

SINUMERIK

SINUMERIK 840D sl / 828D Pro pokročilé

Programovací příručka

Platí pro:

Řídicí systém
SINUMERIK 840D sl / 840DE sl / 828D

Software CNC verze 4.8 SP3

08/2018
6FC5398-2BP40-6UA2

Předmluva

Základní bezpečnostní upozornění	1
Flexibilní programování NC systémů	2
Správa souborů a programů	3
Chráněné oblasti	4
Speciální příkazy dráhy	5
Transformace souřadného systému (Frame)	6
Transformace	7
Kinematické řetězce	8
Zabránění vzniku kolizí u kinematických řetězců	9
Transformace pomocí kinematických řetězců	10
Korekční parametry nástroje	11
Chování při pohybu po dráze	12
Spojení os vazbou	13
Synchronní akce	14
Pohyb tam a zpět	15
Lisování a prostřihování	16
Broušení	17
Další funkce	18
Vlastní programy pro oddělování třísky	19
Programování externích cyklů	20
Tabulky	21
Přílohy	A

Právní upozornění

Koncept výstražných upozornění

Tato příručka obsahuje pokyny, které musíte dodržovat z důvodu své osobní bezpečnosti a zamezení materiálními škodami. Upozornění ohledně Vaší osobní bezpečnosti jsou zvýrazněny výstražným trojúhelníkem, upozornění týkající se pouze materiálních škod jsou uvedeny bez výstražného trojúhelníku. Podle stupně ohrožení jsou výstražná upozornění zobrazena v sestupném pořadí následujícím způsobem.

NEBEZPEČÍ

znamená, že **nastane** smrt nebo těžké ublížení na zdraví, když se neučiní příslušná bezpečnostní opatření.

VÝSTRAHA

znamená, že **může** nastat smrt nebo těžké ublížení na zdraví, když se neučiní příslušná bezpečnostní opatření.

POZOR

znamená, že **může** nastat lehké ublížení na zdraví, když se neučiní příslušná bezpečnostní opatření.

UPOZORNĚNÍ

znamená, že mohou nastat materiální škody, když se neučiní příslušná bezpečnostní opatření.

Při výskytu více stupňů ohrožení bude vždy použito výstražné upozornění s nejvyšším stupněm. Je-li ve výstražném upozornění s výstražným trojúhelníkem výstraha před škodami na zdraví, pak může být v tomtéž výstražném upozornění ještě připojena výstraha před materiálními škodami.

Kvalifikovaný personál

Výrobek nebo systém, ke kterému náleží tato dokumentace, může obsluhovat pouze **personál s odpovídající kvalifikací**, který bude při provádění stanovených úkolů dodržovat pokyny uvedené v dokumentaci, zejména pak předpisy týkající se bezpečnosti práce. Kvalifikovaný personál je na základě svého vzdělání a zkušeností způsobilý odhalit rizika v souvislosti s obsluhou těchto výrobků či systémů a zabránit možnému ohrožení.

Používání výrobků Siemens v souladu s určením

Mějte na zřeteli následující:

VÝSTRAHA

Výrobky Siemens se smí používat pouze pro účely uvedené v katalogu a v příslušné technické dokumentaci. Pokud se používají cizí výrobky a komponenty, musí být doporučeny nebo schváleny firmou Siemens. Bezporuchový a bezpečný provoz předpokládá odbornou přepravu, skladování, ustavení, montáž, instalaci, uvedení do provozu, obsluhu a údržbu. Musí se dodržovat přípustné podmínky prostředí. Dodržovat se musí také pokyny v příslušné dokumentaci.

Známky

Všechny názvy označené ochrannou známkou ® jsou zapsané známky firmy Siemens AG. Ostatní názvy v této tiskovině mohou být značkami, jejichž používání třetími subjekty pro své účely může porušovat práva majitelů.

Vyloučení odpovědnosti

Zkontrolovali jsme obsah tiskoviny, zda je v souladu s popsaným hardwarem a softwarem. Přesto nelze vyloučit odchylky, takže nemůžeme převzít odpovědnost za kompletní shodu. Údaje v této tiskovině jsou pravidelně kontrolovány, potřebné opravy jsou uvedeny v následujících vydáních.

Předmluva

Dokumentace systému SINUMERIK

Dokumentace systému SINUMERIK je rozčleněna do následujících kategorií:

- Všeobecná dokumentace/katalogy
- Uživatelská dokumentace
- Dokumentace výrobce / servisní dokumentace

Doplňkové informace

Na následující adrese (<https://support.industry.siemens.com/cs/de/en/view/108464614>) naleznete informace k těmto tématům:

- Objednávání dokumentace / přehled tištěných materiálů
- Další odkazy pro stažení dokumentů
- Používejte on-line dokumentaci (vyhledávání a prohledávání příruček/informací)

Pokud budete mít dotazy týkající se technické dokumentace (např. návrhy, opravy), zašlete e-mail na následující adresu (<mailto:docu.motioncontrol@siemens.com>).

mySupport/Dokumentation

Na následující adrese (<https://support.industry.siemens.com/My/ww/en/documentation>) naleznete informace, jak můžete svou dokumentaci individuálně sestavovat na základě dokumentů firmy Siemens a jak ji přizpůsobit, aby odpovídala dokumentaci vašeho vlastního stroje.

Vzdělávání

Na následující adrese (<http://www.siemens.com/sitrain>) naleznete informace týkající se aplikace SITRAIN - školení firmy Siemens pro produkty, systémy a řešení z oblasti pohonové a automatizační techniky.

Často kladené otázky

Často kladené otázky naleznete na stránkách Service&Support (Služby a podpora) v rámci podpory pro jednotlivé produkty (<https://support.industry.siemens.com/cs/de/en/ps/faq>).

SINUMERIK

Pokud budete potřebovat informace o systému SINUMERIK, využijte následující adresu (<http://www.siemens.com/sinumerik>).

Cílová skupina

Předkládaná dokumentace je určena následujícím pracovníkům:

- Programátoři
- Technici mající na starost konfiguraci systémů

Použití

Pomocí této příručky pro programování mohou pracovníci cílové skupiny vyvíjet, psát, testovat a odstraňovat chyby v programech a v obrazovkách uživatelského rozhraní.

Standardní rozsah

V předkládané programovací příručce jsou popisovány funkce standardního rozsahu dodávky. Doplnění nebo změny, které byly provedeny výrobcem stroje, jsou popsány v dokumentaci od tohoto výrobce stroje.

V rámci řídicího 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 zásahu.

Z důvodů zachování přehlednosti neobsahuje tato dokumentace všechny podrobné informace ke všem typům produktu a také nemůže pokrýt veškeré myslitelné případy, které se mohou v průběhu instalace, provozování a údržby vyskytnout.

Technická podpora

Specifická telefonní čísla na pracovníky technické podpory v dané zemi naleznete na internetu na následující adrese (<https://support.industry.siemens.com/sc/ww/en/sc/2090>) v oblasti „Kontakt“.

Informace vztahující se ke struktuře a obsahu

Programovací příručka, "Základy" / "Pro pokročilé"

Popisy programování NC systémů jsou rozděleny do dvou příruček:

1. Základy

Příručka programování "Základy" je určena pro zkušené kvalifikované pracovníky obsluhy stroje a předpokládá odpovídající znalosti pro operace vrtání, frézování a soustružení. Pro vysvětlení příkazů a výrazů, které jsou definovány rovněž podle normy DIN 66025, se používají jednoduché příklady programování.

2. Pro pokročilé

Příručka programování „Pro pokročilé“ slouží technologům, kteří disponují znalostmi o všech možnostech programování. Řídicí systémy SINUMERIK umožňují pomocí speciálního programovacího jazyka vytváření programů pro výrobu složitých obrobků (např. modelované povrchy volných tvarů, koordinace kanálů atd.) a technologům výrazně usnadňuje programování složitých operací.

Dostupnost popisovaných prvků jazyka NC-systému

Všechny prvky jazyka NC-systému, které jsou popisovány v předkládané příručce, jsou pro systém SINUMERIK 840D sl k dispozici. Dostupnost prvků týkající se systému SINUMERIK 828D je zapotřebí zjistit v tabulce "Příkazy: Použitelnost u systému SINUMERIK 828D (Strana 886)".

Obsah

	Předmluva.....	3
1	Základní bezpečnostní upozornění.....	17
1.1	Všeobecná bezpečnostní upozornění.....	17
1.2	Záruční podmínky pro příklady aplikace.....	18
1.3	Industrial Security.....	19
2	Flexibilní programování NC systémů.....	21
2.1	Proměnné.....	21
2.1.1	Systémová data.....	21
2.1.2	Předdefinované uživatelské proměnné: Početní parametry.....	24
2.1.2.1	Kanálový početní parametr (R).....	24
2.1.2.2	Globální početní parametry (RG).....	25
2.1.3	Předdefinované uživatelské proměnné: linkové proměnné.....	27
2.1.4	Definice uživatelských proměnných (DEF).....	29
2.1.5	Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF).	35
2.1.6	Atribut: Inicializační hodnota.....	38
2.1.7	Atribut: Mezní hodnoty (LLI, ULI).....	41
2.1.8	Atribut: Fyzikální jednotka (PHU).....	43
2.1.9	Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB).....	45
2.1.10	Atribut: Datová třída (DCM, DCI, DCU) - jen v systému SINUMERIK 828D.....	50
2.1.11	Přehled definovatelných a znovu definovatelných atributů.....	51
2.1.12	Definice a inicializace proměnných typu pole (DEF, SET, REP).....	52
2.1.13	Definice a inicializace proměnných typu pole (DEF, SET, REP): Další informace.....	56
2.1.14	Datové typy.....	58
2.1.15	Kontrola existence proměnné (ISVAR).....	59
2.1.16	Načítání hodnot atributu/datového typu (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP).....	60
2.2	Nepřímé programování.....	67
2.2.1	Nepřímé programování adres.....	67
2.2.2	Nepřímé programování příkazů G-funkcí.....	69
2.2.3	Nepřímé programování atributů polohy (GP).....	70
2.2.4	Nepřímé programování řádků výrobního programu (EXECSTRING).....	73
2.3	Matematické funkce.....	75
2.4	Porovnávací a logické operace.....	78
2.5	Korekce přesnosti při příkazech porovnávání (TRUNC).....	80
2.6	Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND).....	82
2.7	Priorita operací.....	84
2.8	Možné převádění typů.....	85
2.9	Operace s řetězci.....	86
2.9.1	Převod do datového typu STRING (AXSTRING).....	86

2.9.2	Převod z datového typu STRING (NUMBER, ISNUMBER, AXNAME).....	87
2.9.3	Zřetězení proměnných typu String (<<).....	88
2.9.4	Převádění na malá/velká písmena (TOLOWER, TOUPPER).....	89
2.9.5	Zjištění délky řetězce (STRLEN).....	90
2.9.6	Vyhledávání znaků/řetězců v řetězci (INDEX, RINDEX, MINDEX, MATCH).....	91
2.9.7	Vybírání dílčího řetězce (SUBSTR).....	92
2.9.8	Načítání a zápis jednotlivých znaků.....	93
2.9.9	Formátování řetězce (SPRINT).....	94
2.10	Programové skoky a větvení.....	103
2.10.1	Skok zpátky na začátek programu (GOTOS).....	103
2.10.2	Programové skoky na návěští skoků (GOTOB, GOTOF, GOTO, GOTOC).....	104
2.10.3	Větvení programu (CASE ... OF ... DEFAULT ...).....	107
2.11	Opakování části programu (REPEAT, REPEATB, ENDLABEL, P).....	109
2.12	Řídící struktury.....	115
2.12.1	Podmíněné příkazy a větvení (IF, ELSE, ENDIF).....	116
2.12.2	Nekonečná programová smyčka (LOOP, ENDLOOP).....	118
2.12.3	Smyčka s počítadlem (FOR ... TO ..., ENDFOR).....	118
2.12.4	Programová smyčka s podmínkou na začátku smyčky (WHILE, ENDWHILE).....	120
2.12.5	Programová smyčka s podmínkou na konci smyčky (REPEAT, UNTIL).....	121
2.12.6	Příklad programu se vnořenými řídicími strukturami.....	121
2.13	Příkazy pro koordinaci (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM).....	123
2.14	Rutiny přerušení (ASUP).....	128
2.14.1	Funkce rutiny přerušení.....	128
2.14.2	Sestavování rutin přerušení.....	129
2.14.3	Přiřazování a spouštění rutin přerušení (SETINT, PRIO, BLSYNC).....	130
2.14.4	Deaktivování/opětovné aktivování přiřazení rutiny přerušení (DISABLE, ENABLE).....	132
2.14.5	Vymazání přiřazení rutiny přerušení (CLRINT).....	132
2.14.6	Rychlé pozvednutí od kontury (SETINT LIFTFAST, ALF).....	133
2.14.7	Směr pohybu při rychlém pozvednutí od kontury	135
2.14.8	Posloupnost pohybů při rutinách přerušení.....	138
2.15	Výměna osy, výměna větvena (RELEASE, GET, GETD).....	140
2.16	Předávání osy jinému kanálu (AXTOCHAN).....	145
2.17	Aktivování strojních parametrů (NEWCONF).....	147
2.18	Zápis do souboru (WRITE).....	148
2.19	Vymazání souboru (DELETE).....	152
2.20	Čtení řádků v souboru (READ).....	153
2.21	Kontrola existence souboru (ISFILE).....	155
2.22	Zjišťování informací o souboru (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO).....	156
2.23	Zaokrouhlení (ROUNDUP).....	158
2.24	Technika podprogramů.....	159
2.24.1	Všeobecně.....	159
2.24.1.1	Podprogram.....	159
2.24.1.2	Názvy podprogramů.....	160
2.24.1.3	Vnoření podprogramů.....	161

2.24.1.4	Cesta pro vyhledávání.....	162
2.24.1.5	Formální a skutečný parametr.....	163
2.24.1.6	Předávání parametrů.....	163
2.24.2	Definice podprogramu.....	165
2.24.2.1	Podprogram bez předávání parametrů.....	165
2.24.2.2	Podprogram s předáváním parametrů Call-by-Value (PROC).....	166
2.24.2.3	Podprogram s předáváním parametrů Call-by-Reference (PROC, VAR).....	168
2.24.2.4	Ukládání modálních G-funkcí (SAVE).....	170
2.24.2.5	Potlačení zpracovávání blok po bloku (SBLOF, SBLON).....	171
2.24.2.6	Potlačení vypisování aktuálního bloku (DISPLOF, DISPLON, ACTBLOCNO).....	176
2.24.2.7	Označení podprogramů s přípravou (PREPRO).....	179
2.24.2.8	Návrat zpět z podprogramu - M17.....	180
2.24.2.9	Návrat z podprogramu pomocí příkazu RET.....	181
2.24.2.10	Návrat z podprogramu s nastavitelnými parametry (RET ...).....	182
2.24.2.11	Návrat z podprogramu s nastavitelnými parametry (RETB ...).....	188
2.24.3	Volání podprogramu.....	192
2.24.3.1	Volání podprogramu bez předávání parametrů.....	192
2.24.3.2	Volání podprogramu s předáváním parametrů (EXTERN).....	194
2.24.3.3	Počet opakování programu (P).....	196
2.24.3.4	Modální volání podprogramu (MCALL).....	197
2.24.3.5	Nepřímé volání podprogramu (CALL).....	199
2.24.3.6	Nepřímé volání podprogramu se zadáním úseku programu, který á být zpracován (CALL BLOCK ... TO ...).....	200
2.24.3.7	Nepřímé volání programu naprogramovaného v jazyce ISO (ISOCALL).....	202
2.24.3.8	Volání podprogramu s udáním cesty a parametrů (PCALL).....	202
2.24.3.9	Rozšíření cesty pro vyhledávání při volání podprogramu (CALLPATH).....	203
2.24.3.10	Zpracovávání externího podprogramu (840D sl) (EXTCALL).....	204
2.24.3.11	Zpracovávání externího podprogramu (828D) (EXTCALL).....	208
2.25	Technika maker (DEFINE ... AS).....	212
3	Správa souborů a programů.....	215
3.1	Programová paměť.....	215
3.1.1	Programová paměť v NC systému.....	215
3.1.2	Externí programová paměť.....	217
3.1.3	Adresování souborů v paměti pro programy.....	219
3.1.4	Cesta pro vyhledávání při volání podprogramu.....	223
3.1.5	Zjištění cesty a názvu souboru.....	225
3.2	Pracovní paměť (CHANDATA, COMPLETE, INITIAL).....	227
4	Chráněné oblasti.....	231
4.1	Definice chráněných oblastí (CPROTDEF, NPROTDEF).....	231
4.2	Aktivování/deaktivování chráněné oblasti (CPROT, NPROT).....	235
4.3	Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI).....	239
5	Speciální příkazy dráhy.....	251
5.1	Najíždění na kódované pozice (CAC, CIC, CDC, CACP, CACN).....	251
5.2	Splinová interpolace (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL).....	252

5.3	Splínový svazek (SPLINEPATH).....	263
5.4	Aktivování/deaktivování komprese NC bloků (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF).....	265
5.5	Polynomická interpolace (POLY, POLYPATH, PO, PL).....	266
5.6	Nastavitelný vztah k dráze (SPATH, UPATH).....	272
5.7	Měření se spínací sondou (MEAS, MEAW).....	274
5.8	Měření pomocí specifické osy (MEASA, MEAWA, MEAC) (volitelný doplněk).....	277
5.9	Speciální funkce pro uživatele OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829).....	287
5.10	Snížení posuvu se zpožděním v rozích (FENDNORM, G62, G621)	288
5.11	Programovatelné kritérium konce pohybu (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA).....	289
6	Transformace souřadného systému (Frame).....	293
6.1	Transformace souřadnic pomocí proměnné typu frame.....	293
6.1.1	Předdefinované proměnné typu frame (\$P_CHBFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME).....	295
6.2	Přiřazování hodnoty framu.....	299
6.2.1	Přímé přiřazení hodnoty (hodnota osy, úhlu, měřítko).....	299
6.2.2	Načítání a editace složek framu (TR, FI, RT, SC, MI).....	301
6.2.3	Výpočty s framy.....	302
6.2.4	Definice proměnných typu frame (DEF FRAME).....	304
6.3	Hrubé a jemné posunutí (CTRANS, CFINE).....	305
6.4	Externí posunutí počátku (\$AA_ETRANS).....	307
6.5	Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod (PRESETON).....	309
6.6	Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod (PRESETONS).....	311
6.7	Výpočet framu na základě 3 změřených bodů v prostoru (MEAFRAME).....	313
6.8	Globální framy v NCU.....	317
6.8.1	Specifické kanálové framy (\$P_CHBFR, \$P_UBFR).....	318
6.8.2	Framy aktivní v kanálu.....	318
7	Transformace.....	325
7.1	Všeobecné programování různých druhů transformace.....	325
7.1.1	Pohyby provádějící změnu orientace při transformacích.....	327
7.1.2	Přehled transformace orientace TRAORI.....	331
7.2	Transformace ve třech, čtyřech a pěti osách (TRAORI).....	333
7.2.1	Všeobecné souvislosti v případě kardanové nástrojové hlavičky.....	333
7.2.2	Transformace ve třech, čtyřech a pěti osách (TRAORI).....	336
7.2.3	Variety programování orientace a základního nastavení (ORIRESET).....	337
7.2.4	Programování orientace nástroje (A..., B..., C..., LEAD, TILT).....	339
7.2.5	Frézování na čelní ploše (A4, B4, C4, A5, B5, C5).....	345
7.2.6	Vztah orientačních os (ORIWKS, ORIMKS).....	346
7.2.7	Programování orientačních os (ORIAxes, ORIVect, ORIEuler, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2).....	348

7.2.8	Programování orientace podél kuželové plášťové plochy (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO).....	350
7.2.9	Zadání orientace pomocí druhého styčného bodu (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=).....	353
7.3	Polynomický popis orientace (PO[úhel], PO[souřadnice]).....	356
7.4	Otáčení orientace nástroje (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA).....	358
7.5	Orientace vzhledem k dráze.....	361
7.5.1	Druhy orientace vztažené k dráze.....	361
7.5.2	Otáčení orientace nástroje vzhledem k dráze (ORIPATH, ORIPATHS, úhel otočení).....	362
7.5.3	Interpolace otočení nástroje vzhledem k dráze (ORIROTC, THETA).....	363
7.5.4	Vyhazení průběhu orientace (ORIPATHS A8=, B8=, C8=).....	365
7.6	Komprimování orientace (COMPON, COMPCURV, COMPCAD, COMPSURF).....	367
7.7	Aktivování/deaktivování vyhlazování charakteristiky orientace (ORISON, ORISOF):.....	370
7.8	Kinematická transformace.....	372
7.8.1	Aktivování transformace čelní plochy (TRANSMIT).....	372
7.8.2	Aktivování transformace válcového pláště (TRACYL).....	372
7.8.3	Aktivování transformace šikmého úhlu s naprogramovatelným úhlem (TRAANG).....	375
7.8.4	Šikmé zapichování na bruskách (G5, G7).....	376
7.9	Aktivování zřetěžené transformace (TRACON).....	378
7.10	Posuv PTP v kartézských souřadnicích.....	380
7.10.1	Aktivování/deaktivování posuvu PTP v kartézských souřadnicích (PTP, PTPG0, PTPWOC, CP).....	380
7.10.2	Zadávání polohy kloubu (STAT).....	381
7.10.3	Zadání znaménka úhlu osy (TU).....	385
7.10.4	Příklad 1: Pohyb PTP robota se 6 osami pomocí transformace ROBX.....	388
7.10.5	Příklad 2: Najíždění PTP při obecné 5-osé transformaci.....	389
7.10.6	Příklad 3: PTPG0 a TRANSMIT.....	389
7.11	Okrajové podmínky při aktivování transformace.....	391
7.12	Deaktivování transformace (TRAFOOF).....	392
8	Kinematické řetězce.....	393
8.1	Mazání komponent (DELOBJ).....	393
8.2	Zjišťování indexu podle názvu (NAMETOINT).....	396
9	Zabránění vzniku kolizí u kinematických řetězců.....	399
9.1	Kontrola dvojice oblastí chráněných kvůli kolizi (COLLPAIR).....	400
9.2	Vyvolání nového výpočtu modelu stroje pro prevenci kolizí (PROTA).....	401
9.3	Nastavení stavu chráněné oblasti (PROTS).....	402
9.4	Stanovení vzdálenosti mezi dvěma chráněnými oblastmi (PROTD).....	403
10	Transformace pomocí kinematických řetězců.....	405
10.1	Aktivování transformace (TRAFOON).....	405
10.2	Modifikace transformace orientace po měření stroje (CORRTRAFO).....	407

11	Korekční parametry nástroje.....	415
11.1	Paměť korekcí.....	415
11.2	Aditivní korekce.....	418
11.2.1	Aktivování aditivních korekcí (DL).....	418
11.2.2	Definice opotřebenění a seřizovacích hodnot (\$TC_SCPxy[t,d], \$TC_ECPxy[t,d]).....	419
11.2.3	Vymazání aditivních korekcí (DELDL).....	420
11.3	Korekce nástroje - speciální zacházení.....	422
11.3.1	Zrcadlové převrácení délek nástroje.....	424
11.3.2	Vyhodnocování znaménka opotřebenění.....	424
11.3.3	Souřadný systém aktivního obrábění (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS).....	425
11.3.4	Délka nástroje a změna roviny.....	428
11.4	On-line korekce nástroje.....	430
11.4.1	Definice polynomické funkce (FCTDEF).....	430
11.4.2	Zápis on-line korekce nástroje, kontinuálně (PUTFTOCF).....	431
11.4.3	Diskrétní zápis on-line korekce nástroje (PUTFTOC).....	432
11.4.4	Aktivování/deaktivování on-line korekce nástroje (FTOCON/FTOCOF).....	433
11.5	3D korekce rádiusu nástroje.....	434
11.5.1	Aktivování 3D korekce rádiusu nástroje pro 3D obvodové frézování (CUT3DC, CUT3DCD, ISD).....	434
11.5.2	Aktivování 3D korekce rádiusu nástroje pro 3D frézování na čelní ploše (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD).....	438
11.5.3	3D obvodové frézování se zohledněním hraniční plochy (CUT3DCC, CUT3DCCD).....	444
11.6	Orientace nástroje (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST).....	449
11.7	Volné zadávání D-čísel, číslo bříty.....	455
11.7.1	Volné zadávání D-čísel, číslo bříty (adresa CE).....	455
11.7.2	Volné zadávání D-čísel: Kontrola D-čísla (CHKDNO).....	455
11.7.3	Volné zadávání D-čísel: Přejmenovávání D-čísel (GETDNO, SETDNO).....	456
11.7.4	Volné zadávání D-čísel: Zjišťování T-čísla k zadanému D-číslu (GETACTTD).....	457
11.7.5	Volné zadávání D-čísel: Nastavení D-čísla jako neplatného (DZERO).....	457
11.8	Kinematika držáku nástroje.....	458
11.9	Délková korekce nástroje pro orientovatelný držák nástroje (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ).....	463
11.10	On-line korekce délky nástroje (TOFFON, TOFFOF).....	466
11.11	Modifikace korekčních parametrů u točivých nástrojů.....	469
11.11.1	Výpočet orientace (ORISOLH).....	469
11.11.2	Aktivování modifikací korekčních parametrů u točivých nástrojů (CUTMOD, CUTMODK).....	479
11.12	Práce s nástrojovým prostředím.....	487
11.12.1	Uložení nástrojového prostředí (TOOLENV).....	487
11.12.2	Vymazání nástrojového prostředí (DELTOOLENV).....	490
11.12.3	Načtení T-čísla, D-čísla a DL-čísla (GETTENV).....	491
11.12.4	Načítání informací týkajících se uložených nástrojových prostředí (\$P_TOOLENV, \$P_TOOLENV).....	492
11.12.5	Načtení délek nástroje, příp. složek délky nástroje (GETTCOR).....	492
11.12.6	Změna komponent nástroje (SETTCOR).....	499

11.13	Načítání přiřazení délek nástroje L1, L2, L3 osám souřadného systému (LENTOAX).....	512
12	Chování při pohybu po dráze.....	515
12.1	Tangenciální řízení.....	515
12.1.1	Definice vazby (TANG).....	515
12.1.2	Aktivování vytváření pomocných bloků (TLIFT).....	517
12.1.3	Aktivování vazby (TANGON).....	518
12.1.4	Deaktivování vazby (TANGOF).....	519
12.1.5	Vymazání vazby (TANGDEL).....	519
12.2	Průběh charakteristiky posuvu (FNORM, FLIN, FCUB, FPO).....	521
12.3	Chování zrychlení:.....	526
12.3.1	Způsoby chování při změnách zrychlení (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA).....	526
12.3.2	Ovlivňování chování zrychlení u vlečných os (VELOLIMA, ACCLIMA, JERKLIMA).....	528
12.3.3	Aktivování specifických technologických hodnot dynamiky (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH).....	530
12.4	Najíždění s dopřednou regulací (FFWON, FFWOF).....	532
12.5	Programovatelná přesnost kontury (CPRECON, CPRECOF).....	533
12.6	Zpracování programu s pamětí předběžného zpracování (STOPFIFO, STARTFIFO, FIFCTRL, STOPRE)	535
12.7	Definice oblasti odloženého zastavení (DELAYFSTON, DELAYFSTOF).....	538
12.8	Zabránění pozice programu pro SERUPRO (IPTRLOCK, IPTRUNLOCK).....	541
12.9	Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)	543
12.10	Ovlivňování vedení pohybu.....	552
12.10.1	Procentuální korekce ryvu (JERKLIM).....	552
12.10.2	Procentuální korekce rychlosti (VELOLIM).....	553
12.10.3	Příklad programování pro příkazy JERKLIM a VELOLIM.....	555
12.11	Programování tolerance kontury/orientace (CTOL, OTOL, ATOL).....	556
12.12	Chování při přechodu na další blok, když je aktivní vazba (CPBC).....	561
13	Spojení os vazbou.....	563
13.1	Vlečení (TRAILON, TRAILOF).....	563
13.2	Tabulky křivek (CTAB).....	568
13.2.1	Definice tabulek křivek (CTABDEF, CATBEND).....	568
13.2.2	Kontrola existence tabulky křivky (CTABEXISTS).....	575
13.2.3	Mazání tabulek křivek (CTABDEL).....	575
13.2.4	Blokování tabulek křivek proti mazání a přepisování (CTABLOCK, CTABUNLOCK).....	576
13.2.5	Tabulky křivek: Zjišťování vlastností tabulek (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD).....	578
13.2.6	Čtení hodnot z tabulky křivky (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX).....	579
13.2.7	Tabulky křivek: Kontrola využití systémových zdrojů (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL).....	584

13.3	Osová vazba řídící hodnotou (LEADON, LEADOF).....	586
13.4	Elektronická převodovka (EG).....	592
13.4.1	Definice elektronické převodovky (EGDEF).....	592
13.4.2	Zapnutí elektronické převodovky (EGON, EGONSYN, EGONSUNE).....	593
13.4.3	Vypnutí elektronické převodovky (EGOFS, EGOFC).....	596
13.4.4	Vymazání definice elektronické převodovky (EGDEL).....	597
13.4.5	Otáčkový posuv (G95) / elektronická převodovka (FPR).....	598
13.5	Synchronní vřeteno.....	599
13.5.1	Synchronní vřeteno: Programové příkazy (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC).....	599
13.6	Generická vazba (CP...).....	610
13.7	Vazba Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS).....	617
14	Synchronní akce.....	621
14.1	Definice synchronní akce.....	621
15	Pohyb tam a zpět.....	623
15.1	Asynchronní oscilační pohyby (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB).....	623
15.2	Oscilační pohyby řízené prostřednictvím synchronních akcí (OSCILL).....	628
16	Lisování a prostřihování.....	635
16.1	Aktivování/deaktivování.....	635
16.1.1	Aktivování/deaktivování lisování a prostřihování (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC).....	635
16.2	Automatické rozdělení dráhy.....	640
16.2.1	Rozdělení cesty u dráhových os.....	642
16.2.2	Rozdělení cesty u jednotlivých os.....	644
17	Broušení.....	647
17.1	Aktivování/deaktivování speciálního monitorování pro brusné nástroje (TMON, TMOF) ...	647
18	Další funkce.....	649
18.1	Osové funkce (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL).....	649
18.2	Přepínatelné geometrické osy (GEOAX).....	652
18.3	Osový zásobník (AXCTSWE, AXCTSWED, AXCTSWEC).....	657
18.4	Čekání na platnou pozici osy (WAITENC).....	659
18.5	Programovatelné přepnutí sady parametrů (SCPARA).....	661
18.6	Kontrola existujícího rozsahu NC jazyka (STRINGIS).....	663
18.7	Interaktivní vyvolávání oken z výrobního programu (MMC).....	667
18.8	Doba zpracování programu / počítadlo obrobků.....	672
18.8.1	Doba zpracování programu.....	672
18.8.2	Počítadlo obrobků.....	675
18.9	Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE).....	677

18.10	Alarmy (SETAL).....	682
18.11	Rozšířené zastavování a odjíždění (ESR).....	684
18.11.1	ESR řízené NC systémem.....	685
18.11.1.1	Odjíždění řízené NC systémem (POLF, POLFA, POLFMASK, POLFMLIN).....	685
18.11.1.2	Zastavení NC systémem.....	689
18.11.2	ESR nezávislé na pohonu.....	690
18.11.2.1	Konfigurace zastavování nezávislého na pohonu (ESRS).....	690
18.11.2.2	Konfigurace odjíždění nezávislého na pohonu (ESRR).....	690
18.12	Definice surového obrobku (WORKPIECE).....	692
18.13	Přepnutí režimu jazyka (G290, G291).....	696
19	Vlastní programy pro oddělování třísky.....	699
19.1	Podporované funkce pro oddělování třísky.....	699
19.2	Sestavování kontury (CONTPRON).....	700
19.3	Sestavování kódované tabulky kontury (CONTPRON).....	706
19.4	Zjištění průsečíku mezi dvěma konturovými prvky (INTERSEC).....	710
19.5	Pohyb po konturových prvcích v tabulce blok po bloku (EXECTAB).....	712
19.6	Výpočet parametrů kruhu (CALCDAT).....	713
19.7	Ukončení přípravy kontury (EXECUTE).....	715
20	Programování externích cyklů.....	717
20.1	Technologické cykly.....	717
20.1.1	Úvod.....	717
20.1.2	Přehled pro jednotlivé technologie.....	718
20.1.3	HOLES1 - Řada děr.....	720
20.1.4	HOLES2 - Díry uspořádané na kruhovém oblouku.....	720
20.1.5	POCKET3 - Frézování pravoúhlé kapsy.....	722
20.1.6	POCKET4 - Frézování kruhové kapsy.....	725
20.1.7	SLOT1 - Podélná drážka.....	728
20.1.8	SLOT2 - Kruhová drážka.....	730
20.1.9	LONGHOLE - Podlouhlá díra.....	733
20.1.10	CYCLE60 - Cyklus pro gravírování.....	735
20.1.11	CYCLE61 - Rovinné frézování.....	737
20.1.12	CYCLE62 - volání kontury.....	739
20.1.13	CYCLE63 - Frézování konturové kapsy.....	740
20.1.14	CYCLE64 - Předvrtání konturové kapsy.....	742
20.1.15	CYCLE70 - Frézování závitů.....	744
20.1.16	CYCLE72 - Frézování po dráze.....	745
20.1.17	CYCLE76 - Frézování pravoúhlého čepu.....	750
20.1.18	CYCLE77 - Frézování kruhového čepu.....	752
20.1.19	CYCLE78 - Frézování vrtaných závitů.....	754
20.1.20	CYCLE79 - Mnohohran.....	757
20.1.21	CYCLE81 - Vrtání, navrtávání středících důlků.....	759
20.1.22	CYCLE82 - Vrtání, čelní zahlubování.....	760
20.1.23	CYCLE83 - Vrtání hlubokých děr.....	762
20.1.24	CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky.....	765
20.1.25	CYCLE85 - Vystružování.....	768

20.1.26	CYCLE86 - Vyvrtávání.....	769
20.1.27	CYCLE92 - Upichování.....	771
20.1.28	CYCLE95 - Oddělování třísky od kontury.....	772
20.1.29	CYCLE98 - Řetězec závitů.....	774
20.1.30	CYCLE99 - Soustružení závitu.....	778
20.1.31	CYCLE435 - Nastavení souřadného systému orovnávače.....	783
20.1.32	CYCLE495 - Profilování.....	783
20.1.33	CYCLE800 - Naklápění.....	785
20.1.34	CYCLE801 - Mřížka nebo obdélník.....	789
20.1.35	CYCLE802 - Libovolné pozice.....	790
20.1.36	CYCLE830 - Vrtání hlubokých děr 2.....	793
20.1.37	CYCLE832 - Vysokorychlostní obrábění.....	799
20.1.38	CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou.....	802
20.1.39	CYCLE899 - Frézování otevřené drážky.....	805
20.1.40	CYCLE930 - Zápich.....	808
20.1.41	CYCLE940 - Odlehčovací zápich určitého tvaru.....	810
20.1.42	CYCLE951 - Oddělování třísky.....	813
20.1.43	CYCLE952 - Zápichy na kontuře.....	816
20.1.44	CYCLE4071 - Podélné broušení s přísuvem v bodu obratu.....	822
20.1.45	CYCLE4072 - Podélné broušení s přísuvem v bodě obratu a se signálem pro přerušení.....	823
20.1.46	CYCLE4073 - Podélné broušení se spojitým přísuvem.....	827
20.1.47	CYCLE4074 - Podélné broušení se spojitým přísuvem a se signálem pro přerušení.....	828
20.1.48	CYCLE4075 - Plošné broušení s přísuvem v bodě obratu.....	831
20.1.49	CYCLE4077 - Plošné broušení s přísuvem na bod obratu a se signálem pro přerušení.....	834
20.1.50	CYCLE4078 - Plošné broušení se spojitým přísuvem.....	837
20.1.51	CYCLE4079 - Plošné broušení s přerušovaným přísuvem.....	839
20.1.52	GROUP_BEGIN - Začátek programového bloku.....	841
20.1.53	GROUP_END - konec programového bloku.....	842
20.1.54	GROUP_ADDEND - konec doplňkového bloku najíždění.....	842
20.1.55	Okrajové podmínky.....	842
20.1.55.1	Odstupňování technologie na obrazovkách cyklů.....	842
20.2	Měřicí cykly.....	845
21	Tabulky.....	847
21.1	Příkazy.....	847
21.2	Příkazy: Použitelnost u systému SINUMERIK 828D	886
21.2.1	Variety řídicího systému pro frézování / soustružení.....	886
21.2.2	Variety řídicího systému pro broušení.....	913
21.3	Aktuální jazyk v HMI.....	941
A	Přílohy.....	943
A.1	Seznam zkratk.....	943
A.2	Přehled dokumentace.....	952
	Glosář.....	953
	Rejstřík.....	975

Základní bezpečnostní upozornění

1.1 Všeobecná bezpečnostní upozornění

 VÝSTRAHA
Ohrožení života v případě zanedbání bezpečnostních upozornění a zbývajících rizik V případě zanedbání bezpečnostních upozornění a zbývajících rizik, která jsou uvedena v odpovídající dokumentaci k hardwaru, může dojít k nehodám s těžkými nebo i smrtelnými úrazy. • Dodržujte bezpečnostní upozornění v dokumentaci k hardwaru. • Mějte na paměti zbývající rizika, která byla zjištěna při jejich vyhodnocování.

 VÝSTRAHA
Nesprávná funkce stroje v důsledku chybného nebo změněného nastavení parametrů V důsledku chybného nebo změněného nastavení parametrů se může vyskytnout chybná funkce stroje, která může mít za následek úraz nebo i smrt. • Chraňte nastavení parametrů před přístupem nepovolaných osob. • Pro případ možné nesprávné funkce zajistěte vhodná opatření, např. spínače pro nouzové zastavení nebo nouzové vypínače.

1.2 Záruční podmínky pro příklady aplikace

Příklady aplikací jsou nezávazné a nekladou si žádné nároky na úplnost, ani pokud jde o konfiguraci a vybavení, ani pokud jde o všechny možné eventuality. Příklady aplikací nepřestávají žádná specifická zákaznická řešení, mají pouze nabízet určitou pomoc při řešení typických problémů.

Jako uživatel nesete odpovědnost za odborné zacházení s popisovanými produkty. Příklady aplikací vás nezbavují povinnosti zajistit bezpečné zacházení při použití, instalaci, provozu a údržbě.

1.3 Industrial Security

Poznámka

Industrial Security

Siemens nabízí produkty a řešení s funkcemi Industrial Security, které podporují bezpečný provoz zařízení, systémů, strojů a sítí.

Aby byla zařízení, systémy, stroje a sítě zabezpečeny proti kybernetickým útokům, je potřeba implementovat komplexní koncept Industrial Security, který odpovídá aktuálnímu stavu techniky (a následně jej udržovat). Produkty a řešení firmy Siemens představují pouze součást takového konceptu.

Za zabránění přístupu nepovolaných osob ke svým zařízením, systémům, strojům a sítím je odpovědný zákazník. Systémy, stroje a komponenty by měly být propojeny pouze podnikovou sítí nebo internetem, pokud a jakmile je to nutné a byla učiněna příslušná ochranná opatření (např. používání firewallu a segmentace sítě).

Kromě toho by měla být zohledněna doporučení firmy Siemens k příslušným ochranným opatřením. Pokud budete potřebovat další informace týkající se Industrial Security, naleznete je na adrese:

Industrial Security (<http://www.siemens.com/industrialsecurity>)

Produkty a řešení firmy Siemens jsou soustavně dále vyvíjeny, aby byly ještě bezpečnější. Firma Siemens výslovně doporučuje provádět aktualizace, jakmile jsou odpovídající aktualizací soubory k dispozici, a vždy používat pouze aktuální verze produktu. Používání zastaralých nebo již nepodporovaných verzí může zvyšovat riziko kybernetických útoků.

Abyste byli neustále informováni o aktualizacích produktu, přihlaste se k odběru zpravodaje Siemens Industrial Security RSS Feed na stránkách:

Industrial Security (<http://www.siemens.com/industrialsecurity>)

Další informace naleznete na internetu:

Příručka pro návrh konstrukce Industrial Security (<https://support.industry.siemens.com/cs/ww/en/view/108862708>)



VÝSTRAHA

Nejisté provozní stavy způsobené manipulací s programovým vybavením

Manipulace s programovým vybavením, např. viry, trojské koně, malware nebo červi, mohou způsobit nejisté provozní stavy vašeho zařízení, které mohou mít za následek smrt, těžké ublížení na zdraví a materiální škody.

- Provádějte pravidelné aktualizace softwaru.
- Komponenty automatizačního a pohonového systému integrujte do celkové koncepce Industrial Security zařízení nebo stroje v souladu s nejmodernějšími technickými poznatky.
- Do své celkové koncepce Industrial Security zahrňte všechny použité produkty.
- Prostřednictvím odpovídajících ochranných opatření, např. antivirovým programem, chraňte soubory na výměnných paměťových médiích před škodlivým softwarem.
- Na měniči aktivujte funkci „Ochrana know-how“, aby byl pohon chráněn před úpravami provedenými osobami bez příslušného oprávnění.

Flexibilní programování NC systémů

2.1 Proměnné

Prostřednictvím použití proměnných z oblastí systémových dat a uživatelských dat, zejména ve spojení s matematickými funkcemi a řídicími strukturami, mohou být sestavovány flexibilní NC programy a cykly.

- **Systémová data**
V systémových datech se nacházejí proměnné, které jsou v systému předem definovány. Tyto proměnné mají definovaný význam. Používají se především v rámci programového vybavení systému. Uživatel může v MC programech a v cyklech tyto proměnné číst a může do nich i zapisovat. Příklad: Strojní parametry, nastavované parametry, systémové proměnné.
I když je význam systémových dat pevně předem stanoven, jejich vlastnosti může uživatel prostřednictvím opětovných definic v určité míře měnit.
Viz "Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF) (Strana 35)".
- **Uživatelská data**
V uživatelských datech jsou uloženy proměnné definované uživatelem a jejich význam je určen pouze uživatelem. Systém je nevyhodnocuje.
Uživatelská data jsou rozdělena následujícím způsobem:
 - **Předdefinované uživatelské proměnné**
Předdefinované uživatelské proměnné jsou proměnné, které již jsou v systému definovány, a jejich počet může být nastaven pomocí strojních parametrů. Vlastnosti těchto proměnných mohou být uživatelem upravovány.
Viz "Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF) (Strana 35)".
 - **Uživatelem definované proměnné**
Uživatelem definované proměnné jsou proměnné, které jsou definovány uživatelem a v systému jsou založeny při zpracovávání programu. Jejich počet, datový typ, možnosti zobrazování a všechny ostatní vlastnosti jsou definovány výlučně uživatelem.
Viz "Definice uživatelských proměnných (DEF) (Strana 29)".

2.1.1 Systémová data

Systémová data zahrnují v systému předem definované proměnné, které ve NC programech a cyklech umožňují přístup k aktuálnímu nastavení parametrů řídicího systému, ale také ke stavovým informacím o stroji, řídicím systému a o procesech.

Proměnné předběžného zpracování

Proměnné předběžného zpracování jsou systémová data, která jsou čtena a do kterých se zapisuje v souvislosti s předběžným zpracováním, tzn. v okamžiku interpretace bloku, v němž jsou tyto proměnné naprogramovány. Proměnné předběžného zpracování nezpůsobují jeho zastavení.

Proměnné hlavního zpracování

Proměnné hlavního zpracování jsou systémová data, která jsou čtena nebo do kterých se zapisuje v souvislosti s hlavním zpracováním, tzn. v okamžiku zpracování bloku, v němž jsou tyto proměnné naprogramovány. Proměnné hlavního zpracování následující:

- Proměnné, které mohou být naprogramovány v synchronních akcích (čtení/zápis)
- Proměnné, které mohou být naprogramovány v NC programu a které spouštějí zastavení předběžného zpracování (čtení/zápis)
- Proměnné, které mohou být naprogramovány v NC programu a jejichž hodnota se zjišťuje v rámci předběžného zpracování, zapisuje se do nich ale až v rámci hlavního zpracování (synchronizovaně s hlavním zpracováním: jen zápis)

Systém předpon

Kvůli zvláštnímu označování systémových dat se za normálních okolností k jejich názvu připojuje předpona, která se skládá ze znaku \$, za nímž následuje jedno nebo dvě písmena a znak podtržení:

\$ + 1. Písmeno	Význam: Druh dat
Data předběžného zpracování (systémová data, jejichž čtení/zápis se uskutečňují v rámci předběžného zpracování)	
\$M	Strojní parametry ¹⁾
\$S	Nastavované parametry, chráněné oblasti ¹⁾
\$T	Parametry správy nástrojů
\$P	Programovatelné hodnoty
\$C	Proměnné cyklů v cyklech ISO
\$O	Data volitelných doplňků
R	R-Parametry (početní parametry) ²⁾
Data hlavního zpracování (systémová data, jejichž čtení/zápis se uskutečňují v rámci hlavního zpracování)	
\$\$M	Strojní parametry ¹⁾
\$\$S	Nastavované parametry ¹⁾
\$A	Aktuální data hlavního zpracování
\$V	Data regulátoru polohy

\$ + 1. Písmeno	Význam: Druh dat
\$R	R-Parametry (početní parametry) ²⁾
¹⁾ To, zda se se strojními nebo s nastavovanými parametry zachází jako s proměnnými předběžného nebo hlavního zpracování programu, závisí na tom, zda jsou zapsány s jedním nebo se dvěma znaky \$. Způsob zápisu si můžete zvolit v závislosti na specifické aplikaci. ²⁾ Když jsou R-parametry ve výrobním programu / cyklu používány jako proměnné předběžného zpracování, nezapisuje se žádná předpona, např. R10. V případě použití v synchronní akci jako proměnná hlavního zpracování se předpona zapisuje s jedním znakem \$, např. \$R10.	

2. písmeno	Význam: Zobrazování
N	Globální proměnná NC systému (NCK)
C	Kanálová proměnná (Channel)
A	Osová proměnná (Axis)

Okrajové podmínky

Výjimky ze systému předpon

Následující systémové proměnné se odlišují od výše popisovaného systému předpon:

- \$TC_...: 2. písmeno v tomto případě neodkazuje na kanálové proměnné, nýbrž na systémové proměnné vztahující se k držáku nástroje (TC = Tool Carrier).
- \$P_ ...: Kanálové systémové proměnné

Použití strojních a nastavovaných parametrů v synchronních akcích

Při použití strojních a nastavovaných parametrů v synchronních akcích je možné pomocí předpony stanovit, zda se má čtení/zápis daného strojního/nastavovaného parametru uskutečňovat synchronně s předběžným nebo hlavním zpracováním.

Pokud zůstává parametr v průběhu zpracovávání nezměněn, je možno jej číst v průběhu předběžného zpracování. Předpona strojního nebo nastavovaného parametru se pro tento účel zapisuje s jedním znakem \$.

```
ID=1 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Pokud se parametr v průběhu zpracovávání mění, musí jeho čtení/zápis probíhat synchronně s hlavním zpracováním. Předpona strojního nebo nastavovaného parametru se pro tento účel zapisuje se dvěma znaky \$.

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Poznámka

Zapisování do strojních a nastavovaných parametrů

Při zapisování do strojních nebo nastavovaných parametrů je nutno dávat pozor na to, aby aktivní úroveň přístupových oprávnění při zpracovávání výrobního programu / cyklu umožňovala přístup za účelem zápisu a aby byl v platnosti parametr "IMMEDIATE".

Literatura

Pokud budete potřebovat úplný přehled všech systémových proměnných, viz:
Příručka pro sestavy systémových proměnných

Viz také

Proměnné (Strana 21)

2.1.2 Předdefinované uživatelské proměnné: Početní parametry

2.1.2.1 Kanálový početní parametr (R)

Kanálové početní parametry nebo R-parametry jsou předem definované uživatelské proměnné s označením R definované jako pole s datovým typem REAL. Z historických důvodů je pro R-parametry vedle způsobu zápisu s indexem pole, např. $R[10]$, přípustný také způsob zápisu bez indexu pole, např. $R10$.

Při použití v synchronních akcích musí být na začátku uveden znak \$, např. $\$R10$.

Syntaxe

V případě použití jako proměnná předběžného zpracování:

$R<n>$

$R[<\text{výraz}>]$

V případě použití jako proměnná hlavního zpracování:

$\$R<n>$

$\$R[<\text{výraz}>]$

Význam

R:	Identifikátor při použití jako proměnná předběžného zpracování, např. ve výrobním programu	
\$R:	Identifikátor při použití jako proměnná hlavního zpracování, např. v synchronních akcích	
	Typ:	REAL
	Rozsah hodnot:	V případě neexponenciálního způsobu zápisu: $\pm (0.000\ 0001 \dots 9999\ 9999)$ Upozornění: Povoleno je maximálně 8 desetinných míst.
		V případě exponenciálního způsobu zápisu: $\pm (1*10^{-300} \dots 1*10^{+300})$ Upozornění: • Způsob zápisu: $\langle \text{mantisa} \rangle \text{EX} \langle \text{exponent} \rangle$ např. 8.2EX-3 • Je povoleno maximálně 10 znaků včetně znaménka a desetinné tečky.

<n>:	Číslo R-Parametru	
	Typ:	INT
	Rozsah hodnot:	0 - MAX_INDEX Upozornění: MAX_INDEX vyplývá z počtu R-parametrů, který je definován parametrem: MAX_INDEX = (MD28050 \$MN_MM_NUM_R_PARAM) - 1
<výraz>:	Index pole Jako index pole může být uveden libovolný výraz, pokud ale výsledek tohoto výrazu může být převeden na datový typ INT (INT, REAL, BOOL, CHAR).	

Příklad:

Přiřazení R-parametru a použití R-parametrů v matematických funkcích:

Programový kód	Komentář
R0=3.5678	; přiřazení v předběžném zpracování
R[1]=-37.3	; přiřazení v předběžném zpracování
R3=-7	; přiřazení v předběžném zpracování
\$R4=-0.1EX-5	; přiřazení v hlavním zpracování: $R4 = -0.1 * 10^{-5}$
\$R[6]=1.874EX8	; přiřazení v hlavním zpracování: $R6 = 1.874 * 10^8$
R7=SIN(25.3)	; přiřazení v předběžném zpracování
R[R2]=R10	; nepřímé adresování pomocí R-parametru
R[(R1+R2)*R3]=5	; nepřímé adresování pomocí matematického výrazu
X=(R1+R2)	; najíždění osou X na pozici, která vyplývá se součtu R1 a R2
Z=SQRT(R1*R1+R2*R2)	; najíždění osou Z na pozici odpovídající druhé odmocnině z výrazu $(R1^2 + R2^2)$

Viz také

Proměnné (Strana 21)

2.1.2.2 Globální početní parametry (RG)**Funkce**

Kromě kanálových R-parametrů jsou uživateli k dispozici také globální R-parametry. V rámci řídicího systému existují jen jednou a je možno je číst a zapisovat do nich ve všech kanálech.

Globální R-parametry se používají např. pro přenášení informací z jednoho kanálu do druhého. Dalším příkladem mohou být globální nastavení, která mají být vyhodnocována ve všech kanálech, jako je např. vzdálenost o kterou surový obrobek vyčnívá z vřetena.

Čtení a zapisování globálních R-parametrů se uskutečňuje prostřednictvím uživatelského rozhraní nebo při zpracovávání NC-programu. Použití v synchronních akcích nebo v technologických cyklech není možné.

Poznámka

Při čtení a zapisování globálních R-parametrů se mezi kanály neuskutečňuje **žádná** synchronizace.

Protože se čtení a zapisování provádí v rámci předběžného zpracování, okamžik, ve kterém je hodnota zapisovaná v jednom kanálu v platnosti ve druhém kanálu, není definován.

Příklad:

V kanálu 1 je zpracovávána programová smyčka, ve které se R-parametr používá jako počítadlo průchodů. V kanálu 2 se zapisuje do tohoto globálního R-parametru hodnota, která způsobuje přerušení zpracování smyčky v kanálu 1. Všechny smyčky, které jsou až do tohoto okamžiku interpretovány v rámci předběžného zpracování v kanálu 1, jsou však ještě zpracovávány. Kolik ale těchto smyček je, není definováno a závisí to mimo jiné na vytížení daného kanálu.

Synchronizaci mezi kanály musí tedy uživatel realizovat sám, např. pomocí značek WAIT!

Syntaxe

Zápis v NC programu

RG[<n>]=<hodnota>

RG[<výraz>]=<hodnota>

Čtení v NC programu

R...=RG[<n>]

R...=RG[<výraz>]

Význam

RG:	Předdefinovaný název adresy NC systému pro globální R-parametry Poznámka: Název této NC-adresy může být nastaven pomocí strojního parametru MD15800 \$MN_R_PARAM_NCK_NAME.	
<n>:	Číslo globálního R-parametru	
	Typ:	INT
	Rozsah hodnot:	0 ... MAX_INDEX Poznámka MAX_INDEX vyplývá z počtu globálních R-parametrů, který je definován parametrem: MAX_INDEX = (MD18156 \$MN_MM_NUM_R_PARAM_NCK) - 1
<výraz>:	Jako index pole může být uveden libovolný výraz, pokud ale výsledek tohoto výrazu může být převeden na datový typ INT (INT, REAL, BOOL, CHAR).	

<hodnota>:	Hodnota globálního R-parametru	
	Typ:	REAL
	Rozsah hodnot:	V případě neexponenciálního způsobu zápisu: $\pm (0.000\ 0001 \dots 9999\ 9999)$ Poznámka: Povoleno je maximálně 8 desetinných míst. V případě exponenciálního způsobu zápisu: $\pm (1*10^{-300} \dots 1*10^{+300})$ Poznámka: • Způsob zápisu: <mantisa>EX<exponent> např. 8.2EX-3 • Je povoleno maximálně 10 znaků včetně znaménka a desetinné tečky.

2.1.3 Předdefinované uživatelské proměnné: linkové proměnné

Prostřednictvím linkových proměnných mohou být v rámci funkce "NCU-Link" cyklicky vyměňována data mezi jednotkami NCU, které jsou spolu spojeny pomocí sítě. Tyto jednotky přitom umožňují přístup do paměti linkových proměnných v závislosti na formátu těchto dat. Velikost a také datová struktura paměti linkových proměnných je definována uživatelem / výrobcem zařízení tak, aby vyhovovala danému zařízení.

Linkové proměnné jsou globální systémové uživatelské proměnné, jejichž čtení a zápis je možné v případě linkové komunikace nastavené v konfiguraci provádět všemi jednotkami NCU linkového svazku pomocí výrobních programů a cyklů. Oproti globálním uživatelským proměnným (GUD) se mohou linkové proměnné používat také v synchronních akcích.

U zařízení bez aktivního modulu NCU-Link mohou být linkové proměnné používány lokálně v řídicím systému vedle globálních uživatelských proměnných (GUD) jako další globální uživatelské proměnné.

Syntaxe

```
$A_DLB [<Index>]
$A_DLW [<Index>]
$A_DLD [<Index>]
$A_DLR [<Index>]
```

Význam

\$A_DLB:	Linková proměnná pro datový formát BYTE (1 byte)	
	Datový typ:	UINT
	Rozsah hodnot:	0 ... 255
\$A_DLW:	Linková proměnná pro datový formát WORD (2 byty)	
	Datový typ:	INT
	Rozsah hodnot:	-32768 ... 32767

2.1 Proměnné

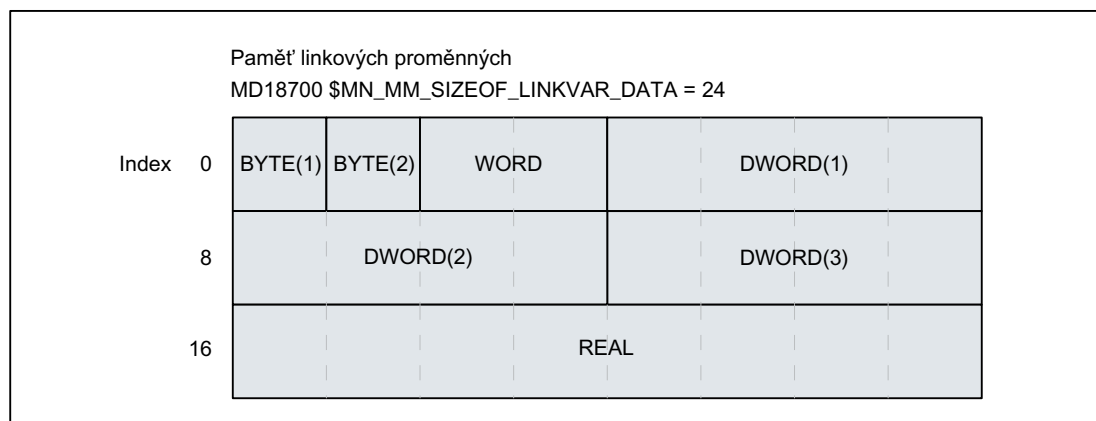
\$A_DLD:	Linková proměnná pro datový formát DWORD (4 byty)	
	Datový typ:	INT
	Rozsah hodnot:	-2147483648 ... 2147483647
\$A_DLR:	Linková proměnná pro datový formát REAL (8 bytů)	
	Datový typ:	REAL
	Rozsah hodnot:	$\pm(2,2 \cdot 10^{-308} \dots 1,8 \cdot 10^{308})$
<Index>:	Index adresy v bytech, vypočítaný od začátku paměti linkových proměnných	
	Datový typ:	INT
	Rozsah hodnot:	0 - MAX_INDEX Upozornění - MAX_INDEX vyplývá z velikosti paměti linkových proměnných, která je definována parametrem: $\text{MAX_INDEX} = (\text{MD18700 } \\$\text{MN_MM_SIZEOF_LINKVAR_DATA}) - 1$ - V programech se smí používat jedině indexy, takže byty adresované v paměti linkových proměnných leží na hranici datového formátu => Index = $n \cdot \text{byty}$, kde $n = 0, 1, 2, \dots$ \$A_DLB[i]: $i = 0, 1, 2, \dots$ \$A_DLW[i]: $i = 0, 2, 4, \dots$ \$A_DLD[i]: $i = 0, 4, 8, \dots$ \$A_DLR[i]: $i = 0, 8, 16, \dots$

Příklad:

V automatickém zařízení jsou 2 jednotky NCU (NCU1 a NCU2). K NCU1 je připojena osa stroje AX2, jejíž pohyb má být ovládán z NCU2 mechanismem linkové osy.

NCU1 cyklicky zapisuje skutečnou hodnotu proudu (\$VA_CURR) osy AX2 do paměti linkových proměnných. NCU2 cyklicky čte skutečnou hodnotu proudu přenášenou pomocí linkové komunikace a při překročení mezní hodnoty vypisuje alarm 61000.

Datová struktura v paměti linkových proměnných je uvedena na následujícím obrázku. Skutečná hodnota proudu se přenáší pomocí hodnoty typu REAL.



NCU1

NCU1 cyklicky v taktu IPO zapisuje v rámci statické synchronní akce skutečnou hodnotu proudu osy AX2 prostřednictvím linkové proměnné \$A_DLR[16] do paměti linkových proměnných.

Programový kód

N111 IDS=1 WHENEVER TRUE DO \$A_DLR[16]=\$VA_CURR[AX2]
--

NCU2

NCU2 cyklicky v taktu IPO čte v rámci statické synchronní akce skutečnou hodnotu proudu osy AX2 prostřednictvím linkové proměnné \$A_DLR[16] z paměti linkových proměnných. Pokud je skutečná hodnota proudu větší než 23,0 A, aktivuje se alarm 61000.

Programový kód

N222 IDS=1 WHEN \$A_DLR[16] > 23.0 DO SETAL(61000)
--

Viz také

Proměnné (Strana 21)

2.1.4 Definice uživatelských proměnných (DEF)

Pomocí příkazu `DEF` mohou být specifické uživatelské proměnné nebo uživatelské proměnné (User Data) definovány a je možno jim dosadit hodnoty.

V závislosti na oblasti jejich platnosti, tzn. oblasti, ve které se proměnná zobrazuje, existují následující kategorie uživatelských proměnných:

- **Lokální uživatelské proměnné (LUD)**
Lokální uživatelské proměnné (LUD) jsou proměnné, které jsou definovány v NC programu, který v okamžiku zpracovávání není hlavním programem. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídicího systému. Přístup k LUD je možný pouze v rámci NC programu, v němž jsou definovány.
- **Programově globální uživatelské proměnné (PUD)**
Programově globální uživatelské proměnné (PUD) jsou proměnné, které jsou definovány v NC programu, který je používán jako hlavní program. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídicího systému. Přístup k PUD je k dispozici v hlavním programu a ve všech podprogramech.

Poznámka

Dostupnost programových globálních uživatelských proměnných (PUD)

Programově globální uživatelské proměnné (PUD) definované v hlavním programu jsou k dispozici i v podprogramech jen tehdy, pokud je nastaven následující strojní parametr:

MD11120 \$MN_LUD_EXTENDED_SCOPE = 1

Pokud je MD11120 = 0, jsou programově globální uživatelské proměnné definované v hlavním programu k dispozici pouze v tomto hlavním programu.

- **Globální uživatelské proměnné (GUD)**
Globální uživatelské proměnné (GUD) jsou v globálními proměnnými v NC systému, příp. v kanálu, které jsou definovány v datovém modulu (SGUD, MGUD, UGUD, GUD4 ... GUD9) a které zůstávají zachovány i po resetu na konci programu, příp. po následujícím spuštění řídicího systému. Přístup ke GUD je k dispozici ve všech NC programech.

Uživatelské proměnné musí být před svým použitím (čtení / zápis) definovány. Přitom je potřeba dbát na následující pravidla:

- GUD musí být definována v definičním souboru, např. _N_DEF_DIR/_N_UGUD_DEF.
- PUD a LUD musí být definovány v definiční části NC programu.
- Definice dat se musí provádět v samostatném bloku.
- V každé definici dat smí být použit jen jeden datový typ.
- V každé definici dat může být definován větší počet proměnných stejného datového typu.

Syntaxe

LUD a PUD

```
DEF <typ> <fyzikální jednotka> <mezí hodnota> <název>[<hodnota_1>,
<hodnota_2>, <hodnota_3>]=<počáteční_hodnota>
```

GUD

```
DEF <oblast> <VL_Stop> <přístupová oprávnění> <datová třída> <typ>
<fyzikální jednotka> <mezí hodnota> <název>[<hodnota_1>,
<hodnota_2>, <hodnota_3>]=<počáteční_hodnota>
```

Význam

DEF:	Příkaz pro definici uživatelských proměnných GUD, PUD, LUD	
<oblast>:	Oblast platnosti, týká se pouze GUD:	
	NC:	Uživatelská proměnná globální v NC systému
	CHAN:	Uživatelská proměnná globální v kanálu
<VL_Stop>:	Zastavení předběžného zpracování, týká se pouze GUD (není nutné)	
	SYNR:	Zastavení předběžného zpracování při čtení
	SYNW:	Zastavení předběžného zpracování při zápisu
	SYNRW:	Zastavení předběžného zpracování při čtení/zápisu
<přístupová oprávnění>:	Úroveň ochrany pro čtení / zápis GUD prostřednictvím NC programu nebo BTSS (není nutné)	
	APRP <úroveň ochrany>:	Čtení: NC program
	APWP <úroveň ochrany>:	Zápis: NC program
	APRB <úroveň ochrany>:	Čtení: BTSS
	APWB <úroveň ochrany>:	Zápis: BTSS
	<úroveň ochrany>:	Rozsah hodnot: 0 ... 7
	Viz "Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB) (Strana 45)".	
<Datová třída>:	Přiřazení datové třídy (pouze SINUMERIK 828D!)	
	DCM:	Datová třída M (= Manufacturer (výrobce))
	DCI:	Datová třída I (= Individual (individuální))
	DCU:	Datová třída U (= User (Uživatel))
	Viz "Atribut: Datová třída (DCM, DCI, DCU) - jen v systému SINUMERIK 828D (Strana 50)".	
<typ>:	Datový typ:	
	INT:	Celočíselné hodnoty se znaménkem
	REAL :	Reálné číslo (LONG REAL podle IEEE)
	BOOL:	Boolovská hodnota TRUE (1) / FALSE (0)
	CHAR:	Znaky ASCII
	STRING[<Max. délka>]:	Řetězec znaků definované délky
	AXIS:	Identifikátor osy/vřetena
	FRAME:	Geometrické údaje pro statickou transformaci souřadného systému
	Viz "Datové typy (Strana 58)".	
<fyzikální jednotka>:	Fyzikální jednotka (není nutné)	
	PHU <jednotka>:	fyzikální jednotka
	Viz "Atribut: Fyzikální jednotka (PHU) (Strana 43)".	
<mezní hodnota>:	horní / dolní mezní hodnota (není nutné)	
	LLI <mezní hodnota>:	spodní mezní hodnota (lower limit)
	ULI <mezní hodnota>:	horní mezní hodnota (upper limit)
	Viz "Atribut: Mezní hodnoty (LLI, ULI) (Strana 41)".	

2.1 Proměnné

<název>:	Název proměnné Upozornění - Maximálně 31 znaků - První dva znaky musí být písmena a/nebo znak podtržení. - Znak "\$" je vyhrazen pro systémové proměnné a nesmí se používat.
[<hodnota_1>, <hodnota_2>, <hodnota_3>]:	Údaje o velikosti pole pro 1- až 3-rozměrné proměnné typu pole (není nutné) Pokud budete potřebovat informace o inicializaci proměnných typu pole, viz "Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 52)".
<počáteční hodnota>:	Inicializační hodnota (není nutné) viz "Atribut: Inicializační hodnota (Strana 38)". Pokud budete potřebovat informace o inicializaci proměnných typu pole, viz "Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 52)".

Příklady

Příklad 1: Definice uživatelských proměnných v datovém modulu pro výrobce stroje

Programový kód	Komentář
%_N_MGUD_DEF	; Modul GUD: Výrobce stroje
\$PATH=/_N_DEF_DIR	
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2	
; popis	
; definice dvou GUD: STROM_1, STROM_2	
; oblast platnosti: v kanálu	
; datový typ: REAL	
; zastavení předběžného zpracování: není naprogramováno => předdefinovaná hodnota = žádné zastavení předběžného zpracování (VL-Stop)	
; fyzikální jednotka: 24 = [A]	
; mezní hodnota: Low = 0.0, High = 10.0	
; přístupová oprávnění: není naprogramováno --> předdefinovaná hodnota = 7 = poloha přepínače na klíč 0	
; inicializační hodnota: není naprogramováno --> předdefinovaná hodnota = 0.0	
DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45	
; popis	
; definice dvou GUD: ZEIT_1, ZEIT_2	
; oblast platnosti: V celém NC systému	
; datový typ: REAL	
; zastavení předběžného zpracování: není naprogramováno => předdefinovaná hodnota = žádné zastavení předběžného zpracování (VL-Stop)	
; fyzikální jednotka: 13 = [s]	
; mezní hodnota: Low = 10.0, High = není naprogramováno => horní hranice definiční oblasti	
; přístupová oprávnění:	

Programový kód	Komentář
<pre> ; NC program: Čtení/zápis = 3 = koncový uživatel ; BTSS: Zápis = 0 = Siemens, Čtení = 3 = koncový uživatel ; inicializační hodnota: ZEIT_1 = 12.0, ZEIT_2 = 45.0 DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER" ; popis ; definice jednoho GUD: GUD5_NAME ; oblast platnosti: V celém NC systému ; datový typ: STRING, max. 5 znaků ; zastavení předběžného zpracování: není naprogramováno => předdefinovaná hodnota = žádné zastavení předběžného zpracování (VL-Stop) ; fyzikální jednotka: není naprogramováno --> předdefinovaná hodnota = 0 = žádná fyzikální jednotka ; mezní hodnota: není naprogramováno => hranice definiční oblasti: Low = 0, High = 255 ; přístupová oprávnění: ; NC program: Čtení/zápis = 3 = koncový uživatel ; BTSS: Zápis = 0 = Siemens, Čtení = 3 = koncový uživatel ; inicializační hodnota: "COUNTER" M30 </pre>	

Příklad 2: V programu globální a lokální uživatelské proměnné (PUD / LUD)

Programový kód	Komentář
PROC MAIN	; hlavní program:
DEF INT VAR1	; definice PUD
...	
SUB2	; volání podprogramu
...	
M30	

Programový kód	Komentář
PROC SUB2	; podprogram SUB2
DEF INT VAR2	; DEFINICE LUD
...	
IF (VAR1==1)	; čtení PUD
VAR1=VAR1+1	; čtení a zápis PUD
VAR2=1	; zápis LUD
ENDIF	
SUB3	; volání podprogramu
...	
M17	

Programový kód	Komentář
PROC SUB3	; podprogram SUB3
...	

Programový kód	Komentář
IF (VAR1==1)	; čtení PUD
VAR1=VAR1+1	; čtení a zápis PUD
VAR2=1	; chyba: LUD ze SUB2 není známa
ENDIF	
...	
M17	

Příklad 3: Definice a použití uživatelských proměnných datového typu AXIS

Programový kód	Komentář
DEF AXIS ABSZISSE	; 1. Geometrická osa
DEF AXIS SPINDLE	; vřeteno
...	
IF ISAXIS(1) == FALSE GOTO WEITER	
ABSZISSE = \$P_AXN1	
WEITER:	
...	
SPINDLE=(S1)	; 1. Vřeteno
OVRA[SPINDLE]=80	; korekce vřetena = 80%
SPINDLE=(S3)	; 3. Vřeteno

Okrajové podmínky**Globální uživatelské proměnné (GUD)**

V rámci definice globálních uživatelských proměnných (GUD) je potřeba mít na paměti následující strojní parametry:

Č.	identifikátoru: \$MN_	Význam
11140	GUD_AREA_SAVE_TAB	Doplňkové zálohování modulů GUD
18118 ¹⁾	MM_NUM_GUD_MODULES	Počet souborů GUD v aktivním systému souborů
18120 ¹⁾	MM_NUM_GUD_NAMES_NCK	Počet názvů globálních GUD
18130 ¹⁾	MM_NUM_GUD_NAMES_CHAN	Počet názvů kanálových GUD
18140 ¹⁾	MM_NUM_GUD_NAMES_AXIS	Počet názvů osových GUD
18150 ¹⁾	MM_GUD_VALUES_MEM	Paměťový prostor pro hodnoty globálních GUD
18660 ¹⁾	MM_NUM_SYNACT_GUD_REAL	Počet GUD typu REAL, které mohou být nastaveny v konfiguraci
18661 ¹⁾	MM_NUM_SYNACT_GUD_INT	Počet GUD typu INT, které mohou být nastaveny v konfiguraci
18662 ¹⁾	MM_NUM_SYNACT_GUD_BOOL	Počet GUD typu BOOL, které mohou být nastaveny v konfiguraci
18663 ¹⁾	MM_NUM_SYNACT_GUD_AXIS	Počet GUD typu AXIS, které mohou být nastaveny v konfiguraci

Č.	identifikátoru: \$MN_	Význam
18664 ¹⁾	MM_NUM_SYNACT_GUD_CHAR	Počet GUD typu CHAR, které mohou být nastaveny v konfiguraci
18665 ¹⁾	MM_NUM_SYNACT_GUD_STRING	Počet GUD typu STRING, které mohou být nastaveny v konfiguraci

¹⁾ V systému SINUMERIK 828D je tento MD jen ke čtení!

Použití uživatelských proměnných datového typu AXIS globálních v NC systému přes hranice kanálu

V NC globální uživatelské proměnné datového typu **AXIS**, které byly při definici datového modulu inicializovány identifikátorem osy, mohou být používány v různých kanálech NC systému jedině tehdy, pokud má osa v těchto kanálech stejné číslo kanálové osy.

Jestliže tomu tak není, musí být proměnná načtena na začátku programu NC systému, jak ukazuje následující příklad, musí být použita funkce **AXNAME(...)** (viz "Osové funkce (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Strana 649)").

Programový kód	Komentář
DEF NCK STRING[5] ACHSE="X"	; definice v datovém modulu
...	
N100 AX[AXNAME(ACHSE)]=111 G00	; použití v NC programu

2.1.5 Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF)

Pomocí příkazu **REDEF** mohou být změněny atributy systémových dat, uživatelských dat a příkazů NC jazyka. Základním předpokladem pro změnu definice je, že musí být uskutečněna časově až po odpovídající definici.

V rámci opětovné definice není možné měnit větší počet atributů současně. Pro každý atribut, který má být změněn, musí být naprogramován vlastní příkaz **REDEF**.

Jestliže je naprogramováno několik vzájemně si odporujících změn atributu, je v platnosti vždycky ta poslední změna.

Obnovení původních hodnot atributů

Pro atributy pro přístupová oprávnění, které byly změněny příkazem **REDEF**, a okamžik inicializace mohou být obnovena jejich původní předdefinovaná nastavení prostřednictvím opětovného naprogramování příkazu **REDEF**, za nímž následuje název proměnné nebo příkaz NC jazyka.

- Přístupová oprávnění: Úroveň ochrany 7
- Okamžik inicializace: žádná inicializace, příp. zachování aktuálních hodnot

Atributy, jejichž definice může být změněna

Viz "Přehled definovatelných a znovu definovatelných atributů (Strana 51)".

Lokální uživatelské proměnné (PUD / LUD)

Pro lokální uživatelské proměnné (PUD / LUD) se žádné změny definice nesmí provádět.

Syntaxe

```

REDEF <název> <VL_Stop>
REDEF <název> <fyzikální jednotka>
REDEF <název> <mezní hodnota>
REDEF <název> <přístupová oprávnění>
REDEF <název> <okamžik inicializace>
REDEF <název> <okamžik inicializace> <počáteční hodnota>
REDEF <název> <Datová třída>
REDEF <název>

```

Význam

REDEF:	Příkaz pro opětovnou definici určitého atributu, příp. pro obnovení původního nastavení atributu "Přístupová oprávnění" a/ nebo "Okamžik inicializace" pro systémové proměnné, uživatelské proměnné a příkazy NC jazyka	
<název>:	Název již definované proměnné nebo příkazu NC jazyka	
<VL-Stop>:	Zastavení předběžného zpracování	
	SYNR:	Zastavení předběžného zpracování při čtení
	SYNW:	Zastavení předběžného zpracování při zápisu
	SYNRW:	Zastavení předběžného zpracování při čtení/zápisu
<fyzikální jednotka>:	Fyzikální jednotka	
	PHU <jednotka>:	fyzikální jednotka
	Viz "Atribut: Fyzikální jednotka (PHU) (Strana 43)". Poznámka: Změny definice jsou nepřípustné pro následující objekty: • Systémové proměnné • Globální uživatelská data (GUD) následujících datových typů: BOOL, AXIS, STRING, FRAME	
<mezní hodnota>:	horní / dolní mezní hodnota	
	LLI <mezní hodnota>:	spodní mezní hodnota (lower limit)
	ULI <mezní hodnota>:	horní mezní hodnota (upper limit)
	Viz "Atribut: Mezní hodnoty (LLI, ULI) (Strana 41)". Poznámka: Změny definice jsou nepřípustné pro následující objekty: • Systémové proměnné • Globální uživatelská data (GUD) následujících datových typů: BOOL, AXIS, STRING, FRAME	

<přístupová oprávnění>:	Přístupová oprávnění pro čtení / zápis prostřednictvím výrobního programu nebo BTSS	
	APX <úroveň ochrany>:	Spouštění: Prvek NC jazyka
	APRP <úroveň ochrany>:	Čtení: Výrobní program
	APWP <úroveň ochrany>:	Zápis: Výrobní program
	APRB <úroveň ochrany>:	Čtení: BTSS
	APWB <úroveň ochrany>:	Zápis: BTSS
	<úroveň ochrany>:	Rozsah hodnot: 0 ... 7
	Viz "Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB) (Strana 45)".	
<okamžik inicializace>:	Okamžik, ve kterém je proměnná znovu inicializována	
	INIPO:	Power On
	INIRE:	Konec hlavního programu, reset NC systému nebo Power-On
	INICE:	Příkaz NEWCONF nebo konec hlavního programu, reset NC systému nebo Power-On
	PRLOC:	Konec hlavního programu, reset NC systému po lokální změně nebo Power-On
	Viz "Atribut: Inicializační hodnota (Strana 38)".	
<počáteční hodnota>:	Inicializační hodnota	
	Při opětovné definici inicializační hodnoty musí být vždy zadán také časový okamžik inicializace (viz <okamžik inicializace>). Viz "Atribut: Inicializační hodnota (Strana 38)". Pokud budete potřebovat informace o inicializaci proměnných typu pole, viz "Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 52)". Poznámka: Změny definice jsou nepřipustné pro systémové proměnné, kromě nastavovaných parametrů.	
<Datová třída>:	Přiřazení datové třídy (pouze SINUMERIK 828D!)	
	DCM:	Datová třída M (= Manufacturer (výrobce))
	DCI:	Datová třída I (= Individual (individuální))
	DCU:	Datová třída U (= User (Uživatel))
	Viz "Atribut: Datová třída (DCM, DCI, DCU) - jen v systému SINUMERIK 828D (Strana 50)".	

Příklad:**Opětovná definice systémové proměnné \$TC_DPCx v datovém modulu pro výrobce stroje**

Programový kód	
%_N_MGUD_DEF	; Modul GUD: Výrobce stroje
N100 REDEF \$TC_DPC1 APWB 2 APWP 3	
N200 REDEF \$TC_DPC2 PHU 21	
N300 REDEF \$TC_DPC3 LLI 0 ULI 200	
N400 REDEF \$TC_DPC4 INIPO (100, 101, 102, 103)	

Programový kód

```
N800 REDEF $TC_DPC1
N900 REDEF $TC_DPC4
M30
```

k bloku Přístupová oprávnění pro zápis: BTSS = úroveň ochrany 2, výrobní program = úroveň
N100: ochrany 3

k bloku Fyzikální jednotka [%]
N200:

k bloku spodní mezní hodnota = 0, horní mezní hodnota = 200
N300:

k bloku Proměnná typu pole se při zapnutí systému (Power On) inicializuje danými čtyřmi
N400: hodnotami.

k bloku Obnovení původního nastavení atributu "Přístupová oprávnění" a/nebo "Okamžik
N800/ inicializace"
N900:

Poznámka**Použití souborů typu ACCESS**

Jestliže se používají soubory typu ACCESS, musí být opětovná definice přístupových oprávnění přenesena z _N_MGUD_DEF do _N_MACCESS_DEF.

Okrajové podmínky**Granularita**

Opětovná definice se vždy vztahuje na celé, svými názvy jednoznačně určené proměnné. Není tedy možné např. u proměnných typu pole, přiřazovat jednotlivým prvkům pole odlišné hodnoty atributů.

2.1.6 Atribut: Inicializační hodnota**Definice (DEF) uživatelských proměnných**

Při definici je možné pro následující uživatelské proměnné předem zadat inicializační hodnotu:

- globální uživatelské proměnné (GUD)
- programově globální uživatelské proměnné (PUD)
- lokální uživatelské proměnné (LUD)

Opětovná definice (REDEF) systémových a uživatelských proměnných

Při opětovné definici je možné pro následující proměnné předem zadat inicializační hodnotu:

- Systémová data
 - Nastavované parametry
- Uživatelská data
 - R-Parametry
 - Proměnné synchronních akcí (\$AC_MARKER, \$AC_PARAM, \$AC_TIMER)
 - GUD synchronních akcí (SYG_xy[], kde x = R, I, B, A, C, S a y = S, M, U, 4, ..., 9)
 - Parametry EPS
 - Parametry nástroje OEM
 - Parametry zásobníku OEM
 - globální uživatelské proměnné (GUD)

Okamžik opětovné inicializace

Při opětovné definici může být zadán časový okamžik, ve kterém má být proměnná znovu inicializována, tzn. kdy má být znovu dosazena inicializační hodnota:

- INIPO (Power On)
Proměnná se znovu inicializuje při zapnutí systému (Power On).
- INIRE (Reset)
Proměnná bude znovu inicializována při resetu NC systému, resetu BAG, na konci výrobního programu (M02 / M30) nebo při zapnutí systému (Power On).
- INICF (NEWCONF)
Proměnná bude znovu inicializována při aktivování funkce „Aktivování strojních parametrů“ prostřednictvím HMI, při zpracování příkazu NEWCONF ve výrobním programu, při resetu NC systému, resetu BAG, na konci výrobního programu (M02 / M30) nebo při zapnutí systému (Power On).
- PRLOC (programově lokální změna)
Proměnná se při resetu NC systému, resetu BAG nebo na konci výrobního programu (M02 / M30) znovu inicializuje jen tehdy, jestliže byla v rámci aktuálního výrobního programu změněna.
Atribut PRLOC se smí používat jedině v souvislosti s programovatelnými nastavovanými parametry (viz následující tabulka).

Tabulka 2-1 Programovatelný nastavovaný parametr

Číslo	Identifikátor	G-příkaz ¹⁾
42000	\$SC_THREAD_START_ANGLE	SF
42010	\$SC_THREAD_RAMP_DISP	DITS / DITE
42400	\$SA_PUNCH_DWELLTIME	PDELAYON
42800	\$SA_SPIND_ASSIGN_TAB	SETMS
43210	\$SA_SPIND_MIN_VELO_G25	G25
43220	\$SA_SPIND_MAX_VELO_G26	G26
43230	\$SA_SPIND_MAX_VELO_LIMS	LIMS

Číslo	Identifikátor	G-příkaz ¹⁾
43300	\$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
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
43790	\$SA_OSCILL_START_POS	OSB
1) pomocí tohoto G-příkazu je nastavovaný parametr vyžádán		

Okrajové podmínky

Inicializační hodnota: globální uživatelské proměnné (GUD)

- Pro globální uživatelské proměnné (GUD), jejichž oblastí platnosti je NC, je možné jako okamžik inicializace zadat jediné INIPO (Power On).
- Pro globální uživatelské proměnné (GUD), jejichž oblastí platnosti je CHAN, je možné jako okamžik inicializace zadat vedle INIPO (Power On) také INIRE (Reset) nebo INICF (NEWCONF).
- U globálních uživatelských proměnných (GUD), které mají oblast platnosti CHAN a okamžik inicializace INIRE (Reset) nebo INICF (NEWCONF), jsou v případě resetu NC systému, resetu BAG a příkazu „Aktivování strojních parametrů“ proměnné nově inicializovány pouze v těch kanálech, v nichž byla výše uvedená událost uskutečněna.

Inicializační hodnota: Datový typ FRAME

Pro proměnné datového typu FRAME nesmí být žádná inicializační hodnota zadávána. Proměnné datového typu FRAME jsou vždy implicitně inicializovány předdefinovaným framem.

Inicializační hodnota: Datový typ CHAR

Pro proměnné datového typu CHAR může být namísto ASCII kódu (0...255) naprogramován také odpovídající znak ASCII v uvozovkách, např. "A".

Inicializační hodnota: Datový typ STRING

U proměnných datového typu STRING musí být dosazen řetězec znaků v uvozovkách, např.: ...= "MACHINE_1".

Inicializační hodnota: Datový typ AXIS

Pro proměnné datového typu `AXIS` musí být v případě rozšířeného způsobu zápisu adresy dosazen identifikátor osy v závorkách, např.: `...= (X3)`.

Inicializační hodnota: Systémová proměnná

Pro systémové proměnné nemohou být prostřednictvím opětovné definice zadávány žádné specifické uživatelské inicializační hodnoty. Inicializační hodnoty systémových proměnných jsou pevně definovány v systému. Opětovnou definicí může být ale změněn okamžik (`INIRE`, `INICF`), kdy má být systémová proměnná znovu inicializována.

Implicitní inicializační hodnota: Datový typ AXIS

Pro proměnné datového typu `AXIS` se používají následující implicitní inicializační hodnoty:

- Systémová data: "první geometrická osa"
- GUD synchronních akcí (označení: `SYG_A*`), `PUD`, `LUD`:
Identifikátor osy ze strojního parametru: `MD20082`
`$MC_AXCONF_CHANAX_DEFAULT_NAME`

Implicitní inicializační hodnota: Data nástrojů a zásobníků

Pro data nástrojů a zásobníků mohou být inicializační hodnoty předem definovány prostřednictvím následujícího strojního parametru: `MD17520`
`$MN_TOOL_DEFAULT_DATA_MASK`

Poznámka**Synchronizace**

Za synchronizaci událostí, které spouštějí opětovnou inicializaci globální proměnné, se čtením této proměnné na jiném místě nese veškerou odpovědnost uživatel / výrobce stroje.

Viz také

Proměnné (Strana 21)

2.1.7 Atribut: Mezní hodnoty (LLI, ULI)

Horní a dolní mezní hodnotu definiční oblasti je možné zadat jen pro následující datové typy.

- `INT`
- `REAL`
- `CHAR`

Definice (DEF) uživatelských proměnných: Mezní hodnoty a implicitní inicializační hodnoty

Jestliže není při definici uživatelských proměnných jednoho z výše uvedených datových typů definována žádná explicitní inicializační hodnota, bude proměnné dosazena implicitní inicializační hodnota datového typu:

- INT: 0
- REAL : 0.0
- CHAR: 0

Jestliže implicitní inicializační hodnota leží mimo definiční oblast stanovenou naprogramovanými mezními hodnotami, bude proměnná inicializována tou mezní hodnotou, která leží blíže k implicitní inicializační hodnotě:

- implicitní inicializační hodnota < spodní mezní hodnota (LLI) =>
Inicializační hodnota = spodní mezní hodnota
- implicitní inicializační hodnota > horní mezní hodnota (ULI) =>
Inicializační hodnota = horní mezní hodnota

Příklady:

Programový kód	Komentář
DEF REAL GUD1	; spodní mezní hodnota = hranice definiční oblasti ; horní mezní hodnota = hranice definiční oblasti ; žádná inicializační hodnota není naprogramována ; --> implicitní inicializační hodnota = 0.0
DEF REAL LLI 5.0 GUD2	; spodní mezní hodnota = 5,0 ; horní mezní hodnota = hranice definiční oblasti ; => inicializační hodnota = 5,0
DEF REAL ULI -5 GUD3	; spodní mezní hodnota = hranice definiční oblasti ; horní mezní hodnota = -5,0 ; => inicializační hodnota = -5,0

Opětovná definice (REDEF) uživatelských proměnných: mezní hodnoty a aktuální skutečné hodnoty

Jestliže jsou při opětovné definici mezních hodnot uživatelské proměnné tyto mezní hodnoty změněny tak, že aktuální skutečná hodnota leží mimo novou definiční oblast, aktivuje se alarm a mezní hodnoty se nepřevzímou.

Poznámka**Opětovná definice (REDEF) uživatelských proměnných**

Při opětovné definici mezních hodnot uživatelské proměnné je potřeba dávat pozor, aby změny následujících hodnot byly konzistentní:

- Mezní hodnoty
- Skutečná hodnota
- Inicializační hodnota při opětovné definici a při automatické opětovné inicializaci na základě INIPO, INIRE nebo INICF

Viz také

Proměnné (Strana 21)

2.1.8 Atribut: Fyzikální jednotka (PHU)

Fyzikální jednotka může být zadána pouze pro proměnné následujících datových typů:

- INT
- REAL

Programovatelná fyzikální jednotka (PHU)

Fyzikální jednotka se zadává jako číslo s pevnou řádovou čárkou: PHU <jednotka>

Mohou být naprogramovány následující fyzikální jednotky:

<Jednotka>	Význam	Fyzikální jednotka
0	Žádná fyzikální jednotka	-
1	Lineární nebo úhlová poloha ¹⁾²⁾	[mm], [palce], [stupně]
2	Lineární poloha ²⁾	[mm], [palce]
3	Úhlová poloha	[stupně]
4	Lineární nebo úhlová rychlost ¹⁾²⁾	[mm/min], [palce/min], [ot/min]
5	Lineární rychlost ²⁾	[mm/min]
6	Úhlová rychlost	[ot/min]
7	Lineární nebo úhlové zrychlení ¹⁾²⁾	[m/s ²], [palce/s ²], [ot/s ²]
8	Lineární zrychlení ²⁾	[m/s ²], [palce/s ²]
9	Úhlové zrychlení	[ot/s ²]
10	Lineární nebo úhlový ryv ¹⁾²⁾	[m/s ³], [palce/s ³], [ot/s ³]
11	Lineární ryv ²⁾	[m/s ³], [palce/s ³]
12	Úhlový ryv	[ot/s ³]
13	Čas	[s]
14	Zesílení regulátoru polohy	[16.667/s]
15	Otáčkový posuv ²⁾	[mm/ot], [palce/ot]
16	Teplotní kompenzace ¹⁾²⁾	[mm], [palce]
18	Síla	[N]
19	Hmotnost	[kg]
20	Moment setrvačnosti ³⁾	[kgm ²]
21	Procenta	[%]
22	Frekvence	[Hz]
23	Napětí	[V]
24	Proud	[A]
25	Teplota	[°C]
26	Úhel	[stupně]
27	KV	[1000/min]

<Jednotka>	Význam	Fyzikální jednotka
28	Lineární nebo úhlová poloha ³⁾	[mm], [palce], [stupně]
29	Řezná rychlost ²⁾	[m/min], [stopy/min]
30	Obvodová rychlost ²⁾	[m/s], [stopy/s]
31	Odpor	[ohm]
32	indukčnost	[mH]
33	Točivý moment ³⁾	[Nm]
34	Konstanta točivého momentu ³⁾	[Nm/A]
35	Zesílení proudového regulátoru	[V/A]
36	Zesílení regulátoru otáček ³⁾	[Nm/(rad*s)]
37	Otáčky	[ot/min]
42	Výkon	[kW]
43	Proud, malý	[µA]
46	Točivý moment, malý ³⁾	[µNm]
48	Promile	-
49	-	[Hz/s]
65	Průtok	[l/min]
66	Tlak	[bar]
67	Objem ³⁾	[cm ³]
68	Dráhové zesílení ³⁾	[mm/(V*min)]
69	Dráhové zesílení regulátoru síly	[N/V]
155	Stoupání závitu ³⁾	[mm/ot], [palce/ot]
156	Změna stoupání závitu ³⁾	[mm/ot / ot], [palce/ot / ot]

1) Fyzikální jednotka závisí na typu osy: lineární nebo kruhová osa

2) Přepínání systému jednotek

G70/G71(palce/metrické jednotky)

Po přepnutí základního systému (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) pomocí příkazu G70/G71 se v případě přístupu k systémovým a uživatelským proměnným souvisejícím s délkou za účelem čtení/zápisu neuskutečňuje **žádné** přepočítávání hodnot (skutečná hodnota, předdefinovaná hodnota a mezní hodnoty).

G700/G710(palce/metrické jednotky)

Po přepnutí základního systému (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) pomocí příkazu G700/G710 se v případě přístupu k systémovým a uživatelským proměnným souvisejícím s délkou za účelem čtení/zápisu **uskutečňuje** přepočítávání hodnot (skutečná hodnota, předdefinovaná hodnota a mezní hodnoty).

3) Proměnná **není** automaticky přepočítávána do momentálně nastaveného systému jednotek NC systému (palce/metrické jednotky). Za přepočítání nese plnou odpovědnost uživatel / výrobce stroje.

Poznámka

Přeběh roviny v důsledku přepočítání formátu

Interně je pro ukládání všech uživatelských proměnných (GUD / PUD / LUD) s fyzikálními jednotkami vztahujícími se k délce používán metrický formát. Nadměrné používání proměnných tohoto druhu v hlavní větvi NCK, např. v synchronních akcích, může mít při přepnutí systému měřicích jednotek za následek přetečení výpočetního času roviny interpolátoru, alarm 4240.

Poznámka**Kompatibilita jednotek**

Při používání proměnných (přiřazení, porovnávání, výpočty atd.) se neprovádí žádná kontrola, zda jsou jednotky podílející se na operaci kompatibilní. Za přepočítání, které je eventuálně nezbytné, nese plnou odpovědnost uživatel / výrobce stroje.

Viz také

Proměnné (Strana 21)

2.1.9 Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB)**Označení**

Označení přístupového atributu **AP** se skládá z těchto částí:

1. **A**: Access (přístup)
2. **P**: Protection (ochrana)
3. **R / W**: Read (čtení) / Write (zápis)
4. **P / B**: Program / BTSS

Přístupová oprávnění / úrovně ochrany

Přístupová oprávnění odpovídají následujícím úrovním ochrany, které se zadávají při programování:

Přístupová oprávnění	Úroveň ochrany
Heslo systému	0
Heslo výrobce stroje	1
Heslo servisní služby	2
Heslo koncového uživatele	3
3. poloha přepínače na klíč	4
2. poloha přepínače na klíč	5
1. poloha přepínače na klíč	6
0. poloha přepínače na klíč	7

Definice (DEF) uživatelských dat

Přístupová oprávnění (APR... / APW...) mohou být definována pro následující data:

- Globální uživatelská data (GUD)

Opětovná definice (REDEF) systémových a uživatelských dat

Přístupová oprávnění (APR... / APW...) mohou být opětovně definována pro následující data:

- Systémová data
 - Strojní parametry

Poznámka

Změna definice oprávnění ke čtení strojních parametrů

Úroveň ochrany pro čtení strojních parametrů může být nastavována pomocí klíčového slova `APR` pouze společně pro výrobní program a BTSS.

Klíčová slova `APRP` a `APRB` nejsou pro změnu definice oprávnění ke čtení podporována a mají za následek alarm 12490 s hlášením „Přístupová oprávnění APRP/APRB <úroveň ochrany> nebyla nastavena“.

- Nastavované parametry
- Systémové proměnné
- Data procesů
- Data zásobníku
- Parametry nástroje
- Uživatelská data
 - R-Parametry
 - Proměnné synchronních akcí (`$AC_MARKER`, `$AC_PARAM`, `$AC_TIMER`)
 - GUD synchronních akcí (`SYG_xy[ ]`, kde `x` = R, I, B, A, C, S a `y` = S, M, U, 4, ..., 9)
 - Parametry EPS
 - Parametry nástroje OEM
 - Parametry zásobníku OEM
 - globální uživatelské proměnné (GUD)

Poznámka

Při opětovné definici mohou být přístupová oprávnění k určité proměnné volně nastavitelná v rozmezí mezi nejnižším stupněm ochrany 7 až po vlastní stupeň ochrany, např. 1 (výrobce stroje).

Opětovná definice (REDEF) příkazů NC jazyka

Oprávnění k přístupu, příp. ke spouštění (APX) mohou být opětovně definována pro následující příkazy NC jazyka:

- Příkazy G-funkce / podmínky dráhy
Literatura:
Programovací příručka, Základy; kapitola: Příkazy G-funkce / podmínky dráhy
- Předem definované funkce
Literatura:
Programovací příručka, Základy; kapitola: Předem definované funkce
- Vyvolávání předem definovaných podprogramů
Literatura:
Programovací příručka, Základy; kapitola: Vyvolávání předem definovaných podprogramů
- Příkaz DO u synchronních akcí
- Programový identifikátor cyklů
Cyklus musí být uložen v adresáři cyklů a musí obsahovat příkaz PROC.

Přístupová oprávnění týkající se NC programů a cyklů (APRP, APWP)

Jednotlivá přístupová oprávnění mají pro přístup z NC programu, příp. z cyklu, následující účinky:

- APRP 0 / APWP 0
 - Při zpracovávání NC programu musí být zadáno heslo systému.
 - Cyklus musí být uložen v adresáři _N_CST_DIR (systém).
 - Pro adresář _N_CST_DIR musí být ve strojním parametru MD11160 \$MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání na "systém".
- APRP 1 / APWP 1 příp. APRP 2 / APWP 2
 - Při zpracovávání NC programu musí být zadáno heslo výrobce stroje, příp. servisní služby.
 - Cyklus musí být uložen v adresáři _N_CMA_DIR (výrobce stroje) nebo _N_CST_DIR.
 - Pro adresáře _N_CMA_DIR, příp. _N_CST_DIR musí být ve strojních parametrech MD11161 \$MN_ACCESS_EXEC_CMA, příp. MD11160 \$MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání minimálně na "výrobce stroje".

- APRP 3 / APWP 3
 - Při zpracovávání NC programu musí být zadáno heslo koncového uživatele.
 - Cyklus musí být uložen v adresáři _N_CUS_DIR (uživatel), _N_CMA_DIR nebo _N_CST_DIR.
 - Pro adresáře _N_CUS_DIR, _N_CMA_DIR příp. _N_CST_DIR musí být ve strojních parametrech MD11162 \$MN_ACCESS_EXEC_CUS, MD11161 \$MN_ACCESS_EXEC_CMA, příp. MD11160 \$MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání minimálně na "koncového uživatele".
- APRP 4...7 / APWP 4...7
 - Při zpracovávání NC programu musí být přepínač na klíč nastaven v poloze 3 ... 0.
 - Cyklus musí být uložen v adresáři _N_CUS_DIR, _N_CMA_DIR nebo _N_CST_DIR.
 - Pro adresáře _N_CUS_DIR, _N_CMA_DIR příp. _N_CST_DIR musí být ve strojních parametrech MD11162 \$MN_ACCESS_EXEC_CUS, MD11161 \$MN_ACCESS_EXEC_CMA, příp. MD11160 \$MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání minimálně na odpovídající polohu přepínače na klíč.

Přístupová oprávnění týkající se BTSS (APRB, APWB)

Přístupová oprávnění (APRB, APWB) omezují přístup k systémovým a uživatelským proměnným prostřednictvím BTSS stejnou měrou pro všechny systémové komponenty (HMI, PLC, externí počítač, služby EPS atd.).

Poznámka

Lokální přístupová oprávnění HMI

Při změnách přístupových oprávnění k systémovým datům je nutno dávat pozor, aby byly tyto změny v souladu s přístupovými oprávněními definovanými pomocí mechanismů HMI.

Přístupové atributy APR / APW

Z důvodů kompatibility jsou atributy APR a APW implicitně přenášeny do atributů APRP / APRB a APWP / APWB.

- APR x => APRP x APRB x
- APW y => APWP y APWB y

Přístupová oprávnění prostřednictvím souborů typu ACCESS

V případě, že jsou pro předávání přístupových oprávnění používány soubory typu ACCESS, smí být opětovné definice přístupových oprávnění pro systémová data, uživatelská data a příkazy NC jazyka prováděny výlučně v souborech typu ACCESS. Výjimku tvoří globální uživatelská data (GUD). V jejich případě musí být opětovná definice přístupových oprávnění i nadále uskutečňována v odpovídajících definičních souborech *_DEF.

Aby byla přístupová ochrana konzistentní, musí být odpovídajícím způsobem přizpůsobeny strojní parametry pro oprávnění ke spouštění funkcí a pro přístupovou ochranu pro příslušné adresáře.

Z toho vyplývají následující postupy:

1. Vytvořte potřebné definiční soubory:

- _N_DEF_DIR/_N_SACCESS_DEF
- _N_DEF_DIR/_N_MACCESS_DEF
- _N_DEF_DIR/_N_UACCESS_DEF

2. Nastavte parametry oprávnění k zápisu pro tyto definiční soubory na hodnotu, která je zapotřebí pro opětovnou definici:

- MD11170 \$MN_ACCESS_WRITE_SACCESS = <úroveň ochrany>
- MD11171 \$MN_ACCESS_WRITE_MACCESS = <úroveň ochrany>
- MD11172 \$MN_ACCESS_WRITE_UACCESS = <úroveň ochrany>

3. Za účelem přístupu k chráněným prvkům z cyklů je nutné přizpůsobit oprávnění pro spouštění funkcí a pro zápis pro adresáře cyklů _N_CST_DIR, _N_CMA_DIR a _N_CST_DIR.

Oprávnění ke spouštění funkcí

- MD11160 \$MN_ACCESS_EXEC_CST = <úroveň ochrany>
- MD11161 \$MN_ACCESS_EXEC_CMA = <úroveň ochrany>
- MD11162 \$MN_ACCESS_EXEC_CUS = <úroveň ochrany>

Oprávnění k zápisu

- MD11165 \$MN_ACCESS_WRITE_CST = <úroveň ochrany>
- MD11166 \$MN_ACCESS_WRITE_CMA = <úroveň ochrany>
- MD11167 MN_ACCESS_WRITE_CUS = <úroveň ochrany>

Oprávnění pro spouštění funkcí musí být nastaveno minimálně na stejnou úroveň ochrany, jaká je nejvyšší úroveň ochrany používaného prvku.

Oprávnění pro zápis musí být nastaveno minimálně na stejnou úroveň ochrany jako oprávnění pro spouštění funkcí.

4. Oprávnění k zápisu pro lokální adresáře cyklů HMI musí být nastaveno minimálně na stejnou úroveň ochrany, jaká je pro lokální adresáře cyklů v NC systému.

Literatura:

Příručka pro obsluhu

Vyvolávání podprogramů v souborech typu ACCESS

Kvůli dalšímu strukturování ochrany proti přístupu mohou být v souborech typu ACCESS také vyvolávány podprogramy (identifikace SPF nebo MPF). Tyto podprogramy přitom dědí oprávnění ke spouštění funkcí ze souboru typu ACCESS, z něhož byly vyvolány.

Poznámka

V souborech typu ACCESS mohou být znovu definována pouze přístupová oprávnění. Všechny ostatní atributy musí být i nadále naprogramovány, příp. znovu definovány v odpovídajících definičních souborech.

Viz také

Proměnné (Strana 21)

2.1.10 Atribut: Datová třída (DCM, DCI, DCU) - jen v systému SINUMERIK 828D

Aby se zjednodušila manipulace s daty v rámci operací, jako jsou uvádění do provozu, sériové uvádění do provozu a modernizace strojů a typových řad strojů, jsou všechny systémová a uživatelská data rozdělena do datových tříd.

Datová třída	Data
S = Systém	Systémová data připravená firmou Siemens, jako jsou strojní a nastavované parametry, standardní a měřicí cykly, definice (SGUD), makra (SMAC) atd.
M = Manufacturer (výrobce stroje)	Data pro uvádění do provozu, která jsou určena pro specifickou typovou řadu strojů, jako jsou cykly výrobce, definice (MGUD), makra (MMAC) a strojní parametry, které definují množinu funkcí stroje.
I = Individual (pro určitý konkrétní stroj)	Data pro uvádění do provozu, která jsou určena pro konkrétní stroj, jako jsou kompenzační parametry a posunutí týkající se referenčních bodů.
U = User (uživatel)	Specifická data daného stroje, která jsou generována v průběhu jeho provozu, jako jsou parametry nástrojů, nastavované parametry, výrobní programy, uživatelské cykly, definice (UGUD) a makra (UMAC).

Literatura:

SINUMERIK 828D, Příručka pro uvedení do provozu, Soustružení a Frézování, kapitola: "Úvod a použití datových tříd"

Definice (DEF) uživatelských dat

Prostřednictvím souboru nebo adresáře, ve kterém je uživatelský datový objekt definován, je implicitně stanovena datová třída daného datového objektu. Datovou třídu datového objektu není možné změnit.

Při definici (DEF) uživatelského datového objektu je ale možné pro jeho **hodnotu** přiřadit vlastní datovou třídu, která se liší od datové třídy daného datového objektu.

Pro datové třídy daného datového objektu musí přitom platit:

Priorita datové třídy hodnoty datového objektu $\leq$ priorita datové třídy dat

Příklad:

Definice GUD, do kterého zapisuje měřicí sonda, by se měla nacházet v datové třídě M (= výrobce), protože je zapotřebí pro zpracování cyklů výrobce stroje. Hodnota tohoto datového objektu ale má patřit do datové třídy I (individuální), protože typ měřicí sondy může být u různých strojů různý.

MGUD.DEF (datová třída M)

...

DEF CHAN **DCI** INT CALIPER

MGUD.DEF (datová třída M)

...

Opětovná definice (REDEF) systémových dat

Datová třída systémových dat může být změněna pomocí opětovné definice (REDEF). Opětovná definice musí být uskutečněna v definičním souboru datové třídy S nebo M.

Pokud se používají soubory ve formátu ACCESS, smí být opětovné definice zadávány jen v rámci těchto souborů typu ACCESS.

Příslušné popisy datových tříd pro strojní, nastavované a volitelné parametry, jakož i pro systémové proměnné, se nacházejí v dokumentaci:

- Příručka pro sestavy, Podrobný popis strojních parametrů, parametr: „Třída“
- Příručka pro sestavy systémových proměnných

2.1.11 Přehled definovatelných a znovu definovatelných atributů

Následující tabulky ukazují, u kterých druhů dat mohou být definovány (DEF) a/nebo znovu definovány (REDEF) jednotlivé druhy atributů.

Systémová data

Druh dat	Počáteční hodnota	Mezní hodnoty	fyzikální jednotka	Přístupová oprávnění	Datová třída (pouze systém 828D)
Strojní parametry	---	---	---	REDEF	REDEF
Nastavované parametry	REDEF	---	---	REDEF	---
Data typu FRAME	---	---	---	REDEF	---
Data procesů	---	---	---	REDEF	---
Kompenzace chyby stoupání vřetena (EEC)	---	---	---	REDEF	---
Kompenzace průvěsu (CEC)	---	---	---	REDEF	---
Kompenzace chyby kvadrantu (QEC)	---	---	---	REDEF	---
Data zásobníku	---	---	---	REDEF	---
Parametry nástroje	---	---	---	REDEF	---
Chráněné oblasti	---	---	---	REDEF	---
Orientovatelný držák nástroje	---	---	---	REDEF	---
Kinematické řetězce	---	---	---	REDEF	---
3D chráněné oblasti	---	---	---	REDEF	---
Ohraničení pracovního pole	---	---	---	REDEF	---

Uživatelská data

Druh dat	Počáteční hodnota	Mezní hodnoty	fyzikální jednotka	Přístupová oprávnění	Datová třída
R-parametry	REDEF	REDEF	REDEF	REDEF	---
Proměnné synchronních akcí (\$AC_...)	REDEF	REDEF	REDEF	REDEF	---
GUD synchronních akcí (SYG_...)	REDEF	REDEF	REDEF	REDEF	---
Parametry EPS	REDEF	REDEF	REDEF	REDEF	---
Parametry nástroje OEM	REDEF	REDEF	REDEF	REDEF	---
Parametry zásobníku OEM	REDEF	REDEF	REDEF	REDEF	---
globální uživatelské proměnné (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF	DEF / REDEF
lokální uživatelské proměnné (PUD / LUD)	DEF	DEF	DEF	---	---

Viz také

Proměnné (Strana 21)

2.1.12 Definice a inicializace proměnných typu pole (DEF, SET, REP)

Uživatelská proměnná může být definována i jako 1- až maximálně 3-rozměrné pole (Array):

- 1-rozměrné: DEF <datový typ> <název proměnné>[<n>]
- 2-rozměrné: DEF <datový typ> <název proměnné>[<n>,<m>]
- 3-rozměrné: DEF <datový typ> <název proměnné>[<n>,<m>,<o>]

Poznámka

Uživatelské proměnné datového typu STRING mohou být definovány jako maximálně 2-rozměrné pole.

Datové typy

Uživatelské proměnné typu pole mohou být definovány pro následující datové typy: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Přiřazování hodnoty prvku pole

Přiřazování hodnot prvkům pole je možné uskutečňovat v následujících okamžicích:

- při inicializaci pole (inicializační hodnoty)
- v průběhu zpracování programu

Přiřazování hodnot se přitom může provádět následujícími způsoby:

- explicitní zadání prvku pole
- explicitní zadání prvku pole jako prvku počátečního a uvedení seznamu hodnot (SET)
- explicitní zadání prvku pole jako prvku počátečního a uvedení hodnoty spolu s četností jejího opakování (REP)

Poznámka

Uživatelským proměnným datového typu FRAME není možné přiřadit žádné inicializační hodnoty.

Syntaxe (DEF)

```
DEF <datový typ> <název proměnné> [<n>, <m>, <o>]
DEF STRING [<délka řetězce>] <název proměnné> [<n>, <m>]
```

Syntaxe (DEF...=SET...)

Použití seznamu hodnot:

- při definici:

```
DEF <datový typ> <název proměnné> [<n>, <m>, <o>] =
SET(<hodnota1>, <hodnota2>, ...)
```

stejný význam má:

```
DEF <datový typ> <název proměnné> [<n>, <m>, <o>] =
(<hodnota1>, <hodnota2>, ...)
```

Poznámka

Při inicializaci prostřednictvím seznamu hodnot je použití příkazu SET nepovinné.

- při přiřazování hodnot:

```
<název proměnné> [<n>, <m>, <o>] = SET(<hodnota1>, <hodnota2>, ...)
```

Syntaxe(DEF...=REP...)

Použití hodnoty s opakováním

- při definici:

```
DEF <datový typ> <název proměnné> [<n>, <m>, <o>] = REP(<hodnota>)
DEF <datový typ> <název proměnné> [<n>, <m>, <o>] =
REP(<hodnota>, <počet prvků pole>)
```
- při přiřazování hodnot:

```
<název proměnné> [<n>, <m>, <o>] = REP(<hodnota>)
<název proměnné> [<n>, <m>, <o>] = REP(<hodnota>, <počet prvků pole>)
```

Význam

DEF:	Příkaz pro definici proměnné
<datový typ>:	Datový typ proměnné
	Rozsah hodnot: - u systémové proměnné: BOOL, CHAR, INT, REAL, STRING, AXIS - u proměnných GUD nebo LUD: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME
<délka řetězce>:	Maximální počet znaků u datového typu STRING
<název proměnné>:	Název proměnné
[<n>, <m>, <o>]:	Velikost pole, příp. indexy pole
<n>:	Velikost pole, příp. index pole pro 1. rozměr
	Typ: INT (u systémových proměnných také AXIS)
	Rozsah hodnot: Max. velikost pole: 65535 Index pole: $0 \leq n \leq 65534$
<m>:	Velikost pole, příp. index pole pro 2. rozměr
	Typ: INT (u systémových proměnných také AXIS)
	Rozsah hodnot: Max. velikost pole: 65535 Index pole: $0 \leq m \leq 65534$
<o>:	Velikost pole, příp. index pole pro 3. rozměr
	Typ: INT (u systémových proměnných také AXIS)
	Rozsah hodnot: Max. velikost pole: 65535 Index pole: $0 \leq o \leq 65534$
SET:	Přiřazení hodnot prostřednictvím seznamu
(<hodnota1>, <hodnota2>, ...):	Seznam hodnot
REP:	Přiřazení hodnoty uvedené v poli <hodnota>
<hodnota>:	Hodnota, která se má při inicializaci zapisovat pomocí příkazu REP do prvků pole.
<počet prvků pole>:	Počet prvků pole, do kterých se má zapsat hodnota uvedená v poli <hodnota>. Pro zbývající prvky pole platí v závislosti na daném okamžiku: - Inicializace při definici pole: --> Do zbývajících prvků pole se dosazuje nula. - Přiřazení v průběhu zpracování programu: --> Momentální hodnoty prvků pole zůstávají zachovány. Pokud tento parametr není naprogramován, do prvků pole je dosazována hodnota uvedená v poli <hodnota>. Jestliže je tento parametr roven nule, platí v závislosti na daném okamžiku: - Inicializace při definici pole: --> Všem prvkům je dosazena nulová hodnota - Přiřazení v průběhu zpracování programu: --> Momentální hodnoty prvků pole zůstávají zachovány.

Index pole

Implicitní posloupnost prvků pole, např. při přiřazování hodnot pomocí příkazů SET nebo REP, se odvíjí od postupného přičítání indexů pole zprava doleva.

Příklad: Inicializace 3-rozměrného pole s 24 prvky:

```
DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1      1. Prvek pole
  FELD[0,0,1] = 1      2. Prvek pole
  FELD[0,0,2] = 1      3. Prvek pole
  FELD[0,0,3] = 1      4. Prvek pole
  ...
  FELD[0,1,0] = 1      5. Prvek pole
  FELD[0,1,1] = 1      6. Prvek pole
  ...
  FELD[0,2,3] = 1      12. Prvek pole
  FELD[1,0,0] = 1      13. Prvek pole
  FELD[1,0,1] = 1      14. Prvek pole
  ...
  FELD[1,2,3] = 1      24. Prvek pole
```

čemuž je ekvivalentní:

```
FOR n=0 TO 1
  FOR m=0 TO 2
    FOR o=0 TO 3
      FELD[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

Příklad: Inicializace kompletního pole proměnných

Aktuální dosazení viz obrázek.

Programový kód
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 definici				N20/N30: Inicializace pomocí identických hodnot			
	0	1	2		0	1	2		0	1	2
0	0	0	0		100	100	100		0	1	2
1	10	11	12		100	100	100		-10	-11	-12
2	20	20	20		100	100	100		-20	-20	-20
3	30	30	30		100	100	100		-30	0	0
4	40	40	40		100	100	100		0	-40	-40
5	0	0	0		-100	-100	-100		-50	-60	-70
6	0	0	0		-100	-100	-100		-100	-100	-100
7	0	0	0		-100	-100	-100		-100	-100	-100
8	0	0	0		-100	-100	-100		-100	8.1	8.2
9	0	0	0		-100	-100	-100		9.0	9.1	9.2
Prvky pole [5,0] až [9,2] byly inicializovány pomocí předdefinované hodnoty (0,0).								Prvky pole [3,1] až [4,0] byly inicializovány pomocí předdefinované hodnoty (0,0). Prvky pole [6,0] až [8,0] nebyly změněny.			

Viz také

Definice a inicializace proměnných typu pole (DEF, SET, REP): Další informace (Strana 56)
 Proměnné (Strana 21)

2.1.13 Definice a inicializace proměnných typu pole (DEF, SET, REP): Další informace

Další informace (SET)

Inicializace při definici:

- Počínaje od 1. prvku bude inicializováno hodnotami ze seznamu tolik prvků pole, kolik prvků je naprogramováno v seznamu hodnot.
- Prvkům pole bez explicitně uvedené hodnoty v seznamu (mezery v seznamu hodnot) je dosazována 0.
- U proměnných datového typu AXIS jsou mezery v seznamu hodnot nepřipustné.
- Pokud seznam obsahuje více hodnot, než kolik je definováno prvků pole, aktivuje se alarm.

Přiřazování hodnot během zpracovávání programu

Při přiřazování hodnot v průběhu zpracování programu platí pravidla uvedená výše v rámci definic. Kromě toho existují ještě následující možnosti:

- Jako prvky v seznamu hodnot jsou přípustné také výrazy.
- Přiřazování hodnot začíná u naprogramovaného indexu pole. Takto se dají cíleně obsazovat hodnotami dílčí pole.

Příklad:

Programový kód	Komentář
DEF INT FELD[5,5]	; definice pole
FELD[0,0]=SET(1,2,3,4,5)	; přiřazení hodnoty pro prvních 5 prvků pole [0,0] - [0,4]
FELD[0,0]=SET(1,2, , ,5)	; přiřazení hodnoty s mezerami pro prvních 5 prvků pole [0,0] - [0,4], prvky pole [0,2] a [0,3] = 0
FELD[2,3]=SET(VARIABLE,4*5.6)	; přiřazování hodnot s proměnnou a výrazem od indexu pole [2,3]: [2,3] = VARIABLE [2,4] = 4 * 5.6 = 22.4

Další informace (REP)**Inicializace při definici:**

- Všechny nebo libovolně zadaný počet prvků pole je inicializován uvedenou hodnotou (konstanta).
- Proměnné datového typu FRAME nemohou být inicializovány.

Příklad:

Programový kód	Komentář
DEF REAL varName[10]=REP(3.5,4)	; definice pole a inicializace prvků pole [0] až [3] s hodnotou 3,5

Přiřazování hodnot během zpracovávání programu

Při přiřazování hodnot v průběhu zpracování programu platí pravidla uvedená výše v rámci definic. Kromě toho existují ještě následující možnosti:

- Jako prvky v seznamu hodnot jsou přípustné také výrazy.
- Přiřazování hodnot začíná u naprogramovaného indexu pole. Takto se dají cíleně obsazovat hodnotami dílčí pole.

Příklady:

Programový kód	Komentář
DEF REAL varName[10]	; definice pole
varName[5]=REP(4.5,3)	; prvky pole [5] až [7] = 4,5
R10=REP(2.4,3)	; R-parametr R10 až R12 = 2,4
DEF FRAME FRM[10]	; definice pole

2.1 Proměnné

Programový kód	Komentář
FRM[5] = REP(CTTRANS (X,5))	; prvky pole [5] až [9] = CTTRANS(X,5)

Viz také

Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 52)

2.1.14 Datové typy

U NC systémů jsou k dispozici následující datové typy:

Datový typ	Význam	Rozsah hodnot
INT	celočíslné hodnoty se znaménkem	-2147483648 ... +2147483647
REAL	Reálné číslo (LONG REAL podle IEEE)	$\pm(\sim 2,2 \cdot 10^{-308} \dots \sim 1,8 \cdot 10^{+308})$
BOOL	Logická hodnota TRUE (1) a FALSE (0)	1, 0
CHAR	Znaky ASCII	Kód ASCII 0 ... 255
STRING	Řetězec znaků definované délky	Maximálně 200 znaků (žádné speciální znaky)
AXIS	Identifikátor osy/vřetena	Identifikátor kanálové osy
FRAME	Geometrické údaje pro statické transformace souřadného systému (posunutí, otočení, změna měřítka, zrcadlové převrácení)	---

Implicitní převody datového typu

Jsou možné následující převody datových typů a při přiřazování a při přenášení parametrů se implicitně uskutečňují:

z ↓ / do →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x

x: možné bez omezení
 o: Je možná ztráta dat v důsledku překročení rozsahu hodnot ⇒ alarm;
 Zaokrouhlení: Hodnota za desetinnou čárkou $\geq 0,5$ → zaokrouhleno nahoru, hodnota za desetinnou čárkou $< 0,5$ → zaokrouhleno dolů
 &: Hodnota $\neq 0$ ⇒ TRUE, hodnota $= 0$ ⇒ FALSE

Viz také

Proměnné (Strana 21)

2.1.15 Kontrola existence proměnné (ISVAR)

Prostřednictvím předdefinované funkce ISVAR je možné zkontrolovat, zdali je systémová/ uživatelská proměnná (např. strojní parametr, nastavovaný parametr, systémová proměnná, všeobecná proměnná, jako např. GUD) v NC známa.

Proměnná

Proměnná, která je předmětem dotazu, má následující strukturu:

Bezrozměrná proměnná: <proměnná>
 Jednorozměrná proměnná bez indexu pole <proměnná>[]
 Jednorozměrná proměnná s indexem pole <proměnná>[<n>]
 Dvourozměrná proměnná bez indexu pole <proměnná>[,]
 Dvourozměrná proměnná s indexy pole n a m: <proměnná>[<n>,<m>]

Syntaxe

<výsledek>=ISVAR (<proměnná> [<n>,<m>])

Význam

<výsledek>:	Výsledná hodnota		
	Datový typ:	BOOL	
	Rozsah hodnot:	1	Proměnná existuje
		0	Proměnná je neznámá
ISVAR:	Kontrola, zda je uvedená systémová/uživatelská proměnná v NC systému známa.		
<proměnná>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<n>:	Index prvního rozměru (není nutné)		
	Datový typ:	INT	
<m>:	Index druhého rozměru (není nutné)		
	Datový typ:	INT	

V závislosti na předávaném parametru jsou prováděny následující zkoušky:

- Je název znám
- Je proměnná typu pole
- Je to jedno- nebo dvourozměrné pole
- Je příslušný index pole v rozsahu přípustných hodnot

Když odpovědi na všechny tyto otázky jsou kladné, je výsledkem hodnota TRUE (1).

Jestliže je odpověď na jednu z těchto otázek záporná nebo pokud se vyskytla syntaktická chyba, je výsledkem hodnota FALSE (0).

Příklady

Programový kód	Komentář
DEF INT VAR1 DEF BOOL IS_VAR=FALSE N10 IS_VAR=ISVAR("VAR1")	; IS_VAR je v tomto případě TRUE
Programový kód	Komentář
DEF REAL VARARRAY[10,10] DEF BOOL IS_VAR=FALSE N10 IS_VAR=ISVAR("VARARRAY[,]") N20 IS_VAR=ISVAR("VARARRAY") N30 IS_VAR=ISVAR("VARARRAY[8,11]") N40 IS_VAR=ISVAR("VARARRAY[8,8]") N50 IS_VAR=ISVAR("VARARRAY[,8]") N60 IS_VAR=ISVAR("VARARRAY[8,]")	; IS_VAR je v tomto případě TRUE, je to dvourozměrné pole ; IS_VAR je TRUE, proměnná existuje ; IS_VAR je FALSE, index pole je nepřipustný ; IS_VAR je FALSE, chybí znak "]" (syntaktická chyba) ; IS_VAR je TRUE, index pole je přípustný ; IS_VAR je TRUE, index pole je přípustný
Programový kód	Komentář
DEF BOOL IS_VAR=FALSE N100 IS_VAR=ISVAR("\$MC_GCODE_RESET_VALUES[1]")	; předávaný parametr je strojní parametr, IS_VAT je TRUE
Programový kód	Komentář
DEF BOOL IS_VAR=FALSE N10 IS_VAR=ISVAR("\$P_EP") N20 IS_VAR=ISVAR("\$P_EP[X]")	; IS_VAR je v tomto případě TRUE ; IS_VAR je v tomto případě TRUE

2.1.16 Načítání hodnot atributu/datového typu (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

Pomocí předem definovaných funkcí GETVARPHU, GETVARAP, GETVARLIM a GETVARDFT je možné načítat hodnoty atributů systémových/uživatelských proměnných, pomocí funkce GETVARTYP se zjišťuje datový typ systémové/uživatelské proměnné.

Načítání fyzikálních jednotek

Syntaxe:

<výsledek>=GETVARPHU(<název>)

Význam:

<výsledek>:	Číselná hodnota fyzikální jednotky	
	Datový typ:	INT
	Rozsah hodnot:	viz tabulka v kapitole "Atribut: Fyzikální jednotka (PHU) (Strana 43)"
	V případě chyby:	
	- 2	Údaj zadaný v parametru <název> není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
GETVARPHU:	Čtení fyzikálních jednotek systémové/uživatelské proměnné	
<název>:	Název systémové/uživatelské proměnné	
	Datový typ:	STRING

Příklad:

NC obsahuje následující proměnné GUD:

```
DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric
```

Programový kód	Komentář
DEF INT result=0	
result=GETVARPHU("elec- tric")	; zjišťování fyzikálních jednotek proměnných typu GUD
IF (result < 0) GOTO error	

Jako výsledek systém vrací hodnotu 42. To odpovídá fyzikální jednotce [kW].

Poznámka

Pomocí funkce GETVARPHU se dá např. zkontrolovat, zda při přiřazování hodnot proměnným $a = b$ mají obě proměnné očekávané fyzikální jednotky.

Načítání přístupových oprávnění**Syntaxe:**

```
<výsledek>=GETVARAP (<název>,<přístup>)
```

Význam:

<výsledek>:	Úroveň ochrany pro zadaný parametr <přístup>		
	Datový typ:	INT	
	Rozsah hodnot:	0 ... 7	Viz "Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB) (Strana 45)".
		V případě chyby:	
		- 1	zápis není možný (má smysl jen v případě, že pro <přístup> je nastaveno „WP“ a „WB“)
		- 2	Údaj zadaný v parametru <název> není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
- 3	nesprávná hodnota pro parametr <přístup>		
GETVARAP:	Čtení přístupových oprávnění k systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<přístup>:	Druh přístupu		
	Datový typ:	STRING	
	Rozsah hodnot:	"RP"	Čtení prostřednictvím výrobního programu
		"WP"	Zápis prostřednictvím výrobního programu
		"RB"	Čtení prostřednictvím BTSS
		"WB"	Zápis prostřednictvím BTSS

Příklad:

Programový kód	Komentář
DEF INT result=0	
result=GETVARAP("&TC_MAP8", "WB")	; zjištění ochrany proti přístupu pro systémový parametr "Magazinposition" (Poloha zásobníku), pokud jde o zápis pomocí BTSS
IF (result < 0) GOTO error	

Jako výsledek systém vrací hodnotu 7. To odpovídá poloze 0 přepínače na klíč (= žádná ochrana proti přístupu).

Poznámka

Pomocí funkce GETVARAP je možné realizovat například kontrolní program, který zkouší přístupová oprávnění očekávaná aplikací.

Čtení mezních hodnot**Syntaxe:**

<Status>=GETVARLIM(<název>,<mezní hodnota>,<výsledek>)

Význam:

<Status>:	Stav funkce		
	Datový typ:	INT	
	Rozsah hodnot:	1	o. k.
		-1	není definována žádná mezní hodnota (u proměnných typu AXIS, STRING, FRAME)
		-2	Údaj zadaný v parametru <název> není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
-3		nesprávná hodnota pro parametr <mezní hodnota>	
GETVARLIM:	Čtení dolní/horní mezní hodnoty systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<mezní hodnota>:	Udává, která mezní hodnota má být načtena.		
	Datový typ:	CHAR	
	Rozsah hodnot:	"L" :	= dolní mezní hodnota
		"U" :	= horní mezní hodnota
<výsledek>:	Výsledná mezní hodnota		
	Datový typ:	VAR REAL	

Příklad:

Programový kód	Komentář
DEF INT state=0	
DEF REAL result=0	
state=GETVAR-	; zjištění dolní mezní hodnoty pro parametr
LIM("\$MA_MAX_AX_VELO", "L", result)	MD32000 \$MA_MAX_AX_VELO
IF (result < 0) GOTO error	

Načtení atributu/datového typu**Syntaxe:**

<výsledek>=GETVARDIM(<název>, index)

Význam:

<výsledek>:	Rozměr/počet polí <index>	
	Datový typ:	INT
GETVARDIM:	Čtení dolní/horní mezní hodnoty systémové/uživatelské proměnné	
<název>:	Čtení počtu prvků pole	
	Datový typ:	STRING
<Index>:	Počet polí, max. 3.	
	Datový typ:	INT

Příklad:

Programový kód	Komentář
N5 DEF REAL myReal[5,4]	
N10 R1 = GETVATDIM(„myReal“,1)	R1 = 5
N15 R2 = GETVATDIM(„myReal“,2)	R2 = 4

Načtení standardní hodnoty**Syntaxe:**

```
<Status>=GETVARDFT(<název>,<výsledek>[,<Index_1>,<Index_2>,<Index_3>])
```

Význam:

<Status>:	Stav funkce		
	Datový typ:	INT	
	Rozsah hodnot:	1	o. k.
		-1	není k dispozici žádná standardní hodnota (např. protože parametr <výsledek> obsahuje pro parametr <název> nesprávný typ)
		-2	Údaj zadaný v parametru <název> není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
		-3	nesprávná hodnota pro parametr <Index_1>, rozměr je menší než jedna (= žádné pole = skalár)
		-4	nesprávná hodnota pro parametr <Index_2>
-5	nesprávná hodnota pro parametr <Index_3>		
GETVARDFT:	Čtení standardní hodnoty systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<výsledek>:	Výsledná standardní hodnota		
	Datový typ:	VAR REAL (při čtení standardní hodnoty proměnných typů INT, REAL, BOOL, AXIS)	
		VAR STRING (při čtení standardní hodnoty proměnných typů STRING a CHAR)	
		VAR FRAME (při čtení standardní hodnoty proměnných typu FRAME)	
<Index_1>:	Index na první rozměr (není nutné)		
	Datový typ:	INT	
	Pokud není naprogramováno, znamená = 0		
<Index_2>:	Index na druhý rozměr (není nutné)		
	Datový typ:	INT	
	Pokud není naprogramováno, znamená = 0		

<Index_3>:	Index na třetí rozměr (není nutné)	
	Datový typ:	INT
	Pokud není naprogramováno, znamená = 0	

Příklad:

Programový kód	Komentář
DEF INT state=0	
DEF REAL resultR=0	; proměnná pro převzetí standardní hodnoty typu INT, REAL, BOOL, AXIS
DEF FRAME resultF=0	; proměnná pro převzetí standardní hodnoty typu FRAME
IF (GETVARTYP("\$MA_MAX_AX_VELO") <> 4) GOTO error	
state=GETVARDF("MA_MAX_AX_VELO", resultR, AXTOINT(X))	; zjištění standardní hodnoty osy "X".
IF (resultR < 0) GOTO error	
IF (GETVARTYP("\$TC_TP8") <> 3) GOTO error	
state=GETVARDF("\$TC_TP8", resultR)	
IF (GETVARTYP("\$P_UBFR") <> 7) GOTO error	
state=GETVARDF("\$P_UBFR", resultF)	

Načtení datového typu**Syntaxe:**

<výsledek>=GETVARTYP(<název>)

Význam:

<výsledek>:	Datový typ uvedené systémové/uživatelské proměnné		
	Datový typ:	INT	
	Rozsah hodnot:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		V případě chyby:	
< 0	Údaj zadaný v parametru <název> není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.		
GETVARTYP:	Čtení datového typu systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	

Příklad:

Programový kód	Komentář
DEF INT result=0	
DEF STRING name="R"	
result=GETVARTYP(name)	; zjištění typu R-parametru
IF (result < 0) GOTOF error	

Jako výsledek systém vrací hodnotu 4. To odpovídá datovému typu REAL.

2.2 Nepřímé programování

2.2.1 Nepřímé programování adres

Při nepřímém programování adres je rozšířená adresa (<Index>) nahrazována proměnnou vhodného typu.

Poznámka

Nepřímé programování není možné u těchto adres:

- N (číslo bloku)
- L (podprogram)
- Nastavitelné adresy
(např. X[1] namísto X1 není přípustné)

Syntaxe

<ADRESA> [<Index>]

Význam

<ADRESA> [...]:	Pevná adresa s rozšířením (index)
<Index>:	Proměnná, např. pro číslo vřetena, osy, ...

Příklady

Příklad 1: Nepřímé programování čísla vřetena

Přímé programování:

Programový kód	Komentář
S1=300	; otáčky 300 ot/min pro vřeteno s číslem 1

Nepřímé programování:

Programový kód	Komentář
DEF INT SPINU=1	; definice proměnné typu INT a přiřazení hodnoty
S[SPINU]=300	; otáčky 300 ot/min pro vřeteno, jehož číslo je uloženo v proměnné SPINU (v tomto příkladu je to vřeteno s číslem 1)

Příklad 2: Nepřímé programování osy

Přímé programování:

Programový kód	Komentář
FA[U]=300	; posuv 300 pro osu "U"

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR2=U	; definice proměnné typu AXIS a přiřazení hodnoty
FA[AXVAR2]=300	; posuv 300 pro osu, jejíž adresový název je uložen v proměnné s názvem AXVAR2

Příklad 3: Nepřímé programování osy

Přímé programování:

Programový kód	Komentář
\$AA_MM[X]	; načtení hodnoty změřené měřicí sondou (MCS) pro osu "X"

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR3=X	; definice proměnné typu AXIS a přiřazení hodnoty
\$AA_MM[AXVAR3]	; načtení hodnoty změřené měřicí sondou (MCS) pro osu, jejíž název je uložen v proměnné AXVAR3

Příklad 4: Nepřímé programování osy

Přímé programování:

Programový kód
X1=100 X2=200

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR1 AXVAR2	; definice dvou proměnných typu AXIS
AXVAR1=(X1) AXVAR2=(X2)	; přiřazení názvů os
AX[AXVAR1]=100 AX[AXVAR2]=200	; pohyb osami, jejichž adresové názvy jsou uloženy v proměnných s názvy AXVAR1 a AXVAR2

Příklad 5: Nepřímé programování osy

Přímé programování:

Programový kód
G2 X100 I20

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR1=X	; definice proměnné typu AXIS a přiřazení hodnoty
G2 X100 IP[AXVAR1]=20	; nepřímé programování zadání středu pro osu, jejíž adresovací název je uložen v proměnné s názvem AXVAR1

Příklad 6: Nepřímé programování prvků pole

Přímé programování:

Programový kód	Komentář
DEF INT FELD1[4,5]	; definice pole 1

Nepřímé programování:

Programový kód	Komentář
DEFINE DIM1 AS 4	; v případě rozměrů pole musí být velikosti pole uvedeny jako pevné hodnoty
DEFINE DIM2 AS 5	
DEF INT FELD[DIM1,DIM2]	
FELD[DIM1-1,DIM2-1]=5	

Příklad 7: Nepřímé volání podprogramu

Programový kód	Komentář
CALL "L" << R10	; Volání programu, jehož číslo je uloženo v R10 (skládání řetězců).

2.2.2 Nepřímé programování příkazů G-funkcí

Nepřímé programování příkazů G-funkcí umožňuje efektivní programování cyklů.

Syntaxe

G[<skupina>]=<číslo>

Význam

G[...]:	G-příkaz s rozšířením (index)	
<skupina>:	Indexový parametr: G-skupina	
	Typ:	INT
<číslo>:	Proměnná pro číslo příkazu G-funkce	
	Typ:	INT nebo REAL

Poznámka

Nepřímo mohou být zpravidla programovány pouze příkazy G-funkcí, které nejsou syntakticky určující.

Ze syntakticky určujících příkazů G-funkcí jsou možné pouze kódy z 1. skupiny G-funkcí. Příkazy G-funkcí, které určují syntaxi, ze skupin G-funkcí 2, 3 a 4 možné nejsou.

Poznámka

V nepřímém programování příkazů G-funkcí nejsou dovoleny žádné aritmetické funkce. Nutný výpočet čísla příkazu G-funkce musí proběhnout na samostatném řádku výrobního programu před nepřímým programováním příkazu G-funkce.

Příklady**Příklad 1: Nastavitelné posunutí počátku (skupina G-funkcí č. 8)**

Programový kód	Komentář
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 příkazu G-funkce
N1110 G[8]=INT_VAR G1 X0 Y0	; G55

Příklad 2: Volba roviny (skupina G-funkcí č. 6)

Programový kód	Komentář
N2010 R10=\$P_GG[6]	; načtení aktivního příkazu G-funkce ze skupiny 6
...	
N2090 G[6]=R10	

Literatura

Pokud budete potřebovat informace ke skupinám G-funkcí, viz:
Programovací příručka, Základy, kapitola "Skupiny G-funkcí"

2.2.3 Nepřímé programování atributů polohy (GP)

Atributy polohy, jako např. inkrementální nebo absolutní programování poloh os, mohou být ve spojení s klíčovým slovem GP programovány nepřímo jako proměnné.

Aplikace

Nepřímé programování atributů polohy nachází uplatnění v **nahrazovacích cyklech**, protože zde oproti programování atributů polohy jako klíčových slov (např. IC, AC, ...) existuje následující výhoda:

Díky nepřímému programování, které využívá proměnné, není zapotřebí **žádný** příkaz CASE, který by zajišťoval větvení na všechny možné atributy polohy.

Syntaxe

```
<PŘÍKAZ POLOHOVÁNÍ>[<osa/vřeteno>]=
GP(<poloha>,<atribut polohy>)
<osa/vřeteno>=GP(<poloha>,<atribut polohy>)
```

Význam

<PŘÍKAZ POLOHOVÁNÍ>[]:	Spolu s klíčovým slovem GP mohou být naprogramovány následující polohovací příkazy: POS, POSA, SPOS, SPOSA Kromě toho je možné zadat: - všechny identifikátory os/vřeten vyskytující se v kanálu: <osa/vřeteno> - proměnný identifikátor osy/vřetena AX
<osa/vřeteno>:	Osa/vřeteno, jehož poloha má být nastavena
GP():	Klíčové slovo pro polohování
<pozice>:	Parametr 1 Pozice osy/vřetena jako konstanta nebo jako proměnná
<atribut polohy>:	Parametr 2 Atribut polohy (např. režim najíždění na pozici) jako proměnná (např. \$P_SUB_SPOSMODE) nebo jako klíčové slovo (IC, AC, ...)

Hodnoty předávané proměnnými mají následující význam:

Hodnota	Význam	Přípustné pro:
0	Žádná změna atributu polohy	
1	AC	POS, POSA, SPOS, SPOSA, AX, adresa osy
2	IC	POS, POSA, SPOS, SPOSA, AX, adresa osy
3	DC	POS, POSA, SPOS, SPOSA, AX, adresa osy
4	ACP	POS, POSA, SPOS, SPOSA, AX, adresa osy
5	ACN	POS, POSA, SPOS, SPOSA, AX, adresa osy
6	OC	-
7	PC	-
8	DAC	POS, POSA, AX, adresa osy
9	DIC	POS, POSA, AX, adresa osy
10	RAC	POS, POSA, AX, adresa osy
11	RIC	POS, POSA, AX, adresa osy
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Příklad:

Jestliže je aktivní synchronní vazba mezi řídicím vřetenem S1 a vlečným vřetenem S2, příkazem SPOS v hlavním programu se vyvolává následující nahrazovací cyklus, který polohování vřeten zajišťuje.

Polohování se uskutečňuje prostřednictvím příkazu v bloku N2230:

```
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE) SPOS[2]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
```

Poloha, na kterou se má najet, se načítá ze systémové proměnné \$P_SUB_SPOSIT, režim najíždění na pozici se načítá ze systémové proměnné \$P_SUB_SPOSMODE.

Programový kód	Komentář
N1000 PROC LANG_SUB DISPLOF SBLOF	
...	
N2100 IF(\$P_SUB_AXFCT==2)	
N2110	; nahrazení příkazu SPOS / SPOSA / M19 v případě aktivní synchronní vazby
N2185 DELAYFSTON	; začátek oblasti zastavení-prodleva
N2190 COUPOF(S2,S1)	; deaktivování synchronní vazby mezi vřeteny
N2200	; polohování řídicího a vlečného vřetena
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220	; polohování vřetena s příkazem SPOS:
N2230 SPOS[1]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
SPOS[2]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
N2250 ELSE	
N2260	; polohování vřetena s příkazem M19:
N2270 M1=19 M2=19	; polohování řídicího a vlečného vřetena
N2280 ENDIF	
N2285 DELAYFSTOF	; konec oblasti zastavení-prodleva
N2290 COUPON(S2,S1)	; aktivování synchronní vazby mezi vřeteny
N2410 ELSE	
N2420	; dotaz na další nahrazování
...	
N3300 ENDIF	
...	
N9999 RET	

Okrajové podmínky

- V synchronních akcích není nepřímé programování atributů polohy možné.

Literatura

Příručka k funkcím, Základní funkce; BAG, kanál, zpracování programu, chování při resetu (K1),
kapitola: Nahrazování funkcí NC systému pomocí podprogramů

2.2.4 Nepřímé programování řádků výrobního programu (EXECSTRING)

Pomocí příkazu výrobního programu `EXECSTRING` je možné spustit zpracování dříve sestavené proměnné typu `STRING`, jako by se jednalo o řádek výrobního programu.

Syntaxe

Příkaz `EXECSTRING` musí být naprogramován na samostatném řádku výrobního programu.
`EXECSTRING (<proměnná typu string>)`

Význam

<code>EXECSTRING:</code>	Příkaz pro zpracování proměnné typu <code>STRING</code> , jako kdyby šlo o řádek výrobního programu
<code><proměnná typu String>:</code>	Proměnná typu <code>STRING</code> , která obsahuje řádek výrobního programu, který může být samostatně zpracován.

Poznámka

Pomocí příkazu `EXECSTRING` mohou být předávány veškeré konstrukce výrobního programu, které mohou být naprogramovány **v programové části výrobního programu**, s výjimkou řídicích struktur (Strana 115). Vyloučeny jsou tedy příkazy `PROC` a `DEF`, jakož i obecně použití souborů `INI` a `DEF`.

Příklad:

Programový kód	Komentář
N100 DEF STRING[100] MY_BLOCK	; definice proměnné typu String pro sestavení zpracovatelného řádku výrobního programu
N110 DEF STRING[10] MFCT1="M7"	
...	
N200 EXECSTRING(MFCT1 << "M4711")	; zpracování řádku výrobního programu "M7 M4711"
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	

2.2 Nepřímé programování

Programový kód	Komentář
N350 EXECSTRING(MY_BLOCK)	; zpracování řádku výrobního programu "M3 M7"

2.3 Matematické funkce

Operátory / matematické funkce	Význam
+	sečítání
-	odečítání
*	násobení
/ ¹⁾	Dělení ¹⁾
DIV ¹⁾	Celočíselné dělení ¹⁾
MOD ¹⁾	Dělení Modulo (výsledkem je zbytek po celočíselném dělení) ¹⁾
:	Řetězcový operátor pro proměnné typu FRAME
SIN ()	sinus
COS ()	kosinus
TAN ()	tangens
ASIN ()	arkus sinus
ACOS ()	arkus kosinus
ATAN2 (,) ¹⁾	Arkus tangens na druhou ¹⁾
SQRT ()	druhá odmocnina
ABS ()	absolutní hodnota
POT ()	2. mocnina (čtverec)
TRUNC ()	celočíselná část Nastavení přesnosti při příkazech porovnávání pomocí funkce TRUNC (viz "Korekce přesnosti při příkazech porovnávání (TRUNC) (Strana 80)")
ROUND ()	zaokrouhlování na celá čísla
LN ()	přirozený logaritmus
EXP ()	exponenciální funkce
MINVAL ()	menší hodnota ze dvou proměnných (viz "Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND) (Strana 82)")
MAXVAL ()	větší hodnota ze dvou proměnných (viz "Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND) (Strana 82)")
BOUND ()	Hodnota proměnné, která leží v definovaném intervalu hodnot (viz "Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND) (Strana 82)")
CTrans ()	Posunutí
CROT ()	Otočení
CSCALE ()	změna měřítka
CMIRROR ()	zrcadlové převrácení
1) Viz odstavec „Příklady“	

Programování

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

Příklady

Dělení: /

(Typ REAL) = (Typ INT nebo Typ REAL) / (Typ INT nebo Typ REAL);

Příklad: $3 / 4 = 0,75$

Celočíselné dělení: DIV

(Typ INT) = (Typ INT nebo REAL) / (Typ INT nebo REAL);

Příklad: $7 \text{ DIV } 4.1 = 1$

Dělení Modulo (výsledkem je zbytek po celočíselném dělení): MOD

(Typ REAL) = (Typ INT nebo REAL) MOD (Typ INT nebo REAL);

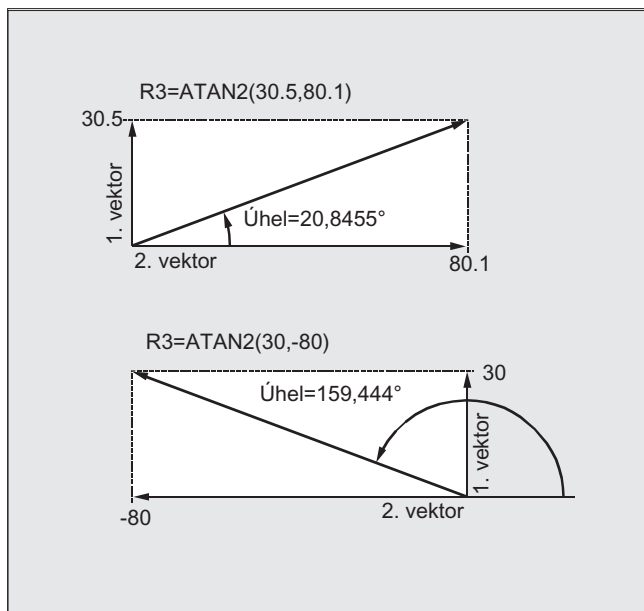
Příklad: $7 \text{ MOD } 4.1 = 2.9$

Arkus tangens na druhou: ATAN2

Matematická funkce ATAN2 vypočte ze dvou vůči sobě kolmo ležících vektorů úhel součtu těchto vektorů.

Výsledek vždy leží v jednom ze čtyř kvadrantů ($-180^\circ < 0 < +180^\circ$).

Základem pro úhlový vztah je vždy 2. hodnota v pozitivním směru.



Příklady programování

Programový kód	Komentář
R1=R1+1	; nová hodnota R1 = stará hodnota R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; násobení se provádí před sečítáním
R14=(R1+R2)*R3	; výrazy v závorkách se vypočítají napřed
R15=SQRT(POT(R1)+POT(R2))	; napřed se vyhodnotí vnitřní závorky:
	R15 = druhá odmocnina((R1 ² + R2 ²))
RESFRAME=FRAME1:FRAME2	; FRAME=spojení s řetězcovým operátorem
FRAME3=CTrans (...):CROT(...)	Přiřazování hodnot složkám framu

2.4 Porovnávací a logické operace

Porovnávací operace se mohou používat např. pro formulování podmínky skoku. Přitom je možné porovnávat i složité výrazy.

Relační operace jsou použitelné pro proměnné typů `CHAR`, `INT`, `REAL` a `BOOL`. U typu `CHAR` je porovnávána kódová hodnota.

U typů `STRING`, `AXIS` a `FRAME` jsou možné operace `==` a `<>`, které se mohou používat také pro operace s typem `STRING` v synchronních akcích.

Výsledek relačních operací je vždy typu `BOOL`.

Logické operátory slouží ke spojení pravdivostních hodnot.

Logické operace mohou být používány jen s proměnnými typu `BOOL`. Prostřednictvím interního převádění typů je možno pracovat i s datovými typy `CHAR`, `INT` a `REAL`.

U logických (boolovských) operací platí pro datové typy `BOOL`, `CHAR`, `INT` a `REAL`:

- 0 odpovídá hodnotě: `FALSE`
- cokoli nerovnajícího se 0 odpovídá hodnotě: `TRUE`

Bitové logické operace

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

Programování

Relační operátor	Význam
<code>==</code>	rovná se
<code><></code>	nerovná se
<code>></code>	je větší než
<code><</code>	je menší než
<code>>=</code>	je větší nebo rovno
<code><=</code>	je menší nebo rovno

Logický operátor	Význam
<code>AND</code>	logické a
<code>OR</code>	logické nebo
<code>NOT</code>	negace
<code>XOR</code>	logické XOR

Bitový logický operátor	Význam
<code>B_AND</code>	bitové logické a
<code>B_OR</code>	bitové logické nebo
<code>B_NOT</code>	bitová negace
<code>B_XOR</code>	bitové logické XOR

Poznámka

V aritmetických výrazech může být pomocí kulatých závorek definováno pořadí zpracování všech operátorů, které může být odlišné od normálních pravidel priority operací.

Poznámka

Mezi boolovskými operandy a operátory musí být zapisována mezera.

Poznámka

Operátor `B_NOT` se vztahuje pouze na jeden operand. Tento operand se nachází za operátorem.

Příklady

Příklad 1: Relační operátory

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

nebo

```
R11=R10>=100
```

```
IF R11 GOTOF ZIEL
```

Výsledek porovnávání `R10>=100` se napřed ukládá do pomocné proměnné `R11`.

Příklad 2: Logické operátory

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

nebo

```
IF NOT R10 GOTOB START
```

`NOT` se vztahuje pouze na jeden operand.

Příklad 3: Bitové logické operace

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

2.5 Korekce přesnosti při příkazech porovnávání (TRUNC)

Příkaz TRUNC oddělí celočíselnou část operandu po vynásobení faktorem přesnosti.

Nastavitelná přesnost u příkazů porovnávání

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 příkazů porovnávání nikoli absolutní shodnost, nýbrž relativní shodnost.

Syntaxe

Korekce přesnosti při chybách porovnávání

TRUNC (R1*1000)

Význam

TRUNC:	Odříznutí míst za desetinnou čárkou
--------	-------------------------------------

Uvažovaná relativní rovnost 10^{-12} při následujících operacích

- Rovnost: (==)
- Nerovnost: (<>)
- Je větší nebo rovno: (>=)
- Je menší nebo rovno: (<=)
- Je větší/menší: (><) s absolutní rovností
- Je větší: (>)
- Je menší: (<)

Kompatibilita

Z důvodu kompatibility může být zkouška relativní shodnosti u operací (>) a (<) nastavením strojního parametru MD10280 \$MN_ PROG_FUNCTION_MASK Bit0 = 1 deaktivována.

Poznámka

Porovnávání s daty typu REAL je z výše uvedených důvodů obecně zatíženo určitou nepřesností. V případě neakceptovatelných odchylek musí být použit přepočít na typ INTEGER, při kterém jsou operandy vynásobeny faktorem přesnosti a pak jsou oříznuty pomocí funkce TRUNC.

Synchronní akce

Popisované chování relačních příkazů platí i při synchronních akcích.

Příklady

Příklad 1: Sledování shodnosti

Programový kód	Komentář
N40 R1=61.01 R2=61.02 R3=0.01	; přiřazení počátečních hodnot
N41 IF ABS(R2-R1) > R3 GOTOF FEHLER	; skok byl už dříve proveden
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)	; korekce přesnosti
R13=TRUNC(R3*1000)	
IF ABS(R12-R11) > R13 GOTOF FEHLER	; skok se už neprovede
M30	; konec programu
FEHLER: SETAL(66000)	

Příklad 2: Vytvoření a vyhodnocení koeficientu obou operandů

Programový kód	Komentář
R1=61.01 R2=61.02 R3=0.01	; přiřazení počátečních hodnot
IF ABS((R2-R1)/R3)-1 > 10EX-5 GOTOF FEHLER	; skok se neprovede
M30	; konec programu
FEHLER: SETAL(66000)	

2.6 Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND)

Pomocí příkazů `MINVAL` a `MAXVAL` mohou být mezi sebou porovnávány hodnoty dvou proměnných. Jako výsledek je na výstupu menší hodnota (v případě funkce `MINVAL`), příp. větší hodnota (v případě funkce `MAXVAL`).

Pomocí příkazu `BOUND` je možné zkontrolovat, zda se hodnota zkoušené proměnné nachází v rámci definovaného rozmezí hodnot.

Syntaxe

```
<menší hodnota>=MINVAL(<proměnná1>,<proměnná2>)
<větší hodnota>=MAXVAL(<proměnná1>,<proměnná2>)
<vracená hodnota>=<BOUND>(<minimum>,<maximum>,<zkoušená proměnná>)
```

Význam

MINVAL:	Zjišťuje menší hodnotu ze dvou proměnných (<proměnná1>, <proměnná2>)
<menší hodnota>:	Proměnná výsledku pro příkaz <code>MINVAL</code> Je jí dosazena hodnota menší proměnné.
MAXVAL:	Zjišťuje větší hodnotu ze dvou proměnných (<proměnná1>, <proměnná2>)
<větší hodnota>:	Proměnná výsledku pro příkaz <code>MAXVAL</code> Je jí dosazena hodnota větší proměnné.
BOUND:	Zkontroluje, zda proměnná (<zkoušená proměnná>) leží v rámci definovaného rozsahu hodnot.
<minimum>:	Proměnná, která definuje minimální hodnotu rozsahu hodnot.
<maximum>:	Proměnná, která definuje maximální hodnotu rozsahu hodnot.
<vracená hodnota>:	Proměnná výsledku pro příkaz <code>BOUND</code> Pokud hodnota zkoušené proměnné leží v rámci definovaného rozsahu hodnot, potom se do proměnné výsledku dosadí hodnota zkoušené proměnné. Pokud je hodnota zkoušené proměnné větší než maximální hodnota, potom se do proměnné výsledku dosadí maximální hodnota definovaného rozsahu. Pokud je hodnota zkoušené proměnné menší než minimální hodnota, potom se do proměnné výsledku dosadí minimální hodnota definovaného rozsahu.

Poznámka

Funkce `MINVAL`, `MAXVAL` a `BOUND` mohou být naprogramovány i v synchronních akcích.

Poznámka**Chování v případě rovnosti**

V případě rovnosti budou funkce MINVAL/MAXVAL vracet tuto stejnou hodnotu. V případě funkce BOUND bude výsledkem opět hodnota zkoušené proměnné.

Příklad:

Programový kód	Komentář
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar	
rValMin=MINVAL(rVar1,rVar2)	; proměnná rValMin se nastaví na hodnotu 10,5
rValMax=MAXVAL(rVar1,rVar2)	; proměnná rValMax se nastaví na hodnotu 33,7
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; proměnná rVar3 leží v rámci hranic, rRetVar se nastaví na hodnotu 19,7
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; proměnná rVar3 leží pod hranicí minima, rRetVar se nastaví na hodnotu 10,5
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; proměnná rVar3 leží nad hranicí maxima, rRetVar se nastaví na hodnotu 33,7

2.7 Priorita operací

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 definováno pořadí zpracování všech operátorů, které může být odlišné od normálních pravidel priority operací.

Posloupnost operátorů

Od nejvyšší po nejnižší prioritu

1.	NOT, B_NOT	Negace, bitová negace
2.	*, /, DIV, MOD	Násobení, dělení
3.	+, -	Sečítání, odečítání
4.	B_AND	bitové logické a
5.	B_XOR	bitové logické XOR
6.	B_OR	bitové logické nebo
7.	AND	logické a
8.	XOR	logické XOR
9.	OR	logické nebo
10.	<<	Zřetězení řetězců, typ výsledku STRING
11.	==, <>, >, <, >=, <=	Relační operátory

Poznámka

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

Příklad: Příkaz IF

```
If (otto==10) and (anna==20) goto end
```


2.8 Možné převádění typů

Funkce

Konverze typu při přiřazení

Konstantní číselná hodnota, proměnná nebo výraz, který je přiřazen proměnné, musí být kompatibilní s tímto typem proměnné. Je-li tomu tak, je při přiřazení typ automaticky převáděn.

Možné konverze typu

na	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
z							
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

Vysvětlení

- * Při konverzi z typu REAL na typ INT se desetinná část $\geq 0,5$ zaokrouhluje nahoru, jinak dolů (srov. s funkcí ROUND).
- ¹⁾ Hodnotě 0 odpovídá TRUE, hodnotě $\neq 0$ odpovídá FALSE
- ²⁾ Když hodnota leží v přípustném intervalu čísel
- ³⁾ Když jen jeden znak
- ⁴⁾ Délka řetězce 0 = FALSE, jinak TRUE

Poznámka

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

Objeví-li se v jednom výrazu smíšené typy, tak je automaticky provedeno přizpůsobení typu. Převody datových typů jsou možné i v synchronních akcích, viz kapitola "Pohybové synchronní akce, implicitní konverze typů".

2.9 Operace s řetězcí

Operace s řetězcí

Vedle klasických operací "přiřazování" a "Porovnávání" je s řetězcí možno provádět ještě i následující operace:

- Převod do datového typu STRING (AXSTRING) (Strana 86)
- Převod z datového typu STRING (NUMBER, ISNUMBER, AXNAME) (Strana 87)
- Zřetěžení proměnných typu String (<<) (Strana 88)
- Převádění na malá/velká písmena (TOLOWER, TOUPPER) (Strana 89)
- Zjištění délky řetězce (STRLEN) (Strana 90)
- Vyhledávání znaků/řetězců v řetězci (INDEX, RINDEX, MINDEX, MATCH) (Strana 91)
- Vybírání dílčího řetězce (SUBSTR) (Strana 92)
- Načítání a zápis jednotlivých znaků (Strana 93)
- Formátování řetězce (SPRINT) (Strana 94)

Zvláštní význam znaku 0

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

Příklad:

Programový kód	Komentář
DEF STRING[20] STRG="Achse . steht"	
STRG[6]="X"	
MSG(STRG)	; výpis hlášení "Achse X steht"
STRG[6]=0	
MSG(STRG)	; výpis hlášení "Achse"

2.9.1 Převod do datového typu STRING (AXSTRING)

Pomocí funkce "Převod do datového typu STRING" se dají využít proměnné rozdílných typů jako součást jednoho hlášení (MSG).

Proběhne při použití operátoru << implicitně pro všechna data typu INT, REAL, CHAR a BOOL (viz "Zřetěžení proměnných typu String (<<) (Strana 88) ").

Hodnota typu INT bude převedena do normálně čitelné formy. U hodnot typu REAL bude uvedeno až 10 míst za čárkou.

Pomocí příkazu AXSTRING mohou být na proměnné typu STRING převáděny také proměnné typu AXIS.

Syntaxe

```
<STRING_ERG> = << <libovolný typ>
<STRING_ERG> = AXSTRING(<identifikátor osy>)
```

Význam

<STRING_ERG>:	Proměnná pro výsledek převodu typu	
	Typ:	STRING
<libovolný typ>:	Proměnné typu INT, REAL, CHAR, STRING a BOOL	
AXSTRING:	Výsledkem příkazu AXSTRING je specifikovaný identifikátor osy jako řetězec.	
<identifikátor osy>:	Proměnná pro identifikátor osy	
	Typ:	AXIS

Poznámka

Proměnné typu FRAME konvertovat nelze.

2.9.2**Převod z datového typu STRING (NUMBER, ISNUMBER, AXNAME)**

Prostřednictvím funkce NUMBER se provádí konverze z typu STRING do typu REAL. To, zda je konverze vůbec možná, lze zkontrolovat pomocí příkazu ISNUMBER.

Pomocí příkazu AXNAME je možné proměnnou typu STRING převádět na datový typ AXIS.

Syntaxe

```
<REAL_ERG>=NUMBER("<řetězec>")
<BOOL_ERG>=ISNUMBER("<řetězec>")
<AXIS_ERG>=AXNAME("<řetězec>")
```

Význam

NUMBER:	Výsledkem příkazu NUMBER je řetězcem <řetězec> představované číslo uložené jako reálná hodnota.	
<řetězec>:	Proměnná typu STRING, která má být konvertována	
<REAL_ERG>:	Proměnná pro výsledek převodu typu pomocí příkazu NUMBER	
	Typ:	REAL
ISNUMBER:	Pomocí příkazu ISNUMBER je možné zkontrolovat, může-li být řetězec <řetězec> převeden na platné číslo.	

<BOOL_ERG>:	Proměnná pro výsledek dotazu pomocí příkazu ISNUMBER		
	Typ:	BOOL	
	Hodnota:	TRUE	Výsledkem funkce ISNUMBER je hodnota TRUE, když řetězec <řetězec> představuje podle pravidel jazyka platné reálné číslo.
		FALSE	Jestliže je výsledkem funkce ISNUMBER hodnota FALSE, je při vyvolání funkce NUMBER se stejným řetězcem <řetězec> spuštěn alarm.
AXNAME:	Funkce AXNAME převádí udaný řetězec <řetězec> na identifikátor osy. Upozornění: Pokud řetězec <řetězec> není možné přiřadit žádnému z identifikátorů os nastavených v konfiguraci, aktivuje se alarm.		
<AXIS_ERG>:	Proměnná pro výsledek převodu typu pomocí příkazu AXNAME		
	Typ:	AXIS	

Příklad:

Programový kód	Komentář
DEF BOOL BOOL_ERG	
DEF REAL REAL_ERG	
DEF AXIS AXIS_ERG	
BOOL_ERG=ISNUMBER("1234.9876Ex-7")	; BOOL_ERG == TRUE
BOOL_ERG=ISNUMBER("1234XYZ")	; BOOL_ERG == FALSE
REAL_ERG=NUMBER("1234.9876Ex-7")	; REAL_ERG == 1234.9876Ex-7
AXIS_ERG=AXNAME("X")	; AXIS_ERG == X

2.9.3 Zřetězení proměnných typu String (<<)

Funkce "Zřetězení proměnných typu String" poskytuje možnost skládat řetězec z jednotlivých součástí.

Řetězení je realizováno prostřednictvím operátoru "<<". Dieser Operator hat für alle Kombinationen der Basistypen CHAR, BOOL, INT, REAL und STRING als Zieltyp STRING. Případně nutná konverze je provedena dle stávajících pravidel.

Syntaxe

```
<libovolný typ> << <libovolný typ>
```

Význam

<libovolný typ>:	Proměnné typu CHAR, BOOL, INT, REAL nebo STRING
<< :	Operátor pro zřetězení proměnných (<libovolný typ>) do takto poskládaného řetězce znaků (typ STRING). Tento operátor je k dispozici i samostatně jako tzv. „unární“ varianta. Tak je možné provést explicitní změnu do typu STRING (ne pro typy FRAME a AXIS): << <libovolný typ>

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)
```

Poznámka

Mezivýsledky při zřetězení typu string nesmí překročit maximální délku řetězce.

Poznámka

Typy FRAME a AXIS nemohou být použity spolu s operátorem "<<".

Příklady

Příklad 1: Zřetězení proměnných typu String

Programový kód	Komentář
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:"<<VALUE)	; Výpis: "Index:2/Wert:9.654"
NO_MSG:	

Příklad 2: Explicitní konverze typu pomocí příkazu <<

Programový kód	Komentář
DEF REAL VALUE=3.5	
<<VALUE	; Uvedená proměnná typu REAL je konvertována na typ STRING.

2.9.4 Převádění na malá/velká písmena (TOLOWER, TOUPPER)

Funkce "Převádění na malá/velká písmena" umožňuje změnit všechna písmena řetězce znaků do jednotné formy.

Syntaxe

```
<STRING_ERG>=TOUPPER("<řetězec>")
<STRING_ERG>=TOLOWER("<řetězec>")
```

Význam

TOUPPER:	Prostřednictvím příkazu TOUPPER jsou všechna písmena v řetězci znaků převedena na velká písmena.	
TOLOWER:	Prostřednictvím příkazu TOLOWER jsou všechna písmena v řetězci znaků převedena na malá písmena.	
<řetězec>:	Řetězec znaků, který má být převeden.	
	Typ:	STRING
<STRING_ERG>:	Proměnná pro výsledek převodu	
	Typ:	STRING

Příklad:

Protože uživatelské vstupy mohou být zobrazovány také na uživatelském rozhraní, lze jim dát standardní formu (velká nebo malá písmena):

Programový kód

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

2.9.5 Zjištění délky řetězce (STRLEN)

Prostřednictvím příkazu STRLEN můžete zjistit délku řetězce znaků.

Syntaxe

```
<INT_ERG>=STRLEN("<STRING>")
```

Význam

STRLEN:	Prostřednictvím příkazu STRLEN se určuje délka zadaného řetězce znaků. Výsledkem bude počet znaků, které – počítáno od začátku řetězce znaků – nejsou znakem 0.	
<řetězec>:	Řetězec znaků, jehož délka má být určena.	
	Typ:	STRING
<INT_ERG>:	Proměnná pro výsledek určování	
	Typ:	INT

Příklad:

Funkce v souvislosti s přístupem k jednotlivým znakům umožňuje stanovit konec řetězce znaků.

Programový kód

```
IF (STRLEN (BAUSTEIN_NAME) > 10) GOTO FFEHLER
```

2.9.6 Vyhledávání znaků/řetězců v řetězci (INDEX, RINDEX, MINDEX, MATCH)

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.

Syntaxe

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 vrácena hodnota -1. První znak má přitom pozici 0.

Význam

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, např. podle pozic se znakem mezery nebo s oddělovacím znakem („/") v udání cesty.

Příklad:**Rozložení zadání na název cesty a název modulu**

Programový kód	Komentář
DEF INT PFADIDX, PROGIDX	
DEF STRING[26] EINGABE	
DEF INT LISTIDX	

Programový kód	Komentář
EINGABE = "/_N_MPF_DIR/_N_EXECU- TE_MPF"	
LISTIDX = MINDEX (EINGABE, "M,N,O,P") + 1	; jako hodnota v proměnné LISTIDX je vráceno 3, protože "N" je první znak od předu v parametru EINGABE ze seznamu možností
PFADIDX = INDEX (EINGABE, "/") +1	; přitom tedy platí: PFADIDX = 1
PROGIDX = RINDEX (EINGABE, "/") +1	; přitom tedy platí: PROGIDX = 12
	; s pomocí v další části uvedené funkce SUBSTR se dá proměnná EINGABE rozložit na součásti "cesta" a "modul":
VARIABLE = SUBSTR (EINGABE, PFADIDX, PROGIDX-PFADIDX-1)	; výsledkem pak je "_N_MPF_DIR"
VARIABLE = SUBSTR (EINGABE, PROGIDX)	; výsledkem pak je "_N_EXECUTE_MPF"

2.9.7 Vybírání dílčího řetězce (SUBSTR)

Pomocí funkce SUBSTRING je možné načíst libovolný úsek v rámci daného řetězce.

Syntaxe

<STRING_ERG>=SUBSTR(<řetězec>,<index>,<délka>)

<STRING_ERG>=SUBSTR(<řetězec>,<index>)

Význam

SUBSTR:	Funkce poskytuje z daného <řetězce> určitý dílčí řetězec, který začíná místem daným parametrem <index> a má délku stanovenou parametrem <délka>. Pokud parametr <délka> není uveden, vrací tato funkce dílčí řetězec, který začíná na místě daném parametrem <index> a končí posledním znakem původního řetězce.
<Index>:	Počáteční místo dílčího řetězce v původním řetězci. Leží-li počáteční pozice za koncem řetězce, je převzat prázdný (" ") řetězec. První znak řetězce: Index = 0 Rozsah hodnot: 0 ... (délka řetězce -1)
<délka>:	Délka dílčího řetězce. Pokud je zadána příliš velká délka, bude výsledkem funkce dílčí řetězec až do konce původního řetězce. Rozsah hodnot: 1 ... (délka řetězce -1)

Příklad:

Programový kód	Komentář
DEF STRING [29] ERG	
;	1
;	0123456789012345678
ERG = SUBSTR ("QUITUNG: 10 bis 99", 10, 2)	; ERG == "10"
ERG = SUBSTR ("QUITUNG: 10 bis 99", 10)	; ERG == "10 až 99"

2.9.8 Načítání a zápis jednotlivých znaků

Jednotlivé znaky v rámci určitého řetězce je možné číst i zapisovat.

Přitom je nutno mít na paměti následující okrajové podmínky:

- Je možné pracovat pouze s proměnnými definovanými uživatelem, nikoli se systémovými proměnnými.
- Jednotlivé znaky řetězce jsou předávány při volání podprogramu metodou "call by value" (volání hodnotou).

Syntaxe

```
<znak>=<řetězec>[<index>]
<znak>=<řetězec_pole>[<pole_index>,<index>]
<řetězec>[<index>]=<znak>
<řetězec_pole>[<pole_index>,<index>]=<znak>
```

Význam

<řetězec>:	Libovolný řetězec
<znak>:	Proměnná typu CHAR.
<Index>:	Poloha znaku v rámci řetězce První znak řetězce: Index = 0 Rozsah hodnot: 0 ... (délka řetězce -1)

Příklady

Příklad 1: Proměnné hlášení

Programový kód	Komentář
; 0123456789	
DEF STRING [50] MELDUNG = "Osa n dosahla pozice"	
MELDUNG [6] = "X"	
MSG (MELDUNG)	; „osa X dosáhla pozice"

Příklad 2: Vyhodnocování systémových proměnných

Programový kód	Komentář
DEF STRING [50] STRG	; vyrovnávací paměť pro systémové proměnné
...	
STRG = \$P_MMCA	; načítání systémové proměnné
IF STRG[0] == "E" GOTO ...	; vyhodnocování systémové proměnné

Příklad 3: Předávání parametru metodou "call by value" a "call by reference"

Programový kód	Komentář
; 0123456	
DEF STRING[50] STRG = "Achse X"	

Programový kód	Komentář
DEF CHAR CHR	
...	
EXTERN UP_VAL (ACHSE)	; definice podprogramu s parametrem typu "call by value"
EXTERN UP_REF (VAR ACHSE)	; definice podprogramu s parametrem typu "call by ref." Parametry
...	
UP_VAL (STRG[6])	; předávání parametru metodou "by value"
...	
CHR = STRG[6]	; uložení do vyrovnávací paměti
UP_REF (CHR)	; předávání parametru metodou "by reference"

2.9.9 Formátování řetězce (SPRINT)

Prostřednictvím předdefinované funkce SPRINT mohou být formátovány řetězce znaků a mohou být tak připravovány např. pro výstup na externím zařízení (viz také "Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE) (Strana 677)").

Syntaxe

```
"<Výsledek_řetězec>"=SPRINT("<Formát_řetězce>",<Hodnota_1>,<Hodnota_2>,...,<Hodnota_n>)
```

Význam

SPRINT:	Identifikátor předdefinované funkce, jejímž výsledkem je hodnota typu STRING.
"<Formát_řetězce>":	Řetězec znaků, který obsahuje pevné a proměnné složky. Proměnné složky jsou definovány pomocí řídicího formátovacího znaku % a následným popisem formátu.
< hodnota_1>,< hodnota_2>,...,< hodnota_n>:	Hodnota ve formě konstanty nebo proměnné NC systému, která se vkládá na místo, na němž se nachází n-tý řídicí formátovací znak v souladu s popisem formátu v parametru <Formát_řetězce>.
"<Výsledek_řetězec>":	Formátovaný řetězec znaků (maximálně 400 bytů)

Popisy formátů, které jsou k dispozici

%B:	Přeměna na řetězec "TRUE", pokud se převáděná hodnota: • nerovná 0. • nerovná mezeře (v případě hodnot typu STRING). Přeměna na řetězec "FALSE", pokud se převáděná hodnota: • rovná 0. • je mezerou. Příklad: N10 DEF BOOL BOOL_VAR=1 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF BOOL_VAR:TRUE".
%C:	Přeměna na řetězec ASCII znaků. Příklad: N10 DEF CHAR CHAR_VAR="X" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C", CHAR_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF CHAR_VAR:X".
%D:	Přeměna na řetězec s celočíselnou hodnotou (INTEGER). Příklad: N10 DEF INT INT_VAR=123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D", INT_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF INT_VAR:123".
%<m>D:	Přeměna na řetězec s celočíselnou hodnotou (INTEGER). Řetězec má minimální délku <m> znaků. Chybějící místa jsou zleva doplněna mezerami. Příklad: N10 DEF INT INT_VAR=-123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D", INT_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF INT_VAR:xx-123" ("x" v příkladu reprezentuje znak "mezera").
%F:	Přeměna na řetězec s desetinným číslem, které má 6 míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Příklad: N10 DEF REAL REAL_VAR=-1.2341234EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F", REAL_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: -1234.123400".

%<m>F:	Přeměna na řetězec s desetinným číslem, které má 6 míst za desetinnou tečkou a celkovou délku minimálně <m> znaků. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Znaky, které chybějí, aby bylo dosaženo celkové délky <m>, jsou zleva doplněna mezerami. Příklad: <pre>N10 DEF REAL REAL_VAR=-1.23412345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: xxx-1234.123457" ("x" v příkladu reprezentuje znak "mezer").
%.<n>F:	Přeměna na řetězec s desetinným číslem, které má <n> míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Příklad: <pre>N10 DEF REAL REAL_VAR=-1.2345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: -1234.568".
%<m>.<n>F:	Přeměna na řetězec s desetinným číslem, které má <n> míst za desetinnou tečkou a celkovou délku minimálně <m> znaků. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Znaky, které chybějí, aby bylo dosaženo celkové délky <m>, jsou zleva doplněna mezerami. Příklad: <pre>N10 DEF REAL REAL_VAR=-1.2341234567890EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xx-1234.12" ("x" v příkladu reprezentuje znak "mezer").
%E:	Přeměna na řetězec s desetinným číslem v exponenciálním způsobu zápisu. Mantisa je normalizována na jedno místo před desetinnou tečkou a 6 míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo. Příklad: <pre>N10 DEF REAL REAL_VAR=-1234.567890 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:-1.234568EX+03".
%<m>E:	Přeměna na řetězec s desetinným číslem v exponenciálním formátu, který má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami. Mantisa je normalizována na jedno místo před desetinnou tečkou a 6 míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo. Příklad: <pre>N10 DEF REAL REAL_VAR=-1234.5 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" ("x" v příkladu reprezentuje znak "mezer").

%.<n>E:	Přeměna na řetězec s desetinným číslem v exponenciálním způsobu zápisu. Mantisa je normalizována na jedno místo před desetinnou tečkou a <n> míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo. Příklad: <pre>N10 DEF REAL REAL_VAR=-1234.5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:-1.23EX+03".
%<m>.<n>E:	Přeměna na řetězec s desetinným číslem v exponenciálním formátu, který má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami. Mantisa je normalizována na jedno místo před desetinnou tečkou a <n> míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo. Příklad: <pre>N10 DEF REAL REAL_VAR=-1234.5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xx-1.23EX+03" ("x" v příkladu reprezentuje znak "mezera").
%G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot: Pokud je zobrazovaná hodnota menší než 1.0EX-04 nebo větší/rovna 1.0EX+06, zvolí se exponenciální způsob zápisu, jinak se použije formát desetinného čísla. Vypisuje se maximálně šest významových míst, v případě nutnosti se číslo zaokrouhlí. Příklad s formátem desetinného čísla: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: 0.000123457". Příklad s exponenciálním způsobem zápisu: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:1.23457EX+06".

%<m>G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot (jako %G). Řetězec má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami. Příklad s formátem desetinného čísla: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxxx0.000123457" ("x" v příkladu reprezentuje znak "mezera"). Příklad s exponenciálním způsobem zápisu: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxx1.23457EX+06" ("x" v příkladu reprezentuje znak "mezera").
%.<n>G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot. Vypisuje se maximálně <n> významových míst, v případě nutnosti se číslo zaokrouhlí. Pokud je zobrazovaná hodnota menší než 1.0EX-04 nebo větší/rovna 1.0EX(+<n>), zvolí se exponenciální způsob zápisu, jinak se použije formát desetinného čísla. Příklad s formátem desetinného čísla: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: 0.000123". Příklad s exponenciálním způsobem zápisu: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+03 N20 DEF STRING[80] RESULT N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:1.23EX+03".
%<m>.<n>G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot (jako %.<n>G). Řetězec má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami. Příklad s formátem desetinného čísla: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxx0.0001235" ("x" v příkladu reprezentuje znak "mezera"). Příklad s exponenciálním způsobem zápisu: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xx1.235EX+06" ("x" v příkladu reprezentuje znak "mezera").

% .<n>P:	Přeměna hodnoty typu REAL na hodnotu typu INTEGER, přičemž se bere v úvahu <n> míst za desetinnou tečkou. Výsledná hodnota typu INTEGER je reprezentována 32-bitovým binárním číslem. Pokud převáděná hodnota nemůže být zobrazena pomocí 32 bitů, operace se přeruší a aktivuje se alarm. Protože posloupnost bytů generovaná formátovacím příkazem % .<n>P může obsahovat také binární nuly, celkový vytvořený řetězec již neodpovídá konvencím pro datový typ STRING NC-systému. Proto jej není možné ukládat do proměnných typu STRING a ani jej nelze dále zpracovávat pomocí příkazů NC jazyka, které jsou určeny pro řetězce. Jediným možným použitím je předávání parametrů do příkazu WRITE s výstupem na odpovídající externí zařízení (viz následující příklad). Jestliže parametr <Formát_řetězce> obsahuje popis formátu typu %P, bude na výstup odeslán celý řetězec, s výjimkou binárního čísla generovaného pomocí parametru % .<n>P, v závislosti na nastavení parametru MD10750 \$MN_SPRINT_FORMAT_P_CODE ve znakovém kódu ASCII, ISO (DIN 6024) nebo EIA (RS 244). Jestliže je naprogramován znak, který nemůže být převeden, bude zpracování přerušeno a aktivuje se alarm. Příklad: <pre> N10 DEF REAL REAL_VAR=123.45 N20 DEF INT ERROR N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1" ... N100 EXTOPEN(ERROR,EXT_DEVICE) N110 IF ERROR <> 0 ... ; zpracování chyby N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%. 3P",REAL_VAR) N210 IF ERROR <> 0 ... ; zpracování chyby </pre> Výsledek: Řetězec "INTEGER BINARY CODED: 'H0001E23A'" se přenese na výstupní zařízení /ext/dev/1. Hexadecimální hodnota 0x0001E23A odpovídá decimální hodnotě 123450.
----------	--

%<m>.<n>P:	Převod hodnoty typu REAL na řetězec v souladu s nastavením strojního parametru MD10751 \$MN_SPRINT_FORMAT_P_DECIMAL, přičemž tímto řetězcem je: - celé číslo s <m> + <n> místy nebo - desetinné číslo s maximální <m> místy před desetinnou tečkou a přesně <n> místy za desetinnou tečkou. Stejně jako v případě popisu formátu %.<n>P se celý řetězec ukládá ve znakovém kódu definovaném v parametru MD10750 \$MN_SPRINT_FORMAT_P_CODE. Přeměna v případě MD10751 = 0: Hodnota typu REAL je převedena na řetězec s celým číslem, které má <m> + <n> míst. V případě potřeby jsou místa za desetinnou tečkou zaokrouhlena na <n> míst nebo doplněna 0. Chybějící místa před desetinnou tečkou jsou doplněna mezerami. Znaménko mínus se připojuje z levé strany, místo znaménka plus se vkládá mezera. Příklad: <pre>N10 DEF REAL REAL_VAR=-123.45 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "PUNCHED TAPE FORMAT:-xx123450" ("x" v příkladu reprezentuje znak "mezera"). Přeměna v případě MD10751 = 1: Hodnota typu REAL je převedena na řetězec s desetinným číslem, které má maximálně <m> míst před desetinnou tečkou a přesně <n> míst za desetinnou tečkou. V případě potřeby jsou místa před desetinnou tečkou oříznuta a místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Jestliže se <n> rovná 0, odpadá i desetinná tečka. Příklad: <pre>N10 DEF REAL REAL_VAR1=-123.45 N20 DEF REAL REAL_VAR2=123.45 N30 DEF STRING[80] RESULT N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "PUNCHED TAPE FORMAT:-123.450 VAR2:23".
%S:	Slouží pro vkládání řetězce. Příklad: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:ABCDEFGH".
%<m>S:	Vložení řetězce s minimálně <m> znaky. Chybějící místa jsou doplněna mezerami. Příklad: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:xxxABCDEFGH" ("x" v příkladu reprezentuje znak "mezera").

%.<n>S:	Vložení <n> znaků řetězce (počínaje prvním znakem). Příklad: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S", STRING_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:ABC".
%<m>.<n>S:	Vložení <n> znaků řetězce (počínaje prvním znakem). Celková délka sestavovaného řetězce je minimálně <m> znaků. Chybějící místa jsou doplněna mezerami. Příklad: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:xxxABCDE" ("x" v příkladu reprezentuje znak "mezeru").
%X:	Převod hodnoty typu INTEGER na řetězec v hexadecimálním způsobu zápisu. Příklad: <pre>N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X", INT_VAR)</pre> Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "INTEGER HEXADECIMAL:A5B8".

Poznámka

Konvence jazyka NC systému týkající se rozlišování velkých a malých písmen v názvech identifikátorů a v klíčových slovech platí také pro popisy formátů. Při programování proto můžete používat velká i malá písmena, aniž by to mělo na funkce nějaký vliv.

Možnosti kombinování

V následující tabulce naleznete přehled o tom, které datové typy NC systémů a které popisy formátování mohou být kombinovány. Platí také pravidla pro implicitní převody datových typů (viz "Datové typy (Strana 58)").

	Datové typy NC systému						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

Poznámka

Tato tabulka ukazuje, že NC datové typy AXIS a FRAME není možné pomocí funkce SPRINT převádět přímo. Je ale možné:

- Datový typ AXIS je možné převádět na řetězec pomocí funkce AXSTRING, který je potom možné dále zpracovávat pomocí funkce SPRINT.
 - Jednotlivé hodnoty datového typu FRAME je možné načítat pomocí funkcí pro přístup k jednotlivým složkám framu. Tímto způsobem získáte data typu REAL, která je pak možné pomocí příkazu SPRINT dále zpracovávat.
-

2.10 Programové skoky a větvení

2.10.1 Skok zpátky na začátek programu (GOTOS)

Pomocí příkazu `GOTOS` je možné skočit zpět na začátek hlavního programu nebo podprogramu za účelem jeho opakovaného zpracování.

Prostřednictvím strojních parametrů je možné nastavit, aby se při každém skoku zpět na začátek programu uskutečňovaly následující operace:

- Nastavení doby zpracovávání programu na "0".
- Zvýšení hodnoty počítadla obrobků o "1".

Syntaxe

`GOTOS`

Význam

GOTOS:	Příkaz skoku, kdy cílem skoku je začátek programu.	
	Vlastní realizace je ovládána signálem rozhraní NC/PLC: DB21, ... DBX384.0 (řízení větvení programu)	
	Hodnota:	Význam:
	0	Žádný skok zpět na začátek programu. Zpracovávání programu bude pokračovat následujícím blokem výrobního programu za příkazem <code>GOTOS</code> .
	1	Skok zpět na začátek programu. Výrobní program je opakován.

Okrajové podmínky

- Příkaz `GOTOS` interně spouští `STOPRE` (zastavení předběžného zpracování).
- U výrobních programů v definici dat (proměnné LUD) se skok s příkazem `GOTOS` uskutečňuje na první programový blok za úsekem definic, tzn. definice dat se znovu neprovádějí. Definované proměnné si proto ponechávají hodnotu, kterou měly v bloku s příkazem `GOTOS` a nejsou jim tedy dosazeny standardní hodnoty naprogramované v úseku definic.
- V synchronních akcích a v technologických cyklech není příkaz `GOTOS` k dispozici.

Příklad:

Programový kód	Komentář
N10 ...	; začátek programu
...	
N90 GOTOS	; skok na začátek programu
...	

2.10.2 Programové skoky na návěští skoků (GOTOB, GOTOF, GOTO, GOTOC)

Do programu je možno vkládat návěští skoků (label), na která lze potom skákat z jiných míst téhož programu pomocí příkazů GOTOF, GOTOB, GOTO, příp. GOTOC. Zpracování programu potom pokračuje příkazem, který se nachází bezprostředně za návěštím skoku. Tímto způsobem je možné realizovat větvení v rámci programu.

Kromě návěští skoků je možno jako cíl skoku používat také čísla hlavních a vedlejších bloků.

Jestliže je před příkazem kroku formulována podmínka skoku (IF . . .), potom se programový skok uskuteční jen tehdy, pokud je podmínka skoku splněna.

Syntaxe

```
GOTOB <cíl skoku>
IF <podmínka skoku> == TRUE GOTOB <cíl skoku>
```

```
GOTOF <cíl skoku>
IF <podmínka skoku> == TRUE GOTOF <cíl skoku>
```

```
GOTO <cíl skoku>
IF <podmínka skoku> == TRUE GOTO <cíl skoku>
```

```
GOTOC <cíl skoku>
IF <podmínka skoku> == TRUE GOTOC <cíl skoku>
```

Význam

GOTOB:	Příkaz skoku s cílem skoku hledaným směrem k začátku programu	
GOTOF:	Příkaz skoku s cílem skoku hledaným směrem ke konci programu.	
GOTO:	Příkaz skoku s vyhledáváním cíle skoku. Vyhledávání se uskutečňuje napřed směrem ke konci programu, potom směrem k začátku programu.	
GOTOC:	Tento příkaz funguje stejně jako příkaz GOTO jen s tím rozdílem, že je potlačen alarm 14080 "Cíl skoku nenalezen". To znamená, že se zpracovávání programu v případě neúspěšného vyhledávání cíle skoku nepřerušuje, nýbrž bude pokračovat na programovém řádku, který se nachází za příkazem GOTOC.	
<cíl skoku>:	Parametr cíle skoku Možné údaje jsou následující:	
	<návěští skoku>:	Cíl skoku je návěští skoku umístěné v programu, jehož název je definován uživatelem: <návěští skoku>:
	<číslo bloku>:	Cíl skoku je číslo hlavního či vedlejšího bloku (např.: 200, N300)
	Proměnná typu STRING:	Proměnný cíl skoku. Proměnná představuje návěští skoku nebo číslo bloku.
IF:	Klíčové slovo pro formulování podmínky skoku. Podmínka skoku připouští všechny porovnávací a logické operátory (výsledek: TRUE nebo FALSE). Programový skok se uskuteční, jestliže výsledek této operace je TRUE.	

Poznámka**Návěští skoku (label)**

Návěští skoku se vždy nachází na začátku bloku. Jestliže existuje také číslo programu, návěští skoku se zapisuje vždy bezprostředně za číslem bloku.

Pro určování názvů návěští skoků platí následující pravidla:

- Počet znaků:
 - minimálně 2
 - maximálně 32
- Povolenými znaky jsou:
 - Písmena
 - Číslice
 - Znak podtržení
- První dva znaky musí být písmena nebo znak podtržení.
- Za názvem návěští skoku následuje dvojtečka (":").

Okrajové podmínky

- Cílem skoku může být pouze blok s návěštím skoku nebo s číslem bloku, které se nacházejí **uvnitř** daného programu.
- Příkaz skoku bez podmínky skoku musí být naprogramován v samostatném bloku. U příkazů skoku s podmínkou skoku toto omezení neplatí. V jednom bloku může být formulováno několik příkazů skoků.
- U programů s příkazy skoků bez podmínek skoků se může stát, že se konec programu M2/ M30 nebude nacházet na konci programu.

Příklady**Příklad 1: Skoky na návěští skoku**

Programový kód	Komentář
N10 ...	
N20 GOTOF Label_1	; skok ve směru konce programu na ; návěští skoku "Label_1".
N30 ...	
N40 Label_0: R1=R2+R3	; návěští skoku "Label_0" nastaveno
N50 ...	
N60 Label_1:	; návěští skoku "Label_1" nastaveno
N70 ...	
N80 GOTOB Label_0	; skok ve směru začátku programu ; na návěští skoku "Label_0".
N90 ...	

Příklad 2: Nepřímý skok na číslo bloku

Programový kód	Komentář
IF <podmínka> == TRUE	
R10=100	; přiřazení cíle skoku
ELSE	
R10=110	; přiřazení cíle skoku
ENDIF	
; skok ve směru konce programu na blok, jehož číslo bloku je uloženo v R10	
N10 GOTOF "N"<<R10	
...	
N90 ...	
N100 ...	; cíl skoku
N110 ...	
...	

Příklad 3: Skok na proměnný cíl skoku

Programový kód	Komentář
DEF STRING[20] ZIEL	
IF <podmínka> == TRUE	
ZIEL = „Marke1“	; přiřazení cíle skoku
ELSE	
ZIEL = "Marke2"	; přiřazení cíle skoku
ENDIF	
; skok ve směru konce programu na proměnný cíl skoku, která je v proměnné „ZIEL“	
GOTOF ZIEL	
Marke1: T="Bohrer1"	; cíl skoku 1
...	
Marke2: T="Bohrer2"	; cíl skoku 2
...	

Příklad 4: Skok s podmínkou skoku

Programový kód	Komentář
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; přiřazení počátečních hodnot
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; návěští skoku LA1
N42 R1=R1+R3 R4=R4-1	
; IF podmínka skoku == TRUE	
; THEN skok ve směru začátku programu na návěští skoku LA1	
N43 IF R4>0 GOTOB LA1	
N44 M30	; konec programu

2.10.3 Větvení programu (CASE ... OF ... DEFAULT ...)

Funkce CASE nabízí možnost překontrolovat aktuální hodnotu (typ: INT) nějaké proměnné nebo matematické funkce a v závislosti na výsledku skočit v programu na různá místa.

Syntaxe

```
CASE(<výraz>) OF <konstanta_1> GOTOF <cíl skoku_1> <konstanta_2>
GOTOF <cíl skoku_2> ... DEFAULT GOTOF <cíl skoku_n>
```

Význam

CASE:	Příkaz skoku	
<výraz>:	Proměnná nebo matematická funkce	
OF:	Klíčové slovo pro formulování podmíněného programového větvení.	
<konstanta_1>:	První zadaná konstantní hodnota pro proměnnou nebo matematickou funkci	
	Typ:	INT
<konstanta_2>:	Druhá zadaná konstantní hodnota pro proměnnou nebo matematickou funkci	
	Typ:	INT
DEFAULT:	Pro případy, při kterých se hodnota proměnné nebo matematické funkce nerovná žádné z uvedených konstant, může být cíl skoku určen pomocí příkazu DEFAULT. Upozornění: Jestliže příkaz DEFAULT není naprogramován, použije se v těchto případech jako cíl skoku blok, který následuje za příkazem CASE.	
GOTOF:	Příkaz skoku s cílem skoku hledaným směrem ke konci programu. Namísto příkazu GOTOF je možné naprogramovat také všechny ostatní příkazy GOTO (viz kapitola "Programové skoky na návěští skoku").	
<cíl skoku_1>:	Na tento cíl skoku se odbočí, jestliže hodnota proměnné nebo matematické funkce odpovídá první ze zadaných konstant. Cíl skoku může být zadán následujícím způsobem:	
	<návěští skoku>:	Cíl skoku je návěští skoku umístěné v programu, jehož název je definován uživatelem: <návěští skoku>:
	<číslo bloku>:	Cíl skoku je číslo hlavního či vedlejšího bloku (např.: 200, N300)
	Proměnná typu STRING:	Proměnný cíl skoku. Proměnná představuje návěští skoku nebo číslo bloku.
<cíl skoku_2>:	Na tento cíl skoku se odbočí, jestliže hodnota proměnné nebo matematické funkce odpovídá druhé ze zadaných konstant.	
<cíl skoku_n>:	Na tento cíl skoku se odbočí, jestliže hodnota proměnné nebo matematické funkce neodpovídá žádné z uvedených konstant.	

Příklad:

Programový kód

```
...  
N20 DEF INT VAR1 VAR2 VAR3  
N30 CASE (VAR1+VAR2-VAR3) OF 7 GOTOF Label_1 9 GOTOF La-  
bel_2 DEFAULT GOTOF Label_3  
N40 Label_1: G0 X1 Y1  
N50 Label_2: G0 X2 Y2  
N60 Label_3: G0 X3 Y3  
...
```

Příkaz CASE z bloku N30 definuje následující možnosti rozvětvení programu:

1. Jestliže se hodnota matematické funkce $VAR1+VAR2-VAR3 = 7$, potom se skáče na blok s definicí návěští skoku "Label_1" (--> N40).
2. Jestliže se hodnota matematické funkce $VAR1+VAR2-VAR3 = 9$, potom se skáče na blok s definicí návěští skoku "Label_2" (--> N50).
3. Jestliže se hodnota matematické funkce $VAR1+VAR2-VAR3$ nerovná ani 7 ani 9, potom se skáče na blok s definicí návěští skoku "Label_3" (--> N60).

2.11 Opakování částí programu (REPEAT, REPEATB, ENDLABEL, P)

Opakování částí programu umožňuje opětovné zpracování již napsané části v rámci daného programu v libovolné podobě.

Řádky, příp. úsek programu, který se má opakovat, je označen návěštími skoku (label).

Poznámka

Návěští skoku (label)

Návěští skoku se vždy nachází na začátku bloku. Jestliže existuje také číslo programu, návěští skoku se zapisuje vždy bezprostředně za číslem bloku.

Pro určování názvů návěští skoků platí následující pravidla:

- Počet znaků:
 - minimálně 2
 - maximálně 32
- Povolenými znaky jsou:
 - Písmena
 - Číslice
 - Znak podtržení
- První dva znaky musí být písmena nebo znak podtržení.
- Za názvem návěští skoku následuje dvojtečka (":").

Syntaxe

1. Opakování jednoho jediného programového řádku:

```
<návěští skoku>: ...
...
REPEATB <návěští skoku> P=<n>
...
```

2. Opakování úseku programu mezi návěštím skoku a příkazem REPEAT:

```
<návěští skoku>: ...
...
REPEAT <návěští skoku> P=<n>
...
```

3. Opakování úseku mezi dvěma návěštlími skoku:

```
<počáteční návěští skoku>: ...
...
<koncové návěští skoku>: ...
...
REPEAT <počáteční návěští skoku> <koncové návěští skoku> P=<n>
```

...

Poznámka

Není možné, aby se příkaz REPEAT nacházel mezi počátečním a koncovým návěštím. Jestliže se <počáteční návěští skoku> nachází před příkazem REPEAT a pokud <koncové návěští skoku> není dosaženo před příkazem REPEAT, potom bude se opakován úsek programu mezi <počátečním návěštím skoku> a příkazem REPEAT.

4. Opakování úseku mezi návěštím skoku a návěštím ENDLABEL:

