL	CPROBE(<výběr>[,<typ výběru>]) Načítá naměřenou hodnotu vždy pro jednu souřadnici	5-12	CPROBE	-
DATE	-	CPL	<Proměnná String> = DATE Přiřazuje proměnné <STRING> datum ve formě DD.MM.	5-46	DATE	-
DBSEA	-	CPL	DBSEA(<DbTab>,<Key1>,<Key2>,<Search-Cond>,<SeachRes>[,<ResVar>]) Vyhledává datové záznamy v databázové tabulce.	5-23		
DBTAB	-	CPL	DBTAB(<DbTabelle>,<Key1>,<Key2>[,<Res-Var>]) Načítá kompletní datový záznam nebo dílčí strukturu databázové tabulky do proměnné CPL nebo ji z ní vypisuje.	5-22		
DCT	-	CPL	DCT(<výběr hodnoty>,<datový záznam>[,<tabulka>] [,<jednotka>]) Čtecí a zápisový přístup do libovolné tabulky D-korektur nebo k externím hodnotám oprav nástroje.	5-21	TC	-
DIM	-	CPL	DIM <název proměnné>(<velikost pole1>[,<velikost pole2>]) Stanovení velikosti pole (dimenzování) proměnných ARRAY s konstantami INTEGER.	2-53, 5-93	DIM	-
DIRCR	-	CPL	DIRCR(<adresář>) Vytváří nový adresář.	5-115		
DIRDEL	-	CPL	DIRDEL(<adresář>) Maže prázdný adresář.	5-116		
DIRINF	-	CPL	DIRINF(4,[<index2>],<Ergebnisvar>[,<číslo souboru>]) Vývolává data správy systému souborů.	5-114		
DPC	-	CPL	DPC(<výběr>[,<typ výběru>]) Dává pro aktuální kanál poslední naprogramované parametry opravy polohy obrobku BcsCorr souřadnice (hodnoty posunutí a úhlu otočení).	5-17	DPC	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
EOF	-	CPL	EOF(<n> Kontrola konce souboru.	5-123	EOF	-
ERASE	-	CPL	ERASE(<identifikace programu> Vymaže soubory.	5-130	ERASE	-
ERRNO	-	CPL	Chybové hlášení různých funkcí CPL.	2-56	-	-
FALSE	-	CPL	<Proměnná BOOL> = FALSE Pravdivostní hodnota proměnné BOOL	2-52	FALSE	-
FILEACCESS	-	CPL	FILEACCESS(<název souboru> Určuje, zda existuje soubor a jaká má přístupová oprávnění.	5-131	FILEAC- CESS	-
FILECOPY	-	CPL	Kopírovat soubory.	5-133	-	-
FILEDATE	-	CPL	FILEDATE(<název souboru>[,<režim>]) Zjišťuje datum/čas souboru.	5-132	FILEDATE	-
FILENO	-	CPL	Dává další platné logické číslo souboru.	5-113		
FILEPOS	-	CPL	FILEPOS(<n>[,<režim>]) Dává číslo aktuálního záznamu a offset záznamu souboru Random. Dává u sekvenčních souborů aktuální bajtovou pozici ukazatele souboru.	5-124	FILEPOS	-
FILESIZE	-	CPL	FILESIZE(<n>[,<k>]) Dává velikost souboru nebo mez, po kterou je již soubor zapsán.	5-128	FILESIZE	-
FOR NEXT	-	CPL	FOR <číselná prom.>=<počáteční hodnota> [STEP <velikost kroku>] TO <koncová hod- nota><procedura> NEXT [<číselná proměnná>] Smyčková konstrukce s automaticky běžícím počítadlem.	2-42	FOR NEXT	-
GETERR	-	CPL	GETERR(<kanál> [,<kategorie>],<chyba>-č.> [,<počet>]) Udává pro aktuální chybu č. chyby, č. kanálu a kategorii chyby.	5-47	GETERR	-
GOTO	-	CPL	GOTO <cíl skoku> Nepodmíněně programové skoky na číslo řádku, číslo věty nebo návěští.	2-37	GOTO	-
IF ENDIF	-	CPL	IF <podmínka> THEN <procedura> [ELSE <al- ternativní procedura>] ENDIF Podmíněný skok na proceduru nebo alterna- tivní proceduru.	2-39	IF ENDIF	-
INP#	-	CPL	INP#(<n>,<proměnná>[,<proměnná>] [...] [:]) Čtecí přístup do souboru.	5-121	INP#	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
INSTR	-	CPL	INSTR(<znakový řetězec>,<String- výraz>[,<počáteční bod>]) Prohledává výraz <String> od <počátečního bodu> s ohledem na <znakový řetězec> a udává jeho počáteční polohu jako hodnotu IN- TEGER.	5-96	INSTR	-
INT	-	CPL	<Číslo Integer>=INT(<výraz Real>) Konvertuje <výraz Real> na <číslo Integer> odříznutím míst za desetinnou tečkou.	2-60	INT	-
LEN	-	CPL	LEN(<výraz String>) Udává počet znaků výrazu STRING.	5-96	LEN	-
LJUST	-	CPL	Přepnutí na datový výstup zarovnaný doleva, účinný až do konce běhu programu.	5-117	LJUST	-
MCODS	-	CPL	MCODS(<typ>,<kanál>,<verze>,<vyrovnávací paměť>,<velikost>[,<P1>]) Vyvolává datové služby Motion Control v NCS. Tak je možné načítat data a stavy z NC.	5-52	MCODS	-
MCOPS	-	CPL	MCOPS(<funkce>,<kanál>[[,<P1>][,<P2>], [<P3>]][,<P4>]) Vyvolává procesové služby Motion Control v NCS. Tak lze řídit kanály v NC.	5-82	MCOPS	-
MID\$	-	CPL	MID\$(<výraz String>,<počáteční bod> [,<počet znaků>]) Vyjímá část výrazu STRING a vydává ji jako text. Výsledek může být předán do proměnné STRING nebo do vhodně dimenzovaného zna- kového pole. MID\$(<znakové pole>,<počáteční bod>[,<počet znaků>]) Přepisuje části znakového pole.	5-94	MID\$	-
MMC	-	CPL	MMC(<CPL-Var1>[,<CPL-Var2>...[,<CPL- VarN>]...]) Odesílá informace o chodu programu z dílčího programu do klienta a čeká na výsledek z to- hoto klienta.	5-134	MMC	-
MPOS	-	CPL	MPOS(<výběr osy>[,<typ osy>[,<kanál>]]) Předává aktuálně interpolovanou požadovanou polohu, vztaženou k nulovému bodu soustavy souřadnic stroje MCS.	5-9	MPOS	-
NCF	-	CPL	NCF(<NC-funkce>) Předává syntaxi aktivní NC-funkce uvnitř modální NC-skupiny <NC-funkce>.	5-30	NCF	-
NJUST	-	CPL	Předčasné přepnutí datového výstupu zarov- naného doleva na formátovaný výstup.	5-117	NJUST	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
NOT	-	CPL	NOT <výraz> Negace výrazu BOOL, resp. bitová negace výrazu INTEGER.	2-62	NOT	-
NUL	-	CPL	<proměnná> = NUL Vymazání proměnné.	2-59	NUL	-
OPENR	-	CPL	OPENR(<n>, <název prog.>[, <délka záznamu>]) Otevírá soubor pro následný čtecí přístup.	5-111	OPENR	-
OPENW	-	CPL	OPENW(<n>, <název prog.>[, <délka záznamu>][, <komentář k prog.>][, <délka záznamu>]) Otevírá soubor pro následný zapisovací přístup.	5-111	OPENW	-
OR	-	CPL	<výraz> OR <výraz> Binární operace se dvěma výrazy typu BOOL, resp. INTEGER s funkcí NEBO.	2-62	OR	-
PCS	-	CPL	PCS(<souřadnice>[, <typ výběru>]) Předává poslední naprogramovanou absolutní polohu souřadnice.	5-6	CPOS	-
PDIM	-	CPL	PDIM <název parametru>(<velikost pole>) Musí se používat, když podprogram: <ul style="list-style-type: none"> • má být vyvolán s konstantou String jako předávacím parametrem a • volající program je zvolen bez spojení. 	2-31	PDIM	-
PMT	-	CPL	PMT(<výběr souřadnice>, <kód Pl>[, <řada Pl>][, <tabulka>][, <jednotka>]) Čtecí a zapisový přístup do tabulky umístění XML.	5-26	-	-
PMV	-	CPL	PMV(<výběr souřadnice>[, <řada Pl>]) Udává účinné hodnoty umístění pro souřadnici.	5-25	-	-
PPOS	-	CPL	PPOS(<výběr osy>[, <typ osy>]) Dotaz na skutečnou polohu osy v okamžiku sepnutí měřicího dotyku.	5-14	PPOS	-
PRN#	-	CPL	PRN#(<n>[, <výraz>][, <výraz>][, <výraz>][, ...][:]) Zápisový přístup do souboru.	5-117	PRN#	-
PROBE	-	CPL	PROBE(<výběr osy>[, <typ osy>]) Dotaz na polohu osy v okamžiku sepnutí měřicího dotyku, vzhledem k souřadnicím nulového bodu osy v MCS.	5-13	PROBE	-
REM	-	CPL	REM <text komentáře> Komentování programu.	2-22	REM	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
REPEAT	-	CPL	REPEAT <procedura> UNTIL <podmínka> Smyčková konstrukce s dotazem na podmínku ukončení po prvním zpracování procedury.	2-43	REPEAT	-
REWRITE	-	CPL	REWRITE(<n>) Přepsání existujícího souboru.	5-120	REWRITE	-
ROUND	-	CPL	<Číslo Integer> = ROUND(<výraz Real>) Konverze výrazu REAL zaokrouhlením nahoru, resp. dolů na číslo INTEGER.	2-61	Round	-
SCL	-	CPL	SCL(<SCL-výběr>[,<výběr>[,<typ výběru>]]) Dává pro aktuální kanál poslední naprogramované parametry funkcí PLS a ROT (souřadnice pólu, faktory měřítka a úhly otočení).	5-29	SCL	-
SCS	-	CPL	SCS(<index osy>,<typ ID>,<ID-č.>[,<výsledná proměnná>]) Čtecí přístup k parametrům pohonu SERCOS z aktivní sady parametrů.	5-31	SCS	-
SCSL	-	CPL	SCSL(<index osy>,<typ ID>,<ID-č.>,<název souboru>[,<výsledná proměnná>]) Vytvoření souboru pro seznamy parametrů SERCOS.	5-32	SCSL	-
SD	-	CPL	SD(<skupina>[,<index1>[,<index2>[,<index3>]]]) Čtení aktivních systémových dat NC-řízení.	5-34	SD	-
SDR	-	CPL	SDR(<skupina>[,<index1>[,<index2>]]) Čtení aktivních systémových dat NC-řízení ve formátu REAL.	5-40	SDR	-
SEEK	-	CPL	SEEK(<n>,<k>[,<o>]) Umísuje ukazatel souboru na <k>-tý záznam souboru Random nebo na <k>-tý bajt sekvence souboru.	5-126	SEEK	-
SETERR	-	CPL	Po vyhodnocení ERRNO generovat běhovou chybu.	2-57	-	-
SETWARN	-	CPL	Po vyhodnocení ERRNO generovat výstražné hlášení.	2-58	-	-
SIN	-	CPL	<Funkční hodnota> = SIN(<vstupní hodnota>) Použití funkce sinus na vstupní hodnotu.	2-61	SIN	-
SPOS	-	CPL	SPOS(<výběr osy>) Předává aktuální požadovanou hodnotu fyzické osy.	5-10	SPOS	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
SQRT	-	CPL	<Funkční hodnota> = SQRT(<vstupní hodnota>) Použití funkce druhé odmocniny na <vstupní hodnotu>.	2-61	SQRT	-
STR\$	-	CPL	STR\$([<formátový řetězec>,<hodnota>) Konvertuje číselný výraz <hodnota> na znakový řetězec, který může být přiřazen výhradně znakovému poli. S <formátovým řetězcem> může být znakový řetězec vydán zformátovaný.	5-97	STR\$	-
TAN	-	CPL	<Funkční hodnota> = TAN(<vstupní hodnota>) Použití funkce tangens na <vstupní hodnotu>.	2-61	TAN	-
TCV	-	CPL	TCV(<výběr hodnot>[,<výběr opravy>]) Dává poslední naprogramované hodnoty opravy nástroje.	5-20	TC	-
TIME	-	CPL	<Proměnná String> = TIME Přiřazuje proměnné STRING čas ve formě HH.MM.SS.	5-46	TIME	-
TRIM\$	-	CPL	TRIM\$(<řetězec znaků>) TRIM\$(<řetězec znaků>,"L") TRIM\$(<řetězec znaků>,"R") Dává při přiřazení oblasti znakového pole k proměnné STRING nebo ke znakovému poli znakový řetězec bez mezer vepředu (index "L"), resp. vzadu (index "R"). Bez indexu se vynechávají mezery vepředu i vzadu.	5-100	TRIM\$	-
TRUE	-	CPL	<Proměnná Boolean> = TRUE Pravdivostní hodnota proměnné BOOL	2-52	TRUE	-
VAL	-	CPL	VAL(<výraz String>) Vrací číselnou hodnotu výrazu STRING.	5-99	VAL	-
VERSINF\$	-	CPL	VERSINF\$(<index1>[,<index2>]) Vyvolává administrativní data, např. SW verzi.	5-114		
WAIT	-	CPL	WAIT(bez parametru) Zastavuje zpracování vět, dokud nejsou zpracovány všechny věty naprogramované před příkazem WAIT. WAIT(<čekací doba>) Zastavuje zpracování vět, dokud neuplyne stanovený časový interval. WAIT(BITIF(...)) Zastavuje zpracování vět, dokud na bitovém rozhraní SPS-NC nenastane určitý stav.	2-68	WAIT	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("." znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
WHILE	-	CPL	WHILE <podmínka> DO <procedura> END Smyčková konstrukce s dotazem na podmínku ukončení před prvním proběhnutím smyčky.	2-44	WHILE	-
WPOS	-	CPL	WPOS(<výběr osy>[,<typ výběru>[,<kanál>]]) Předává interpolovanou požadovanou polohu, vzhledem k nulovému bodu obrobku v aktuální WCS.	5-8	WPOS	-
XOR	-	CPL	<výraz> XOR <výraz> Binární operace se dvěma výrazy typu BOOL, resp. INTEGER s funkcí EXKLUZIVNÍ NEBO.	2-62	XOR	-
XTAB	-	CPL	XTAB(<tabulka>,<dílčí struktura>) Čtecí a zápisový přístup do libovolné tabulky XML, která může být definována také uživatelem.	5-28	TDA	-
ZOT	-	CPL	ZOT(<výběr sloupce>,<kód NPV>[[<řada NPV>][,<tabulka>][,<jednotka>]]) Čtecí a zápisový přístup do libovolné tabulky posunutí nulových bodů XML.	5-18	FXC	-
ZOV	-	CPL	ZOV(<výběr osy>[.<řada NPV>]) Udává účinné hodnoty posunutí nulového bodu pro osu / souřadnici stroje.	5-18	FXC	-
D	D	D-korektura	Dx: Volba D-korektury (1-99 nástrojů) D0: Zrušení volby D-korektury.	3-100	G147 - G847,G148	
DiaProg	DIA	Průměr/ poloměr	Programování průměru	4-35	DIA	G16
RadProg	RAD	Průměr/ poloměr	Programování poloměru	4-35	RAD	G15
G16	-	Volba roviny	žádná rovina	3-27	G16	-
G17	-	Volba roviny	Výběr roviny XY	3-28	G17/G20	G17/G20
G17(...), G18(...), G19(...)	-	Volba roviny	Rozšířené přepínání rovin G17/18/19(<osa1>,<osa2>,<osa3>) Osy v závorkách upínají WCS a dostávají význam X, Y a Z. Následně se vybírá naprogramovaná rovina. G17/18/19() Nastavte soustavu souřadnic obrobku opět na standardní hodnoty a potom vyberte naprogramovanou rovinu.	3-30	-	-
G18	-	Volba roviny	Výběr roviny XZ	3-28	G18/G20	G18/G21
G19	-	Volba roviny	Výběr roviny YZ	3-28	G19/G20	G19/G22

