SIEMENS

SINUMERIK 840D / 840Di / 810D

Příručka programování

Pro pokročilé

SIEMENS

SINUMERIK 840D/840Di/810D

Příručka programování

Pro pokročilé

Platí pro:

Řídící systém	*Stav software*
SINUMERIK 840D	7
SINUMERIK 840DE (verze pro export)	7
SINUMERIK 840D powerline	7
SINUMERIK 840DE powerline	7
SINUMERIK 840Di	2
SINUMERIK 840DiE (verze pro export)	2
SINUMERIK 810D	3
SINUMERIK 810DE (verze pro export)	3
SINUMERIK 810D powerline	7
SINUMERIK 810D powerline	7

Vydání 03.04

1

Technika podprogramů, technika maker 2

Správa souborů a programů 3

Oblasti ochrany 4

Speciální příkazy dráhy 5

Framy 6

Transformace 7

Chování při pohybu po dráze 9

Synchronní akce pohybu 10

Oscilační pohyb 11

Lisování a prostřihování 12

Další funkce 13

Vlastní programy pro oddělování třísky 14

Tabulky 15

Přílohy A

Dokumentace systému SINUMERIK

Klíč k vydáním
V následujících odstavcích naleznete stručné podrobnosti týkající se tohoto a předcházejících vydání.

Ve sloupci „Poznámka“ je písmenem kódem uvedeno, v jakém stavu se jednotlivá vydání nacházejí.

Označení stavu ve sloupci „Poznámka“:

A ... Nová dokumentace
B ... Nezměněný dotisk s novým objednacím číslem
C ... Přepracovaná verze s novým číslem verze.

Vydání	**Objednací číslo**	**Poznámka**
02.95	6FC5298-2AB00-0AP0	A
04.95	6FC5298-4AB00-0AP1	C
12.95	6FC5298-3AB00-0AP0	C
03.96	6FC5298-3AB00-0AP1	C
08.97	6FC5298-4AB00-0AP0	C
12.97	6FC5298-4AB00-0AP1	C
12.98	6FC5298-5AB00-0AP0	C
08.99	6FC5298-5AB00-0AP1	C
04.00	6FC5298-5AB00-0AP2	C
10.00	6FC5298-6AB00-0AP0	C
09.01	6FC5298-6AB00-0AP1	C
11.02	6FC5298-6AB00-0AP2	C
03.04	6FC5298-7AB00-0AP1	C

Tato příručka je také součástí dokumentace na CD-ROM (**DOCONCD**)

Vydání	Objednací číslo	Poznámka
03.04	6FC5298-7CA00-0AG0	C

Značky
SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® a SIMODRIVE® jsou chráněnými značkami firmy Siemens. I zbývající označení v této příručce mohou být ochrannými značkami, jejichž použití třetími stranami pro vlastní účely může poškodit práva jejich majitele.

Další informace naleznete na internetu na adrese:
http://www.ad.siemens.de/sinumerik

V rámci řídícího systému se mohou vyskytovat i další funkce nepopsané v rámci této dokumentace, které lze spustit. S ohledem na tyto funkce však není možné vznést žádný nárok pro případ nové dodávky nebo servisního případu..

Tato dokumentace byla vytvořena v prostředí WinWord V 8.0 a Designer V 6.0.
Další šíření, jakož i rozmnožování této dokumentace, neoprávněné využívání a rozšiřování jejího obsahu je nepřípustné, pokud nebylo výslovně povoleno. Jednání v rozporu s těmito pokyny zavazuje k náhradě škody. Všechna práva vyhrazena, zejména pro případ udělování patentů nebo zápis GM.

Obsah této dokumentace byl kontrolován, zda je v souladu s popisovaným hardware a software. Přesto však není možné vyloučit odchylky, takže nedáváme žádnou záruku, že se dokumentace bude dokonale shodovat. Údaje v této příručce jsou však pravidelně kontrolovány a potřebné korekce jsou obsaženy v následujících vydáních. Budeme Vám velmi vděční za jakékoli návrhy na zlepšení.

© Siemens AG 1997 – 2001. Všechna práva vyhrazena.

Technické změny vyhrazeny.

Předmluva

Rozčlenění dokumentace

Dokumentace systému SINUMERIK je rozčleněna do tří úrovní:

- Všeobecná dokumentace
- Uživatelská dokumentace
- Servisní / výrobní dokumentace

Komu je dokumentace určena

Tato dokumentace je určena pro pokročilé uživatele obráběcích strojů. Poskytuje podrobné informace, které uživatel potřebuje, aby mohl programovat řídící systémy SINUMERIK 840D/840Di/810D.

Standardní rozsah

Tato Příručka pro programování popisuje možnosti poskytované standardními funkcemi. Doplňky nebo změny implementované výrobcem stroje naleznete v dokumentaci dodávané výrobcem stroje.

Podrobnější informace o dalších publikacích vztahujících k systémům SINUMERIK 840D/840Di/810D a publikace, které se vztahují na všechny řídící systémy SINUMERIK (např. popis univerzálního rozhraní, měřicí cykly atd.), si můžete vyžádat od svého oblastního zastoupení firmy Siemens.

V rámci řídícího systému se mohou vyskytovat další v této dokumentaci nepopisované funkce. V souvislosti s mini však není možné vznášet žádné nároky na jejich dodání v rámci nového systému nebo při servisním zásahu.

© Siemens AG 2004, Všechna práva vyhrazena

Platnost

Tato Příručka pro programování platí pro následující řídící systémy:

SINUMERIK 840D	SW7
SINUMERIK 840DE (varianta pro export)	SW7
SINUMERIK 840D powerline	SW7
SINUMERIK 840DE powerline	SW7
SINUMERIK 840Di	SW2
SINUMERIK 840 DiE (varianta pro export)	SW2
SINUMERIK 810D	SW3
SINUMERIK 810DE (varianta pro export)	SW3
SINUMERIK 810D powerline	SW7
SINUMERIK 810D powerline	SW7

s čelními ovládacími panely OP 010, OP 010C, OP 010S, OP 12 nebo OP 15 (PCU 20 nebo PCU 50)

SINUMERIK 840D powerline

Od 09.2001 jsou k dispozici následující systémy se zvýšeným výkonem:
SINUMERIK 840D powerline a
SINUMERIK 840DE powerline
Pokud budete potřebovat seznam modulů, které jsou k dispozici v systémech **powerline**, nahlédněte do popisu hardware /PHD/ v kapitole 1.1.

SINUMERIK 810D powerline

Od 12.2001 jsou k dispozici následující systémy se zvýšeným výkonem:
SINUMERIK 810D powerline a
SINUMERIK 810DE powerline
Pokud budete potřebovat seznam modulů, které jsou k dispozici v systémech **powerline**, nahlédněte do popisu hardware /PHD/ v kapitole 1.1.

© Siemens AG 2004, Všechna práva vyhrazena

Horká linka

Internetová adresa

Funkce	810DE	840DE
Sada pro obrábění v 5 osách	-	-
Sada pro transformace (5 os)	-	-
Interní vícenásobná interpolace (> 4 osy)	-	-
Spirální interpolace 2D+6	-	-
Synchronizované akce, stupeň 2	-	O[1)]
Měření, stupeň 2	-	O[1)]
Adaptivní regulace	-	O[1)]
Kontinuální opracování	-	O[1)]
Využití kompilačních cyklů (OEM)	-	-
Vícerozměrná kompenzace průhybu	-	O[1)]

- Funkce není k dispozici

1) Chování funkce je omezeno

© Siemens AG 2004, Všechna práva vyhrazena

Struktura popisů

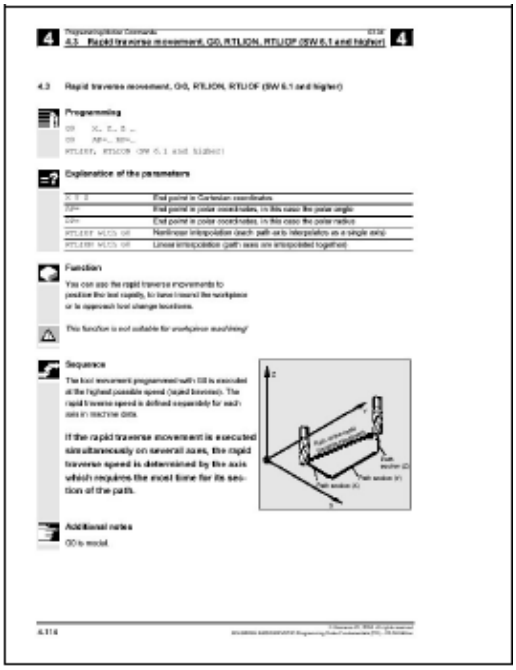
Pokud je to smysluplné a realizovatelné, jsou všechny cykly a možnosti programování popisovány pomocí stejné vnitřní struktury. Pomocí rozčlenění na různé informační úrovně můžete cíleně vyhledat právě ty informace, jež právě potřebujete.

1. Rychlý přehled

Jestliže hledáte zřídka používaný příkaz nebo význam nějakého parametru, můžete se podívat do přehledu, jak se funkce programuje, a vyhledat zde vysvětlení k jednotlivým příkazům a parametrům

Tyto informace se vždy nacházejí na začátku stránky.

Upozornění:
Z důvodu nedostatku místa není možné uvádět veškeré režimy zobrazování nabízené programovacím jazykem pro jednotlivé příkazy a parametry. Uváděná metoda programování je proto tou, která se nejčastěji používá ve výrobě.



© Siemens AG 2004, Všechna práva vyhrazena

2. podrobná vysvětlení

V teoretické části naleznete výslovný popis:



K čemu příkaz slouží?

Co přesně příkaz dělá?



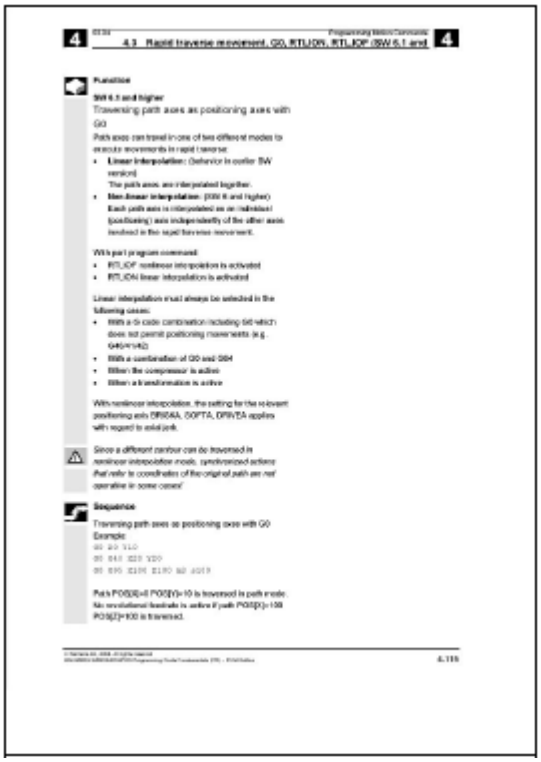
Jak je příkaz prováděn a programován?



Jaký je význam parametrů?

Co ještě bych měl vědět?

Teoretické části jsou určeny především jako výukový materiál, když se se systémem zpočátku učíte pracovat. Alespoň jednou byste si měli příručku podrobně prostudovat, abyste získali představu o rozsahu funkcí a možnostech Vašeho řídícího systému SINUMERIK.

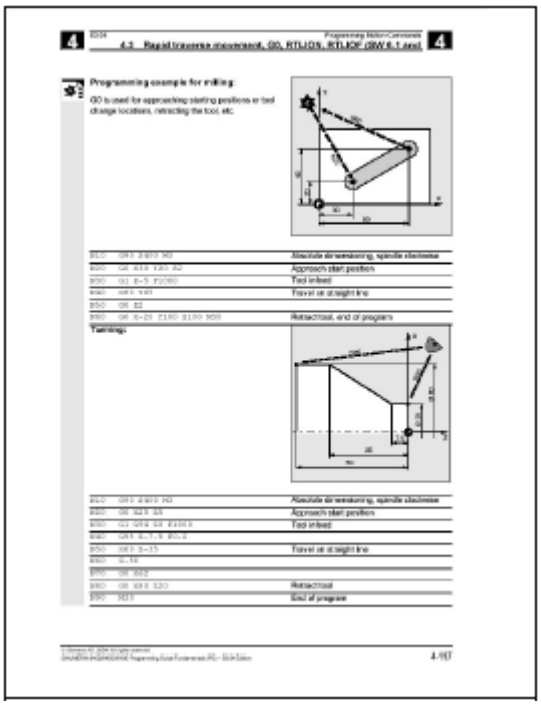


3. Od teorie k praxi



Příklady programování ukazují, jak mohou být příkazy aplikovány v praxi.

Za teoretickou částí naleznete pro téměř každý příkaz příklad jeho použití.



© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení symbolů


Postup


Vysvětlení


Funkce


Parametr


Příklad programování


Programování


Další upozornění


Křížové odkazy do jiné dokumentace a kapitol


Upozornění a nebezpečí



© Siemens AG 2004, Všechna práva vyhrazena

Základy

Váš systém SINUMERIK 840D/840Di/810D byl navržen a sestaven podle nejmodernějších technologických poznatků a podle schválených bezpečnostních předpisů a norem.

Další zařízení

Použití řídících systémů firmy Siemens může být kvůli specifickým účelům rozšířeno o speciální zařízení, přístroje a rozšíření dodávaná firmou SIEMENS.

Pracovníci obsluhy

S tímto zařízením smí pracovat výhradně **spolehliví** a **náležitě vyškolení** pracovníci, kteří disponují příslušným **oprávněním**. S řídícím systémem nesmí nikdy pracovat, a to ani dočasně, osoby, které nejsou znalé nebo školené.
Musí být jasně definovány **kompetence** pracovníků, kteří provádějí seřizování, nastavování, obsluhu a údržbu zařízení; náležité plnění těchtoúkolů musí být **kontrolováno**.

Chování

Před uvedením řídícího systému do provozu musí být zaručeno, že odpovědní pracovníci si prostudovali provozní příručky a že je pochopili. Společnost provozovatele zodpovídá také za **soustavné monitorování** celkového technického stavu řídícího systému (viditelné poruchy a poškození, změna chování za provozu).

© Siemens AG 2004, Všechna práva vyhrazena

Servis

Opravy se musí uskutečňovat v souladu s informacemi uvedenými v příručce pro servis a údržbu a smí být prováděny **výhradně osobami, které jsou speciálně vyškolené a kvalifikované** v dané technické odbornosti. Musí být dodrženy veškeré předpisy související s bezpečností.

Upozornění

Následující použití je považováno za **nesprávné a zbavující výrobce veškerých závazků:**

Jakékoli uplatnění, které neodpovídá pravidlům pro správné použití popsaným výše.

Pokud řídící systém **není v technicky naprosto dokonalém stavu** nebo pokud je provozován bez náležitého dodržování bezpečnostních předpisů a pokynů pro prevenci nehod a úrazů uvedených v této příručce.

Jestliže poruchy, které by mohly mít vliv na bezpečnost zařízení, nejsou odstraňovány **před** spuštěním řídícího systému.

Jakékoli **úpravy**, **přemostění** nebo **vypínání** částí řídícího systému, které jsou zapotřebí pro bezporuchový provoz, neomezené uplatnění a aktivní a pasivní bezpečnost.

Nesprávné použití přináší **nepředpokládatelná nebezpečí** pro:

- Život a končetiny obsluhujícího pracovníka
- Řídící systém, stroj nebo jiné hmotné statky vlastníka a uživatele.

© Siemens AG 2004, Všechna práva vyhrazena

V rámci této dokumentace se používají následující symboly a klíčová slova:



Upozornění
Tento symbol se v této dokumentaci objevuje vždy tam, kdy je zapotřebí upoutat pozornost na nějaké důležité informace.



Kdekoli v této dokumentaci naleznete u tohoto symbolu informace týkající se objednávání volitelného doplňku. Popisovaná funkce se může spouštět pouze tehdy, pokud řídící systém obsahuje příslušný volitelný doplněk.

Výstražná upozornění:
V této příručce se používají následující výstražná upozornění s odstupňovaným významem:



Nebezpečí
Označuje bezprostřední nebezpečnou situaci, která **bude** mít za následek smrt, vážný úraz nebo značné materiální škody, jestliže budou zanedbána příslušná bezpečnostní opatření.



Varování
Označuje potenciálně nebezpečnou situaci, která **by mohla** mít za následek smrt, vážný úraz nebo značné materiální škody, jestliže budou zanedbána příslušná bezpečnostní opatření.



Pozor
Tato výstraha (s výstražným trojúhelníkem) označuje potenciálně nebezpečnou situaci, která **by mohla** mít za následek menší nebo středně vážné zranění nebo materiální škody, pokud se jí nepodaří zabránit.

Pozor
Pokud je tato výstraha použita bez výstražného trojúhelníku, označuje potenciálně nebezpečnou situaci, která **by mohla** mít za následek hmotné škody, pokud jí nezabráníte.

© Siemens AG 2004, Všechna práva vyhrazena

Upozornění

Tato výstraha znamená, že pokud zanedbáte odpovídající pokyny, mohl by se vyskytnout nežádnoucí výsledek nebo nežádoucí stav.

© Siemens AG 2004, Všechna práva vyhrazena

Obsah

Flexibilní NC-programování 1-23

1.1 Proměnná a výpočetní parametr 1-24
1.2 Definice proměnné 1-26
1.3 Definice pole 1-31
1.4 Nepřímé programování 1-37
1.5 Přiřazení 1-42
1.6 Matematické operace/funkce 1-43
1.7 Porovnávací a logické operátory 1-45
1.8 Priorita operátorů 1-50
1.9 Možné konverze charakteristiky 1-51
1.10 Operace s řetězci znaků 1-52
1.10.1 Konverze charakteristiky 1-53
1.10.2 Zřetězení proměnných typu STRING 1-55
1.10.3 Změna psaní malých/velkých písmen 1-56
1.10.4 Délka řetězce 1-57
1.10.5 Hledání znaku/řetězce v proměnné typu STRING 1-57
1.10 6 Výběr dílčího řetězce 1-58
1.10.7 Selekce jednotlivého znaku 1-59
1.11 Příkaz CASE 1-61
1.12 Řídící struktury 1-63
1.13 Koordinace programu 1-67
1.14 Rutina přerušení 1-72
1.15 Změna osy, změna vřetene 1-80
1.16 NEWCONF: Aktivování strojních parametrů (od SW 4.3) 1-85
1.17 WRITE: Zápis do souboru (od SW 4.3) 1-85
1.18 DELETE: Vymazání souboru (od SW 4.3) 1-88
1.19 READ: Čtení řádků v souboru (od SW 5.2) 1-89
1.20 ISFILE: Soubor je k dispozici v uživatelské paměti NCK (od SW 5.2) 1-91
1.21 CHECKSUM: Vytvoření kontrolního součtu pole (od SW 5.2) 1-93

Technika podprogramů, technika maker 2-95

2.1 Použití podprogramů 2-96
2.2 Podprogram s mechanismem SAVE 2-97
2.3 Podprogramy s předáváním parametrů 2-98
2.4 Vyvolání podprogramů: L resp. EXTERN 2-103
2.5 Návrat z podprogramu s možností zadání parametrů (od SW 6.4) 2-107

© Siemens AG 2004, Všechna práva vyhrazena

2.1 Použití podprogramů 2-96
2.2 Podprogram s mechanismem SAVE 2-97
2.3 Podprogramy s předáváním parametrů 2-98
2.4 Vyvolání podprogramů: L resp. EXTERN 2-103
2.5 Návrat z podprogramu s možností zadání parametrů (od SW 6.4) 2-107
2.6 Podprogram s opakováním části programu: P 2-111
2.7 Modální podprogram: MCALL 2-111
2.8 Nepřímé volání podprogramu: CALL 2-112
2.9 Opakování části programu s nepřímým programováním (od SW 6.4) 2-114
2.10 Nepřímé volání programu programovaného v jazyce ISO: ISOCALL 2-115
2.11 Volání podprogramu se zadáním cesty a s parametry: PCALL 2-116
2.12 Rozšíření cesty při vyvolání podprogramu pomocí CALLPATH (od SW 6.4) 2-117
2.13 Potlačit zobrazení aktuální sady: DISPLOF 2-119
2.14 Potlačení zpracování blok po bloku: SBLOF, SBLON (od SW 4.3) 2-120
2.15 Zpracování externího podprogramu: EXTCALL (od SW 4.2) 2-126
2.16 Vyvolání podprogramu funkcí M-/T- 2-129
2.17 Cykly: Zadávání parametrů uživatelských cyklů 2-131
2.18 Technika maker: DEFINE…AS 2-135

Správa souborů a programů 3-139

3.1 Přehled 3-140
3.2 Paměť programů 3-141
3.3 Pracovní paměť 3-147
3.4 Definice uživatelských dat 3-150
3.5 Definice úrovní ochrany pro uživatelská data (GUD) 3-155
3.6 Automatická aktivace GUD a MAC (od SW 4.4) 3-157
3.7 Změna úrovní ochrany pro strojní a nastavované parametry 3-159
3.8 Úrovně ochrany pro příkazy NC-jazyka (od SW 7.1) 3-160
3.9 Změna atributů prvků NC- jazyka (od SW 6.4) 3-163
3.10 Příkaz ke strukturování SEFORM v editoru STEP (od SW 6.4) 3-170

Chráněné oblasti 4-173

4.1 Stanovení chráněných oblastí 4-174
4.2 Aktivace, deaktivace chráněných oblastí: CPROT, NPROT 4-178
4.3 Kontrola porušení chráněných oblastí, vymezení pracovního pole a softwarových koncových poloh 4-182

© Siemens AG 2004, Všechna práva vyhrazena

Speciální příkazy pohybu po dráze 5-189

5.1 Najetí na kódované pozice, CAC, CIC, CDC, CACP, CACN, ... 5-190
5.2 Splinová interpolace, ASPLINE, B-/CSPLINE, BAUTO, BNAT, BTAN 5-191
5.3 Kompresor COMPOF/ON, COMPCURV, COMPCAD, (SW 6.2) 5-199
5.4 Polynomická interpolace – POLY, POLYPATH (od SW 5) 5-207
5.5 Nastavitelná referenční dráha, SPATH, UPATH 5-213
5.6 Měření spínací sondou, MEAS, MEAW 5-217
5.7 Rozšířené funkce měření, MEASA, MEAWA, MEAC Od SW 4, opce) 5-220
5.8 Speciální funkce pro OEM-uživatele, G810 až G829 5-230
5.9 Redukce posuvu se zpomalením v rozích, G62, G621(od SW 6.1) 5-230
5.10 Programovatelné kritérium konce pohybu (od SW 5.1) 5-232
5.11 Programovatelná sada servo-parametrů (od SW 5.1) 5-235

Framy 6-237

6.1 Transformace souřadnic prostřednictvím proměnné framu 6-238
6.2 Přiřazení hodnot proměnným framů / framům 6-243
6.3 Hrubé a jemné posunutí 6-250
6.4 Posunutí ručním kolečkem 6-251
6.5 Externí posunutí počátku 6-252
6.6 Programování předdefinovaného posunutí, PRESETON 6-253
6.7 Deaktivace framů, DRFOF, G53, G153 a SUPA 6-254
6.8 Výpočet framu ze 3 změřených bodů v prostoru, MEAFRAME 6-255
6.9 NCU-globální framy (od SW 5) 6-259
6.9.1 Framy specifické pro určitý kanál 6-260
6.9.2 Framy s kanálovou působností 6-262

Transformace 7-269

7.1 Tří-, čtyř-, a pětiosová transformace: TRAORI 7-270
7.1.1 Programování orientace nástroje 7-274
7.1.2 Otáčení orientace nástroje: ORIROTA, ORIROTR, ORIROTT 7-279
7.1.3 Vztažný systém os orientace – ORIWKS, ORIMKS 7.281
7.1.4 Singulární postavení a jejich úprava 7-282
7.1.5 Osy orientace (od SW 5.2) ORIAXES, ORIVECT, ORIEULER, ORIRPY 7-283
7.1.6 Kartézský pohyb PTP (od SW 5.2) 7-286
7.1.7 Online korekce délky nástroje: TOFON, TOFOF (od SW 6.4) 7-290
7.2 Frézování na rotačních součástech: TRANSMIT 7-293
7.3 Transformace pláště válce: TRACYL 7-296
7.4 Šikmá osa: TRAANG 7-302
7.4.1 Programování šikmé osy: G05, G07, (od SW 5.3) 7-306
7.5 Okrajové podmínky při volbě transformace 7-307

© Siemens AG 2004, Všechna práva vyhrazena

7.6 Deaktivování transformace: TRAFOOF 7.309
7.7 Zřetězené transformace 7-310
7.8 Přepínatelné geometrické osy, GEOAX 7-313

Korekce nástroje 8-319

8.1 Paměť korekcí 8-320
8.2 Příkazy jazyka pro správu nástrojů 8-322
8.3 On-line korekce nástroje, PUTTOCF, PUTFTOC, FTOCON, FTOCOF 8-325
8.4 Udržování konstantní korekce rádiusu nástroje, CUTCONON (od SW 4) 8-330
8.5 Aktivování 3D korekčních parametrů nástroje, CUT3DC, CUT3DF, CUT3DFS/CUT3DFF 8-333
8.5.1 3D korekce rádiusu nástroje, frézování na obvodové ploše, frézování na čelní ploše 8-335
8.5.2 Typy nástrojů/výměna nástroje se změněnými rozměry G40/41/42 8-336
8.5.3 Korekce po dráze, zakřivení dráhy a hloubka zajíždění ISD 8-337
8.5.4 Posuv na vnitřních/vnějších rozích a průsečících, G450/G451 (od SW 5) 8-338
8.5.5 Obvodové 3D frézování s omezovacími plochami, CUT3DCC, CUT3DCCD 8-340
8.6 Orientace nástroje, ORIC, ORID, OSOF, OSC, OSS, OSSE 8-345
8.7 Volné předávání D-čísel, číslo břitu CE (od SW 5) 8-350
8.7.1 Kontrola D-čísla (CHKDNO) 8-351
8.7.2 Změna názvu D-čísla (GETDNO, SETDNO) 8-352
8.7.3 Zjištění T-čísla k předem zadanému D-číslu (GETACTTD) 8-353
8.7.4 Prohlášení D-čísla za neplatné 8-355
8.8 Kinematika držáku nástroje 8-355

Chování při pohybu po dráze 9-361

9.1 Tangenciální řízení,TAN, TANGON, TANGOF, TLIFT, TANGDEL 9-362
9.2 Vlečení, TRAILON, TRAILOF 9-369
9.3 Tabulky křivky, CTABDEF, CTABEND, CTABDEL, CTAB, CTABINV/CTABFNO 9-373
9.3.1 Příkazy s tabulkami křivky, CTABIB, CTABLOCK, CTABUNLOCK 9-373
9.3.2 Chování na okrajích tabulky křivky, CTABTSV/CTABTSP, CTABTMIN 9-383
9.3.3 Přístup na pozice a segmenty v tabulce křivky CTAB, CTABINV 9-386
9.4 Spojení os řídící hodnotou, LEADON, LEADOF 9-388
9.5 Průběh posuvu, FNORM, FLIN, FCUB, FPO 9-394
9.6 Zpracování programu s pamětí předběžného zpracování, STARTFIFO, STOPFIFO, STOPRE 9-399
9.7 Úseky programu s podmíněným přerušením, DELAYFSTON, DELAYFSTOF 9-401
9.8 Zablokování programového místa pro SERUPRO, IPTRLOCK, IPTRUNLOCK 9-407
9.9 Opětovné najíždění na konturu, REPOSA/L, REPOSQ/H, RMI/N, RMB/E 9-410

© Siemens AG 2004, Všechna práva vyhrazena

Synchronní pohybové akce 10-417

10.1 Struktura, všeobecné základy.....10-419
10.1.1 Programování a příkazové prvky.....10-421
10.1.2 Oblast platnosti: Identifikační číslo ID.....10-422
10.1.3 Klíčové slovo.....10-423
10.1.4 Akce.....10-426
10.1.5 Přehled synchronních akcí.....10-427

10.2 Základní prvky pro podmínky a akce.....10-429

10.3 Speciální proměnné v reálném čase pro synchronní akce.....10-432
10.3.1 Ukazatel/čítač $AC_MARKER[n].....10-432
10.3.2 Proměnná časovače $AC_TIMER[n], od SW 4.....10-432
10.3.3 Parametr synchronní akce $AC_PARAM[n].....10-433
10.3.4 Přístup k početním parametrům $Rxx.....10-434
10.3.5 Načítání/zápis strojních/nastavovaných parametrů, od SW 4.....10-435
10.3.6 Proměnné FIFO $AC_FIFO1[n] ... $AC_FIFO10[n], od SW 4.....10-436
10.3.7 Informace o typu bloku v interpolátoru (SW 6.4 a 7.1).....10-438

10.4 Operace v synchronních akcích.....10-441
10.4.1 Výstup pomocných funkcí.....10-441
10.4.2 Aktivování blokování načítání, RDISABLE.....10-442
10.4.3 Deaktivování zastavení předběžného zpracování, STOPREOF.....10-443
10.4.4 Vymazání zbytkové dráhy.....10-444
10.4.5 Vymazání zbytkové dráhy s přípravou, DELDTG, DELDTG (osa 1 až X)..10-444
10.4.6 Definice polynomu, FCTDEF, bloková synchronizace.....10-446
10.4.7 Řízení výkonu laseru.....10-448
10.4.8 Vyhodnocovací funkce SYNFCT.....10-449
10.4.9 AC regulace (aditivní).....10-450
10.4.10 AC regulace (multiplikativní).....10-451
10.4.11 Regulace vzdálenosti s omezenou korekcí.....10-452
10.4.12 On-line korekce nástroje, FTOC.....10-454
10.4.13 Pohyby při najíždění do polohy.....10-456
10.4.14 Polohování osy, POS.....10-458
10.4.15 Spuštění/zastavení osy, MOV.....10-458
10.4.16 Posuv osy, FA.....10-459
10.4.17 Softwarový koncový spínač.....10-459
10.4.18 Koordinace os.....10-460
10.4.19 Dosazení skutečné hodnoty.....10-461
10.4.20 Pohyby vřetena.....10-462
10.4.21 Vlečení: TRAILON, TRAILOF.....10-463
10.4.22 Spojení řídící hodnotou, LEADON, LEADOF.....10-464
10.4.23 Měření.....10-466
10.4.24 Stanovení/vymazání značky pro čekání, SETM, CLEARM (od SW 5.2).... 10.466
10.4.25 Reakce v případě chyby.....10.467
10.4.26 Najíždění na pevný doraz, FXS a FOCON/FOCOF.....10.467
10.4.27 Stanovení parametru s v synchronních akcích.....10.470
10.4.28 Stanovení aktuální korekce (override).....10.470
10.4.29 Vyhodnocování vytížení pomocí časové náročnosti synchronní akce.....10.471

10.5 Technologické cykly.....10-473
10.5.1 Zablokování, odblokování, přerušení, LOCK, UNLOCK, RESET.....10-475

10.6 Vymazání synchronní akce, CANCEL.....10-477

10.7 Okrajové podmínky.....10-477

© Siemens AG 2004, Všechna práva vyhrazena

Oscilační pohyb 11-481

11.1 Asynchronní oscilační pohyb 11-482

11.2 Oscilační pohyb řízený synchronními akcemi 11-489

Lisování a prostřihování 12-501

12.1 Aktivace, deaktivace 12-502
12.1.1 Příkazy SPOF, SON, PON, SONS, PONS, PDEALYON/OF 12-502
12.1.2 Použití M-příkazů 12-505

12.2 Automatické rozdělení dráhy 12-506
12.2.1 Rozdělení dráhy u dráhových os 12-507
12.2.2 Rozdělení dráhy u samostatných os 12-508
12.2.3 Příklady programování 12-510

Další funkce 13-513

13.1 Osové funkce AXNAME, SPI, ISAXIS, AXSTRING (od SW 6) 13-514

13.2 Volání funkce ISVAR ( ) (od SW 6.3) 13-516

13.3 Zadávání kompenzačních charakteristik:QECLRNON, QECLRNOF 13-518

13.4 Synchronní vřeteno 13-520

13.5 EG: Elektronická převodovka (od SW 5) 13-530
13.5.1 Definice elektronické převodovky: EGDEF 13-531
13.5.2 Aktivování elektronické převodovky 13-532
13.5.3 Deaktivování elektronické převodovky 13-535
13.5.4 Vymazání definice elektronické převodovky 13-536
13.5.5 Otáčkový posuv (G95) / elektronická převodovka (od SW 5.2) 13-536
13.5.6 Chování EG při Power ON, resetu, změně provozního režimu, hledání 13-537
13.5.7 Systémové proměnné elektronické převodovky 13-537

13.6 Rozšířené zastavování a stahování (od SW 5) 13-538
13.6.1 Reakce závislé na pohonu 13-539
13.6.2 Reakce ovládané NC systémem 13-540
13.6.3 Možné zdroje spouštění 13-544
13.6.4 Logické spouštěcí funkce: Obsluha zdrojů/reakcí 13-545
13.6.5 Aktivování 13-546
13.6.6 Obsluha generátoru / záloha pomocného obvodu 13-546
13.6.7 Zastavení závislé na pohonu 13-547
13.6.8 Zpětný pohyb závislý na pohonu 13-548
13.6.9 Příklad: Použití reakce závislé na pohonu 13-549

13.7 Linková komunikace (od SW 5.2) 13-550

13.8 Zásobník os (od SW 5.2) 13-553

13.9 Strojový čas programu/počítadlo obrobků (od SW 5.2) 13-555
13.9.1 Strojový čas programu 13-555
13.9.2 Počítadlo obrobků 13-557

13.10 Příkaz pro interaktivní vyvolání okna z výrobního programu: MMC (od SW 4.4) 13-559

13.11 Ovlivňování vedení pohybu 13-560
13.11.1 Procentuální korekce trhavých pohybů: JERKLIM 13-560
13.11.2 Procentuální korekce rychlosti: VELOLIM 13-561

13.12 Spojení Master/Slave 13-562

© Siemens AG 2004, Všechna práva vyhrazena

Uživatelské programy pro oddělování třísky.....14-567

14.1 Pomocné funkce pro oddělování třísky.....14-568
14.2 Příprava kontury - CONTPRON.....14-569
14.3 Dekódování kontury – CONTDCON (od SW 5.2).....14-576
14.4 Průsečík dvou konturových prvků – INTERSEC 14-580
14.5 Obrábění konturových prvků z tabulky – EXECTAB 14-582
14.6 Výpočet kruhových dat – CALCDAT.....14-583

Tabulky 15-585

15.1 Seznam příkazů.....15-586
15.2 Seznam systémových proměnných.....15-612

Přílohy A-613

A Zkratky A-613
B Pojmy A-622
C Rejstřík.....A-643
D Příkazy, identifikátory.....A-653

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky:

© Siemens AG 2004, Všechna práva vyhrazena

Flexibilní NC-programování 1-23

1.1 Proměnná a výpočetní parametr ... 1-24
1.2 Definice proměnné ... 1-26
1.3 Definice pole ... 1-31
1.4 Nepřímé programování ... 1-37
1.5 Přiřiřazení ... 1-42
1.6 Matematické operace/funkce ... 1-43
1.7 Porovnávací a logické operátory ... 1-45
1.8 Priorita operátorů ... 1-50
1.9 Možné konverze charakteristiky ... 1-51
1.10 Operace s řetězci znaků ... 1-52
1.10.1 Konverze charakteristiky ... 1-53
1.10.2 Zřetězení proměnných typu STRING ... 1-55
1.10.3 Změna psaní malých/velkých písmen ... 1-56
1.10.4 Délka řetězce ... 1-57
1.10.5 Hledání znaku/řetězce v proměnné typu STRING ... 1-57
1.10 6 Výběr dílčího řetězce ... 1-58
1.10.7 Selekce jednotlivého znaku ... 1-59
1.11 Příkaz CASE ... 1-61
1.12 Řídící struktury ... 1-63
1.13 Koordinace programu ... 1-67
1.14 Rutina přerušení ... 1-72
1.15 Změna osy, změna vřetene ... 1-80
1.16 NEWCONF: Aktivování strojních parametrů (od SW 4.3) ... 1-85
1.17 WRITE: Zápis do souboru (od SW 4.3) ... 1-85
1.18 DELETE: Vymazání souboru (od SW 4.3) ... 1-88
1.19 READ: Čtení řádků v souboru (od SW 5.2) ... 1-89
1.20 ISFILE: Soubor je k dispozici v uživatelské paměti NCK (od SW 5.2) ... 1-91
1.21 CHECKSUM: Vytvoření kontrolního součtu pole (od SW 5.2) ... 1-93

© Siemens AG 2004, Všechna práva vyhrazena

1.1 Proměnné a výpočetní parametry



Funkce

Užitím proměnných místo pevných hodnot můžete flexibilně utvářet program. Můžete tak reagovat na signály jako např. naměřené hodnoty, nebo můžete použitím proměnných jako požadované hodnoty tentýž program použít pro různé geometrie.

Společně s výpočtem proměnných a programovými skoky toto otvírá dovednému programátorovi možnost založit si vysoce flexibilní archív programů a ušetřit si tak mnoho programovací práce.



Druhy proměnných

Řízení rozlišuje 3 druhy proměnných:

Uživatelem definovaná proměnná	Uživatelem se jménem a typem definovaná proměnná, např.výpočetní parametr.
Výpočetní parametr	Speciální předdefinovaná výpočetní proměnná, pro kterou je určena adresa R s následujícím číslem. Předdefinované výpočetní proměnné jsou typu REAL.
Systémová proměnná	Řízením k dispozici dané proměnné, které mohou být zpracovány v programu (zápis, čtení). Systémové proměnné nabízejí přístup k posunutím počátku, korekcím nástroje, skutečným hodnotám, naměřeným hodnotám os, stavům řízení atd. (význam systémových proměnných viz příloha)

© Siemens AG 2004, Všechna práva vyhrazena

Typy proměnných

Typ	Význam	Oblast hodnot
INT		$\pm(2^{31} - 1)$
REAL	reálná čísla (zlomková čísla desetinnou tečkou, LONG REAL podle IEEE)	$\pm(10^{-300} \ldots 10^{+300})$
BOOL	hodnoty pravdivosti: TRUE (1) a FALSE (0)	1,0
CHAR	1 znak ASCII, odpovídající kódu	0…255
STRING	řetězec znaků, číslice znaků v […], maximálně 200 znaků	sled hodnot 0…255
AXIS	jen jména os (adresy os)	všechny obsažené v kanálu identifikátory os a vřeten
FRAME	Geometrické údaje pro posunutí, otáčení, změnu měřítka, zrcadlové převracení, viz kapitola 4.	

Výpočtová proměnná

Pod adresou R je k dispozici bez další definice standardně 100 výpočtových proměnných typu REAL.

Přesný počet výpočtových proměnných (maximálně 1000) je stanoven pomocí strojních parametrů.

Příklad: `R10=5`

Systémové proměnné

Řízení dává k dispozici systémové proměnné, které jsou k dispozici a mohou být zpracovány ve všech běžných programech.

© Siemens AG 2004, Všechna práva vyhrazena

Pro zvláštní identifikaci začíná pojmenování systémových proměnných znakem $. Potom následují specifická označení.

Přehled typů systémových proměnných

1. písmeno	Význam
$M	Strojní parametry
$S	Nastavované parametry
$T	Data správy nástrojů
$P	Programované hodnoty
$A	Aktuální hodnoty
$V	Servisní data

2. písmeno	význam
N	NCK-globální
C	Specifický pro kanál
A	Specifický pro osu

Příklad: SAA_IM
Znamená: aktuální axiální skutečná hodnota v souřadném systému stroje.

1.2 Definování proměnné

Uživatelem definovaná proměnná
Vedle předdefinovaných proměnných může programátor definovat vlastní proměnné a dosadit do nich hodnoty.
Lokální proměnné (LUD) platí jenom v programu, ve kterém jsou definovány.
Globální proměnné (GUD) platí ve všech programech.
Od SW 4.4:
Pomocí strojních dat jsou v hlavním programu definované lokální uživatelské proměnné (LUD) předefinovány na programové globální uživatelské proměnné (PUD).

Výrobce stroje
Viz údaje výrobce stroje

Jsou-li definovány v hlavním programu, platí také ve všech úrovních vyvolaných podprogramů. Jsou vytvořeny startem výrobního programu a vymazány koncem výrobního programu popř. pomocí RESET.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:

```
$MN_LUD_EXTENDED_SCOPE=1

PROC MAIN              ; hlavní program
DEF INT VAR1           ; definice PUD
…
SUB2                   ; vyvolání podprogramu
…
M30

PROC SUB2              ; podprogram SUB2
DEF INT VAR2           ; definice LUD
…
IF (VAR1==1)           ; čtení a zápis PUD
  VAR1=VAR1+1

  VAR2=1               ; zápis LUD
ENDIF
SUB3                   ; vyvolání podprogramu
…
M17
PROC SUB3              ; podprogram SUB3
…
IF (VAR1==1)           ; čtení PUD
  VAR1=VAR1+1          ; čtení a zápis PUD

  VAR2=1               ; chyba: LUD ze SUB2
                       ; není známa
ENDIF
…
M17
```



Když je definován strojní parametr `$MN_LUD_EXTENDED_SCOPE`, není již možné v hlavních programech nebo podprogramech definovat proměnnou se stejným jménem.

Pojmenování proměnných
Jméno proměnné sestává z maximálně 31 znaků. Oba první znaky musí být písmeno nebo podtržítko.

Znak „$“ nemůže být použit pro proměnnou definovanou uživatelem, protože tento znak je použit pro systémové proměnné.

© Siemens AG 2004, Všechna práva vyhrazena

Programování

```
DEF INT NAME
```
nebo DEF INT name=hodnota

DEF REAL name

nebo `DEF REAL name1, name2=3, name4`

nebo `DEF REAL name [index pole1]=[index pole2]`

```
DEF BOOL name
```

```
DEF CHAR name
```
nebo `DEF CHAR name [index pole]=(„A", „B", …)`

```
DEF STRING [délka stringu] name
```

```
DEF AXIS name
```
nebo `DEF AXIS name [index pole]`

```
DEF FRAME name
```

Není-li proměnné při definování přiřazena žádná hodnota, je systémem dosazena nula.

Proměnné musí být před jejich použitím na začátku programu definovány. Definování musí proběhnout ve vlastním bloku, v jednom bloku může být definován jen jeden typ proměnné.

Vysvětlení

`INT`	Typ proměnné Integer, tzn. celočíselný
`REAL`	Typ proměnné Real, tzn. zlomkové číslo s desetinou tečkou
`BOOL`	Typ proměnné Bool, tzn. 1 nebo 0 (TRUE nebo FALSE)
`CHAR`	Typ proměnné Char, tzn. znak odpovídající ASCII kódu (0 až 255)
`STRING`	Typ proměnné String, tzn. řetězec znaků
`AXIS`	Typ proměnné Axis, tzn. adresy hřídelí a vřetena
`FRAME`	Typ proměnné Frame, tzn. geometrické údaje
`name`	Jméno proměnné

© Siemens AG 2004, Všechna práva vyhrazena

Příklady programování

Typ proměnné INT	
`DEF INT ANZAHL`	Bude vytvořena proměnná typu Integer se jménem ANZAHL (počet). Systémem bude dosazena nula.
`DEF INT ANZAHL=7`	Bude vytvořena proměnná typu Integer se jménem ANZAHL. Proměnná má počáteční hodnotu 7.
Typ proměnné REAL	
`DEF REAL TIEFE`	Bude vytvořena proměnná typu REAL se jménem TIEFE (hloubka). Systémem bude dosazena nula (0.0).
`DEF REAL TIEFE=6.25`	Bude vytvořena proměnná typu REAL se jménem TIEFE. Proměnná má počáteční hodnotu 6.25.
`DEF REAL TIEFE=3.1, LAENGE=2, ANZAHL`	V jednom řádku může být definováno i více proměnných.
Typ proměnné BOOL	
`DEF BOOL WENN_ZUVIEL`	Bude vytvořena proměnná typu BOOL se jménem WENN_ZUVIEL (když příliš mnoho) Systémem bude dosazena 0 (FALSE).
`DEF BOOL WENN_ZUVIEL=1` **nebo** `DEF BOOL WENN_ZUVIEL=TRUE` **nebo** `DEF BOOL WENN_ZUVIEL=FALSE`	Bude vytvořena proměnná typu BOOL se jménem WENN_ZUVIEL.
Typ proměnné CHAR	
`DEF CHAR GUSTAV_1=65`	Proměnným typu Char může být přiřazena kódová hodnota pro odpovídající ASCII znak nebo
`DEF CHAR GUSTAV_1="A"`	Přímo znak ASCII (kódová hodnota 65 odpovídá písmenu „A“).
Typ proměnné STRING	
`DEF STRING[6] MUSTER_1="ANFANG"`	Proměnné typu STRING mohou pojmout řetězec n znaků. Maximální počet znaků je v hranaté závorce za typem proměnné.

© Siemens AG 2004, Všechna práva vyhrazena

Typ proměnné AXIS	
`DEF AXIS ACHSNAME= (X1)`	Proměnné typu AXIS mají jméno ACHSNAME (jméno osy) a obsahují identifikátor osy kanálu – zde X1. (Jména os s rozšířenou adresou jsou v kulatých závorkách)
Typ proměnné FRAME	
`DEF FRAME SCRAEG_1`	Proměnné typu FRAME mají jméno SCHRAEG_1 (šikmo).



Další upozornění

Proměnná typu AXIS obsahuje identifikátory osy a vřetena jednoho kanálu.
Upozornění:
Jména os s rozšířenou adresou musí být psána v kulatých závorkách.



Příklad programování s programově lokálními proměnnými

```
DEF INT ZAEHLER
SCHLEIFE: GO X…                          ; Smyčka
ZAEHLER=ZAEHLER+1
IF ZAEHLER<50 GOTOB SCHLEIFE
M30
```



Příklad programování
Dotaz na geometrické osy, které jsou k dispozici

```
DEF AXIS ABZISSE;                        ;1. geometrická osa
IF ISAXIS(1) = = FALSE GOTOF WEITER
ABZISSE = $P_ANX1
…
WEITER
```

Nepřímé programování vřetena

```
DEF AXIS SPINDLE
SPINDLE=(S1)
OVRA(SPINDLE)=80                         ;korekce vřetena = 80 %
SPINDLE=(S3)
…
```


© Siemens AG 2004, Všechna práva vyhrazena

1.3 Definování pole



Programování

```
DEF CHAR NAME [n,m]
DEF INT NAME [n,m]
DEF REAL NAME [n,m]
DEF AXIS NAME [n,m]
DEF FRAME NAME [n,m]
DEF STRING [délka řetězce] NAME[m]
DEF BOOL [n,m]
```

Vysvětlení

`INT NAME [n,m]` `REAL NAME [n,m]`	Typ proměnné (CHAR,INTEGER,REAL,AXIS,FRAME,BOOL) n = velikost pole pro 1. rozměr m = velikost pole pro 2. rozměr
`DEF STRING [délka řetězce] NAME [m]`	Typ dat string může být definován pouze jednorozměrnými poli.
`NAME`	Jméno proměnné

Pro typ BOOL platí stejná velikost paměti jako pro typ CHAR.

Do SW3.
Maximální velikost pole je nastavena prostřednictvím strojních parametrů.



Výrobce
Viz údaje výrobce

Typ	Potřeba paměti pro prvek pole
BOOL	1 byte
CAR	1 byte
INT	4 byty
REAL	8 byty
STRING	Délka řetězce + 1
FRAME	~ 400 bytů, v závislosti na počtu os
AXIS	4 byty

Maximální velikost pole určuje velikost paměťových bloků, ve kterých je paměť proměnné uchovávána. Neměla by být nastavena větší, než je nutné.
Standard: 812 bytů
Nejsou-li definována žádná velká pole, volte: 256 bytů.

© Siemens AG 2004, Všechna práva vyhrazena

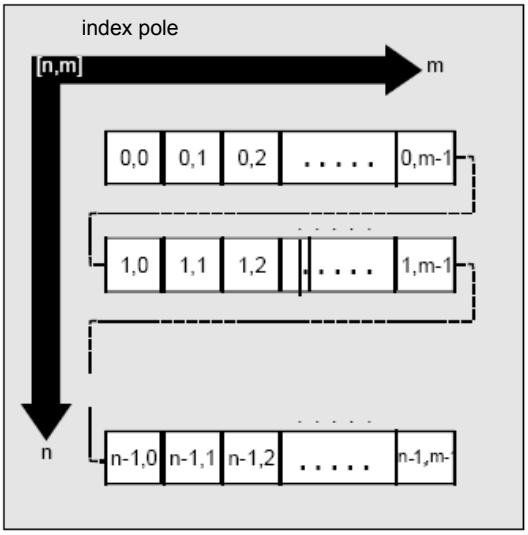
Od SW4:
Pole může být větší než paměťový blok.
Hodnota strojního parametru pro velikost bloku by měla být zvolena tak, aby k rozdělení polí docházelo výjimečně.
Standard 256 bytů
Potřeba paměti na prvek: viz výše.
Příklad:
Globální uživatelská data by měla obsahovat strojní parametry PLC pro zapnutí a vypnutí řízení (definice BOOL-polí).

Další upozornění
Mohou být definována pole s maximálně 2 rozměry.

Pole s proměnnými typu STRING smějí být pouze jednorozměrná. Délka řetězce je udávána podle datového typu string.

Index pole
K prvkům jednoho pole je přístup pomocí indexu pole. Pomocí tohoto indexu pole mohou být prvky pole buď obsazeny hodnotami nebo mohou být čteny hodnoty prvků polí.

První prvek pole začíná indexem [0,0]; při velikosti pole [3,4] je například maximální možný index pole [2,3].



© Siemens AG 2004, Všechna práva vyhrazena

Ve výše uvedeném příkladu byly hodnoty při inicializaci zvoleny tak, aby současně představovaly index prvku pole. Tak je patrné pořadí jednotlivých prvků pole.

Inicializace polí

Prvkům polí mohou být během průběhu programu nebo také již při definici pole přiřazeny inicializační hodnoty

U dvojrozměrných polí je nejprve inkrementován pravý index pole

Inicializace pomocí seznamů hodnot, SET

1. Možnosti při definici pole

```
DEF Typ VARIABLE = SET (WERT)
DEF Typ FELD [m] = SET (Wert, Wert,…)
```

- Je přiřazeno tolik prvků pole, kolik je programováno inicializačních hodnot.
- Prvky pole bez hodnot (mezery v seznamu) jsou automaticky obsazeny 0.
- Je-li programováno více hodnot než je k dispozici zbývajících prvků pole, spustí systém alarm.

Příklad:

```
DEF REAL FELD[2,3] = (10, 20, 30, 40)
```

Při definici pole můžete dle libosti použít příkaz SET.

© Siemens AG 2004, Všechna práva vyhrazena

2. Možnosti v průběhu programu

```
FELD[n,m]= SET(hodnota, hodnota, hodnota,…)
FELD[n,m]= SET(výraz, výraz, výraz,…)
```

- Inicializace proběhne jako při definování.
- Jako hodnoty jsou zde možné i výrazy.
- Inicializace začíná u naprogramovaných indexů pole. Takto se dají cíleně obsazovat hodnotami i dílčí pole.

Příklad:

```
DEF INT FELD [5,5]
FELD [0,0] = SET (1,2,3,4,5)
FELD [2,3] = SET (PROMĚNNÁ, 4*5.6)
```

U osových proměnných nesmí dojít k přeběhu indexu osy:
Příklad:
Inicializace v jednom řádku

```
$MA_AX_VELO_LIMIT [1, AX1] = SET(1.1,2.2,3.3)
```

Odpovídá:

```
$MA_AX_VELO_LIMIT [1, AX1] = 1.1
$MA_AX_VELO_LIMIT [2, AX1] = 2.2
$MA_AX_VELO_LIMIT [3, AX1] = 3.3
```


© Siemens AG 2004, Všechna práva vyhrazena

Inicializace pomocí stejných hodnot, REP

1. Možnosti při definování pole

```
DEF TYP FELD[n,m] = REP(hodnota)
```

Všechny prvky pole jsou obsazeny stejnou hodnotou (konstantou).

Proměnné typu FRAME se nedají inicializovat.

Příklad:
```
DEF REAL FELD5[10,3] = REP(9.9)
```

2. Možnosti v průběhu programu

```
FELD[n,m] = REP(hodnota)
FELD[n,m] = REP(výraz)
```

- Jako hodnoty jsou zde možné také výrazy
- Všechny prvky pole jsou inicializovány stejnou hodnotou.
- Inicializace začíná u naprogramovaných indexů pole
- Takto se dají cíleně obsazovat hodnotami i dílčí pole.

Proměnné typu FRAME jsou přípustné a mohou takto být velmi jednoduše inicializovány.

Příklad:
Inicializace všech prvků jednou hodnotou

```
DEF FRAME FRM[10]
FRM[5] = REP(CTRANS (X,5))
```

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Inicializace kompletních polí proměnných.
V daném případě aktuální obsazení vidíte na obrázku.

```
N10 DEF REAL FELD1[10,3] = SET(0, 0, 0, 10, 11, 12, 20, 20, 20, 30,
30, 30, 40, 40, 40,)
N20 FELD1[0,0] = REP(100)
N30 FELD1[5,0] = REP(-100)
N40 FELD1[0,0] = SET(0, 1, 2, -10, -11, -12, -20, -20, -20, -30, , , ,
-40, -40, -50, -60, -70)
N50 FELD1[8,1] = SET(8.1, 8.2, 9.0, 9.1, 9.2)
```

Index pole 2

[1,2]	N10: inicializace při definování		
	0	1	2
0	0	0	0
1	10	11	12
2	20	20	20
3	30	30	30
4	40	40	40
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
	Prvky pole [5,0] až [9,2] byly inicializovány prostřednictvím předem definované hodnoty (0.0).		

N20/N30: inicializace s identickou hodnotou		
0	1	2
100	100	100
100	100	100
100	100	100
100	100	100
100	100	100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100

N40/N50: inicializace s různými hodnotami		
0	1	2
0	1	2
-10	-11	-12
-20	-20	-20
-30	0	0
0	-40	-40
-50	-60	-70
-100	-100	-100
-100	-100	-100
-100	8.1	8.2
9.0	9.1	9.2
Prvky pole [3,1] až [4,0] byly inicializovány prostřednictvím předem definované hodnoty (0.0). Prvky pole [6,0] až [8,0] nebyly změněny		

1

© Siemens AG 2004, Všechna práva vyhrazena

1.4 Nepřímé programování



Pomocí nepřímého programování mohou být programy využívány velmi univerzálně. Přitom je rozšířená adresa (index) vždy nahrazena proměnnou vhodného typu.



Všechny adresy jsou zadatelné parametry kromě:

- N – číslo bloku
- G – G-příkaz
- L – podprogram

Pro všechny nastavitelné adresy není možné nepřímé programování.
Příklad: X[1] namísto X1 není přípustné.



Programování

```
ADRESSE [INDEX]
```



Příklady programování

Vřeteno

```
S1=300
```

Přímé programování

```
DEF INT SPINU=1
S [SPINU] = 300
```

Nepřímé programování:
Počet otáček 300 ot/min pro vřeteno, jehož číslo je obsaženo v proměnné SPINU (v tomto příkladu 1).

Posuv

```
FA [U] = 300
```

Přímé programování

```
DEF AXIS AXVAR2=U
FA [AXVAR2] = 300
```

Nepřímé programování:
Posuv pro polohovací osu, jejíž jméno adresy je obsaženo v proměnných typu AXIS s názvem proměnné AXVAR2

Naměřená hodnota

```
$AA_MM[X]
```

Přímé programování

```
DEF AXIS AXVAR3=X
$AA_MM[AXVAR3]
```

Nepřímé programování:
Naměřená hodnota v souřadnicích stroje pro osu, jejíž jméno je obsaženo v proměnné AXVAR3.

© Siemens AG 2004, Všechna práva vyhrazena

Prvek pole

`DEF INT FELD1[4,5]`	Přímé programování
`DEFINE DIM1 AS 4` `DEFINE DIM2 AS 5` `DEF INT FELD[DIM1, DIM2]` `FELD[DIM1-1, DIM2-1]=5`	Nepřímé programování: U rozměrů pole musí být veličiny pole zadány jako pevné hodnoty pole.

Instrukce pro osu s osovými proměnnými

`X1=100 X2=200`	Přímé programování
	Nepřímé programování:
`DEF AXIS AXVAR1 AXVAR2`	Definování proměnných
`AXVAR1=(X1)  AXVAR2=(X2)`	Přiřazení jmen osám. Pohybovat se budou osy, které jsou v proměnných uloženy na 100 resp. 200.
`AX[AXVAR1]=100  AX[AXVAR2]=200`	

Interpolační parametr s osovými proměnnými

`G2 X100 I20`	Přímé programování
	Nepřímé programování:
`DEF AXIS AXVAR1=X`	Definování a přiřazení názvu osy
`G2 X100 IP[AXVAR1]=20`	Nepřímé programování údaje středu

Nepřímé vyvolání podprogramu

`CALL „L" << R10`	Vyvolání programu, jehož číslo je umístěno v R10



Další upozornění

Také R-parametry mohou být chápány jako pole s jedním rozměrem se zkráceným způsobem zápisu (R10 odpovídá R[10]).

© Siemens AG 2004, Všechna práva vyhrazena

Nepřímé programování G-kódu od SW 5

```
G[<Gruppen-Index>] = <Integer/Real-Variable>
```

Nepřímé programování G-kódu pomocí proměnné pro efektivní programování cyklů

Význam parametrů

G<Gruppen-Index	Celočíselná konstanta, pomocí které je zvolena skupina G-kódu.
<Integer/Real-Variable>	Proměnná typu Integer nebo Real pomocí které je zvoleno číslo G-kódu.

Funkce

Nepřímé programování G-kódu od (SW 5)

Nepřímé programování G-kódu pomocí proměnné umožňuje efektivní programování cyklů. K tomu jsou k dispozici dva parametry:

- G-kód-skupiny celočíselná konstanta
- G-kód-čísla proměnná typu Integer/Real

Platné skupiny G-kódu

Mohou být nepřímo programovány jen skupiny G-kódů s modální platností.
Skupiny G-kódů s blokovou platností jsou odmítnuty s alarmem 12470.

Platná čísla G-kódu

V nepřímém programování G-kódu nejsou dovoleny žádné aritmetické funkce.
Číslo G-kódu musí být uloženo v proměnné typu Integer nebo Real. Neplatná čísla G-kódu jsou odmítnuta s alarmem 12475.
Nutný výpočet čísla G-kódu musí proběhnout na samostatném řádku výrobního programu před nepřímým programováním G-kódu.

Další upozornění

Všechny platné G-kódy jsou uvedeny v Příručce programování - Základy, v kapitole „Seznam G-funkcí / podmínek dráhy“ v různých skupinách. Viz Příručka programování – Základy, „Tabulky“.

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování

Nepřímé programování G-kódu

; Nastavitelné posunutí počátku, skupina G-kódů 8

```
N1010 DEF INT INT_VAR
N1020 INT_VAR = 2
...
N1090 G[8] = INT_VAR G1 X0 Y0          ;G54
N1100 INT_VAR = INT_VAR +1             ;výpočet G-kódu
N1110 G[8] = INT_VAR G1 X0 Y0          ;G55
```

;Volba roviny, skupina G-kódů 6

```
N2010 R10 = SP_GG[6]                   ;čtení G-kódu pro aktuální rovinu
...
N2090 G[6] = R10                       ;G17 – G19
```


© Siemens AG 2004, Všechna práva vyhrazena

Zpracování řetězce jako řádku výrobního programu

`EXESTRING (<String-Variable>)`

Příkazem EXESTRING je nepřímo zpracován jeden řádek výrobního programu

Význam parametrů

<String-Variable>	Parametr typu string, který je předán pomocí EXESTRING

Funkce

EXESTRING (od SW 6.4)

Příkazem výrobního programu EXESTRING je předán jako parametr řetězec obsahující řádek zpracovávaného výrobního programu.

Další upozornění

Mohou být předávány všechny konstrukty výrobního programu, které mohou být programovány ve zpracovávané části výrobního programu. Vyloučeny jsou tak pokyny PROC a DEF jakož i obecně použití souborů INI a DEF.

Příklad programování

Nepřímé zadání řádku výrobního programu

`N100 DEF STRING[100] BLOCK`	Proměnná typu string pro převzetí řádku výrobního programu
`N110 DEF STRING[10] MFCT1 = „M7"`	
`N200 EXESTRING (MFCT1 << „M4711")`	Zpracování řádku výrobního programu „M7 M4711“
`N300 R10 = 1`	
`N310 BLOCK = „M3"`	
`N320 IF (R10)`	
`N330 BLOCK = BLOCK << MFCT1`	
`N340 ENDIF`	
`N350 EXESTRING (BLOCK)`	Zpracování řádku výrobního programu „M7 M4711“

© Siemens AG 2004, Všechna práva vyhrazena

1.5 Přiřazení

Proměnným/výpočtovým parametrům mohou být v programu přiřazeny hodnoty vhodného typu.

Přiřazení vyžaduje přitom vždy vlastní blok, v rámci jednoho bloku je možných více přiřazení. Přiřazení adresám os (instrukce posuvu) vyžaduje samostatný blok oproti přiřazení proměnných.

Příklad programování

`R1=10.518 R2=4 VARI1=45 VARI1=R4 X=47.11 Y=R2`	Přiřazení numerické hodnoty
`R1=R3 VARI1=R4`	Přiřazení proměnné vhodného typu
`R4=-R5 R7=-VARI8`	Přiřazení s protikladným znaménkem (dovoleno jen u typu INT a REAL)

Přiřazení proměnné typu string

V rámci jednoho řetězce CHAR nebo STRING je rozlišováno mezi psaním velkých a malých znaků. Má-li být součástí řetězce znaků ´ nebo “, musí být tento znak nebo řetězec ohraničen následující závorkou ´…´.

Příklad:

```
MSG(„Viene lavorata l´´´ultima
figura“)
```

Bude znamenat na obrazovce text ´Viene lavorata l´ultima figura´.

Nezobrazitelné znaky mohou být předávány jako binární nebo hexadecimální konstanta.

© Siemens AG 2004, Všechna práva vyhrazena

1.6 Matematické operace/funkce

Matematické funkce jsou přednostně použitelné pro R-parametry a proměnné (nebo konstanty a funkce) typu REAL. Přípustné jsou také typy INT a CHAR.

U matematických operací platí obvyklý matematický způsob zápisu. Priority ve zpracování jsou stanoveny pomocí kulatých závorek. Pro trigonometrické a k nim inverzní funkce platí udání stupňů (pravý úhel = 90°).

Operátory / matematické funkce

+	Sčítání
−	Odčítání
*	Násobení
/	Dělení **Pozor:** (typ INT)/(typ INT)=(typ REAL);Příklad:3/4= 0.75
DIV	Dělení, pro typ proměnné INT a REAL **Pozor:** (typINT)DIV(typINT)=(typINT);Příklad:3DIV4=0
MOD	Modulo-dělení (INT nebo REAL) dá zbytek INT dělení, např. 3 MOD 4=3
:	Řetězový operátor (u proměnných FRAME)
Sin ()	Sinus
COS ()	Cosinus
TAN ()	Tangens
ASIN ()	Arcussinus
ACOS ()	Arcuscosinus
ATAN2(,)	Arcustangens2
SQRT ()	Druhá odmocnina
ABS ()	Absolutní hodnota
POT ()	2. Mocnina
TRUNC ()	Celočíselná část
ROUND ()	Zaokrouhlení na celé číslo
LN ()	Přirozený logaritmus
EXP ()	Exponenciální funkce
CTRANS ()	Posunutí
CROT ()	Otáčení
CSCALE ()	Změna měřítka
CMIRROR ()	Zrcadlové převrácení

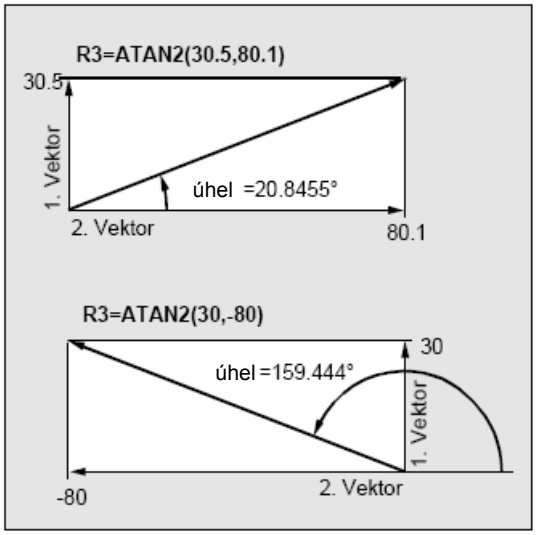
© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

`R1=R1+1`	Nové R1=staréR1+1
`R1R2+R3 R4=R5-R6 R7=R8*R9`	
`R10=R11/R12 R13=SIN(25.3)`	
`R14=R1*R2+R3`	Bodový výpočet předchází čárkovému výpočtu.
`R14=(R1+R2)*R3`	Nejprve je vypočtena závorka
`R15=SQRT(POT(R1)+POT(R2))`	Vnitřní závorka je zrušena R15= druhá odmocnina z (R12+R22)
`RESFRAME= FRAME1:FRAME2` `FRAME3=CTRANS (…):CROT (…)`	Pomocí řetězového operátoru jsou framy spojeny do výsledného framu nebo jsou součástem framů přiřazeny hodnoty.

Matematické funkce ATAN2(,)

Funkce vypočte ze dvou vůči sobě kolmo ležících vektorů úhel součtu vektorů. Výsledek leží v oblasti čtyř kvadrantů (-180<0<+180°). Základem pro vztah úhlu je vždy 2. hodnota v pozitivním směru.



© Siemens AG 2004, Všechna práva vyhrazena

1.7 Porovnávací a logické operátory



Porovnávací operátory

Porovnávací operátory jsou použitelné pro proměnné typu CHAR, INT, REAL, a BOOL. U typu CHAR je porovnávána kódová hodnota.

U typu STRING, AXIS a FRAME jsou možné: == a <>.

Výsledek srovnávacích operátorů je vždy typu BOOL.

Porovnávací operace mohou být použity např. k formulování podmínky skoku. Porovnatelné jsou při tom také složité výrazy.



Význam porovnávacích operátorů

==	Rovná se
<>	Nerovná se
>	Větší
<	Menší
>=	Větší nebo stejný
<=	Menší nebo stejný



Příklad programování

```
IF R10>=100 GOTOF ZIEL
```

nebo

```
R11=R10>=100
IF R11 GOTOF ZIEL
```

Výsledek porovnání R10<=100 je nejprve prozatímně uložen do paměti R11.

© Siemens AG 2004, Všechna práva vyhrazena

Korekce přesnosti u chyb porovnávání

```
TRUNC (R1*1000)
```

Příkaz TRUNC oddělí celočíselnou část operandu po vynásobení faktorem přesnosti.

Funkce

Nastavitelná přesnost u porovnávacích příkazů
Data výrobního programu typu REAL jsou interně zobrazena v IEEE formátu se 64 bity. Na základě této formy zobrazení mohou být vyobrazena nepřesně desetinná čísla, která při porovnávání s ideálně vypočtenými hodnotami mohou vést k neočekávaným výsledkům.

Relativní shodnost
Aby formou zobrazení vyvolané nepřesnosti negativně neovlivňovaly běh programu, kontroluje se u srovnávacích příkazů nikoli absolutní shodnost, nýbrž relativní shodnost.

Do SW 6.3
Zohledněná relativní shodnost 10^{-12} při operacích:

- shodnost (==)
- nerovnost (<>)
- větší-rovno (>=)
- menší-rovno (<=)
- větší/menší (><) s absolutní shodností

Od SW 6.4
Zohledněná relativní shodnost 10^{-12} při operacích:

- je větší (>)
- je menší (<)

Upozornění pro programování

Porovnání s daty typu REAL jsou z uvedených důvodů obecně spojeny s určitou mírou nepřesnosti. Při neakceptovatelných odchylkách se musí přejít na INTEGER- výpočet tak, že operandy jsou násobeny faktorem přesnosti a poté je oddělena příkazem TRUNC celočíselná část.

© Siemens AG 2004, Všechna práva vyhrazena

Synchronní akce
Popsané chování porovnávacích příkazů platí i u synchronních akcí.
Kompatibilita
Z důvodů kompatibility může být kontrola relativní rovnosti u (>) a (<) deaktivována zadáním MD 10280:PROG_FUNCTION_MASK Bit0 = 1.

Příklady programování

Problémy s přesností

`N40 R1=61.01 R2=61.02 R3=0.01`	Přiřazení počátečních hodnot
`N41 IF ABS (R2=R1) > R2 GOTOF FEHLER`	Skok bude proveden (do SW 6.3)
`N42 M30`	Konec programu
`N43 FEHLER: SETAL (66000)`	

`R1=61.01 R2=61.02 R3=0.01`	Přiřazení počátečních hodnot
`R11=TRUNC (R1*1000) R12=TRUNC (R2*1000) R13=TRUNC (R3*1000)`	Korekce přesnosti
`IF ABS (R12-R11) > R13 GOTOF FEHLER`	Skok nebude proveden
`M30`	Konec programu
`FEHLER:SETAL (66000)`	

Vytvoření kvocientu obou operandů a jeho vyhodnocení

`R1=61.01 R2=61.02 R3=0.01`	Přiřazení počátečních hodnot
`IF ABS ((R2-R1)/R3)-1) > 10EX-5 GOTOF FEHLER`	Skok nebude proveden
`M30`	Konec programu
`FEHLER: SETAL (66000)`	

© Siemens AG 2004, Všechna práva vyhrazena

Logické operátory

Logické operátory slouží ke spojení pravdivostních hodnot. AND, OR, NOT a XOR jsou použitelné jen pro proměnné typu BOOL. Přes implicitní konverzi typů jsou použitelné také pro data typu CHAR, INT a REAL.

Mezi operandy typu BOOL a operátory musí být vepsány mezery.

U logických operací (typu BOOL) platí pro data typu BOOL, CHAR, INT a REAL:
0 odpovídá FALSE
nerovné 0 odpovídá TRUE

Význam logických operátorů

`AND`	A
`OR`	NEBO
`NOT`	Negace
`XOR`	Exkluzivní NEBO

V aritmetických výrazech může být pomocí kulatých závorek stanoveno pořadí zpracování všech operací a tak je umožněno odchýlení se od normálních pravidel priority.

Příklad programování

```
IF (R10<50) AND ($AA_IM[X]>=17.5)GOTOF ZIEL

IF NOT R10 GOTOB START
```

NOT se vztahuje pouze na jeden operand.

© Siemens AG 2004, Všechna práva vyhrazena

Bitové logické operátory

S proměnnými typu CHAR a INT mohou být prováděny i bitové logické operace. V případě potřeby proběhne konverze typu automaticky.

Význam logických bitových operátorů

B_AND	Bitové A
B_OR	Bitové NEBO
B_NOT	Bitová negace
B_XOR	Bitové exkluzivní NEBO

Operátor B_NOT se vztahuje pouze na jeden operand;
Tento je hned za operátorem.

Příklad programování

```
IF $MC_RESET_MODE_MASK B_AND ´B10000´GOTOF ACT_PLANE
```

© Siemens AG 2004, Všechna práva vyhrazena

1.8 Priorita operátorů

Priorita operátorů

Každému operátoru je přiřazena priorita. Při vyhodnocení výrazu jsou stále nejprve použity operátory vyšší s prioritou. U operátorů stejné úrovně proběhne vyhodnocení zleva doprava.

V aritmetických výrazech může být pomocí kulatých závorek stanoveno pořadí zpracování všech operací a tak je umožněno odchýlení se od normálních pravidel priority.

Pořadí operátorů
(od nejvyšší po nejnižší prioritu)

1.	`NOT, B_NOT`	Negace, bitová negace
2.	`*, /, DIV, MOD`	Násobení, dělení
3.	`+, -`	Sčítání, odčítání
4.	`B_AND`	Bitové A
5.	`B_XOR`	Bitové exkluzivní NEBO
6.	`B_OR`	Bitové NEBO
7.	`AND`	A
8.	`XOR`	exkluzivní NEBO
9.	`OR`	NEBO
10.	`<<`	Zřetězení řetězců, typ výsledku STRING
11.	`==, <>, >, <, >=, <=`	Porovnávací operátory

Příklad pro příkaz IF:
If (otto==10) and (anna==20) gotof end

Operátor zřetězení ":" pro framy se nesmí vyskytnout s ostatními operátory v jednom výrazu. Zařazení tohoto operátoru do stupnice priority proto není nutné.

© Siemens AG 2004, Všechna práva vyhrazena

1.9 Možné konverze typu



Konverze typu u přiřazení
Konstantní číselná hodnota, proměnná nebo výraz, který je přiřazen proměnné, se musí snášet s typem této proměnné, tak je při přiřazení typ automaticky změněn.

Možné konverze typu

z \ na	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
REAL	ano	ano*	ano[1)]	ano*	-	-	-
INT	ano	ano	ano[1)]	ano[2)]	-	-	-
BOOL	ano	ano	ano	ano	ano	-	-
CHAR	ano	ano	ano[1)]	ano	ano	-	-
STRING	-	-	ano[4)]	ano[3)]	ano	-	-
AXIS	-	-	-	-	-	ano	-
FRAME	-	-	-	-	-	-	ano

* Při změně typu z REAL na INT je u zlomkové hodnoty >=0.5 zaokrouhleno nahoru, jinak je zaokrouhlováno dolů (srovnej funkci ROUND)

1) Hodnotě <> 0 odpovídá TRUE, hodnotě == 0 odpovídá FALSE

2) Když hodnota leží v přípustné oblasti čísel

3) Když jen jeden znak

4) Délka řetězce 0 => FALSE, jinak TRUE



Je-li při konverzi některá hodnota větší než cílová oblast, bude to mít za následek chybové hlášení.



Další upozornění

Objeví-li se v jednom výrazu smíšené typy, tak je automaticky provedeno přizpůsobení typu.

© Siemens AG 2004, Všechna práva vyhrazena

1.10 Operace s řetězci

Přehled

Vedle v této kapitole popsaných klasických operací „přiřazení“ a „porovnání“ jsou k dispozici další možnosti pro manipulaci s řetězci.

Vysvětlení

Typ konverze do typu STRING:

`STRING_ERG = <<bel._Typ`[1)]	Typ výsledku: STRING
`STRING_ERG = AXSTRING (AXIS)`	Typ výsledku: STRING

Typ konverze z typu STRING:

`BOOL_ERG = ISNUMBER (STRING)`	Typ výsledku: BOOL
`REAL_ERG = NUMBER (STRING)`	Typ výsledku: REAL
`AXIS_ERG = AXNAME (STRING)`	Typ výsledku: AXIS

Zřetězení proměnných typu STRING:

`bel._Typ`[1)] `<< bel._Typ`[1)]	Typ výsledku: STRING

Změna psaní malých/velkých písmen:

`STRING_ERG = TOUPPER (STRING)`	Typ výsledku: STRING
`STRING_ERG = TOLOWER (STRING)`	Typ výsledku: STRING

Délka řetězce typu STRING:

`INT_ERG = STRLEN (STRING)`	Typ výsledku: INT

Hledat v řetězci znak/řetězec:

`INT_ERG = INDEX (STRING, CHAR)`	Typ výsledku: INT
`INT_ERG = RINDEX (STRING, CHAR)`	Typ výsledku: INT
`INT_ERG = MINDEX (STRING, STRING)`	Typ výsledku: INT
`INT_ERG = MATCH (STRING, STRING)`	Typ výsledku: INT

Výběr dílčího řetězce:

`STRING_ERG = SUBSTR (STRING, INT)`	Typ výsledku: INT
`STRING_ERG = SUBSTR (STRING, INT, INT)`	Typ výsledku: INT

Selekce jednotlivého znaku:

`CHAR_ERG = STRINGVAR [IDX]`	Typ výsledku: CHAR
`CHAR_ERG = STRINGFELD [IDX_FELD, IDX_CHAR]`	Typ výsledku: CHAR

„bel._Typ[1)]“ zastupuje proměnné typu INT, REAL, CHAR, STRING a BOOL.

© Siemens AG 2004, Všechna práva vyhrazena

Zvláštní význam znaku 0
Znak 0 je interně interpretován jako identifikace konce řetězce.
Je-li znak nahrazen znakem 0, je tím řetězec zkrácen.

Příklad:

```
DEF STRING[20] STRG = „Achse.steht“    ; výpis hlášení „Achse x steht“
STRG[6] = „X“

MSG[STRG]
STRG[6] = 0                            ; výpis hlášení „Achse“
MSG[STRG]
```

1.10.1 Konverze typů

Tak se dají využít proměnné rozdílných typů jako součást jednoho hlášení (MSG).

Konverze do typu STRING
Proběhne při použití operátoru << implicitně pro všechna data typu INT, REAL, CHAR a BOOL (viz „Zřetězení proměnných typu STRING“). Hodnota typu INT bude převedena do normálně čitelné formy. U hodnot typu REAL bude uvedeno až 10 míst za čárkou.

S proměnnými typu AXIS je možné převádět prostřednictvím funkce AXSTRING do typu STRING. Proměnné FRAME nemohou být konvertovány.

Příklad:

```
MSG („Position:“<<$AA_IM[X])
```

© Siemens AG 2004, Všechna práva vyhrazena

Konverze z typu STRING

Prostřednictvím funkce NUMBER se provádí konverze z typu STRING do typu REAL.
Dá-li ISNUMBER hodnotu FALSE, je při vyvolání NUMBER se stejným parametrem spuštěn alarm.
Pomocí funkce AXNAME je možné proměnnou typu STRING převádět na datový typ AXIS. Pokud řetězec není možné přiřadit žádnému z konfigurovaných identifikátorů os, aktivuje se alarm.

Syntaxe

`BOOL_ERG = ISNUMBER (STRING)`	Typ výsledku: BOOL
`REAL_ERG = NUMBER (STRING)`	Typ výsledku: REAL
`STRING_ERG = AXSTRING (AXIS)`	Typ výsledku: STRING
`AXIS_ERG = AXNAME = (STRING)`	Typ výsledku: AXIS

Sémantika:

ISNUMBER (STRING) dá TRUE, když řetězec představuje podle pravidel jazyka platné reálné číslo. Tak může být zkontrolováno, může-li být řetězec změněn na platné číslo.
NUMBER (STRING) předává řetězcem představované číslo jako reálnou hodnotu.
ASTRING (AXIS) předává udaný identifikátor osy jako řetězec.
AXNAME (STRING) změní udaný řetězec na identifikátor osy.

Příklady

`DEF BOOL BOOL_ERG`	
`DEF REAL REAL_ERG`	
`DEF AXIS AXIS_ERG`	
`DEF STRING[32] STRING_ERG`	
`BOOL_ERG = ISNUMBER („1234.9876Ex-7")`	;nyní: BOOL_ERG == TRUE
`BOOL_ERG = ISNUMBER („1234XYZ")`	;nyní: BOOL == FALSE
`REAL_ERG = NUMBER („1234.9876Ex-7")`	;nyní: REAL_ERG == 1234.9876Ex-7
`STRING_ERG = AXSTRING(X)`	;nyní: STRING_ERG == „X“
`AXIS_ERG = AXNAME („X")`	

© Siemens AG 2004, Všechna práva vyhrazena

1.10.2 Zřetězení proměnných typu STRING

Tato funkce poskytuje možnost skládat řetězec z jednotlivých součástí. Řetězení je realizováno prostřednictvím operátoru: <<. Tento operátor má pro všechny kombinace základních typů CHAR, BOOL, INT, REAL a STRING jako výsledný typ STRING. Případně nutná konverze je provedena dle stávajících pravidel. Typy FRAME a AXIS nemohou být použity s tímto operátorem.

Syntaxe:

`bel._Typ<<bel._Typ`	Typ výsledku: STRING

Sémantika:
Dané proměnné typu STRING (příp. implicitně změněný jiný typ) jsou spolu zřetězeny.

Tento operátor je k dispozici i samostatně jako tzv. „unární“ varianta. Tak je možné provést explicitní změnu do typu STRING (ne pro FRAME a AXIS).

Syntaxe:

`<< bel._Typ`	Typ výsledku: STRING

Sémantika:
Udaný typ je implicitně konvertován na typ STRING.

Například se tak dá složit hlášení nebo povel z textových seznamů a dají se doplnit parametry (asi jedno jméno prvku)

```
MSG (STRG_TAB[LOAD_IDX]<<BAUSTEIN_NAME)
```

Mezivýsledky při zřetězení typu STRING nesmí překročit maximální délku řetězce.

© Siemens AG 2004, Všechna práva vyhrazena

Příklady programování

```
DEF INT IDX = 2
DEF REAL VALUE = 9.654
DEF STRING[20]STRG = „INDEX:2“
IF STRG == „INDEX:“ <<IDX GOTOF NO_MSG
MSG („INDEX:“ <<IDX <<“/Wert:“          ; Vypíše se: „Index: 2/Wert: 9.654“
<<VALUE)
NO_MSG:
```

1.10.3 Změna psaní malých/velkých písmen

Tato funkce umožňuje změnit všechna písmena řetězce do jednotné formy.

Syntaxe:

STRING_ERG = TOUPPER	(STRING)	Typ výsledku: STRING
STRING_ERG = TOLOWER	(STRING)	Typ výsledku: STRING

Sémantika:
Všechna malá písmena budou změněna na velká resp. na malá.

Příklad:
Protože uživatelské vstupy mohou být realizovány také na MMC, lze jim dát standardní formu (velká nebo malá písmena):

```
DEF STRING [29] STRG
...
IF „LEARN.CNC“ == TOUPPER (STRG) GOTOF
LOAD_LEARN
```


© Siemens AG 2004, Všechna práva vyhrazena

1.10.4 Délka řetězce



Tato funkce umožňuje stanovení délky řetězce.

Syntaxe:

`INT_ERG = STRLEN (STRING)`	Typ výsledku: INT

Sémantika:
Bude zobrazen počet znaků, které – počítáno od začátku řetězce – nejsou znakem 0.

Příklad:
Toto umožňuje např. v souvislosti s dále popsaným přístupem k jednotlivému znaku, stanovení konce řetězce:

```
IF (STRLEN (BAUSTEIN_NAME) > 10) GOTOF FEHLER
```

1.10.5 Hledání znaku/řetězce v proměnné typu STRING



Tato funkce umožňuje hledat jednotlivé znaky popř. řetězec v jiné proměnné typu STRING. Výsledky této funkce udávají, na které pozici řetězce byl znak/řetězec v prohledávaném řetězci nalezen.

`INT_ERG = INDEX`	`(STRING, CHAR)`	Typ výsledku: INT
`INT_ERG = RINDEX`	`(STRING, CHAR)`	Typ výsledku: INT
`INT_ERG = MINDEX`	`(STRING, STRING)`	Typ výsledku: INT
`INT_ERG = MATCH`	`(STRING, STRING)`	Typ výsledku: INT

Sémantika:
Vyhledávací funkce: Zobrazují pozici v řetězci (první parametr), kde bylo hledání úspěšné. Jestliže znak/řetězec nemůže být nalezen, je zobrazena hodnota –1. První znak má přitom pozici 0.

© Siemens AG 2004, Všechna práva vyhrazena

INDEX	Hledá znak zadaný jako druhý parametr (odpředu) v prvním parametru.
RINDEX	Hledá znak zadaný jako druhý parametr (odzadu) v prvním parametru.
MINDEX	Odpovídá funkci INDEX, s výjimkou toho, že je předán seznam znaků (jako řetězec), ze kterých je předán index prvního nalezeného znaku.
MATCH	Hledá řetězec v řetězci.

Tak se dají řetězce rozložit podle určitých kritérií, na pozice se znakem mezery nebo s oddělovacím znakem („/“) v udání cesty.

Příklad programování

Příklad pro rozložení zadání v názvu cesty a prvku:

`DEF INT PFADIDX, PROGIDX`	
`DEF STRING[26] EINGABE`	
`DEF INT LISTIDX`	
`EINGABE = „/_N_MPF_DIR/_N_EXECUTE_MPF“`	
`LISTIDX = MINDEX (EINGABE, „M,N,O,P“) + 1`	Jako hodnota v LISTIDX je zobrazeno 3; jelikož „N“ je první znak v parametru EINGABE, z výběrového seznamu odpředu.
`PFADIDX = INDEX (EINGABE, „/“) +1`	;tak platí: PFADIDX = 1
`PROGIDX = RINDEX (EINGABE, „/“) +1`	;tak platí: POGIDX = 12
	; s pomocí v další části uvedené funkce SUBSTR se dá proměnná EINGABE rozložit na součásti „cesta“ a „prvek“:
`VARIABLE = SUBSTR (EINGABE, PFADIDX, PROGIDX-PFADIDX-1)`	Zobrazí pak “_N_MPF_DIR“
`VARIABLE = SUBSTR (EINGABE, PROGIDX)`	Zobrazí pak “_N_EXECUTE_MPF“

1.10.6 Výběr dílčího řetězce

Tato funkce umožňuje vytáhnout z určitého řetězce nějaký dílčí řetězec. Za tím účelem se zadá index prvního znaku a případně požadovaná délka. Není-li zadána informace o délce, je míněna zbývající část řetězce.

`STRING_ERG = SUBSTR (STRING,INT)`	Typ výsledku: INT
`STRING-ERG = SUBSTR (STRING, INT, INT)`	Typ výsledku: INT

© Siemens AG 2004, Všechna práva vyhrazena

Sémantika:
V prvním případě je dílčí řetězec převzat od pozice, která je určena prvním parametrem, až do konce řetězce.
Ve druhém případě je výsledný řetězec omezen na maximální délku, danou třetím parametrem.
Leží-li počáteční pozice za koncem řetězce, je převzat prázdný (“ “) řetězec.
Je-li počáteční pozice nebo délka negativní, aktivuje se alarm.

Příklad:

```
DEF STRING [29] ERG
ERG = SUBSTR („QUITTUG: 10 bis 99“,  ;tak platí: ERG == „10“
10, 2)
```

1.10.7 Selekce jednotlivého znaku

Tato funkce dovoluje selektovat jednotlivé znaky řetězce. Toto se týká jak čtecích, tak zápisových přístupových operací.

Syntaxe:

```
CHAR_ERG = STRINGVAR [IDX]                    Typ výsledku: CHAR
CHAR_ERG = STRINGFELD [IDX_FELD,              Typ výsledku: CHAR
IDX_CHAR]
```

Sémantika:
Bude načten/zapsán ten znak, který se nachází na udaném místě. Je-li zadání pozice negativní nebo větší než řetězec, je aktivován alarm.

Příklad hlášení:
Dosazení identifikátoru osy do předem vytvořeného řetězce.

```
DEF STRING [50] MELDUNG = „Achse n
hat Position erreicht“
MELDUNG[6] = „X“
MSG (MELDUNG)                    ; dá hlášení „Osa X dosáhla polohy“
```

© Siemens AG 2004, Všechna práva vyhrazena

Přístup k jednotlivým znakům je možný pouze u uživatelem definovaných proměnných (LUD-, GUD- a PUD-data).
Kromě toho je tento druh přístupu při vyvolání podprogramu možný jen pro parametry typu „Call-By-Value“.

Příklady:

Přístup k jednotlivým znakům systémových a strojních parametrů, ...:

```
DEF STRING [50] STRG
DEF CHAR QUITTUNG
…
STRG = $P_MMCA
QUITTUNG = STRG [0]                    ;vyhodnocení potvrzovací komponenty
```

Přístup k jednotlivým znakům u parametru Call-BY-Reference:

```
DEF STRING [50]STRG
DEF CHAR CHR1
EXTERN UP_CALL (VAR CHAR1)             ;Parametr Call-By-Reference!
…
UP_CALL (CHR1)                         ;Call-By-Rference
STRG [5] = CHR1
```


© Siemens AG 2004, Všechna práva vyhrazena

1.11 Příkaz CASE



Programování

CASE (výraz) OF konstanta1 GOTOF LABEL1 … DEFAULT GOTOF LABELn
CASE (výraz) OF konstanta1 GOTOF LABEL1 … DEFAULT GOTOB LABELn



Vysvětlení příkazů

CASE	Klíčové slovo pro příkaz skoku
GOTOB	Příkaz skoku s cílem skoku vzad (směr k začátku programu)
GOTOF	Příkaz skoku s cílem skoku vpřed (směr ke konci programu)
GOTO	Příkaz skoku s cílem skoku nejprve vpřed a vzápětí vzad (směr nejprve ke konci programu a potom k začátku programu)
GOTOC	Potlačení alarmu 14080 „Cíl skoku nenalezen“. Příkaz skoku s cílem skoku nejprve vpřed a vzápětí vzad (směr nejprve ke konci programu a potom k začátku programu)
LABEL	Cíl (značení uvnitř programu)
LABEL:	Po názvu cíle skoku následuje dvojtečka
Ausdruck	Aritmetický výraz
Konstante	Konstanta typu INT
DEFAULT	Cesta programu, nehodí-li se žádná z dříve uvedených konstant



Funkce

Příkaz CASE umožňuje v závislosti na hodnotě typu INT zpracovávat různé větve programu.



Postup

Podle toho, jaká hodnota je dosazena konstantě, která je vyhodnocována před příkazem CASE, provede program skok na místo, které bylo určeno příslušným cílem skoku.



Více informací k příkazům GOTO viz kapitola 10 „Početní parametry a programové skoky“.

© Siemens AG 2004, Všechna práva vyhrazena

Pro případy, kdy konstantě není dosazena žádná z předem určených hodnot, může být pokynem DEFAULT určen cíl skoku.

V případě, že příkaz DEFAULT není programován, je cílem skoku blok, navazující na příkaz CASE.



Příklad programování

Příklad 1

```
CASE (výraz)OF 1 GOTOF LABEL1 2 GOTOF LABEL2 … DEFAULT GOTOF LABELn
```

„1“ a „2“ jsou možné konstanty.
Je-li hodnota výrazu = 1 (INT-konstanta), skok na blok s návěštím LABEL1.
Je-li hodnota výrazu = 2 (INT-konstanta), skok na blok s návěštím LABEL2.

Jinak skok na bloku s návěštím LABELn.

Příklad 2

```
DEF INT VAR1 VAR2 VAR3
CASE(VAR1+VAR2-VAR3)OF 7 GOTOF MARKE1 9 GOTOF MARKE2 DEFAULT GOTOF
MARKE3
MARKE1: G0 X1 Y1
MARKE2: G0 X2 Y2
MARKE3: G0 X3 Y3
```


© Siemens AG 2004, Všechna práva vyhrazena

1.12 Řídící struktury



Vysvětlení

IF-ELSE-ENDIF	Volba mezi dvěma alternativami
LOOP-ENDLOOP	Nekonečná smyčka
FOR-ENDFOR	Čítací smyčka
WHILE-ENDWHILE	Smyčka s podmínkou na začátku smyčky
REPEAT-UNTIL	Smyčka s podmínkou na konci smyčky



Funkce

Řízení standardně zpracovává NC-bloky v programovaném pořadí.

Těmito příkazy se dají stanovit - vedle v této kapitole popsaných programových skoků – dodatečné alternativy a programové smyčky.

Příkazy umožňují strukturované programování a dobrou čitelnost programů.



Postup

1. IF-ELSE-ENDIF

Blok IF-ELSE-ENDIF slouží k výběru mezi 2 alternativami:

IF (výraz)
NC-bloky
ELSE
NC-bloky
ENDIF

Má-li výraz hodnotu TRUE, tzn. je splněna podmínka, bude proveden následující programový blok. Není-li podmínka splněna, proběhne větev ELSE.
Větev ELSE může odpadnout.

© Siemens AG 2004, Všechna práva vyhrazena

2. Nekonečná programová smyčka LOOP

Nekonečná smyčka nalezne uplatnění v nekonečných programech. Na konci smyčky dojde vždy ke skoku zpět na začátek smyčky.

LOOP
NC-bloky
ENDLOOP

3.Čítací smyčka FOR

Smyčka FOR je použita, má-li být pracovní činnost opakována s pevným počtem průběhů. Čítací proměnná je přitom počítána nahoru od počáteční hodnoty ke konečné. Počáteční hodnota musí být menší než konečná. Proměnná musí být typu INT.

FOR proměnná = počáteční hodnota **TO** konečná hodnota
NC-bloky
ENDFOR

4. programová smyčka s podmínkou na začátku smyčky WHILE

Smyčka WHILE proběhne, pokud je splněna podmínka.

WHILE výraz
NC-bloky
ENDWHILE

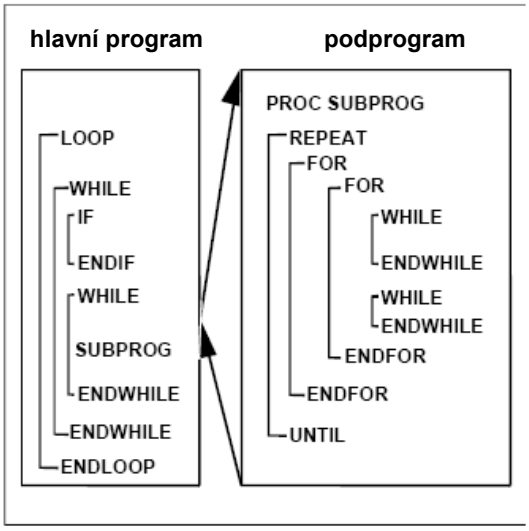
© Siemens AG 2004, Všechna práva vyhrazena

5. Programová smyčka s podmínkou na konci smyčky REPEAT

Smyčka REPEAT proběhne jednou a bude se tak dlouho opakovat, dokud nebude splněna podmínka.

REPEAT
NC-bloky
UNTIL (výraz)

Hloubka vnoření
Řídící struktury mají v programu lokální platnost. V rámci každé úrovně podprogramu je možná hloubka vnoření až 8 řídících struktur.



Doba zpracování
Ve standardně aktivním režimu interpretu může být použitím programových skoků dosaženo rychlejšího zpracování programu než s pomocí řídících struktur.
V předem přeložených cyklech není rozdíl mezi programovými skoky a řídícími strukturami.

© Siemens AG 2004, Všechna práva vyhrazena

Okrajové podmínky

Bloky s prvky řídících struktur nemohou být vyjmuty ze zpracování. Návěští nejsou v takových blocích přípustné.

Řídící struktury jsou zpracovány interpretativně.
Při rozpoznání konce smyčky je při zohlednění při tom nalezených řídících struktur hledán začátek smyčky.
Proto není v režimu interpretu kompletně kontrolována bloková struktura programu.
Doporučuje se nepoužívat programové skoky a řídící struktury současně.
Při předběžném zpracování cyklů může být prověřeno správné vnoření řídících struktur

⚠ Řídící struktury jsou možné jen v rámci příkazové části programu. Definice v záhlaví programu nemohou být prováděny podmíněně nebo opakovaně.

Klíčová slova pro řídící struktury nesmí být stejně jako cíle skoku překrývány makry. Pokud je definováno makro, kontrola se neuskutečňuje.

Příklad programování

1. Nekonečný program

```
%_N_LOOP_MPF
LOOP
     IF NOT SP_SEARCH                    ;žádné hledání bloku
          G01 G90 X0 Z10 F1000
          WHILE $AA_IM[X] <= 100
               G1 G91 X10 F500          ;vrtací vzor
               Z-5 F100
               Z5
          ENDWHILE
          Z10
     ELSE                                ;hledání bloku
MSG („Při vyhledávání není vrtáno“)
     ENDIF
```


© Siemens AG 2004, Všechna práva vyhrazena

```
      $A_OUT[1] = 1                          ;následující vrtaná deska
      G4 F2
ENDLOOP
M30
```

2. Zhotovení pevného počtu kusů

```
%_N_STUECKZAHL_MPF

DEF INT STUECKZAHL
FOR STUECKZAHL = 0 TO 100
      G01 …
ENDFOR
M30
```

1.13 Koordinace programu

Kanály

Kanál může svůj vlastní program zpracovat nezávisle na ostatních kanálech.
Tím jsou prostřednictvím programu ovladatelné mu občas přiřazené osy a vřetena.
Při uvedení do provozu mohou být pro ovládání zřízeny dva nebo více kanálů.

Koordinace programu

Jestliže se na zhotovení obrobku podílí více kanálů, může být potřebná synchronizace průběhů programu.
Pro tuto koordinaci programu existují zvláštní pokyny (příkazy). Jsou umístěny vždy v jednom samostatném bloku .

Upozornění
Od SW 5.3 je možná koordinace programu ve vlastním kanálu.

© Siemens AG 2004, Všechna práva vyhrazena

Příkazy pro koordinaci programu

• **Zápis s absolutním zápisem cesty**	
	Přitom je absolutní cesta tvořena podle následujících pravidel:
`INIT ) (n,"/_HUGO_DIR/_N_name_MPF")` nebo `INIT (n,"/_N_MPF_DIR/_N_name_MPF")`	• *Aktuální adresář/_N_name_MPF* „aktuální adresář“ znamená zvolený adresář obrobku nebo standardní adresář */_N_MPF_DIR*. • Zvolení určitého programu ke zpracování v určitém kanálu: N: číslo kanálu, hodnota podle konfigurace řízení • Kompletní název programu
Příklad: `INIT (2,"/_N_WKS_DIR/_ABRICHT_MPF")` `G01 F0.1` `START` `INIT (2,"/_N_WKS_DIR/_N_UNTER_1_SPF")`	**Do SW 3:** Mezi příkazem **INIT** (bez synchronizace) a **NC-Start** musí být nejméně jeden proveditelný blok. Při vyvolání podprogramu musí být doplněno „SPF“ do zápisu cesty.
• **Zápis s relativním zápisem cesty**	
Příklad: `INIT (2,"ABRICHT")` `INIT (3,"UNTER_1_SPF")`	Při relativním zápisu cesty platí tatáž pravidla jako pro vyvolání podprogramu. Při vyvolání podprogramu musí být „_SPF“doplněno do názvu programu.
`START` (n,n)	Spuštění zvolených programů v jiných kanálech. n,n: počítání čísel kanálů směrem nahoru: Hodnota v závislosti na konfigurací řízení.
`WAITM (Marker-Nr., n,n,…)`	Umístí značku „Marker-Nr.“ ve vlastním kanálu. Předchozí blok se ukončí přesným najetím. Počká se na značku se stejným „Marker-Nr.“ v zadaných kanálech „n“ (vlastní kanál nemusí být zadán). Značka bude po synchronizaci vymazána. Současně může být umístěno maximálně 10 značek v jednom kanálu.

© Siemens AG 2004, Všechna práva vyhrazena

`WAITMC (Marker-Nr., n, n, …)`	Umístí značku „Marker-Nr.“ ve vlastním kanálu. Přesné najetí nastane jen tehdy, pokud ostatní kanály ještě značky nedosáhly. Čekání na značku se stejným „Marker-Nr.“ v zadaných kanálech „n“ (vlastní kanál nemusí být zadán). Jakmile je v zadaných kanálech dosaženo značky „číslo značky“, pokračuje se ve zpracování bez ukončení přesného najetí.
`WAITE` (n,n)	Čekání na konec programu zadaných kanálů (vlastní kanál se nezadává).
`SETM (Marker-Nr., Marker-Nr., …)`	Umístění návěstidla „Marker-Nr.“ ve vlastním kanálu, bez vlivu na probíhající zpracování. SETM( ) si zachová platnost přes RESET a NC-START. SETM( ) může být programováno také ze synchronní akce.
`CLEARM (Marker-Nr., Marker-Nr., …)`	Vymazání značky „Marker-Nr.“ ve vlastním kanálu bez vlivu na probíhající zpracování. Všechny značky v kanálu mohou být vymazány prostřednictvím CLEARM( ). CLEARM(0) vymaže značku „0“. Příkaz CLEARM( ) si zachová platnost přes RESET a NC-START. CLEARM( ) může být programován také ze synchronní akce.



Upozornění
Všechny výše uvedené příkazy musí být v samostatných blocích.
Počet značek je závislý na instalované CPU.

Názvy kanálů
Názvy kanálů musí být přes proměnnou (viz kapitola 10 „Proměnné a početní parametry“) změněny na čísla.



Je třeba ochránit přiřazení čísel před náhodnou změnou.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:
Kanál s názvem „MASCHINE“ má obdržet číslo
kanálu 1,
kanál s názvem „LADER“ má obdržet číslo kanálu 2:

```
DEF INT MASCHINE=1, LADER=2
```

Proměnné obdrží stejné názvy jako kanály.
Potom má například příkaz start podobu:

```
START(MASCHINE)
```

Příklad koordinování programu

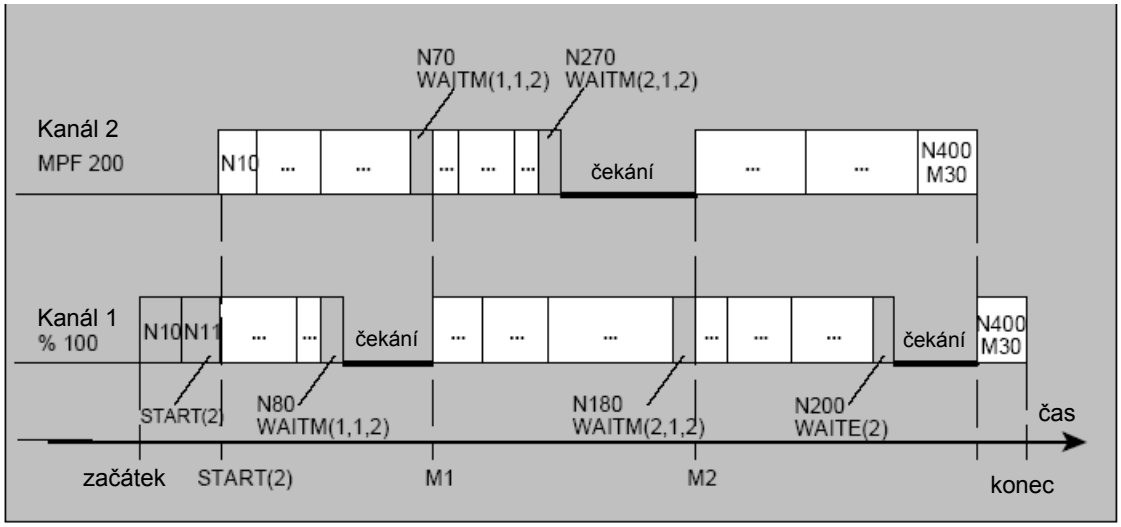
Kanál 1:

`%_N_MPF100_MPF`	
`N10 INIT(2, „MPF200“)`	
`N11 START(2)`	Zpracování v kanálu 2
.	
`N80 WAITM(1,1,2)`	Čekání na značce WAIT 1 v kanálu 1 a
.	v kanálu 2 další zpracování v kanálu 1
`N180 WAITM(2,1,2)`	Čekání na značce WAIT 2 v kanálu 1 a
.	v kanálu 2 další zpracování do kanálu 1
`N200 WAITE(2)`	Čekání na konec programu kanálu 2
`N201 M30`	Konec programu kanál1, celkový konec
...	

Kanál 2:

`%_N_MPF200_MPF`	
`;$PATH=/_N_MPF_DIR`	
	Zpracování v kanálu 2
`N70 WAITM(1,1,2)`	Čekání na značce WAIT 1 v kanálu 1 a
.	v kanálu 2 další zpracování do kanálu 1
`N270 WAITM(2, 1, 2)`	Čekání na značce WAIT 2 v kanálu 1 a
.	v kanálu 2 další zpracování do kanálu 2
`N400 M30`	Konec programu kanálu 2

© Siemens AG 2004, Všechna práva vyhrazena



Příklad pro program Obrobek

```
N10 INIT(2,"/_N_WKS_DIR/_N_WELLE1_WPD/_N_ABSPAN1_MPF")
```

Příklad pro příkaz INIT s relativním zápisem cesty

```
; v kanálu 1 je zvolen program /_N_MPFDIR/_N_MAIN_MPF
N10 INIT(2"MYPROG") ; program /_N_MPF_DIR/_N_MYPROG_MPF zvolit v
Kanálu 2.
```

Další upozornění

Pro výměnu dat mezi programy mohou být použity proměnné, kterými kanály disponují společně (NCK-specifické globální proměnné). Jinak je vytvoření programu prováděno pro každý kanál odděleně.

Do SW 3:
Po příkazu start nesmí být ihned dotazováno WAITE, jinak bude rozpoznán konec programu dříve, než byl program vůbec spuštěn.
Pomoc: Naprogramování doby prodlevy.
Příklad:

```
N30 START(2)
N31 G4 F0.01
N40 WAITE(2)
```

© Siemens AG 2004, Všechna práva vyhrazena

1.14 Rutina přerušení

Programování

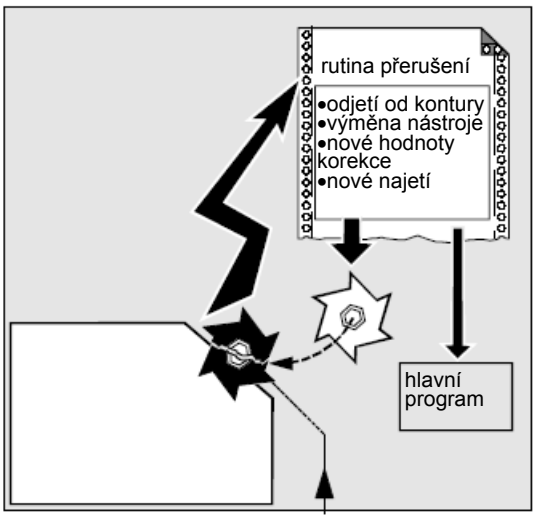
```
SETINT (3) PRIO=1 NAME
SETINT (3) PRIO=1 LIFTFAST
SETINT (3) PRIO=1 NAME LIFTFAST
G… X… Y… ALF=…
DISABLE (3)
ENABLE (3)
CLRINT (3)
```

Vysvětlení příkazů

SETINT (n)	Spuštění rutiny přerušení, pokud sepne vstup n, n(1…8) znamená číslo vstupu
PRIO=1	Stanovení priority 1 až 128 (1 má přednost)
LITFAST	Rychlé odjetí od kontury
NAME	Zde se nachází název podprogramu, který má být zpracován
ALF=…	Programovatelný směr posuvu (nachází se v pohybovém bloku)
DISABLE (n)	Vypnout rutinu přerušení číslo n
ENABLE (n)	Znovu zapnout rutinu přerušení n
CLRINT (n)	Vymazat přiřazení přerušení rutiny přerušení číslo n

Funkce

Příklad: Během opracování se zlomí nástroj. Tímto je aktivován signál, který zastaví průběh opracování a současně spustí podprogram - tzv. rutinu přerušení. V tomto podprogramu jsou všechny příkazy, které mají být v tomto případě provedeny.
Je-li podprogram dokončen (a tak je dosaženo stavu připravenosti k provozu), skočí řízení zpět do hlavního programu a pokračuje se v opracování - podle příkazu REPOS - na místě přerušení.



Více informací k příkazu REPOS naleznete v kapitole 9 „Chování při pohybu po dráze, Opětovné najetí na konturu“.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Rutinu přerušení sestavte jako podprogram
Rutina přerušení je při definování označena jako podprogram.

Příklad:
```
PROC ABHEB_Z
N10...
N50 M17
```

Název programu ABHEB_Z, potom následují NC-bloky, nakonec M 17 konec programu a návrat do hlavního programu.

Upozornění:
V rámci rutiny přerušení mohou být programovány pokyny SETINT a tímto aktivovány další rutiny přerušení. Jsou spouštěny příslušným vstupem.

Další informace pro sestavování podprogramů viz kapitola 2.

Uložení pozice přerušení, SAVE
Rutina přerušení může být při definování opatřena atributem SAVE.

Příklad:
```
PROC_ABHEB_Z SAVE
N10…
N50 M17
```

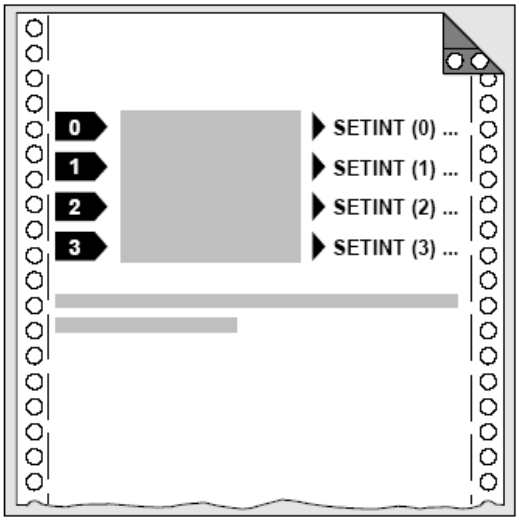
Prostřednictvím atributu SAVE budou modální funkce po ukončení rutiny přerušení nastaveny na hodnotu, kterou měly při spuštění rutiny přerušení. Dodatečně k nastavitelnému posunutí počátku (skupina modálních G-funkcí 8) bude obnoveno programovatelné posunutí počátku a základní posunutí. Vznikne-li změna skupiny G-funkcí 15 (typ posuvu), např. z G94 na G95, je obnovena také odpovídající hodnota F.

Díky tomu může později opracování pokračovat na místě přerušení.

© Siemens AG 2004, Všechna práva vyhrazena

Přiřazení a spuštění rutiny přerušení, SETINT

Řízení disponuje signály (vstup 1…8), které mohou spouštět přerušení probíhajícího programu a aktivovat odpovídající rutinu přerušení.



Přiřazení který vstup který program spustí, probíhá v programu opracování.

Příklad:

```
N10 SETINT(3) PRIO=1 ABHEB_Z
```

Při zapnutí vstupu 3 je ihned spuštěna rutina ABHEB_Z.

Spuštění více rutin, stanovení pořadí, PRIO=

Jestliže je ve Vašem NC-programu více příkazů SETINT a může se tedy vyskytnout více signálů současně, musíte rutiny přerušení označit pořadím, podle kterého mají být zpracovávány: PRIO 1 až 128, 1 má přednost.

Příklad:

```
N10 SETINT(3) PRIO=1 ABHEB_Z
N20 SETINT(2) PRIO=2 ABHEB_X
```

Rutiny budou provedeny ve stanoveném pořadí za sebou, pokud se vstupy aktivují současně. Nejprve SETINT (3) potom SETINT(2).

Pokud se v průběhu rutiny objeví nové signály, přeruší rutiny s vyšší prioritou právě zpracovávanou rutinu přerušení.

Zapnout/vypnout rutinu přerušení, DISABLE, ENABLE

Rutiny přerušení v NC-programu můžete vypnout prostřednictvím DISABLE (n) a opět zapnout prostřednictvím ENABLE(n) (n znamená číslo vstupu).

Přiřazení vstup/rutina zůstane při DISABLE zachováno a je prostřednictvím ENABLE znovu aktivováno.

© Siemens AG 2004, Všechna práva vyhrazena

Nové přiřazení rutin přerušení
Je-li obsazenému vstupu přiřazena nová rutina, stává se původní přiřazení automaticky neplatným.

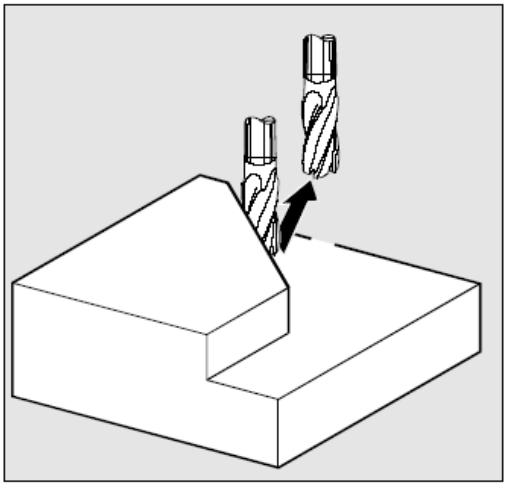
Příklad:
```
N20 SETINT(3) PRIO=2 ABHEB_Z
...
...
N120 SETINT(3) PRIO=1 ABHEB_X
```

Vymazání přiřazení,CLRINT
Prostřednictvím příkazu CLRINT(n) mohou být přiřazení vymazána.

Příklad:
```
N20 SETINT(3) PRIO=2 ABHEB_Z
N50 CLRINT(3)
```

Přiřazení mezi vstupem 3 a rutinou ABHEB_Z je vymazáno.

Rychlé odjetí od kontury, LIFTFAST
Prostřednictvím příkazu LIFTFAST při aktivování vstupu nástroj rychlým posuvem odjíždí od kontury obrobku.



Pokud pokyn SETINT obsahuje kromě LIFTFAST také rutinu přerušení, je rychlé zvednutí provedeno **před** rutinou přerušení.
Příklad:
```
N10 SETINT(2) PRIO=1 LIFTFAST
```
nebo
```
N30 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST
```

V obou případech bude při aktivování vstupu 2 s nejvyšší prioritou provedeno rychlé zvednutí.

- U N10 bude opracování zastaveno s alarmem 16010 (jelikož nebyl zadán žádný asynchronní podprogram, použije se příkaz ASUP).
- U N30 bude proveden ASUP „ABHEB_Z“.

© Siemens AG 2004, Všechna práva vyhrazena

Při určení směru odjíždění je prověřováno, zda je aktivní frame se zrcadlovým převracením.
V tomto případě je při směru odjíždění, které je vztaženo na směr tečny, zaměněno vlevo a vpravo. Složky směru ve směru nástroje nejsou zrcadlově převráceny. Toto chování je aktivováno prostřednictvím
MD $MC_LITFAST_WITH_MIRROR=TRUE



Průběh pohybu u rychlého zvednutí
Hodnota, o kterou geometrické osy při rychlém zvednutí od kontury odjedou, je nastavitelná pomocí strojního parametru.

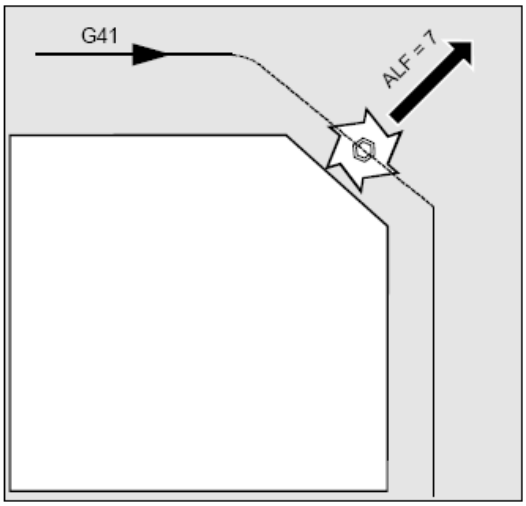
Rutina přerušení bez LIFTFAST
Zabrzdí se na dráze, a jakmile se pohyb na dráze zastaví, spustí se rutina přerušení.
Tato pozice je uložena v paměti jako pozice přerušení a při příkazu REPOS s RMI na konci rutiny přerušení se na znovu najíždí.

Rutina přerušení s LIFTFAST
Zabrzdí se na dráze a současně se provede superponovaný pohyb LITFAST. Pokud došlo k zastavení pohybu po dráze a pohybu LITFAST, je spuštěna rutina přerušení.
Jako pozice přerušení je uložena do paměti pozice na povrchu, při které byl spuštěn LIFTFAST-pohyb a tím byla opuštěna dráha.

Od SW 5.1 se chová rutina přerušení s příkazem LIFTFAST a ALF=0 shodně s rutinou přerušení bez příkazu LIFTFAST.

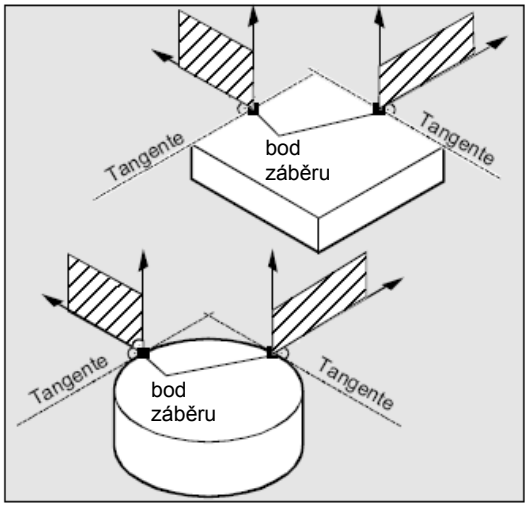
© Siemens AG 2004, Všechna práva vyhrazena

Programovatelný směr odjíždění, ALF =…
Směr, kterým se nástroj při rychlém zvednutí má pohybovat, zadáte v NC-programu.



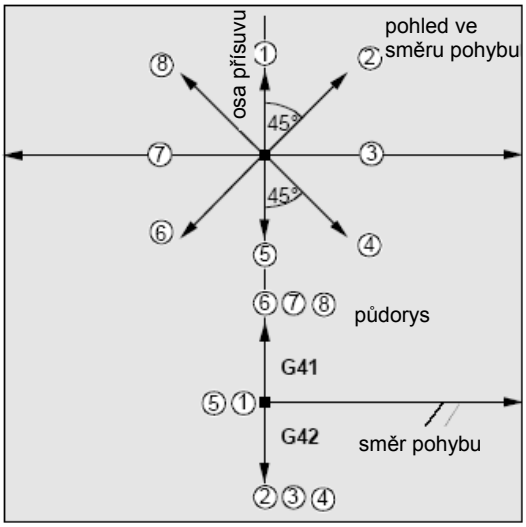
Vztažná rovina pro popis směrů odjíždění
V bodě záběru nástroje na naprogramované kontuře stanovena rovina, která slouží jako vztažná pro zadání pohybu pozvednutí odpovídajícím kódovým číslem.

Vztažná rovina je definována podélnou osou nástroje (směr přísuvu) a vektorem, který je kolmý k ní a k tečně v bodě záběru nástroje na kontuře.



Přehled číselných kódů se směry pohybu
Na vedlejším obrázku najdete číselné kódy se směry pohybu.

Prostřednictvím ALF=0 je funkce rychlého pozvednutí vypnuta.



Upozornění:
Při zapnuté korekci radiusu nástroje by
při G41 neměly být používány kódy 2,3,4 a
při G42 neměly být používány kódy 6,7,8.

V těchto případech by nástroj najel na konturu a došlo by ke kolizi s obrobkem.

Zpětný pohyb od SW 4.3
Směr zpětného pohybu od kontury je programován prostřednictvím G-kódu LFTXT popř. LFWP s proměnnou ALF.

- **LFTXT**
  Rovina zpětného pohybu je určena tečnou dráhy a směrem nástroje. Pomocí tohoto G-kódu (standardní nastavení) je programováno dosavadní chování při rychlém zvednutí.
- **LFWP**
  Rovina zpětného pohybu je aktivní pracovní rovina, která je zvolena G-kódy G17, G18 nebo G19. Směr zpětného pohybu je nezávislý na tečně dráhy. Takto je programovatelné rychlé zvednutí rovnoběžně s osou.
- **LFPOS (od SW 5)**
  Zpětný pohyb osy určené příkazem POLFMASK na absolutní pozici osy programovanou pomocí příkazu POLF. Viz také popis zpětného pohybu prováděného NC systémem, který je uveden v popisu funkcí M3.
  ALF nemá žádný vliv na směr zvednutí pro více os (od SW 6) ani na více os, jež jsou v lineární závislosti (od SW 7).

V rovině zpětného pohybu je jako doposud prostřednictvím ALF programován směr v malých krocích 45 stupňů. U LFTXT je pro ALF=1 stanoven zpětný pohyb ve směru nástroje.
U LFWP vyplyne směr v pracovní rovině z následujícího přiřazení:

- **G17:** rovina X/Y ALF=1 zpět ve směru X
  ALF=3 zpět ve směru Y
- **G18:** rovina Z/X ALF=1 zpět ve směru Z
  ALF=3 zpět ve směru X
- **G19:** rovina Y/Z ALF=1 zpět ve směru Y
  ALF=3 zpět ve směru Z

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

V tomto příkladu má být zlomený nástroj automaticky nahrazen náhradním nástrojem. Opracování pak pokračuje s novým nástrojem.

Hlavní program

`N10 SEINT(1) PRIO=1 W_WECHS ->` `-> LITFAST`	Pokud aktivuje vstup 1, je ihned nástroj odsunut od povrchu procesem rychlého zvednutí (číselný kód 7 pro korekci radiu nástroje G41). Potom proběhne rutina přerušení W-WECHS.
`N20 G0 Z100 G17 T1 ALF=7 D1`	
`N30 G0 X-5 Y-22 Z2 M3 S300`	
`N40 Z-7`	
`N50 G41 G1 X16 Y16 F200`	
`N60 Y35`	
`N70 X53 Y65`	
`N90 X71.5 Y16`	
`N100 X16`	
`N110 G40 G0 100 M30`	

Podprogram

`PROC_W_WECHS SAVE`	Podprogram s uložením aktuálního provozního stavu
`N10 G0 Z100 M5`	Pozice výměny nástroje, zastavení vřetena
`N20 T11 M6 D1 G41`	Vyměnit nástroj
`N30 REPOSL RMB M3`	Znovu najet na konturu a skok zpět do hlavního programu

-> je programováno v jednom bloku.

Pokud v podprogramu nenaprogramujete žádné příkazy REPOS, bude se najíždět na pozici na konci bloku, který následuje za blokem, v němž došlo k přerušení.

© Siemens AG 2004, Všechna práva vyhrazena

1.15 Výměna osy, výměna vřetena



Vysvětlení příkazů

`RELEASE (Achsname, Achsname, …)`	Uvolnění osy
`GET (Achsname, Achsname, …)`	Převzetí osy
`GETD (Achsname, Achsname, …)`	Přímé převzetí osy
`Achsname`	Přiřazení osy v systému: AX1, AX2, … nebo Údaj o názvech os stroje.
`RELEASE(S1)`	Uvolnění vřetena S1, S2, …
`GET(S2)`	Převzetí vřetena S1, S2, ---
`GETD(S3)`	Přímé převzetí vřetena S1, S2, …



Funkce

Jedna nebo více os resp. vřeten mohou být použity vždy pouze v jednom kanálu. Musí-li jedna osa pracovat střídavě ve dvou kanálech (např. měnič palet), tak musí být nejprve v aktuálním kanálu uvolněna a pak převzata do jiného kanálu: osa je mezi kanály vyměněna.



Další informace k funkci výměny osy nebo vřetene viz
/FB/, K5 BAGs, Kanály, výměna osy



Postup

Předpoklady pro výměnu osy

- Osa musí být definována prostřednictvím strojních parametrů ve všech kanálech, které chtějí osu používat.
- Prostřednictvím **pro osu** specifického strojního parametru musí být stanoveno, kterému kanálu má být osa přiřazena po POWER ON.

© Siemens AG 2004, Všechna práva vyhrazena

Uvolnění osy: RELEASE

Při uvolňování osy je třeba mít na zřeteli:

1. Osa se nesmí podílet na žádné transformaci.
2. V případě spojených os (tangenciální řízení) musí být uvolněny všechny osy spojené linkou.
3. Konkurující polohovací osa nemůže být v tomto stavu vyměněna.
4. U řídící osy gantry jsou změněny i všechny vlečné osy.
5. V případě spojených os (vlečení, spojení pomocí řídící hodnoty, elektronická převodovka) může být uvolněna jen řídící osa soustavy.

Převzetí osy: GET

Prostřednictvím tohoto příkazu bude provedena vlastní změna osy. Úplnou odpovědnost za tuto osu má kanál, ve kterém byl příkaz programován.

Účinek příkazu GET:

Výměna osy se synchronizací:
Osa musí být synchronizována vždy tehdy, když byla přechodně přiřazena jinému kanálu nebo PLC, a před příkazem GET nedošlo k synchronizaci prostřednictvím „WAITP“, G74 nebo vymazání zbytku dráhy.

- Dojde k zastavení předběžného zpracování (jako při STOPRE)
- **Opracování bude přerušeno tak dlouho, dokud nebude zcela provedena výměna.**

© Siemens AG 2004, Všechna práva vyhrazena

Výměna osy bez synchronizace
Nemusí-li být osa synchronizována, není příkazem GET vyvoláno zastavení předběžného zpracování.

```
Příklad:
N01 G0 X0
N02 RELEASE(AX5)
N03 G64 X10
N04 X20
N05 GET(AX5)

N06 G01 F5000
N07 X20

N08 X30
N09 …
```

Není-li nutná synchronizace, nebude toto proveditelný blok.
Neproveditelný blok.
Neproveditelný blok, jelikož X-pozice je stejná jako v N04.
První proveditelný blok po N05.

Automatické „GET“
Jestliže je osa principielně v kanálu k dispozici, ovšem v současnosti není k dispozici jako „kanálová osa“, je automaticky proveden příkaz GET. V případě, že osa(osy) je (jsou) již synchronizovány, zastavení předběžného zpracování není vyvoláno.



Prostřednictvím příkazu GET převzatá osa zůstane i po tlačítkovém nebo programovém příkazu RESET přiřazena tomuto kanálu. Při novém spuštění programu musí být přiřazení vyměněných os resp. vřeten programově technicky proběhnout v případě, že osa je potřebná v jejím základním kanálu.
Při příkazu POWER ON bude přiřazena kanálu, uloženému ve strojním parametru.

Přímo převzít osu: GETD
Prostřednictvím příkazu GETD (GET Directly) bude osa přímo převedena z jiného kanálu. To znamená, že k tomuto GET nemusí být programováno vhodné RELEASE v jiném kanálu. To ale také znamená, že nyní musí být vytvořena nová komunikace mezi kanály (například značky Wait).

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Ze 6 os jsou v kanálu jedna pro opracování použity: 1., 2., 3. a 4. osa.
5. a 6. osa jsou použity v kanálu 2 pro výměnu nástroje.

Je zapotřebí, aby osa 2 mohla být vyměněna mezi oběma kanály a po příkazu POWER ON musí být přiřazena kanálu 1.

Program „MAIN“ v kanálu 1

```
%_N_MAIN_MPF
```

`INIT (2,"TAUSCH2")`	Zvolení program TAUSCH2 v kanálu 2
`N… START (2)`	Spuštění programu v kanálu 2
`N… GET (AX2)`	Převzetí osy AX2
…	
…	
`N… RELEASE (AX2)`	Uvolnění osy AX2
`N… WAITM (1,1,2)`	Čekání na značce Wait v kanálu 1 a 2 pro synchronizaci v obou kanálech
`N…`	Další zpracování po výměně osy
`N… M30`	

Program „Tausch2“ V kanálu 2

```
%_N_TAUSCH2_MPF
```

`N… RELEASE (AX2)`	
`N160 WAIT (1,1,2)`	Čekání na značce Wait v kanálu 1 a 2 pro synchronizaci v obou kanálech
`N150 GET (AX2)`	Převzetí osy AX2
`N…`	Další zpracování po výměně osy
`N…M30`	

© Siemens AG 2004, Všechna práva vyhrazena

Nastavit chování při výměně proměnné osy

Okamžik odevzdání os se dá prostřednictvím MD 10772: AXCHANGE_MASK nastavit následovně:

- Automatická výměna os proběhne mezi kanály i tehdy, když byla osa prostřednictvím příkazu WAITP uvedena do neutrálního stavu (chování jako doposud).
- **Od SW 5.3** mohou být všechny osy umístěné do zásobníku os pomocí příkazů GET nebo GETD vyměněny teprve po rotaci zásobníku os.
- **Od SW 6.4** je po vloženém přechodovém bloku v hlavním průběhu prověřováno, zda je nebo není nutná reorganizace. Pouze v případě, že stavy os tohoto bloku nesouhlasí s aktuálními stavy os, je nutná reorganizace



Příklad programování

Aktivování výměny bez zastavení předběžného zpracování

```
N10 M4 S100
N011 G4 F2
N020 M5
N021 SPOS%0
N022 POS[B]=1
N023 WAITP[B]                    Osa B se stane neutrální osou
N030 X1 F10
N031 X100 F500
N032 X200
N040 M3 S500
N041 G4 F2
N050 M5
N099 M30
```

Jestliže se vřeteno (osa B) bezprostředně po bloku N023 jako **PLC-osa** přemístí např. na 180 stupňů a zpět na 1 stupeň a opět se stane neutrální osou, tak blok 024 nevyvolá zastavení předběžného zpracování a reorganizaci.

© Siemens AG 2004, Všechna práva vyhrazena

1.16 NEWCONF: Aktivování strojních parametrů (od SW 4.3)



Funkce

Příkazem jazyka NEWCONF jsou aktivovány všechny strojní parametry úrovně platnosti „NEW_CONFIG“. Funkce odpovídá stisknutí programového tlačítka „Set MD active“.
Při provedení funkce NEWCONF proběhne implicitní zastavení předběžného zpracování, tzn. pohyb po dráze je přerušen.



Vysvětlení

NEWCONF	Všechny strojní parametry úrovně platnosti „NEW_CONFIG“ budou aktivovány



Příklad programování

Opracování frézováním: opracování vrtané pozice různými technologiemi

```
N10 $MA_CONTOUR_TOL[AX]=1.0          ;změna strojního parametru
N20 NEWCONF                          ; aktivování strojního parametru
```

1.17 WRITE: Zápis do souboru (od SW 4.3)



Programování

```
WRITE (var int error, char[160] filename, char[200] string)
```

Příkaz WRITE přidá blok na konec uvedeného souboru.

© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení parametrů

`error`	Proměnná pro předání chybového hlášení 0 Žádná chyba 1 Nedovolená cesta 2 Nenalezená cesta 3 Nenalezený soubor 4 Chybný typ souboru 10 Soubor je plný 11 Soubor je používán 12 Žádné volné zdroje 13 Žádná přístupová oprávnění 20 Ostatní chyby
`filename`	Název souboru, do kterého má být řetězec zapsán. Obsahuje-li tento název souboru prázdné nebo řídící znaky (znaky s decimálním ASCII-kódem<=32), pak je příkaz WRITE přerušen chybovým hlášením 1 „nedovolená cesta“. Název souboru může být zadán s cestou a identifikací souboru. Názvy souborů musí být absolutní, tzn. že začínají „/“. Neobsahuje-li název souboru identifikaci domény (_N_), bude odpovídajícím způsobem doplněn. Není-li zadána identifikace (_MPF nebo _SPF), je automaticky doplněno _MPF. Dojde-li k zadání bez uvedení cesty, bude soubor uložen v aktuálním adresáři (= adresář zvoleného programu). Délka názvu souboru smí činit maximálně 32 bytů, délka názvu cesty maximálně 128 bytů. Příklad: PROTFILE _N_PROTFILE _N_PROTFILE_MPF /_N_MPF_DIR_/_N_PROTFILE_MPF/
`string`	Text , který má být napsán. Interně je ještě připojeno LF, tzn. že text bude o jeden znak delší.

Funkce

Prostřednictvím příkazu WRITE mohou být data (např. výsledky měření u měřicích cyklů) připojena na konec udaného souboru.
Prostřednictvím strojního parametru MD 11420 LEN_PROTOCOL_FILE je nastavena maximální délka protokolových souborů v kbytech. Tato délka platí pro všechny soubory, které jsou založeny prostřednictvím příkazu WRITE.

© Siemens AG 2004, Všechna práva vyhrazena

Dosáhne-li soubor uvedené délky, je vydáno chybové hlášení a řetězec není uložen. Pokud paměť dostačuje, může být založen nový soubor. Založené soubory mohou

- Být všemi uživateli čteny, měněny a mazány,
- Zapsány do výrobního programu, který je právě zpracováván.

Bloky jsou přidávány na konec souboru, tedy po M30.

Příklad programování

```
N10 DEF INT ERROR                                   ;
N20 WRITE(ERROR,"TEST1","PROTOKOLL VOM 7.2.97")     ; vypiš text PROTOKOL ze 7.7.97 do souboru TEST1
N30 IF ERROR                                        ;
N40 MSG („Fehler bei WRITE-Befehl:"
                                    <<ERROR)
M50 N0
N60 ENDIF
. . .
WRITE ERROR,                                        ; absolutní zadání cesty
„/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF",
„PROTOKOLL VOM 7.2.97")
```

Další upozornění

- Prostřednictvím příkazu WRITE popsaný soubor bude nově založen, pokud v NC neexistuje.
- Jestliže existuje soubor stejného jména na pevném disku, bude tento po uzavření souboru (v NC) přepsán.
  Pomoc: V systémové oblasti Service pomocí programového tlačítka „Properties“ změňte název v NC.

Výrobce stroje

Prostřednictvím příkazu WRITE mohou být bloky z výrobního programu uloženy v souboru. Velikost souboru pro protokolové soubory (kByte) je stanovena v MD.

© Siemens AG 2004, Všechna práva vyhrazena

1.18 DELETE: Vymazání souboru (od SW 4.3)



Programování

```
DELETE (var int error, char[160] filename)
```

Příkaz DELETE vymaže uvedený soubor.



Vysvětlení parametrů

`error`	Proměnná pro předání chybového hlášení 0 Žádná chyba 1 Nedovolená cesta 2 Nenalezená cesta 3 Nenalezený soubor 4 Chybný typ souboru 11 Soubor je používán 12 Žádné volné zdroje 20 Ostatní chyby
`filename`	Název souboru, který má být smazán. Název souboru může být zadán s cestou a identifikací souboru. Názvy cest musí být absolutní, tzn. že začínají „/“. Neobsahuje-li název souboru identifikaci domény (_N_), bude odpovídajícím způsobem doplněn. Identifikace souboru („_“ plus 3 znaky), např. _SPF) je volitelná. Není-li zadána identifikace (_MPF nebo _SPF), je automaticky doplněno _MPF. Dojde-li k zadání bez uvedení cesty, bude soubor uložen v aktuálním adresáři (= adresář zvoleného programu). Délka názvu souboru smí činit maximálně 32 bytů, délka názvu cesty maximálně 128 bytů.
	Příklad: PROTFILE _N_PROTFILE _N_PROTFILE_MPF /_N_MPF_DIR_/_N_PROTFILE_MPF/



Funkce

Prostřednictvím příkazu DELETE mohou být vymazány všechny soubory, je lhostejné zda vznikly prostřednictvím příkazu WRITE nebo ne. Také soubory, které byly vytvořeny při vyšší úrovni přístupových práv, mohou být příkazem DELETE smazány.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

N10 DEF INT ERROR	;
N15 STOPRE	;zastavení předběžného zpracování
N20 DELETE (ERROR, „/_N_SPF_DIR/_N_TEST1_SPF")	;vymaání soubor TEST1 ve větvi ; podprogramu
N30 IF ERROR	;
N40 MSG („Fehler bei DELTE-Befehl:" <<ERROR)	;
N50 M0	;
N60 ENDIF	;
. . .	

1.19 READ: Čtení řádků v souboru (od SW 5.2)

Programování

```
READ(var int error, string[160] file, int line, int number var
String[255] result[ ])
```

Příkaz READ přečte v zadaném souboru jeden nebo více řádků a uloží je v poli typu STRING. Každý přečtený řádek obsadí v tomto poli jeden prvek pole.

Vysvětlení parametrů

error	Proměnná pro předání chybového hlášení (Call-By-Reference-Parameter, typ INT) 0 Žádná chyba 1 Nedovolená cesta 2 Nenalezená cesta 3 Nenalezený soubor 4 Chybný typ souboru 13 Nedostatečná práva přístupu 21 Řádek neexistuje (parametr „line" nebo „number" větší než počet řádků souboru) 22 Délka pole výsledné proměnné „result" je příliš malá 23 Oblast řádků je příliš velká (parametr „number" je zvolen tak velký, že je čteno za konec souboru)
file	Název/cesta souboru, který má být čten (Call-By-Value-parametr typu STRING s maximální délkou 160 Byte). Soubor se musí nacházet v uživatelské paměti NCK (pasivní systém souborů). Názvu souboru může být předřazena identifikace domény _N_ . Neobsahuje-li název souboru identifikaci domény (_N_), bude odpovídajícím způsobem doplněna. Identifikace souboru („_" plus tři znaky např. _SPF je volitelná.

© Siemens AG 2004, Všechna práva vyhrazena

	Není-li zadána žádná identifikace, bude název souboru automaticky opatřen příponou _MPF. Není-li ve „file“ žádný údaj o cestě, bude soubor hledán v aktuálním adresáři (= adresář zvoleného programu). Zadané údaje cesty ve „file“ musí začínat znakem „/“ (absolutní zadání cesty).
`line`	Zadání pozice oblasti řádků, která má být čtena (Call-By-Value-parametr typu INT). 0 Bude čten počet řádků před koncem souboru zadaný parametrem „number“. 1 až n číslo prvního řádku, který má být čten
`number`	Počet řádků, které mají být čteny (Call-By-Value-parametr typu INT).
`Result`	Pole typu STRING, ve kterém je čtený text uložen (Call-By-Reference-parametr s délkou 255).

Funkce

Prostřednictvím příkazu READ můžete načíst jeden nebo více řádků ze souboru. Čtené řádky jsou uloženy v jednom prvku pole určitého pole. Informace je k dispozici jako STRING.

Další upozornění

- Binární soubory nemohou být načteny. Bude hlášena chyba error = 4: chybný typ souboru. Následující typy souborů není možné číst: _BIN,_EXE,_OBJ,_LIB,_TRC,_ACC, _CYC,_NCK.
- Aktuálně nastavený stupeň ochrany musí být stejný nebo vyšší než je oprávnění ke čtení souboru. Není-li tomu tak, bude přístup odmítnut hlášením error = 13
- Je-li v parametru „number“ zadáno méně řádků, než činí délka pole „result“, nebudou zbývající prvky pole změněny.
- Konec řádku prostřednictvím řídícího znaku „LF“ (Line FEED) nebo „CR LF“ (Carrige Return Line Feed) nebude v cílové proměnné „result“ uložen. Načtené řádky budou oříznuty, pokud je délka řádku větší než délka řetězce cílové proměnné „result“. Nenastane žádné hlášení chyby.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

```
N10 DEF INT ERROR                              ; chybová proměnná
N20 STRING[255] RESULT[5]                      ; výsledná proměnná
. . .
N30 READ(ERROR, „TESTFILE", 1, 5,              ; název souboru bez identifikace
                              RESULT)          domény a souboru
. . .
N30 READ(ERROR,"_N_TESTFILE_MPF",              ; název souboru s identifikací domény a
                       1,5, RESULT)            souboru
. . .
N30 READ(ERROR,"/_N_CST_DIR/N _TESTFILE        ; název souboru s identifikací domény a
                     _MPF", 1, 5 RESULT)       souboru a zadáním cesty
^. . .
N40 IF ERROR <>0                               ;vyhodnocení chyby
N50 MSG(„FEHLER"<<ERROR<<" BEI READ_BEFEHL")
N60 M0
N70 ENDIF
 . . .
```

1.20 ISFILE: Soubor k dispozici v uživatelské paměti NCK (od SW 5.2)

Programování

```
Result=isfile(string[60]file)
```

Prostřednictvím příkazu ISFILE je prověřováno, zda existuje soubor v uživatelské paměti NCK (pasivní systém souborů). Výsledkem je TRUE (soubor existuje) nebo False (soubor neexistuje.

Vysvětlení parametrů

`file`	Název resp. cesta souboru, který má být čten (Call-By-Value-parametr typu STRING s maximální délkou 160 Byte). Soubor se musí nacházet v uživatelské paměti NCK (pasivní systém souborů). Názvu souboru může být předřazena identifikace domény _N_. Jestliže identifikace domény schází, bude odpovídajícím způsobem doplněna. Identifikace souboru („_“ plus tři znaky např. _SPF) je nutná. Není-li zadána žádná identifikace , bude název souboru automaticky opatřen o příponu _MPF.

© Siemens AG 2004, Všechna práva vyhrazena

	Není-li v parametru „file“ žádný údaj o cestě, bude soubor hledán v aktuálním adresáři (= adresář zvoleného programu). Zadané údaje cesty v parametru „file“ musí začínat znakem „/“ (absolutní zadání cesty).
result	Proměnná pro podání výsledku typu BOOL (TRUE nebo FALSE)



Příklad programování

```
N10 DEF BOOL RESULT
N20 RSULT=ISFILE(„TESTFILE“)
N30 IF(RESULT==FALSE)
N40 MSG(„DATEI NICHT VORHANDEN“)
N50 M0
N60 ENDIF
. . .
nebo:
N30 IF(NOT ISFILE(„TESTFILE“))
N40 MSG(„DATEI NICHT VORHANDEN“)
N50 M0
N60 ENDIF
. . .
```


© Siemens AG 2004, Všechna práva vyhrazena

1.21 CHECKSUM: Vytvoření kontrolního součtu pole (od SW 5.2)

Programování

```
Error=CHECKSUM(var string[16] chksum,string[32]array, int first, int
last)
```

Funkce CHECKSUM vytvoří kontrolní součet pole.

Vysvětlení parametrů

`error`	Proměnná pro předání chybového hlášení 0 Žádná chyba 1 Symbol nenalezen 2 Žádné pole 3 Index 1 příliš velký 4 Index 2 příliš velký 5 Neplatný typ dat 10 Přetečení kontrolního součtu
`chksum`	Kontrolní součet pole jako řetězec (Call-By-Reference-parametr typu string se stanovenou délkou 16). Kontrolní součet je zobrazen jako řetězec znaků 16 Hex-čísel. Nejsou však připojovány žádné formátovací znaky. Příklad: „A6FC3404E534047C“
`array`	Jméno pole, jehož kontrolní součet má být vytvořen. (Call-By-Value-parametr typu string s maximální délkou 32). Přípustná pole: 1- nebo 2-rozměrná pole typů BOOL, CHAR, INT, REAL, STRING Pole strojních parametrů jsou nepřípustná.
`first`	Číslo sloupce počátečního sloupce (není nutné)
`last`	Číslo sloupce posledního sloupce (není nutné)

© Siemens AG 2004, Všechna práva vyhrazena



Funkce

Prostřednictvím funkce CHECKSUM vytvoříte kontrolní součet jednoho pole.
Použití při oddělování třísky:
Prověření, zda se vstupní kontura změnila.