```
<návěští skoku>: ...
...
ENDLABEL: ...
...
REPEAT <návěští skoku> P=<n>
...
```

Poznámka

Není možné, aby se příkaz REPEAT nacházel mezi <návěštím skoku> a návěštím ENDLABEL. Jestliže se <počáteční návěští skoku> nachází před příkazem REPEAT a pokud návěští ENDLABEL není dosaženo před příkazem REPEAT, potom bude se opakován úsek programu mezi <počátečním návěštím skoku> a příkazem REPEAT.

Význam

REPEATB:	Příkaz pro opakování řádku programu.
REPEAT:	Příkaz pro opakování úseku programu.
<návěští skoku>:	Řetězec <návěští skoku> označuje: • Řádek programu, který má být opakován (v případě příkazu REPEATB) nebo • Začátek úseku programu, který má být opakován (v případě příkazu REPEAT) Řádek programu označený jako <návěští skoku> se může nacházet před nebo za příkazem REPEAT / REPEATB. Hledání se napřed provádí směrem k začátku programu. Pokud návěští skoku není v tomto směru nalezeno, spustí se hledání směrem ke konci programu. Výjimka: Jestliže má být opakován úsek programu mezi návěštím skoku a příkazem REPEAT (viz bod 2. v odstavci Syntaxe), potom se musí být programový řádek označený jako <návěští skoku> nacházet před příkazem REPEAT, protože v tomto případě se vyhledávání uskutečňuje pouze ve směru začátku programu. Jestliže řádek s návěštím <návěští skoku> obsahuje další příkazy, budou tyto příkazy znovu prováděny při každém opakování.

2.11 Opakování částí programu (REPEAT, REPEATB, ENDLABEL, P)

ENDLABEL:	Klíčové slovo, které označuje konec úseku programu, který se má opakovat. Jestliže řádek s návěštím ENDLABEL obsahuje další příkazy, budou tyto příkazy znovu prováděny při každém opakování. Příkaz ENDLABEL může být v programu použit i vícekrát.	
P:	Adresa pro zadání počtu opakování	
<n>:	Počet opakování úseku programu	
	Typ:	INT
	Opakovaný úsek programu bude opakován <n>-krát. Po posledním opakování bude zpracování programu pokračovat řádkem, který následuje za příkazem REPEAT / REPEATB.	
	Upozornění: Jestliže příkaz P=<n> není uveden, bude opakování úseku programu opakován právě jedenkrát.	

Příklady

Příklad 1: Opakování jednoho jediného programového řádku

Programový kód	Komentář
N10 POSITION1: X10 Y20	
N20 POSITION2: CYCLE(0,,9,8)	; cyklus pro polohování
N30 ...	
N40 REPEATB POSITION1 P=5	; blok N10 provést pětkrát
N50 REPEATB POSITION2	; blok N20 provést jednou
N60 ...	
N70 M30	

Příklad 2: Opakování úseku programu mezi návěštím skoku a příkazem REPEAT

Programový kód	Komentář
N5 R10=15	
N10 Begin: R10=R10+1	; šířka
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	
N50 X=-R10	
N60 Y=-R10	
N70 Z=10+R10	
N80 REPEAT BEGIN P=4	; úsek programu N10 až N70 provést 4x
N90 Z10	
N100 M30	

Příklad 3: Opakování úseku mezi dvěma návěštimi skoku

Programový kód	Komentář
N5 R10=15	
N10 Begin: R10=R10+1	; šířka

2.11 Opakování části programu (REPEAT, REPEATB, ENDLABEL, P)

Programový kód	Komentář
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	
N50 X=-R10	
N60 Y=-R10	
N70 END: Z=10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	; úsek programu N10 až N70 provést 3x
N110 Z10	
N120 M30	

Příklad 4: Opakování úseku mezi návěštím skoku a návěštím ENDLABEL

Programový kód	Komentář
N10 G1 F300 Z-10	
N20 BEGIN1:	
N30 X10	
N40 Y10	
N50 BEGIN2:	
N60 X20	
N70 Y30	
N80 ENDLABEL: Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3: X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	; úsek programu N110 až N120 provést 3x
N140 REPEAT BEGIN2 P=2	; úsek programu N50 až N80 provést dvakrát
N150 M100	
N160 REPEAT BEGIN1 P=2	; úsek programu N20 až N80 provést dvakrát
N170 Z10	
N180 X0 Y0	
N190 M30	

Příklad 5: Obrábění frézováním, opracování pozic pro vrtání pomocí různých technologií

Programový kód	Komentář
N10 ZENTRIERBOHRER()	; výměna nástroje pro navrtávání středících důlků
N20 POS_1:	; pozice pro vrtání 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	

2.11 Opakování částí programu (REPEAT, REPEATB, ENDLABEL, P)

Programový kód	Komentář
N80 POS_2:	; pozice pro vrtání 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 BOHRER()	; výměna nástroje za vrták a cyklus vrtání
N140 GEWINDE(6)	; výměna závitníku M6 a cyklus pro vrtání závitu
N150 REPEAT POS_1	; opakování úseku programu od POS_1 do návěští ENDLABEL jedenkrát
N160 BOHRER()	; výměna nástroje za vrták a cyklus vrtání
N170 GEWINDE(8)	; výměna závitníku M8 a cyklus pro vrtání závitu
N180 REPEAT POS_2	; opakování úseku programu od POS_2 do návěští ENDLABEL jedenkrát
N190 M30	

Další informace

- Opakování části programu může být voláno i vnořeným způsobem. Každé takové volání obsazuje jednu úroveň podprogramů.
- Pokud je v průběhu zpracování opakování části programu naprogramován příkaz M17 nebo RET, opakování části programu se přeruší. Zpracování programu bude pokračovat blokem, který následuje za řádkem s příkazem REPEAT.
- V okně aktuálního programu se vypisuje opakování části programu jako vlastní úroveň podprogramu.
- Jestliže během zpracování části programu dojde ke zrušení úrovně, program bude pokračovat na místě za voláním opakování úseku programu.

Příklad:

Programový kód	Komentář
N5 R10=15	
N10 BEGIN: R10=R10+1	; šířka
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	; zrušení úrovně
N50 X=-R10	
N60 Y=-R10	
N70 END: Z10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	; zpracovávání programu bude pokračovat zde
N130 M30	

2.11 Opakování části programu (REPEAT, REPEATB, ENDLABEL, P)

- Řídící struktury a opakování částí programu mohou být používány kombinovaně. Neměly by se však mezi nimi vyskytovat žádná překrývající se místa. Opakování úseku programu by mělo být uvnitř větve řídící struktury, příp. řídící struktura by měla ležet uvnitř opakování úseku programu.
- V případě směřování skoků a opakování úseku programu jsou bloky zpracovávány čistě sekvenčně. Jestliže se vyskytuje např. skok z opakování části programu, bude se program tak dlouho zpracovávat, dokud nebude nalezen naprogramovaný konec úseku programu.
Příklad:

Programový kód

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL: Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

Poznámka

Příkaz REPEAT by se měl nacházet za bloky posuvu.

2.12 Řídící struktury

Řídící systém standardně zpracovává NC bloky v naprogramované posloupnosti.

Tato posloupnost může být naprogramováním alternativních programových bloků a programových smyček změněna. Pro programování těchto řídicích struktur se používají klíčová slova IF, ELSE, ENDIF, LOOP, FOR, WHILE a REPEAT.

UPOZORNĚNÍ

Chyba v programu

Řídící struktury je možné používat pouze v rámci příkazové části programu. Definice v hlavičce programu nemohou být prováděny ani podmíněně, ani opakovaně.

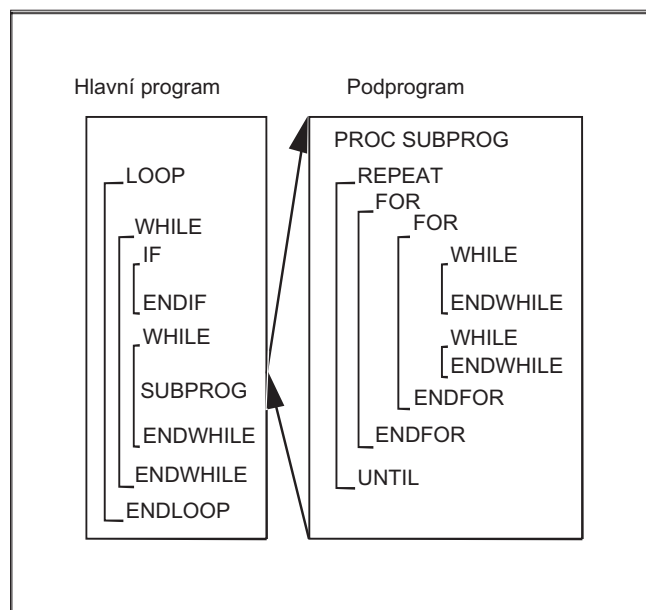
Klíčová slova pro řídicí struktury, stejně jako cíle skoku, nesmí být superponovány makry. Kontrola při definici maker se v této záležitosti neprovádí.

Platnost

Řídící strukturu není možné použít tak, aby přesahovala hranice programu.

Hloubka vnoření

V rámci každé úrovně podprogramu je možná hloubka vnoření až 16 řídicích struktur.



Chování doby zpracování

Při standardně aktivním režimu překladače je možné v důsledku použití programových skoků dosáhnout rychlejšího zpracování než s řídicími strukturami.

U předem přeložených cyklů není mezi programovými skoky a řídícími strukturami žádný rozdíl.

Výpis aktuálního bloku u programových smyček

Jestliže jsou v rámci programové smyčky zpracovávány pouze bloky předběžného zpracování, ve výpisu aktuálního bloku se bude zobrazovat poslední blok hlavní větve zpracováváný **před** programovou smyčkou.

Aby se např. pro potřeby diagnostiky objevovaly ve výpisu aktuálního bloku také bloky předběžného zpracování, musí být aktivován příkaz SBL2 pro dekodování jednotlivých bloků.

Literatura

Příručka k funkcím, Základní funkce, kapitola: BAG, Kanál, zpracování programu, chování při resetu (K1) > Jednotlivé bloky > Dekodování jednotlivých bloků SBL2 s implicitním zastavením předběžného zpracování

Broušení bez bloku v hlavní větvi programu

Jestliže není v rámci programové smyčky naprogramován žádný blok v hlavní větvi programu, bude smyčka zpracovávána v rámci předběžného zpracování tak dlouho, dokud nebude podmínka smyčky splněna.

Tímto způsobem vzniká vysoké zatížení systému a může být nepříznivě ovlivněno zobrazování dat.

Jako řešení se nabízí vložení příkazu STOPRE nebo doby prodlevy G04 o délce 0 sekund **dovnitř** smyčky.

Okrajové podmínky

- Bloky s prvky řídících struktur nemohou být přeskakovány.
- Návěští skoku (label) jsou v blocích s prvky řídících struktur nepřipustné.
- Řídící struktury jsou zpracovávány podle překladače. Když je rozpoznán konec smyčky, bude vyhledán začátek této smyčky a přitom se budou brát v úvahu další nalezené řídící struktury. Z tohoto důvodu není v režimu překladače zkontrolována bloková struktura programu úplně.
- V zásadě se doporučuje nepoužívat řídící struktury a programové skoky současně.
- Při přípravě cyklů může být správné vnoření řídících struktur zkontrolováno.

2.12.1 Podmíněné příkazy a větvení (IF, ELSE, ENDIF)

Podmíněný příkaz: IF - programový blok - ENDIF

V případě podmíněného příkazu je programový blok nacházející se mezi příkazy IF a ENDIF zpracován jen tehdy, pokud je podmínka splněna.

Větvení: IF - programový blok_1 - ELSE - programový blok_2 - ENDIF

V případě větvení se vždy zpracovává jeden ze dvou programových bloků.

Pokud je podmínka splněna, je zpracován programový blok 1 nacházející se mezi příkazy IF a ELSE.

Pokud podmínka splněna **není**, je zpracován programový blok 2 nacházející se mezi příkazy ELSE a ENDIF.

Syntaxe

Podmíněný příkaz

```
IF <podmínka>
    Programový blok                ; zpracovat, pokud: <podmínka> == TRUE
ENDIF
```

Větvení

```
IF <podmínka>
    Programový blok_1                ; zpracovat, pokud: <podmínka> == TRUE
ELSE
    Programový blok_2                ; zpracovat, pokud: <podmínka> == FALSE
ENDIF
```

Význam

IF:	Uvádí podmíněný příkaz, příp. větvení.
ELSE:	Uvádí alternativní programový blok.
ENDIF:	Označuje konec podmíněného příkazu, příp. větvení.
<podmínka>:	Logický výraz, z jehož vyhodnocení vyplývá buď TRUE nebo FALSE.

Příklad: Podprogram pro výměnu nástroje

Programový kód	Komentář
PROC L6	Rutina pro výměnu nástroje
N500 DEF INT TNR_AKTUELL	Proměnná pro aktivní T-číslo.
N510 DEF INT TNR_VORWAHL	Proměnná pro předem zvolené T-číslo.
	Zjištění aktuálního nástroje.
N520 STOPRE	
N530 IF \$P_ISTEST	V režimu testování programu se ...
N540 TNR_AKTUELL = \$P_TOOLNO	... z kontextu programu načte "aktuální" nástroj
N550 ELSE	V opačném případě se ...
N560 TNR_AKTUELL = \$TC_MPP6[9998,1]	... načte nástroj ve vřetenu
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	Načtení T-čísla předem zvoleného nástroje pro vřeteno.

Programový kód	Komentář
N590 IF TNR_AKTUELL <> TNR_VORWAHL	Pokud předem zvolený nástroj ještě není aktuálním nástrojem, potom ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... najíždění na bod pro výměnu nástroje ...
N610 M206	... a provedení výměny nástroje
N620 ENDIF	
N630 M17	

2.12.2 Nekonečná programová smyčka (LOOP, ENDLOOP)

Nekonečná smyčka nachází uplatnění v nekonečných programech. Na konci smyčky se vždy provádí skok zpět na její začátek.

Syntaxe

```
LOOP
...
ENDLOOP
```

Význam

LOOP:	Uvádí nekonečnou smyčku.
ENDLOOP:	Označuje konec smyčky a způsobuje návrat zpět na její začátek.

Příklad:

```
Programový kód
...
LOOP
MSG ("Žádný břit nástroje není aktivní")
M0
STOPRE
ENDLOOP
...
```

2.12.3 Smyčka s počítadlem (FOR ... TO ..., ENDFOR)

Smyčka s počítadlem se používá, jestliže má být opakovaně zpracován úsek programu a počet těchto opakování je předem znám.

Syntaxe

```
FOR <proměnná> = <počáteční hodnota> TO <konečná hodnota>
...
ENDFOR
```

Význam

FOR:	Uvádí smyčku s počítadlem.	
ENDFOR:	Označuje konec smyčky a způsobuje skok zpět na její začátek, a to tak dlouho, dokud není dosaženo koncové hodnoty počítadla.	
<proměnná>:	Počítající proměnná, která načítá nahoru od počáteční až po koncovou hodnotu a s každým průchodem se o hodnotu "1" zvyšuje.	
	Typ	INT nebo REAL Upozornění Typ REAL se použije, jestliže je jako počítadlo smyčky naprogramován např. R-parametr. Jestliže je jako počítadlo použita proměnná typu REAL, její hodnota se zaokrouhluje na celé číslo.
<počáteční hodnota>:	Počáteční hodnota pro počítání Podmínka: Počáteční hodnota musí být menší než hodnota konečná.	
<konečná hodnota>:	Konečná hodnota pro počítání	

Příklady

Příklad 1: Proměnná typu INTEGER nebo R-parametr jako proměnná počítadla

Proměnná typu INTEGER jako proměnná počítadla:

Programový kód	Komentář
DEF INT iVARIABLE1 R10=R12-R20*R1 R11=6 FOR iVARIABLE1= R10 TO R11 R20=R21*R22+R33 ENDFOR M30	; proměnná počítadla = proměnná typu INTEGER

R-Parametr jako proměnná počítadla:

Programový kód	Komentář
R11=6 FOR R10=R12-R20*R1 TO R11 R20=R21*R22+R33 ENDFOR M30	; proměnná počítadla = R-parametr (proměnná typu REAL)

Příklad 2: Výroba pevně daného počtu kusů

Programový kód	Komentář
DEF INT STUECKZAHL	; definice proměnné typu INT s názvem "STUECKZAHL"
FOR STUECKZAHL = 0 TO 100	; uvádí smyčku s počítadlem Proměnná "STUECKZAHL" se bude postupně zvyšovat od počáteční hodnoty "0" až po konečnou hodnotu "100".
G01 ...	
ENDFOR	; konec smyčky s počítadlem
M30	

2.12.4 Programová smyčka s podmínkou na začátku smyčky (WHILE, ENDWHILE)

U smyčky WHILE se podmínka nachází na začátku smyčky. Dokud je podmínka splněna, bude se smyčkou WHILE procházet.

Syntaxe

```
WHILE <podmínka>
...
ENDWHILE
```

Význam

WHILE:	Uvádí programovou smyčku.
ENDWHILE:	Označuje konec smyčky a způsobuje návrat zpět na její začátek.
<podmínka>:	Podmínka, která musí být splněna, aby se smyčkou WHILE procházelo.

Příklad:

Programový kód	Komentář
...	
WHILE \$AA_IW[BOHRACHSE] > -10	; vyvolání smyčky WHILE za následujících podmínek: Aktuální požadovaná hodnota ve WCS pro osu vrtáku musí být větší než -10.
G1 G91 F250 AX[BOHRACHSE] = -1	
ENDWHILE	
...	

2.12.5 Programová smyčka s podmínkou na konci smyčky (REPEAT, UNTIL)

U smyčky REPEAT se podmínka nachází na konci smyčky. Smyčka REPEAT je tedy jednou zpracována a potom je opakována tak dlouho, dokud je podmínka splněna.

Syntaxe

```
REPEAT
...
UNTIL <podmínka>
```

Význam

REPEAT:	Uvádí programovou smyčku.
UNTIL:	Označuje konec smyčky a způsobuje návrat zpět na její začátek.
<podmínka>:	Podmínka, která musí být splněna, aby se smyčkou REPEAT už neprocházelo.

Příklad:

Programový kód	Komentář
...	
REPEAT	; vyvolání smyčky REPEAT
...	
UNTIL ...	; kontrola, zda je podmínka splněna
...	

2.12.6 Příklad programu se vnořenými řídicími strukturami

Programový kód	Komentář
LOOP	
IF NOT \$P_SEARCH	; IF žádné vyhledávání bloku
G1 G90 X0 Z10 F1000	
WHILE \$AA_IM[X] <= 100	; WHILE (požadovaná hodnota osy X <= 100)
G1 G91 X10 F500	; Vrtací vzor
Z-5 F100	
Z5	
ENDWHILE	
ELSE	; ELSE vyhledávání bloku
MSG("Im Suchlauf wird nicht gebohrt")	
ENDIF	; ENDIF
\$A_OUT[1] = 1	; následující vrtaná deska
G4 F2	

Programový kód	Komentář
ENDLOOP	
M30	

2.13 Příkazy pro koordinaci (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

V kanálu NC systému může být v principu zpracováván program, který je v něm spuštěn, a to nezávisle na jiných kanálech jeho skupiny provozních režimů (BAG). Pokud se ale na opracování obrobku podílí více programů ve více kanálech jedné BAG současně, musí být zpracování těchto programů v jednotlivých kanálech koordinováno pomocí následujících koordinačních příkazů.

Předpoklad

Všechny kanály, které se na koordinaci programů podílejí, musí patřit do **stejné** skupiny provozních režimů (BAG).

MD10010 \$MC_ASSIGN_CHAN_TO_MODE_GROUP[<kanál>] = <číslo BAG>

Název kanálu místo čísla kanálu

Jako parametr v těchto předem definovaných procedurách koordinace programů se mohou namísto čísel kanálů používat také názvy kanálů, které jsou zadány v parametru MD20000 \$MC_CHAN_NAME[<index kanálu>]. Používání názvů kanálů v NC programech musí být napřed odblokováno:

MD10280 \$MN_PROG_FUNCTION_MASK, Bit 1 = TRUE

Syntaxe

```
INIT(<ChanNr>, <Prog>, <AckMode>)
START(<ChanNr>, <ChanNr>, ...)
WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)
WAITE(<ChanNr>, <ChanNr>, ...)
WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)
SETM(<MarkNr>, <MarkNr>, ...)
CLEARM(<MarkNr>, <MarkNr>, ...)
```

Význam

INIT():	Předem definovaná procedura v zvolení NC programu, který má být zpracováván v uvedeném kanálu.
START():	Předdefinovaná procedura pro spouštění programu vybraného v příslušném kanálu
WAITM():	Předem definovaná procedura pro čekání na dosažení značky pro čekání v uvedených kanálech Ve vlastním kanálu se uvedená značka pro čekání nastavuje příkazem WAITM. Předcházející blok se ukončí s přesným najetím. Po synchronizaci se značka pro čekání vymaže. Současně může být nastaveno max. 10 značek na každý kanál.
WAITE():	Předem definovaná procedura pro čekání na konec programu v jednom nebo ve více jiných kanálech
WAITMC(): ¹⁾	Předem definovaná procedura pro čekání na dosažení značky pro čekání v uvedených kanálech Oproti funkci WAITM se zabrzdění os na přesné najetí uskuteční jen tehdy, pokud v jiném kanálu nebylo značky pro čekání dosud dosaženo.

SETM () : ¹⁾	Předem definovaná procedura pro nastavení jedné nebo více značek pro čekání pro koordinaci kanálů Zpracování ve vlastním kanálu není touto funkcí nijak ovlivněno. Příkaz SETM zůstává v platnosti, i když je aktivován reset kanálu a NC-Start.		
CLEARM () : ¹⁾	Předem definovaná procedura pro vymazání jedné nebo více značek pro čekání pro koordinaci kanálů Zpracování ve vlastním kanálu není touto funkcí nijak ovlivněno. Příkaz CLEARM () vymaže všechny značky pro čekání v kanálu. Příkaz CLEARM (0) vymaže pouze značku pro čekání „0“. Příkaz CLEARM zůstává v platnosti, i když je aktivován reset kanálu a NC-Start.		
<ChanNr>:	Číslo kanálu Číslo vlastního kanálu se uvádět nemusí.		
	Typ:	INT	
<Prog>:	Absolutní nebo relativní zadání cesty (není nutné) + název programu		
	Typ:	STRING	
	Pokud budete potřebovat informace o zadávání cesty, viz: Literatura Příručka programování, Pro pokročilé; kapitola „Správa souborů a programů“ > „Paměť programů“ > „Adresování souborů v paměti pro programy“ (Strana 219)		
<AckMode>:	Režim potvrzování (volitelně)		
	Typ:	CHAR	
	Hod-noty:	"N"	Bez potvrzení
		"S"	Synchronní potvrzování
		Zpracovávání programu bude pozastaveno tak dlouho, dokud komponent příjemce nepotvrdí příkaz. Při pozitivním potvrzení bude zpracován následující příkaz. Při negativním potvrzení se vypíše chybové hlášení.	
<MarkNr>:	Číslo značky, na kterou se čeká. Poznámka V případě systému s více kanály je k dispozici maximálně 100 značek pro čekání (značka pro čekání 0 ... 99). V systému, který má jen jeden kanál, je k dispozici pouze značka pro čekání 0.		
	1) Pro potřeby specifické uživatelské komunikace a/nebo koordinace kanálů mohou být značky pro čekání ovládané příkazy SETM / CLEARM používány také bez příkazů podmíněného čekání WAITMC. Značky pro čekání si přitom ponechávají své hodnoty i po resetu kanálu a operaci NC-Start.		

Příklady

Příkaz START prostřednictvím názvů kanálů z parametru MD20000

• Nastavení parametru

```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "BEARBEITUNG" ; název pro kanál 1
$MC_CHAN_NAME[ 1 ] = "ZUFUEHRUNG" ; název pro kanál 2
```

• Programování

Programový kód	Komentář
START(BEARBEITUNG)	; spuštění kanálu 1
START(ZUFUEHRUNG)	; spuštění kanálu 2

Spouštění prostřednictvím lokálních „názvů kanálů“ a uživatelské proměnné

Programový kód	Komentář
DEF INT MASCHINE = 1	; definice uživatelské proměnné pro kanál 1
DEF INT LADER = 2	; definice uživatelské proměnné pro kanál 2
...	
START(MASCHINE)	; spuštění kanálu 1
START(LADER)	; spuštění kanálu 2

Spouštění prostřednictvím lokálních „názvů kanálů“, uživatelské proměnné a názvů kanálů zadaných v parametrech

Programový kód	Komentář
DEF INT chanNo1	; definice uživatelské proměnné pro kanál 1
DEF INT chanNo2	; definice uživatelské proměnné pro kanál 2
chanNo1 = CHAN_1	; přiřazení názvu kanálu 1 nastaveného v parametrech
chanNo2 = CHAN_2	; přiřazení názvu kanálu 2 nastaveného v parametrech
...	
START(chanNo1)	; spuštění kanálu 1
START(chanNo2)	; spuštění kanálu 2

Příkaz INIT s absolutním udáním cesty

Deaktivování programu /_N_MPF_DIR/_N_ABSPAN1_MPF v kanálu 2.

Programový kód

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

Příkaz INIT s názvem programu

Zvolení programu s názvem „MYPROG“. Řídící systém vyhledá program na základě cesty pro vyhledávání.

Programový kód

```
INIT(2, "MYPROG")
```

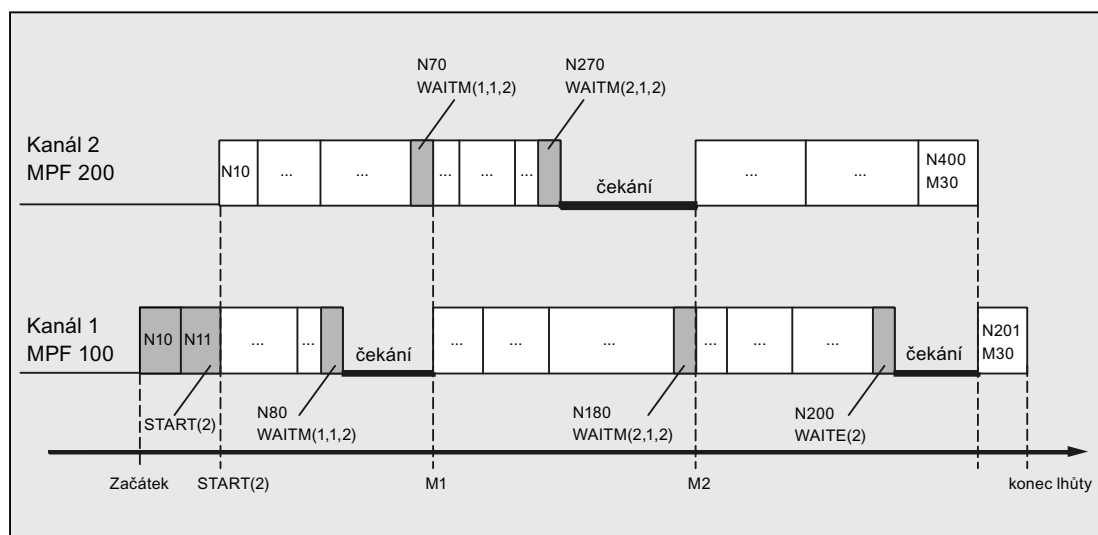
Koordinace programů pomocí funkce WAITM

- Kanál 1: Program /_N_MPF_DIR/_N_MPF100_MPF je již vybrán a je i spuštěn.

Programový kód	Komentář
	; program MPF100
N10 INIT(2,"MPF200","N")	; vyvolání programu MPF200, kanál 2
N11 START(2)	; spuštění kanálu 2
...	
N80 WAITM(1,1,2)	; čekání na značku WAIT 1 v kanálu 1 a 2
N81 ...	; kanál 1, N81, a kanál 2, N71, začnou ; synchronizovaně
...	
N180 WAITM(2,1,2)	; čekání na značku WAIT 2 v kanálu 1 a 2
N181 ...	; kanál 1, N181, a kanál 2, N271, začnou ; synchronizovaně
...	
N200 WAITE(2)	; čekání na konec programu v kanálu 2
N201 ...	; zpracování N201 se spustí teprve po skončení programu ; MPF200 v kanálu 2
N201 M30	; konec programu v kanálu 1

- Kanál 2: V kanálu 1 se prostřednictvím bloků N10 a N20 vyvolává a spouští program MPF200_MPF pro kanál 2.

Programový kód	Komentář
; \$PATH=/_N_MPF_DIR	; Programm MPF200
...	
N70 WAITM(1,1,2)	Čekání na značku WAIT 1 v kanálu 1 a 2
N71 ...	; kanál 1, N81, a kanál 2, N71, začnou ; synchronizovaně
...	
N270 WAITM(2,1,2)	Čekání na značku WAIT 2 v kanálu 1 a 2
N271 ...	; kanál 1, N181, a kanál 2, N271, začnou ; synchronizovaně
...	
N400 M30	Konec programu v kanálu 2



Okrajové podmínky

Nesynchronizované zaházení zpracování bloků následujících po značce WAIT

V případě koordinování kanálů pomocí značek WAIT může dojít k nesynchronizovanému spuštění zpracování následujících bloků. Toto chování se vyskytne tehdy, jestliže je bezprostředně před dosažením společné značky WAIT v jednom z kanálů, jež mají být synchronizovány, spuštěna akce, která má v tomto případě za následek vymazání zbytkové dráhy a implicitní nastavení původní polohy (REPOSA).

Předpoklad: Aktuální přiřazení os v kanálech 1 a 2

- Kanál 1: Osy X1 a U
- Kanál 2: Osa X2

Tabulka 2-2 Časový průběh v kanálech 1 a 2

Kanál 1	Kanál 2	Popis
...	...	Libovolné zpracování v kanálu 1 a 2
N100 WAITM(20,1,2)		Kanál 1: dosáhne značky WAIT a čeká na synchronní akci v kanálu 2
Začátek zpracování příkazu GETD(U): • Výměna osy • Vymazání zbytkové dráhy • REPOSA Konec	N200 GETD(U)	Kanál 2: vyžádá osu U z kanálu 1 Kanál 1: Zpracování příkazu GET(U) na pozadí
	N210 WAITM(20,1,2)	Kanál 2: Dosáhne značky WAIT. ⇒ Synchronizace kanálů 1 a 2 je tím ukončena.
	N220 G0 X2=100	Kanál 2: Zaházení zpracování od bloku N220
N110 G0 X1=100		Kanál 1: Časově posunuté zaházení zpracování od bloku N110

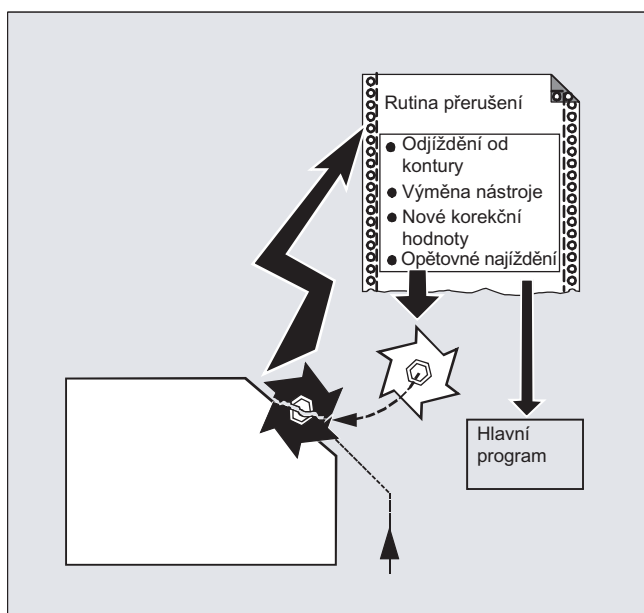
2.14 Rutiny přerušení (ASUP)

2.14.1 Funkce rutiny přerušení

Poznámka

Pojmy "asynchronní podprogram (ASUP)" a "Rutina přerušení", které se střídavě vyskytují v následujícím popisu, označují stejnou funkci.

Funkce rutiny přerušení bude objasněna na základě typického příkladu:



V průběhu zpracování se zlomí nástroj. V důsledku toho se aktivuje signál, který momentálně probíhající obráběcí operaci zastaví a současně spustí podprogram - tak zvanou rutinu přerušení. V tomto podprogramu se nacházejí všechny příkazy, které mají být v tomto případě provedeny.

Jakmile je podprogram zpracován (a v důsledku toho je obnovena připravenost k další práci), skočí řídicí systém zpátky do hlavního programu a obrábění bude pokračovat - po aktivování příkazu `REPOS` - na místě, kde došlo k přerušení (viz "Opětovné najíždění na konturu (Strana 543)").

POZOR

Nebezpečí kolize

Jestliže v podprogramu není žádný příkaz `REPOS` naprogramován, potom se najede do polohy odpovídající koncovému bodu bloku, který odpovídá přerušenému bloku.

Literatura

Příručka k funkcím, Základní funkce; BAG, kanál, zpracování programu, chování při resetu (K1), kapitola: "Asynchronní podprogramy (ASUP), rutiny přerušení"

2.14.2 Sestavování rutin přerušení**Sestavení rutiny přerušení jako podprogramu**

Při definici je rutina přerušení označena jako podprogram.

Příklad:

Programový kód	Komentář
PROC ABHEB_Z	; název programu "ABHEB_Z"
N10 ...	; potom následují NC bloky
...	
N50 M17	; v závěru konec programu a návrat do hlavního programu

Modální ukládání příkazu G-funkce (SAVE)

Při definici může být rutina přerušení označena atributem `SAVE`.

Atribut `SAVE` způsobuje, že modální příkazy G-funkcí, které byly před voláním rutiny přerušení aktivní, se uloží a po skončení rutiny přerušení se znovu aktivují (viz " Podprogramy s mechanismem `SAVE` (`SAVE`) (Strana 170) ").

Díky tomu je možné, aby obrábění po zpracování rutiny přerušení pokračovalo od místa, kde k přerušení došlo.

Příklad:

Programový kód
PROC ABHEB_Z SAVE
N10 ...
...
N50 M17

Další přiřazení rutin přerušení (SETINT)

V rámci rutiny přerušení mohou být naprogramovány příkazy `SETINT` (viz "Přiřazování a spouštění rutin přerušení (`SETINT`)" (Strana 130)), takže je možné spouštět další vnořené rutiny přerušení. Spouštění se uskutečňuje až prostřednictvím vstupu.

Literatura

Pokud budete potřebovat další informace týkající se sestavování podprogramů, viz kapitola "Technika podprogramů, technika maker".

2.14.3 Přřazování a spouštění rutin přerušení (SETINT, PRIO, BLSYNC)

Řídící systém disponuje několika rychlými vstupy (vstup 1 ... 8), které spouštění příslušné přerušení (1 ... 8). Každému přerušení je možno pomocí příkazu `SETINT` přiřadit prioritu a příslušnou rutinu přerušení. Jestliže je spuštěno přerušení aktivováním určitého rychlého vstupu, aktuální zpracování v kanálu se přeruší a spustí se odpovídající rutina přerušení.

Priorita přerušení

Pokud jsou v jednom výrobním programu přiřazena přerušení několika vstupům, musí být jednotlivým přerušením přiřazeny různé priority.

Jednomu přerušení může být přiřazena hodnota priority 1 ... 128. Hodnota 1 odpovídá nejvyšší, 128 nejnižší prioritě.

Syntaxe

```
SETINT (<n>) <NAME>
SETINT (<n>) PRIO=<hodnota> <NAME>
SETINT (<n>) PRIO=<hodnota> <NAME> BLSYNC
SETINT (<n>) PRIO=<hodnota> <NAME> LIFTFAST
```

Význam

SETINT (<n>) :	Signál přerušení <n> je přiřazen NC programu (ASUP) < název>. Přiřazená rutina přerušení se spouští, jakmile je rozpoznán signál přerušení <n> == 1. Poznámka: Jestliže je signálu přerušení <n> přiřazena jiná rutina přerušení, stává se předcházející přiřazení neplatným.	
<n>:	Číslo signálu přerušení	
	Typ:	INT
	Rozsah hodnot:	1 ... 32
PRIO=:	Priorita přerušení (nepovinné)	
<hodnota>:	(nepovinné) Hodnota priority	
	Typ:	INT
	Rozsah hodnot:	1 ... 128 (1 ⇒ nejvyšší priorita)
<NAME>:	Název NC programu (ASUP)	
BLSYNC:	(nepovinné) Příkaz <code>BLSYNC</code> způsobuje, že po spuštění přerušení se napřed počká, dokud nebude zpracován aktuální blok. Teprve potom se spustí rutina přerušení.	
LIFTFAST:	(nepovinné) Příkaz <code>LIFTFAST</code> způsobuje, že po spuštění přerušení se napřed uskuteční rychlé pozvednutí nástroje (viz kapitola "Rychlé pozvednutí od kontury (SETINT LIFTFAST, ALF) (Strana 133)"). Teprve potom se spustí rutina přerušení.	

Okrajové podmínky

Pravidla pro přerušení

1. Pro každé přerušení, které nemůže být okamžitě zpracováno nebo které je v daném okamžiku už zpracováváno, se do paměti uloží další požadavek na přerušení. Další požadavky na toto stejné přerušení jsou ztraceny.
2. Pokud je momentálně zpracováváno nějaké přerušení a pokud se aktivuje další přerušení s vyšší prioritou, přeruší toto přerušení s nižší prioritou. Po ukončení přerušení s vyšší prioritou pokračuje zpracování přerušení s nižší prioritou. Jestliže se během zpracování přerušení s vyšší prioritou objeví další požadavky pro přerušení s nižší prioritou, ukládají se tyto požadavky do paměti. Další se ztratí.
3. Pokud je momentálně zpracováváno nějaké přerušení a pokud se aktivuje další přerušení s vyšší prioritou, přeruší toto přerušení s nižší prioritou. Zpracuje se přerušení s vyšší prioritou. Pokud se znovu aktivuje přerušení s vyšší prioritou, zpracování aktuálního přerušení se pozastaví a zpracuje se přerušení s vyšší prioritou. Může existovat maximálně šest aktivních úrovní přerušení. Jedna momentálně zpracovávaná úroveň přerušení a pět čekajících úrovní přerušení. Pro každou aktivní úroveň přerušení se ukládá maximálně jeden další požadavek na přerušení. Všechny další požadavky na přerušení jsou ztraceny. Ztraceny jsou také požadavky na přerušení, jestliže by pro ně byly požadovány další úrovně přerušení (úroveň přerušení ≥ 7).

Příklady

Příklad 1: Přřazení rutin přerušení a definice priorit

Programový kód	Komentář
N20 SETINT(3) PRIO=1 ABHEB_Z	; IF vstup 3 == 1
	; THEN spustit rutinu přerušení "ABHEB_Z"
N30 SETINT(2) PRIO=2 ABHEB_X	; IF vstup 2 == 1
	; THEN spustit rutinu přerušení „ABHEB_X"

Jestliže se vstupy aktivují současně, jsou rutiny přerušení zpracovávány postupně po sobě v posloupnosti podle svých hodnot priority: napřed "ABHEB_Z", pak "ABHEB_X".

Příklad 2: Nové přiřazení rutiny přerušení

Programový kód	Komentář
N20 SETINT(3) PRIO=2 ABHEB_Z	; IF vstup 3 == 1
	; THEN spustit rutinu přerušení "ABHEB_Z"
...	
N80 SETINT(3) PRIO=1 ABHEB_X	; IF vstup 3 == 1
	; THEN spustit rutinu přerušení „ABHEB_X"

2.14.4 Deaktivování/opětovné aktivování přiřazení rutiny přerušení (DISABLE, ENABLE)

Příkaz `SETINT` může být pomocí příkazu `DISABLE` deaktivován a pomocí příkazu `ENABLE` opět aktivován, aniž by se přiřazení vstup --> rutina přerušení ztratilo.

Syntaxe

`DISABLE (<n>)`

`ENABLE (<n>)`

Význam

<code>DISABLE (<n>):</code>	Příkaz: Deaktivování přiřazení rutiny přerušení pro vstup <n>.	
<code>ENABLE (<n>):</code>	Příkaz: Opětovné aktivování přiřazení rutiny přerušení pro vstup <n>.	
<n>:	Parametr: Číslo signálu přerušení	
	Typ:	INT
	Rozsah hodnot:	1 ... 32

Příklad:

Programový kód	Komentář
N20 SETINT(3) PRIO=1 ABHEB_Z	; jestliže je aktivován vstup 3, pak by se
	; měla spustit rutina přerušení "ABHEB_Z"
...	
N90 DISABLE(3)	; příkaz SETINT z bloku N20 se deaktivuje
...	
N130 ENABLE(3)	; příkaz SETINT z bloku N20 se znovu aktivuje ak-
	tivuje
...	

2.14.5 Vymazání přiřazení rutiny přerušení (CLRINT)

Přiřazení signálu přerušení k nějakému NC programu (ASUP), které bylo definováno příkazem `SETINT`, může být vymazáno příkazem `CLRINT`.

Syntaxe

`CLRINT (<n>)`

Význam

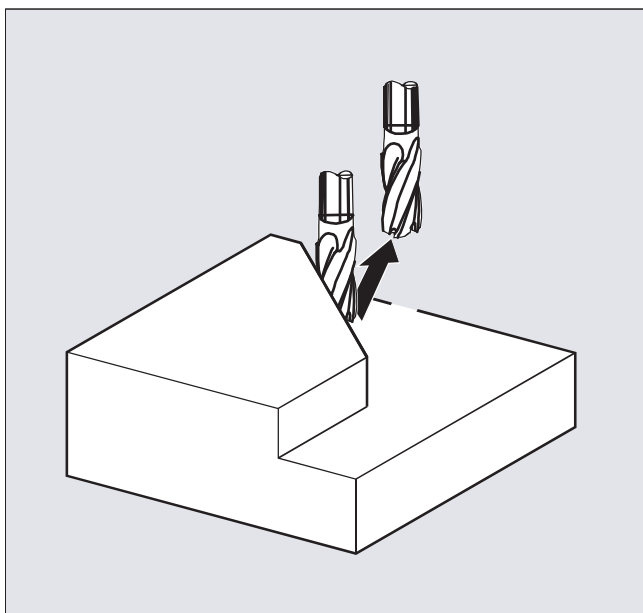
CLRINT (<n>):	Příkaz: Vymazání přiřazení signálu přerušení <n> NC programu (ASUP) <n> definovanému pomocí příkazu SETINT.	
<n>:	Parametr: Číslo signálu přerušení	
	Typ:	INT
	Rozsah hodnot:	1 ... 32

Příklad:

Programový kód	Komentář
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; přiřazení mezi vstupem "3" a rutinou přerušení "ABHEB_Z" se vymaže

2.14.6 Rychlé pozvednutí od kontury (SETINT LIFTFAST, ALF)

Pokud je příkaz SETINT naprogramován s parametrem LIFTFAST, pak když se aktivuje vstup, nástroj se rychlým pozvednutím odsune od kontury obrobku.



Další postup závisí na tom, zda příkaz SETINT obsahuje vedle parametru LIFTFAST ještě i rutinu přerušení.

S rutinou přerušení:

Po rychlém pozvednutí se spustí rutina přerušení.

Bez rutiny přerušení:

Po rychlém pozvednutí se zpracování zastaví a aktivuje se alarm.

2.14 Rutiny přerušení (ASUP)

Syntaxe

```
SETINT(<n>) PRIO=1 LIFTFAST
SETINT(<n>) PRIO=1 <NAME> LIFTFAST
```

Význam

SETINT(<n>):	Příkaz: Přiřazení vstupu <n> rutině přerušení. Jakmile se vstup <n> aktivuje, spustí se přiřazená rutina přerušení.	
<n>:	Parametr: Číslo vstupu	
	Typ:	INT
	Rozsah hodnot:	1 ... 8
PRIO= :	Definice priority	
<hodnota>:	Hodnota priority	
	Rozsah hodnot:	1 ... 128
	Hodnota 1 odpovídá nejvyšší prioritě.	
<NAME>:	Název podprogramu (rutiny přerušení), který má být zpracován.	
LIFTFAST:	Příkaz: Rychlé pozvednutí od kontury	
ALF=... :	Příkaz: Programovatelný směr pohybu (nachází se v pohybovém bloku) Pokud budete potřebovat informace o možnostech programování s příkazem ALF, viz kapitola " Směr pohybu při rychlém pozvednutí od kontury (Strana 135) ".	

Okrajové podmínky

Chování v případě aktivního framu se zrcadlovým převrácením

Při určování směru pozvednutí se kontroluje, zda není aktivní nějaký frame se zrcadlovým převrácením. V takovém případě se u směru pozvedávání vztaženého na směr tečny prohodí směry vlevo a vpravo. Směrové složky v nasměrování nástroje se zrcadlově nepřevracejí. Toto chování se aktivuje nastavením strojního parametru:

```
MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE
```

Příklad:

Zlomený nástroj má být automaticky nahrazen náhradním nástrojem. Opracovávání bude potom pokračovat s novým nástrojem.

Hlavní program:

Hlavní program	Komentář
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; když se aktivuje vstup 1, bude nástroj rychlým pozvednutím okamžitě odsunut od kontury (kód č. 7 pro korekci rádiusu nástroje G41) Potom se zpracuje rutina přerušení "W_WECHS".
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	

Hlavní program	Komentář
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

Podprogram:

Podprogram	Komentář
PROC W_WECHS SAVE	; podprogram s uložením aktuálního provozního stavu
N10 G0 Z100 M5	; poloha pro výměnu nástroje, zastavení vřetena
N20 T11 M6 D1 G41	; výměna nástroje
N30 REPOS L RMBBL M3	; opětovné najíždění na konturu a skok zpátky do hlavního programu (naprogramuje se v jednom bloku)

2.14.7 Směr pohybu při rychlém pozvednutí od kontury

Zpětný pohyb

Rovina zpětného pohybu je určena prostřednictvím následujících příkazů G-funkcí:

- **LFTXT**
Rovina zpětného pohybu se určuje z tečny ke dráze a ze směru nástroje (standardní nastavení).
- **LFWP**
Rovina zpětného pohybu je momentálně aktivní pracovní rovina, která je zvolena jedním z příkazů G-funkcí G17, G18 nebo G19. Směr zpětného pohybu je nezávislý na tečně ke dráze. Tímto způsobem může být naprogramováno rychlé pozvednutí rovnoběžně s osou.
- **LFPOS**
Zpětný pohyb osy stanovené příkazem POLFMASK / POLFLIN na absolutní pozici naprogramovanou pomocí příkazu POLF.
Příkaz ALF nemá žádný vliv na směr pozvednutí pro větší počet os, jakož i pro větší počet os v lineárním vztahu.

Literatura:

Programovací příručka, Základy; kapitola: "Rychlý zpětný pohyb během řezání závitů"

Programovatelný směr pohybu (ALF=...)

Směr se programuje v diskrétních úhlových krocích 45 stupňů pomocí příkazu ALF v rovině zpětného pohybu.

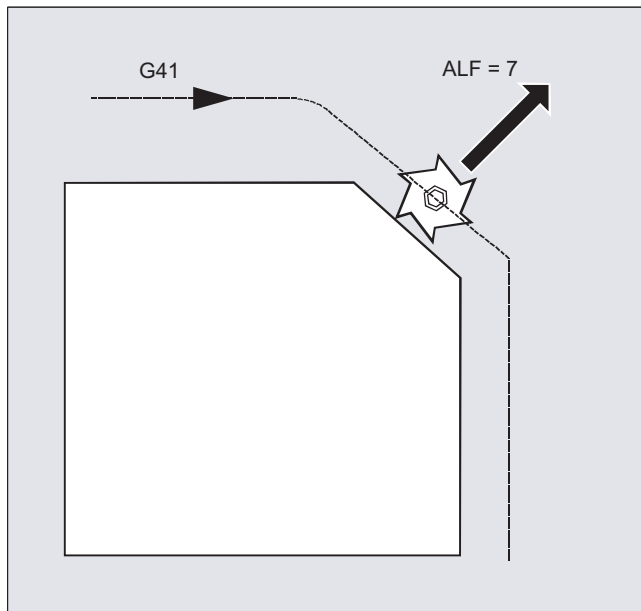
Možné směry pohybů jsou uloženy v řídicím systému pod speciálními kódovými čísly a pomocí těchto čísel je možné je vyvolat.

Příklad:

Programový kód

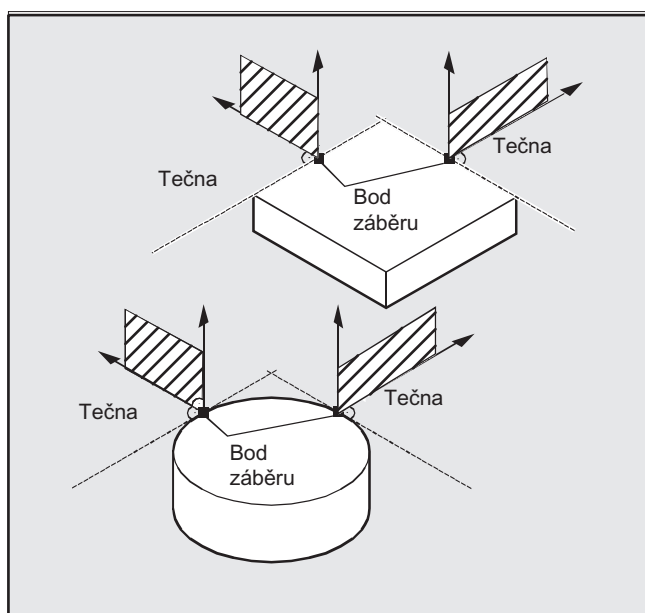
```
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST  
ALF=7
```

Když je aktivován příkaz G41 (směr opracování vlevo od kontury), pohybuje se nástroj kolmo od kontury.

**Vztažná rovina pro popis směru pohybu u příkazu LFTXT**

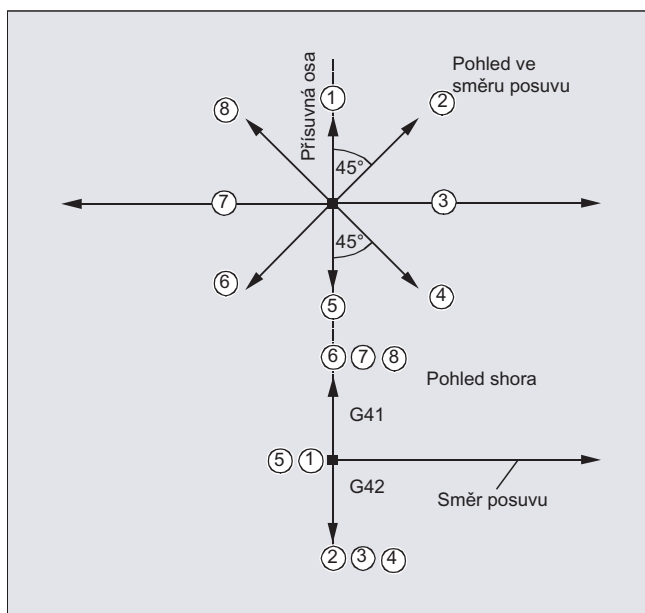
V bodě záběru nástroje na naprogramované kontuře je definována rovina, která slouží jako vztažná rovina pro udávání směru při pozvedávání pomocí odpovídajících kódových čísel.

Vztažná rovina je definována na základě osy přísuvu nástroje (směr přísuvu) a vektoru, který je kolmý jednak na tuto osu a jednak na tečnu v bodě záběru nástroje na kontuře.



Kódová čísla se směry pohybů u příkazu LFTXT

Na následujícím obrázku naleznete kódová čísla se směry pohybů, které se určují na základě vztažné roviny.



Pro ALF=1 definován návrat ve směru nástroje.

Když je nastaveno $ALF=0$, je funkce "Rychlé pozvednutí" vypnuta.



POZOR

Nebezpečí kolize

Když je aktivována korekce rádiusu nástroje:

- v případě příkazu G41 by se kódy 2, 3, 4
- v případě příkazu G42 by se kódy 6, 7, 8

neměly používat, protože v těchto případech nástroj přijíždí ke kontuře a dostal by se do kolize s obrobkem.

Kódová čísla se směry pohybů u příkazu LFWP

V případě LFWP je směr v pracovní rovině přiřazen podle následujícího uspořádání:

- G17: Rovina X/Y
ALF=1: Zpětný pohyb ve směru X
ALF=3: Zpětný pohyb ve směru Y
- G18: Rovina Z/X
ALF=1: Zpětný pohyb ve směru Z
ALF=3: Zpětný pohyb ve směru X
- G19: Rovina Y/Z
ALF=1: Zpětný pohyb ve směru Y
ALF=3: Zpětný pohyb ve směru Z

2.14.8 Posloupnost pohybů při rutinách přerušení

Rutina přerušení bez příkazu LIFTFAST

Pohyby os na dráze se zabrzdí až do zastavení. Potom se spustí rutina přerušení.

Poloha, kde došlo k zabrždění, se uloží jako místo, kde došlo k přerušení, a při aktivování příkazu REPOS s RMIBL na konci rutiny přerušení se na ni najede.

Rutina přerušení s příkazem LIFTFAST

Pohyby os na dráze se zabrzdí. Současně se jako superponovaný pohyb uskuteční pohyb LIFTFAST. Jakmile se pohyb po dráze a pohyb LIFTFAST úplně zastaví, spustí se rutina přerušení.

Jako místo, kde došlo k přerušení, se uloží pozice na kontuře, na které byl zahájen pohyb LIFTFAST a na které byla tedy dráha opuštěna.

Rutina přerušení s parametry `LIFTFAST` a `ALF=0` se chová identicky jako rutina přerušení bez příkazu `LIFTFAST`.

Poznámka

Vzdálenost, o kterou geometrické osy odjedou při rychlém pozvednutí od kontury, může být nastavena pomocí strojního parametru.

2.15 Výměna osy, výměna vřetena (RELEASE, GET, GETD)

Interpolace jedné nebo více os, příp. vřeten se smí uskutečňovat vždy jen v jednom kanálu. Pokud musí osa pracovat ve dvou různých kanálech střídavě (např. podavač pro výměnu palet), potom tato osa musí být napřed v aktuálním kanálu uvolněna a pak v jiném kanálu převzata. Kanály si osu vyměňují.

Rozšíření výměny os

Osa/vřeteno mohou být mezi vedlejším a hlavním programem vyměněny se zastavením předběžného zpracování a se synchronizací nebo alternativně i bez zastavení předběžného zpracování. Kromě toho je výměna os možná také pomocí následujících operací:

- Rotace zásobníku os AXCTSWE příp. AXCTWED pomocí implicitních příkazů GET/GETD.
- Frame s otočením, jestliže je tato osa spojena vazbou s jinou osou.
- Synchronní akce, viz "Pohybové synchronní akce", "Výměna os RELEASE, GET".

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje. Prostřednictvím v konfiguraci nastavitelných strojních parametrů musí být osa pro výměnu os ve všech kanálech jednoznačně definována a chování při výměně os může být pomocí strojních parametrů nastaveno tak, aby je bylo možné změnit.

Syntaxe

RELEASE(název osy, název osy, ...) nebo RELEASE(S1)

GET(název osy, název osy, ...) nebo GET(S2)

GETD(název osy, název osy, ...) nebo GETD(S3)

Pomocí příkazu GED (GET Directly) se osa přímo vyzvedne z nějakého jiného kanálu. To znamená, že k tomuto příkazu GETD nemusí být naprogramován žádný odpovídající příkaz RELEASE v nějakém jiném kanálu. Znamená to ale také, že nyní musí být s oním kanálem navázána komunikace (např. značky pro čekání).

Význam

RELEASE (název osy, název osy, ...):	Uvolnění jedné nebo více os
GET (název osy, název osy, ...):	Převzetí jedné nebo více os
GETD (název osy, název osy, ...):	Přímé převzetí jedné nebo více os
název osy:	Přiřazení osy v systému: AX1, AX2, ... nebo zadání názvů os stroje
RELEASE(S1):	Uvolnění vřetena S1, S2, ...
GET(S2):	Převzetí vřetena S1, S2, ...
GETD(S3):	Přímé převzetí vřetena S1, S2, ...

Požadavek GET bez zastavení předběžného zpracování

Pokud je po požadavku GET **bez** zastavení předběžného zpracování osa opět uvolněna některým z příkazů RELEASE (osa) nebo WAITP (osa), má následující příkaz GET za následek GET **se** zastavením předběžného zpracování.

**POZOR****Přiřazení os změněno**

Osa, příp. vřeteno, které byly příkazem GET převzaty, zůstávají tomuto kanálu přiřazeny i po RESETU, ať už byl vyvolán tlačítkem nebo programem.

Jestliže je osa zapotřebí ve svém základním kanálu, při spuštění nového programu se přiřazení vyměňovaných os, příp. vřeten musí programově-technicky ošetřit.

Při vypnutí a zapnutí systému (Power-On) jsou osy a vřetena přiřazena kanálům podle definice ve strojních parametrech.

Příklady**Příklad 1: Výměna os mezi dvěma kanály**

Ze 6 os jsou v kanálu 1 pro obrábění používány: 1., 2., 3. a 4. osa.

5. a 6. osa jsou používány v kanálu 2 pro výměnu obrobku.

Osu 2 má být možno vyměňovat mezi oběma kanály a po zapnutí systému (Power-On) má být tato osa přiřazena kanálu 1.

Program "MAIN" v kanálu 1:

Programový kód	Komentář
INIT (2,"TAUSCH2")	; Aktivování programu TAUSCH2 v kanálu 2.
N... START (2)	; Spuštění programu v kanálu 2.
N... GET (AX2)	; Převzít osu AX2.
...	
N... RELEASE (AX2)	; Uvolnit osu AX2.
N... WAITM (1,1,2)	; Čekání na značku WAIT v kanálu 1 a 2 kvůli synchronizaci v obou kanálech.
...	; Další postup po výměně osy.
N... M30	

Program "TAUSCH2" v kanálu 2:

Programování	Komentář
N... RELEASE (AX2)	
N160 WAITM(1,1,2)	; Čekání na značku WAIT v kanálu 1 a 2 kvůli synchronizaci v obou kanálech.
N150 GET(AX2)	; Převzít osu AX2.
...	; Další postup po výměně osy.
N... M30	

Příklad 2: Výměna osy bez synchronizace

Jestliže osa nemusí být synchronizována, není příkazem GET generováno žádné zastavení předběžného zpracování.

Programování	Komentář
N01 G0 X0	
N02 RELEASE (AX5)	
N03 G64 X10	
N04 X20	
N05 GET (AX5)	; Jestliže není zapotřebí žádná synchronizace, nejedná se o zpracovatelný blok.
N06 G01 F5000	; Žádný zpracovatelný blok.
N07 X20	; Žádný zpracovatelný blok, protože X má stejnou pozici jako v bloku N04.
N08 X30	; První zpracovatelný blok po bloku N05.
...	

Příklad 3: Aktivování výměny os bez zastavení předběžného zpracování

Předpoklad: Výměna os bez zastavení předběžného zpracování musí být nastavena v konfiguraci pomocí strojního parametru.

Programování	Komentář
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP (B)	; Osa B je nyní neutrální osou.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; Osa nespouští zastavení předběžného zpracování/příkaz REORG.
N041 G4 F2	
N050 M5	
N099 M30	

Jestliže se vřeteno, příp. osa B bezprostředně po bloku N023 jako **osa PLC** pohybuje na pozici 180 stupňů a zpět na pozici 1 stupeň, potom se tato osa opět stane neutrální osou a v bloku N40 nebude spouštět žádné zastavení předběžného zpracování.

Další informace

Předpoklady pro výměnu osy

- Prostřednictvím strojních parametrů musí být osa definována ve všech kanálech, ve kterých se používá.
- Pomocí strojního parametru pro specifickou **osu** musí být definováno, kterému kanálu má být osa po zapnutí systému (Power-On) přiřazena.

Popis

Uvolnění osy: RELEASE

Při uvolnění osy je potřeba dávat pozor na následující záležitosti:

1. Osa se nesmí podílet na žádné transformaci.
2. V případě spojení os (tangenciální řízení) musí být všechny do svazku seskupené osy uvolněny.
3. Konkurenční polohovací osa nemůže být v tomto stavu vyměňována.
4. V případě řídicí osy gantry jsou vyměněny také všechny vlečné osy.
5. V případě vazeb mezi osami (vlečení, spojení řídicí hodnotou, elektronická převodovka) může být uvolněna pouze řídicí osa vazby.

Převzetí osy: GET

Pomocí tohoto příkazu se uskutečňuje vlastní výměna osy. Veškerá odpovědnost za osu se tím přenáší na kanál, v němž byl příkaz naprogramován.

Důsledky použití příkazu GET:

Výměna osy se synchronizací:

Osa musí být synchronizována vždy, když byla v mezičase přiřazena v nějakém jiném kanálu nebo v PLC a pokud před příkazem GET nebyla uskutečněna žádná synchronizace pomocí příkazu "WAITP", G74 nebo vymazáním zbytkové dráhy.

- Předběžné zpracování se zastaví (stejně jako při příkazu STOPRE).
- Zpracování bude pozastaveno tak dlouho, dokud není výměna úplně dokončena.

Automatický příkaz "GET"

Jestliže je osa v principu k dispozici v nějakém kanálu, v daném okamžiku ale není "kanálovou osou", provede se příkaz "GET" automaticky. Jestliže už jsou osy synchronizovány, k zastavení předběžného zpracování nedochází.

Nastavení měnitelného chování při výměně osy

Okamžik předání osy může být nastaven pomocí strojního parametru, a to následujícím způsobem:

- Automatická výměna osy mezi dvěma kanály se uskuteční také tehdy, pokud byla osa pomocí příkazu WAITP uvedena do neutrálního stavu (stejně chování jak bylo popsáno výše).
- V případě požadavku na rotaci zásobníku os jsou všechny osy zásobníku os, které jsou přiřazeny uvedenému kanálu, převedeny pomocí implicitních příkazů GET, příp. GETD do daného kanálu. Následující výměna osy je znovu umožněna teprve po dokončení rotace zásobníku os.
- Po posunutém pomocném bloku v hlavní větvi programu se zkontroluje, zda je či není nezbytná reorganizace. Jedině v případě, že stavy os v tomto bloku **neodpovídají** momentálním stavům os, je reorganizace potřebná.
- Místo bloku s příkazem GET se zastavením předběžného zpracování a se synchronizací mezi vedlejším a hlavním programem mohou být osy vyměněny bez zastavení předběžného zpracování. Potom je generován jen jeden pomocný blok s příkazem GET. V hlavním programu se při zpracovávání tohoto bloku kontroluje, zda se stavy os v bloku shodují s aktuálními stavy os.

Pokud budete potřebovat další informace o funkcích souvisejících s výměnou os a vřeten, viz Příručka k funkcím, Rozšiřovací funkce; BAG, kanály, výměna os (K5).

2.16 Předávání osy jinému kanálu (AXTOCHAN)

Pomocí příkazu NC-jazyka AXTOCHAN může být pro určitou osu přikázáno, aby se tato osa předala do jiného kanálu. Osa může být do odpovídajícího kanálu přenesena jak z výrobního NC programu, tak i ze synchronní akce.

Syntaxe

```
AXTOCHAN(název osy, číslo kanálu[, název osy, číslo kanálu[, ...]])
```

Význam

Prvek	Popis
AXTOCHAN:	Vyžádání osy pro určitý kanál
název osy:	Přiřazení osy v systému: X, Y, ... nebo zadání názvů os stroje podílejících se na operaci. Jako kanál, který je v příkazu uveden, nesmí být veden vlastní kanál a nesmí se jednat ani o kanál, který v daném okamžiku disponuje interpolačními oprávněními pro danou osu.
číslo kanálu:	Číslo kanálu, jemuž má být v dané chvíli osa přiřazena.

Poznámka

Konkurenční polohovací osa a osa ovládaná výlučně prostřednictvím PLC

Osa PLC nesmí být jako konkurenční polohovací osa v kanálu vyměňována. Osa, která je ovládána výlučně prostřednictvím PLC, nesmí být NC programu přiřazena.

Literatura:

Příručka k funkcím, Rozšiřovací funkce; Polohovací osy (P2)

Příklad:

Příkaz AXTOCHAN v NC programu

Osy X a Y jsou známy v 1. kanálu a ve 2. kanálu. Momentálně disponuje interpolačními oprávněními kanál 1 a v kanálu 1 je spuštěn následující program:

Programový kód	Komentář
N110 AXTOCHAN(Y,2)	; osa Y se posune ve 2. kanálu
N111 M0	
N120 AXTOCHAN(Y,1)	; osa Y je přenesena opět zpátky (neutrální stav)
N121 M0	
N130 AXTOCHAN(Y,2,X,2)	; osa Y a osa X se posunou ve 2. kanálu (neutrální osy)
N131 M0	
N140 AXTOCHAN(Y,2)	; osa Y se posune ve 2. kanálu (NC program)
N141 M0	

Další informace

Příkaz AXTOCHAN v NC programu

V tomto případě se příkaz `GET` uskutečňuje a tím pádem se také čeká na skutečnou změnu stavu jen tehdy, vyskytne-li se požadavek na tuto osu pro NC program ve vlastním kanálu. Pokud je osa vyžádána pro jiný kanál nebo jestliže má být ve vlastním kanálu neutrální osou, potom je požadavek odpovídajícím způsobem pouze odložen.

Příkaz AXTOCHAN ze synchronní akce

Jestliže je osa vyžádána pro vlastní kanál, potom se příkaz `AXTOCHAN` ze synchronní akce převádí na příkaz `GET` ze synchronní akce. V tomto případě se osa v rámci prvního vyžádání pro vlastní kanál stává neutrální osou. V případě druhého vyžádání osy z NC programu je osa přiřazena analogicky k příkazu `GET` v NC programu. Pokud budete potřebovat informace k příkazu `GET` v synchronních akcích, viz kapitola "Pohybové synchronní akce".

2.17 Aktivování strojních parametrů (NEWCONF)

Pomocí příkazu `NEWCONF` jsou všechny strojní parametry nastaveny jako platné. Tato funkce může být aktivována také v uživatelském rozhraní HMI pomocí programového tlačítka "Aktivovat MD".

V rámci zpracovávání funkce "NEWCONF" se provádí implicitní zastavení předběžného zpracování, tzn. pohyb po dráze se přeruší.

Syntaxe

`NEWCONF`

Význam

<code>NEWCONF:</code>	Příkaz pro aktivování všech strojních parametrů do stavu platnosti "NEW_CONFIG".
-----------------------	--

Vyvolání příkazu NEWCONF z výrobního programu přes hranice kanálu

Jestliže jsou strojní parametry v důsledku zpracování výrobního programu změněny a potom jsou příkazem `NEWCONF` aktivovány, nastaví příkaz `NEWCONF` do aktivního stavu jen ty strojní parametry, které způsobují změny pro kanál výrobního programu.

Poznámka

Aby se všechny změny bezpečně aktivovaly do daného stavu platnosti, musí být příkaz `NEWCONF` uskutečněn v každém kanálu, ve kterém jsou momentálně vypočítávány také osy nebo funkce, které souvisejí se změněnými strojními parametry.

Příkazem `NEWCONF` nejsou aktivovány žádné osové strojní parametry.

V případě os ovládaných pomocí PLC musí být spuštěna funkce "reset osy".

Příklad:

Obrábění frézováním: Opracování pozic pro vrtání pomocí různých technologií

Programový kód	Komentář
N10 \$MA_CONTOUR_TOL[AX]=1.0	; změna strojního parametru
N20 NEWCONF	; aktivování strojních parametrů
...	

2.18 Zápis do souboru (WRITE)

Pomocí příkazu `WRITE` je možné zapisovat bloky/data z NC programu na konec souboru, který se nachází v pasivním systému souborů nebo v externí paměti programů (protokolový soubor). Může se jednat také o program, který je momentálně zpracováván.

Poznámka

Jestliže soubor, do něhož se má pomocí příkazu `WRITE` zapisovat, ještě v paměti programů neexistuje, bude nově založen.

Předpoklady

Momentálně nastavená úroveň ochrany musí být stejná nebo vyšší, než je úroveň oprávnění pro zápis do souboru pomocí příkazu `WRITE`. Pokud tomu tak není, přístup je odmítnut a vypíše se chybové hlášení (výsledná hodnota v chybové proměnné = 13).

Syntaxe

```
DEF INT <chyba>
...
WRITE(<Chyba>,"<Název souboru>"/"<ExtG>","<Blok/Data>")
```

Význam

WRITE:	Příkaz pro vložení bloku, příp. dat na konec uvedeného souboru		
<chyba>:	Parametr 1: Proměnná pro výslednou chybovou hodnotu		
	Typ:	INT	
	Hodnota:	0	žádná chyba
		1	Cesta není povolena
		2	Cesta není nalezena
		3	Soubor není nalezen
		4	nesprávný datový typ
		10	Soubor je plný
		11	Soubor je využíván
		12	žádné volné kapacity
		13	žádná přístupová oprávnění
		14	Chybějící příkaz <code>EXTOPEN</code> pro výstupní zařízení nebo zpracování tohoto příkazu skončilo neúspěchem
		15	Chyba při zápisu na externí zařízení
		16	naprogramována neplatná externí cesta

<název souboru>:	Parametr 2: Název souboru, do kterého se má uvedený blok, příp. uvedená data, vložit.	
	Typ:	STRING
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 219)".	
<ExtG>:	Jestliže mají být data odesílána na externí zařízení/do externího souboru pomocí funkce „Process DataShare“, musí být místo názvu souboru uveden symbolický identifikátor pro externí zařízení/soubor, které se mají otevřít.	
	Typ:	STRING
	Pokud budete potřebovat další informace, viz "Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE) (Strana 677)". Upozornění: Tento identifikátor musí být identický s identifikátorem použitým v příkazu EXTOPEN.	
<blok/data>:	Parametr 3: Blok, příp. data, která mají být do uvedeného souboru vložena.	
	Typ:	STRING

Poznámka

Při zápisu do pasivního systému souborů nebo do externí paměti programů implicitně vkládá příkaz WRITE na konec výstupního řetězce implicitně znak "LF" (LINE-FEED = konec řádku).

Při odesílání do externího zařízení/souboru prostřednictvím funkce „Process DataShare“ toto chování neplatí. Jestliže má být znak "LF" odeslán, musí být explicitně uveden ve výstupním řetězci.

--> K tomu viz příklad 3: Implicitní/explicitní znak "LF"!

Okrajové podmínky

- **Maximální velikost souboru (--> výrobce stroje!)**
Maximální možná velikost protokolového souboru v pasivním systému souborů se nastavuje pomocí strojního parametru:
MD11420 \$MN_LEN_PROTOCOL_FILE
Maximální velikost souboru platí pro všechny soubory, které byly založeny příkazem WRITE v pasivním systému souborů. V případě překročení této velikosti se aktivuje chybové hlášení a blok, příp. data se do paměti neuloží. Pokud postačuje kapacita paměti, může být založen nový soubor.

Příklady

Příklad 1: Příkaz WRITE do pasivního systému souborů bez absolutního udání cesty

Programový kód	Komentář
N10 DEF INT ERROR	; definice chybové proměnné
N20 WRITE (ERROR, "PROT", "PROTOKOLL VOM 7.2.97")	; zapiše text "PROTOKOLL VOM 7.2.97" do souboru _N_PROT_MPF

2.18 Zápis do souboru (WRITE)

Programový kód	Komentář
N30 IF ERROR	; vyhodnocování chyby
N40 MSG ("Fehler bei WRITE-Befehl:" << ERROR)	
N50 M0	
N60 ENDIF	
...	

Příklad 2: Příkaz WRITE do pasivního systému souborů s absolutním udáním cesty

Programový kód
...
WRITE(ERROR, "/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF", "PROTOKOLL VOM 7.2.97")
...

Příklad 3: Implicitní/explicitní znak "LF"

a, Zápis do pasivního systému souborů s implicitně vkládaným znakem "LF"

Programový kód
...
N110 DEF INT ERROR
N120 WRITE(ERROR, "/_N_MPF_DIR/_N_MYPROTFILE_MPF", "MY_STRING")
N130 WRITE(ERROR, "/_N_MPF_DIR/_N_MYPROTFILE_MPF", "MY_STRING")
N140 M30

Výsledný výstup:

MY_STRING

MY_STRING

b, Zápis do externího souboru bez implicitně vkládaného znaku "LF"

Programový kód
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR, DEV_1)
N240 WRITE(ERROR, DEV_1, "MY_STRING")
N250 WRITE(ERROR, DEV_1, "MY_STRING")
N260 EXTCLOSE(ERROR, DEV_1)
N270 M30

Výsledný výstup:

MY_STRINGMY_STRING

c, Zápis do externího souboru s explicitně vkládaným znakem "LF"

Aby bylo dosaženo stejného výsledku jako u případu a, musí být naprogramováno následující:

Programový kód

```
...  
N200 DEF STRING[30] DEV_1  
N210 DEF INT ERROR  
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"  
N230 EXTOPEN(ERROR,DEV_1)  
N240 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")  
N250 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")  
N260 EXTCLOSE(ERROR,DEV_1)  
N270 M30
```

Výsledný výstup:

MY_STRING

MY_STRING

2.19 Vymazání souboru (DELETE)

Pomocí příkazu `DELETE` je možné vymazat všechny soubory bez ohledu na to, zda tyto soubory vznikly příkazem `WRITE` nebo ne. Pomocí příkazu `DELEE` mohou být mazány také soubory, které byly vytvořeny s vyšší úrovní přístupových práv.

Syntaxe

```
DEF INT <chyba>
DELETE(<chyba>,"<název souboru>")
```

Význam

DELETE:	Příkaz pro vymazání zadaného souboru.		
<chyba>:	Proměnná pro výslednou chybovou hodnotu		
	Typ:	INT	
	Hodnota:	0	žádná chyba
		1	Cesta není povolena
		2	Cesta není nalezena
		3	Soubor není nalezen
		4	nesprávný datový typ
		11	Soubor je využíván
		12	žádné volné kapacity
		20	jiná chyba
<název souboru>:	Název souboru, který má být vymazán.		
	Typ:	STRING	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 219)".		

Příklad:

Programový kód	Komentář
N10 DEF INT ERROR	; definice chybové proměnné
N15 STOPRE	; zastavení předběžného zpracování
N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")	; vymazání souboru TEST1 v adresáři podprogramu
N30 IF ERROR	; vyhodnocování chyby
N40 MSG("Fehler bei DELETE-Befehl:" <<ERROR)	
N50 M0	
N60 ENDIF	

2.20 Čtení řádků v souboru (READ)

Příkaz `READ` načte v zadaném souboru jeden nebo více řádků a načtené informace uloží do pole typu `STRING`. Každý načtený řádek obsadí v tomto poli jeden jeho prvek.

Předpoklady

Momentálně nastavená úroveň ochrany musí být stejná nebo vyšší, než je úroveň oprávnění pro čtení souboru pomocí příkazu `READ`. Pokud tomu tak není, přístup je odmítnut a vypíše se chybové hlášení (výsledná hodnota v chybové proměnné = 13).

Syntaxe

```
DEF INT <chyba>
DEF STRING[<délka řetězce>] <výsledek>[<n>,<m>]
READ(<chyba>,"<název souboru>",<počáteční řádek>,<počet řádků>,<výsledek>)
```

Význam

READ:	Příkaz pro čtení řádků ze zadaného souboru a pro uložení těchto řádků do pole proměnných.		
<chyba>:	Proměnná pro výslednou chybovou hodnotu (parametr Call-By-Reference)		
	Typ:	INT	
	Hodnota:	0	žádná chyba
		1	Cesta není povolena
		2	Cesta není nalezena
		3	Soubor není nalezen
		4	nesprávný datový typ
		11	Soubor je využíván
		13	Nedostatečná přístupová oprávnění
		21	Řádek není k dispozici (parametr <počáteční řádek> nebo parametr <počet řádků> je větší než počet řádků v zadaném souboru).
		22	Délka pole v proměnné pro výsledky (<výsledek>) je příliš malá.
		23	Oblast řádků je zvolena příliš velká (parametr <počet řádků>, takže čtení probíhá až za konec souboru.
<název souboru>:	Název souboru, z něhož se má číst (parametr Call-By-Value)		
	Typ:	STRING	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 219)".		

<počáteční řádek>:	Počáteční řádek oblasti v souboru, z něhož se má číst (parametr Call-By-Value)		
	Typ:	INT	
	Hodnota:	0	Načte se určitý počet řádků před koncem souboru, přičemž jejich počet je zadán parametrem <počet řádků>
		1 ... n	Číslo prvního řádku, která se má načíst.
<počet řádků>:	Počet řádků, které se mají načíst (parametr Call-By-Value)		
	Typ:	INT	
<výsledek>:	Proměnná pro výsledek (parametr "Call-By-Reference")		
	Pole proměnných, do kterého se ukládá načtený text.		
	Typ:	STRING (max. délka: 255)	
	Jestliže parametr <počet řádků> specifikuje méně řádků, než kolik odpovídá velikosti pole výsledků [<n>, <m>], potom zůstanou zbývající prvky pole nezměněny.		
	Konec řádku reprezentovaný řídícími znaky "LF" (Line Feed) nebo "CR LF" (Carriage Return Line Feed) se do pole výsledků neukládá .		
Jestliže je řádek delší, než je definovaná délka řetězce, jsou načtené řádky oříznuty. Nevypisuje se žádné chybové hlášení.			

Poznámka

Binární soubory není možné načítat. V takovém případě je nahlášena chyba "Nesprávný datový typ" (výsledná hodnota v chybové proměnné = 4). Následující datové typy není možné číst: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Příklad:

Programový kód	Komentář
N10 DEF INT ERROR	; definice chybové proměnné
N20 DEF STRING[255] RESULT[5]	; definice proměnné pro výsledky
N30 READ(ERROR, "/_N_CST_DIR/_N_TESTFILE_MPF", 1, 5, RESULT)	; název souboru s doménou, příponou souboru a zadání cesty
N40 IF ERROR <> 0	; vyhodnocování chyby
N50 MSG("FEHLER"<<ERROR<<"BEI READ-BEFEHL")	
N60 M0	
N70 ENDIF	
...	

2.21 Kontrola existence souboru (ISFILE)

Pomocí příkazu ISFILE je možné zkontrolovat, zda určitý soubor existuje v paměti programu.

Syntaxe

```
<výsledek>=ISFILE("<název souboru>")
```

Význam

ISFILE:	Příkaz pro kontrolu, zda určitý soubor existuje		
<název souboru>:	Název souboru, jehož existence má být zkontrolována.		
	Typ:	STRING	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 219)".		
<výsledek>:	Proměnná, do které se uloží výsledek kontroly		
	Typ:	BOOL	
	Hodnota:	TRUE	Soubor existuje
		FALSE	Soubor neexistuje

Příklady

Příklad 1

Programový kód	Komentář
N10 DEF BOOL RESULT	; definice proměnné pro výsledky
N20 RESULT=ISFILE("TESTFILE")	
N30 IF (RESULT==FALSE)	
N40 MSG("SOUBOR NEEXISTUJE")	
N50 M0	
N60 ENDIF	
...	

Příklad 2

Programový kód	Komentář
N10 DEF BOOL RESULT	; definice proměnné pro výsledky
N20 RESULT=ISFILE("TESTFILE")	
N30 IF (NOT ISFILE("TESTFILE"))	
N40 MSG("SOUBOR NEEXISTUJE")	
N50 M0	
N60 ENDIF	
...	

2.22 Zjišťování informací o souboru (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Prostřednictvím příkazů FILEDATE, FILETIME, FILESIZE, FILESTAT a FILEINFO mohou být zjišťovány určité informace o daném souboru, jako jsou datum/přesný čas, kdy došlo k poslednímu přístupu za účelem zápisu, momentální velikost souboru, stav souboru nebo shrnutí všech těchto informací.

Předpoklady

Momentálně nastavená úroveň ochrany musí být stejná nebo vyšší, než je úroveň oprávnění pro vyvolání výpisu nadřazeného adresáře. Pokud tomu tak není, přístup je odmítnut a vypíše se chybové hlášení (výsledná hodnota v chybové proměnné = 13).

Syntaxe

```
FILE....(<chyba>,"<název souboru>",<výsledek>)
```

Význam

FILEDATE:	Poskytuje datum posledního přístupu do souboru za účelem zápisu	
FILETIME:	Poskytuje přesný čas posledního přístupu do souboru za účelem zápisu	
FILESIZE:	Poskytuje údaj o momentální velikosti souboru	
FILESTAT:	Poskytuje pro status následující oprávnění , která se týkají určitého souboru. • Čtení (r: read) • Zápis (w: write) • Spustit (x: execute) • Zobrazit (s: show) • Vymazat (d: delete) Poznámka: Tyto úrovně ochrany jsou speciálními vlastnostmi pasivního systému souborů. Při přístupu do externí paměti pro programy proto příkaz FILESTAT vrací pouze předdefinovaná přístupová oprávnění (77777).	
FILEINFO:	Pro daný soubor poskytuje součet všech informací , které je možné načíst pomocí funkcí FILEDATE, FILETIME, FILESIZE a FILESTAT.	
<chyba>:	Proměnná pro výslednou chybovou hodnotu (parametr Call-By-Reference)	
	Typ:	VAR INT
	Hodnota:	0 žádná chyba
		1 Cesta není povolena
		2 Cesta není nalezena
		3 Soubor není nalezen
		4 nesprávný datový typ
		13 Nedostatečná přístupová oprávnění
	22 Délka řetězce v proměnné pro výsledky (<výsledek>) je příliš malá.	

2.22 Zjišťování informací o souboru (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

<název souboru>:	Název souboru, o němž mají být zjištěny požadované informace		
	Typ:	CHAR[160]	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 219)".		
<výsledek>:	Proměnná pro výsledek (Call-By-Reference-parametr)		
	Proměnná, do které se ukládají požadované informace o souboru.		
	Typ:	VAR CHAR[8]	kde: FILEDATE Formát: „dd.mm.rr"
		VAR CHAR[8]	kde: FILETIME Formát: "hh:mm:ss"
		VAR INT	kde: FILESIZE Výpis velikosti souboru v bytech.
		VAR CHAR[5]	kde: FILESTAT Formát: "rwxsd" (r: read, w: write, x: execute, s: show, d: delete)
		VAR CHAR[32]	kde: FILEINFO Formát: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

Příklad:

Programový kód	Komentář
N10 DEF INT ERROR	; definice chybové proměnné
N20 STRING[32] RESULT	; definice proměnné pro výsledky
N30 FILEINFO(ERROR, "/_N_MPF_DIR/_N_TESTFILE_MPF", RESULT)	; název souboru s doménou, příponou souboru a udáním cesty
N40 IF ERROR <> 0	; vyhodnocování chyby
N50 MSG("FEHLER"<<ERROR<<"u příkazu FILEINFO")	
N60 M0	
N70 ENDIF	
...	

Program v příkladu by mohl do proměnné pro výsledek RESULT uložit například následující řetězec:

"77777 12345678 26.05.00 13:51:30"

2.23 Zaokrouhlení (ROUNDUP)

Pomocí funkce "ROUNDUP" mohou být vstupní hodnoty typu REAL (desetinná čísla s desetinnou tečkou) zaokrouhlována na nejbližší vyšší celé číslo.

Syntaxe

ROUNDUP (<hodnota>)

Význam

ROUNDUP:	Příkaz pro zaokrouhlení vstupní hodnoty
<hodnota>:	Vstupní hodnota typu REAL

Poznámka

Vstupní hodnoty typu INTEGER (celá čísla) se přenesou do výsledku beze změny.

Příklady

Příklad 1: Různé vstupní hodnoty a výsledky jejich zaokrouhlení

Příklad:	Výsledek zaokrouhlení
ROUNDUP (3.1)	4.0
ROUNDUP (3.6)	4.0
ROUNDUP (-3.1)	-3.0
ROUNDUP (-3.6)	-3.0
ROUNDUP (3.0)	3.0
ROUNDUP (3)	3.0

Příklad 2: Funkce ROUNDUP v NC programu

Programový kód

```
N10 X=ROUNDUP (3.5) Y=ROUNDUP (R2+2)
N15 R2=ROUNDUP ($AA_IM[Y])
N20 WHEN X=100 DO Y=ROUNDUP ($AA_IM[X])
...
```

2.24 Technika podprogramů

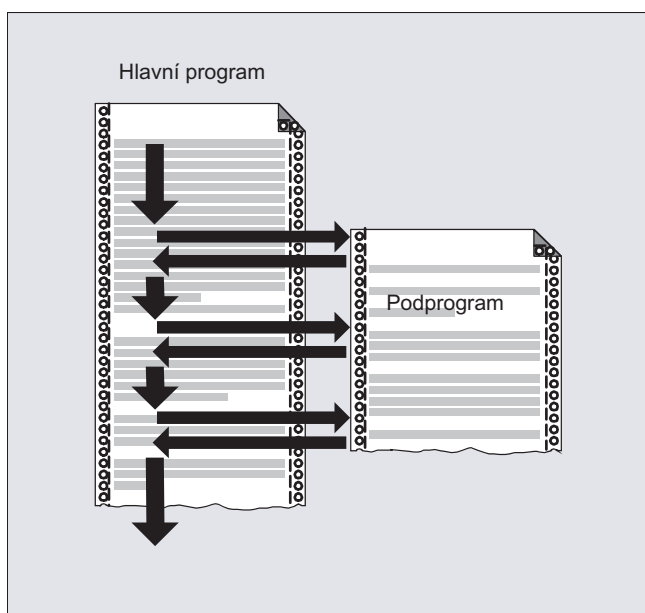
2.24.1 Všeobecně

2.24.1.1 Podprogram

Označení podprogram pochází ještě z dob, kdy byly výrobní programy pevně rozděleny na hlavní programy a podprogramy. Hlavní programy přitom byly výrobní programy, které se v řídicím systému vybraly ke zpracování a které s potom spouštěly. Podprogramy bývaly výrobní programy, které bývaly vyvolávány z hlavního programu.

Toto pevné rozdělení v dnešním jazyku systému SINUMERIK už neexistuje. V principu může být každý výrobní program vybrán a spuštěn jako program hlavní nebo může být vyvolán z jiného výrobního programu jako podprogram.

Proto jsou v dalším popisu jako podprogramy označovány ty výrobní programy, které jsou vyvolávány z jiného výrobního programu.



Aplikace

Stejně jako u všech vyšších programovacích jazyků se i v jazyku NC systému používají podprogramy k tomu, aby se části programu, které se mají použít vícekrát, uložily do samostatného uzavřeného programu.

Podprogram nabízí následující výhody:

- Výrazně vyšší přehlednost a čitelnost programu
- Zlepšení jakosti díky opětovnému použití testovaných částí programu

- Nabízí možnost vytvořit si specifické knihovny obráběcích programů
- Úspora místa v paměti

2.24.1.2 Názvy podprogramů

Pravidla pro sestavování názvů

Název podprogramu může být libovolný, musí však být dodržena následující pravidla:

- Přípustné znaky:
 - Písmena: A ... Z, a ... z
 - Číslice: 0 ... 9
 - Znak podtržení: _
- První dva znaky by měly být dvě písmena nebo znak podtržení, za kterým následuje písmeno.

Poznámka

Jestliže jsou tyto podmínky splněny, potom může být tento NC program samotný vyvoláván jako podprogram z nějakého jiného programu zadáním tohoto názvu programu. Jestliže název programu proti tomu začíná číslicemi, potom je volání podprogramu možné jediné pomocí příkazu CALL.

- Maximální délka: 24 znaků

Poznámka

Velká/malá písmena

V jazyku NC systému SINUMERIK **nejsou** rozlišována velká a malá písmena.

Poznámka

Nepřípustné názvy programů

Aby se zabránilo problémům s aplikacemi Windows, následující názvy programů se **nesmějí** používat:

- CON, PRN, AUX, NUL
 - COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9
 - LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9
-

Interní přípony řídicího systému

Název podprogramu zadaný při jeho sestavování je uvnitř řídicího systému rozšířen o předponu a o příponu:

- Počáteční řetězec: `_N_`
- Koncový řetězec: `_SPF`

Použití názvu programu

Když je název programu použit např. při volání podprogramu, jsou možné všechny kombinace předpony, názvu programu a přípony.

Příklad:

Podprogram s názvem programu SUB_PROG může být spuštěn pomocí následujících identifikátorů:

1. SUB_PROG
2. _N_SUB_PROG
3. SUB_PROG_SPF
4. _N_SUB_PROG_SPF

Hlavní program a podprogram stejného názvu

Jestliže existuje hlavní program (.MPF) a podprogram (.SPF) se stejným názvem programu, musí se při používání názvů těchto programů v NC programu zadávat také odpovídající přípona souboru, aby byla zajištěna jejich jednoznačná identifikace. Jinak se použije program, který bude při procházení podle cesty pro vyhledávání nalezen jako první objekt s uvedeným názvem programu.

2.24.1.3 Vnoření podprogramů

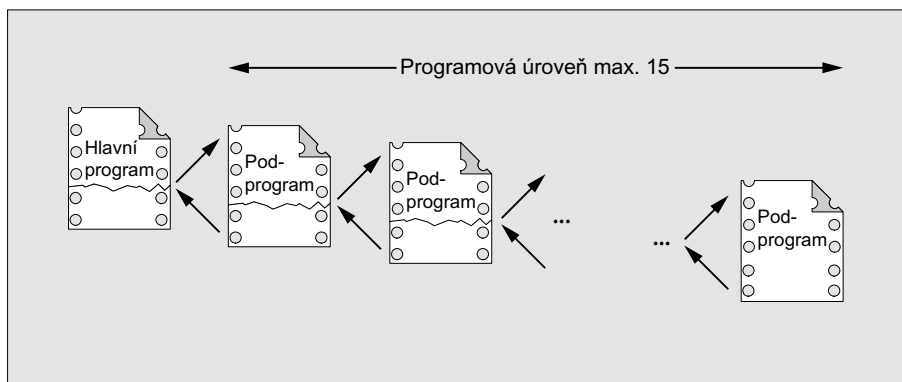
Hlavní program může vyvolávat podprogramy, které mohou opět vyvolávat další podprogramy. Zpracování programů takto může být postupně do sebe vnořeno. Každý program přitom běží na své vlastní programové úrovni.

Hloubka vnoření

Jazyk NC systému má v současnosti k dispozici 16 programových úrovní. Hlavní program vždy běží na nejvyšší programové úrovni 0. Podprogram vždy běží na nejbližší nižší programové úrovni, která následuje za úrovní, z níž byl vyvolán. Programová úroveň 1 je tedy první úroveň pro podprogramy.

Rozdělení programových úrovní:

- Programová úroveň 0: Úroveň hlavního programu
- Programová úroveň 1 - 15: Úroveň podprogramu 1 - 15



Rutiny přerušení (ASUP)

Jestliže je v rámci rutiny přerušení vyvoláván podprogram, nebude zpracováván na programové úrovni, která je momentálně v kanálu aktivní (n), nýbrž rovněž na nejbližší nižší programové úrovni (n+1). By byla tato operace možná i na nejnižší programové úrovni, jsou v souvislosti s rutinami přerušení k dispozici ještě další 2 programové úrovně (16 a 17).

Jestliže jsou zapotřebí více než 2 programové úrovně, je nutné na to explicitně brát zřetel při návrhu struktury výrobního programu, který má být v kanálu zpracováván. To znamená, že se smí počítat jen s maximálně tolika programovými úrovněmi, aby byl pro zpracování přerušení k dispozici dostatečný počet programových úrovní.

Jestliže tedy zpracování přerušení potřebuje např. 4 programové úrovně, musí být výrobní program strukturován tak, aby se obsazovala maximálně 13. programová úroveň. Pokud se potom vyskytne přerušení, budou pro ně potřebné 4 programové úrovně (14 až 17) k dispozici.

Cykly firmy Siemens

Cykly firmy Siemens potřebují 3 programové úrovně. Volání cyklů firmy Siemens se proto musí uskutečňovat maximálně:

- zpracovávání výrobního programu: Programová úroveň 12
- Rutina přerušení: Programová úroveň 14

2.24.1.4 Cesta pro vyhledávání

Při volání podprogramu bez udání cesty řídící systém prohledává paměti pro programy, které jsou k dispozici, podle předem definované posloupnosti při vyhledávání (viz "Cesta pro vyhledávání při volání podprogramu (Strana 223)").

2.24.1.5 Formální a skutečný parametr

O formálních a skutečných parametrech se mluví v souvislosti s definicí a s vyvoláváním podprogramů s předávanými parametry.

Formální parametr

Při definici podprogramu musí být definovány také parametry, které mají být podprogramu předány, což jsou tak zvané formální parametry, spolu s jejich typem a s názvy.

Formální parametry tak definují rozhraní podprogramu.

Příklad:

Programový kód	Komentář
PROC KONTUR (REAL X, REAL Y)	; formální parametry: X a Y, oba typu REAL
N20 X1=X Y1=Y	; najíždění osou X1 na pozici X a osou Y1 na pozici Y
...	
N100 RET	

Skutečné parametry

Podprogramu musí být při jeho volání předány absolutní hodnoty nebo proměnné, což jsou tak zvané skutečné parametry.

Skutečný parametr takto při volání doplní do rozhraní podprogramu skutečné hodnoty.

Příklad:

Programový kód	Komentář
N10 DEF REAL BREITE	; definice proměnných
N20 BREITE=20.0	; přiřazení hodnoty proměnné
N30 KONTUR(5.5, BREITE)	; volání podprogramu se skutečnými parametry: 5.5 a BREITE
...	
N100 M30	

2.24.1.6 Předávání parametrů

Definice podprogramu s předáváním parametrů

Definice podprogramu s předáváním parametrů se uskutečňuje pomocí klíčového slova PROC a výpisem úplného seznamu všech podprogramem očekávaných parametrů.

Neúplné předávání parametrů

Při volání podprogramu nemusí být explicitně předávány vždy všechny parametry, které jsou v rozhraní podprogramu definovány. Pokud je nějaký parametr vypuštěn, bude se pro tento parametr předávat standardní hodnota "0".

Aby ale byla posloupnost parametrů jednoznačně identifikována, musí být vždy zapsány čárky, které slouží jako oddělovací znak pro parametry. Výjimku tvoří poslední parametry. Pokud jsou při volání vypuštěny, mohou odpadnout i poslední čárky.

Příklad:

Podprogram:

Programový kód	Komentář
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; formální parametry: X, Y a Z
...	
N100 RET	

Hlavní program:

Programový kód	Komentář
PROC MAIN_PROG	
...	
N30 SUB_PROG(1.0,2.0,3.0)	; volání podprogramu s úplným předáváním parametrů: X=1.0, Y=2.0, Z=3.0
...	
N100 M30	

Příklady pro volání podprogramu v bloku N30 s neúplným předáváním parametrů:

N30 SUB_PROG(, 2.0, 3.0)	; X=0.0, Y=2.0, Z=3.0
N30 SUB_PROG(1.0, , 3.0)	; X=1.0, Y=0.0, Z=3.0
N30 SUB_PROG(1.0, 2.0)	; X=1.0, Y=2.0, Z=0.0
N30 SUB_PROG(, , 3.0)	; X=0.0, Y=0.0, Z=3.0
N30 SUB_PROG(, ,)	; X=0.0, Y=0.0, Z=0.0

UPOZORNĚNÍ**Předávání parametrů typu Call-By-Reference**

Parametry, které jsou předávány prostřednictvím Call-by-Reference, nesmí být při volání podprogramu vypuštěny.

UPOZORNĚNÍ**Datový typ AXIS**

Parametry datového typu AXIS nesmí být při volání podprogramu vypuštěny.

Kontrola předávaných parametrů

Prostřednictvím systémové proměnné \$P_SUBPAR [n], kde n = 1, 2, ..., může být v podprogramu zkontrolováno, zda se parametr explicitně předává nebo zda byl vypuštěn. Index n se vztahuje na posloupnost formálních parametrů. Index n = 1 se vztahuje na 1. formální parametr, index n = 2 na 2. formální parametr atd.

Následující úsek programu ukazuje na příkladu 1. formálního parametru, jak může být toto přezkoumání realizováno:

Programování	Komentář
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; formální parametry: X, Y a Z
N20 IF \$P_SUBPAR[1]==TRUE	; přezkoumání 1. formálního parametru X
...	; tyto operace se uskuteční, jestliže byl formální parametr X explicitně předán
N40 ELSE	
...	; tyto operace se uskuteční, jestliže formální parametr X nebyl předán
N60 ENDIF	
...	; všeobecné operace
N100 RET	

2.24.2 Definice podprogramu

2.24.2.1 Podprogram bez předávání parametrů

Při definici podprogramů bez předávání parametrů může definiční řádek na začátku programu odpadnout.

Syntaxe

```
[PROC <název programu>]
...
```

Význam

PROC:	Definiční příkaz na začátku programu
<název programu>:	Název programu

Příklad:

Příklad 1: Podprogram s příkazem PROC

Programový kód	Komentář
PROC SUB_PROG	; definiční řádek
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; skok zpátky z podprogramu

Příklad 2: Podprogram bez příkazu PROC

Programový kód	Komentář
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; skok zpátky z podprogramu

Viz také

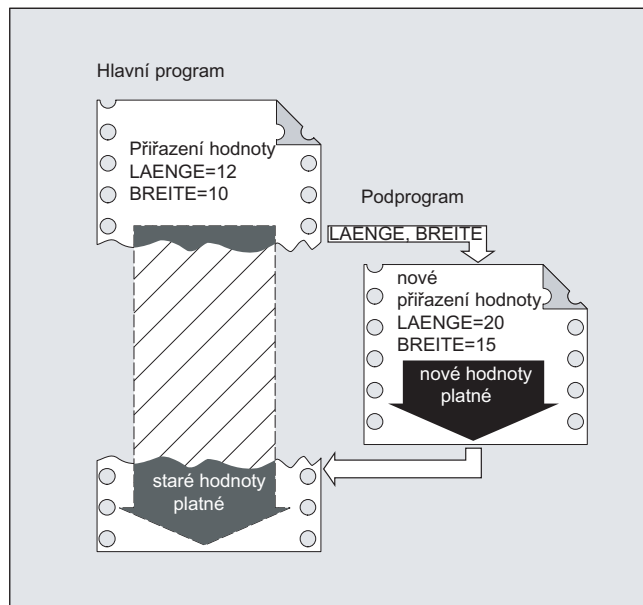
Volání podprogramu bez předávání parametrů (Strana 192)

2.24.2.2 Podprogram s předáváním parametrů Call-by-Value (PROC)

Definice podprogramu s předáváním parametrů Call-by-Value se uskutečňuje pomocí klíčového slova `PROC`, za kterým následuje název programu a výpis úplného seznamu všech parametrů spolu s jejich typy a názvy. Definiční příkaz se musí nacházet na prvním řádku programu.

Call-by-Value (Volání hodnotou)

Při předávání parametrů typu Call-By-Value volající program přenáší do podprogramu pouze hodnotu proměnné. Podprogram v důsledku toho nemá k proměnné žádný přímý přístup. Jestliže se hodnota parametru změní, změní se jen hodnota, která je viditelná v podprogramu. Hodnota proměnné definovaná ve volajícím programu zůstává nezměněna. Předávání parametrů typu Call-by-Value proto nemá žádný zpětný vliv na volající program.



Syntaxe

```
PROC <název programu> (<typ parametru> <název parametru>=<počáteční  
hodnota>, ...)
```

Poznámka

Může být předáváno až 127 parametrů.

Význam

PROC:	Definiční příkaz na začátku programu
<název programu>:	Název programu
<typ parametru>:	Datový typ parametru (např. REAL, INT, BOOL)
<název parametru>:	Název parametru
<počáteční hodnota>:	Volitelná hodnota pro inicializaci parametru (nepovinné) Jestliže při volání podprogramu není uveden žádný parametr, obsahuje tento parametr inicializační hodnotu, která je mu přiřazena.

Příklady**Příklad 1**

Definice podprogramu SUB_PROG se třemi parametry typu REAL s předdefinovanými hodnotami:

Programový kód

```
PROC SUB_PROG (REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

Příklad 2

Různé varianty volání

Programový kód

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; varianty volání
  SUB_PROG
  SUB_PROG (PAR_1, PAR_2, PAR_3)
  SUB_PROG (PAR_1)
  SUB_PROG (PAR_1, , PAR_3)
  SUB_PROG ( , , PAR_3)
N100 RET
```

Viz také

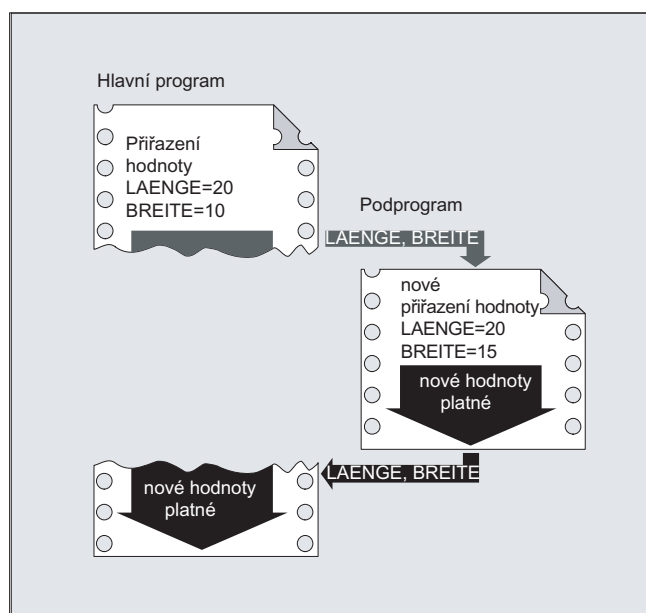
Volání podprogramu s předáváním parametrů (EXTERN) (Strana 194)

2.24.2.3 Podprogram s předáváním parametrů Call-by-Reference (PROC, VAR)

Definice podprogramu s předáváním parametrů Call-by-Reference se uskutečňuje pomocí klíčového slova `PROC`, za kterým následuje název programu a výpis úplného seznamu všech parametrů s klíčovým slovem `VAR` a jejich typy a názvy. Definiční příkaz se musí nacházet na prvním řádku programu. Jako parametr mohou být předávány také odkazy na pole.

Call-by-Reference (Volání odkazem)

Při předávání parametrů typu Call-By-Reference volající program přenáší do podprogramu ne hodnotu proměnné, ale odkaz (ukazatele) na příslušnou proměnnou. Podprogram v důsledku toho nemá k proměnné žádný přímý přístup. Jestliže se hodnota parametru změní, nezmění se jen hodnota, která je viditelná v podprogramu, ale změní se i hodnota proměnné definovaná ve volajícím programu. Předávání parametrů typu Call-by-Reference má tedy i po skončení podprogramu zpětný vliv na volající program.



Poznámka

Předávání parametrů typu Call-by-Reference je zapotřebí jen tehdy, pokud je předávána proměnná ve volajícím programu lokálně definována (LUD). Proměnné, které jsou globální v kanálu nebo v NC systému, nemusí být předávány, protože podprogram k nim může mít přímý přístup.

Syntaxe

```
PROC <název programu> (VAR <typ parametru> <název parametru>, ...)
PROC <název programu> (VAR <typ pole> <název pole>
[<m>,<n>,<o>], ...)
```

Poznámka

Může být předáváno až 127 parametrů.

Význam

PROC:	Definiční příkaz na začátku programu
VAR:	Klíčové slovo pro předávání parametru typu Call-by-Reference
<název programu>:	Název programu
<typ parametru>:	Datový typ parametru (např. REAL, INT, BOOL)
<název parametru>:	Název parametru
<typ pole>:	Datový typ prvku pole (např. REAL, INT, BOOL)
<název pole>:	Název pole
[<m>, <n>, <o>]:	Velikost pole V současnosti je možné pracovat s maximálně 3-rozměrnými poli:
	<m>: Velikost pole pro 1. rozměr
	<n>: Velikost pole pro 2. rozměr
	<o>: Velikost pole pro 3. rozměr

Poznámka

- Název programu zadaný po klíčovém slově PROC se musí shodovat s názvem programu, který je uveden na uživatelském rozhraní.
- Pomocí polí s neurčenou délkou fungujícími jako předávané parametry mohou být podprogramem zpracovávána pole proměnné délky. Za tím účelem se při definici např. dvourozměrného pole jako formální parametr délka v 1. rozměru nezadá. Čárka ale musí být zapsána.

Příklad: PROC <název programu> (VAR REAL FELD[, 5])

Příklad:

Definice podprogramu se dvěma parametry Call-by-Reference typu REAL:

Programový kód

```
; parametr 1: odkaz na typ: REAL, název: LAENGE
; parametr 2: odkaz na typ: REAL, název: BREITE
PROC SUB_PROG(VAR REAL LAENGE, VAR REAL BREITE)
```

Viz také

Volání podprogramu s předáváním parametrů (EXTERN) (Strana 194)

2.24.2.4 Ukládání modálních G-funkcí (SAVE)

Atribut `SAVE` způsobuje, že všechny příkazy modálních G-funkcí, které byly před voláním podprogramu aktivní, se uloží a po ukončení podprogramu se znovu aktivují.

UPOZORNĚNÍ

Přerušení režimu řízení pohybu po dráze

Jestliže je v době aktivního režimu řízení pohybu po dráze vyvolán podprogram s atributem `SAVE`, režim řízení pohybu po dráze se po skončení podprogramu (skok zpět) přeruší.

Syntaxe

```
PROC <název podprogramu> SAVE
```

Význam

SAVE:	Uložení stavu příkazů modálních G-funkcí před voláním podprogramu a jejich opětovné nastavení po skončení podprogramu.
-------	--

Příklad:

V podprogramu `KONTUR` je aktivní příkaz modální G-funkce `G91` (inkrementální rozměry). V hlavním programu je aktivní příkaz modální G-funkce `G90` (absolutní rozměry). Když bude v definici podprogramu zadán příznak `SAVE`, po skončení podprogramu bude v hlavním programu znovu v platnosti `G90`.

Definice podprogramu:

Programový kód	Komentář
PROC KONTUR (REAL WERT1) SAVE	; definice podprogramu s parametrem <code>SAVE</code>
N10 G91 ...	; příkaz modální G-funkce <code>G91</code> : Řetězové kótování
N100 M17	; konec podprogramu

Hlavní program:

Programový kód	Komentář
N10 G0 X... Y... G90	; příkaz modální G-funkce <code>G90</code> : Absolutní rozměry
N20 ...	
...	
N50 KONTUR (12.4)	; volání podprogramu
N60 X... Y...	; příkaz modální G-funkce <code>G90</code> je díky příkazu <code>SAVE</code> znovu aktivován

Okrajové podmínky

Framy

Chování framů v souvislosti s podprogramy s atributem `SAVE` závisí na typu framu a může být nastaveno pomocí strojních parametrů.

Literatura

Příručka k funkcím, Základní funkce; "Osy, souřadné systémy, framy" (K2), kapitola: "Návrat zpět z podprogramu s atributem `SAVE`"

2.24.2.5 Potlačení zpracovávání blok po bloku (`SBLOF`, `SBLON`)

Potlačení zpracovávání blok po bloku pro celý program

Když je aktivováno zpracování blok po bloku, programy označené atributem `SBLOF` budou zpracovány kompletně jako jeden blok, tzn. pro celý program se zpracovávání blok po bloku potlačí.

Příkaz `SBLOF` se nachází na řádku `PROCa` platí až do konce nebo do přerušení podprogramu. Pomocí příkazu pro návrat se rozhone, zda se má nebo nemá na konci podprogramu zpracování pozastavit.

Návrat s příkazem `M17`: Zastavení na konci podprogramu

Návrat s příkazem `RET`: Žádné zastavení na konci podprogramu

Potlačení zpracování blok po bloku v rámci programu

Příkaz `SBLOF` musí být naprogramován v samostatném bloku. Od tohoto bloku je zpracovávání bloku po bloku vypnuto, a to až do:

- následujícího příkazu `SBLON` nebo
- na konci aktivní úrovně podprogramu

Syntaxe

Potlačení zpracovávání blok po bloku pro celý program:

```
PROC ... SBLOF
```

Potlačení zpracování blok po bloku v rámci programu:

```
SBLOF
...
SBLON
```

Význam

PROC:	První příkaz programu
SBLOF:	Příkaz pro deaktivování zpracování blok po bloku Příkaz SBLOF se může nacházet v bloku s příkazem PROC nebo v samostatném bloku.
SBLON:	Příkaz pro aktivování zpracování blok po bloku Příkaz SBLON se musí nacházet v samostatném bloku.

Okrajové podmínky

- Potlačení zpracování blok po bloku a vypisování bloků**
 Zobrazování právě zpracovávaného bloku může být v cyklech/podprogramech potlačeno pomocí příkazu DISPLOF. Pokud je příkaz DISPLOF naprogramován spolu s příkazem SBLOF, pak se při zastavení během zpracovávání blok po bloku uvnitř cyklu/podprogramu vypisuje totéž co před voláním cyklu/podprogramu.
- Potlačení zpracovávání blok po bloku v systémových nebo uživatelských asynchronních podprogramech (ASUP)**
 Jestliže je zastavování při zpracovávání blok po bloku v systémových nebo uživatelských asynchronních podprogramech (ASUP) prostřednictvím nastavení ve strojním parametru MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK potlačeno (bit0 = 1 příp. bit1 = 1), potom může být zastavování během zpracovávání blok po bloku opět aktivováno naprogramováním příkazu SBLON v programu ASUP.
 Pokud je zastavování při zpracovávání blok po bloku v uživatelských asynchronních podprogramech (ASUP) prostřednictvím nastavení ve strojním parametru MD20117 \$MC_IGNORE_SINGLEBLOCK_ASUP potlačeno, potom naprogramováním příkazu SBLON v programu ASUP **nemůže** být zastavování během zpracovávání blok po bloku znovu aktivováno.
- Zvláštnosti potlačení zpracovávání blok po bloku u různých typů zpracování blok po bloku**
 Když je aktivní zpracovávání blok po bloku SBL2 (zastavování po každém bloku výrobního programu), v bloku s příkazem SBLON **nedochází** k pozastavení, pokud je v parametru MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK (zabránit zastavení při zpracování blok po bloku) nastaven bit 12 na hodnotu "1".
 Když je aktivní zpracovávání blok po bloku SBL3 (zastavení po každém bloku výrobního programu i v cyklech) je příkaz SBLOF potlačen.

Příklady

Příklad 1: Potlačení zpracování blok po bloku v rámci programu

Programový kód	Komentář
N10 G1 X100 F1000	
N20 SBLOF	; vypnutí zpracování blok po bloku
N30 Y20	
N40 M100	
N50 R10=90	
N60 SBLON	; opětovné zapnutí zpracování blok po bloku
N70 M110	

Programový kód	Komentář
N80 ...	

Oblast mezi bloky N20 a N60 se v režimu zpracování blok po bloku zpracuje jako jeden krok.

Příklad 2: Cyklus se má pro uživatele chovat jako jeden příkaz

Hlavní program:

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

Cyklus CYCLE1:

Programový kód	Komentář
N100 PROC CYCLE1 DISPLOF SBLOF	; potlačení zpracovávání 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 se při aktivním zpracovávání blok po bloku zpracuje najednou, tzn. aby se cyklus CYCLE1 zpracoval, stačí jen jednou stisknout tlačítko Start.

Příklad 3: ASUP spouštěný z PLC pro aktivování změněného posunutí počátku a korekčních parametrů nástroje se nemá vypisovat.

Programový kód
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

Programový kód

```

N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET

```

Příklad 4: S nastavením MD10702 Bit 12 = 1 se nebude zastavovat

Výchozí situace:

- Zpracování blok po bloku je aktivní.
- MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

Hlavní program:

Programový kód	Komentář
N10 G0 X0	; na tomto řádku výrobního programu zastavovat
N20 X10	; na tomto řádku výrobního programu zastavovat
N30 CYCLE	; pohybový blok generovaný cyklem
N50 G90 X20	; na tomto řádku výrobního programu zastavovat
M30	

Cyklus CYCLE:

Programový kód	Komentář
PROC CYCLE SBLOF	; potlačení zastavování při zpracovávání blok po bloku
N100 R0 = 1	
N110 SBLON	; kvůli nastavení MD10702 Bit12=1 se na tomto řádku výrobního programu nebude zastavovat
N120 X1	; na tomto řádku výrobního programu se bude zastavovat
N140 SBLOF	
N150 R0 = 2	
RET	

Příklad 5: Potlačení zpracování blok po bloku v případě vnořených podprogramů

Výchozí situace:

Zpracování blok po bloku je aktivní.

Vnoření podprogramů:

Programový kód	Komentář
N10 X0 F1000	; na tomto bloku se bude zastavovat

Programový kód	Komentář
N20 UP1 (0)	
PROC UP1 (INT _NR) SBLOF	; potlačení zastavování při zpracovávání blok po bloku
N100 X10	
N110 UP2 (0)	
PROC UP2 (INT _NR)	
N200 X20	
N210 SBLON	; zapnutí zastavování při zpracovávání blok po bloku
N220 X22	; na tomto bloku se bude zastavovat
N230 UP3 (0)	
PROC UP3 (INT _NR)	
N300 SBLOF	; potlačení zastavování při zpracovávání blok po bloku
N305 X30	
N310 SBLON	; zapnutí zastavování při zpracovávání blok po bloku
N320 X32	; na tomto bloku se bude zastavovat
N330 SBLOF	; potlačení zastavování při zpracovávání blok po bloku
N340 X34	
N350 M17	; příkaz SBLOF je aktivní
N240 X24	; na tomto bloku se bude zastavovat Příkaz SBLON je aktivní.
N250 M17	; na tomto bloku se bude zastavovat Příkaz SBLON je aktivní.
N120 X12	
N130 M17	; na tomto návratovém bloku se bude zastavovat Je aktivní příkaz SBLOF z příkazu PROC.
N30 X0	; na tomto bloku se bude zastavovat
N40 M30	; na tomto bloku se bude zastavovat

Další informace

Zablokování zpracování blok po bloku pro asynchronní podprogramy

Aby se program ASUP při zpracování blok po bloku zpracoval v jednom kroku, musí být v programu ASUP naprogramován příkaz `PROC` s příkazem `SBLOF`. To platí také pro funkci "Editovatelný systémový ASUP" (MD11610 \$MN_ASUP_EDITABLE).

Příklad pro editovatelný systémový ASUP:

Programový kód	Komentář
N10 PROC ASUP1 SBLOF DISPLOF	
N20 IF \$AC_ASUP=='H200'	
N30 RET	; žádný příkaz REPOS při změně provozního režimu

Programový kód	Komentář
N40 ELSE	
N50 REPOSA	; REPOS ve všech ostatních případech
N60 ENDIF	

Ovlivňování zpracování programu při zpracování blok po bloku

Při zpracovávání blok po bloku může uživatel zpracovat výrobní program jeden blok po druhém. Existují následující druhy nastavení:

- SBL1: Zpracování IPO blok po bloku se zastavením po každém bloku s funkcemi stroje.
- SBL2: Zpracování blok po bloku se zastavením po každém bloku.
- SBL3: Zastavení v cyklu (aktivováním SBL3 se příkaz SBLOF potlačí).

Potlačení zpracování blok po bloku v případě vnořených podprogramů

Pokud bylo v podprogramu v příkazu PROC naprogramováno SBLOF, při návratu z podprogramu pomocí příkazu M17 bude zpracování pozastaveno. Tím se zabrání, aby se ve volajícím programu ihned zpracoval následující blok. Jestliže je v podprogramu, který nemá příkaz SBLOF v příkazu PROC, příkazem SBLOF aktivováno potlačení zpracování blok po bloku, bude zpracování pozastaveno teprve až po následujícím bloku s funkcemi stroje ve volajícím programu. Je-li toto chování nežádoucí, musí být v podprogramu ještě před návratem (M17) znovu naprogramován příkaz SBLON. Při návratu pomocí příkazu RET se zpracování v nadřazeném programu nezastaví.

2.24.2.6 Potlačení vypisování aktuálního bloku (DISPLOF, DISPLON, ACTBLOCNO)

V příslušném okně se standardně vypisuje aktuální programový blok. V cyklech, příp. v podprogramech, může být vypisování aktuálního bloku pomocí příkazu DISPLOF potlačeno. Namísto aktuálního bloku se potom vypisuje volání cyklu, příp. podprogramu. Pomocí příkazu DISPLON může být potlačení vypisování bloku opět odstraněno.

Příkazy DISPLOF, příp. DISPLON se programují na řádek programu s příkazem PROC, takže platí pro celý podprogram a implicitně také pro všechny podprogramy volané tímto podprogramem, které neobsahují žádný z příkazů DISPLON, příp. DISPLOF. Toto chování platí také pro programy ASUP.

Syntaxe

```
PROC ... DISPLOF
PROC ... DISPLOF ACTBLOCNO
PROC ... DISPLON
```

Význam

DISPLOF:	Příkaz pro potlačování vypisování aktuálního bloku.	
	Umístění:	Na konci programového řádku s příkazem PROC.
	Platnost:	Až do návratu z podprogramu nebo do konce programu.
	Upozornění: Jestliže jsou z podprogramu s příkazem DISPLOF vyvolávány další podprogramy, potom je vypisování aktuálního bloku potlačeno také v těchto podprogramech, pokud v nich není explicitně naprogramován příkaz DISPLON.	
DISPLON:	Příkaz pro odstranění potlačování vypisování aktuálního bloku	
	Umístění:	Na konci programového řádku s příkazem PROC.
	Platnost:	Až do návratu z podprogramu nebo do konce programu.
	Upozornění: Jestliže jsou z podprogramu s příkazem DISPLON vyvolávány další podprogramy, potom se i v těchto podprogramech aktuální programový blok, pokud v nich není explicitně naprogramován příkaz DISPLOF.	
ACTBLOCNO:	Příkaz DISPLOF ve spojení s atributem ACTBLOCNO způsobuje, že v případě alarmu se bude vypisovat číslo aktuálního bloku, ve kterém se alarm vyskytl. To platí i tehdy, jestliže je na některé z nižších programových úrovní naprogramován příkaz DISPLOF. V případě příkazu DISPLOF bez atributu ACTBLOCNO se oproti tomu vypisuje číslo bloku s voláním cyklu, příp. podprogramu z poslední programové úrovně, která není opatřena příkazem DISPLOF.	

Příklady

Příklad 1: Potlačení vypisování aktuálního bloku v cyklu

Programový kód	Komentář
PROC CYCLE(AXIS TOMOV, REAL POSITION)	; potlačení vypisování aktuálního bloku
SAVE DISPLOF	Místo toho se má vypisovat volání cyklu, např.: CYCLE(X,100.0)
DEF REAL DIFF	; obsah cyklu
G01 ...	
...	
RET	; návrat zpět z podprogramu V okně pro vypisování bloku se vypisuje blok, který následuje za voláním cyklu.

Příklad 2: Vypisování bloku v případě aktivování alarmu

Podprogram SUBPROG1 (s atributem ACTBLOCNO):

Programový kód	Komentář
PROC SUBPROG1 DISPLOF ACTBLOC-	
NO	
N8000 R10 = R33 + R44	
...	

Programový kód	Komentář
N9040 R10 = 66 X100	; aktivování alarmu 12080
...	
N10000 M17	

Podprogram SUBPROG2 (bez atributu ACTBLOCNO):

Programový kód	Komentář
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; aktivování alarmu 12080
...	
N7000 M17	

Hlavní program:

Programový kód	Komentář
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; hlášení alarmu = "12080 kanál K1 blok N9040 Syntaktická chyba u textu R10="
N2060 ...	
N2350 SUBPROG2	; hlášení alarmu = "12080 kanál K1 blok N2350 Syntaktická chyba u textu R10="
...	
N3000 M30	

Příklad 3: Odstranění potlačení vypisování aktuálního bloku

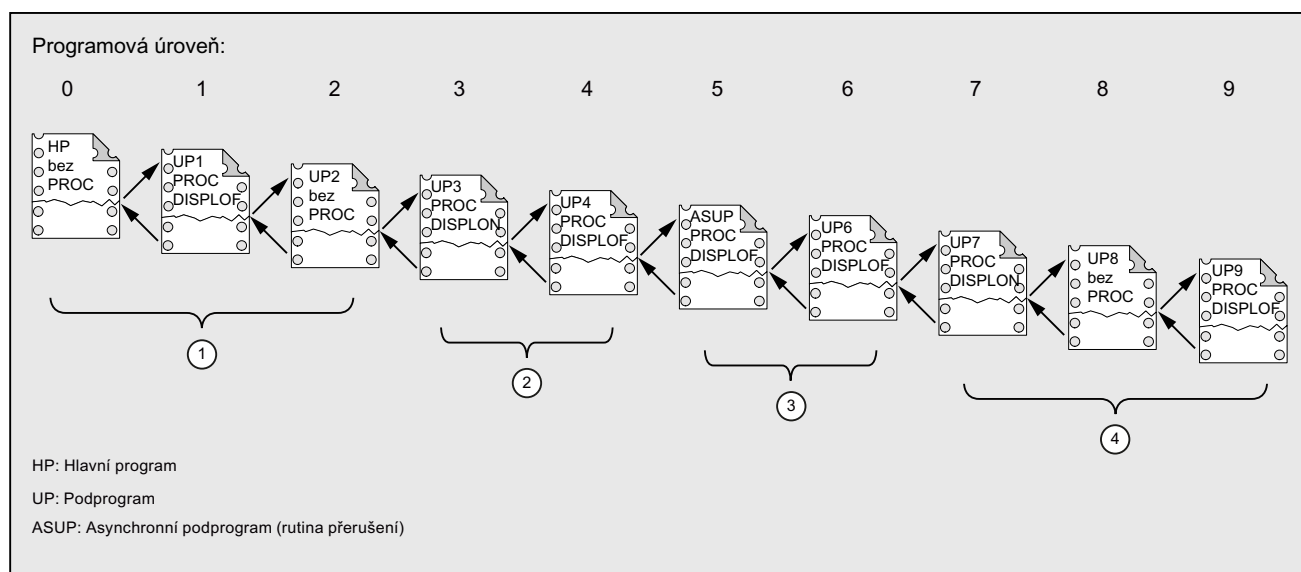
Podprogram SUB1 s potlačením vypisování:

Programový kód	Komentář
PROC SUB1 DISPLOF	; potlačení vypisování aktuálního bloku v podprogramu SUB1 Místo toho se má vypisovat blok s voláním podprogramu SUB1.
...	
N300 SUB2	; volání podprogramu SUB2
...	
N500 M17	

Podprogram SUB2 bez potlačení vypisování:

Programový kód	Komentář
PROC SUB2 DISPLON	; odstranění potlačení vypisování aktuálního bloku v podprogramu SUB2
...	
N200 M17	; návrat zpět do podprogramu SUB1 V podprogramu SUB1 je vypisování aktuálního bloku opět potlačeno.

Příklad 4: Chování vypisování při různých kombinacích příkazů DISPLON/DISPLOF



2.24.2.7 Označení podprogramů s přípravou (PREPRO)

Klíčovým slovem `PREPRO` na konci řádku s příkazem `PROC` mohou být označeny všechny soubory, které mají být při náběhu systému předběžně zpracovány.

Poznámka

Tento druh přípravy programu je závislý na odpovídajícím způsobem nastaveném strojním parametru. Věnujte prosím pozornost informacím od výrobce stroje.

Literatura:

Příručka k funkcím, Speciální funkce; Přípravné zpracování (V2)

Syntaxe

PROC ... PREPRO

Význam

PREPRO:	Klíčové slovo pro označení všech souborů, které mají být v rámci náběhu systému předběžně zpracovány a které patří k NC programům uloženým v adresáři cyklů.
---------	--

Načítání a vyvolávání podprogramů s přípravou

Jak podprogramy s parametry připravené při náběhu systému, tak také při volání podprogramu, se adresáři cyklů prochází ve stejné posloupnosti:

1. _N_CUS_DIR - cykly uživatele
2. _N_CMA_DIR - cykly výrobce
3. _N_CST_DIR - standardní cykly

V případě NC programů stejného názvu a s odlišnými významy se aktivuje první nalezený příkaz PROC a ostatní příkazy PROC jsou přeskakovány, aniž by bylo generováno alarmové hlášení.

2.24.2.8 Návrat zpět z podprogramu - M17

N konci podprogramu se nachází příkaz pro návrat M17 (příp. příkaz konce výrobního programu M30). Způsobuje návrat zpátky do volajícího výrobního programu na blok, který následuje za voláním podprogramu.

Poznámka

S příkazy M17 a M30 mají v NC jazyce zcela stejné postavení.

Syntaxe

```
PROC <název programu>
...
M17/M30
```

Okrajové podmínky**Vliv návratu z podprogramu na režim řízení pohybu po dráze**

Jestliže se příkaz M17 (příp. M30) nachází v bloku výrobního programu samotný, aktivní režim řízení pohybu po dráze v kanálu se tím přeruší.

Aby se přerušování režimu řízení pohybu po dráze zabránilo, je zapotřebí příkaz M17 (příp. M30) napsat v posledním bloku posuvu. Kromě toho musí být nastaven na "0" následující strojní parametr:

MD20800 \$MC_SPF_END_TO_VDI = 0 (žádný výstup příkazu M30/M17 na rozhraní NC/PLC)

Příklad:**1. Podprogram s příkazem M17 v samostatném bloku**

Programový kód	Komentář
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	
N30 M17	; návrat s přerušением režimu řízení pohybu po dráze

2. Podprogram s příkazem M17 v posledním bloku posuvu

Programový kód	Komentář
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; návrat bez přerušení režimu řízení pohybu po dráze

2.24.2.9 Návrat z podprogramu pomocí příkazu RET

Jako náhrada příkazu pro návrat z podprogramu M17 je možné v podprogramu používat také příkaz RET. Příkaz RET musí být naprogramován v samostatném bloku výrobního programu. Stejně jako příkaz M17 i příkaz RET způsobuje návrat zpátky do volajícího výrobního programu na blok, který následuje za voláním podprogramu.

Poznámka

Naprogramováním parametrů může být chování při návratu pomocí příkazu RET změněno (viz "Návrat z podprogramu s nastavitelnými parametry (RET ...) (Strana 182)").

Aplikace

Příkaz RET je potřeba používat tehdy, pokud režim řízení pohybu po dráze (G64) (G641 ... G645) nemá být návratem z podprogramu přerušen.

Předpoklady

Příkaz RET se může používat pouze v podprogramech, které nejsou definovány s atributem SAVE.

Syntaxe

```
PROC <název programu>
...
RET
```

Příklad:

Hlavní program:

Programový kód	Komentář
PROC MAIN_PROGRAM	; začátek programu
...	
N50 SUB_PROG	; volání podprogramu: SUB_PROG
N60 ...	
...	
N100 M30	; konec programu

Podprogram:

Programový kód	Komentář
PROC SUB_PROG	
...	
N100 RET	; návrat se uskuteční na blok N60 v hlavním programu

2.24.2.10 Návrat z podprogramu s nastavitelnými parametry (RET ...)

Obecně platí, že pomocí příkazu `RET` naprogramujete skok z podprogramu zpět do volajícího programu. Zpracování bude potom pokračovat řádkem programu, který následuje za voláním podprogramu. Jestliže má zpracování programu pokračovat na jiném místě, máte k dispozici následující možnosti:

- Pokračování zpracováním programu po vyvolání cyklu pro oddělování třísky v režimu dialektu ISO (po popisu kontury).
- Návrat zpět do hlavního programu z libovolné úrovně podprogramu (i po ASUP) při zpracování chybových stavů.
- Návrat zpět přes více programových úrovní pro speciální aplikace v cyklech překladače a v režimu dialektu ISO.

Za tím účelem je potřeba naprogramovat příkaz `RET` s dalšími parametry.

Směr vyhledávání

Jestliže je zadán parametr `<cílový blok>`, uskuteční se skok zpět napřed na blok, který následuje za blokem volání. Potom se vyhledá cílový blok ve směru **konce** programu, na který se uskuteční návratový skok. Pokud je vyhledávání neúspěšné, spustí se další hledání, tentokrát směrem k začátku programu.

Syntaxe

```
RET("<cílový blok>")
RET("<cílový blok>",<blok za cílovým blokem>)
RET("<cílový blok>",<blok za cílovým blokem>,<počet úrovní při návratu>)
RET("<cílový blok>",<počet úrovní při návratu>)
```

```
RET("<cílový blok>",<blok za cílovým blokem>,<počet úrovní při návratu>,<návrat na začátek programu>)
RET( , ,<počet úrovní při návratu>,<návrat na začátek programu>)
```

Význam

RET:	Konec podprogramu			
<cílový blok>:	V parametru je jako cíl skoku uveden blok, na kterém má zpracování programu pokračovat. Jestliže není naprogramován parametr <počet úrovní při návratu>, potom se cíl skoku nachází v programu, z něhož byl aktuální podprogram vyvolán. Možné údaje jsou následující:			
	<číslo bloku>	Číslo cílového bloku. Vyhledávání čísla bloku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho konci.		
	<návěští skoku>	Návěští skoku, na které se v programu uskutečňuje skok zpět, musí existovat. Vyhledávání návěští skoku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho konci.		
	<řetězec znaků>	Řetězec znaků, na který se v programu uskutečňuje skok zpět, musí existovat (např. jako název programu nebo název proměnné). Vyhledávání řetězce znaků v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho konci. Pro programování řetězce znaků v cílovém bloku platí následující pravidla: • Mezera na konci (na rozdíl od návěští skoku, které je na konci označeno ":"). • Před řetězcem znaků smí být uvedeny pouze číslo bloku a/nebo návěští skoku, žádné programové příkazy.		
<blok za cílovým blokem>:	Pomocí tohoto parametru se zadává, zda má zpracování programu pokračovat blokem, který je zadán v parametru <cílový blok>, nebo blokem, který za ním následuje.			
	Typ:	INT		
	Hodnota:	0	Skok zpět se uskuteční na blok, který je zadán v parametru <cílový blok>.	
		> 0	Skok zpět se uskuteční na blok, který následuje za blokem uvedeným v parametru <cílový blok>.	

<počet úrovní při návratu>:	V tomto parametru se zadává počet programových úrovní, o které se má skočit zpět, aby se zde našel cílový blok a aby zpracování programu pokračovalo.		
	Typ:	INT	
	Hodnota:	1	Program bude pokračovat na "aktuální programové úrovni - 1" (tedy stejně jako v případě příkazu <code>RET</code> bez parametrů).
		2	Program bude pokračovat na "aktuální programové úrovni - 2", tzn. jedna úroveň bude přeskočena.
		3	Program bude pokračovat na "aktuální programové úrovni - 3", tzn. dvě úrovně budou přeskočeny.
		...	
<Skok zpátky na začátek programu>:	Rozsah hodnot:	1 ... 15	
	Typ:	BOOL	
	Hodnota:	1	Jestliže se uskutečňuje návrat zpět do hlavního programu a jestliže je v něm aktivní režim dialektu ISO , má pokračovat ve zpracování tohoto programu od jeho začátku.
			Jestliže se uskutečňuje návrat zpět do hlavního programu a jestliže je v něm aktivní režim dialektu ISO , skočí se na začátek programu.

Poznámka

Při návratu z podprogramu, při němž je pro vyhledání cílového bloku zadán řetězec znaků, se ve volaném programu vždy napřed hledá návěští skoku.

Pokud má být cíl skoku jednoznačně definován prostřednictvím řetězce znaků, nesmí proto řetězec znaků odpovídat názvu návěští skoku, neboť potom by se při návratu z podprogramu vždy skákalo na návěští skoku a nikoli na řetězec znaků (viz příklad 2).

Okrajové podmínky

Při návratu zpět přes několik programových úrovní jsou vyhodnocovány příkazy `SAVE` jednotlivých programových úrovní.

Jestliže je při návratu zpět přes několik programových úrovní aktivní modální volání podprogramu a pokud je v některém z přeskakovaných podprogramů naprogramován příkaz pro deaktivování modálního podprogramu `MCALL`, zůstává modální podprogram i nadále aktivní.

UPOZORNĚNÍ**Chyba v programu**

Při návratu přes více programových úrovní je odpovědností uživatele, aby se postaral, že potřebná modální nastavení zůstávají i nadále v platnosti. Toho je možno dosáhnout např. naprogramováním odpovídajícího hlavního bloku.

Příklady

Příklad 1: Opětovné pokračování v hlavním programu po zpracování programu ASUP

Programování	Komentář
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 3 (zpracování programu ASUP)
...	
N19100 RET("N10900", , \$P_STACK)	; návrat z podprogramu do hlavního programu
	; \$P_STACK: aktuální programová úroveň
N10900	; cílový blok v hlavním programu
N10910 MCALL	; deaktivování modálního volání podprogramu
N10920 G0 G60 G40 M5	; inicializace dalších modálních nastavení

Příklad 2: Řetězec znaků (<STRING>) jako zadání pro vyhledávání cílového bloku

Hlavní program:

Programový kód	Komentář
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; Volání podprogramu "subProg1"
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; Volání podprogramu "subProg2"
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; "subProg1" je zde definováno jako návěští skoku.
N1560 ...	
N1600 subProg3	; Volání podprogramu "subProg3"
N1610 ...	
N1900 M30	

Podprogram subProg1:

Programový kód	Komentář
PROC subProg1	

Programový kód	Komentář
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; návrat zpět do hlavního programu na blok N1400

Podprogram subProg2:

Programový kód	Komentář
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; návrat zpět do hlavního programu na blok N1500

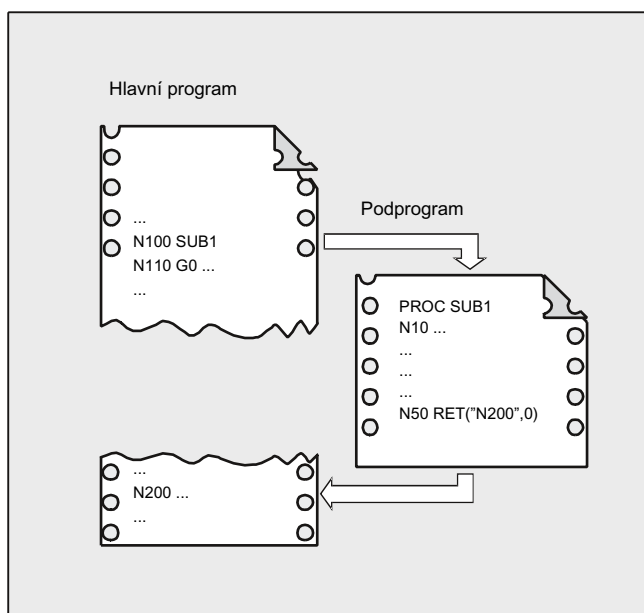
Podprogram subProg3:

Programový kód	Komentář
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; návrat zpět do hlavního programu na blok N1550

Další informace

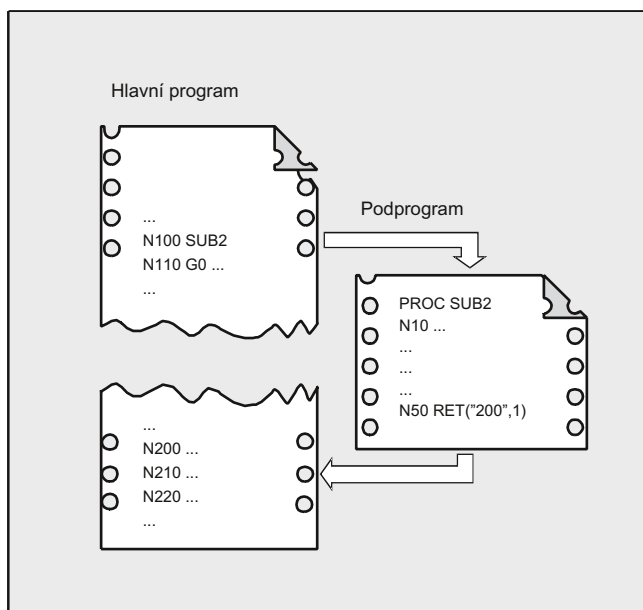
Následující obrázky ukazují výsledky různých nastavení návratových parametrů

1. <cílový blok> = "N200", <blok za cílovým blokem> = 0



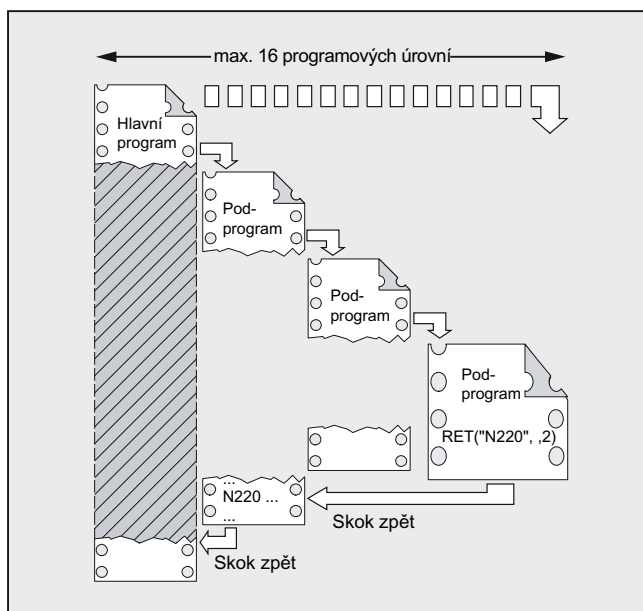
Po příkazu RET bude zpracovávání programu pokračovat blokem N200 v hlavním programu.

2. <cílový blok> = "N200", <blok za cílovým blokem> = 1



Po příkazu `RET` bude zpracovávání programu pokračovat blokem (N210), který v hlavním programu následuje za blokem N200.

3. <cílový blok> = "N220", <počet úrovní při návratu> = 2



Po příkazu `RET` budou při návratu zpět přeskočeny dvě programové úrovně a zpracovávání programu bude pokračovat blokem N220.

2.24.2.11 Návrat z podprogramu s nastavitelnými parametry (RETB ...):

Obecně platí, že pomocí příkazu `RETB` naprogramujete skok z podprogramu zpět do volajícího programu. Zpracování bude potom pokračovat řádkem programu, který následuje za voláním podprogramu. Jestliže má zpracování programu pokračovat na jiném místě, máte k dispozici následující možnosti:

- Pokračování zpracování programu po vyvolání cyklu pro oddělování třísky v režimu dialektu ISO (po popisu kontury).
- Návrat zpět do hlavního programu z libovolné úrovně podprogramu (i po ASUP) při zpracování chybových stavů.
- Návrat zpět přes více programových úrovní pro speciální aplikace v cyklech překladače a v režimu dialektu ISO.

Za tím účelem je potřeba naprogramovat příkaz `RETB` s dalšími parametry.

Směr vyhledávání

Jestliže je zadán parametr `<cílový blok>`, uskuteční se skok zpět napřed na blok, který následuje za blokem volání. Potom se vyhledá cílový blok ve směru **začátku** programu, na který se uskuteční návratový skok. Pokud je vyhledávání neúspěšné, spustí se další hledání, tentokrát směrem ke konci programu.

Syntaxe

```
RETB („<cílový blok>“)
RETB ("<cílový blok>", <blok za cílovým blokem>)
RETB ("<cílový blok>", <blok za cílovým blokem>, <počet úrovní při návratu>)
RETB ("<cílový blok>", , <počet úrovní při návratu>)
RETB ("<cílový blok>", <blok za cílovým blokem>, <počet úrovní při návratu>,
<skok zpátky na začátek programu>)
RETB( , , <počet úrovní při návratu>, <návrat na začátek programu>)
```


Význam

RETB:	Konec podprogramu			
<cílový blok>:	V parametru je jako cíl skoku uveden blok, na kterém má zpracování programu pokračovat. Jestliže není naprogramován parametr <počet úrovní při návratu>, potom se cíl skoku nachází v programu, z něhož byl aktuální podprogram vyvolán. Možné údaje jsou následující:			
	<číslo bloku>	Číslo cílového bloku. Vyhledávání čísla bloku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho začátku.		
	<návěští skoku>	Návěští skoku, na které se v programu uskutečňuje skok zpět, musí existovat. Vyhledávání návěští skoku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho začátku.		
	<řetězec znaků>	Řetězec znaků, na který se v programu uskutečňuje skok zpět, musí existovat (např. jako název programu nebo název proměnné). Vyhledávání řetězce znaků v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho začátku. Pro programování řetězce znaků v cílovém bloku platí následující pravidla: • Mezera na konci (na rozdíl od návěští skoku, které je na konci označeno ":"). • Před řetězcem znaků smí být uvedeny pouze číslo bloku a/nebo návěští skoku, žádné programové příkazy.		
<blok za cílovým blokem>:	Pomocí tohoto parametru se zadává, zda má zpracování programu pokračovat blokem, který je zadán v parametru <cílový blok>, nebo blokem, který za ním následuje.			
	Typ:	INT		
	Hodnota:	0	Skok zpět se uskuteční na blok, který je zadán v parametru <cílový blok>.	
		> 0	Skok zpět se uskuteční na blok, který následuje za blokem uvedeným v parametru <cílový blok>.	

<počet úrovní při návratu>:	V tomto parametru se zadává počet programových úrovní, o které se má skočit zpět, aby se zde našel cílový blok a aby zpracování programu pokračovalo.		
	Typ:	INT	
	Hodnota:	1	Program bude pokračovat na "aktuální programové úrovni - 1" (tedy stejně jako v případě příkazu <code>RET</code> bez parametrů).
		2	Program bude pokračovat na "aktuální programové úrovni - 2", tzn. jedna úroveň bude přeskočena.
		3	Program bude pokračovat na "aktuální programové úrovni - 3", tzn. dvě úrovně budou přeskočeny.
...			
	Rozsah hodnot:	1 ... 15	
<Skok zpátky na začátek programu>:	V tomto parametru se zadává, zda se při návratu do hlavního programu, ve kterém je aktivní režim dialektu ISO , má pokračovat ve zpracování tohoto programu od jeho začátku.		
	Typ:	BOOL	
	Hodnota:	1	Jestliže se uskutečňuje návrat zpět do hlavního programu a jestliže je v něm aktivní režim dialektu ISO , skočí se na začátek programu.

Poznámka

Při návratu z podprogramu, při němž je pro vyhledání cílového bloku zadán řetězec znaků, se ve volaném programu vždy napřed hledá návěští skoku.

Pokud má být cíl skoku jednoznačně definován prostřednictvím řetězce znaků, nesmí proto řetězec znaků odpovídat názvu návěští skoku, neboť potom by se při návratu z podprogramu vždy skákalo na návěští skoku a nikoli na řetězec znaků (viz příklad 2).

Okrajové podmínky

Při návratu zpět přes několik programových úrovní jsou vyhodnocovány příkazy `SAVE` jednotlivých programových úrovní.

Jestliže je při návratu zpět přes několik programových úrovní aktivní modální volání podprogramu a pokud je v některém z přeskakovaných podprogramů naprogramován příkaz pro deaktivování modálního podprogramu `MCALL`, zůstává modální podprogram i nadále aktivní.

UPOZORNĚNÍ**Chyba v programu**

Při návratu přes více programových úrovní je odpovědností uživatele, aby se postaral, že potřebná modální nastavení zůstávají i nadále v platnosti. Toho je možno dosáhnout např. naprogramováním odpovídajícího hlavního bloku.

Příklad

Programový kód	Komentář
BEISPIEL.MPF	
...	
N3000 START_CYC(param1, param2, ...)	
N3010 TECH_CYC1(param1, param2, ...)	
N3020 TECH_CYC2(param1, param2, ...)	
N3030 TECH_CYC3(param1, param2, ...)	
N3040 END_CYC(param1, param2, ...)	
N3040 END_CYC (param1, param2, ...)	
N3050 ...	
N4500 START_CYC(param11, param12, ...)	
N4510 ...	
N4590 END_CYC(param11, param12, ..)	
N5000 ...	
...	
N6000 M30	

Programový kód	Komentář
PROC END_CYC(...)	; volání v hlavním programu, řádek N3040
N10000 ...	
N15000 if status == 1	
N15010 RETB("START_CYC")	; skok zpět do volajícího programu BEISPIEL.MPF ; vyhledat řetězec znaků "START_CYC" ; směr vyhledávání: dozadu směrem k začátku programu ; pokračování zpracování programu na řádce N3000
N15020 endif	
N15030 if status == 0	
N15040 RET	; skok zpět do volajícího programu BEISPIEL.MPF ; pokračování zpracování programu na řádce N3050
N15050 endif	
N16000 RET("START_CYC")	; skok zpět do volajícího programu BEISPIEL.MPF ; vyhledat řetězec znaků "START_CYC" ; směr vyhledávání: dopředu směrem ke konci programu ; pokračování zpracování programu na řádce N4500
N17060 RETB	; skok zpět do volajícího programu BEISPIEL.MPF ; pokračování zpracování programu na řádce N3050 ; příkaz RETB bez zadání parametru je identický s příkazem RET

2.24.3 Volání podprogramu

2.24.3.1 Volání podprogramu bez předávání parametrů

Volání podprogramu se uskutečňuje buď pomocí adresy L a číslem podprogramu nebo zadáním názvu podprogramu.

Jako podprogram je možné volat i hlavní program. Konec programu M2 nebo M30 uvedený v hlavním programu bude v tomto případě vyhodnocován jako M17 (konec programu s návratem zpátky do volajícího programu).

Poznámka

Analogicky je možné spustit podprogram jako program hlavní.

Strategie řídicího systému při vyhledávání:

Existuje *_MPF ?

Existuje *_SPF ?

Z toho vyplývá: Jestliže jsou název volaného podprogramu a název hlavního programu identické, potom bude znovu volán program, který spustil volání. Tento zpravidla nežádoucí efekt je nutno eliminovat jednoznačnou volnou názvu hlavního programu a podprogramu.

Poznámka

Podprogramy, které nevyžadují žádné předávání parametrů, mohou být volány také z inicializačního souboru.

Syntaxe

L<číslo>/<název programu>

Poznámka

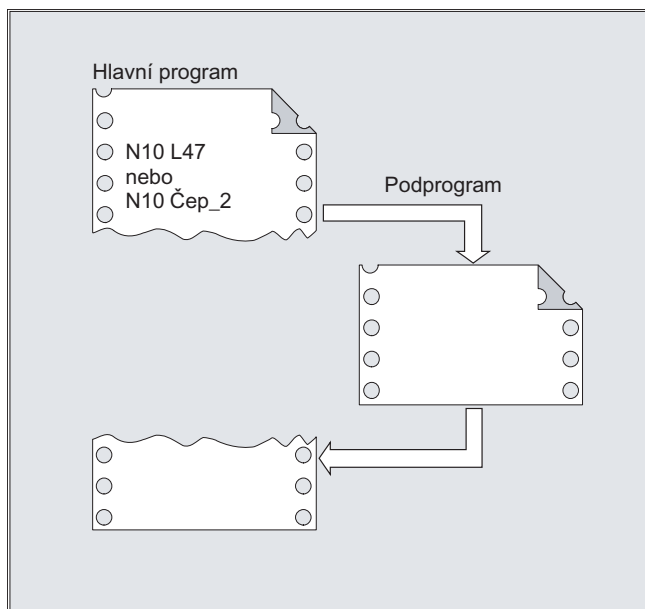
Volání podprogramu musí být vždy naprogramováno v samostatném NC-bloku.

Význam

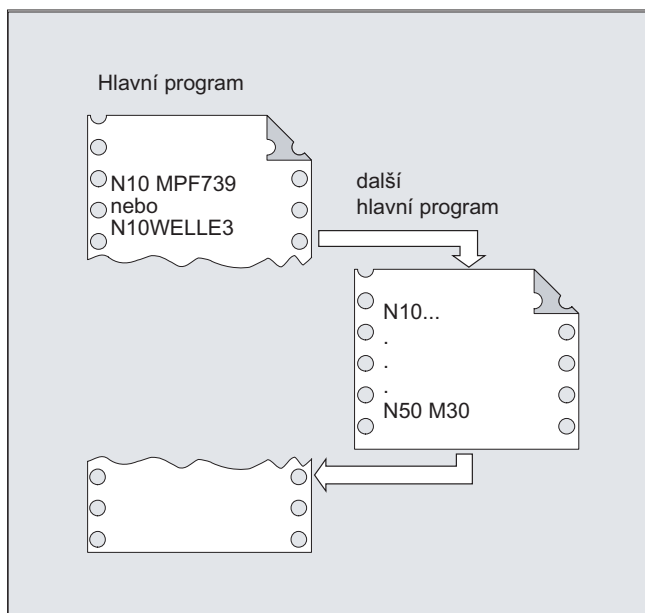
L:	Adresa pro volání podprogramu	
<číslo>:	Číslo podprogramu	
	Typ:	INT
	Hodnota:	Maximálně 7 míst v desítkové soustavě Pozor: Nuly na začátku mají při vyhodnocování názvů svůj význam (--> L123, L0123 a L00123 jsou tři různé podprogramy).
<název programu>:	Název podprogramu (nebo hlavního programu)	

Příklady

Příklad 1: Volání podprogramu bez předávání parametrů



Příklad 2: Volání hlavního programu jako podprogramu



Viz také

Podprogram bez předávání parametrů (Strana 165)

2.24.3.2 Volání podprogramu s předáváním parametrů (EXTERN)

Při volání podprogramu s předáváním parametrů mohou být proměnné nebo hodnoty předávány přímo (nikoli u parametrů typu VAR).

Podprogramy s předáváním parametrů musí být před voláním v hlavním programu s příkazem EXTERN deklarovány (např. na začátku programu). Uvádí se přitom název podprogramu a typy proměnných v posloupnosti, v jaké jsou předávány.

UPOZORNĚNÍ

Nebezpečí záměny

Jak typy proměnných, tak i posloupnost při předávání se musí shodovat s definicemi, které byly v podprogramu zadány v příkazu PROC. Názvy parametrů se mohou v hlavním programu a v podprogramu lišit.

Syntaxe

```
EXTERN <název programu>(<typ_par1>,<typ_par2>,<typ_par3>)
...
<název programu>(<hodnota_par1>,<hodnota_par2>,<hodnota_par3>)
```

Poznámka

Volání podprogramu musí být vždy naprogramováno v samostatném NC-bloku.

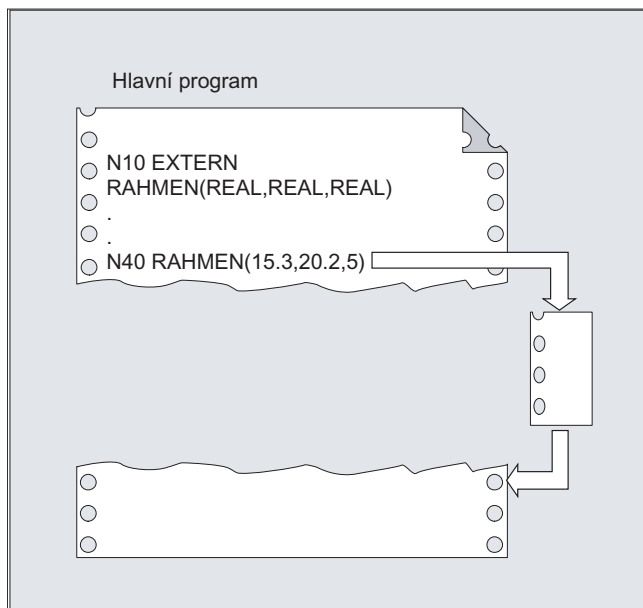
Význam

<název programu>:	Název podprogramu
EXTERN:	Klíčové slovo pro identifikaci podprogramu s předáváním parametrů Upozornění: Příkaz EXTERN musí být uváděn jen tehdy, pokud se podprogram nachází v adresáři obrobku nebo v globálním adresáři podprogramů. Při volání cyklů se příkaz EXTERN používat nemusí.
<typ_par1>,<typ_par2>,<typ_par3>:	Typy proměnných předávaných parametrů v pořadí jejich předávání
<hodnota_par1>,<hodnota_par2>,<hodnota_par3>:	Hodnota proměnné pro předávaný parametr

Příklady

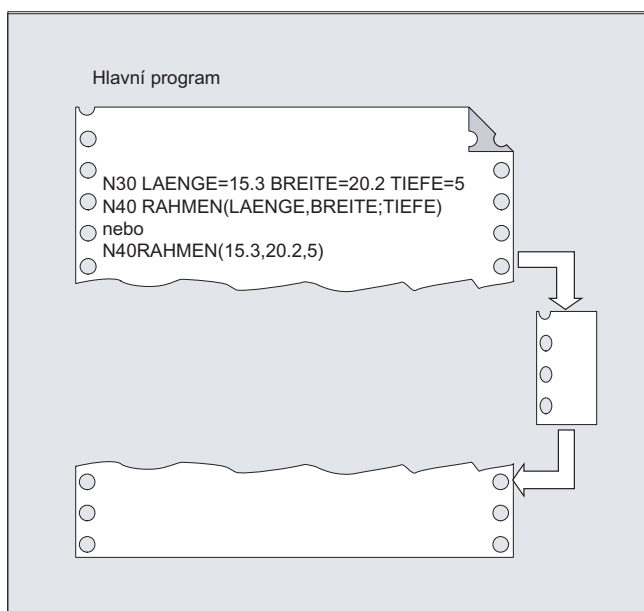
Příklad 1: Volání podprogramu s předcházející deklarací

Programový kód	Komentář
N10 EXTERN RAHMEN (REAL, REAL, REAL)	; Zadání podprogramu.
...	
N40 RAHMEN(15.3, 20.2, 5)	; Volání podprogramu s předáváním parametrů.



Příklad 2: Volání podprogramu bez deklarace

Programový kód	Komentář
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)

**Viz také**

Podprogram s předáváním parametrů Call-by-Value (PROC) (Strana 166)

Podprogram s předáváním parametrů Call-by-Reference (PROC, VAR) (Strana 168)

2.24.3.3 Počet opakování programu (P)

Jestliže má být podprogram zpracován několikrát po sobě, je možné v bloku s voláním podprogramu naprogramovat pomocí adresy **P** požadovaný počet opakování tohoto programu.

! POZOR

Volání podprogramu s opakováním programu a s předáváním parametrů

Parametry se předávají pouze při volání programu, resp. při prvním průchodu. Pro další opakování zůstávají tyto parametry nezměněné. Jestliže si přejete, aby se při opakováních programu tyto parametry měnily, musíte v podprogramu definovat odpovídající opatření.

Syntaxe

`<název programu> P<hodnota>`

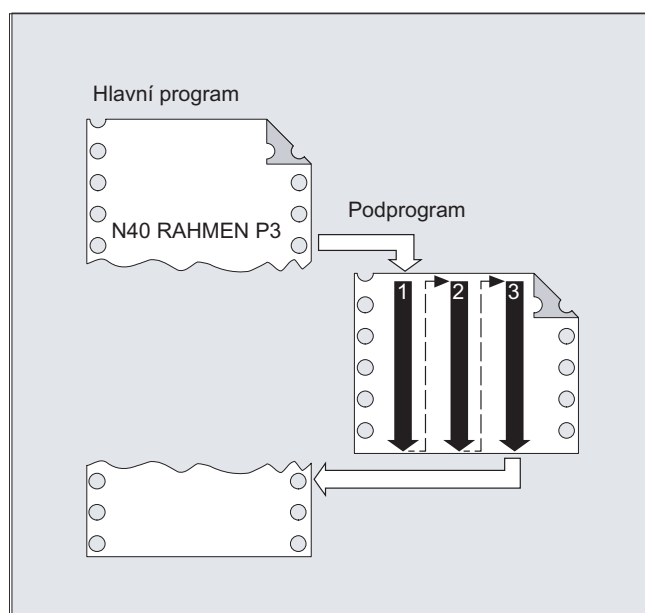
Význam

<code><název programu>:</code>	Volání podprogramu
<code>P:</code>	Adresa pro naprogramování opakování programu

<hodnota>:	Počet opakování programu	
	Typ:	INT
	Rozsah hodnot:	1 ... 9999 (bez znaménka)

Příklad:

Programový kód	Komentář
...	
N40 RAHMEN P3	; podprogram RAHMEN má být zpracován třikrát po sobě
...	

**2.24.3.4 Modální volání podprogramu (MCALL)**

Prostřednictvím modálního volání podprogramu `MCALL(<název programu>)` není zadáný podprogram vyvolán okamžitě. Místo toho se jeho volání uskutečňuje od tohoto okamžiku ve výrobním programu automaticky po každém bloku s příkazem pohybu po dráze. A to i na různých programových úrovních.

Poznámka

Při zpracování programu se vždy uplatňuje jen poslední modální volání podprogramu `MCALL(<název programu>)`. Aktuální modální volání podprogramu nahrazuje to, které bylo v platnosti předtím.

Pokud jsou do podprogramu předávány nějaké parametry, uskutečňuje se předávání parametrů jen při volání `MCALL(<název programu>(Par1, Par2, ...))`.

UPOZORNĚNÍ**Modální volání podprogramu bez pohybu po dráze**

Modální podprogram se vyvolává v následujících situacích i bez toho, že by byl naprogramován nějaký pohyb po dráze:

- Naprogramování adres S nebo F, pokud je aktivní příkaz G0 nebo G1.
- Pokud se v bloku nachází samotný příkaz G0 nebo G1 nebo pokud byl naprogramován spolu s dalšími příkazy G-funkcí.

Syntaxe

```
MCALL <název programu>
...
MCALL
```

Význam

MCALL <název programu>:	Aktivování funkce „Modální volání podprogramu“
<název programu>:	Název podprogramu
MCALL:	Prostřednictvím příkazu MCALL bez zadání názvu programu je funkce "Modální volání podprogramu" deaktivována.

Okrajové podmínky**ASUP**

Jestliže je zpracování výrobního programu přerušeno příkazem ASUP (viz kapitola "Rutiny přerušení (ASUP) (Strana 128)."), v tomto stavu ASUP se žádná modální volání podprogramů neprovádějí.

Pokud je příkaz ASUP spuštěn, když je kanál ve stavu „Reset“, chová se vzhledem k modálním voláním podprogramů jako normální výrobní program.

Cyklus pro výměnu nástroje

Jestliže je v cyklu pro výměnu nástroje funkce "Modální volání podprogramu" deaktivována, je potřeba zajistit, aby byl cyklus pro výměnu nástroje vyvoláván případně také po vyhledávání bloku implicitně prostřednictvím vyhledávání bloku s příkazem ASUP nebo manuálně pomocí přepsání v paměti. V této situaci nesmí být funkce „Modální volání podprogramu“ deaktivována, protože jinak by byl výsledek vyhledávání bloku nesprávný. Z tohoto důvodu se doporučuje deaktivování funkce "Modální volání podprogramu" v cyklu pro výměnu nástroje naprogramovat následujícím způsobem:

Programový kód	Komentář
...	

Programový kód	Komentář
IF \$AC_ASUP == 0	; volání se neuskutečňuje prostřednictvím vyhledávání bloku s příkazem ASUP nebo přepsáním v paměti
MCALL	; deaktivování funkce modálního volání podprogramu
ENDIF	
...	

Příklady

Příklad 1

Programový kód	Komentář
N10 G0 X0 Y0	
N20 MCALL L70	; aktivování modálního volání podprogramu L70
N30 X10 Y10	; najíždění na bod X10 Y10, potom se vyvolá program L70
N40 X20 Y20	; najíždění na bod X20 Y20, potom se vyvolá program L70
...	
N100 MCALL	; deaktivování funkce modálního volání podprogramu
N110 X0 Y0	; najíždění na bod X0 Y0, podprogram L70 se nevyvolává

Příklad 2

Programový kód
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

V tomto příkladu se nacházejí následující NC bloky s naprogramovanými dráhovými osami v podprogramu L80. Podprogram L70 se vyvolává prostřednictvím podprogramu L80.

2.24.3.5 Nepřímé volání podprogramu (CALL)

V závislosti na daných podmínkách mohou být na jednom místě vyvolávány různé podprogramy. Za tím účelem se název podprogramu ukládá do proměnné typu STRING. Volání podprogramu se pak uskutečňuje příkazem CALL a názvem proměnné.

Poznámka

Nepřímé volání podprogramu je možné pouze pro podprogramy bez předávání parametrů. Pro účely přímého volání podprogramu uložte jeho název do konstanty typu STRING.

Syntaxe

CALL <název programu>

Význam

CALL:	Příkaz pro nepřímé volání podprogramu	
<název programu>:	Název podprogramu (proměnná nebo konstanta)	
	Typ:	STRING

Příklad:

Přímé volání s konstantou typu STRING:

Programový kód	Komentář
...	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Podprogram TEIL1 s přímým voláním typu CALL.
...	

Nepřímé volání pomocí proměnné:

Programový kód	Komentář
...	
DEF STRING[100] PROGNAME	; Definice proměnné.
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Přiřazení podprogramu TEIL1 proměnné PROGNAME.
CALL PROGNAME	; Nepřímé volání podprogramu TEIL1 pomocí příkazu CALL a proměnné PROGNAME.
...	

2.24.3.6 Nepřímé volání podprogramu se zadáním úseku programu, který á být zpracován (CALL BLOCK ... TO ...)

Pomocí příkazu CALL a kombinace klíčových slov BLOCK ... TO se nepřímo vyvolává podprogram a v něm se zpracovává úsek, který začíná počátečním a končí koncovým návěštím.

Syntaxe

```
CALL <název programu> BLOCK <počáteční návěští> TO <koncové návěští>
CALL BLOCK <počáteční návěští> TO <koncové návěští>
```

Význam

CALL:	Příkaz pro nepřímé volání podprogramu	
<název programu>:	Název podprogramu (proměnná nebo konstanta), který obsahuje úsek programu, který má být zpracován (údaj je nepovinný).	
	Typ:	STRING
	Upozornění: Jestliže parametr <název programu> není naprogramován, úsek programu označený návěštími <počáteční návěští> a <koncové návěští> bude vyhledán v aktuálním programu a bude zpracován.	
BLOCK ... TO ... :	Kombinace klíčových slov pro nepřímé zpracování úseku programu	
<počáteční návěští>:	Proměnná, která odkazuje na začátek úseku programu, který má být zpracován.	
	Typ:	STRING
<koncové návěští>:	Proměnná, která odkazuje na konec úseku programu, který má být zpracován.	
	Typ:	STRING

Příklad:

Hlavní program:

Programový kód	Komentář
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; definice proměnných pro počáteční a koncové návěští
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; nepřímé volání podprogramu a označení úseku programu, který má být zpracován
...	

Podprogram:

Programový kód	Komentář
PROC CONTUR_1 ...	
LABEL_1	; počáteční návěští: Začátek zpracování úseku programu
N1000 G1 ...	
...	
LABEL_2	; koncové návěští: Konec zpracování úseku programu
...	

2.24.3.7 Nepřímé volání programu naprogramovaného v jazyce ISO (ISOCALL)

Pomocí nepřímého volání programu pomocí příkazu `ISOCALL` je možné vyvolat program naprogramovaný v jazyce ISO. Přitom se aktivuje režim ISO nastavený ve strojních parametrech. Na konci programu se znovu stane platným původní režim zpracování. Jestliže není ve strojních parametrech nastaven žádný režim ISO, uskuteční se volání podprogramu v režimu Siemens.

Pokud budete potřebovat další informace k režimu ISO, viz:

Literatura:

Příručka k funkcím, Dialekty ISO

Syntaxe

`ISOCALL <název programu>`

Význam

ISOCALL:	Klíčové slovo pro nepřímé volání podprogramu, při kterém se aktivuje režim ISO nastavený ve strojních parametrech.
<název programu>:	Název programu naprogramovaného v jazyce ISO (proměnná nebo konstanta typu STRING)

Příklad: Vyvolání kontury s programováním cyklu v režimu ISO

Programový kód	Komentář
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 (cyklus) od firmy Siemens
...	
N2000 R11 = \$AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; zpracování programu 0122 v režimu ISO
...	
N2400 M30	

2.24.3.8 Volání podprogramu s udáním cesty a parametrů (PCALL)

Pomocí příkazu `PCALL` mohou být vyvolávány podprogramy s absolutním zadáním cesty a s předáváním parametrů.

Syntaxe

PCALL <cesta/název programu>(<parametr 1>,...,<parametr n>)

Význam

PCALL:	Klíčové slovo pro volání podprogramu s absolutním zadáním cesty.
<cesta/název programu>:	Absolutní zadání cesty včetně názvu podprogramu. Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 219)". Jestliže žádná absolutní cesta nebyla zadána, chová se příkaz PCALL stejně jako standardní volání s identifikátorem podprogramu. Název programu se zadává bez předpony a bez přípony souboru. Jestliže má být název programu naprogramován s úvodním řetězcem a s příponou souboru, potom musí být explicitně vysvětlen i s předponou a s příponou pomocí příkazu EXTERN.
<parametr 1>, ...:	Aktuální parametry v souladu s příkazem PROC v podprogramu.

Příklad:

Programový kód
PCALL/_N_WKS_DIR/_N_WELLE_WPD/WELLE (parametr1,parametr2,...)

2.24.3.9**Rozšíření cesty pro vyhledávání při volání podprogramu (CALLPATH)**

Pomocí příkazu CALLPATH je možné rozšířit cestu pro vyhledávání při volání podprogramu. Tímto způsobem mohou být vyvolávány také podprogramy z adresáře obrobku, který není vybrán, aniž by bylo nutno udávat úplný absolutní název cesty tohoto podprogramu.

Další možnost použití se nabízí v provozním režimu EES „EES bez GDIR“, jestliže se pro ukládání globálních podprogramů používá adresář v externí paměti pro programy. V tomto případě je možné pomocí příkazu CALLPATH rozšířit cestu pro vyhledávání o tento adresář pro podprogramy.

Rozšíření cesty pro vyhledávání se provádí před zadáním pro uživatelské cykly (_N_CUS_DIR).

Následující události způsobí, že se rozšíření cesty pro vyhledávání opět deaktivuje:

- Příkaz CALLPATH s mezerami
- Příkaz CALLPATH bez parametrů
- Konec výrobního programu
- Reset

Syntaxe

CALLPATH("<název cesty>")

Význam

CALLPATH:	Klíčové slovo pro programovatelné rozšíření cesty pro vyhledávání Musí být naprogramováno na samostatném řádku výrobního programu.
<název cesty>:	Konstanta / proměnná typu STRING. Obsahuje absolutní údaj cesty do adresáře, o který má být cesta pro vyhledávání rozšířena. Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 219)".

Příklad:

Cesta pro vyhledávání by měla být rozšířena o adresář příslušného obrobku:

Programový kód
...
CALLPATH("/_N_WKS_DIR/_N_MYWPD_WPD")
...

Tímto příkazem se vytvoří následující cesta pro vyhledávání (položka 5 je nová):

1. Aktuální adresář/název
2. Aktuální adresář/název_SPF
3. Aktuální adresář/název_MPF
4. //NC:/_N_SPF_DIR/název_SPF
5. /_N_WKS_DIR/_N_MYWPD_WPD/název_SPF
6. /_N_CUS_DIR/název_SPF
7. /_N_CMA_DIR/název_SPF
8. /_N_CST_DIR/název_SPF

Okrajové podmínky

- Příkaz CALLPATH kontroluje, zda naprogramovaný název cesty skutečně existuje. Vyskytne-li se chyba, zpracování výrobního programu se přeruší a aktivuje se alarm korekčního bloku 14009.
- Příkaz CALLPATH může být naprogramován i v souborech INI. Potom je v platnosti po celou dobu zpracovávání souboru INI (soubor WPD-INI nebo inicializační program pro data aktivní v NC systému, např. framy v 1. kanálu _N_CH1_UFR_INI). Pak se znovu obnoví původní cesta pro vyhledávání.

2.24.3.10 Zpracovávání externího podprogramu (840D sl) (EXTCALL)

Pomocí příkazu EXTCALL je možné výrobní program načíst a zpracovat z externí paměti.

Jako externí paměť jsou k dispozici:

- Lokální jednotka
- Síťová jednotka
- Jednotka připojená přes USB

Poznámka

Jako rozhraní pro zpracování externího programu nacházejícího se na jednotce připojené přes USB se smí používat pouze rozhraní USB na ovládacím panelu, příp. na TCU.

UPOZORNĚNÍ

Poškození nástroje/obrobku kvůli flash-disku s USB

Pro zpracovávání externích podprogramů nedoporučujeme používat USB flash disky. Přerušení komunikace s USB Flash diskem v průběhu zpracovávání výrobního programu kvůli komplikacím s kontakty, výpadkům, přerušením v důsledku nárazu nebo kvůli vytažení v důsledku neopatrnosti má za následek okamžité přerušení zpracovávání. Nástroj a/nebo obrobek by se přitom mohly poškodit.

Předdefinované nastavení cesty k externímu programu

Cesta do externího programového adresáře může být předem určena pomocí nastavovaného parametru:

SD42700 \$SC_EXT_PROG_PATH

Spolu s cestou a s identifikátorem programu, které jsou zadány při volání pomocí příkazu `EXTCALL`, tak vzniká celková cesta k volanému výrobnímu programu.

Poznámka

Parametry

Při vyvolávání externího programu mu není možné předávat žádné parametry.

Syntaxe

```
EXTCALL("<cesta/><název programu>")
```

Význam

EXTCALL:	Příkaz pro vyvolání externího podprogramu	
"<cesta/><název programu>":	Konstanta / proměnná typu STRING	
	<cesta/>:	Absolutní nebo relativní zadání cesty (není nutné)
	<název programu>:	Název programu se zadává bez předpony "_N_". Příponu souboru ("MPF", "SPF") je možné spolu se znaky "_" nebo "." v názvu programu uvést (není nutné). Příklad: "WELLE" "WELLE_SPF" "WELLE.SPF"

Zadání cesty: Zkrácené popisy

Při zadávání cesty se mohou používat následující zkrácené popisy:

- Lokální jednotka: "LOCAL_DRIVE:"
- CF karta: "CF_CARD:"
- Jednotka připojená přes USB (ovládací panel): "USB:"

Jako alternativu můžete používat také zkrácená označení „CF_CARD:“ a „LOCAL_DRIVE:“.

Příklad:

Zpracovávání z lokální jednotky

Hlavní program "MAIN.MPF" se nachází v paměti NC systému a je vybrán pro zpracování:

Podprogram "SP_1"

Externí podprogram "SP_1.SPF", příp. "SP_1.MPF", se nachází na lokální jednotce v adresáři "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Cestu do externího adresáře programů je zapotřebí nastavit pomocí:

SD42700 \$SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

Poznámka

Údaj cesty pro volání externích podprogramů:

- Bez použití předdefinovaného nastavení: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- S použitím předdefinovaného nastavení: "SP_1"

Podprogram "SP_2"

Externí podprogram "SP_2.SPF", příp. "SP_2.MPF" se nalézá v adresáři WKS.DIR/WST1.WPD na jednotce připojené přes USB. Pro cestu do podprogramu "SP_1" se použije předdefinované nastavení cesty do externího adresáře programů a také v hlavním programu se neuvádí. Z tohoto důvodu musí být při volání podprogramu "SP_2" uvedena plná cesta.

Hlavní program "MAIN"

Programový kód
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30

Další informace**Volání příkazem EXTCALL s absolutním udáním cesty**

Pokud je podprogram k dispozici v adresáři podle zadané cesty, voláním příkazem `EXTCALL` se spustí jeho zpracování. Pokud podprogram v adresáři podle zadané cesty k dispozici není, zpracování programu se voláním pomocí příkazu `EXTCALL` zastaví.

Volání příkazem EXTCALL s relativním udáním cesty / bez udání cesty

V případě volání pomocí příkazu `EXTCALL` s relativním zadáním cesty, příp. bez udání cesty, jsou stávající programové paměti prohledávány podle následujícího vzoru:

1. Jestliže je v nastavovaném parametru `SD42700 $SC_EXT_PROG_PATH` předem stanovena nějaká cesta, bude po zadání volání pomocí příkazu `EXTCALL` (název programu, příp. s relativním udáním cesty) vyhledávání začínat v rámci této cesty. Absolutní cesta potom vznikne zřetěžením následujících posloupností znaků:
 - Předem definované udání cesty v parametru `SD42700 $SC_EXT_PROG_PATH`
 - Oddělovací znak "/"
 - Zadání cesty a název podprogramu v příkazu `EXTCALL`
2. Pokud nebyl pomocí postupu uvedeného v bodě 1. nalezen, jsou prohledány adresáře v uživatelské paměti.

Vyhledávání skončí, jakmile je podprogram poprvé nalezen. Pokud podprogram není nalezen, zpracování programu se voláním pomocí příkazu `EXTCALL` zastaví.

Nastavitelná paměť pro načítání (vyrovnávací paměť FIFO)

Pro zpracovávání externích podprogramů je zapotřebí vyrovnávací paměť pro načítání. Velikost této paměti je předem nastavena na 30 kbyťů a tato velikost může být změněna pouze výrobcem stroje (prostřednictvím `MD18360 MM_EXT_PROG_BUFFER_SIZE`).

Poznámka**Podprogramy s příkazy skoků**

U externích podprogramů, které obsahují příkazy skoku (`GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF` atd.), se musí cíle skoků nacházet uvnitř paměti pro načítání tohoto programu.

Poznámka**Programy ShopMill/ShopTurn**

Programy ShopMill a ShopTurn musí být kvůli popisům kontur, které jsou vkládány až na konec souboru, uloženy v paměti pro načítání programů celé.

Pro paralelně zpracovávané externí podprogramy je rovněž zapotřebí vlastní vyrovnávací paměť pro načítání.

Reset / Konec programu / POWER ON

Operace RESET a POWER ON způsobují přerušení externího volání podprogramu a vymazání obsahu vyrovnávací paměti, kam se program načítá.

Podprogram vybraný pro "zpracování z externího zdroje" zůstává pro tuto funkci vybrán, i když je proveden RESET, POWER ON nebo nastane konec programu. Toto chování se neliší vůči interně vybraným programům, neboť externí paměť programů zůstává i nadále k dispozici.

Literatura

Pokud budete potřebovat další informace týkající se funkce "Zpracování z externího zdroje", viz:

Příručka k funkcím, Základní funkce; BAG, kanál, zpracování programu, chování při resetu (K1)

2.24.3.11 Zpracovávání externího podprogramu (828D) (EXTCALL)

Pomocí příkazu `EXTCALL` je možné výrobní program načíst a zpracovat z externí paměti.

Jako externí paměť jsou k dispozici:

- Uživatelská CF karta
- Síťová jednotka
- Jednotka připojená přes USB

Poznámka

Jako rozhraní pro zpracování externího programu nacházejícího se na jednotce připojené přes USB se smí používat pouze rozhraní USB na ovládacím panelu (PPU).

UPOZORNĚNÍ**Poškození nástroje/obrobku kvůli flash-disku s USB**

Pro zpracovávání externích podprogramů nedoporučujeme používat USB flash disky. Přerušení komunikace s USB Flash diskem v průběhu zpracovávání výrobního programu kvůli komplikacím s kontakty, výpadkům, přerušením v důsledku nárazu nebo kvůli vytažení v důsledku neopatrnosti má za následek okamžité přerušení zpracovávání. Nástroj a/nebo obrobek by se přitom mohly poškodit.

Předdefinované nastavení cesty k externímu programu

Cesta do externího programového adresáře může být předem určena pomocí nastavovaného parametru:

SD42700 \$SC_EXT_PROG_PATH

Spolu s cestou a s identifikátorem programu, které jsou zadány při volání pomocí příkazu EXTCALL, tak vzniká celková cesta k volanému výrobnímu programu.

Poznámka**Parametr**

Při vyvolávání externího programu mu není možné předávat žádné parametry.

Syntaxe

```
EXTCALL("<cesta/><název programu>")
```

Význam

EXTCALL:	Příkaz pro vyvolání externího podprogramu	
"<cesta/><název programu>":	Konstanta / proměnná typu STRING	
	<cesta/>:	Absolutní nebo relativní zadání cesty (není nutné)
	<název programu>:	Název programu se zadává bez předpony "_N_". Příponu souboru ("MPF", "SPF") je možné spolu se znaky "_" nebo "." v názvu programu uvést (není nutné). Příklad: "WELLE" "WELLE_SPF" "WELLE.SPF"

Zadání cesty: Zkrácené popisy

Při zadávání cesty se mohou používat následující zkrácené popisy:

- Uživatelská CF karta: "**CF_CARD**:"
- Jednotka připojená přes USB (ovládací panel): "**USB**:"

Příklad:

Hlavní program "MAIN.MPF" se nachází v paměti NC systému a je vybrán pro zpracování:

Podprogram "SP_1"

Externí podprogram "SP_1.SPF", příp. "SP_1.MPF" se nalézá na uživatelské CF kartě v adresáři "/WKS.DIR/WST1.WPD".

Cestu do externího adresáře programů je zapotřebí nastavit pomocí:

SD42700 \$SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

Poznámka

Údaj cesty pro volání externích podprogramů:

- Bez použití předdefinovaného nastavení: "CF_CARD:WKS.DIR/WST1.WPD/SP_1"
- S použitím předdefinovaného nastavení: "SP_1"

Podprogram "SP_2"

Externí podprogram "SP_2.SPF", příp. "SP_2.MPF" se nalézá v adresáři WKS.DIR/WST1.WPD na jednotce připojené přes USB. Pro cestu do podprogramu "SP_1" se použije předdefinované nastavení cesty do externího adresáře programů a také v hlavním programu se neuvádí. Z tohoto důvodu musí být při volání podprogramu "SP_2" uvedena plná cesta.

Hlavní program "MAIN"

Programový kód

```
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

Další informace

Volání příkazem EXTCALL s absolutním udáním cesty

Pokud je podprogram k dispozici v adresáři podle zadané cesty, voláním příkazem EXTCALL se spustí jeho zpracování. Pokud podprogram v adresáři podle zadané cesty k dispozici není, zpracování programu se voláním pomocí příkazu EXTCALL zastaví.

Volání příkazem EXTCALL s relativním udáním cesty / bez udání cesty

V případě volání pomocí příkazu EXTCALL s relativním zadáním cesty, příp. bez udání cesty, jsou stávající programové paměti prohledávány podle následujícího vzoru:

1. Jestliže je v nastavovaném parametru SD42700 \$SC_EXT_PROG_PATH předem stanovena nějaká cesta, bude po zadání volání pomocí příkazu EXTCALL (název programu, příp. s relativním udáním cesty) vyhledávání začínat v rámci této cesty. Absolutní cesta potom vznikne zřetěžením následujících posloupností znaků:
 - Předem definované udání cesty v parametru SD42700 \$SC_EXT_PROG_PATH
 - Oddělovací znak "/"
 - Zadání cesty a název podprogramu v příkazu EXTCALL
2. Pokud nebyl pomocí postupu uvedeného v bodě 1. nalezen, jsou prohledány adresáře v uživatelské paměti.

Vyhledávání skončí, jakmile je podprogram poprvé nalezen. Pokud podprogram není nalezen, zpracování programu se voláním pomocí příkazu `EXTCALL` zastaví.

Nastavitelná paměť pro načítání (vyrovnávací paměť FIFO)

Pro zpracovávání externích podprogramů je zapotřebí vyrovnávací paměť pro načítání. Velikost této vyrovnávací paměti pro načítání je předem nastavena (viz MD18360 `MM_EXT_PROG_BUFFER_SIZE`).

Poznámka

Podprogramy s příkazy skoků

U externích podprogramů, které obsahují příkazy skoku (`GOTO`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF` atd.), se musí cíle skoků nacházet uvnitř paměti pro načítání tohoto programu.

Poznámka

Programy ShopMill/ShopTurn

Programy ShopMill a ShopTurn musí být kvůli popisům kontur, které jsou vkládány až na konec souboru, uloženy v paměti pro načítání programů celé.

Pro paralelně zpracovávané externí podprogramy je rovněž zapotřebí vlastní vyrovnávací paměť pro načítání.

Reset / Konec programu / POWER ON

Operace `RESET` a `POWER ON` způsobují přerušení externího volání podprogramu a vymazání obsahu vyrovnávací paměti, kam se program načítá.

Podprogram vybraný pro "zpracování z externího zdroje" zůstává pro tuto funkci vybrán, i když je proveden `RESET`, `POWER ON` nebo nastane konec programu. Toto chování se neliší vůči interně vybraným programům, neboť externí paměť programů zůstává i nadále k dispozici.

Literatura

Pokud budete potřebovat další informace týkající se funkce "Zpracování z externího zdroje", viz:

Příručka Popis funkcí, Základní funkce; BAG, kanál, zpracování programu, chování při resetu (K1)

2.25 Technika maker (DEFINE ... AS)

UPOZORNĚNÍ

Technika maker zvyšuje komplexnost programování.

Pomocí techniky maker je možné programovací jazyk řídicího systému silně změnit. Technika maker se smí používat jen s velkou opatrností.

Makro představuje posloupnost jednotlivých příkazů, kterou jsou soustředěny do nového celku, který má vlastní název. Při vyvolání makra v průběhu zpracování programu jsou postupně zpracovány příkazy, které byly pod název makra naprogramovány.

V závislosti na oblasti jejich platnosti, tzn. oblasti, ve které definice makra platí, existují následující kategorie maker:

- **Lokální makra**
Lokální makra jsou makra, která jsou definována na začátku NC programu, který v okamžiku zpracovávání není hlavním programem. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídicího systému. Přístup k lokálním makrům je možný pouze v rámci NC programu, v němž jsou definovány.
- **Makra globální v daném programu**
Programově globální makra jsou makra, která jsou definována na začátku NC programu, který se používá jako hlavní program. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídicího systému. Přístup k programově globálním makrům je k dispozici v hlavním programu a ve všech podprogramech.

Poznámka

Dostupnost programových globálních maker

Programově globální makra definovaná v hlavním programu jsou k dispozici i v podprogramech jen tehdy, pokud je nastaven následující strojní parametr:

MD11120 \$MN_LUD_EXTENDED_SCOPE = 1

Pokud je MD11120 = 0, jsou programově globální makra definovaná v hlavním programu k dispozici pouze v tomto hlavním programu.

- **Globální makra**
Globální makra jsou globálními makry v NC systému, příp. v kanálu, jsou definována v definičním modulu (soubor maker) a zůstávají zachována i po resetu na konci programu, příp. po následujícím spuštění řídicího systému. Globální makra je potom možné vyvolat v libovolném hlavním programu nebo podprogramu a nechat jej tak zpracovat.

Poznámka

Aby bylo možné používat makra z **externího** souboru maker v NC-programu, musí být soubor maker načten do NC systému.

Makra musí být před svým použitím definována. Přitom je potřeba dbát na následující pravidla:

- V makru mohou být definovány libovolné identifikátory, G-, M-, H-funkce a L-názvy podprogramů.
- Definice maker se může nacházet na začátku programu nebo ve vlastním definičním souboru (soubor maker).
- Lokální a programově globální makra jsou definována na začátku programu.
- Globální makra musí být definována v souboru maker, např. _N_DEF_DIR/_N_UMAC_DEF.
- Příkazy G-funkcí maker mohou být definovány jen jako globální makra.
- H- a L-funkce mohou být naprogramovány jako 2-místné.
- Příkazy M- a G-funkcí mohou být naprogramovány jako 3-místné.

Poznámka

Definice klíčových slov a rezervované názvy nesmí být pomocí maker přepisovány.

Syntaxe

Definice makra:

```
DEFINE <název makra> AS <operace_1> <operace_2> ...
```

Volání v NC programu:

```
<název makra>
```

Význam

DEFINE ... AS:	Kombinace klíčových slov pro definici makra
<název makra>:	Název makra Jako názvy maker jsou přípustné pouze identifikátory. Pomocí názvu makra se potom v NC programu makro vyvolává.
<Operace_1>:	První programový příkaz v makru
<Operace_2>:	Druhý programový příkaz v makru

Okrajové podmínky

Vnoření

Vnoření maker není možné.

Příklady

Příklad 1: Definice makra na začátku programu

Programový kód	Komentář
DEFINE LINIE AS G1 G94 F300	; Definice makra

Programový kód	Komentář
...	
N70 LINIE X10 Y20	; Volání makra
...	

Příklad 2: Definice maker v souboru makra

Programový kód	Komentář
DEFINE M6 AS L6	; Při výměně nástroje se bude vyvolávat podprogram, který převezme přenos nezbytných dat. V podprogramu se spouští M-funkce pro vlastní výměna nástroje (např. M106).
DEFINE G81 AS DRILL(81)	; kopírování příkazů G-funkcí podle normy DIN
DEFINE G33 AS M333 G333	; Při řezání závitu je požadována synchronizace s PLC. Předcházející příkaz G-funkce G33 byl pomocí strojního parametru přejmenován na G333, programování ale zůstává pro uživatele stejné.

Příklad 3: Externí soubor maker

Po načtení externího souboru maker do řídicího systému musí být soubor maker načten do NC systému. Teprve potom mohou být makra v NC-programech používána.

Programový kód	Komentář
%_N_UMAC_DEF	
;\$PATH=/_N_DEF_DIR	; Specifická uživatelská makra
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Vřeteno se otáčí vpravo, zapnutí chladicí kapaliny
DEFINE M14 AS M4 M7	; Vřeteno se otáčí vlevo, zapnutí chladicí kapaliny
DEFINE M15 AS M5 M9	; Zastavení vřetena, vypnutí chladicí kapaliny
DEFINE M6 AS L6	; Vyvolání programu pro výměnu nástroje
DEFINE G80 AS MCALL	; Deaktivování cyklu pro vrtání
M30	

Správa souborů a programů

3.1 Programová paměť

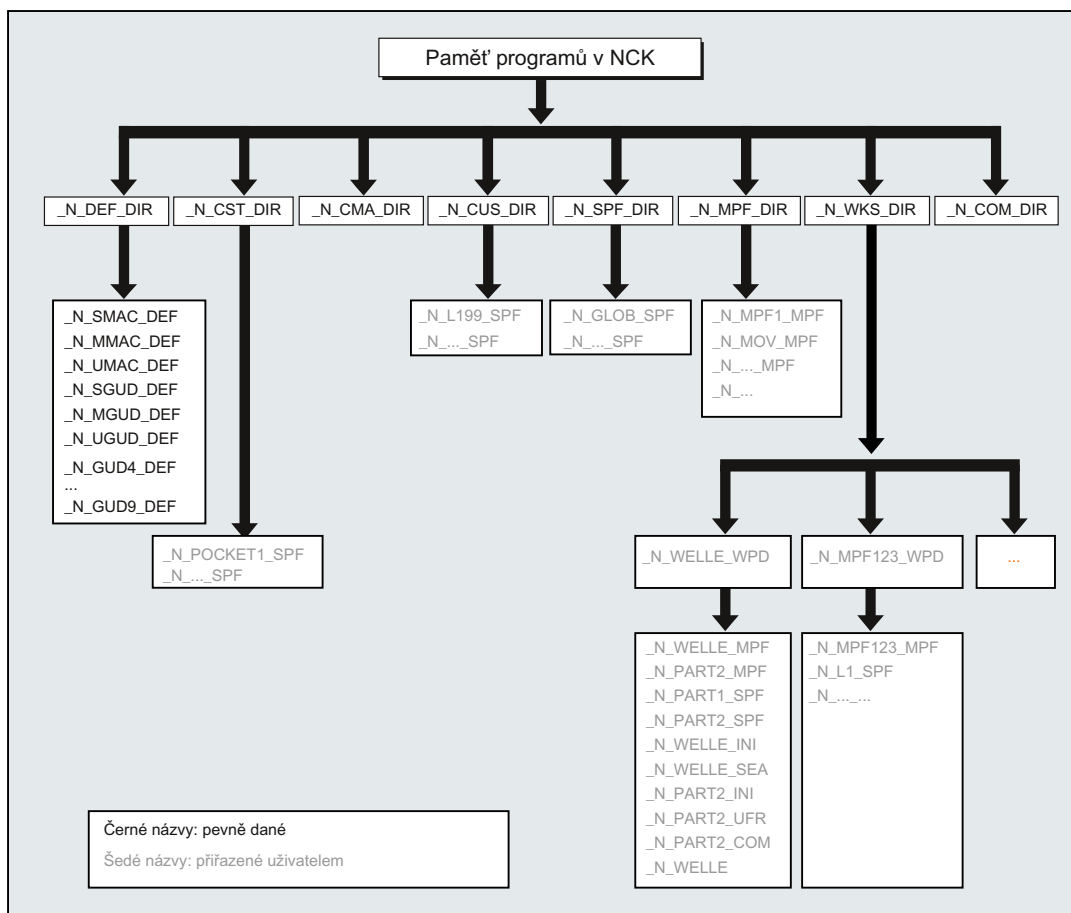
3.1.1 Programová paměť v NC systému

V paměti programů jsou trvale uloženy soubory a programy (např. hlavní programy a podprogramy, definice maker) (--> pasivní systém souborů).

Literatura:

Příručka k funkcím, Rozšiřovací funkce; Konfigurace paměti (S7)

Kromě toho existuje určitý počet typů souborů, které zde mohou být dočasně uloženy a v případě potřeby (např. při opracovávání určitých obrobků) jsou přeneseny do pracovní paměti (např. pro účely inicializace).



Standardní adresáře

Standardně jsou k dispozici následující adresáře:

Adresář	Obsah
_N_DEF_DIR	Datové moduly a moduly maker
_N_CST_DIR	Standardní cykly
_N_CMA_DIR	Cykly výrobce
_N_CUS_DIR	Uživatelské cykly
_N_WKS_DIR	Obrobky
_N_SPF_DIR	Globální podprogramy
_N_MPF_DIR	Hlavní programy
_N_COM_DIR	Komentáře

Typy souborů

Do paměti programů se mohou přenášet následující typy souborů:

Typ souboru	Popis
<název>_MPF	Hlavní program
<název>_SPF	Podprogram
<název>_TEA	Strojní parametry
<název>_SEA	Nastavované parametry
<název>_TOA	Korekční parametry nástroje
<název>_UFR	Posunutí počátku / frame
<název>_INI	Inicializační soubory
<název>_GUD	Globální uživatelská data
<název>_RPA	R-Parametry
<název>_COM	Komentář
<název>_DEF	Definice pro globální uživatelská data a makra

Hlavní adresář obrobků (_N_WKS_DIR)

Hlavní adresář obrobků je standardně vytvořen v paměti programů pod označením _N_WKS_DIR. Hlavní adresář obrobků obsahuje všechny obrobky, které máte naprogramovány, a odpovídající adresáře obrobků.

Adresáře obrobků (..._WPD)

Adresář obrobku obsahuje soubory, které jsou nezbytné pro opracovávání obrobku. Může se jednat o hlavní programy, podprogramy, libovolné inicializační programy a soubory komentářů.

Inicializační programy se po zvolení příslušného programu při prvním spuštění výrobního programu jednou zpracují (v závislosti na strojním parametru MD11280 \$MN_WPD_INI_MODE).

Příklad:

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

Soubor	Popis
_N_WELLE_MPF	Hlavní program
_N_PART2_MPF	Hlavní program
_N_PART1_SPF	Podprogram
_N_PART2_SPF	Podprogram
_N_WELLE_INI	Všeobecný inicializační program pro data daného obrobku
_N_WELLE_SEA	Inicializační program nastavovaných parametrů
_N_PART2_INI	Všeobecný inicializační program pro data programu PART2
_N_PART2_UFR	Inicializační program pro data framu pro program PART2
_N_WELLE_COM	Soubor komentářů

Kromě toho mohou být v adresáři obrobku ukládána také data, která nejsou bezprostředně zapotřebí pro zpracování programu prostřednictvím NC systému. Vedle souborů ASCII to mohou být také binární soubory, jako např. obrázky ve formátu JPG nebo popisy ve formátu PDF. Protože NC systém by mohl tyto objekty interpretovat jako binární soubory, musí být přípony souborů v NC systému známy (nastavení v rámci uvádění do provozu pomocí parametru MD17000 \$MN_EXTENSIONS_OF_BIN_FILES; v základním nastavení jsou předem definovány následující přípony souborů: JPG, GIF, PNG, BMP, PDF, ICO, HTM).

Volba obrobku pro zpracování

Adresář obrobku může být vybrán pro zpracování v určitém kanálu, Pokud se v tomto adresáři nachází hlavní program **stejného názvu** nebo jen jediný hlavní program (_NPF), bude tento program automaticky vybrán pro zpracování

Literatura:

Příručka pro obsluhu

3.1.2 Externí programová paměť

Kromě pasivního systému souborů v NC mohou být na stroji k dispozici také externí programové paměti (např. na lokálním pevném disku nebo na síťové jednotce).

Prostřednictvím funkce "Zpracování z externího zdroje" nebo "EES (Execution from External Storage)" mohou být výrobní programy zpracovávány **přímo** z externí paměti programů.

Literatura:

Příručka k funkcím – Základní funkce; K1: BAG, Kanál, programový režim, chování při resetu

Globální paměť výrobních programů (GDIR)

Při konfiguraci diskových jednotek může být jedna jednotka vybrána jako globální paměť výrobních programů (GDIR).

Literatura:

Příručka pro obsluhu; kapitola: "Správa programů" > „Instalace jednotek“

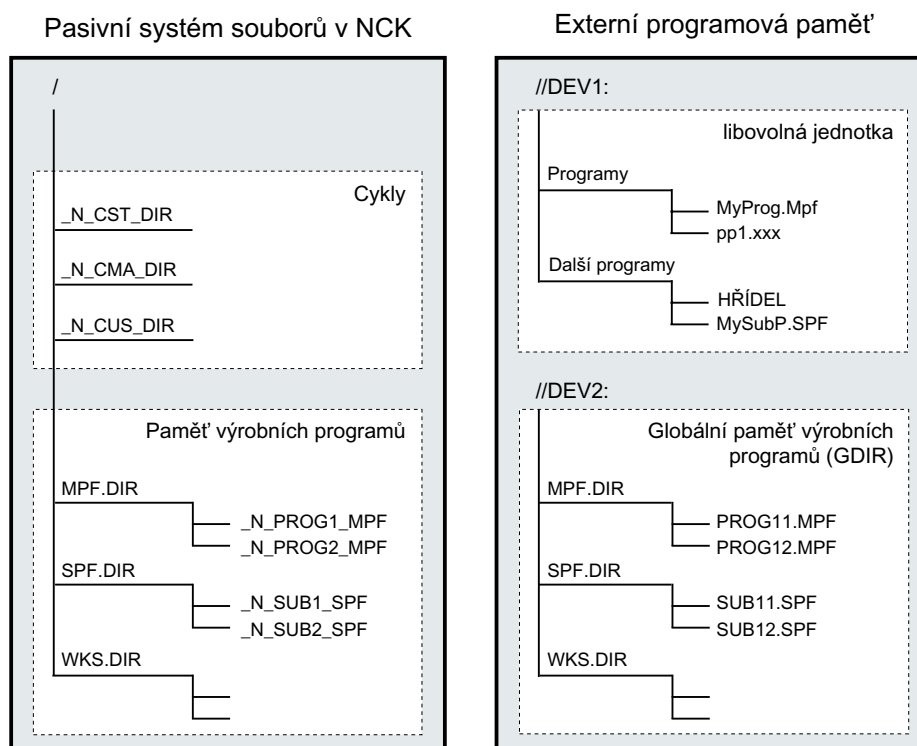
Systém potom na diskové jednotce automaticky založí adresáře MPF.DIR, SPF.DIR a WKS.DIR. Tyto tři adresáře tvoří paměť GDIR.

GDIR hraje roli výlučně pro fungování EES. V závislosti na konfiguraci jednotek GDIR nahrazuje nebo rozšiřuje GDIR paměti výrobních programů NC systému. Zřízení paměti GDIR není ale pro fungování EES bezpodmínečně nutné.

Adresáře a soubory v paměti GDIR mohou být ve výrobním programu adresovány stejným způsobem, jako v pasivním systému souborů. Díky tomu je možné kompatibilní přenášení NC programů s údajem cesty z pasivního systému souborů do GDIR. Adresář SPF.DIR v paměti GDIR je obsažen v cestě pro vyhledávání podprogramů.

Organizace programů

Organizaci programů v externí paměti programů má ozřejmit následující obrázek:



Systém souborů nerozlišuje velká a malá písmena

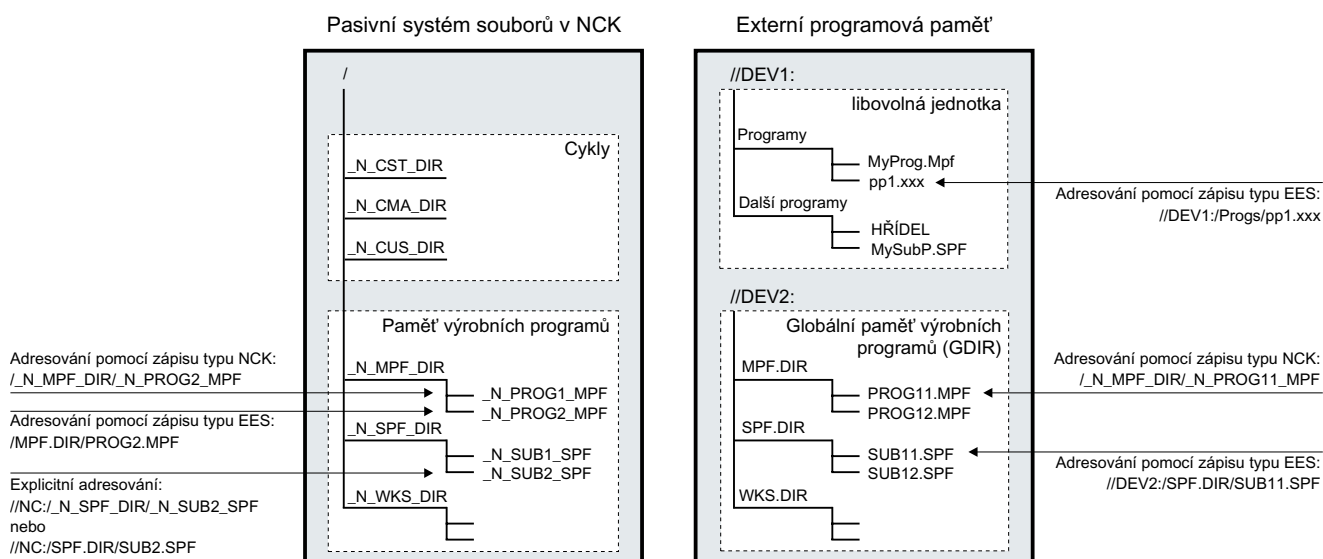
Poznámka

Aby se při adresování souborů zabránilo problémům v důsledku zápisu velkých a malých písmen (viz "Adresování souborů v paměti pro programy (Strana 219)"), měl by se jako externí programová paměť používat systém souborů, který **nerozlišuje velká a malá písmena**.

3.1.3 Adresování souborů v paměti pro programy

Soubor v paměti pro programy, na který je směrován příkaz pro manipulaci se soubory (např. WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO), je popsán buď prostřednictvím absolutní cesty plus názvu souboru nebo jen prostřednictvím samotného názvu souboru. Ve druhém případě se jako cesta k souboru použije cesta vybraného programu.

Adresování pomocí zápisu typu NC/EES



Adresování souborů v pasivním systému souborů

Adresování souborů v pasivním systému souborů se uskutečňuje obvyklým způsobem v **zápisu ve formátu NC systému** (názvy adresářů a názvy souborů začínají identifikací domény "`_N_`", oddělovací znak pro příponu adresáře/souboru je "`_`") bez udání názvu jednotky. Adresování pomocí **způsobu zápisu EES** (bez identifikace domény "`_N_`", oddělovací znak pro příponu názvu adresáře/souboru je "`.`") je ale přípustné také.

Příklad:

- Způsob zápisu typu NC: `"/_N_SPF_DIR/_N_SUB1_SPF"`
- Způsob zápisu typu EES: `"/SPF.DIR/SUB1.SPF"`

Poznámka

Adresování souborů v pasivním systému souborů při způsobu zápisu typu EES se interně převádí do způsobu zápisu typu NC podle následujících pravidel:

- Názvy adresářů a názvy souborů jsou rozšířeny o identifikaci domény "`_N_`".
- Jestliže je čtvrtý znak od konce v názvu adresáře nebo v názvu souboru tečka ("`.`"), je převedena na znak podtržení ("`_`").