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G20	-	Volba roviny	Volný výběr roviny (nezávislý na WCS) Umístění má vliv na WCS, rovina je definována nezávisle na WCS.	3-32	-	-
ED	ED	ED-korektura	ED x: Volba externí opravy nástroje (1..16 nástrojů/břitů) ED0: Zrušení volby externí opravy nástroje	3-101	G145 - G845,G146	
G43	-	Strategie vkládání	Strategie vkládání kruhového oblouku	3-41	G68	G43
G44	-	Strategie vkládání	Strategie vkládání průsečíku	3-41	G69	G44
ChLength	CHL	Fasety/zao- blení	CHL(<délka fasety>) : Zapnout CHL() : Vypnout Vložit přechodové fasety s definovanou délkou.	4-25	G234/G35	-
ChSection	CHS	Fasety/zao- blení	CHS(<úsek fasety>) : Zapnout CHS() : Vypnout Vložit přechodové fasety s definovaným úse- kem.	4-25	G234/G35	-
RoundEps	RNE	Fasety/zao- blení	RNE(<hodnota>): Zapnutí zaoblení RNE() : Vypnutí zaoblení Zaoblení s definovanou odchylkou kontur.	4-114	G34/G35	-
Rounding	RND	Fasety/zao- blení	RND(<hodnota>): Zapnutí zaoblení RND() : Vypnutí zaoblení Zaoblení s definovaným poloměrem zaoblení	4-115	G134/G35	-
FlyMeas	FME	Průběžné měření	FME(MpiAxis< i>) X... Y... Z... Pojezd s průběžným měřením	4-44	G275	-
G61		Přesné zasta- vení	Zapnutí přesného zastavení	3-49	G61/G161	G61
G61(IPS...)	-	Přesné zasta- vení	Zapnutí přesného zastavení s oknem přesného zastavení IPS1, IPS2 nebo IPS3	3-49	G61/G161	G61
G62	-	Přesné zasta- vení	Vypnutí přesného zastavení	3-49	G62/G162	G62
PrecProg	PRP	Přesné pro- gramování	PRP(DIST... EPS...) : Přesné programování zapnout PRP() : Přesné programování vypnout	4-100	G328/G329	-
G45	-	Rychlost frézy	Rychlost bodu záběru frézy	3-43	G64	G99
G46	-	Rychlost frézy	Rychlost středu frézy	3-43	G65	G98
G33	-	Závit	Řezání závitů	3-33	G33	G33
G63	-	Závit	G63(M3/M4, S.../H...) F... Z... ... Řezání vnitřních závitů bez vyrovnávacího sklíčidla	3-51	G32	G63/G64