Další upozornění

Parametry `first` a `last` jsou volitelné.
Není-li zadán index sloupce, je kontrolní součet vytvořen pro celé pole.

Výsledek kontrolního součtu je vždy jednoznačný.
Při změnách jednoho prvku pole je výsledkem i jiný výsledný řetězec.



Příklad programování

```
N10 DEF INT ERROR
N20 DEF STRING[16] MY_CHECKSUM
N30 DEF INT MY_VAR[4,4]
N40 MY_VAR=. . .
N50 ERROR=CHECKSUM
    (CHECKSUM;"MY_VAR", 0, 2)
. . .

Dá v MY_CHECKSUM hodnotu
„A6FC3404E534047C"
```


© Siemens AG 2004, Všechna práva vyhrazena

Technika podprogramů, technika maker

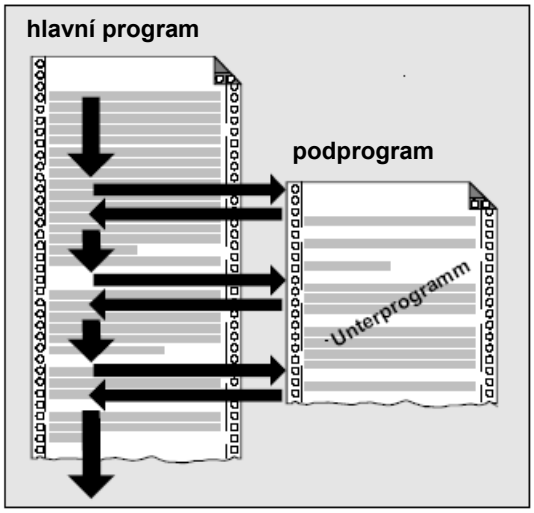
2.1 Použití podprogramů ... 2-96
2.2 Podprogram s mechanismem SAVE ... 2-97
2.3 Podprogramy s předáváním parametrů ... 2-98
2.4 Vyvolání podprogramů: L resp. EXTERN ... 2-103
2.5 Návrat z podprogramu s možností zadání parametrů (od SW 6.4) ... 2-107
2.6 Podprogram s opakováním části programu: P ... 2-111
2.7 Modální podprogram: MCALL ... 2-111
2.8 Nepřímé volání podprogramu: CALL ... 2-112
2.9 Opakování části programu s nepřímým programováním (od SW 6.4) ... 2-114
2.10 Nepřímé volání programu programovaného v jazyce ISO: ISOCALL ... 2-115
2.11 Volání podprogramu se zadáním cesty a s parametry: PCALL ... 2-116
2.12 Rozšíření cesty při vyvolání podprogramu pomocí CALLPATH (od SW 6.4) ... 2-117
2.13 Potlačit zobrazení aktuální sady: DISPLOF ... 2-119
2.14 Potlačení zpracování blok po bloku: SBLOF, SBLON (od SW 4.3) ... 2-120
2.15 Zpracování externího podprogramu: EXTCALL (od SW 4.2) ... 2-126
2.16 Vyvolání podprogramu funkcí M-/T- ... 2-129
2.17 Cykly: Zadávání parametrů uživatelských cyklů ... 2-131
2.18 Technika maker: DEFINE…AS ... 2-135

© Siemens AG 2004, Všechna práva vyhrazena

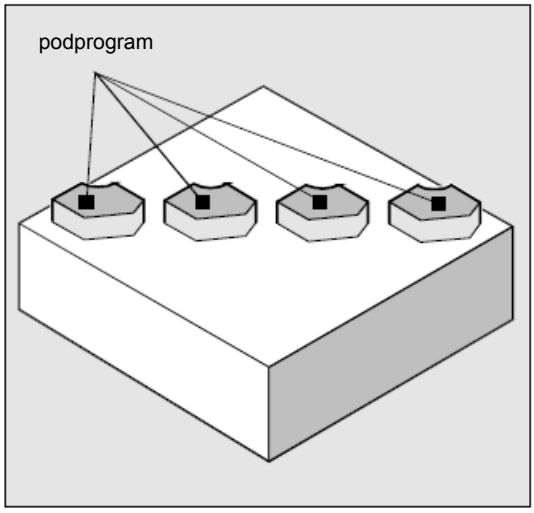
2.1 Použití podprogramů

Co je to podprogram?
V principu je podprogram sestaven stejně jako výrobní program. Skládá se z NC-bloků s příkazy pohybu a spínání.

Principielně není mezi hlavním programem a podprogramem žádný rozdíl. Podprogram obsahuje buď pracovní postupy nebo pracovní úseky, které mají být zpracovány vícekrát.



Použití podprogramů
Procesy opracování, které se opakují, se programují jen jednou v podprogramu. Například různé tvary kontury, které se vyskytují opakovaně nebo také cykly opracování.



Tento podprogram může být v libovolném hlavním programu vyvolán a zpracován.

Stavba podprogramu
Stavba podprogramu je identická se stavbou hlavního programu.

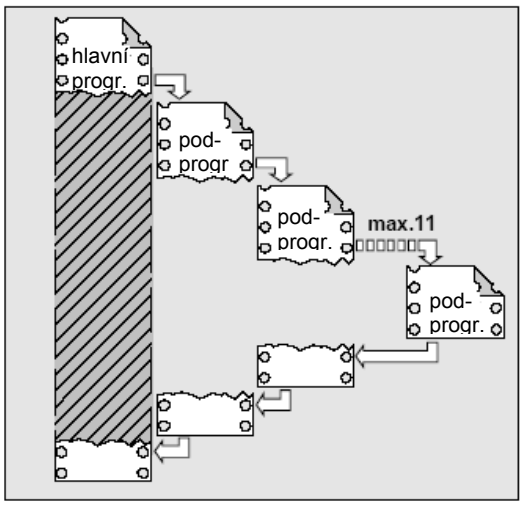
Navíc může být v podprogramu naprogramována hlavička programu s definováním parametrů.

© Siemens AG 2004, Všechna práva vyhrazena

Hloubka vnoření

Vnoření podprogramů

V podprogramu může být umístěno opět vyvolání podprogramu. V tomto podprogramu je umístěno opět vyvolání podprogramu atd.
Maximální počet programových úrovní resp. hloubka vnoření činí 12.



Toto znamená:
Z jednoho hlavního programu může vycházet 11 vnořených vyvolání podprogramu.

Omezení

Také v rutinách přerušení mohou být vyvolány programy. Pro práci s rutinami přerušení by jste měli nechat 4 úrovně volné resp. vnořit jen 7 vyvolání podprogramu.

Pro obráběcí a měřicí cykly Siemens jsou třeba tři úrovně. Má-li být cyklus vyvolán podprogramem, může to proběhnout maximálně v úrovni 5 (pokud jsou 4 úrovně rezervovány pro rutiny přerušení).

2.2 Podprogram s mechanismem SAVE

Funkce

Pro tento účel zadejte u pokynů k definování prostřednictvím PROC navíc příkaz SAVE.

Díky atributu SAVE budou modální G-funkce po ukončení podprogramu nastaveny na hodnotu, kterou měly při spuštění podprogramu. Dojde-li tím ke změně skupiny G-funkcí 8 (nastavitelné posunutí počátku), skupiny G-funkcí 52 (otáčení framu otočného obrobku) nebo skupiny G-funkcí 53 (otáčení framu ve směru nástroje), tak budou příslušné framy obnoveny.

- Aktivní základní frame nebude při skoku zpět podprogramu změněn
- Programovatelné posunutí počátku bude obnoveno

© Siemens AG 2004, Všechna práva vyhrazena

Od SW 6.1 může být chování nastavitelného posunutí počátku a základního framu změněno prostřednictvím strojního parametru MD 10617: FRAME_SAVE_MASK.



Další informace k tomuto procesu viz /FB/ K1, Všeobecné strojní parametry

Příklad:

Definice podprogramu

```
PROC KONTUR (REAL WERT1) SAVE
N10 G91 …
N100 M17
```

Hlavní program

```
%123
N10 GO X… Y… G90
N20…
N50 KONTUR (12.4)
N60 X… Y…
```

V podprogramu KONTUR aktivuje G91 zadávání řetězových kót. Po skoku zpět do hlavního programu se opět aktivuje zadávání absolutních souřadnic, neboť modální funkce hlavního programu byly uloženy prostřednictvím příkazu SAVE.

2.3 Podprogramy s předáním parametru



Začátek programu, PROC

Podprogram, který má při průběhu vyvolávajícího programu převzít parametry, je označen klíčovým slovem PROC.

Konec podprogramu M17, RET

Prostřednictvím příkazu M17 se značí konec podprogramu se současným pokynem ke skoku do vyvolávajícího hlavního programu.
Alternativně k M17: Klíčové slovo RET znamená konec podprogramu bez přerušení procesu řízení pohybu po dráze a bez výstupu funkce na PLC.

Proces řízení dráhy bez přerušení

Předpokladem pro to, aby proces řízení pohybu po dráze nebyl přerušen, je:
Podprogram nesmí obsahovat **žádný** atribut SAVE. Další odkazy na mechanismus SAVE viz kapitola 2.2.

© Siemens AG 2004, Všechna práva vyhrazena



RET musí být programováno v samostatném NC-bloku.

Příklad:

```
PROC KONTUR
N10…
…
N100 M17
```

Předání parametrů mezi hlavním programem a podprogramem

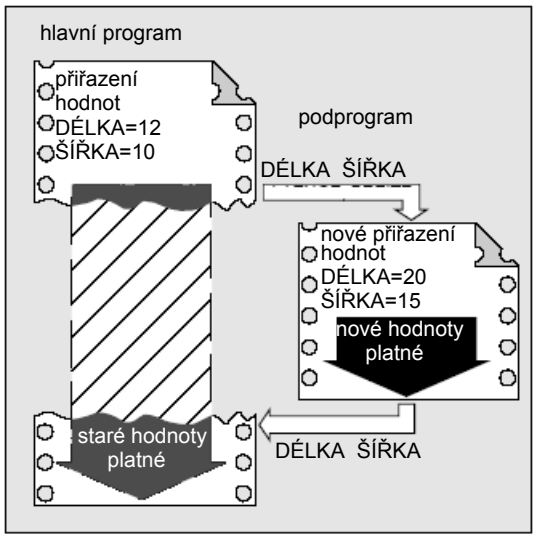
Pokud pracujete v hlavním programu s parametry, můžete použít odpovídající vypočtené nebo přiřazené hodnoty také v podprogramu.
To je realizováno tak, že hodnoty **aktuálních parametrů** hlavního programu jsou při vyvolání podprogramu předány **formálním parametrům** podprogramu a potom jsou při průběhu podprogramu zpracovány.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:

```
N10 DEF REAL LAENGE,BREITE
N20 LAENGE12 BREITE=10
N30 RAHMEN(LAENGE,BREITE)
```

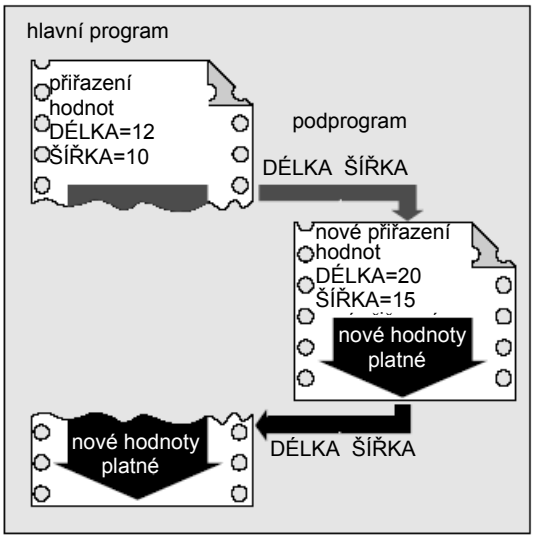
V hlavním programu v N20 přiřazené hodnoty jsou při vyvolání podprogramu předány do N30.
Předání parametrů probíhá v udaném pořadí.
Jména parametrů nemusí být v hlavním programu a v podprogramu identická.



2 Možnosti pro předání parametrů

Jsou předávány jen hodnoty (Call-by-value)
Jestliže jsou předané parametry v průběhu podprogramu změněny, nemá to vliv na hlavní program. Zde zůstávají parametry nezměněny (viz obrázek).

Předání parametrů se změnou dat (Call-by-reference)
Každá změna parametrů v podprogramu má současně za následek změnu parametru v hlavním programu (viz obrázek).



© Siemens AG 2004, Všechna práva vyhrazena

Programování

Pro předání parametrů musí být relevantní parametry na začátku podprogramu uvedeny včetně typu a názvu.

Předání parametrů Call-by-value

```
PROC PROGRAMMNAME(VARIABLENTYP1 VARIABLE1,VARIABLENTYP2 VARIABLE2,…)
```

Příklad:

```
PROC KONTUR(REAL LAENGE, REAL BREITE)
```

Předání parametrů Call-by-reference,
Označení klíčovým slovem VAR

```
PROC PROGRAMMNAME(VAR VARIABLENTYP1 VARIABLE1,VAR VARIABLENTYP2 ...,)
```

Příklad:

```
PROC KONTUR(VAR REAL LAENGE, VAR REAL BREITE)
```

Předání parametrů Call-by-reference,
Označení klíčovým slovem VAR

```
PROC PROGRAMMNAME(VAR VARIABLENTYP1 FELDNAME1[velikost pole],
VAR VARIABLENTYP2 FELDNAME2[velikost pole], VAR VARIABLENTYP3
FELDNAME3[velikost pole 1, velikost pole 2], VAR VARIABLENTYP4
FELDNAME4[ ],VAR VARIABLENTYP5 FELDNAME5 [,velikost pole])
```

Příklad:

```
PROC PALETTE(VAR INT FELD[,10])
```

Další upozornění

Příkaz definice prostřednictvím PROC musí být programován v samostatném NC-bloku. Pro předání parametrů může být stanoveno maximálně 127 parametrů.

© Siemens AG 2004, Všechna práva vyhrazena

Definování pole

Pro definování formálních parametrů platí:
U dvojrozměrných polí nemusí být počet polí 1. dimenze udán, čárka však musí být napsána.
Příklad:

```
VAR REAL FELD[,5]
```

S neurčenými délkami mohou podprogramy zpracovávat pole variabilní délky. Při definování proměnných musí být však stanoveno, kolik prvků má být vybráno.

Vysvětlení k definování polí najdete v Příručce programování „Pro pokročilé“.

Příklad programování

Programování s variabilními délkami polí	
`%_N_BOHRPLATTE_MPF`	Hlavní program
`DEF REAL TABELLE[100,2]`	Definování tabulky pozic
`EXTERN BOHRBILD (VAR REAL[,2],INT)`	
`TABELLE[0,0]=-17.5`	Stanovení pozic
…	
`TABELLE[99,1]=45`	
`BOHRBILD(TABELLE,100)`	Vyvolání podprogramu
`M30`	

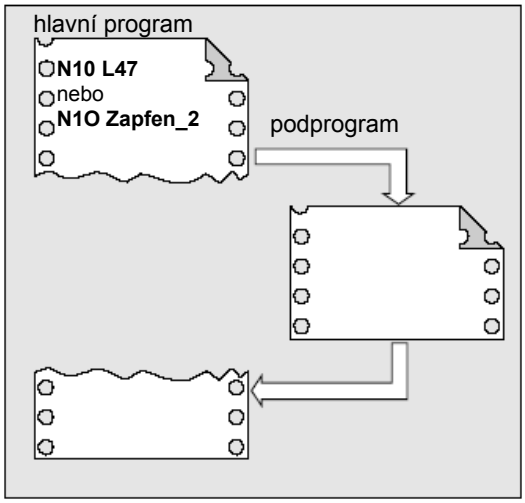
Vytvoření vrtacího vzoru na základě předané tabulky pozic variabilní délky	
`%_N_BOHRBILD_SPF`	Podprogram
`PROC BOHRBILD(VAR REAL FELD[,2],-> -> INT ANZAHL)`	Předání parametrů
`DEF INT ZAEHLER`	
`STEP: G1 X=FELD[ZAEHLER,0]-> -> Y=FELD[ZAEHLER,1] F100`	Postup zpracování
`Z=IC(-5)`	
`Z=IC(5)`	
`ZAEHLER=ZAEHLER+1`	
`IF ZAEHLER<ANZAHL GOTOB STEP`	
`RET`	Konec podprogramu

© Siemens AG 2004, Všechna práva vyhrazena

2.4 Vyvolání podprogramů: L, resp. EXTERN

Vyvolání podprogramu bez předání parametrů

V hlavním programu vyvoláte podprogram buď prostřednictvím adresy L a čísla podprogramu nebo zadáním názvu programu.

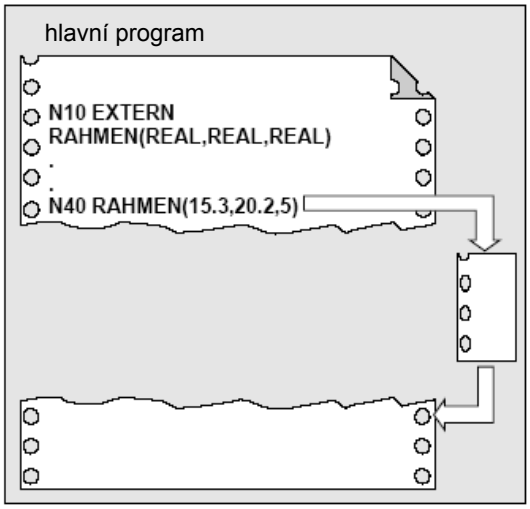


Příklad:

```
N10 L47 nebo
N10 ZAPFEN_2
```

Podprogram s předáním parametrů, Vysvětlení s EXTERN

Podprogramy s předáním parametrů musí být před vyvoláním v hlavním programu být uvedeny s příkazem EXTERN, např. na začátku programu. Zadán je název podprogramu a typy proměnných v pořadí předání.



Příkaz EXTERN musí být uveden jen tehdy, když se podprogram nachází v adresáři obrobků nebo v globálním adresáři podprogramů.
Cykly nemusí být označeny jako EXTERN.

```
Příkaz EXTERN
EXTERN NAME(TYP1, TYP2, TYP3, …) nebo
EXTERN NAME(VAR TYP1, VAR TYP2, …)
```

Příklad:

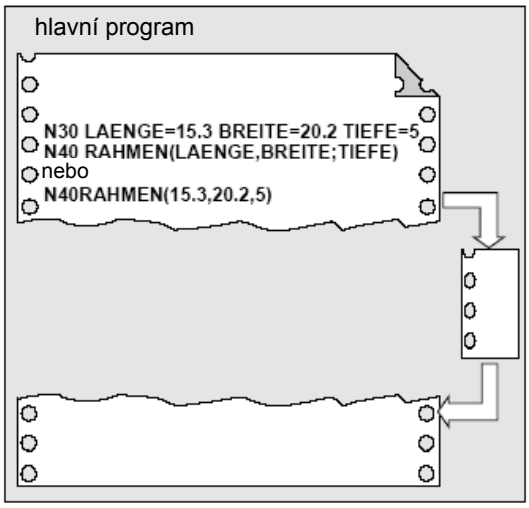
```
N10 EXTERN RAHMEN(REAL, REAL, REAL)
…
N40 RAHMEN(15.3,20.2,5)
```

N10 je deklarace podprogramu, N40 vyvolání podprogramu s předáním parametrů.

© Siemens AG 2004, Všechna práva vyhrazena

Vyvolání podprogramu s předáním parametrů

V hlavním programu vyvoláte podprogram zadáním jeho názvu a předáním parametrů. U předání parametrů můžete proměnné nebo hodnoty předat přímo (nikoli u VAR-parametrů).



Příklad:

```
N10 DEF REAL LAENGE,BREITE,TIEFE
N20 …
N30 LAENGE=15.3 BREITE=20.2 TIEFE=5
N40 RAHMEN(LAENGE,BREITE,TIEFE)
```

nebo

```
N40 RAHMEN(15.3,20.2,5)
```

Definování podprogramu odpovídá vyvolání podprogramu

Jak typy proměnných, tak i pořadí předání musí souhlasit s definicemi, které byly uvedeny v názvu programu pod PROC. Názvy parametrů mohou být v hlavním programu a v podprogramu rozdílné.

Příklad:

Definice v podprogramu:

```
PROC RAHMEN(REAL LAENGE, REAL BREITE,
REAL TIEFE)
```

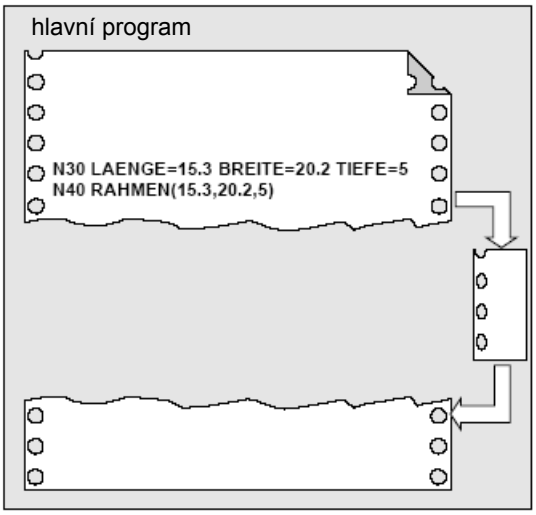
Vyvolání v hlavním programu:

```
N30 RAHMEN(LAENGE, BREITE, TIEFE)
```


© Siemens AG 2004, Všechna práva vyhrazena

Neúplné předání parametrů

Při vyvolání podprogramu mohou být předepsané hodnoty nebo parametry i vynechány. V tomto případě bude odpovídajícímu parametru v podprogramu dosazena **nula.**



Pro zachování posloupnosti musí být napsána čárka. Nacházejí-li se parametry na konci posloupnosti, může být vynechána i čárka.

Zpět k poslednímu příkladu:

```
N40 RAHMEN(15.3, ,5)
```

Střední hodnota 20.2 zde byla vynechána.

Upozornění

Aktuální parametry typu AXIS nesmí být vynechány.
VAR-parametry musí být předány úplné.

Od SW 4.4:

Při neúplném předání parametrů může být prostřednictvím systémové proměnné `$P_SUBPAR[i]` rozpoznáno, zda předávaný parametr pro podprogram byl opravdu naprogramován nebo ne.
Systémová proměnná obsahuje jako argument (`i`) číslo předávaného parametru.
Systémová proměnná `$P_SUBPAR[i]` poskytuje hodnotu:

- TRUE, když byl předávaný parametr programován
- FALSE, když jako předávaný parametr nebyla dosazena žádná hodnota.

Bylo-li zadáno nepřípustné číslo parametru, bude zpracování výrobního programu přerušeno s alarmem.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:

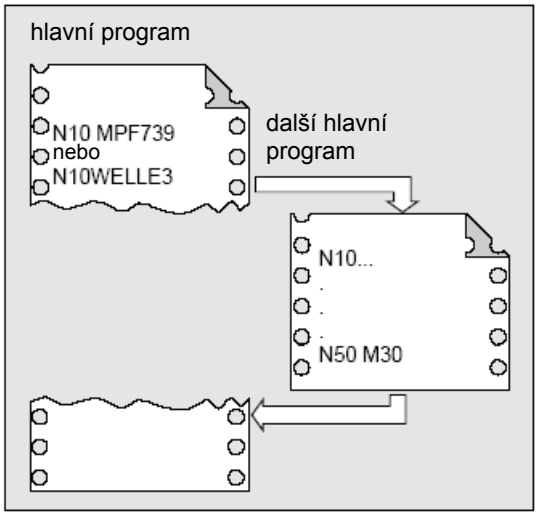
Podprogram

```
PROC SUB1 (INT VAR1, DOUBLE VAR2)

IF $P_SUBPAR[1]==TRUE
  ;Parametr VAR1 byl programován ve
  ;vyvolání podprogramu
ELSE
  ;Parametr VAR1 nebyl programován
  ;ve vyvolání podprogramu a
  ;systémem byla dosazena
  ;Default hodnota 0
ENDIF
IF $P_SUBPAR[2]==TRUE
  ;Parametr VAR2 byl programován ve
  ;vyvolání podprogramu
ELSE
  ;Parametr VAR2 nebyl programován
  ;ve vyvolání podprogramu a
  ;systémem byla dosazena
  ;Default hodnota 0.0
ENDIF
;Parametr 3 není definován
IF $P_SUBPAR[3]==TRUE -> Alarm
17020
M17
```

Vyvolání hlavního programu jako podprogramu

Také hlavní program může být vyvolán jako podprogram. V hlavním programu nastavený konec programu M2 nebo M30 bude v tomto případě (konec programu se skokem zpět do volávajícího programu) vyhodnocen jako M17.



Vyvolání naprogramujete zadáním jména programu.

Příklad:

`N10 MPF739` nebo

`N10 WELLE3`

Odpovídajícím způsobem může být podprogram spuštěn také jako hlavní program.

© Siemens AG 2004, Všechna práva vyhrazena

2.5 Návrat z podprogramu se zadáním parametrů (od SW 6.4)

Programování

Návrat z podprogramu se zadáním příslušných parametrů

```
RET (<Číslo bloku/Návěští>, <Blok po bloku s číslem bloku/Návěští>,
<Počet úrovní pro návrat>), <Návrat na začátek programu>)
```

`RET (<Číslo bloku/Návěští>, < >, < >)`	
`RET (, , <Počet úrovní pro návrat>, <Návrat na počátek programu>)`	Návratový skok z podprogramu o více úrovní (skok o zadaný počet úrovní podprogramu zpět).

Vysvětlení

`<Číslo bloku/Návěští>`	1. parametr: Číslo bloku nebo návěští jako STRING (konstanta nebo proměnná) bloku, ve kterém má zpracování programu pokračovat. Zpracování programu ve volajícím programu bude pokračovat blokem s „číslo bloku/návěští“.
`<Blok po bloku s číslem bloku/Návěští>,`	2. parametr typu INTEGER Je-li **hodnota větší než 0**, bude se pokračovat blokem následujícím po bloku s uvedeným „číslem bloku/návěštím“, je-li **hodnota rovna 0**, návrat z podprogramu se uskuteční do bloku s <číslem bloku/návěštím>.
`<Počet úrovní pro návrat>,`	3. parametr typu INTEGER s přípustnými **hodnotami 1 až 11**. Hodnota = 1: Program bude pokračovat v aktuální programové úrovni –1 (jako RET bez parametru). Hodnota = 2: program bude pokračovat v aktuální programové úrovni –2, přitom bude jedna úroveň přeskočena atd.
`<Návrat na začátek programu>,`	4. parametr typu BOOL **hodnota 1 nebo 0.** Hodnota = 1 Při návratu do hlavního programu, ve kterém je aktivní režim dialektu ISO, zpracování bude pokračovat od začátku programu.

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Návrat z podprogramu se obecně provádí pomocí příkazu konce podprogramu RET nebo M17 do volajícího programu a zpracování výrobního programu pokračuje řádkem následující za voláním podprogramu. Jsou ovšem případy použití, kde je žádoucí, aby zpracování programu pokračovalo na jiném místě:

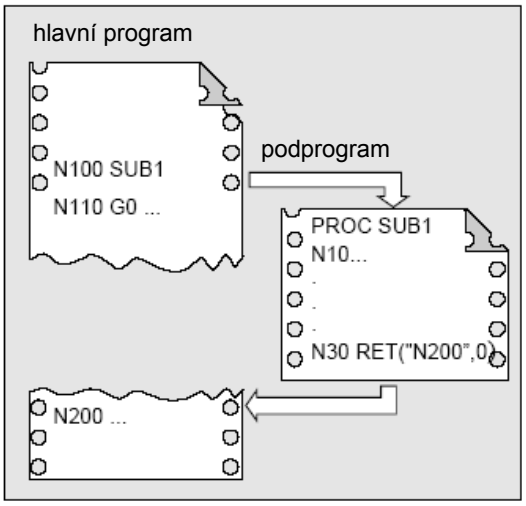
- Pokračování zpracování po vyvolání cyklů oddělování třísky v dialektu ISO, po popsání kontury.
- Návrat do hlavního programu z libovolné úrovně podprogramu (také po ASUP) v případě zpracování chyby.
- Návrat přes více programových úrovní pro speciální použití v kompilačních-cyklech a v dialektu ISO.

Prostřednictvím příkazu se zadáním parametrů RET mohou být tyto požadavky být splněny čtyřmi parametry:

1. <číslo bloku/návěští>
2. <blok za blokem s číslem bloku/návěští>
3. <počet úrovní pro návrat>
4. <Návrat na začátek programu>

1. <číslo bloku/návěští>

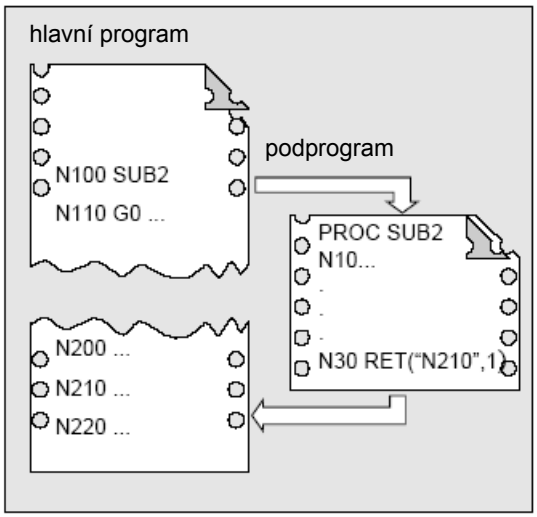
Ve volajícím programu (hlavní program) bude pokračováno blokem, který obsahuje uvedené <číslo bloku/návěští>.



© Siemens AG 2004, Všechna práva vyhrazena

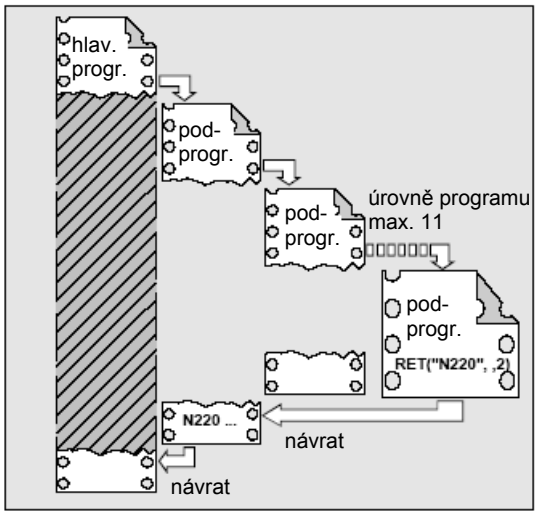
2. <blok za blokem s číslem bloku/návěští>

Návrat z podprogramu se uskutečňuje do bloku za blokem, který obsahuje uvedené <číslo bloku/návěští>.



3. <počet úrovní pro návrat>

Program bude pokračovat v aktuální programové úrovni mínus <počet úrovní pro návrat>.



Nepřípustné úrovně pro návrat

Je-li pro počet úrovní návratu programována

- záporná hodnota
- hodnota vyšší než je hodnota právě aktivní programové úrovně –1(maximálně 11), bude hlášen alarm 14091 s parametrem 5.

Návrat s příkazy SAVE

Při návratu o více programových úrovní budou vyhodnoceny příkazy SAVE jednotlivých programových úrovní .

Při skoku zpět je aktivní modální podprogram

Je-li při návratu o více programových úrovní aktivní nějaký modální podprogram a je-li v některém z přeskočených podprogramů zadán odvolávací příkaz MCALL pro modální podprogram, zůstane modální podprogram i nadále aktivní.

*Uživatel musí **vždy sám zajistit,** aby skoku zpět o více programových úrovní bylo pokračováno se správnými modálními nastaveními.*
Toho je např. dosaženo naprogramováním odpovídajícího hlavního bloku.

Příklad programování 1

Zpracování chyby: Opětovné nasazení v hlavním programu po zpracování příkazu ASUP.

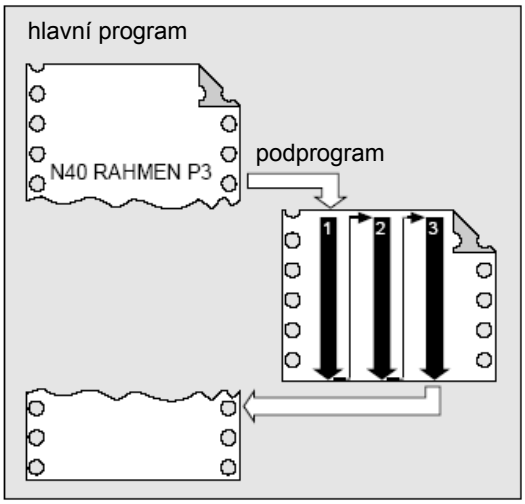
`N10010 CALL "UP1"`	; programová úroveň 0, hlavní program
`N11000 PROC UP1`	; programová úroveň 1
`N11010 CALL "UP2"`	
`N12000 PROC UP2`	; programová úroveň 2
`N19000 PROC ASUP`	; programová úroveň 2 (zpracování ASUP)
`... RET("N10900", , ...`	; programová úroveň 3
`N19100 RET(N10900, ,$P_STACK)`	; návrat z podprogramu
`N10900`	; opětovné nasazení v hlavním programu
`N10910 MCALL`	; vypnutí modálního podprogramu
`N10920 G0 G60 G40 M5`	; korekce dalších modálních nastavení

© Siemens AG 2004, Všechna práva vyhrazena

2.6 Podprogram s opakováním části programu

Opakování programu, P

Má-li být podprogram zpracován několikrát za sebou, může být v bloku s voláním podprogramu pomocí adresy P být naprogramován požadovaný počet opakování tohoto programu.



Příklad:
```
N40 RAHMEN P3
```

Podprogram Rahmen má být zpracován 3-krát za sebou.

Rozsah hodnot
P: 1…9999

Pro každé vyvolání podprogramu platí:
Volání podprogramu musí být vždy programováno v samostatném NC-bloku.

Vyvolání podprogramu s opakováním části programu a předáváním parametrů
Parametry budou předány pouze při volání programu resp. prvním průběhu. Pro další opakování zůstanou parametry nezměněné.

Pokud chcete při opakování části programu parametry změnit, musíte v podprogramu definovat odpovídající deklarace.

2.7 Modální podprogram: MCALL

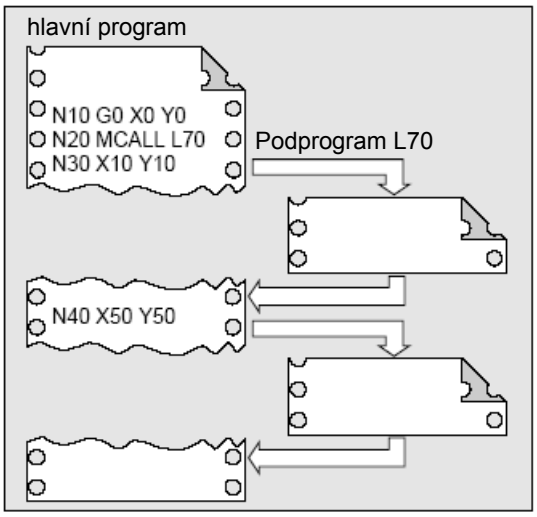
Modální volání podprogramu, MCALL

Prostřednictvím této funkce bude podprogram automaticky vyvolán a zpracován po každém bloku s příkazem pohybu po dráze.
Takto se dá automatizovat vyvolání podprogramů, které mají být zpracovány v různých pozicích obrobku. Například pro výrobu vrtacích vzorů.

© Siemens AG 2004, Všechna práva vyhrazena

Příklady:

```
N10 G0 X0 Y0
N20 MCALL L70
N30 X10 Y10
N40 X50 Y50
```



V blocích N30 a N40 se najíždí na programovanou pozici a následně je zpracován podprogram L70.

```
N10 G0 X0 Y0
N20 MCALL L70
N30 L80
```

V tomto příkladu se nacházejí po sobě jdoucí bloky s programovanými dráhovými osami v podprogramu L80. L70 je vyvolán prostřednictvím L80.

V jednom zpracování programu může být současně zadáno jen jedno volání MCALL. Při volání prostřednictvím MCALL se parametry předávají jen jednou.
Modální podprogram bude volán i bez programování nějakého pohybu v těchto situacích:
Při programování adres S a F, je-li G0 nebo G1 aktivní.
G0/G1 bylo v bloku programováno samotně nebo s dalšími G-kódy.

Vypnutí modálního volání podprogramu
Prostřednictvím příkazu MCALL bez uvedení podprogramu nebo programováním nového modálního volání podprogramu pro nový podprogram.

2.8 Nepřímé volání podprogramu: CALL

Programování

```
CALL <název programu>
```

Vysvětlení

`CALL`	Klíčové slovo pro nepřímé volání podprogramu
`<Název programu>`	Proměnná nebo konstanta typu STRING Název programu obsahujícího část programu, která má být zpracována

© Siemens AG 2004, Všechna práva vyhrazena



Nepřímé volání podprogramu, CALL

V závislosti na daných podmínkách mohou na jednom místě být volány různé podprogramy. Za tím účelem je jméno podprogramu uloženo v proměnné typu STRING. Vyvolání podprogramu proběhne prostřednictvím příkazu CALL a jména proměnné.



Nepřímé vyvolání podprogramu je možné jen u podprogramů bez předání parametrů.



Pro přímé vyvolání podprogramu uložte jeho název do konstanty typu STRING.

Příklad:

Přímé vyvolání konstantou typu STRING:

```
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"
```

Nepřímé vyvolání přes proměnnou:

```
DEF STRING[100] PROGNAME
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"
CALL PROGNAME
```

Podprogram TEIL1 bude přiřazen proměnné PROGNAME. Prostřednictvím příkazu CALL a zadáním cesty bude podprogram nepřímo vyvolán.

© Siemens AG 2004, Všechna práva vyhrazena

2.9 Opakování části programu s nepřímým programováním (od SW 6.4)



Programování

```
CALL <Název programu> BLOCK <Počáteční návěští> TO <Koncové návěští>
CALL BLOCK <Počáteční návěští> TO <Koncové návěští>
```



Vysvětlení

`CALL`	Klíčové slovo pro nepřímé vyvolání podprogramu
`<Název programu> (není nutné)`	Proměnná nebo konstanta typu string Název programu jehož část má být zpracována. Není-li <název programu> naprogramován, bude vyhledána a zpracována část aktuálního programu mezi <počátečním návěštím> a <koncovým návěštím>.
`BLOCK . . . TO . . .`	Klíčové slovo pro nepřímé opakování části programu.
`<Počáteční návěští> <Koncové návěští>`	Proměnná nebo konstanta typu string odkazuje na začátek resp. konec částí programu, které mají být zpracovány



Funkce

Prostřednictvím CALL mohou být nepřímo vyvolány podprogramy, u kterých budou provedena opakování definovaná pomocí příkazu BLOCK a počátečního a koncového návěští.



Příklad programování

```
DEF STRING[20] STARTLABEL, ENDELABEL
STARTLABEL = "LABEL_1"
ENDELABEL = "LABEL_2"
...
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDELABEL ...
M17
PROC CONTUR_1 …
LABEL_1                                   ;začátek opakování části programu
N1000 G1 …
LABEL_2                                   ;konec opakování části programu
```


© Siemens AG 2004, Všechna práva vyhrazena

2.10 Nepřímé volání programu naprogramovaného v jazyce ISO: ISOCALL



Programování

```
ISOCALL <Název programu>
```



Vysvětlení

`ISOCALL`	Vyvolání podprogramu, kterým bude aktivován ve strojových datech nastavený ISO-mód.
`<Název programu>`	Proměnná nebo konstanta typu String Jméno programu, programovaného v nějakém ISO-jazyce.



Funkce

Nepřímým vyvoláním podprogramu pomocí příkazu ISOCALL může být vyvolán program sestavený v jazyce ISO. Přitom je aktivován režim ISO nastavený ve strojních parametrech.
Na konci programu se znovu aktivuje původní režim zpracování. Není-li ve strojních parametrech nastaven žádný režim ISO, proběhne volání podprogramu v režimu Siemens.



Další informace k režimu ISO viz /FBPA/, „Popis funkcí dialektu ISO“

Příklad:

Vyvolání kontury s programováním cyklu z režimu ISO:

```
%_N_0122_SPF                                  Popis kontury v režimu ISO
N1010 G1 X10 Z20
N1020 X30 R5
N1030 Z50 C10
N1040 X50
N1050 M99

N0010 DEF STRING[5] PROGNAME = "0122"         Výrobní program Siemens (- cyklus)
...
N2000 R11 = $AA_IW[X]
N2010 ISOCALL PROGNAME                        Zpracování program 0122.spf v režimu ISO
N2020 R10 = R10+1
N2300 ...
N2400 M30
```

© Siemens AG 2004, Všechna práva vyhrazena

2.11 Volání podprogramu se zadáním cesty a s parametry: PCALL

Programování

Vyvolání podprogramu s absolutním zadáním cesty a předáním parametrů

```
PCALL <Cesta/Název programu>(Parametr 1, …, Parametr n)
```

Vysvětlení

`PCALL`	Klíčové slovo pro vyvolání podprogramu s absolutním zadáním cesty.
`<Název cesty>`	Absolutní zadání cesty začínající „/“ včetně názvu podprogramu. Nebyla-li zadána absolutní cesta, chová se PCALL jako standardní podprogram s identifikátorem programu. Identifikátor programu je zadán bez předpony _N_ a bez přípony. Má-li být jméno programu zadáno bez předpony a bez přípony, musí být explicitně označeno s předponou a příponou jako externí.
`Parametr 1 až n`	Aktuální parametr odpovídající příkazu podprogramu PROC.

Funkce

Prostřednictvím PCALL mohou být vyvolány podprogramy s absolutním zadáním cesty a s předáním parametrů.

Příklad:

```
PCALL/_N_WKS_DIR/_N_WELLE_WPD/WELLE(parametr1, parametr2, …)
```


© Siemens AG 2004, Všechna práva vyhrazena

2.12 Rozšíření cesty při volání podprogramu pomocí CALLPATH (od SW 6.4)



Programování

Vkládání podprogramů uložených mimo existující systém souborů NCK do již existujícího systému souborů NCK.

```
CALLPATH <Název cesty>
```



Vysvětlení

`CALLPATH`	Klíčové slovo pro programovatelné rozšíření cesty. Příkaz CALLPATH se programuje v samostatném řádku výrobního programu.
`<Název cesty>`	Konstanta nebo proměnná typu String Obsahuje absolutní zadání cesty adresáře začínající „/“, o který má být cesta rozšířena. Cesta musí být zadána úplně s předponami a příponami. (např. /_N_WKS_DIR/_N_WST_WPD) jestliže <název cesty> obsahuje prázdný řetězec nebo je-li CALLPATH vyvoláno bez parametrů, je příkaz hledání cesty zrušen. Maximální délka cesty činí 128 Bytes.



Funkce

Prostřednictvím povelu CALLPATH může být rozšířena cesta pro vyvolání podprogramů. Pomocí toho mohou být vyvolány také podprogramy z nevybraného adresáře obrobků bez zadání úplného absolutního názvu cesty. Rozšíření cesty se vkládá před zápis pro uživatelské cykly. (_N_CUS_DIR).

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:

```
CALLPATH ("/_N_WKS_DIR/_N_MYWPD_WPD")
```

Tímto je nastavena následující vyhledávací cesta (pozice 5. je nová):

1. aktuální adresář/ identifikátor podprogramu
2. aktuální adresář/ identifikátor podprogramu _SPF
3. aktuální adresář/ identifikátor podprogramu_MPF
4. /_N_SPF_DIR/ identifikátor podprogramu_SPF
5. **/_N_WKS_DIR/_N_MYWPD/ identifikátor podprogramu_SPF**
6. N_CUS_DIR/_N_MYWPD/ identifikátor podprogramu_SPF
7. /_N_CMA_DIR/ identifikátor podprogramu_SPF
8. /_N_CST_DIR/ identifikátor podprogramu_SPF

Zrušení rozšíření cesty

Rozšíření vyhledávací cesty bude zrušeno následujícími ději

- CALLPATH s prázdným řetězcem
- CALLPATH bez parametrů
- Konec výrobního programu
- Reset

Další upozornění

- Příkaz CALLPATH kontroluje, zda naprogramovaný název cesty skutečně existuje. V případě chyby se zpracování výrobního programu přeruší a aktivuje se alarm korekčního bloku 14009.
- Příkaz CALLPATH může být naprogramován i v souborech INI. Zůstává v platnosti po celou dobu zpracovávání souboru INI (soubor WPD-INI nebo inicializační program pro aktivní data NC, např. framy v 1. kanálu _N_CH1_UFR_INI). Potom se inicializační program znovu resetuje.

© Siemens AG 2004, Všechna práva vyhrazena

2.13 Potlačení zobrazení aktuálního bloku: DISPLOF



Programování

```
PROC … DISPLOF
```



Funkce

Prostřednictvím DISPLOF je pro daný podprogram potlačeno vypisování aktuálního bloku. DISPLOF se nachází na konci příkazu PROC. Namísto aktuálního bloku se bude vypisovat vyvolání cyklu resp. podprogramu.

Standardně je aktivování vypisování bloku. Vypnutí vypisování bloku prostřednictvím DISPLOF platí do návratu z podprogramu nebo do konce programu. Jsou-li z podprogramu atributem DISPLOF vyvolány další podprogramy, je i v nich potlačeno vypisování aktuálního bloku. Je-li podprogram s potlačeným vypisováním přerušen asynchronním programem, budou se vypisovat bloky aktuálního podprogramu.



Příklad programování

Potlačení vypisování aktuálního bloku v cyklu

`%_N_CYCLE_SPF`	
`;$PATH=/_N_CUS_DIR`	
`PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF`	
	;potlačení vypisování aktuálního bloku
	;nyní se jako aktuální blok vypisuje volání cyklu
	;např.: CYCLE(X,100.0)
`DEF REAL DIFF`	;obsah cyklu
`G01 …`	;
…	
`RET`	;návrat z podprogramu, se opět vypisuje následující blok volajícího programu

© Siemens AG 2004, Všechna práva vyhrazena

2.14 Potlačení zpracování blok po bloku: SBLOF, SBLON (od SW 4.3)



Programování

`PROC … SBLOF`	;příkaz se může nacházet v bloku s příkazem PROC nebo v samostatném bloku
`SBLON`	;příkaz se musí nacházet v samostatném bloku



Vysvětlení

`SBLOF`	Deaktivování zpracování blok po bloku
`SBLON`	Opětovné aktivování zpracování blok po bloku



Funkce

Programově specifické potlačení zpracování blok po bloku

Prostřednictvím SBLOF označené programy budou u každého typu zpracování blok po bloku kompletně zpracovány jako jeden blok. SBLOF se nachází na řádku příkazu PROC a platí do konce nebo přerušení podprogramu.

Prostřednictvím příkazu pro návrat je rozhodnuto, zda má být na konci podprogramu zpracování zastaveno nebo ne.

Návrat prostřednictvím M17:
Zastavení na konci podprogramu

Návrat prostřednictvím RET:
Bez zastavení na konci podprogramu

SBLOF platí také ve volaných podprogramech.

Příklad podprogramu bez zastavení v režimu zpracování blok po bloku:

```
PROC BEISPIEL SBLOF
G1 X10
RET
```

Potlačení zpracování blok po bloku v programu

SBLOF musí být v samostatném bloku. Od tohoto bloku bude zpracování blok po bloku deaktivováno až

- do příštího příkazu SBLON nebo
- do konce aktivní úrovně podprogramu.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:

```
N10 G1 X100 F1000
N20 SBLOF
N30 Y20
N40 M100
N50 R10=90
N60 SBLON
N70 M110
N80 ...
```



Oblast mezi N20 a N60 bude v režimu blok po bloku zpracována jako jeden krok.

Blokování zpracování blok po bloku pro asynchronní podprogramy

Aby byl v režimu blok po bloku zpracován příkaz ASUP v jednom kroku, musí být příkaz ASUP naprogramován v příkazu PROC s příkazem SBLOF.
Toto platí také pro funkci „editovatelný systémový ASUP“ prostřednictvím MD 11610: ASUP_EDITABLE.

Příklad pro „editovatelný systémový ASUP“

```
N10 PROC ASUP1 SBLOF DISPLOF
N20 IF $AC_ASUP==’H200’
N30 RET                          žádný Repos při změže režimu
N40 ELSE
N50 REPOSA                       Repos ve všech ostatních případech
N60 ENDIF
```

Ovlivnění programu v režimu zpracování blok po bloku

V režimu zpracování blok po bloku může uživatel zpracovat výrobní program po blocích. Existují následující druhy nastavení funkce blok po bloku:

- SBL1: Blok po bloku IPO se zastavením po každém funkčním bloku stroje.
- SBL2: Blok po bloku se zastavením po každém bloku.
- SBL3: Zastavení v cyklu (aktivováním SBL3 je potlačen příkaz SBLOF).

© Siemens AG 2004, Všechna práva vyhrazena

Potlačení zpracování blok po bloku při vnoření programu

Bylo-li v podprogramu naprogramováno SBLOF v příkazu PROC, při návratu z podprogramu pomocí M17 bude zpracování zastaveno. Tak je zabráněno spuštění následujícího bloku ve volajícím programu.
Je-li v podprogramu se SBLON/SBLOF (nezávisle na atributu SBLOF) deaktivováno potlačení zpracování blok po bloku, bude zpracování zastaveno až po následujícím bloku volajícího programu.
Pokud to není žádoucí, musí být v podprogramu ještě před příkazem návratu (M17) naprogramováno opět SBLON.
Při návratu příkazem RET do vyšší programové úrovně nebude zpracování zastaveno.

Okrajové podmínky

- Vypisování aktuálního bloku může být v cyklech potlačeno prostřednictvím DIPLOF.
- Je-li programováno DISPLOF společně se SBLOF, bude se při zastaveních při zpracování blok po bloku v rámci cyklu vypisovat i nadále vyvolání cyklu.
- Jestliže bylo zastavení při zpracování blok po bloku v režimu systémového ASUP nebo uživatelského ASUP s Bit0=1 resp. Bit1=1 potlačeno prostřednictvím MD 10702 IGNORE_SINGLEBLOCK_MASK, tak může být zastavení v režimu blok po bloku opět aktivováno naprogramováním SBLON v režimu ASUP.
- Zastavení v zpracovávání blok po bloku v uživatelském ASUP je potlačeno prostřednictvím MD 20117: IGNORE_SINGLEBLOCK_ASUP a nemůže být programováním SBLON již znovu aktivováno.
- Aktivováním SBL3 bude potlačen příkaz SBLOF.
- **Od SW 6.4**
  Zastavení v režim blok po bloku typu 2 ignorujte. V typu zpracování blok po bloku 2 (SBL2) se **nebude** v bloku se SBLON zastavovat, je-li v MD 10702: IGNORE_SINGLEBLOCK_MASK dosazeno Bit12=1.



Další informace k vypisování bloku s/bez potlačení zpracování blok po bloku viz /FB/, K1 BAG, Kanál, programový režim „blok po bloku“

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Cyklus se má pro uživatele chovat jako příkaz

Hlavní program

```
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30
```

```
Program cycle:1
N100 PROC CYCLE1 DISPLOF SBLOF          ;potlačení zpracování blok po bloku
N110 R10=3*SIN(R20)+5
N120 IF (R11 <= 0)
N130 SETAL(61000)
N140 ENDIF
N150 G1 G91 Z=R10 F=R11
N160 M17
```

Cyklus CYCLE1 je zpracován v aktivním režimu blok po bloku. Tzn. pro zpracování CYCLE1 musí být jednou zmáčknuto tlačítko Start.

Příklad programování 2

Prostřednictvím PLC spuštěný ASUP pro aktivaci změněného posunutí počátku a korekcí nástroje nemá být viditelný

```
N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF 0 GOTOF
_G500
-->1 GOTOF _G54 2 GOTOF _G55 3
-->GOTOF _G56 4 GOTOF _G57
-->DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET
```

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování 3

S MD 10702: IGNORE_SINGLEBLOCK_MASK, Bit 12=1 nebude od SW 6.4 zastaveno
V typu zpracování blok po bloku SBL2 (zastavení na každém řádku výrobního programu) v příkazu SBLON

```
;SBL2 je aktivní
;$MN_IGNORE_SINGLEBLOCK_MASK = `H1000`;v MD 10702 dosadit: Bit 12=1

N10 G0 X0                          ;zastavit na tomto řádku výrob. programu
N20 X10                            ;zastavit na tomto řádku výrob. programu
N30 CYCLE                          ;cyklem generovaný blok posuvu
     PROC CYCLE SBLOF              ;potlačit zastavení při zpracování blok po
                                   bloku
     N100 R0 = 1
     N110 SBLON                    ;vzhledem k MD 10702: Bit 12=1 se
                                   neprovede žádné zastavení
     N120 X1                       ;zastavit na tomto řádku výrob. programu
     N140 SBLOF
     N150 R0 = 2
     RET
N50 G90 X20                        ;zastavit na tomto řádku výrob. programu
M30
```


© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování 4

Potlačení zpracování blok po bloku při vnoření programu

			;zpracování blok po bloku je aktivní
N10 X0 F1000			;v tomto bloku bude **zastaveno**
N20 UP1(0)			;
	PROC UP1(INT _NR) SBLOF N100 X10		;zpracování blok po bloku deaktivováno
	N100 X10		;
	N110 UP2(0)		
		PROC UP2(INT _NR	;
		N200 X20	;
		N210 SBLON	;zpracování blok po bloku aktivováno
		N220 X22	;v tomto bloku bude **zastaveno**
		N230 UP3(0)	
			PROC UP3(INT _NR)
			N302 SBLOF ;
			N300 X30
			N310 SBLON ;zpracování blok po bloku aktivováno
			N320 X32 ;v tomto bloku bude **zastaveno**
			N330 SBLOF ;zpracování blok po bloku deaktivováno
			N340 X34
			N350 M17 ;SBLOF aktivní
		N240 X24	;v tomto bloku bude **zastaveno**, SBLON aktivní
		N250 M17	;v tomto bloku bude **zastaveno,** SBLON aktivní
	N120 X12		
	N130 M17		; v tomto bloku skoku zpět bude **zastaveno**, SBLOF příkazu PROC aktivní
N30 X0			;v tomto bloku bude **zastaveno**
N40 M30			;v tomto bloku bude **zastaveno**

© Siemens AG 2004, Všechna práva vyhrazena

2.15 Zpracování externího podprogramu: EXTCALL (od SW 4.2)

Programování

```
EXTCALL (<Cesta/Název programu>)
```

Vysvětlení

`EXTCALL`	Klíčové slovo pro volání podprogramu
`(<Cesta/Název programu>)`	Konstanta/proměnná typu STRING. Může být zadán absolutní název cesty nebo název programu. Název programu je zadán s/bez předpony _N_ a bez přípony. Přípona může být připojena ke jménu programu znakem <_>.

Příklad:

```
EXTCALL ("/_N_WKS_DIR/_N_WELLE_WPD/_N_WELLE_SPF") nebo
EXTCALL ("WELLE")
```

Funkce

Prostřednictvím EXTCALL můžete znovu načíst program z HMI v režimu „externího zpracování“. Načítány a zpracovávány přitom mohou být všechny programy, které jsou dosažitelné přes adresářovou strukturu HMI.

Zadání externí cesty programu

Prostřednictvím SD 42700: EXT_PROG_PATH může být flexibilně nastavena vyvolávací cesta. SD 42700 obsahuje zadání cesty, které společně se zadaným identifikátorem podprogramu tvoří absolutní název cesty vyvolávaného programu.

Vyvolání externího podprogramu

Vyvolání externího podprogramu probíhá pomocí příkazu výrobního programu **EXTCALL**.

Z

- názvů naprogramovaných pomocí příkazu EXTCALL a
- nastavovaného parametru SD 42700: EXT_PROG_PATH

vznikne cesta programu pro externí volání podprogramu zřetězením následujících znakových řetězců:

© Siemens AG 2004, Všechna práva vyhrazena

- obsah SD 42700: EXT_PROG_PATH (např. /_N_WKS_DIR/_N_WKST1_WPD
- znak „/“ jako oddělovacího znaku (když je určena cesta pomocí SD 42700: EXT_PROG_PATH)
- v EXTCALL zadaná cesta podprogramu popř. identifikátor podprogramu.

Do parametru SD 42700: EXT_PROG_PATH je předem dosazena mezera. Je-li vyvolán externí podprogram bez absolutního zadání cesty, proběhne na HMI Advanced tatáž vyhledávací cesta jako při vyvolání podprogramu z paměti NCK:

1. aktuální adresář / identifikátor podprogramu
2. aktuální adresář / identifikátor podprogramu_SPF
3. aktuální adresář / identifikátor podprogramu_MPF
4. /_N_SPF_DIR / identifikátor podprogramu_SPF
5. /_N_CUS_DIR / identifikátor podprogramu_SPF
6. /_N_CMA_DIR / identifikátor podprogramu_SPF
7. /_N_CST_DIR / identifikátor podprogramu_SPF

„aktuální adresář“: znamená adresář, ve kterém byl zvolen hlavní program .

„identifikátor podprogramu“: znamená název podprogramu nyprogramovaný prostřednictvím EXTCALL.

Nastavitelná paměť pro načítání (FIFO-Buffer)

Pro zpracování programu v režimu „externího zpracování“ (hlavní program nebo podprogram) je v NCK potřebná paměť pro načítání. Velikost paměti pro načítání je předem nastavena 30 kbytů.
Pomocí MD 18360: MM_EXT_PROG_BUFFER_SIZE může být velikost paměti pro načítání nastavena. Počet pamětí pro načítání je nastaven v MD 18362: MM_EXT_PROG_BUFFER_NUM. Pro všechny programy (hlavní programy nebo podprogramy) , které jsou současně zpracovávány v režimu „externího zpracování“, musí být nastavena paměť pro načítání.

© Siemens AG 2004, Všechna práva vyhrazena

Příklady programování

1. Opětovně zaváděný program se nachází na **lokálním pevném disku HMI Advanced:**

V nastavovaném parametru SD 42700:
EXT_PROG_PATH je uložena následující cesta:
„/_N_WKS_DIR/_N_WST1“.
Hlavní program _N_MAIN_MPF
je v pracovní paměti a je zvolen.

```
N10 PROC MAIN
N20 ...
N30 EXTCALL "SCHRUPPEN"          ;vyvolání externího podprogramu
                                 ;SCHRUPPEN
N40 ...
N50 M30
```

Podprogram „SCHRUPPEN“ (nachází se v adresářové struktuře HMI Advanced pod obrobky -> WST1):

```
N10 PROC SCHRUPPEN
N20 G1 F1000
N30 X=... Y=... Z=...
N40 ...
N90 M17
```

2. Opětovně zaváděný program se nachází na **síťové jednotce nebo kartě ATA v HMI**

EXTCALL Windows – zadání cesty

Vyvolání pro **síťovou jednotku** (HMI Embedded nebo Advanced)
EXTCALL \\R4711\Werkstücke\Kontur1.spf

Vyvolání pro **kartu ATA** (HMI Embedded) např.
EXTCALL C:\Werkstücke\Kontur2.spf

U HMI Embedded musí být vždy zadána absolutní cesta.

Další informace pro obsluhu viz:
/BEM/ HMI Embedded
/BAD/ HMI Advanced

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Externí podprogramy nesmí obsahovat žádné příkazy skoku jako GOTOF, GOTOB, CASE, FOR, LOOP, WHILE, nebo REPEAT.

Volání podprogramů – také vnořená volání EXTCALL jsou možná.

Od SW 6.3
Konstrukty IF-ELSE-ENDIF jsou možné.

POWER ON, RESET
Prostřednictvím Reset a POWER ON budou volání externích podprogramů přerušena a příslušné paměti opětovného zavedení vymazány.

Další informace k „externímu zpracování“ viz /FB/ K1, BAG, Kanál, Programový režim

2.16 Vyvolání podprogramu funkcí M/T

Funkce

Prostřednictvím strojního parametru může být nastaveno, že funkce T resp. M je nahrazena voláním podprogramu.
Toto může být např. využito pro vyvolání rutiny výměny nástroje.
Při vyhledávání bloku se volání podprogramu s funkcemi M/T chová jako standardní volání podprogramu.

Další informace k „externímu zpracování“ viz /FB/ K1, BAG, Kanál, Programový režim

© Siemens AG 2004, Všechna práva vyhrazena

Příklad 1: Výměna nástroje prostřednictvím M6
M funkce M6 je nahrazena rutinou výměny nástroje WZP_UP_M6

```
N10 PROC SCHRUPPEN3
N20 G1 F1000
N30 X=... Y=... Z=...
N40 T1234 M6 ;                          ; vyvolání WZW_UP_M6
M30
```

Příslušný podprogram WZW_UP_M6:

```
N110 PROC WZW_UP_M6
...
N130 G53 D0 G0 X=... Y=... Z=... ;      ; najetí na bod výměny nástroje
N140 M6 ;                               ; provedení výměny nástroje
N190 M17
```

Příklad 2: Výměna nástroje s programováním T funkce
T funkce je nahrazena rutinou výměny nástroje WZW_UP_T

```
N10 PROC SCHRUPPEN4
N20 G1 F1000
N30 X=... Y=... Z=...
N40 T1234 ;                             ; vyvolání WZW_UP_T
M30
```

Příslušný podprogram WZW_UP_T:

```
N310 PROC WZW_UP_T
...
N330 IF $C_T_PROG == 1
N340 G53 D0 G0 X=... Y=... Z=... ;      ; najetí na bod výměny nástroje
N350 T=$C_T                             ; provedení výměny nástroje
N360 ENDIF
...
N390 M17
```


© Siemens AG 2004, Všechna práva vyhrazena

Rozšíření nahrazení T funkce

Od SW 6.4 je nahrazení T funkce rozšířeno tak, že prostřednictvím strojního parametru může být nastaveno, že při současném naprogramování:
D-čísel, resp. DL-čísel, a T-čísel

- v jednom bloku jsou D-, resp. DL-čísla předány je jako parametr cyklu pro náhradu T podle předdefinovaného nastavení **nebo**
- má být provedeno před voláním cyklu pro náhradu T.

Prostřednictvím MD 10719:
T_NO_FCT_CYCLE_MODE se nastaví parametry nahrazení T-funkce
Hodnota 0: jako doposud, D- resp. DL-číslo předat přímo cyklu (předdefinovaná hodnota).
Hodnota 1: D- resp. DL-číslo je započteno přímo v bloku.

Tato funkce je aktivní pouze tehdy, byla-li konfigurována výměna nástroje M funkcí (MD 22550: TOOL_CHANGE_MODE = 1), jinak jsou vždy předány D-, resp. DL- hodnoty.

- in one block, D or DL will be passed as parameters to the T substitution cycle as predefined (default) or
- run before the T substitution cycle call.

2.17 Cykly: Zadávání parametrů uživatelských cyklů

Soubory a cesty



Vysvětlení

`cov.com`	Přehled cyklů
`uc.com`	Popis vyvolání cyklu



Funkce

Prostřednictvím těchto souborů mohou být zadávány parametry vlastních cyklů.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Soubor cov.com je dodán se standardními cykly a je třeba jej odpovídajícím způsobem rozšířit. Soubor uc.com musí vytvořit sám uživatel.

Oba soubory je třeba v pasivním systému souborů uložit do adresáře „uživatelské cykly“ (popř. opatřit odpovídajícím zadáním cesty:
`;$PATH=/_N_CUS_DIR`
v programu).

Přizpůsobení cov.com – Přehled cyklů

Se standardními cykly dodaný soubor cov.com má následující strukturu:

`%_N_COV_COM`	Název souboru
`;$PATH=/_N_CST_DIR`	Zadání cesty
`;Vxxx 11.12.95 Sca Zyklenuebersicht`	Řádek komentáře
`C1(CYCLE81) Bohren, Zentrieren`	Volání pro 1. cyklus
`C2(CYCLE82) Bohren, Plansenken...`	Volání pro 2. cyklus
`...`	
`C24(CYCLE98) Ketten von Gewinden`	Volání pro poslední cyklus
`M17`	Konec souboru

Pro každý nový cyklus je třeba doplnit řádek v následující syntaxi:

`C<číslo > (<jméno cyklu>) text komentáře`
Číslo: libovolné celé číslo, nesmí být v souboru doposud použito;
Název cyklu: název programu cyklu, který má být začleněn
Text komentáře: volitelný text komentáře k cyklu

Příklad:

```
C25 (MEIN_ZYKLUS_1) uživatelský cyklus_1
C26 (SPEZIALZYKLUS)
```


© Siemens AG 2004, Všechna práva vyhrazena

Příklad pro soubor uc.com - popis uživatelských cyklů

Vysvětlení probíhá na základě pokračování příkladu:

Pro oba následující cykly má být nově vytvořeny parametry cyklu:

```
PROC MEIN_ZYKLUS_1 (REAL PAR1, INT PAR2, CHAR PAR3, STRING[10] PAR4)
; Cyklus má následující předávané parametry:
;
;PAR1:          Reálná hodnota v rozsahu –1000.001 <=PAR2<= 123.456,
                předdefinovaná hodnota 100
;PAR2:          Pozitivní celočíselná hodnota mezi 0<=PAR3<=999999,
                předdefinovaná hodnota 0
;PAR3:          1 ASCII znak
;PAR4:          Řetězeco délce 10 pro název podprogramu
;
. . .
M17

PROC SPEZIALZYKLUS (REAL WERT1, INT WERT2)
; Cyklus má následující předávané parametry:
;
;WERT1          Reálná hodnota bez omezení rozsahu hodnot a předdefinované
                hodnoty
;WERT2          Celočíselná hodnota bez omezení rozsahu hodnot a předdefinované
                hodnoty
. . .
M17
```

Příslušný soubor uc.com

```
%_N_UC_COM
;$PATH=/_N_CUS_DIR
//C25(MEIN_ZYKLUS_1) uživatelský cyklus_1
(R/-1000.001 123.456 / 100 /Parametr_2 cyklu)
(I/0 999999 / 1 / celočíselná hodnota)
(C//"A" / znakový parametr)
(S///název podprogramu)

//C26(SPEZIALZYKLUS)
(R///celková délka)
(I/*123456/3/způsob opracování)
M17
```

© Siemens AG 2004, Všechna práva vyhrazena

Syntaktický popis pro soubor uc.com –
Popis uživatelských cyklů

Řádek záhlaví cyklu:
Jako v souboru cov.com s dopředu zadaným „//“
`//C <číslo> (<jméno cyklu>) text komentáře`
Příklad:
`//C25 (MEIN_ZYKLUS_1) uživatelský cyklus_`

Řádek pro popis pro každý parametr:
`(<identifikace typu dat> / <minimální hodnota> <maximální hodnota> / <předdefinovaná hodnota> / <komentář>`

Identifikace typu dat

`R`	pro Real
`I`	pro Integer
`C`	Pro Charakter (1 znak)
`S`	Pro String

Minimální hodnota, maximální hodnota
(může být vynechána)
Meze zadávané hodnoty, které jsou při zadání prověřovány;hodnoty mimo tuto oblast nemohou být zadány.

Mohou být zadány vypočítané hodnoty, které mohou být obsluhovány tlačítkem Toggle; tyto jsou vyčísleny počínaje“∗“, jiné hodnoty jsou pak nepřípustné.

Příklad:
`(I/*123456/1/způsob opracování)`
u typů String a Charakter neexistují meze.

Předdefinovaná hodnota (může být vynechána)
Hodnota, která je při vyvolání cyklu předem dosazena v odpovídající masce; může být obsluhou změněna.

Komentář
Text, maximálně 50 znaků, který je zobrazen v masce vyvolání cyklu před polem pro zadání parametru.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad zobrazení pro oba cykly

Maska zobrazení pro cyklus `MEIN_ZYKLUS_1`

Parametr 2 cyklu	100
Celočíselná hodnota	1
Znakový parametr	
Podprogramy	

Maska zobrazení pro cyklus `SPEZIALZYKLUS`

Celková hodnota	100
Způsob zpracování	1

2.18 Technika maker, DEFINE…AS

Co je to makro?
Jako makro se označuje shrnutí jednotlivých příkazů do nového souhrnného příkazu s vlastním názvem. Také G-, M- a H-funkce nebo L-názvy podprogramů mohou být vytvořeny jako makra. Při vyvolání makra v průběhu programu budou příkazy programované pod názvem makra postupně zpracovány.

Použití maker
Sledy příkazů, které se opakují, se programují jenom jednou jako makro v jednom vlastním prvku makra nebo jednou na začátku programu. Makro může být pak v libovolném hlavním programu nebo podprogramu vyvoláno a zpracováno.

© Siemens AG 2004, Všechna práva vyhrazena

Programování

Makra bývají označena klíčovým slovem
`DEFINE…AS.`
Definice makra zní:
`DEFINE NAME AS <příkaz>`
Příklad:
Definice makra:
`DEFINE LINIE AS G1 G94 F300`
Vyvolání v NC-programu:
`N20 LINIE X10 Y20`

Aktivování makra

- do **SW 4**
  Po POWER ON řídícího systému je makro aktivní.
- od **SW 5**
  Je-li makro zavedeno do NC (programové tlačítko „Load“), je aktivní.

Třímístná M/G funkce (od SW 5)

- do **SW 4**
  Po naprogramování třímístné M funkce je hlášen alarm 12530.
- od **SW 5**
  třímístné M a G funkce mohou být programovány.
  Příklad:
  N20 DEFINE M100 AS M6
  N80 DEFINE M999 AS M6



Další upozornění

Vnoření maker není možné.
H a L funkce jsou programovatelné dvojmístně.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Příklady pro definování maker

`DEFINE M6 AS L6`	Při výměně nástroje je vyvolán podprogram, který převezme nutnou výměnu dat. V podprogramu je vyvolána vlastní M funkce výměny nástroje (např. 106).
`DEFINE G81 AS DRILL(81)`	Emulace DIN G-funkce
`DEFINE G33 AS M333 G333`	Při řezání závitů je požadována synchronizace s PLC. Původní G funkce G33 byla prostřednictvím MD přejmenována na G333, programování zůstane pro uživatele stejné.

Příklad pro globální soubor makra:
Po načtení souboru makra do řídícího systému aktivujte makra (viz výše). Makra mohou být použita jen ve výrobním programu.

`%_N_UMAC_DEF`	
`;$PATH=/_N_DEF_DIR; uživatelsky specifikovaná makra`	
`DEFINE PI AS 3.14`	
`DEFINE TC1 AS M3 S1000`	
`DEFINE M13 AS M3 M7`	;vřeteno doprava, spustit chladicí kapalinu
`DEFINE M14 AS M4 M7`	;vřeteno doleva, spustit chladicí kapalinu
`DEFINE M15 AS M5 M9`	;vřeteno zastavit, vypnout chladicí kapalinu
`DEFINE M6 AS L6`	;vyvolání programu výměny nástroje
`DEFINE G80 AS MCALL`	;zrušení cyklu vrtání
`M30`	;

- Klíčová slova a rezervované názvy nesmí být makry předefinována.
- Prostřednictvím techniky maker může být programovací jazyk řízení silně pozměněn. Používejte proto techniku maker velmi uvážlivě!
- Makra mohou být deklarována v NC-programu. Jako názvy maker jsou přípustné pouze identifikátory. Makra G funkcí mohou být popsána v rámci řídícího systému jen globálně v modulu makra.
- Prostřednictvím techniky maker mohou být definovány libovolné identifikátory, G, M, H funkce a názvy programů L....
- Jsou přípustné identifikátory makra s jedním písmenem a jednou číslicí (**jen FM-NC**).

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Správa souborů a programů

3.1 Přehled 3-140
3.2 Paměť programů 3-141
3.3 Pracovní paměť 3-147
3.4 Definice uživatelských dat 3-150
3.5 Definice úrovní ochrany pro uživatelská data (GUD) 3-155
3.6 Automatická aktivace GUD a MAC (od SW 4.4) 3-157
3.7 Změna úrovní ochrany pro strojní a nastavované parametry 3-159
3.8 Úrovně ochrany pro příkazy NC-jazyka (od SW 7.1) 3-160
3.9 Změna atributů prvků NC- jazyka (od SW 6.4) 3-163
3.10 Příkaz ke strukturování SEFORM v editoru STEP (od SW 6.4) 3-170

© Siemens AG 2004, Všechna práva vyhrazena

3.1 Přehled



Struktura paměti

Uživateli je k dispozici paměťová struktura, která je rozdělena na dvě oblasti.

1. Pracovní paměť
Pracovní paměť obsahuje aktuální systémová a uživatelská data, která řídící systém využívá pro svou práci (aktivní systém souborů).
Příklad:
Aktivní strojní parametry, data korekcí nástroje, posunutí počátku.

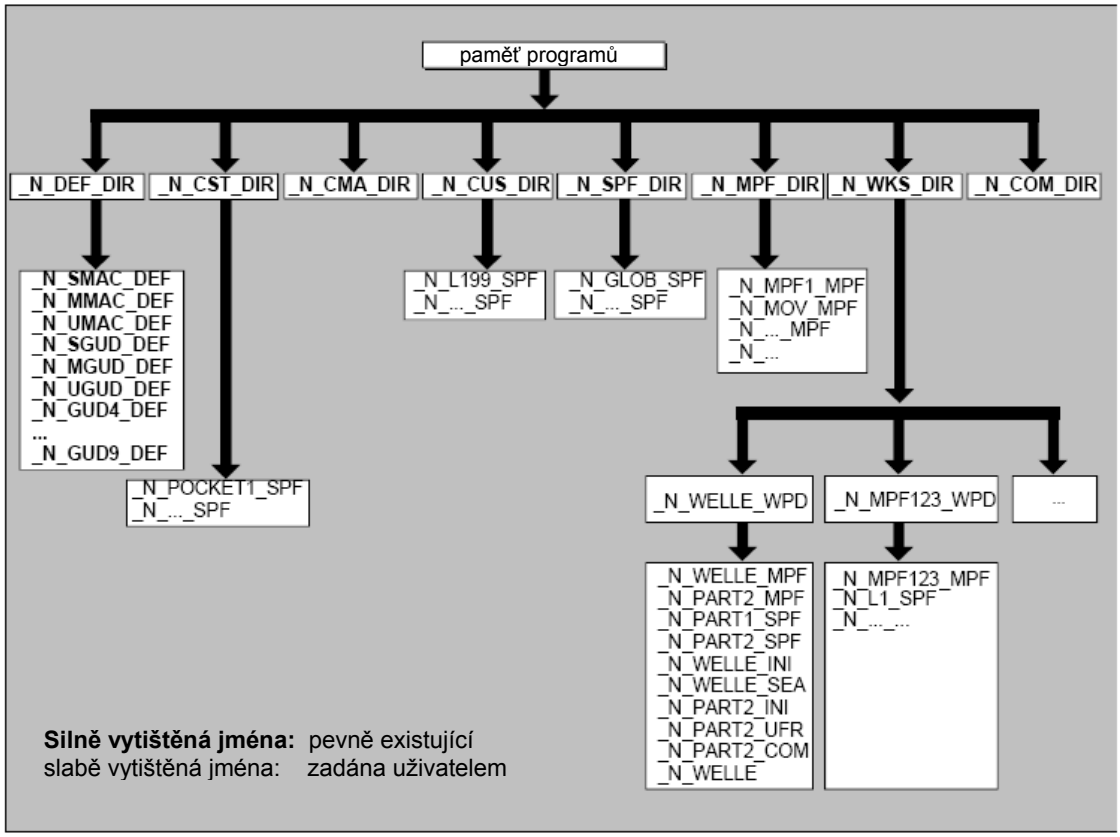
2. Paměť programů
Do paměti programů jsou ukládány soubory a programy a jsou tak trvale uloženy (pasivní systém souborů).
Příklad:
Hlavní programy a podprogramy, definice maker.

© Siemens AG 2004, Všechna práva vyhrazena

3.2 Paměť programů

Přehled

Hlavní programy a podprogramy jsou ukládány do paměti programů. Vedle toho existuje řada typů dat, které zde mohou být dočasně uloženy a v případě potřeby (např. při obrábění určitého obrobku) přeneseny do pracovní paměti (např. pro účely inicializace).



Adresáře

Standardně jsou k dispozici následující adresáře:

1. _N_DEF_DIR	Moduly dat a moduly maker
2. _N_CST_DIR	Standardní cykly
3. _N_CMA_DIR	Cykly výrobce
4. _N_CUS_DIR	Uživatelské cykly
5. _N_WKS_DIR	Obrobky
6. _N_SPF_DIR	Globální podprogramy
7. _N_MPF_DIR	Standardní adresář pro hlavní programy
8. _N_COM_DIR	Standardní adresář pro komentáře

Typy souborů

V paměti programů mohou být uloženy následující typy souborů:

*Name*_MPF	Hlavní program
*name*_SPF	Podprogram

*name*_TEA	Strojní parametry
*Name*_SEA	Nastavované parametry
*name*_TOA	Korekce nástroje
*name*_UFR	Posunutí počátku/framy
*name*_INI	Inicializační soubor
*name*_GUD	Globální uživatelská data
*name*_RPA	R-parametr
*name*_COM	Komentář
*name*_DEF	Definice globálních uživatelských dat a maker

© Siemens AG 2004, Všechna práva vyhrazena

Adresář obrobků, _N_WKS_DIR
Adresář obrobků je standardně zřízen v paměti programů pod označením _N_WKS_DIR.

Adresář obrobků obsahuje pro všechny obrobky, které jste naprogramovali, odpovídající podadresáře obrobků.

Adresáře obrobků, označení WPD
Pro flexibilní nakládání s daty a programy mohou být určitá data a programy ukládány ve složkách nebo v jednotlivých adresářích obrobků.
Adresář obrobku obsahuje všechny soubory, které jsou potřebné pro opracování jednoho obrobku.

Mohou to být hlavní programy, podprogramy, libovolné inicializační programy a soubory komentářů.

Inicializační programy budou jednorázově provedeny po zvolení programu s prvním spuštěním výrobního programu, způsobem odpovídajícím strojnímu parametru: MD 11280: WPD_INI_MODE.

Příklad:
Adresář obrobku `_N_WELLE_WPD`, který byl založen pro obrobek `WELLE`, obsahuje následující soubory:

_N_WELLE_MPF	Hlavní program
_N_PART2_MPF	Hlavní program
_N_PART1_SPF	Podprogram
_N_PART2_SPF	Podprogram
_N_WELLE_INI	Všeobecný inicializační program dat pro obrobek
_N_WELLE_SEA	Inicializační program nastavované parametry
_N_PART2_INI	Všeobecný inicializační program dat pro program Part 2
_N_PART2_UFR	Inicializační program pro data framu pro program Part 2
_N_WELLE_COM	Soubor komentáře

© Siemens AG 2004, Všechna práva vyhrazena

Vytvoření adresáře obrobků na externím PC

Následně popsaný postup je prováděn na externí datové stanici.

Informace o správě souborů a programů (z PC k řídícímu systému) nebo přímo na řídícím systému najdete v návodu k obsluze.

Příkaz ;$PATH

Ve druhém řádku souboru je prostřednictvím příkazu `$PATH=…` zadána cílová cesta.

Příklad:

```
$PATH=/_N_WKS_DIR/_N_WELLE_WPD
```

Soubor je uložen pod uvedenou cestou.

Důležité

Chybí-li zadání cesty, budou soubory typu `SPF` uloženy do adresáře `/_N_SPF_DIR`, soubory s příponou `_INI` do pracovní paměti a všechny ostatní soubory do adresáře `/_N_MPF_DIR`.

Příklad se zadáním cesty pro předchozí příklad:

```
WELLE: _/N_WELLE_MPF bude uložen do
       /_N_WKS_DIR/_N_WELLE_WPD

%_N_WELLE_MPF
;$PATH=/_N_WKS_DIR/_N_WELLE_WPD
N10 G0 X… Z…
•
M2

WELLE: _/N_WELLE_SPF bude uložen do
       /_N_SPF_DIR
•
%_N_WELLE_SPF
•
M17
```


© Siemens AG 2004, Všechna práva vyhrazena

Zvolit obrobek pro opracování
Adresář obrobku může být zvolen pro zpracování v jednom kanálu.
Nachází-li se v tomto adresáři hlavní program **stejného jména** nebo jen jediný hlavní program (MPF), tak bude tento automaticky zvolen pro zpracování.

Příklad:
Adresář obrobku
`/_N_WKS_DIR/_N_WELLE_WPD` obsahuje soubory `_N_WELLE_SPF` a `_N_WELLE_MPF`.

Od SW5 (jen MMC102/103):
Viz „Návod k obsluze“ /BA/ kapitola „Seznam úloh“ a „Volba programu pro zpracování“.

Prohledávané cesty při volání podprogramu
Není-li cesta vyvolání zadána explicitně ve výrobním programu při vyvolání podprogramu (nebo také inicializačního souboru), bude volaný program zjišťován podle pevně dané vyhledávací posloupnosti.

Příklad pro vyvolání podprogramu s absolutním zadáním cesty:
`CALL"/_N_CST_DIR/_N_CYCLE1_SPF"`

Zpravidla jsou programy vyvolávány bez zadání cesty:

Příklad:
`CYCLE1`

Posloupnost prohledávaných cest

1. aktuální adresář /*name*	Adresář obrobku nebo Standardní adresář _N_MPF_DIR
2. aktuální adresář /*name_SPF*	
3. aktuální adresář /*name_MPF*	
4. `/_N_SPF_DIR` /*name_SPF*	Globální podprogramy
5. `/_N_CUS_DIR` /*name_SPF*	Uživatelské cykly
6. `/_N_CMA_DIR` /*name_SPF*	Cykly výrobce
7. `/_N_CST_DIR` /*name_SPF*	Standardní cykly

© Siemens AG 2004, Všechna práva vyhrazena

Programování cesty pro hledání při vyvolání podprogramu (od SW 6.4)

Příkaz CALLPATH

Cesta pro hledání může být při volání podprogramu rozšířena prostřednictvím příkazu výrobního programu `CALLPATH`.

Příklad:

```
CALLPATH („/_N_WKS_DIR/_N_MYWPD_WPD“)
```

Vyhledávací cesta bude uložena před pozicí 5 (uživatelský cyklus) způsobem, odpovídajícím zadanému programování.



Další informace k programovatelné vyhledávací cestě při hledání podprogramu prostřednictvím CALLPATH viz kapitola 2.12

© Siemens AG 2004, Všechna práva vyhrazena

3.3 Pracovní paměť

Inicializační programy
Zde se jedná o programy, jejichž prostřednictvím jsou předem dosazovány hodnoty dat v pracovní paměti (inicializace).

K tomu mohou být použity následující typy souborů:

Name-TEA	Strojní parametry
Name-SEA	Nastavované parametry
Name-TOA	Korekce nástroje
Name-UFR	Posunutí počátku/framy
Name-INI	Inicializační soubory
Name-GUD	Globální uživatelská data
Name-RPA	R-parametry

Datové oblasti
Data mohou být začleněna do rozdílných oblastí, v nichž mají platit. Například může jeden řídící systém disponovat více kanály (ne u 810D CCU1, 840D NCU 571) nebo obvykle také více osami. Existují:

Identifikátory	Datové oblasti
`NCK`	Specifická data pro NCK
`CHn`	Specifická data pro kanál (n udává číslo kanálu)
`Axn`	Specifická data pro osu (n udává číslo osy stroje)
`TO`	Data nástrojů
`COMPLETE`	Všechna data

© Siemens AG 2004, Všechna práva vyhrazena

Vytvoření inicializačního programu na externím PC

S pomocí identifikátorů datových oblastí a datových typů mohou být určeny oblasti, které jsou při zabezpečení dat považovány za jednotku.

Příklad:

`_N_AX5_TEA_INI`	Strojní parametry pro osu 5
`_N_CH2_UFR_INI`	Framy kanálu 2
`_N_COMPLETE_TEA_INI`	Všechny strojní parametry

Po uvedení řídícího systému do provozu existuje v pracovní paměti datový blok, který zaručuje řádný provoz řídícího systému.

Ukládání inicializačních programů

Soubory pracovní paměti mohou být zálohovány na externím PC a odsud pak mohou být znovu načteny.

- soubory jsou zálohovány prostřednictvím `COMPLETE`
- Prostřednictvím `INITIAL` je vytvořen ze všech oblastí jeden INI-soubor: `_N_INITIAL_INI`.

Načítání inicializačních programů

INI-programy mohou být vybrány a vyvolány také jako výrobní programy, pokud používají pouze data jednoho kanálu. Tak je také možné inicializovat programově řízená data.



Informace ke všem typům souborů najdete v návodu k obsluze.

© Siemens AG 2004, Všechna práva vyhrazena

Postup u řídících systémů s více kanály

Příkaz CHANDATA (číslo kanálu) pro více kanálů je přípustný pouze v souboru: `N_INITIAL_INI`.
`N_INITIAL_INI` je soubor pro uvádění do provozu, prostřednictvím kterého jsou inicializována všechna data řídícího systému.

Příklad:

```
%_N_INITIAL_INI
CHANDATA(1)
;přiřazení osy stroje Kanal 1
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED[2]=3
CHANDATA(2)
;přiřazení osy stroje Kanal 2
$MC_AXCONF_MACHAX_USED[0]=4
$MC_AXCONF_MACHAX_USED[1]=5
CHANDATA(1)
;strojní parametry os
;okno přesného najetí hrubé:
$MA_STOP_LIMIT_COARSE[AX1]=0.2 ;osa 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2 ;osa 2
okno přesného najetí jemné:
$MA_STOP_LIMIT_COARSE[AX1]=0.01 ;osa 1
$MA_STOP_LIMIT_COARSE[AX1]=0.01 ;osa 2
```



Ve výrobním programu může být příkaz CHANDATA zadán jen pro ten kanál, ve kterém bude NC-program zpracován; tzn. příkaz může být použit k tomu, aby NC-programy byly chráněny před jejich zpracováním na nezamýšleném kanálu.
V případě chyby je zpracování programu přerušeno.



Upozornění
INI-soubory v seznamech úloh neobsahují žádné příkazy CHANDATA.

© Siemens AG 2004, Všechna práva vyhrazena

3.4 Definice uživatelských dat



Funkce



Definování uživatelských dat (GUD) probíhá v okamžiku uvedení do provozu.
Potřebné strojní parametry musí být zadány odpovídajícím způsobem.
Uživatelská paměť musí být konfigurována.
Všechny globální strojní parametry mají součást jména `GUD`.

- Od **SW 5** (01.99):
  Definice uživatelských dat (GUD) může být provedena na obslužném panelu HMI v systémové oblasti Services. Tím odpadá časově náročné opětovné načtení zálohování dat (`%_N_INITIAL_INI`).
  Přitom platí:
- Definiční soubory, které se nacházejí na pevném disku, nejsou aktivní.
- Definiční soubory, které se nacházejí v NC, jsou vždy aktivní.

Rezervované názvy modulů
V adresáři `/_N_DEF_DIR` mohou být uloženy následující moduly:

_N_SMAC_DEF	Obsahuje definice maker (Siemens – systémové aplikace)
_N_MMAC_DEF	Obsahuje definice maker (výrobce)
_N_UMAC_DEF	Obsahuje definice maker (uživatel)
_N_SGUD_DEF	Obsahuje definice pro globální data (Siemens – syst. aplikace)
_N_MGUD_DEF	Obsahuje definice pro globální data (výrobce)
_N_UGUD_DEF	Obsahuje definice pro globální data (uživatel)
_N_GUD4_DEF	Volně definovatelné
_N_GUD5_DEF	Obsahuje definice pro měřicí cykly (Siemens – systémové aplikace)
_N_GUD6_DEF	Obsahuje definice pro měřicí cykly(Siemens – systémové aplikace)
_N_GUD7_DEF	Obsahuje definice pro standardní cykly(Siemens – systém. aplikace)
_N_GUD8_DEF	Volně definovatelné
_N_GUD9_DEF	Volně definovatelné



Upozornění
Nejsou-li k dispozici žádné měřicí/standardní cykly, pak jsou i pro ně rezervované moduly volně definovatelné.

© Siemens AG 2004, Všechna práva vyhrazena

Programování

Programování jednotlivých proměnných GUD probíhá prostřednictvím příkazu DEF:

```
DEF Bereich VL-stop Typ Name [.., …]=Wert
```

Vysvětlení

`Bereich` (oblast)	Oblast označuje proměnnou jako proměnnou GUD a stanovuje oblast platnosti: `NCK` v rámci NCK `CHAN` v rámci kanálu
`VL-stop` (zastavení předběžného zpracování)	Volitelný atribut předběžného zpracování: `SYNR` zastavení předb. zprac. při čtení `SYNW` zastavení předb. zprac. při zápisu `SYNRW` zastavení předb. zprac. při čtení/zápisu
`Typ`	Typ dat `BOOL` `REAL` `INT` `AXIS` `FRAME` `STRING` `CHAR`
`Name` (název)	Název proměnné
`[.., …]`	Volitelné meze pro proměnné pole
`Wert` (hodnota)	Volitelná hodnota počátečního dosazení, více hodnot u polí oddělených čárkou nebo `REP` (w1), SET (w1, W2, …), (w1,w2,…) U typu frame nejsou možné žádné inicializační hodnoty

Postup

Definice uživatelských dat (GUD)

1. Uložení modulu _N_INITIAL_INI.
2. Vytvoření definičního souboru pro uživatelská data.
   - Na externím PC (do **SW 4**)
   - V systémové oblasti Services (od **SW 5**).
3. Načtení definičního souboru do paměti programů řídícího systému.
4. Aktivování definičních souborů.
5. Uložení dat.

© Siemens AG 2004, Všechna práva vyhrazena

2. Vytvoření definičního souboru pro uživatelská data

Definiční soubory mohou být vytvořeny na externím PC nebo v systémové oblasti Services.
Existují také předdefinované názvy souborů (viz k tomu „Rezervované názvy modulů“):

```
_N_SGUD_DEF
_N_MGUD_DEF
_N_UGUD_DEF
_N_GUD4_DEF … _N_GUD9_DEF
```

Soubory s tímto názvem mohou obsahovat definice proměnné GUD.

3. Zavedení definiční soubor do paměti programů.

Řídící systém vždy založí předem definovaný adresář _N_DEF_DIR.
Tento název se uvádí jako cesta do hlavičky definičního souboru GUD a při načtení přes rozhraní V.24 se vyhodnocuje.

Příklad programování

Příklad pro definiční soubor, globální data (Siemens):

```
%_N_SGUD_DEF
;$PATH=/_N_DEF_DIR
DEF NCK REAL RTP                ;návratová úroveň
DEF CHAN INT SDIS               ;bezpečnostní vzdálenost
M30
```

4. Aktivování definičních souborů

- do **SW 4**
  Před načtením `_N_INITIAL_INI` *uložte všechny programy, framy a strojní parametry, jelikož statická paměť bude formátována.*
  Definiční soubor bude aktivní až po načtení souboru `_N_INITIAL_INI`.
- Od **SW 5**
  Je-li definiční soubor GUD načten do NC (programové tlačítko „Load“), stane se aktivním. Viz „Automatická aktivace …“

© Siemens AG 2004, Všechna práva vyhrazena

5. Uložení dat

Při archivování souboru _N_COMPLETE_GUD z pracovní paměti budou uloženy pouze obsahy souboru. Vytvořené definiční soubory pro globální uživatelské proměnné musí být archivovány odděleně.

Přiřazení proměnných globálním uživatelským datům jsou uložena také v _N_INITIAL_INI, názvy musí být shodné s názvy v definičních souborech.

Příklad pro definiční soubor pro globální data (výrobce stroje):

```
%_N_MGUD_DEF
;$PATH=/_N_DEF_DIR
;globální definice dat výrobce stroje
DEF NCK SYNRW INT STUECKZAHL          ;implicitní zastavení předběžného zpracová-
                                      ní pří čtení/zápisu
                                      ;spec. data se nacházejí v řídícím systému
                                      ;přístup ze všech kanálů
DEF CHAN INT WERKZEUGTABELLE[100]     ;tabulka nástrojů pro kanálově specifické
                                      ;zobrazení čísel nástrojů na místa zásobníku
M30                                   ;tabulka založena odděleně pro každý kanál
```

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Konfigurovatelné oblasti parametrů od SW 7.1
Prostřednictvím strojních parametrů mohou být jednotlivé moduly GUD rozšířeny o kanálové oblasti parametrů, které budou provedeny i v synchronních akcích. Díky tomu se tyto parametry GUD chovají jako R-parametry.
Nově založené oblasti GUD mohou být definovány jako typ dat REAL, INT nebo BOOL, viz seznam předdefinovaných názvů proměnných.

Seznam předdefinovaných názvů proměnných

Názvy parametrů Syntact			
Typu dat **REAL**	Typu dat **INT**	Typu dat **BOOL**	V **modulu**
SYP_RS[]	SYP_IS[]	SYP_BS[]	Modul SGUD
SYP_RM[]	SYP_IM[]	SYP_BM[]	Modul MGUD
SYP_RU[]	SYP_IU[]	SYP_BU[]	Modul UGUD
SYP_R4[]	SYP_I4[]	SYP_B4[]	Modul GUD4
SYP_R5[]	SYP_I5[]	SYP_B5[]	Modul GUD5
SYP_R6[]	SYP_I6[]	SYP_B6[]	Modul GUD6
SYP_R7[]	SYP_I7[]	SYP_B7[]	Modul GUD7
SYP_R8[]	SYP_I8[]	SYP_B8[]	Modul GUD8
SYP_R9[]	SYP_I9[]	SYP_B9[]	Modul GUD9

Další informace najdete v:
/IAD/, kapitola 6 „Zadávání parametrů řídícího systému“

© Siemens AG 2004, Všechna práva vyhrazena

3.5 Definice úrovní ochrany pro uživatelská data (GUD)

Programování

Úrovně ochrany pro celý modul jsou zadány po řádcích hlavičky

```
%_N_MGUD_DEF                    ;druh modulu
;$PATH=/_N_DEF_DIR              ;cesta
APR Wert APW n                  ;úrovně ochrany na samostatném řádku
```

Vysvětlení

`Úroveň ochrany:`	Ochrana přístupu	(**A**cces **P**rotection)
`APWn`	pro zápis	(**W**rite)
`APRn`	pro čtení	(**R**ead)
`n`	Úroveň ochrany n od 0 nebo 10 (nejvyšší úroveň) do 7 nebo 17 (nejnižší úroveň)	

Význam úrovní ochrany n:

0 nebo 10	SIEMENS
1 nebo 11	OEM_HIGH
2 nebo 12	OEM_LOW
3 nebo 13	Koncový uživatel
4 nebo 14	Přepínač na klíč 3
…	…
7 nebo 17	přepínač na klíč 0
APW 0-7, APR 0-7 Přes NC-program a v režimu MDA není možné proměnné prvků zapisovat/číst.	Tyto hodnoty jsou v modulech GUD a v údajích úrovních ochrany přípustné pro jednotlivé proměnné v příkazu REDEF.
APW 10-17, APR 10-17 Přes NC-program a v režimu MDA je možné proměnné prvků zapisovat/číst.	Tyto hodnoty jsou přípustné jen u úrovně ochrany pro specifický modul GUD.

Upozornění

Je-li chráněn kompletní soubor, pak musí být příkazy umístěny před prvními definicemi souboru, jinak v příkazu **REDEF** příslušného data.

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Definováním přístupových kritérií se dají moduly GUD chránit před manipulacemi. V cyklech lze zjišťovat stav proměnných GUD, které jsou tak chráněny před změnou prostřednictvím ovládacího panelu HMI nebo zevnitř programu.
Ochrana přístupu se vztahuje na **všechny** proměnné, které jsou v těchto modulech definovány. Při nedovoleném přístupu hlásí řídící systém odpovídající alarm.

První aktivování definičního souboru GUD
Při prvním aktivování definičního souboru GUD bude možná v něm definované právo přístupu vyhodnoceno a automaticky zpětně přeneseno na práva zápisu nebo čtení definičního souboru GUD.

Upozornění
Zápisy práva přístupu v definičním souboru GUD mohou samy omezit nutné oprávnění přístupu do definičního souboru GUD, ale nemohou je rozšířit.

Příklad:
V definičním souboru _N_GUD7_DEF je: APW2

a) soubor _N_GUD7_DEF má jako ochranu proti zápisu hodnotu 3. Hodnota 3 bude pak přepsána hodnotou 2.
b) Soubor _N_GUD7_DEF má jako ochranu proti zápisu hodnotu 0. Neproběhne žádná změna.

Příkazem APW bude zpětně ovlivněno právo zápisu souboru.
Příkazem APR bude zpětně ovlivněno právo čtení souboru.

Upozornění
Pokud někdo zadá omylem vyšší právo v definičním souboru GUD, než je jemu samotnému přístupné, musí být znovu importován archivní soubor.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Ochrana přístupu je programována v modulu GUD sama před prvním definováním nějaké proměnné s požadovaným stupněm ochrany. Klíčová slova musí být umístěna do samostatného bloku.

Příklad definičního souboru s ochranou přístupu zápis (výrobce stroje), čtení (přepínač na klíč 2):

Příklad programování

```
%_N_GUD6_DEF
;$PATH=/_N_DEF_DIR
APR 15 APW 12                    ;úrovně ochrany pro všechny následující proměnné
DEF CHAN REAL_CORRVAL
DEF NCK INT MYCOUNT
…
M30
```

3.6 Automatická aktivace GUD a MAC (od SW 4.4)

Funkce

Definiční soubory pro definování GUD a maker jsou editovány

- Pro MMC 102/103 v systémové oblasti Services.

Je-li definiční soubor editován v NC, objeví se při opuštění editoru dotaz, zda definice mají být nastaveny do aktivního stavu.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:
„Chcete aktivovat definice ze souboru GUD7.DEF?“
„OK“→ objeví se dotaz, zda momentálně aktivní data mají být zachována.
„Mají být dosavadní data zachována?“
„OK“→ Moduly GUD zpracovávaného definičního souboru jsou uloženy, nové definice jsou aktivovány a uložená data jsou opět načtena.
„Abort“→ Nové definice jsou aktivovány, stará data jsou ztracena.
„Abort“→ Změny v definičním souboru budou odmítnuty, příslušný datový modul se nezmění.

Vyjmutí
Je-li definiční soubor vyjmut, bude příslušný datový modul po zobrazení dotazu vymazán.
Zavedení
Je-li definiční soubor zaveden, objeví se dotaz, má-li být soubor aktivován, popř. mají-li být data zachována. Odmítnete-li aktivaci, nebude soubor zaveden.
Nachází-li se kurzor v zavedeném definičním souboru, změní se popis programového tlačítka z „Load“ na „Activate“, aby bylo možné definice aktivovat. Zvolíte-li „Activate“, objeví se opět dotaz, zda mají být data zachována.


Uložení dat proběhne jen u definičních souborů proměnných, nikoli u maker.



Další upozornění (MMC 103)

V případě, že pro aktivaci definičního souboru není v paměti dostatek místa, musí být po změně velikosti paměti soubor nahrán z NC do MMC a opět zpátky do NC a tím aktivován.

© Siemens AG 2004, Všechna práva vyhrazena

3.7 Změna úrovní ochrany pro strojní a nastavované parametry



Programování

```
REDEF strojní parametry/nastavované parametry úroveň ochrany
```



Vysvětlení

`REDEF`	Nová definice	
`Strojní parametry/nastavované parametry`	Strojní parametry nebo nastavované parametry, kterým má být přiřazena určitá úroveň ochrany.	
`Stupeň ochrany:`	Ochrana přístupu	(**A**cces **P**rotection)
`APWn`	pro zápis	(**W**rite)
`APRn`	pro čtení	(**R**ead)
`n`	Úroveň ochrany n od 0 (nejvyšší stupeň) do 7 (nejnižší stupeň)	



Funkce

Uživatel má možnost **změnit** úrovně ochrany. U strojních parametrů mohou být přiřazeny jen úrovně ochrany nižší priority, u nastavovaných parametrů i vyšší.
Pro předefinování uživatelem nejsou používána žádná hesla. Změna definice proběhne v definičních souborech GUD, např. _N_UGUD_DEF.



Příklad programování

Změna oprávnění u jednotlivých MD

```
%_N_SGUD_DEF
;$PATH=/_N_DEF_DIR
REDEF $MA_CTRLOUT_SEGMENT_NR APR 2 APW 2
REDEF $MA_ENC_SEGMENT_NR APR 2 APW
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 2 APW 2
M30
```

© Siemens AG 2004, Všechna práva vyhrazena

Změna úrovně ochrany

Obnovení nastavení strojních/nastavovaných parametrů

Má-li být změna úrovní ochrany vrácena zpět do původního stavu, musí být opět zapsány původní úrovně ochrany.

Příklad programování

Vrácení oprávnění u jednotlivých MD zpět na původní hodnotu

```
%_N_SGUD_DEF
;$PATH=/_N_DEF_DIR
REDEF $MA_CTRLOUT_SEGMENT_NR APR 7 APW 2
REDEF $MA_ENC_SEGMENT_NR APR 0 APW 0
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 7 APW 7
M30
```

3.8 Úrovně ochrany pro příkazy NC jazyka (od SW 7.1)

Programování

```
REDEF (prvek NC-jazyka) APX Wert
REDEF (systémová proměnná) APW Wert
REDEF (strojní parametr/nastavovaný parametr) APR Wert
REDEF (strojní parametr/nastavovaný parametr) APW Wert
REDEF (strojní parametr/nastavovaný parametr) APR Wert APW Wert
```

Vysvětlení

Prvek NC-jazyka	Prvek jazyka, kterému má být přiřazena úroveň ochrany pro provedení příkazu: 1. Předdefinované podprogramy/funkce (stejnojmenný seznam) 2. Klíčové slovo „DO“ (příkaz „DO“ pro synchronní akce) 3. G-funkce (seznam G-funkcí/podmínek dráhy) 4. Identifikátor programu pro cyklus Cyklus musí být uložen v některém z adresářů cyklů a musí obsahovat příkaz PROC.
Systémová proměnná	Systémová proměnná, které má být přiřazen stupeň ochrany pro zápis (viz seznam systémových proměnných).

© Siemens AG 2004, Všechna práva vyhrazena

`Strojní/nastavované parametry`	Strojní popř. nastavované parametry, jimž má být přiřazen stupeň ochrany pro přístup pro zápis/čtení
	Klíčové slovo pro ochranu přístupu
`APX`	spuštění
`APW, APR`	zápis, čtení
`Hodnota`	Číselná hodnota úrovně ochrany (0 až 7), od 0 (nejvyšší stupeň) do 7 (nejnižší stupeň)
`Hodnota 7`	Poloha přepínače na klíč 0 odpovídá počátečnímu nastavení všech příkazů výrobních programů, které jsou k dispozici

Funkce

Existující koncepce úrovní ochrany pro přístup ke strojním/nastavovaným parametrům a modulům GUD je rozšířena na výše uvedené příkazy výrobních programů. K tomu je příkazem REDEF určitému výrobnímu programu přiřazena úroveň ochrany 0 až 7. Tento příkaz bude nyní při zpracování výrobního programu proveden jen tehdy, existuje-li odpovídající oprávnění pro provedení.

Funkce a použití příkazu REDEF

Příkaz REDEF působí globálně pro všechny kanály a BAG a může být použit:

- Pro předdefinované podprogramy a funkce (předdefinované podprogramy).
- Pro G-kódy podle seznamu G-funkcí/podmínek dráhy.
- Pro přístup pro zápis z výrobního programu nebo synchronní akce k systémové proměnné. Přístup pro čtení je možný vždy.
- Pro příkaz „DO“ u synchronních akcí.
- Pro změnu přístupu pro zápis/čtení u strojních a nastavovaných parametrů jako doposud.
- Pro ochranu volání cyklů.

Další informace viz /BAD/, Návod k použití HMI, kapitola 2
/IAD/, Návod pro uvedení do provozu, Nastavení parametrů řídícího systému, kapitola, „Koncepce úrovní ochrany“

Postup

Analogicky k definicím GUD jsou k dispozici vlastní definiční soubory, které jsou vyhodnocovány při spouštění řídícího systému:

Konečný uživatel: /_N_DEF_DIR/_N_UACCESS_DEF
Výrobce: /_N_DEF_DIR/_N_MACCESS_DEF
Siemens: /_N_DEF_DIR/_N_SACCESS_DEF

Volání podprogramu v definičních souborech
Ve výše uvedených definičních souborech mohou být volány také podprogramy, které obsahují příkazy REDEF.
Tyto příkazy musí být zásadně umístěny jako příkazy DEF na začátku datové části.
Podprogramy musí mít příponu SPF nebo MPF a dědí prostřednictvím $MN_ACCES_WRITE_xACCES nastavenou ochranu proti zápisu definičních souborů.

Příklad:

```
N10 REDEF GEOAX APX 3
N20 IF(ISFILE("/_N_CST_DIR/_N_SACCESS_SUB1_SPF"))
N30 PCALL /_N_CST_DIR/_N_SACCESS_SUB1_SPF
N40 ENDIF
N40 M17
```

Upozornění k REDEF-příkazu od SW 7.1
Když je tato funkce úrovně ochrany pro příkazy NC-jazyka aktivní, musí být všechny doposud založené úrovně ochrany pomocí strojních/nastavovaných parametrů s příkazem REDEF v definičních souborech GUD změněny na nové definiční soubory. Tím je dovoleno zadávání úrovní ochrany pro strojní a nastavované parametry jen ve výše uvedených definičních souborech úrovní ochrany.

Do SW 6.4 existující možnost nezávisle na právě platném oprávnění přístupu ve všech definičních souborech prostřednictvím příkazu REDEF umožnit přístup k libovolným strojním parametrům je nyní odmítnuta hlášením alarmu 15420. Zato jsou pro stupně ochrany 0 až 3 (Siemens, výrobce stroje, konečný uživatel) k dispozici vlastní definiční soubory.

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Zadání inicializačních a synchronizačních atributů je nadále možné jen v definičních souborech GUD.

Úrovně ochrany pro systémovou proměnnou
Úrovně ochrany pro systémovou proměnnou platí jen pro přiřazení hodnot prostřednictvím příkazu výrobního programu. Na ovládacím panelu je v platnosti koncepce úrovní ochrany příslušného systému HMI-Advanced/Embedded.

3.9 Změna atributů prvků NC-jazyka

Programování

```
REDEF prvek NC-jazyka atribut hodnota
```

Vysvětlení

Prvek NC-jazyka		Sem patří
		GUD
		R-parametr
		Strojní/nastavované parametry
		Synchronní proměnné ($AC_PARAM, $AC_MARKER, $AC_TIMER)
		Z výrobních programů zapisovatelné systémové proměnné (viz příloha)
		Uživatelské framy (G500, atd.)
		Konfigurace zásobníku/nástroje
Atribut		**Přípustný u:**
	INIPO	GUD, R-parametry, synchronní proměnné
	INIRE	GUD, R-parametry, synchronní proměnné
	INICF	GUD, R-parametry, synchronní proměnné
	PRLOC	Nastavované parametry
	SYNR	GUD
	SYNW	GUD
	SYNRW	GUD
	APW	Strojní parametry/a nastavované parametry
	APR	Strojní parametry/a nastavované parametry

© Siemens AG 2004, Všechna práva vyhrazena

`Hodnota`	Volitelný parametr u atributů INIPO, INIRE, INCF, PRLOC: dodatečná(é) počáteční hodnota(y) Formy: Jednotlivá hodnota např. 5 Seznam hodnot např. (0,1,2,3,4,5,6,7,8,9) pro proměnnou s 10 prvky `REP`(w1) s w1: opakovaný seznam hodnot pro proměnné s více prvky např. REP(12) `SET(w1, w2, w3, …)` nebo `(w1, w2, w3, …)` seznam hodnot `n` nutný parametr úrovně ochrany u atributů při APR nebo APW pro **GUD** může být při definování zadána počáteční hodnota (DEF NCK INT_MYGUD=5). Není-li tato počáteční hodnota zadána (např. při DEF NCK INT_MYINT), tak může být tato hodnota při REDEF pokynu dodatečně stanovena. **Nepoužitelné** pro R-parametry a systémové proměnné. Mohou být přiřazeny pouze konstanty. Výrazy nejsou jako hodnoty dovoleny.

Funkce

V předchozí podkapitole popsané funkce pro definování datových objektů a stanovení úrovní ochrany se od SW 6.4 prostřednictvím příkazu REDEF stávají zobecněně rozhraním pro nastavení atributů a hodnot.

Význam atributů

INIPO	INIcializovat při Power On Data budou při opětovném spuštění NC ze zálohy přepsána předdefinovanými hodnotami.
INIRE	INIcializovat při resetu ovládacího panelu nebo ukončení výrobního programu Při konci hlavního programu prostřednictvím např. M2 M30, atd. nebo přerušení prostřednictvím resetu budou data přepsána předdefinovanými hodnotami. INIRE je účinný i u INIPO.

© Siemens AG 2004, Všechna práva vyhrazena

INICF	**INI**cializovat při požadavku **N**ew**C**onf nebo příkazu NEWCONF výrobního programu Při požadavku NewConf nebo příkazu NEWCONF výrobního programu budou data přepsána předdefinovanou hodnotou. INICF je účinný i u INIRE a INIPO.

Synchronizaci událostí vyvolávajících inicializaci musí realizovat uživatel. Tzn. je-li např. konec výrobního programu proveden ve dvou **různých kanálech**, tak budou u každého z těchto procesů inicializovány proměnné. Toto se projeví na globálních nebo osových datech!

Standardní předdefinované hodnoty:

Je-li prostřednictvím příkazu
REDEF <název> INIRE, INIPO INICF; PRLOC
změněno chování systémové proměnné nebo GUD, musí být nastaven strojní parametr
MD 11270: DEFAULT_VALUES_MEM_MASK=1
(paměť pro inicializační hodnoty je aktivní).
Nestane-li se tak, vede to k alarmu 12261 „Inicializace nedovolena“.

PRLOC	Jen lokální programová změna. Jsou-li data změněna ve výrobním programu, podprogramu, cyklu nebo Asup, tak nabudou po ukončení hlavního programu (konec pomocí např.M2, M30, atd. nebo při přerušení prostřednictvím Reset na ovládacím panelu) opět svoji původní hodnotu. Tento atribut je přípustný jen pro programovatelné nastavované parametry. Možné jen u GUD:
SYNR	Zastavení předběžného zpracování při čtení
SYNW	Zastavení předběžného zprac. při zápisu
SYNRW	Zastavení předběž. zprac. při čtení a zápisu
APW	Právo přístupu při zápisu
APR	Právo přístupu při čtení Pro strojní a nastavované parametry mohou být předem nastavená oprávnění k přístupu dodatečně přepsána. Přípustné hodnoty přitom sahají od „0“ (heslo firmy Siemens) do „7“ (poloha přepínače na klíč 0)

© Siemens AG 2004, Všechna práva vyhrazena

Okrajové podmínky pro REDEF

Změna atributů NC-objektů může proběhnout teprve **po definování** objektu.

- Obzvláště u GUD je třeba dbát na pořadí DEF…/REDEF.
- (Nastavované parametry/systémové proměnné budou vytvořeny implicitně a již před zpracováním definičních souborů).
- Nejprve musí být vždy symbol definován (implicitně prostřednictvím systému nebo příkazu DEF) a teprve v návaznosti na toto může být prostřednictvím REDEF změněn.

Je-li programováno více vzájemně si konkurujících změn atributu, je vždy aktivní poslední změna.

Atributy polí nemohou být zadány pro jednotlivé prvky, nýbrž vždy jen **pro celé pole**:

```
DEF NCK INT _MYGUD[10,10]
REDEF _MYGUD INIRE                    // ok
REDEF _MYGUD[1,1] INIRE               // není možné, bude aktivován alarm
                                      // (hodnota pole)
```

Inicializování **polí GUD** samotné zůstane nedotčeno.

```
DEF NCK INT _MYGUD[10] =(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
DEF NCK INT _MYGUD[100,100] = REP (12)
DEF NCK INT _MYGUD[100,100] ;
```

Příkazy REDEF s **R-parametrem** musí být zadány se závorkou.

```
REDEF R[ ] INIRE
```


© Siemens AG 2004, Všechna práva vyhrazena

INI-atributy
Je třeba přitom však dbát toho, že při zadávání atributů INI pro tyto proměnné musí být k dispozici odpovídající velikost **paměti pro inicializační hodnoty** nastavitelná prostřednictvím MD 18150: MM_GUD_VAL_MEM. V MD 11270: DEFAULT_VALUES_MEM_MASK musí být zadáno Bit1=1 (paměť pro inicializační hodnoty je aktivní).

Příliš malá paměť má za následek alarm 12261 „Nedovolená inicializace“.

R-parametry a systémové proměnné
Pro R-parametry a systémové proměnné neexistuje možnost zadat předdefinovanou hodnotu odlišnou od nakompilované hodnoty. Vrácení na nakompilovanou hodnotu prostřednictvím INIPO, INIRE nebo INICF je ale možné.

Pro **typ dat FRAME z GUD** nemůže být rovněž (stejně jako již při definování dat) zadána od nakompilované hodnoty odlišná předdefinované hodnoty.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování 1

Chování při Reset u GUD

```
/_N_DEF_DIR/_N_SGUD_DEF
DEF NCK INT _MYGUD1                    ; definice
DEF NCK INT _MYGUD2 = 2
DEF NCK INT _MYGUD3 = 3
```

Inicializování při
Reset na ovládacím panelu/konci výrobního programu

```
REDEF MYGUD2 INIRE                     ; inicializace
M17
```

Tak bude při resetu na ovládacím panelu/konci výrobního programu „_MYGUD2“ opět dosazena hodnota „2“, zatímco „_MYGUD1“ a „_MYGUD33“ si ponechají svoji hodnotu.

Příklad programování 2

Modální omezení počtu otáček ve výrobním programu (nastavované parametry)

```
/_N_DEF_DIR/_N_SGUD_DEF
REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC
M17                                    ;nastavovaný parametr pro mezní otáčky
```

```
/_N_MPF_DIR/_N_MY_MPF
N10 SETMS (3)
N20 G96 S100 LIMS=2500
...
M30
```

Omezení otáček stanovené v nastavovaném parametru ($SA_SPIND_MAX_VELO_LIMS) je 1200 ot/min. Jelikož ve vytvořeném a otestovaném výrobním programu mohou být umožněny i vyšší otáčky, je zde programováno LIMS=2500. Po ukončení programu se ale stane platnou opět hodnota konfigurovaná pomocí nastavovaných parametrů.

© Siemens AG 2004, Všechna práva vyhrazena

Programovatelné nastavované parametry
Následující nastavované parametry je možno inicializovat ve spojení s pokynem REDEF:

Číslo	Identifikátor	G-kód
42000	$SC_THREAD_START_ANGLE	SF
42010	$SC_THREAD_RAMP_DISP	DITS/DITE
43210	$SA_SPIND_MIN_VELO_G25	G25
43220	$SA_SPIND_MAX_VELO_G26	G26
43230	$SA_SPIND_MAX_VELO_LIMS	LIMS
43420	$SA_WORKAREA_LIMIT_PLUS	G26
43430	$SA_WORKAREA_LIMIT_MINUS	G25
43510	$SA_FIXED_STOP_TORQUE	FXST
43520	$SA_FIXED_STOP_WINDOW	FXSW
43700	$SA_OSCILL_REVERSE_POS1	OSP1
43710	$SA_OSCILL_REVERSE_POS2	OSP2
43720	$SA_OSCILL_DWELL_TIME1	OST1
43730	$SA_OSCILL_DWELL_TIME2	OST2
43740	$SA_OSCILL_VELO	FA
43750	$SA_OSCILL_NUM_SPARK_CYCLES	OSNSC
43760	$SA_OSCILL_END_POS	OSE
43770	$SA_OSCILL_CTRL_MASK	OSCTRL
43780	$SA_OSCILL_IS_ACTIVE	OS

Systémové proměnné, které mohou být zapisovány z výrobního programu:
V kapitole 15.2 tohoto popisu najdete seznam systémových proměnných. Všechny systémové proměnné, které jsou ve sloupci „Výrobní program“ označeny W (write) nebo WS (write se zastavením předběžného zpracování), mohou být inicializovány příkazem RESET.

© Siemens AG 2004, Všechna práva vyhrazena

3.10 Příkaz ke strukturování SEFORM v editoru STEP (od SW 6.4)



Programování

```
SEFORM (STRING[128] název úseku, úroveň INT, STRING[128] ikona)
```



Vysvětlení parametrů

SEFORM	Vyvolání funkce příkazu ke strukturování s parametry: název úseku, úroveň a ikona
Název úseku	Identifikátor pracovního úseku
Úroveň	Index pro hlavní úroveň a podúrovně =0 odpovídá hlavní úrovni =1, … odpovídá podúrovni 1 až n
ikona	Název ikony, která má být pro tento úsek zobrazena



Funkce

Příkaz SEFORM je vyhodnocen v editoru STEP, aby mohl být generován přehled kroků pro HMI-Advanced. Přehled kroků je od SW 6.3 k dispozici v HMI-Advanced a slouží k lepší čitelnosti NC-podprogramu. Prostřednictvím příkazu ke strukturování SEFORM je editor STEP (v editoru spočívající podpora programu) podporován prostřednictvím tří uvedených parametrů.

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

- Příkazy SEFORM jsou vytvářeny v editoru STEP.
- Prostřednictvím parametru <název úseku> předaný string je analogicky s příkazem MSG uložen synchronně s hlavním během v BTSS-proměnné. Informace zůstane uchována do přepsání dalšího příkazu SEFROM. Prostřednictvím resetu a konce výrobního programu je obsah vymazán.
- Parametry úroveň a ikona jsou prověřová-ny při zpracování podprogramu pomocí NCK, ale nejsou dále zpracovány.

Další informace k podpoře programu založené na editoru najdete v:
/BAD/ Návod k použití HMI Advanced

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Chráněné oblasti

4.1 Stanovení chráněných oblastí 4-174
4.2 Aktivace, deaktivace chráněných oblastí: CPROT, NPROT 4-178
4.3 Kontrola porušení chráněných oblastí, vymezení pracovního pole a softwarových koncových poloh 4-182

© Siemens AG 2004, Všechna práva vyhrazena

4.1 Stanovení chráněných oblastí CPROTDEF, NPROTDEF

Programování

```
DEF INT NOT_USED
CPROTDEF(n,t,applim,applus,appminus)
NPROTDEF(n,t,applim,applus,appminus)
EXECUTE (NOT_USED)
```

Vysvětlení příkazů

DEF INT NOT_USED	Lokální proměnná, definice typu dat Integer (srovnej kap. 10)
CPROTDEF	Definice chráněné oblasti specifické pro kanál (jen pro NCU 572/573)
NPROTDEF	Definice chráněné oblasti specifické pro stroj
EXECUTE	Ukončení definice

Vysvětlení parametrů

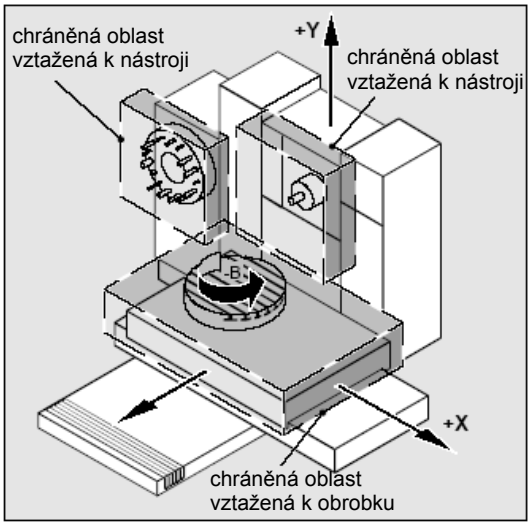
N	Jméno definované chráněné oblasti
T	TRUE = chráněná oblast vztažená k **nástroji** FALSE = chráněná oblast vztažená k **obrobku**
Applim	Druh ohraničení ve 3. rozměru 0 = bez ohraničení 1 = ohraničení v kladném směru 2 = ohraničení v záporném směru 3 = ohraničení v kladném a záporném směru
Applus	Hodnota při ohraničení v kladném směru 3. rozměru
Appminus	Hodnota ohraničení v záporném směru 3. rozměru
NOT_USED	Chybová proměnná je u chráněných oblastí s EXECUTE neúčinná

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Pomocí chráněných oblastí se dají chránit různé prvky stroje, vybavení a obrobky před chybnými pohyby.

Chráněné oblasti vztahující se k nástrojům:
Pro části, které patří k nástrojům (např. nástroje, držák nástroje).
Chráněné oblasti vztahující se k obrobkům:
Pro části, které patří k obrobku (např. části obrobku, upínací stůl, upínací lišta, sklíčidlo vřetena, koník).



Postup

Definice chráněné oblasti
K definování chráněných oblastí patří:

- CPROTDEF pro chráněné oblasti specifické pro kanál
- NPROTDEF pro chráněné oblasti specifické pro stroj
- Popis kontury chráněné oblasti
- Ukončení definice prostřednictvím EXECUTE

Při aktivaci chráněné oblasti ve výrobním programu NC můžete relativně posunout vztažný bod chráněné oblasti.

© Siemens AG 2004, Všechna práva vyhrazena

Vztažný bod popisu kontury
Chráněné oblasti vztahující se k obrobku jsou definovány v základním souřadném systému. Chráněné oblasti vztahující se k nástroji jsou zadány ve vztahu ke vztažnému bodu držáku nástroje F.

Popis kontury chráněných oblastí
Kontura chráněných oblastí je zadána maximálně 11 pohyby posuvu ve zvolené úrovni. Přitom je prvním pohybem posuvu najíždění na konturu. Jako chráněná oblast platí přitom oblast nalevo od kontury. Pohyby posuvu nacházející se mezi CPROTD popř. NPROTDEF a EXECUTE nebudou provedeny, nýbrž definují chráněnou oblast.

Pracovní rovina
Požadovaná rovina je před CPROTDEF popř. NPROTDEF zvolena prostřednictvím příkazu G17, G18, G19 a nesmí být před EXECUTE změněna. Programování aplikáty mezi CPROTDEF popř. NPROTDEF a EXECUTE není přípustné.

Prvky kontury
Přípustné jsou:
- G0, G1 pro přímé prvky kontury
- G2 pro kruhové úseky ve směru hodinových ručiček (jen pro chráněné oblasti vztahující se k obrobku)
- G3 pro kruhové úseky proti směru hodinových ručiček



U SINUMERIK FM-NC jsou k dispozici pro definování jedné chráněné oblasti (max. 4 chráněné oblasti) maximálně 4 konturové prvky.
U 810D jsou k dispozici pro definování jedné chráněné oblasti (max. 4 chráněné oblasti specifické pro kanál a 4 pro NCK) maximálně 4 konturové prvky.

© Siemens AG 2004, Všechna práva vyhrazena

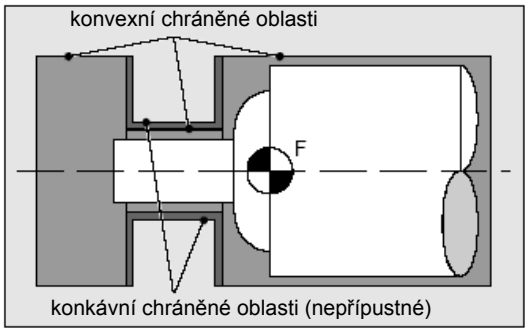
Má-li být chráněná oblast popsána celým kruhem, je třeba jej rozdělit na dva dílčí kruhy. Sled G2, G3 popř. G3, G2 není přípustný. Zde je případně nutno vsunout krátký blok G1.

Poslední bod popisu kontury se musí shodovat s prvním bodem.

Oblasti mimo ochranu (možné jen u chráněných oblastí vztahujících se k obrobku) je nutno definovat **ve směru hodinových ručiček.**

U **rotačně symetrických** chráněných oblastí (např. sklíčidlo vřetena) musíte popsat **celou konturu** (nejenom ke středu otáčení!).

K nástroji se vztahující chráněné oblasti musí být vždy **konvexní**. Je-li žádoucí konkávní oblast ochrany, musí být tato rozložena na více konvexních oblastí ochrany.



Během definování chráněných oblastí nesmí být aktivní

- Žádná korekce rádiusu frézy popř. korekce rádiusu břitu,
- Žádná transformace
- Žádný frame.

Nesmí být také naprogramováno najetí na referenční bod (G74), najetí na pevný bod (G75), zastavení předběžného zpracování v bloku nebo konec programu.

© Siemens AG 2004, Všechna práva vyhrazena

4.2 Aktivace, deaktivace chráněných oblastí: CPROT, NPROT

Programování

```
CPROT (n,state,xMov,yMov,zMov)
NPROT (n,state,xMov,yMov,zMov)
```

Vysvětlení příkazů a parametrů

CPROT	Vyvolání chráněné oblasti specifické pro kanál (jen pro NCU 572/573
NPROT	Vyvolání chráněné oblasti specifické pro stroj
N	Číslo chráněné oblasti
State	Zadání statutu 0 = deaktivování chráněné oblasti 1 = předběžné aktivování chráněné oblasti 2 = aktivování chráněné oblasti
xMov, yMov, zMov	Posunutí již definované oblasti ochrany v geometrických osách

Funkce

Dříve definované chráněné oblasti pro kontrolu kolizí můžete aktivovat, předem aktivovat nebo aktivní chráněné oblasti deaktivovat.

Maximální počet současně v jednom kanálu aktivních chráněných oblastí je stanoven prostřednictvím strojního parametru.

Není-li aktivní žádná chráněná oblast vztahující se k nástroji, je dráha nástroje prověřována vzhledem k chráněným oblastem vztahujícím se k obrobku.

Není-li aktivní žádná chráněná oblast vztahující se k obrobku, chráněná oblast není monitorována.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Stav aktivování

Chráněná oblast je obecně aktivována ve výrobním programu nastavením stavu = 2.

Status je vždy specifický pro kanál, také u chráněných oblastí vztahujících se ke stroji.

Je-li v uživatelském programu PLC počítáno s tím, že chráněná oblast může být účinně zadána prostřednictvím uživatelského programu PLC, je zapotřebí uskutečnit předběžnou aktivaci nastavením stavu = 1.

Deaktivace a tím vypnutí chráněných oblastí se uskuteční nastavením stavu = 2. Přitom není nutné žádné posunutí.

Posunutí chráněných oblastí při (předběžném) aktivování

Posunutí může proběhnout v 1, 2 nebo 3 rozměrech.
Zadání posunutí se vztahuje k:

- Počátku souřadné soustavy stroje u chráněných oblastí specifických pro obrobek,
- Vztažnému bodu držáku nástroje F u chráněných oblastí vztahujících se k nástroji.

Další upozornění

Chráněné oblasti mohou být aktivovány již po rozběhu a následném najetí na referenční bod. K tomu musí být zadána systémová proměnná `$SN_PA_ACTIV_IMMED [n]` popř. `$SN_PA_ACTIV_IMMED [n] = TRUE`.
Oblasti jsou vždy aktivovány nastavením stavu = 2 a nemají žádné posunutí.

© Siemens AG 2004, Všechna práva vyhrazena

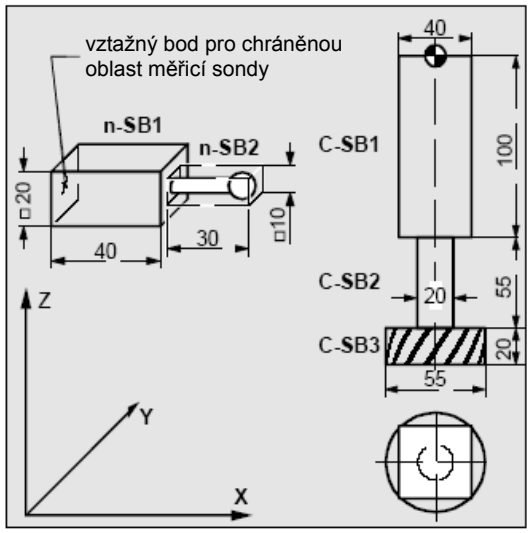
Vícenásobné aktivování chráněných oblastí

Jedna chráněná oblast může být současně aktivní i ve více kanálech (např. pinola u dvou proti sobě ležících suportů).
Kontrola chráněných oblastí proběhne jen v případě, že se všemi geometrickými osami bylo najeto na referenční bod. Přitom platí:

- Chráněnou oblast není možno aktivovat v jednom kanálu současně vícenásobně s různými posunutími.
- Chráněné oblasti vztahující se ke stroji musí v obou kanálech vykazovat stejnou orientaci.

Příklad programování

U frézy má být kontrolována možná kolize frézy s měřicí sondou. Poloha měřicí sondy má být při aktivaci zadána posunutím.
Budou pro to definovány následující chráněné oblasti:



- Vždy jedna chráněná oblast specifická pro stroj a obrobek vztahující se na držák měřicí sondy (n-SB1) a pro měřicí sondu samotnou (n-SB2).

- Vždy jedna chráněná oblast specifická pro kanál a nástroj vztahující se na držák frézy (c-SB1), stopku frézy (c-SB2) a pro frézu samotnou (c-SB3).

Orientace všech oblastí ochrany leží ve směru Z.

Poloha vztažného bodu měřicí sondy při aktivaci má být X=-120, Y=60 a Z=80.

© Siemens AG 2004, Všechna práva vyhrazena

```
DEF INT SCHUTZB                          Definování pomocné proměnné
```

Definování chráněných oblastí Nastavení orientace

```
G17
```

```
NPROTDEF(1,FALSE,3,10,-10)               Chráněná oblast n-SB1
G01 X0 Y-10
X40
Y10
X0
Y-10
EXECUTE(SCHUTZB)
```

```
NPROTDEF(2,FALSE,3,5,-5)                 Chráněná oblast n-SB2
G01 X40 Y-5
X70
Y5
X40
Y-5
EXECUTE(SCHUTZB)
```

```
CPROTDEF(1,TRUE,3,0,-100)                Chráněná oblast c-SB1
G01 X-20 Y-20
X20
Y20
X-20
Y-20
EXECUTE(SCHUTZB)
```

```
CPROTDEF(2,TRUE,3,-100,-150)             Chráněná oblast c-SB2
G01 X0 Y-10
G03 X0 Y10 J10
X0 Y-10 J-10
EXECUTE(SCHUTZB)
```

```
CPROTDEF(3,TRUE,3,-150,-170)             Chráněná oblast c-SB3
G01 X0 Y-27,5
G03 X0 Y27,5 J27,5
X0 Y27,5 J-27,5
EXECUTE(SCHUTZB)
```

Aktivování chráněných oblastí

```
NPROT(1,2,-120,60,80)                    Aktivování chráněné oblasti n-SB1 s posunutím
NPROT(2,2,-120,60,80)                    Aktivování chráněné oblasti n-SB2 s posunutím
CPROT(1,2,0,0,0)                         Aktivování chráněné oblasti c-SB1 s posunutím
CPROT(2,2,0,0,0)                         Aktivování chráněné oblasti c-SB2 s posunutím
CPROT(3,2,0,0,0)                         Aktivování chráněné oblasti c-SB3 s posunutím
```

© Siemens AG 2004, Všechna práva vyhrazena

4.3 Kontrola porušení chráněných oblastí, vymezení pracovního pole a softwarových koncových poloh

Programování

Status=CALCPOSI(_STARTPOS,_MOVDIST,_DLIMIT,_MAXDIST,_BASE_SYS,_TESTLIM)

Vysvětlení

Status 0: Funkce o.k., zadaná dráha může být zcela projeta.
-1: V _DLIMIT je nejméně jedna komponenta negativní
-2: Ve výpočtu transformace se vyskytla chyba
Nemůže-li být zadaná dráha zcela projeta, je reakcí pozitivní, decimálně kódovaná hodnota:

Místo jednotek (druh porušené meze):

1: Dráha posuvu omezena softwarovými koncovými polohami.
2: Dráha posuvu omezena ohraničením pracovního pole.
3: Dráha posuvu omezena chráněnými oblastmi.
Je-li současně porušeno více mezí (např. softwarové koncové polohy a chráněné oblasti), je místem jednotek hlášeno ohraničení, které vede k nejvýraznějšímu omezení zadané dráhy posuvu.

Místo desítek

10: Ohraničení narušeno počáteční hodnotou.
20: Zadaná přímka porušuje ohraničení. Tato hodnota je reakcí i tehdy, pokud koncový bod sám neporušuje žádné ohraničení, ale na dráze od počátečního ke koncovému bodu se vyskytlo porušení hodnoty ohraničení (např. projetí chráněné oblasti, zakřivené softwarové koncové polohy v souřadném systému obrobku při nelineárních transformacích, např. Transmit).

Místo stovek

100: Pozitivní mezní hodnota je porušena (jen, když na místě jednotek je 1 nebo 2, tzn. u softwarových koncových poloh a vymezení pracovního pole).
100: Je porušena chráněná oblast specifická pro NCK (jen, když místo jednotek je 3).
200: Je porušena negativní mezní hodnota (jen, když místo jednotek je 1 nebo 2, tzn. u softwarových koncových poloh a vymezení pracovního pole)
200: Je porušena chráněná oblast specifická pro kanál (jen, když místo jednotek je 3).

Místo tisíců

1000: Faktor, kterým je násobeno číslo osy, která porušuje ohraničení (jen, když místo jednotek je 1 nebo 2, tzn. u softwarových koncových poloh a vymezení pracovního pole).
Počítání os začíná u 1 a vztahuje se u porušených softwarových koncových poloh (místo jednotek = 1) k osám stroje a při porušeném vymezení pracovního pole (místo jednotek = 2) ke geometrickým osám.

© Siemens AG 2004, Všechna práva vyhrazena

1000: Faktor, kterým je číslo porušené chráněné oblasti násobeno (jen, když je místo jednotek 3).
Je-li porušeno více chráněných oblastí, je na místech stovek a tisíců hlášena chráněná oblast, která vede k **nejvýraznějšímu** omezení zadané dráhy pohybu.

_STARTPOS Počáteční hodnota pro abscisu[0], ordinátu[1] a aplikátu[2] ve (WKS)

_MOVEDIST Zadání dráhy inkrementálně pro abscisu[0], ordinátu[1] a aplikátu[2]

_DLIMIT [0] -[2]: Minimální vzdálenosti, které jsou přiřazeny geometrickým osám.
[3]: Minimální vzdálenost, která je přiřazena lineární ose stroje při nelineární transformaci, když nemůže být jednoznačně přiřazena žádná geometrická osa.
[4]: Minimální vzdálenost, která je přiřazena kruhové ose stroje při nelineární transformaci, když nemůže být jednoznačně přiřazena žádná geometrická osa.
Pouze u speciálních transformací, kdy mají být monitorovány softwarové koncové polohy.

_MAXDIST Pole [0] - [2] pro vrácenou hodnotu. Inkrementální dráha ve všech třech geometrických osách, aniž by zadaná minimální vzdálenost jedné meze osy v účastnících se osách stroje nebyla dosažena. Není-li dráha pohybu omezena, je obsah tohoto parametru odmítnutí stejný s obsahem _MOVDIST.

_BASE_SYS FALSE:
Nebo parametr není zadán: Při vyhodnocování údajů o pozicích a délkách je vyhodnocen G-kód skupiny 13 (G70, G71, G700, G710; palce/metricky). Při aktivním G70 a metrickém základním systému (resp. aktivním G71 a základním systému v palcích) jsou v základním systému produkovány systémové proměnné $AA_IW[X] a $AA_MW[X] vztahující se k systému souřadnic nástroje a musí být případně pro použití přepočítány prostřednictvím funkce CALCPOSI.

TRUE:
Při posuzování údajů o pozicích a délkách je stále používán základní systém řídícího systému nezávisle na hodnotě aktivního G-kódu skupiny 13.

_TESTLIM Omezení, která mají být monitorována (binárně kódovaná):
1: kontrolovat softwarové koncové polohy
2: kontrolovat vymezení pracovního pole
3: kontrolovat aktivované chráněné oblasti
4: kontrolovat předem aktivované chráněné oblasti
Zadává se kombinace sčítáním hodnot. Předdefinováno: 15; prověřit všechny.



Funkce

Funkce CALCPOSI slouží k prověření, zda počínaje daným počátečním bodem se mohou geometrické osy pohybovat zadanou drahou, aniž by porušily meze os (softwarové koncové polohy), vymezení pracovního pole nebo chráněné oblasti.

Pro případ, že zadaná dráha nemůže být absolvována, je reakcí maximální přípustná hodnota.
Funkce CALCPOSI je předdefinovaný podprogram.
Musí se nacházet samostatně v samostatném bloku.

Zvláštní případy a další podrobnosti
Všechny údaje o dráze jsou vždy v kótovány jako rádiusy, a to i u příčné osy s aktivním G-kódem "DIAMON".
Nemůže-li být dráha jedné z ze zúčastněných os zcela absolvována, budou ve vrácené hodnotě _MAXDIST odpovídajícím způsobem redukovány i dráhy ostatních os, a to tak, že výsledný koncový bod leží na zadané dráze.

Je přípustné, aby pro jednu nebo více zúčastněných os nebyly definovány žádné softwarové koncové polohy, resp. vymezení pracovního pole.
Všechny meze budou kontrolovány jen, pokud bylo s příslušnými osami **najeto na referenční bod**.
Případné kruhové osy jsou kontrolovány jen v případě, že se nejedná o modulo.

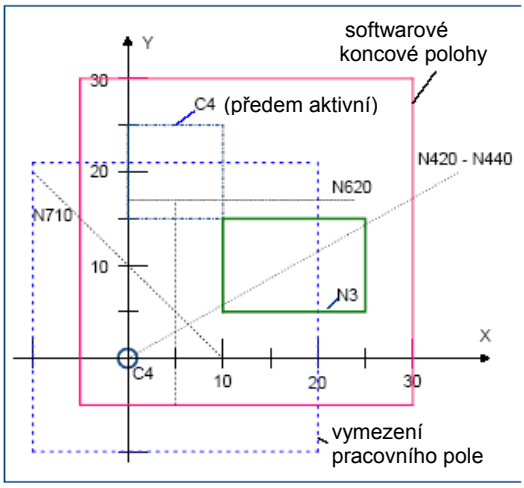
Kontrola softwarových koncových poloh a vymezení pracovního pole v normálním provozu pohybu závislá na aktivních nastaveních (signály rozhraní pro výběr softwarové koncové polohy1 popř. softwarové koncové polohy 2, G-kód WALIMON/WALIMOF, data nastavení pro individuální aktivování ohraničení pracovního pole a pro zjištění, zda má při kontrole mezí pracovního pole být zohledněn rádius aktivního nástroje nebo ne).
Při určitých kinematických transformacích (např. Transmit) nemůže být pozice os stroje jednoznačně určena (víceznačnost) z pozic v souřadného systému obrobku (WKS).
V normálním provozu pohybu vyplývá jednoznačnost zpravidla z předchozího děje a podmínky, že kontinuálnímu pohybu ve WKS musí odpovídat kontinuální pohyb os stroje. Při kontrole softwarových koncových poloh pomocí funkce CALCPOSI je proto v takových případech použita k zamezení mnohoznačnosti současná pozice stroje. Případně musí být proto před CALCPOSI programováno **STOPRE**, aby měla funkce k dispozici platnou pozici os stroje.

© Siemens AG 2004, Všechna práva vyhrazena

Není zaručeno, že vzdálenost specifikovaná ke chráněným oblastem v _DLIMIT[3] bude při pohybu po zadané dráze pohybu všude dodržena. Je pouze zaručeno, že při prodloužení koncového bodu vráceného v _MOVDIST o tuto vzdálenost nebude porušena žádná chráněná oblast. Přímka může ve svém průběhu vést libovolně těsně kolem chráněné oblasti.

Příklad programování

V příkladu (obrázek) jsou v X-Y-úrovni zakresleny softwarové koncové polohy a vymezení pracovního pole.
Navíc jsou definovány tři chráněné oblasti, obě chráněné oblasti specifické pro kanál C2 a C4 jakož i chráněná oblast N3 specifická pro NCK. C2 je kruhovitá, aktivní, k nástroji se vztahující chráněná oblast s rádiusem 2 mm. C4 je kvadratická, předem aktivovaná a k obrobku se vztahující chráněná oblast s délkou strany 10 mm a N3 je pravoúhlá aktivní chráněná oblast s délkou strany 10 mm, resp. 15 mm.
V následujícím NC-programu jsou nejprve definovány dle nákresu chráněné oblasti a vymezení pracovního pole a následně je vyvolána funkce CALCPOSI s různými zadáními parametrů.
Výsledky jednotlivých volání CALCPOSI jsou shrnuty v tabulce na konci příkladu.




```
N10 def real _STARTPOS[3]
N20 def real _MOVDIST[3]
N30 def real _DLIMIT[5]
N40 def real _MAXDIST[3]
N50 def int _SB
N60 def int _STATUS

N70 cprotdef(2, true, 0)
N80 g17 g1 x-2 y0
N90 g3 i2 x2
N100 i-2 x-2
N110 execute(_SB)

N120 cprotdef(4, false, 0)          ; chráněná oblast vztahující se k obrobku
N130 g17 g1 x0 y15
N140 x10
N150 y25
```

```
N160 x0
N170 y15
N180 execute(_SB)

N190 nprotdef(3, false, 0)          ; oblast ochrany vztahující se ke stroji
N200 g17 g1 x10 y5
N210 x25
N220 y15N230 x10
N240 y5
N250 execute(_SB)

N260 cprot(2,2,0, 0, 0)             ; aktivovat resp. předem aktivovat chráněné
N270 cprot(4,1,0, 0, 0)             ; oblasti
N280 nprot(3,2,0, 0, 0)

N290 g25 XX=-10 YY=-10              ; definovat vymezení pracovního pole
N300 g26 xx= 20 yy= 21
N310 _STARTPOS[0] = 0.
N320 _STARTPOS[1] = 0.
N330 _STARTPOS[2] = 0.

N340 _MOVDIST[0] = 35.
N350 _MOVDIST[1] = 20.
N360 _MOVDIST[2] = 0.

N370 _DLIMIT[0] = 0.
N380 _DLIMIT[1] = 0.
N390 _DLIMIT[2] = 0.
N400 _DLIMIT[3] = 0.
N410 _DLIMIT[4] = 0.

; různá vyvolání funkcí
N420 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)
N430 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,3)
N440 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,1)

N450 _STARTPOS[0] = 5.              ; jiný počáteční bod
N460 _STARTPOS[1] = 17.
N470 _STARTPOS[2] = 0.

N480 _MOVDIST[0] = 0.               ; jiný počáteční bod
N490 _MOVDIST[1] =-27.
N500 _MOVDIST[2] = 0.

; různá vyvolání funkcí
N510 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,14)
N520 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)

N530 _DLIMIT[1] = 2.
N540 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)
N550 _STARTPOS[0] = 27.
N560 _STARTPOS[1] = 17.1
N570 _STARTPOS[2] = 0.

N580 _MOVDIST[0] =-27.
N590 _MOVDIST[1] = 0.
```


© Siemens AG 2004, Všechna práva vyhrazena

```
N600 _MOVDIST[2] = 0.

N610 _DLIMIT[3] = 2.
N620 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 12)

N630 _STARTPOS[0] = 0.
N640 _STARTPOS[1] = 0.
N650 _STARTPOS[2] = 0.
N660 _MOVDIST[0] = 0.
N670 _MOVDIST[1] = 30.
N680 _MOVDIST[2] = 0.
N690 trans x10
N700 arot z45
N710 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)
N720 M30
```

Výsledky prověřování v příkladu

Blok č. N…	_STATUS	_MAXDIST [0] (=X)	_MAXDIST [1] (=Y)	Poznámky
420	3123	8.040	4.594	Je porušena chráněná oblast N3.
430	1122	20.000	11.429	Žádná kontrola chráněné oblastí, porušení vymezení pracovního pole.
440	1121	30.000	17.143	Pouze kontrola softwarových koncových poloho je aktivní.
510	4213	0.000	0.000	Je porušen počáteční bod chráněné oblasti C4
520	0000	0.000	-27.000	Předem aktivovaná chráněná C4 není kontrolována. Zadaná dráha může být zcela absolvována.
540	2222	0.000	-25.000	Vzhledem k _DLIMIT[1]=2 je dráha posuvu omezena vymezením pracovního pole.
620	4223	-13.000	0.000	Vzdálenost k C4 vzhledem k C2 a _DLIMIT[3] celkem 4 mm. Vzdálenost C2 – N3 0.1 mm nevede k omezení dráhy posuvu.
710	1221	0.000	21.213	Frame s translací a rotací je aktivní. Přípustná dráha pohybu v _MOVDIST platí v posunutém a otočeném souřadném systému obrobku.