Prostřednictvím předdefinovaných názvů jednotek "//NC:" je možno pasivní systém souborů adresovat také cíleně.

Příklad:

- Způsob zápisu typu NC: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- Způsob zápisu typu EES: "//NC:/SPF.DIR/SUB1.SPF"

Adresování souborů v externí paměti pro programy

Adresování souborů v externí paměti pro programy, která není konfigurována jako paměť GDIR, se musí uskutečňovat pomocí způsobu typu EES. Na začátku adresovací cesty musí být uveden název jednotky (např. "//DEV1:"). Přípustné jsou všechny symbolické názvy jednotek, které jsou konfigurovány v souboru /user/sinumerik/hmi/cfg/logdrive.ini.

Příklad:

- Způsob zápisu typu EES: "//DEV1:/MyProgDir/pp1.xxx"
- Způsob zápisu typu NC: nepřipustný

Adresování souborů v globální paměti pro výrobní programy (GDIR)

Při adresování souborů v paměti GDIR je kromě udání cesty pomocí způsobu zápisu typu EES přípustné také zadání cesty pomocí způsobu zápisu typu NC.

Příklad:

- Způsob zápisu typu EES: "//DEV2:/MPF.DIR/PROG11.MPF"
- Způsob zápisu typu NC: "/_N_MPF_DIR/_N_PROG11_MPF"

Poznámka

Adresování souborů v paměti GDIR pomocí způsobu zápisu typu NC se interně převádí do způsobu zápisu typu EES podle následujících pravidel:

- Identifikace domény "_N_" v názvech adresářů a názvech souborů se odstraní.
 - Jestliže je čtvrtý znak od konce v názvu adresáře nebo v názvu souboru znak podtržení („_“), je převeden na tečku („.“).
-

Pravidla pro zadávání cesty

Úplné udání cesty se skládá z názvu jednotky, cesty popisující adresář a z názvu souboru.

Název jednotky

Pro zadávání názvu jednotky platí následující pravidla:

- Přípustné jsou všechny symbolické názvy jednotek, které jsou konfigurovány v souboru /user/sinumerik/hmi/cfg/logdrive.ini.
- N začátku se nachází řetězec znaků "//", za kterým následuje minimálně jedno písmeno nebo číslice.
- Následující znaky mohou být libovolnou kombinací písmen, číslic, znaků podtržení ("_") a mezer.

- Název musí být ukončen písmenem nebo číslicí, za nimiž následuje „:“.
- Další speciální znaky jsou nepřipustné.

Poznámka

Pro pasivní systém souborů je předem definován název jednotky "//NC:“.

Příklady:

- Externí programová paměť:
 - //Drive1:
 - //Drive_1:
 - //Drive 1:
 - //A B:
 - //1 B C 2:

Cesta do adresáře

Pro zadávání cesty do adresáře platí následující pravidla:

- Na začátku a na konci cesty adresáře a jako oddělovací znak pro jednotlivé složky cesty se používá znak "/".

Poznámka

Dvojitý znak podtržení ("/") v rámci cesty adresáře je **nepřípustný!**

- Názvy adresářů:
 - Název adresáře musí začínat písmenem nebo číslicí. Jen v případě adresování pomocí zápisu typu NC začínají názvy adresářů identifikací domény "_N_".
 - Následující znaky mohou být libovolnou kombinací písmen, číslic a znaků podtržení ("_").

Poznámka

V případě externích programových pamětí jsou v názvech adresářů přípustné také mezery. To však neplatí, pokud je externí programová paměť konfigurována jako globální paměť výrobních programů (GDIR).

- Další speciální znaky jsou nepřipustné.
- Přípony názvů adresářů:
 - Přípony názvů adresářů se musí skládat přesně ze třech písmen/číslic.
 - Jsou oddělovány buď znakem "_" (zápis typu NC) nebo "." (zápis typu EES) od názvu adresáře.

Poznámka

V pasivním systému souborů existují pouze přípony názvů adresářů _DIR a _WPD.

Příklady:

- Pasivní systém souborů, příp. GDIR:
 - Způsob zápisu typu NC: `_N_WKS_DIR/_N_MYNCPROGS_WPD/...`
 - Způsob zápisu typu EES: `WKS.DIR/MYPROGS.WPD/...`
- Externí programová paměť:
 - `/abc`
 - `/ab_c.def`
 - `/ab c1.def`
 - `/a b c .d11`
 - `/abc.def/ghi.klm`

Název souboru

Pro názvy souborů platí následující pravidla:

- Jen v případě adresování pomocí zápisu typu NC začínají názvy souborů identifikací domény `"_N_"`.
- Následující dva znaky by měly být dvě písmena nebo znak podtržení, za kterým následuje písmeno.

Poznámka

Jestliže jsou tyto podmínky splněny, potom může být tento NC program samotný vyvoláván jako podprogram z nějakého jiného programu zadáním tohoto názvu programu. Jestliže název programu proti tomu začíná číslicemi, potom je volání podprogramu možné jedině pomocí příkazu CALL.

- Následující znaky mohou být libovolnou kombinací písmen, číslic a znaků podtržení (`"_"`).
- Přípona názvu souboru:
 - Přípona názvu souboru se musí skládat přesně ze třech písmen/číslic.

Poznámka

Pokud budete potřebovat informace o příponách souborů v pasivním systému souborů, viz "Programová paměť v NC systému (Strana 215)".

- Jsou oddělovány buď znakem `"_"` (zápis typu NC) nebo `"."` (zápis typu EES) od názvu souboru.

Příklady:

- Pasivní systém souborů, příp. GDIR:
 - Způsob zápisu typu NC: _N_SUB1_SPF
 - Způsob zápisu typu EES: SUB1.SPF
- Externí programová paměť:
 - Teil1
 - _Teil1
 - Teil_1.spf
 - Teil1.mpf

Podprogramy podle normy DIN

Pro podprogramy podle normy DIN platí následující pravidla:

- Prvním znakem je písmeno „L“.
- Následující znaky jsou číslice (minimálně jedna).
- Přípona názvu souboru:
 - Přípona názvu souboru se musí skládat přesně ze třech písmen.
 - Jsou oddělovány buď znakem "_" (zápis typu NC) nebo "." (zápis typu EES) od názvu souboru.

Příklady:

- L123
- L1_SPF (způsob zápisu typu NC) příp. L1.SPF (způsob zápisu typu EES)

Maximální délka cesty

Pro zadání názvu jednotky a cesty do adresáře je k dispozici maximálně 128 bytů, délka řetězce udávajícího název souboru smí činit maximálně 31 bytů. Z toho vyplývá, že celá cesta může mít maximální délku 159 bytů.

3.1.4 Cesta pro vyhledávání při volání podprogramu

Při volání podprogramu bez udání cesty se absolutní cesta zjišťuje prostřednictvím zpracování pevně dané cesty pro vyhledávání.

Paměť pro programy je přitom prohledávána v následující posloupnosti:

	Adresář	Popis
1	aktuální adresář / <i>název</i>	Aktuálním adresářem je adresář, ve kterém je vybírán program.
2	aktuální adresář / <i>název_SPF</i>	Může se jednat o následující: • adresář obrobku nebo standardní adresář <code>_N_MPF_DIR</code> v paměti výrobních programů NC systému, příp. globální paměť pro programy. nebo • libovolný adresář v externí paměti pro programy
3	aktuální adresář / <i>název_MPF</i>	
4	a //NC:/_N_SPF_DIR / <i>name_SPF</i>	Adresář podprogramu v paměti výrobních programů v NC systému
	b //DEV2:/_N_SPF_DIR / <i>název_SPF</i> ¹⁾	Adresář podprogramu v globální paměti výrobních programů Upozornění: Tento krok vyhledávání odpadá, pokud součástí konfigurace není žádná globální paměť výrobních programů nebo pokud vyhledávání probíhá v paměti pro programy v NC systému.
5	Rozšíření cesty pro vyhledávání naprogramované pomocí příkazu <code>CALLPATH</code> (viz "Rozšíření cesty pro vyhledávání při volání podprogramu (<code>CALLPATH</code>) (Strana 203)"). Poznámka: Jestliže není naprogramován žádný příkaz <code>CALLPATH</code> , tento krok vyhledávání odpadá.	
6	/_N_CUS_DIR / <i>název_SPF</i>	Adresář uživatelských cyklů
7	/_N_CMA_DIR / <i>název_SPF</i>	Adresář cyklů výrobce
8	/_N_CST_DIR / <i>název_SPF</i>	Adresář standardních cyklů

¹⁾ //DEV2:" představuje například jednotku, na které je konfigurována globální paměť pro výrobní programy.

Pro vyhledávání platí následující pravidla:

- Cesta pro vyhledávání se prochází pro každé jednotlivé volání podprogramu, tzn. nezáleží na tom, kde se nachází nadřazený program.
- V závislosti na adresáři je potřeba mít na zřeteli různé typy souborů.
- V zásadě se v adresáři nehledá v podřízených podadresářích.

3.1.5 Zjištění cesty a názvu souboru

Pokud potřebujete zjistit cestu a název souboru NC programu, máte ve výrobním programu k dispozici následující systémové proměnné, které můžete načíst:

Systémová proměnná	Typ	Význam
\$P_STACK	INT	Obsahuje programovou úroveň, na které je momentální NC program zpracováván.
\$P_PATH[<n>]	STRING	Obsahuje cestu NC programu, který je zpracováván na programové úrovni zvolené indexem pole <n>. Příklady: \$P_PATH[0] poskytuje cestu pro hlavní program, např. "/_N_WKS_DIR/_N_WELLE_WPD/". \$P_PATH[\$P_STACK - 1] obsahuje cestu volaného programu. Jestliže se cesta vztahuje na NC program, který je uložen v pasivním systému souborů NC nebo v globální paměti pro výrobní programy (GDIR), je pro cestu použit způsob zápisu typu NC systému. Jestliže se cesta vztahuje na NC program, který je zpracováván z nějaké jiné externí paměti pro programy, než je globální paměť pro výrobní programy, poskytuje proměnná \$P_PATH cestu ve formátu EES.
\$P_PROG[<n>]	STRING	Obsahuje název NC programu, který je zpracováván na programové úrovni zvolené indexem pole <n>. Jestliže je NC program uložen v pasivním systému souborů NC nebo v globální paměti pro výrobní programy, je pro název programu použit způsob zápisu typu NC. Jestliže je NC program zpracováván z nějaké jiné externí jednotky, než je globální paměť výrobních programů, poskytuje proměnná \$P_PROG název ve způsobu zápisu typu EES.
\$P_PROGPATH	STRING	Obsahuje cestu NC programu, který je právě zpracováván. Volání proměnné \$P_PROGPATH je stejné jako proměnné \$P_PATH[\$P_STACK].

3.1 Programová paměť

Systémová proměnná	Typ	Význam
\$P_IS_EES_PATH[<n>]	BOOL	Umožňuje zjistit, zda cesta poskytovaná proměnnou \$P_PATH[<n>], příp. název programu v proměnné \$P_PROG[<n>] odpovídá způsobu zápisu NC systému nebo EES.
		= FALSE \$P_PATH[<n>] a \$P_PROG[<n>] poskytují zápis typu NC. Tzn. každý identifikátor obsahuje počáteční řetězec „_N_“. Oddělovací znak pro identifikaci souborů je „_“. Příklady: • Cesta s typem zápisu NC: "/_N_WKS_DIR/_N_MYWPD_WPD/" • Název programu s typem zápisu NC: "_N_MYPROG_MPF" Cesta ve způsobu zápisu typu NC se může vztahovat jak na pasivní systém souborů v NC, tak i na globální paměť pro výrobní programy.
		= TRUE \$P_PATH[<n>] a \$P_PROG[<n>] poskytují zápis typu EES. Tzn. identifikátory neobsahují počáteční řetězec „_N_“. Oddělovací znak pro identifikaci souborů je „.“. Příklady: • Cesta s typem zápisu EES: "//DEV1:/WKS.DIR/MYWPD.WPD/" • Název programu s typem zápisu EES: "MYPROG.MPF"

<n>: Index <n> definuje programovou úroveň, ze které se mají informace o cestě načítat (rozsah hodnot: 0 ... 17)

Poznámka

V režimu EES jsou kromě globální paměti pro výrobní programy (GDIR) systémovými proměnnými \$P_PROG, \$P_PATH, a \$P_PROGPATH poskytovány názvy cest ve způsobu zápisu typu EES. Uživatelské programy, které tyto názvy cest vyhodnocují a dále zpracovávají, musí proto být pro režim EES rozšířeny tak, aby se i názvy cest daly zpracovávat ve způsobu zápisu typu EES.

3.2 Pracovní paměť (CHANDATA, COMPLETE, INITIAL)

Funkce

Pracovní paměť obsahuje aktuální systémová a uživatelská data, se kterými systém pracuje (aktivní systém souborů), např.:

- Aktivní strojní parametry
- Korekční parametry nástroje
- Posunutí počátku
- ...

Inicializační programy

V této souvislosti se jedná o programy, které zajišťují dosazování předdefinovaných hodnot pro data v pracovní paměti (inicializace). Za tím účelem se používají následující typy dat:

Typ souboru	Popis
název_TEA	Strojní parametry
název_SEA	Nastavované parametry
název_TOA	Korekční parametry nástroje
název_UFR	Posunutí počátku / frame
název_INI	Inicializační soubory
název_GUD	Globální uživatelská data
název_RPA	R-Parametry

Datové oblasti

Data mohou být rozčleněna do různých oblastí, v nichž mají platit. Řídící systém může například disponovat větším počtem kanálů nebo může být k dispozici více os, což je obvyklé.

Existují:

Označení	Datové oblasti
NC	Specifická data NC systému
CH<n>	Data specifického kanálu (<n> udává číslo kanálu)
AX<n>	Data specifické osy (<n> udává číslo osy stroje)
TO	Parametry nástroje
COMPLETE	Všechna data

Založení inicializačního programu na externím PC

Prostřednictvím identifikace datové oblasti a identifikace datového typu mohou být definovány oblasti, které jsou při ukládání dat považovány za jednu jednotku.

_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 řídicího systému do provozu je v pracovní paměti k dispozici datový blok, který zaručuje, že řídicí systém pracuje správně.

Postup v případě řídicích systémů s více kanály (CHANDATA)

Příkaz CHANDATA (<číslo kanálu>) pro větší počet kanálů je přípustný pouze v souboru _N_INITIAL_INI. Jedná se o inicializační soubor, pomocí kterého se uskutečňuje inicializace všech dat v řídicím systému.

Programový kód	Komentář
%_N_INITIAL_INI	
CHANDATA (1)	
	; Přiřazení os stroje kanálu 1:
\$MC_AXCONF_MACHAX_USED[0]=1	
\$MC_AXCONF_MACHAX_USED[1]=2	
\$MC_AXCONF_MACHAX_USED[2]=3	
CHANDATA (2)	
	; Přiřazení os stroje kanálu 2:
\$MC_AXCONF_MACHAX_USED[0]=4	
\$MC_AXCONF_MACHAX_USED[1]=5	
CHANDATA (1)	
	; Osové strojní parametry:
	; 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_FINE[AX1]=0.01	; Osa 1
\$MA_STOP_LIMIT_FINE[AX1]=0.01	; Osa 2

UPOZORNĚNÍ

Příkaz CHANDATA

Ve výrobním programu smí být příkaz CHANDATA použit pouze v kanálu, v němž se daný NC program zpracovává, tzn. tento příkaz se může používat také pro ochranu NC programů, aby nemohly být zpracovávány v kanálu, pro který nejsou určeny.

Pokud se vyskytne chyba, zpracování programu se přeruší.

Poznámka

Soubory INI v seznamu úloh neobsahují žádné příkazy CHANDATA.

Ukládání inicializačních programů (COMPLETE, INITIAL)

Soubory z pracovní paměti se mohou ukládat také na externí PC, odkud je možné je opět načíst.

- Soubory se ukládají pomocí příkazu COMPLETE.
- Pomocí příkazu INITIAL se vytvoří soubor INI ((_N_INITIAL_INI) zahrnující všechny oblasti.

Načtení inicializačního programu**UPOZORNĚNÍ****Ztráta dat**

Pokud je načten soubor s názvem "INITIAL_INI", potom jsou všechna data, do nichž tento soubor nedosazuje žádné hodnoty, inicializována standardními hodnotami. Výjimku z tohoto pravidla představují pouze strojní parametry. Standardní hodnoty (za normálních okolností "NULA" jsou dosazovány také do **nastavovaných parametrů, parametrů nástrojů, posunutí počátku, hodnot GUD,**

Pro načítání jednotlivých strojních parametrů se hodí např. soubor COMPLETE_TEA_INI. V tomto souboru řídící systém očekává strojní parametry. V tomto případě díky tomu zůstávají ostatní datové oblasti nedotčeny.

Načítání inicializačního programu

Pokud programy INI využívají pouze data určitého kanálu, mohou být zvoleny a vyvolány stejně jako výrobní programy. Takto je tedy možné inicializovat také data řízená programem.

Chráněné oblasti

4.1 Definice chráněných oblastí (CPROTDEF, NPROTDEF)

Chráněné oblasti, které mají chránit součásti stroje proti kolizím, jsou ve výrobním programu definovány v určitých blocích. Ty se skládají z následujících prvků:

1. Definice pracovní roviny
Před vlastní definicí chráněné oblasti musí být zvolena pracovní rovina, na kterou se má popis kontury chráněné oblasti vztahovat.
2. Začátek definice
V závislosti na příkazu NC jazyka bude založena chráněná oblast vztahující se ke kanálu nebo ke specifickému stroji.
3. Popis kontury chráněné oblasti
Kontura chráněné oblasti je popisována prostřednictvím pohybů pracovním posuvem. Tyto pohyby se neuskutečňují a nemají žádnou souvislost s předcházejícími nebo následujícími popisy geometrie. Definují pouze chráněnou oblast.
4. Konec definice

Syntaxe

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF (<n>, <t>, <AppLim>, <AppPlus>, <AppMinus>)
G0/G1/... X/Y/Z...
...
EXECUTE (<Var>)
```

Význam

DEF INT <Var>:	Definice lokálních pomocných proměnných datového typu INTEGER
<Var>:	Název pomocných proměnných
G17/G18/G19:	Pracovní rovina Upozornění: Pracovní rovina nesmí být před koncem definice změněna. Programování aplikáty mezi začátkem a koncem definice je nepřipustné.
CPROTDEF():	Předem definovaná procedura za účelem definice kanálové chráněné oblasti
NPROTDEF():	Předem definovaná procedura za účelem definice strojní chráněné oblasti

4.1 Definice chráněných oblastí (CPROTDEF, NPROTDEF)

<n>:	Číslo definované chráněné oblasti		
	Datový typ:	INT	
<t>:	Typ chráněné oblasti		
	Datový typ:	BOOL	
	Hodnota:	TRUE	Chráněná oblast vztahující se k nástroji :
		FALSE	Chráněná oblast vztahující se k obrobku :
<AppLim>:	Druh ohraničení ve 3. rozměru		
	Datový typ:	INT	
	Hodnota:	0	žádné ohraničení
		1	Ohraničení v kladném směru
		2	Ohraničení v záporném směru
		3	Ohraničení v kladném a záporném směru
<AppPlus>:	Hodnota ohraničení v kladném směru ve 3. rozměru		
	Datový typ:	REAL	
<AppMinus>:	Hodnota ohraničení v záporném směru ve 3. rozměru		
	Datový typ:	REAL	

4.1 Definice chráněných oblastí (CPROTDEF, NPROTDEF)

G0/G1/... X/Y/Z... :	Kontura chráněné oblasti je popsána maximálně 11 posuvovými pohyby ve zvolené pracovní rovině. První posuvový pohyb je přitom pohyb na konturu. Poslední bod popisu kontury se vždy musí shodovat s pevným bodem popisu kontury. Jako chráněná platí oblast vlevo od kontury: • Vnitřní chráněná oblast Konturu pro vnitřní chráněnou oblast je potřeba popisovat proti směru hodinových ručiček. • Vnější chráněná oblast (přípustná pouze pro chráněné oblasti vztahující se k obrobku!) Konturu pro vnější chráněnou oblast je potřeba popisovat ve směru hodinových ručiček. Přípustné jsou následující konturové prvky: • příkazy G0, G1 pro přímé konturové prvky • G2 pro kruhové oblouky ve směru hodinových ručiček Přípustné pouze pro chráněné oblasti vztahující se k obrobku! Nepřípustné u chráněných oblastí vztahujících se k nástroji, protože ty smí být jen konvexní. • G3 pro kruhové oblouky proti směru hodinových ručiček Upozornění: Chráněná oblast nemůže být popsána prostřednictvím celé kružnice. Celá kružnice musí být rozdělena na dva půlkruhy. Upozornění: Posloupnost G2 → G3 příp. G3 → G2 je nepřipustná. Mezi oba kruhové bloky musí být vložen krátký blok s příkazem G1.
EXECUTE (<Var>):	Předem definovaná procedura, která označuje konec definice Pomocí příkazu EXECUTE se vrátíte zpět k normálnímu zpracování programu.

Příklad

Viz příklad „Aktivování/deaktivování chráněné oblasti (CPROT, NPROT) (Strana 235)“.

Další informace

Chráněné oblasti vztahující se ke specifickému stroji

Chráněná oblast vztahující se ke konkrétnímu stroji, příp. její kontura, je definována prostřednictvím geometrických os, tzn. je vztažena k základnímu souřadnému systému (BCS) kanálu. Aby se mohlo uskutečňovat správné monitorování chráněných oblastí ve všech kanálech, ve kterých je příslušná strojní chráněná oblast aktivní, musí být základní souřadný systém (NCS) ve všech kanálech, jichž se to týká, identický:

- Poloha počátku souřadného systému vztažená na počátek souřadného systému stroje
- Orientace souřadných os

Vztažný bod popisu kontury

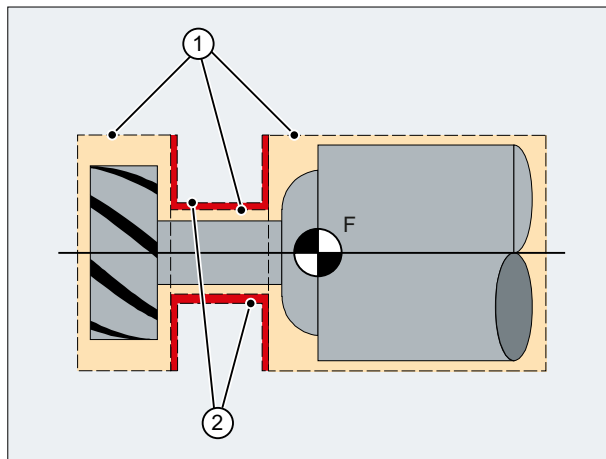
- Chráněné oblasti vztahující se k nástroji
Souřadnice pro chráněné oblasti vztahující se k **nástroji** je potřeba zadávat ve vztahu ke **vztažnému bodu držáku nástroje F**.
- Chráněné oblasti vztahující se k obrobku
Souřadnice pro chráněné oblasti vztahující se k **obrobku** je potřeba zadávat ve vztahu k počátku **základního souřadného systému (BCS)**.

Rotačně symetrické chráněné oblasti

U rotačně symetrických chráněných oblastí (např. vřetenové sklíčidlo) musí být popsána celá kontura, nejen kontura po osu otáčení.

Chráněné oblasti vztahující se k nástroji

Chráněné oblasti vztahující se k nástroji musí být vždy konvexní. Jestliže je požadována konkávní chráněná oblast, je potřeba ji rozložit na několik konvexních chráněných oblastí.



- ① Konvexní chráněné oblasti
- ② Konkávní chráněné oblasti (**nepřípustné!**)
- F Vztažný bod držáku nástroje

Okrajové podmínky

V průběhu definice chráněné oblasti nesmí být aktivní, příp. nesmí být použity, následující funkce:

- Korekce rádiusu nástroje (korekce rádiusu frézy, korekce rádiusu bříty)
- Transformace
- Najíždění na referenční bod (G74)
- Najíždění na pevný bod (G75)
- Doba prodlevy (G4)
- Zastavení předběžného zpracování (STOPRE)
- Konec programu (M17, M30)
- M-funkce: M0, M1, M2

4.2 Aktivování/deaktivování chráněné oblasti (CPROT, NPROT)

Chráněné oblasti, které byly předtím definovány ve výrobním programu, mohou být kdykoli aktivovány, příp. mohou být předběžně aktivovány pro pozdější aktivování prostřednictvím uživatelského programu PLC. Aktivní chráněné oblasti mohou být kdykoli opět deaktivovány.

Při aktivování, příp. při předběžném aktivování, existuje kromě toho možnost relativního posunutí vztažného bodu chráněné oblasti.

Poznámka

Chráněná oblast je zohledňována teprve po najetí na referenční bod všemi geometrickými osami kanálu, ve kterém byla aktivována.

Poznámka

Monitorování chráněných oblastí

Jestliže není aktivní žádná chráněná oblast vztahující se k nástroji, kontroluje se dráha nástroje vzhledem ke chráněné oblasti vztahující se k obrobku.

Jestliže není aktivní žádná chráněná oblast vztahující se k obrobku, žádné monitorování chráněných oblastí se neuskutečňuje.

Syntaxe

CPROT (<n>, <Status>, <XMov>, <YMov>, <ZMov>)

NPROT (<n>, <Status>, <XMov>, <YMov>, <ZMov>)

Význam

CPROT:	Předem definovaná procedura za účelem aktivování kanálové chráněné oblasti			
NPROT:	Předem definovaná procedura za účelem aktivování strojní chráněné oblasti			
<n>:	Číslo chráněné oblasti			
	Datový typ:	INT		
<Status>:	Pomocí tohoto parametru se definuje stav aktivování ve specifickém kanálu			
	Datový typ:	INT		
	Hodnota:	0	Deaktivovat chráněnou oblast	
		1	Chráněnou oblast předběžně aktivovat	
		2	Aktivovat chráněnou oblast	
		3	Předběžné aktivování chráněné oblasti s podmíněným zastavením	

<XMov>, <YMov>, <ZMov>:	Aditivní hodnoty posunutí ve směru X/Y/Z Posunutí může být provedeno v 1, 2 nebo ve 3 směrech. Hodnoty posunutí jsou vztaženy na:	
	• počátek souřadného systému stroje v případě chráněné oblasti vztahující se k obrobku • vztažný bod držáku nástroje F v případě chráněné oblasti vztahované k nástroji	
	Datový typ:	REAL

Příklad:

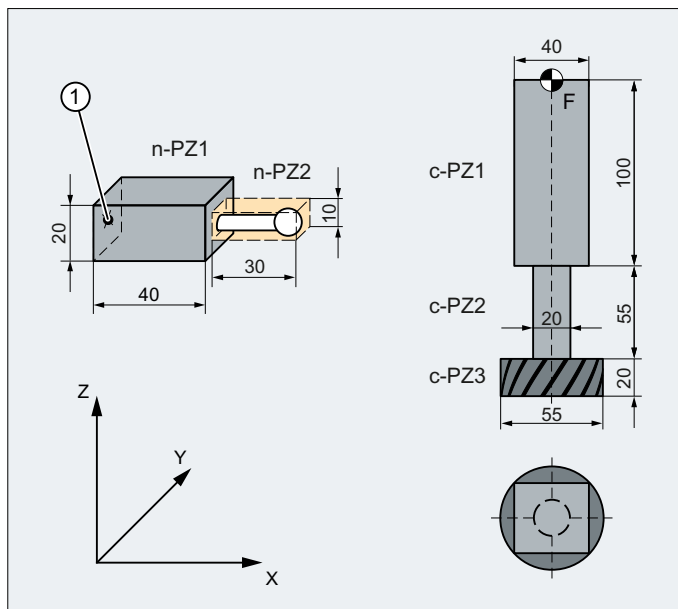
V případě frézky mají být monitorovány možné kolize frézy s měřicí sondou. Délka měřicí sondy má být při svém aktivování zadána prostřednictvím posunutí.

Pro tento účel jsou definovány následující chráněné oblasti:

- Specifická chráněná oblast daného stroje vztažená na obrobek pro držák měřicí sondy (n-PZ1) a pro měřicí sondu samotnou (n-PZ2).
- Chráněná oblast daného kanálu vztažená k nástroji pro držák frézy (c-PZ1), stopku frézy (c-PZ2) a frézu samotnou (c-PZ3).

Potvrzení všech chráněných oblastí je ve směru osy Z.

Souřadnice polohy vztažného bodu měřicí sondy při aktivování mají být X = -120, Y = 60 a Z = 80.



① Vztažný bod pro chráněnou oblast měřicí sondy

F Vztažný bod držáku nástroje

Programový kód

```
DEF INT PROTZONE
```

Komentář

```
; definice pomocných proměnných
```


4.2 Aktivování/deaktivování chráněné oblasti (CPROT, NPROT)

Programový kód	Komentář
G17	; pracovní rovina XY
; definice chráněné oblasti:	
NPROTDEF(1,FALSE,3,10,-10)	; chráněná oblast n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE (PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	; chráněná oblast n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE (PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	; chráněná oblast c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE (PROTZONE)	
CPROTDEF(2,TRUE,3,-100,-150)	; chráněná oblast c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE (PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; chráněná oblast c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE (PROTZONE)	
; aktivování chráněných oblastí:	
NPROT(1,2,-120,60,80)	; aktivování chráněné oblasti n-PZ1 s posunutím
NPROT(2,2,-120,60,80)	; aktivování chráněné oblasti n-PZ2 s posunutím
CPROT(1,2,0,0,0)	; aktivování chráněné oblasti c-PZ1
CPROT(2,2,0,0,0)	; aktivování chráněné oblasti c-PZ2
CPROT(3,2,0,0,0)	; aktivování chráněné oblasti c-PZ3

Další informace

Stav aktivování po rozběhu řídicího systému

Chráněná oblast může být aktivní už po náběhu řídicího systému a najetí osami na referenční bod. Jestliže se jedná o tento případ, jsou pro chráněnou oblast nastaveny následující systémové proměnné na hodnotu TRUE:

- \$SN_PA_ACTIV_IMMED[<n>] (pro chráněnou oblast vztahující se ke stroji), příp.
- \$SC_PA_ACTIV_IMMED[<n>] (pro chráněnou oblast vztahující se ke kanálu)
Index „<n>“ odpovídá číslu chráněné oblasti: 0 = 1. chráněná oblast

Chráněná oblast je aktivována se stavem = 2 a bez posunutí.

Vícenásobné aktivování jedné chráněné oblasti

Chráněná oblast vztažená ke stroji může být v platnosti i ve více kanálech současně (např. chráněná oblast pinoly v případě dvou proti sobě umístěných saní). Monitorování chráněných oblastí se uskutečňuje jen tehdy, pokud bylo všemi geometrickými osami najeto na referenční bod.

Chráněná oblast nemůže být v jednom kanálu aktivována současně s různými hodnotami posunutí.

Monitorování chráněné oblasti v případě aktivní korekce rádiusu nástroje

Pokud je korekce rádiusu nástroje aktivní, je funkční monitorování chráněných oblastí možné jen tehdy, pokud je rovina korekce rádiusu nástroje identická s rovinou, ve které je definována chráněná oblast.

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

Funkce

Funkce CALCPOSI kontroluje v souřadném systému obrobku (WCS), zda když se vychází z počáteční polohy, mohou **geometrické osy** urazit předem zadanou dráhu, aniž by došlo k narušení aktivních ohraničení. V případě, že dráhu uvažovaného posuvu není možné z důvodu ohraničení realizovat úplně celou, poskytne funkce na svém výstupu kladné decimálně kódované stavové slovo a maximální možnou dráhu posuvu.

Definice

```
INT CALCPOSI (VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

Syntaxe

```
<Status> = CALCPOSI (VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

Význam

CALCPOSI (...):	Předem definovaná funkce pro zkoušku narušení ohraničení se zřetelem na geometrické osy	
	Zastavení předběžného zpracování:	ne
	Samostatný blok:	ano

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOS)

<Status>: (část 1)	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.	
	Datový typ:	INT
	Rozsah hodnot:	$-8 \leq x \leq 100000$
	Hodnota:	0 Dráhu posuvu je možné realizovat úplně celou
		-1 V proměnné <Limit> je nejméně jedna složka záporná.
		-2 Chyba při výpočtu transformace. Příklad: Dráha posuvu prochází singularitou, takže polohy os nejsou definovány.
		-3 Uváděná dráha posuvu <Dist> a maximální možná dráha posuvu <MaxDist> jsou lineárně závislé. Upozornění Může se vyskytnout pouze ve spojení s nastavením parametru <TestLim>, bit 4 == 1.
		-4 Projekce směru pohybu obsažená v parametru <Dist> na ohraničující plochu představuje nulový vektor, příp. směr pohybu je kolmý na narušovanou ohraničující plochu. Upozornění Může se vyskytnout pouze ve spojení s nastavením parametru <TestLim>, bit 5 == 1.
		-5 V parametru <TestLim> jsou bit 4 == 1 A bit 5 == 1
		-6 S minimálně jednou strojní osou, která musí být pro přezkoumání ohraničení pohybů uvažována, nebylo najeto na referenční bod.
		-7 Funkce prevence kolize: Neplatná definice kinematického řetězce nebo chráněné oblasti.
		-8 Funkce prevence kolize: Funkce nemohla být zpracována kvůli chybě paměti.
<Status>: (část 2)	Místo jednotek	
	Upozornění Pokud je narušeno více ohraničení současně, je uvedeno to, které způsobuje nejsilnější omezení předem zadané dráhy posuvu.	
	Hodnota:	1 Dráha je omezena softwarovým koncovým spínačem
		2 Dráha je omezena ohraničením pracovního pole
		3 Dráha je omezena chráněnými oblastmi
		4 Funkce prevence kolize: Dráha je omezena chráněnými oblastmi
	Místo desítek	
	Hodnota:	1x Počáteční hodnota narušuje hranici
		2x Předem zadaná přímka narušuje hranici. Tato hodnota je výsledkem také tehdy, pokud samotný koncový bod žádnou hranici nenarušuje, ale na dráze z počátečního do koncového bodu k narušení některé z mezních hodnot došlo (např. průjezd chráněnou oblastí, zakřivené polohy softwarových koncových spínačů ve WCS kvůli nelineárním transformacím, např. Transmit).

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

<Status>: (část 3)	Místo stovek		
	Hodnota:	1xx	A místo jednotek == 1 nebo 2: Je narušena kladná mezní hodnota.
			A místo jednotek == 3 ¹⁾ : Je narušena chráněná oblast specifikovaná v NC systému.
		2xx	A místo jednotek == 1 nebo 2: Je narušena záporná mezní hodnota.
			A místo jednotek == 3 ¹⁾ : Je narušena chráněná oblast specifického kanálu.
<Status>: (část 4)	Místo tisíců		
	Hodnota:	1xxx	A místo jednotek == 1 nebo 2: Faktor, kterým se násobí číslo osy, která narušuje ohraničení. Počítání os začíná 1. Přiřazení: - Softwarový koncový spínač: Strojní osy - Ohraničení pracovního pole: Geometrické osy A místo jednotek == 3 ¹⁾ : Faktor, se kterým se násobí číslo narušované chráněné oblasti.
<Status>: (část 5)	Místo stovek tisíců		
	Hodnota:	0xxxxx	Místo stovek tisíců == 0: Parametr <Dist> zůstává nezměněn.
		1xxxxx	V parametru <Dist> funkce ukládá směrový vektor, který definuje další směr pohybu na ohraničující ploše. Mohou se vyskytnout následující okrajové podmínky: - Je narušen softwarový koncový spínač nebo ohraničení pracovního pole (nikoli v počátečním bodě) - Transformace není aktivní <TestID>, bit 4 nebo bit 5 == 1
<Start>:	Referenční vztah na vektor s počátečními pozicemi: <Start> [0]: 1. geometrická osa <Start> [1]: 2. geometrická osa <Start> [2]: 3. geometrická osa		
	Typ parametru:		Vstup
	Datový typ:		VAR REAL[3]
	Rozsah hodnot:		-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOS)

<Dist>:	Referenční vztah na vektor.		
	Vstup: inkrementální dráha posuvu • <Dist> [0]: 1. geometrická osa • <Dist> [1]: 2. geometrická osa • <Dist> [2]: 3. geometrická osa		
	Výstup (jen pokud je aktivováno místo stovek tisíců v proměnné <Status>):		
	Parametr <Dist> obsahuje jako výstupní hodnotu jednotkový vektor v , který definuje další směr posuvu ve WCS.		
	Případ 1: Stanovení vektoru v v případě, že <TestID>, bit 4 == 1 Vstupní vektory <Dist> a <MaxDist> definují rovinu pohybu. Tato rovina protíná narušovanou ohraničující plochu. Průsečnice obou rovin definuje směr vektoru v . Orientace (znaménko) je přitom zvolena tak, aby úhel mezi vstupním vektorem <MaxDist> a vektorem v nebyl větší než 90 stupňů.		
	Případ 2: Stanovení vektoru v v případě, že <TestID>, bit 5 == 1 Vektor v je jednotkový vektor ve směru projekce vektoru posuvu obsaženého v parametru <Dist> do ohraničující plochy. Jestliže je projekce vektoru posuvu do ohraničující plochy nulový vektor, výpočet skončí s chybou.		
Typ parametru:		Vstup/výstup	
Datový typ:		VAR REAL[3]	
Rozsah hodnot:		-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL	
<Limit>:	Odkaz na pole délky 5.		
	• <Limit> [0 - 2]: Minimální vzdálenosti geometrických os od ohraňování: – <Limit> [0]: 1. geometrická osa – <Limit> [1]: 2. geometrická osa – <Limit> [2]: 3. geometrická osa Minimální vzdálenosti jsou dodrženy v případech: – Ohraňování pracovního pole: Žádná omezení – Softwarový koncový spínač: Pokud buď není aktivní žádná transformace nebo je aktivní jedna transformace, u které je ale možné jednoznačné přiřazení geometrických os lineárním strojním osám, např. u 5-osé transformace. • <Limit> [3]: Obsahuje minimální vzdálenost pro lineární strojní osy, které nemohou být např. na základě nelineární transformace přiřazeny žádné geometrické ose. Tato hodnota se kromě toho používá jako mezní hodnota při kontrole tradičních chráněných oblastí a oblastí chráněných kvůli prevenci kolize. • <Limit> [4]: Obsahuje minimální vzdálenost pro kruhové osy stroje, které např. na základě nelineární transformace nemohou být přiřazeny žádné geometrické ose.		
	Upozornění Tato hodnota se uplatňuje pouze při monitorování softwarových koncových spínačů při speciálních transformacích.		
	Typ parametru:		Vstup
	Datový typ:		VAR REAL[5]
Rozsah hodnot:		-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL	

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

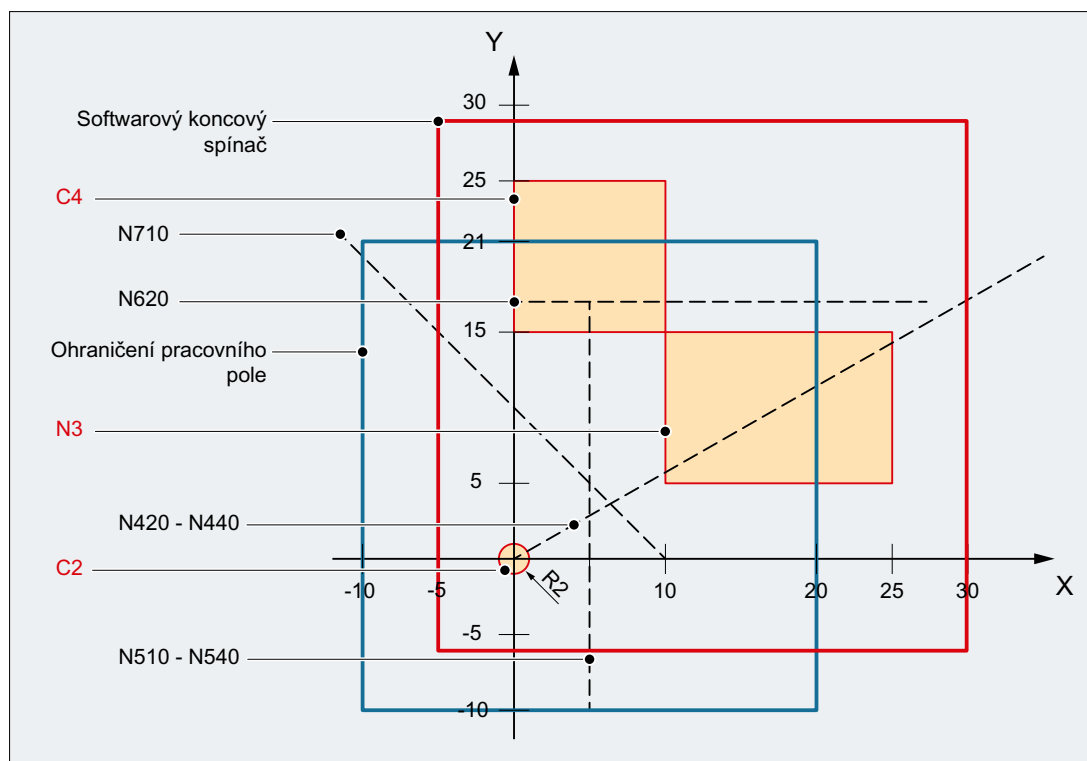
<MaxDist>:	Odkaz na vektor s inkrementální dráhou posuvu, u kterého je překročena předem zadaná minimální vzdálenost od ohraničení osy pro všechny strojní osy podílející se na pohybu:	
	• <Dist> [0]: 1. geometrická osa • <Dist> [1]: 2. geometrická osa • <Dist> [2]: 3. geometrická osa	
	Pokud dráha není nijak omezena, je obsah tohoto výsledného parametru roven obsahu parametru <Dist>.	
	V případě, že <TestID>, bit 4 == 1: <Dist> a <MaxDist>	
	<MaxDist> a <Dist> musí jako vstupní hodnoty obsahovat vektory, které popisují rovinu, ve které se uskutečňuje pohyb. Oba vektory musí být vůči sobě lineárně nezávislé. Absolutní hodnota <MaxDist> je libovolná. Pokud budete potřebovat informace o výpočtu směru pohybu, viz popis parametru <Dist>.	
<MeasSys>:	Typ parametru:	Výstup
	Datový typ:	VAR REAL[3]
	Rozsah hodnot:	-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL
	Systém měřicích jednotek (palce/metrický systém) pro údaje poloh a délek (ne-povinný)	
	Datový typ:	BOOL
Hodnota:	FALSE (předdefinované nastavení)	Systém měřicích jednotek v souladu s momentálně aktivním příkazem G-funkce ze skupiny 13 (G70, G71, G700, G710). Upozornění Když je aktivní příkaz G70 a základní systém je metrický nebo když je aktivní příkaz G71 a základní systém je imperiální (palce), jsou systémové proměnné \$AA_IW a \$AA_MW udávány v základním systému a v případě nutnosti musí být při použití funkce CALCPOSI přepočítány.
	TRUE	Systém měřicích jednotek v souladu s nastaveným základním systémem: MD52806 \$MN_ISO_SCALING_SYSTEM

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

<TestLim>:	Bitový kód pro volbu ohraničení, které má být kontrolováno (nepovinný)	
	Datový typ:	INT
	Předdefinovaná hodnota:	Bit 0, 1, 2, 3, 6, 7 == 1 (207)
	Bit	decimální
	0	1
	1	2
	2	4
	3	8
	4	16
	5	32
	6	64
	7	128
	8	256
Význam		
Softwarový koncový spínač		
Ohraničení pracovního pole		
Aktivované tradiční chráněné oblasti		
Předběžně aktivované tradiční chráněné oblasti		
V případě narušení softwarových koncových spínačů, příp. ohraničení pracovního pole, je na výstupu v parametru <Dist> směr pohybu v souladu s případem 1 (viz výše).		
V případě narušení softwarových koncových spínačů, příp. ohraničení pracovního pole, je na výstupu v parametru <Dist> směr pohybu v souladu s případem 2 (viz výše).		
Aktivované oblasti chráněné kvůli prevenci kolize		
Předběžně aktivované oblasti chráněné kvůli prevenci kolize		
Dvojice aktivovaných a předběžně aktivovaných oblastí chráněných kvůli prevenci kolize		
¹⁾ Jestliže je narušeno několik chráněných oblastí, je na výstupu chráněná oblast, která způsobuje nejsilnější omezení předem zadané dráhy.		

Příklad:

Ohraničení



Příklad ukazuje platné softwarové koncové spínače a ohraničení pracovního pole v rovině X-Y a následující tři chráněné oblasti:

- C2: Kanálová chráněná oblast vztahující se k nástroji, aktivní, kruhová, rádius = 2 mm
- C4: Kanálová chráněná oblast vztahující se k obrobku, předběžně aktivovaná, čtvercová, délka strany = 10 mm
- N3: Chráněná oblast specifického stroje, aktivní, pravoúhlá, délky stran = 10 mm x 15 mm

NC program

V NC programu jsou napřed definovány chráněné oblasti a ohraničení pracovního pole. Potom je vyvolána funkce `CALCPOSI()` s různými nastaveními parametrů.

Programový kód

```

N10 DEF REAL _START[3]
N20 DEF REAL _DIST[3]
N30 DEF REAL _LIMIT[5]
N40 DEF REAL _MAXDIST[3]
N50 DEF INT _PA
N60 DEF INT _STATUS

```

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

Programový kód

```
; chráněná oblast C2 vztahující se k nástroji
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)

; chráněná oblast C4 vztahující se k obrobku
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)

; chráněná oblast N3 vztahující se ke specifickému stroji
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)

; aktivování, příp. předběžné aktivování chráněných oblastí
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)

; definice ohraničení pracovního pole
N290 G25 XX=-10 YY=-10
N300 G26 XX=20 YY=21
N310 _START[0] = 0.
N320 _START[1] = 0.
N330 _START[2] = 0.
N340 _DIST[0] = 35.
N350 _DIST[1] = 20.
N360 _DIST[2] = 0.
N370 _LIMIT[0] = 0.
N380 _LIMIT[1] = 0.
N390 _LIMIT[2] = 0.
N400 _LIMIT[3] = 0.
N410 _LIMIT[4] = 0.
N420 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
N430 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,3)
N440 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,1)
N450 _START[0] = 5.
N460 _START[1] = 17.
N470 _START[2] = 0.
N480 _DIST[0] = 0.
N490 _DIST[1] = -27.
N500 _DIST[2] = 0.
N510 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,14)
N520 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N530 _LIMIT[1] = 2.
```

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

Programový kód

```

N540 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N550 _START[0] = 27.
N560 _START[1] = 17.1
N570 _START[2] = 0.
N580 _DIST[0] = -27.
N590 _DIST[1] = 0.
N600 _DIST[2] = 0.
N610 _LIMIT[3] = 2.
N620 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,12)
N630 _START[0] = 0.
N640 _START[1] = 0.
N650 _START[2] = 0.
N660 _DIST[0] = 0.
N670 _DIST[1] = 30.
N680 _DIST[2] = 0.
N690 TRANS X10
N700 AROT Z45
N710 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
; framy z řádků N690 a N700 opět vymazat
N720 TRANS
N730 _START[0] = 0.
N740 _START[1] = 10.
N750 _START[2] = 0.
; vektory _DIST a _MAXDIST definují rovinu pohybu
N760 _DIST[0] = 30.
N770 _DIST[1] = 30.
N780 _DIST[2] = 0.
N790 _MAXDIST[0] = 1.
N800 _MAXDIST[1] = 0.
N810 _MAXDIST[2] = 1.
N820 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30

```

Výsledky funkce CALCPOSI()

N...	<Status>	<MaxDist>[0] Δ X	<MaxDist>[1] Δ Y	Poznámky
420	3123	8.040	4.594	N3 je narušena.
430	1122	20.000	11.429	Žádné monitorování chráněných oblastí, došlo k narušení ohraničení pracovního pole.
440	1121	30.000	17.143	Je aktivní ještě monitorování softwarových koncových spínačů.
510	4213	0.000	0.000	Počáteční bod narušuje C4
520	0000	0.000	-27.000	Předběžně aktivovaná C4 není monitorována. Zadanou dráhu je možné objet úplně celou.
540	2222	0.000	-25.000	Kvůli nastavení _LIMIT[1] = 2 je dráha omezena kvůli ohraničení pracovního pole.

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

N...	<Status>	<MaxDist>[0] Δ X	<MaxDist>[1] Δ Y	Poznámky
620	4223	-13.000	0.000	Vzdálenost k C4 je kvůli C2 a parametru _LIMIT[3] celkem 4 mm. Vzdaľenost C2 → N3 o hodnotě 0,1 mm nezpůsobuje omezení dráhy pohybu.
710	1221	0.000	21.213	Je aktivní frame se složkou posunu a otočení. Přípustná dráha pohybu v parametru _DIST platí v posunutém a otočeném systému WCS.
820	102121	18.000	18.000	Dochází k narušení softwarového koncového spínače osy Y. Prostřednictvím nastavení <_TESTLIM> = 17 si vyžádáte výpočet směru dalšího pohybu. Tento směr je v proměnné <_DIST> (0.707, 0.0, 0.707). Hodnota je platná, protože v proměnné <_STATUS> je aktivováno místo stovek tisíců.

Další informace

Stav osy „najeto na referenční bod“

Všemi strojními osami, které mají být příkazem `CALCPOSI()` vyhodnocovány, musí být najeto na referenční bod.

Údaje dráhy vztažené na kruh

Všechny údaje dráhy zahrnující kruhové úseky jsou **vždy** interpretovány jako hodnoty rádiusu. Na to je potřeba myslet zejména u příčných os, pro něž je aktivováno programování průměrů (`DIAMON` / `DIAM90`).

Zkrácení dráhy posuvu

Jestliže je dráha posuvu uváděné osy omezena, jsou ve výstupní hodnotě <MaxDist> funkce odpovídajícím způsobem zkráceny dráhy posuvu také ostatních os. Výsledný koncový bod proto i nadále leží na předem zadané dráze.

Kruhové osy

Kruhové osy jsou monitorovány jen tehdy, pokud se nejedná o osy typu modulu.

Je přípustné, aby pro jednu nebo více od podílejících se na pohybu nebyly definovány žádné softwarové koncové spínače, příp. ohraničení pracovního pole nebo chráněné oblasti.

Stav softwarových koncových spínačů a ohraničení pracovního pole

Softwarové koncové spínače a ohraničení pracovního pole jsou zohledňovány jen tehdy, pokud jsou při zpracovávání funkce `CALCPOSI()` aktivní. Jejich stav může být ovlivňován např. prostřednictvím následujícího:

- Strojní parametry: `MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS`
- Nastavované parametry: `$AC_WORKAREA_CS_...`
- Signály rozhraní NC/PLC: `DB31, ... DBX12.2 / 3`
- Příkazy: `WALIMON` / `WALIMOF`

4.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

Softwarové koncové spínače a transformace

U funkce `CALCPOSI()` se může stát, že u různých kinematických transformací (např. `TRANSMIT`) nebude vždy možné kvůli víceznačnosti na určitých místech v oblasti pohybu jednoznačně stanovit polohy strojních os (MCS) z poloh geometrických os (WCS). Při normálních posuvech vyplývá jednoznačnost zpravidla z předcházejícího dění a z podmínky, že spojitý pohyb ve WCS musí odpovídat spojitému pohybu os stroje v MCS. Při monitorování softwarových koncových spínačů se proto v případech tohoto typu připojuje poloha stroje v okamžiku zpracování funkce `CALCPOSI()`, aby se víceznačnost odstranila.

Poznámka

Zastavení předběžného zpracování

Při použití funkce `CALCPOSI()` ve spojení s transformacemi je ale samotný uživatel odpovědný za to, že kvůli synchronizaci poloh os stroje s předběžným zpracováním bude před příkazem `CALCPOSI()` naprogramováno zastavení předběžného zpracování (`STOPRE`).

Vzdálenost chráněné oblasti a tradiční chráněné oblasti

U tradičních chráněných oblastí **není** zaručeno, že při posuvech po předem zadané dráze bude dodržena bezpečnostní vzdálenost vůči všem chráněným oblastem, která je zadána v parametru `<Limit>[3]`. Je zaručeno pouze to, že při prodloužení koncového bodu, který je funkcí poskytován v parametru `<Dist>`, o bezpečnostní vzdálenost ve směru posuvu nebude narušena žádná chráněná oblast. Příмка se ale může na své dráze dostat libovolně blízko k chráněné oblasti.

Vzdálenost od chráněné oblasti a oblasti chráněné kvůli prevenci kolize

U oblastí chráněných kvůli prevenci kolize je zaručeno, že při posuvech po předem zadané dráze bude dodržena bezpečnostní vzdálenost vůči všem chráněným oblastem, která je zadána v parametru `<Limit>[3]`.

Bezpečnostní vzdálenost zadaná v parametru `<Limit>[3]` je v platnosti jen tehdy, pokud platí:

`<Limit>[3] > (MD10619 $MN_COLLISION_TOLERANCE)`

Pokud je nastaven bit 4 v parametru `<TestLim>` (výpočet dalšího směru pohybu), je směrový vektor obsažený v parametru `<DIST>` platný jen tehdy, pokud je nastaveno místo stovek tisíců ve výstupní hodnotě funkce (`<status>`). Jestliže takový směr nemůže být stanoven, buď proto, že by došlo k narušení chráněné oblasti, nebo proto, že je aktivní transformace, zůstává vstupní hodnota v parametru `<DIST>` nezměněna. Žádné další chybové hlášení se neaktivuje.

Speciální příkazy dráhy

5.1 Najíždění na kódované pozice (CAC, CIC, CDC, CACP, CACN)

Pomocí následujících příkazů můžete lineárními a kruhovými osami pomocí čísel pozic najíždět na pevné polohy os uložené v tabulkách strojních parametrů. Tento druh programování je označován jako "Najíždění na kódované pozice".

Syntaxe

CAC (<n>)
CIC (<n>)
CACP (<n>)
CACN (<n>)

Význam

CAC (<n>):	Najíždění na kódovanou pozici s číslem polohy n.
CIC (<n>):	Najíždění na kódovanou pozici, která je uvedena o n míst pozic před (+n) nebo za (-n) od aktuálního čísla pozice.
CDC (<n>):	Najíždění na kódovanou pozici s číslem polohy n po nejkratší dráze (jen pro kruhové osy)
CACP (<n>):	Najíždění na kódovanou pozici s číslem polohy n v kladném směru dráze (jen pro kruhové osy)
CACN (<n>):	Najíždění na kódovanou pozici s číslem polohy n v záporném směru dráze (jen pro kruhové osy)
<n>:	Číslo pozice v tabulce strojních parametrů Rozsah hodnot: 0, 1, ... (max. počet míst v tabulce - 1)

Příklad: Najíždění polohovací osou na kódované pozice

Programový kód	Komentář
N10 FA[B]=300	; posuv pro polohovací osu B
N20 POS[B]=CAC(10)	; najíždění na kódovanou pozici s číslem polohy 10
N30 POS[B]=CIC(-4)	; najíždění na kódovanou pozici s číslem "aktuální číslo polohy" - 4

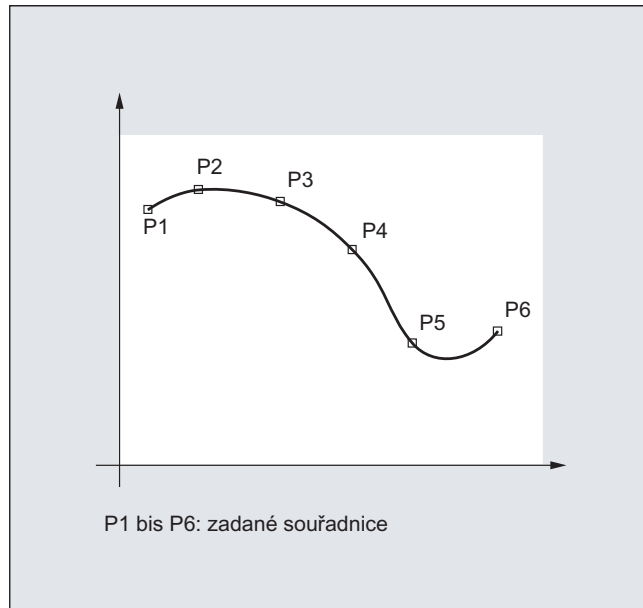
Literatura

- Příručka k funkcím, Rozšiřovací funkce; Osy s dělením (T1)
- Příručka k funkcím, Synchronní akce

5.2 Splinová interpolace (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Libovolně zakřivené kontury na obrobku nemusí být analyticky popsány úplně přesně. Kontury tohoto druhu jsou proto aproximovány pomocí omezeného počtu uzlových bodů, např. při digitalizaci povrchů. Za účelem vytvoření digitalizovaného povrchu na obrobku musí být uzlové body spojeny, takže vznikne popsaná kontura. To umožňuje splinová interpolace.

Spline definuje křivku, která se skládá z polynomů 2. nebo 3. stupně. Vlastnosti na uzlových bodech splinu mohou být definovány v závislosti na použitém typu splinu.



U systému SINUMERIK solution line jsou k dispozici následující typy splinů:

- A-Spline
- B-Spline
- C-Spline

Syntaxe

Všeobecně:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

U B-splinů může být ještě navíc naprogramováno:

PW=<n>

SD=2

PL=<hodnota>

U A- a C-splinů může být ještě navíc naprogramováno:

BAUTO / BNAT / BTAN

EAUTO / ENAT / ETAN

5.2 Splinová interpolace (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Význam

Typ splinové interpolace:				
ASPLINE:		Příkaz pro aktivování interpolace pomocí A-splinů		
BSPLINE:		Příkaz pro aktivování interpolace pomocí B-splinů		
CSPLINE:		Příkaz pro aktivování interpolace pomocí C-splinů		
		Příkazy ASPLINE, BSPLINE a CSPLINE mají modální platnost a patří do skupiny příkazů dráhy.		
Uzlové, příp. řídící body:				
X... Y... Z... A... B... C...		Polohy v kartézských souřadnicích.		
Váha bodu (jen B-spliny):				
PW:		Pomocí příkazu PW je možno pro každý uzlový bod naprogramovat tak zva- nou "váhu bodu".		
<n>:		"Váha bodu"		
		Rozsah hodnot:	$0 \leq n \leq 3$	
		Velikost kroku:	0.0001	
		Působnost:	$n > 1$	Křivka je silněji přitahována k uzlovému bodu.
			$n < 1$	Křivka je přitahována k uzlovému bodu méně silně.
Stupeň splinu (jen B-spliny):				
SD:		Standardně se používají se polynomy 3. stupně. Naprogramováním příka- zu SD=2 se mohou ale používat také polynomy 2. stupně.		
vzdálenost mezi uzly (jen B-spliny):				
PL:		Vzdálenosti mezi uzly jsou interně vypočítávány tak, aby byly vhodné. Řídí- cí systém ale může pracovat i se zadanými vzdálenostmi uzlů, které se udávají pomocí příkazu PL jako tzv. parametr interval-délka.		
<hodnota>:		Parametr interval-délka		
		Rozsah hodnot:	stejně jako rozměrový údaj dráhy	
Chování na přechodu na začátku splinové křivky (jen A- a C-spliny):				
BAUTO:		Žádný údaj pro chování na přechodu. Začátek vyplývá z polohy prvního bodu.		
BNAT:		Nulové zakřivení		
BTAN:		Tangenciální přechod k předcházejícímu bloku.		
Chování na přechodu na konci splinové křivky (jen A- a C-spliny):				
EAUTO:		Žádný údaj pro chování na přechodu. Konec vyplývá z polohy posledního bodu.		
ENAT:		Nulové zakřivení		

ETAN:	Tangenciální přechod k předcházejícímu bloku.

Poznámka

Naprogramované chování na přechodu nemá na B-spliny žádný vliv. B-spline je v počátečním a koncovém bodě vždy napojen tangenciálně k řídicímu mnohoúhelníku.

Okrajové podmínky

- Korekce rádiusu nástroje může být uplatněna.
- Monitorování kolize se uskutečňuje v projekci v rovině.

Příklady

Příklad 1: B-Spline

Programový kód 1 (všechny váhy 1)

```

N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0

```

5.2 Splinová interpolace (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Programový kód 2 (různé váhy)

```

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

```

Programová kód 3 (řídící mnohoúhelník)

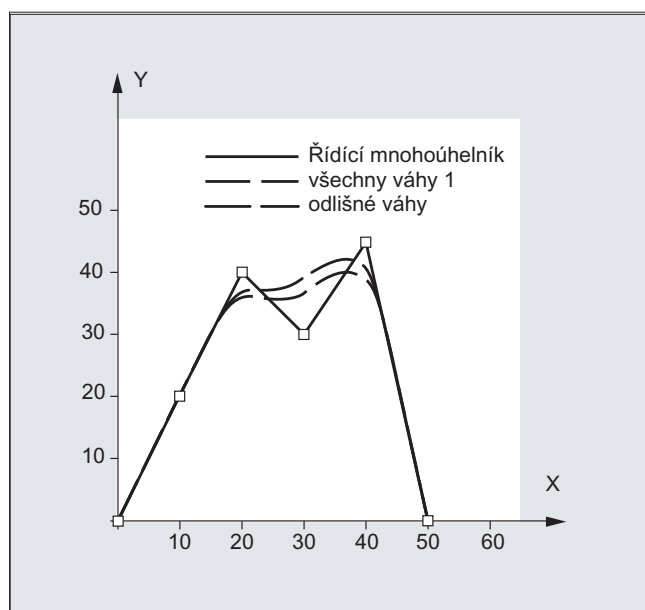
Komentář

```

N10 G1 X0 Y0 F300 G64
N20
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0

```

; odpadá



Příklad 2: C-spline, na začátku a na konci nulové zakřivení

Programový kód

```

N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10

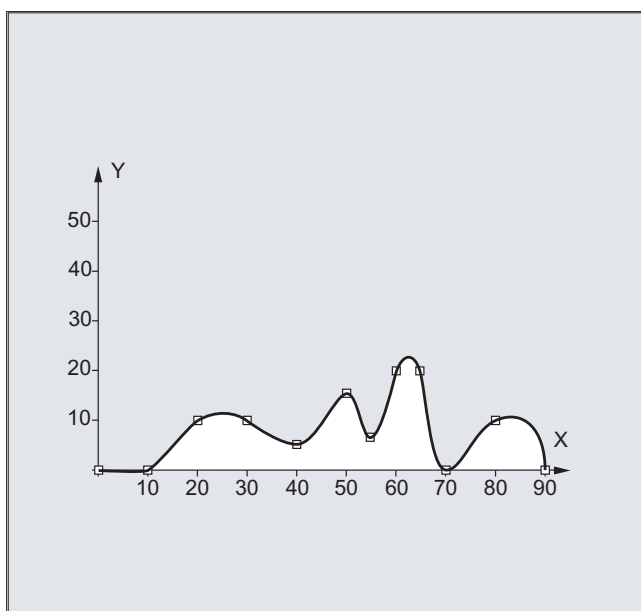
```

Programový kód

```

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

```

**Příklad 3: Splinová interpolace (A-spline) a transformace souřadného systému (ROT)**

Hlavní program:

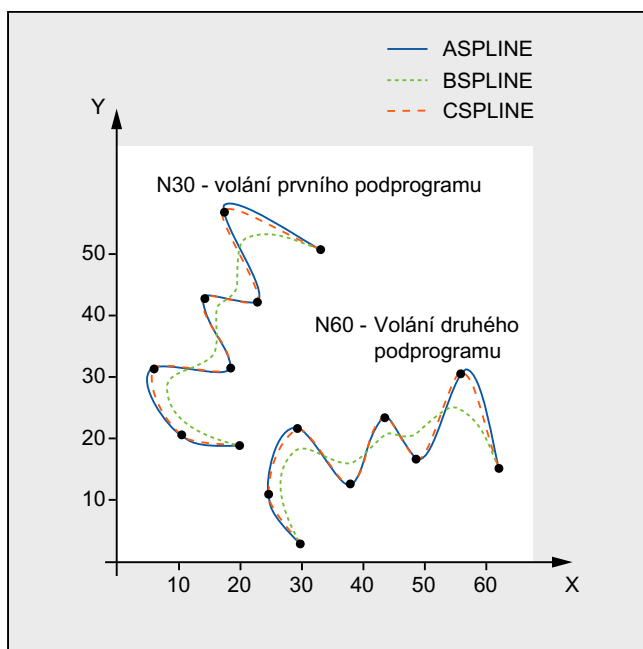
Programový kód	Komentář
N10 G00 X20 Y18 F300 G64	; najetí na počáteční bod
N20 ASPLINE	; aktivování typu interpolace A-Spline
N30 KONTUR	; první volání podprogramu
N40 ROT Z-45	; transformace souřadného systému: Otočení WCS o -45° okolo osy Z.
N50 G00 X20 Y18	; najíždění na počáteční bod kontury
N60 KONTUR	; druhé volání podprogramu
N70 M30	; konec programu

Podprogram "Kontura" (obsahuje souřadnice uzlového bodu):

Programový kód

```
N10 X20 Y18
N20 X10 Y21
N30 X6 Y31
N40 X18 Y31
N50 X13 Y43
N60 X22 Y42
N70 X16 Y58
N80 X33 Y51
N90 M1
```

V následujícím obrázku jsou vedle splinové křivky, která vyplynula z příkladu programování (ASPLINE), obsaženy ještě i splinové křivky, které by vznikly při aktivování interpolace pomocí B- nebo C-splinů (BSPLINE, CSPLINE):



Další informace

Výhody splinové interpolace

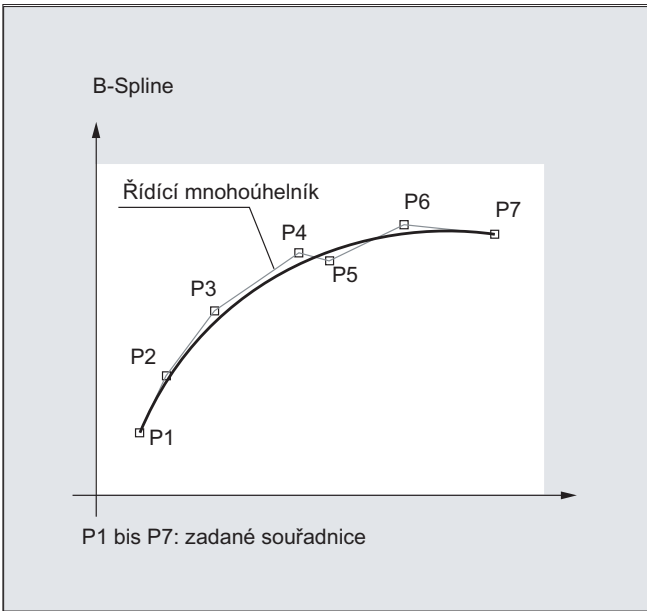
Použitím splinové interpolace se dá oproti použití přímkové interpolace G01 dosáhnout následujících výhod:

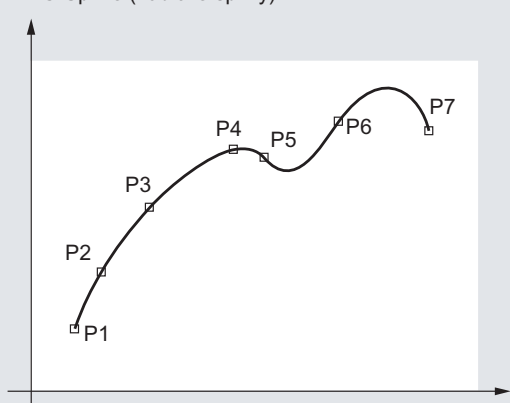
- Snížení počtu potřebných bloků výrobního programu pro popis kontury
- Hladší průběh kontury na přechodech mezi programovými bloky, čímž se šetří mechanika stroje

Vlastnosti a použití různých typů splinů

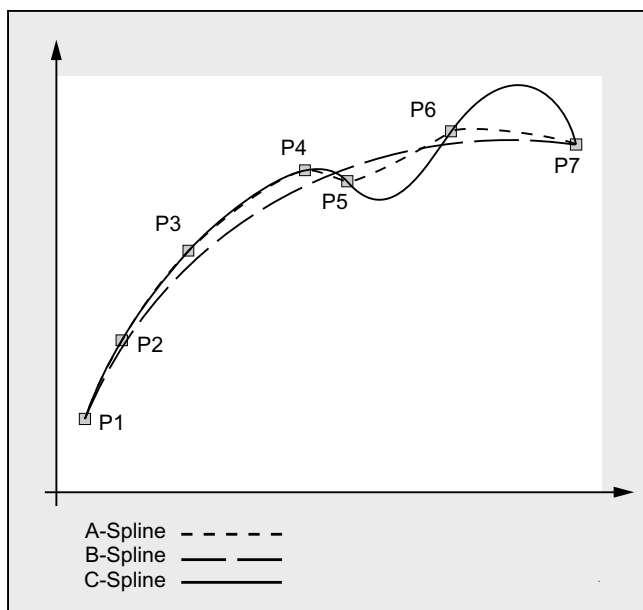
Typ splinu	Vlastnosti a použití
A-Spline	A-Spline (Akimovy spliny) P1 bis P7: zadané souřadnice Vlastnosti: • Probíhá přesně zadanými body. • Průběh křivky má tangenciální přechody, nemá však spojitě zakřivení • Zřídka způsobuje nežádoucí vibrace. • Oblast vlivu změny uzlového bodu je lokální, tzn. změna jednoho uzlového bodu bude mít vliv pouze na maximálně 6 sousedících uzlových bodů. Použití: A-spliny se hodí především pro interpolaci křivkových průběhů s velkými změnami stoupání (např. pro schodovité průběhy křivek).

5.2 Splinová interpolace (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Typ splinu	Vlastnosti a použití
B-Spline	 Vlastnosti: • Křivka neprochází zadanými uzlovými body, nýbrž v jejich blízkosti. Křivka je přitahována k uzlovým bodům. Přiřazováním váhy k uzlovým bodům pomocí faktoru je možné průběh křivky ještě dodatečně ovlivňovat. • Průběh křivky má tangenciální přechody a spojité zakřivení • Nezpůsobuje žádné nežádoucí vibrace. • Oblast vlivu změny uzlového bodu je lokální, tzn. změna jednoho uzlového bodu bude mít vliv pouze na maximálně 6 sousedících uzlových bodů. Použití: B-spliny jsou zamýšleny především jako rozhraní pro systémy CAD.

Typ splinu	Vlastnosti a použití
C-Spline	C-Spline (kubické spliny)  P1 bis P7: zadané souřadnice Vlastnosti: • Probíhá přesně zadanými body. • Průběh křivky má tangenciální přechody a spojitě zakřivení • Často způsobuje nežádoucí vibrace, a to zejména na místech s velkými změnami stoupání. • Oblast vlivu změny uzlového bodu je globální, tzn. změna jednoho uzlového bodu bude mít vliv celý průběh křivky. Použití: C-spliny mohou být výborně uplatněny tehdy, pokud se uzlové body nacházejí na analyticky známé křivce (kruh, parabola, hyperbola).

Porovnání třech typů splinů pomocí stejných uzlových bodů



Minimální počet bloků splinu

Příkazy G-funkcí `ASPLINE`, `BSPLINE` a `CSPLINE` spojují koncové body bloků pomocí splinů. Za tím účelem musí být při předběžném zpracování současně vypočítána řada bloků (koncových bodů). Velikost vyrovnávací paměti pro tento výpočet standardně činí 10 bloků. Ne každá informace v bloku je koncovým bodem splinu. Řídící systém však potřebuje z 10 bloků určitý počet bloků s koncovými body splinů:

Typ splinu	Minimální počet bloků splinu
A-Spline:	Z každých 10 bloků musí být minimálně 4 bloky splinové. Bloky komentářů a výpočty parametrů se nepočítají.
B-Spline:	Z každých 10 bloků musí být minimálně 6 bloků splinových. Bloky komentářů a výpočty parametrů se nepočítají.
C-Spline:	Potřebný minimální počet splinových bloků vyplývá z následujícího součtu: Hodnota z parametru MD20160 \$MC_CUBIC_SPLINE_BLOCKS + 1 Do parametru MD20160 se zadává počet bodů, přes které se splinový úsek vypočítává. Standardní nastavení je 8. Z každých 10 bloků musí proto být ve standardním případě minimálně 9 bloků splinových.

Poznámka

V případě překročení tolerovatelných hodnot se aktivuje alarm, stejně tak, jako když je osa podílející se na splinu naprogramována jako polohovací osa.

Shrnutí krátkých splineových bloků

Při splineové interpolaci mohou vzniknout krátké splineové bloky, což má za následek zbytečné snížení rychlosti pohybu po dráze. Pomocí funkce "Shrnutí krátkých splineových bloků" mohou být tyto bloky soustředěny takový způsobem, že výsledná délka bloků je dostatečně velká a nemá za následek snížení rychlosti pohybu po dráze.

Funkce se aktivuje pomocí kanálového strojního parametru:

MD20488 \$MC_SPLINE_MODE (Nastavení pro splineovou interpolaci)

Literatura:

Příručka k funkcím, Základní funkce, Režim řízení pohybu po dráze, přesné najetí, funkce Look Ahead (B1), kapitola Shrnutí krátkých splineových bloků

5.3 Splinový svazek (SPLINEPATH)

Pomocí příkazu `SPLINEPATH` jsou vybrány osy, které mají být interpolovány ve splinovém svazku. Při splinové interpolaci je možné pracovat s až osmi dráhovými osami.

Poznámka

Jestliže příkaz `SPLINEPATH` není explicitně naprogramován, zachází se s prvními třemi osami kanálu jako se splinovým svazkem.

Syntaxe

Definice splinového svazku se uskutečňuje v samostatném bloku:

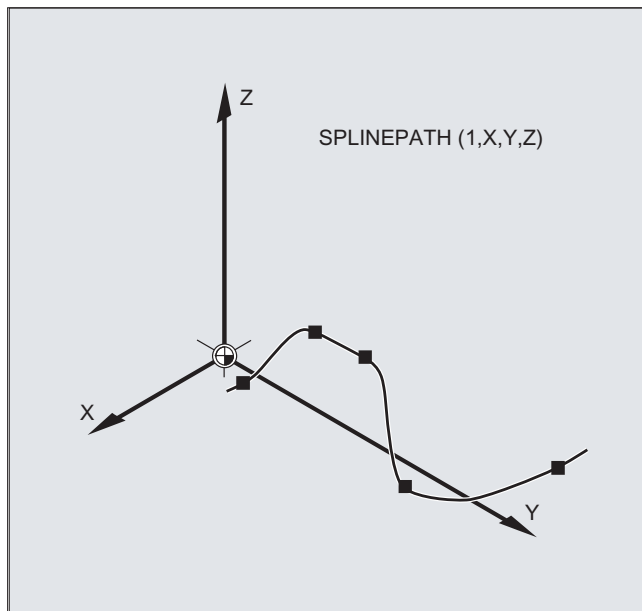
```
SPLINEPATH (n, X, Y, Z, ...)
```

Význam

<code>SPLINEPATH:</code>	Příkaz pro vytvoření splinového svazku.
<code>n:</code>	<code>=1</code> (pevná hodnota)
<code>X, Y, Z, ... :</code>	Identifikátory dráhových os, které mají být interpolovány ve splinovém svazku.

Příklad: Splinový svazek se třemi dráhovými osami

Programový kód	Komentář
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1,X,Y,Z)	; splinový svazek
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; C-Spline
N14 X30 Y40 Z50 A60 B70	; uzlové body
...	
N100 G1 X... Y...	; deaktivování splinové interpolace



5.4 Aktivování/deaktivování komprese NC bloků (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

Funkce pro kompresi lineárních bloků (a v závislosti na nastavení parametrů také kruhových bloků a bloků rychlým posuvem) se aktivuje/deaktivuje pomocí příkazů G-funkcí ze skupiny 30. Příkazy mají modální platnost.

Syntaxe

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

Význam

COMPON:	Aktivování funkce kompresoru COMPON
COMPCURV:	Aktivování funkce kompresoru COMPCURV
COMPCAD:	Aktivování funkce kompresoru COMPCAD
COMPSURF:	Aktivování funkce kompresoru COMPSURF
COMPOF:	Deaktivování momentálně aktivní funkce kompresoru

Poznámka

K dalšímu zlepšení jakosti povrchu se mohou použít ještě i funkce pro přechodová zaoblení G642 a funkce pro omezení ryvu SOFT. Tyto příkazy je zapotřebí zapsat na začátku programu.

Příklad: COMPCAD

Programový kód	Komentář
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	; aktivování: funkce pro přechodová zaoblení G642
N30 SOFT	; aktivování: Omezení ryvu SOFT
N40 COMPCAD	; aktivování: Funkce kompresoru COMPCAD
N50 STOPFIFO	
N24050 Z32.499	; 1. blok posuvu
N24051 X41.365 Z32.500	; 2. blok posuvu
...	
N99999 X... Z...	; poslední blok posuvu
COMPOF	; vypnutí funkce kompresoru
...	

5.5 Polynomická interpolace (POLY, POLYPATH, PO, PL)

V zásadě se u polynomické interpolace (POLY) nejedná o druh splinové interpolace. V první řadě je míněna jako rozhraní pro programování externě vytvořených splinových křivek. Tímto způsobem mohou být splinové úseky naprogramovány přímo.

Tento druh interpolace zbavuje NC systém nutnosti vypočítávat koeficienty polynomu. Může být optimálně využit tehdy, pokud jsou koeficienty dodávány přímo ze systému CAD nebo ze systému pro dodatečné zpracování.

Syntaxe

Polynom 3. stupně:

POLY PO[X]=(xe, a2, a3) PO[Y]=(ye, b2, b3) PO[Z]=(ze, c2, c3) PL=n

Polynomy 5. stupně a nová syntaxe polynomu:

POLY X=PO(xe, a2, a3, a4, a5) Y=PO(ye, b2, b3, b4, b5) Z=PO(ze, c2, c3, c4, c5)

PL=n

POLYPATH("AXES", "VECT")

Poznámka

Součet koeficientů polynomu naprogramovaných v NC bloku a os nesmí překročit maximální přípustný počet os najeden blok.

Význam

POLY :	Polynomická interpolace se aktivuje blokem s příkazem POLY.
POLYPATH :	Polynomická interpolace může být vybrána pro obě skupiny os AXIS nebo VECT
PO[identifikátor osy/proměnná] :	Koncové body a koeficienty polynomu
X, Y, Z :	Identifikátor osy
xe, ye, ze :	Údaj koncové pozice pro příslušné osy, rozsah hodnot a jednotka
a2, a3, a4, a5 :	Koeficienty a_2 , a_3 , a_4 a a_5 jsou popsány svou hodnotou, rozsahem hodnot a jednotkou. Poslední koeficient může odpadnout, jestliže má nulovou hodnotu.
PL :	Délka intervalu parametru, na kterém je polynom definován (definiční oblast funkce $f(p)$). Interval má svůj počátek vždy v 0, p může nabývat hodnot od 0 do PL. Teoretický rozsah hodnot pro PL: 0,0001 ... 99 999,9999 Upozornění Hodnota PL platí pro blok, ve kterém se nachází. Pokud hodnota PL není naprogramována, platí PL =1.

Aktivování/deaktivování polynomické interpolace.

Polynomická interpolace se ve výrobním programu zapíná příkazem G-funkce POLY.

Příkaz G-funkce POLY patří spolu s příkazy G0, G1, G2, G3, ASPLINE, BSPLINE a CSPLINE do 1. G-skupiny.

Osy, které jsou naprogramovány pouze svým názvem a koncovým bodem (např. X10) se pohybují lineárně. Pokud jsou tímto způsobem naprogramovány všechny osy v NC bloku, chová se řídicí systém jako při funkci G1.

Naprogramováním jiného příkazu z 1. skupiny (např. G0, G1) se polynomická interpolace implicitně deaktivuje.

Koeficienty polynomu

Hodnota PO (PO[]=), příp. ...=PO(...) udává všechny koeficienty polynomu pro danou osu. V souladu se stupněm polynomu se větší počet hodnot zadává oddělených čárkou. V rámci jednoho bloku mohou být pro jednotlivé osy zadány polynomy různých stupňů.

Podprogram POLYPATH

Pomocí příkazu POLYPATH(...) může být polynomická interpolace selektivně odblokována pro určité skupiny os.

Jen dráhové osy a doplňkové osy:	POLYPATH("AXES")
Jen orientační osy:	POLYPATH("VECT")
(při pohybech s transformací orientace)	

Osy, které nebyly tímto způsobem odblokovány, se pohybují jako osy lineární.

Standardně je polynomická interpolace uvolněna pro obě skupiny os.

Naprogramováním příkazu POLYPATH() bez parametrů se polynomická interpolace pro všechny osy deaktivuje.

Příklad:

Programový kód	Komentář
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	; aktivování polynomické interpolace
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)	; hodnota PL není naprogramována, platí PL=1
N30 G1 X... Y... Z.	; deaktivování polynomické interpolace
...	

Příklad: Nová syntaxe polynomu

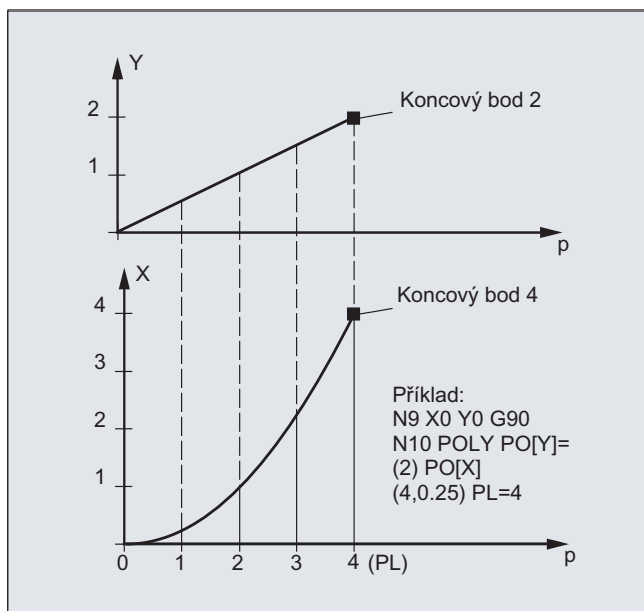
Dále platná syntaxe polynomu	Nová syntaxe polynomu
PO[identifikátor osy]=(.. , ..)	Identifikátor osy=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[proměnná]=IC(.. , ..)	proměnná=PO IC(.. , ..)

Příklad: Křivka v rovině X/Y

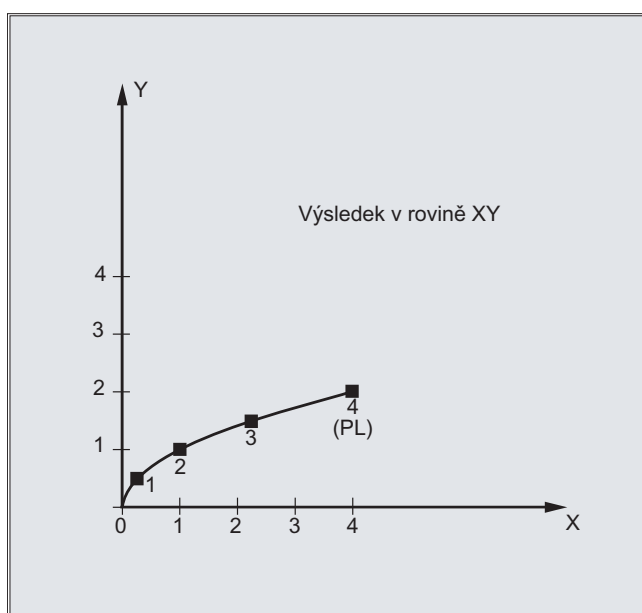
Programování

Programový kód
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

Průběh křivky X(p) a Y(p)



Průběh křivky v rovině XY



Popis

V obecném tvaru má polynomičká funkce následující podobu:

$$f(p) = a_0 + a_1p + a_2p^2 + \dots + a_np^n$$

kde: a_i : konstantní koeficienty ($i = 0, 1, \dots, n$)

p : Parametr

V řídicím systému je možné programovat maximálně polynomy 5. stupně:

$$f(p) = a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

Dosazením konkrétních hodnot koeficientů je možné vytvářet různé průběhy křivek, jako jsou přímky, paraboly a mocninné funkce.

Přímka vzniká dosazením $a_2 = a_3 = a_4 = a_5 = 0$:

$$f(p) = a_0 + a_1p$$

Kromě toho platí:

a_0 : Poloha osy na konci předcházejícího bloku

$p = PL$

$$a_1 = (x_E - a_0 - a_2 \cdot p^2 - a_3 \cdot p^3) / p$$

Je možné naprogramovat polynomy, **aniž** by byla polynomičká interpolace příkazem G-funkce `POLY` aktivována. V tomto případě nejsou interpolovány naprogramované polynomy, ale na naprogramované koncové body os se najíždí po lineárních drahách (`G1`). Teprve po explicitním aktivování polynomičké interpolace ve výrobním programu (`POLY`) jsou pohyby uskutečňovány po křivkových drahách odpovídajících naprogramovaným polynomům.

Zvláštnost: Polynom jmenovatele

Pro geometrické osy je možno pomocí příkazu `PO[] = (...)` bez udání názvu osy naprogramovat také společný polynom jmenovatele, což znamená, že pohyb geometrické osy bude potom interpolován jako podíl dvou polynomů.

Tímto způsobem se dají přesně zobrazit např. kuželosečky (kruh, elipsa, parabola, hyperbola).

Příklad:

Programový kód	Komentář
<code>POLY G90 X10 Y0 F100</code>	; geometrické osy lineárně najíždějí na pozici X10 Y0
<code>PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)</code>	; geometrické osy najíždějí po čtvrtkruhu na bod X0 Y10

Koeficient konstanty (a_0) polynomu jmenovatele je vždy nastaven na 1. Naprogramovaný koncový bod je nezávislý na příkazech G90 / G91.

Z naprogramovaných hodnot se vypočítají hodnoty $X(p)$ a $Y(p)$:

$$X(p) = (10 - 10 * p^2) / (1 + p^2)$$

$$Y(p) = 20 * p / (1 + p^2)$$

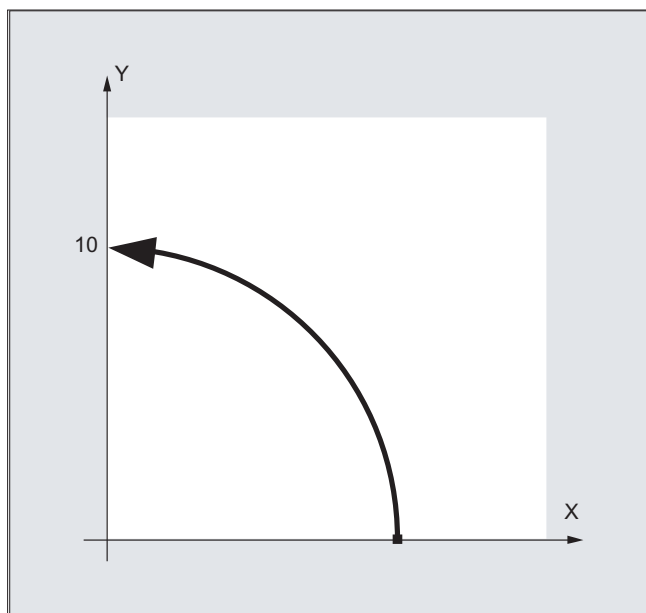
$$\text{kde } 0 \leq p \leq 1$$

Na základě naprogramovaných počátečních bodů, koncových bodů, koeficientu a_2 a $PL=1$ vyplývají následující mezivýsledky:

$$\text{Čitatel (X)} = 10 + 0 * p - 10 * p^2$$

$$\text{Čitatel (Y)} = 0 + 20 * p + 0 * p^2$$

$$\text{Jmenovatel} = 1 + p^2$$



Když je aktivována polynomická interpolace, je naprogramování polynomu jmenovatele s nulovými koeficienty v rámci intervalu $[0, PL]$ odmítnuto a aktivuje se alarm. Pohyb doplňkových os nemá na polynom jmenovatele žádný vliv.

Poznámka

Korekce rádiusu nástroje může být v případě polynomické interpolace pomocí příkazů G41, G42 aktivována/deaktivována a může se používat stejně jako při přímkové nebo kruhové interpolaci.

5.6 Nastavitelný vztah k dráze (SPATH, UPATH)

V případě polynomicke interpolace (POLY, ASPLINE, BSPLINE, CSPLINE, COMPON, COMPCURV) jsou polohy dráhové osy i definovány prostřednictvím polynomu $p_i(U)$. Křivkový parametr U se přitom v rámci NC bloku pohybuje v hodnotách od 0 do 1.

Pomocí příkazu FGROUPO jsou vybrány osy (osy FGROUPO), na které se vztahuje posuv po dráze F . Interpolace s konstantní rychlostí na úseku dráhy S u těchto os FGROUPO však v průběhu polynomicke interpolace zpravidla znamená nekonstantní změnu křivkového parametru U . Pro osy, které nejsou obsaženy v příkazu FGROUPO, si proto můžete vybrat ze dvou možností, jak tyto osy mají sledovat osy FGROUPO:

- Synchronizovaně úsekem dráhy S (SPATH)
- Synchronizovaně vůči křivkovému parametru U (UPATH)

Syntaxe

SPATH
UPATH

Význam

SPATH:	Osy, které nejsou obsaženy v příkazu FGROUPO, se budou pohybovat ve vztahu k úseku dráhy S
UPATH:	Osy, které nejsou obsaženy v příkazu FGROUPO, se budou pohybovat ve vztahu ke křivkovému parametru U

Poznámka

Funkce UPATH a SPATH určují také souvislost mezi polynomem F -slova (FPOLY, FCUB, FLIN) a pohybem po dráze.

Okrajové podmínky

Příkazy SPATH, příp. UPATH, nemají v následujících případech žádný význam:

- Lineární interpolace (G1)
- Kruhová interpolace (G2, G3)
- Bloky pro výrobu závitu (G33, G34, G35, G33x, G63)
- Všechny dráhové osy jsou obsaženy ve skupině FGROUPO

Příklad

Následující příklad ukazuje rozdíl mezi oběma druhy řízení pohybu.

Programový kód

```
N10 FGROUPO (X,Y,Z)
N15 G1 X0 A0 F1000 SPATH ; SPATH
N20 POLY PO[X]=(10,10) A10
```

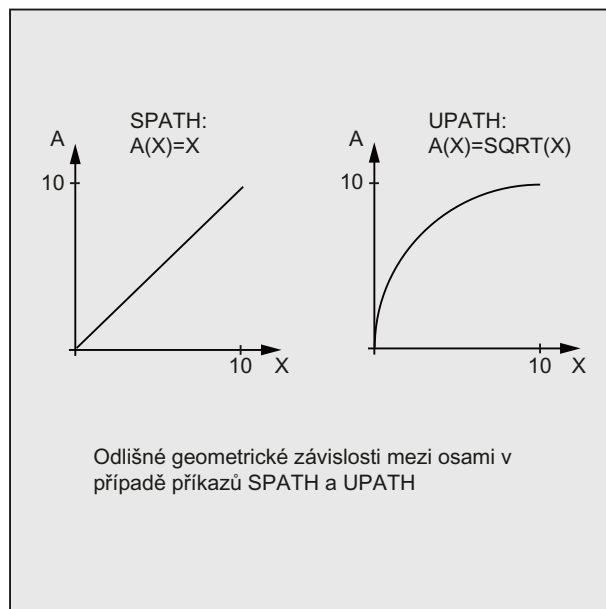
Programový kód

```

N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH          ; UPATH
N20 POLY PO[X]=(10,10) A10

```

V obou úsecích programu závisí v bloku N20 dráha S os ve skupině FGROUP na druhé mocnině křivkového parametru U. Z toho vyplývají různé polohy synchronní osy A podél dráhy osy X podle toho, jestli je aktivován příkaz *SPATH* nebo *UPATH*.

**Další informace****Chování řídicího systému při resetu a strojní parametry/parametry doplňků**

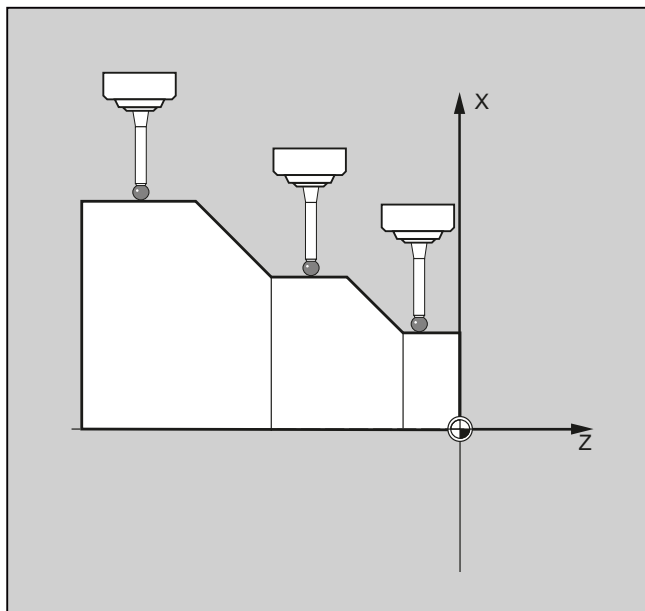
Po resetu je prostřednictvím parametru MD20150 \$MC_GCODE_RESET_VALUES[44] aktivován příkaz určité G-funkce (45. G-skupina).

Hodnota základního nastavení pro druh přechodových zaoblení je definována parametrem MD20150 \$MC_GCODE_RESET_VALUES[9] (10. G-skupina).

Osový strojní parametr MD33100 \$MA_COMPRESS_POS_TOL[<n>] má rozšířený význam: Obsahuje tolerance pro funkci kompresoru a pro přechodová zaoblení pomocí příkazu G642.

5.7 Měření se spínací sondou (MEAS, MEAW)

Pomocí funkce "Měření se spínací sondou" se najíždí na skutečné polohy na obrobku, a když je zaznamenána skoková změna (sepnutí) signálu spínací sondy, pro všechny osy naprogramované v bloku měření se změní daná pozice a ta se pro každou z těchto os zapíše do odpovídající paměťové buňky.



Pro programování této funkce máte k dispozici následující dvě pevné adresy:

- **MEAS**
Pomocí příkazu **MEAS** se zbytková dráha mezi skutečnou a požadovanou polohou vymaže.
- **MEAW**
Příkaz **MEAW** se používá pro měřicí úlohy, u kterých se má v každém případě najet na naprogramovanou pozici.

Příkazy **MEAS** a **MEAW** mají blokovou platnost a jsou naprogramovány spolu s příkazy pro pohyb. Posuv a druh interpolace (G0, G1, ...), stejně jako počet os, přitom musí být přizpůsobeny konkrétnímu měřicímu problému.

Syntaxe

MEAS=<TE> G... X... Y... Z...

MEAW=<TE> G... X... Y... Z...

Význam

MEAS:	Příkaz: Měření s vymazáním zbytkové dráhy	
	Platnost:	bloková
MEAW:	Příkaz: Měření bez vymazání zbytkové dráhy	
	Platnost:	bloková

<TE>:	Událost sepnutí za účelem spuštění měření		
	Typ:	INT	
	Rozsah hodnot:	-2, -1, 1, 2	
	Význam:		
	(+)1	náběžná hrana signálu měřicí sondy 1 (na měřicím vstupu 1)	
	-1	sestupná hrana signálu měřicí sondy 1 (na měřicím vstupu 1)	
	(+)2	náběžná hrana signálu měřicí sondy 2 (na měřicím vstupu 2)	
	-2	sestupná hrana signálu měřicí sondy 2 (na měřicím vstupu 2)	
G . . . :	Upozornění: Existují maximálně 2 měřicí sondy (v závislosti na úrovni vybavení).		
	Druh interpolace, např. G0, G1, G2 nebo G3		
X . . . Y . . . Z . . . :	Koncové body v kartézských souřadnicích		

Příklad:

Programový kód	Komentář
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; blok měření s měřicí sondou na prvním měřicím vstupu a přímková interpolace Je automaticky generováno zastavení předběžného zpracování.
...	

Další informace**Stav měřicí úlohy**

Jestliže je v programu zapotřebí vyhodnotit, zda měřicí sonda sepnula nebo ne, může být zjišťován obsah stavové proměnné \$AC_MEA[<n>] (<n> = číslo měřicí sondy).

Hodnota	Význam
0	Měřicí úloha není splněna
1	Měřicí úloha byla úspěšně dokončena (měřicí sonda sepnula).

Poznámka

Pokud je měřicí sonda v programu vychýlena, je proměnná nastavena na 1. Při spuštění bloku měření se proměnná automaticky nastaví podle počátečního stavu sondy.

Odečítání změřených hodnot

Jsou zaznamenávány pozice všech dráhových a polohovacích os v bloku, které se pohybovaly (maximální počet os závisí na konfiguraci řídicího systému). V případě příkazu `MEAS` se pohyb po sepnutí měřicí sondy definovaným způsobem zastaví.

Poznámka

Jestliže je v měřicím bloku naprogramována geometrická osa, jsou změřené hodnoty uloženy pro všechny momentálně definované geometrické osy.

Jestliže je v měřicím bloku naprogramována osa podílející se na nějaké transformaci, jsou změřené hodnoty uloženy i pro všechny na této transformaci se podílející osy.

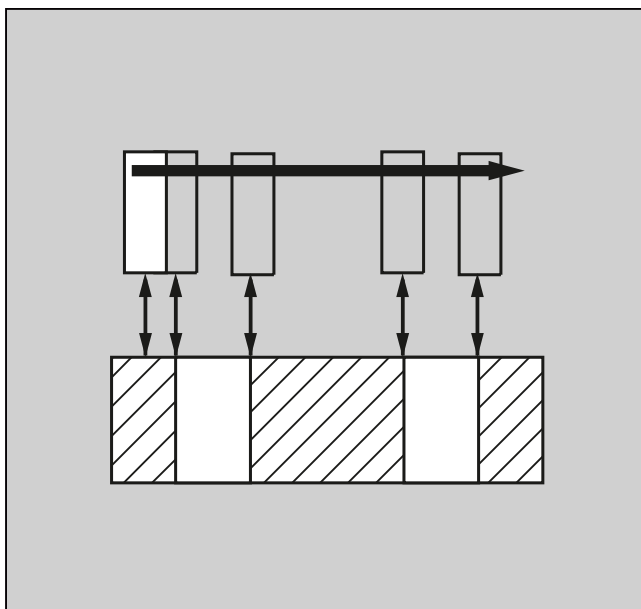
Načtení výsledků měření

Výsledky měření pro osy, které byly měřicí sondou zaznamenány, mohou být načítány prostřednictvím následujících systémových proměnných:

- `$AA_MM[<osa>]`
Výsledky měření v souřadném systému stroje
- `$AA_MW[<osa>]`
Výsledky měření v souřadném systému obrobku

5.8 Měření pomocí specifické osy (MEASA, MEAWA, MEAC) (volitelný doplněk)

V případě měření se specifickou osou se může používat větší počet měřicích sond a větší počet měřicích systémů.



Pro programování těchto funkcí máte k dispozici klíčová slova **MEASA**, **MEAWA** a **MEAC**.

Pomocí příkazů **MEASA**, příp. **MEAWA** jsou pro příslušnou naprogramovanou osu zaznamenány v každém měření až čtyři změřené hodnoty, které se pak ukládají podle události spouštění do systémových proměnných.

Kontinuální měřicí úlohy je možné uskutečňovat pomocí příkazu **MEAC**. V tomto případě jsou výsledky měření ukládány do proměnných typu FIFO.

Syntaxe

```
MEASA [<osa>] = (<režim>, <TE1>, ..., <TE4>)
MEAWA [<osa>] = (<režim>, <TE1>, ..., <TE4>)
MEAC [<osa>] = (<režim>, <paměť měření>, <TE1>, ..., <TE4>)
```

Poznámka

Příkazy **MEASA** a **MEAWA** mají blokovou platnost a mohou být naprogramovány společně v jednom bloku. Pokud je oproti tomu příkaz **MEASA/MEAWA** naprogramován v jednom bloku spolu s příkazem **MEAS/MEAW**, je generováno chybové hlášení.

Význam

MEASA:	Klíčové slovo: Měření se specifickou osou a s vymazáním zbytkové dráhy	
	Platnost:	bloková

MEAWA:	Klíčové slovo: Měření se specifickou osou a bez vymazání zbytkové dráhy	
	Platnost:	bloková
MEAC:	Klíčové slovo: Kontinuální měření se specifickou osou bez vymazání zbytkové dráhy	
	Platnost:	bloková
<osa>:	Název kanálové osy použité pro měření	
<režim>:	Dvumístné číslo pro zadávání provozního režimu (režim měření a měřicí systém)	
	Místo jednotek (režim měření):	
	0	Přerušení měřicí úlohy.
	1	Až 4 různé spouštěcí události, které mohou být aktivovány současně.
	2	Až 4 spouštěcí události, které mohou být aktivovány jedna po druhé.
	3	Až 4 spouštěcí události, které mohou být aktivovány jedna po druhé, avšak žádné monitorování spouštěcí události 1 při spuštění (alarmy 21700/21703 jsou potlačeny).
	Upozornění V případě příkazu MEAC není tento režim možný.	
	Místo desítek (měřicí systém):	
	0 (žádný údaj)	aktivní měřicí systém
	1	Měřicí systém 1
<TE>:	2	Měřicí systém 2
	3	Oba měřicí systémy
	Událost sepnutí za účelem spuštění měření	
	Typ:	INT
	Rozsah hodnot:	-2, -1, 1, 2
	Význam:	
	(+)1	náběžná hrana měřicí sondy 1
	-1	sestupná hrana měřicí sondy 1
<paměť měření>:	(+)2	náběžná hrana měřicí sondy 2
	-2	sestupná hrana měřicí sondy 2
Číslo FIFO (zásobníková paměť)		

Příklady

Příklad 1: Měření se specifickou osou a s vymazáním zbytkové dráhy v režimu 1 (vyhodnocování v časové posloupnosti)

a) s 1 měřicím systémem

Programový kód	Komentář
...	
N100 MEASA[X]=(1,1,-1) G01 X100 F100	; měření v režimu 1 s aktivním měřicím systémem Čekání na měřicí signál s náběžnou/sestupnou hranou signálu měřicí sondy 1 na dráze po X=100.

5.8 Měření pomocí specifické osy (MEASA, MEAWA, MEAC) (volitelný doplněk)

Programový kód	Komentář
N110 IF \$AC_MEA[1]==FALSE GOTOF ENDE	; kontrola úspěšného měření
N120 R10=\$AA_MM1[X]	; uložit změřenou hodnotu patřící k první naprogramované spouštěcí události (náběžná hrana)
N130 R11=\$AA_MM2[X]	; uložit změřenou hodnotu patřící ke druhé naprogramované spouštěcí události (sestupná hrana)
N140 ENDE:	

b) se 2 měřicími systémy

Programový kód	Komentář
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; měření v režimu 1 s oběma měřicími systémy Čekání na měřicí signál s náběžnou/sestupnou hranou signálu měřicí sondy 1 na dráze po X=100.
N210 IF \$AC_MEA[1]==FALSE GOTOF ENDE	; kontrola úspěšného měření
N220 R10=\$AA_MM1[X]	; uložení změřené hodnoty z měřicího systému 1 při náběžné hraně signálu
N230 R11=\$AA_MM2[X]	; uložení změřené hodnoty z měřicího systému 2 při náběžné hraně signálu
N240 R12=\$AA_MM3[X]	; uložení změřené hodnoty z měřicího systému 1 při sestupné hraně signálu
N250 R13=\$AA_MM4[X]	; uložení změřené hodnoty z měřicího systému 2 při sestupné hraně signálu
N260 ENDE:	

Příklad 2: Měření se specifickou osou a s vymazáním zbytkové dráhy v režimu 2 (vyhodnocování v naprogramované posloupnosti)

Programový kód	Komentář
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; měření v režimu 2 s aktivním měřicím systémem Čekání na měřicí signály v posloupnosti náběžná hrana měřicí sondy 1, sestupná hrana měřicí sondy 1, náběžná hrana měřicí sondy 2, sestupná hrana měřicí sondy 2 na dráze po X=100.
N110 IF \$AC_MEA[1]==FALSE GOTOF MESSTASTER2	; kontrola úspěšného měření měřicí sondou 1
N120 R10=\$AA_MM1[X]	; uložit změřenou hodnotu patřící k první naprogramované spouštěcí události (náběžná hrana měřicí osy 1)
N130 R11=\$AA_MM2[X]	; uložit změřenou hodnotu patřící ke druhé naprogramované spouštěcí události (náběžná hrana měřicí osy 1)
N140 MESSTASTER2:	

Programový kód	Komentář
N150 IF \$AC_MEA[2]==FALSE GOTOF ENDE	; kontrola úspěšného měření měřicí sondou 2
N160 R12=\$AA_MM3[X]	; uložit změřenou hodnotu patřící ke třetí naprogramované spouštěcí události (náběžná hrana měřicí osy 2)
N170 R13=\$AA_MM4[X]	; uložit změřenou hodnotu patřící ke čtvrté naprogramované spouštěcí události (náběžná hrana měřicí osy 2)
N180 ENDE:	

Příklad 3: Kontinuální měření se specifickou osou v režimu 1 (vyhodnocování v časové posloupnosti)

a) Měření až 100 změřených hodnot

Programový kód	Komentář
...	
N110 DEF REAL MESSWERT[100]	
N120 DEF INT Schleife=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; měření v režimu 1 s aktivním měřicím systémem, s ukládáním změřených hodnot do proměnné \$AC_FIFO1, čekání na měřicí signál se sestupnou hranou signálu měřicí sondy 1 na dráze po X=1000
N135 STOPRE	
N140 MEAC[X]=(0)	; ukončení měření po dosažení polohy osy
N150 R1=\$AC_FIFO1[4]	; ukládání počtu zaznamenaných změřených hodnot do parametru R1
N160 FOR Schleife=0 TO R1-1	
N170 MESSWERT[Schleife]=\$AC_FIFO1[0]	; načítání a ukládání změřených hodnot z proměnné \$AC_FIFO1
N180 ENDFOR	

b) Měření s vymazáním zbytkové dráhy po 10 změřených hodnotách

Programový kód	Komentář
...	
N10 WHEN \$AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; vymazání zbytkové dráhy
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC[X]=(0)	
N40 R1=\$AC_FIFO1[4]	; počet změřených hodnot
...	

c) Měření sestupného/náběžného boku zubu pomocí 2 měřících sond

Programový kód	Komentář
...	
N110 DEF REAL MESSWERT[16]	
N120 DEF INT Schleife=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; měření v režimu 1 s aktivním měřicím systémem, s ukládáním změřených hodnot do proměnné \$AC_FIFO1, čekání na měřicí signál v posloupnosti sestupná hrana signálu měřicí sondy 1, vzestupná hrana měřicí sondy 2 na dráze po X=100
N140 STOPRE	; zastavení předběžného zpracování
N150 MEAC[X]=(0)	; ukončení měření po dosažení polohy osy
N160 R1=\$AC_FIFO1[4]	; ukládání počtu zaznamenaných změřených hodnot do parametru R1
N170 FOR Schleife=0 TO R1-1	
N180 MESSWERT[Schleife]=\$AC_FIFO1[0]	; načítání a ukládání změřených hodnot z proměnné \$AC_FIFO1
N190 ENDFOR	

Další informace

Měřicí úloha

Programování měřicích úloh se může uskutečňovat ve výrobním programu nebo ze synchronní akce (viz kapitola "Synchronní akce (Strana 621)"). Pro každou osu přitom může být v jednom a totéž okamžiku aktivní jen jedna měřicí úloha.

Poznámka

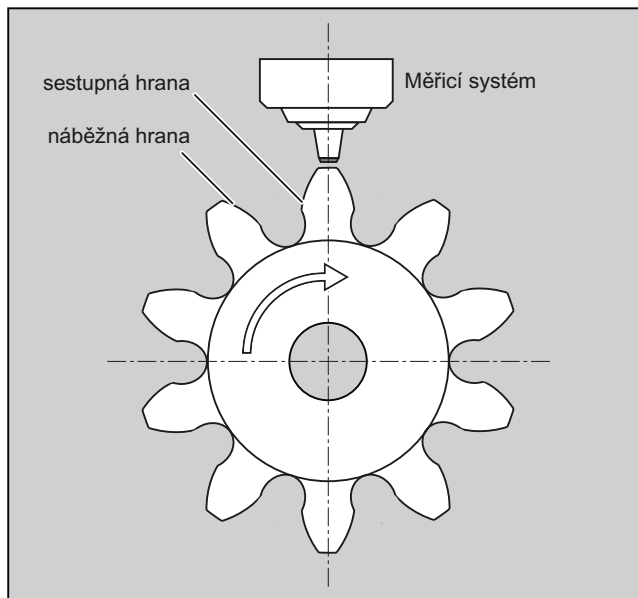
Posuv je potřeba přizpůsobit příslušnému měřicímu problému.

U příkazů MEASA a MEAWA mohou být správné výsledky zaručeny jedině při hodnotách posuvu, při kterých se nemůže vyskytnout více než jedna stejná a více než 4 různé spouštěcí události na jeden takt regulátoru polohy.

V případě kontinuálního měření s funkcí MEAC nesmí být poměr mezi taktem interpolátoru a taktem polohového regulátoru větší než 1:8.

Spouštěcí událost

Spouštěcí událost se skládá z čísla měřicí sondy a ze spouštěcího kritéria (náběžná nebo sestupná hrana) signálu měřicí sondy.



Pro každé měření mohou být zpracovány vždy až 4 spouštěcí události spínající měřicí sondy, tedy až dvě měřicí sondy, každá se dvěma hranami měřicího signálu. Posloupnost zpracování, jakož i také maximální počet spouštěcích událostí jsou přitom závislé na zvoleném režimu.

Poznámka

Pro měřicí režim 1 platí: Stejná spouštěcí událost smí být v dané měřicí úloze naprogramována jen jednou!

V případě příkazu MEAC může být počet změřených hodnot zvýšen díky použití telegramu PROFIBUS 395 na celkem 8 změřených hodnot s náběžnou hranou a 8 hodnot s klesající hranou na jednu spouštěcí událost a takt regulátoru polohy.

- Jedna měřicí sonda: 8 změřených hodnot s náběžnou a 8 s klesající hranou signálu
- Dvě měřicí sondy: 4 změřené hodnoty s náběžnou a 4 s klesající hranou signálu na jednu měřicí sondu

Díky tomu je možno při použití telegramu PROFIBUS 395 realizovat vyšší rychlosti posuvu nebo otáčky.

Literatura:

Příručka k funkcím, Rozšiřovací funkce; Měření (M5), kapitola: Axiální měření

Provozní režim

Pomocí první číslice (místo desítek) provozního režimu se vybírá požadovaný měřicí systém. Jestliže je k dispozici jen jeden měřicí systém, je však naprogramován ten druhý, automaticky se použije ten, který je k dispozici.

Pomocí druhé číslice (místo jednotek) se nastavuje požadovaný režim měření. Tímto způsobem může být měřicí operace možností příslušného řídicího systému.

- **Režim 1**
Vyhodnocování spouštěcích událostí se uskutečňuje v časové posloupnosti, v jaké se vyskytly. V tomto režimu může být při použití modulů se šesti osami naprogramována jen jedna spouštěcí událost, příp. při zadání většího počtu spouštěcích událostí se režim automaticky přepne do režimu 2 (bez hlášení).
- **Režim 2**
Vyhodnocování spouštěcích událostí se uskutečňuje v naprogramované posloupnosti.
- **Režim 3**
Vyhodnocování spouštěcí události se uskutečňuje v naprogramované posloupnosti, neprovádí se však žádné monitorování spouštěcí události 1 při spuštění.

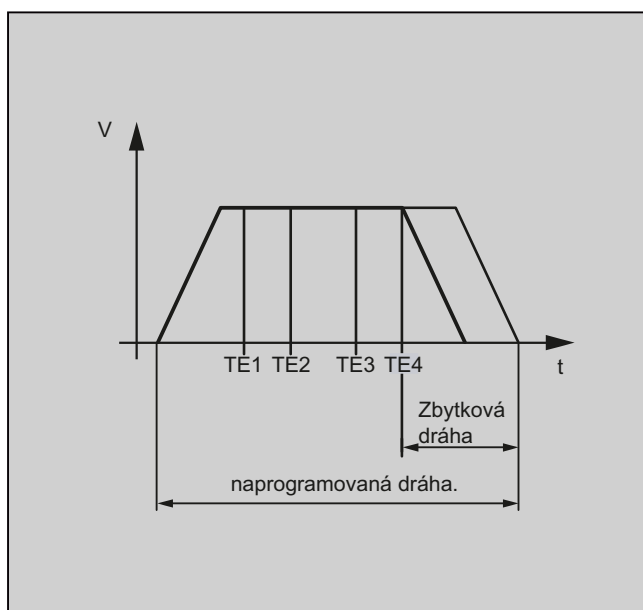
Poznámka

Pokud jsou použity 2 měřicí systémy, mohou být naprogramovány pouze dvě spouštěcí události.

Měření s a bez vymazání zbytkové dráhy

Jestliže je naprogramován příkaz **MEASA**, vymazání zbytkové dráhy se uskuteční až po zaznamenání všech požadovaných měřených hodnot.

Příkaz **MEAWA** se používá v případě speciálních měřicích úloh, u kterých se má v každém případě najet na naprogramovanou pozici.



Poznámka

Příkaz **MEASA** nemůže být naprogramován v synchronních akcích. Místo toho může být jako synchronní akce naprogramován příkaz **MEAWA** plus vymazání zbytkové dráhy.

Pokud je ze synchronní akce spuštěna měřicí úloha s příkazem **MEAWA**, jsou změřené hodnoty k dispozici pouze v souřadném systému stroje.

Výsledky měření pro příkazy MEASA, MEAWA

Výsledky měření jsou k dispozici v následujících systémových proměnných:

- V souřadném systému stroje:

\$AA_MM1 [<osa>]	změřená hodnota z naprogramovaného měřicího systému při spouštění události 1
...	...
\$AA_MM4 [<osa>]	změřená hodnota z naprogramovaného měřicího systému při spouštění události 4

- v souřadném systému obrobku:

\$AA_MW1 [<osa>]	změřená hodnota z naprogramovaného měřicího systému při spouštění události 1
...	...
\$AA_MW4 [<osa>]	změřená hodnota z naprogramovaného měřicího systému při spouštění události 4

Geometrické osy / transformace

Jestliže má být měření pomocí jedné osy spuštěno pro geometrickou osu, musí být stejná měřicí úloha explicitně naprogramována také pro všechny zbývající geometrické osy. Totéž platí i pro osy, které se podílejí na transformaci.

Příklady:

```
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

Jestliže se měřicí úloha uskutečňuje se dvěma měřicími systémy, zaznamenává se pro příslušnou osu každá s obou možných spouštěcích událostí z obou měřicích systémů. Obsazení rezervovaných proměnných je předem definováno takto:

\$AA_MM1 [<osa>]	nebo	\$AA_MW1 [<osa>]	změřená hodnota z měřicího systému 1 při spouštěcí události 1
\$AA_MM2 [<osa>]	nebo	\$AA_MW2 [<osa>]	změřená hodnota z měřicího systému 2 při spouštěcí události 1

5.8 Měření pomocí specifické osy (MEASA, MEAWA, MEAC) (volitelný doplněk)

\$AA_MM3 [<osa>] nebo \$AA_MW3 [<osa>] změřená hodnota z měřicího systému 1 při spouštěcí události 2

\$AA_MM4 [<osa>] nebo \$AA_MW4 [<osa>] změřená hodnota z měřicího systému 2 při spouštěcí události 2

Systémové proměnné

Stav měřicí sondy je k dispozici v následujících systémových proměnných:

\$A_PROBE[<n>]

Hodnota	Význam
1	Měřicí sonda je vychýlena
0	Měřicí sonda není vychýlena

Omezení měřicí sondy je k dispozici v následujících systémových proměnných:

\$A_PROBE_LIMITED[<n>]

Hodnota	Význam
1	Omezení měřicí sondy je aktivní
0	Omezení měřicí sondy není aktivní

<n> = měřicí sonda

Literatura

Příručka pro sestavy systémových proměnných

Stav měřicí operace pro příkazy MEASA, MEAWA

Pokud je v programu zapotřebí vyhodnocování, pak je možno zjistit stav měřicí operace pomocí proměnné \$AC_MEA[<n>], kde <n> = číslo měřicí sondy. Jakmile jsou úspěšně zaznamenány všechny v bloku naprogramované spouštěcí události měřicí sondy <n>, je této proměnné dosazena hodnota 1. Jinak má hodnotu 0.

Poznámka

Pokud je měření spuštěno ze synchronní akce, proměnná \$AC_MEA se neaktualizuje. V tomto případě je potřeba zjistit stav nových signálů rozhraní PLC DB31, ... DBX62.3, příp. proměnné \$AA_MEA[<osa>], která má stejný význam.

Význam:

\$AA_MEA[<osa>]==1: Měření je aktivní

\$AA_MEA[<osa>]==0: Měření není aktivní

Kontinuální měření (MEAC)

V případě funkce MEAC jsou změřené hodnoty v souřadném systému stroje a ukládají se do zadané paměti typu FIFO[n] (zásobníková paměť). Jestliže je nastaveno, že měření má probíhat se dvěma měřicími sondami, jsou hodnoty změřené druhou sondou ukládány odděleně do pro tento účel vytvořené paměti FIFO [n+1] (nastavitelné pomocí strojního parametru).

Paměť FIFO je zásobníkovou pamětí, do které jsou změřené hodnoty ukládány v proměnné \$AC_FIFO na principu "první dovnitř - první ven" (viz kapitola "Synchronní akce (Strana 621)").

Poznámka

Obsah paměti FIFO je možné ze zásobníkové paměti přečíst jen jednou. Pokud potřebujete změřené údaje používat vícekrát, musíte je přechodně uložit do uživatelské paměti.

Pokud počet změřených hodnot pro paměť FIFO překročí maximální možný počet definovaný ve strojním parametru, měření se automaticky ukončí.

Nekonečné měření je možné realizovat pomocí cyklického načítání změřených hodnot. Načítání se přitom musí uskutečňovat s minimálně stejnou četností, s jakou probíhá přisun nových změřených hodnot.

Literatura:

- Příručka k funkcím, Synchronní akce; Podrobný popis, kapitola: Parametr (\$AC_FIFO)
- Příručka k funkcím, Rozšiřovací funkce; Měření (M5), kapitola: Axiální měření

Ochrana před chybami v programu

Jsou rozpoznávány následující chyby v programování a při jejich výskytu se vypíše chyba:

- Příkaz MEASA/MEAWA je naprogramován v jednom bloku spolu s příkazem MEAS/MEAW.
Příklad:
N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
- MEASA/MEAWA s počtem parametrů < 2 nebo > 5
Příklad:
N01 MEAWA[X]=(1) G01 F100 POS[X]=100
- MEASA/MEAWA se spouštěcí událostí nerovnající se 1/ -1/ 2/ -2
Příklad:
N01 MEASA[B]=(1,1,3) B100
- MEASA/MEAWA s nesprávným režimem
Příklad:
N01 MEAWA[B]=(4,1) B100
- MEASA/MEAWA s dvakrát naprogramovanou spouštěcí událostí
Příklad:
N01 MEASA[B]=(1,1,-1,2,-1) B100
- MEASA/MEAWA a chybějící geometrická osa
Příklad:
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;GEO-Achse X/Y/Z
- Měřicí úloha u geometrických os bez jednotek
Příklad:
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100

5.9 Speciální funkce pro uživatele OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

Adresy OEM

Význam adres OEM určuje uživatel OEM. Funkce se instalují prostřednictvím cyklů překladače. Je rezervováno 5 adres OEM (OMA1 ... OMA5). Identifikátory adres jsou nastavitelné. Adresy OEM jsou přípustné v každém bloku.

Rezervovaná volání příkazů G-funkcí

Pro uživatele OEM jsou rezervována následující volání příkazů G-funkcí:

- OEMIPO1, OEMIPO2 (ze skupiny G-funkcí č. 1)
- G810 ... G819 (skupina G-funkcí č. 31)
- G820 ... G829 (skupina G-funkcí č. 32)

Funkce se instalují prostřednictvím cyklů překladače.

Funkce a podprogramy

Kromě toho mohou uživatelé OEM zakládat také předdefinované funkce a podprogramy s předáváním parametrů.

Poznámka

Simulace obrobku

Až do verze SW 4.4 nebyly při simulaci obrobku podporovány žádné cykly překladače (CC), od verze SW 4.4 jsou podporovány jen vybrané.

Příkazy jazyka ve výrobním programu, které spadají do nepodporovaných cyklů překladače (OMA1 ... OMA5, OEMIPO1/2, G810 ... G829, některé procedury a funkce), pokud nejsou individuálně ošetřeny, mají proto za následek alarmové hlášení a přerušení simulace.

Řešení: Pro chybějící specifické prvky jazyka CC ve výrobním programu zajistěte individuální zpracování (zjišťování stavu \$P_SIM).

Příklad:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;při simulaci CC nejsou aktivní
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 Snížení posuvu se zpožděním v rozích (FENDNORM, G62, G621)

V případě automatického zpoždění v rozích se hodnota posuvu krátce před dosažením příslušného rohu snižuje podle křivky zvonovitého tvaru. Kromě toho může být pomocí nastavovaných parametrů definován přírůstek rozměru pro odpovídající chování nástroje v rohu. Jedná se o následující:

- Začátek a konec snížené hodnoty posuvu
- Korekce, o kterou je posuv snížen
- Rozpoznávání odpovídajícího rohu

Rohy, kterých se toto snížení posuvu týká, budou ty, jejichž vnitřní úhel je menší než úhel stanovený příslušným nastavovaným parametrem.

Prostřednictvím předdefinované hodnoty `FENDNORM` se funkce automatické korekce v rozích vypíná.

Literatura

/FBFA/, Příručka Popis funkcí, Dialekty ISO

Syntaxe

`FENDNORM`

`G62 G41`

`G621`

Význam

<code>FENDNORM:</code>	Deaktivování automatického zpoždění v rozích
<code>G62:</code>	Zpoždění na vnitřních rozích při aktivní korekci rádiusu nástroje
<code>G621:</code>	Zpoždění na všech rozích při aktivní korekci rádiusu nástroje

Příkaz G62 je v platnosti jen na vnitřních rozích, když je

- aktivní korekce rádiusu nástroje `G41`, `G42` a
- aktivní režim řízení pohybu po dráze `G64`, `G641`

Na odpovídající roh se najíždí se sníženou hodnotou posuvu, která vyplývá z následujícího vzorce:

$F * (\text{korekce pro snížení hodnoty posuvu}) * \text{korekce snížení hodnoty posuvu}$

Maximálního možného snížení rychlosti posuvu je dosaženo přesně tehdy, když nástroj, vztaženo na dráhu jeho středu, dosáhne bodu, v němž se má uskutečnit změna směru na příslušném rohu.

G621 funguje analogicky k příkazu G62 na každém rohu, na který osy stanovené příkazem `FGROUP` narazí.

5.11 Programovatelné kritérium konce pohybu (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Podobně jako kritérium přechodu na další blok v případě dráhové interpolace (G601, G602 a G603) může být ve výrobním programu, příp. v synchronních akcích pro příkazové/PLC osy naprogramováno u interpolace jednotlivých os kritérium konce pohybu

V závislosti na tom, které kritérium konce pohybu je nastaveno, jsou bloky výrobního programu, příp. bloky technologických cyklů s pohyby jednotlivých os ukončeny různě rychle. Totéž platí pro PLC prostřednictvím FC15/16/18.

Syntaxe

```

FINEA [<osa>]
COARSEA [<osa>]
IPOENDA [<osa>]
IPOBRKA (<osa>[, <časový okamžik>])
ADISPOSA (<osa>[, <režim>, <velikost okna>])

```

Význam

FINEA:	Kritérium konce pohybu: "Jemné přesné najetí"	
	Platnost:	modální
COARSEA:	Kritérium konce pohybu: "Hrubé přesné najetí"	
	Platnost:	modální
IPOENDA:	Pohybové kritérium: "zastavení interpolátoru"	
	Platnost:	modální
IPOBRKA:	Kritérium přechodu na další blok: Brzdná charakteristika	
	Platnost:	modální
ADISPOSA:	Toleranční okno pro kritérium konce pohybu	
	Platnost:	modální
<osa>:	Název kanálové osy (X, Y,)	
<časový okamžik>:	Časový okamžik přechodu na další blok, vztaženo ke hraně brzdné charakteristiky v %:	
	100% = začátek brzdné charakteristiky 0% = Konec brzdné charakteristiky, to má stejný význam jako příkaz IPOENDA	
	Typ:	REAL
<režim>:	Vztah tolerančního okna	
	Rozsah hodnot:	0 Toleranční okno není aktivní
		1 Toleranční okno vztaženo na požadovanou polohu
		2 Toleranční okno vztaženo na skutečnou polohu
	Typ:	INT
<velikost okna>:	Velikost tolerančního okna	
	Typ:	REAL

Příklady

Příklad 1: Kritérium konce pohybu: "Zastavení interpolátoru"

Programový kód

```
; Najíždění polohovací osou X na pozici 100, rychlost
200 m/min, zrychlení 90%,
; Kritérium konce pohybu: Zastavení interpolátoru
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]

; synchronní akce:
; VŽDY KDYŽ: vstup 1 je nastaven
; POTOM najíždění polohovací osou X na pozici 50, ry-
chlost 200 m/min, zrychlení 140%,
;      Kritérium konce pohybu: Zastavení interpolátoru
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

Příklad 2: Kritérium přechodu na další blok: "Brzdná charakteristika"

Programový kód

Komentář

Programový kód	Komentář
N40 POS[X]=100	; předdefinovaná nastavení jsou platná ; polohovací pohyb osou X na pozici 100
N20 IPOBRKA(X,100)	Kritérium přechodu na další blok: Jemné přesné najetí ; kritérium přechodu na další blok: "Brzdná charakteristika", 100% = začátek brzdné charakteristiky
N30 POS[X]=200	; přechod na další blok se uskuteční, jakmile osa X začne brzdit
N40 POS[X]=250	; Osa X už nebrzdí na pozici 200, nýbrž jede dál na pozici 250. Jakmile osa začne brzdit, následuje přechod na další blok
N50 POS[X]=0	; Osa X brzdí a pak jede zpět na pozici 0. Přechod na další blok se uskutečňuje na pozici 0 a při "jemném přesném najetí".
N60 X10 F100	; Osa X najíždí jako dráhová osa na pozici 10.

Další informace

Systémové proměnné pro kritérium konce pohybu

Kritérium konce pohybu, které je momentálně v platnosti, je možné načíst pomocí systémové proměnné \$AA_MOTEND.

Literatura: /LIS2sl/, Popis seznamů, svazek 2

Kritérium přechodu na další blok: "Brzdná charakteristika" (IPOBRKA)

Jestliže je při aktivování kritéria přechodu na další blok "Brzdná charakteristika" pro volitelný bod pro přechod na další blok naprogramována hodnota, bude tato hodnota v platnosti pro následující polohovací pohyb a synchronně se zpracováním hlavní větve se bude zapisovat

5.11 Programovatelné kritérium konce pohybu (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

do strojního parametru. Pokud pro okamžik přechodu na další blok není udána žádná hodnota, použije se aktuální hodnota z nastavovaného parametru.

SD43600 \$SA_IPOBRAKE_BLOCK_EXCHANGE

Při následujícím naprogramování kritéria konce pohybu osy (FINEA, COARSEA, IPOENDA) se funkce IPOBRKA pro odpovídající osu deaktivuje.

Doplňkové kritérium přechodu na další blok: "Toleranční okno" (ADISPOSA)

Pomocí příkazu ADISPOSA může být jako doplňkové kritérium pro přechod na další blok definováno toleranční okno okolo koncového bodu bloku (lze zvolit, zda je to skutečná poloha nebo požadovaná poloha). Pro přechod na další blok musí být splněny obě podmínky:

- Kritérium přechodu na další blok: "Brzdná charakteristika"
- Kritérium přechodu na další blok: "Toleranční okno"

Literatura

Pokud budete potřebovat další informace týkající se kritéria pro přechod polohovacími osami na další blok, viz:

- Příručka Popis funkcí, Rozšiřovací funkce; Polohovací osy (P2)
- Příručka programování, Základy; kapitola "Regulace posuvu"

Transformace souřadného systému (Frame)

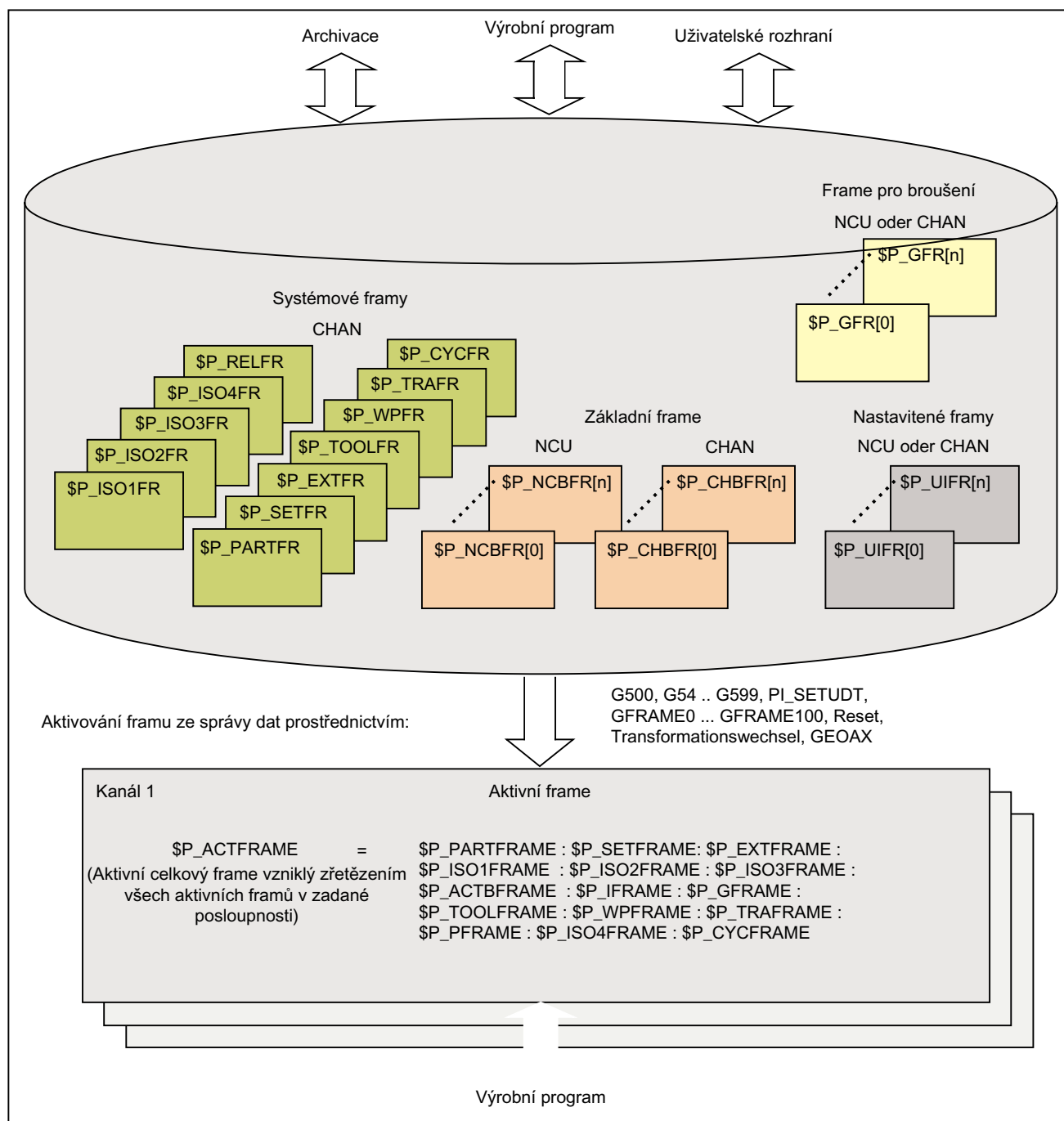
6.1 Transformace souřadnic pomocí proměnné typu frame

Kromě příkazů, které jsou popsány v dokumentu Programovací příručka - Základy, kapitola „Transformace souřadného systému (Frame)“, jako jsou např. příkazy `ROT`, `AROT`, `SCALE` atd., může být souřadný systém obrobku (WCS) transformován také prostřednictvím proměnných typu frame `$P_...FR` (framy správy dat) a `$P_...FRAME` (aktivní framy).

Následující obrázek poskytuje přehled o strukturování proměnných typu frame:

- Frame správy dat
- Aktivní frame
- Aktivní celkový frame: Zřetězení Všech aktivních framů
- Globální framy v NCU
- Kanálové framy

6.1 Transformace souřadnic pomocí proměnné typu frame



Obrázek 6-1 Přehled proměnných typu frame

6.1.1 Předdefinované proměnné typu frame (\$P_CHBFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME)

Aktivní: kanálové základní framy \$P_CHBFRAME[<n>] (\$P_BFRAME)

Poznámka

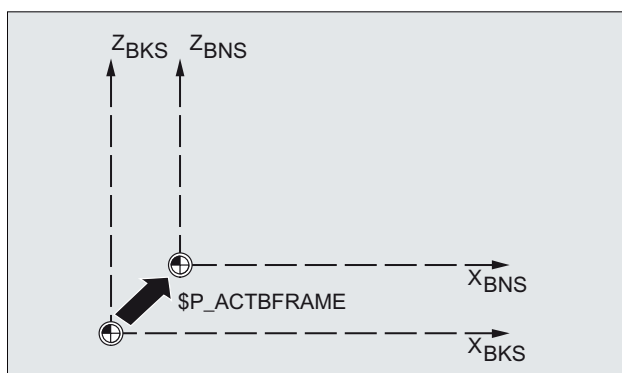
Aktuální základní frame $\$P_BFRAME$ a základní frame správy dat $\$P_UBFR$ jsou z důvodů kompatibility zachovány.

- $\$P_BFRAME \square \$P_CHBFRAME[0]$
- $\$P_UBFR \square \$P_CHBFR[0]$.

Proměnné framy $\$P_CHBFRAME[<n>]$ definují vztah mezi základním souřadným systémem (BCS) a základním souřadným systémem počátku (nuly) (BNS).

Pokud má být aktuální kanálový základní frame $\$P_CHBFRAME[<n>]$ v NC programu okamžitě v platnosti, máte k dispozici následující možnosti.

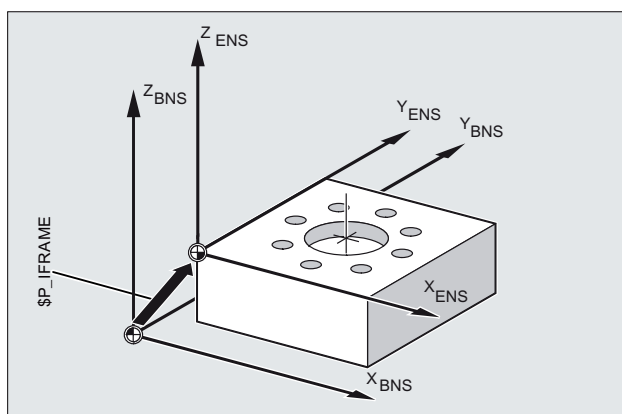
- Příkazy:
 - G500 (deaktivování všech nastavitelných framů, základní frame zůstává aktivní)
 - G54 ... G599 (nastavitelná posunutí počátku)
- Přiřazení základního framu konkrétního kanálu ve správě dat určitému aktuálnímu kanálovému základnímu framu:
 $\$P_CHBFRAME[<n>] = \$P_CHBFR[<m>]$



Aktivní: kanálový nastavitelný frame \$P_IFRAME

Framová proměnná $\$P_IFRAME$, která definuje vztah mezi základním souřadným systémem počátku (nuly) (BNS) a nastavitelným souřadným systémem (ENS).

- $\$P_IFRAME$ odpovídá $\$P_UIFR[\$P_IFRNUM]$
- Proměnná $\$P_IFRAME$ obsahuje po naprogramování např. G54 posunutí, otočení, změny měřítka a zrcadlová převrácení definovaná příkazem G54.

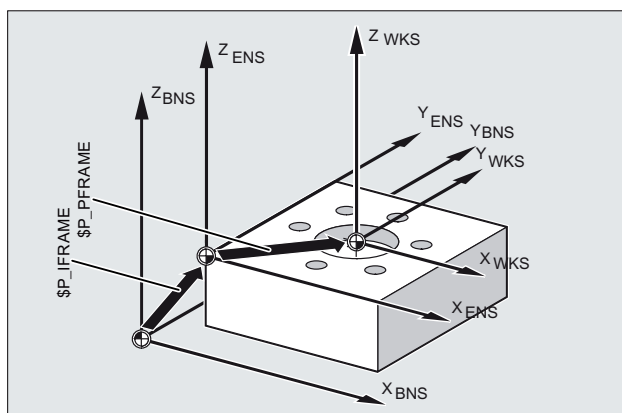


Aktivní: kanálový programovatelný frame \$P_PFRAME

Framová proměnná \$P_PFRAME\$, která definuje vztah mezi nastavitelným souřadným systémem (ENS) a souřadným systémem obrobku (WCS).

Proměnná \$P_PFRAME\$ obsahuje výsledný frame, který vyplývá

- **z naprogramovaných příkazů** TRANS/ATRANS, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR, příp.
- **z přiřazení hodnot pomocí příkazů** CTRANS, CROT, CMIRROR, CSCALE do programovatelného framu



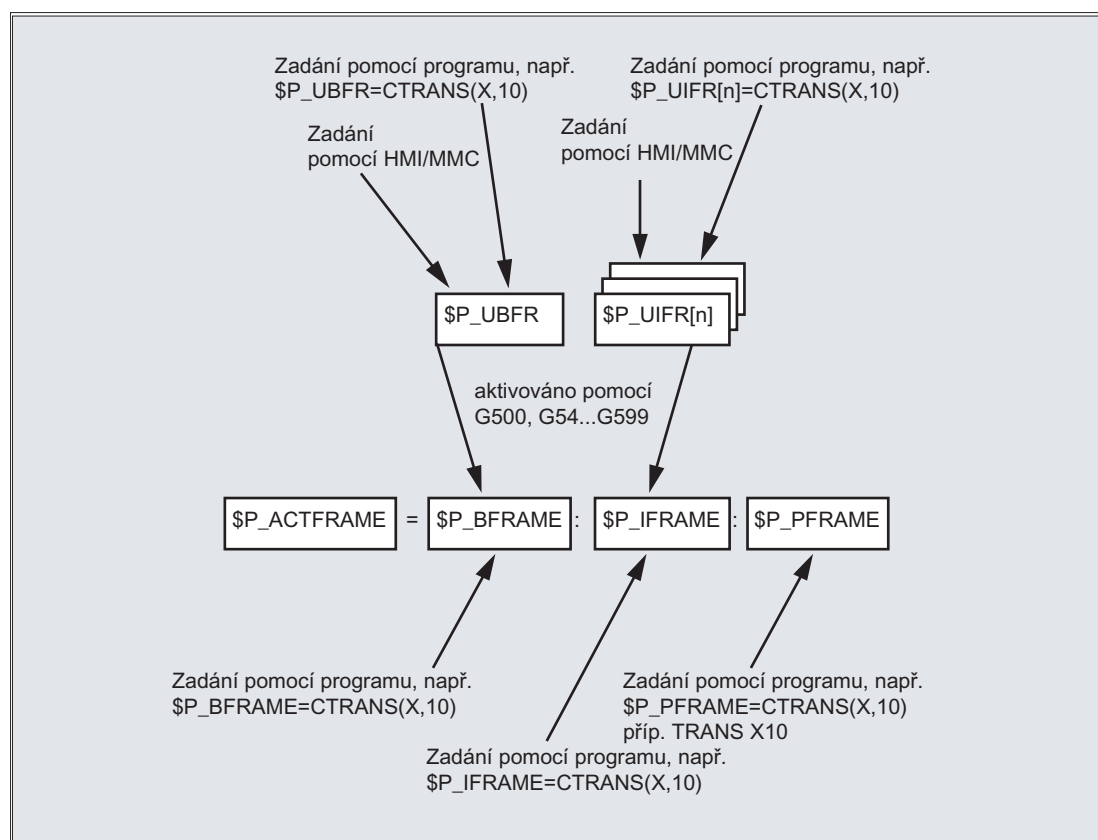
Aktivní: Celkový frame \$P_ACTFRAME

Celkový frame, který je v platnosti v daném kanálu, vyplývá ze zřetězení všech framů, které jsou v daném kanálu v platnosti.

```
$P\_ACTFRAME =    $P\_PARTFRAME : $P\_SETFRAME : $P\_EXTFRAME :
                  $P\_ISO1FRAME : $P\_ISO2FRAME : $P\_ISO3FRAME :
                  $P\_ACTBFRAME : $P\_IFRAME : $P\_GFRAME :
                  $P\_TOOLFRAME : $P\_WPFRAME : $P\_TRAFRAME :
                  $P\_PFRAME : $P\_ISO4FRAME : $P\_CYCFRAME
```

Obrázek 6-2 Proměnná typu FRAME - \$P_ACTFRAME

Jestliže se změní jeden z následujících framů \$P_BFRAME / \$P_CHBFRAME [*n*], \$P_IFRAME oder \$P_PFRAME, aktuální celkový frame \$P_ACTFRAME se nově vypočítá.



Základní frame a nastavitelný frame zůstávají v platnosti i po resetu, pokud je strojný parametr MD 20110 RESET_MODE_MASK nastaven následujícím způsobem:

Bit0=1, Bit14=1 --> \$P UBFR (základní frame) je v platnosti

Bit0=1, Bit5=1 --> \$P_UIFR[\$P_UIFRNUM] (nastavitelný frame) je v platnosti

Správa dat: Kanálový základní frame \$P_CHBFR[<n>]

Prostřednictvím framových proměnných \$P_CHBFR[<n>] je možno číst základní framy ve správě dat a zapisovat do nich. Když jsou do framu ve správě dat zapsána nějaká data, neaktivují se v kanálu okamžitě. Aktivování framu, do něhož byla zapsána data, se uskuteční, když:

- se provede reset kanálu a MD20110 \$MC_RESET_MODE_MASK, bit0 == 1 a bit14 == 1
- Příkaz G500, G54 ... G57, G505 ... G599 (aktivování/deaktivování základního framu s následným novým výpočtem aktuálního celkového framu)

Správa dat: Kanálový nastavitelný frame \$P_UIFR[<n>]

Prostřednictvím framových proměnných \$P_UIFR[<n>] je možno číst nastavitelné framy ve správě dat a zapisovat do nich. Když jsou do framu ve správě dat zapsána nějaká data, neaktivují se v kanálu okamžitě. Započítání framu, do něhož byla v kanálu zapsána data, se uskuteční, když:

- Příkaz G500 (deaktivování všech nastavitelných framů, příp. posunutí počátku)
- Příkaz G54 ... G57, G505 ... G599 (aktivování nastavitelného framu, příp. posunutí počátku)

Aktivní nastavitelný frame	Frame ve správě dat	(odpovídá příkazu)
\$P_IFRAME =	\$P_UIFR[0]	G500
	\$P_UIFR[1]	G54
	\$P_UIFR[2]	G55
	\$P_UIFR[3]	G56
	\$P_UIFR[4]	G57
	\$P_UIFR[5]	G505
	\$P_UIFR[6]	G506
	...	...
	\$P_UIFR[99]	G599

6.2 Přiřazování hodnoty framu

6.2.1 Přímé přiřazení hodnoty (hodnota osy, úhlu, měřítko)

V NC programu můžete dosazovat hodnoty přímo do framů nebo do proměnných typu FRAME.

Syntaxe

Syntaxe

```
$P_PFRAME = CTRANS(X, <hodnota posunutí>, Y, <hodnota posunutí>, Z, <hodnota posunutí>, ...)
```

```
$P_PFRAME = ROT(X, <úhel>, Y, <úhel>, Z, <úhel>, ...)
```

```
$P_UIFR[...] = CROT(X, <úhel>, Y, <úhel>, Z, <úhel>, ...)
```

```
$P_PFRAME = CSCALE(X, <Měřítko>, Y, <Měřítko>, Z, <Měřítko>, ...)
```

```
$P_PFRAME = CMIRROR(X, Y, Z)
```

Syntaxe pro příkaz \$P_CHBFRAME[<n>] je identická jako u příkazu \$P_PFRAME.

Význam

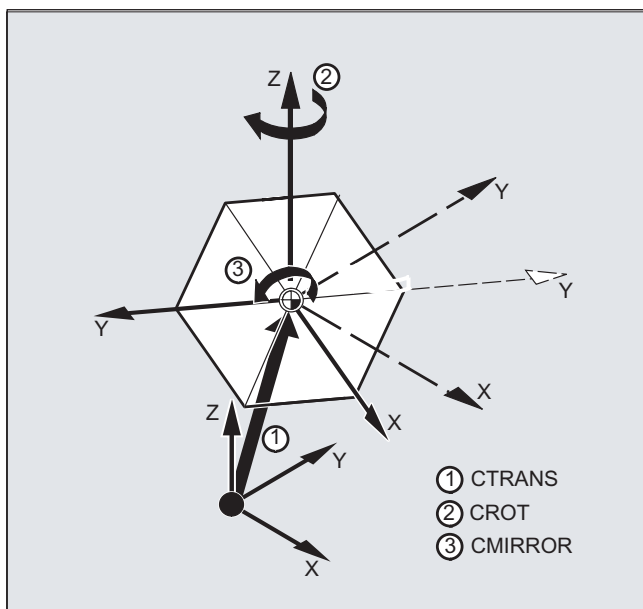
CTTRANS:	Posunutí v uvedených osách
CROT:	Otočení okolo uvedených os
CSCALE:	Změna měřítka v uvedených osách
CMIRROR:	Zrcadlové převrácení pro uvedenou osu
X, Y, Z:	Hodnota posunutí ve směru uvedené geometrické osy
<hodnota posunutí>:	Hodnota posunutí
<úhel>:	Úhel, o který se provádí otočení
<Měřítko>:	Údaj měřítka

Příklady

Přiřazení hodnot složkám aktuálního programovatelného framu

Přiřazení hodnot složkám posunutí, otočení, změny měřítka a zrcadlového převrácení aktuálního programovatelného framu:

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```



Zápis do složek otočení framu

Přiřazení hodnot pro všechny tři osy složkám otočení nastavitelného framu ve správě dat \$P_UIFR pomocí příkazu CROT:

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

Další alternativou je přímé přiřazení jednotlivých hodnot pro příslušnou osu přímo složce otočení ve framu ve správě dat:

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

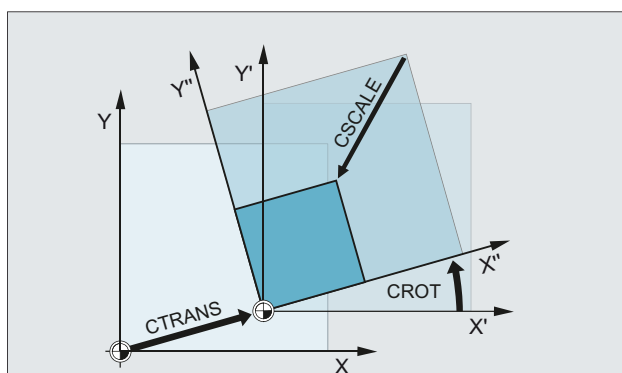
Popis

Větší počet operací na jednom framu je možno zapsat pomocí operátoru zřetězení : a tak je spojit. Operace se potom uskuteční postupně zleva doprava.

Příklad

Zřetěžené operace ve framu \$P_PFRAME s posunutím, otočením a změnou měřítka:

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```

6.2.2 Načítání a editace složek framu (TR, FI, RT, SC, MI)

Pokud potřebujete, můžete mít přístup k **jednotlivým** datům framu, např. k hodnotám pro určité posunutí nebo úhel otočení. Tyto hodnoty můžete změnit nebo je můžete přiřadit jiné proměnné.

Syntaxe

`R10=$P_UIFR[$P_UIFNUM, X, RT]`

Úhel otočení RT okolo osy X z momentálně platného nastavitelného počátku \$P_UIFNUM má být přiřazen proměnné R10.

`R12=$P_UIFR[25, Z, TR]`

Hodnota posunutí TR ve směru osy Z z datového bloku nastavitelného framu č. 25 má být přiřazena proměnné R12.

`R15=$P_PFRAME[Y, TR]`

Hodnota posunutí TR ve směru osy Z z aktuálního programovatelného framu má být přiřazena proměnné R15.

`$P_PFRAME[X, TR] = 25`

Hodnota posunutí TR ve směru osy X z aktuálního programovatelného framu má být změněna. Od tohoto okamžiku platí X25.

Význam

\$P_UIFNUM:	Pomocí této proměnné se automaticky vytváří vztah k momentálně platnému nastavitelnému posunutí počátku.
P_UIFR[n, ..., ...] :	Zadáním čísla framu n se otevírá přístup k nastavitelnému framu č. n.
	Zadání složky, která má být načtena nebo změněna.
TR:	TR - posunutí
FI:	FI - jemné posunutí
RT:	RT - otočení
SC:	SC - změna měřítka

6.2 Přiřazování hodnoty framu

MI:	MI - zrcadlové převrácení
X, Y, Z:	Kromě toho (viz příklady) je zapotřebí zadat odpovídající osy X, Y, Z.

Rozsah hodnot pro otočení RT

Otočení okolo 1. geometrické osy: -180° až $+180^{\circ}$

Otočení okolo 2. geometrické osy: -90° až $+90^{\circ}$

Otočení okolo 3. geometrické osy: -180° až $+180^{\circ}$

Popis**Vyvolání framu**

Prostřednictvím systémové proměnné \$P_UIFRNUM můžete mít přístup přímo k momentálně nastavenému posunutí počátku, které bylo zadáno pomocí příkazů \$P_UIFR příp. G54, G55, ...
(\$P_UIFRNUM obsahuje číslo momentálně nastaveného framu).

Všechny ostatní uložené nastavitelné framy \$P_UIFR vyvoláte prostřednictvím zadání odpovídajícího čísla \$P_UIFR[n].

Pro předem definované framy a vlastní definované framy zadejte příslušné názvy, např. \$P_IFRAME.

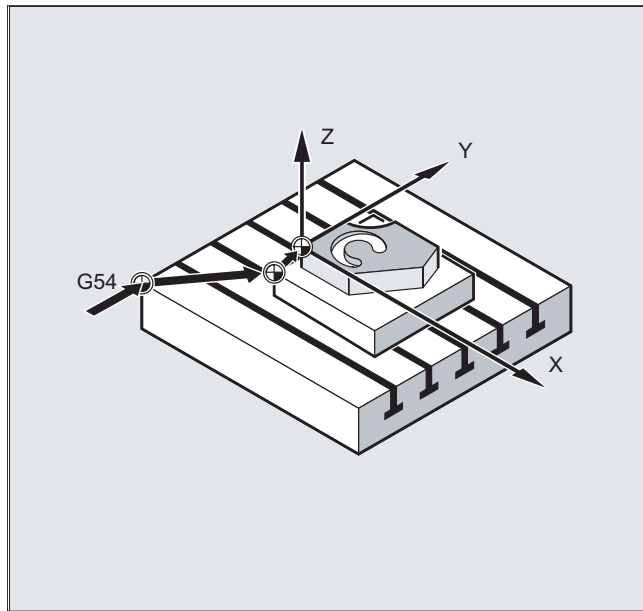
Vyvolávání dat

V hranatých závorkách jsou uvedeny hodnoty názvu osy a složky framu, kterým chcete mít přístup nebo které si přejete změnit, např. [X, RT] nebo [Z, MI].

6.2.3 Výpočty s framy

V NC programu může být frame přiřazen jinému framu nebo framy mohou být vzájemně zřetězovány.

Zřetěžení framů je užitečné např. pro popis většího počtu obrobků, které jsou uspořádány na paletě a které mají být opracovány v rámci jedné výrobní operace.



Za účelem popisu obráběcích úloh na paletách by mohly složky framu obsahovat např. pouze určité dílčí hodnoty, jejichž zřetězením se vygenerují různé počátky souřadného systému obrobku.

Příklady

Přiřazení

Programový kód	Komentář
DEF FRAME EINSTELLUNG_1	; definice lokální proměnné typu frame
EINSTELLUNG_1 = CTRANS(X,10)	; přiřazení výsledku funkce provedené s proměnnou typu frame
\$P_PFRAME = EINSTELLUNG_1	; přiřazení framové proměnné aktuálnímu framu
DEF FRAME EINSTELLUNG_4	; definice lokální proměnné typu frame
EINSTELLUNG_4 = \$P_PFRAME	; uložení aktuálního framu do proměnné typu frame
...	
\$P_PFRAME = EINSTELLUNG_4	; zpětné načtení aktuálního framu z proměnné typu frame

Zřetězení

Framy mohou být prostřednictvím operátoru : v naprogramované posloupnosti spolu zřetězeny. Složky framů, jako např. posunutí, otočení atd., jsou uskutečňovány aditivně jedna po druhé.

Programový kód	Komentář
\$P_IFRAME = \$P_UIFR[15] :	; přiřazení výsledného framu ze zřetězení
\$P_UIFR[16]	; obou nastavitelných framů ze správy dat
	; do nastavitelného celkového framu
	; příklad použití:
	; \$P_UIFR[15]: Posunutí
	; \$P_UIFR[16]: Otočení
\$P_UIFR[3] = \$P_UIFR[4] :	; přiřazení výsledného framu ze zřetězení
\$P_UIFR[5]	; obou nastavitelných framů ze správy dat
	; do jiného nastavitelného framu ve správě dat

6.2.4 Definice proměnných typu frame (DEF FRAME)

Kromě předem definovaných proměnných typu frame můžete definovat také své vlastní framové proměnné. Framové proměnné, které jste sami definovali, jsou uživatelskými proměnnými typu FRAME. Název framu může být libovolný, musí být jen dodržena pravidla pro uživatelské proměnné.

Framovým proměnným, které jste sami definovali, můžete dosazovat hodnoty pomocí funkcí CTRANS, CROT, CSCALE, CMIRROR.

Syntaxe

```
DEF FRAME <název>
```

Význam

DEF FRAME:	Definice uživatelské proměnné typu FRAME.
<název>:	Název framové proměnné

Příklad

Definice proměnné typu FRAME s názvem „PALETTE“ a přiřazení hodnot posunutí a otočení:

Programový kód	Komentář
DEF FRAME PALETTE	; definice framové proměnné PALETTE
PALETTE = CTRANS(...) : CROT(...)	; přiřazení výsledného framu ze zřetězení
	; posunutí a otočení framové proměnné PALETTE

6.3 Hrubé a jemné posunutí (CTrans, CFine)

Jemné posunutí

Jemné posunutí `CFine ( . . . )` se může použít v následujících framech:

- Nastavitelné framy: `$P_UIFR`, příp. `$P_IFRAME`
- Základní framy: `$P_NCBFR[<n>]`, `$P_CHBFR[<n>]`, příp. `$P_CHBFRAMES[<n>]` nebo `$P_ACTBFRAME`
- Programovatelný frame: `$P_PFRAME`

Pro programování jemného posunutí framu se používá příkaz `CFine ( . . . )`.

Hrubé posunutí

Hrubé posunutí `CTrans ( . . . )` se může používat ve všech framech.

Celkové posunutí

Celkové posunutí se skládá z hrubého a z jemného posunutí.

Strojní parametry

Odblokování jemného posunutí

Jemné posunutí je možno odblokovat pomocí strojního parametru:

MD18600 `$MN_MM_FRAME_FINE_TRANS = 1`

Syntaxe

Jemné posunutí

- Celkový frame
 - `<Frame> = CFine (<K_1>, <hodnota>)`
 - `<Frame> = CFine (<K_1>, <hodnota>, <K_2>, <hodnota>)`
 - `<Frame> = CFine (<K_1>, <hodnota>, <K_2>, <hodnota>, <K_3>, <hodnota>)`
- Složky framu
 - `<Frame>[<n>, <K_1>, FI] = <hodnota>`

Hrubé posunutí

- Celkový frame
 - `<Frame> = CTrans (<K_1>, <hodnota>)`
 - `<Frame> = CTrans (<K_1>, <hodnota>, <K_2>, <hodnota>)`
 - `<Frame> = CTrans (<K_1>, <hodnota>, <K_2>, <hodnota>, <K_3>, <hodnota>)`
- Složky framu
 - `<Frame>[<n>, <K_1>, TR] = <hodnota>`

Speciálně pro programovatelný frame \$P_PFRAME:

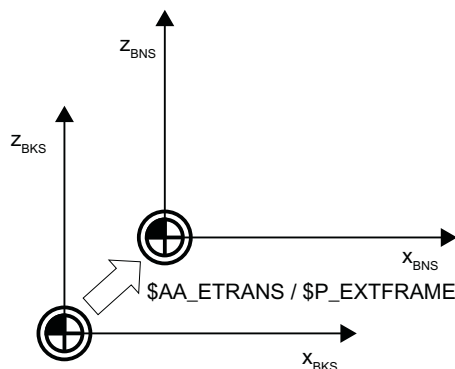
- TRANS <K_1> <hodnota>
- TRANS <K_1> <hodnota><K_2> <hodnota>
- TRANS <K_1> <hodnota><K_2> <hodnota><K_3> <hodnota>

Význam

<Frame>:	Frame, např. Nastavitelný frame ze správy dat \$P_UIFR[<n>]
CFINE:	Jemné posunutí, aditivní posunutí.
CTrans:	Hrubé posunutí, absolutní posunutí
TRANS:	Jen programovatelný frame: Hrubé posunutí, absolutní posunutí
<K_n>:	Osy souřadného systému X, Y, Z
<hodnota>:	Hodnota posunutí

6.4 Externí posunutí počátku (\$AA_ETRANS)

Externí posunutí počátku je lineární posunutí mezi základním souřadným systémem (BCS) a základním systémem počátku (nuly) (BNS).



Externí posunutí počátku pomocí \$AA_ETRANS funguje v závislosti na nastavení strojních parametrů dvěma způsoby:

1. Systémová proměnná \$AA_ETRANS se uplatňuje po aktivování prostřednictvím signálu rozhraní NC/PLC přímo jako hodnota posunutí
2. Hodnota systémové proměnné \$AA_ETRANS se uplatňuje po aktivování prostřednictvím signálu rozhraní NC/PLC v aktivním systémovém framu \$P:EXTFRAME a přebírá se do framu \$P_EXTFR ve správě dat. Potom se nově vypočítá aktivní celkový frame \$P_ACTFRAME.

Strojní parametry

V souvislosti se systémovou proměnnou \$AA_ETRANS je potřeba rozlišovat mezi dvěma postupy, které se vybírají prostřednictvím následujícího strojního parametru:

MD28082 \$MC_MM_SYSTEM_FRAME_MASK, Bit1 = <hodnota>

<hodnota>	Význam
0	Funkce: Přímý zápis \$AA_ETRANS[<osa>] prostřednictvím PLC, HMI nebo NC programu. Uvolnění za účelem posunu realizujícího posunutí počátku \$AA_ETRANS[<osa>] v nejbližším možném bloku posuvu: DB31, ... DBX3.0
1	Funkce: Aktivování aktivního systémového framu \$P:EXTFRAME a framu ve správě dat \$P_EXTFR Uvolnění za účelem posunu realizujícího posunutí počátku \$AA_ETRANS[<osa>] prostřednictvím: DB31, ... DBX3.0. V kanálu se potom stane následující: • Zastavení všech posuvů v kanálu (kromě příkazových a PLC-os) • Zastavení předběžného zpracování s následnou reorganizací (STOPRE) • Hrubé posunutí aktivního framu \$P_EXTFRAME[<osa>] = \$AA_ETRANS[<osa>] • Hrubé posunutí framu ve správě dat \$P_EXTFR[<osa>] = \$AA_ETRANS[<osa>] • Nový výpočet aktivního celkového framu \$P_ACTFRAME • Uskutečnění posunutí v naprogramovaných osách. • Pokračování v přerušeném posunu, příp. ve zpracování NC programu

Programování

- Syntaxe
\$AA_ETRANS[<osa>] = <hodnota>
- Význam

\$AA_ETRANS:	Systémová proměnná pro uložení externího posunutí počátku do vyrovnávací paměti
<osa>:	Kanálová osa
<hodnota>:	Hodnota posunutí

Signál rozhraní NC/PLC

DB31, ... DBX3.0 = 0 → 1 ⇒ \$P_EXTFRAME[<osa>] = \$P_EXTFR[<osa>] = \$AA_ETRANS[<osa>]

6.5 Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod (PRESETON)

Procedura `PRESETON()` nastavuje pro jednu nebo více os novou skutečnou hodnotu v souřadném systému stroje (MCS). To odpovídá posunutí počátku (nuly) MCS osy. Osa se v důsledku toho nepohybuje.

Příkazem `PRESETON` se spouští zastavení předběžného zpracování se synchronizací. Skutečná hodnota je ose přiřazena až v klidovém stavu.

Pokud osa při příkazu `PRESETON` není tomuto kanálu přiřazena, závisí další postup na specifické konfiguraci při výměně os:

MD30552 \$MA_AUTO_GET_TYPE

Stav najíždění na referenční bod:

Prostřednictvím nastavení nové skutečné hodnoty v souřadném systému stroje se uskuteční reset stavu najetí strojní osou na referenční bod:

DB31, ... DBX60.4 / .5 = 0 (najeto na referenční bod / měřicí systém 1 / 2 synchronizován)

Proto se doporučuje používat příkaz `PRESETON` jen v případech os, u kterých není povinnost najíždět na referenční bod.

Jestliže si přejete opět obnovit původní souřadný systém stroje, je nutné z výrobního programu (G74) příslušnou strojní osou znovu najet na referenční bod, aby se synchronizoval měřicí systém.



POZOR

Ztráta stavu najetí na referenční bod

Prostřednictvím nastavení nové skutečné hodnoty v souřadném systému stroje pomocí příkazu `PRESETON` se uskuteční reset stavu najetí strojní osou na referenční bod, který pak je „není najeto na referenční bod/není synchronizováno“.

Programování

Syntaxe

`PRESETON(<osa_1>, <hodnota_1> [, <osa_2>, <hodnota_2>, ... <osa_8>, <hodnota_8>])`

Význam

PRESETON:	Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod	
	Zastavení předběžného zpracování:	ano
	Samostatný blok:	ano
<osa_x>:	Název osy stroje	
	Typ:	AXIS
	Rozsah hodnot:	identifikátory strojních os definovaných v kanálu

<hodnota_x>:	Nová skutečná hodnota strojní osy v souřadném systému stroje (MCS) Zadání se uskutečňuje v momentálně platném systému jednotek (palce / metrické jednotky). Aktivní programování průměrů (DIAMON) je zohledňováno.	
	Typ:	REAL

Literatura

Příkazy PRESETON v NC-programech

Pokud budete potřebovat podrobný popis věnovaný příkazu `PRESETON` v NC programech, viz:

Příručka k funkcím, Základní funkce, kapitola „K2: Osy, souřadné systémy, framy“ > „Souřadné systémy“ > „Souřadný systém stroje (MCS)“ > „Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod (PRESETON)“

Příkazy PRESETONS v synchronních akcích

Pokud budete potřebovat podrobný popis věnovaný příkazu `PRESETON` v synchronních akcích, viz:

Příručka k funkcím, Synchronní akce; kapitola: „Podrobný popis“ > „Akce v synchronních akcích“ > „Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod (PRESETON)“

6.6 Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod (PRESETONS)

Procedura `PRESETONS()` nastavuje pro jednu nebo více os novou skutečnou hodnotu v souřadném systému stroje (MCS). To odpovídá posunutí počátku (nuly) MCS osy. Osa se v důsledku toho nepohybuje.

Příkazem `PRESETONS` se spouští zastavení předběžného zpracování se synchronizací. Skutečná hodnota je ose přiřazena až v klidovém stavu.

Pokud osa při příkazu `PRESETONS` není tomuto kanálu přiřazena, závisí další postup na specifické konfiguraci při výměně os:

MD30552 \$MA_AUTO_GET_TYPE

Stav najíždění na referenční bod:

Prostřednictvím nastavení nové skutečné hodnoty v souřadném systému stroje (MCS) pomocí příkazu `PRESETONS` se stav najetí strojní osou na referenční bod **nezmění**.

Předpoklady

- **Typ snímače**

Příkaz `PRESETONS` je možný pouze v případě následujících typů snímačů aktivního měřicího systému:

- MD30240 \$MA_ENC_TYPE[<Měřicí systém>] = 0 (simulovaný snímač)
- MD30240 \$MA_ENC_TYPE[<Měřicí systém>] = 1 (snímač vysílající surový signál)

- **Režim najíždění na referenční bod**

Příkaz `PRESETONS` je možný pouze v případě následujících režimů najíždění na referenční bod aktivního měřicího systému:

- MD34200 \$MA_ENC_REFP_MODE[<měřicí systém>] = 0 (najíždění na referenční bod není možné)
- MD34200 \$MA_ENC_REFP_MODE[<měřicí systém>] = 1 (najíždění na referenční bod inkrementálních, otočných nebo lineárních měřicích systémů: Nulový impuls na výstupu snímače)

Programování

Syntaxe

`PRESETONS(<osa_1>, <hodnota_1> [, <osa_2>, <hodnota_2>, ... <osa_8>, <hodnota_8>])`

Význam

PRESETONS:	Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod	
	Zastavení předběžného zpracování:	ano
	Samostatný blok:	ano
<osa_x>:	Název osy stroje	
	Typ:	AXIS
	Rozsah hodnot:	identifikátory strojních os definovaných v kanálu

<hodnota_x>:	Nová aktuální skutečná hodnota strojní osy v souřadném systému stroje (MCS) Zadání se uskutečňuje v aktivním systému jednotek (palce / metrické jednotky). Aktivní programování průměrů (DIAMON) je zohledňováno.	
	Typ:	REAL

Literatura

Příkazy PRESETON v NC-programech

Pokud budete potřebovat podrobný popis věnovaný příkazu PRESETONS v NC programech, viz:

Příručka k funkcím, Základní funkce, kapitola „K2: Osy, souřadné systémy, framy“ > „Souřadné systémy“ > „Souřadný systém stroje (MCS)“ > „Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod (PRESETONS)“

Příkazy PRESETONS v synchronních akcích

Pokud budete potřebovat podrobný popis věnovaný příkazu PRESETONS v synchronních akcích, viz:

Příručka k funkcím, Synchronní akce; kapitola: „Podrobný popis“ > „Akce v synchronních akcích“ > „Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod (PRESETONS)“

6.7 Výpočet framu na základě 3 změřených bodů v prostoru (MEAFRAME)

Funkce MEAFRAME se používá pro podporu v měřicích cyklech. Vypočítává frame na základě tří ideálních a vzájemně korespondujících změřených bodů.

Když je obrobek nastaven do polohy pro obrábění, je jeho pozice vzhledem ke kartézskému souřadnému systému stroje vůči jeho ideální poloze zpravidla jednak posunutá, ale i pootočená. Pro přesné opracování nebo měření je zapotřebí uskutečnit buď nákladné fyzické nastavování polohy nebo změny pohybů ve výrobním programu.

Frame může být definován nasnímáním trojice bodů v prostoru, jejichž ideální pozice je známa. Pro snímání polohy se používají buď dotykové nebo optické snímače, speciální na nosné destičce přesně fixované díry nebo měřicí kuličky.

Syntaxe

MEAFRAME(<ideální body>,<měřené body>,<kvalita>)

Význam

MEAFRAME:	Volání funkce		
<ideální body>:	2-rozměrné pole typu REAL, které obsahuje tři souřadnice ideálních bodů.		
<měřené body>:	2-rozměrné pole typu REAL, které obsahuje tři souřadnice měřených bodů.		
<kvalita>:	Proměnná, pomocí které systém vrací informace o kvalitě výpočtu proměnné typu FRAME.		
	Typ:	VAR REAL	
	Hodnota:	-1	Ideální body leží příliš blízko jedné přímky: Frame nemohl být vypočítán. Výsledná proměnná typu FRAME obsahuje neutrální frame.
		-2	Změřené body leží příliš blízko jedné přímky: Frame nemohl být vypočítán. Výsledná proměnná typu FRAME obsahuje neutrální frame.
		-4	Výpočet rotační matice skončil z nějakého jiného důvodu neúspěchem.
		≥ 0.0	Součet deformací (vzdálenosti mezi body), které jsou zapotřebí pro převedení změřeného trojúhelníku na trojúhelník, který se blíží ideálnímu.

Poznámka**Jakost měření**

Aby změřené souřadnice bylo možné pomocí kombinace otočení/posunutí přiřadit ideálním souřadnicím, musí se trojúhelník tvořený změřenými body co možno nejvíce shodovat s ideálním trojúhelníkem. To bude zajištěno kompenzačním algoritmem, který se snaží minimalizovat součet druhých mocnin odchylek, kterými se změřený trojúhelník převádí na ideální trojúhelník.

Případně potřebná deformace poloh měřených bodů může posloužit jako ukazatel jakosti měření a z tohoto důvodu se ukládá jako doplňková proměnná funkce MEAFRAME.

Poznámka

Frame vytvořený pomocí funkce MEAFRAME může být pomocí funkce ADDFRAME transformován do jiného framu v řetězci framů (viz příklad "Zřetězení s příkazem ADDFRAME").

Příklady**Příklad 1:**

Výrobní program 1:

Programový kód

```
...
DEF FRAME CORR_FRAME
```

Nastavení měřicích bodů:

Programový kód**Komentář**

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]=	; na zkoušku
SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8)	
DEF REAL FIT_QUALITY=0	
DEF REAL ROT_FRAME_LIMIT=5	; povolené pootočení polohy součásti max. 5 stupňů
DEF REAL FIT_QUALITY_LIMIT=3	; povolené posunutí mezi ideálním a změřeným trojúhelníkem max. 3 mm
DEF REAL SHOW_MCS_POS1[3]	
DEF REAL SHOW_MCS_POS2[3]	
DEF REAL SHOW_MCS_POS3[3]	

6.7 Výpočet framu na základě 3 změřených bodů v prostoru (MEAFRAME)

Programový kód	Komentář
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)	
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. úhlu RPY
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; omezení 2. úhlu RPY
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; omezení 3. úhlu RPY
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 \$P_IFRAME=CORR_FRAME	; aktivování nasnímaného framu s nastavitelným framem ; kontrola framu najetím geometric- kými osami na ideální body
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 fra- mu nulovým framem (předem není do- sazena žádná hodnota)

Programový kód	Komentář
No_FRAME	; deaktivování nastavitelného framu nulovým framem (předem není dosazena žádná hodnota)
M0	
M30	

Příklad 2: Zřetězení framů

Zřetězení framů MEAFRAME za účelem korekce

Výsledkem funkce MEAFRAME je korekční frame. Pokud je tento korekční frame zřetězen s nastavitelným framem \$P_UIFR[1], který byl aktivní při vyvolání funkce (např. G54), je takto získán nastavitelný frame pro další přepočítávání pro potřeby posuvů nebo obrábění.

Zřetězení s příkazem ADDFRAME

Pokud se má tento korekční frame uplatnit v řetězci framů na nějakém jiném místě nebo pokud jsou před nastavitelným framem aktivní ještě nějaké jiné framy, potom je možné pro začlenění tohoto framu do řetězce v základním framu kanálu nebo v systémovém framu použít funkci ADDFRAME.

Ve framu přitom nesmí být aktivní:

- Zrcadlové převrácení s příkazem MIRROR
- Změna měřítka s příkazem SCALE

Vstupními parametry pro požadovanou a skutečnou hodnotu jsou souřadnice obrobku. V základním systému řídicího systému je potřeba tyto souřadnice vždy udávat v metrických jednotkách nebo v palcích (G71/G70) a jako rozměr vztažený k rádiusu (DIAMOF).

Literatura:

Pokud budete potřebovat další informace k příkazu ADDFRAME, viz:
Příručka k funkcím, Základní funkce; (K2): Osy, souřadné systémy, framy

6.8 Globální framy v NCU

Pro každou NCU existuje jen jeden globální frame platný pro všechny kanály. Globální framy v NCU mohou být ze všech kanálů čteny a všechny kanály do nich mohou také zapisovat. Aktivování globálního framu v NCU se uskutečňuje v příslušném kanálu.

Prostřednictvím globálního framu mohou být pro **kanálové osy a osy stroje** realizována posunutí, změny měřítka a zrcadlová převrácení.

Geometrické souvislosti a řetězce framů

V případě globálních framů neexistuje žádná geometrická souvislost mezi osami. Z tohoto důvodu není možné provádět otočení nebo používat programování identifikátorů geometrických os.

- Na globální framy se nedají používat žádná otočení. Naprogramování otočení je odmítnuto a aktivuje se přitom alarm: "18310 Kanál %1 blok %2 Frame: Otočení nepřipustné".
- Zřetězení globálních framů a kanálových framů je možné. Výsledný frame obsahuje všechny složky framu včetně otočení pro všechny osy. Přiřazení framu se složkami otočení globálnímu gramu je odmítnuto, přičemž se aktivuje alarm "Frame: Otočení nepřipustné".

Globální framy v NCU

Globální základní frame v NCU \$P_NCBFR[n]

V konfiguraci může být nastaveno až 8 globálních základních framů NCU.

Současně mohou být k dispozici kanálové základní framy.

Globální framy mohou být ze všech kanálů NCU čteny a všechny kanály NCU do nich mohou také zapisovat. Při zápisu do globálního framu se musí uživatel postarat o koordinaci kanálů. To může být realizováno např. značkami pro čekání (WAITMC).

Výrobce stroje

Počet globálních základních framů se nastavuje v konfiguraci pomocí strojního parametru.

Literatura:

Příručka k funkcím, Základní funkce; Osy, souřadné systémy, framy (K2)

V NCU globální nastavitelné framy \$P_UIFR[n]

Všechny nastavitelné framy G500, G54...G599 mohou být konfigurovány buď jako globální v NCU nebo jako kanálové.

Výrobce stroje

Prostřednictvím strojního parametru MD18601 \$MN_MM_NUM_GLOBAL_USER_FRAMES mohou být všechny nastavitelné framy překonfigurovány na globální framy.

Jako identifikátory os se u příkazů pro programování framů mohou používat identifikátory kanálových os a identifikátory os stroje. Programování geometrických os je odmítnuto, přičemž se aktivuje alarm.

6.8.1 Specifické kanálové framy (\$P_CHBFR, \$P_UBFR)

Do nastavitelných framů nebo základních framů je možné zapisovat nebo je číst pomocí výrobního programu nebo pomocí BTSS prostřednictvím ovládacího panelu a PLC.

Pro globální framy je možné také jemné posunutí. Potlačení globálních framů se uskutečňuje stejně jako u kanálových framů, tedy pomocí příkazů G53, G153, SUPA a G500.

Výrobce stroje

Pomocí strojního parametru MD28081 \$MC_MM_NUM_BASE_FRAMES může být v konfiguraci nastaven počet základních framů v kanálu. Standardní konfigurace je nastavena tak, aby v každém kanálu existoval nejméně jeden základní frame. V každém kanálu je možných maximálně 8 základních framů. Kromě 8 základních framů může v kanálu existovat ještě dalších 8 globálních základních framů v NCU.

Kanálové framy

\$P_CHBFR[n]

Pomocí systémové proměnné \$P_CHBFR[n] je možné základní framy číst a zapisovat do nich. Při zápisu do základního framu není zřetězený celkový základní frame ještě aktivován; toto aktivování se uskuteční teprve se zpracováním některého z příkazů G500, G54 ... G599. Tato proměnná slouží především jako paměť pro operace zápisu do základního framu z HMI nebo z PLC. Tyto proměnné typu FRAME se ukládají prostřednictvím zálohování dat.

První základní frame v kanálu

Zápis do předem definované proměnné \$P_UBFR neaktivuje základní frame s indexem pole 0 v daný okamžik; toto aktivování se uskuteční teprve se zpracováním některého z příkazů G500, G54 ... G599. Do této proměnné lze zapisovat také v programu a lze ji tam i číst.

\$P_UBFR

Proměnná \$P_UBFR je identická s proměnnou \$P_CHBFR[0]. Standardně existuje v kanálu vždy jeden základní frame, takže systémová proměnná je kompatibilní i se staršími verzemi. Pokud žádný kanálový základní frame neexistuje, při zápisu nebo při čtení se aktivuje alarm: "Frame: Nepřípustný příkaz".

6.8.2 Framy aktivní v kanálu

V kanálu platné framy jsou ve výrobním programu zadávány prostřednictvím příslušných systémových proměnných těchto framů. K tomu patří i systémové framy. Pomocí těchto systémových proměnných je možné ve výrobním programu načíst aktuální systémový frame a lze do něj i zapisovat.

Aktuální framy aktivní v kanálu

Přehled

Aktuální systémové framy
\$P_PARTFRAME

pro:
TCARR a PAROT

\$P_SETFRAME	Dosazení skutečné hodnoty a škrábnutí
\$P_EXTFRAME	Externí posunutí počátku
\$P_NCBFRAME[n]	Aktuální globální základní framy NCU
\$P_CHBFRAME[n]	Aktuální kanálové základní framy
\$P_BFRAME	Aktuální 1. základní frame v kanálu
\$P_ACTBFRAME	Celkový základní frame
\$P_CHBFRMASK a \$P_NCBFRMASK	Celkový základní frame
\$P_IFFRAME	Aktuální nastavitelný frame
Aktuální systémové framy	pro:
\$P_TOOLFRAME	TOROT a TOFRAME
\$P_WPFRAME	Vztažné body obrobku
\$P_TRAFRAME	Transformace
\$P_PFRAME	Aktuální programovatelný frame
Aktuální systémový frame	pro:
\$P_CYCFRAME	Cykly
P_ACTFRAME	Aktuální celkový frame
Zřetězení framů	Aktuální frame se skládá z celkového základního framů

\$P_NCBFRAME[n] - Aktuální globální základní framy NCU

Pomocí systémové proměnné **\$P_NCBFRAME[n]** je možné prvky pole aktuálního globálního základního framů číst a zapisovat do nich. Výsledný celkový základní frame se prostřednictvím operace zápisu v kanálu započítá.

Změněný frame se aktivuje pouze v kanálu, v němž byl frame naprogramován. Jestliže má být frame změněn pro všechny kanály dané NCU, musí být data zapsána současně do obou proměnných **\$P_NCBFR[n]** a **\$P_NCBFRAME[n]**. Ostatní kanály musí ještě frame aktivovat, např. příkazem G54. Při zápisu základního framů se celkový základní frame nově vypočítá.

\$P_CHBFRAME[n] - Aktuální kanálové základní framy

Pomocí systémové proměnné **\$P_CHBFRAME[n]** je možné prvky pole aktuálního kanálového základního framů číst a zapisovat do nich. Výsledný celkový základní frame se prostřednictvím operace zápisu v kanálu započítá. Při zápisu základního framů se celkový základní frame nově vypočítá.

\$P_BFRAME - Aktuální 1. základní frame v kanálu

Prostřednictvím předem definované proměnné **\$P_BFRAME** typu frame je možné ve výrobním programu číst aktuální základní frame s indexem 0, který je v kanálu platný, a lze do něj i zapisovat. Zapsaný základní frame se okamžitě započítá.

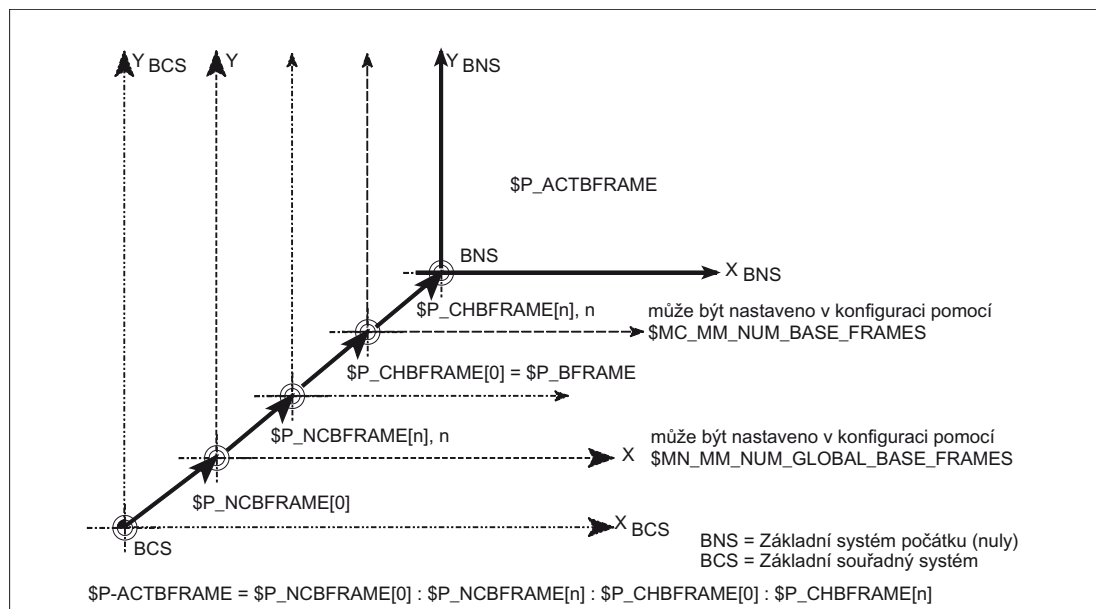
Proměnná **\$P_BFRAME** je identická s proměnnou **\$P_BFRAME [0]**. Tato systémová proměnná má standardně vždy platnou hodnotu. Pokud žádný kanálový základní frame neexistuje, při zápisu nebo při čtení se aktivuje alarm: "Frame: Nepřípustný příkaz".

\$P_ACTFRAME - celkový základní frame

Proměnná \$P_ACTFRAME zjišťuje zřetězený celkový základní frame. Tato proměnná je jen pro čtení.

Proměnná \$P_ACTFRAME odpovídá:

$\$P_NCBFRAME[0] : \dots : \$P_NCBFRAME[n] : \$P_CHBFRAME[0] : \dots : \$P_CHBFRAME[n]$.

**\$P_CHBFMASK a \$P_NCBFMASK - celkový základní frame**

Pomocí systémových proměnných \$P_CHBFMASK a \$P_NCBFMASK může uživatel zvolit, které základní framy by chtěl zahrnout do výpočtu "celkového" základního framu. Proměnné mohou být naprogramovány jen v programu a mohou být čteny pomocí BTSS. Hodnota proměnné je interpretována jako bitová maska a udává, který prvek pole základního framu z proměnné \$P_ACTFRAME je vtažen do výpočtu.

Pomocí parametrů \$P_CHBFMASK (kanálové základní framy) a \$P_NCBFMASK (základní framy globální v NCU) může být zadáno, které framy se započítávají.

Naprogramováním těchto proměnných se celkový základní frame a celkový frame znovu vypočítají. Po resetu a v základním nastavení jsou hodnoty \$P_CHBFMASK a \$P_NCBFMASK následující:

$\$P_CHBFMASK = \$MC_CHBFRAME_RESET_MASK$

$\$P_NCBFMASK = \$MC_CHBFRAME_RESET_MASK$

Příklad:

$\$P_NCBFMASK = 'H81' ; \$P_NCBFRAME[0] : \$P_NCBFRAME[7]$

$\$P_CHBFMASK = 'H11' ; \$P_CHBFRAME[0] : \$P_CHBFRAME[4]$

\$P_IFRAME - Aktuální nastavitelný frame

Prostřednictvím předem definované proměnné \$P_IFRAME typu frame je možné ve výrobním programu číst aktuální nastavitelný frame, který je v kanálu platný, a lze do něj i zapisovat. Zapsaný nastavitelný frame se okamžitě započítá.

V případě v NCU globálních nastavitelných framů se změněný frame se uplatňuje pouze v kanálu, v němž byl frame naprogramován. Jestliže má být frame změněn pro všechny kanály dané NCU, musí být data zapsána současně do obou proměnných \$P_UIFR[n] a \$P_IFRAME. Ostatní kanály musí ještě odpovídající frame aktivovat, např. příkazem G54.

\$P_PFRAME - Aktuální programovatelný frame

\$P_PFRAME je programovatelný frame, který vyplývá z programových příkazů TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR, příp. z přiřazení CTRANS, CROT, CMIRROR, CSCALE do programovatelné proměnné FRAME.

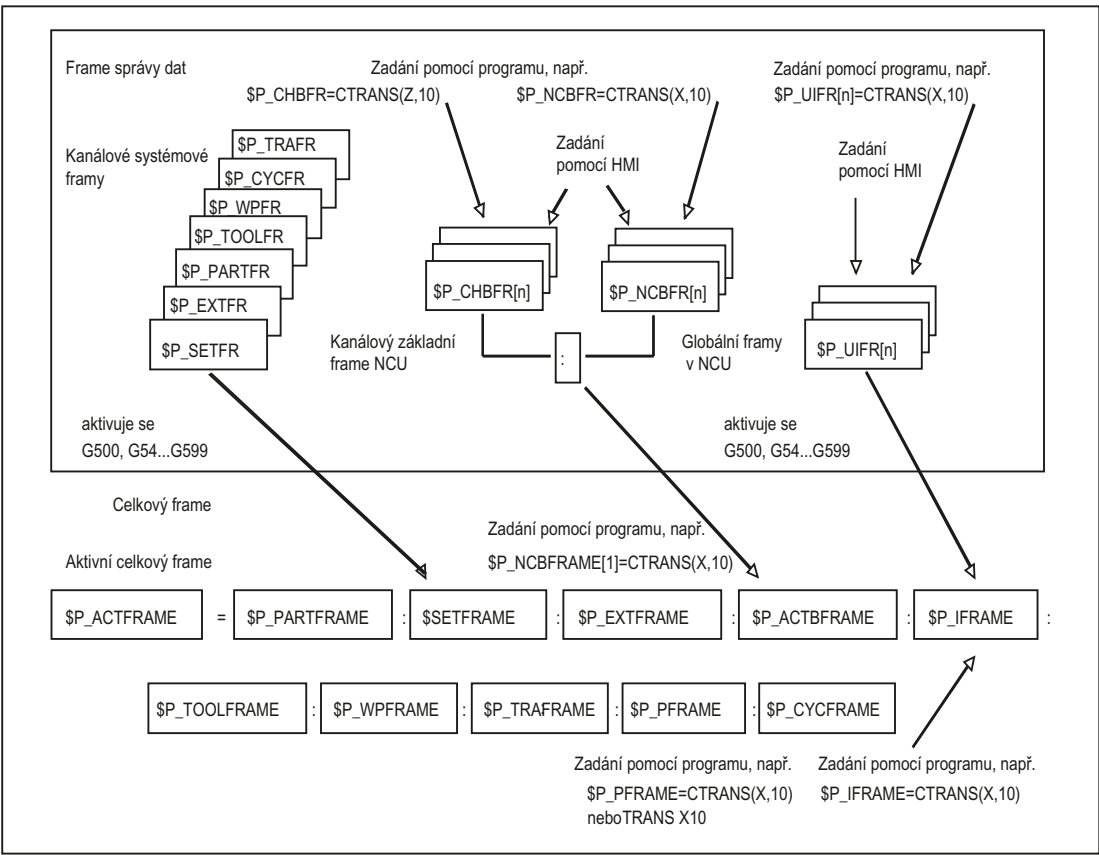
Proměnná aktuálního programovatelného framu, která definuje vztah mezi nastavitelným souřadným systémem (ENS) a souřadným systémem obrobku (WCS).

P_ACTFRAME - Aktuální celkový frame

Aktuální výsledný celkový frame \$P_ACTFRAME nyní vzniká jako zřetězení všech základních framů. aktuálního nastavitelného framu a programovatelného framu. Aktuální frame se aktualizuje, kdykoli se změní některá ze složek tohoto framu.

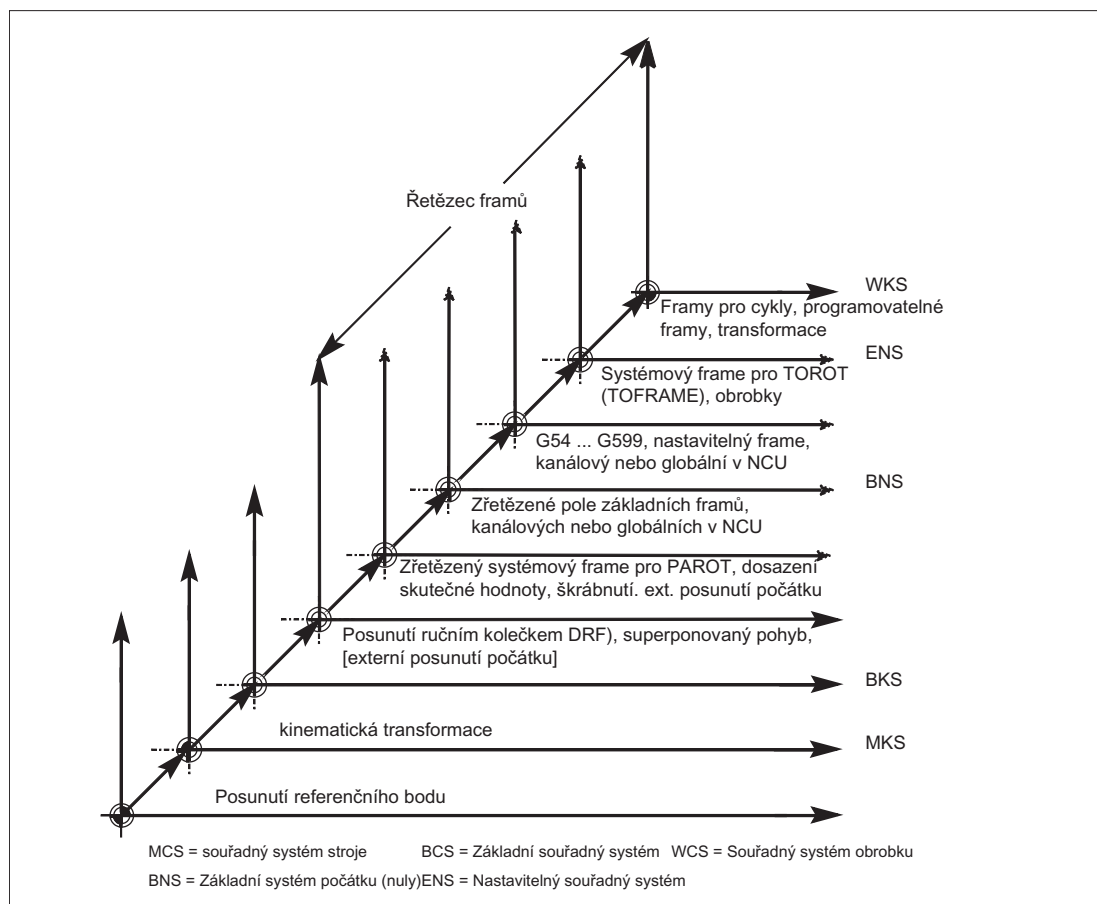
Proměnná \$P_ACTFRAME odpovídá:

```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :  
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```



Zřetězení framů

Aktuální frame se skládá z celkového základního framu, nastavitelného framu, systémového framu a z programovatelného framu podle výše uvedeného aktuálního celkového framu.



Transformace

7.1 Všeobecné programování různých druhů transformace

Všeobecná funkce

Aby bylo možné řídicí systém přizpůsobit různým kinematikám stroje, je možné naprogramovat celou řadu různých transformací s vhodnými parametry. Pomocí těchto parametrů je možno pro vybranou transformaci odpovídajícím způsobem nastavit jak orientaci nástroje v prostoru, tak také orientační pohyby kruhových os.

V případě transformací tří, čtyř a pěti os se naprogramované údaje polohy vždy vztahují na špičku nástroje, který je nastavován tak, aby byl vždy ortogonálně vůči obráběné ploše nacházející se v prostoru. Kartézské souřadnice jsou ze základního souřadného systému přepočítány do souřadného systému stroje a jsou vztaženy na geometrické osy. Tyto souřadnice popisují pracovní bod. Virtuální kruhové osy popisují orientaci nástroje v prostoru a programují se pomocí funkce TRAORI.

V případě kinematické transformace mohou být programovány polohy v kartézském souřadném systému. Řídicí systém transformuje posuvové pohyby naprogramované pomocí příkazů TRANSMIT, TRACYL a TRAANG v kartézském souřadném systému na posuvové pohyby reálných os stroje.

Programování

Transformace ve třech, čtyřech a pěti osách (TRAORI)

Deklarovaná transformace orientace se aktivuje příkazem TRAORI se třemi možnými parametry pro číslo transformace, vektor orientace a offset kruhové osy.

TRAORI(číslo transformace, vektor orientace, offset kruhové osy)

Kinematické transformace

Ke kinematickým transformacím patří deklarované transformace TRANSMIT(číslo transformace)

TRACYL(pracovní průměr, číslo transformace)

TRAANG(úhel šikmo postavené osy, číslo transformace)

Deaktivování aktivní transformace

Pomocí příkazu TRAFOOF mohou být momentálně aktivní transformace deaktivovány.

Transformace orientace

Transformace ve třech, čtyřech a pěti osách (TRAORI)

Za účelem optimálního opracování prostorově tvarovaných ploch v pracovním prostoru stroje potřebují obráběcí stroje kromě tří lineárních os X, Y a Z ještě další osy. Tyto doplňkové osy

popisují orientaci v prostoru a jsou proto nazývány orientačními osami. U mnoha typů strojů s různou kinematikou jsou k dispozici jako kruhové osy.

1. Naklápěcí hlavička se dvěma osami, např. kardanová nástrojová hlavička s kruhovou osou rovnoběžně s lineární osou s pevným nástrojovým stolem.
2. Otočný stůl se dvěma osami, např. pevná naklápěcí hlavička s otočným nástrojovým stolem okolo dvou os.
3. Naklápěcí hlavička s jednou osou a otočný stůl s jednou osou, např. otočná naklápěcí hlavička s otočeným nástrojem a s otočným nástrojovým stolem okolo jedné osy.
4. Naklápěcí hlavička se dvěma osami a otočný stůl s jednou osou, např. s otočným nástrojovým stolem okolo jedné osy a s otočnou naklápěcí hlavičkou s nástrojem, který se může otáčet okolo sebe sama.

3- a 4-osá transformace jsou zvláštní případy 5-osé transformace a programují se analogicky k této 5-osé transformaci.

"**Generická 3-/4-/5-/6-osá transformace** pokrývá svou funkcí pravoúhle uspořádané kruhové osy, stejně jako transformace pro kardanovou frézovací hlavičku a může být aktivována příkazem TRAORI stejně jako jakákoli jiná transformace orientace i pro tyto typy strojů se čtyřmi osami. V případě generické 5-/6-osé transformace má orientace nástroje ještě další třetí stupeň volnosti, který umožňuje, že se nástroj nacházející se v libovolném směru v prostoru může ještě otáčet okolo své vlastní osy.

Literatura: /FB3/ Příručka k funkcím, Speciální funkce; 3- až 5-osá transformace (F2)

Základní nastavení orientace nástroje nezávislé na kinematice

ORIRESET

Pokud je aktivní transformace orientace TRAORI, potom je možné pomocí příkazu ORIRESET zadat základní polohy až 3 orientačních os s volitelnými parametry A, B, C. Přiřazení posloupnosti naprogramovaných parametrů kruhovým osám se uskutečňuje podle posloupnosti orientačních os definované prostřednictvím transformace. Naprogramování příkazu ORIRESET(A, B, C) způsobí, že orientační osy lineárně a synchronně najedou ze své momentální polohy do zadané základní polohy.

Kinematické transformace

TRANSMIT a TRACYL

V případě frézovacích prací na soustruzích mohou být naprogramovány pro deklarované transformace buď

1. opracování na čelní ploše při upnutí pro soustružení (TRANSMIT) nebo
2. obrábění drážek libovolného průběhu na válcových tělesech pomocí

transformace TARCYL.