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
ThreadSet	TST	Závit	Řezání závitů, přídatné funkce	4-146		
G70	-	Inch/metrické	Coulové programování, působí na naprogramované dráhové a délkové rozměry, posuvy a zrychlení.	3-53	G70	G70
G71	-	Inch/metrické	Metrické programování	3-54	G71	G71
G00	G0	Interpolace	Lineární interpolace (rychl oběh)	3-3	G00/G10	G00
G00(..)	G0(..)	Interpolace	Rychloběh s možnostmi Volitelné parametry: NIPS: bez přesného zastavení IPS1: Okno přesného zastavení 1 IPS2: Okno přesného zastavení 2 IPS3: Okno přesného zastavení 3 (pouze brzdění na V=0) POL/POLAR: s programováním polárních souřadnic, např. G0(POL) X50 A45 B10, s A, B = polární úhel 1/2	3-3		
G01	G1	Interpolace	Lineární interpolace (posuv)	3-5	G01/G11	G01
G02	G2	Interpolace	Kruhová interpolace ve směru hodinových ručiček (vč. Helical-N)	3-7	G02/G12	G02
G02(POL)	G2(..)	Interpolace	Kruhová interpolace ve směru hodinových ručiček (vč. Helical-N) s programováním polárních souřadnic	3-11	G02/G12	G02
G03	G3	Interpolace	Kruhová interpolace proti směru hodinových ručiček (vč. Helical-N)	3-7	G03/G13	G03
G03(POL)	G3(..)	Interpolace	Kruhová interpolace proti směru hodinových ručiček (vč. Helical-N) s programováním polárních souřadnic	3-11	G02/G12	G02
G05	G5	Interpolace	Tangenciální vstup kruhu (vč. Helical-N)	3-13	G05	-
G06	G6	Interpolace	Spline křivky, nurb křivky	3-14	G99	-
Collision	CLN	Monitorování kolizí	CLN(1): Zapnout CLN(CollErr.. LA... DLA... DEF) : Zapnout s parametrizací CLN() : Vypnout Monitorování kolizí pro opravu 2D dráhy frézy.	4-27		
KvProg	KVP	Progr. KV	KVP(Y..., Y..., Z..., ...): Zapnutí programování KV KVP() : Vypnutí programování KV	4-60	G14/G15	AXD
G75	-	Měřicí dotyk	Najetí k měřicímu dotyku (přerušeni pohybu)	3-57	G75	-

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G53	-	Posunutí nulového bodu	Vypnutí posunutí všech nulových bodů	3-46	-	
G54.1	G54	Posunutí nulového bodu, řada 1	1. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G55.1	G55	Posunutí nulového bodu, řada 1	2. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G56.1	G56	Posunutí nulového bodu, řada 1	3. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G57.1	G57	Posunutí nulového bodu, řada 1	4. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G58.1	G58	Posunutí nulového bodu, řada 1	5. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G59.1	G59	Posunutí nulového bodu, řada 1	6. Posunutí nulového bodu, řada 1 zapnutí	3-46	G54-G254	
G53.1-G53.5	-	Posunutí nulového bodu, řada 1-5	Vypnutí posunutí nulového bodu, řada 1-5	3-46	G53-G253	
G54.2-G54.5	-	Posunutí nulového bodu, řada 2-5	1. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G54-G254	
G55.2-G55.5	-	Posunutí nulového bodu, řada 2-5	2. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G55-G255	
G56.2-G56.5	-	Posunutí nulového bodu, řada 2-5	3. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G56-G256	
G57.2-G57.5	-	Posunutí nulového bodu, řada 2-5	4. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G57-G257	
G58.2-G58.5	-	Posunutí nulového bodu, řada 2-5	5. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G58-G258	
G59.2-G59.5	-	Posunutí nulového bodu, řada 2-5	6. Posunutí nulového bodu, řada 2-5 zapnutí	3-46	G59-G259	
PoleSet	PLS	Nastavení pólu	PLS(X..., Y..., Z...): Nastavení pólu PLS(): Pól v počátku Pól pro pomůcky zadávání zrcadlového převrácení, otočení.	4-97	G37/G39	