Další upozornění

Podrobnosti k vymezení pracovního pole najdete v návodu k programování PG, k softwarovým koncovým polohám v popisu funkcí A3.

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Speciální příkazy pro pohyb po dráze

5.1 Najetí na kódované pozice, CAC, CIC, CDC, CACP, CACN, ... 5-190
5.2 Splinová interpolace, ASPLINE, B-/CSPLINE, BAUTO, BNAT, BTAN 5-191
5.3 Kompresor COMPOF/ON, COMPCURV, COMPCAD, (SW 6.2) 5-199
5.4 Polynomická interpolace – POLY, POLYPATH (od SW 5)..... 5-207
5.5 Nastavitelná referenční dráha, SPATH, UPATH 5-213
5.6 Měření spínací sondou, MEAS, MEAW..... 5-217
5.7 Rozšířené funkce měření, MEASA, MEAWA, MEAC Od SW 4, opce) 5-220
5.8 Speciální funkce pro OEM-uživatele, G810 až G829 5-230
5.9 Redukce posuvu se zpomalením v rozích, G62, G621(od SW 6.1) 5-230
5.10 Programovatelné kritérium konce pohybu (od SW 5.1)..... 5-232
5.11 Programovatelná sada servo-parametrů (od SW 5.1)..... 5-235

© Siemens AG 2004, Všechna práva vyhrazena

5.1 Najetí na kódované pozice, CAC, CIC, CDC, CCP, CACN

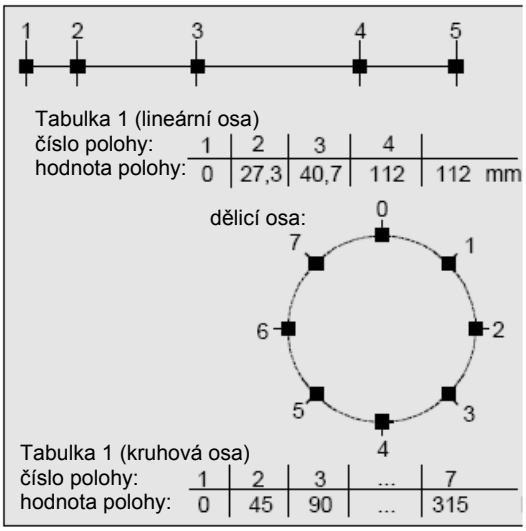
Vysvětlení příkazů

CAC (n)	Absolutní najíždění na kódovanou pozici
CIC (n)	Inkrement. najíždění na kódovanou pozici o n-míst vpřed (+) nebo vzad (-)
CDC (n)	Najet na kódovanou pozici po nejkratší dráze (jen pro kruhové osy)
CACP(n)	Kódovaná pozice absolutně v pozitivním směru (jen pro kruhové osy)
CACN(n)	Kódovaná pozice absolutně v negativním směru (jen pro kruhové osy
(n)	Číslo pozice 1, 2, … max. 60 pozic na osu

Postup

Prostřednictvím strojních parametrů můžete pro dvě osy zadat vždy maximálně 60 (0 až 59) poloh v tabulkách poloh.

Příklad možné tabulky pozic viz obrázek.



Další vysvětlení

V případě, že se osa nachází mezi dvěma polohami, nedojde při inkrementálním zadání s CIC (…) k pohybu.
Doporučuje se programovat první příkaz pohybu vždy s absolutním udáním polohy.

Příklad programování

N10 FA[B] = 300	Posun vpřed pro osu pozice B
N20 POS[B] = CAC (10)	Najet na kódovanou polohu (absolutně)
N30 POS[B] = CIC (-4)	Jeď z aktuální polohy 4 místa vzad

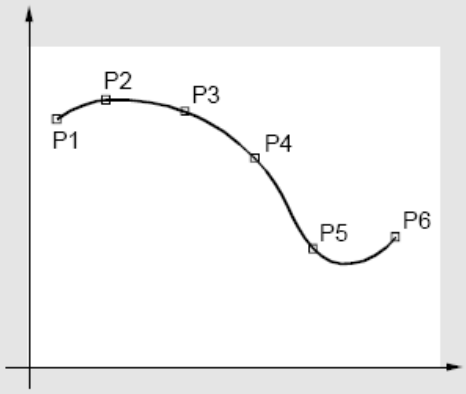
© Siemens AG 2004, Všechna práva vyhrazena

5.2 Splinová interpolace, ASPLINE, B-/CSPLINE, BAUTO, BNAT, BTAN

Úvod

Prostřednictvím splinové interpolace mohou být sledy bodů spojeny plynulými křivkami. Spliny mohou být použity např. pro spojení digitalizovaných bodů do křivek.

Existují různé typy splinů s rozdílnými vlastnostmi, které také vedou k rozdílným výsledkům. Vedle výběru typu splinu má uživatel ještě možnost ovlivnit řadu parametrů. Často je potřeba několika pokusů, než je dosaženo požadovaného obrazu.



P1 až P6: zadané souřadnice

Programování

```
ASPLINE X Y Z A B C
```

nebo

```
BSPLINE X Y Z A B C
```

nebo

```
CSPLINE X Y Z A B C
```

Funkce

Spline programujete, má-li být sled bodů spojen křivkou.

Jsou možné tři druhy splinů:

- A-spline (Akima spline)
- B-spline (neuniformní, racionální základní spline, NURBS)
- C-spline (kubický spline)

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

A-, B- a C-spliny mají modální platnost a patří ke skupině příkazů dráhy. Korekce rádiusu nástroje je použitelná. Kontrola kolize probíhá v projekci do roviny.

Interpolační osy ve svazku splinu jsou zvoleny příkazem SPLINEPATH (více informací na následujících stranách).

Postup

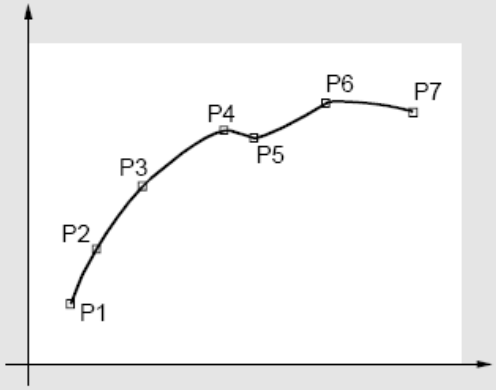
A-SPLINE

A-spline (Akima spline) probíhá přesně skrze uzlové body. Téměř nevytváří nechtěné kmitání, zato ale nemá v uzlových bodech stálé zakřivení.

Akima spline je lokální, tzn. změna jednoho uzlového bodu se projeví pouze na maximálně 6 sousedních uzlových bodech.

Jsou vhodné zejména pro interpolaci digitalizovaných bodů. Pro spline mohou být programovány okrajové podmínky (více informací na následujících stranách). Pro interpolaci se používá polynom 3. stupně.

A-spline (Akima spline)

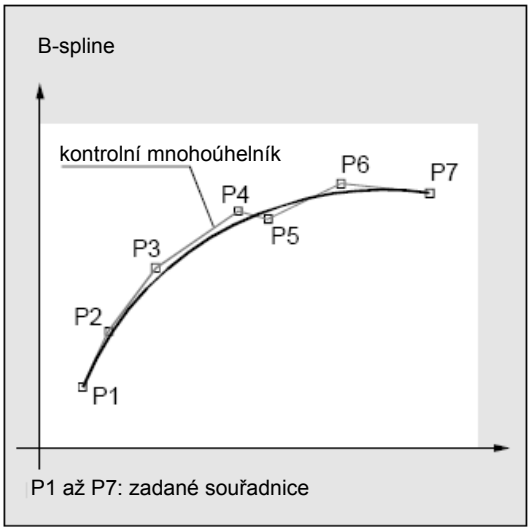


P1 až P7: zadané souřadnice

© Siemens AG 2004, Všechna práva vyhrazena

B-SPLINE

U B-splinu nejsou naprogramované body požadovanými uzlovými body, nýbrž pouze kontrolními body splinu, tzn, že křivka neprobíhá přímo těmito body, ale je jimi „přitahována“.



Spojení bodů přímkami tvoří řídící polygon splinu. B-splin je optimální k popisu drah nástrojů na plochách volného tvaru. Je primárně zamýšlen jako rozhraní pro CAD-systémy. B-spline 3. stupně nevytváří kmitání navzdory přechodům se stálým zakřivením.

Programovatelné okrajové podmínky (více informací na následujících stranách) nemají na B-spline žádný vliv. B-spline je v počátečním a koncovém bodě vždy tangenciální k řídícímu polygonu.

Váha bodu

Pro každý bod průběhu je možno zadat údaj váhy.

Programování:

`PW = n`

Oblast hodnot:

0 <= n <= 3; v krocích po 0.0001

Účinek:

n > 1 Křivka je uzlovým bodem silněji přitahována

n < 1 Křivka je uzlovým bodem méně přitahována

Stupeň splinu:

Standardně je používán polygon 3. stupně. Můžete použít také polygon 2. stupně.

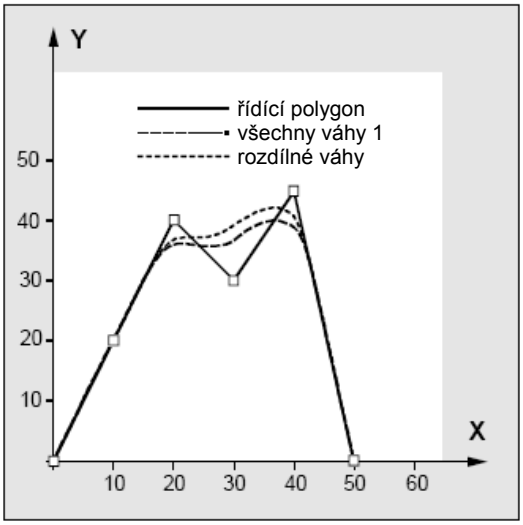
Programování:

`SD = 2`

© Siemens AG 2004, Všechna práva vyhrazena

Vzdálenost uzlů

Vzdálenosti jsou vypočteny dle interní vhodnosti. Řízení může ale také zpracovat zadané vzdálenosti uzlů.



Programování:

`PL` = oblast hodnot jako porovnání dráhy

Příklad pro B-spline:

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

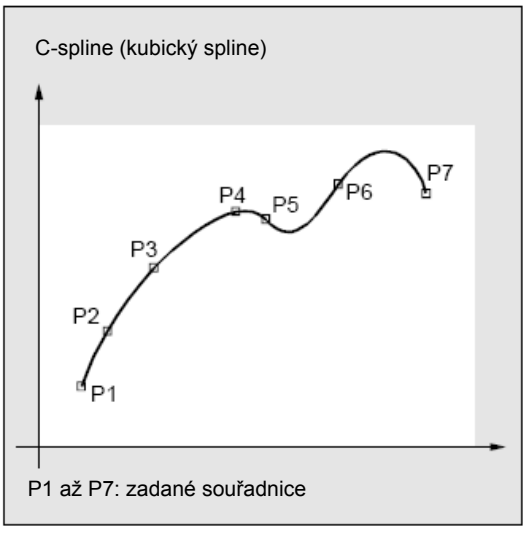
```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

```
N10 G1 X0 Y0 F300 G64
N20 ;odpadá
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

C-SPLINE

Na rozdíl od Akima-splinu má kubický spline (C-spline) v uzlových bodech stálé zakřivení. Má však sklony k neočekávanému kmitání. Může být použit tehdy, jestliže se body nacházejí na analyticky známé křivce.

C-spline není lokální, tzn. změna jednoho uzlového bodu se může projevit v mnoha blocích (s klesající intenzitou).



© Siemens AG 2004, Všechna práva vyhrazena

Okrajové podmínky

Okrajové podmínky platí pouze pro Akima- a kubický spline (A-spline, C-spline).

Prostřednictvím dvou skupin příkazů vždy se třemi příkazy je možné nastavit přechodové chování (začátek resp. konec) těchto křivek splinu.

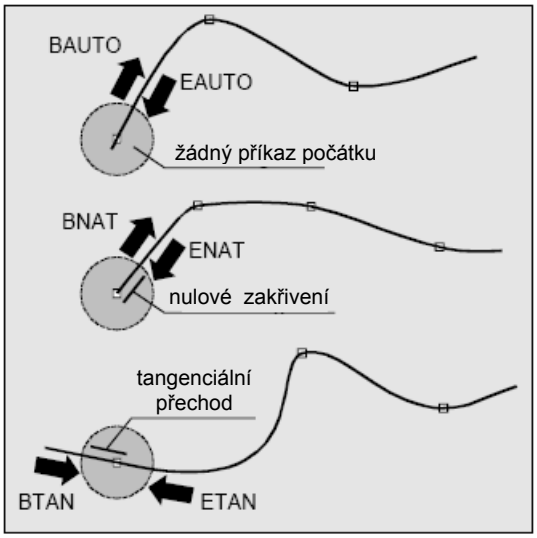
Vysvětlení příkazů

Začátek splinové křivky

BAUTO	Žádný příkaz počátku; počátek vyplyne z polohy prvního bodu
BNAT	Nulové zakřivení
BTAN	Tangenciální přechod k předchozímu bloku (počáteční nastavení)

Konec splinové křivky

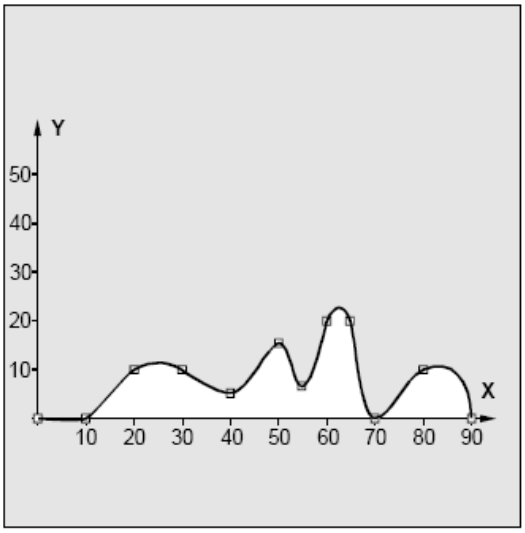
EAUTO	Žádný příkaz počátku; počátek vyplyne z polohy posledního bodu
ENAT	Nulové zakřivení
ETAN	Tangenciální přechod k následujícímu bloku (počáteční nastavení)



© Siemens AG 2004, Všechna práva vyhrazena

Příklad

C-spline, na začátku a na konci nulové zakřivení.



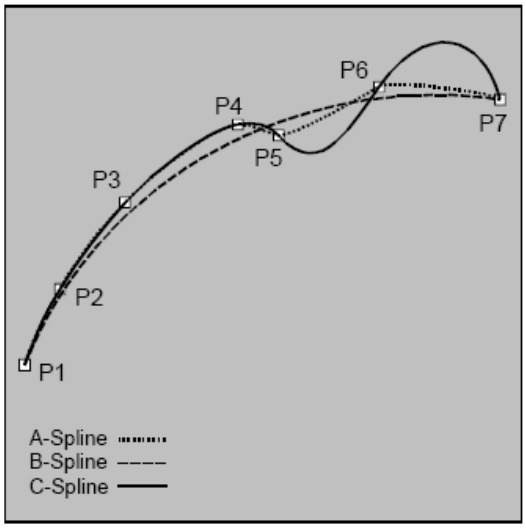

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT                    C-spline, na začátku a na konci je nulové zakřivení
N30 CSPLINE X20 Y10
N40 X30
N50 X40 Y5
N60 X50 Y15
N70 X55 Y7
N80 X60 Y20
N90 X65 Y20
N100 X70 Y0
N110 X80 Y10
N120 X90 Y0
N130 M30
```


© Siemens AG 2004, Všechna práva vyhrazena

Jak působí který spline?

Srovnání tří typů splinů při stejných bodech průběhu:

A-spline (Akima spline)
B-spline (Bézierův spline)
C-spline (kubický spline)



Svazek splinu

Na splinové interpolaci se může podílet až osm dráhových os. Pomocí příkazu SPLINEPATH jsou stanoveny osy, které se na splinu podílejí. Určení probíhá ve zvláštním bloku. Není-li SPLINEPATH explicitně programováno, budou se pohybovat první tři osy kanálu jako svazek splinu.

© Siemens AG 2004, Všechna práva vyhrazena

Programování

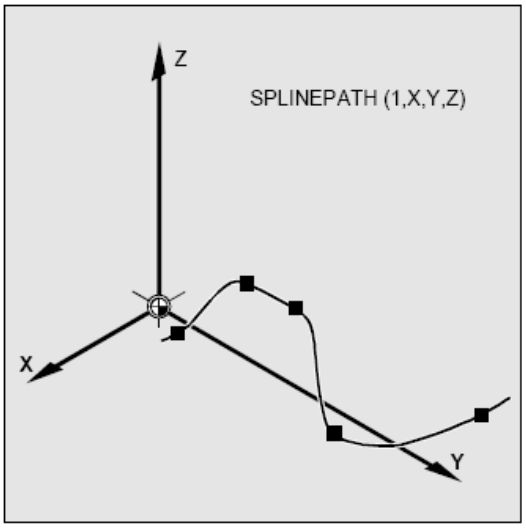
```
SPLINEPATH (n,X,Y,Z,…)
```

Vysvětlení

`SPLINEPATH (n,X,Y,Z,…)`	N = 1, pevná hodnota X,Y,Z,… udání dráhových os

Příklad

Svazek splinu se třemi osami dráhy



`N10 G1 X10 Y20 Z30 A40 B50 F350`	
`N11 SPLINEPATH(1,X,Y,Z)`	Svazek splinu
`N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60`	C-spline
`N14 X30 Y40 Z50 A60 B70` …	Uzlové body
`N100 G1 X… Y…`	Zrušení splinové interpolace

© Siemens AG 2004, Všechna práva vyhrazena

Nastavení pro spliny

G-kódy ASPLINE, BSPLINE a CSPLINE spojují koncové body bloků spliny.

K tomu musí být při předběžném zpracování vypočtena současně celá řada bloků (koncových bodů).

Velikost vyrovnávací paměti pro výpočet činí standardně 10 bloků.

Ne každá informace bloku je koncový bod splinu. Řídící systém však potřebuje z 10 bloků určitý počet koncových bodů bloků splinu.

To jsou pro:

A-spline:	Vždy z 10 bloků musí být nejméně 4 bloky splinu. Bloky komentáře a výpočtu parametrů se do toho nepočítají.
B-spline:	Vždy z 10 bloků musí být nejméně 6 bloků splinu. Bloky komentáře a výpočtu parametrů se do toho nepočítají.
C-spline:	Z každých 10 bloků musí být nejméně tolik splinových, kolik uvádí strojní parametr $MC_CUBIC_SPLINE_BLOCKS+1 (tedy ve standardním případě 9). Ve strojním parametru $MC_CUBIC_SPLINE_BLOCKS (standardní hodnota 8) je zapsán počet bodů, prostřednictvím nichž je vypočten úsek splinu.

Při překročení tolerovatelné hodnoty je hlášen alarm, rovněž v případě, že je-li některá na splinu se podílející osa programována jako polohovací osa.

5.3 Kompresor COMPOF/ON, COMPCURV, COMPCAD (SW 6.2)

Programování

```
COMPON/COMPCURV/COMPCAD
COMPOF
```

Vysvětlení

`COMPON/COMPCURV/COMPCAD`	Kompresor zapnut
`COMPOF`	Kompresor vypnut

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

S použitím G-kódu COMPON jsou přechody mezi bloky stálé jen co do rychlosti, zatímco zrychlení podílejících se os může na přechodech mezi bloky měnit skokově. Toto může vést k vyvolání vibrací stroje.

od SW 4:
S použitím G-kódu COMPCURV budou přechody mezi bloky prováděny se stálým zrychlením. Tím je na přechodech mezi bloky zajištěn hladký průběh jak rychlosti, tak zrychlení všech os.

od SW 6.2:
Pomocí G-kódu COMPCAD může být zvolena další komprese, která provádí optimalizaci ve vztahu **ke kvalitě povrchu a rychlosti**, přičemž prostřednictvím strojních parametrů může být rovněž stanovena přesnost interpolace.
COMPCAD je náročný z hlediska času výpočtu a kapacity paměti. Měl by být používán jen tehdy, jestliže opatření ke zlepšení kvality povrchu nemohly být programem CAD/CAM provedeny předem.
Charakteristiky:

- COMPCAD vytváří z hlediska zrychlení stálé, do sebe přecházející polynomické bloky.
- U sousedních drah vedou odchylky vždy stejným směrem.
- Prostřednictvím nastavovaného parametru $SC_CRIT_SPLINE_ANGLE může být stanoven mezní úhel, od kterého COMPCAD rohy nechává.
- Počet komprimovaných bloků není omezen na 10.
- COMPCAD eliminuje nedostatky na přechodech ploch. Přitom jsou do značné míry dodrženy tolerance, ovšem mezní úhel rohů není zohledněn.
- Může být také použita funkce zaoblení G642.

© Siemens AG 2004, Všechna práva vyhrazena

od SW 6.3

Kompresory COMPON, COMPCURV a COMPCAD jsou tak rozšířeny, že také NC-programy, ve kterých je orientace programována prostřednictvím směrových vektorů, mohou být při dodržení zadatelné tolerance komprimovány.
Funkce „kompresor pro orientace“ se vyskytuje jen v případě, že je k dispozici volba transformace orientace.
Výše pod „podmínky použití“ uvedená omezení jsou uvolněna natolik, že nyní může nastavení hodnot poloh probíhat prostřednictvím přiřazení parametrů.

Obecná forma NC-bloku:

```
N10 G1 X=<...> Y=<...> Z=<...> A=<...>
B=<...> F=<...> ; komentář
```

Polohy os jako parametrické výrazy s <…> parametrický výraz např. `X=R1*(R2+R3)`

Při aktivní transformaci orientace (TRAORI) jsou pro stroje s 5 osami možná kinematicky nezávislá programování pro směr nástroje:

1. Programování směrového vektoru prostřednictvím:
   `A3=< ...> B3=< ... > C3=< ... >`
2. Programování Eulerova úhlu, popř. RPY úhlu prostřednictvím:
   `A2=< ...> B2=< ... > C2=< ... >`

Pohyb orientace je komprimován jen tehdy, když je aktivní interpolace hlavní kružnice, tzn. že změna orientace nástroje probíhá v rovině, která je odvozena z počáteční a konečné orientace.
Interpolace hlavní kružnice je provedena za následujících podmínek:

1: Pro MD 21104: ORI_IPO_WITH_G_CODE = TRUE, když ORIWKS je aktivní a orientace je programována jako vektor (s A3, B3, C3, popř. A2, B2, C2).
2. Pro MD 21104: ORI_IPO_WITH_G_CODE = TRUE, když ORIVECT, popř. ORIPLANE je aktivní. Orientace nástroje může být programována buď jako směrový vektor nebo pomocí poloh kruhových os. Je-li některý z G-kódů ORICONxx nebo ORICURVE aktivní nebo

© Siemens AG 2004, Všechna práva vyhrazena

jsou programovány polynomy pro úhel orientace (PO[PH] a PO[PSI]), nebude provedena interpolace hlavní kružnice, tzn. takové bloky nebudou komprimovány.

U strojů s **6 osami** může být navíc k orientaci nástroje programováno ještě otáčení nástroje. Programování úhlu otáčení probíhá pomocí identifikátoru THETA (`THETA=<...>`).
NC-bloky, ve kterých je navíc programováno otáčení, jsou komprimovatelné jen tehdy, když se úhel otáčení mění lineárně. Tzn. pro úhel otáčení nesmí být programován žádný polynom s PO[THT]=(…).
Obecná forma NC-bloku:

```
N... X=<...> Y=<...> Z=<...> A3=<...>
     B3=<...> C3=<...> THETA=<...> F=<...>
popř.
N... X=<...> Y=<...> Z=<...> A2=<...>
     B2=<...> C2=<...> THETA=<...> F=<...>
```

Je-li orientace nástroje zadána prostřednictvím poloh kruhových os, např. ve formě:

```
N... X=<...> Y=<...> Z=<...> A=<...>
          B=<...> THETA=<...> F=<...>
```

je komprimace provedena dvěma rozdílnými způsoby, v závislosti na tom, zda je nebo není provedena interpolace hlavní kružnice. V případě, že nedojde k interpolaci hlavní kružnice, je komprimovaná změna orientace zobrazena prostřednictvím axiálních polynomů pro kruhové osy.

Přesnost
NC-bloky jsou komprimovatelné jen tehdy, jestliže jsou přípustné odchylky kontury od naprogramované kontury. Maximální odchylka je nastavitelná jako tolerance kompresoru v nastavovaných parametrech. Čím větší jsou dovolené tolerance, tím více bloků může být komprimováno.

Přesnost os
Kompresor vytvoří pro každou osu křivku splinu, která se odchyluje od naprogramovaných koncových bodů maximálně o hodnotu tolerance, nastavenou v MD osy.

© Siemens AG 2004, Všechna práva vyhrazena

Přesnost kontury

Jsou kontrolovány maximální geometrické odchylky kontury (geometrické osy) a orientace nástroje. To probíhá pomocí nastavovaných parametrů pro:

1. maximální toleranci pro konturu.
2. maximální úhlovou odchylku pro orientaci nástroje.
3. maximální úhlovou odchylku pro úhel otáčení nástroje (k dispozici jen u strojů s 6 osami).

Prostřednictvím kanálového parametru MD 20482: COMPRESSOR_MODE mohou být nastaveny zadané tolerance:

0: Přesnost os: axiální tolerance pro všechny osy (geometrické osy a orientační osy).
1: Přesnost kontury: zadání tolerance kontury (1.), tolerance pro orientaci podle axiálních tolerancí (a.).
2: Zadaná hodnota maximální úhlové odchylky pro orientaci nástroje (2.), tolerance pro konturu podle axiálních tolerancí (a.).
3: Zadaná hodnota tolerance kontury (1.) a zadaná hodnota maximální úhlové odchylky pro orientaci nástroje (2.).

Zadání hodnoty maximální úhlové odchylky orientace nástroje je možné jen tehdy, když je aktivní transformace orientace (TRAORI).

Aktivování

Aktivování „Kompresor pro orientace“ probíhá jedním z příkazů: COMPON, COMPCURV (COMPCAD není možný).

Literatura: /FB3/, F2: „Transformace 3 až 5 os“

© Siemens AG 2004, Všechna práva vyhrazena

Výrobce stroje

Pro funkci kompresoru existují tři strojní parametry:

- $MC_COMPRESS_BLOCK_PATH_LIMIT
  Je nastavena maximální délka cesty, do níž platí bloky za komprimovatelné. Delší bloky nebudou komprimovány.
- $MA_COMPRESS_POS_TOL
  Pro každou osu může být nastavena tolerance. Vytvořená splinová křivka se odchyluje maximálně o tuto hodnotu od naprogramovaných koncových bodů. Čím větší jsou tyto hodnoty, tím víc bloků může být komprimováno.
- $C_COMPRESS_VELO_TOL
  Může být zadána maximální přípustná odchylka posuvu vpřed po dráze při aktivním kompresoru ve spojení s FLIN a FCUB.

Zvláštnosti u COMPCAD:

- $MN_MM_EXT_PROG_BUFFER_SIZE má být zvolena velká např. 100 (kB).
- $MC_COMPRESS_BLOCK_PATH_LIMIT musí být zadáno výrazně větší např. 50 (mm).
- $MC_MM_NUM_BLOCKS_IN_PREP musí být zadáno >=60, aby bylo možno zpracovat výrazně více než 10 bodů.
- FLIN a FCUB nemohou být použity.

Doporučeno pro velkou délku bloků a optimální rychlost:

- $MC_MM_MAX_AXISPOLY_PER_BLOCK = 5
  $MC_MM_PATH_VELO_SEGMENTS = 5
  $MC_MM_ARCLENGTH_SEGMENTS = 10.

Systémy CAD/CAM vytvářejí zpravidla lineární bloky, které dodržují prostřednictvím parametrů zadanou přesnost.
To vede u složitých kontur k velkému množství dat a k eventuelně krátkým úsekům dráhy. Tyto krátké úseky dráhy omezují rychlost opracování.
Kompresor má možnost, shrnout určitý počet (max. 10) těchto krátkých úseků dráhy do jednoho úseku dráhy.

© Siemens AG 2004, Všechna práva vyhrazena

Prostřednictvím modálních G-kódů COMPON resp. COMPCURV může být zapnut „kompresor NC-bloků“.
Tato funkce shromažďuje při lineární interpolaci řadu přímkových bloků (počet je omezen na 10) a přibližuje je v rámci tolerance chyb zadatelné strojním parametrem prostřednictvím polynomů 3. stupně (COMPON) resp. 5. stupně (COMPCURV). Místo mnoha malých je pomocí NC zpracován jeden velký blok pohybu.

Podmínky použití:
Tento proces komprese je prováděn jen pro lineární bloky (G1). Je přerušen jakýmkoli jiným NC-pokynem, např. výstupem pomocné funkce, ne však výpočty parametrů.
Jsou komprimovány pouze bloky, které neobsahují více než číslo bloku, G1, adresy os, posuv a komentář. Všechny ostatní bloky budou zpracovány v nezměněné podobě (bez komprese)
Proměnné nesmí být použity.

Příklad COMPON

`N10 COMPON`	Popř. COMPCURV, kompresor zapnut
`N11 G1 X0.37 Y2.9 F600`	G1 musí být před koncovým bodem a posuvem
`N12 X16.87 Y-4.698` `N13 X16.865 Y-4.72` `N14 X16.91 Y-4.799` …	
`N1037 COMPOF` …	Kompresor vypnut

Komprimovány budou všechny bloky, které vyhovují jednoduché syntaxi.
Např.

```
N19 X0.103 Y0. Z0.
N20 X0.102 Y-0.018
N21 X0.097 Y-0.036
N22 X0.089 Y-0.052
N23 X0.078 Y-0.067
```

Komprimovány nebudou např. rozšířené adresy jako C=100 nebo A=AC(100).
Od NC-SW 6.3: bloky pohybu s rozšířenými adresami budou nyní také komprimovány.

© Siemens AG 2004, Všechna práva vyhrazena



Příklad COMPCAD

G00 X30 Y6 Z40	
G1 F10000 G642	
SOFT	
COMPCAD	Zapnuta optimalizace povrchu pomocí kompresoru
STOPFIFO	
N24050 Z32.499	
N24051 X41.365 Z32.500	
N24052 X43.115 Z32.497	
N24053 X43.365 Z32.477	
N24054 X43.556 Z32.449	
N24055 X43.818 Z32.387	
N24056 X44.076 Z32.300	
...	
COMPOF	Kompresor vypnut
G00 Z50	
M30	



Příklad „Kompresor pro orientace“

V následujícím příkladu programu je komprimován kruh, který je přiblížen polygonem. Orientace nástroje se přitom pohybuje synchronně k tomu po plášti kužele. Přestože za sebou následující programované změny orientace probíhají nestále, generuje kompresor plynulý průběh orientace.

```
DEF INT ANZAHL = 60
DEF REAL RADIUS = 20
DEF INT COUNTER
DEF REAL WINKEL
N10 G1 X0 Y0 F5000 G64

$SC_COMPRESS_CONTUR_TOL = 0.05
$SC_COMPRESS_ORI_TOL = 5
TRAORI
COMPCURV
N100 X0 Y0 A3=0 B3=-1 C3=1
N110 FOR COUNTER = 0 TO ANZAHL
N120 WINKEL= 360 * COUNTER /ANZAHL
N130 X=RADIUS*COS(WINKEL)Y=RADIUS*
SIN(WINKEL) A3=SIN(WINKEL)
B3=-COS(WINKEL) C3=1
N140 ENDFOR
...
```

maximální odchylka

kontury	0.05 mm
orientace	5 stupňů

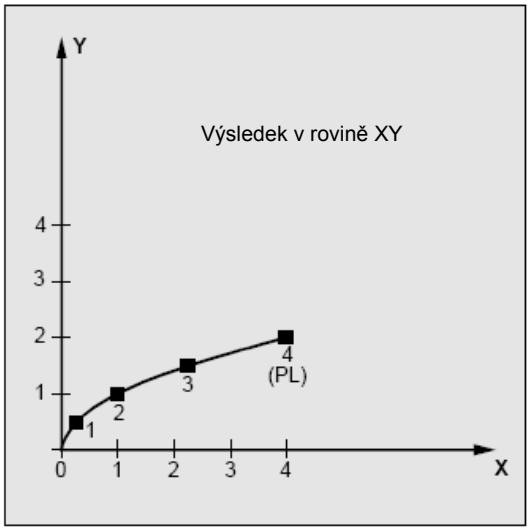
Výsledným pohybem je kruh, který je tvořen polygony.
Orientace se přitom pohybuje po kuželu kolem osy Z
s úhlem výseče 45 stupňů

© Siemens AG 2004, Všechna práva vyhrazena

5.4 Polynomická interpolace – POLY, POLYPATH (od SW 5)



ˇ5ídící systém je schopen provádět posuvy po křivkách (drahách), u kterých se každá zvolená dráhová osa sleduje funkci do SW 5 (polynom, max. 3. stupně), od SW 6 (polynom, max. 5 stupňů).



Obecná forma funkce polynomu je:
$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3$ (do SW 5) nebo
$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$
(od SW 6)
přitom znamenají:
a_n: konstantní koeficienty
p: parametry

Dosazením konkrétních hodnot do koeficientů je možné vytvořit nejrůznější křivky jako funkce přímek, parabol, mocnin.

Při dosazení do koeficientů $a_2 = a_3 = 0$ (do SW 5) nebo $a_2 = a_3 = a_4 = a_5 = 0$ (od SW 6) vznikne např. přímka s:
$f(p) = a_0 + a_1p$

Platí:
a_0 = poloha osy na konci předcházejícího bloku
a_1 = rozdíl mezi polohou osy na konci definiční oblasti (PL) a počáteční polohou

Definice

U polynomické interpolace (POLY) se vlastně nejedná o druh splinové interpolace. V první řadě je myšlena jako rozhraní pro programování externě vytvořených křivek splinu . Přitom mohou být úseky splinu přímo programovány.

Tento druh interpolace ulehčuje NC od výpočtu koeficientů polynomu. Je pak optimálně použitelná, přicházejí-li koeficienty přímo ze systému CAD nebo z procesoru.

© Siemens AG 2004, Všechna práva vyhrazena

Polynomická interpolace se nachází společně s G0, G1, G2, G3, A-splinem, B-splinem, a C-splinem v první G-skupině. Je-li aktivní, není nutné programovat syntaxi polynomu: Osy, které jsou programovány pouze prostřednictvím jejich jména a koncového bodu, se budou do svého koncového bodu pohybovat lineárně. Jsou-li takto programovány všechny osy, chová se řízení jako při G1.

Polynomická interpolace je vypnuta pomocí jiného příkazu G-skupiny (např. G0, G1).

Od SW 5

Vyvolání podprogramu POLYPATH:

Prostřednictvím příkazu POLYPATH může být polynomická interpolace zadána selektivně pro následující skupiny os:

- POLYPATH („AXES“)
  Všechny dráhové osy a přídavné osy.
- POLYPATH („VECT“) Orientační osy (při transformaci orientace).

Standardně budou pro obě skupiny programované polynomy také jako polynom interpolovány.

Příklady:
POLYPATH („VECT“)
Pro polynomickou interpolaci budou vybrány pouze osy orientace, všechny ostatní osy zůstávají lineární.

POLPATH ()
Deaktivuje polynomickou interpolaci pro všechny osy.

Koeficient polynomu

PO hodnota `(PO[] =)` popř. `…=PO(…)` udává všechny koeficienty polynomu pro jednu osu. Způsobem, odpovídajícím stupni polynomu bude uvedeno více hodnot oddělených čárkami.
V rámci jednoho bloku jsou možné různé stupně polynomů pro různé osy.

© Siemens AG 2004, Všechna práva vyhrazena

Okrajové podmínky

Do SW 5

- Polynomy pro geometrické osy popř. pomocné dráhové osy jsou programovatelné jen tehdy, když je aktivní buď G0/G1 nebo POLY. Tzn. při kruhové interpolaci není možné přídavnými osami posouvat prostřednictvím polynomu. Standardně mohou být polynomy programovány prostřednictvím PO[…] jen tehdy, když je aktivní G-kód POLY.

Od SW 5

- Je možné programovat polynomy, **aniž** je aktivní G-kód POLY. V tomto případě však nejsou interpolovány programované polynomy, nýbrž se lineárně najíždí na naprogramovaný koncový bod osy (G1). Programováním POLY je pak aktivována polynomická interpolace.
- Dále může být při aktivním G-kódu POLY prostřednictvím předdefinovaného podprogramu POLYPATH (…) vybráno, které osy mají být interpolovány polynomem.

Od SW 6

- Koeficienty a_4 a a_5 jsou možné jen od SW 6.
- Nová syntaxe polynomu prostřednictvím PO. Dosavadní syntaxe zůstává nadále platná

Příklad platné syntaxe polynomu prostřednictvím PO

Dosavadní syntaxe polynomu zůstává platná	Nová syntaxe polynomu (od SW 6)
PO[identifikátor osy]=(.. , ..)	Identifikátor osy=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[proměnná]=IC(.. , ..)	proměnná=PO IC(.. , ..)

© Siemens AG 2004, Všechna práva vyhrazena

Programování

```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n (do SW 5)
POLYPATH ("AXES", "VECT")(od SW 5)
Rozšíření na polynomy 5.stupně a nová syntax polynomu (od SW 6)
POLY X=PO(xe,a2,a3,a4,a5) Y=PO(ye,b2,b3,b4,b5) Z=PO(ze,c2,c3,c4,c5) PL=n
```

Vysvětlení

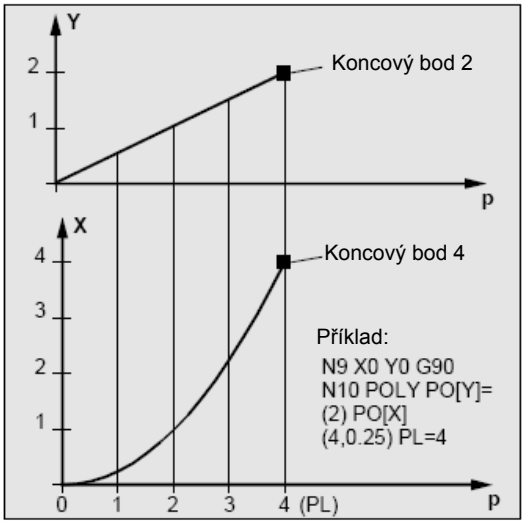
`POLY`	Zapnutí polynomické interpolace pomocí bloku s POLY.
`POLYPATH`	Polynomická interpolace výběrově pro obě skupiny os AXIS nebo VECT
`PO [identifikátor osy/proměnná] = (…,…,…)`	Koncové body a koeficienty polynomu
`X, Y, Z`	Identifikátory osy
x_e, y_e, z_e	Údaj koncové polohy pro příslušnou osu; rozsah hodnot jako údaj dráhy
a_2, a_3, a_4, a_5	Koeficienty a_2, a_3, a_4 a a_5 jsou zapsány se svou hodnotou; rozsah hodnot jako údaj dráhy. Příslušný poslední koeficient může být vynechán, má-li hodnotu nula.
`PL`	Délka intervalu parametrů, ve kterém jsou definovány polynomy (definiční oblast funkce f(p)). Interval začíná vždy O, p může mít hodnoty od 0 do PL. Teoretický rozsah hodnot pro PL: 0,0001 … 99 999,9999. Hodnota PL platí pro blok, ve kterém se nachází. Není-li PL hodnota programována, platí PL = 1.

Příklad

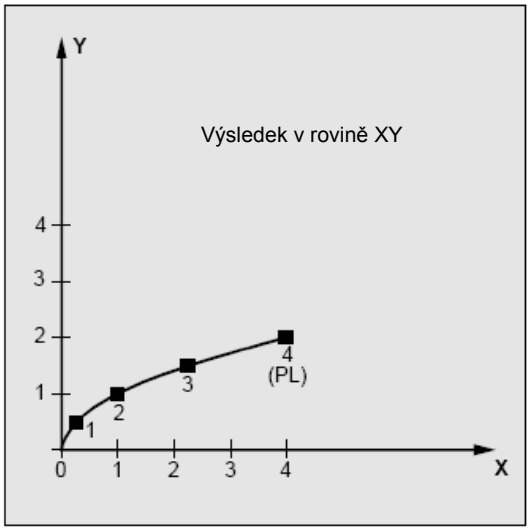
`N10 G1 X… Y… Z… F600`	
`N11 POLY PO[X]=(1,2.5,0.7) ->` `-> PO[Y]=(0.3,1,3.2) PL=1.5`	Polynomická interpolace zapnuta
`N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7)` `PL=3` … `N20 M8 H126 …`	
`N25 X70 PO[Y]=(9.3,1,7.67) PL=5`	Smíšené údaje pro osy
`N27 PO[X]=(10,2.5) PO[Y]=(2.3)`	Není programována hodnota PL, platí PL=1
`N30 G1 X… Y… Z.`	Polynomická interpolace vypnuta
…	

© Siemens AG 2004, Všechna práva vyhrazena

Příklad pro křivku v rovině X/Y




```
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4
```



Zvláštnost: polynom jmenovatele

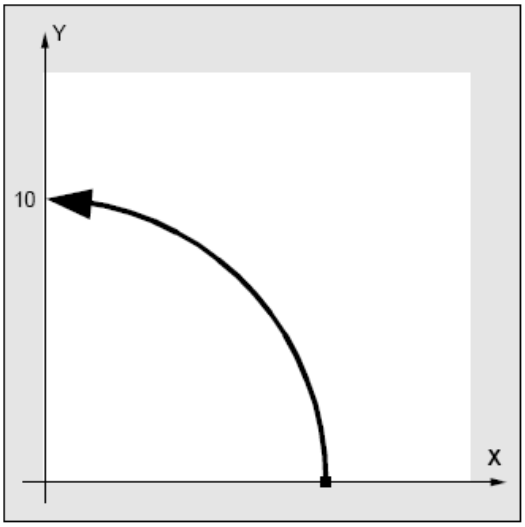
Pro geometrické osy může být prostřednictvím PO[] = (…) bez zadání názvu osy programován společný polynom jmenovatele, tzn. pohyb geometrických os je interpolován jako kvocient dvou polynomů.

Tak se dají exaktně zobrazit např. kuželosečky (kruh, elipsa, parabola, hyperbola).

Příklad

`POLY G90 X10 Y0 F100`	Geometrické osy se pohybují lineárně do polohy X10, Y0
`PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)`	Geometrické osy se pohybují ve čtvrtkruhu do polohy X0, Y10

Konstantní koeficient (a_0) polynomu jmenovatele je stále předpokládán s hodnotou 1, zadaný koncový bod je nezávislý na G90/G91.



Z výše uvedeného příkladu vyplývá následující výsledek:

$X(p)=10(1-p^2)$ a $Y(p)=20p/(1+p^2)$
S $0<=p<=1$

Na základě programovaných počátečních bodů, koncových bodů, koeficientu a_2 a PL=1 vzniknou následující mezihodnoty:

Čitatel (X) $=10+0*p-10p_2$
Čitatel (Y) $=0+20*p+0*p_2$
Jmenovatel $= 1+2*p+1*p_2$

© Siemens AG 2004, Všechna práva vyhrazena

Při zapnuté polynomické interpolaci bude programování polynomu jmenovatele s nulovými body uvnitř intervalu [0,PL] odmítnuto s alarmem. Na pohyb pomocných os nemá polynom jmenovatele žádný vliv.

Další upozornění

Korekce rádiusu nástroje je při polynomické interpolaci spustitelná prostřednictvím G41, G42 a použitelná jako pro interpolaci přímky nebo kruhu.

5.5 Nastavitelná referenční dráha, SPATH, UPATH (od SW 4.3)

Programování

SPATH	Referenční dráha pro osy FGROUP je délka oblouku
UPATH	Referenční dráha pro osy FGROUP je parametr křivky

Úvod

Během polynomické interpolace mohou být uživatelem žádány dva různé vztahy mezi rychlost určujícími osami FGROUP a ostatními dráhovými osami: posledně jmenované by měly být vedeny

- buď synchronně vůči dráze os FGROUP
- nebo synchronně vůči parametru křivky.

Doposud byla realizována jenom první varianta vedení pohybu, od stavu SW 4.3 může být prostřednictvím G-kódu (SPATH, UPATH) vybráno a programováno žádoucí chování.

Funkce

Během polynomické interpolace – a tím je nadále vždy chápána polynomická interpolace v užším smyslu (POLY), všechny druhy splinové interpolace (ASPLINE, BSPLINE, CSPLINE) a lineární interpolace s kompresorem (COMPON, COMPCURV) – jsou polohy všech dráhových os dány polynomy $p_i(U)$. Parametr křivky U se přitom pohybuje uvnitř jednoho NC-bloku od 0 do 1, je tedy normován.

© Siemens AG 2004, Všechna práva vyhrazena

Prostřednictvím příkazu jazyka FGROUP mohou být v rámci dráhových os vybrány ty osy, na které se má vztahovat programovaný posuv po dráze. Interpolace s konstantní rychlostí na dráze S těchto os znamená však během polynomické interpolace zpravidla nekonstantní změnu parametru křivky U.
Pro osy, které nejsou obsaženy v FGROUP, jsou proto dvě možnosti, jak sledovat dráhu:

1. buď jsou synchronní s drahou S (SPATH)
2. nebo jsou synchronní s parametrem křivky U os FGROUP (UPATH).

Oba druhy dráhové interpolace jsou využívány v různých aplikacích a mohou být přepínány prostřednictvím G-kódů SPATH a UPATH.

UPATH a SPATH určují také souvislost polynomu F-slova (FPOLY, FCUB, FLIN) s pohybem po dráze.

Příklad

V následujícím příkladu je čtverec o délce hrany 20 mm zaoblen prostřednictvím G643.

```
Maximální odchylky od přesné kontury
jsou přitom stanoveny pro každou osu
strojními parametry
MD 33100: COMPRESS_POS_TOL[...]
```

N10 G1 X… Y… Z… F500	
N20 G643	Zaoblení uvnitř bloku prostřednictvím G643
N30 XO Y0	
N40 X20 Y0	délka hrany pro osy 20 mm
N50 X20 Y20	
N60 X0 Y20	
N70 X0 Y0	
N100 M30	

© Siemens AG 2004, Všechna práva vyhrazena

Okrajové podmínky

Nastavená referenční dráha nemá žádný význam

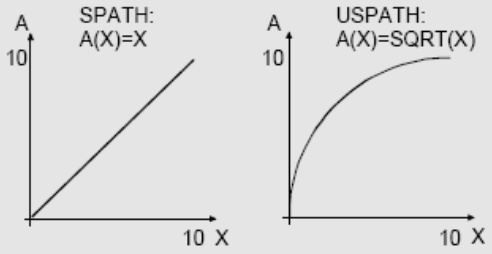
- při lineární interpolaci a kruhové interpolaci
- v závitových blocích
- tehdy, když všechny osy dráhy jsou obsaženy v FGROUP.

Aktivování

Referenční dráha pro osy, které nejsou obsaženy v FGROUP, je nastavena prostřednictvím obou příkazů SPATH a UPATH jazyka obsažených ve 45. skupině G-kódů. Příkazy mají modální platnost. Při aktivním SPATH jsou osy vedeny synchronně s dráhou, při aktivním UPATH synchronně s parametrem křivky.

Příklad programování

Následující příklad programování ilustruje rozdíl mezi oběma druhy vedení pohybu. V obou případech je aktivní přednastavení FGROUP(X,Y,Z).



Rozdílné geometrické vztahy mezi osami při SPATH a UPATH

```
N10 G1 X0 A0 F1000 SPATH
N20 POLY PO[X]=(10, 10) A10
resp.
N10 G1 X0 F1000 UPATH
N20 POLY PO[X]=(10, 10) A10
```

V bloku N20 závisí dráha S FGROUP-os na čtverci parametru křivky. Proto vznikají podél dráhy X různé polohy synchronní osy A podle toho, zda je aktivní SPATH nebo UPATH.

© Siemens AG 2004, Všechna práva vyhrazena

Chování řízení při Power ON, změně provozního režimu, resetu, vyhledávání bloku, funkci Repos

Po resetu je v platnosti určitý G-kód (45. skupina G-kódů) určený prostřednictvím MD 20150: GCODE_RESET_VALUES [44].

Hodnota základního nastavení pro druh zaoblení je stanovena prostřednictvím MD 20150: GCODE_RESET_VALUES [9] (10. skupina G-kódů).

Strojní/volitelné parametry

Po resetu aktivní hodnota skupiny G-kódů je určena prostřednictvím strojního parametru MD 20150: GCODE_RESET_VALUES [44]. Aby byla zachována kompatibilita se stávajícími konfiguracemi, je zde jako standardní hodnota předem nastaveno SPATH.

Hodnota základního nastavení pro druh zaoblení je stanovena prostřednictvím MD 20150: GCODE_RESET_VALUES [9] (10. skupina G-kódů).

Strojní parametry osy MD 33100: COMPRESS_POS_TOL mají od SW 4.3 rozšířený význam: obsahují tolerance pro funkci kompresoru a pro zaoblení prostřednictvím G642.

© Siemens AG 2004, Všechna práva vyhrazena

5.6 Měření spínací sondou, MEAS, MEAW

Programování

MEAS=±1	G… X… Y… Z…	(+1/+2 měření s vymazáním zbytkové dráhy a náběžnou hranou)
MEAS=±2	G… X… Y… Z…	(-1/-2 měření s vymazáním zbytkové dráhy a sestupnou hranou)
MEAW=±1	G… X… Y… Z…	(+1/+2 měření bez vymazání zbytkové dráhy a náběžnou hranou)
MEAW=±2	G… X… Y… Z…	(-1/-2 měření bez vymazání zbytkové dráhy a sestupnou hranou)

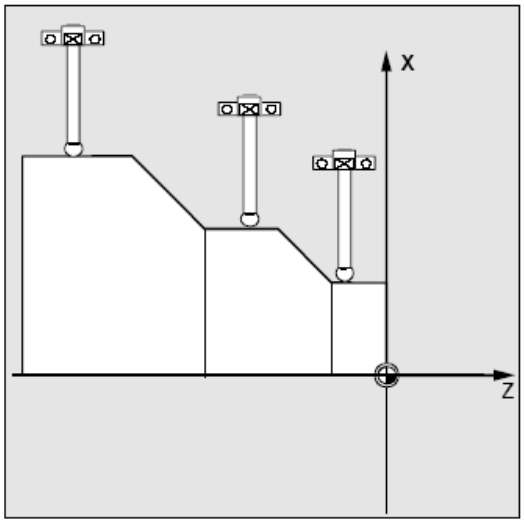
Vysvětlení příkazů

MEAS=±1	Měření sondou 1 na měřicím vstupu 1
MEAS=±2*	Měření sondou 2 na měřicím vstupu 2
MEAW=±1	Měření sondou 1 na měřicím vstupu 1
MEAW=±2*	Měření sondou 2 na měřicím vstupu 2

*podle stupně vybavení max. 2 vstupy

Postup

Pro všechny osy naprogramované v NC-bloku jsou snímány polohy spínací hranou signálu sondy a pro každou osu zapsány do odpovídající buňky paměti. Existují maximálně 2 měřicí sondy.



Výsledek měření

Výsledek měření je pro tyto osy k dispozici pod následujícími proměnnými:

- v souřadném systému stroje pod $AA_MM[osa]
- v souřadném systému obrobku pod $AA_MW[osa]

Při čtení těchto proměnných nedojde k internímu zastavení předběžného zpracování. Prostřednictvím STOPRE musí být v NC-programu na vhodném místě programováno zastavení předběžného zpracování, jinak budou čteny chybné hodnoty.

© Siemens AG 2004, Všechna práva vyhrazena

Status měření

Je-li v programu nutné vyhodnocení, zda sonda sepnula nebo ne, může být vyhledána stavová proměnná `$AC_MEA[n]` (n= číslo sondy):

0 měření neprovedeno

1 měření úspěšně ukončeno (sonda sepnula)



Je-li sonda během zpracování programu vychýlena, je za proměnná nastavena na 1. Při spuštění měřicího bloku je proměnná automaticky nastavena na počáteční stav sondy.

Programování měřicích bloků, MEAS, MEAW

Prostřednictvím příkazu MEAS a druhem interpolace jsou zaujaty skutečné polohy na obrobku a přitom jsou převzaty naměřené hodnoty. Zbytková dráha mezi skutečnou a zadanou polohou je vymazána.

Pro speciální měření, při nichž má být v každém případě zaujata programovaná poloha, je použita funkce MEAW.

MEAS a MEAW jsou programovány v bloku s příkazy pohybu. Posuvy a druhy interpolace (G0, G1, …) je třeba přizpůsobit příslušnému problému měření; rovněž počet os.

Příklad:

```
N10 MEAS=1 G1 F1000 X100 Y730 Z40
```

Blok měření s měřicí sondou prvního měřicího vstupu a přímkovou interpolací. Zastavení předběžného zpracování je vytvořeno automaticky.

© Siemens AG 2004, Všechna práva vyhrazena

Snímání naměřených hodnot

Jsou zachyceny polohy všech dráhových a polohovacích os bloku (maximální počet os podle konfigurace řídícího systému).
U příkazu MEAS je pohyb po sepnutí sondy definovaným způsobem zbržděn.

Poznámka

Je-li v bloku měření programována geometrická osa, jsou uloženy hodnoty pro všechny aktuální geometrické osy.
Je-li v bloku měření programována osa podílející se na transformaci, jsou uloženy hodnoty všech os, podílejících se na této transformaci.



Další upozornění

Funkce MEAS a MEAW mají blokovou platnost.

© Siemens AG 2004, Všechna práva vyhrazena

5.7 Rozšířené funkce měření MEASA, MEAWA, MEAC (od SW 4, volitelný doplněk)

Programování

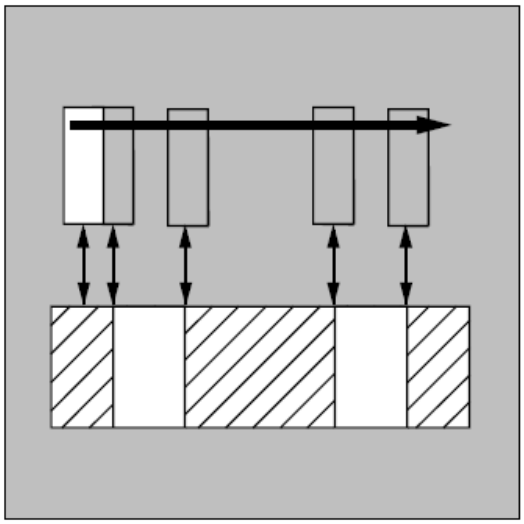
`MEASA[osa]=(Modus, TE1,…, TE 4)`	Měření s vymazáním zbytkové dráhy
`MEAWA [osa]=(Modus, TE 1,…, TE 4)`	Měření bez vymazání zbytkové dráhy
`MEAC [osa]=(Modus, měřicí paměť, TE 1,...TE4)`	Kontinuální měření bez vymazání zbytkové dráhy

Vysvětlení

Osa	Jméno osy kanálu, použité pro měření
Modus	Dvoumístný údaj k režimu provozu; sestávající z těchto částí: **Režim měření** (dekáda jednotek) 0 Přerušení měřicí úlohy 1 Režim **1**: až 4 různé **současně** aktivovatelné spouštěcí události 2 Režim **2**: až 4 **postupně** aktivovatelné spouštěcí události 3 Režim **3**: až 4 **postupně** aktivovatelné spouštěcí události, ovšem žádná kontrola spouštěcí události 1 při spouštění (alarmy 21700 a 21703 budou potlačeny) Upozornění: Režim 3 při MEAC není možný **Měřicí systém** (dekáda desítek) 0 nebo žádný údaj: aktivní měřicí systém 1 Měřicí systém 1 2 Měřicí systém 2 3 Oba měřicí systémy
TE 1…4	**Spouštěcí událost** 1 náběžná hrana, měřicí sonda 1 -1 sestupná hrana, měřicí sonda 1 2 náběžná hrana, měřicí sonda 2 -2 sestupná hrana, měřicí sonda 2
Měřicí paměť	Číslo FIFO (cyklické paměti)

© Siemens AG 2004, Všechna práva vyhrazena

Funkce



Od SW 4 je k dispozici axiální měření.
Při těchto měřeních může měření probíhat axiálně s více měřicími sondami a více měřicími systémy.

Při MEAWA, MEASA jsou pro příslušnou naprogramovanou osu zachyceny až čtyři naměřené hodnoty na jedno měření a uloženy v systémových proměnných v závislosti na spouštěcí události.
MEASA a MEAWA mají blokovou působnost.

Kontinuální měření mohou být provedena prostřednictvím MEAC. V tomto případě jsou výsledky měření ukládány do proměnných FIFO. Také pro MEAC jsou na jedno měření možné maximálně čtyři naměřené hodnoty.

Postup

Programování se může uskutečnit ve výrobním programu nebo ze synchronní akce (kapitola 10). Na jednu osu může být přitom ve stejný okamžik aktivní jen jedno měření.

Další upozornění

- Posuv je třeba přizpůsobit příslušnému problému měření.
- Při MEASA a MEAWA mohou být zaručeny správné výsledky jen při posuvech, při kterých nenastane více než jedna stejna a ne více než 4 různé spouštěcí události na takt regulátoru polohy.
- Při kontinuálním měření prostřednictvím MEAC nesmí být poměr mezi taktem interpolace a taktem regulátoru polohy.větší než 8 : 1.

© Siemens AG 2004, Všechna práva vyhrazena

Spouštěcí události

Spouštěcí událost se skládá z čísla měřicí sondy a spouštěcího kritéria (náběžná nebo sestupná hrana) měřicího signálu.

V rámci každého měření mohou být vždy zpracovány až 4 spouštěcí události spínané měřicí sondy, tedy až dvě měřicí sondy po dvou měřicích hranách.
Pořadí zpracování a maximální počet spouštěcích událostí jsou přitom závislé na zvoleném režimu.

Stejná spouštěcí událost může být v jednom měření programována jen jednou (platí jen pro režim 1)!

Režim provozu

Prostřednictvím první číslice režimu je zvolen požadovaný měřicí systém. Je-li k dispozici jen jeden měřicí systém, avšak je programován druhý, je automaticky použit ten, který je k dispozici.

Prostřednictvím druhé číslice, režimu měření, je proces měření přizpůsoben možnostem příslušného řídícího systému:

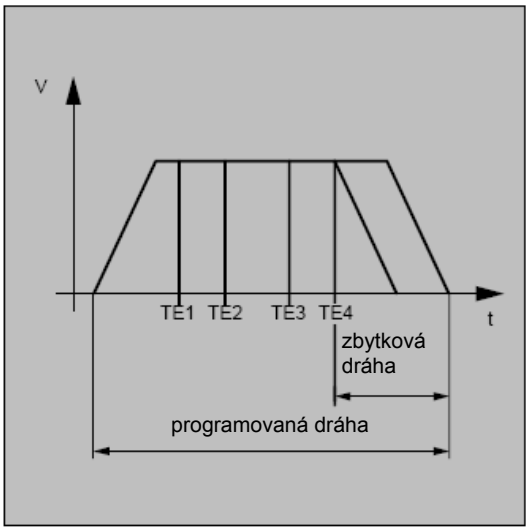
- **Režim 1:** Vyhodnocení spouštěcích událostí probíhá v **časové** posloupnosti jejich výskytu. V tomto režimu je při použití konstrukčních prvků se šesti osami programovatelná jen jedna spouštěcí událost, resp. se při zadání více spouštěcích událostí automaticky přechází na režim 2 (bez hlášení).
- **Modus 2:** Vyhodnocení spouštěcích událostí probíhá v naprogramované posloupnosti.
- **Modus 3:** Vyhodnocení spouštěcích událostí probíhá v naprogramované posloupnosti, ovšem bez kontroly spouštěcí události 1 při spuštění systému.

Další upozornění

Při použití 2 měřicích systémů je možné programovat jen dva spouštěcí děje.

© Siemens AG 2004, Všechna práva vyhrazena

Měření s a bez vymazání zbytkové dráhy



Při programování MEASA je provedeno vymazání zbytkové dráhy teprve po zaznamenání všech požadovaných hodnot měření.

Pro speciální měření, při kterých má být programovaná poloha zaujata v každém případě, je použito MEWA.

MEASA a MEAWA mohou být programovány v jednom bloku.

Je-li v jednom bloku programováno MEASA/MEAWA s MEAS/MEAW, proběhne hlášení chyby.

- MEASA není programovatelné u synchronních akcí.
  Jako náhrada může být programováno MEAWA a vymazání zbytkové dráhy.
- Je-li proces měření spuštěn prostřednictvím MEAWA ze synchronních akcí, jsou hodnoty měření k dispozici jen v souřadném systému stroje.

Výsledky měření pro MEASA, MEAWA

Výsledky měření jsou k dispozici pod následujícími systémovými proměnnými:

- V souřadném systému stroje:

`$AA_MM1 [osa]`	Změřená hodnota z naprogramovaného měřicího systému při spouštěcí události 1
...	...
`$AA_MM4 [osa]`	Změřená hodnota z naprogramovaného měřicího systému při spouštěcí události 4

- V souřadném systému obrobku:

`$AA_WM1 [osa]`	Změřená hodnota z naprogramovaného měřicího systému při spouštěcí události 1
...	...
`$AA_WM4 [osa]`	Změřená hodnota z naprogramovaného měřicího systému při spouštěcí události 4

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Při čtení těchto proměnných nedojde k internímu zastavení předběžného zpracování. Prostřednictvím STOPRE (kapitola 15.1) musí být na vhodném místě programováno zastavení předběžného zpracování. Jinak budou načteny chybné hodnoty.

Má-li být axiální měření spuštěno pro geometrickou osu, musí být stejné měření explicitně programováno pro všechny ostatní geometrické osy. Totéž platí pro osy, které se účastní transformace.

Příklad:

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1)
MEASA[X]=(1,1) G0 Z100;
```

nebo

```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Měření se 2 měřicími systémy

Je-li prováděno měření se dvěma měřicími systémy, je zaznamenána každá z obou možných spouštěcích událostí obou měřicích systémů příslušné osy. Dosazení za rezervované proměnné je pak dáno:

$AA_MM1[osa]	resp.	$AA_MW1[osa]	Změřená hodnota z měřicího systému 1 při spouštěcí události 1
$AA_MM2[osa]	resp.	$AA_MW2[osa]	Změřená hodnota z měřicího systému 2 při spouštěcí události 1
$AA_MM3[osa]	resp.	$AA_MW3[osa]	Změřená hodnota z měřicího systému 1 při spouštěcí události 2
$AA_MM4[osa]	resp.	$AA_MW4[osa]	Změřená hodnota z měřicího systému 2 při spouštěcí události 2

Status měřicí sondy čitelný prostřednictvím

$A_PROBE[n]

n = měřicí sonda
1 == měřicí sonda vychýlena
0 == měřicí sonda nevychýlena

© Siemens AG 2004, Všechna práva vyhrazena

Status měření při MEASA, MEAWA
Je-li v programu nutné vyhodnocení, může být vyhledán status měření prostřednictvím **$AC_MEA**[n], kde n = číslo měřicí sondy. Jakmile proběhly všechny spouštěcí události měřicích sondy „n“ naprogramované v jednom bloku, je výsledkem této proměnné hodnota 1. V opačném případě je hodnota 0.

Je-li měření spuštěno ze synchronní akce, není $AC_MEA již aktualizováno. V tomto případě je třeba vyhledat nové stavové signály PLC DB(31-38) DBB62 Bit3 popř. rovnocennou proměnnou $AA_MEAACT[“osa“].
Význam: $AA_MEAACT==1: měření aktivní
$AA_MEAACT==0: měření není aktivní
Literatura:/PGA/ kapitola 10, Synchronní akce

Další upozornění

- Obsah FIFO může být z cyklické paměti čten jen jednou. Pro vícenásobné použití hodnot měření musí být tato prozatímně uložena v uživatelských datech.
- Překročí-li počet hodnot měření pro paměť FIFO maximální počet stanovený ve strojních parametrech, bude měření automaticky ukončeno.
- Nekonečné měření se dá realizovat prostřednictvím cyklického čtení hodnot měření. Čtení musí přitom proběhnout s nejméně stejnou četností jako vstup nových hodnot měření.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Měření s mazáním zbytkové dráhy v režimu 1
(Vyhodnocení v časové posloupnosti)
a) s 1 měřicím systémem

…	
N100 MEASA[X] = (1,1,-1) G01 X100 F100	Měření v módu 1 s aktivním měřicím systémem. Čekání na měřicí signál se náběžnou/sestupnou hranou od měřicí sondy 1 na dráze posuvu po X = 100.
N110 STOPRE	Zastavení předběžného zpracování
N120 IF $AC_MEA[1] == FALSE gotof ENDE	Kontrola úspěšnosti měření.
N130 R10 = $AA_MM1[X]	Uložení hodnoty z měření patřící k první programované spouštěcí události (náběžná hrana).
N140 R11 = $AA_MM2[X]	Uložení hodnoty z měření patřící ke druhé programované spouštěcí události (sestupná hrana).
N150 ENDE:	

Příklad programování

b) se 2 měřicími systémy

…	
N200 MEASA[X] = (31,1-1) G01 X100 F100	Měření v režimu 1 se dvěma měřicími systémy. Čekání na měřicí signál s náběžnou / sestupnou hranou od měřicí sondy 1 na dráze posuvu po X = 100.
N210 STOPRE	Zastavení předběžného zpracování
N220 IF $AC_MEA[1] == FALSE gotof ENDE	Kontrola úspěšnosti měření.
N230 R10 = $AA_MM1[X]	Uložení změřené hodnoty z měřicího systému 1 při náběžné hraně.
N240 R11 = $AA_MM2[X]	Uložení změřené hodnoty z měřicího systému 2 při náběžné hraně.
N250 R12 = $AA_MM3[X]	Uložení změřené hodnoty z měřicího systému 1 při sestupné hraně.
N260 R13 = $AA_MM4[X]	Uložení změřené hodnoty z měřicího systému 3 při sestupné hraně.
N270 ENDE:	

© Siemens AG 2004, Všechna práva vyhrazena

Měření s vymazáním zbytkové dráhy v režimu 2

(vyhodnocení v naprogramované posloupnosti)

...	
N100 MEASA[X] = (2,1,-1,2,-2) G01 X100 F100	Měření v režimu 2 s aktivním měřicím systémem. Čekání na měřicí signál v posloupnosti náběžná hrana měřicí sondy 1, sestupná hrana měřicí sondy 1, náběžná hrana měřicí sondy 2, sestupná hrana měřicí sondy 2, na dráze posuvu po X = 100.
N110 STOPRE	Zastavení předběžného zpracování
N120 IF $AC_MEA[1] == FALSE gotof MESSTASTER2	Kontrola úspěšnosti měření prostřednictvím měřicí sondy 1.
N130 R10 = $AA_MM1[X]	Uložení změřené hodnoty patřící k první naprogramované spouštěcí události (náběžná hrana z měřicí sondy 1).
N140 R11 = $AA_MM2[X]	Uložení změřené hodnoty patřící ke druhé naprogramované spouštěcí události (náběžná hrana z měřicí sondy 1).
N150 MESSTASTER2	
N160 IF $AC_MEA[2] == FALSE gotof ENDE	Kontrola úspěšnosti měření prostřednictvím měřicího sondy 2.
N170 R12 = $AA_MM3[X]	Uložení změřené hodnoty patřící ke třetí naprogramované spouštěcí události (náběžná hrana z měřicí sondy 2).
N180 R13 = $AA_MM4[X]	Uložení změřené hodnoty patřící ke čtvrté naprogramované spouštěcí události (náběžná hrana z měřicí sondy 2).
N190 ENDE:	

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování

Kontinuální měření v režimu 1
(vyhodnocení v časové posloupnosti)
Měření až 100 hodnot měření

…	
N110 DEF REAL MESSWERT[100]	
N120 DEF INT smyčka= 0	
N130 MEAC [X] = (1,1,-1) G01 X1000 F100	Měření v režimu 1 s aktivním měřicím systémem, uložením změřených hodnot do $AC_FIFO1, čekání na sestupnou hranu měřicího signálu 1 na dráze posuvu po X = 1000.
N135 STOPRE	
N140 MEAC[X] = (0)	Přerušení měření po dosažení polohy osy.
N150 R1 = $AC_FIFO1[4]	Uložení počtu zaznamenaných změřených hodnot v parametru R1.
N160 FOR smyčka= 0 TO R1-1	
N170 MESSWERT[smyčka] = $AC_FIFO1[0]	Načtení a uložení změřené hodnoty ze $AC_FIFO1
N180 ENDFOR	

Měření s vymazáním zbytkové dráhy po 10 změřených hodnotách

…	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG (x)	Vymazat zbytkovou dráhu
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC[X]=(0)	
N40 R1=$AC_FIFO1[4]	Počet změřených hodnot
...	

© Siemens AG 2004, Všechna práva vyhrazena

Následující chyby v programování jsou rozpoznány a oznámeny chybovým hlášením:

- Příkaz MEASA/MEAWA je programován společně s příkazem MEAS/MEAW v jednom bloku
  Příklad:
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```

- Příkaz MEASA/MEAWA s počtem parametrů <2 nebo >5
  Příklad:
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```

- Příkaz MEASA/MEAWA s jinou spouštěcí událostí než 1/-1/2/-2
  Příklad:
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```

- Příkazy MEASA/MEAWA s nesprávným režimem
  Příklad:
  ```
  N01 MEASA[B]=(4,1) B100
  ```

- Příkaz MEASA/MEAWA s dvakrát programovanou spouštěcí událostí
  Příklad:
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```

- Příkaz MEASA/MEAWA a chybějící geometrická osa
  Příklad:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50
      Z50 F100
  ```
  geometrická osa X/Y/Z

- Nejednotné měření u geometrických os
  Příklad:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1)
      MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

© Siemens AG 2004, Všechna práva vyhrazena

5.8 Speciální funkce pro OEM-uživatele, G810 až G829



OEM-adresy
Význam OEM-adres určuje OEM-uživatel.
Funkce je aktivována prostřednictvím cyklů překladače. Rezervováno je 5 OEM-adres.
Identifikátory adres jsou nastavitelné.
OEM-adresy jsou přípustné v každém bloku.

OEMIPO1, OEMIPO2
OEM-uživatel může definovat dva názvy G-funkcí OEMIPO1, OEMIPO2 navíc. Tato funkce je aktivována prostřednictvím cyklů překladače a je rezervována pro OEM-uživatele.

Rezervované G-skupiny
Skupina 31 s G810 až G819
Skupina 32 s G820 až G828

Pro OEM-uživatele jsou rezervovány dvě G-skupiny vždy s 10 OEM G-funkcemi.
Tak se dají OEM-uživatelem vyžádané funkce transportovat k externímu použití.

Funkce a podprogramy
Navíc mohou OEM-uživatelé vytvářet také předdefinované funkce a podprogramy s předáváním parametrů.

5.9 Zpomalení posuvu v rozích, G62, G621 (od SW 6.1)



Programování

```
FENDNORM
G62 G41
G621
```


© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení příkazů

FENDNORM	Automatické zpomalení v rozích vypnuto
G62	Zpomalení ve vnitřních rozích při aktivní korekci radiusu nástroje
G621	Zpomalení ve všech rozích při aktivní korekci radiusu nástroje

Funkce

Při automatickém zpomalení v rozích je posuv zvonovitě zpomalen krátce před dotyčným rohem. Kromě toho mohou být pomocí nastavovaných parametrů zadány přídavky rozměru pro zpracování relevantního chování nástroje. Toto jsou:

- Začátek a konec zpomalení posuvu
- Override, kterým je posuv redukován
- Rozpoznání relevantního rohu

Jako relevantní rohy jsou zohledněny ty rohy, jejichž vnitřní úhel je menší, než jsou pomocí nastavovaných paremtrů zadané parametry rohu.

G62 se projevuje jen ve vnitřních rozích s

- Aktivní korekcí rádiusu nástroje G41, G42 a
- Aktivním režimem řízení pohybu po dráze G64, G641

Na odpovídající roh bude najíždět s redukovaným posuvem, který vyplývá z:

F∗ (Override pro redukci posuvu) ∗ Override posuvu

Maximální možné snížení hodnoty posuvu je dosaženo přesně tehdy, když nástroj, vztaženo ke dráze středového bodu, má učinit změnu směru v dotyčném rohu.

G621 se chová analogicky k G62, ale v každém rohu os definovaných příkazem FGROUP.
Příkaz FENDNORM s předdefinovanou hodnotou funkci automatického snižování hodnoty posuvu v rozích vypíná.

Další upozornění

Tato funkce nepatří ke standardnímu rozsahu funkcí systému SINUMERIK a musí být pro příslušné verze softwaru uvolněna.
Literatura: /FBA/, Popis funkcí dialektu ISO

© Siemens AG 2004, Všechna práva vyhrazena

5.10 Programovatelné kritérium konce pohybu (od SW 5.1)

Programování

```
FINEA [<osa>]
COARSEA [<osa>]
IPOENDA [<osa>]
IPOBRKA(<osa>[, [<hodnota v procentech>]])        Je možné vícenásobné zadání
ADISPOSA(<osa>, [<Int>][,[<Real>]])               Je možné vícenásobné zadání
```

Vysvětlení příkazů

FINEA	Konec pohybu při dosažení „jemného přesného najetí“
COARSEA	Konec pohybu při dosažení „hrubého přesného najetí“
IPOENDA	Konec pohybu při dosažení „zastavení interpolátoru“
IPOBRKA	Změna bloku v brzdné rampě možná (od SW 6.2)
ADISPOSA	Velikost okna tolerance pro kritérium konce pohybu (od SW 6.4)
Osa	Jméno kanálové osy (X, Y, …)
Hodnota v proc.	Kdy má proběhnout změna bloku, vztaženo k brzdné rampě v %
Int	Režim 0: okno tolerance není aktivní 1: okno tolerance týkající se zadané polohy 2: okno tolerance týkající se skutečné polohy
Real	Velikost okna tolerance. Tato hodnota se přenáší synchronně se zpracováním hlavního programu a nastavovaného parametru 43610: ADISPOSA_VALUE

Funkce

Podobně jako u kritéria změny bloku při dráhové interpolaci (G601, G602 a G603) může být kritérium konce pohybu při interpolaci jednotlivé osy programováno ve výrobním programu popř. v synchronních akcích pro příkaz PLC-osy.
Podle toho, jaké kritérium konce pohybu je nastaveno, budou bloky výrobního programu popř. bloky technologických cyklů s pohyby jednotlivé osy rozdílně rychle ukončeny.
Totéž platí pro příkazy polohování PLC pomocí FC15/ 16/ 18.

© Siemens AG 2004, Všechna práva vyhrazena

Systémová proměnná $AA_MOTEND

Nastavené kritérium konce pohybu může být vyhledáno prostřednictvím systémové proměnné $AA_MOTEND[<osa>] .

- `$AA_MOTEND [<osa>] = 1` Konec pohybu prostřednictvím „přesné najetí jemné“
- `$AA_MOTEND [<osa>] = 2` Konec pohybu prostřednictvím „přesné najetí hrubé“
- `$AA_MOTEND [<osa>] = 3` Konec pohybu prostřednictvím „zastavení IPO“
- `$AA_MOTEND [<osa>] = 4 (od SW 6.2)` Kritérium změny bloku brzdná charakteristika pohybu osy
- `$AA_MOTEND [<osa>] = 5 (od SW 6.4)` Změna bloku v brzdné charakteristice s oknem tolerance vztaženým k „zadané poloze“
- `$AA_MOTEND [<osa>] = 6 (od SW 6.4)` Změna bloku v brzdné charakteristice s oknem tolerance vztaženým ke „skutečné poloze“

Další upozornění

Po resetu zůstane zachována poslední naprogramovaná hodnota.
Literatura: /FB1/, V1 Posuvy

Od SW 6.2

Kritérium změny bloku v brzdné charakteristice

Procentuální hodnota je synchronně se zpracováním hlavního programu zaznamenána do SD 43600: IPOBRAKE_BLOCK_EXCHANGE. Není-li zadána žádná hodnota, je účinná aktuální hodnota tohoto nastavovaného parametru.
Je nastavitelný rozsah od 0 % do 100 %.

Přídavné okno tolerance pro IPOBRKA

K již existujícímu kritériu změny bloku v brzdné charakteristice může být od SW 6.4 zvoleno přídavné kritérium změny bloku okno tolerance. Uvolnění proběhne teprve, když osa

- jako doposud dosáhla zadanou %-tní hodnotu na své brzdné charakteristice a současně
- od SW 6.4 její aktuální skutečná nebo zadaná poloha není vzdálena o více než jednu toleranci od koncové polohy osy v bloku.

Další informace ke kritériu změny bloku polohovacích os viz:
Literatura:/FB2/, P2 Polohovací osy
/PG/, Regulace posuvu a pohyb vřetena

© Siemens AG 2004, Všechna práva vyhrazena

Příklady programování

```
…
N110 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 IPOENDA[X]
```

Pohyb do polohy X100 rychlostí dráhy 1000 ot/min s hodnotou zrychlení 90 % a s koncem pohybu při dosažení zastavení interpolátoru.

t

```
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140 IPOENDA[X]
```

Pohyb do polohy X50, když je vstup 1 aktivní, rychlostí pohybu po dráze 2000 ot/min, hodnotou zrychlení 140 % a koncem pohybu při dosažení zastavení interpolátoru

…

Kritérium změny bloku: brzdná charakteristika ve výrobním programu

```
; Předdefinované nastavení je v platnosti
N40 POS[X]=100
; změna bloku proběhne, když X-osa dosáhne polohu 100 a jemného přesného najetí
N20 IPOBRKA(X,100)      ; aktivování kritéria změny bloku podle brzdné charakteristiky
N30 POS[X]=200          ; změna bloku proběhne, jakmile X-osa začne brzdit
N40 POS[X]=250
;X-osa nebrzdí v poloze 200, nýbrž se jede dál do polohy 250,
;jakmile X-osa začne brzdit, proběhne změna bloku
N50 POS[X]=0            ; X-osa brzdí a pohybuje se zpět do polohy 0
                        ;změna bloku proběhne v poloze 0 a při dosažení jemného
                        přesného najetí
N60 X10F100
N70 M30
…
```

Kritérium změny bloku: brzdná charakteristika v synchronních akcích

V technologickém cyklu:

```
    FINEA               ;Kritérium konce pohybu jemné přesné najetí
    POS[X]=100          ;Technologický cyklus změna bloku proběhne, když X-osa
                        ;dosáhne polohu 100 a polohy jemného přesného najetí
    IPOBRKA(X,100)      ; aktivování kritéria změny bloku podle brzdné charakteristiky
    POS[X]=100          ;POS[X]=100; technologický cyklus změna bloku proběhne,
                        ;jakmile X-osa začne brzdit
    POS[X]=250          ;X-osa nebrzdí v poloze 200, nýbrž se jede dál do polohy 250,
                        jakmile X-osa začne brzdit, proběhne změna bloku
                        ;v technologickém cyklu
    POS[X]=250          ;X-osa brzdí a pohybuje se zpět do polohy 0
                        ;změna bloku proběhne v poloze 0 a dosažení jemného
                        přesného najetí
    M17
```


© Siemens AG 2004, Všechna práva vyhrazena

5.11 Programovatelný blok servo-parametrů (od SW 5.1)



Programování

```
SCPARA [<osa>] = <hodnota>
```



Vysvětlení příkazů

SCPARA	Stanovení bloku parametrů
Osa	Název osy kanálu (X, Y, …)
hodnota	Požadovaný blok parametrů (1<= hodnota <=6)



Funkce

Prostřednictvím příkazu SCPARA může být blok parametrů (sestávající z MD) naprogramován ve výrobním programu a v synchronních akcích (doposud pouze prostřednictvím PLC).

DB3 DBB9 Bit3

Aby nedocházelo ke konfliktům mezi přáním uživatele z PLC a přáním uživatele z NC, je na rozhraní PLC->NCK definován další Bit:
DB3n DBB9 Bit3 „Zadání bloku parametrů prostřednictvím SCPARA blokováno“.



Při blokovaném zadání bloku parametrů prostřednictvím SCPARA nedojde k chybovému hlášení, i když toto bude přesto programováno.

Aktuální blok parametrů může být vyhledán prostřednictvím systémové proměnné $AA_SCPAR[<osa>].

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Do SW 5.1 může být blok servo-parametrů zadán jen prostřednictvím PLC (DB3n DBB9 Bit0-2). U G33, G331 popř. G332 je řídícím systémem vybrán nejvhodnější blok parametrů.
V případě, že blok servo-parametrů má být změněn ve výrobním programu, popř. synchronní akci a v PLC, musí být uživatelský program PLC rozšířen.
Literatura: /FB1/ V1 Posuvy

Příklad programování

```
…
N110 SCPARA[X]= 3          3. blok parametrů je vybrán pro osu X
…
```


© Siemens AG 2004, Všechna práva vyhrazena

Framy

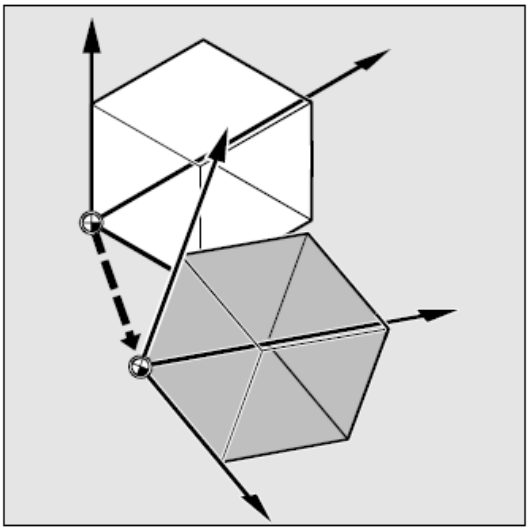
6.1 Transformace souřadnic prostřednictvím proměnné framu 6-238
6.2 Přiřazení hodnot proměnným framů / framům 6-243
6.3 Hrubé a jemné posunutí 6-250
6.4 Posunutí ručním kolečkem 6-251
6.5 Externí posunutí počátku 6-252
6.6 Programování předdefinovaného posunutí, PRESETON 6-253
6.7 Deaktivace framů, DRFOF, G53, G153 a SUPA 6-254
6.8 Výpočet framu ze 3 změřených bodů v prostoru, MEAFRAME 6-255
6.9 NCU-globální framy (od SW 5) 6-259
6.9.1 Framy specifické pro určitý kanál 6-260
6.9.2 Framy s kanálovou působností 6-262

© Siemens AG 2004, Všechna práva vyhrazena

6.1 Transformace souřadnic prostřednictvím proměnné framu

Stanovení transformace souřadnic pomocí proměnné framu

Vedle možností programování popsaných již v návodu k programování „Základy“, můžete stanovit systémy souřadnic také prostřednictvím předdefinovaných proměnných typu frame.



Systémy souřadnic

Jsou definovány následující systémy souřadnic:

MKS: Souřadný systém stroje
BKS: Základní souřadný systém
BNS: Počátek základního souřadného systému
ENS: Nastavitelný počátek souřadného systému
WKS: Souřadný systém obrobku

Co je to předem definovaná proměnná typu frame?

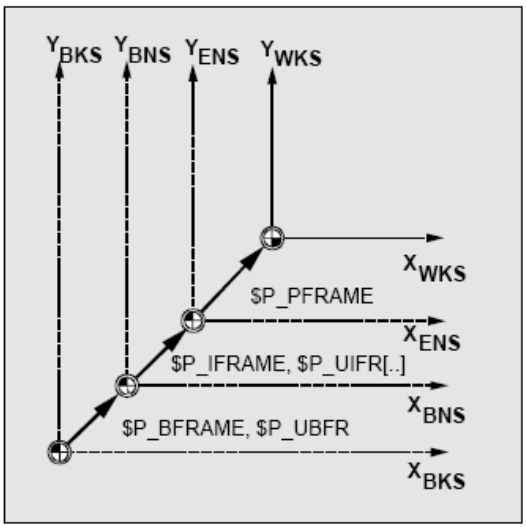
Předdefinované proměnné typu frame jsou klíčová slova, jejichž použití a efekt jsou již v jazyce řídícího systému definovány a které zpracovány v NC-programu.

Možné proměnné typu frame:

- Základní frame (základní posunutí)
- Nastavitelné framy
- Programovatelné framy

© Siemens AG 2004, Všechna práva vyhrazena

Souvislost proměnná typu frame/frame
Transformace souřadnic může být aktivována prostřednictvím přiřazení hodnot framu proměnné typu frame.



Příklad: `SP_FRAME=CTRANS (X,10)`

Proměnná typu frame:
`SP_FRAME` znamená: aktuální programovatelný frame.

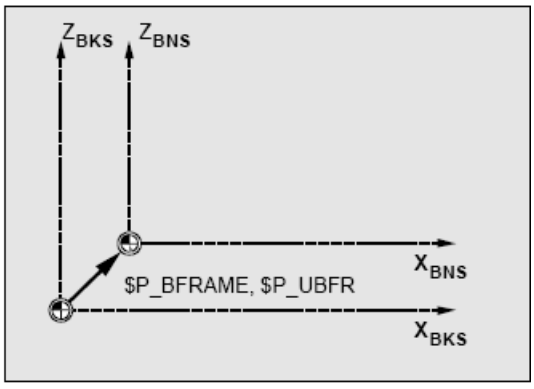
Frame:
`CTRANS (X,10)` znamená: programovatelné posunutí počátku osy X o 10 mm.

Čtení skutečné hodnoty
Prostřednictvím předdefinovaných proměnných ve výrobním programu mohou být čteny aktuální skutečné hodnoty souřadných systémů:

$AA_IM[osa]	Čtení skutečné hodnoty v MKS
$AA_IB[osa]	Čtení skutečné hodnoty v BKS
$AA_IBN[osa]	Čtení skutečné hodnoty v BNS
$AA_IEN[osa]	Čtení skutečné hodnoty v ENS
$AA_IW[osa]	Čtení skutečné hodnoty v WKS

Přehled předdefinovaných proměnných

$P_BFRAME
Aktuální proměnná základního framu, která stanoví vztah mezi základním souřadným systémem (BKS) a základním systémem s počátkem (BNS).



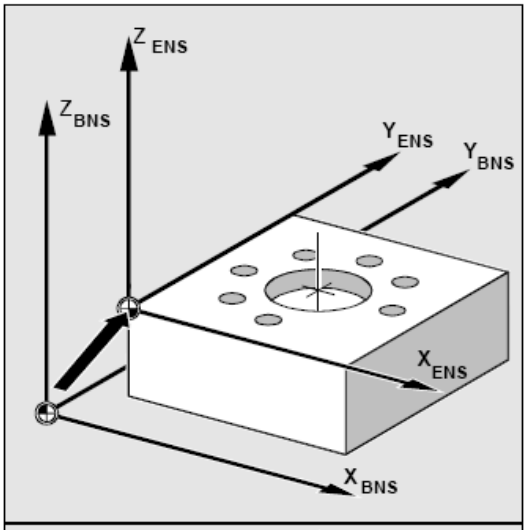
Má-li být základní frame popsaný prostřednictvím $P_UBFR v programu ihned v platnosti, musí být buď

- programováno G500, G54…G599 nebo
- být popsáno $P_FRAME prostřednictvím $P_UBFR.

© Siemens AG 2004, Všechna práva vyhrazena

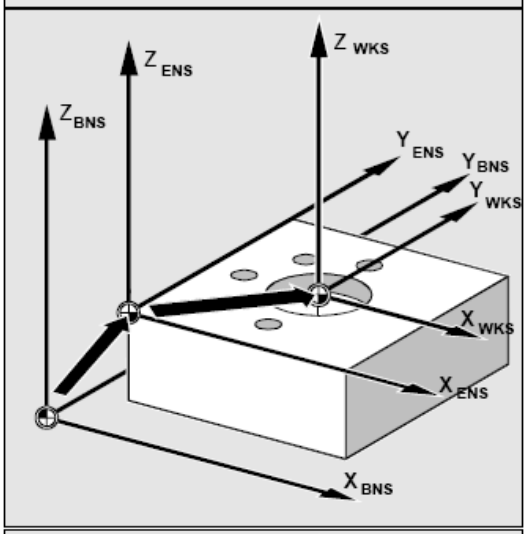
$P_IFRAME
Aktuální nastavitelná proměnná framu, která stanoví vztah mezi základním souřadným systémem s počátkem (BNS) a nastavitelným systémem s počátkem (ENS).
`$P_IFRAME` odpovídá `$P_UIFR[$P_INFRNUM]`

`$P_IFRAME` obsahuje po naprogramování např. G54 prostřednictvím G54 definované posunutí, rotaci, změnu měřítka a zrcadlové převrácení.



$P_PFRAME
Aktuální nastavitelná proměnná framu, která stanoví vztah mezi nastavitelným systémem s počátkem (ENS) a souřadným systémem obrobku (WKS).

$P_PFRAME obsahuje výsledný frame, který vzniká z naprogramování TRANS/ATRANS, ROT/AROT, SCALE/ESCALE, MIRROR/AMIRROR popř. z přiřazení CTRANS, CROT, CMIRROR, CSCALE programovatelnému framu.

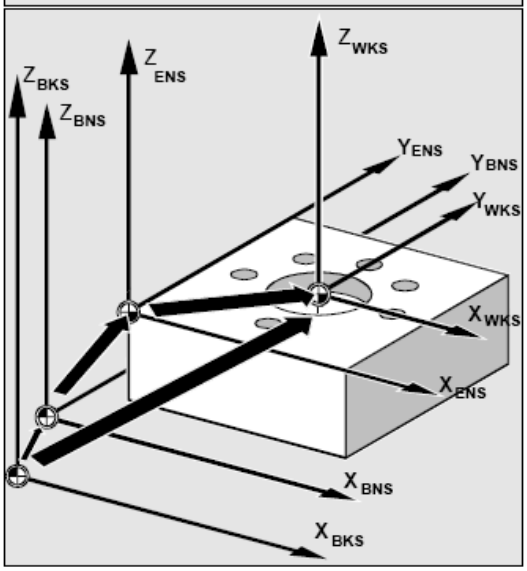


$P_ACTFRAME
Aktuální výsledný celkový frame, který vzniká zřetězením z aktuální proměnné základního framu `$P_BFRAME`, aktuální nastavitelné proměnné framu `$P_IFRAME` a aktuální programovatelné proměnné framu `$P_PFRAME`.

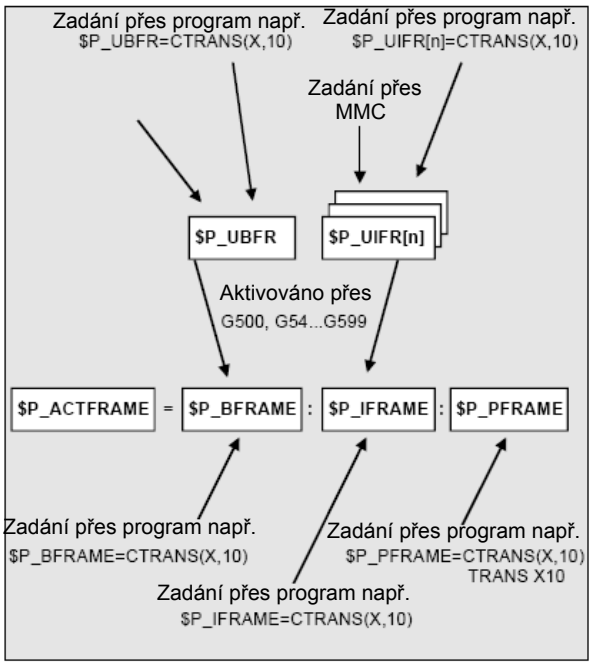
`$P_ACTFRAME` popisuje aktuálně platný počátek obrobku.

V případě, že `$P_IFRAME`, `$P_BFRAME` nebo `$P_PFRAME` jsou změněny, je `$P_ACTFRAME` nově vypočten.

`$P_ACTFRAME` odpovídá
`$P_BFRAME:$P_IFRAME:$P_PFRAME`



© Siemens AG 2004, Všechna práva vyhrazena



Základní frame a nastavitelný frame jsou v platnosti po resetu, jestliže MD 20110 RESET_MODE_MASK je nastaveno následovně:
Bit0=1, Bit14=1 -->$P_UBFR (základní frame) je v platnosti
Bité=1, Bit5=1 -->$P_UIFR[$P_UIFRNUM] (nastavitelný frame) je v platnosti

Předdefinované nastavitelné framy $P_UBFR
Prostřednictvím $P_UBFR je naprogramován základní frame, není ale současně aktivní ve výrobním programu.
Prostřednictvím $P_UBFR zapsaný základní frame je započítán, když

- byl proveden reset a byly zadány Bity 0 a 14 strojního parametru RESET_MODE_MASK,
- byly provedeny příkazy G500, G54…G599.

Předdefinované nastavitelné framy $P_UIFR[n]
Prostřednictvím předdefinované proměnné framu $P_UIFR[n] mohou být z výrobního programu čtena a zapisována nastavitelná posunutí počátku G54 až G599.

Tyto proměnné představují svojí stavbou jednorozměrné pole typu FRAME se jménem $P_UIFR[n].

Přiřazení ke G-příkazům
Standardně je předem nastaveno 5 nastavitelných framů $P_UIFR[0]…$P_UIFR[4] resp. 5 G-příkazů se stejným významem G500 a G54 až G57, pod jejichž adresami mohou být uloženy hodnoty.

© Siemens AG 2004, Všechna práva vyhrazena

`$P_IFRAME=$P_UIFR[0]` odpovídá G500
`$P_IFRAME=$P_UIFR[1]` odpovídá G54
`$P_IFRAME=$P_UIFR[2]` odpovídá G55
`$P_IFRAME=$P_UIFR[3]` odpovídá G56
`$P_IFRAME=$P_UIFR[4]` odpovídá G57

Prostřednictvím strojního parametru můžete změnit počet framů:
`$P_IFRAME=$P_UIFR[5]` odpovídá G505
… … …
`$P_IFRAME=$P_UIFR[99]` odpovídá G599

Tak je možno vytvořit celkem 100 souřadných systémů, které mohou být globálně vyvolány v různých programech, např. jako počátek pro různá zařízení.

Programování proměnných framu a framů vyžaduje v NC-programu samostatný NC-blok. Výjimka: Programování nastavitelného framu prostřednictvím G54. G55, …

© Siemens AG 2004, Všechna práva vyhrazena

6.2 Přiřazení hodnot proměnným framů/framům

V NC-programu mohou být přímo přiřazeny hodnoty, framy mohou být skládány do řetězců nebo lze framy přiřazeny jiným framům.

Přímé přiřazení hodnot

Programování

```
$P_PFRAME=CTRANS X, hodnota osy, Y, hodnota osy, Z, hodnota osy, …)
$P_PFRAME=CROT (X, úhel, Y, úhel, Z, úhel, …)
$P_PFRAME=CSCALE (X, měřítko, Y, měřítko, Z, měřítko, …)
$P_PFRAME=CMIRROR (X, Y, Z)
```

Programování `$P_BFRAME` probíhá analogicky jako `$P_PFRAME`.

Vysvětlení příkazů

`CTRANS`	Posunutí v zadaných osách
`CROT`	Otáčení kolem zadaných os
`CSCALE`	Změna měřítka v zadaných osách
`CMIRROR`	Převrácení směru zadané osy

Funkce

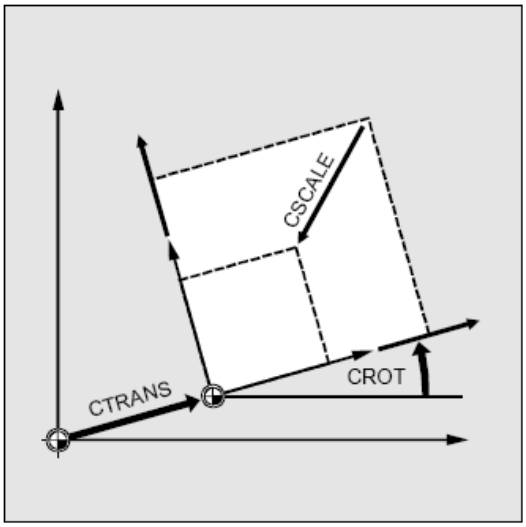
Prostřednictvím těchto funkcí můžete v přímo v NC-programu dosadit do framů nebo proměnných framů hodnoty.

Postup

Můžete programovat více pravidel výpočtu za sebou.

Příklad:
`$P_PFRAME=CTRANS(…):CROT(…):CSCALE…`

Mějte na zřeteli, že příkazy musí být spolu spojeny prostřednictvím operátoru řetězení dvojtečka (…):(…).
Tak budou příkazy za prvé spolu spojeny a za druhé v programovaném pořadí aditivně provedeny.



© Siemens AG 2004, Všechna práva vyhrazena

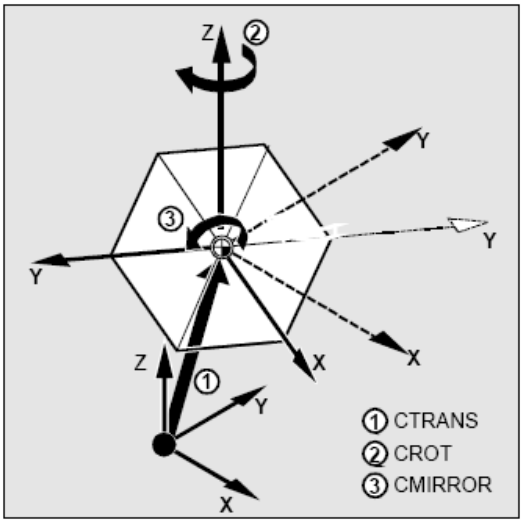
Další upozornění

Prostřednictvím uvedených příkazů jsou programované hodnoty přiřazeny framům a uloženy.

Aktivními se hodnoty stanou teprve, když jsou přiřazeny framu aktivní proměnné framu `$P_BFRAME`, resp. `$P_PFRAME`.

Příklad programování

Přiřazením hodnoty aktuálnímu programovatelnému framu jsou aktivovány posunutí, otočení a zrcadlové převrácení.




```
N10 $P_PFRAME=CTRANS(X,10,Y,20,Z,5):CROT(Z,45):CMIRROR(Y)
```


© Siemens AG 2004, Všechna práva vyhrazena

Čtení a změny komponent framů



Programování (příklady)

`R10=$P_UIFR[$P_UIFRNUM, X, RT]`	Úhel otáčení RT kolem osy X z aktuálně platného nastavitelného posunutí počátku $P_UIFRNUM má být přiřazen proměnné R10.
`R12=$P_UIFR[25, Z, TR]`	Hodnota posunutí TR v ose Z z datového bloku nastaveného framu č. 25 má být přiřazena proměnné R12.
`R15=$P_PFRAME[Y, TR]`	Hodnota posunutí TR v ose Y aktuálního programovatelného framu má být přiřazena proměnné R15.
`$P_PFRAME[X, TR]=25`	Hodnota posunutí TR v ose X aktuálního programovatelného framu má být změněna. Okamžitě platí X25.



Vysvětlení příkazů

`$P_UIFRNUM`	Prostřednictvím této proměnné je automaticky vytvořen vztah k aktuálně platnému nastavitelnému posunutí počátku.
`P_UIFR[n, …, …]`	Prostřednictvím zadání čísla framu n získáte přístup do nastavitelného framu č. n.
`TR` `FI` `RT` `SCMI`	Zadání komponenty, která má být čtena nebo změněna: TR posunutí, FI jemné posunutí, RT otočení, SC změna měřítka, MI zrcadlové převrácení. Navíc (viz příklady) je zadána odpovídající osa.

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Máte možnost přístupu k jednotlivým datům framu, např. k určité hodnotě posunutí nebo úhlu otočení.

Tyto hodnoty můžete měnit nebo přiřadit jiné proměnné.

Postup

Vyvolání framu

Prostřednictvím zadání systémové proměnné $P_UIFRNUM získáte přímý přístup k posunutí počátku aktuálně nastavenému prostřednictvím $P_UIFR popř. G54, G55, … ($P_UIFRNUM obsahuje číslo aktuálně nastaveného framu).

Všechny ostatní uložené nastavitelné framy $P_UIFR vyvoláte pomocí zadání odpovídajícího čísla $P_UIFR[n].

Pro předdefinované proměnné framu a vlastní definované framy zadejte jméno např. $P_FRAME.

Vyvolání data

V hranaté závorce se nachází jméno osy a komponenty framu hodnoty, ke které chcete získat přístup nebo kterou chcete změnit, např. [X, RT] nebo [Z, MI].

© Siemens AG 2004, Všechna práva vyhrazena

Spojení kompletních framů

Kompletní frame může být přiřazen jinému framu.

Programování (příklady)

```
DEF FRAME EINSTELLUNG1
EINSTELLUNG1=CTRANS(X,10)
$P_PFRAME=EINSTELLUNG1
```

Aktuálnímu programovatelnému framu budou přiřazeny hodnoty uživatelem definovaného framu EINSTELLUNG1.

```
DEF FRAME EINSTELLUNG4
EINSTELLUNG4=$P_PFRAME
$P_PFRAME=EINSTELLUNG4
```

Aktuální programovatelný frame je prozatímně uložen a může být v případě potřeby opět uložen zpět.

Další upozornění

Rozsah hodnot pro otáčení RT

Otáčení kolem 1. geometrické osy: -180°až +180°
Otáčení kolem 2. geometrické osy: -89.999°až +90°
Otáčení kolem 3. geometrické osy: -180°až +180°

Řetězce framů

Programování (příklady)

```
$P_IFRAME=$P_UIFR[15]:$P_UIFR[16]
```

$P_UIFR[15] obsahuje např data pro posunutí počátku. Následně budou – na to navazujíce – zpracována data z $P_UIFR[16], např. data pro otočení.

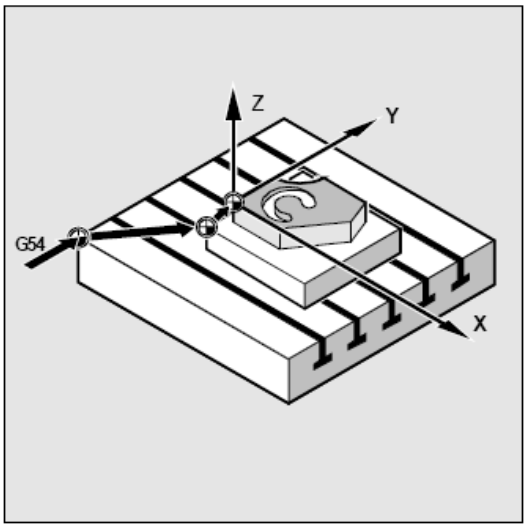
```
$P_UIFR[3]=$P_UIFR[4]:$P_UIFR[5]
```

Nastavitelný frame 3 je vytvořen zřetězením nastavitelných framů 4 a 5.

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Zřetězení framů jsou vhodná např. pro popis více obrobků, které jsou uspořádány na jedné paletě a mají být opracovány v jednom výrobním procesu.



Postup

Framy jsou spojeny v naprogramovaném pořadí, komponenty framů (posunutí, otočení atd.) jsou provedeny postupně aditivně.

Pro popis paletových úkolů by mohly komponenty framu např. obsahovat jen určité dílčí hodnoty, jejichž zřetězením budou generovány různé počátky obrobku.

Mějte na zřeteli, že framy musí být spojeny řetězícím operátorem dvojtečka „:“.

© Siemens AG 2004, Všechna práva vyhrazena

Definice nového framu



Programování

```
DEF FRAME PALETTE 1

PALETTE1=CTRANS(…):CROT(…)…
```



Funkce

Vedle doposud popsaných, předdefinovaných, nastavitelných framů máte také možnost, vytvářet nové framy.
Přitom se jedná o proměnné typu FRAME, které definujete volným pojmenováním.



Postup

Prostřednictvím funkcí CTRANS, CROT, CSCALE a CMIRROR můžete do vašich framů v NC-programu dosadit hodnoty.
Více informací k tomu najdete na předchozích stranách.

Definice otočení framu



Funkce

Orientace v prostoru mohou být definovány pomocí otáčení framu specificky vzhledem k užití:

- ROT: Jednotlivá otočení pro všechny geometrické osy
- ROTS, AROTS, CROTS: Otáčení pomocí zadání prostorového úhlu (maximálně 2); viz popis v /FB1/ K2: Souřadné systémy.
- TOFRAME: Otočení prostřednictvím framu „TOFRAME“, jehož osa Z je shodná se směrem nástroje.
- TOROT: Otáčení prostřednictvím framu „TOROT“, který přepíše pouze rotační podíl již programovaných framů.
- PAROT: Otočení framu vztažené k obrobku. Rotační podíl je určen rotačním podílem orientovatelného držáku nástroje.

© Siemens AG 2004, Všechna práva vyhrazena

6.3 Hrubé a jemné posunutí

Funkce

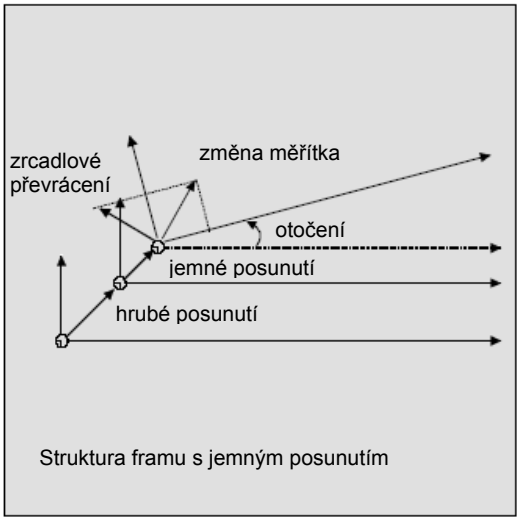
Jemné posunutí

Prostřednictvím příkazu `CFINE (X,.., Y, …)` Může být programováno jemné posunutí základního framu a všech nastavitelných framů.

Hrubé posunutí

Prostřednictvím `CTRANS (. . .)` je stanoveno hrubé posunutí.

Hrubé a jemné posunutí se sčítají do celkového posunutí.



Struktura framu s jemným posunutím

Programování

```
$P_UBFR=CTRANS(x, 10) : CFINE(x, 0.1)      : CROT(x, 45)  ; zřetězení
                                           ;posunutí, jemného posunutí
                                           ;a otočení
$P_UIFR[1]=CFINE(x, 0.5, y, 1.0, z, 0.1)   ; celkový frame je přepsán
                                           ; příkazem CFINE včetně
                                           ; hrubého posunutí
```

Přístup k jednotlivým komponentům jemného posunutí probíhá prostřednictvím zadání složky FI.

Programování

```
DEF REAL FINEX                          ; definování proměnné FINEX
FINEX=$P_UIFR[$P_UIFRNUM, x, FI]        ; čtení jemného posunutí
                                        ; prostřednictvím proměnné FINEX

FINEX=$P_UIFR[3, X, FI]                 ;čtení jemného posunutí osy X ;ve 3. framu
                                        ; prostřednictvím proměnné FINEX
```

Jemné posunutí může proběhnout jen tehdy, když je MD 18600: MM_FRAME_FINE_TRANS=1.

Ovládání změněné jemné posunutí je aktivní až po aktivování odpovídajícího framu, tzn. aktivování proběhne prostřednictvím G500, G54…G599. Aktivované jemné posunutí framu je aktivní tak dlouho, dokud je aktivní frame.

© Siemens AG 2004, Všechna práva vyhrazena

Programovatelný frame nemá žádnou složku jemného posunutí. Je-li programovatelnému framu přiřazen frame s jemným posunutím, bude celek jeho posunutí tvořen součtem hrubého a jemného posunutí. Při čtení programovatelného framu je jemné posunutí vždy nula.

Výrobce stroje

Od SW 5

Pomocí MD 18600: MM_FRAME_FINE_TRANS může být konfigurováno jemné posunutí v následujících variantách:
0: Jemné posunutí nemůže být zadáno, resp. naprogramováno, G58 a G59 nejsou možné.
1: Jemné posunutí pro nastavitelné framy, základní framy, programovatelné framy, G58 a G59 může být zadáno, resp. naprogramováno.

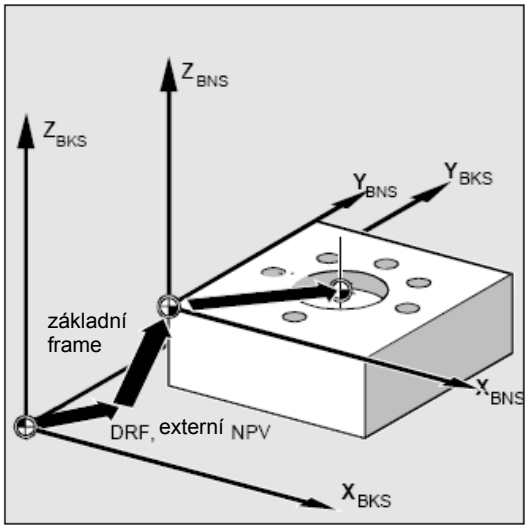
6.4 Posunutí ručním kolečkem (DRF posunutí)

Posunutí ručním kolečkem, DRF

Navíc ke všem v této kapitole popsaným posunutím mohou být stanovena posunutí počátku prostřednictvím ručního kolečka (DRF-posunutí).

DRF-posunutí se promítá do základního souřadného systému. Souvislosti viz obrázek.

Více informací viz návod k obsluze.



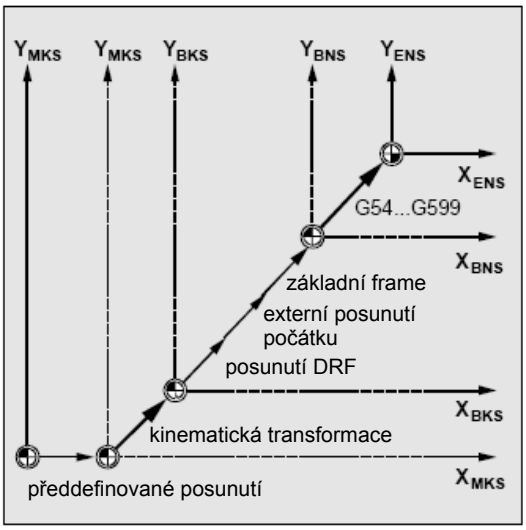
Vymazání posunutí DRF, DRFOF

Příkazem DRFOF je vymazáno posunutí ručním kolečkem pro všechny osy přiřazené kanálu. DRFOF se nachází v samostatném NC-bloku.

6.5 Externí posunutí počátku

Externí posunutí počátku

Toto je další možnost pro posunutí počátku mezi základním souřadným systémem a souřadným systémem obrobku.



Při externím posunutí počátku mohou být programována jen lineární posunutí.

Programování hodnot posunutí, $AA_ETRANS

Programování probíhá prostřednictvím dosazení do systémových proměnných specifických pro osu.

Přiřazení hodnoty posunutí

```
$AA_ETRANS[osa]=RI
```

RI je početní proměnná typu REAL, která obsahuje novou hodnotu.

Externí posunutí není zpravidla zadáno ve výrobním programu, nýbrž dosazeno prostřednictvím PLC.

Ve výrobním programu zapsaná hodnota je v platnosti teprve tehdy, když na rozhraní VDI (NCU-PLC-rozhraní) je nastaven odpovídající signál.

© Siemens AG 2004, Všechna práva vyhrazena

6.6 Programování předdefinovaného posunutí, PRESETON

Programování

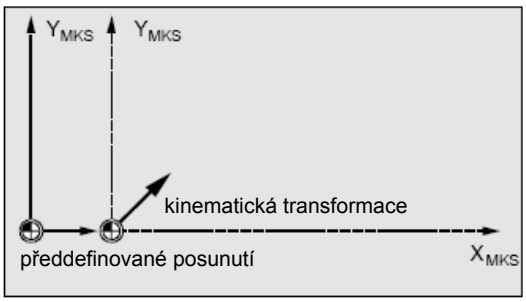
```
PRESETON (osa, hodnota, …)
```

Vysvětlení příkazů

PRESETON	Zadání skutečné hodnoty
Osa	Označení osy stroje
Hodnota	Nová skutečná hodnota, která má platit pro označenou osu

Funkce

Pro speciální použití může být nutné jedné nebo více osám v aktuální poloze (v klidu) přiřadit novou, naprogramovanou skutečnou hodnotu.



Upozornění: Zadání skutečné hodnoty se synchronními akcemi by mělo proběhnout jen prostřednictvím klíčového slova „WHEN“ nebo „EVEREY“.

Postup

Přiřazení skutečných hodnot probíhá v souřadném systému stroje – hodnoty se vztahují k osám stroje.

Příklad:

```
N10 G0 A760
N20 PRESETON(A1,60)
```

Osa A se pohybuje do polohy 760. Osa stroje A1 obdrží v poloze 760 novou skutečnou hodnotu 60.
Od tohoto okamžiku polohování probíhá v novém systému skutečných hodnot.

S funkcí PRESETON je referenční bod neplatný. Proto byste měli používat tuto funkci jen pro osy bez povinnosti najíždět na referenční bod. Má-li být obnoven původní systém, musí být prostřednictvím G74 najeto na referenční body – viz kapitola 3.1.

© Siemens AG 2004, Všechna práva vyhrazena

6.7 Deaktivace framů, DRFOF, G53, G153 a SUPA

Vysvětlení příkazů

DRFOF	Vypnutí (vymazání) posunutí ručním kolečkem (DRF)
G53	Blokové vypnutí programovatelného framu a všech nastavitelných framů
G153	Blokové vypnutí programovatelného framu, základního framu a všech nastavitelných framů
SUPA	Blokové vypnutí všech programovatelných framů, základního framu a posunutí ručním kolečkem (DRF)

Další upozornění

Programovatelné framy vymažete prostřednictvím přiřazení „nulového framu“ (bez zadání osy) programovatelnému framu.

Příklad:

```
$P_PFRAME=TRANS( )
$P_PFRAME=ROT( )
$P_PFRAME=SCALE( )
$P_PFRAME=MIRROR( )
```


© Siemens AG 2004, Všechna práva vyhrazena

6.8 Výpočet framu ze 3 změřených bodů v prostoru, MEAFRAME

MEAFRAME je rozšířením jazyka 840D pro podporu měřicích cyklů.
Tato funkce platí od SW 4.3

Funkce

Je-li obrobek uveden do polohy pro obrábění, je jeho relativní poloha vůči kartézskému souřadnému systému stroje ve vztahu k jeho ideální poloze jak posunutá, tak i otočená.
Pro přesné opracování nebo měření je nezbytné nutné buď nákladné fyzikální nastavení nebo změna pohybů ve výrobním programu.
Frame může být stanoven snímáním tří bodů v prostoru, jejichž ideální polohy jsou známé.
Snímání se provádí pomocí dotykového nebo optického senzoru, který se dotýká speciálních na nosné desce precizně fixovaných otvorů nebo měřicích kuliček.
Funkce MEAFRAME vypočte frame ze tří ideálních a vzájemně si odpovídajících změřených bodů.
Aby změřené hodnoty mohly být přiřazeny ideálním souřadnicím pomocí kombinace otočení a posunutí, musí být měřicími body odvozený trojúhelník kongruentní k ideálnímu trojúhelníku.
Toto se provádí kompenzačním algoritmem, který minimalizuje součet čtverců odchylek, které změřené polohy převádějí do ideálního trojúhelníka.
Efektivně nutné zkreslení měřicích bodů může sloužit jako indikátor kvality měření a je proto výstupem MEAFRAME jako další proměnná.

Další upozornění

Prostřednictvím MEAFRAME vytvořený frame může být pomocí funkce ADDFRAME od SW 6.3 transformován v jiný frame v řetězci framů.
Viz příklad použití k zřetězení MEAFRAME pro další korekce.

© Siemens AG 2004, Všechna práva vyhrazena

Programování

```
MEAFRAME IDEAL_POINT,MEAS_POINT,FIT_QUALITY)
```

Vysvětlení příkazů

MEAFRAME	Výpočet framu ze 3 měřicích bodů v prostoru
IDEAL_POINT	2-rozměrné pole Real, které obsahuje tři souřadnice ideálních bodů
MEAS_POINT	2-rozměrné pole Real, které obsahuje tři souřadnice změřených bodů
FIT_QUALITY	Proměnná Real, pomocí níž jsou předány následující informace -1: Ideální body leží téměř na přímce: frame nemohl být vypočítán. Předaná proměnná obsahuje neutrální frame. -2: Body měření leží téměř na přímce: frame nemohl být vypočítán. Předaná proměnná obsahuje neutrální fram. -4: Výpočet matice rotace se z jiného důvodu nedaří. pozitivní hodnota: Součet zkreslení (vzdáleností mezi body), které jsou nutné pro převedení změřeného trojúhelníka v trojúhelník, který je kongruentní k ideálnímu trojúhelníku.

Příklad použití

```
; výrobní program1
;
DEF FRAME CORR_FRAME
;
; stanovení měřicích bodů
DEF REAL IDEAL_POINT[3,3] = SET(10.0,0.0,0.0, 0.0,10.0,0.0, 0.0,0.0,10.0)
DEF REAL MEAS_POINT[3,3] = SET(10.1,0.2,-0.2, -0.2,10.2,0.1, -0.2,0.2, 9.8) ; pro zkoušku
DEF REAL FIT_QUALITY = 0
;
DEF REAL ROT_FRAME_LIMIT = 5      ; dovoluje max. 5°pootočení polohy součásti
DEF REAL FIT_QUALITY_LIMIT = 3    ; dovoluje max. 3 mm posunutí mezi ideálním
                                  ; a naměřeným trojúhelníkem
DEF REAL SHOW_MCS_POS1[3]
DEF REAL SHOW_MCS_POS2[3]
DEF REAL SHOW_MCS_POS3[3]
; ======================================================
;
N100 G01 G90 F5000
N110 X0 Y0 Z0
;
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY)
;
N230 IF FIT_QUALITY < 0
SETAL(65000)
```


© Siemens AG 2004, Všechna práva vyhrazena

```
GOTOF NO_FRAME
ENDIF
,
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT
SETAL(65010)
GOTOF NO_FRAME
ENDIF
;
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT          ; omezení 1. RPY-úhlu
SETAL(65020)
GOTOF NO_FRAME
ENDIF
;
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT          ; omezení 2. RPY-úhlu
SETAL(65021)
GOTOF NO_FRAME
ENDIF
;
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT          ; omezení 3. RPY-úhlu
SETAL(65022)
GOTOF NO_FRAME
ENDIF
;
N300 $P_IFRAME=CORR_FRAME; Aktivovat snímací frame pomocí nastavitelného framu
;
; prověřit frame umístěním geometrických os do ideálních bodů
;
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]
N410 SHOW_MCS_POS1[0]=$AA_IM[X]
N420 SHOW_MCS_POS1[1]=$AA_IM[Y]
N430 SHOW_MCS_POS1[2]=$AA_IM[Z]
;
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]
N510 SHOW_MCS_POS2[0]=$AA_IM[X]
N520 SHOW_MCS_POS2[1]=$AA_IM[Y]
N530 SHOW_MCS_POS2[2]=$AA_IM[Z]
;
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]
N610 SHOW_MCS_POS3[0]=$AA_IM[X]
N620 SHOW_MCS_POS3[1]=$AA_IM[Y]
N630 SHOW_MCS_POS3[2]=$AA_IM[Z]
;
N700 G500                ; deaktivování nastavitelného framu, jelikož je předem nastavený nulový frame
                         ; (není zadána žádná hodnota)
;
NO_FRAME:
M0
M30
```

© Siemens AG 2004, Všechna práva vyhrazena

Příklad použití pro zřetězení framů

Zřetězení MEAFRAME pro korekce

Funkce MEAFRAME () dává frame korekce.
Je-li tento frame korekce zřetězen s nastavitelným framem $P_UIFR[1], který byl aktivní při vyvolání funkce např. G54, tak obdržíme nastavitelný frame pro další přepočty k posuvu nebo opracování.

Zřetězení s ADDFRAME (od SW6.3)

Má-li být tento frame korekce v řetězci framů platný na jiném místě, může být použita funkce ADDFRAME () pro vřazení do některého ze základních framů kanálu nebo systémového framu.

Ve framech nesmí být přitom aktivní:

- zrcadlové převracení prostřednictvím MIRROR
- změna měřítka prostřednictvím SCALE

Vstupními parametry požadovaných a skutečných hodnot jsou souřadnice obrobku. V základním systému řídícího systému je třeba tyto souřadnice vždy zadat

- metricky nebo v palcích (G71, G70) a jako
- k rádiusu se vztahující rozměr.

Další upozornění

Další informace ke zřetězení framů viz
/FB/ K2, Osy, souřadné systémy, framy

© Siemens AG 2004, Všechna práva vyhrazena

6.9 NCU-globální framy (od SW 5)

Funkce

NCU-globální framy se v celém NCU vyskytují jen jednou pro všechny kanály. NCU-globální framy mohou být zapisovány a čteny ze všech kanálů. Aktivování NCU-globálních framů probíhá v příslušném kanálu.
Prostřednictvím globálních framů mohou být pro kanálové osy a osy stroje s posunutími aktivovány změny měřítka a zrcadlová převrácení.
U globálních framů neexistuje žádná geometrická souvislost mezi osami. Proto nemohou být prováděna žádná otáčení a programování identifikátorů geometrických os.

- Pro globální framy se nedají použít žádné rotace. Programování rotace je odmítnuto alarmem: „18310 kanál % 1 blok % 2 frame: Rotace nepřípustná“.
- Zřetězení globálních framů a framů specifických pro kanály je možné. Výsledný frame obsahuje všechny podíly framů včetně rotací pro všechny osy. Přiřazení framu s podíly rotace globálnímu framu je odmítnuto alarmem „Frame: Rotace nepřípustná“.

NCU-globální základní framy: $P_NCBFR[n]
Může být navrženo až 8 NCU-globálních základních framů.

© Siemens AG 2004, Všechna práva vyhrazena

Výrobce stroje

Počet globálních framů je konfigurován pomocí SD. (viz /FB/ K2, Osy, souřadnice, framy)
Současně mohou existovat základní kanálové framy.

Globální framy mohou být zapisovány a čteny ze všech kanálů jedné NCU. Při zapisování globálních framů musí mít uživatel na zřeteli koordinaci kanálů. To může být realizováno např. prostřednictvím Wait-značek (WAITMC).

NCU-globální nastavitelné framy: $P_UIFR[n]
Všechny nastavitelné framy G500, G54…G599 mohou být navrženy buď NCU-globálně nebo specificky pro kanál.

Výrobce stroje

Všechny nastavitelné framy mohou být s pomocí SD 18601 MM_NUM_GLOBAL_USER_FRAMES překonfigurovány v globální framy.
Viz /FB/ K2, Osy, souřadnice, framy.
Jako identifikátor os u programovacích příkazů framů mohou být použity identifikátory os kanálu a identifikátory os stroje.
Programování identifikátorů geometrických os je odmítnuto alarmem.

6.9.1 Framy specifické pro určitý kanál

Funkce

Pomocí MD 28081 MM_NUM_BASE_FRAMES může být konfigurován počet základních framů v kanálu. Standardní konfigurace je uspořádána tak, že existuje nejméně jeden základní frame v rámci jednoho kanálu.
Maximálně je možných 8 základních framů na jeden kanál. Navíc k 8 základním framům v kanálu může existovat ještě 8 NCU-globálních základních framů.

© Siemens AG 2004, Všechna práva vyhrazena

Nastavitelné framy nebo základní framy mohou být
- prostřednictvím výrobního programu a
- prostřednictvím BTSS

obsluhou a z PLC zapisovány a čteny.
Jemné posunutí je možné i pro globální framy.
Potlačení globálních framů probíhá stejně jako u framů specifických pro kanál prostřednictvím G53, G153, SUPA a G500.

$P_CHBFR[n]
Pomocí systémové proměnné $P_CHBFR[n] mohou být zapisovány a čteny základní framy.
Při zápisu základního framu není zřetězený celkový základní frame aktivován, aktivování proběhne teprve provedením některého z příkazů G500, G54…G599.
Proměnná slouží převážně jako paměť pro zápisy MMC nebo PLC. Tyto proměnné framu jsou jištěny prostřednictvím zajištění dat.

První základní frame v kanálu
Zápis do předdefinované proměnné $P_UBFR neaktivuje základní frame s indexem pole 0 současně, aktivování proběhne teprve provedením některého příkazu G500, G54…G599. Proměnná může být zapisována a čtena také v programu.

$P_UBFR
$P_UBFR je identický s $P_CHBFR[0].
Standardně se v kanálu nachází jeden základní frame, takže systémová proměnná je kompatibilní se staršími stavy. Neexistuje-li základní frame specifický pro kanál, je při zápisu nebo čtení hlášen alarm „Frame: Příkaz nepřípustný“.

© Siemens AG 2004, Všechna práva vyhrazena

6.9.2 Framy s kanálovou působností



Funkce

Od SW 6.1
Aktuální systémové framy pro
$P_PARTFRAME TCARR a PAROT
$P_SETFRAME nastavení skutečné hodnoty a škrábnutí,
$P_EXTFRAME externí posunutí počátku.
Prostřednictvím těchto systémových proměnných může být ve výrobním programu čten a zapisován aktuální systémový frame.

$P_NCBFRAME[n]
Aktuální NCU-globální základní framy
Prostřednictvím systémové proměnné $P_NCBFRAME[n] mohou být čteny a zapisovány aktuální globální prvky pole základního framu. Výsledný celkový základní frame je započítán v procesu zápisu v kanálu.
Změněný frame bude aktivní pouze v tom kanálu, ve kterém byl frame programován. Má-li být frame změněn pro všechny kanály NCU, musí být současně popsán pomocí $P_NCBFR[n] a $P_NCBFRAME[n]. Ostatní kanály pak ještě musí frame aktivovat prostřednictvím např. G54. Při zápisu základního framu je celkový základní frame znovu vypočítán.

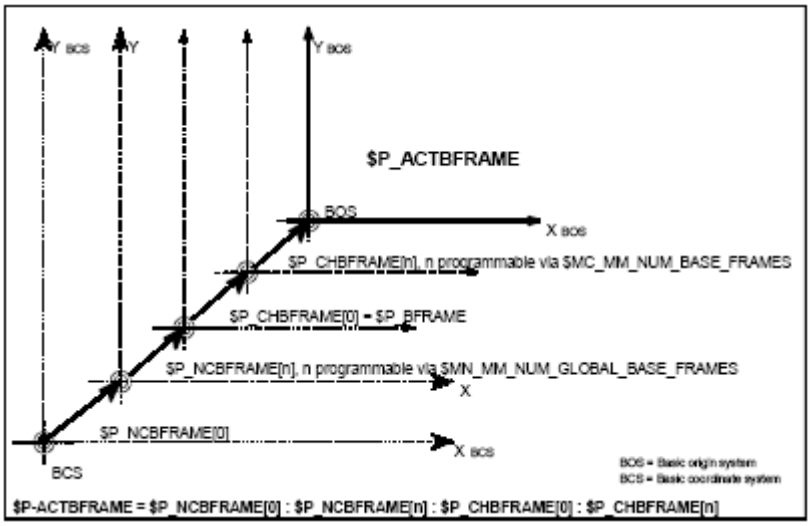
$P_CHBFRAME[n]
aktuální základní kanálové framy
Prostřednictvím systémové proměnné $P_CHBFRAME[n] mohou být čteny a zapisovány prvky pole základního framu kanálu. Výsledný celkový základní frame je započítán v procesu zápisu v kanálu. Při zápisu základního framu je celkový základní frame znovu vypočítán.

© Siemens AG 2004, Všechna práva vyhrazena

$P_BFRAME
Aktuální 1. základní frame v kanálu
Prostřednictvím předdefinované proměnné framu $P_BFRAME může být aktuální základní frame s indexem pole 0, který je v kanálu platný, ve výrobním programu čten a zapisován. Zapsaný frame je ihned započítán.
$P_BFRAME je identický s $P_CHBFRAME[0].
Systémová proměnná má standardně vždy platnou hodnotu. Neexistuje-li základní frame specifický pro kanál, je při zápisu nebo čtení hlášen alarm „Frame: Příkaz nepřípustný“.

$P_ACTBFRAME
Celkový základní frame
Proměnná $P_ACTBFRAME zjišťuje zřetězený celkový základní fram. Proměnná je pouze ke čtení.

$P_ACTBFRAME odpovídá
$P_NCBFRAME[0] **:** ... **:** $P_NCBFRAME[n] **:**
$P_CHBFRAME[0] **:** ... **:** $P_CHBFRAME[n].



$P_CHBFRMASK a $P_NCBFRMASK
Celkový základní frame
Prostřednictvím systémové proměnné $P_CHBFRMASK a $P_NCBFRMASK může uživatel zvolit, které základní framy chce zařadit do výpočtu „celkového“ základního framu. Proměnné mohou být programovány jen v programu a čteny prostřednictvím BTSS. Hodnota proměnných je interpretována jako bitová maska a udává, který prvek pole základního framu bude funkcí $P_ACTBFRAME zařazen do výpočtu.

Prostřednictvím $P_CHBFRMASK může být zadáno, které základní framy specifické pro kanál, a prostřednictvím $P_NBCBFRMASK, které NCU-globální základní framy budou započítány.

Programováním proměnných bude celkový základní frame a celkový frame nově vypočítán.
Po resetu a v základním nastavení je hodnota
$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK a
$P_NCBFRMASK = $MN_NCBFRAME_RESET_MASK.
např.
$P_NCBFRMASK = 'H81' ; $P_NCBFRAME[0] : $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11' ; $P_CHBFRAME[0] : $P_CHBFRAME[4]

$P_IFRAME
Aktuální nastavitelný frame
Prostřednictvím předdefinované proměnné framu $P_IFRAME může být aktuální nastavitelný frame, který je v kanálu platný, ve výrobním programu čten a zapisován. Zapsaný nastavitelný frame je ihned započítán.
U NCU-globálních nastavitelných framů je změněný frame v platnosti jen v kanálu, ve kterém byl frame programován. Má-li být frame změněn pro všechny kanály NCU, musí být současně popsán $P_UIFR[n] a $P_IFRAME. Ostatní kanály musí potom odpovídající frame aktivovat např. prostřednictvím G54.

© Siemens AG 2004, Všechna práva vyhrazena

Od SW 6.1
Aktuální systémové framy pro
$P_TOOLFRAME TOROT a TOFRAME
Ab SW 6.3
$P_WPFRAME vztažné body obrobku
Ab SW 6.4
$P_TRAFRAME transformace

Pomocí těchto proměnných může být ve výrobním programu aktuální systémový frame načítán a zapisován.

$P_FRAME
Aktuální programovatelný frame
$P_FRAME je programovatelný frame, který vyplývá z programování TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ESCALE, MIRROR/AMIRROR, resp. z přiřazení CTRANS, CROT, CMIRROR, CSCALE programovatelnému framu.
Aktuální, programovatelná proměnná framu, která určuje vztah mezi nastavitelným počátkem systému (ENS) a souřadného systému obrobku (WKS).

Od SW 6.3
Aktuální systémový frame pro
$P_CYCFRAME cykly
Prostřednictvím této systémové proměnné může být ve výrobním programu aktuální systémový frame načítán a zapisován.

© Siemens AG 2004, Všechna práva vyhrazena

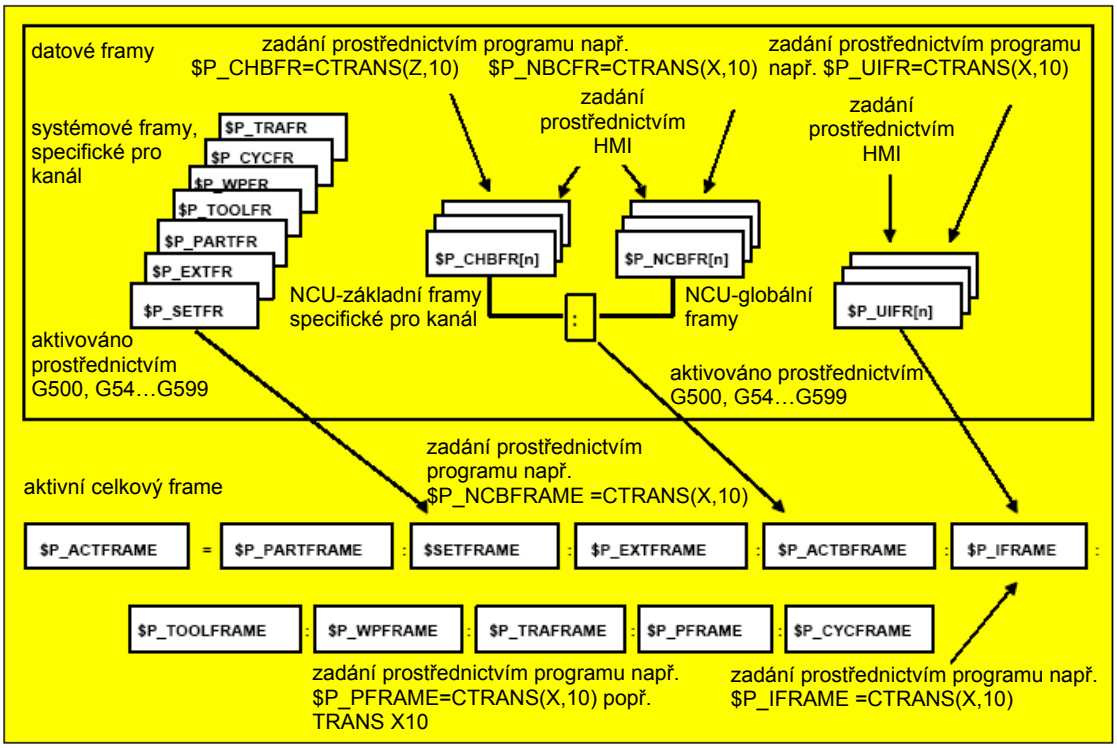
$P_ACTFRAME

Aktuální celkový frame

Aktuální výsledný celkový frame $P_ACTFRAME je výsledkem zřetězení všech základních framů, aktuálního nastavitelného framu a programovatelného framu. Aktuální frame je aktualizován vždy tehdy, když se změní nějaká část framu.

$P_ACTFRAME odpovídá od SW 6.3
$P_SETFRAME : $P_EXTFRAME : $P_PARTFRAME : $P_ACTBFRAME : $P_IFRAME : $P_TOOLFRAME : $P_WPFRAME : $P_PFRAME : $P_CYCFRAME

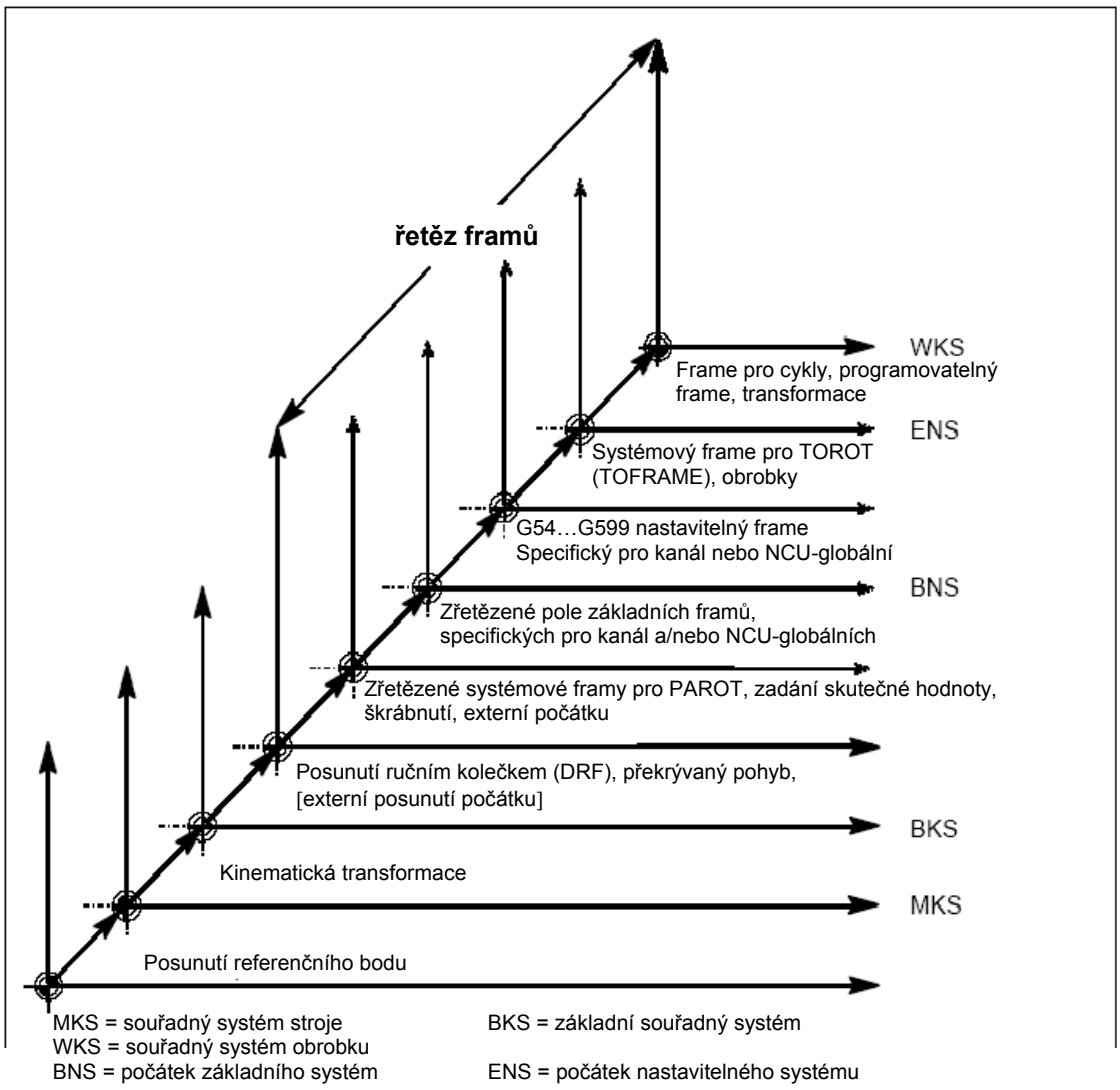
$P_ACTFRAME odpovídá od SW 6.4
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME: $P_ACTBFRAME : $P_IFRAME : $P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME: $P_PFRAME : $P_CYCFRAME



© Siemens AG 2004, Všechna práva vyhrazena

Zřetězení framů

Aktuální frame se skládá z celkového základního framu, nastavitelného framu, systémového framu a programovatelného framu dle výše uvedeného aktuálního celkového framu.



Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Transformace

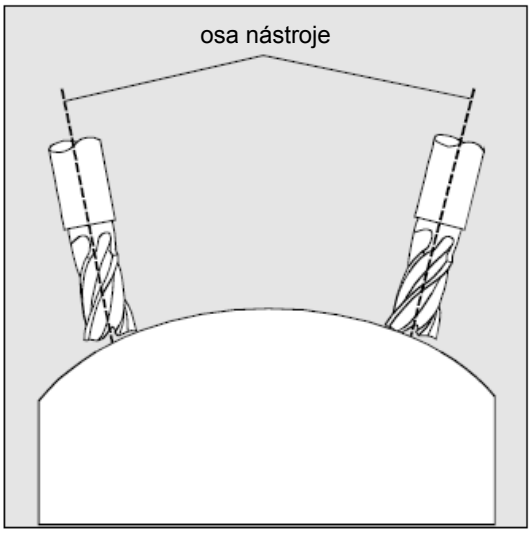
7.1 Tří-, čtyř-, a pětiosová transformace: TRAORI 7-270
7.1.1 Programování orientace nástroje 7-274
7.1.2 Otáčení orientace nástroje: ORIROTA, ORIROTR, ORIROTT 7-279
7.1.3 Vztažný systém os orientace – ORIWKS, ORIMKS 7.281
7.1.4 Singulární postavení a jejich úprava 7-282
7.1.5 Osy orientace (od SW 5.2) ORIAXES, ORIVECT, ORIEULER, ORIRPY 7-283
7.1.6 Kartézský pohyb PTP (od SW 5.2) 7-286
7.1.7 Online korekce délky nástroje: TOFON, TOFOF (od SW 6.4) 7-290
7.2 Frézování na rotačních součástech: TRANSMIT 7-293
7.3 Transformace pláště válce: TRACYL 7-296
7.4 Šikmá osa: TRAANG 7-302
7.4.1 Programování šikmé osy: G05, G07, (od SW 5.3) 7-306
7.5 Okrajové podmínky při volbě transformace 7-307
7.6 Deaktivování transformace: TRAFOOF 7.309
7.7 Zřetězené transformace 7-310
7.8 Přepínatelné geometrické osy, GEOAX 7-313

© Siemens AG 2004, Všechna práva vyhrazena

7.1 Tří-, čtyř- a pětiosová transformace: TRAORI

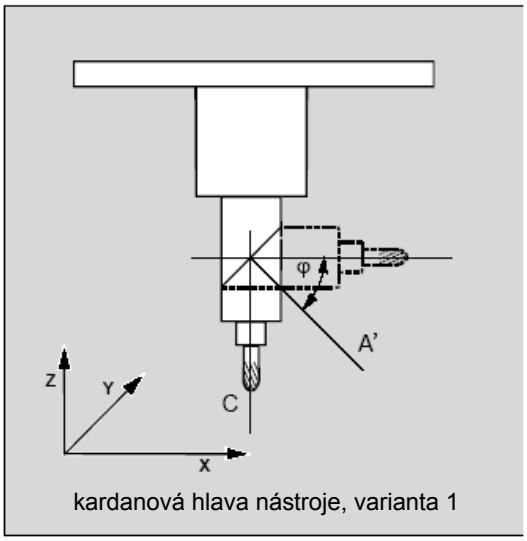
Aby bylo dosaženo optimálních řezných podmínek při opracování prostorově zakřivených ploch, musí být úhel nastavení nástroje měnitelný.

Jakou konstrukcí stroje je toho dosaženo, je uloženo v datech os.



Kardanová hlava nástroje
Zde stanovují tři lineární osy (X,Y,Z) a dvě osy orientace úhel nastavení a pracovní bod nástroje. Jedna z obou os orientace je vložena jako šikmá osa v příkladu A´- v mnoha případech jako 45°-uspořádání.

Sled os orientace a směr orientace nástroje jsou nastaveny v závislosti na kinematice stroje prostřednictvím strojních parametrů. Ve zde ukázaných příkladech vidíte uspořádání na příkladu kinematiky stroje CA!



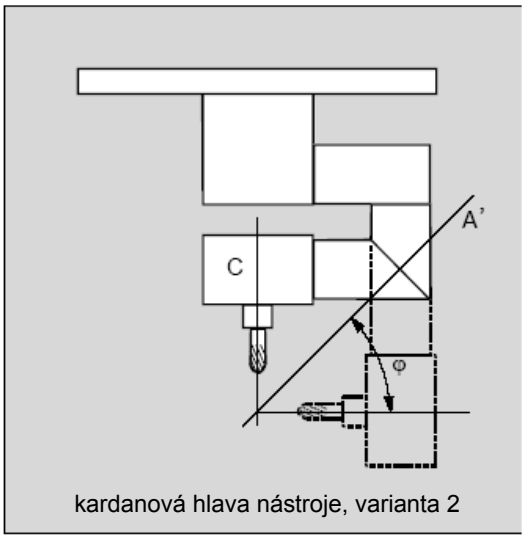
kardanová hlava nástroje, varianta 1

© Siemens AG 2004, Všechna práva vyhrazena

Platí následující možné souvislosti:
A´ leží pod úhlem φ k ose X
B´ leží pod úhlem φ k ose Y
C´ leží pod úhlem φ k ose Z

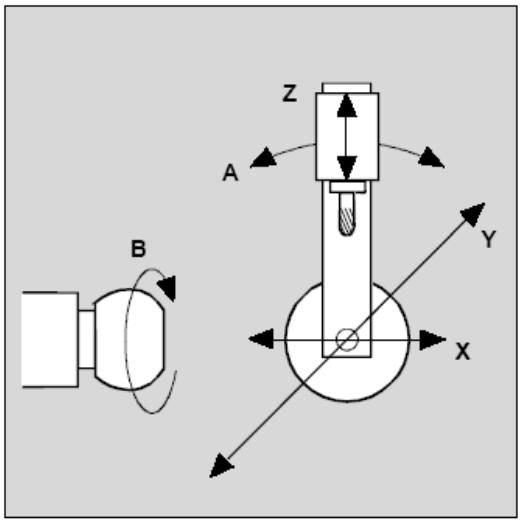
Úhel φ může být prostřednictvím strojních parametrů zadán v rozsahu 0°až +89°.

V závislosti na zvoleném směru orientace nástroje musí být v NC-programu aktivní pracovní rovina ((G17, G18, G19) nastavena tak, aby se korekce délky nástroje uplatňovala ve směru orientace nástroje.



kardanová hlava nástroje, varianta 2

Transformace s otočnou lineární osou
Zde se jedná o uspořádání s pohnutým obrobkem a pohnutým nástrojem.
Kinematika se skládá ze tří lineárních os (X,Y,Z) a dvou pravoúhle uspořádaných otočných os.
První kruhová osa je např. pohybována dvěma lineárními osami přes křížové sáně, nástroj je umístěn paralelně ke třetí lineární ose.
Druhá otočná osa otáčí obrobkem.
Třetí lineární osa (otočná osa) leží v rovině křížových saní.



Sled rotačních os a směr orientace nástroje jsou v závislosti na kinematice stroje nastaveny prostřednictvím strojních parametrů.

Platí následující možné souvislosti:

Osy:	Sledy os:
1. kruhová osa	A A B B C C
2. kruhová osa	B C A C A B
Otočená lineární osa	Z Y Z X Y X

© Siemens AG 2004, Všechna práva vyhrazena

3- a 4-osové transformace

3- a 4-osové transformace jsou zvláštní formy 5 osových transformací.
Uživatel může konfigurovat dvě resp. tři translační osy a jednu rotační osu.
Transformace vycházejí z toho, že rotační osa stojí pravoúhle k rovině orientace.
Orientace nástroje je možná jen v rovině, která je kolmá na rotační osu. Transformace podporuje typy stroje s pohyblivým nástrojem a pohyblivým obrobkem.

Programování

```
TRAORI (n)
TRAORI (n,X,Y,Z,A,B)
TRAFOOF
```

Vysvětlení příkazů

TRAORI	Aktivuje první stanovenou transformaci orientace
TRAORI(n)	Aktivuje prostřednictvím n stanovenou transformaci orientace
n	Číslo transformace (n=1 nebo 2), TRAORI(1) odpovídá transformace zapnuta
X,Y,Z	Složka vektoru orientace, do níž ukazuje nástroj.
A,B	Programovatelný offset pro kruhové osy (od SW 7.1)
TRAFOOF	Vypnutí transformaci

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Po zapnutí transformace se údaje polohy (X,Y,Z) vztahují vždy ke špičce nástroje. Změny poloh na transformaci se podílejících kruhových os vedou k takovým vyrovnávacím pohybům ostatních os stroje, že poloha špičky nástroje zůstane nezměněna.
Transformace orientace je vždy směrována od špičky nástroje k upnutí nástroje.

Příklady pro generické transformace:

TRAORI(1,0,0,1)	Z-směr
TRAORI(1,0,1,0)	Y-směr
TRAORI(1,0,1,1)	Y/Z-směr (odpovídá postavení –45°)

Offset pro osy orientace (od SW 7.1)
Při aktivování transformace orientace může být navíc přímo programován offset pro osy orientace.
Smějí být vynechány parametry, pokud je při programování dodržena správná posloupnost.

Příklad:
TRAORI(,,,A,B) má-li být zadán jediný offset

Alternativně k přímému programování může tento dodatečný offset pro osy orientace být automaticky převzat z momentálně aktivního posunutí počátku. Převzetí je navrženo prostřednictvím strojních parametrů.

Literatura
/FB/ F2, 3 až 5 osové transformace

© Siemens AG 2004, Všechna práva vyhrazena

7.1.1 Programování orientace nástroje

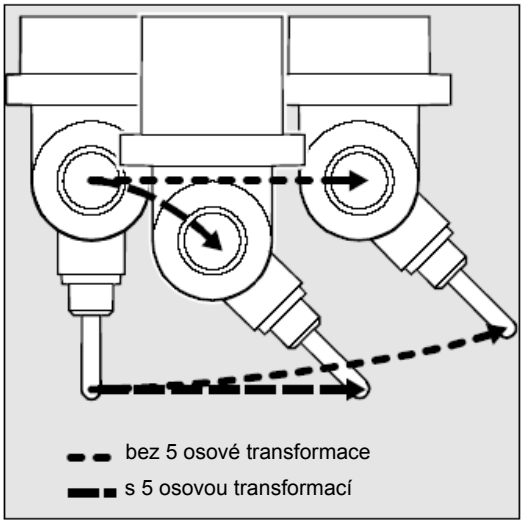
Zpravidla jsou 5 osové programy vytvářeny systémy CAD/CAM a nikoli zadávány na ovládacím panelu řídícího systému. Proto se následující vysvětlení týkají hlavně programátorů postprocesorů.

Pro programování orientace nástroje existují 3 možnosti:

1. Přímé programování pohybu kruhových os. Změna orientace probíhá vždy v základním systému souřadnic, resp. v souřadném systému stroje. Osy orientace jsou brány za synchronní osy.

2. Programování v Eulerových úhlech nebo RPY úhlech prostřednictvím A2, B2, C2 nebo
   Programování směrového vektoru pomocí A3, B3, C3. Směrový vektor směřuje od špičky nástroje k upnutí nástroje.

3. Programování prostřednictvím úhlu předstihu LEAD a bočního úhlu TILT (čelní frézování)



Ve všech případech je programování orientace přípustné jen tehdy, když je transformace orientace zapnutá.

Výhoda: Tyto programy jsou přizpůsobitelné jakékoli kinematice stroje.

© Siemens AG 2004, Všechna práva vyhrazena

Programování

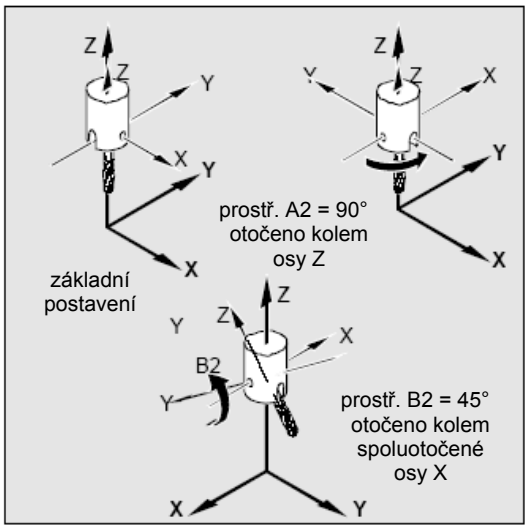
`G1 X Y Z A B C`	Programování pohybu kruhových os
`G1 X Y Z A2= B2= C2=`	Programování v Eulerových úhlech
`G1 X Y Z A3= B3= C3=`	Programování směrového vektoru
`G1 X Y Z A4== B4== C4==`	Programování vektoru normály plochy na začátku bloku
`G1 X Y Z A5= B5== C5==`	Programování vektoru normály plochy na konci bloku
`LEAD=`	Úhel předstihu pro programování orientace nástroje
`TILT=`	Boční úhel pro programování orientace nástroje

Prostřednictvím strojního parametru může být přepínáno mezi Eulerovými úhly a mezi RPY úhly.

Programování v Eulerových úhlech
Při programování orientace jsou prostřednictvím A2, B2, C2 programované hodnoty interpretovány jako Eulerovy úhly (ve stupních).

Vektor orientace vznikne, je-li otočen vektor ve směru Z nejprve prostřednictvím A2 kolem osy Z, potom prostřednictvím B2 kolem nové osy X a nakonec prostřednictvím C2 kolem nové osy Z.

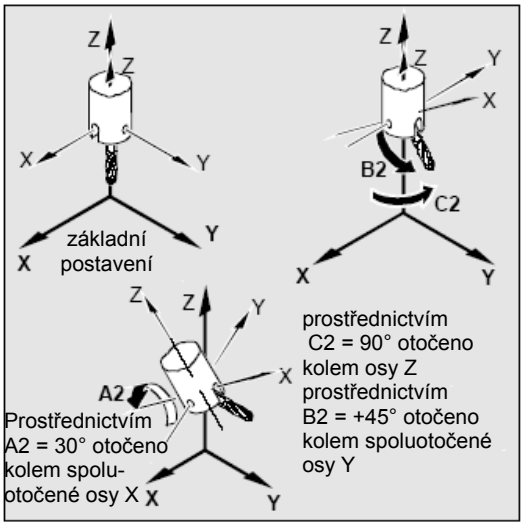
V tomto případě je hodnota C2 (otáčení kolem nové osy Z) bezvýznamná a nemusí být programována.



Programování v RPY úhlech

Při programování orientace jsou hodnoty naprogramované prostřednictvím A2, B2, C2 interpretovány jako RPY úhly (ve stupních).

Vektor orientace vznikne, je-li otočen vektor ve směru Z nejprve pomocí C2 kolem osy Z, potom prostřednictvím B2 kolem nové osy Y a nakonec pomocí A2 kolem nové osy X.

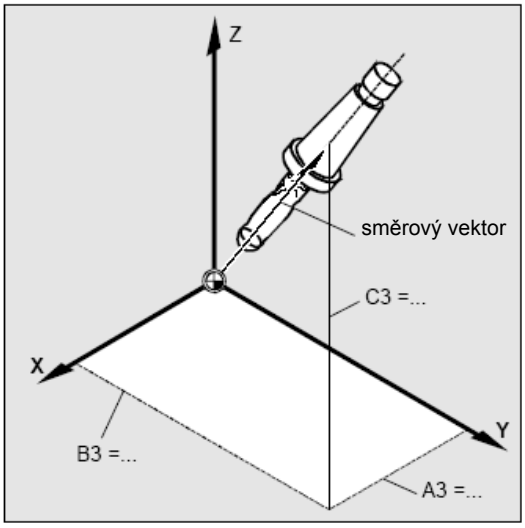


V protikladu k programování Eulerových úhlů mají zde všechny tři hodnoty vliv na vektor orientace.

Programování směrového vektoru

Složky směrového vektoru jsou programovány prostřednictvím A3, B3, C3. Vektor směřuje k upnutí nástroje; délka vektoru je přitom bezvýznamná.

Za složky vektoru, které nejsou programovány, je dosazena 0.



© Siemens AG 2004, Všechna práva vyhrazena

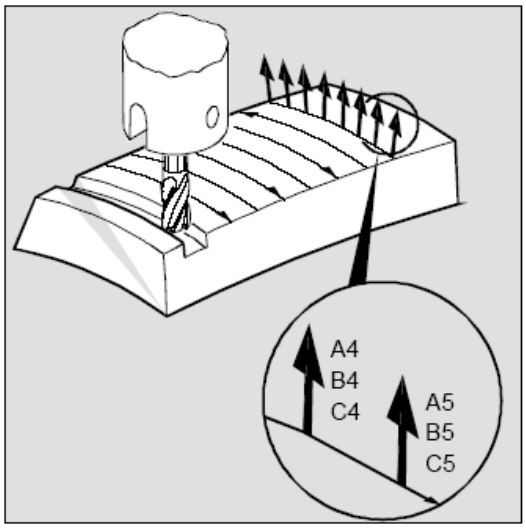
Čelní frézování

Čelní frézování s louží k opracování libovolně zakřivených ploch.

Pro tento druh 3D frézování potřebujete popis 3D drah na povrchu obrobku v řádcích.
Výpočty jsou při zohlednění tvaru nástroje a jeho rozměrů obvykle prováděny v CAM.
Vypočtené NC-bloky jsou potom prostřednictvím postprocesorů načteny do řídícího systému.

Popis ploch

Popis zakřivení dráhy probíhá prostřednictvím vektorů normály plochy s následujícími složkami:
A4, B4, C4 počáteční vektor na začátku bloku
A5, B5, C5 koncový vektor na konci bloku



Nachází-li se v bloku jen jeden počáteční vektor, zůstává vektor normály ploch v celém bloku konstantní.

Nachází-li se v bloku jen jeden konečný vektor, je interpolováno z konečné hodnoty předchozího bloku prostřednictvím interpolace hlavní kružnice k programované konečné hodnotě.

Jsou-li programovány počáteční a koncový vektor, je mezi oběma směry interpolováno rovněž prostřednictvím interpolace hlavní kružnice. Tak je možné vytvářet spojité a plynulé dráhy.

V základním postavení směřují vektory normály plochy – nezávisle na aktivní ploše G17 až G19 – ve směru osy Z.

Délka vektoru je bezvýznamná.

Za složky vektoru, které nejsou programovány, je dosazena 0.
Při aktivním ORIWKS (viz následující stránky) se vztahují vektory normály plochy k aktivnímu framu a při otáčení framu se otáčejí také.

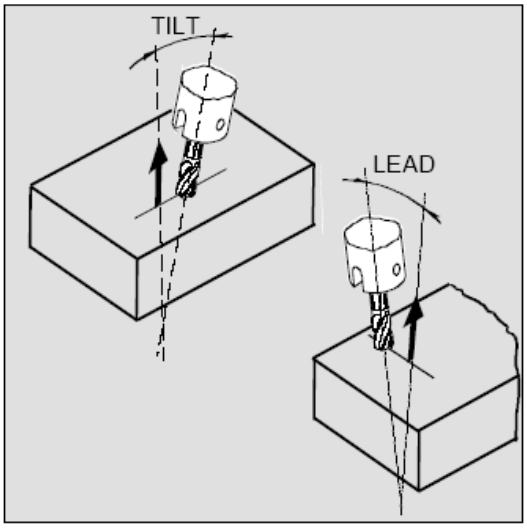
© Siemens AG 2004, Všechna práva vyhrazena

Vektor normály plochy musí být v rámci mezní hodnoty nastavitelné prostřednictvím strojního parametru kolmý k tečně dráhy, jinak je hlášen alarm.

Programování orientace nástroje pomocí LEAD a TILT

Výsledná orientace nástroje je zjištěna z:

- tečny dráhy
- vektoru normály plochy
- úhlu předstihu LEAD
- bočního úhlu TILT na konci bloku



Vysvětlení příkazů

LEAD	Úhel vztažený vůči vektoru normály plochy, v rovině definované tečnou dráhy a vektorem normály plochy
TILT	Úhel v rovině, kolmý k tečně dráhy relativní k vektoru normály plochy

Chování ve vnitřních rozích (u 3D nástrojů)

Je-li blok ve vnitřním rohu zkrácen, je výsledné orientace nástroje dosaženo rovněž na konci bloku.

© Siemens AG 2004, Všechna práva vyhrazena

7.1.2 Otočení orientace nástroje: ORIROTA, ORIROTR, ORIROTT

Programování

`N.. ORIROTA` nebo `N.. ORIROTR` nebo `N.. ORIROTT`	Stanovení interpolace vektoru otáčení
`THETA`	Aktivování otáčení vektoru orientace
`THETA=<hodnota>`	Úhel otáčení, který je dosažen na konci bloku
`THETA=`Θ_e	Úhel otáčení s konečným úhlem Θ_e vektoru otáčení
`THETA=AC(....)`	Blokové přepnutí na zadávání absolutních rozměrů
`THETA=IC(....)`	Blokové přepnutí na zadávání inkrement. rozměrů
`PO[THETA]=(`d_2`, `d_3`, `d_4`, `d_5`)`	Interpolace úhlu otáčení mnohočlenem 5. stupně

Vysvětlení příkazů

`ORIROTA`	Úhel otáčení k absolutně zadanému směru otáčení
`ORIROTR`	Úhel otáčení vzhledem k rovině mezi počáteční a konečnou orientací
`ORIROTT`	Úhel otáčení vzhledem ke změně vektoru orientace
`THETA`	Otáčení vektoru orientace
Θ_e	Konečný úhel vektoru otáčení jak absolutně prostřednictvím G90, tak relativně prostřednictvím G91 (zadávání inkrementálních kót) je aktivní
`PO[THETA]=(. . .)`	Mnohočlen pro úhel otáčení

Funkce

Má-li být u typů strojů s pohyblivým nástrojem měnitelná také orientace nástroje, je každý blok programován s konečnou orientací. V závislosti na kinematice stroje může být programován buď směr orientace os orientace nebo směr otáčení vektoru orientace THETA. Pro tyto vektory otáčení jsou programovatelné různé druhy interpolace:

- ORIROTA: úhel otáčení k absolutně zadanému směru otáčení.
- ORIROTR: úhel otáčení relativní k rovině mezi počáteční a konečnou orientací.
- ORIROTT: úhel otáčení relativní ke změně vektoru orientace

Orientace nástroje prostřednictvím os orientace je popsána v následujících kapitolách.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

ORIROTA
Úhel otáčení THETA je interpolován ve vztahu k absolutně stanovenému směru v prostoru. Nastavení základního směru otáčení probíhá prostřednictvím strojních parametrů.

ORIROTR
Úhel otáčení THETA je interpretován vzhledem k rovině, která je definována počáteční a konečnou orientací.

ORIROTT
Úhel otáčení THETA je interpretován vzhledem ke změně orientace. Pro THETA=0 je vektor otáčení interpolován tangenciálně ke změně orientace a odlišuje se od ORIROTR jen tehdy, když byl pro orientaci programován nejméně jeden mnohočlen pro „úhel nachýlení PSI“. Tak vzniká změna orientace, která neprobíhá v rovině. Prostřednictvím navíc programovaného úhlu otáčení THETA může pak být např. vektor otáčení interpolován tak, že tvoří vždy určitou hodnotu ke změně orientace.

Jen je-li aktivní druh interpolace ORIROTA, může být úhel otáčení nebo vektor otáčení naprogramován čtyřmi možnými následujícími způsoby:

1. přímo polohy kruhových os A,B,C
2. Eulerův úhel (ve stupních) prostřednictvím A2, B2, C2
3. RPY úhel (ve stupních) prostřednictvím A2, B2, C2
4. směrový vektor prostřednictvím A3, B3, C3 (úhel otáčení pomocí THETA=<hodnota>).

V případě, že ORIROTR nebo ORIROTT jsou aktivní, může být úhel otáčení programován jen přímo prostřednictvím THETA.

Otáčení může být programováno i samotné v bloku, aniž se změna orientace koná. Přitom nemají ORIROTR a ORIROTT žádný význam. V tomto případě je úhel otáčení interpretován vždy ve vztahu k absolutnímu směru (ORIROTA).

© Siemens AG 2004, Všechna práva vyhrazena

7.1.3 Vztažný systém os orientace – ORIWKS, ORIMKS

Programování

```
N.. ORIMKS=
```
nebo
```
N..ORIWKS=
```

Vysvětlení příkazů

ORIMKS	Otáčení v souřadném systému stroje
ORIWKS	Otáčení v souřadném systému obrobku

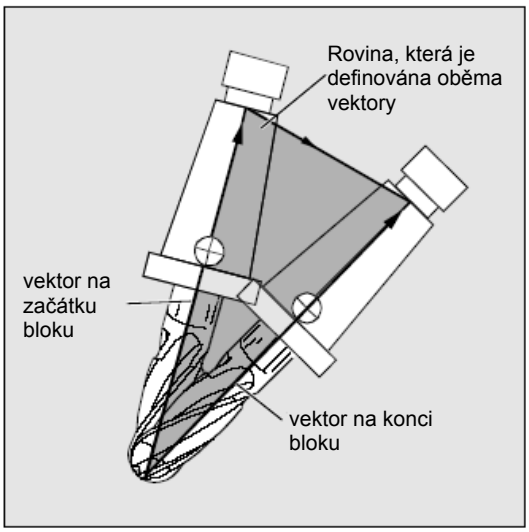
Funkce

Při programování orientace v souřadném systému obrobku prostřednictvím Eulerových nebo RPY úhlů nebo vektoru orientace může být nastaven průběh pohybu otáčení pomocí příkazů ORIMKS/ORIWKS.

Postup

U ORIMKS je provedený pohyb nástroje závislý na kinematice stroje. Při změně orientace s hrotem nástroje pevně v prostoru se mezi polohami kruhových os provádí lineární interpolace.

U ORIWKS je pohyb nástroje nezávislý na kinematice nástroje. Při změně orientace s hrotem nástroje pevným v prostoru se nástroj pohybuje v rovině, definované počátečním a konečným vektorem.



Další upozornění

ORIWKS je základní nastavení. Není-li při 5 osovém programu od začátku jasné, na kterém stroji má proběhnout, je zásadně třeba volit ORIWKS. Které pohyby stroj skutečně provede, závisí na jeho kinematice.

© Siemens AG 2004, Všechna práva vyhrazena

Prostřednictvím ORIMKS můžete programovat skutečné pohyby stroje, abyste např. zabránili kolizi se zařízeními apod.

Prostřednictvím strojního parametru $MC_ORI_IPO_WITH_G_CODE je stanoveno, který druh interpolace je v platnosti: ORIMKS/ORIWKS nebo ORIMACHAX/ORIVIRTAX (viz kapitola 7.1.4).

7.1.4 Singulární postavení a jejich úprava

Upozornění týkající se ORIWKS:
Pohyby orientace v oblasti singulárního postavení stroje s 5 osami vyžadují velké pohyby os stroje. (Například jsou u revolverové kyvné hlavy s osou C jako osou otáčení a osou A jako kyvnou osou všechna postavení s A = 0 singulární.)

Aby osy stroje nebyly přetěžovány, regulace rychlosti v blízkosti singulárních míst silně sníží rychlost dráhy.

Prostřednictvím strojních parametrů
$MC_TRAFO5_NON_POLE_LIMIT
$MC_TRAFO5_POLE_LIMIT
mohou být parametry transformace stanoveny tak, že pohyby orientace v blízkosti pólu jsou prostřednictvím pólu složeny tak, že je možné plynulé opracování.

Upozornění k SW 5.2:
Od SW 5.2 jsou singulární místa spravována jen strojním parametrem
$MC_TRAFO5_POLE_LIMIT.
(viz Popis funkcí, Díl 3, odstavec 2.8.4).

© Siemens AG 2004, Všechna práva vyhrazena

7.1.5 Osy orientace (od SW 5.2) ORIAXES, ORIVECT, ORIEULER, ORIRPY

Programování

```
N.. ORIAXES nebo ORIVECT                ; interpolace orientace:
N.. G1 X Y Z A B C                      ; interpolace lineární nebo i. hlavní kružnicí
N.. ORIEULER nebo ORIRPY                ; programování orientace:
N.. G1 X Y Z A2= B2= C2=                ; úhel orientace pomocí Eulerova/RPY úhlu
nebo
N.. ORIVIRT1 nebo ORIVIRT2              ; úhel orientace prostřednictvím virtuálních
N.. G1 X Y Z A3= B3= C3=                 ; os orientace

N.. ORIPLANE                            ; interpolace orientace (od SW 5.3)
N.. ORICONCW nebo ORICONCCW             ; rozšířená interpolace (od SW 6.1)
N.. ORICONTO nebo ORICURVE              ; na ploše kuželového pláště
N.. G1 X Y Z A6= B6= C6=
N.. ORICONIO
N.. G1 X Y Z A7= B7= C7=
```

Vysvětlení příkazů

ORIAXES	Lineární interpolace os stroje nebo os orientace
ORIVECT	Interpolace hlavní kružnice (identická s ORIPLANE)
ORIMKS	Otáčení v souřadného systému stroje
ORIWKS	Otáčení v souřadném systému obrobku
	Popis viz kapitola 7.1.2
G1 X Y Z A B C	Programování poloh os stroje
ORIEULER	Programování orientace prostřednictvím Eulerova úhlu
ORIRPY	Programování orientace prostřednictvím RPY úhlu
G1 X Y Z A2= B2= C2=	Programování úhlu virtuálních os
	Programování orientace prostřednictvím virtuálních os orientace
ORIVIRT1	(definice 1), stanovení podle MD $MC_ORIAX_TURN_TAB_1
ORIVIRT2	(definice 2), stanovení podle MD $MC_ORIAX_TURN_TAB_2
G1 X Y Z A3= B3= C3=	Programování směrového vektoru směrové osy
	Interpolace orientace (od SW 5.3)
ORIPLANE	Interpolace v rovině (interpolace hlavní kružnice)
ORICONCW	Interpolace na ploše pláště kužele ve směru hodinových ručiček
ORICONCCW	Interpolace na ploše pláště kužele proti směru hodinových ručiček
ORICONTO	Interpolace na ploše pláště kužele tangenciální přechod
G1 X Y Z A6= B6= C6=	Programování osy otáčení kužele (normovaný vektor)
ORICONIO	Interpolace na ploše kužele se zadáním meziorientace
G1 X Y Z A7= B7= C7=	(programování jako normovaný vektor)
ORICURVE	Interpolace orientace se zadáním pohybu dvou kontaktních bodů
XH YH ZH	nástroje. Kromě příslušných konečných bodů jsou navíc
s mnohočleny	programovatelné mnohočleny prostorových křivek.
PO[XH] = (x2, x3..)	

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Funkce os orientace popisuje orientaci nástroje v prostoru. Tím je zaveden další třetí stupeň volnosti, který popisuje otáčení kolem sebe sama. Toto je nutné pro 6 osovou transformaci.

Prostřednictvím MD `$MC_ORI_DEF_WITH_G_CODE` je stanoveno, jak jsou definovány programované úhly A2, B2, C2:
Definice proběhne podle G_skupiny 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Prostřednictvím MD `$MC_ORI_IPO_WITH_G_CODE` je stanoveno, který druh interpolace je v platnosti: ORIWKS/ORIMKS nebo ORIAXES/ORIVECT.

Provozní režim JOG

Úhly orientace jsou v tomto druhu provozu vždy interpolovány lineárně. Při kontinuálním a inkrementálním postupu prostřednictvím tlačítek posuvu může být postupováno jen v jedné ose orientace.
Prostřednictvím ručních koleček lze postupovat současně ve více osách orientace současně.

Pro ruční postup po osách orientace je účinný spínač korekce posuvu vpřed specifický pro kanál popř. spínač korektury spěšného chodu při překrytí spěšného chodu.

Prostřednictvím následujících strojních parametrů je možné oddělené zadání rychlosti:
`$MC_JOG_VELO_RAPID_GEO`
`$MC_JOG_VELO_GEO`
`$MC_JOG_VELO_RAPID_ORI`
`$MC_JOG_VELO_ORI`

od SW 6.3
Funkce kartézský ruční posuv může být v režimu JOG u systému SINUMERIK 840D nastavena odděleně pomocí modulu „Správa transformací“ a u systému SINUMERIK 810D powerline od SW 6.1
Posunutí geometrických os ve vztažných systémech MKS, WKS a TKS.

© Siemens AG 2004, Všechna práva vyhrazena

Programování posuvu

FORI1	Posuv pro vychýlení vektoru orientace na hlavní kružnici
FORI2	Posuv pro superponované otáčení kolem vychýleného vektoru orientace

Programovatelný posuv odpovídá u pohybů orientace úhlové rychlosti [stupeň/min].

Platnost posuvu prostřednictvím G-kódu:
Při programování ORIAXES může být posuv pro osu orientace omezen příkazem FL[].

Při programování ORIVECT musí být posuv programován pomocí FORI1 nebo FORI2. V NC-bloku smí být FORI1 a FORI2 programováno jen jednou. Při tomto programování se postupuje vždy po nejkratší dráze.
Při superponovaném pohybu otáčení a vychýlení se uplatňuje vždy nejmenší posuv. Posuv odpovídá u pohybů orientace úhlové rychlosti [stupeň/min].

Pohybují-li se geometrické osy a osy orientace po společné dráze, je pohyb tohoto posuvu určen z nejmenšího posuvu vpřed.

Rozšířená interpolace od SW 6.1
Prostřednictvím rozšířené orientace je možné provádět změny orientace podél plochy pláště kužele nacházející se v prostoru.
Je nutno naprogramovat následující údaje:

- orientace počátku pomocí předchozího bloku
- konečná orientace buď pomocí směrového vektoru s A3, B3, C3 nebo v Eulerově úhlu popř. RPY úhlu prostřednictvím A2, B2, C2
- osa otáčení kužele jako vektor s A6, B6, C6
- úhel rozevření PSI s identifikátorem NUT
- pomocné orientace s A7, B7, C7

Odkazy na literaturu:
SINUMERIK 840D/FM-NC Popis funkcí (díl 3),
„Modul Správa transformací“.
/FB/ F2, 3 až 5 osové transformace

© Siemens AG 2004, Všechna práva vyhrazena

7.1.6 Kartézský pohyb PTP (od SW 5.2)

Programování

```
N.. TRAORI
N.. STAT=`B10` TU=`B100` PTP
N.. CP
```

Vysvětlení příkazů

PTP	**P**oint **t**o **P**oint (pohyb od bodu k bodu) Pohyb je proveden jako pohyb synchronních os, nejpomalejší na pohybu se podílející osa je dominující osou pro rychlost.
CP	**c**ontinuous **p**ath (pohyb po dráze) Pohyb je proveden jako kartézský pohyb po dráze.
STAT =	Postavení kloubů; hodnota je závislá na transformaci.
TU=	Informace o otočení Tím je možné jednoznačně najet na úhel osy mezi –360°a+360°.

Funkce

Pomocí této funkce může být naprogramována poloha v kartézském systému souřadnic, pohyb probíhá však osách stroje.
Funkce může být použita např. při změně postavení kloubu, když přitom pohyb vede singularitou.

Upozornění:

Funkce má smysl jen ve spojení s aktivní transformací. Nadále je „pohyb PTP“ přípustný jen ve spojení s G0 a G1.

Postup

Přepínání mezi kartézským pohybem a pohybem os stroje probíhá prostřednictvím příkazů PTP a CP, které mají modální platnost. CP je standardní nastavení.

© Siemens AG 2004, Všechna práva vyhrazena

Programování postavení (STAT=)

Postavení stroje není prostřednictvím zadání polohy kartézskými souřadnicemi a orientace nástroje jednoznačně určeno. Podle toho, o jakou kinematiku se jedná, existuje až 8 odlišných resp. rozlišujících postavení kloubu. Tyto jsou tak z hlediska transformace specifické. Aby bylo možné kartézskou polohu jednoznačně přepočítat na úhly os, musí být postavení kloubů udáno prostřednictvím příkazu STAT=. Příkaz „STAT“ obsahuje jako binární hodnotu, v níž je pro každé z možných postavení jeden bit.

Odkazy na literaturu:

Rozdílné transformace jsou obsaženy v příručce: SINUMERIK 840D/FM-NC Popis funkcí (díl 3), „Modul Správa transformací“.

Bity postavení, které je třeba naprogramovat v příkazu „STAT“, jsou obsaženy v příručce: SINUMERIK 840D/FM-NC Popis funkcí (díl 3), „3- až 5-osové transformace“.

Programování úhlů os (TU=)

Aby bylo možné jednoznačně najet na úhel osy $< \pm$ 360 stupňů, musí být tato informace naprogramována prostřednictvím příkazu „TU=“. Tento příkaz má blokovou platnost.

Osy se pohybují po nejkratší dráze:

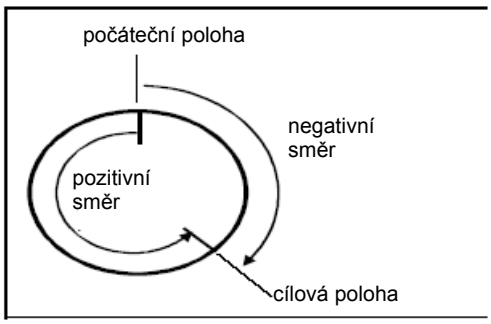
- když u polohy není programován příkaz TU
- u os, které mají rozsah pohybu $> \pm$ 360 stupňů

Příklad:

Na obrázku uvedená cílová poloha může být dosažena v negativním nebo pozitivním směru. V adrese A1 je směr programován.

A1=225°, TU=bit 0, → pozitivní směr

A1=-135°, TU=bit 1, → negativní směr



© Siemens AG 2004, Všechna práva vyhrazena

Zaoblení mezi pohyby CP a PTP
Mezi bloky je prostřednictvím G641 možné zadat zaoblení přechodu. Velikost oblasti zaoblení je dráha v mm nebo palcích, od které, resp. ke které má být přechod mezi bloky zaoblen. Velikost je třeba zadat následovně:

- pro G0 bloky prostřednictvím ADISPOS
- pro všechny ostatní příkazy dráhy pomocí příkazu ADIS

Výpočet dráhy odpovídá zohlednění F-adres u bloků bez G0. Posuv je dodržen v osách uvedených v FGROUP(..).

Výpočet posuvu:
Pro CP-bloky jsou pro výpočet použity kartézské osy základního souřadného systému.
Pro PTP-bloky jsou pro výpočet použity odpovídající osy souřadného systému stroje.

Další upozornění

Změna provozního režimu
Funkce „Kartézský pohyb PTP“ má smysl jen v provozních režimech AUTO a MDA. Při změně provozního režimu na JOG zůstane zachováno aktuální nastavení.
Je-li nastaven G-kód PTP, osy se pohybují v souřadném systému stroje. Je-li nastaven G-kód CP, osy se pohybují v souřadném systému obrobku.

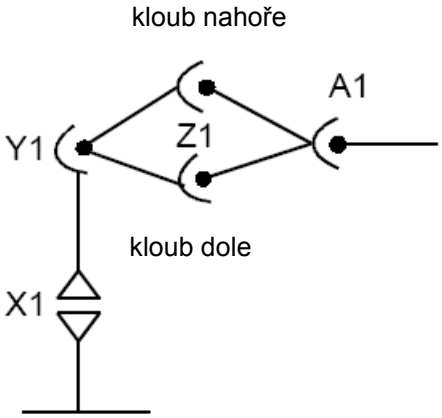
Power On/Reset
Po Power On nebo po resetu je nastavení závislé na strojním parametru
`$MC_GCODE_RESET_VALUES[48]`.
Standardně je nastaven druh pohybu „CP“.

Příkaz Repos
Jestliže byla během bloku přerušení nastavena funkce „Kartézský pohyb PTP“, také návrat zpět do původní polohy se děje prostřednictvím PTP.

© Siemens AG 2004, Všechna práva vyhrazena

Superponované pohyby
Posunutí ručním kolečkem nebo externí posunutí počátku jsou možné při kartézském pohybu PTP jen omezeně. Při změně pohybu PTP na CP pohyb se v základním souřadném systému nesmí vyskytovat žádné superpozice.

Příklad programování



`N10 G0 X0 Y-30 Z60 A-30 F10000`	Výchozí postavení → kloub nahoře
`N20 TRAORI(1)`	Transformace zapnuta
`N30 X1000 Y0 Z400 A0`	
`N40 X1000 Z500 A0 STAT=´B10´ TU=´B100´ PTP`	Změna orientace bez transformace → kloub dole
`N50 X1200 Z400 CP`	Transformace opět aktivní
`N60 X1000 Z500 A20`	
`N70 M30`	

© Siemens AG 2004, Všechna práva vyhrazena

7.1.7 On-line změna korekce nástroje: TOFFON, TOFFOF (od SW 6.4)

Programování

```
N.. TRAORI
N.. TOFFON (X, 25)
N.. WHEN TRUE DO $AA_TOFF[X]
```

Vysvětlení příkazů

TOFFON	Tool Offset ON (aktivovat on-line korekci délky nástroje) Při aktivování může být pro odpovídající směr korekce zadána offsetová hodnota, na kterou se ihned najíždí.
TOFFOF	Tool Offset OF (zrušit on-line korekci délky nástroje) Odpovídající hodnoty korekce jsou vynulovány a dojde k zastavení předběžného zpracování.
X, Y, Z,	Směr korekce pro zadanou hodnotu offsetovou hodnotu

Upozornění:
On-line korekce délky nástroje je volitelný doplněk, který napřed musí být uvolněn. Jen ve spojení s aktivní transformací orientace nebo aktivním orientovatelným držákem nástroje má tato funkce smysl.

Funkce

Pomocí systémové proměnné $AA_TOFF[] je možné provádět trojrozměrnou superpozici efektivní délky nástroje ve třech směrech nástroje v reálném čase.
Jako index jsou použity identifikátory tří geometrických os. Tím je stanoven počet aktivních směrů korekce prostřednictvím v tomtéž čase aktivních geometrických os.
Všechny korekce mohou být aktivní současně.

Použití
Funkce on-line korekce délky nástroje je použitelná při

- transformaci orientace TRAORI
- orientovatelném držáku nástroje TCARR

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Příprava bloku

Při přípravě bloku v předběžném zpracování je aktuální korekce délky nástroje uplatňovaná v hlavním zpracování zohledněna také. Aby bylo možno využít maximální přípustné osové rychlosti, je nutné zastavit předběžné zpracování příkazem STOPRE, zatímco je vytvářena korekce nástroje.
Korekce nástroje je v okamžiku předběžného zpracování také známá vždy tehdy, když nejsou korekce délky nástroje po spuštění programu měněny nebo když po změně korekce délky nástroje bylo zpracováno více bloků, než může vyrovnávací paměť IPO mezi předběžným a hlavním zpracováním pojmout.

Proměnná $AA-TOFF_PREP_DIFF

Rozměr diference mezi aktuální korekcí uplatňovanou v interpolátoru a korekcí, která byla v platnosti v okamžiku přípravy bloku, může být zjištěn z proměnné $AA-TOFF_PREP_DIFF[].

Nastavení strojních parametrů a nastavovaných parametrů

Pro on-line korekci délky nástroje jsou k dispozici následující strojní parametry:

- MD 20610: ADD_MOVE_ACCEL_RESERVE Rezerva při plánování rychlosti
- MD 21190: TOFF_MODE Obsah systémové proměnné $AA_TOFF[] je výstupem jako absolutní hodnota nebo je integrován.
- MD 21194: TOFF_VELO Rychlost on-line korekce délky nástroje.
- MD 21196: TOFF_ACCEL Zrychlení on-line korekce délky nástroje
- Nastavovaný parametr pro zadání mezních hodnot SD 42970: TOFF_LIMIT Horní mez hodnoty korekce délky nástroje

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Volba korekce délky nástroje

```
MD 21190: TOFF_MODE =1                    ; najíždění na absolutní hodnoty
MD 21194: TOFF_VELO[0] =1000
MD 21196: TOFF_VELO[1] =1000
MD 21194: TOFF_VELO[2] =1000
MD 21196: TOFF_ACCEL[0] =1
MD 21196: TOFF_ACCEL[1] =1
MD 21196: TOFF_ACCEL[2] =1

N5 DEF REAL XOFFSET
N10 TRAORI(1)                             ; transformace zapnuta
N20 TOFFON(Z)                             ; aktivování on-line korekce délky nástroje
                                          ; pro směr nástroje Z
N30 WHEN TRUE DO $AA_TOFF[Z] = 10         ; pro směr nástroje Z je interpolována
    G4 F5                                 ; korekce délky nástroje 10
...
N40 TOFFON(X)                             ; aktivování on-line korekce délky nástroje
                                          ; pro směr nástroje X
N50 ID=1 DO $AA_TOFF[X] = $AA_IW[X2]      ; pro směr nástroje X je provedena korekce
    G4 F5                                 ; závislá na poloze osy X2
...
N100 XOFFSET = $AA_TOFF_VAL[X]
N120 TOFFON(X, -XOFFSET)                  ; přiřazení aktuální korekce ve směru X
     G4 F5                                ; pro směr nástroje X je korekce délky
                                          ; nástroje vrácena zpět na 0
```

Zrušení volby korekce délky nástroje

```
N10 TRAORI(1)                             ; transformace zapnuta
N20 TOFFON(X)                             ; aktivování směru nástroje Z
N30 WHEN TRUE DO $AA_TOFF[X] = 10         ; pro směr nástroje X je interpolována
G4 F5                                     ; korekce délky nástroje 10
...
N80 TOFFOF(X)                             ; offset polohy směru nástroje X
 ; No axis is traversed                   ; je vymazán: …$AA_TOFF[X]=0
 ; To the current position in WCS, the    ; Na aktuální pozici v souřadném systému
 ; positional offset is added in accordance ; obrobku nebude najíždět žádná osa, offset
 ; with the current orientation           ; polohy se přičte podle aktuální orientace.
```

Literatura

/FB/ F2, 3- až 5-osové transformace

© Siemens AG 2004, Všechna práva vyhrazena

7.2 Frézování na rotačních součástech: TRANSMIT

Programování

```
TRANSMIT nebo TRANSMIT (n)
TRAFOOF
```

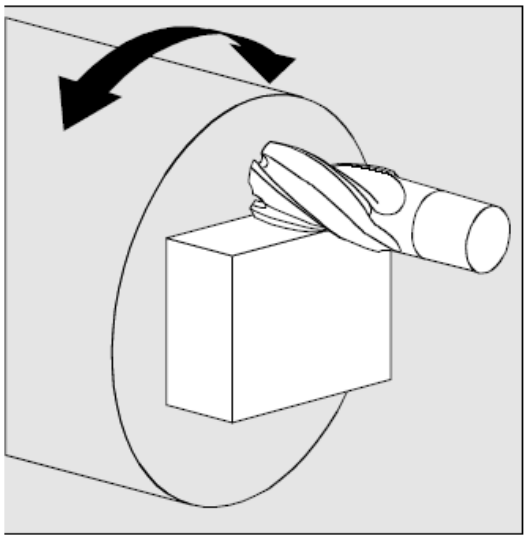
Vysvětlení příkazů

`TRANSMIT`	Aktivuje první dohodnutou funkci TRANSMIT
`TRANSMIT (n)`	Aktivuje n-tou dohodnutou funkci TRANSMIT; n smí být maximálně 2 (TRANSMIT(1) odpovídá TRANSMIT)
`TRAFOOF`	Vypne aktivní transformaci

Aktivní transformace TRANSMIT je vypnuta také tehdy, když je v příslušném kanálu aktivována některá z ostatních transformací (např. TRACYL, TRAANG, TRAORI).

Funkce TRANSMIT umožňuje následující výkony:

- Čelní opracování soustružených dílů s rotujícím upnutím (vrtání, kontury).
- Pro programování těchto opracování může být použit kartézský systém souřadnic.
- Řízení transformuje programované pohyby kartézského systému souřadnic na pohyby reálných os stroje (standardní případ):
  – kruhová osa
  – osa přistavení kolmá k ose otáčení
  – podélná osa paralelní k ose otáčení
  Lineární osy jsou na sebe kolmé.
- Přesazení středu nástroje relativně vůči středu rotace je přípustné.
- Regulace rychlosti zohlední omezení, definovaná pro rotační pohyby.



© Siemens AG 2004, Všechna práva vyhrazena

Kruhová osa
Kruhová osa nemůže být programována, protože je obsazena geometrickou osou a tím není jako osa kanálu přímo programovatelná.

Pól
Do **SW 3.x**
Projetí přes pól (počátek kartézského souřadného systému) je nepřípustné. Přes pól vedoucí pohyb je zastaven na pólu a je hlášen alarm. Při přesazení středu frézy zůstane pohyb stát na okraji oblasti, do které není možno vjet.

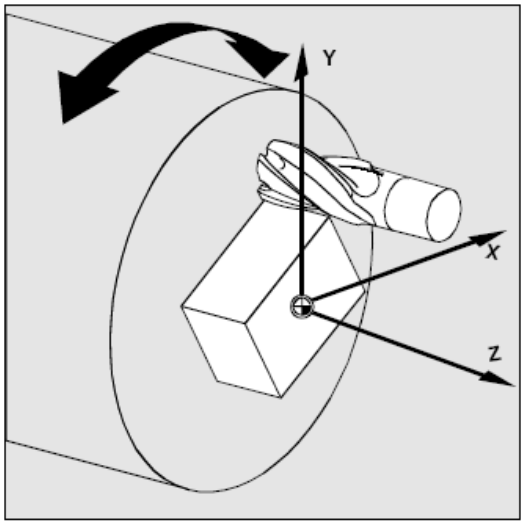
Od **SW 4**
Pro projetí pólu jsou 2 možnosti:
1. Pohyb lineární osy samotné
2. Pohyb na pól s otočením kruhové osy v pólu a pohyb z pólu

Výběr probíhá prostřednictvím SD 24911 a 24951.

Literatura
/FB/ M1 Kinematické transformace

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování



N10 T1 D1 G54 G17 G90 F5000 G94	Zvolení nástroje
N20 G0 X20 Z10 SPOS=45	Najíždění do výchozího postavení
N30 TRANSMIT	Aktivování funkce TRANSMIT
N40 ROT RPL=-45 N50 ATRANS X-2 Y10	Nastavení framu
N60 G1 X10 Y-10 G41 OFFN=1 N70 X-10 N80 Y10 N90 X10 N100 Y-10	Obrábění čtyřhranu nahrubo; přídavek rozměru na opracování načisto 1 mm
N110 G0 Z20 G40 OFFN=0 N120 T2 D1 X15 Y-15 N130 Z10 G41	Výměna nástroje
N140 G1 X10 Y-10 N150 X-10 N160 Y10 N170 X10 N180 Y-10	Obrábění čtyřhranu načisto
N190 Z20 G40 N200 TRANS N210 TRAFOOF	Zrušení volby framu
N220 G0 X20 Z10 SPOS=45	Pohyb do výchozího postavení
N230 M30	

© Siemens AG 2004, Všechna práva vyhrazena

7.3 Transformace pláště válce: TRACYL

Programování

`TRACYL (d)` nebo `TRACYL (d,t)`
`TRAFOOF`

Vysvětlení příkazů

`TRACYL (d)`	Aktivuje první, ve strojních parametrech kanálu definovanou funkci TRACYL, d je parametr pro pracovní průměr
`TRACYL (d,n)`	Aktivuje n-tou, ve strojních parametrech kanálu definovanou funkci TRACYL, n smí být maximálně 2, TRACYL(d,1) odpovídá TRACYL(d)
`d`	Hodnota pro pracovní průměr. Pracovní průměr je dvojnásobná vzdálenost mezi špičkou nástroje a středem otáčení.
`TRAFOOF`	Transformace vypnuta (základní souřadný systém a souřadný systém stroje jsou opět identické)
`OFFN`	Offset normály kontury: vzdálenost stěny drážky od naprogramované vztažné kontury

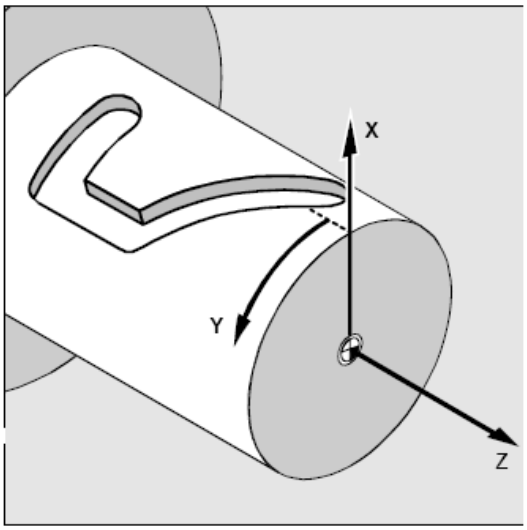
Aktivní transformace TRACYL je vypnuta rovněž tehdy, když je v příslušném kanálu aktivována některá z ostatních transformací (např. TRANSMIT, TRAANG, TRAORI).

Funkce

Transformace křivek na plášti válce TRACYL
Transformace křivek na plášti válce TRACYL umožňuje následující výkony:

Opracování
- podélných drážek na válcovitých tělesech,
- příčných drážek na válcovitých tělesech,
- libovolně probíhajících drážek na válcovitých tělesech.

Průběh drážky je programován ve vztahu k rozvinuté, rovinné ploše pláště kužele.



Systém souřadnic obrobku

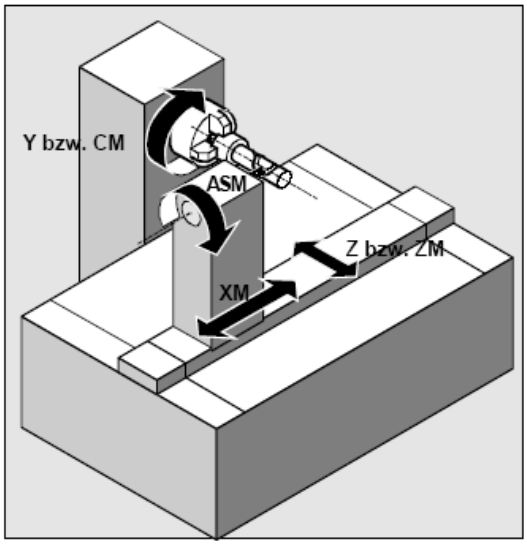
© Siemens AG 2004, Všechna práva vyhrazena

Transformace souřadnic pláště válce se vyskytuje ve dvou variantách:

- bez korekce stěny drážky (TRAFO_TYPE_n=512)
- s korekcí stěny drážky (TRAFO_TYPE_n=513)

Bez korekce stěny drážky:
Řídící systém transformuje programované pohyby souřadného systému válce na pohyby reálných os stroje:
- kruhová osa
- osa přísuvu kolmá k ose otáčení
- podélná osa paralelní k ose otáčení

Lineární osy jsou k sobě kolmé. Osa přísuvu protíná kruhovou osu.

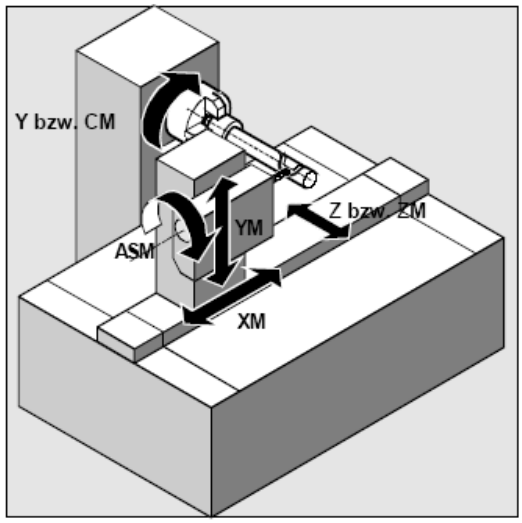


Souřadný systém stroje (bzw. = resp.)

S korekcí stěny drážky:
Kinematika jako výše uvedeno, ale navíc
- podélná osa paralelní ke směru obvodu

lineární osy jsou k sobě kolmé.

Regulace rychlosti zohlední omezení definovaná pro rotační pohyby.



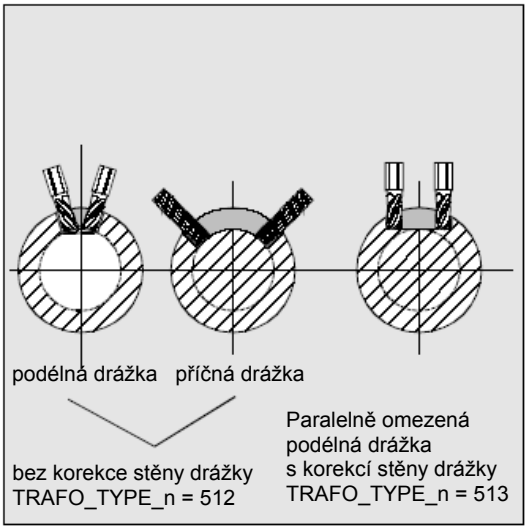
Souřadný systém stroje (bzw. = resp.)

© Siemens AG 2004, Všechna práva vyhrazena

Průřez drážky

Při konfiguraci os 1 jsou drážky podélné ke kruhové ose paralelně omezeny, když šířka drážky odpovídá přesně rádiusu nástroje.

Drážky paralelní k obvodu (příčné drážky) nejsou na začátku a na konci paralelní.



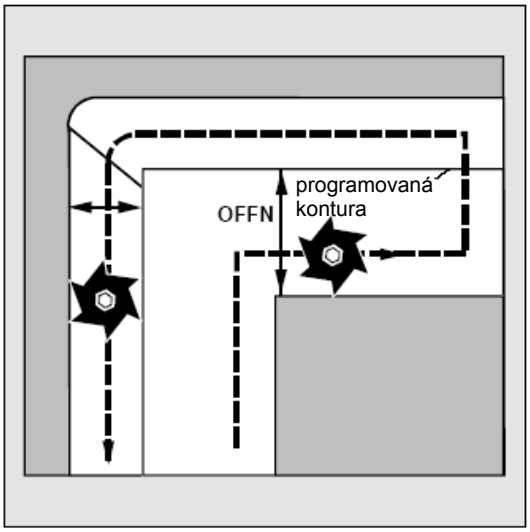
Offset normály kontury OFFN (513)

K frézování drážek prostřednictvím TRACYL je

- ve výrobním programu programována dráha středu drážky
- prostřednictvím OFFN je programována **poloviční šířka drážky**.

OFFN je v platnosti až se zvolenou korekcí rádiusu nástroje, aby bylo zabráněno poškození stěny drážky.

Dále by mělo platit OFFN>=rádius nástroje, aby bylo vyloučeno poškození protilehlé stěny drážky.



Výrobní program pro frézování drážky se zpravidla skládá z následujících kroků:

1. Volba nástroje
2. Aktivování funkce TRACYL
3. Volba vhodného předpisu souřadnic (FRAME)
4. Nastavení polohy
5. Naprogramování OFFN
6. Aktivování korekce radiusu nástroje
7. Blok najetí (najetí na korekci rádiusu nástroje a najetí na stěnu drážky)
8. Kontura středu drážky
9. Deaktivování korekce rádiusu nástroje
10. Blok odjetí (odjetí od korekce rádiusu nástroje a odjetí od stěny drážky)
11. Nastavení polohy
12. TRAFOOF
13. Aktivování opět původního předpisu souřadnic (FRAME)

© Siemens AG 2004, Všechna práva vyhrazena

Zvláštnosti:

- Volba korekce rádiusu nástroje
  Korekce rádiusu nástroje není programována vzhledem ke stěně drážky, nýbrž vůči naprogramované středové dráze drážky. Aby se nástroj pohyboval vlevo od stěny drážky, je zadáno G42 (místo G41). Zamezíte tomu, když je v OFFN zaznamenána šířka drážky se záporným znaménkem.
- OFFN s TRACYL se projeví jinak než bez TRACYL. Jelikož při aktivní korekci rádiusu nástroje je OFFN započteno i bez TRACYL, mělo by být OFFN pro TRAFOOF znovu vráceno na nulu.
- Změna OFFN v rámci výrobního programu je možná. Tím by mohla být středová dráha drážky posunuta ze středu (viz obrázek).
- Vodicí drážky:
  Pomocí příkazu TRACYL není u vodicích drážek vyrobena stejná drážka jako s nástrojem, jehož průměr odpovídá šířce drážky. Principielně není možné pomocí menšího válcového nástroje vytvořit stejnou geometrii stěny drážky jako s nástrojem větším.
  Příkaz TRACYL minimalizuje chybu. Aby se předešlo problémům s přesností, měl by být rádius nástroje jen o málo menší než je polovina šířky drážky.

Upozornění

OFFN a korekce rádiusu nástroje

- Při TRAFO_TYPE_n = 512 působí hodnota pod OFFN jako přídavek rozměru pro opracování načisto ke korekci rádiusu nástroje.
- Při TRAFO_TYPE_n = 513 je v OFFN programována poloviční šířka drážky. Kontura je projeta s OFFN-korekce rádiusu nástroje.

© Siemens AG 2004, Všechna práva vyhrazena

Při transformaci křivky na plášti válce s korekcí stěny drážky by měla být pro korekci použitá osa na nule (y = 0), aby byla drážka zhotovena středově vůči programované střední linii drážky.

Kruhová osa
Kruhová osa nemůže být programována, jelikož je obsazena geometrickou osou a tak není jako osa kanálu přímo programovatelná.

Použití os
Následující osy nemohou být použity jako polohovací osy, resp. osy kyvného pohybu:

- Geometrická osa ve směru obvodu plochy pláště válce (osa Y)
- Dodatečná lineární osa při korekci stěny drážky (osa Z)

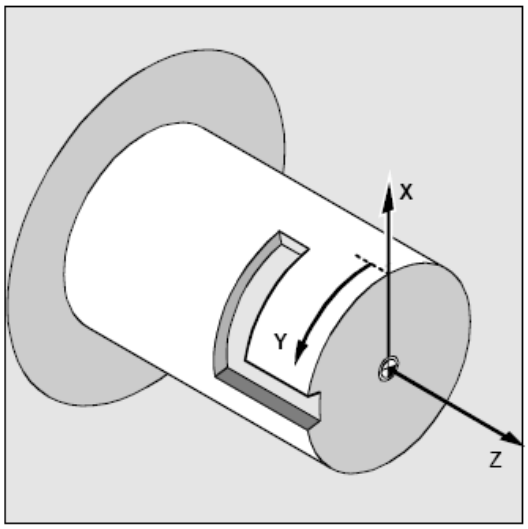
- Definice nástroje

Následující příklad je vhodný pro testování nastavení parametrů transformace válce TRACYL:

Parametr nástroje Číslo (DP)	**Význam**	**Poznámka**
`$TC_DP1[1,1]=120`	Typ nástroje	Fréza
`$TC_DP2[1,1]=0`	Délka břitu	Jen pro rotační nástroje
Geometrie	**Korekce délky**	
`$TC_DP3[1,1]=8.`	Vektor korekce délky	Výpočet podle typu
`$TC_DP4[1,1]=9.`		a roviny
`$TC_DP5[1,1]=7.`		
Geometrie	**Rádius**	
`$TC_DP6[1,1]=6.`	Rádius	Rádius nástroje
`$TC_DP7[1,1]=0`	Šířka drážky b pro drážkovací pilu, rádius zaoblení pro frézovací nástroje	
`$TC_DP8[1,1]=0`	Přesah k	Jen pro drážkovací pilu
`$TC_DP9[1,1]=0`		
`$TC_DP10[1,1]=0`		
`$TC_DP11[1,1]=0`	Úhel pro kuželovité frézovací nástroje	
Opotřebení	**Korekce délky a rádiusu**	
`$TC_DP12[1,1]=0`	Ostatní parametry do $TC_DP24=0	Základní míra/adaptér

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování



N10 T1 D1 G54 G90 F5000 G94	Volba nástroje, kompenzace upnutí
N20 SPOS=0 N30 G0 X25 Y0 Z105 CC=200	Zaujetí výchozího postavení
N40 TRACYL (40)	Aktivování transformace křivky na plášti válce
N50 G19	Volba roviny

Zhotovení hákovité drážky:

N60 G1	Přísuv nástroje na dno drážky
N70 OFFN=12	Stanovení vzdálenosti stěny drážky 12 mm vzhledem ke středové linii drážky
N80 G1 Z100 G42	Najetí na pravou stranu drážky
N90 G1 Z50	Úsek drážky paralelní k ose válce
N100 G1 Y10	Úsek drážky paralelní k obvodu
N110 OFFN=4 G42	Najetí na levou stěnu drážky; stanovení vzdálenosti stěny drážky 4 mm vzhledem ke středové linii drážky
N120 G1 Y70	Úsek drážky paralelní k obvodu
N130 G1 Z100	Úsek drážky paralelní k ose válce
N140 G1 Z105 G40	Odjetí od stěny válce
N150 G1 X25	Volný pohyb
N160 TRAFOOF	
N170 G0 X25 Y0 Z105 CC=200	Zaujetí výchozího postavení
N180 M30	

© Siemens AG 2004, Všechna práva vyhrazena

7.4 Šikmá osa TRAANG

Programování

`TRAAANG (α)` nebo `TRAANG (α,n)`
`TRAFOOF`

Vysvětlení příkazů

`TRAANG`	Je-li úhel α vynechán nebo je-li zapsána nula, je aktivována transformace s parametry předchozí volby. Při prvním aktivování platí předběžné nastavení podle strojních parametrů.
`TRAANG α`	Aktivuje první specifikovanou transformaci šikmé osy.
`TRAANG (α,n)`	Aktivuje první specifikovanou transformaci šikmé osy, n smí být maximálně 2. TRAANG (α,1) odpovídá TRAANG (α)
α	Úhel šikmé osy
`TRAFOOF`	Vypnutí transformace

Je-li úhel α vynechán nebo je-li zapsána nula, je aktivována transformace s parametry předchozí volby. Při prvním aktivování se použije předběžné dosazení strojních parametrů. (chování do SW stavu < 6.4, nové stavy viz níže).

Aktivní transformace TRAANG je vypnuta rovněž tehdy, když je v příslušném kanálu aktivována jedna z ostatních transformací.
(např. TRACYL, TRANSMIT, TRAORI).

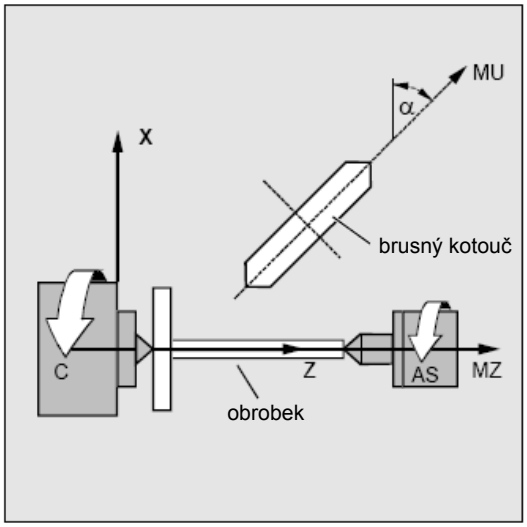
© Siemens AG 2004, Všechna práva vyhrazena

(Chování od SW stavu 6.4)
Je-li α (úhel) vynechán (např. TRAANG(), TRAANG(n,), je aktivována transformace s parametry předchozí volby. Při prvním aktivování se použije předběžné dosazení strojních parametrů.
Úhel α = 0 (např. TRAANG(0), TRAANG(0,n)), je platné zadání parametru a neodpovídá již vynechání parametru u starších verzí.
Přípustné hodnoty pro α jsou:
-90 stupňů < α < + 90 stupňů

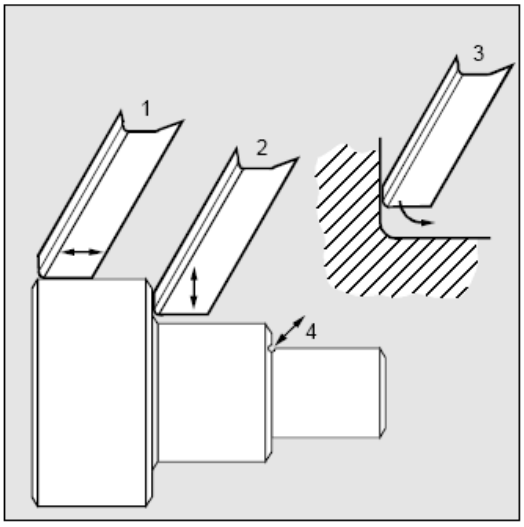
Funkce

Funkce šikmá osa je zamýšlena pro technologii broušení a umožňuje následující úkony:

- Opracování se šikmou osou přísuvu
- Pro programování může být použit kartézský systém souřadnic.
- Řídící systém transformuje programované pohyby kartézského systému souřadnic na pohyby reálných os stroje (standardní případ): šikmá osa přísuvu.



Jsou možná následující druhy opracování:
1. podélné broušení
2. příčné broušení
3. broušení určité kontury
4. šikmé zapichovací broušení



© Siemens AG 2004, Všechna práva vyhrazena

Prostřednictvím strojních parametrů jsou definována následující nastavení:

- Úhel mezi jednou osou stroje a šikmou osou
- Poloha počátku nástroje vztažená k počátku souřadného systému zvoleného při funkci „Šikmá osa“
- Rezerva rychlosti, která je v pohotovosti na paralelní ose pro vyrovnávací pohyb.
- Rezerva zrychlení osy, která je v pohotovosti na paralelní ose pro vyrovnávací pohyb.

Konfigurace os

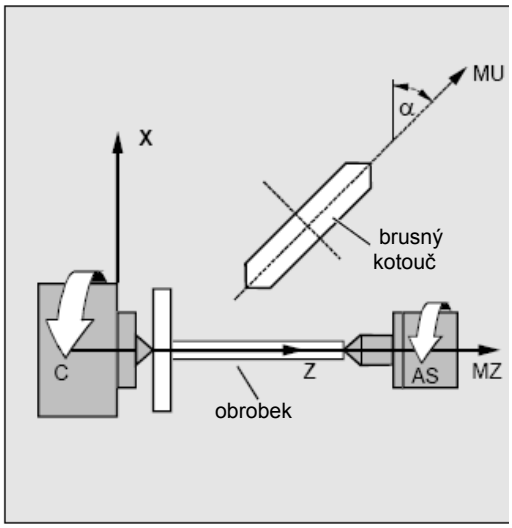
Aby bylo možné programovat v kartézském souřadném systému, musí být řídícímu systému sdělena souvislost mezi tímto souřadným systémem a skutečně existujícími osami stroje (MU, MZ):

- Pojmenování geometrických os
- Přiřazení geometrických os k osám kanálu
  - obecný případ (šikmá osa není aktivní)
  - šikmá osa aktivní
- Přiřazení os kanálu k číslům os stroje
- Označení vřeten
- Přiřazení ¨názvů os stroje

Postup odpovídá s výjimkou „Šikmá osa aktivní“ postupu při normální konfiguraci os.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování



`N10 G0 G90 Z0 MU=10 G54 F5000 ->` `-> G18 G64 T1 D1`	Volba nástroje, kompenzace upnutí, volba roviny
`N20 TRAANG(45)`	Aktivování transformaci šikmé osy
`N30 G0 Z10 X`	Najetí do počáteční polohy
`N40 WAITP(Z)`	Uvolnění osy pro kyvný pohyb
`N50 OSP[Z]=10 OSP2[Z]=5 OST1[Z]=-2 ->` `-> OST2[Z]=-2 FA[Z]=5000` `N60 OS[Z]=1` `N70 POS[X]=4.5 FA[X]=50` `N80 OS[Z]=0`	Kyvný pohyb, dokud není dosaženo rozměru (kyvný pohyb viz kapitola 9)
`N90 WAITP(Z)`	Uvolnění os kyvného pohybu jako polohovacích osy
`N100 TRAFOOF`	Deaktivování transformace
`N110 G0 Z10 MU=10`	Volný pohyb
`N120 M30`	

`-> na`programováno v jednom bloku

7.4.1 Programování šikmé osy: G05, G07 (od SW 5.3)

Programování

```
G07
G05
```

Vysvětlení příkazů

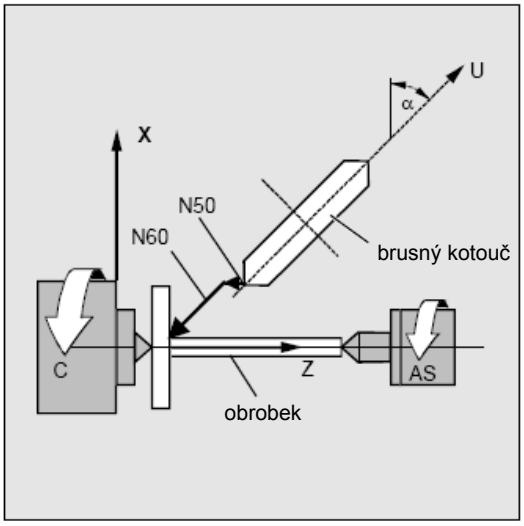
G07	Najetí do počáteční polohy
G05	Aktivování šikmého zapichování

Příkazy G07/G05 slouží k usnadnění programování šikmé osy.
Přitom mohou být programovány a zobrazovány polohy v kartézském systému souřadnic. Korekce nástroje a posunutí počátku jsou započteny kartézsky. Po naprogramování úhlu pro šikmou osu v NC-programu může být zaujata výchozí pozice (G07) a poté provedeno šikmé zapichování (G05).
V režimu JOG je možné brusným kotoučem pohybovat volitelně kartézsky nebo ve směru šikmé osy (zobrazení zůstane kartézské).
Pohybuje se jen reálná osa U, zobrazení osy Z je aktualizováno.

- S posuvy Repos je nutno v režimu JOG najíždět zpět kartézsky.
- Přejetí kartézského ohraničení pracovního pole je v režimu JOG při aktivním „pohybu PTP“ kontrolováno, odpovídající osa je předem zbrzděna. Není-li „pohyb PTP“ aktivní, může se po ose pohybovat přesně až k ohraničení pracovního pole.
  Literatura: /FB2/ F2: 3-5 osové transformace, kapitola 2 „Kartézský pohyb PTP“.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování



`N..`	Programování úhlu pro šikmou osu
`N20 G07 X70 Z40 F4000`	Najetí do počátečního bodu
`N30 G05 X70 F100`	Šikmý zápich
`N40 ...`	

7.5 Okrajové podmínky při volbě transformace

Zvolení transformací je možné prostřednictvím výrobního programu, popř. v MDA. Přitom je třeba mít na zřeteli následující:

- Pomocný pohybový blok nebude vložen (faseta/rádius).
- Splinový sled bloků musí být ukončen; pokud není, objeví se hlášení.
- Jemná korekce nástroje musí být deaktivována (G40); pokud není, objeví se hlášení.
- Aktivovaná korekce délky nástroje je řídícím systémem převzata do transformace.
- Před transformací platný aktuální frame je řídícím systémem deaktivován.
- Aktivní ohraničení pracovního pole je řídícím systémem pro osy, kterých se transformace dotýká, potlačeno (odpovídá WALIMOF).
- Monitorování chráněných oblastí je potlačeno.

© Siemens AG 2004, Všechna práva vyhrazena

- Režim řízení pohybu po dráze a zaoblování v rozích jsou přerušeny.
- Posuny ručním kolečkem (DRF) v osách, které se účastní transformace, se nesmí mezi předběžným a hlavním zpracováním programu změnit (do SW 3).
- Všechny ve strojním parametru zadané osy musí být blokově synchronizovány.
- Vyměněné osy jsou vráceny zpět, pokud ne, objeví se hlášení.
- U závislých os vypíše hlášení.

Změna nástroje

Změna nástroje je přípustná jen při deaktivované korekci rádiusu nástroje.

Změna korekce délky nástroje a aktivování/deaktivování korekce ráadiusu nástroje nesmí být naprogramovány ve stejném bloku.

© Siemens AG 2004, Všechna práva vyhrazena

Změna framu
Všechny příkazy, které se vztahují jen k základnímu souřadnému systému, jsou dovolené (FRAME, korekce rádiusu nástroje). Změna framu při G91 (inkrementální rozměry) není ale nijak zvlášť zpracována – jinak než při neaktivní transformaci. Inkrement, který je potřeba urazit, je vyhodnocen v souřadném systému obrobku nového framu – nezávisle na tom, který frame byl v platnosti v předchozím bloku.

Výjimky
Osy podílející se na transformaci nemohou být použity:
- Jako osa předvolby (alarm)
- Pro najetí na pevný bod (alarm)
- Pro najetí na referenční bod (alarm)

7.6 Deaktivování transformace: TRAFOOF



Programování

```
TRAFOOF
```



Vysvětlení příkazů

TRAFOOF	Deaktivuje všechny aktivní transformace/framy



Funkce

Prostřednictvím příkazu TRAFOOF jsou aktivní transformace a framy deaktivovány.



Potřebné framy musí být potom aktivovány prostřednictvím nových příkazů programu.

Přitom je třeba mít na zřeteli:
Pro deaktivování transformace platí tatáž omezení jako pro jejich aktivování (viz předchozí kapitola „Okrajové podmínky při volbě transformace“).

© Siemens AG 2004, Všechna práva vyhrazena

7.7 Zřetězené transformace

Od SW 5 mohou být připojeny za sebe (zřetězeny) vždy **dvě** transformace tak, že složky pohybu os z první transformace jsou vstupními daty pro zřetězenou druhou transformaci. Složky pohybu ze druhé transformace působí na osy stroje.

- Řetězec smí v SW 5 obsahovat **dvě** transformace.
- **Druhá** transformace musí být „**šikmá osa**“ (TRAANG).
- Jako první transformace jsou možné:
  - transformace orientace (TRAORI), včetně kardanové frézovací hlavy
  - TRANSMIT
  - TRACYL
  - TRAANG

Použití

- Broušení kontur, které byly programovány jako linie na rozvinuté válcové ploše (TRACYL) se šikmo umístěným brusným kotoučem, např. broušení nástrojů.
- Jemné opracování nekruhové kontury vytvořené pomocí příkazu TRANSMIT se šikmo umístěným brusným kotoučem.

Předpokladem pro použití spouštěcího příkazu pro zřetězenou transformaci je, že jednotlivé transformace, které mají být zřetězeny, a zřetězená transformace, která má být aktivována, jsou definovány prostřednictvím strojních parametrů.
Také při použití v rámci zřetězení je třeba mít na zřeteli okrajové podmínky a zvláštní případy uvedené v popisech pro jednotlivé transformace.

Další upozornění

Informace pro konfiguraci strojních parametrů transformací najdete v popisu funkcí M1 a F2.

© Siemens AG 2004, Všechna práva vyhrazena

Výrobce stroje (MH 7.1)

Dbejte upozornění výrobce stroje k případným pomocí strojních parametrů předem definovaným transformacím.

Transformace a zřetězené transformace jsou volitelné doplňky. O možnosti disponovat určitými transformacemi v řetězci u určitých typů řídícího systému dává informace vždy příslušný aktuální katalog.
Pro zřetězené transformace jsou k dispozici příkazy TRACON pro aktivování a TRAFOOF pro deaktivování.

Aktivování

Programování

`TRACON (trf, par)`	Je aktivována zřetězená transformace

Vysvětlení parametrů

`trf`	Číslo zřetězené transformace: 0 nebo 1 pro první/jedinou zřetězenou transformaci. Není-li na tomto místě programováno nic, má to tejný význam jako zadání hodnoty 0 nebo 1, tzn. je aktivována první/jediná transformace. 2 pro druhou zřetězenou transformaci. (hodnoty, které se nerovnají 0 – 2, vyvolají alarm).

© Siemens AG 2004, Všechna práva vyhrazena

par	Jeden nebo více parametrů pro transformace ve zřetězení oddělených čárkou, u kterých jsou parametry očekávány, např. úhel šikmé osy. Při nezadání parametrů budou v platnosti předběžná nastavení nebo poslední použité parametry. Zápisem čárek musí být zajištěno, že zadané parametry budou vyhodnoceny v pořadí, ve kterém jsou očekávány, mají-li pro předcházející parametry být účinná předběžná nastavení. Obzvláště musí být při zadání nejméně jednoho parametru před tímto parametrem čárka, i když zadání trf není nutné, tedy např. TRACON (, 3.7)



Funkce

Zřetězená transformace je aktivována. Předtím aktivní jiná transformace je prostřednictvím TRACON() implicitně vypnuta.



Nástroj je vždy přiřazen první transformaci řetězce. Následující transformace se pak chová tak, jako kdyby aktivní délka nástroje byla nula. Pro první transformaci řetězce jsou v platnosti jen základní délky jednoho nástroje (_BASE_TOOL_) nastavené prostřednictvím strojních parametrů.

Deaktivování



Programování

```
TRAFOOF
```



Funkce

Příkaz deaktivuje poslední aktivovanou (zřetězenou) transformaci.

© Siemens AG 2004, Všechna práva vyhrazena

7.8 Přepínatelné geometrické osy, GEOAX

Programování

```
GEOAX (n,osa kanálu,n,osa kanálu,…)

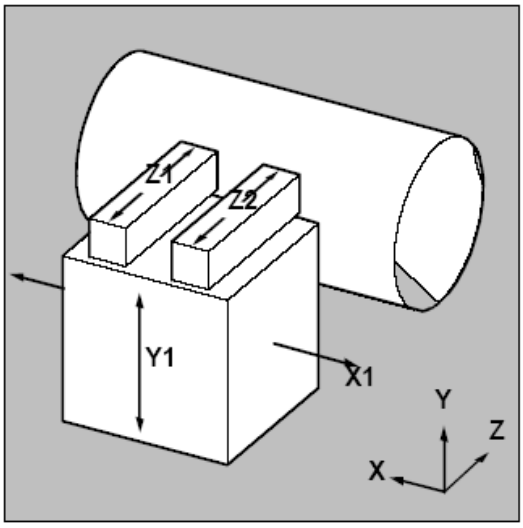
GEOAX ()
```

Vysvětlení parametrů

GEOAX (n,osa kanálu,n,osa kanálu,…)	Přepnutí geometrických os
GEOAX ()	Vyvolání základní konfiguraci geometrických os
n	Číslo geometrické osy (n=1,2 nebo 3), které má být přiřazena jiná osa kanálu. n=0: Bez náhrady odstranit zadanou osu kanálu ze svazku geometrických os.
osa kanálu	Jméno kanálové osy, která má být přijata do svazku geometrických os.

Funkce

Prostřednictvím funkce „Přepínatelné geometrické osy) je možné z výrobního programu změnit prostřednictvím strojních parametrů konfigurovaný svazek geometrických os. Přitom může osa kanálu, definovaná jako přídavná synchronní osa, nahradit libovolnou geometrickou osu.



Příklad:
Nástrojové saně se mohou pohybovat po osách kanálu X1, Y1, Z1, Z2 . Ve výrobním programu by měly být osy Z1 a Z2 střídavě použity jako osa Z. přepínání mezi osami probíhá ve výrobním programu pomocí příkazu GEOAX.

Po zapnutí je v platnosti spojení X1, Y1, Z1 (nastavitelné prostřednictvím SD).

```
N100 GEOAX (3,Z2)
N110 G1 . . . . .
N120 GEOAX (3,Z1)
```

Jako osa Z funguje osa kanálu Z2

Jako osa Z funguje osa kanálu Z1

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Číslo geometrické osy

V příkazu GEOAX (n, osa kanálu…) označuje číslo n geometrickou osu, které má být následně zadaná osa kanálu přiřazena.
Pro změnu kanálové osy jsou přípustná čísla geometrických os 1 až 3 (X-,Y-, Z-osa).
Prostřednictvím n=0 je přiřazená osa bez nového obsazení geometrické osy odstraněna ze svazku geometrických os.

Pomocí přepnutí ve svazku geometrických os je nahrazená osa po přepnutí programovatelná prostřednictvím svého názvu kanálové osy jako přídavná osa.

Přepnutím geometrických os jsou vymazány všechny framy, chráněné oblasti a omezení pracovního pole.

Polární souřadnice:

Změna geometrických os pomocí příkazu GEOAX nastaví analogicky změně rovin (G17 – G19) modální polární souřadnice na hodnotu 0.

Posunutí ručním kolečkem, posunutí počátku:

Případné posunutí ručním kolečkem (DRF) nebo externí posunutí počátku zůstávají po přepnutí v platnosti.

Změna polohy os

Prostřednictvím nového přiřazení čísel os k již přiřazeným osám kanálu je možná také změna polohy v rámci svazku geometrických os.

N… GEOAX (1,XX,2,YY,3,ZZ)	Osa kanálu XX je první, YY druhá a ZZ třetí geometrická osa,
N… GEOAX (1,U,2,V,3,W)	Osa kanálu U je první, V druhá a W třetí geometrická osa.

© Siemens AG 2004, Všechna práva vyhrazena

Předpoklady a omezení

1. Přepínání geometrických os není možné při:
   - aktivní transformaci,
   - aktivní splinové interpolaci,
   - aktivní korekci rádiusu nástroje,
   - aktivní jemné korekci nástroje

   (Příručka programování – Základy, kap. 8)
   (Příručka programování – Základy, kap. 8)
2. Mají-li geometrická osa a osa kanálu stejné názvy, není možná změna příslušné geometrické osy.
3. Žádná z os, které se na přepínání podílejí, se nesmí účastnit akce, která by mohla trvat za hranice bloku, jak je to možné např. u polohovacích os typu A nebo u vlečných os.
4. Prostřednictvím příkazu GEOAX mohou být nahrazeny jenom geometrické osy, které se vyskytovaly již při zapnutí (tedy nemohou být dodatečně definovány žádné nové).
5. Změna os prostřednictvím GEOAX během zpracování tabulky kontur (CONTPRON, CONTDCON) vede k alarmu.

Deaktivování přepínání

Příkaz GEOAX() vyvolá základní konfiguraci svazku geometrických os.
Po POWER ON a při přepnutí do provozního režimu najíždění na referenční bod je automaticky přepnuto zpět do základní konfigurace.

Další upozornění

Proces přepnutí a korekce délky nástroje

Aktivní korekce délky nástroje je v platnosti také po procesu přepnutí. Geometrické osy, které byly nově přijaty, resp. u nich došlo k výměně polohy, jsou však považovány za osy, s nimiž se dosud nepohybovalo.
Při prvním příkazu pohybu pro tyto geometrické osy sestává výsledná dráha pohybu ze součtu korekce délky nástroje a programované dráhy pohybu.
Geometrické osy, které si při přepnutí zachovaly svoji polohu ve sazku os, si zachovají i svůj stav ve vztahu ke korekci délky nástroje.

© Siemens AG 2004, Všechna práva vyhrazena

Konfigurace geometrických os a změna transformace

Konfigurace geometrických os platná v aktivní transformaci (stanovená prostřednictvím strojních parametrů) není změnitelná pomocí funkce „přepínatelné geometrické osy“.

Nastane-li nutnost v souvislosti s transformacemi změnit konfiguraci geometrických os, je toto možné jen prostřednictvím další transformace.

Konfigurace geometrických os změněná pomocí příkazu GEOAX je aktivováním transformace vymazána.

Jestliže si odporují nastavení pro transformaci a pro přepnutí geometrických os, má nastavení pro transformaci přednost.

Příklad:

Transformace je aktivní. Podle strojních parametrů má transformace při resetu zůstat zachována, současně má být při resetu však obnovena základní konfigurace geometrických os. V tomto případě zůstane konfigurace geometrických os zachována tak, jak byla stanovena transformací.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Stroj má 6 kanálových os s názvy XX, YY, ZZ, U, V, W. Základní nastavení konfigurace geometrických os prostřednictvím strojních parametrů je:
Kanálová osa XX = 1. geometrická osa (osa X)
Kanálová osa YY = 2. geometrická osa (osa Y)
Kanálová osa ZZ = 3. geometrická osa (osa Z)

`N10 GEOAX()`	Je v platnosti základní konfigurace geometrických os
`N20 G0 X0 Y0 Z0 U0 V0 W0`	Všechny osy v rychlém posuvu v poloze 0
`N30 GEOAX(1,U,2,V,3,W)`	Kanálová osa U se stane první (X), V druhou (Y) , W třetí (Z) geometrickou osou.
`N40 GEOAX(1,XX,3,ZZ)`	Kanálová osa XX se stane první (X), ZZ třetí (Z) geometrickou osou. Kanálová osa V zůstane druhou geometrickou osou (Y)
`N50 G17 G2 X20 I10 F1000`	Kružnice v rovině X, Y. Pohybují se kanálové osy XX a V.
`N60 GEOAX(2,W)`	Kanálová osa W se stane druhou geometrickou osou (Y)
`N80 G17 G2 X20 I10 F1000`	Kružnice v rovině X, Y. Pohybují se kanáovéu osy XX a W.
`N90 GEOAX()N100`	Obnovení základního stavu
`N100 GEOAX(1,U,2,V,3,W)`	Kanálová osa U se stane první (X), V druhou (Y), W třetí (Z) geometrickou osou.
`N110 G1 X10 Y10 Z10 XX=25`	Kanálové osy U, V, W se pohybují do polohy 10, XX jako přídavná osa se pohybuje do polohy 25.
`N120 GEOAX(0,V)`	Osa V je vyjmuta ze svazku geometrických os. U a W jsou nadále první (X) a třetí (Z) geometrickou osou. Druhá geometrická osa (Y) zůstane neobsazena.
`N130 GEOAX(1,U,2,V,3,W)`	Kanálová osa U zůstane první (X), V se stane druhou (Y), W zůstane třetí (Z) geometrickou osou.
`N140 GEOAX(3,V)`	V se stane třetí geometrickou osou (Z), přičemž W je přepsána a tím vyjmuta ze svazku geometrických os. Druhá geometrická osa (Y) zůstává jako předtím neobsazená.

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Korekce nástroje

8.1 Paměť korekcí 8-320
8.2 Příkazy jazyka pro správu nástrojů 8-322
8.3 On-line korekce nástroje, PUTTOCF, PUTFTOC, FTOCON, FTOCOF 8-325
8.4 Udržování konstantní korekce rádiusu nástroje, CUTCONON (od SW 4) ... 8-330
8.5 Aktivování 3D korekčních parametrů nástroje, CUT3DC, CUT3DF, CUT3DFS/CUT3DFF 8-333
8.5.1 3D korekce rádiusu nástroje, frézování na obvodové ploše, frézování na čelní ploše 8-335
8.5.2 Typy nástrojů/výměna nástroje se změněnými rozměry G40/41/42 8-336
8.5.3 Korekce po dráze, zakřivení dráhy a hloubka zajíždění ISD 8-337
8.5.4 Posuv na vnitřních/vnějších rozích a průsečících, G450/G451 (od SW 5)... 8-338
8.5.5 Obvodové 3D frézování s omezovacími plochami, CUT3DCC, CUT3DCCD 8-340
8.6 Orientace nástroje, ORIC, ORID, OSOF, OSC, OSS, OSSE..... 8-345
8.7 Volné předávání D-čísel, číslo břitu CE (od SW 5)..... 8-350
8.7.1 Kontrola D-čísla (CHKDNO) 8-351
8.7.2 Změna názvu D-čísla (GETDNO, SETDNO)..... 8-352
8.7.3 Zjištění T-čísla k předem zadanému D-číslu (GETACTTD) 8-353
8.7.4 Prohlášení D-čísla za neplatné..... 8-355
8.8 Kinematika držáku nástroje 8-355

© Siemens AG 2004, Všechna práva vyhrazena

8.1 Paměť korekcí

Struktura paměti korekcí

Každé datové pole může být vyvoláno pomocí T- a D-čísla (s výjimkou prostých D-čísel) a obsahuje kromě geometrických údajů o nástroji i další záznamy, např. typ nástroje.

Od SW 4:
Jestliže je správa nástrojů uskutečňována mimo NCK, použije se „struktura prostých D-čísel“. V tom případě jsou D-čísla s příslušnými korekčními bloky nástrojů generována bez přiřazení k nástrojům.
Ve výrobním programu můžete i přesto naprogramovat T. Toto T však nemá žádnou souvislost k naprogramovanému D-číslu.

Pro geometrické veličiny (např. délka 1 nebo rádius) existují další složky záznamu, které se přičítají k výslednému parametru (např. celková délka 1, celkový rádius), který se pak uplatňuje.

Nepotřebným korekčním parametrům je zapotřebí dosadit nulu.



Jednotlivé hodnoty pamětí korekcí P1 až P25 mohou být prostřednictvím systémových proměnných čteny nebo zapisovány.

© Siemens AG 2004, Všechna práva vyhrazena

Parametry nástroje	Význam systémových proměnných	Poznámka
Číslo DP		
$TC_DP_1	Typ nástroje	Přehled viz seznam
$TC_DP_2	Poloha břitu	pro soustr. nástroje
Geometrie	**Korekce délky**	
$TC_DP_3	Délka 1	Započítání podle typu
$TC_DP_4	Délka 2	a roviny
$TC_DP_5	Délka 3	
Geometrie	**Rádius**	
$TC_DP_6	Rádius	
$TC_DP_7	Šířka drážky pro drážkovací pilu, rádius zaoblení pro frézovací nástroje	
$TC_DP_8	Přesah k	Jen pro drážkovací pilu
$TC_DP_11	Úhel pro kuželové frézovací nástroje	
Opotřebení	**Korekce délky a korekce rádiusu**	
$TC_DP_12	Délka 1	
$TC_DP_13	Délka 2	
$TC_DP_14	Délka 3	
$TC_DP_15	Rádius	
$TC_DP_16	Šířka drážky pro drážkovací pilu, rádius zaoblení pro frézovací nástroje	
$TC_DP_17	Přesah k	Jen pro drážkovací pilu
$TC_DP_20	Úhel pro kuželové frézovací nástroje	
Základní rozměr/adaptér	**Korekce délky**	
$TC_DP_21	Délka 1	
$TC_DP_22	Délka 2	
$TC_DP_23	Délka 3	
Technologie		
$TC_DP_24	Úhel volného řezání	Pro soustružnické nástroje



Další upozornění

Všechny zbývající parametry jsou vyhrazeny.



Výrobce stroje

Uživatelské parametry břitů lze konfigurovat pomocí MD.

© Siemens AG 2004, Všechna práva vyhrazena

8.2 Příkazy jazyka pro správu nástrojů

Vysvětlení příkazů

T="WZ"	Volba nástroje jeho názvem
NEWT ("WZ",DUPLO_NR)	Vytvoření nového nástroje, duplo-číslo volitelné
DELT ("WZ",DUPLO_NR)	Vymazání nástroje, duplo-číslo volitelné
GETT ("WZ",DUPLO_NR)	Stanovení T-čísla
SETPIECE (x,y)	Nastavení počtu kusů
GETSELT (x)	Načtení předem definovaného čísla nástroje (T-čísla)
"WZ"	Identifikátor nástroje
DUPLO_NR	Počet kusů
x	Číslo vřetena, volitelný údaj)

Při použití správy nástrojů můžete zakládat a vyvolávat nástroje pomocí jejich názvů, například T="BOHRER" nebo T="123".

Funkce NEWT

Pomocí funkce NEWT je možné v NC programu založit nový nástroj s jeho názvem. Funkce automaticky vrací vytvořené T-číslo, kterým je možné se na nástroj potom obracet.

```
Vracený param.= NEWT ("WZ",DUPLO_NR)
```

Pokud duplo-číslo není zadáno, bude vytvořeno ve správě nástrojů.

Příklad:

```
DEF INT DUPLO_NR
DEF INT T_NR
DUPLO_NR = 7
T_NR=NEWT("BOHRER",DUPLO_NR)
```

Založení nového nástroje „BOHRER" s duplo-číslem 7. Vytvořenému T-číslu bude dosazena hodnota T_NR.

Funkce DELT

Pomocí funkce DELT je možné nástroj vymazat bez odkazu na jeho T-číslo.

```
DELT ("WZ",DUPLO_NR)
```


© Siemens AG 2004, Všechna práva vyhrazena

Funkce GETT

Funkce GETT vrací T-číslo potřebné pro nastavení parametrů nástroje, pro který je znám pouze jeho název.

```
Vracený param.= GETT ("WZ",DUPLO_NR)
```

Pokud existuje více nástrojů s uvedeným identifikátorem, bude se předávat T-číslo prvního z nich.

Pokud je vracená hodnota –1: Uvedenému označení nástroje, příp. duplo-číslu není možné přiřadit žádný nástroj.

Příklady:

`T="BOHRER"`	
`R10=GETT("BOHRER", DUPLO_NR)`	Zjištění T-čísla pro nástroj BOHRER s duplo-číslem = DUPLO_NR.

Nástroj „BOHRER“ musí být předtím deklarován pomocí příkazu NEWT nebo `$TC_TP[ ]`.

`$TC_DP1[GETT("BOHRER",DUPLO_NR,1]=100`	Zápis parametru nástroje (systémová proměnná) s názvem nástroje

Funkce SETPIECE

Tato funkce slouží pro aktualizaci parametru sledování počtu kusů.
Funkce počítá všechny břity nástroje, které byly vyměněny od posledního aktivování příkazu SETPIECE na uvedeném číslu vřetena.

SETPIECE(x,y)

`x`	Počet vyrobených obrobků
`y`	Číslo vřetena, 0 znamená hlavní vřeteno (standardní nastavení).

© Siemens AG 2004, Všechna práva vyhrazena

Funkce GETSELT

Funkce vrací T-číslo nástroje, který byl předem aktivován pro vřeteno.
Díky této funkci se lze ještě před příkazem M6 dostat ke korekčním parametrům nástroje a tedy je možné o něco dříve dosáhnout synchronizace se zpracováním hlavního programu.

Příklad pro výměnu nástroje se správou nástrojů

T1 Předvolba nástroje; tzn. zásobník nástrojů může být souběžně se zpracováváním nastaven do polohy pro daný nástroj.

M6 Výměna předem zvoleného nástroje (v závislosti na nastavení ve strojních parametrech může být naprogramováno i bez M6).

Příklad:

`T1 M6`	Výměna nástroje T1
`D1`	Aktivování korekce délky
`G1 X10 ...`	Práce s nástrojem T1
`T="BOHRER"`	Předvolba nástroje BOHRER
`D2 Y20 ...`	Výměna břitu nástroje T1
`X10 ...`	Práce s nástrojem T1
`M6`	Výměna nástroje BOHRER
`SETPIECE(4)`	Počet vyrobených obrobků
`D1 G1 X10 ...`	Práce s vrtákem



Úplný seznam všech proměnných pro správu nástrojů naleznete v seznamu systémových proměnných v příloze.

© Siemens AG 2004, Všechna práva vyhrazena

8.3 On-line korekce nástroje, PUTTOCF, PUFTOC, FTOCON, FTOCOF

Programování

```
FCTDEF(č.polynomu, LLimit, ULimit,a0,a1,a2,a3)
PUTFTOCF (č.polynomu, Vztažná hodnota, Délka1_2_3, Kanál, Vřeteno)
PUTFTOC (Hodnota, Délka1_2_3, Kanál, Vřeteno)
FTOCON
FTOCOF
```

Vysvětlení příkazů

PUTFTOCF	Kontinuální zápis on-line korekce nástroje
FCTDEF	Nastavení parametrů funkce PUTFTOCF
PUTFTOC	Diskrétní zápis on-line korekce nástroje
FTOCON	Aktivování on-line korekce nástroje
FTOCOF	Deaktivování on-line korekce nástroje

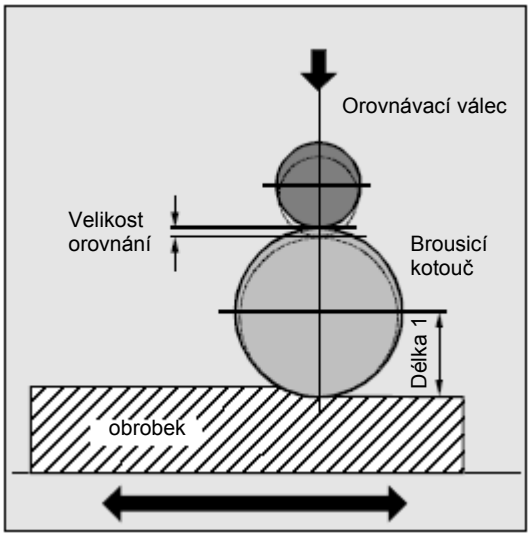
Vysvětlení parametrů

č.polynomu	Hodnota 1 až 3: Současně jsou možné maximálně 3 polynomy; polynom maximálně 3. stupně.
Vztažná hodnota	Vztažná hodnota, od které se korekce odvozuje
Délka1_2_3	Parametr opotřebení, k němuž se hodnota korekce nástroje přičítá
Kanál	Číslo kanálu, v němž se korekce nástroje uplatňuje. Udává se jen tehdy, pokud se nejedná o vlastní kanál.
Vřeteno	Číslo vřetena, pro které on-line korekce nástroje platí; údaj je zapotřebí pouze pro neaktivní brusné kotouče.
LLimit	Horní mezní hodnota
ULimit	Dolní mezní hodnota
a_0, a_1, a_2, a_3	Koeficienty polynomické funkce
Hodnota	Hodnota, která se přičítá k parametru opotřebení

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Pomocí této funkce jsou pomocí on-line korekce délky okamžitě započítávány korekční parametry nástroje, které vyplývají z obrábění (např. CD-orovnávání: brousící kotouč je orovnáván souběžně s obráběním). Korekci délky nástroje je možné měnit z kanálu, v němž obrábění probíhá, nebo z paralelního kanálu (kanálu orovnávače).



On-line korekci nástroje je možné používat pouze u brousicích nástrojů.

Všeobecně k on-line korekcím nástroje

V závislosti na okamžiku, kdy se operace orovnávání provádí, se pro zápis on-line korekce nástroje používají různé funkce:

- Blokový kontinuální zápis: PUTFTOCF
- Modální kontinuální zápis: ID=1 DO FTOC (viz kapitola „Synchronní akce“)
- Diskrétní zápis:PUTFTOC

Při kontinuálním zápisu (s každým taktem IPO) se po aktivování vyhodnocovací funkce každá změna aditivně započítává do paměti opotřebení, aby se zabránilo skokovým změnám požadované hodnoty.
V každém případě platí:
On-line korekce nástroje se mohou uplatňovat v každém kanálu, pro každé vřeteno a délku 1, 2 **nebo** 3 parametrů opotřebení.

Přiřazení délek geometrickým osám se uskutečňuje podle aktuální roviny.

Přiřazení vřetena nástroji probíhá prostřednictvím parametrů nástroje příkazy GWPSON, příp. TMON, pokud se však nejedná o aktivní brusný kotouč (viz Příručka programování – Základy). Korekce se vždy týká parametru opotřebení pro aktuální stranu kotouče, příp. levou stranu kotouče u neaktivních nástrojů.

© Siemens AG 2004, Všechna práva vyhrazena

V případě identických korekcí pro více stran brousicího kotouče je pomocí předpisu zřetězení (popis uveden v návodu k obsluze) zapotřebí zajistit, aby hodnota byla automaticky přebírána pro druhou sadu brousicího kotouče.

Pokud jsou pro nějaký obráběcí kanál definovány on-line korekce, hodnoty opotřebení aktuálního nástroje nesmí být v tomto měněny z výrobního programu nebo zásahem obsluhy.

On-line korekce nástroje jsou zohledňovány také pro konstantní obvodovou rychlost brusného kotouče (SUG), jakož i pro monitorování nástroje TMON a mimostředné broušení CLGON.

Postup

PUTFTOCF = Kontinuální zápis

Operace orovnávání se provádí souběžně s obráběním:
Orovnávání přes celou šířku brousicího kotouče pomocí orovnávacího válce nebo orovnávacího diamantu z jedné strany brousicího kotouče na druhou.

Obrábění a orovnávání se může provádět v různých kanálech. Není-li naprogramován žádný kanál, bude korekce uplatněna v aktuálním kanálu.

```
PUTFTOCF (č.polynomu, vztažná hodnota, Délka1_2_3, Kanál, Vřeteno)
```

Korekce nástroje se bude v kanálu pro opracování průběžně měnit podle polynomické funkce 1., 2. nebo 3. stupně, která musí být definována předem pomocí příkazu FCTDEF.
Korekce bude odvozována od proměnné „vztažná hodnota“, např. od skutečné hodnoty, která se má měnit.
Jestliže není naprogramováno žádné číslo vřetena, korekce se bude týkat právě používaného nástroje.
Nastavení parametrů funkce FCTDET
Dosazení parametrů se provádí v samostatném bloku:

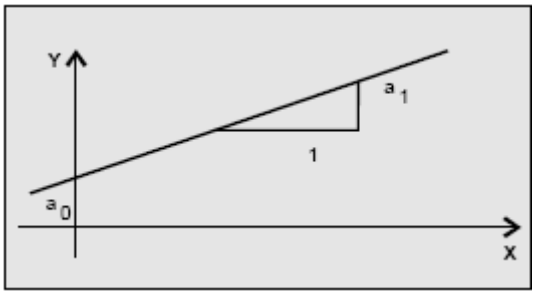
```
FCTDEF(č.polynomu, LLimit, ULimit,a0,a1,a2,a3)
```

Polynom může být 1., 2. nebo 3. stupně. Proměnná Limit označuje mezní hodnotu (LLimit = dolní, ULimit = horní mezní hodnota).

© Siemens AG 2004, Všechna práva vyhrazena

Příklad:
Přímka (y = a0 + a1x) se stoupáním 1

```
FCTDEF(1, -1000, 1000, -$AA_IW[X], 1)
```



Diskrétní zápis on-line korekce:PUTFTOC
Pomocí tohoto příkazu lze jedenkrát zapsat hodnotu korekce. Korekce je v cílovém kanálu okamžitě v platnosti.
Použití příkazu PUTFTOC:
Brousicí kotouč je orovnáván z paralelního kanálu, ne však souběžně s obráběním.

```
PUTFTOC (Hodnota, Délka1_2_3, Kanál, Vřeteno)
```

On-line korekce nástroje pro uvedenou délku1, 2 **nebo** 3 bude změněna o uvedenou hodnotu, tzn. hodnota bude přičtena k parametru opotřebení.

Započítání on-line korekce nástroje: FTOCON, FTOCOF
Cílový kanál může zachytit on-line korekce nástroje jen tehdy, pokud je FTOCON aktivní.

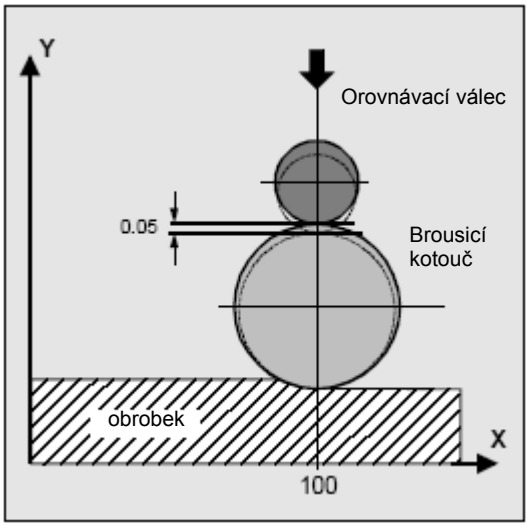
- FTOCON musí být zapsáno v kanálu, v němž se má korekce uplatňovat.
  Pomocí FTOCOF se korekce přestane uplatňovat, kompletní hodnota zapsaná příkazem PUTFTOC do korekčních parametrů příslušného břitu však zůstane korigována.
- Po resetu je vždy v platnosti FTOCOF.
- Příkaz PUTFTOCF má vždy blokovou platnost, tzn. v následujícím pohybovém bloku.
- Příkazem FTOC může být on-line korekce nástroje aktivována také modálně. Další informace viz kapitola "Pohybové synchronní akce“.

Příklad programování

Úkol
U plošné brusky s následujícími parametry se má po zahájení brusného pohybu u X100 brousicí kotouč orovnat o 0,05 mm. Velikost orovnávání má být kontinuálně aplikována pomocí zápisu on-line korekčních parametrů brousicího nástroje.

Y: Přísuvná osa pro brousicí nástroj
V: Přísuvná osa pro orovnávací válec

Obrábění: kanál 1 s osami X, Z, Y
Orovnávání: kanál 2 s osou V



© Siemens AG 2004, Všechna práva vyhrazena

Program pro obrábění v kanálu 1:

%_N_BEARB_MPF	
...	
N110 G1 G18 F10 G90	základní nastavení polohy
N120 T1 D1	volba aktuálního nástroje
N130 S100 M3 X100	spuštění vřetena, najíždění na poč. pozici
N140 INIT (2, "ABRICHT", "S")	aktivování orovnávacího programu v kanálu 2
N150 START (2)	spuštění orovnávacího programu v kanálu 2
N160 X200	najíždění na cílovou pozici
N170 FTOCON	aktivování on-line korekce nástroje
N... G1 X100	další obrábění

Program pro orovnávání v kanálu 2:

%_N_ABRICHT_MPF	
...	
N40 FCTDEF (1,-1000,1000,-$AA_IW[V],1)	definice funkce: přímka
N50 PUTFTOCF (1, $AA_IW[V], 3, 1)	kontinuální zápis on-line korekce nástroje: délka 3 aktuálního brusného kotouče je odvozena od pohybu osy V a korigována v kanálu 1.
N60 V-0.05 G1 F0.01 G91	přísuvný pohyb pro orovnání, příkaz PUTFTOCF je v platnosti jen v tomto bloku
...	
N M30	

Orovnávací program s modální platností:

%_N_ABRICHT_MPF	
FCTDEF(1,-1000,1000,-$AA_IW[V],1)	definice funkce
ID=1 DO FTOC(1,$AA_IW[V],3,1)	aktivování on-line korekce nástroje: Skutečná hodnota osy V je vstupní hodnotou pro polynom 1; výsledek se přičítá v kanálu 1 jako korekční hodnota pro délku 3 aktivního brusného kotouče.
WAITM(1,1,2)	synchronizace s kanálem pro obrábění
G1 V-0.05 F0.01, G91	přísuvný pohyb pro orovnávání
G1 V-0.05 F0.02	
...	
CANCEL(1)	deaktivování on-line korekce
...	

© Siemens AG 2004, Všechna práva vyhrazena

8.4 Udržování konstantní korekce rádiusu nástroje, CUTCONON (od SW 4)



Programování

```
CUTCONON
CUTCONOF
```



Vysvětlení příkazů

CUTCONON	Aktivování funkce pro udržování konstantní korekce rádiusu nástroje
CUTCONOF	Deaktivování funkce pro udržování konstantní korekce rádiusu nástroje (standardní nastavení)



Funkce

Funkce „udržování konstantní korekce rádiusu nástroje“ slouží pro potlačení korekce rádiusu nástroje pro určitý počet bloků, přičemž diference mezi naprogramovanou a skutečně ujetou dráhou středu nástroje, která vznikla v předešlých blocích kvůli korekci rádiusu nástroje, zůstává zachována jako posunutí.
To může být užitečné, např. když při frézování po přímkách je v bodech obratu zapotřebí větší počet pohybových bloků, u nichž však nejsou žádoucí kontury vznikající v důsledku korekce rádiusu nástroje (strategie objíždění).
Příkaz lze používat nezávisle na druhu korekce rádiusu nástroje (2½D, 3D-čelní frézování, 3D-obvodové frézování).



Postup

V obvyklém případě je korekce rádiusu nástroje před aktivováním svého potlačení již v platnosti a v platnosti zůstává, když je potlačení korekce opět deaktivováno.
V posledním bloku posuvu před příkazem CUTCONON se najíždí na bod offsetu v koncovém bodu bloku. Posuv ve všech následujících blocích, v nichž je potlačení korekce aktivní, se provádí bez korekce.
Jsou však přitom posunuty o vektor z koncového bloku s korekcí k příslušnému bodu offsetu.

© Siemens AG 2004, Všechna práva vyhrazena

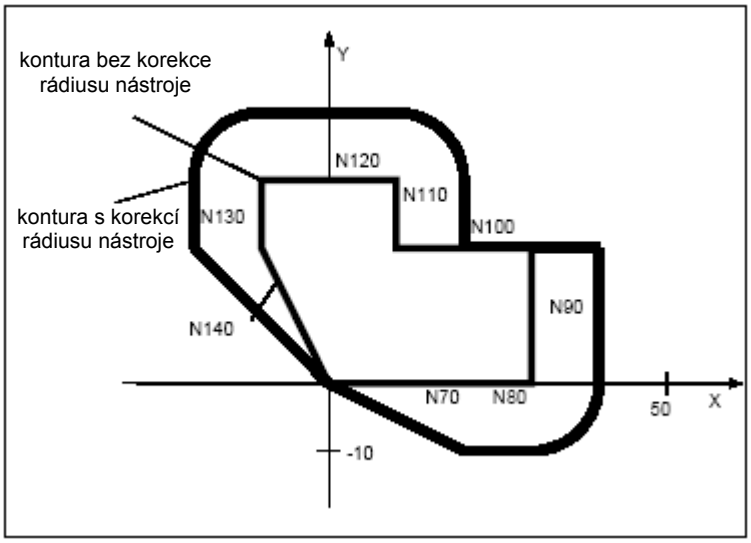
Typ interpolace těchto bloků (lineární, kruhová, polynomická) je libovolný.
V bloku s deaktivováním potlačení korekce, tzn. v bloku, který obsahuje příkaz CUTCONOF, je korekce normálně uplatňována; začíná v počátečním bodě posunutém o offset. Mezi koncovým bodem předešlého bloku, tzn. posledním naprogramovaným blokem s posuvem, v němž je aktivní příkaz CUTCONON, a tímto bodem se vkládá lineární blok.
Kruhové bloky, u nichž se rovina kruhu nachází kolmo na rovinu korekce (vertikální kruhy), budou zpracovány tak, jako by v nich byl naprogramován příkaz CUTCONON.
Toto implicitní aktivování potlačení korekce se automaticky zpětně uskutečňuje v prvním bloku posuvu, který obsahuje pohyb posuvu v rovině korekce a žádný kruh tohoto druhu. Vertikální kruhy v tomto smyslu se mohou vyskytnout pouze při obvodovém frézování.



Příklad

N10	; definice nástroje D1
N20 $TC_DP1[1,1]= 110	; typ
N30 $TC_DP6[1,1]= 10.	; rádius
N40	
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60	
N70 X20 G42 NORM	
N80 X30	
N90 Y20	
N100 X10 CUTCONON	; aktivování potlačení korekce
N110 Y30 CONT	; při deaktivování potlačení korekce v případě potřeby vložit objížděcí kružnici
N120 X-10 CUTCONOF	
N130 Y20 NORM	; žádná objížděcí kružnice se nevkládá při deaktivování korekce rádiusu nástroje
N140 X0 Y0 G40	
N150 M30	

© Siemens AG 2004, Všechna práva vyhrazena



Další upozornění

1. Pokud není žádná korekce rádiusu nástroje aktivní (G40), nemá funkce CUTCONON žádný efekt. Neaktivuje se žádný alarm. G-kód však zůstává aktivní. To je důležité, pokud v některém z následujících bloků má být pomocí příkazů G41, příp. G42 korekce rádiusu nástroje znovu aktivována.
2. Změna G-kódu v 7. skupině G-kódů (korekce rádiusu nástroje; G40 / G41 / G42) při aktivním příkazu CUTCONON je přípustná. Změna bude v platnosti ihned po příkazu G40. Uskuteční se posunutí, která se uplatňovala v předcházejících blocích.
3. Pokud je naprogramován příkaz CUTCONON nebo CUTCONOF v bloku bez posuvu v aktivní rovině korekcí, bude jeho uplatnění odloženo až do následujícího bloku s takovým posuvem.

Další informace:
/FB/, W1 Korekce nástroje.

© Siemens AG 2004, Všechna práva vyhrazena

8.5 Aktivování 3D korekčních parametrů nástroje, CUT3DC, CUT3DF, CUT3DFS/CUT3DFF

Vysvětlení

`CUT3DC`	Aktivování 3D korekce rádiusu pro obvodové frézování
`CUT3DFS`	3D korekce nástroje pro čelní frézování s konstantní orientací. Orientace nástroje je definována příkazy G17 – G19 a není ovlivňována framy.
`CUT3DFF`	3D korekce nástroje pro čelní frézování s konstantní orientací. Orientace nástroje je definována příkazy G17 – G19 a v případě potřeby může být směrově otočena pomocí framu.
`CUT3DF`	3D korekce nástroje pro čelní frézování se změnou orientace (jen při aktivní 5-osové transformaci).
`CUT3DCC`	3D korekce nástroje pro obvodové frézování s omezujícími plochami.
`CUT3DCCD`	3D korekce nástroje pro obvodové frézování s omezujícími plochami a s diferenčním nástrojem.
`G40 X Y Z`	Pro deaktivování: Lineární blok G0/G1 s geometrickými osami.
`ISD=hodnota`	Hloubka zajíždění

Příkazy mají modální platnost a nachází se ve stejné skupině jako příkazy CUT2D a CUT2DF.

Deaktivování se uskuteční teprve s následujícím pohybem v aktuální rovině. To platí vždy pro příkaz G40 a není závislé na příkazu CUT.

Funkce

Při korekci rádiusu pro válcové nástroje se zohledňuje proměnná orientace nástroje.

Pro aktivování 3D korekce rádiusu nástroje platí stejné programové příkazy jako u 2D korekce rádiusu nástroje. Pomocí příkazů G41/G42 se zadává korekce vlevo/vpravo ve směru pohybu. Chování při najíždění je vždy NORM.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad

`N10 A0 B0 X0 Y0 Z0 F5000`	; volání nástroje, vyvolání hodnot korekcí nástroje
`N20 T1 D1`	; aktivování transformace
`N40 CUT3DC`	; aktivování 3D korekce rádiusu nástroje
`N50 G42 X10 Y10`	; aktivování korekce rádiusu nástroje
`N60 X60`	
`N70 ...`	

Další upozornění

Pomocné bloky při aktivní 3D korekci rádiusu nástroje jsou přípustné. Platí definice 2½D korekce rádiusu nástroje.

3D korekce rádiusu nástroje je v platnosti, jen když je aktivována 5-osová transformace.

Na vnějších rozích se vždy vkládá blok kruhového oblouku. G450/G451 nemají žádný efekt.

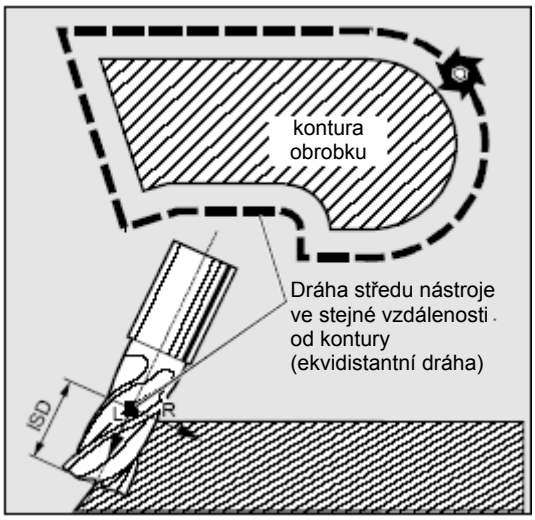
Příkaz DISC se nevyhodnocuje.

Rozdíl mezi 2 ½ D a 3D korekcí rádiusu nástroje

Při 3D korekci rádiusu nástroje je orientace nástroje proměnná.

Při 2 ½ D korekci rádiusu nástroje se počítá jen s nástrojem s konstantní orientací.

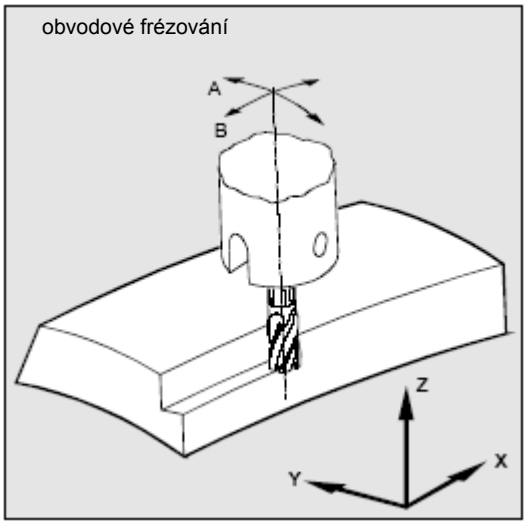
3D korekce rádiusu nástroje je označována také jako 5D korekce, protože v tomto případě je pro polohu nástroje v prostoru k dispozici 5 stupňů volnosti.



© Siemens AG 2004, Všechna práva vyhrazena

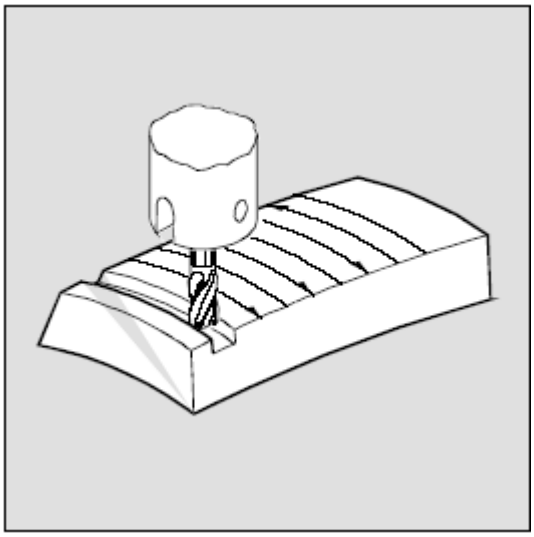
8.5.1 3D korekce rádiusu nástroje, frézování na obvodové ploše, frézování na čelní ploše

Frézování na obvodové ploše
Zde používaná varianta obvodového frézování je realizována zadáním dráhy (vodicí čáry) a příslušné orientace. Při tomto druhu opracování nemá tvar nástroje žádný vliv na dráhu. Jedinou rozhodující veličinou je rádius v bodě záběru nástroje.



Funkce 3D korekce nástroje je omezena na válcové nástroje.

Frézování na čelní ploše
Pro tento druh 3D frézování potřebujete popis 3D drah na povrchu obrobku, úsek po úseku.
Výpočty se obvykle uskutečňují v systému CAM a berou se při nich v úvahu tvar nástroje a rozměry nástroje.
Postprocesor zapisuje d výrobního programu vedle NC bloků i údaje o orientaci nástroje (při aktivní 5-osové transformaci) a G-kód pro požadovanou 3D korekci nástroje.



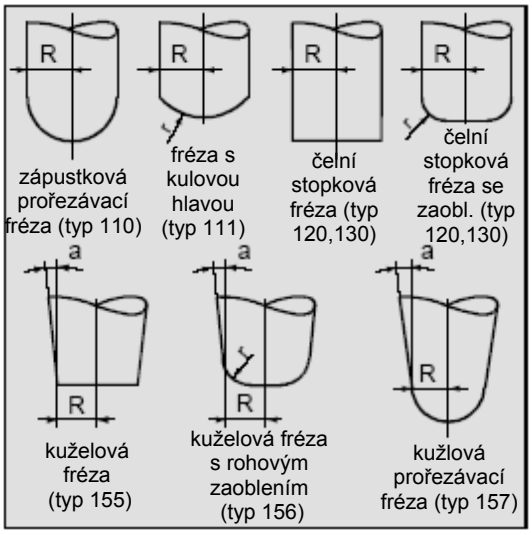
Takto má obsluha stroje možnost použít o něco menší nástroj, než se předpokládal při výpočtu NC-drah.

Příklad:
NC-bloky byly počítány s frézou 10 mm. V tomto případě by obrobek mohl být opracováván také s frézou o průměru 9,9 mm, přičemž by se pak muselo počítat s jiným profilem povrchu.

© Siemens AG 2004, Všechna práva vyhrazena

8.5.2 Typy nástrojů/výměna nástroje se změněnými rozměry G40/41/42

Tvary fréz, parametry nástrojů
V této tabulce jsou shrnuty možné tvary fréz a mezní hodnoty parametrů nástroje pro čelní frézování.
Tvary těla frézy nejsou zohledňovány – typy nástrojů 120 a 155 jsou z hlediska efektu identické.
Jestliže je v NC programu uvedeno jiné číslo typu, než jaké se vypisuje v tabulce, systém automaticky použije typ 110: zápustková prořezávací fréza.
Při narušení mezních hodnot pro parametry nástroje se aktivuje alarm.



Typ frézy	Číslo typu	R	r	a
Válcová zápustková fréza	110	> 0	X	X
Fréza s kulovou hlavou	111	> 0	> R	X
Čelní stopková fréza, fréza s úhlovou hlavou	120, 130	> 0	X	X
Čelní stopková fréza, fréza s úhlovou hlavou a s rohovým zaoblením	121, 131	> r	> 0	X
Kuželová fréza	155	> 0	X	> 0

Údaje nástroje	Parametry nástroje		X = nevyhodnocuje se
Rozměry nástr.	Geometrie	Opotřebení	
R	$TC_DP6	$TC_DP15	**R** = rádius těla frézy (rádius nástroje)
r	$TC_DP7	$TC_DP16	**r** = rádius rohu
a	$TC_DP11	$TC_DP20	**a** = úhel mezi podélnou osou nástroje a horní hranou břitu

Korekce délky nástroje
Jako vztažný bod pro délkovou korekci platí špička nástroje (průsečík podélné osy a povrchu).

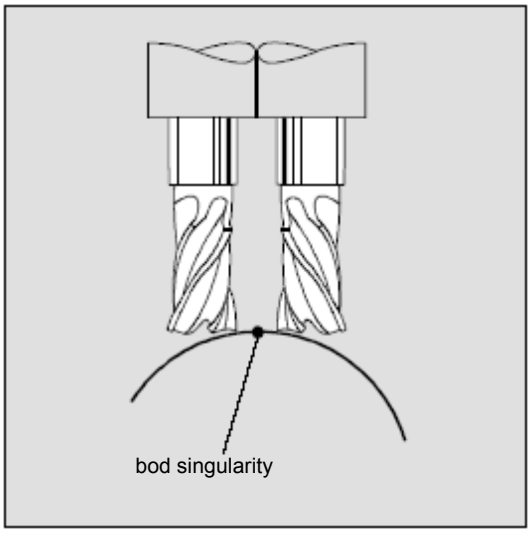
3D korekce nástroje, výměna nástroje
Nový nástroj se změněnými rozměry (R, r, a) nebo jiný tvar se smí udávat pouze s naprogramováním příkazu G41, příp. G42 (přechod G40 do G41, resp. G42, příp. nové naprogramování G41, resp. G42).
Všech ostatních údajů o nástroji, např. délek nástroje, se toto pravidlo netýká, takže tyto nástroje je možné vyměnit i bez opětovného zadání G41, resp. G42.

© Siemens AG 2004, Všechna práva vyhrazena

8.5.3 Korekce po dráze, zakřivení dráhy a hloubka zajíždění ISD

Korekce na dráze

U čelních fréz je potřeba dávat pozor na případ skokové změny bodu dotyku na povrchu nástroje, jak ukazuje příklad obrábění konvexní plochy s kolmo postaveným nástrojem. Použití uvedené na obrázku lze považovat za mezní případ.



Tento mezní případ je řídícím systémem monitorován, takže na základě úhlového uspořádání mezi nástrojem a vektorem normály plochy jsou rozpoznávány skokové změny opracovávaného bodu. Na těchto místech řídící systém vkládá lineární bloky, takže pohyb může být uskutečněn.

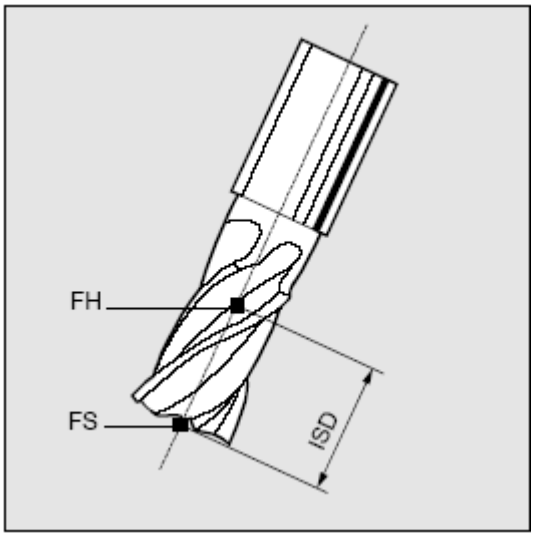
Pro výpočet těchto lineárních bloků jsou ve strojních parametrech uloženy rozsahy přípustných úhlů pro úhly směrnic.
Jestliže jsou překročeny mezní hodnoty pro úhlové rozsahy uložené ve strojních parametrech, aktivuje řídící systém alarm.

Zakřivení dráhy

Zakřivení dráhy není monitorováno. Také zde se doporučuje, abyste používali jen takové nástroje, s nimiž je možné obrobek opracovat bez narušení kontury.

Hloubka zajíždění ISD

Pomocí příkazu ISD (Insertion Depth) se naprogramuje hloubka zajíždění nástroje při obvodovém frézování. Takto je možné měnit polohu obráběného bodu na ploše pláště nástroje.



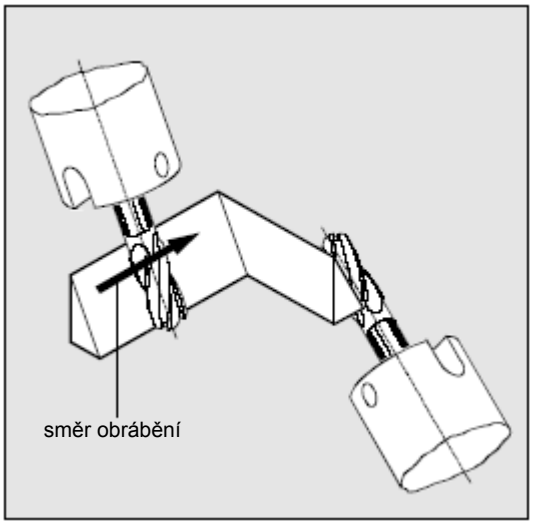
ISD udává vzdálenost mezi špičkou frézy (FS) a pomocným bodem frézy (FH). Bod FH vzniká promítnutím naprogramovaného bodu obrábění na osu nástroje. ISD se vyhodnocuje jen při aktivní 3D korekci rádiusu nástroje.

© Siemens AG 2004, Všechna práva vyhrazena

8.5.4 Posuv na vnitřních/vnějších rozích a průsečících, G450/G451 (od SW 5)

Vnitřní rohy / vnější rohy
Zacházení s vnitřními a vnějšími rohy je odlišné. Označení vnitřního nebo vnějšího rohu závisí na orientaci nástroje.

Při změně orientace v rohu se může vyskytnout případ, že se typ rohu během obrábění změní. Nastane-li tato situace, obrábění se přeruší a vypíše se chybové hlášení.



Zacházení s průsečíky pro 3D korekci
Při 3D obvodovém frézování se nyní na vnějších rozích vyhodnocuje G-kód G450/G451, tzn. je možné najet na průsečík offsetových křivek. Až do SW 4 se na vnějších rozích vždy vkládal kruhový oblouk.
Nové řešení, které je k dispozici od SW 5, je obzvláště výhodné pro typické 3D programy generované systémem CAD. Často se skládají z krátkých přímkových bloků (pro aproximaci hladkých křivek), přičemž přechody mezi sousedními bloky jsou téměř tangenciální.
Až dosud při korekci rádiusu nástroje na vnější straně kontury byly pro objíždění vnějších rohů v zásadě vkládány kruhové oblouky. Protože tyto bloky jsou při téměř tangenciálních přechodech velmi krátké, mělo to za následek nežádoucí poklesy rychlosti.
V těchto případech analogicky ke 2½D korekci rádiusu jsou obě podílející se křivky prodlouženy a najíždí se na průsečík těchto dvou prodloužených křivek.
Průsečík se určuje tak, že se offsetové křivky obou podílejících se bloků prodlouží, a stanoví se jejich průsečík v rohu v rovině kolmé na orientaci nástroje. Pokud takový průsečík neexistuje, bude roh obroben jako dříve, tzn. vloží se kruhový oblouk.

© Siemens AG 2004, Všechna práva vyhrazena



Další informace o chování v průsečících:
/FB/ W5, 3D korekce rádiusu nástroje.

8.5.5 Obvodové 3D frézování s omezovacími plochami, CUT3DCC, CUT3DCCD



Funkce

NC programy generované systémy CAD zpravidla aproximují dráhu středu normovaného nástroje velkým množstvím krátkých lineárních bloků. Aby takto vytvořené bloky mnoha dílčích kontur co možno nejpřesněji zobrazovaly původní konturu, je nezbytné uskutečnit ve výrobním programu určitá nastavení.

Důležité informace, které by byly zapotřebí pro optimální korekci, nejsou však ve výrobním programu obsaženy, takže musí být nahrazeny vhodnými opatřeními. V následujících odstavcích jsou uvedeny typické metody, které umožňují se vypořádat s kritickými přechody buď
přímo ve výrobním programu nebo
zjištěním reálné kontury, např. přísuvem nástroje.

Kromě typického případu je zde také pojednáno o případech, ve kterých normovaný nástroj neopisuje středovou dráhu, nýbrž konturu na obráběné ploše. Omezující plochy jsou přitom na nástroji nezávislé. Pro výpočet kolmých offsetů k omezovacím plochám se použije celkový rádius, stejně jako při obvyklé korekci rádiusu nástroje.

© Siemens AG 2004, Všechna práva vyhrazena

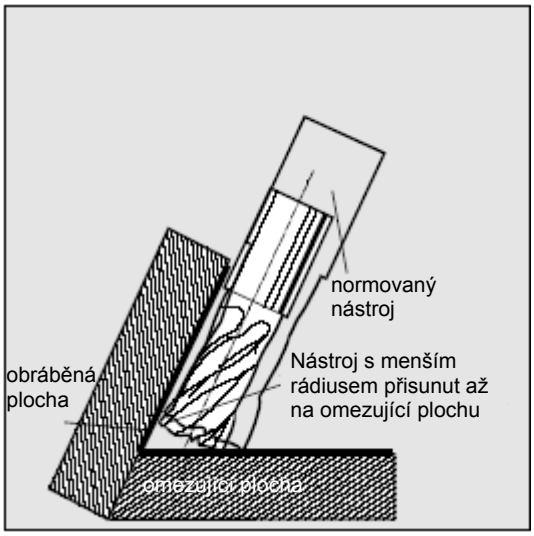
3D korekce rádiusu s normovanými nástroji
Při obvodovém 3D frézování se spojitou nebo konstantní změnou orientace nástroje se často programuje dráha středu pro definovaný normovaný nástroj. Protože se v praxi často stává, že vyhovující normovaný nástroj není k dispozici, je možné použít nástroj, který se od normovaného nástroje příliš neodlišuje.

Frézování dutin s šikmými bočními stěnami pro obvodové frézování s příkazem CUT3DC
U těchto 3D korekcí rádiusu nástroje se provádí kompenzace odchylky rádiusu frézy přísuvem ve směru normály plochy, která se má obrábět. Rovina, ve které leží čelní plocha frézy zůstává přitom nezměněna, pokud se nemění hloubka zajíždění ISD. Fréza s například menším rádiusem oproti normovanému nástroji by potom nedosáhla až na dno dutiny, které rovněž představuje omezující plochu. Pro automatický přísuv nástroje musí řídící systém znát tyto omezující plochy.



Další informace o monitorování kolizí viz:
PG/, Základy, „Korekce nástroje“.

Dráha středu nástroje s přísuvem až na omezující plochy, CUT3DCCD (od SW 6.4)
Pokud se používá nástroj, který má ve srovnání s vyhovujícím normovaným nástrojem menší rádius, pak se přísuv v jeho podélném směru prodlouží natolik, aby se znovu dotýkal dna dutiny. Tím bude roh, který vzniká mezi obráběnými a omezujícími plochami, obroben natolik, kam až to nástroj připouští. Jedná se přitom o smíšený způsob obrábění, který zahrnuje jak obvodové, tak i čelní obrábění.
Analogicky k nástroji s menším nástrojem se v případě nástroje s větším rádiusem přísuv upraví v opačném směru.



© Siemens AG 2004, Všechna práva vyhrazena

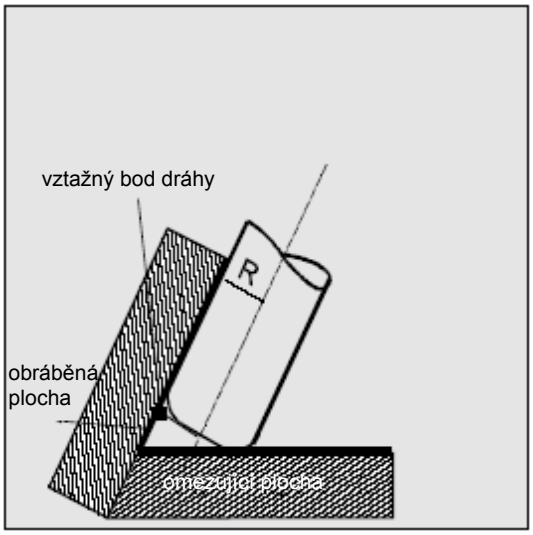
Použití válcových nástrojů

Při použití válcových nástrojů je přísuv zapotřebí jen tehdy, pokud obráběná a omezující plocha svírají ostrý úhel (menší než 90°). Jestliže jsou použity válcové frézy se zaoblením v rohu, je požadován jak u ostrých, tak i u tupých úhlů přísuv v podélném směru nástroje.

3D korekce nástroje s příkazem CUT3DCC (od SW 6.4)

Kontura na obráběné ploše

Pokud je aktivní příkaz CUT3DCC s válcovou frézou se zaoblením v rohu, je naprogramovaná dráha vztažena na fiktivní válcovou frézu stejného průměru. Vztažný bod dráhy, který z toho vyplývá, je při použití válcové frézy se zaoblením v rohu uveden na obrázku vpravo.



Je přípustné, aby se úhel mezi obráběnou a omezující plochou i v rámci jednoho bloku měnil z ostrého na tupý nebo obráceně.

Oproti normovanému nástroji smí být použitý reálný nástroj jak větší, tak i menší. Vznikající rohový rádius přitom nesmí být záporný a znaménko výsledného rádiusu nástroje musí zůstat zachováno.

© Siemens AG 2004, Všechna práva vyhrazena

Normované nástroje se zaoblením v rohu
Zaoblení normovaného nástroje v rohu je popsáno parametrem nástroje $TC_DP7. Z parametru nástroje $TC_DP16 vyplývá odchylka rohového zaoblení reálného nástroje oproti normovanému nástroji.

Příklad:
Rozměry nástroje pro válcovou frézu se zmenšeným rohovým zaoblením oproti normovanému nástroji.

Typ nástroje	**R = rádius frézy r = rádius zaoblení**
Normovaný nástroj se zaoblením v rohu	R = $TC_TP6 r = $TC_TP7
Reálný nástroj se zaoblením v rohu: typy nástrojů 121 a 131	R' = $TC_TP6 + $TC_TP15 + OFFN r' = $TC_TP7 + $TC_TP6
Přípustné jsou pouze typy fréz s válcovým tělem a válcové frézy se zaoblenými rohy. (typ 121 a 131) a v mezním případě válcové zápustkové frézy (typ 110) Až do SW 6.4 jsou **nepřípustné** kuželové frézy (typ 155 – 157) a válcové frézy se zaoblenou špičkou (typ 111)	U těchto povolených typů frézy je rádius v rohu r roven rádiusu těla R. Všechny ostatní povolené typy nástrojů jsou interpretovány jako válcové frézy a případně udávaný rozměr rohového zaoblení se nevyhodnocuje.
Přípustné jsou všechny typy nástrojů s čísly 1 – 399 s výjimkou čísel 111 a 155 – 157.	

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Dráha středu nástroje s přísuvem až na omezující plochy CUT3DCCD (od SW 6.4):
Oproti všem ostatním korekcím nástroje nemají G-kódy z 22. skupiny žádný vliv na rádius nástroje pro parametr nástroje $TC_DP6 uvedený pro příkaz CUT3DCCD a výslednou konturu neovlivňují. Korekční offset vyplývá z

- hodnoty opotřebení (parametr nástroje $TC_DP15) a
- programovatelného offsetu OFFN, pro výpočet offsetů kolmých na omezovací plochu.

Jestli obráběná plocha leží vlevo nebo vpravo od dráhy, není možné ze sestaveného výrobního programu zjistit. Vychází se proto z kladného rádiusu a záporné hodnoty opotřebení původního nástroje. Záporná hodnota opotřebení vždy popisuje nástroj se zmenšeným průměrem.

Pro korekci rádiusu nástroje s příkazy G41, G42 s aktivním příkazem CUT3DCCD nebo CUT3DCC musí být k dispozici volitelný doplněk „Transformace orientace“.

3D korekce rádiusu s CUT3DCC (od SW 6.4)
Kontura na obráběné ploše
U příkazu CUT3DCC se vztahuje výrobní program na konturu na obráběné ploše. Přitom se používá, stejně jako při obvyklé korekci rádiusu nástroje, celkový rádius, který se skládá ze součtu

- rádiusu nástroje (parametr nástroje $TC_DP6)
- hodnoty opotřebení (parametr nástroje $TC_DP15)

a pro výpočet kolmých offsetů k omezující ploše

- programovatelného offsetu OFFN.

Poloha omezující plochy bude stanovena z rozdílu mezi oběma hodnotami

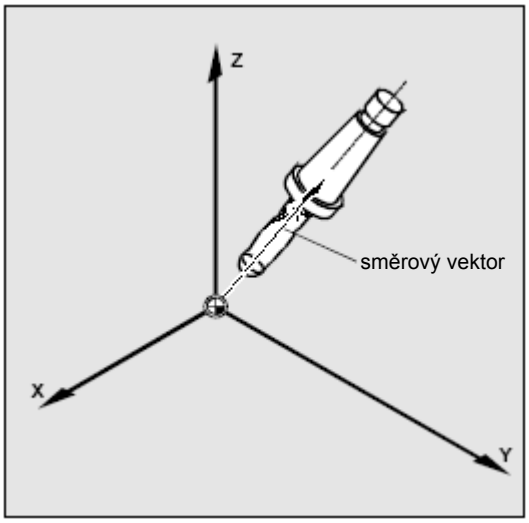
- rozměry normovaného nástroje a
- rádius nástroje (parametr nástroje $TC_DP6)

© Siemens AG 2004, Všechna práva vyhrazena

8.6 Orientace nástroje, ORIC, ORID, OSOF, OSC, OSS, OSSE

Orientací nástroje se rozumí geometrické nastavení nástroje v prostoru.

V případě obráběcího stroje s 5 osami je orientace nástroje nastavitelná pomocí programových příkazů.

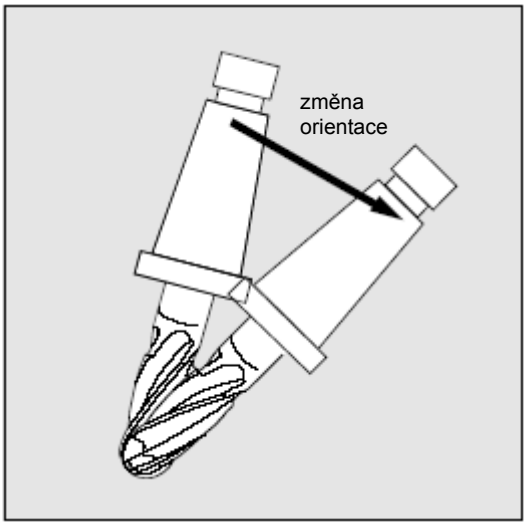


Programování orientace nástroje

Změna orientace nástroje může být naprogramována těmito prostředky:
Přímé programování kruhových os
Eulerův úhel nebo úhel RPY
Směrový vektor
LEAD/TILT (čelní frézování)

Vztažným souřadným systémem je buď souřadný systém stroje (ORIMCS) nebo aktuální souřadný systém obrobku (ORIWCS).