TRAANG

Pokud má být přísuvná osa např. pro technologii broušení přisouvána také šikmo, může být pomocí transformace TRAANG pro deklarovanou transformaci naprogramován příslušný úhlový parametr.

Posuv PTP v kartézských souřadnicích

Ke kinematickým transformacím patří také "Posuv PTP" v kartézských souřadnicích", při kterém může být naprogramováno až 8 různých kloubových nastavení STAT=. Polohy jsou naprogramovány v kartézském souřadném systému, přičemž pohyb se uskutečňuje v souřadném systému stroje.

Literatura:

/FB2/, Příručka k funkcím, Rozšiřovací funkce; Kinematická transformace (M1)

Zřetězené transformace

Mohou být za sebou poskládány vždy maximálně dvě transformace. U takto zřetězené druhé transformace se pohybové složky os přebírají z první transformace.

Jako první transformaci je možno zadat následující:

- Transformace orientace TRAORI
- Polární transformace TRANSMIT
- Transformace válcového pláště TRACYL
- Transformace Šikmá osa TRAANG

Druhou transformací musí být Šikmá osa TRAANG.

7.1.1 Pohyby provádějící změnu orientace při transformacích**Posuvové pohyby a orientační pohyby**

Posuvové pohyby programovatelných orientací závisí především na typu stroje. V případě tří-, čtyř- a pětiosé transformace s funkcí TRAORI popisují rotační osy nebo naklápěcí lineární osy orientační pohyby nástroje.

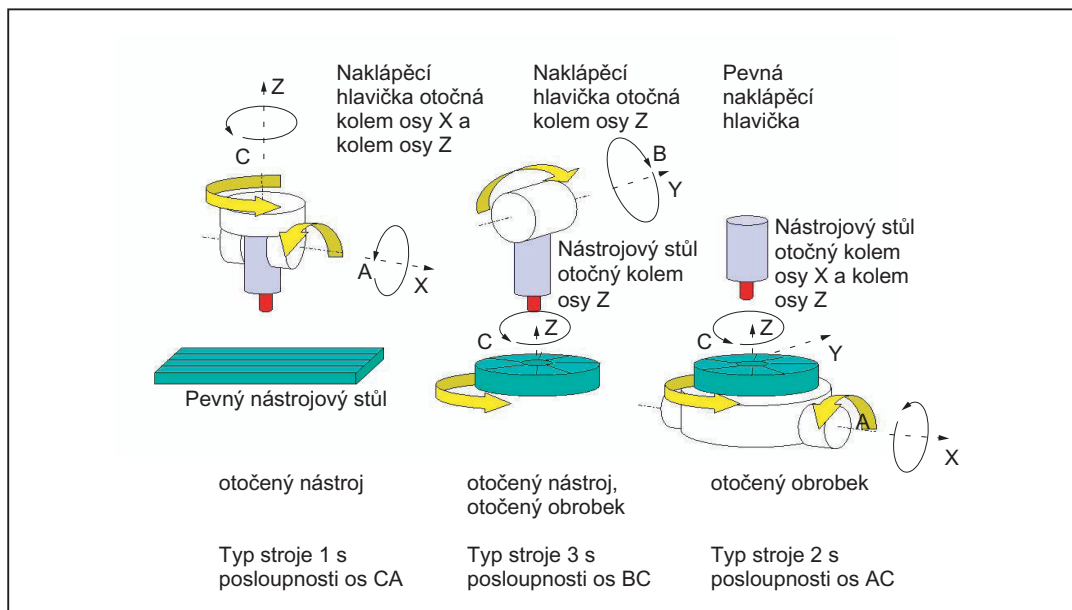
Změny poloh kruhových os podílejících se na transformaci orientace mají za následek kompenzační pohyby zbývajících strojních os. Poloha špičky nástroje zůstává přitom nezměněna.

Orientační pohyby nástroje mohou být v závislosti na použití naprogramovány prostřednictvím identifikátorů virtuálních kruhových os A..., B..., C... buď prostřednictvím zadání Eulerova úhlu nebo úhlu RPY nebo směrových vektorů nebo normálových vektorů plochy, normovaných vektorů pro osu kužele nebo pro pomocnou orientaci na kuželovou plášťovou plochu.

Při kinematické transformaci pomocí příkazů TRANSMIT, TRACYL a TRAANG řídicí systém transformuje posuvové pohyby naprogramované v kartézském souřadném systému na posuvové pohyby reálných os stroje.

Kinematika stroje při transformace ve třech, čtyřech a pěti osách (TRAORI)

Otočný může být buď nástroj nebo nástrojový stůl s až maximálně dvěma kruhovými osami. Možné jsou také kombinace naklápěcí hlavičky s jednou osou a otočného stolu.



Typ stroje	Programování orientace
Třiosá transformace u strojů typu 1 a 2	Programování orientace nástroje jen v rovině, která je kolmá na osu otáčení Existují dvě translační osy (lineární osy) a jedna rotační osa (kruhová osa).
Čtyřosá transformace u strojů typu 1 a 2	Programování orientace nástroje jen v rovině, která je kolmá na osu otáčení Existují tři translační osy (lineární osy) a jedna rotační osa (kruhová osa).
Pětiosá transformace pro typ stroje 3 Naklápěcí hlavička s jednou osou a otočný stůl s jednou osou	Programování transformace orientace. Kinematika se třemi lineárními osami a dvěma ortogonálními kruhovými osami. Kruhové osy jsou rovnoběžné se dvěma ze tří lineárních os. První kruhová osa se pohybuje dvěma kartézskými lineárními osami. Otáčí třetí lineární osu s nástrojem. Druhá kruhová osa otáčí obrobkem.

Generické 5/6-osé transformace

Typ stroje	Programování transformace orientace
Generická Pěti-/šestiosá transformace, typ stroje 4 Naklápěcí hlavička se dvěma osami a s nástrojem, který se může otáčet okolo sebe sama, a otočný stůl s jednou osou	Programování transformace orientace. Kinematika se třemi lineárními osami a třemi ortogonálními kruhovými osami. Kruhové osy jsou rovnoběžné se dvěma ze tří lineárních os. První kruhová osa se pohybuje dvěma kartézskými lineárními osami. Otáčí třetí lineární osu s nástrojem. Druhá kruhová osa otáčí obrobkem. Základní orientace nástroje může být naprogramována pomocí dodatečného otočení nástroje okolo jeho osy o úhel THETA.

Při vyvolávání "generické tří-, čtyř- a pěti-/šestiosé transformace" může být navíc ještě předávána také základní orientace nástroje. Omezení týkající se směrů u kruhových os už neplatí. Jestliže kruhové osy nejsou uspořádány přesně kolmo na sebe nebo pokud se stávají

kruhové osy nenacházejí přesně rovnoběžně s lineárními osami, je možné pomocí "generické pěti-/šestiosé transformace dosáhnout lepších výsledků orientace nástroje.

Kinematické transformace TRANSMIT, TRACYL a TRAANG

Pro frézovací práce na soustruzích nebo pro šikmo nastavitelnou osu při broušení platí v závislosti na transformaci ve standardním případě pro osy následující požadavky:

TRANSMIT	Aktivování polární transformace
Obrábění na čelní ploše při upnutí pro soustružení	Kruhová osa Přisuvná osa kolmá ke kruhové ose Podélná osa rovnoběžná s otočnou osou

TRACYL	Aktivování transformace válcového pláště
Obrábění drážek libovolného průběhu na válcovém tělese	Kruhová osa Přisuvná osa kolmá ke kruhové ose Podélná osa rovnoběžná s otočnou osou

TRAANG	Aktivování transformace šikmé osy
Obrábění s šikmou přisuvnou osou	Kruhová osa Přisuvná osa s úhlem, který může být nastaven pomocí parametru Podélná osa rovnoběžná s otočnou osou

Posuv PTP v kartézských souřadnicích

Pohyb stroje se uskutečňuje pomocí souřadnic stroje a pro jeho programování se používá:

TRAORI	Aktivování transformace
PTP - posuv od bodu k bodu	Poloha v kartézském souřadném systému (MCS)
CP	Pohyb po dráze kartézskými osami (BCS)
STAT	Nastavení polohy kloubu závisí na transformaci
TU	O jaký úhel mají osy najíždět po nejkratší dráze

Najíždění PTP při generické 5/6-osé transformaci

Pohyb stroje se uskutečňuje v souřadnicích stroje a orientaci nástroje je možné naprogramovat jak pomocí poloh kruhových os, tak také pomocí Eulerova vektoru, příp. úhlu RPY nezávislých na kinematické stroje nebo pomocí směrových vektorů.

Přitom jsou možné interpolace kruhových os, vektorové interpolace s využitím interpolace největší kružnice koule nebo interpolace vektoru orientace na kuželové válcové ploše.

Příklad tří- až pětiosé transformace u kardanové frézovací hlavičky

Obráběcí stroj má minimálně 5 os, z nichž jsou:

- tři osy lineární s přímkovými pohyby, které mohou najet na pracovní bod v jakékoli libovolné poloze v pracovním prostoru
- dvě otočné naklápěcí osy, které jsou uspořádány pod úhlem nastaveným v konfiguraci (většinou 45 stupňů) a které umožňují nastavování orientace nástroje v prostoru, které je v případě uspořádání pod úhlem 45 stupňů omezeno na polokouli

7.1.2 Přehled transformace orientace TRAORI

Možné druhy programování v souvislosti s transformací TRAORI

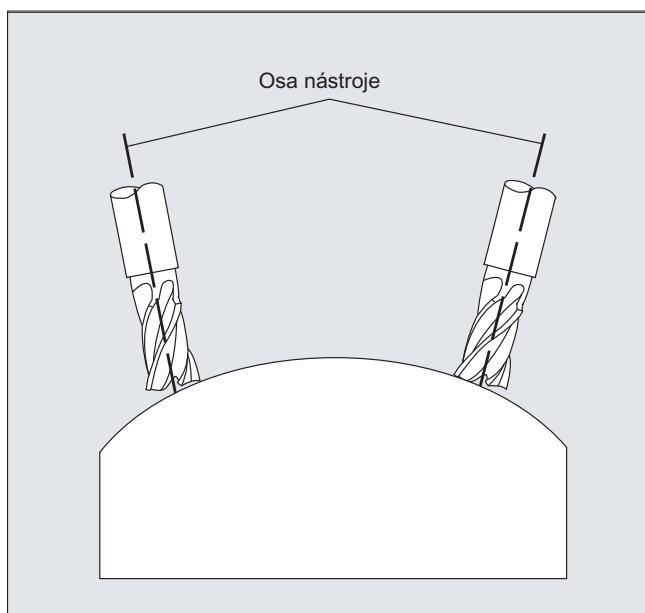
Typ stroje	Programování při aktivní transformaci TRAORI
Typy strojů 1, 2 nebo 3. Naklá- pěcí hlavička se dvěma osami nebo otočný stůl se dvěma osami nebo kombinace naklá- pěcí hlavičky a otočného sto- lu, které mají po jedné ose.	Posloupnost orientačních os a směrů orientace nástroje je buď vztažena ke stroji nastavuje se v konfiguraci pomocí strojních para- metrů v závislosti na kinematice stroje nebo vztaženo na souř. systém obrobku s programovatelnou orientací nezávisle na kinematice stroje Směr otáčení orientačních os se ve vztažném systému naprogramuje pomocí příkazů - ORIMKS vztažný systém = souřadný systém stroje - ORIWKS vztažný systém = souřadný systém obrobku Základní nastavení je ORIWKS. Programování orientačních os zadáním těchto dat: A, B, C - polohy os stroje přímo A2, B2, C2 - programování úhlu virtuálních os pomocí - ORIEULER - pomocí Eulerova úhlu (standardní) - ORIRPY - pomocí úhlu RPY - ORIVIRT1 - pomocí virtuálních orientačních os (definice 1) - ORIVIRT2 - pomocí virtuálních orientačních os (definice 2) s rozlišením druhu interpolace: lineární interpolace - ORIAxes pro orientační osy nebo osy stroje interpolace pomocí největší kružnice koule (interpolace vektoru orien- tace) - ORIVECT pro orientační osy Programování orientačních os zadáním těchto dat: A3, B3, C3 - vektorové složky (směr/normála k ploše) Programování výsledné orientace nástroje A4, B4, C4 - normálové vektory plochy na začátku bloku A5, B5, C5 - normálové vektory plochy na konci bloku LEAD - předozadní úhel pro orientaci nástroje TILT - úhel bočního naklopení pro orientaci nástroje

Typ stroje	Programování při aktivní transformaci TRAORI
	Interpolace vektoru orientace na kuželové plášťové ploše Změny orientace na kuželové plášťové ploše nacházející se libovolně v prostoru prostřednictvím interpolace: - ORIPLANE v rovině (interpolace pomocí největší kružnice koule) - ORICONCW - po ploše pláště kužele ve směru hodinových ručiček - ORICONCCW - po ploše pláště kužele proti směru hodinových ručiček A6, B6, C6 - směrové vektory (osa otáčení kužele) -OICONIO - Interpolace na kuželové plášťové ploše pomocí: A7, B7, C7 - pomocné vektory (počáteční a koncová orientace) nebo - ORICONTO - na ploše pláště kuželu s tangenciálním přechodem Změny orientace vztahující se na dráhu s příkazy - ORICURVE - zadání pohybu přes dva uzlové body pomocí PO[XH]=(xe, x2, x3, x4, x5) - polynom orientace až do 5. stupně PO[YH]=(ye, y2, y3, y4, y5) - polynom orientace až do 5. stupně PO[ZH]=(ze, z2, z3, z4, z5) - polynom orientace až do 5. stupně - ORIPATHS - vyhlazování charakteristiky orientace A8, B8, C8 - Fáze změny orientace nástroje odpovídá: Směru a délce dráhy nástroje při pozvedávacím pohybu
Typy stroje 1 a 3 Další typy strojů, které umožňují otáčení nástroje okolo sebe sama, vyžadují 3. osu Transformace orientace, jako např. generická 6-osá transformace. Otáčení vektoru orientace.	Programování otočení orientace nástroje pomocí následujících parametrů: LEAD - předozadní úhel vzhledem k normálovému vektoru plochy PO[PHI] - programování polynomu až 5. stupně TILT - boční úhel otočení k tečně ke dráze (směr Z) PO[PSI] - programování polynomu až 5. stupně THETA - úhel otočení (otáčení okolo směru nástroje ve směru Z) THETA= hodnota, která bude dosažena na konci bloku THETA=AC(...) - blokové přepnutí na zadávání absolutních rozměrů THETA=AC(...) - blokové přepnutí na zadávání inkrementálních rozměrů THETA=Θ_e - interpolace naprogramovaného úhlu G90/G91 PO[THT]=(...) - programování polynomu až 5. stupně Programování vektoru otočení - ORIROTA - absolutní otočení - ORIROTR - relativní vektor otočení - ORIROTT - tangenciální vektor otočení
Orientace vztahující se k dráze pro popis změn orientace vzhledem ke dráze nebo otáčení vektoru tangenciálně k dráze	Změny orientace vztažené k dráze pomocí příkazů: - ORIPATH - Orientace nástroje vztažená na dráhu - ORIPATHS - navíc v případě skokové změny na charakteristice orientace Programování vektoru otočení - ORIROTC - tangenciální vektor otočení, otáčení k tečně dráhy

7.2 Transformace ve třech, čtyřech a pěti osách (TRAORI)

7.2.1 Všeobecné souvislosti v případě kardanové nástrojové hlavičky

Aby bylo dosaženo optimálních řezných podmínek při obrábění prostorově zakřivených ploch, musí být možné měnit úhel nastavení nástroje.

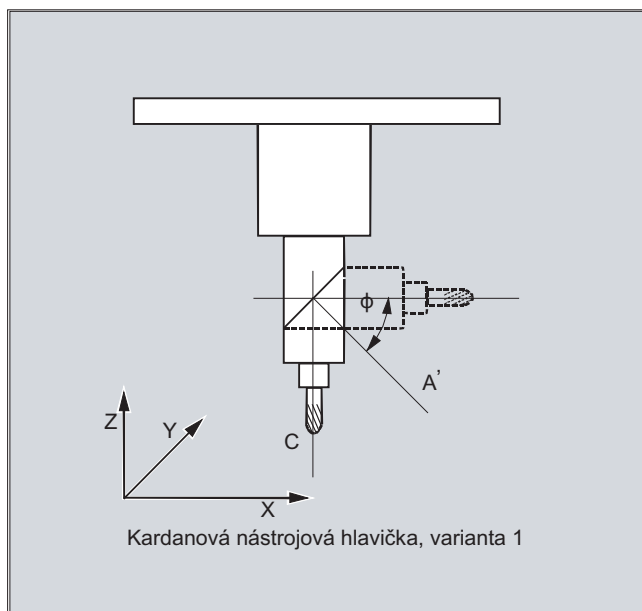


To, se kterými konstrukcemi stroje se toho dosáhne, je uloženo v parametrech os.

5-osá transformace

Kardanová nástrojová hlavička

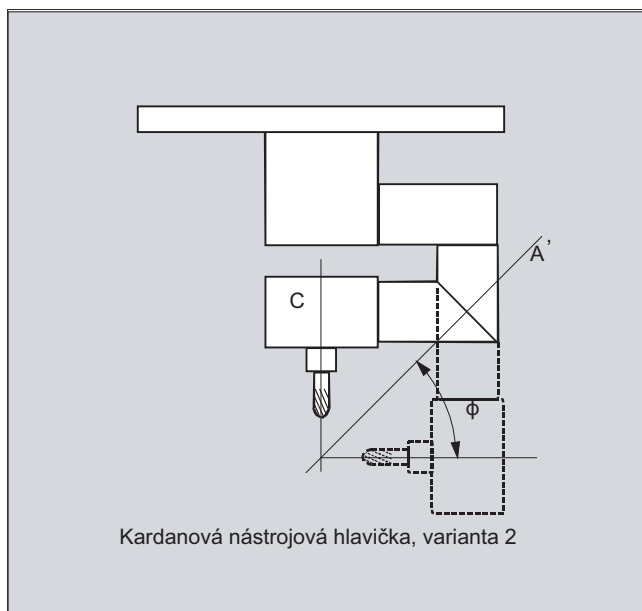
V tomto případě definují tři lineární osy (X, Y, Z) a dvě orientační osy (C, A) úhel nastavení a pracovní bod nástroje. Jedna z obou orientačních os je nastavena jako šikmá osa, zde v tomto příkladu A' - v mnoha případech jako uspořádání pod úhlem 45°.



Ve zde uváděných příkladech vidíte uspořádání na případu kinematiky stroje CA s kardanovou nástrojovou hlavičkou!

Výrobce stroje

Posloupnost orientačních os a směr orientace nástroje mohou být pomocí strojních parametrů nastaveny nezávisle na kinematice stroje.



V tomto příkladu leží A' pod úhlem ϕ vůči ose X

Obecně platí následující možné souvislosti:

A' leží pod úhlem φ vůči ose Osa X

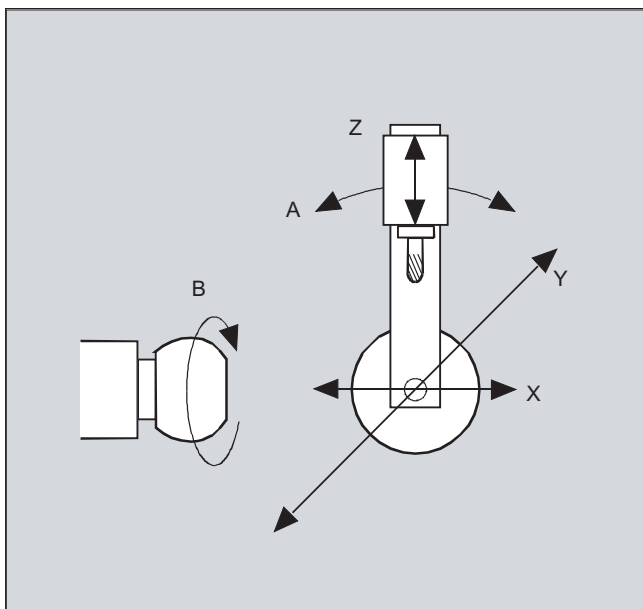
B' leží pod úhlem φ vůči ose Osa Y

C' leží pod úhlem φ vůči ose Osa Z

Úhel φ může být v rozsahu 0° až $+89^\circ$ nastaven v konfiguraci pomocí strojních parametrů.

S naklápěcí lineární osou

V tomto případě se jedná o uspořádání s pohyblivým obrobkem a pohyblivým nástrojem. Kinematika se skládá ze tří lineárních os (X, Y, Z) a tří kruhových os uspořádaných vůči sobě navzájem v pravých úhlech. První kruhová osa se pohybuje např. nad křížovým suportem dvou lineárních os, nástroj se nachází rovnoběžně se třetí lineární osou. Druhá kruhová osa otáčí obrobek. Třetí lineární osa (naklápěcí osa) leží v rovině křížového suportu.



Posloupnost rotační os a směr orientace nástroje mohou být pomocí strojních parametrů nastaveny nezávisle na kinematice stroje.

Platí následující možné souvislosti:

Osy:	Posloupnost os:
1. Kruhová osa	A A B B C C
2. Kruhová osa	B C A C A B
Naklopená lineární osa	Z Y Z X Y X

Pokud budete potřebovat další vysvětlení týkající se možností konfigurace posloupností os pro směr orientace nástroje, viz:

Literatura: /FB3/ Příručka k funkcím, Speciální funkce; 3- až 5-osá transformace (F2), kapitola "Kardanová frézovací hlavička", "Nastavování pomocí parametru".

7.2.2 Transformace ve třech, čtyřech a pěti osách (TRAORI)

Uživatel může nastavit v konfiguraci dvě, příp. tři lineární osy a jednu osu kruhovou. Transformace vychází z toho, že se kruhová osa nachází kolmo na orientační rovinu.

Orientace nástroje je možná pouze v rovině, která je kolmá na kruhovou osu. Tato transformace podporuje typy strojů s pohyblivým nástrojem a pohyblivým obrobkem.

Konfigurace a programování tří- a čtyřosé transformace se provádí analogicky k pětiosé transformaci.

Literatura:

Příručka k funkcím, Speciální funkce; Transformace ve více osách (F2)

Syntaxe

TRAORI (<n>)

TRAORI (<n>, <X>, <Y>, <Z>, <A>,)

TRAFOOF

Význam

TRAORI:	Aktivuje první deklarovanou transformaci orientace	
TRAORI (<n>):	Aktivuje n-tou deklarovanou transformaci orientace	
<n>:	Číslo transformace	
	Hodnota:	1 nebo 2
	Příklad: TRAORI(1) aktivuje transformaci orientace 1	
<X>, <Y>, <Z>:	Složky vektorů orientace, do kterých ukazuje nástroj	
<A>, :	Programovatelný offset pro kruhové osy	
TRAFOOF:	Deaktivování transformace	

Orientace nástroje

V závislosti na zvolených směrech orientace nástroje musí v NC programu nastavena aktivní pracovní rovina (G17, G18, G19) takovým způsobem, aby korekce délky nástroje působila ve směru orientace nástroje.

Poznámka

Po aktivování transformace jsou údaje o poloze (X, Y, Z) vztaženy vždy na špičku nástroje. Změna polohy kruhových os podílejících se na transformaci vede ke kompenzačním pohybům zbývajících os stroje, čímž se zajistí, že poloha špičky nástroje zůstane nezměněna.

Transformace orientace je vždy definována od špičky nástroje k jeho upevnění.

Offset pro orientační osy

Při aktivování transformace orientace je možné přímo naprogramovat doplňkový offset pro orientační osy.

Pokud je při programování dodržena správná posloupnost, smí být některé parametry vypuštěny.

Příklad:

TRAORI (, , , A, B) ; jestliže se mají zadat jen některé offsety.

Další možností vedle přímého naprogramování, jak mohou být doplňkové offsety pro orientační osy také zadány, je automatické převzetí z momentálně aktivního posunutí počátku. Tato funkce pro přebírání se nastavuje v konfiguraci pomocí strojních parametrů.

Příklady

TRAORI (1,0,0,1)	; v základní orientaci je nástroj nasměrován v ose Z
TRAORI (1,0,1,0)	; v základní orientaci je nástroj nasměrován v ose Z
TRAORI (1,0,1,1)	; v základní orientaci nástroj ukazuje ve směru Y/Z (odpovídá poloze -45°)

7.2.3 Varianty programování orientace a základního nastavení (ORIRESET)

Programování orientace nástroje pomocí funkce TRAORI

Ve spojení s programovatelnou transformací orientace TRAORI mohou být kromě lineárních os X, Y, Z pomocí identifikátorů kruhových os programovány také polohy os A..., B..., C... nebo virtuální osy s úhly nebo komponenty vektorů. Pro orientační osy a osy stroje jsou možné různé druhy interpolace. Nezávisle na tom, které polynomy orientace PO[úhel] a polynomy os PO[osa] jsou právě aktivní, může být naprogramován větší počet různých druhů polynomů, jako např. G1, G2, G3, CIP nebo POLY.

Změna orientace nástroje může být naprogramována také pomocí vektorů orientace. Orientace na konci každého bloku přitom může být zadávána buď přímým naprogramováním vektorů nebo naprogramováním poloh kruhových os.

Varianty programování orientace v případě tří- až pětiosých transformací

Následující varianty programování nastavení orientace se vzájemně vylučují.

A, B, C	Přímé zadávání poloh os stroje
A2, B2, C2	Programování úhlu virtuálních os pomocí Eulerova úhlu nebo úhlu RPY
A3, B3, C3	Zadání vektorových složek
LEAD, TILT	Zadávání předozadního a bočního úhlu vztažené na dráhu a plochu
A4, B4, C4 A5, B5, C5	Normálový vektor plochy na začátku bloku a na konci bloku
A6, B6, C6 A7, B7, C7	Interpolace vektoru orientace na kuželové plášťové ploše
A8, B8, C8	Změna orientace nástroje, směr a délka dráhy pohybu při pozvednutí

Najíždění na základní nastavení orientace nástroje (ORIRESET)

Prostřednictvím funkce `ORIRESET ( . . . )` jsou osy příslušné kinematiky stroje pro nastavování orientace lineárně a synchronně přestaveny ze své aktuální polohy do naprogramované základní polohy. Jestliže pro nějakou osu není naprogramována žádná základní poloha, potom se použije poloha stanovená odpovídajícím strojním parametrem `$MC_TRAFO5_ROT_AX_OFFSET_1/2`.

Aktivní framy kruhových os nejsou nijak zohledňovány.

Příklady kinematiky stroje CA (názvy kanálových os C, A)

Příkazy	Popis
<code>ORIRESET (90, 45)</code>	Osa C: 90° Osa A: 45°
<code>ORIRESET (, 30)</code>	Osa C: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[0]</code> Osa A: 30°
<code>ORIRESET ()</code>	Osa C: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[0]</code> Osa A: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[1]</code>

Příklady kinematiky stroje CAC (názvy kanálových os C, A, B)

Příkazy	Popis
<code>ORIRESET (90, 45, 90)</code>	Osa C: 90° Osa A: 45° Osa B: 90°
<code>ORIRESET ()</code>	Osa C: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[0]</code> Osa A: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[1]</code> Osa B: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[2]</code>

Poznámka

Najíždění do základního nastavení orientace nástroje pomocí příkazu `ORIRESET ( . . . )` se smí uskutečňovat, jen když je aktivní transformace orientace `TRAORI ( . . . )`.

Programování otočení LEAD, TILT a THETA**Předozadní úhel LEAD a boční úhel TILT**

Otočení orientace nástroje se v případě tří- až pětiosé transformace programují pomocí předozadního úhlu LEAD a pomocí bočního úhlu TILT.

Úhel otočení THETA

V případě transformace s **třetí kruhovou osou** může být naprogramováno nastavení orientace jak pomocí vektorových složek, tak také může být pomocí úhlů LEAD a TILT naprogramováno otočení nástroje okolo sebe sama s úhlem natočení THETA.

7.2.4 Programování orientace nástroje (A..., B..., C..., LEAD, TILT)

Pro programování orientace nástroje existují následující možnosti:

1. Přímé programování pohybu kruhových os. Změna orientace se vždy uskutečňuje v základním souřadném systému nebo v souřadném systému stroje. Orientační osy se pohybují jako synchronní osy.
2. Naprogramování Eulerových úhlů nebo úhlů RPY podle definice prostřednictvím A2, B2, C2.
3. Naprogramování směrových vektorů pomocí A3, B3, C3. Směrový vektor ukazuje od špičky nástroje ve směru jeho unášeče (držáku).
4. Programování normálových vektorů plochy na začátku bloku pomocí A4, B4, C4 na na konci bloku pomocí A5, B5, C5 (frézování na čelní ploše).
5. Naprogramování pomocí předozadního úhlu LEAD a bočního úhlu TILT.
6. Naprogramování osy kužele jako normovaného vektoru pomocí A6, B6, C6 nebo pomocné orientace na kuželové plášťové ploše pomocí A7, B7, C7, viz kapitola "Programování orientace podél kuželové plášťové plochy (ORIPANE, ORICONxx)".
7. Programování změny orientace, směru a délky dráhy nástroje v průběhu pohybu při pozvednutí pomocí A8, B8, C8, viz kapitola "Vyhazení průběhu orientace (ORIPATHS A8=, B8=, C8=)".

Poznámka

Ve všech případech je programování orientace přípustné jen tehdy, pokud je transformace orientace aktivována.

Výhoda: Tyto programy jsou přenositelné na jakoukoli kinematiku stroje.

Definice orientace nástroje pomocí příkazu G-funkce

Poznámka

Výrobce stroje

Pomocí strojního parametru je možné přepínat mezi Eulerovými úhly a úhly RPY. V případě odpovídajících nastavení strojních parametrů je přepínání možné jak nezávisle, tak i v závislosti na aktivním příkazu G-funkce ze skupiny 50. Můžete si vybrat z následujících možností nastavení:

1. Jestliže jsou oba strojní parametry pro definici orientačních os a definici orientačních úhlů nastaveny pomocí příkazů G-funkcí na nulu: Úhel naprogramovaný pomocí A2, B2, C2 je **v závislosti na strojním parametru** pro definici úhlu pro programování orientace interpretován buď jako Eulerův úhel nebo jako úhel RPY.
2. Pokud je strojní parametr pro definici orientačních os nastaven pomocí příkazu G-funkce na jedna, provádí se přepínání **v závislosti** na aktivním příkazu G-funkce ze skupiny 50:
Úhly naprogramované pomocí A2, B2, C2 jsou interpretovány podle aktivního příkazu G-funkce ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS a ORIPY2.
Hodnoty naprogramované prostřednictvím orientačních os jsou interpretovány v souladu s aktivním příkazem G-funkce ze skupiny 50 také jako orientační úhel.
3. Jestliže je strojní parametr pro definici úhlu orientace prostřednictvím příkazu G-funkce nastaven na jedničku a strojní parametr pro definici orientačních os pomocí příkazu G-funkce je nastaven na nulu, provádí se přepínání **nezávisle** na aktivním příkazu G-funkce ze skupiny 50:
Úhly naprogramované pomocí A2, B2, C2 jsou interpretovány podle aktivního příkazu G-funkce ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS a ORIPY2.
Hodnoty naprogramované prostřednictvím orientačních os jsou interpretovány nezávisle na aktivním příkazu G-funkce ze skupiny 50 vždy jako polohy kruhových os.

Syntaxe

Polohy kruhových os

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A<hodnota> B<hodnota> C<hodnota>
```

Eulerův úhel

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A2=<hodnota> B2=<hodnota>  
C2=<hodnota>
```

Směrový vektor

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A3=<hodnota> B3=<hodnota>  
C3=<hodnota>
```

Normálový vektor plochy na začátku bloku

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A4=<hodnota> B4=<hodnota>  
C4=<hodnota>
```


Normálový vektor plochy na konci bloku

G1 X<hodnota> Y<hodnota> Z<hodnota> A5=<hodnota> B5=<hodnota>
C5=<hodnota>

Předožadní úhel

LEAD=<hodnota>

Úhel bočního naklonění

TILT=<hodnota>

Význam

G1:	Lineární interpolace
X, Y, Z:	Polohy lineárních os
A, B, C:	Polohy kruhových os
A2=, B2=, C2=:	Naprogramovaný úhel (Eulerův úhel nebo úhel RPY)
A3=, B3=, C3=:	Směrový vektor v souřadnicích X, Y a Z ve WCS
A4=, B4=, C4=:	Normálový vektor plochy na začátku bloku v souřadnicích X, Y a Z ve WCS
A5=, B5=, C5=:	Normálový vektor plochy na konci bloku v souřadnicích X, Y a Z ve WCS
LEAD= :	Předožadní úhel ¹⁾
TILT= :	Boční úhel ¹⁾
1) Interpretace zadaných hodnot úhlu závisí na nastavení parametru MD21094 \$MC_ORIPATH_MODE.	

Další informace

5-osé programy jsou zpravidla vytvářeny v systémech CAD/CAM a nejsou zadávány na řídicím systému. Z tohoto důvodu jsou následující vysvětlení určena především pro programátory postprocesorových systémů.

Pro programování orientace vám jsou k dispozici následující příkazy:

Příkaz	Význam
ORIEULER:	Eulerův úhel s posloupností při otáčení ZX'Z"
ORIRPY:	Úhel RPY s posloupností při otáčení XY'Z"
ORIRPY2:	Úhel RPY s posloupností při otáčení ZY'X"
ORIVIRT1:	Virtuální orientační osy s volně definovatelnou posloupností při otáčení pomocí parametru: MD21120 \$MC_ORIAX_TURN_TAB_1
ORIVIRT2:	Virtuální orientační osy s volně definovatelnou posloupností při otáčení pomocí parametru: MD21130 \$MC_ORIAX_TURN_TAB_2
ORIAPOS:	Virtuální orientační osy v polohách kruhových os

Poznámka

Prostřednictvím strojních parametrů mohou být výrobcem stroje definovány různé varianty. Věnujte prosím pozornost informacím od výrobce stroje.

Programování pomocí Eulerových úhlů `ORIEULER`, posloupnost při otáčení $Z\ X'\ Z''$

Hodnoty naprogramované pomocí příkazu `ORIEULER` s hodnotami $A2$, $B2$, $C2$ při programování orientace jsou interpretovány jako Eulerův úhel (ve stupních).

Nový vektor orientace vznikne prostřednictvím následujících tří otočení původního vektoru orientace

1. pomocí kruhové osy $A2$ okolo souřadné osy Z
2. pomocí kruhové osy $B2$ okolo nové souřadné osy X'
3. pomocí kruhové osy $C2$ okolo nové souřadné osy Z''

V tomto případě nemá hodnota $C2$ (otočení okolo nové osy Z) žádný význam a nemusí být proto naprogramována.

Programování v úhlech RPY pomocí příkazu `ORIRPY`, posloupnost při otáčení $X\ Y'\ Z''$

Hodnoty naprogramované pomocí funkce `ORIRPY` s parametry $A2$, $B2$, $C2$ při programování orientace jsou interpretovány jako úhel RPY (ve stupních) s posloupností při otáčení $X\ Y'\ Z''$.

Poznámka

Oproti programování pomocí příkazu `ORIEULER` mají při použití funkce `ORIRPY` na vektor orientace vliv všechny tři hodnoty.

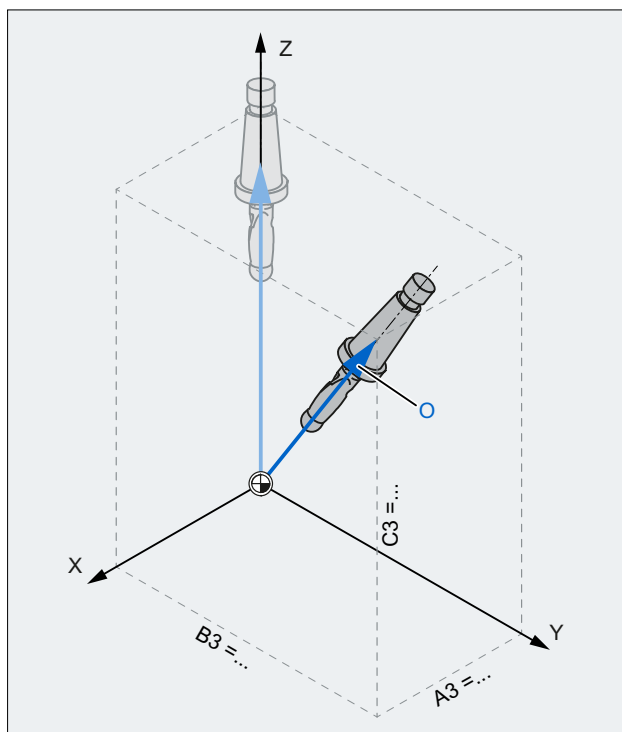
Nový vektor orientace vznikne prostřednictvím následujících tří otočení původního vektoru orientace

1. pomocí kruhové osy $A2$ okolo souřadné osy X
2. pomocí kruhové osy $B2$ okolo nové souřadné osy Y'
3. pomocí kruhové osy $C2$ okolo nové souřadné osy Z''

Programování směrových vektorů

Komponenty směrového vektoru jsou naprogramovány pomocí $A3$, $B3$, $C3$. Vektor ukazuje ve směru upnutí nástroje, délka vektoru přitom nemá žádný význam.

Nenaprogramovaným složkám vektoru je dosazena nulová hodnota.



X, Y, Z Souřadné osy WCS

A3, B3, Složky směrového vektoru
C3

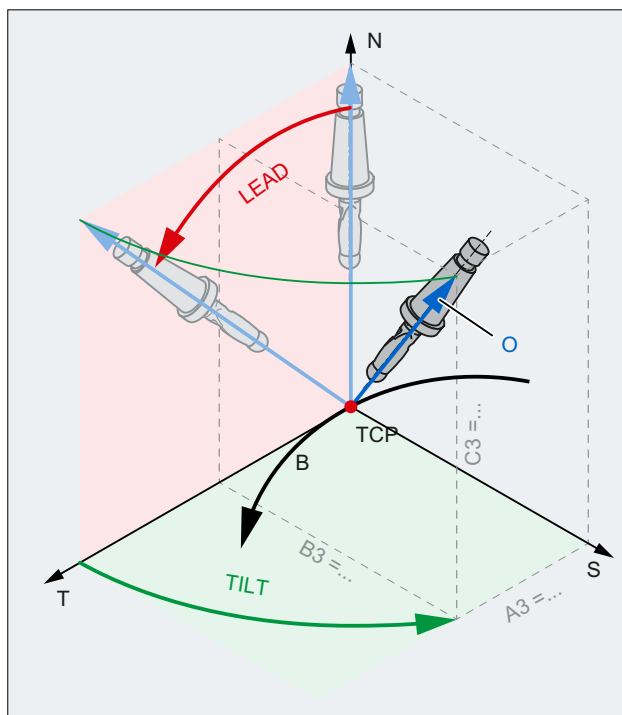
O Vektor orientace

Obrázek 7-1 Programování směrových vektorů

Programování orientace nástroje pomocí příkazů LEAD a TILT

Výsledná orientace nástroje je zjišťována z těchto údajů:

- Tečna ke dráze
- Normálový vektor plochy
Na začátku bloku A4, B4, C4 a na konci bloku A5, B5, C5
- Předozadní úhel LEAD
Úhel v rovině definované tečnou ke dráze a normálovým vektorem plochy
- Úhel bočního naklopení TILT na konci bloku
Úhel v rovině, která je kolmá na tečnu ke dráze vzhledem k normálovému vektoru plochy



- T Tečna ke dráze
- S Kolmo na tečnu ke dráze
- N Normálový vektor plochy
- B Dráha
- TCP Tool Center Point
- O Vektor orientace

Obrázek 7-2 Programování pomocí parametrů LEAD, TILT

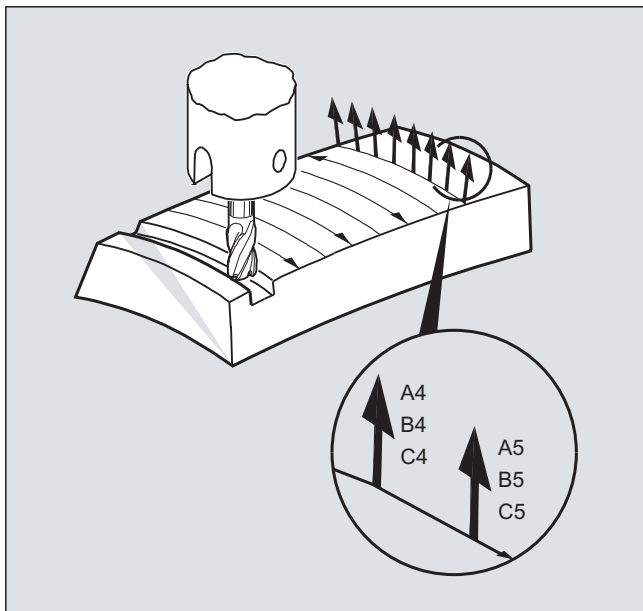
Poznámka

Chování na vnitřních rozích v případě 3D korekce nástroje

Jestliže je blok ve vnitřním rohu zkrácený, je i přesto naprogramovaná orientace nástroje nastavena na konci bloku.

7.2.5 Frézování na čelní ploše (A4, B4, C4, A5, B5, C5)

Frézování na čelní ploše slouží pro opracovávání libovolně zakřivených povrchů.



Pro tento druh 3D frézování potřebujete popis řádek po řádku jednotlivých 3D drah na povrchu obrobku.

Při výpočtech se obvykle uskutečňuje v systému CAM a zohledňuje se při něm tvar nástroje a rozměry nástroje. Nahotovo vypočítané NC bloky jsou potom prostřednictvím postprocesoru načteny do řídicího systému.

Programování zakřivení dráhy

Popis ploch

Pro popis zakřivení dráhy se využívá normálových vektorů plochy s následujícími složkami:

A4, B4, C4 - počáteční vektor na začátku bloku

A5, B5, C5 - koncový vektor na konci bloku

Jestliže se v bloku nachází pouze počáteční vektor, zůstává normálový vektor plochy po celý blok konstantní. Pokud se v bloku nachází pouze koncový vektor, uskuteční se interpolace pomocí největší kružnice koule z koncové hodnoty předcházejícího bloku do naprogramované koncové hodnoty.

Jestliže jsou naprogramovány jak počáteční, tak i koncový vektor, mezi oběma směry se rovněž uskuteční interpolace pomocí největší kružnice koule. Takto se dají vytvářet spojitě hladké pohyby po dráze.

V základním nastavení ukazují normálové vektory plochy nezávisle na aktivní rovině G17 až G19 ve směru osy Z.

Délka vektoru nemá žádný význam.

Nenaprogramovaným složkám vektoru je dosazena nulová hodnota.

Když je aktivní příkaz ORIWKS (viz kapitola "Vztah orientačních os (ORIWKS, ORIMKS) (Strana 346)", jsou normálové vektory plochy vztaženy na aktivní frame a při otáčení framu se otáčejí také.

Výrobce stroje

Normálový vektor plochy musí být v rámci mezních hodnot nastavených pomocí strojních parametrů kolmý na tečnu ke dráze, jinak se aktivuje alarm.

7.2.6 Vztah orientačních os (ORIWKS, ORIMKS)

Při programování orientace v souřadném systému obrobku pomocí

- Eulerova úhlu nebo úhlu RPY nebo
- vektoru orientace

může být pomocí funkcí ORIMKS/ORIWKS nastaven průběh otáčivého pohybu.

Poznámka

Výrobce stroje

Druh interpolace pro orientaci je definován pomocí strojního parametru:

MD21104 \$MC_ORI_IPO_WITH_G_CODE

= FALSE: Vztah určují příkazy G-funkcí ORIWKS a ORIMKS

= TRUE: Vztah určují příkazy G-funkcí z 51. Skupiny (ORIXES, ORIVECT, ORIPLANE, ...)

Syntaxe

ORIMKS=...

ORIWKS=...

Význam

ORIMKS:	Otáčení v souřadném systému stroje
ORIWKS:	Otáčení v souřadném systému obrobku

Poznámka

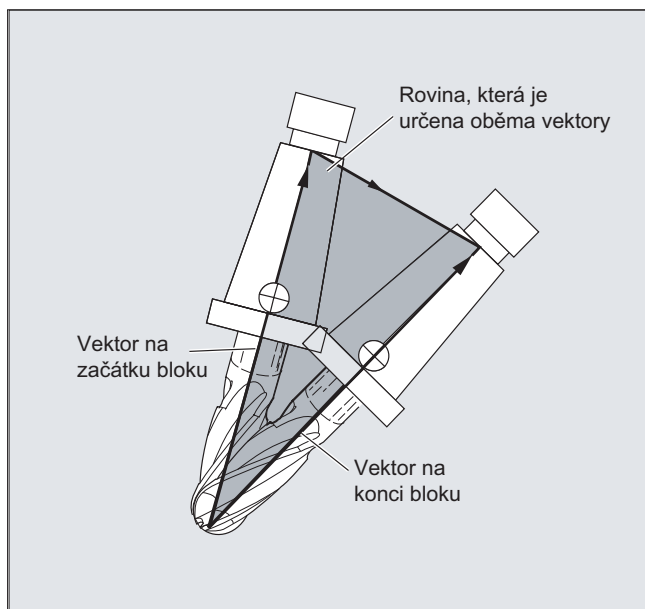
Základním nastavením je ORIWKS. Jestliže není u programu pracujícího s pěti osami předem jasné, na jakém stroji má být zpracováván, mělo by být v zásadě zvoleno ORIWKS. To, jaké pohyby stroje budou ve skutečnosti provedeny, závisí na kinematice stroje.

Pomocí ORIMKS mohou být programovány skutečné pohyby stroje, např. aby se zabránilo kolizi s různými zařízeními atd.

Další informace

V případě **ORIMKS** je prováděný pohyb nástroje **závislý** na kinematice stroje. Při změně orientace, kdy je špička nástroje na pevném místě v prostoru, se mezi dvěma polohami kruhové osy uskutečňuje lineární interpolace.

V případě **ORIWKS** je prováděný pohyb nástroje **nezávislý** na kinematice stroje. V případě změny orientace, při které špička nástroje zůstává pevně na jednom místě, se nástroj pohybuje v rovině definované počátečním a koncovým vektorem.



Singulární polohy

Poznámka

ORIWKS

Orientační pohyby v oblasti singulární polohy u stroje s pěti osami vyžadují velké pohyby os stroje. (Například v případě otočné naklápěcí hlavičky s osou C jako otočnou osou a osou A jako naklápěcí osou jsou singulární všechny polohy, kdy $A=0$.)

Výrobce stroje

Aby nebyly osy stroje příliš přetíženy, regulační systém v blízkosti singulárních poloh výrazně snižuje rychlost pohybu po dráze.

Pomocí strojních parametrů

`$MC_TRAFO5_NON_POLE_LIMIT`

`$MC_TRAFO5_POLE_LIMIT`

mohou být pro transformaci dosazeny takové parametry, aby se orientační pohyby v blízkosti pólů byly provedeny skrz pól a aby tak bylo umožněno plynulé opracování.

Singulární místa jsou ošetřena pouze strojním parametrem `$MC_TRAFO5_POLE_LIMIT`.

Literatura:

/FB3/, Příručka k funkcím, Speciální funkce; 3- až 5-osá transformace (F2), kapitola "Singulární místa a jejich ošetření".

7.2.7 Programování orientačních os (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Funkce "Orientační osy" popisuje orientaci nástroje v prostoru a daného nastavení se dosahuje naprogramováním offsetů kruhových os. Dalšího třetího stupně volnosti může být dosahováno dalším otáčením nástroje okolo sebe sama. Tyto orientace nástroje se libovolně v prostoru nastavují pomocí třetí kruhové osy a vyžadují transformaci se šesti osami. Vlastní otočení nástroje okolo sebe sama je definováno v závislosti na druhu interpolace vektorů otočení pomocí úhlu otočení THETA (viz "Otáčení orientace nástroje (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Strana 358)".

Orientační osy jsou programovány prostřednictvím identifikátorů os A2, B2, C2.

Syntaxe

```
N... ORIAXES/ORIVECT           ; Lineární interpolace nebo interpolace
                                pomocí největší kružnice koule

N... G1 X Y Z A B C

N... ORIPLANE                   ; Interpolace orientace roviny

N... ORIEULER/ORIRPY/ORIRPY2   ; Úhel orientace - Eulerův úhel / úhel RPY.
N... G1 X Y Z A2= B2= C2=      ; Programování úhlů virtuálních os

N... ORIVIRT1/ORIVIRT2         ; definice virtuálních orientačních os 1/2
N... G1 X Y Z A3= B3= C3=      ; programování směrového vektoru
```

Poznámka

Pro změny orientace podél kuželové plášťové plochy nacházející se v prostoru mohou být naprogramovány další offsety kruhových orientačních os (viz "Programování orientace podél kuželové plášťové plochy (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Strana 350)").

Význam

ORIAXES:	Lineární interpolace os stroje nebo orientačních os
ORIVECT:	Interpolace pomocí největší kružnice koule (identická s ORIPLANE)
ORIMKS:	Otáčení v souřadném systému stroje
ORIWKS:	Otáčení v souřadném systému obrobku Popis viz "Vztah orientačních os (ORIWKS, ORIMKS) (Strana 346)".
A= B= C=:	Programování poloh os stroje

ORIEULER:	Programování orientace pomocí Eulerova úhlu
ORIRPY:	Programování orientace pomocí úhlu RPY. Posloupnost při otáčení je X Y Z, přičemž platí: • A2 je úhel otočení okolo osy X • B2 je úhel otočení okolo osy Y • C2 je úhel otočení okolo osy Z
ORIRPY2:	Programování orientace pomocí úhlu RPY. Posloupnost při otáčení je Z Y X, přičemž platí: • A2 je úhel otočení okolo osy Z • B2 je úhel otočení okolo osy Y • C2 je úhel otočení okolo osy X
A2= B2= C2=:	Programování úhlů virtuálních os
ORIVIRT1/ORIVIRT2:	Programování orientace pomocí virtuálních orientačních os Definice 1: stanovení podle MD21120 \$MC_ORIAX_TURN_TAB_1 Definice 2: stanovení podle MD21130 \$MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Programování směrového vektoru směrové osy

Další informace

Výrobce stroje

Pomocí strojního parametru MD21102 \$MC_ORI_DEF_WITH_G_CODE je stanoveno, jak jsou definovány naprogramované úhly A2, B2, C2:

Definice se uskutečňuje podle strojního parametru MD21100 \$MC_ORIENTATION_IS_EULER (standardní) nebo se definice uskutečňuje podle G-funkcí ze skupiny 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Pomocí strojního parametru MD21104 \$MC_ORI_IPO_WITH_G_CODE se určuje, jaký druh interpolace se uplatňuje: ORIWKS/ORIMKS nebo ORIAxes/ORIVECT.

Provozní režim JOG

V tomto provozním režimu je úhel orientace interpolován vždy lineárně. Při spojitých a inkrementálních posuvech prostřednictvím tlačítek posuvů je možno pohybovat jen jednou orientační osou. Pomocí ručních koleček je možné pohybovat orientačními osami současně.

Za účelem manuálního ovládání orientačních os se používá kanálový spínač korekce (override) posuvu, příp. spínač pro korekci (override) rychlého posuvu.

Pomocí následujících strojních parametrů je možné oddělené nastavení rychlosti:

MD21160 \$MC_JOG_VELO_RAPID_GEO

MD21165 \$MC_JOG_VELO_GEO

MD21150 \$MC_JOG_VELO_RAPID_ORI

MD21155 \$MC_JOG_VELO_ORI

Poznámka**SINUMERIK 840D sl s "modulem pro práci s transformacemi"**

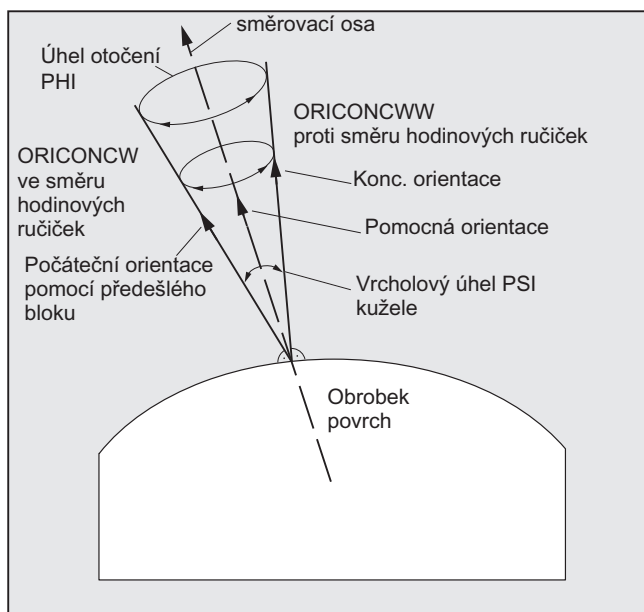
Pomocí funkce "Kartézské manuální pohyby" je možné v provozním režimu JOG od sebe odděleně nastavovat posunutí geometrických os ve vztazných systémech MCS, WCS a TCS.

Literatura:

Příručka k funkcím, Rozšiřovací funkce; Kinematická transformace (M1)

7.2.8**Programování orientace podél kuželové plášťové plochy (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)**

Pomocí rozšířené orientace je možné uskutečňovat změny orientace podél kuželové plášťové plochy nacházející se v prostoru. Interpolace vektorů orientace se na kuželové plášťové ploše uskutečňuje pomocí modálních příkazů ORICONxx. Pro interpolaci v rovině může být naprogramována koncová orientace ORIPLANE. Obecně je počáteční orientace určena na základě předcházejících bloků.



Programování

Koncová orientace se určuje buď prostřednictvím naprogramováním zadaného Eulerova úhlu nebo úhlu RPY pomocí proměnných A2, B2, C2 nebo naprogramováním poloh kruhových os A, B, C. Pro orientační osy podél kuželové plášťové plochy jsou zapotřebí další naprogramované údaje:

- Osa kužele jako vektor pomocí proměnných A6, B6, C6
- Úhel kužele PSI s identifikátorem NUT
- Pomocná orientace na plášti kužele pomocí proměnných A7, B7, C7

Poznámka

Programování směrového vektoru A6, B6, C6 pro osu kužele

Naprogramování koncové orientace není bezpodmínečně nutné. Jestliže koncová orientace není uvedena, potom se bude provádět interpolace na celém plášti kužele s úhlem 360 stupňů.

Programování úhlu kužele pomocí parametru NUT = úhel

Zadání koncové orientace je bezpodmínečně nutné.

Úplný plášť kužele s úhlem 360 stupňů není možné tímto způsobem interpolovat.

Programování pomocné orientace na plášti kužele pomocí proměnných A7, B7, C7

Zadání koncové orientace je bezpodmínečně nutné. Změna orientace a směr otáčení jsou jednoznačně určeny pomocí tří vektorů počáteční, koncové a pomocné orientace. Všechny tři vektory se přitom musí od sebe lišit. Jestliže je pomocná orientace rovnoběžná s počáteční nebo s koncovou orientací, potom se uskuteční lineární interpolace orientace pomocí největší kružnice koule v rovině, která je určena počátečním a koncovým vektorem.

Rozšířená interpolace orientace na kuželové plášťové ploše

N... ORICONCW nebo ORICONCCW
 N... A6= B6= C6= A3= B3= C3=
 nebo
 N... ORICONTO
 N... G1 X Y Z A6= B6= C6=
 nebo
 N... ORICONIO
 N... G1 X Y Z A7= B7= C7=
 N... PO[PHI]=(a2, a3, a4, a5)
 N... PO[PSI]=(b2, b3, b4, b5)

Interpolace na plášti kuželu s těmito parametry

Směrový vektor kuželu ve/proti směru hodinových ručiček a koncová orientace nebo

tangenciální přechod a
 údaj koncové orientace

nebo

údaj koncové orientace a údaj pomocné orientace na kuželovém plášti pomocí

polynomu pro úhel otočení a
 polynomu pro vrcholový úhel

Parametr

ORIPLANE:	Interpolace v rovině (interpolace pomocí největší kružnice koule)
ORICONCW:	Interpolace po ploše pláště kužele ve směru hodinových ručiček
ORICONCCW:	Interpolace po ploše pláště kužele proti směru hodinových ručiček
ORICONTO:	Interpolace na ploše pláště kuželu s tangenciálním přechodem

A6= B6= C6=:	Programování osy kužele (normovaný vektor)
NUT=úhel:	Úhel kužele ve stupních
NUT=+179:	Úhel pohybu menší nebo roven 180 stupňům
NUT=-181:	Úhel pohybu větší nebo roven 180 stupňům
ORICONIO:	Interpolace na kuželové plášťové ploše
A7= B7= C7=:	Pomocná orientace (programuje se jako normovaný vektor)
PHI:	Úhel otočení orientace okolo směrové osy kužele
PSI:	Úhel kužele
možné polynomy PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Kromě příslušných úhlů mohou být naprogramovány také polynomy maximálně 5. stupně

Příklad: Různé změny orientace

Programový kód	Komentář
...	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; aktivování transformace orientace
N30 ORIVECT	; interpolace orientace nástroje jako vektor
...	; orientace nástroje v rovině
N40 ORIPLANE	; aktivování interpolace pomocí největší kružnice koule
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; orientace v rovině Y/Z otočená o 45 stupňů, na konci bloku bude dosaženo orientace (0,1/√2,1/√2)
...	
N70 ORICONCW	; programování orientace na plášti kužele:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	; interpolace vektoru orientace na plášti kužele se směrem (0,0,1) až do orientace (1/√2,0,1/√2) ve směru hodinových ručiček, úhel přitom činí 270 stupňů
N90 A6=0 B6=0 C6=1	; orientace nástroje proběhne celou jednu otáčku po stejném plášti kuželu

Další informace

Jestliže mají být popsány změny orientace na kuželové plášťové ploše nacházející se libovolně v prostoru, potom musí být znám vektor, okolo kterého se má orientace nástroje otáčet. Kromě toho musí být zadána počáteční a koncová orientace. Počáteční orientace vyplývá z předcházejícího bloku a koncová orientace musí být buď naprogramována nebo musí být stanovena na základě dalších podmínek.

Programování v rovině ORIPLANE odpovídá funkci ORIVECT

Naprogramování interpolace pomocí největší kružnice koule spolu s polynomy popisujícími úhel odpovídá lineární a polynomické interpolaci kontur. Orientace nástroje je interpolována

v rovině, která je definována počáteční a koncovou orientací. Pokud je kromě toho naprogramován polynom, může se vektor orientace naklápět i mimo tuto rovinu.

Programování kruhových drah v rovině pomocí příkazů G2/G3, CIP a CT

Rozšířená orientace odpovídá interpolaci kruhových oblouků v rovině. Pokud budete potřebovat odpovídající informace o možnostech programování kruhových oblouků pomocí středu nebo rádiusu (G2/G3), pomocí vnitřního bodu CIP a kruhových oblouků s tangenciálním napojením, viz:

Literatura: Programovací příručka, Základy; "Programování příkazů dráhy"

Programování orientace

Interpolace vektoru orientace na kuželové plášťové ploše (ORICONxx)

Pro interpolaci orientace na kuželové plášťové ploše mohou být používány čtyři různé druhy interpolace ze skupiny příkazů G-funkcí č. 51.

1. Interpolace na plášti kužele ve směru hodinových ručiček `ORICONCW` se zadáním koncové orientace a směru kužele nebo úhlu kužele. Směrový vektor se programuje pomocí identifikátorů `A6`, `B6`, `C6` a vrcholový úhel kužele se programuje pomocí identifikátoru `NUT=` s intervalem hodnot 0 až 180 stupňů.
2. Interpolace na plášti kužele proti směru hodinových ručiček `ORICONCWW` se zadáním koncové orientace a směru kužele nebo úhlu kužele. Směrový vektor se programuje pomocí identifikátorů `A6`, `B6`, `C6` a vrcholový úhel kužele se programuje pomocí identifikátoru `NUT=` s intervalem hodnot 0 až 180 stupňů.
3. Interpolace na plášti kužele pomocí příkazu `ORICONIO` se zadáním koncové orientace a pomocné orientace, která se programuje pomocí identifikátorů `A7`, `B7`, `C7`.
4. Interpolace po ploše pláště kužele pomocí příkazu `ORICONTO` s tangenciálním přechodem a se zadáním koncové orientace. Směrový vektor se programuje pomocí identifikátorů `A6`, `B6`, `C6`.

7.2.9 Zadání orientace pomocí druhého styčného bodu (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Programování změny orientace pomocí druhé prostorové křivky ORICURVE

Další možnost, jak naprogramovat změny orientace, spočítá v tom, že se kromě pohybu špičky nástroje podél prostorové křivky naprogramuje pomocí příkazu `ORICURVE` také pohyb druhého styčného bodu nástroje. Tímto způsobem mohou být jednoznačně definovány změny orientace nástroje stejně jako při naprogramování vektoru nástroje samotného.

Výrobce stroje

Věnujte prosím pozornost pokynům výrobce stroje, které se týkají identifikátorů osy, které jsou nastavitelné pomocí strojních parametrů a které slouží pro programování 2. dráhy pro popis orientace nástroje.

Programování

U tohoto druhu interpolace mohou být pro obě prostorové křivky naprogramovány body pomocí příkazů `G1`, příp. polynomy pomocí příkazu `POLY`. Kruhové oblouky a evolventy jsou nepřipustné. Kromě toho mohou být aktivovány splinová interpolace pomocí příkazu `BSPLINE` a funkce "Shrnutí krátkých splinových bloků"

Literatura:

Příručka k funkcím, Základní funkce, Režim řízení pohybu po dráze, přesné najetí a funkce Look Ahead (B1), kapitola: Shrnutí krátkých splinových bloků

Použití jiných druhů splinů `ASPLINE` a `CSPLINE`, stejně jako aktivování kompresoru pomocí příkazů `COMPON`, `COMPCURV` nebo `COMPCAD` nejsou přípustné.

Souřadnice pro pohyb dvou styčných bodů nástroje může být zadán naprogramováním polynomu orientace až maximálně 5. stupně.

Rozšířená interpolace orientace pomocí doplňkové prostorové křivky a polynomu pro souřadnice

N... ORICURVE

N... PO[XH]=(xe, x2, x3, x4, x5)

N... PO[YH]=(ye, y2, y3, y4, y5)

N... PO[ZH]=(ze, z2, z3, z4, z5)

Zadání pohybu druhého styčného bodu nástroje a doplňkový polynom příslušné souřadnice

Parametr

ORICURVE	Interpolace orientace s udáním pohybu dvou styčných bodů nástroje.
XH YH ZH	Identifikátory souřadnic druhého kontaktního bodu nástroje, které popisují doplňkovou konturu jako prostorovou křivku
možné polynomy PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)	Kromě odpovídajících koncových bodů mohou být prostorové křivky naprogramovány také pomocí polynomů.
xe, ye, ze	Koncové body prostorové křivky
xi, yi, zi	Koeficienty polynomu maximálně 5. stupně

Poznámka**Identifikátory XH YH ZH pro programování 2. dráhy orientace**

Tyto identifikátory musí být zvoleny tak, aby nemohlo dojít ke konfliktu s jinými identifikátory lineárních os

X Y Z - osy

a kruhovými osami, jako např.

A2 B2 C2 - Eulerův úhel, příp. úhel RPY

A3 B3 C3 - směrové vektory

A4 B4 C4 příp. A5 B5 C5 - normálové vektory plochy

A6 B6 C6 - vektory otáčení, příp. A7 B7 C7 souřadnice vnitřního bodu
nebo jinými interpolačními parametry.

7.3 Polynomický popis orientace (PO[úhel], PO[souřadnice])

Nezávisle na tom, která polynomická interpolace ze skupiny příkazů G-funkcí č.1 je momentálně aktivní, mohou být při tří- až pětiosé transformaci naprogramovány dva různé typy polynomů pro popis orientace až maximálně 5. stupně.

1. Polynom pro **úhel**: Předozadní úhel LEAD, boční úhel TILT vztažený na rovinu, která je definována počáteční a koncovou orientací.
2. Polynom pro **souřadnice**: XH, YH, ZH druhé prostorové křivky pro orientaci nástroje pomocí vztažného bodu na nástroji.

V případě transformace se šesti osami může být pro účely orientace nástroje naprogramováno navíc ještě i otáčení vektoru otočení THT pomocí polynomu maximálně 5. stupně pro otáčení nástroje okolo sebe sama.

Syntaxe

Polynom orientace pro typ 1 pro **úhel**

N... PO[PHI]=(a2, a3, a4, a5)

Tří- až pětiosá transformace

N... PO[PSI]=(b2, b3, b4, b5)

Polynom orientace pro typ 2 pro **souřadnice**

N... PO[XH]=(xe, x2, x3, x4, x5)

Identifikátor pro souřadnice druhé dráhy orientace pro orientaci nástroje

N... PO[YH]=(ye, y2, y3, y4, y5)

N... PO[ZH]=(ze, z2, z3, z4, z5)

Navíc může být v obou případech naprogramován polynom pro **otáčení** v případě transformace se šesti osami pomocí

N... PO[THT]=(c2, c3, c4, c5)

interpolace otáčení vztažená k dráze

nebo

absolutní, relativní a tangenciální interpolace pro změnu orientace

N... PO[THT]=(d2, d3, d4, d5)

vektoru orientace. To je možné tehdy, pokud je podporována transformace vektoru otočení s offsetem, který může být naprogramován a interpolován pomocí úhlu otočení THETA.

Význam

PO[PHI]	Úhel v rovině mezi počáteční a koncovou orientací
PO[PSI]	Popisuje úhel naklonění orientace ven z roviny mezi počáteční a koncovou orientací
PO[THT]	Úhel otočení, který vzniká otočením vektoru naprogramovaného pomocí příkazu G-funkce THETA ze skupiny 54
PHI	Předozadní úhel LEAD
PSI	Úhel bočního naklonění TILT
THETA	Otočení okolo směru nástroje v ose Z

PO [XH]	Souřadnice X vztažného bodu na nástroji
PO [YH]	Souřadnice Y vztažného bodu na nástroji
PO [ZH]	Souřadnice Z vztažného bodu na nástroji

Další informace

Polynomy orientace nemusí být naprogramovány,

- pokud jsou aktivní splinové interpolace ASPLINE, BSPLINE, CSPLINE.
Polynomy typu 1 pro úhel orientace jsou možné pro každý druh interpolace kromě splinů, tzn. při lineární interpolaci s rychlým posuvem G00, příp. s pracovním posuvem G01, při polynomické interpolaci POLY a při kruhové, příp. evolventní interpolaci s příkazy G02, G03, CIP, CT, INVCW a INCCCW.

Polynom typu 2 pro souřadnice orientace jsou oproti tomu možné jen tehdy, pokud jsou aktivní

lineární interpolace s rychlým posuvem G00, příp. s pracovním posuvem G01 nebo polynomická interpolace POLY.

- jestliže je orientace interpolována pomocí osové interpolace ORIAXES. V tomto případě mohou být přímo pomocí příkazů PO[A] a PO[B] naprogramovány polynomy pro orientační osy A a B.

Polynomy orientace typu 1 s příkazy ORIVECT, ORIPLANE a ORICONxx

V případě interpolace pomocí největší kružnice koule a interpolace plochy pláště kužele pomocí příkazů ORIVECT, ORIPLANE a ORICONxx je možné používat jediné polynomy orientace typu 1.

Polynom orientace pro typ 2 s příkazem ORICURVE

Jestliže je aktivní interpolace s doplňkovou prostorovou křivkou ORICURVE, jsou interpolovány kartézské složky vektoru orientace a tato operace je možná pouze s polynomem orientace typu 2.

7.4 Otáčení orientace nástroje (ORIROTA, ORIROT, ORIROT, ORIROT, THETA)

Jestliže má u typů obráběcích strojů s pohyblivým nástrojem existovat možnost měnit orientaci nástroje, potom se v každém bloku programuje koncová orientace. V závislosti na kinematice stroje mohou být naprogramovány buď směr orientace orientačních os nebo směr otáčení vektoru orientace THETA. Pro tyto vektory otočení mohou být naprogramovány různé druhy interpolace:

- ORIROTA: Úhel otočení vztažen ke směru otáčení zadanému absolutně.
- ORIROT: Úhel otočení relativně vůči rovině mezi počáteční a koncovou orientací.
- ORIROT: Úhel otočení relativně vůči změně vektoru orientace.
- ORIROT: Tangenciální úhel otočení k tečně dráhy.

Syntaxe

Jen tehdy, pokud je aktivován druh interpolace ORIROTA, může být úhel otočení nebo vektor otočení naprogramován jedním z následujících čtyř možných způsobů:

1. Přímé polohování kruhových os A, B, C
2. Eulerův úhel (ve stupních) pomocí A2, B2, C2
3. Úhel RPY (ve stupních) pomocí A2, B2, C2
4. Směrový vektor pomocí A3, B3, C3 (úhel otočení pomocí THETA=<hodnota>)

Pokud jsou aktivní funkce ORIROT nebo ORIROT, může být úhel otočení naprogramován ještě i přímo pomocí příkazu THETA.

Otočení může být naprogramováno také samotně v bloku, aniž by se uskutečňovala změna orientace. Příkazy ORIROT a ORIROT přitom nemají žádný význam. V tomto případě je úhel otočení vždy interpretován ve vztahu k absolutnímu směru (ORIROTA).

N... ORIROTA	Definice interpolace vektoru otočení
N... ORIROT	
N... ORIROT	
N... ORIROT	
N... A3= B3= C3=	Definice otáčení vektoru orientace
THETA=<hodnota>	
N... PO[THT]=(d ₂ , d ₃ , d ₄ , d ₅)	Interpolace úhlu otočení pomocí polynomu 5. stupně

Význam

ORIROTA:	Úhel otočení vztažen ke směru otáčení zadanému absolutně
ORIROT:	Úhel otočení relativně vůči rovině mezi počáteční a koncovou orientací
ORIROT:	Úhel otočení jako tangenciální vektor otočení ke změně orientace
ORIROT:	Úhel otočení jako tangenciální vektor otočení vůči tečně ke dráze
THETA:	Otočení vektoru orientace

7.4 Otáčení orientace nástroje (ORIROTA, ORIROT, ORIROT, ORIROT, THETA)

THETA=<hodnota>:	Úhel otočení ve stupních, který bude dosažen na konci bloku
THETA= Θ_e :	Úhel otočení s Eulerovým úhlem Θ_e tohoto vektoru
THETA=AC (...):	Blokové přepnutí na zadávání absolutních rozměrů
THETA=AC (...):	Blokové přepnutí na zadávání inkrementálních rozměrů
Θ_e :	Koncový úhel vektoru otočení buď jako absolutní rozměr, když je aktivní G90, nebo jako relativní rozměr (inkrementální), když je aktivní G91
PO[THT]=(...):	Polynom pro úhel otočení

Příklad: Otáčení orientace

Programový kód	Komentář
N10 TRAORI	; aktivování transformace orientace
N20 G1 X0 Y0 Z0 F5000	; orientace nástroje
N30 A3=0 B3=0 C3=1 THETA=0	; ve směru Z s úhlem otočení 0
N40 A3=1 B3=0 C3=0 THETA=90	; ve směru X a otočení o 90 stupňů
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	; orientace
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; ve směru Y a otočení do polohy 180 stupňů
N70 ORIROT	; zůstává konstantní a otočení do polohy 90 stupňů
N80 A3=1 B3=0 C3=0 THETA=30	; úhel otočení relativně vůči změně orientace ; vektor otočení pod úhlem 30 stupňů vůči rovině X-Y

Při interpolaci bloku N40 se úhel otočení lineárně interpoluje od počáteční hodnoty 0 stupňů do koncové hodnoty 90 stupňů. V bloku N50 se mění úhel otočení o 90 stupňů do polohy 180 stupňů podle paraboly $\theta(u) = +90u^2$. V bloku N60 může být uvedeno také otočení, aniž by se konala nějaká změna orientace.

V bloku N80 se orientace nástroje otáčí ze směru Y do směru X. Změna orientace přitom leží v rovině X-Y a vektor otočení svírá s touto rovinou úhel 30 stupňů.

Další informace

ORIROTA

Úhel otočení THETA bude interpolován vzhledem k absolutně definovanému směru v prostoru. Základní směr otočení se zadává pomocí strojních parametrů.

ORIROT

Úhel otočení THETA je interpretován vzhledem k rovině, která je definována počáteční a koncovou orientací.

ORIROT

Úhel otočení THETA je interpretován vzhledem ke změně orientace- Pro případ THETA=0 je vektor otočení interpolován tangenciálně vůči změně orientace a od situace s příkazem ORIROT se liší jen tehdy, pokud byl pro orientaci naprogramován minimálně jeden polynom pro "Úhel naklopení PSI". Z toho pak vyplývá změna orientace, která neprobíhá v dané rovině. Prostřednictvím dále naprogramovaného úhlu otočení THETA může být potom interpolován např. vektor otočení tak, že neustále zaujímá určitou hodnotu vůči změně orientace.

ORIROT

Vektor otočení je interpolován vzhledem k tečně ke dráze s offsetem naprogramovaným pomocí úhlu *THETA*. Pro úhel offsetu přitom může být naprogramovaný také polynom $PO[TH] = (c_2, c_3, c_4, c_5)$ maximálně 5. stupně.

7.5 Orientace vzhledem k dráze

7.5.1 Druhy orientace vztažené k dráze

Pomocí této rozšiřovací funkce je dosahováno orientace vztažené nejen na konec bloku, nýbrž po celé dráze. Orientace dosažená v předcházejícím bloku se pomocí interpolace pomocí největší kružnice koule převádí do naprogramované koncové orientace. V zásadě existují dvě možnosti, jak naprogramovat požadovanou orientaci vzhledem k dráze:

1. Interpolace orientace nástroje také jako otáčení nástroje pomocí příkazů ORIPATH, ORPATHTS vzhledem k dráze.
2. Vektor orientace je naprogramován a interpolován stejně jako dříve. Pomocí příkazu ORIOTC se nastavuje otočení vektoru orientace vzhledem k tečně ke dráze.

Syntaxe

Druh interpolace orientace a otočení nástroje se programují pomocí těchto příkazů:

N... ORIPATH	Orientace vzhledem k dráze
N... ORIPATHS	Orientace vzhledem ke dráze s vyhlazováním charakteristiky orientace
N... ORIOTC	Interpolace vektoru otočení vzhledem ke dráze

Skoková změna orientace vyvolaná místem, kde se na dráze nachází roh, může být vyhlazena pomocí příkazu ORIPATHS. Směr a délka dráhy pohybu při pozvednutí nástroje se programuje pomocí vektoru se složkami A8=X, B8=Y C8=Z.

Pomocí příkazů ORIPATH/ORIPATHS mohou být naprogramovány různé vztahy vůči tečně ke dráze prostřednictvím tří úhlů

- LEAD= - zadání předozadního úhlu vztaženého na dráhu a na povrch
- TILT= - zadání úhlu bočního naklopení vztaženého na dráhu a na povrch
- THETA= - úhel otočení

pro celou dráhu. Pro úhel otočení THETA je možné pomocí příkazu PO[THT]=(...) naprogramovat navíc ještě i polynom maximálně 5. stupně.

Poznámka

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje. Prostřednictvím konfigurace strojních a nastavovaných parametrů mohou být realizována ještě i další nastavení týkající se druhu orientace vztaženého k dráze. Další vysvětlení viz:

Literatura:

/FB3/, Příručka k funkcím, Speciální funkce; 3- až 5-osá transformace (F2), kapitola "Orientace".

Význam

Interpolace úhlů **LEAD** a **TILT** může být pomocí strojního parametru nastavena různě:

- Vztažený směr orientace nástroje naprogramovaný pomocí příkazů **LEAD** a **TILT** zůstává po celou dobu trvání bloku zachován.
- Předozadní úhel **LEAD**: Otočení okolo směru, který je kolmý na tečnu a normálový vektor **TILT**: Otočení orientace okolo normálového vektoru.
- Předozadní úhel **LEAD**: Otočení okolo směru, který je kolmý na tečnu a normálový vektor, úhel bočního naklopení **TILT**: Otočení orientace okolo směru tečny ke dráze.
- Úhel otočení **THETA**: Otočení nástroje okolo sebe sama s doplňkovou třetí kruhovou osou, která představuje orientační osu u transformace se šesti osami.

Poznámka

Nastavování orientace vztažené ke dráze je při použití příkazů **OSC, **OSS**, **OSSE**, **OSD**, **OST** nepřípustné**

Interpolaci orientace vztahující se ke dráze pomocí příkazů **ORIPATH**, příp. **ORIPATHS** a **ORIOTC** není možné naprogramovat spolu s vyhlazením průběhu orientace pomocí příkazů **G**-funkce ze skupiny 34. Pro tento účel musí být aktivován příkaz **OSOF**.

7.5.2

Otáčení orientace nástroje vzhledem k dráze (**ORIPATH**, **ORIPATHS**, úhel otočení)

V případě transformace se šesti osami může být orientace nástroje nastavována libovolně v prostoru a kromě toho se nástroj může pomocí třetí kruhové osy otáčet okolo sebe sama. V případě otáčení orientace nástroje pomocí příkazů **ORIPATH**, příp. **ORIPATHS** vztaženého ke dráze může být doplňkové otočení naprogramováno pomocí úhlu otočení **THETA**. Kromě toho existuje možnost naprogramovat pomocí příkazů **LEAD** a **TILT** vektor, který leží v rovině kolmé na směr nástroje.

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje. Pomocí strojního parametru může být interpolace úhlů **LEAD** a **TILT** nastavena odlišně.

Syntaxe

Otáčení orientace nástroje a nástroje samotného

Druh orientace nástroje vzhledem ke dráze se aktivuje pomocí příkazu **ORIPATH** nebo **ORIPATHS**.

N... **ORIPATH**

Aktivování druhu orientace vztaženého na dráhu

N... **ORIPATHS**

Aktivování druhu orientace vztaženého na dráhu s vyhlazením průběhu orientace

Aktivování tří možných úhlů způsobujících následující otočení:

N... **LEAD=**

Úhel pro naprogramovanou orientaci vzhledem k normálovému vektoru plochy

N... TILT=

Úhel pro naprogramovanou orientaci v rovině, která je kolmá na tečnu ke dráze vzhledem k normálovému vektoru plochy

N... THETA=

Úhel otočení třetí kruhové osy vzhledem ke změně orientace okolo směru nástroje

Hodnota úhlu na konci bloku se naprogramuje pomocí příkazů LEAD=hodnota, TILT=hodnota, příp. THETA=hodnota. Kromě konstantních úhlů mohou být pro všechny tři úhly naprogramovány polynomy maximálně 5. stupně.

N... PO[PHI]=(a2, a3, a4, a5)

Polynom pro předozadní úhel LEAD

N... PO[PSI]=(b2, b3, b4, b5)

Polynom pro úhel bočního naklopení TILT

N... PO[THT]=(d2, d3, d4, d5)

Polynom pro úhel otočení THETA

Při programování mohou být vypuštěny vyšší koeficienty polynomu, které jsou nulové. Příklad: PO[PHI]=a2 udává pro předozadní úhel LEAD parabol.

Význam

Orientace nástroje vztahující se ke dráze

ORIPATH:	Orientace nástroje vztažená na dráhu
ORIPATHS:	Orientace nástroje vztažená na dráhu, zlom v průběhu orientace se vyhladí
LEAD:	Úhel je vztažen k normálovému vektoru plochy, a svírají jej tečna dráhy a normálový vektor plochy v dané rovině
TILT:	Otočení orientace okolo směru Z, příp. otočení okolo tečny ke dráze
THETA:	Otočení okolo směru nástroje (Z)
PO[PHI]:	Polynom orientace pro předozadní úhel LEAD
PO[PSI]:	Polynom orientace pro úhel bočního naklopení TILT
PO[THT]:	Polynom orientace pro úhel otočení THETA

Poznámka

Úhel otočení THETA

Pro otočení nástroje okolo sebe sama pomocí třetí kruhové osy jako orientační osy je zapotřebí transformace se šesti osami.

7.5.3 Interpolace otočení nástroje vzhledem k dráze (ORIOTC, THETA)

Interpolace s vektory otočení

K otočení nástroje vzhledem k tečně ke dráze naprogramovanému pomocí příkazu ORIOTC může být vektor otočení interpolován s offsetem, který je možno naprogramovat pomocí úhlu

otočení THETA. Pro offsetový úhel přitom může být pomocí příkazu PO[THT] naprogramován polynom maximálně 5. stupně.

Syntaxe

N... ORIOTC

Nastavení otočení nástroje vzhledem k tečně ke dráze

N... A3= B3= C3= THETA=hodnota

Definice otáčení vektoru orientace

N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)

Interpolace úhlu offsetu pomocí polynomu 5. stupně

Otočení může být naprogramováno také samotně v bloku, aniž by se uskutečňovala změna orientace.

Význam

Interpolace otočení nástroje vztažená k dráze v případě transformace se šesti osami

ORIOTC:	Nastavení tangenciálního vektoru otočení k tečně dráhy
THETA=hodnota:	Úhel otočení ve stupních, který bude dosažen na konci bloku
THETA= θ_e :	Úhel otočení s Eulerovým úhlem Θ_e tohoto vektoru
THETA=AC (...):	Blokové přepnutí na zadávání absolutních rozměrů
THETA=IC (...):	Blokové přepnutí na zadávání inkrementálních rozměrů
PO[THT]=(c2, c3, c4, c5):	Interpolace offsetového úhlu pomocí polynomu 5. stupně

Poznámka

Interpolace vektoru otočení ORIOTC

Jestliže má být vedle směru orientace nástroje nastavováno také otočení nástroje vzhledem k tečně ke dráze, potom je pro tuto operaci nutné používat transformaci se šesti osami.

Když je aktivní příkaz ORIOTC

Vektor otočení ORIOTA nemůže být naprogramován. V případě naprogramování se aktivuje alarm 14128 "Absolutní programování otočení nástroje při aktivním příkazu ORIOTC".

Směr orientace nástroje v případě tří- až pětiosé transformace

Směr orientace nástroje může být naprogramován stejně, jak je obvyklé v případě tří- až pětiosé transformace, pomocí Eulerova úhlu, příp. pomocí úhlu RPY nebo pomocí směrových vektorů. Jsou možné také změny orientace nástroje v prostoru prostřednictvím naprogramování interpolace pomocí největší kružnice koule s příkazem ORIVECT, lineární interpolace orientačních os ORIAXES, všech interpolací na kuželové plášťové ploše ORICONxx, jakož i interpolace pomocí dvou styčných bodů nástroje pomocí příkazu ORICURVE s další prostorovou křivkou.

G....:	Zadání druhu pohybu kruhových os
X, Y, Z:	Zadání pro lineární osy

ORIAxes:	Lineární interpolace os stroje nebo orientačních os
ORIVect:	Interpolace pomocí největší kružnice koule (identická s ORIPLANE)
ORIMKS:	Otáčení v souřadném systému stroje
ORIWKS:	Otáčení v souřadném systému obrobku Popis viz kapitola Otáčení orientace nástroje
A= B= C=:	Programování poloh os stroje
ORIEULER:	Programování orientace pomocí Eulerova úhlu
ORIRPY:	Programování orientace pomocí úhlu RPY.
A2= B2= C2=:	Programování úhlů virtuálních os
ORIVIRT1:	Programování orientace pomocí virtuálních orientačních os (Definice 1), stanovení podle MD \$MC_ORIAX_TURN_TAB_1 (Definice 2), stanovení podle MD \$MC_ORIAX_TURN_TAB_2
ORIVIRT2:	
A3= B3= C3=:	Programování směrového vektoru směrové osy
ORIPLANE:	Interpolace v rovině (interpolace pomocí největší kružnice koule)
ORICONCW:	Interpolace po ploše pláště kužele ve směru hodinových ručiček
ORICONCCW:	Interpolace po ploše pláště kužele proti směru hodinových ručiček
ORICONTO:	Interpolace na ploše pláště kuželu s tangenciálním přechodem
A6= B6= C6=:	Programování osy kužele (normovaný vektor)
NUT=úhel	Úhel kužele ve stupních
NUT=+179	Úhel pohybu menší nebo roven 180 stupňům
NUT=-181	Úhel pohybu větší nebo roven 180 stupňům
ORICONIO:	Interpolace na kuželové plášťové ploše
A7= B7= C7=:	Pomocná orientace (programuje se jako normovaný vektor)
ORICURVE XH YH ZH z.B. s polynomem PO[XH]=(xe, x2, x3, x4, x5)	Interpolace orientace s udáním pohybu dvou styčných bodů nástroje. Kromě odpovídajících koncových bodů mohou být další prostorové křivky naprogramovány pomocí polynomu.

Poznámka

Pokud je orientace nástroje s aktivním příkazem ORIAxes interpolována pro orientační osy, potom je nastavení úhlu otočení vztahené ke dráze splněno jen na konci bloku.

7.5.4

Vyhlazení průběhu orientace (ORIPATHS A8=, B8=, C8=)

V případě změn orientace s konstantním zrychlením na kontuře jsou přerušeny pohyby po dráze, která se mohou vyskytnout zejména na rozích kontury, nežádoucí. Takto vznikající skoková změna na průběhu orientace může být vyhlazena vložením samostatného pomocného bloku. Jestliže je v průběhu změny orientace aktivní také příkaz ORIPATHS, změna orientace se potom uskutečňuje s konstantním zrychlením. V této fázi může být uskutečněno pozvednutí nástroje.

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje, pokud jde o případně se vyskytující předem definované strojní a nastavované parametry, s nimiž se tato funkce aktivuje.

Prostřednictvím strojního parametru může být nastaveno, jak je interpretován vektor pozvednutí.

1. V souřadném systému nástroje je souřadnice Z definována prostřednictvím směru nástroje.
2. V souřadném systému obrobku je souřadnice Z definována prostřednictvím aktivní roviny.

Pokud budete potřebovat další vysvětlení týkající se funkce "Orientace vzhledem k dráze", viz

Literatura:

Příručka k funkcím, Speciální funkce; Transformace ve více osách (F2)

Syntaxe

Aby byly orientace nástroje vztažené na celkovou dráhu spojitě, jsou zapotřebí v rozích kontury další programové údaje. Směr a délka dráhy tohoto pohybu se programují pomocí vektoru se složkami A8=X, B8=Y C8=Z.

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Význam

ORIPATHS:	Orientace nástroje vztažená na dráhu, zlom v průběhu orientace se vyhladí.
A8= B8= C8=:	Vektorové složky pro směr a délku dráhy
X, Y, Z:	Pohyb pro pozvednutí ve směru nástroje

Poznámka**Programování směrového vektoru A8, B8, C8**

Pokud je délka tohoto vektoru rovna nule, žádné pozvednutí nástroje se neuskutečňuje.

ORIPATHS

Orientace nástroje vztažená ke dráze se aktivuje pomocí příkazu ORIPATHS. Jinak je orientace interpolována od počáteční do koncové orientace prostřednictvím lineární interpolace pomocí největší kružnice koule.

7.6 Komprimování orientace (COMPON, COMPCURV, COMPCAD, COMPSURF)

NC-programy, ve kterých je aktivní transformace orientace (TRAORI) a orientace nástroje (jakéhokoli druhu) jsou naprogramovány, mohou být komprimovány, přičemž jsou dodrženy předem nastavené tolerance.

Programování

Orientace nástroje

Jestliže je aktivní transformace orientace (TRAORI), může být u strojů s 5 osami programována orientace nástroje následujícím způsobem (nezávisle na kinematice):

- Programování směrových **vektorů** prostřednictvím příkazů:
A3=<...> B3=<...> C3=<...>
- Programování **Eulerova úhlu**, příp. **úhlu RPY** prostřednictvím příkazů:
A2=<...> B2=<...> C2=<...>

Otočení nástroje

U strojů se **6 osami** je možné navíc ještě k orientaci nástroje naprogramovat také otočení nástroje.

Pro programování otočení nástroje se používá příkaz:

THETA=<...>

Viz " Otáčení orientace nástroje (Strana 358) ".

Poznámka

NC bloky, ve kterých je naprogramováno ještě i otočení, mohou být komprimovány jen tehdy, pokud se úhel otočení mění **lineárně**. Tzn. pro úhel otočení nesmí být naprogramován žádný polynom PO[THT]=(...).

Všeobecná forma komprimovatelných NC bloků

Z těchto důvodů mohou mít komprimovatelné NC bloky následující obecnou formu:

N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>

nebo

N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>

Poznámka

Hodnoty polohy mohou být zadávány buď přímo (např. X90) nebo nepřímo pomocí přiřazení parametrů (např. X=R1*(R2+R3)).

Programování orientace nástroje pomocí poloh kruhových os

Orientace nástroje může být zadávána také pomocí poloh kruhových os, např. ve formě:

N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>

V tomto případě se komprimování uskutečňuje dvěma různými způsoby v závislosti na tom, zda se provádí interpolace pomocí největší kružnice koule nebo neprovádí. Jestliže se interpolace pomocí největší kružnice koule nekoná, potom je komprimovaná změna orientace reprezentována obvyklým způsobem pomocí axiálních polynomů pro kruhové osy.

Přesnost kontury

V závislosti na nastaveném režimu komprese (MD20482 \$MC_COMPRESSOR_MODE) jsou pro geometrické osy a orientační osy při kompresi uplatňovány buď v konfiguraci nastavené tolerance pro specifické osy (MD33100 \$MA_COMPRESS_POS_TOL) nebo následující kanálové tolerance nastavitelné pomocí strojních parametrů:

SD42475 \$SC_COMPRESS_CONTUR_TOL (maximální odchylka od kontury)

SD42476 \$SC_COMPRESS_ORI_TOL (maximální úhlová odchylka pro orientaci nástroje)

SD42477 \$SC_COMPRESS_ORI_ROT_TOL (maximální úhlová odchylka pro úhel otočení nástroje (k dispozici jen u strojů se 6 osami))

Literatura:

Příručka k funkcím, Základní funkce; 3- až 5-osá transformace (F2), kapitola: "Komprimování orientace"

Aktivování / deaktivování

Funkce kompresoru se aktivují pomocí příkazů G-funkcí s modální platností COMPON, COMPCURV, COMPCAD, příp. COMPSURF.

Funkce kompresoru se příkazem COMPOF ukončí.

Viz "Komprese NC-bloků (COMPON, COMPCURV, COMPCAD) (Strana 265)".

Poznámka

Orientační pohyby budou komprimovány jen tehdy, když je aktivní interpolace pomocí největší kružnice koule (tzn. změna orientace nástroje se uskutečňuje v rovině, která je určena počáteční a koncovou orientací).

Interpolace pomocí největší kružnice koule se provádí, pokud jsou splněny následující podmínky:

- MD21104 \$MC_ORI_IPO_WITH_G_CODE = 0, ORIWKS je aktivní a orientace je naprogramována pomocí vektorů (pomocí příkazů A3, B3, C3, příp. A2, B2, C2).
 - MD21104 \$MC_ORI_IPO_WITH_G_CODE = 1 a ORIVECT, příp. ORIPLANE je aktivní. Orientace nástroje může být naprogramována buď jako směrový vektor nebo pomocí poloh kruhových os. Jestliže je aktivní některý z příkazů G-funkce ORICONxx nebo ORICURVE nebo pokud jsou naprogramovány polynomy pro úhel orientace (PO[PHI] a PO[PSI]), interpolace pomocí největší kružnice koule se neprovádí.
-

Příklad:

V následujícím příkladu programování bude komprimován kruh, který je aproximován polynomicou křivkou. Orientace nástroje se přitom pohybuje synchronně po plášti kužele. Přestože postupně za sebou naprogramované změny orientace mohou být nespojité, generuje funkce kompresoru hladký průběh orientace.

Programování	Komentář
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	; maximální odchylka od kontury = 0,05 mm
\$SC_COMPRESS_ORI_TOL=5	; maximální odchylka orientace = 5 stupňů
TRAORI	
COMPCURV	; pohyb se bude uskutečňovat po kruhu, který se skládá z mnohoúhelníků Orientace se přitom pohybuje po plášti kužele okolo osy Z a s vrcholovým úhlem 45 stupňů.
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	

7.7 Aktivování/deaktivování vyhlazování charakteristiky orientace (ORISON, ORISOF):

Funkci „Vyhlazování průběhu orientace“ je možno aktivovat/deaktivovat ve výrobním programu pomocí příkazu G-funkce ze skupiny 61. Příkazy mají modální platnost.

Předpoklady

- Systém s transformací s 5/6 osami.
- Funkce kompresoru COMPCAD je aktivní.

Syntaxe

```
ORISON
...
ORISOF
```

Význam

ORISON:	Aktivování vyhlazování průběhu orientace
ORISOF:	Deaktivování vyhlazování průběhu orientace

Příklad:

Programový kód	Komentář
...	
TRAORI()	; aktivování transformace orientace
COMPCAD	; aktivování funkce kompresoru COMPCAD
ORISON	; aktivování vyhlazování orientace
\$SC_ORISON_TOL=1.0	; maximální úhlová odchylka orientace nástroje = 1,0 stupně
G91	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
...	
ORISOF	; deaktivování vyhlazování orientace
...	

7.7 Aktivování/deaktivování vyhlazování charakteristiky orientace (ORISON, ORISOF):

Orientace bude naklopena o 90 stupňů v rovině XZ od úhlu -45 stupňů do úhlu +45 stupňů. Prostřednictvím vyhlazování průběhu orientace není už dosahováno maximálních úhlových hodnot orientace -45 stupňů, příp. +45 stupňů.

7.8 Kinematická transformace

7.8.1 Aktivování transformace čelní plochy (TRANSMIT)

Transformace čelní plochy (TRANSMIT) se ve výrobním programu nebo v synchronní akci aktivuje prostřednictvím příkazu TRANSMIT.

Syntaxe

TRANSMIT

TRANSMIT (<n>)

Význam

TRANSMIT:	Aktivování transformace TRANSMIT s prvním datovým blokem TRANSMIT
TRANSMIT (n):	Aktivování transformace TRANSMIT s <n>-tým datovým blokem TRANSMIT

Poznámka

Deaktivování aktivní transformace TRANSMIT v kanálu:

- Deaktivování transformace: TRAFOOF
- Aktivování jiné transformace: např. TRACYL, TRAANG, TRAORI

7.8.2 Aktivování transformace válcového pláště (TRACYL)

Transformace válcového pláště (TRACYL) se ve výrobním programu nebo v synchronní akci aktivuje prostřednictvím příkazu TRACYL.

Syntaxe

TRACYL (<d>)

TRACYL (<d>, <n>)

TRACYL (<d>, <n>, <k>)

Význam

TRACYL (<d>):	Aktivování transformace TRACYL s prvním datovým blokem TRACYL a pracovním průměrem <d>
TRACYL (<d>, <n>):	Aktivování transformace TRACYL s <n>-tým datovým blokem TRACYL a pracovním průměrem <d>
<d>:	Vztažný, příp. pracovní průměr Hodnota musí být větší než 1.

<n>:	Číslo datového bloku TRACYL (nepovinné)	
	Rozsah hodnot:	1, 2
<k>:	Parametr <k> má smysl pouze pro typ transformace 514	
	k = 0:	bez korekce stěny drážky
	k = 1:	s korekcí stěny drážky
	Jestliže parametr není zadán, uplatňuje se základní nastavení zadané v parametru: \$MC_TRACYL_DEFAULT_MODE_<n> kde <n> = číslo datového bloku transformace TRACYL	

Poznámka

Deaktivování aktivní transformace TRACYL v kanálu:

- Deaktivování transformace: TRAFOOF
- Aktivování jiné transformace: např. TRAANG, TRANSMIT, TRAORI

Příklad:

Programový kód	Komentář
...	
N40 TRACYL(40.)	; aktivování transformace TRACYL s prvním datovým blokem TRACYL a pracovním průměrem 40 mm
...	

Další informace

Struktura programu

Výrobní program pro frézování drážky pomocí transformace TRACYL 513 (TRACYL s korekcí stěny drážky) se zpravidla skládá z následujících kroků:

1. Volba nástroje.
2. Aktivování funkce TRACYL.
3. Aktivování vhodného posunutí počátku souřadného systému (FRAME).
4. Najíždění na pozici.
5. Naprogramování příkazu OFFN.
6. Aktivování korekce rádiusu nástroje.
7. Najížděcí blok (najíždění na korekci rádiusu nástroje a najíždění na stěnu drážky).
8. Kontura osy drážky.
9. Deaktivování korekce rádiusu nástroje.
10. Odjížděcí blok (odjíždění od korekce rádiusu nástroje a odjíždění od stěny drážky).
11. Najíždění na pozici.

12. TRAFOOF.

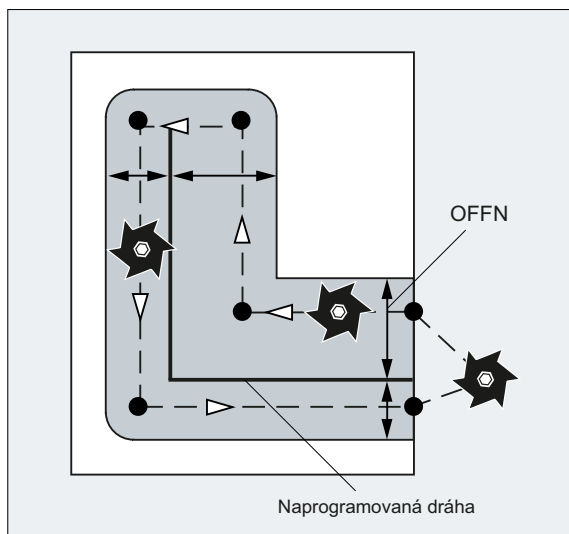
13. Opětovné aktivování původního posunutí počátku souřadného systému (FRAME).

Offset kontury (OFFN)

Abyste mohli pomocí transformace TRACYL 513 frézovat drážky, ve výrobním programu naprogramujete osu drážky a pomocí adresy OFFN **polovinu šířky drážky**.

Aby se zabránilo poškození stěny drážky, příkaz OFFN se uplatní teprve spolu s aktivovanou korekcí rádiusu nástroje.

Změny parametru OFFN v rámci výrobního programu jsou možné. Tím může být osa drážky posunuta ze středu:



Poznámka

Parametr OFFN by měl být minimálně tak velký, jako rádius nástroje, aby se vyloučilo poškození protilehlé stěny drážky.

Poznámka

OFFN s funkcí TRACYL funguje jinak než bez funkce TRACYL. Protože se OFFN započítává i bez funkce TRACYL, když je aktivní korekce rádiusu nástroje, po příkazu TRAFOOF by měla být do OFFN znovu dosazena nula.

UPOZORNĚNÍ

Působení parametru OFFN závisí na typu transformace

V případě transformace TRACYL 513 (TRACYL s korekcí stěny drážky) se do parametru OFFN naprogramuje polovina šířky drážky.

V případě transformace TRACYL 512 (TRACYL bez korekce stěny drážky) se oproti tomu hodnota zadaná do parametru OFFN uplatňuje jako přídavek rozměru ke korekci rádiusu nástroje.

Korekce rádiusu nástroje (WRK)

V případě transformace TRACYL 513 se korekce rádiusu nástroje nezapočítává vzhledem ke stěně drážky, nýbrž vůči jejímu naprogramovanému středu. Aby se nástroj pohyboval vlevo od stěny drážky, je potřeba naprogramovat příkaz G42, nikoli G41 nebo hodnotu parametru OFFN zadat se záporným znaménkem.

Průměr nástroje

Pomocí transformace TRACYL a nástroje, jehož průměr je menší než šířka drážky, není vyrobena stejná geometrie stěny drážky jako s nástrojem, jehož průměr se rovná šířce drážky. Aby se zvýšila přesnost, doporučuje se zvolit nástroj, jehož průměr je jen o málo menší, než je šířka drážky.

Využití os**Poznámka**

Následující osy nemohou být používány jako polohovací osy, příp. osy vykonávající kyvný pohyb:

- Geometrické osy v obvodovém směru plášťové plochy válce (osa Y)
- Doplňkové lineární osy při korekci stěny drážky (osa Z)

7.8.3 Aktivování transformace šikmého úhlu s naprogramovatelným úhlem (TRAANG)

Transformace šikmého úhlu s naprogramovatelným úhlem se ve výrobním programu nebo v synchronní akci aktivuje prostřednictvím příkazu TRAANG.

Syntaxe

```
TRAANG
TRAANG ( )
TRAANG ( , <n> )
TRAANG ( <α> )
TRAANG ( <α> , <n> )
```

Význam

TRAANG: TRAANG () :	Aktivování transformace TRAANG s prvním datovým blokem TRAANG a s naposled platným úhlem <α>
TRAANG (, <n>) :	Aktivování transformace TRAANG s <n>-tým datovým blokem TRAANG a s naposled platným úhlem <α>
TRAANG (<α>) :	Aktivování transformace TRAANG s prvním datovým blokem TRAANG a s úhlem <α>
TRAANG (<α> , <n>) :	Aktivování transformace TRAANG s <n>-tým datovým blokem TRAANG a s <α>

< α >:	Úhel šikmo nastavené osy (nepovinné)	
	Rozsah hodnot:	$-90^\circ < \alpha < +90^\circ$
	Pokud není zadán žádný úhel, použije se základní nastavení definované ve strojním parametru: MD2xxxx \$MC_TRAANG_ANGLE_<n>	
<n>:	Číslo datového bloku TRAANG (nepovinné)	
	Rozsah hodnot:	1, 2

Poznámka

Transformace šikmého úhlu TRAANG, která je aktivní v daném kanálu, se deaktivuje pomocí některé z těchto možností

- Deaktivování transformace: TRAFOOF
- Aktivování jiné transformace: např. TRACYL, TRANSMIT, TRAORI

Příklad:

Programový kód	Komentář
N20 TRAANG(45)	; aktivování transformace TRAANG s prvním datovým blokem TRAANG a s úhlem 45°

7.8.4 Šikmé zapichování na bruskách (G5, G7)

Příkazy G-funkcí G7 a G5 slouží pro zjednodušení programování šikmých zápichů na bruskách s transformací "Šikmá osa (TRAANG)", takže se při výrobě zápichu pohybuje pouze příslušná šikmá osa.

Naprogramovat je přitom potřeba pouze požadovanou koncovou pozici zapichovacího pohybu ve směrech X a Z. Je-li aktivován příkaz G7, je odpovídající počáteční pozice, na kterou se najíždí, přitom vypočítána NC systémem, přičemž se vychází z aktuální pozice osy X, naprogramované koncové pozice a úhlu α .

Počáteční poloha vyplývá z průsečíku dvou přímek:

- Přímka rovnoběžná s osou Z ve vzdálenosti aktuální pozice osy X
- Přímka rovnoběžná s šikmou osou a procházející naprogramovanou koncovou pozicí

Pomocí následujícího příkazu G5 se šikmou osou najíždí na naprogramovanou koncovou pozici.

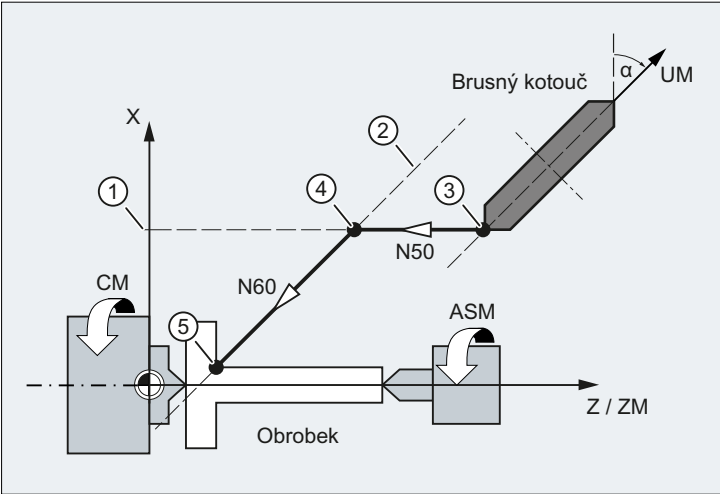
Syntaxe

```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

Význam

G7:	Výpočet počáteční pozice pro šikmé zapichování a najíždění na tuto pozici
G5:	Najíždění šikmou osou na naprogramovanou koncovou pozici
<Endpos_X>:	Koncová poloha osy X
<Endpos_Z>:	Koncová poloha osy Z

Příklad:



- ① Přímka rovnoběžná s osou Z ve vzdálenosti aktuální pozice osy X
- ② Přímka rovnoběžná s šikmou osou a procházející naprogramovanou koncovou pozicí
- ③ Výchozí pozice
- ④ Výroba zápichu: Počáteční pozice
- ⑤ Výroba zápichu: Koncová pozice
- X Geometrická osa
- Z Geometrická osa
- ZM Osa stroje
- UM Osa stroje

Programový kód	Komentář
N... G18	; aktivování roviny XZ
N40 TRAANG(45.0)	; aktivování transformace TRAANG, úhel = 45°
N50 G7 X40 Z70 F4000	; výpočet počáteční pozice a najíždění na tuto pozici
N60 G5 X40 F100	; najíždění šikmou osou na koncovou pozici
N70 ...	

7.9 Aktivování zřetězené transformace (TRACON)

Zřetězené transformace se ve výrobním programu nebo v synchronní akci aktivuje prostřednictvím příkazu TRACON.

Syntaxe

```
TRACON(<Trafo_Nr>,<Par_1>,<Par_2>,...)
...
TRAFOOF
```

Význam

TRACON:	Aktivování zřetěžené transformace Jiná dříve aktivovaná transformace se příkazem TRACON () implicitně zruší.		
<Trafo_Nr>:	Číslo zřetěžené transformace		
	Typ:	INT	
	Rozsah hodnot:	0 ... 2	
	Hodnota:	0, 1	první/jediná zřetěžená transformace
		2	druhá zřetěžená transformace
		žádný údaj	znamená totéž, co 0, příp. 1
	Poznámka: Jiné hodnoty než 0, 1, 2 budou mít za následek alarm.		
<Par_1>,<Par_2>,...:	Parametry pro vzájemně zřetěžené transformace (např. úhel šikmé osy). Jestliže tyto parametry nejsou nastaveny, uplatní se předdefinovaná nastavení nebo naposled použité parametry. Jestliže mají být pro předtím uvedené parametry uplatněna nastavení předvoleb, musí být nastavením čárek zajištěno, aby byly zadané parametry vyhodnocovány v posloupnosti, ve které jsou očekávány. Zejména v případě zadání nejméně jednoho parametru musí být před tímto parametrem uvedena čárka, a to i když zadání parametru <Trafo_Nr> není zapotřebí, například TRACON (, 3.7). U funkce TRACON s transformací TRANSMIT nebo TRAORI (opracování s 5. osou) se druhý parametr Par_2 nepoužívá jako úhel pro šikmou osu. U funkce TRACON s transformací TRACYL (obrábění plášťových ploch) se parametr Par používá pro jednotky průměru.		
TRAFOOF:	Deaktivování naposled aktivované (zřetěžené) transformace		

Příklad

Programový kód	Komentář
...	
N230 TRACON(1,45.)	; aktivování první zřetězené transformace ; dříve aktivní transformace se automaticky deaktivuje ; parametr pro šikmou osu je 45°

Programový kód	Komentář
...	
N330 TRACON(2,40.)	; aktivování druhé zřetěžené transformace ; parametr pro šikmou osu je 40°
...	
N380 TRAFOOF	; deaktivování druhé zřetěžené transformace
...	

7.10 Posuv PTP v kartézských souřadnicích

7.10.1 Aktivování/deaktivování posuvu PTP v kartézských souřadnicích (PTP, PTPG0, PTPWOC, CP)

Kartézské pohyby od bodu k bodu (PTP) je možno v NC programu aktivovat/deaktivovat pomocí příkazů z G-skupiny 49.

Příkazy mají modální platnost. Předdefinované nastavení odpovídá kartézskému pohybu po dráze (CP).

Na rozdíl od CP se při pohybech typu PTP uskutečňuje ještě navíc transformace kartézského cílového bodu a strojní osy se pohybují synchronně.

Aby bylo možno kartézský cílový bod jednoznačně přepočítat na hodnoty strojích os, jsou kromě údajů poloh a úhlů zapotřebí také informace, které popisují polohy os. Tyto údaje se zadávají prostřednictvím nastavitelných adres STAT (Strana 381) a TU (Strana 385).

Předpoklady

Transformace TRAORI, TRANSMIT, RCTRA nebo ROBX je aktivní.

Syntaxe

```
PTP / PTPG0 / PTPWOC
...
CP
```

Význam

PTP:	Aktivování pohybu od bodu k bodu (PTP) Prostřednictvím synchronního pohybu os se najede na kartézskou pozici naprogramovanou v blocích s příkazy G0 a G1.
PTPG0:	Aktivování pohybu od bodu k bodu PTPG0 Prostřednictvím synchronního pohybu os se najede na kartézskou pozici, která byla naprogramována jen v blocích s příkazy G0. V blocích s G1 se přepíná na pohyb po dráze CP.
PTPWOC:	Aktivování pohybu od bodu k bodu PTPWOC (možné pouze tehdy, když je aktivní transformace orientace) Stejně jako PTP, avšak bez kompenzačních pohybů, které jsou způsobovány pohyby kruhových os a os nastavujících orientaci.
CP:	Zrušení pohybu typu od bodu k bodu a aktivování pohybu po dráze CP Prostřednictvím příkazu CP se uskutečňuje kartézský pohyb po dráze.

Poznámka**PTPWOC**

V kombinaci s transformací RCTRA nebo ROBX nemá použití příkazu PTPWOC žádný smysl!

Příklady

Viz:

- Příklad 1: Pohyb PTP robota se 6 osami pomocí transformace ROBX (Strana 388)
- Příklad 2: Najíždění PTP při obecné 5-osé transformaci (Strana 389)
- Příklad 3: PTPG0 a TRANSMIT (Strana 389)

7.10.2 Zadávání polohy kloubu (STAT)

Údaje polohy pomocí kartézských souřadnic a údaje orientace nástroje nejsou pro jednoznačné stanovení polohy stroje postačující, protože při stejné orientaci nástroje je možný větší počet poloh kloubů. V závislosti na tom, o jakou kinematiku se jedná, existuje až 8 rozdílných poloh kloubů. Tyto odlišné polohy kloubů závisejí na konkrétní transformaci.

Aby se zabránilo nejednoznačností, zadává se poloha kloubu pomocí adresy STAT.

Poznámka

Řídící systém bere ohled na naprogramovanou polohu STAT jen při pohybech typu PTP. Při pohybech v režimu CP je ignorována, protože při pohybech s aktivní transformací není za normálních okolností změna polohy možná. Při pohybech s aktivní funkcí CP se pro cílový bod přebírá poloha platná pro počáteční bod.

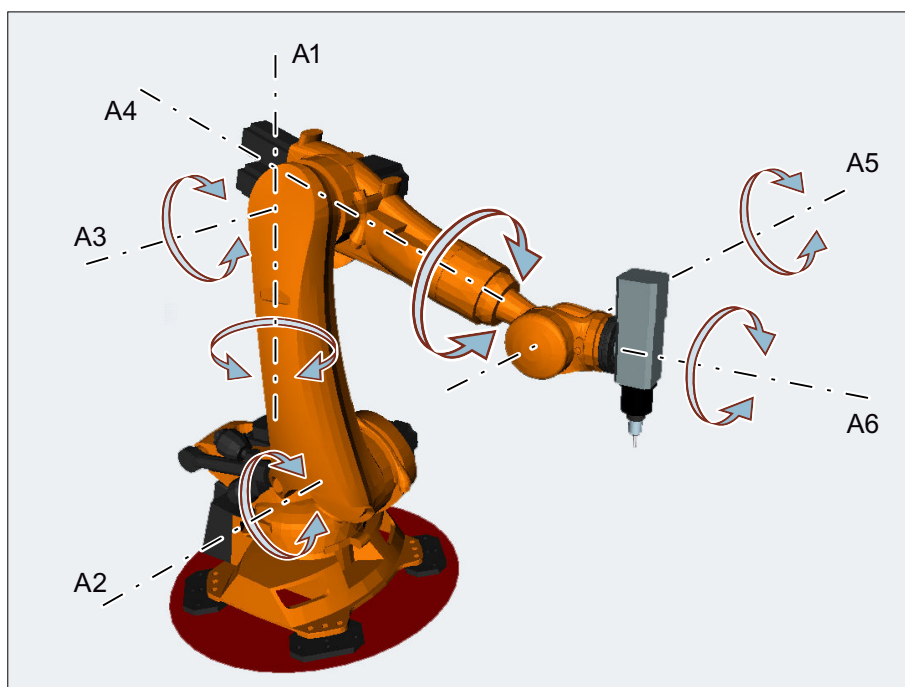
Syntaxe

STAT=<hodnota>

Význam

STAT:	Nastavitelná adresa pro zadání poloh kloubu
<hodnota>:	Binární nebo decimální hodnota Obsahuje pro každou z možných poloh jeden bit. Význam bitů je definován příslušnou transformací.

Použití příkazu STAT má být znázorněno na příkladu robota s kloubovým ramenem s frézovacím vřetenem a se 6 osami. Kinematická transformace se má uskutečňovat pomocí robotické transformace ROBX (předpoklad: Cyklus překladače „Rozšířená robotická transformace RMCC/ROBX je aktivní).

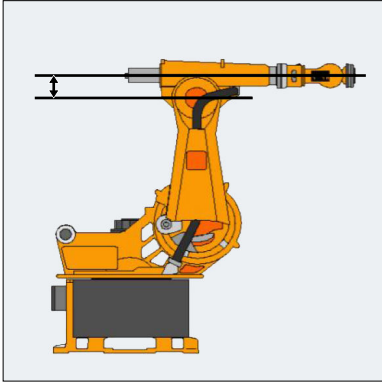


Osy A1, A2 a A3 jsou hlavními osami robota s kloubovým ramenem. Prostřednictvím hlavních os jsou v pracovním prostoru nastavovány polohy os A4, A5 a A6, které jsou označeny jako osy hlavičky nebo manuální osy. Prostřednictvím doplňkových pohybových možností manuálních os může být frézovací vřeteno nastaveno v prostoru do orientace, která je pro obráběcí úlohu zapotřebí. Při stejné orientaci nástroje jsou přitom možná různá nastavení polohy kloubů.

Pro zvolení polohy kloubů, která se použije pro obrábění, se používá programování bitů 0 ... 2 nastavitelné adresy STAT:

Bit 0	Poloha průsečíku manuálně nastavovaných os (A4 A5 A6)	
	= 0	Základní oblast (rameno vpravo) Robot se nachází v základní oblasti, jestliže vzhledem k souřadnému systému A1 je hodnota X průsečíku manuálně ovládaných os kladná.
	= 1	Obrácená oblast (rameno vlevo) Robot se nachází v obrácené oblasti, jestliže vzhledem k souřadnému systému A1 je hodnota X průsečíku manuálně ovládaných os záporná.

Příklad: Průsečík manuálních os se nachází v základní oblasti

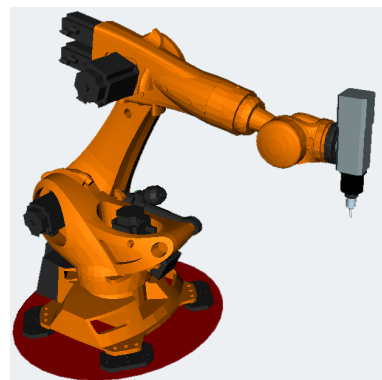
Bit 1	Poloha osy 3	
	Úhel, při kterém se hodnota bitu 1 změní, závisí na typu robota. Pro robota, jehož osa 3 a 4 se protínají, platí:	
	= 0	$A3 < 0^\circ$ (loket dolů)
	= 1	$A3 \geq 0^\circ$ (loket nahoru)
Poznámka: U robotů, které mají mezi osou 3 a osou 4 offset, je úhel, při kterém se hodnota bitu 1 změní, závislý na velikosti tohoto offsetu.		
Offset mezi A3 a A4		
Bit 2	Poloha osy 5	
	= 0	$A5 \leq 0^\circ$ (žádné obrácení konce)
	= 1	$A5 > 0^\circ$ (obrácení konce)

Příklad programování:

Programový kód	Komentář
...	
N14 T="T8MILLD20" D1	; \$TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 STAT=... F2000	; hodnota STAT určuje polohy kloubů (viz dále)
...	

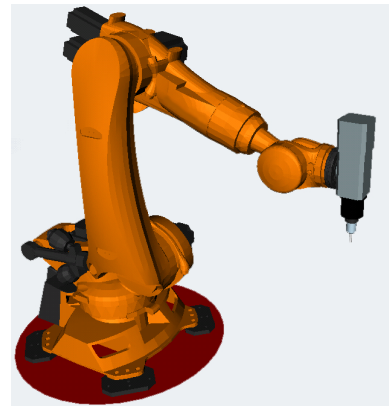
STAT=1 ('B001')

- rameno vlevo
- loket dolů
- žádné obrácení konce



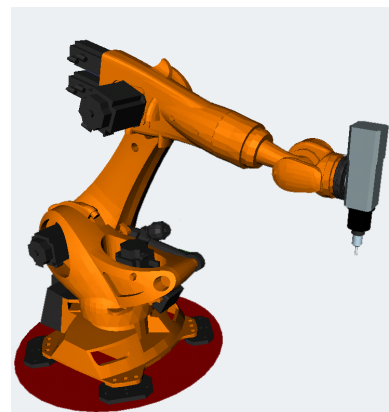
STAT=2 ('B010')

- rameno vpravo
- loket nahoru
- žádné obrácení konce



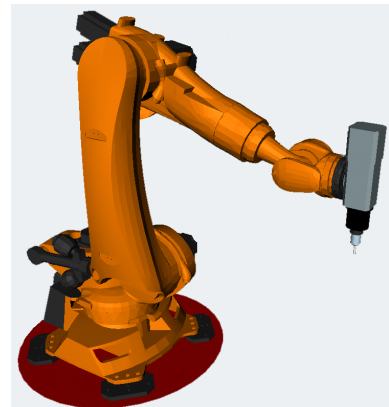
STAT=5 ('B101')

- rameno vlevo
- loket dolů
- obrácení konce



STAT=6 ('B110')

- rameno vpravo
- loket nahoru
- obrácení konce



TRANSMIT

U transformace TRANSMIT se adresa STAT používá pro odstranění nejednoznačnosti, pokud jde o polohu pólu.

Jestliže se kruhová osa musí otočit o 180° , příp. pokud by v režimu CP procházela kontura pólem, platí:

Bit 0	Má smysl jen tehdy, pokud \$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1, příp. 2:	
	= 0	Kruhová osa se pohybuje o +180°, příp. se otáčí ve směru hodinových ručiček.
	= 1	Kruhová osa se pohybuje o -180°, příp. se otáčí proti směru hodinových ručiček.
Bit 1	Má smysl jen tehdy, pokud \$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0:	
	= 0	Bude se projíždět pólem. Kruhová osa se neotáčí.
	= 1	Použije se otáčení okolo pólu. Přitom se vyhodnocuje bit 0 adresy STAT.

7.10.3 Zadání znaménka úhlu osy (TU)

Aby bylo možno u točivých os najíždět také na úhly os, které jsou větší než $+180^\circ$ nebo je menší než -180° , aniž by bylo potřeba definovat nějakou zvláštní strategii (např. pomocí vnitřního bodu), musíte prostřednictvím nastavitelné adresy TU zadat znaménko úhlu osy.

Poznámka

Řídící systém zohledňuje naprogramované hodnoty TU pouze při pohybech v režimu PTP. Při pohybech v režimu CP je tento parametr ignorován.

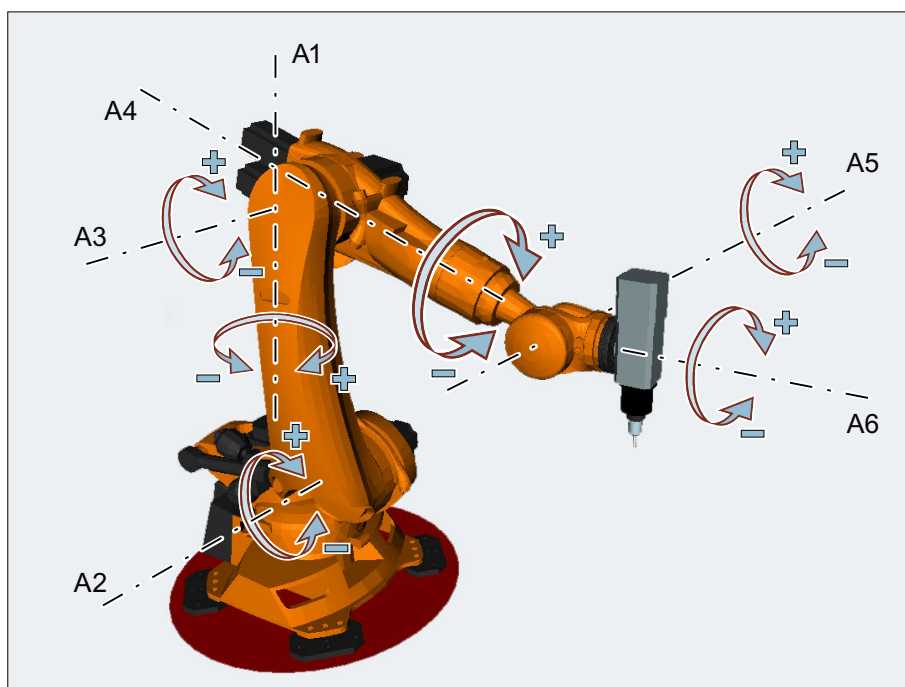
Syntaxe

TU=<hodnota>

Význam

TU:	Nastavitelná adresa pro zadání znaménka úhlu osy			
<hodnota>:	Binární nebo decimální hodnota			
	Pro každou osu, která vstupuje do transformace, existuje jeden bit, který udává znaménko úhlu osy (θ), a tím pádem také směr pohybu.			
	Bit	= 0	Znaménko úhlu osy: +	Úhlový rozsah osy: $0^\circ \leq \theta < 360^\circ$
		= 1	Znaménko úhlu osy: -	Úhlový rozsah osy: $-360^\circ < \theta < 0^\circ$

Příklad: Robot s kloubovým ramenem se 6 osami



Bit	Význam	Hodnota	Znaménko úhlu osy	Úhel osy
Bit 0 ¹⁾	Znaménko pro úhel osy A1	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
Bit 1 ¹⁾	Znaménko pro úhel osy A2	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
Bit 2 ¹⁾	Znaménko pro úhel osy A3	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
Bit 3 ¹⁾	Znaménko pro úhel osy A4	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
Bit 4 ¹⁾	Znaménko pro úhel osy A5	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
Bit 5 ¹⁾	Znaménko pro úhel osy A6	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$

¹⁾ Skutečná čísla bitů TU vyplývají z čísel kanálových os robota! V příkladu odpovídají osy robota (A1 až A6) prvním šesti osám v kanálu, v důsledku čehož se používají bity 0 ... 5 proměnné TU. V případě jiného přiřazení kanálových os osám robota by se čísla bitů proměnné TU os robota odpovídajícím způsobem změnila (např.: osy robota jsou 3. až 8. kanálová osa, tzn. pro osy robota se budou používat bity 2 ... 7 proměnné TU.)

TU=19 (odpovídá TU='B010011) by tedy znamenalo:

Bit	Hodnota		Úhel osy
0	= 1	$\Rightarrow$	$\theta_{A1} < 0^\circ$
1	= 1	$\Rightarrow$	$\theta_{A2} < 0^\circ$

2	= 0	⇒	$\theta_{A3} \geq 0^\circ$
3	= 0	⇒	$\theta_{A4} \geq 0^\circ$
4	= 1	⇒	$\theta_{A5} < 0^\circ$
5	= 0	⇒	$\theta_{A6} \geq 0^\circ$

Poznámka

U os s rozsahem pohybu $> \pm 360^\circ$ se pohyb vždy uskutečňuje po nejkratší dráze, protože pomocí informace v parametru TU není možno polohu osy jednoznačně určit.

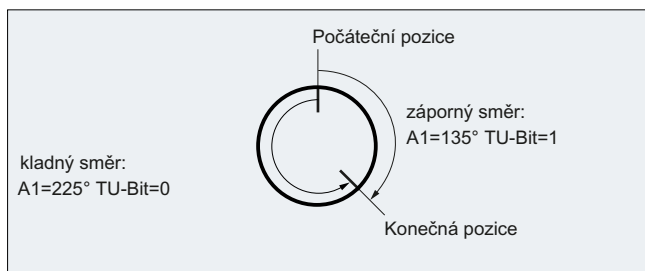
Jestliže u nějaké pozice není naprogramována žádná hodnota TU, bude se systém v závislosti na parametru MD30455 \$MA_MISC_FUNCTION_MASK nastavovat po kratší nebo po delší dráze (viz kapitola „Zohledňování SW koncových spínačů při pohybech typu PTP“ v dokumentu Příručka k funkcím, Rozšiřovací funkce).

TRANSMIT

Při pohybu v režimu PTP a při aktivní transformaci TRANSMIT nemá adresa TU žádný význam!

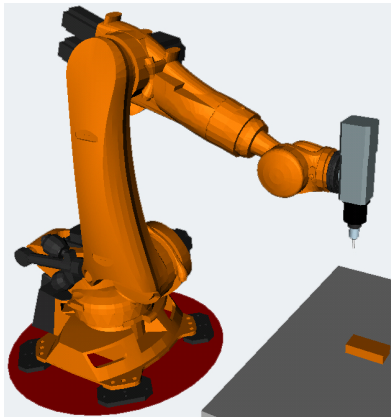
Příklad

Na pozici kruhové osy uvedenou na následujícím obrázku je možné najet v záporném nebo v kladném směru. Pomocí adresy A1 je naprogramována úhlová poloha. Teprve prostřednictvím zadání adresy TU je směr pohybu jednoznačný.



7.10.4 Příklad 1: Pohyb PTP robota se 6 osami pomocí transformace ROBX

V následujícím příkladu použití je vysvětlován kartézský pohyb PTP a s ním související příkazy NC jazyka.



Obrázek 7-3 Robot s kloubovým ramenem se 6 osami a s frézovacím vřetenem

```

N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
; $P_UIFR[1]=CTTRANS(X,1500,Y,0,Z,400):CROT(X,0,Y,0,Z,-90)
N4 G54
N5 M3 S20000
N6 ORIWKS
N7 ORIVIRT1
N8 CYCLE832(0.01,_FINISH,1)
;HOME
N9 TRAFOOF
N10 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N11 TRAORI
N12 G54
N13 G0 PTP X1369.2426 Y956.7528 Z502.5517 A=135.5761 B=-33.2223 C=161.1435
STAT='B010' TU='B001011'
N14 G0 X1355.1242 Y1014.9394 Z424.9695 A=135.8491 B=-33.1439 C=160.9941
STAT='B010' TU='B001011'
N15 G1 CP X1354.8361 Y1016.1269 Z423.3862 A=136.0635 B=-33.0819 C=160.8770
F1000
N16 G1 X1336.4283 Y1016.1269 Z426.6311 A=136.0484 B=-32.2151 C=160.9643
F2000
N17 G1 X1317.9831 Y1016.1269 Z429.6730 A=136.0175 B=-31.3394 C=161.0655
;HOME
N18 TRAFOOF
N19 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N20 M30

```


7.10.5 Příklad 2: Najíždění PTP při obecné 5-osé transformaci

Předpoklad: Základem je pravoúhlá kinematika typu CA.

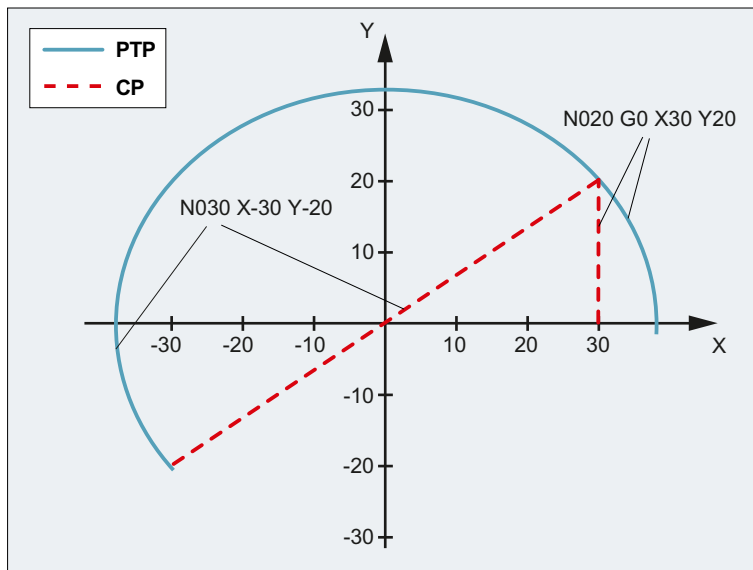
Programový kód	Komentář
TRAORI	; aktivování transformace kinematiky typu CA
PTP	; aktivování posuvu PTP
N10 A3=0 B3=0 C3=1	; polohy kruhových os C=0 A=0
N20 A3=1 B3=0 C3=1	; polohy kruhových os C=90 A=45
N30 A3=1 B3=0 C3=0	; polohy kruhových os C=90 A=90
N40 A3=1 B3=0 C3=1 STAT=1	; polohy kruhových os C=270 A=-45

Volba jednoznačně určené polohy kruhových os pro najíždění:

V bloku N40 v důsledku programového příkazu **STAT=1** přitom najíždějí kruhové osy ze svého počátečního bodu (C=90, A=90) do koncového bodu (C=270, A=-45) po delší dráze. Pokud by bylo naprogramováno **STAT=0**, kruhové osy by oproti tomu do koncového bodu (C=90, A=45) najížděly po kratší dráze.

7.10.6 Příklad 3: PTPG0 a TRANSMIT

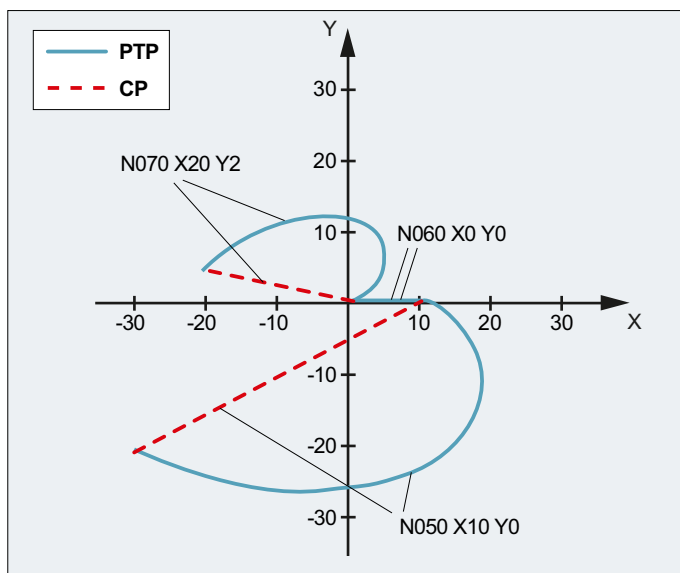
Objíždění pólu s funkcemi PTPG0 a TRANSMIT



Programový kód	Komentář
N001 G0 X30 Z0 F10000 T1 D1 G90	; výchozí pozice v absolutních rozměrech
N002 SPOS=0	
N003 TRANSMIT	; transformace TRANSMIT
N010 PTPG0	; ke každému bloku s G0 automaticky PTP a potom znovu CP

Programový kód	Komentář
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Odjíždění od pólu pomocí funkcí PTPG0 a TRANSMIT



Programování	Komentář
N001 G0 X90 Z0 F10000 T1 D1 G90	; výchozí nastavení
N002 SPOS=0	
N003 TRANSMIT	; transformace TRANSMIT
N010 PTPG0	; ke každému bloku s G0 automaticky PTP a potom znovu CP
N020 G0 X90 Y60	
N030 X-90 Y-60	
N040 X-30 Y-20	
N050 X10 Y0	
N060 X0 Y0	
N070 X-20 Y2	
N170 G1 X0 Y0	
N160 X10 Y0	
N150 X-30 Y-20	
M30	

7.11 Okrajové podmínky při aktivování transformace

Funkce

Aktivování transformací je možné pomocí výrobního programu, příp. MDA. Přitom je potřeba mít na paměti následující:

- Pomocný pohybový blok se nevkládá (fasety/rádusy).
- Posloupnost splinových bloků musí být uzavřena; pokud není, objeví se hlášení.
- Jemná korekce nástroje musí být deaktivována (FTOCOF); pokud není, objeví se hlášení.
- Korekce rádiusu nástroje musí být deaktivována (G40); pokud není, objeví se hlášení.
- Aktivovaná korekce délky nástroje je řídicím systémem přenášena do transformace.
- Aktuální frame, který byl před transformací aktivní, bude řídicím systémem deaktivován.
- Aktivní ohraničení pracovního pole jsou pro osy podílející se na transformaci řídicím systémem deaktivována (odpovídá příkazu WALOMOF).
- Monitorování chráněných oblastí je deaktivováno.
- Režim řízení pohybu po dráze a přechodová zaoblení se přeruší.
- Všechny osy uvedené ve strojním parametru musí být vzhledem k danému bloku synchronizovány.
- Vyměněné osy musí být vyměněny zpátky, pokud tomu tak není, objeví se hlášení.
- V případě závislých os se objeví hlášení.

Výměna nástroje

Výměna nástroje je přípustná, jen když je korekce rádiusu nástroje deaktivována.

Změna korekce délky nástroje a aktivování/deaktivování korekce rádiusu nástroje nesmí být naprogramovány ve stejném bloku.

Změna framu

Všechny příkazy, které se vztahují jen na základní souřadný systém, jsou přípustné (frame, korekce rádiusu nástroje). Změna framu, když je aktivní funkce G91 (inkrementální rozměry) ale není - narozdíl od situace, kdy transformace není aktivní - nijak zvlášť ošetřena. Inkrement, který se má urazit, se vyhodnocuje v souřadném systému obrobku nového framu - nezávisle na tom, který frame se uplatňoval v předcházejícím bloku.

Výjimky

Osy, které se podílejí na transformaci, se nemohou používat:

- jako osy s příkazem PRESET (alarm)
- pro najíždění na pevný bod (alarm)
- pro najíždění na referenční bod (alarm)

7.12 Deaktivování transformace (TRAFOOF)

Pomocí předem definované procedury TRAFOOF jsou všechny aktivní transformace a framy deaktivovány.

Poznámka

Pro deaktivování transformací platí stejné okrajové podmínky (Strana 391) jako pro jejich aktivování.

Framy, které jsou potom zapotřební, musí být novým naprogramováním přepnuty do aktivního stavu.

Syntaxe

```
...  
TRAFOOF
```

Význam

TRAFOOF:	Deaktivování všech aktivních transformací/framů
----------	---

Kinematické řetězce

8.1 Mazání komponent (DELOBJ)

Funkce `DELOBJ()` „vymaže“ komponenty prostřednictvím nastavení přiřazených systémových proměnných na jejich původní předdefinované hodnoty:

- Prvky kinematických řetězců
- Chráněné oblasti, prvky chráněných oblastí a dvojice oblastí chráněných kvůli kolizi
- Data transformací

Syntaxe

```
[<RetVal>=] DELOBJ (<CompType> [ , , , <NoAlarm> ] )  
[<RetVal>=] DELOBJ (<CompType> , <Index1> [ , , <NoAlarm> ] )  
[<RetVal>=] DELOBJ (<CompType> [ , <Index1> ] [ , <Index2> ] [ , <NoAlarm> ] )
```

Význam

DELOBJ:	Mazání prvků z kinematických řetězců, chráněných oblastí, prvků chráněných oblastí, dvojic oblastí chráněných kvůli kolizi a dat transformací
<CompType>:	Typ komponentu, který má být vymazán
	Datový typ: STRING
	Hodnota: "KIN_CHAIN_ELEM" Význam: Systémové proměnné pro všechny kinematické prvky: \$NK_...
	Hodnota: "KIN_CHAIN_SWITCH" Význam: Systémová proměnná \$NK_SWITCH[<i>]
	Hodnota: "KIN_CHAIN_ALL" Význam: Všechny kinematické prvky a přepínače. Stejný význam má postupné vyvolání funkce DELOBJ s hodnotami "KIN_CHAIN_ELEM" a "KIN_CHAIN_SWITCH"
	Hodnota: "PROT_AREA" Význam: Systémové proměnné chráněných oblastí: • \$NP_PROT_NAME • \$NP_CHAIN_NAME • \$NP_CHAIN_ELEM • \$NP_1ST_PROT
	Hodnota: "PROT_AREA_ELEM" Význam: Systémové proměnné prvků chráněných oblastí stroje a/nebo automatických chráněných oblastí nástroje: • \$NP_NAME • \$NP_NEXT • \$NP_NEXTP • \$NP_COLOR • \$NP_D_LEVEL • \$NP_USAGE • \$NP_TYPE • \$NP_FILENAME • \$NP_PARA • \$NP_OFF • \$NP_DIR • \$NP_ANG
	Hodnota: "PROT_AREA_COLL_PAIRS" Význam: Systémové proměnné dvojic oblastí chráněných kvůli kolizi: • \$NP_COLL_PAIR • \$NP_SAFETY_DIST
	Hodnota: "PROT_AREA_ALL" Význam: Všechny chráněné oblasti, prvky chráněných oblastí a dvojice oblastí chráněných kvůli kolizi (systémové proměnné \$NP_...) Stejný význam má postupné vyvolání funkce DELOBJ s hodnotami "PROT_AREA", "PROT_AREA_ELEM" a "PROT_AREA_COLL_PAIRS"
	Hodnota: "TRAFO_DATA" Význam: Systémové proměnné všech transformací \$NT_...

<Index1>:	Index prvního komponentu, který má být vymazán (nepovinné)	
	Datový typ:	INT
	Předdefinovaná hodnota:	-1
	Rozsah hodnot:	$-1 \leq x \leq$ (maximální počet komponent nastavený v konfiguraci -1)
	Hodnota	Význam
	0, 1, 2,	Index komponentu, který má být vymazán.
<Index2>:	Index posledního komponentu, který má být vymazán (nepovinné) Pokud <Index2> není naprogramováno, jsou vymazány pouze systémové proměnné komponentů, na které se vztahuje parametr <Index1>.	
	Datový typ:	INT
	Předdefinovaná hodnota:	Jsou vymazány pouze systémové proměnné komponentů, na které se vztahuje parametr <Index1>.
	Rozsah hodnot:	<Index1> < x ≤ (max. počet komponent nastavený v konfiguraci -1)
<NoAlarm>:	Potlačení alarmů (nepovinné)	
	Datový typ:	BOOL
	Předdefinovaná hodnota:	FALSE
	Hodnota	Význam
	FALSE	V případě chyby (<RetVal> < 0) se zpracovávání programu pozastaví a aktivuje se alarm ..
	TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití: Specifická uživatelská reakce v závislosti na výstupní hodnotě.
<RetVal>:	Výsledná hodnota funkce	
	Datový typ:	INT
	Rozsah hodnot:	0, -1, -2, ... -7
	Hodnota	Význam
	0	Nevyskytla se žádná chyba.
	-1	Volání funkce bez parametru. Musí být zadán alespoň parametr <CompType>.
	-2	<CompType> označuje neznámý komponent
	-3	<Index1> je menší než -1
	-4	<Index1> je větší než počet komponent nastavený v konfiguraci
	-5	<Index1> má při mazání skupiny komponent hodnotu nerovnající se -1
	-6	<Index2> je menší než <Index1>
	-7	<Index2> je větší než počet komponent nastavený v konfiguraci

8.2 Zjišťování indexu podle názvu (NAMETOINT)

Do polí systémových proměnných typu STRING se ukládají specifické uživatelské názvy. Na základě identifikátorů systémových proměnných a názvů zjišťuje funkce `NAMETOINT()` hodnotu indexu, která patří k danému názvu a pod kterou je v poli systémových proměnných uložen.

Syntaxe

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

Význam

NAMETOINT:	Zjištění indexu systémové proměnné	
<SysVar>:	Název systémové proměnné typu STRING	
	Datový typ:	STRING
	Rozsah hodnot:	Názvy všech systémových proměnných NC systému typu STRING
<Name>:	Řetězec znaků, příp. název, ke kterému má být zjištěn index systémové proměnné.	
	Datový typ:	STRING
<NoAlarm>:	Potlačení alarmů (nepovinné)	
	Datový typ:	BOOL
	Předdefinovaná hodnota:	FALSE
	Hodnota	Význam
	TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití: Specifická uživatelská reakce v závislosti na výstupní hodnotě.
	FALSE	V případě chyby (<RetVal> < 0) se zpracovávání programu pozastaví a aktivuje se alarm.
<RetVal>:	Index systémové proměnné nebo chybové hlášení	
	Datový typ:	INT
	Rozsah hodnot:	$-1 \leq x \leq$ (max. počet komponent nastavený v konfiguraci -1)
	Hodnota	Význam
	≥ 0	Hledaný název byl nalezen pod uvedeným indexem systémové proměnné.
	-1	Hledaný název nebyl nalezen, příp. se vyskytla chyba.

Příklad

Programový kód	Komentář
DEF INT INDEX	
\$NP_PROT_NAME[27] = "Abdeckung"	
...	

Programový kód	Komentář
INDEX = NAMETOINT("\$NP_PROT_NAME","Abdeckung")	; INDEX == 27

Zabránění vzniku kolizí u kinematických řetězců

Poznámka

Chráněné oblasti

Chráněné oblasti zmiňované v následujících kapitolách se vztahují na funkci "Geometrické modelování stroje".

Literatura:

Příručka k funkcím, Speciální funkce; kapitola "Geometrické modelování stroje"

9.1 Kontrola dvojice oblastí chráněných kvůli kolizi (COLLPAIR)

Funkce `COLLPAIR ( . . . )` zjišťuje, zda dvě chráněné oblasti netvoří kolizní pár.

Syntaxe

```
[<RetVal> =] COLLPAIR (<Name_1>, <Name_2>[, <NoAlarm>])
```

Význam

COLLPAIR:	Kontrola příslušnosti k protikolizní dvojici		
<RetVal>:	Výsledná hodnota funkce		
	Datový typ:	INT	
	Hodnota:	≥ 0	Obě chráněné oblasti tvoří protikolizní dvojici. Výsledná hodnota == Index protikolizní dvojice m (viz \$NP_COLL_PAIR)
		-1	Buď byly zadány méně než dva řetězce nebo minimálně jeden z obou těchto řetězců je prázdný.
		-2	Chráněná oblast zadaná v prvním parametru nebyla nalezena.
		-3	Chráněná oblast zadaná ve druhém parametru nebyla nalezena.
		-4	Žádná z obou zadaných chráněných oblastí nebyla nalezena.
		-5	Obě zadané chráněné oblasti byly nalezeny, netvoří ale společně protikolizní dvojici.
<Name_1>:	Název první chráněné oblasti		
	Datový typ:	STRING	
	Rozsah hodnot:	Názvy chráněných oblastí, které mohou být nastaveny v parametrech	
<Name_2>:	Název druhé chráněné oblasti		
	Datový typ:	STRING	
	Rozsah hodnot:	Názvy chráněných oblastí, které mohou být nastaveny v parametrech	
<NoAlarm>:	Potlačení alarmů (nepovinné)		
	Datový typ:	BOOL	
	Hodnota:	FALSE (předdefinované nastavení)	V případě chyby (<RetVal> < 0) se zpracovávání programu pozastaví a aktivuje se alarm.
		TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití Specifická uživatelská reakce v závislosti na výstupní hodnotě.

9.2 Vyzvání nového výpočtu modelu stroje pro prevenci kolizí (PROTA)

Jestliže jsou ve výrobním programu zapsány systémové proměnné kinematického řetězce \$NK_..., proměnné pro geometrické modelování stroje nebo pro prevenci kolizí \$NP_..., musí být potom vyzvána procedura `PROTA`, aby se změna interního modelu stroje pro prevenci kolizí uvnitř NC systému vstoupila v platnost.

Syntaxe

`PROTA [ (<Par>) ]`

Význam

PROTA:	Vyzvání nového výpočtu modelu stroje pro prevenci kolizí		
	• Spouští zastavení předběžného zpracování. • Musí se nacházet v samostatném bloku.		
<Par>:	Parametr (volitelný)		
	Datový typ:	STRING	
	Hodnota:	---	Bez parametru- Uskuteční se nový výpočet modelu stroje. Stavby chráněných oblastí zůstávají zachovány.
		"R"	Uskuteční se nový výpočet modelu stroje. Chráněné oblasti se nastaví do svého stavu po inicializaci v souladu s parametrem \$NP_INIT_STAT.

Okrajové podmínky

Simulace

Procedura `PROTA` se ve výrobních programech nesmí používat v souvislosti se simulací (`simNC`).

Příklad: Zabránění volání příkazu `PROTA`, když je aktivní simulace.

Programový kód	Komentář
...	
IF \$P_SIM == FALSE	; IF simulace není aktivní
PROTA	; THEN nově vypočítat protikolizní model
ENDIF	; ENDIF
...	

Viz také

Nastavení stavu chráněné oblasti (PROTS) (Strana 402)

9.3 Nastavení stavu chráněné oblasti (PROTS)

Procedura PROTS (. . .) nastavuje stav chráněných oblastí na zadanou hodnotu.

Syntaxe

PROTS (<State>[, <Name_1>, ..., <Name_n>])

Význam

PROTS:	Nastavení stavu chráněných oblastí		
	● Musí se nacházet v samostatném bloku.		
<State>:	Status, do kterého má být zadaná chráněná oblast nastavena		
	Datový typ:	CHAR	
	Hodnota:	"A" nebo "a"	Status: Aktivní
		"I" nebo "i"	Status: Neaktivní
		"P" nebo "p"	Status: Předběžně aktivováno, příp. řízeno PLC ¹⁾
"R" nebo "r"		Status: Interní hodnota stavu po inicializaci v NC systému ²⁾	
<Name_1> ... <Name_n>:	Název jedné nebo více chráněných oblastí, které mají být nastaveny do zadaného stavu (volitelné) Pokud není zadán žádný název, bude zadaný status nastaven pro všechny definované chráněné oblasti.		
	Datový typ:	STRING	
	Rozsah hodnot:	Názvy chráněných oblastí, které mohou být nastaveny v parametrech	
	Upozornění: Maximální počet chráněných oblastí, které mohou být zadány jako parametr, závisí pouze na maximálním možném počtu znaků na jeden řádek programu.		

¹⁾ Aktivování / deaktivování se uskutečňuje prostřednictvím: DB10.DBX234.0 - DBX241.7

²⁾ Status se nastavuje na interní hodnotu NC systému po inicializaci, tzn. na hodnotu, kterou měla systémová proměnná \$NP_INIT_STAT v okamžiku posledního vyvolání funkce PROTA(...) (Strana 401).

9.4 Stanovení vzdálenosti mezi dvěma chráněnými oblastmi (PROTD)

Funkce `PROTD ( . . . )` vypočítává vzdálenost dvou chráněných oblastí.

Vlastnosti funkce:

- Výpočet vzdálenosti se uskutečňuje nezávisle na stavu chráněné oblasti (aktivovaná, deaktivovaná, předběžně aktivovaná).
- Pro výpočet vzdálenosti dvou chráněných oblastí se používají jen takové prvky chráněných oblastí, pro které je přiřazeno `$NP_USAGE = „C“` nebo `„A“`. Prvky chráněných oblastí, které mají přiřazeno `$NP_USAGE = „V“`, nejsou uvažovány.
- Chráněné oblasti, jejichž všechny prvky mají přiřazeno `$NP_USAGE = „V“`, nemohou být do výpočtu vzdálenosti zahrnuty.
- Výpočet vzdálenosti se uskutečňuje s polohou, která byla platná na konci předcházejícího bloku.
- Superponované pohyby, které jsou ve zpracování hlavní větve započítávány, např. posunutí DRF nebo externí posunutí počátku, jsou do výpočtu vzdálenosti začleněny, přičemž se berou hodnoty, které jsou platné v **okamžiku interpretace** funkce.

Poznámka

Synchronizace

Při použití funkce `PROTD ( . . . )` je výlučnou odpovědností uživatele, aby v případě nutnosti prostřednictvím zastavení předběžného zpracování pomocí příkazu `STOPRE` zajistil synchronizaci hlavního a předběžného zpracování.

Kolize

Pokud mezi zadanými chráněnými oblastmi existuje kolize, je výsledkem funkce vzdálenost 0,0. Kolize existuje, jestliže se obě chráněné oblasti dotýkají nebo existuje jejich průnik.

Bezpečnostní vzdálenost pro kontrolu kolize (`MD10622 $MN_COLLISION_SAFETY_DIST`) se při výpočtu vzdálenosti nebere v úvahu.

Syntaxe

```
[<RetVal> =] PROTD ([<Name_1>], [<Name_2>], VAR <Vector>[, <System>])
```

Význam

PROTD:	Výpočet vzdálenosti mezi dvěma zadanými chráněnými oblastmi		
	• Musí se nacházet v samostatném bloku.		
<RetVal>:	Výsledná hodnota funkce: Absolutní hodnota vzdálenosti dvou chráněných oblastí nebo v případě kolize 0,0 (viz výše: přesah při kolizi)		
	Datový typ:	REAL	
	Rozsah hodnot:	$0,0 \leq x \leq +\text{max. hodnota REAL}$	

9.4 Stanovení vzdálenosti mezi dvěma chráněnými oblastmi (PROTD)

<Name_1>, <Name_2>:	Názvy obou chráněných oblastí, jejichž vzdálenost vůči sobě má být vypočítána (nepovinné).	
	Datový typ:	STRING
	Rozsah hodnot:	Názvy chráněných oblastí, které mohou být nastaveny v parametrech
	Předdefinovaná hodnota:	„“ (prázdný řetězec) Pokud nejsou uvedeny žádné chráněné oblasti, funkce vypočítá aktuální nejmenší vzdálenost ze všech aktivovaných a deaktivovaných chráněných oblastí, které jsou obsaženy v protikolizním modelu.
<Vector>:	Výsledná hodnota: 3-rozměrný vektor vzdálenosti od chráněné oblasti <Name_2> ke chráněné oblasti <Name_1>, přičemž:	
	• <Vector>[0]: souřadnice X v globálním souřadném systému • <Vector>[1]: Souřadnice Y v globálním souřadném systému • <Vector>[2]: Souřadnice Z v globálním souřadném systému	
	V případě kolize: <Vector> == nulový vektor	
	Datový typ:	VAR REAL [3]
<System>:	Systém měřicích jednotek (palce/metrický systém) pro vzdálenost a vektor vzdálenosti (nepovinný)	
	Datový typ:	BOOL
	Hodnota:	FALSE (předdefinované nastavení)
		TRUE
	Systém měřicích jednotek v souladu s momentálně aktivním příkazem G-funkce ze skupiny 13 (G70, G71, G700, G710)).	
	Systém měřicích jednotek odpovídá nastavenému základnímu systému: MD52806 \$MN_ISO_SCALING_SYSTEM	

Transformace pomocí kinematických řetězců

10.1 Aktivování transformace (TRAFOON)

Transformace definovaná prostřednictvím kinematického řetězce se aktivuje pomocí předem definované procedury TRAFOON. Volání se musí nacházet v samostatném bloku.

Poznámka

Další možností, kterou můžete využít, je aktivovat transformaci definovanou pomocí kinematického řetězce pomocí konvenčního příkazu jazyka, jako např. TRAORI nebo TRANSMIT. Za tím účelem musí být v systémové proměnné \$NT_TRAFO_INDEX uložena odpovídající hodnota nerovnáající se nule.

Pokud budete potřebovat další informace týkající se proměnné \$NT_TRAFO_INDEX, viz „Příručka pro sestavy, Systémové proměnné“.

Syntaxe

TRAFOON (<Trafoname>, <Diameter>, <k>)

Význam

TRAFOON:	Procedura pro aktivování transformace definované prostřednictvím kinematického řetězce	
<Trafoname>:	Název datového bloku transformace	
	Datový typ:	STRING
	Rozsah hodnot:	Všechny názvy datových bloků transformace definované pomocí proměnné \$NT_NAME
	Poznámka: Název datového bloku transformace musí být jedinečný. V proměnné \$NT_NAME se smí vyskytnout jen jednou.	
<Diameter>:	Vztažný, příp. pracovní průměr(pouze TRACYL)	
	Datový typ:	REAL
	Hodnota musí být > 1.	
<k>:	Definice použití korekce stěny drážky (jen TRACYL).	
	Datový typ:	BOOL
	Hodnota:	FALSE bez korekce stěny drážky
		TRUE s korekcí stěny drážky
	Odpovídá typu transformace TRACYL 514 (programovatelná korekce stěny drážky). Pokud parametr <k> není uveden, uplatňuje se nastavení v bitu 10 parametru \$NT_CNTRL[<n>].	

Příklad

Programový kód	Komentář
TRAFOON["Trans_1"]	Aktivování transformace s názvem Trans_1.

10.2 Modifikace transformace orientace po měření stroje (CORRTRAFO)

U strojů s transformacemi orientace, které byly definovány prostřednictvím kinematických řetězců, může uživatel používat předem definovanou funkci CORRTRAFO, která po měření stroje upravuje offsetové vektory nebo směrové vektory orientačních os v kinematickém modelu stroje.

Syntaxe

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode> [,  
<No_Alarm>])
```

Význam

CORRTRAFO:	Volání funkce
------------	---------------

10.2 Modifikace transformace orientace po měření stroje (CORRTRAFO)

<Corr_Status>:	Výsledná hodnota funkce	
	Datový typ:	INT
	Hodnoty:	0 Funkce byla zpracována bezchybně.
		1 Není aktivní žádná transformace.
		2 Momentálně aktivní transformace není transformací orientace.
		3 Aktivní transformace orientace nebyla definována prostřednictvím kinematického řetězce.
		10 Parametr <Corr_Index> sloužící při volání je záporný.
		11 Parametr <Corr_Mode> sloužící při volání je záporný.
		12 Neplatný odkaz na dílčí úsek v řetězci znaků (místo jednotek v parametru <Corr_Index>). Hodnota nesmí být větší než počet orientačních os v dílčím řetězci.
		13 Neplatný odkaz na orientační osu v dílčím řetězci (místo jednotek v parametru <Corr_Index>). Tato hodnota musí být menší než počet orientačních os v dílčím řetězci.
		14 Neplatný odkaz na řetězec znaků (místo desítek v parametru <Corr_Index>). Přípustné jsou pouze hodnoty 0 a 1 (odkaz na řetězec součástí nebo nástroje). Toto číslo chyby se vyskytne také tehdy, když dílčí řetězec, na který odkazuje parametr <Corr_Index>, neexistuje.
		15 V úseku, na který odkazuje parametr <Corr_Index>, není definován žádný prvek korekce (\$NT_CORR_ELEM_P, příp. \$NT_CORR_ELEM_T).
		20 Neplatný režim korekce (místo jednotek v parametru <Corr_Mode>). Přípustné jsou pouze hodnoty 0 a 1.
		21 Neplatný režim korekce (místo desítek a/nebo místo stovek v parametru <Corr_Mode>). Při zápisu směru osy se smí pouze místo jednotek nerovnat nule.
		30 Místo stovek parametru <Corr_Mode> je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		31 Místo tisíců parametru <Corr_Mode> je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		40 Směrový vektor, který se má převzít jako směr osy, je nulový vektor. Tato chyba se může vyskytnout jen tehdy, pokud je místo tisíců v parametru <Corr_Mode> rovno nule. Jestliže se místo tisíců v tomto parametru rovná 1 (monitorování maximální korekce je deaktivováno), je možno zapsat i nulový vektor.
		41 U korekce offsetového vektoru je odchylka oproti aktuální hodnotě u minimálně jedné souřadnice větší, než je předem stanovená maximální hodnota definovaná v nastavovaném parametru SD41610 \$SN_CORR_TRAFO_LIN_MAX. Parametr <Corr_Vect> se přepisuje chybovým vektorem. To platí i tehdy, když je zpracování přerušeno s alarmem (viz parametr <No_Alarm>).
		Chybový vektor obsahuje složky, jejichž hodnota korekce překračuje přípustnou mezní hodnotu, tedy rozdíl mezi zjištěnou hodnotou korekce a mezní hodnotou se správným znaménkem.

			Obsah složek, které svou mezní hodnotu nepřekračují, je nulový.
		42	U korekce směrového vektoru je úhlová odchylka oproti aktuálnímu směru větší, než je předem stanovená maximální hodnota definovaná v nastavovaném parametru SD41611 \$SN_CORR_TRAFO_DIR_MAX.
		43	Pokus o zápis do systémové proměnné byl kvůli chybějícímu oprávnění k zápisu zablokován.
<Corr_Vect>:	Vektor korekce Obsah vektoru korekce je definován prostřednictvím následujících parametrů <Corr_Index> a <Corr_Mode>. Pokud je <Corr_Status> = 41, obsah vektoru se přepisuje (viz výše.).		
	Datový typ:	REAL	
<Corr_Index>:	Úsek, jehož prvek korekce má být modifikován / Index orientační osy, jejíž směrový vektor má být modifikován.		
	Datový typ:	INT	
	U parametru <Corr_Index> se používá desítkové kódování (místo jednotek až desítek):		
	Místo jednotek:	Obsahuje index úseku, příp. orientační osy v dílčím řetězci.	
	Místo desítek:	Odkazuje na dílčí řetězec.	
		0x	Řetězec obrobku
1x		Řetězec nástroje	

10.2 Modifikace transformace orientace po měření stroje (CORRTrafo)

<Corr_Mode>:	Režim korekce	
	Datový typ:	INT
	U parametru <Corr_Mode> se používá desítkové kódování (místo jednotek až tisíců):	
	Místo jednotek:	Určuje, u kterého prvku má být korekce provedena.
		xxx0 Korekce lineárního offsetového vektoru
		xxx1 Korekce směrového vektoru orientační osy
	Místo desítek:	Určuje, jak má být prvek korekce, na který odkazuje obsah parametru <Corr_Index>, modifikován.
		xx0x Vektor korekce se okamžitě zapisuje do prvku korekce. Tato varianta se může použít, aby se prvek korekce zapisoval okamžitě, aniž by musel být znám index <n> příslušného systémového parametru (\$NK_OFF_DIR[<n>, ...]).
		xx1x Stejně jako u hodnoty 0, ale s tím rozdílem, že předávaná hodnota korekce je interpretována v globálních souřadnicích. Rozdíl mezi variantou 0 a 1 může vzniknout vždy, když kinematický řetězec v základním nastavení (pozice všech orientačních os se rovnají 0) obsahuje další otočení.
		xx2x Stejně jako 1, ale s tím rozdílem, že hodnota korekce se vztahuje na celý úsek, tzn. že v prvku korekce je uložena taková hodnota, že celý úsek dosahuje délky definované hodnotou korekce.
		Poznámka: Hodnoty 1 a 2 jsou při zápisu směrového vektoru orientační osy nepřipustné.
	Místo stovek:	Určuje, jak má být obsah parametru <Corr_Vect> interpretován.
		x0xx Předávaný vektor korekce <Corr_Vect> obsahuje celkovou novou délku prvku korekce, příp. úseku, na který odkazuje parametr <Corr_Index> ve spojení s místem desítek parametru <Corr_Mode> (absolutní korekce).
		x1xx Předávaný vektor korekce <Corr_Vect> obsahuje jen rozdíl oproti aktuální délce prvku korekce, příp. úseku, na který odkazuje parametr <Corr_Index> ve spojení s místem desítek parametru <Corr_Mode> (inkrementální korekce).
		Poznámka: U korekce směrového vektoru orientační osy musí místo stovek obsahovat hodnotu 0.
	Místo tisíců:	Určuje, zda má být korekce omezena prostřednictvím následující maximální hodnoty:
		- SD41610 \$SN_CORR_TRAFO_LIN_MAX nebo - SD41611 \$SN_CORR_TRAFO_DIR_MAX
		0xxx Monitorování maximální korekce je aktivní.
		1xxx Monitorování maximální korekce není aktivní.

<No_Alarm>:	Chování v případě chyby (vracená hodnota > 0) (volitelné)		
	Datový typ:	BOOL	
	Hodnota:	FALSE (předdefinované)	Pokud se vyskytne chyba, zpracování programu se pozastaví a aktivuje se alarm 14103.
		TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití: Specifická uživatelská reakce v závislosti na výstupní hodnotě.

Poznámka

Jestliže se při volání funkce vyskytne chyba, buď se aktivuje alarm nebo se na výstupu objeví číslo chyby (viz parametr <No_Alarm>), takže uživatel může sám vhodným způsobem na chybový stav reagovat. Příčinu chyby blíže popisuje alarmový parametr. Číslo chyby vrácené místo alarmu je identické s alarmovým parametrem.

Další informace k funkci CORRTRAFO

Kinematická struktura stroje s transformací orientace je popsána prostřednictvím jednoho nebo dvou kinematických řetězců (dílčí řetězce), které vycházejí z počátku globálního souřadného systému. Jeden z těchto dvou řetězců, **řetězec nástroje**, končí ve vztažném bodě nástroje, zatímco ten druhý, **řetězec obrobku**, končí v počátku základního souřadného systému.

Funkce CORRTRAFO měří délky ramen pák a směry os u strojů s transformací orientace a tyto hodnoty zapisuje do speciálních korekčních prvků. Kinematický řetězec je mimo jiné popsán prvky typu OFFSE, které jsou definovány prostřednictvím proměnné \$NK_TYPE.

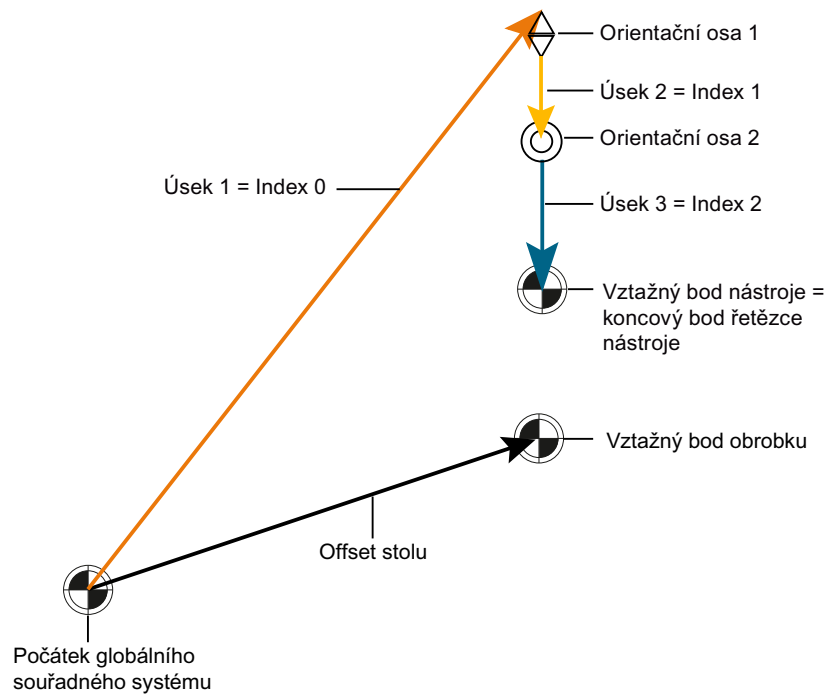
Funkce CORRTRAFO pracuje s úseky

Oba dílčí řetězce mohou být rozděleny na maximálně čtyři úseky:

- Úsek 1 začíná v počátečním bodě řetězce a končí v první ose orientace.
- Úsek 2 je vzdálenost mezi orientační osou 1 a orientační osou 2.
- Úsek 3 je vzdálenost mezi orientační osou 2 a orientační osou 3.
- Úsek 4 je vzdálenost mezi orientační osou 3 a koncem nástrojového nebo obrobkového řetězce.

Úseky mohou obsahovat příslušné konstantní prvky řetězce typu OFFSET nebo ROT_CONST.

Následující obrázek znázorňuje transformaci orientace se 2 orientačními osami.



Obrázek 10-1 CORRTrafo - příklad

Úseky jsou definovány jednoznačně: Když budete procházet kinematickým řetězcem od jeho počátečního bodu do koncového bodu, má první úsek index 0, následující úsek index 1 atd. Index posledního úseku je pak vždy roven počtu orientačních os.

Prvky korekce

Na kterýkoli konstantní prvek kinematického řetězce (prvek řetězce typu $\$NK_TYPE[<n>] = "OFFSET"$) v každém tomto úseku je možno se odkázat pomocí systémových proměnných $\$NT_CORR_ELEM_T[<n>, 0 \dots 3]$, příp. $\$NT_CORR_ELEM_P[<n>, 0 \dots 3]$. Do takto označených prvků je možno pomocí funkce CORRTrafo zapsat hodnoty korekcí, které byly zjištěny při měření stroje.

Příklad s indexem transformace = 1:

- $\$NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"$; Offset osy C (orientační osa 1) v úseku 1 je definován jako korekční prvek.
- $\$NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"$; Offset osy B (orientační osa 2) v úseku 2 je definován jako korekční prvek.
- $\$NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"$; Offset osy B až po vztažný bod nástroje v úseku 3 je definován jako korekční prvek.

Posloupnost odkazů v proměnné $\$NT_CORR_ELEM_T/P[<n>, 0 \dots 3]$ musí korespondovat s výše popsanými úseky, tzn. v proměnné $\$NT_CORR_ELEM_T/P[<n>, 0]$ se může nacházet jen prvek řetězce, který leží před první orientační osou atd.

Funkce CORRTrafo zapisuje do takto definovaných korekčních prvků hodnoty, které byly stanoveny prostřednictvím měření stroje. Úpravy hodnot korekcí jsou ve funkci CORRTrafo definovány prostřednictvím parametru $\langle Corr_Mode \rangle$.

Zavření řetězce

Jestliže je v systémové proměnné \$NT_CNTRL[<n>] nastaven bit 7 nebo bit 8, jsou na konci řetězce obrobku (bit 7), příp. před počátečním bodem řetězce nástroje (bit 8) interně automaticky vkládány doplňkové konstantní prvky řetězce, které vytvářejí spojení od koncového bodu řetězce k počátku souřadného systému stroje („zavření řetězce“).

Do těchto automaticky vkládaných prvků není možné externě nic zapisovat, lze je pouze číst (podrobnosti viz systémové proměnné \$AC_TRAFO_CORR_ELEM_P/T).

Bod pro uzavření řetězce nástroje

Jestliže systémová proměnná \$NT_CLOSE_CHAIN_T není prázdná, uzavírá se řetězec nástroje nikoli v koncovém bodě řetězce, ale v koncovém bodě popisovaného prvku řetězce. Další prvky řetězce, které se nacházejí za tímto bodem, mají potom při aktivování transformace za následek odpovídající posunutí počátku.

Index orientační osy

Kromě konstantních offsetů mezi orientačními osami je možno pomocí funkce CORRTRAFO popsat také směrové vektory orientačních os. Pod pojmem index orientační osy se v tomto případě rozumí index, který je přiřazen, když se prochází kinematickým řetězcem od jeho počátku až do konce, přičemž počítání začíná nulou. Index orientační osy je proto vždy roven indexu předcházejícího úseku.

Index orientační osy je možno také zjistit pomocí systémové proměnné \$AC_TRAFO_ORIAX_LOC.

Maximální přípustná změna prvku řetězce

Maximální přípustná změna prvku řetězce může být omezena pomocí dvou nastavovaných parametrů, a to SD41610 \$SN_CORR_TRAFO_LIN_MAX pro offsetové vektory a SD41611 \$SN_CORR_TRAFO_DIR_MAX pro směrové vektory orientačních os. SD41610 \$SN_CORR_TRAFO_LIN_MAX udává absolutní hodnotu, o kterou se každá jednotlivá složka vektoru může maximálně změnit oproti své vztažné hodnotě. SD41611 \$SN_CORR_TRAFO_DIR_MAX udává, o jaký úhel se smí maximálně změnit směr vektoru osy oproti své vztažné hodnotě. Vztažná hodnota je přitom vždy odpovídající hodnota, která je v platnosti při volání aktivní transformace s funkcí CORRTRAFO. To znamená, že obsah dat kinematiky v datové paměti, který byl eventuálně změněn po aktivování transformace, nemá žádný vliv na způsob, jakým se funkce CORRTRAFO chová.

Korekční parametry nástroje

11.1 Paměť korekcí

Struktura paměti korekčních parametrů

Každé datové pole může být vyvoláno pomocí T- a D-čísel (s výjimkou "prostého D-čísla") a tato pole obsahují vedle geometrických údajů nástroje ještě i další záznamy, např. typ nástroje.

Struktura prostých D-čísel

"Struktura prostých D-čísel" se používá tehdy, pokud je správa nástrojů realizována mimo NC systém. V tomto případě jsou vytvářena D-čísla s odpovídajícími bloky korekčních parametrů bez přiřazení určitému nástroji.

Ve výrobním programu může být naprogramováno T-slovo, to však nemá žádný vztah k naprogramovanému D-číslu.

Uživatelské parametry bříty

Uživatelské parametry bříty mohou být konfigurovány pomocí strojního parametru. Věnujte prosím pozornost informacím od výrobce stroje.

Parametry nástroje

Poznámka

Jednotlivé hodnoty v paměti korekčních parametrů

Jednotlivé hodnoty v paměti korekčních parametrů P1 až P25 je možné číst a zapisovat z programu pomocí systémových proměnných. Všechny zbývající parametry jsou rezervovány.

Korekční parametry nástroje \$TC_DP6 až \$TC_DP8, \$TC_DP10 a \$TC_DP11, jakož i \$TC_DP15 až \$TC_DP17, \$TC_DP19 a \$TC_DP20 mají v závislosti na typu nástroje odlišný význam.

Číslo parametrů nástroje (DP)	Význam systémových proměnných	Poznámka
\$TC_DP1	Typ nástroje	Přehled viz seznam
\$TC_DP2	Poloha bříty	jen pro soustružnické nástroje
Geometrie	Korekce délky	
\$TC_DP3	Délka 1	Započítává se podle
\$TC_DP4	Délka 2	typu a roviny
\$TC_DP5	Délka 3	
Geometrie	Rádus	

Číslo parametrů nástroje (DP)	Význam systémových proměnných	Poznámka
\$TC_DP6 ¹⁾ \$TC_DP6 ²⁾	Rádus 1 / délka 1 průměr d	Frézovací/soustružnické/brusné nástroje Drážková pila
\$TC_DP7 ¹⁾ \$TC_DP7 ²⁾	Délka 2 / Rohový rádus kuželové frézy, šířka drážky b, rohový rádus	Frézovací nástroje Drážková pila
\$TC_DP8 ¹⁾ \$TC_DP8 ²⁾	Rádus zaoblení 1 pro frézovací nástroje Přesah k	Frézovací nástroje Drážková pila
\$TC_DP9 ^{1) 3)}	Rádus zaoblení 2	rezervováno
\$TC_DP10 ¹⁾	Úhel 1 čelní strany nástroje	Kuželové frézovací nástroje
\$TC_DP11 ¹⁾	Úhel 2 podélné osy nástroje	Kuželové frézovací nástroje
Opatřebení	Korekce délky a rádiu	
\$TC_DP12	Délka 1	
\$TC_DP13	Délka 2	
\$TC_DP14	Délka 3	
\$TC_DP15 ¹⁾ \$TC_DP15 ²⁾	Rádus 1 / délka 1 průměr d	Frézovací/soustružnické/brusné nástroje Drážková pila
\$TC_DP16 ¹⁾ \$TC_DP16 ³⁾	Délka 2 / Rohový rádus kuželové frézy, šířka drážky b, rohový rádus	Frézovací nástroje Drážková pila
\$TC_DP17 ¹⁾ \$TC_DP17 ²⁾	Rádus zaoblení 1 pro frézovací nástroje Přesah k	Frézování / 3D frézování na čelní ploše Drážková pila
\$TC_DP18 ^{1) 3)}	Rádus zaoblení 2	rezervováno
\$TC_DP19 ¹⁾	Úhel 1 čelní strany nástroje	Kuželové frézovací nástroje
\$TC_DP20 ¹⁾	Úhel 2 podélné osy nástroje	Kuželové frézovací nástroje
Základní rozměr / adaptér	Korekce délky	
\$TC_DP21	Délka 1	
\$TC_DP22	Délka 2	
\$TC_DP23	Délka 3	
Technologie		
\$TC_DP24	Úhel volného řezání	jen pro soustružnické nástroje
\$TC_DP25		rezervováno

¹⁾ Platí i u frézovacích nástrojů pro 3D frézování na čelní ploše

²⁾ U nástroje typu drážková pila

³⁾ rezervováno (systémem SINUMERIK 840D sl není využito)

Poznámky:

Pro geometrické veličiny (např. Délka 1 nebo Rádus) existuje několik vstupních komponent. Tyto komponenty se aditivně započítávají k výsledné veličině (např. celková délka 1, celkový rádus), která se potom uplatňuje při obrábění.

Do nepotřebných parametrů nástroje je nutno dosadit nulu.

Parametry nástroje \$TC-DP1 až \$TC-DP23 u nástrojů pro obrábění kontur

Poznámka

Parametry nástroje, které v tabulce nejsou uvedeny, jako např. \$TC_DP7, nejsou vyhodnocovány, tzn. jejich obsah nemá žádný význam.

Číslo parametrů nástroje (DP)	Význam	Břity Dn		Poznámka
\$TC_DP1	Typ nástroje			400 až 599
\$TC_DP2	Poloha břitu			
Geometrie	Korekce délky			
\$TC_DP3	Délka 1			
\$TC_DP4	Délka 2			
\$TC_DP5	Délka 3			
Geometrie	Rádus			
\$TC_DP6	Rádus			
Geometrie	Mezní úhel			
\$TC_DP10	Minimální mezní úhel			
\$TC_DP11	maximální mezní úhel			
Opotřebení	Korekce délky a rádiu			
\$TC_DP12	Opotřebení – délka 1			
\$TC_DP13	Opotřebení – délka 2			
\$TC_DP14	Opotřebení – délka 3			
\$TC_DP15	Rádus opotřebení			
Opotřebení	Mezní úhel			
\$TC_DP19	Opotřebení - minimální mezní úhel			
\$TC_DP20	Opotřebení - maximální mezní úhel			
Základní rozměr / adaptér	Korekce délky			
\$TC_DP21	Délka 1			
\$TC_DP22	Délka 2			
\$TC_DP23	Délka 3			

Základní hodnota a hodnota opotřebení

Výsledné veličiny jsou vždy vypočítávány jako součet základní hodnoty a hodnoty opotřebení (např. \$TC_DP6 + \$TC_DP15 pro rádus). K délce prvního břitu nástroje se kromě toho přičítá ještě základní rozměr (\$TC_DP21 – \$TC_DP23). Kromě toho se u těchto délek nástroje uplatňují ještě i další veličiny, které i v případě běžných nástrojů mohou ovlivňovat jejich efektivní délku (adaptér, orientovatelný držák nástroje, nastavované parametry).

Mezní úhel 1 a 2

Mezní úhel 1, příp. mezní úhel 2, se vztahují vždy na vektor od středu břitu ke vztažnému bodu břitu a měří se proti směru hodinových ručiček.

11.2 Aditivní korekce

11.2.1 Aktivování aditivních korekcí (DL)

Na aditivní korekce je možné pohlížet jako na procesní korekce programovatelné v průběhu obrábění. Vztahují se na geometrické údaje o břítku a tvoří tedy součást údajů o břítech nástroje.

Údaje aditivních korekcí jsou adresována pomocí DL-čísla (DL: Location Dependent; korekce vztahená na určité místo použití) a zadávají se prostřednictvím uživatelského rozhraní.

Aplikace

Prostřednictvím aditivních korekcí mohou být kompenzovány chyby rozměrů podmíněné místem použití.

Syntaxe

DL=<číslo>

Význam

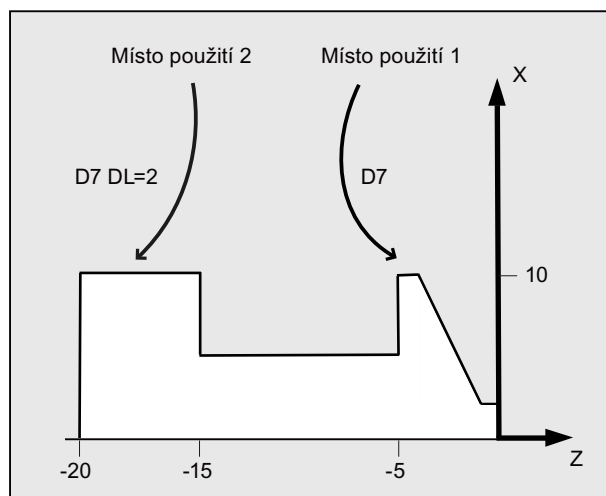
DL:	Příkaz pro aktivování aditivní korekce
<číslo>:	Prostřednictvím parametru <číslo> se zadává blok aditivních korekčních parametrů nástroje, který má být aktivován.

Poznámka

Definice počtu a aktivování aditivních korekcí se uskutečňuje prostřednictvím strojních parametrů (--> Věnujte prosím pozornost informacím od výrobce stroje!).

Příklad:

Pro 2 ložisková sedla se použije stejný břit:



Programový kód	Komentář
N110 T7 D7	; revolver se nastaví na místo č. 7 Aktivují se D7 a DL=1 a v následujícím bloku se použijí.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; aditivně k D7 se aktivuje DL=2 a v následujícím bloku se použijí
N150 G1 Z-21	
N160 G0 X200 Z200	; najíždění na bod pro výměnu nástroje
...	

11.2.2 Definice opotřebení a seřizovacích hodnot (\$TC_SCPxy[t,d], \$TC_ECPxy[t,d])

Hodnoty opotřebení a seřizovacích parametrů mohou být načítány a zapisovány pomocí systémových proměnných. Logika je přitom orientována na logiku odpovídajících systémových proměnných pro nástroje a břity.

Systémové proměnné

\$TC_SCPxy[<t>,<d>]:	Hodnoty opotřebení, které jsou prostřednictvím xy přiřazeny příslušnému geometrickému parametru, přičemž x je číslo hodnoty opotřebení a y odkaz na geometrický parametr
\$TC_ECPxy[<t>,<d>]:	Seřizovací hodnoty, které jsou prostřednictvím xy přiřazeny příslušnému geometrickému parametru, přičemž x je číslo hodnoty seřizovací a y odkaz na geometrický parametr
<t>: T-číslo nástroje	
<d>: D-číslo břitu nástroje	

Poznámka

Takto definované hodnoty opotřebení a seřizovací hodnoty s přičítají ke geometrickým parametrům a ostatním korekčním parametrům.

Příklad:

Hodnota opotřebení pro Délku 1 pro břit <d> nástroje (t) je stanovena na hodnotu 1,0.

Parametr: \$TC_DP3 (délka 1, soustružnický nástroj)

Hodnoty opotřebení: \$TC_SCP13 až \$TC_SCP63

Seřizovací hodnoty: \$TC_ECP13 až \$TC_ECP63

\$TC_SCP43 [<t>,<d>] = 1.0

11.2.3 Vymazání aditivních korekcí (DELDL)

Příkazem **DELDL** se provádí mazání aditivních korekcí břitů nástroje (vymazání z paměti). Přitom se vymažou jak definované hodnoty opotřebení, tak i seřizovací hodnoty.

Syntaxe

```
DELDL [<t>,<d>]
DELDL [<t>]
DELDL
<Status>=DELDL [<t>,<d>]
```

Význam

DELDL:	Příkaz pro vymazání aditivních korekcí	
<t>:	T-číslo nástroje	
<d>:	D-číslo břitu nástroje	
DELDL [<t>,<d>]:	Vymažou se všechny aditivní korekce břitu <d> nástroje <t>.	
DELDL [<t>]:	Vymažou se všechny aditivní korekce všech břitů nástroje <t>	
DELDL:	Budou vymazány všechny aditivní korekce všech břitů všech nástrojů jednotky TO (pro kanál, ve kterém je příkaz naprogramován).	
<Status>:	Status vymazání	
	Hodnota:	Význam:
	0	Vymazání bylo úspěšně provedeno.
	-	Vymazání nebylo provedeno (pokud nastavení parametrů označuje přesně jeden břit) nebo vymazání nebylo provedeno úplně (pokud nastavení parametrů popisuje více břitů).

Poznámka

Hodnoty opotřebení a seřizovací hodnoty aktivního nástroje není možné vymazat (analogicky k chování při mazání D-čísel a parametrů nástroje).

11.3 Korekce nástroje - speciální zacházení

Pomocí nastavovaných parametrů SD 42900 až SD 42960 je možné ovládat vyhodnocování znaménka pro délku a opotřebení nástroje.

To platí jednak pro chování složek opotřebení při zrcadlovém převrácení geometrických os, jednak při změně roviny obrábění, ale také pro teplotní kompenzaci ve směru nástroje.

Hodnoty opotřebení

Jestliže jsou v následujících odstavcích zmiňovány hodnoty opotřebení, je tím v každém případě míněn součet vlastních hodnot opotřebení (\$TC_DP12 až \$TC_DP20) a součtových korekcí s hodnotami opotřebení (\$SCPX3 až \$SCPX11) a seřizovacími hodnotami (\$ECPX3 až \$ECPX11).

Pokud budete potřebovat bližší informace o součtových korekčních parametrech, viz:

Literatura:

Příručka k funkcím, Správa nástrojů

Nastavované parametry

SD42900 \$SC_MIRROR_TOOL_LENGTH	Zrcadlové převrácení složek délky nástroje a složek základních rozměrů.
SD42910 \$SC_MIRROR_TOOL_WEAR	Zrcadlové převrácení hodnot opotřebení složky délky nástroje.
SD42920 \$SC_WEAR_SIGN_CUTPOS	Vyhodnocování znaménka složky opotřebení v závislosti na poloze bříty.
SD42930 \$SC_WEAR_SIGN	Invertování znaménka rozměru opotřebení.
SD42935 \$SC_WEAR_TRANSFORM	Transformace hodnot opotřebení.
SD42940 \$SC_TOOL_LENGTH_CONST	Přiřazení složky délky nástroje geometrické ose.
SD42950 \$SC_TOOL_LENGTH_TYPE	Přiřazení složky délky nástroje nezávisle na typu nástroje.
SD42960 \$SC_TOOL_TEMP_COMP	Hodnota teplotní kompenzace ve směru nástroje. Funguje, i když je programována orientace nástroje.

Literatura

Příručka Popis funkcí, Základní funkce; "Korekční parametry nástrojů (W1)

Další informace

Aktivování změněných nastavovaných parametrů

Nové vyhodnocení složek nástroje při změně popisovaných nastavovaných parametrů se provádí až tehdy, když je následně aktivován břit nástroje. Pokud je nějaký nástroj už aktivní a vyhodnocování změněných údajů tohoto nástroje se má stát platným, je nutné tento nástroj znovu vyvolat.

Analogický postup se použije v případě, kdy se změní výsledná délka nástroje, protože se změnil stav zrcadlového převrácení osy. Po příkazu zrcadlového převrácení musí být nástroj znovu aktivován, aby se uplatnily změněné složky délky nástroje.

Orientovatelný držák nástroje a nové nastavované parametry

Nastavované parametry SD 42900 až SD 42940 nepůsobí na složky případného aktivního orientovatelného držáku nástroje. Avšak výpočet s orientovatelným držákem nástroje vždy umožňuje pracovat s nástrojem v celé jeho výsledné délce (délka nástroje + opotřebení + rozměr základny nástroje). Při výpočtu výsledné celkové délky se zohledňují všechny změny, jež byly způsobeny nastavovanými parametry; tzn. vektory orientovatelného držáku nástroje jsou nezávislé na rovině opracování.

Poznámka

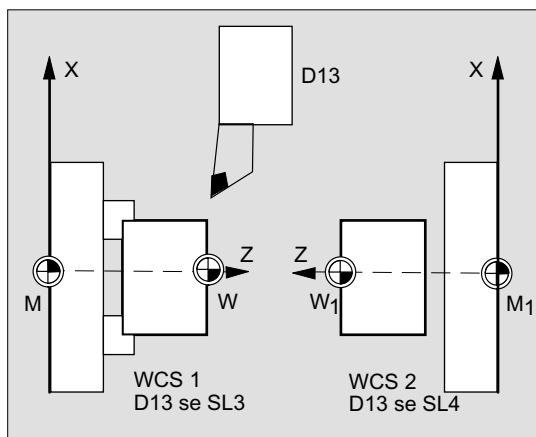
Při použití orientovatelného držáku nástroje je často užitečné definovat všechny nástroje pro základní systém bez zrcadlového převrácení a pak také ty, které se používají jenom při obrábění v zrcadlovém převrácení. Při obrábění se zrcadlově převrácenými osami se pak držák nástroje otočí tak, aby skutečná poloha nástroje byla popisována správně. Všechny složky délky nástroje jsou potom automaticky přiřazeny správným směrům, takže řídicí systém si může ušetřit vyhodnocování jednotlivých složek pomocí nastavovaných parametrů v závislosti na stavu zrcadlového převrácení jednotlivých os.

Další možnosti použití

Použití funkce orientovatelného držáku nástroje může být užitečné i tehdy, pokud stroj fyzicky nenabízí žádnou možnost nástroje otáčet do různých poloh, místo toho jsou však pevně instalovány nástroje s různými orientacemi. Při určování rozměrů nástrojů je potom možno pracovat s jednotnou základní orientací a rozměr, který je zapotřebí obrobit, vznikne příslušným otočením virtuálního držáku nástroje.

11.3.1 Zrcadlové převrácení délek nástroje

Když je nastavovaným parametrem SD42900 \$SC_MIRROR_TOOL_LENGTH a SD42910 \$SC_MIRROR_TOOL_WEAR přiřazena nenulová hodnota, můžete provádět zrcadlové převrácení složek délky nástroje a složek základního rozměru odpovídajících příslušným osám.



SD42900 \$SC_MIRROR_TOOL_LENGTH

Nastavovaný parametr se **nerovná** nule:

Zrcadlové převrácení – prostřednictvím změny znaménka – se bude týkat složek délky nástroje (\$TC_DP3, \$TC_DP4 a \$TC_DP5) a složek základních rozměrů (\$TC_DP21, \$TC_DP22 a \$TC_DP23), jejichž osy jsou zrcadlově převráceny.

Zrcadlové převrácení se **netýká** hodnot opotřebení. Pokud mají být převráceny také, musí být aktivován nastavovaný parametr SD42910 \$SC_MIRROR_TOOL_WEAR.

SD42910 \$SC_MIRROR_TOOL_WEAR

Nastavovaný parametr se **nerovná** nule:

Hodnoty opotřebení složek délky nástroje, pro něž jsou příslušné osy zrcadlově převráceny, se budou změnou znaménka zrcadlově převracet také.

11.3.2 Vyhodnocování znaménka opotřebení

Když je nastavovaným parametrem SD42920 \$SC_WEAR_SIGN_CUTPOS a SD42930 \$SC_WEAR_SIGN přiřazena nenulová hodnota, můžete invertovat vyhodnocování znaménka u složek opotřebení.

SD42920 \$SC_WEAR_SIGN_CUTPOS

Nastavovaný parametr se **nerovná** nule:

U nástrojů s relevantní polohou bříty (soustružnické a brusné nástroje – typy nástrojů 400) závisí vyhodnocování znaménka složek opotřebení v rovině obrábění na poloze bříty. U typů nástrojů bez relevantní polohy bříty nemá tento nastavovaný parametr žádný význam.

V následující tabulce jsou značkou X značeny rozměry, jejichž znaménko je nastavovaným parametrem SD42920 (různým od 0) invertováno.

Poloha bříty	Délka 1	Délka 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Poznámka

Vyhodnocování znaménka prostřednictvím parametrů SD42920 a SD42910 jsou na sobě nezávislá. Jestliže se např. změní znaménko některého rozměru pomocí obou nastavovaných parametrů, zůstane výsledné znaménko nezměněno.

SD42930 \$SC_WEAR_SIGN

Nastavovaný parametr se **nerovná nule**:

Znaménka všech rozměrů opotřebení jsou invertována. Tato změna se týká jak délky nástroje, tak i zbývajících veličin, jako jsou rádius nástroje, rádius zaoblení atd.

Jestliže je zadán kladný rozměr opotřebení, nástroj se stane „kratším“ a „tenčím“, viz "Korekční parametry nástroje – zvláštní zacházení"; kapitola „Aktivování změněných nastavovaných parametrů“.

11.3.3 Souřadný systém aktivního obrábění (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

V závislosti na kinematice stroje nebo možnosti použití orientovatelného držáku nástroje se hodnoty opotřebení změřené v jednom z těchto souřadných systémů převádějí nebo transformují do vhodného souřadného systému.

Souřadné systémy momentálně probíhajícího obrábění

Korekce délky nástroje mohou vycházet z následujících souřadných systémů, které se mohou používat pro započítávání složky délky nástroje „opotřebení“ do aktivního nástroje pomocí odpovídající skupiny příkazů G-funkcí č. 56.

- Souřadný systém stroje (MCS)
- Základní souřadný systém (BCS)
- Souřadný systém obrobku (WCS)

- Souřadný systém nástroje (TCS)
- Souřadný systém obrobku kinematické transformace (KCS)

Syntaxe

TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS

Význam

TOWSTD:	Počáteční nastavení pro korekce hodnoty opotřebení v délce nástroje
TOWMCS:	Korekce délky nástroje v MCS
TOWWCS:	Korekce délky nástroje ve WCS
TOWBCS:	Korekce délky nástroje v BCS
TOWTCS:	Korekce délky nástroje na vztažný bod držáku nástroje (orientovatelný držák nástroje)
TOWKCS:	Korekce délky nástroje na hlavu nástroje (kinematická transformace)

Další informace

Rozlišovací charakteristiky

V následující tabulce jsou shrnuty nejdůležitější rozlišovací charakteristiky:

Příkaz G-funkce	Hodnota opotřebení	Aktivní orientovatelný držák nástroje
TOWSTD:	Hodnota základního nastavení, délka nástroje	Hodnoty opotřebení podléhají otáčení.
TOWMCS:	Hodnoty opotřebení v MCS. TOWMCS a TOWSTD jsou identické, pokud není aktivní žádný orientovatelný držák nástroje.	Otáčí se pouze vektor výsledné délky nástroje bez ohledu na opotřebení.
TOWWCS:	Hodnota opotřebení ve WCS se přepočítává na MCS.	Vektor nástroje se počítá bez ohledu na opotřebení stejně jako u TOWMCS.
TOWBCS:	Hodnota opotřebení v BCS se přepočítává na MCS.	Vektor nástroje se počítá bez ohledu na opotřebení stejně jako u TOWMCS.
TOWTCS:	Hodnota opotřebení v souřadném systému nástroje se přepočítává na MCS.	Vektor nástroje se počítá bez ohledu na opotřebení stejně jako u TOWMCS.

TOWWCS, TOWBCS, TOWTCS: Vektor opotřebení se přičítá k vektoru nástroje.

Lineární transformace

Délka nástroje je v MCS smysluplně definovatelná jen tehdy, pokud MCS vychází z BCS prostřednictvím lineární transformace.

Nelineární transformace

Pokud je např. pomocí TRANSMIT aktivována nelineární transformace, pak bude-li jako požadovaný souřadný systém zadán MCS, automaticky se použije BCS.

Žádná kinematická transformace a žádný orientovatelný držák nástroje

Pokud nejsou aktivní ani kinematická transformace a ani orientovatelný držák nástroje, potom s výjimkou WCS jsou všechny čtyři zbývající souřadné systémy shodné. V důsledku toho se pouze WCS odlišuje od zbývajících. Protože se mají vyhodnocovat pouze délky nástroje, nemají posunutí mezi jednotlivými souřadnými systémy žádný význam.

Literatura:

Pokud budete potřebovat další informace týkající se korekčních parametrů nástroje, viz: Příručka k funkcím, Základní funkce; "Korekční parametry nástrojů (W1)

Započítávání hodnot opotřebení

Nastavovaný parametr **SD42935 \$SC_WEAR_TRANSFORM** definuje, které z následujících tří složek opotřebení:

- Opotřebení
- Součtová korekce jemná
- Součtová korekce hrubá

mají podléhat rotaci pomocí transformace adaptéru nebo orientovatelného držáku nástroje, pokud je aktivní některý z následujících příkazů G-funkcí.

- TOWSTD
Základní nastavení. Pro korekce hodnoty ve směru délky nástroje.
- TOWMCS
Hodnoty opotřebení v souřadném systému stroje (MCS).
- TOWWCS
Hodnoty opotřebení v souřadném systému obrobku (WCS).
- TOWBCS
Hodnoty opotřebení v základním souřadném systému (BCS).

- TOWTCS
Hodnoty opotřebení v souřadném systému nástroje na držáku nástroje (vztažný bod na držáku nástroje T).
- TOWKCS
Hodnoty opotřebení v souřadném systému hlavy nástroje při kinetické transformaci.

Poznámka

Vyhodnocování jednotlivých složek opotřebení (přiřazení geometrickým osám, vyhodnocení znaménka) je ovlivňováno následujícími faktory:

- Aktivní rovina
- Transformace adaptéru
- Nastavované parametry:
 - SD42910 \$SC_MIRROR_TOOL_WEAR
 - SD42920 \$SC_WEAR_SIGN_CUTPOS
 - SD42930 \$SC_WEAR_SIGN
 - SD42940 \$SC_TOOL_LENGTH_CONST
 - SD42950 \$SC_TOOL_LENGTH_TYPE

11.3.4 Délka nástroje a změna roviny

Když je nastavovanému parametru SD42940 \$SC_TOOL_LENGTH_CONST přiřazena nenulová hodnota, můžete jednotlivé složky délky nástroje, jako jsou délka, opotřebení a základní rozměr, přiřadit při změně roviny geometrickým osám pro soustružnické a brusné nástroje.

SD42940 \$SC_TOOL_LENGTH_CONST

Nastavovaný parametr se **nerovná nule**:

Přiřazení složek délky nástroje (délka, opotřebení a základní rozměr) geometrickým osám se při změně roviny obrábění (G17 - G19) nemění.

Následující tabulka ukazuje přiřazení složek délky nástroje geometrickým osám pro soustružnické a brusné nástroje (typ nástroje 400 – 599):

Obsah	Délka 1	Délka 2	Délka 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Každá hodnota, která se nerovná nule a ani některé z těchto šesti uváděných hodnot, bude vyhodnocena jako hodnota 18.

Následující tabulka ukazuje přiřazení složek délky nástroje geometrickým osám pro všechny ostatní nástroje (typ nástroje < 400, příp. > 599):

Rovina obrábění	Délka 1	Délka 2	Délka 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Každá hodnota, která se nerovná nule a ani některé z těchto šesti uváděných hodnot, bude vyhodnocena jako hodnota 17.

Poznámka

Při zobrazování v tabulkách se vychází z toho, že geometrické osy až 3 jsou označeny X, Y, Z. Pro přiřazení korekce ose je určující ne identifikátor osy, ale posloupnost os.

11.4 On-line korekce nástroje

11.4.1 Definice polynomické funkce (FCTDEF)

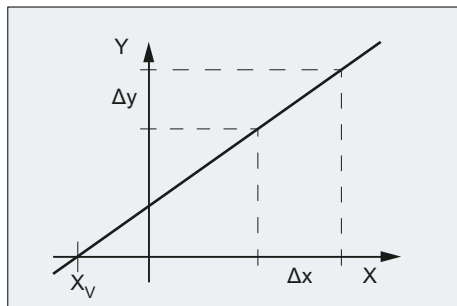
Určité strategie orovnávaní (např. orovnávací kotouč) se vyznačují tím, že brusný kotouč je spojitě (lineárně) s určitou hodnotou přísluvu orovnávacího kotouče obroben na požadovaný rádius. Za tím účelem potřebujete lineární funkci mezi hodnotou přísluvu orovnávacího kotouče a zápisem hodnoty opotřebení příslušné délky. Definice této lineární funkce se provádí pomocí předem definované procedury FCTDEF(...) pro polynomické funkce maximálně třetího stupně.