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Strana	Funkce v PNC MTC	
PolarPol	POP	PolarPol	POP(X...,Y...,Z...): Nastavení pólu POP(): Pól v počátku Pól pro programování polárních souřadnic.	4-92	-	
PosMode	PMD	Typ polohování	PMD(A..., B...): Přepnutí typu polohování PMD(): Typ polohování podle parametrů stroje Typ polohování pro nekonečné osy	4-98	G151/G150	G36ff.
ATrans	ATR	Posunutí programových souřadnic	ATR(X..., Y..., Z..., ...): Zapnutí aditivního posunutí programových souřadnic ATR(): Vypnutí aditivního posunutí programových souřadnic	4-148		
Trans	TRS	Posunutí programových souřadnic	TRS(X ..., Y ..., Z ..., ...): Zapnutí posunutí programových souřadnic TRS(): Vypnutí posunutí programových souřadnic	4-148	G60/G67	
G153	-	SE	Umístění: Šikmá rovina, vše vypnout	3-84, 3-84	-	
G152.1	G152	SE řada 1	Umístění: Šikmá rovina, programovatelná, řada 1	3-84	G352	
G154.1	G154	SE řada 1	Umístění: 1. Šikmá rovina, řada 1 zapnout	3-84	G354	
G155.1	G155	SE řada 1	Umístění: 2. Šikmá rovina, řada 1 zapnout	3-84	G355	
G156.1	G156	SE řada 1	Umístění: 3. Šikmá rovina, řada 1 zapnout	3-84	G356	
G157.1	G157	SE řada 1	Umístění: 4. Šikmá rovina, řada 1 zapnout	3-84	G357	
G158.1	G158	SE řada 1	Umístění: 5. Šikmá rovina, řada 1 zapnout	3-84	G358	
G159.1	G159	SE řada 1	Umístění: 6. Šikmá rovina, řada 1 zapnout	3-84	G359	
G153.1-G153.5	-	SE řada 1-5	Umístění: Šikmá rovina, řada 1-5 vypnout	3-84, 3-84	G353	
G152.2-G152.5	-	SE řada 2-5	Umístění: Šikmá rovina, programovatelná, řada 2-5	3-84	G452	
G154.2-G154.5	-	SE řada 2-5	Umístění: 1. Šikmá rovina, řada 2-5 zapnout	3-84	G454/G554	
G155.2-G155.5	-	SE řada 2-5	Umístění: 2. Šikmá rovina, řada 2-5 zapnout	3-84	G455/G555	
G156.2-G156.5	-	SE řada 2-5	Umístění: 3. Šikmá rovina, řada 2-5 zapnout	3-84	G456/G556	
G157.2-G157.5	-	SE řada 2-5	Umístění: 4. Šikmá rovina, řada 2-5 zapnout	3-84	G457/G557	
G158.2-G158.5	-	SE řada 2-5	Umístění: 5. Šikmá rovina, řada 2-5 zapnout	3-84	G458/G558	
G159.2-G159.5	-	SE řada 2-5	Umístění: 5. Šikmá rovina, řada 2-5 zapnout	3-84	G459/G559	
Scale	SCL	Měřítka	SCL(X..., Y..., Z..., ...): nastavení měřítka souřadnic zapnout SCL(): Vypnutí nastavení měřítka souřadnic	4-116	G38/G39	G78/G79

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("-" znamená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
Mirror	MIR	Zrcadlové převrácení	MIR(X1, Y1,Z1,...): Zrcadlové převrácení souřadnic zapnout MIR(): Vypnout zrcadlové převrácení souřadnic	4-71	G38/G39	
G96	-	Prog. vřetena	G96({{<vztažná osa>{,<působíště>}}}) Konstantní řezná rychlost	3-75	G96/G196	G96
G97	-	Prog. vřetena	Přímé programování počtu otáček	3-75	G97	G97
Nibble	NIB	Děrování	NIB(NUM...): Zapnutí obrábění děrováním NIB(): Vypnutí obrábění děrováním	4-74	G662/G660	-
Punch	PUN	Děrování	PUN(NUM...): Zapnutí obrábění děrováním PUN(): Vypnutí obrábění děrováním	4-106	G661/G660	-
PtBlkEnd	PTE	Zdvih děrování	PTE(X..., Y..., Z..., ...) Spuštění zdvihu na konci věty	4-102	G612	-
PtDefault	PTD	Zdvih děrování	Spuštění zdvihu podle parametrizace	4-104	G610	-
PtInpos	PTI	Zdvih děrování	PTI(X..., Y..., Z..., ...) Spuštění zdvihu při Inpos (Pt = punch time)	4-105	G611	-
TangToolOri	TTO	Tangenciální orient. WZ	TTO(SYM..., ANG...): Zapnutí tangenciální orientace nástroje TTO(): Vypnutí tangenciální orientace nástroje	4-140	G630/G631	-
DefTangTrans	DTT	Přechodový úhel	DTT(<přechodový úhel>): Aktivovat DTT(): Deaktivovat Definice tangenciálního přechodu kontury		G228	-
ConstFeed	CFD	V-Profil	Konstantní rychlost	4-66	G310	-
LinDownFeed	LND	V-Profil	Lineární brzdění	4-67	G312	-
LinUpFeed	LNU	V-Profil	Lineární zrychlení	4-66	G311	-
Sin2DownFeed	S2D	V-Profil	Brzdění tvaru Sin2	4-67	G316	-
Sin2UpFeed	S2U	V-Profil	Zrychlení tvaru Sin2	4-66	G315	-
SinDownFeed	SND	V-Profil	Brzdění sinusového tvaru	4-67	G314	-
SinUpFeed	SNU	V-Profil	Zrychlení sinusového tvaru	4-66	G313	-
Rotate	ROT	Otáčení	ROT(<úhel>): Zapnutí otáčení souřadnic ROT(): Vypnutí otáčení souřadnic	4-112	G38/G39	
OvrDis	OVD	Posuv 100%	Vypnutí potenciometru posuvu	4-87	G63	-
OvrEna	OVE	Posuv 100%	Zapnutí potenciometru posuvu	4-87	G66	-
FeedAd	FAD	Přizpůsobení posuvu	FAD(1) : Osu z vytváření posuvu vyjmout FAD(): Zachovat osu ve vytváření posuvu	4-41	G594/G595	
G93	-	Prog. posuvu	Časové programování	3-70	G93	G93

Příloha

Název NC-funkce Úplné označení Zkrácené označení		Skupina ("_" zna- mená: účinné po větách)	Popis / vysvětlení	Stran a	Funkce v PNC MTC	
G94	-	Prog. posuvu	Programování posuvu (za min)	3-71	G94	G94
G94(...)	-	Prog. posuvu	G94({DF <hodnota>},{ DS1 <hodnota>, DS2 <hodnota>, ...}) Inkrementální programování rychlosti s přizpůsobením zrychlení.	3-72	G94	G94
G95	-	Prog. posuvu	Programování posuvu (na ot.)	3-74	G95	G95
FeedForward	FFW	Servo řízení	FFW(X..., Y..., Z..., ...): Zapnout FFW(): Vypnout Aktivovat servo řízení	4-42	G114/G115	G6/G7
BcsCorr	BCR	Oprava polohy obrobku	BCR({{<X _W -offset>},{<Y _W -offset> {,{{<Z _W -offset>},{<úhel1> }{,{{<úhel2> {,{{<úhel3>}}}}}})}): Zapnout BCR(): Vypnout Umístění: Oprava polohy obrobku.	4-23	G138/G139	-
G47	-	Oprava nástroje	Zapnutí opravy délky nástroje	3-44	G146/8	G47
G47(..)	-	Oprava nástroje	G47(<souřadnice L1>,<souřadnice L2>,<souřadnice L3>) G47(ActPlane) Zapnutí opravy délky nástroje s přepnutím přifazení opravy. G47(): Zapnutí opravy délky nástroje s přifazením opravy podle parametrů stroje.	3-44	G146/8	G47
G48	-	Oprava nástroje	Vypnutí opravy délky nástroje	3-44	G145/7ff.	G48/G49
TangTool	TTL	Orient. WZ	TTL(TAX..., SYM..., ANG..., IA..., PLC...): Zapnutí tangenciálního vedení nástroje TTL(): Vypnutí tangenciálního vedení nástroje	4-137	G131/G130	-

Příloha

A.3 Sada znaků ASCII

Dec.	Hex	ASCII	Dec.	Hex	ASCII	Dec.	Hex	ASCII	Dec.	Hex	ASCII
0	00	NUL	32	20	SP	64	40	@	96	60	'
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(72	48	H	104	68	h
9	09	HT	41	29)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	m	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	↑ (^)	126	7E	~
31	1F	US	63	3F	?	95	5F	→(_)	127	7F	DEL

█ : Znaky, které se používají na formátování standardního výstupu. Vynechávají.