Změna orientace může být ovlivňována následujícími příkazy:

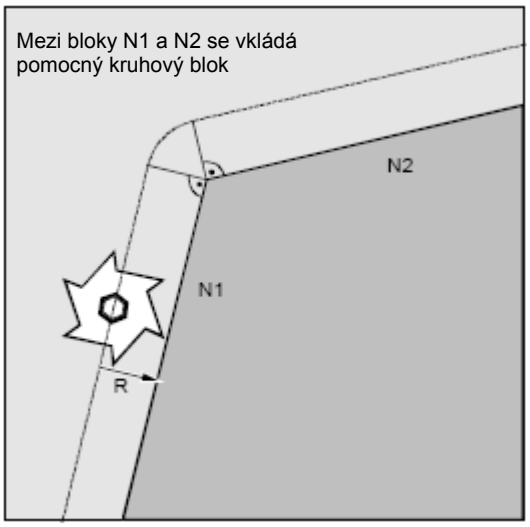


ORIC	Orientace a pohyb po dráze paralelně.
ORID	Orientace a pohyb po dráze jedno po druhém.
OSOF	Žádné vyhlazení změny orientace
OSC	Konstantní orientace
OSS	Vyhlazení orientace jen na začátku bloku
OSSE	Vyhlazení orientace na začátku a na konci bloku
ORIS	Rychlost změny orientace při aktivovaném vyhlazení změny orientace ve stupních/mm; platí pro OSS a OSSE.

© Siemens AG 2004, Všechna práva vyhrazena

Chování na vnějších rozích

Na vnějších rozích se vždy vkládá kruhový blok s rádiusem frézy.



Pomocí programových příkazů ORIC, resp. ORID může být definováno, zda změny orientace, které byly naprogramovány mezi bloky N1 a N2, mají být provedeny před začátkem vkládaného kruhového bloku nebo současně s ním.

Pokud je na vnějších rozích potřebná změna orientace, je možné zvolit, zda se má uskutečnit souběžně s interpolací nebo odděleně spolu s pohybem po dráze.

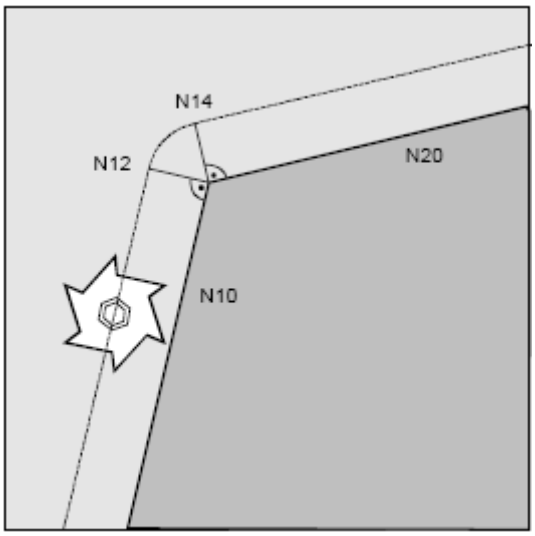
U příkazu ORID se provádí napřed vkládaný blok bez pohybu po dráze. Kruhový blok se vkládá bezprostředně před druhý z obou bloků posuvu, jež tvoří roh.

Pokud je na vnějším rohu vloženo více bloků změny orientace a je-li zvolen příkaz ORIC, bude kruhový pohyb rozložen v souladu se složkami změny orientace do jednotlivých vkládaných bloků.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování pro ORIC

Pokud jsou mezi bloky posuvu N10 a N20 naprogramovány dva nebo více bloků se změnou orientace (např. A2=, B2=, C2=) a je-li příkaz ORIC aktivní, bude vložený kruhový blok rozložen podle složek úhlových změn na tyto pomocné bloky.

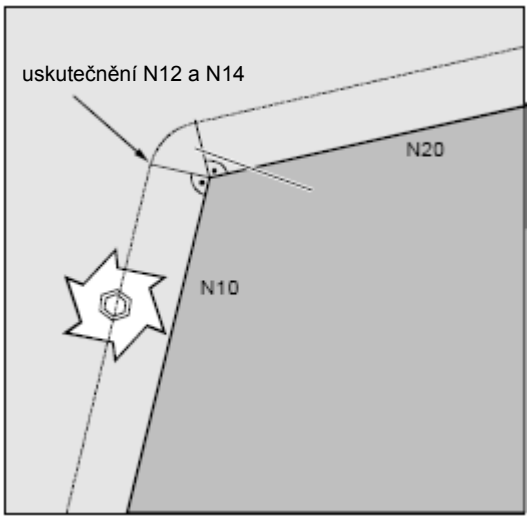


`ORIC`	
`N8 A2=... B2=... C2=...`	
`N10 X... Y... Z...`	
`N12 C2=... B2=...` `N14 C2=... B2=...`	Kruhový blok, který se vkládá na vnější roh, je rozdělen na bloky N12 a N14 v souladu se změnou orientace. Kruhový pohyb a změna orientace se přitom uskuteční souběžně.
`N20 X=... Y=... Z=... G1 F200`	

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování pro ORID

Je-li aktivní příkaz ORID, všechny bloky mezi oběma bloky posuvu se budou provádět na konci prvního bloku posuvu. Kruhový blok s konstantní orientací se uskuteční bezprostředně před druhým blokem posuvu.



`ORID`	
`N8 A2=... B2=... C2=...`	
`N10 X... Y... Z...`	
`N12 A2=... B2=... C2=...`	Blok N12 a N14 se provede na konci bloku N10. Potom následuje pohyb kruhového bloku s aktuální orientací.
`N14 M20`	Pomocné funkce atd.
`N20 X=... Y=... Z=...`	

Pro druh změny orientace na vnějším rohu je rozhodující programový příkaz, který je aktivní v prvním bloku posuvu vnějšího rohu.

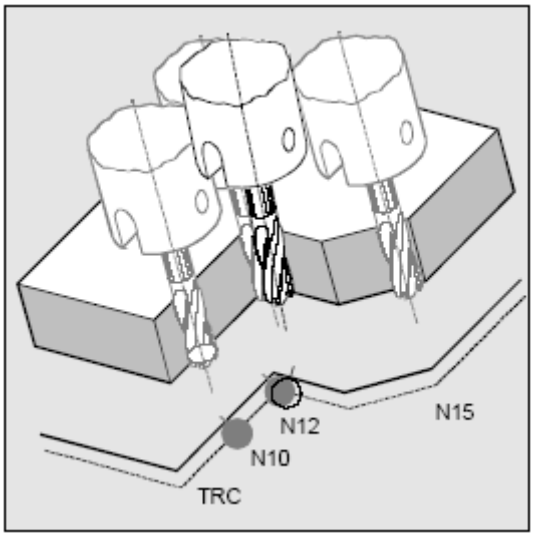
© Siemens AG 2004, Všechna práva vyhrazena

Bez změny orientace

Pokud se orientace na hranicích bloků nemění, je průřezem nástroje kruh, který se dotýká obou kontur.

Příklad programování

Změna orientace na vnitřním rohu




```
ORIC
N10 X... Y... Z... G1 F500
N12 X... Y... Z... A2=... B2=... C2=...
N15 X Y Z A2 B2 C2
```

© Siemens AG 2004, Všechna práva vyhrazena

8.7 Volné předávání D-čísel, číslo břitu CE (od SW 5)

Od SW 5 se mohou D-čísla používat jako čísla korekcí. Kromě toho lze pomocí adresy CE uvádět čísla břitů.
Pomocí systémové proměnné $TC_DPCE může být číslo břitu popsáno. Předem definováno je, že číslo korekce == číslu břitu.
Literatura: FB, W1 (korekce nástroje).



Výrobce stroje (MH 8.12)

Prostřednictvím strojních parametrů je definován maximální počet D-čísel (čísel břitů) a maximální počet břitů na jeden nástroj. Následující příkaz má smysl jen tehdy, pokud byl stanoven maximální počet břitů (MD 18105) větší než počet břitů na jeden nástroj (MD 18106). Věnujte pozornost údajům výrobce stroje.



Další upozornění

Kromě uvádění relativních D-čísel mohou být D-čísla předávána také jako „prostá“, příp. „absolutní"“D-čísla (1-32000) bez vztahu na T-číslo (v rámci funkce „prostá struktura D-čísel“).

© Siemens AG 2004, Všechna práva vyhrazena

8.7.1 Kontrola D-čísla (CHKDNO)



Programování

```
stav=CHKDNO(Tno1,Tno2,Dno)
```



Vysvětlení příkazů

`stav`	TRUE:	D-čísla byla pro kontrolovanou oblast jednoznačně předána.
	FALSE:	Vyskytuje se kolize D-čísel nebo dosazení parametrů je neplatné. Pomocí Tno1, Tno2 a Dno jsou předávány parametry, které vedou ke kolizi. Tato data mohou být ve výrobním programu vyhodnocována.
`CHKDNO (Tno1,Tno2)`	Spustí se kontrola D-čísel uvedených nástrojů.	
`CHKDNO (Tno1)`	Spustí se kontrola všech D-čísel nástroje Tno proti všem ostatním nástrojům.	
`CHKDNO`	Uskuteční se kontrola všech D-čísel všech nástrojů proti ostatním nástrojům.	



Funkce

Příkazem CHKNO zkontrolujete, zda jsou existují D-čísla předávána jednoznačně.D-čísla všech nástrojů definovaných v jedné jednotce TO se smí vyskytnout jen jednou. Na náhradní nástroje se přitom nebere ohled.

© Siemens AG 2004, Všechna práva vyhrazena

8.7.2 Změna názvu D-čísla (GETDNO, SETDNO)

Programování

```
d = GETDNO (t,ce)
stav = SETDNO (t,ce,d)
```

Vysvětlení příkazů

d	D-číslo břitu nástroje
t	T-číslo nástroje
ce	Číslo břitu nástroje (CE-číslo)
stav	Udává, zda příkaz mohl být uskutečněn bezchybně (TRUE nebo FALSE).

Funkce

GETDNO

Tento příkaz dodává D-číslo určitého břitu (ce) nástroje s T-číslem t. Pokud k zadaným parametrům žádné D-číslo neexistuje, bude dosazeno d=0. Jestliže je D-číslo neplatné, bude vrácena hodnota větší než 32000.

SETDNO

Pomocí tohoto příkazu přiřazujete hodnotu d D-číslu břitu (ce) nástroje (t). Prostřednictvím stavu operace poskytuje výsledek tohoto příkazu (TRUE nebo FALSE).
Pokud pro zadané parametry neexistuje žádný datový blok, je vracena hodnota FALSE. Syntaktická chyba aktivuje alarm. D-číslu není možné explicitně dosadit nulu.

Příklad: (změna názvu D-čísla)

```
$TC_DP2[1,2]  = 120
$TC_DP3[1,2]  = 5.5
$TC_DPCE[1,2] = 3;   číslo břitu CE
...
N10 def int DNrALT, DNrNEU = 17
N20 DNrALT = GETDNO(1,3)
N30 SETDNO(1,3,DNrNEU)
```

Tím bude břitu CE=3 přiřazena nová hodnota D-čísla 17.

© Siemens AG 2004, Všechna práva vyhrazena

Nyní budou údaje tohoto břitu přístupné pomocí D-čísla 17; jak prostřednictvím systémových proměnných, tak i v dosazování parametrů s NC adresou.

Další upozornění

Zadaná D-čísla musí být jednoznačná. Dva různé břity nástroje nesmí mít stejné D-číslo.

8.7.3 Zjištění T-čísla k předem zadanému D-číslu (GETACTTD)

Programování

```
stav = GETACTTD (Tnr, Dnr)
```

Vysvětlení příkazů

`Dnr`	D-číslo, pro které má být vyhledáno T-číslo.
`Tnr`	Nalezené T-číslo.
`stav`	0: T-číslo bylo nalezeno. Tnr obsahuje hodnotu T-čísla. -1: K uvedenému D-číslu neexistuje žádné T-číslo; Tnr=0 -2: D-číslo není absolutní. Tnr obsahuje hodnotu prvního nalezeného nástroje, který obsahuje D-číslo s hodnotou Dnr. -5: Funkce nemohla být uskutečněna z nějakého jiného důvodu.

Funkce

Pomocí příkazu GETACTTD zjišťujete k absolutnímu D-číslu odpovídající T-číslo. Neprovádí se žádná kontrola jednoznačnosti. Pokud v rámci jedné jednotky TO existuje větší počet stejných D-čísel, výsledkem bude T-číslo prvního nalezeného nástroje. Při použití „prostých“ D-čísel použití tohoto příkazu nemá smysl, protože výsledkem je vždy hodnota 1 (mezi daty se žádné T-číslo nenachází).

© Siemens AG 2004, Všechna práva vyhrazena

8.7.4 Prohlášení D-čísla za neplatné



Programování

DZERO



Vysvětlení příkazů

ZERO	Prohlášení všech D-čísel v rámci jednotky TO za neplatné.

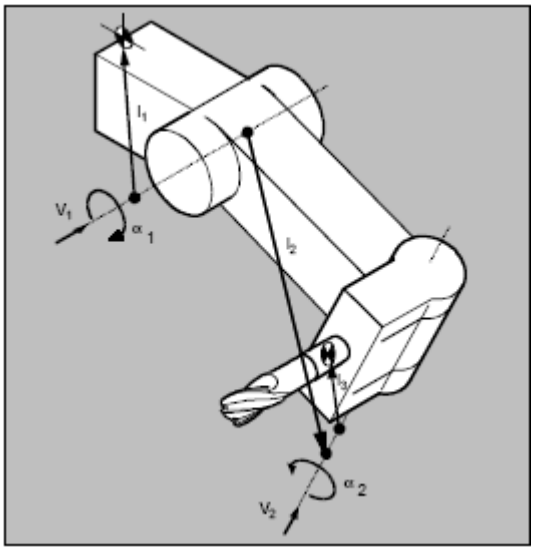


Funkce

Příkaz souží jako pomocná funkce při změně vybavení stroje.
Takto označené bloky korekčních dat nejsou kontrolovány příkazem CHKDNO. Aby tato data byla znovu přístupná, je nutno D-čísla znovu dosadit příkazem SETDNO.

© Siemens AG 2004, Všechna práva vyhrazena

8.8 Kinematika držáku nástroje



Kinematika držáku nástroje s maximálně dvěma kruhovými osami je popsána 17 systémovými proměnnými $TC_CARR1[m] až $TC_CARR17[m]. Popis držáku nástroje se skládá z následujících součástí:

- Vektorová vzdálenost od první kruhové osy ke vztažnému bodu držáku nástroje l_1, vektorová vzdálenost od první ke druhé kruhové ose l_2 a vektorová vzdálenost od druhé kruhové osy ke vztažnému držáku nástroje l_3.
- Směrové vektory obou kruhových os V_1 a V_2.
- Úhel otočení α_1, α_2 okolo obou os. Úhel otočení se považuje za kladný ve směru hodinových ručiček při pohledu ve směru vektoru kruhové osy.

Rozložená kinematika (od SW 5.3)

Pro stroje s rozloženou kinematikou (jak nástroj, tak i obrobek jsou otočné) jsou systémové proměnné rozšířeny o $TC_CARR18[m] až $TC_CARR23[m], které jsou popsány následujícím způsobem:

Otočný nástrojový stůl se skládá z:

- vektorové vzdálenosti druhé kruhové osy V_2 ke vztažnému bodu otočného nástrojového stolu l_4 třetí kruhové osy.

Kruhová osa má tyto součásti:

- Oba kanálové identifikátory pro popis vztahu mezi kruhovými osami V_1 a V_2. Tyto pozice jsou vyvolávány podle potřeby, aby se stanovila orientace orientovatelného držáku vřetena.

Typ kinematiky s hodnotami T, P nebo M:

- Typ kinematiky T: Otočný je pouze nástroj.
- Typ kinematiky P: Otočný je pouze obrobek.
- Typ kinematiky M: Otočný je jak nástroj, tak i obrobek.

© Siemens AG 2004, Všechna práva vyhrazena



Funkce systémových proměnných pro orientovatelný držák nástroje

Označení	Složka X	Složka Y	Složka Z
Vektor offsetu I_1	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
Vektor offsetu I_2	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
Kruhová osa V_1	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
Kruhová osa V_2	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
Úhel otočení α_1 Úhel otočení α_2	$TC_CARR13[m] $TC_CARR14[m]		
Vektor offsetu I_3	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]
Vektor offsetu I_4	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
Identifikátor kruhové osy V_1 kruhové osy V_2	Identifikátor kruhových os V_1 a V_2 (předdefinované nastavení je nula) $TC_CARR21[m] $TC_CARR22[m]		
Typ kinematiky Předdefinované nastavení T	$TC_CARR23[m]		
	Typ kinematiky T ⇒	Typ kinematiky P ⇒	Typ kinematiky M ⇒
	Pouze nástroj je otočný (**T**ool)	Pouze obrobek je otočný (**P**art)	Nástroj i obrobek jsou otočné (**M**ixed)
Offset kruhové osy V_1 kruhové osy V_2	Úhel ve stupních kruhových os V_1 a V_2 při zadávání základ. nastavení $TC_CARR24[m] $TC_CARR25[m]		
Úhlový offset kruhové osy V_1 kruhové osy V_2	Offset Hirthova ozubení kruhových os V_1 a V_2 ve stupních $TC_CARR21[m] $TC_CARR22[m]		
Úhlový inkrement kruhové osy V_1 kruhové osy V_2	Inkrement Hirthova ozubení kruhových os V_1 a V_2 ve stupních $TC_CARR21[m] $TC_CARR22[m]		
Minimální pozice kruhové osy V_1 kruhové osy V_2	Softwarová minimální koncová poloha kruhových os V_1 a V_2 $TC_CARR21[m] $TC_CARR22[m]		
Maximální pozice kruhové osy V_1 kruhové osy V_2	Softwarová maximální koncová poloha kruhových os V_1 a V_2 $TC_CARR21[m] $TC_CARR22[m]		

© Siemens AG 2004, Všechna práva vyhrazena

Parametry kruhových os od SW 6.1
Systémové proměnné byly rozšířeny o položky $TC_CARR24[m] až $TC_CARR33[m]. Jsou popsány takto:

Offset kruhových os

- Změna polohy kruhových os V_1 nebo V_2 při základním nastavení orientovatelného držáku nástroje.

Úhlový offset/úhlový inkrement kruhových os

- Offset nebo úhlový inkrement Hirthova ozubení kruhových os V_1 nebo V_2. Naprogramovaný nebo vypočítaný úhel bude zaokrouhlen na nejbližší hodnotu, která vyplývá ze vztahu
  `phi = s + n * d`.

Minimální/maximální pozice kruhové osy

- Mezní úhel (softwarová koncová poloha) kruhové osy V_1 nebo V_2.

Rozšíření systémových proměnných pro orientovatelný držák nástroje (od SW 6.4)

Označení	Složka X	Složka Y	Složka Z
Držák nástroje Název	Namísto čísla může být držák nástroje opatřen názvem. $TC_CARR34[m]		
Uživatel Název osy 1 Název osy 2 Identifikace	Zamýšlené použití v rámci uživatelských měřicích cyklů. $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
	Složka X	Složka Y	Složka Z
Pozice	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
Jemné posunutí	Parametr, který je možné přičíst k hodnotám v základních parametrech		
Vektor offsetu I_1	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
Vektor offsetu I_2	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
Vektor offsetu I_3	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
Vektor offsetu I_4	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
Kruhová osa V_1	$TC_CARR64[m]		
Kruhová osa V_2	$TC_CARR65[m]		



Další upozornění
Za „m“ je potřeba dosadit příslušné číslo držáku nástroje, jehož potřebujete naprogramovat.

© Siemens AG 2004, Všechna práva vyhrazena

Počáteční a konečné body vektorů vzdálenosti na osách můžete zvolit dle libosti. Úhly otočení α_1, α_2 okolo obou os jsou v základním postavení držáku nástroje definovány jako 0°. Možnosti popisu kinematiky držáku nástroje jsou tak téměř nevyčerpatelné.

Držák nástroje s jen jednou nebo žádnou kruhovou osou může být popsán dosazením nuly do směrových vektorů jedné nebo obou rotačních os. U držáku nástroje bez kruhových os se vektory vzdálenosti chovají jako dodatečné korekce nástroje, jejichž složky nejsou při přepnutí roviny obrábění G17 až G19 nijak ovlivněny.

Vymazání dat držáku nástroje
Pomocí $TC_CARR1[0] = 0 mohou být data všech datových bloků držáku nástroje vymazána.

Od SW 5.3
Typu kinematiky $TC_CARR23[T] = T musí být dosazeno jedno ze tří přípustných velkých nebo malých písmen (T, P, M) a z tohoto důvodu by tento parametr neměl být vymazán.

Editace parametrů držáku nástroje
Každá z popisovaných hodnot může být změněna přiřazením nových hodnot ve výrobním programu. Jakýkoli jiný znak než T, P nebo M má za následek alarm při pokusu orientovatelný držák nástroje aktivovat.

Načítání parametrů držáku nástroje
Každá z popisovaných hodnot může být ve výrobním programu čtena přiřazením nějaké proměnné.

Od SW 6.4
$TC_CARR34[m] až $TC_CARR40[m] obsahují parametry, které jsou uživateli volně k dispozici a až do SW 6.4 nejsou standardně v rámci NCK vyhodnocovány nebo nemají žádný význam.

© Siemens AG 2004, Všechna práva vyhrazena

$TC_CARR41[m] až $TC_CARR65[m] obsahují parametry jemného posunutí, které se mohou přičítat k hodnotám základních parametrů. Hodnotu jemného posunutí, která je přiřazena určitému základnímu parametru, dostanete tak, že k číslu parametru přičtete hodnotu 40.

$TC_CARR47[m] až $TC_CARR54[m], jakož i $TC_CARR61[m] až $TC_CARR63[m] nejsou definovány a při pokusu je číst nebo do nich zapisovat se aktivuje alarm.

Okrajové podmínky

Držák nástroje může nástroj orientovat do všech možných prostorových směrů jen tehdy, pokud:

- jsou k dispozici dvě kruhové osy V_1 a V_2
- kruhové osy jsou vůči sobě kolmé
- podélná osa nástroje se nachází kolmo na druhou kruhovou osu V_2.

Od SW 5.3

Kromě toho platí u strojů, u nichž musí být nastavitelné všechny možné orientace, následující požadavek:

- Orientace nástroje musí být kolmo na první kruhovou osu V_1.

Jemné posunutí (od SW 6.4)

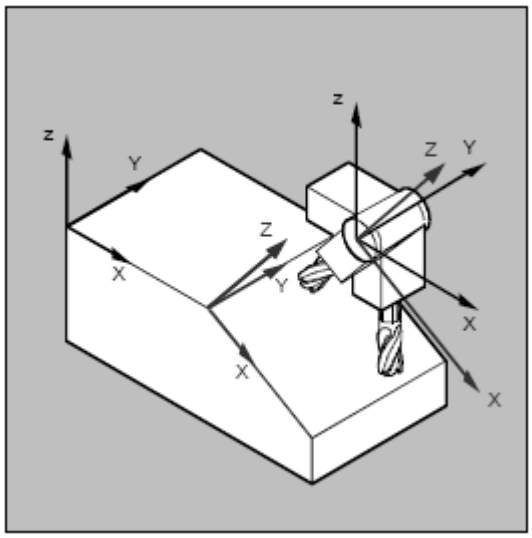
Nepřípustná hodnota jemného posunutí bude rozpoznána až tehdy, když je aktivován orientovatelný držák nástroje, který obsahuje takovou hodnotu, a současně se v nastavovaném parametru SD 42974: TOCARR_FINE_CORRECTION nachází hodnota TRUE.

Přípustná hodnota jemného posunutí je strojními parametry omezena na určité maximum.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Držák nástroje použitý v následujícím příkladu může být plně popsán otáčením okolo osy Y.



`N10 $TC_CARR8[1]=1`	Definice složky Y první kruhové osy držáku nástroje 1
`N20 $TC_DP1[1,1]=120`	Definice válcové frézy
`N30 $TC_DP3[1,1]=20`	s délkou 20 mm
`N40 $TC_DP6[1,1]=5`	a s rádiusem 5 mm
`N50 ROT Y37`	Definice framu s otočením o 37° okolo osy Y
`N60 X0 Y0 Z0 F10000`	Najíždění na počáteční pozici
`N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10`	Nastavení korekce rádiusu a korekce délky nástroje v otočeném framu, aktivování držáku nástroje 1, nástroje 1
`N80 X40`	Obrábění s otočením o 37°
`N90 Y40`	
`N100 X0`	
`N110 Y0`	
`N120 M30`	

© Siemens AG 2004, Všechna práva vyhrazena

Chování při pohybu po dráze

9.1 Tangenciální řízení,TAN, TANGON, TANGOF, TLIFT, TANGDEL 9-362

9.2 Vlečení, TRAILON, TRAILOF 9-369

9.3 Tabulky křivky, CTABDEF, CTABEND, CTABDEL, CTAB, CTABINV/CTABFNO 9-373

9.3.1 Příkazy s tabulkami křivky, CTABIB, CTABLOCK, CTABUNLOCK 9-373

9.3.2 Chování na okrajích tabulky křivky, CTABTSV/CTABTSP, CTABTMIN 9-383

9.3.3 Přístup na pozice a segmenty v tabulce křivky CTAB, CTABINV 9-386

9.4 Spojení os řídící hodnotou, LEADON, LEADOF 9-388

9.5 Průběh posuvu, FNORM, FLIN, FCUB, FPO 9-394

9.6 Zpracování programu s pamětí předběžného zpracování, STARTFIFO, STOPFIFO, STOPRE 9-399

9.7 Úseky programu s podmíněným přerušením, DELAYFSTON, DELAYFSTOF 9-401

9.8 Zablokování programového místa pro SERUPRO, IPTRLOCK, IPTRUNLOCK 9-407

9.9 Opětovné najíždění na konturu, REPOSA/L, REPOSQ/H, RMI/N, RMB/E.. 9-410

© Siemens AG 2004, Všechna práva vyhrazena

9.1 Tangenciální řízení,TAN, TANGON, TANGOF, TLIFT, TANGDEL

Programování

```
TANG (FOsa, LOsa1, LOsa2, Spojení, KS, Opt)
TANGON (FOsa, Úhel, Dist, Winkeltol)
FANGOF (FOsa)
TLIFT (FOsa)
TANGDEL (FOsa)
```

Vysvětlení příkazů

TANG	Přípravný příkaz pro definici tangenciálního sledování
TANGON	Aktivování tangenciálního řízení s udáním vlečné osy a úhlu offsetu, případně také zaoblení a úhlové odchylky
TANGOF	Deaktivování tangenciálního řízení s udáním vlečné osy
TLIFT	Vkládání pomocného bloku na rohu kontury
TANGDEL	Vymazání definice tangenciálního sledování

Vysvětlení parametrů

FOsa	Vlečná osa: tangenciálně řízená doplňková kruhová osa
LOsa1, LOsa2	Řídící osy: Dráhové osy, z nichž se určuje tečna pro sledování
Spojení	Faktor spojení: Souvislost mezi změnou úhlu tečny a vlečenou osou. Parametr není povinný; předdefinované nastavení: 1
KS	Identifikační písmeno souřadného systému: „B“ = základní souřadný systém; údaj nepovinný, předdefinováno:1 [„W“ = souřadný systém obrobku]
Opt	Optimalizace: „S“ Standardní (jako až do SW 6) „P“ Automatické přizpůsobení časového průběhu ose tečny a kontuře
Úhel	Úhel offsetu vlečné osy
Dist	Dráha zaoblení vlečné osy, potřeba při zvolení Opt = „P“
Winkeltol	Úhlová tolerance pro vlečnou osu (není nutné), vyhodnocuje se pouze když je zvoleno Opt = „P“.

© Siemens AG 2004, Všechna práva vyhrazena

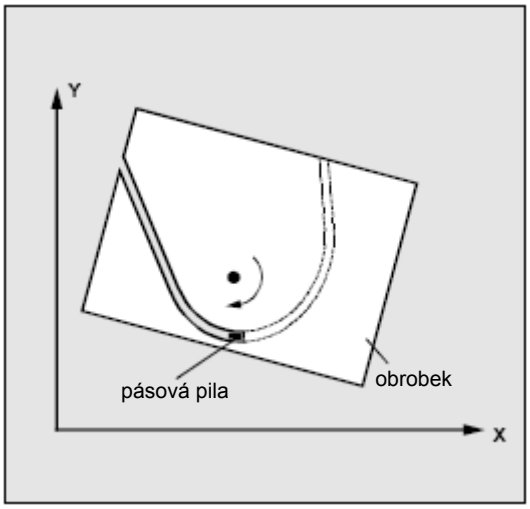
Funkce

Vlečná osa je naváděna podle tečny dráhy definované řídícími osami. Tak může být nástroj nastaven rovnoběžně s konturou. Nástroj může být nastaven vůči tečně o určitý úhel, který lze naprogramovat pomocí příkazu TANGON.

Oblasti použití

Tangenciální řízení může být mimo jiné použito v těchto případech:

- Tangenciální nastavení otočného nástroje při vystřihování
- Navádění směru obrobku u pásových pil (viz obrázek)
- Nastavování polohy orovnávacího nástroje na brusném kotouči (viz obrázek dole)
- Nastavování polohy řezacích koleček při zpracování skla nebo papíru
- Tangenciální přivádění drátu při pětiosém svařování



Postup

Definice vlečných a řídících os

Definice vlečných a řídících os se provádí příkazem TANG.

Faktor spojení udává souvislost mezi úhlovou změnou tečny a vlečenou osou. Jeho hodnota je zpravidla 1 (předdefinované nastavení).

Sledování může probíhat v základním souřadném systému „B“ (předdefinované nastavení) nebo v souřadném systému obrobku „W“.

Příklad:

```
TANG(C,X,Y,1,"B")
```

Znamená:

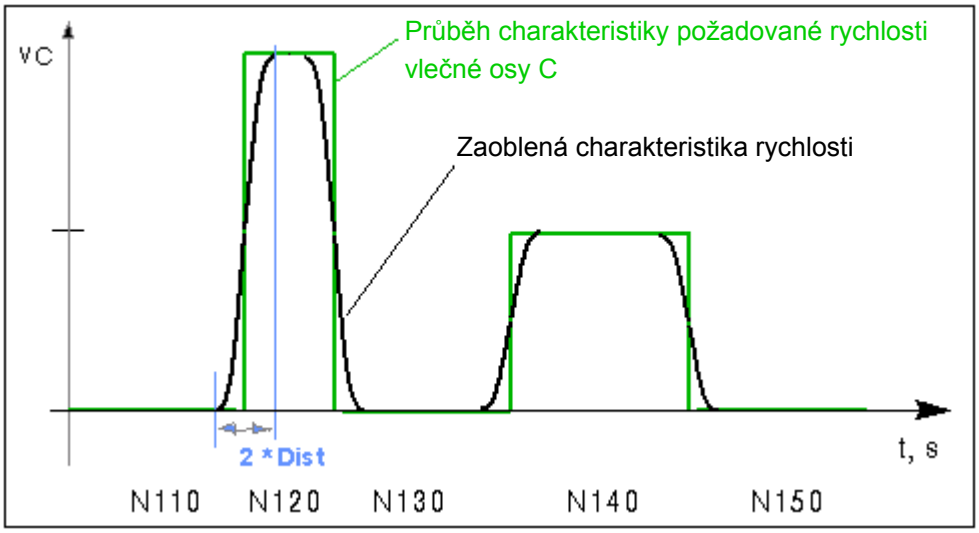
Kruhová osa C sleduje geometrické osy X a Y.

Možnosti optimalizace od verze SW 7

Jestliže je nastaveno `Opt="P"`, bude zohledňována dynamika sledující osy při omezení rychlosti řídící osy.

© Siemens AG 2004, Všechna práva vyhrazena

Parametry (`Dist` a `Winkeltol`) cíleně omezují chybu mezi sledující osou a tečnou řídící osy. Skokové změny rychlosti vlečné osy v důsledku skokových změn na řídící kontuře jsou pomocí parametrů `Dist` a `Winkeltol` zaobleny a vyhlazeny.
Vlečná osa je přitom řízena s využitím vyhodnocení budoucích kroků (viz diagram), aby odchylky byly udrženy co možno nejmenší.
Požití nastavení `Opt="P"` se doporučuje především při kinematických transformacích.



Zjednodušené programování
Faktor spojení 1 nemusí být explicitně naprogramován. Příkaz TANG(C, X, Y, 1, “B“, “P“) může být zkráceně zapsán TANG(C, X, Y,,,“P“). Stejně jako dříve může být zápis TANG(C, X, Y, 1, “B“, “S“) zapsán jako TANG(C, X, Y).

Příkaz TLIFT(...) zadáváte v návaznosti na přiřazení os pomocí příkazu TANG(...). Příklad:

```
TANG(C,X,Y,...)
TLIFT(C)
```

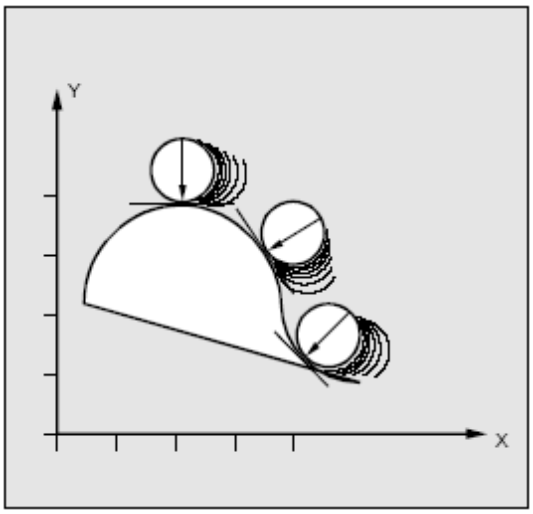
Deaktivování příkazu TLIFT
Za tímto účelem opakujte přiřazení os příkazem TANG(...) bez následujícího příkazu TLIFT(...).

© Siemens AG 2004, Všechna práva vyhrazena

Aktivování/deaktivování tangenciálního řízení: TANGON, TANGOF

Aktivování tangenciálního řízení se provádí příkazem TANGON a zadáním vlečné osy a požadovaného úhlu offsetu vlečné osy:

```
TANGON(C,90)
```



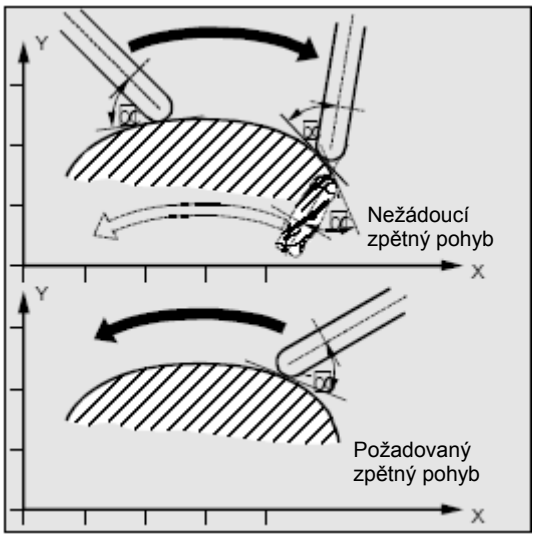
Znamená:
Osa C je vlečnou osou. Při každém pohybu dráhové osy se bude otáčet do polohy 90° vůči tečně ke dráze.

Pokud budete chtít tangenciální řízení vlečné osy deaktivovat, zadejte:

```
TANGOF(C)
```

Mezní úhel v důsledku ohraničení pracovního pole

Při pohybech po dráze tam a zpátky se tečna v bodě obratu skokově obrací o 180° a podle toho se mění i nastavení vlečné osy.
Toto chování obvykle postrádá smysl: zpětný pohyb by se měl uskutečňovat se stejným záporným úhlem offsetu jako pohyb tam.



Kvůli tomu je vhodné ohraničení pracovního pole vlečné osy (G25, G26). Ohraničení pracovního pole musí být v okamžiku převrácení směru pohybu aktivní (WALIMON).

Pokud úhel offsetu leží mimo ohraničení pracovního pole, bude učiněn pokus znovu najet do pracovní oblasti se záporným offsetovým úhlem.

Software verze 7:
Pokud je nastaveno Opt="P", tato funkce není aktivní. Dosažení ohraničení pracovního pole v tomto případě způsobí zastavení a alarm.

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Vkládání pomocného bloku na rozích kontury, TLIFT

V rozích kontury se skokově mění tečna a tím pádem i požadovaná pozice naváděné osy. Za normálních okolností se osa snaží vyrovnat tento skok svou maximální možnou rychlostí. Přitom však na určité dráze na kontuře za rohem vzniká odchylka od požadovaného tangenciálního nastavení. Pokud je tato odchylka z technologických důvodů nepřípustná, může být řídící systém přinucen příkazem TLIFT, aby se pohyb v rohu pozastavil a aby se v automaticky vytvořeném pomocném bloku naváděná osa pootočila do nového tangenciálního směru.

Otáčení se uskutečňuje s naprogramovanou dráhovou osou, pokud se s naváděnou osou alespoň jednou pohybovalo jako s osou dráhovou. Pomocí funkce FGREF[ax] = 0.001 je zde možné stanovit maximální rychlost, jíž může naváděná osa dosáhnout.

Jestliže se zde s naváděnou osou dosud nepracovalo jako s osou dráhovou, bude se s ní pohybovat jako s osou polohovací. Rychlost je pak závislá na rychlosti polohování uložené v příslušném strojním parametru.

Otáčení se uskutečňuje s maximální rychlostí naváděné osy.

Úhlová změna, od které se automaticky vkládá pomocný blok, je definována strojním parametrem `$MA_EPS_TLIFT_TANG_STEP`.

© Siemens AG 2004, Všechna práva vyhrazena

Vymazání definice tangenciálního sledování
Existující uživatelem definované tangenciální sledování musí být vymazáno, pokud má být definováno nové tangenciální sledování se stejnou vlečnou osou v příkazu přípravného volání TANG.

`TANGDEL (FOsa)`	Vymazání tangenciálního sledování

Vymazání je možné jen tehdy, pokud je pomocí příkazu TANGOF(FOsa) deaktivováno spojení.



Příklad programování

Příklad se změnou roviny

`N10 TANG(A, X, Y, 1)`	1. definice tangenciálního sledování
`N20 TANGON(A)`	Aktivování spojení
`N30 X10 Y20`	
`...`	
`N80 TANGOF(A)`	Deaktivování 1. spojení
`N90 TANGDEL(A)`	Vymazání 1. definice
`...`	
`TANG(A, X, Z)`	1. definice tangenciálního sledování
`TANGON(A)`	Aktivování nového spojení
`...`	
`N200 M30`	



Příklad programování

S přepínáním geometrické osy a příkazem TANGDEL

Nebude aktivován žádný alarm.

`N10 GEOAX(2,Y1)`	Y1 je geometrická osa 2.
`N20 TANG(A, X, Y)`	
`N30 TANGON(A, 90)`	
`N40 G2 F8000 X0 Y0 I0 J50`	
`N50 TANGOF(A)`	Deaktivování sledování s Y1
`N60 TANGDEL(A)`	Vymazání 1. definice
`N70 GEOAX(2, Y2)`	Y2 je novou geometrickou osou 2
`N80 TANG(A, X, Y)`	2. definice tangenciálního sledování
`N90 TANGON(A, 90)`	Aktivování sledování podle 2. definice
`...`	

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Tangenciální sledování s automatickou optimalizací pomocí příkazů `Dist` a `Winkeltoleranz`.

```
N80 G0 C0
N100 F=50000
N110 G1 X1000 Y500
N120 TRAORI                          ; zaoblení s axiální tolerancí
N130 G642
N171 TRANS X-1200 Y-550              ; autom. optimalizace rychlosti po dráze
N180 TANG(C,X,Y, 1,,"P")             ; dráha zaoblení 5 mm
N190 TANGON(C, 0, 5.0, 2.0)          ; úhlová tolerance 2 stupně
N210 G1 X1310 Y500
N215 G1 X1420 Y500
N220 G3 X1500 Y580 I=AC(1420)_
                   J=AC(580)
N230 G1 X1500 Y760
N240 G3 X1360 Y900 I=AC(1360)_
                   J=AC(760)
N250 G1 X1000 Y900
N280 TANGOF(C)
N290 TRAFOOF
N300 M02
```

Další upozornění

Vliv transformací

Pozice sledující kruhové osy může být vstupní hodnotou pro transformaci.

Explicitní polohování vlečné osy

Jestliže se u některé z řídících os sledující vlečné osy uskutečňuje explicitní polohování, je údaj polohy přičítán k naprogramovanému úhlu offsetu.
Jsou přípustné všechny druhy zadání cesty: pohyby dráhových i polohovacích os.

Stav spojení

Ve výrobním programu pro NC je možné zjistit stav spojení pomocí následující systémové proměnné:

`$AA_COUP_ACT[Osa]`

0 Není aktivní žádné spojení
1, 2, 3 Tangenciální sledování je aktivní

© Siemens AG 2004, Všechna práva vyhrazena

9.2 Vlečení, TRAILON, TRAILOF

Programování

```
TRAILON(FOsa, LOsa, Spojení1)
TRAILOF(FOsa, LOsa, Osa)
```

Vysvětlení příkazů a parametrů

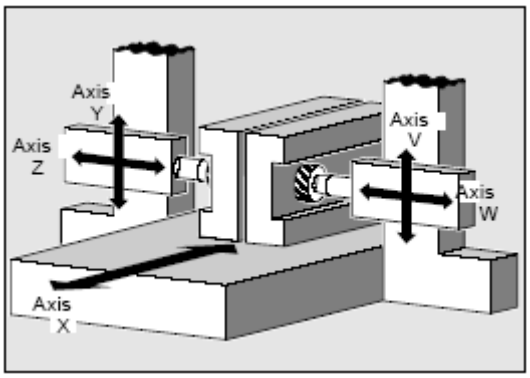
TRAILON	Aktivování a definice vlečné vazby; modální platnost
TRAILOF	Deaktivování vlečné vazby
FOsa	Identifikátor vlečené osy
LOsa	Identifikátor řídící osy
Spojení1	Faktor spojení = dráha vlečené osy/dráha řídící osy předdefinované nastavení = 1

Funkce

Při pohybech definované osy se pohybují jí přiřazené vlečné osy po drahách posuvu odvozených od řídící osy, přičemž se zohledňuje faktor spojení.
Řídící osa a vlečné osy dohromady tvoří vlečný svazek.

Oblasti použití

- Ovládání osy simulovanou osou. Řídící osou je simulovaná osa a vlečnou osou je reálná osa. Takto lze pohybovat reálnou osou s ohledem na faktor spojení.
- Dvoustranné obrábění se dvěma vlečnými spojeními:
  1. Řídící osa Y, vlečná osa V
  2. Řídící osa Z, vlečná osa W



© Siemens AG 2004, Všechna práva vyhrazena

Postup

Definice vlečného spojení, TRAILON
Definice a aktivování vlečného svazku se provádí pomocí příkazu jazyka TRAILON, který má modální platnost.

```
TAILON(V,Y)
```

V = vlečná osa, Y = řídící osa

Počet vlečných svazů, které mohou být aktivovány současně, je omezen pouze kombinačními možnostmi os, které na stroji existují.

Vlečení se uskutečňuje vždy v základním souřadném systému (BCS).

Typy os ve svazku
Vlečný svazek se může skládat z libovolných kombinací lineárních a kruhových os. Jako řídící osa přitom může být definována i osa simulovaná.

Vlečné osy
Vlečná osa může mít současně přiřazeny dvě řídící osy. Přiřazení se uskutečňuje ve dvou různých vlečných svazcích.

Pro vlečnou osu mohou být naprogramovány všechny příkazy pohybu, které máte k dispozici (G0, G1, G2, G3, ...). Kromě nezávisle definovaných drah uskutečňuje vlečná osa pohyby, které jsou odvozeny od jejích řídících os po přepočítání faktorem spojení.

Vlečená osa může být řídící osou pro další vlečné osy. Tímto způsobem mohou být vytvářeny různé vlečné svazky.

© Siemens AG 2004, Všechna práva vyhrazena

Faktor spojení

Faktor spojení udává požadovaný poměr drah vlečné a řídící osy.

$$\text{Faktor spojení} = \frac{\text{dráha vlečné osy}}{\text{dráha řídící osy}}$$

Pokud faktor spojení není při programování zadán, automaticky platí, že jeho hodnota je 1.

Faktor se zadává jako desetinné číslo s desetinnou tečkou (typ REAL). Zadání záporné hodnoty způsobí, že řídící a vlečná osa se budou pohybovat v opačných směrech.

Deaktivování vlečného svazku

Vypnutí spojení s řídící osou se bude provádět pomocí příkazu:

```
TRAILOF(V,Y)
```

V = vlečná osa, Y = řídící osa

Příkaz TRAILOF se 2 parametry deaktivuje pouze spojení s jednou řídící osou.

Pokud má vlečná osa 2 řídící osy, např. V = vlečná osa a X, Y = řídící osy, je zapotřebí za účelem deaktivování spojení vyvolat příkaz TRAILOF se třemi parametry:

```
TRAILOF(V,X,Y)
```

© Siemens AG 2004, Všechna práva vyhrazena

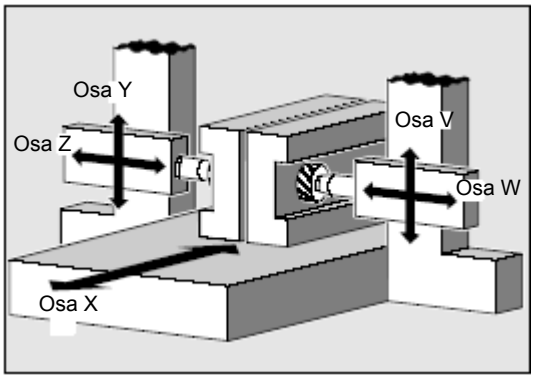
Další upozornění

Zrychlení a rychlost

Mezní hodnoty zrychlení a rychlosti os náležejících ke spojení jsou dány „nejslabší osou“ vlečného svazku.

Příklad programování

Obrobek má být opracován ze dvou stran pomocí uspořádání os, které je uvedeno na obrázku. Za tím účelem vytvořte dva vlečné svazky.



...	
N100 TRAILON(V,Y)	Aktivování 1. vlečného svazku
N110 TRAILON(W,Z,-1)	Aktivování 2. vlečného svazku Faktor spojené je záporný: Vlečná osa se pohybuje vždy opačným směrem než osa řídící.
N120 G0 Z10	Přísuv as Z a W opačnými směry pohybu
N130 G0 Y20	Přísuv os Z a V ve stejném směru osy
...	
N200 G1 Y22 V25 F200	Superpozice nezávislého a závislého pohybu vlečné osy „V“.
...	
TRAILOF(V,Y)	Deaktivování 1. vlečného svazku
TRAILOF(W,Z)	Deaktivování 2. vlečného svazku

© Siemens AG 2004, Všechna práva vyhrazena

9.3 Tabulky křivky, CTABDEF, CTABEND, CTABDEL, CTAB, CTABINV/CTABFNO

9.3.1 Příkazy s tabulkami křivky, CTABIB, CTABLOCK, CTABUNLOCK

Programování

Příkazy s tabulkami křivky s modální platností

A) Hlavní funkce

Definice tabulek křivky se provádí ve výrobním programu. Vysvětlení parametrů naleznete na konci seznamu příkazů.

`CTABDEF (FOsa,LOsa,n,applim,memTyp)`	Začátek definice tabulky křivky.
`CTABEND ()`	Konec tabulky křivky.
`CTABDEL ()`	Vymazání všech tabulek křivky **bez ohledu na typ paměti** (od SW 6.3).
B) Všeobecné tvary (od SW 6.3)	V typu paměti SRAM nebo DRAM:
`CTABDEL (n,m,memTyp)`	Vymazání tabulek křivky z jejich oblastí, které jsou uloženy v paměti memTyp.
`CTABNONEM (memTyp)  (od SW 6.4)`	Počet **definovaných** tabulek křivky.
`CTABNFO (memTyp)  (od SW 6.4)`	Počet ještě **možných** tabulek křivky.
`CTABID (n, memTyp)  (od SW 6.4)`	Zjištění čísla tabulky, které je v zadaném typu paměti uloženo jako n-tá tabulka křivky
`CTABLOCK (n, m, memTyp)`	**Aktivování ochrany** proti vymazání a přepisu.
`CTABUNLOCK (n, m, memTyp)`	**Deaktivování ochrany** proti vymazání a přepisu. Příkaz CTABUNLOCK uvolní blokování tabulky, jejíž ochrana byla aktivována příkazem CTABLOCK. Tabulky, které jsou součástí nějaké aktivní vazby, zůstávají i nadále zablokované a není možné je vymazat. Blokování příkazem CTABLOCK je však zrušeno, jakmile je deaktivována příslušná vazba. Tabulka potom může být vymazána. Opětovné použití příkazu CTABUNLOCK není nutné.

© Siemens AG 2004, Všechna práva vyhrazena

C) Použití dalších forem (od SW 6.3)	**Volitelné údaje pro výběr:**
`CTABDEL(n)`	Vymazání **jedné** tabulky křivky.
`CTABDEL(n, m)`	Vymazání **jedné** oblasti tabulky křivky.
`CTABDEL(, , memTyp)`	Vymazání **všech** tabulek křivky v uvedené paměti.
`CTABLOCK(n)`	Aktivování ochrany proti vymazání a přepisu: Tabulka křivky s číslem n.
`CTABLOCK (n, m)`	Aktivování ochrany pro tabulky křivky v číselné oblasti n až m.
`CTABLOCK()`	Všechny dosud existující tabulky křivky
`CTABLOCK(, , memTyp)`	Všechny tabulky křivky v uvedeném typu paměti.
`CTABUNLOCK(n)`	**Deaktivování** ochrany proti vymazání a přepisu: Tabulka křivky s číslem n.
`CTABUNLOCK (n, m)`	Deaktivování ochrany pro tabulky křivky v číselné oblasti n až m.
`CTABUNLOCK()`	Všechny dosud existující tabulky křivky
`CTABUNLOCK(, , memTyp)`	Všechny tabulky křivky **v** uvedeném typu paměti.
D) Použití dalších forem (od SW 6.4)	**Pro diagnostiku spojených os**
`CTABID (n, memTyp)`	Funkce vrací číslo n-/p-té tabulky křivky z **typu paměti** uvedeného v memTyp.
`CTABID (p, memTyp)`	
`CTABID (n)`	Funkce vrací číslo n-té tabulky křivky s typem paměti **definovaným** v MD 20905: CTAB_DEFAULT_MEMORY_TYPE.
`CTABISLOCK(n)`	Umožňuje zjistit stav ochrany **tabulky křivky s číslem n**.
`CTABEXIST (n)`	**Kontrola** tabulky křivky s číslem n.
`CTABMEMTYP (n)`	Funkce vrací **paměť**, ve které je uložena tabulka křivky s číslem n.
`CTABPERIOD (n)`	Funkce vrací **periodicitu tabulky**.
`CTABSEG (memTyp)`	Počet již **použitých** křivkových prvků v paměti daného typu.
`CTABSEGID (n)`	Počet křivkových prvků používaných tabulkou křivky s číslem n.
`CTABFSEG (memTyp)`	Počet **možných** křivkových prvků.
`CTABMSEG (memTyp)`	**Maximální** možný počet křivkových prvků.
`CTABPOLID (n)`	Počet křivkových polynomů použitých tabulkou křivky **s číslem n**.
`CTABFPOL (memTyp)`	Počet **ještě možných** křivkových polynomů v příslušném typu paměti.
`CTABMPOL (memTyp)`	Maximální možný **počet křivkových polynomů** v příslušném typu paměti.

© Siemens AG 2004, Všechna práva vyhrazena

Zjištěné vlečné nebo řídící pozice z tabulky křivky pomocí CTAB, CTABINV **(od SW 5.1)**

`R10=CTAB (LW,n,grad,FOsa,LOsa)`	Vlečná hodnota řídící hodnotě
`R10=CTABINV (FW,aproxLW,n,grad,FOsa,LOsa)`	Řídící hodnota vlečné hodnotě

Segmenty tabulky křivky zadáním řídící hodnoty určené pomocí CTABSSV, CTABSEV **(od SW 6.3)**

`R10=CTABSSV (LW,n,grad,FOsa,LOsa)`	Počáteční hodnota vlečné osy v segmentu náležejícím řídící hodnotě
`R10=CTABSEV (FW,n,grad,FOsa,LOsa)`	Koncová hodnota vlečné osy v segmentu náležejícím řídící hodnotě

Hodnoty vlečné a řídící osy, které se nacházejí na počátku a na konci tabulky křivky CTABTSV, CTABTEV, CTABTSP, CTABTEP **(od SW 6.4)**

`R10=CTABTSV (n,grad,FOsa)`	Vlečná hodnota na začátku tabulky křivky
`R10=CTABTEV (n,grad,FOsa)`	Vlečná hodnota na konci tabulky křivky
`R10=CTABTSP (n,grad,FOsa)`	Řídící hodnota na začátku tabulky křivky
`R10=CTABTEP (n,grad,FOsa)`	Řídící hodnota na konci tabulky křivky

Rozsah hodnot tabulky křivky vlečné hodnoty, CTABTMIN, CTABTMAX

`R10=CTABTMIN(n,FOsa)`	Minimální vlečná hodnota tabulky křivky v celém intervalu
`R10=CTABTMAX(n,FOsa)`	Maximální vlečná hodnota tabulky křivky v celém intervalu
`R10=CTABTMIN(n,a,b,FOsa,LOsa)`	Minimální vlečná hodnota tabulky křivky v intervalu a...b řídící hodnoty
`R10=CTABTMAX(n,a,b,FOsa,LOsa)`	Maximální vlečná hodnota tabulky křivky v intervalu a...b řídící hodnoty

Přiřazení R-parametru v rámci definice tabulky mají zpětnou platnost.
Příklad:

```
...
R10=5 R11=20
...
CTABDEF
G1 X=10 Y=20 F1000
R10=R11+5              ; R10=25
X=R10
CTABEND
...                    ; R10=5
```

Vysvětlení příkazu

FOsa	Vlečná osa Osa, která je programována pomocí tabulky křivky.
LOsa	Řídící osa Osa, kterou je programována řídící hodnota.
R10	Název parametru Název parametru, do něhož se má ukládat počáteční, příp. koncová hodnota.
LW	Řídící hodnota Údaj polohy řídící osy, který se má přepočítat pro vlečnou osu.
FW	Vlečná hodnota Údaj polohy vlečné osy, který se má přepočítat pro řídící osu.
n, m	Číslo tabulky křivky; n < m např. u CTABDEL(n, m). Číslo tabulky křivky je jednoznačné a nezávislé na typu paměti. V paměti SRAM a DRAM nemohou existovat tabulky se stejnými čísly.
a, b	Interval řídících hodnot (nepovinný údaj u příkazů CTABMIN a CTABMAX). Pro zadání intervalu jsou potřebné obě hodnoty. Pro hodnoty, které leží mimo definiční rozsah řídící hodnoty tabulky křivky se použije jedna z obou mezních hodnot řídící hodnoty.
grad	Název parametru směrnice.
P	Paměťové místo (v paměťové oblasti memTyp).
applim	Označení pro periodicitu tabulky: 0 tabulka není periodická 1 tabulka je periodická, pokud jde o řídící hodnotu 2 tabulka je periodická, pokud jde o řídící a vlečnou hodnotu
aproxLW	Přibližný údaj řídící hodnoty, jestliže pro danou vlečnou hodnotu není možné určit žádnou jednoznačnou řídící hodnotu.
FOsa, LOsa	Libovolné udání řídící a/nebo vlečné osy.
memTyp	Volitelné udání typu paměti v NC: “DRAM“ / “SRAM“ Pokud pro tento parametr není naprogramována žádná hodnota, použije se standardní typ paměti definovaný strojním parametrem MD 20905: CTAB_DEFAULT_MEMORY_TYPE.

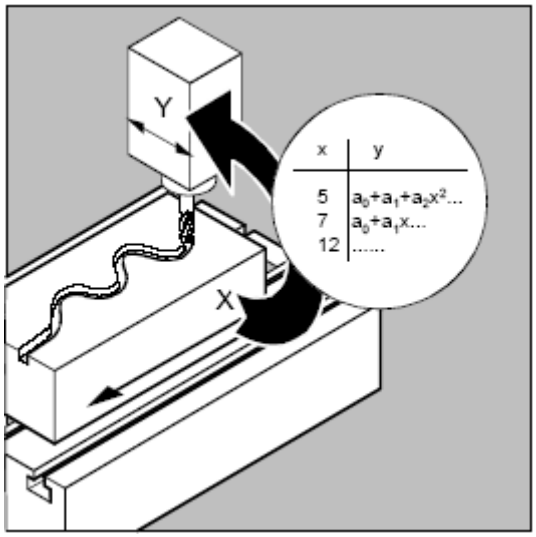
Další informace k tématu řídící hodnota a vlečná hodnota naleznete v odstavci „Spojení os řídící hodnotou“ v této kapitole.

Pro diagnostiku a optimalizaci využití zdrojů existují i další funkce. Další informace viz: /FB/, M3, Spojení os a ESR

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Pomocí tabulek křivky můžete naprogramovat vztahy mezi polohami a rychlostmi mezi dvěma osami.



Příklad náhrady mechanické vačky: Tabulka křivky tvoří podklad pro spojení os pomocí řídící hodnoty tím, že popisuje funkční závislost mezi řídící a vlečnou hodnotou.
Na základě odpovídajícího programu řídící systém vypočítává z zájemně si přiřazených poloh řídící a vlečné osy polynom, který odpovídá řídící vačce.

Další upozornění

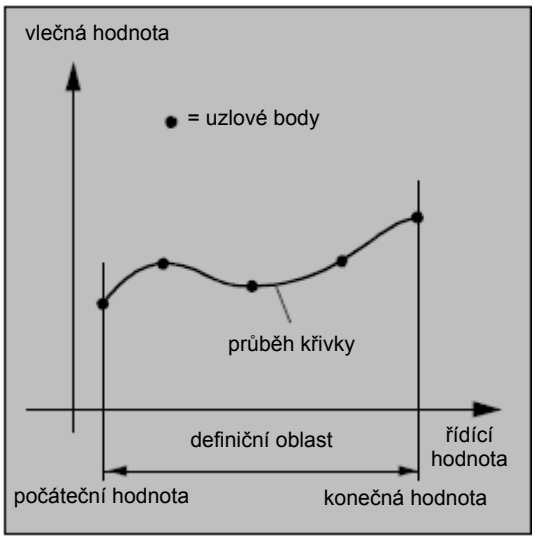
Aby bylo možné vytvořit tabulku křivky, je nutno nastavením strojních parametrů vyhradit určitou část paměťového prostoru.

Definice tabulky křivky: CTABDEF, CTABEND

Tabulka křivky reprezentuje výrobní program nebo část výrobního programu, který začíná příkazem CTABDEF a končí příkazem CTABEND.

V rámci tohoto úseku výrobního programu jsou jednoznačné pozice vlečné osy přiřazeny pomocí příkazů pohybu jednotlivým polohám řídící osy a používají se jako uzlové body pro výpočet křivky ve formě polynomu maximálně 3. stupně.

Od SW 6 mohou uzlové body sloužit pro výpočet křivky ve formě polynomu maximálně 5. stupně.



© Siemens AG 2004, Všechna práva vyhrazena

Počáteční a koncová hodnota tabulky křivky
Jako počáteční hodnota definiční oblasti tabulky křivky platí první údaj souvisejících poloh os (první příkaz pohybu) v rámci definice tabulky křivky. Koncová hodnota definiční oblasti tabulky křivky bude analogicky určena posledním příkazem pohybu.
V rámci definice tabulky křivky je k dispozici celé spektrum příkazů NC jazyka.

Všechny příkazy s modální platností, které se vyskytnou v rámci definice tabulky křivky, se stávají s ukončením definice tabulky neplatnými. Výrobní program, v němž je definice tabulky obsažena, se proto před a po definici tabulky nachází ve stejném stavu.

Další upozornění

Nepřípustné jsou:

- Zastavení předběžného zpracování
- Skokové změny pohybu řídící osy (např. při změně transformace).
- Samotné příkazy pohybu pro vlečnou osu.
- Převrácení směru pohybu řídící osy, tzn. poloha řídící osy musí být vždy jednoznačná.
- Příkazy CTABDEF a CTABEND na odlišných programových úrovních.

Aktivování A-splinů, B-splinů, C-splinů
Pokud je v rámci tabulky křivky CTABDEF( ) ... CTABEND aktivován příkaz ASPLINE, BSPLINE nebo CSPLINE, měl by být před tímto aktivováním splinu naprogramován minimálně počátek. Okamžitému aktivování po příkazu CTABDEF byste se měli vyhnout, jinak by spline závisel na aktuální poloze osy před definicí tabulky křivky.
Příklad:

```
...
CTABDEF(X, Y, 1, 0)
X0 Y0
ASPLINE
X=5, Y=10
X10, Y40
...
CTABEND
```


© Siemens AG 2004, Všechna práva vyhrazena

Od SW 6.3
V závislosti na strojním parametru MD 20990: CTAB_ENABLE_NO_LEADMOTION mohou být tolerovány skokové změny vlečné osy v případě chybějícího pohybu řídící osy. Zbylá omezení uvedená v upozornění platí i nadále.
Při založení a vymazání tabulky lze využít údajů o typu paměti.

Opakované použití tabulky křivky
Funkční závislost mezi vlečnou a řídící osou vypočítaná podle tabulky křivky zůstává zachována pod zvoleným číslem tabulky i po skončení výrobního programu, a pokud se tabulka uložena ve statické paměti SRAM, i po vypnutí napájení. Tabulka, která byla založena v dynamické paměti (DRAM), bude při vypnutí a zapnutí vymazána, takže v případě potřeby bude nutné ji znovu vytvořit. Jedenkrát vytvořená tabulka křivky může být využita libovolnou kombinací řídící a vlečné osy a je nezávislá na tom, které osy byly použity pro vytvoření tabulky křivky.

Vymazání tabulky křivky, CTABDEL
Příkazem CTABDEL je možné tabulku křivky vymazat. Tabulky křivky, které jsou aktivní ve spojení os, nemohou být vymazány. Pokud je alespoň jedna tabulka křivky z vícenásobného příkazu mazání CTABDEL( ) nebo CTABDEL (n, m) aktivní v nějakém spojení, nebude vymazána **žádná** z uvedených os. **Od SW 6.3** mohou být mazány tabulky z určitého typu paměti tak, že tento typ paměti zadáte.

Načítání tabulek křivky pomocí příkazu pro externí zpracování
Při externím zpracovávání tabulky křivky musí být velikost vyrovnávací paměti pro načítání zvolena pomocí MD 18360: MM:EXT_PROG_BUFFER_SIZE tak, aby bylo možné do této paměti najednou uložit celou definici tabulky křivky. Jinak se zpracovávání výrobního programu přeruší s alarmem 15050.

© Siemens AG 2004, Všechna práva vyhrazena

Přepsání tabulky křivky
Tabulka křivky se přepíše, jestliže je při nové definici tabulky použito její číslo.
Výjimka: Tabulka křivky je aktivní v nějakém spojení os nebo je zablokována pomocí příkazu CTABLOCK( ).

Další upozornění

- Při přepisu tabulky křivky se nevypisuje žádné upozornění!
- Pomocí systémové proměnné $P_CTABDEF je kdykoli možné z výrobního programu zjistit, zda je definice tabulky aktivní.
- Úsek výrobního programu je po vyjmutí z příkazů pro definici tabulky křivky opět použitelný jako reálný výrobní program.



Příklad programování
Použití příkazů CTABDEF a CTABEND

```
Úsek programu má být beze změny
použit pro definici tabulky křivky.
Příkaz STOPRE pro zastavení
předběžného zpracování, který se v
něm vyskytuje, může zůstat a bude
okamžitě znovu aktivní, jakmile úsek
programu nebude použit pro definici
tabulky, tedy když budou příkazy
CTABDEF a CTABEND odstraněny:
CTABDEF(Y,X,1,1)
...
...
IF NOT ($P_CTABDEF)
STOPRE
ENDIF
...
...
CTABEND
```

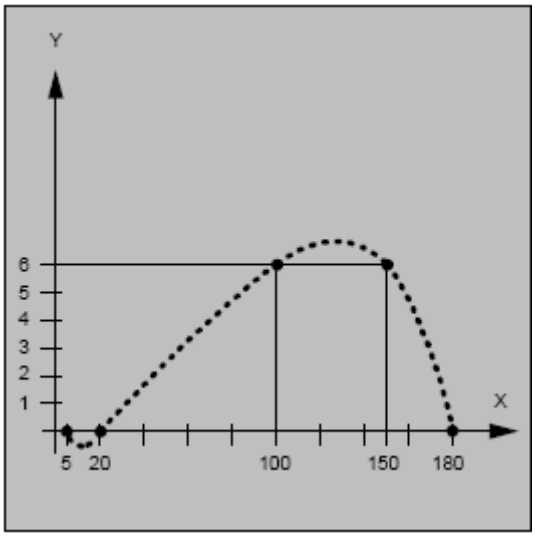


Tabulky křivky a různé provozní stavy
Když probíhá vyhledávání bloku, výpočet tabulky křivky není možný. Pokud cílový blok leží uvnitř definice tabulky křivky, při dosažení příkazu CTABEND se aktivuje alarm.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování 1

Definice tabulky křivky



`N100 CTABDEF(Y,X,3,0)`	Začátek definice neperiodické tabulky křivky s číslem 3
`N110 X0 Y0`	1. Příkaz pohybu, definuje počáteční polohu a první uzlový bod: řídící hodnota: 0, vlečná hodnota : 0
`N120 X20 Y0`	2. Uzlový bod: řídící hodnota 0...20, vlečná hodnota: 0
`N130 X100 Y6`	3. Uzlový bod: řídící hodnota 20...100, vlečná hodnota: 0...6
`N140 X150 Y6`	4. Uzlový bod: řídící hodnota 100...150, vlečná hodnota 6...6
`N150 X180 Y0`	5. Uzlový bod; řídící hodnota 150...180, vlečná hodnota 6...0
`N200 CTABEND`	Konec definice; tabulka křivky bude ve svém interním zobrazení vytvořena jako polynom maximálně 3. stupně; výpočet průběhu křivky se zadanými uzlovými body závisí na modálně zvoleném druhu interpolace (kruhová, lineární nebo splinová interpolace); bude znovu obnoven stav výrobního programu před začátkem definice.

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování 2

Definice periodické tabulky křivky s číslem 2, rozsah řídící hodnoty 0 až 360, pohyb vlečné osy od 0 do 45 a pak zpět na 0.

```
N10 DEF REAL DEPPOS;
N20 DEF REAL GRADIENT;
N30 CTABDEF(Y,X,2,1);                    začátek definice
N40 G1 X=0 Y=0
N50 POLY
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,
-90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,
90)
N100 PO[X]=(360.0)
N110 CTABEND;                            konec definice
```

Test křivky spojením os Y a X:

```
N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)
```

Načtení funkce tabulky pro řídící hodnotu 75.0:

```
N170 DEPPOS=CTAB(75.0,2,GRADIENT)
```

Polohování řídící a vlečné osy:

```
N180 G0 X75 Y=DEPPOS
```

Po aktivování spojení není zapotřebí žádná synchronizace vlečné osy:

```
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30
```

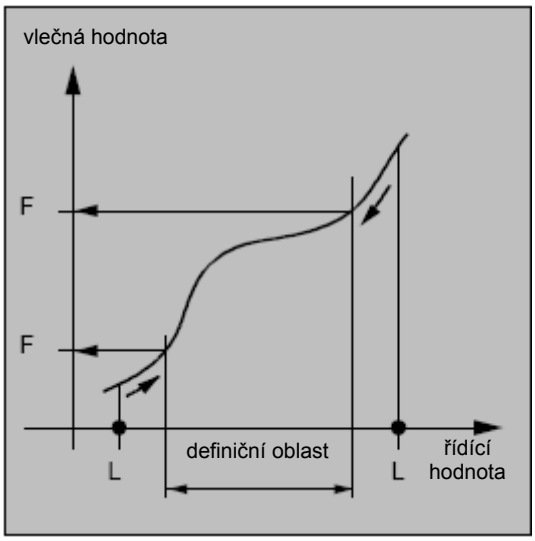

© Siemens AG 2004, Všechna práva vyhrazena

9.3.2 Chování na okrajích tabulky křivky, CTABTSV/CTABTSP, CTABTMIN

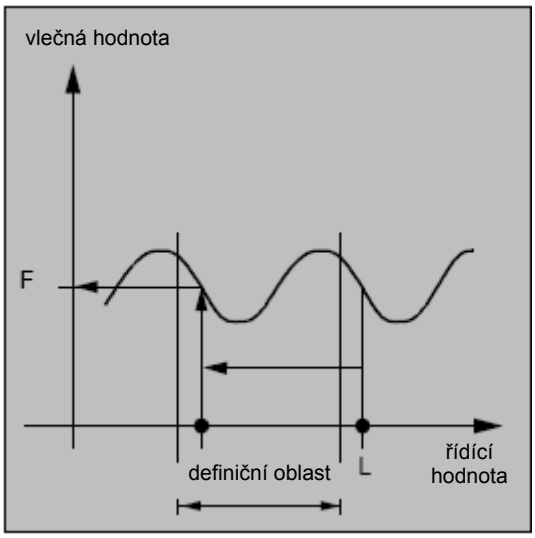
Okrajová počáteční a koncová hodnota: **CTABTSV, CTABTEV, CTABTSP, CTABTEP (od SW 6.4)**

Min a Max rozsahu hodnot: **CTABMIN, CTABMAX (od SW 6.4)**

Neperiodické tabulky křivek
Pokud řídící hodnota leží mimo definiční oblast, bude vlečné hodnotě přiřazena horní, resp. dolní mezní hodnota.



Periodické tabulky křivek
Pokud řídící hodnota leží mimo definiční oblast, bude vyhodnocena řídící hodnota modulo definiční oblast a vlečné hodnotě bude přiřazena odpovídající hodnota.



© Siemens AG 2004, Všechna práva vyhrazena

Načítání okrajových hodnot tabulky křivky
CTABTSV, CTABTEV, CTABTSP, CTABTEP
Pomocí příkazu CTABTSV lze načíst hodnotu **vlečné osy** na **začátku** tabulky křivky.
Pomocí příkazu CTABTEV lze načíst hodnotu **vlečné osy** na **konci** tabulky křivky.

Pomocí příkazu CTABTSP lze načíst hodnotu **řídící osy** na **začátku** tabulky křivky.
Pomocí příkazu CTABTEP lze načíst hodnotu **řídící osy** na **konci** tabulky křivky.

Počáteční a koncové hodnoty tabulky křivky jsou nezávislé na tom, zda je tabulka definována se stoupající nebo klesající řídící hodnotou. Vždy bude určena počáteční hodnota pro spodní hranici intervalu a koncová hodnota pro horní hranici intervalu.

CTABMIN, CTABMAX

Pomocí příkazů CTABMIN a CTABMAX můžete stanovit **minimální** nebo **maximální** hodnotu tabulky křivky v celé oblasti nebo v definovaném intervalu. Pro odpovídající interval řídících hodnot se zadávají obě hranice.



Příkazy CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABMIN a CTABMAX mohou být používány:

- výrobním programem
- přímo ze synchronní akce

Interní doba provedení funkcí

- CTABINV( ) je **závislá** a
- CTABTSV, CTABTEV, CTABTSP, CTABTEP, (CTABMIN a CTABMAX jen tehdy, pokud nebyl zadán interval řídících hodnot)

jsou **nezávislé**
na počtu segmentů tabulky.

© Siemens AG 2004, Všechna práva vyhrazena

Načítání v synchronních akcích

Při použití příkazů CTABINV( ) nebo CTABMIN( ) a CTABMAX( ) v synchronních akcích musí uživatel dávat pozor, aby v okamžiku provádění byl buď

- k dispozici dostatečný výkon NC systému nebo
- před voláním je potřeba zjistit počet segmentů tabulky křivky, aby v případě nutnosti bylo možné tabulku rozdělit.

Další souvislosti týkající se programování synchronních akcí jsou popsány v kapitole 10.



Příklad programování

Použití příkazů CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABMIN a CTABMAX

Stanovení minimální a maximální hodnoty v tabulce křivky.

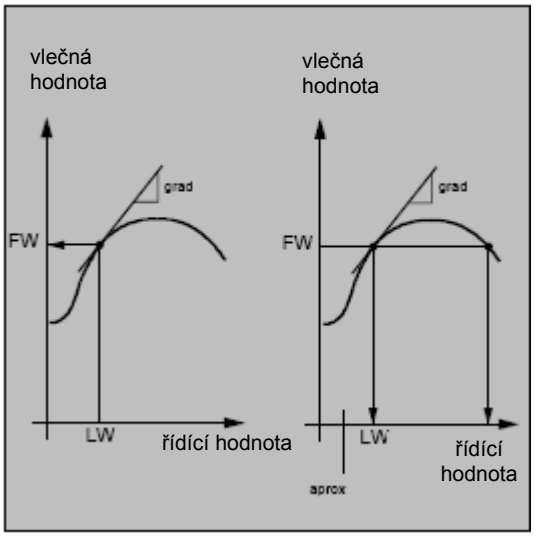
```
N10 DEF REAL STARTVAL
N20 DEF REAL ENDVAL
N30 DEF REAL STARTPARA
N40 DEF REAL ENDPARA
N50 DEF REAL MINVAL
N60 DEF REAL MAXVAL
N70 DEF REAL GRADIENT
...
```

```
N100 CTABDEF(Y,X,1,0)          ; Začátek definice tabulky
N110 X0 Y10                    ; Počáteční hodnota 1. segmentu tabulky
N120 X30 Y40                   ; Konc.hodnota 1. segmentu tab. = poč. hodn.
N130 X60 Y5                    ; 2. segment tabulky
N140 X70 Y30
N150 X80 Y20
N160 CTABEND                   ; Konec definice tabulky
...
N200 STARTPOS = CTABTSV(1, GRADIENT)  ; Načtení počáteční pozice STARTPOS = 10,
N210 ENDPOS = CTABTEV(1, GRADIENT)    ; koncové pozice ENDPOS = 20 tabulky,
N220 SRARTPARA = CTABTSP(1,GRADIENT)  ; jakož i STARTPARA = 10, ENDPARA = 80
N230 ENDPARA = CTABTEP(1, GRADIENT)   ; z rozsahu hodnot vlečné osy.
...
N240 MINVAL = CTABTMIN(1)      ; Minimální hodnota při Y =5 a
N250 MAXVAL = CTABTMAX(1)      ; maximální hodnota při Y = 40
```

© Siemens AG 2004, Všechna práva vyhrazena

9.3.3 Přístup na pozice a segmenty v tabulce křivky CTAB, CTABINV

Načítání pozic tabulky, CTAB, CTABINV



Pomocí příkazu CTAB je možné z výrobního programu nebo ze synchronní akce (kapitola 10) přímo načíst vlečnou hodnotu k řídící hodnotě.

Příkaz CTABINV slouží pro načtení řídící hodnoty k určité vlečné hodnotě. Toto přiřazení nemusí být vždy jednoznačné. Příkaz CTABINV proto potřebuje přibližnou hodnotu (aproxLW) pro očekávanou řídící hodnotu. Příkaz CTABINV potom dodá řídící hodnotu, která leží nejblíže k hodnotě očekávané. Přibližnou hodnotou může být například řídící hodnota z předešlého interpolačního taktu.

Obě funkce navíc poskytují hodnotu parametru směrnice tabulky funkce na odpovídající pozici (grad), s jejíž pomocí lze vypočítat rychlost řídící nebo vlečné osy v daném místě.

Načítání pozic segmentů křivky (CTABSSV, CTABSEV (od SW 6.3)

Pomocí příkazu CTABSSV je možné zjistit **počáteční hodnotu** segmentu křivky, který patří k zadané řídící hodnotě.

Pomocí příkazu CTABSEV je možné zjistit **konečnou hodnotu** segmentu křivky, který patří k zadané řídící hodnotě.

Příkazy CTAB, CTABINV a CTABSSV, CTABSEV mohou být používány:

- výrobním programem
- přímo ze synchronní akce
-

Další souvislosti týkající se programování synchronních akcí jsou popsány v kapitole 10.

© Siemens AG 2004, Všechna práva vyhrazena



Další upozornění

Nepovinné udání řídící nebo vlečné osy v příkazech CTAB/CTABINV/CTABSSV/CTABSEV je důležité., jestliže jsou řídící a vlečná osa konfigurovány v různých jednotkách.

Příkazy CTABSSV a CTABSEV v následujících případech nejsou vhodné pro zjišťování naprogramovaného segmentu:

1. Jsou naprogramovány kruh nebo evolventa
2. Jsou aktivní fasety nebo rádiusy pomocí příkazu CHF, RND
3. Je aktivní zaoblení pomocí příkazu G643
4. Je aktivní kompresor např. pomocí příkazu COMPON, COMPCURV, COMPCAD.



Příklad programování

Použití příkazů CTABSSV a CTABSEV

Stanovení křivkových segmentů náležejících k řídící hodnotě X = 30.

```
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL GRADIENT
...
                                    Začátek definice tabulky
N100 CTABDEF(Y,X,1,0)               Počáteční poloha 1. segmentu tabulky
N110 X0 Y0                          Koncová poloha 1. segmentu tabulky =
N120 X20 Y10                        počáteční poloze 2. segmentu tabulky ...
N130 X40 Y40
N140 X60 Y10
N150 X80 Y0
N160 CTABEND                        Konec definice tabulky
...
N200 STARTPOS = CTABSSV(30.0, 1,    Počáteční pozice Y v segmentu 2 = 10
     GRADIENT)
...                                 Koncová pozice Y v segmentu 2 = 40
N210 ENDPOS = CTABSEV(30.0, 1,      K řídící hodnotě = 30.0 náleží segment 2
     GRADIENT)
```

© Siemens AG 2004, Všechna práva vyhrazena

9.4 Spojení os řídící hodnotou, LEADON, LEADOF

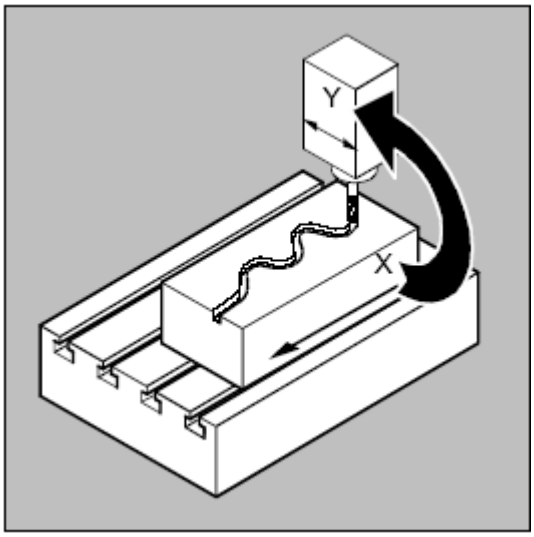
Programování

```
LEADON(FOsa,LOsa,n)
LEADOF(Fosy,LOsa,n)
```

Vysvětlení

LEADON	Aktivování spojení řídící hodnotou
LEADOF	Deaktivování spojení řídící hodnotou
FOsa	Vlečná osa
LOsa	Řídící osa
n	Číslo tabulky křivky

Funkce



V případě spojení os řídící hodnotou se řídící a vlečná osa pohybují synchronně. Přitom je příslušná pozice vlečné osy jednoznačně přiřazena prostřednictvím tabulky křivky, resp. z ní vypočítaného polynomu pozici osy, která může být případně i simulovaná.

Řídící osou je ta osa, která poskytuje vstupní hodnoty pro tabulku křiky. Vlečnou osou je osa, která se nastavuje do poloh vypočítaných pomocí tabulky křivky.

Spojení řídící hodnotou může být aktivováno a deaktivováno jak z výrobního programu, tak i během pohybu ze synchronní akce (kapitola 10).

Spojení řídící hodnotou platí vždy v základním souřadném systému.

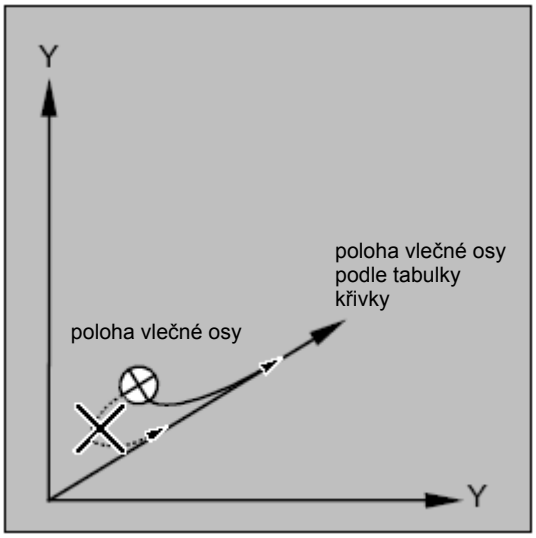
Informace o vytváření tabulek křivky naleznete v odstavci „Tabulky křivky“ v této kapitole. Pokud budete potřebovat informace o spojení řídící hodnotou, viz:
/FB/, M3, Vlečení a spojení řídící hodnotou.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Spojení řídící hodnotou vyžaduje synchronizaci řídící a vlečné osy. Synchronizace je možné dosáhnout jen tehdy, pokud se vlečná osa v okamžiku aktivování spojení řídící hodnotou nachází v rámci tolerančního pásma křivkové charakteristiky vypočítané z tabulky křivky.

Toleranční pásmo pro polohu vlečné osy je definováno strojním parametrem 37200 COUPLE_POS_TOL_COARSE.



Pokud se vlečná osa při aktivování spojení řídící hodnotou ještě nenalézá v odpovídající pozici, automaticky se vytvoří synchronizační proces, který bude trvat, dokud se vypočítaná požadovaná hodnota pozice vlečné osy nepřiblíží její skutečné poloze. Vlečná osa se přitom v průběhu synchronizačního procesu bude pohybovat ve směru, který je definován její požadovanou rychlostí (vypočítané z rychlosti řídící osy a CTAB).

Další upozornění

Pokud je při aktivování spojení řídící hodnotou odstraněna vypočítaná požadovaná poloha vlečné osy ze skutečné polohy této osy, synchronizační proces se neuskuteční.

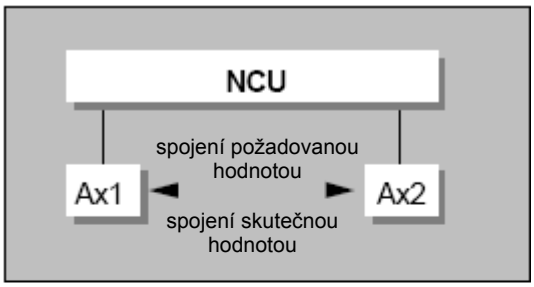
Spojení skutečnou a požadovanou hodnotou
Jako řídící hodnoty mohou být pro zjišťování polohy vlečné osy použity také výstupní hodnoty:

- Skutečná hodnota polohy řídící osy: spojení skutečnou hodnotou
- Požadovaná hodnota polohy řídící osy: spojení požadovanou hodnotou

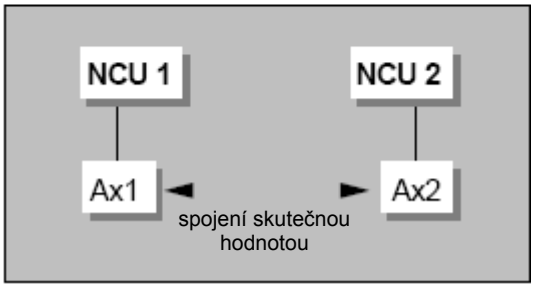
© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Spojení požadovanou hodnotou přináší ve srovnání se spojením skutečnou hodnotou lepší synchronizaci mezi řídící a vlečnou osou a proto je nastaveno standardně.



Spojení požadovanou hodnotou je možné jen tehdy, pokud interpolaci řídící a vlečné osy provádí tatáž NCU. V případě externí řídící osy může být vlečná osa spojena s řídící osou jen pomocí skutečné hodnoty.



Přepínání mezi spojením skutečnou a požadovanou hodnotou

Přepínání je možné uskutečňovat pomocí nastavovaného parametru $SA_LEAD_TYPE.

Přepínání mezi spojením skutečnou a požadovanou hodnotou by se vždy mělo provádět jen tehdy, je-li vlečná osa v klidu. Jedině potom se po přepnutí provede nová synchronizace.

Příklad použití:

V případě velkých třesů stroje se může stát, že odečítání skutečné hodnoty nebude probíhat bezchybně. Při použití spojení řídící hodnotou v přenosu tlaků v pracovních krocích s velkými otřesy se proto může vyskytnout nutnost přepnout ze spojení skutečnou na spojení požadovanou hodnotou.

© Siemens AG 2004, Všechna práva vyhrazena

Simulace řídící hodnoty při spojení požadovanou hodnotou

Pomocí strojního parametru je možné oddělit interpolátor pro řídící osu od servomechanismu. Tak mohou být při spojení požadovanou hodnotou vytvářeny požadované hodnoty, aniž by se řídící osa opravdu pohybovala.

Řídící hodnoty vytvářené přes spojení požadovanou hodnotou je možné načíst kvůli použití např. při synchronních akcích z následujících proměnných:

- `$AA_LEAD_P` řídící hodnota pozice
- `$AA_LEAD_V` řídící hodnota rychlosti

Další upozornění

Budete-li si přát, mohou být řídící hodnoty vytvářeny pomocí jiných Vámi naprogramovaných postupů. Takto vznikající řídící hodnoty jsou ukládány do následujících proměnných, odkud mohou být také načítány:

- `$AA_LEAD_SP` řídící hodnota pozice
- `$AA_LEAD_SV` řídící hodnota rychlosti

Aby bylo možné tyto proměnné používat, musí být dosazeno do nastavovaného parametru `$AA_LEAD_TYPE = 2`.

Stav spojení

Ve výrobním programu NC systému můžete stav spojení zjistit pomocí následující systémové proměnné:

`$AA_COUP_ACT[osa]`

0	Není aktivní žádné spojení
16	Je aktivní spojení řídící hodnotou

Deaktivování spojení řídící hodnotou

Deaktivováním spojení řídící hodnotou se vlečná osa znovu stane normální osou ovládanou příkazy!

Spojení os řídící hodnotou a různé provozní stavy

V závislosti na nastavení strojního parametru se spojení řídící hodnotou při resetu deaktivuje.

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování

U lisovacího zařízení má být nahrazeno obvyklé mechanické spojení mezi řídící osou (osa lisovníku) a osami dopravního systému, který se skládá z os podavače a pomocných os, elektronickým spojovacím systémem. Je zde demonstrováno, jak se u lisovacího zařízení mechanický dopravní systém nahrazuje elektronickým přepravním systémem. Operace vytvoření a zrušení spojení jsou realizovány jako **statické synchronní akce**. Osy podavače a pomocné osy jsou ovládány od řídící osy (řídící hodnota osy lisovníku) jako osy vlečné pomocí definovaných tabulek křivky.

Vlečné osy

X	osa posuvu, resp. podélná osa
YL	osa zavírání, resp. příčná osa
ZL	osa zdvihu
U	válečkový podavač, pomocná osa
V	směrovací hlava, pomocná osa
W	mazání, pomocná osa

Správa stavů

Spínací a spojovací operace jsou spravovány pomocí následující proměnné v reálném čase:

$AC_MARKER[i] = n

i	číslo značky
n	stavová hodnota

Akce

Jako akce se vyskytují následující synchronní akce:

- Aktivování spojení, LEADON(vlečná osa, řídící osa, číslo tabulky křivky)
- Deaktivování spojení, LEADOF(vlečná osa, řídící osa)
- Stanovení skutečné hodnoty, PRESETON (osa, hodnota)
- Stanovení značky $AC_MARKER[i]= hodnota
- Druh spojení: reálná/virtuální řídící hodnota
- Najíždění osami na pozice, POS[osa] = hodnota

Podmínky

Jako podmínky budou vyhodnocovány rychlé digitální vstupy, proměnná v reálném čase $AC_MARKER a porovnání pozic, které jsou spojeny logickým operátorem AND.

Upozornění

V následujícím příkladu byly přechody na další řádek, zařazení a **tučné** písmo použity výhradně proto, aby se zvýšila čitelnost programu. Pro řídící systém musí být všechny zápisy za číslem řádku na tomto řádku.

© Siemens AG 2004, Všechna práva vyhrazena

Komentář

; definuje všechny synchronní akce

; **** vynulování značky

```
N2    $AC_MARKER[0]=0 $AC_MARKER[1]=0
      $AC_MARKER[2]=0 $AC_MARKER[3]=0
      $AC_MARKER[4]=0 $AC_MARKER[5]=0
      $AC_MARKER[6]=0 $AC_MARKER[7]=0
                                        ; ****  E1 0=>1 spojení posuvu aktivováno
N10   IDS=1 EVERY ($A_IN[1]==1) AND
      ($A_IN[16]==1) AND ($AC_MARKER[0]==0)
DO    LEADON(X,LW,1) LEADON(YL,LW,2)
      LEADON(ZL,LW,3) $AC_MARKER[0]=1
                                        ; **** E1 0=>1 spojení válečkového podavače aktivováno
N20   IDS=11 EVERY ($A_IN[1]==1) AND
      ($A_IN[5]==0) AND ($AC_MARKER[5]==0)
DO    LEADON(U,LW,4) PRESETON(U,0)
      $AC_MARKER[5]=1
                                        ; **** E1 0->1 spojení směrovací hlavy aktivováno
N21   IDS=12 EVERY ($A_IN[1]==1) AND
      ($A_IN[5]==0) AND ($AC_MARKER[6]==0)
DO    LEADON(V,LW,4) PRESETON(V,0)
      $AC_MARKER[6]=1
                                        ; **** E1 0-> 1 spojení mazání aktivováno
N22   IDS=13 EVERY ($A_IN[1]==1) AND
      ($A_IN[5]==0) AND ($AC_MARKER[7]==0)
DO    LEADON(W,LW,4) PRESETON(W,0)
      $AC_MARKER[7]=1
                                        ; **** E2 0=>1 spojení deaktivováno
N30   IDS=3 EVERY ($A_IN[2]==1)
DO    LEADOF(X,LW) LEADOF(YL,LW)
      LEADOF(ZL,LW) LEADOF(U,LW)
      LEADOF(V,LW) LEADOF(W,LW) $AC_MARKER[0]=0
      $AC_MARKER[1]=0 $AC_MARKER[3]=0
      $AC_MARKER[4]=0 $AC_MARKER[5]=0
      $AC_MARKER[6]=0 $AC_MARKER[7]=0
....
N110  G04 F01
N120  M30
```

9.5 Průběh posuvu, FNORM, FLIN, FCUB, FPO

Programování

```
F... FNORM
F... FLIN
F... FCUB
F=FPO(...,...,...)
```

Vysvětlení

FNORM	Základní nastavení. Hodnota posuvu se zadává jako funkce dráhového bloku posuvu a pak platí jako modální hodnota.
FLIN	Lineární profil rychlosti po dráze: Hodnota posuvu se lineárně nastavuje na úseku dráhy z aktuální hodnoty na počátku bloku až po konec bloku a pak platí jako modální hodnota. Toto chování může být kombinováno s příkazy G93 a G94.
FCUB	Kubický profil rychlosti po dráze: Blokově naprogramované hodnoty F – vztahující se na konec bloku – jsou napojeny pomocí splinu. Spline začíná a končí tangenciálně napojený na následující, resp. předcházející hodnotu posuvu a je ovlivňován příkazy G93 a G94. Pokud v bloku F-slovo chybí, použije se naposled naprogramovaná hodnota F.
F=FPO...	Polynomický profil rychlosti po dráze: F-slovo popisuje průběh posuvu pomocí polynomu z aktuální hodnoty až po hodnotu na konci bloku. Koncová hodnota má potom modální platnost.

Funkce

Za účelem dosažení pružného zadávání průběhu posuvu bylo jeho programování rozšířeno podle normy DIN 66025 o lineární a kubické charakteristiky. Kubické průběhy mohou být naprogramovány přímo nebo jako interpolované spliny. Díky tomu je možné – v závislosti na zakřivení obráběného obrobku naprogramovat spojitý hladký průběh rychlosti.
Tyto průběhy rychlosti umožňují spojité změny zrychlení bez trhavých změn a tedy výrobu hladkých povrchů obrobků.

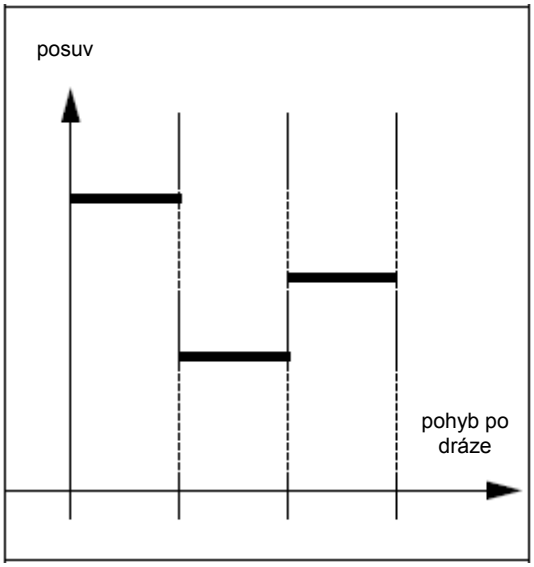
© Siemens AG 2004, Všechna práva vyhrazena

Postup

FNORM

Adresa posuvu F označuje rychlost posuvu po dráze jako konstantní hodnotu podle normy DIN 66025.

Další informace k tomuto tématu naleznete v Příručce programování – Základy.

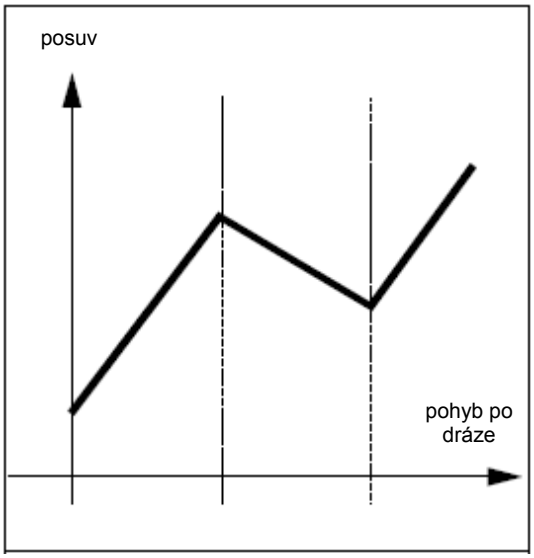


FLIN

Průběh rychlosti se mění lineárně od aktuální hodnoty posuvu k naprogramované hodnotě F až do konce bloku.

Příklad:

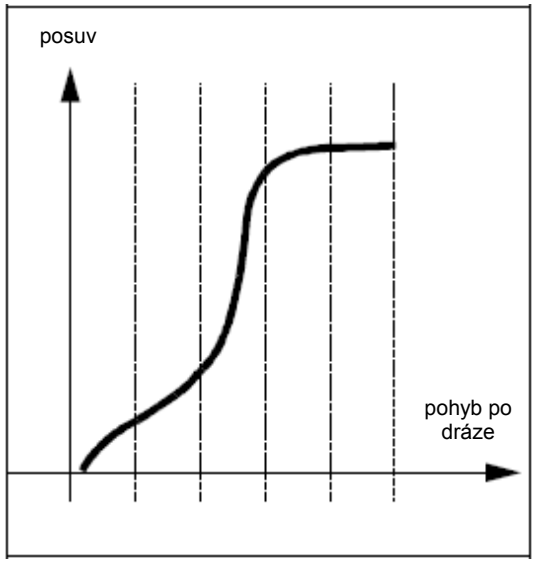
```
N30 F1400 FLIN X50
```



© Siemens AG 2004, Všechna práva vyhrazena

FCUB

Hodnota posuvu je nastavována z aktuální hodnoty po naprogramovanou hodnotu F na konci bloku po kubické charakteristice. Řídící systém v bloku, v němž je uveden příkaz FCUB, spojí všechny naprogramované hodnoty posuvu pomocí splinů. Hodnoty posuvu zde slouží jako uzlové body pro výpočet splinové interpolace.

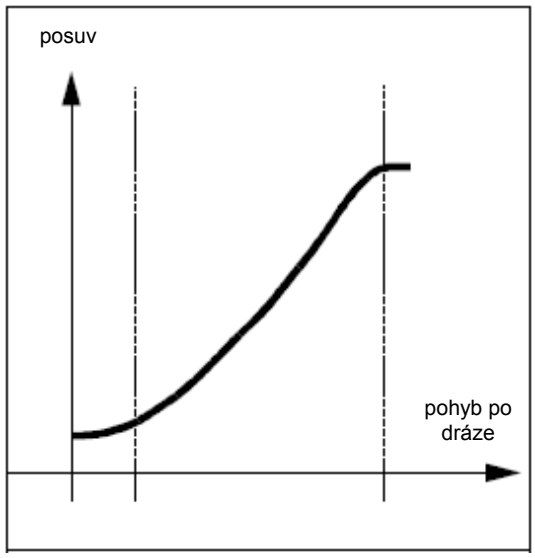


Příklad:

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
...
```

F=FPO(...,...,...)

Průběh posuvu je naprogramován přímo pomocí polynomu. Zadávání koeficientů polynomu probíhá analogicky k polynomické interpolaci.



Příklad:

```
F=FPO(endfeed, quadf, cubf)
```

`endfeeed`, `quadf` a `cubf` jsou dříve definované proměnné.

`endfeed:`	Posuv na konci bloku
`quadf:`	Kvadratický koeficient polynomu
`cubf:`	Kubický koeficient polynomu

Pokud je aktivován příkaz FCUB, napojuje se spline na začátku a na konci bloku tangenciálně na průběh definovaný příkazem FPO.

© Siemens AG 2004, Všechna práva vyhrazena

Okrajové podmínky
Nezávisle na naprogramovaném průběhu posuvu platí funkce pro naprogramování chování při pohybu po dráze.
Programovatelný průběh posuvu platí v zásadě absolutně – nezávisle na G90 nebo G91.



Průběh posuvu FLIN a FCUB **je v platnosti** s příkazy G93 a G94.

Příkazy FLIN a FCUB **nejsou v platnosti** s příkazy G95, G96/G961 a G97/G971.



Další upozornění

Kompresor
Pokud je aktivní kompresor příkazem COMPON, platí při shrnutí více bloků do jednoho splinového segmentu následující:

FNORM:
Pro splinový segment platí F-slovo posledního spojeného bloku.

FLIN:
Pro splinový segment platí F-slovo posledního spojeného bloku.
Naprogramované F-slovo platí do konce segmentu a potom se bude pohyb uskutečňovat lineárně.

FCUB:
Vytvořený spline posuvu se od naprogramovaného koncového bodu bude odchylovat maximálně o hodnotu definovanou ve strojním parametru $MC_COMPESS_VELO_TOL.

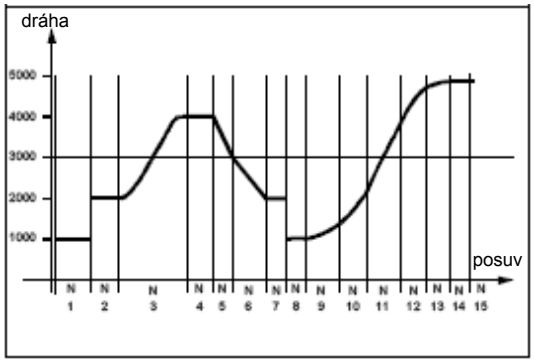
F=FPO(...,...,...)
U těchto bloků se komprimace neprovádí.

Optimalizace posuvu na zakřivených úsecích dráhy
Polynom posuvu F=FPO a spline posuvu FCUB by se měly vždy objíždět s konstantní řeznou rychlostí CFC. Takto bude vznikat profil požadovaného posuvu se spojitým zrychlením.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

V tomto příkladu naleznete programování a grafické zobrazení různých profilů posuvu.



`N1 F1000 FNORM G1 X8 G91 G64`	Konstantní profil posuvu, zadávání inkrementálních rozměrů
`N2 F2000 X7`	Skoková změna požadované rychlosti
`N3 F=FPO(4000, 6000, -4000)`	Profil posuvu pomocí polynomu s posuvem 4000 na konci bloku
`N4 X6`	Polynomický posuv platí jako modální hodnota
`N5 F3000 FLIN X5`	Lineární profil posuvu
`N6 F2000 X8`	Lineární profil posuvu
`N7 X5`	Lineární posuv platí jako modální hodnota
`N8 F1000 FNORM X5`	Konstantní profil posuvu se skokovou změnou zrychlení
`N9 F1400 FCUB X8`	Všechny následující blokově naprogramované hodnoty F budou spojeny spliny
`N10 F2200 X6`	
`N11 F3900 X7`	
`N12 F4600 X7`	
`N13 F4900 X5`	Deaktivování splinového profilu
`N14 FNORM X5`	
`N15 X20`	

© Siemens AG 2004, Všechna práva vyhrazena

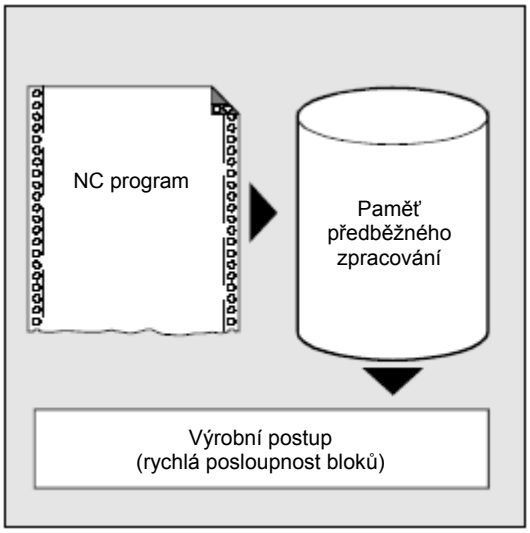
9.6 Zpracování programu s pamětí předběžného zpracování, STARTFIFO, STOPFIFO, STOPRE

Vysvětlení příkazů

STOPFIFO	Pozastavení rychlého zpracovávání úseků programu, naplnění paměti předběžného zpracování, dokud nebude nalezen příkaz STARTFIFO, „paměť předběžného zpracování plná“ nebo „konec programu“.
STARTFIFO	Zahájení rychlého zpracování úseků programu souběžně s plněním paměti předběžného zpracování.
STOPRE	Zastavení předběžného zpracování.

Funkce

V závislosti na své konstrukci disponuje řídící systém určitým objemem tzv. paměti předběžného zpracování, do které se ukládají připravené bloky před svým zpracováním a které jsou pak odtud předávány jako vysokorychlostní posloupnosti bloků.
Díky tomu lze i krátké úseky dráhy objíždět s vysokou rychlostí.
Pokud to volný strojový čas řídícího systému připouští, bude se paměť předběžného zpracování vždy plnit. Příkaz STOPFIFO zpracování pozastavuje, dokud není paměť předběžného zpracování naplněna nebo dokud není rozpoznán příkaz STARTFIFO nebo STOPRE.



Postup

Označení úseku pro zpracování
Úsek pro zpracování, který má být dočasně uložen do paměti předběžného zpracování, je na svém začátku a na konci označen příkazy STOPFIFO, resp. STARTFIFO.

Příklad:

```
N10 STOPFIFO
N20 ...
N100 ...
N110 STARTFIFO
```

Zpracování bloků začne až tehdy, když je paměť předběžného zpracování naplněna nebo po příkazu STARTFIFO.

© Siemens AG 2004, Všechna práva vyhrazena

Omezení

Plnění paměti předběžného zpracování se neuskuteční, příp. bude přerušeno, pokud úsek pro zpracování obsahuje příkazy, které vyžadují zpracování bez ukládání do tohoto typu paměti (najíždění na referenční bod, měřicí funkce, ...).

Zastavení předběžného zpracování

Při naprogramování příkazu **STOPRE** bude následující blok zpracován až tehdy, až budou úplně zpracovány všechny předtím připravené a uložené bloky. Předcházející blok bude pozastaven s přesným najetím (jako při G9).

Příklad:

```
N10 ...
N30 MEAW=1 G1 F1000 X100 Y100 Z50
N40 STOPRE
```

Při přístupu ke stavovým datům stroje ($A...) provede řídící systém interní zastavení předběžného zpracování.

Příklad

```
R10 = $AA_IM[X]                    ; načtení skutečné hodnoty osy X
```



Upozornění

Při aktivované korekci nástroje a při splinové interpolaci byste příkaz STOPRE neměli používat, jinak by došlo k přerušení posloupnosti k sobě náležejících bloků.

© Siemens AG 2004, Všechna práva vyhrazena

9.7 Úseky programu s podmíneným přerušením, DELAYFSTON, DELAYFSTOF

Programování

```
N... DELAYFSTON
N... DELAYFSTOF
```

Příkazy se nacházejí na samostatném řádku výrobního programu. **DEALY F**eed **ST**p **ON/OF**F.

Vysvětlení příkazu

DELAYFSTON	Definice začátku oblasti, v níž bude aktivováno zpoždění měkkého zastavení, dokud nebude dosaženo konce oblasti zpožděného zastavení.
DELAYFSTOF	Definice konce oblasti zpožděného zastavení.
	Oba příkazy jsou přípustné pouze ve výrobních programech, v synchronních akcích jsou nepřípustné.
	Při nastavení strojního parametru 11550: STOP_MODE_MASK bit 0 = 0 (předdefinované nastavení) je oblast zpožděného zastavení definována implicitně, pokud je aktivní příkaz G331/G332 a je naprogramován pohyb po dráze, příp. G4. Viz upozornění níže.

Funkce

V rámci určitého úseku programu nemá dojít k **zastavení** a také **posuv** *se nemá* měnit.
V zásadě by krátké úseky programu, které slouží např. k vyrobení závitu, měly být chráněny před téměř všemi událostmi, které by mohly způsobit zastavení. Případné zastavení se uskuteční, až když je tento úsek programu zpracován až do konce.
V oblasti zpožděného zastavení jsou změny **posuvu a zablokování posuvu** ignorovány. Budou uplatněny až za oblastí zpožděného zastavení.
Události způsobující zastavení lze rozdělit:

- „Měkké“ události způsobující zastavení — Reakce: zpožděná
- „Tvrdé“ události způsobující zastavení — Reakce: okamžitá

© Siemens AG 2004, Všechna práva vyhrazena

Název události	Reakce	Poznámka
RESET	okamžitá	VDI: DB21.DBB7.bit7 a DB11.DBB20.bit7
PROG_END	alarm 16954	NC-Prog.: M30
INTERRUPT	zpožděná	VDI: FC-9 a DB10.DBB1 (ASUP)
DELDISTOGO_SYNC	okamžitá	VDI: Vymazání zbytkové dráhy pro kanál a osu
PROGRESETREPEAT	zpožděná	VDI: DB21.DBB6.bit3
PROGCANCELSUB	zpožděná	VDI: DB21.DBB6.bit4
SINGLEBLOCKSTOP	zpožděná	VDI: Pozastavení po každém bloku. Upozornění: Pokud režim zpracování blok po bloku deaktivujete v oblasti zpožděného zastavení, NCK se zastaví až v 1. bloku za touto oblastí. Pokud je režim zpracování blok po bloku aktivován už před oblastí zpožděného zastavení, NCK se zastaví na každé hranici bloku (i v oblasti zpožděného zastavení).
SINGLEBLOCK_IPO	zpožděná	VDI: Aktivování režimu zpracování blok po bloku typ 1
SINGLEBLOCK_DECODIER	zpožděná	VDI: Aktivování režimu zpracování blok po bloku typ 2
STOPALL	okamžitá	VDI: DB21.DBB7.bit4 a DB11.DBB20.bit6
STOPPROG	zpožděná	VDI: DB21.DBB7.bit3 a DB11.DBB20.bit5
OVERSTORE_BUFFER_END_ REACHED	alarm 16954	NC-Prog: Zastavení kvůli prázdné paměti Overstore.
PREP_STOP	alarm 16954	NC-Prog: STOPRE a všechna implicitní zastavení předběžného zpracování
PROG_STOP	alarm 16954	NC-Prog: M0 a M1
STOPPROGATBLOCKEND	zpožděná	VDI: DB21.DBB7bit2
STOPPROGATASUPEND	syst. chyba	Konec UP by měl oblast zpož. zast. vždy deaktivovat
WAITM	alarm 16954	NC-Prog: WAITM
WAITE	alarm 16954	NC-Prog: WAITE
INIT_SYNC	alarm 16954	NC-Prog: INIT s parametrem „S“
MMCCMD	alarm 16954	NC-Prog: MMC (STRING, CHAR)
PROGMODESLASHON	zpožděná	VDI: DB21.DBB26 Aktivování/přepnutí přeskakování bloků
PROGMODESLASHOFF	zpožděná	VDI: DB21.DBB26 Deaktivování přeskakování bloků
PROGMODEDRYRUNON	zpožděná	VDI: DB21.DBB0.bit6 Aktivování zkušebního režimu
PROGMODEDRYRUNOFF	zpožděná	VDI: DB21.DBB0.bit6 Deaktivování zkušeb. režimu
BLOCKREADINHIBIT_ON	zpožděná	VDI: DB21.DBB6.bit1 Aktivování blokování načítání
STOPATEND_ALARM	okamžitá	Alarm: Konfigurace alarmu STOPATENDBYALARM
STOP_ALARM	okamžitá	Alarm: Konfigurace alarmu STOPBYALARM
STOPATIPOBUFFER_ISEMPTY_ALARM	okamžitá	Interní: Stop po alarmu při prázdném zásobníku interpolace
STOPATIPOBUF_EMPTY_ALARM_ REORG	okamžitá	Interní: Stop po alarmu při prázdném zásobníku interpolace
RETREAT_MOVE_THREAD	alarm 16954	NC-Prog: Alarm 16954 při LFON (Stopp a Fastlift v G33 nejsou možné)
WAITMC	alarm 16954	NC-Prog: WAITMC
NEWCONF_PREP_STOP	alarm 16954	NC-Prog: NEWCONF
BLOCKSEARCHRUN_NEWCONF	alarm 16954	NC-Prog: NEWCONF
SET_USER_DATA	zpožděná	BTSS: PI „_N_SETUDT“
SYSTEM_SHUTDOWN	okamžitá	Vypnutí systému u 840i
ESR	zpožděná	Rozšířené zastavení a odsunutí
EXT_ZERO_POINT	zpožděná	Externí posunutí počátku
STOPRUN	alarm 16955	BTSS: PI „_N_FINDST“ STOPRUN

© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení **reakcí**

okamžitá („tvrdá“ událost způsobující zastavení)	K zastavení dojde okamžitě i v oblasti zpožděného zastavení.
zpožděná („měkká“ událost způsobující zastavení)	Zastavení (i krátkodobé) následuje teprve za oblastí zpožděného zastavení.
alarm 16954	Program se přeruší, protože v oblasti zpožděného zastavení byly použity nepřípustné programové příkazy.
alarm 16955	Program bude pokračovat, v oblasti zpožděného zastavení byly uskutečněny nepřípustné akce.

Výhody oblasti se zpožděným zastavením

Úsek programu bude zpracován beze změny rychlosti. Pokud je program, který se zastavil, přerušen uživatelem pomocí funkce Reset, bude se přerušený programový blok nacházet za chráněnou oblastí. Tento programový blok se pak hodí jako cíl pro následné vyhledávání.

Dokud probíhá zpracování oblasti se zpožděním zastavením, následující hlavní osy se nezastaví:
- příkazové osy a
- polohovací osy ovládané příkazem POSA

Příkaz výrobního programu G4 je v oblasti se zpožděným zastavení přípustný, oproti tomu jsou však jiné příkazy výrobního programu, které způsobují dočasné zastavení (např. WAITM), nepřípustné.

G4 aktivuje, jako pohyb po dráze, oblast se zpožděným zastavením, příp. zachová její platnost.

Příklad: Zásahy do posuvu

Pokud korekce (override) **před** oblastí se zpožděným zastavením klesne na 6%, bude tato korekce v platnosti i v oblasti se zpožděným zastavením. Pokud korekce (override) klesne na 6% **v** oblasti se zpožděným zastavením, bude posuv probíhat s korekcí 100% až na konec oblasti se zpožděným zastavením a pak bude se uplatní korekce 6%.

Zastavení posuvu se v oblasti se zpožděným zastavením neuplatní, k zastavení dojde až po opuštění této oblasti.

© Siemens AG 2004, Všechna práva vyhrazena

Překrývání/vnoření
Pokud se dvě oblasti se zpožděným zastavením překrývají, jedna díky příkazu v programu a druhá ze strojního parametru 11550, bude vytvořena největší možná oblast se zpožděným zastavením.

Vzájemné vztahy mezi příkazy DELAYFSTON a DELAYFSTOF s vnořením a s koncem podprogramu se řídí následujícími zásadami:

1. S koncem podprogramu, v němž byl volán příkaz DELAYFSTON, se implicitně aktivuje příkaz DELAYFSTOF.
2. Oblast se zpožděným zastavením s DELAYFSTON zůstane bez efektu.
3. Pokud podprogram 1 v oblasti se zpožděným zastavením volá podprogram 2, je celý podprogram 2 zpracován v oblasti se zpožděným zastavením. Zejména příkaz DELAYFSTOF je v podprogramu 2 bez efektu.

Poznámka: Příkaz REPOSA je koncem podprogramu a příkaz DELAYFSTON je tedy v každém případě deaktivován.
Vyskytne-li se v oblasti se zpožděným zastavením „tvrdá“ událost způsobující zastavení, bude celá tato oblast deaktivována! Tzn. vyskytne-li se v tomto úseku programu další libovolné pozastavení, bude ihned provedeno. Teprve po dalším naprogramování příkazu DELAYFSTON začne nová oblast se zpožděným zastavením.
Pokud je tlačítko STOP stisknuto před oblastí se zpožděným zastavením a pokud NCK musí kvůli zabrždění najet do oblasti se zpožděným zastavením, NCK zastaví v této oblasti a oblast bude deaktivována!

V oblasti se zpožděným zastavením je možné zabrzdit příkazem STOPALL. Tímto příkazem se ale aktivují všechny ostatní události způsobující zastavení, které byly dosud zpožděné.

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování

Vnoření oblastí se zpožděným zastavením na dvou programových úrovních:

```
N10010 DELAYFSTON()                ; bloky s N10xxx – programová úroveň 1
N10020 R1 = R1 + 1
N10030 G4 F1                       ; začátek oblasti se zpožděným zastavením
...
N10040 Podprogram2
...
      ...                          ; interpretace podprogramu 2
      N20010 DELAYFSTON()          ; bez efektu, opakovaný začátek, 2. úroveň
      ...
      N20020 DELAYFSTOF()          ; bez efektu, konec na jiné úrovni
      N20030 RET

N10050 DELAYFSTOF()                ; konec oblasti se zpožděným zastavením
                                     na téže úrovni
...
N10060 R2 = R2 + 2
N10070 G4 F1                       ; konec oblasti se zpožděným zastavením.
                                     Zastavení se od této chvíle provádějí hned.
```

© Siemens AG 2004, Všechna práva vyhrazena

Systémové proměnné:
Oblast se zpožděným zastavením může být ve výrobním programu ¨rozpoznána pomocí proměnné **$P_DELAYFST**. Pokud je bit 0 této systémové proměnné nastaven na 1, nachází se zpracování výrobního programu v daném okamžiku v oblasti se zpožděným zastavením.
Pomocí proměnné **$AC_DELAYFST** může být oblast se zpožděným zastavením rozpoznána v synchronní akci. Pokud je bit 0 této systémové proměnné nastaven na 1, nachází se zpracování výrobního programu v daném okamžiku v oblasti se zpožděným zastavením.

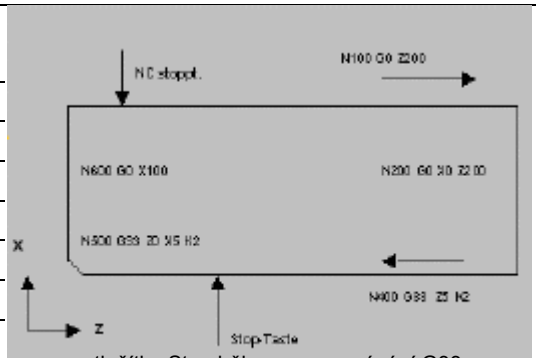
Další upozornění

Kompatibilita:
Pokud je bit 0 strojního parametru 11550: STOP_MODE_MASK nastaven na 0, implicitně vznikne oblast se zpožděným zastavením během příkazů skupiny G-kódů G331/G332 a když je naprogramován pohyb po dráze, příp. G4.
Bit 0 = 1 umožní zastavení v průběhu příkazů skupiny G-kódů G331/G332 a když je naprogramován pohyb po dráze, příp. G4 (chování do SW 6). Pro definici oblast se zpožděným zastavením musí být použity příkazy DELAYFSTON/DELAYFSTOF.

Příklad programu ukazující použití

```
...
N99  MY_LOOP:
N100 G0 Z200
N200 G0 X0 Z200
N300 DELAYFSTON()
N400 G33 Z5 K2 M3 S1000
N500 G33 Z0 X5 K3
N600 G0 X100
N700 DELAYFSTOF()
N800 GOTOB MY_LOOP
...
```

; ve smyčce se opakuje následující programový blok



tlačítko Stop během zpracovávání G33

V obrázku je vidět, že uživatel stiskl v oblasti se zpožděným zastavením tlačítko „Stop“ a že NC systém zahájil operaci brždění mimo oblast se zpožděným zastavením, tzn. v bloku N100. Tím se NC zastaví v přední oblasti před blokem N100.

© Siemens AG 2004, Všechna práva vyhrazena



Podrobnosti o vyhledávání bloku typu SERUPRO viz:
/FB/, K1, Kanál, Programový režim, chování v klidu
/FB/, V1, Posuvy při obrábění pomocí G331/G332, Posuv při vrtání závitů bez vyrovnávací hlavičky.

9.8 Zablokování programového místa pro SERUPRO, IPTRLOCK, IPTRUNLOCK



Programování

```
N... IPTRLOCK
N... IPTRUNLOCK
```

Příkazy se nacházejí v samostatném řádku výrobního programu a umožňují programovatelný ukazatel místa přerušení.



Vysvětlení

IPTRLOCK	Začátek vyhledatelného úseku programu
IPTRUNLOCK	Konec vyhledatelného úseku programu
	Oba příkazy se nacházejí pouze ve výrobních programech, v synchronních akcích jsou nepřípustné.



Funkce

Pro určité komplikované mechanické situace na stroji je zapotřebí zabránit vyhledávání bloku SERUPRO. Pomocí programovatelného ukazatele přerušení existuje možnost zásahu při aktivování funkce "vyhledávání místa přerušení“ dostat se před místo, které je schopné vyhledání. Mohou být také definovány vyhledatelné oblasti ve výrobním programu, do kterých se už NCK nemůže dostat. Při přerušení programu si NCK zaznamená naposled zpracovávaný blok, který pak může být pomocí uživatelského rozhraní HMI vyhledán.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Záznam a vyhledávání těchto oblastí

Vyhledatelné úseky programu jsou označeny příkazy IPTRLOCK a IPTRUNLOCK.
Příkaz IPTRLOCK zablokuje ukazatele přerušení na určitém zpracovatelném bloku (SBL1). Tento blok je v následujících odstavcích označován jako „blok zastavení“. Pokud se po příkazu IPTRLOCK vyskytne přerušení programu, je možné na uživatelském rozhraní HMI tento tzv. blok zastavení vyhledat.

Opětovné nastavení na aktuální blok

Pomocí příkazu IPTRUNLOCK se ukazatel přerušení pro následující úsek programu nastaví na aktuální blok.

Po nalezení cíle vyhledávání je možné se stejným zastavovacím blokem operaci opakovat pro nový cíl vyhledávání.

Další upozornění

Pravidla pro vnoření

Vzájemné vztahy mezi příkazy IPTRUNLOCK a IPTRLOCK s vnořením a s koncem podprogramu se řídí následujícími zásadami:

1. S koncem podprogramu, v němž byl volán příkaz IPTRLOCK, se implicitně aktivuje příkaz IPTRUNLOCK.
2. Příkaz IPTRLOCK ve vyhledávatelné oblasti zůstane bez efektu.
3. Pokud podprogram 1 ve vyhledávatelné oblasti volá podprogram 2, zůstává celý podprogram 2 vyhledávatelný. Zejména příkaz IPTRUNLOCK je v podprogramu 2 bez efektu.

Systémové proměnné

Vyhledávatelná oblast může být ve výrobním programu rozpoznána pomocí proměnné **$P_IPTRLOCK**.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Vnoření vyhledatelných úseků programu na dvou programových úrovních s implicitním příkazem IPTRUNLOCK. Vyhledávatelná oblast v podprogramu 1 je ukončena implicitním IPTRUNLOCK.