Rovnice přímky

$$y = f(x) = a_0 + a_1 \cdot x_1$$

a_1 : Směrnice přímky, kde $a_1 = \Delta x / \Delta y$

a_0 : Posunutí přímky podél osy X, kde $a_0 = -a_1 \cdot X_v$



Syntaxe

FCTDEF(<Func>, <LLimit>, <ULimit>, <a0>, <a1>, <a2>, <a3>)

Význam

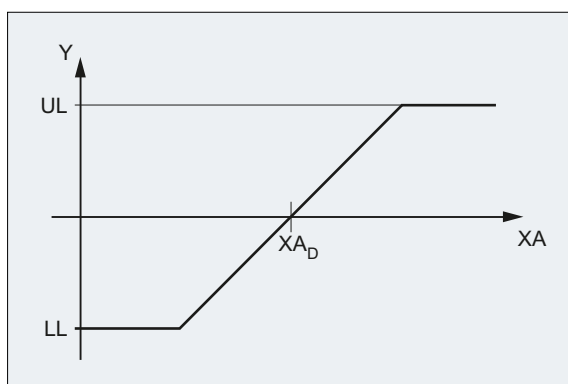
FCTDEF(...):	Definice polynomické funkce pro PUTFTOCF(...): $y = f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$	
<Func>:	Číslo funkce	
	Datový typ:	INT
	Rozsah hodnot:	1, 2, 3
<LLimit>:	Spodní mezní hodnota	
	Datový typ:	REAL
<ULimit>:	Horní mezní hodnota	
	Datový typ:	REAL
<a0>, <a1>, <a2>, <a3>:	Koeficienty polynomické funkce	
	Datový typ:	REAL

Příklad

Definice

- Číslo funkce: 1
- Horní a dolní mezní hodnota: -100, 100
- Stoupání charakteristiky: $a_1 = 1$
- Pracovní bod by se měl nacházet uprostřed charakteristiky. Za tím účelem musí být charakteristika posunuta podle požadované polohy osy XA ve WCS v okamžiku definice funkce v NC programu v záporném směru osy Y: $a_0 = -a_1 * XA_D = -1 * \AA_IW
- $a_2 = a_3 = 0$

Charakteristika



UL Horní mezní hodnota

LL Spodní mezní hodnota

XA_D Požadovaná hodnota osy XA v okamžiku definice funkce v NC programu

Programování

Programový kód	Komentář
FCTDEF(1,-100,100,-\$AA_IW[XA],1)	; definice funkce

11.4.2

Zápis on-line korekce nástroje, kontinuálně (PUTFTOCF)

Prostřednictvím předem definované procedury PUTFTOCF(...) se uskutečňuje on-line korekce nástroje podle polynomické funkce definované předtím pomocí příkazu FCTDEF(...) (Strana 430).

Poznámka

On-line korekce nástroje se může uskutečňovat také pomocí synchronní akce.

Pokud budete potřebovat další informace, viz Příručka k funkcím, Synchronní akce.

Syntaxe

```
PUTFTOCF (<Func>, <RefVal>, <ToolPar>, <Chan>, <Sp>)
```

Význam

PUTFTOCF (...):	Zápis on-line korekce nástroje, kontinuálně v bloku pomocí polynomicke funkce definované pomocí příkazu FCTDEF(...)	
<Func>:	Číslo funkce, stanovené při definici funkce pomocí příkazu FCTDEF(...)	
	Datový typ:	INT
	Rozsah hodnot:	1, 2, 3
<RefVal>:	Vztažná hodnota, od které má být korekce odvozena (např. požadovaná hodnota osy).	
	Datový typ:	VAR REAL
<ToolPar>:	Číslo parametru opotřebení (Délka 1, 2 nebo 3), do kterého se má hodnota korekce započítat	
	Datový typ:	INT
<Chan>:	Číslo kanálu, v němž má být on-line korekce nástroje v platnosti	
	Poznámka: Je zapotřebí jen tehdy, pokud se hodnota korekce nemá uplatňovat v aktivním kanálu.	
	Datový typ:	INT
<Sp>:	Číslo vřetena, pro které má být on-line korekce nástroje v platnosti	
	Poznámka: Je zapotřebí jen tehdy, pokud má být korekce uplatňována na neaktivní brusný kotouč a nikoli na nástroj, který je momentálně používán.	
	Datový typ:	INT

11.4.3 Diskrétní zápis on-line korekce nástroje (PUTFTOC)

Funkce

Prostřednictvím předem definované procedury PUTFTOC(...) se uskutečňuje on-line korekce nástroje s pevně danou hodnotou korekce.

Syntaxe

```
PUTFTOC (<CorrVal>, <ToolPar>, <Chan>, <Sp>)
```

Význam

PUTFTOC (...):	Zápis do on-line korekčních parametrů nástroje	
<CorrVal>:	Hodnota korekce, která se přičítá k parametru opotřebení	
	Datový typ:	REAL

<ToolPar>:	Číslo parametru opotřebení (Délka 1, 2 nebo 3), do kterého se má hodnota korekce započítat	
	Datový typ:	INT
<Chan>:	Číslo kanálu, v němž má být on-line korekce nástroje v platnosti Poznámka: Je zapotřebí jen tehdy, pokud se hodnota korekce nemá uplatňovat v aktivním kanálu.	
	Datový typ:	INT
<Sp>:	Číslo vřetena, pro které má být on-line korekce nástroje v platnosti Poznámka: Je zapotřebí jen tehdy, pokud má být korekce uplatňována na neaktivní brusný kotouč a nikoli na nástroj, který je momentálně používán.	
	Datový typ:	INT

11.4.4 Aktivování/deaktivování on-line korekce nástroje (FTOCON/FTOCOF)

On-line korekce nástroje se aktivuje, příp. deaktivuje pomocí příkazů G-funkcí FTOCON a FTOCOF.

Syntaxe

```
FTOCON
...
FTOCOF
```

Význam

FTOCON:	Aktivování on-line korekce nástroje Příkaz musí být naprogramován v kanálu, v němž má být on-line korekce nástroje aktivována.
FTOCOF:	Deaktivování on-line korekce nástroje Příkaz musí být naprogramován v kanálu, v němž má být on-line korekce nástroje deaktivována. Poznámka: Pomocí příkazu FTOCOF se další najíždění na korekci nástroje neuskutečňuje. V korekčních parametrech specifického bříty zůstává ale hodnota vypočítaná pomocí funkcí PUTFTOC/PUTFTOCF zachována. Aby byla on-line korekce nástroje s konečnou platností deaktivována, musí být po příkazu FTOCOF provedeno ještě aktivování/deaktivování nástroje (T...).

11.5 3D korekce rádiusu nástroje

11.5.1 Aktivování 3D korekce rádiusu nástroje pro 3D obvodové frézování (CUT3DC, CUT3DCD, ISD)

3D korekce rádiusu nástroje (3D-WRK) pro 3D obvodové frézování bez vyhodnocování omezujících ploch se aktivuje pomocí příkazů G-funkce CUT3DC, příp. CUT3DCD, s modální platností.

Pro vlastní aktivování se používají příkazy G41, příp. G42. Příkazem G40 se korekce rádiusu nástroje deaktivuje.

Syntaxe

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Význam

CUT3DC:	3D-korekce rádiusu nástroje pro obvodové frézování (jen když je aktivní 5-osá transformace)	
CUT3DCD:	3D-korekce rádiusu nástroje pro obvodové frézování vztažená na diferenční nástroj (jen když je aktivní 5-osá transformace) Rozdíl rádiusu se určuje prostřednictvím parametru nástroje \$TC_DP15.	
G41/G42 X... Y... Z...:	Aktivování korekce rádiusu nástroje	
	G41:	Korekce rádiusu nástroje vlevo od kontury
	G42:	Korekce rádiusu nástroje vpravo od kontury
	Poznámka: Aktivování se musí nacházet v bloku lineárního pohybu (G0/G1).	
CDOF2:	Deaktivování monitorování kolize pro 3D obvodové frézování	
ORIC/ORID:	Prostřednictvím příkazů G-funkcí ORIC a ORID se nastavuje chování při změnách orientace na vnějších rozích.	
	ORIC:	Změny orientace na vnějších rozích jsou superponovány s vkládaným kruhovým blokem.
	ORID:	Změny orientace na vnějších rozích jsou prováděny před vkládaným kruhovým blokem.

ISD=<hodnota>:	Prostřednictvím adresy ISD je možno při obvodovém frézování a při aktivní 3D korekci rádiusu nástroje měnit hloubku zajiždění nástroje do materiálu.	
	<hodnota>: >:	Údaj hloubky zajiždění nástroje
G40 X... Y... Z...:	Deaktivování korekce rádiusu nástroje Poznámka: Deaktivování se musí nacházet v lineárním bloku (G0/G1) s pohybem geometrických os.	

Poznámka

Příkazy G-funkcí pro aktivování 3D korekce rádiusu nástroje jsou vyhodnocovány v najížděcím bloku, tzn. typicky v bloku, který obsahuje příkaz G41 nebo G42.

Příkaz G41, příp. G42, může být naprogramován také v blocích, ve kterých geometrické osy, jichž se korekce týká, neuskutečňují žádné posuvy. V tomto případě je najížděcím blokem první blok s posuvy, který za tímto blokem následuje.

Změna varianty 3D korekce rádiusu nástroje, když je korekce rádiusu nástroje aktivní, je ignorována, přičemž se neaktivuje žádný alarm.

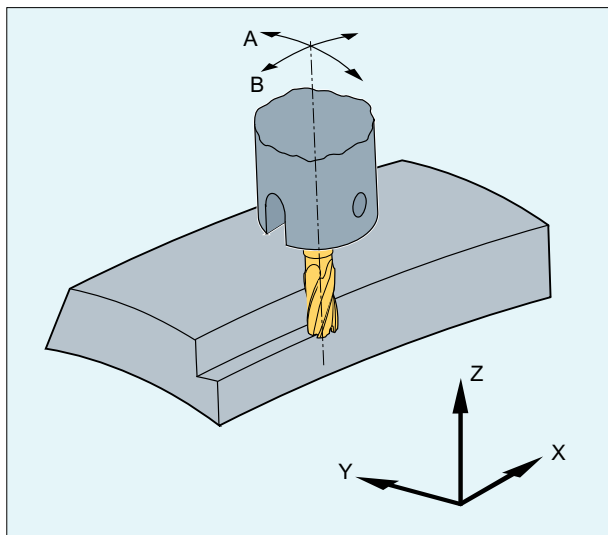
Příklad:

Programový kód	Komentář
	; Definice nástroje D1:
\$TC_DP1[1,1]=120	; Typ (stopková fréza)
\$TC_DP3[1,1]=20	; Vektor pro korekci délky
\$TC_DP6[1,1]=8	; rádius
N10 X0 Y0 Z0 T1 D1 F12000	; Aktivování nástroje.
N20 TRAORI(1)	; Aktivování transformace.
N30 G42 ORIC ISD=10 CUT3DC G64 X30	; aktivování 3D obvodového frézování
	; Spojité změny orientace na vnějších rozích
	; Hloubka zajiždění nástroje: 10 mm
N40 ORIWKS A30 B15	; Změna orientace na rohu prostřednictvím zadání polohy osy.
N50 Y20 A3=1 C3=1	; Blok posuvu se změnou orientace.
	; Zadání orientace pomocí směrového vektoru.
N60 X50 Y30	; Blok posuvu s konstantní orientací.
N70 Y50 A3=0.5 B3=1 C3=5	; Blok posuvu se změnou orientace.
N80 M63	; Blok bez informací o posuvu.
N90 X0 ISD=20	; blok posuvu se změnou hloubky zajiždění nástroje.
N100 G40 Y0	; Deaktivování korekce rádiusu nástroje.
N110 M30	

Další informace

Dráha a orientace

Zde využívaná varianta obvodového frézování je realizována zadáním dráhy (řídící křivky) a k ní odpovídající orientace. U tohoto druhu obrábění nemá tvar nástroje na dráze žádný význam. Rozhodující je pouze rádius v bodě záběru nástroje.



Obrázek 11-1 Obvodové frézování

Chování při najíždění

Pro chování při najíždění je u 3D variant korekce rádiusu nástroje vždy zvoleno NORM.

Chování na vnějších rozích

Na vnějších rozích se při obvodovém frézování s 3D korekcí rádiusu nástroje analogicky s 2½D korekcí rádiusu nástroje vyhodnocuje příkaz G-funkce ze skupiny 18 (chování korekce nástroje v rozích):

- G450: Přejížděcí prvek kruh (nástroj objíždí rohy obrobku po kruhové dráze)
Na rozdíl od řešení při 2½D korekci rádiusu nástroje je vkládaným konturovým prvkem na vnějším rohu vždy kruh s rádiusem 0, na který se uplatňuje korekce rádiusu nástroje úplně stejně, jako na kteroukoli jinou naprogramovanou dráhu. Na místě tohoto kruhu není možné vložit kuželovité průchody nástroje. Adresa DISC proto v tomto případě nemá žádný význam a nevyhodnocuje se.
- G451: Průsečík ekvidistantních drah (nástroj řeže ostré rohy obrobku)
Průsečík se určuje tak, že se offsetové křivky z obou na pohybu se podílejících bodů prodlouží a určí se jejich průsečík v rovině kolmé na orientaci nástroje v rohu.

Najíždění na průsečík (G451) se nepoužívá, jestliže je mezi podílejícími se bloky posuvů vložen minimálně jeden blok, který obsahuje změnu orientace nástroje. V takovém případě se vždy v rohu vkládá kruh.

Chování při změnách orientace na vnějších rozích

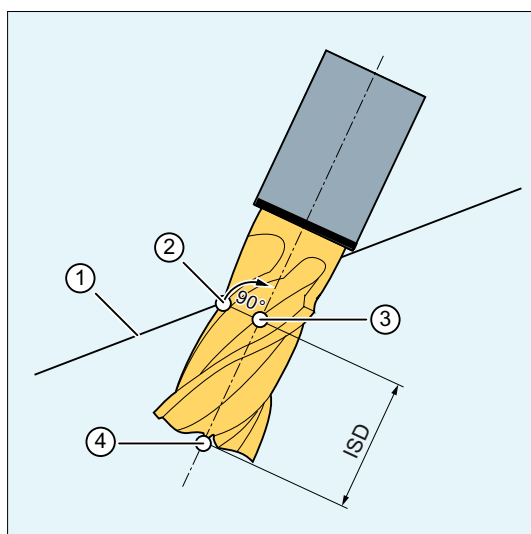
Pomocí příkazů G-funkcí ORIC a ORID může být definováno, zda se změny orientace, které byly naprogramovány mezi oběma bloky posuvů vytvářejících roh, uskuteční před zahájením zpracování vloženého bloku kruhového oblouku (ORID) nebo současně s ním (ORIC).

Hloubka zajiždění nástroje

Hloubka zajiždění frézy se rovná vzdálenosti mezi pomocným bodem frézy a špičkou nástroje.

Pomocný bod frézy je kolmá projekce obráběcího bodu frézy na naprogramovanou dráhu na podélné ose nástroje.

Prostřednictvím hloubky zajiždění nástroje je možné nastavovat polohu obráběcího bodu na ploše pláště nástroje.



- ① Naprogramovaná dráha
- ② Obráběcí bod frézy
- ③ Pomocný bod frézy
- ④ Špička frézy

ISD Hloubka zajiždění (InSertion Depth)

Obrázek 11-2 Hloubka zajiždění nástroje

Korekce nástroje vztažená na diferenční nástroj

3D korekce rádiusu nástroje pro obvodové frézování vztažená na diferenční nástroj se aktivuje prostřednictvím příkazu CUT3DCD. Je potřeba ji použít tehdy, jestliže je naprogramovaná kontura vztažena na dráhu středu normovaného nástroje a pokud se pro opracování používá jeden z nástrojů, které se od něj odlišují. Při výpočtu 3D korekce rádiusu nástroje se potom započítává pouze hodnota opotřebení rádiusu aktivního nástroje (\$TC_DP15) a v případě potřeby naprogramovaný korekční offset nástroje OFFN a TOFFR. Základní rádius (\$TC_DP6) aktivního nástroje se **nezapočítává**.

Frézování kapes se šikmými bočními stěnami pro obvodové frézování s příkazem CUT3DC

Při této 3D korekci rádiusu nástroje se uskutečňuje kompenzace odchylky rádiusu frézy tím, že se provádí přísuv ve směru normálového vektoru obráběné plochy. Jestliže je ponechána hloubka zajiždění nástroje ISD na stejné hodnotě, zůstává rovina, ve které leží čelní plocha frézy, přitom nezměněna. Fréza s např. menším rádiusem, než jaký má normovaný nástroj, by potom nedosáhla na dno kapsy, které představuje také hraniční plochu. Aby se uskutečňoval automatický přísuv nástroje, musí být řídicímu systému tyto hraniční plochy známy, viz kapitola „3D obvodové frézování se zohledněním hraniční plochy (CUT3DCC, CUT3DCCD) (Strana 444)“.

Advanced Surface / Top Surface**Poznámka**

Při použití korekce rádiusu nástroje CUT3DCD ve spojení s funkcí „Advanced Surface“ nebo „Top Surface“, pro které je nezbytná licence, je zapotřebí dodržet doporučení pro nastavení parametrů, která se týkají funkcí „Advanced Surface“ / „Top Surface“!

Pro kontrolu nastavených dat jsou k dispozici speciální zkušební programy, které si lze stáhnout přes portál SIOS:

- Zkušební programy pro funkci Advanced Surface (<https://support.industry.siemens.com/cs/ww/en/view/78956392>)
- Zkušební programy pro funkci Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

11.5.2 Aktivování 3D korekce rádiusu nástroje pro 3D frézování na čelní ploše (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

3D korekce rádiusu nástroje (3D-WRK) pro 3D frézování na čelní ploše se aktivuje pomocí příkazů G-funkcí CUT3DF, CUT3DFS, CUT3DFF, příp. CUT3DFD, s modální platností.

Pro vlastní aktivování se používají příkazy G41, příp. G42.

Aby bylo možné korekci rádiusu nástroje vypočítat, je při 3D frézování na čelní ploše zapotřebí definice normálového vektoru plochy pro rovinu, která má být opracována. To musí být uskutečněno v bloku s příkazem G41, příp. G42, pomocí adres A4, B4, C4 a A5, B5, C5.

Příkazem G40 se korekce rádiusu nástroje deaktivuje.

Syntaxe

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

Význam

CUT3DFS:	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a nemůže být ovlivňována pomocí framů.
CUT3DFF:	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a v případě potřeby může být pomocí framu pootočena do jiného směru.
CUT3DF:	3D-korekce rádiusu nástroje pro frézování na čelní straně se změnami orientace (jen když je aktivní 5-osá transformace).

CUT3DFD:	3D-korekce rádiusu nástroje pro frézování na čelní ploše vztažená na diferenční nástroj se změnami orientace (jen když je aktivní 5-osá transformace). Rozdíl rádiusu se určuje prostřednictvím parametru nástroje \$TC_DP15. Poznámka: Příkaz CUT3DFD je možný pouze ve spojení s funkcí „Vyhlazení normálového vektoru plochy při 3D frézování na čelní ploše“. Ta se aktivuje prostřednictvím volání funkce „Top Surface“, pro kterou je nezbytná licence, pomocí cyklu CYCLE832(...) (Strana 799).	
G41/G42 X... Y... Z...:	Aktivování korekce nástroje Chování v případě příkazů G41 a G42 je při 3D frézování na čelní ploše identické. Poznámka: Aktivování se musí nacházet v bloku lineárního pohybu (G0/G1).	
A4/5=... B4/5=... C4/5=...:	Definice normálového vektoru plochy pro opracovávanou rovinu	
	A4=... B4=... C4=...:	Definice makra na začátku bloku
	A5=... B5=... C5=...:	Definice makra na konci bloku
ORIC/ORID:	Prostřednictvím příkazů G-funkcí ORIC a ORID se nastavuje chování při změnách orientace na vnějších rozích.	
	ORIC:	Změny orientace na vnějších rozích jsou superponovány s vkládaným kruhovým blokem.
	ORID:	Změna orientace se bude provádět před kruhovým blokem.
G40 X... Y... Z...:	Deaktivování korekce rádiusu nástroje Poznámka: Deaktivování se musí nacházet v lineárním bloku (G0/G1) s pohybem geometrických os.	

Poznámka

Příkaz G41, příp. G42, může být naprogramován také v blocích, ve kterých geometrické osy, jichž se korekce týká, neuskutečňují žádné posuvy. V tomto případě je najížděcím blokem první blok s posuvy, který za tímto blokem následuje.

Změna varianty 3D korekce rádiusu nástroje, když je korekce rádiusu nástroje aktivní, je ignorována, přičemž se neaktivuje žádný alarm.

Příklady

Příklad 1: 3D frézování na čelní ploše s funkcí CUT3DF

Programový kód	Komentář
N10	; Definice nástroje D1:
N20 \$TC_DP1[1,1]=121	; Typ nástroje (Torus fréza)

11.5 3D korekce rádiusu nástroje

Programový kód	Komentář
N30 \$TC_DP3[1,1]=20	; -Korekce délky
N40 \$TC_DP6[1,1]=5	; rádius
N50 \$TC_DP7[1,1]=3	; Rádius zaoblení
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; Aktivování nástroje.
N90 TRAORI(1)	; Aktivování transformace orientace
N100 B4=-1 C4=1	; Definice roviny.
N110 G41 ORID CUT3DF G64 X10 Y0 Z0	; Aktivování korekce nástroje.
N120 X30	
N130 Y20 A4=1 C4=1	; Vnější rohy, nová definice roviny.
N140 B3=1 C3=5	; Změna orientace s funkcí ORID.
N150 B3=1 C3=1	; Změna orientace s funkcí ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; Změna orientace s funkcí ORIC.
N180 A3=-1 C3=1	; Změna orientace s funkcí ORIC.
N190 Y-10 A4=-1 C4=3	; Nová definice roviny.
N200 X-20 Y-20 Z10	; Vnitřní roh s předcházejícím blokem.
N210 X-30 Y10 A4=1 C4=1	; Vnitřní roh, nová definice roviny.
N220 A3=1 B3=0.5 C3=1.7	; Změna orientace s funkcí ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; Změna orientace.
N250 X0 Y30 C4=1	; Pohyb po dráze, nová rovina.
	; Orientace s programováním relativních hodnot.
N260 BSPLINE X20 Z15	; Začátek splinu, relativní programování orientace
N270 X30 Y25 Z18	; zůstává během splinu aktivní.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; Spirála, orientace s programováním os.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; Pohyb po dráze, orientace, rovina není konstantní.
N330 X-20 G40	; Deaktivování korekce rádiusu nástroje.
N340 M30	

Příklad 2: NC program vytvořený v systému CAD (pouze část) s funkcí CUT3DFD

Programový kód	Komentář
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
; Najíždění na počáteční pozici v systému MCS při zohledňování délky nástroje.	

Programový kód	Komentář
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; Konec nastavování polohy v MCS.	
;	
TRAORI(1)	; Aktivování transformace orientace
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FI- NISH, 1)	; Volání cyklu CYCLE832 s parametry: ; Tolerance kontury= 0,01 mm, ; Způsob opracování: Top Surface s vyhlazením, ; Obrábění načisto se zadáním tolerance orienta- ce, ; Tolerance orientace = 1 stupeň
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; Bloky N09 až N10 jsou rychlou částí najížděcí- ho pohybu na obrobek s konstantní orientací.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; V blocích N11 až N21 je realizována pomalá část najížděcího pohybu. Nástroj se již nachází v blízkosti obrobku a kromě toho jsou z cyklu CYCLE832 (díky aktivnímu příkazu G01) již nyní aktivní nastavení pro výrobu forem (např. COM- PSURF). Úsek dráhy této tak zvané natáčecí fáze pro správné chování při výrobě forem by měl od- povídat přibližně 1000-násobku tolerance kontury (v tomto případě tedy 10 mm).
N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000	
N13 X-247.76560 Y33.13299 Z21.73022	
N14 X-247.82755 Y32.55897 Z19.18691	
N15 X-247.87918 Y32.08062 Z17.06748	
N16 X-247.92220 Y31.68200 Z15.30129	
N17 X-247.95805 Y31.34981 Z13.82947 A3=-.216361686 B3=.934284391 C3=-.283373071	
N18 X-247.98792 Y31.07299 Z12.60295 A3=-.216360662 B3=.934280801 C3=-.283385691	
N19 X-248.01282 Y30.84230 Z11.58085 A3=-.216336015 B3=.934194446 C3=-.283689030	
N20 X-248.03357 Y30.65006 Z10.72910 A3=-.216233089 B3=.933833626 C3=-.284952647	
N21 X-248.05086 Y30.48986 Z10.01931 A5=-.060687572 B5=.974940255 C5=-.214029243 A3=-.215712821 B3=.932005189 C3=-.291263295	

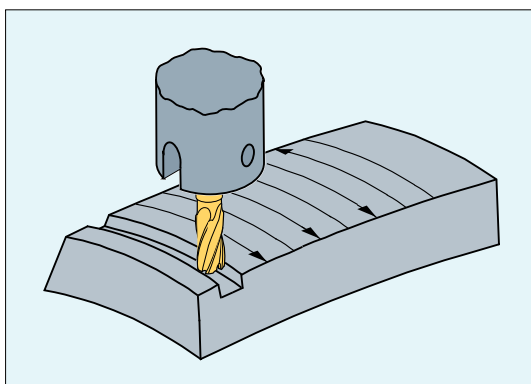
Programový kód	Komentář
N22 G41 X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; Od bloku N22 je normálový vektor plochy poprvé definován v rámci celého bloku (tzn. normálový vektor plochy na konci předcházejícího bloku N21 je již k dispozici, stejně jako normálový vektor plochy na začátku bloku N22 a normálový vektor plochy na konci bloku N22). Tím je předpoklad pro aktivování korekce nástroje pomocí příkazů G41/G42 splněn.
N23 X-248.07130 Y30.15119 Z8.71082 A5=-.060165696 B5=.971048883 C5=-.231179920 A3=-.214177198 B3=.926684940 C3=-.308841625	
N24 X-248.07829 Y29.97126 Z8.05094 A5=-.059884286 B5=.968941717 C5=-.239928784 A3=-.213318480 B3=.923853466 C3=-.317789237	
N25 X-248.08317 Y29.78487 Z7.38844 A5=-.059584206 B5=.966718449 C5=-.248807482 A3=-.212397895 B3=.920898045 C3=-.326854594	
N26 X-248.08578 Y29.59254 Z6.72679 A5=-.059263963 B5=.964380907 C5=-.257793037 A3=-.211418355 B3=.917822366 C3=-.336012474	
...	

Poznámka

Při 3D frézování na čelní ploše s funkcí CUT3DFD je pro aktivování korekce nástroje pomocí příkazů G41/G42 zapotřebí definice normálového vektoru plochy. Naprogramování příkazů G41/G42 bez definice normálového vektoru plochy má za následek aktivování alarmu.

Další informace**3D frézování na čelní ploše**

Pro tento druh 3D frézování je zapotřebí popis řádek po řádku jednotlivých 3D drah na povrchu obrobku. Výpočty se obvykle uskutečňují v systému CAM a zohledňuje se při nich tvar nástroje a rozměry nástroje. Postprocesor zapisuje do výrobního programu - vedle NC bloků - orientace nástroje (když je aktivní 5-osá transformace) a příkazy G-funkcí pro požadovanou 3D korekci nástroje. V důsledku toho má obsluha stroje možnost - na rozdíl od případu, kdy se nástroj používá pro výpočet jeho drah v NC systému - používat nepatrně menší nástroje.



Obrázek 11-3 Čelní frézování

Chování při najíždění

Pro chování při najíždění je u 3D variant korekce rádiusu nástroje vždy zvoleno NORM.

Chování na vnějších rozích

S vnějšími rohy se při frézování na čelní ploše zachází jako s kruhovými oblouky s rádiusem 0, přičemž rovina tohoto kruhu je dána tečnou na konci prvního bloku a tečnou na začátku druhého bloku. Díky tomu je možná také změna orientace při přechodu na další blok. Na vnějších rozích se proto vždy vkládá kruh jako konturový prvek. Najíždění na průsečík není při frézování na čelní ploše možné.

Chování při změnách orientace na vnějších rozích

Pomocí příkazů G-funkcí ORIC a ORID může být definováno, zda se změny orientace, které byly naprogramovány mezi oběma bloky posuvů vytvářejících roh, uskuteční před zahájením zpracování vloženého bloku kruhového oblouku (ORID) nebo současně s ním (ORIC).

Korekce nástroje vztažená na diferenční nástroj

3D korekce rádiusu nástroje vztažená na diferenční nástroj se aktivuje prostřednictvím příkazu CUT3DFD. Je potřeba ji použít tehdy, jestliže je naprogramovaná kontura vztažena na dráhu středu normovaného nástroje a pokud se pro opracování používá jeden z nástrojů, které se od něj odlišují. Při výpočtu 3D korekce rádiusu nástroje se potom započítává pouze hodnota opotřebení rádiusu aktivního nástroje (\$TC_DP15) a v případě potřeby naprogramovaný korekční offset nástroje OFFN a TOFFR. Základní rádius (\$TC_DP6) aktivního nástroje se **nezapočítává**.

3D frézování na čelní ploše s příkazem CUT3DFD je možné pouze ve spojení s funkcí „Vyhlazení normálového vektoru plochy při 3D frézování na čelní ploše“. Ta se aktivuje prostřednictvím volání funkce „Top Surface“, pro kterou je nezbytná licence, pomocí cyklu CYCLE832(...) (Strana 799). Aktivování musí být uskutečněno **před** zapnutím korekce nástroje pomocí příkazů G41/G42, a sice nikoli přímo před zakousnutím nástroje, ale již o jeden úsek délky dráhy dříve, který odpovídá přibližně 1000-násobku tolerance kontury (např. 1000 x 0,01 mm = 10 mm). Deaktivování musí probíhat v opačném pořadí operací: Napřed deaktivujte korekci nástroje pomocí příkazu G40, potom, po jednom úseku dráhy, který odpovídá přibližně 1000-násobku tolerance kontury, funkci deaktivujte příkazem např. CUT2D (a podobným).

Aby bylo možno použít „Vyhlazení normálového vektoru plochy při 3D frézování na čelní ploše“, je zapotřebí navíc ještě odblokování funkce „Interpolace normálového vektoru plochy pomocí polynomů“:

MD28291 \$MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Poznámka

Pro 3D frézování na čelní ploše s funkcí CUT3DFD ve spojení s funkcí „Top Surface“ je zapotřebí dodržet doporučení pro nastavení parametrů, která se týkají funkce „Top Surface“!

Pro kontrolu nastavených dat jsou k dispozici (<https://support.industry.siemens.com/cs/ww/en/view/109738423>) speciální zkušební programy, které si lze stáhnout přes portál SIOS:

11.5.3 3D obvodové frézování se zohledněním hraniční plochy (CUT3DCC, CUT3DCCD)

V případě 3D obvodového frézování se spojitou nebo konstantní změnou orientace nástroje je často programována dráha středu definovaného normovaného nástroje. Protože v praxi se nestává často, že by byl vyhovující normovaný nástroj k dispozici, může se použít nástroj, který se od normovaného nástroje přespříliš neliší ($\leq 5\%$).

Pomocí příkazu CUT3DCCD je možné pro specifický reálný nástroj zohlednit omezující plochu, která by popisovala naprogramovaný normovaný nástroj. NC program popisuje dráhu středu normovaného nástroje.

Pomocí příkazu CUT3DCC je možné při použití válcových nástrojů zohlednit omezující plochu, na kterou by měl naprogramovaný normovaný nástroj dosáhnout. NC program popisuje konturu obráběné plochy.

Normálový vektor omezující plochy se zadává stejně jako při 3D frézování na čelní ploše pomocí příkazů A4, B4, C4 a A5, B5, C5.

Syntaxe

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Význam

CUT3DCCD:	3D korekce rádiusu nástroje pro obvodové frézování se zohledněním omezovací plochy s diferenčním nástrojem na dráze středu nástroje: Přísuv k omezující ploše.
CUT3DCC:	3D korekce rádiusu nástroje pro obvodové frézování se zohledněním omezovací plochy s 3D korekcí rádiusu: Kontura na obráběné ploše

G41/G42 X... Y... Z...:	Aktivování korekce rádiusu nástroje	
	G41:	Korekce rádiusu nástroje vlevo od kontury
	G42:	Korekce rádiusu nástroje vpravo od kontury
	Poznámka: Aktivování se musí nacházet v bloku lineárního pohybu (G0/G1).	
CDOF2:	Deaktivování monitorování kolize pro 3D obvodové frézování	
A4/5=... B4/5=... C4/5=...:	Definice normálového vektoru plochy pro omezovací rovinu	
	A4=... B4=... C4=...:	Definice makra na začátku bloku
	A5=... B5=... C5=...:	Definice makra na konci bloku
G40 X... Y... Z...:	Deaktivování korekce rádiusu nástroje	
	Poznámka: Deaktivování se musí nacházet v lineárním bloku (G0/G1) s pohybem geometrických os.	

Poznámka

Příkazy G-funkcí pro aktivování 3D korekce rádiusu nástroje jsou vyhodnocovány v najížděcím bloku, tzn. typicky v bloku, který obsahuje příkaz G41 nebo G42.

Příkaz G41, příp. G42, může být naprogramován také v blocích, ve kterých geometrické osy, jichž se korekce týká, neuskutečňují žádné posuvy. V tomto případě je najížděcím blokem první blok s posuvy, který za tímto blokem následuje.

Změna varianty 3D korekce rádiusu nástroje, když je korekce rádiusu nástroje aktivní, je ignorována, přičemž se neaktivuje žádný alarm.

Příklad:

Programový kód	Komentář
N10 \$TC_DP1[1,1]=120	; Válcová fréza
N20 \$TC_DP6[1,1]=10	
N30 \$TC_DP15[1,1]=-3	
...	
; Opracování pomocí válcové frézy a funkce CUT3DCCD	
N110 TRAORI	; Aktivování transformace.
N120 A4=0 B4=0 C4=1	; Definice normálového vektoru plochy pro omezovací rovinu na začátku bloku
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64	; Aktivování 3D obvodového frézování při zohledňování omezovací plochy + deaktivování monitorování kolize.
N150 X20	
N160 X30 A45	; Tupý úhel ==> žádný přísuv
N170 X40 A-45	; Špičatý úhel ==> přísuv
N180 X55	
N190 Y10 Z10	; Pohyb ve směru nástroje.
N200 Y20	

Programový kód	Komentář
N210 C45	; Čistá změna orientace.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; Změna povrchu.
N240 Y50 G40	; Deaktivování korekce rádiusu nástroje.
...	

Další informace

Typ nástroje

Vyhodnocuje se typ nástroje (parametr nástroje \$TC_DP1). Přípustné jsou pouze frézovací nástroje s válcovou stopkou (válcová nebo stopková fréza, Torus fréza a v mezním případě ještě i válcová zápusťková fréza). To odpovídá typům nástrojů s čísly 1 - 399 s výjimkou čísel 111 a 155 až 157.

Normovaný nástroj se zaoblením rohů

Rohové zaoblení normovaného nástroje je popisováno pomocí parametru nástroje \$TC_DP7. Z parametru nástroje \$TC_DP16 vyplývá odchylka rohového zaoblení reálného nástroje oproti normovanému nástroji.

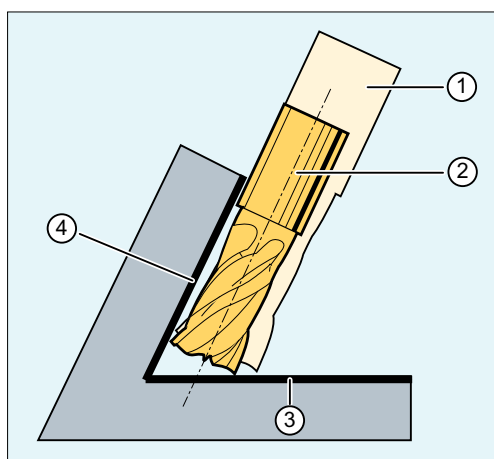
Příklad: Torus frézy se zmenšeným rádiusem oproti normovanému nástroji

Typ nástroje	Rádus stopky (R)	Rádus v rohu (r)
Normovaný nástroj se zaoblením rohů	$R = \$TC_DP6$	$r = \$TC_DP7$
Reálný nástroj se zaoblením rohů Typy nástrojů 121 a 131 Torus frézy (stopkové frézy se zaoblením rohů)	$R' = \$TC_DP6 + \$TC_DP15 + OFFN$	$r' = \$TC_DP7 + \TC_DP16

V tomto příkladu je záporná jak hodnota $\$TC_DP15 + OFFN$, tak také hodnota $\$TC_DP16$.

3D korekce rádiusu nástroje s funkcí CUT3DCCD: Dráha středu nástroje s přísuvem až po omezující plochu

Jestliže je použit nástroj, který má ve srovnání s vyhovujícím normovaným nástrojem menší rádius, potom se v podélném směru přisouvaná fréza pohybuje ještě dál o takovou hodnotu, dokud znovu nedojde k dotyku se dnem kapsy. Tímto způsobem jsou rýžky, které jsou tvořeny mezi obráběnou a omezující plochou, obrobena až do takové míry, kterou daný nástroj připouští. Přitom se jedná o smíšený způsob obrábění skládající se z obvodového a čelního frézování. Analogicky k nástroji s menším rádiusem se v případě použití nástroje se zvětšeným rádiusem přísuv zmenšuje o odpovídající hodnotu v opačném směru.



- ① Normovaný nástroj
- ② Nástroj s menším rádiusem je přisouván až na omezující plochu
- ③ Omezující plocha
- ④ Obráběná plocha

Oproti všem ostatním korekcím nástroje ze skupiny G-funkcí č. 22 nemá pro příkaz CUT3DCCD zadávaný parametr nástroje \$TC_DP6 žádný význam pro rádius nástroje a výslednou konturu proto neovlivňuje. Korekční offset vyplývá ze součtu hodnot opotřebení rádiusu nástroje (parametr nástroje \$TC_DP15) a offsetu nástroje OFFN naprogramovaného pro výpočet kolmého offsetu k omezující ploše.

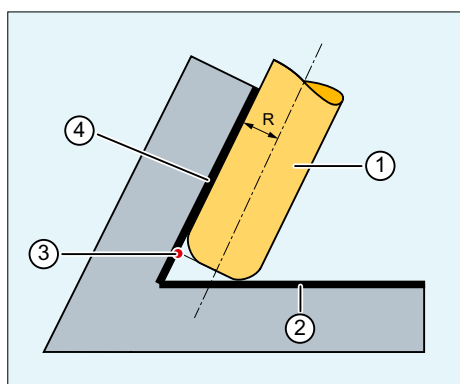
To, zda obráběná plocha leží vlevo nebo vpravo od dráhy, nemůže být ze sestaveného výrobního programu zjištěno. Vychází se proto z kladné hodnoty rádiusu a záporné hodnoty opotřebení původního nástroje. Záporná hodnota opotřebení vždy popisuje nástroj se zmenšeným průměrem.

Použití válcových nástrojů

Při použití válcových nástrojů je přísuv potřebný jedině tehdy, pokud obráběná plocha a omezující plocha mezi sebou svírají ostrý úhel (menší než 90 stupňů). Pokud se používají Torus frézy (stopková fréza s rohovým zaoblením), potom tato situace vyžaduje přísuv v podélném směru nástroje tak u ostrých, tak i u tupých úhlů.

3D korekce rádiusu nástroje s funkcí CUT3DCC: Kontura na obráběné ploše

Pokud je aktivní příkaz CUT3DCC a používá se Torus fréza, je naprogramovaná dráha vztažena na fiktivní válcovou frézu stejného průměru. Z toho vyplývající vztažný bod dráhy je v případě použití Torus frézy uveden na následujícím obrázku.



- ① Torus fréza
- ② Omezující plocha
- ③ Vztažný bod dráhy
- ④ Obráběná plocha
- R Rádus stopky (rádius nástroje)

Je přípustné, aby se úhel mezi obráběnou a omezující plochou i v rámci jednoho bloku měnil z ostrého na tupý a obráceně.

Oproti normovanému nástroji smí být použitý reálný nástroj jak větší, tak také menší. Výsledný rohový rádius přitom nesmí být záporný a znaménko výsledného rádiusu nástroje musí zůstat zachováno.

U příkazu CUT3DCC se NC program vztahuje na konturu v rovině obrábění. Přitom se používá stejně jako v případě obvyklé korekce rádiusu nástroje celkový rádius, který se skládá z následujících složek:

- Rádus nástroje (parametr nástroje \$TC_DP6)
- Hodnota opotřebení (parametr nástroje \$TC_DP15)
- Offset nástroje OFFN naprogramovaný pro výpočet kolmého offsetu k omezující ploše

Poloha omezující plochy se určuje z následujícího rozdílu:

Rozměry normovaného nástroje - rádus nástroje (parametr nástroje \$TC_DP6).

Advanced Surface / Top Surface

Poznámka

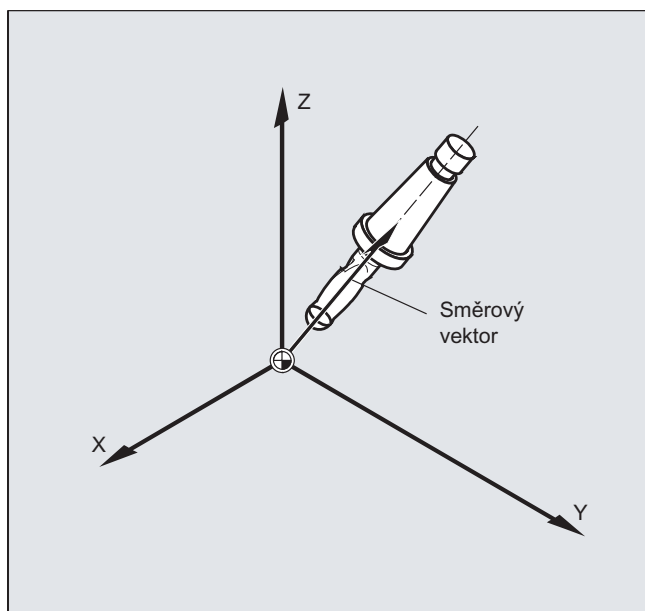
Při použití korekce rádiusu nástroje CUT3DCC / CUT3DCCD ve spojení s funkcí „Advanced Surface“ nebo „Top Surface“, pro které je nezbytná licence, je zapotřebí dodržet doporučení pro nastavení parametrů, která se týkají funkcí „Advanced Surface“ / „Top Surface“!

Pro kontrolu nastavených dat jsou k dispozici speciální zkušební programy, které si lze stáhnout přes portál SIOS:

- Zkušební programy pro funkci Advanced Surface (<https://support.industry.siemens.com/cs/ww/en/view/78956392>)
- Zkušební programy pro funkci Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

11.6 Orientace nástroje (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Pod pojmem orientace nástroje se rozumí geometrické nasměrování nástroje v prostoru. V případě obráběcího stroje s 5 osami může být orientace nástroje nastavována pomocí programových příkazů.



Vyhlazovací pohyby orientace aktivované příkazy **OSD** a **OST** jsou v závislosti na druhu interpolace orientace nástroje realizovány různými způsoby.

Když je aktivní vektorová interpolace, interpoluje se vyhlazený průběh orientace také pomocí vektorové interpolace. Oproti tomu se při aktivní interpolaci kruhových os orientace vyhlazuje přímo pomocí pohybů kruhových os.

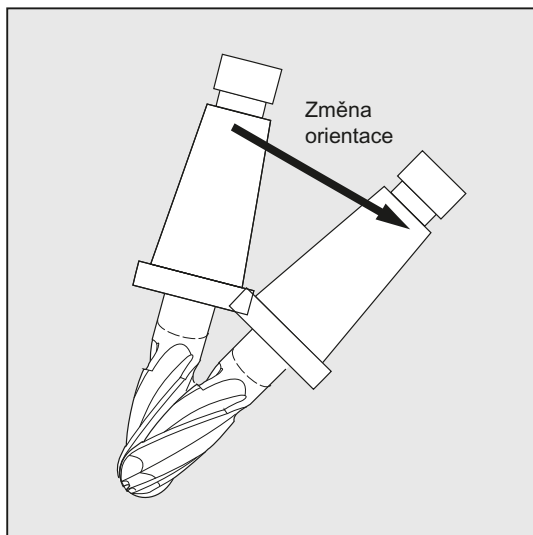
Programování

Programování změny orientace:

Změna orientace nástroje může být naprogramována pomocí těchto prostředků:

- Přímé programování kruhových os A, B, C (interpolace kruhových os)
- Eulerův úhel nebo úhel RPY.
- Směrový vektor (vektorová interpolace pomocí zadání příkazů **A3** nebo **B3** nebo **C3**)
- **LEAD/TILT** (frézování na čelní ploše)

Vztažným souřadným systémem je buď souřadný systém stroje (**ORIMKS**) nebo aktuální souřadný systém obrobku (**ORIWKs**).

**Programování orientace nástroje:**

ORIC:	Orientace a dráhový pohyb probíhají souběžně
ORID:	Orientace a dráhový pohyb probíhají jeden po druhém
OSOF:	Žádné vyhlazování orientace
OSC:	Konstantní orientace
OSS:	Vyhlazení orientace jen na začátku bloku
OSSE:	Vyhlazení orientace na počátku a konci bloku
ORIS:	Rychlost změn orientace při aktivovaném vyhlazování orientace ve stupních na mm (platí pro funkce OSS a OSSE)
OSD:	Vyhlazení orientace zadáním délky zaoblení pomocí nastavovaného parametru: SD42674 \$SC_ORI_SMOOTH_DIST
OST:	Vyhlazení orientace v případě vektorové interpolace zadáním úhlové tolerance ve stupních pomocí nastavovaného parametru: SD42676 \$SC_ORI_SMOOTH_TOL V případě interpolace kruhových os se zadaná tolerance používá jako maximální odchylka orientačních os.

Poznámka

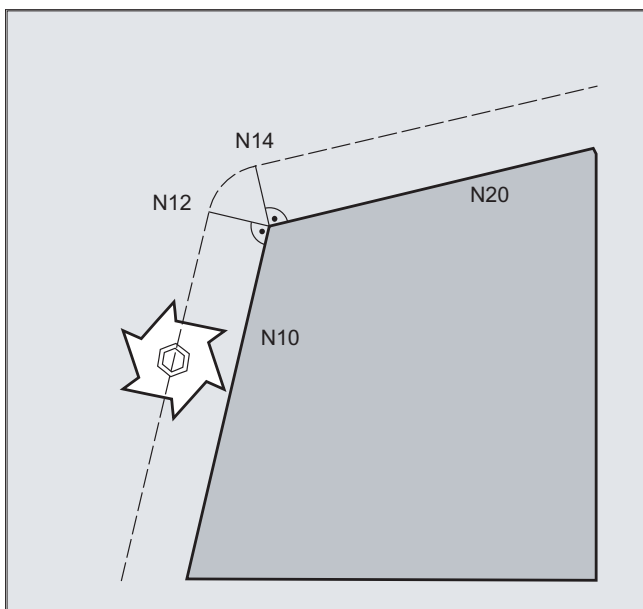
Všechny příkazy pro vyhlazování orientace nástroje (OSOF, OSC, OSS, OSSE, OSD a OST) jsou soustředěny ve skupině G-funkcí č. 34. Mají modální platnost, tzn. může být v platnosti vždy jen jeden z těchto příkazů.

Příklady**Příklad 1: ORIC**

Pokud jsou mezi bloky posuvů N10 a N20 naprogramovány dva nebo více bloků se změnami orientace (např. A2=... B2=... C2=...) a pokud je aktivní funkce ORIC, potom se

11.6 Orientace nástroje (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

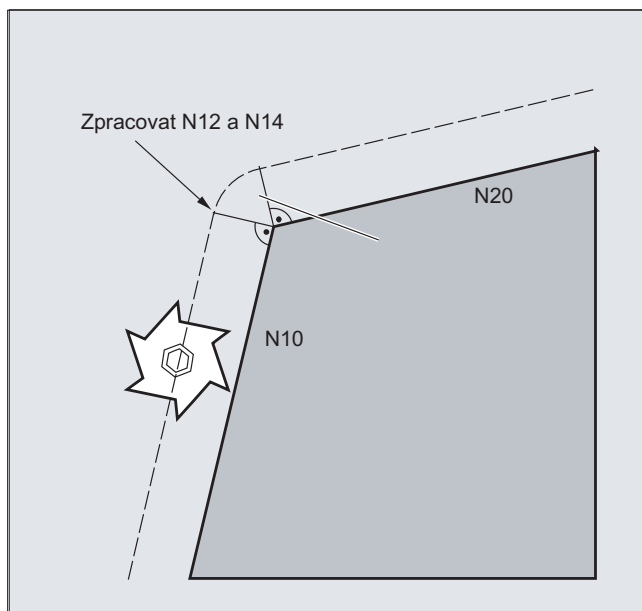
vkládáný blok kruhového oblouku rozdělí v závislosti na počtu úhlových změn na tyto pomocné vložené bloky.



Programový kód	Komentář
ORIC	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 C2=... B2=...	; blok kruhového oblouku, který se vkládá na vnějších rozích, se rozdělí na N12 a N14 v souladu se změnou orientace Pohyb po kruhovém oblouku a změny orientace budou přitom prováděny souběžně.
N14 C2=... B2=...	
N20 X =...Y=... Z=... G1 F200	

Příklad 2: ORID

Jestliže je aktivní příkaz ORID, jsou všechny bloky mezi oběma posuvovými bloky uskutečňovány na konci prvního bloku posuvu. Blok kruhového oblouku s konstantní orientací bude uskutečněn bezprostředně před druhým blokem posuvu.

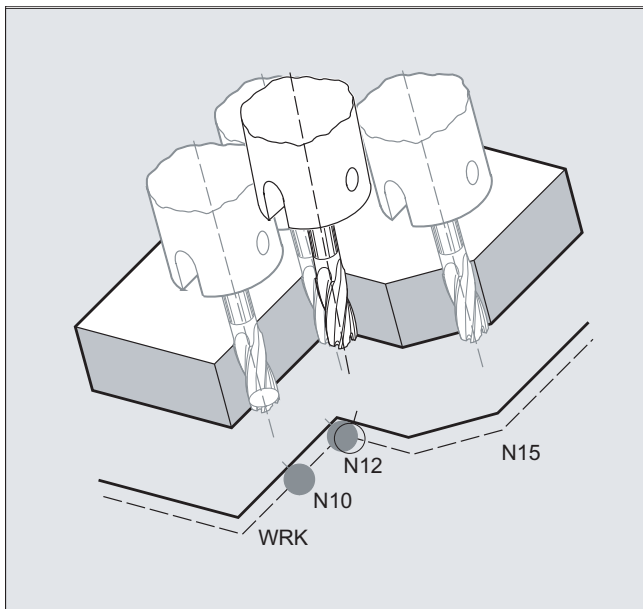


Programový kód	Komentář
ORID	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 A2=... B2=... C2=...	; bloky N12 a N14 se uskuteční na konci bloku N10 Potom bude následovat blok kruhového oblouku s momentálně nastavenou orientací.
N14 M20	; pomocné funkce atd.
N20 X... Y... Z...	

Poznámka

Pro způsob změny orientace na vnějším rohu je určující programový příkaz, který je aktivní v prvním bloku posuvu na vnějším rohu.

Beze změny orientace: Jestliže orientace na hranici bloku není změněna, potom se nástroj pohybuje po kruhové dráze, která se dotýká obou kontur.

Příklad 3: Změna orientace ve vnitřním rohu**Programový kód**

```

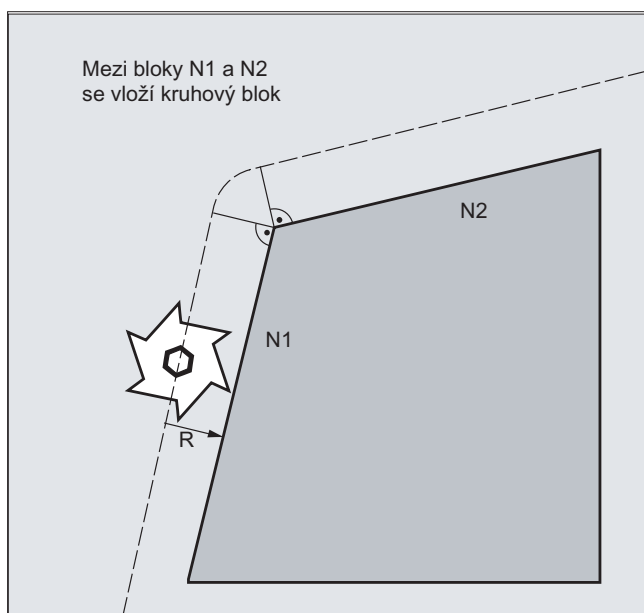
ORIC
N10 X ...Y... Z... G1 F500
N12 X ...Y... Z... A2=... B2=... C2=...
N15 X ...Y... Z... A2=... B2=... C2=...

```

Další informace**Chování na vnějších rozích**

Na vnějších rozích se vždy vkládá blok kruhového oblouku s rádiusem frézy.

Pomocí programových příkazů ORIC, příp. ORID může být definováno, zda se změny orientace, které byly naprogramovány mezi bloky N1 a N2, uskuteční před zahájením zpracování vloženého bloku kruhového oblouku nebo současně s ním.



Jestliže je na vnějším rohu nezbytná změna orientace, potom si můžete vybrat, jestli se má uskutečnit souběžně s interpolací nebo odděleně od pohybu po dráze.

Je-li aktivní funkce **ORID**, uskuteční se napřed vložené bloky bez pohybu po dráze. Blok kruhového oblouku se vloží bezprostředně před druhý z obou bloků posuvu, kterými je roh tvořen.

Jestliže je na vnějším rohu vložen větší počet bloků změny orientace a pokud je aktivní funkce **ORIC**, potom se pohyb po kruhovém oblouku rozdělí na jednotlivé vkládané bloky v závislosti na počtu změn orientace.

Vyhlazení průběhu orientace pomocí příkazů **OSD**, příp. **OST**

V případě přechodových zaoblení aktivovaných funkcí **G642** se nemůže maximální odchylka pro konturové osy a orientační osy příliš lišit. Menší z obou tolerancí určuje formu pohybu na přechodových zaobleních, příp. toleranci úhlu, aby se průběh orientace relativně silně vyhladil, aniž by však přitom muselo docházet k velkým odchylkám od kontury.

Aktivováním příkazů **OSD**, příp. **OST** je možné zajistit, aby se s předem zadanou délkou přechodového zaoblení, příp. úhlovou tolerancí, probíhalo vyhlazení s jen velmi malými odchylkami průběhu orientace a bez velkých odchylek od kontury.

Poznámka

Narozdíl od vyhlazování kontury (a průběhu orientace) pomocí příkazu **G642** se při vyhlazování orientace pomocí příkazů **OSD**, příp. **OST** nevytváří žádný vlastní blok, nýbrž místo toho se zaoblovací pohyb vkládá přímo do naprogramovaných původních bloků.

Pomocí příkazů **OSD**, příp. **OST** nemohou být zaoblovány žádné přechody mezi bloky, ve kterých se uskutečňuje změna druhu interpolace pro orientaci nástroje (vektor --> kruhové osy, kruhové osy --> vektor). Tyto přechody mezi bloky mohou být v případě potřeby zaobleny pomocí běžných funkcí pro přechodová zaoblení **G641**, **G642**, příp. **G643**.

11.7 Volné zadávání D-čísel, číslo bříty

11.7.1 Volné zadávání D-čísel, číslo bříty (adresa CE)

D-číslo

D-čísla mohou být stejně jako čísla korekcí zřetězena. Kromě toho je možné pomocí adresy CE určovat číslo bříty. Pomocí systémové proměnné \$TC_DPCE může být číslo bříty popsáno.

Předdefinované nastavení: Číslo korekčních parametrů == číslo bříty

Prostřednictvím strojních parametrů jsou definovány maximální počet D-čísel (čísel břitů) a maximální počet břitů na jeden nástroj (--> výrobce stroje). Následující příkazy mají smysl jedině tehdy, pokud byl definován maximální počet břitů (MD18105) větší než počet břitů na jeden nástroj (MD18106). Věnujte prosím pozornost informacím od výrobce stroje.

Poznámka

Kromě zadávání relativních D-čísel mohou být zadávána D-čísla také jako "prostá", příp. "absolutní" D-čísla (1 - 32000) bez jakékoli souvislosti s T-čísly (v rámci funkce "Struktura prostých D-čísel").

Literatura

Příručka Popis funkcí, Základní funkce; "Korekční parametry nástrojů (W1)

11.7.2 Volné zadávání D-čísel: Kontrola D-čísla (CHKDNO)

Pomocí příkazu `CHKDNO` zkontrolujete, zda příslušné D-číslo již bylo jednoznačně přiděleno. D-čísla všech v rámci jedné jednotky TO definovaných nástrojů se smí vyskytnout jen jedenkrát. Náhradní nástroje se přitom nezohledňují.

Syntaxe

```
state=CHKDNO (Tno1, Tno2, Dno)
```

Význam

state:	= TRUE:	D-číslo bylo pro kontrolovaný rozsah jednoznačně přiděleno.
	= FALSE:	Vyskytla se kolize D-čísel nebo nastavení parametrů je neplatné. Pomocí parametrů Tno1, Tno2 a Dno jsou předávány parametry, které měly za následek kolizi.. Tato data mohou být ve výrobním programu vyhodnocována.
CHKDNO (Tno1, Tno2) :		Budou zkontrolována všechna D-čísla uvedených nástrojů.

CHKDNO (Tno1):	Budou zkontrolována všechna D-čísla nástroje Tno1 proti všem ostatním nástrojům.
CHKDNO:	Budou zkontrolována všechna D-čísla všech nástrojů proti všem ostatním nástrojům.

11.7.3 Volné zadávání D-čísel: Přejmenovávání D-čísel (GETDNO, SETDNO)

D-čísla musí být přiřazována jednoznačně. Dva různé bříty nástroje nemohou mít stejné D-číslu.

GETDNO

Tento příkaz zjišťuje D-číslu určitého bříty (ce) nástroje s T-číslem t. Jestliže žádné D-číslu k zadanému parametru neexistuje, bude nastaveno d=0. Pokud je D-číslu neplatné, bude výsledkem hodnota větší než 32000.

SETDNO

Pomocí tohoto příkazu přiřadíte hodnotu d D-číslu bříty ce nástroje t. Pomocí stavové proměnné je oznamován výsledek tohoto příkazu (TRUE nebo FALSE). Jestliže žádný datový blok k uvedeným parametrům neexistuje, bude výsledkem hodnota FALSE. Syntaktická chyba bude mít za následek alarm. D-číslu není možné explicitně nastavit na 0.

Syntaxe

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Význam

d:	D-číslu bříty nástroje
t:	T-číslu nástroje
ce:	Číslu bříty (CE-číslu) nástroje
state:	Udává, zda bylo možné uskutečnit příkaz bezchybně (TRUE nebo FALSE).

Příklad - Přejmenování D-čísla

Programování	Komentář
\$TC_DP2[1,2] = 120	
\$TC_DP3[1,2] = 5.5	
\$TC_DPCE[1,2] = 3	; číslo bříty CE
...	
N10 def int DNrAlt, DNrNeu = 17	
N20 DNrAlt = GETDNO(1,3)	
N30 SETDNO(1,3,DNrNeu)	

Tímto je bříty CE=3 přiřazena nová hodnota D-čísla 17. Nyní je přístup k datům tohoto bříty zajištěn přes D-číslo 17, a to jak prostřednictvím systémových proměnných, tak také při programování pomocí NC adresy.

11.7.4 Volné zadávání D-čísel: Zjišťování T-čísla k zadanému D-číslu (GETACTTD)

Pomocí příkazu `GETACTTD` zjistíte T-číslo, které odpovídá k absolutnímu D-číslu. Neuskutečňuje se žádná kontrola jednoznačnosti. Jestliže v rámci jedné jednotky TO existuje větší počet stejných D-čísel, bude výsledkem T-číslo prvního nalezeného nástroje. Při použití "prostých" D-čísel nemá použití tohoto příkazu žádný smysl, protože výsledkem je vždy hodnota "1" (ve správě dat se nenachází žádné T-číslo).

Syntaxe

`status=GETACTTD (Tnr, Dnr)`

Význam

Dnr:	D-číslo, pro které má být vyhledáno T-číslo.	
Tnr:	Nalezené T-číslo	
status:	Hodnota:	Význam:
	0	T-číslo bylo nalezeno Parametr Tnr obsahuje hodnotu T-čísla.
	-1	K zadanému D-číslu neexistuje žádné T-číslo, Tnr = 0.
	-2	D-číslo není absolutní. Tnr obsahuje hodnotu prvního nalezeného nástroje, který má D-číslo, jehož hodnota odpovídá parametru Dnr.
	-5	Funkce nemohla být z nějakého jiného důvodu uskutečněna.

11.7.5 Volné zadávání D-čísel: Nastavení D-čísla jako neplatného (DZERO)

Příkaz `DZERO` poskytuje podporu během přezbrojení. Takto označené bloky korekčních parametrů už nejsou příkazem `CHKDNO` kontrolovány. Aby se znovu staly přístupnými, musí být D-číslo znovu nastavena pomocí příkazu `SETDNO`.

Syntaxe

`DZERO`

Význam

DZERO:	Označení všech D-čísel dané jednotky TO jako neplatné.
--------	--

11.8 Kinematika držáku nástroje

Předpoklady

Držák nástroje je schopen nastavit nástroj do všech možných prostorových orientací jen tehdy, pokud:

- Jsou k dispozici dvě kruhové osy V_1 a V_2
- Tyto kruhové osy jsou vůči sobě kolmé
- Podélná osa nástroje se nachází kolmo na druhou kruhovou osu V_2 .

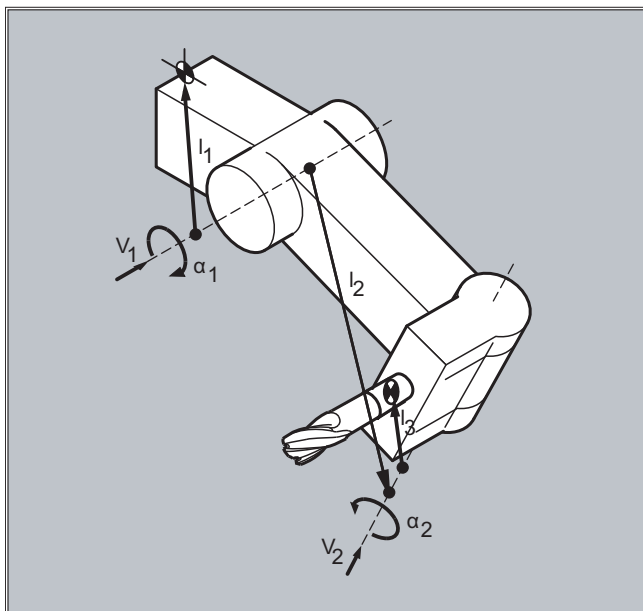
Kromě toho platí u strojů, u nichž musí být nastavitelné všechny možné orientace, následující předpoklad:

- Orientace nástroje musí být kolmo na první kruhovou osu V_1 .

Funkce

Kinematika držáku nástroje s maximálně dvěma otočnými osami V_1 nebo V_2 je popsána pomocí 17 systémových proměnných \$TC_CARR1[m] až \$TC_CARR17[m]. Popis držáku nástroje se skládá z těchto částí:

- Vektorová vzdálenost od první kruhové osy po vztažný bod držáku nástroje I_1 , vektorová vzdálenost od první ke druhé kruhové ose I_2 a vektorová vzdálenost od druhé kruhové osy ke vztažnému bodu nástroje I_3 .
- Směrové vektory obou kruhových os V_1 a V_2
- úhlů otočení α_1 , α_2 okolo obou os Úhly otočení se určují při pohledu ve směru vektorů kruhových os a ve směru hodinových ručiček jsou považovány za kladné.



Pro stroje s **kinematikou s více stupni volnosti** (jak nástroj, tak i obrobek se mohou otáčet) byly systémové proměnné rozšířeny o položky \$TC_CARR18[m] až \$TC_CARR23[m].

Parametr

Funkce systémových proměnných pro orientovatelný držák nástroje			
Popis	složka x	složka y	složka z
l_1 vektor offsetu	\$TC_CARR1[m]	\$TC_CARR2[m]	\$TC_CARR3[m]
l_2 vektor offsetu	\$TC_CARR4[m]	\$TC_CARR5[m]	\$TC_CARR6[m]
v_1 kruhová osa	\$TC_CARR7[m]	\$TC_CARR8[m]	\$TC_CARR9[m]
v_2 kruhová osa	\$TC_CARR10[m]	\$TC_CARR11[m]	\$TC_CARR12[m]
α_1 úhel otočení	\$TC_CARR13[m]		
α_2 úhel otočení	\$TC_CARR14[m]		
l_3 vektor offsetu	\$TC_CARR15[m]	\$TC_CARR16[m]	\$TC_CARR17[m]

Rozšíření systémových proměnných pro orientovatelný držák nástroje			
Popis	složka x	složka y	složka z
l_4 vektor offsetu	\$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	\$TC_CARR23[m]		
Tool	Typ kinematiky T ->	Typ kinematiky P ->	Typ kinematiky M ->
Part	Otočný je jen nástroj (předdefinované nastavení)	Otočný je jen obrobek	Obrobek i nástroj jsou otočné
Mixed mode			
Offset kruhové osy V_1 kruhové osy V_2	Úhel ve stupních kruhové osy V_1 a V_2 při zaujmutí základní polohy \$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_CARR26[m] \$TC_CARR27[m]		
Úhlový inkrement v_1 kruhová osa v_2 kruhová osa	Inkrement Hirthova ozubení kruhových os V_1 a V_2 ve stupních \$TC_CARR28[m] \$TC_CARR29[m]		
Min. poloha kruhové osy V_1 kruhové osy V_2	Softwarové koncové spínače pro minimální pozici kruhových os V_1 a V_2 \$TC_CARR30[m] \$TC_CARR31[m]		
Max. poloha kruhové osy V_1 kruhové osy V_2	Softwarové koncové spínače pro maximální pozici kruhových os V_1 a V_2 \$TC_CARR32[m] \$TC_CARR33[m]		
Název držáku nástroje	Namísto čísla může být držáku nástroje přiřazen název. \$TC_CARR34[m]		
Uživatel: Název osy 1 Název osy 2 Identifikace	Zamýšlené použití uživatelem v rámci měřicího systému. \$TC_CARR35[m] \$TC_CARR36[m] \$TC_CARR37[m] \$TC_CARR38[m]		
Pozice	\$TC_CARR39[m]	\$TC_CARR40[m]	\$TC_CARR41[m]
Jemné posunutí	Parametr, který se může přičítat k hodnotám v základních parametrech.		
l_1 vektor offsetu	\$TC_CARR41[m]	\$TC_CARR42[m]	\$TC_CARR43[m]

Rozšíření systémových proměnných pro orientovatelný držák nástroje			
I_2 vektor offsetu	\$TC_CARR44[m]	\$TC_CARR45[m]	\$TC_CARR46[m]
I_3 vektor offsetu	\$TC_CARR55[m]	\$TC_CARR56[m]	\$TC_CARR57[m]
I_4 vektor offsetu	\$TC_CARR58[m]	\$TC_CARR59[m]	\$TC_CARR60[m]
v_1 kruhová osa	\$TC_CARR64[m]		
v_2 kruhová osa	\$TC_CARR65[m]		

Poznámka**Vysvětlivky k parametrům**

Pomocí "m" se vždy zadává číslo popisovaného držáku nástroje.

Parametry \$TC_CARR47 až \$TC_CARR54, jakož i \$TC_CARR61 až \$TC_CARR63 nejsou definovány a v případě pokusu o přístup kvůli jejich čtení nebo zápisu se aktivuje alarm.

Počáteční, příp. koncové body vektorů vzdálenosti na osách mohou být vybrány libovolně. Pro úhly otočení α_1 , α_2 okolo obou os je v základním postavení držáku nástroje definována poloha 0° . Kinematika držáku nástroje takto může být popsána libovolným množstvím možností.

Držák nástroje s jen jednou nebo s žádnou kruhovou osou může být popsán dosazením nul směrovým vektorům jedné nebo obou kruhových os.

U držáku nástrojů bez kruhové osy se vektory vzdálenosti uplatňují jako doplňkové korekce nástroje, jejichž komponenty nejsou při přepnutí roviny obrábění (G17 až G19) ovlivňovány.

Rozšíření parametrů**Parametry kruhových os**

Systémové proměnné byly rozšířeny o položky \$TC_CARR24[m] až \$TC_CARR33[m] a jsou popsány následujícím způsobem:

Offset kruhové osy V_1 , V_2	Změna polohy kruhové osy V_1 nebo V_2 v základní poloze orientovatelného držáku nástroje.
Úhlový offset / úhlový inkrement kruhové osy V_1 , V_2	Offset nebo úhlový inkrement Hirthova ozubení kruhových os V_1 a V_2 . Naprogramovaný nebo vypočítaný úhel se zaokrouhluje na nejbližší hodnotu, která je dána výrazem $\phi = s + n \cdot d$, kde n je celé číslo.
Minimální a maximální poloha kruhové osy V_1 , V_2	Minimální pozice/maximální pozice kruhové osy, úhel otočení (softwarový koncový spínač) kruhové osy V_1 a V_2 .

Parametry pro uživatele

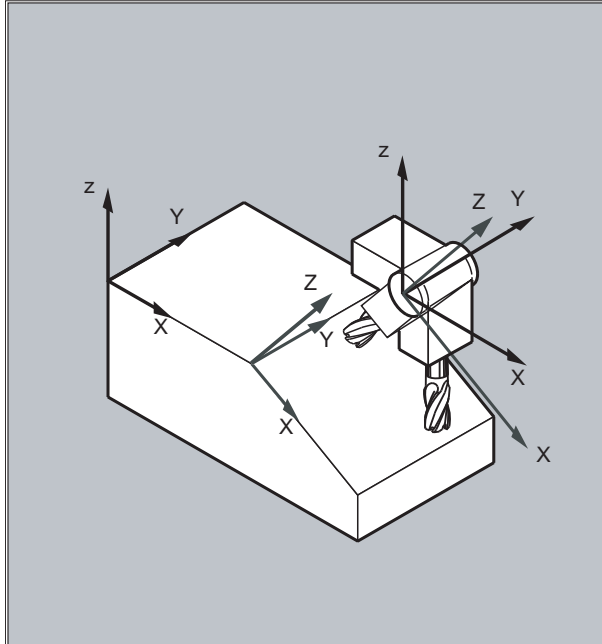
Proměnné \$TC_CARR34 až \$TC_CARR40 obsahují parametry, které jsou uživateli volně k dispozici a až do verze SW 6.4 standardně nejsou v rámci NCK dále vyhodnocovány a ani nemají žádný význam.

Parametry jemného posunutí

\$TC_CARR41 až \$TC_CARR65 obsahují parametry jemného posunutí, které se mohou přičítat k hodnotám základních parametrů. K hodnotě jemného posunutí přiřazené k určitému základnímu parametru se dostanete tak, že se k číslu parametru přičte hodnota 40.

Příklad:

Držák nástroje používaný v následujícím příkladu se dá úplně popsat pomocí otočení okolo osy Y.



Programový kód	Komentář
N10 \$TC_CARR8[1]=1	; definice složky Y první kruhové osy držáku nástroje 1
N20 \$TC_DP1[1,1]=120	; definice stopkové frézy
N30 \$TC_DP3[1,1]=20	; definice stopkové frézy s délkou 20 mm
N40 \$TC_DP6[1,1]=5	; definice stopkové frézy s rádiusem 5 mm
N50 ROT Y37	; definice framu s otočením o 37° okolo osy Y
N60 X0 Y0 Z0 F10000	; najíždění na výchozí pozici
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; nastavení korekce rádiusu, korekce délky nástroje v otočeném framu, aktivování držáku nástroje 1, nástroje 1
N80 X40	; uskutečnění obrábění s otočením 37°
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Další informace

Kinematika s více stupni volnosti

Pro stroje s kinematikou s více stupni volnosti (jak nástroj, tak i obrobek se mohou otáčet) byly systémové proměnné rozšířeny o položky `$TC_CARR18 [m]` až `$TC_CARR23 [m]`, které jsou popsány následujícím způsobem:

Otočný nástrojový stůl se skládá z těchto částí:

- Vektorová vzdálenost mezi otočnou osou V_2 a vztažným bodem otočného nástrojového stolu I_4 třetí kruhové osy.

Kruhové osy se skládají z těchto součástí:

- Dva kanálové identifikátory pro popis vztahu mezi kruhovými osami V_1 a V_2 , jejichž pozic systém v případě potřeby využívá při určování orientace orientovatelného držáku nástroje.

Typ kinematiky s hodnotou T, P nebo M:

- Typ kinematiky T: Otočný je jen nástroj.
- Typ kinematiky P: Otočný je jen obrobek.
- Typ kinematiky M: Obrobek i nástroj jsou otočné.

Vymazání dat držáku nástroje

Pomocí nastavení `$TC_CARR1 [0]=0` mohou být data všech datových bloků držáku nástroje vymazána.

Pro typ kinematiky `$TC_CARR23 [T]=T` musí být dosazeno jedno ze tří přípustných velkých nebo malých písmen (T, P, M) a z tohoto důvodu by proměnná neměla být vymazána.

Editace dat držáku nástroje

Kterákoli z popisovaných hodnot může být změněna přiřazením nové hodnoty ve výrobním programu. Jakýkoli jiný znak než T, P nebo M bude mít při pokusu o aktivování orientovatelného držáku nástroje za následek aktivování alarmu.

Načítání dat držáku nástroje

Kterákoli z popisovaných hodnot může být čtena přiřazením proměnné ve výrobním programu.

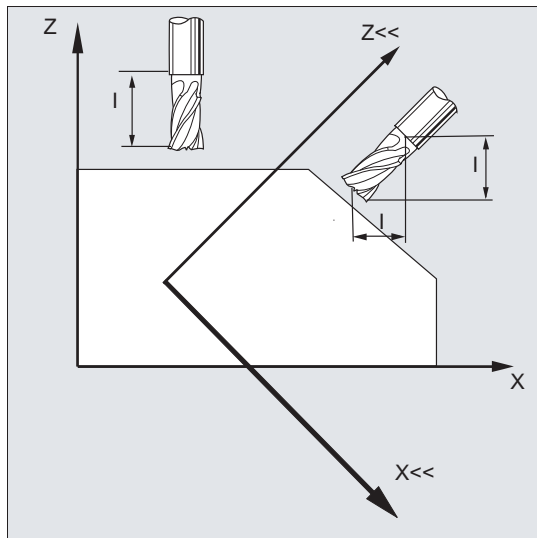
Jemná posunutí

Nepřípustná hodnota jemného posunutí bude rozpoznána až tehdy, když je aktivován orientovatelný držák nástroje, který takovou hodnotu obsahuje, a když je současně nastaven nastavovaný parametr `SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE`.

Hodnota nepřípustného jemného posunutí je pomocí strojního parametru omezena na maximální přípustnou hodnotu.

11.9 Délková korekce nástroje pro orientovatelný držák nástroje (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Při změně prostorové orientace nástroje se mění také jeho složky délky.



Po přestavení, např. manuálním nastavením nebo změnou držáku nástroje s pevným prostorovým nasměrováním, je proto nutné znovu zjistit jednotlivé složky délky nástroje. To se provádí pomocí příkazů dráhy TCOABS a TCOFR.

Pro orientovatelný držák nástroje aktivního framu je možné při volbě nástroje stanovit pomocí příkazů TCOFRZ, TCOFRY a TCOFRX směr, do něž má nástroj ukazovat.

Syntaxe

TCARR= [<m>]
 TCOABS
 TCOFR
 TCOFRZ
 TCOFRY
 TCOFRX

Význam

TCARR= [<m>] :	Vyžádání držáku nástroje s číslem [m]
TCOABS:	Výpočet složek délky nástroje z aktuální orientace držáku nástroje
TCOFR:	Stanovení složek délky nástroje z orientace aktivního framu
TCOFRZ:	Orientovatelný držák nástroje z aktivního framu, jehož nástroj je namířen ve směru osy Z.
TCOFRY:	Orientovatelný držák nástroje z aktivního framu, jehož nástroj je namířen ve směru osy Y.
TCOFRX:	Orientovatelný držák nástroje z aktivního framu, jehož nástroj je namířen ve směru osy X.

Další informace

Korekce délky nástroje z orientace držáku (TCOABS)

Příkaz TCOABS vypočítává korekci délky nástroje z aktuálního úhlu orientace držáku nástroje; ukládá se do systémových proměnných \$TC_CARR13 a \$TC_CARR14.

Budete-li potřebovat nastudovat definici kinematiky držáku nástroje pomocí systémových proměnných, viz " Kinematika držáku nástroje (Strana 458) ".

Aby se uskutečnil nový výpočet korekce délky nástroje při změně framu, musí být nástroj ještě jednou aktivován.

Nasměrování nástroje podle aktivního framu

Orientovatelný držák nástroje může být nastaven tak, aby nástroj ukazoval do následujícího směru:

- pomocí příkazu TCOFR, příp. TCOFRZ bude natočen ve směru osy Z.
- pomocí příkazu TCOFRY bude natočen ve směru osy Y.
- pomocí příkazu TCOFRX bude natočen ve směru osy X.

Přepnutí mezi příkazy TCOFR a TCOABS má za následek nový výpočet korekce délky nástroje.

Vyžádání držáku nástroje (TCARR)

Pomocí příkazu TCARR si vyžádáte držák nástroje m spolu s jeho geometrickými údaji (paměť korekcí).

Když bude m=0, bude aktuální držák nástroje deaktivován.

Geometrické údaje o držáku nástroje se aktivují až po vyvolání nástroje. Zvolený nástroj zůstává aktivní i po výměně držáku nástroje.

Aktuální geometrické údaje o držáku nástroje mohou být definovány i ve výrobním programu prostřednictvím odpovídajících systémových proměnných.

Nový výpočet korekce délky nástroje (TCOABS) při změně framu

Aby se uskutečnil nový výpočet korekce délky nástroje při změně framu, musí být nástroj ještě jednou aktivován.

Poznámka

Orientace nástroje musí být manuálně přizpůsobena aktivnímu framu.

Při výpočtu délkové korekce nástroje se v jednom z mezikroků vypočítává také úhel natočení držáku nástroje. Protože u držáků nástrojů se dvěma kruhovými osami obecně existují také dva páry úhlů natočení, pomocí kterých je možné přizpůsobit orientaci nástroje aktuálnímu

11.9 Délková korekce nástroje pro orientovatelný držák nástroje (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

ramu, musí hodnoty úhlu natočení uložené v systémových proměnných alespoň přibližně odpovídat mechanicky nastaveným úhlům natočení.

Poznámka

Orientace nástroje

Může se stát, že řídicí systém nebude kontrolovat, jestli je možné úhly natočení vypočítané na základě orientace ramu nastavit na stroji.

Pokud jsou kruhové osy držáku nástroje konstrukčně uspořádány tak, že orientace nástroje vypočítané na základě orientace ramu nebude možné dosáhnout, bude generován alarm.

Kombinace jemné korekce nástroje a funkcí pro korekci délky nástroje nejsou u pohyblivých držáků nástrojů přípustné. Při pokusu vyvolat obě funkce současně se vypíše chybové hlášení.

Pomocí příkazu `TOFRAME` je možné definovat frame na základě směru orientace zvoleného držáku nástroje. Pokud budete potřebovat přesnější informace, viz kapitola "Framy".

Když je aktivní transformace orientace (transformace ve 3, 4, 5 osách), je možné zvolit držák nástroje s orientací odlišující se od nulové polohy, aniž by byl přitom generován alarm.

Předávané parametry pro standardní a měřicí cykly

Pro předávané parametry standardních a měřicích cyklů platí definované rozsahy hodnot.

V případě hodnot úhlů je rozsah hodnot definován následujícím způsobem:

- Otočení okolo 1. geometrické osy: -180 stupňů až +180 stupňů
- Otočení okolo 2. geometrické osy: -90 stupňů až +90 stupňů
- Otočení okolo 3. geometrické osy: -180 stupňů až +180 stupňů

Viz kapitola Framy, "Programovatelné otočení (ROT, AROT, RPL)".

Poznámka

Při předávání hodnot úhlů standardním a měřicím cyklům je potřeba mít na paměti:

Hodnoty menší než je přesnost výpočtu NC systému jsou zaokrouhleny na nulu!

Přesnost výpočtu úhlových pozic v NC systému je definována ve strojním parametru:

MD10210 \$MN_INT_INCR_PER_DEG

11.10 On-line korekce délky nástroje (TOFFON, TOFFOF)

Prostřednictvím systémové proměnné \$AA_TOFF[<n>] mohou být trojrozměrně a v reálném čase korigovány efektivní délky nástroje ve všech třech jeho rozměrech.

Jako index <n> se používají tři identifikátory geometrických os. Tímto způsobem je definován počet aktivních směrů korekce prostřednictvím geometrických os, které jsou v daném okamžiku aktivní.

Všechny korekce mohou být aktivní současně.

Funkce On-line korekce délky nástroje se může používat v těchto případech:

- Transformace orientace TRAORI
- Orientovatelný držák nástroje TCARR

Poznámka

On-line korekce délky nástroje je **volitelný doplněk**, který musí být napřed instalován. Tato funkce má smysl jedině tehdy, když je aktivní transformace orientace nebo když je aktivní orientovatelný držák nástroje.

Syntaxe

```
TRAORI
TOFFON(<směr korekce>[,<hodnota offsetu>])
WHEN TRUE DO $AA_TOFF[<směr korekce>] ; v synchronních akcích
...
TOFFOF(<směr korekce>)
```

Pokud budete potřebovat další vysvětlení týkající se programování on-line korekce délky nástroje v pohybových synchronních akcích, viz "Synchronní akce (Strana 621)".

Význam

TOFFON:	Aktivování on-line korekce délky nástroje	
	<směr korekce>:	Směr nástroje (X, Y, Z), ve kterém se má on-line korekce délky nástroje uplatňovat.
	<hodnota offsetu>:	Při aktivování může být pro odpovídající směr korekce zadána hodnota offsetu, přičemž na tento offset se bude ihned najíždět.
TOFFOF:	Vynulování on-line korekce délky nástroje	
	Hodnoty korekce v zadaném směru budou vynulovány a aktivuje se přitom zastavení předběžného zpracování.	

Příklady

Příklad 1: Aktivování korekce délky nástroje

Programový kód	Komentář
MD21190 \$MC_TOFF_MODE =1	; najiždí se na absolutní hodnoty
MD21194 \$MC_TOFF_VELO[0] =1000	
MD21196 \$MC_TOFF_VELO[1] =1000	
MD21194 \$MC_TOFF_VELO[2] =1000	
MD21196 \$MC_TOFF_ACCEL[0] =1	
MD21196 \$MC_TOFF_ACCEL[1] =1	
MD21196 \$MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; aktivování transformace
N20 TOFFON(Z)	; aktivování On-line korekce délky nástroje pro směr Z
N30 WHEN TRUE DO \$AA_TOFF[Z]=10 G4 F5	; pro rozměr nástroje ve směru osy Z se interpoluje korekce délky nástroje 10
...	
N100 XOFFSET=\$AA_TOFF_VAL[X]	; přiřazení aktuální korekce ve směru osy X
N120 TOFFON(X,-XOFFSET) G4 F5	; pro rozměr nástroje ve směru osy X se najede zpátky na hodnotu korekce délky nástroje 0

Příklad 2: Deaktivování korekce délky nástroje

Programový kód	Komentář
N10 TRAORI(1)	; aktivování transformace
N20 TOFFON(X)	; aktivování On-line korekce délky nástroje pro směr X
N30 WHEN TRUE DO \$AA_TOFF[X]=10 G4 F5	; pro rozměr nástroje ve směru osy X se interpoluje korekce délky nástroje 10
...	
N80 TOFFOF(X)	; offset polohy ve směru X nástroje se vymaže ...\$AA_TOFF[X]=0 Žádná osa se nebude pohybovat. K aktuální poloze ve WCS se připočítá offset polohy v souladu s momentální orientací.

Další informace

Příprava bloku

Při přípravě bloku v předběžném zpracování jsou v rámci hlavního zpracování započítávány také momentálně platné offsety délky nástroje. Aby bylo možné v maximální možné míře využívat nejvyšší možnou rychlost pohybu os, je zapotřebí v době, kdy se započítávají offsety nástroje, přerušit přípravu bloku pomocí zastavení předběžného zpracování příkazem `STOPRE`.

Offset nástroje je v okamžiku předběžného zpracování znám také vždy tehdy, pokud se korekce délky nástroje po spuštění programu už nemění nebo pokud bylo po změně korekce délky nástroje zpracováno více bloků, než kolik jich může být uloženo do vyrovnávací paměti IPO mezi předběžným a hlavním zpracováním.

Proměnná \$AA_TOFF_PREP_DIFF

Hodnota rozdílu mezi korekcí, která je momentálně v platnosti v interpolátoru, a korekcí, která byla v platnosti v okamžiku přípravy bloku, může být zjištěna v proměnné \$AA_TOFF_PREP_DIFF[<n>].

Nastavení strojních parametrů a nastavovaných parametrů

Pro On-line korekci délky nástroje máte k dispozici následující systémové proměnné:

- MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE (rezerva zrychlení pro superponovaný pohyb)
- MD21190 \$MC_TOFF_MODE
Na obsah systémové proměnné \$AA_TOFF[<n>] se buď najede jako na absolutní hodnotu nebo se tato hodnota integruje do výpočtů.
- MD21194 \$MC_TOFF_VELO (rychlost on-line korekce délky nástroje)
- MD21196 \$MC_TOFF_ACCEL (zrychlení on-line korekce délky nástroje)
- Nastavovaný parametr pro zadání mezních hodnot:
SD42970 \$SC_TOFF_LIMIT (horní mezní hodnota korekce délky nástroje)

Literatura:

Příručka k funkcím, Speciální funkce; F2: Transformace ve více osách

11.11 Modifikace korekčních parametrů u točivých nástrojů

11.11.1 Výpočet orientace (ORISOLH)

Předdefinovaná funkce ORISOLH pomáhá uživateli nastavovat polohy kruhových os stroje tak, aby byl soustružnický nástroj přestaven do definované polohy vzhledem k obrobku, a to nezávisle na kinematice stroje. Předpokladem je, že je aktivována 6-osá transformace, jejíž parametry jsou nastaveny prostřednictvím kinematického řetězce.

K dispozici jsou dvě základní funkce:

- **Nastavení orientace nástroje**
Jsou zadávány oba úhly β a γ . Funkce vypočítává pro daný účel potřebné úhly tří orientačních os.
- **Přímé nastavování orientace nástroje**
Je zadán úhel druhé a třetí orientační osy. Funkce na základě těchto údajů vypočítává odpovídající úhly β a γ , jakož i úhel dosud chybějící první orientační osy.

Poznámka

Posloupnost orientačních os

Když se prochází kinematickým řetězcem, který popisuje konstrukci stroje od obrobku až po nástroj, platí pro posloupnost tří orientačních os 6-osé transformace následující pravidla:

- Orientační osa, která leží nejbližší **obrobku**, je **první** orientační osou.
- Orientační osa, která leží nejbližší **nástroji**, je **třetí** orientační osou.

První orientační osou je zpravidla vřeteno a odpovídající otočení se proto v těchto případech realizuje prostřednictvím otočného framu.

Syntaxe

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

Význam

ORISOLH:	Volání funkce	
<RetVal>:	Výsledná hodnota funkce	
	Datový typ:	INT
	Rozsah hodnot:	0, -2, -3, ..., -17
	Hodnoty:	0 Funkce byla ukončena bezchybně.
		-2 Není aktivní žádná platná transformace (6-osá transformace orientace).
		-3 První parametr (<Cntrl>) je záporný.
		-4 Místo jednotek prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		-5 Místo desítek prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 až 3.
		-6 Místo stovek prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		-7 Místo tisíců prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 až 3.
		-8 U funkce „Přímé nastavování orientace nástroje“ je úhel γ příliš velký.
		-9 U funkce „Přímé nastavování orientace nástroje“ minimálně jedna ze zadaných pozic osy narušuje mezní hodnotu osy.
		-10 Není aktivní žádný nástroj.
		-11 Požadovaná orientace není nastavitelná.
		-12 Přizpůsobení volného úhlu osy u Hirthova ozubení není pro první nebo jediné řešení možné.
		-13 Přizpůsobení volného úhlu osy u Hirthova ozubení není pro druhé řešení možné.
		-14 Přizpůsobení volného úhlu osy u Hirthova ozubení není pro žádné z obou řešení možné.
	-15	V parametrech je nastaveno, že první orientační osa je Hirthova osa.
	-16	V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.
	-17	U funkce „Přímé naklápění“ je minimálně jedna ze zadaných poloh osy nekompatibilní s odpovídajícím Hirthovým ozubením.

<Ctrl>:	Ovládá chování funkce	
	Datový typ:	INT
	U parametru <Ctrl> se používá desítkové kódování (místo jednotek až tisíců):	
	Místo jednotek:	Místo jednotek ovládá chování v případě chyby.
	xxx0	V případě chyby (výsledná hodnota < 0) je zpracovávání programu přerušeno a aktivuje se alarm 14106. Poznámka: Tento alarm se aktivuje nezávisle na hodnotě na místě jednotek také tehdy, když je hodnota parametru <Ctrl> záporná.
	xxx1	V případě chyby (výsledná hodnota < 0) se neaktivuje žádný alarm. Uživatel má v programu možnost sám vhodným způsobem reagovat.
	Místo desítek:	Ovládání chování v případě, že je k dispozici jedna orientační osa s Hirthovým ozubením. Poznámka: Tento parametr je vyhodnocován pouze v případě funkce „Nastavení orientace nástroje“ (tzn. jestliže je na místě stovek hodnota „0“).
	xx0x	Poloha osy je zaokrouhlována na následující pozici.
	xx1x	Poloha osy je zaokrouhlována tak, že odchylka úhlu β je od jeho naprogramované hodnoty minimální.
	xx2x	Poloha osy je zaokrouhlována tak, aby se úhle β rovnal největší možné hodnotě, která je menší než naprogramovaná hodnota (úhel β se zaokrouhluje dolů).
	xx3x	Poloha osy je zaokrouhlována tak, aby se úhle β rovnal nejmenší možné hodnotě, která je větší než naprogramovaná hodnota (úhel β se zaokrouhluje nahoru).
	Místo stovek:	Udává, jaká funkce se má uskutečňovat, příp. jaký význam mají dva následující parametry <W1> a <W2>.
	x0xx	Funkce „Nastavení orientace nástroje“ Parametry <W1> a <W2> mají následující význam: <W1> = β <W2> = γ Vypočítávají se odpovídající úhly orientačních os.
	x1xx	Funkce „Přímé nastavování orientace nástroje“ <W1> je údaj polohy pro druhou orientační osu, <W2> je údaj polohy pro třetí orientační osu v 6-osé transformaci. Určí se poloha první orientační osy a úhly β a γ , které jsou kompatibilní s oběma předem zadanými údaji poloh. Jestliže se nevyskytne žádná chyba, v systémové proměnné \$P_ORI_POS[<n>, <m>] jsou vždy poskytnuta dvě řešení. První index <n> (0 nebo 1) přitom odkazuje na řešení a druhý index <m> (0 ... 2) odkazuje na orientační osu: \$P_ORI_POS[0/1, 0]: Poloha první orientační osy \$P_ORI_POS[0/1, 1]: Úhel β \$P_ORI_POS[0/1, 2]: Úhel γ Kontroluje se, zda údaje polohy <W1> a <W2> jsou kompatibilní s eventuálně aktivním Hirthovým ozubením nebo s aktivními softwarovými mezními hodnotami. Jestliže tomu tak

		není, je výsledkem odpovídající číslo chyby (viz parametr <RetVal>). Jestliže jsou oba úhly <W1> a <W2> zvoleny libovolně, břit nástroje se obecně nenachází v rovině obrábění. Úhel γ , o který je břit z roviny obrábění otočený, nesmí být větší než určitá mezní hodnota, která je stanovena prostřednictvím nastavovaného parametru SD42999 \$SC_ORISOLH_INCLINE_TOL.
	Místo tisíců:	Udává, jak jsou pozice vyplývající z řešení v případě potřeby upravovány, jestliže je na místě stovek hodnota „0“, tzn. když je aktivována funkce „Nastavení orientace nástroje“.
	0xxx	Vypočítaná pozice osy by se měla nacházet co možno nejblíže momentální polohy strojní osy.
	1xxx	Vypočítané pozice os by měly být v případě os typu modulo co možno nejblíže středu oblasti modulo, u jiných os co možno nejblíže 0. V případě os, které nejsou typu modulo, to znamená, že polohy os jsou omezeny na rozmezí $-180^\circ \dots 180^\circ$.
	2xxx	Vypočítané polohy os mají být nezávisle na jejich typu omezeny na rozmezí $-180^\circ \dots 180^\circ$.
<W1>:	První úhel Význam vyplývá z hodnoty na místě stovek parametru <Cntrl>.	
	Datový typ:	REAL
<W1>:	Druhý úhel Význam vyplývá z hodnoty na místě stovek parametru <Cntrl>.	
	Datový typ:	REAL

Poznámka

Nenaprogramovaným parametrům je dosazována standardní hodnota "0".

Další informace

Počet nalezených řešení spolu s dalšími stavovými informacemi je možno při zpracování funkce ORISOLH načítat prostřednictvím následujících systémových proměnných:

Systémové proměnné	Význam
\$P_ORI_POS [<n>, <m>]	Tato proměnná obsahuje úhly os pro nastavení orientace, které vyplývají z naprogramované orientace.
	<n>: Index řešení
	Rozsah hodnot: 0, 1
	<m>: Index orientační osy
	Rozsah hodnot: 0 ... 2 Posloupnost orientačních os (1 ... 3) se vztahuje na definici os v parametru \$NT_ROT_AX_NAME.
	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ obsahují proměnné \$P_ORI_POS[0/1, 1] a \$P_ORI_POS[0/1, 2] hodnoty obou úhlů β a γ, které patří k oběma řešením. První řešení uložené v proměnné \$P_ORI_POS[<n>, <m>], tzn. s indexem <n> = 0, je vždy to, které je vybráno řídicím systémem, když se přímo najíždí na požadovanou orientaci. Druhý index <m> se přitom vztahuje na orientační osu, tzn. na \$NT_ROT_AX_NAME. Polohy os, které jsou uloženy v proměnné \$P_ORI_POS[<n>, <m>] zohledňují offsety, které jsou uloženy v proměnných \$NK_OFF a \$NK_OFF_FINE, tzn. že tyto úhly os se mohou v následujících blocích bez dalších úprav používat, aby byla nastavena požadovaná orientace. Pokud je kruhová osa Hirthovou osou, jsou pozice vyplývající z daného řešení zakrouhlovány na nejbližší následující stabilní polohu Hirthova ozubení. V systémových proměnných \$P_ORI_DIFF mohou být pro kruhové osy s Hirthovým ozubením načítány rozdíly poloh os, aby bylo možné pracovat s přesnými řešeními, a řešení, která jsou přizpůsobena stabilním polohám Hirthova ozubení.
\$P_ORI_DIFF [<n>, <m>]	Tato proměnná poskytuje rozdíly mezi přesnými polohami orientačních os a polohami těchto os, které jsou k dispozici v proměnné \$P_ORI_POS a které vyplývají z naprogramované orientace.
	<n>: Index řešení
	Rozsah hodnot: 0, 1
	<m>: Index orientační osy
	Rozsah hodnot: 0 ... 2 Posloupnost orientačních os (1 ... 3) se vztahuje na definici os v parametru \$NT_ROT_AX_NAME.
	Obsah může být jiný než nulový jen tehdy, pokud jsou polohy indexovány (Hirthovo ozubení), tzn. jestliže se systémový parametr \$NT_HIRTH_INCR příslušné osy nerovná nule a pokud se v případě této osy jedná o manuálně nastavovanou kruhovou osu.

Systémové proměnné	Význam																								
\$P_ORI_SOL	Jestliže se v případě transformace orientace s více než jednou orientační osou vypočítávají úhly os, které mají vést k nastavení zadané orientace, obecně existuje více než jedno řešení. V systémové proměnné \$P_ORI_SOL je obsažen počet platných řešení spolu s doplňkovými stavovými informacemi. Obsah proměnné \$P_ORI_SOL je kódován následujícím způsobem:																								
	<table> <tr> <td>Hodnoty < 0</td><td>Všeobecné chybové stavy</td></tr> <tr> <td>-1</td><td>Pro aktivní transformaci nebylo dosud vypočítáno žádné řešení (chybějící volání funkce ORISOLH).</td></tr> <tr> <td>-2</td><td>Není aktivní žádná transformace nebo aktivní transformace není transformací orientace (6-osá transformace), která by mohla poskytnout pozice odpovídající požadované naprogramované orientaci.</td></tr> <tr> <td>-4</td><td>Požadovaná orientace není prostřednictvím zadané kinematiky nastavitelná.</td></tr> <tr> <td>-5</td><td>Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ nebylo nalezeno žádné řešení.</td></tr> <tr> <td>-6</td><td>Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ je úhel γ příliš velký.</td></tr> <tr> <td>-7</td><td>Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byl zadán úhel, který prostřednictvím Hirthova ozubení není nastavitelný.</td></tr> <tr> <td>-8</td><td>V parametrech je nastaveno, že první orientační osa (osa rámu) je Hirthova osa.</td></tr> <tr> <td>-9</td><td>V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.</td></tr> <tr> <td>-10</td><td>Nebylo nalezeno žádné přizpůsobení řešení Hirthovu ozubení.</td></tr> <tr> <td>Hodnoty > 0 Místo jednotek</td><td> Počet matematicky možných řešení bez zohlednění mezních poloh os a případných chybových podmínek. <table> <tr> <td>0</td><td> Neexistuje žádné řešení, tzn. požadovaná orientace není nastavitelná. Tento případ může mít tři různé příčiny: - Požadovaná orientace je i při libovolném rozsahu pohybů orientačních os principiálně nedosažitelná, a to v důsledku dané kinematiky stroje (orientační osy nejsou vůči sobě v pravých úhlech). V tomto případě jsou na místě desítek a stovek proměnné \$P_ORI_SOL nuly, ve stavové proměnné \$P_ORI_STAT, která je přiřazena orientačním osám, je hodnota „-4“. - Vypočítaná řešení nejsou dosažitelná, protože by znamenala narušení mezních hodnot os. Polohy orientačních os, které by vyplynuly, kdyby mezní hodnoty os neplatily, je možno načíst v proměnné \$P_ORI_POS. - Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou, </td></tr> </table> </td></tr> </table>	Hodnoty < 0	Všeobecné chybové stavy	-1	Pro aktivní transformaci nebylo dosud vypočítáno žádné řešení (chybějící volání funkce ORISOLH).	-2	Není aktivní žádná transformace nebo aktivní transformace není transformací orientace (6-osá transformace), která by mohla poskytnout pozice odpovídající požadované naprogramované orientaci.	-4	Požadovaná orientace není prostřednictvím zadané kinematiky nastavitelná.	-5	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ nebylo nalezeno žádné řešení.	-6	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ je úhel γ příliš velký.	-7	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byl zadán úhel, který prostřednictvím Hirthova ozubení není nastavitelný.	-8	V parametrech je nastaveno, že první orientační osa (osa rámu) je Hirthova osa.	-9	V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.	-10	Nebylo nalezeno žádné přizpůsobení řešení Hirthovu ozubení.	Hodnoty > 0 Místo jednotek	Počet matematicky možných řešení bez zohlednění mezních poloh os a případných chybových podmínek. <table> <tr> <td>0</td><td> Neexistuje žádné řešení, tzn. požadovaná orientace není nastavitelná. Tento případ může mít tři různé příčiny: - Požadovaná orientace je i při libovolném rozsahu pohybů orientačních os principiálně nedosažitelná, a to v důsledku dané kinematiky stroje (orientační osy nejsou vůči sobě v pravých úhlech). V tomto případě jsou na místě desítek a stovek proměnné \$P_ORI_SOL nuly, ve stavové proměnné \$P_ORI_STAT, která je přiřazena orientačním osám, je hodnota „-4“. - Vypočítaná řešení nejsou dosažitelná, protože by znamenala narušení mezních hodnot os. Polohy orientačních os, které by vyplynuly, kdyby mezní hodnoty os neplatily, je možno načíst v proměnné \$P_ORI_POS. - Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou, </td></tr> </table>	0	Neexistuje žádné řešení, tzn. požadovaná orientace není nastavitelná. Tento případ může mít tři různé příčiny: - Požadovaná orientace je i při libovolném rozsahu pohybů orientačních os principiálně nedosažitelná, a to v důsledku dané kinematiky stroje (orientační osy nejsou vůči sobě v pravých úhlech). V tomto případě jsou na místě desítek a stovek proměnné \$P_ORI_SOL nuly, ve stavové proměnné \$P_ORI_STAT, která je přiřazena orientačním osám, je hodnota „-4“. - Vypočítaná řešení nejsou dosažitelná, protože by znamenala narušení mezních hodnot os. Polohy orientačních os, které by vyplynuly, kdyby mezní hodnoty os neplatily, je možno načíst v proměnné \$P_ORI_POS. - Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou,
Hodnoty < 0	Všeobecné chybové stavy																								
-1	Pro aktivní transformaci nebylo dosud vypočítáno žádné řešení (chybějící volání funkce ORISOLH).																								
-2	Není aktivní žádná transformace nebo aktivní transformace není transformací orientace (6-osá transformace), která by mohla poskytnout pozice odpovídající požadované naprogramované orientaci.																								
-4	Požadovaná orientace není prostřednictvím zadané kinematiky nastavitelná.																								
-5	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ nebylo nalezeno žádné řešení.																								
-6	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ je úhel γ příliš velký.																								
-7	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byl zadán úhel, který prostřednictvím Hirthova ozubení není nastavitelný.																								
-8	V parametrech je nastaveno, že první orientační osa (osa rámu) je Hirthova osa.																								
-9	V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.																								
-10	Nebylo nalezeno žádné přizpůsobení řešení Hirthovu ozubení.																								
Hodnoty > 0 Místo jednotek	Počet matematicky možných řešení bez zohlednění mezních poloh os a případných chybových podmínek. <table> <tr> <td>0</td><td> Neexistuje žádné řešení, tzn. požadovaná orientace není nastavitelná. Tento případ může mít tři různé příčiny: - Požadovaná orientace je i při libovolném rozsahu pohybů orientačních os principiálně nedosažitelná, a to v důsledku dané kinematiky stroje (orientační osy nejsou vůči sobě v pravých úhlech). V tomto případě jsou na místě desítek a stovek proměnné \$P_ORI_SOL nuly, ve stavové proměnné \$P_ORI_STAT, která je přiřazena orientačním osám, je hodnota „-4“. - Vypočítaná řešení nejsou dosažitelná, protože by znamenala narušení mezních hodnot os. Polohy orientačních os, které by vyplynuly, kdyby mezní hodnoty os neplatily, je možno načíst v proměnné \$P_ORI_POS. - Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou, </td></tr> </table>	0	Neexistuje žádné řešení, tzn. požadovaná orientace není nastavitelná. Tento případ může mít tři různé příčiny: - Požadovaná orientace je i při libovolném rozsahu pohybů orientačních os principiálně nedosažitelná, a to v důsledku dané kinematiky stroje (orientační osy nejsou vůči sobě v pravých úhlech). V tomto případě jsou na místě desítek a stovek proměnné \$P_ORI_SOL nuly, ve stavové proměnné \$P_ORI_STAT, která je přiřazena orientačním osám, je hodnota „-4“. - Vypočítaná řešení nejsou dosažitelná, protože by znamenala narušení mezních hodnot os. Polohy orientačních os, které by vyplynuly, kdyby mezní hodnoty os neplatily, je možno načíst v proměnné \$P_ORI_POS. - Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou,																						
0	Neexistuje žádné řešení, tzn. požadovaná orientace není nastavitelná. Tento případ může mít tři různé příčiny: - Požadovaná orientace je i při libovolném rozsahu pohybů orientačních os principiálně nedosažitelná, a to v důsledku dané kinematiky stroje (orientační osy nejsou vůči sobě v pravých úhlech). V tomto případě jsou na místě desítek a stovek proměnné \$P_ORI_SOL nuly, ve stavové proměnné \$P_ORI_STAT, která je přiřazena orientačním osám, je hodnota „-4“. - Vypočítaná řešení nejsou dosažitelná, protože by znamenala narušení mezních hodnot os. Polohy orientačních os, které by vyplynuly, kdyby mezní hodnoty os neplatily, je možno načíst v proměnné \$P_ORI_POS. - Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou,																								

Systémové proměnné	Význam		
			jejíž poloha má být vypočítána. V těchto případech poloha osy není definována.
		1	Existuje jedno řešení. Tento případ může mít tři různé příčiny: - Na základě zadané orientace a kinematiky stroje, a také bez zohlednění mezních poloh os, existuje jen jedno řešení (matematicky viděno se jedná o dvě vzájemně se překrývající řešení). Tento případ se vyskytuje u nepravoúhlých kinematik na okrajích oblastí, v nichž lze provádět nastavování orientace. Proměnná \$P_ORI_POS obsahuje obě identická řešení. - Existuje jen jedno řešení, protože druhé řešení je v důsledku narušení mezních hodnot polohy osy neplatné. Platné řešení je potom vždy prvním řešením v proměnné \$P_ORI_POS. Druhé řešení, které by vyplynulo, pokud by nebyly zohledňovány mezní hodnoty polohy osy, je v případě potřeby možné načíst z proměnné \$P_ORI_POS. - Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ se jedná o normální případ. K předem zadaným polohám druhé orientační osy obecně existuje jen jedna platná poloha orientační osy, která ještě nebyla vypočítána.
		2	Existují dvě řešení.
		8	Existuje nekonečně mnoho řešení, tzn. poloha orientační osy (osa procházející pólem) je libovolná. Z obou možných poloh ostatních os (osy) je však jedna kvůli narušení mezních hodnot osy vyloučena.
		9	Existuje nekonečně mnoho řešení, tzn. poloha orientační osy (osa procházející pólem) není stanovena. Osu, jejíž poloha není stanovena, je možno zjistit pomocí místa stovek nebo ze systémové proměnné \$P_ORI_STAT.
	Hodnoty > 0 Místo desítek	Bitově kódovaný údaj narušených mezních hodnot polohy osy. Přesnou příčinu chyby je možno zjistit ze systémové proměnné \$P_ORI_STAT.	
		Bit 0 (hodnota 10):	Pro minimálně jedno řešení došlo k narušení minimálně jedné mezní hodnoty polohy 1. orientační osy.
		Bit 1 (hodnota 20):	Pro minimálně jedno řešení došlo k narušení minimálně jedné mezní hodnoty polohy 2. orientační osy.
		Bit 2 (hodnota 40):	Pro minimálně jedno řešení došlo k narušení minimálně jedné mezní hodnoty polohy 3. orientační osy.
	Hodnoty > 0 Místo stovek	Bitově kódovaný údaj pro nedefinované polohy os (může se vyskytnout tehdy, když existuje nekonečně mnoho řešení, tzn. když je na místě jednotek hodnota „9“).	
		Bit 0 (hodnota 100):	Poloha 1. orientační osy není definována.

Systémové proměnné	Význam		
		Bit 1 (hodnota 200):	Poloha 2. orientační osy není definována.
		Bit 2 (hodnota 400):	Poloha 3. orientační osy není definována.
	Označení 1., 2. a 3. orientační osy se vztahuje na definici os v parametru \$NT_ROT_AX_NAME.		

Systémové proměnné	Význam
\$P_ORI_STAT [<n>]	Po zpracování příkazu ORISOLH obsahuje stavové informace pro každou z maximálně 3 orientačních os.
	<n>: Index orientační osy (odpovídá indexu příslušné orientační osy v parametru \$NT_ROT_AX_NAME)
	Rozsah hodnot: 0 ... 2
	Posloupnost orientačních os (1 ... 3) se vztahuje na definici os v parametru \$NT_ROT_AX_NAME.
	Obsah proměnné \$P_ORI_STAT je kódován následujícím způsobem:
	Hodnoty < 0 Všeobecné chybové stavy
	-1 Stavové informace nejsou definovány (chybějící volání funkce ORISOLH).
	-2 Není aktivní žádná transformace nebo aktivní transformace není transformací orientace (6-osá transformace), která by mohla poskytnout pozice odpovídající požadované naprogramované orientaci.
	-3 Daná osa není v aktivní transformaci obsažena.
	-4 Polohu osy není možné vypočítat, protože s uvedenou kinematikou je požadovaná orientace i v případě libovolně definovaného rozsahu pohybů osy nedosažitelná.
Hodnoty > 0 Místo jednotek	-5 Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou, jejíž poloha má být vypočítána. V těchto případech poloha osy není definována.
	-6 Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ je úhel γ příliš velký.
	-7 Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byl zadán úhel, který prostřednictvím Hirthova ozubení není nastavitelný.
	-8 V parametrech je nastaveno, že první orientační osa (osa rámu) je Hirthova osa.
	-9 V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.
	-10 Nebylo nalezeno žádné přizpůsobení řešení Hirthovu ozubení.
	Bitově kódovaný údaj narušených mezních hodnot polohy osy pro první řešení.
	Bit 0 (hodnota 1): První řešení narušuje dolní mezní hodnotu polohy osy.
	Bit 1 (hodnota 2): První řešení narušuje horní mezní hodnotu polohy osy.
	Bitově kódovaný údaj narušených mezních hodnot polohy osy pro druhé řešení.
Hodnoty > 0 Místo desítek	Bit 0 (hodnota 10): Druhé řešení narušuje dolní mezní hodnotu polohy osy.

Systémové proměnné	Význam		
		Bit 1 (hodnota 20):	Druhé řešení narušuje horní mezní hodnotu polohy osy.
	Hodnoty > 0	Údaj nedefinované polohy osy.	
	Místo stovek	Bit 0 (hodnota 100):	Poloha orientační osy její definována, tzn. požadovaná orientace je dosažena s jakýmkoli libovolným nastavením kruhové osy (osa prochází pólem). Tato informace je obsažena také v systémové proměnné \$P_ORI_SOL.
	Pokud jde o čísla chyb, které signalizují narušení mezních hodnot polohy osy, může se jich vyskytnout i více najednou. Protože se při narušení mezní hodnoty polohy osy systém pokouší přičítáním nebo odečítáním násobků 360° dosáhnout polohy v rámci přípustných mezních polohy osy, není v případě, že je to možné, jednoznačně definováno, zda došlo k narušení dolní nebo horní mezní hodnoty polohy osy. Jestliže pro požadovanou orientaci neexistuje žádné Řešení (\$P_ORI_SOL = 0), je stavu orientačních os podílejících se na transformaci přiřazena hodnota "0".		

Poznámka**\$NT_ROT_AX_NAME**

Pomocí této systémové proměnné je možno se odkazovat na maximálně 3 osy, které slouží pro nastavování orientace. Obsahuje názvy prvků řetězce (\$NK_NAME), které definují ty strojní osy (kruhové osy), jež musí uskutečňovat pohyby potřebné pro nastavení orientace, které vyplývají z dané kinematické transformace. Posloupnost, v jaké jsou maximálně tři kruhové osy v této systémové proměnné obsaženy, nemá pro kinematiku stroje žádný význam, protože tato posloupnost je odvozena ze struktury kinematického řetězce. Protože jste ale definovali posloupnost, kterou pro přístup ke kruhovým osám používají jiné proměnné, musí se pořadí orientačních os v proměnné \$NT_ROT_AX_NAME shodovat s popisem kinematiky.

Poznámka**Stavové informace**

Stavové informace, které např. ukazují, že orientace je nedosažitelná nebo že je dosažitelná jen v případě narušení určitých mezních hodnot polohy osy, nezpůsobují alarm NC systému. Je odpovědností uživatele, aby na uváděné podmínky vhodným způsobem reagoval.

11.11.2 Aktivování modifikací korekčních parametrů u točivých nástrojů (CUTMOD, CUTMODK)

Modifikace korekčních parametrů u točivých nástrojů jsou v NC programu aktivovány prostřednictvím příkazů jazyka CUTMOD (ve spojení s orientovatelnými držáky nástrojů), příp. CUTMODK (pro transformace orientace, které byly definovány prostřednictvím kinematického řetězce).

Poznámka

Protože orientovatelný držák nástroje a transformace orientace, které byly definovány prostřednictvím kinematického řetězce, nemohou být aktivní současně, nedochází ke konfliktům mezi oběma variantami.

Syntaxe

CUTMOD = <hodnota>

nebo

CUTMODK = <příkaz>

Význam

CUTMOD:	Volání funkce ve spojení s orientovatelnými držáky nástrojů		
<hodnota>:	Přiřazovaná hodnota		
	Datový typ:	INT	
	Hodnota:	0	Funkce je deaktivována. Hodnoty poskytované systémovými proměnnými \$P_AD... se rovnají odpovídajícím parametrům nástroje.
		> 0	Funkce se aktivuje, jestliže je orientovatelný držák nástroje s uvedeným číslem aktivní, tzn. aktivování je vázáno na určitý orientovatelný držák nástroje. Hodnoty poskytované systémovými proměnnými \$P_AD... jsou oproti odpovídajícím parametrům nástroje v případě potřeby modifikovány v závislosti na aktivním otočení. Deaktivováním uvedeného orientovatelného držáku nástroje se funkce dočasně deaktivuje, aktivováním jiného orientovatelného držáku nástroje se deaktivuje trvale. V prvním případě se proto funkce při opětovném vyvolání stejného orientovatelného držáku nástroje znovu aktivuje, ve druhém případě je nové vyvolání nezbytné, a to i tehdy, když se orientovatelný držák nástroje s příslušným číslem znovu aktivuje později. Reset systému nemá na funkci žádný vliv.
		-1	Funkce je aktivována vždy, když je aktivní orientovatelný držák nástroje. Při výměně držáku nástroje nebo při jeho deaktivování a pozdějším opětovném aktivování nemusí být funkce CUTMOD znovu nastavována.
		-2	Funkce je aktivována vždy, když je aktivován orientovatelný držák nástroje, jehož číslo se rovná momentálně aktivnímu orientovatelnému držáku nástroje. Jestliže žádný orientovatelný držák není aktivní, má tato situace stejný význam, jako kdyby bylo nastaveno CUTMOD=0. Jestliže je aktivní orientovatelný držák, má tato situace stejný význam, jako kdyby bylo bezprostředně zadáno číslo aktuálního držáku nástroje.
		< -2	Hodnoty menší než 2 jsou ignorovány, tzn. v tomto případě je situace stejná, jako kdyby příkaz CUTMOD nebyl vůbec naprogramován. Poznámka: Tento interval hodnot by se vůbec neměl používat, protože je rezervován pro eventuální pozdější rozšíření.
CUTMODK:	Volání funkce ve spojení s transformacemi orientace, které byly definovány prostřednictvím kinematického řetězce.		

<příkaz>:	Přiřazený příkaz		
	Datový typ:	STRING	
	Hodnota:	"NEW"	Do paměti se ukládají stavy aktivní transformace definované prostřednictvím kinematického řetězce, které jsou důležité pro funkci „Modifikace korekčních parametrů“, název transformace a frame aktuální kontury. Poznámka: Tento příkaz je přípustný jen tehdy, pokud je aktivní vhodná transformace (TRAORI_DYN, TRAORI_STAT nebo TRAANG_K).
		"OFF"	Deaktivování aktivní funkce „Modifikace korekčních parametrů“. Data, která byla předtím uložena pomocí příkazu „NEW“, zůstávají zachována. Poznámka: Tento příkaz je přípustný jen tehdy, pokud není aktivní příkaz CUTMODK. Je-li tomu tak, nemá žádnou funkci. Případně existující datový blok pro funkci „Modifikace korekčních parametrů“ zůstává zachován.
		"ON"	Pomocí tohoto příkazu se znovu aktivuje funkce „Modifikace korekčních parametrů“ s datovým blokem, který byl předtím uložen pomocí příkazu „NEW“. Jestliže je při zpracovávání tohoto příkazu aktivní transformace s názvem uloženého datového bloku, funkce „Modifikace korekčních parametrů“ je okamžitě v platnosti. Jinak se aktivování zpozdí, dokud není aktivní transformace aktivována.
		"CLEAR"	Stejně jako příkaz „OFF“ deaktivuje funkci „Modifikace korekčních parametrů“ a kromě toho ještě vymaže uložený datový blok. Poznámka: Tento příkaz je přípustný jen tehdy, pokud není aktivní příkaz CUTMODK.

Poznámka**SD42984 \$SC_CUTDIRMOD**

Funkce, která může být aktivována příkazem CUTMOD, příp. CUTMODK, nahrazuje funkci, která může být aktivována pomocí nastavovaného parametru SD42984 \$SC_CUTDIRMOD. Tato funkce však zůstává i nadále k dispozici nezměněna. Protože ale není rozumné využívat souběžně obě funkce, může být tato funkce aktivována jen tehdy, pokud je CUTMOD roven nule a CUTMODK se rovná nulovému řetězci.

Další informace

Načítání modifikovaných korekčních parametrů

Modifikované korekční parametry jsou k dispozici v následujících systémových proměnných a proměnných BTSS:

Význam	Systémová proměnná	Proměnná BTSS
Poloha bříty	\$P_AD[2]	cuttEdgeParam2
Úhel držáku	\$P_AD[10]	cuttEdgeParam10
Směr bříty	\$P_AD[11]	cuttEdgeParam11
Úhel volného řezání	\$P_AD[24]	cuttEdgeParam24

Tato data jsou oproti odpovídajícím parametrům nástroje (\$TC_DP2[...], ...) upravována vždy, když byla pomocí příkazu `CUTMOD`, příp. `CUTMODK` aktivována funkce „Modifikace korekčních parametrů u točivých nástrojů“ a když byl nástroj otočen prostřednictvím orientovatelného držáku nástroje, příp. pomocí vhodné transformace orientace.

Další systémové proměnné důležité pro tuto funkci

Systémová proměnná	Význam
\$P_CUTMOD_ANG / \$AC_CUTMOD_ANG	Obsahuje údaj úhlu, o který byl nástroj otočen v aktivní rovině obrábění a který u funkcí <code>CUTMOD</code> , příp. <code>CUTMODK</code> , představuje základ pro stanovení upravených parametrů bříty.
\$P_CUTMOD / \$AC_CUTMOD	Načítá momentální platnou hodnotu, která byla naposled naprogramována pomocí příkazu <code>CUTMOD</code> (číslo držáku nástroje, pro který má být aktivována modifikace korekčních parametrů). Jestliže byla naposled naprogramovaná hodnota pro funkci <code>CUTMOD=-2</code> (aktivování s momentálně aktivním orientovatelným držákem nástroje), v systémových proměnných není k dispozici hodnota „-2“, ale číslo orientovatelného držáku nástroje, který byl v okamžiku naprogramování aktivní.
\$P_CUTMODK / \$AC_CUTMODK	Načítá název transformace, pod kterou byl založen momentálně platný datový blok pro „modifikaci korekčních parametrů“.

Systémová proměnná	Význam																		
\$P_CUT_INV / \$AC_CUT_INV	Pokud je nástroj otočený tak, že směr otáčení vřetena musí být invertován, je výslednou hodnotou TRUE. Za tím účelem musí být v bloku, na který se příslušná operace čtení vztahuje, splněny následující podmínky: 1. Je aktivní soustružnický nebo brusný nástroj (typy nástrojů 400 až 599 a / nebo SD42950 \$SC_TOOL_LENGTH_TYPE = 2). 2. Modifikace korekčních parametrů byla aktivována příkazem CUTMOD, příp. CUTMODK. 3. Je aktivní orientovatelný držák nástroje, příp. transformace orientace definovaná pomocí kinematického řetězce, které byly zvoleny prostřednictvím příkazu CUTMOD, příp. CUTMODK. 4. Nástroj je s využitím orientovatelného držáku nástroje, příp. kinematické transformace orientace, otočen tak, aby výsledná normála břitu nástroje byla oproti výchozí poloze otočena o více než 90° (za obvyklých okolností 180°). Pokud alespoň jedna z výše uvedených čtyř podmínek není splněna, obsahuje proměnná hodnotu FALSE. Pro nástroje, jejichž poloha břitu není definována, obsahuje tato proměnná vždy hodnotu FALSE.																		
\$P_CUTMOD_ERR	Chybový stav po posledním volání funkce CUTMOD. Funkce CUTMOD může být vyvolávána také implicitně při výměně nástroje. Proměnná se při resetu nastavuje zpět na nulu. Při každé výměně nástroje se napřed resetuje a potom se jí v případě potřeby nastavuje nová hodnota. Tato proměnná je bitově kódována. Bity mají následující význam: <table border="1"> <tr> <td>Bit 0:</td><td>Pro aktivní nástroj není definován žádný platný směr břitu.</td></tr> <tr> <td>Bit 1:</td><td>Oba úhly břitu (úhel volného řezání a úhel držáku) aktivního nástroje jsou nulové.</td></tr> <tr> <td>Bit 2:</td><td>Úhel volného řezání aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 180^\circ$).</td></tr> <tr> <td>Bit 3:</td><td>Úhel držáku aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 90^\circ$).</td></tr> <tr> <td>Bit 4:</td><td>Úhel destičky aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 90^\circ$).</td></tr> <tr> <td>Bit 5:</td><td>Kombinace polohy břitu - úhel držáku aktivního nástroje je nepřipustná (jestliže je poloha břitu 1 až 4, musí být úhel držáku $\leq 90^\circ$, jestliže je poloha břitu 5 až 8, musí být úhel $\geq 90^\circ$).</td></tr> <tr> <td>Bit 6:</td><td>Nepřipustné otočení aktivního nástroje. Nástroj byl otočen z aktivní roviny obrábění o $\pm 90^\circ$ (s tolerancí přibližně 1°). V důsledku toho poloha břitu v rovině obrábění již není definována.</td></tr> <tr> <td>Bit 7:</td><td>Řezná destička se již nenachází v rovině obrábění a úhel mezi řeznou destičkou a rovinou obrábění překračuje horní mezní hodnotu definovanou pomocí nastavovaného parametru SD42998 \$SC_CUTMOD_PLANE_TOL.</td></tr> <tr> <td>Bit 8:</td><td>Řezná destička se nenachází v rovině obrábění. Absolutní hodnota úhlu α je větší než 1°. Úhel α je úhel otočení okolo souřadné osy, která je kolmá jak na osu otáčení, k níž je vztažen úhel β, tak také na osu otáčení, k níž je vztažen úhel γ (v případě G18 je to osa X).</td></tr> </table>	Bit 0:	Pro aktivní nástroj není definován žádný platný směr břitu.	Bit 1:	Oba úhly břitu (úhel volného řezání a úhel držáku) aktivního nástroje jsou nulové.	Bit 2:	Úhel volného řezání aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 180^\circ$).	Bit 3:	Úhel držáku aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 90^\circ$).	Bit 4:	Úhel destičky aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 90^\circ$).	Bit 5:	Kombinace polohy břitu - úhel držáku aktivního nástroje je nepřipustná (jestliže je poloha břitu 1 až 4, musí být úhel držáku $\leq 90^\circ$, jestliže je poloha břitu 5 až 8, musí být úhel $\geq 90^\circ$).	Bit 6:	Nepřipustné otočení aktivního nástroje. Nástroj byl otočen z aktivní roviny obrábění o $\pm 90^\circ$ (s tolerancí přibližně 1°). V důsledku toho poloha břitu v rovině obrábění již není definována.	Bit 7:	Řezná destička se již nenachází v rovině obrábění a úhel mezi řeznou destičkou a rovinou obrábění překračuje horní mezní hodnotu definovanou pomocí nastavovaného parametru SD42998 \$SC_CUTMOD_PLANE_TOL.	Bit 8:	Řezná destička se nenachází v rovině obrábění. Absolutní hodnota úhlu α je větší než 1° . Úhel α je úhel otočení okolo souřadné osy, která je kolmá jak na osu otáčení, k níž je vztažen úhel β , tak také na osu otáčení, k níž je vztažen úhel γ (v případě G18 je to osa X).
Bit 0:	Pro aktivní nástroj není definován žádný platný směr břitu.																		
Bit 1:	Oba úhly břitu (úhel volného řezání a úhel držáku) aktivního nástroje jsou nulové.																		
Bit 2:	Úhel volného řezání aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 180^\circ$).																		
Bit 3:	Úhel držáku aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 90^\circ$).																		
Bit 4:	Úhel destičky aktivního nástroje má nepřipustnou hodnotu ($< 0^\circ$ nebo $> 90^\circ$).																		
Bit 5:	Kombinace polohy břitu - úhel držáku aktivního nástroje je nepřipustná (jestliže je poloha břitu 1 až 4, musí být úhel držáku $\leq 90^\circ$, jestliže je poloha břitu 5 až 8, musí být úhel $\geq 90^\circ$).																		
Bit 6:	Nepřipustné otočení aktivního nástroje. Nástroj byl otočen z aktivní roviny obrábění o $\pm 90^\circ$ (s tolerancí přibližně 1°). V důsledku toho poloha břitu v rovině obrábění již není definována.																		
Bit 7:	Řezná destička se již nenachází v rovině obrábění a úhel mezi řeznou destičkou a rovinou obrábění překračuje horní mezní hodnotu definovanou pomocí nastavovaného parametru SD42998 \$SC_CUTMOD_PLANE_TOL.																		
Bit 8:	Řezná destička se nenachází v rovině obrábění. Absolutní hodnota úhlu α je větší než 1° . Úhel α je úhel otočení okolo souřadné osy, která je kolmá jak na osu otáčení, k níž je vztažen úhel β , tak také na osu otáčení, k níž je vztažen úhel γ (v případě G18 je to osa X).																		

\$P_...: Proměnné předběžného zpracování

\$AC_...\$: Proměnné hlavního zpracování

Všechny proměnné hlavního zpracování mohou být čteny v synchronních akcích. Přístup kvůli čtení z předběžného zpracování má za následek zastavení předběžného zpracování.

Změna roviny

Pro stanovení změněné polohy břitu, směru řezání a úhlu držáku, příp. úhlu volného řezání je určující sledování břitu v právě aktivní rovině (G17 - G19).

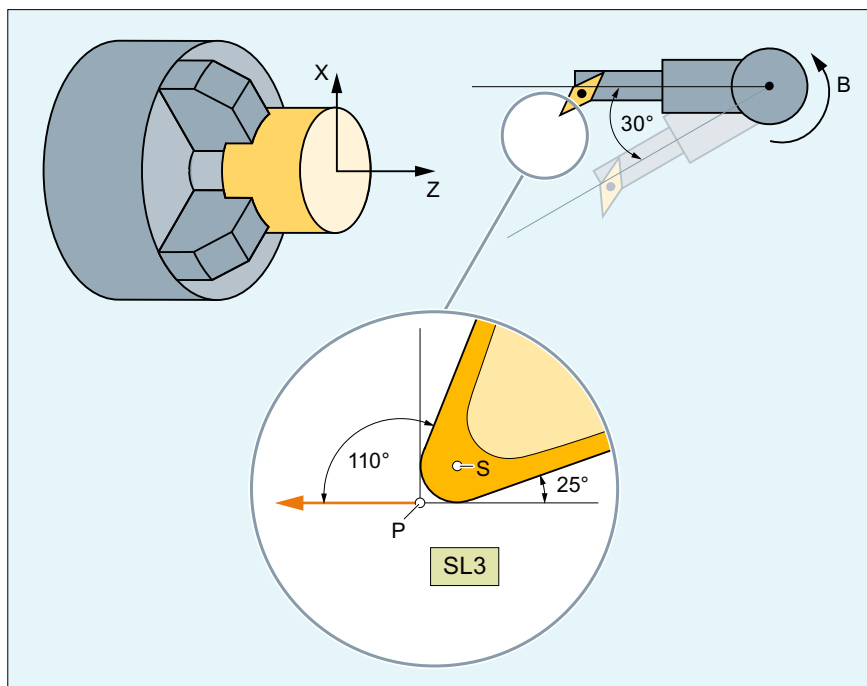
Jestliže však nastavovaný parametr SD42940 \$SC_TOOL_LENGTH_CONST (změna složek délky nástroje v případě změny roviny) obsahuje platnou hodnotu nerovnající se nule (plus nebo minus 17, 18 nebo 19), potom určuje její obsah rovinu, ve které jsou příslušné veličiny započítány.

Toto pravidlo dané nastavovaným parametrem mající přednost před G-kódem může být zneplatněno nastavením bitu 18 strojního parametru

\$MC_TOOL_PARAMETER_DEF_MASK. To znamená, že pokud je tento bit nastaven, platí vždy rovina definovaná příkazem G-funkce ze skupiny 6.

Platnost změněných parametrů břitu

Změněná poloha břitu a změněný vztažný bod břitu jsou při programování okamžitě v platnosti i pro již aktivní nástroj. Nové aktivování nástroje není proto nutné.



S: Střed břitu

P: Vztažný bod břitu

SL: Poloha břitu

Obrázek 11-4 Nástroj s polohou břitu 3 a na orientovatelný držák nástroje, který může nástroj otáčet okolo osy B.

Programový kód	Komentář		
N10 \$TC_DP1[1,1]=500			
N20 \$TC_DP2[1,1]=3	; Poloha břitu		
N30 \$TC_DP3[1,1]=12			
N40 \$TC_DP4[1,1]=1			
N50 \$TC_DP6[1,1]=6			
N60 \$TC_DP10[1,1]=110	; úhel držáku		
N70 \$TC_DP11[1,1]=3	; směr břitu		
N80 \$TC_DP24[1,1]=25	; úhel volného řezání		
N90 \$TC_CARR7[2]=0 \$TC_CARR8[2]=1 \$TC_CARR9[2]=0	; osa B		
N100 \$TC_CARR10[2]=0 \$TC_CARR11[2]=0	; osa C		
\$TC_CARR12[2]=1			
N110 \$TC_CARR13[2]=0			
N120 \$TC_CARR14[2]=0			
N130 \$TC_CARR21[2]=X			
N140 \$TC_CARR22[2]=X			
N150 \$TC_CARR23[2]="M"			
N160 TCOABS CUTMOD=0			
N170 G18 T1 D1 TCARR=2	; X	Y	Z
N180 X0 Y0 Z0 F10000	; 12.000	0.000	1.000
N190 \$TC_CARR13[2]=30			
N200 TCARR=2			
N210 X0 Y0 Z0	; 10.892	0.000	-5.134
N220 G42 Z-10	; 8.696	0.000	-17.330
N230 Z-20	; 8.696	0.000	-21.330
N240 X10	; 12.696	0.000	-21.330
N250 G40 X20 Z0	; 30.892	0.000	-5.134
N260 CUTMOD=2 X0 Y0 Z0	; 8.696	0.000	-7.330
N270 G42 Z-10	; 8.696	0.000	-17.330
N280 Z-20	; 8.696	0.000	-21.330
N290 X10	; 12.696	0.000	-21.330
N300 G40 X20 Z0	; 28.696	0.000	-7.330
N310 M30			

Numerické hodnoty v komentářích uvádějí vždy pozici na konci bloku v souřadném systému stroje (MCS) v posloupnosti X → Y → Z.

Vysvětlení

V bloku N180 je napřed aktivován nástroj s nastavením CUTMOD=0 a neotočeným orientovatelným držákem nástroje. Protože všechny offsetové vektory orientovatelného

držáku nástroje jsou 0, najede se na pozici, která odpovídá délkám nástroje uloženým v parametrech `$TC_DP3[1,1]` a `$TC_DP4[1,1]`.

V bloku N200 se aktivuje orientovatelný držák nástroje s otočením o 30° okolo osy B. Protože poloha břitu kvůli nastavení `CUTMOD=0` není změněna, je stejná jako před starým vztažným bodem břitu. Z tohoto důvodu se v bloku N210 najíždí na pozici, která v počátku (nule) obsahuje starý vztažný bod břitu (tzn. vektor (1, 12) je v rovině Z/X otočen o 30°).

V bloku N260 je na rozdíl od bloku N200 aktivní příkaz `CUTMOD=2`. V důsledku otočení orientovatelného držáku nástroje se stává platnou změněná poloha břitu 8. Z toho vyplývají také odlišné polohy os.

V blocích N220, příp. N270 je aktivována korekce rádiusu nástroje. Odlišné polohy břitu v obou úsecích programu nemají na koncovou pozici v blocích, v nichž je korekce rádiusu nástroje aktivována, žádný vliv, odpovídající polohy jsou proto identické. Teprve po blocích s deaktivováním N260, příp. N300 se odlišné polohy břitu opět projeví.

11.12 Práce s nástrojovým prostředím

Přehled funkcí

- Uložení nástrojového prostředí (TOOLENV) (Strana 487)
- Vymazání nástrojového prostředí (DELTOOLENV) (Strana 490)
- Načtení T-čísla, D-čísla a DL-čísla (GETTENV) (Strana 491)
- Načtení délek nástroje, příp. složek délky nástroje (GETTCOR) (Strana 492)
- Změna komponent nástroje (SETTCOR) (Strana 499)

Přehled systémových proměnných

- Načítání informací týkajících se uložených nástrojových prostředí (\$P_TOOLENVN, \$P_TOOLENV) (Strana 492)

11.12.1 Uložení nástrojového prostředí (TOOLENV)

Funkce TOOLENV slouží pro uložení do paměti všech aktuálních stavů, které jsou důležité pro vyhodnocování parametrů nástroje uchovávaných v paměti.

Jedná se o následující data:

- Aktivní příkaz G-funkce ze skupiny:
 - 6 (G17, G18, G19)
 - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- Aktivní příčná osa
- Strojní parametry:
 - MD18112 \$MN_MM_KIND_OF_SUMCORR (vlastnosti součtové korekce v oblasti TO)
 - MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (definice parametru nástroje)

- Nastavované parametry:
 - SD42900 \$SC_MIRROR_TOOL_LENGTH (změna znaménka délky nástroj při zrcadlovém převrácení)
 - SD42910 \$SC_MIRROR_TOOL_WEAR (změna znaménka opotřebení nástroje při zrcadlovém převrácení)
 - SD42920 \$SC_WEAR_SIGN_CUTPOS (znaménko opotřebení u nástrojů se zařízeními pro nastavení břitu)
 - SD42930 \$SC_WEAR_SIGN (znaménko opotřebení)
 - SD42935 \$SC_WEAR_TRANSFORM (transformace pro komponenty nástroje)
 - SD42940 \$SC_LENGTH_CONST (změna délkové složky rozměrů nástroje při změně roviny)
 - SD42942 \$SC_TOOL_LENGTH_CONST_T (změna délkové složky rozměrů nástroje při změně roviny pro soustružnické nástroje)
 - SD42950 \$SC_TOOL_LENGTH_TYPE (přiřazení složek délky nástroje nezávisle na typu nástroje)
 - SD42954 \$SC_TOOL_ORI_CONST_M (výměna složek orientace pro frézovací nástroje pro změně roviny)
 - SD42956 \$SC_TOOL_ORI_CONST_T (výměna složek orientace pro soustružnické nástroje pro změně roviny)
- Součást aktuálního celkového framu popisující orientaci (otočení a zrcadlové převrácení, žádná posunutí počátku nebo změny měřítka)
- Součást popisující orientaci a výsledná délka aktivního orientovatelného držáku nástroje
- Součást popisující orientaci a výsledná délka aktivní transformace

Kromě výše uvedených dat, která popisují nástrojové prostředí, jsou ukládány také T-číslo, D-číslo a DL-číslo aktivního nástroje, aby se později bylo možné dostat k tomuto nástroji ve stejném prostředí, jako při vyvolání příkazu TOOLENV, aniž by se nástroj musel znovu popisovat.

Syntaxe

<Status> = TOOLENV(<název>)

Význam

TOOLENV(. . .):	Předem definovaná funkce pro uložení nástrojového prostředí	
	Samostatný blok:	ano

<Status>:		Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
		Datový typ: INT		
		Hodnota:	0	Funkce je v pořádku
			-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí“ není k dispozici.
			-2	Pro nástrojové prostředí už nejsou v paměti k dispozici žádná volná místa.
			-3	Nulový řetězec je jako název nástrojového prostředí nepřipustný.
		-4	Není zadán žádný parametr (<název>).	
Parametry				
1	<název>:	Název, pod kterým má být aktuální datový blok uložen. Jestliže datový blok se stejným názvem už existuje, bude přepsán. Status je v tomto případě „0“.		
		Datový typ:	STRING	

Další informace

Základní rozměr/rozměr adaptéru - korekce délky nástroje

To, zda do výpočtu délky nástroje vstupuje délka adaptéru nebo základní rozměr (parametr konkrétního bříty \$TC_DP21, \$TC_DP22 a \$TC_DP23), rozhoduje v případě aktivní správy zásobníku nástrojů (k dispozici pouze spolu s volitelným doplňkem „Správa nástrojů“) hodnota strojního parametru:

MD18104 \$MN_MM_NUM_TOOL_ADAPTER (adaptér nástroje v oblasti TO)

Protože změna tohoto strojního parametru vstupuje v platnost pouze při spouštění řídicího systému, do nástrojového prostředí se neukládá.

Výsledná délka orientovatelných držáků nástroje a transformace

Poznámka

Jak v případě orientovatelných držáků nástroje, tak také u transformací existují systémové proměnné, příp. strojní parametry, které se chovají jako doplňkové složky délky nástroje a které jsou zcela nebo částečně podřízeny otočením, která se mají uskutečnit. Doplňkové složky délky nástroje, které takto vznikají, musí být při volání příkazu TOOLENV ukládány také, protože tvoří součást prostředí, ve kterém se nástroj používá.

Transformace adaptéru

Transformace adaptéru je vlastností adaptéru nástroje a tedy také celého nástroje. Není proto součástí nástrojového prostředí, které může být použito i na nějaký jiný nástroj.

Prostřednictvím uložení veškerých dat, která jsou zapotřebí pro stanovení celkové délky nástroje, je možné později vypočítat efektivní délku nástroje, a to i když v tomto okamžiku nástroj už není aktivní, příp. když se změnila okolní podmínky (např. příkazy G-funkcí nebo nastavované parametry). Stejně tak je možné vypočítat efektivní délku jiného nástroje za

předpokladu, že by byl použit za stejných podmínek jako nástroj, pro který byly stavové parametry uloženy.

Maximální počet datových bloků pro nástrojové prostředí

Prostřednictvím strojního parametru MD18116 \$MN_MM_NUM_TOOL_ENV se určuje, kolik datových bloků pro popis nástrojových prostředí může být maximálně uloženo. Tato data jsou uložena v oblasti TOA. Zůstávají zachována i po vypnutí řídicího systému.

Ukládání dat není možné. To znamená, že tato data není možné přenášet mezi různými řídicími systémy.

11.12.2 Vymazání nástrojového prostředí (DELTOOLENV)

Pomocí funkce DELTOOLENV můžete vymazat datové bloky pro popis nástrojového prostředí. Vymazání znamená, že k datovému bloku, který byl uložen pod určitým názvem, se již není možné dostat (pokus o přístup má za následek aktivování alarmu).

Poznámka

Datové bloky je možné vymazat pouze pomocí funkce DELTOOLENV, načtením souboru INITIAL.INI nebo prostřednictvím studeného startu (spuštění NC systému se standardními strojními parametry). Neexistují žádné další operace automatického mazání.

Syntaxe

```
<Status> = DELTOOLENV (<název>)
<Status> = DELTOOLENV ()
```

Význam

DELTOOLENV (...):		Předem definovaná funkce pro vymazání nástrojového prostředí	
		Samostatný blok:	ano
<Status>:		Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.	
		Datový typ:	INT
		Hodnota:	0 Funkce je v pořádku
			-1 Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojové prostředí“ není k dispozici.
		-2	Nástrojové prostředí se zadaným názvem už neexistuje.
Parametry			
1	<název>:	Název datového bloku, který má být vymazán	
		Datový typ:	STRING
DELTOOLENV ():			
Příkaz DELTOOLENV () bez udání názvu vymaže všechny datové bloky pro popis nástrojového prostředí			

11.12.3 Načtení T-čísla, D-čísla a DL-čísla (GETTENV)

Funkce GETTENV slouží k tomu, aby bylo možno načíst T-, D- a DL-čísla, která byla uložena v nástrojovém prostředí.

Syntaxe

<Status> = GETTENV(<název>, <TDDL>)

Význam

GETTENV (. . .) :		Předem definovaná funkce pro načítání T-čísel, D.-čísel a DL-čísel v datovém bloku pro popis nástrojového prostředí		
		Samostatný blok:	ano	
<Status>:		Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
		Datový typ:	INT	
		Hodnota:	0	Funkce je v pořádku
			-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí“ není k dispozici.
			-2	Nástrojové prostředí se zadaným názvem už neexistuje.
Parametry				
1	<název>:	Název datového bloku, ze kterého se má T-číslo, D-číslo a DL-číslo načíst.		
		Datový typ:	STRING	
2	<TDDL>:	Pole tohoto výsledného parametru obsahují T-číslo, D-číslo a DL-číslo nástroje, jehož nástrojové prostředí je v uvedeném datovém bloku uloženo: • <TDDL> [0]: T-číslo • <TDDL> [1]: D-číslo • <TDDL> [2]: DL-číslo		
		Datový typ:	INT[3]	
GETTENV (, <TDDL>) , GETTENV (" " , <TDDL>) :		Při vyvolávání funkce GETTENV je přípustné první parametr vypustit nebo jako první parametr předávat nulový řetězec. V obou těchto zvláštních případech funkce v proměnné <TDDL> poskytuje T-číslo, D-číslo a DL-číslo aktivního nástroje.		

11.12.4 Načítání informací týkajících se uložených nástrojových prostředí (\$P_TOOLENVN, \$P_TOOLENV)

Informace týkající se uložených nástrojových prostředí je možno načítat pomocí následujících systémových proměnných:

\$P_TOOLENVN:	Poskytuje počet datových bloků pro popis nástrojových prostředí definovaných (a ještě nevymazaných) pomocí příkazu TOOLENV.			
	Syntaxe:	<n> = \$P_TOOLENVN		
	Význam:	<n>:	Počet definovaných datových bloků	
			Datový typ:	INT
			Rozsah hodnot:	0 ... MD18116 \$MN_MM_NUM_TOOL_ENV
Přístup k této systémové proměnné je přípustný také tehdy, když žádná nástrojová prostředí nejsou možná (MD18116 = 0). V tomto případě funkce vrací hodnotu „0“.				
\$P_TOOLENV:	Poskytuje název <i>-tého datového bloku pro popis nástrojového prostředí			
	Syntaxe:	<název> = \$P_TOOLENV [<i>]		
	Význam:	<název>:	Název datového bloku s číslem <i>	
			Datový typ:	STRING
		<i>:	Číslo datového bloku	
			Datový typ:	INT
			Rozsah hodnot:	1 ... \$P_TOOLENVN
Přiřazování čísel datovým blokům není pevné, může se v důsledku mazání a zakládání nových datových bloků měnit. Číslování datových bloků se uskutečňuje interně. Pokud index <i> odkazuje na datový blok, který není definován, funkce vrací nulový řetězec. Jestliže je index <i> neplatný, tzn. <i> je menší než 1 nebo větší, než je maximální počet datových bloků pro nástrojová prostředí (MD18116 \$MN_MM_NUM_TOOLENV), aktivuje se následující alarm: Alarm 17020 „Nepřípustný index1 pole“				

11.12.5 Načtení délek nástroje, příp. složek délky nástroje (GETTCOR)

Funkce GETTCOR slouží pro načítání délek, příp. složek délky nástroje.

Přitom může být prostřednictvím nastavení parametrů zadáno, které komponenty mají být zohledňovány, a při jakých podmínkách použití má být nástroj monitorován.

Syntaxe

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```


Význam

GETTCOR (. . .) :	Předem definovaná funkce pro načítání délek nástroje, příp. složek délky nástroje		
	Samostatný blok:		ano
<Status>:	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:		INT
	Hodnota:	0	Funkce je v pořádku
		-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí“ není k dispozici.
		-2	Nástrojové prostředí s názvem zadaným v parametru <Stat> neexistuje.
		-3	Neplatný řetězec v parametru <Comp>. Příčinou této chyby může být neplatný nebo dvojité naprogramovaný znak.
		-4	Neplatné T-číslo
		-5	Neplatné D-číslo
		-6	Neplatné DL-číslo
		-7	Pokus o přístup do paměťového modulu, který není k dispozici.
		-8	Pokus o přístup k volitelnému doplňku, který není k dispozici (např. programovatelná orientace nástroje, správa nástrojů).
		-9	Řetězec <Comp> obsahuje dvojtečku (identifikace pro specifikace souřadného systému), nenásleduje ale žádné platné znaky pro popis požadovaného souřadného systému.
Parametry			

1	<Len>:	Výsledný vektor:	
		Datový typ:	REAL[11]
		Vektorové složky jsou uspořádány v následující posloupnosti: • <Len> [0]: Typ nástroje • <Len> [1]: Poloha břitu • <Len> [2]: Abscisa • <Len> [3]: Ordináta • <Len> [4]: Aplikáta • <Len> [5]: Rádus nástroje Jako vztažný souřadný systém pro komponenty délky platí souřadný systém definovaný v parametrech <Comp> a <Stat>. Pokud v parametru <Comp> žádný souřadný systém definován není, budou se délky nástroje zobrazovat v souřadném systému stroje. Přiřazení abscisy, ordináty a aplikáty ke geometrickým osám závisí na rovině, která je v použitém nástrojovém prostředí aktivní. Tzn. v případě G17 je abscisa rovnoběžná s osou X, v případě G18 s osou Z atd. Složky <Len>[6] až <Len>[10] obsahují doplňkové parametry, které mohou být zadávány za účelem popisu geometrie nástroje (např. \$TC_DP7 až \$TC_DP11 pro geometrii, příp. odpovídající komponenty pro opotřebení, příp. součtové a seřizovací korekce). Těchto 5 doplňkových prvků a rádus nástroje jsou definovány pouze pro složky E, G, S a W. Jejich vyhodnocování na parametru <Stat> nezávisí. Odpovídající hodnoty v parametrech <Len>[6] až <Len>[10] mohou být proto nenulové jen tehdy, pokud se na výpočtu délky nástroje podílí minimálně jedna z těchto čtyř jmenovaných složek. Zbývající složky nemají na výsledek žádný vliv. Údaje rozměrů jsou vztaženy na základní systém řídicího systému (palce nebo metrické jednotky).	

2	<Comp>:	Složky délky nástroje (volitelné)																		
		Datový typ:	STRING																	
		Řetězec znaků skládající se ze dvou dílčích řetězců, které jsou od sebe odděleny dvojtečkou.																		
		Obecný formát: "<SubStr_1> [: <SubStr_2>]"																		
		<SubStr_1>:	První dílčí řetězec označuje složky délky nástroje, které mají být zohledňovány při výpočtu délky nástroje. Posloupnost znaků v dílčím řetězci, stejně jako způsob zápisu (malá nebo velká písmena), jsou libovolné. Mezi znaky může být vložen libovolný počet mezer nebo znaky tabulátoru (netisknutelné znaky). Upozornění: Znaky v dílčím řetězci nesmí být naprogramovány dvakrát!																	
		Znaky:	<table><tr><td>-</td><td>Znak mínus (přípustný jen jako první znak) Bude vypočítána celková délka nástroje mínus složky, které jsou specifikovány v následujícím řetězci.</td></tr><tr><td>C</td><td>Rozměr adaptéru nebo základní rozměr (použije se ta z obou alternativně existujících složek, která je pro používaný nástroj aktivní)</td></tr><tr><td>E</td><td>Korekční parametry pro seřizování</td></tr><tr><td>G</td><td>Geometrie</td></tr><tr><td>K</td><td>Kinematická transformace (vyhodnocuje se pouze v případě generické transformace se 3, 4 a 5 osami!)</td></tr><tr><td>S</td><td>Součtové korekce</td></tr><tr><td>T</td><td>Orientovatelný držák nástroje</td></tr><tr><td>W</td><td>Opotřebení</td></tr></table>		-	Znak mínus (přípustný jen jako první znak) Bude vypočítána celková délka nástroje mínus složky, které jsou specifikovány v následujícím řetězci.	C	Rozměr adaptéru nebo základní rozměr (použije se ta z obou alternativně existujících složek, která je pro používaný nástroj aktivní)	E	Korekční parametry pro seřizování	G	Geometrie	K	Kinematická transformace (vyhodnocuje se pouze v případě generické transformace se 3, 4 a 5 osami!)	S	Součtové korekce	T	Orientovatelný držák nástroje	W	Opotřebení
		-	Znak mínus (přípustný jen jako první znak) Bude vypočítána celková délka nástroje mínus složky, které jsou specifikovány v následujícím řetězci.																	
		C	Rozměr adaptéru nebo základní rozměr (použije se ta z obou alternativně existujících složek, která je pro používaný nástroj aktivní)																	
		E	Korekční parametry pro seřizování																	
		G	Geometrie																	
K	Kinematická transformace (vyhodnocuje se pouze v případě generické transformace se 3, 4 a 5 osami!)																			
S	Součtové korekce																			
T	Orientovatelný držák nástroje																			
W	Opotřebení																			
Pokud je první dílčí řetězec prázdný (s výjimkou mezer), znamená to, že má být vypočítána celková délka nástroje při zohlednění všech složek. Totéž platí i tehdy, když parametr <Comp> není vůbec uveden.																				
<Substr_2>:	Nepovinný druhý dílčí řetězec označuje souřadný systém, ve kterém má být délka nástroje udána. Druhý dílčí řetězec se skládá z jednoho jediného relevantního znaku.																			
Znaky:	<table><tr><td>A</td><td>Nastavitelný souřadný systém (ACS)</td></tr><tr><td>B</td><td>Základní souřadný systém (BCS)</td></tr><tr><td>K</td><td>Souřadný systém nástroje kinematické transformace (KCS)</td></tr><tr><td>M</td><td>Souřadný systém stroje (MCS)</td></tr><tr><td>T</td><td>Souřadný systém nástroje (TCS)</td></tr><tr><td>W</td><td>Souřadný systém obrobku (WCS)</td></tr></table>		A	Nastavitelný souřadný systém (ACS)	B	Základní souřadný systém (BCS)	K	Souřadný systém nástroje kinematické transformace (KCS)	M	Souřadný systém stroje (MCS)	T	Souřadný systém nástroje (TCS)	W	Souřadný systém obrobku (WCS)						
A	Nastavitelný souřadný systém (ACS)																			
B	Základní souřadný systém (BCS)																			
K	Souřadný systém nástroje kinematické transformace (KCS)																			
M	Souřadný systém stroje (MCS)																			
T	Souřadný systém nástroje (TCS)																			
W	Souřadný systém obrobku (WCS)																			
Jestliže není uveden žádný souřadný systém, uskutečňuje se vyhodnocování v souřadném systému stroje (MKS). Otočení, která je v případě potřeby nutno započítat, jsou stanovena prostřednictvím nástrojového prostředí definovaného v parametru <Stat>.																				

3	<_Stat>:	Název datového bloku pro popis nástrojového prostředí (volitelné)
		Datový typ: STRING
		Jestliže se v tomto parametru nachází nulový řetězec („") nebo pokud není udán, použije se aktuální stav. Pokud není specifikován žádný nástroj, použije se ten aktuální.
4	<T>:	Interní T-číslo nástroje (nepovinné)
		Datový typ: INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, použije se nástroj určený v parametru <Stat>. Je-li hodnota tohoto parametru „-1“, použije se T-číslo aktivního nástroje. Je rovněž přípustné číslo aktivního nástroje explicitně zadat. Upozornění: Pokud parametr <Stat> není uveden, jako nástrojové prostředí se použije momentální stav. Protože zadání <T> = 0 znamená odkaz na T-číslo uložené v nástrojovém prostředí, použije se v tomto případě aktivní nástroj, což znamená, že zadání <T> = 0 a <T> = -1 mají za těchto speciálních okolností stejný význam.
5	<D>:	Číslo břitu nástroje (nepovinné)
		Datový typ: INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, řídí se použité D-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj.
6	<DL>:	Číslo korekce závislé na místě (nepovinné)
		Datový typ: INT
		Jestliže tento parametr není uveden, řídí se použité DL-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj.

Příklady

GETTCOR (_LEN)	Vypočítá se délka momentálně aktivního nástroje v souřadném systému stroje, přičemž budou zohledněny všechny složky.
GETTCOR (_LEN, "CGW:W")	Vypočítá se délka nástroje skládající se z rozměru adaptéru, příp. základního rozměru, geometrie a opotřebení pro aktivní nástroj. Další složky, jako např. orientovatelný držák nástroje nebo kinematická transformace, nejsou zohledňovány. Výsledek je uveden v souřadném systému obrobku.
GETTCOR (_LEN, "-K:B")	Vypočítává se celková délka aktivního nástroje bez zohlednění délkových složek eventuálně aktivní kinematické transformace. Výstup je v základním souřadném systému.
GETTCOR (_LEN, ":M", "Testenv1", , 3)	Vypočítává se celková délka nástroje pro nástroj uložený v nástrojovém prostředí s názvem „Testenv1“ v souřadném systému stroje. Výpočet se ale uskutečňuje nezávisle na uloženém čísle břitu pro břit D3.

Další informace

Transformace adaptéru / orientovatelný držák nástroje / kinematická transformace

Otočení a záměny složek, které v případě potřeby vyplývají z transformace adaptéru, orientovatelného držáku nástroje a kinematické transformace, jsou součástí nástrojového prostředí. Z tohoto důvodu jsou vždy započítávány, a to i tehdy, když odpovídající složky délky nemají být brány v úvahu. Pokud to není požadováno, musí být definována nástrojová prostředí, v nichž odpovídající transformace nejsou aktivní. V mnoha případech (jmenovitě vždy, když se na stroji nepoužívají žádné transformace, příp. žádný orientovatelný držák nástroje) splňují ukládané datové bloky pro nástrojová prostředí tyto podmínky automaticky, takže uživatel na ně nemusí brát žádný zvláštní zřetel.

Soustružnické a brusné nástroje: Výpočet délky nástroje v závislosti na parametru MD20360 \$MC_TOOL_PARAMETER_DEF_MASK

To, jak je u soustružnických a brusných nástrojů zapotřebí vyhodnocovat opotřebení, příp. délku nástroje ve směru eventuálně existující osy, ve které je průměr nástroje, je definováno ve strojním parametru:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (definice parametru nástroje)

Bit	Hodnota
0	U soustružnických a brusných nástrojů se parametr opotřebení v příčné ose započítává jako hodnota průměru.
	= 0 (předdefinovaná hodnota) ne
	= 1 ano
1	U soustružnických a brusných nástrojů se délková složka rozměrů nástroje v příčné ose započítává jako hodnota průměru.
	= 0 (předdefinovaná hodnota) ne
	= 1 ano

Jestliže jsou příslušné bity nastaveny, vyhodnocuje se odpovídající údaj s faktorem 0,5. Toto vyhodnocení se projevuje také v délce nástroje poskytované funkcí GETTCOR.

Příklad:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 \$MC_DIAMETER_AX_DEF (geometrická osa s funkcí příčné osy) = "X"

X je osa, ve které je průměr (standardní konfigurace soustruhu)

Programový kód	Komentář
N30 \$TC_DP1[1,1]=500	
N40 \$TC_DP2[1,1]=2	
N50 \$TC_DP3[1,1]=3.0	; geometrie L1
N60 \$TC_DP4[1,1]=4.0	
N70 \$TC_DP5[1,1]=5.0	
N80 \$TC_DP12[1,1]=12.0	; opotřebení L1
N90 \$TC_DP13[1,1]=13.0	

Programový kód	Komentář
N100 \$TC_DP14[1,1]=14.0	
N110 T1 D1 G18	
N120 R1=GETTCOR(_LEN,"GW")	
N130 R3=_LEN[2]	; 17.0 (= 4.0 + 13.0)
N140 R4=_LEN[3]	; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0)
N150 R5=_LEN[4]	; 19.0 (= 5.0 + 14.0)
N160 M30	

Složky délky v rámci kinematické transformace a orientovatelného držáku nástroje

Pokud je v případě výpočtu délky nástroje zohledňován **orientovatelný držák nástroje**, vstupují do výpočtu délky nástroje následující vektory:

Typ	Vektory
M	I1 a I2
T	I1, I2 a I3
P	Délka nástroje není prostřednictvím orientovatelného držáku nástroje ovlivňována.

V případě generické **5-osé transformace** vstupují u typů transformace 24 a 56 do výpočtu délky nástroje následující strojní parametry:

Typ transformace	Strojní parametry
24	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 MD24558/24658 \$MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2

Typ transformace 56 (pohyblivý nástroj a pohyblivý obrobek) odpovídá typu M v případě orientovatelného držáku nástroje.

U této transformace s 5 osami odpovídá v dosavadních verzích programového vybavení vektor MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 (vektor kinematického offsetu 1./2. transformaci s 5 osami v kanálu) součtu obou vektorů I₁ a I₃ u orientovatelného držáku nástroje typu M.

Pro transformaci je v obou případech důležitý pouze součet. Způsob, jakým se skládá z obou jednotlivých komponent, nemá žádný význam. Při výpočtu délky nástroje je ale důležité, která složka je přiřazena nástroji a která nástrojovému stolu. Z tohoto důvodu byl zaveden strojní parametr MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 (vektor kinematického offsetu ve stole). Odpovídá vektoru I₃. Strojní parametr MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 nyní již neodpovídá součtu vektorů I₁ a I₃, nýbrž pouze vektoru I₁. Jestliže je strojní parametr MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 roven nule, je toto chování zcela shodné s dřívějším stavem.

Kompatibilita

Funkce GETTCOR slouží spolu s funkcemi TOOLENV a SETTCOR mimo jiné k tomu, aby nahradily část funkcí, které byly až dosud realizovány externě v rámci měřicích cyklů.

V měřicích cyklech byla vyhodnocována pouze část parametrů, které nakonec určovaly efektivní délku nástroje. Pro uváděné funkce mohou být parametry nastaveny tak, že chování měřicích cyklů, pokud jde o výpočet délky nástroje, může být reprodukováno.

11.12.6 Změna komponent nástroje (SETTCOR)

Funkce SETTCOR slouží k tomu, aby bylo možné směnit složky nástroje při zohledňování všech okrajových podmínek, které se na vyhodnocování jednotlivých složek podílejí.

Poznámka

Poznámka k terminologii: Jestliže je v následujícím textu v souvislosti s délkou nástroje řeč o složkách nástroje, jsou míněny složky vektorové povahy, ze kterých se celková délka nástroje skládá (např. geometrie nebo opotřebení). Taková složka se proto skládá ze tří jednotlivých hodnot (L1, L2, L3), které jsou v následujícím textu označovány jako hodnoty souřadnic.

Složka nástroje označovaná jako „geometrie“ se tak například skládá ze tří hodnot souřadnic \$TC_DP3 až \$TC_DP5.

Syntaxe

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>, <Stat>, <T>, <D>, <DL>])
```

Význam

SETTCOR(...):	Předem definovaná funkce pro změnu složek nástroje	
	Samostatný blok:	ano

<Status>:		Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
		Datový typ: INT		
		Hodnota:	0	Funkce je v pořádku
			-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí“ není k dispozici.
			-2	Nástrojové prostředí s názvem zadaným v parametru <Stat> neexistuje.
			-3	Neplatný řetězec v parametru <Comp>. Příčinou této chyby může být neplatný nebo dvojité naprogramovaný znak.
			-4	Neplatné T-číslo
			-5	Neplatné D-číslo
			-6	Neplatné DL-číslo
			-7	Pokus o přístup do paměťového modulu, který není k dispozici.
			-8	Pokus o přístup k volitelnému doplňku, který není k dispozici (např. programovatelná orientace nástroje, správa nástrojů).
			-9	Nepřípustná číselná hodnota pro parametr <CorComp>.
			-10	Nepřípustná číselná hodnota pro parametr <CorMode>.
			-11	Údaje obsažené v parametrech <Comp> a <CorComp> si vzájemně odporují.
			-12	Údaje obsažené v parametrech <Comp> a <CorMode> si vzájemně odporují.
			-13	Obsah parametru <GeoAx> neoznačuje žádnou geometrickou osu.
			-14	Pokus o zápis do seřizovacího korekčního parametru, který není k dispozici.
Parametry				
1	<CorVal>:	Vektor korekce V souřadném systému obrobku (WCS) definovaném prostřednictvím parametru <Stat> platí následující přiřazení: • <CorVal> [0]: Abscisa • <CorVal> [1]: Ordináta • <CorVal> [2]: Aplikáta Jestliže má být korekce uskutečněna jen v jedné složce nástroje (tzn. žádná vektorová korekce, viz parametr <CorMode>), nachází se hodnota korekce vždy v <CorVal>[0] bez ohledu na to, na kterou osu je uplatňována. Obsah obou zbývajících složek se potom nevyhodnocuje. Pokud se vztahuje hodnota v parametru <CorVal>, příp. jedna ze složek parametru <CorVal>, na příčnou osu, je tento údaj vyhodnocován jako rozměr rádiusu. To znamená, že pokud je nástroj např. o uvedený rozměr „delší“, způsobí to dvakrát tak velkou změnu průměru obrobku. Údaje rozměrů jsou vztaženy na základní systém řídicího systému (palce nebo metrické jednotky).		
Datový typ:		REAL[3]		

2	<Comp>:	Složky nástroje			
		Datový typ:		STRING	
		Řetězec znaků skládající se ze dvou dílčích řetězců, které jsou od sebe odděleny dvojtečkou.			
		Obecný formát: "<SubStr_1> [: <SubStr_2>]"			
		<SubStr_1>:	První dílčí řetězec musí být vždy k dispozici a může se skládat z jednoho nebo ze dvou znaků. První nebo jediný znak přitom reprezentuje 1. složku (Val ₁) a druhý znak 2. složku (Val ₂), které jsou zpracovávány v závislosti na následujících parametrech <CorComp> a <CorMode>.		
			Znaky:	C	Rozměr adaptéru nebo základní rozměr (použije se ta z obou alternativně existujících složek, která je pro používaný nástroj aktivní)
				E	Korekční parametry pro seřizování
				G	Geometrie
				S	Součtové korekce
		W		Opotřebení	
<SubStr_2>:	Druhý dílčí řetězec je nepovinný. Může se skládat z (jediného) písmena „W“ nebo „T“.				
	Znaky:	W	Pokud je druhý dílčí řetězec prázdný nebo pokud obsahuje písmeno „W“, jsou hodnoty korekcí započítávány tak, jako kdyby byly změřeny v souřadném systému obrobku (WKS).		
		T	Pokud obsahuje druhý dílčí řetězec písmeno „T“, jsou hodnoty korekcí započítávány tak, jako kdyby byly změřeny v souřadném systému nástroje (Tool Coordinate System, TCS).		
Způsob zápisu (malá nebo velká písmena) je libovolný. Může být vloženo libovolně mnoho prázdných znaků nebo znaků tabulátoru (netisknutelné znaky).					

3	<CorComp>:	Specifikuje složky datového bloku nástroje, které mají být popsány (nepovinné).		
		Datový typ:	INT	
		Hodnota:	0	Hodnota korekce <CorVal>[0] se vztahuje na geometrickou osu v souřadném systému obrobku nebo v souřadném systému nástroje předávanou v parametru <GeoAx> (bližší informace viz popis parametru <Comp>). To znamená, že hodnota korekce musí být ve stanovených složkách nástroje započítána tak, že jako výsledek vznikne změna celkové délky nástroje v definovaném směru osy o předdefinovanou hodnotu, přičemž jsou zohledněny všechny parametry, které mohou výpočet délky nástroje ovlivňovat. Tato změna má být dosažena prostřednictvím korekce složky zadané v parametru <Comp> a symbolického matematického předpisu zadaného v parametru <CorMode> (viz následující parametr). Výsledná korekce se proto může uplatňovat ve složkách všech tří os.
		1	Jako „0“, ale vektorově. Obsah vektoru <CorVal> se vztahuje na abscisu, ordinátu a aplikátu v souřadném systému obrobku nebo v souřadném systému nástroje předávanou v parametru <GeoAx> (bližší informace viz popis parametru <Comp>). Následující parametr <GeoAx> se nevyhodnocuje.	
		2	Vektorovou korekci, tzn. L1, L2 a L3, je možno upravovat simultánně. Na rozdíl od variant „0“ a „1“ se ale hodnoty korekcí obsažené v parametru <CorVal> vztahují na souřadnice složky Val ₁ (viz následující parametr <CorMode>) nástroje. Šikmá poloha nástroje vůči souřadnému systému obrobku, která se případně může vyskytnout, nemá na korekce žádný vliv.	
		3 - 5	Korekce délky nástroje L1 až L3 (\$TC_DP3 až \$TC_DP5), příp. odpovídající hodnoty u opotřebení, seřizovacích korekcí nebo součtových korekcí. Hodnota korekce je obsažena v parametru <CorVal>[0]. Je změřena v souřadnicích složky Val ₁ (viz následující parametr <CorMode>) nástroje. Šikmá poloha nástroje vůči souřadnému systému obrobku, která se případně může vyskytnout, nemá na korekce žádný vliv.	
		6	Korekce rádiusu nástroje (\$TC_DP6), příp. odpovídající hodnoty u opotřebení, seřizovacích korekcí nebo součtových korekcí. Bity 10 a 11 (údaje průměru, příp. opotřebení ve směru průměru, jsou vyhodnocovány volitelně jako údaje rádiusu nebo průměru) ve strojním parametru MD20360 \$MC_TOOL_PARAMETER_DEF_MASK jsou zohledňovány.	
		7 – 11	Korekce \$TC_DP7 až \$TC_DP11, příp. odpovídající hodnoty u opotřebení, seřizovacích korekcí nebo součtových korekcí. S těmito parametry se zachází stejně jako s rádiusem nástroje.	
		Jestliže tento parametr není uveden, je jeho hodnota „0“.		

4

<CorMode>:

Specifikuje druh operace zápisu, kterou je potřeba uskutečnit (nepovinné)		
Datový typ:		INT
Hodnota:	0	$Val_{1nová} = <CorVal>$
	1	$Val_{1nová} = Val_{1stará} + <CorVal>$
	2	$Val_{1nová} = <CorVal>$ $Val_{2nová} = 0$
	3	$Val_{1nová} = Val_{1stará} + Val_{2stará} + <CorVal>$ $Val_{2nová} = 0$

Způsobu zápisu $Val_{1stará} + Val_{2stará}$ je potřeba rozumět symbolicky. Pokud jsou obě složky (kvůli stavu parametru <_Stat>) vyhodnocovány odlišně, tzn. mezi oběma složkami se uplatňuje otočení, je hodnota parametru $Val_{2stará}$ před přičtením transformována tak, aby výsledná délka nástroje po vymazání hodnoty $Val_{2nová}$ a po přičtení parametru <CorVal> zůstala nezměněna.

<CorVal> se vždy vztahuje na parametr Val_1 . <CorVal> je hodnota, která se v závislosti na druhé části parametru <Comp> měří buď v souřadném systému obrobku (WCS), nebo v souřadném systému nástroje (TCS). Z tohoto důvodu je na rozdíl od složek nástroje, ve kterých se má započítávat, v případě potřeby již podrobena transformaci. Není tedy možné počítat bezprostředně s uloženými hodnotami, nýbrž před přičítáním k hodnotám Val_1 , příp. Val_2 musí být provedena zpětná transformace. To může mít za následek, že se korekce uplatňuje v jiné ose, než jaká je definována prostřednictvím parametru <CorComp>, příp. že se korekce uplatňuje ve více osách.

V případě, že se <CorComp> = 0, tzn. jestliže parametr <CorVal> neobsahuje vektor, ale jednotlivou hodnotu, jsou popisované operace uskutečňovány v souřadnicích, ve kterých byla hodnota <CorVal> změřena (WCS/TCS). To platí zejména také pro dosazení nuly pro parametr $Val_{2nový}$ ve variantách 2 a 3. Tento výsledek se potom zpětně transformuje do souřadnic nástroje. To může mít za následek, že žádná z vynulovaných souřadnic (L1, L2, L3) nebude nulová, příp. že hodnoty souřadnic, které byly nulové, už nulové nejsou. Jestliže jsou ale odpovídající operace uskutečněny postupně pro všechny tři geometrické osy, je zaručeno, že také všechny tři hodnoty souřadnic mazaných složek jsou nulové. Jestliže nástroj není oproti souřadnému systému obrobku otočený, nebo pokud je otočený, ale tak, že všechny složky nástroje zůstaly rovnoběžně se souřadnými osami (záměna os), je zaručeno, že se také změní jen jedna souřadnice nástroje.

Postupné uskutečnění stejné operace (<CorMode>) s parametrem <CorComp> = 0 pro všechny tři osy souřadného systému v libovolné posloupnosti má identický výsledek jako jednorázové provedení stejné operace s nastavením <CorComp>=1.

Pro hodnoty parametru „0“ a „1“ musí parametr <Comp> obsahovat jeden a pro hodnoty parametru „2“ a „3“ dva znaky.

Příklad:

<Comp> obsahuje řetězec „ES“, <CorMode> hodnotu „2“.

⇒ Seřizovací korekce_{nová} = <CorVal>, součtová korekce_{nová} = 0

Jestliže parametr <CorMode> není uveden, je jeho hodnota „0“.

5	<GeoAx>:	Udává index geometrické osy, ve které byla změřena hodnota korekce <Cor-Val>[0] (nepovinné).	
		Datový typ:	INT
		Rozsah hodnot:	0 ... 2
		Indexy 0 až 2 se vztahují na abscisu, ordinátu a aplikátu v platné rovině (G17/G18/G19) momentálního nástrojového prostředí. Obsah tohoto parametru se vyhodnocuje jen tehdy, pokud má parametr <Cor-Comp> hodnotu „0“.	
6	<Stat>:	Název datového bloku pro popis nástrojového prostředí (volitelné)	
		Datový typ:	STRING
		Jestliže se v tomto parametru nachází nulový řetězec („“) nebo pokud není udán, použije se aktuální stav. Pokud není specifikován žádný nástroj, použije se ten aktuální.	
7	<T>:	Interní T-číslo nástroje (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, použije se nástroj určený v parametru <Stat>.	
		Je-li hodnota tohoto parametru „-1“, použije se T-číslo aktivního nástroje. Je rovněž přípustné číslo aktivního nástroje explicitně zadat. Upozornění: Pokud parametr <Stat> není uveden, jako nástrojové prostředí se použije momentální stav. Protože zadání <T> = 0 znamená odkaz na T-číslo uložené v nástrojovém prostředí, použije se v tomto případě aktivní nástroj, což znamená, že zadání <T> = 0 a <T> = -1 mají za těchto speciálních okolností stejný význam.	
8	<D>:	Číslo břitu nástroje (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, řídí se použité D-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj.	
9	<TL>:	Číslo korekce závislé na místě (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden, řídí se použité DL-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj. Jestliže parametry T, D a DL specifikují nástroj bez korekcí závislých na místě, nesmí být v parametru <Comp> specifikovány žádné součtové nebo seřizovací korekce (chybový kód v parametru <Status>).	

Poznámka

Negace všechny možné kombinace trojice parametrů <Comp>, <CorComp> a <CorMode> mají smysl. Například matematický předpis 3 v parametru <CorComp> vyžaduje v parametru <Comp> zadání dvou znaků. Jestliže je zadána nepřipustná kombinace parametrů, v parametru <Status> se objeví příslušný chybový kód.

Příklady

Příklad 1

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; frézovací nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP12[1,1]=1.0	; opotřebení L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"G",0,0,2)	
N80 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X0.000 Y0.000 Z1.333
N90 M30	

<CorComp> má hodnotu „0“, proto je hodnota souřadnice geometrické složky, která se uplatňuje ve směru Z, nahrazena hodnotou korekce 0,333.

Výsledná celková délka nástroje tedy je: $L1 = 0,333 + 1,000 = 1,333$

Příklad 2

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; frézovací nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP12[1,1]=1.0	; opotřebení L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"W",0,1,2)	
N80 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> má hodnotu „1“, proto je hodnota korekce 0,333, která se uplatňuje ve směru Z, přičtena k hodnotě opotřebení 1,0.

Výsledná celková délka nástroje tedy je: $L1 = 10,0 + 1,333 = 11,333$

Příklad 3

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; frézovací nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP12[1,1]=1.0	; opotřebení L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,2,2)	
N80 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> má hodnotu „2“, z tohoto důvodu se hodnota korekce, která se uplatňuje ve směru osy Z, přenáší do složky geometrie (stará hodnota se přepisuje) a hodnota opotřebení se vymaže.

Výsledná celková délka nástroje tedy je: $L1 = 0,333 + 0,0 = 0,333$

Příklad 4

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; frézovací nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP12[1,1]=1.0	; opotřebení L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,3,2)	
N80 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> má hodnotu „3“, proto se hodnota opotřebení a hodnota korekce přičítají ke složkám geometrie a složky opotřebení se vymažou.

Výsledná celková délka nástroje tedy je: $L1 = 11,333 + 0,0 = 11,333$.

Příklad 5

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; frézovací nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP12[1,1]=1.0	; opotřebení L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,3,0)	
N80 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> má hodnotu „3“ jako v předcházejícím příkladu, korekce se ale nyní uplatňuje na geometrickou osu s indexem „0“ (osa X), které je kvůli G17 u frézovacího nástroje přiřazena složka nástroje L3. Vyvolání příkazu SETTCOR z tohoto důvodu parametry nástroje \$TC_DP3 a \$TC_DP12 neovlivňuje. Hodnota korekce se místo toho ukládá do parametru \$TC_DP5.

Příklad 6

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; soustružnický nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP4[1,1]=15.0	; geometrie L2
N50 \$TC_DP12[1,1]=10.0	; opotřebení L1
N60 \$TC_DP13[1,1]=0.0	; opotřebení L2

Programový kód	Komentář
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",0,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X24.330 Y0.000 Z17.500
N120 M30	

Nástrojem je soustružnický nůž. V bloku N80 je aktivováno otočení framu, takže základní souřadný systém (BCS) je vůči souřadnému systému obrobku (WCS) pootočený. Hodnota korekce (N70) se ve WCS uplatňuje na geometrickou osu s indexem „1“, tzn., protože je aktivní G18, na osu X. Protože se parametr <CorMode> = 3, platí, že opotřebení nástroje ve směru osy X ve WCS musí být po zpracování řádku N100 rovno nule.

Obsah příslušných parametrů nástroje na konci programu proto je:

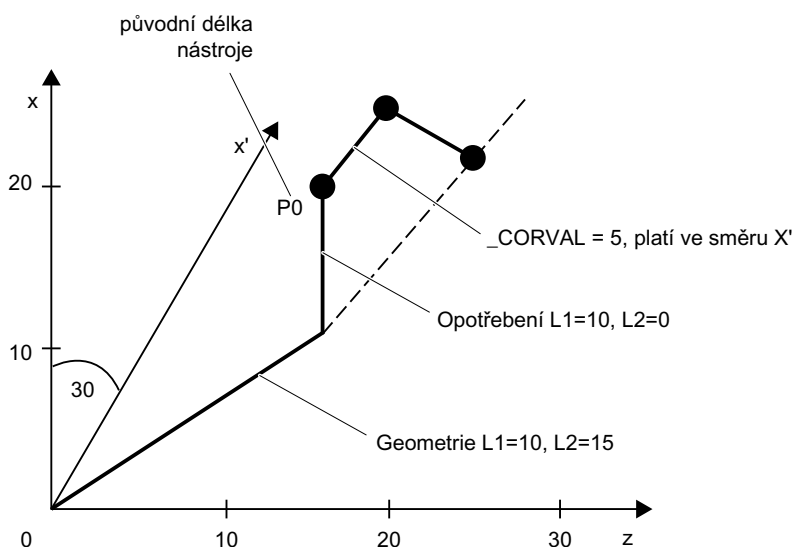
\$TC_DP3[1,1] : 21.830 ; geometrie L1

\$TC_DP4[1,1] : 21.830 ; geometrie L2

\$TC_DP12[1,1] : 2.500 ; opotřebení L1

\$TC_DP13[1,1] : -4.330 ; opotřebení L2

Tyto geometrické poměry zachycuje následující obrázek. Celkové opotřebení včetně hodnoty _CORVAL se promítá ve směru osy X' ve WCS. Z toho vyplývá bod P2. Souřadnice tohoto bodu (změřené v souřadnicích X-Y) jsou uloženy v geometrických složkách nástroje. V opotřebení zbývá rozdílový vektor $P_2 - P_1$. V důsledku toho již nemá opotřebení žádnou složku ve směru _CORVAL.



Pokud by program v příkladu pokračoval po bloku N110 následujícími příkazy, přeneslo by se zbývající opotřebení úplně do geometrie, protože korekce se nyní uplatňuje ve směru osy Z' (parametr <GeoAx> = 0):

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
```

N140 T1 D1 X0 Y0 Z0 ; ==> poloha v MCS X24.330 Y0.000 Z17.500

Protože nová hodnota korekce je „0“, nesmí se celková délka nástroje a tím pádem také pozice, na kterou se najíždí v bloku N140, změnit. Pokud by se hodnota `_CORVAL` v bloku N120 nerovnila „0“, vznikla by v bloku N140 nová celková délka nástroje a proto také změněná poloha, složka opotřebení délky nástroje by však byla v každém případě nulová, tzn. celková délka nástroje je potom v každém případě obsažena ve složce geometrie nástroje.

Stejný výsledek jako při dvojnásobném vyvolání funkce `SETTCOR` s nastavením parametru `<CorComp> = 0` se dosáhne také prostřednictvím jediného vyvolání s nastavením `<CorComp> = 1` (vektorová korekce):

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; soustružnický nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP4[1,1]=15.0	; geometrie L2
N50 \$TC_DP12[1,1]=10.0	; opotřebení L1
N60 \$TC_DP13[1,1]=0.0	; opotřebení L2
N70 _CORVAL[0]=0.0	
N71 _CORVAL[1]=5.0	
N72 _CORVAL[2]=0.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",1,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X24.330 Y0.000 Z17.500
N120 M30	

V tomto případě jsou všechny složky opotřebení nástroje okamžitě po prvním vyvolání funkce `SETTCOR` v bloku N100 rovny nule.

Příklad 7

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; soustružnický nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP4[1,1]=15.0	; geometrie L2
N50 \$TC_DP12[1,1]=10.0	; opotřebení L1
N60 \$TC_DP13[1,1]=0.0	; opotřebení L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",3,3)	
N110 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X25.000 Y0.000 Z15.000
N120 M30	

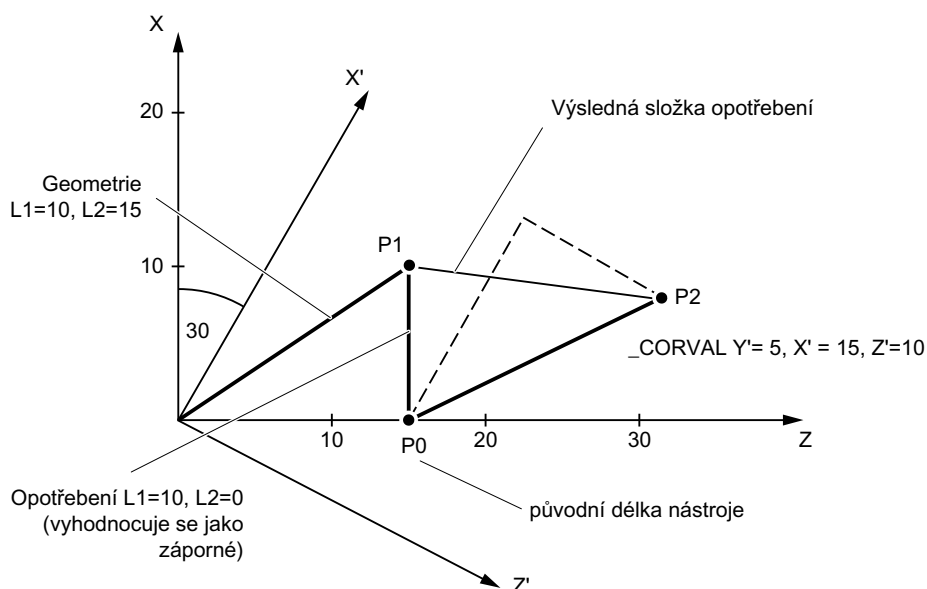
Oproti příkladu 6 je v tomto případě parametr `<CorComp> = 3`, proto může údaj parametru `<GeoAx>` odpadnout. Hodnota obsažená v parametru `_CORVAL[0]` se nyní uplatňuje bezprostředně na složku nástroje L1, otočení v bloku N80 nemá na výsledek žádný vliv, složky

opotřebení v \$TC_DP12 se spolu s hodnotou v parametru _CORVAL[0] přenáší do složky geometrie, takže kvůli \$TC_DP13 se celková délka nástroje již po prvním vyvolání funkce SETTCOR v bloku N100 nachází ve složce geometrie nástroje.

Příklad 8

Programový kód	Komentář
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; soustružnický nástroj
N30 \$TC_DP3[1,1]=10.0	; geometrie L1
N40 \$TC_DP4[1,1]=15.0	; geometrie L2
N50 \$TC_DP5[1,1]=20.0	; geometrie L3
N60 \$TC_DP12[1,1]=10.0	; opotřebení L1
N70 \$TC_DP13[1,1]=0.0	; opotřebení L2
N80 \$TC_DP14[1,1]=0.0	; opotřebení L3
N90 \$SC_WEAR_SIGN=TRUE	
N100 _CORVAL[0]=10.0	
N110 _CORVAL[1]=15.0	
N120 _CORVAL[2]=5.0	
N130 ROT Y-30	
N140 T1 D1 G18 G0	
N150 R1=SETTCOR(_CORVAL,"W",1,1)	
N160 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X7.990 Y25.000 Z31.160
N170 M30	

V bloku N90 je přiřazována hodnota nastavovanému parametru SD42930 \$SC_WEAR_SIGN, tzn. opotřebení musí být vyhodnocováno se záporným znaménkem. Korekce je vektorová (<CorComp> = 1) a vektor korekce musí být přičten k opotřebení (<CorMode> = 1). Tyto geometrické poměry v rovině Z-X ukazuje následující obrázek:



Kvůli nastavení <CorMode> = 1 zůstává složka geometrie nástroje nezměněna. Vektor korekce definovaný ve WCS (otočení okolo osy Y) musí být převzat do složky opotřebení tak, aby celková délka nástroje v obrázku 3 odpovídala bodu P₂. Výsledná složka opotřebení nástroje je proto dána vzdáleností bodů P₁ a P₂.

Protože je ale hodnota opotřebení kvůli nastavovanému parametru SD42930

\$SSC_WEAR_SIGN vyhodnocována jako záporná, musí být takto zjištěná korekce přenesena do paměti korekčních parametrů se záporným znaménkem. Obsah příslušných parametrů nástroje na konci programu proto je:

\$TC_DP3[1,1] : 10.000 ; geometrie L1 (nezměněna)

\$TC_DP4[1,1] : 15.000 ; geometrie L2 (nezměněna)

\$TC_DP5[1,1] : 10.000 ; geometrie L3 (nezměněna)

\$TC_DP12[1,1] : 2.010 ; opotřebení L1 ($= 10 - 15 * \cos(30) + 10 * \sin(30)$)

\$TC_DP13[1,1] : -16.160 ; opotřebení L2 ($= -15 * \sin(30) - 10 * \cos(30)$)

\$TC_DP14[1,1] : -5.000 ; opotřebení L3

Jako složka L3 ve směru osy Y se dá vliv nastavovaného parametru SD42930

\$SSC_WEAR_SIGN bez dalších komplikací rozpoznat prostřednictvím otočení framu.

Další informace

Soustružnické / brusné nástroje: Výpočet délky nástroje v závislosti na parametru MD20360 \$MC_TOOL_PARAMETER_DEF_MASK

To, jak je u soustružnických / brusných nástrojů zapotřebí vyhodnocovat opotřebení, příp. délku nástroje ve směru eventuálně existující osy, ve které je průměr nástroje, je definováno ve strojním parametru:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <hodnota>

<Bit>	<hodnota>	Význam
0	0	U soustružnických a/ brusných nástrojů se parametr opotřebení v příčné ose započítává jako hodnota rádiusu :
	1	U soustružnických / brusných nástrojů se parametr opotřebení v příčné ose započítává jako hodnota průměru :
1	0	U soustružnických / brusných nástrojů se délková složka rozměrů nástroje v příčné ose započítává jako hodnota rádiusu :
	1	U soustružnických / brusných nástrojů se délková složka rozměrů nástroje v příčné ose započítává jako hodnota průměru :

Jestliže jsou příslušné bity nastaveny, vyhodnocuje se odpovídající údaj s faktorem 0,5.

Korekce je prostřednictvím funkce SETTCOR realizována tak, že se celková efektivní změna délky nástroje rovné přenášené hodnotě v proměnné <CorVal>. Pokud je délka na základě strojního parametru MD20360 \$MC_TOOL_PARAMETER_DEF_MASK při výpočtu vyhodnocována s faktorem 0,5, musí se proto korekce této složky uskutečňovat s dvojnásobkem přenášené hodnoty.

Příklad

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK = 2 (délka nástroje musí být v ose, ve které se měří průměr, vyhodnocována s faktorem 0,5)

Osa X je osou průměru.

Programový kód	Komentář
N10 DEF REAL _LEN[11]	
N20 DEF REAL _CORVAL[3]	
N30 \$TC_DP1[1,1]=500	; Typ nástroje
N40 \$TC_DP2[1,1]=2	; Poloha břitu
N50 \$TC_DP3[1,1]=3.	; Geometrie - délka 1
N60 \$TC_DP4[1,1]=4.	; Geometrie - délka 2
N70 \$TC_DP5[1,1]=5.	; Geometrie - délka 3
N80 _CORVAL[0]=1.	
N90 _CORVAL[1]=1.	
N100 _CORVAL[2]=1.	
N110 T1 D1 G18 G0 X0 Y0 Z0	; ==> poloha v MCS X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)	
N130 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X2.5 Y6 Z5
N140 R3=\$TC_DP3[1,1]	; = 5. = (3.000 + 2.*1.000)
N150 R4=\$TC_DP4[1,1]	; = 5. = (4.000 + 1.000)
N160 R5=\$TC_DP5[1,1]	; = 6. = (5.000 + 1.000)
N170 M30	

Korekce délky nástroje by v každé ose měla být 1 mm (N80 až N100). V délkových parametrech L2 a L3 se proto k původní délce přičte vždy 1 mm. K původní délce nástroje v parametru L1 se oproti tomu přičítá dvojnásobek hodnoty korekce (2 mm), aby se celková délka nástroje prodloužila o 1 mm, jak je požadováno. Při porovnání pozic, na které se najíždí v blocích N110 a N130 je možno zjistit, že se všechny pozice os změnily o 1 mm.

11.13 Načítání přiřazení délek nástroje L1, L2, L3 osám souřadného systému (LENTOAX)

Funkce LENTOAX poskytuje informace o přiřazení délek nástroje L1, L2 a L3 **aktivního** nástroje abscise, ordinátě a aplikátě. Přiřazení abscisy, ordináty a aplikáty geometrickým osám je ovlivňováno framy a aktivní rovinou (G17 - G19).

Uvažovány jsou přitom pouze složky geometrie nástroje (\$TC_DP3[<t>,<d>] až \$TC_DP5[<t>,<d>]), tzn. přiřazení dalších složek osám, které se v případě potřeby od tohoto liší (např. opotřebení), nemá na výsledek žádný vliv.

Syntaxe

<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])

Princip

LENTOAX(...):		Předem definovaná funkce pro načítání přiřazení délek nástroje L1, L2 a L3 aktivního nástroje osám souřadného systému.	
		Samostatný blok:	ano
<Status>:		Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.	
		Datový typ:	INT
Hodnota:		0	Funkce je v pořádku Informace v parametru <AxInd> jsou pro popis postačující (všechny složky délky nástroje jsou rovnoběžné s geometrickými osami).
		1	Funkce je v pořádku, aby byl ale popis správný, musí být vyhodnocován obsah parametru <Matrix> (složky délky nástroje nejsou rovnoběžné s geometrickými osami).
		-1	Neplatný řetězec v parametru <Coord>.
		-2	Není aktivní žádný nástroj.
Parametry			
1	<AxInd>:	Pokud jsou složky délky nástroje rovnoběžné s geometrickými osami, jsou v poli <AxInd> funkcí poskytovány indexy os, které jsou přiřazeny délkovým složkám L1 až L3.	
		• <AxInd> [0]: Abscisa • <AxInd> [1]: Ordináta • <AxInd> [2]: Aplikáta	
		Datový typ:	INT[3]
Hodnota:		0	Žádné přiřazení není k dispozici (osa neexistuje)
		1 ... 3 nebo -1 ... -3	Číslo délky, které se uplatňuje v odpovídající souřadné ose. Pokud složka délky nástroje ukazuje v záporném směru příslušné souřadnice, je znaménko záporné.
		Jestliže ne všechny složky délky jsou rovnoběžné, příp. antiparalelní, s geometrickými osami, bude parametr <AxInd> poskytovat příslušný index osy, která obsahuje největší podíl složky délky nástroje. V tomto případě (jestliže funkce z nějakého jiného důvodu nehlásí chybu), je výsledná hodnota <Status> = 1. Zobrazení složek délky nástroje L1 až L3 do geometrických os 1 až 3 je potom úplně popsáno obsahem 2. parametru <Matrix>.	

11.13 Načítání přiřazení délek nástroje L1, L2, L3 osám souřadného systému (LENTOAX)

2	<Matrix>:	Matice, která provádí zobrazení vektoru délek nástroje (L1=1, L2=1, L3=1) do vektoru souřadných os (abscisa, ordináta, aplikáta), tzn. sloupcům jsou přiřazeny složky délky nástroje v posloupnosti L1, L2, L3 a řádkům osy v posloupnosti abscisa, ordináta a aplikáta.	
		Datový typ:	REAL
		V parametru Matrix jsou vždy v platnosti všechny prvky, a to i tehdy, když geometrická osa patří k určité ose souřadného systému vůbec neexistuje, tzn. když je odpovídající položka v parametru <AxInd> nulová.	
3	<Coord>:	Souřadný systém, pro který přiřazení platí (nepovinné)	
		Datový typ:	STRING
		Znaky:	MCS M
			Zobrazení délky nástroje v souřadném systému stroje
			BCS B
			Zobrazení délky nástroje v základním souřadném systému
			WCS W
			Zobrazení délky nástroje v souřadném systému obrobku (standardní nastavení)
		KCS K	Zobrazení délky nástroje v souřadném systému nástroje kinematické transformace
		TCS T	Zobrazení délky nástroje v souřadném systému nástroje
		Způsob zápisu (malá nebo velká písmena) je libovolný. Jestliže parametr <Coord> není uveden, použije se WCS (předdefinované nastavení).	

Poznámka

V TCS jsou všechny složky délky nástroje vždy paralelní nebo antiparalelní s osami.

Antiparalelní mohou být složky jen tehdy, jestliže je aktivní zrcadlové převrácení a je-li aktivován následující nastavovaný parametr:

SD42900 \$SC_MIRROR_TOOL_LENGTH (změna znaménka délky nástroj při zrcadlovém převrácení)

Příklad

Standardní případ frézovacího nástroje v případě G17.

L1 se uplatňuje v ose Z (aplikáta), L2 se uplatňuje v ose Y (ordináta), L3 se uplatňuje v ose X (abscisa).

Vyvolání funkce ve formě:

```
<Status>=LENTOAX (<AxInd>, <Matrix>, "WCS")
```

Výsledný parametr <AxInd> potom obsahuje hodnoty:

```
<AxInd>[0] = 3
```

```
<AxInd>[1] = 2
```

```
<AxInd>[2] = 1
```

logické zkráceně: (3, 2, 1)

Odpovídající matice v tomto případě je:

$$\langle \text{Matrix} \rangle = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Přepnutím z G17 do G18 nebo G19 se výsledek nezmění, protože přiřazení složek délky ke geometrickým osám se změní stejným způsobem jako přiřazení abscisy, ordináty a aplikáty.

Když je aktivní funkce G17, naprogramuje se nyní otočení framu okolo osy Z o 60°, např.:

ROT Z60

Směr aplikáty (směr osy Z) zůstává nezměněn, hlavní složka L2 nyní leží ve směru nové osy X, hlavní složka L1 se nachází v záporném směru osy Y. Výsledná hodnota (<Status>) je proto „1“, <AxInd> obsahuje hodnoty (2, -3, 1).

Odpovídající matice v tomto případě je:

$$\langle \text{Matrix} \rangle = \begin{pmatrix} 0 & \sin 60^\circ & \cos 60^\circ \\ 0 & \cos 60^\circ & -\sin 60^\circ \\ 1 & 0 & 0 \end{pmatrix}$$

Chování při pohybu po dráze

12.1 Tangenciální řízení

12.1.1 Definice vazby (TANG)

Prostřednictvím předem definované procedury TANG(...) je definována tangenciální vazba mezi kruhovou osou, která je vlečnou osou, a dvěma geometrickými osami, které jsou řídicími osami. Vlečná osa se přitom kontinuálně nastavuje vzhledem k tečně dráhy řídicích os.

Poznámka

Převodový faktor

Faktor vazby 1 nemusí být explicitně naprogramován.

Prostřednictvím faktoru vazby -1 se směr tangenciální osy obrátí.

Syntaxe

TANG(<vlečná osa>, <řídicí osa_1>, <řídicí osa_2>, <faktor vazby>, <souřadný systém>, <optimalizace>)

Význam

TANG(...):	Definice tangenciální vazby	
<vlečná osa>:	Název vlečné osy (kruhová osa)	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os
<řídicí osa_1> <řídicí osa_2>:	Názvy řídicích os (geometrické osy) ¹⁾	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy geometrických os v kanálu
<faktor vazby>:	Faktor n pro změnu úhlu vlečné osy při změně tečny dráhy řídicích os: Změna hodnoty úhlu _{vlečná osa} = Změna hodnoty úhlu _{tečna dráhy} * n	
	Datový typ:	REAL
	Předdefinovaná hodnota:	1,0

12.1 Tangenciální řízení

<Souřadný systém>:	Souřadný systém, který je v platnosti ²⁾		
	Datový typ:	CHAR	
	Hodnota:	"B":	Základní souřadný systém (předdefinovaná hodnota)
		"W":	Souřadný systém obrobku (není k dispozici)
<optimalizace>:	Druh optimalizace		
	Datový typ:	CHAR	
	Hodnota:	"S":	Standardní (předdefinovaná hodnota) Dynamika kruhové osy nemá žádný zpětný vliv na řídicí osy. Jestliže je dynamika kruhové osy vyšší, než je pro vlečení zapotřebí, je její pohyb dostatečně přesný. Pokud není dynamika kruhové osy dostatečně vysoká, aby stíhala náležitě reagovat na změnu tečny dráhy, odlišuje se nastavení kruhové osy podél nedefinovaných úseků zaoblení od požadovaného nasměrování.
		"P":	Dynamika kruhové osy je při plánování dráhy řídicích os zohledňována. Za tím účelem musí být při aktivování tangenciální vazby pomocí příkazu TANGON() zadány dva doplňkové parametry: • Dráha zaoblení • Tolerance úhlu Viz kapitola "Aktivování vazby (TANGON) (Strana 518)". Poznámka Ve spojení s kinematickými transformacemi doporučujeme používat druh optimalizace "P".