A.4 Přídavné kódy kláves

Kód kláves (dec.)	Význam
134	KURZOR NAHOŘE
135	KURZOR DOLE
136	KURZOR VPRAVO
137	KURZOR VLEVO
139	NÁVRAT NA ÚROVEŇ
141	SOFT KLÁVES A 1
142	SOFT KLÁVES A 2
143	SOFT KLÁVES A 3
144	SOFT KLÁVES A 4
145	SOFT KLÁVES A 5
146	SOFT KLÁVES A 6
147	SOFT KLÁVES A 7
148	SOFT KLÁVES A 8

Příloha

A.5 Rejstřík

Znaky

; (Začátek komentáře), 2-21
((Začátek komentáře), 2-21
) (Konec komentáře), 2-21
\$ (označení kanálu), 2-20
žádná rovina, 3-27
Čas spuštění zdvihu
 Koncový bod interpolace, 4-26
 nastavení na standardní hodnotu, 4-28
 Okno Inpos, 4-29
Časové programování, 3-71
Čísla, 5-98
Čísla vět, Instrukce N, 2-20
Řady NPV, 3-46
Řezání vnitřních závitů, 3-51
 Výběr vřetena, 4-66
Řezání vnitřních závitů bez vyrovnávacího sklíčidla,
 3-51
 Výběr vřetena, 4-66
Řezání závitů, 3-33
 Přídavné funkce k, 4-70
Řezná rychlost, konstantní, 3-76
Řízení rychlosti s omezením škrubání, 3-23
 pro polohovací provoz, 3-24
 pro posuvový provoz, 3-24
Řízení výkonu laseru, 4-64
Šikmá rovina, 3-83, 5-26
Šroubovicová interpolace, 3-7
/ (vynechat větu), 2-38
// (Začátek komentáře), 2-22
| (oddělovací znak alternativních parametrů), 2-11,
 3-2, 4-1
\\ (Konec komentáře), 2-22

Čísla

3-D oprava poloměru nástroje, 3-80

A

AAC, 4-17
AAS, 4-17
ABS, 2-60
AC (lokální programování absolutních rozměrů), 3-69
ACOS, 2-61
ACS, 5-10
Adresy os, 2-16
 variabilní, 5-44
Akcelerační interpolátory, 4-66
Aktivace tabulek, 4-33, 4-79
 Tabulky umístění, 4-15

Aktivace tabulek D-korektur, 4-33
Aktivace tabulek posunutí nulových bodů, 4-79
ALN, 4-10
AND, 2-62
APOS, 5-12
ARA, 4-2
Area, 4-2
ARRAY, 2-53
ASC, 5-98
ASIN, 2-61
AssLogName, 4-10
ASTOPA / ASTOPO, Synchronizace kanálů zasta-
 vením pohybu, 2-78
Asynchronní osy, Rychlost, 3-98
ATAN, 2-61
ATCAL, 4-11
ATFWD, 4-13
ATGET, 4-14
ATPUT, 4-15
ATrans, 4-73
Atributy, 2-14
ATS, 4-22, 4-73
AxAcc, 4-17
AxAccSave, 4-17
AXC, 4-20
AxCouple, 4-20
AxisToSpindle, 4-22
AXO, 5-16
AXP, 5-44

B

BCD, 2-62
BCR, 4-24
BcsCorr, 4-24
Bezpečnostní pokyny, 1-5
BIN, 2-62
BITIF, 5-45
BikNmb, 2-71
BNB, 2-71
Bod záběru frézy (oprava posuvu), 3-43
BOOLEAN, 2-48, 2-52
Brzdné interpolátory, 4-67
BSTOPA / BSTOPO, Synchronizace kanálů zasta-
 vením pohybu, 2-80

C

CALL, 2-29
CASE-LABEL...LABEL-OTHERWISE-ENDCASE,
 2-40

Příloha

CFD, 4-67
 CHARACTER, 2-48, 2-54
 CHL, 4-26
 ChLength, 4-26
 CHR\$, 5-98
 CHS, 4-26
 ChSection, 4-26
 Chyba kontury, 4-24
 Chyby a jejich kategorie, 5-48
 CLN, 4-28
 CLOCK, 5-47
 CLOSE, 5-125
 CLRWARN, 2-58
 COF, 5-17
 COFFS, 3-81
 Collision, 4-27
 ConstFeed, 4-67
 Coord, 4-31
 COS, 2-61
 Coulové programování, 3-53
 CoupleSplineTab, 4-32
 CRD, 4-31, 4-32

D

Děrování, 4-30
 D-korektura, 3-101
 Databáze nástrojů, 5-23
 DATE, 5-47
 DAX, 4-34
 DBSEA, 5-24
 DBTAB, 5-23
 DCA, 4-37
 DcAxis, 4-37
 DCB, 4-38
 DcBreak, 4-38
 DCC, 4-38
 DcCont, 4-38
 DCF, 4-37
 DcFilter, 4-37
 DCL, 4-38
 DcLimit, 4-38
 DCM, 4-38
 DcMon, 4-38
 DCR, 4-37
 DCS, 4-33
 DCT, 5-22
 DcTSel, 4-33
 Deaktivovat rovinu, 3-27
 DefAxis, 4-34
 Definice bodu otáčení, 4-21
 Definice bodu zrcadlového převrácení, 4-21
 Definice TCS v programových souřadnicích, 4-67
 Definice, aktivace, rozpojení a deaktivace svazku spojení, 4-47

Definovat typ spline křivky, 4-56
 Délka fasety, 4-25
 DIA, 4-35
 Digitalizace, Regulace výšky, 4-37
 Dílčí program, 2-1
 DIM, 5-94
 DIRCR, 5-117
 DIRDEL, 5-118
 DIRINF, 5-116
 DistCtrl, 4-37
 DN, 3-15
 Doběh dráhy, 4-24
 Doba prodlevy, 3-12
 Doba zpracování, Časové programování, 3-71
 Doplňující podmínky, 2-8
 DOUBLE, 2-48, 2-52
 DPC, 5-18
 Dráhové funkce, 2-8

E

EBC, 3-17
 ED-korektura, 3-102
 EGB, 1-7
 Elektrostatický výboj
 ochrana, 1-7
 Pracoviště, 1-7
 Elektrostaticky ohrožený modul, 1-7
 EndPosCouple, 4-39
 EOF, 5-125
 ERASE, 5-132
 ERRNO, 2-56

F

F-adresa, 3-97
 F-slovo, 3-97
 F-slovo (čas), 3-71
 F-slovo (posuv)
 na ot., 3-75
 za min., 3-72
 FA-adresa, 3-98
 FA-slovo, 3-98
 FAD, 4-41
 FeedAd, 4-41
 FeedForward, 4-42
 FFW, 4-42
 FILEACCESS, 5-133
 FILECOPY, 5-135
 FILEDATE, 5-134
 FILENO, 5-115
 FILEPOS, 5-126
 FILESIZE, 5-130
 FlyMeas, 4-44
 FME, 4-44
 FOR - STEP - TO - NEXT, 2-42

Příloha

FSM, 4-49
FsMove, 4-49
FSP, 4-46
FsProbe, 4-46
FSR, 4-49
FsReset, 4-49
FST, 4-49
FsTorque, 4-49
Funkce, pro připojení NCS, 5-53

G

G0, 3-3
G1, 3-5
G2, 3-9, 3-10
G3, 3-9, 3-10
G4, 3-12
G140, 3-81
G141, 3-81
G142, 3-81
G152, 3-85
G153, 3-85, 3-86
G154, 3-85
G155, 3-85
G156, 3-86
G157, 3-86
G158, 3-86
G159, 3-86
G16, 3-27
G17, 3-28
G17(...), 3-30
G18, 3-28
G18(...), 3-30
G184, 3-68
G19, 3-28
G19(...), 3-30
G20, 3-32
G33, 3-34
G40, 3-39
G41, 3-39
G42, 3-39
G43, 3-42
G44, 3-42
G45, 3-43
G46, 3-43
G47, 3-44
G48, 3-44
G5, 3-13
G53 až G59.5, 3-47
G53-G59, 3-46
G6, 3-14
G61, 3-49
G62, 3-49
G63, 3-51
G70, 3-53