```
N10010 IPTRLOCK()
N10020 R1 = R1 + 1
N10030 G4 F1                    ; blok zastavení, začátek vyhledatelného
...                             ; úseku programu
N10040 Unterprogramm2
...                             ; interpretace podprogramu 2
N20010 IPTRLOCK ()              ; bez efektu, opakovaný začátek
...
N20020 IPTRUNLOCK ()            ; bez efektu, konec na jiné úrovni
N20030 RET
...
N10060 R2 = R2 + 2
N10070 RET                      ; konec vyhledatelného úseku programu
N100   G4 F2                    ; pokračování hlavního programu
```

Automatický ukazatel přerušení
Funkce automatického ukazatele přerušení automaticky definuje dříve stanovený druh spojení jako vyhledatelný. Pomocí strojního parametru MD22680: AUTO_IPR_LOCK se automatický ukazatel přerušení aktivuje pro:

- Elektronické převodovky s příkazem EGON
- Spojení os řídící hodnotou s příkazem LEADON

Pokud se naprogramovaný a automatický ukazatel místa přerušení (jeden z příkazu v programu a druhý z MD22680: AUTO_IPR_LOCK) překrývají, vznikne největší možná vyhledatelná oblast.

Další informace viz:
/FB/, K1, Kanál, Programový režim, chování v klidu

© Siemens AG 2004, Všechna práva vyhrazena

9.9 Opětovné najíždění na konturu, REPOSA/L, REPOSQ/H, RMI/N, RMB/E

Programování

REPOSA RMI DISPR=… nebo REPOSA RMB nebo REPOSA RME nebo REPOSA RMN

REPOSL RMI DISPR=… nebo REPOSL RMB nebo REPOSL RME nebo REPOSL RMN

REPOSQ RMI DISPR=… DISR=… nebo REPOSQ RMB DISR=… nebo REPOSQ RME DISR=… nebo REPOSQ RMN DISR=…

REPOSH RMI DISPR=… DISR=… nebo REPOSH RMB DISR=… nebo REPOSH RME DISR=… nebo REPOSH RMN DISR=…

Vysvětlení příkazů

Pohyb při najíždění

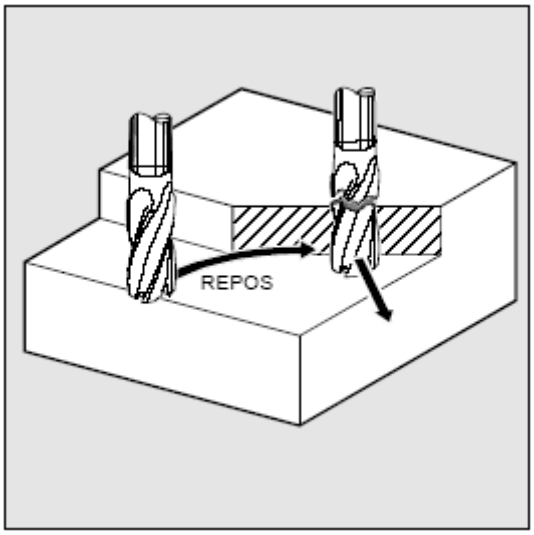
REPOSA	Najíždění po přímce se všemi osami
REPOSL	Najíždění po přímce
REPOSQ DISR=…	Najíždění po čtvrtkruhu s rádiusem DISR
REPOSQA DISR=…	Najíždění po čtvrtkruhu s rádiusem DISR se všemi osami
REPOSH DISR=…	Najíždění po půlkruhu s průměrem DISR
REPOSHA DISR=…	Najíždění po půlkruhu s průměrem DISR se všemi osami

Opětovné najíždění na bod

RMI	Najíždění na bod, kde došlo k přerušení
RMI DISPR=…	Vstupní bod ve vzdálenosti DISPR v mm/palcích **před** bodem, kde došlo k přerušení
RMB	Najíždění na počáteční bod bloku
RME	Najíždění na koncový bod bloku
RME DISPR=…	Najíždění na koncový bod bloku ve vzdálenosti DISPR před tímto bodem
RMN	Najíždění na následující bod dráhy
A0 B0 C0	Osy, ve kterých se má najíždět

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

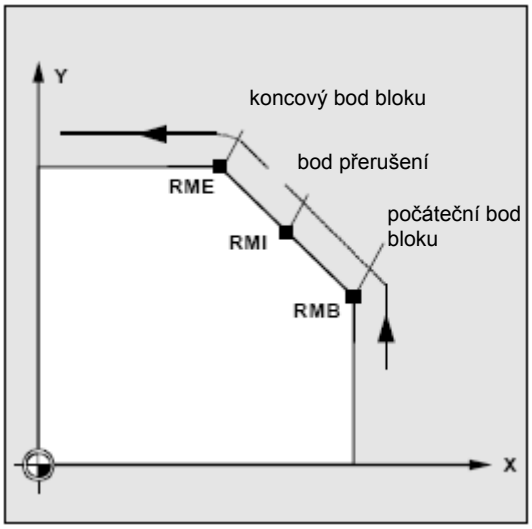


Jestliže je v průběhu obrábění právě zpracovávaný program přerušen a nástrojem se odjede – například kvůli zlomení nástroje nebo protože potřebujete něco změřit – můžete na konturu pomocí programového řízení znovu najet v bodě, který si zvolíte.

Příkaz REPOS se chová jako návrat z podprogramu (např. pomocí M17). Následující bloky v rutině přerušení se už neuskuteční.

Informace o přerušení programu viz kapitola „Rutina přerušení“ v Příručce programování „Pro pokročilé“.

Postup



Definice bodu pro opětovné najetí (od SW 6.2)
Vzhledem k NC bloku, v němž bylo zpracování programu přerušeno, si můžete vybrat ze tří bodů pro opětovné najetí:

- RMI, bod přerušení
- RMB, počáteční bod bloku, příp. poslední koncový bod
- RME, koncový bod bloku

Pomocí příkazů `RMI DISPR=…`, příp. `RME DISPR=…` můžete definovat bod pro zpětné najetí, který leží před bodem přerušení, příp. před koncovým bodem bloku.
Pomocí `DISPR=…` popisujete dráhu na kontuře v mm/palcích, o kterou bod pro zpětné najetí leží před bodem přerušení, resp. před koncovým bodem. Tento bod může – i pro větší hodnoty – ležet maximálně v počátečním bodě bloku.
Pokud příkaz `DISPLR=…` není naprogramován, platí `DISPR=0` a tedy i bod přerušení (v případě `RMI`), příp. koncový bod bloku (u `RME`).

Od SW 5.2:
Znaménko před DISPR se vyhodnocuje. Při kladném znaménku je chování stejné jako dříve. Při záporném znaménku se najíždí za bod přerušení, příp. u `RMB` za počáteční bod.

© Siemens AG 2004, Všechna práva vyhrazena

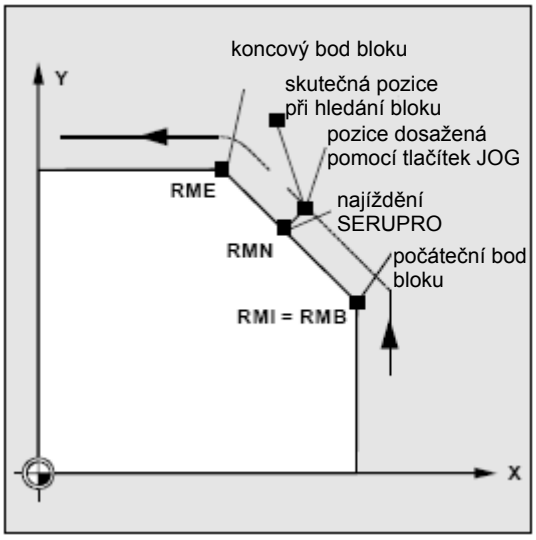
Vzdálenost bod přerušení – bod nasazení vyplývá z hodnoty v příkazu DISPR. I v případě větších hodnot může tento bod ležet maximálně v koncovém bodě bloku.

Příklad použití:

Snímač bude rozpoznávat přiblížení k uchopovacím kleštím. Aby se kleště obešly, aktivuje se ASUB. Potom se pomocí záporné hodnoty DISPR najede zpět na místo za kleštěmi a program bude pokračovat.

Najíždění pomocí SERUPRO s RMN (od SW 6.3)

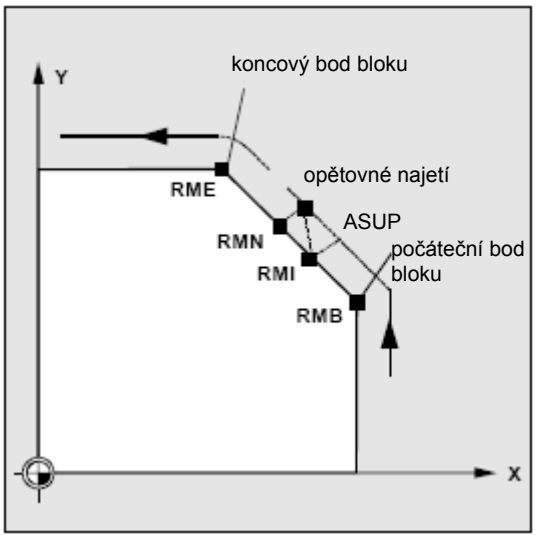
Jestliže se při obrábění na libovolném místě vyskytla nutnost zpracování přerušit, potom lze pomocí příkazu SERUPRO najet po nejkratší dráze s RMN před místo přerušení a nakonec obrobit pouze zbytkovou dráhu. Za tím účelem uživatel spouští operaci SERUPRO za přerušeným blokem a nastavuje polohu před místem poškození cílového bloku pomocí tlačítek JOG.



Pro příkaz SERUPRO jsou RMI a RMB identické. RMN není omezen pouze na SERUPRO, má všeobecnou platnost.

Najíždění na nejbližší bod na dráze RMN

V okamžiku interpretace příkazu REPOSA se po přerušení blok zpětného najíždění s RMN nezačíná ještě jednou úplně znovu, ale obrábí se pouze zbytková dráha. Najíždí se na nejbližší bod na dráze bloku, v němž došlo k přerušení.

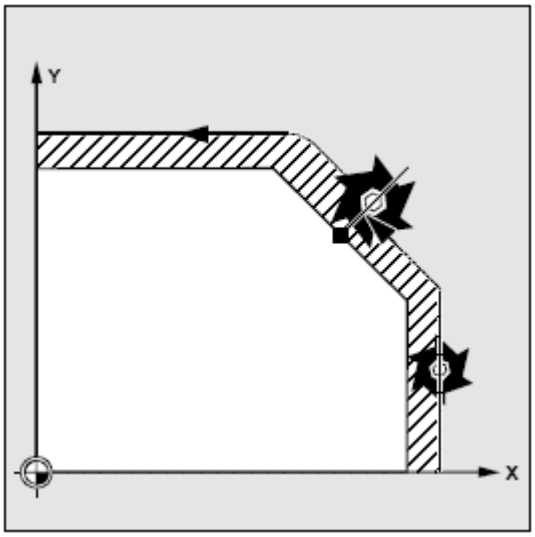


© Siemens AG 2004, Všechna práva vyhrazena

Systémové proměnné od SW 6.4:
Platný režim REPOS přerušeného bloku může být načten pomocí synchronní akce s proměnnou **$AC_REPOS_PATH_MODE**:

0: Najíždění není definováno
1 RMB: Najíždění na začátek
2 RMI: Najíždění na místo přerušení
3 RME: Najíždění na koncový bod bloku
4 RMN: Najíždění na bod na dráze bloku, v němž došlo k přerušení.

Najíždění novým nástrojem
Jestliže bylo zpracování programu přerušeno v důsledku zlomení nástroje:
Naprogramováním nového D-čísla bude program pokračovat od místa zpětného najetí se změněnými hodnotami korekcí nástroje.



Může se stát, že při změněných hodnotách korekcí nástroje už nebude možné najet na místo, kde došlo k přerušení. V tomto případě se bude najíždět na bod na nové kontuře, který se nachází co možno nejblíže místa, kde došlo k přerušení (v případě potřeby s posunem o vzdálenost DISPR).

Najíždění na konturu
Pohyb, při kterém nástroj najíždí zpátky na konturu, je programovatelný. Adresám os, jež se mají pohybovat, zadejte nulovou hodnotu.

Pomocí příkazů REPOSA, REPOSQA a REPOSHA se provádí návrat všech os do původní polohy. Není zapotřebí zadat žádné osy.

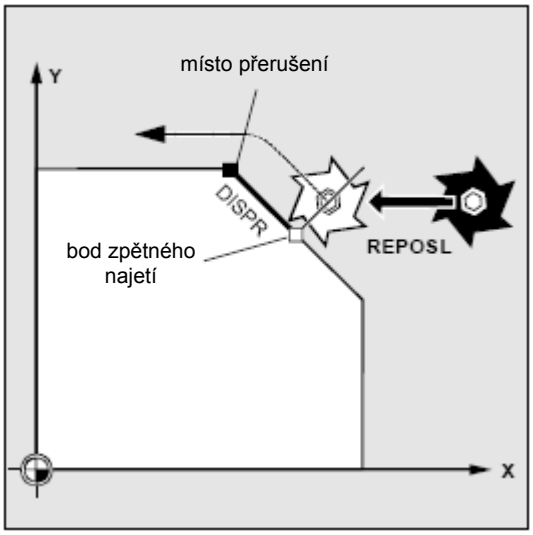
Při naprogramování příkazů REPOSA, REPOSQ a REPOSH se automaticky pohybují všechny geometrické osy, i když nebyly v příkazu uvedeny. Všechny ostatní osy, které mají být uvedeny do původní polohy, musí být v příkazu uvedeny.

© Siemens AG 2004, Všechna práva vyhrazena

Najíždění po přímkách, REPOSA, REPOSL
Nástroj se pohybuje do bodu zpětného najetí rovně po přímce.

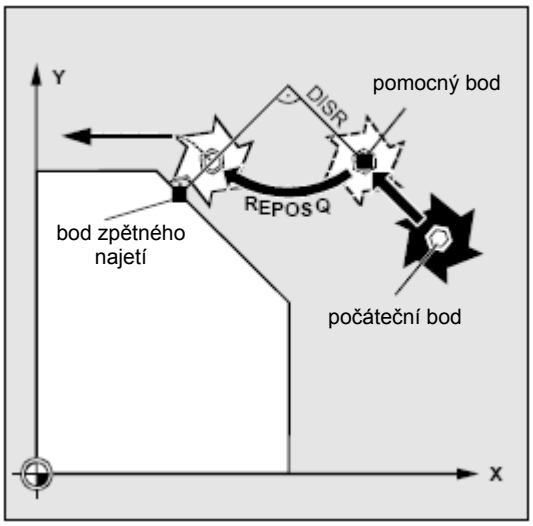
Pomocí příkazu REPOSA se automaticky pohybují všechny osy. U příkazu REPOSL je možné osy, které se mají pohybovat, definovat.

Příklad:
```
REPOSL RMI DISPR=6 F400
```
nebo
```
REPOSA RMI DISPR=6 F400
```



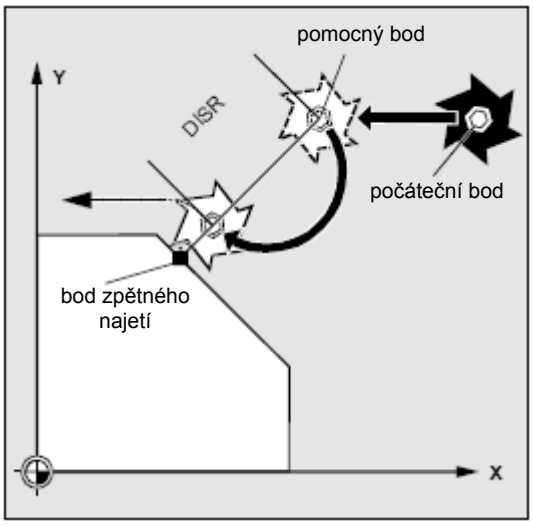
Najíždění po čtvrtkruhu, REPOSQ, REPOSQA
Nástroje se pohybuje do bodu pro zpětné najetí po čtvrtkruhu s rádiusem DISR=…. Nezbytný pomocný bod mezi počátečním bodem a bodem zpětného najetí vypočítá řídící systém automaticky sám.

Příklad:
```
REPOSQ RMI DISR=10 F400
```



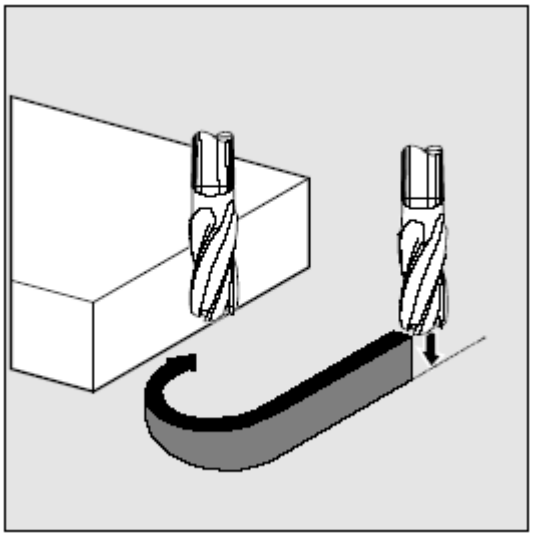
Najíždění po půlkruhu, REPOSH, REPOSHA
Nástroje se pohybuje do bodu pro zpětné najetí po půlkruhu s průměrem DISR=…. Nezbytný pomocný bod mezi počátečním bodem a bodem zpětného najetí vypočítá řídící systém automaticky sám.

Příklad:
```
REPOSH RMI DISR=20 F400
```



© Siemens AG 2004, Všechna práva vyhrazena

Pro pohyby po kruhové dráze REPOSH a REPOSQ platí:
Kruhový oblouk se bude uskutečňovat v zadané pracovní rovině G17 až G19.
Jestliže je v bloku pro najetí naprogramována třetí geometrická osa (směr přísuvu), bude se v případě, že si poloha nástroje a naprogramovaná pozice ve směru přísuvu neodpovídají, najíždět po spirále.



V následujících případech se automaticky přepne na lineární najíždění pomocí příkazu REPOSL:

- Není zadána žádná hodnota parametru DISR.
- Není definován žádný směr najíždění (přerušení v bloku bez informací o posuvu).
- V případě směru najíždění, který je kolmý na aktuální pracovní rovinu.

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Synchronní pohybové akce

10.1 Struktura, všeobecné základy ... 10-419
10.1.1 Programování a příkazové prvky ... 10-421
10.1.2 Oblast platnosti: Identifikační číslo ID ... 10-422
10.1.3 Klíčové slovo ... 10-423
10.1.4 Akce ... 10-426
10.1.5 Přehled synchronních akcí ... 10-427

10.2 Základní prvky pro podmínky a akce ... 10-429

10.3 Speciální proměnné v reálném čase pro synchronní akce ... 10-432
10.3.1 Ukazatel/čítač $AC_MARKER[n] ... 10-432
10.3.2 Proměnná časovače $AC_TIMER[n], od SW 4 ... 10-432
10.3.3 Parametr synchronní akce $AC_PARAM[n] ... 10-433
10.3.4 Přístup k početním parametrům $Rxx ... 10-434
10.3.5 Načítání/zápis strojních/nastavovaných parametrů, od SW 4 ... 10-435
10.3.6 Proměnné FIFO $AC_FIFO1[n] ... $AC_FIFO10[n], od SW 4 ... 10-436
10.3.7 Informace o typu bloku v interpolátoru (SW 6.4 a 7.1) ... 10-438

10.4 Operace v synchronních akcích ... 10-441
10.4.1 Výstup pomocných funkcí ... 10-441
10.4.2 Aktivování blokování načítání, RDISABLE ... 10-442
10.4.3 Deaktivování zastavení předběžného zpracování, STOPREOF ... 10-443
10.4.4 Vymazání zbytkové dráhy ... 10-444
10.4.5 Vymazání zbytkové dráhy s přípravou, DELDTG, DELDTG (osa 1 až X) ... 10-444
10.4.6 Definice polynomu, FCTDEF, bloková synchronizace ... 10-446
10.4.7 Řízení výkonu laseru ... 10-448
10.4.8 Vyhodnocovací funkce SYNFCT ... 10-449
10.4.9 AC regulace (aditivní) ... 10-450
10.4.10 AC regulace (multiplikativní) ... 10-451
10.4.11 Regulace vzdálenosti s omezenou korekcí ... 10-452
10.4.12 On-line korekce nástroje, FTOC ... 10-454
10.4.13 Pohyby při najíždění do polohy ... 10-456
10.4.14 Polohování osy, POS ... 10-458
10.4.15 Spuštění/zastavení osy, MOV ... 10-458
10.4.16 Posuv osy, FA ... 10-459
10.4.17 Softwarový koncový spínač ... 10-459
10.4.18 Koordinace os ... 10-460
10.4.19 Dosazení skutečné hodnoty ... 10-461
10.4.20 Pohyby vřetena ... 10-462
10.4.21 Vlečení: TRAILON, TRAILOF ... 10-463
10.4.22 Spojení řídící hodnotou, LEADON, LEADOF ... 10-464
10.4.23 Měření ... 10-466
10.4.24 Stanovení/vymazání značky pro čekání, SETM, CLEARM (od SW 5.2) ... 10.466
10.4.25 Reakce v případě chyby ... 10.467

© Siemens AG 2004, Všechna práva vyhrazena

10.4.26 Najíždění na pevný doraz, FXS a FOCON/FOCOF 10.467
10.4.27 Stanovení parametru s v synchronních akcích 10.470
10.4.28 Stanovení aktuální korekce (override) 10.470
10.4.29 Vyhodnocování vytížení pomocí časové náročnosti synchronní akce 10.471

10.5 Technologické cykly 10-473
10.5.1 Zablokování, odblokování, přerušení, LOCK, UNLOCK, RESET 10-475

10.6 Vymazání synchronní akce, CANCEL 10-477

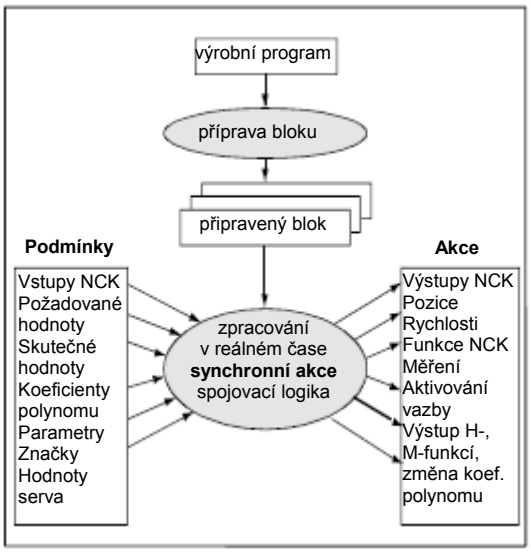
10.7 Okrajové podmínky 10-477

© Siemens AG 2004, Všechna práva vyhrazena

10.1 Struktura, všeobecné základy

Funkce

Synchronní akce nabízejí možnost spouštět z aktuálního výrobního programu různé akce a zpracovávat je synchronně vůči němu. Okamžik spuštění těchto akcí může být definován pomocí podmínek, jejichž vyhodnocování probíhá v reálném čase (v interpolačním taktu). Akce tedy představují reakce na události v reálném čase; jejich uskutečnění není vázáno na hranice bloků. Synchronní akce kromě toho obsahují informace o efektivitě akce a o frekvenci, s jakou jsou skenovány proměnné v reálném čase a tedy i o frekvenci, s jakou jsou akce spouštěny. Díky tomu může být akce spuštěna jen jednou nebo cyklicky (podle interpolačního cyklu).



Programování

```
DO akce1 Akce2 ...
Klíčové slovo Podmínka DO akce1 Akce2
ID=n Klíčové slovo Podmínka DO akce1 Akce2
IDS=n Klíčové slovo Podmínka DO akce1 Akce2
```

Vysvětlení

Identifikační číslo ID/IDS	
`ID=n`	Synchronní akce s **modální** platností v automatickém režimu, **lokální v programu**; n = 1 ... 255
`IDS=n`	Synchronní akce s **modální** platností ve všech provozních režimech, **statické**; n = 1 ... 255
`bez ID/IDS`	Synchronní akce s **blokovou** platností v automatickém režimu
Klíčové slovo	
`žádné klíčové slovo`	Provedení akce není vázáno na žádnou podmínku. Akce se bude provádět cyklicky v každém interpolačním cyklu.
`WHEN`	Podmínka bude kontrolována tak dlouho, dokud nebude jednou splněna; příslušná akce se uskuteční jednou.
`WHENEVER`	Podmínka je cyklicky kontrolována. Příslušná akce se bude provádět cyklicky, kdykoli bude podmínka splněna.
`FROM`	Po jednom splnění podmínky bude akce tak dlouho cyklicky prováděna, dokud bude synchronní akce aktivní.

© Siemens AG 2004, Všechna práva vyhrazena

EVERY	Akce se spustí jednou, když je podmínka splněna, a potom bude prováděna pokaždé, když se podmínka změní ze stavu FALSE do stavu TRUE. Podmínka je cyklicky kontrolována. Pokaždé, když je splněna, příslušná akce se jedenkrát uskuteční.
Podmínka	Spojovací logika pro proměnné v reálném čase, podmínka bude vyhodnocována v taktu IPO. Od **SW 5** jsou G-kódy v synchronních akcích pro vyhodnocování podmínky programovatelné.
DO	Pokud je podmínka splněna, spouští danou akci.
Akce	Akce spouštěná při splněné podmínce, např. přiřazení proměnné, aktivování spojení os, nastavení výstupu NCK, výstup M-, H-funkcí atd. Od **SW 5** jsou G-kódy v synchronních akcích pro akce/technologické cykly programovatelné.
Kódování synchronních akcí/technologických cyklů	
CANCEL[n]	Vymazání synchronní akce
LOCK[n]	Zablokování technologického cyklu
UNLOCK[n]	Odblokování technologického cyklu
RESET	Reset technologického cyklu



Příklad programování

WHEN $AA_IW[Q1]>5 DO M172 H510	; Pokud skutečná hodnota osy Q1 překročí 5 mm, na rozhraní PLC budou odeslány pomocné funkce M172 a H510.



Pokud se ve výrobním programu vyskytnou proměnné v reálném čase (např. skutečná hodnota, nastavení digitálních vstupů nebo výstupů atd.), předběžné zpracování se zastaví, dokud není zpracován předcházející blok a dokud nejsou hodnoty proměnných v reálném čase získány. Použité proměnné v reálném čase jsou vyhodnocovány v taktu interpolace.

Výhoda synchronních akcí:

Nedochází zde k žádnému zastavení předběžného zpracování.

Možné případy použití:

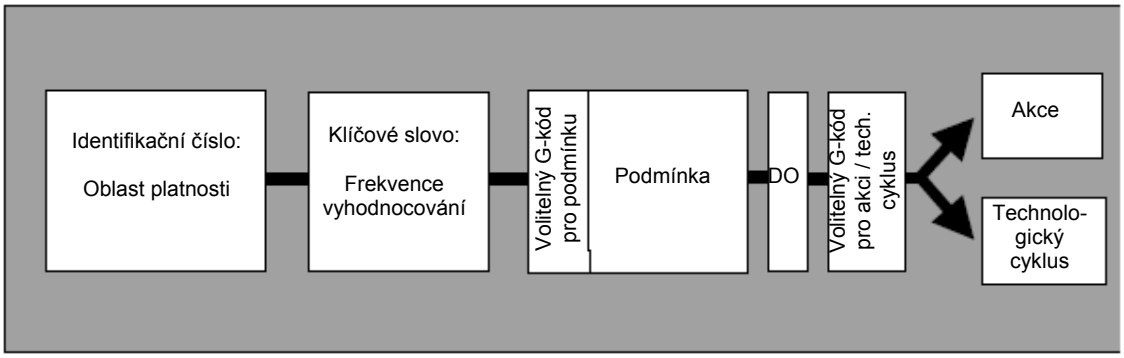
- Optimalizace aplikací, které kritické z hlediska strojového času (např. výměna nástroje)
- Rychlé reakce na externí události
- Programování AC pravidel
- Vytváření bezpečnostních funkcí
- ...

© Siemens AG 2004, Všechna práva vyhrazena

10.1.1 Programování a příkazové prvky

Funkce

Synchronní akce se nachází v samostatném bloku a v následujícím proveditelném bloku ovlivňuje nějakou funkci stroje (např. posuvy pomocí funkcí G0, G1, G2, G3).
Synchronní akce se mohou skládat z až 5 příkazových prvků s odlišnými úlohami:



Příklad:

`ID=1`	`WHENEVER`	`$A_IN[1]==1`	`DO`	`$SA_OUT[1]=1`
Synchronní akce č.1	pokaždé když	vstup 1 aktivní	pak	aktivovat výstup 1

10.1.2 Oblast platnosti: Identifikační číslo ID



Funkce

Oblast platnosti synchronní akce je definována identifikačním číslem (modální identifikace):

- **Žádná** modální identifikace
  Synchronní akce je v platnosti jen v automatickém režimu. Platí jen pro následující proveditelný blok (blok s příkazem pohybu nebo jiná akce stroje), a má také **blokovou** platnost.
  Příklad:

```
WHEN $A_IN[3]==TRUE DO$AOUTA[4]=10
G1 X20                                  ; proveditelný blok
```

- **ID**=n; n = 1 ... 255
  Synchronní akce funguje v následujících blocích modálně a deaktivuje se příkazem CANCEL(n) nebo naprogramováním jiné synchronní akce se stejnou identifikací.
  Synchronní akce, které jsou v platnosti v bloku s M30, zůstávají v platnosti i nadále (v případě potřeby je vymažte příkazem CANCEL).
  Synchronní akce jsou v platnosti **pouze** v **automatickém režimu**.
  Příklad:

```
ID=2 EVERY $A_IN[1]==1 DO POS[X]=0
```

- **IDS**=n; n = 1 ... 255
  Tyto **statické** synchronní akce mají **modální** platnost ve **všech provozních režimech**.
  Mohou být definovány nejen z výrobního programu, ale i přímo po zapnutí (Power On) z asynchronně spuštěného podprogramu z PLC (ASUP). Tak mohou být aktivovány akce, které mají probíhat v NC nezávisle na zvoleném provozním režimu.
  Příklad:

```
IDS=1 EVERY $A_IN[1]==1 DO POS[X]=100
```


© Siemens AG 2004, Všechna práva vyhrazena

Použití:

- AC broušení v režimu JOG
- Spojovací logika pro modul Safety Integrated
- Monitorovací funkce, reakce na stavy stroje ve všech provozních režimech

Posloupnost zpracování

Statické a modální synchronní akce jsou zpracovávány v pořadí jejich identifikačních čísel (v interpolačním taktu).
Synchronní akce s blokovou platností (bez identifikačního čísla) se spouští v naprogramované posloupnosti až po zpracování modálních synchronních akcí.

10.1.3 Klíčové slovo

Funkce

Klíčové slovo určuje, jak často je následující podmínka kontrolována a jak často jsou příslušné akce prováděny:

- Žádné klíčové slovo:
  Pokud není naprogramováno žádné klíčové slovo, platí podmínka vždy jako splněná. Synchronní příkazy jsou cyklicky prováděny.
- WHEN
  V každém interpolačním taktu bude podmínka tak dlouho kontrolována, dokud není jednou splněna, akce se pak provede právě jednou.
- WHENEVER
  Podmínka je kontrolována v každém interpolačním taktu. Akce bude prováděna v každém interpolačním taktu, dokud je podmínka splněna.
- FROM
  Podmínka bude v každém interpolačním taktu kontrolována, dokud není jednou splněna. Akce bude od té chvíle uskutečňována tak dlouho, dokud není synchronní akce zrušena, tzn. i tehdy, když podmínka už není splněna.

Příklad:

```
DO $A_OUTA[1]=$AA_IN[X]
```
; skutečná hodnota na analogový výstup

© Siemens AG 2004, Všechna práva vyhrazena

- EVERY
  Podmínka bude kontrolována v každém interpolačním taktu. Akce pak bude vždy uskutečněna jen jednou, je-li podmínka splněna.
  Řízení hranou:
  Akce se bude provádět, když se stav podmínky změní z ALSE na TRUE.

Podmínka

Porovnáním dvou proměnných v reálném čase nebo proměnné v reálném čase s výrazem vypočítaným v předběžném zpracování definuje, zda se má akce uskutečnit.

Od SW 4:

V podmínce mohou být výsledky porovnávání spojovány také pomocí boolovských operátorů ( ). Podmínka je kontrolována v interpolačním taktu. Pokud je splněna, příslušná akce se uskuteční.

Od SW 5:

Podmínky mohou být zadávány i s G-kódy. To znamená, že je možné definovat nastavení pro vyhodnocování podmínky a akce/technologického cyklu bez ohledu na stav právě aktivního výrobního programu. Je však nezbytné oddělit synchronní akce od programového prostředí, protože synchronní akce mají provádět své akce v definovaném počátečním stavu, kdykoli jsou spouštěcí podmínky splněny.

Příklady použití:

Definice měřicího systému pro vyhodnocování podmínky a akci pomocí G-kódů G70, G71, G700 a G710.

Ve verzi SW 5 jsou přípustné pouze tyto G-kódy. Uvedený G-kód pro podmínku platí pro vyhodnocování podmínky, **stejně jako** pro akci, pokud u akce není žádný G-kód uveden.

V každé části podmínky smí být naprogramován jen jeden G-kód z této skupiny.

Příklad:

```
ID=1 EVERY $AA_IM[B]>75 DO
POS[U]=IC(10) FA[U]=900
```

Pokud skutečná hodnota osy B v MCS překračuje 75, má se osa U posunout o 10 dále.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

`WHENEVER $AA_IM[X] > 10.5*SIN(45) DO…`	porovnávání s výrazem vypočítaným v rámci předběžného zpracování
`WHENEVER $AA_IM[X] > $AA_IM[X1] DO…`	porovnávání s další proměnnou v reálném čase
`WHENEVER ($A_IN[1]==1)` **`OR`** `($A_IN[3]==0 DO…`	dvě spolu spojené porovnání

Možné podmínky:

- Porovnání proměnných v reálném čase (analogové/digitální vstupy/výstupy atd.)
- Boolovské spojení mezi výsledky porovnání
- Výpočet výrazů v reálném čase
- Čas/vzdálenost od začátku bloku
- Vzdálenost od konce bloku
- Změřené hodnoty, výsledky měření
- Hodnoty servomechanismů
- Rychlosti, stav os

© Siemens AG 2004, Všechna práva vyhrazena

10.1.4 Akce



Funkce

V každé synchronní akci může být naprogramována jedna nebo více akcí. Všechny akce naprogramované v jednom bloku se budou spouštět ve stejném interpolačním taktu:
Od **verze SW 5** se mohou používat akce s G-kódem pro akce/technologické cykly. Tento G-kód specifikuje G-kód z jiné skupiny, než jaká byla použita v podmínce, a platí pro všechny akce v bloku a technologickém cyklu.Pokud jsou v části akcí technologické cykly, platí tento G-kód i po skončení technologického cyklu pro všechny následující akce, dokud není zadán další G-kód s modální platností.
V jedné části akcí smí být naprogramován jen jeden G-kód ze skupiny (G70, G71, G700, G710).

Možné akce:

- Přiřazení proměnné
- Zápis do nastavovaného parametru
- Nastavení řídícího parametru
- DELDTG: rychlé vymazání zbytkové dráhy
- RDISABLE: aktivování blokování načítání
- Výstup pomocných M-, S- a H-funkcí
- STOPREOF:aktivování zastavení předběžného zpracování
- FTOC: on-line korekce nástroje
- Definice vyhodnocovacích funkcí (polynomy)
- SYNFCT: aktivování vyhodnocovacích funkcí: AC regulace
- Přepínání mezi více naprogramovanými posuvy v jednom bloku v závislosti na binárních nebo analogových signálech
- Korekce posuvu
- Spouštění/polohování/zastavování polohovacích os (POS) a vřeten (SPOS)
- PRESETON: nastavení skutečné hodnoty
- Aktivování nebo deaktivování vlečení/spojení řídící hodnotou
- Měření
- Nastavení dodatečných bezpečnost. funkcí
- Výstup digitálních a analogových signálů
- ...

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Synchronní akce se dvěma akcemi

```
WHEN $AA_IM[Y] >= 35.7 DO M135 $AC_PARAM=50
```

; Je-li podmínka splněna, na PLC bude vysláno M135 a korekce (override) bude nastavena na 50%.

Jako akce může být uveden také program (program pro jednotlivou osu, technologický cyklus). Ten se však smí skládat jen z takových akcí, které mohou být v synchronní akci také jednotlivě naprogramovány. Jednotlivé akce takového programu jsou zpracovávány postupně v interpolačním taktu.

Upozornění

Akce mohou být prováděny nezávisle na provozním režimu.
Akce, které jsou funkční jen v automatickém režimu při aktivním programu, jsou následující:

- STOPREOF
- DELDTG

10.1.5 Přehled synchronních akcí

Do SW 3.x

- Programování postupů v interpolačním taktu na úrovni uživatele (výrobní program)
- Reakce na události/stavy v interpolačním taktu
- Spojovací logika v reálném čase
- Přístup k periferiím, stavy řídícího systému a stroje
- Programování cyklických postupů, které se provádějí v interpolačním taktu
- Spouštění specifických NC-funkcí (blokování načítání, superponované pohyby os, ...)
- Zpracovávání technologických funkcí souběžně s pohybem po dráze
- Spouštění technologických funkcí nezávisle na hranicích bloků

© Siemens AG 2004, Všechna práva vyhrazena

Od SW 4:

- Diagnostické možnosti pro synchronní osy
- Rozšíření proměnných z hlavního zpracování, které lze použít v synchronních akcích
- Složité podmínky v synchronních akcích
- Rozšíření výrazů v synchronních akcích:
  Spojení proměnných v reálném čase se základními matematickými operacemi a funkcemi v taktu IPO,
  Nepřímá adresace proměnných hlavního zpracování s indexem může být měněna on-line,
  Nastavované parametry mohou být ze synchronních akcí on-line měněny a vyhodnocovány
- Konfigurovatelnost: pomocí strojních parametrů lze nastavit počet současně aktivních synchronních akcí
- Spouštění pohybů polohovacích os a vřeten ze synchronních akcí (příkazové osy)
- Předvolba ze synchronní akce
- Aktivování, deaktivování a dosazování parametrů spojení os: spojení řídící hodnotou, vlečení
- Aktivování/deaktivování měřicí funkce osy
- Softwarové vačky
- Vymazání zbytkové dráhy bez zastavení předběžného zpracování
- Programy pro jednotlivé osy, technologické cykly
- Aktivování synchronních akcí v režimu JOG, i za hranicemi programu
- Synchr. akce mohou být ovlivňovány z PLC
- Chráněné synchronní akce
- Rozšíření pro superponovaný pohyb / regulace vzdálenosti

Od SW 5.x:

- Najíždění na pevný doraz FXS: Spouštění synchronních akcí s FXS, FXST a FXSW.
- Najíždění s omezeným momentem/silou FOC:
  Aktivování (FOCON), příp. deaktivování (FOCOF) synchronní akce s modální nebo blokovou platností.

© Siemens AG 2004, Všechna práva vyhrazena

10.2 Základní prvky pro podmínky a akce

Proměnné v reálném čase
Proměnné v reálném čase jsou vyhodnocovány a zapisovány v interpolačním taktu.
Proměnnými v reálném čase jsou:

- $A... proměnné hlavního zpracování
- $V... proměnné servomechanismu

Kvůli zvláštnímu označení mohou být tyto proměnné v synchronních akcích programovány s **$$**: $AA_IM[X] je totéž co $$AA_IM[X]. Strojní a nastavované parametry musí být pomocí $$ označeny vždy, když se v taktu IPO provádí vyhodnocování nebo přiřazení.

Seznam proměnných naleznete v příloze.

Výpočty v reálném čase
Výpočty v reálném čase jsou omezeny na typy INT, REAL a BOOL.
Výrazy v reálném čase jsou výpočty, které mohou být provedeny v interpolačním taktu, které lze použít v podmínce a v akci pro přiřazení NC adrese a proměnné.

- **Porovnání**
  V podmínkách mohou být porovnávány proměnné nebo dílčí výrazy stejného datového typu. Výsledek je vždy datového typu BOOL.
  Přípustné jsou všechny známé logické operátory (==, <>, <=, >=).

- **Boolovské operátory**
  Proměnné, konstanty nebo výrazy mohou být spojovány známými boolovskými operátory (NOT, AND, OR, XOR).

- **Bitové operátory**
  Je možné používat také bitové operátory B_NOT, B_AND, B_OR, B_XOR.

© Siemens AG 2004, Všechna práva vyhrazena

- **Základní matematické operace**
  Proměnné v reálném čase typů INTEGER a REAL mohou být spojovány vzájemně nebo s konstantami také pomocí základních matematických operací (+, -, *, /, DIV, MOD).

- **Matematické funkce**
  Na proměnné v reálném čase datového typu REAL mohou být použity také matematické funkce (SIN, COS, TAN, ASIN, ACOS, ABS, TRUNC, ROUND, LN, EXP, ANAT2, POT, SQRT, CTAB, CTABINV).
  Příklad:

```
DO $AC_PARAM[3] = COS($AC_PARAM[1])
```


© Siemens AG 2004, Všechna práva vyhrazena

Upozornění

Spojovány smí být pouze proměnné téhož datového typu

Správně: $R10 = $AC_PARAM[1]
Nesprávně: $R10 = $AC_MARKER[1]

Při výpočtech mají přednost násobení a dělení před sčítáním a odčítáním, použití závorek je ve výrazech přípustné. Operátory DIV a MOD jsou přípustné i pro datový typ REAL (od SW 4).

Příklad:

```
DO $AC_PARAM[3] = $A_INA[1]-$AA_IM[21]  ; odečtení 2. proměnné v reálném čase
WHENEVER $AA_IM[x2] < $AA_IM[x1] – 1.9 DO $A_OUT[5] = 1
                                        ; odečtení konstanty od proměnné v reálném čase
DO $AC_PARAM[3] = $INA[1] – 4*SIN(45.7 $P_EP[Y])*R4
                                        ; konstantní výraz vypočítaný v rámci předběžného zpracování
```

- **Indexace**
  Proměnné v reálném čase mohou být indexovány proměnnými v reálném čase

Upozornění

Proměnné, které nejsou tvořeny v reálném čase, nesmí být indexovány proměnnými v reálném čase.

Příklad:

```
WHEN…DO $AC_PARAM[$AC_MARKER[1]] = 3
```

Nepřípustné je:

```
$AC_PARAM[1] = $P_EP[$AC_MARKER]
```

Příklad programování

Příklady výrazů v reálném čase

`ID=1 WHENEVER ($AA_IM[Y]>30) AND ($AA_IM[Y]<40) DO $AA_OVR[S1]=80`	Volba okna pro pozici
`ID=67 DO $A_OUT[1]=$A_IN[2] XOR $AN_MARKER[1]`	Vyhodnocení dvou boolovských signálů
`ID=89 DO $A_OUT[4]=$A_IN[1] OR ($AA_IM[Y]>10)`	Výstup výsledků porovnávání

© Siemens AG 2004, Všechna práva vyhrazena

10.3 Speciální proměnné v reálném čase pro synchronní akce

V synchronních akcích jsou přípustné proměnné v reálném čase uvedené v následujících odstavcích.

10.3.1 Ukazatel/čítač $AC_MARKER[n]



Funkce

Proměnné značek mohou být v synchronních akcích jak čteny, tak i zapisovány.

Kanálová značka/čítač
$AC_MARKER[n]
Datový typ: INTEGER
Proměnná značky pro specifický kanál je pod stejným názvem k dispozici pro každý kanál jen jednou.

Příklad:

```
WHEN ... DO $AC_MARKER[0] = 2
WHEN ... DO $AC_MARKER[0] = 3
WHEN $AC_MARKER == 3 DO $AC_OVR=50
```

10.3.2 Proměnná časovače $AC_TIMER[n], od SW 4



Funkce

(není k dispozici u 840D NCU 571, FM-NC)

Systémová proměnná $AC_TIMER[n] umožňuje spouštění akcí po uplynutí definované doby prodlevy.
Datový typ: REAL
Jednotky: sekundy
n: číslo proměnné časovače

- **Nastavení časovače**
  Načítání proměnné časovače se spouští přiřazením hodnoty $AC_TIMER[n] = hodnota
  n: číslo časové proměnné
  hodnota: počáteční hodnota

© Siemens AG 2004, Všechna práva vyhrazena

- **Zastavení časovače**
  Načítání proměnné časovače se zastaví přiřazením záporné hodnoty $AC_TIMER[n]=-1
- **Čtení časovače**
  Aktuální časová hodnota může být načtena jak při běžící, tak i při zastavené proměnné časovače. Po zastavení proměnné časovače přiřazením –1 zůstane poslední časový údaj zachován a může být dále načítán.

Příklad:
Výstup skutečné hodnoty přes analogový výstup 500 ms po rozpoznání stavu digitálního vstupu.

```
WHEN $A_IN[1]==1 DO $AC_TIMER[1]=0    ; vynulování a spuštění časovače
WHEN $AC_TIMER[1]>=0.5 DO $A_OUTA[3]=$AA_IM[X] $AC_TIMER[1]=-1
```

10.3.3 Parametr synchronní akce $AC_PARAM[n]



Funkce

Datový typ: REAL
n: číslo parametru 0 – n
Parametr synchronní akce $AC_PARAM[n] slouží při výpočtech a jako pomocná paměť v synchronních akcích.
Počet proměnných AC parametrů, které jsou k dispozici v jednom kanálu, je definován strojním parametrem MD 28254: MM_NUM_AC_PARAM.
Parametry jsou pod jedním názvem v každém kanálu k dispozici jen jednou. Proměnná $AC_PARAM se ukládá do dynamické paměti.

© Siemens AG 2004, Všechna práva vyhrazena

10.3.4 Přístup k početním parametrům $Rxx



Funkce

Datový typ: REAL
Tyto statické proměnné slouží mimo jiné při výpočtech ve výrobním programu. V aktu IPO se k nim lze dostat připojením znaku **$**.
Příklady

```
WHEN $AA_IM[X]>=40.5 DO $R10=$AA_MM[Y] ; přístup kvůli čtení do R-parametru 10
WHEN $AA_IM[X]>=6.7 DO                 ; přístup kvůli čtení do R-parametru, jehož
$R[$AC_MARKER[1]]=30.6                 číslo je uvedeno v proměnné Marker 1
```



Upozornění

Použití
Použití R-parametrů v synchronních akcích umožňuje:

- Ukládání hodnot, které si přejete zachovat i přes konec programu, reset NC a Power On.
- Vypisování uložených hodnot na obrazovce R-parametrů.
- Archivace hodnot zjištěných při synchronní akci

R-parametry by se měly používat buď jako „normální početní parametry Rxx **nebo** jako proměnné v reálném čase $Rxx.
Pokud má být R-parametr po svém využití v synchronní akci znovu použit jako "normální“ početní proměnná, musí explicitní programování zastavení předběžného zpracování pomocí příkazu STOPRE zajistit, aby zpracování hlavního a pomocného programu byla synchronizována.
Příklad:

```
WHEN $AA_IM[X]>=40.5 DO $R10=$AA_MM[Y] ; Použití R10 v synchronní akci
G01 X500 Y70 F1000
STOPRE                                 ; zastavení předběžného zpracování
IF R10>20                              ; vyhodnocení početní proměnné
```


© Siemens AG 2004, Všechna práva vyhrazena

10.3.5 Načítání/zápis strojních/nastavovaných parametrů, od SW 4

Funkce

Od SW 4 je možné načítání a zápis strojních a nastavovaných parametrů (MD, SD) i ze synchronních akcí.

- **Načítání neměnných MD, SD**
  K těmto parametrům se ze synchronních akcí přistupuje stejně jako v normálních příkazech výrobního programu; jsou uvozeny znakem **$**.
  Příklad:

```
ID=2 WHENEVER $AA_IM[z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

; V tomto příkladu se požaduje reverzní pozice 2 pro oscilaci, která je považována za neměnnou.

- **Načítání proměnných MD, SD**
  K těmto parametrům je ze synchronních akcí přístup, jsou uvozeny znaky **$$** a jsou vyhodnocovány v interpolačním taktu.
  Příklad:

```
ID=1 WHENEVER $AA_IM[z]<$$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

; Zde se vychází z toho, že reverzní pozice může být příkazem v průběhu zpracování změněna.

- **Zápis MD, SD**
  Předpoklad:
  Právě platná přístupová oprávnění musí umožňovat zápis. Měnit MD a SD ze synchronní akce má smysl jen tehdy, pokud je změna **okamžitě** uplatněna. Platnost změn pro všechny MD a SD je uvedena v **literatuře:** /LIS/, Seznamy.
  Adresace:
  MD a SD, které chcete změnit, musí být uvozeny znaky **$$**.
  Příklad:

```
ID=1 WHEN $AA_IW[X]>10 DO $$SN_SW_CAM_PLUS_POS_TAB_1[0]=20
             $$SN_SW_CAM_MINUS_POS_TAB_1[0]=30
```

; Změna spínací pozice SW vačky. Upozornění: Spínací pozice musí být změněny 2-3 takty IPO před dosažením dané pozice.

© Siemens AG 2004, Všechna práva vyhrazena

10.3.6 Proměnné FIFO $AC_FIFO1[n] ... $AC_FIFO10[n], od SW 4

Funkce

Datový typ: REAL
Pro ukládání souvisejících datových posloupností je k dispozici 10 proměnných FIFO (posuvných registrů).
Použití:

- Cyklická měření
- Postupné obrábění

Ke každému prvku je přístup za účelem čtení a zápisu.
Počet hodnot, které mohou být zapsány do proměnné FIFO, je definován strojním parametrem MD 28260: NUM_AC_FIFO.
Počet hodnot, které mohou být zapsány do jedné proměnné FIFO, je stanoven strojním parametrem MD 28264: LEN_AC_FIFO. Všechny proměnné FIFO mají stejnou délku.
Součet všech prvků FIFO se bude počítat jen tehdy, je-li bit0 v MD 28266: MODE_AC_FIFO aktivován.

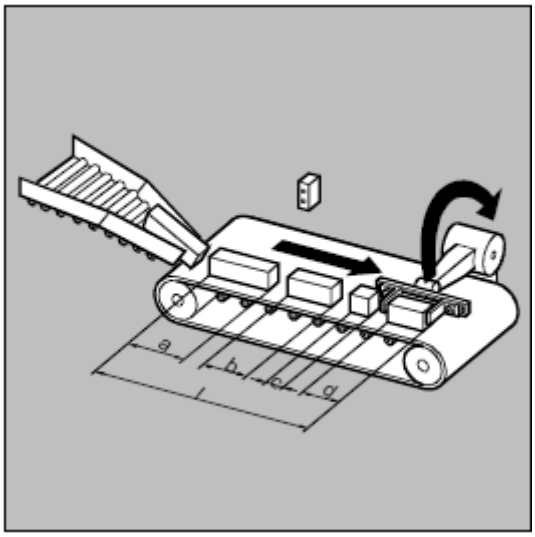
Indexy 0 až 5 mají zvláštní význam:

n=0: Při zápisu: Nová hodnota je uložena do FIFO.
Při čtení: Nejstarší prvek je načten a odstraněn z IFO.
n=1: Přístup k nejstaršímu uloženému prvku.
n=2: Přístup k nejmladšímu uloženému prvku.
n=3: Součet všech prvků FIFO.
n=4: Počet prvků, které jsou ve FIFO k dispozici.
Ke každému prvku lze mít přístup za účelem čtení nebo zápisu.
Vynulování proměnné FIFO se provádí vynulováním počtu prvků, např. pro první proměnnou FIFO: $AC_FIFO1[4]=0.
n=5: Aktuální index zápisu vzhledem k počátku FIFO.
n=6 až $6+n_{max}$:
Přístup k n-tému prvku FIFO.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Paměť typu posuvný registr
V rámci výrobního postupu se pro přepravu výrobků různých délek (a, b, c, d) používá pásový dopravník. Na tomto dopravníku s délkou „l“ jsou proto současně přepravovány různé počty výrobků v závislosti na jejich délkách. Při konstantní rychlosti dopravníku je proto nutné přizpůsobit odebírání výrobků z pásu proměnnému času jejich příchodu.



`DEF REAL INTV=2.5`	Konstantní vzdálenost mezi pokládanými výrobky
`DEF REAL TOTAL=270`	Vzdálenost mezi pozicí měření délky a pozicí odebírání
`EVERY $A_IN[1]==1 DO $AC_FIFO1[4]=0`	Vynulování FIFO na začátku procesu
`EVERY $A_IN[2]==1 DO $AC_TIMER[0]=0`	Pokud výrobek přeruší světelnou závoru, spustí se měření času
`EVERY $A_IN[2]==0 DO $AC_FIFO1[0]=$AC_TIMER[0]*$AA_VACTM[B]`	Jakmile se světelná závora uvolní, ze změřeného času a rychlosti dopravníku se vypočítá délka výrobku, která se pak uloží do FIFO.
`EVERY $AC_FIFO1[3]+$AC_FIFO1[4]*BETW>=TOTAL DO POS[Y]=-30` `$R1=$AC_FIFO1[0]`	Jakmile je součet všech délek produktů a vzdáleností mezi nimi větší nebo roven vzdálenosti mezi pozicí pro pokládání a odebírání, je výrobek na pozici pro odebrání odebrán z pásového dopravníku, příslušná délka výrobku se načte z FIFO.

© Siemens AG 2004, Všechna práva vyhrazena

10.3.7 Informace o typu bloku v interpolátoru (SW 6.4 a 7.1)

Pro synchronní akce jsou k dispozici následující systémové proměnné, abyste mohli získávat informace o bloku, který je právě zpracováván.

$AC_BLOCKTYPE
$AC_BLOCKTYPEINFO
$AC_SPLITBLOCK

$AC_BLOCKTYPE Hodnota:		**$AC_BLOCKTYPEINFO Hodnota:**				
0	jiná než 0					Význam
Původní blok	Pomocný blok	T	H	Z	E	Spouštění pro pomocný blok:
	1 (SW 6.4)	1	0	0	0	Interně generovaný blok, žádné další informace (od SW 7.1)
(od SW 7.1)		(od SW 7.1)				
	2	2	0	0	1	Faseta/zaoblení: přímka
	2	2	0	0	2	Faseta/zaoblení: kruh
	3	3	0	0	1	Zpětné najetí: najetí po přímce
	3	3	0	0	2	Zpětné najetí: najetí po čtvrtkruhu
	3	3	0	0	3	Zpětné najetí: najetí po půlkruhu
						Korekce nástroje:
	4	4	0	0	1	Blok najetí po STOPRE
	4	4	0	0	2	Spojovací bloky po nenalezení průsečíku
	4	4	0	0	3	Bodový kruh na vnitřních rozích (pouze v případě transformace TRACYL)
	4	4	0	0	4	Objížděcí kruhový oblouk (příp. kuželový řez) na vnějších rozích)
	4	4	0	0	5	Najížděcí bloky s potlačením korekce
	4	4	0	0	6	Najížděcí bloky při opětovné aktivaci korekce rádiusu nástroje
	4	4	0	0	7	Rozložení bloku kvůli příliš velkému zakřivení
	4	4	0	0	8	Vyrovnávací bloky při 3D čelním frézování (vektor nástroje II vektor plochy)
						Zaoblení prostřednictvím
	5	5	0	0	1	G641
	5	5	0	0	2	G642
	5	5	0	0	3	G643
	5	5	0	0	4	G644

© Siemens AG 2004, Všechna práva vyhrazena

$AC_BLOCKTYPE Hodnota:		$AC_BLOCKTYPEINFO Hodnota:				
0	jiná než 0					Význam
Původní blok	Pomocný blok	T	H	Z	E	Spouštění pro pomocný blok:
	6	6	0	0	1	Blok TLIFT s: lineární pohyb tangenciální osy a bez zdvihového pohybu
	6	6	0	0	2	nelineární pohyb tangenciální osy (polynom) a bez zdvihového pohybu
	6	6	0	0	3	zdvihový pohyb, pohyb tangenciální osy a zdvihový pohyb se spouští současně
	6	6	0	0	4	zdvihový pohyb, tangenciální osa se spouští, až když je dosaženo určité zdvihové pozice.
	7	7	0	0	1	Rozčlenění dráhy: Naprogramované rozčlenění dráhy bez lisování a prostřihování je aktivní
	7	7	0	0	2	Naprogramované rozčlenění dráhy s aktivním lisováním nebo prostřihováním
	7	7	0	0	3	Automaticky interně generované rozčlenění dráhy
	8	Identifikace aplikace				Překlad cyklu: Identifikace aplikace překladače cyklů, který vytvořil blok

Upozornění

$AC_BLOCKTYPEINFO obsahuje na místě tisíců (T) vždy hodnotu typu bloku pro případ, že pomocný blok existuje. ($AC_BLOCKTYPE se nerovná 0)

T místo tisíců
H místo stovek
Z místo desítek
E Místo jednotek
v $AC_BLOCKTYPEINFO

$AC_SPLITBLOCK Hodnota:	Význam
0	Nezměněný naprogramovaný blok (s blokem generovaným kompresorem se zachází stejně jako s naprogramovaným blokem).
1	Jedná se o interně generovaný blok nebo o zkrácený původní blok
3	Jedná se o poslední blok v řetězci interně generovaných bloků nebo zkrácených původních bloků

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Počítání bloků zaoblení

```
$AC_MARKER[0]=0
$AC_MARKER[1]=0
$AC_MARKER[2]=0
...
; Definice synchronních akcí, pomocí
; kterých jsou počítány bloky zaoblení.
; Všechny bloky zaoblení se počítají v $AC_MARKER[0].
ID = 1 WHENEVER ($AC_TIMEC ==0) AND ($AC_BLOCKTYPE==5) DO _
      $AC_MARKER[0]= $AC_MARKER[0] + 1
...
; Bloky zaoblení vytvořené pomocí G641 se počítají v $AC_MARKER[1]
ID = 2 WHENEVER ($AC_TIMEC ==0) AND ($AC_BLOCKTYPEINFO==5001) DO _
      $AC_MARKER[1]= $AC_MARKER[1] + 1
...
; Bloky zaoblení vytvořené pomocí G642 se počítají v $AC_MARKER[2]
ID = 3 WHENEVER ($AC_TIMEC ==0) AND ($AC_BLOCKTYPEINFO==5002) DO _
      $AC_MARKER[2]= $AC_MARKER[2] + 1
...
```


© Siemens AG 2004, Všechna práva vyhrazena

10.4 Operace v synchronních akcích

10.4.1 Výstup pomocných funkcí



Funkce

Pokud je podmínka splněna, je možné v každém bloku obráběcího postupu aktivovat maximálně 10 M-, H- a S-funkcí.
Výstup pomocných funkcí se aktivuje pomocí akčního slova „DO“. Výstup pomocných akcí se uskutečňuje okamžitě v interpolačním taktu. Okamžik výstupu pomocných funkcí definovaný strojním parametrem se nebere v úvahu.
Okamžik výstupu se určí, jakmile je podmínka splněna.

Příklad:
Zapnutí chladicí kapaliny při určité poloze osy:

```
WHEN $AA_IM[X]>=15 DO M07
POS[X]=20 FA[X]=250
```



Postup

Pomocné funkce smí být programovány pouze ve spojení s klíčovými slovy WHEN nebo EVERY v synchronních akcích s blokovou platností (bez modální identifikace). Zda je pomocná funkce aktivní nebo není, se určuje z PLC, např. přes NC-Start.



Upozornění

Z pohybových synchronních akcí není možné:

- M0, M1, M2, M17, M30: Zastavení/konec programu (M2, M17, M30 jsou možné u technologických cyklů)
- M70: Funkce vřetena
- M6, příp. M-funkce pro výměnu nástroje nastavené pomocí strojních parametrů
- M40, M41, M42, M43, M44, M45: Změna převodového stupně



Příklad programování

`WHEN $AA_IW[Q1]>5 DO M172 H510`	Pokud skutečná hodnota osy Q1 překročí 5 mm, z PLC aktivovat pomocné funkce M172 a H510.

© Siemens AG 2004, Všechna práva vyhrazena

10.4.2 Aktivování blokování načítání, RDISABLE



Funkce

Příkazem RDISABLE se při splněné podmínce pozastaví další příprava bloku v hlavním programu. Programovatelné synchronní pohybové akce budou dále zpracovávány, následující bloky budou dále připravovány.

Na začátku bloku s příkazem RDISABLE se vždy aktivuje přesné najetí, ať už je příkaz RDISABLE aktivní nebo není.



Příklad programování

Spuštění programu v interpolačním taktu v závislosti na externích vstupech.

...	
WHENEVER $A_INA[2]<7000 DO RDISABLE	Jestliže napětí na vstupu 2 poklesne pod 7 V, další zpracování programu bude pozastaveno (1000 = 1 V).
N10 G1 X10	Pokud je podmínka splněna, na konci bloku N10 je blokování načítání aktivní.
N10 G1 X10	
...	

© Siemens AG 2004, Všechna práva vyhrazena

10.4.3 Deaktivování zastavení předběžného zpracování, STOPREOF



Funkce

Při explicitně naprogramovaném zastavení předběžného zpracování příkazem STOPRE nebo v důsledku implicitního zastavení předběžného zpracování aktivní synchronní akcí lze pomocí příkazu STOPREOF toto zastavení odstranit po následujícím bloku programu, jakmile je podmínka splněna.

Upozornění
Příkaz STOPREOF musí být naprogramován s klíčovým slovem WHEN a s blokovou platností (bez identifikačního čísla).



Příklad programování

Rychlé větvení programu na konci bloku.

`WHEN $AC_DTEB<5 DO STOPREOF`	; Pokud vzdálenost ke konci bloku poklesne pod 5 mm, zastavení předběžného zpracování bude zrušeno.
`G01 X100`	; Po uskutečnění lineární interpolace bude zastavení předběžného zpracování odstraněno.
`IF $A_INA[7]>500 GOTOF MARKE1=X100`	; Pokud napětí na vstupu 7 poklesne pod 5 V, provede se skok na návěští 1.

© Siemens AG 2004, Všechna práva vyhrazena

10.4.4 Vymazání zbytkové dráhy

V závislosti na podmínce může být uskutečněno vymazání zbytkové dráhy pro dráhu a pro uvedenou osu.

Máte k dispozici tyto možnosti:

- Rychlé předem připravené vymazání zbytkové dráhy
- Vymazání zbytkové dráhy bez přípravy (od SW 4.3).

10.4.5 Vymazání zbytkové dráhy s přípravou, DELDTG, DELDTG (osa 1 až x)

Upozornění
Identifikátor osy uvedený v závorkách u příkazu DELDTG platí jen pro jednu polohovací osu.

Funkce

Předem připravené vymazání zbytkové dráhy pomocí příkazu DELDTG umožňuje velmi rychlou reakci na spouštěcí událost a bude se tedy používat v aplikacích kritických z časového hlediska, např.:

- Čas mezi vymazáním zbytkové dráhy a spuštěním dalšího bloku má být velmi krátký.
- Podmínka pro vymazání zbytkové dráhy je splněna s velmi vysokou pravděpodobností.

Postup

Na konci pohybového bloku, v němž bylo spuštěno předem připravené vymazání zbytkové dráhy, se aktivuje implicitní zastavení předběžného zpracování.
Režim řízení pohybu po dráze, příp. pohyby polohovací osy proto budou na konci bloku s rychlým vymazáním zbytkové dráhy přerušeny, příp. pozastaveny.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Rychlé vymazání zbytkové dráhy na dráze

```
WHEN $A_IN[1]==1 DO DELDTG
N100 G01 X100 Y100 F1000          ; Jakmile je vstup aktivován, pohyb se přeruší
N110 G01 X…
IF $AA_DELT>50…
```

Příklad programování

Rychlé vymazání zbytkové dráhy osy

Zastavení naprogramovaného polohovacího pohybu:

`ID=1 WHEN $A_IN[1]==1 DO MOV[V]=3 FA[V]=700`	Spuštění osy
`WHEN $A_IN[2]==1 DO DELDTG(V)`	Vymazání zbytkové dráhy, osa se zastaví pomocí MOV = 0

Vymazání zbytkové dráhy v závislosti na napětí vstupu

`WHEN $A_INA[5]>8000 DO DELDTG(X1)`	Jakmile napětí na vstupu 5 překročí 8 V, vymazat zbytkovou dráhu osy X1. Pohyb po dráze bude pokračovat.

```
POS[X1]=100 FA[X1]=10 G1 Z100 F1000
```

Omezení

Předem připravené vymazání zbytkové dráhy:

- nelze použít, když je aktivní korekce rádiusu nástroje.
- smí být naprogramováno pouze v synchronních akcích s blokovou platností (bez identifikačního čísla).

© Siemens AG 2004, Všechna práva vyhrazena

10.4.6 Definice polynomu, FCTDEF, bloková synchronizace



Programování

FCTDEF(č.polynomu,LLIMIT,ULIMIT,a_0,a_1,a_2,a_3)



Vysvětlení

č.polynomu	Číslo polynomu 3. stupně
LLIMIT	Dolní mez pro funkční hodnotu
ULIMIT	Horní mez pro funkční hodnotu
a_0,a_1,a_2,a_3	Koeficienty polynomu



Funkce

Příkazem FCTDEF můžete definovat polynomy 3. stupně ve formě $y = a_0 + a_1.x + a_2.x^2 + a_3.x^3$. Tyto polynomy jsou využívány on-line korekcí nástroje FTOC a vyhodnocovací funkcí SYNFCT pro výpočty funkčních hodnot z proměnných hlavního zpracování (proměnných v reálném čase).

Polynomy jsou definovány buď synchronně v bloku pomocí funkce FCTDEF nebo pomocí systémových proměnných:

$AC_FCTLL[n]	Dolní mez pro funkční hodnotu
$AC_FCTUL[n]	Horní mez pro funkční hodnotu
$AC_FCT0[n]	a_0
$AC_FCT1[n]	a_1
$AC_FCT2[n]	a_2
$AC_FCT3[n]	a_3
n	Číslo polynomu

© Siemens AG 2004, Všechna práva vyhrazena

Upozornění

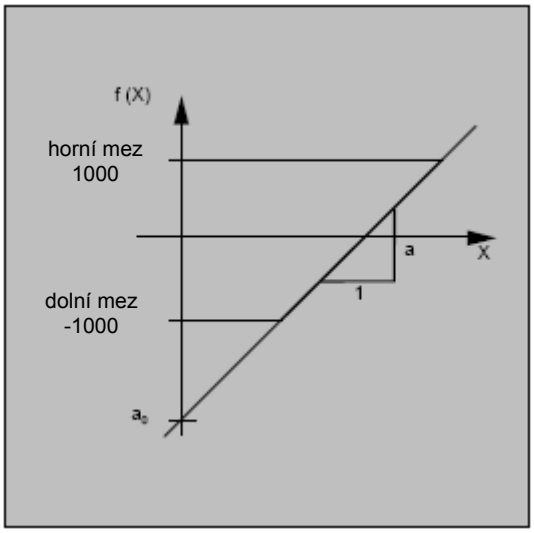
- Do systémových proměnných je možné zapisovat z výrobního programu nebo ze synchronní akce. Pří zápisu z výrobního programu je nutno se zadáním příkazu STOPRE postarat o blokovou synchronizaci zápisu.
- Od SW 4:
  Ze synchronních akcí je možné měnit systémové proměnné $AC_FCTLL[n], $AC_FCTUL[n] a $AC_FCFT0[n] až $AC_FCTn[n] (nikoli u systému SINUMERIK FM-NC, nikoli u systému SINUMERIK 840D s NCU 571).

Při zápisu ze synchronní akce jsou koeficienty polynomu a meze funkční hodnoty v platnosti okamžitě.

Příklad programování

Polynom pro přímkový úsek:

Při horní mezní hodnotě 1000, dolní mezní hodnotě –1000, s úsekem na ordinátě a_0=$AA_IM[X] a s lineárním gradientem přímky 1 bude definice polynomu znít:



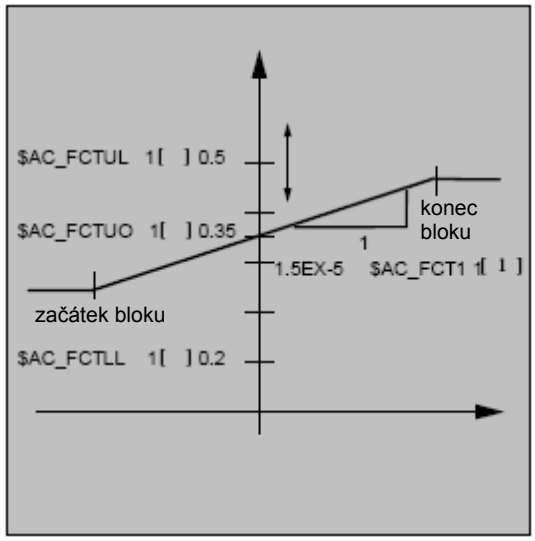

```
FCTDEF(1, -1000,1000,$AA_IM[X],1)
```

10.4.7 Řízení výkonu laseru

Příklad programování

Definice polynomu pomocí proměnných:

Jedno z možných využití definice polynomu je řízení výkonu laseru. Řízení výkonu laseru znamená:
Ovlivňování analogového výstupu v závislosti na např. rychlosti pohybu po dráze.



`$AC_FCTLL[1]=0.2`	Definice koeficientů polynomu
`$AC_FCTUL[1]=0.5`	
`$AC_FCT0[1]=0.35`	
`$AC_FCT1[1]=1.5EX-5`	
`STOPRE`	
`ID=1 DO $AC_FCTUL[1]=$A_INA[2]*0.1 +0.35`	On-line změna horní mezní hodnoty
`ID=2 DO SYNFCT(1,$A_OUTA[1],$AC_VACTW)`	
	V závislosti na rychlosti pohybu po dráze (uložené v $AC_VACTW) se řízení výkonu laseru uskutečňuje přes analogový výstup 1.

Upozornění
Dalším využitím výše uvedeného polynomu je popis funkce SYNFCT.

© Siemens AG 2004, Všechna práva vyhrazena

10.4.8 Vyhodnocovací funkce SYNFCT



Programování

```
SYNFCT (č.polynomu,proměnná v reálném čase-výstup, proměnná v reálném
čase-vstup)
```



Vysvětlení

`Č.polynomu`	Polynom definovaný pomocí příkazu FCTDEF (viz kapitola „Definice polynomu)
`proměnná v reálném čase-výstup`	Zápis do proměnné v reálném čase
`proměnná v reálném čase-vstup`	Načtení proměnné v reálném čase



Funkce

Funkce SYNFCT načítá synchronní proměnnou v reálném čase (např. analogový vstup, skutečnou hodnotu, ...) a na jejím základě vypočítává funkční hodnotu (např. override, rychlost, polohu osy,…), přičemž se využívá vyhodnocovací polynom (FCTDEF) max. 3. stupně. Výsledek je k dispozici jako proměnná v reálném čase a je shora a zdola ohraničen příkazem FCTDEF (viz kap. 10.4.7).

Jako výstupní proměnné v reálném čase mohou být zvoleny proměnné, které do operace zpracování vstupují:

- s aditivním ovlivňováním
- s multiplikativním ovlivňováním
- jako offset pozice
- přímo



Použití

Vyhodnocovací funkce nachází uplatnění:

- při AC regulaci (Adaptive Control)
- při regulaci výkonu laseru
- v polohové regulaci

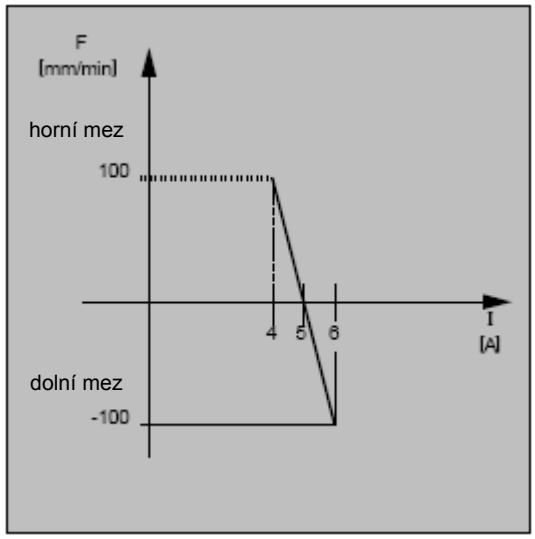
© Siemens AG 2004, Všechna práva vyhrazena

10.4.9 AC regulace (aditivní)

Příklad programování

Aditivní ovlivňování naprogramovaného posuvu

Naprogramovaný posuv má být regulován aditivně pomocí proudu pro osu X (osa přísuvu): Posuv se má měnit v rozsahu ± 100 mm/min, přičemž proud se mění o ± 1 A okolo pracovního bodu 5 A.



1. Definice polynomu
   Stanovení koeficientů
   $y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$
   a_1 = -100 mm / 1 min A
   a_0 = -(-100)*5 = 500
   $a_2 = a_3 = 0$ (žádný kvadratický a kubický člen
   horní mezní hodnota = 100
   dolní mezní hodnota = -100

   z toho plyne:

```
FCTDEF(1,-100,100,500,-100,0,0)
```

2. Aktivování AC regulace

```
ID=1 DO SYNFCT(1,$AC_VC,$AA_LOAD[x])
```

Přes proměnnou $AA_LOAD[x] se načítá aktuální zatížení osy (% max. proudu pohonu), korekce posuvu po dráze se vypočítá pomocí výše uvedeného polynomu.

© Siemens AG 2004, Všechna práva vyhrazena

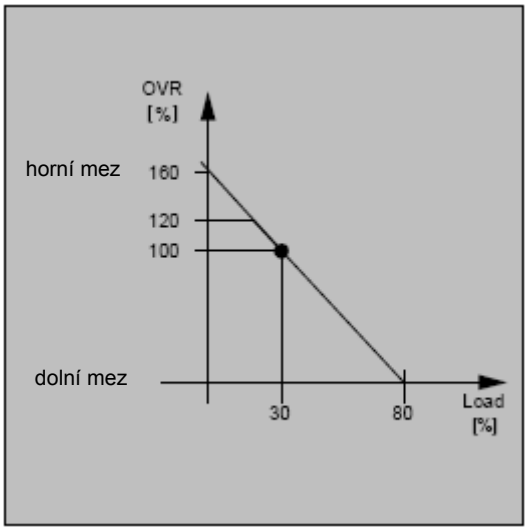
10.4.10 AC regulace (multiplikativní)

Příklad programování

Multiplikativní ovlivňování naprogramovaného posuvu

Naprogramovaný posuv má být regulován multiplikativně, přičemž posuv – v závislosti na zatížení pohonu – nesmí překročit určité hranice:

- Při zatížení pohonu na 80% se má posuv zastavit: Override = 0.
- Při zatížení pohonu 30% se smí nástroj pohybovat s naprogramovaným posuvem: Override = 100
- Rychlost posuvu smí být překročena o maximálně 20%: max. override = 120%.



1. Definice polynomu
   Stanovení koeficientů
   $y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$
   $a_1 = -100\% / (80 – 30)\% = -2$
   $a_0 = 100 + (2 * 30) = 160$
   $a_2 = a_3 = 0$ (žádný kvadratický a kubický člen
   horní mezní hodnota = 120
   dolní mezní hodnota = 0

   z toho plyne:

```
FCTDEF(2,0,120,160,-2,0,0)
```

2. Aktivování AC regulace

```
ID=1 DO SYNFCT(2,$AC_OVR,$AA_LOAD[x])
```

Přes proměnnou $AA_LOAD[x] se načítá aktuální zatížení osy (% max. proudu pohonu), override posuvu se vypočítá pomocí výše uvedeného polynomu.

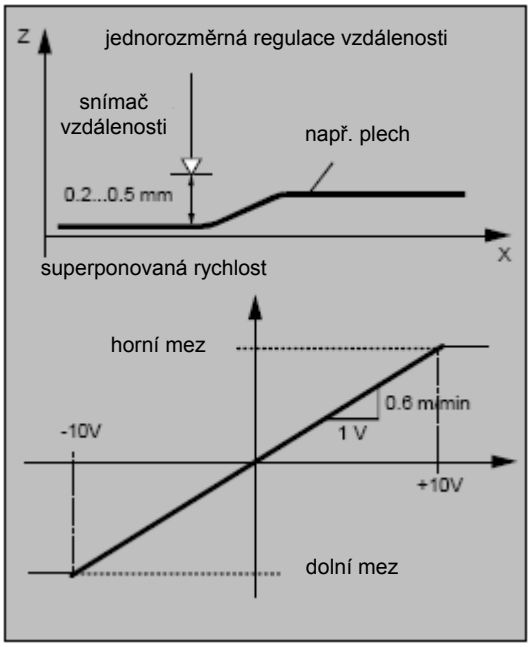
© Siemens AG 2004, Všechna práva vyhrazena

10.4.11 Regulace vzdálenosti s omezenou korekcí

Příklad programování

Integrovaný výpočet hodnoty vzdálenosti s kontrolou mezních oblastí.
$AA_OFF_MODE = 1
Pozor:
Zesílení superponované regulační smyčky je závislé na nastavení interpolačního taktu. Pomoc: Načtení a započítání MD pro interpolační takt.
Upozornění:
Mezní hodnota rychlosti pro superponované interpolátory pomocí MD 32020: JOG_VELO při taktu IPO 12 ms:
Rychlost:

$$\frac{0.120\,\text{ms}}{0.6\,\text{ms}} / \text{mV} = 0.6 \frac{\text{m}}{\text{min}} / \text{V}$$



Podprogram: Zapnutí regulace vzdálenosti

%_N_AON_SPF	Zapnutí podprogramu pro regulaci vzdálenosti
PROC AON	
$AA_OFF_LIMIT[Z]=1	Definice mezní hodnoty
FCTDEF(1, -10, +10, 0, 0.6, 0.12)	Definice polynomu
ID=1 DO SYNFCT(1,$AA_OFF[Z],$A_INA[3])	Aktivování regulace vzdálenosti
ID=2 WHENEVER $AA_OFF_LIMIT[Z]<>0 DO $AA_OVR[X] = 0	Při překročení mezí oblasti zablokování osy X
RET	
ENDPROC	

Podprogram: Vypnutí regulace vzdálenosti

%_N_AOFF_SPF	
PROC AOFF	Vypnutí podprogramu pro regulaci vzdálenosti
CANCEL(1)	Vymazání synchronní akce regulace vzdálenosti
CANCEL(2)	Vymazání kontroly mezních oblastí
RET	
ENDPROC	

Hlavní program

%_N_MAIN_MPF	
AON	Aktivování regulace vzdálenosti
...	
G1 X100 F1000	
AOFF	Deaktivování regulace vzdálenosti
M30	

© Siemens AG 2004, Všechna práva vyhrazena

Upozornění

Offset pozice v základním souřadném systému

Pomocí systémové proměnné $AA_OFF[osa] může být každé ose v kanálu superponován pohyb. Tento pohyb se chová jako offset pozice v základním souřadném systému.

Takto naprogramovaný offset pozice je okamžitě superponován na příslušnou osu nezávisle na tom, zda je či není pro osu naprogramován pohyb. Od SW 4 je možné omezit absolutní hodnotu, která má být korigována (proměnná v reálném čase-výstup), na hodnotu uloženou v nastavovaném parametru SD 43350: AA_OFF_LIMIT.

Pomocí strojního parametru MD 36750: AA_OFF_MODE se definuje způsob superpozice vzdálenosti:

0 proporcionální vyhodnocování
1 integrační vyhodnocování

Pomocí systémové proměnné $AA_OFF_LIMIT [osa] je možné v závislosti na směru zjistit, zda se hodnota korekce nalézá v rámci mezních hodnot. Tyto systémové proměnné mohou být kontrolovány ze synchronních akcí a při dosažení mezní hodnoty lze např. osu zastavit nebo aktivovat alarm.

0 Hodnota korekce není v rámci mezních hodnot
1 Dosaženo mezní hodnoty korekce v kladném směru
-1 Dosaženo mezní hodnoty korekce v záporném směru

© Siemens AG 2004, Všechna práva vyhrazena

10.4.12 On-line korekce nástroje, FTOC



Programování

```
FTOC(Č.polynomu, EV, Délka1_2_3 nebo Rádius4, Kanál, Vřeteno)
```



Vysvětlení příkazu

`Č.polynomu`	Polynom definovaný příkazem FCTDEF, viz kapitola „Definice polynomu“.
`EV`	Proměnná v reálném čase, pro kterou má být vypočítána funkční hodnota pomocí uvedeného polynomu.
`Délka1_2_3` `Rádius4`	Korekce délky ($TC_DP1 až 3), příp. korekce rádiusu, k níž má být funkční hodnota přičtena.
`Kanál`	Číslo kanálu, v němž se korekce uplatňuje. V případě korekce v aktivním kanálu se neudává. V cílovém kanálu musí být aktivován příkaz FTOCON.
`Vřeteno`	Udává se jen tehdy, pokud se má korekce týkat jiného než aktivního vřetena.



Funkce

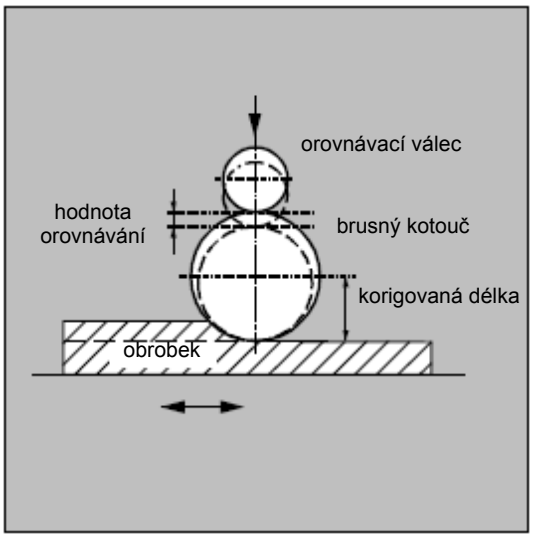
Příkaz FTOC umožňuje superponovaný pohyb geometrické osy podle polynomu naprogramovaného pomocí FCTDEF v závislosti na vztažné hodnotě, kterou může být např. skutečná hodnota osy. Tak můžete naprogramovat také modální on-line korekce nástroje nebo regulace vzdálenosti jako synchronní akce.

Použití
Opracování obrobků a orovnávání brusného kotouče v tomtéž nebo v jiném kanálu (kanál obrábění a orovnávání).
Okrajové podmínky pro orovnávání brusného kotouče platí pro příkaz FTOC analogicky k on-line korekci nástroje s příkazem PUTFTOCF.
Příslušné informace naleznete v kapitole 5 „Korekce nástroje“.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

V příkladu má být prováděna korekce délky aktivního brusného kotouče nacházejícího se v záběru.



`%_N_ABRICHT_MPF`	
`FCTDEF(1,-1000,1000,-$AA_IW[V],1)`	Definice funkce
`ID=1 DO` **`FTOC`**`(1,$AA_IW[V],3,1)`	Aktivování on-line korekce nástroje: Skutečná hodnota osy V je vstupní hodnotou pro polynom 1; výsledek se přičítá v kanálu 1 jako hodnota korekce k délce 3 aktivního brusného kotouče.
`WAITM(1,1,2)`	Synchronní akce v kanálu obrábění
`G1 V-0.05 F0.01 G91`	Přísuvný pohyb kvůli orovnávání
`G1 V-0.05 F0.02`	
`...`	
`CANCEL(1)`	Deaktivování on-line korekce
`...`	

© Siemens AG 2004, Všechna práva vyhrazena

10.4.13 Pohyby při najíždění do polohy

Funkce

Polohování os může probíhat ze synchronních akcí úplně asynchronně vůči výrobnímu programu. Programování polohovacích os ze synchronních akcí se doporučuje pro cyklické postupy nebo operace, jejichž řízení je silně závislé na událostech. Osy programované ze synchronních akcí se nazývají **příkazové osy**.

Od SW 5 mohou být v synchronních akcích programovány příkazy G70/G71/G700/G710. Díky tomu lze v synchronních akcích definovat systém jednotek pro operace polohování.

Literatura:
/PG/, kapitola 3, „Zadávání dráhy“
/FBSY/, „Spouštění příkazových os“

Systém jednotek je definován prostřednictvím příkazů G70/G71/G700/G710.

Naprogramováním G-funkcí v synchronní akci může být vyhodnocování palce/metrické jednotky pro synchronní akci definováno nezávisle na kontextu výrobního programu.

Příklad 1: Programové okolí ovlivňuje dráhu polohovací osy (žádná G-funkce v příkazové části synchronní akce).

`N100 R1=0`		
`N110 G0 X0 Z0`		
`N120 WAITP(X)`		
`N130 ID=1 WHENEVER $R==1 DO POS[X]=10`		
`N140 R1=1`		
`N150 G71 Z10 F10`	Z=10 mm	X=10 mm
`N160 G70 Z10 F10`	Z=254 mm	X=254 mm
`N170 G71 Z10 F10`	Z=10 mm	X=10 mm
`N180 M30`		

© Siemens AG 2004, Všechna práva vyhrazena

Příklad 2: Příkaz G71 v příkazové části synchronní akce jednoznačně určuje dráhu polohovací osy (metrické jednotky) nezávisle na programovém okolí.

N100 R1=0		
N110 G0 X0 Z0		
N120 WAITP(X)		
N130 ID=1 WHENEVER $R==1 DO G71 POS[X]=10		
N140 R1=1		
N150 G71 Z10 F10	Z=10 mm	X=10 mm
N160 G70 Z10 F10	Z=254 mm	X=10 mm (X se vždy nastavuje na polohu 10 mm)
N170 G71 Z10 F10	Z=10 mm	X=10 mm
N180 M30		

Příklad programování

Zablokování naprogramovaného pohybu osy

Pokud pohyb osy nemá začínat na začátku bloku, je možné ze synchronní akce až do požadovaného okamžiku pozastavit pohyb korekcí (override) nastavenou na nulu.

```
WHENEVER    $A_IN[1]==0 DO $AA_OVR[W]=0
            G01 X10 Y25 F750 POS[W]=1500
            FA=1000
            ; Polohovací osa bude pozastavena tak dlouho, dokud se digitální vstup 1 = 0.
```

© Siemens AG 2004, Všechna práva vyhrazena

10.4.14 Polohování osy, POS



Funkce

```
POS[Osa]=Hodnota
```

Pohyb polohovací osy nemá oproti programování z výrobního programu žádný vliv na zpracování tohoto výrobního programu.



Vysvětlení

`Osa:`	Název osy, která se má pohybovat
`Hodnota:`	Zadání hodnoty posuvu (v závislosti na režimu posuvu)



Příklad programování

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100
```

; Osa U uskuteční posuv v závislosti na režimu posuvu inkrementálně o 100 (palců/mm), příp. na pozici 100 (palce/mm) od počátečního bodu řídícího systému

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=$AA_MW[V]-$AA_IM[W]+13.5
```

; Osa U se posune o dráhu vypočítanou z proměnné v reálném čase.

10.4.15 Spuštění/zastavení osy, MOV



Programování

```
MOV [Osa] = Hodnota
```



Vysvětlení

`Osa`	Název osy, která má být spuštěna
`Hodnota`	Spouštěcí příkaz pro posuv/zastavení pohybu. Znaménko určuje směr pohybu Datový typ hodnoty je INTEGER.
`Hodnota > 0` (obvykle +1)	Kladný směr
`Hodnota < 0` (obvykle -1)	Záporný směr
`Hodnota == 0`	Zastavení pohybu osy

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Pomocí příkazu MOV[Osa]=Hodnota je možné spustit příkazovou osu bez udání koncového bodu. Příslušná osa se bude pohybovat v naprogramovaném směru, dokud není pomocí nového příkazu polohování nebo pohybu zadán jiný pohyb nebo dokud není osa příslušným příkazem zastavena.

Příklad programování

`... DO MOV[U]=0`	Zastavení osy U.

Upozornění
Pokud je dělicí osa zastavena příkazem MOV [Osa] = 0, osa se zastaví na nejbližší pozici dělení.

10.4.16 Posuv osy, FA

Příklad programování

FA[Osa] = Posuv

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=990
```
; Pevné zadání hodnoty posuvu

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=$AA_VACTM[W]+100
```
; Vytvoření hodnoty posuvu z proměnné v reálném čase

10.4.17 Softwarový koncový spínač

Funkce

Ohraničení pracovního pole naprogramované pomocí příkazů G25/G26 se bude pro příkazové osy zohledňovat v závislosti na nastavovaném parametru $SA_WORKAREA_PLUS_ENABLE. Aktivování a deaktivování ohraničení pracovního pole pomocí funkcí WALINON/WALIMOF příkazové osy neovlivňuje.

© Siemens AG 2004, Všechna práva vyhrazena

10.4.18 Koordinace os

Funkce

Za normálních okolností se osa pohybuje buď v důsledku pohybového bloku ve výrobním programu nebo jako polohovací osa díky synchronní akci. Pokud se má tatáž osa pohybovat střídavě z výrobního programu jako dráhová nebo polohovací osa a ze synchronní akce, je potřeba zajistit koordinované předávání mezi oběma pohyby osy. Jestliže se má příkazová osa následně pohybovat pomocí výrobního programu, je nezbytné zajistit změnu organizace předběžné přípravy, které si však vyžádá přerušení zpracování výrobního programu, které je srovnatelné se zastavením předběžného zpracování.

Příklad programování

Je možno zvolit, zda se má osa X pohybovat z výrobního programu nebo ze synchronní akce:

`N10 G01 X100 Y200 F1000`	Osa X programována ve výrobním programu
`...`	
`N20 ID=1 WHEN $A_IN[1]==1 DO POS[X]=150 FA[X]=200`	Spuštění polohování ze synchronní akce, když se aktivuje digitální vstup.
`...`	
`CANCEL(1)`	Deaktivování synchronní akce
`...`	
`N100 G01 X240 Y200 F1000`	
; X je dráhová osa; před spuštěním pohybu se vyskytuje doba prodlevy způsobovaná předáváním osy, pokud byl digitální vstup na 1 a polohování osy X prováděla synchronní akce.	

Příklad programování

Změna příkazu pohybu pro tutéž osu:

```
ID=1 EVERY $A_IN[1]>=1 DO POS[V]=100 FA[V]=560
                                ; spuštění polohování ze synchronní akce, je-li digitální vstup >= 1.
ID=2 EVERY $A_IN[2]>=1 DO POS[V]=$AA_IM[V] FA[V]=790
                                ; osa je v režimu sledování, 2. vstup se aktivuje,
                                tzn. koncová pozice a posuv pro osu V jsou soustavně sledovány,
                                když jsou obě synchronní akce současně aktivní.
```


© Siemens AG 2004, Všechna práva vyhrazena

10.4.19 Dosazení skutečné hodnoty



Funkce

Při provádění příkazu PRESETON (osa, hodnota) se aktuální poloha osy nemění, je jí však přiřazena nová hodnota.



Upozornění

Funkce PRESETON je ze synchronních akcí možná pro:

- Kruhové modulo osy, které se mají spouštět z výrobního programu
- Všechny příkazové osy, které se mají spouštět ze synchronní akce.

Omezení:
Funkci PRESETON není možné použít pro osy, které se podílejí na transformaci.



Příklad programování

```
WHEN $AA_IM[a] >= 89.5 DO PRESETON(a4,10.5)
```

; Posunutí počátečního bodu řídícího systému osy a o 10.5 délkových jednotek (palce nebo mm) v kladném směru osy.



Omezení

Jednou a toutéž osou je možné pohybovat z výrobního programu i ze synchronní akce, jen v různých časech. Z tohoto důvodu se mohou objevit v programování osy z výrobního programu zpoždění, pokud tato osa byla dříve naprogramována v synchronní akci.
Pokud je tatáž osa používána střídavě, je přepínání mezi oběma pohyby osy koordinováno.
Zpracování výrobního programu musí být za tímto účelem přerušeno.

© Siemens AG 2004, Všechna práva vyhrazena

10.4.20 Pohyby vřetena



Funkce

Polohování vřeten se může uskutečňovat pomocí synchronních akcí zcela asynchronně vůči výrobnímu programu. Tento druh programování se doporučuje pro cyklické procesy, jejichž řízení je silně závislé na událostech.



Příklad programování

Spuštění/zastavení/polohování vřetena

`ID=1 EVERY $A_IN[1]==1 DO` **`M3 S1000`**	Nastavení směru otáčení a otáček
`ID=2 EVERY $A_IN[2]==1 DO` **`SPOS=270`**	Polohování vřetena



Postup

Jestliže jsou v důsledku současně aktivních synchronních akcí pro vřeteno zadány příkazy, které si odporují, platí časově poslední příkaz vřetena.



Příklad programování

Nastavení směru otáčení a otáček vřetena / Polohování vřetena

`ID=1 EVERY $A_IN[1]==1 DO M3 S300`	Nastavení směru otáčení a otáček vřetena
`ID=2 EVERY $A_IN[2]==1 DO M4 S500`	Zadání nového směru otáčení a otáček
`ID=3 EVERY $A_IN[3]==1 DO S1000`	Zadání nových otáček
`ID=4 EVERY ($A_IN[4]==1) AND ($A_IN[1]==0) DO SPOS=0`	Polohování vřetena

© Siemens AG 2004, Všechna práva vyhrazena

10.4.21 Vlečení: TRAILON, TRAILOF

Funkce

`DO TRAILON(vlečná osa, řídící osa, faktor spojení)`	Aktivování vlečení
`DO TRAILOF(vlečná osa, řídící osa, faktor spojení)`	Deaktivování vlečení

Při aktivování spojení ze synchronní akce může být řídící osa v pohybu. Vlečná osa bude v tomto případě urychlena na požadovanou rychlost. Poloha řídící osy vzhledem k okamžiku synchronizace rychlosti je výchozí pozicí pro vlečení. Funkce vlečení je popsána v kapitole „Chování při pohybu po dráze".

Aktivování asynchronního vlečení:

`... DO` **`TRAILON`**`(FA, LA, Kf)` kde FA: vlečná osa
LA: řídící osa
Kf: faktor spojení

Deaktivování asynchronního vlečení:

`... DO` **`TRAILOF`**`(FA, LA, LA2)` kde FA: vlečná osa
LA: řídící osa
LA2: řídící osa 2 (není nutná)

Příklad programování

`$A_IN[1]==0 DO TRAILON(Y,V,1)`	Aktivování 1. svazku vlečení, když je nastaven digitální vstup 1
`$A_IN[2]==0 DO TRAILON(Z,W,-1)`	Aktivování 2. svazku vlečení
`G0 Z10`	Přísuv osy Z a W v opačných směrech
`G0 Y20`	Přísuv osy Y a V ve stejných směrech os
`...`	
`G1 Y22 V25`	Superpozice závislého a nezávislého pohybu vlečné osy „V“
`...`	
`TRAILOF(Y,V)`	Deaktivování 1. svazku vlečení
`TRAILOF(Z,W)`	Deaktivování 2. svazku vlečení

© Siemens AG 2004, Všechna práva vyhrazena

Aby se osa začleněná do svazku opět uvolnila pro přístup jako ke kanálové ose, je nutné napřed vyvolat funkci TRAILOF. Je nutné zabezpečit, aby byl příkaz TRAILOF uskutečněn, než si kanál příslušnou osu vyžádá. Tak tomu není v následujícím příkladu:

```
...
N50 WHEN TRUE DO TRAILOF(Y,X)
N60 Y100
...
```

V tomto případě osa není v pravý čas uvolněna, protože synchronní akce s blokovou platností s příkazem TRAILOF bude aktivována synchronně s 60 (viz 10.1).
Abyste předešli konfliktním situacím, mělo by se najíždět následujícím způsobem:

```
...
N50 WHEN TRUE DO TRAILOF (X,Y)
N55 WAITP(Y)
N60 Y100
```

10.4.22 Spojení řídící hodnotou, LEADON, LEADOF

Funkce

Spojení os řídící hodnotou může být v synchronních akcích programováno bez omezení.

Aktivování spojení os řídící hodnotou:

```
... DO LEADON(FA, LA, NR)
```

kde
FA: vlečná osa
LA: řídící osa
NR: číslo uložené tabulky křivky

Deaktivování spojení os řídící hodnotou:

```
... DO LEADOF(FA, LA)
```

kde
FA: vlečná osa
LA: řídící osa

Pokud potřebujete osu, která je začleněna do tohoto spojení uvolnit kvůli přístupu synchronní akce, je nutno pro připojenou osu napřed vyvolat funkci RELEASE.

Příklad:

```
RELEASE (XKAN)
ID=1 every SR1==1 to LEADON(CACH,XKAN,1)
```


© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Plovoucí oddělování

Tyčový materiál, který se soustavně pohybuje pracovním prostorem oddělovacího zařízení, má být dělen na stejně dlouhé kusy.

Osa X: Osa, v níž se pohybuje tyčový materiál. Souřadný systém obrobku.

Osa X1: Osa stroje tyčového materiálu. Souřadný systém stroje.

Osa Y: Osa, ve které je oddělovací zařízení „vezeno“ s tyčovým materiálem.

Předpokládá se, že přísuv oddělovacího zařízení a jeho řízení zabezpečuje PLC. Pro zajištění synchronizace mezi tyčovým materiálem a oddělovacím zařízením mohou být vyhodnocovány signály rozhraní PLC.

Akce: Aktivování vazby, LEADON
Deaktivování vazby, LEADOF
Dosazování skutečné hodnoty, PRESETON

```
%_N_SCHERE1_MPF
;$PATH=/_N_WKS_DIR/_N_DEMOFBE_WPD
N100 R3=1500                                    ; délka oddělovaného kusu
N200 R2=100000 R13=R2/300
N300 R4=100000
N400 R6=30                                      ; počáteční pozice osy Y
N500 R1=1                                       ; počáteční podmínka pro dráhovou osu
N600 LEADOF(Y,X)                                ; vymazání příp. existující vazby
N700 CTABDEF(Y,X,1,0)                           ; definice tabulky
N800 X=30 Y=30                                  ; dvojice hodnot
N900 X=R13 Y=R13
N1000 X=2*R13 Y=30
N1100 CTABEND                                   ; konec definice tabulky
N1200 PRESETON(X1,0)                            ; PRESET na začátku
N1300 Y=R6 G0                                   ; počáteční pozice osy Y, osa je lineární
N1400 ID=1 WHENEVER $AA_IW[X]>$R3 DO PRESETON(X1,0)
                                                ; PRESET po délce R3, nový začátek po oddělení
N1500 RELEASE(Y)
N1800 ID=6 EVERY $AA_IM[X]<10 DO LEADON(Y,X,1)
                                                ; Y pomocí tabulky ve vazbě na X při X < 10
N1900 ID=10 EVERY $AA_IM[X]>$R3-30 DO EADOF(Y,X)
                                                ; > 30 před ujetím oddělované vzdálenosti, zrušení vazby
N2000 WAITP(X)
N2100 ID=7 WHEN $R1==1 DO MOV[X]=1              ; nastavení osy materiálu do soustavného
FA[X]=$R4                                         pohybu
N2200 M30
```

© Siemens AG 2004, Všechna práva vyhrazena

10.4.23 Měření

Funkce

Ve srovnání s použitím v pohybových blocích výrobního programu mohou být měřicí funkce ze synchronních akcí libovolně aktivovány a deaktivovány.

- Měření v ose bez vymazání zbytkové dráhy:
  MEAWA[Osa]=[Režim, Spouštěcí událost_1,..._4]

Kontinuální měření bez vymazání zbytkové dráhy:
MEAC[Osa]=[Režim, Měřicí paměť, Spouštěcí událost_1,..._4]

Další informace k měření: Viz kapitola 5: „Rozšířené měřicí funkce“.

10.4.24 Stanovení/vymazání značky pro čekání, SETM, CLEARM (od SW 5.2)

Funkce

`SETM(Číslo značky)`	Nastavení čekací značky pro kanál
`CLEARM (Číslo značky)`	Vymazání čekací značky pro kanál

V synchronních akcích mohou být nastavovány, příp. mazány čekací značky, aby bylo možné zajistit např. vzájemnou koordinaci kanálů.

SETM
Příkaz SETM může být zapsán ve výrobním programu nebo v příkazové části synchronní akce. Definuje značku s daným číslem pro kanál, v němž je příkaz zpracováván.

CLEARM
Příkaz CLEARM může být zapsán ve výrobním programu nebo v příkazové části synchronní akce. Vymaže značku s daným číslem pro kanál, v němž je příkaz zpracováván.

© Siemens AG 2004, Všechna práva vyhrazena

10.4.25 Reakce v případě chyby



Funkce

V rámci synchronní akce mohou být naprogramovány reakce na chybu, které monitorují stavové proměnné a spouštějí odpovídající reakce.

Možné reakce na chybové stavy jsou:

- Zastavení osy: Override = 0
- Aktivování alarmu: Pomocí příkazu SETAL mohou být ze synchronních akcí spouštěny alarmové cykly.
- Nastavení výstupu
- Veškeré operace, které jsou v synchronních akcích možné.



Příklad programování

```
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO $AA_OVR[X2]=0
                    ; Pozastavení osy X2, pokud je vzdálenost mezi osami X1 a X2 příliš malá.
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO SETAL(61000)
          ; Aktivování alarmu, pokud je bezpečnostní vzdálenost mezi osami X1 a X2 příliš malá.
```

10.4.26 Najíždění na pevný doraz, FXS a FOCON/FOCOF



Vysvětlení

`FXS[Osa]`	Aktivování jen v systémech s digitálními pohony (VSA, HSA, HLA).
`FXST[Osa]`	Změna momentu pro uchopení FXST
`FXSW[Osa]`	Změna monitorovacího okna FXSW
`FOCON[Osa]`	Aktivování omezení momentu/síly s modální platností
`FOCOF[Osa]`	Deaktivování omezení momentu/síly s modální platností
`FOCON/FOCOF`	Programování osy se uskutečňuje v hranatých závorkách. Přípustné jsou: • Geometrické osy – identifikátor • Kanálové osy – identifikátor • Osy stroje – identifikátor

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Příkazy **najíždění na pevný doraz** se programují v synchronních akcích/technologických cyklech pomocí příkazů výrobního programu FXS, FXST a FXSW.
Aktivování se může provádět i bez pohybu, moment je omezen okamžitě. Jakmile se osa začne pohybovat vzhledem k požadované hodnotě, je doraz monitorován.
Posuv s omezeným momentem/silou (FOC):
Funkce umožňuje pomocí synchronní akce moment/sílu kdykoli změnit; může být aktivována s modální nebo blokovou platností.

Upozornění

Opakované aktivování
Aktivování smí být provedeno jen jednou. Pokud je v důsledku chybného programování funkce po aktivování (FXS[Osa]=1) volána ještě jednou, spustí se alarm 20092 „Najíždění na pevný doraz je ještě aktivní“.
Programování, které v podmínce vyhodnocuje buď $AA_FXS[ ] nebo nějakou vlastní značku (zde R1), zabraňuje opakovanému aktivování funkce.
Část výrobního programu:

```
N10 R1=0
N20 IDS=1 WHENEVER ($R1==0 AND $AA_IW[AX3] > 7) DO R1=1 FXST[AX1]=12
```

Blokové synchronní akce
Naprogramováním blokové synchronní akce je možné spustit najíždění na pevný doraz i v průběhu najížděcího pohybu.
Příklad programování:

```
N10 G0 G90 X0 Y0
N20 WHEN $AA_IW[X] > 17 DO FXS[X]=1   ; Pokud osa X dosáhne větší pozice než
N30 G1 F200 X100 Y110                 ; 17 mm, aktivuje se FXS
```

Statické a blokové synchronní akce
Ve statických a blokových synchronních akcích je možné používat stejné příkazy FXS, FXST a FXSW, jako při zpracování normálního výrobního programu. Přiřazované hodnoty mohou pocházet i z výpočtů.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Najíždění na pevný doraz (FXS)

Spouštěno synchronní akcí.

Osa Y:	; aktivování statické synchronní akce:
N10 IDS=1 WHENEVER (($R1==1) AND ($AA_FXS[y]==0)) DO $R1=0 FXS[Y]=1 FXST[Y]=10 FA[Y]=200 POS[Y]=150	; nastavením $R1=1 se pro osu Y aktivuje ; FXS, uplatňovaný moment se zmenší na ; 10% a spustí se pohyb najíždění ve směru ; dorazu
N11 IDS=2 WHENEVER ($AA_FXS[Y]==4) DO FXST[Y]=30	; jakmile je doraz rozpoznán ($AA_FXS[Y] ; ==4), moment bude zvýšen na 30%
N12 IDS=3 WHENEVER ($AA_FXS[Y]==1) DO FXST[Y]=$R0	; po dosažení dorazu bude moment řízen ; v závislosti na R0
N13 IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1)) DO FXS[Y]=0 FA[Y]=1000 POS[Y]=0	; deaktivování závislosti ; parametru R3 a ; zpětný pohyb
N20 FXS[Y]=0 G0 G90 X0 Y0	; normální zpracování programu: uvolnění
N30 RELEASE(Y)	; osy Y pro pohyb v synchronní akci
N40 G1 F1000 X100	; pohyb jiné osy
N50	;
N60 GET(Y)	; opětovné včlenění osy Y do svazku os

Příklad programování

Aktivování omezení momentu/síly

N10 FOCON[X]	; modální aktivování omezení
N20 X100 Y200 FXST[X]=15	; X najíždí se sníženým momentem
N30 FXST[X]=75 X20	; změna momentu na 75%, X se pohybuje ; s tímto omezením momentu
N40 FOCOF[X]	; deaktivování omezení momentu

© Siemens AG 2004, Všechna práva vyhrazena

10.4.27 Stanovení parametru s v synchronních akcích

Funkce

Systémová proměnná $AC_TANEB (**T**angent **AN**gel at **E**nd of **B**lock) zjišťuje úhel mezi tečnou dráhy v koncovém bodě aktuálního bloku a tečnou dráhy v počátečním bodě naprogramovaného následujícího bloku.

V intervalu 0.0 až 180.0 se úhel tečny vždy vypisuje jako kladný. Pokud v následujícím programu neexistuje žádný následující blok, bude se uvádět úhel –180.0.

Systémová proměnná $AC_TANEB by se neměla načítat pro bloky, které jsou vytvářeny systémem (pomocné bloky). Pro rozlišení, zda se jedná o naprogramovaný blok (hlavní blok), slouží systémová proměnná $AC_BLOCKTYPE.

Příklad programování

```
ID=2 EVERY $AC_BLOCKTYPE==0 DO $R1 = $AC_TANEB;
```

10.4.28 Stanovení aktuální korekce (override)

Funkce

Aktuální korekce

Aktuální korekci (NC-podíl) je možné načítat a zapisovat pomocí systémových proměnných:

$AA_OVR	korekce osy
$AC_OVR	korekce dráhy

Korekce zadaná z PLC může být pro synchronní akce nachystána ke čtení v systémových proměnných:

$AA_PLC_OVR	korekce osy
$AC_PLC_OVR	korekce dráhy

© Siemens AG 2004, Všechna práva vyhrazena

Výsledná korekce (override):
může být pro synchronní akce připravena ke čtení v systémových proměnných:
$AA_TOTAL_OVR korekce osy
$AC_TOTAL_OVR korekce dráhy

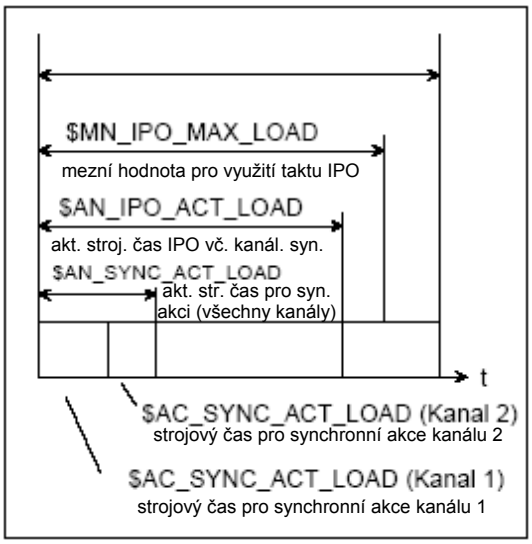
Výsledná korekce se vypočítá jako součin:
$AA_OVR * $AA_PLC_OVR příp.
$AC_OVR * $AC_PLC_OVR

10.4.29 Vyhodnocování vytížení pomocí časové náročnosti synchronní akce

Funkce

V interpolačním taktu musí být interpretovány jak synchronní akce, tak také musí být vypočítány pohyby z NC. Pomocí systémových proměnných popsaných v dalších odstavcích se dají do synchronních akcí předávat informace prostřednictvím aktuálních časových podílů synchronních akcí na interpolačním taktu a prostřednictvím strojového času regulátoru polohy.
Proměnné obsahují platné hodnoty jen tehdy, pokud je strojní parametr $MN_IPO_MAX_LOAD větší než nula, jinak tyto proměnné vždy udávají hrubý čas. Tento čas vyplývá z:
- času synchronní akce
- času regulátoru polohy a
- zbývajícího strojového času IPO

Proměnné vždy obsahují hodnoty **předcházejícího** interpolačního taktu.



$AN_IPO_ACT_LOAD	Aktuální strojový čas IPO (včetně synchronních akcí ve všech kanálech)
$AN_IPO_MAX_LOAD	Nejdelší strojový čas IPO (včetně synchronních akcí ve všech kanálech)
$AN_IPO_MIN_LOAD	Nejkratší strojový čas IPO (včetně synchronních akcí ve všech kanálech)
$AN_IPO_LOAD_PERCENT	Aktuální strojový čas IPO v poměru vůči taktu IPO (%)
$AN_SYNC_ACT_LOAD	Aktuální strojový čas pro synchronní akce přes všechny kanály
$AN_SYNC_MAX_LOAD	Nejdelší strojový čas pro synchronní akce přes všechny kanály

© Siemens AG 2004, Všechna práva vyhrazena

$AN_SYNC_TO_IPO	Procentuální podíl celkové synchronní akce na celkovém strojovém času IPO (přes všechny kanály)
$AC_SYNC_ACT_LOAD	Aktuální strojový čas pro synchronní akce v kanálu
$AC_SYNC_MAX_LOAD	Nejdelší strojový čas pro synchronní akce v kanálu
$AC_SYNC_AVERAGE_LOAD	Průměrný strojový čas pro synchronní akce v kanálu
$AN_SERVO_ACT_LOAD	Aktuální strojový čas regulátoru polohy
$AN_SERVO_MAX_LOAD	Nejdelší strojový čas regulátoru polohy
$AN_SERVO_MIN_LOAD	Nejkratší strojový čas regulátoru polohy

Signalizace přetížení:
Pomocí MD $MN_IPO_MAX_LOAD je možné nastavit, od kterého hrubého strojového času IPO (v % z interpolačního taktu) se bude systémová proměnná $AN_IPO_LOAD_LIMIT nastavovat na TRUE. Pokud aktuální zatížení opět poklesne pod tuto mez, bude proměnná znovu nastavena na FALSE. Je-li tento MD nastaven na nulu, je celá diagnostická funkce vypnuta.
Vyhodnocováním proměnné $AN_IPO_LOAD_ LIMIT může uživatel sestavit svou vlastní strategii pro prevenci přeběhu v rovině.
Ze synchronních akcí je možné zapisovat do následujících systémových proměnných:

$AN_SERVO_MAX_LOAD	Nejdelší strojový čas regulátoru polohy
$AN_SERVO_MIN_LOAD	Nejkratší strojový čas regulátoru polohy
$AN_IPO_MAX_LOAD	Nejdelší strojový čas IPO (včetně synchronních akcí ve všech kanálech)
$AN_IPO_MIN_LOAD	Nejkratší strojový čas IPO (včetně synchronních akcí ve všech kanálech)
$AN_SYNC_MAX_LOAD	Nejdelší strojový čas pro synchronní akce přes všechny kanály
$AC_SYNC_MAX_LOAD	Nejdelší strojový čas pro synchronní akce v kanálu

Tyto proměnné jsou při každém přístupu kvůli zápisu nastaveny na aktuální zatížení, a to nezávisle na zapisované hodnotě.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

$MN_IPO_MAX_LOAD = 80

MD: Mezní hodnota využití pro takt IPO. Je-li $AN_IPO_LOAD_PERCENT > 80%, bude $AN_IPO_LOAD_LIMIT nastavena na TRUE.

```
N01 $AN_SERVO_MAX_LOAD=0
N02 $AN_SERVO_MIN_LOAD=0
N03 $AN_IPO_MAX_LOAD=0
N04 $AN_IPO_MIN_LOAD=0
N05 $AN_SYNC_MAX_LOAD=0
N06 $AC_SYNC_MAX_LOAD=0
N10 IDS=1 WHENEVER $AN_IPO_LOAD_LIMIT == TRUE DO M4711 SETAL(63111)
N20 IDS=2 WHENEVER $AN_SYNC_TO_IPO > 30 DO SETAL(63222)
N30 G0 X0 Y0 Z0
...
N999 M30
```

První synchronní akce aktivuje výstup pomocné funkce a alarm, pokud je překročena celková mezní hodnota využití. Druhá synchronní akce spouští alarm tehdy, pokud podíl synchronních akcí na strojovém času IPO (přes všechny kanály) překročí 30%.

10.5 Technologické cykly

Funkce

Jako akce v synchronních akcích mohou být spouštěny také programy, které však musí být sestaveny jen jako funkce a které jsou také jako akce v synchronních akcích přípustné. Takto vytvořené programy se nazývají technologické cykly.

Technologické cykly se v řídícím systému ukládají jako podprogramy. Pro uživatele se volání uskutečňuje stejně jako u podprogramů. Předávání parametrů není možné.

V jednom kanálu může být souběžně zpracováváno i více technologických cyklů nebo akcí.

Pro konec programu je třeba naprogramovat příkazy M02/M17/M30/RET. V jednom bloku smí být naprogramován maximálně jeden pohyb osy.

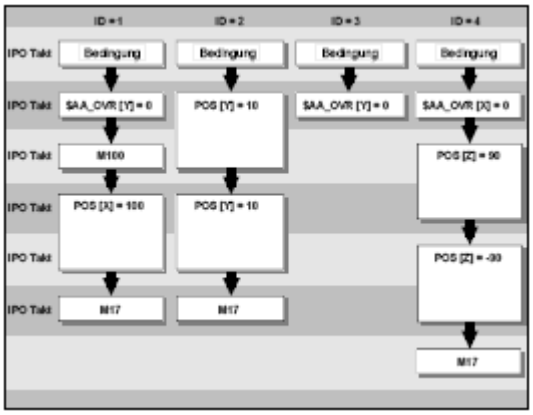
© Siemens AG 2004, Všechna práva vyhrazena

Použití

Technologické cykly jako programy pro osu:
Každý technologický cyklus řídí jen jednu osu. Tak mohou být různé pohyby osy řízeny na základě událostí v tomtéž interpolačním taktu. V extrémním případě slouží výrobní program pouze pro správu synchronních akcí.

Příklad programování

Aktivováním digitálních vstupů se spouští programy pro osy.



Hlavní program:

`ID=1 EVERY $A_IN[1]==1 DO AXIS_X`	Je-li vstup 1 na 1, spouští se programy osy X
`ID=2 EVERY $A_IN[2]==1 DO AXIS_Y`	Je-li vstup 2 na 1, spouští se programy osy Y
`ID=3 EVERY $A_IN[3]==1 DO $AA_OVR[Y]=0`	Je-li vstup 3 na 1, override osy Y se nastaví na 0
`ID=4 EVERY $A_IN[4]==1 DO AXIS_Z`	Je-li vstup 4 na 1, spouští se programy osy Z
`M30`	

Technologický cyklus AXIS_X:

```
$AA_OVR[Y]=0
M100
POS[X]=100 FA[X]=300
M17
```

Technologický cyklus AXIS_Y:

```
POS[Y]=10 FA[Y]=200
POS[Y]=-10
M17
```

Technologický cyklus AXIS_Z:

```
$AA_OVR[X]=0
POS[Z]=90 FA[Z]=250
POS[Z]=-90
M17
```


© Siemens AG 2004, Všechna práva vyhrazena

Technologické cykly se spouští, jakmile jsou jejich podmínky splněny. U polohovacích os je pro jejich uskutečnění zapotřebí více taktů IPO. Jiné funkce (OVR) jsou prováděny během jednoho taktu.
V technologickém cyklu jsou bloky zpracovávány sekvenčně.



Upozornění
Pokud jsou v tomtéž interpolačním taktu vyvolávány akce, které se vzájemně vylučují, spustí se ta akce, která byla vyvolána ze synchronní akce s vyšším identifikačním číslem.

10.5.1 Zablokování, odblokování, přerušení, LOCK, UNLOCK, RESET



Programování

`LOCK (n, n, ...)`	Zablokování technologického cyklu, aktivní akce bude přerušena
`UNLOCK (n, n, ...)`	Uvolnění technologického cyklu
`RESET (n, n, ...)`	Přerušení technologického cyklu, aktivní akce bude přerušena
`n`	Identifikační číslo synchronní akce



Funkce

Zpracování technologického cyklu může být ze synchronní akce nebo z jiného technologického cyklu zablokováno, uvolněno nebo přerušeno.

Zablokování technologického cyklu, LOCK
Pomocí příkazu LOCK je možné ze synchronní akce nebo z technologického cyklu zablokovat zpracování jiného technologického cyklu.
Příklad:

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
```

© Siemens AG 2004, Všechna práva vyhrazena

Uvolnění technologického cyklu, UNLOCK

Zablokované technologické cykly je možné pomocí příkazu UNLOCK z jiné synchronní akce nebo technologického cyklu opět odblokovat. Po příkazu UNLOCK bude jeho zpracování pokračovat na aktuálním místě, rovněž v případě přerušené operace polohování.

Příklad:

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
...
N250 ID=3 WHENEVER $A_IN[3]==1 DO UNLOCK(1)
```

Přerušení technologického cyklu, RESET

Technologický cyklus je možné přerušit pomocí příkazu RESET z jiné synchronní akce nebo jiného technologického cyklu.

Příklad:

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO RESET(1)
```

Zablokování ze strany PLC

Modální synchronní akce s identifikačními čísly **n=1 ... 64** mohou být blokovány i z PLC. Příslušná podmínka se pak už vůbec nevyhodnocuje a spouštění odpovídající funkce je v NCK zablokováno.

Signálem z rozhraní PLC se dají paušálně synchronní akce zablokovat.

Upozornění

Naprogramovaná synchronní akce je standardně aktivní a může být chráněna proti přepsání/zablokování pomocí strojního parametru.

Použití:
Synchronní akce definované výrobcem stroje nemají být ovlivňovány koncovými zákazníky.

© Siemens AG 2004, Všechna práva vyhrazena

10.6 Vymazání synchronní akce, CANCEL

Programování

`CANCEL(n,n,...)`	Vymazání synchronní akce
`n`	Identifikační číslo synchronní akce

Vysvětlení

Synchronní akce s modální platností s identifikačním číslem ID = n mohou být mazány pouze přímo z výrobního programu.

Příklad:

```
N100 ID=2 WHENEVER $A_IN[1]==1 DO M130
...
N200 CANCEL(2)                    ; vymazání synchronní akce č.2
```

Upozornění
Pohyby spuštěné ze synchronní akce, která byla vymazána, ale které ještě probíhají, budou dokončeny v souladu s naprogramováním.

10.7 Okrajové podmínky

- **Power ON**
  Při zapnutí (Power ON) nejsou v zásadě žádné synchronní akce aktivní.
  Při zapnutí však mohou být aktivovány statické synchronní akce spouštěné pomocí asynchronního podprogramu (ASUP) z PLC.

- **Změna provozního režimu**
  Synchronní akce, které byly spuštěny pomocí klíčového slova IDS, zůstávají aktivní i po změně provozního režimu.
  Všechny zbývající synchronní akce se při změně provozního režimu stávají neaktivními (např. polohování os) a s uvedením do původní polohy a přepnutím zpět do automatického režimu se znovu aktivují.

© Siemens AG 2004, Všechna práva vyhrazena

- **Reset**
  Při resetu NC-systému se přeruší všechny operace spouštěné synchronními akcemi. Statické synchronní akce zůstávají aktivní. Z nich mohou být spouštěny nové akce. Příkazem **RESET** ze synchronní akce nebo z technologického cyklu mohou být resetovány synchronní akce s modální platností. Jestliže je resetována modální akce, zatímco dosud probíhá z ní spuštěný pohyb polohovací osy, bude pohyb polohovací osy přerušen. Synchronní akce typu WHEN, které už byly provedeny, se po resetu znovu nespouští.

Chování po resetu		
Synchronní akce / technologický cyklus	**Modální / bloková platnost**	**Statické (IDS)**
	Aktivní akce se přeruší, synchronní akce je vymazána	Aktivní akce se přeruší, technologický cyklus je resetován
Osa / polohovací vřeteno	Pohyb se přeruší	Pohyb se přeruší
Vřeteno s regulací otáček	$MA_SPIND_ACTIVE_AFTER_RESET==1: Vřeteno zůstává aktivní $MA_SPIND_ACTIVE_AFTER_RESET==0: Vřeteno se zastaví	$MA_SPIND_ACTIVE_AFTER_RESET==1: Vřeteno zůstává aktivní $MA_SPIND_ACTIVE_AFTER_RESET==0: Vřeteno se zastaví
Spojení řídící hodnotou	$MC_RESET_MODE_MASK, bit13==1: Spojení řídící hodnotou zůstává aktivní $MC_RESET_MODE_MASK, bit13==0: Spojení řídící hodnotou se zruší	$MC_RESET_MODE_MASK, bit13==1: Spojení řídící hodnotou zůstává aktivní $MC_RESET_MODE_MASK, bit13==0: Spojení řídící hodnotou se zruší
Měřicí operace	Měřicí operace spouštěné ze synchronní akce budou přerušeny.	Měřicí operace spouštěné ze statické synchronní akce budou přerušeny.

- **NC-Stop**
  Statické synchronní akce zůstávají při NC-Stop aktivní. Pohyby spuštěné ze statické synchronní akce nebudou přerušeny.
  Programové lokální synchronní akce náležející k aktivnímu bloku zůstávají aktivní, z nich spouštěné pohyby jsou přerušeny.

- **Konec programu**
  Konec programu a synchronní akce se vzájemně neovlivňují. Probíhající synchronní

© Siemens AG 2004, Všechna práva vyhrazena

akce nejsou po skončení programu ukončovány.
Synchronní akce aktivní v loku M30 zůstávají i v tomto bloku aktivní. Pokud je to nežádoucí, musí být synchronní akce před koncem programu zrušena pomocí příkazu CANCEL (viz předešlá podkapitola).

Chování po skončení programu		
Synchronní akce / technologický cyklus	**modální a blokové** budou přerušeny	**statické (IDS)** zůstanou zachovány
Osa / polohovací vřeteno	M30 bude zpožděn, dokud osa/vřeteno stojí.	Pohyb pokračuje.
Vřeteno s regulací otáček	Konec programu: $MA_SPIND_ACTIVE_AFTER_RESET==1: Vřeteno zůstane aktivní. $MA_SPIND_ACTIVE_AFTER_RESET==1: Vřeteno se zastaví Při změně provozního režimu zůstává vřeteno aktivní.	Vřeteno zůstává aktivní.
Spojení řídící hodnotou	$MC_RESET_MODE_MASK, bit13==1: Spojení řídící hodnotou zůstává aktivní $MC_RESET_MODE_MASK, bit13==1: Spojení řídící hodnotou je deaktivováno.	Spojení spuštěné ze statické synchronní akce zůstává zachováno
Měřicí operace	Měřicí operace spouštěné ze synchronní akce budou přerušeny.	Měřicí operace spouštěné ze statické synchronní akce zůstanou aktivní.

- **Vyhledávání bloku**
  Synchronní akce nalezené během hledání bloku budou shromážděny a při NC-Start vyhodnoceny; v případě potřeby se spustí odpovídající akce.
  Statické synchronní akce zůstávají v platnosti i během vyhledávání bloku.
  Pokud jsou při hledání bloku nalezeny koeficienty polynomu naprogramované pomocí příkazu FCTDEF, budou okamžitě uplatněny.

- **Přerušení programu asynchronním podprogramem**
  ASUP-počátek:
  Modální a statické pohybové synchronní akce zůstanou zachovány a jsou v platnosti také v asynchronním podprogramu.

© Siemens AG 2004, Všechna práva vyhrazena

ASUP-konec
Pokud asynchronní podprogram nepokračuje příkazem REPOS, zůstávají modální a statické pohybové synchronní akce v hlavním programu změněné asynchronním podprogramem v platnosti i nadále.

- **Najíždění na původní pozici**
  Po návratu na původní pozici příkazem REPOS zůstávají synchronní akce, které byly v platnosti v přerušeném bloku, opět aktivní.
  Modální synchronní akce, které byly změněny asynchronním podprogramem, se po příkazu REPOS při zpracování zbývajících bloků neuplatňují.
  Koeficienty polynomu naprogramované příkazem FCTDEF nejsou asynchronními podprogramy a příkazy REPOS nijak ovlivňovány. Tam, kde byly naprogramovány, mohou být v synchronním podprogramu a v hlavním programu po provedené příkazu REPOS kdykoli použitelné.

- **Deaktivování příkazem CANCEL**
  Pokud je aktivní synchronní akce deaktivována příkazem CANCEL, nebudou ostatní aktivní akce nijak ovlivněny. Polohovací pohyby budou dokončeny podle programu.
  Příkazem CANCEL mohou být zrušeny synchronní akce s modální nebo statickou platností.
  Pokud je zrušena synchronní akce, která spustila polohovací pohyb, který ještě probíhá, bude tento pohyb dokončen. Jestliže je to nežádoucí, může být pohyb osy před příkazem CANCEL ukončen s vymazáním zbytkové dráhy.
  Příklad:

```
ID=17 EVERY $A_IN[3]==1 DO POS[X]=15 FA[X]=1500
                                        ; spuštění pohybu polohovací osy
...
WHEN ... DO DELDTG(X)                   ; ukončení pohybu polohovací osy
CANCEL(1)
```


© Siemens AG 2004, Všechna práva vyhrazena

Oscilační pohyb

11.1 Asynchronní oscilační pohyb 11-482
11.2 Oscilační pohyb řízený synchronními akcemi 11-489

© Siemens AG 2004, Všechna práva vyhrazena

11.1 Asynchronní oscilační pohyb

Vysvětlení příkazu

`OSP1 [Osa] =`	Poloha bodu obratu 1
`OSP2 [Osa] =`	Poloha bodu obratu 2
`OST1 [Osa] =`	Doba prodlevy v bodech obratu v sekundách
`OST2 [Osa] =`	
`FA [Osa] =`	Posuv oscilační osy
`OSCTRL [Osa] =`	(možnosti nastavení, resetu)
`OSNSC [Osa] =`	Počet vyjiskřovacích zdvihů
`OSE [Osa] =`	Koncová pozice
`OS [Osa] =`	1 = aktivování oscilační osy, 0 = deaktivování oscilační osy

Funkce

Oscilační osa se pohybuje tam a zpátky mezi dvěma body obratu 1 a 2 se zadaným posuvem, dokud není oscilační pohyb deaktivován. Ostatní osy mohou být v průběhu oscilačního pohybu interpolovány libovolně.
Pohybem po dráze nebo polohovací osou je možno dosáhnout kontinuálního přísuvu. Přitom však neexistuje žádná souvislost mezi oscilačním a přísuvným pohybem.

© Siemens AG 2004, Všechna práva vyhrazena

Oscilační osa

Pro oscilační osu platí:

- Každá osa může být použita jako oscilační osa.
- Současně může být aktivních i více oscilačních os (maximum: počet polohovacích os).
- Pro oscilační osu je vždy – nezávisle na G-funkci, která je právě platná v programu – aktivní lineární interpolace G1.

Oscilační osa může:

- Být vstupní osou pro dynamickou transformaci
- Být vodicí osou u os gantry – vlečných os
- Být ovládána
  - bez omezení trhavých pohybů (BRISK)
  - s omezením trhavých pohybů (SOFT)
  - se zalomenou charakteristikou zrychlení (jako polohovací osy).

Body obratu oscilace

Při definici oscilačních poloh je třeba dávat pozor na aktuální posunutí:

- Absolutní zadání
  OSP1[Z] = hodnota
  Poloha bodu obratu = součet posunutí + naprogramovaná hodnota
- Relativní udání
  OSP1[Z] = IC(hodnota)
  Poloha bodu obratu = bod obratu 1 + naprogramovaná hodnota

Příklad:

```
N10 OSP1[Z]=100 OSP2[Z]=110
...
...
N40 OSP1[Z]=IC(3)
```

© Siemens AG 2004, Všechna práva vyhrazena

Vlastnosti asynchronního oscilačního pohybu

- Asynchronní oscilační pohyb je specifický pro osu a má platnost i za hranice bloku.
- Blokově synchronizované aktivování oscilačního pohybu je zabezpečováno výrobním programem.
- Společná interpolace více os a superpozice oscilačních drah nejsou možné.

Nastavované parametry

Nastavované parametry, které jsou zapotřebí pro asynchronní oscilaci, mohou být nastavovány ve výrobním programu.

Pokud jsou hodnoty nastavovaným parametrům dosazovány přímo ve výrobním programu, jsou změny v platnosti už v okamžiku předběžného zpracování. Synchronního chování je možné dosáhnout pomocí zastavení předběžného zpracování příkazem STOPRE.

Příklad:

Oscilace s on-line změnou polohy bodu obratu

```
$SA_OSCILL_REVERSE_POS1[Z]=-10
$SA_OSCILL_REVERSE_POS2[Z]=10

G0 X0 Z0
WAITP(Z)

ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0
                                        ; pokud je skutečná poloha oscilační osy
                                        ; v bodě obratu překročena,
                                        ; osa přísuvu se zastaví
OS[Z]=1 FA[X]=1000 POS[X]=40            ; aktivování oscilace
OS[Z]=0                                 ; deaktivování oscilace
M30
```


© Siemens AG 2004, Všechna práva vyhrazena

Informace o jednotlivých funkcích

Asynchronní oscilace může být aktivována a ovládána z výrobního programu pomocí následujících adres.
Naprogramované hodnoty se synchronně v bloku hlavního programu přenášejí do odpovídajících nastavovaných parametrů a zde zůstávají v platnosti, dokud nejsou změněny.

WAITP(Osa)
Pokud se má geometrická osa pohybovat oscilačním pohybem, musí být pro tento druh pohybu uvolněna příkazem WAITP.
Po ukončení oscilačního pohybu se pomocí tohoto příkazu oscilační osa znovu aktivuje jako polohovací osa, takže může být znovu normálně používána

Doba pozastavení v bodech obratu: OST1, OST2

Doba pozastavení	Pohybové chování v oblasti přesného najetí, v bodě obratu
-2	Interpolace bude pokračovat bez pozastavení na přesné najetí
-1	Čekání na hrubé přesné najetí
0	Čekání na jemné přesné najetí
> 0	Čekání na jemné přesné najetí a pak čekání na dobu pozastavení

Jednotka pro dobu pozastavení je identická s dobou čekání naprogramovanou pomocí G4.

Upozornění

Oscilace s pohybovými synchronními akcemi a dobou pozastavení: OST1/OST2
Po uplynutí nastavené doby pozastavení se při oscilaci uskutečňuje interní přechod na další blok (je vidět podle nových hodnot zbytkových drah os). Při přechodu na další blok se kontroluje vypínací funkce. Přitom je podle nastavení systému pro pohybovou operaci „OSCTRL“ definována vypínací funkce.

© Siemens AG 2004, Všechna práva vyhrazena

Toto časové chování je možné ovlivňovat korekcí posuvu (override)
Za určitých okolností se oscilační zdvih uskuteční ještě dříve, než se spustí vyjiskřovací zdvihy nebo je dosaženo koncové polohy.
Vzniká tedy dojem, že se odezva na deaktivaci mění. Tak tomu ale není.

Nastavení posuvu FA
Jako rychlost posuvu platí rychlost posuvu, která byla definována pro polohovací osu.
Pokud žádná rychlost posuvu není stanovena, použije se hodnota uložená ve strojním parametru.

Definice pohybového režimu OSCTRL
Nastavení řídícího systému pro režim pohybu prostřednictvím voleb a možností resetu.

Možnosti resetu
Tyto možnosti jsou deaktivovány (jen když předtím byly aktivovány volby nastavení).

Volby nastavení
Tyto volby jsou přepínány. Při naprogramování příkazu OSE (koncová pozice) platí implicitně volba 4.

© Siemens AG 2004, Všechna práva vyhrazena

Hodnota volby	Význam
0	Při deaktivování oscilačního pohybu dojde k zastavení v následujícím bodu obratu (předdefinované nastavení); možné jen při resetu hodnot 1 a 2.
1	Při deaktivování oscilačního pohybu dojde k zastavení v bodu obratu 1.
2	Při deaktivování oscilačního pohybu dojde k zastavení v bodu obratu 2.
3	Při deaktivování oscilačního pohybu se nenajíždí do žádného bodu obratu, jestliže nejsou naprogramovány žádné vyjiskřovací zdvihy.
4	Po vyjiskřování se najíždí na koncovou pozici.
8	Pokud je oscilační pohyb přerušen vymazáním zbytkové dráhy: následné zpracování vyjiskřovacích zdvihů a příp. najetí na koncovou pozici.
16	Pokud je oscilační pohyb přerušen vymazáním zbytkové dráhy: najíždění na odpovídající bod obratu jako při deaktivování.
32	Změněná hodnota posuvu bude uplatněna až od následujícího bodu obratu.
64	FA = 0: Superpozice dráhy je aktivní FA 0: Superpozice rychlosti je aktivní
128	U kruhové osy DC (nejkratší dráha)
256	0 = Vyjiskřovací zdvih bude proveden jako dvojitý zdvih (standardní) 1 = Vyjiskřovací zdvih bude proveden jako jednoduchý zdvih

Více možností se zadává prostřednictvím znaménka plus.
Příklad:

```
OSCTRL[Z] = (1+4,16+32+64)
```

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování

Oscilační osa Z se má pohybovat mezi hodnotami 10 a 100. Na bod obratu 1 se má najíždět s jemným přesným najetím, na bod obratu 2 se má najíždět s hrubým přesným najetím. Obrábění má probíhat s rychlostí posuvu oscilační osy 250. Na konci obrábění mají být provedeny 3 vyjiskřovací zdvihy a pak má oscilační osa odjet na koncovou pozici 200.
Posuv pro přísuvnou osu je 1, konec přísuvu ve směru osy X leží na hodnotě 15.

`WAITP(X,Y,Z)`	Výchozí nastavení
`G0 X100 Y100 Z100`	Přepnutí do režimu polohovací osy
`N40 WAITP(X,Z)`	
`N50 OSP1[Z]=10 OSP2[Z]=100 ->`	Bod obratu 1, bod obratu 2
`-> OSE[Z]=200 ->`	Koncová pozice
`-> OST1[Z]=0 OST2[Z]=-1 ->`	Pozastavení v bodě 1: jemné přesné najetí Pozastavení v bodě 2: hrubé přesné najetí
`-> FA[Z]=250 FA[X]=1 ->`	Posuv oscilační osy, přísuvné osy
`-> OSCTRL[Z]=(4,0) ->`	Volby nastavení
`-> OSNSC[Z]=3 ->`	Tři vyjiskřovací zdvihy
`N60 OS[Z]=1`	Spuštění oscilačního pohybu
`N70 WHEN $A_IN[3]==TRUE ->` `-> DO DELDTG(X)`	Vymazání zbytkové dráhy
`N80 POS[X]=15`	Výstupní pozice osy X
`N90 POS[X]=50`	
`N100 OS[Z]=0`	Zastavení oscilačního pohybu
`M30`	

-> Může být naprogramováno v jednom bloku.

© Siemens AG 2004, Všechna práva vyhrazena

11.2 Oscilační pohyb řízený synchronními akcemi

Programování

1. Definice parametrů oscilačního pohybu
2. Definice pohybové synchronní akce
3. Přiřazení os, definice přísuvu

Parametry pro oscilační pohyb

`OSP1[oscilační osa]=`	Poloha bodu obratu 1
`OSP2[oscilační osa]=`	Poloha bodu obratu 2
`OST1[oscilační osa]=`	Doba prodlevy v bodu obratu 1 v sekundách
`OST2[oscilační osa]=`	Doba prodlevy v bodu obratu 2 v sekundách
`FA[oscilační osa]=`	Posuv oscilační osy
`OSCTRL[oscilační osa]=`	Volby nastavení a resetu
`OSNSC[oscilační osa]=`	Počet vyjiskřovacích zdvihů
`OSE[oscilační osa]=`	Koncová pozice
`WAITP(oscilační osa)`	Uvolnění osy pro oscilační pohyb

Přiřazení os, přísuv

```
OSCILL[oscilační osa] = (osa přísuvu1, osa přísuvu2, osa přísuvu3)
POSP[osa přísuvu] = (koncová pozice, dílčí délka, režim)
```

`OSCILL`	Přiřazení přísuvných os oscilační ose
`POSP`	Definice celkového a dílčího přísuvu (viz kapitola 3)
`koncová pozice`	Koncová pozice pro přísuvnou osu poté, co byly dokončeny všechny dílčí přísuvy
`dílčí délka`	Velikost dílčího přísuvu v bodě/oblasti obratu
`režim`	Rozdělení celkového přísuvu do dílčích přísuvů: 0 = zbytek rozdělen na dva stejně velké kroky (standardní) 1 = všechny dílčí přísuvy jsou stejně velké.

Synchronní pohybové akce

`WHEN… … DO`	Jestliže..., pak ...
`WHENEVER … DO`	Vždycky když ..., pak ...

© Siemens AG 2004, Všechna práva vyhrazena

Řízení oscilačního pohybu synchronní akcí

Při tomto druhu oscilačního pohybu je možné provádět přísuv pouze v bodech obratu, příp. v oblasti obratu.

V závislosti na požadavcích může oscilační pohyb během přísuvu:

- pokračovat
- být pozastaven, dokud není přísuv úplně dokončen.

Postup

1. Definice parametrů oscilačního pohybu
Před pohybovým blokem, který obsahuje přiřazení přísuvné a oscilační osy a definici přísuvu, je třeba definovat parametry pro oscilační pohyb (viz kapitola „Asynchronní oscilační pohyb“).

2. Definice pohybové synchronní akce
Pomocí podmínek synchronní akce se provádí:

- **Potlačení přísuvu**, dokud se oscilační osa nenachází v oblasti bodu obratu (ii1, ii2) nebo v bodu obratu (U1, U2).
- **Pozastavení oscilačního pohybu** během přísuvu v bodě obratu.
- **Spuštění oscilačního pohybu** po ukončení dílčího přísuvu.
- Stanovení **spuštění následujícího dílčího přísuvu**.

3. Přiřazení oscilační a přísuvné osy, jakož i celkového a dílčích přísuvů.

© Siemens AG 2004, Všechna práva vyhrazena

Přiřazení oscilační a přísuvné osy: OSCILL

```
OSCILL[oscilační osa] = (osa přísuvu1, osa přísuvu2, osa přísuvu3)
```

Příkazem OSCILL se provádí přiřazení os a spuštění oscilačního pohybu.

Jedné oscilační ose je možné přiřadit maximálně 3 přísuvné osy.

Před spuštěním oscilačního pohybu je nutné definovat podmínky synchronní akce pro chování os.

Definice přísuvů: POSP

```
POSP[osa přísuvu] = (koncová pozice, dílčí délka, režim)
```

Příkazem POSP se řídícímu systému sdělují následující záležitosti:

- Celkový přísuv (pomocí koncové pozice)
- Velikost jednotlivých dílčích přísuvů v bodech obratu, příp. v oblasti obratu.
- Chování dílčích přísuvů při dosažení koncové pozice (pomocí režimu)

režim = 0	Pro poslední dva dílčí přísuvy se zbývající dráha až k cílovému bodu rozdělí na dva stejně velké zbytkové kroky (předdefinované nastavení).
režim = 1	Všechny dílčí přísuvy jsou stejně velké. Vypočítávají se z celkového přísuvu.

© Siemens AG 2004, Všechna práva vyhrazena

Synchronní akce

Synchronní akce popisované v následujících odstavcích se používají pro zcela obecné oscilační pohyby.
Naleznete zde příklady řešení jednotlivých požadavků, které Vám mohou posloužit jako stavební kameny při sestavování uživatelských oscilačních pohybů.

V jednotlivých případech mohou být podmínky synchronní akce naprogramovány i jinak.

Klíčová slova

`WHEN … DO …`	Jestliže..., pak ...
`WHENEVER …DO …`	Vždycky když ..., pak ...

Pomocí jazykových prostředků v dalším podrobně popsaných můžete realizovat následující funkce:
1. Přísuv v bodě obratu
2. Přísuv v oblasti obratu
3. Přísuv v obou bodech obratu
4. Pozastavení oscilačního pohybu v bodě obratu
5. Opětovné spuštění oscilačního pohybu
6. Nikoli předčasné spuštění dílčího přísuvu

Pro všechny synchronní akce uvedené v těchto příkladech platí předpoklady:
- Bod obratu 1 < bod obratu 2
- Z = osa oscilačního pohybu
- X = přísuvná osa

Bližší vysvětlení pohybových synchronních akcí je uvedeno v kapitole 11.3.

© Siemens AG 2004, Všechna práva vyhrazena

Přísuv v oblasti obratu

Přísuvný pohyb má začínat v přísuvné oblasti, ještě než je dosaženo bodu obratu.

Tyto synchronní akce zabraňují přísuvnému pohybu, dokud se oscilační osa nenachází v oblasti obratu.

Za uvedených předpokladů (viz výše) je možné sestavit následující příkazy:

Bod obratu 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_REVERSE_POS1[Z]+ii1 DO $AA_OVR[X]=0
```

Vždycky když	aktuální poloha oscilační osy v MCS
je větší	než začátek oblasti obratu 1,
pak	nastavit hodnotu korekce (override) přísuvné osy na 0%.

Bod obratu 2:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_REVERSE_POS2[Z]+ii2 DO $AA_OVR[X]=0
```

Vždycky když	aktuální poloha oscilační osy v MCS
je menší	než začátek oblasti obratu 2,
pak	nastavit hodnotu korekce (override) přísuvné osy na 0%.

© Siemens AG 2004, Všechna práva vyhrazena

Přísuv v bodě obratu

Dokud oscilační osa nedosáhla bodu obratu,
v přísuvné ose se neprovádí žádný pohyb.

Za uvedených předpokladů (viz výše) je možné
sestavit následující příkazy:

Bod obratu 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0 ->
-> $AA_OVR[Z]=100
```

Vždycky když	aktuální poloha oscilační osy Z v MCS
je větší nebo menší	než poloha bodu obratu 1,
pak	nastavit hodnotu korekce (override) přísuvné osy X na 0%
a	hodnotu korekce oscilační osy Z na 100%.

Bod obratu 2:

Pro bod obratu 2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0 ->
-> $AA_OVR[Z]=100
```

Vždycky když	aktuální poloha oscilační osy Z v MCS
je větší nebo menší	než poloha bodu obratu 2,
pak	nastavit hodnotu korekce (override) přísuvné osy X na 0%
a	hodnotu korekce oscilační osy Z na 100%.

© Siemens AG 2004, Všechna práva vyhrazena

Pozastavení oscilačního pohybu v bodě obratu

Oscilační osa bude pozastavena v bodě obratu a současně začne přísuvný pohyb.
Oscilační pohyb bude pokračovat, jakmile je přísuvný pohyb úplně dokončen.

Současně mohou být tyto synchronní akce použity k tomu, aby se přísuvný pohyb spustil, jestliže byl zastaven předcházející synchronní akcí, která je ještě aktivní.

Za uvedených předpokladů (viz výše) je možné sestavit následující příkazy:

Bod obratu 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_REVERSE_POS1[Z]DO $AA_OVR[Z]=0 ->
-> $AA_OVR[X] = 100
```

Vždycky když	aktuální poloha oscilační osy v MCS
je rovna	poloze bodu obratu 1,
pak	nastavit hodnotu korekce (override) oscilační osy na 0%
a	hodnotu korekce přísuvné osy na 100%.

Bod obratu 2:

Pro bod obratu 2:

```
WHENEVER $SA_IM[Z] ==$SA_OSCILL_REVERSE_POS2[Z]DO $AA_OVR[Z]= 0 ->
-> $AA_OVR[X]=100
```

Vždycky když	aktuální poloha oscilační osy v MCS
je rovna	poloze bodu obratu 2,
pak	nastavit hodnotu korekce (override) oscilační osy na 0%
a	hodnotu korekce přísuvné osy na 100%.

© Siemens AG 2004, Všechna práva vyhrazena

On-line vyhodnocování bodu obratu
Jestliže se na pravé straně výrazu porovnávání nachází proměnná hlavního zpracování označená $$, budou obě proměnné vyhodnoceny a porovnány v rámci taktu IPO.

Další informace o tomto tématu naleznete v kapitole „Pohybové synchronní akce“.

Opětovné spuštění oscilačního pohybu

Tyto synchronní akce se používají k opětovnému spouštění pohybu oscilační osy, jestliže je pohyb do dílčí přísuvné hloubky ukončen.

Za uvedených předpokladů (viz výše) je možné sestavit následující příkazy:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]= 100
```

Vždycky když	je zbytková dráha pro dílčí přísuv přísuvné osy X v MCS
rovna	nule,
pak	nastavit hodnotu korekce (override) oscilační osy na 100%

© Siemens AG 2004, Všechna práva vyhrazena

Následující dílčí přísuv

Po dokončení přísuvu je nutné zabránit příliš brzkému spuštění následujícího dílčího přísuvu. Pro tento účel se používá kanálová značka (`$AC_MARKER[Index]`), která se nastavuje na konci dílčího přísuvu (zbytková dráha přísuvu = 0) a při opuštění oblasti obratu se vymaže. Tak se pomocí synchronní akce zabrání následujícímu přísuvnému pohybu.

Za uvedených předpokladů (viz výše) je možné např. pro bod obratu 1 sestavit následující příkazy:

1. Nastavení značky:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1]=1
```

Vždycky když	je zbytková dráha pro dílčí přísuv přísuvné osy X ve WCS
rovna	nule,
pak	nastavit značku s indexem 1 na 1.

2. Vymazání značky:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_REVERSE_POS1[Z] D0 $AC_MARKER[1]=0
```

Vždycky když	je aktuální pozice oscilační osy Z v MCS
větší nebo menší	než pozice bodu obratu 1,
pak	nastavit značku 1 na 0.

3. Zabránění přísuvu:

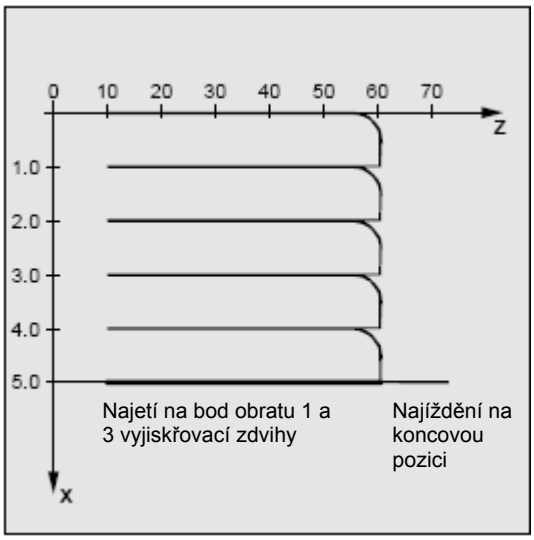
```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

Vždycky když	značka 1
je rovna	jedné,
pak	nastavit korekci (override) přísuvné osy X na 0%.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

V bodě obratu 1 se neprovádí žádný přísuv.
V bodě obratu 2 se má uskutečnit přísuv již ve vzdálenosti ii2 před bodem obratu 2 a oscilační osa nemá v bodě obratu čekat, až bude dílčí přísuv ukončen. Oscilační osou je osa Z a osa X je přísuvnou osou.



`DEF INT ii2`	Definice proměnné pro oblast obratu 2
`OSP1[Z]=10 OSP2[Z]=60`	Definice bodů obratu 1 a 2
`OST1[Z]=0 OST2[Z]=0`	Bod obratu 1: jemné přesné najetí Bod obratu 2: jemné přesné najetí
`FA[Z]=150 FA[X]=0.5`	Posuv oscilační osy Z, posuv přísuvné osy X
`OSCTRL[Z]=(2+8+16,1)`	Deaktivování oscilačního pohybu v bodu obratu 2; po RWL vyjiskřování a najetí na koncový bod; po RWL najíždění na odpovídající pozici obratu
`OSNC[Z]=3`	3 vyjiskřovací zdvihy
`OSE[Z]=70`	Koncová pozice = 70
`ii2=2`	Nastavení oblasti obratu
`WAITP(Z)`	Povolení oscilačního pohybu pro osu Z

© Siemens AG 2004, Všechna práva vyhrazena

2. Pohybové synchronní akce

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z]-ii2 DO ->
-> $AA_OVR[X]=0 $AC_MARKER[0]=0
```

Vždycky když	je aktuální pozice oscilační osy Z v MCS
menší	než začátek oblasti obratu 2,
pak	nastavit korekci (override) přísuvné osy X na 0%
a	

```
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0
```

Vždycky když	je aktuální pozice oscilační osy Z v MCS
větší nebo rovna	poloze bodu obratu 2,
pak	nastavit korekci (override) oscilační osy Z na 0%

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1
```

Vždycky když	je zbytková dráha dílčího přísuvu
rovna	nule,
pak	značku s indexem 0 nastavit na hodnotu 1.