Poznámka
Předdefinované hodnoty nemusí být explicitně naprogramovány.
¹⁾ Poznámka Jako řídicí osy tangenciální vazby se musí používat geometrické osy, které objíždějí naprogramovanou dráhu v souřadném systému stroje (MCS) (vztaženo na základní nastavení stroje). Jestliže je např. na frézce s naklápěcí hlavičkou použit cyklus naklápění CYCLE800, uskutečňuje se v závislosti na konfiguraci cyklu interpolace v systému WCS, např. s geometrickými osami X a Y. Tangenciální vazba ale musí být definována s geometrickými osami, jež jsou pak řídicími osami, které objíždějí naprogramovanou dráhu v systému MCS. Pro tento účel je jako řídicí osy potřeba použít geometrické osy v nena-klopeném stavu stroje.
²⁾ Poznámka Základní souřadný systém (BCS) nesmí být vůči MCS otočený. Jestliže je systém BCS otočený, např. pomocí příkazu ROT nebo prostřednictvím cyklu naklápění CYCLE800, nepracuje tangenciální řízení správně.

12.1.2 Aktivování vytváření pomocných bloků (TLIFT)

Jestliže tangenciální změna vlečné osy na určitém místě naprogramované dráhy řídících os překročí mezní hodnotu nastavenou ve strojním parametru MD37400 \$MA_EPS_TLIFT_TANG_STEP, bude další plánování dráhy záviset na nastaveném chování v rozích. Pokud není použita předem definovaná procedura TLIFT(...), bude pohyb probíhat v závislosti na chování na rohových zaobleních, které bylo naprogramováno v souvislosti s funkcemi TANG(...) (Strana 515) a TANGON(...) (Strana 518).

Aktivování vytváření pomocných bloků

Když je v návaznosti na funkci TANG(...) naprogramován příkaz TLIFT(...), pak když je během předběžného zpracování rozpoznán roh, na tomto místě dráhy řídící systém automaticky vloží vygenerovaný pomocný blok.

Při zpracování programu se pak po dosažení tohoto pomocného bloku řídící osy pozastaví. V pomocném bloku se vlečná osa se svou maximální dynamikou otočí ve směru tečny ke dráze v následujícím bloku. Potom se řídící osy dále pohybují po své naprogramované dráze.

Deaktivování vytváření pomocných bloků

Abyste vytváření pomocných bloků deaktivovali, musí být prostřednictvím příkazu TANG(...) tangenciální vazba nově definována, ale tentokrát bez navazujícího aktivování vkládání pomocných bloků pomocí příkazu TLIFT(...).

Syntaxe

TLIFT(<vlečná osa>)

Význam

TLIFT(...):	Aktivování rozpoznávání rohů s výpočtem pomocných bloků	
<vlečná osa>:	Název vlečné osy (kruhová osa)	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os

Rychlost otáčení vlečné osy

Dráhová osa

Pokud se vlečná osa již před aktivováním tangenciální vazby pohybovala jako osa dráhová, probíhá otáčivý pohyb v pomocném bloku v režimu dráhové osy.

Prostřednictvím zadání vztažného rádiusu pomocí příkazu FGREF[<osa>]=0.001 bude otáčivý pohyb probíhat s maximální rychlostí osy nastavenou v parametru:

MD32000 \$MA_MAX_AX_VELO[<vlečná osa>]

Polohovací osa

Pokud se vlečná osa před aktivováním tangenciální vazby ještě nepohybovala jako osa dráhová, probíhá otáčení v pomocném bloku v režimu polohovací osy.

Pro otáčivý pohyb se přitom použije rychlost polohovací osy nastavená v parametru:

MD32060 \$MA_POS_AX_VELO[<vlečná osa>]

12.1.3 Aktivování vazby (TANGON)

Prostřednictvím předem definované procedury TANGON(...) se aktivuje tangenciální vazba, která byla předtím definována pomocí příkazu TANG(...) (Strana 515). Vlečná osa se potom při následujících pohybech řídicích os kontinuálně nastavuje vzhledem k tečně její dráhy.

Úhel vlečné osy

Úhel, který vlečná osa zaujímá vzhledem k tečně k dráze, závisí na přenosovém poměru zadaném předem v příkazu TANG(...), na úhlu offsetu stanoveném ve strojním parametru MD37402 \$MA_TANG_OFFSET a na přičítaném úhlu offsetu zadaném v příkazu TANGON(...).

Optimalizace "P"

Pokud byla při definici tangenciální vazby (TANG...) zadána jako optimalizační parametr hodnota „P“, musí být při aktivování vazby zadány parametry "Dráha zaoblení" a případně také parametr "Tolerance úhlu" (nepovinné).

Jestliže je jako tolerance úhlu zadána hodnota 0, uplatňuje se pouze parametr "Dráha zaoblení".

Pokud je jako úhlová tolerance zadána hodnota větší než 0, odpovídá používaná dráha zaoblení minimu z dráhy zaoblení nastavené v parametrech a z dráhy zaoblení vyplývající ze zadané úhlové tolerance.

Jestliže dynamika vlečné osy není postačující, takže podmínky definované v parametrech nelze splnit, rychlost pohybu řídicích os po dráze se odpovídajícím způsobem sníží.

Syntaxe

TANGON(<vlečná osa>, <offset úhlu>, <dráha zaoblení>, <tolerance úhlu>)

Význam

TANGON(...):	Aktivování tangenciální vazby	
<vlečná osa>:	Název vlečné osy (kruhová osa)	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os
<offset úhlu>:	Úhlový offset vlečné osy vzhledem k tečně dráhy Vztažným bodem je nula (počátek) kruhové osy.	
	Datový typ:	REAL
<dráha zaoblení>:	Maximální přípustná dráha zaoblení Pokud by byla dráha zaoblení v důsledku dynamických podmínek větší, rychlost pohybu řídicích os po dráze se sníží.	
	Datový typ:	REAL

<úhlová tolerance>:	Maximální přípustná tolerance vztahující se na zadaný úhel mezi nulovou polohou vlečné osy a tečnou ke dráze	
	Datový typ:	REAL

12.1.4 Deaktivování vazby (TANGOF)

Prostřednictvím předem definované procedury TANGOF(...) se deaktivuje tangenciální vazba, která byla předtím definována pomocí příkazu TANG(...) (Strana 515) a aktivována pomocí příkazu TANGON(...) (Strana 518). Vlečná osa se potom přestane nastavovat vzhledem k tečně dráhy řídící osy. Vazba vlečné osy vzhledem k řídícím osám zůstává ale i po svém deaktivování definována, což např. blokuje následující funkce:

- Změna roviny
- Přepínání geometrických os
- Definice nové tangenciální vazby pro danou vlečnou osu

K úplnému zrušení vazby mezi vlečnou osou a řídícími osami dojde teprve až po vymazání vazby pomocí příkazu TANGDEL(...) (Strana 519).

Programování

TANGOF(<vlečná osa>)

Význam

TANGOF(...):	Deaktivování tangenciální vazby	
<vlečná osa>:	Název vlečné osy (kruhová osa)	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os

12.1.5 Vymazání vazby (TANGDEL)

Když je tangenciální vazba definovaná pomocí příkazu TANG(...) (Strana 515) deaktivována pomocí příkazu TANGOF(...) (Strana 519), nepřestává existovat. Existující tangenciální vazba potom blokuje např. následující funkce:

- Změna roviny
- Přepínání geometrických os
- Definice nové tangenciální vazby pro danou vlečnou osu

Pomocí předem definované procedury TANGDEL(...) se existující tangenciální vazba po svém deaktivování pomocí příkazu TANGOF(...) vymaže.

Syntaxe

TANGDEL(<vlečná osa>)

Význam

TANGDEL(...):	Vymazání tangenciální vazby definované pomocí příkazu TANG()	
	Platnost:	Bloková
<vlečná osa>:	Název vlečné osy, jejíž tangenciální vazba má být vymazána	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os

Příklady

Změna řídící osy

Předtím, než je možno pro danou vlečnou osu definovat novou tangenciální vazbu s jinou řídící osou, musí být napřed již existující tangenciální vazba vymazána.

Programový kód	Komentář
N10 TANG (A, X, Y, 1)	; definice tangenciální vazby pro vlečnou osu A: A vzhledem k osám X a Y
N20 TANGON(A)	; aktivování tangenciální vazby pro vlečnou osu A
N30 X10 Y20	
...	
N80 TANGOF(A)	; deaktivování tangenciální vazby pro vlečnou osu A
N90 TANGDEL (A)	; vymazání tangenciální vazby pro vlečnou osu A
...	
N120 TANG (A, X, Z)	; definice nové tangenciální vazby pro vlečnou osu A
N130 TANGON(A)	; aktivování nové tangenciální vazby pro vlečnou osu A
...	

Přepínání geometrických os

Předtím, než je možno pro již existující vazbu uskutečnit přepnutí geometrických os, musí být tato vazba napřed vymazána.

Programový kód	Komentář
N10 GEOAX(2, Y1)	; 2. geometrická osa = strojní osa Y1
N20 TANG(A, X, Y)	; definice tangenciální vazby pro vlečnou osu A
N30 TANGON(A, 90)	; aktivování tangenciální vazby pro vlečnou osu A
N40 G2 F8000 X0 Y0 I0 J50	; pohybový blok
N50 TANGOF(A)	; deaktivování tangenciální vazby pro vlečnou osu A
N60 TANGDEL (A)	; vymazání tangenciální vazby pro vlečnou osu A
N70 GEOAX (2, Y2)	; 2. geometrická osa = strojní osa Y2
N80 TANG(A, X, Y)	; definice nové tangenciální vazby pro vlečnou osu A
N90 TANGON(A, 90)	; aktivování nové tangenciální vazby pro vlečnou osu A
...	

12.2 Průběh charakteristiky posuvu (FNORM, FLIN, FCUB, FPO)

Pro účely pružnějšího zadávání průběhu posuvu bylo jeho programování podle normy DIN 66025 rozšířeno o lineární a kubické průběhy.

Kubické průběhy mohou být naprogramovány buď přímo nebo jako interpolační spliny. Tak je možné naprogramovat - v závislosti na zakřivení opracovávaného obrobku - spojitě a hladké funkční průběhy rychlosti.

Tyto funkční průběhy rychlosti umožňují změny zrychlení bez trhavých pohybů a tím tedy i výrobu stejnoměrných povrchů obrobku.

Syntaxe

F... FNORM
F... FLIN
F... FCUB
F=FPO (... , ... , ...)

Význam

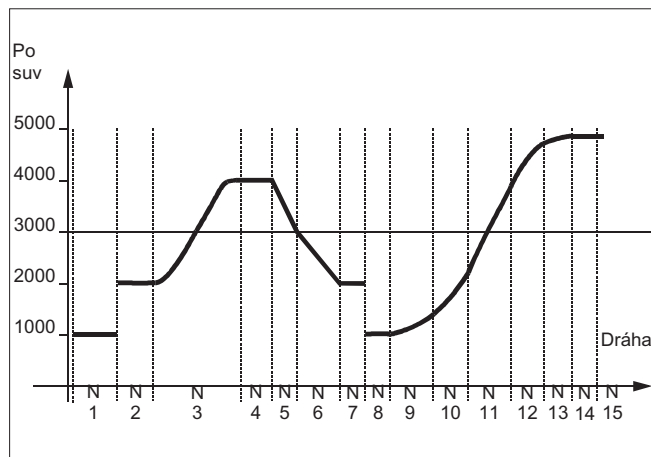
FNORM:	Základní nastavení. Hodnota posuvu se zadává prostřednictvím dráhového úseku daného bloku a potom platí jako modální hodnota.
FLIN:	Profil rychlosti pohybu po dráze je lineární : Hodnota posuvu se z hodnoty aktuální na začátku bloku do hodnoty na konci bloku po celém úseku dráhy lineárně mění a potom platí jako modální hodnota. Toto chování může být kombinováno s příkazy G93 a G94.
FCUB:	Profil rychlosti pohybu po dráze je kubický : Blokově naprogramované hodnoty F - vztažené na konec bloku - jsou spojovány spliny. Spline začíná a končí s tangenciálním napojením na předešlý, resp. následující údaj posuvu a uplatňuje se s příkazy G93 a G94. Pokud v nějakém bloku F-adresa chybí, použije se pro tento účel naposled naprogramované F-slovo.
F=FPO... :	Profil rychlosti pohybu po dráze je zadán pomocí polynomu : F-adresa popisuje průběh posuvu od aktuální hodnoty do hodnoty na konci bloku pomocí polynomu. Koncová hodnota pak platí jako modální.

Optimalizace posuvu v případě zakřivených úseků dráhy

Pro pohyby s posuvem daným polynomem $F=FPO$ a splinem $FCUB$ by vždy měla být nastavena konstantní řezná rychlost CFC . Tímto způsobem je možné vytvořit profil požadované hodnoty posuvu se spojitým zrychlením.

Příklad: Různé profily hodnoty posuvu

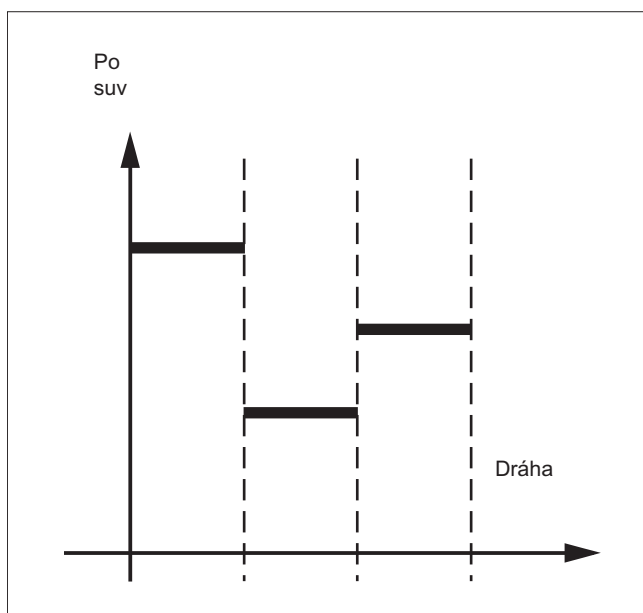
V tomto příkladu naleznete programové příkazy a grafické zobrazení různých profilů hodnoty posuvu.



Programový kód	Komentář
N1 F1000 FNORM G1 X8 G91 G64	; konstantní profil hodnoty posuvu, zadávání inkrementálních rozměrů
N2 F2000 X7	; skoková změna požadované hodnoty rychlosti
N3 F=FPO(4000, 6000, -4000)	; profil hodnoty posuvu zadáný pomocí polynomu s hodnotou posuvu 4000 na konci bloku
N4 X6	; posuv 4000 daný polynomem 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 hodnoty posuvu se skokovou změnou zrychlení
N9 F1400 FCUB X8	; všechny následující blokově naprogramované hodnoty F budou pospojovány spliny
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; deaktivování splinového profilu
N14 FNORM X5	
N15 X20	

Další informace**FNORM**

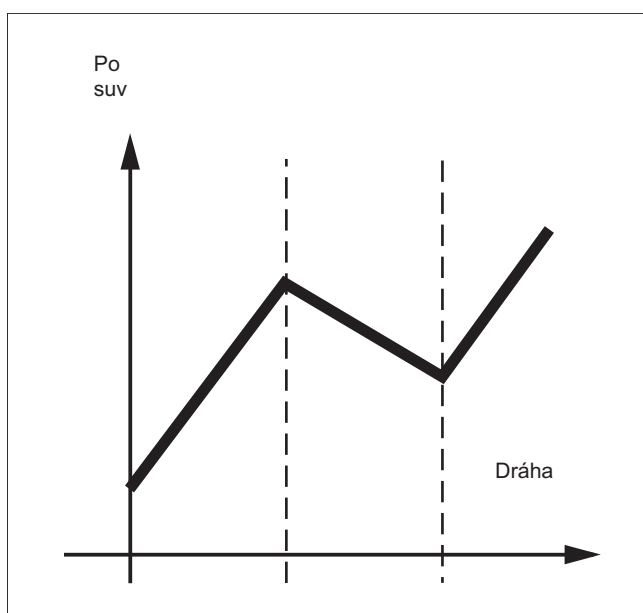
Podle normy DIN 66025 označuje adresa posuvu F posuv po dráze jako konstantní hodnotu. Více informací o této problematice naleznete v Programovací příručce, "Základy".

**FLIN**

Hodnota posuvu se bude lineárně měnit od momentální hodnoty posuvu až do naprogramovaného F-slova, přičemž této hodnoty bude dosaženo na konci bloku.

Příklad:

```
N30 F1400 FLIN X50
```

**FCUB**

Hodnota posuvu se bude měnit od momentální hodnoty posuvu až do naprogramovaného F-slova podle kubické funkce, přičemž této hodnoty bude dosaženo na konci bloku. Řídící systém pomocí splinových funkcí pospojuje všechny hodnoty posuvu naprogramované pomocí

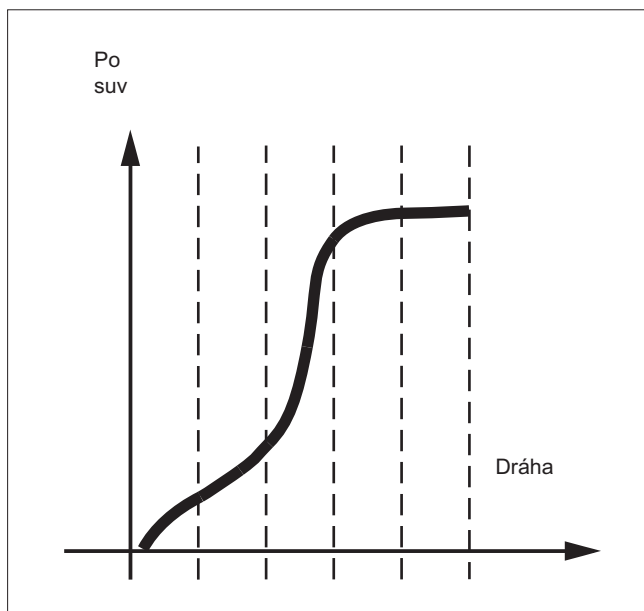
příkazu FCUB s blokovou platností. Hodnoty posuvu přitom slouží jako uzlové body pro výpočet splinové interpolace.

Příklad:

N50 F1400 FCUB X50

N60 F2000 X47

N70 F3800 X52



F=FPO(.....)

Průběh hodnoty posuvu je přímo naprogramován pomocí polynomu. Zadávání koeficientů polynomu se uskutečňuje podobně jako při polynomicke interpolaci.

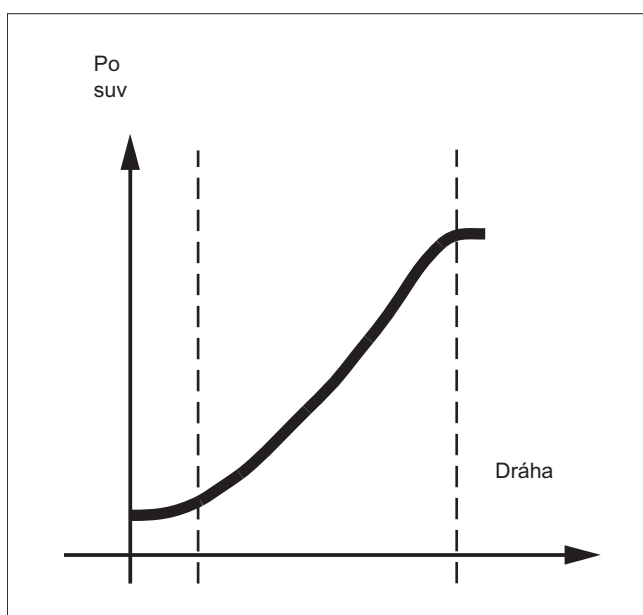
Příklad:

F=FPO(endfeed, quadf, cubf)

Parametry endfeed, quadf a cubf jsou dříve definované proměnné.

endfeed:	Posuv na konci bloku.
quadf:	Kvadratický koeficient polynomu
cubf:	Kubický koeficient polynomu

Když je aktivní příkaz FCUB, napojují se spliny na začátku bloku a na konci bloku tangenciálně na průběh zadáný pomocí funkce FPO.



Okrajové podmínky

- Nezávisle na naprogramovaném průběhu posuvu platí funkce pro programování chování při pohybu po dráze.
- Naprogramovaný průběh posuvu má v zásadě absolutní platnost - nezávisle na příkazech G90 nebo G91.
- Průběh hodnoty posuvu **FLIN** a **FCUB** se uplatňuje při příkazech G93 a G94, **nikoli** u příkazů G95, G96/G961 a G97/G971.
- Když je aktivní kompresor **COMPON**, platí při spojování většího počtu bloků do jednoho splinového úseku následující:

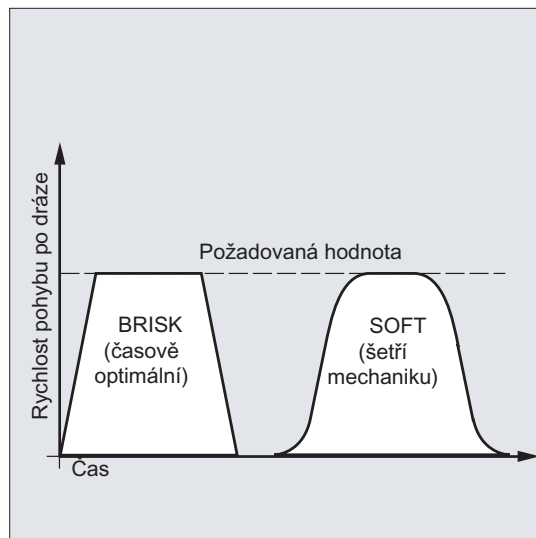
FNORM:	Pro splinový segment platí F-slovo z posledního bloku, který do tohoto segmentu patří.
FLIN:	Pro splinový segment platí F-slovo z posledního bloku, který do tohoto segmentu patří. Naprogramovaná hodnota F platí na konci segmentu a bude jí dosaženo pomocí lineární funkce.
FCUB:	Vytvořený spline posuvu se od naprogramované hodnoty v koncovém bodu odchyluje maximálně o hodnotu definovanou ve strojním parametru MD20172 \$MC_COMPRESS_VELO_TOL.
F=FPO (... , ... , ...) :	Tyto bloky nejsou nijak komprimovány.

12.3 Chování zrychlení:

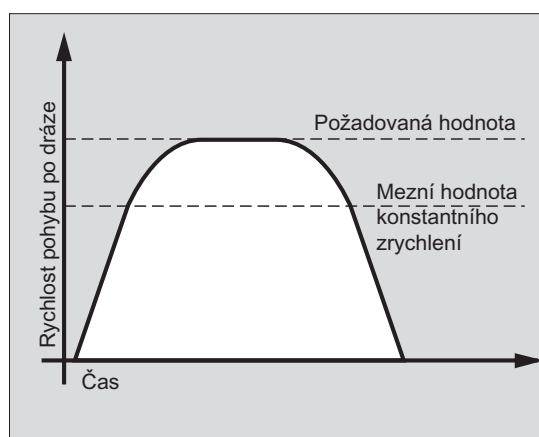
12.3.1 Způsoby chování při změnách zrychlení (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Pro programování způsobu chování při změnách zrychlení jsou k dispozici následující příkazy výrobního programu:

- "BRISK, BRISKA"
Jednotlivé osy, příp. dráhové osy se pohybují s maximálním zrychlením až do dosažení naprogramované rychlosti posuvu (**zrychlení bez omezení ryvu**).
- "SOFT, SOFTA"
Jednotlivé osy, příp. dráhové osy se pohybují se spojitým zrychlením až do dosažení naprogramované rychlosti posuvu (**zrychlení s omezením ryvu**).
- "DRIVE, DRIVEA"
Jednotlivé osy, příp. dráhové osy se pohybují s maximálním zrychlením až do dosažení mezní hodnoty rychlosti nastavené v konfiguraci (nastavení pomocí MD!). Potom dochází ke snížení zrychlení (nastavení pomocí MD!) až do dosažení naprogramované rychlosti posuvu.



Obrázek 12-1 Průběh rychlosti pohybu po dráze v případě příkazů BRISK a SOFT

Obrázek 12-2 Průběh rychlosti pohybu po dráze v případě příkazu `DRIVE`

Syntaxe

```
BRISK
BRISKA (<osa1>, <osa2>, ...)
SOFT
SOFTA (<osa1>, <osa2>, ...)
DRIVE
DRIVEA (<osa1>, <osa2>, ...)
```

Význam

BRISK:	Příkaz pro aktivování "Zrychlení bez omezení ryvu" pro dráhové osy.
BRISKA:	Příkaz pro aktivování "Zrychlení bez omezení ryvu" pro pohyby jednotlivých os (JOG, JOG/INC, polohovací osy, kyvné osy atd.).
SOFT:	Příkaz pro aktivování "Zrychlení s omezením ryvu" pro dráhové osy.
SOFTA:	Příkaz pro aktivování "Zrychlení s omezením ryvu" pro pohyby jednotlivých os (JOG, JOG/INC, polohovací osy, kyvné osy atd.).
DRIVE:	Příkaz pro aktivování omezené hodnoty zrychlení nad mezní hodnotou rychlosti nastavenou v konfiguraci (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT) pro dráhové osy.
DRIVEA:	Příkaz pro aktivování omezené hodnoty zrychlení nad mezní hodnotou rychlosti nastavenou v konfiguraci (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT) pro pohyby jednotlivých os (JOG, JOG/INC, polohovací osy, kyvné osy atd.).
(<osa1>, <osa2>, ...):	Jednotlivé osy, pro které má vyvolávaný způsob chování zrychlení platit.

Okrajové podmínky

Změna způsobu chování při změnách zrychlení v průběhu obrábění

Jestliže se ve výrobním programu v průběhu opracovávání mění způsob chování při změnách zrychlení (BRISK <--> SOFT), potom se i při řízení pohybu po dráze uskutečňuje na přechodech mezi bloky přesné najetí na konci bloku.

Příklady

Příklad 1: SOFT a BRISKA

Programový kód
N10 G1 X... Y... F900 SOFT
N20 BRISKA (AX5, AX6)
...

Příklad 2: DRIVE a DRIVEA

Programový kód
N05 DRIVE
N10 G1 X... Y... F1000
N20 DRIVEA (AX4, AX6)
...

Literatura

Příručka k funkcím, Základní funkce, Zrychlení (B2)

12.3.2 Ovlivňování chování zrychlení u vlečných os (VELOLIMA, ACCLIMA, JERKLIMA)

V případě vazby mezi osami (tangenciální vlečení, synchronizované osy, spojení s řídicí hodnotou a elektronická převodovka; viz "Spojení os vazbou (Strana 563)") se pohyb vlečných os/vřeten uskutečňuje v závislosti na jedné nebo na několika řídicích osách/vřetenech.

Omezení dynamiky pohybů vlečných os/vřeten je možno ovlivňovat buď z výrobního programu nebo ze synchronní akce pomocí funkcí VELOLIMA, ACCLIMA a JERKLIMA, a to i když je vazba mezi osami už aktivní.

Poznámka

Funkce JERLIMA není k dispozici pro všechny druhy vazeb mezi osami.

Literatura:

- Příručka k funkcím, Speciální funkce; Vazby mezi osami (M3)
- Příručka k funkcím, Rozšiřovací funkce; Synchronní vřetena (S3)

Poznámka**Použitelnost u systému SINUMERIK 828D**

Funkce VELOLIMA, ACCLIMA a JERKLIMA se mohou u systému SINUMERIK 828D používat jedině ve spojení s funkcí "Vlečení"!

Syntaxe

VELOLIMA (<osa>)=<hodnota>

ACCLIMA (<osa>)=<hodnota>

JERKLIMA (<osa>)=<hodnota>

Význam

VELOLIMA:	Příkaz pro korekci v parametrech nastavené maximální rychlosti .
ACCLIMA:	Příkaz pro korekci v parametrech nastaveného maximálního zrychlení .
JERKLIMA:	Příkaz pro korekci v parametrech nastaveného maximálního ryvu .
<osa>:	Vlečná osa, jejíž omezení dynamiky pohybů má být korigováno.
<hodnota>:	Procentuální hodnota korekce

Příklady**Příklad 1: Korekce omezení dynamiky pohybů pro jednu vlečnou osu (AX4)**

Programový kód	Komentář
...	
VELOLIMA[AX4]=75	; omezovací korekce na 75% maximální rychlosti osy uložené ve strojním parametru
ACCLIMA[AX4]=50	; omezovací korekce na 50% maximálního zrychlení osy uložené ve strojním parametru
JERKLIMA[AX4]=50	; omezovací korekce při pohybu po dráze na 50% maximálního ryvu osy uloženého ve strojním parametru
...	

Příklad 2: Elektronická převodovka

Osa 4 je prostřednictvím vazby "Elektronická převodovka" spojena s osou X. Chování zrychlení vlečné osy je omezeno na 70 % maximálního zrychlení. Maximální přípustná rychlost bude omezena na 50 % maximální rychlosti. Po úspěšném aktivování vazby bude maximální rychlost znovu nastavena na 100 %.

Programový kód	Komentář
...	
N120 ACCLIMA[AX4]=70	; snížené maximální zrychlení
N130 VELOLIMA[AX4]=50	; snížená maximální rychlost
...	
N150 EGON(AX4,"FINE",X,1,2)	; aktivování vazby typu "Elektronická převodovka"

12.3 Chování zrychlení:

Programový kód	Komentář
...	
N200 VELOLIMA[AX4]=100	; plná maximální rychlost
...	

Příklad 3: Spojení pomocí řídící hodnoty s ovlivňováním pomocí statické synchronní akce

Osa 4 je spojena pomocí vazby s řídící hodnotou s osou X. Chování zrychlení je omezeno prostřednictvím statické synchronní akce 2 od pozice 100 na 80%.

Programový kód	Komentář
...	
N120 IDS=2 WHENEVER \$AA_IM[AX4] >	; synchronní akce
100 DO ACCLIMA[AX4]=80	
N130 LEADON(AX4, X, 2)	; aktivování vazby řídící hodnotou
...	

12.3.3 Aktivování specifických technologických hodnot dynamiky (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)

Prostřednictvím G-skupiny "Technologie" je možné pro až 5 různých technologických kroků postupu opracování aktivovat k nim vyhovující dynamiku.

Hodnoty dynamiky a příkazy G-funkcí jsou nastavovány v konfiguraci a proto jsou závislé na nastaveních strojních parametrů (--> výrobce stroje!).

Literatura:

Příručka k funkcím, Základní funkce, Režim řízení pohybu po dráze, přesné najetí, funkce Look Ahead (B1)

Syntaxe**Aktivování hodnot dynamiky:**

DYNNORM
DYNPOS
DYNROUGH
DYNSEMIFIN
DYNFINISH

Poznámka

Hodnoty dynamiky jsou v platnosti už v bloku, ve kterém je příslušný příkaz G-funkce naprogramován. Nedochází k žádnému zastavení obrábění.

Čtení nebo zápis určitého prvku pole:

R<m>=\$MA... [n, X]
\$MA... [n, X]=<hodnota>

Význam

DYNNORM:	Příkaz G-funkce pro aktivování normální dynamiky		
DYNPOS:	Příkaz G-funkce pro aktivování dynamiky pro režim nastavování polohy, vrtání závitů		
DYNROUGH:	Příkaz G-funkce pro aktivování dynamiky pro obrábění nahrubo		
DYNSEMIFIN:	Příkaz G-funkce pro aktivování dynamiky pro obrábění načisto		
DYNFINISH:	Příkaz G-funkce pro aktivování dynamiky pro jemné obrábění načisto		
R<m>:	Početní parametr s číslem <m>		
\$MA...[n,X]:	Strojní parametr s prvkem pole určujícím dynamiku		
<n>:	Index pole		
	Rozsah hodnot:		0 ... 4
	0	Normální dynamika (DYNNORM)	
	1	Dynamika pro režim polohování (DYNPOS)	
	2	Dynamika pro obrábění nahrubo (DYNROUGH)	
	3	Dynamika pro obrábění načisto (DYNSEMIFIN)	
	4	Dynamika pro jemné obrábění načisto (DYNFINISH)	
<X>:	Adresa osy		
<hodnota>:	Hodnota dynamiky		

Příklady

Příklad 1: Aktivování hodnot dynamiky

Programový kód	Komentář
DYNNORM G1 X10	; základní nastavení
DYNPOS G1 X10 Y20 Z30 F...	; režim polohování, vrtání závitu
DYNROUGH G1 X10 Y20 Z30 F10000	; obrábění nahrubo
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; obrábění načisto
DYNFINISH G1 X10 Y20 Z30 F1000	; jemné obrábění načisto

Příklad 2: Čtení nebo zápis určitého prvku pole

Maximální zrychlení pro obrábění nahrubo, osa X.

Programový kód	Komentář
R1=\$MA_MAX_AX_ACCEL[2,X]	; čtení
\$MA_MAX_AX_ACCEL[2,X]=5	; zápis

12.4 Najíždění s dopřednou regulací (FFWON, FFWOF)

Prostřednictvím dopředné regulace se při pohybu po dráze snižuje na rychlosti závislá velikost doběhového úseku na nulu. Pohyby s dopřednou regulací umožňují vyšší přesnost pohybu po dráze a tím i lepší výsledky obrábění.

Syntaxe

FFWON

FFWOF

Význam

FFWON:	Příkaz pro zapnutí dopředné regulace
FFWOF:	Příkaz pro vypnutí dopředné regulace

Poznámka

Druh dopředné regulace je definován pomocí strojních parametrů, stejně jako je takto určeno, u kterých dráhových os má být pohyb pomocí dopředné regulace řízen.

Standard: Dopředná regulace závisující na rychlosti

Volba: Dopředná regulace závisující na zrychlení

Příklad:

Programový kód

N10 FFWON

N20 G1 X... Y... F900 SOFT

12.5 Programovatelná přesnost kontury (CPRECON, CPRECOF)

Funkce "Programovatelná přesnost kontury" snižuje prostřednictvím automatického přizpůsobení rychlosti chybu dráhy na zakřivených konturách.

Přesnost kontury, kterou je potřeba dodržet, se zadává v závislosti na konfiguraci stroje (MD20470 \$MC_MC_CPPEC_WITH_FFW, viz informace od výrobce stroje) buď pomocí nastavovaného parametru \$SC_CONTPREC nebo pomocí naprogramované tolerance kontury CTOL. Čím menší je tato hodnota a čím menší je parametr K_v geometrických os, tím více se rychlost pohybu po dráze na zakřivených konturách snižuje.

Funkce "Programovatelná přesnost kontury" se v NC programu aktivuje, příp. deaktivuje pomocí příkazů CPRECON a CPRECOF.

Syntaxe

CPRECON

...

CPRECOF

Význam

CPRECON:	Volání příkazu G-funkce: Aktivování funkce "Programovatelná přesnost kontury"	
	Platnost:	modální
CPRECOF:	Volání příkazu G-funkce: Deaktivování funkce "Programovatelná přesnost kontury"	
	Platnost:	modální

Funkce CPRECON a CPRECOF dohromady tvoří skupinu G-funkcí 39 (Programovatelná přesnost kontury).

Poznámka

Pomocí nastavovaného parametru \$SC_MINFEED (minimální posuv po dráze při CPRECON) může uživatel zadat minimální rychlost pro pohyb po dráze.

Posuv nebude omezen na hodnotu nižší, než je tato hodnota, leda že by byla naprogramována nižší hodnota F nebo si dynamická omezení os vynutila nižší rychlost pohybu po dráze.

Poznámka

Funkce "Programovatelná přesnost kontury" uvažuje jen geometrické osy dané dráhy. Na rychlosti polohovacích os nemá žádný vliv.

Příklad:

Programový kód	Komentář
N10 G0 X0 Y0	
N20 CPRECON	; aktivování funkce "programovatelná přesnost kontury"

Programový kód	Komentář
N30 G1 G64 X100 F10000	; obrábění s rychlostí 10 m/min v režimu řízení pohybu po dráze
N40 G3 Y20 J10	; automatické omezení posuvu v bloku kruhového oblouku
N50 G1 X0	; posuv opět bez omezení (10 m/min)
...	
N100 CPRECOF	; deaktivování funkce "programovatelná přesnost kontury"
N110 G0 ...	

Literatura

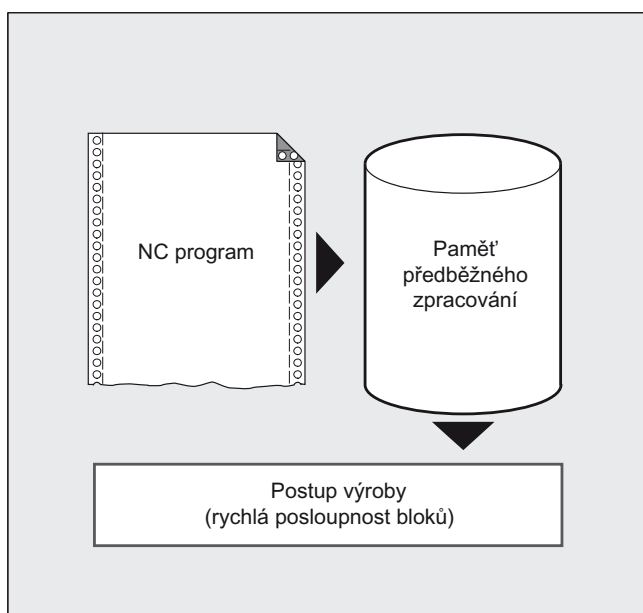
Pokud budete potřebovat informace o programování příkazu CTOL, viz "Programování tolerance kontury/orientace (CTOL, OTOL, ATOL) (Strana 556)".

Pokud budete potřebovat podrobné informace o funkci "Programovatelná přesnost kontury", viz:

Příručka k funkcím, Speciální funkce, Monitorování tunelu kontury (K6), kapitola: "Programovatelná přesnost kontury"

12.6 Zpracování programu s pamětí předběžného zpracování (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

V závislosti na úrovni svého vybavení disponuje řídicí systém určitým množstvím tzv. paměti pro předběžné zpracování, do které se ukládají připravené hotové bloky před svým zpracováním a odkud jsou v průběhu samotného obrábění předávány jako rychlá posloupnost bloků. Díky tomu mohou být krátké bloky obrobny s vysokými rychlostmi. Jestliže to doba prodlevy řídicího systému připouští, je paměť předběžného zpracování v zásadě naplněna.



Označení zpracovávaného úseku

Zpracovávaný úsek programu, který má být dočasně uložen do paměti předběžného zpracování, je ve výrobním programu označen na začátku příkazem „STOPFIFO“ a na konci příkazem „STARTFIFO“. Zpracování připravených a do vyrovnávací paměti uložených bloků začíná teprve po příkazu „STARTFIFO“ nebo když je paměť předběžného zpracování plná.

Automatické řízení paměti předběžného zpracování

Automatické řízení paměti předběžného zpracování se vyvolává pomocí příkazu „FIFOCTRL“. Příkaz „FIFOCTRL“ z počátku funguje úplně stejně jako příkaz „STOPFIFO“. Při každé sekvenci programových příkazů se čeká, dokud není paměť předběžného zpracování plná, a potom se spouští zpracování. Rozdíl mezi oběma možnostmi spočívá v chování v případě vyprázdnění paměti předběžného zpracování: V případě příkazu „FIFOCTRL“ se od úrovně 2/3 naplnění postupně snižuje rychlost pohybu po dráze, aby se zabránilo úplnému vyprázdnění a zabrždění pohybu os až do úplného klidu.

Zastavení předběžného zpracování

Když je v bloku naprogramován příkaz „STOPRE“, příprava bloků a ukládání do vyrovnávací paměti se zastaví. Následující blok bude zpracován teprve tehdy, až jsou zcela zpracovány

všechny bloky, které již dříve byly připraveny a uloženy do paměti. Předcházející blok bude zastaven v přesném najetí (jako při G9).

UPOZORNĚNÍ

Přerušení programu

Když jsou aktivovány korekce nástroje a splinová interpolace, neměl by být naprogramován žádný příkaz „STOPRE“, protože by došlo k přerušení posloupnosti k sobě patřících bloků.

Syntaxe

Tabulka 12-1 Označení zpracovávaného úseku:

```
STOPFIFO
...
STARTFIFO
```

Tabulka 12-2 Automatické řízení paměti předběžného zpracování:

```
...
FIFOCTRL
...
```

Tabulka 12-3 Zastavení předběžného zpracování:

```
...
STOPRE
...
```

Poznámka

Příkazy „STOPFIFO“, „STARTFIFO“, „FIFOCTRL“ a „STOPRE“ musí být naprogramovány v samostatném bloku.

12.6 Zpracování programu s pamětí předběžného zpracování (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Význam

STOPFIFO:	Příkaz "STOPFIFO" označuje začátek zpracovávaného úseku, který má být dočasně uložen do paměti předběžného zpracování. Pomocí příkazu "STOPFIFO" se zpracovávání pozastaví a paměť předběžného zpracování se bude plnit, dokud: • není rozpoznán příkaz „STARTFIFO“ nebo „STOPRE“. • se paměť předběžného zpracování nenaplní • nebo • není dosaženo konce programu.
STARTFIFO:	Pomocí příkazu "STARTFIFO" se spouští rychlé zpracování úseku programu; souběžně s tím probíhá plnění paměti předběžného zpracování.
FIFOCTRL:	Spuštění automatického ukládání do paměti předběžného zpracování
STOPRE:	Zastavení předběžného zpracování

Poznámka

Plnění paměti předběžného zpracování se neprovádí, příp. je přerušeno, pokud zpracovávaný úsek programu obsahuje příkazy, které vyžadují okamžité zpracování (najíždění na referenční bod, měřicí funkce, ...).

Poznámka

Při přístupu ke stavovým údajům stroje (\$SA...) generuje řídicí systém interní zastavení předběžného zpracování.

Příklad: Zastavení předběžného zpracování

Programový kód	Komentář
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; Blok měření s měřicí sondou na prvním měřicím vstupu a přímková interpolace.
N40 STOPRE	; Zastavení předběžného zpracování.
...	

12.7 Definice oblasti odloženého zastavení (DELAYFSTON, DELAYFSTOF)

Pro definování podmíněně přerušitelných oblastí ve výrobním programu (Stop-Delay) slouží předem definované procedury DELAYFSTON a DELAYFSTOF.

Poznámka

V synchronních akcích je použití funkcí DELAYFSTON a DELAYFSTOF **nepřípustné!**

Syntaxe

```
DELAYFSTON
...
DELAYFSTOF
```

Význam

DELAYFSTON:	Definice začátku oblasti odloženého zastavení	
	Samostatný blok:	ano
DELAYFSTOF:	Definice konce oblasti odloženého zastavení (Stop_Delay)	
	Samostatný blok:	ano

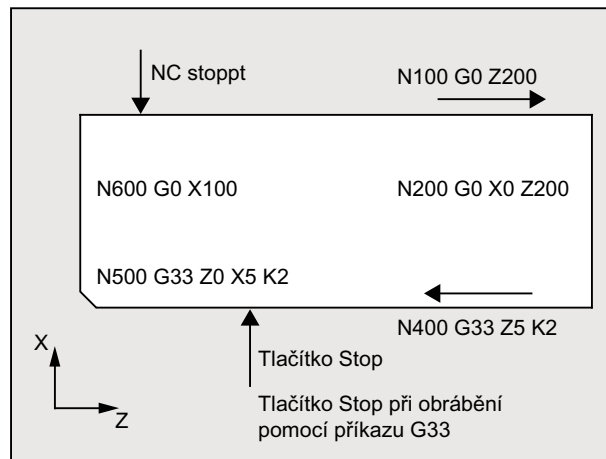
Příklad programování

Ve smyčce se opakuje následující blok:

Programový kód

```
...
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
...
```

V následujícím obrázku je vidět, že když uživatel v oblasti odloženého zastavení (Stop-Delay) stiskne tlačítko "Stop", zahájí NC systém operaci brždění mimo oblast odloženého zastavení, tzn. v bloku N100. Díky tomu se NC systém v oblasti před blokem N100 zastaví.



Další informace

Konec podprogramu

Na konci podprogramu, v němž byl vyvolán příkaz DELAYFSTON, se implicitně aktivuje příkaz DELAYFSTOF.

Vnoření

Pokud podprogram 1 vyvolává v oblasti odloženého zastavení (Stop-Delay) podprogram 2, stává se celý podprogram 2 oblastí odloženého zastavení. Zejména příkazy DELAYFSTOF nemají v podprogramu 2 žádný efekt.

Příklad:

Programový kód	Komentář
N10010 DELAYFSTON	; bloky s čísly N10xxx z programové úrovně 1
N10020 R1 = R1 + 1	
N10030 G4 F1	; začátek oblasti odloženého zastavení (Stop-Delay)
...	
N10040 Podprogram 2	
...	
...	; interpretace podprogramu 2
N20010 DELAYFSTON	; žádná platnost, opakovaný začátek, 2. rovina
...	
N20020 DELAYFSTOF	; žádná platnost, konec na jiné úrovni
N20030 RET	
N10050 DELAYFSTOF	; konec oblasti odloženého zastavení (Stop-Delay) na stejné úrovni
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; oblast odloženého zastavení (Stop-Delay) končí Zastavení jsou od tohoto okamžiku uskutečňována okamžitě.

Systémové proměnné

To, zda se zpracování výrobního programu v aktuálním okamžiku nachází v oblasti odloženého zastavení, je možno zjistit pomocí následujících systémových proměnných:

- ve výrobním programu pomocí proměnné \$P_DELAYFST
- v synchronních akcích pomocí proměnné \$AC_DELAYFST

Hodnota	Význam
0	Oblast odloženého zastavení není aktivní
1	Oblast odloženého zastavení je aktivní

12.8 Zabránění pozice programu pro SERUPRO (IPTRLOCK, IPTRUNLOCK)

Pro určité komplikované mechanické situace na stroji je zapotřebí, aby se zabránilo vyhledávání bloku pomocí příkazu SEROPRO.

Pomocí programovatelného ukazatele přerušení existuje možnost, jak při operaci "Vyhledávání místa přerušení" zasáhnout a přesunout se před prohledávané místo.

Ve výrobním programu mohou být definovány také oblasti vyhledávání, do kterých NC systém ještě nemůže znovu vstoupit. Při přerušení programu se v NC systému zaznamená naposled zpracovávaný blok, který potom může být pomocí uživatelského rozhraní HMI vyhledán.

Syntaxe

IPTRLOCK
IPTRUNLOCK

Tyto příkazy musí být uvedeny na samostatném řádku výrobního programu a umožňují pracovat s programovatelným ukazatelem přerušení.

Význam

IPTRLOCK:	Začátek úseku programu pro vyhledávání
IPTRUNLOCK:	Konec úseku programu pro vyhledávání

Oba příkazy jsou přípustné jen ve výrobních programech, **nikoli** v synchronních akcích.

Příklad:

Vnoření úseků programu, které mohou být prohledávány, na dvou programových úrovních s implicitním příkazem „IPTRUNLOCK“. Implicitní příkaz „IPTRUNLOCK“ v podprogramu 1 končí oblast určenou pro vyhledávání.

Programový kód	Komentář
N10010 IPTRLOCK()	
N10020 R1 = R1 + 1	
N10030 G4 F1	; blok s pozastavením, úsek programu určený pro vyhledávání začíná
...	
N10040 Podprogram 2	
...	; interpretace podprogramu 2
N20010 IPTRLOCK ()	; žádná platnost, opakovaný začátek
...	
N20020 IPTRUNLOCK ()	; žádná platnost, konec v jiné rovině
N20030 RET	
...	
N10060 R2 = R2 + 2	
N10070 RET	; konec úseku programu pro vyhledávání
N100 G4 F2	; hlavní program pokračuje

Přerušení na řádce 100 potom znovu vytvoří ukazatel přerušení.

Další informace

Zjišťování a prohledávání oblastí určených pro vyhledávání

Úseky programu určené pro vyhledávání jsou označeny příkazy NC jazyka „IPTRLOCK“ a „IPTRUNLOCK“.

Příkaz „IPTRLOCK“ zmrazí ukazatele přerušení na určitém bloku (SBL1), který může být realizován v hlavním zpracování. Tento blok se v následujícím označí jako blok se zastavením. Pokud se po příkazu „IPTRLOCK“ vyskytne přerušení programu, pak je možné na uživatelském rozhraní HMI tento tak zvaný blok zastavení vyhledat.

Opětovné nastavení na aktuální blok

Ukazatel přerušení se pomocí příkazu „IPTRLOCK“ pro následující úsek programu nastaví na aktuální blok k bodu, kde došlo k přerušení.

Po nalezení cíle vyhledávání může být pomocí stejného bloku zastavení cíl vyhledávání opakován.

Uživatelé editovaný ukazatel přerušení musí být pomocí HMI znovu odstraněn.

Pravidla pro vnoření

Chování příkazů NC jazyka „IPTRLOCK“ a „IPTRUNLOCK“ ve spojení s vnořeními a konci podprogramů se řídí následujícími pravidly:

1. Na konci podprogramu, v němž byl vyvolán příkaz "IPTRLOCK", se implicitně aktivuje příkaz "IPTRUNLOCK".
2. Oblast určená pro vyhledávání nastavená příkazem „IPTRLOCK“ nemá žádný efekt.
3. Pokud podprogram 1 vyvolá v oblasti určené pro vyhledávání podprogram 2, zůstává celý podprogram 2 určený pro vyhledávání. Zejména příkaz "IPTRUNLOCK" v podprogramu 2 nemá žádný efekt.

Pokud budete potřebovat další informace, viz
/FB1/, Příručka k funkcím, Základní funkce; BAG, kanál, zpracování programu (K1).

Systémové proměnné

Oblast určená pro vyhledávání může být ve výrobním programu rozpoznána pomocí proměnné "\$P_IPTRLOCK".

Automatický ukazatel vyhledávání

Funkce automatického ukazatele vyhledávání automaticky definuje dříve nastavený typ vazby jako objekt určený pro vyhledávání. Prostřednictvím strojního parametru se pro

- Elektronickou převodovku s příkazem „EGON“
- Osovou vazbu řídicí hodnotou s příkazem „LEADON“

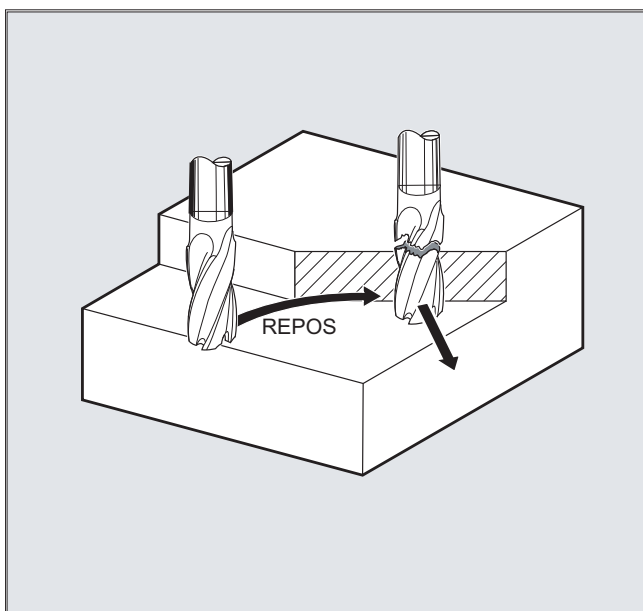
aktivuje automatický ukazatel přerušení. Jestliže se naprogramované a prostřednictvím strojního parametru aktivované automatické ukazatele přerušení překrývají, vytvoří se největší možná oblast určená pro vyhledávání.

12.9 Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Jestliže v průběhu zpracovávání běžící program přerušíte a s nástrojem odjedete od obrobku - např. kvůli zlomení nástroje nebo kvůli změření obrobku - můžete potom na konturu znovu najet ve zvoleném bodě, přičemž tento pohyb je řízený programem.

Příkaz `REPOS` způsobuje ASUP, stejně jako návrat zpět z podprogramu (např. M17).

Následující bloky se už neuskuteční. Pokud budete potřebovat informace o přerušování programu, viz také "Rutiny přerušení (ASUP) (Strana 128)".



Syntaxe

```
REPOSA RMIBL DISPR=...
REPOSA RMBBL
REPOSA RMEBL
REPOSA RMNBL
REPOSL RMIBL DISPR=...
REPOSL RMBBL
REPOSL RMEBL
REPOSL RMNBL
REPOSQ RMIBL DISPR=... DISR=...
REPOSQ RMBBL DISR=...
REPOSQ RMEBL DISR=...
REPOSQA DISR=...
REPOSH RMIBL DISPR=... DISR=...
REPOSH RMBBL DISR=...
REPOSH RMEBL DISR=...
REPOSHA DISR=...
```

Význam

Volba najížděcí dráhy

REPOSA:	Najetí zpět na konturu geometrickými osami po přímkách. Svou polohu mění také všechny ostatní kanálové osy.
REPOSL:	Najetí zpět na konturu geometrickými osami po přímkách. Všechny ostatní kanálové osy musí být explicitně naprogramovány.
REPOSQ DISR=... :	Najetí zpět na konturu geometrickými osami po čtvrtkruhu s rádiusem DISR. Všechny ostatní kanálové osy musí být explicitně naprogramovány.
REPOSQA DISR=... :	Najetí zpět na konturu geometrickými osami po čtvrtkruhu s rádiusem DISR. Svou polohu mění také všechny ostatní kanálové osy.
REPOSH DISR=... :	Najetí zpět na konturu geometrickými osami po půlkruhu s průměrem DISR. Všechny ostatní kanálové osy musí být explicitně naprogramovány.
REPOSHA DISR=... :	Najetí zpět na konturu geometrickými osami po půlkruhu s rádiusem DISR. Svou polohu mění také všechny ostatní kanálové osy.

Volba bodu pro opětovné najetí

RMIBL:	Najíždění na bod, kde došlo k přerušení
RMIBL DISPR=...:	Vstupní bod ve vzdálenosti DISPR v mm/palcích před bodem, kde došlo k přerušení
RMBBL:	Najíždění na počáteční bod bloku
RMEBL:	Najíždění na koncový bod bloku
RMEBL DISPR=... :	Najíždění na koncový bod bloku ve vzdálenosti DISPR před koncovým bodem
RMNBL:	Najíždění na nejbližší ležící bod dráhy
A0 B0 C0 :	Osy, kterými se má najíždět

Poznámka

Kompatibilita

Aby byla zajištěna kompatibilita se staršími verzemi programového vybavení, můžete režim najíždění pro funkci REPOS i nadále programovat pomocí modálních příkazů RMI, RMB, RME a RMN. Při použití ve stavu ASUP by měl být tento příkaz opatřen atributem SAVE v příkazu PROC. Jinak se bude uplatňovat v režimu ASUP modální režim najíždění REPOS, a pokud se liší od standardního nastavení RMI, použije se i v následujících operacích REPOS.

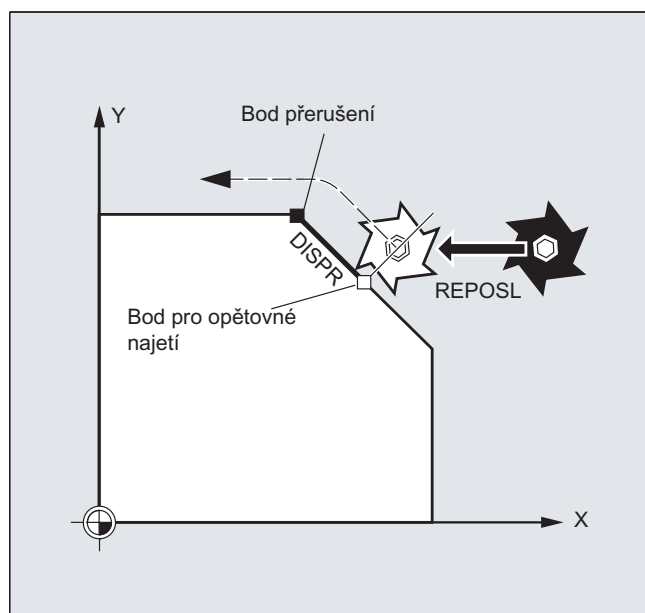
Najetí zpět na konturu po přímkách, REPOSA, REPOSL

Nástroj najíždí na bod pro opětovné najetí přímo po přímce.

Příklad

```
REPOSL RMIBL DISPR=6 F400
```

12.9 Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

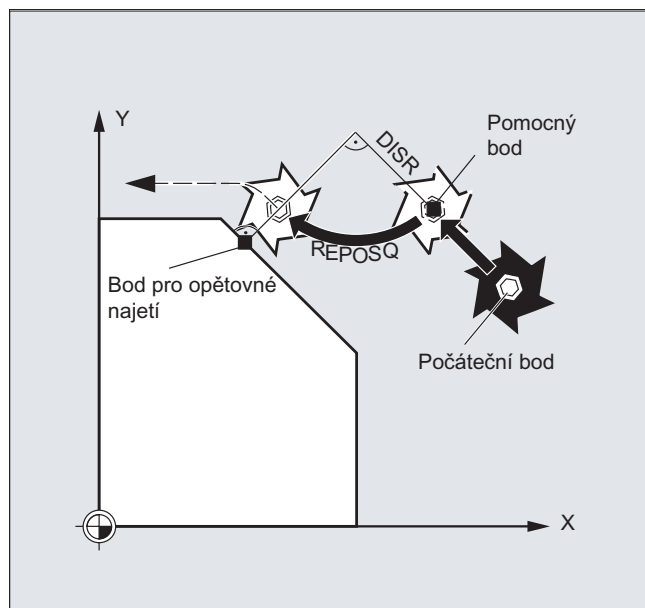


Najetí zpět na konturu po čtvrtkruhu, REPOSQ, REPOSQA

Nástroj najíždí na bod pro opětovné najetí po čtvrtkruhové dráze s rádiusem $DISR = \dots$.
Potřebný pomocný bod mezi počátečním bodem a bodem pro opětovné najetí vypočítává řídicí systém automaticky.

Příklad

REPOSQ RMIBL DISR=10 F400



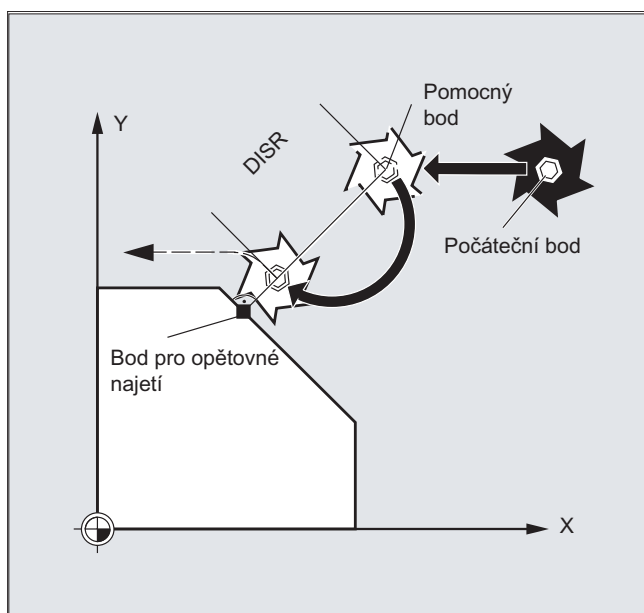
Najetí zpět na konturu po půlkruhu, REPOSH, REPOSHA

Nástroj najíždí na bod pro opětovné najetí po půlkruhové dráze s průměrem $DISR = \dots$

Potřebný pomocný bod mezi počátečním bodem a bodem pro opětovné najetí vypočítává řídicí systém automaticky.

Příklad

REPOSH RMIBL DISR=20 F400

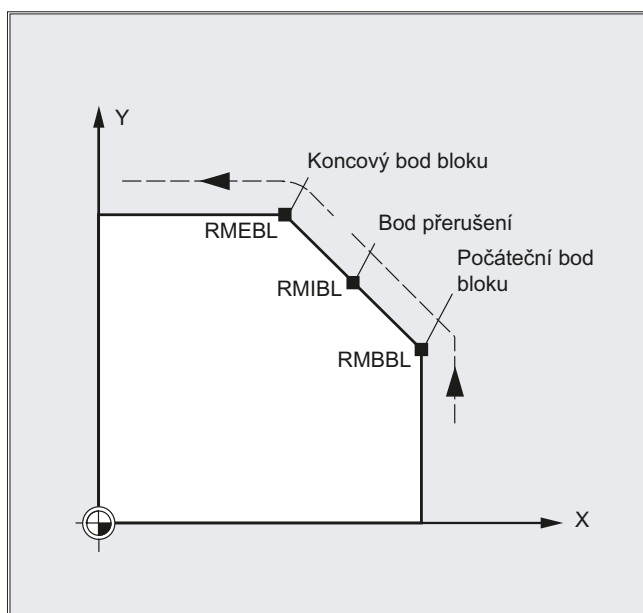


Definice bodu pro opětovné najíždění (nikoli pro najíždění SERUPRO s RMNBL)

V souvislosti s NC blokem, ve kterém bylo zpracování programu přerušeno, si můžete vybrat mezi třemi body pro opětovné najetí:

- RMIBL, bod, kde došlo k přerušení
- RMBBL, počáteční bod bloku, příp. poslední koncový bod
- RMEBL, koncový bod bloku

12.9 Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)



Pomocí příkazů RMIBL DISPR=..., příp. RMEBL DISPR=... můžete definovat bod pro opětovné najíždění, který leží před bodem, v němž došlo k přerušení, příp. před koncovým bodem bloku.

Pomocí příkazu DISPR=... zadáváte dráhu na kontuře v mm/palcích, o kterou bod pro opětovné najíždění leží před bodem, v němž došlo k přerušení, příp. před koncovým bodem. Tento bod může - i v případě větších hodnot - ležet maximálně v počátečním bodu bloku.

Jestliže není naprogramována žádná hodnota parametru DISPR=..., platí nastavení DISPR=0 a v důsledku toho se použije bod, kde došlo k přerušení (v případě příkazu RMIBL), příp. koncový bod bloku (v případě RMEBL).

Znaménko parametru DISPR

Znaménko parametru DISPR se vyhodnocuje. Je-li znaménko kladné, funkce se chová, jak bylo popsáno výše.

V případě záporného znaménka se nástroj znovu nasazuje za bod, kde došlo k přerušení, příp. v případě příkazu RMBBL za počáteční bod.

Vzdálenost bodu přerušení a bodu nasazení nástroje vyplývá z hodnoty uložené v parametru DISPR. Tento bod může i v případě větších hodnot ležet maximálně v počátečním bodu bloku.

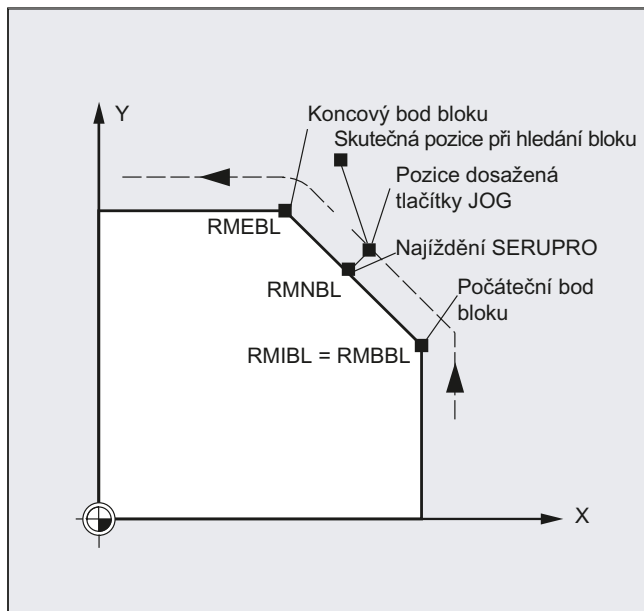
Příklad použití:

Pomocí senzoru bylo zjištěno přiblížení k upínací čelisti. Aktivuje se program ASUP, pomocí kterého se upínací čelist objede.

Nakonec se znovu najede na bod za upínací čelistí se zápornou hodnotou parametru DISPR a program bude pokračovat.

Najíždění pomocí příkazu SEROPRO s RMNBL

Jestliže si v průběhu obrábění okolnosti vyžádají na libovolném místě přerušení, pak se pomocí najíždění SERUPRO s RMNBL najede co možno nejbližší před místo, kde došlo k přerušení, aby se nakonec opracovala pouze zbývající dráha. Za tím účelem spouští uživatel operaci SERUPRO na bloku, kde došlo k přerušení, a pomocí tlačítek JOG najíždí před poškozené místo cílového bloku.



Poznámka

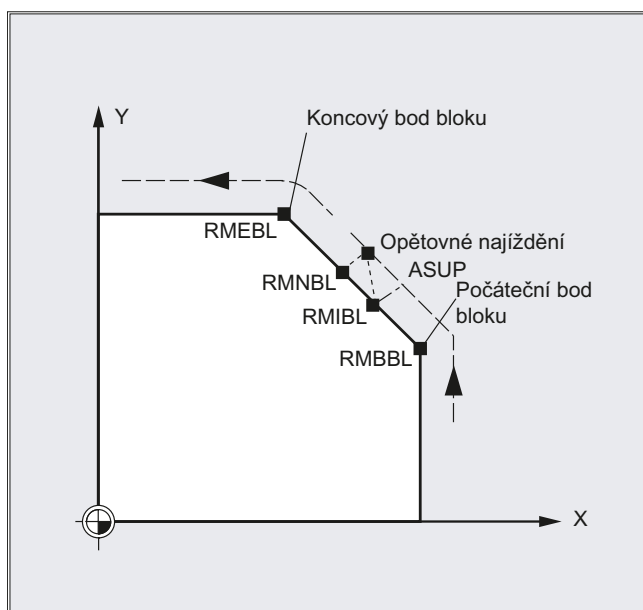
SERUPRO

Pro operaci SERUPRO jsou parametry RMIBL a RMBBL identické. Parametr RMNBL není omezen pouze na operaci SERUPRO, nýbrž platí všeobecně.

12.9 Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Najíždění před nejbližší ležící bod dráhy RMNBL

Pokud jde o okamžik interpretace příkazu REPOSA, po přerušení není blok pro opětovné najetí s příkazem RMNBL zahájen celý ještě jednou, nýbrž je opracovávána pouze zbytková dráha. Najíždí se na bod dráhy, který leží co možno nejbližší bloku, v němž došlo k přerušení.



Stavové informace pro platný režim REPOS

Platný režim funkce REPOS přerušeného bloku je možné načíst pomocí synchronních akcí prostřednictvím proměnné \$AC_REPOS_PATH_MODE.

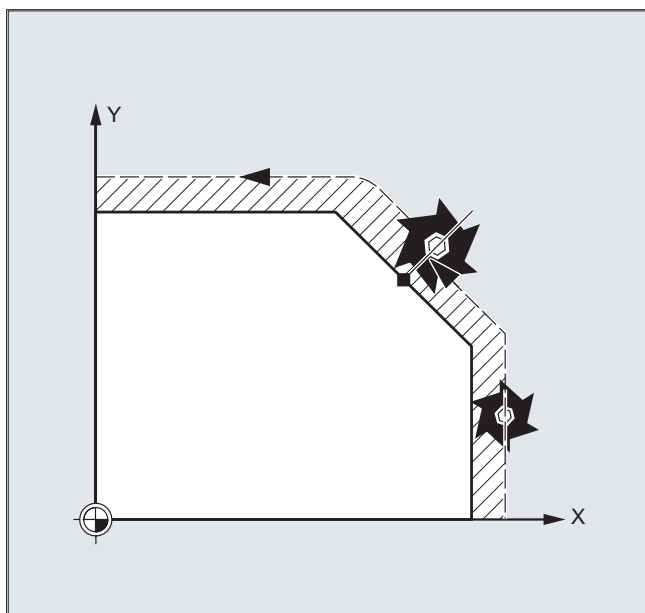
- 0 Najíždění není definováno
- 1 RMBBL: Najíždění na začátek
- 2 RMIBL: Najíždění na místo, kde došlo k přerušení
- 3 RMEBL: Najíždění na koncový bod bloku
- 4 RMNBL: Najíždění na bod dráhy, který leží nejbližší bloku, kde došlo k přerušení

Najíždění s novým nástrojem

Jestliže jste zpracovávání programu přerušili kvůli zlomení nástroje:

Naprogramováním nového D_číslo bude zpracování programu pokračovat od bodu opětovného najetí se změněnými hodnotami korekčních parametrů nástroje.

V případě změněných hodnot korekčních parametrů nástroje se může stát, že na místo, v němž došlo k přerušení, nebude možné najet. V tomto případě se bude najíždět na bod na nové kontuře, který leží co možno nejbližší bodu, v němž došlo k přerušení (v případě potřeby s úpravou podle parametru DISPR).



Najíždění na konturu

Pohyb, se kterým nástroj znovu najíždí na konturu, může být naprogramován. Pro adresy os, které se mají pohybovat, zadejte nulovou hodnotu.

Pomocí příkazů REPOSA, REPOSQA a REPOSHA jsou automaticky všechny osy nastavovány na původní pozici. Zadání os není zapotřebí.

Jestliže je naprogramován příkaz REPOSL, REPOSQ a REPOSH, pohybují se všechny geometrické osy automaticky, aniž by byly v příkazu zadány. Všechny ostatní osy musí být v příkazu uvedeny.

Pro kruhové pohyby s příkazy REPOSH a REPOSQ platí:

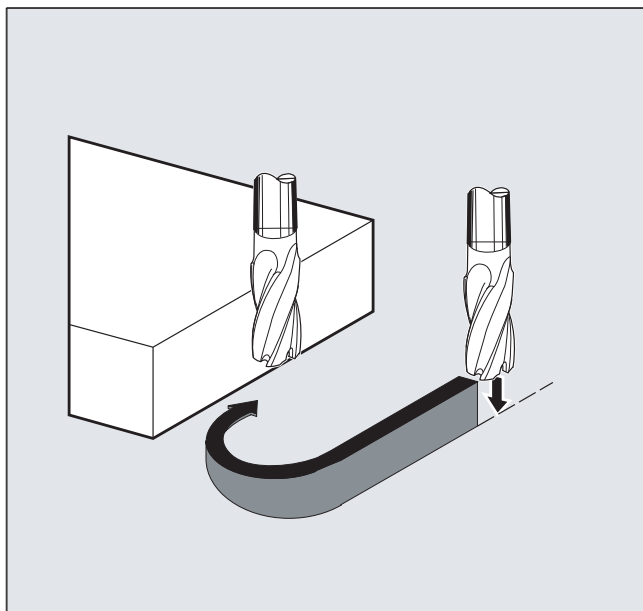
Kruhový pohyb se uskutečňuje v zadané pracovní rovině G17 až G19.

Jestliže je v najížděcím bloku uvedena třetí geometrická osa (směr přísuvu), potom se v případě, že si poloha nástroje a naprogramovaná poloha ve směru přísuvu neodpovídají, uskuteční zpětný pohyb na bod pro opětovné najetí po spirále.

12.9 Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

V následujících případech dochází k automatickému přepnutí na lineární najíždění REPOSL:

- Nezadali jste žádnou hodnotu pro parametr DISR.
- Neexistuje žádný definovaný směr najíždění (přerušení programu v bloku bez informací o posuvech).
- V případě směru najíždění kolmo na aktuální pracovní rovinu.



12.10 Ovlivňování vedení pohybu

12.10.1 Procentuální korekce ryvu (JERKLIM)

Pomocí příkazu NC jazyka „JERKLIM“ může být snížen nebo zvýšen prostřednictvím strojního parametru nastavený maximální možný rýv osy při pohybu po dráze v kritických úsecích programu.

Předpoklady

Musí být aktivní režim zrychlení SOFT.

Platnost

Funkce se uplatňuje::

- v automatických provozních režimech
- jen pro dráhové osy

Syntaxe

JERKLIM[<osa>]=<hodnota>

Význam

JERKLIM:	Příkaz pro korekci ryvu	
<osa>:	Osa stroje, jejíž mezní hodnota ryvu má být přizpůsobena.	
<hodnota>:	Procentuální hodnota korekce vztahující se na v konfiguraci nastavený maximální rýv osy při pohybu po dráze (MD32431 \$MA_MAX_AX_JERK).	
	Rozsah hodnot:	1 ... 200
	Hodnota 100 má za následek žádné ovlivňování ryvu.	

Poznámka

Chování příkazu JERKLIM na konci výrobního programu a při resetu kanálu se nastavuje v konfiguraci pomocí bitu 0 strojního parametru MD32320 \$MA_DYN_LIMIT_RESET_MASK:

- Bit 0 = 0:
Naprogramovaná hodnota pro příkaz JERKLIM a při resetu kanálu/M30 se nastavuje zpět na 100%.
- Bit 0 = 1:
Naprogramovaná hodnota pro příkaz JERKLIM zůstává při resetu kanálu/M30 zachována.

Příklad:

Programový kód	Komentář
...	
N60 JERKLIM[X]=75	; osové obrábění načisto ve směru osy X má být uskutečňováno se zrychlením/zpomalením odpovídajícím jen s maximálně 75% ryvu přípustného pro osu
...	

12.10.2 Procentuální korekce rychlosti (VELOLIM)

Pomocí příkazu VELOLIM je možné ve výrobním programu nebo v synchronní akci snížit maximální možnou rychlost osy nastavenou prostřednictvím strojního parametru, příp. maximální možné otáčky vřetena závislé na stupni převodovky.

Platnost

Funkce se uplatňuje::

- v automatických provozních režimech
- na dráhové a polohovací osy
- na vřetena v režimu vřetena/osy

Syntaxe

VELOLIM[<osa/vřeteno>]=<hodnota>

Význam

VELOLIM:	Příkaz pro korekci rychlosti
<osa/vřeteno>:	Osa nebo vřeteno, jejichž mezní hodnota rychlosti nebo otáček má být přizpůsobena. Funkce VELOLIM pro vřetena Prostřednictvím strojního parametru (MD30455 \$MA_MISC_FUNCTION_MASK, bit 6) může být pro programování výrobních programů nastaveno, zda má být funkce „VELOLIM“ uplatňována nezávisle na momentálním použití jako vřeteno nebo jako osa (bit 6 = 1) nebo zda může být naprogramována odděleně pro každý provozní režim (bit 6 = 0). Jestliže je v konfiguraci nastaveno oddělené působení, je možné při programování pomocí identifikátorů vybírat z následujících možností: • Identifikátor vřetena S<n> pro provozní režimy vřetena • Identifikátor osy, např. "C", pro režim osy

<hodnota>:	Procentuální hodnota korekce	
	Hodnota korekce se vztahuje:	
	- v případě os / vřeten v režimu osy (MD30455 bit 6 == 0): na maximální rychlost osy nastavenou v konfiguraci (MD32000 \$MA_MAX_AX_VELO). - v případě vřeten v režimu vřetena/osy (MD30455 bit 6 == 1): na maximální otáčky momentálního převodového stupně (MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])	
	Rozsah hodnot:	1 ... 100
	Hodnota 100 má za následek, že rychlost, příp. otáčky nejsou ovlivňovány.	

Poznámka**Chování na konci výrobního programu a při resetu kanálu**

Chování příkazu „VELOLIM“ na konci výrobního programu a při resetu kanálu může být nastaveno pomocí strojního parametru: MD32320 \$MA_DYN_LIMIT_RESET_MASK, Bit 0

Rozpoznání aktivního omezení otáček v režimu vřetena

Omezení otáček pomocí příkazu „VELOLIM“ (menší jak 100%) může být v režimu vřetena rozpoznáno prostřednictvím následující systémové proměnné:

- \$AC_SMAXVELO (maximální možné otáčky vřetena)
- \$AC_SMAXVELO_INFO (identifikace příčiny omezení otáček)

Příklady**Příklad 1: Omezení rychlosti osy stroje**

Programový kód	Komentář
...	
N70 VELOLIM[X]=80	; osově obrábění načisto ve směru osy X má být uskutečňováno s rychlostí odpovídající jen s maximálně 80% rychlosti přípustné pro osu
...	

Příklad 2: Omezení otáček vřetena

Programový kód	Komentář
N05 VELOLIM[S1]=90	; omezení maximálních otáček vřetena 1 na 90% z 1000 ot/min
...	
N50 VELOLIM[C]=45	; omezení otáček vřetena na 45% z 1000 ot/min, pokud C je osový identifikátor S1
...	

Nastavení strojních parametrů pro vřeteno 1 (AX5):

- Maximální otáčky stupně převodovky 1 = 1000 ot/min
MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- Naprogramování příkazu „VELOLIM“ se uplatňuje společně pro režim osy a režim vřetena nezávisle na naprogramovaném identifikátoru:
MD30455 \$MA_MISC_FUNCTION_MASK[AX5], bit 6 = 1

12.10.3 Příklad programování pro příkazy JERKLIM a VELOLIM

Následující program představuje příklad použití pro procentuální omezení ryvu a rychlosti:

Programový kód	Komentář
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	; osově obrábění načisto ve směru osy X má být uskutečňováno s rychlostí odpovídající jen s max. 5% rychlosti přípustné pro osu ; osově obrábění načisto ve směru osy Y může být uskutečňováno se zrychlením/zpomalením odpovídajícím s maximálně 200% ryvu přípust- ného pro osu
N1300 G1 X0 JERKLIM[X]=2	; osově obrábění načisto ve směru osy X má být uskutečňováno se zrychlením/zpomalením odpovídajícím jen s max. 2% ryvu přípustného pro osu
N1400 G1 Y0	
M30	

12.11 Programování tolerance kontury/orientace (CTOL, OTOL, ATOL)

Prostřednictvím adres CTOL, OTOL a ATOL je možné přizpůsobovat tolerance opracování definované ve strojních a nastavovaných parametrech pro funkce kompresoru, zaoblení a vyhlazování orientace ve výrobním programu.

Naprogramované hodnoty tolerancí platí, dokud nejsou znovu naprogramovány nebo dokud nejsou přiřazením záporné hodnoty vymazány. Kromě toho jsou vymazány na konci programu, při resetu. Po vymazání jsou opět v platnosti hodnoty tolerancí nastavené v parametrech.

Syntaxe

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

Význam

CTOL:	Adresa pro programování tolerance kontury		
	Oblast použití:	- všechny funkce kompresoru - všechny pohyby na přechodových zaobleních, kromě příkazů G641 a G644	
	Zastavení předběžného zpracování:	ne	
	Platnost:	modální	
	<Value>:	Hodnota pro toleranci kontury je délkový údaj.	
		Typ:	REAL
		Jednotka:	Palce/mm (v závislosti na momentálním nastavení měřicích jednotek)
		Rozsah hodnot:	≥ 0: Hodnota tolerance
			< 0: Vymazání naprogramované hodnoty tolerance ⇒ Znovu platí hodnoty tolerancí definované ve strojních, příp. nastavovaných parametrech.

12.11 Programování tolerance kontury/orientace (CTOL, OTOL, ATOL)

OTOL:	Adresa pro programování tolerance orientace		
	Oblast použití:	- všechny funkce kompresoru - Vyhlazování orientace pomocí příkazu ORISON - všechny pohyby na přechodových zaobleních, kromě příkazů G641, G644 a OSD	
	Zastavení předběžného zpracování:	ne	
	Platnost:	modální	
	<Value>:	Hodnota pro toleranci orientace je úhlový údaj.	
		Typ:	REAL
		Jednotka:	stupně
		Rozsah hodnot:	≥ 0: Hodnota tolerance < 0: Vymazání naprogramované hodnoty tolerance $\Rightarrow$ Znovu platí hodnoty tolerancí definované ve strojních, příp. nastavovaných parametrech.
ATOL:	Adresa pro programování tolerance určité osy		
	Oblast použití:	- všechny funkce kompresoru - Vyhlazování orientace pomocí příkazu ORISON - všechny pohyby na přechodových zaobleních, kromě příkazů G641, G644 a OSD	
	Zastavení předběžného zpracování:	ne	
	Platnost:	modální	
	<Axis>:	Název kanálové osy, pro kterou má naprogramovaná tolerance platit	
	<Value>:	Hodnota pro toleranci osy je v závislosti na typu osy (lineární nebo kruhová osa) buď délkovým nebo úhlovým údajem.	
		Typ:	REAL
		Jednotka:	pro lineární osy: Palce/mm (v závislosti na momentálním nastavení měřících jednotek) pro kruhové osy: stupně
		Rozsah hodnot:	≥ 0: Hodnota tolerance < 0: Vymazání naprogramované hodnoty tolerance $\Rightarrow$ Znovu platí hodnoty tolerancí definované ve strojních, příp. nastavovaných parametrech.

Poznámka

Kanálové hodnoty tolerancí naprogramované pomocí příkazů CTOL a OTOL mají vyšší prioritu než osové hodnoty tolerancí naprogramované pomocí příkazu ATOL.

Poznámka**Framy provádějící změnu měřítka**

Framy provádějící změnu měřítka ovlivňují naprogramované tolerance stejným způsobem, jako polohy os, tzn. relativní tolerance zůstává stejná.

Příklad:

Programový kód	Komentář
COMPCAD G645 G1 F10000	; aktivování funkce kompresoru COMPCAD
X... Y... Z...	; zde se uplatňují strojní a nastavované parametry
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; od tohoto místa se uplatňuje tolerance kontury 0,02 mm
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; od tohoto místa se uplatňuje tolerance kontury 0,005 mm
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; od tohoto místa se znovu uplatňují strojní a nastavované parametry
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Systémové proměnné

Čtení se zastavením předběžného zpracování

Prostřednictvím následujících systémových proměnných je možno ve výrobním programu a v synchronních akcích načítat momentálně platné tolerance:

- **\$AC_CTOL**
Kanálová tolerance kontury, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.
Jestliže žádná tolerance kontury není v platnosti, v proměnné \$AC_CTOL se nachází odmocnina ze součtu druhých mocnin tolerancí geometrických os.
- **\$AC_OTOL**
Kanálová tolerance orientace, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.
Jestliže žádná tolerance orientace není v platnosti, v proměnné \$AC_OTOL se v průběhu aktivní transformace orientace nachází odmocnina ze součtu druhých mocnin tolerancí orientačních os, jinak je tam hodnota "-1".
- **\$AA_ATOL[<osa>]**
Osová tolerance kontury, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.
Pokud je aktivní tolerance kontury, obsahuje proměnná \$AA_ATOL[<geometrická osa>] toleranci kontury dělenou odmocninou z počtu geometrických os.
Pokud je aktivní tolerance orientace a pokud je aktivní nějaká transformace orientace, obsahuje proměnná \$AA_ATOL[<orientační osa>] toleranci orientace dělenou odmocninou z počtu orientačních os.

Poznámka

Jestliže nebyly naprogramovány žádné hodnoty tolerance, potom proměnné \$A... nejsou dostatečně diferencované, aby bylo možné rozlišit tolerance jednotlivých funkcí.

Takové případy se mohou vyskytnout, jestliže jsou pomocí strojních a nastavovaných parametrů nastaveny odlišné tolerance pro funkce kompresoru, přechodová zaoblení a vyhlazování orientace. Systémové proměnné potom obsahují největší hodnotu, jaká se vyskytuje u momentálně aktivních funkcí. Jestliže je např. aktivní funkce kompresoru s tolerancí orientace 0,1 a vyhlazování orientace pomocí funkce ORISON s hodnotou 1°, poskytuje proměnná \$AC_OTOL hodnotu "1". Když se vyhlazování orientace vypne, poskytuje proměnná je možné \$AC_OTOL hodnotu "0.1".

Načítání bez zastavení předběžného zpracování

Prostřednictvím následujících systémových proměnných je možno ve výrobním programu načítat momentálně platné tolerance:

- **\$P_CTOL**
Momentálně platná kanálová tolerance kontury.
- **\$P_OTOL**
Momentálně platná kanálová tolerance orientace.
- **\$PA_ATOL**
Momentálně platná osová tolerance kontury.

Okrajové podmínky

Tolerance naprogramované pomocí příkazů CTOL, OTOL a ATOL platí také pro funkce, které jsou na těchto tolerancích nepřímo závislé:

- Omezení chyby tětiny při výpočtu požadované hodnoty
- Základní funkce režimu pro frézování ploch volných tvarů

Následující funkce pro vyhlazení průběhu **nejsou** naprogramováním příkazů CTOL, OTOL a ATOL ovlivňovány:

- Vyhlazení průběhu orientace pomocí příkazu OSD
Příkaz OSD nepoužívá žádnou toleranci, nýbrž vzdálenost k přechodu na další blok.
- Přechodová zaoblení u G644
Příkaz G644 neslouží k obrábění, ale k optimalizaci výměn nástroje a jiných pohybů, při kterých neprobíhá obrábění.
- Přechodová zaoblení u G645
Příkaz G645 se chová téměř vždy jako příkaz G642, a tudíž naprogramované tolerance používá. Pouze na hranicích bloků s tangenciálně spojitým napojením se skokovým zakřivením, např. na tangenciálním přechodu kruh-přímka, se používá hodnota tolerance ze strojního parametru MD33120 \$MA_PATH_TRANS_POS_TOL. Dráha přechodového zaoblení se na těchto místech může nacházet téměř až na vnější straně naprogramované kontury, kde je u moha aplikací tolerance menší. Kromě toho zpravidla postačuje malá, napevno nastavená tolerance pro kompenzaci změn zakřivení, s nimiž si programátor NC systému pak už nemusí dělat žádné starosti.

12.12 Chování při přechodu na další blok, když je aktivní vazba (CPBC)

Pomocí příkazu CPBC se zadává kritérium pro přechod na další blok, které musí být splněno, aby se ve výrobním programu s aktivní vazbou uskutečnil přechod na další blok.

Syntaxe

CPBC[<vlečná osa >] = <Kritérium>

Význam

CPBC:	Kritérium pro přechod na další blok, když je aktivní vazba	
<vlečná osa>:	Identifikátor vlečné osy	
<Kritérium>:	Kritérium přechodu na další blok	
	Typ:	STRING
	Hodnota	Význam: Přechod na další blok se uskuteční
	"NOC"	Nezávisle na stavu vazby
	"IPOSTOP"	Při synchronním zpracování na základě žádané hodnoty
	"COARSE"	Při synchronním zpracování na základě "hrubé" skutečné hodnoty
	"FINE"	Při synchronním zpracování na základě "jemné" skutečné hodnoty

Příklad

Programový kód

```
; Na další blok se přechází, když:
; - Vazba na vlečnou osu X2 == aktivní
; - Synchronní zpracování na základě žádané hodnoty ==
aktivní
CPBC[X2]="IPOSTOP"
```


Spojení os vazbou

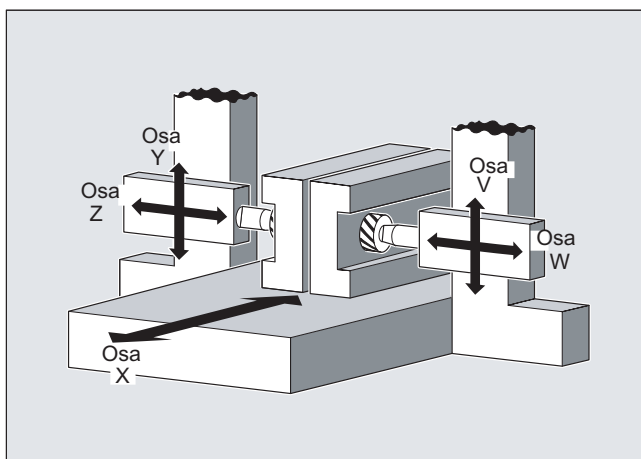
13.1 Vlečení (TRAILON, TRAILOF)

Při pohybech definované řídící osy se pohybují k ní přiřazené vlečené osy (= závislé osy) po drahách, které jsou odvozeny od pohybů řídící osy, přičemž je zohledňován ještě faktor vazby.

Řídící osa a vlečné osy spolu dohromady tvoří vlečné spojení os.

Oblasti použití

- Ovládání pohybu osy pomocí simulované osy. Řídící osa je simulovanou osou a vlečná osa je reálnou osou. Tímto způsobem je možné ovládat pohyb reálné osy, přičemž je zohledněn faktor vazby.
- Obrábění na dvou stranách se 2 vlečnými spojeními os:
 1. řídící osa Y, vlečná osa V
 2. řídící osa Z, vlečná osa W



Syntaxe

```
TRAILON(<vlečná osa>,<řídící osa>,<faktor vazby>)
TRAILOF(<vlečná osa>,<řídící osa>,<řídící osa 2>)
TRAILOF(<vlečná osa>)
```

Význam

TRAILON:	Příkaz pro aktivování a definici vlečného spojení os.	
	Platnost:	modální
<vlečná osa>:	Parametr 1: Identifikátor vlečné osy Upozornění: Vlečná osa může být také řídící osou pro další vlečené osy. Tímto způsobem je možné vytvářet různá další vlečná spojení os.	
<řídící osa>:	Parametr 2: Identifikátor řídící osy	

<faktor vazby>:	Parametr 3: Faktor vazby	
	Faktor vazby udává požadovaný poměr mezi dráhou vlečné osy a dráhou řídící osy:	
	<Faktor vazby> = Dráha vlečné osy / Dráha řídící osy	
	Typ:	REAL
TRAILOF:	Předdefinované nastavení:	1
	Zadání záporné hodnoty má za následek, že se vlečná osa bude pohybovat v opačném směru než osa řídící.	
	Jestliže faktor vazby není při programování uveden, automaticky potom platí faktor vazby 1.	
TRAILOF:	Příkaz pro deaktivování vlečného spojení os	
	Platnost:	modální
	Příkaz TRAILOF se 2 parametry vypíná pouze vazbu k uvedené řídící ose: TRAILOF(<vlečná osa>,<řídící osa>)	
	Jestliže je vlečná osa řízena 2 řídícími osami, pro zrušení obou vazeb je možné vyvolat příkaz TRAILOF se 3 parametry: TRAILOF(<vlečná osa>,<řídící osa>,<řídící osa 2>)	

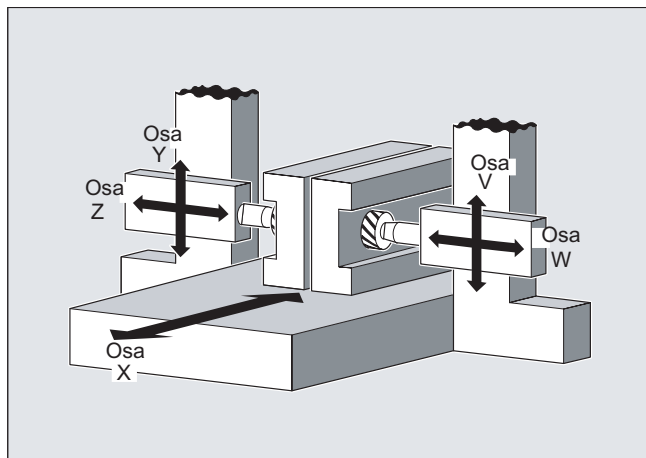
Poznámka

Vlečení se vždy uskutečňuje v základním souřadném systému (BCS).

Počet vlečných spojení, která mohou být aktivována současně, je omezen pouze kombinačními možnostmi os, které jsou na stroji k dispozici.

Příklad:

Obrobek má být opracován na obou stranách s konfigurací os uvedenou na obrázku. Za tím účelem jsou vytvořena 2 vlečná spojení os.



Programový kód	Komentář
...	
N100 TRAILON(V,Y)	; aktivování 1. vlečného spojení os
N110 TRAILON(W,Z,-1)	; aktivování 2. vlečného spojení os Faktor vazby je záporný: Vlečená osa se pohybuje vždy v opačném směru než osa řídící.
N120 G0 Z10	; přísuv osy Z a osy W v opačných směrech os
N130 G0 Y20	; přísuv osy Z a osy V ve stejném směru
...	
N200 G1 Y22 V25 F200	; superpozice závislého a nezávislého pohybu vlečné osy V
...	
TRAILOF(V,Y)	; deaktivování 1. vlečného spojení os
TRAILOF(W,Z)	; deaktivování 2. vlečného spojení os

Další informace

Typy os

Vlečné spojení os se může skládat z libovolné kombinace lineárních a kruhových os. Jako řídící osa přitom může být definována také simulovaná osa.

Vlečené osy

Vlečné ose je možné přiřadit maximálně 2 řídící osy současně. Přiřazení se provádí v různých vlečných spojeních.

Pro programování vlečné osy je možné využívat všech pohybových příkazů, které jsou k dispozici (G0, G1, G2, G3, ...). Pohyby vlečné osy se skládají z nezávisle definovaných drah, které se skládají s drahami odvozenými od řídících os, jež jsou upraveny pomocí faktoru vazby.

Omezení dynamiky

Omezení dynamiky závisí na druhu aktivování vlečné vazby:

- Aktivování ve výrobním programu
Pokud je aktivování uskutečněno ve výrobním programu a pokud jsou všechny řídicí osy programovými osami v aktivovaném kanálu, potom se při pohybech řídicích os zohledňuje dynamika vlečných os tak, aby žádná z vlečných os nebyla přetížena.
Jestliže je aktivování provedeno ve výrobním programu s řídicími osami, které nejsou aktivní jako programové osy v aktivovaném kanálu (\$AA_TYP ≠ 1), potom se při pohybech řídicích os na dynamiku vlečné osy nebere zřetel. V důsledku toho může u vlečných os s menší dynamikou, než jaká je pro vazbu potřebná, dojít k přetížení.
- Aktivování v synchronních akcích
Pokud se aktivování uskuteční v synchronní akci, nebude se při pohybech řídicích os dynamika vlečných os zohledňovat. V důsledku toho může u vlečných os s menší dynamikou, než jaká je pro vazbu potřebná, dojít k přetížení.

**POZOR****Přetížení osy**

Pokud je vlečné spojení os aktivováno

- v synchronních akcích
- ve výrobním programu s řídicími osami, které nejsou programovými osami v kanálu vlečné osy,

potom uživatel/výrobce stroje v plné míře odpovídají za to, že budou přijata vhodná opatření, aby v důsledku posuvových pohybů řídicí osy nemohlo dojít k přetížení vlečných os.

Stav vazby

Stav vazby dané osy může být ve výrobním programu zjištěn pomocí následujících systémových proměnných:

\$AA_COUP_ACT[<osa>]

Hodnota	Význam
0	Žádná vazba není aktivní
8	Je aktivní vlečení

Výpis zbytkové dráhy vlečené osy u kruhových os typu modulo

Pokud jsou řídicí a vlečná osa kruhovými osami typu modulo, sečítají se posuvy při pohybech řídicí osy $n \cdot 360^\circ$, kde $n = 1, 2, 3, \dots$, ve vypisovaném údaji zbytkové dráhy vlečené osy až do deaktivování vazby.

Příklad: Úsek programu s příkazem TRAILON, řídicí osou B a vlečnou osou C

Programový kód	Komentář
TRAILON (C,B,1)	; Aktivování vazby
G0 B0	; Výchozí pozice
	; Údaj zbytkové dráhy na začátku bloku:
G91 B360	; B=360, C=360

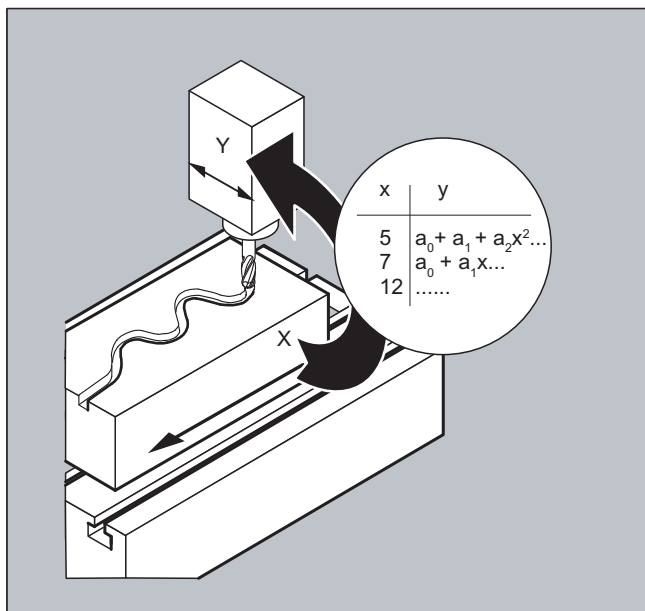
Programový kód	Komentář
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

13.2 Tabulky křivek (CTAB)

Pomocí tabulek křivek je možné naprogramovat polohové a rychlostní vztahy mezi dvěma osami (řídící a vlečná osa). Definice tabulek křivek se uskutečňuje ve výrobním programu.

Aplikace

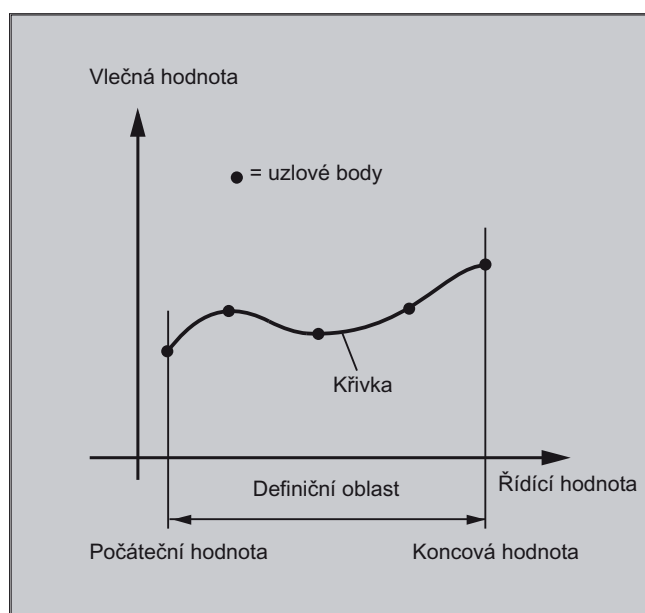
Tabulky křivek nahrazují mechanické křivkové vačky. Tabulky křivek přitom tvoří základ pro vazbu řídící hodnotou mezi osami, neboť zajišťují funkční vztah mezi řídící a závislou hodnotou. Při odpovídajícím naprogramování řídící systém vypočítává ze vzájemně přiřazených pozic řídící a závislé osy polynom, který odpovídá křivkové vačce.



13.2.1 Definice tabulek křivek (CTABDEF, CATBEND)

Tabulka křivky představuje výrobní program nebo úsek výrobního programu, který je na začátku uveden příkazem CTABDEF a ukončen závěrečným příkazem CATBEND.

V rámci tohoto úseku výrobního programu jsou prostřednictvím pohybových příkazů přiřazeny jednotlivým pozicím řídící osy polohy závislé osy, které se pak používají jako uzlové body pro výpočet aproximační křivky ve formě polynomu maximálně 5. stupně.



Předpoklady

Pro definici tabulky křivky musí být prostřednictvím nastavení odpovídajících konfiguračních strojních parametrů vyhrazen patřičný paměťový prostor (--> výrobce stroje!).

Syntaxe

```
CTABDEF(<vlečná osa>,<řídící osa>,<n>,<periodicita>[,<paměťové  
místo>])  
...  
CTABEND
```

Význam

CTABDEF ():	Začátek definice tabulky křivky	
CTABEND:	Konec definice tabulky křivky	
<vlečná osa>:	Osa, jejíž pohyb se má vypočítávat prostřednictvím tabulky křivky.	
<řídící osa>:	Osa, která poskytuje řídící hodnotu pro výpočet pohybu vlečné osy.	
<n>:	Číslo (ID) tabulky křivky Číslo tabulky křivky je jednoznačné a nezávisí na umístění v paměti. Ani ve statické, ani v dynamické paměti NC systému se nemohou nacházet tabulky se stejnými čísly.	
<periodicita>:	Periodicita tabulky	
	0	Tabulka není periodická (bude zpracována jen jednou, a to i v případě kruhových os)
	1	Tabulka je vzhledem k řídící ose periodická.
	2	Tabulka je periodická vzhledem k řídící ose i vlečné ose.

<paměťové místo>:	Údaj místa v paměti (nepovinné)	
	"SRAM"	Tabulka křivky se ukládá ve statické paměti NC systému.
	"DRAM"	Tabulka křivky se ukládá v dynamické paměti NC systému.
Upozornění: Pokud pro tento parametr není naprogramována žádná hodnota, potom se použije standardní místo v paměti nastavené parametrem MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE.		

Poznámka**Přepisování**

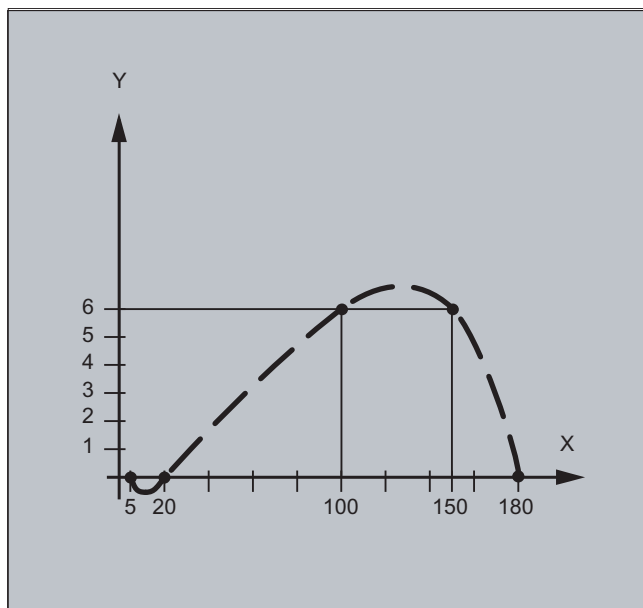
Tabulka křivky se přepíše, jakmile je při nové definici tabulky použito její číslo (<n>) (výjimka: Tabulka křivky je aktivní v rámci vazby mezi osami nebo je blokována příkazem CTABLOCK).
Při přepisování tabulky se nevypisuje žádná odpovídající výstraha!

Příklady**Příklad 1: Úsek programu jako definice tabulky křivky**

Úsek programu má být použit nezměněn pro definici tabulky křivky. Příkaz pro zastavení předběžného zpracování STOPRE, který se v něm vyskytuje, může zůstat na svém místě a okamžitě se znovu aktivuje, jakmile se úsek programu přestane používat pro definici tabulky a jakmile je pomocí příkazů CTABDEF a CTABEND odstraněn.

Programový kód	Komentář
...	
CTABDEF(Y,X,1,1)	; definice tabulky křivky
...	
IF NOT (\$P_CTABDEF)	
STOPRE	
ENDIF	
...	
CTABEND	

Příklad 2: Definice neperiodické tabulky křivky



Programový kód	Komentář
N100 CTABDEF(Y,X,3,0)	; začátek definice neperiodické tabulky křivky s číslem 3
N110 X0 Y0	; 1. pohybový příkaz, definuje počáteční hodnotu a 1. uzlové místo: Řídící hodnota: 0, vlečná hodnota: 0
N120 X20 Y0	; 2. uzlové místo: Řídící hodnota: 0...20, vlečná hodnota: Počáteční hodnota... 0
N130 X100 Y6	; 3. uzlové místo: Řídící hodnota: 20...100, vlečná hodnota: 0...6
N140 X150 Y6	; 4. uzlové místo: Řídící hodnota: 100...150, vlečná hodnota: 6...6
N150 X180 Y0	; 5. uzlové místo: Řídící hodnota: 150...180, vlečná hodnota: 6...0
N200 CTABEND	; konec definice Tabulka křivky se ve své interní reprezentaci převede na polynom maximálně 5. stupně. Výpočet křivkového průběhu se zadanými uzlovými body závisí na zvoleném modálním druhu interpolace (kruhová, lineární, splinová interpolace). Opět se obnoví stav výrobního programu před zahájením definice.

Příklad 3: Definice periodické tabulky křivky

Definice periodické tabulky křivky s číslem 2, rozmezí řídicí hodnoty od 0 do 360, pohyb vlečné osy od 0 do 45 a zpátky do 0.

Programový kód	Komentář
N10 DEF REAL DEPPOS	

Programový kód	Komentář
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
;Text křivky pomocí vazby mezi Y a X:	
N120 G1 F1000 X0	
N130 LEADON(Y,X,2)	
N140 X360	
N150 X0	
N160 LEADOF(Y,X)	
N170 DEPPPOS=CTAB(75.0,2,GRADIENT)	; načtení tabulkové funkce s řídicí hodnotou 75,0
N180 G0 X75 Y=DEPPPOS	; polohování řídicí a vlečné osy
;Po aktivování vazby už není žádná synchronizace vlečné osy zapotřebí.	
N190 LEADON(Y,X,2)	
N200 G1 X110 F1000	
N210 LEADOF(Y,X)	
N220 M30	

Další informace

Počáteční a koncová hodnota tabulky křivky

Jako počáteční hodnota pro začátek definiční oblasti tabulky křivky platí první údaj k sobě patřících poloh os (první pohybový příkaz) uvnitř definice tabulky křivky. Koncová hodnota definiční oblasti tabulky křivky je analogicky určena prostřednictvím posledního pohybového příkazu.

Rozsah jazykových prostředků, které jsou k dispozici

V rámci definice tabulky křivky je k dispozici celý rozsah NC jazyka.

Poznámka

Následující příkazy jsou v definicích tabulek křivek nepřipustné:

- Zastavení předběžného zpracování
- Skokové pohyby řídící osy (např. při změně transformací)
- Příkaz pohybu pro samotnou závislou osu
- Změna směru pohybu řídící osy na opačný, tzn. poloha řídící osy musí být vždy jednoznačná.
- Příkazy CTABDEF a CTABEND na různých programových úrovních.

Platnost modálních příkazů

Veškeré příkazy s modální platností, které se vyskytují v rámci definice tabulky křivky, se stávají po ukončení definice tabulky neplatnými. Výrobní program, ve kterém se definice tabulky nachází, se takto nachází před a po definici tabulky ve stejném stavu.

Přiřazování R-Parametrů

Přiřazení R-parametrů v rámci definice tabulky se po příkazu CTABEND resetuje.

Příklad:

Programový kód	Komentář
...	
R10=5 R11=20	; R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	; R10=25
X=R10	
CTABEND	
...	; R10=5

Aktivování funkcí ASPLINE, BSPLINE, CSPLINE

Jestliže je v rámci definice tabulky křivky pomocí příkazů CTABDEF ... CTABEND aktivován některý z příkazů ASPLINE, BSPLINE nebo CSPLINE, potom by měl být před tímto příkazem splinu naprogramován minimálně jeden počáteční bod. Okamžitému aktivování po příkazu CTABDEF byste se měli vyhnout, protože jinak by spline závisel na momentální poloze osy před definicí tabulky křivky.

Příklad:

Programový kód
...
CTABDEF(Y,X,1,0)
X0 Y0

Programový kód

```
ASPLINE  
X=5 Y=10  
X10 Y40  
...  
CTABEND
```

Opakované použití tabulky křivky

Funkční závislost mezi řídicí a vlečnou osou, která byla vypočítána pomocí tabulky křivky, zůstává pod vybraným číslem tabulky i po skončení výrobního programu a po vypnutí a zapnutí systému (Power off), jestliže je ale tabulka uložena ve statické paměti NC systému.

Tabulka, která byla uložena v dynamické paměti (DRAM), se při vypnutí a zapnutí systému (power-on) vymaže a v případě potřeby musí být sestavena ještě jednou.

Jednou sestavená tabulka křivky může být využita libovolnou kombinací řídicí a vlečné osy a je nezávislá na tom, které osy byly pro její sestavení použity.

Přepisování tabulek křivek

Tabulka křivky se přepíše, jakmile je při definici tabulky znovu použito její číslo.

Výjimka: Tabulka křivky je aktivní v rámci vazby mezi osami nebo je blokována příkazem CTABLOCK.

Poznámka

Při přepisování tabulky křivky se nevypisuje žádná odpovídající výstraha!

Je definice tabulky křivky aktivní?

Prostřednictvím systémové proměnné \$P_CTABDEF je možné z výrobního programu kdykoli zjistit, zda je definice tabulky křivky aktivní.

Ukončení definice tabulky křivky

Úsek výrobního programu je po vymezení definice tabulky křivky, které označují její začátek a konec, opět použitelný jako reálný výrobní program.

Načítání tabulek křivek pomocí funkce "Zpracování z externího zdroje"

Při zpracovávání tabulek křivek z externího zdroje musí být velikost používané vyrovnávací paměti (DRAM) nastavena pomocí parametru MD18360 \$MN_MM_EXT_PROG_BUFFER_SIZE tak, aby bylo možné do této vyrovnávací paměti uložit celou definici tabulky naráz. Zpracovávání výrobního programu se jinak přeruší a aktivuje se alarm.

Skokové změny vlečné osy

V závislosti na nastavení strojního parametru

MD20900 \$MC_CTAB_ENABLE_NO_LEADMOTION

mohou být skokové změny vlečné osy v případě chybějícího pohybu řídicí osy tolerovány.

13.2.2 Kontrola existence tabulky křivky (CTABEXISTS)

Pomocí příkazu CTABEXISTS může být zkontrolováno, zda se tabulka s určitým číslem křivky vyskytuje v paměti NC systému.

Syntaxe

CTABEXISTS (<n>)

Význam

CTABEXISTS:	Zkontroluje se, zda je tabulka křivky s číslem <n> k dispozici ve statické nebo dynamické paměti NC systému.	
	0	Tabulka neexistuje
	1	Tabulka existuje
<n>:	Číslo (ID) tabulky křivky	

13.2.3 Mazání tabulek křivek (CTABDEL)

Pomocí příkazu CTABDEL je možné mazat tabulky křivek.

Poznámka

Tabulky křivky, které jsou aktivní ve vazbě mezi osami, nemohou být vymazány.

Syntaxe

CTABDEL (<n>)
CTABDEL (<n>, <m>)
CTABDEL (<n>, <m>, <paměťové místo>)
CTABDEL ()
CTABDEL (, , <paměťové místo>)

Význam

CTABDEL:	Příkaz pro vymazání tabulky křivky
<n>:	Číslo (ID) tabulky křivky, která má být vymazána Při mazání intervalu tabulek křivek pomocí příkazu CTABDEL (<n>, <m>) se pomocí parametru <n> zadává číslo první tabulky křivky tohoto intervalu.
<m>:	Při mazání intervalu tabulek křivek pomocí příkazu CTABDEL (<n>, <m>) se pomocí parametru <m> zadává číslo poslední tabulky křivky tohoto intervalu. Parametr <m> musí být větší než <n>!

<paměťové místo>:	Údaj místa v paměti (nepovinné)	
	Při mazání bez zadání paměťového místa jsou uvedené tabulky křivek vymazány ze statické a dynamické paměti NC systému.	
	Při mazání se zadáním paměťového místa jsou vymazány jen ty uvedené tabulky křivek, které se nacházejí v zadané paměti. Zbývající tabulky zůstanou nedotčeny.	
	"SRAM"	Mazání ve statické paměti NC systému
	"DRAM"	Mazání v dynamické paměti NC systému

Pokud je naprogramován příkaz CTABDEL bez udání tabulky křivky, která má být vymazána, potom se vymažou **všechny** tabulky křivek, příp. všechny tabulky křivek z uvedené paměti.

CTABDEL () :	Vymažou se všechny tabulky křivek ve statické a dynamické paměti NC systému.
CTABDEL (, , "SRAM") :	Vymažou se všechny tabulky křivek ve statické paměti NC systému.
CTABDEL (, , "DRAM") :	Vymažou se všechny tabulky křivek v dynamické paměti NC systému.

Poznámka

Pokud je v případě mazání většího počtu křivek pomocí příkazů CTABDEL (<n> , <m>) nebo CTABDEL () alespoň jedna z mazaných tabulek křivek součástí aktivní vazby, potom se příkaz mazání neuskuteční, tzn. nevymaže se **žádná** ze zadaných tabulek křivek.

13.2.4 Blokování tabulek křivek proti mazání a přepisování (CTABLOCK, CTABUNLOCK)

Tabulky křivek mohou být aktivováním blokování chráněny proti nechtěnému vymazání a přepsání. Aktivované blokování může být kdykoli opět zrušeno.

Syntaxe

Aktivování blokování:

```
CTABLOCK (<n>)
CTABLOCK (<n> , <m>)
CTABLOCK (<n> , <m> , <paměťové místo>)
CTABLOCK ( )
CTABLOCK ( , , <paměťové místo>)
```

Deaktivování blokování:

```
CTABUNLOCK (<n>)
CTABUNLOCK (<n> , <m>)
CTABUNLOCK (<n> , <m> , <paměťové místo>)
CTABUNLOCK ( )
CTABUNLOCK ( , , <paměťové místo>)
```

Význam

CTABLOCK:	Příkaz pro aktivování blokování proti vymazání/přepsání
CTABUNLOCK:	Příkaz pro deaktivování blokování proti vymazání/přepsání Příkazem CTABUNLOCK se tabulky křivek zablokované příkazem CTABLOCK opět odblokují. Tabulky křivky, které se uplatňují v aktivní vazbě, zůstanou i nadále zablokovány a nemohou být vymazány. Blokování příkazem CTABLOCK je odstraněno, jakmile pomine ochrana vyvolávaná aktivní vazbou v důsledku toho, že je vazba deaktivována. Potom může být tato tabulka vymazána. Opětovné vyvolání příkazu CTABUNLOCK není zapotřebí.
<n>:	Číslo (ID) tabulky křivky, která má být blokována/odblokována Při aktivování/deaktivování blokování intervalu tabulek křivek pomocí příkazů CTABLOCK (<n>, <m>) /CTABUNLOCK (<n>, <m>) se pomocí parametru <n> zadává číslo první tabulky křivky tohoto intervalu.
<m>:	Při aktivování/deaktivování blokování intervalu tabulek křivek pomocí příkazů CTABLOCK (<n>, <m>) /CTABUNLOCK (<n>, <m>) se pomocí parametru <m> zadává číslo poslední tabulky křivky tohoto intervalu. Parametr <m> musí být větší než <n>!
<paměťové místo>:	Údaj místa v paměti (nepovinné) Při aktivování/deaktivování blokování bez zadání paměťového místa jsou uvedené tabulky křivek blokovány/odblokovány ve statické a dynamické paměti NC systému. Při aktivování/deaktivování blokování se zadáním paměťového místa jsou blokovány/odblokovány jen ty uvedené tabulky křivek, které se nacházejí v zadané paměti. Zbývajících se blokování/odblokování netýká.
	"SRAM" Aktivování/deaktivování blokování ve statické paměti NC systému
	"DRAM" Aktivování/deaktivování blokování v dynamické paměti NC systému

Pokud je naprogramován příkaz CTABLOCK/CTABUNLOCK bez udání tabulky křivky, jejíž blokování má být aktivováno/deaktivováno, potom se aktivování/deaktivování blokování týká **všech** tabulek křivek, příp. všech tabulek křivek z uvedené paměti.

CTABLOCK () :	Zablokují se všechny tabulky křivek ve statické a dynamické paměti NC systému.
CTABLOCK (, , "SRAM") :	Zablokují se všechny tabulky křivek ve statické paměti NC systému.
CTABLOCK (, , "DRAM") :	Zablokují se všechny tabulky křivek v dynamické paměti NC systému.
CTABUNLOCK () :	Odblokují se všechny tabulky křivek ve statické a dynamické paměti NC systému.
CTABUNLOCK (, , "SRAM") :	Odblokují se všechny tabulky křivek ve statické paměti NC systému.
CTABUNLOCK (, , "DRAM") :	Odblokují se všechny tabulky křivek v dynamické paměti NC systému.

13.2.5 Tabulky křivek: Zjišťování vlastností tabulek (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Pomocí těchto příkazů mohou být zjišťovány důležité vlastnosti tabulek křivek (číslo tabulka, stav blokování, místo uložení, periodičita).

Syntaxe

```
CTABID(<p>)
CTABID(<p>, <paměťové místo>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
CTABPERIOD(<n>)
```

Význam

CTABID:	Výsledkem je číslo tabulky, která je v zadané paměti uložena jako <p>-tá tabulka křivky. Příklad: Výsledkem příkazu CTABID(1, "SRAM") bude číslo první tabulky křivky ve statické paměti NC systému. První tabulka křivky přitom odpovídá tabulce křivky s nejvyšším číslem tabulky. Upozornění: Jestliže se mezi dvěma po sobě následujícími voláními funkce CTABID posloupnost tabulek křivek v paměti změní, např. v důsledku vymazání tabulek křivek pomocí příkazu CTABDEL, může se stát, že výsledkem příkazu CTABID(<p>, ...) se stejným číslem <p> bude jiná tabulka křivky než předtím.	
CTABISLOCK:	Výsledkem je stav blokování tabulky křivky s číslem <n>.	
	0	Tabulka není blokována
	1	Tabulka je blokována příkazem CTABLOCK
	2	Tabulka je blokována díky aktivní vazbě
	3	Tabulka je blokována příkazem CTABLOCK a díky aktivní vazbě
	-1	Tabulka neexistuje
CTABMEMTYP:	Výsledkem je místo uložení tabulky křivky s číslem <n>.	
	0	Tabulka ve statické paměti NC systému
	1	Tabulka v dynamické paměti NC systému
	-1	Tabulka neexistuje
CTABPERIOD:	Výsledkem je periodičita tabulky křivky s číslem <n>.	
	0	Tabulka není periodická
	1	Tabulka je vzhledem k řídicí ose periodická.
	2	Tabulka je periodická vzhledem k řídicí ose i vlečné ose.
	-1	Tabulka neexistuje
<p>:	Číslo položky v paměti	
<n>:	Číslo (ID) tabulky křivky	

<paměťové místo>:	Údaj místa v paměti (nepovinné)	
	"SRAM"	statická paměť NC systému
	"DRAM"	dynamická paměť NC systému
Upozornění: Pokud pro tento parametr není naprogramována žádná hodnota, potom se použije standardní místo v paměti nastavené parametrem MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE.		

13.2.6 Čtení hodnot z tabulky křivky (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Ve výrobním programu mohou být čteny následující hodnoty z tabulky křivky:

- Hodnoty vlečné a řídící osy na začátku a na konci tabulky křivky
- Hodnoty vlečné osy na začátku a na konci úseku křivky
- Hodnota vlečné osy pro určitou hodnotu řídící osy
- Hodnota řídící osy pro určitou hodnotu vlečné osy
- Minimální a maximální hodnota vlečné osy
 - v celkovém definičním rozsahu tabulky křivky
nebo
 - v definovaném intervalu tabulky křivky

Syntaxe

```
CTABTSV(<n>,<gradient>[,<vlečná osa>])
CTABTEV(<n>,<gradient>[,<vlečná osa>])
CTABTSP(<n>,<gradient>[,<řídící osa>])
CTABTEP(<n>,<gradient>[,<řídící osa>])
CTABSSV(<řídící hodnota>,<n>,<gradient>[,<vlečná osa>])
CTABSEV(<řídící hodnota>,<n>,<gradient>[,<vlečná osa>])
CTAB(<řídící hodnota>,<n>,<gradient>[,<vlečná osa>,<řídící osa>])
CTABINV(<vlečná hodnota>,<hodnota
přiblížení>,<n>,<gradient>[,<vlečná osa>,<řídící osa>])
CTABTMIN(<n>[,<vlečná osa>])
CTABTMAX(<n>[,<vlečná osa>])
CTABTMIN(<n>,<a>,<b>[,<vlečná osa>,<řídící osa>])
CTABTMAX(<n>,<a>,<b>[,<vlečná osa>,<řídící osa>])
```

Význam

CTABTSV:	Načtení hodnoty vlečné osy na začátku tabulky křivky s číslem <n>.
CTABTEV:	Načtení hodnoty vlečné osy na konci tabulky křivky s číslem <n>.
CTABTSP:	Načtení hodnoty řídící osy na začátku tabulky křivky s číslem <n>.
CTABTEP:	Načtení hodnoty řídící osy na konci tabulky křivky s číslem <n>.

CTABSSV:	Načtení hodnoty vlečné osy na začátku úseku křivky, který patří k zadané hodnotě řídicí osy (<řídicí hodnota>)
CTABSEV:	Načtení hodnoty vlečné osy na konci úseku křivky, který patří k zadané hodnotě řídicí osy (<řídicí hodnota>)
CTAB:	Načtení hodnoty vlečné osy k zadané hodnotě řídicí osy (<řídicí hodnota>)
CTABINV:	Načtení hodnoty řídicí osy k zadané hodnotě vlečné osy (<vlečná hodnota>)
CTABTMIN:	Stanovení minimální hodnoty vlečné osy: - v celkovém definičním rozsahu tabulky křivky nebo - v definovaném intervalu <a> ...
CTABTMAX:	Stanovení maximální hodnoty vlečné osy: - v celkovém definičním rozsahu tabulky křivky nebo - v definovaném intervalu <a> ...
<n>:	Číslo (ID) tabulky křivky
<gradient>:	V parametru <gradient> systém poskytuje údaj o stoupání funkce popísané tabulkou křivky v bodě, který je předmětem zájmu
<vlečná osa>:	Osa, jejíž pohyb se má vypočítávat prostřednictvím tabulky křivky (nepovinné)
<řídicí osa>:	Osa, která poskytuje řídicí hodnotu pro výpočet pohybu vlečné osy (nepovinné)
<vlečná hodnota>:	Hodnota vlečné osy pro načtení k ní náležící hodnoty řídicí osy pomocí příkazu CTABINV
<řídicí hodnota>:	Hodnota řídicí osy - pro načtení k ní náležící hodnoty vlečné osy pomocí příkazu CTAB nebo - pro volbu úseku křivky pomocí příkazu CTABSSV/CTABSEV
<hodnota přiblížení>:	Přiřazení hodnoty řídicí osy hodnotě vlečné osy u příkazu CTABINV nemusí být vždy jednoznačné. Příkaz CTABINV potřebuje proto jako parametr hodnotu, která se blíží očekávané hodnotě řídicí osy.
<a>:	Spodní mezní hodnota intervalu řídicích hodnot u příkazů CTABTMIN/CTABTMAX
:	Horní mezní hodnota intervalu řídicích hodnot u příkazů CTABTMIN/CTABTMAX
	Upozornění: Interval řídicích hodnot <a> ... musí ležet v rámci definičního rozsahu tabulky křivky.

Příklady

Příklad 1:

Stanovení hodnot vlečné osy a řídící osy na začátku a na konci tabulky křivky, jakož i maximální a minimální hodnoty vlečné osy v celém definičním rozsahu tabulky křivky.

Programový kód	Komentář
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
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 pozice 1. úseku tabulky
N120 X30 Y40	; koncová pozice 1. úseku tabulky = počáteční pozice 2. úseku tabulky
N130 X60 Y5	; Koncová pozice 2. úseku tabulky = ...
N140 X70 Y30	
N150 X80 Y20	
N160 CTABEND	; Konec definice tabulky.
...	
N200 STARTPOS=CTABTSV(1,GRADIENT)	; Hodnota vlečné osy na začátku tabulky křivky = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)	; Hodnota vlečné osy na konci tabulky křivky = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)	; Hodnota řídící osy na začátku tabulky křivky = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)	; Hodnota řídící osy na konci tabulky křivky = 80
N240 MINVAL=CTABTMIN(1)	; Minimální hodnota vlečné osy v bodě Y=5
N250 MAXVAL=CTABTMAX(1)	; Maximální hodnota vlečné osy v bodě Y=40

Příklad 2:

Stanovení hodnoty vlečné osy na začátku a na konci úseku křivky, který patří k hodnotě řídící osy X=30.

Programový kód	Komentář
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; Začátek definice tabulky.
N110 X0 Y0	; Počáteční pozice 1. úseku tabulky

Programový kód	Komentář
N120 X20 Y10	; koncová pozice 1. úseku tabulky = počáteční pozice 2. úseku tabulky
N130 X40 Y40	; Koncová pozice 2. úseku tabulky = ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; Konec definice tabulky.
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; Počáteční pozice Y ve 2. úseku = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	; Koncová pozice Y ve 2. úseku = 40

Další informace

Použití v synchronních akcích

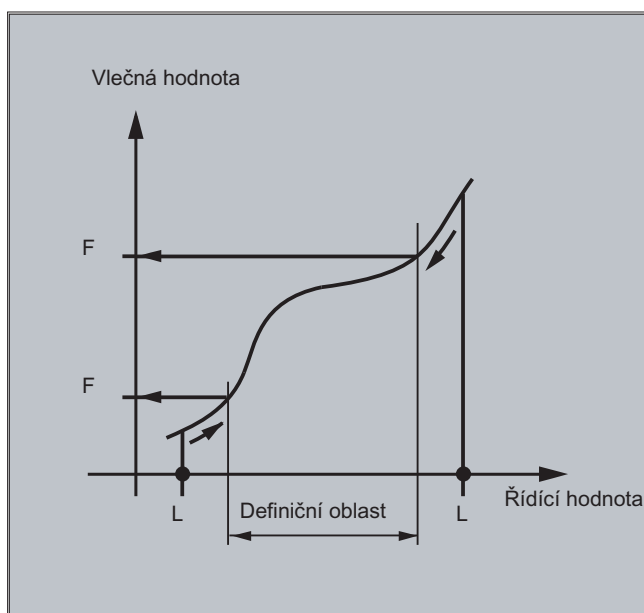
Všechny příkazy pro čtení hodnot z tabulky křivky se mohou používat také v synchronních akcích (viz také kapitola "Pohybové synchronní akce").

Při použití příkazů CTABINV, CTABTMIN a CTABTMAX je zapotřebí mít na paměti, aby:

- v okamžiku zpracování těchto příkazů byl k dispozici dostatečný výkon NC systému nebo
- aby před vyvoláním příkazu byl zjištěn počet úseků v tabulce křivky, aby bylo možné v případě potřeby příslušnou tabulku rozdělit

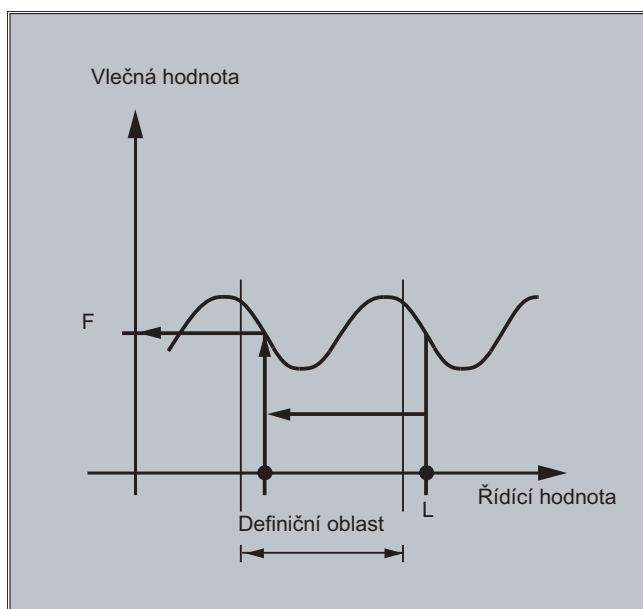
Příkaz CTAB v neperiodických tabulkách křivky

Jestliže hodnota zadaná v parametru <řídící hodnota> leží mimo definiční rozsah, bude jako vlečná hodnota uvedena horní, příp. dolní mezní hodnota:



Příkaz CTAB v periodických tabulkách křivky

Jestliže hodnota zadaná v parametru <řídící hodnota> leží mimo definiční rozsah, bude řídící hodnota upravena funkcí Modulo pro definiční rozsah a potom se zjistí odpovídající vlečná hodnota:



Hodnota přiblížení pro příkaz CTABINV

Příkaz CTABINV potřebuje hodnotu, která se blíží očekávané řídící hodnotě. Výsledkem funkce CTABINV je řídící hodnota, která leží nejbližší hodnotě přiblížení. Hodnotou přiblížení může být např. řídící hodnota z předcházejícího taktu interpolátoru.

Stoupání funkce dané tabulkou křivky

Zadání stoupání pomocí parametru <gradient> umožňuje započítat rychlost řídící nebo vlečné osy na odpovídajícím místě charakteristiky.

Zadání řídící nebo vlečné osy

Nepovinné zadání řídící a/nebo vlečné osy je důležité, jestliže jsou pro řídící a vlečnou osu v konfiguraci nastaveny různé jednotky délky.

CTABSSV, CTABSEV

Příkazy CTABSSV a CTABSEV **nejsou** v následujících případech vhodné k tomu, aby se zjišťovaly informace o naprogramovaných segmentech:

- Jsou naprogramovány kruhy nebo evolventy.
- Jsou aktivní fasety, příp. zaoblení pomocí příkazů CHF/RND.
- Jsou aktivní přechodová zaoblení aktivovaná příkazem G643.
- Pomocí příkazů COMPON/COMP CURV/COMPCAD je aktivní komprese NC bloků.

13.2.7 Tabulky křivek: Kontrola využití systémových zdrojů (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Pomocí těchto příkazů má programátor možnost získat aktuální informace o využití systémových zdrojů pro tabulky křivek, úseky tabulek a polynomy.

Syntaxe

```
CTABNO
CTABNOMEM(<paměťové místo>)
CTABFNO(<paměťové místo>)
CTABSEGID(<n>,<paměťové místo>)
CTABSEG(<paměťové místo>,<druh úseku>)
CTABFSEG(<paměťové místo>,<druh úseku>)
CTABMSEG(<paměťové místo>,<druh úseku>)
CTABPOLID(<n>)
CTABPOL(<paměťové místo>)
CTABFPOL(<paměťové místo>)
CTABMPOL(<paměťové místo>)
```

Význam

CTABNO:	Stanovení celkového počtu definovaných tabulek křivek (ve statické a dynamické paměti NC systému)
CTABNOMEM:	Stanovení počtu definovaných tabulek křivek v zadané paměti <paměťové místo>
CTABFNO:	Stanovení počtu ještě možných tabulek křivek v zadané paměti <paměťové místo>
CTABSEGID:	Stanovení počtu křivkových úseků odpovídajících zadanému <druhu úseku>, které jsou využívány tabulkou křivky s číslem <n>
CTABSEG:	Stanovení počtu používaných úseků křivky odpovídajících zadanému <druhu úseku>, které jsou uloženy v zadané paměti <paměťové místo>
CTABFSEG:	Stanovení počtu ještě možných úseků křivky odpovídajících zadanému <druhu úseku>, které jsou uloženy v zadané paměti <paměťové místo>
CTABMSEG:	Stanovení počtu maximálně možných úseků křivky odpovídajících zadanému <druhu úseku>, které jsou uloženy v zadané paměti <paměťové místo>
CTABPOLID:	Stanovení počtu křivkových polynomů, které jsou využívány tabulkou křivky s číslem <n>
CTABPOL:	Stanovení počtu používaných křivkových polynomů v zadané paměti <paměťové místo>
CTABFPOL:	Stanovení počtu ještě možných křivkových polynomů v zadané paměti <paměťové místo>
CTABMPOL:	Stanovení počtu maximálně možných křivkových polynomů v zadané paměti <paměťové místo>
<n>:	Číslo (ID) tabulky křivky

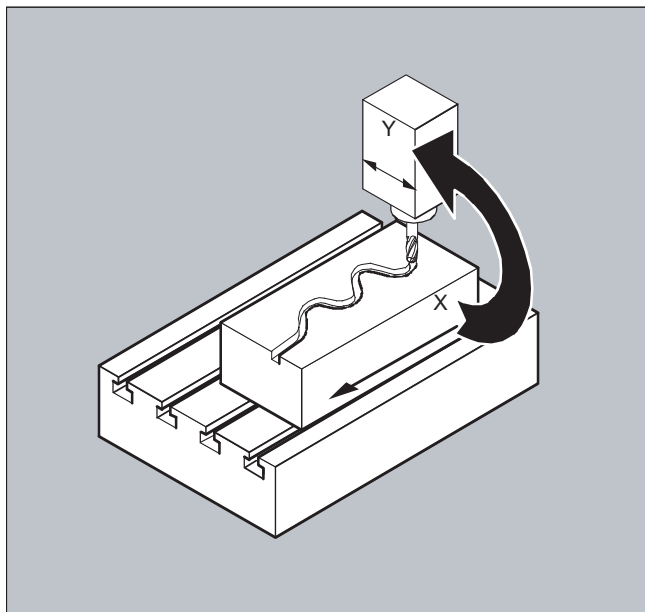
<paměťové místo>:	Údaj místa v paměti (nepovinné)	
	"SRAM"	statická paměť NC systému
	"DRAM"	dynamická paměť NC systému
	Upozornění: Pokud pro tento parametr není naprogramována žádná hodnota, potom se použije standardní místo v paměti nastavené parametrem MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE.	
<druh úseku>:	Zadání druhu úseku (nepovinné)	
	"L"	Lineární úseky
	"P"	Polynomický úsek
	Upozornění: Jestliže pro tento parametr není naprogramována žádná hodnota, bude výsledkem součet lineárních a polynomických úseků.	

13.3 Osová vazba řídicí hodnotou (LEADON, LEADOF)

Poznámka

U systému SINUMERIK 828D není tato funkce k dispozici!

V případě osové vazby řídicí hodnotou se řídicí a vlečná osa pohybují synchronně. Příslušná poloha vlečné osy je přitom prostřednictvím tabulky křivky, příp. z ní vypočítaného polynomu jednoznačně přiřazena poloze - v případě potřeby simulované - řídicí osy.



Řídicí osou je ta osa, která poskytuje vstupní hodnoty pro tabulku křivky. **Vlečnou osou** je nazývána osa, která zaujímá polohy vypočítané prostřednictvím tabulky křivky.

Vazba pomocí skutečné a požadované hodnoty

Jako řídicí hodnoty se mohou používat také výstupní hodnoty pro zjišťování polohy vlečné osy.

- Skutečné hodnoty polohy řídicí osy: Vazba pomocí skutečné hodnoty
- Požadované hodnoty polohy řídicí osy: Vazba pomocí požadované hodnoty

Vazba pomocí řídicí hodnoty platí vždy v základním souřadném systému.

Pokud budete potřebovat informace o tabulkách křivek, viz kapitola "Tabulky křivek".

Syntaxe

```
LEADON(<vlečná osa>,<řídicí osa>,<n>)
LEADOF(<vlečná osa>,<řídicí osa>)
```

nebo zrušení příkazem bez uvedení řídicí osy:

```
LEADOF(<vlečná osa>)
```

Vazba řídicí hodnotou může být aktivována a deaktivována jak z výrobního programu, tak také v průběhu pohybu ze synchronních akcí.

Význam

LEADON:	Aktivování vazby řídicí hodnotou
LEADOF:	Deaktivování vazby řídicí hodnotou
<vlečná osa>:	Vlečná osa
<řídící osa>:	Řídící osa
<n>:	Číslo tabulky křivky
\$SA_LEAD_TYPE:	Přepnutí mezi vazbou pomocí požadované a skutečné hodnoty

Deaktivování vazby řídicí hodnotou, LEADOF

Po deaktivování vazby řídicí hodnotou se vlečná osa znovu stává normální příkazovou osou!

Osová vazba řídicí hodnotou a různé provozní stavy, RESET

V závislosti na nastavení ve strojním parametru se vazby řídicí hodnotou při RESETu vypínají.

Příklad vazby řídicí hodnotou ze synchronní akce

U lisovacího zařízení má být obvyklá mechanická vazba mezi řídicí osou (hřídel razníku) a osami podávacího systému, který se skládá z os podavače a pomocných os, nahrazena systémem elektronických vazeb.

Na tomto příkladu je ukázáno, jak se u lisovacího zařízení nahrazuje mechanický podávací systém elektronickým podávacím systémem. Operace vytváření a rozpojování vazeb jsou realizovány jako **statické synchronní akce**.

Na základě řídicí osy LW (osa razníku) jsou definovaným způsobem prostřednictvím tabulek křivek řízeny osy podavače a pomocné osy jako vlečné osy.

Vlečné osy

X posuv, příp. podélná osa
 YL zavírání, příp. příčná osa
 ZL osa zdvihu
 U posuv válce, pomocná osa
 V vyrovnávací hlava, pomocná osa
 W mazání, pomocná osa

Akce

V rámci synchronních akcí se uskutečňují např. následující činnosti:

- Aktivování vazby, LEADON(<vlečná osa>,<řídící osa>,<číslo tabulky křivky>)
- Deaktivování vazby, LEADOF(<vlečná osa>,<řídící osa>)
- Nastavení skutečné hodnoty, PRESETON(<osa>, <hodnota>)
- Nastavení značky, \$AC_MARKER[i] = <hodnota>
- Druh vazby: reálná/virtuální řídicí hodnota
- Najíždění osami na pozice, POS[<osa>]=<hodnota>

Podmínky

13.3 Osová vazba řídicí hodnotou (LEADON, LEADOF)

Jako podmínky jsou vyhodnocovány rychlé digitální vstupy, proměnné v reálném čase \$AC_MARKER a porovnávání pozic, které jsou spojeny pomocí logického operátoru AND.

Poznámka

V následujícím příkladu byly řádkování, řazení bloků a bloky **mazání** použity výlučně k tomu účelu, aby se zvýšila čitelnost programu. Z hlediska řídicího systému je všechno pod jedním číslem bloku považováno za jeden řádek.

Komentář

Programový kód	Komentář
	; definuje veškeré statické synchronní akce
	; ****Vynulování návěští
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 Zapnutí vazby podavače
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 Zapnutí vazby posuvu válce
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 Zapnutí vazby vyrovnávací hlavy
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 Zapnutí vazby mazání
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 Vypnutí vazby
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	

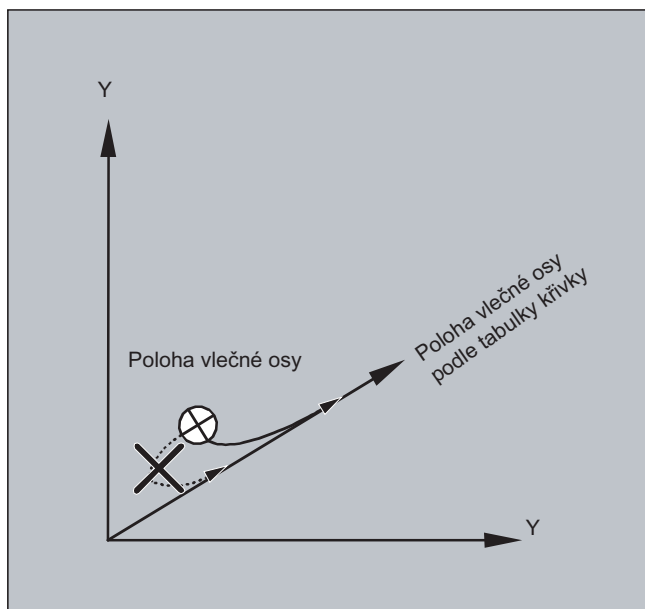
Popis

Vazba řídicí hodnotou vyžaduje synchronizaci řídicí a vlečné osy. Této synchronizace může být dosaženo jen tehdy, pokud se vlečná osa při zapnutí vazby řídicí hodnotou nachází v rámci toleranční oblasti průběhu kontury vypočítaného z tabulky křivky.

Toleranční oblast pro polohu vlečné osy je definována prostřednictvím strojního parametru MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE.

Jestliže se vlečná osa v okamžiku zapnutí vazby řídicí hodnotou ještě nenachází na odpovídající pozici, automaticky se uskuteční synchronizační operace, která zajistí, aby se vypočítaná požadovaná poloha pro vlečnou osu blížila její skutečné poloze. Vlečná osa se přitom v průběhu synchronizační operace pohybuje ve směru, který je definován

prostřednictvím požadované hodnoty rychlosti vlečné osy (vypočítá se na základě rychlosti řídicí osy a podle tabulky křivky CTAB).

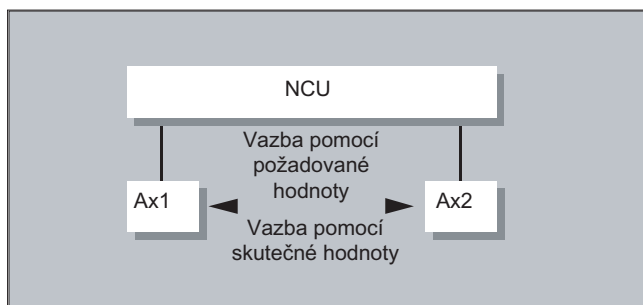


Žádná synchronizační operace

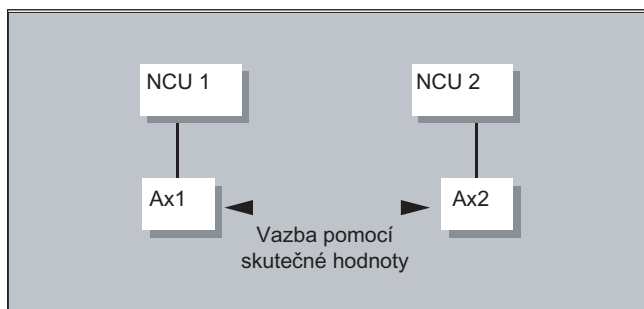
Jestliže se vypočítaná požadovaná poloha vlečné osy při zapnutí vazby řídicí hodnotou vzdaluje od aktuální polohy vlečné osy, žádná synchronizační operace se neuskutečňuje.

Vazba pomocí skutečné a požadované hodnoty

Vazba požadovanou hodnotou poskytuje ve srovnání s vazbou pomocí skutečné hodnoty lepší synchronizaci mezi řídicí a vlečnou osou a je proto předem nastavena jako standardní.



Vazba pomocí požadované hodnoty je možná jedině tehdy, pokud interpolace řídicí a vlečné probíhá ve stejné NCU. V případě externí řídicí osy může být vytvořena vazba mezi touto řídicí osou a vlečnou osou jen pomocí skutečné hodnoty.



Přepnutí je možné uskutečnit pomocí nastavovaného parametru \$SA LEAD TYPE.

Přepnutí mezi vazbou pomocí skutečné a požadované hodnoty by se mělo vždy provádět, když je vlečná osa v klidu. Důvodem je, že po přepnutí proběhne nová synchronizace jen v klidovém stavu.

Příklad použití

Může se stát, že při velkých otřesech stroje nebude čtení skutečné hodnoty bezchybné. Při použití vazby řídicí hodnotou se v případě podavačů u lisů může ukázat jako nezbytné v pracovních krocích s velkými otřesy přepnout z vazby pomocí skutečné hodnoty na vazbu pomocí požadované hodnoty.

Simulace řídicí hodnoty ve vazbách pomocí požadované hodnoty

Pomocí strojního parametru je možné oddělit interpolátor pro řídicí osu od příslušného servomechanismu. Díky tomu je potom možné v případě vazby pomocí požadované hodnoty vytvořit požadovanou hodnotu, aniž by se řídicí osa ve skutečnosti pohybovala.

Rídící hodnoty, ktoré sú vytvorené prostredníctvom väzby pomocí požadovanej hodnoty a ktoré sa využívajú napr. v synchrónnych akciách, je možné načítať z nasledujúcich premenných:

- | | |
|---------------|--------------------------|
| - \$AA_LEAD_P | Řídící hodnota polohy |
| - \$AA_LEAD_V | Řídící hodnota rychlosti |

Vytváření řídicích hodnot

Řídící hodnoty mohou být generovány pomocí libovolných posuvů, které jste sami naprogramovali. Tyto řídící hodnoty vytvořené tímto způsobem se zapisují do proměnných

- | | |
|----------------|--------------------------|
| - \$AA_LEAD_SP | Řídící hodnota polohy |
| - \$AA_LEAD_SV | Řídící hodnota rychlosti |

a je možné je odtud také číst. Aby bylo možné tyto proměnné využívat, musí být nastavovanému parametru přiřazena hodnota `$$SA_LEAD_TYPE = 2`.

Stav vazby

Ve výrobním programu v NC systému můžete pomocí následujících systémových proměnných zjistit stav vazby:

\$AA COUP ACT[osa]

0: Žádná vazba není aktivní

16: Vazba řídící hodnotou je aktivní

Správa stavových proměnných při synchronních akcích

Spínací operace a operace s vazbami jsou spravovány pomocí proměnných v reálném čase:

`$AC_MARKER[i] = n`

, přičemž:

i = číslo značky

n = stavová hodnota

13.4 Elektronická převodovka (EG)

Pomocí funkce "Elektronická převodovka" je možné ovládat pohyb jedné **vlečné osy** po lineárním pohybovém bloku v závislosti na až pěti **řídících osách**. Vzájemné závislosti mezi řídícími osami a vlečnou osou jsou pro každou řídící osu zvlášť definovány pomocí faktoru vazby.

Vypočítaná pohybová složka vlečné osy se vypočítává součtem pohybových složek jednotlivých lineárních os vynásobených příslušnými faktory vazby. Při aktivování svazku os typu "Elektronická převodovka" se může ukázat jako nezbytné synchronizovat vlečnou osu na určitou definovanou pozici. S převodovou vazbou je možné ve výrobním programu uskutečňovat následující operace:

- definice
- aktivování
- deaktivování
- vymazání

.

Podle Vašeho přání může být pohyb vlečné osy odvozen od následujících veličin:

- Požadované hodnoty řídících os
- Skutečné hodnoty řídících os

Jako rozšíření mohou být prostřednictvím **tabulek křivek** (viz kapitola "Chování při pohybu po dráze") realizovány také nelineární vztahy mezi řídícími osami a vlečnou osou. Elektronické převodovky umožňují kaskádovité řazení, tzn. vlečná osa jedné elektronické převodovky se může stát řídící osou pro další elektronickou převodovku.

13.4.1 Definice elektronické převodovky (EGDEF)

Svazek os typu "Elektronická převodovka" je definován stanovením vlečné osy a minimálně jedné, avšak nejvýše pěti řídících os s příslušnými faktory vazby.

Předpoklady

Předpoklad pro definici svazku os typu "Elektronická převodovka":

Pro vlečnou osu nesmí být definována žádná další vazba mezi osami (v případě potřeby musí předem být předcházející vazba vymazána pomocí příkazu EGDEL).

Syntaxe

EGDEF(vlečná osa, řídící osa1, typ vazby1, řídící osa2, typ vazby2, ...)

Význam

EGDEF:	Definice elektronické převodovky
Vlečná osa:	Osa, která je řídícími osami ovlivňována

řídící osa1 , . . . , řídící osa5	Osy, které ovlivňují vlečnou osu	
typ vazby1 , . . . , typ vazby5	Typ vazby Typ vazby nemusí být pro všechny řídící osy stejný a proto je potřeba jej zadat pro každou řídící osu zvlášť.	
	Hodnota:	Význam:
	0	Vlečná osa je ovlivňována skutečnou hodnotou odpovídající řídící osy.
	1	Vlečná osa je ovlivňována požadovanou hodnotou odpovídající řídící osy.

Poznámka

Při definici svazku os typu "Elektronická převodovka" jsou faktorům vazby dosazeny nuly.

Poznámka

Příkaz EGDEF zastavuje předběžné zpracování. Definici převodovky pomocí příkazu EGDEF je potřeba v nezměněné podobě používat také v případě systémů, u kterých jedna nebo více řídících os ovlivňují vlečnou osu na základě **tabulky křivky**.

Příklad:

Programový kód	Komentář
EGDEF(C,B,1,Z,1,Y,1)	; definice svazku os typu "Elektronická převodovka" Řídící osy B, Z, Y ovlivňují vlečnou osu C pomocí požadované hodnoty.

13.4.2 Zapnutí elektronické převodovky (EGON, EGONSYN, EGONSYNE)

Pro zapínání svazku os typu "Elektronická převodovka" existují 3 varianty:

Syntaxe**Varianta 1:**

Svazek os typu "Elektronická převodovka" se zapíná selektivně bez synchronizace pomocí příkazu:

```
EGON(FA,"režim přechodu na další  
blok",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)
```

Varianta 2:

Svazek os typu "Elektronická převodovka" se zapíná selektivně se synchronizací pomocí příkazu:

```
EGONSYN(FA,"režim přechodu na další blok",SynPosFA,  
[,LAi,SynPosLAi,Zi,Ni])
```

Varianta 3:

Svazek os typu "Elektronická převodovka" se zapíná selektivně se synchronizací a s předem zadaným režimem najíždění pomocí příkazu:

EGONSYNE(FA,"režim přechodu na další blok",SynPosFA,režim najíždění[,LAi,SynPosLAi,Zi,Ni])

Význam**Varianta 1:**

FA	Vlečná osa	
Režim přechodu na další blok:	Mohou se používat následující režimy:	
	"NOC"	Na další blok se přechází okamžitě
	"FINE"	Přechod na další blok při „jemném chodu synchronizace“
	"COARSE"	Přechod na další blok při „hrubém chodu synchronizace“
	"IPOSTOP"	Přechod na další blok při dosažení synchronizace s požadovanou hodnotou
LA1, ... LA5	Řídící osy	
Z1, ... Z5	Čítatel pro faktor vazby i	
N1, ... N5	jmenovatel pro faktor vazby i Faktor vazby i = čítatel i/jmenovatel i	

Smí být naprogramovány jediné řídící osy, které byly předtím specifikovány pomocí příkazu EGDEF. Musí být naprogramována nejméně jedna řídící osa.

Varianta 2:

FA	Vlečná osa	
Režim přechodu na další blok:	Mohou se používat následující režimy:	
	"NOC"	Na další blok se přechází okamžitě
	"FINE"	Přechod na další blok při „jemném chodu synchronizace“
	"COARSE"	Přechod na další blok při „hrubém chodu synchronizace“
	"IPOSTOP"	Přechod na další blok při dosažení synchronizace s požadovanou hodnotou
[,LAi,SynPosLAi,Zi,Ni]	(hranaté závorky nezapisovat) Minimálně. 1, max. 5 posloupností s těmito prvky:	
LA1, ... LA5	Řídící osy	
SynPosLAi	Synchronizační pozice pro i-tou řídící osu	
Z1, ... Z5	Čítatel pro faktor vazby i	
N1, ... N5	jmenovatel pro faktor vazby i Faktor vazby i = čítatel i/jmenovatel i	

Smí být naprogramovány jediné řídící osy, které byly předtím specifikovány pomocí příkazu EGDEF. Prostřednictvím naprogramovaných "Synchronizačních pozic" pro vlečnou osu (SynPosFA) a pro řídící osy (SynPosLA) jsou definovány polohy, ve kterých je svazek os

považován za *synchronizovaný*. Jestliže se elektronická převodovka při svém aktivování nenachází v synchronizovaném stavu, najede vlečná osa na svou definovanou synchronizační pozici.

Varianta 3:

Parametry odpovídají variantě 2, a kromě toho ještě:

Způsob odjíždění:	Mohou se používat následující režimy:	
	"NTGT"	V následujícím mezizubí najíždět s časovou optimalizací
	"NTGP"	V následujícím mezizubí najíždět s dráhovou optimalizací
	"ACN"	Kruhovou osou najíždět v záporném směru otáčení na absolutní pozici
	"ACP"	Kruhovou osou najíždět v kladném směru otáčení na absolutní pozici
	"DCT"	Na naprogramovanou synchronizační pozici s časovou optimalizací
	"DCP"	Na naprogramovanou synchronizační pozici s dráhovou optimalizací

Varianta 3 se uplatňuje pouze u vlečných os typu Modulo, které jsou navázány na řídicí osy typu Modulo. Časová optimalizace bere v úvahu mezní hodnoty rychlosti vlečné osy.

Další informace

Popis variant zapnutí

Varianta 1:

Pozice jak řídicí osy, tak i vlečné osy v okamžiku aktivování se ukládají jako "synchronizační pozice". "Synchronizační pozice" mohou být načteny pomocí systémových proměnných \$AA_EG_SYN.

Varianta 2:

Jestliže jsou ve svazku osy typu modulo, jsou hodnoty jejich poloh zmenšeny pomocí funkce Modulo. Tímto způsobem je zajištěno, že se najíždí na nejbližší možnou synchronizační polohu (tzv. *relativní synchronizace*: např. následující mezizubí). Jestliže pro vlečnou osu není nastaven signál rozhraní DB(30 + číslo osy), DBX 26 bit 4 "Uvolnění superpozice vlečné osy", nebude se na synchronní pozici najíždět. Místo toho se program bloku s příkazem EGONSYN pozastaví a aktivuje se alarm 16771 s automatickým potvrzováním, dokud není výše uvedený signál nastaven.

Varianta 3:

Vzdálenost mezi zuby vyplývá z následujícího vztahu: $360 \cdot Z_i/N_i$. V případě, že je vlečná osa v okamžiku vyvolání zastavena, je chování v případě dráhové a časové optimalizace stejné.

Pokud se vlečná osa už pohybuje, pomocí příkazu NTGP se uskuteční synchronizace na následující mezizubí nezávisle na momentální rychlosti vlečné osy. Pokud se vlečná osa už pohybuje, pomocí příkazu NTGT se uskuteční synchronizace na následující mezizubí v závislosti na momentální rychlosti vlečné osy. Za tím účelem se osa v případě potřeby zabrzdí.

Tabulky křivek

Jestliže se pro jednu z řídících os používá **tabulka křivky**, pak musí být splněny tyto podmínky:

Ni	Jmenovatel faktoru lineární vazby musí být nastaven na 0. (Hodnota jmenovatele 0 by byla pro lineární vazby nepřipustná.) Nulový jmenovatel řídicímu systému signalizuje, že:
Zi	má být interpretováno jako číslo tabulky křivky, která se má použít. Tabulka křivky se zadaným číslem musí být už v okamžiku aktivování vazby definována.
LAi	Zadání řídicí osy odpovídá údajům řídicí osy ve vazbě pomocí faktoru vazby (lineární vazba).

Pokud budete potřebovat další informace týkající se používání tabulek křivek a kaskádového řazení elektronických převodovek a jejich synchronizace, viz:

Literatura:

Příručka k funkcím, Speciální funkce; Spojení os a ESR (M3), kapitola "Vlečení a vazba řídicí hodnotou".

Chování elektronické převodovky při zapnutí (Power-On), resetu, změně provozního režimu, vyhledávání bloku

- Po zapnutí systému není aktivní **žádná** vazba.
- V případě resetu a změny provozního režimu zůstávají aktivní vazby zachovány.
- Při vyhledávání bloku nejsou příkazy pro aktivování, mazání a definice elektronických převodovek prováděny a nejsou ani shromažďovány, nýbrž jsou přeskokovány.

Systémové proměnné elektronických převodovek

Pomocí systémových proměnných elektronických převodovek mohou být ve výrobním programu zjišťovány aktuální stavy svazku os typu "Elektronická převodovka" a v případě nutnosti je možno na ně reagovat.

Systémové proměnné elektronických převodovek jsou označeny následujícím způsobem:

\$AA_EG_ ...

nebo

\$VA_EG_ ...

Literatura:

Příručka k systémovým proměnným

13.4.3 Vypnutí elektronické převodovky (EGOFS, EGOFC)

Pro vypínání aktivního svazku os typu "Elektronická převodovka" existují 3 varianty:

Programování

Varianta 1:

Syntaxe	Význam
EGOFS (vlečná osa)	Elektronická převodovka se vypne. Vlečná osa bude zabrzděna až do úplného klidu. Tento příkaz zastavuje předběžné zpracování.

Varianta 2:

Syntaxe	Význam
EGOFS (vlečná osa, řídící osa1, ..., řídící osa5)	Tyto programové příkazy umožňují selektivně odstraňovat vliv jednotlivých řídících os na pohyby vlečné osy.

Musí být uvedena nejméně jedna řídící osa. Vliv uvedených řídících os na vlečnou osu se cíleně vypne. Tento příkaz zastavuje předběžné zpracování. Pokud zůstávají ještě nějaké aktivní řídící osy, pohybuje se vlečná osa podle jejich působení i nadále. Jestliže jsou tímto způsobem zrušeny všechny vlivy řídících os, vlečná osa se zabrzdí do úplného klidu.

Varianta 3:

Syntaxe	Význam
EGOFC (vlečná osa1)	Elektronická převodovka se vypne. Vlečné vřeteno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí. Tento příkaz zastavuje předběžné zpracování.

Poznámka

Tato varianta je povolena jen pro vřetena.

13.4.4 Vymazání definice elektronické převodovky (EGDEL)

Svazek os typu "Elektronická převodovka" musí být vypnutý, jinak jeho definici není možné vymazat.

Programování

Syntaxe	Význam
EGDEL (vlečná osa)	Definice vazby svazku os se vymaže. Dokud není dosaženo maximálního počtu současně aktivních svazků os je možné pomocí příkazu EGDEF nově definovat další svazky os. Tento příkaz zastavuje předběžné zpracování.

13.4.5 Otáčkový posuv (G95) / elektronická převodovka (FPR)

Pomocí příkazu FPR může být i vlečná osa elektronické převodovky zadána jako osa, která určuje hodnotu otáčkového posuvu. V tomto případě se systém chová následujícím způsobem:

- Posuv je závislý na požadované hodnotě rychlosti vlečné osy elektronické převodovky.
- Požadovaná hodnota rychlosti se vypočítává z rychlosti řídicího vřetena a řídicích os typu Modulo (které nejsou dráhovými osami) a z jim přiřazených faktorů vazby.
- Složky rychlosti z lineárních os, příp. z jiných os než typu Modulo, a superponované pohyby vlečné osy nejsou zohledňovány.

13.5 Synchronní vřeten

V synchronním režimu tvoří řídicí vřeten (LS) a vlečné vřeten (FS) tzv. **synchronní pár vřeten**. Když je aktivní vazba (synchronní režim), sleduje vlečné vřeten pohyby řídicího vřeten v souladu s definovanou funkční závislostí.

Dvojice synchronních vřeten mohou být u každého stroje napevno konfigurovány pomocí specifických kanálových strojních parametrů, mohou být ale definovány také uživatelsky pomocí CNC výrobního programu. V jednom kanálu NC systému mohou být současně připraveny až 2 páry synchronních vřeten.

Vazba může být ve výrobním programu:

- definována, příp. upravována
- aktivována
- deaktivována
- vymazána

Kromě toho je možné v závislosti na stavu programového vybavení provádět tyto operace:

- Čekání na splnění podmínky pro synchronizaci
- Změna chování na přechodech mezi bloky
- Volba druhu vazby, tzn. buď vazba pomocí požadované nebo pomocí řídicí hodnoty, nebo je možno zadat úhlové posunutí mezi řídicím a vlečným vřetenem
- Před aktivováním vazby lze převzít předcházející programové příkazy vlečného vřeten
- Lze provést korekci změřené nebo již známé odchylky synchronního režimu

13.5.1 Synchronní vřeten: Programové příkazy (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

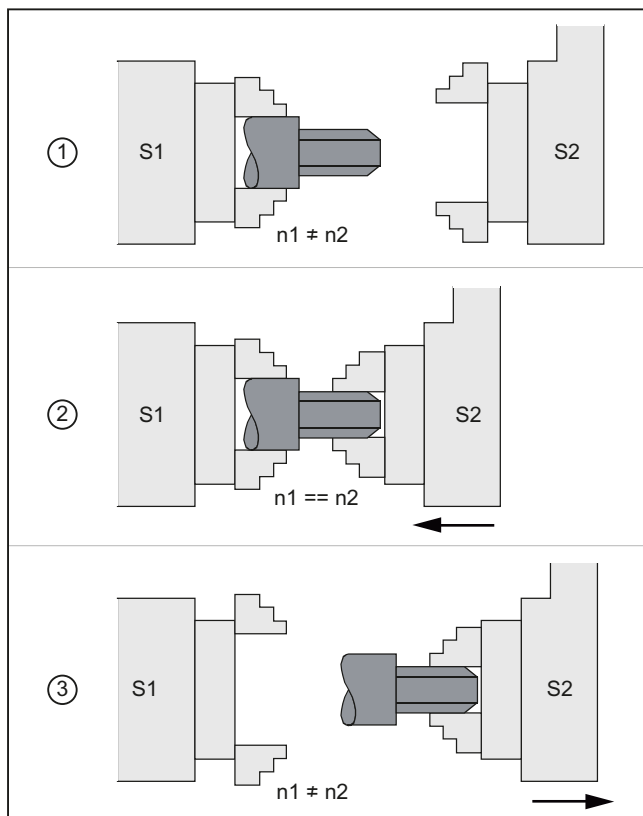
Funkce "Synchronní vřeten" umožňuje ovládat pohyb vlečného vřeten (FS) podle řídicího vřeten (LS) se synchronizovanými otáčkami a s programovatelným převodovým poměrem.

Tato funkce nabízí následující režimy:

- Synchronizace otáček ($n_{FS} = n_{LS}$)
- Synchronizace polohy ($\phi_{FS} = \phi_{LS}$)
- Synchronizace polohy s úhlovým offsetem ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

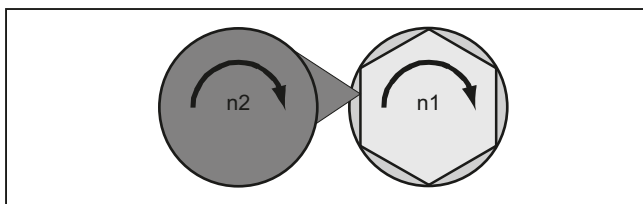
Příklady aplikace:

- Předávání obrobku za pohybu, např. kvůli opracování na zadní straně, převodový poměr: 1:1



- ① Synchronizace otáček
- ② Předání obrobku
- ③ Opracování na zadní straně

- Opracování mnohohranu (soustružení mnohoúhelníku), synchronizace otáček, převodový poměr: $n_1:n_2$



Syntaxe

```
COUPDEF (<FS>, <LS>, <ZFS>, <NLS>, <přechod na další blok>, <druh vazby>)
COUPON (<FS>, <LS>, <POSFS>)
COUPONC (<FS>, <LS>)
COUPOF (<FS>, <LS>, <POSFS>, <POSLS>)
COUPOFS (<FS>, <LS>)
COUPOFS (<FS>, <LS>, <POSFS>)
COUPRES (<FS>, <LS>)
```

COUPDEL (<FS>, <LS>)
WAITC (<FS>, <přechod na další blok>, <LS>, <přechod na další blok>)

Poznámka

Zkrácený způsob zápisu

V případě příkazů COUPOF, COUPOFS, COUPRES a COUPDEL je možný také zkrácený způsob zápisu bez udání řídicího vřeten.

Význam

COUPDEF:	Definice/editace specifické uživatelské vazby
COUPON:	Aktivování vazby. Vlečné vřeten se synchronizuje podle řídicího vřeten, přičemž se vychází z momentálních otáček.
COUPONC:	Vazba při svém aktivování převezme předcházející naprogramované příkazy M3 S... nebo M4 S... Rozdíl otáček vlečného vřeten se okamžitě převezme.
COUPOF:	Deaktivování vazby. - s okamžitým přechodem na další blok: COUPOF (<S2>, <S1>) - Přechod na další blok teprve po přejetí vypínací pozice <POSFS>, příp. <POSLS>: COUPOF (<S2>, <S1>, <POSFS>) COUPOF (<S2>, <S1>, <POSFS>, <POSLS>)
COUPOFS:	Vypnutí vazby se zastavením vlečného vřeten. Co možno nejrychlejší přechod na další blok. COUPOFS (<S2>, <S1>) Přechod na další blok teprve po přejetí vypínací pozice: COUPOFS (<S2>, <S1>, <POSFS>)
COUPRES:	Reset parametrů vazby na hodnoty nastavené v konfiguraci pomocí MD a SD
COUPDEL:	Vymazání uživatelem definované vazby
WAITC:	Čekání na splnění synchronizační podmínky (NOC se při přechodu na další blok uloží do IPO)
<FS>:	Popis vlečného vřeten
Volitelné parametry:	
<LS>:	Označení řídicího vřeten Zadání čísla vřeten: např. S2, S1
<ZFS>, <NLS>:	Převodový poměr mezi FS a LS. $\langle ZFS \rangle / \langle NLS \rangle = \text{čitatel} / \text{jmenovatel}$ Předdefinované nastavení: $\langle ZFS \rangle / \langle NLS \rangle = 1.0$; uvedení jmenovatele je nepovinné

<přechod na další blok>:	Chování při přechodu na další blok	
	Přechod na další blok se uskuteční:	
	"NOC"	okamžitě
	"FINE"	při dosažení "jemného chodu synchronizace"
	"COARSE"	při dosažení "hrubého chodu synchronizace"
<druh vazby>:	"IPOSTOP"	při dosažení zastavení interpolátoru (IPOSTOP), tzn. po dosažení synchronizace s požadovanou hodnotou (předdefinované nastavení)
	Chování při přechodu na další blok má modální působnost.	
	Druh vazby: Vazba mezi FS a LS.	
	"DV"	Vazba pomocí požadované hodnoty (předdefinované nastavení)
	"AV"	Vazba pomocí skutečné hodnoty
<POSFS>:	"VV"	Vazba pomocí rychlosti
	Druh vazby má modální působnost.	
	Úhlové posunutí mezi řídicím a vlečným vřetenem	
	Rozsah hodnot:	0°... 359,999°
	Vypínací pozice vlečného a řídicího vřeten	
<POSFS>, <POSLS>:	"Přechod na další blok se odblokuje až po přejetí pozic POS _{FS} , POS _{LS} ."	
	Rozsah hodnot:	0°... 359,999°

Příklady

Práce s řídicím a vlečným vřetenem

Programový kód	Komentář
N05 M3 S3000 M2=4 S2=500	Řídicí vřeten = Hlavní vřeten = Vřeten 1 Vlečné vřeten = Vřeten 2
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	Řídicí vřeten se otáčí s otáčkami 3000 ot/min, vlečné vřeten se otáčí s otáčkami 500 ot/min.
...	Definice vazby (může být nastavena také v konfiguraci).
N70 SPCON	Řídicí vřeten v režimu polohové regulace (vazba pomocí požadované hodnoty).
N75 SPCON(2)	Vlečné vřeten v režimu polohové regulace.
N80 COUPON(S2,S1,45)	Aktivování vazby za pohybu s offsetovou pozicí = 45 stupňů.
...	
N200 FA[S2]=100	Rychlost polohování = 100 stupňů/min
N205 SPOS[2]=IC(-90)	Najetí na superponovanou pozici 90 stupňů v záporném směru.
N210 WAITC(S2,"Fine")	Čekání na "jemné chod synchronizace".
N212 G1 X... Y... F...	Opracování
...	
N215 SPOS[2]=IC(180)	Najetí na superponovanou pozici 180 stupňů v kladném směru.

Programový kód	Komentář
N220 G4 S50	Doba prodlevy = 50 otáček hlavního vřetena
N225 FA[S2]=0	Aktivování rychlosti nastavené v konfiguraci (MD).
N230 SPOS[2]=IC(-7200)	20 otáček. Pohyb s rychlosti nastavenou v konfiguraci v záporném směru.
...	
N350 COUPOF(S2,S1)	Zrušení vazby za pohybu, S=S2=3000
N355 SPOSA[2]=0	Zastavení FS v poloze nula stupňů.
N360 G0 X0 Y0	
N365 WAIT(2)	Čekání na vřeteno 2.
N370 M5	Zastavení FS.
N375 M30	

Programování rozdílných otáček

Programový kód	Komentář
	Řídicí vřeteno = Hlavní vřeteno = Vřeteno 1 Vlečné vřeteno = Vřeteno 2
N01 M3 S500	Řídicí vřeteno se otáčí s otáčkami 500 ot/min.
N02 M2=3 S2=300	vlečné vřeteno se otáčí s otáčkami 300 ot/min.
...	
N10 G4 F1	Doba prodlevy pro hlavní vřeteno.
N15 COUPDEF (S2,S1,-1)	Faktor vazby s převodovým poměrem -1:1
N20 COUPON (S2,S1)	Aktivování vazby. Otáčky vlečného vřetena vyplývají z otáček řídicího vřetena a z faktoru vazby.
...	
N26 M2=3 S2=100	Programování rozdílných otáček.

Příklady převzetí pohybu pro rozdílné otáčky

1. Aktivování vazby pomocí příkazu COUPON, když byly předtím pro vlečné vřeteno naprogramovány nějaké příkazy

Programový kód	Komentář
	Řídicí vřeteno = Hlavní vřeteno = Vřeteno 1 Vlečné vřeteno = Vřeteno 2
N05 M3 S100 M2=3 S2=200	Řídicí vřeteno se otáčí s otáčkami 100 ot/min, vlečné vřeteno s otáčkami 200 ot/min.
N10 G4 F5	Doba prodlevy = 5 sekund hlavního vřetena
N15 COUPDEF(S2,S1,1)	Převodový poměr FS ku LS je 1,0 (předdefinované nastavení)
N20 COUPON (S2,S1)	Aktivování vazby na řídicí vřeteno za pohybu.
N10 G4 F5	vlečné vřeteno se otáčí s otáčkami 100 ot/min.

2. Aktivování vazby pomocí příkazu COUPONC, když byly předtím pro vlečné vřeten naprogramovány nějaké příkazy

Programový kód	Komentář
	Řídicí vřeten = Hlavní vřeten = Vřeten 1 Vlečné vřeten = Vřeten 2
N05 M3 S100 M2=3 S2=200	Řídicí vřeten se otáčí s otáčkami 100 ot/min, vlečné vřeten s otáčkami 200 ot/min.
N10 G4 F5	Doba prodlevy = 5 sekund hlavního vřeten
N15 COUPDEF(S2,S1,1)	Převodový poměr FS ku LS je 1,0 (předdefinované nastavení)
N20 COUPONC(S2,S1)	Aktivování vazby na řídicí vřeten za pohybu a převzetí předcházejících otáček pro S2.
N10 G4 F5	S2 se otáčí s rychlostí 100 ot/min + 200 ot/min = 300 ot/min

3. Aktivování vazby pomocí příkazu COUPON, když vlečné vřeten stojí

Programový kód	Komentář
	Řídicí vřeten = Hlavní vřeten = Vřeten 1 Vlečné vřeten = Vřeten 2
N05 SPOS=10 SPOS[2]=20	Vlečné vřeten S2 v polohovacím režimu.
N15 COUPDEF(S2,S1,1)	Převodový poměr FS ku LS je 1,0 (předdefinované nastavení)
N20 COUPON (S2,S1)	Aktivování vazby na řídicí vřeten za pohybu.
N10 G4 F1	Vazba je vytvořena, S2 zůstává stát v poloze 20 stupňů.

4. Aktivování vazby pomocí příkazu COUPONC, když vlečné vřeten stojí

Poznámka

Polohovací nebo osový režim

Jestliže se vlečné vřeten před zapojením do vazby nachází v polohovacím nebo v osovém režimu, potom se toto vlečné vřeten chová v případě příkazů COUPON (<FS>, <LS>) a COUPONC (<FS>, <LS>) stejně.

Poznámka

Řídicí vřeten a osový režim

Jestliže se řídicí vřeten nachází před definicí vazby v režimu osy, po aktivování vazby se bude uplatňovat mezní hodnota rychlosti ze strojního parametru:

MD32000 \$MA_MAX_AX_VELO (maximální rychlost osy)

Aby se tomuto chování zabránilo, musí být osa před definicí vazby přepnuta do režimu vřeten (M3 S... nebo M4 S...).

Další informace

Vazba nastavená v konfiguraci

V případě vazby nastavené v konfiguraci jsou řídicí (LS) a vlečné (FS) vřeteno definovány pomocí strojního parametru. Vřetena nastavená v konfiguraci mohou být ve výrobním programu změněna. Parametry vazby je možno ve výrobním programu nastavovat pomocí příkazu `COUPDEF` (předpoklad: ochrana proti zápisu není aktivována).

Uživatelsky definovaná vazba

Pomocí příkazu `COUPDEF` může být vazba ve výrobním programu nově definována nebo změněna. Jestliže už je nějaká vazba aktivní, musí být před definicí nové vazby napřed vymazána pomocí příkazu `COUPDEL`.

Vazba je úplně definována pomocí následujícího příkazu:

`COUPDEF (<FS>, <LS>, <ÜFS>, <ÜLS>, chování na přechodu mezi bloky, druh vazby)`

Vlečné vřeteno (FS) a řídicí vřeteno (LS)

Pomocí názvů os pro vlečné (FS) a řídicí (LS) vřeteno je vazba jednoznačně definována. Názvy os musí být naprogramovány u každého příkazu `COUPDEF`. Ostatní parametry vazby mají modální platnost a musí být naprogramovány jedině tehdy, pokud jsou měněny.

Příklad:

`COUPDEF (S2, S1)`

Převodový poměr

Převodový poměr se udává jako poměr otáček mezi vlečným (FS) a řídicím (LS) vřetenem:

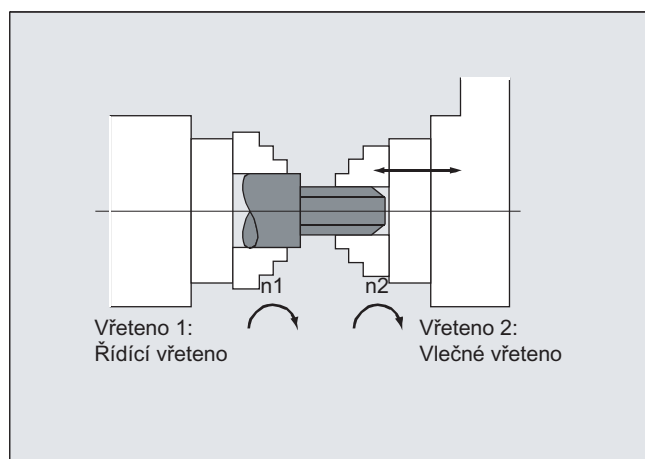
$\text{Vlečné vřeteno} / \text{řídicí vřeteno} = \text{čitatel} / \text{jmenovatel}$

Čítatel musí být naprogramován. Jmenovatel nemusí být naprogramován. Pro jmenovatele se potom dosazuje předdefinovaná hodnota 1,0.

Příklad:

Vlečné vřeteno S2 a řídicí vřeteno S1, převodový poměr = 1 / 1

`COUPDEF (S2, S1, 1.0)`



Poznámka

Převodový poměr může být upravován i v době, kdy je vazba aktivní a vřeteno se otáčí.

Chování na přechodech mezi bloky NOC, FINE, COARSE, IPOSTOP

Při programování chování na přechodech mezi bloky jsou možné následující zkrácené způsoby zápisu:

- "NO": okamžitě (předdefinované nastavení)
- "FI": při dosažení "jemného chodu synchronizace"
- "CO": při dosažení "hrubého chodu synchronizace"
- "IP": při dosažení zastavení interpolátoru (IPOSTOP), tzn. po dosažení synchronizace s požadovanou hodnotou

Druh vazby

Poznámka

Druh vazby se smí měnit jedině tehdy, když je vazba deaktivovaná.

Aktivování synchronizačního režimu, COUPON, <POSFS>

- Aktivování vazby s libovolným úhlovým offsetem mezi osami LS a FS:
 - COUPON (S2, S1)
 - COUPON (S2)
- Aktivování vazby s úhlovým offsetem <POSFS>
Příkaz <POSFS> se vztahuje na polohu 0° řídicího vřetena v kladném směru otáčení
Rozsah hodnot pro příkaz <POSFS>: 0°... 359,999°
 - COUPON (S2, S1, 30)

Poznámka

Úhlový offset je možné změnit, i když je vazba aktivní.

Polohování vlečného vřetena

I když je aktivována vazba mezi synchronními vřeteny, je možno provádět nastavování polohy FS v rozsahu $\pm 180^\circ$, a to nezávisle na LS.

- Nastavování polohy vlečného vřetena (FS) pomocí příkazu SPOS
Příklad: SPOS [2] = IC (-90)
Pokud budete potřebovat podrobnější informace týkající se příkazu SPOS, viz:
Literatura:
Programovací příručka, Základy

Rozdíl otáček

Rozdíl otáček v režimu regulace otáček a s aktivní vazbou mezi synchronními vřeteny vzniká prostřednictvím sečtení s příslušným znaménkem otáček FS na základě pohybu LS a otáček FS na základě naprogramovaných příkazů pro vřeteno:

- Vazba mezi synchronními vřeteny COUPONC
- $S<FS>=<otáčky> [M<FS>=<směr otáčení>]$

Poznámka

Okrajové podmínky

- Při změně směru otáčení M3 / M4 musí být také znovu naprogramovány otáčky S . . .
- Sčítání otáček vřetena ($M<směr otáčení> S<FS>$) prostřednictvím pohybu LS v případě vazby synchronních vřeten COUPONC je v platnosti jen tehdy, pokud je superpozice tohoto druhu povolena.
- Dynamika řídicího vřetena musí být omezena natolik, aby při superpozici vlečného vřetena nemohlo dojít k překročení mezních hodnot jeho dynamiky.

Pokud budete potřebovat další informace, viz:

Literatura:

Příručka k funkcím, Rozšiřovací funkce; Synchronní vřeteno (S3)

Rychlost, zrychlení: FA, ACC, OVRA, VELOLIMA

Rychlost a zrychlení osy vlečného vřetena mohou být naprogramovány pomocí těchto příkazů:

- $FA[SPI(S<n>)]$, příp. $FA[S<n>]$ (axiální rychlost)
- $ACC[SPI(S<n>)]$, příp. $ACC[S<n>]$ (axiální zrychlení)
- $OVRA[SPI(S<n>)]$, příp. $OVRA[S<n>]$ (axiální korekce (override))
- $VELOLIMA[SPI(S<n>)]$, příp. $VELOLIMA[S<n>]$ (axiální zvýšení, příp. snížení rychlosti)

Kde $<n> = 1, 2, 3, \dots$ (číslo vlečného vřetena)

Literatura:

Programovací příručka, Základy

Poznámka

Zmenšení nebo zvětšení maximálního ryvu pro danou osu nemá u vřeten žádný význam.

Pokud budete potřebovat podrobnější informace týkající se dynamiky os, viz:

Literatura:

Příručka k funkcím, Rozšiřovací funkce; Kruhové osy (R2)

Programovatelné chování při přechodech mezi bloky, WAITC

Pomocí příkazu WAITC může být zadáno chování na přechodech mezi bloky, např. po změně parametrů vazby nebo při polohovacích operacích, s různými podmínkami chování synchronizace (hrubá, jemná, IPOSTOP). Jestliže není žádná podmínka chodu synchronizace uvedena, platí chování při přechodech mezi bloky zadané při definici v příkazu COUPDEF.

Příklady

- Čekání na dosažení podmínky chodu synchronizace FINE u vlečného vřetena S2 a COARSE u vlečného vřetena S4: WAITC (S2, "FINE", S4, "COARSE")
- Čekání na dosažení podmínky synchronizovaného chodu v souladu s příkazem COUPDEF: WAITC ()

Deaktivování vazby COUPOF

Pomocí příkazu COUPOF je možné definovat chování při deaktivování vazby:

- Vypnutí vazby s okamžitým přechodem na další blok:
 - COUPOF (S2, S1) (s uvedením řídicího vřetena)
 - COUPOF (S2) (bez uvedení řídicího vřetena)
- Deaktivování vazby po přejetí vypínací pozice. Přechod na následující blok se provede po přejetí vypínací pozice.
 - COUPOF (S2, S1, 150) (vypínací pozice FS: 150°)
 - COUPOF (S2, S1, 150, 30) (vypínací pozice FS: 150°, LS: 30°)

Deaktivování vazby se zastavením vlečného vřetena, COUPOFS

Pomocí příkazu COUPOFS je možné definovat chování při deaktivování vazby se zastavením vlečného vřetena:

- Vypnutí vazby se zastavením vlečného vřetena a s okamžitým přechodem na další blok:
 - COUPOFS (S2, S1) (s uvedením řídicího vřetena)
 - COUPOFS (S2) (bez uvedení řídicího vřetena)
- Deaktivování vazby po přejetí vypínací pozice se zastavením vlečného vřetena. Přechod na následující blok se provede po přejetí vypínací pozice.
 - COUPOFS (S2, S1, 150) (vypínací pozice FS: 150°)

Vymazání vazby, COUPDEL

Pomocí příkazu COUPDEL se vazba vymaže:

- COUPDEL (S2, S1) (s uvedením řídicího vřetena)
- COUPDEL (S2) (bez uvedení řídicího vřetena)

Obnovení původního nastavení parametrů vazby, COUPRES

Pomocí příkazu COUPRES je možno aktivovat hodnoty parametrů vazby, které jsou uloženy ve strojních a nastavovaných parametrech:

- COUPRES (S2, S1) (s uvedením řídicího vřetena)
- COUPRES (S2) (bez uvedení řídicího vřetena)

Systémové proměnné

- Aktuální stavové informace o vazbě vlečného vřetena
Aktuální stavové informace o vazbě vlečného vřetena je možné načítat jako bitově kódované pomocí:
<hodnota> = \$AA_COUP_ACT [<FS>]

Bit	<hodnota>	Význam
-	0	žádná vazba není aktivní
2	4	synchronní vazba mezi vřeteny je aktivní

Poznámka

- Všechny ostatní hodnoty se vztahují na režim osy
- Jestliže je vřetení vlečným vřetenem pro větší počet vazeb, bude systém poskytovat jako hodnotu stavu všech vazeb jejich součtový stav.

- Aktuální úhlový offset

Aktuální úhlový offset vlečného vřetení vůči řídicímu vřetení je možné načítat pomocí:

- \$AA_COUP_OFFSETS [<FS>] (úhlový offset vztahující se na požadovanou hodnotu)
- \$VA_COUP_OFFSETS [<FS>] (úhlový offset vztahující se na skutečnou hodnotu)

Příklad použití

Korekce rozdílu úhlového offsetu v NC programu po deaktivování režimu vlečení:

Rozdíl úhlového offsetu = naprogramovaný úhlový offset - systémová proměnná

Literatura

Pokud budete potřebovat podrobné informace týkající se systémových proměnných, viz:

Příručka pro sestavy systémových proměnných

13.6 Generická vazba (CP...)

"Generická vazba" je všeobecná funkce pro vytváření vazeb, ve které jsou zahrnuty všechny vlastnosti již existujících druhů vazby (vlečení, vazba řídicí hodnotou, elektronická převodovka a synchronní vřetenno).

Tato funkce umožňuje flexibilní způsob programování:

- Pro své aplikace může uživatel volně vybírat vlastnosti vazeb, které jsou zapotřebí (modulární princip).
- Každá vlastnost vazby je samostatně programovatelná.
- Vlastnosti vazby, která již byla definována (např. faktor vazby), je možné změnit.
- Je možné také pozdější využití dalších vlastností vazby.
- Vztažný souřadný systém pro vlečnou osu (základní souřadný systém nebo souřadný systém stroje) může být naprogramován.
- Určité vlastnosti vazeb mohou být programovány i v synchronních akcích.

Literatura: Příručka k funkcím, Synchronní akce

Poznámka

Až dosud bylo volání vazby pro vlečení (TRAIL*), vazbu řídicí hodnotou (LEAD*), elektronickou převodovku (EG*) a synchronní vřetenno (COUP*) dále podporováno pomocí přizpůsobovacích cyklů.

Přehled všech klíčových slov a vlastností vazeb

Následující tabulka ukazuje přehled všech klíčových slov generické vazby a vlastností vazby, jež mohou být jimi naprogramovány:

Klíčové slovo	Vlastnost vazby / význam	Syntaxe
CPDEF	Definice režimu vazby	CPDEF= (<FAx>)
CPDEL	Vymazání režimu vazby	CPDEL= (<FAx>)
CPLA	Definice řídicí osy	CPLA [<FAx>] = (<LAx>)
CPLDEF	Definice řídicí osy a definice režimu osy (možné také pomocí příkazů CPDEF + CPLA)	CPLDEF [<FAx>] = (<LAx>) nebo CPDEF= (<FAx>) CPLA [<FAx>] = (<LAx>)
CPLDEL	Vymazání řídicí osy a režimu vazby (možné také pomocí příkazů CPDEL + CPLA)	CPLDEL [<FAx>] = (<LAx>) nebo CPDEL= (<FAx>) CPLA [<FAx>] = (<LAx>)
CPON	Aktivování režimu vazby	CPON= (<FAx>)
CPOF	Deaktivování režimu vazby	CPOF= (<FAx>)
CPLON	Aktivování řídicí osy režimu vazby	CPLON [<FAx>] = <LAx>

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPLOF	Deaktivování řídicí osy režimu vazby	CPLOF [<FAx>] = <LAx>		
CPLNUM	Číselník faktoru vazby	CPLNUM [FAx, LAx] = <hodnota>		
CPLDEN	Jmenovatel faktoru vazby	CPLDEN [FAx, LAx] = <hodnota>		
CPLCTID	Číslo tabulky křivky	CPLCTID [FAx, LAx] = <hodnota>		
CPLSETVAL	Vztažný objekt vazby	CPLSETVAL [FAx, LAx] = "<Vztažný objekt vazby>"		
		"<Vztažný objekt vazby>":	"CMDPOS"	Vazba pomocí požadované hodnoty
			"CMDVEL"	Vazba pomocí rychlosti
			"ACTPOS"	Vazba pomocí skutečné hodnoty
CPFRS	Vztažný souřadný systém	CPFRS [FAx] = "<vztažný souřadný systém>"		
		"<vztažný souřadný systém>":	"BCS"	Základní souřadný systém
			"MCS"	Souřadný systém stroje
CPBC	Kritérium přechodu na další blok	CPBC [FAx] = "<kritérium přechodu na další blok>"		
		"<Kritérium přechodu na další blok>":	"NOC"	Přechod na další blok se uskutečňuje nezávisle na stavu vazby.
			"IPOSTOP"	Přechod na další blok se uskutečňuje při dosažení synchronizace s požadovanou hodnotou.
			"COARSE"	Přechod na další blok se uskutečňuje při dosažení synchronizace s "hrubou" skutečnou hodnotou.
			"FINE"	Přechod na další blok se uskutečňuje při dosažení synchronizace s "jemnou" skutečnou hodnotou.
CPFPOS + CPON	Synchronní poloha vlečné osy při zapnutí	CPON=FAx CPFPOS [FAx] = <hodnota>		
CPLPOS + CPON	Synchronní poloha řídicí osy při zapnutí	CPLPOS [FAx, LAx] = <hodnota>		

Klíčové slovo	Vlastnost vazby / význam	Syntaxe
CPFMSON	Režim synchronní akce	CPFMSON[FAx]="<režim synchronní akce>"
		"<režim synchronní akce>":
		"CFAST" Vazba se ukončí s časovou optimalizací.
		"CCOARSE" Vazba se aktivuje až tehdy, když se poloha vlečné osy vyžadovaná podle předpisu, kterým se vazba řídí, nachází v oblasti aktuální polohy vlečné osy.
		"NTGT" Na následující mezeru mezi zuby se najíždí s časovou optimalizací.
		"NTGP" Na následující mezeru mezi zuby se najíždí s dráhovou optimalizací.
		"NRGT" Na následující segment se najíždí s časovou optimalizací podle poměru počtu chodů ku počtu zubů.
		"NRGP" Na následující segment se najíždí s dráhovou optimalizací podle poměru počtu chodů ku počtu zubů.
		"ACN" Jen u kruhových os! Kruhová osa najíždí ve svém záporném směru na synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.
		"ACP" Jen u kruhových os! Kruhová osa najíždí ve svém kladném směru na synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.
		"DCT" Jen u kruhových os! Kruhová osa najíždí s časovou optimalizací na naprogramovanou synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.
		"DCP" Jen u kruhových os! Kruhová osa najíždí s dráhovou optimalizací na naprogramovanou synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPFMON	Chování vlečné vazby při aktivování	CPFMON[FAx]="<chování při aktivování>"		
		<chování při aktivování>":	"STOP"	Jen u vřeten! Aktivní pohyb vlečného vřetena se před aktivováním zastaví.
			"CONT"	Jen u vřeten a os v hlavní zpracovávané větvi! Aktuální pohyb vlečné osy/vlečného vřetena se přebírá do vazby jako počáteční pohyb.
			"ADD"	Jen u vřeten! Pohybové složky vazby se uplatňují navíc k aktuálnímu superponovanému pohybu, tzn. aktuální pohyb vlečné osy/vlečného vřetena zůstává zachován jako superponovaný pohyb.
CPFMOF	Chování vlečné osy při úplném deaktivování	CPFMOF[FAx]="<chování při deaktivování>"		
		<chování při deaktivování>":	"STOP"	Zastavení vlečné osy/vlečného vřetena Aktivní superponovaný pohyb bude zabrzděn až do klidu. Potom se vazba deaktivuje.
			"CONT"	Jen u vřeten a os v hlavní zpracovávané větvi! Vlečné vřeteno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí.
CPFPOS + CPOF	Vypínací pozice vlečné vazby při deaktivování	CPOF=(FAx) CPFPOS[FAx]=<hodnota>		

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPMRESET	Chování vazby při resetu	CPMRESET[FAx]="<chování při resetu>"		
		"<chování při resetu>":	"NONE"	Aktuální stav vazby zůstává zachován.
			"ON"	Pokud je definován odpovídající modul vazby, vazba se aktivuje. Aktivují se všechny definované vztahy s řídicími osami. To se uskuteční i tehdy, když již všechny nebo některé z těchto vztahů řídicích os jsou již aktivovány, tzn. také v případě zcela aktivované vazby se uskuteční nová synchronizace.
			"OF"	Aktivní superponovaný pohyb bude zabrzděn až do klidu. Potom se vazba deaktivuje. Jestliže byl definován odpovídající modul vazby bez explicitní definice (CPDEF), modul vazby se vymaže. Pokud tomu tak není, zůstává i nadále definován, tzn. je možno jej opět využívat.
			"OFC"	Může být použito jen u vřeten! Vlečné vřetenno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí. Vazba se deaktivuje. Jestliže byl definován odpovídající modul vazby bez explicitní definice (CPDEF), modul vazby se vymaže. Pokud tomu tak není, zůstává i nadále definován, tzn. je možno jej opět využívat.
			"DEL"	Aktivní superponovaný pohyb bude zabrzděn až do klidu. Potom se vazba deaktivuje a následně se vymaže.
			"DELC"	Může být použito jen u vřeten! Vlečné vřetenno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí. Vazba se deaktivuje a následně se vymaže.

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPMSTART	Chování vazby při spuštění výrobního programu	CPMSTART[FAx]="<chování při spuštění>"		
		<chování při spuštění>":	"NONE"	Aktuální stav vazby zůstává zachován.
			"ON"	Vazba se aktivuje. Aktivují se všechny definované vztahy s řídicími osami. To se uskuteční i tehdy, když již všechny nebo některé z těchto vztahů řídicích os jsou již aktivovány, tzn. také v případě zcela aktivované vazby se uskuteční nová synchronizace.
			"OF"	Vazba se deaktivuje. Jestliže byl definován odpovídající modul vazby bez explicitní definice (CPDEF), modul vazby se vymaže. Pokud tomu tak není, zůstává i nadále definován, tzn. je možno jej opět využívat.
CPMPRT	Chování vazby při spuštění výrobního programu pomocí hledání bloku prostřednictvím zkušebního zpracování programu	CPMPRT[FAx]="<chování při spuštění>"		
		<chování při spuštění>":	viz CPMSTART	
CPLINTR	Hodnota posunutí pro vstupní hodnotu řídicí osy	CPLINTR[FAx, LAx]=<hodnota>		
CPLINSC	Faktor měřítka pro vstupní hodnotu řídicí osy	CPLINSC[FAx, LAx]=<hodnota>		
CPLOUTTR	Hodnota posunutí pro výstupní hodnotu vazby	CPLOUTTR[FAx, LAx]=<hodnota>		
CPLOUTSC	Faktor měřítka pro výstupní hodnotu vazby	CPLOUTSC[FAx, LAx]=<hodnota>		
CPSYNCOF	Prahová hodnota pro "hrubé" synchronní najíždění na určitou pozici	CPSYNCOF[FAx]=<hodnota>		
CPSYNFIP	Prahová hodnota pro "jemné" synchronní najíždění na určitou pozici	CPSYNFIP[FAx]=<hodnota>		
CPSYNCOF2	Druhá mezní hodnota pro "hrubé" synchronní najíždění na určitou pozici	CPSYNCOF2[FAx]=<hodnota>		
CPSYNFIP2	Druhá mezní hodnota pro "jemné" synchronní najíždění na určitou pozici	CPSYNFIP2[FAx]=<hodnota>		

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPSYNCOV	Mezní hodnota pro "hrubé" synchronní najíždění na určitou rychlost	CPSYNCOV [FAx]=<hodnota>		
CPSYNFIV	Mezní hodnota pro "jemné" synchronní najíždění na určitou rychlost	CPSYNFIV [FAx]=<hodnota>		
CPMBRAKE	Chování vlečné osy při určitých signálech a příkazech pro zastavení	CPMBRAKE [FAx]=<bitově kódovaná hodnota>		
CPMVDI	Chování vlečné osy při určitých signálech rozhraní NC/PLC	CPMVDI [FAx]=<bitově kódovaná hodnota>		
CPMALARM	Potlačení výstupu speciálních alarmů vztahujících se k vazbě	CPMALARM [FAx]=<bitově kódovaná hodnota>		
CPSETTYPE	Typ vazby	CPSETTYPE [FAx]="<typ vazby>"		
		<typ vazby>":	"CP"	Možnost volného programování
			"TRAIL"	Typ vazby "Vlečení"
			"LEAD"	Typ vazby "Vazba řídicí hodnotou"
			"EG"	Typ vazby "Elektronická převodovka"
			"COUP"	Typ vazby "Synchronní vřetenno"

FAx: Vlečná osa/vlečné vřetenno

LAX: Řídicí osa/řídicí vřetenno

Poznámka

Vlastnostem vazby, které nejsou explicitně naprogramovány (ve výrobním programu nebo v synchronních akcích), jsou dosazena jejich standardní nastavení.

V závislosti na nastavení klíčového slova CPSETTYPE mohou být dosazeny místo standardních nastavení (CPSETTYPE="CP") také předem nastavené vlastnosti vazby.

Literatura

Pokud budete potřebovat podrobné informace týkající se generické vazby, viz:

- Příručka k funkcím, Speciální funkce; M3: Spojení os pomocí vazby, kapitola: "Generická vazba"

13.7 Vazba Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS):

"Vazba Master/Slave" umožňuje následující:

- Aktivování vazby osy typu Slave s osou typu Master, když jsou obě tyto osy v klidu.
- Aktivování a deaktivování vazby **otáčejících se** vřeten v režimu regulace otáček.
- Dynamická konfigurace.

Poznámka

Režim polohování

U os a vřeten v režimu polohování se vazba aktivuje a rozpojuje, jen když jsou osy v klidu.

Syntaxe

```
MASLON(<Slave_1>,<Slave_2>,...)
MASLOF(<Slave_1>,<Slave_2>,...)
MASLOFS(<Slave_1>,<Slave_2>,...)
```

Dynamická konfigurace:

```
MASLDEF(<Slave_1>,<Slave_2>,...,<Master>)
MASLDEL(<Slave_1>,<Slave_2>,...)
```

Význam

MASLON:	Aktivování dočasné vazby typu Master/Slave	
	<Slave_x>, ...:	Osa typu Slave 1 ... n
MASLOF:	Deaktivování aktivní vazby typu Master/Slave	
	<Slave_1>, ...:	Osa typu Slave 1 ... n
MASLOFS:	Deaktivování vazby typu Master/Slave a automatické zabrždění vřeten typu Slave (viz upozornění "Chování vazby u vřeten v režimu regulace otáček"!)	
	<Slave_1>, ...:	Osa typu Slave 1 ... n
MASLDEF:	Definice/změna spojení typu Master/Slave pomocí výrobního programu	
	<Slave_1>, ...:	Osa typu Slave 1 ... n
	<Master>:	Osa typu Master
MASLDEL:	Zrušení vazby Master/Slave a vymazání definice tohoto spojení	
	<Slave_1>, ...:	Osa typu Slave 1 ... n
	Upozornění: Definice spojení Master/Slave konfigurovaná ve strojních parametrech zůstává zachována.	

Poznámka**Chování vazby u vřeten v režimu regulace otáček**

U vřeten v režimu regulace otáček je chování vazby v případě příkazů MASLON, MASLOF, MASLOFS a MASLDEL explicitně definováno pomocí následujícího strojního parametru:

MD37263 \$MA_MS_SPIND_COUPLING_MODE

Při standardním nastavení MD37263 = 0 se uskutečňuje aktivování a rozpojování vazby s osami typu Slave jen tehdy, jsou-li všechny podílející se osy v klidu. Příkaz MASLOFS odpovídá příkazu MASLOF.