G71, 3-54
G74, 3-55
G74(HOME), 3-56
G75, 3-57
G76, 3-58
G8, 3-22
G80, 3-62
G81, 3-62
G82, 3-63
G83, 3-64
G84, 3-65
G85, 3-66
G86, 3-67
G9, 3-22
G90, 3-69
G91, 3-69
G93, 3-71
G94, 3-72
G94(...), 3-73
G95, 3-75
G96, 3-78
G97, 3-78
GAX, 4-51
GetAxis, 4-51
GETERR, 5-48
GOA, 2-33
GoAhead, 2-33
GOB, 2-34
GoBack, 2-34
GOC, 2-35
GoCond, 2-35
GOTO, 2-37
GoTo, 2-36

H

HSB, 4-52, 4-55
HsBlkSwitch, 4-52
HWOC, 4-57
HWOCDEL, 4-57
HWOCOFF, 4-57

I

IC (lokální programování relativních rozměrů), 3-69
Identifikátory os, 2-16
Identifikátory souřadnic, 2-16
IF - THEN - ELSE - ENDIF, 2-39
IME, 4-58
InitMeas, 4-58
Inkrementální programování, 3-69
Inkrementální programování rychlosti, 3-73
INP#, 5-123
INSDEP, 3-81
INSTR, 5-97
Instrukce, 2-8

Příloha

Instrukce (CPL), 2-59
Instrukce CPL, 2-3, 2-59
Instrukce skoku, 2-24, 2-32
INT, 2-60
INTEGER, 2-52, 2-64
Interpolátor
 Brzdění, 4-67
 Konstantní pojezd, 4-66
 Zrychlení, 4-66
Interpolátor konstantního pojezdu, 4-66
IPS, 3-4, 3-5
IPS1, 3-4, 3-5, 3-50
IPS2, 3-4, 3-5, 3-50
IPS3, 3-4, 3-6, 3-50

J
JogWCSSelect, 4-59

K
Kalibrace osových kinematik
 načtení parametrů z NC, 4-14
 optimalizace parametrů, 4-11
 přepočítání parametrů, 4-13
 zápis parametrů do NC, 4-15
Klíčové pojmy, 2-66
Klíčové znaky, 2-65
Komentáře, 2-21
 vnořené, 2-21
Komentování programu, 2-21
Komunikace, 5-136
Konec hlavního programu, 3-88
Konec podprogramu, 2-17
Konec programu, 2-17, 3-88
Konstantní řezná rychlost, 3-76
Konstanty, 2-64
 s dvojnásobnou přesností, 2-64
Kontrolní práce, 1-6
Konturové přechody pro opravu dráhy frézy, 3-41
Konturové posunutí, 5-17
Konverze, Číselné systémy, 2-62
Krokový provoz v souřadnicích obrobku, 4-59
Kruhový oblouk (konturový přechod), 3-41
Kruhová/spirálová interpolace, 3-7
Kvalifikovaný personál, 1-3
KVP, 4-60
KvProg, 4-60

L
LEN, 5-97
LEN (pro děrování/vysekávání), 4-62
LFC, 4-64
LFCnf, 4-64
LFP, 4-64
LinDownFeed, 4-67

Lineární interpolace
 Posuv (G01), 3-5
 Rychlý chod (G00), 3-3
Lineární změnové funkce, 4-66
LinModZp, 4-69
LinUpFeed, 4-67
LJUST, 5-119
LMZ, 4-69
LND, 4-67
LNU, 4-67
Logické operace, 2-61
 Logické, 2-61
lokální, 2-19
lomené závorky, 2-3

M
Měření na pevném dorazu, 4-46
Měření, průběžné, 4-44
 Inicializace, 4-58
Měřicí dotyk, 3-57
 číst, 5-13
Měřítka, 5-30
Měrná jednotka
 Coul, 3-53
 metrická, 3-54
Měrné jednotky , dodávaných poloh os a souřadnic,
 5-2
M00, 3-87
M01, 3-87
M02, 3-88
M103, 3-89
M104, 3-90
M105, 3-91
M113, 3-89
M114, 3-90
M13, 3-89
M14, 3-90
M2, 3-88
M203, 3-89
M204, 3-90
M205, 3-91
M213, 3-89
M214, 3-90
M3, 3-89
M30, 3-88
M4, 3-90
M5, 3-91
MCODS, Motion Control datové služby, 5-53
MCOPS, 5-83
MCS, 5-9
Metrické programování, 3-54
Mezery, Odstraňování, 5-101
MID\$, 5-95, 5-96
MIR, 4-71, 4-72

Příloha

Mirror, 4-71
MMC, 5-136
modální, 2-18
Modulo osa, vynulování, 4-69
Monitorování kolizí, 4-27
Monitorování oblastí, 4-2
Monitorování určité oblasti, 4-2
Motion Control datové služby, MCODES, 5-53
Motion Control procesové služby, 5-83

N

N (čísla vět), 2-20
Náhradní díly, 1-7
Najetí do osové polohy, pevné stroje, 3-58
Najetí do pevné osové polohy stroje, 3-58
Najetí do referenčního bodu, 3-56
Najetí na pevný doraz, 4-48
Najetí na souřadnice referenčního bodu, 3-55
Naprogramované posunutí kontur, 4-44
Následné vřeten
 Vzdálenost spojení, 4-48
 Změna úhlového posunutí při aktivním spojení,
 4-50
Nastavení měřítka, 4-40
Nastavení programové polohy (G92), 4-43
Nastavení skutečné hodnoty, 4-43
Názvy os, 2-16
Názvy souřadnic, 2-16
NC programování, 2-2
NC-program, 2-1
NCF, 5-31
Nekonečné osy, Přepnutí typu polohování, 4-22
nemodální, 2-19
Nepodmíněný skok, 2-36
Nepodmíněný skok s CPL, 2-37
nesamodržné, 2-19
NIB, 4-74
Nibble, 4-74
NIPS, 3-4
Nízkonapěťová směrnice, 1-1
NJUST, 5-119
NOT, 2-62
NUL, 2-59
NUM (pro děrování/vysekávání), 4-76
NURBS, 3-19

O

O(), 4-1
Ochranné známky, 1-8
Oddělovací znaky, mezi dílčími slovy, 2-15
Odkazové programování, 2-23
Odstranění osy ze skupiny os, 4-33
OFFSTOPA / OFFSTOPO, 2-83
Omega, 3-98

Omega-adresa, 3-98
Omega-slovo, 3-98
Omega-slovo (posuv), za min., 3-72
Omezení maximálního momentu, 4-32
Omezení momentu, 4-32
Omezení přípravy vět: Počet vět, 2-71
Omezení počtu otáček, 4-45
Online oprava v souřadnicích obrobku, 4-57
Opakovací instrukce, 2-24, 2-42
OPENR, 5-113
OPENW, 5-113
Operace
 matematické, 2-60
 s dvojnásobnou přesností, 2-64
Oprava délky nástroje, 3-44
Oprava dráhy, 3-39
Oprava dráhy frézy, 3-39
 Konturové přechody, 3-41
 Oprava posuvu, 3-43
Oprava polohy obrobku, 4-23, 5-18
Oprava poloměru nástroje, 3D, 3-80
Oprava posuvu pro opravu dráhy frézy, 3-43
Oprava upnutí, 4-23
Opravy nástroje, 5-21
OR, 2-62
Orientační programování, 4-1
Orientace nástroje, statická, 4-60
Otáčení kontury, 4-36
OVD, 4-11
OVE, 4-11
OvrDis, 4-11
OvrEna, 4-11
Označení kanálů, 2-20