```
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
```

Vždycky když	je značka s indexem 0
rovna	jedné,
pak	nastavit korekci (override) přísuvné osy X na 0%, aby se zabránilo příliš brzkému přísuvu (oscilační osa Z ještě oblast obratu neopustila, přísuvná osa X je však připravena pro nový přísuv); nastavit korekci (override) oscilační osy Z na 100% (čímž bude 2. synchronní akce zrušena.

-> musí být naprogramováno v jednom bloku

3. Spuštění oscilačního pohybu

`OSCILL[Z]=(X) POSP[X]=(5,1,1)`	Spuštění os Osa Z je přiřazena ose X jako přísuvná osa. Osa X má najet do koncové pozice 5 v krocích po 1.
`M30`	Konec programu

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Lisování a prostřihování

12.1 Aktivace, deaktivace 12-502
12.1.1 Příkazy SPOF, SON, PON, SONS, PONS, PDEALYON/OF 12-502
12.1.2 Použití M-příkazů 12-505

12.2 Automatické rozdělení dráhy 12-506
12.2.1 Rozdělení dráhy u dráhových os 12-507
12.2.2 Rozdělení dráhy u samostatných os 12-508
12.2.3 Příklady programování 12-510

© Siemens AG 2004, Všechna práva vyhrazena

12.1 Aktivace, deaktivace

12.1.1 Příkazy SPOF, SON, PON, SONS, PONS, PDEALYON/OF

Programování

```
PDELAYON
PON G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYOF
SON G... X... Y... Z...
SONS G... X... Y... Z...
SPOF
PUNCHACC(Smin, Amin, Smax, Amax)
```

Vysvětlení parametrů

PON	Aktivování lisování
PONS	Aktivování lisování se zaváděním
SON	Aktivování prostřihování
SONS	Aktivování prostřihování se zaváděním
SPOF	Deaktivování prostřihování, lisování
PDELAYON	Aktivování lisování se zpožděním
PDELAYOF	Deaktivování lisování se zpožděním
PUNCHACC	Zrychlení závisející na dráze PUNCHACC (S_{min}, A_{min}, S_{max}, A_{max})
S_{min}	Nejmenší vzdálenost děr
A_{min}	Největší vzdálenost děr
S_{max}	Počáteční zrychlení A_{min} může být větší než A_{max}
A_{max}	Koncové zrychlení A_{max} může být menší než A_{min}

Funkce

Aktivování lisování/prostřihování PON/SON

Pomocí příkazů PON a SON aktivujete funkci lisování, resp. prostřihování. Příkazem SPOF se všechny specifické funkce pro prostřihování a lisování ukončí.

Příkazy PON a SON mají modální platnost a vzájemně se vylučují, tzn. příkaz PON deaktivuje SON a obráceně.

© Siemens AG 2004, Všechna práva vyhrazena

Lisování a prostřihování se zaváděním, PONS/SONS

Funkce PONS a SONS rovněž aktivují funkce pro lisování a prostřihování.
Oproti příkazům PON/SON – řízení zdvihu na interpolační úrovni – se u těchto funkcí uskutečňuje ještě signálové řízení spouštění zdvihu na úrovni servomechanismu.
Díky tomu je možné pracovat s vyšší frekvencí zdvihů a tedy s vyšším výkonem lisu.

V průběhu vyhodnocování signálu v zavádění jsou zablokovány všechny funkce, které slouží ke změně polohy os provádějících lisování nebo prostřihování.
Příklad: Posuv ručním kolečkem, změna framu pomocí PLC, měřicí funkce.

Lisování a prostřihování se zaváděním není možné ve více kanálech současně. Příkazy POS, resp. SONS mohou být aktivovány jen v jednom kanálu.
Od **SW 7.1**: Pokud jsou příkazy PONS nebo SONS aktivovány ve více než jednom kanálu současně, tato nepovolená akce bude rozpoznána s alarmem „Kanál %1 rychlé lisování/prostřihování ve více kanálech není možné“.

Jinak se funkce PONS a SONS chovají stejně jako funkce PON a SON.

Zrychlení závislé na dráze PUNCHACC

Příkaz PUNCHACC(S_{min}, A_{min}, S_{max}, A_{max}) definuje charakteristiku zrychlení, podle které jsou nastavována různá zrychlení (A) v závislosti na vzdálenosti děr (S).
Příklad PUNCHACC (2, 50, 10, 100):

Vzdálenosti děr pod 2 mm:
Posuv se bude provádět s 50% zrychlením oproti maximálnímu zrychlení.

Vzdálenosti děr 2 mm až 10 mm:
Zrychlení se bude zvyšovat přímo úměrně vzdálenosti děr až do 100%.

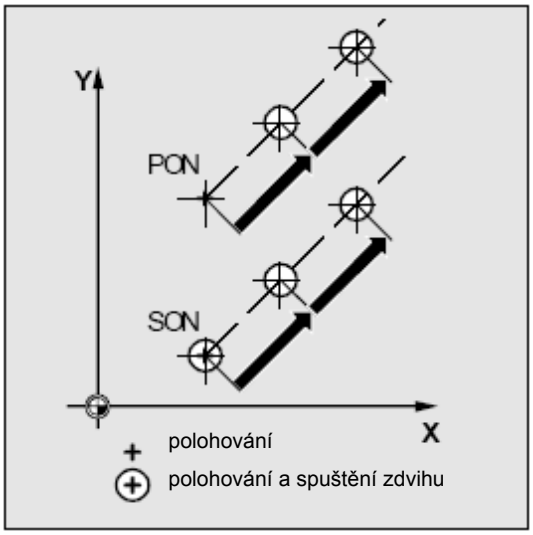
Vzdálenost děr větší než 10 mm:
Posuv bude probíhat se zrychlením 100%.

© Siemens AG 2004, Všechna práva vyhrazena

Lisování se zpožděním
Příkaz PDELAYON způsobí zpožděné spuštění lisovacího zdvihu. Tento příkaz s modální platností má přípravnou funkci a proto se zpravidla nachází před příkazem PON.
Po příkazu PDELAYOF probíhá lisování opět normálně.

Spouštění zdvihu

Spuštění prvního zdvihu
Spouštění prvního zdvihu po aktivování funkce probíhá u lisování a prostřihování časově odlišně:



PON/PONS
- Všechny zdvihy – včetně prvního bloku po aktivování – se provádějí na konci bloku.

SON/SONS
- První zdvih po aktivování prostřihování se uskutečňuje už na začátku bloku.
- Všechny ostatní zdvihy jsou spouštěny vždy na konci bloku.

Lisování a prostřihování na místě
Zdvih se spouští jen tehdy, pokud blok obsahuje informace o posuvu pro osy lisovacího nebo prostřihovacího zařízení (osy aktivní roviny). Pokud byste si přesto přáli spustit zdvih na tomtéž místě, naprogramujte lisovací/prostřihovací osu s nulovou drahou.

Další upozornění

Práce s otočnými nástroji
Aby se otočné nástroje nastavovaly tangenciálně na naprogramovanou dráhu, použijte tangenciální řízení.

© Siemens AG 2004, Všechna práva vyhrazena

12.1.2 Použití M-příkazů



Pokud použijete techniku maker, můžete namísto příkazů jazyka používat M-funkce:

`DEFINE M22 AS SON`	Aktivování prostřihování
`DEFINE M122 AS SONS`	Aktivování prostřihování se zaváděním
`DEFINE M25 AS PON`	Aktivování lisování
`DEFINE M125 AS PONS`	Aktivování lisování se zaváděním
`DEFINE M26 AS PDELAYON`	Aktivování lisování se zpožděním
`DEFINE M20 AS SPOF`	Deaktivování lisování/prostřihování
`DEFINE M23 AS SPOF`	Deaktivování lisování/prostřihování

© Siemens AG 2004, Všechna práva vyhrazena

12.2 Automatické rozdělení dráhy



Programování

SPP=
SPN=



Vysvětlení

SPP	Velikost segmentu dráhy (maximální vzdálenost zdvihů), modální
SPN	Počet úseků dráhy na jeden blok, bloková platnost



Funkce

Rozdělení na segmenty

Když je aktivní lisování nebo prostřihování, příkazy SPP a SPN způsobí, že celková dráha posuvu naprogramovaná pro osy bude rozdělena na daný počet dráhových úseků stejné délky (ekvidistantní segmentace dráhy). Každý dráhový segment interně odpovídá jednomu bloku.

Počet zdvihů

Při lisování se první zdvih uskutečňuje v koncovém bodě prvního dráhového segmentu, při prostřihování oproti tomu na začátku prvního dráhového segmentu.
Z celkové dráhy potom tedy vyplývají následující údaje:
Lisování:

Počet zdvihů = počet dráhových segmentů

Prostřihování:

Počet zdvihů = počet dráhových segmentů + 1

Pomocné funkce

Pomocné funkce se uskutečňují v prvním vygenerovaném bloku.

© Siemens AG 2004, Všechna práva vyhrazena

12.2.1 Rozdělení dráhy u dráhových os

Postup

Délka dráhového segmentu, SPP
Pomocí SPP zadáváte maximální vzdálenost zdvihů a tedy také maximální délku dráhového segmentu, na které má být celková dráha posuvu rozdělena.

Deaktivování příkazu se provádí příkazy SPOF nebo SPP = 0.

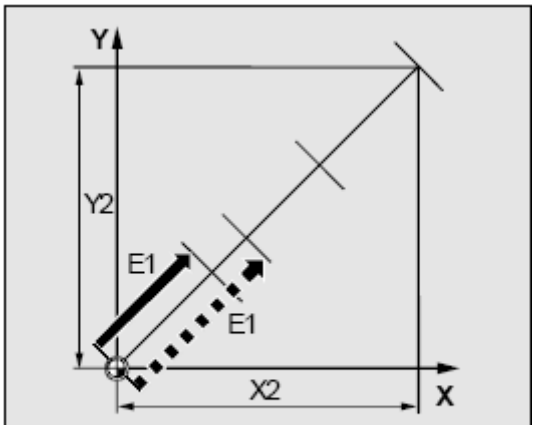
Příklad:
N10 G1 SON X0 Y0
N20 **SPP=2** X10

Celková dráha posuvu 10 mm bude rozdělena na 5 dráhových segmentů po 2 mm (SPP=2).

Při použití příkazu SPP je dráha vždy rozdělena na stejné úseky. To znamená, že naprogramovaná velikost dráhových segmentů (hodnota SPP) se bude uplatňovat jen tehdy, je-li podíl celkové dráhy posuvu a hodnoty SPP celé číslo.
Pokud tomu tak není, bude interně velikost dráhového segmentu zmenšena tak, aby byl výše zmiňovaný podíl celočíselný.

Příklad:
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15

Při celkové dráze posuvu 15 mm a délce dráhového segmentu 3,5 mm vyplývá podíl, který není celočíselný (4,28).
V důsledku toho se provede zmenšení hodnoty SPP na nejbližší možný celočíselný podíl.
V tomto případě vyplývá délka dráhového segmentu 3 mm.



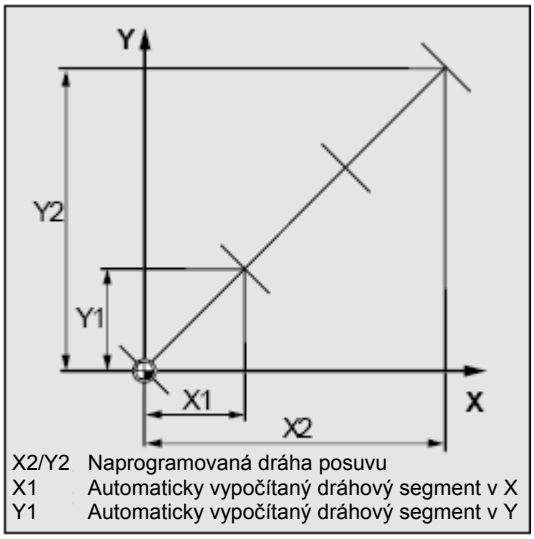
X2/Y2 Naprogramovaná dráha posuvu (blok prostřihování nebo lisování)
E1 Naprogramovaná délka dráhového segmentu
E2 Automaticky zaokrouhlená délka segmentu

Počet dráhových úseků, SPN
Pomocí příkazu SPN definujete počet dráhových segmentů, které mají být vytvořeny z celkové dráhy posuvu. Délka dráhových segmentů bude vypočítána automaticky.

Protože příkaz SPN má blokovou platnost, musí být předtím aktivováno lisování nebo prostřihování pomocí příkazů PON nebo SON.

SPP a SPN v jednom bloku
Pokud v jednom bloku naprogramujete jak délku dráhového segmentu (SPP), tak i počet dráhových segmentů (SPN), platí pro tento blok SPN, pro všechny ostatní bloky SPP.

Pokud byl příkaz SPP aktivován už někdy před příkazem SPN, bude po bloku se SPN znovu v platnosti.



X2/Y2 Naprogramovaná dráha posuvu
X1 Automaticky vypočítaný dráhový segment v X
Y1 Automaticky vypočítaný dráhový segment v Y

Další upozornění

Pokud jsou prostřihování/lisování ve Vašem řídícím systému k dispozici, můžete při programování používat příkazy SPN, příp. SPP pro automatické rozdělení dráhy používat i nezávisle na této technologii.

12.2.2 Rozdělení dráhy u samostatných os

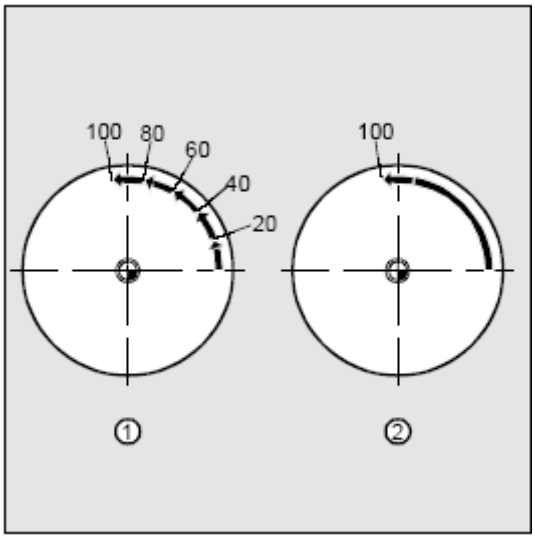
Pokud jsou vedle dráhových os definovány také samostatné osy použité jako osa pro lisování nebo prostřihování, mohou i ony podléhat automatické segmentaci dráhy.

Chování samostatné osy v případě SPP
Naprogramovaná délka dráhového segmentu se vztahuje v zásadě na dráhové osy. Z tohoto důvodu v bloku, v němž není vedle pohybu samostatné osy a hodnoty SPP naprogramována žádná dráhová osa, bude hodnota SPP ignorována.

© Siemens AG 2004, Všechna práva vyhrazena

Jestliže jsou v bloku naprogramovány jak dráhová, tak i samostatná osa, řídí se chování samostatné osy podle nastavení v příslušných strojních parametrech.

1. Standardní nastavení
Dráha samostatné osy bude rovnoměrně rozdělena na pomocné bloky vytvořené podle SPP.



Příklad:
N10 G1 SON X10 A0
N20 SPP=3 X25 A100

V důsledku naprogramované vzdálenosti mezi zdvihy o hodnotě 3 mm vznikne při celkové dráze posuvu 15 mm v ose X (dráhová osa) 5 bloků. Osa A se proto v každém bloku pootočí o 20°.

2. Samostatná osa bez rozdělení dráhy
Samostatná osa urazí celou svou dráhu v prvním vytvořeném bloku.

3. Odlišné rozdělení dráhy
Chování samostatné osy je závislé na interpolaci dráhové osy:

- Kruhová interpolace: rozdělení dráhy
- Lineární interpolace: žádné rozdělení dráhy

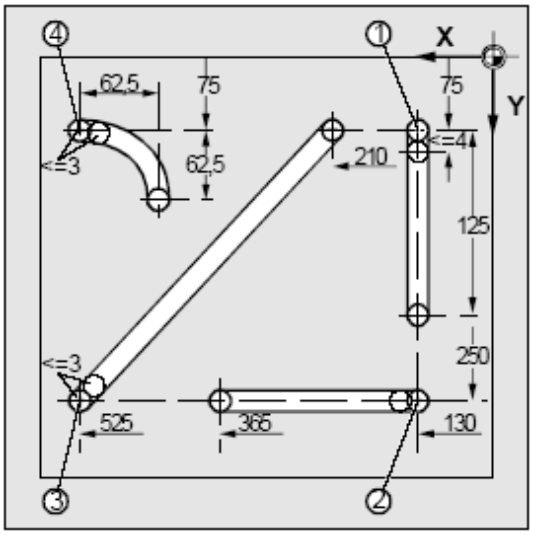
Chování v případě SPN
Naprogramovaný počet dráhových segmentů platí i tehdy, pokud není současně naprogramována žádná dráhová osa.
Předpoklad: Samostatná osa je definována jako osa pro lisování/prostřihování.

© Siemens AG 2004, Všechna práva vyhrazena

12.2.3 Příklady programování

Příklad programování 1

Naprogramované úseky vyráběné prostřihováním mají být automaticky rozděleny na stejně velké dráhové segmenty.



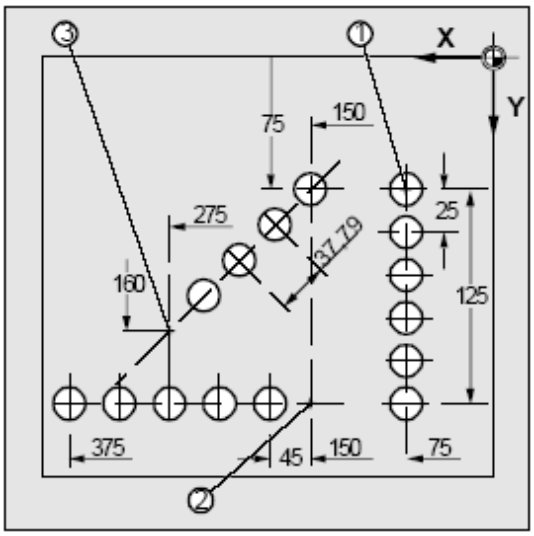
Úsek programu

`N100 G90 X130 Y75 F60 SPOF`	Najíždění na počáteční bod 1
`N110 G91 Y125 SPP=4 SON`	Aktivování prostřihování; maximální délka úseku dráhy pro automatické rozdělení je 4 mm
`N120 G90 Y250 SPOF`	Deaktivování prostřihování; najíždění na počáteční bod 2
`N130 X365 SON`	Aktivování prostřihování; maximální délka úseku dráhy pro automatické rozdělení je 4 mm
`N140 X525 SPOF`	Deaktivování prostřihování; najíždění na počáteční bod 3
`N150 X210 Y75 SPP=3 SON`	Aktivování prostřihování; maximální délka úseku dráhy pro automatické rozdělení je 3 mm
`N160 X525 SPOF`	Deaktivování prostřihování; najíždění na počáteční bod 4
`N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON`	Aktivování prostřihování; maximální délka úseku dráhy pro automatické rozdělení je 3 mm
`N180 G00 G90 Y300 SPOF`	Deaktivování prostřihování

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování 2

Pro jednotlivé řady děr má být provedeno automatické rozdělení dráhy. Pro rozdělení se vždy udává maximální délka dráhového segmentu (hodnota SPP).



Úsek programu

`N100 G90 X75 Y75 F60 PON`	Najíždění na počáteční bod 1, aktivování lisování, lisování jedné díry.
`N110 G91 Y125 SPP=25`	Maximální délka dráhového segmentu pro automatické rozdělení dráhy: 25 mm
`N120 G90 X150 SPOF`	Deaktivování lisování; najíždění na počáteční bod 2.
`N130 X375 SPP=45 PON`	Aktivování lisování; maximální délka dráhového segmentu pro automatické rozdělení dráhy: 45 mm
`N140 X275 Y160 SPOF`	Deaktivování lisování; najíždění na počáteční bod 3.
`N150 X150 Y75 SPP=40 PON`	Aktivování lisování; namísto naprogramované délky dráhového segmentu 40 mm se použije vypočítaná délka dráhového segmentu 37,79 mm
`N160 G00 Y300 SPOF`	Deaktivování lisování, polohování

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Další funkce

13.1 Osové funkce AXNAME, SPI, ISAXIS, AXSTRING (od SW 6) 13-514

13.2 Volání funkce ISVAR ( ) (od SW 6.3) 13-516

13.3 Zadávání kompenzačních charakteristik:QECLRNON, QECLRNOF 13-518

13.4 Synchronní vřeteno 13-520

13.5 EG: Elektronická převodovka (od SW 5) 13-530
13.5.1 Definice elektronické převodovky: EGDEF 13-531
13.5.2 Aktivování elektronické převodovky 13-532
13.5.3 Deaktivování elektronické převodovky 13-535
13.5.4 Vymazání definice elektronické převodovky 13-536
13.5.5 Otáčkový posuv (G95) / elektronická převodovka (od SW 5.2) 13-536
13.5.6 Chování EG při Power ON, resetu, změně provozního režimu, hledání 13-537
13.5.7 Systémové proměnné elektronické převodovky 13-537

13.6 Rozšířené zastavování a stahování (od SW 5) 13-538
13.6.1 Reakce závislé na pohonu 13-539
13.6.2 Reakce ovládané NC systémem 13-540
13.6.3 Možné zdroje spouštění 13-544
13.6.4 Logické spouštěcí funkce: Obsluha zdrojů/reakcí 13-545
13.6.5 Aktivování 13-546
13.6.6 Obsluha generátoru / záloha pomocného obvodu 13-546
13.6.7 Zastavení závislé na pohonu 13-547
13.6.8 Zpětný pohyb závislý na pohonu 13-548
13.6.9 Příklad: Použití reakce závislé na pohonu 13-549

13.7 Linková komunikace (od SW 5.2) 13-550

13.8 Zásobník os (od SW 5.2) 13-553

13.9 Strojový čas programu/počítadlo obrobků (od SW 5.2) 13-555
13.9.1 Strojový čas programu 13-555
13.9.2 Počítadlo obrobků 13-557

13.10 Příkaz pro interaktivní vyvolání okna z výrobního programu: MMC (od SW 4.4) 13-559

13.11 Ovlivňování vedení pohybu 13-560
13.11.1 Procentuální korekce trhavých pohybů: JERKLIM 13-560
13.11.2 Procentuální korekce rychlosti: VELOLIM 13-561

13.12 Spojení Master/Slave 13-562

© Siemens AG 2004, Všechna práva vyhrazena

13.1 Osové funkce AXNAME, SPI, ISAXIS, AXSTRING (od SW 6)



Programování

```
AXNAME(„PLANACHSE“)
AX[AXNAME(„String“)]
AXSTRING (SPI(n))
SPI(n)
ISAXIS(číslo geometrické osy)
```



Vysvětlení příkazů

AXNAME	Převádí vstupní řetězec na identifikátor osy; vstupní řetězec musí obsahovat platný název osy.
SPI	Převádí číslo vřetena na identifikátor osy; předávaný parametr musí obsahovat platné číslo vřetena
n	Číslo vřetena
AXSTRING	Do SW 5 byl index osy, která byla přiřazena vřetenu vypisován jako číslo vřetena. Od SW 6 se vypisuje řetězec s přiřazeným číslem vřetena.
AX	Proměnný identifikátor osy.
ISAXIS	Kontrola, zda uvedená geometrická osa existuje.



Funkce

Příkaz AXNAME se používá např. při sestavování všeobecně použitelných cyklů, kdy názvy os nejsou známé (viz také kapitola 13.10 „Řetězcové funkce“).

Příkaz SPI se používá pokud se pro vřeteno používají osové funkce, např. pro synchronizaci vřetena.

Příkaz ISAXIS se používá v cyklech se všeobecnou platností, aby bylo zabezpečeno, že určitá geometrická osa je k dispozici a že tedy následující volání $P_AXNX nebude přerušeno chybou.

(od SW 6):

Rozšíření SPI(n)

Funkce SPI(n) se nyní může používat i pro čtení a zápis složek framu. Díky tomu je možné zapisovat framy např. se syntaxí $P_PFRAME[SPI(1),TR]=2.22

Dodatečným naprogramováním poloh osy pomocí adresy AX[SPI(1)]=<pozice osy> je možné osou pohybovat.

© Siemens AG 2004, Všechna práva vyhrazena

Odstraňování chyby pro `AXSTRING (SPI(n))`
Při naprogramování příkazu `AXSTRING (SPI(n))` se až do SW 5 vypisovalo číslo vřetena, které bylo ose přiřazeno.
Příklad:
Vřeteno 1 je přiřazeno 5. ose `($MA_SPIND_ASSIGN_TO_MACHAX[AX5]=1)`, `AXSTRING(SPI(1))` tedy vrací nesprávný řetězec „`S4`".

Od SW 6 se příkazem `AXSTRING[SPI(n)]` vypisuje řetězec „`Sn`".
Příklad:
`AXSTRING( SPI(2) )` vrací řetězec „`S2`".



Příklad programování

Má se pohybovat osa, která je definována jako osa příčná.

`OVRA[AXNAME("Planachse")]=10`	příčná osa
`AX[AXNAME("Planachse")]=50.2`	koncová pozice pro příčnou osu
`OVRA[SPI(1)]=70`	override pro vřeteno 1
`IF ISAXIS(1)==FALSE GOTOF WEITER`	Existuje abscisa?
`AX[$P_AXN1]=100`	posuv abscisy
`WEITER:`	

© Siemens AG 2004, Všechna práva vyhrazena

13.2 Volání funkce ISVAR ( ) (od SW 6.3)



Programování

```
ISVAR („Identifikátor proměnné“)
ISVAR (Identifikátor, [Hodnota,Hodnota])
```



Vysvětlení příkazů

`Identifikátor proměnné`	Předávaný parametr typu String může být buď bezrozměrný, jednorozměrný nebo dvojrozměrný.
`Identifikátor`	Identifikátor je proměnná známá NC systému s indexem pole nebo i bez něj jako strojní nebo nastavovaný parametr nebo systémová proměnná nebo všeobecná proměnná.
`Hodnota`	Hodnota funkce typu BOOL.

Struktura
Předávaný parametr může mít následující strukturu:

1. Bezrozměrná proměnná:
   Identifikátor
2. Jednorozměrná proměnná bez indexu pole:
   Identifikátor [ ]
3. Jednorozměrná proměnná s indexem pole:
   Identifikátor [hodnota]
4. Dvourozměrná proměnná bez indexu pole:
   Identifikátor [,]
5. Dvourozměrná proměnná s indexem pole:
   Identifikátor [hodnota,hodnota]



Funkce

Příkaz ISVAR je funkcí ve smyslu NC jazyka s:

- Funkční hodnotou typu BOOL
- Předávaným parametrem typu STRING

Výsledkem příkazu ISVAR je hodnota TRUE, pokud předávaný parametr obsahuje proměnnou, která je NC systému známá (strojní parametr, nastavovaný parametr, systémová proměnná, všeobecná proměnná, jako třeba GUD).

© Siemens AG 2004, Všechna práva vyhrazena

Kontroly

V souladu s předávaným parametrem se provádějí následující kontroly:
Je identifikátor k dispozici?
Jedná se o jedno- nebo dvojrozměrné pole?
Je povolen index pole?
Pouze tehdy, pokud jsou odpovědi na všechny tři otázky kladné, je vracena hodnota TRUE. Pokud byť jen jedna z kontrol není splněna nebo vyskytne-li se syntaktická chyba, je vracena hodnota FALSE. Osové proměnné jsou jako index pro názvy os akceptovány, nejsou však blíže kontrolovány.

Příklady:

```
DEF INT VAR1
DEF BOOL IS_VAR=FALSE                ; předávaný parametr je všeobecná proměnná
N10 IS_VAR=ISVAR("VAR1")             ; IS_VAR je v tomto případě TRUE
DEF REAL VARARRAY[10,10]
DEF BOOL IS_VAR=FALSE                ; různé varianty syntaxe
N20 IS_VAR=ISVAR("VARARRAY[,]")      ; IS_VAR je TRUE s dvojrozměrným polem
N30 IS_VAR=ISVAR("VARARRAY")         ; IS_VAR je TRUE, proměnná existuje
N40 IS_VAR=ISVAR                     ; IS_VAR je FALSE, index pole není možný

      ("VARARRAY[8,11]")
N50 IS_VAR=ISVAR("VARARRAY[8,8")     ; IS_VAR je FALSE, syntaktická chyba kvůli
                                     chybějící „]“.
N60 IS_VAR=ISVAR("VARARRAY[,8]")     ; IS_VAR je TRUE, index pole je dovolen
N70 IS_VAR=ISVAR("VARARRAY[8,]")     ; IS_VAR je TRUE

DEF BOOL IS_VAR=FALSE                ; předávaný parametr je strojní parametr
N100 IS_VAR=ISVAR                    ; IS_VAR je TRUE
     ("$MC_GCODE_RESET_VALUES[1]"

DEF BOOL IS_VAR=FALSE                ; předávaný parametr je systémová proměnná
N10 IS_VAR=ISVAR("$P_EP")            ; IS_VAR je v tomto případě TRUE
N10 IS_VAR=ISVAR("$P_EP[X]")         ; IS_VAR je v tomto případě TRUE
```

© Siemens AG 2004, Všechna práva vyhrazena

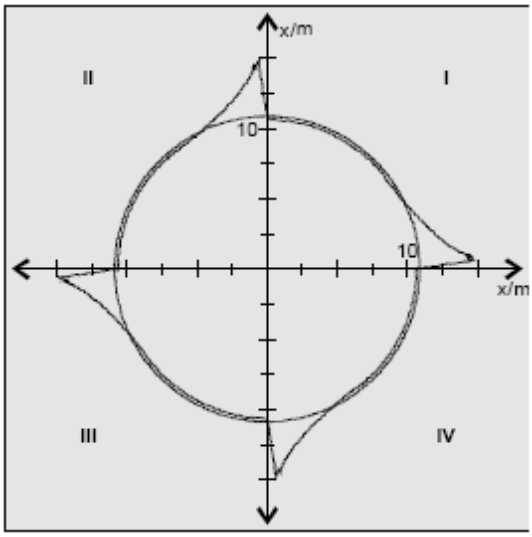
13.3 Zadávání kompenzačních charakteristik: QECLRNON, QECLRNOF

Vysvětlení příkazů

`QECLRNON (Osa.1,…4)`	Aktivování funkce „Učení kompenzace kvadrantové chyby“
`QECLRNOF`	Deaktivování funkce „Učení kompenzace kvadrantové chyby“

Funkce

Kompenzace kvadrantové chyby (QFK) snižuje chybu kontury, která vzniká při převrácení směru pohybu v důsledku mechanických nelinearit (např. tření, ztráty) nebo torze.



Optimální kompenzační údaje mohou být přizpůsobeny na základě neurální sítě řídícího systému v průběhu fáze učení a tak mohou být automaticky zjištěny kompenzační charakteristiky.

Učení může probíhat pro až 4 osy současně.

Postup

Pohyby posuvu os, které jsou zapotřebí pro učení, jsou generovány pomocí NC programu. Učební pohyby jsou v něm uloženy ve formě výukového cyklu.

První učení

Vzorové učební NC programy pro učební pohyby a pro dosazení systémových proměnných QFK pro první učební proces při uvádění do provozu jsou obsaženy na disketě základního programu PLC:

QECLRN.SPF	Učební cyklus
QECDAT:MPF	Vzorový NC program pro dosazení systémových proměnných a pro nastavení parametrů učebního cyklu
QECTEST.MPF	Vzorový NC program pro kruhový test.

© Siemens AG 2004, Všechna práva vyhrazena

Dodatečné učení

Dodatečná optimalizace již naučených charakteristik je možná pomocí „dodatečného učení“. Jako základ pro tuto optimalizaci se používají data, jež byla dříve uložena do uživatelské paměti.

Pro účely dodatečného učení je třeba přizpůsobit Vaši požadavkům vzorový NC program.
Pro „dodatečné učení je zapotřebí v případě potřeby změnit parametry učebního cyklu (např. QECLRN.SPF):

- Nastavte „režim učení“ = 1
- „Počet učebních průchodů“ v případě potřeby snižte.
- V případě nutnosti aktivujte „modulární učení“ a definujte hranice příslušných oblastí.

Aktivování učebního procesu:QECLRNON

Vlastní operace učení se v NC programu aktivuje příkazem QECLRNON a zadáním osy:

```
QECLRNON (X1, Y1, Z1, Q)
```

Jestliže je aktivní pouze tento příkaz, budou charakteristiky změněny.

Deaktivování učebního procesu: QECLRNOF

Jakmile jsou učební pohyby požadovaných os ukončeny, proces učení se pro všechny osy současně deaktivuje příkazem QECLRNOF.

© Siemens AG 2004, Všechna práva vyhrazena

13.4 Synchronní vřeteno

Programování

```
COUPDEF (FS,LS,Ü_FS,Ü_LS,Chování při změně bloku,Vazba)
COUPDEL (FS,LS)
COUPRES (FS,LS)
COUPON (FS,LS,PS_FS)
COUPOF (FS,LS,POS_FS,POS_LS)
WAITC (FS, Chování při změně bloku,LS, Chování při změně bloku)
```

Vysvětlení příkazů

COUPDEF	Vytvoření/změna uživatelem definované vazby
COUPON	Aktivování vazby
COUPOF	Deaktivování vazby
COUPRES	Reset parametrů vazby
COUPDEL	Vymazání uživatelem definované vazby
WAITC	Čekání na splnění podmínky synchronního chodu

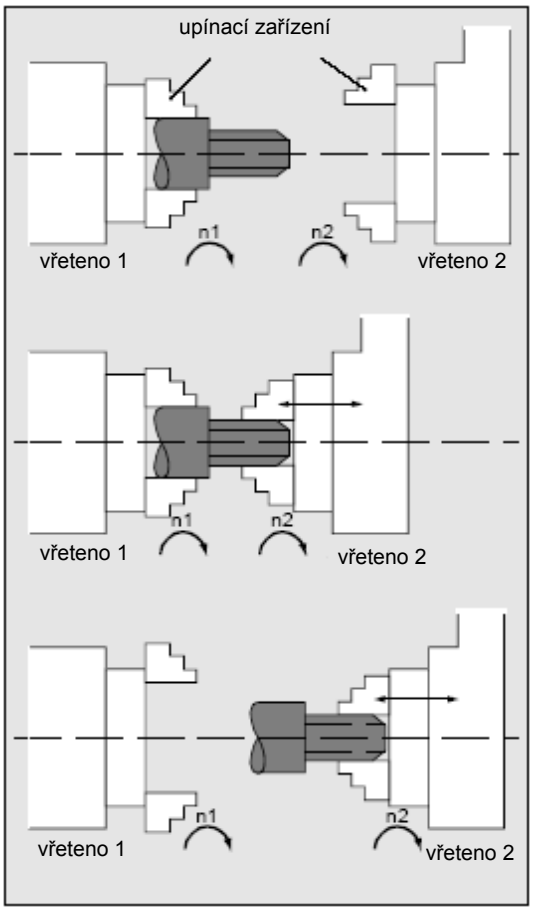
Vysvětlení parametrů

FS, LS	Označení vlečného a řídícího vřetena; zadává se číslo vřetena, např. S2
$Ü_{FS}$, $Ü_{LS}$	Převodový poměr pro vlečné a řídící vřeteno; předdefinované nastavení = 1.0; specifikace jmenovatele není nutná
Chování při změně bloku	Chování při přechodu na další blok. Přechod na další blok se uskuteční:
- „NOC“	okamžitě (předdefinované nastavení
- „FINE“	při „synchronním chodu jemném“
- „COARSE“	při „synchronním chodu hrubém“
- „IPOSTOP“	při IPOSTOP (tzn. po náběhu synchronizace podle požadované hodnoty
Vazba	Druh vazby: Vazba mezi FS a LS
- „DV“	Spojení požadovanou hodnotou (předdefinované nastavení)
- „SV“	Spojení skutečnou hodnotou
PS_{FS}	Úhlové posunutí mezi řídícím a vlečným vřetenem
POS_{FS}, POS_{LS}	Vypínací pozice pro vlečné a řídící vřeteno

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

V synchronním režimu existuje řídící vřeteno (LS) a vlečné vřeteno (FS), které tvoří tzv. **dvojici synchronních vřeten**. Když je vazba aktivní (synchronní režim) sleduje vlečné vřeteno pohyby řídícího vřetena v souladu s definovanou funkční závislostí.

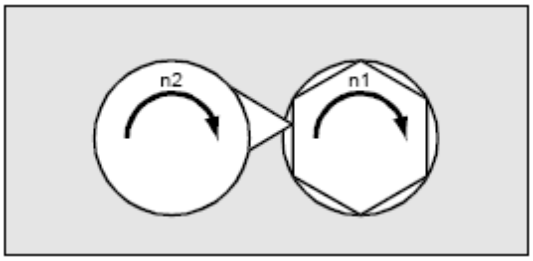


Tyto funkce na soustruzích umožňují průběžné předávání obrobku, když jsou vřeteno 1 i vřeteno 2 spuštěné, např. kvůli obrobení druhé strany, aby se odstranily ztrátové časy potřebné pro uvolnění a opětovné upnutí obrobku.

Předávání obrobku se může provádět:

- Se synchronními otáčkami ($n_{FS} = n_{LS}$)
- S polohovou synchronizací ($\varphi_{FS} = \varphi_{LS}$)
- S polohovou synchronizací s úhlovým posunutím ($\varphi_{FS} = \varphi_{LS} + \Delta\varphi$)

Mezi hlavním vřetenem a nástrojovým vřetenem může být definován ještě poměr rychlostí $K_Ü$, aby bylo možné obrábět mnohohrany (soustružení mnohoúhelníků).



Dvojice synchronních vřeten mohou být pro každý stroj konfigurovány také pomocí strojních parametrů specifických pro kanály nebo je možné ji definovat uživatelsky pomocí výrobního programu CNC. V každém kanálu NC je možné současně provozovat až dva páry synchronních vřeten.

© Siemens AG 2004, Všechna práva vyhrazena

Postup

Definice dvojice synchronních vřeten: možnosti

Konfigurovaná vazba:
Řídící a vlečné vřeteno jsou definovány pomocí strojních parametrů.
V případě této vazby nemohou být stanovené strojní parametry pro FS a LS z výrobního programu NC nijak měněny. Parametry vazby však mohou být v NC programu dosazeny pomocí příkazu COUPDEF (předpoklad: není aktivní ochrana proti zápisu).

Uživatelem definovaná vazba:
Pomocí příkazu COUPDEF mohou být ve výrobním NC programu vazby upravovány a vytvářeny nové. Jestliže má být definován nový vazební vztah, musí být v případě potřeby již dříve existující uživatelská vazba vymazána příkazem COUPDEL.

Definice nové vazby, COUPDEF

Dosazování parametrů do již vytvořeného podprogramu je vysvětleno v dalších odstavcích:
COUPDEF (FS,LS,$Ü_{FS}$,$Ü_{LS}$,Chování při změně bloku,Vazba)

Vlečné a řídící vřeteno: FS a LS

Identifikátory os FS a LS je vazba jednoznačně určena.
Při každém příkazu COUP musí být naprogramovány. Další parametry vazby musí být zadány jen tehdy, pokud je zapotřebí je změnit (modální platnost).

Příklad:
`N... COUPDEF (S2,S1,`$Ü_{FS}$`,`$Ü_{LS}$`)`
Znamená:
S2 = vlečné vřeteno, S1 = řídící vřeteno

© Siemens AG 2004, Všechna práva vyhrazena

Polohování vlečného vřetena: možnosti
Když je synchronní vazba vřeten aktivována, můžete vlečné vřeteno nastavovat do požadované polohy v rozsahu ± 180° také pohybem, který je nezávislý na řídícím vřetenu.

Polohování, SPOS
Vlečné vřeteno může být interpolováno příkazem SPOS.

Další informace o příkazu SPOS naleznete v Příručce programování – Základy.

Příklad:
```
N30 SPOS[2]=IC(-90)
```

FA, ACC, OVRA:
rychlost, zrychlení
Pomocí příkazů FA[SPI(Sn)] nebo FA[Sn], ACC[SPI(Sn)] nebo ACC[Sn] a OVRA[SPI(n)] nebo OVRA[Sn] je možné naprogramovat rychlosti při polohování a hodnoty zrychlení pro vlečná vřetena (viz Příručka programování – Základy). „n“ je číslo vřetena 1 ... n.

Programovatelný přechod na další blok, WAITC
Příkazem WAITC je možné definovat chování při přechodu na další blok s odlišnými podmínkami synchronního běhu pro pokračování programu např. po změně parametrů vazby nebo po polohovacích operacích (jemné, hrubé IPOSTOP).

Příkaz WAITC způsobuje zpoždění v důsledku vložení nových bloků, dokud nejsou splněny podmínky synchronního chodu, takže synchronního stavu je možné dosáhnout rychleji.

Jestliže pro synchronní chod nejsou specifikovány žádné podmínky, platí naprogramované nebo konfigurované chování při přechodu na další blok pro odpovídající vazbu.

© Siemens AG 2004, Všechna práva vyhrazena

Příklady:

```
N200 WAITC
```

Čekání na podmínky synchronního chodu pro všechna aktivní vlečná vřetena, aniž by tyto podmínky byly udány.

```
N300 WAITC(S2,"FINE",S4,"COARSE")
```

Čekání na uvedené podmínky synchronního chodu – „jemné“ pro S2 a „hrubé“ pro S4.

Poměr rychlostí $K_Ü$

Tento přenosový poměr se udává zadáním hodnot pro FS (čitatel) a pro LS (jmenovatel).

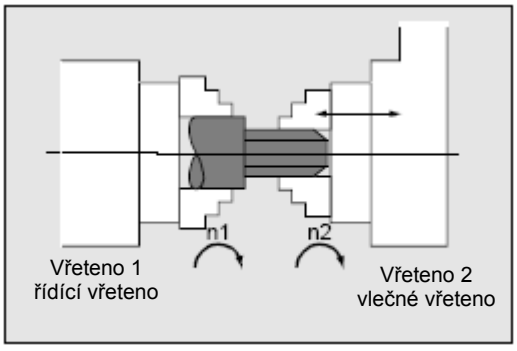
Možnosti:

- Vlečné a řídící vřeteno se otáčejí se stejnými otáčkami ($n_{FS} = n_{LS}$; $K_Ü$ kladné)
- Stejnosměrný, příp. protisměrný pohyb mezi vřeteny FS a LS ($K_Ü$ záporné)
- Vlečné a řídící vřeteno se otáčejí s různými rychlostmi ($n_{FS} = K_Ü \cdot n_{LS}$; $K_Ü \neq 0$)
  Použití: Obrábění mnohohranů

Příklad:

```
N… COUPDEF(S2, S1, 1.0, 4.0)
```

Znamená:
Vlečné vřeteno S2 a řídící vřeteno S1 se otáčejí s přenosovým poměrem 0,25.



- Musí být naprogramován alespoň čitatel. Pokud není naprogramován žádný jmenovatel, předpokládá se, že má hodnotu 1.
- Přenosový poměr je možné měnit, i když je vazba aktivována a vřetena se pohybují.

© Siemens AG 2004, Všechna práva vyhrazena

Chování při přechodu na další blok
Pomocí tohoto parametru je možné při definici vazby zvolit jednu z následujících možností a tak určit, kdy se má uskutečnit přechod na další blok:

„**NO**C“	okamžitě (předdefinované nastavení)
„**FI**NE“	při „jemném synchronním chodu“
„**CO**ARSE“	při „hrubém synchronním chodu“
„**IP**OSTOP“	při IPOSTOP (tzn. po synchronizaci podle požadované hodnoty

Pro zadání chování při přechodu na další blok je postačující zápis tučně vytištěných písmen.

Chování při přechodu na další blok má modální platnost!

Druh vazby

„DV“	Vazba požadovanou hodnotou mezi FS a LS (předdefinované nastavení)
„AV“	Vazba skutečnou hodnotou mezi FS a LS.

Druh vazby má modální platnost.



Pozor
Druh vazby se smí měnit jen v době, kdy je vazba deaktivována!

© Siemens AG 2004, Všechna práva vyhrazena

Aktivování synchronního režimu

- Nejrychlejší možné aktivování vazby s libovolným úhlovým vztahem mezi FS a LS:

```
N... COUPON (S2, S1)
```

- Aktivování s úhlovým posunutím POS_{FS}:
  Slouží pro vazbu s polohovou synchronizací pro obrábění profilovaných obrobků.
  POS_{FS} se vztahuje na polohu řídícího vřetena 0° v kladném směru otáčení.

  Rozsah hodnot POS_{FS}: 0° ... 359,999°

```
COUPON (S2, S1, 30)
```

Tímto způsobem je možné měnit úhlové posunutí, i když je vazba aktivní.

Deaktivování synchronního režimu, COUPOF
Jsou možné tři varianty:

- Nejrychlejší možné deaktivování vazby, přechod na další blok bude okamžitě uvolněn:

```
COUPOF (S2,S1)
```

- Po přejetí vypínací pozice;
  Přechod na další blok se uskuteční až po přejetí vypínací pozice POS_{FS} a případně POS_{LS}.

  Rozsah hodnot: 0° ... 359,999°

```
COUPOF (S2,S1,150)
COUPOF (S2,S1,150,30)
```


© Siemens AG 2004, Všechna práva vyhrazena

Deaktivování vazby se zastavením vlečného vřetena, COUPOFS (od SW 6.4)
Jsou možné dvě varianty:

- Nejrychlejší možné deaktivování vazby a zastavení bez udání polohy, přechod na další blok bude uvolněn okamžitě:

```
COUPOFS (S2,S1)
```

- Po přejetí naprogramované vypínací pozice vlečné osy, která je vztažena na souřadný systém stroje, se přechod na další blok uvolní až po přejetí vypínací pozice POS_{FS}:

```
COUPOFS (S2,S1,POS_FS)
```

Rozsah hodnot: 0° ... 359,999°

Vymazání vazby, COUPDEL (od SW 6.3)
Existující uživatelem definovaná vazba synchronních vřeten musí být vymazána, jestliže má být definován nový vazební vztah a pokud všechny vazby, které mohou být volně konfigurovány (1 nebo 2) jsou již obsazeny.

```
N ... COUPON (S2,S1)
```

SPI(2) = vlečné vřeteno, SPI(1) = řídící vřeteno

⚠ *Vymazání vazby je možné jen tehdy, pokud je vazba deaktivována (COUPOF).*
Napevno konfigurované vazby není možné až do SW 6.3 pomocí příkazu COUPDEL vymazat.

⚠ **Vymazání vazby, COUPDEL (od SW 6.4)**
Když je tento příkaz uplatněn na aktivní vazbu mezi synchronními vřeteny, vazba se deaktivuje a všechny její parametry budou vymazány. Vlečné vřeteno přebírá poslední otáčky a jeho chování bude odpovídat dřívějšímu po použití příkazu COUPOF (FS,LS).

© Siemens AG 2004, Všechna práva vyhrazena

Reset parametrů vazby, COUPRES

Pomocí příkazu COUPRES jsou:

- aktivovány parametry uložené ve strojních a nastavovaných parametrech (napevno konfigurovaná vazba)
- aktivována předem definovaná nastavení (uživatelem definovaná vazba)

Parametry naprogramované pomocí příkazu COUPDEF (včetně přenosového poměru) jsou přitom ztraceny.

```
N ... COUPRES (S2,S1)
```

S2 = vlečné vřeteno, S1 = řídící vřeteno

Systémové proměnné

Aktuální stav vazby, vlečné vřeteno

Pro vlečné vřeteno může být aktuální stav vazby načítán v NC programu pomocí následujících systémových proměnných:

```
$AA_COUP_ACT[FS]
```

FS = identifikátor osy pro vlečné vřeteno s číslem vřetena, např. S2.

Načtená hodnota má pro vlečné vřeteno následující význam:
0: žádná vazba není aktivní
4: je aktivní vazba synchronních vřeten

Aktuální úhlové posunutí

Aktuální offset polohy FS vůči LS vzhledem k požadované hodnotě může být v C programu načítán pomocí následujících osových systémových proměnných:

```
$AA_COUP_OFFS[S2]
```

Offset polohy vzhledem ke skutečné hodnotě lze načítat pomocí:

```
$VA_COUP_OFFS[S2]
```

FS = identifikátor osy pro vlečné vřeteno s číslem vřetena, např. S2.

© Siemens AG 2004, Všechna práva vyhrazena

Pokud byl v době, kdy byla aktivována vazba a režim vlečení, regulátor odstaven a následně znovu zapnut, bude se offset pozice v době aktivování regulátoru lišit od původní naprogramované hodnoty. V tom případě může být nový offset polohy načten a v případě potřeby opraven ve výrobním NC programu.

Příklad programování

Práce s řídícím a vlečným vřetenem.

	; řídící vřeteno = hlavní vřeteno = vřeteno 1
	; vlečné vřeteno = vřeteno 2
`N05 M3 S3000 M2=4 S2=500`	; řídící vřeteno se otáčí s 3000 ot/min, vlečné vřeteno s 500 ot/min
`N10 COUPDEF (S2, S1, 1, 1, "NOC", "Dv")`	; definice vazby; může být i konfigurováno
`...`	
`N70 SPCON`	; řídící vřeteno do režimu polohové regulace (vazba požadovanou hodnotou
`N75 SPCON(2)`	; vlečné vřeteno do režimu polohové regulace
`N80 COUPON (S2, S1, 45)`	; aktivování vazby při otáčení s offsetovou pozicí 45 stupňů
`...`	
`N200 FA [S2] = 100`	; rychlost polohování = 100 stupňů/min
`N205 SPOS[2] = IC(-90)`	; najíždění na superponovanou polohu 90° v záporném směru
`N210 WAITC(S2, "Fine")`	; čekání na „jemný“ synchronní chod
`N212 G1 X… Y… F…`	; opracování
`...`	
`N215 SPOS[2] = IC(180)`	; najíždění na superponovanou polohu 180° v kladném směru
`N220 G4 S50`	; doba prodlevy 50 otáček hlavního vřetena
`N225 FA [S2] = 0`	; aktivování konfigurované rychlosti (MD)
`N230 SPOS[2] = IC (-7200)`	; 20 otáček s konfigurovanou rychlostí v záporném směru
`...`	
`N350 COUPOF (S2, S1)`	; zrušení vazby během otáčení, S=S2=3000
`N355 SPOSA[2] = 0`	; zastavení FS v poloze 0 stupňů
`N360 G0 X0 Y0`	
`N365 WAITS(2)`	; čekání na vřeteno 2
`N370 M5`	; zastavení FS
`N375 M30`	

© Siemens AG 2004, Všechna práva vyhrazena

13.5 EG: Elektronická převodovka (od SW 5)



Úvod

Pomocí funkce „Elektronická převodovka“ je možné pohyb jedné **vlečné osy** ovládat podle lineárních pohybových bloků v závislosti na až pěti **řídících osách**. Závislosti mezi řídícími osami a vlečnou osou jsou definovány pro každou řídící osu zvlášť pomocí faktoru vazby.
Vypočítávané pohybové složky vlečné osy jsou tvořeny součtem pohybových složek jednotlivých řídících os vynásobených příslušnými faktory vazby.
Při aktivování svazku os EG může být vlečná osa synchronizována podle definované polohy.
Se svazkem převodovky je možné ve výrobním programu provádět tyto operace:

- definice
- aktivování
- deaktivování
- vymazání

Pohyby vlečné osy mohou být volitelně odvozovány:

- z požadovaných hodnot řídících os
- ze skutečných hodnot řídících os

Jako rozšíření je možné **od verze SW 6** realizovat také nelineární závislosti mezi řídícími osami a vlečnou osou prostřednictvím **tabulek křivky** (viz kapitola 9). Elektronické převodovky mohou být také řazeny kaskádovitě, tzn. vlečná osa jedné elektronické převodovky může být řídící osou pro další elektronickou převodovku.

© Siemens AG 2004, Všechna práva vyhrazena

13.5.1 Definice elektronické převodovky: EGDEF



Funkce

Svazek os elektronické převodovky může být definován zadáním vlečné osy a nejméně jedné, nejvíce však pěti řídících os s příslušným typem vazby.
EGDEF (vlečná osa, řídící osa1, typ vazby1, řídící osa2, typ vazby2,...)



Vysvětlení

`Vlečná osa`	Osa, která je ovlivňována řídícími osami
`Řídící osa1 ... řídící osa 5`	Osy, které vlečnou osu ovlivňují
`Typ vazby1 ... typ vazby 5`	Vlečné osy jsou ovlivňovány: 0: skutečnou hodnotou 1: požadovanou hodnotou odpovídající řídící osy



Programování

`EGDEF (C, B, 1, Z, 1, Y, 1)` B, Z, Y ovlivňují osu C pomocí požadované hodnoty

Typ vazby nemusí být pro všechny řídící osy stejný a tudíž je třeba jej zadávat pro každou osu zvlášť.
Faktorům vazby je při definici svazku EG předem dosazena nula. Předpoklad pro definici svazku os EG:
Pro vlečnou osu nesmí být v dané chvíli definována žádná osová vazba (v případě potřeby musí být předtím vymazána pomocí příkazu EGDEL).

Upozornění
Příkaz EGDEL způsobuje zastavení předběžného zpracování. Definici převodovky s příkazem EGDEF je možné beze změn použít i tehdy, pokud u systémů od **verze SW 6** ovlivňuje jedna nebo více řídících os vlečnou osu pomocí **tabulky křivky**.

© Siemens AG 2004, Všechna práva vyhrazena

13.5.2 Aktivování elektronické převodovky

Pro příkaz aktivování existují tři varianty:

- **Varianta 1:**

Svazek os EG se aktivuje selektivně bez synchronizace příkazem:

```
EGON(FA, „režim změny bloku“, LA1,
Z1, N1, LA2, Z2, N2, .. LA5, Z5, N5)
```

Vysvětlení

FA	Vlečná osa	
Režim změny bloku	Mohou se používat následující režimy:	
	„NOC“	Přechod na další blok se uskutečňuje okamžitě.
	„FINE“	Přechod na další blok nastane při „jemném“ synchronním chodu.
	„COARSE“	Přechod na další blok nastane při „hrubém“ synchronním chodu.
	„IPOSTOP“	Přechod na další blok nastane při synchronizaci podle požadované hodnoty.
LA1, ... LA5	Řídící osy	
Z1, ... Z5	Čitatel faktoru vazby i	
N1, ... N5	Jmenovatel faktoru vazby i Faktor vazby i = čitatel i / jmenovatel i	

Smí být naprogramovány pouze řídící osy, které byly předtím specifikovány pomocí příkazu EGDEF. Musí být naprogramována nejméně jedna řídící osa.
Pozice řídících os a vlečné osy se v okamžiku aktivování ukládají jako „synchronní pozice“. Synchronní pozice je možné načíst pomocí systémových proměnných $AA_EG_SYN.

- **Varianta 2:**

Svazek EG se aktivuje selektivně **se** synchronizací pomocí příkazu:

```
EGONSYN(FA, "Režim změny bloku", SynPosFA,[, LAi, SynPosLAi, Zi, Ni])
```


© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení

FA	Vlečná osa
Režim změny bloku	Mohou se používat následující režimy: „NOC“ Přechod na další blok se uskutečňuje okamžitě. „FINE“ Přechod na další blok nastane při „jemném“ synchronním chodu. „COARSE“ Přechod na další blok nastane při „hrubém“ synchronním chodu. „IPOSTOP“ Přechod na další blok nastane při synchronizaci podle požadované hodnoty.
[, LAi, SynPosLAi, Zi, Ni]	(Hranaté závorky se nezapisují) min. 1, max. 5 posloupností skládajících se z:
LA1, ... LA5	Řídící osy
SynPosLAi	Synchronní pozice pro i-tou řídící osu
Z1, ... Z5	Čitatel faktoru vazby i
N1, ... N5	Jmenovatel faktoru vazby i Faktor vazby i = čitatel i / jmenovatel i

- **Varianta 3:**

Svazek EG se aktivuje selektivně **se** synchronizací a se zadáním **režimu najíždění** pomocí příkazu:

```
EGONSYNE(FA, "Režim změny bloku", SynPosFA, Režim najíždění [, LAi,
SynPosLAi, Zi, Ni])
```

Vysvětlení

	Parametry odpovídají variantě 2 a kromě toho:
Režim najíždění	Mohou se používat následující režimy: „NTGT“ Najíždění na následující mezizubí s časovou optimalizací „NTGP“ Najíždění na následující mezizubí s dráhovou optimalizací „ACN“ Najíždění kruhovou osou na absolutní polohu v záporném směru „ACP“ Najíždění kruhovou osou na absolutní polohu v kladném směru „DCT“ S časovou optimalizací pro naprogramovanou synchronní polohu „DCP“ S dráhovou optimalizací pro naprogramovanou synchronní polohu

© Siemens AG 2004, Všechna práva vyhrazena

Varianta 3 se uplatňuje pouze u vlečných modulo os, které jsou spojeny s řídícími modulo osami. Časová optimalizace zohledňuje mezní hodnoty rychlosti pro vlečnou osu. Vzdálenost zubů vyplývá z 360 * Zi/Ni (ve stupních). Pro případ, že vlečná osa v okamžiku volání stojí, má dráhová optimalizace stejný výsledek jako optimalizace časová. Pokud se vlečná osa už pohybuje, při NTGP se provádí synchronizace nezávisle na aktuální rychlosti vlečné osy na následující mezizubí.

Pokud se už vlečná osa pohybuje, bude NTGT provádět synchronizaci v závislosti na aktuální rychlosti vlečné osy na následující mezizubí. Za tím účelem může dojít v příp. potřeby i k zabrždění osy.

Verze SW 6

Pokud se pro řídící osu používá tabulka křivky, musíte:

Ni	Jmenovatel faktoru lineární vazby je nastaven na 0 (jmenovatel 0 byl pro lineární vazby nepřípustný). Nulový jmenovatel řídícímu systému oznamuje, že
Zi	má být interpretováno jako číslo tabulky křivky, která se má použít. Tabulka křivky s uvedeným číslem musí být v okamžiku aktivování už definována.
LAi	Údaj řídící osy odpovídá udání řídící osy při vazbě pomocí faktoru vazby (lineární vazba).

Další informace o používání tabulek křivky a o kaskádovém řazení elektronických převodovek a jejich synchronizaci naleznete v:
/FB/, M3, Vlečení, vazba řídící hodnotou.

Smí být naprogramovány pouze řídící osy, které byly předtím specifikovány pomocí příkazu EGDEF.

© Siemens AG 2004, Všechna práva vyhrazena

Naprogramováním „synchronních pozic“ pro vlečnou osu (SynPosFA) a pro řídící osu (SynPosLA) budou definovány pozice, v nichž je spojený svazek považován za *synchronizovaný*. Pokud se elektronická převodovka při svém aktivování nenachází v synchronním stavu, najede vlečná osa na svou definovanou synchronní pozici.
Pokud jsou ve svazku os modulo osy, budou jejich hodnoty pozic redukovány pomocí funkce modulo. Tím bude zaručeno, že se bude najíždět na nejbližší možnou synchronní pozici (tzv. relativní synchronizace: např. následující mezizubí).
Jestliže pro vlečnou osu není nastaven signál rozhraní DB(30 + číslo osy), DBX 26 bit 4 „Uvolnění superpozice pro vlečnou osu", na synchronní pozici se najíždět nebude.
Místo toho se program nastaví na bloku s příkazem EGONSYN a aktivuje se alarm 16771, dokud nebude výše uvedený signál nastaven.

13.5.3 Deaktivování elektronické převodovky

Pro vypnutí aktivního svazku os EG existují tři různé možnosti.

Varianta 1:

```
EGOFS(vlečná osa)
```

Elektronická převodovka se deaktivuje. Vlečná osa bude zabrzděna do klidu. Volání spouští zastavení předběžného zpracování.

Varianta 2:

```
EGOFS(vlečná osa, řídící osa1, ... řídící osa 5)
```

Toto nastavení parametrů příkazu umožňuje **selektivní** odstranit vliv jednotlivých řídících os na pohyb vlečné osy.

Musí být zadána minimálně jedna řídící osa. Vliv uvedených řídících os na vlečnou osu se tím cíleně odstraňuje. Volání spouští zastavení předběžného zpracování.
Pokud nějaké řídící osy zůstanou aktivní, bude se vlečná osa pohybovat podle jejich vlivu dále.
Pokud jsou všechny řídící osy tímto způsobem deaktivovány, bude vlečná osa zabrzděna do klidového stavu.

© Siemens AG 2004, Všechna práva vyhrazena

Varianta 3:

`EGOFC(vlečné vřeteno)`

Elektronická převodovka bude deaktivována. Vlečné vřeteno se dále pohybuje s otáčkami/rychlostí, jež byly platné v okamžiku deaktivování. Volání spouští zastavení předběžného zpracování.

Upozornění

Tato funkce je k dispozici pouze pro vřetena.

13.5.4 Vymazání definice elektronické převodovky

Svazek os elektronické převodovky musí být napřed deaktivován podle pokynů v předešlé kapitole; pak je možné jeho definici vymazat.

`EGDEL(vlečná osa)`

Definice vazby svazku os bude vymazána. Pomocí příkazu EGDEF pak mohou být vytvářeny další svazky os, dokud není dosaženo maximálního počtu současně aktivních svazků os. Volání spouští zastavení předběžného zpracování.

13.5.5 Otáčkový posuv (G95) / elektronická převodovka (od SW 5.2)

Pomocí příkazu FPR( ) je možné od SW 5 zadávat také vlečnou osu elektronické převodovky jako osu, která určuje otáčkový posuv. Pro tento případ platí následující záležitosti:

- Posuv je závislý na požadované hodnotě rychlosti vlečné osy elektronické převodovky.
- Požadovaná rychlost se vypočítává z řídících vřeten a řídících modulo os (které nejsou dráhovými osami) a z jim přiřazených faktorů vazeb.
- Na složky rychlosti pocházející z lineárních os, příp. os, jež nejsou osami modulo, a na superponované pohyby vlečné osy se nebere zřetel.

© Siemens AG 2004, Všechna práva vyhrazena

13.5.6 Chování EG při Power ON, resetu, změně provozního režimu, hledání

Po zapnutí není žádná vazba aktivní. Vazby však zůstávají aktivní i přes reset systému a přes změnu provozního režimu.
Při vyhledávání bloku se příkazy pro aktivování, vymazání a definování elektronických převodovek neprovádějí a nejsou ani zaznamenávány, místo toho jsou přeskakovány.

13.5.7 Systémové proměnné elektronické převodovky



Pomocí systémových proměnných elektronických převodovek může výrobní program zjišťovat aktuální stavy svazků os elektronické převodovky a v případě potřeby na ně reagovat.



Další upozornění
Systémové proměnné elektronické převodovky naleznete v Příloze. Jsou označeny svým názvem, který začíná:
`$AA_EG_ ...`
nebo
`$VA_EG_ ...`

13.6 Rozšířené zastavování a stahování (od SW 5)



Funkce

Funkce „Rozšířené zastavování a stahování“ ESR (Extended Stopping and Retract) nabízí možnost reagovat na volitelné zdroje chyb pružně a tak, aby byl obrobek uchráněn. Funkce „Rozšířené zastavování a stahování“ dává k dispozici následující dílčí reakce:

- **„Rozšířené zastavení“** (závislé na pohonu, od SW 5) je definované časově zpožděné zastavení.
- **„Stahování“** (závislé na pohonu) znamená „únik“ z roviny obrábění na bezpečnou návratovou polohu. Tím se má zabránit případnému nebezpečí kolize mezi nástrojem a obrobkem.
- **„Režim generátoru“** (závislé na pohonu) V případech, kdy energie pomocného obvodu není postačující pro bezpečné stažení, je možné použít režimu generátoru. Jedná se o nezávislý režim pohonu, v němž je stejnosm. přípojce pohonu dodávána potřebná energie, aby mohl uskutečnit běžné zastavení nebo stažení v případě výpadku napájení nebo podobné události.

Od SW 6 navíc kromě toho

- **„Rozšířené zastavení“** (ovládané NC) je definované časově zpožděné zastavení pod kontrolou NC, které chrání konturu.
- **„Stahování“** (ovládané NC) znamená „únik“ z roviny obrábění na bezpečnou návratovou polohu pod kontrolou NC. Tím se má zabránit případnému nebezpečí kolize mezi nástrojem a obrobkem. Při obrábění ozubeného kola to např. znamená vyjetí z právě obráběného mezizubí.

Všechny reakce se mohou používat nezávisle na ostatních.



Další informace viz:
/FB/ M3, Spojení os a ESR.

© Siemens AG 2004, Všechna práva vyhrazena

13.6.1 Reakce závislé na pohonu



Funkce

Reakce závislé na pohonu jsou definovány pro určitou osu, tzn. každý pohon zpracovává své požadavky na zastavení/stažení v případě jejich aktivace samostatně. Interpolační, příp. dráhové spojení os při zastavování nebo stahování není uvedeno, pro osy je uplatňováno časové ovládání.
V průběhu a po provedení reakce závislé na pohonu se příslušný pohon přestává řídit příkazy NC pro uvolnění nebo posuv. Upozorňuje na to alarm „26110: Spuštěno zastavení/stažení závislé na pohonu“.

Režim generátoru
Režim generátoru je

- konfigurován: pomocí MD 37500: **10**
- uvolněn: systémová proměnná $AA_ESR_ENABLE
- aktivován: v závislosti na nastavení strojních parametrů pohonu při poklesu napětí v pomocném obvodu.

Stahování (závislé na pohonu)
Stahování závislé na pohonu je

- konfigurováno: pomocí MD 37500: **11**; čas a rychlost při stahování se provádějí pomocí MD, viz „Příklad: Použití reakce závislé na pohonu“ na konci této kapitoly
- uvolněno: systémová proměnná $AA_ESR_ENABLE
- aktivováno: systémová proměnná $AA_ESR_TRIGGER.

Zastavení (závislé na pohonu)
Zastavení závislé na pohonu je

- konfigurováno: pomocí MD 37500: **12**; čas se zadává pomocí MD
- uvolněno: systémová proměnná $AA_ESR_ENABLE
- spuštěno: systémová proměnná $AA_ESR_TRIGGER.

© Siemens AG 2004, Všechna práva vyhrazena

13.6.2 Reakce ovládané NC systémem

Funkce

Stahování

Předpoklady

- Osy jsou zvoleny příkazem POLFMASK nebo POLFMLIN.
- Pozice pro jednotlivé osy jsou definovány příkazem POLF.
- Pro jednotlivé osy jsou definovány návratové pozice pomocí příkazu POLFA (od SW 6.4).
- Časové okno v parametrech
  MD 21380: ESR_DELAY_TIME1 a
  MD 21381: ESR_DELAY_TIME2
- Je definováno spouštění v
  $AC_ESR_TRIGGER
  $AA_ESR_TRIGGER (pro jednotlivé osy)
- Smluvená reakce ESR
  MD 37500: ESR_REACTION = 21
- LFPOS z modální 46. skupiny G-kódů.

Pokud je nastavena systémová proměnná $AC_ESR_TRIGGER = 1 a pokud je v tomto kanálu konfigurována stahovací osa (tzn. MD 37500: ESR_REACTION = **21**) a pro tuto osu je nastaveno $AA_ESR_ENABLE = 1, potom se v tomto kanále aktivuje příkaz **LIFTFAST**.

Ve výrobním programu musí být pomocí příkazu POLF naprogramována návratová pozice.
Při stahování jednotlivých os pomocí příkazu POLFA (osa, typ, hodnota) musí být naprogramována hodnota a musí být dodrženy následující podmínky:

- Je nastaveno $AA_ESR_ENABLE = 1.
- POLFA(osa) musí být v okamžiku spouštění samostatnou osou.
- POLFA(typ) musí být buď typ = 1 nebo typ = 2.

Pro zpětný pohyb musí být nastaveny signály pro jeho uvolnění a ty musí zůstat beze změny.

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Příkazy **LFPOS, POLF** konfigurovaný zvedací pohyb os zvolených pomocí **POLFMASK** nebo **POLFMLIN** nahrazuje pohyb po dráze pro tyto osy stanovený ve výrobním programu.
Rozšířený zpětný pohyb (tzn. LIFTFAST/LFPOS spouštěný prostřednictvím $AC_ESR_TRIGGER) je **nepřerušitelný** a může být předčasně ukončen pouze nouzovým vypínačem. Pro zpětný pohyb je k dispozici maximálně součet časů MD 21380: ESR_DELAY_TIME1 a MD 21381: ESR_DELAY_TIME2. Po uplynutí tohoto času se i pro osu zpětného pohybu spustí rychlé brždění s následným sledováním.
Frame, který je v platnosti v okamžiku aktivování rychlého brždění, se zohledňuje.
Důležité:
Framy s otáčením ovlivňují prostřednictvím příkazu **POLF** také směr, v němž se pozvednutí provádí.
Stahování řízené NC je:

- konfigurováno: pomocí MD 37500: **21** a dvou časových údajů v MD, viz výše
- uvolněno: ($AA_ESR_ENABLE) a
- spuštěno: systémová proměnná $AC_ESR_TRIGGER, $AA_ESR_TRIGGER (pro jednotlivé osy)

Příklad pro stahování v samostatné ose

`MD 37500: ESR_REACTION[AX1] = 21`	; stahování prováděné NC
`...`	
`$AA_ESR_ENABLE[AX1] = 1`	
`POLFA(AX1,1, 20.0)`	; AX1 je přiřazena osová návratová pozice ; 20.0
`$AA_ESR_TRIGGER[AX1] = 1`	; od této chvíle začíná stahování

© Siemens AG 2004, Všechna práva vyhrazena

Programování

`POLF[geo | mach]= hodnota`	Cílová pozice návratové osy
`POLFA(Osa, Typ, Hodnota)`	Návratová pozice jednotlivé osy. Možné jsou následující zkrácené formy:
`POLFA(Osa, Typ)`	Zkrácená forma pro stažení samostatné osy
`POLFA(Osa, 0/1/2)`	Rychlé deaktivování nebo aktivování
`POLFA(Osa, 0, $AA_POLFA[Osa])`	Způsobí zastavení předběžného zpracování
`POLFA(Osa, 0)`	Bez zastavení předběžného zpracování
`POLFMASK(název osy1, název osy2,...)`	Volba os pro zpětný pohyb, osy bez vzájemné vazby
`POLFMLIN(název osy1, název osy2,...)`	Volba os pro zpětný pohyb, osy s lineární vazbou

Výstraha

Pokud je při použití zkrácené formy příkazu POLFA změněn pouze typ, musí uživatel zabezpečit, aby buď návratová pozice nebo zpětná dráha obsahovaly smysluplnou hodnotu.Po zapnutí je zapotřebí zejména nově nastavit návratovou pozici a zpětnou dráhu.

Vysvětlení příkazu

`geo | mach`	Geometrická osa nebo kanálová/strojní osa, která se stahuje.
`Osa`	Identifikátor platné samostatné osy (os SW 6.4).
`Typ`	Hodnota polohy pro samostatnou osu (od SW 6.4) je typu: 0 hodnota polohy se stává neplatnou 1 hodnota polohy je absolutní 2 hodnota polohy je inkrementální
`Hodnota`	Návratová pozice, pro geometrické osy platí WCS, jinak MCS. V případě **stejných** identifikátorů pro geometrickou a kanálovou/strojní osu se použije souřadný systém obrobku. Inkrementální polohování je přípustné. Návratová pozice typ 1 pro samostatné osy (od SW 6.4) Návratová dráha typ 2 pro samostatné osy (od SW 6.4) hodnota se bude přebírat i s typem 0. Hodnota je pak pouze označena jako neplatná a pro stahování musí být znovu naprogramována.
`POLF`	Příkaz POLF má modální platnost.
`POLFA`	Pokud osa není samostatnou osou nebo chybí typ, příp. typ = 0, pak se aktivuje příslušný alarm 26080 nebo 26081.

© Siemens AG 2004, Všechna práva vyhrazena

POLFMASK	Příkazem **POLFMASK** jsou uvedené osy uvolněny pro zpětný pohyb – bez vzájemné souvislosti mezi osami. Příkaz **POLFMASK( )** bez udání osy deaktivuje rychlé pozvednutí pro všechny osy, které byly bez souvislosti mezi osami staženy.
POLFMLIN	Příkazem **POLFMLIN** jsou uvedené osy uvolněny pro zpětný pohyb – s lineární souvislostí mezi osami. Příkaz **POLFMLIN( )** bez udání osy deaktivuje rychlé pozvednutí pro všechny osy, které byly staženy s lineární souvislostí mezi osami.
název osy i	Názvy os, které mají při LIFTFAST najíždět na své pozice definované příkazem POLF. Všechny uvedené osy se musí nacházet ve stejném souřadném systému. Než může být pomocí příkazů **POLFMASK** nebo **POLFMLIN** uvolněno rychlé pozvednutí na pevnou pozici, musí být pro zvolené osy naprogramována pozice pomocí příkazu **POLF**. Pro počáteční dosazení hodnot příkazem **POLF** nejsou k dispozici žádné strojní parametry. Při interpretaci příkazů **POLFMASK** nebo **POLFMLIN** bude generován alarm 16016, pokud příkaz **POLF** nebyl dosud naprogramován.

Pokud jsou osy postupně uvolněny příkazy POLFMASK, POLFMLIN nebo POLFMLIN, POLFMASK, platí pro příslušnou osu vždy poslední nastavení.

Pozor

*Pozice naprogramované příkazem **POLF** a aktivování příkazy **POLFMASK** nebo **POLFMLIN** jsou při spuštění programu vymazány. To znamená, že uživatel musí v každém výrobním programu hodnoty pro **POLF** a zvolené osy v příkazech **POLFMASK**, příp. **POLFMLIN** znovu naprogramovat.*

Další informace o změnách souřadného systému, práci s modulo osami atd. naleznete v popisu funkcí M3.

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Zastavení

Postup pro rozšířené zastavení (prováděné NC) je dán oběma strojními parametry MD 21380: ESR_DELAY_TIME1 a MD 21381: ESR_DELAY_TIME2. Po dobu trvání MD 21380 provádí osa nerušeně interpolaci, jak bylo naprogramováno. Po uplynutí doby MD 21380 se spouští interpolačně řízené brždění.
Pro interpolačně řízené brždění je k dispozici jen časová lhůta MD 21381, po jejím uplynutí se aktivuje rychlé brždění a následné sledování.
Zastavení řízení NC je

- konfigurováno: pomocí MD 37500: **22** a dvěma časovými údaji v MD, viz výše
- uvolněno: ($AA_ESR_ENABLE)
- spuštěno: systémová proměnná $AC_ESR_TRIGGER,
  pro samostatné osy s $AA_ESR_TRIGGER.

Příklad zastavení samostatné osy

```
MD 37500: ESR_REACTION[AX1] = 22      ; zastavení řízené NC
MD 21380: ESR_DELAY_TIME1[AX1] =0.3
MD 21381: ESR_DELAY_TIME2[AX1] =0.06
...
$AA_ESR_ENABLE[AX1] = 1
$AA_ESR_TRIGGER[AX1] = 1              ; od tohoto okamžiku začíná zastavování
```

13.6.3 Možné zdroje spouštění

Funkce

Pro spouštění „rozšířeného zastavování a stahování jsou možné následující zdroje chybových stavů:

- Všeobecné zdroje (NC-externí/globální nebo specifické pro BAG/kanál):
  - Digitální vstupy (např. na modulu NCU, příp. na svorkovém bloku), příp. mapování digitálních výstupů v rámci řídícího systému ($A_IN, $A_OUT).

© Siemens AG 2004, Všechna práva vyhrazena

- Stav kanálu ($AC_STAT)
- Signály VDI ($A_DBB)
- Součtová hlášení od určitého počtu alarmů ($AC_ALARM_STAT)

- Axiální zdroje:
  - Nouzová návratová prahová hodnota vlečné osy (synchronní chod elektronické vazby, $VA_EG_SYNCDIFF [vlečná osa]
  - Pohon: Prahová hodnota výstrahy pomocného obvodu (ohrožující podpětí) $AA_ESR_STAT[osa]
  - Pohon: Prahová hodnota minimálních otáček generátoru (není už k dispozici žádná využitelná rotační energie), $AAESR_STAT[osa].

13.6.4 Logické spouštěcí funkce: Obsluha zdrojů/reakcí



Funkce

Používají se pružné možnosti spouštění statických synchronních vazeb, aby se s jejich pomocí na základě zdrojů aktivovaly příslušné reakce. Spojování všech relevantních zdrojů pomocí statických synchronních akcí je v rukou uživatele. Může vyhodnocovat systémové proměnné zdrojů jako celek nebo také selektivně pomocí bitových map a s nimi svázat požadované reakce. Statické synchronní akce jsou v platnosti ve všech provozních režimech.
Podrobný popis použití synchronních akcí viz:
Literatura:
/FBSY/ Popis funkcí, Synchronní akce.

© Siemens AG 2004, Všechna práva vyhrazena

13.6.5 Aktivování



Uvolnění funkce:
$AA_ESR_ENABLE
Funkce režimu generátoru, zastavování a stahování se uvolňují nastavením příslušných řídících signálů ($AA_ESR_ENABLE). Tyto řídící signály mohou být měněny synchronními akcemi.

Spuštění funkce (společné spuštění všech uvolněných os)
$AN_ESR_TRIGGER

- Režim generátoru se aktivuje v pohonu „automaticky“, kdykoli je zjištěno nebezpečně nízké napětí v pomocném obvodu.
- Zastavení a/nebo stahování spouštěné pohonem se aktivuje při rozpoznání výpadku komunikace (mezi NC a pohonem), jakož i při rozpoznání nebezpečně nízkého napětí v pomocném obvodu pohonu (předpokládá se konfigurace a uvolnění).
- Zastavení a/nebo stahování spouštěné pohonem mohou být spouštěny také ze strany NC nastavením odpovídajících řídících signálů $AN_ESR_TRIGGER (společný příkaz pro všechny pohony).

13.6.6 Obsluha generátoru / záloha pomocného obvodu



Funkce

Konfigurací MD pohonu a odpovídajícím naprogramováním odpovídajících statických synchronních akcí ($AA_ESR_ENABLE) mohou být kompenzovány krátkodobé výpadky napětí pomocného obvodu. Čas, který může být přemostěn, závisí na uložené energii generátoru, který bude použit pro zálohování pomocného obvodu, a na energii, která je zapotřebí pro udržení právě probíhajících pohybů (záloha pomocného obvodu a monitorování mezních otáček generátoru).

© Siemens AG 2004, Všechna práva vyhrazena

Při napětí pomocného obvodu pod dolní prahovou hodnotu přechází příslušná osa/vřeteno z režimu polohové nebo otáčkové regulace do režimu generátoru. Brzděním pohonu (předem definovaná požadovaná hodnota otáček = 0) se energie dodává zpátky do pomocného obvodu.

Další informace viz:
/FB/ M3, Vlečení, spojení řídící hodnotou

13.6.7 Zastavení závislé na pohonu

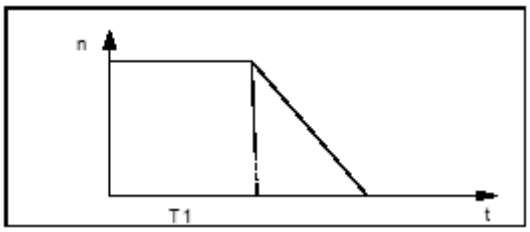
Funkce

Pohony dříve vytvořeného svazku mohou být pomocí časově řízeného zpoždění vypnutí zastaveny s co možno nejmenší odchylkou vůči sobě, jestliže to není možné zajistit ze strany řídícího systému.
Zastavení závislé na pohonu je konfigurováno a uvolňována pomocí MD (doba trvání zpoždění T1 v MD), uvolňováno pomocí systémové proměnné $AA_ESR_ENABLE a spouštěno pomocí $AN_ESR_TRIGGER.

For time T1 the speed setpoint that was active when the error occurred is still output. This is an attempt to maintain the movement that was active before the failure until the physical contact is annulled or the retraction movement initiated simultaneously in other drives is completed. This can be necessary for all leading/following drives or for drives that are coupled or in a grouping.
After time T1, all axes with speed setpoint feedforward zero are stopped at the current limit, and the pulses are deleted when zero speed is reached or when the time has expired (+drive MD).

Reakce

Po dobu T1 se v případě chyby i nadále používá právě aktivní požadovaná hodnota otáček. Tím se systém pokouší zachovat pohyby, které byly prováděny před výpadkem, dokud není přerušen fyzický kontakt nebo dokud není dokončen stahovací pohyb spuštěný současně v ostatních pohonech. To může být nezbytné pro všechny řídící/vlečné pohony nebo pro pohony, které jsou ve vazbě nebo ve skupině.



Po vypršení doby T1 se všechny osy s požadovanou hodnotou rychlosti nula a s proudovým omezením zastaví, a jakmile bude dosaženo nulové rychlosti nebo pokud daný čas (+ MD pohonu) vyprší, pulzy se vymažou.

© Siemens AG 2004, Všechna práva vyhrazena

13.6.8 Zpětný pohyb závislý na pohonu

Funkce

Osy s digitálními pohony 611D mohou (pokud jsou náležitě konfigurovány a uvolněny) samostatně provést stažení, nastane-li některá z těchto situací:

- Výpadek řídícího systému (rozpoznávání výpadku provozního signálu)
- Pokles napětí v pomocném obvodu pod výstražnou prahovou hodnotu
- Aktivování systémové proměnné $AN_ESR_TRIGGER

Stahovací pohyb je prováděn samostatně pohonem 611D.
Od začátku fáze stahování udržuje pohon sám své uvolnění na dříve platných hodnotách.

Další informace viz:
/FB/ M3, Vlečení, spojení řídící hodnotou

© Siemens AG 2004, Všechna práva vyhrazena

13.6.9 Příklad: Použití reakce závislé na pohonu

Příklad konfigurace

- Osa A má pracovat jako generátorový pohon
- Osa X se má v případě chyby stáhnout s maximální rychlostí o 10 mm

Osy Y a Z mají být zastaveny se zpožděním 100 ms, aby stahovaná osa měla čas zrušit mechanickou vazbu.

Postup

1. Uvolnění volitelných doplňků „rozšířené zastavování a stahování“ a „akce přesahující provozní režim“ (obsahuje statické synchronní akce IDS...)“.
2. Přiřazení funkcí:
   $MA_ESR_REACTION[X]=11,
   $MA_ESR_REACTION[Y]=12,
   $MA_ESR_REACTION[Z]=12,
   $MA_ESR_REACTION[A]=10;
3. Konfigurace pohonu:
   MD1639 RETRACT_SPEED[X] = 400000H v kladném směru (max. rychlost)
   = FFC00000H v záporném směru
   MD1638 RETRACT_TIME[X] 10 ms (doba stahování)
   MD1637 GEN_STOP_DELAY[Y] = 100 ms
   MD1637 GEN_STOP_DELAY[Z] = 100 ms
   MD1635 GEN_AXIS_MIN_SPEED[A] = minimální otáčky generátoru (ot/min)
4. Uvolnění funkce (z výrobního programu nebo ze synchronní akce):
   $AA_ESR_ENABLE[X]=1,
   $AA_ESR_ENABLE[Y]=1,
   $AA_ESR_ENABLE[Z]=1,
   $AA_ESR_ENABLE[A]=1
5. Přepnutí pohonu generátoru na otáčky (např. v režimu vřetena M03 S1000)
6. Formulování spouštěcí podmínky jako statické synchronní akce, např.:
   - v závislosti na záběru osy generátoru:
     IDS=01 WHENEVER $AA_ESR_STAT[A]>0 DO $AN_ESR_TRIGGER=1
   - a/nebo v závislosti na alarmech, jež spouští sledování (bit13 = 2000H)
     IDS=02 WHENEVER ($AC_ALARM_STAT B_AND 'H2000')>0
     DO $AN_ESR_TRIGGER=1
   - jakož i v závislosti na monitorování synchronního chodu EG (pokud je např. osa Y definována jako vlečná osa EG a max. přípustná odchylka synchronního chodu má být 100 μs):
     IDS=03 WHENEVER ABS($VA_EG_SYNCDIFF[Y])>0.1 DO
     $AN_ESR_TRIGGER=1

© Siemens AG 2004, Všechna práva vyhrazena

13.7 Linková komunikace (od SW 5.2)



Funkce

Linka NCU, spojení mezi větším počtem jednotek NCU v jednom zařízení, nachází své uplatnění v zařízeních s decentralizovanou konstrukcí systému. V případě velkých požadavků na kanály a osy, např. u strojů s kruhovým stolem pracujících v aktu a u strojů s více vřeteny, mohou požadavky na výpočetní výkon přesáhnout možnosti konfigurace a paměťové kapacity praktických hranic pro jednu NCU.
Větší počet NCU zapojených do sítě pomocí linkového modulu NCU nabízí shora otevřené řešení, které je schopné naplnit požadavky takových obráběcích strojů. Linkový modul NCU (HW) realizuje rychlou komunikaci NCU-NCU.



Funkce je spojena s volitelnými doplňky, které si musíte objednat.



Funkce

Větší počet NCU spojených linkovým modulem může mít přístup pomocí v následujících odstavcích popsaných přístupových proměnných do globální paměťové oblasti NCU za účelem čtení a zápisu.

- Každá NCU připojená pomocí linkového modulu může používat **globální linkové proměnné**, které mohou být jednotně využívány všemi připojenými NCU.
- Linkové proměnné jsou programovatelné jako systémové proměnné.
  Význam těchto proměnných je zpravidla definován a dokumentován výrobcem stroje.
- Použití linkových proměnných:
  - globální stavy stroje
  - upnutí obrobku otevřeno/zavřeno
  - atd.
- Srovnatelně malé datové objemy
- Přenosová rychlost je velmi velká, z čehož vyplývá, že se předpokládá použití pro informace, které jsou z časového hlediska kritické.

© Siemens AG 2004, Všechna práva vyhrazena

- Přístup k těmto systémovým proměnným je možný jak z **výrobního programu**, tak i ze **synchronních akcí**. Velikost paměťové oblasti pro globální systémové proměnné NCU lze nastavit v konfiguraci.

Po interpolačním taktu mohou všechny připojené NCU konzistentně číst nově zapsanou hodnotu v některé z globálních systémových proměnných.

Linkové proměnné jsou **globálními systémovými daty**, k nimž se připojené NCU mohou obracet jako na **systémové proměnné**.
- **Obsah** těchto proměnných,
- jejich **datový typ**,
- jejich pozice (**přístupový index**) v linkové paměti jsou definovány uživatelem (to je v tomto případě zpravidla výrobce stroje).

Linkové proměnné se ukládají v linkové paměti. Po náběhu systému je linková paměť inicializována nulami.

V rámci linkové paměti je možné pracovat s následujícími linkovými proměnnými:
- INT $A_DLB[i] ; datový byte (8 bitů)
- INT $A_DLW[i] ; datové slovo (16 bitů)
- INT $A_DLD[i] ; dvojité datové slovo (32 bitů)
- REAL $A_DLR[i] ; data typu real (64 bitů)

V závislosti na typu mají linkové proměnné při čtení/zápisu1, 2, 4 nebo 8 bytů.

Index **i** označuje počátek odpovídající proměnné vzhledem k počátku konfigurované linkové paměti. Index se počítá od 0.

Rozsahy hodnot

S datovými typy jsou svázány následující rozsahy hodnot:

BYTE:	0 až 255
WORD:	-32768 až 32767
DWORD:	-2147483648 až 2147483647
REAL:	-4.19e-308 až 4.19e-307

© Siemens AG 2004, Všechna práva vyhrazena



Různé aplikace NCU sdílející přístup do linkové paměti **současně** musí linkovou paměť používat **jednotným způsobem**. Pokud je proces v čase zcela oddělen, může být linková paměť použita odlišně.



Výstraha

Proces zápisu linkové proměnné je ukončen až tehdy, když je zapsaná informace k dispozici také všem ostatním NCU. Pro tento proces jsou zapotřebí přibližně dva interpolační cykly. Lokální zápis do linkové paměti je současně zpožďován, aby byla zachována konzistentnost.



Další informace naleznete v Popisu funkcí B3 (SW 5).



Příklad programování

`$A_DLB[5]=21`	5. byte ve společné linkové paměti dostává hodnotu 21.

© Siemens AG 2004, Všechna práva vyhrazena

13.8 Zásobník os (od SW 5.2)

Funkce

U taktových kruhových strojů/strojů s více vřeteny se osy držící obrobek pohybují od jedné obráběcí jednotky k další.
Protože jsou obráběcí jednotky řízeny různými kanály, musí být při každé změně stanice/polohy ose nesoucí obrobek dynamicky nově přiřazován odpovídající kanál NCU. Tomuto účelu slouží **zásobník os**.
V daném okamžiku je na lokální obráběcí jednotce aktivní vždy jen jedna osa upínající obrobek / vřeteno. Zásobník os soustřeďuje všechny možnosti spojení pro všechny osy upínající obrobek / vřetena, z nichž je vždy právě **jedna aktivována** pro obráběcí jednotku.
Pomocí zásobníku os mohou být přiřazeny:

- lokální osy a/nebo
- linkové osy (viz Základy)

Změna použitelné osy definované pomocí zásobníku os se provádí posunutím v zásobníku os. Toto posunutí může být spouštěno z **výrobního programu**.
Zásobníky os s linkovými osami jsou provozními prostředky platnými ve všech NCU (globální pro NCU) a jsou koordinovány řídícím systémem. Zásobníky os, v nichž jsou spravovány výlučně lokální osy, jsou možné také.

Podrobné informace o konfiguraci zásobníků os naleznete v /FB/, B3, (SW 5.2).

Posouvání záznamů v zásobníku os o krok o délce n se provádí následujícími příkazy:

Programování

`AXCTSWE (CT`$_i$`)`	AXIS CONTAINER SWITCH ENABLE
`AXCTSWED (CT`$_i$`)`	AXIS CONTAINER SWITCH ENABLE DIRECT

© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení

CT_i nebo

Číslo zásobníku os, jehož obsah se má posunovat nebo

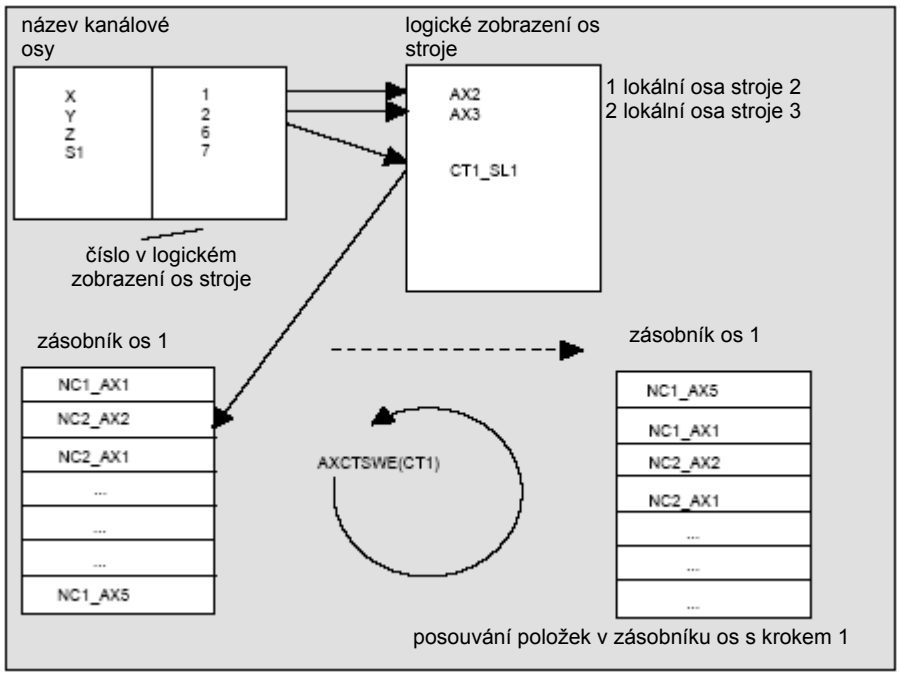
např. A_CONT1

pomocí MD nastavený individuální název zásobníku os.

Funkce

AXCTSWE ( )

Každý kanál, jehož osy jsou zaneseny do uvedeného zásobníku, poskytuje **uvolnění pro otočení zásobníku**, jakmile je obrábění v dané poloze / stanici ukončeno. Jakmile řídící systém dostane uvolnění od všech kanálů pro osy v zásobníku, zásobník se pootočí o přírůstek definovaný v SD.



Po pootočení zásobníku os o 1 je v uvedeném příkladu je osa AX5 na NCU1 přiřazena kanálové ose Z namísto osy AX1 na NCU1.

Varianta příkazu AXCTSWED(CT_i) může být použita kvůli zjednodušení uvedení do provozu. Zásobník os se otáčí pouze působením aktivního kanálu o krok, jehož délka je uložena v SD. Volání tohoto typu se smí používat jen tehdy, pokud se zbývající kanály, jež mají osy v zásobníku, nacházejí ve stavu **RESET**.

© Siemens AG 2004, Všechna práva vyhrazena

Po otočení zásobníku jsou **všechny NCU**, jejichž kanály se prostřednictvím logického zobrazení os stroje odkazují na pootočení zásobník os, ovlivněny novým přiřazením os.

13.9 Strojový čas programu/počítadlo obrobků (od SW 5.2)

Funkce

Jako pomoc pro obsluhu obráběcích strojů jsou k dispozici informace o strojovém času programu a o počítání kusů.
Tyto informace jsou specifikovány v odpovídajících strojních parametrech a mohou být zpracovávány jako systémové proměnné v NC- a/nebo PLC programu. Současně jsou tyto informace k dispozici MMC na rozhraní ovládacího panelu.

13.9.1 Strojový čas programu

Funkce

Pomocí této funkce se připraví časovač jako systémová proměnná, kterou je možné používat pro monitorování technologického procesu.
K tomuto časovači existují přístupová oprávnění jen pro čtení. Pomocí MMC si jeho obsah můžete kdykoli načíst.

© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení

Oba následující časovače jsou definovány jako specifické systémové proměnné NCK a jsou vždy aktivní.

`$AN_SETUP_TIME`	Čas od posledního nastavení v minutách; vynuluje se příkazem SETUP.
`$AN_POWERON_TIME`	Čas od posledního zapnutí systému (Power on) v minutách; vynuluje se s POWERON.

Následující tři časovače jsou definovány jako specifické kanálové systémové proměnné a mohou být aktivovány pomocí strojních parametrů.

$AC_OPERATING_TIME	Celková doba zpracování NC programu v automatickém provozním režimu v minutách.
$AC_CYCLE_TIME	Doba zpracování zvoleného NC programu v sekundách.
$AC_CUTTING_TIME	Doba záběru nástroje v sekundách.
$MC_RUNTIMER_MODE	Doba záběru nástroje v sekundách.

Všechny časovače se při náběhu řídícího systému vynulují svými předdefinovanými hodnotami a mohou být čteny nezávisle na jejich aktivaci.

Příklad programování

1. Aktivování doby zpracování pro aktivní NC program, přitom žádné měření při aktivním zkušebním zpracování a testování programu:

```
$MC_PROCESSTIMER_MODE = 'H2'
```

2. Aktivování měření doby záběru nástroje, přitom i měření při aktivním zkušebním zpracování a testování programu:

```
$MC_PROCESSTIMER_MODE= 'H34'
```

3. Aktivování měření celkové doby zpracování a doby záběru nástroje, přitom měření při testování programu:

```
$MC_PROCESSTIMER_MODE= 'H25'
```


© Siemens AG 2004, Všechna práva vyhrazena

13.9.2 Počítadlo obrobků

Funkce

Pomocí funkce „počítadlo obrobků“ je možné připravit počítadlo, které pak můžete použít např. pro počítání obrobků uvnitř řídícího systému. Tato počítadla existují jako kanálové systémové proměnné s přístupem pro čtení a zápis v rozsahu hodnot 0 až 999 999 999.
Pomocí strojních parametrů je možné provádět aktivování čítačů, okamžik vynulování a lze také ovlivňovat algoritmus počítání.

Vysvětlení

Jsou připravena následující počítadla:

$AC_REQUIRED_PARTS	Počet požadovaných obrobků (požadovaná hodnota). V tomto počítadle je definován počet obrobků, při jehož dosažení je počet aktuálních obrobků v proměnné $AC_ACTUAL_PARTS nulový. Pomocí MD je možné aktivovat generování signalizačních alarmů „Požadovaný počet obrobků dosažen“ a VDI signálu kanálu „Požadovaný počet obrobků dosažen“.
$AC_TOTAL_PARTS	Počet celkem vyrobených obrobků (celková skutečná hodnota). Počítadlo udává počet všech obrobků vyrobených od okamžiku spuštění. Počítadlo se automaticky vynuluje při náběhu řídícího systému.
$AC_ACTUAL_PARTS	Aktuální počet vyrobených obrobků (aktuální skutečná hodnota). V tomto počítadle je registrován počet všech obrobků vyrobených od jeho spuštění. Při dosažení požadovaného počtu obrobků ($AC_REQUIRED_PARTS) se počítadlo automaticky vynuluje (předpokladem je, že $AC_REQUIRED_PARTS se nerovná nule).
$AC_SPECIAL_PARTS	Počet uživatelem specifikovaných obrobků. Toto počítadlo umožňuje uživateli počítání obrobků podle vlastní definice. Může být také definováno aktivování alarmu při dosažení požadovaného počtu obrobků $AC_REQUIRED_PARTS. Vynulování počítadla musí provádět uživatel sám.

Funkce „Počítadlo obrobků“ je nezávislá na funkcích pro správu nástrojů. Všechny čítače mohou být čteny pomocí MMC a odtud lze do nich i zapisovat.
Všem počítadlům je při náběhu systému dosazena předdefinovaná nulová hodnota a pak mohou být nezávisle na svém aktivování obsluhovány pomocí funkcí pro jejich čtení a zápis..

© Siemens AG 2004, Všechna práva vyhrazena



Příklad programování

1. Aktivování čítače obrobků $AC_REQUIRED_PARTS:

`$MC_PART_COUNTER='H3'`	$AC_REQUIRED_PARTS je aktivní, signalizace alarmu při $AC_REQUIRED_PARTS == $AC_SPECIAL_PARTS.
2. Aktivování počítadla obrobků $AC_TOTAL_PARTS	
`$MC_PART_COUNTER='H10'` `$MC_PART_COUNTER_MCODE[0]=80`	$AC_TOTAL_PARTS je aktivní, s každým příkazem M02 se počítadlo o 1 zvýší, $MC_PART_COUNTER_MCODE[0] nemá žádný význam
3. Aktivování počítadla obrobků $AC_ACTUAL_PARTS	
`$MC_PART_COUNTER='H300'` `$MC_PART_COUNTER_MCODE[1]=17`	$AC_TOTAL_PARTS je aktivní, s každým příkazem M17 se počítadlo o 1 zvýší.
4. Aktivování počítadla obrobků $AC_SPECIAL_PARTS	
`$MC_PART_COUNTER='H3000'` `$MC_PART_COUNTER_MCODE[2]=77`	$AC_SPECIAL_PARTS je aktivní, s každým příkazem M77 se počítadlo o 1 zvýší.
5. Aktivování počítadla obrobků $AC_ACTUAL_PARTS	
`$MC_PART_COUNTER='H200'` `$MC_PART_COUNTER_MCODE[1]=50`	$AC_TOTAL_PARTS není aktivní, zbytek nemá žádný význam.
6. Aktivování všech počítadel v příkladech 1 – 4:	
`$MC_PART_COUNTER ='H3313'` `$MC_PART_COUNTER_MCODE[0] =80` `$MC_PART_COUNTER_MCODE[1] =17` `$MC_PART_COUNTER_MCODE[2] =77`	$AC_REQUIRED_PARTS je aktivní, signalizace alarmu při $AC_REQUIRED_PARTS == $AC_SPECIAL_PARTS. $AC_TOTAL_PARTS je aktivní, s každým příkazem M02 se počítadlo o 1 zvýší, $MC_PART_COUNTER_MCODE[0] nemá žádný význam. $AC_ACTUAL_PARTS je aktivní, s každým příkazem M17 se počítadlo o 1 zvýší. $AC_SPECIAL_PARTS je aktivní, s každým příkazem M77 se počítadlo o 1 zvýší.

© Siemens AG 2004, Všechna práva vyhrazena

13.10 Příkaz pro interaktivní vyvolání okna z výrobního programu: MMC (od SW 4.4)



Programování

```
MMC („CYCLES, PICTURE_ON, T_SK.COM, BILD, MGUD.DEF, BILD_3.AWB,
TEST_1, A1“,“S“)
```



Vysvětlení

CYCLES	Systémová oblast, ve které má být konfigurované uživatelské dialogové okno otevíráno.
PICTURE_ON, příp. PICTURE_OF	Příkaz: Aktivování, příp. deaktivování obrázku
T_SK.COM	Soubor Com: Název souboru dialogového obrázku (uživatelské cykly). Zde je definován vzhled dialogového obrázku. V dialogovém obrázku se mohou vypisovat uživatelské proměnné a/nebo textové komentáře.
BILD	Název dialogového obrázku: Jednotlivé obrázky jsou aktivovány prostřednictvím svých názvů.
MGUD.DEF	Uživatelský datový definiční soubor, do něhož je přístup při čtení/zápisu proměnných.
BILD_3.AWB	Grafický soubor
TEST_1	Doba zobrazování nebo potvrzovací proměnná.
A1	Textové proměnné ...
„S“	Režim potvrzování: synchronní, potvrzení pomocí programového tlačítka "OK“.



Funkce

Pomocí příkazu MMC můžete z výrobního programu zobrazit na HMI/MMC uživatelem definované dialogové okno.
Vzhled dialogového okna je konfigurován čistě textově (soubor COM v adresáři cyklů), systémové programové vybavení MMC/HMI zůstává přitom nezměněno.
Uživatelem definovaná dialogová okna nemohou být vyvolávána současně v různých kanálech.



Podrobné pokyny pro programování příkazů MMC (včetně příkladů programování) naleznete v /IAM/, knihy IM1 až IM4 v závislosti na používaném programovém vybavení MMC/HMI.

© Siemens AG 2004, Všechna práva vyhrazena

13.11 Ovlivňování vedení pohybu

13.11.1 Procentuální korekce trhavých pohybů: JERKLIM

Programování

```
JERKLIM[Osa]
```

Vysvětlení příkazu

JERKLIM	Procentuální změna pro maximální přípustné trhnutí vztažená na hodnotu nastavenou pro osu ve strojním parametru.
Osa	Osa stroje, jejíž mezní hodnota trhnutí má být přizpůsobena.

Funkce

V kritických úsecích programu se může vynořit nutnost omezit trhnutí pod maximální možnou hodnotu, aby se např. snížily nároky na zatížení stroje. Režim ovládání zrychlení SOFT musí být aktivní. Funkce je uplatňována jen na dráhové osy.

Postup

V automatickém provozním režimu je pro naprogramovanou osu omezena hodnota trhnutí na uvedenou procentuální hodnotu z mezní hodnoty trhnutí, která je uložena ve strojním parametru.

Příklad: JERKLIM[X] = 75
Znamená: Osa saní ve směru X smí být zrychlována/brzděna jen na 75% pro osu přípustného trhnutí.

Rozsah hodnot: 1 ... 200
100 odpovídá: žádné ovlivňování trhnutí. Hodnota 100 je v platnosti po resetu a po spuštění výrobního programu.

Další upozornění

Další příklad se nachází na konci další podkapitoly.

© Siemens AG 2004, Všechna práva vyhrazena

13.11.2 Procentuální korekce rychlosti: VELOLIM

Programování

```
VELOLIM[Osa]
```

Vysvětlení příkazu

VELOLIM	Procentuální změna pro maximální přípustnou rychlost vztažená na hodnotu nastavenou pro osu ve strojním parametru.
Osa	Osa stroje, jejíž mezní hodnota rychlosti má být přizpůsobena.

Funkce

V kritických úsecích programu se může vynořit nutnost omezit rychlosti pod maximální možnou hodnotu, aby se např. snížily nároky na zatížení stroje nebo se zlepšila jakost povrchu. Funkce je uplatňována na dráhové a polohovací osy.

Postup

V automatickém provozním režimu je pro naprogramovanou osu omezena hodnota rychlosti na uvedenou procentuální hodnotu z mezní hodnoty rychlosti, která je uložena ve strojním parametru.

Příklad: `N70 VELOLIM[X] = 80`
Znamená: Osa saní se ve směru X smí pohybovat pouze 80% své maximální přípustné rychlosti.

Rozsah hodnot: 1 ... 100
100 odpovídá: žádné ovlivňování rychlosti.
Hodnota 100 je v platnosti po resetu a po spuštění výrobního programu.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

```
N1000 G0 X0 Y0 F10000 SOFT G64
N1100 G1 X20 RNDM=5 ACC[X]=20 ACC[Y]=30
N1200 G1 Y20 VELOLIM[X]=5 JERKLIM[Y]=200
N1300 G1 X0 JERKLIM[X]=2
N1400 G1 Y0
M30
```

13.12 Spojení Master/Slave

Programování

`MASLDEF(Slv1, Slv2, ..., osa Master)`	Pro dynamickou konfiguraci (od SW 6.4)
`MASLDEL(Slv1, Slv2, ..., )`	Pro dynamickou konfiguraci (od SW 6.4)
`MASLON(Slv1, Slv2, ..., )`	
`MASLOF(Slv1, Slv2, ..., )`	
`MASLOFS(Slv1, Slv2, ..., )`	(od SW 6.4)

Vysvětlení parametrů

`Slv1, Slv2, ...`	Osy Slave, které jsou řízeny osou Master
`osa Master`	Osa, která ve svazku Master/Slave vede definované osy Slave

© Siemens AG 2004, Všechna práva vyhrazena

Funkce

Vazba Master/Slave do SW 6.4 umožňovala spojení os Slave s jejich osou Master jen tehdy, pokud byly příslušné osy v klidu.
Rozšíření od SW 6.4 připouští aktivování a rozpojování vazby i u **otáčejících** se vřeten s regulací otáček a dynamickou konfiguraci.

Dynamická konfigurace

MASLDEF	(od SW 6.4) Definice svazku Master/Slave z výrobního programu. Před SW 6.4 se definice prováděla výlučně pomocí strojních parametrů.
MASLDEL	(od SW 6.4) Příkaz ruší přiřazení os Slave ose Master a současně rozpojuje analogicky k příkazu MASLOF aktuální vazbu. Definice Master/Slave nastavené pomocí strojních parametrů zůstávají zachovány.

Všeobecné

MASLON	Aktivování dočasné vazby.
MASLOF	Aktivní vazba je tímto příkazem rozpojena. (od SW 6.4) U vřeten s regulací otáček se tento příkaz provádí ihned. Vřetena Slave, která se v tomto okamžiku otáčela, si své otáčky zachovají, dokud pro ně nejsou naprogramovány nové otáčky.
MASLOFS	(od SW 6.4) Aby bylo možné vřetena Slave při rozpojení vazby automaticky zabrzdit, je možné použít příkaz MASLOFS. U os a vřeten s polohovou regulací je možné vazbu rozpojit jen v klidu.

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění (od SW 6.4)

U příkazů MASLOF/MASLOFS odpadá implicitní zastavení předběžného zpracování. V důsledku chybějícího zastavení předběžného zpracování neobsahují systémové proměnné $P os Slave aktuální hodnoty, dokud nejsou znovu naprogramovány.

Příklad programování

Dynamická konfigurace vazby Master/Slave z výrobního programu.

Příslušná osa po pootočení zásobníku os se má stát osou Master.

`MASLDEF(AUX,S3)`	; S3 je Master pro AUX
`MASLON(AUX)`	; aktivování vazby pro AUX
`M3=3 S3=4000`	; otáčení směrem vpravo
`MASLDEL(AUX)`	; vymazání konfigurace a rozpojení vazby
`AXCTSWE(CT1)`	; otočení zásobníku

Další upozornění (od SW 6.4)

Pro osu Slave může být skutečná hodnota pomocí příkazu PRESETON synchronizována na stejnou hodnotu, jakou má osa Master. Za tím účelem musí být stálá vazba Master/Slave krátkodobě rozpojena, aby po Power On bylo možné ose Slave, se kterou se nenajíždí na referenční bod, dosadit hodnotu osy Master. Potom se trvalá vazba znovu vytvoří.

Trvalá vazba typu Master/Slave se aktivuje přiřazením strojního parametru MD 37262: MS_COUPLING_ALWAYS_ACTIVE = 1 a nemá žádný vliv na příkazy dočasné vazby.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

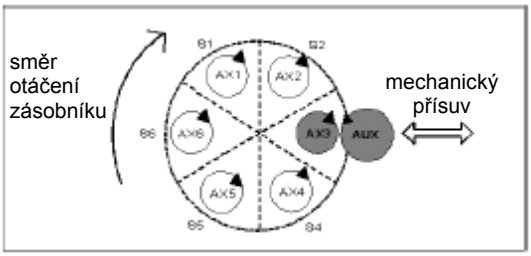
Vazba skutečnou hodnotou osy Slave na stejnou hodnotu osy Master pomocí příkazu PRESETON.

V případě trvalé vazby Master/Slave má být skutečná hodnota osy Slave příkazem PRESETON změněna.

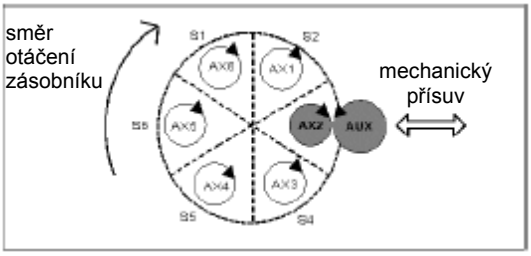
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; krátké vypnutí trvalé vazby
N37263 NEWCONF	;
N37264 STOPRE	;
MASLOF(Y1)	; dočasná vazba deaktivována
N5 PRESETON(Y1, 0, Z1, 0, B1, 0, C1, 0, U1, 0)	; dosazení skutečné hodnoty ose Slave, ; která nenajíždí na referenční bod, protože ; tyto osy jsou při zapnutí aktivovány
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; opětovná aktivace trvalé vazby
N37263 NEWCONF	

Aby bylo možné vytvořit vazbu s dalším vřetenem po pootočení zásobníku, musí být předcházející vazba rozpojena, její konfigurace vymazána a pak musí být konfigurována nová vazba.
Příklad posloupnosti vytvoření vazby, pozice 3 / zásobník CT1.

Viz /FB/, B3, kapitola 2.6, Zásobník os.



výchozí pozice



po pootočení o jednu pozici

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

Uživatelské programy pro oddělování třísky

14.1 Pomocné funkce pro oddělování třísky....14-568
14.2 Příprava kontury - CONTPRON....14-569
14.3 Dekódování kontury – CONTDCON (od SW 5.2)....14-576
14.4 Průsečík dvou konturových prvků – INTERSEC....14-580
14.5 Obrábění konturových prvků z tabulky – EXECTAB....14-582
14.6 Výpočet kruhových dat – CALCDAT....14-583

© Siemens AG 2004, Všechna práva vyhrazena

14.1 Pomocné funkce pro oddělování třísky



Uživatelské programy pro oddělování třísky
Pro oddělování třísky jsou Vám nabízeny již hotové cykly pro obrábění. Díky nim máte možnost si sestavovat své vlastní programy pro oddělování třísky využívající následujících funkcí.

CONTPRON	Aktivování přípravy kontury ve formě tabulky (11 sloupců)
CONTDCON	Aktivování dekódování kontury ve formě tabulky (6 sloupců)
INTERSEC	Vypočítání průsečíku mezi dvěma konturovými prvky. (Pouze pro tabulky, které byly vytvořeny pomocí příkazu CONTPRON.)
EXECTAB	Blokové obrobení konturových prvků z tabulky. (Pouze pro tabulky, které byly vytvořeny pomocí příkazu CONTPRON.)
CALCDAT	Výpočet rádiusu a středu.



Tyto funkce můžete používat nejen pro oddělování třísky, ale i univerzálně.

© Siemens AG 2004, Všechna práva vyhrazena

14.2 Příprava kontury - CONTPRON

Programování

```
CONTPRON (TABNAME, BEARBART, NN, MODE)
EXECUTE (FEHLER)
```

Vysvětlení parametrů

CONTPRON	Aktivování přípravy kontury
TABNAME	Název tabulky kontury
BEARBART	Parametr pro druh opracování: “G“: podélné soustružení, vnitřní obrábění „L“: podélné soustružení, vnější obrábění „N“: příčné soustružení, vnitřní obrábění „P“: příčné soustružení, vnější obrábění
NN	Počet podříznutí ve výsledné proměnné typu INT
MODE (od SW 4.4)	Směr obrábění, typ INT 0 = příprava kontury směrem dopředu (jako dříve do SW 4.3, předem definovaná hodnota) 1 = příprava kontury v obou směrech
EXECUTE	Ukončení přípravy kontury
FEHLER	Proměnná pro zpětné chybové hlášení, typ INT 1 = chyba, 0 = žádná chyba

Funkce

Bloky zpracovávané po příkazu CONTPRON popisují konturu, která má být připravena.
Bloky se nebudou obrábět, budou se ukládat do tabulky kontury.
Každému prvku kontury odpovídá jeden řádek ve dvourozměrném poli tabulky kontury.
Jsou poskytovány informace o počtu zjištěných podříznutí.
Příkazem EXECUTE se příprava kontury ukončí a současně se přepne do normálního režimu zpracování.
Příklad:

```
N30 CONTPRON (...)
N40 G1 X... Z...
N50 ...
N100 EXECUTE (...)
```

© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Předpoklady pro vyvolání

Před vyvoláním příkazu CONTPRON:

- se musí najet na počáteční bod, který umožňuje bezkolizní zpracování.
- příkazem G40 je nutno deaktivovat korekci rádiusu nástroje.

Povolené příkazy posuvu, souřadný systém

Pro programování kontury jsou přípustné pouze příkazy G0 až G3 a také zaoblení a faseta.
Od SW 4.4 je možné při programování kruhů používat ještě příkazy CIP a CT.
Funkce pro spliny, polynomy a závity vedou k chybám.
Změny souřadného systému aktivováním framu jsou mezi příkazy CONTPRON a EXECUTE nepřípustné. Totéž se týká i přepínání mezi G70 a G71 / G700 a G710. Záměna geometrických os pomocí příkazu GEOAX během přípravy tabulky kontury má za následek alarm.

Ukončení přípravy

Vyvoláním příkazu EXECUTE(proměnná) přejdete po zápisu kontury zpět do normálního zpracování programu a příprava kontury tím bude ukončena. Hodnota proměnné má tento význam:

1 = chyba
0 = žádná chyba (kontura mohla být bezchybně připravena)

Prvky podříznutí

Je možné zvolit, zda popis kontury v jednotlivých prvcích podříznutí se má uskutečňovat v podprogramu nebo v jednotlivých blocích.

Oddělování třísky nezávisle na naprogramovaném směru kontury (od SW 4.4)

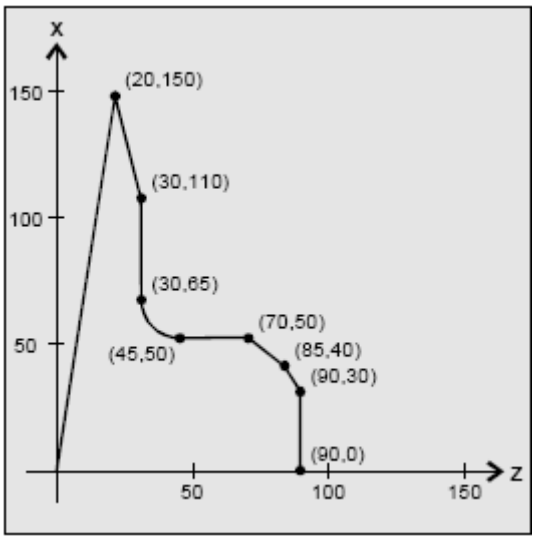
Od SW 4.4 je příprava kontury pomocí příkazu CONTPRON rozšířena tak, že po jeho vyvolání je tabulka kontury k dispozici nezávisle na naprogramovaném směru.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování 1

Vytvoření tabulky kontury s těmito údaji:

- Název KTAB
- Max. 30 konturových prvků (kruhové oblouky a přímky)
- Proměnná pro počet vyskytujících se prvků podříznutí
- Proměnná pro chybové hlášení



Výrobní program pro NC

`N10 DEF REAL KTAB[30,11]`	Tabulka kontury s názvem KTAB a s např. max. 30 konturovými prvky Hodnota parametru 11 je pevná veličina
`N20 DEF INT ANZHINT`	Proměnná pro počet prvků podříznutí s názvem ANZHINT
`N30 DEF INT FEHLER`	Proměnná pro potvrzení 0 = žádná chyba, 1 = chyba
`N40 G18`	
`N50 CONTPRON (KTAB,"G",ANZHINT)`	Vyvolání přípravy kontury
`N60 G1 X150 Z20` `N70 X110 Z30` `N80 X50 RND=15` `N90 Z70` `N100 X40 Z85` `N110 X30 Z90` `N120 X0`	N60 až N120 je popis kontury
`N130 EXECUTE(FEHLER)`	Ukončení plnění tabulky kontury, přepnutí do normálního programového režimu
`N140 …`	Další zpracování tabulky

© Siemens AG 2004, Všechna práva vyhrazena

Odpovídající tabulka KTAB

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7(	(8)	(9)	(10)
7	7	11	0	0	20	150	0	82.40535663	0	0
0	2	11	20	150	30	110	-1111	104.0362435	0	0
1	3	11	30	110	30	65	0	90	0	0
2	4	13	30	65	45	50	0	180	45	65
3	5	11	45	50	70	50	0	0	0	0
4	6	11	70	50	85	40	0	146.3099325	0	0
5	7	11	85	40	90	30	0	116.5650512	0	0
6	0	11	90	30	90	0	0	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Vysvětlení obsahu jednotlivých sloupců

(0) Ukazatel na následující konturový prvek (na číslo řádku v tomto sloupci)

(1) Ukazatel na předcházející konturový prvek

(2) Kódování konturového režimu pro daný pohyb
možné hodnoty pro X = abc

a = 10^2	G90 = 0	G91 = 1		
b = 10^1	G70 = 0	G71 = 1		
c = 10^0	G0 = 0	G1 = 1	G1 = 2	G3 = 3

(3), (4) Počáteční bod konturového prvku
(3) = abscisa, (4) = ordináta v aktuální rovině

(5), (6) Koncový bod konturového prvku
(5) = abscisa, (6) = ordináta v aktuální rovině

(7) Ukazatel, který obsahuje lokální minima a maxima na kontuře

(8) Maximální hodnota mezi konturovým prvkem a abscisou (při podélném obrábění), příp. ordinátou (při příčném obrábění)

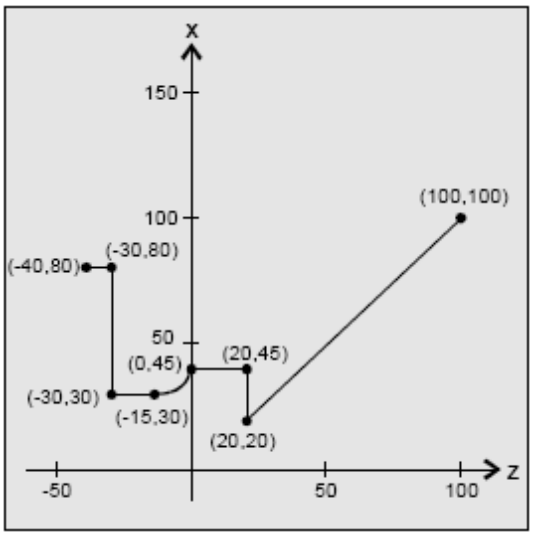
(9),(10) Souřadnice středu konturového prvku, jedná-li se o kruhový blok.
(9) = abscisa, (10) = ordináta

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování 2

Vytvořte tabulku kontury s následujícími parametry:

- Název KTAB
- max. 92 konturových prvků (kruhové oblouky, přímky)
- Režim opracování: podélné soustružení, vnější opracování
- Příprava směrem dopředu a dozadu



Výrobní program pro NC

`N10 DEF REAL KTAB[92,11]`	Tabulka kontury s názvem KTAB a s např. 92 konturovými prvky hodnota parametru 11 je pevná hodnota
`N20 CHAR BT="L"`	Režim obrábění pro CONTPRON: podélné soustružení, vnější opracování
`N30 DEF INT HE=0`	Počet prvků podříznutí = 0
`N40 DEF INT MODE=1`	Příprava dopředu i zpátky
`N50 DEF INT ERR=0`	Zpětné chybové hlášení
`...`	
`N100 G18 X100 Z100 F1000`	
`N105 CONTPRON (KTAB, BT, HE, MODE)`	Vyvolání přípravy kontury
`N110 G1 G90 Z20 X20` `N120 X45` `N130 Z0` `N140 G2 Z-15 X30 K=AC(-15) I=AC(45)` `N150 G1 Z-30` `N160 X80` `N170 Z-40`	
`N180 EXECUTE(ERR)`	Ukončení plnění tabulky kontury, přepnutí do normálního programového režimu
`...`	

© Siemens AG 2004, Všechna práva vyhrazena

Odpovídající tabulka KTAB

řádek	sloupec										
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7(	(8)	(9)	(10)
0	6[1]	7[2]	11	100	100	20	20	0	45	0	0
1	0[3]	2	11	20	20	20	45	-3	90	0	0
2	1	3	11	20	45	0	45	0	0	0	0
3	2	4	12	0	45	-15	30	5	90	-15	45
4	3	5	11	-15	30	-30	30	0	0	0	0
5	4	7	11	-30	30	-30	45	-1111	90	0	0
6	7	0[4]	11	-30	80	-40	80	0	0	0	0
7	5	6	11	-30	45	-30	80	0	90	0	0
8	1[5]	2[6]	0	0	0	0	0	0	0	0	0
	...										
83	84	0[7]	11	20	45	20	80	0	90	0	0
84	90	83	11	20	20	20	45	-1111	90	0	0
85	0[8]	86	11	-40	80	-30	80	0	0	0	0
86	85	87	11	-30	80	-30	30	88	90	0	0
87	86	88	11	-30	30	-15	30	0	0	0	0
88	87	89	13	-15	30	0	45	-90	90	-15	45
89	88	90	11	0	45	20	45	0	0	0	0
90	89	84	11	20	45	20	20	84	90	0	0
91	83[9]	85[10]	11	20	20	100	100	0	45	0	0

Vysvětlení obsahu jednotlivých sloupců

(0) Ukazatel na následující konturový prvek (na číslo řádku v tomto sloupci)

(1) Ukazatel na předcházející konturový prvek

(2) Kódování konturového režimu pro daný pohyb
možné hodnoty pro X = abc

a = 10^2	G90 = 0	G91 = 1		
b = 10^1	G70 = 0	G71 = 1		
c = 10^0	G0 = 0	G1 = 1	G1 = 2	G3 = 3

(3), (4) Počáteční bod konturového prvku
(3) = abscisa, (4) = ordináta v aktuální rovině

(5), (6) Koncový bod konturového prvku
(5) = abscisa, (6) = ordináta v aktuální rovině

(7) Ukazatel, který obsahuje lokální minima a maxima na kontuře

(8) Maximální hodnota mezi konturovým prvkem a abscisou (při podélném obrábění), příp. ordinátou (při příčném obrábění)

(9),(10) Souřadnice středu konturového prvku, jedná-li se o kruhový blok.
(9) = abscisa, (10) = ordináta

© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení poznámek ve sloupcích

Vždy v řádku tabulky 0: 1) Předcházející: Řádek n obsahuje konec kontury směrem dopředu
2) Následující: Řádek n obsahuje konec tabulky kontury směrem dopředu.

Pokaždé jednou v rámci konturových prvků směrem dopředu:
3) Předcházející: začátek kontury (dopředu)
4) Následující: konec kontury (dopředu)

Vždy na řádku konce tabulky kontury (dopředu) + 1:
5) Předcházející: počet prvků podříznutí směrem dopředu
6) Následující: počet prvků podříznutí směrem dozadu

Pokaždé jednou v rámci konturových prvků směrem dozadu:
7) Následující: konec kontury (dozadu)
8) Předcházející: začátek kontury (dozadu)

Vždy na posledním řádku tabulky:
9) Předcházející: Řádek n obsahuje začátek kontury (dozadu)
10) Následující: Řádek n obsahuje začátek tabulky kontury (dozadu)

© Siemens AG 2004, Všechna práva vyhrazena

14.3 Dekódování kontury – CONTDCON (od SW 5.2)



Programování

```
CONTCDON (TABNAME,MODE)
EXECUTE (FEHLER)
```



Vysvětlení parametrů

CONTCDON	Aktivování přípravy kontury
TABNAME	Název tabulky kontury
MODE	Směr obrábění, typ INT 0 = příprava kontury (předdefinovaná hodnota) podle posloupnosti konturových bloků
EXECUTE	Ukončení přípravy tabulky
FEHLER	Proměnná pro chybové hlášení, typ INT 1 = chyba, 0 = žádná chyba



Funkce

Bloky prováděné po příkazu CONTDCON popisují konturu, která má být dekódována. Bloky se nebudou provádět, místo toho se budou ukládat do tabulky kontury se 6 sloupci, která v paměti zabírá málo místa.
Každý konturový prvek odpovídá jednomu řádku v tabulce kontury. Ze znalosti níže uvedených pravidel kódování mohou být z řádků tabulky sestavovány aplikace (např. cykly) programů v kódu DIN. V řádku tabulky 0 jsou uloženy údaje výchozího bodu. Rozsah G-kódů které jsou přípustné pro příkaz CONTDCON v úseku programu, který má být převeden do tabulky, je širší než u příkazu CONTPRON. Díky tomu je možné uložit také posuvy a typ posuvu pro každý úsek kontury.
Pomocí příkazu EXECUTE se příprava kontury ukončí a současně se vrátíte do normálního režimu zpracování.
Příklad:

```
N30 CONTDCON(…)
N40 G1 X… Z…
N50…
N100 EXECUTE(…)
```


© Siemens AG 2004, Všechna práva vyhrazena

Další upozornění

Předpoklady pro vyvolání

Před vyvoláním příkazu CONTDCON:

- se musí najet na počáteční bod, který umožňuje bezkolizní zpracování.
- příkazem G40 je nutno deaktivovat korekci rádiusu nástroje.

Povolené příkazy posuvu, souřadný systém

Pro programování kontury jsou přípustné následující G-skupiny s uvedenými příkazy:

G-skupina 1: G0, G1, G2, G3
G-skupina 10: G9
G-skupina 11: G60, G44, G641, G642
G-skupina 13: G70, G71, G700, G710
G-skupina 14: G90, G91
G-skupina 15: G93, G94, G95, G96/G961

a navíc ještě zaoblení a faseta.
Je možné používat také příkazy pro programování kruhu CIP a CT. Funkce pro spliny, polynom a závit způsobují chybu.

Změny souřadného systému aktivováním framu jsou mezi příkazy CONTDCON a EXECUTE nepřípustné. Totéž se týká i přepínání mezi G70 a G71 / G700 a G710. Záměna geometrických os pomocí příkazu GEOAX během přípravy tabulky kontury má za následek alarm.

© Siemens AG 2004, Všechna práva vyhrazena

Ukončení přípravy

Vyvoláním příkazu EXECUTE(FEHLER) přejdete po zápisu tabulky kontury zpět do normálního zpracování programu a příprava kontury tím bude ukončena. Hodnota proměnné FEHLER má tento význam:

0 = žádná chyba (kontura mohla být bezchybně připravena)
1 = chyba

Nepřípustné příkazy, nesprávné výchozí podmínky, opakované vyvolání příkazu CONTDCON bez příkazu EXECUTE ( ), příliš málo konturových bloků nebo příliš malá definovaná tabulka mají kromě toho za následek aktivování alarmu.

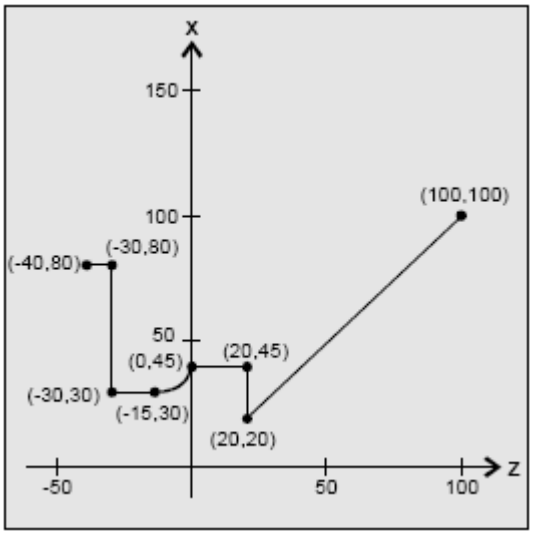
Oddělování třísky v naprogramovaném směru kontury

Tabulka kontury vytvořená prostřednictvím příkazu CONTDCON je určena pro oddělování třísky podél kontury v naprogramovaném směru.

Příklad programování

Vytvořte tabulku kontury s následujícími parametry:

- Název KTAB
- Konturové prvky (kruhové oblouky, přímky)
- Režim opracování: soustružení
- Příprava směrem dopředu



© Siemens AG 2004, Všechna práva vyhrazena

Výrobní program pro NC

`N10 DEF REAL KTAB[9,6]`	Tabulka kontury s názvem KTAB a s 9 řádky. Ty umožňují 8 konturových bloků. Hodnota parametru 6 (počet sloupců v tabulce) je pevnou veličinou.
`N20 DEF INT MODE = 0`	Standardní hodnota 0: Jen v naprogramovaném směru kontury, hodnota 1 je nepřípustná.
`N30 DEF INT ERROR = 0`	Zpětné chybové hlášení
`...`	
`N100 G18 G64 G90 G94 G710`	
`N101 G1 Z100 X100 F1000`	
`N105 CONTDCON (KTAB, MODE)`	Volání dekódování kontury MODE je možno vypustit, viz výše.
`N110 G1 Z20 X20 F200` `N120 G9 X45 F300` `N130 Z0 F400`	Popis kontury
`N140 G2 Z-15 X30 K=AC(-15) I=AC(45)F100`	
`N150 G64 Z-30 F600` `N160 X80 F700` `N170 Z-40 F800`	
`N180 EXECUTE(ERROR)`	Ukončení plnění tabulky kontury, přepnutí do normálního programového režimu
`...`	

Index sloupce	0	1	2	3	4	5
Index řádku	Režim kontury	Koncový bod abscisa	Koncový bod ordináta	Střed abscisa	Střed ordináta	Posuv
0	30	100	100	0	0	7
1	11031	20	20	0	0	200
2	111031	20	45	0	0	300
3	11031	0	45	0	0	400
4	11032	-15	30	-15	45	100
5	11031	-30	30	0	0	600
6	11031	-30	80	0	0	700
7	11031	-40	80	0	0	800
8	0	0	0	0	0	0

© Siemens AG 2004, Všechna práva vyhrazena

Vysvětlení obsahu sloupců

Řádek 0 Kódování pro **počáteční bod:**

- sloupec 0: 10^0 (místo jednotek): G0 = 0
  10^1 (místo desítek): G70 = 0, G71 = 1, G700 = 2, G710 = 3
- sloupec 1: počáteční bod na abscise
- sloupec 2: počáteční bod na ordinátě
- sloupce 3-4: 0
- sloupec 5: index řádku posledního úseku kontury v tabulce

Řádky 1-n: Záznamy **úseků kontury**

- sloupec 0: 10^0 (místo jednotek): G0 = 0
  10^1 (místo desítek): G70 = 0, G71 = 1, G700 = 2, G710 = 3
  10^2 (místo stovek): G90 = 0, G91 = 1
  10^3 (místo tisíců): G93 = 0, G94 = 1, G95 = 2, G96 = 3
  10^4 (místo desetitisíců): G60 = 0, G44 = 1, G641 = 2, G642 =3
  10^5 (místo stotisíců): G9 =1
- sloupec 1: koncový bod na abscise
- sloupec 2: koncový bod na ordinátě
- sloupec 3: střed na abscise při kruhové interpolaci
- sloupec 4 střed na ordinátě při kruhové interpolaci
- sloupec 5: posuv

14.4 Průsečík dvou konturových prvků – INTERSEC



Programování

```
VARIB=INTERSEC (TABNAME1[n1], TABMANE2[n2], TABNAME3)
```



Vysvětlení parametrů

VARIB	Proměnná pro status:	TRUE: průsečík nalezen FALSE: průsečík nenalezen
TABNAME1[n1]	Název tabulky a n1-tý konturový prvek v první tabulce	
TABMANE2[n2]	Název tabulky a n2-tý konturový prvek ve druhé tabulce	
TABNAME3	Název tabulky pro souřadnice průsečíku v aktivní rovině G17–G19	

© Siemens AG 2004, Všechna práva vyhrazena



Funkce

Příkaz INTERSEC zjišťuje průsečík dvou normovaných konturových prvků z tabulek kontur vytvořených pomocí příkazu CONTPRON. Výstupní stavová informace udává, zda průsečík existuje (TRUE = průsečík) nebo neexistuje (FALSE = žádný průsečík).



Další upozornění

Nezapomeňte, že proměnné musí být před jejich použitím definovány.



Příklad programování

Je třeba zjistit průsečík konturového prvku 3 tabulky kontury KTAB1 a konturového prvku 7 tabulky KTAB2. Souřadnice průsečíku v aktivní rovině budou uloženy do proměnné SCHNITT (1. prvek – abscisa, 2. prvek = ordináta).
Pokud průsečík neexistuje, je požadován skok na návěští KEINSCH (žádný průsečík nenalezen).

`DEF REAL KTAB1 [12, 11]`	tabulka kontury 1
`DEF REAL KTAB2 [10, 11]`	tabulka kontury 2
`DEF REAL SCHNITT [2]`	tabulka průsečíku
`DEF BOOL ISPOINT`	proměnná pro status
`...`	
`N10 ISPOINT=INTERSEC (KTAB1[3],KTAB2[7],SCHNITT)`	
	volání průsečíku konturových prvků
`N20 IF ISPOINT==FALSE GOTOF KEINSCH`	skok na KEINSCH
`...`	

© Siemens AG 2004, Všechna práva vyhrazena

14.5 Obrábění konturových prvků z tabulky – EXECTAB



Programování

```
EXECTAB (TABNAME[n])
```



Vysvětlení příkazu

TABNAME[n]	Název tabulky s číslem prvku n



Funkce

Pomocí příkazu EXECTAB můžete blok po bloku zpracovat konturové prvky v tabulce, která byla vytvořena např. příkazem CONTPRON.



Příklad programování

Pomocí podprogramu EXECTAB budou blok po bloku zpracovány konturové prvky z tabulky KTAB. Při volání se budou postupně předávat prvky 0 až 2.

N10 EXECTAB (KTAB[0])	Provedení prvku 0 z tabulky KTAB
N20 EXECTAB (KTAB[1])	Provedení prvku 1 z tabulky KTAB
N30 EXECTAB (KTAB[2])	Provedení prvku 2 z tabulky KTAB

© Siemens AG 2004, Všechna práva vyhrazena

14.6 Výpočet kruhových dat – CALCDAT

Programování