Jestliže je nastaveno MD37263 = 1 příkazy pro ovládání vazby se uskuteční okamžitě, tedy během pohybu. V případě příkazu MASLON se vazba okamžitě zapojí a v případě příkazů MASLOFS nebo MASLOF se ihned rozpojí. Když je v platnosti příkaz MASLOF, vřetena typu Slave, které se v tomto okamžiku otáčejí, si své otáčky zachovávají, dokud nebudou naprogramovány nové. V případě příkazu MASLOFS jsou oproti tomu automaticky zabržděna.

Poznámka

V případě příkazů MASLOF/MASLOFS odpadá implicitní zastavení předběžného zpracování. Protože se neuskutečňuje zastavení předběžného zpracování, neposkytují systémové proměnné \$P... pro osy typu Slave až do okamžiku opětovného naprogramování žádné aktualizované hodnoty.

Poznámka

Pro osu Slave může být pomocí příkazu PRESETON skutečná hodnota synchronizována na stejnou hodnotu, jakou má osa Master. Za tím účelem musí být trvalá vazba Master/Slave na krátkou chvíli deaktivována, aby bylo možné ose Slave, která po zapnutí systému (Power-on) nenajela na referenční bod, dosadit hodnotu osy Master. Potom se trvalá vazba znovu obnoví.

Trvalá vazba typu Master/Slave se aktivuje pomocí následujícího nastavení strojního parametru:

MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE = 1

Toto nastavení nemá žádný vliv na příkazy NC jazyka pro dočasnou osu.

Příklady**Příklad 1: Dosazení skutečné hodnoty ose typu Slave ve vazbě Master-slave**

U permanentní vazby typu Master-slave je možno pomocí příkazu PRESETON nastavit skutečnou hodnotu osy typu slave na hodnotu osy typu master.

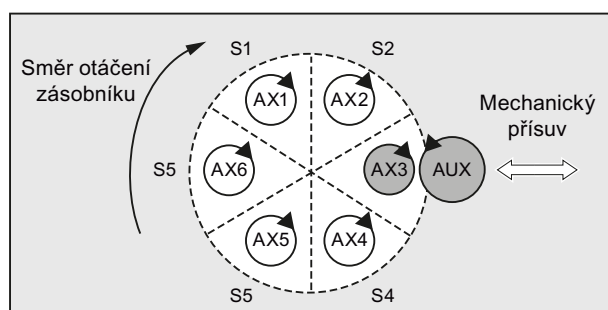
Programový kód	Komentář
\$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; deaktivování permanentní vazby osy typu Slave
NEWCONF	; aktivování změny strojního parametru
STOPRE	
MASLOF(Y1)	; deaktivování dočasného spojení

13.7 Vazba Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS):

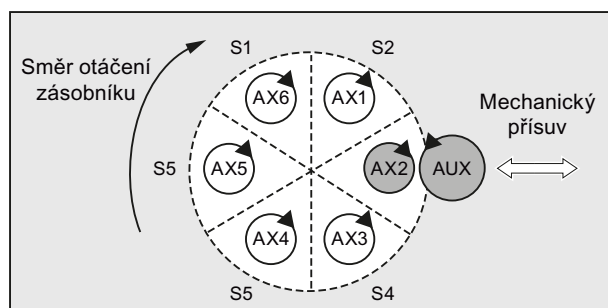
Programový kód	Komentář
PRESETON (AX2, \$VA_IM (M_AX))	; skutečná hodnota osy typu Slave = skutečná hodnota osy typu Master
\$MA_MS_COUPLING_ALWAYS_ACTIVE [AX2]=1	; aktivování permanentní vazby osy typu Slave
NEWCONF	; aktivování změny strojního parametru

Příklad 2: Dynamická konfigurace spojení os typu Master/Slave

Aby po otočení osového zásobníku mohla být vazba připojena na jiné vřeteno, musí být napřed stará vazba rozpojena, její konfigurace vymazána a zadána nová konfigurace vazby.



Obrázek 13-1 Před otáčením osového zásobníku.



Obrázek 13-2 Nové otočení osového zásobníku o jeden slot

Programový kód	Komentář
MASLDEF (AUX, S3)	; AUX: Slave, S3: Master = AX3
MASLON (AUX)	; aktivování vazby
M3=3 S3=4000	; otáčení osy typu Master
MASLDEL (AUX)	; deaktivování a vymazání vazby
AXCTSWE (CT1)	; uvolnění otáčení osového zásobníku
MASLDEF (AUX, S3)	; AUX: Slave, S3: Master = AX2

Literatura

- Příručka k funkcím, Speciální funkce; kapitola „TE3: Vazba pomocí otáček/momentu, master-slave“
- Příručka k funkcím, Rozšiřovací funkce; kapitola „B3: Decentralizované systémy - jen 840D sl" > "NCU-Link" > „Osový zásobník“

Synchronní akce

14.1 Definice synchronní akce

Synchronní akce je definována v jednom bloku výrobního programu. V rámci tohoto bloku nesmějí být naprogramovány žádné další příkazy, které nejsou součástí synchronní akce.

Synchronní akce se skládají z následujících komponent:

Platnost, ident. č. (volitelný)	Podmínková část (volitelný)			Část popisující akci		
	Četnost	G-příkaz (volitelný)	Podmínka	Klíčové slovo	G-příkaz (volitelný)	Akce
--- ¹⁾ ID=<Č.> IDS=<Č.>	--- ¹⁾ WHENEVER FROM WHEN EVERY	G...	Logický výraz	DO	G...	Akce 1 ... Akce n

¹⁾ není naprogramováno

Syntaxe

```
DO <Akce 1> ... <Akce n>
<Četnost> [<G-Funce>] <Podmínka> DO <Akce 1> ... <Akce n>
ID=<Č.> <Četnost> [<G-Funkce>] <Podmínka> DO <Akce 1> ... <Akce n>
IDS=<Č.> <Četnost> [<G-Funkce>] <Podmínka> DO <Akce 1> ... <Akce n>
```

Literatura

Pokud budete potřebovat podrobný popis fungování a příkazů synchronních akcí, viz:

Příručka Popis funkcí, Synchronní akce

Pohyb tam a zpět

15.1 Asynchronní oscilační pohyby (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Osa vykonávající oscilační pohyb se pohybuje mezi dvěma body obratu 1 a 2 se zadanou hodnotou posuvu, dokud se pohyb tam a zpět nevypne.

Ostatní osy mohou být v průběhu oscilačního pohybu libovolně interpolovány. Prostřednictvím pohybu po dráze nebo s polohovací osou je možné dosáhnout spojitého přísuvu. Přitom však neexistuje **žádná souvislost** mezi oscilačním pohybem a přísuvným pohybem.

Vlastnosti asynchronního oscilačního pohybu

- Asynchronní oscilační pohyb je vztažen na specifickou osu a je v platnosti i přes hranice bloků.
- Blokově synchronní zapínání oscilačního pohybu je zajišťováno pomocí výrobního programu.
- Společná interpolace většího počtu os a superpozice úseku pro oscilační pohyb nejsou možné.

Programování

Zapínání a ovlivňování asynchronního oscilačního pohybu u výrobního programu, které odpovídá zpracování NC programu, je možné prostřednictvím následujících příkazů.

Naprogramované hodnoty se synchronně s blokem v rámci zpracování hlavního programu zadávají do odpovídajících nastavovaných parametrů a zůstávají v platnosti, dokud nejsou změněny.

Syntaxe

```
OSP1 [<osa>]=<hodnota> OSP2 [<osa>]=<hodnota>
OST1 [<osa>]=<hodnota> OST2 [<osa>]=<hodnota>
FA [<osa>]=<hodnota>
OSCTRL [<osa>]=(<volba pro aktivování>,<volba pro deaktivování>)
OSNSC [<osa>]=<hodnota>
OSE [<osa>]=<hodnota>
OSB [<osa>]=<hodnota>
OS [<osa>]=1
OS [<osa>]=0
```

Význam

<osa>:	Posuv oscilační osy		
OS:	Zapínání a vypínání oscilačního pohybu		
	Hodnota:	1	Aktivování oscilačního pohybu
		0	Deaktivování oscilačního pohybu

OSP1:	Definice polohy bodu obratu 1		
OSP2:	Definice polohy bodu obratu 2 Upozornění: Jestliže je aktivní inkrementální zadávání rozměrů, vypočítá se poloha inkrementálně vůči odpovídajícímu bodu obratu naposled naprogramovanému v NC programu.		
OST1:	Definice doby pozastavení v bodě obratu 1 v [s].		
OST2:	Definice doby pozastavení v bodě obratu 2 v [s].		
	<hodnota >:	-2	Interpolace bude pokračovat bez čekání na přesné najetí
		-1	Čekání na hrubé přesné najetí
		0	Čekání na jemné přesné najetí
		>0	Čekání na jemné přesné najetí a následné setrvání po zadanou dobu pozastavení Upozornění: Jednotky pro dobu pozastavení jsou identické s dobou pozastavení naprogramovanou pomocí příkazu G4.
FA:	Definice rychlosti posuvu Jako rychlost posuvu platí definovaná rychlost posuvu polohovací osy. Jestliže rychlost posuvu není definována, platí hodnota uložená ve strojním parametru.		
OSCTRL:	Zadání možností pro aktivování a deaktivování Hodnoty voleb 0 - 3 určují chování v bodech obratu při vypínání. Mohou být zvoleny varianty 0 - 3. Zbývající nastavení mohou být podle potřeby kombinovány se zvolenou variantou. Větší počet voleb může být uveden postupně pomocí znamének plus (+).		
	<hodnota >:	0	Při vypnutí oscilačního pohybu zastavit v následujícím bodě obratu (předdefinované nastavení). Upozornění: Hodnoty 1 a 2 jsou možné jen prostřednictvím resetu.
		1	Při vypnutí oscilačního pohybu zastavit v bodě obratu 1.
		2	Při vypnutí oscilačního pohybu zastavit v bodě obratu 2.
		3	Při vypnutí oscilačního pohybu nenajíždět do žádného bodu obratu, jestliže nejsou naprogramovány žádné vyjiskřovací zdvihy.
		4	Po vyjiskření najet na koncovou pozici.
		8	Jestliže je oscilační pohyb přerušen prostřednictvím vymazání zbytkové dráhy, mají se potom zpracovat vyjiskřovací zdvihy a v případě potřeby se má najet na koncovou pozici.
		16	Jestliže je oscilační pohyb přerušen prostřednictvím vymazání zbytkové dráhy, má se po vypnutí najet do odpovídajícího bodu obratu.
		32	Změněná hodnota posuvu se uplatní teprve až od následujícího bodu obratu.
		64	FA se rovná 0, FA = 0: Superpozice dráhy je aktivní. FA se nerovná 0, FA <> 0: Superpozice rychlosti je aktivní.
		128	V případě kruhové osy DC (nejkratší dráha)
		256	Vyjiskřovací zdvih se uskuteční jako dvojitý zdvih (standardní) 1 = vyjiskřovací zdvih se uskuteční jako jeden zdvih.
		512	Najíždění napřed na počáteční pozici
OSNSC:	Stanovení počtu vyjiskřovacích zdvihů		

15.1 Asynchronní oscilační pohyby (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

OSE:	Definice koncové pozice (ve WCS), na kterou se má po vypnutí oscilačního pohybu najet. Poznámka: Jestliže byl naprogramován příkaz „OSE“, pro příkaz „OSCTRL“ systém automaticky aktivuje volbu 4.
OSB:	Definice počáteční pozice (ve WCS), na kterou se má před zapnutím oscilačního pohybu najet. Na počáteční pozici se najíždí před bodem obratu 1. Jestliže se počáteční pozice kryje s polohou bodu obratu 1, jako následující se najíždí na bod obratu 2. Při dosažení počáteční pozice se neuplatňuje žádný doba prodlevy, a to i když se počáteční pozice kryje s bodem obratu 1. Místo toho se čeká na jemné přesné najetí. Nastavená podmínka přesného najetí je dodržena. Upozornění: Aby se najelo na počáteční pozici, musí být v nastavovaném parametru SD43770 \$SA_OSCILL_CTRL_MASK nastaven bit 9.

Příklady

Příklad 1: Osa vykonávající oscilační pohyb se má pohybovat mezi dvěma body obratu

Osa Z se má pohybovat tam a zpět mezi polohami 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. Posuv pro oscilační osu má činit 250. Na konci opracování se mají uskutečnit 3 vyjiskřovací zdvihy a potom má oscilační osa odjet na koncovou pozici 200. Posuv pro přísuvnou osu má být 1, konec přísuvu ve směru osy X má být dosažen v poloze 15.

Programový kód	Komentář
WAITP(X,Y,Z)	; výchozí nastavení
G0 X100 Y100 Z100	; přepnutí do režimu polohování osy
WAITP(X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; bod obratu 1, bod obratu 2
OSE[Z]=200	; koncová pozice
OST1[Z]=0 OST2[Z]=-1	; doba pozastavení v bodě U1: Jemné přesné najetí ; doba pozastavení v bodě U2: Hrubé přesné najetí
FA[Z]=250 FA[X]=1	; posuv oscilační osy, posuv přísuvné osy
OSCTRL[Z]=(4,0)	; možnosti nastavení
OSNSC[Z]=3	; 3 vyjiskřovací zdvihy
OS[Z]=1	; spuštění oscilačního pohybu
WHEN \$A_IN[3]==TRUE DO DELDTG(X)	; vymazání zbytkové dráhy
POS[X]=15	; počáteční poloha osy X
POS[X]=50	; koncová poloha osy X
OS[Z]=0	; zastavení oscilačního pohybu
M30	

Poznámka

Posloupnost příkazů „OSP1[Z]=...“ až „OSNCS[Z]=...“ může být naprogramována také v jednom bloku.

Příklad 2: Oscilační pohyb s on-line změnou bodu obratu

Nastavované parametry, které jsou zapotřebí pro asynchronní oscilační pohyby, mohou být nastavovány ve výrobním programu.

Jestliže se do nastavovaných parametrů zapisují hodnoty přímo ve výrobním programu, je změna v platnosti už v okamžiku předběžného zpracování. Synchronního chování lze dosáhnout pomocí zastavení předběžného zpracování (STOPRE).

Programový kód	Komentář
\$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	; jestliže dojde k překročení skutečné hodnoty oscilační osy v bodě obratu, příslušná osa se zastaví
ID=2 WHENEVER \$AA_IM[Z] < \$\$AA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; zapnutí oscilačního pohybu
OS[Z]=0	; vypnutí oscilačního pohybu
M30	

Další informace

Osa vykonávající pohyb tam a zpět

Pro osu vykonávající pohyb tam a zpět platí:

- Pro oscilační pohyb je možné využít kteroukoli osu.
- Současně může být aktivních i více os vykonávajících pohyb tam a zpět (maximum: celkový počet polohovacích os).
- Pro oscilační osu je vždy aktivní lineární interpolace G1, a to nezávisle na G-příkazu, který momentálně platí v programu.

Oscilační osa může:

- být vstupní osou pro dynamickou transformaci
- být řídicí osou pro osy gantry a vlečné osy
- se pohybovat:
 - bez omezení ryvu „BRISK“ nebo
 - s omezením ryvu „SOFT“ nebo
 - s nespojitou charakteristikou zrychlení (jako u polohovacích os)

Body obratu oscilačního pohybu

Při určování poloh pro pohyb tam a zpět je zapotřebí mít na paměti momentálně platná posunutí:

- Zadání absolutních hodnot
„OSP1[Z]=<hodnota>”
Poloha bodu obratu = součet posunutí + naprogramovaná hodnota
- Zadání relativních hodnot
„OSP1[Z]=IC(<hodnota>)”
Poloha bodu obratu = bod obratu 1 + naprogramovaná hodnota

Příklad:

Programový kód
N10 OSP1[Z] = 100 OSP2[Z] = 110
...
N40 OSP1[Z] = IC(3)

WAITP

Jestliže má oscilační pohyb vykonávat geometrická osa, musí být pro účely tohoto pohybu uvolněna příkazem "WAITP".

Po ukončení oscilačního pohybu se pomocí příkazu "WAITP" osa vykonávající oscilační pohyb znovu zaregistruje jako polohovací osa a je možné ji znovu normálně používat.

Oscilační pohyby s pohybovými synchronními akcemi a doby pozastavení

Po uplynutí nastavené doby pozastavení se v případě oscilačního pohybu uskuteční interní přechod na další blok (což lze pozorovat podle nové zbytkové dráhy os). Při přechodu na další blok se kontroluje vypínací funkce. Přitom je podle určeného nastavení řídicího systému pro průběh pohybu (OSCTRL) stanovena vypínací funkce. *Toto časové chování může být ovlivněno pomocí korekce (override) posuvu.*

Podle okolností se potom uskuteční ještě jeden oscilační zdvih, než se spustí vyjiskřovací zdvihy nebo než se najede do koncové pozice. *Přitom vzniká dojem, jakoby se změnilo chování při vypínání. Ale není tomu tak.*

15.2 Oscilační pohyby řízené prostřednictvím synchronních akcí (OSCILL)

U tohoto druhu oscilačních pohybů je přísluvný pohyb povolen jedině v bodech obratu, příp. v rámci definované oblasti okolo bodu obratu.

V závislosti na požadavcích mohou být oscilační pohyby v průběhu přísluvu

- uskutečňovány nebo
- pozastaveny, dokud není přísluv zcela dokončen.

Syntaxe

1. Definice parametrů pro oscilační pohyby
2. Definice pohybových synchronních akcí
3. Přřazení os, definice přísluvu

Význam

OSP1[<oscilační osa>]=	Poloha bodu obratu 1
OSP2[<oscilační osa>]=	Poloha bodu obratu 2
OST1[<oscilační osa>]=	Doba prodlevy v bodě obratu 1 v sekundách
OST2[<oscilační osa>]=	Doba prodlevy v bodě obratu 2 v sekundách
FA[<oscilační osa>]=	Posuv oscilační osy
OSCTRL[<oscilační osa>]=	Možnosti aktivování/deaktivování
OSNSC[<oscilační osa>]=	Počet vyjiskřovacích zdvihů
OSE[<oscilační osa>]=	Koncová pozice
WAITP(<oscilační osa>)	Uvolnění osy pro oscilační pohyby

Přřazení os, přísluv

OSCILL[<oscilační osa>]=(<přísluvná osa 1>,<přísluvná osa 2>,<přísluvná osa 3>)

POSP[<přísluvná osa>]=(<koncová pozice>,<dílčí délka>,<režim>)

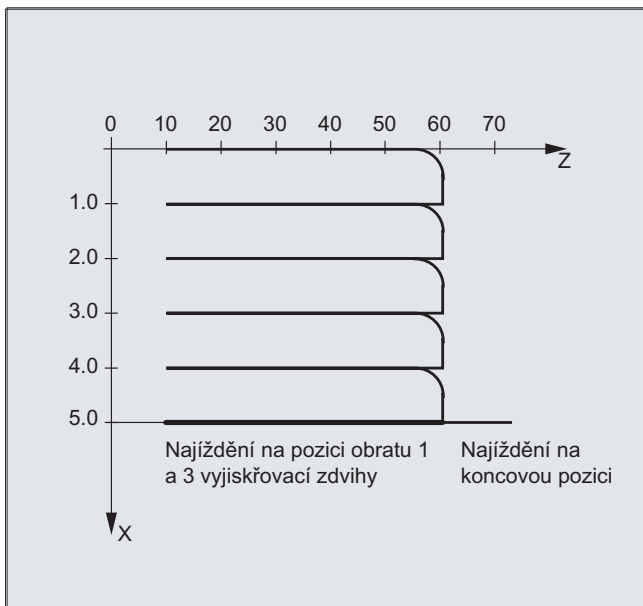
OSCILL:	Přřazení přísluvných os k oscilační ose
POSP:	Definice celkového a dílčího přísluvu (viz kapitola správa souborů a programů)
koncová pozice:	Koncová pozice pro přísluvnou osu poté, co byly uskutečněny všechny dílčí přísluvy.
dílčí délka:	Velikost dílčího přísluvu v bodě obratu/oblasti obratu
režim:	Rozdělení celkového přísluvu do dílčích přísluvů = dva stejně velké zbytkové přísluvy (standardní nastavení), = všechny dílčí přísluvy jsou stejně veliké

Pohybové synchronní akce

WHEN... .. DO	jestliže..., potom...
WHENEVER ... DO	vždycky když..., potom...

Příklad:

V bodě obratu 1 se nemá provádět žádný přísuv. U bodu obratu 2 se má uskutečnit přísuv již ve vzdálenosti $ii2$ před bodem obratu 2 a oscilační osa v bodě obratu nemá čekat na ukončení dílčího přísuvu. Oscilační osou je osa Z a přísuvnou osou je osa X.



1. Parametry pro oscilační pohyby

Programový kód	Komentář
DEF INT ii2	; definice proměnné pro oblast obratu 2
OSP1[Z]=10 OSP2[Z]=60	; definice bodu 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)	; oscilační pohyb vypnout v bodě obratu 2, po vymazání zbytkové dráhy vyjiskření a najíždění do koncové polohy, po vymazání zbytkové dráhy najíždění do odpovídajícího bo- du obratu
OSNC[Z]=3	; vyjiskřovací zdvihy
OSE[Z]=70	; koncová pozice = 70
ii2=2	; nastavení oblasti obratu
WAITP(Z)	; povolení oscilačního pohybu pro osu Z

2. Pohybové synchronní akce

Programový kód	Komentář
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z] DO -> \$AA_OVR[X]=0 \$AC_MARKER[0]=0	; vždy když je aktuální pozice osci- lační osy Z v MCS menší, než je za- čátek oblasti obratu 2, nastavit axiální korekci (override) přísuvné osy X na 0% a značku s indexem 0 na hodnotu 0

Programový kód	Komentář
WHENEVER \$AA_IM[Z]>=\$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[Z]=0	; vždy když je aktuální pozice oscilační osy Z v MCS větší nebo rovny poloze bodu obratu 2, nastaví se axiální korekce (override) oscilační osy Z na 0%
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[0]=1	; vždy když je zbytková dráha dílčího přísuvu rovna 0, nastavit značku s indexem 0 na hodnotu 1
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	; vždy když je značka s indexem 0 rovna 1, nastavit axiální korekci (override) přísuvné osy X na 0% Tím se zabrání příliš brzkému přísuvu (oscilační osa Z ještě oblast obratu 2 znovu neopustila, přísuvná osa X je ale připravena pro nový přísuv). Nastavení axiální korekce (override) oscilační osy Z z 0% (akce 2. synchronní akce) znovu na 100% pro další posuv.

-> musí být naprogramováno v jednom bloku

3. Spuštění oscilačního pohybu

Programový kód	Komentář
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; spuštění os Oscilační osa Z je přiřazena ose X jako přísuvné ose. Osa X má najíždět až do koncové pozice 5 v krocích po 1.
M30	; konec programu

Další informace

- Definice parametrů oscilačního pohybu**
Před pohybovým blokem, který obsahuje přiřazení přísuvné a oscilační osy, jakož i stanovení přísuvu, je zapotřebí definovat parametry pro oscilační pohyby (viz "Asynchronní oscilační pohyby").
- Definice pohybových synchronních akcí**
Prostřednictvím podmínek synchronních akcí se uskutečňuje:
Potlačení přísuvu, dokud se oscilační osa nachází uvnitř oblasti obratu (ii1, ii2) nebo v bodě obratu (U1, U2).
Pozastavení oscilačního pohybu v průběhu přísuvu v bodě obratu.
Spuštění oscilačního pohybu po skončení dílčího přísuvu.
Stanovení **spuštění následujícího dílčího přísuvu**.
- Přiřazení oscilační a přísuvné osy a definice celkového a dílčího přísuvu.**

Definice parametrů oscilačního pohybu

Přiřazení oscilační a přísluvné osy: OSCILL

OSCILL[<oscilační osa>]=(<přísluvná osa 1>,<přísluvná osa 2>,<přísluvná osa 3>)

Pomocí příkazu „OSCILL“ se přiřazují osy a spouští se oscilační pohyb.

Maximálně mohou být oscilační ose přiřazeny 3 přísluvné osy.

Poznámka

Před spuštěním oscilačního pohybu musí být definovány synchronizační podmínky pro chování os.

Definice přiřazení: POSP

POSP[<přísluvná osa>]=(<koncová pozice>,<dílčí délka>,<režim>)

Pomocí příkazu „POSP“ je řídicímu systému sdělováno:

- Celkový přísluv (přes koncovou pozici)
- Velikost jednotlivých dílčích přísluvů v bodě obratu, příp. v oblasti obratu.
- Chování dílčího přísluvu při dosažení koncové pozice (pomocí parametru "režim").

Režim = 0	Posledním dvěma dílčím přísluvům je přiřazena taková délka, aby zbývající dráha do cílového bodu byla rozdělena na 2 stejně velké úseky (předdefinované nastavení).
Režim = 1	Všechny dílčí přísluvy jsou stejně velké. Vypočítají se z celkového přísluvu.

Definice pohybových synchronních akcí

Pohybové synchronní akce popisované v následujících odstavcích se používají zcela obecně pro oscilační pohyby.

Představují vzorové postupy pro řešení jednotlivých požadavků, takže Vám mohou posloužit jako stavební kameny pro sestavování specifických uživatelských oscilačních pohybů.

Poznámka

V jednotlivých případech mohou být synchronizační podmínky naprogramovány i jinak.

Klíčová slova

WHEN ... DO ...	jestliže..., potom...
WHENEVER ... DO	vždycky když..., potom...

Funkce

Pomocí následujících podrobně popisovaných prostředků NC jazyka můžete realizovat následující:

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. Dílčí přísuv nespouštět příliš brzy.

Pro všechny synchronní akce uváděné v těchto příkladech platí následující předpoklady:

- Bod obratu 1 < bod obratu 2
- Z = oscilační osa
- X = přísuvná osa

Poznámka

Pokud budete potřebovat podrobnější vysvětlení, viz kapitola "Pohybové synchronní akce".

Přiřazení oscilační a přísuvné osy a definice celkového a dílčího přísuvu

Přísuv v oblasti obratu

Přísuvný pohyb by měl začínat v rámci oblasti obratu, předtím než je dosaženo bodu obratu.

Tyto synchronní akce blokují přísuvný pohyb, dokud se oscilační osa nachází v oblasti obratu.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

Oblast obratu 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+i1 DO $AA_OVR[X]=0
```

Vždy když je aktuální pozice oscilační osy v MCS větší, než je začátek oblasti obratu 1, nastaví se axiální korekce (override) přísuvné osy na 0%.

Oblast obratu 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+i2 DO $AA_OVR[X]=0
```

Vždy když je aktuální pozice oscilační osy v MCS menší, než je začátek oblasti obratu 2, nastaví se axiální korekce (override) přísuvné osy na 0%.

Přísuv v bodě obratu

Dokud oscilační osa nedosáhla bodu obratu, neuskutečňuje se žádný pohyb v přísuvné ose.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

Oblast obratu 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy Z v MCS větší nebo menší, než je poloha bodu obratu 1, potom se nastaví axiální korekce (override) přísluvné osy X na 0% a axiální korekce (override) oscilační osy Z na hodnotu 100%.

Oblast obratu 2:

Pro bod obratu 2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy Z v MCS větší nebo menší, než je poloha bodu obratu 2, potom se nastaví axiální korekce (override) přísluvné osy X na 0% a axiální korekce (override) oscilační osy Z na hodnotu 100%.

Pozastavení oscilačního pohybu v bodě obratu

Oscilační osa se v bodě obratu pozastaví a současně se spustí přísluvný pohyb. Oscilační pohyb bude pokračovat, až když je přísluvný pohyb zcela dokončen.

Současně se může tyto synchronní akce používat k tomu, aby se přísluvný pohyb spouštěl, jestliže byl zastaven předcházející synchronní akcí, která je stále ještě v platnosti.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

Oblast obratu 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy v MCS rovna poloze bodu obratu 1, nastaví se axiální korekce (override) oscilační osy na 0% a axiální korekce (override) přísluvné osy na 100%.

Oblast obratu 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy Z v MCS rovna poloze bodu obratu 2, nastaví se axiální korekce (override) oscilační osy X na 0% a axiální korekce (override) přísluvné osy na 100%.

On-line vyhodnocování bodu obratu

Jestliže se na pravé straně porovnávaného výrazu nachází proměnná hlavního zpracování označená znaky \$\$, jsou obě proměnné průběžně vyhodnocovány během taktu IPO a vzájemně porovnávány.

Poznámka

Pokud budete potřebovat další související informace, viz kapitola "Pohybové synchronní akce".

Opětovné spuštění oscilačního pohybu

Tyto synchronní akce se použije tehdy, když potřebujete, aby pohyb oscilační osy pokračoval, jakmile je dílčí přísluvný pohyb dokončen.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]=100
```

Vždy když je zbytková dráha pro dílčí pohyb přísluvné osy X v MCS rovna nule, potom se nastaví axiální korekce (override) oscilační osy na 100%.

Následující dílčí přísluv

Po dokončení přísluvu je nutno zajistit, aby se následující dílčí přísluv nespustil příliš brzy.

Pro tento účel se používá kanálová značka (\$AC_MARKER[Index]), která se na konci dílčího přísluvu (dílčí zbytková dráha $\equiv 0$) nastavuje a která se při opuštění oblasti obratu opět vymaže. Potom se pomocí synchronní akce zablokuje následující přísluvný pohyb.

Za uvedených předpokladů (viz výše) z toho vyplývají např. pro bod obratu 1 následující příkazy:

1. Nastavení značky:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1]=1
```

Vždy když je zbytková dráha pro dílčí pohyb přísluvné osy X v MCS rovna nule, potom se značka s indexem 1 nastaví na 1.

2. Vymazání značky

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1]=0
```

Vždy když je aktuální pozice oscilační osy Z v MCS větší nebo menší, než je poloha bodu obratu 1, nastaví se značka 1 na 0.

3. Blokování přísluvu

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

Vždy když je značka 1 rovna, nastaví se axiální korekce (override) přísluvné osy X na 0%.

Lisování a prostřihování

16.1 Aktivování/deaktivování

16.1.1 Aktivování/deaktivování lisování a prostřihování (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Aktivování/deaktivování lisování, příp. prostřihování

Lisování, příp. prostřihování se aktivují pomocí příkazů **PON** a **SON**. Příkazem **SPOF** se všechny specifické funkce pro lisování a prostřihování ukončí. Příkazy **PON** a **SON** s modální platností se vzájemně vylučují, tzn. příkaz **PON** deaktivuje příkaz **SON** a naopak.

Lisování/prostřihování s hlavičkou

Funkce pro lisování, příp. prostřihování se spouštějí také pomocí příkazů **SONS** a **PONS**.

Oproti příkazům **SON/PON**, u kterých se řízení zdvihu uskutečňuje na úrovni interpolace, probíhá u těchto funkcí řízení spouštění zdvihu na úrovni technických signálů servomechanismů. Díky tomu lze pracovat s vyšší frekvencí zdvihů a tedy i s vyšším výkonem při lisování.

V průběhu vyhodnocování signálů v hlavičce jsou blokovány všechny funkce, které mají za následek změnu polohy lisovacích nebo prostřihovacích os (např. manuální posuvy ručním kolečkem, změny framu pomocí PLC, měřicí funkce).

Lisování se zpožděním

Funkce **PDELAYON** způsobuje zpožděné spuštění lisovacího zdvihu. Tento příkaz s modální platností má přípravnou funkci a zpravidla se nachází před příkazem **PON**. Po příkazu **PDELAYOF** probíhá normální další lisování.

Poznámka

Doba prodlevy se nastavuje pomocí nastavovaného parametru **SD42400 \$SC_PUNCH_DWELLTIME**.

Zrychlení závisící na dráze

Pomocí příkazu **PUNCHACC** může být definována charakteristika zrychlení, která definuje různá zrychlení v závislosti na vzdálenosti díry.

Druhé rozhraní pro lisování

Stroje, které mají střídavě využívat druhé stanoviště s rozhraním (druhá lisovací jednotka nebo srovnatelné médium) mohou být přepínány na druhou dvojici rychlých digitálních vstupů a výstupů řídicího systému (V/V-pár). Obě stanoviště s rozhraním mohou využívat plné spektrum

funkcí pro lisování/prostřihování. Pro přepínání mezi prvním a druhým stanovištěm s rozhraním slouží příkazy SPIF1 a SPIF2.

Poznámka

Předpoklad: Prostřednictvím strojních parametrů musí být druhý pár V/V definován pro lisovací funkce (--> viz informace od výrobce stroje!).

Syntaxe

```
PON G... X... Y... Z...
SON G... X... Y... Z...
SONS G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYON
PDELAYOF
PUNCHACC (<Smin>, <Amin>, <Smax>, <Amax>)
SPIF1/SPIF2
SPOF
```

Význam

PON:	Aktivování lisování	
SON:	Aktivování prostřihování	
PONS:	Aktivování lisování s hlavičkou	
SONS:	Aktivování prostřihování s hlavičkou	
SPOF:	Deaktivování lisování/prostřihování	
PDELAYON:	Aktivování lisování se zpožděním	
PDELAYOF:	Deaktivování lisování se zpožděním	
PUNCHACC:	Aktivování zrychlení závisícího na dráze	
	Parametr:	
	<Smin>	Nejmenší vzdálenost od díry
	<Amin>	Počáteční zrychlení Parametr <Amin> může být větší než <Amax>.
	<Smax>	Největší vzdálenost od díry
	<Amax>	Koncové zrychlení Parametr <Amax> může být větší než <Amin>.
SPIF1:	Aktivování prvního rozhraní pro lisování Řízení zdvihu se uskutečňuje prostřednictvím první dvojice rychlých V/V.	
SPIF2:	Aktivování druhého rozhraní pro lisování Řízení zdvihu se uskutečňuje prostřednictvím druhé dvojice rychlých V/V.	
	Upozornění: Po resetu nebo po náběhu řídicího systému je aktivní vždy první rozhraní. Jestliže se pro lisování používá jen jedno rozhraní, nemusí být naprogramováno.	

Příklady

Příklad 1: Aktivování prostřihování

Programový kód	Komentář
...	
N70 X50 SPOF	; najíždění na polohu bez spuštění lisování
N80 X100 SON	; aktivování prostřihování, spuštění zdvihu před pohybem (X=50) a na konci naprogramovaného pohybu (X=100)
...	

Příklad 2: Lisování se zpožděním

Programový kód	Komentář
...	
N170 PDELAYON X100 SPOF	; najíždění na polohu bez spuštění lisování, aktivování zpožděného spuštění lisování
N180 X800 PON	; aktivování lisování Po dosažení koncové pozice se uskuteční zpožděný lisovací zdvih.
N190 PDELAYOF X700	; deaktivování lisování se zpožděním, normální spuštění lisování na konci naprogramovaného pohybu
...	

Příklad 3: Lisování se dvěma stanovišti s rozhraním

Programový kód	Komentář
...	
N170 SPIF1 X100 PON	; na konci bloku se uskutečňuje spuštění zdvihu na prvním rychlém výstupu Na prvním rychlém vstupu je monitorován signál "Zdvih aktivní".
N180 X800 SPIF2	; druhý zdvih se spouští na druhém rychlém výstupu Na druhém rychlém vstupu je monitorován signál "Zdvih aktivní".
N190 SPIF1 X700	; řízení zdvihu se pro všechny další zdvihy uskutečňuje na prvním rozhraní
...	

Další informace

Lisování a prostřihování s hlavičkou (PONS/SONS)

Lisování a prostřihování s hlavičkou není možné provádět ve více kanálech současně. Příkazy PONS, příp. SONS mohou být aktivovány vždy jen v jednom kanálu.

Aktivování zrychlení závisícího na dráze (PUNCHACC)

Příklad:

PUNCHACC (2, 50, 10, 100)

Vzdálenost od díry menší než 2 mm:

Pohyb bude probíhat se zrychlením 50% maximálního zrychlení.

Vzdálenost od díry 2 mm až 10 mm:

Zrychlení se bude přímo úměrně vzdálenosti zvyšovat až na 100%.

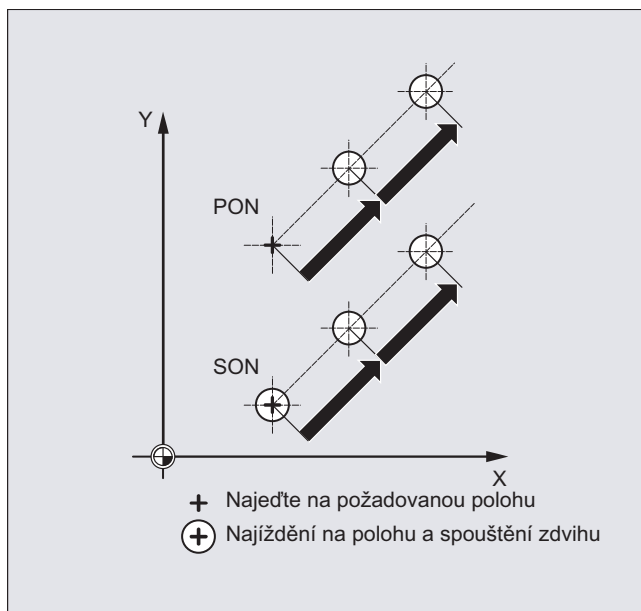
Vzdálenost od díry větší než 10 mm:

Pohyb se zrychlením 100%.

Spouštění prvního zdvihu

Spouštění prvního zdvihu po aktivování funkce se uskutečňuje při prostřihování časově odlišně než při lisování.

- PON/ PONS:
 - Všechny zdvihy - i zdvih prvního bloku po aktivování - probíhají na konci bloku.
- SON/ SONS:
 - První zdvih po aktivování prostřihování se uskuteční už na začátku bloku.
 - Všechny ostatní zdvihy jsou však spouštěny až na konci bloku.



Lisování a prostřihování na místě

Ke spuštění zdvihu dojde jen tehdy, pokud blok obsahuje informace o pohybu lisovacích nebo prostřihovacích os (osy aktivní roviny).

Aby se přesto spustil zdvih na stejném místě, naprogramují se lisovací/prostřihovací osy s délkou dráhy 0.

Práce s otočnými nástroji

Poznámka

Aby byly otočné nástroje nastavovány tangenciálně vůči naprogramované dráze, používejte tangenciální řízení.

Použití příkazů M-funkcí

Pomocí techniky maker je stejně jako dříve možné používat speciální M-funkce namísto příkazů NC jazyka (kompatibilita). Vzhledem ke starším systémům přitom platí následující konvence:

M20, M23	△	SPOF
M22	△	SON
M25	△	PON
M26	△	PDELAYON

Příklad souboru makra:

Programový kód	Komentář
DEFINE M25 AS PON	; zapnutí lisování
DEFINE M125 AS PONS	; aktivování lisování s hlavičkou
DEFINE M22 AS SON	; aktivování prostřihování
DEFINE M122 AS SONS	; aktivování prostřihování s hlavičkou
DEFINE M26 AS PDELAYON	; aktivování lisování se zpožděním
DEFINE M20 AS SPOF	; lisování, prostřihování deaktivovat
DEFINE M23 AS SPOF	; lisování, prostřihování deaktivovat

Příklad programování:

Programový kód	Komentář
...	
N100 X100 M20	; najíždění na polohu bez spuštění lisování
N110 X120 M22	; aktivování prostřihování, spuštění zdvihu před a po pohybu
N120 X150 Y150 M25	; aktivování lisování, spuštění zdvihu na konci pohybu
...	

16.2 Automatické rozdělení dráhy

Rozčlenění na dílčí úseky

Když je aktivní lisování, příp. prostřihování, zajišťují příkazy SPP a SPN rozčlenění celkové dráhy posuvu, která byla naprogramována pro dráhové osy, na daný počet stejně dlouhých dílčích úseků (ekvidistantní rozdělení dráhy). Interně odpovídá každému dílčímu úseku jeden blok.

Počet zdvihů

Při lisování se první zdvih uskuteční na konci prvního dílčího úseku, při prostřihování oproti tomu v počátečním bodě prvního dílčího úseku. V důsledku toho na celkové dráze platí následující počty:

Lisování: Počet zdvihů = počet dílčích úseků

Prostřihování: Počet zdvihů = počet dílčích úseků + 1

Pomocné funkce

Pomocné funkce jsou zpracovávány v prvním vygenerovaném úseku.

Syntaxe

SPP=

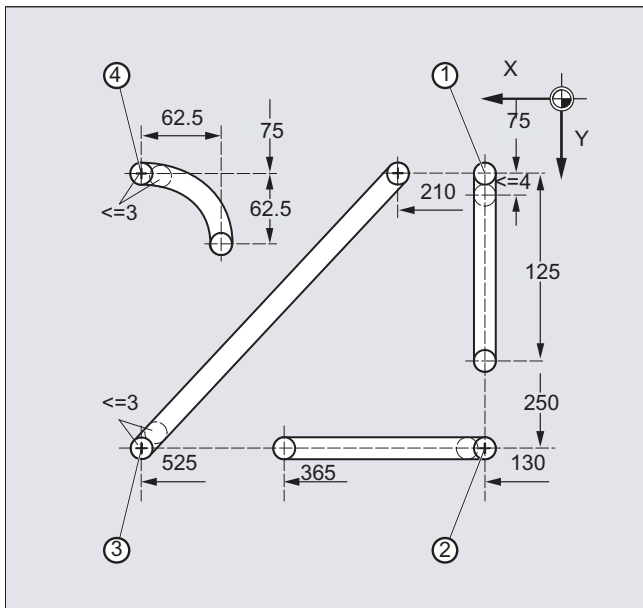
SPN=

Význam

SPP:	Velikost dílčího úseku (maximální vzdálenost zdvihů), modální platnost
SPN:	Počet úseků na blok, bloková platnost

Příklad 1

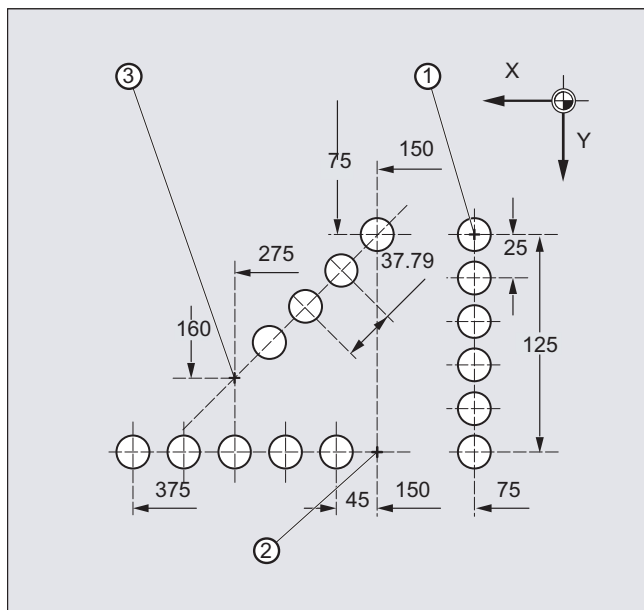
Naprogramované úseky pro prostřihování mají být automaticky rozděleny na stejně velké úseky.



Programový kód	Komentář
N100 G90 X130 Y75 F60 SPOF	; najíždění na polohu počátečního bodu 1
N110 G91 Y125 SPP=4 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 4 mm
N120 G90 Y250 SPOF	; vypnutí prostřihování, najíždění na pozici počátečního bodu 2
N130 X365 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 4 mm
N140 X525 SPOF	; vypnutí prostřihování, najíždění na pozici počátečního bodu 3
N150 X210 Y75 SPP=3 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 3 mm
N160 X525 SPOF	; vypnutí prostřihování, najíždění na pozici počátečního bodu 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 3 mm
N180 G00 G90 Y300 SPOF	; vypnutí prostřihování

Příklad 2

Pro jednotlivé řady děr má být provedeno automatické rozdělení dráhy na úseky. Pro rozdělení se vždy zadává maximální délka úseku dráhy (hodnota SPP).



Programový kód	Komentář
N100 G90 X75 Y75 F60 PON	; najíždění na počáteční bod 1, lisování jednotlivé díry
N110 G91 Y125 SPP=25	; maximální délka dílčího úseku pro automatické rozdělení dráhy: 25 mm
N120 G90 X150 SPOF	; vypnutí lisování, najíždění na pozici počátečního bodu 2
N130 X375 SPP=45 PON	; aktivování lisování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 45 mm
N140 X275 Y160 SPOF	; vypnutí lisování, najíždění na pozici počátečního bodu 3
N150 X150 Y75 SPP=40 PON	; aktivování lisování, namísto naprogramované délky dílčího úseku 40 mm se použije vypočítaná délka dílčího úseku 37,79 mm
N160 G00 Y300 SPOF	; vypnutí lisování, najíždění na danou pozici

16.2.1 Rozdělení cesty u dráhových os

Délka dílčího úseku SPP

Pomocí příkazu SPP zadáváte maximální vzdálenost zdvihů a v důsledku toho maximální délku dílčího úseku, na které má být celková dráha rozdělena. Pro deaktivování této funkce slouží příkazy SPOF nebo SPP=0.

Příklad:

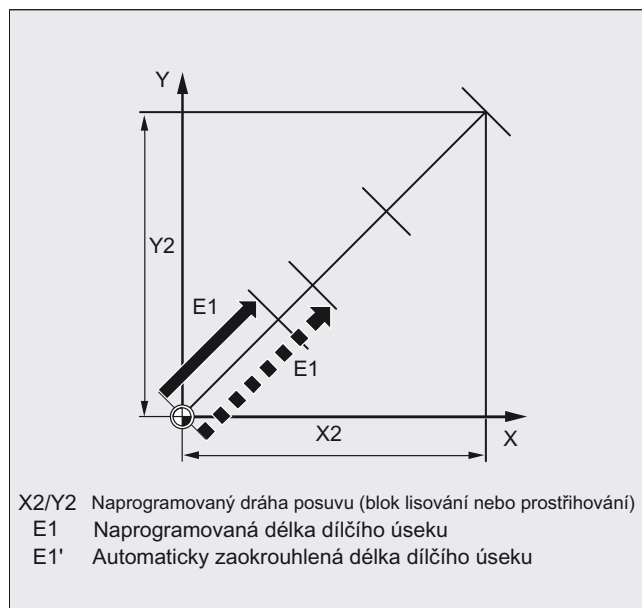
```
N10 SON X0 Y0
```

```
N20 SPP=2 X10
```

Celková dráha 10 mm se rozdělí na 5 dílčích úseků každý o délce 2 mm ($SPP=2$).

Poznámka

Rozdělení dráhy pomocí příkazu `SPP` je vždy ekvidistanční: všechny dílčí úseky jsou stejně dlouhé. To znamená, že naprogramovaná velikost dílčího úseku (hodnota `SPP`) platí jen tehdy, pokud je podíl celkové délky dráhy a hodnoty `SPP` celé číslo. Jestliže tomu tak není, interně se velikost dílčího úseku zmenší natolik, aby byl podíl celé číslo.



Příklad:

```
N10 G1 G91 SON X10 Y10
```

```
N20 SPP=3.5 X15 Y15
```

V případě délky celé dráhy 15 mm a délky dílčích úseků 3,5 mm vyplývá hodnota podílu, která není celočíselná (4,28). V důsledku toho systém hodnotu `SPP` zmenší, aby vyšel nejbližší možný celočíselný podíl. V tomto případě vyjde délka dílčího úseku 3 mm.

Počet dílčích úseků SPN

Pomocí příkazu `SPN` definujete počet dílčích úseků, které mají být z celkové délky dráhy vytvořeny. Délka dílčího úseku se vypočítá automaticky. Protože příkaz `SPN` má blokovou platnost, musí být předtím pomocí příkazů `PON` nebo `SON` aktivováno lisování nebo prostřihování.

16.2.2 Rozdělení cesty u jednotlivých os

Jestliže jsou vedle dráhových os také samostatné osy definovány jako osy pro lisování/prostřihování, je možné i na tyto osy aplikovat automatické rozdělení dráhy.

Chování samostatné osy v případě SPP

Naprogramovaná délka dílčího úseku (SPP) se v zásadě vztahuje na dráhové osy. Proto je v bloku, ve kterém kromě pohybu samostatnou osou a hodnoty příkazu SPP není naprogramována žádná dráhová osa, je hodnota SPP ignorována.

Jestliže jsou v bloku naprogramovány jak samostatné osy, tak i dráhové osy, řídí se chování samostatné osy podle nastavení odpovídajícího strojního parametru.

1. Standardní nastavení

Dráha samostatné osy je prostřednictvím pomocných bloků vygenerovaných příkazem SPP rozdělena na stejné úseky.

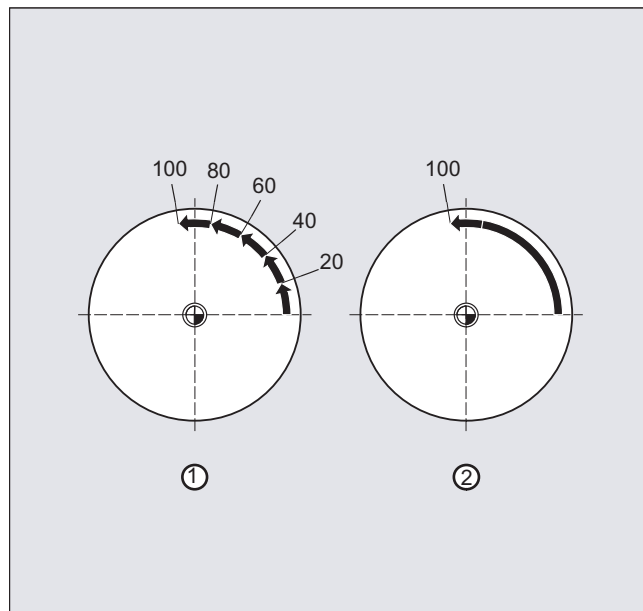
Příklad:

```
N10 G1 SON X10 A0
```

```
N20 SPP=3 X25 A100
```

Na základě zdvihové vzdálenosti 3 mm se v případě celkové délky posuvu osy X (dráhová osa) 15 mm vytvoří 5 bloků.

Osa A se tedy v každém bloku otočí o 20° .



1. Samostatná osa bez rozdělení dráhy

Samostatná osa urazí svou celkovou dráhu v rámci prvního z vytvořených bloků.

2. Odlišné rozdělení dráhy

Chování samostatné osy je závislé na interpolaci dráhových os.

- Kruhová interpolace: Rozdělení dráhy
- Lineární interpolace: Žádné rozdělení dráhy

Chování v případě SPN

Naprogramovaný počet dílčích úseků platí také tehdy, pokud dráhová osa není naprogramována současně.

Předpoklad: Samostatná osa je definována jako osa pro prostřihování/lisování.

17.1 Aktivování/deaktivování speciálního monitorování pro brusné nástroje (TMON, TMOF)

Prostřednictvím předem definovaných procedur TMON(...) a TMOF(...) je možno aktivovat, příp. deaktivovat specifické monitorování pro brusné nástroje (monitorování geometrie a otáček).

Předpoklady

Musí být nastaveny parametry specifického nástroje \$TC_TPG1 až \$TC_TPG9.

Syntaxe

```
TMON (<Tnr>)
...
TMOF (<Tnr>)
```

Význam

TMON (. . .) :	Aktivování specifického monitorování pro brusné nástroje Příkaz musí být naprogramován v kanálu, v němž má být specifické monitorování pro brusní nástroje aktivováno.
TMOF (. . .) :	Deaktivování specifického monitorování pro brusné nástroje Příkaz musí být naprogramován v kanálu, v němž má být specifické monitorování pro brusní nástroje deaktivováno.
<Tnr>:	T-číslo Poznámka: Je zapotřebí jen tehdy, pokud má být monitorování aktivováno, příp. deaktivováno pro neaktivní brusný kotouč a nikoli na aktivní nástroj, který je momentálně používán.
TMOF (0) :	Deaktivování monitorování pro všechny nástroje

Další funkce

18.1 Osové funkce (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

Příkaz „AXNAME“ se používá např. při sestavování všeobecně platných cyklů, pokud názvy os nejsou známy.

Příkaz „AX“ se používá pro nepřímé programování geometrických a synchronních os. Identifikátor osy je přitom uložen v proměnné typu AXIS nebo je zjištěn pomocí příkazu, jako např. „AXNAME“ nebo „SPI“.

Příkaz „SPI“ se použije, jestliže se programují funkce osy pro vřetenno, např. pro synchronní vřetenno.

Příkaz „AXTOSPI“ slouží v situacích, kdy potřebujete identifikátor osy převést na index vřetenno (inverzní funkce k „SPI“).

Příkaz „AXSTRING“ slouží v situacích, kdy potřebujete identifikátor osy (datový typ AXIS) převést na řetězec (inverzní funkce k „AXNAME“).

Příkaz „ISAXIS“ se používá ve všeobecně platných cyklech, aby se zjistilo, že určitá geometrická osa je k dispozici a aby následující volání proměnné \$P_AXNX nebylo přerušeno s chybou.

Funkce „MODAXVAL“ se používá pro zjišťování polohy kruhových os typu Modulo po zpracování funkcí Modulo.

Syntaxe

```
AXNAME ("řetězec")
AX[AXNAME ("řetězec")]
SPI (n)

AXTOSPI (A) nebo AXTOSPI (B) nebo AXTOSPI (C)
AXSTRING (SPI (n))
ISAXIS (<číslo geometrické osy>)
<pozice modulo>=MODAXVAL (<osa>, <poloha osy>)
```

Význam

AXNAME:	Převádí vstupní řetězec na identifikátor osy, vstupní řetězec musí obsahovat platný název osy.
AX:	Proměnný identifikátor osy
SPI:	Převádí číslo vřetenno na identifikátor osy; předávaný parametr musí obsahovat platné číslo vřetenno.
n:	Číslo vřetenno
AXTOSPI:	Převádí identifikátor osy na index vřetenno typu INTEGER. Příkaz "AXTOSPI" odpovídá inverzní funkci k příkazu „SPI“.
X, Y, Z:	Identifikátor osy typu AXIS jako proměnná nebo jako konstanta.

AXSTRING:	Na výstupu bude řetězec, jemuž je přiřazeno číslo vřetena.
ISAXIS:	Zkontroluje, zda je uvedená geometrická osa k dispozici.
MODAXVAL:	Zjišťuje polohu kruhových os typu Modulo po zpracování funkcí Modulo, tato hodnota odpovídá zbytku po celočíselném dělení vzhledem k rozsahu Modulo (v případě standardního nastavení je to 0° až 360°, pomocí strojních parametrů MD30340 MODULO_RANGE_START a MD30330 \$MA_MODULO_RANGE mohou být začátek s velikost rozsahu Modulo změněny).

Poznámka

Rozšíření SPI

Osová funkce SPI(n) může být použita také pro čtení a zápis komponentů framu. Tímto způsobem je možné zapisovat údaje do framu, např. pomocí syntaxe

\$P_PFRAME[SPI(1), TR]=2.22.

Prostřednictvím doplňkového naprogramování pozice os pomocí adresy

AX[SPI(1)]=<pozice osy> je možné osou pohybovat. Předpokladem pro tuto operaci ale je, že se vřeteno nachází v režimu polohování nebo v režimu osy.

Příklady

Příklad 1: AXNAME, AX, ISAXIS

Programový kód	Komentář
OVRA[AXNAME("Planachse")]=10	; korekce (override) pro příčnou osu
AX[AXNAME("Planachse")]=50.2	; koncová pozice pro příčnou osu
OVRA[SPI(1)]=70	; korekce (override) pro vřeteno 1
AX[SPI(1)]=180	; koncová pozice pro vřeteno 1
IF ISAXIS(1) == FALSE GOTOF WEITER	; Je k dispozici abscisa?
AX[\$P_AXN1]=100	; pohyb abscisy
WEITER:	

Příklad 2: AXSTRING

Při programování příkazu AXSTRING[SPI(n)] se nezadáva index osy pro osu, která je přiřazena vřetenu, jako číslo vřetena, nýbrž místo toho se zadává řetězec "Sn".

Programový kód	Komentář
AXSTRING[SPI(2)]	; na výstupu bude řetězec "S2"

Příklad 3: MODAXVAL

Tímto způsobem se má zjistit poloha kruhové osy A typu Modulo po zpracování funkcí modulo. Výchozí hodnota pro výpočet je poloha osy 372,55.

Rozsah Modulo nastavený v parametrech je 0 až 360 stupňů:

MD30340 MODULO_RANGE_START = 0

18.1 Osové funkce (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

MD30330 \$MA_MODULO_RANGE = 360

Programový kód	Komentář
R10=MODAXVAL(A,372.55)	; vypočítaná pozice Modulo je R10 = 12,55

Příklad 4: MODAXVAL

Jestliže se naprogramovaný identifikátor osy nevztahuje na kruhovou osu typu Modulo, potom bude převáděná hodnota (<poloha osy>) vrácena zpět nezměněná.

Programový kód	Komentář
R11=MODAXVAL(X,372.55)	; X je lineární osa; R11 = 372,55

18.2 Přepínatelné geometrické osy (GEOAX)

Pomocí funkce "Přepínatelné geometrické osy" můžete geometrické osy nastavené v konfiguraci pomocí strojních parametrů vyměnit za jiné kanálové osy.

Syntaxe

```
GEOAX (<n>, <kanálová osa>, <n>, <kanálová osa>, <n>, <kanálová osa>)
GEOAX ()
```

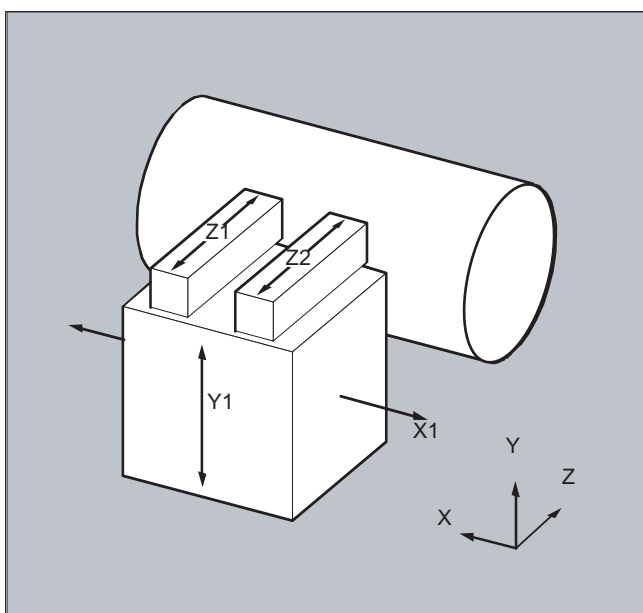
Význam

GEOAX (. . .)	Funkce pro přepnutí geometrických os. Poznámka: GEOAX) bez udání parametrů znovu aktivuje základní konfiguraci geometrických os nastavenou ve strojních parametrech.
<n>	Číslo geometrické osy, která má být nahrazena za uvedenou kanálovou osu. Rozsah hodnot: 0, 1, 2, 3 Poznámka: 0: Uvedená kanálová osa bude bez náhrady odstraněna ze svazku geometrických os 1: 1. Geometrická osa $\triangleq$ Souřadná osa X (abscisa) WCS 2: 2. Geometrická osa $\triangleq$ Souřadná osa Y (ordináta) WCS 3: 3. Geometrická osa $\triangleq$ Souřadná osa Z (aplikáta) WCS
<kanálová osa>	Název kanálové osy, která má být přijata do svazku geometrických os.

Příklady

Příklad 1: Střídavé přepínání dvou os na místě jedné geometrické osy

Nástrojovými saněmi je možné pohybovat pomocí kanálových os X1, Y1, Z1 a Z2.



Konfigurace geometrických os je taková, že po zapnutí systému se napřed jako platnou 3. geometrickou osou pod názvem geometrické osy "Z" stává osa Z1 a spolu s osami X1 a Y1 vytváří svazek geometrických os.

Ve výrobním programu se nyní mají osy Z1 a Z2 střídavě používat jako geometrická osa Z.

Programový kód	Komentář
...	
N100 GEOAX(3,Z2)	; jako 3. geometrická osa (Z) funguje kanálová osa Z2
N110 G1 ...	
N120 GEOAX(3,Z1)	; jako 3. geometrická osa (Z) funguje kanálová osa Z1
...	

Příklad 2: Přepínání geometrických os v případě 6 kanálových os

Stroj je vybaven 6 kanálovými osami 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)

Programový kód	Komentář
N10 GEOAX()	; základní konfigurace geometrických je v platnosti
N20 G0 X0 Y0 Z0 U0 V0 W0	; všechny osy rychlým posuvem na pozici 0
N30 GEOAX(1,U,2,V,3,W)	; přiřazení kanálové osy U první (X), osy V druhé (Y) ; a osy W třetí geometrické ose (Z)
N40 GEOAX(1,XX,3,ZZ)	; Přiřazení kanálové osy XX první (X) a osy ZZ třetí ; geometrické ose (Z). Kanálová osa V zůstává druhou ; geometrickou osou (Y).

18.2 Přepínatelné geometrické osy (GEOAX)

Programový kód	Komentář
N50 G17 G2 X20 I10 F1000	; Plná kružnice v rovině X/Y. Pohybují se ; kanálové osy XX a V.
N60 GEOAX(2,W)	; kanálová osa W se stává druhou geometrickou osou (Y)
N80 G17 G2 X20 I10 F1000	; Plná kružnice v rovině X/Y. Pohybují se ; kanálové osy XX a W.
N90 GEOAX()	; obnovení původního stavu
N100 GEOAX(1,U,2,V,3,W)	; přiřazení kanálové osy U první (X), osy V druhé ; (Y) a osy W třetí geometrické ose (Z)
N110 G1 X10 Y10 Z10 XX=25	; Kanálové osy U, V a W najíždějí na své ; pozice 10. Osa XX jako přísuvná osa najíždí na po- zici 25.
N120 GEOAX(0,V)	; osa V je vyjmuta ze svazku geometrických os ; osy U a W jsou nadále první (X) a třetí ; geometrickou osou (Z) ; druhá geometrická osa (Y) zůstává neobsazena
N130 GEOAX(1,U,2,V,3,W)	; kanálová osa U zůstává přiřazena první (X), osa V ; druhé (Y) a osa W třetí geometrické ose (Z)
N140 GEOAX(3,V)	; osa V se stává třetí geometrickou osou (Z), přičemž přepisuje osu W ; a tím pádem ji vyjímá ; ze svazku geometrických os ; druhá geometrická osa (Y) ; je stejně jako předtím neobsazena.

Strojní parametry

Konfigurace osy

Přiřazení geometrických, doplňkových a strojních os kanálovým osám:

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB
- MD20070 \$MC_AXCONF_MACHAX_USED
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX

Chování při resetu

Chování změněného přiřazení geometrických os při resetu:

- MD20110 \$MC_RESET_MODE_MASK, Bit 12
- MD20118 \$MC_GEOAX_CHANGE_RESET

Chování při spuštění NC systému

- MD20112 \$MC_START_MODE_MASK, Bit 12

Předávání uživatelskému programu PLC

Možnost nastavení parametrů příkazu M-funkce, která je vysílána na rozhraní NC-systému/ PLC při přepnutí geometrických os:

- MD22532 \$MC_GEOAX_CHANGE_M_CODE

Okrajové podmínky**Žádné přepínání geometrických os**

- Pokud je aktivní některá z následujících funkcí, přepínání geometrických os není možné:
 - Transformace
 - Interpolace spliny
 - Korekce rádiusu nástroje
 - Jemná korekce nástroje
- Geometrická osa a další kanálová osa mají stejný Název.
- Jedna z os, které se podílejí na operaci přepnutí geometrických os, je zapojena do akce, která svým trváním přesahuje hranice bloku. Např. polohovací osa, jejíž pohyb trvá přes hranice bloku, nebo vlečná osa začleněná do vazby mezi osami.

Kruhové osy

Kruhové osy se nemohou stát geometrickými osami.

Stav os po nahrazení

Osa, která byla ve svazku geometrických os nahrazena pomocí přepnutí, může být po přepnutí naprogramována jako doplňková osa pomocí svého kanálového názvu.

Framy, chráněné oblasti, ohraničení pracovního pole

Přepnutím geometrických os jsou všechny framy, chráněné oblasti a ohraničení pracovního pole vymazány.

Polární souřadnice

Výměna geometrických os pomocí příkazu `GEOAX` nastavuje analogicky se změnou roviny pomocí příkazů `G17-G19` modální polární souřadnice na hodnotu 0.

DRF, NPV

Případná posunutí pomocí ručního kolečka (DRF) nebo externí posunutí počátku (NPV) zůstávají i po přepnutí v platnosti.

Základní konfigurace geometrických os

Příkaz `GEOAX ( )` vyvolá základní konfiguraci svazku geometrických os.

Po náběhu systému (Power-On) a při přepnutí do provozního režimu "Najíždění na referenční bod" se automaticky nastavuje základní konfigurace.

Korekce délky nástroje

Aktivní korekce délky nástroje je v platnosti i po operaci přepnutí. Platí však pro geometrické osy, které byly nově připojeny, příp. které si vyměnili pozici, jako hodnota, na kterou se zatím nenajelo. Při prvním pohybovém příkazu pro tyto geometrické osy vznikne výsledná dráha posuvu, která je odpovídajícím způsobem složena z korekce délky nástroje a naprogramované dráhy posuvu.

Geometrické osy, které při přepnutí své pozice zůstaly ve svazku os zachovány, si ponechávají i svůj stav týkající se korekce délky nástroje.

Konfigurace geometrických os při aktivní transformaci

- Konfigurace geometrických os, která je pro aktivní transformaci definována pomocí strojních parametrů transformací, nemůže být pomocí funkce "Přepínatelné geometrické osy" změněna.
- Pro odlišné konfigurace geometrických os pro transformace musí být ve strojních parametrech transformací vytvořeny různé datové bloky.
- Konfigurace geometrických os změněná pomocí funkce `GEOAX` se aktivováním transformace vymaže.
- Pokud jde o geometrické osy, mají nastavení parametrů těchto geometrických os vztahující se ke specifické aktivní transformaci přednost před nastaveními parametrů, která se týkají přepínání geometrických os.
Příklad: Je aktivní transformace. Podle strojních parametrů má tato transformace při resetu kanálu zůstat zachována. Současně však ale při resetu kanálu má být obnovena původní základní konfigurace geometrických os. Zůstane zachována konfigurace geometrických os, která byla definována pro transformaci.
- Po deaktivování transformace bude opět v platnosti základní nastavení konfigurace geometrických os stanovené v parametrech.

Provozní režim JOG, funkce stroje REF

Když přepnete do provozního režimu JOG a spustíte strojní funkci REF (najíždění na referenční bod), vstoupí v platnost konfigurace geometrických os nastavená ve strojních parametrech.

18.3 Osový zásobník (AXCTSWE, AXCTSWED, AXCTSWEC)

Příkaz „AXCTSWE“, příp. „AXCTSWED“ provádí odblokování otáčení uvedeného osového zásobníku.

Pomocí příkazu „AXCTSWEC“ může být již aktivované uvolnění pro otáčení osového zásobníku opět odvoláno.

Syntaxe

AXCTSWE (<ID>)
 AXCTSWED (<ID>)
 AXCTSWEC (<ID>)

Význam

AXCTSWE:	Uvolnění otáčení osového zásobníku Zpracovávání programu není příkazem „AXCTSWE“ pozastaveno. Otočení se uskuteční, jakmile je všem kanálům podílejícím se na osovém zásobníku poskytnuto uvolnění.	
AXCTSWED:	Uvolnění osového zásobníku pro otáčení bez ohledu na jiné kanály začleněné do osového zásobníku Upozornění • Varianta příkazu určená pro zjednodušení uvádění do provozu výrobního programu, příp. synchronní akce. • Chování v souvislosti s jinými kanály začleněnými do osového zásobníku může být předem nastaveno pomocí strojního parametru: MD12760 \$MN_AXCT_FUNCTION_MASK, Bit 0	
AXCTSWEC:	Deaktivování uvolnění otáčení osového zásobníku Upozornění: Uvolnění osového zásobníku za účelem otáčení může být deaktivováno jen tehdy, pokud toto otáčení ještě nezačalo. \$AN_AXCTSWA[<osový zásobník>] == 0 Pokud potřebujete informace o systémových proměnných, viz "Osový zásobník (AXCTSWE, AXCTSWED, AXCTSWEC) (Strana 657)"	
<ID>:	Identifikátor osového zásobníku nebo osy ze zásobníku:	
	CT<Číslo>:	Předdefinovaný identifikátor osového zásobníku: MD12750 \$MN_AXCT_NAME_TAB Příklad: "CT1"
	<Zásobník>:	Uživatelský identifikátor osového zásobníku: MD12750 \$MN_AXCT_NAME_TAB Příklad: "CONTAINER_1"
	<osa>:	Identifikátor osy ze zásobníku, která je v kanálu známá

Poznámka

Velikosti kroku

Velikost kroku při otáčení osového zásobníku se nastavuje pomocí nastavovaného parametru:

SD41700 \$SN_AXCT_SWWIDTH

Další informace

Diagnostika

Aktuální stavové informace o osovém zásobníku je možné načítat pomocí následující systémové proměnné:

Systémové proměnné	Typ	Popis
\$AC_AXCTSWA[<Název>]	BOOL	Status osového zásobníku specifického kanálu
\$AN_AXCTSWA[<osový zásobník>]	BOOL	Status osového zásobníku specifické NCU
\$AN_AXCTSWE[<osový zásobník>]	INT	Status otáčení osového zásobníku specifického slotu Tato systémová proměnná poskytuje informace o stavu slotů osového zásobníku prostřednictvím jednotlivých bitů . Každý bit odpovídá jednomu slotu.
\$AN_AXCTAS[<osový zásobník>]	INT	Počet míst (slotů), o které byl osový zásobník momentálně pootočený.

Otočení osového zásobníku s implicitními příkazy GET/GETD

Prostřednictvím následujícího strojního parametru může být nastaveno, že pomocí příkazu „AXCTSWE“ jsou všechny zásobníkové osy kanálu převedeny prostřednictvím implicitních příkazů "GET" / „GETD" do daného kanálu. Výměna osy je opět možná teprve po úspěšném otočení zásobníku.

MD10722 \$MN_AXCHANGE_MASK, Bit 1 = 1

Poznámka

Otočení osového zásobníku pomocí implicitních příkazů „GET“ / „GETD“ **nebude** pro osu nacházející se ve stavu "osy hlavního zpracování" (např. pro osu PLC) provedeno, protože osa by potom musela tento stav kvůli otáčení osového zásobníku opustit.

18.4 Čekání na platnou pozici osy (WAITENC)

Pomocí příkazu NC jazyka „WAITENC“ je možné v NC programu počkat, dokud pro osy nastavené v konfiguraci pomocí parametru MD34800 \$MA_WAIT_ENC_VALID = 1 nebude k dispozici synchronizovaná, příp. obnovená poloha osy.

V čekacím stavu může dojít k přerušení, např. v důsledku spuštění programu ASUP nebo přepnutí do provozního režimu JOG. Při pokračování programu bude čekací stav v případě potřeby opět obnoven.

Poznámka

Na uživatelském rozhraní se čekací stav zobrazuje prostřednictvím stavového hlášení o čekání "Čekání na měřicí systém".

Syntaxe

Příkaz „WAITENC“ může být naprogramován v programové části libovolného NC programu.

Programování se musí provádět v samostatném bloku:

```
...
WAITENC
...
```

Příklad:

Příkaz „WAITENC“ se používá např. v uživatelském programu .../_N_CMA_DIR/_N_PROG_EVENT_SPF řízeném prostřednictvím událostí, jak ukazuje následující příklad použití.

Příklad použití: Zpětný pohyb nástroje po POWER-OFF s transformací orientace

Obrábění s orientací nástroje bylo v důsledku výpadku proudu přerušeno.

Při následujícím opětovném náběhu systému se vyvolává uživatelský program .../_N_CMA_DIR/_N_PROG_EVENT_SPF řízený na základě událostí.

V uživatelském programu řízeném na základě událostí se pomocí příkazu „WAITENC“ čeká na synchronizovanou, příp. obnovenou polohu os, aby potom bylo možné vypočítat frame, který nasměruje WCS podle směru nástroje.

Programový kód	Komentář
...	
IF \$P_PROG_EVENT == 4	; náběh systému
IF \$P_TRAFO <> 0	; transformace byla deaktivována
WAITENC	; čekání na platné pozice orientačních os
TOROTZ	; otočení osy Z systému WCS ve směru osy nástroje
ENDIF	
M17	
ENDIF	
...	

18.4 Čekání na platnou pozici osy (WAITENC)

Potom může nástroj v provozním režimu JOG volně vyjet pomocí zpětného pohybu ve směru osy nástroje.

18.5 Programovatelné přepnutí sady parametrů (SCPARA)

Pomocí příkazu „SCPARA“ je možno si pro danou osu vyžádat přepnutí na určitou sadu parametrů.

Poznámka

Během opracovávání závitu není přepnutí na jinou sadu parametrů možné

Při řezání závitů pomocí příkazu G33 a vrtání závitu pomocí příkazu G331 / G332 je sada parametrů vybírána řídicím systémem a není možná ji změnit.

Zablokované přepnutí sady parametrů

Přepnutí sady parametrů je možné si vyžádat také pomocí rozhraní NC/PLC. Aby se zabránilo konfliktům, které by při přepínání mohly vzniknout, je možné pomocí rozhraní NC/PLC přepínání sady parametrů NC systému (SCPARA) zablokovat.

DB31, ... DBX9.3 (zadaná sada parametrů prostřednictvím NC systému blokována)

Poznámka

Pokud je pomocí příkazu „SCPARA“ vyžádáno přepnutí sady parametrů, zatímco je přepínání sady parametrů pomocí rozhraní NC/PLC zablokováno, přepnutí se neuskuteční, aniž by se vypisovalo chybové hlášení.

Syntaxe

SCPARA [<osa>] = <hodnota>

Význam

SCPARA:	Příkaz: Přepnutí sady parametrů	
<osa>:	Identifikátor osy (kanálová osa)	
	Typ:	AXIS
<hodnota>:	Číslo sady parametr: 1, 2, 3, ... max. číslo sady parametrů	

Příklad:

Programový kód	Komentář
...	
N110 SCPARA[X]= 3	; aktivování: Osa X, 3. sada parametrů
...	

Další informace

Odblokování přepnutí sady parametrů

Přepínání sady parametrů pro danou osu musí být explicitně odblokováno:

MD35590 \$MA_PARAMSET_CHANGE_ENABLE[<osa>]

Načtení čísla sady parametrů

Číslo zvolené sady parametrů (požadovaná sada parametrů) je možné načíst prostřednictvím systémové proměnné \$AA_SCPAR.

Literatura

Pokud budete potřebovat podrobné informace týkající se sad parametrů, viz:

Příručka k funkcím, Základní funkce; kapitola "Rychlosti, systém požadovaných a skutečných hodnot, regulace (G2)" > "regulace" > "Sady parametrů regulátoru polohy".

18.6 Kontrola existujícího rozsahu NC jazyka (STRINGIS)

Pomocí funkce "STRINGIS(...)" je možné zkontrolovat, zda je zadaný řetězec k dispozici jako prvek aktuálního rozsahu programovacího jazyka NC systému.

Definice

```
INT STRINGIS (STRING <název>)
```

Syntaxe

```
STRINGIS (<název>)
```

Význam

STRINGIS:	Funkce s výstupní hodnotou
<název>:	Název zkoumaného prvku programovacího jazyka NC systému
Výsledná hodnota:	Výsledná hodnota má formát yxx (desetinný formát).

Prvky programovacího jazyka NC systému

Mohou být kontrolovány následující prvky programovacího jazyka NC systému:

- Příkazy G-funkcí všech existujících skupin G-funkcí, např. „G0“, „INVCW“, „POLY“, „ROT“, „KONT“, „SOFT“, „CUT2D“, „CDON“, „RMBBL“, „SPATH“
- DIN- nebo NC-adresy, jako např. „ADIS“, „RNDM“, „SPN“, „SR“, „MEAS“
- Funkce, jako např. „TANG(...)“ nebo „GETMDACT“
- Procedury, např. „SBLOF“.
- Klíčová slova, např. „ACN“, „DEFINE“ nebo „SETMS“
- Systémová data, např. strojní parametry \$M... , nastavované parametry \$S... nebo data volitelných doplňků \$O...
- Systémové proměnné \$A... , \$V... , \$P...
- Početní parametry R...
- Názvy aktivovaných cyklů
- Proměnné GUD a LUD
- Názvy maker
- Názvy návěští

18.6 Kontrola existujícího rozsahu NC jazyka (STRINGIS)

Výsledná hodnota

Pro výslednou hodnotu jsou důležité pouze první tři desetinná místa. Výsledná hodnota má formát yxx, kde y = základní informace a xx = podrobné informace.

Výsledná hodnota	Význam
000	Řetězec "název" není v předkládaném systému znám ¹⁾
100	Řetězec "název" je prvek programovacího jazyka NC systému, momentálně ale nemůže být naprogramován (volitelný doplněk/funkce jsou neaktivní)
2xx	Řetězec "název" je naprogramovatelný prvek programovacího jazyka NC systému (volitelný doplněk/funkce jsou aktivní) Podrobné informace xx obsahují další údaje o druhu prvku:
	xx Význam
	01 Adresa podle normy DIN nebo NC adresa ²⁾
	02 Příkaz G-funkce (např. G04, INVCW)
	03 Funkce s výstupní hodnotou
	04 Funkce bez výstupní hodnoty
	05 Klíčové slovo (např. DEFINE)
	06 Strojní parametry (\$M...), nastavované parametry (\$S...) nebo parametry volitelných doplňků (\$O...)
	07 Systémové parametry, např. systémové proměnné (\$...) nebo početní parametry (R...)
	08 Cyklus (cyklus musí být načten v NC systému a program cyklu musí být aktivní ³⁾)
	09 Proměnná GUD (proměnné GUD musí být definovány v definičním souboru GUD a proměnné GUD musí být aktivovány)
	10 Název makra (makro musí být definováno v definičním souboru makra a makra musí být aktivována) ⁴⁾
	11 Proměnná LUD aktuálního výrobního programu
	12 Příkaz G-funkce podle normy ISO (režim jazyka ISO musí být aktivní)
400	Řetězec "název" je NC-adresa, která nebyla rozpoznána jako xx = 01 nebo xx = 10 a není typu G nebo R ²⁾
y00	Není možné žádné specifické přiřazení

1) V závislosti na řídicím systému je za určitých okolností známá pouze určitá podmnožina příkazů programovacího jazyka NC systému, např. v případě systému SINUMERIK 802D sl. U těchto řídicích systémů je pro řetězce, které jsou v principu příkazy NC jazyka firmy Siemens, výsledkem hodnota 0. Toto chování může být pomocí strojního parametru MD10711 \$MN_NC_LANGUAGE_CONFIGURATION změněno. Je-li nastaveno MD10711 = 1, potom bude pro příkazy NC jazyka firmy Siemens vždy výsledkem hodnota 100.

2) NC adresami jsou následující písmena: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Tyto NC adresy mohou být naprogramovány také s rozšířením adresy. Rozšíření adresy může být zadáno při kontrole pomocí funkce STRINGIS. Příklad: 201 == STRINGIS("A1").

Písmena: D, F, H, L, M, N, O, P, S, T jsou NC adresy nebo pomocné funkce, které jsou používány podle uživatelských definic. V jejich případě je výsledkem vždy hodnota 400. Příklad: 400 == STRINGIS("D"). Tyto NC adresy mohou být zadány při kontrole pomocí funkce STRINGIS, ale bez rozšíření adresy.

Příklad: 000 == STRINGIS("M02"), ale 400 == STRINGIS("M").

3) Názvy parametrů cyklů nemohou být pomocí funkce STRINGIS kontrolovány.

4) Adresa definovaná jako makro, např. G, H, M, L, je jako makro identifikována.

Příklady

V následujících příkladech se předpokládá, že řetězec zadaný jako prvek jazyka NC systému, pokud není uvedeno jinak, může být v principu v NC systému naprogramován.

1. Řetězec "T" je definován jako pomocná funkce:

```
400 == STRINGIS ("T")
000 == STRINGIS ("T3")
```

2. Řetězec "X" je definován jako osa:

```
201 == STRINGIS ("X")
201 == STRINGIS ("X1")
```

3. Řetězec "A2" je definován jako NC adresa s rozšířením:

```
201 == STRINGIS ("A")
201 == STRINGIS ("A2")
```

4. Řetězec "INVCW" je definován jako specifický příkaz G-funkce:

```
202 == STRINGIS ("INVCW")
```

5. Řetězec "\$MC_GCODE_RESET_VALUES" je definován jako strojní parametr:

```
206 == STRINGIS (" $MC_GCODE_RESET_VALUES")
```

6. Řetězec "GETMDACT" je funkce NC-jazyka:

```
203 == STRINGIS ("GETMDACT ")
```

7. Řetězec "DEFINE" je klíčové slovo:

```
205 == STRINGIS ("DEFINE")
```

8. Řetězec "\$TC_DP3" je systémový parametr (složka délky nástroje):

```
207 == STRINGIS (" $TC_DP3")
```

9. Řetězec "\$TC_TP4" je systémový parametr (velikost nástroje):

```
207 == STRINGIS (" $TC_TP4")
```

10. Řetězec "\$TC_MPP4" je systémový parametr (stav místa v zásobníku):

- Správa zásobníku nástrojů je aktivní. 207 == STRINGIS (" \$TC_MPP4") ;
- Správa zásobníku nástrojů není aktivní. 000 == STRINGIS (" \$TC_MPP4")

Viz také kapitola: Správa zásobníku nástrojů.

11. Řetězec "MACHINERY_NAME" je definován jako proměnná typu GUD.

```
209 == STRINGIS ("MACHINERY_NAME")
```

12. Řetězec "LONGMACRO" je definován jako makro:

```
210 == STRINGIS ("LONGMACRO")
```

13. Řetězec "MYVAR" je definován jako proměnná typu LUD.

```
211 == STRINGIS ("MYVAR")
```

14. Řetězec "XYZ" není žádný příkaz, který by byl v NC systému známý, není ani názvem proměnné GUD, makra nebo cyklu:

```
000 == STRINGIS ("XYZ")
```

Správa zásobníku nástrojů

Jestliže funkce "Správa zásobníku nástrojů" není aktivní, poskytuje funkce STRINGIS pro systémové parametry správy zásobníku nástrojů, nezávisle na strojním parametru, následující výsledky:

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION

Vždy hodnota 000.

Režim ISO

Pokud je funkce "Režim ISO" aktivní:

- MD18800 \$MN_MM_EXTERN_LANGUAGE (aktivování externích BC jazyků)
- MD10880 \$MN_MM_EXTERN_CNC_SYSTEM (řídící systémy, které se mají přizpůsobit)

kontroluje funkce STRINGIS zadaný řetězec napřed jako příkaz G-funkce systému SINUMERIK. Pokud řetězec není příkazem G-funkce systému SINUMERIK, zkontroluje se potom, zda se nejedná o příkaz G-funkce podle normy ISO.

Naprogramovaná přepnutí (G290 (režim SINUMERIK), G291 (režim ISO)) nemají na funkci STRINGIS žádný vliv.

Příklad:

Strojní parametry, které jsou pro funkci STRINGIS(...) důležité, mají následující hodnoty:

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION = 2 (jako známé jsou rozpoznány jen ty příkazy NC-jazyka, jejichž volitelné doplňky jsou aktivovány)
- MD19410 \$ON_TRAFO_TYPE_MASK = 'H0' (volitelný doplněk: transformace)
- MD10700 \$MN_PREPROCESSING_LEVEL='H43' (předběžné zpracování pro cykly je aktivní)

Program uvedený v následujícím příkladu se zpracuje bez chybového hlášení:

Programový kód	Komentář
N1 R1=STRINGIS("TRACYL")	R1 == 0, protože příkaz TRACYL je kvůli chybějícímu volitelnému doplňku "Transformace" rozpoznán jako "neznámý".
N2 IF STRINGIS("TRACYL") == 204	
N3 TRACYL(1,2,3)	; N3 se bude přeskakovat
N4 ELSE	
N5 G00	; a místo něj se bude zpracovávat N5
N6 ENDIF	
N7 M30	

18.7 Interaktivní vyvolávání oken z výrobního programu (MMC)

Prostřednictvím předem definovaného podprogramu MMC(...) mohou být z NC programu zobrazovány na uživatelském rozhraní např. systému SINUMERIK Operate specifická uživatelská dialogová okna.

Dialogová okna mohou být navrhována pro následující druhy těchto objektů:

- Run MyScreens
- Easy XML
- Uživatelské XML

Literatura

- Run MyScreens, příručka programátora
- Easy XML, příručka programátora

Syntaxe

MMC ("<ADDRESS>, <COMMAND>, <FILE>, <DIALOG>", "<QUIT>")

Význam

MMC (. . .) :	Identifikátor podprogramu Parametry mají kódovaná místa a čárkou zadávají se rozdělené do dvou řetězců, což jsou příkazový řetězec a potvrzovací řetězec.	
Parametry v rámci příkazového řetězce:		
<ADDRESS>:	Systémová oblast, ve které se v konfiguraci nastavený dialog s uživatelem uskutečňuje.	
	Funkce	Systémové oblasti
	Uživatelské dialogové okno „Run MyScreens“	CYCLES
	Uživatelské dialogové okno „Easy XML“	CYCLES
	Uživatelské XML	XML
	Okno v popředí „Run MyScreens“	POPUPDLG
	Okno v popředí „Easy XML“	POPUPDLG
<COMMAND>:	Příkaz, který má být zpracován	
	Funkce	Příkazy
	Uživatelské dialogové okno „Run MyScreens“	PICTURE_ON, PICTURE_OFF
	Uživatelské dialogové okno „Easy XML“	PICTURE_ON, PICTURE_OFF
	Uživatelské XML	XML_ON, XML_OFF
	Okno v popředí „Run MyScreens“	PICTURE_ON, PICTURE_OFF
	Okno v popředí „Easy XML“	PICTURE_ON, PICTURE_OFF

<FILE>:	Název souboru, ve kterém je naprogramováno dialogové okno, které se má zobrazit	
	Funkce	Soubory
	Uživatelské dialogové okno „Run MyScreens“	<name>.com
	Uživatelské dialogové okno „Easy XML“	<name>.xml
	Uživatelské XML	<name>.xml
	Okno v popředí „Run MyScreens“	<name>.com
	Okno v popředí „Easy XML“	<name>.xml
	Okno v popředí „Easy XML“ sestavené přímo v NC programu (viz příklad 2)	xmldial_emb.xml
<DIALOG>:	Název dialogového okna, které se má zobrazovat	
	Funkce	Název dialogového okna
	Všechny funkce kromě okna v popředí „Easy XML“ sestavené přímo v NC programu	Název dialogového okna zadaný v souboru <FILE>
	Okno v popředí „Easy XML“ sestavené přímo v NC programu (viz příklad 3)	main
Parametr v rámci potvrzovacího řetězce:		
<QUIT>:	Typ potvrzování	
	N:	Žádné potvrzování. Zpracování programu se uskutečňuje po odeslání příkazu. Jestliže příkaz nebylo možné úspěšně zpracovat, nejste o tom nijak informováni. Upozornění Typ potvrzování „N“ je potřeba použít, jestliže je v NC programu naprogramována doba zobrazování (doba prodlevy) (viz příklad 2 níže)
	A:	Asynchronní potvrzování Zpracování programu se uskutečňuje po odeslání příkazu. Výstupní hodnota se ukládá do specifické uživatelské potvrzovací proměnné (proměnné GUD) definované v rámci vytváření dialogového okna a je možno ji v NC programu načíst.

Příklad

Příklad 1

Zobrazení dialogového okna a reakce na uživatelskou podmínku v NC programu.

Programový kód	Komentář
; potvrzovací proměnná QUIT již byla deklarována jako globální uživatelská proměnná (GUD)	
; typu STRING při vytváření dialogového okna:	
; DEF NCK STRING[20] QUIT	
QUIT = "XXX"	; inicializace potvrzovací proměnné
G4 F5	

18.7 Interaktivní vyvolávání oken z výrobního programu (MMC)

Programový kód	Komentář
MMC("CYCLES,PICTURE- RE_ON,test.com,test1","A")	; zobrazení dialogového okna ; - systémová oblast: CYCLES ; - stav obrazovky: PICTURE_ON (zobrazovat) ; - soubor obrazovky s dialogového okna: test.com ; - obrazovka s dialogovým oknem test1
INPUT:	; čekání na uživatelskou podmínku
STOPRE	; - zastavení předběžného zpracování
IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK	; - programové tlačítko „RUN“
IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK	; - programové tlačítko „CHK“
GOTOB INPUT	; => čekání
WORK:	; - programové tlačítko „RUN“ stisknuto
MSG(„Pokračovat ve zpracování -> NC-Start“)	; výstup hlášení
MMC("CYCLES,PICTURE_OFF","N")	; zavření dialogového okna
M0	; Čekání na tlačítko NC-Start
GOTOF END	; => konec programu
CHECK:	; - programové tlačítko „CHK“ stisknuto
MSG(„Najíždění na pozici -> NC-Start“)	; výstup hlášení
MMC("CYCLES,PICTURE_OFF","N")	; zavření dialogového okna
M0	; Čekání na tlačítko NC-Start
GOTOF END	; => konec programu
END:	
...	

Příklad 2

Doba zobrazování dialogového okna se v NC programu určuje např. dobou prodlevy.

Programový kód	Komentář
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")	; zobrazení dialogového okna
X200	
Z40	
MMC("POPUPDLG,PICTURE_OFF","N")	; zavření dialogového okna

Příklad 3

Vložení skriptu okna v popředí do NC programu a jeho použití.

Programový kód

```
PROC POPUP_TEST
```

Programový kód

```

; ----- Skript -----
; <main_dialog entry="rpara_main">
;   <let name="xpos" />
;   <let name="ypos" />
;   <let name="field_name" type="string" />
;   <let name="num" />
;   <menu name="rpara_main">
;     <open_form name="rpara_form"/>
;     <softkey_back>
;       <close_form />
;     </softkey_back>
;   </menu>
;   <form name="rpara_form">
;     <init>
;       <caption>mask from NC part program</caption>
;       <let name="count" >0</let>
;       <op>
;         xpos = 120;
;         ypos = 34;
;         "nck/Channel/Parameter/R[10]" = 10;
;       </op>
;       <!-- load the number of controls -->
;       <op>
;         num = "nck/Channel/Parameter/R[10]";
;       </op>
;       <while>
;         <condition> count < num</condition>
;         <print name="field_name" text="edit%d">count</print>
;         <op>
;           ypos = ypos + 24;
;           count = count + 1;
;         </op>
;       </while>
;     </init>
;     <paint>
;       <op>
;         xpos = 8;
;         ypos = 36;
;         count = 0;
;       </op>
;       <while>
;         <condition>count < num</condition>
;         <print name="field_name" text="R-Parameter%d">count</print>
;         <text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>

```

Programový kód

```

;          <op>
;          ypos = ypos + 24;
;          count = count + 1;
;          </op>
;          </while>
;          </paint>
;        </form>
; </main_dialog>
; ===== Úsek programu =====
...
G94 F100
MMC ("POPUPDLG, PICTURE_ON, xmldial_emb.xml, main", "N")
G4 F4
X200
MMC ("POPUPDLG, PICTURE_OFF", "N")
G4 F2
X0
...

```

Okrajové podmínky

- Definiční soubory *.com dialogových oken musí být uloženy ve složce „proj“.
- Definiční soubory Easy XML *.xml dialogových oken musí být uloženy ve složce „appl“.
Pokud jsou definiční soubory uloženy v nějakém jiném adresáři, musí být nepřímo zadán údaj cesty, který vychází z adresáře „appl“.
- Uživatelem definovaná dialogová okna nemohou být zobrazována současně z různých kanálů.

18.8 Doba zpracování programu / počítadlo obrobků

Za účelem podpory pracovníků obsluhy obráběcích strojů jsou k dispozici informace týkající doby zpracovávání programu a počtu kusů.

Tyto informace mohou být zpracovávány jako systémové proměnné v NC programu a/nebo v programu PLC. Současně jsou tyto informace k dispozici pro zobrazování na uživatelském rozhraní.

18.8.1 Doba zpracování programu

Funkce "Doba zpracování programu" dává k dispozici interní časovač NC systému pro monitorování technologických procesů, jehož hodnotu je možné načítat ve výrobním programu a v synchronních akcích pomocí systémových proměnných kanálu a NC systému.

Spouštění měření doby zpracování (\$AC_PROG_NET_TIME_TRIGGER) je jediná systémová proměnná této funkce, do které lze zapisovat, a slouží pro selektivní měření úseků programu, tzn. zapsáním hodnoty pro spuštění může být v NC programu měření času zastaveno a znovu spuštěno.

Systémové proměnné	Význam	Aktivita
V NC systému		
\$AN_SETUP_TIME	Čas od posledního náběhu řídicího systému se standardními hodnotami ("studený start") v minutách Při každém náběhu řídicího systému se standardními hodnotami se automaticky nastaví na "0".	• vždy aktivní
\$AN_POWERON_TIME	Čas od posledního normálního náběhu řídicího systému ("teplý start") v minutách Při každém normálním náběhu řídicího systému se automaticky nastaví na "0".	
V kanálu		
\$AC_OPERATING_TIME	Celková doba, po kterou jsou zpracovávány NC-programy v automatickém provozním režimu v sekundách Hodnota se s každým náběhem řídicího systému se automaticky nastaví na "0".	• Aktivování prostřednictvím MD27860 • v provozním režimu "AUTO"
\$AC_CYCLE_TIME	Doba zpracování zvoleného NC programu v sekundách Hodnota se s každým spuštěním nového NC programu automaticky nastaví na "0".	
\$AC_CUTTING_TIME	Doba opracovávání v sekundách Měří se doba, po kterou se pohybují dráhové osy (minimálně jedna je aktivní) bez aktivního rychlého posuvu ve všech NC programech od stisknutí tlačítka NC-Start do konce programu/resetu NC systému. Měření této doby se přeruší také tehdy, když je aktivní doba prodlevy. Hodnota se při každém náběhu řídicího systému se standardními hodnotami se automaticky nastaví na "0".	

Systémové proměnné	Význam	Aktivita	
\$AC_ACT_PROG_NET_TIME	Aktuální čistá doba zpracování momentálně vybraného NC programu v sekundách S každým spuštěním nového NC programu automaticky nastaví na "0".	- vždy aktivní - v provozním režimu "AUTO"	
\$AC_OLD_PROG_NET_TIME	Čistá doba zpracování právě správně a pomocí příkazu M30 ukončeného programu v sekundách		
\$AC_OLD_PROG_NET_TIME_COUNT	Změny parametru \$AC_OLD_PROG_NET_TIME Po náběhu systému je proměnná \$AC_OLD_PROG_NET_TIME_COUNT nastavena na "0". Proměnná \$AC_OLD_PROG_NET_TIME_COUNT se zvýší vždy, když řídící systém zapíše do proměnné \$AC_OLD_PROG_NET_TIME novou hodnotu.		
\$AC_PROG_NET_TIME_TRIGGER	Spouštění pro měření doby zpracování:		v provozním režimu "AUTO"
	0	Neutrální stav Spouštění není aktivní.	
	1	Ukončit Ukončí měření a zkopíruje hodnotu z proměnné \$AC_ACT_PROG_NET_TIME do \$AC_OLD_PROG_NET_TIME. \$AC_ACT_PROG_NET_TIME se znovu nastaví na "0" a potom se zpusť od začátku.	
	2	Start Spustí se měření a proměnná \$AC_ACT_PROG_NET_TIME se přitom nastaví na "0". Proměnná \$AC_OLD_PROG_NET_TIME zůstane nezměněna.	
	3	Zastavit Zastaví měření: Proměnná \$AC_OLD_PROG_NET_TIME se nemění a obsahuje hodnotu proměnné \$AC_ACT_PROG_NET_TIME konstantní, dokud se nespustí další pokračování.	
4	Pokračovat Pokračování měření, tzn. dříve zastavené měření bude pokračovat dál. \$AC_ACT_PROG_NET_TIME bude běžet dále. \$AC_OLD_PROG_NET_TIME zůstane nezměněna.		
Při náběhu systému (Power-On) jsou všechny systémové proměnné nastaveny na "0"!			

Poznámka

Výrobce stroje

Spuštění aktivovatelného časovače se uskutečňuje pomocí strojního parametru MD27860 \$MC_PROCESSTIMER_MODE.

Chování aktivního měření času v případě určitých funkcí (např. GOTOS, korekce (override) = 0%, aktivní posuv při zkušebním zpracování, zkouška programu, ASUP, PROG_EVENT, ...) se nastavuje v konfiguraci pomocí strojních parametrů MD27850 \$MC_PROG_NET_TIMER_MODE a MD27860 \$MC_PROCESSTIMER_MODE.

Literatura:

Příručka k funkcím, Základní funkce; BAG, kanál, zpracování programu, chování při resetu (K1), kapitola: Doba zpracování programu

Poznámka

Zbývajíc čas pro obrobek

Jestliže mají být jeden po druhém vyráběny stejné obrobky, je možno pomocí následujících časových hodnot vypočítat zbývajíc čas pro daný obrobek:

- dobu opracovávání pro naposled vyrobený obrobek (viz \$AC_OLD_PROG_NET_TIME)
- aktuální dobu opracovávání (viz \$AC_ACT_PROG_NET_TIME)

Zbývajíc čas se vedle aktuální doby opracovávání vypisuje na uživatelském rozhraní.

Poznámka

Použití příkazu STOPRE

Systémové proměnné \$AC_OLD_PROG_NET_TIME a \$AC_OLD_PROG_NET_TIME_COUNT nezpůsobují žádné implicitní zastavení předběžného zpracování. Při použití ve výrobním programu nelze poznat, kdy hodnota systémové proměnné pochází z předchozího zpracování programu. Jestliže se ale příkazy pro spuštění měření doby zpracování (\$AC_PROG_NET_TIME_TRIGGER) zapisují s velmi vysokou frekvencí a hodnota v proměnné \$AC_OLD_PROG_NET_TIME se v důsledku toho mění velmi často, pak by měl být ve výrobním programu použit explicitní příkaz STOPRE.

Okrajové podmínky

- **Vyhledávání bloku**

Při vyhledávání bloku se žádná doba zpracovávání programu nezjišťuje.

- **REPOS**

Doba trvání operace REPOS se do aktuální doby zpracovávání programu (\$AC_ACT_PROG_NET_TIME) započítává.

Příklady

Příklad 1: Měření doby zpracovávání programu "mySubProgrammA"**Programový kód**

```

...
N50 DO $AC_PROG_NET_TIME_TRIGGER=2
N60 FOR ii= 0 TO 300
N70 mySubProgrammA
N80 DO $AC_PROG_NET_TIME_TRIGGER=1
N95 ENDFOR
N97 mySubProgrammB
N98 M30

```

Poté, co byl zpracován programový řádek N80, se v proměnné \$AC_OLD_PROG_NET_TIME nachází čistá doba zpracování programu "mySubProgrammA".

Hodnota proměnné \$AC_OLD_PROG_NET_TIME:

- zůstává zachována i za příkazem M30.
- po každém průchodu smyčkou se aktualizuje.

Příklad 2: Měření doby zpracovávání programů "mySubProgrammA" a "mySubProgrammC"**Programový kód**

```

...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammd
N90 M30

```

18.8.2 Počítadlo obrobků

Funkce "Počítadlo obrobků" poskytuje k dispozici různé čítače, které lze použít zejména pro počítání obrobků v rámci řídicího systému.

Počítadla existují jako specifické kanálové proměnné s oprávněními pro čtení a pro zápis a s rozsahem hodnot od 0 do 999 999 999.

Systémové proměnné	Význam
\$AC_REQUIRED_PARTS	Počet obrobků, které mají být vyrobeny (požadovaná hodnota obrobků) V tomto čítači může být definován počet obrobků, při jehož dosažení se počet aktuálních obrobků v proměnné (\$AC_ACTUAL_PARTS) nastaví zpět na "0".
\$AC_TOTAL_PARTS	Celkový počet vyrobených obrobků (celková skutečná hodnota) Tento čítač udává počet všech obrobků vyrobených od okamžiku spuštění. Hodnota se automaticky nastaví na "0" jen při náběhu řídicího systému se standardními hodnotami.
\$AC_ACTUAL_PARTS	Počet vyrobených obrobků (skutečná hodnota počtu obrobků) V tomto čítači je zaznamenáván počet všech obrobků vyrobených od okamžiku spuštění. Jestliže je dosaženo požadované hodnoty počtu kusů (\$AC_REQUIRED_PARTS), čítač se automaticky nastaví zpět na "0" (předpokladem je, že \$AC_REQUIRED_PARTS > 0).
\$AC_SPECIAL_PARTS	Počet uživatelem počítaných obrobků Tento čítač umožňuje uživateli počítání obrobků podle jeho vlastní definice. Je možno také definovat aktivování alarmu, jakmile je dosaženo požadovaného počtu obrobků (\$AC_REQUIRED_PARTS). Vynulování čítače musí provádět sám uživatel.

Poznámka

Všechna počítadla obrobků se při náběhu řídicího systému se standardními hodnotami nastavují na "0" a mohou být nezávisle na svém aktivování čteny nebo lze do nich zapisovat.

Poznámka

Aktivování čítačů, okamžik vynulování a algoritmus počítání je možno ovlivňovat pomocí kanálových strojních parametrů.

Poznámka

Počítání obrobků s příkazem M-funkce definovaným uživatelem

Pomocí strojního parametru může být nastaveno, že počítané impulzy pro různá počítadla obrobků mohou být spouštěny nikoli pomocí příkazu pro konec programu M2/M30, ale pomocí uživatelem definované M-funkce.

18.9 Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE)

Zápis dat z výrobního programu do externího zařízení/souboru probíhá ve třech krocích:

1. Otevření externího zařízení/souboru
Pomocí příkazu EXTOPEN se pro kanál otevře externí zařízení/soubor, které lze použít pro zápis.
2. Zápis dat
Výstupní data mohou být připravena pomocí funkcí NC jazyka pro práci s řetězci, např. SPRINT. Samotný zápis se uskutečňuje pomocí příkazu WRITE.
3. Zavření externího zařízení/souboru
Pomocí příkazu EXTCLOSE nebo při dosažení konce programu (M30), jakož i při resetu kanálu, že v kanálu obsazený externí zařízení/soubor bude opět uvolněn.

Syntaxe

```
DEF INT <výsledek>
DEF STRING[<n>] <výstup>
...
EXTOPEN(<výsledek>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
...
<výstup>="výstup dat"
WRITE(<výsledek>,<ExtDev>,<Output>)
...
EXTCLOSE(<výsledek>,<ExtDev>)
```

Význam

EXTOPEN:	Předem definovaná procedura pro otevření externího zařízení/souboru	
<výsledek>:	Parametr 1: Proměnná pro výsledek Na základě hodnoty proměnné pro výsledek je možné v programu vyhodnocovat, jak operace skončila, a na základě toho potom odpovídajícím způsobem postupovat.	
	Typ:	INT
	Hodnoty:	0 žádná chyba
		1 externí zařízení nebylo možné otevřít
		2 externí zařízení není konfigurováno
		3 pro externí zařízení je v konfiguraci zadána nesprávná cesta
		4 pro externí zařízení nemáte přístupová oprávnění
		5 Režim využívání: externímu zařízení je přiřazen "exkluzivní" přístup
		6 Režim využívání: externímu zařízení je přiřazen "sdílený" přístup
		7 Délka souboru je větší než kolik udává parametr LOCAL_DRIVE_MAX_FILESIZE
		8 je překročen maximální počet externích zařízení
		9 funkce pro LOCAL_DRIVE není aktivována
		11 rozhraní RS-232 je již obsazeno funkcí Easy-Message (jen 828D)
		12 Režim zápisu: zadání je v rozporu se souborem extdev.ini
		16 naprogramována neplatná externí cesta
	22	externí zařízení není připojeno

18.9 Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE)

<ExtDev>:	Parametr 2: Symbolický identifikátor pro otevírané externí zařízení/soubor	
	Typ:	STRING
	Symbolický identifikátor se skládá z těchto částí:	
	1. název logického zařízení	
	2. v případě potřeby následovaný cestou k danému souboru (připojuje se pomocí znaku "/")	
	Jsou definovány následující názvy logických zařízení :	
	"LOCAL_DRIVE":	Lokální kompaktní Flash-karta (předdefinováno)
	"CYC_DRIVE":	Rezervované udání jednotky používané v cyklech firmy SIEMENS (předdefinováno)
	"/dev/ext/1", ... "/dev/ext/9":	Síťové jednotky, které jsou k dispozici Upozornění: Je zapotřebí nastavit konfiguraci v souboru extdev.ini!
	"/dev/cyc/1", "/dev/cyc/2":	Rezervované udání jednotky používané v cyklech firmy SIEMENS Upozornění: Je zapotřebí nastavit konfiguraci v souboru extdev.ini!
	"/dev/v24":	Rozhraní RS-232 Upozornění: Je zapotřebí nastavit konfiguraci v souboru extdev.ini!
Cesta k souboru:		
- Pro "LOCAL_DRIVE" a "CYC_DRIVE" musí být zadána cesta k souboru, např.: "LOCAL_DRIVE/my_dir/my_file.txt" - Názvy logických zařízení "/dev/ext/1...9" a "/dev/cyc/1...2" mohou prostřednictvím konfigurace odkazovat: - na určitý soubor, potom smí být uváděny pouze názvy logických zařízení, např.: "/dev/ext/4" - nebo na adresář, potom ale musí být cesta k souboru zadána, např.: "/dev/ext/5/my_dir/my_file.txt" - U názvu "/dev/v24" nesmí být žádná cesta k souboru připojena.		
Poznámka:		
Pro názvy logických zařízení "/dev/ext/1...9", "/dev/v24" a "/dev/cyc/1...2" nejsou velká a malá písmena rozlišována, u zadání cesty k souboru jsou velká a malá písmena rozlišována. Pro "LOCAL_DRIVE" a "CYC_DRIVE" jsou přípustná jen velká písmena.		

<SyncMode>:	Parametr 3: Režim zpracování pro příkazy WRITE k tomuto zařízení/souboru		
	Typ:	STRING	
	Hodnoty:	"SYN": Synchronní zápis Zpracovávání programu je pozastaveno, dokud není operace zápisu dokončena. Úspěšné ukončení synchronního zápisu může být zkontrolováno pomocí vyhodnocování chybové proměnné příkazu WRITE.	
		"ASYN": Asynchronní zápis Zpracovávání programu není příkazem WRITE přerušeno. Poznámka: Proměnná pro výsledek příkazu WRITE nemá v tomto režimu žádnou vypovídací hodnotu a má vždy hodnotu 0 (žádná chyba). V tomto režimu si nemůžete být jisti, že byl příkaz WRITE úspěšně zpracován.	
<AccessMode>:	Parametr 4: Režim využívání pro toto zařízení/tento soubor		
	Typ:	STRING	
	Hodnoty:	"SHARED": Zařízení/soubor jsou přiřazovány ve sdíleném režimu. Zařízení může být používáno také jinými kanály, tzn. otevřenými rovněž v tomto režimu.	
		"EXCL": Zařízení/soubor jsou používány výlučně v tomto kanálu a žádný další kanál nemůže toto zařízení používat také.	
<WriteMode>:	Parametr 5: Režim zápisu pro příkazy WRITE k tomuto zařízení/souboru (volitelné)		
	Typ:	STRING	
	Hodnoty:	"APP": Připojit Obsah souboru zůstává zachován, zapisovaná data se vkládají na jeho konec.	
		"OVR": Přepisování Obsah souboru se vymaže a následující operací zápisu se znovu vytvoří.	
Poznámka: Režim zápisu nastavený v konfiguraci v souboru extdev.ini není možné pomocí tohoto parametru přepsat. V případě konfliktu je volání funkce EXTOPEN potvrzeno s chybou.			

WRITE:	Předem definovaná procedura pro zápis výstupních dat
--------	--

18.9 Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE)

EXTCLOSE	Předem definovaná procedura pro zavření otevřeného externího zařízení/souboru		
<výsledek>:	Parametr 1: Proměnná pro výsledek		
	Typ:	INT	
	Hodnoty:	0	žádná chyba
		16	naprogramována neplatná externí cesta
		21	chyba při zavírání externího zařízení
<ExtDev>:	Parametr 2: Symbolický identifikátor pro zavírané externí zařízení/soubor; popis naleznete v části věnované příkazu EXTOPEN! Poznámka: Tento identifikátor musí být identický s identifikátorem použitým ve volání pomocí příkazu EXTOPEN!		

Příklad:

Programový kód	
N10	DEF INT RESULT
N20	DEF BOOL EXTDEVICE
N30	DEF STRING[80] OUTPUT
N40	DEF INT PHASE
N50	EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60	IF RESULT > 0
N70	MSG("Fehler bei EXTOPEN:" << RESULT)
N80	ELSE
N90	EXTDEVICE=TRUE
N100	ENDIF
...	
N200	PHASE=4
N210	IF EXTDEVICE
N220	OUTPUT=SPRINT("Ende Phase: %D",PHASE)
N230	WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240	ENDIF
...	

Viz také

Operace s řetězcí (Strana 86)

Zápis do souboru (WRITE) (Strana 148)

18.10 Alarmy (SETAL)

V NC programu mohou být aktivovány alarmy, které se pak vypisují na uživatelském rozhraní ve zvláštním poli. S alarmem je vždy spojena nějaká reakce řídicího systému závisající na kategorii alarmu.

Literatura:

Pokud budete potřebovat podrobnější informace o reakcích na alarm, nahlédněte do Příručky pro uvedení do provozu.

Syntaxe

SETAL(<číslo alarmu>[,<znakový řetězec>])

Význam

SETAL:	Klíčové slovo pro programování alarmu. Příkaz SETAL musí být naprogramován v samostatném NC-bloku.	
<číslo alarmu>	Proměnná typu INT. Obsahuje číslo alarmu. Platným rozsahem alarmových čísel je interval 60000 až 69999, z čehož 60000 až 64999 je vyhrazeno pro cykly Siemens, zatímco 65000 až 69999 je k dispozici uživateli.	
<Řetězec znaků>:	Při programování alarmů uživatelských cyklů může být kromě alarmového čísla zadán ještě řetězec znaků obsahující až 4 parametry. V těchto parametrech mohou být definovány proměnné uživatelské texty. Budou Vám k dispozici ale i následující předdefinované parametry:	
	Parametr	Význam
	%1	Číslo kanálu
	%2	Číslo bloku, návěští
	%3	Textový index pro alarmy cyklů
	%4	Pomocný parametr alarmu

Poznámka

Texty alarmu musí být konfigurovány v uživatelském rozhraní.

Poznámka

Jestliže se má alarm vypisovat v aktivním jazyce uživatelského rozhraní, potřebuje uživatel informace o tom, který jazyk je momentálně na HMI nastaven. Tento údaj je možné ve výrobním programu a v synchronních akcích zjistit pomocí systémové proměnné \$AN_LANGUAGE_ON_HMI (viz "Aktuální jazyk v HMI (Strana 941)").

Příklad:

Programový kód	Komentář
...	
N100 SETAL (65000)	; nastavení alarmu č. 65000

Programový kód	Komentář
...	

18.11 Rozšířené zastavování a odjíždění (ESR)

Funkce "Rozšířené zastavování a odjíždění", v dalším textu označovaná zkratkou ESR, nabízí možnost flexibilní reakce v chybových situacích v závislosti na procesu.

- **Rozšířené zastavování**
Pokud to specifická chybová situace umožňuje, jsou všechny osy, které byly pro rozšířené zastavování uvolněny, řízeným způsobem zastaveny.
- **Odjíždění**
Nástroj, který se momentálně nachází v záběru, je co možno nejrychleji stažen pryč od obrobku.
- **Generátorický režim (funkce pohonu SINAMICS "Regulace stejnosměrného napětí")**
Jestliže napětí meziobvodu poklesne pod hodnotu nastavitelnou v parametrech, např. při výpadku síťového napětí, elektrická energie potřebná pro odjížděcí pohyb se vytvoří prostřednictvím napájení z druhé strany z brzdící energie pohonu, který toto umožňuje (generátorický režim).

Zdroje pro spuštění

Obecné zdroje (externí mimo NC systém/globální nebo BAG/kanálové)

- Digitální vstupy (např. na modulu NCU), zpětně čitelné zobrazení digitálních výstupů existující uvnitř řídicího systému (\$A_IN, \$A_OUT)
- Stav kanálu (\$AC_STAT)
- Signály VDI (\$A_DBB)
- Součtová hlášení určitého počtu alarmů (\$AC_ALARM_STAT)

Osové zdroje

- Mezní hodnota pro nouzový zpětný pohyb vlečné osy (synchronní chod elektronické spojky, \$VC_EG_SYNCDIFF[<vlečná osa>])
- Pohon: Mezní hodnota výstrahy meziobvodu (hrozící příliš nízké napětí), \$AA_ESR_STAT[<Osa>]
- Pohon: Spodní mezní hodnota otáček generátoru (již není k dispozici žádná další rotační energie, která by se mohla použít pro napájení z druhé strany), \$AA_ESR_STAT[<Osa>].

Spojovací logika statických synchronních akcí: Vazba zdroj/reakce

Aby se na základě zdrojových událostí relativně brzy spouštěly určité reakce, používají se flexibilní možnosti vazeb ve statických synchronních akcích.

Zajištění spojení všech důležitých zdrojů pomocí statických synchronních vazeb má plně v rukou uživatel. Systémové proměnné zdrojů může vyhodnocovat buď jako celek nebo také selektivně pomocí bitových masek a k tomu připojovat požadované reakce. Statické synchronní akce se uplatňují ve všech provozních režimech.

Literatura:

Příručka k funkcím, Synchronní akce

Aktivování

Odblokování funkcí

Pro odblokování funkcí generátorického režimu, zastavování a odjíždění se používá nastavení odpovídajících řídicích signálů \$AA_ESR_ENABLE. Tyto řídicí signály mohou být ze synchronních akcí měněny.

Spouštění funkcí

ESR se spouští společně pro všechny odblokované osy nastavením systémové proměnné \$AC_ESR_TRIGGER.

Generátorický režim se aktivuje v pohonu "automaticky", kdykoli je rozpoznán hrozící pokles napětí meziobvodu.

Zastavování a/nebo odjíždění nezávislé na pohonu se aktivuje při rozpoznání výpadku komunikace (mezi NC systémem a pohonem), ale také v případě zjištění poklesu napětí v meziobvodu v pohonu (za předpokladu příslušné konfigurace a odblokování funkcí).

Zastavení a/nebo odjíždění nezávislé na pohonu může být kromě toho spuštěno ještě i na straně NC systému prostřednictvím aktivování odpovídajících řídicích signálů \$AN_ESR_TRIGGER (příkaz přenášení do všech pohonů).

Literatura

Pokud budete potřebovat podrobné informace o ESR, viz:

Příručka k funkcím, Speciální funkce; Rozšířené zastavování a odjíždění (R3)

18.11.1 ESR řízené NC systémem

18.11.1.1 Odjíždění řízené NC systémem (POLF, POLFA, POLFMASK, POLFMLIN)

Pro odjíždění řízené NC systémem musí být splněny určité výchozí podmínky (viz "Odjíždění řízené NC systémem (POLF, POLFA, POLFMASK, POLFMLIN) (Strana 685)"). Jsou-li tyto výchozí předpoklady splněny, potom se pro osy provádějící odjíždění, které jsou nastavené v konfiguraci kanálu, aktivuje nastavením systémových proměnných \$AC_ESR_TRIGGER (příp. \$AA_ESR_TRIGGER pro samostatné osy) operace rychlého pozvednutí (LIFTFAST).

Syntaxe

```
POLF(<Osa>)=<Poloha>
POLFA(<Osa>,<Typ>,<Poloha>)
POLFMASK(<Osa_1>,<Osa_2>,...)
POLFMLIN(<Osa_1>,<Osa_2>,...)
```

U příkazu POLFA jsou přípustné následující zkrácené způsoby zápisu:

```
POLFA(<Osa>,<Typ>) ; zkrácená forma pro odjíždění samostatné osy
POLFA(Osa,0/1/2) ; rychlé deaktivování nebo aktivování
```

18.11 Rozšířené zastavování a odjíždění (ESR)

POLFA (Osa, 0, \$AA_POLFA [Osa]) ; způsobí zastavení předběžného zpracování
 POLFA (Osa, 0) ; nezpůsobí žádné zastavení předběžného zpracování

Význam

POLF:	Adresa pro zadání cílové pozice osy, která má odjíždět	
	Příkaz POLF má modální působnost..	
	<osa>:	Název geometrické nebo kanálové/strojní osy, která má provádět odjíždění
	<pozice>:	Pozice pro odjíždění Typ: REAL Pro geometrické osy platí WCS, jinak MCS. V případě stejných identifikátorů pro geometrické a kanálové/strojní osy se odjíždění provádí ve WCS.
POLFA:	Volání předem definovaného podprogramu pro zadání pozice pro zpětný pohyb pro jednotlivé osy	
	<osa>:	Identifikátor kanálové osy
	<typ>:	Režim zadávání polohy
		Typ: INT
		Hodnota: 0: Označení hodnoty polohy jako neplatné
		1: Hodnota polohy je absolutní
		2: Hodnota polohy je inkrementální (vzdálenost)
	Upozornění: Pokud osa není samostatnou osou nebo pokud chybí typ, příp. Typ = 0, aktivuje se odpovídající alarm.	
	<pozice>:	Pozice pro zpětný pohyb (viz výše) Upozornění: Hodnota pozice se přebírá, i když je nastaveno Typ = 0. Tato hodnota je jen potom označena jako neplatná, a aby se zpětný pohyb uskutečnil, musí být naprogramována znovu.
POLFMASK:	Volání předdefinovaného programu pro volbu os, u kterých se má po spuštění rychlého pozvednutí uskutečňovat zpětný pohyb nezávisle na sobě .	
	<Osa_1>, ...:	Názvy os, které mají při rychlém pozvednutí najíždět na pozice definované příkazem POLF. Všechny uvedené osy se musejí nacházet ve stejném souřadném systému.
	Příkaz POLFMASK () bez udání osy deaktivuje operaci rychlého pozvednutí pro všechny osy, které mají uskutečňovat zpětný pohyb nezávisle na sobě .	
POLFMLIN:	Volání předdefinovaného programu pro volbu os, u kterých se má po spuštění rychlého pozvednutí uskutečňovat zpětný pohyb v lineárním vztahu .	
	<Osa_1>, ...:	viz výše
	Příkaz POLFMLIN () bez udání osy deaktivuje operaci rychlého pozvednutí pro všechny osy, které mají uskutečňovat zpětný pohyb v lineárním vztahu .	

Poznámka

Předtím, než může být rychlé pozvednutí na pevně danou pozici pomocí příkazů `POLFMASK` nebo `POLFMLIN` uvolněno, musí být pro vybrané osy naprogramována pomocí příkazu `POLF` požadovaná pozice.

Poznámka

Jestliže jsou osy uvolněny postupně pomocí příkazů `POLFMASK`, `POLFMLIN` nebo `POLFMLIN`, `POLFMASK`, platí pro příslušnou osu vždy poslední definice.

Poznámka

Polohy naprogramované pomocí příkazu `POLF` a aktivování prostřednictvím funkce `POLFMASK` nebo `POLFMLIN` jsou při spuštění výrobního programu vymazány. To znamená, že uživatel musí v každém výrobním programu znovu naprogramovat hodnoty pro příkaz `POLF` a osy vybrané v příkazech `POLFMASK`, příp. `POLFMLIN`.

Poznámka

Pokud se při použití zkrácených forem příkazu `POLFA` změní pouze parametr `Typ`, pak musí uživatel zajistit, že parametry pro polohu zpětného pohybu nebo pro odjížděcí dráhu obsahují smysluplnou hodnotu. Polohu pro zpětný pohyb a odjížděcí dráhu je zapotřebí znovu nastavit zejména po spuštění systému (Power On).

Příklad:

Zpětný pohyb samostatné osy:

Programový kód	Komentář
MD37500 \$MA_ESR_REACTION[AX1]=21	; Odjíždění řízené NC systémem.
...	
\$AA_ESR_ENABLE[AX1]=1	
POLFA(AX1,1,20.0)	; Ose AX1 je přiřazena axiální poloha zpětného pohybu 20,0 (absolutně).
\$AA_ESR_TRIGGER[AX1]=1	; Od tohoto místa začíná zpětný pohyb.

Další informace**Předpoklady pro odjíždění řízené NC systémem**

- V konfiguraci kanálu je nastavena osa určená pro odjíždění řízené NC systémem:
MD37500 \$MA_ESR_REACTION = 21
- Pro tuto osu nesmí být ESR blokováno:
\$AA_ESR_ENABLE = 1
- Jsou definována časová zpoždění:
MD21380 \$MC_ESR_DELAY_TIME1
MD21381 \$MC_ESR_DELAY_TIME2

- Ve výrobním programu jsou pro příslušné osy pomocí příkazu `POLF` naprogramovány pozice pro zpětný pohyb.
- Osy jsou pomocí příkazů `POLFMASK`/`POLFMLIN` vybrány pro odjíždění řízené NC systémem.
- Pro zpětný pohyb musí být nastaveny signály zajišťující odblokování a tyto signály musí v tomto stavu zůstat.

Uvolnění a spouštění odjíždění řízeného NC systémem

Jestliže je nastavena systémová proměnná `$AC_ESR_TRIGGER = 1` a jestliže je v konfiguraci tohoto kanálu nastavena osa pro zpětný pohyb (tzn. `MD37500 $MA_ESR_REACTION = 21`) a pro tuto osu je nastavena proměnná `$AA_ESR_ENABLE = 1`, pak se v tomto kanálu aktivuje rychlé pozvednutí (`LIFTFAST`).

Pozvedávací pohyb, jehož konfigurace je nastavena pomocí funkcí `POLF` (příp. `LFPOS`), os vybraných pomocí příkazů `POLFMASK` nebo `POLFMLIN` nahrazuje pro tyto osy pohyb po dráze definovaný ve výrobním programu.

Pro odjetí je k dispozici maximálně součet časů `MD21380 $MC_ESR_DELAY_TIME1` a `MD21381 $MC_ESR_DELAY_TIME2`. Po uplynutí tohoto řasového intervalu se i pro osy provádějící zpětný pohyb spustí rychlé zabrzdění s následným monitorováním.

Poznámka

Rozšířený zpětný pohyb (tzn. pomocí proměnné `$AC_ESR_TRIGGER` spuštěná funkce `LIFTFAST/LFPOS`) **nemůže být přerušen** a může být předčasně ukončen pouze nouzovým zastavením.

Poznámka

Odjíždění spouštěné pomocí proměnné `$AC_ESR_TRIGGER` je blokováno vůči několikanásobnému zpětnému pohybu.

Odjíždění samostatnou osou

Při odjíždění samostatnou osou musí být pomocí příkazu `POLFA` naprogramována pozice samostatné osy pro zpětný pohyb a musí být splněny následující podmínky:

- `$AA_ESR_ENABLE = 1`
- <Osa> musí být v okamžiku spuštění (`$AA_ESR_TRIGGER = 1`) samostatnou osou.
- Parametr <Typ> musí mít hodnotu 1 nebo 2.

Směr odjíždění při rychlém pozvednutí

V okamžiku aktivování rychlého pozvednutí se bere ohled na momentálně platný frame.

Poznámka

Framy s otočením ovlivňují pomocí příkazu `POLF` také směr, ve kterém se pozvednutí nástroje uskutečňuje.

Výměna osy

Osy provádějící zpětný pohyb jsou vždy přiřazeny právě jednomu kanálu a nesmí být vyměňovány mezi kanály. Dojde-li k pokusu vyměnit osu provádějící odjíždění do jiného kanálu, aktivuje se alarm. Teprve poté, co byla osa pomocí proměnné \$AA_ESR_ENABLE[AX] = 0 opět deaktivována, může být vyměněna do nového kanálu. Po dokončení výměny mohou být osy znovu pomocí nastavení proměnné \$AA_ESR_ENABLE[AX] = 1 ovládány.

Neutrální osy

Neutrální osy nejsou schopny ESR řízené NC systémem uskutečňovat.

18.11.1.2 Zastavení NC systémem

Pro osy provádějící zastavení, které jsou nastavené v konfiguraci kanálu, aktivuje nastavením systémových proměnných \$AC_ESR_TRIGGER (příp. \$AA_ESR_TRIGGER pro samostatné osy) operace zastavení řízeného NC systémem.

Předpoklady

- V konfiguraci kanálu je nastavena osa určená pro zastavení řízené NC systémem:
MD37500 \$MA_ESR_REACTION = 22
- Pro tuto osu nesmí být ESR blokováno:
\$AA_ESR_ENABLE = 1
- Jsou definována časová zpoždění:
MD21380 \$MC_ESR_DELAY_TIME1 (časové zpoždění pro ESR os)
MD21381 \$MC_ESR_DELAY_TIME2 (čas ESR pro interpolační brždění)

Postup

Po dobu časového intervalu v parametru MD21380 provádí osa nerušený interpolační pohyb, jak bylo naprogramováno. Po uplynutí časového intervalu daného parametrem MD21380 se spustí brždění řízené pomocí interpolace. Pro brždění řízené pomocí interpolace je potom k dispozici maximálně časové interval daný parametrem MD21381. Po uplynutí tohoto časového intervalu se spouští rychlé brždění s následným monitorováním.

Příklad:

Zastavení samostatné osy:

Programový kód	Komentář
MD37500 \$MC_ESR_REACTION[AX1]=22	; zastavení řízené NC systémem
MD21380 \$MC_ESR_DELAY_TIME1[AX1]=0.3	
MD21381 \$MC_ESR_DELAY_TIME2[AX1]=0.06	
...	
\$AA_ESR_ENABLE[AX1]=1	
\$AA_ESR_TRIGGER[AX1]=1	; od tohoto místa začíná zastavová- ní

18.11.2 ESR nezávislé na pohonu

18.11.2.1 Konfigurace zastavování nezávislého na pohonu (ESRS)

Pomocí funkce `ESRS(...)` se nastavuje konfigurace parametrů pohonu pro "zastavování" v rámci funkce ESR nezávislé na pohonu.

Syntaxe

```
ESRS(<osa_1>,<doba zastavení_1>[,...,<osa_n>,<doba zastavení_n>])
```

Význam

<code>ESRS(...):</code>	Funkce pro zápis parametrů pohonu pro funkci ESR "zastavení" Funkce: • musí se nacházet v samostatném bloku • spouští předběžné zpracování • nemůže se používat v synchronních akcích	
<code><osa_1>, ..., <osa_n>:</code>	Osa, pro kterou má být konfigurováno zastavování nezávislé na pohonu. V pohonu se pro tuto osu zapisuje parametr pohonu p0888 (konfigurace): p0888 = 1	
	Typ:	AXIS
	Rozsah hodnot:	Identifikátor kanálové osy
<code><doba zastavení_1>, ..., <doba zastavení_n>:</code>	Časový interval po který se pohon poté, co se vyskytla chyba, pohybuje konstantně s momentální požadovanou hodnotou otáček. V pohonu se pro uvedenou osu zapisuje parametr pohonu p0892 (časový interval): p0892 = <doba zastavení>	
	Jednotka:	s
	Typ:	REAL
	Rozsah hodnot:	0.00 - 20.00
V jednom volání funkce může být naprogramováno maximálně 5 os; n = 5		

18.11.2.2 Konfigurace odjíždění nezávislého na pohonu (ESRR)

Pomocí funkce `ESRR(...)` se nastavuje konfigurace parametrů pohonu pro "odjíždění" v rámci funkce ESR nezávislé na pohonu.

Syntaxe

```
ESRR(<osa_1>,<návratová dráha_1>,<návratová  
rychlost_1>[,...,<osa_n>,<návratová dráha_n>,<návratová  
rychlost_n>])
```

Význam

ESRR (...):	Funkce pro zápis parametrů pohonu pro funkci ESR "odjíždění" Funkce: • musí se nacházet v samostatném bloku • spouští předběžné zpracování • nemůže se používat v synchronních akcích	
<code><osa_1>, ... <osa_n>:</code>	Osa, pro kterou má být konfigurováno odjíždění nezávislé na pohonu V pohonu se pro tuto osu zapisuje parametr pohonu p0888 (konfigurace): $p0888 = 2$	
	Typ:	AXIS
	Rozsah hodnot:	Identifikátor kanálové osy
<code><návratová dráha_1>, ... <návratová dráha_n>:</code>	Návratová dráha se pro pohon přepočítává na návratové otáčky. Hodnota se pro uvedenou osu zapisuje do parametru pohonu p0893 (otáčky): $p0893 = (<návratová dráha_n> \text{ přepočítaná na návratové otáčky})$	
	Jednotka:	mm/min, palce/min, stupně/min (v závislosti na jednotkách nastavených pro osu)
	Typ:	REAL
	Rozsah hodnot:	MIN - MAX
<code><návratová rychlost_1>, ... <návratová rychlost_n>:</code>	Návratová rychlost se pro pohon přepočítává na časový interval. Hodnota se pro uvedenou osu zapisuje do parametru pohonu p0892 (časový interval) [s]: $p0892 = <návratová dráha_n> / <návratová rychlost_n>$	
	Jednotka:	mm/min, palce/min, stupně/min (v závislosti na jednotkách nastavených pro osu)
	Typ:	REAL
	Rozsah hodnot:	0.00 - MAX
V jednom volání funkce může být naprogramováno maximálně 5 os; n = 5		

18.12 Definice surového obrobku (WORKPIECE)

Řídící systém musí znát tvar a velikost surového obrobku, aby jej mohl zobrazovat v rámci grafické simulace. Uživatel má proto možnost definovat surový obrobek prostřednictvím uživatelského rozhraní nebo přímo v rámci NC programu. Definice surového obrobku zůstává zachována i po resetu (konec programu/reset kanálu nebo BAG). V průběhu následujícího rozběhu řídicího systému se automaticky vymaže.

Syntaxe

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

Význam

WORKPIECE (...):		Předdefinovaná procedura pro definici surového obrobku	
		Zastavení předběžného zpracování:	ano
		Samostatný blok:	ano
Parametry:			
1	"<WP>":	Název obrobku (nepovinné)	
		Datový typ:	STRING
		Tento údaj je zapotřebí jen tehdy, jestliže v jednom kanálu může být zadáno více obrobků. Pokud není parametr zadán, automaticky se přiřadí „WORKP<n>“, kde <n> je číslo deklarovaného kanálu.	
2	"<RefP>":	Upnutí (nepovinné; jen v případě frézek)	
		Datový typ:	STRING
		Rozsah hodnot:	"Tisch" Upnutí na pevném stole
			"A" Upnutí na kruhové ose A
			"B" Upnutí na kruhové ose B
			"C" Upnutí na kruhové ose C
		Předpoklad: Stůl, příp. kruhová osa musí být prostřednictvím odpovídajícího strojního parametru pro upínání obrobků odblokovány (viz Příručka pro uvedení do provozu, SINUMERIK Operate).	
3	"<ZeroOffset>":	Nastavitelné posunutí počátku pro umístění surového obrobku (nelze naprogramovat) Volba nastavitelného posunutí počátku pro umístění surového obrobku je nabízena pouze při zadávání surového obrobku prostřednictvím uživatelského rozhraní. Při přímé definici surového obrobku v rámci výrobního programu se oproti tomu surový obrobek vždy vztahuje na momentálně platné posunutí počátku.	

18.12 Definice surového obrobku (WORKPIECE)

4	" <Type> ":	Tvar surového obrobku		
		Datový typ:	STRING	
		Rozsah hodnot:	"CYLINDER":	Válec
			"PIPE":	Trubka
			"RECTANGLE":	Středově upnutý kvádr
			"BOX":	Kvádr
	"N_CORNER":	N-hran		
5 ... 12	<Par5> ... <Par12>:	Parametry pro popis tvaru surového obrobku		
		Datový typ:	REAL	
		Počet potřebných parametrů a jejich význam jsou závislé na příslušném tvaru surového obrobku a na hodnotě bitových parametrů. Viz: • Tabulka „Parametry pro popis tvaru surového obrobku“ • Tabulka „Bitové parametry“		
WORPIECE () :		Voláním příkazu WORKPIECE bez parametrů se vymažou všechny definice surového obrobku.		
WORPIECE (<WP>) :		Voláním příkazu WORKPIECE s názvem obrobku se vymaže jen tato definice surového obrobku.		

Tabulka 18-1 Parametry pro popis tvaru surového obrobku

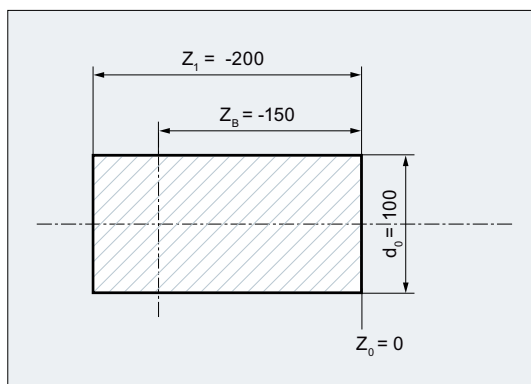
Tvar surového obrobku	Parametry							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
Válec	Bitový parametr Reálná hodnota, která je interpretována jako bitově kódované číslo typu Integer. Tyto bity definují význam následujících parametrů (viz tabulka „Bitové parametry“).	Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Vnější průměr d_0	-	Otočení okolo kruhové osy	-
Trubka		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Vnější průměr d_0	Tloušťka stěny (ink) / vnější průměr d_1 (abs)	Otočení okolo kruhové osy	-
Středově upnutý kvádr		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Šířka W	Délka L	Otočení okolo kruhové osy	-
Kvádr		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	X_0	Y_0	X_1	Y_1
N-hran		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Počet hran	Šířka klíče	Otočení okolo kruhové osy	-

Tabulka 18-2 Bitový parametr

Bit	Význam	
4 (0x0010)	Kvadr: X_1	
	= 0	ink
	= 1	abs
5 (0x0020)	Kvadr: Y_1	
	= 0	ink
	= 1	abs
6 (0x0040)	Délka Z_1 (konečný rozměr)	
	= 0	ink
	= 1	abs
Bit 7 (0x0080)	Rozměr pro opracování Z_B	
	= 0	ink
	= 1	abs
Bit 8 (0x0100)	Trubka: Tloušťka stěny/vnitřní průměr	
	= 0	ink
	= 1	abs
9 (0x0200)	N-hran	
	= 0	Šířka klíče
	= 1	Délka hrany
12 (0x1000)	Upnutí pro soustruhy	
	= 0	Hlavní vřeteno
	= 1	Protivřeteno
13 (0x2000)	Protivřeteno	
	= 0	se zrcadlovým převrácením
	= 1	bez zrcadlového převrácení

Příklady

Příklad 1: Válcový surový obrobek na soustruhu



Programový kód

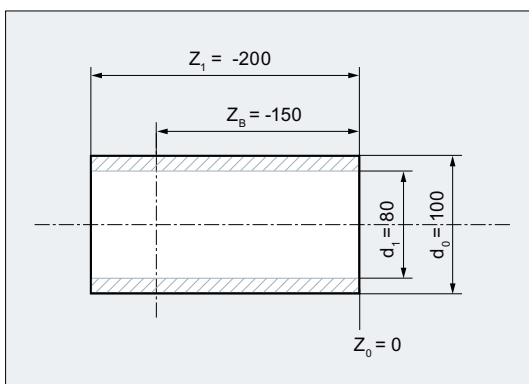
Komentář

...

18.12 Definice surového obrobku (WORKPIECE)

Programový kód	Komentář
WORKPIECE(,,, "CYLINDER",0,0,-200,-150,100)	; definice surového obrobku
...	; tvar surového obrobku: Válec
	; bitový parametr=0 (není nastaven žádný bit) → hodnoty pro délku a rozměr pro opracování jsou inkrementální, surový obrobek v hlavním vřetenu
	; vztažný bod (Z0)=0
	; délka e(Z1)=-200
	; rozměr pro opracování (ZB)=-150
	; vnější průměr (d0)=100

Příklad 2: Trubkový surový obrobek na soustruhu



Programový kód	Komentář
...	; definice surového obrobku
WORKPIECE(,,, "PIPE",256,0,-200,-150,100,80)	; tvar surového obrobku: Trubka
	; bitový parametr=256(Bit8=1) → vnitřní průměr je absolutní, délka a rozměr pro opracování jsou inkrementální, surový obrobek v hlavním vřetenu
	; vztažný bod (Z0)=0
	; délka e(Z1)=-200
	; rozměr pro opracování (ZB)=-150
	; vnější průměr (d0)=100
	; vnitřní průměr (d1)=80
...	

18.13 Přepnutí režimu jazyka (G290, G291)

Řídící systém nabízí možnost načítat a zpracovávat výrobní programy z externích CNC systémů. Předpokladem je, že při uvádění systému do provozu byl definován odpovídající režim jazyka NC systému (dialekt ISO).

Literatura:

Příručka k funkcím, Dialekty ISO

Režim dialektu ISO může být aktivován pro každý kanál odděleně. Například kanál 1 může pracovat v režimu dialektu ISO, zatímco v kanálu 2 je aktivní režim systému SINUMERIK.

Přepnutí mezi režimem systému SINUMERIK a režimem dialektu ISO se uskutečňuje v NC programu pomocí příkazů G-funkcí ze skupiny 47. Aktivní nástroj, korekční parametry nástroje a posunutí počátku (nuly) nejsou přepnutím nijak ovlivněny.

Syntaxe

```
G291
...
G290
```

Význam

G290:	Aktivování režimu jazyka systému SINUMERIK	
	Samostatný blok:	ano
	Platnost:	modální
G291:	Aktivování režimu jazyka ISO	
	Samostatný blok:	ano
	Platnost:	modální

Podmínky

Režim systému SINUMERIK

- Předdefinované nastavení příkazů G-funkcí může být pro každý kanál definováno pomocí strojního parametru.
- V režimu systému SINUMERIK nemohou být naprogramovány žádné příkazy jazyka z dialektu ISO.

Režim dialektu ISO

- Režim dialektu ISO jako základní nastavení řídicího systému může být nastaven pomocí strojních parametrů. Řídící systém se potom standardně spouští v režimu dialektu ISO.
- Mohou být programovány pouze příkazy G-funkcí z dialektu ISO. Programování funkcí systému SINUMERIK není v režimu dialektu ISO možné.
- Směšování dialektu ISO a jazyka systému SINUMERIK v jednom NC bloku není možné.
- Přepínání prostřednictvím příkazů G-funkce mezi dialektem ISO M (frézování) a dialektem ISO T (soustružení) není možné.

- Mohou být vyvolávány podprogramy, které jsou naprogramovány v režimu systému SINUMERIK.
- Jestliže se mají používat funkce systému SINUMERIK, je nutné napřed přepnout do režimu systému SINUMERIK (viz příklad).

Příklad

Kompresse lineárních bloků v režimu dialektu ISO

Programový kód	Komentář
N5 G290	; aktivování režimu jazyka systému SINUMERIK
N10 COMPON	; COMPON je příkaz jazyka firmy Siemens, který aktivuje funkci kompresoru, která po sobě následující lineární bloky nahrazuje bloky polynomů s co možno nejdelšími úseky dráhy.
N15 G291	; aktivování režimu jazyka ISO
N20 G01 X100 Y100 F1000	; Protože byl příkaz COMPON aktivován v režimu systému SINUMERIK, mohou být pomocí této funkce komprimovány také lineární bloky v režimu dialektu ISO.
...	

Vlastní programy pro oddělování třísky

19.1 Podporované funkce pro oddělování třísky

Pro oddělování třísky jsou Vám nabízeny již hotové obráběcí cykly. Kromě toho máte možnost pomocí funkcí uvedených v následujících odstavcích sestavovat své vlastní programy pro obrábění:

- Sestavování kontury (CONTPRON)
- Sestavování kódované tabulky kontury (CONTPRON)
- Ukončení přípravy kontury (EXECUTE)
- Zjištění průsečíku mezi dvěma konturovými prvky (INTERSEC)
(jen pro tabulky, které byly vytvořeny pomocí funkce CONTPRON.)
- Zpracovávání konturových prvků v tabulce blok po bloku (EXECTAB)
(jen pro tabulky, které byly vytvořeny pomocí funkce CONTPRON.)
- Výpočet parametrů kruhu (CALCDAT)

Poznámka

Tyto funkce můžete používat nejen pro oddělování třísky, jsou použitelné univerzálně.

Předpoklady

Před vyvoláváním funkcí CONTPRON nebo CONTDCON musíte:

- Najet na počáteční bod, který umožňuje bezkolizní obrábění.
- Deaktivovat korekci rádiusu nástroje pomocí příkazu G40.

19.2 Sestavování kontury (CONTPRON)

Pomocí příkazu `CONTPRON` aktivujete funkci pro přípravu kontury. Následně vyvolávané NC-bloky nejsou zpracovávány, ale jsou rozděleny do jednotlivých pohybů a uloženy do tabulky kontury. Každému konturovému prvku odpovídá řádek ve dvourozměrném poli tabulky kontury. Systém uvádí počet zjištěných podříznutí.

Syntaxe

Aktivování přípravy kontury:

```
CONTPRON(<tabulka kontury>,<druh opracování>,<podříznutí>,<směr opracování>)
```

Deaktivování přípravy kontury a návrat zpět do normálního režimu zpracování:

```
EXECUTE (<FEHLER>)
```

Viz "Ukončení přípravy kontury (EXECUTE) (Strana 715)".

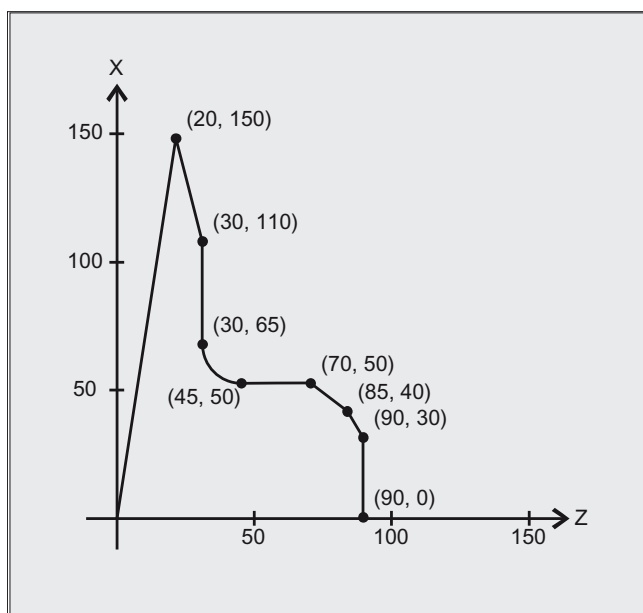
Význam

CONTPRON:	Předdefinovaná procedura pro aktivování přípravy kontury za účelem sestavení tabulky kontury		
<tabulka kontury>:	Název tabulky kontury		
<druh opracování>:	Parametr pro stanovení druhu opracování		
	Typ:	CHAR	
	Hodnota:	"G":	podélné soustružení: vnitřní opracování
		"L":	podélné soustružení: vnější opracování
		"N":	příčné soustružení: vnitřní opracování
"P":		příčné soustružení: vnější opracování	
<podříznutí>:	Varianta výsledku pro počet zjištěných prvků podříznutí		
	Typ:	INT	
<směr opracování>:	Parametr pro stanovení směru opracování		
	Typ:	INT	
	Hodnota:	0	Ve směru přípravy kontury (standardní hodnota)
		1	V obou směrech vzhledem ke směru přípravy kontury

Příklad 1

Sestavení tabulky kontury s následujícími parametry:

- Název "KTAB"
- max. 30 konturových prvků (kruhy, přímky)
- Proměnná pro počet zjištěných prvků podříznutí
- Proměnná pro chybová hlášení



NC program:

Programový kód	Komentář
N10 DEF REAL KTAB[30,11]	; tabulka kontury s názvem KTAB a s max. 30 konturovými prvky, hodnota parametru 11 (počet sloupců v tabulce) je pevná hodnota
N20 DEF INT ANZHINT	; proměnná pro počet prvků podříznutí s názvem ANZHINT
N30 DEF INT FEHLER	; proměnná pro zpětná chybová hlášení (0 = žádná chyba, 1 = chyba)
N40 G18	
N50 CONTPRON(KTAB,"G",ANZHINT)	; aktivování přípravy kontury
N60 G1 X150 Z20	; N60 až N120: popis kontury
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(FEHLER)	; ukončení plnění tabulky kontury, přepnutí do normálního programového režimu
N140 ...	; další zpracovávání tabulky

Tabulka kontury KTAB:

Index Řádek	Sloupec									
(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

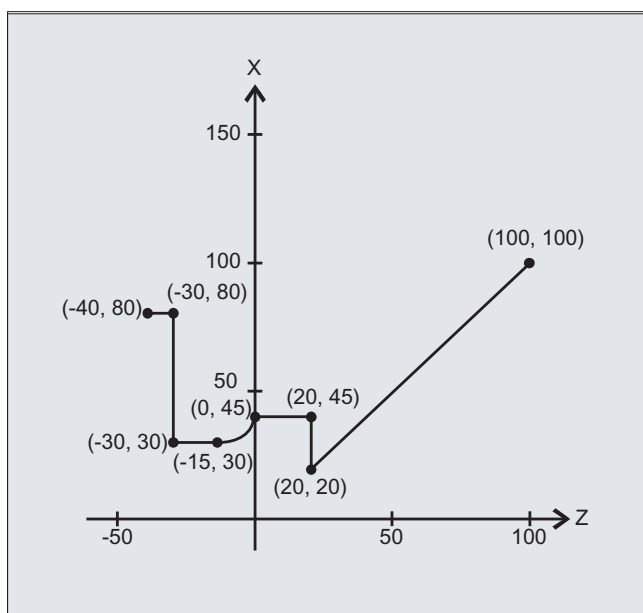
Vysvětlení obsahu sloupců:

- (0) Ukazatel na nejbližší konturový prvek (na číslo jeho řádku)
- (1) Ukazatel na předcházející konturový prvek
- (2) Kódování režimu kontury pro pohyb
Možné hodnoty pro X = abc
 $a = 10^2$ $G90 = 0$ $G91 = 1$
 $b = 10^1$ $G70 = 0$ $G71 = 1$
 $c = 10^0$ $G0 = 0$ $G1 = 1$ $G2 = 2$ $G3 = 3$
- (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 max./min.: označuje lokální maxima a minima na kontuře
- (8) Maximální hodnota mezi konturovým prvkem a abscisou (při podélném obrábění), příp. ordinátou (při příčném obrábění). Úhel je závislý na naprogramovaném druhu opracování.
- (9), (10) Souřadnice středu konturového prvku, jestliže se jedná o kruhový blok.
(9) = abscisa, (10) = ordináta

Příklad 2

Sestavení tabulky kontury s následujícími parametry:

- Název KTAB
- max. 92 konturových prvků (kruhy, přímky)
- Provozní režim: podélné obrábění, vnější obrábění
- Příprava směrem dopředu a dozadu



NC program:

Programový kód	Komentář
N10 DEF REAL KTAB[92,11]	; tabulka kontury s názvem KTAB a s max. 92 konturovými prvky, hodnota parametru 11 je pevná hodnota
N20 DEF CHAR BT="L"	; druh opracování CONTPRON: podélné obrábění, vnější obrábění
N30 DEF INT HE=0	; počet prvků podříznutí = 0
N40 DEF INT MODE=1	; příprava směrem dopředu a dozadu
N50 DEF INT ERR=0	; chybové hlášení
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; aktivování 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
...	

Tabulka kontury KTAB:

Po ukončení přípravy kontury je kontura k dispozici v obou směrech.

Index	Sloupec										
Řádek	(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 sloupců a poznámky k řádkům 0, 1, 6, 8, 83, 85 a 91

Pokud jde o obsah sloupců, platí vysvětlení uvedená u příkladu 1.

Vždy v řádku tabulky 0:

- 1) Předcházející blok: Řádek n obsahuje konec tabulky kontury směrem dopředu
- 2) Následující blok: Řádek n je koncem tabulky kontury směrem dopředu

Pokaždé v rámci prvku kontury směrem dopředu:

- 3) Předcházející blok: začátek kontury (směrem dopředu)
- 4) Následující blok: konec kontury (směrem dopředu)

Vždy na řádku konec tabulky kontury (směrem dopředu) + 1:

- 5) Předcházející blok: Počet podříznutí směrem dopředu
- 6) Následující blok: Počet podříznutí směrem dozadu

Pokaždé v rámci prvku kontury směrem dozadu:

- 7) Následující blok: konec kontury (směrem dozadu)
- 8) Předcházející blok: začátek kontury (směrem dozadu)

Vždy na posledním řádku tabulky:

- 9) Předcházející blok: Řádek n je počátek tabulky kontury (směrem dozadu)

10) Následující blok: Řádek n obsahuje počátek kontury (směrem dozadu)

Další informace

Povolené příkazy pohybu, souřadný systém

Pro programování kontury jsou přípustné následující příkazy G-funkcí:

- G-skupina 1: G0, G1, G2, G3

Kromě toho jsou k dispozici následující možnosti:

- Zaoblení a faseta
- Programování kruhových oblouků pomocí příkazů CIP a CT

Funkce pro spliny, polynomy a závity mají za následek chybu.

Změny souřadného systému prostřednictvím aktivování framu jsou v úseku mezi příkazy CONTPRON a EXECUTE nepřipustné. Totéž platí i v případě přepnutí mezi funkcemi G70 a G71, příp. G700 a G710.

Výměna geometrických os pomocí příkazu GEOAX v době, kdy probíhá příprava tabulky kontury má za následek alarm.

Prvky podříznutí

Popis kontury jednotlivých prvků podříznutí se může dle Vašeho přání uskutečnit buď v podprogramu nebo v jednotlivých blocích.

Oddělování třísky nezávisle na naprogramovaném směru kontury

Příprava kontury pomocí funkce CONTPRON byla rozšířena tak, aby po svém vyvolání byla tabulka kontury k dispozici nezávisle na směru, ve kterém byla naprogramována.

19.3 Sestavování kódované tabulky kontury (CONTPRON)

Při přípravě kontury aktivované příkazem `CONTDCON` jsou následně vyvolávané NC-bloky ukládány kódovaně do tabulky se 6 sloupci, což je výhodné z hlediska využití paměti. Každému konturovému prvku odpovídá řádek v tabulce kontury. Ze znalosti níže uvedených pravidel pro kódování můžete např. z řádků tabulky sestavit program v kódu DIN pro cyklus. V řádku tabulky s číslem 0 jsou uložena data výchozího bodu.

Syntaxe

Aktivování přípravy kontury:

`CONTDCON(<tabulka kontury>,<směr zpracování>)`

Deaktivování přípravy kontury a návrat zpět do normálního režimu zpracování:

`EXECUTE (<FEHLER>)`

Viz "Ukončení přípravy kontury (EXECUTE) (Strana 715)".

Význam

<code>CONTDCON:</code>	Předdefinovaná procedura pro aktivování přípravy kontury za účelem sestavení kódované tabulky kontury		
<code><tabulka kontury>:</code>	Název tabulky kontury		
<code><směr zpracování>:</code>	Parametr pro stanovení směru zpracování		
	Typ:	INT	
	Hodnota:	0	Příprava kontury podle posloupnosti konturových bloků (standardní hodnota)
		1	nepřípustné

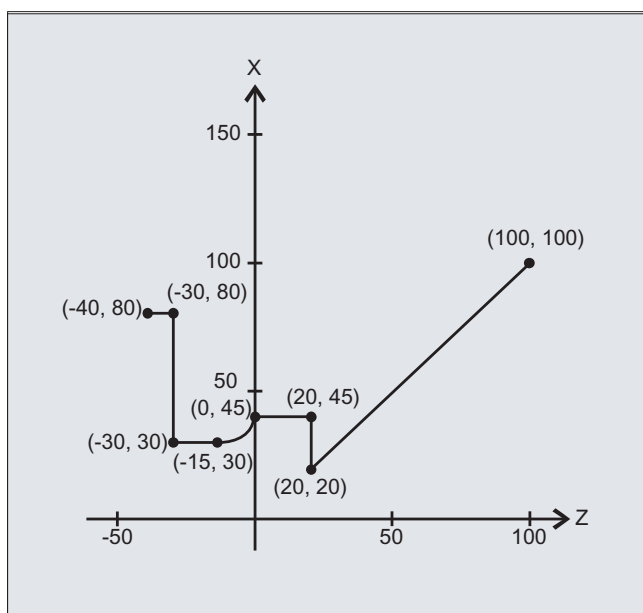
Poznámka

Spektrum příkazů G-funkcí, které jsou přípustné pro funkci `CONTDCON` v úseku programu, který se má uložit do tabulky, je širší než v případě příkazu `CONTPRON`. Kromě toho jsou s každým úsekem kontury ukládány také posuvy a typ posuvu.

Příklad:

Sestavení tabulky kontury s následujícími parametry:

- Název "KTAB"
- Konturové prvky (kruhy, přímky)
- Provozní režim: Soustružení
- Směr obrábění: Směrem dopředu

**NC program:**

Programový kód	Komentář
N10 DEF REAL KTAB[9,6]	; Tabulka kontury s názvem KTAB a s 9 řádky, které umožňují 8 konturových bloků. Hodnota parametru 6 (počet sloupců v tabulce) je pevně danou velikostí.
N20 DEF INT MODE = 0	; proměnná pro stanovení směru opracování Standardní hodnota 0: jen v naprogramovaném směru kontury.
N30 DEF INT ERROR = 0	; proměnná pro chybové hlášení
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; vyvolání přípravy tabulky (parametr MODE lze vypustit)
N110 G1 Z20 X20 F200	; popis kontury
N120 G9 X45 F300	
N130 Z0 F400	
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
...	

Tabulka kontury KTAB:

	Index sloupce					
	0	1	2	3	4	5
Index řádku	Režim kontury	Koncový bod abscisa	Koncový bod ordináta	Střed na abscise	Střed na ordinátě	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

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, abscisa
- Sloupec 2: Počáteční bod, ordináta
- Sloupec 3-4: 0
- Sloupec 5: Index řádku posledního úseku kontury v tabulce

Řádky 1-n: Údaje reprezentující **úseky kontury**

- Sloupec 0: 10^0 (místo jednotek): G0 = 0, G1 = 1, G2 = 2, G3 = 3
 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 desítek tisíců): G60 = 0, G44 = 1, G641 = 2, G642 = 3
 10^5 (místo stovek tisíců): G9 = 1
- Sloupec 1: Koncový bod, abscisa
- Sloupec 2: Koncový bod, ordináta
- Sloupec 3: Střed kruhu na abscise při kruhové interpolaci
- Sloupec 4: Střed kruhu na ordinátě při kruhové interpolaci
- Sloupec 5: Posuv

Další informace

Povolené příkazy pohybu, souřadný systém

Pro programování kontury jsou přípustné následující G-skupiny a následující příkazy G-funkcí:

G-skupina 1:	G0, G1, G2, G3
G-skupina 10:	G60, G64, G641, G642
G-skupina 11:	G9
G-skupina 13:	G70, G71, G700, G710
G-skupina 14:	G90, G91
G-skupina 15:	G93, G94, G95, G96, G961

Kromě toho jsou k dispozici následující možnosti:

- Zaoblení a faseta
- Programování kruhových oblouků pomocí příkazů CIP a CT

Funkce pro spliny, polynomy a závity mají za následek chybu.

Změny souřadného systému prostřednictvím aktivování framu jsou v úseku mezi příkazy `CONTDCON` a `EXECUTE` nepřipustné. Totéž platí i v případě přepnutí mezi funkcemi G70 a G71, příp. G700 a G710.

Výměna geometrických os pomocí příkazu `GEOAX` v době, kdy probíhá příprava tabulky kontury má za následek alarm.

Směr obrábění

Při opracovávání pomocí tabulky kontury sestavené pomocí funkce `CONTDCON` se smí postupovat ve směru naprogramování kontury.

19.4 Zjištění průsečíku mezi dvěma konturovými prvky (INTERSEC)

Funkce `INTERSEC` zjišťuje dvou normovaných konturových prvků z tabulek kontury sestavených pomocí funkce `CONTPRON`.

Syntaxe

```
<Status>=INTERSEC(<tabulka kontury_1>[<prvek kontury_1>],  
<tabulka kontury_2>[<prvek kontury_2>],<průsečík>,<druh zpracování>)
```

Význam

INTERSEC:	Předem definovaná funkce pro výpočet průsečíku dvou konturových prvků z tabulek kontury vytvořených pomocí funkce <code>CONTPRON</code> .		
<Status>:	Proměnná pro stav průsečíku		
	Typ:	BOOL	
	Hodnota:	TRUE	Průsečík nalezen
		FALSE	Žádný průsečík nenalezen
<tabulka kontury_1>:	Název první tabulky kontury		
<prvek kontury_1>:	Číslo konturového prvku z první tabulky		
<tabulka kontury_2>:	Název druhé tabulky kontury		
<prvek kontury_2>:	Číslo konturového prvku ze druhé tabulky		
<průsečík>:	Souřadnice průsečíku v aktivní rovině (G17 / G18 / G19)		
	Typ:	REAL	
<druh zpracování>:	Parametr pro stanovení druhu zpracování		
	Typ:	INT	
	Hodnota:	0	Výpočet průsečíku v aktivní rovině dané parametrem 2 (standardní hodnota)
		1	Výpočet průsečíku nezávisle na předávané rovině

Poznámka

Mějte prosím na paměti, že proměnné musí být před svým použitím definovány.

Předávání kontur vyžaduje, aby byly dodrženy hodnoty definované příkazem `CONTPRON`.

Parametr	Význam
2	Kódování režimu kontury pro pohyb
3	Počáteční bod kontury, abscisa
4	Počáteční bod kontury, ordináta
5	Koncový bod kontury, abscisa
6	Koncový bod kontury, ordináta
9	Souřadnice středu pro abscisu (jen v případě kruhové kontury)
10	Souřadnice středu pro ordinátu (jen v případě kruhové kontury)

Příklad:

Zjištění průsečíku konturového prvku 3 z tabulky TABNAME1 a konturového prvku 7 z tabulky TABNAME2. Souřadnice průsečíku v aktivní rovině jsou uloženy v proměnné ISCOORD (1. prvek = abscisa, 2. prvek = ordináta). Jestliže neexistuje žádný průsečík, uskuteční se skok na návěští KEINSCH (žádný průsečík nenalezen).

Programový kód	Komentář
DEF REAL TABNAME1[12,11]	; tabulka kontury 1
DEF REAL TABNAME2[10,11]	; tabulka kontury 2
DEF REAL ISCOORD[2]	; proměnná pro souřadnice průsečíku
DEF BOOL ISPOINT	; proměnná pro stav průsečíku
DEF INT MODE	; proměnná pro druh obrábění
...	
MODE=1	; výpočet nezávisle na aktivní rovině
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)	; volání průsečíku konturových prvků
N20 IF ISPOINT==FALSE GOTO KEINSCH	; skok na návěští KEINSCH
...	

19.5 Pohyb po konturových prvcích v tabulce blok po bloku (EXECTAB)

Pomocí příkazu `EXECTAB` můžete uskutečnit pohyb blok po bloku po konturových prvcích tabulky, která byla vytvořena například příkazem `CONTPRON`.

Syntaxe

```
EXECTAB(<tabulka kontury>[<konturový prvek>])
```

Význam

EXECTAB:	Předdefinovaná procedura pro aktivování pohybu po konturových prvcích
<tabulka kontury>:	Název tabulky kontury
<konturový prvek>:	Číslo konturového prvku

Příklad:

Má být uskutečněn pohyb blok po bloku po konturových prvcích 0 až 2 z tabulky KTAB.

Programový kód	Komentář
N10 EXECTAB(KTAB[0])	; pohyb po prvku 0 z tabulky kontury KTAB
N20 EXECTAB(KTAB[1])	; pohyb po prvku 1 z tabulky kontury KTAB
N30 EXECTAB(KTAB[2])	; pohyb po prvku 2 z tabulky kontury KTAB

19.6 Výpočet parametrů kruhu (CALCDAT)

Pomocí příkazu `CALCDAT` můžete ze tří nebo čtyř známých bodů na kružnici vypočítat její rádius a souřadnice jejího středu. Zadávané body musí být různé.

V případě 4 bodů, které neleží na kružnici úplně přesně, se pro střed a rádius kružnice vypočítá střední hodnota.

Poznámka

Matematický předpis pro výpočet střední hodnoty

Výpočet kruhu se uskuteční 4x:

1. s body na kružnici 1, 2, 3
2. s body na kružnici 1, 2, 4
3. s body na kružnici 1, 3, 4
4. s body na kružnici 2, 3, 4

Hodnota abscisy a hodnota ordináty pro souřadnice středu kruhu se vypočítají tak, že se hodnoty abscisy, příp. hodnoty ordináty, ze čtyř výše uvedených výpočtů kruhu sečtou a podělí 4.

Rádius se vypočítá tak, že se určí odmocnina se součtu čtyř rádiusů dříve vypočítaných kruhů a výsledek se vynásobí 0,5.

Syntaxe

`<Status>=CALCDAT(<body kružnice>[<počet>,<druh>],<počet>,<výsledek>)`

Význam

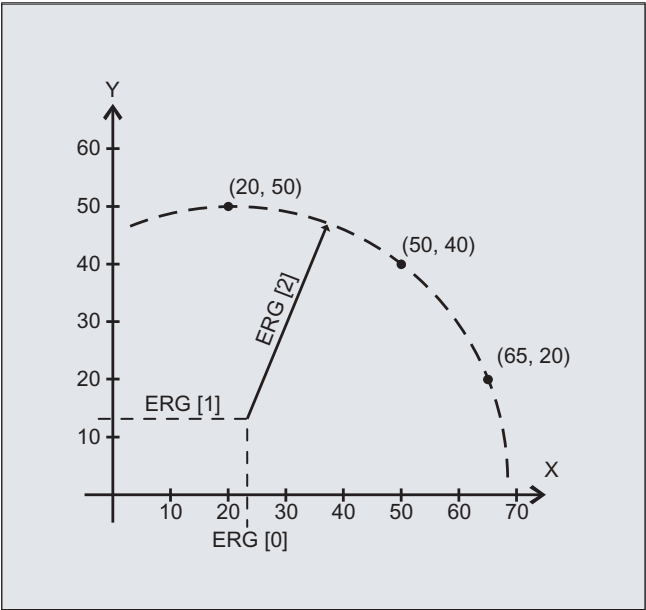
CALCDAT:	Předdefinovaná funkce pro výpočet rádiusu a souřadnic středu kružnice na základě 3 nebo 4 bodů.		
<Status>:	Proměnná pro stav výpočtu kružnice		
	Typ:	BOOL	
	Hodnota:	TRUE	Zadané body leží na kružnici.
		FALSE	Zadané body neleží na kružnici.
<body kružnice>[]:	Proměnná pro zadání bodů na kružnici s následujícími parametry:		
	<počet>:	Počet bodů na kružnici (3 nebo 4)	
	<druh>:	Druh zadání souřadnic, např. 2 pro souřadnice 2 bodů	
<počet>:	Parametr pro zadání počtu bodů použitých pro výpočet (3 nebo 4)		
<výsledek>[3]:	Proměnná pro výsledek		
	Zadání souřadnic středu kružnice a rádiusu		
	0	Souřadnice středu kružnice: hodnota abscisy	
	1	Souřadnice středu kružnice: hodnota ordináty	
	2	Rádus	

Poznámka

Mějte prosím na paměti, že proměnné musí být před svým použitím definovány.

Příklad:

U tří bodů se má zjistit, zda leží na kruhovém oblouku.



Programový kód	Komentář
N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)	; proměnná pro zadání bodů na kružnici
N20 DEF REAL ERG[3]	; proměnná pro výsledek
N30 DEF BOOL STATUS	; proměnná pro status
N40 STATUS=CALCDAT(PKT,3,ERG)	; vyvolání zjištěných parametrů kružnice
N50 IF STATUS == FALSE GOTOF ERROR	; skok v případě chyby

19.7 Ukončení přípravy kontury (EXECUTE)

Pomocí příkazu `EXECUTE` se příprava kontury ukončí a současně se přepne zpátky do normálního režimu zpracovávání programu.

Syntaxe

`EXECUTE (<FEHLER>)`

Význam

EXECUTE:	Předdefinovaná procedura pro ukončení přípravy kontury	
<FEHLER>:	Proměnná pro zpětné chybové hlášení	
	Typ:	INT
	Hodnota proměnné ukazuje, zda bylo možné zpracovat konturu bezchybně.	
	0	Chyba
	1	Žádná chyba

Příklad:

Programový kód

```
...  
N30 CONTPRON (...)  
N40 G1 X... Z...  
...  
N100 EXECUTE (...)  
...
```


Programování externích cyklů

20.1 Technologické cykly

20.1.1 Úvod

Obsah

Tyto kapitola obsahuje popis cyklů pro technologie soustružení, frézování a broušení.

Struktura

Popis cyklu má následující strukturu:

- **Syntaxe**
Název cyklu a posloupnost vyvolávání předávaných parametrů
- **Parametry**
Tabulka s vysvětleními jednotlivých parametrů

Popis parametrů

V tabulce jsou k jednotlivým parametrům uváděny následující údaje: Název, popis, rozsah hodnot a souvislosti s jinými parametry.

Sloupec s odkazem na parametr na obrazovce slouží pro usnadnění opětovného hledání hodnot naprogramovaných na řídicím systému při zpětném překladu externě generovaného volání cyklu.

Parametry "jen pro pracovní plochu"

V tabulce jsou některé parametry označeny příznakem "jen pro pracovní plochu". Tyto parametry nemají pro fungování cyklu žádný význam. Jsou zapotřebí jen k tomu, aby volání cyklu bylo možné dokonale zpětně přeložit. Pokud nejsou naprogramovány, může být cyklus i přesto zpětně přeložen, pole jsou ale potom odpovídajícím způsobem barevně označena a musí být vyplněna na obrazovce.

Parametry "rezervováno"

Pro parametry, které jsou popsány jako "rezervované", musí být naprogramována hodnota 0 nebo musí zůstat prázdné, aby přiřazení následujících parametrů volání odpovídalo interním parametrům cyklu. Výjimka: U parametrů typu STRING hodnota "" nebo prázdný znak.

Opakování cyklu na polohovacím vzoru

Vrtací a frézovací cykly mohou být opakovány na polohovacím vzoru (modální volání). Před cyklem je potom na tomtéž řádku zapotřebí napsat příkaz `MCALL`, např. `MCALL CYCLE83 ( . . . )`.

Poznámka

Pokud jsou určité předávané parametry (např. `<_VARI>`, `<_GMODE>`, `<_DMODE>`, `<_AMODE>`) naprogramovány nepřímo jako parametry, při zpětném překladu se otevře vstupní obrazovka, tu však není možné uložit, protože pro některá pole s volitelnou hodnotou neexistuje žádné jednoznačné přiřazení.

20.1.2 Přehled pro jednotlivé technologie

V následující tabulce se nachází přehled všech externě naprogramovatelných technologických cyklů, které jsou k dispozici a které jsou přiřazeny jednotlivým technologiím:

Technologie	Technologický cyklus
Vrtání	• CYCLE81 - Vrtání, navrtávání středících důlků (Strana 759) • CYCLE82 - Vrtání, čelní zahlubování (Strana 760) • CYCLE85 - Vystružování (Strana 768) • CYCLE86 - Vyvrtávání (Strana 769) • CYCLE83 - Vrtání hlubokých děr (Strana 762) • CYCLE830 - Vrtání hlubokých děr 2 (Strana 793) • CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky (Strana 765) • CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou (Strana 802) • CYCLE78 - Frézování vrtaných závitů (Strana 754) • CYCLE802 - Libovolné pozice (Strana 790) • HOLES1 - Řada děr (Strana 720) • CYCLE801 - Mřížka nebo obdélník (Strana 789) • HOLES2 - Díry uspořádané na kruhovém oblouku (Strana 720)
Soustružení	• CYCLE951 - Oddělování třísky (Strana 813) • CYCLE930 - Zápich (Strana 808) • CYCLE940 - Odlehčovací zápich určitého tvaru (Strana 810) • CYCLE99 - Soustružení závitu (Strana 778) • CYCLE98 - Řetězec závitů (Strana 774) • CYCLE92 - Upichování (Strana 771)
Soustružení kontur	• CYCLE62 - volání kontury (Strana 739) • CYCLE952 - Zápichy na kontuře (Strana 816)

Technologie	Technologický cyklus
Frézování	• CYCLE61 - Rovinné frézování (Strana 737) • POCKET3 - Frézování pravoúhlé kapsy (Strana 722) • POCKET4 - Frézování kruhové kapsy (Strana 725) • CYCLE76 - Frézování pravoúhlého čepu (Strana 750) • CYCLE77 - Frézování kruhového čepu (Strana 752) • CYCLE79 - Mnohohran (Strana 757) • SLOT1 - Podélná drážka (Strana 728) • SLOT2 - Kruhová drážka (Strana 730) • CYCLE899 - Frézování otevřené drážky (Strana 805) • LONGHOLE - Podlouhlá díra (Strana 733) • CYCLE70 - Frézování závitu (Strana 744) • CYCLE60 - Cyklus pro gravírování (Strana 735)
Frézování kontury	• CYCLE62 - volání kontury (Strana 739) • CYCLE72 - Frézování po dráze (Strana 745) • CYCLE63 - Frézování konturové kapsy (Strana 740) • CYCLE64 - Předvrtání konturové kapsy (Strana 742)
Broušení	• CYCLE495 - Profilování (Strana 783) • CYCLE435 - Nastavení souřadného systému orovnávače (Strana 783) • CYCLE4071 - Podélné broušení s přísuvem v bodu obratu (Strana 822) • CYCLE4072 - Podélné broušení s přísuvem v bodě obratu a se signálem pro přerušení (Strana 823) • CYCLE4073 - Podélné broušení se spojitým přísuvem (Strana 827) • CYCLE4074 - Podélné broušení se spojitým přísuvem a se signálem pro přerušení (Strana 828) • CYCLE4075 - Plošné broušení s přísuvem v bodě obratu (Strana 831) • CYCLE4077 - Plošné broušení s přísuvem na bod obratu a se signálem pro přerušení (Strana 834) • CYCLE4078 - Plošné broušení se spojitým přísuvem (Strana 837) • CYCLE4079 - Plošné broušení s přerušovaným přísuvem (Strana 839)
Ostatní	• CYCLE800 - Naklápění (Strana 785) • CYCLE832 - Vysokorychlostní obrábění (Strana 799)
Všechny	• GROUP_BEGIN - Začátek programového bloku (Strana 841) • GROUP_END - konec programového bloku (Strana 842) • GROUP_ADDEND - konec doplňkového bloku najíždění (Strana 842)

20.1.3 HOLES1 - Řada děr

Syntaxe

HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>, <_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	X0	<SPCA>	REAL	Vztažný bod na 1. ose pro řadu děr (abs)
2	Y0	<SPCO>	REAL	Vztažný bod na 2. ose pro řadu děr (abs)
3	α 0	<STA1>	REAL	Základní úhel otočení (úhel vzhledem k 1. ose)
4	L0	<FDIS>	REAL	Vzdálenost 1. díry od vztažného bodu
5	L	<DBH>	REAL	Vzdálenost mezi dírami
6	N	<NUM>	INT	Počet děr
7		<_VARI>	INT	rezervováno
8		<_UMODE>	INT	rezervováno
9		<_HIDE>	STRING [200]	Přeskakované pozice - max. 198 znaků - Údaj pořadového čísla pozice, např. "1,3" (pozice 1 a 3 se nebudou uskutečňovat)
10		<_NSP>	INT	rezervováno
11		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)

20.1.4 HOLES2 - Díry uspořádané na kruhovém oblouku

Syntaxe

HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>, <_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)

Parametr

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam	
1	X0	<CPA>	REAL	Střed kruhu na 1. ose pro díry na kruhovém oblouku (abs) Souřadnice vztažného bodu na 1. ose	(v případě XY) (v případě XA, YB, ZC)
2	Y0	<CPO>	REAL	Střed kruhu na 2. ose pro díry na kruhovém oblouku (abs) Souřadnice vztažného bodu na 2. ose	(v případě XY) (v případě XA, YB, ZC)
3	R	<RAD>	REAL	Rádus kruhového oblouku	(v případě XY)
4	$\alpha 0$	<STA1>	REAL	Počáteční úhel nebo 1. poloha kruhové osy	(v případě XY) (v případě XA, YB, ZC)
5	$\alpha 1$	<INDA>	REAL	Úhlový krok, kterým jsou drážky od sebe vzdáleny (jen u kruhového oblouku)	(v případě XY, XA, YB, ZC)
				< 0 =	Ve směru hodinových ručiček
				> 0 =	Proti směru hodinových ručiček
6	N	<NUM>	INT	Počet pozic	
7		<_VARI>	INT	Způsob opracování	
				JEDNOTKY:	rezervováno
				DESÍTKY:	Způsob najíždění na pozici
				0 =	Najíždění na pozici po lineární dráze
				1 =	Najíždění na pozici po kruhové dráze
				STOVKY:	rezervováno
				TISÍCE:	Kruhový vzor
				0 =	Režim kompatibility, pokud je INDA = 0, pak celá kružnice, INDA <> 0, potom kruhový oblouk
				1 =	Kružnice
				2 =	Kruhový oblouk
				DESÍTKY TISÍC:	Polohovací vzor s kruhovou osou
				0 =	XY (bez kruhové osy) (v případě XY)
				1 =	XA (osa X a kruhová osa oko osy X) (jen v případě XA)
				2 =	YB (osa Y a kruhová osa oko osy Y) (jen v případě YB)
				3 =	ZC (osa Z a kruhová osa oko osy C) (jen v případě ZC)
				JEDNOTKY MILIONŮ + STOVKY TISÍC	Offset (pokud se více kruhových otáčí okolo stejné osy; pokud je index příliš veliký, potom 1. osa)
				00 =	1. osa A, B nebo C
				01 =	2. osa A, B nebo C
				...	
				10 =	20. osa A, B nebo C
8		<_UMODE>	INT	rezervováno	

Č.	Obrazovka parametru	Parametry interní	Datový typ	Význam
9		<_HIDE>	STRING [200]	rezervováno
10		<_NSP>	INT	rezervováno
11		<_DMODE>	INT	Režim zobrazování
			JEDNOTKY:	Rovina obrábění G17/G18/G19
			0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
			1 =	G17 (aktivní jen v cyklu)
			2 =	G18 (aktivní jen v cyklu)
			3 =	G19 (aktivní jen v cyklu)

20.1.5 POCKET3 - Frézování pravoúhlé kapsy

Syntaxe

POCKET3(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_CRAD>, <_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_MIDA>, <_AP1>, <_AP2>, <_AD>, <_RAD1>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametru	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka kapsy (abs/ink), viz <_AMODE>)
5	L	<_LENG>	REAL	Délka kapsy (ink, je potřeba zadávat se znaménkem)
6	W	<_WID>	REAL	Šířka kapsy (ink, je potřeba zadávat se znaménkem)
7	R	<_CRAD>	REAL	Rohový rádius kapsy
8	X0	<_PA>	REAL	Vztažný bod, 1. osa (abs)
9	YO	<_PO>	REAL	Vztažný bod, 2. osa (abs)
10	α0	<_STA>	REAL	Úhel otočení, úhel mezi podélnou osou (L) a 1. osou
11	DZ	<_MID>	REAL	Maximální přísuv do hloubky
12	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině
13	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně
14	F	<_FFP1>	REAL	Posuv v rovině
15	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky
16		<_CDIR>	INT	Směr frézování:
			0 =	Sousledné frézování
			1 =	Nesousledné frézování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
17		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				4 = Obrábění okraje načisto
				5 = Srážení hran
				DESÍTKY:
				0 = předvrtáno, přísuv s G0
				1 = kolmo, přísuv s G1
				2 = po šroubovici
				3 = oscilačním (kyvným) pohybem po podélné ose kapsy
				STOVKY: rezervováno
18	DXY	<_MIDA>	REAL	Maximální přísuv v rovině, jednotky viz <_AMODE>
19	L1	<_AP1>	REAL	Délka částečně obrobené kapsy (ink)
20	W1	<_AP2>	REAL	Šířka částečně obrobené kapsy (ink)
21	AZ	<_AD>	REAL	Hloubka částečně obrobené kapsy (ink)
22	ER	<_RAD1>	REAL	Rádus spirální dráhy při zajiždění po šroubovici
	EW			Maximální úhel zajiždění pro oscilační pohyb
23	EP	<_DP1>	REAL	Stoupání spirály při zajiždění po spirální dráze
24		<_UMODE>	INT	rezervováno
25	FS	<_FS>	REAL	Šířka fasety (ink)
26	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
27		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
				STOVKY: Volba opracování/pouze výpočet počátečního bodu
				0 = Režim kompatibility
				1 = normální opracování
				TISÍCE: Zadávání rozměrů pomocí středu/rohu
				0 = Režim kompatibility
				1 = Zadávání rozměrů pomocí středu
				2 = Kótování pomocí rohového bodu, poloha kapsy +LENG/+WID
				3 = Kótování pomocí rohového bodu, poloha kapsy -LENG/+WID
				4 = Kótování pomocí rohového bodu, poloha kapsy +LENG/-WID
				5 = Kótování pomocí rohového bodu, poloha kapsy -LENG/-WID
				DESÍTKY TISÍC: Kompletní opracování/dopracování
				0 = Režim kompatibility (s parametry <_AP1>, <_AP2> a <_AD> se zachází jako předtím)
				1 = Kompletní opracování
				2 = Dodatečné opracování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
28		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY:
				Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky
				0 = Režim kompatibility
29		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Hloubka kapsy (Z1)
				0 = absolutní (režim kompatibility)
				1 = inkrementální
				DESÍTKY:
				Jednotky pro příusv v rovině (DXY)
				0 = mm
				1 = procentuální hodnota z průměru nástroje
				STOVKY:
				Hloubka zajištění při srážení hran (ZFS)
				0 = absolutní
				1 = inkrementální

20.1.6 POCKET4 - Frézování kruhové kapsy

Syntaxe

```
POCKET4(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_CDIAM>, <_PA>, <_PO>,
<_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_MIDA>,
<_AP1>, <_AD>, <_RAD1>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka kapsy (abs/ink), viz <_AMODE>		
5	Ø	<_CDIAM>	REAL	Průměr kapsy nebo rádius kapsy, viz <_DMODE>		
6	X0	<_PA>	REAL	Vztažný bod, 1. osa (abs)		
7	Y0	<_PO>	REAL	Vztažný bod, 2. osa (abs)		
8	DZ	<_MID>	REAL	maximální přísuv do hloubky, viz <_VARI> = po rovinách maximální stoupání spirály, viz <_VARI> = po šroubovici		
9	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině		
10	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
11	F	<_FFP1>	REAL	Posuv pro opracovávání plochy		
12	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky		
13		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
14		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Opracování	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					4 =	Obrábění okraje načisto
					5 =	Srážení hran
				DESÍTKY:	Způsob přísuvu (obrábění nahrubo a obrábění načisto)	
					0 =	předvrtáno, přísuv s G0 (kapsa je předem opracována)
					1 =	kolmo, přísuv s G1
					2 =	po šroubovici
				STOVKY:	rezervováno	
				TISÍCE:		
					0 =	po rovinách
					1 =	po šroubovici
15	DXY	<_MIDA>	REAL	Maximální přísuv v rovině, viz <_AMODE>, 0 = 0,8 x průměr nástroje		
16	Ø	<_AP1>	REAL	Průměr/rádius částečně obrobené kapsy (ink)		
17	AZ	<_AD>	REAL	Hloubka částečně obrobené kapsy (ink)		
18	ER	<_RAD1>	REAL	Rádius spirální dráhy při zajiždění po šroubovici		
19	EP	<_DP1>	REAL	Stoupání spirály při zajiždění po spirální dráze		
20		<_UMODE>	INT	rezervováno		
21	FS	<_FS>	REAL	Šířka fasety (ink)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
22	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>
23		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty) JEDNOTKY: rezervováno DESÍTKY: rezervováno STOVKY: Opracování/výpočet počátečního bodu 0 = Režim kompatibility 1 = normální opracování TISÍCE: rezervováno DESÍTKY TISÍC: Kompletní opracování/dopracování 0 = Režim kompatibility (s parametry <_AP1> a <_AD> se zachází jako předtím) 1 = Kompletní opracování 2 = Dodatečné opracování
24		<_DMODE>	INT	Režim zobrazování JEDNOTKY: Rovina obrábění G17/G18/G19 0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu 1 = G17 (aktivní jen v cyklu) 2 = G18 (aktivní jen v cyklu) 3 = G19 (aktivní jen v cyklu) DESÍTKY: Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky 0 = Režim kompatibility 1 = Příkaz G-funkce stejný jako před voláním cyklu G94/G95 pro posuv v rovině a posuv do hloubky zůstávají stejné. STOVKY: 0 = Režim kompatibility (<_CDIAM>/<_AP1> zadat jako rádius) 1 = <_CDIAM>/<_AP1> se zadávají jako průměr TISÍCE: --- rezervováno DESÍTKY TISÍC: Odstupňování technologie na obrazovkách cyklů (Strana 842) 0 = Zadání: kompletní 1 = Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
25		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Hloubka kapsy (Z1)
				0 = absolutní (režim kompatibility)
				1 = inkrementální
				DESÍTKY:
				Jednotky pro přísuv do šířky (DXY)
				0 = mm
				1 = procentuální hodnota z průměru nástroje
				STOVKY:
				Hloubka zajiždění při srážení hran (ZFS)
				0 = absolutní
				1 = inkrementální

20.1.7 SLOT1 - Podélná drážka

Syntaxe

SLOT1 (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <WID>, <_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>, <CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FALD>, <_STA2>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka drážky (abs)
5		<_DPR>	REAL	Hloubka drážky (ink), vztažená na Z0 (zadává se bez znaménka)
6		<NUM>	INT	Počet drážek = 1
7	L	<LENG>	REAL	Délka drážky
8	W	<WID>	REAL	Šířka drážky
9	X0	<_CPA>	REAL	Vztažný bod, 1. osa roviny
10	Y0	<_CPO>	REAL	Vztažný bod, 2. osa roviny
11		<RAD>	REAL	rezervováno
12	α	<STA1>	REAL	Úhel otočení
13		<INDA>	REAL	rezervováno
14	FZ	<FFD>	REAL	Rychlost přísuvu do hloubky
15	F	<FFP1>	REAL	Posuv
16	DZ	<_MID>	REAL	Maximální přísuv do hloubky

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
17		<CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
18	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině nebo na okrajích drážky		
19		<VARI>	INT	Způsob opracování		
				JEDNOTKY:		
					0 =	rezervováno
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					4 =	Obrábění načisto okraje (opracování jen okraje)
					5 =	Srážení hran
				DESÍTKY:	Najíždění	
					0 =	předvrtáno, přísuv s G0 (drážka je předem opracována)
					1 =	kolmo, přísuv s G1
					2 =	po šroubovici
					3 =	oscilačním pohybem
				STOVKY:	rezervováno	
20	DZF	<_MIDF>	REAL	rezervováno		
21	FF	<FFP2>	REAL	rezervováno		
22	SF	<SSF>	REAL	rezervováno		
23	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
24	ER	<_STA2>	REAL	Rádus spirální dráhy při zajiždění po šroubovici		
	EW			Maximální úhel zajiždění pro oscilační pohyb		
25	EP	<_DP1>	REAL	Hloubka zajiždění na jednu otáčku pro spirální pohyb		
26		<_UMODE>	INT	rezervováno		
27	FS	<_FS>	REAL	Šířka fasety (ink) u srážení hran		
28	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		
29		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování nebo pouze výpočet počátečního bodu	
					1 =	normální opracování
				TISÍCE:	Kótování vztažného bodu, poloha drážky	
					0 =	ve středu
					1 =	vlevo uvnitř +L
					2 =	vpravo uvnitř -L
					3 =	levý okraj +L
					4 =	pravý okraj -L

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
30		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: rezervováno
				STOVKY: rezervováno
				TISÍCE: Identifikace verze SW
				1 = Rozšířené funkce SLOT1
				DESÍTKY TISÍC: Odstupňování technologie na obrazovkách cyklů (Strana 842)
31		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Konečná hloubka Z1 (abs/ink)
				0 = Kompatibilita
				1 = Z1 (ink)
				2 = Z1 (abs)
				DESÍTKY: rezervováno
				STOVKY: Hloubka zajiždění při srážení hran ZFS
				0 = ZFS (abs)
				1 = ZFS (ink)

Poznámka

Cyklus je oproti předcházejícím verzím SW vybaven novými funkcemi. To má za následek, že určité parametry se na vstupní obrazovce už nevypisují (<NUM>, <RAD>, <INDA>). Větší počet drážek na polohovacím vzoru je možné naprogramovat pomocí příkazu "MCALL" a voláním požadovaného polohovacího vzoru, např. HOLES2.