P

Předávání os
 Monitorování oblastí, 4-3
 Odstranění osy ze skupiny os, 4-33
 Převzetí osy, 4-51
 Převzetí osy, příp. čekání, 4-75
 Převzetí standardního nastavení os z MP, 4-34
 Vymazání logického názvu osy, 4-34
 vypnutí provozu osy C, 4-22
 Zadání logického názvu osy, 4-10
 Zapnutí vřeten v provozu osy C, 4-55
Přehled funkce, MCODES, 5-54
Přejmenování osy (log. název osy), 4-10
Přepínání oprav G78, G79, 3-60
Přepnutí hlavního vřeten, 4-70
Přepnutí roviny, 3-28, 3-30
Přesné programování, 4-24
Přesné zastavení
 Rychlý chod (G00), 3-3
 ZAP/VYP (G61,G62), 3-49

Příloha

- Převzetí nastavení os z MP, 4-34
- Převzetí osy, 4-51
- Převzetí osy, příp. čekání, 4-75
- Převzetí standardního nastavení os z MP, 4-34
- Příkazová slova, rezervovaná, 2-66
- Příklady programování
 - Funkce NCS, 5-91
 - Znakové řetězce, 5-102
- Přímé programování počtu otáček, 3-76
- Připojení NCS, 5-51
- Připojení NCS pomocí MCODES, 5-53
- Přírůstek zrychlení, 3-26
- Přístupová oprávnění pro soubory, 2-1
- P-adresa, 2-26
- PAC, 4-12
- Parametrizace statické orientace nástroje, 4-60
- Parametry interpolace, 3-29
- PathAcc, 4-12
- PCS, 5-7
- PCSPROBE, 5-13
- PDIM, 2-31
- Pevný doraz
 - měření na, 4-46
 - najetí na, 4-48
- phi, 4-1
- PHS, 4-13
- PL, 3-16, 3-18, 3-19
- PLC, 5-46
- PLS, 4-21
- PMD, 4-22
- PMS, 4-15
- PMT, 5-27
- PmTSel, 4-15
- PMV, 5-26
- Počet otáček vřetena, 3-99
- Počet vět, Omezení přípravy vět, 2-71
- Podmíněný skok, 2-35
- Podprogramy, 2-25, 2-67
 - lokální, 2-25
 - modální, 2-25
 - Vyvolání bez P-adresy, 2-27
 - Vyvolání pomocí příkazu CALL (CPL), 2-29
 - Vyvolání s předáním parametrů, 2-30
 - Vyvolání s P-adresou, 2-26
 - Vyvolání uživatelsky definované modální, 2-28
 - Vyvolání uživatelsky definované s G/M-kódy, 2-28
- Pohony, virtuální, 4-74
- POL
 - při G0, 3-4
 - při G2, G3, 3-11
 - Příklad programování, 4-19
 - pro G1, 3-6
- PolarPol, 4-16
- Pole proměnných, 2-53
- PoleSet, 4-21
- Polohy os, číst, 5-5
 - Přehled, 5-2
- Polohy souřadnic, číst, 5-5
 - Přehled, 5-2
- POP, 4-16
- PosDepHSOut, 4-13
- PosMode, 4-22
- Posunutí a umístění, číst, 5-16
- Posunutí kontur, naprogramované, 4-44
- Posunutí nulových bodů, 3-46, 5-19
- Posunutí nulového bodu, 3-46
- Posuv, 3-5, 3-98
 - F-adresa, 3-97
 - synchronní osy, 3-97
- Posuv 100%, 4-11
- Použití k určenému účelu, 1-1
- PPOS, 5-15
- Průběžné měření, 4-44
 - Inicializace, 4-58
- Průběh programu, 2-24
- Průsečík (konturový přechod), 3-41
- Prázdné řádky, 2-7, 2-21
- PrecProg, 4-25
- PRN#, 5-119
- PROBE, 5-14
- Program
 - přerušit, 3-87
 - podmíněně přerušit, 3-87
 - ukončit, 3-88
- Programová slova, 2-10
 - jako parametry, 2-13
 - z NC-funkcí, 2-11
- Programovací atributy, 2-14
- Programování absolutních rozměrů, 3-69
 - lokální, 3-69
- Programování CPL, 2-2
- Programování faset, 4-25
- Programování koeficientů, Spline, 3-15
- Programování KV, 4-60
- Programování návěští, 2-32
- Programování přírůstků souřadnic, 3-69
- Programování počtu otáček (vřeteno), 3-99
- Programování počtu otáček, přímé, 3-76
- Programování polárních souřadnic, 4-17
 - Definice pólu, 4-16
- Programování poloměru, 3-9, 4-35
- Programování posuvu
 - inkrementální, 3-73
 - na ot., 3-75
 - za min, 3-72
- Programování průměru, 4-35
- Programování proměnných, 2-45

Příloha

Programování relativních rozměrů, 3-69
 lokální, 3-69
Programování středu, 3-10
Programovatelné rozdělení dráhy, 4-58
Programové věty, 2-7
Proměnné
 CHARACTER, 2-54
 Definovatelné permanentní, 2-47
 Globální, 2-46
 Lokální, 2-46
 Přehled, 2-54
 Permanentní, 2-47
 STRING, 2-54
 Strukturované, 2-51
Provedení vět, Rozdíly DIN - CPL, 2-4
PRP, 4-25
psi, 4-1
PtBikEnd, 4-26
PTD, 4-28
PtDefault, 4-28
PTE, 4-26
PTI, 4-29
PtInpos, 4-29
PUN, 4-30
Punch, 4-30
PW, 3-19

R

RAD, 4-35
RAX, 4-33
RDT, 4-32
REAL, 2-52, 2-64
RedTorque, 4-32
Regulace výšky pro digitalizaci, 4-37
Relační operace, 2-63
REM, 2-22
RemAxis, 4-33
RemLogName, 4-34
REPEAT - UNTIL, 2-43
REPOSTP, 4-35
REWRITE, 5-122
RLN, 4-34
RND, 4-39
RNE, 4-38
ROT, 4-36
Rotate, 4-36
ROTAX(), 4-1
ROUND, 2-61
RoundEps, 4-38
Rounding, 4-39
Rozdělení dráhy, programovatelné, 4-58

Rozdělení pojezdové věty
 Délka dílčí dráhy, 4-62
 Počet dílčích drah, 4-76
 všeobecně, 4-58
Rozhodovací instrukce, 2-24, 2-38
Rozhraní SPS, 5-45
Rozvětovací instrukce, 2-24, 2-38
Rychlý chod, 3-3
Rychlost asynchronních os, FA-adresa, 3-98
Rychlostní profily, Definice vlastních, 4-66

S

S-slovo, 3-99
S2D, 4-67
S2U, 4-67
samodržné, 2-18
SBC, 3-17
Scale, 4-40
SCC, 4-42
SCL, 4-40, 5-30
SCS, 5-32
SCSL, 5-33
SD, 5-34
SDF, 4-56
SDR, 5-40
SEEK, 5-128
SelCrdCouple, 4-42
Selektivní aditivní spojení souřadnic, 4-42
Servo řízení, 4-42
SETERR, 2-57
SetPos, 4-43
SETWARN, 2-58
Shape. *Siehe* Řízení rychlosti s omezením šhubání
Shift, 4-44
SHT, 4-44
Signál Highspeed, průběžná změna věty, 4-52
Signál Highspeed, změna věty s přerušením, 4-55
SIN, 2-61
Sin2DownFeed, 4-67
Sin2UpFeed, 4-67
SinDownFeed, 4-67
SinUpFeed, 4-67
Skočit dopředu, 2-33
Skočit dozadu, 2-34
Skok
 dopředu, 2-33
 dozadu, 2-34
 nepodmíněný, 2-36
 nepodmíněný s CPL, 2-37
 podmíněný, 2-35
Skok CPL, 2-37
Skupiny NPV, 3-46
Skupiny proměnných, 2-46
složené závorky, 2-11