```
VARIB = CALCDAT(PKT[n,2], ANZ, ERG)
```

Vysvětlení příkazu

VARIB	Proměnná pro status:	TRUE: kruh nalezen FALSE: kruh nenalezen
PKT[n,2]	Body pro výpočet n = počet bodů (3 nebo 4); 2 = zadávání dvou souřadnic bodů	
ANZ	Počet bodů použitých pro výpočet: 3 nebo 4	
ERG [3]	Proměnná pro výsledek: údaj souřadnic středu kruhu a jeho rádiusu; 0 = abscisa, 1 = ordináta středu kruhu; 2 = rádius	

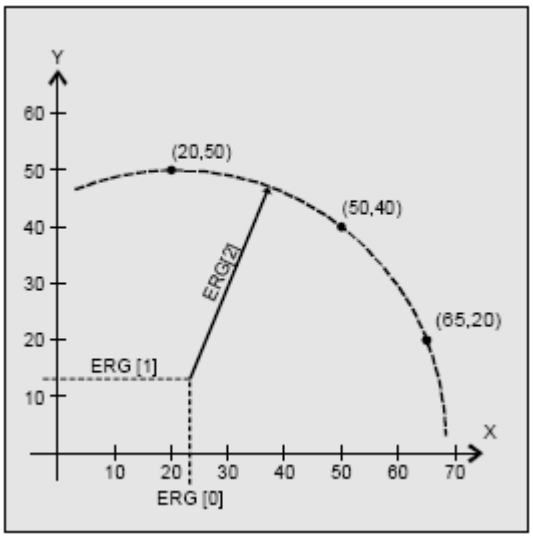
Funkce

Výpočet rádiusu a souřadnic středu kruhu ze tří nebo čtyř bodů ležících na jeho obvodu. Uváděné body musí být odlišné. V případě 4 bodů, které neleží přesně na kružnici, bude pro střed kruhu a rádius zvolena střední hodnota.

© Siemens AG 2004, Všechna práva vyhrazena

Příklad programování

Ze tří bodů má být zjištěno, zda leží na kruhovém oblouku.



`N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)`	Definice bodů
`N20 DEF REAL ERG[3]`	Výsledek
`N30 DEF BOOL STATUS`	Proměnná pro status
`N40 STATUS = CALCDAT(PKT,3,ERG)`	Vyvolání zjištěných údajů o kruhu
`N50 IF STATUS == FALSE GOTOF ERROR`	Skok v případě chyby

© Siemens AG 2004, Všechna práva vyhrazena

Tabulky

15.1 Seznam příkazů 15-586
15.2 Seznam systémových proměnných 15-612

© Siemens AG 2004, Všechna práva vyhrazena

Legenda:

[1] Standardní nastavení na začátku programu (stav systému při dodávce, pokud není naprogramováno nic jiného.

[2] Číslování skupin odpovídá tabulce „Seznam G-funkcí/podmínek dráhy“ v /PG/, Příručka programování – Základy, kapitola 12.3.

[3] Absolutní koncové body: modální; inkrementální koncové body: blokové; jinak modální/blokové v závislosti na syntaktických pravidlech G-funkce.

[4] Jako střed kruhu se parametr IPO chová inkrementálně. Pomocí AC můžete naprogramovat absolutně. Při jiných významech (např. stoupání závitu) je modifikace adresy ignorována.

[5] Klíčové slovo neplatí pro SINUMERIK FM-NC/810D.

[6] Klíčové slovo neplatí pro SINUMERIK FM-NC/810D/NCU571.

[7] Klíčové slovo neplatí pro SINUMERIK 810D.

[8] Uživatel OEM může zapojit další dva druhy interpolace. Uživatel OEM může měnit názvy.

[9] Klíčové slovo platí pouze pro SINUMERIK FM-NC.

[10] Pro tyto funkce je rozšířený způsob zápisu adresy nepřípustný.

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
:	Číslo bloku, hlavní blok (viz N)	0 ... 9999 9999 jen celá čísla bez znaménka	Zvláštní označení bloků namísto N...; tento blok by měl obsahovat všechny příkazy pro kompletní následující obráběcí operaci	např. :20		
A	osa	real			m,b[3]	
A2 [5]	Orientace nástroje: Eulerův úhel	real			b	
A3 [5]	Orientace nástroje: složka směrového vektoru	real			b	
A4 [5]	Orientace nástroje pro začátek bloku	real			b	
A5 [5]	Orientace nástroje pro konec bloku; složka normálového vektoru	real			b	
ABS	Absolutní hodnota	real				
AC	Absolutní zadávání rozměrů	0 ... 359.9999°		X=AC(100)	b	
ACC [5]	Osové zrychlení (acceleration axial)	real, bez znaménka			m	
ACN	Zadání absolutního rozměru pro kruhovou osu, na pozici se najíždí v záporném směru			A=ACN(...) B=ACN(...) C=ACN(...)	b	
ACP	Zadání absolutního rozměru pro kruhovou osu, na pozici se najíždí v kladném směru			A=ACP(...) B=ACP(...) C=ACP(...)	b	
ACOS	Arkus-kosinus (trigonometrická funkce)	real				
ADIS	Vzdálenost zaoblení pro dráhové funkce G1, G2, G3	real, bez znaménka			m	
ADISPOS	Vzdálenost zaoblení pro rychlý posuv G0	real, bez znaménka			m	
ADISPOSA	Velikost tolerančního okna pro IPOBRKA	INT, real, bez znaménka		ADISPOSA=.. nebo ADISPOSA(<osa>[real])	m	
ALF	Úhel rychlého zvedání (angle tilt fast)	real, bez znaménka			m	
AMIRROR	Programovatelné zrcadlové převrácení (additive mirror)			AMIRROR X0 Y0 Z0 ; vlastní blok	b	3
AND	Logická operace AND					
ANG	Úhel konturové křivky				b	
AP	Polární úhel (angle polar)	0 ... ± 360°			m,b [3]	

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
APR	Načtení / zobrazení ochrany proti přístupu (access protection read)	INT, bez znaménka				
APW	Zápis ochrany proti přístupu (access protection write)	INT, bez znaménka				
AR	Úhel kruhové výseče (angle circular)	0 ... 360°			m,b [3]	
AROT	Programovatelné otočení (additive rotation)	Otočení okolo 1. geom. osy: -180°...180° 2.geom. osa: -89.999° ... 90° 3.geom. osa: -180°...180°		AROT X… Y… Z… AROT RPL= ; vlastní blok	b	3
AROTS	Programovatelné otáčení framu o prostorový úhel (additive rotation)			AROTS X... Y... AROTS Z... X... ; vlastní AROTS Y... Z... ; blok AROTS RPL=	b	3
AS	Definice makra	string				
ASCALE	Programovatelná změna měřítka (additive scale)			ASCALE X... Y... Z... ; vlastní blok	b	3
ASIN	Arkus-sinus (trigonometrická funkce)	real				
ASPLINE	Akima spliny				m	1
ATAN2	Arkus tangens2	real				
ATRANS	Aditivní programovatelné posunutí (additive translation)			ATRANS X... Y... Z... ; vlastní blok	b	3
AX	Proměnný identifikátor osy	real			m,b [3]	
AXCSWAP	Další poloha osy kontejneru			AXCSWAP(CTn,CTn+1)		25
AXIS	Datový typ: identifikátor osy		Může obsahovat název souboru.			
AXNAME	Převádí vstupní řetězec na identifikátor osy (get axname)	string	Pokud vstupní řetězec neobsahuje platný název osy, aktivuje se alarm			
AXSTRING	Do SW 5 převádí identifikátor osy na řetězec (get axis as string) Od SW 6 převádí řetězec čísla vřetena (get string)	do SW 5 AXIS, od SW 6 string	Může obsahovat název souboru.	AXSTRING(SPI(n)) Od SW 6 AXSTRING[SPI(n)]		
B	Osa	real			m,b [3]	
B_AND	Bitová logická operace AND					
B_NOT	Bitová negace					
B_OR	Bitová logická operace OR					
B_XOR	Bitová logická operace XOR					
B2 [5]	Orientace nástroje: Eulerův úhel	real			b	
B3 [5]	Orientace nástroje: složka směrového vektoru	real			b	
B4 [5]	Orientace nástroje pro začátek bloku	real			b	
B5 [5]	Orientace nástroje pro konec bloku; složka normálového vektoru	real			b	

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
BAUTO	Definice prvního splinového úseku pomocí následujících tří bodů (begin not a knot)				m	19
BLSYNC	Zpracování rutiny přerušení má být zahájeno až s následujícím přechodem na další blok.					
BNAT [1]	Přirozený přechod na první splinový blok (begin natural)				m	19
BOOL	Datový typ: hodnoty pravdivosti TRUE / FALSE, příp. 0 / 1					
BRISK [1]	Skokové zrychlení po dráze				m	21
BRISKA	Aktivování skokového zrychlení po dráze pro naprogramované osy					
BSPLINE	B-spliny				m	1
BTAN	Tangenciální přechod na první splinový blok (begin tangential)				m	19
C	Osa	real			m,b [3]	
C2 [5]	Orientace nástroje: Eulerův úhel	real			b	
C3 [5]	Orientace nástroje: složka směrového vektoru	real			b	
C4 [5]	Orientace nástroje pro začátek bloku	real			b	
C5 [5]	Orientace nástroje pro konec bloku; složka normálového vektoru	real			b	
CAC	Absolutní najíždění na pozici (coded position: absolute coordinate)		Kódovaná hodnota je tabulkový index; bude se najíždět na hodnotu v tabulce			
CACN	Na hodnotu uloženou v tabulce se bude najíždět absolutně v záporném směru (coded position: absolute negative)		Přípustné pro programování kruhových os jako os polohovacích			
CACP	Na hodnotu uloženou v tabulce se bude najíždět absolutně v kladném směru (coded position: absolute positive)					
CALCDAT	Výpočet rádiusu a středu kruhu ze 3 nebo 4 bodů (calculate circle data)	VAR real [3]	Body se musí lišit.			
CALL	Nepřímé volání podprogramu			CALL BUDVAR		
CALLPATH	Programovatelná cesta pro vyhledávání při volání podprogramu.		Pomocí příkazu CALLPATH lze naprogramovat cestu k existujícímu systému souborů v CK.	CALLPATH(/_N_WKS_DIR/_N_MYWPD/identifikátor podprogramu_SPF)		
CANCEL	Přerušení modální synchronní akce	INT	Přerušení akce s uvedeným identifikačním číslem. Bez parametru budou deaktivovány všechny modální synchronní akce.			
CASE	Podmíněné větvení programu					
CDC	Přímé najíždění na pozici (coded position: direct coordinate)		Viz CAC			

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
CDOF [1]	Vypnutí monitorování kolizí (collision detection OFF)				m	23
CDON	Zapnutí monitorování kolizí (collision detection ON)				m	23
CDOF2	Vypnutí monitorování kolizí (collision detection OFF)		pouze pro CUT3DC		m	23
CFC [1]	Konstantní posuv po kontuře (constant feed at contour)				m	16
CFIN	Konstantní posuv jen u vnitřních zakřivení, nikoli u vnějších zakřivení (constant feed at internal radius)				m	16
CFTCP	Konstantní posuv ve vztažném bodě břitu nástroje (dráha středu) (constant feed in tool-center-point)				m	16
CHAN	Specifikace oblasti platnosti dat		Je k dispozici jednou / kanál			
CHANDATA	Nastavení čísla kanálu pro přístup ke kanálovým datům	INT	Přípustné jen v inicializačním modulu.			
CHAR	Datový typ: znaky ACSII	0, ... , 255				
CHECKSUM	Vytvoří kontrolní součet pole a předá jej jako řetězec (STRING) s pevně definovanou délkou	Max. délka: 32	Vrací řetězec 16 HEX čísel	ERROR = CHECKSUM		
CHF od SW 3.5 CHR	Faseta; hodnota = délka fasety ve směru pohybu Faseta; hodnota = délka fasety (chamfer)	real, bez znaménka			b	
CHKDNO	Kontrolka jednoznačnosti D-čísel					
CIC	Inkrementální najíždění na pozici (coded position> incremental coordinate)		viz CAC			
CIP	Kruhová interpolace přes vnitřní bod			CIP X... Y... Z... I1=... J1=... K1=...	m	1
CLEARM	Vymazání jedné nebo více značek pro koordinaci kanálů	INT 1 – n	Zpracování ve vlastním kanálu není ovlivněno			
CLGOF	Konst. otáčky obrobku pro mimostředné broušení VYP.					
CLGON	Konst. otáčky obrobku pro mimostředné broušení ZAP.					
CLRINT	Deaktivování přerušení	INT	Parametr : číslo přerušení			
CMIRROR	Zrcadlové převrácení na jedné souřadné ose	FRAME				
COARSEA	Konec pohybu při dosažení „hrubého přesného najetí“			COARSEA= .. nebo COARSEA[n]=...	m	
COMPOF [1,6]	Vypnutí kompresoru				m	30
COMPON [6]	Zapnutí kompresoru				m	30
COMPCURV	Zapnutí kompresoru: polynomy s konst. zakřivením				m	30
COMPCAD	Zapnutí kompresoru: CAD program pro jakost povrchu				m	30
CONTPRON	Aktivování přípravy referenční kontury (contour preparation ON)				m	49
COS	kosinus (trigonometrická funkce	real				

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
COUPDEF	Definice svazku elektronické převodovky / svazku synchronních vřeten (couple definition)	String	Chování při přechodu na další blok: NOC: žádné řízení přechodu FINE/COARSE: přechod při „synchronním chodu jemném/hrubém" IPOSTOP: přechod při dokončení superp. pohybu s požad. hodnotou			
COUPDEL	Vymazání svazku elektronické převodovky (couple delete)					
COUPOF	Deaktivování svazku EG / dvojice synchronních vřeten (couple OFF)					
COUPON	Aktivování svazku EG / dvojice synchronních vřeten (couple ON)					
COUPRES	Reset svazku EG (couple reset)		Naprogramované hodnoty neplatné Strojní parametry platné			
CP	Spojitá dráha; pohyb po dráze				m	49
CPRECOF[1,6]	Vypnutí programového přesného dodržení kontury (contour precision OFF)				m	39
CPRECON[6]	Zapnutí programového přesného dodržení kontury (contour precision ON)				m	39
CPROT	Aktivování/deaktivování chráněné oblasti pro specifický kanál					
CPROTDEF	Definice chráněné oblasti pro specifický kanál (channel specific protection area definition)					
CR	Rádius kruhu (circle radius)	real, bez znaménka			b	
CROT	Otočení aktuálního souřadného systému	FRAME	Max. počet parametrů: 6			
CROTS	Programovatelné otočení framu o prostorový úhel (otáčení v uvedených osách)			CROTS X... Y... CROTS Z... X... CROTS Y... Z... ; vlastní CROTS RPL= ; blok	b	
CSCALE	Faktor měřítka pro větší počet os	FRAME	max. počet parametrů: 2 * max. počet os			
CSPLINE	Kubické spliny				m	1
CT	Kruh s tangenciálním přechodem			CT X... Y... Z...	m	1
CTAB	Zjištěná pozice vlečné osy na základě pozice řídící osy z tabulky křivky	real	Pokud parametr 4/5 není naprogramován, standardní měřítko			
CTABDEF	Aktivování definice tabulky					
CTABDEL	Vymazání tabulky křivky					
CTABEND	Deaktivování tabulky křivky					
CTABEXIST	Kontrola tabulky křivky s číslem n		parametr n			
CTABNFO	Počet ještě možných tabulek křivky v paměti		memTyp			

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina 2
CTABFPOL	Počet ještě možných polynomů v paměti		memTyp			
CTABFSEG	Počet ještě možných křivkových segmentů v paměti		memTyp			
CTABID	Umožňuje zjistit číslo n-té tabulky křivky		2 parametry n a memTyp			
CTABINV	Zjištění pozice řídící osy na základě pozice vlečné osy z tabulky křivky	real	viz CTAB			
CTABISLOCK	Vrací zpět stav zablokování tabulky křivky s číslem n		parametr n			
CTABLOCK	Aktivování ochrany proti vymazání a přepsání		3 parametry n, m a memTyp			
CTABMEMTYP	Vrací zpátky paměť, ve které je založena tabulka křivky s číslem n		parametr n			
CTABMPOL	Maximální možný počet polynomů v paměti		memTyp			
CTABMSEG	Maximální možný počet křivkových segmentů v paměti		memTyp			
CTABNO	Počet definovaných tabulek křivky bez ohledu na typ paměti		žádné udávání parametrů			
CTABNOMEM	Počet definovaných tabulek křivky v paměti SRAM nebo DRAM		memTyp			
CTABPERIOD	Vrací zpět periodicitu tabulky s číslem n		parametr n			
CTABPOL	Počet dosud použitých polynomů v paměti		memTyp			
CTABPOLID	Počet křivkových polynomů použitých v tabulce křivky s číslem n		parametr n			
CTABSEG	Počet již použitých křivkových segmentů v daném typu paměti		memTyp			
CTABSEGID	Počet křivkových segmentů použitých v tabulce křivky s číslem n		parametr n			
CTABSEV	Vrací koncovou hodnotu vlečné osy pro segment tabulky křivky		segment je určen pomocí LW	R10 = CTABSEV(LW, n, grad, FOsa, LOsa)		
CTABSSV	Vrací počáteční hodnotu vlečné osy pro segment tabulky křivky		segment je určen pomocí LW	R10 = CTABSSV(LW, n, grad, FOsa, LOsa)		
CTABTEP	Vrací hodnotu řídící osy na konci tabulky křivky		řídící hodnota na konci tabulky křivky	R10 = CTABTEP(n, grad, LOsa)		
CTABTEV	Vrací hodnotu vlečené osy na konci tabulky křivky		vlečná hodnota na konci tabulky křivky	R10 = CTABTEV(n, grad, FOsa)		
CTABMAX	Poskytuje maximální hodnotu vlečné osy z tabulky křivky		vlečná hodnota tabulky křivky	R10 = CTABTMAX(n, FOsa)		
CTABMIN	Poskytuje minimální hodnotu vlečné osy z tabulky křivky		vlečná hodnota tabulky křivky	R10 = CTABTMIN(n, FOsa)		
CTABTSP	Vrací hodnotu řídící osy na začátku tabulky křivky		řídící hodnota na začátku tabulky křivky	R10 = CTABTSP(n, grad, LOsa)		
CTABTSP	Vrací hodnotu vlečné osy na začátku tabulky křivky		vlečná hodnota na začátku tabulky křivky	R10 = CTABTSV(n, grad, FOsa)		
CTABUNLOCK	Deaktivování ochrany proti přepsání a vymazání		3 parametry n, m, a memTyp			
CTRANS	Posunutí počátku pro více os	FRAME	max. 8 os			

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
CUT2D [1]	2 1/2D korekce nástroje (cutter compensation type 2dimensional)				m	22
CUT2DF	2 1/2D korekce nástroje (cutter compensation type 2dimensional); korekce nástroje se započítává vzhledem k aktuálnímu framu (šikmá rovina)				m	22
CUT3DC[5]	3D korekce nástroje, obvodové frézování (cutter compensation type 3dimensional circumference)				m	22
CUT3DCC[5]	3D korekce nástroje, obvodové frézování s omezujícími plochami (cutter compensation type 3dimensional circumference)				m	22
CUT3DCCD[5]	3D korekce nástroje, obvodové frézování s omezujícími plochami s diferenčním nástrojem (cutter compensation type 3dimensional circumference)				m	22
CUT3DF[5]	3D korekce nástroje, čelní frézování (cutter compensation type 3dimensional face)				m	22
CUT3DFF[5]	3D korekce nástroje, čelní frézování s konstantní orientací nástroje v závislosti na aktivním framu (cutter compensation type 3dimensional face frame)				m	22
CUT3DFS[5]	3D korekce nástroje, čelní frézování s konstantní orientací nástroje nezávisle na aktivním framu (cutter compensation type 3dimensional face				m	22
CUTCONOF[1]	Deaktivování konstantní korekce rádiusu				m	40
CUTCONON	Aktivování konstantní korekce rádiusu				m	40
D	Číslo korekce nástroje	1 ... 9 od SW 3.5 1 .. 32 000	obsahuje korekční parametry pro určitý nástroj T...: D0 → hodnoty korekce pro nástroj	D...		
DC	Údaj absolutního rozměru pro kruhové osy, na pozici se najíždí přímo			A=DC(...) B=DC(...) C=DC(...) SPOS=DC(...)	b	
DEF	Definice proměnných	INT, bez znaménka				
DEFAULT	Větev ve větvení CASE		Bude použita pro skok, pokud výraz neodpovídá žádné z uvedených hodnot			
DEFINE	Definice makra					
DELAYFSTON	Definice začátku oblasti zpožděného zastavení (DELAY Feed Stop ON)		Implicitní, je-li aktivní G331/G332		m	
DELAYFSTOF	Definice konce oblasti zpožděného zastavení (DELAY Feed Stop OFF)				m	
DELDTG	Vymazání zbytkové dráhy (Delete distance to go)					
DELETE	Vymazání uvedeného souboru. Název souboru může být uveden s cestou a názvem souboru.		Může vymazat všechny soubory			
DELT	Vymazání nástroje		Duplo-číslo může odpadnout.			
DIAMCYCOF	Zapnutí programování rádiusu pro G90/G91. Pro vypisování zůstává aktivní naposled aktivní G-kód této skupiny.		Programování rádiusu, naposled aktivní G-kód		m	29

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
DIAMOF[1]	Vypnutí programování průměrů (Diametral programming OFF)		Program. rádiusů pro G90/G91		m	29
DIAMON	Zapnutí programování průměrů (Diametral programming ON)		Program. průměrů pro G90/G91		m	29
DIAM90	Programování průměrů pro G90, rádiusů pro G91				m	29
DILF	Délka pro rychlé pozvednutí				m	
DISABLE	Deaktivování přerušení					
DISC	Převýšení přechodové kružnice – korekce rádiusu nástroje	0 ... 100			m	
DISPLOF	Potlačení vypisování aktuálního bloku (Display OFF)					
DISPR	Rozdíl dráhy pro zpětné polohování	real, bez znaménka			b	
DISR	Vzdálenost pro zpětné polohování	real, bez znaménka			b	
DITE	Výběr závitu	real			m	
DITS	Náběh závitu	real			m	
DIV	Celočíselné dělení					
DL	Korekce čísla nástroje	INT			m	
DRFOF	Vypnutí posunutí ručním kolečkem (DRF)				m	
DRIVE [9]	Zrychlení po dráze závislé na rychlosti				m	21
DRIVEA	Aktivování zalomené charakteristiky zrychlení pro naprogramované osy					
DZERO	Nastavení D-čísel všech nástrojů jednotky TO přiřazené kanálu jako neplatné					
EAUTO	Definice posledního splinového úseku pomocí posledních 3 bodů (end not a knot)				m	20
EGDEF	Definice elektronické převodovky (Electronic gear define)		Pro 1 vlečnou osu a max. 5 řídících os			
EGDEL	Vymazání definice vazby pro vlečnou osu (Electronic gear delete)		Spouští zastavení předběžného zpracování			
EGOFC	Kontinuální vypnutí elektronické převodovky (Electronic gear OFF continuous)					
EGOFS	Selektivní vypnutí elektronické převodovky (Electronic gear OFF selective)					
EGON	Aktivování elektronické převodovky (Electronic gear ON)		bez synchronizace			
EGONSYN	Aktivování elektronické převodovky (Electronic gear ON synchronized)		se synchronizací			
EGONSYNE	Aktivování elektronické převodovky se zadáním režimu najíždění (Electronic gear ON synchronized)		se synchronizací			
ELSE	Větvení programu, pokud podmínka IF není splněna					
ENABLE	Aktivování přerušení					
ENAT [1,7]	Přirozený křivkový přechod na následující blok posuvu (end natural)				m	20
ENDFOR	Koncový řádek pro počítací smyčku FOR					
ENDIF	Koncový řádek pro větvení IF					
ENDLOOP	Koncový řádek pro nekonečnou programovou smyčku LOOP					

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
ENDPROC	Koncový řádek programu s počátečním řádkem PROC					
ENDWHILE	Koncový řádek smyčky WHILE					
ETAN	Tangenciální křivkový přechod na následující blok posuvu na začátku splinů (end tangential)				m	20
EVERY	Uskutečnění synchronní akce, pokud se stav podmínky změní z FALSE na TRUE					
EXECSTRING	Předávání řetězcové proměnné s řádkem programu, který má být proveden.		Nepřímé řádky programu	EXECSTRING(MFCT1<<M4711)		
EXECTAB	Zpracování prvky z tabulky pohybů (Execute table)					
EXECUTE	Zpracování programu		Přepnutí zpátky z režimu přípravy zpracování nebo po sestavení chráněné oblasti do normálního zpracování programu.			
EXP	Exponenciální funkce e^x.	real				
EXTCALL	Zpracování externího podprogramu		Načtení programu z MI v režimu "Externí zpracování“.			
EXTERN	Uvedení podprogramu s předáváním parametrů					
F	Hodnota posuvu (ve spojení s G4 se pomocí F programuje také doba prodlevy)	0.001 ... 99999.999	Rychlost pohybu nástroje/obrobku po dráze; jednotky mm/min nebo mm/ot v závislosti na G94 nebo G95	F=100 G1 ...		
FA	Osový posuv (feed axial)	0.001 ... 999999.999 mm/min, stupňů/min; 0.001 ... 39999.9999 palců/min		FA[X]=100	m	
FAD	Posuv při přísuvu pro měkké najíždění a odjíždění	real, bez znaménka				
FALSE	Logická konstanta: nepravda	BOOL	Může být nahrazeno konstantou INT 0			
FCTDEF	Definice polynomické funkce		Bude se vyhodnocovat ve fcích SYNFCT nebo PUTFTOCF			
FCUB [6]	Proměnný posuv podle kubického splinu (feed cubic)		Působí na posuv s G93 a G94		m	37
FD	Posuv po dráze pro korekci ručním kolečkem (feed DRF)	real, bez znaménka			b	
FDA	Osový posuv po dráze pro korekci ručním kolečkem (feed DRF)	real, bez znaménka			b	
FENDNORM	Vypnutí zpožďování v rozích				m	57
FFWOF [1]	Vypnutí dopředné regulace (feed forward OFF)				m	24
FFWON	Zapnutí dopředné regulace (feed forward ON)				m	24

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina[2]
FIFOCTRL	Řízení zásobníku předběžného zpracování				m	4
FIFOLEN	Programovatelná hloubka předběžného zpracování (Preprocessing depth)					
FL	Mezní rychlost pro synchronní osy	real, bez znaménka	Platí jednotka nastavená příkazy G93,G94,G95	FL[osa]=...	m	
FLIN[6]	Lineárně proměnný posuv (feed linear)		Platí na posuv s G93 a G94		m	37
FMA	Větší počet osových posuvů (feed multiple axial)	real, bez znaménka			m	
FNORM[1,6]	Normální posuv podle DIN 66015 (feed normal)				m	37
FOCOF	Deaktivování najíždění s omezeným momentem/silou				m	
FOCON	Aktivování najíždění s omezeným momentem/silou				m	
FOR	Počítací smyčka s pevným počtem průchodů					
FORI1	Posuv pro otáčení orientačního vektoru na velké kružnici				m	
FORI2	Posuv pro superponované otáčení okolo otočeného orientačního vektoru				m	
FP	Pevný bod: číslo pevného bodu, na který se má najíždět	Integer, bez znaménka		G75 FP=1	b	
FPO	Průběh posuvu naprogramovaný pomocí polynomu	real	koeficienty kvadratického nebo kubického polynomu			
FPR	Označení kruhové osy	0.001... 999999.999		FPR (kruhová osa)		
FRAME	Datový typ pro definici souřadných systémů		Pro každou geometrickou osu obsahuje:posunutí, otočení, úhel střihu změnu měřítka, zrcadlové převrácení Pro pomocnou osu: Posunutí, změna měřítka, zrcadlové převrácení			
FRC	Posuv pro rádius a fasetu				b	
FRCM	Modální posuv pro rádius a fasetu				m	
FTOC	Změna jemné korekce nástroje		V závislosti na polynomu 3. stupně definovaného pomocí FCTDEF			
FTOCOF[1,6]	Vypnutí online působící jemné korekce nástroje (fine tool offset OFF)				m	23
FTOCON[6]	Zapnutí online působící jemné korekce nástroje (fine tool offset ON)				m	23
FXS	Zapnutí najíždění na pevný doraz (fixed stop)	INT, bez znaménka	1 = aktivování 0 = deaktivování		m	
FXST	Hranice momentu pro najíždění na pevný doraz (fixed stop torque)	%	Volitelný údaj		m	
FXSW	Monitorované okno pro najíždění na pevný doraz (fixed stop window)	mm, palce, stupně	volitelný údaj			

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
G	G-funkce (podmínky dráhy) G-funkce jsou rozděleny do G-skupin. V jednom bloku může být napsána jen jedna G-funkce z G-skupiny. G-funkce může mít modální platnost (do odvolání jinou funkcí ze stejné skupiny) nebo platí jen pro blok, ve kterém se nachází (bloková působnost).	Pouze celá čísla, předem definované hodnoty		G...		
G0	Lineární interpolace rychlým posuvem (pohyb rychlým posuvem)		Příkazy pohybu	G0 X... Z...	m	1
G1[1]	Lineární interpolace s pracovním posuvem (přímková interpolace)			G1 X... Z... F...	m	1
G2	Kruhová interpolace ve směru hodinových ručiček			G2 X... Z... I... K... F... ; střed a koncový bod G2 X... Z... CR=... F... ; rádius a konc. bod G2 AR=... I... K... F... ; úhel výseče a střed G2 AR=... X... Z... F... ; úhel výseče a konc. bod	m	1
G3	Kruhová interpolace proti směru hodinových ručiček			G3 ... ; jinak jako G2	m	1
G4	Doba prodlevy určená časově		speciální pohyb	G4 F... ; doba prodlevy v sekundách G5 S... ; doba prodlevy v otáčkách vřetena vlastní blok	b	2
G9	Přesné najetí – snižování rychlosti				b	11
G17[1]	Volba pracovní roviny X/Y		směr přísuvu Z		m	6
G18	Volba pracovní roviny Z/X		směr přísuvu Y		m	6
G19	Volba pracovní roviny Y/Z		směr přísuvu X		m	6
G25	Dolní ohraničení pracovního pole		přiřazení hodnoty v kanál. osách	G25 X.. Y.. Z.. ; vlast. blok	b	3
G26	Horní ohraničení pracovního pole			G26 X.. Y.. Z.. ; vlast. blok	b	3
G33	Závitová interpolace s konstantním stoupáním	0.001 ... 2000.00 mm/ot	příkaz pohybu	G33 Z... K... SF=... ; válcový závit G33 X... I... SF=... ; rovinný závit G33 Z... X... K... SF=... ; kuželový závit (dráha v ose Z větší jak dráha v ose X) G33 Z... X... I... SF=... ; kuželový závit (dráha v ose X větší jak dráha v ose Z)	m	1
G34	Lineárně degresivní změna rychlosti [mm/ot^2]		příkaz pohybu	G34 Z.. K.. F_{ZU}..	m	1
G35	Lineárně progresivní změna rychlosti [mm/ot^2]		příkaz pohybu	G35 Z.. K.. F_{AB}..	m	1
G40 [1]	Vypnutí korekce rádiusu nástroje				m	7
G41	Korekce rádiusu nástroje vlevo od kontury				m	7
G42	Korekce rádiusu nástroje vpravo od kontury				m	7
G53	Potlačení aktuálního posunutí počátku (blokové)		vč. programovatelných posunutí		b	9
G54	1. nastavitelné posunutí počátku				m	8
G55	2. nastavitelné posunutí počátku				m	8

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
G56	3. nastavitelné posunutí počátku				m	8
G57	4. nastavitelné posunutí počátku				m	8
G58	Osové programovatelné posunutí počátku - absolutní				b	3
G59	Osové programovatelné posunutí počátku - aditivní				b	3
G60 [1]	Přesné najetí – snížení rychlosti				m	10
G62	Zpoždění na vnitřních rozích při aktivní korekci rádiusu nástroje (G41, G42)		pouze spolu s režimem řízení pohybu po dráze	G62 Z... G1	m	57
G63	Vrtání závitu s vyrovnávací hlavičkou			G63 Z... G1	b	2
G64	Přesné najetí – režim řízení pohybu po dráze				m	10
G70	Zadávání rozměrů v palcích (délky)				m	13
G71 [1]	Zadávání rozměrů v metrických jednotkách (délky)				m	13
G74	Najíždění na referenční bod		osy stroje	G74 X.. Z.. ; vlastní blok	b	2
G75	Najíždění na pevný bod			G75 FP=... X1=... Z1=... ; vlastní blok	b	2
G90 [1]	Zadávání absolutních rozměrů			G90 X... Y... Z... (...) Y=AC(...) nebo X=AC(...) Z=AC(...)	m b	14
G91	Zadávání inkrementálních rozměrů			G91 X... Y... Z... nebo X=IC(...) Y=IC(...) Z=IC(...)	m b	14
G93	Časově reciproční posuv 1/min		provádění bloku: čas	G93 G01 X... F...	m	15
G94 [1]	Lineární posuv F v mm/min, v palcích/min nebo °/min				m	15
G95	Otáčkový posuv F v mm/ot nebo palcích/ot				m	15
G96	Zapnutí konstantní řezné rychlosti			G96 S... LIMS=... F...	m	15
G97	Vypnutí konstantní řezné rychlosti				m	15
G110	Programování pólu vzhledem k poslední naprogramované požadované poloze			G110 X... Y... Z...	b	3
G111	Programování pólu vzhledem k počátku aktuální souřadné soustavy obrobku			G111 X... Y... Z...	b	3
G112	Programování pólu vzhledem k poslednímu platnému pólu			G112 X... Y... Z...	b	3
G140 [1]	Směr najíždění WAB definován příkazy G41/G42				m	43
G141	Směr najíždění WAB vlevo od kontury				m	43
G142	Směr najíždění WAB vpraod kontury				m	43
G143	Směr najíždění WAB v závislosti na tečně				m	43
G147	Měkké najíždění po přímce				b	2
G148	Měkké odjíždění po přímce				b	2
G153	Potlačení aktuálního framu včetně základního framu				b	9
G247	Měkké najíždění po čtvrtkruhu				b	2
G248	Měkké odjíždění po čtvrtkruhu				b	2
G290	Aktivování přepnutí na režim SINUMERIK				m	47
G291	Aktivování přepnutí na režim FANUC				m	47
G331	Vrtání závitu	± 0.001 ... 2000.00 mm/ot	Příkaz pohybu		m	1
G332	Zpětný pohyb (vrtání závitu)				m	1
G340 [1]	Prostorový najížděcí blok (hloubka a v rovině stejné – spirála)		působí pro měkké na-/odjíždění		m	44
G341	Napřed přísuv v kolmé ose, pak najíždění v rovině				m	44

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
G347	Měkké najíždění po půlkruhu				b	2
G348	Měkké odjíždění po půlkruhu				b	2
G450 [1]	Přechodový kruh		chování v rozích při korekci rádiusu nástroje		m	18
G451	Průsečík ekvidistantních drah				m	18
G460 [1]	Chování při najíždění a odjíždění při WRK				m	48
G461	Chování při najíždění a odjíždění při WRK				m	48
G462	Chování při najíždění a odjíždění při WRK				m	48
G500 [1]	Vypnutí všech nastavitelných framů, pokud v G500 není žádná hodnota				m	8
G505 ... G599	5. ... 99. nastavitelné posunutí počátku				m	8
G601 [1]	Přechod na další blok při jemném přesném najetí		v platnosti jen s G60 nebo G9 s programovatelným zaoblením rohů		m	12
G602	Přechod na další blok při hrubém přesném najetí				m	12
G603	Přechod na další blok při konci bloku IPO				m	12
G641	Přesné najetí – režim řízení pohybu po dráze			G641 ADIS=...	m	10
G642	Zaoblení rohů s axiální přesností				m	10
G643	Interní blokové zaoblení rohů				m	10
G644	Zaoblení rohů se specifikovanou dynamikou osy				m	10
G621	Rohové snížení rychlosti na všech rozích		jen s režimem řízení pohybu po dráze	G621 ADIS=...	m	57
G700	Udávání rozměrů v palcích nebo v palcích/min (délky + rychlosti + systémové proměnné)				m	13
G710 [1]	Udávání rozměrů v metrických jednotkách v mm a v mm/min (délky + rychlosti + systémové proměnné)				m	13
G810 [1] ... G819	G-skupina vyhrazená pro uživatele OEM					31
G820 [1] ... G829	G-skupina vyhrazená pro uživatele OEM					32
G931	Zadávání posuvu dobou pohybu		Doba pohybu		m	15
G942	Zmrazení lineárního posuvu a konstantní řezné rychlosti nebo otáček vřetena				m	15
G952	Zmrazení otáčkového posuvu a konstantní řezné rychlosti nebo otáček vřetena				m	15
G961	Zapnutí konstantní řezné rychlosti (jako u G94)		typ posuvu	G961 S... LIMS=... F...	m	15
G962	Lineární posuv nebo otáčkový posuv a konstantní řezná rychlost				m	15
G971	Vypnutí konstantní řezné rychlosti (jako u G94)		typ posuvu		m	15
G972	Zmrazení lineárního nebo otáčkového posuvu a konstantních otáček vřetena				m	15
GEOAX	Přiřazení geometrickým osám 1 – 3 nové kanálové osy		bez parametrů: platí definice pomocí MD			
GET	Obsazení osy stroje		Osa musí být ve druhém kanálu uvolněna pomocí RELEASE			
GETD	Přímé obsazení osy stroje		viz GET			
GETACTT	Stanovení aktivního nástroje ze skupiny nástrojů téhož názvu					

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
GETSELT	Dodání předem zvoleného T-čísla					
GETT	Přiřazení T-čísla názvu nástroje					
GOTOF	Příkaz skoku dopředu (směrem ke konci programu)					
GOTOB	Příkaz skoku dozadu (směrem k začátku programu)					
GOTO	Příkaz skoku napřed dopředu a potom dozadu (napřed směrem ke konci a potom směrem k začátku programu)					
GOTOC	Potlačení alarmu 14080: Cíl skoku nenalezen		viz GOTO			
GWPSOF	Deaktivování konstantní obvodové rychlosti brusného kotouče			GWPSOF (T-číslo)	b	
GWPSON	Aktivování konstantní obvodové rychlosti brusného kotouče			GWPSON (T-číslo)	b	
H..	Výstup pomocných funkcí do PLC		real/INT	nastavitelné pomocí MD (výrobce stroje)	H100 nebo H2=100	
I [4]	Interpolační parametr	real			b	
I1	Souřadnice vnitřního bodu	real			b	
IC	Zadávání inkrementálních rozměrů	0 ... ±99999.999		X=IC(10)	b	
IDS	Označení statické synchronní akce					
IF	Úvod podmíněného skoku		Struktura IF – ELSE - ENDIF			
INDEX	Stanovení indexu znaku ve vstupním řetězci	0, ... INT	Řetězec: 1. par., znak: 2. param.			
INIT	Aktivování modulu pro zpracování v jednom kanálu					
INT	Datový typ: celočíselná hodnota se znaménkem	$-(2^{31} - 1)$ ... $2^{31} - 1$				
INTERSEC	Výpočet průsečíku mezi dvěma konturovými prvky	VAR REAL [2]	Chybový status: BOOL			
IP	Proměnný interpolační parametr	real				
IPOBRKA	Pohybové kritérium od okamžiku aktivování brzdné charakteristiky		Brzdná charakteristika 100% až 0%	IPOBRKA=... nebo IPOBRKA(<osa>[,real])	m	
IPOENDA	Pohybové kritérium při dosažení „IPO-Stop“			IPOENDA=... nebo IPOENDA[n]	m	
IPTRLOCK	Zmrazení začátku vyhledatelného úseku programu na následujícím funkčním bloku stroje		Uvolnění ukazatele přerušení		m	
IPTRUNLOCK	Zmrazení konce vyhledatelného úseku programu na aktuálním bloku k okamžiku přerušení		Stanovení ukazatele přerušení		m	
ISAXIS	Kontrola, zda geometrická osa 1 – 3 uvedená jako parametr je k dispozici	BOOL				
ISD	Hloubka zajíždění (insertion depth)	real			m	
ISFILE	Kontrola, zda soubor v uživatelské paměti NCK je k dispozici	BOOL	Vrací výsledek typu BOOL	RESULT=ISFILE(„Tsetfile) IF (RESULT==FALSE)		
ISNUMBER	Kontrola, zda vstupní řetězec může být převeden na číslo	BOOL	Převod vstupního řetězce na číslo			
ISVAR	Kontrola, zda předávaný parametr obsahuje proměnnou známou NC	BOOL	Strojní parametry nastavované parametry a proměnné, jako jsou GUD			

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
J [4]	Interpolační parametr	real			b	
J1	Souřadnice vnitřního bodu	real			b	
JERKA[6]	Aktivování chování zrychlení nastaveného pomocí MD pro naprogramované osy				m	
K [4]	Interpolační parametr	real			b	
K1	Souřadnice vnitřního bodu	real			b	
KONT	Objíždění kontury s korekcí nástroje				m	17
KONTC	Najíždění/odjíždění se spojitým polynomickým zakřivením				m	17
KONTT	Najíždění/odjíždění s spojitým tangenciálním polynomem				m	17
L	Číslo podprogramu	INT, až 7 číslic		L10	b	
LEAD [5]	Úhel stoupání	real			m	
LEADOF	Deaktivování spojení řídící hodnotou					
LEADON	Aktivování spojení řídící hodnotou					
LFOF [1]	Vypnutí zastavení před řezáním závitu				m	41
LFON	Zapnutí zastavení před řezáním závitu				m	41
LFPOS	Pozvednutí osy na pozici				m	46
LFTXT [1]	Tangenciální směr nástroje při zvedání				m	46
LFWP	Směr nástroje při zvedání není tangenciální				m	46
LIFTFAST	Rychlé pozvednutí před vyvoláním rutiny přerušení					
LIMS	Omezení otáček (limit spindle speed) u příkazů G96/G961 a G97	0.001 ... 99999.999			m	
LN	Přirozený logaritmus	real				
LOCK	Zablokování synchronní akce s identifikačním číslem (zastavení technologického cyklu)					
LOG	Desítkový logaritmus	real				
LOOP	Úvod nekonečné smyčky		Struktura: LOOP – ENDLOOP			
M...	Spínací funkce	0 ... 999 999 999	max. 5 volných M-funkcí může být přiřazeno výrobcem stroje			
M0 [10]	Programové zastavení					
M1 [10]	Volitelné zastavení					
M2 [10]	Konec hlavního programu s návratem na počátek programu					
M3	Směr otáčení vpravo pro hlavní vřeteno					
M4	Směr otáčení vlevo pro hlavní vřeteno					
M5	Zastavení pro hlavní vřeteno					
M6	Výměna nástroje					
M17 [10]	Konec podprogramu					
M19	Polohování vřetena					
M30 [10]	Konec programu, jako M2					
M40	Automatická volba stupně převodovky					
M41 ... M45	Stupně převodovky 1 ... 5					
M70	Přechod do osového režimu					

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina 2
MASLDEF	Definice svazku os Master/Slave					
MASLDEL	Rozpojení svazku os Master/Slave a vymazání definice svazku					
MASLOF	Deaktivování dočasné vazby					
MASLOFS	Deaktivování dočasné vazby s automatickým zastavením osy Slave					
MASLON	Aktivování dočasné vazby					
MCALL	Modální volání podprogramu					
MEAC	Kontinuální měření bez mazání zbytkové dráhy	INT, bez znaménka			B	
MEAFRAME	Výpočet framu ze změřených bodů	FRAME				
MEAS	Měření se spínací sondou	INT, bez znaménka			B	
MEASA	Měření s vymazáním zbytkové dráhy				b	
MEAW	Měření se spínací sondou bez vymazání zbytkové dráhy (measure without deleting distance to go)	INT, bez znaménka			B	
MEAWA	Měření bez vymazání zbytkové dráhy				b	
MI	Přístup k datům framu: zrcadlové převrácení					
MINDEX	Stanovení indexu znaku ve vstupním řetězci	0, ... , INT	Řetězec: 1. par., znak: 2. param.			
MIRROR	Programové zrcadlové převrácení			MIRROR X0 Y0 Z0 ; vlastní blok	b	3
MMC	Vyvolání dialogového okna na MMC/HMI interaktivně z výrobního programu	STRING				
MOD	Dělení modulo					
MOV	Spušění polohovací osy (start moving positioning axis)	real				
MSG	Programovatelné poznámky			MSG(„Hlaseni“)	m	
N	Číslo bloku – vedlejší blok	0 ... 9999 9999 jen celá čísla bez znaménka	Může se používat pro označení bloků čísly; nachází se na začátku bloku	např. N20		
NCK	Specifikace oblasti platnosti dat		K dispozici jednou na jednu NCK			
NEWCONF	Převzetí nových strojních parametrů. Odpovídající strojní parametry jsou aktivovány.		Možno také z HMI pomocí programového tlačítka			
NEWT	Založení nového nástroje		Duplo-číslo může odpadnout			
NORM [1]	Normální nastavení v počátečním a koncovém bodě při korekci nástroje				m	17
NOT	Logická negace					
NPROT	Aktivování/deaktivování chráněné oblasti specifické pro stroj					
NPROTDEF	Definice chráněné oblasti specifické pro stroj (NCK specific protection area definition]					
NUMBER	Přeměna vstupního řetězce v číslo		real			

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Sku-pina 2
OEMIPO1 6,8	Interpolace OEM 1				m	1
OEMIPO2 6,8	Interpolace OEM 2				m	1
OF	Klíčové slovo pro větvení CASE					
OFFN	Přídavek rozměru pro naprogramovanou konturu			OFFN=5		
OMA1 6	Adresa OEM 1	real			m	
OMA1 6	Adresa OEM 2	real			m	
OMA1 6	Adresa OEM 3	real			m	
OMA1 6	Adresa OEM 4	real			m	
OMA1 6	Adresa OEM 5	real			m	
OFFN	Korekce offsetu - normální	real			m	
OR	Logické NEBO					
ORIC 1,6	Změny orientace na vnějších rozích jsou superponovány vkládaným kruhovým blokem (orientace se mění spojitě)				m	27
ORID 6	Změny orientace se provádějí před kruhovým blokem (orientace se mění nespojitě).				m	27
ORIAXPOS	Úhel orientace pomocí virtuálních orientačních os s polohováním kruhové osy				m	50
ORIEULER	Úhel orientace pomocí Eulerova úhlu				m	50
ORIAXES	lineární interpolace os stroje nebo orientačních os		Konc. orientace: udání vektoru A3, B3, C3 nebo Eulerova/RPY úhlu A2, B2, C2 Další zadání: vektory otočení A6, B6, C6 úhel kužele ve st. 0<NUT<180° přechodové vektory A7,B7,C7 2. styčný bod nástroje XH,YH,ZH	Zadávání parametrů: Směrové vektory normované A6=0, B6=0, C6=0 Úhel oblouku je zadán jako úhel posuvu pomocí NUT=... NUT=+... pro ≤ 180° NUT=-... pro ≥ 180° Přechodová orientace normovaná A7=0 B7=0 C7=1	m	51
ORICONCW	Interpolace na kruhové plášťové ploše ve směru hodinových ručiček				m	51
ORICONCCW	Interpolace na kruhové plášťové ploše proti směru hodinových ručiček				m	51
ORICONIO	Interpolace na kruhové plášťové ploše s udáním přechodové orientace				m	51
ORICONTO	Interpolace na kruhové plášťové ploše v tangenciálním přechodu				m	51
ORICURVE	Interpolace orientace s udáním pohybu dvou kontaktních bodů nástroje				m	51
ORIPLANE	Interpolace v rovině (odpovídá ORIVECT) kruhová interpolace s velkým rádiusem				m	51
ORIPATH	Cesta orientace nástroje vztažená na dráhu		Sada pro práci s transformacemi viz /FB/ TE4		m	51
ORIRPY	Úhel orientace prostřednictvím úhlu RPY				m	50
ORIS 5	Změna orientace (orientation smoothing factor)	real	vztahuje se na dráhu		m	
ORIVECT	Kruhová interpolace s velkým rádiusem (identická s ORIPLANE)				m	51
ORIVIRT1	Úhel orientace pomocí virt. orientačních os (definice 1)				m	50
ORIVIRT2	Úhel orientace pomocí virt. orientačních os (definice 1)				m	50
ORIMKS 6	Orientace nástroje v souřadném systému stroje (tool orientation in machine coordinate system)				m	25
ORIWKS 1,6	Orientace nástroje v souřadném systému obrobku (tool orientation in workpiece coordinate system)				m	25

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
OS	Oscilační pohyb ZAP/VYP	INT, bez znaménka			m	
OSC	Konstantní vyhlazení orientace nástroje				m	34
OSCILL	Přiřazení os pro oscilační pohyb – zapnutí oscilačního pohybu		Osa: 1 – 3 osy pro přísuv		m	
OSCTRL	Volby oscilačního pohybu	INT, bez znaménka			m	
OSE	Oscilační pohyb: koncový bod				m	
OSNCS	Oscilační pohyb: počet vyjiskřovacích cyklů (oscillating: number of spark out cycles)				m	
OSOF [1,6]	Vypnutí vyhlazování orientace nástroje				m	
OSP1	Oscilační pohyb: levý bod obratu (oscillating: Position 1)	real			m	
OSP2	Oscilační pohyb: pravý bod obratu (oscillating: Position 1)	real			m	
OSS [6]	Vyhlazení orientace nástroje na konci bloku				m	34
OSSE [6]	Vyhlazení orientace nástroje na počátku a konci bloku				m	34
OST1	Oscilační pohyb: zastavení v levém bodu obratu	real			m	
OST2	Oscilační pohyb: zastavení v pravém bodu obratu	real			m	
OVR	Korekce otáček (override)	1 ... 200%			m	
OVRA	Osová korekce otáček (override)	1 ... 200%			m	
P	Počet průchodů podprogramu	1 ... 9999, INT, bez znaménka		např. L781 P... ; vlastní blok		
PCALL	Volání podprogramu s absolutním zadáním cesty a s předáváním parametrů		žádná absolutní cesta, chování jako CALL			
PDELAY-OF [6]	Vypnutí lisování se zpožděním (punch with delay OFF)				m	36
PDELAY-ON [1,6]	Zapnutí lisování se zpožděním (punch with delay ON)				m	36
PL	Délka intervalu parametru	real, bez znaménka			B	
PM	za minutu			posuv za minutu		
PO	Polynom	real, bez znaménka			B	
POLF	Pozice LIFTFAST	real, bez znaménka	Geometrická osa ve WKS, jinak MKS	POLF[Y]=10 ; cílová pozice osy zpětného pohybu	m	
POLFA	Návratová pozice pro jednotlivé osy, spouští se pomocí $AA_ESR_TRIGGER		pro jednotlivé osy	POLFA(AX1,1,20.0)	m	
POLFMASK	Uvolnění os bez vzájemné závislosti pro zpětný pohyb		vybrané osy	POLFMASK(AX1,AX2,..)	m	
POLFMIN	Uvolnění os s lineární závislostí pro zpětný pohyb		vybrané osy	POLFMIN(AX1,AX2,..)	m	
POLY [5]	Polynomická interpolace				m	1
POLYPATH [5]	Volitelná polynomická interpolace pro skupiny os AXIS nebo VECT			POLYPATH(„AXES“) POLYPATH(„VECT“)	m	1

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
PON [6]	Zapnutí lisování (punch ON)				m	35
PONS [6]	Zapnutí lisování v taktu IPO (punch ON slow)				m	35
POS	Polohování osy			POS[X]=20		
POSA	Polohování osy přes hranici bloku			POSA[Y]=20		
POSP	Polohování v dílčích úsecích (oscilace) (Position Axis in Parts)	real: konc. pozice, délka úseku: INT nepovinné			m	
POT	Mocnina (aritmetická funkce)	real				
PR	na otáčku (per revolution)			otáčkový posuv		
PRESETON	Dosazení skutečné hodnoty pro naprogramované osy		Programuje se vždy jeden identifikátor osy a v následujícím parametru odpovídající hodnota. Je možné pracovat s až 8 osami.	PRESETON(X,10,Y,4.5)		
PRIO	Klíčové slovo pro definici proirity při správě přerušení					
PROC	První příkaz programu			číslo bloku – PROC - identifikátor		
PTP	point to point:: pohyb od bodu k bodu				m	49
PUTFTOC	Jemná korekce nástroje pro paralelní orovnávání					
PUTFTOCF	PutFineToolCorrectionFunctionDependant: Jemná korekce nástroje v závislosti na funkci určené příkazem FCTtDEF pro paralelní orovnávání (continuous dressing)					
PW	Bodová hmotnost (point weight)	real, bez znaménka			B	
QECLRNOF	Deaktivování učení se kompenzace kvadrantové chyby (Quadrant Error Compensation Learning OFF)					
QECLRNON	Aktivování učení se kompenzace kvadrantové chyby (Quadrant Error Compensation Learning ON)					
QU	Rychlý výstup doplňkové (pomocné) funkce					
R...	Početní parametr od SW 5: také jako nastavitelný adresový identifikátor a s numerickým rozšířením	±0.0000001 ... 9999 9999	R10=3; přiřazení R-parametru X=R10; hodn.osy R[R10]=6; nepřímé programování	R10=3 ; přiřazení R-par. X=R10 ; hodnota osy R[R10]=6 ; nepřímé programování		
RDISABLE	Zablokování načítání (Read disable)					
READ	Načtení jednoho nebo více řádků v uvedeném souboru a uložení načtených informací do pole.		Informace jsou k dispozici jako STRING.			
READAL	Načtení alarmu (Read alarm)		Alarmy jsou vyhledávány podle stoupajících čísel			
REAL	Datový typ: proměnná s pohyblivou řádovou čárkou se znaménkem (reálná čísla)	Odpovídá 64-bit. formátu s plovoucí řádovou čárkou				

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
REDEF	Nastavení pro strojní parametry, prvky NC jazyka a systémové proměnné, při kterých uživatelských skupinách se mají vypisovat					
RELEASE	Uvolnění osy stroje		Může být naprogramováno i více os			
REP	Klíčové slovo pro inicializaci všech prvků pole se stejnou hodnotou					
REPEAT	Opakování programové smyčky		dokud není splněna podmínka UNTIL			
REPEATB	Opakování řádku programu		nnn-krát			
REPOSA	Lineární zpětné polohování všech os: Zpětné najíždění na konturu lineárně všemi osami				b	2
REPOSH	Zpětné polohování po půlkruhu: Najetí zpět na konturu po půlkruhu.				b	2
REPOSHA	Zpětné polohování všemi osami po půlkruhu: Najetí zpět na konturu všemi osami po půlkruhu; Geometrické osy po půlkruhu				b	2
REPOSL	Lineární zpětné polohování: Lineární najíždění zpět na konturu.				b	2
REPOSQ	Zpětné polohování po čtvrtkruhu: Najetí zpět na konturu po čtvrtkruhu.				b	2
REPOSQA	Zpětné polohování všemi osami po čtvrtkruhu: Najetí zpět na konturu všemi osami po čtvrtkruhu; Geometrické osy po čtvrtkruhu				b	2
RESET	Reset technologického cyklu		Může být naprogramováno jedno nebo více ID			
RET	Konec podprogramu		Používá se místo M17 – bez výstupu funkce do PLC	RET		
RINDEX	Stanovení indexu znaku ve vstupním řetězci	0, ..., INT	Řetězec: 1. param znak: 2.parametr			
RMB	Opětovné najíždění na začátek bloku (Repos mode begin of block)				m	26
RME	Opětovné najíždění na konec bloku (Repos mode end of block)				m	26
RMI [1]	Opětovné najíždění na bod přerušení (Repos mode interrupt)				m	26
RMN	Opětovné najíždění na následující bod dráhy (Repos mode of nearest orbital block)				m	26
RND	Zaoblení rohu kontury	real, bez znaménka		RND=...	b	
RNDM	Modální zaoblení	real, bez znaménka		RNDM=... RNDM=0 ; vypnutí funkce	m	

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Sku-pina [2]
ROT	Programovatelné otočení (rotation)	Otočení okolo 1. geom. osy -180°..180° 2.geom.osa –89.999°... 90° 3.geom.osa -180°..180°		ROT X... Y... Z... ROT RPL=... ; vlastní blok	b	3
ROTS	Programovatelné otočení framu o prostorový úhel			ROT X.. Y.. ROT Z.. X.. ROT Y.. Z.. ; vlastní blok ROT RPL=	b	3
ROUND	Zaokrouhlení desetinných míst	real				
RP	Polární rádius (radius polar)	real			m, b [3]	
RPL	Otáčení v rovině (rotation plane)	real, bez znaménka			B	
RT	Parametr pro přístup k datům framu: otáčení (Rotation]					
S	Otáčky vřetena nebo (u G4, G96/G961) jiný význam	real, výpis: ±999 999 999.9999, program: ±3.4028ex38	Otáčky vřetena v ot/min G4: doba prodlevy v otáčkách vřetena; G96/G961:řezná rychlost v m/min	S...: otáčky hlavního vřetena S1...: otáčky pro vřeteno 1	m, b	
SAVE	Atribut pro návrat informací při vyvolávání podprogramu		Zachráněny jsou: všechny modální G-funkce a aktuální frame			
SBLOF	Potlačení režimu blok po bloku (Single block OFF)		Následující bloky budou v režimu blok po bloku zpracovány jako jeden blok			
SBLON	Zrušení potlačení režimu blok po bloku (Single block OFF)					
SC	Parametr pro přístup k datům framu: změna měřítka (Scale)					
SCALE	Programovatelná změna měřítka (scale)			SCALE X... Y... Z... ; vlastní blok	b	3
SD	Stupeň splinu (spline degree)	INT, bez znaménka			B	
SEFORM	Strukturovací příkaz ve Step- editoru		Bude se vyhodnocovat ve Step-editoru	SEFORM(<název úseku>, <rovina>, <ikona>)		
SET	Klíčové slovo pro inicializaci všech prvků pole s uvedenými hodnotami					
SETAL	Aktivování alarmu (Set alarm)					
SETDNO	Nastavení nového D-čísla nástroje (T) a jeho břitu					
SETINT	Stanovení, které rutiny přerušení mají být aktivovány, když se aktivuje vstup NCK		Vyhodnocována je hrana 0 → 1			

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
SETM	Nastavení jedné nebo více značek pro potřeby koordinace kanálů		Zpracování ve vlastním kanálu není nijak ovlivněno			
SETMS	Zpětné přepnutí na řídící vřeteno definované strojním parametrem					
SETMS(n)	Vřeteno n má platit jako řídící vřeteno					
SETPIECE	Bude se sledovat počet kusů pro všechny nástroje, které jsou přiřazeny vřetenu		Bez čísla vřetena platí pro hlavní vřeteno			
SF	Posunutí počátečního bodu při řezání závitu (spline offset)	0.0000 ... 359.999°			m	
SIN	Sinus (Trigonometrická funkce)	real				
SOFT	Zrychlení po dráze s omezením trhnutí				m	21
SOFTA	Aktivování zrychlení pro naprogramované osy s omezením trhnutí					
SON [6]	Zapnutí prostřihování (stroke ON)				m	35
SONS [6]	Zapnutí prostřihování v taktu IPO (stroke ON slow)				m	35
SPATH [1]	Referenční dráha pro osy v FGROUP je délka oblouku				m	45
SPCOF	Přepnutí řídícího vřetena nebo vřetena (n) z regulace otáček do polohové regulace			SPCOF SPCOF(n)		
SPCON	Přepnutí řídícího vřetena nebo vřetena (n) z polohové regulace do regulace otáček			SPCON SPCON(n)		
SPIF1 [1,6]	Rychlé vstupy/výstupy NCK pro lisování/prostřihování byte 1 (stroke/punch interface 1)				m	38
SPIF2 [6]	Rychlé vstupy/výstupy NCK pro lisování/prostřihování byte 2 (stroke/punch interface 2)				m	38
SPLINE-PATH	Definice pásma hodnot pro spliny		max. 8 os			
SPOF [1,6]	Vypnutí zdvihu, lisování, vypnutí prostřihování (stroke/punch OFF)				m	35
SPN [6]	Počet dílčích úseků dráhy na jeden blok (stroke/punch number)	integer			b	
SPP [6]	Délka úseku dráhy (stroke/punch path)	integer			m	
SPOS	Poloha vřetena			SPOS=10 nebo SPOS[n]=10	m	
SPOSA	Polohování vřetena přes hranici bloku			SPOSA=5 nebo SPOSA[n]=5	m	
SQRT	Odmocnina (aritmetická funkce)	real				
SR	Zpětný návrat při oscilačním pohybu (sparking out retract path)	real, bez znaménka			B	
SRA	Zpětný návrat oscilační osy při externím vstupním signálu osy (sparking out retract)			SRA[Y]=0.2	m	
ST	Doba vyjiskřování při oscilačním pohybu (sparking out time) pro synchronní akci	real, bez znaménka			B	
STA	Doba vyjiskřování při oscilačním pohybu pro osu (sparking out time axial)				m	

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
START	Spuštěn zvoleného programu ve více kanálech současně z právě zpracovávaného programu.		Žádný vliv ve vlastním kanálu.			
STAT	Poloha kloubu	integer			b	
STARTFIFO	Zpracovávat; souběžně s plněním paměti předběžného zpracování				m	4
STOPFIFO	Zastavení zpracování; plnění paměti předběžného zpracování, dokud není zjištěn příkaz STARTFIFO, naplnění paměti předběžného zpracování nebo konec programu				m	4
STOPRE	Zastavení předběžného zpracování, dokud nejsou zpracovány všechny bloky z hlavního chodu (Stop preprocessing)					
STOPREOF	Deaktivování zastavení předběžného zpracování (Stop preprocessing OFF)					
STRING	Datový typ: řetězec znaků	max. 200 znaků				
STRLEN	Stanovení délky řetězce	INT				
SUBSTR	Stanovení indexu znaku ve vstupním řetězci	real	Řetězec: 1.par. znak: 2.parametr			
SUPA	Potlačení aktuálního posunutí počátku, včetně programovatelných posunutí, systémových framů, posunutí ručním kolečkem (DRF), externích posunutí a superponovaných pohybů				b	9
SYNFCT	Vyhodnocení polynomu v závislosti na podmínce v pohybové synchronní akci	VAR REAL				
SYNR	Načtení proměnné se provádí synchronně, tzn. v okamžiku zpracování					
SYNRW	Načtení a zápis proměnné se provádí synchronně, tzn. v okamžiku zpracování					
SYNW	Zápis proměnné se provádí synchronně, tzn. v okamžiku zpracování					
T	Vyvolání nástroje (výměna jen tehdy, je-li nastaveno strojním parametrem, jinak je třeba příkaz M6)	1 ... 32000	Volání pomocí T-čísla nebo identifikátoru nástroje	např. T3, příp. T=3 např. T="BOHRER"		
TAN	Tangens (trigonometrická funkce)	real				
TANG	Stanovení tečny pro sledování na základě obou uvedených řídících os.					
TANGON	Deaktivování tangenciálního sledování (Tangential follow up mode OFF)					
TANGOF	Aktivování tangenciálního sledování (Tangential follow up mode ON)					
TCARR	Vyžádání držáku nástroje (číslo „m“)	integer	m=0: deaktivování aktivního držáku nástroje	TCARR=1		
TCOABS [1]	Stanovení složek délky nástroje z aktuální orientace nástroje		Potřebné po přezbrojení, např. manuálním nastavením		m	42
TCOFR	Stanovení složek délky nástroje z orientace aktuálního framu				m	42

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
TCOFRX	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj namířený v ose X		Nástroj kolmo na šikmou plochu		m	42
TCOFRY	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj namířený v ose Y		Nástroj kolmo na šikmou plochu		m	42
TCOFRZ	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj namířený v ose Z		Nástroj kolmo na šikmou plochu		m	42
TILT [5]	Úhel naklonění	real			m	
TMOF	Deaktivování monitorování nástroje		T-číslo je třeba tehdy, pokud nástroj s tímto číslem není aktivní. T-číslo=0: vypnutí monitorování pro všechny nástroje	TMOF (T-číslo)		
TMON	Aktivování monitorování nástroje			TMON (T-číslo)		
TO	Označuje koncovou hodnotu pro počítací smyčku FOR					
TOFFOF	Deaktivování on-line korekce délky nástroje					
TOFFON	Aktivování on-line korekce délky nástroje		Údaj trojrozměrného směru korekce	TOFFON (Z,25) se směrem korekce Z, hodnota offsetu 25		
TOFRAME	Aktuální programovatelný frame nastavit na souřadný systém nástroje		Otáčení framu do směru nástroje		m	53
TOFRAMEX	Osa X rovnoběžně se směrem nástroje, vedl. osy Y, Z				m	53
TOFRAMEY	Osa Y rovnoběžně se směrem nástroje, vedl. osy Z, X				m	53
TOFRAMEZ	Osa Z rovnoběžně se směrem nástroje, vedl. osy X, Y				m	53
TOFROT	Vypnutí otáčení framu do směru nástroje				m	53
TOFROT	Osa Z rovnoběžně s orientací nástroje		Otáčení framu ZAP, rotační složka programovatelného framu		m	53
TOFROTX	Osa X rovnoběžně s orientací nástroje				m	53
TOFROTY	Osa Y rovnoběžně s orientací nástroje				m	53
TOFROTZ	Osa Z rovnoběžně s orientací nástroje				m	53
TOLOWER	Písmena v řetězci budou převedena z velkých na malé.					
TOWSTD	Základ. nastavení pro korekce ve směru délky nástroje		Započítávání hodnot opotřebení nástroje		m	56
TOWBCS	Hodnoty opotřebení v základním souřadném systému				m	56
TOWKCS	Hodnoty opotřebení v souřadném systému hlavy nástroje při kinetické transformaci (liší se od MCS otočením nástroje)				m	56
TOWMCS	Hodnoty opotřebení v souřadném systému stroje (MCS)				m	56
TOWTCS	Hodnoty opotřebení v souřadném systému nástroje (vztažný bod držáku nástroje T na jeho držáku)				m	56
TOWWCS	Hodnoty opotřebení v souřadném systému obrobku				m	56
TOUPPER	Písmena v řetězci budou převedena z malých na velké..					
TR	Parametr pro přístup k datům framu: posunutí (tramslation)					
TRAANG	Transformace šikmé osy		Na jeden kanál lze nastavit i více transformací			
TRACEOF	Zkouška s kruhovým formátem: deaktivování předávání hodnoty					
TRACEON	Zkouška s kruhovým formátem: aktivování předávání hodnoty					

© Siemens AG 2004, Všechna práva vyhrazena

Název	Význam	Přiřazení hodnot	Popis, komentář	Syntaxe	modal/ blok.	Skupina [2]
TRACON	Kaskádová transformace (Transformation Concatenated)					
TRACYL	Válec: transformace plášťové plochy		viz TRAANG			
TRAFOOF	Deaktivování transformace			TRAFOOF( )		
TRAILOF	Deaktivování asynchronního vlečení (Trailing OFF)					
TRAILON	Aktivování asynchronního vlečení (Trailing ON)					
TRANS	Programovatelné posunutí (translation)			TRANS X... Y... Z... ; vlastní blok	b	3
TRANSMIT	Polární transformace		viz TRAANG			
TRAORI	4, 5-osová transformace, generická transformace (Transformation oriented)		Aktivování nastavené transformace orientace	Generická transformace TRAORI(1,X,Y,Z)		
TRUE	Logická konstanta: pravda	BOOL	nahraditelná konstantou INT = 1			
TRUNC	Oddělení desetinných míst	real				
TU	Úhel osy	INT		TU=2	b	
TURN	Počet závitů u spirály	0 ... 999			b	
UNLOCK	Uvolnění synchronní akce s daným identifikačním číslem (pokračování technologického cyklu)					
UNTIL	Podmínka pro ukončení smyčky REPEAT					
UPATH	Referenční dráha pro osy v FGROUP je křivkový parametr				m	45
VAR	Klíčové slovo: druh předávané proměnné		Pomocí VAR: volání odkazem			
WAITC	Počkání, dokud nebude splněna podmínka kritéria vazby pro přechod na další blok pro danou osu/vřeteno (wait for couple condition)		Mohou být naprogramovány až 2 osy/vřetena	WAITC(1,1,2)		
WAITM	Čekání na značku v uvedeném kanálu; konec předešlého bloku s přesným najetím			WAITM(1,1,2)		
WAITMC	Čekání na značku v uvedeném kanálu; přesné najetí jen tehdy, pokud ostatní kanály značky ještě nedosáhly			WAITMC(1,1,2)		
WAITP	Čekání na konec posuvu			WAITP(X) ; vlastní blok		
WAITS	Čekání na dosažení pozice vřetena			WAITS (hlavní vřeteno) WAITS (n,n,n)		
WALIMOF	Vypnutí ohraničení pracovního pole (working area limitation OFF)			; vlastní blok	m	28
WALIMON [1]	Zapnutí ohraničení pracovního pole (working area limitation ON)			; vlastní blok	m	28
WHILE	Začátek programové smyčky WHILE		Konec: ENDWHILE			
WRITE	Zápis bloku do systému souborů. Blok bude připojen na konec uvedeného souboru.		Bloky budou vloženy po M30			
X	Osa	real			m, b [3]	
XOR	Logická operace XOR					
Y	Osa	real			m, b [3]	
Z	Osa	real			m, b [3]	

© Siemens AG 2004, Všechna práva vyhrazena

Legenda:

1 Standardní nastavení na začátku programu (stav systému při dodávce, pokud není naprogramováno nic jiného.

2 Číslování skupin odpovídá tabulce „Seznam G-funkcí/podmínek dráhy“ v /PG/, Příručka programování – Základy, kapitola 12.3.

3 Absolutní koncové body: modální; inkrementální koncové body: blokové; jinak modální/blokové v závislosti na syntaktických pravidlech G-funkce.

4 Jako střed kruhu se parametr IPO chová inkrementálně. Pomocí AC můžete naprogramovat absolutně. Při jiných významech (např. stoupání závitu) je modifikace adresy ignorována.

5 Klíčové slovo neplatí pro SINUMERIK FM-NC/810D.

6 Klíčové slovo neplatí pro SINUMERIK FM-NC/810D/NCU571.

7 Klíčové slovo neplatí pro SINUMERIK 810D.

8 Uživatel OEM může zapojit další dva druhy interpolace. Uživatel OEM může měnit názvy.

9 Klíčové slovo platí pouze pro SINUMERIK FM-NC.

10 Pro tyto funkce je rozšířený způsob zápisu adresy nepřípustný.

© Siemens AG 2004, Všechna práva vyhrazena

15.2 Seznam proměnných



Odkaz na literaturu:

Od SW 7.1 je seznam proměnných uveden v samostatném dokumentu: SINUMERIK 840D/ 840Di/810D „Seznamy systémových proměnných“.

© Siemens AG 2004, Všechna práva vyhrazena

Přílohy

A Zkratky A-613
B Pojmy A-622
C Rejstřík A-643
D Příkazy, identifikátory A-653

© Siemens AG 2004, Všechna práva vyhrazena

A Zkratky

A	Výstup
AS	Automatizační systém
ASCII	American Standard Code for Information Exchange
ASIC	Specifický aplikační integrovaný obvod
ASUP	Asynchronní subrutina
AW	Příprava práce
AWL	Seznam proměnných
BA	Provozní režim
BAG	Skupina režimů
BB	Připraven ke spuštění
BuB, B&B	Obsluha a monitorování
BCD	Binárně kódovaná desítková čísla
BCS	Základní souřadný systém
BHG	Ruční ovladač
BIN	Binární soubory
BIOS	Základní vstup/výstupní systém
BOF	Pracovní plocha systému
BOT	Zaváděcí soubory pro SIMODRIVE 611 D
BP	Základní program
BT	Ovládací panel
BTSS	Rozhraní ovládacího panelu
C Bus	Komunikační sběrnice
C1 .. C4	Kanál 1 až kanál 4
CAD	Konstrukce s podporou počítače
CAM	Výroba s podporou počítače
CNC	Počítačově řízené numerické řízení
COM	Communication: komunikace
COR	Rotace souřadnice
CP	Komunikační procesor
CPU	Centrální procesorová jednotka
CR	Znak „carriage return“

© Siemens AG 2004, Všechna práva vyhrazena

CRT	Katodová trubice
CSB	Centrální ovládací panel
CSF	Schéma řídícího systému (metoda programování PLC)
CTS	Clear to Send (signál sériového komunikačního rozhraní)
CUTOM	Korekce rádiusu frézy
DAC	Digitálně-analogový převodník
DB	Datový blok v PLC
DBB	Byte datového bloku v PLC
DBW	Slovo datového bloku v PLC
DBX	Bit datového bloku v PLC
DC	Přímé řízení: Kruhová osa se pohybuje po nejkratší dráze do požadované absolutní pozice v rámci jedné otáčky.
DCD	Carrier Detect
DCE	Datové komunikační zařízení
DDE	Dynamická výměna dat
DEE	Koncové datové zařízení
DIN	Německá průmyslová norma
DIO	Vstup/výstup dat: zobrazování přenosu dat
DIR	Adresář
DLL	Knihovna dynamických spojení: Modul, který je využíván běžícím programem. Často obsahuje úseky programu, které jsou zapotřebí různými programy.
DOE	Zařízení pro přenos dat
DOS	Diskový operační systém
DPM	Paměť se dvěma porty
DPR	Paměť RAM se dvěma porty
DRAM	Dynamická paměť RAM
DRF	Diferenciální funkce otočného snímače (ruční kolečko)
DRY	Dry Run: Posuv zkušebního chodu
DSB	Dekódování jednotlivých bloků
DTE	Datové koncové zařízení
DW	Datové slovo

© Siemens AG 2004, Všechna práva vyhrazena

E	Vstup
E/A	Vstup/výstup
E/R	Napájecí jednotka modulu SIMODRIVE 611 (D)
EIA Code	Zvláštní kód děrné pásky, počet děr na znak je vždy lichý
ENC	Dekodér
EPROM	Paměť EPROM
ERROR	Chyba z tiskárny
FB	Funkční blok
FBS	Plochá obrazovka
FC	Volání funkce: Funkční blok v PLC
FDB	Databáze výrobků
FDD	Pohon posuvu
FDD	Disketová jednotka
FEPROM	Paměť Flash EPROM
FIFO	First-In-First-Out: Paměť, která pracuje bez specifikace adresy, ze které jsou data čtena v pořadí, v jakém byla uložena.
FIPO	Jemný interpolátor
FM	Funkční modul
FM-NC	Funkční modul – numerické řízení
FPU	Jednotka plovoucí řádové čárky
FRA	Blok framu
FRAME	Datový záznam (frame)
FRK	Korekce rádiusu frézy
FST	Zastavení posuvu
FUP	Schéma řídícího systému (metoda programování PLC)
GP	Základní program
GUD	Global User Data: Globální uživatelská data
HD	Pevný disk
HEX	Zkratka pro hexadecimální
HHU	Ruční ovladač
HMI	Rozhraní člověk-stroj

© Siemens AG 2004, Všechna práva vyhrazena

HiFu	Pomocná funkce
HMS	Měřicí systém s vysokým rozlišením
HW	Hardware
I	Vstup
I/O	Vstup/výstup
I/RF	Napájení/regenerativní zpětná vazba pro SIMODRIVE 611 (D)
IBN	Uvedení do provozu
IF	Uvolnění impulzů modulu pohonu
IK (GD)	Implicitní komunikace (globální data)
IKA	Interpolační kompenzace
IM	Modul rozhraní
IMR	Modul rozhraní pro příjem
IMS	Modul rozhraní pro odesílání
INC	Inkrement, velikost kroku
INI	Inicializace dat
IPO	Interpolátor
IS	Signál rozhraní
ISA	Mezinárodní standardní architektura
ISO	International Standard Organiyation
ISO Code	Zvláštní kód děrné pásky, počet děr na znak je vždy sudý
JOG	Režim JOG
KD	Otáčení souřadnice
KOP	Schéma kontaktů (metoda programování PLC)
K_0	Převodový poměr
K_V	Faktor zesílení regulační smyčky
LAD	Žebříkový diagram (metoda programování PLC)
LCD	Displej z kapalných krystalů
LEC	Korekce chyby vodicího šroubu
LED	Light Emitting Diode: světelná dioda
LF	Line Feed
LMS	Systém pro měření polohy

© Siemens AG 2004, Všechna práva vyhrazena

LR	Regulátor polohy
LUD	Lokální uživatelská data
MB	Megabyte
MC	Měřicí obvod
MCP	Řídící panel stroje
MCS	Souřadný systém stroje
MD	Parametry stroje
MDA	Manual Data Automatic (MDI)
MK	Měřicí smyčka
MKS	Souřadný systém stroje
MLFB	Strojově čitelné označení výrobků
MMC	Komunikace člověk-stroj: Uživatelské rozhraní na numerickém řídícím systému pro ovládání, programování a simulace. MMC a HMI jsou významově identické.
MPF	Main Program File: Hlavní program
MPI	Rozhraní s více porty
MS-	Microsoft
MSD	Pohon hlavního vřetena
MSTT	Řídící panel stroje
NC	Numerical Control: Numerické řízení
NCK	Numerical Control Kernel: Jádro řídícího systému Složka NC-řízení, která zpracovává programy a koordinuje ve velké míře pohyby stroje
NCU	Numerická řídící jednotka: Hardwarová jednotka NCK
NRK	Označení operačního systému pro NCK
NST	Signál rozhraní
NURBS	Neuniformní Racionální B-spliny
NV	Posunutí počátku
O	Výstup
OB	Organizační blok v PLC
OEM	Original Equipment Manufacturer: Výrobce zařízení, které je prodáváno jiným prodejce, obvykle pod jiným názvem
OI	Rozhraní pro obsluhu

© Siemens AG 2004, Všechna práva vyhrazena

OP	Panel operátora
OPI	Rozhraní panelu operátora
OPT	Volitelné doplňky
OSI	Vzájemné propojení otevřených systémů
P Bus	Sběrnice pro periferie
PC	Osobní počítač
PCIN	Název SW pro výměnu dat s řídícím systémem
PCMCIA	Mezinárodní asociace pro paměťové karty do osobních počítačů
PDB	Databáze produktů
PG	Programovací zařízení
PLC	Programovatelné logické řízení
POS	Polohování
PP	Plánování produktu
RAM	Random Access Memory: Paměť RAM
REF	Funkce Najíždění na referenční bod
REPOS	Funkce Návrat na původní pozici
RISC	Počítače s redukovaným instrukčním souborem: Typ procesoru s malým instrukčním souborem a velkým výkonem
ROV	Rapid Override
RPA	R-parametr aktivní: Paměťová oblast v NCK pro početní parametry
RPY	Roll Pitch Yaw: Typ rotace souřadného systému.
R-232	Sériové rozhraní: Definice signálových vodičů mezi DTE a DCE
RTS	Request To Send (sériové datové rozhraní)
SBL	Single Block: Blok po bloku
SD	Nastavovaný parametr
SDB	Systémový datový blok
SEA	Setting Data Active: Identifikace (typ souboru) pro nastavované parametry
SFB	Systémový funkční blok
SFC	Volání systémové funkce
SK	Softkey: Programové tlačítko

© Siemens AG 2004, Všechna práva vyhrazena

SKP	Skip: Přeskočení bloku
SM	Krokový motor
SOP	Shopfloor Oriented Programming: Dílensky orientované programování
SPF	Sub Program File: Podprogram
SR	Subrutina
SRAM	Statická paměť RAM (zálohovaná baterií)
SRK	Korekce rádiusu břitu
SSFK	Kompenzace chyby stoupání vřetena
SSI	Synchronní sériové rozhraní
STL	Seznam hlášení
SW	Software
SYF	Systémové soubory
T	Nástroj
TC	Výměna nástroje
TEA	Testing Data Active: Identifikátor pro strojní parametry
TLC	Korekce délky nástroje
TNRC	Korekce rádiusu špičky nástroje
TO	Korekce nástroje
TOA	Tool Offset Active: Identifikace (typ souboru) pro korekce nástroje
TRANSMIT	Transform Milling to Turning: konverze souřadného systému na soustruzích pro frézovací operace
TRC	Korekce rádiusu nástroje
UFR	Uživatelský frame: posunutí počátku
UP	Podprogram
VSA	Pohon posuvu
V.24	Sériové rozhraní: Definice signálových vodičů mezi DTE a DCE
WCS	Souřadná soustava obrobku
WKZ	Nástroj
WLK	Korekce délky nástroje
WO	Pracovní posunutí
WOA	Pracovní posunutí aktivní

© Siemens AG 2004, Všechna práva vyhrazena

WOP	Dílensky orientované programování
WDP	Adresář obrobků
WRK	Korekce rádiusu nástroje
WZK	Korekce nástroje
WZW	Výměna nástroje
ZO	Posunutí počátku (WO)
ZOA	Zero Offset Active (WOA): Identifikace (typ souboru) pro data posunutí počátku
µC	Mikrokontrolér (mikroprocesor)

© Siemens AG 2004, Všechna práva vyhrazena

B Pojmy

Tyto důležité pojmy jsou seřazeny podle abecedy. Na pojem, který se nachází v odstavci vysvětlení a pro který existuje samostatné heslo, odkazuje znak ->.

A

Absolutní rozměr Údaj cíle pohybu osy prostřednictvím rozměru, který je vztažen na počátek momentálně platného souřadného systému. Viz také -> Inkrementální rozměry.

Adresa Adresa je označení pro určitý operand nebo oblast operandů, např. vstup, výstup atd.

Adresa osy viz -> Identifikátor osy

Alarm Všechna -> hlášení a alarmy se na řídícím panelu vypisují srozumitelným textem spolu s údajem času a data a příslušným symbolem pro kritérium vymazání. Vypisování se uskutečňuje odděleně pro alarmy a hlášení.

1. Alarmy a hlášení ve výrobním programu
   Alarmy a hlášení se mohou přenášet z výrobního programu ve srozumitelné textové podobě přímo na displej.
2. Alarmy a hlášení z PLC
   Alarmy a hlášení stroje mohou být z programu PLC přenášeny na displej ve srozumitelné textové podobě. Pro tento účel nejsou zapotřebí žádné další funkční moduly.

Archivace Ukládání souborů a/nebo adresářů na **externí** paměťové zařízení.

A-spliny Akima spliny procházejí tangenciálně naprogramovanými interpolačními body (polynom 3. stupně).

Asynchronní podprogram Výrobní program, který může být spouštěn prostřednictvím signálu přerušení (např. signál „rychlý NC vstup“) asynchronně (nezávisle) na stavu programu.

Auto Provozní režim řídícího systému (sekvenční blokový podle DIN): Provozní režim NC systému, v němž se vybírá -> výrobní program, který je potom kontinuálně zpracováván.

B

Bezpečnostní funkce Řídící systém obsahuje neustále aktivní kontroly, které se snaží rozpoznat poruchy v -> CNC, v -> PLC a na stroji dostatečně včas, aby byly z větší části vyloučeny poškození obrobku, nástroje nebo stroje. V případě poruchy se operace obrábění přeruší a pohony se vypnou, příčina poruchy se uloží do paměti a aktivuje se alarm. Současně se sdělí do PLC, že se spustit alarm CNC.

© Siemens AG 2004, Všechna práva vyhrazena

Blok Součást -> výrobního programu, která je vymezena znakem LF. Rozlišují se -> hlavní bloky a -> vedlejší bloky.

Bootování Zavádění systémových programů po zapnutí systému.

B-spliny U B-splinů nepředstavují naprogramované pozice pevné uzlové body, ale pouze „kontrolní body“. Vytvořená křivka neprochází přímo těmito body, ale jen v jejich blízkosti (volitelné polynomy 1. 2. nebo 3. stupně).

C

Celkový reset Při mazání jsou v CPU vymazány následující paměti:
- pracovní paměť
- oblasti pro čtení a zápis
- systémová paměť
- zálohovaná paměť

CNC -> NC

COM Součásti NC řídícího systému pro provádění a koordinaci komunikace.

CPU Centrální procesorová jednotka, -> programovatelný řídící systém s pamětí.

C-spliny C-spline je nejznámějším a nejčastěji používaným splinem. Přechody na uzlových bodech jsou tangenciální a spojitě zakřivené. Používají se polynomy 3. stupně.

Cyklus Chráněný podprogram pro provádění opakovaně se vyskytujících soustružnických operací na -> obrobku.

Č

Časově reciproční posuv U systému SINUMERIK 840D může být namísto rychlosti posuvu pro pohyb osy naprogramován čas, za jaký se má úsek dráhy v bloku urazit (G93).

D

Datové slovo Dva byty dlouhá datová jednotka v rámci -> datového modulu.

Datový modul
1. Datová jednotka -> PLC, do které mají přístup programy -> HIGHSTEP.
2. Datová jednotka -> NC: Datové moduly obsahují definice dat pro globální uživatelská data. Data mohou být při své definici přímo inicializována.

Definice proměnné Definice proměnné zahrnuje stanovení datového typu a názvu proměnné. Pomocí názvu proměnné je přístup k hodnotě proměnné.

© Siemens AG 2004, Všechna práva vyhrazena

Délka kroku Údaje délek posuvu pomocí inkrementů. Délka tohoto inkrementu může být uložena do nastavovaného parametru a potom s ní lze pracovat pomocí odpovídajících tlačítek 10, 100, 1000 10000 atd.

Diagnose
1. Systémová oblast řídícího systému
2. Řídící systém obsahuje jak program pro interní diagnostiku, tak i testovací pomůcky pro servisní práce: stavové, alarmové a servisní výpisy.

Dráhová osa Dráhovými osami jsou všechny osy -> kanálu podílející se na obrábění, které jsou -> interpolátorem vedeny tak, že jsou současně spouštěny, zrychlovány a zastavovány, aby dosáhly koncového bodu.

DRF Differential Resolver Function: NC-funkce, která ve spojení s elektronickým ručním kolečkem vytváří v automatickém režimu inkrementální posunutí počátku.

Dynamická funkce předběžného zpracování Nepřesnosti kontury způsobované vlečnou chybou se dají téměř eliminovat dynamickou funkcí předběžného zpracování, která je závislá na zrychlení. Díky tomu se dosahuje i při vysokých rychlostech pohybu po dráze vynikající přesnosti opracování. Předběžné zpracování může být pro jednotlivé osy ve výrobním programu aktivováno a deaktivováno.

E

Editor Editor umožňuje sestavování, upravování, doplňování, vzájemné přesuny a vkládání programů / textů / programových bloků.

Elektronické ruční kolečko Pomocí elektronických ručních koleček je možné simultánně pohybovat vybranými osami v manuálním režimu. Vyhodnocování dílků stupnice ručního kolečka je definováno prostřednictvím stanovené délky kroku.

Externí posunutí počátku Posunutí počátku zadané z -> PLC.

F

Frame Frame představuje matematický předpis, který převádí souřadnice jednoho kartézského souřadného systému na souřadnice jiného kartézského systému. Frame obsahuje složky -> posunutí počátku, -> rotace, -> změna měřítka, -> zrcadlové převrácení.

G

Geometrická osa Geometrické osy se používají pro popis dvou a trojrozměrných oblastí v souřadném systému obrobku.

Geometrie Popis -> obrobku v -> souřadném systému obrobku.

© Siemens AG 2004, Všechna práva vyhrazena

Globální hlavní program/podprogram Každý globální hlavní program/podprogram se smí v adresáři pod svým názvem vyskytovat jen jednou. Stejný název programu v různých adresářích s různým obsahem je jako globální program nepřípustný.

H

HIGHSTEP Shrnutí programovacích možností pro -> PLC systému AS 300 / AS 400.

Hlášení Všechna hlášení naprogramovaná v programu pro výrobu součásti a systémem rozpoznané -> alarmy se vypisují na řídícím panelu stroje srozumitelným textem doplněným o udání data a času a o příslušný symbol pro kritérium vymazání. Vypisování se uskutečňuje odděleně pro alarmy a hlášení.

Hlavní blok Blok uvedený „:“ (dvojtečkou), který obsahuje pokyny potřebné pro uskutečnění pracovního postupu v rámci -> výrobního programu.

Hlavní program Číslem nebo identifikátorem označený -> výrobní program, ve kterém mohou být vyvolávány další hlavní programy, podprogramy nebo -> cykly.

Hodnota kompenzace Rozdíl mezi pozicí osy zjištěnou měřicí sondou a požadovanou naprogramovanou pozicí osy.

Chráněná oblast Trojrozměrný prostor v rámci pracovního prostoru, do kterého se špička nástroje nesmí dostat.

I

Identifikátor Slova jsou podle DIN 66025 jsou doplňována identifikátory (názvy) pro proměnné (početní, systémové, uživatelské proměnné), pro podprogramy, pro klíčová slova a slova s více adresovými písmeny. Tato doplnění nabývají významu při sestavování bloku. Identifikátory musí být jednoznačné. Jeden identifikátor se nesmí použít pro různé objekty.

Identifikátor osy Osy označují písmeny X, Y, Z podle DIN 66025 pravotočivý pravoúhlý -> souřadný systém.
Kruhové osy otáčející se okolo os X, Y, Z získávají identifikátory A, B, C. Pomocné osy mohou být paralelně k již uvedeným označeny dalšími adresovými písmeny.

Inicializační modul Inicializační moduly jsou speciální -> programové moduly, které obsahují přiřazení hodnot, jež je potřeba uskutečnit před zpracováváním programu.
Inicializační moduly slouží především pro inicializaci předem definovaných dat nebo globálních uživatelských dat.

© Siemens AG 2004, Všechna práva vyhrazena

Inicializační soubor Každému -> obrobku je možné založit jeden inicializační soubor. V něm mohou být uloženy různé příkazy pro hodnoty proměnných, které mají platit speciálně pro daný obrobek.

Instalace
- SINUMERIK FM-NC se zasouvá do řady CPU systému SIMATIC S7-300. Tento úplně uzavřený modul šířky 200 mm odpovídá vnější konstrukcí modulům systému SIMATIC S7-300
- SINUMERIK 840D se zapojuje jako kompaktní modul do systému měniče SIMODRIVE 611D. Rozměry odpovídají modulu SIMODRIVE 611D šířky 50 mm. Modul SINUMERIK 840D se skládá z modulu NCU a skříňky NCU.

Interpolace spliny Pomocí splinové interpolace je řídící systém schopen pouze na základě několika předem zadaných opěrných bodů vytvořit požadovanou konturu s hladkým křivkovým průběhem.

Interpolační kompenzace Pomocí interpolační kompenzace je možné kompenzovat výrobou podmíněné chyby stoupání vřetena a chyby měřicího systému (SSFK a MSFK).

Interpolátor Logická jednotka -> NCK, která po zadání cílové pozice ve výrobním programu stanoví pomocné hodnoty pro jednotlivé osy odpovídající pohybu, který je potřeba uskutečnit.

J

Jazyky Texty, které se vypisují na uživatelském rozhraní, a systémová hlášení a alarmy jsou k dispozici v pěti jazycích (disketa): **němčina, angličtina, francouzština, italština** a **španělština**. V řídícím systému mohou být implementovány a používány vždy dva z výše uvedených jazyků.

Jog Provozní režim řídícího systému (seřizovací režim): V provozním režimu Jog je možné provádět seřizování stroje. Jednotlivými osami a vřeteny je možné pohybovat směrovými tlačítky v tipovacím režimu. Dalšími funkcemi v provozním režimu Jog jsou -> najíždění na referenční bod, -> REPOS a -> PRESET (nastavení skutečné hodnoty).

K

K_0 Převodový poměr

Kanál Kanál se vyznačuje tím, že může zpracovávat -> program nezávisle na jiných kanálech. Kanál řídí výlučně osy a vřetena, která mu byla přiřazena. Programové postupy různých kanálů mohou být prostřednictvím -> synchronizace koordinovány.

Kanálová struktura Kanálová struktura umožňuje simultánně a asynchronně zpracovávat -> programy jednotlivých kanálů.

© Siemens AG 2004, Všechna práva vyhrazena

Klíč programátora Znak a posloupnost znaků, která má v programovacím jazyce pro -> výrobní programy definovaný význam (Viz Příručka programování).

Klíčová slova Slova s pevně definovaným způsobem zápisu, která mají v programovacím jazyku pro -> výrobní program definovaný význam.

Kompenzace chyby kvadrantu Chyby kontury na přechodech mezi kvadranty, které vznikají v důsledku měnících se podmínek tření a vodicích drahách, mohou být do značné míry odstraněny kompenzací chyby kvadrantu. Dosazení parametrů pro kompenzaci chyby kvadrantu se provádí pomocí zkoušky kruhového tvaru.

Kompenzace chyby stoupání vřetena Vyrovnávání mechanické nepřesnosti vřetena podílejícího se na posuvu prováděné řídícím systémem na základě změřených hodnot odchylek.

Kompenzace ztrát Vyrovnávání mechanických vůlí stroje, např. vůle vřetena na valivých ložiscích při změně směru. Pro každou osu se může kompenzace vůle zadávat odděleně.

Kompenzační osa Osa, jejíž požadovaná a skutečná hodnota byly modifikovány hodnotou kompenzace.

Kompenzační tabulka Tabulka uzlových bodů. Jsou zde uvedeny kompenzační hodnoty kompenzační osy pro zvolené pozice základní osy.

Konfigurace S7 Konfigurace S7 je nástroj, s jehož pomocí se nastavují parametry modulů. Pomocí S7 se vytvářejí různé -> bloky parametrů -> CPU a periferních modulů na -> PG. Tyto parametry se přenášejí do CPU.

Kontrola kontury Jako měřítko pro zachování kontury se sleduje, zda vlečná chyba leží v rámci definovaného tolerančního pásma. Nepřípustně vysoká vlečná chyba může mít např. za následek přetížení pohonu. V takovém případě se aktivuje alarm a osy se zastaví.

Kontura Obrys -> obrobku.

Kontura hotového dílu Kontura nahotovo obráběného obrobku. Viz také -> surový obrobek.

Kontura obrobku Požadovaná kontura vyráběného/obráběného -> obrobku.

Korekce nástroje Naprogramováním **T-čísla** (5 dekád, celá čísla) v bloku se uskuteční volba nástroje. Každému nástroji může být přiřazeno až 9 břitů (D-adres). Počet nástrojů používaných v řídícím systému se nastavuje při instalaci.

Korekce rádiusu břitu Při programování kontury se vychází z toho, že nástroj je špičatý. Jelikož toto v praxi není realizovatelné, zadává se do řídícího systému rádius zakřivení použitého nástroje, který se potom bere v úvahu. Při vedení nástroje podél kontury se střed zakřivení pohybuje ve stále stejné vzdálenosti rovnající se rádiusu zakřivení.

© Siemens AG 2004, Všechna práva vyhrazena

Korekce rádiusu nástroje Abyste mohli požadovanou -> konturu obrobku přímo naprogramovat, musí řídící systém pohybovat nástrojem po ekvidistantní dráze vzhledem ke kontuře, přičemž musí znát přesný rádius použitého nástroje (G41/G42).

Kostra Za kostru se považuje celek složený ze všech vzájemně spojených neaktivních dílů výrobního prostředku, kde se ani v případě poruchy nemůže vyskytnout nebezpečné dotykové napětí.

Kruhová interpolace -> Nástroj se má pohybovat po kruhové dráze mezi pevně zvolenými body kontury s uvedeným posuvem a přitom opracovávat obrobek.

Kruhová osa Kruhové osy zabezpečují otáčení obrobku nebo nástroje do předem definované úhlové polohy.

K_V Zesílení smyčky, regulační charakteristika regulačního obvodu.

L

Lineární osa Lineární osa je osa, která oproti kruhové ose opisuje přímku.

Look ahead Pomocí funkce **Look Ahead** řídící systém vyhodnocuje několik bloků dopředu (tento počet lze nastavit pomocí parametru), čímž se dosahuje optimální rychlosti při zpracování.

M

Machine Systémová oblast řídícího systému

Makra Shrnutí určitého množství příkazů do jednoho identifikátoru. Tento identifikátor potom v programu reprezentuje sadu soustředěných příkazů.

MDA Provozní režim řídícího systému: Manual Data Automatic.
V provozním režimu MDA mohou být jednotlivé bloky programu nebo jejich posloupnosti zadávány bez vztahu na hlavní program nebo podprogram a potom mohou být stisknutím tlačítka NC-Start ihned uskutečňovány.

Měřicí jednotky palce nebo metrické V programu pro obrábění můžete pozice a hodnoty stoupání programovat v palcích. Nezávisle na programovatelných měřicích jednotkách (G70/G71) je řídící systém převede na základní systém.

Měřicí obvod

- SINUMERIK FM-NC: Potřebné měřicí obvody pro osy a vřetena jsou v modulu řídícího systému standardně integrovány. Celkem může být realizováno maximálně 4 osy a vřetena, z nichž 2 mohou být vřetena.
- SINUMERIK 840D: Vyhodnocování měřicích převodníků se nachází v modulech pohonů SIMODRIVE 611D. Maximální konfigurace činí celkem 8 os a vřeten, z čehož může být až 5 vřeten.

© Siemens AG 2004, Všechna práva vyhrazena

Měřicí systém využívající palce Měřicí systém, který vzdálenosti udává v „palcích“ a jejich zlomcích.

Metrický měřicí systém Normovaný měřicí systém jednotek: pro délky např. mm (milimetr), m (metr).

Mez přesného najetí Jestliže všechny dráhové osy dosáhnou své meze přesného najetí, chová se řídící systém tak, jako by bylo přesně dosaženo cílového bodu. Uskuteční se přechod na další blok výrobního programu.

Mezní otáčky Maximální/minimální otáčky (vřetena): Maximální otáčky vřetena mohou být omezeny zadáním hodnoty do strojního parametru, -> PLC nebo -> nastavovaného parametru.

Modul Jako moduly jsou označovány všechny soubory, které jsou zapotřebí pro sestavování a zpracovávání programů.

Modul analogových vstupů/výstupů Moduly analogových vstupů/výstupů jsou zdroji signálů pro analogové procesní signály.
Moduly analogových vstupů převádějí analogové měřicí veličiny na digitální hodnoty, které pak mohou být zpracovány v CPU.
Moduly analogových výstupů převádějí digitální hodnoty na analogové řídící signály.

Modul digitálních vstupů/výstupů Digitální moduly slouží pro zpracovávání binárních procesních signálů.

Modul periferií Moduly periferií zabezpečují spojení mezi CPU a procesem.
Periferními moduly jsou:
- -> Moduly digitálních vstupů/výstupů
- -> Moduly analogových vstupů/výstupů
- -> Moduly simulátoru

Modul simulátoru Modul simulátoru je modul, který může
- prostřednictvím ovládacích prvků simulovat digitální vstupní veličiny a
- zobrazovat digitální výstupní veličiny

N

Najíždění na pevný bod Obráběcí stroje mohou najíždět na definované pevné body, jako jsou bod pro výměnu nástroje, bod pro upnutí obrobku, bod výměny palety atd. Souřadnice těchto bodů jsou uloženy v řídícím systému. Pokud je to možné, řídící systém najíždí na tyto body příslušnými osami -> rychlým posuvem.

Najíždění na pevný bod stroje Pohyby vedoucí k dosažení předem definovaného -> pevného bodu stroje.

© Siemens AG 2004, Všechna práva vyhrazena

Najíždění na referenční bod Jestliže se v používaném systému pro měření dráhy nevyskytuje žádný snímač absolutní polohy, je zapotřebí najíždění na referenční bod, aby bylo zajištěno, že měřicím systémem předávaný údaj skutečné polohy je v souladu s hodnotami na souřadnicích systému stroje.

Nastavované parametry Parametry, které definovaným způsobem systémovým programovým vybavením zprostředkovávají řídícímu systému NC vlastnosti obráběcího stroje.

Nástroj Nástroj je softwarový nástroj pro zadávání a úpravy -> parametrů v bloku parametrů. Nástroji jsou mimo jiné:

- -> Konfigurace S7
- S7-TOP
- S7-Info

Nástroj Pracovní součást na obráběcím stroji, která způsobuje obrábění, např. soustružnický nůž, vrták, laserový paprsek …

Název osy viz -> Identifikátor osy

NC Numerické řízení: Řídící systém zahrnující všechny komponenty pro ovládání obráběcího stroje: -> NCK, ->PLC, ->MMC/HMI, -> COM.

NCK Kernel řídícího systému: Součást řídícího systému, která zpracovává -> výrobní program a v zásadě koordinuje pohybové operace obráběcího stroje.

Nekonečné otáčení kruhové osy V závislosti na použití může být rozsah pohybu kruhové osy nastaven na 360 stupňů a méně nebo se může otáčet libovolným směrem pořád dokola. Kruhové osy s nekonečným otáčením se používají například pro zaoblování, broušení a úhlové polohování.

NRK Numerický robotický kernel (operační systém -> NCK).

NURBS Interní v řídícím systému prováděné vedení pohybů a dráhové interpolace se provádějí na bázi NURBS (neuniformní racionální B-spliny). Díky tomu je v řídícím systému k dispozici jednotné chování pro všechny interpolace (SINUMERIK 840D).

O

Obrábění šikmých ploch Vrtání a frézování na plochách obrobku, které neleží v souřadných rovinách stroje, se mohou pohodlně uskutečňovat s podporou funkce „obrábění šikmých ploch“.

Obrobek Součást, která má být vyráběna nebo opracovávána obráběcím strojem.

© Siemens AG 2004, Všechna práva vyhrazena

OEM Pro výrobce stroje, který si přeje v řídícím systému instalovat své vlastní uživatelské rozhraní nebo specifické technologické funkce, existuje prostor pro individuální řešení (aplikace OEM) pro SINUMERIK 840D.

Ohraničení pracovní oblasti Pomocí ohraničení pracovní oblasti je možné omezit rozsah pohybu os. Pro každou osu může být přiřazena dvojice hodnot popisujících chráněný pracovní prostor.

Orientované zastavení vřetena Zastavení vřetena obrobku v předem definované úhlové poloze, např. aby bylo možné uskutečnit další obrábění na určitém místě

Orientovaný návrat nástroje RETTOOL: Při přerušení obrábění (např. při zlomení nástroje) je možné nástroj pomocí programového příkazu stáhnout zpět s předem definovanou orientací.

Osa C Osa okolo které se provádí řízený otočný pohyb a polohování s vřetenem obrobku.

Osy Osy CNC jsou rozděleny podle své funkce následujícím způsobem:

- Osy: Interpolační dráhové osy
- Pomocné osy:Přísuvné a polohovací osy neschopné interpolace se specifickým osovým posuvem Pomocné osy nejsou zapotřebí pro vlastní obrábění, např. podavač nástrojů, zásobník nástrojů.

Osy stroje Osy fyzicky existující na obráběcím stroji.

Override Manuální, příp. programovatelná možnost zásahu, která obsluhujícímu pracovníkovi umožňuje změnit naprogramované posuvy nebo otáčky, aby je bylo možné přizpůsobit určitému obrobku nebo materiálu.

Override posuvu Naprogramovaná rychlost je nahrazena aktuálním nastavením rychlosti uskutečněným pomocí řídícího panelu stroje nebo na PLC (0-200 %). Rychlost posuvu může být dodatečně měněna v programu pro opracování součásti prostřednictvím programovatelného procentuálního faktoru (1 – 200 %).

P

Paměť korekcí Datová oblast řídícího systému, ve které jsou uloženy hodnoty korekcí nástroje.

Paměťové programovatelné řízení Paměťové programovatelné řízení (SPS) jsou elektronické řídící systémy, jejichž funkce je uložena ve formě programu v paměťovém zařízení. Konstrukce a zapojení zařízení tedy nezávisí na funkci řídícího systému. Paměťové programovatelné řídící systémy mají konstrukci počítače: skládají se z CPU (centrální modul) s pamětí, modulů vstupů/výstupů a interního sběrnicového systému. Periferie a programovací jazyk jsou podřízeny potřebám řízení.

© Siemens AG 2004, Všechna práva vyhrazena

Parametr

1. **S7 –300**: Rozlišujeme dva druhy parametrů:
   - Parametr příkazu STEP 7
     Parametr příkazu STEP 7 je adresa operandu nebo konstanty, které je zapotřebí zpracovat.
   - Parametr -> programového bloku
     Parametr programového bloku určuje chování modulu.
2. **840D**:
   - Systémová oblast řídícího systému
   - Početní parametr, může mu být programátorem výrobního programu dosazena libovolná hodnota pro libovolné účely nebo tato hodnota může být použita.

Pevný bod stroje Obráběcím strojem jednoznačně definovaný bod, např. referenční bod.

PG Programovací přístroj

PLC Programmable Logic Control: -> Paměťový programovatelný řídící systém. Součást -> řídícího systému NC: Přizpůsobení řídícího systému pro řídící logiku obráběcího stroje.

Počátek souřadné soustavy obrobku Počátek souřadné soustavy obrobku tvoří výchozí bod této soustavy. Je definován vzdáleností od počátku souřadné soustavy stroje.

Počátek souřadné soustavy stroje Pevný bod obráběcího stroje, ke kterému jsou vztaženy všechny odvozené měřicí systémy.

Podpora cyklů V systémové oblasti „Program“, v menu „Cycle support“ se nachází seznam cyklů, které jsou Vám k dispozici. Po aktivování požadovaného obráběcího cyklu se srozumitelným textem vypíší potřebné parametry, jimž je potřeba přiřadit odpovídající hodnoty.

Podprogram Posloupnost pokynů -> výrobního programu, která může být opakovaně vyvolávána s různými vstupními parametry. Volání podprogramu se uskutečňuje z hlavního programu. Každý podprogram může být zablokován proti neoprávněnému čtení a vypisování. Jednou z forem podprogramu jsou -> cykly.

Pohon

- SINUMERIK FM-NC nabízí analogové rozhraní ± 10 V pro systém měniče SIMODRIVE 611A.
- Řídící systém SINUMERIK 840D je spojen se systémem měniče SIMODRIVE 611D prostřednictvím rychlé digitální paralelní sběrnice.

Polární souřadnice Souřadný systém, ve kterém je poloha bodu v rovině dána vzdáleností od počátku a úhlem, který svírá vektor rádiusu s definovanou osou.

Polohovací osa Osa, která provádí pomocné pohyby na obráběcím stroji (např. zásobník nástrojů, přeprava palet). Polohovací osy jsou osy, které nejsou interpolovány spolu s -> dráhovými osami.

© Siemens AG 2004, Všechna práva vyhrazena

Polynomická interpolace
Pomocí polynomické interpolace mohou být konstruovány křivky rozmanitých průběhů, jako jsou **přímka, parabola, mocninná funce** atd. (SINUMETIK 840D).

Pomocné bloky
Pracovní posuvy s aktivovanou korekcí nástroje (G41/G42) smí být přerušeny omezeným počtem pomocných bloků (bloků bez pohybu os v rovině korekce), přičemž korekce nástroje se ještě dá správně vypočítat. Přípustný počet pomocných bloků, které je řídící systém schopen dopředu načíst, je nastavitelný pomocí systémového parametru.

Pomocné funkce
Prostřednictvím pomocných funkcí mohou být ve -> výrobních programech předávány -> parametry do -> PLC, které tam potom spouští výrobcem stroje definovanou reakci.

Posunutí počátku
Udání nového vztažného bodu pro souřadný systém, které je vztaženo na již existující počátek a frame.

1. Nastavitelné
   SINUMERIK FM-NC: Pro každou CNC osu mohou být zvolena čtyři nezávislá posunutí počátku.
   SINUMERIK 840D: K dispozici je určitý předem daný počet nastavitelných posunutí počátku pro každou CNC osu.
   Alternativně lze používat posunutí aktivovaná pomocí G-funkcí.
2. Externí
   Kromě všech posunutí, jež určují polohu počátku souřadné soustavy obrobku, mohou být superponována ještě posunutí počátku pocházející:
   - od ručního kolečka (posunutí DRF)
   - z PLC
3. Programovatelná
   Pomocí příkazu TRANS lze naprogramovat posunutí pro všechny dráhové a polohovací osy.

Posuv po dráze
Posuv po dráze se vztahuje na -> dráhové osy. Představuje geometrický součet posuvů -. geometrických os, které se na něm podílejí.

Power On
Vypnutí a opětovné zapnutí řídícího systému.

Pracovní paměť
Pracovní paměť je paměť RAM v -> CPU, do které má přístup procesor v průběhu zpracovávání uživatelského programu.

Pracovní plocha
Pracovní plocha je zobrazovací médium řídícího systému představované obrazovkou. Nachází se zde osm vertikálních a osm horizontálních programových tlačítek.

Pracovní prostor
Trojrozměrný prostor, ve kterém se může pohybovat špička nástroje na základě konstrukce obráběcího stroje.
Viz také -> Chráněný prostor

© Siemens AG 2004, Všechna práva vyhrazena

Preset Pomocí funkce Preset může být nově definován počátek soustavy řídícího systému v souřadném systému stroje. Při funkci Preset se neprovádí žádný pohyb os, momentálním polohám os se pouze přiřadí nová hodnota.

Profilová kolejnice Profilová kolejnice slouží pro upevnění modulů S7-300.

Program
1. Systémová oblast řídícího systému
2. Posloupnost pokynů pro řídící systém.

Program PCIN pro přenos dat PCIN je pomocný program pro odesílání a příjem CNC uživatelských dat prostřednictvím sériového rozhraní, jako jsou např. výrobní programy, korekce nástroje atd. Program PCIN se mže spouštět na standardních průmyslových PC pod MS-DOSem.

Programová paměť PLC SINUMERIK 840D: V uživatelské paměti PLC jsou společně uloženy uživatelský program PLC a uživatelská data a základní program PLC. Uživatelská paměť PLC může být rozšířena až na 96 kBytů.

Programovací jazyk CNC Programovací jazyk CNC nabízí: -> uživatelské proměnné, -> předem definované uživatelské proměnné, -> systémové proměnné, -> nepřímé programování, -> matematické a úhlové funkce, -> logické operátory a logická spojení, -> programové skoky a větvení, -> koordinaci programů (SINUMERIK 840D), -> makra.

Programovací jazyk CNC Základem programovacího jazyka CNC je DIN 66025 s rozšířením. Programovací jazyk CNC a programování umožňují mimo jiné definici maker (shrnutí jednotlivých příkazů).

Programování PLC PLC se programuje pomocí softwaru **STEP 7**. Programovací software STEP 7 je založen na standardním operačním systému **Windows** a obsahuje funkce systému STEP 5 s nově vyvinutými rutinami.

Programovatelné framy Pomocí programovatelných -> framů je možné dynamicky v průběhu zpracovávání výrobního programu definovat nové počátky souřadného systému. Je třeba rozlišovat mezi absolutní definicí na základě nového framu a aditivní definicí vycházející z již existujícího počátečního bodu.

Programovatelné ohraničení pracovního prostoru Ohraničení prostoru pro pohyby nástroje na prostor vymezený programovými mezemi.

Programové tlačítko Tlačítko, jehož popis je reprezentován políčkem na obrazovce. Toto tlačítko se dynamicky přizpůsobuje aktuální situaci obsluhy systému. Volně obsaditelným funkčním tlačítkům jsou programovým vybavením přiřazovány definované funkce.

Programový modul Programové moduly obsahují hlavní programy a podprogramy -> výrobního programu.

© Siemens AG 2004, Všechna práva vyhrazena

Provozní režim Způsob provozu řídícího systému SINUMERIK. Existují provozní režimy-> Jog, -> MDA a ->Auto.

Předvídání narušení kontury Řídící systém rozpoznává a hlásí následující kolizní případy:
1. Úsek dráhy je kratší než rádius nástroje.
2. Šířka vnitřního rohu je menší než průměr nástroje.

Přenosová rychlost Rychlost při přenosu dat (bitů/s).

Přepínač na klíč
1. **S7-300**: Přepínač na klíč zabezpečuje volbu provozního režimu -> CPU. Tento přepínač je ovládán vyjímatelným klíčem.
2. **840D**: Přepínač na klíč na -> ovládacím panelu stroje má 4 polohy, které jsou obsazeny funkcemi operačního systému řídícího systému. K přepínači na klíč dále patří tři barevně odlišené klíče, které je možné vytáhnout v příslušných polohách.

Přesné najetí Při programovatelném příkazu přesného najetí se na pozici uvedenou v bloku najíždí pomalu a přesně. Pro zkrácení doby přibližování jsou pro rychlý a pracovní posuv definovány -> meze přesného najetí.

Přímková interpolace Nástroj se pohybuje po přímce do cílového bodu a přitom opracovává obrobek.

Přístupová práva Programové moduly CNC a data jsou chráněny sedmistupňovým systémem přístupových práv:
- Tři úrovně hesla pro výrobce systému, výrobce stroje a uživatele
- Čtyři polohy přepínače na klíč, které jsou vyhodnocovány PLC

R

Referenční bod Bod obráběcího stroje, na který je vztažen měřicí systém -> os stroje.

Remanence Remanentní jsou datové oblasti v datových modulech, jako jsou časy, číslice a ukazatele, pokud se jejich obsah neztrácí při vypnutí a opětovném zapnutí systému.

REPOS
1. Opětovné najíždění na konturu obsluhou
   Pomocí funkce REPOS můžete pomocí směrových tlačítek najet zpět na místo na kontuře, kde došlo k přerušení.
2. Opětovné najíždění na konturu programem
   Pomocí programových příkazů si můžete vybrat z několika strategií najíždění: Najíždění na místo, kde došlo k přerušení, najíždění na počáteční bod bloku, najíždění na koncový bod bloku, najíždění na bod na dráze mezi počátkem bloku a místem přerušení.

Rotace Součást -> framu, která definuje otočení souřadného systému o určitý úhel.

© Siemens AG 2004, Všechna práva vyhrazena

Rozsah posuvu Maximální přípustný rozsah pohybu u lineárních os je ± 9 dekád. Absolutní hodnota závisí na zvolené jemnosti zadávané hodnoty a polohové regulace a na systému jednotek (palce nebo metrický systém).

R-parametr Početní parametr, který může být programátorem -> výrobního programu v programu použit pro různé účely a s nímž lze pracovat.

Rutina přerušení Rutiny přerušení jsou speciální -. podprogramy, které se mohou spouštět v důsledku určité události (externí signál) z technologického procesu. Právě zpracovávaný výrobní program se přeruší a pozice os, na které k přerušení došlo, se automaticky uloží.

Rychlé digitální vstupy/výstupy Pomocí digitálních vstupů se mohou spouštět např. rychlé programové CNC rutiny (rutiny přerušení). Pomocí digitálních CNC výstupů se mohou spouštět rychlé programem řízené spínací funkce (SINUMERIK 840D).

Rychlé pozvednutí od kontury Vyskytne-li se přerušení, může být pomocí programu CNC spouštěn pohyb, který umožňuje rychlé pozvednutí nástroje od právě obráběné kontury obrobku. Kromě toho lze v parametrech nastavit úhel zpětného pohybu a délku této dráhy. Po rychlém pozvednutí se může spouštět navíc i rutina přerušení (SINUMERIK 840D).

Rychlost pohybu po dráze Cílem řízení pohybu po dráze je zabránit velkému brždění -> dráhových os, kterým se opotřebovávají řídící systém, stroj a ostatní hmotné statky provozovatele a uživatele, na hranicích bloků výrobního programu a na další blok přecházet pokud možno se stejnou rychlostí pohybu po dráze.

Rychlý posuv Nejvyšší rychlost posuvu osy. Používá se např. tehdy, jestliže nástroj najíždí z klidové poloh na -> konturu obrobku nebo je stahován zpět od kontury obrobku.

Ř

Řetězové kótování Také inkrementální rozměr: Stanovení cíle pohybu osy udané dráhou a směrem, které je potřeba urazit, vztažené na již dosažený bod. Viz také -> absolutní rozměr.

Řídící panel stroje Řídící panel obráběcího stroje s ovládacími prvky, jako jsou tlačítka, otočné přepínače atd. a signalizačními prvky, jako jsou světelné diody. Slouží k bezprostřednímu ovlivňování obráběcího stroje pomocí PLC.

Řízení rychlosti Aby při pracovních posuvech o velmi krátké vzdálenosti na blok bylo možné dosáhnout přijatelné rychlosti pohybu, je možné aktivovat vyhodnocování průběhu rychlosti na několik bloků dopředu (-> Look Ahead).

© Siemens AG 2004, Všechna práva vyhrazena

S

Sběrnice S7-300 Sběrnice S7-300 je sériovou datovou sběrnicí, pomocí které spolu komunikují jednotlivé moduly a přes kterou jsou napájeny potřebným napětím. Spojení mezi moduly je zabezpečováno -> sběrnicovou spojkou.

Sběrnicová spojka Sběrnicová spojka je součást příslušenství k S7-300 a dodává se spolu s modulem periferií. Sběrnicová spojka rozšiřuje -> sběrnici S7-300 z -> CPU, příp. modulu periferií pro sousední moduly periferií.

Sériové rozhraní V.24 Je k dispozici pro vstup a výstup dat na:

- MMC modul MMC 100 je k dispozici jedno rozhraní V.24 (RS-232)
- MMC modul MMC 101 a MMC 102 jsou k dispozici dvě tato rozhraní.

Pomocí těchto rozhraní je možné načítat a ukládat výrobní programy, jakož i data výrobce a uživatele.

Services Systémová oblast řídícího systému

Síť Síť je spojení několika systémů S7 – 300 a dalších koncových zařízení, např. PG pomocí -> spojovacího kabelu. Prostřednictvím sítě se uskutečňuje výměna dat mezi připojenými zařízeními.

Skupina provozních režimů V daném okamžiku jsou všechny osy/vřetena přiřazeny právě jednomu kanálu. Každému kanálu je přiřazena jedna skupina provozních režimů. Kanálům BAG je vždy přiřazen jeden -> provozní režim.

Softwarový koncový spínač Softwarový koncový spínač omezuje rozsah pohybu osy a zabraňuje najíždění saní na hardwarový koncový spínač. Pro každou osu je možné zadat 2 páry hodnot, které pak mohou být odděleně aktivovány pomocí PLC.

Souřadný systém Viz -> souřadný systém stroje, -> souřadný systém obrobku.

Souřadný systém obrobku Souřadný systém obrobku je svým počátkem vztažen na obrobek. Při programování v souřadném systému obrobku jsou rozměry a směry vztaženy na tento systém.

Souřadný systém stroje Souřadný systém stroje (MCS) je ten, který se vztahuje na osy obráběcího stroje.

Spirální interpolace Spirální interpolace se hodí obzvláště pro jednoduchou výrobu vnějších a vnitřních závitů s tvarovými frézami a pro frézování mazacích drážek. Spirála se přitom skládá ze dvou pohybů:
1. Kruhový pohyb v rovině
2. Lineární pohyb kolmo na tuto rovinu

© Siemens AG 2004, Všechna práva vyhrazena

Spojovací kabel Spojovací kabely jsou dodávaná nebo uživatelem vyrobená dvoudrátová spojení s konektory na obou koncích. Tyto spojovací kabely propojují -> CPU pomocí -> vícebodového rozhraní (MPI) s -> PG nebo s jinou CPU.

Správa výrobních programů Správa výrobních programů může být organizována podle -> obrobků. Počet programů a dat, která lze spravovat, je dána velikostí uživatelské paměti. Každý soubor (program a data) může být opatřen názvem skládajícím se z maximálně 24 znaků.

SPS Viz -> Paměťové programovatelné řízení

Standardní cykly Pro často se opakující obráběcí operace jsou k dispozici standardní cykly:

- pro technologie vrtání/frézování
- pro technologie soustružení

V systémové oblasti „Program“ pod menu „Cycle support“ se nachází seznam cyklů, které jsou Vám k dispozici. Po aktivování požadovaného obráběcího cyklu se srozumitelným textem vypíší potřebné parametry, jimž je potřeba přiřadit odpovídající hodnoty.

Surový obrobek Díl, kterým se opracovávání obrobku začíná.

Synchronizace Příkazy na určitých místech ve -> výrobním programu pro koordinaci operací v různých -> kanálech.

Synchronní akce

1. Výstup pomocných funkcí
   Při opracovávání obrobku se mohou předávat z CNC programu do PLC technologické funkce (-> pomocné funkce). Pomocí těchto pomocných funkcí jsou např. řízeny pomocná zařízení obráběcího stroje, jako jsou pinola, podavač, upínací sklíčidlo atd.
2. Výstup rychlých pomocných funkcí
   Pro časově kritické spínací funkce mohou být minimalizovány potvrzovací časy (-> pomocné funkce). Zbytečné body pozastavení jsou z obráběcího procesu odstraněny.

Synchronní osy Synchronní osy potřebují pro provedení svého pohybu stejný čas, jaký potřebuje geometrická osa pro svůj pohyb po dráze.

Systémová paměť Systémová paměť je paměť v NCU, do které se ukládají následující data:

- Data, která potřebuje řídící systém
- Operandy času, počítadel a ukazatelů

Systémová proměnná Proměnná, která existuje bez přičinění programátora -> výrobního programu. Je definována svým datovým typem a názvem, který začíná znakem $. Viz také -> Uživatelská proměnná.

© Siemens AG 2004, Všechna práva vyhrazena

T

Teach In Pomocí funkce **Teach In** je možné sestavovat a opravovat výrobní programy. Jednotlivé programové bloky jsou zadávány pomocí klávesnice a ihned prováděny. Mohou se ukládat také směrová tlačítka nebo ruční kolečka pro najíždění na pozice. Doplňkové údaje, jako jsou G-funkce, posuvy nebo M-funkce, se mohou zadávat do stejného bloku.

Textový editor -> Editor

Transformace Programování v kartézském souřadném systému, zpracování v nekartézském souřadném systému (např. s osami stroje, jež jsou kruhovými osami).

U

Uživatelem definované proměnné Uživatel může definovat uživatelské proměnné pro libovolné použití ve -> výrobním programu nebo datovém modulu (globální uživatelská data). Definice obsahuje údaj datového typu a název proměnné. Viz také -> Systémová proměnná.

Uživatelská paměť Všechny programy a data, jako jsou výrobní programy, podprogramy, komentáře, korekce nástroje, posunutí počátku/framy, jakož i data kanálů a uživatelů programů se ukládají do společné uživatelské paměti CNC.

Uživatelský program Uživatelské programy pro automatizační systémy S7-300 jsou sestavovány pomocí programovacího jazyka STEP 7. Uživatelská program má modulární strukturu a skládá se z těchto modulů:
Kódový modul: Tento modul obsahuje příkazy jazyka STEP 7
Datový modul: Tento modul obsahuje konstanty a proměnné pro program v jazyce STEP 7.

Vedlejší blok Blok začínající „N“ a obsahující informace pro krok pracovního postupu, např. udání polohy.

Vícebodové rozhraní Vícebodové rozhraní (MPI) je 9-pólové D-Sub-rozhraní. Na toto rozhraní je možné připojit v parametrech definovaný počet přístrojů, které pak vzájemně komunikují:

- PG
- Ovládací a monitorovací systémy
- Další automatizační systémy

Blok parametrů „Multipoint Interface MPI“ CPU obsahuje -> parametry, které definují vlastnosti tohoto rozhraní.

Vrtání závitu bez vyrovnávací hlavičky Pomocí této funkce je možné vrtat závity bez vyrovnávací hlavičky. Díky interpolačnímu chování vřetena, které je řízeno jako kruhová osa a osa vrtání, jsou závity odříznuty přesně na koncové vrtané hloubce, např. závity ve slepých dírách (předpoklad: osový režim vřetena).

© Siemens AG 2004, Všechna práva vyhrazena

Vřetena Funkce vřetena jsou na dvou úrovních:

1. Vřetena: Pohony vřetena s digitální regulací otáček a polohy (SINUMERIK 840D)
2. Pomocná vřetena: Pohony vřeten s regulací otáček, soubor funkcí „Pomocné vřeteno“, např. pro poháněné nástroje.

Vyhledávání bloku Při testování výrobního programu nebo po přerušení jeho zpracování je možné pomocí této funkce vyhledat libovolné místo ve výrobním programu, od kterého se má zpracování znovu spustit nebo odkud má pokračovat.

Výrobní program Posloupnost příkazů pro výrobní program, který zabezpečí opracování určitého -> obrobku. Také uskutečnění určitého opracování na zadaném -> surovém obrobku.

Vyrovnání driftu Během fáze konstantního posuvu provádějí CNC osy automatické vyrovnávání driftu analogové regulace otáček (SINUMERIK FM-NC).

Z

Základní osa Osa, jejíž požadovaná a skutečná hodnota se používají pro výpočet hodnot korekcí.

Základní souřadný systém Kartézský souřadný systém, který se prostřednictvím transformace zobrazuje na souřadný systém stroje.
Ve -> výrobním programu programátor používá názvy os základního souřadného systému. Pokud není aktivní žádná -> transformace, splývá se souřadným systémem stroje. Rozdíl mezi nimi spočívá v identifikátorech os.

Zálohování Stažení obsahu paměti na externí paměťové zařízení.

Záložní baterie Záložní baterie zaručuje, že -> uživatelský program v -> CPU je chráněn proti výpadku napájení a že definované datové oblasti a ukazatele, časy a čísla zůstanou nezměněny.

Záložní paměť Záložní paměť zaručuje zálohování paměťových oblastí -> CPU bez záložní baterie. Zálohovány jsou pomocí parametrů nastavitelné časy, čísla, ukazatele a datové byty.

Změna měřítka Komponent -> framu, který způsobuje změnu měřítka pro určitou osu.

Zpracovatelský kanál Prostřednictvím kanálové struktury je možné, aby souběžně probíhaly pohybové operace, např. posuv zakládacího portálu zároveň s obráběním, čímž se zkracují vedlejší časy. Na CNC kanál je potřeba nahlížet jako na samostatný řídící CNC systém s dekódováním, přípravou bloků a interpolací.

© Siemens AG 2004, Všechna práva vyhrazena

Zrcadlové převrácení Při zrcadlovém převrácení jsou znaménka hodnot souřadnic osy vztahující se k dané kontuře vyměněny. Současně je možné zrcadlově převrátit i několik os.

Zrychlení s omezením trhavých pohybů Aby se dosáhlo optimální charakteristiky zrychlení na stroji a aby se současně šetřila jeho mechanika, je možné ve výrobním programu přepínat mezi skokovým průběhem zrychlení a spojitým zrychlením (bez trhnutí).

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky

© Siemens AG 2004, Všechna práva vyhrazena

C Rejstřík

$
$AA_COUP_ACT 9-372, 9-391, 13-528
$AA_COUP_OFFS 13-528
$AA_LEAD_SP 9-391
$AA_LEAD_SV 9-391
$AA_MOTEND 5-233
$AA_SCPAR 5-235
$MC_COMPESS_VELO_TOL 9-397
$SA_LEAD_TYPE 9-390, 9-391
$TC_CARR1...14 8-354
$TC_CARR18[m] 8-354
$TC_CARR24[m] 8-356
3
3D-obvodové frézování s omezujícími plochami 8-339

A

AC-regulace, aditivní 10-450
AC-regulace, multiplikativní 10-451
Adresáře obrobků 3-143
Adresy OEM 5-230
Aktivování lisování 12-502
Aktivování lisování se zpožděním 12-502
Aktivování prostřihování 12-502
Aktivování tangenciálního řízení, TANGON 9-365
Aktuální 1. základní frame v kanálu 6-262
Aktuální celkový frame 6-265
Aktuální nastavitelný frame 6-263
Aktuální programovatelný frame 6-264
Aktuální systémový frame 6-261, 6-264
Aktuální systémový frame 6-264
Aktuální základní frame globální v NCU 6-261
Aktuální
  stav vazby vlečného vřetena 13-528
  úhlové posunutí 13-528
ASUP 10-479
Asynchronní oscilace 11-482
Automatické "GET" 1-82
Automatické rozdělení dráhy 12-506
Automatický ukazatel přerušení 9-409

B

Blok parametrů servomechanismu programovatelný 5-235
Blok pozastavení 9-408
Boční úhel 7-275
Body obratu oscilačního pohybu 11-483

C

CALCDAT 14-568
CANCEL 10-480
Celkový základní frame 6-262, 6-263
CONTDCON 14-568
CONTPRON 14-568
cov.com, uživatelské cykly 2-131
CTAB 9-386
CTABDEF 9-377
CTABDEL 9-379
CTABEND 9-377
CTABINV 9-386
CTABSEV 9-384, 9-385, 9-386
CTABSSV 9-384, 9-386
Cykly
  dosazování parametrů uživatelským cyklům 2-128, 2-130

Č

Časová náročnost
  synchronní akce 10-471
Čelní frézování 7-277
Číslo břitu 8-349
Čtení 13-518

D

D-čísla
  kontrola 8-350
  přejmenování 8-351
  volné zadávání 8-349
  zjišťování T-čísel 8-352
Deaktivování dílčího řetězce 1-58
Deaktivování framu 6-253
Deaktivování lisování se zpožděním 12-502
Deaktivování lisování, prostřihování 12-502

© Siemens AG 2004, Všechna práva vyhrazena

Deaktivování tangenciálního řízení 9-365
Deaktivování transformace: TRAFOOF 7-309
Deaktivování/opětovné aktivování rutiny
přerušení 1-74
Definice pole 1-31
Definice pole, seznam hodnot 1-33
Definice polynomu 10-446
Definice proměnných 1-26
Definice uživatelských dat 3-150
DELETE 1-87
Délka řetězce 1-57
Délka součásti 11-489
Dílčí přísuv 11-490
Dílčí úsek 12-506
Dílčí úseky 12-506
Doba pozastavení 11-485
Doba prodlevy 1-71
Doba zpracování programu 13-555
Dráhové osy 5-213
DRF-posunutí 6-250
Druh vazby 13-520
Držák nástroje 8-356
kinematika 8-354
mazání/změny/načítání dat 8-357

E

EG
elektronická převodovka 13-530
EGONSYNE 13-533
Elektronická převodovka 13-530
Eulerův úhel8-344
EXECTAB 14-568
EXECUTE 4-174, 14-569, 14-576
EXTCALL 2-125
Externí posunutí počátku 6-251

F

F osa 9-388
FGROUP
- osy 5-213
FOR 1-64
FOsa 9-362, 9-369, 9-376
Fréza
- pomocný bod (FH) 8-337
- špička (FS) 8-337
F-slovo: polynom 5-214
Funkce OEM 5-230

G

G643 5-214
G-kód 5-213
- skupina 5-215
GUD
automatické aktivování 3-157

H

Hloubka vnoření 1-65
Hrubé posunutí 6-249
CHECKSUM 1-93
Chování na okraji tabulky křivky 9-383
Chráněná oblast
aktivování/deaktivování chráněné oblasti 4-178
definice chráněné oblasti 4-175
chráněné oblasti specifické pro určitý kanál 4-174
chráněné oblasti specifické pro stroj 4-174
popis kontury chráněné oblasti 4-176

I

Identifikační číslo 10-422
Index pole 1-32
Inicializační program 3-147
načtení inicializačního programu 3-148
uložení inicializačního programu 3-148
uživatelská data 3-150
vytvoření inicializačního programu 3-148
Interpolace orientace 7-283
Interpolace vektoru otočení 7-279
Interpolační takt 13-550
INTERSEC 14-568
ISD (hloubka vnoření) 8-333
ISFILE 1-91
Jemné posunutí 6-249

K

Kanálové framy 6-259
Klíčové slovo 10-423
Koeficienty polynomu 5-208
Kompenzace chyby kvadrantu
aktivování operace učení 13-519
deaktivování operace učení 13-519
dodatečné učení 13-519
Kompresor 5-199, 5-213

© Siemens AG 2004, Všechna práva vyhrazena

Kompresor pro orientaci
COMPON, COMPCURV 5-201
Komunikace NCU-NCU 13-550
Koncový úhel 7-279
Konec programu 1-71, 10-479
Konfigurovatelné oblasti parametrů 3-154
Kontrolní struktury 1-63
Konturové prvky, průsečík 14-580
Konturový prvek 14-572, 14-574
Konverze typů 1-53
Koordinace os 10-460
Koordinace programu 1-67
příkazy pro koordinaci programů 1-68
příklad 1-70
Korekce (override) 11-496
aktuální 10-470
výsledná 10-471
Korekce nástroje
3D-čelní frézování 8-335
čelní frézování 8-333
korekce na dráze, zakřivení dráhy a hloubka zajíždění 8-337
on-line 8-325
paměť korekcí 8-320
výměna nástroje 8-336
Korekce rádiusu nástroje s normovanými nástroji
dráha středu nástroje 8-340
kontura na obráběné ploše 8-343
Korekce rádiusu nástroje
3D-obvodové frézování bez omezujících ploch 8-339
3D-obvodové frézování s CUT3DC 8-340
3D-obvodové frézování s normovanými nástroji 8-340
3D-obvodové frézování s omezujícími plochami 8-340
zpoždění v rozích 5-230
Korekce rádiusu nástroje, 3D 8-333
hloubka zajíždění ISD 8-337
vnitřní rohy/vnější rohy 8-337
Kritérium ukončení pohybu
programovatelné 5-232
Kruhová interpolace 5-215
Kruhové osy
směrové vektory V1, V2 8-354
vektory vzdálenosti I1, I2 8-354
Křivkové parametry 5-213
KS 9-362

L

L osa 9-388
Lineární interpolace 5-213, 5-215
Linková komunikace 13-550
Linková osa 13-553
Linková proměnná
globální 13-550
Linkový modul 13-550
Lisování 12-502, 12-506
Logické operátory 1-48
LOsa 9-362, 9-369, 9-376

M

M6
příslušný podprogram 2-129
volání podprogramu 2-129
MAC
automatická aktivace 3-157
Makra 2-134, 12-505
Matematické funkce 1-43
Matematické operace/funkce 1-43
MEAFRAME 6-254, 6-258
MEASA 5-220
Měření 10-466
Měření se spínací sondou
programování měřicích bloků 5-218
stavové proměnné 5-218
M-funkce
třímístné 2-135
minimální/maximální pozice kruhové osy 8-356
MODE 14-569
Monitorování nástroje, specifické pro broušení 8-327
M-příkazy 12-505

N

n 9-376
Najíždění na kódované pozice 5-190
Najíždění na nejbližší bod na dráze 9-412
Najíždění na pevný doraz a FOCON/FOCOF 10-467
Nastavení proměnného chování při výměně osy 1-84

© Siemens AG 2004, Všechna práva vyhrazena

Nastavitelný referenční bod dráhy 5-213
Nastavované parametry 11-484
Návrat do původní polohy 10-480
NC-Stop 10-478
NCU zapojené do sítě 13-550
NCU-
linkový modul 13-550
NCU-globální nastavitelný frame: 6-259
NCU-globální základní frame: 6-258
Nekonečný program 1-66
Neperiodická tabulka křivky 9-383
Nepřímé programování 1-37
Nepřímé programování G-kódu 1-39
Nepřímé vyvolání podprogramu 1-38
Nepřímé vyvolání programu sestaveného v jazyce ISO pomocí ISOCALL 2-114
NEWCONF 1-85
Nulový frame 6-253

O

Objíždění konturového prvku 14-582
Obrat
- bod 11-490
- oblast 11-490
Obrobky: adresář 3-143
Obvodové frézování 8-333
Obvodové frézování s omezujícími plochami 8-333
Oddělování třísky 14-568
Offset kruhové osy 8-356
Offset normály kontury OFFN 7-298
Okrajové podmínky 1-66, 5-215, 10-477
Okrajové podmínky při transformacích 7-307
On-line korekce délky nástroje 7-290
On-line korekce nástroje 10-454
Opakované použití tabulky křivky 9-379
Opakování programu 2-110
Opakování programu s nepřímým programováním CALL 2-113
Operace s řetězci 1-52
Opětovné najíždění na konturu 9-410
bod zpětného najetí 9-411
najíždění novým nástrojem 9-412
najíždění po přímkách 9-413
najíždění po půlkruhu 9-414
najíždění v čtvrtkruhu 9-413
Optimalizace tangenciálního sledování 9-362
ORIC 8-344
Orientace nástroje 7-274, 8-344
s příkazy LEAD a TILT 7-278
Orientační osy 7-274, 7-281, 7-283
Osa
lokální 13-553
Osa/vřeteno pro upnutí 13-553
Oscilace
aktivování/deaktivování oscilace 11-485
asynchronní oscilace 11-482, 11-484
definice pohybové operace 11-486
řízení pomocí synchronní akce 11-490
synchronní oscilace 11-489
Oscilační osa 11-483
Osy
zásobník 13-553
Otáčení vektoru orientace
programování pomocí THETA 7-279
Override 11-496
aktuální 10-470
výsledná 10-471

P

Paměť předběžného zpracování 9-399
Paměť
pracovní paměť 3-140
programová paměť 3-140
struktura paměti 3-140
Parametry pro kruhové osy 8-356
Parametry pro návrat z podprogramu 2-106, 2-107
Parametry synchronních akcí 10-433
Periodické tabulky křivky 9-383
Pevně stanovený počet obrobků 1-67
Početní parametr 1-24
Počítadlo obrobků 13-557
Podélné soustružení
vnější obrábění 14-569
vnitřní obrábění 14-569
podprogram s předáváním parametrů
předávání parametrů mezi hlavním programem a podprogramem 2-99
Podprogram, externí 2-125

© Siemens AG 2004, Všechna práva vyhrazena

Podprogramy 2-96
mechanismus SAVE 2-97
modální volání podprogramu 2-110
nepřímé volání podprogramu 2-112
opakování programu 2-110
podprogram s předáváním parametrů 2-103
vnoření 2-97
volání podprogramu 2-103
Podprogramy s předáváním parametrů
definice pole 2-102
Podříznutí 14-569
Pohyb samostatné osy 12-508
Pohybové synchronní akce
akce 10-426
programování 10-419
přehled 10-427
Pohyby vřetena 10-462
Polohová synchronizace 13-521
Polohovací pohyby 10-456
Polohování os 10-458
Polynom jmenovatele 5-212
Polynom
- interpolace 5-213
Polynomická interpolace 5-207
Polynom jmenovatele 5-212
Polynomy maximálně 3. stupně 9-377
Polynomy maximálně 5. stupně 9-377
Pomocné funkce 10-441, 12-506
Porovnávací a logické operátory 1-45
Porovnávací a logické operátory
priorita operátorů 1-50
Posunutí počátku
deaktivování transformace 6-253
externí posunutí počátku 6-251
posunutí ručním kolečkem 6-250
PRESETON 6-252
Posuv osy 10-459
Posuv po dráze 5-214
Posuv
osy 10-459
Potlačení režimu blok po bloku 2-119
Power On 10-477
Pracovní paměť 3-147
Datové oblasti 3-147
Inicializační program 3-147
Vyhrazené názvy modulů 3-150
Priorita
operace 1-50
Programová paměť 3-140
adresář obrobků 3-143
adresáře 3-142
cesta pro vyhledávání při volání podprogramu 3-145, 3-146
přehled 3-141
typy souborů 3-142
volba obrobku 3-145
založení adresáře obrobků 3-144
Programování cesty pro vyhledávání při volání podprogramu 3-146
Programování orientace 7-283
Programování orientace nástroje 8-344
chování na vnějších rozích 8-345
obvodové frézování 8-335
orientace nástroje 8-344
Programování šikmé osy G05, G07 7-306
Programovatelná cesta pro vyhledávání při volání podprogramu 2-116
Programovatelný ukazatel přerušení 9-407
Proměnná 1-24
definice pole 1-31
definovaná uživatelem 1-24
druhy proměnných 1-24
globální proměnná specifická pro NCK 1-71
nepřímé programování 1-37
přiřazení 1-42
početní proměnná 1-25
převádění typů 1-51
systémová proměnná 1-25
typy proměnných 1-25
uživatelem definovaná proměnná 1-26
Proměnná časovače 10-432
Proměnná FIFO 10-436
Proměnná typu FRAME 6-238
Definice nového framu 6-248
Načítání nebo změna komponent framu 6-244
Předdefinované proměnné typu FRAME 6-239
Přiřazení hodnot 6-242
Vyvolání transformace soustavy souřadnic 6-238
Proměnná v reálném čase 10-429

© Siemens AG 2004, Všechna práva vyhrazena

Proměnná značky 10-432
Prostřihování 12-502
Provozní režim 5-222
Průsečík konturových prvků 14-568
Prvky podříznutí 14-570
První základní frame v kanálu 6-260
Přebírání změřených hodnot 5-219
Předdefinované posunutí 6-252
Předem definované názvy proměnných
GUD 3-154
Přenosový poměr 13-524
Přepisování tabulky křivky 9-380
Příčné obrábění
vnější obrábění 14-569
vnitřní obrábění 14-569
Příkaz CASE 1-61
Příkaz skoku
příkaz CASE 1-61
Příkaz strukturování pro Step-editor 3-170
Příkazové osy 10-456
Příkazové prvky 10-421
Příkazy pro tabulky křivky 9-373
Přímé převzetí osy: GETD 1-82
Příprava kontury 14-569, 14-576
Prvky podříznutí 14-570
Přiřazení 1-42
Přiřazení a spuštění rutiny přerušení 1-74
Přístup na pozice a segmenty v tabulce
křivky 9-386
Přísuv
- osa 11-498
- pohyb 11-493, 11-495
Přísuv
potlačení 11-490

R

READ 1-89
Reakce prováděné NC 13-540
Reakce závislé na pohonu 13-539
Regulace vzdálenosti 10-451
REPEAT 1-65
Reset 10-478
Režim 11-489
Režim generátoru 13-546
Rozdělení dráhy 12-508
Rozdělení dráhy u dráhových os 12-507
Rozdělení dráhy u samostatných os 12-508
Rozšířené měřicí funkce 5-220, 7-286
Rozšířené zastavování a stahování 13-538
R-parametr 10-434
Rutina přerušení 1-72
programovatelný směr posuvu 1-72
rychlé pozvednutí od kontury 1-75
stanovení posloupnosti 1-74
uložení pozice přerušení 1-73
Řetězce framů 6-246
Řídící osa 9-388
Řízení výkonu laseru 10-448

S

SBLON 2-119
Sestavení rutiny přerušení jako
podprogramu 1-73
SCHNITT 14-581
Simulace řídící hodnoty 9-391
Singulární místa 7-282
Skládání řetězců 1-55
Softwarový koncový spínač 10-459
Splinová interpolace 5-191, 5-213
A-Spline 5-192
B-Spline 5-193
C-Spline 5-194
kompresor 5-197
Splinový svazek 5-197
Spojení os řídící hodnotou 9-388
Spojení řídící hodnotou 10-464
Spojení skutečnou a požadovanou
hodnotou 9-389
Spojení skutečnou hodnotou 13-520
Spouštěcí události 5-222
Spouštění zdvihu 12-504
Správa nástrojů 8-322
Správy strojového času 1-65
Spuštění/zastavení osy 10-458
Stahování 13-540
Stahování závislé na pohonu 13-548
Stanovení otočení framu 6-248
Stanovení skutečné hodnoty 10-461
Stav měřicí sondy 5-224
Stav stroje, globální 13-550
Stav vazby 9-391
Strojní a nastavované parametry 10-435

© Siemens AG 2004, Všechna práva vyhrazena

Stupeň 9-376
Svazek os s vlečením 9-370
Synchronní akce 13-550
Synchronní akce
statické 9-392
Synchronní chod
hrubý 13-520
jemný 13-520
synchronní chod podle požadované hodnoty 13-520
Synchronní oscilace
definice přísuvu 11-491
přiřazení oscilačních a přísuvných os 11-491
přísuv v oblasti obratu 11-493
synchronní akce 11-492
zastavení v bodě obratu 11-495
Synchronní vřeteno 13-520
aktivování synchronního chodu 13-526
deaktivování synchronního chodu 13-526
druh vazby 13-525
- definice dvojice 13-522
- dvojice 13-521
chování při přechodu na další blok 13-525
přenosový poměr $K_Ü$ 13-524
systémové proměnné 1-25
vymazání vazby 13-527
Systémová proměnná 1-24, 13-550, 15-614
globální 13-550

Š

Šikmá osa, TRAANG 7-284, 7-303

T

Tabulka kontury 14-569, 14-576
Tabulky křivek 9-373
Takt IPO 11-496
TANG 9-363
Tangenciální řízení
definice vlečné a řídící osy 9-363
mezní úhel v důsledku omezení pracovního pole 9-365
Technologické cykly 10-473
Torze 13-518
TRAANG 7-302
TRACYL 7-296
TRAFOOF 7-270
Transformace křivek na ploše pláště válce 7-296, 7-300
Transformace křivek na ploše pláště válce Offset normály kontury OFFN 7-298
Transformace s otočnou lineární osou 7-271
Transformace šikmé osy 7-302
Transformace TRACYL 7-296
Transformace TRANSMIT 7-293
Transformace TRAORI 7-272
Transformace, 3/4-osová 7-272
Transformace, 5-osová
orientace nástroje 7-274
programování Eulerových úhlů 7-275
programování směrových vektorů 7-276
programování úhlů RPY 7-276
Transformace, 5-osová, čelní frézování 7-277
Transformace, 5-osová, programování pomocí příkazů LEAD/TILT 7-274
Transformace,
zřetězené 7-310
TRANSMIT 7-293
TRAORI 7-270
Trvalá vazba Master/Slave 13-564
Třímístné M-/G-funkce 2-135
Typ kinematiky 8-357
Typ kinematiky M 8-354
Typ kinematiky P 8-354
Typ kinematiky T 8-354
Typ proměnných 1-29
Typy nástrojů
tvary fréz, data nástroje 8-336

U

uc.com, uživatelské cykly 2-132
Učení se kompenzačních charakteristik 13-518
Úhel otočení 7-279
Úhel pootočení $\alpha 1$, $\alpha 2$ 8-354
Úhel RPY 8-344
Úhel tečen drah 10-470
Úhlový offset/úhlový inkrement kruhové osy 8-356
Úhlový vztah 13-526
Ukazatel max./min. 14-572, 14-574
Upnutí obrobku 13-550
Určitá místa v programu pro SERUPRO
zablokování 9-407

© Siemens AG 2004, Všechna práva vyhrazena

Úrovně ochrany
popis systémových proměnných a provádění prvků NC-jazyka 3-160
pro uživatelská data 3-155
změna atributů jazyka 3-163
změna pomocí strojních a nastavovaných parametrů 3-159
Úseky programu s podmíněným přerušením 9-401
Uvolněná kinematika 8-354

V

Vazba 9-362, 9-369
Vazba požadovanou hodnotou 13-520
Vedení pohybu 13-560
Velká/malá písmena 1-56
Vlečení 9-369, 10-463
faktor vazby 9-371
vlečená osa 9-370
Vlečná osa 9-388
Volání podprogramu pomocí M-/T-funkcí 2-128
Volání podprogramu se zadáním cesty a s parametry 2-115
Volání podprogramu
nepřímé 1-38
Volání podprogramu, cesta pro vyhledávání 3-145
Vybírání jednotlivých znaků 1-59
Vyhledávání bloku 10-479
Vyhledávání znaků 1-57
Vyhodnocovací funkce 10-449
Vyhodnocování 3D obvodového frézování na vnějších rozích
najíždění na průsečík 8-338
Vyhodnocování vytížení 10-471
Vyjiskřovací zdvih 11-488
Vymazání synchronní akce 10-477
Vymazání vazby 13-527
Vymazání zbytkové dráhy 5-223, 10-444, 11-488
Vymazání zbytkové dráhy s přípravou 10-444
Výměna os
GET 1-79
RELEASE 1-79
převzetí osy 1-81
uvolnění osy 1-80
Výměna vřetena
GET 1-80
RELEASE 1-80
Vypínací pozice 13-526
Výpis aktuálního bloku 2-118
Vypisování bloku 2-114, 2-118
Výpočet dat kruhového oblouku 14-583
Výpočet framu 6-254
Výpočetní výkon 13-550
Výrobní program 13-550, 13-553
Výsledek měření 5-223
Vyvolávání framu 6-245

W

WHEN-DO 11-492
WHILE 1-64
WKS 3-143
WPD 3-143
WRITE 1-85

Z

Zablokování načítání 10-442
Zadávání cesty
absolutní 1-68
relativní 1-68
Základní frame aktivního kanálu 6-261
Zálohování pomocného obvodu 13-546
Zaměnitelné geometrické osy 7-313
Zásobník os 13-553, 13-554
Zastavení 13-544
Zastavení a stahování
rozšířené 13-538
Zastavení předběžného zpracování 10-443
Zastavení závislé na pohonu 13-547
Závitové bloky 5-215
Zjišťování a využívání vyhledatelných oblastí 9-408
Změna provozního režimu 10-477
Změna stanice/polohy 13-553
Značky pro čekání 10-466
Zobrazení naposled naprogramovaného čísla bloku 2-114
Zpětné chybové hlášení 14-569, 14-576
Zpoždění v rohu na vnitřních rozích 5-231
Zpoždění v rohu na všech rozích 5-231
Zpracování externího podprogramu 2-125

© Siemens AG 2004, Všechna práva vyhrazena

Zpracování programu s pamětí předběžného
zpracování 9-399
Zpracování řetězce jako části výrobního
programu 1-41
Zrychlení závislé na dráze PUNCHACC 12-502,
12-503
Zřetězení framů 6-266
Ztráty 13-518

© Siemens AG 2004, Všechna práva vyhrazena

Pro poznámky:

© Siemens AG 2004, Všechna práva vyhrazena

D Příkazy, identifikátory

- 1-43
*
* 1-43
/
/ 1-43
:
: 1-43
+
+ 1-43
<
< 1-45
<= 1-45
<> 1-45
=
== 1-45
>
> 1-45
>= 1-45

A

A 7-302
A1, A2 8-354, 8-356
A2 7-275
A3 7-275
A4 7-275
A5 7-275
ABS 1-43
ACC 13-523
ACOS 1-43
ACTFRAME 6-240
ADISPOSA 5-232
ALF 1-72
A_{max} 12-502
A_{min} 12-502
AND 1-48
ANZ 14-583
ANZHINT 14-571, 14-573
applim 9-376
APR 3-155, 3-159, 3-161
aproxLW 9-376
APW 3-155, 3-159, 3-161
APX 3-161
AROTS 6-248
AS 2-134
ASIN 1-43
ASPLINE 5-191
ATAN2 1-43
AV 13-520
AV 13-525
AX 13-514
AXCTSWE 13-553
AXIS 1-28
AXNAME 1-54, 13-514
AXSTRING 1-54, 13-514

B

B_AND 1-49
B_NOT 1-49
B_OR 1-49
B_XOR 1-49
B2 7-275
B3 7-275
B4 7-275
B5 7-275
BAUTO 5-195
BEARBART 14-569
BFRAME 6-239
BLOCK 2-113
BNAT 5-195
BOOL 1-28
BRISK 11-483
BSPLINE 5-191
BTAN 5-195

C

C2 7-275
C3 7-275
C4 7-275
C5 7-275
CAC 5-190
CACN 5-190
CACP 5-190
CALCDAT 14-583
CALL 2-111, 2-112, 2-113
CALLPATH 2-116, 3-146
CANCEL 10-420

© Siemens AG 2004, Všechna práva vyhrazena

CASE 1-61
CDC 5-190
CFINE 6-249
CIC 5-190
CLEARM 1-69
CLRINT 1-72
CMIRROR 1-43, 6-242
COARSE 13-520, 13-524, 13-525
COARSEA 5-232
COMCAD 5-199
COMPCURV 5-199
COMPLETE 3-147, 3-148
COMPOF 5-199, 5-213
COMPON 5-199, 5-213, 9-397
CONTDCON 14-576
CONTPRON 14-569, 14-581, 14-582
COS 1-43
COUPDEF 13-520, 13-522, 13-524
COUPDEL 13-520, 13-522
COUPOF 13-520, 13-526, 13-527
COUPOFS 13-527
COUPON 13-520, 13-526, 13-527
COUPRES 13-520, 13-528
CP 7-286
CPROT 4-178
CPROTDEF 4-174, 4-176
CROT 1-43, 6-242
CROTS 6-248
CSCALE 1-43, 6-242
CSPLINE 5-191
CTAB 9-375
CTABDEF 9-373
CTABDEL 9-373
CTABEND 9-373
CTABEXIST 9-374
CTABFNO 9-373
CTABFPOL 9-374
CTABID 9-373, 9-374
CTABINV 9-375
CTABISLOCK 9-374
CTABLOCK 9-373
CTABMEMTYP 9-374
CTABMPOL 9-374
CTABNOMEM 9-373
CTABPERIOD 9-374
CTABPOLID 9-374
CTABSEG 9-374
CTABSEGID 9-374
CTABTEP 9-375
CTABTEV 9-375
CTABTMAX 9-375
CTABTMIN 9-375
CTABTSP 9-375
CTABTSV 9-375
CTABUNLOCK 9-373
CTRANS 1-43, 6-242
CUT3DC 8-333
CUT3DCC 8-333
CUT3DCCD 8-333
CUT3DF 8-333
CUT3DFF 8-333
CUT3DFS 8-333
CUTCONOF 8-330
CUTCONON 8-330

D

D 7-296
DEF 1-29, 3-151
DEFAULT 1-61
DEFINE 2-134
DELDTG 5-228
DELETE 1-88
DELT 8-322
DISABLE 1-72
DISPLOF 2-118
DISPR 9-410
DIV 1-43
DO 10-420, 11-489
DRFOF 6-253
DUPLO_NR 8-322
DV 13-520
DV 13-525
DZERO 8-353

E

EAUTO 5-195
ELSE 1-63
ENABLE 1-72
ENAT 5-195
ENDFOR 1-63
ENDIF 1-63
ENDLOOP 1-63

© Siemens AG 2004, Všechna práva vyhrazena

ENDPOS 11-489
ENDPROC 10-452
ENDWHILE 1-63
ERG 14-583
ETAN 5-195
EVERY 10-420
EXECSTRING 1-41
EXECTAB 14-582
EXECUTE 4-176, 14-569, 14-576
EXP 1-43
EXTCALL 2-125
EXTERN 2-103

F

FA 11-486, 13-523
FALSE 1-25
FCTDEF 8-325, 10-446
FCUB 9-394
FEHLER 14-569, 14-576
FENDNORM 5-231
FINE 13-520, 13-525
FINEA 5-232
FLIN 9-394
FMA 15-596
FNORM 9-394
FOR 1-63
FPO 9-394
FRAME 1-28
FRC 15-597
FRCM 15-597
FROM 10-419
FS 13-520
FTOCOF 8-325
FTOCON 8-325
FW 9-376

G

G[<Gruppen-Index>] 1-39
G05 7-306
G07 7-306
G1 11-483
G153 6-253
G25,G26 9-365
G4 11-485
G53 6-253
G62 5-231
G621 5-231
GEOAX 7-313
GET 1-79
GETACTTD 8-352
GETD 1-79
GETDNO 8-351
GETSELT 8-322
GETT 8-322
GEWINDE 14-570, 14-577
GOTO 1-61
GOTOB 1-61
GOTOC 1-61
GOTOF 1-61
GUD 3-142, 3-147, 3-150, 3-152, 3-155
CHANDATA 3-149
CHAR 1-28
CHKDNO 8-350

I

I1,I2 8-354
ID 10-419
IDS 10-419
IF 1-63
IF-ELSE-ENDIF 1-63
IFRAME 6-239
II1,II2 11-490
INDEX 1-57
INIT 1-68
INITIAL 3-148
INT 1-28
Integer/Real: proměnná 1-39
INTERSEC 14-581
IPOBRKA 5-232
IPOENDA 5-232
IPOSTOP 13-520, 13-523, 13-525
IPTRLOCK 9-407
IPTRUNLOCK 9-407
ISAXIS 13-514
ISD 8-333, 8-337
ISFILE 1-91
ISNUMBER 1-54
ISOCALL 2-114
ISVAR ( ) 13-516

J

JERKLIM 13-560

© Siemens AG 2004, Všechna práva vyhrazena

K

KTAB 14-571, 14-573, 14-579, 14-582

L

LEAD 7-275, 8-344
LEADOF 9-388
LEADON 9-388
LIFTFAST 1-72
LLIMIT 10-446
LN 1-43
LOCK 10-420
LOOP 1-63
LOOP-ENDLOOP 1-64
LS 13-520
LW 9-376

M

M 8-356
M17 2-98
MATCH 1-57
MCALL 2-110
MEAC 5-220, 5-228
MEAFRAME 6-255
MEAS 5-217
MEAW 5-217
MEAWA 5-220
MI 6-244
MINDEX 1-57
MIRROR 6-239
MMC 13-559
MOD 1-43
Modus 11-489
MOV 10-458
MPF 3-142
MU 7-303
MZ 7-303

N

NEWT 8-322
NN 14-569
NOC 13-520, 13-525
NOT 1-48
NPROT 4-178
NPROTDEF 4-174, 4-176
NUMBER 1-54

O

OEMIPO1/2 5-230
OF 1-62
OFFN 7-295, 7-296
OR 1-48
ORIAXES 7-283
ORICONCCW 7-283
ORICONCW 7-283
ORICONIO 7-283
ORICONTO 7-283
ORICURVE 7-283
ORID 8-344
ORIEULER 7-283
ORIMKS 7-281, 8-344
ORIPLANE 7-283
ORIROTA 7-279
ORIROTR 7-279
ORIROTT 7-279
ORIRPY 7-283
ORIS 8-344
ORIVECT 7-283
ORIVIRT1 7-283
ORIVIRT2 7-283
ORIWKS 7-281, 8-344
OS 11-482, 11-485
OSC 8-344
OSCILL 11-489, 11-491
OSCTRL 11-482, 11-486
OSE 11-482, 11-486
OSNSC 11-482, 11-489
OSOF 8-344
OSP 11-483
OSP1 11-482, 11-489
OSP2 11-482, 11-489
OSS 8-344
OSSE 8-344
OST 11-485
OST1 11-482, 11-489
OST2 11-482, 11-489
OVRA 13-523

P

PAROT 6-248
PCALL 2-115
PDELAYOF 12-502
PDELAYON 12-502

© Siemens AG 2004, Všechna práva vyhrazena

PFRAME 6-239
PKT 14-583
PL 5-194, 5-210
PO 5-210
POLF 13-542
POLFA 13-542
POLFMASK 13-543
POLFMLIN 13-543
POLY 5-210
POLYNOM 14-570, 14-577
POLYPATH 5-210
PON 12-502, 12-508
PONS 12-502
POS 13-526
POS_{FS} POS_{LS} 13-520
POSP 11-489
POT 1-43
PRESETON 6-252
PRIO 1-72
PROC 2-98
PS_{FS} 13-520
PTP 7-286
PUNCHACC 12-502
PUTFTOC 8-325
PUTFTOCF 8-325
PW 5-193

Q

QECDAT.MPF 13-519
QECLRN.SPF 13-519
QECLRNOF 13-518
QECLRNON 13-518
QECTEST.MPF 13-519
QFK 13-518

R

R
R10 9-376
RDISABLE 10-442
READ 1-89
REAL 1-28
REDEF 3-159
RELEASE 1-79
REP 1-35
REPEAT 1-63
REPOS 1-72, 1-79
REPOSA 9-410
REPOSH 9-410
REPOSHA 9-410
REPOSL 1-79, 9-410
REPOSQ 9-410
REPOSQA 9-410
RET 2-98
RET(<číslo bloku/návěští>, < >, < >) 2-107
RINDEX 1-57
RMB 9-410
RME 9-410
RMI 9-410
ROTS 6-248
ROUND 1-43
RPY 8-344
RT 6-244

S

S1,S2 13-522, 13-528
SAVE 1-73, 2-97
SBLOF 2-119
SBLON 2-119
SC 6-244
SCPARA 5-235
SD 5-193
SEFORM 3-170
SET 1-33
SETDNO 8-351
SETINT 1-72
SETM 1-69
SETPIECE 8-322
SIN 1-43
S_{max} 12-502
S_{min} 12-502
SOFT 11-483
SON 12-502, 12-507, 12-508
SONS 12-502
SPATH 5-213
SPI 13-514, 13-523
SPIF1 15-609
SPIF2 15-609
SPLINE 14-570, 14-577
SPLINEPATH 5-197
SPN 12-506
SPOF 12-502
SPOS 13-523

© Siemens AG 2004, Všechna práva vyhrazena

SPP 12-506
SQRT 1-43
SR 15-609
SRA 15-610
ST 15-610
STA 15-610
START 1-68
STARTFIFO 9-399
STAT 7-286
STOPFIFO 9-399
STOPRE 5-217, 5-224, 5-226, 9-399, 11-484
STOPREOF 10-443
STRING 1-28
STRINGFELD 1-52
STRINGVAR 1-52
STRLEN 1-57
SUBSTR 1-58
SUPA 6-253
SYNFCT 10-449

T

TABNAME 14-569, 14-576, 14-580, 14-582
TAN 1-43
TANG 9-362
TANGDEL 9-362
TANGOF 9-362
TANGON 9-362
TE 5-220
THETA 7-279
TILT 7-275, 8-344
TLIFT 9-362
TOFFOF 7-290
TOFFON 7-290
TOFRAME 6-248
TOLOWER 1-56
TOROT 6-248
TOUPPER 1-56
TR 6-244
TRAANG 7-302
TRACYL 7-296
TRAFOOF 7-293, 7-296, 7-302, 7-309
TRAILOF 9-369
TRAILON 9-369
TRANSMIT 7-293
TRAORI 7-272
TRUE 1-25
TRUNC 1-43, 1-46
TU 7-286

U

U1,U2 11-490
ULIMIT 10-446
UNLOCK 10-420
UNTIL 1-63, 1-65
UPATH 5-213

V

V1,V2 8-354
VAR 2-101
VARIB 14-580, 14-583
VELOLIM 13-561

W

WAIT 1-69
WAITC 13-520, 13-523
WAITE 1-69
WAITM 1-68
WAITMC 1-69
WAITP 11-485
WALIMON 9-365
WHEN 10-419
WHEN-DO 11-489
WHENEVER 10-419
WHENEVER-DO 11-489, 11-492
WHILE 1-63
WKS 11-496
WRITE 1-85
WZ 8-322

X

X 8-322
XOR 1-4

© Siemens AG 2004, Všechna práva vyhrazena