Příloha

- Směrnice EMV, 1-1
- SMax, 4-45
- SMin, 4-45
- SMN, 4-45
- SMX, 4-45
- SND, 4-67
- SNU, 4-67
- Souřadnice obrobku
 - krokování, 4-59
 - Online oprava, 4-57
- Soubor
 - číst, 5-123
 - otevřít, 5-112
 - identifikovat konec, 5-125
 - kopírovat, 5-135
 - Nastavit ukazatel, 5-128
 - Názvy, 5-111
 - Pozice ukazatele, 5-126
 - Určení data, 5-134
 - Určení přístupových oprávnění, 5-133
 - Určení velikosti, 5-130
 - vymazat, 5-132
 - zapisovat, 5-119
 - zavřít, 5-125
- SPC_WAIT, 4-53
- SPCC, 4-47
- SPCD, 4-48
- SPCE, 4-49
- SpCouple_Wait, 4-53
- SpCoupleConfig, 4-47
- SpCoupleDist, 4-48
- SpCoupleErrWin, 4-49
- SpCouplePosOffs, 4-50
- SpCouplePosOffs_Wait, 4-51
- SpCoupleSyncWin, 4-52
- SPCP, 4-50
- SPCP_WAIT, 4-51
- SPCS, 4-52
- SPG, 4-54
- SPGALL, 4-54
- SpindleToAxis, 4-55
- Spirálová interpolace, 3-7
- Spline
 - C1-spojité kubická, 3-17
 - Programování koeficientů, 3-15
- Spline funkce, B-spline křivky (NURBS), 3-19
- spline funkce, C2-spojité kubická, 3-17
- Spline programování, 3-14
 - Definice typu spline křivky, 4-56
- SplineDef, 4-56
- SPLIT, 4-58
- Spojení koncových poloh, 4-39
- Spojení os, 4-19
- Spojení souřadnic, ve stejném kanálu, 4-39
- Spojení vřeten
 - Čekání na synchronní provoz, 4-53
 - Čekání na úhlové posunutí, 4-51
 - Okno synchronního chodu, 4-52
- Spojení vřetena
 - Definice, aktivace, rozpojení a deaktivace svazku spojení, 4-47
 - Okno chyby synchronního chodu, 4-49
 - Vzdálenost spojení následného vřetena, 4-48
 - Změna úhlového posunutí při aktivním spojení, 4-50
- Spojování, 2-5
- SPOS, 5-11
- SPS, 4-43
- SPV, SPVE, Zápis permanentní proměnné CPL, 2-77
- SQRT, 2-61
- SSPG, 3-99
- Střed frézy (oprava posuvu), 3-43
- STA, 4-55
- Standardní programování, 2-2
- StatToolOri, 4-60
- STO, 4-60
- Stoupání závitu, 3-51
- STR\$, 5-98
- STRING, 2-54
- Strmost dráhy, 3-21
- Struktura souborů
 - Random, 5-112
 - Sekvenční, 5-112
- Synchronizační funkce
 - přípravy vět, 2-68
 - v okamžiku provádění vět, 2-72
- Synchronní provoz, čekání na (spojení vřeten), 4-53
- Systém souborů a jejich ochrana, 2-1
- Systémová data
 - jednoduché typy, 5-31
 - strukturované typy, 5-42
- T**
- Tabulka spojení, 2-5
- Aktivace tabulek umístění, 4-15
- Tabulky XML, všeobecný přístup, 5-29
- TAN, 2-61
- Tangenciální orientace nástroje, 4-64
- Tangenciální vedení nástroje, 4-61
- Tangenciální vstup kruhu, 3-13
- TangTool, 4-62
- TangToolOri, 4-64
- TappSp, 4-66
- TCS, 4-67
- TcsDef, 4-67
- TCV, 5-21
- theta, 4-1
- ThreadSet, 4-70

Příloha

TIME, 5-47
Trans, 4-73
Transformace dopředu, 4-13
TRIM\$, 5-101
TRS, 4-73
Trvání věty, 3-71
TSP, 4-66
TST, 4-70
TTL, 4-62
TTO, 4-64
Typ polohování pro nekonečné osy, 4-22
Typy proměnných, 2-52

U

Úhlové posunutí při aktivním spojení, 4-50
Úhlové posunutí, čekání na, 4-51
úhlové závorky, 2-11
ukončení programu, 3-88
Umístění, 5-26
 Šikmá rovina, 3-83
 Oprava polohy obrobku, 4-23
Úsek fasety, 4-25
Uzemňovací pásek, 1-7

V

Vřeteno
 Levotočivý chod, 3-90
 přepnout mezi polohovým a otáčkovým provozem,
 4-59
 Přepnutí hlavního vřetena, 4-70
 Pravotočivý chod, 3-89
 seřadit/polohovat, 3-92
 Zastavení, 3-91
 Zrušení rezervace, 3-91, 4-46
Výběr roviny, volný, 3-32
Výrazy STRING
 přiřadit, 5-105
 porovnávat, 5-108
 zřetěžit, 5-108
Výstup Highspeed, závislý na poloze, 4-13
Výstup Highspeed závislý na poloze, 4-13
VAL, 5-100
Vedení nástroje, tangenciální, 4-61
VERSINF\$, 5-116
VirtAxisPos, 4-74
Virtuální pohony, 4-74
Vlečná chyba, 3-49
Vlečná vzdálenost, 3-3, 3-49
Volba transformace os, 4-30
Volný výběr roviny, 3-32
Vracené chybové hodnoty, 5-51
Vstup kruhu, tangenciální, 3-13

Vydání pokynů pro práci, 2-23
Vymazání názvu osy (log.), 4-34
Vynulování modulu osy, 4-69
Vypnutí provozu osy C pro vřetena, 4-22
Vysekávání, 4-74
Vytváření posuvu, skrytí os, 4-41
Vytváření programů, 2-20
vyvolání podprogramů, s M-adresou, 2-28
Vyvolání podprogramu, 2-24
Vzdálenost spojení následného vřetena, 4-48

W

WAIT, 2-68
WAITA, WAITO, Stav na digitálním rozhraní, 2-73
WaitAxis, 4-75
WAX, 4-75
WCS, 5-8
WHILE - DO - END, 2-44
WID, 4-76, 4-77
WPV, WPVE, Hodnota permanentní proměnné CPL,
2-75
Writeld, 4-76, 4-77
WSTOPA / WSTOPO, Synchronizace kanálů zasta-
vením pohybu, 2-81

X

XOR, 2-62
XTAB, 5-29

Z

Zařízení nouzového vypnutí, 1-6
Zaoblení rohů
 se zadáním odchylky, 4-38
 se zadáním poloměru, 4-39
Zápis parametrů SERCOS, 4-76
Zapisovat parametry SERCOS, 4-76
Zapnutí chladicí kapaliny, 3-89, 3-90
Zastavení obrábění, 3-87, 3-88
Zastavení programu, 3-87
 podmíněné, 3-87
Závit
 vícechodý, 3-36
 zřetěžený, 3-36
Závorky
 lomené, 2-3
 složené, 2-11, 3-2, 4-1
 úhlové, 2-11, 3-2, 4-1
Zaznamenávání času, 5-47
Změna věty pomocí signálu Highspeed
 Průběžná změna věty, 4-52
 Změna věty s přerušením, 4-55
Změna zrychlení posuvu, 4-12

Příloha

Znakový řetězec, 5-94

 Délka, 5-97

 hledání, 5-97

 načtení, 5-95

 Změna, 5-96

Znaková řetězcová konstanta, 2-65

Znaková pole, dimenzování, 5-94

Znakové řetězce a čísla, 5-98

ZOS, 4-79

ZOT, 5-19

ZoTSEL, 4-79

ZOV, 5-19

Způsoby zápisu, 3-2, 4-1

Zpracování souborů, 5-111

Zrcadlové převrácení, 4-71

Zrychlení

 os, 4-17

 posuvu, 4-12

Zrychlení os

 dočasné uložení, 4-17

 změna, 4-17

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 (0)93 52-40-50 60
Fax +49 (0)93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