20.1.8 SLOT2 - Kruhová drážka

Syntaxe

SLOT2(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <AFSL>, <WID>, <_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>, <CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FFCP>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka drážky (abs)
5		<_DPR>	REAL	Hloubka drážky (ink), vztažená na Z0 (zadává se bez znaménka)
6	N	<NUM>	INT	Počet drážek
7	$\alpha 1$	<AFSL>	REAL	Úhel kruhové výseče dané drážky
8	W	<WID>	REAL	Šířka drážky
9	X0	<_CPA>	REAL	Vztažný bod = střed kruhového oblouku, 1. osa roviny
10	Y0	<_CPO>	REAL	Vztažný bod = střed kruhového oblouku, 2. osa roviny
11	R	<RAD>	REAL	Rádus kruhového oblouku
12	$\alpha 0$	<STA1>	REAL	Počáteční úhel
13	$\alpha 2$	<INDA>	REAL	Úhlový krok
14	FZ	<FFD>	REAL	Rychlost přísuvu do hloubky
15	F	<FFP1>	REAL	Posuv
16	DZ	<_MID>	REAL	Maximální přísuv do hloubky
17		<CDIR>	INT	Směr frézování
				0 = Sousedné frézování
				1 = Nesousedné frézování
18	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině nebo na okrajích drážky
19		<VARI>	INT	Způsob opracování
				JEDNOTKY:
				0 = Kompletní opracování
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				3 = Obrábění okraje načisto
				5 = Srážení hran
				DESÍTKY:
				0 = najíždění na pomocné polohy po přímkách s G0
				1 = najíždění na pomocné polohy po kruhové dráze
				STOVKY: rezervováno
				TISÍCE:
				0 = Režim kompatibility, pokud je <INDA> = 0, pak celá kružnice, <INDA> <> 0, potom kruhový oblouk
				1 = Kružnice
				2 = Kruhový oblouk
20	DZF	<_MIDF>	REAL	rezervováno

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
21		<FFP2>	REAL	rezervováno
22		<SSF>	REAL	rezervováno
23	FF	<_FFCP>	REAL	rezervováno
24		<_UMODE>	INT	rezervováno
25	FS	<_FS>	REAL	Šířka fasety (ink)
26	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>
27		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
			JEDNOTKY:	rezervováno
			DESÍTKY:	rezervováno
			STOVKY:	Volba opracování nebo pouze výpočet počátečního bodu
			0 =	Režim kompatibility
			1 =	normální opracování
28		<_DMODE>	INT	Režim zobrazování
			JEDNOTKY:	Rovina obrábění G17/G18/G19
			0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
			1 =	G17 (aktivní jen v cyklu)
			2 =	G18 (aktivní jen v cyklu)
			3 =	G19 (aktivní jen v cyklu)
			DESÍTKY:	rezervováno
			STOVKY:	rezervováno
			TISÍCE:	Identifikace verze SW
			1 =	funkce SLOT2 od SW verze 2.5
			DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 842)
			0 =	Zadání: kompletní
			1 =	Zadání: jednoduché
29		<_AMODE>	INT	Alternativní režim
			JEDNOTKY:	Konečná hloubka Z1 (abs/ink)
			0 =	Kompatibilita
			1 =	Z1 (ink)
			2 =	Z1 (abs)
			DESÍTKY:	rezervováno
			STOVKY:	Hloubka zajiždění při srážení hran ZFS
			0 =	ZFS (abs)
			1 =	ZFS (ink)

20.1.9 LONGHOLE - Podlouhlá díra

Syntaxe

```
LONGHOLE (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>,
<_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>,
<_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka podlouhlé díry (abs)
5		<_DPR>	REAL	Hloubka podlouhlé díry (ink), vztažená na Z0 (zadává se bez znaménka)
6		<NUM>	INT	Počet podlouhlých děr = 1
7	L	<LENG>	REAL	Délka podlouhlé díry
8	X0	<_CPA>	REAL	Vztažný bod, 1. osy v rovině
9	Y0	<_CPO>	REAL	Vztažný bod, 2. osy v rovině
10		<RAD>	REAL	rezervováno
11	$\alpha 0$	<STA1>	REAL	Úhel otočení
12		<INDA>	REAL	rezervováno
13	FZ	<FFD>	REAL	Rychlost přísuvu do hloubky
14	F	<FFP1>	REAL	Posuv
15	DZ	<MID>	REAL	Maximální přísuv do hloubky
16		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				Druh přísuvu
				1 = kolmo s G1
				3 = oscilačním pohybem
				STOVKY:
				rezervováno
17		<_UMODE>	INT	rezervováno

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
18		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
				STOVKY: Volba opracování nebo pouze výpočet počátečního bodu
				0 = Režim kompatibility
				1 = normální opracování
				TISÍCE: Kótování vztažného bodu, poloha drážky
				0 = ve středu
				1 = vlevo uvnitř +L
				2 = vpravo uvnitř -L
				3 = levý okraj +L
				4 = pravý okraj -L
19		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky
				0 = Režim kompatibility
				1 = Příkaz G-funkce stejný jako před voláním cyklu G94/G95 pro posuv v rovině a posuv do hloubky zůstávají stejné.
				STOVKY: rezervováno
20		<_AMODE>	INT	Alternativní režim
				JEDNOTKY: Konečná hloubka Z1 (abs/ink)
				0 = Kompatibilita
				1 = Z1 (ink)
				2 = Z1 (abs)

Poznámka

Cyklus je oproti předcházejícím verzím SW vybaven novými funkcemi. To má za následek, že určité parametry se na vstupní obrazovce už nevypisují (<NUM>, <RAD>, <INDA>). Větší počet drážek na polohovacím vzoru je možné naprogramovat pomocí příkazu "MCALL" a voláním požadovaného polohovacího vzoru, např. HOLES2.

20.1.10 CYCLE60 - Cyklus pro gravírování**Syntaxe**

```
CYCLE60 (<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_TEXT>	STRING [200]	Text, který má být vygravírován (maximálně 100 znaků)
2	RP	<_RTP>	REAL	Návratová rovina (abs)
3	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
4	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se k referenční rovině, zadává se bez znaménka)
5	Z1	<_DP>	REAL	Hloubka (abs), viz <_AMODE>
6	Z1	<_DPR>	REAL	Hloubka (ink), viz <_AMODE>
7	X0	<_PA>	REAL	Vztažný bod, 1. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	R			Vztažný bod, délka (rádius) - polární souřadnice, viz <_VARI>
8	Y0	<_PO>	REAL	Vztažný bod, 2. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	α0			Vztažný bod, úhel vztažený na 1. osu - polární souřadnice, viz <_VARI>
9	α1	<_STA>	REAL	Směr textu, úhel textové čáry vztažený na 1. osu, viz <_VARI>
10	XM	<_CP1>	REAL	Střed textového kruhového oblouku, 1. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	LM			Střed textového kruhového oblouku, délka (rádius) vztažená na počátek WCS - polární souřadnice, viz <_VARI>
11	YM	<_CP2>	REAL	Střed textového kruhového oblouku, 2. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	αM			Střed textového kruhového oblouku, úhel vztažený na 1. osu - polární souřadnice, viz <_VARI>
12	W	<_WID>	REAL	Výška znaků (zadává se bez znaménka)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
13	DX1 DX2	<_DF>	REAL	Vzdálenost mezi znaky / celková šířka, viz <_VARI>
	α2			Úhel kruhové výseče, viz <_VARI>
14	FZ	<_FFD>	REAL	Posuv při přísuvu do hloubky, viz <_DMODE>
15	F	<_FFP1>	REAL	Posuv pro opracovávání plochy
16		<_VARI>	INT	Opracování (vyrovnání a vztažný bod gravírovaného textu)
				JEDNOTKY:
				Vztažný bod
				0 = pravoúhlý
				1 = polární
				DESÍTKY:
				Vyrovnání textu
				0 = Text na přímce
				1 = Text shora na kruhovém oblouku
				2 = Text zdola na kruhovém oblouku
				STOVKY:
				rezervováno
				TISÍCE:
				Vztažný bod textu ve vodorovném směru
				0 = vlevo
				1 = na střed
				2 = vpravo
				DESÍTKY TISÍC:
				Vztažný bod textu ve svislém směru
				0 = dole
				1 = na střed
				2 = nahoře
				STOVKY TISÍC:
				Délka textu
				0 = Vzdálenost mezi znaky
				1 = Celková šířka textu (jen u textu na přímce)
				2 = Úhel kruhové výseče (jen u textu na kruhovém oblouku)
				JEDNOTKY MILIONŮ:
				Střed kruhu
				0 = pravoúhlé (kartézské) souřadnice
				1 = polární
				DESÍTKY MILIONŮ:
				Zrcadlově převrácené písmo
				0 = Kompatibilita
				1 = Zrcadlově převrácené písmo aktivováno
				2 = Zrcadlově převrácené písmo deaktivováno
17		<_CODEP>	INT	Číslo kódové stránky pro písmo (v současnosti pouze 1252)
18		<_UMODE>	INT	rezervováno

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
19		_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
				STOVKY: Volba zpracování/pouze výpočet počátečního bodu
				0 = Režim kompatibility
				1 = normální zpracování
20		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky
21		<_AMODE>	INT	Alternativní režim
				JEDNOTKY: Konečná hloubka (<_DP>, <_DPR>)
				0 = Kompatibilita
				1 = inkrementálně (<_DPR>)
				2 = absolutně (<_DP>)

20.1.11 CYCLE61 - Rovinné frézování

Syntaxe

```
CYCLE61 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje, výška surového obrobku (abs)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam																		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)																		
4	Z1	<_DP>	REAL	Výška hotového obrobku (abs/ink), viz <_AMODE>																		
5	X0	<_PA>	REAL	Rohový bod 1 na 1. ose (abs)																		
6	Y0	<_PO>	REAL	Rohový bod 1 na 2. ose (abs)																		
7	X1	<_LENG>	REAL	Rohový bod 2 na 1. ose (abs/ink), viz <_AMODE>																		
8	Y1	<_WID>	REAL	Rohový bod 2 na 2. ose (abs/ink), viz <_AMODE>																		
9	DZ	<_MID>	REAL	Maximální přísuv do hloubky																		
10	DXY	<_MIDA>	REAL	Maximální přísuv v rovině (jednotky viz <_AMODE>)																		
11	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně																		
12	F	<_FFP1>	REAL	Pracovní posuv																		
13		<_VARI>	INT	<table><tr><td colspan="2">Způsob opracování</td></tr><tr><td rowspan="3">JEDNOTKY:</td><td>Opracování</td></tr><tr><td>1 = Obrábění nahrubo</td></tr><tr><td>2 = Obrábění načisto</td></tr><tr><td rowspan="5">DESÍTKY:</td><td>Směr obrábění</td></tr><tr><td>1 = rovnoběžně s 1. osou, jeden směr</td></tr><tr><td>2 = rovnoběžně s 2. osou, jeden směr</td></tr><tr><td>3 = rovnoběžně s 1. osou, střídající se směr</td></tr><tr><td>4 = rovnoběžně s 2. osou, střídající se směr</td></tr></table>	Způsob opracování		JEDNOTKY:	Opracování	1 = Obrábění nahrubo	2 = Obrábění načisto	DESÍTKY:	Směr obrábění	1 = rovnoběžně s 1. osou, jeden směr	2 = rovnoběžně s 2. osou, jeden směr	3 = rovnoběžně s 1. osou, střídající se směr	4 = rovnoběžně s 2. osou, střídající se směr						
Způsob opracování																						
JEDNOTKY:	Opracování																					
	1 = Obrábění nahrubo																					
	2 = Obrábění načisto																					
DESÍTKY:	Směr obrábění																					
	1 = rovnoběžně s 1. osou, jeden směr																					
	2 = rovnoběžně s 2. osou, jeden směr																					
	3 = rovnoběžně s 1. osou, střídající se směr																					
	4 = rovnoběžně s 2. osou, střídající se směr																					
14		<_LIM>	INT	<table><tr><td colspan="2">Ohraničení</td></tr><tr><td rowspan="3">JEDNOTKY:</td><td>Ohraničení pro 1. osu ve směru mínus</td></tr><tr><td>0 = ne</td></tr><tr><td>1 = ano</td></tr><tr><td rowspan="3">DESÍTKY:</td><td>Ohraničení pro 1. osu ve směru plus</td></tr><tr><td>0 = ne</td></tr><tr><td>1 = ano</td></tr><tr><td rowspan="3">STOVKY:</td><td>Ohraničení pro 2. osu ve směru mínus</td></tr><tr><td>0 = ne</td></tr><tr><td>1 = ano</td></tr><tr><td rowspan="3">TISÍCE:</td><td>Ohraničení pro 2. osu ve směru plus</td></tr><tr><td>0 = ne</td></tr><tr><td>1 = ano</td></tr></table>	Ohraničení		JEDNOTKY:	Ohraničení pro 1. osu ve směru mínus	0 = ne	1 = ano	DESÍTKY:	Ohraničení pro 1. osu ve směru plus	0 = ne	1 = ano	STOVKY:	Ohraničení pro 2. osu ve směru mínus	0 = ne	1 = ano	TISÍCE:	Ohraničení pro 2. osu ve směru plus	0 = ne	1 = ano
Ohraničení																						
JEDNOTKY:	Ohraničení pro 1. osu ve směru mínus																					
	0 = ne																					
	1 = ano																					
DESÍTKY:	Ohraničení pro 1. osu ve směru plus																					
	0 = ne																					
	1 = ano																					
STOVKY:	Ohraničení pro 2. osu ve směru mínus																					
	0 = ne																					
	1 = ano																					
TISÍCE:	Ohraničení pro 2. osu ve směru plus																					
	0 = ne																					
	1 = ano																					
15		<_DMODE>	INT	<table><tr><td colspan="2">Režim zobrazování</td></tr><tr><td rowspan="5">JEDNOTKY:</td><td>Rovina obrábění G17/G18/G19</td></tr><tr><td>0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu</td></tr><tr><td>1 = G17 (aktivní jen v cyklu)</td></tr><tr><td>2 = G18 (aktivní jen v cyklu)</td></tr><tr><td>3 = G19 (aktivní jen v cyklu)</td></tr></table>	Režim zobrazování		JEDNOTKY:	Rovina obrábění G17/G18/G19	0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu	1 = G17 (aktivní jen v cyklu)	2 = G18 (aktivní jen v cyklu)	3 = G19 (aktivní jen v cyklu)										
Režim zobrazování																						
JEDNOTKY:	Rovina obrábění G17/G18/G19																					
	0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu																					
	1 = G17 (aktivní jen v cyklu)																					
	2 = G18 (aktivní jen v cyklu)																					
	3 = G19 (aktivní jen v cyklu)																					

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
16		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Konečná hloubka (<_DP>)
				0 = absolutní
				1 = inkrementální
				DESÍTKY:
				Jednotky pro přísuv v rovině (<_MIDA>)
				0 = mm
				1 = procentuální hodnota z průměru nástroje
				STOVKY:
				rezervováno
				TISÍCE:
				Délka plochy
				0 = inkrementální
				1 = absolutní
				DESÍTKY TISÍC:
				Šířka plochy
				0 = inkrementální
				1 = absolutní

20.1.12 CYCLE62 - volání kontury

Syntaxe

CYCLE62 (<_KNAME>, <_TYPE>, <_LAB1>, <_LAB2>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	PRG/CON	<_KNAME>	STRING [140]	Název kontury nebo podprogramu, nemusí být naprogramován, když je _TYPE = 2
2		<_TYPE>	INT	Stanovení zadání kontury
				0 = Podprogram
				1 = Název kontury
				2 = Návěští
				3 = Návěští v podprogramu
3	LAB1	<_LAB1>	STRING[32]	Návěští 1, začátek kontury
4	LAB2	<_LAB2>	STRING[32]	Návěští 2, konec kontury

20.1.13 CYCLE63 - Frézování konturové kapsy

Syntaxe

```
CYCLE63 (<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_FZ>,
<_DXY>, <_DZ>, <_UXY>, <_UZ>, <_CDIR>, <_XS>, <_YS>, <_ER>, <_EP>,
<_EW>, <_FS>, <_ZFS>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam																																									
1	PRG	<_PRG>	STRING [100]	Název programu pro odstraňování materiálu																																									
2		<_VARI>	INT	<table><tr><td colspan="3">Způsob opracování</td></tr><tr><td rowspan="5">JEDNOTKY:</td><td colspan="2">Technologie opracování</td></tr><tr><td>1 =</td><td>Obrábění nahrubo</td></tr><tr><td>3 =</td><td>Obrábění dna načisto</td></tr><tr><td>4 =</td><td>Obrábění okraje načisto</td></tr><tr><td>5 =</td><td>Srážení hran</td></tr><tr><td rowspan="4">DESÍTKY:</td><td colspan="2">Způsob přísuvu</td></tr><tr><td>0 =</td><td>Zajíždění nástroje středem</td></tr><tr><td>1 =</td><td>Zajíždění po šroubovici</td></tr><tr><td>2 =</td><td>Zajíždění oscilačním pohybem</td></tr><tr><td colspan="2">STOVKY:</td><td colspan="2">rezervováno</td></tr><tr><td rowspan="3">TISÍCE:</td><td colspan="2">Způsob pozvednutí nástroje</td></tr><tr><td>0 =</td><td>pozvednout na návratovou rovinu</td></tr><tr><td>1 =</td><td>pozvednout na vztažný bod + bezpečnostní vzdálenost</td></tr><tr><td rowspan="3">DESÍTKY TISÍC:</td><td colspan="2">Počáteční bod při obrábění nahrubo a načisto na dně</td></tr><tr><td>0 =</td><td>automaticky</td></tr><tr><td>1 =</td><td>manuálně</td></tr></table>	Způsob opracování			JEDNOTKY:	Technologie opracování		1 =	Obrábění nahrubo	3 =	Obrábění dna načisto	4 =	Obrábění okraje načisto	5 =	Srážení hran	DESÍTKY:	Způsob přísuvu		0 =	Zajíždění nástroje středem	1 =	Zajíždění po šroubovici	2 =	Zajíždění oscilačním pohybem	STOVKY:		rezervováno		TISÍCE:	Způsob pozvednutí nástroje		0 =	pozvednout na návratovou rovinu	1 =	pozvednout na vztažný bod + bezpečnostní vzdálenost	DESÍTKY TISÍC:	Počáteční bod při obrábění nahrubo a načisto na dně		0 =	automaticky	1 =	manuálně
Způsob opracování																																													
JEDNOTKY:	Technologie opracování																																												
	1 =	Obrábění nahrubo																																											
	3 =	Obrábění dna načisto																																											
	4 =	Obrábění okraje načisto																																											
	5 =	Srážení hran																																											
DESÍTKY:	Způsob přísuvu																																												
	0 =	Zajíždění nástroje středem																																											
	1 =	Zajíždění po šroubovici																																											
	2 =	Zajíždění oscilačním pohybem																																											
STOVKY:		rezervováno																																											
TISÍCE:	Způsob pozvednutí nástroje																																												
	0 =	pozvednout na návratovou rovinu																																											
	1 =	pozvednout na vztažný bod + bezpečnostní vzdálenost																																											
DESÍTKY TISÍC:	Počáteční bod při obrábění nahrubo a načisto na dně																																												
	0 =	automaticky																																											
	1 =	manuálně																																											
3	RP	<_RP>	REAL	Návratová rovina (abs)																																									
4	Z0	<_Z0>	REAL	Vztažný bod osy nástroje (abs)																																									
5	SC	<_SC>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)																																									
6	Z1	<_Z1>	REAL	Konečná hloubka, (viz JEDNOTKY parametru <_AMODE>)																																									
7	F	<_F>	REAL	Posuv v rovině při obrábění nahrubo/načisto																																									
8	FZ	<_FZ>	REAL	Rychlost přísuvu do hloubky																																									
9	DXY	<_DXY>	REAL	Přísuv v rovině - jednotky (viz DESÍTKY parametru <_AMODE>)																																									
10	DZ	<_DZ>	REAL	Přísuv do hloubky																																									
11	UXY	<_UXY>	REAL	Přídavek rozměru pro opracování načisto v rovině																																									
12	UZ	<_UZ>	REAL	Přídavek rozměru pro opracování načisto na dně																																									

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
13		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
14	XS	<_XS>	REAL	Souřadnice X počátečního bodu, absolutně		
15	YS	<_YS>	REAL	Souřadnice Y počátečního bodu, absolutně		
16	ER	<_ER>	REAL	Zajiždění po spirále: Rádus		
17	EP	<_EP>	REAL	Zajiždění po spirále: Stoupání		
18	EW	<_EW>	REAL	Zajiždění oscilačním pohybem: Maximální úhel při zajiždění		
19	FS	<_FS>	REAL	Šířka fasety (ink) u srážení hran		
20	ZFS	<_ZFS>	REAL	Hloubka zajiždění špičky nástroje při srážení hran (viz STOVKY parametru <_AMODE>)		
21	TR	<_TR>	STRING[32]	Název referenčního nástroje při odstraňování zbytkového materiálu		
22	DR	<_DR>	INT	D-číslo referenčního nástroje při odstraňování zbytkového materiálu		
23		<_UMODE>	INT	rezervováno		
24		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	normální opracování (není zapotřebí žádný režim kompatibility)
					1 =	normální opracování
					2 =	rezervováno
25		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	technologický režim	
					1 =	Kapsa
					2 =	Čep
				TISÍCE:	Odstraňování zbytkového materiálu	
					0 =	ne
					1 =	ano
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 842)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
26		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Koncová hloubka (Z1)
				0 = absolutní (režim kompatibility)
				1 = inkrementální
				DESÍTKY:
				Jednotky pro přísuv v rovině (DXY)
				0 = mm
				1 = procentuální hodnota z průměru nástroje
				STOVKY:
				Hloubka zajiždění při srážení hran (ZFS)
				0 = absolutní
				1 = inkrementální
				TISÍCE:
				--- rezervováno

20.1.14 CYCLE64 - Předvrtání konturové kapsy

Syntaxe

```
CYCLE64 (<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	PRG	<_PRG>	STRING [100]	Název programu pro vrtání/navrtávání středících důlků
2		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				rezervováno
				DESÍTKY:
				rezervováno
				STOVKY:
				rezervováno
				TISÍCE:
				Způsob pozvednutí nástroje
				0 = pozvednout na návratovou rovinu
				1 = pozvednout na vztažný bod + bezpečnostní vzdálenost
3	RP	<_RP>	REAL	Návratová rovina (abs)
4	Z0	<_Z0>	REAL	Vztažný bod (abs)
5	SC	<_SC>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
6	Z1	<_Z1>	REAL	Hloubka vrtání/navrtávání středících důlků, (viz JEDNOTKY parametru <_AMODE>)
7	F	<_F>	REAL	Posuv pro vrtání/navrtávání středících důlků

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
8	DXY	<_DXY>	REAL	Přísuv v rovině - jednotky (viz DESÍTKY parametru <_AMODE>)			
9	UXY	<_UXY>	REAL	Přídavek rozměru pro opracování načisto v rovině			
10	UZ	<_UZ>	REAL	Přídavek rozměru pro opracování načisto na dně			
11		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování	
					1 =	Nesousledné frézování	
12	TR	<_TR>	STRING[20]	Název referenčního nástroje			
13	DR	<_DR>	INT	D-číslo referenčního nástroje			
14		<_UMODE>	INT	rezervováno			
15		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)			
				JEDNOTKY:		rezervováno	
				DESÍTKY:		rezervováno	
				STOVKY:		Volba opracování/pouze výpočet počátečního bodu	
				0 =	normální opracování (není zapotřebí žádný režim kompatibility)		
				1 =	normální opracování		
				2 =	rezervováno		
25		<_DMODE>	INT	Režim zobrazování			
				JEDNOTKY:		Rovina obrábění G17/G18/G19	
				0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu		
				1 =	G17 (aktivní jen v cyklu)		
				2 =	G18 (aktivní jen v cyklu)		
				3 =	G19 (aktivní jen v cyklu)		
				DESÍTKY:		technologický režim	
1 =	Předvrtání						
2 =	Navrtávání středících důlků						
26		<_AMODE>	INT	Alternativní režim			
				JEDNOTKY:		Hloubka vrtání/navrtávání středících důlků Z1	
				0 =	absolutní (režim kompatibility)		
				1 =	inkrementální		
				DESÍTKY:		Jednotky pro přísuv v rovině (DXY)	
				0 =	mm		
1 =	procentuální hodnota z průměru nástroje						

20.1.15 CYCLE70 - Frézování závitu

Syntaxe

```
CYCLE70 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Délka závitu (abs, ink), viz <_AMODE> Zohledňovat vyjždění na dně vyvrtané díry (minimálně polovina stoupání)		
5	Ø	<_DIATH>	REAL	Jmenovitý průměr závitu		
6	H1	<_H1>	REAL	Hloubka závitu		
7	U	<_FAL>	REAL	Přídavek rozměru pro opracování načisto		
8	P	<_PIT>	REAL	Stoupání závitu (možnosti na výběr <_PITA>: mm, palce, MODUL, chodů/palec)		
9	NT	<_NT>	INT	Počet zubů na řezné destičce Délka nástroje je vždy vztažena na spodní zub!		
10	DXY	<_MID>	REAL	Maximální přísuv za řez <_MID> > <_H1>: vše s jedním řezem		
11	F	<_FFR>	REAL	Posuv při frézování		
12		<_TYPTH>	INT	Typ závitu	0 =	Vnitřní závit
					1 =	Vnější závit
13	X0	<_PA>	REAL	Střed kruhu, na 1. ose (abs)		
14	Y0	<_PO>	REAL	Střed kruhu, na 2. ose (abs)		
15	αS	<_NSP>	REAL	Počáteční úhel (vícechodý závit)		
16		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:		
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
				DESÍTKY:		
					1 =	shora dolů
					2 =	zdola nahoru
				STOVKY:		
					0 =	Pravý závit
					1 =	Levý závit

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
17		<_PITA>	INT	Vyhodnocování stoupání závitu
				0 = Režim kompatibility
				1 = stoupání v mm
				2 = stoupání v chodech na palec (TPI)
				3 = stoupání v palcích
				4 = stoupání jako MODUL
18		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)
19		<_PTAB>	STRING[20]	Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC") (jen pro pracovní plochu)
20		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) (pouze pro povrch)
21		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
				STOVKY: Opracování/výpočet počátečního bodu
				0 = Režim kompatibility
				1 = normální opracování
22		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
23		<_AMODE>	INT	Alternativní režim
				JEDNOTKY: Délka závitu (<_DP>)
				0 = absolutní
				1 = inkrementální

20.1.16 CYCLE72 - Frézování po dráze

Syntaxe

```
CYCLE72 (<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametru	Interní parametry	Datový typ	Význam
1		<_KNAME>	STRING [141]	Název podprogramu kontury
2	RP	<_RTP>	REAL	Návratová rovina (abs)
3	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
4	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
5	Z1	<_DP>	REAL	Koncový bod, konečná hloubka (abs/ink), viz <_AMODE>
6	DZ	<_MID>	REAL	Maximální přísuv do hloubky (ink, zadává se bez znaménka)
7	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině (ink), přídavek rozměru na okraji kontury
8	UZ	<_FALD>	REAL	Přídavek rozměru pro obrábění načisto do hloubky (ink), přídavek rozměru na dně (zadává se bez znaménka)
9	FX	<_FFP1>	REAL	Posuv na kontuře
10	FZ	<_FFD>	REAL	Posuv pro přísuv do hloubky (nebo přísuv v prostoru)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
11		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				Opracování
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				5 = Srážení hran
				DESÍTKY:
				0 = Pomocné úseky dráhy s G0
				1 = Pomocné úseky dráhy s G1
				STOVKY:
				Zpětný pohyb na konci kontury
				0 = Zpětný pohyb na konci kontury na vztažný bod
				1 = Zpětný pohyb na konci kontury na vztažný bod + <_SDIS>
12		<_RL>	INT	2 = Zpětný pohyb na konci kontury o <_SDIS>
				3 = Žádný zpětný pohyb na konci kontury, na následující počáteční bod se najíždí s posuvem po kontuře
				TISÍCE:
				rezervováno
				DESÍTKY TISÍC:
				Opracování kontury
				0 = Opracování kontury směrem dopředu
				1 = Opracování kontury směrem pozpátku
				Omezení při pohybu směrem pozpátku
				• max. 170 konturových prvků (včetně faset nebo zaoblení)
				• jsou vyhodnocovány pouze hodnoty v rovině (X/Y) a hodnota F
				Směr obrábění
				40 = objíždění kontury středem nástroje (G40, najíždění a odjíždění: pouze po přímce nebo kolmo)
				41 = vlevo od kontury (G41, najíždění a odjíždění: pouze po přímce nebo po kruhové dráze)
				42 = vpravo od kontury (G42, najíždění a odjíždění: pouze po přímce nebo po kruhové dráze)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
13		<_AS1>	INT	Najížděcí pohyb na konturu			
				JEDNOTKY:			
					1 =	Přímka	
					2 =	Čtvrtekruh	
					3 =	Půlkruh	
				4 =	Najíždění a odjíždění kolmo		
				DESÍTKY:			
0 =	poslední pohyb, v rovině						
1 =	poslední pohyb, v prostoru						
14	L1	<_LP1>	REAL	Najížděcí dráha nebo najížděcí rádius (ink, zadává se bez znaménka)			
15	FZ	<_FF3>	REAL	Posuv pro pomocné úseky dráhy (G94/G95, jako na kontuře)			
16		<_AS2>	INT	Odjížděcí pohyb od kontury (nikoli při najíždění/odjíždění kolmo)			
				JEDNOTKY:			
					1 =	Přímka	
					2 =	Čtvrtekruh	
				3 =	Půlkruh		
				DESÍTKY:			
					0 =	poslední pohyb, v rovině	
1 =	poslední pohyb, v prostoru						
17	L2	<_LP2>	REAL	Odjížděcí dráha nebo odjížděcí rádius (ink, zadává se bez znaménka)			
18		<_UMODE>	INT	rezervováno			
19	FS	<_FS>	REAL	Šířka fasety (ink)			
20	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>			
21		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)			
				JEDNOTKY:	rezervováno		
				DESÍTKY:	rezervováno		
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu		
					0 =	Režim kompatibility	
					1 =	normální opracování	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
22		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY:
				Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky
				0 = Režim kompatibility
23		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Koncový bod Z1 (<_DP>)
				0 = absolutní (režim kompatibility)
				1 = inkrementální
				DESÍTKY:
				Jednotky pro přísuv v rovině
				0 = mm, palce
				1 = rezervováno
				STOVKY:
				Hloubka zajiždění při srážení hran (<_ZFS>)
				0 = absolutní
				1 = inkrementální

Poznámka

Pokud jsou následující předávané parametry naprogramovány nepřímo (jako parametry), zpětný překlad vstupní obrazovky nebude možný:

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

20.1.17 CYCLE76 - Frézování pravouhlého čepu

Syntaxe

```
CYCLE76(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_LENG>, <_WID>,
<_CRAD>, <_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>,
<_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_AP2>, <_FS>, <_ZFS>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka čepu (abs)
5		<_DPR>	REAL	Hloubka čepu (ink), vztažená na Z0 (zadává se bez znaménka)
6	L	<_LENG>	REAL	Délka čepu, viz <_GMODE> (zadává se bez znaménka)
7	W	<_WID>	REAL	Šířka čepu, viz <_GMODE> (zadává se bez znaménka)
8	R	<_CRAD>	REAL	Rádus v rohu čepu (zadává se bez znaménka)
9	X0	<_PA>	REAL	Vztažný bod čepu, 1. osa v rovině (abs)
10	Y0	<_PO>	REAL	Vztažný bod čepu, 2. osa v rovině (abs)
11	$\alpha 0$	<_STA>	REAL	Úhel otočení, úhel mezi podélnou osou (L) a 1. osou roviny
12	DZ	<_MID>	REAL	Maximální přísuv do hloubky (ink, zadává se bez znaménka)
13	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině (ink), přídavek rozměru na okraji kontury
14	UZ	<_FALD>	REAL	Přídavek rozměru pro obrábění načisto do hloubky (ink), přídavek rozměru na dně (zadává se bez znaménka)
15	FX	<_FFP1>	REAL	Posuv na kontuře
16	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky
17		<_CDIR>	INT	Směr frézování (zadává se bez znaménka) JEDNOTKY: 0 = Sousedné frézování 1 = Nesousedné frézování
18		<_VARI>	INT	Opracování JEDNOTKY: 1 = Obrábění nahrubo 2 = Obrábění načisto 5 = Srážení hran
19	L1	<_AP1>	REAL	Délka surového čepu
20	W1	<_AP2>	REAL	Šířka surového čepu
21	FS	<_FS>	REAL	Šířka fasety (ink)
22	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs, ink), viz <_AMODE>

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
23		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
				STOVKY: Volba zpracování nebo pouze výpočet počátečního bodu
				0 = Režim kompatibility
				1 = normální zpracování
				TISÍCE: Kótování čepu pomocí středu nebo rohu
				0 = Režim kompatibility
				1 = Zadáání rozměrů pomocí středu
				2 = Kótování pomocí rohového bodu, čep +L +W
				3 = Kótování pomocí rohového bodu, čep -L +W
				4 = Kótování pomocí rohového bodu, čep +L -W
				5 = Kótování pomocí rohového bodu, čep -L -W
				DESÍTKY TISÍC: Kompletní zpracování nebo dopracování
				0 = Režim kompatibility
				1 = Kompletní zpracování
				2 = Dodatečné zpracování
24		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: --- rezervováno
				STOVKY: --- rezervováno
				TISÍCE: --- rezervováno
				DESÍTKY TISÍC: Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadáání: kompletní
				1 = Zadáání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
25		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Koncová hloubka Z1 (DP)
				0 = Kompatibilita
				1 = inkrementální
				2 = absolutní
				DESÍTKY: rezervováno
				STOVKY: Hloubka zajištění při srážení hran (ZFS)
				0 = absolutní
				1 = inkrementální

20.1.18 CYCLE77 - Frézování kruhového čepu

Syntaxe

CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>, <_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka čepu (abs)
5		<_DPR>	REAL	Hloubka čepu (ink), vztažená na Z0 (zadává se bez znaménka)
6	Ø	<_CDIAM>	REAL	Průměr čepu (zadává se bez znaménka)
7	X0	<_PA>	REAL	Vztažný bod čepu, 1. osa v rovině (abs)
8	Y0	<_PO>	REAL	Vztažný bod čepu, 2. osa v rovině (abs)
9	DZ	<_MID>	REAL	Maximální přísuv do hloubky (ink, zadává se bez znaménka)
10	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině (ink), přídavek rozměru na okraji kontury
11	UZ	<_FALD>	REAL	Přídavek rozměru pro obrábění načisto do hloubky (ink), přídavek rozměru na dně (zadává se bez znaménka)
12	FX	<_FFP1>	REAL	Posuv na kontuře
13	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky
14		<_CDIR>	INT	Směr frézování (zadává se bez znaménka)
				JEDNOTKY:
				0 = Sousledné frézování
				1 = Nesousledné frézování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
15		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				Opracování
				1 = Obrábění nahrubo až na přídavek rozměru pro opracování načisto 2 = Obrábění načisto (přídavek rozměru X/Y/Z = 0) 5 = Srážení hran
16	Ø1	<_AP1>	REAL	Průměr surového čepu
17	FS	<_FS>	REAL	Šířka fasety (ink)
18	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>
19		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY:
				rezervováno
				DESÍTKY:
				rezervováno
				STOVKY:
				Volba opracování/pouze výpočet počátečního bodu
				0 = Režim kompatibility 1 = normální opracování
				TISÍCE:
				rezervováno
20		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu 1 = G17 (aktivní jen v cyklu) 2 = G18 (aktivní jen v cyklu) 3 = G19 (aktivní jen v cyklu)
				DESÍTKY:
				--- rezervováno
				STOVKY:
				--- rezervováno
				TISÍCE:
				--- rezervováno
				DESÍTKY TISÍC:
				Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadání: kompletní 1 = Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
21		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Koncová hloubka Z1 (DP)
				0 = absolutní (režim kompatibility)
				1 = inkrementální
				2 = absolutní
				DESÍTKY: rezervováno
				STOVKY:
				Hloubka zajištění při srážení hran (ZFS)
				0 = absolutní
				1 = inkrementální

20.1.19 CYCLE78 - Frézování vrtaných závitů

Syntaxe

```
CYCLE78 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_ADPR>, <_FDPR>, <_LDPR>,
<_DIAM>, <_PIT>, <_PITA>, <_DAM>, <_MDEP>, <_VARI>, <_CDIR>, <_GE>,
<_FFD>, <_FRDP>, <_FFR>, <_FFP2>, <_FFA>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Konečná vrtaná hloubka (abs/ink), viz <_AMODE>
5		<_ADPR>	REAL	Navrtávaná hloubka se sníženou hodnotou posuvu při vrtání (ink), v platnosti podle místa desítek tisíc parametru <_VARI>
6	D	<_FDPR>	REAL	Maximální přísuv do hloubky (ink) D ≥ Z1 --> jeden přísuv až na konečnou vrtanou hloubku D < Z1 --> cyklus pro vrtání hlubokých děr s větším počtem přísuvů a s odstraňováním třísek
7	ZR	<_LDPR>	REAL	Zbytková vrtaná hloubka při provrtávání (ink), s posuvem FR
8	Ø	<_DIAM>	REAL	Jmenovitý průměr závitu
9	P	<_PIT>	REAL	Stoupání závitu jako číselná hodnota
10		<_PITA>	INT	Vyhodnocování stoupání závitu P
				1 = stoupání v mm/ot
				2 = stoupání v chodech/palec
				3 = stoupání v palcích/ot
				4 = stoupání jako MODUL

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
11	DF	<_DAM>	REAL	Absolutní hodnota/procentuální změna pro každý další přísuv (degrese), viz <_AMODE>
12	V1	<_MDEP>	REAL	minimální přísuv (ink), jen když je používána degrese
13		<_VARI>	INT	Způsob opracování JEDNOTKY: rezervováno DESÍTKY: Odstraňování třísek před frézováním závitu 0 = žádné odstraňování třísek před frézováním závitu (uplatňuje se pouze na konečné vrtané hloubce) 1 = odstraňování třísek před frézováním závitu (uplatňuje se pouze na konečné vrtané hloubce) STOVKY: Právý/levý závit 0 = Právý závit 1 = Levý závit TISÍCE: Zbývající vrtaná hloubka s posuvem pro vrtání 0 = žádná zbytková vrtaná hloubka s posuvem při vrtání FR 1 = zbytková vrtaná hloubka s posuvem při vrtání FR DESÍTKY TISÍC: Navrtávání se sníženou hodnotou posuvu 0 = žádné navrtávání se sníženou hodnotou posuvu 1 = Navrtávání se sníženou hodnotou posuvu posuv při navrtávání = 0,3 F1, pokud je $F1 < 0,15 \text{ mm/ot}$ posuv při navrtávání = 0,1 mm/ot, pokud je $F1 \geq 0,15 \text{ mm/ot}$
14		<_CDIR>	INT	Směr frézování 0 = Sousledné frézování 1 = Nesousledné frézování 4 = nesousledné + sousledné (kombinace obrábění nahrubo + načisto)
15	Z2	<_GE>	REAL	Vzdálenost zpětného pohybu před frézováním závitu (ink)
16	F1	<_FFD>	REAL	Posuv při vrtání (mm/min, příp. palce/min nebo mm/ot)
17	FR	<_FRDP>	REAL	Posuv při vrtání pro zbytkovou vrtanou hloubku (mm/min nebo mm/ot)
18	F2	<_FFR>	REAL	Posuv pro frézování závitu (mm/min nebo mm/zub)
19	FS	<_FFP2>	REAL	Posuv při obrábění načisto pro <_CDIR> = 4 (mm/min nebo mm/zub)

Č.	Obrázovka parametrů	Interní parametry	Datový typ	Význam
20		<_FFA>	INT	Vyhodnocování posuvu
				JEDNOTKY: posuv při vrtání F1
				DESÍTKY: posuv při vrtání pro zbytkovou vrtanou hloubku FR
				STOVKY: Posuv pro frézování závitu F2
				TISÍCE: Posuv při obrábění načisto FS
21		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu) ¹⁾
22		<_PTAB>	STRING[20]	Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC") (jen pro pracovní plochu) ¹⁾
23		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) (pouze pro povrch) ¹⁾
24		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty), rezervováno
25		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
26		<_AMODE>	INT	Alternativní režim
				JEDNOTKY: Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)
				0 = absolutní
				1 = inkrementální
				DESÍTKY: Absolutní hodnota/procentuální hodnota DF pro každý další přísuv (degrese)
				0 = absolutní hodnota
				1 = procentuální změna (0,001 až 100%)

Poznámka

¹⁾ Parametry 21, 22 a 23 se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce. Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný.

20.1.20 CYCLE79 - Mnohohran

Syntaxe

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka mnohohranu (abs/ink), viz <_AMODE>
5	N	<_NUM>	INT	Počet hran (1...n)
6	SW/L	<_SWL>	REAL	Velikost klíče nebo délka hrany (v závislosti na <_VARI>) ("SW" v případě velikosti klíče, "L" v případě délky hrany) Velikost klíče jen u sudého počtu hran a jednohranu
7	X0	<_PA>	REAL	Vztažný bod čepu, 1. osa (abs)
8	Y0	<_PO>	REAL	Vztažný bod čepu, 2. osa (abs)
9	α0	<_STA>	REAL	Úhel otočení středu hrany vzhledem k 1. ose (osa X)
10	R1/FS1	<_RC>	REAL	Rohové zaoblení v případě <_NUM> > 2 (rádius/faseta, viz <_AMODE>) (ink, zadává se bez znaménka) ("R1" v případě radiusu, "FS1" v případě fasety)
11	Ø	<_AP1>	REAL	Průměr surového čepu
12	DXY	<_MIDA>	REAL	Maximální přísuv do šířky (jednotky viz <_AMODE>)
13	DZ	<_MID>	REAL	Maximální přísuv do hloubky
14	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině
15	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně
16	F	<_FFP1>	REAL	Pracovní posuv
17		<_CDIR>	INT	Směr frézování
				0 = Sousedné frézování
				1 = Nesousedné frézování
18		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				Opracování
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				3 = Obrábění okraje načisto
				5 = Srážení hran
				DESÍTKY:
				Velikost klíče nebo délka hrany
				0 = Šířka klíče
				1 = Délka hrany

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
19	FS	<_FS>	REAL	Šířka fasety (ink)
20	ZFS	<_ZFS>	REAL	Hloubka zajiždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>
21		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty) JEDNOTKY: rezervováno DESÍTKY: rezervováno STOVKY: Volba opracování nebo pouze výpočet počátečního bodu 1 = normální opracování
22		<_DMODE>	INT	Režim zobrazování JEDNOTKY: Rovina obrábění G17/G18/G19 0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu 1 = G17 (aktivní jen v cyklu) 2 = G18 (aktivní jen v cyklu) 3 = G19 (aktivní jen v cyklu) DESÍTKY: --- rezervováno STOVKY: --- rezervováno TISÍCE: --- rezervováno DESÍTKY TISÍC: Odstupňování technologie na obrazovkách cyklů (Strana 842) 0 = Zadání: kompletní 1 = Zadání: jednoduché
23		<_AMODE>	INT	Alternativní režim JEDNOTKY: Konečná hloubka (<_DP>) 0 = absolutní 1 = inkrementální DESÍTKY: Jednotky pro přísuv v rovině (<_MIDA>) 0 = mm 1 = procentuální hodnota z průměru nástroje STOVKY: Hloubka zajiždění při srážení hran (<_ZFS>) 0 = absolutní 1 = inkrementální TISÍCE: Zaoblením v rohu (<_RC>) 0 = Rádus 1 = Faseta

20.1.21 CYCLE81 - Vrtání, navrtávání středících důlků

Syntaxe

CYCLE81 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1/Ø	<DP>	REAL	Vrtaná hloubka (abs)/průměr středícího důlku (abs), viz <_GMODE>
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink)
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>
7		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: Navrtávání středícího důlku je vztaženo na hloubku/průměr
				0 = Kompatibilita, hloubka
				1 = Průměr
8		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
9		<_AMODE>	INT	Alternativní režim
				JEDNOTKY: Vrtaná hloubka Z1 (abs/ink)
				0 = kompatibilita, podle naprogramování DP/DPR
				1 = inkrementální
				2 = absolutní
				DESÍTKY: Doba prodlevy na konečné vrtané hloubce DT v sekundách/otáčkách
				0 = kompatibilita, vyplývá ze znaménka DTB (> 0 sekundy nebo < 0 otáčky)
				1 = v sekundách
				2 = v otáčkách

20.1.22 CYCLE82 - Vrtání, čelní zahlubování

Syntaxe

CYCLE82 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>, <_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)

Parametr

Č.	Obrázovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>
7		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: Vrtaná hloubka vztažena na špičku/stopku
				0 = Kompatibilita, špička
				1 = Stopka
8		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: rezervováno
				STOVKY: rezervováno
				TISÍCE: rezervováno
				DESÍTKY TISÍC: Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadání: kompletní
				1 = Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
9		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Vrtaná hloubka Z1 (abs/ink)
				0 = Kompatibilita, podle naprogramování DP/DPR
				1 = inkrementální
				2 = absolutní
				DESÍTKY:
				Doba prodlevy DT na konečné vrtané hloubce v sekundách/otáčkách
				0 = kompatibilita, vyplývá ze znaménka DT (> 0 sekundy / < 0 otáčky)
				1 = v sekundách
				2 = v otáčkách
				STOVKY:
				Hloubka pro navrtávání ZA abs/ink
				0 = inkrementální
				1 = absolutní
				TISÍCE:
				Vyhodnocování posuvu při navrtávání
				0 = v % z posuvu při vrtání
				1 = F/min
				2 = F/ot
				DESÍTKY TISÍC:
				Zbytková vrtaná hloubka ZD abs/ink
				0 = inkrementální
				1 = absolutní
				STOVKY TISÍC:
				Vyhodnocování zbytkového posuvu při vrtání
				0 = v % z posuvu při vrtání
				1 = F/min
				2 = F/ot
10		<_VARI>	INT	Způsob opracování při navrtávání/provrtávání
				JEDNOTKY:
				rezervováno
				DESÍTKY:
				rezervováno
				STOVKY:
				rezervováno
				TISÍCE:
				Provrtání
				0 = Provrtávání "ne"
				1 = Provrtávání "ano"
				DESÍTKY TISÍC:
				Navrtání
				0 = Navrtávání „ne“
				1 = Navrtávání „ano“
11	ZA	<S_ZA>	REAL	Navrtávaná hloubka vztažena inkrementálně na vztažný bod nebo absolutně (viz <_AMODE> - STOVKY)
12	FA	<S_FA>	REAL	Posuv při navrtávání jako hodnota nebo v % (ve spojení s místem TISÍCŮ parametru <_AMODE>)

Č.	Obrazovka parametru	Interní parametry	Datový typ	Význam
13	ZD	<S_ZD>	REAL	Zbytková vrтанá hloubka vztažena inkrementálně na konečnou vrтанou hloubku nebo absolutně (ve spojení s místem DESÍTEK TISÍC parametru <_AMODE>)
14	FD	<S_FD>	REAL	Posuv pro zbytkovou vrтанou hloubku jako hodnota nebo v % (ve spojení s místem STOVEK TISÍCŮ parametru <_AMODE>)

20.1.23 CYCLE83 - Vrtání hlubokých děr

Syntaxe

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>,
<DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>,
<_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam										
1	RP	<RTP>	REAL	Návratová rovina (abs)										
2	Z0	<RFP>	REAL	Vztažný bod (abs)										
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)										
4	Z1	<DP>	REAL	Konečná vrtaná hloubka (abs), viz <_AMODE>										
5	Z1	<DPR>	REAL	Konečná vrtaná hloubka (ink), viz <_AMODE>										
6	D	<FDEP>	REAL	1. vrtaná hloubka (abs), viz <_AMODE>										
7	D	<FDPR>	REAL	1. vrtaná hloubka (ink), viz <_AMODE>										
8	DF	<_DAM>	REAL	Degrese/procentuální hodnota pro každý další přísuv, viz <_AMODE>										
9	DTB	<DTB>	REAL	Doba prodlevy na vrtané hloubce, viz <_AMODE>										
10	DTS	<DTS>	REAL	Doba prodlevy v počátečním bodě (jen při odstraňování třísek), viz <_AMODE>										
11	FD1	<FRF>	REAL	Procentuální hodnota pro posuv při prvním přísuvu, viz <_AMODE>										
12		<VARI>	INT	<table><tr><td colspan="2">Způsob opracování</td></tr><tr><td>JEDNOTKY:</td><td>Ulamování třísky / odstraňování třísek</td></tr><tr><td>0 =</td><td>Ulamování třísky</td></tr><tr><td>1 =</td><td>Odstraňování třísek</td></tr></table>	Způsob opracování		JEDNOTKY:	Ulamování třísky / odstraňování třísek	0 =	Ulamování třísky	1 =	Odstraňování třísek		
Způsob opracování														
JEDNOTKY:	Ulamování třísky / odstraňování třísek													
0 =	Ulamování třísky													
1 =	Odstraňování třísek													
13		<_AXN>	INT	<table><tr><td colspan="2">Osa nástroje</td></tr><tr><td>0 =</td><td>3. geometrická osa</td></tr><tr><td>1 =</td><td>1. geometrická osa</td></tr><tr><td>2 =</td><td>2. geometrická osa</td></tr><tr><td>> 2</td><td>3. geometrická osa</td></tr></table>	Osa nástroje		0 =	3. geometrická osa	1 =	1. geometrická osa	2 =	2. geometrická osa	> 2	3. geometrická osa
Osa nástroje														
0 =	3. geometrická osa													
1 =	1. geometrická osa													
2 =	2. geometrická osa													
> 2	3. geometrická osa													
14	V1	<_MDEP>	REAL	minimální přísuv (jen když je degrese v procentech)										

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
15	V2	<_VRT>	REAL	Vzdálenost zpětného pohybu po každém obrobení (jen při ulamování třísky)
				> 0 proměnná vzdálenost zpětného pohybu
				0 = standardní hodnota 1 mm
16	DT	<_DTD>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>
17	V3	<_DIS1>	REAL	Chráněná vzdálenost (jen při odstraňování třísek), viz <_AMODE>
18		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: Vrtaná hloubka vztažena na špičku/stopku
				0 = Špička
				1 = Stopka
19		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: --- rezervováno
				STOVKY: --- rezervováno
				TISÍCE: --- rezervováno
				DESÍTKY TISÍC: Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadáni: kompletní
				1 = Zadáni: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
20		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)
				0 = Kompatibilita, podle naprogramování <DP>/<DPR>
				1 = inkrementální
				2 = absolutní
				DESÍTKY:
				Doba prodlevy na vrtané hloubce DTB v sekundách/otáčkách
				0 = Kompatibilita, vyplývá ze znaménka DTB (> 0 sekundy nebo < 0 otáčky)
				1 = v sekundách
				2 = v otáčkách
				STOVKY:
				Doba prodlevy v počátečním bodě před DTS v sekundách/otáčkách
				0 = kompatibilita, vyplývá ze znaménka DTS (> 0 sekundy nebo < 0 otáčky)
				1 = v sekundách
				2 = v otáčkách
				TISÍCE:
				Doba prodlevy na konečné vrtané hloubce DTD v sekundách/otáčkách
				0 = Kompatibilita, vyplývá ze znaménka DTD (> 0 sekundy nebo < 0 otáčky)
				1 = v sekundách
				2 = v otáčkách
				DESÍTKY TISÍC:
				1. vrtaná hloubka D (abs/ink)
				0 = Kompatibilita, podle naprogramování <FDEPF>/<DPR>
				1 = inkrementální
				2 = absolutní
				STOVKY TISÍC:
				Degrese/procentuální hodnota <_DAM> pro každý další přísuv
				0 = Kompatibilita, vyplývá ze znaménka <_DAM> (> 0 hodnota degrese nebo < 0 faktor 0,001 až 1,0)
				1 = Absolutní hodnota degrese
				2 = Procentuální změna (0,001 až 100%)
				JEDNOTKY MILIONŮ:
				Chráněná vzdálenost V3 automaticky/manuálně

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
				0 = kompatibilita, vyplývá ze znaménka <_DIS1> (= 0 automaticky nebo > 0 manuálně)
				1 = automaticky (vypočítává se v cyklu)
				2 = manuálně (naprogramovaná hodnota)
				DESÍTKY MILIONŮ: Faktor posuvu pro první přísuv <FRF> jako faktor/procentuální změna
				0 = kompatibilita, jako faktor (0,001 až 1,0, FRF = 0 znamená 100%)
				1 = Procentuální změna (0,001 až 999,999%)

20.1.24 CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky

Syntaxe

```
CYCLE84(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDAC>, <MPIT>,
<PIT>, <POSS>, <SST>, <SST1>, <_AXN>, <_PITA>, <_TECHNO>, <_VARI>,
<_DAM>, <_VRT>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<DP>	REAL	Vrtaná hloubka = konečná vrtaná hloubka (abs), viz <_AMODE>
5	Z1	<DPR>	REAL	Vrtaná hloubka = konečná vrtaná hloubka (ink), viz <_AMODE>
6	DT	<DTB>	REAL	Doba prodlevy na vrtané hloubce v sekundách
7	SDE	<SDAC>	INT	Směr otáčení po skončení cyklu
8		<MPIT>	REAL	Velikost závitů, pouze pro "ISO metrický" (stoupání se vypočítává interně v průběhu zpracovávání programu)
9	P	<PIT>	REAL	Stoupání závitů jako hodnota, rozměrové jednotky viz <_PITA>
10	$\alpha S^{(1)}$	<POSS>	REAL	Poloha pro orientované zastavení vřetena
11	S	<SST>	REAL	Otáčky vřetena pro vrtání závitů
12	SR	<SST1>	REAL	Otáčky vřetena pro zpětný pohyb

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
13		<_AXN>	INT	Osa vrtání	0 =	3. Geometrická osa
					1 =	1. Geometrická osa
					2 =	2. Geometrická osa
					≥ 3 =	3. Geometrická osa
14		<_PITA>	INT	Rozměrové jednotky pro stoupání závitu (vyhodnocování <PIT> a <MPIT>)		
					0 =	stoupání v mm - vyhodnocování <MPIT>/<PIT>
					1 =	stoupání v mm - vyhodnocování <PIT>
					2 =	stoupání v TPI - vyhodnocování <PIT> (počet chodů závitu na palec)
					3 =	stoupání v palcích - vyhodnocování <PIT>
					4 =	MODUL - vyhodnocování <PIT>
15		<_TECHNO>	INT	Technologie ¹⁾		
				JEDNOTKY:	Režim přesného najetí	
					0 =	je aktivní chování přesného najetí, jaké bylo před voláním cyklu
					1 =	Přesné najetí G601
					2 =	Přesné najetí G602
					3 =	Přesné najetí G603
				DESÍTKY:	Dopředná regulace	
					0 =	s/bez dopředné regulace, jak bylo aktivní před voláním cyklu
					1 =	s dopřednou regulací FFWON
					2 =	bez dopředné regulace FFWOF
				STOVKY:	Zrychlení	
					0 =	SOFT/BRISK/DRIVE, jak bylo aktivní před voláním cyklu
					1 =	s omezením ryvu SOFT
					2 =	bez omezení ryvu BRISK
					3 =	snížené zrychlení, DRIVE
				TISÍCE:	Režim vřetena při MCALL	
					0 =	s příkazem MCALL se opět aktivuje režim vřetena
					1 =	s příkazem MCALL zůstat v režimu polo-hové regulace

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
16		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				0 = 1 řez
				1 = ulamování třísky (vrtání závitů v hlubokých dírách)
				2 = odstraňování třísek (vrtání závitů v hlubokých dírách)
				TISÍCE:
				Režim ISO/SIEMENS pro vstupní obrazovku není relevantní
				0 = vyvolávání na základě kompatibility s ISO
				1 = vyvolávání na základě kontextu firmy SIEMENS
17	D	<_DAM>	REAL	Maximální přísuv do hloubky (jen při odstraňování třísky/ulamování třísky)
18	V2	<_VRT>	REAL	Vzdálenost zpětného pohybu po každém obrobení (jen při ulamování třísky) viz <_AMODE>
19		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu ²⁾
20		<_PTAB>	STRING[5]	Řetězec pro tabulku závitů ("","ISO","BSW","BSP","UNC") ²⁾
21		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) ²⁾
22		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
23		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: rezervováno
				STOVKY: rezervováno
				TISÍCE:
				Režim kompatibility (jen pro vstupní obrazovku pro zpětný překlad), pokud je MD 52216 Bit0 = 1 ¹⁾
				0 = technologické parametry se vypisují (kompatibilita): parametr TECHNO je v platnosti
				1 = technologické parametry se nevypisují: platí technologie, jaká byla naprogramována před voláním cyklu
				DESÍTKY TISÍC:
				Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadání: kompletní
				1 = Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
24		<_AMODE>	INT	Alternativní režim			
				JEDNOTKY:	Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)		
					0 =	Kompatibilita, podle naprogramování <DP>/<DPR>	
					1 =	inkrementální	
					2 =	absolutní	
				DESÍTKY:		rezervováno	
				STOVKY:		rezervováno	
				TISÍCE:	směr otáčení závitu pravý/levý		
					0 =	kompatibilita, vyplývá ze znaménka parametru PIT/MPIT	
					1 =	vpravo	
					2 =	vlevo	
				DESÍTKY TISÍC:		rezervováno	
				STOVKY TISÍC:		rezervováno	
				JEDNOTKY MILIONŮ:	Vzdálenost zpětného pohybu po každém obrobení V2 manuálně/automaticky		
					0 =	kompatibilita, podle naprogramování parametru <_VRT> (> 0 proměnná hodnota nebo ≤ 0 standardní hodnota 1 mm/0,0394 palce)	
1 =	automaticky (standardní hodnota 1 mm/0,0394 palce)						
2 =	manuálně (jak bylo naprogramováno pro V2)						

¹⁾ V závislosti na nastavovaném parametru SD52216 \$MCS_FUNCTION_MASK_DRILL může být zobrazování polí technologie blokováno.

²⁾ Parametry 19, 22 a 21 se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce. Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný.

20.1.25 CYCLE85 - Vystružování

Syntaxe

CYCLE85 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>
7	F	<FFR>	REAL	Posuv
8	FR	<RFF>	REAL	Posuv při zpětném pohybu
9		<_GMODE>	INT	rezervováno
10		<_DMODE>	INT	Režim zobrazování JEDNOTKY: Rovina obrábění G17/G18/G19 0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu 1 = G17 (aktivní jen v cyklu) 2 = G18 (aktivní jen v cyklu) 3 = G19 (aktivní jen v cyklu)
11		<_AMODE>	INT	Alternativní režim (vrtání) JEDNOTKY: Vrtaná hloubka Z1 (abs/ink) 0 = kompatibilita, podle naprogramování DP/DPR 1 = inkrementální 2 = absolutní DESÍTKY: Doba prodlevy DT na konečné vrtané hloubce v sekundách/otáčkách 0 = kompatibilita, vyplývá ze znaménka DT (> 0 sekundy nebo < 0 otáčky) 1 = v sekundách 2 = v otáčkách

20.1.26 CYCLE86 - Vyvrtávání

Syntaxe

CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>, <RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)

Č.	Obrázovka parametrů	Interní parametry	Datový typ	Význam
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>
7	DIR	<SDIR>	INT	Otáčení vřetena 3 = M3 4 = M4
8	DX	<RPA>	REAL	Vzdálenost pro pozvednutí ve směru X
9	DY	<RPO>	REAL	Vzdálenost pro pozvednutí ve směru Y
10	DZ	<RPAP>	REAL	Vzdálenost pro pozvednutí ve směru Z
11	SPOS	<POSS>	REAL	Poloha vřetena pro pozvednutí (pro orientované zastavení vřetena, ve stupních)
12		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty) JEDNOTKY: Způsob pozvednutí nástroje 0 = pozvednutí, kompatibilita 1 = nepozvednout
13		<_DMODE>	INT	Režim zobrazování JEDNOTKY: Rovina obrábění G17/G18/G19 0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu 1 = G17 (aktivní jen v cyklu) 2 = G18 (aktivní jen v cyklu) 3 = G19 (aktivní jen v cyklu)
14		<_AMODE>	INT	Alternativní režim JEDNOTKY: Vrtaná hloubka Z1 (abs/ink) 0 = Kompatibilita, podle naprogramování <DP>/<DPR> 1 = inkrementální 2 = absolutní DESÍTKY: Doba prodlevy na konečné vrtané hloubce DT v sekundách/otáčkách 0 = kompatibilita, vyplývá ze znaménka DT (> 0 sekundy nebo < 0 otáčky) 1 = v sekundách 2 = v otáčkách

20.1.27 CYCLE92 - Upichování

Syntaxe

CYCLE92 (<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>, <_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	X0	<_SPD>	REAL	Souřadnice vztažného bodu, (abs, vždycky průměr)		
2	Y0	<_SPL>	REAL	Vztažný bod (abs)		
3	X1	<_DIAG1>	REAL	Hloubka pro snížení otáček, viz parametr <_AMODE> (JEDNOTKY)		
4	X2	<_DIAG2>	REAL	Konečná hloubka, viz parametr <_AMODE> (DESÍTKY)		
5	R/FS	<_RC>	REAL	Rádus zaoblení nebo šířka fasety, viz <_AMODE> (TISÍCE)		
6	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
7	S	<_SV1>	REAL	Konstantní otáčky vřetena, viz <_AMODE> (DESÍTKY TISÍC)		
	V			konstantní řezná rychlost		
8	SV	<_SV2>	REAL	Maximální otáčky při konstantní řezné rychlosti		
9	DIR	<_SDAC>	INT	Otáčení vřetena	3 =	pro M3
					4 =	pro M4
10	F	<_FF1>	REAL	Posuv až do hloubky pro snížení otáček		
11	FR	<_FF2>	REAL	Snížená hodnota posuvu až do konečné hloubky		
12	SR	<_SS2>	REAL	Snížená hodnota otáček až do konečné hloubky		
13	XM	<_DIAGM>	REAL	Hloubka pro vyjždění podavače součástek (abs, vždycky průměr)		
14		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Návratová dráha	
					0 =	Zpětný pohyb na rovinu <_SPD> + <_SDIS>
					1 =	žádný zpětný pohyb na konci
				DESÍTKY:	Podavač obrobků	
					0 =	ne, nevyvolávat žádný M-příkaz
					1 =	ano, vyvolání funkce CUST_TECHCYC(101) - vyjždění zásuvky, CUST_TECHCYC(102) - zasunutí zásuvky
15		<_DN>	INT	D-číslo pro 2. břit nástroje, pokud není naprogramováno ⇒ D+1		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
20		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
21		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Hloubka pro snížení otáček (<_DIAG1>)
				0 = absolutně, hodnota příčné osy se zadává jako průměr
				1 = inkrementálně, hodnota příčné osy se zadává jako rádius
				DESÍTKY:
				Konečná hloubka (<_DIAG2>)
				0 = absolutně, hodnota příčné osy se zadává jako průměr
				1 = inkrementálně, hodnota příčné osy se zadává jako rádius
				STOVKY:
				rezervováno
				TISÍCE:
				Rádius / faseta (<_RC>)
				0 = Rádius
				1 = Faseta
				DESÍTKY TISÍC:
				Otáčky vřetena/řezná rychlost (<_SV1>)
				0 = konstantní otáčky vřetena
				1 = konstantní řezná rychlost

20.1.28 CYCLE95 - Oddělování třísky od kontury

Syntaxe

CYCLE95 (<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>, <_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	CON	<NPP>	STRING [140]	Název kontury
2	D	<MID>	REAL	Maximální přísuv do hloubky při obrábění nahrubo, viz <_GMODE>
3	UZ	<FALZ>	REAL	Přídavek rozměru pro opracování načisto ve směru Z
4	UX	<FALX>	REAL	Přídavek rozměru pro opracování načisto ve směru X

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
5	U	<FAL>	REAL	Přídavek rozměru pro opracování načisto rovnoběžně s konturou (uplatňuje se u obou os)		
6	F	<FF1>	REAL	Posuv pro obrábění nahrubo		
7	FY	<FF2>	REAL	Přísuv při opracovávání podříznutí		
8	FS	<FF3>	REAL	Posuv pro obrábění načisto		
9		<_VARI>	INT	Způsob opracování		
				JEDNOTKY a DESÍTKY		
				1 =	Obrábění nahrubo v podélném směru zvnějšku	
				2 =	Obrábění nahrubo v příčném směru zvnějšku	
				3 =	Obrábění nahrubo v podélném směru zevnitř	
				4 =	Obrábění nahrubo v příčném směru zevnitř	
				5 =	Obrábění načisto v podélném směru zvnějšku	
				6 =	Obrábění načisto v příčném směru zvnějšku	
				7 =	Obrábění načisto v podélném směru zevnitř	
				8 =	Obrábění načisto v příčném směru zevnitř	
				9 =	Kompletní opracování v podélném směru zvnějšku	
				10 =	Kompletní opracování v příčném směru zvnějšku	
				11 =	Kompletní opracování v podélném směru zevnitř	
				12 =	Kompletní opracování v příčném směru zevnitř	
				STOVKY:		
				0 =	s vyjžděním podél kontury, bez růžků zbytkového materiálu	
				1 =	bez vyjždění podél kontury	
				2 =	vyjždění podél kontury jen k předcházejícímu průsečíku, mohou vznikat růžky zbytkového materiálu	
10	DT	<DT>	REAL	Doba prodlevy při přerušení posuvu		
11	DI	<DAM>	REAL	Vzdálenost přerušení posuvu		
12	VRT	<_VRT>	REAL	Dráha pozvednutí od kontury		
				0 =	interně se bude používat pozvednutí o 1 mm nezávisle na systému jednotek (palce/mm)	
				> 0 =	dráha pozvednutí nástroje	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
13		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY:
				Vyhodnocování přísluvné hloubky
				0 = Přísluvná hloubka se započítává v souladu s příkazy G-skupiny DIAMON/DIAMOF
				1 = Přísluvná hloubka se uplatňuje jako hodnota rádiusu (nezávisle na DIAMON/DIAMOF)
14		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				TISÍCE:
				0 = Režim kompatibility název kontury je uložen v NPP
				1 = název kontury je naprogramován v cyklu CYCLE62 a předává se v parametru _SC_CONT_NAME

20.1.29 CYCLE98 - Řetězec závitů

Syntaxe

CYCLE98 (<_PO1>, <_DM1>, <_PO2>, <_DM2>, <_PO3>, <_DM3>, <_PO4>, <_DM4>, <APP>, <ROP>, <TDEP>, <FAL>, <_IANG>, <NSP>, <NRC>, <NID>, <_PP1>, <_PP2>, <_PP3>, <_VARI>, <_NUMTH>, <_VRT>, <_MID>, <_GDEP>, <_IFLANK>, <_PITA>, <_PITM1>, <_PITM2>, <_PITM3>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	Z0	<_PO1>	REAL	Souřadnice Z vztažného bodu (abs)
2	X0	<_DM1>	REAL	Souřadnice X vztažného bodu, (abs), jako průměr
3	Z1	<_PO2>	REAL	Vnitřní bod 1, osa Z (abs/ink), viz <_AMODE> (JEDNOTKY)
4	X1	<_DM2>	REAL	Vnitřní bod 1, osa X (abs/ink), viz <_AMODE> (DESÍTKY) nebo
	X1α			Úhel šikmého závitu 1 (-90° až 90°) abs vždy jako průměr, ink vždy jako rádius
5	Z2	<_PO3>	REAL	Vnitřní bod 2, osa Z (abs/ink), viz <_AMODE> (STOVKY)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
6	X2	<_DM3>	REAL	Vnitřní bod 2, osa X (abs/ink), viz <_AMODE> (TISÍCE) nebo
	X2α			Úhel šikmého závitu 2 (-90° až 90°) abs vždy jako průměr, ink vždy jako rádius
7	Z3	<_PO4>	REAL	Koncový bod, osa Z (abs/ink), viz <_AMODE> (DESÍTKY TISÍC)
8	X3	<_DM4>	REAL	Koncový bod, osa X (abs/ink), viz <_AMODE> (STOVKY TISÍC) nebo
	X3α			Úhel šikmého závitu 3 (-90° až 90°) abs vždy jako průměr, ink vždy jako rádius
9	LW	<APP>	REAL	Předsunutý náběh závitu (ink, zadává se bez znaménka)
10	LR	<ROP>	REAL	Výběh závitu (ink, zadává se bez znaménka)
11	H1	<TDEP>	REAL	Hloubka závitu (ink, zadává se bez znaménka)
12	U	<FAL>	REAL	Přídavek rozměru pro opracování načisto ve směrech X a Z
13	DP	<_IANG>	REAL	Šikmý přířuv jako vzdálenost nebo úhel, viz <_AMODE> (JEDNOTKY MILIONŮ)
				Šikmý přířuv se uplatňuje v souladu s nastavením parametru <_VARI> (STOVKY)
	Definice pro parametr<_VARI_HUN-DERTER = 0 - režim kompatibility: > 0 = přířuv po jedné stěně 0 = přířuv kolmo do závitu < 0 = přířuv střídavě po obou stěnách Definice pro parametr _VARI (STOVKY) <> 0: > 0 = přířuv po kladné stěně 0 = přířuv středem < 0 = přířuv po záporné stěně			
αP				
14	α0	<NSP>	REAL	Úhlové posunutí počátečního bodu pro 1. chod závitu
15		<NRC>	INT	Počet průchodů nástroje při obrábění nahrubo, viz <_VARI> (DESÍTKY TISÍC)
16	NN	<NID>	INT	Počet průchodů nástroje naprázdno
17	P0	<_PP1>	REAL	Stoupání závitu pro 1. úsek závitu, viz <_PITA>
18	P1	<_PP2>	REAL	Stoupání závitu pro 2. úsek závitu, viz <_PITA>
19	P2	<_PP3>	REAL	Stoupání závitu pro 3. úsek závitu, viz <_PITA>

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
20		<_VARI>	INT	Opracování
				JEDNOTKY:
				Technologie
				1 = vnější závit s lineárním přísuvem
				2 = vnitřní závit s lineárním přísuvem
				3 = vnější závit s degresivním přísu- vem, průřez třísky zůstává kon- stantní
				4 = vnitřní závit s degresivním přísu- vem, průřez třísky zůstává kon- stantní
				DESÍTKY: rezervováno
				STOVKY: Způsob přísluvu
				0 = Režim kompatibility pro <_IANG>
				1 = přísluv na jedné straně
				2 = přísluv střídavě na obou stranách
				TISÍCE: rezervováno
				DESÍTKY TISÍC: Alternativní přísluv do hloubky
				0 = kompatibilita, zadání počtu prů- chodů nástroje při hrubování (<_NRC>)
				1 = předdefinovaná hodnota pro 1. přísluv (<_MID>)
21	N	<_NUMTH>	INT	Počet chodů závitu
				Návratová vzdálenost (ink)
				0 = interně se bude používat pozved- nutí o 1 mm nezávisle na systé- mu jednotek (palce/mm)
				> 0 = dráha pozvednutí nástroje
22		<_VRT>	REAL	
23	D1	<_MID>	REAL	První přísluv, viz <_VARI> (DESÍTKY TISÍC)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
24	DA	<_GDEP>	REAL	Střídavá hloubka chodů závitu (uplatňuje se jen u nastavení "vícechodý")
				0 = na střídavou hloubku chodů závitu nebrat ohled
				> 0 = na střídavou hloubku chodů závitu brát ohled
25		<_IFLANK>	REAL	Šikmý přísviv jako šířka (jen pro pracovní plochu)
26		<_PITA>	INT	Vyhodnocování stoupání závitu
				0 = Režim kompatibility pro stoupání závitu, vyhodnocování <_PP1> až <_PP3> jako dříve v souladu s aktivním systémem měřících jednotek (metrický/palec)
				1 = stoupání v mm
				2 = stoupání v TPI (počet chodů závitu na palec)
				3 = stoupání v palcích
				4 = MODUL
27		<_PITM1>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)
28		<_PITM2>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)
29		<_PITM3>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)
30		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: --- rezervováno
				STOVKY: --- rezervováno
				TISÍCE: --- rezervováno
				DESÍTKY TISÍC:
				Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadání: kompletní
				1 = Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
31		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				1. Vnitřní bod, souřadnice Z (Z1)
				0 = absolutní
				1 = inkrementální
				DESÍTKY:
				1. Vnitřní bod, souřadnice X (X1)
				0 = absolutní
				1 = inkrementální
				2 = α
				STOVKY:
				2. Vnitřní bod, souřadnice Z (Z2)
				0 = absolutní
				1 = inkrementální
				TISÍCE:
				2. Vnitřní bod, souřadnice X (X2)
				0 = absolutní
				1 = inkrementální
				2 = α
				DESÍTKY TISÍC:
				koncový bod v ose Z (Z3)
				0 = absolutní
				1 = inkrementální
				STOVKY TISÍC:
				koncový bod, souřadnice X (X3)
				0 = absolutní
				1 = inkrementální
				2 = α
				JEDNOTKY MILIONŮ:
				Volba šikého přísuvu jako úhlu nebo šířky
				0 = Úhel přísuvu <_IANG>
				1 = Šikmá přísuvná plocha <_IFLANK>
				DESÍTKY MILIONŮ:
				jednochový/vícechodý
				0 = režim kompatibility (počáteční úhel <_NSP> se vyhodnocuje)
				1 = jednochový (s úhlovým posunutím počátečního bodu <_NSP>)
				2 = vícechodý

20.1.30 CYCLE99 - Soustružení závitů

Syntaxe

```
CYCLE99(<_SPL>, <_SPD>, <_FPL>, <_FPD>, <_APP>, <_ROP>, <_TDEP>,
<_FAL>, <_IANG>, <_NSP>, <_NRC>, <_NID>, <_PIT>, <_VARI>, <_NUMTH>,
<_SDIS>, <_MID>, <_GDEP>, <_PIT1>, <_FDEP>, <_GST>, <_GUD>,
<_IFLANK>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>,
<_S_XRS>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam	
1	Z0	< _SPL>	REAL	Vztažný bod (abs)	
2	X0	< _SPD>	REAL	Souřadnice vztažného bodu, (abs, vždycky průměr)	
3	Z1	< _FPL>	REAL	Koncový bod ve spojení s parametrem < _AMODE> (JEDNOTKY)	
4	X1	< _FPD>	REAL	Koncový bod ve spojení s parametrem < _AMODE> (DESÍTKY)	
5	LW/LW2	< _APP>	REAL	Náběh závitu ve spojení s parametrem < _AMODE> (STOVKY) nebo Náběh závitu = výběh závitu ve spojení s parametrem < _AMODE> (STOVKY)	
6	LR	< _ROP>	REAL	Výběh závitu	
7	H1	< _TDEP>	REAL	Hloubka závitu	
8	U	< _FAL>	REAL	Přídavek rozměru pro opracování načisto ve směrech X a Z	
9	DP	< _IANG>	REAL	Šikmý přířuv jako vzdálenost nebo úhel ve spojení s parametrem < _AMODE> (TISÍCE)	
	αP			> 0 =	přířuv po kladné stěně
				< 0 =	přířuv po záporné stěně
				0 =	přířuv středem
10	α0	< _NSP>	REAL	Úhlové posunutí počátečního bodu (uplatňuje se jen u nastavení "jednochodý")	
11	ND	< _NRC>	INT	Počet průchodů nástroje při obrábění nahrubo, ve spojení s < _VARI> (DESÍTKY TISÍC)	
12	NN	< _NID>	INT	Počet průchodů nástroje naprázdno	
13	P	< _PIT>	REAL	Stoupání závitu jako hodnota, ve spojení s parametrem < _PITA>	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
14		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				Technologie
				1 = vnější závit s lineárním přísuvem
				2 = vnitřní závit s lineárním přísuvem
				3 = vnější závit s degresivním přísuvem, průřez třísky zůstává konstantní
				4 = vnitřní závit s degresivním přísuvem, průřez třísky zůstává konstantní
				DESÍTKY:
				rezervováno
				STOVKY:
				Způsob přísluvu
				1 = přísluv na jedné straně
				2 = přísluv střídavě na obou stranách
				TISÍCE:
				rezervováno
15	N	<_NUMTH>	INT	DESÍTKY TISÍC:
				Alternativní přísluv do hloubky
				0 = zadání počtu průchodů nástroje při hrubování (<_NRC>)
				1 = předdefinovaná hodnota pro 1. přísluv (<_MID>)
				STOVKY TISÍC:
				Způsob opracování
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				3 = Obrábění nahrubo a obrábění načisto
				JEDNOTKY MILIONŮ:
				Posloupnost opracování u vícechodých závitů
				0 = vzestupné pořadí chodů
				1 = obrácené pořadí chodů
16	VR	<_SDIS>	REAL	Počet chodů závitu
17	VR	<_SDIS>	REAL	Návratová vzdálenost, ink
18	D1	<_MID>	REAL	První přísluvná hloubka, ve spojení s parametrem <_VARI> (DESÍTKY TISÍC)
19	DA	<_GDEP>	REAL	Střídavá hloubka chodů závitu (uplatňuje se jen u nastavení "vícechodý")
				0 = na střídavou hloubku chodů závitu nebrat ohled
				> 0 = na střídavou hloubku chodů závitu brát ohled
20	G	<_PIT1>	REAL	Změna stoupání na otáčku
				0 = stoupání závitu je konstantní (G33)
				> 0 = stoupání závitu se zvětšuje (G34)
				< 0 = stoupání závitu se zmenšuje (G35)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
20		<_FDEP>	REAL	Hloubka zajiždění nástroje (zadáva se bez znaménka)
21	N1	<_GST>	INT	Počáteční chod N1 = 1...N, ve spojení s parametrem <_AMODE> (STOVKY TISÍC)
22		<_GUD>	INT	rezervováno
23		<_IFLANK>	REAL	Šikmý přísuv jako šířka (jen pro pracovní plochu)
24		<_PITA>	INT	Rozměrové jednotky pro stoupání závitu (vyhodnocování PIT a/nebo MPIT)
				0 = stoupání v mm - vyhodnocování MPIT/PIT
				1 = stoupání v mm - vyhodnocování PIT
				2 = stoupání v TPI - vyhodnocování PIT (počet chodů závitu na palec)
				3 = stoupání v palcích - vyhodnocování PIT
				4 = MODUL - vyhodnocování PIT
25		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu) ¹⁾
26		<_PTAB>	STRING[20]	Řetězec pro tabulku závitů (jen pro pracovní plochu) ¹⁾
27		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (jen pro pracovní plochu) ¹⁾
28		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY:
				Druh závitu
				0 = Podélný závit
				1 = Příčný závit
				2 = Kuželový závit
				STOVKY:
				--- rezervováno
				TISÍCE:
				--- rezervováno
				DESÍTKY TISÍC:
				Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadání: kompletní
				1 = Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
29		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Délka závitu ve směru Z
				0 = absolutní
				1 = inkrementální
				DESÍTKY:
				Délka závitu ve směru X
				0 = absolutně, hodnota příčné osy se zadává jako průměr
				1 = inkrementálně, hodnota příčné osy se zadává jako rádius
				2 = α
				STOVKY:
				Vyhodnocování náběhu <_APP>
				0 = Předsunutý náběh závitu <_APP>
				1 = Náběh závitu = výběh závitu <_APP> = -<_ROP>
				2 = Je zadán náběh závitu <_APP> = -<_APP>
				TISÍCE:
				Volba šikého přísuvu jako úhlu nebo šířky
				0 = Úhel přísuvu <_IANG>
				1 = Šikmá přísuvná plocha <_IFLANK>
30	XS / RS	<_S_XRS>	REAL	DESÍTKY TISÍC:
				jednochový/vícechodý
				0 = jednochový (s úhlovým posunutím počátečního bodu <_NSP>)
				1 = vícechodý
				STOVKY TISÍC:
				Počáteční chod <_GST>
				0 = kompletní opracování
				1 = zahájit opracování od tohoto chodu
				2 = opracovat jen tento chod
				JEDNOTKY MILIONŮ:
				Kompenzace průhybu u podélných závitů
				0 = Výška segmentu sudovitého závitu XS
				1 = Rádius sudovitého závitu RS
30	XS / RS	<_S_XRS>	REAL	Kompenzace průhybu u podélných závitů ve spojení s parametrem <_AMODE>: JEDNOTKY MILIONŮ

Poznámka

¹⁾ Parametry <_PITM>, <_PTAB> a <_PTABA> se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce.
Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný.

20.1.31 CYCLE435 - Nastavení souřadného systému orovnávače

Syntaxe

CYCLE435 (<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>, <S_PVL>, <S_PD>, <S_PL>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_T>	STRING[32]	Název nástroje brusného kotouče
2		<_DD>	INT	Číslo bříty brusného kotouče
3		<S_TA>	STRING[32]	Vztažný bod orovnávače - název orovnávače
4		<S_DA>	INT	Číslo bříty orovnávače
5		<S_AD>	REAL	Hodnota orovnávání - průměr
6		<S_AL>	REAL	Hodnota orovnávání - v příčné rovině
7		<S_PVD>	REAL	Posunutí profilovacího nástroje - průměr
8		<S_PVL>	REAL	Posunutí profilovacího nástroje - v příčné rovině
9		<S_PD>	REAL	Přídavek rozměru pro profilování - průměr
10		<S_PL>	REAL	Přídavek rozměru pro profilování - v příčné rovině
11		<_AMODE>	INT	Alternativní režim
JEDNOTKY:				Aktivní nástroj na konci cyklu
				0 = Orovnávač aktivní
				1 = Kotouč aktivní

20.1.32 CYCLE495 - Profilování

Syntaxe

CYCLE495 (<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>, <S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_T>	STRING[20]	Název nástroje brusného kotouče
2		<_DD>	INT	Číslo bříty brusného kotouče
3		<_SC>	REAL	Vzdálenost pro pozvednutí při objíždění překážky, inkrementálně
4		<_F>	REAL	Posuv při profilování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
5		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				Druh profilování
				1 = rovnoběžně s osami
				2 = rovnoběžně s konturou
				DESÍTKY:
				Směr obrábění
				0 = vytahování možné s polohami břitu 1 až 4
				1 = tlakem možné s polohami břitu 1 až 4
				2 = střídavě možné s polohami břitu 1 až 8
				3 = začátek → konec možné s polohami břitu 1 až 8
				4 = konec → začátek možné s polohami břitu 1 až 8
6		<_D>	REAL	Orovnávaná hodnota v případě druhu profilování rovnoběžně s osami
				Orovnávaná hodnota ve směru X v případě roviny G18, příp. Y v případě roviny G19, při druhu profilování rovnoběžně s konturou
				Orovnávaná hodnota ve směru Z v případě rovin G18 a G19, při druhu profilování rovnoběžně s konturou
				Přídavek rozměru pro profilování
				Počet zdvihů v profilovacím programu
11		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
12		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Volba nové profilování/pokračování
				1 = Nový
				2 = Pokračovat
				DESÍTKY:
				Volba přídavku rozměru pro profilování
				0 = od kontury surového obrobku až po nejhlubší bod na kontuře
				1 = od kontury surového obrobku až po nejvyšší bod na kontuře
13		<S_FW>	REAL	Úhel volného řezání orovnávače
14		<S_HW>	REAL	Úhel držáku orovnávače

20.1.33 CYCLE800 - Nakládění

Syntaxe

CYCLE800 (<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>, <_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_FR>	INT	Režim volného vyjíždění:
				0 = žádné volné vyjíždění
				1 = volné vyjíždění strojní osy Z
				2 = volné vyjíždění strojní osy Z a potom XY
				3 = rezervováno
				4 = Maximální volné vyjíždění ve směru nástroje
				5 = Inkrementální volné vyjíždění ve směru nástroje
2		<_TC>	STRING[32]	Název datového bloku nakládění:
				"" (žádný název), pokud existuje jen 1 datový blok nakládění
				"0" deaktivování datového bloku nakládění (vymazání framu nakládění)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
3		<_ST>	INT	Status transformací
				JEDNOTKY:
				0 = nová, nakloпенá rovina se vymaže a znovu se vypočítá s aktuálními parametry
				1 = aditivní, nakloпенá rovina se přičte k aktivní nakloпенé rovině
				DESÍTKY:
				Špička nástroje na jednom místě ano/ne (aktivní jen tehdy, je-li instalována funkce IBN SCHWENKEN)
				0 = špička nástroje nezůstává na jednom místě
				1 = špička nástroje zůstává na jednom místě (TRAORI)
				STOVKY:
				Nastavování polohy/orientace (funkce se zobrazuje ve vstupní obrazovce NAKLÁPĚNÍ)
				0 = nástroj se nenastavuje
				1 = nástroj se nastavuje (zejména fréza)
				2 = nastavit orientaci soustružnického nástroje (pokud je ve funkci IBN SCHWENKEN instalována kinematika osy B pro technologii soustružení)
				3 = nastavit orientaci frézovacího nástroje (pokud je ve funkci IBN SCHWENKEN instalována kinematika osy B pro technologii soustružení)
				TISÍCE:
				Interní parametr naklápění v režimu JOG
				DESÍTKY TISÍC:
				Viz parametr pro směr <_DIR>
				0 = Naklápění „ano“
				1 = naklápění "ne", směr "mínus" ³⁾
				2 = naklápění "ne", směr "plus" ³⁾
				STOVKY TISÍC:
				Viz parametr pro směr <_DIR>
				0 = Kompatibilita
				1 = Volba směru "mínus" optimalizována (jen pro uživatelské rozhraní) ⁴⁾
				2 = Volba směru „plus“ optimalizována (jen pro uživatelské rozhraní) ⁴⁾

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
4		<_MODE> ⁵⁾	INT	Režim naklápění: Vyhodnocování úhlu naklápění a posloupnosti naklápění (bitové kódování!)
				Bit: 7 6
				0 0: Úhel naklopení po osách --> viz parametry <_A>, <_B>, <_C>
				0 1: Prostorový úhel --> viz parametry <_A>, <_B> ¹⁾
				1 0: Projekční úhel --> viz parametry <_A>, <_B>, <_C> ¹⁾
				1 1: Režim naklápění, kruhové osy přímo --> viz parametry <_A>, <_B> ¹⁾
				Bit: 5 4 3 2 1 0 (v případě prostorového úhlu nemají žádný význam!)
				x x x x 0 1 1. otáčení _A okolo osy X
				x x x x 1 0 1. otáčení _A okolo osy Y
				x x x x 1 1 1. otáčení _A okolo osy Z
				x x 0 1 x x 2. otáčení _B okolo osy X
				x x 1 0 x x 2. otáčení _B okolo osy Y
				x x 1 1 x x 2. otáčení _B okolo osy Z
5	X0	<_X0>	REAL	Souřadnice X vztažného bodu před otáčením
6	Y0	<_Y0>	REAL	Souřadnice Y vztažného bodu před otáčením
7	Z0	<_Z0>	REAL	Souřadnice Z vztažného bodu před otáčením
8	X(A)	<_A>	REAL	1. Otáčení podle nastavení v parametru <_MODE>
9	Y(B)	<_B>	REAL	2. Otáčení podle nastavení v parametru <_MODE>
10	Z(C)	<_C>	REAL	3. Otáčení podle nastavení v parametru <_MODE>
11	X1	<_X1>	REAL	Souřadnice X vztažného bodu po otáčení
12	Y1	<_Y1>	REAL	Souřadnice Y vztažného bodu po otáčení
13	Z1	<_Z1>	REAL	Souřadnice Z vztažného bodu po otáčení
14	- nebo +	<_DIR>	INT	Spustit pohyb kruhových os pracovním posuvem (předdefinovaná hodnota = -1!)
				-1 = najíždět na menší hodnotu kruhové osy 1 nebo 2 ²⁾
				+1 = najíždět na větší hodnotu kruhové osy 1 nebo 2 ²⁾
				0 = žádné naklápění (jen vypočítat frame naklápění) ^{1) 3)}
15	FR	<_FR_I>	REAL	Hodnota (ink): inkrementální volné vyjždění ve směru nástroje

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
16		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY:
				Zobrazování hodnot úhlu beta při nastavování orientace nástroje
				0 = Hodnota
				1 = Šipka

Poznámka

Pokud jsou následující předávané parametry naprogramovány nepřímo (jako parametry), zpětný překlad vstupní obrazovky nebude možný: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

¹⁾ Volba je možná, pokud je instalována funkce IBN SCHWENKEN.

²⁾ Volba je možná, pokud je ve funkci IBN SCHWENKEN nastaven vztažný směr kruhové osy 1 nebo 2.

žádné pole pro volbu vztažného směru: není

³⁾ Zobrazování volby naklápění "ne" může být blokováno (SD 55221 Bit 0).

Naklápění "ne" ve směnu "mínus" odpovídá <_DIR> = 0 a _ST DESÍTKY TISÍC = 1

Naklápění "ne" ve směnu "plus" odpovídá <_DIR> = 0 a _ST DESÍTKY TISÍC = 2

⁴⁾ Volba směru kruhové osy 1 nebo 2 se uskutečňuje také tehdy, když se kruhová osa se vztažným směrem nachází v poloze pólu (hodnota polohy se rovná nule).

⁵⁾ Příklad kódování: otáčení po osách, posloupnost otáčení ZYX

binárně: 00011011; decimálně: 27

Identifikátory os XYZ odpovídají geometrickým osám NC kanálu. Otáčení okolo os XYZ se smí uskutečňovat jednotlivě. Např. posloupnost otáčení ZXZ je ve volání cyklu CYCLE800 nepřipustná.

20.1.34 CYCLE801 - Mřížka nebo obdélník

Syntaxe

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

Parametr

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam		
1	X0	<_SPCA>	REAL	Vztažný bod pro polohovací vzor (mřížka/obdélník) na 1. ose (abs)		
2	Y0	<_SPCO>	REAL	Vztažný bod pro polohovací vzor (mřížka/obdélník) na 2. ose (abs)		
3	$\alpha 0$	<_STA>	REAL	Základní úhel otočení (úhel vzhledem k 1. ose)	< 0 =	Otáčení ve směru hodinových ručiček
					> 0 =	Opisování proti směru hodinových ručiček
4	L1	<_DIS1>	REAL	Vzdálenost mezi sloupci (vzdálenost mezi pozicemi na 1. ose, zadává se bez znaménka)		
5	L2	<_DIS2>	REAL	Vzdálenost řádků (vzdálenost mezi pozicemi na 2. ose, zadává se bez znaménka)		
6	N1	<_NUM1>	INT	Počet sloupců		
7	N2	<_NUM2>	INT	Počet řádků		
8		<_VARI>	INT	Způsob zpracování		
				JEDNOTKY:	Polohovací vzor	
					0 =	Mřížka
					1 =	Obdélník
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
9		<_UMODE>	INT	rezervováno		
10	αX	<_ANG1>	REAL	Úhel skosení vzhledem k 1. ose (zešikmení řádků vzhledem k 1. ose)		
					< 0 =	měření ve směru hodinových ručiček (0 až -90 stupňů)
					> 0 =	měření proti směru hodinových ručiček (0 až 90 stupňů)
11	αY	<_ANG2>	REAL	Úhel skosení vzhledem k 2. ose (zešikmení sloupců vzhledem ke 2. ose)		
					< 0 =	měření ve směru hodinových ručiček (0 až -90 stupňů)
					> 0 =	měření proti směru hodinových ručiček (0 až 90 stupňů)

Č.	Obrazovka parametru	Parametry interní	Datový typ	Význam
12		<_HIDE>	STRING [200]	Přeskakované pozice - max. 198 znaků - Údaj pořadového čísla pozice, např. "1,3" (pozice 1 a 3 se nebudou uskutečňovat)
13		<_NSP>	INT	rezervováno
14		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)

20.1.35 CYCLE802 - Libovolné pozice

Syntaxe

```
CYCLE802 (<_XA>, <_YA>, <_X0>, <_Y0>, <_X1>, <_Y1>, <_X2>, <_Y2>,
<_X3>, <_Y3>, <_X4>, <_Y4>, <_X5>, <_Y5>, <_X6>, <_Y6>, <_X7>, <_Y7>,
<_X8>, <_Y8>, <_VARI>, <_UMODE>, <_DMODE>, <S_ABA>, <S_AB0>,
<S_AB1>, <S_AB2>, <S_AB3>, <S_AB4>, <S_AB5>, <S_AB6>, <S_AB7>,
<S_AB8>)
```

Parametr

Č.	Obrazovka parametru	Parametry interní	Datový typ	Význam
1		<_XA>	INT	Alternativa pro všechny pozice na ose X (9-místné desítkové číslo) Číslo místa: 876543210 (místo odpovídá poloze vrtané díry Xn)
				Hodnota místa:
				1 = absolutní (1. naprogramovaná pozice vždy absolutně)
				2 = inkrementální
2		<_YA>	INT	Alternativa pro všechny pozice na ose Y (9-místné desítkové číslo) Číslo místa: 876543210 (místo odpovídá poloze vrtané díry Yn)
				Hodnota místa:
				1 = absolutní (1. naprogramovaná pozice vždy absolutně)
				2 = inkrementální
3	X0	<_X0>	REAL	1. poloha X
4	Y0	<_Y0>	REAL	1. poloha Y
5	X1	<_X1>	REAL	2. poloha X
6	Y1	<_Y1>	REAL	2. poloha Y

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam
7	X2	<_X2>	REAL	3. poloha X
8	Y2	<_Y2>	REAL	3. poloha Y
9	X3	<_X3>	REAL	4. poloha X
10	Y3	<_Y3>	REAL	4. poloha Y
11	X4	<_X4>	REAL	5. poloha X
12	Y4	<_Y4>	REAL	5. poloha Y
13	X5	<_X5>	REAL	6. poloha X
14	Y5	<_Y5>	REAL	6. poloha Y
15	X6	<_X6>	REAL	7. poloha X
16	Y6	<_Y6>	REAL	7. poloha Y
17	X7	<_X7>	REAL	8. poloha X
18	Y7	<_Y7>	REAL	8. poloha Y
19	X8	<_X8>	REAL	9. poloha X
20	Y8	<_Y8>	REAL	9. poloha Y

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam
21		<_VARI>	INT	Opracování
				STOVKY:
				(jen pro vyvolávání z aplikace JobShop) (zatím jsou vyhodnocovány jen 0 a 2)
				0 =
				žádné zablokování vřetena
				1 =
				vřeteno zablokovat jen při kolmém zajištění nástroje s příkazem G00 nebo G01
				2 =
				vřeteno zablokovat během celého opracování
				TISÍCE:
				rezervováno
				DESÍTKY TISÍC:
				Polohovací vzor s/bez kruhové osy – kombinace os (s nastavením <_VARI> - STOVKY TISÍC)
				0 =
				XY (jen XY, bez kruhové osy, kompatibilita)
				1 =
				X, Y nebo Z a kruhová osa: XA, YB, ZC (1 kruhová osa s geometrickou osou, okolo které se kruhová osa otáčí)
				2 =
				XY a kruhová osa: XYA, XYB, XYC (1 kruhová osa s 1. a 2. geometrickou osou, bez transformace TRACYL)
		<_UMODE>	INT	STOVKY TISÍC:
				Kruhová osa
				0 =
				bez kruhové osy (jen XY, kompatibilita)
				1 =
				osa A (kruhová osa oko osy X)
				2 =
				osa B (kruhová osa oko osy Y)
				3 =
				osa C (kruhová osa oko osy Z)
		<_UMODE>	INT	DESÍTKY MILIONŮ + MILIONY:
				Polohovací vzor s kruhovou osou - offset (pokud se více kruhových otáčí okolo stejné osy; pokud je index příliš veliký, potom 1. osa)
				00 =
				1. osa A, B nebo C nebo při kompatibilitě
				01 =
				2. osa A, B nebo C
		<_UMODE>	INT	...
				19 =
				20. osa A, B nebo C
22		<_UMODE>	INT	Volba vřetena, které má být zablokováno: (jen při vyvolávání z aplikace JobShop) (vyvolání uživatelského cyklu CUST_TCHCYC)
				3 =
				Zablokování/uvolnění hlavního vřetena
		<_UMODE>	INT	23 =
				Zablokování/uvolnění protivřetena

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam
23		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
24		<S_ABA>	INT	Alternativa pro všechny pozice AB (9-místné desetinné číslo) Číslo místa: 876543210 (místo odpovídá poloze ABn)
				Hodnota místa:
				1 = absolutní (1. naprogramovaná pozice vždy absolutně)
				2 = inkrementální
25	A0	<S_AB0>	REAL	1. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou (ve spojení s příkazem <_VARI>)
26	A1	<S_AB1>	REAL	2. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou
27	A2	<S_AB2>	REAL	3. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou
28	A3	<S_AB3>	REAL	4. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou
29	A4	<S_AB4>	REAL	5. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou
30	A5	<S_AB5>	REAL	6. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou
31	A6	<S_AB6>	REAL	7. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou
32	A7	<S_AB7>	REAL	8. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou
33	A8	<S_AB8>	REAL	9. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou

Poznámka

Nepotřebné pozice dané parametry X1/Y1/A1 až X8/Y8/A8 mohou být vypuštěny. Alternativní hodnoty pro parametry <_XA>, <_YA> a <S_ABA> je však zapotřebí úplně zadat pro všech 9 pozic.

V případě polohovacího vzoru XA, ZB nebo ZC (jedna geometrická osa a kruhová osa) je potřeba, aby osa pracovní roviny, se kterou se prostřednictvím polohovacího vzoru nepohybuje (v případě G17 a XA je to osa Y) byla nastavena do požadované polohy ještě před voláním cyklu.

20.1.36 CYCLE830 - Vrtání hlubokých děr 2**Syntaxe**

```
CYCLE830 (<RTP>, <RFP>, <SDIS>, <_DP>, <FDEP>, <_DAM>, <DTB>, <DTS>,
<FRF>, <VARI>, <_MDEP>, <_VRT>, <_DTD>, <_DIS1>, <S_FP>, <S_SDAC2>,
<S_SV2>, <S_FB>, <_SDAC>, <_SV1>, <S_SPOS>, <S_ZA>, <S_FA>, <S_ZP>,
<S_FS>, <S_ZS1>, <S_ZS2>, <S_N>, <S_ZD>, <S_FD>, <S_FR>, <S_SDAC3>,
```

20.1 Technologické cykly

<S_SV3>, <S_CON>, <S_COFF>, <_GMODE>, <_DMODE>, <_AMODE>,
<S_AMODE2>, <S_AMODE3>, <S_ZPV>)

Parametry

Č.	Obrazovka parametru	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, bez znaménka)
4	Z1	<_DP>	REAL	Konečná vrtaná hloubka, abs/ink (viz <_AMODE> - JEDNOTKY)
5	D	<FDEP>	REAL	1. vrtaná hloubka kvůli ulamování třísky/odstraňování třísek absolutně nebo inkrementálně vzhledem ke vztažnému bodu při navrtávání/bez navrtávání nebo vzhledem ke hloubce předvrtané díry (viz <_AMODE> - DESÍTKY TISÍC)
6	DF	<_DAM>	REAL	Absolutní hodnota/procentuální změna pro každý další přísuv - degrese/procentuální změna (viz <_AMODE> - STOVKY TISÍC)
7	DTB	<DTB>	REAL	Doba prodlevy na každé vrtané hloubce (viz <_AMODE> - DESÍTKY)
8	DTS	<DTS>	REAL	Doba prodlevy při odstraňování třísek v počátečním bodě (viz <_AMODE> - STOVKY)
9	FD1	<FRF>	REAL	Procentuální hodnota pro posuv při prvním přísuvu (viz <_AMODE> - DESÍTKY MILIONŮ)
10		<VARI>	INT	Opracování JEDNOTKY: Ulamování třísky/odstraňování třísek 0 = v jednom průchodu nástroje 1 = Ulamování třísky 2 = Odstraňování třísek 3 = Ulamování třísky a odstraňování třísek DESÍTKY: Zpětný pohyb při odstraňování třísek 0 = na hloubku předvrtané díry 1 = na bezpečnostní vzdálenost STOVKY: měkké zaříznutí nástroje 0 = ne 1 = ano TISÍCE: Provrtání 0 = ne 1 = ano DESÍTKY TISÍC: Navrtávání/předvrtaná díra 0 = bez navrtání 1 = s navrtáním 2 = s předvrtáním STOVKY TISÍC: Zpětný pohyb 0 = na hloubku předvrtané díry 1 = na návratovou rovinu

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
11	V1	<_MDEP>	REAL	minimální přísuv inkrementálně (jen když je degrese v procentech)
12	V2	<_VRT>	REAL	Vzdálenost zpětného pohybu po každém obrobení inkrementálně (jen při ulamování třísky)
				0 = standardní hodnota 1 mm > 0 = proměnná vzdálenost zpětného pohybu
13	DT	<_DTD>	REAL	Doba prodlevy na konečné vrtané hloubce (viz <_AMODE> - TISÍCE)
14	V3	<_DIS1>	REAL	Chráněná vzdálenost inkrementálně, jen při odstraňování třísek (viz <_AMODE> - JEDNOTKY MILIONŮ)
15	FP	<S_FP>	REAL	Posuv pro najíždění v předvrtané díře jako hodnota nebo v % (ve spojení s místem STOVEK parametru <S_AMODE2>)
16		<S_SDAC2>	INT	Směr otáčení vřetena při najíždění
				3 = M3
				4 = M4
				5 = M5 (předdefinované)
17	SP	<S_SV2>	REAL	Najíždění s konstantními otáčkami vřetena (viz <S_AMODE2> - DESÍTKY MILIONŮ)
	V4			konstantní řezná rychlost
				Otáčky vřetena v % z otáček vrtáku
18	F	<S_FB>	REAL	Posuv při vrtání (viz JEDNOTKY parametru <S_AMODE2>)
19		<_SDAC>	REAL	Směr otáčení vřetena při vrtání
				3 = M3
				4 = M4
20	S	<_SV1>	REAL	Vrtání s konstantními otáčkami vřetena (viz <S_AMODE2> - JEDNOTKY MILIONŮ)
	V5			konstantní řezná rychlost
21	SPOS	<S_SPOS>	REAL	Poloha vřetena, jen při najíždění s M5
22	ZA	<S_ZA>	REAL	Navrtávaná hloubka vztažena inkrementálně na vztažný bod nebo absolutně (viz <S_AMODE3> - JEDNOTKY)
23	FA	<S_FA>	REAL	Posuv při navrtávání jako hodnota nebo v % (ve spojení s místem DESÍTEK TISÍCŮ parametru <S_AMODE2>)
24	ZP	<S_ZP>	REAL	Předvrtání inkrementálně vzhledem ke vztažnému bodu, absolutně nebo pomocí faktoru vzhledem k průměru vrtané díry (viz <S_AMODE3> - DESÍTKY)
25	FS	<S_FS>	REAL	Posuv při při zařezávání nástroje jako hodnota nebo v % (ve spojení s místem TISÍCŮ parametru <S_AMODE2>)
26	ZS1	<S_ZS1>	REAL	Hloubka každého zařiznutí nástroje s konstantním posuvem (ink)
27	ZS2	<S_ZS2>	REAL	Hloubka každého zařiznutí nástroje pro zvýšení hodnoty posuvu (ink)
28	N	<S_N>	INT	Počet zdvihů kvůli ulomení třísky před každým odstraněním třísek
29	ZD	<S_ZD>	REAL	Zbytková vrtaná hloubka vztažena inkrementálně na konečnou vrtanou hloubku nebo absolutně (ve spojení s místem STOVEK parametru <S_AMODE3>)
30	FD	<S_FD>	REAL	Posuv při vrtání zbytkové hloubky jako hodnota nebo v % (ve spojení s místem DESÍTKY TISÍCŮ parametru <S_AMODE2>)

Č.	Obrázovka parametrů	Interní parametry	Datový typ	Význam																												
31	FR	<S_FR>	REAL	Posuv při zpětném pohybu (ve spojení s parametrem <S_AMODE2> - STOVKY TISÍC)																												
32		<S_SDAC3>	INT	<table><tr><td colspan="2">Směr otáčení vřetena při zpětném pohybu</td></tr><tr><td>3 =</td><td>M3</td></tr><tr><td>4 =</td><td>M4</td></tr><tr><td>5 =</td><td>M5</td></tr></table>	Směr otáčení vřetena při zpětném pohybu		3 =	M3	4 =	M4	5 =	M5																				
Směr otáčení vřetena při zpětném pohybu																																
3 =	M3																															
4 =	M4																															
5 =	M5																															
33	SR	<S_SV3>	REAL	<table><tr><td>Zpětný pohyb s</td><td>konstantními otáčkami vřetena (viz <S_AMODE2> - STOVKY MILIONŮ)</td></tr><tr><td>V6</td><td>konstantní řezná rychlost</td></tr><tr><td></td><td>Otáčky vřetena v % z otáček vrtáku</td></tr></table>	Zpětný pohyb s	konstantními otáčkami vřetena (viz <S_AMODE2> - STOVKY MILIONŮ)	V6	konstantní řezná rychlost		Otáčky vřetena v % z otáček vrtáku																						
Zpětný pohyb s	konstantními otáčkami vřetena (viz <S_AMODE2> - STOVKY MILIONŮ)																															
V6	konstantní řezná rychlost																															
	Otáčky vřetena v % z otáček vrtáku																															
34	Chladicí kapalina zapnuta	<S_CON>	STRING[10]	Chladicí kapalina zapnuta, M-příkaz nebo volání podprogramu																												
35	Chladicí kapalina vypnuta	<S_COFF>	STRING[10]	Chladicí kapalina vypnuta, M-příkaz nebo volání podprogramu																												
36		<_GMODE>	INT	<table><tr><td colspan="2">Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)</td></tr><tr><td>JEDNOTKY:</td><td>rezervováno</td></tr><tr><td>DESÍTKY:</td><td>Vrtaná hloubka vztažena na špičku/stopku</td></tr><tr><td>0 =</td><td>Špička</td></tr><tr><td>1 =</td><td>Stopka</td></tr></table>	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		JEDNOTKY:	rezervováno	DESÍTKY:	Vrtaná hloubka vztažena na špičku/stopku	0 =	Špička	1 =	Stopka																		
Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)																																
JEDNOTKY:	rezervováno																															
DESÍTKY:	Vrtaná hloubka vztažena na špičku/stopku																															
0 =	Špička																															
1 =	Stopka																															
37		<_DMODE>	INT	<table><tr><td colspan="2">Režim zobrazování</td></tr><tr><td>JEDNOTKY:</td><td><table><tr><td colspan="2">Rovina obrábění G17/G18/G19</td></tr><tr><td>0 =</td><td>kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu</td></tr><tr><td>1 =</td><td>G17 (aktivní jen v cyklu)</td></tr><tr><td>2 =</td><td>G18 (aktivní jen v cyklu)</td></tr><tr><td>3 =</td><td>G19 (aktivní jen v cyklu)</td></tr></table></td></tr><tr><td>DESÍTKY:</td><td>rezervováno</td></tr><tr><td>STOVKY:</td><td>rezervováno</td></tr><tr><td>TISÍCE:</td><td>rezervováno</td></tr><tr><td>DESÍTKY TISÍC:</td><td><table><tr><td colspan="2">Odstupňování technologie na obrazovkách cyklů (Strana 842)</td></tr><tr><td>0 =</td><td>Zadání: kompletní</td></tr><tr><td>1 =</td><td>Zadání: jednoduché</td></tr></table></td></tr></table>	Režim zobrazování		JEDNOTKY:	<table><tr><td colspan="2">Rovina obrábění G17/G18/G19</td></tr><tr><td>0 =</td><td>kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu</td></tr><tr><td>1 =</td><td>G17 (aktivní jen v cyklu)</td></tr><tr><td>2 =</td><td>G18 (aktivní jen v cyklu)</td></tr><tr><td>3 =</td><td>G19 (aktivní jen v cyklu)</td></tr></table>	Rovina obrábění G17/G18/G19		0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu	1 =	G17 (aktivní jen v cyklu)	2 =	G18 (aktivní jen v cyklu)	3 =	G19 (aktivní jen v cyklu)	DESÍTKY:	rezervováno	STOVKY:	rezervováno	TISÍCE:	rezervováno	DESÍTKY TISÍC:	<table><tr><td colspan="2">Odstupňování technologie na obrazovkách cyklů (Strana 842)</td></tr><tr><td>0 =</td><td>Zadání: kompletní</td></tr><tr><td>1 =</td><td>Zadání: jednoduché</td></tr></table>	Odstupňování technologie na obrazovkách cyklů (Strana 842)		0 =	Zadání: kompletní	1 =	Zadání: jednoduché
Režim zobrazování																																
JEDNOTKY:	<table><tr><td colspan="2">Rovina obrábění G17/G18/G19</td></tr><tr><td>0 =</td><td>kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu</td></tr><tr><td>1 =</td><td>G17 (aktivní jen v cyklu)</td></tr><tr><td>2 =</td><td>G18 (aktivní jen v cyklu)</td></tr><tr><td>3 =</td><td>G19 (aktivní jen v cyklu)</td></tr></table>	Rovina obrábění G17/G18/G19		0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu	1 =	G17 (aktivní jen v cyklu)	2 =	G18 (aktivní jen v cyklu)	3 =	G19 (aktivní jen v cyklu)																					
Rovina obrábění G17/G18/G19																																
0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu																															
1 =	G17 (aktivní jen v cyklu)																															
2 =	G18 (aktivní jen v cyklu)																															
3 =	G19 (aktivní jen v cyklu)																															
DESÍTKY:	rezervováno																															
STOVKY:	rezervováno																															
TISÍCE:	rezervováno																															
DESÍTKY TISÍC:	<table><tr><td colspan="2">Odstupňování technologie na obrazovkách cyklů (Strana 842)</td></tr><tr><td>0 =</td><td>Zadání: kompletní</td></tr><tr><td>1 =</td><td>Zadání: jednoduché</td></tr></table>	Odstupňování technologie na obrazovkách cyklů (Strana 842)		0 =	Zadání: kompletní	1 =	Zadání: jednoduché																									
Odstupňování technologie na obrazovkách cyklů (Strana 842)																																
0 =	Zadání: kompletní																															
1 =	Zadání: jednoduché																															

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
38		<_AMODE>	INT	Alternativní režim 1
				JEDNOTKY:
				Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)
				0 = inkrementální
				1 = absolutní
				DESÍTKY:
				Doba prodlevy na každé vrtané hloubce DTB v sekundách/otáčkách
				0 = v sekundách
				1 = v otáčkách
				STOVKY:
				Doba prodlevy kvůli odstranění třísek DTS v sekundách/otáčkách
				0 = v sekundách
				1 = v otáčkách
				TISÍCE:
				Doba prodlevy na konečné vrtané hloubce DT v sekundách/otáčkách
				0 = v sekundách
				1 = v otáčkách
				DESÍTKY TISÍC:
				1. vrtaná hloubka D abs/ink
				0 = inkrementální
				1 = absolutní
				STOVKY TISÍC:
				Absolutní hodnota/procentuální hodnota DF pro každý další přísuv (degrese)
				0 = absolutní hodnota
				1 = Procentuální změna (0,001 až 100%)
				JEDNOTKY MILIONŮ:
				Chráněná vzdálenost V3 automaticky/manuálně
				0 = automaticky (vypočítává se v cyklu)
				1 = manuálně (naprogramovaná hodnota)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
39		<S_AMODE2 >	INT	Alternativní režim 2
				JEDNOTKY:
				JEDNOTKY: Posuv při vrtání F
				0 = F/min
				1 = F/ot
				DESÍTKY:
				Vyhodnocování posuvu při navrtávání FA
				0 = v % z posuvu při vrtání
				1 = F/min
				2 = F/ot
				STOVKY:
				Vyhodnocování posuvu pro najíždění na předvrtanou díru FP
				0 = v % z posuvu při vrtání
				1 = F/min
				2 = F/ot
				TISÍCE:
				Vyhodnocování posuvu pro zaříznutí nástroje FS
				0 = v % z posuvu při vrtání
				1 = F/min
				2 = F/ot
				DESÍTKY TISÍC:
				Vyhodnocování posuvu pro provrtávání FD
				0 = v % z posuvu při vrtání
				1 = F/min
				2 = F/ot
				STOVKY TISÍC:
				Posuv při zpětném pohybu FR
				0 = F/min
				1 = Rychlý posuv
				JEDNOTKY MILIONŮ:
				Vrtání - otáčky vřetena/řezná rychlost (S/V5)
				0 = konstantní otáčky vřetena
				1 = konstantní řezná rychlost
				DESÍTKY MILIONŮ:
				Najíždění s otáčkami vřetena/řeznou rychlostí (SP/V4)
				0 = konstantní otáčky vřetena
				1 = konstantní řezná rychlost
				2 = Otáčky vřetena v % z otáček vrtáku
				STOVKY MILIONŮ:
				Zpětný pohyb - otáčky vřetena/řezná rychlost (SR/V6)
				0 = konstantní otáčky vřetena
				1 = konstantní řezná rychlost
				2 = Otáčky vřetena v % z otáček vrtáku

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
40		<S_AMODE3>	INT	Alternativní režim 3
				JEDNOTKY:
				Hloubka pro navrtávání ZA abs/ink
				0 = inkrementální
				1 = absolutní
				DESÍTKY:
				Hloubka předvrtané díry ZP
				0 = inkrementální
				1 = absolutní
				2 = Faktor pro vynásobení průměru díry
41	ZPV	<S_ZPV>	REAL	STOVKY:
				Zbytková vrtaná hloubka ZD ink/abs
				0 = inkrementální
				1 = absolutní

20.1.37 CYCLE832 - Vysokorychlostní obrábění

Syntaxe

CYCLE832 (<S_TOL>, <S_TOLM>, <S_OTOL>)

Poznámka

Cyklus CYCLE832 nezbavuje výrobce stroje povinnosti uskutečnit nezbytné optimalizační úkoly při uvádění stroje do provozu. To se týká optimalizace os podílejících se na obrábění a nastavení parametrů NCU (dopředná regulace, omezení ryvu atd.).

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam																		
1	Tolerance	<S_TOL>	REAL	Tolerance kontury Tolerance kontury odpovídá toleranci geometrických os.																		
2		<S_TOLM>	INT	Druh obrábění (technologie) JEDNOTKY: <table><tr><td>0 =</td><td>Deaktivování</td></tr><tr><td>1 =</td><td>Obrábění načisto (FINISH)</td></tr><tr><td>2 =</td><td>Předběžné obrábění načisto (SE-MIFINISH)</td></tr><tr><td>3 =</td><td>Obrábění nahrubo (ROUGH)</td></tr></table> DESÍTKY: <table><tr><td>0 =</td><td>kompatibilita¹⁾, příp. žádná tolerance orientace</td></tr><tr><td>1 =</td><td>Tolerance orientace v parametru <S_OTOL></td></tr></table> STOVKY ... STOVKY TISÍC obsazeno kvůli kompatibilitě JEDNOTKY MILIONŮ: <table><tr><td>0 =</td><td>Kompatibilita. Automaticky se použije nejlepší funkce pro výrobu forem, jaká je k dispozici: - Volitelný doplněk Top Surface neaktivní: ⇒ Advanced Surface - Volitelný doplněk Top Surface aktivní: ⇒ Top Surface s vyhlazením </td></tr><tr><td>1 =</td><td>Top Surface bez vyhlazení</td></tr><tr><td>2 =</td><td>Top Surface s vyhlazením</td></tr></table>	0 =	Deaktivování	1 =	Obrábění načisto (FINISH)	2 =	Předběžné obrábění načisto (SE-MIFINISH)	3 =	Obrábění nahrubo (ROUGH)	0 =	kompatibilita ¹⁾ , příp. žádná tolerance orientace	1 =	Tolerance orientace v parametru <S_OTOL>	0 =	Kompatibilita. Automaticky se použije nejlepší funkce pro výrobu forem, jaká je k dispozici: - Volitelný doplněk Top Surface neaktivní: ⇒ Advanced Surface - Volitelný doplněk Top Surface aktivní: ⇒ Top Surface s vyhlazením	1 =	Top Surface bez vyhlazení	2 =	Top Surface s vyhlazením
0 =	Deaktivování																					
1 =	Obrábění načisto (FINISH)																					
2 =	Předběžné obrábění načisto (SE-MIFINISH)																					
3 =	Obrábění nahrubo (ROUGH)																					
0 =	kompatibilita ¹⁾ , příp. žádná tolerance orientace																					
1 =	Tolerance orientace v parametru <S_OTOL>																					
0 =	Kompatibilita. Automaticky se použije nejlepší funkce pro výrobu forem, jaká je k dispozici: - Volitelný doplněk Top Surface neaktivní: ⇒ Advanced Surface - Volitelný doplněk Top Surface aktivní: ⇒ Top Surface s vyhlazením																					
1 =	Top Surface bez vyhlazení																					
2 =	Top Surface s vyhlazením																					
3	Tolerance ORI	<S_OTOL>	REAL	Tolerance orientace, příp. identifikace verze, CYCLE832 Parametr tolerance pro orientaci nástroje. Je zapotřebí při zpracovávání programů pro vysokorychlostní obrábění na strojích s možností dynamické transformace orientace (např. při obrábění v 5 osách). Parametr <S_OTOL> musí být naprogramován. To platí i pro aplikace na strojích se 3 osami u programů bez možnosti nastavovat orientaci nástroje (<S_OTOL> = 1).																		

¹⁾ Tolerance orientace je odvozena od tolerance kontury vynásobené faktorem z nastavovaných parametrů cyklů SD55441 až SD55443.

Literatura:

Příručka pro uvedení do provozu základního softwaru a softwaru pro obsluhu, SINUMERIK Operate (IM9), kapitola "Konfigurace funkce Vysokorychlostní obrábění (CYCLE832)"

Zadávání prostého textu

Aby se zlepšila čitelnost volání cyklů, může být parametr <S_TOLM> (způsob obrábění) zadáván také prostým textem. Prosté texty závisejí na nastaveném jazyce. Přípustné jsou následující možnosti zadání:

_OFF	pro	0	Deaktivování
_FINISH	pro	1	Obrábění načisto
_SEMIFIN	pro	2	Předběžné obrábění načisto
_ROUGH	pro	3	Obrábění nahruho
_ORI_FINISH	pro	11	Obrábění načisto se zadáním tolerance orientace
_ORI_SEMIFIN	pro	12	Předběžné obrábění načisto se zadáním tolerance orientace
_ORI_ROUGH	pro	13	Obrábění nahruho se zadáním tolerance orientace
_TOP_SURFACE_SMOOTH_OFF	pro	1000000	Top Surface bez vyhlazení
_TOP_SURFACE_SMOOTH_ON	pro	2000000	Top Surface s vyhlazením

V případě zadávání prostého textu u funkce Top Surface se prosté texty kombinují způsobem uvedeným v následujícím příkladu:

```
CYCLE832 (0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

Poznámka

Prosté texty jsou založeny na názvech funkcí z G-skupiny 59 (režim dynamiky pro dráhovou interpolaci). Pomocí těchto prostých textů jsou stroje se 3 osami a stroje s transformací orientace s více osami (TRAORI) při aplikacích jasně odděleny.

Deaktivování cyklu CYCLE832

Při deaktivování cyklu CYCLE832 musí být parametr <S_TOL> předáván s nulovou hodnotou.

Příklad: CYCLE832 (0, 0, 1)

Syntaxe CYCLE832 () je pro deaktivování cyklu CYCLE832 přípustná také.

Příklady**Příklad 1: CYCLE832 na stroji se 3 osami bez transformace orientace**

a) Volání cyklu se zadáváním prostých textů

Programový kód	Komentář
G710	; systém jednotek je metrický
CYCLE832 (0.004, _FINISH, 1)	; volání cyklu CYCLE832 s parametry: Tolerance kontury = 0,004 mm, druh opracování: Obrábění načisto

Programový kód	Komentář
...	; zpracování programu pro vysokorychlostní obrábění

b) Volání cyklu bez zadávání prostých textů

Programový kód	Komentář
G710	; viz výše
CYCLE832 (0.004,1,1)	; viz výše
...	; viz výše

Příklad 2: CYCLE832 na stroji s 5 osami a s transformací orientace

a) Volání a deaktivování cyklu se zadáváním prostých textů

Programový kód	Komentář
G710	; systém jednotek je metrický
TRAORI	; aktivování transformace orientace
CYCLE832 (0.3,_ORI_ROUGH,0.8)	; volání cyklu CYCLE832 s parametry: Tolerance kontury = 0,3 mm, druh opracování: Obrábění nahruho se zadáním tolerance orientace, tolerance orientace = 0,8 stupně
...	; zpracování programu pro vysokorychlostní obrábění
CYCLE832 (0,_OFF,1)	; tolerance kontury = 0, Druh opracování: Deaktivování cyklu CYCLE832, Tolerance orientace = 0 stupňů

b) Volání a deaktivování cyklu bez zadávání prostých textů

Programový kód	Komentář
G710	; viz výše
TRAORI	; viz výše
CYCLE832 (0.3,13,0.8)	; viz výše
...	; viz výše
CYCLE832 (0,0,1)	; viz výše

20.1.38 CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou

Syntaxe

CYCLE840 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>, <ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
1	RP	<RTP>	REAL	Návratová rovina (abs)			
2	Z0	<RFP>	REAL	Vztažný bod (abs)			
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)			
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>			
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>			
6	DT	<DTB>	REAL	Doba prodlevy na vrtané hloubce/na bezpečnostní vzdálenosti po zpětném pohybu v sekundách, viz <ENC>			
7		<SDR>	INT	Směr otáčení pro zpětný pohyb			
8	SDE	<SDAC>	INT	Směr otáčení po skončení cyklu			
9		<ENC>	INT	Vrtání závitu s vřetenovým snímačem (G33)/vrtání závitu bez vřetenového snímače (G63)			
					0 =	s vřetenovým snímačem	- stoupání z <MPIT>/<PIT> - bez DT
					20 =	s vřetenovým snímačem	- stoupání z <MPIT>/<PIT> - s DT po zpětném pohybu na bezpečnostní vzdálenost
					11 =	bez vřetenového snímače	- stoupání z <MPIT>/<PIT> - s DT na vrtané hloubce
					1 =	bez vřetenového snímače	- Stoupání z naprogramovaného posuvu - s DT na vrtané hloubce (posuv = otáčky stoupání)
10		<MPIT>	REAL	Velikost závitu, pouze pro "ISO metrický" (stoupání se vypočítává interně v průběhu zpracovávání programu) Rozsah hodnot: 3 až 48 (pro M3 až M48), alternativa k <PIT>			
11		<PIT>	REAL	Stoupání závitu jako hodnota, rozměrové jednotky viz <_PITA> Rozsah hodnot: > 0, alternativa k MPIT			
12		<_AXN>	INT	Osa vrtání	0 =	3. Geometrická osa	
					1 =	1. Geometrická osa	
					2 =	2. Geometrická osa	
					≥ 3 =	3. Geometrická osa	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
13		<_PITA>	INT	Rozměrové jednotky pro stoupání závitu (vyhodnocování <PIT> a <MPIT>)
				0 = stoupání v mm - vyhodnocování <MPIT>/<PIT>
				1 = stoupání v mm - vyhodnocování <PIT>
				2 = stoupání v TPI - vyhodnocování <PIT> (počet chodů závitu na palec)
				3 = stoupání v palcích - vyhodnocování <PIT>
				4 = MODUL - vyhodnocování <PIT>
14		<_TECHNO>	INT	Technologie ¹⁾
				JEDNOTKY: Režim přesného najetí
				0 = je aktivní chování přesného najetí, jaké bylo před voláním cyklu
				1 = Přesné najetí G601
				2 = Přesné najetí G602
				3 = Přesné najetí G603
				DESÍTKY: Dopředná regulace
				0 = s/bez dopředné regulace, jak bylo aktivní před voláním cyklu
15		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu ²⁾
16		<_PTAB>	STRING[5]	Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC") ²⁾
17		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) ²⁾
18		<_GMODE>	INT	rezervováno

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
19		<_DMODE>	INT	Režim zobrazování			
				JEDNOTKY:	Rovina obrábění G17/G18/G19		
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu	
					1 =	G17 (aktivní jen v cyklu)	
					2 =	G18 (aktivní jen v cyklu)	
					3 =	G19 (aktivní jen v cyklu)	
				DESÍTKY:		rezervováno	
				STOVKY:		rezervováno	
				TISÍCE:	Režim kompatibility (jen pro vstupní obrazovku pro zpětný překlad), pokud je MD 52216 Bit0 = 1		
					0 =	technologické parametry se vypisují (kompatibilita): parametr TECHNO je v platnosti	
					1 =	technologické parametry se nevypisují: platí technologie, jaká byla naprogramována před voláním cyklu	
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 842)		
0 =	Zadání: kompletní						
1 =	Zadání: jednoduché						
20		<_AMODE>	INT	Alternativní režim			
				JEDNOTKY:	Vrtaná hloubka Z1 (abs/ink)		
					0 =	Kompatibilita, podle naprogramování <DP> / <DPR>	
					1 =	inkrementální	
					2 =	absolutní	

¹⁾V závislosti na nastavovaném parametru SD52216 MCS_FUNCTION_MASK_DRILL může být zobrazování polí technologie blokováno.

²⁾ Parametry 15, 16 a 17 se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce. Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný!

20.1.39 CYCLE899 - Frézování otevřené drážky

Syntaxe

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

Parametr

Č.	Obrazovka parametru	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka drážky (abs/ink), viz <_AMODE>
5	L	<_LENG>	REAL	Délka drážky (ink)
6	W	<_WID>	REAL	Šířka drážky (ink)
7	X0	<_PA>	REAL	Vztažný bod/počáteční pozice 1. osy (abs)
8	Y0	<_PO>	REAL	Vztažný bod/počáteční pozice 2. osy (abs)
9	α0	<_STA>	REAL	Úhel otočení vzhledem k 1. ose
10	DZ	<_MID>	REAL	Maximální příravná hloubka (ink), pouze u trochoidálního frézování
11	DX	<_MIDA>	REAL	Maximální příravná v rovině (viz <_AMODE>)
12	UX	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině
13	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně
14	F	<_FFP1>	REAL	Posuv
15		<_CDIR>	INT	Směr frézování
				JEDNOTKY:
				0 = Sousedné frézování
				1 = Nesousedné frézování
				4 = Střídavě
16		<_VARI>	INT	Opracování
				JEDNOTKY:
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				3 = Obrábění dna načisto
				4 = Obrábění okraje načisto
				5 = Předběžné obrábění načisto
				6 = Srážení hran
				DESÍTKY: rezervováno
				STOVKY: rezervováno
				TISÍCE:
				1 = Trochoidální frézování
				2 = Zapichovací frézování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
17		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
				STOVKY: Volba zpracování/pouze výpočet počátečního bodu
				1 = normální zpracování
				TISÍCE: Zadávání rozměrů pomocí středu / hrany
				0 = Zadávání rozměrů pomocí středu
				1 = Kótování pomocí hrany "vlevo" ("-" směr 1. osy)
				2 = Kótování pomocí hrany "vpravo" ("+" směr 1. osy)
18		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY: Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY: --- rezervováno
				STOVKY: --- rezervováno
				TISÍCE: --- rezervováno
				DESÍTKY TISÍC: Odstupňování technologie na obrazovkách cyklů (Strana 842)
19		<_AMODE>	INT	Alternativní režim
				JEDNOTKY: Hloubka drážky Z1
				0 = absolutní
				1 = inkrementální
				DESÍTKY: Jednotky pro přísuv v rovině (<_MIDA>) DXY
				0 = mm
				1 = procentuální hodnota z průměru nástroje
				STOVKY: Hloubka zajištění při srážení hran ZFS
				0 = absolutní
				1 = inkrementální
20		<_UMODE>	INT	rezervováno
21	FS	<_FS>	REAL	Šířka fasety (ink)
22	ZFS	<_ZFS>	REAL	Hloubka zajištění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>

20.1.40 CYCLE930 - Zápich

Syntaxe

CYCLE930 (<_SPD>, <_SPL>, <_WIDG>, <_WIDG2>, <_DIAG>, <_DIAG2>, <_STA>, <_ANG1>, <_ANG2>, <_RCO1>, <_RCI1>, <_RCI2>, <_RCO2>, <_FAL>, <_IDEP1>, <_SDIS>, <_VARI>, <_DN>, <_NUM>, <_DBH>, <_FF1>, <_NR>, <_FALX>, <_FALZ>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam	
1	X0	<_SPD>	REAL	Souřadnice vztažného bodu v příčné ose (vždycky průměr)	
2	Z0	<_SPL>	REAL	Souřadnice vztažného bodu v podélné ose	
3	B1	<_WIDG>	REAL	Šířka zápichu dole	
4	B2	<_WIDG2>	REAL	Šířka zápichu nahoře (jen pro pracovní plochu)	
5	T1	<_DIAG>	REAL	Hloubka zápichu ve vztažném bodě, v případě absolutních rozměrů a podélného obrábění = průměr, jinak inkrementální rozměr	
6	T2	<_DIAG2>	REAL	Hloubka zápichu naproti vztažného bodu (jen pro pracovní plochu), v případě absolutních rozměrů a podélného obrábění = průměr, jinak inkrementální rozměr	
7	α0	<_STA>	REAL	Úhel šikmé plochy (-180 ≤ <_STA> ≤ 180)	
8	α1	<_ANG1>	REAL	Úhel stěny zápichu 1 (0 ≤ <_ANG1> < 90) na straně zápichu určeného vztažným bodem	
9	α2	<_ANG2>	REAL	Úhel stěny zápichu 2 (0 ≤ <_ANG2> < 90) naproti vztažného bodu	
10	R1/FS1	<_RCO1>	REAL	Rádus zaoblení nebo šířka fasety 1, vně u vztažného bodu	
11	R2/FS2	<_RCI1>	REAL	Rádus zaoblení nebo šířka fasety 2, uvnitř u vztažného bodu	
12	R3/FS3	<_RCI2>	REAL	Rádus zaoblení nebo šířka fasety 3, uvnitř naproti vztažného bodu	
13	R4/FS4	<_RCO2>	REAL	Rádus zaoblení nebo šířka fasety 4, vně naproti vztažného bodu	
14	U	<_FAL>	REAL	Přídavek rozměru pro opracování načisto ve směru X a Z, viz <_VARI> (DESÍTKY TISÍC) (zadává se bez znaménka)	
15	D	<_IDEP1>	REAL	Maximální přísuv do hloubky při zajiždění nástroje (zadává se bez znaménka)	
				0 =	1. průchod nástroje přímo na celou hloubku
				> 0 =	1. průchod nástroje <_IDEP1>, 2. průchod nástroje 2 · <_IDEP1> atd.
16	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (zadává se bez znaménka)	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
17		<_VARI>	INT	Způsob opracování
				JEDNOTKY: rezervováno
				DESÍTKY: Technologie opracování
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				3 = Obrábění nahrubo a obrábění načisto
				STOVKY: Poloha v podélném/příčném směru, vnější/vnitřní +Z/-Z, příp +X/-X
				1 = podélná/vnější +Z
				2 = příčná/vnitřní -X
				3 = podélná/vnitřní +Z
				4 = příčná/vnitřní +X
				5 = podélná/vnější -Z
				6 = příčná/vnější -X
				7 = podélná/vnitřní -Z
				8 = příčná/vnější +X
				TISÍCE: Poloha vztažného bodu
				0 = Vztažný bod nahoře
				1 = Vztažný bod dole
				DESÍTKY TISÍC: Definice platí jako přídavek rozměru pro opracování načisto
				0 = Přídavek rozměru pro opracování načisto U rovnoběžně s konturou
				1 = Přídavek rozměru pro opracování načisto UX a UZ odděleně
18		<_DN>	INT	D-číslo pro 2. břit nástroje
				> 0 = D-číslo pro korekci nástroje pro 2. břit zapichovacího nože
				0 = není naprogramován žádný 2. břit
19	N	<_NUM>	INT	Počet zápichů (0 = 1 zápich)
20	DP	<_DBH>	REAL	Vzdálenost zápichů (je zapotřebí jen tehdy, pokud je <_NUM> > 1)
21	F	<_FF1>	REAL	Posuv
22		<_NR>	INT	Identifikace tvaru zápichu (odpovídá programovým tlačítkům ve svislém pruhu obrazovky)
				0 = 90° stěny zápichů bez faset/zaoblení
				1 = šikmé stěny s fasetami/zaobleními (bez α0)
				2 = jako 1, ale na kuželu (s α0)
23	UX	<_FALX>	REAL	Přídavek rozměru pro opracování načisto ve směru osy X, viz <_VARI> (DESÍTKY TISÍC) (zadáva se bez znaménka)
24	UZ	<_FALZ>	REAL	Přídavek rozměru pro opracování načisto ve směru osy Z, viz <_VARI> (DESÍTKY TISÍC) (zadáva se bez znaménka)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
25		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
26		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Zadávání rozměrů hloubky (jen pro pracovní plochu)
				0 = na vztažný bod
				1 = naproti vztažného bodu
				DESÍTKY:
				Hloubka
				0 = absolutní
				1 = inkrementální
				STOVKY:
				Zadávání rozměrů šířky (jen pro pracovní plochu)
				0 = na vnějším průměru (nahore)
				1 = na vnitřním průměru (dole)
				TISÍCE:
				Rádus / faseta 1 (<_RCO1>)
				0 = Rádus
				1 = Faseta
				DESÍTKY TISÍC:
				Rádus / faseta 2 (<_RCI1>)
				0 = Rádus
				1 = Faseta
				STOVKY TISÍC:
				Rádus / faseta 3 (<_RCI2>)
				0 = Rádus
				1 = Faseta
				JEDNOTKY MILIONŮ:
				Rádus / faseta 4 (<_RCO2>)
				0 = Rádus
				1 = Faseta

20.1.41 CYCLE940 - Odlehčovací zápich určitého tvaru

Pomocí cyklu CYCLE940 mohou být naprogramovány různé odlehčovací zápichy. Pokud jde o dosazování parametrů, jsou mezi nimi značné rozdíly.

Přidané sloupce v tabulce ukazují, které parametry jsou u jednotlivých tvarů zápichu zapotřebí. Odpovídají programovým tlačítkům ve svislém pruhu na obrazovce cyklu.

- E: Odlehčovací zápich:tvar E
- F: Odlehčovací zápich:tvar F

- A-D: Závitový zápich podle DIN (tvary A - D)
- T: Závitový zápich (volná definice tvaru)

Syntaxe

```
CYCLE940 (<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>,
<_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>,
<_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Progr. u tvaru				Význam		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	Souřadnice vztažného bodu v příčné ose (vždycky průměr)		
2	Z0	<_SPL>	REAL	x	x	x	x	Vztažný bod, podélná osa (abs)		
3	FORM	<_FORM>	CHAR	x	x	x	x	Tvar zápichu (velká písmena, např. "T") Volba, ze které tabulky se mají brát hodnoty parametrů zápichu		
								A =	vnější, podle DIN 76, A = normální	
								B =	vnější, podle DIN 76, B = krátký	
								C =	vnitřní, podle DIN 76, C = normální	
								D =	vnitřní, podle DIN 76, D = krátký	
								E =	Podle DIN 509	
								F =	Podle DIN 509	
								T =	volný tvar	
4	LAGE	<_LAGE>	INT	x	x	x	x	Poloha zápichu (rovnoběžně s osou Z)		
								0 =	Vnější +Z: ____	
								1 =	Vnější -Z: ____/	
								2 =	Vnitřní +Z: /-----	
								3 =	Vnitřní-Z: -----\	
5	SC	<_SDIS>		x	x	x	x	Bezpečnostní vzdálenost (ink)		
6	F	<_FFP>		x	x	x	x	Pracovní posuv (mm/ot)		
7		<_VARI>	INT	-	-	x	x	Způsob opracování		
								JEDNOTKY:	Opracování	
									1 =	Obrábění nahrubo
									2 =	Obrábění načisto
									3 =	Obrábění nahrubo + načisto
								DESÍTKY:	Strategie obrábění	
									0 =	rovnoběžně s konturou
									1 =	podélný
Zápichy tvarů E a F jsou vždy opracovávány v jednom tahu jako při obrábění načisto										

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Progr. u tvaru				Význam			
8	X1	<_EPD>		x	x	-	-	Přídavek rozměru pro opracování načisto ve směru X (abs/ink), viz <_AMODE>			
				-	-	-	x	Hloubka zápichu (abs/ink), viz <_AMODE>			
9	Z1	<_EPL>		-	x	-	-	Přídavek rozměru pro opracování načisto ve směru Z			
				-	-	-	x	Šířka zápichu (abs/ink), viz <_AMODE>			
10	R1	<_R1>		-	-	-	x	Rádus zaoblení na stěnách			
11	R2	<_R2>		-	-	-	x	Rádus zaoblení na rozích			
12	α	<_STA>		-	-	x	x	Úhel zajiždění do obrobku			
13	VX	<_VRT>		x	x	-	-	Příčný pohyb ve směru X (abs/ink), viz <_AMODE>			
				-	-	x	x	Příčný pohyb ve směru X při obrábění načisto (abs/ink), viz <_AMODE>			
14	D	<_MID>		-	-	x	x	Přísuv do hloubky			
15	U	<_FAL>		-	-	x	x	Přídavek rozměru pro obrábění načisto rovnoběžně s konturou, viz<_AMODE>			
16	UX	<_FALX>		-	-	x	x	Přídavek rozměru pro obrábění načisto ve směru X			
17	UZ	<_FALZ>		-	-	x	x	Přídavek rozměru pro obrábění načisto ve směru Z			
18	P	<_PITI>	INT	-	-	x	-	Volba stoupání, tvar A - D, odpovídá M1 ... M68			
								0 = 0.20 1 = 0.25 2 = 0.30 3 = 0.35 4 = 0.40 5 = 0.45	6 = 0.50 7 = 0.60 8 = 0.70 9 = 0.75 10 = 0.80 11 = 1.00	12 = 1.25 13 = 1.50 14 = 1.75 15 = 2.00 16 = 2.50 17 = 3.00	18 = 3.50 19 = 4.00 20 = 4.50 21 = 5.00 22 = 5.50 23 = 6.00
				x	x	-	-	Volba rádiusu/hloubky, tvar E, F			
								0 = 0.6 · 0.3 1 = 1.0 · 0.4 2 = 1.0 · 0.2 3 = 1.6 · 0.3	4 = 2.5 · 0.4 5 = 4.0 · 0.5 6 = 0.4 · 0.2 7 = 0.6 · 0.2	8 = 0.1 · 0.1 9 = 0.2 · 0.1	
19		<_PTAB>	STRING [5]					Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC") (jen pro pracovní plochu)			
20		<_PTABA>	STRING [20]					Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) (pouze pro uživatelské rozhraní)			
21		<_DMODE>	INT					Režim zobrazování			
				x	x	x	x	JEDNOTKY:	Rovina obrábění G17/G18/G19		
								0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu		
								1 =	G17 (aktivní jen v cyklu)		
								2 =	G18 (aktivní jen v cyklu)		
				3 =	G19 (aktivní jen v cyklu)						

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Progr. u tvaru	Význam
22		<_AMODE>	INT		Alternativní režim
				x x - x	JEDNOTKY: Parametr <_EPD>, přídavek rozměru ve směru X nebo hloubka odlehčovacího zápisu
					0 = absolutně (vždy průměr)
					1 = inkrementální
				x x - x	DESÍTKY: Parametr <_EPL>, přídavek rozměru ve směru Z nebo šířka odlehčovacího zápisu
					0 = absolutní
					1 = inkrementální
				x x x x	STOVKY: Parametr <_VRT>, příčný pohyb ve směru X
					0 = absolutně (vždy průměr)
					1 = inkrementální
				- - x x	TISÍCE: Přídavek rozměru pro opracování načisto
					0 = Přídavek rozměru pro obrábění načisto rovnoběžně s konturou (<_FAL>)
					1 = Přídavek rozměru pro opracování načisto odděleně (<_FALX>/<_FALZ>)

20.1.42 CYCLE951 - Oddělování třísky

Syntaxe

CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>, <_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>, <_FF1>, <_NR>, <_DMODE>, <_AMODE>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	X0	<_SPD>	REAL	Souřadnice vztažného bodu, (abs, vždycky průměr)
2	Z0	<_SPL>	REAL	Vztažný bod (abs)
3	X1	<_EPD>	REAL	Koncový bod
4	Z1	<_EPL>	REAL	Koncový bod
5	XM α1 α2	<_ZPD>	REAL	Vnitřní bod, viz <_DMODE> (DESÍTKY)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
6	ZM α1 α2	<_ZPL>	REAL	Vnitřní bod, viz <_DMODE> (DESÍTKY)		
7	Poloha	<_LAGE>	INT	Poloha pro obráběného rohu	0 =	vnější/vzadu
					1 =	vnější/vpředu
					2 =	vnitřní/vzadu
					3 =	vnitřní/vpředu
8	D	<_MID>	REAL	Maximální přísuv do hloubky při zajištění nástroje		
9	UX	<_FALX>	REAL	Přídavek rozměru pro opracování načisto ve směru X		
10	UZ	<_FALZ>	REAL	Přídavek rozměru pro opracování načisto ve směru Z		
11		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Směr oddělování třísky (podélné nebo příčné) v souřadném systému	
					1 =	podélný
					2 =	příčný
				DESÍTKY:		
					1 =	Obrábění nahrubo až na přídavek rozměru pro obrábění načisto
					2 =	Obrábění načisto
				STOVKY:	rezervováno	
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	rezervováno	
12	R1/FS1	<_RF1>	REAL	Rádus zaoblení nebo šířka fasety 1, viz <_AMODE> (DESÍTKY TISÍC)		
13	R2/FS2	<_RF2>	REAL	Rádus zaoblení nebo šířka fasety 2, viz <_AMODE> (STOVKY TISÍC)		
14	R3/FS3	<_RF3>	REAL	Rádus zaoblení nebo šířka fasety 3, viz <_AMODE> (JEDNOTKY MILIONŮ)		
15	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost		
16	F	<_FF1>	REAL	Posuv pro obrábění nahrubo/načisto		
17		<_NR>	INT	Identifikace způsobu oddělování třísky (odpovídá programovým tlačítkům ve svislém pruhu na obrazovce)		
					0 =	oddělování třísky 1, roh 90 stupňů bez faset/zaoblení
					1 =	oddělování třísky 2, roh 90 stupňů s fasetami/zaobleními
					2 =	oddělování třísky 3, libovolný roh s fasetami/zaobleními

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
18		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY:
				Způsob zadávání parametrů <_ZPD>/<_ZPL>
				0 = Xm/Zm
				1 = Xm/ α 1
				2 = Xm/ α 2
				3 = α 1/Zm
				4 = α 2/Zm
				5 = α 1/ α 2
21		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Vnitřní bod, souřadnice X
				0 = absolutně, hodnota příčné osy se zadává jako průměr
				1 = inkrementálně, hodnota příčné osy se zadává jako rádius
				DESÍTKY:
				Vnitřní bod, souřadnice Z
				0 = absolutní
				1 = inkrementální
				STOVKY:
				Koncový bod, souřadnice X
				0 = absolutně, hodnota příčné osy se zadává jako průměr
				1 = inkrementálně, hodnota příčné osy se zadává jako rádius
				TISÍCE:
				koncový bod v ose Z
				0 = absolutní
				1 = inkrementální
				DESÍTKY TISÍC:
				Rádius / faseta 1
				0 = Rádius
				1 = Faseta
				STOVKY TISÍC:
				Rádius / faseta 2
				0 = Rádius
				1 = Faseta
				JEDNOTKY MILIONŮ:
				Rádius / faseta 3
				0 = Rádius
				1 = Faseta

20.1.43 CYCLE952 - Zápichy na kontuře

Syntaxe

```
CYCLE952 (<_PRG>, <_CON>, <_CONR>, <_VARI>, <_F>, <_FR>, <_RP>, <_D>,
<_DX>, <_DZ>, <_UX>, <_UZ>, <_U>, <_U1>, <_BL>, <_XD>, <_ZD>, <_XA>,
<_ZA>, <_XB>, <_ZB>, <_XDA>, <_XDB>, <_N>, <_DP>, <_DI>, <_SC>,
<_DN>, <_GMODE>, <_DMODE>, <_AMODE>, <_PK>, <_DCH>, <_FS>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	PRG	<_PRG>	STRING[100]	Název programu pro oddělování třísky
2	CON	<_CON>	STRING[100]	Název programu, z něhož se načítá aktualizovaná kontura surového obrobku (při obrábění zbytkového materiálu)
3	CONR	<_CONR>	STRING[100]	Název programu, ve kterém je popsána aktualizovaná kontura surového obrobku (viz <_AMODE>, DESÍTKY TISÍC)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
4		<_VARI>	INT	Způsob opracování
				JEDNOTKY:
				Způsob opracování
				1 = podélný
				2 = příčný
				3 = rovnoběžně s konturou
				DESÍTKY:
				Technologie opracování (viz parametr <_GMODE> (STOVKY))
				1 = Obrábění nahrubo
				2 = Obrábění načisto
				3 = rezervováno
				4 = Obrábění nahrubo ve dvou kanaálech
				5 = Obrábění načisto ve dvou kanálech
				STOVKY:
				Směr obrábění
				1 = Směr obrábění X -
				2 = Směr obrábění X +
				3 = Směr obrábění Z-
				4 = směr obrábění Z +
				TISÍCE:
				Směr přísuvu
				1 = Zvnějšku X -
				2 = Zevnitř X +
				3 = Čelní plocha Z -
				4 = Zadní strana Z +
				DESÍTKY TISÍC:
				Definice, platí jako přídavek rozměru pro opracování načisto
				0 = Přídavek rozměru pro opracování načisto UX a UZ odděleně
				1 = Přídavek rozměru pro opracování načisto U rovnoběžně s konturou
				STOVKY TISÍC:
				Vyjíždění podél kontury
				0 = kompatibilita, automatické vyjíždění podél kontury
				1 = s vyjížděním podél kontury
				2 = bez vyjíždění podél kontury
				3 = automatické vyjíždění podél kontury
				JEDNOTKY MILIONŮ:
				Podříznutí
				0 = místo se nevyhodnocuje při zapichování, obrábění zbytkového materiálu a soustružení zápichů, při obrábění zbytkového materiálu
				1 = Podříznutí opracovat

Č.	Obrazovka parametru	Interní parametry	Datový typ	Význam		
					2 =	podříznutí neopracovat
				DESÍTKY MILIONŮ:	Před/za osou otáčení	
					0 =	obrábění před středem otáčení
					1 =	rezervováno
5	F	<_F>	REAL	Posuv pro obrábění nahrubo/načisto		
	FZ			Posuv ve směru abscisy při soustružení zápichů		
6	FR	<_FR>	REAL	Posuv pro zajištění nástroje při obrábění podříznutí nahrubo		
	FX			Posuv ve směru ordináty při soustružení zápichů		
7	RP	<_RP>	REAL	Návratová rovina při vnitřním obrábění (abs, vždycky průměr)		
8	D	<_D>	REAL	Přisuv při obrábění nahrubo, (viz JEDNOTKY parametru <_AMODE>)		
9	DX	<_DX>	REAL	Přisuv ve směru X (viz JEDNOTKY parametru <_AMODE>)		
10	DZ	<_DZ>	REAL	Přisuv ve směru Z (viz JEDNOTKY parametru <_AMODE>)		
11	UX	<_UX>	REAL	Přídavek rozměru pro opracování načisto ve směru X (viz <_VARI> - DESÍTKY TISÍC)		
12	UZ	<_UZ>	REAL	Přídavek rozměru pro opracování načisto ve směru Z (viz <_VARI> - DESÍTKY TISÍC)		
13	U	<_U>	REAL	Přídavek rozměru pro obrábění načisto rovnoběžně s konturou (viz <_VARI> - DESÍTKY TISÍC)		
14	U1	<_U1>	REAL	Dodatečný přídavek rozměru pro opracování načisto (viz <_AMODE> - TISÍCE)		
15	BL	<_BL>	INT	Definice surového obrobku	1 =	válec s přídavkem rozměru
					2 =	přídavek rozměru na kontuře hotového obrobku
					3 =	kontura surového obrobku je zadána
16	XD	<_XD>	REAL	Definice surového obrobku ve směru X (viz <_AMODE> - STOVKY TISÍC)		
17	ZD	<_ZD>	REAL	Definice surového obrobku ve směru Z (viz <_AMODE> (MILIONY))		
18	XA	<_XA>	REAL	Hranice 1 ve směru X (abs, vždycky průměr)		
19	ZA	<_ZA>	REAL	Hranice 1 ve směru Z (abs)		
20	XB	<_XB>	REAL	Hranice 2 ve směru X (viz <_AMODE> - DESÍTKY MILIONŮ)		
21	ZB	<_ZB>	REAL	Hranice 2 ve směru Z (viz <_AMODE> - STOVKY MILIONŮ)		
22	XDA	<_XDA>	REAL	Hranice zápichu 1 pro pozici 1. zápichu na čelní straně (abs, vždy průměr)		
23	XDB	<_XDB>	REAL	Hranice zápichu 2 pro pozici 1. zápichu na čelní straně (abs, vždy průměr)		
24	N	<_N>	INT	Počet zápichů		
25	DP	<_DP>	REAL	Vzdálenost mezi zápichy	Podélný zápich: rovnoběžně s osou Z	
					Příčný zápich: rovnoběžně s osou X	
26	DI	<_DI>	REAL	Vzdálenost pro přerušení posuvu	0 =	Žádné přerušení
					> 0 =	s přerušením
27	SC	<_SC>	REAL	Bezpečnostní vzdálenost pro objíždění překážky, inkrementálně		
28	D2	<_DN>	INT	D-číslo pro 2. břit nástroje, pokud není naprogramováno ⇒ D+1		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
29		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)
				JEDNOTKY: re-zervo-váno
				DESÍTKY: re-zervo-váno
				STOVKY: Volba opracování/pouze výpočet počátečního bodu
				0 = normální opracování (není zapotřebí žádný režim kompatibility)
				1 = normální opracování
				2 = výpočet počáteční polohy - žádné obrábění (jen pro vyvolání ze systémů ShopMill/ShopTurn)
				TISÍCE: Ohraničení
				0 = ne
				1 = ano
				DESÍTKY TISÍC: Zadat hranici 1 ve směru X
				0 = ne
				1 = ano
				STOVKY TISÍC: Zadat hranici 2 ve směru X
				0 = ne
				1 = ano
				JEDNOTKY MILIONŮ: Zadat hranici 1 ve směru Z
				0 = ne
				1 = ano
				DESÍTKY MILIONŮ: Zadat hranici 2 ve směru Z
				0 = ne
				1 = ano

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
30		<_DMODE>	INT	Režim zobrazování
				JEDNOTKY:
				Rovina obrábění G17/G18/G19
				0 = kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
				1 = G17 (aktivní jen v cyklu)
				2 = G18 (aktivní jen v cyklu)
				3 = G19 (aktivní jen v cyklu)
				DESÍTKY:
				technologický režim
				1 = Obrábění kontury
				2 = Zápichy na kontuře
				3 = Soustružení zápichů
				STOVKY:
				Odstraňování zbytkového materiálu
				0 = ne
				1 = ano
				TISÍCE:
				--- rezervováno
				DESÍTKY TISÍC:
				Odstupňování technologie na obrazovkách cyklů (Strana 842)
				0 = Zadání: kompletní
				1 = Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
31		<_AMODE>	INT	Alternativní režim
				JEDNOTKY:
				Volba přísuvu
				0 = přísuv DX a DZ při druhu obrábění rovnoběžně s konturou
				1 = Přísuv D
				DESÍTKY:
				Strategie pro přísuv
				0 = proměnná hloubka řezu (90 ... 100%)
				1 = konstantní hloubka průchodu nástroje
				STOVKY:
				Rozdělení řezů
				0 = rovnoměrné
				1 = optimalizace na hranách
				TISÍCE:
				Volba přidavku rozměru na kontuře U1, dvojí obrábění načisto
				0 = ne
				1 = ano
				DESÍTKY TISÍC:
				Aktualizace volby surového obrobku
				0 = ne
				1 = ano
				STOVKY TISÍC:
				Volba přidavku rozměru na surovém obrobku XD
				0 = absolutně, hodnota příčné osy se zadává jako průměr
				1 = inkrementálně, hodnota příčné osy se zadává jako rádius
				JEDNOTKY MILIONŮ:
				Volba přidavku rozměru na surovém obrobku ZD
				0 = absolutní
				1 = inkrementální
				DESÍTKY MILIONŮ:
				Volba hranice 2 XB
				0 = absolutně, hodnota příčné osy se zadává jako průměr
				1 = inkrementálně, hodnota příčné osy se zadává jako rádius
				STOVKY MILIONŮ:
				Volba hranice 2 ZB
				0 = absolutní
				1 = inkrementální
				JEDNOTKY MILIARD:
				0 = Řídicí kanál
				1 = Vlečný kanál
32		<_PK>	INT	Číslo partnerského kanálu, jestliže jsou na stroji k dispozici více než 2 kanály

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
33	DCH	<_DCH>	REAL	Kanálové posunutí
34	FS	<_FS>	REAL	Posuv při obrábění načisto v rámci kompletního opracování

20.1.44 CYCLE4071 - Podélné broušení s přísuvem v bodu obratu

Syntaxe

CYCLE4071 (<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>, <S_A2>)

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_A>	REAL	Přisuvná hloubka na začátku
2	<S_B>	REAL	Přisuvná hloubka na konci
3	<S_W>	REAL	Broušená šířka
4	<S_U>	REAL	Doba vyjiskřování
5	<S_I>	REAL	Posuv pro přísuv
6	<S_K>	REAL	Posuv pro přísuv v příčném směru
7	<S_H>	INT	Počet opakování
8	<S_A1>	AXIS	Přisuvná osa (nepovinné), příp. 1. geometrická osa
9	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

Funkce

Tento cyklus slouží pro zpracování <S_H> přísuvů. Přisuvná hloubka na začátku a na konci se přitom mohou lišit. Mezi přísuvy se uskutečňuje tangenciální pohyb.

Postup

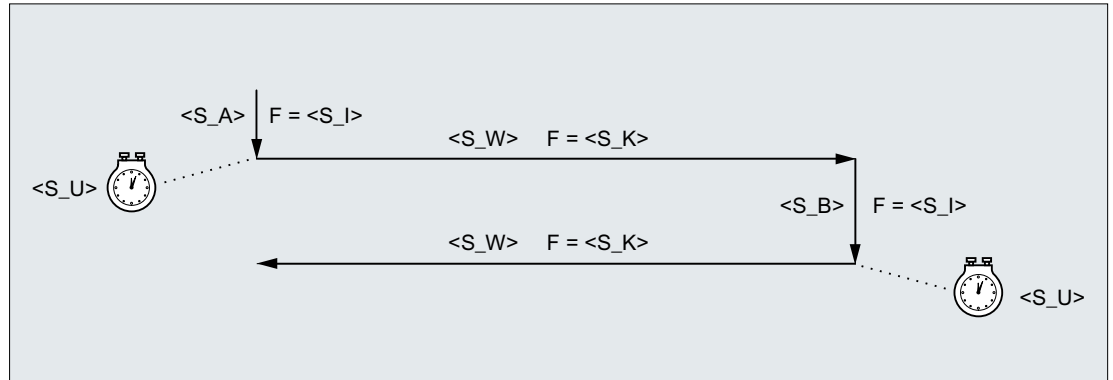
1. Spuštění cyklu na momentální pozici oscilační osy.
2. Pohyb přisuvné osy na přisuvnou hloubku na začátku P1 <S_A> s posuvem pro přísuv P5 <S_I>.
3. Vyjiskřování s dobou vyjiskřování P4 <S_U>.
4. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P3 <S_W> s posuvem pro přísuv v příčném směru P6 <S_K>.
5. Pohyb přisuvné osy na přisuvnou hloubku na konci P2 <S_B> s posuvem pro přísuv P5 <S_I>.

6. Vyjiskřování s dobou vyjiskřování P4 <S_U>.

7. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_A> na počáteční bod a s posuvem pro přísuv v příčném směru P6 <S_K>.

Tento postup při zpracování blok po bloku není možné přerušit.

Tato operace se opakuje v souladu s naprogramovaným počtem opakování P7 (<S_H>).



Příklad

Uskutečňte dva oscilační pohyby s následujícími parametry cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv pro přísuv: 1 mm/min
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Programový kód

```
N10 T1 D1
N20 CYCLE4071 (0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.45

CYCLE4072 - Podélné broušení s přísuvem v bodě obratu a se signálem pro přerušení

Syntaxe

```
CYCLE4072 (<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_GAUGE>	STRING	Podmínky přerušení pro přísuv: 1. Číslo rychlého vstupu 2. Logický výraz
2	<S_A>	REAL	Přisuvná hloubka na začátku
3	<S_B>	REAL	Přisuvná hloubka na konci
4	<S_W>	REAL	Broušená šířka
5	<S_U>	REAL	Doba vyjiskřování
6	<S_I>	REAL	Posuv pro přísuv
7	<S_K>	REAL	Posuv pro přísuv v příčném směru
8	<S_H>	INT	Počet opakování
9	<S_A1>	AXIS	Přisuvná osa (nepovinné), příp. 1. geometrická osa
10	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

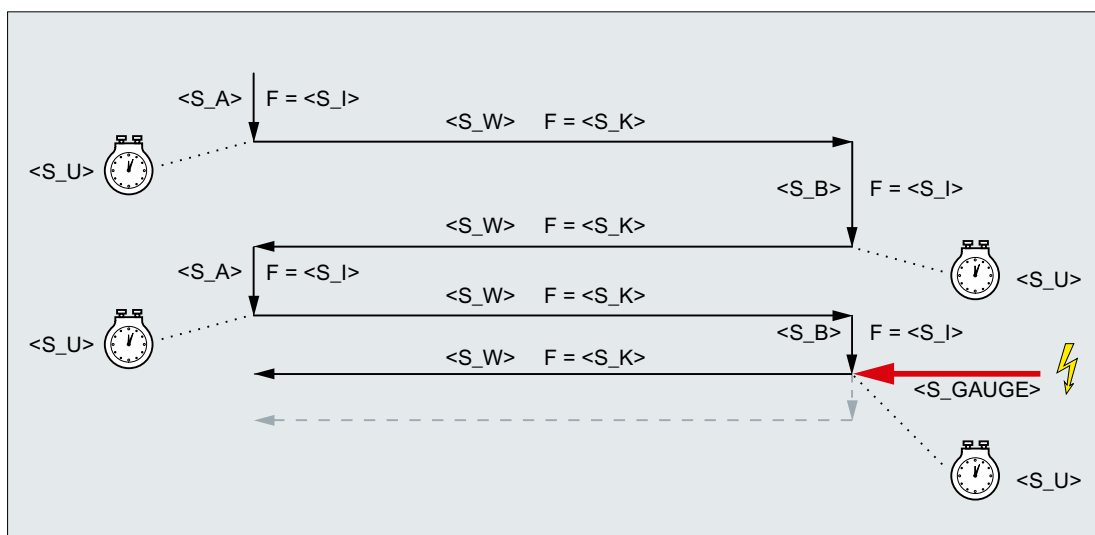
Funkce

Cyklus slouží pro zpracování <S_H> přísuvů, přičemž se vyhodnocuje externí přerušovací signál. Přisuvné hloubky na začátku a na konci se mohou lišit. Mezi přísuvy se uskutečňuje tangenciální pohyb. Když je podmínka pro přerušení splněna, přísuv do hloubky se přeruší. Po přerušení přísuvu do hloubky se vždy uskuteční jeden kompletní zdvih nástroje.

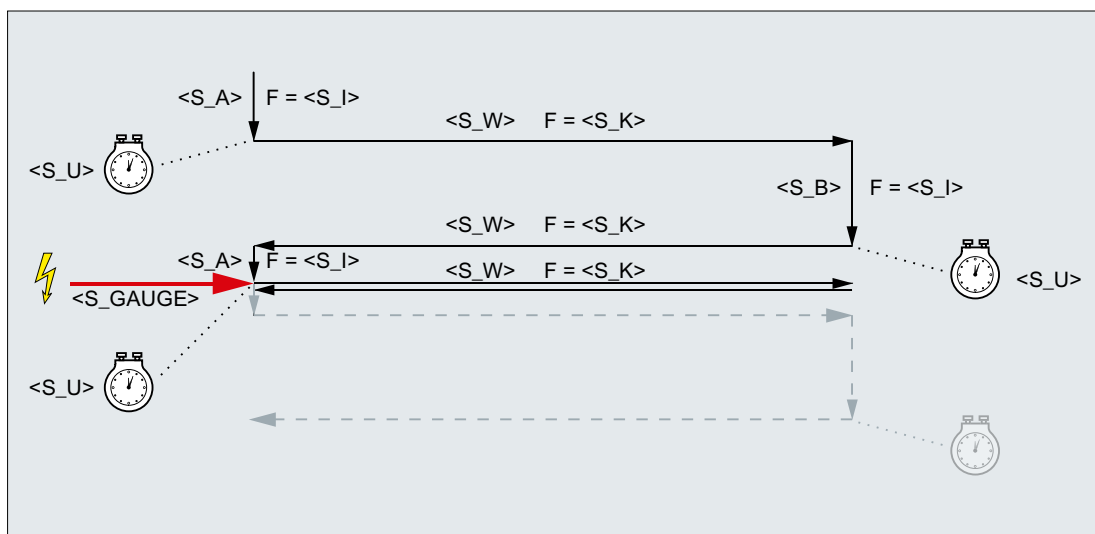
Postup

1. Spuštění cyklu na momentální pozici oscilační osy.
2. Pohyb přísuvné osy na přísuvnou hloubku na začátku P2 <S_A> s posuvem pro přísuv P6 <S_I>.
3. Vyjiskřování s dobou vyjiskřování P5 <S_U>.
4. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_W> s posuvem pro přísuv v příčném směru P7 <S_K>.
5. Pohyb přísuvné osy na přísuvnou hloubku na konci P3 <S_B> s posuvem pro přísuv P6 <S_I>.
6. Vyjiskřování s dobou vyjiskřování P5 <S_U>.
7. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_W> na počáteční bod a s posuvem pro přísuv v příčném směru P7 <S_K>.
8. Bez přerušení: Výše popisovaná operace se opakuje tak dlouho, dokud není dosaženo naprogramovaného počtu opakování P7 (<S_H>).
S přerušením: Po dosažení následujícího počátečního bodu se opracování ukončí.

Tento postup prostřednictvím zpracování blok po bloku není možné přerušit.



Obrázek 20-1 Přerušení přísuvu na konci



Obrázek 20-2 Přerušení přísuvu na začátku

Zdroje

Jako zdroje cyklus používá synchronní akce překračující hranice bloků a proměnnou synchronní akce. Synchronní akce se dynamicky určuje z volné oblasti ve svazku synchronních akcí (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR - 1199 ...). Jako proměnná synchronní akce se používá SYG_IS[1].

Příklady

Příklad 1: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv pro přísuv: 1 mm/min
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Přerušovací signál: rychlý vstup 1 (\$A_IN[1])

Programový kód

```
N10 T1 D1
N20 CYCLE4072 ("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

Příklad 2: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv pro přísuv: 1 mm/min
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Přerušovací signál: Proměnná \$A_DBR[20] < 0,01

Programový kód

```
N10 T1 D1
N20 CYCLE4072 ("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.46 CYCLE4073 - Podélné broušení se spojitým přísuvem

Syntaxe

CYCLE4073 (<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_A>	REAL	Přísluvná hloubka na začátku
2	<S_B>	REAL	Přísluvná hloubka na konci
3	<S_W>	REAL	Broušená šířka
4	<S_U>	REAL	Doba vyjiskřování
5	<S_K>	REAL	Posuv pro přísluv v příčném směru
6	<S_H>	INT	Počet opakování
7	<S_A1>	AXIS	Přísluvná osa (nepovinné), příp. 1. geometrická osa
8	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

Funkce

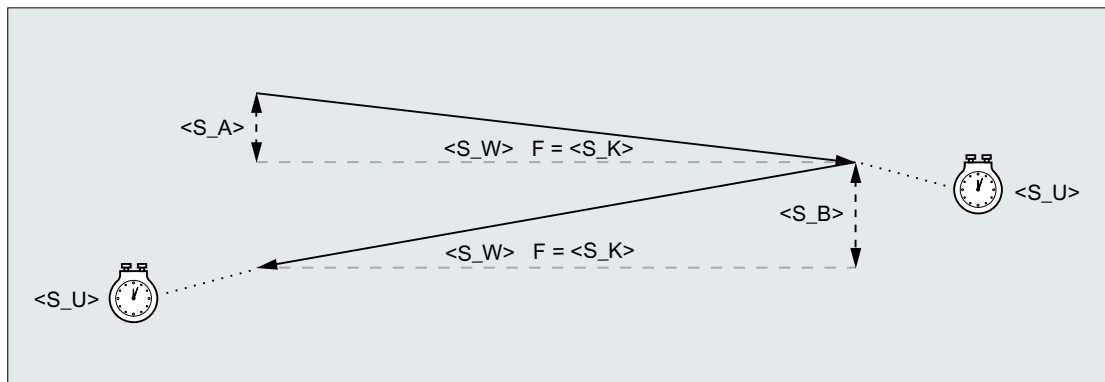
Tento cyklus slouží pro zpracování <S_H> přísluvů. Přísluv od začátku ke konci a od konce na začátek se mohou lišit.

Postup

1. Spuštění cyklu na momentální pozici oscilační osy s přísluvnou hloubkou 0
2. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P3 <S_W> a s posuvem pro přísluv v příčném směru P5 <S_K> se spojitým zvětšováním přísluvné hloubky až po přísluvnou hloubku na začátku P1 <S_A>.
3. Vyjiskřování s dobou vyjiskřování P4 <S_U>.
4. Pohyb oscilační osy na počáteční bod s dráhou posuvu odpovídající broušené šířce P3 <S_W> a s posuvem pro přísluv v příčném směru P5 <S_K> se spojitým zvětšováním přísluvné hloubky až po přísluvnou hloubku na konci P2 <S_B>.
5. Vyjiskřování s dobou vyjiskřování P4 <S_U>.

Tento postup prostřednictvím zpracování blok po bloku není možné přerušit.

Tato operace se opakuje v souladu s naprogramovaným počtem opakování P7 (<S_H>).



Příklad

Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Programový kód

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.47 CYCLE4074 - Podélné broušení se spojitým přísuvem a se signálem pro přerušení

Syntaxe

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```


Parametry

Č.	Parametry	Datový typ	Význam
1	<S_GAUGE>	STRING	Podmínky přerušení pro přísvuv: 1. Číslo rychlého vstupu 2. Logický výraz
2	<S_A>	REAL	Přísvuvná hloubka na začátku
3	<S_B>	REAL	Přísvuvná hloubka na konci
4	<S_W>	REAL	Broušená šířka
5	<S_U>	REAL	Doba vyjiskřování
6	<S_K>	REAL	Posuv pro přísvuv v příčném směru
7	<S_H>	INT	Počet opakování
8	<S_A1>	AXIS	Přísvuvná osa (nepovinné), příp. 1. geometrická osa
9	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

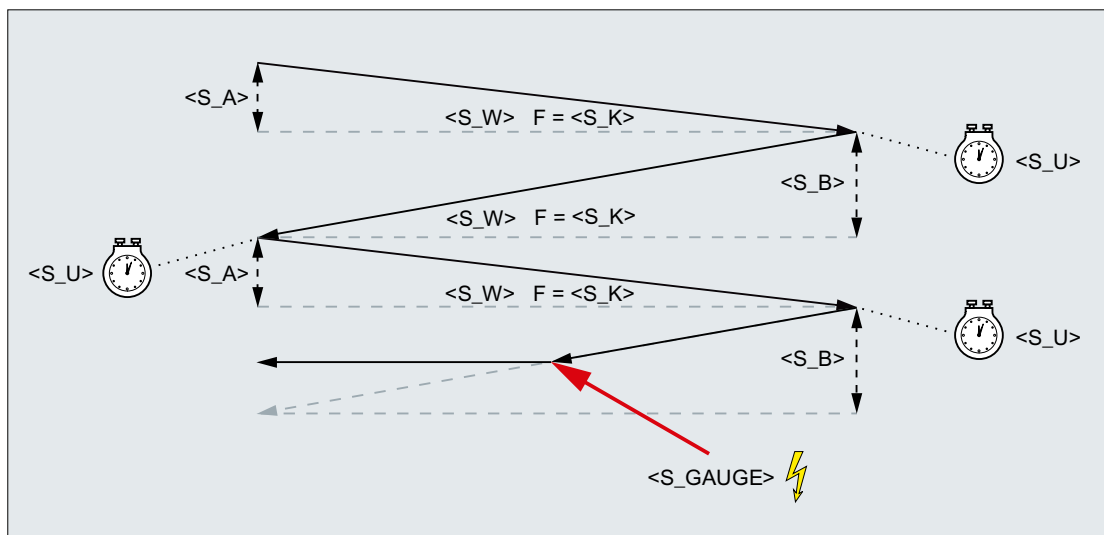
Funkce

Cyklus slouží pro zpracování <S_H> přísvuvů, přičemž se vyhodnocuje např. externí přerušovací signál. Přísvuvné hloubky na začátku a na konci se mohou lišit. Když je podmínka pro přerušení splněna, přísvuv do hloubky se přeruší. Po přerušení přísvuvu do hloubky se vždy uskuteční jeden kompletní zdvih nástroje.

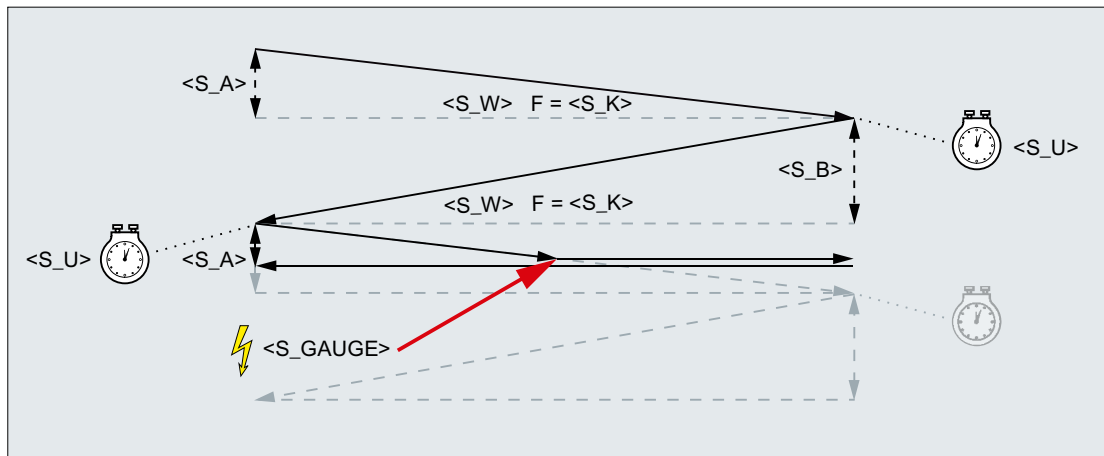
Postup

1. Spuštění cyklu na momentální pozici oscilační osy s přísvuvnou hloubkou 0
2. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_W> a s posuvem pro přísvuv v příčném směru P6 <S_K> se spojitým zvětšováním přísvuvné hloubky až po přísvuvnou hloubku na začátku P2 <S_A>.
3. Vyjiskřování s dobou vyjiskřování P5 <S_U>.
4. Pohyb oscilační osy na počáteční bod s dráhou posuvu odpovídající broušené šířce P4 <S_W> a s posuvem pro přísvuv v příčném směru P6 <S_K> se spojitým zvětšováním přísvuvné hloubky až po přísvuvnou hloubku na konci P3 <S_B>.
5. Vyjiskřování s dobou vyjiskřování P5 <S_U>.
6. Bez přerušení: Výše popisovaná operace se opakuje tak dlouho, dokud není dosaženo naprogramovaného počtu opakování P7 (<S_H>).
S přerušením: Přísvuv do hloubky se přeruší. Po dosažení následujícího počátečního bodu se opracování ukončí.

Tento postup prostřednictvím zpracování blok po bloku není možné přerušit.



Obrázek 20-3 Přerušení přísuvu od konce k začátku



Obrázek 20-4 Přerušení přísuvu od začátku ke konci

Zdroje

Jako zdroje cyklus používá synchronní akci překračující hranice bloků a proměnnou synchronní akce. Synchronní akce se dynamicky určuje z volné oblasti ve svazku synchronních akcí (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR - 1199 ...). Jako proměnná synchronní akce se používá SYG_IS[1].

Příklady

Příklad 1: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přisuvná hloubka na začátku: 0,02 mm
- Přisuvná hloubka na konci: 0,01 mm

- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přírusná osa: Standardní geometrické osy

Přerušovací signál: rychlý vstup 1 (\$A_IN[1])

Programový kód

```
N10 T1 D1
N20 CYCLE4074 ("1",0.02,0.01,100,1,1000,2)
N30 M30
```

Příklad 2: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přírusná hloubka na začátku: 0,02 mm
- Přírusná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přírusná osa: Standardní geometrické osy

Přerušovací signál: Proměnná \$A_DBR[20] < 0,01

Programový kód

```
N10 T1 D1
N20 CYCLE4074 ("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.48 CYCLE4075 - Plošné broušení s přírusvem v bodě obratu

Syntaxe

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_I>	REAL	Přírusná hloubka na začátku
2	<S_J>	REAL	Přírusná hloubka na konci

Č.	Parametry	Datový typ	Význam
3	<S_K>	REAL	Celková příravná hloubka
4	<S_A>	REAL	Broušená šířka
5	<S_R>	REAL	Posuv pro příravn
6	<S_F>	REAL	Posuv pro příravn v příčném směru
7	<S_P>	REAL	Doba vyjiskřování
8	<S_A1>	AXIS	Příravná osa (nepovinné)
9	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

Tento cyklus slouží pro opracování s celkovou hloubkou příravnu P3 <S_K> v několika příravných krocích. Příravné hloubky na začátku P1 <S_I> a na konci P2 <S_J> se mohou lišit. Mezi příravy se uskutečňuje tangenciální pohyb.

Údaje dráhy P1 až P4 mohou být kladné nebo záporné.

Zadání příravné osy P8 <S_A1> a/nebo oscilační osy P9 <S_A2> jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

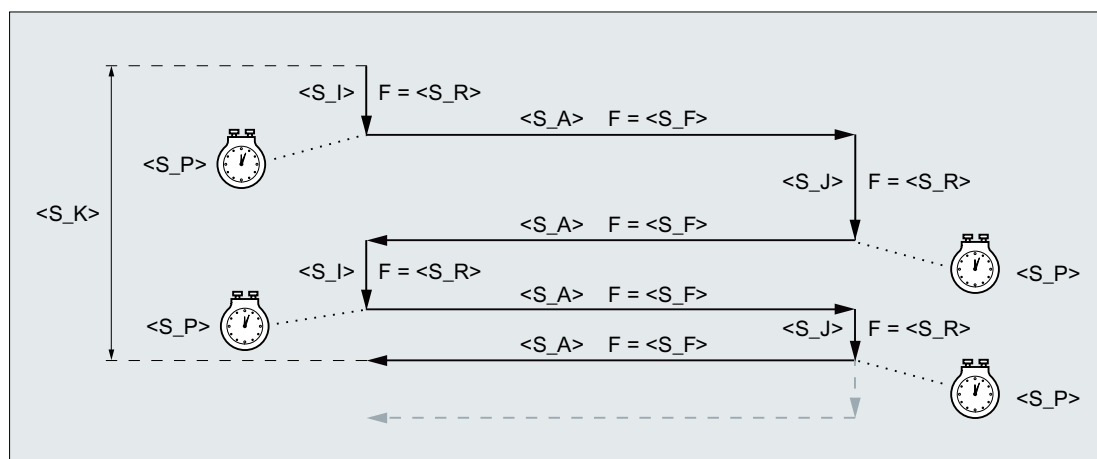
Pokud je součet příravné hloubky na začátku P1 <S_I> a na konci P2 <S_J> roven 0, příp. pokud se celková příravná hloubka P3 <S_K> rovná nule, uskuteční se pouze jeden vyjiskřovací zdvih.

Postup

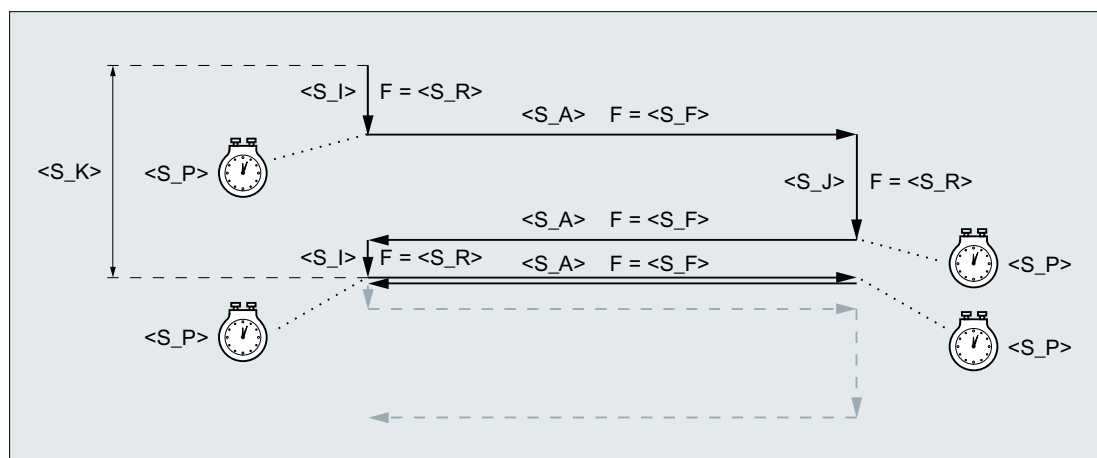
1. Spuštění cyklu na momentální pozici oscilační osy.
2. Pohyb příravné osy na příravnou hloubku na začátku P1 <S_I> s posuvem pro příravn P5 <S_R>.
3. Vyjiskřování s dobou vyjiskřování P7 <S_P>.
4. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_A> s posuvem pro příravn v příčném směru P6 <S_F>.
5. Pohyb příravné osy na příravnou hloubku na konci P2 <S_J> s posuvem pro příravn P5 <S_R>.
6. Vyjiskřování s dobou vyjiskřování P7 <S_P>.
7. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_A> na počáteční bod a s posuvem pro příravn v příčném směru P6 <S_F>.

Tento postup při zpracování blok po bloku není možné přerušit.

Tato operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky příravnu P3 <S_K>. Poslední zdvih je potom rovnoměrně rozdělen.



Obrázek 20-5 Celková příravná hloubka dosažená ve druhém bodu obratu



Obrázek 20-6 Celková příravná hloubka dosažená v prvním bodu obratu

Příklad

Oscilační pohyb s těmito parametry:

- Příravná hloubka na začátku 0,02 mm
- Příravná hloubka na konci 0,01 mm
- Celková příravná hloubka 1 mm
- Zdvih 100 mm
- Posuv při příravnosti 1 mm/min
- Příravnost v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Programový kód

N10 T1 D1

Programový kód

```
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.49 CYCLE4077 - Plošné broušení s přísvem na bod obratu a se signálem pro přerušení

Syntaxe

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_GAUGE>	STRING	Podmínka pro přerušení přísvu: - Číslo rychlého vstupu - Logický výraz
2	<S_I>	REAL	Přísvná hloubka na začátku
3	<S_J>	REAL	Přísvná hloubka na konci
4	<S_K>	REAL	Celková přísvná hloubka
5	<S_A>	REAL	Broušená šířka
6	<S_R>	REAL	Posuv pro přísv
7	<S_F>	REAL	Posuv pro přísv v příčném směru
8	<S_P>	REAL	Doba vyjiskřování
9	<S_A1>	AXIS	Přísvná osa (nepovinné)
10	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

Tento cyklus slouží pro opracování s celkovou hloubkou přísvu P4 <S_K> v několika přísvných krocích. Přísvné hloubky na začátku P2 <S_I> a na konci P3 <S_J> se mohou lišit. Mezi přísvy se uskutečňuje tangenciální pohyb. Když se přerušovací signál na rychlém vstupu rovná 1 nebo je-li splněna podmínka pro přerušení, přísv do hloubky se přeruší. Po přerušení se uskuteční jeden kompletní zdvih nástroje.

Údaje dráhy P2 až P5 mohou být kladné nebo záporné.

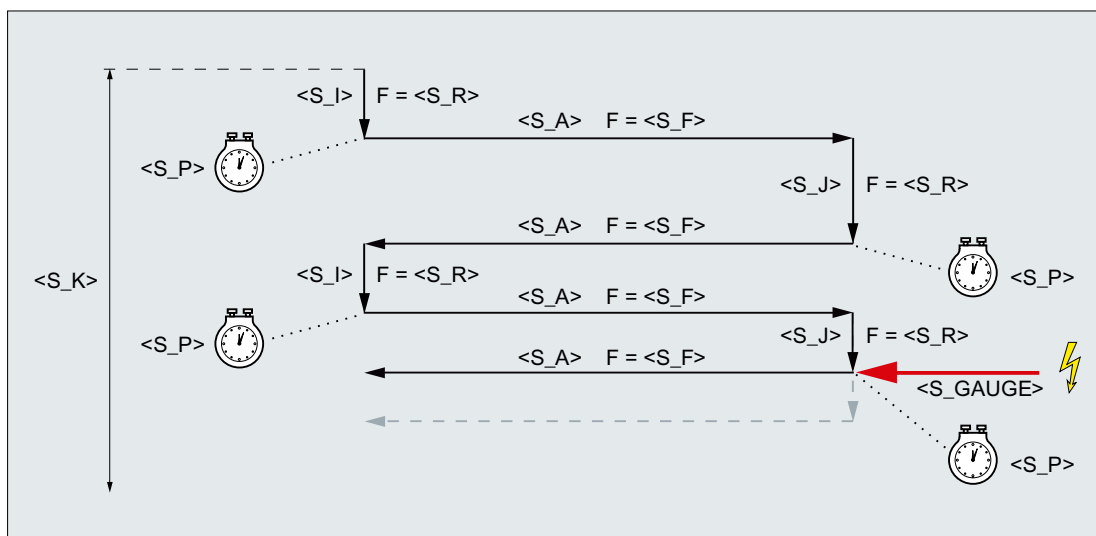
Zadání přísvné osy P9 <S_A1> a/nebo oscilační osy P10 <S_A2> jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

Pokud je součet přísvné hloubky na začátku P2 <S_I> a na konci P3 <S_J> roven 0, příp. pokud se celková přísvná hloubka P4 <S_K> rovná nule, uskuteční se pouze jeden vyjiskřovací zdvih.

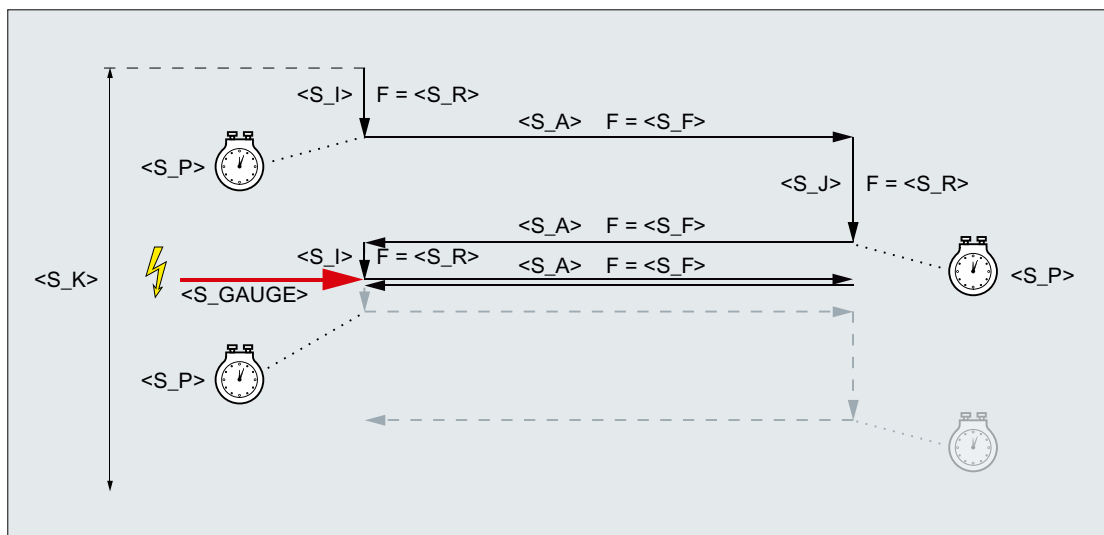
Postup

1. Spuštění cyklu na momentální pozici oscilační osy.
2. Pohyb přísluvné osy na přísluvnou hloubku na začátku P2 <S_I> s posuvem pro přísluv P6 <S_R>.
3. Vyjiskřování s dobou vyjiskřování P8 <S_P>.
4. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P5 <S_A> s posuvem pro přísluv v příčném směru P7 <S_F>.
5. Pohyb přísluvné osy na přísluvnou hloubku na konci P3 <S_J> s posuvem pro přísluv P6 <S_R>.
6. Vyjiskřování s dobou vyjiskřování P8 <S_P>.
7. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P5 <S_A> na počáteční bod a s posuvem pro přísluv v příčném směru P7 <S_F>.
8. Bez přerušení: Výše popisovaná operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky přísluvu P4 <S_K>. Poslední zdvih je potom rovnoměrně rozdělen.
S přerušením: Opracování se ukončí v počátečním bodě.

Tento postup při zpracování blok po bloku není možné přerušit.



Obrázek 20-7 Přerušení přísluvu na konci



Obrázek 20-8 Přerušení přísuvu na začátku

Zdroje

Jako zdroje cyklus používá synchronní akci překračující hranice bloků a proměnnou synchronní akce. Synchronní akce se dynamicky určuje z volné oblasti ve svazku synchronních akcí (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR - 1199 ...). Jako proměnná synchronní akce se používá SYG_IS[1].

Příklady

Příklad 1

Oscilační pohyb s těmito parametry:

- Přisuvná hloubka na začátku 0,02 mm
- Přisuvná hloubka na konci 0,01 mm
- Celková přisuvná hloubka 1 mm
- Zdvih 100 mm
- Posuv při přísuvu 1 mm/min
- Přisuv v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Přerušovací signál: rychlý vstup 1 (\$A_IN[1])

Programový kód

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```


Příklad 2

Oscilační pohyb s těmito parametry:

- Přisuvná hloubka na začátku 0,02 mm
- Přisuvná hloubka na konci 0,01 mm
- Celková přisuvná hloubka 1 mm
- Zdvih 100 mm
- Posuv při přisuvu 1 mm/min
- Přisuv v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Přerušovací signál: Proměnná z RAM se dvěma porty 20 je menší než 0,01 (\$A_DBR[20] < 0,01)

Programový kód

```
N10 T1 D1
N20 CYCLE4077 ("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.50 CYCLE4078 - Plošné broušení se spojitým přisuvem**Syntaxe**

CYCLE4078 (<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_I>	REAL	Přisuvná hloubka od začátku ke konci
2	<S_J>	REAL	Přisuvná hloubka od konce k začátku
3	<S_K>	REAL	Celková přisuvná hloubka
4	<S_A>	REAL	Broušená šířka
5	<S_F>	REAL	Posuv
6	<S_P>	REAL	Doba vyjiskřování
7	<S_A1>	AXIS	Přisuvná osa (nepovinné)
8	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

Tento cyklus slouží pro opracování s celkovou hloubkou přisuvu P3 <S_K> prostřednictvím spojitého přisuvu. Přisuvné hloubky od začátku ke konci P1 <S_I> a od konce k začátku P2 <S_J> se mohou lišit.

Údaje dráhy P1 až P4 mohou být kladné nebo záporné.

Zadání příravné osy P8 <S_A1> a/nebo oscilační osy P9 <S_A2> jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

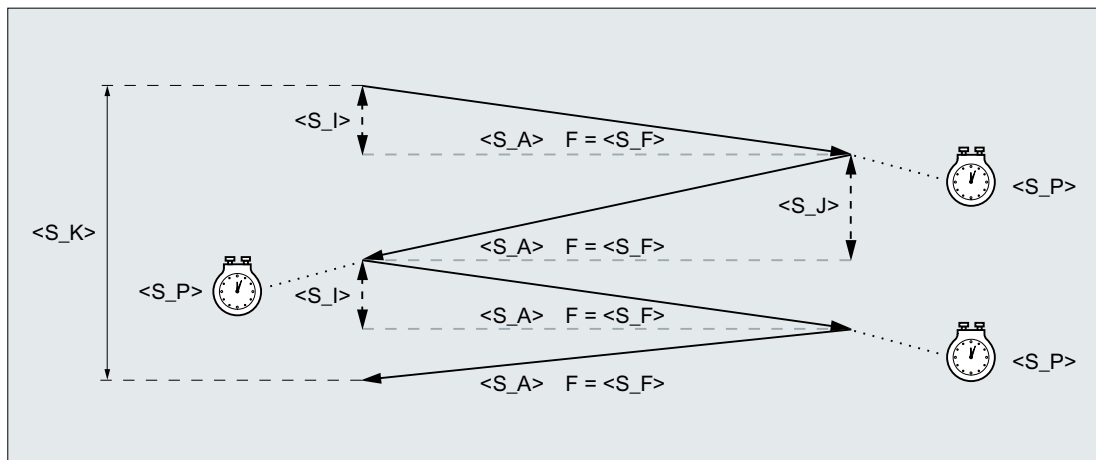
Pokud je součet příravných hloubek P1 <S_I> a P2 <S_J> roven 0, příp. pokud se celková příravná hloubka P3 <S_K> rovná nule, uskuteční se pouze jeden vyjiskřovací zdvih.

Postup

1. Spuštění cyklu na momentální pozici oscilační osy s příravnou hloubkou 0
2. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_A> a s posuvem P5 <S_F> se spojitým zvětšováním příravné hloubky až po příravnou hloubku na začátku P1 <S_I>.
3. Vyjiskřování s dobou vyjiskřování P7 <S_P>.
4. Pohyb oscilační osy na počáteční bod s dráhou posuvu odpovídající broušené šířce P4 <S_A> a s posuvem P5 <S_F> se spojitým zvětšováním příravné hloubky až po příravnou hloubku na konci P2 <S_J>.
5. Vyjiskřování s dobou vyjiskřování P7 <S_P>.
6. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 <S_A> na počáteční bod a s posuvem P5 <S_F>.

Tento postup při zpracování blok po bloku není možné přerušit.

Tato operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky příravnosti P3 <S_K>. Poslední zdvih je potom rovnoměrně rozdělen.



Příklad

Oscilační pohyb s těmito parametry:

- Příravná hloubka na začátku 20 mm
- Příravná hloubka na konci 10 mm
- Celková příravná hloubka 100 mm

- Zdvih 100 mm
- Posuv 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Programový kód

```

N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30

```

20.1.51 CYCLE4079 - Plošné broušení s přerušovaným přísuvem

Syntaxe

CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>, <S_A2>)

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_I>	REAL	Přisuvná hloubka na začátku
2	<S_J>	REAL	Přisuvná hloubka na konci
3	<S_K>	REAL	Celková přisuvná hloubka
4	<S_A>	REAL	Broušená šířka
5	<S_R>	REAL	Posuv pro přísuv
6	<S_F>	REAL	Posuv pro přísuv v příčném směru
7	<S_P>	REAL	Doba vyjiskřování
8	<S_A1>	AXIS	Přisuvná osa (nepovinné)
9	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

Tento cyklus slouží pro opracování s celkovou hloubkou přísluvu P3 <S_K> v několika přisuvných krocích. Přisuvné hloubky na začátku P1 <S_I> a na konci P2 <S_J> se mohou lišit. Mezi přísluvy se uskutečňuje tangenciální pohyb.

Údaje dráhy P1 až P4 mohou být kladné nebo záporné.

Zadání přisuvné osy P8 <S_A1> a/nebo oscilační osy P9 <S_A2> jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

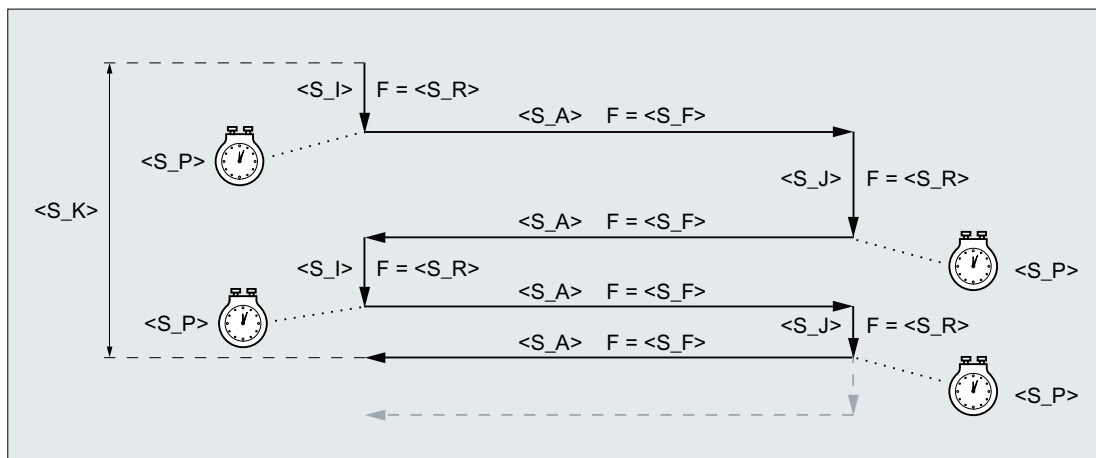
Pokud je součet přisuvné hloubky na začátku P1 <S_I> a na konci P2 <S_J> roven 0, příp. pokud se celková přisuvná hloubka P3 <S_K> rovná nule, uskuteční se pouze jeden vyjiskřovací zdvih.

Postup

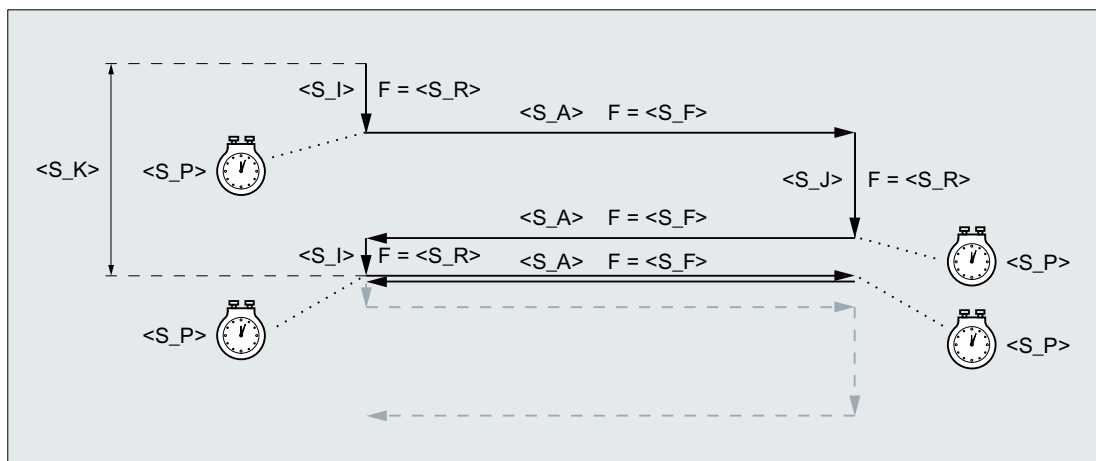
1. Spuštění cyklu na momentální pozici oscilační osy.
2. Pohyb přísluvné osy na přísluvnou hloubku na začátku P1 $\langle S_I \rangle$ s posuvem pro přísluv P5 $\langle S_R \rangle$.
3. Vyjiskřování s dobou vyjiskřování P7 $\langle S_P \rangle$.
4. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 $\langle S_A \rangle$ s posuvem pro přísluv v příčném směru P6 $\langle S_F \rangle$.
5. Pohyb přísluvné osy na přísluvnou hloubku na konci P2 $\langle S_J \rangle$ s posuvem pro přísluv P5 $\langle S_R \rangle$.
6. Vyjiskřování s dobou vyjiskřování P7 $\langle S_P \rangle$.
7. Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce P4 $\langle S_A \rangle$ na počáteční bod a s posuvem pro přísluv v příčném směru P6 $\langle S_F \rangle$.

Tento postup při zpracování blok po bloku není možné přerušit.

Tato operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky přísluvu P3 $\langle S_K \rangle$. Poslední zdvih je potom rovnoměrně rozdělen.



Obrázek 20-9 Celková přísluvná hloubka dosažená ve druhém bodu obratu



Obrázek 20-10 Celková přísluvná hloubka dosažená v prvním bodu obratu

Příklad

Oscilační pohyb s těmito parametry:

- Přísuvná hloubka na začátku 0,02 mm
- Přísuvná hloubka na konci 0,01 mm
- Celková přísuvná hloubka 1 mm
- Zdvih 100 mm
- Posuv při přísuvu 1 mm/min
- Přísuv v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Programový kód

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.52 GROUP_BEGIN - Začátek programového bloku**Syntaxe**

GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_LEVEL>	INT	Úroveň
				0 = Hlavní úroveň
				1 = 1. nižší úroveň
2		<_NAME>	STRING[128]	Název bloku
3		<_SP>	INT	Vřeteno
				0 = žádné vřeteno
				1 = Hlavní vřeteno
				2 = Protivřeteno
4		<_MODE>	INT	Režim
				Bit 0 = 1 GROUP_ADDEND existuje
				Bit 1 = 1 ShopTurn: automatický zpětný pohyb (najíždění na bod pro výměnu nástroje)
				Bit 12 rezervováno
				Bit 13 rezervováno
5		<S_ICON>	STRING[32]	Název ikony (jen pro uživatelské rozhraní)

20.1.53 GROUP_END - konec programového bloku

Syntaxe

GROUP_END (<_LEVEL>, <_SP>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_LEVEL>	INT	Úroveň
				0 = Hlavní úroveň
				1 = 1. nižší úroveň
2		<_SP>	INT	Vřeten
				0 = žádné vřeten
				1 = Hlavní vřeten
				2 = Protivřeten

20.1.54 GROUP_ADDEND - konec doplňkového bloku najíždění

Syntaxe

GROUP_ADDEND (<_LEVEL>, <_SP>)

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_LEVEL>	INT	Úroveň
				0 = Hlavní úroveň
				1 = 1. nižší úroveň
2		<_SP>	INT	Vřeten
				0 = žádné vřeten
				1 = Hlavní vřeten
				2 = Protivřeten

20.1.55 Okrajové podmínky

20.1.55.1 Odstupňování technologie na obrazovkách cyklů

Jestliže je odstupňování technologie aktivní, může být v různých obrazovkách cyklů zvoleno zjednodušené zadání, u kterého se vypisují pouze ty nejdůležitější parametry daného cyklu.

Zjednodušené zadávání může být zvoleno např. u následujících obrazovek cyklů:

Technologie	Obrazovka cyklu
Vrtání	Vrtání hlubokých děr
	Vrtání závitů
Frézování	Pravoúhlá kapsa
	Frézování kontury: Kapsa
Soustružení	Soustružení závitů: Podélný
	Soustružení kontur: Oddělování třísky
	Soustružení kontur: Výroba zápichů
	Soustružení kontur: Soustružení zápichů

U obrazovek cyklů, kterých se to týká, je na ploše uživatelského rozhraní k dispozici možnost volby „Zadání: **jednoduché**“ a „Zadání: **kompletní**„.

Nezobrazované parametry cyklu

Parametrům cyklu, které se při zjednodušeném zadávání nezobrazují, jsou dosazeny pevně dané, technologicky smysluplné hodnoty, které ale nelze změnit. Nebo jsou těmto parametrům cyklu přiřazeny hodnoty definované v příslušných nastavovaných parametrech cyklů daného kanálu. Viz níže odstavec "Uvedení do provozu" > "Kanálové nastavované parametry pro cykly"

Přepínání „Zadávání: kompletní“ > „Zadávání: jednoduché“

Jestliže je obrazovka cyklu vyplněna s nastavením „Zadávání kompletní“ a potom je přepnuto na nastavení „Zadávání jednoduché“, při generování volání cyklu jsou pro parametry, které se už nezobrazují, použity předdefinované hodnoty nebo hodnoty z nastavovaných parametrů.

Uvádění do provozu

Kanálové konfigurační strojní parametry

Prostřednictvím strojního parametru může být aktivováno odstupňování technologie na obrazovkách cyklů:

MD52210 \$MCS_FUNCTION_MASK_DISP, Bit 9 = 1 (volba „Zadávání jednoduché“ se zobrazuje)

Kanálové nastavované parametry pro cykly

Jestliže je na obrazovce cyklu aktivováno jednoduché zadávání, mohou být prostřednictvím následujících nastavovaných parametrů dosazeny určitým parametrům cyklů požadované hodnoty:

Číslo	Identifikátor	Význam
SD55300	\$SCS_EASY_SAFETY_CLEARANCE	Bezpečnostní vzdálenost
SD55301	\$SCS_EASY_DWELL_TIME	Doba prodlevy
SD55305	\$SCS_EASY_DRILL_DEEP_FD1	Vrtání hlubokých děr: Procentuální hodnota: 1. Posuv
SD55306	\$SCS_EASY_DRILL_DEEP_DF	Vrtání hlubokých děr: Procentuální hodnota: Přisuv

Číslo	Identifikátor	Význam
SD55307	\$SCS_EASY_DRILL_DEEP_V1	Vrtání hlubokých děr: Minimální příravná hloubka
SD55308	\$SCS_EASY_DRILL_DEEP_V2	Vrtání hlubokých děr: Vzdálenost zpětného pohybu
SD55309	\$SCS_EASY_THREAD_RETURN_DIST	Soustružení závitů: Vzdálenost pro zpětný pohyb

20.2 Měřicí cykly

Měřicí cykly jsou speciální podprogramy vytvořené firmou Siemens, které slouží pro řešení určitých měřicích úloh. Jak je u cyklů všeobecně běžné, mohou být i měřicí cykly přizpůsobeny konkrétnímu problému prostřednictvím parametrů.

Měřicí cykly jsou k dispozici pro měření v následujících oblastech a technologiích:

- Měření nástroje při soustružení / frézování
- Měření obrobků při soustružení / frézování

Literatura

Pokud budete potřebovat podrobný popis měřicích cyklů, viz:

Programovací příručka, Měřicí cykly

Tabulky

21.1 Příkazy

Poznámka

Cykly

Tento seznam příkazů obsahuje všechny cykly, které se mohou v NC programu (G-kódy) vyskytnout, tzn. které mohou být v editoru programu naprogramovány pomocí obrazovek, příp. které musí být v případě broušení naprogramovány bez podpory programování. Není brát ohled na cykly, které jsou z důvodů kompatibility v řídicím systému stále ještě k dispozici, ale prostřednictvím programového editoru systému SINUMERIK Operate už je není možné editovat („kompatibilní cykly“).

Příkazy A ... C

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
:	O	Číslo hlavního NC-bloku, ukončení návěští skoku, operátor zřetězení		+		PGAsl
*	O	Operátor pro násobení		+		PGAsl
+	O	Operátor pro sečítání		+		PGAsl
-	O	Operátor pro odečítání		+		PGAsl
<	O	Porovnávací operátor, je menší než		+		PGAsl
<<	O	Operátor zřetězení pro řetězce		+		PGAsl
<=	O	Porovnávací operátor, je menší nebo rovno než		+		PGAsl
=	O	Operátor přiřazení		+		PGAsl
>=	O	Porovnávací operátor, je větší nebo rovno než		+		PGAsl
/	O	Operátor pro dělení		+		PGAsl
/0 /7		Blok se bude přeskakovat (1. přeskakovaná úroveň) blok se bude přeskakovat (8. přeskakovaná úroveň)		+		PGsl
A	A	Název osy	m/s	+		PGAsl
A2	A	Orientace nástroje: Úhel RPY nebo Eulerův úhel	s	+		PGAsl
A3	A	Orientace nástroje: 1. Složka směrového vektoru	s	+		PGAsl
A4	A	Orientace nástroje: 1. Složka normálového vektoru plochy na začátku bloku	s	+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
A5	A	Orientace nástroje: 1. Složka normálového vektoru plochy na konci bloku	s	+		PGAsI
A6	A	Orientace nástroje: 1. Složka směrového vektoru pro osu kužele	s	+		PGAsI
A7	A	Orientace nástroje: 1. Vektorová složka orientace ve vnitřním bodě na plášťové ploše kužele	s	+		PGAsI
ABS	F	Absolutní hodnota		+	+	PGAsI
AC	K	Absolutní zadání rozměru pro souřadnice/pohyby	s	+		PGsI
ACC	K	Ovlivňování aktuálního zrychlení osy	m	+	+	PGsI
ACCLIMA	K	Ovlivňování aktuálního maximálního zrychlení osy	m	+	+	PGAsI
ACN	K	Zadání absolutního rozměru pro kruhovou osu, na pozici se najíždí v záporném směru	s	+		PGsI
ACOS	F	arkus kosinus (goniometr. funkce)		+	+	PGAsI
ACP	K	Zadání absolutního rozměru pro kruhovou osu, na pozici se najíždí v kladném směru	s	+		PGsI
ACTBLOCNO	P	Výstup aktuální čísla alarmového bloku, a to i když je aktivní příkaz "Potlačit výpis aktuálního bloku" (DISPLOF)!		+		PGAsI
ADDFRAME	F	Započítání a případné aktivování změřeného framu		+	-	PGAsI, FB1sI (K2)
ADIS	A	Vzdálenost zaoblení pro funkce pohybu po dráze G1, G2, G3, ...	m	+		PGsI
ADISPOS	A	Vzdálenost zaoblení pro rychlý posuv G0	m	+		PGsI
ADISPOSA	P	Velikost tolerančního okna pro IPOBRKA	m	+	+	PGAsI
ALF	A	Úhel rychlého zvedání	m	+		PGAsI
AMIRROR	G	Programovatelné zrcadlové převrácení	s	+		PGsI
AND	K	Logické A		+		PGAsI
ANG	A	Úhel konturové křivky	s	+		PGsI
AP	A	Polární úhel	m/s	+		PGsI
APR	K	Načtení / výpis přístupové ochrany		+		PGAsI
APRB	K	Načtení přístupové ochrany, BTSS		+		PGAsI
APRP	K	Načtení přístupové ochrany, výrobní program		+		PGAsI
APW	K	Zápis přístupové ochrany		+		PGAsI
APWB	K	Zápis přístupové ochrany, BTSS		+		PGAsI
APWP	K	Zápis přístupové ochrany, výrobní program		+		PGAsI
APX	K	Definice ochrany proti přístupu pro provedení uvedeného prvku jazyka		+		PGAsI
AR	A	Úhel kruhové výseče	m/s	+		PGsI
AROT	G	Programovatelné otočení	s	+		PGsI
AROTS	G	Programovatelné otáčení framu o prostorový úhel	s	+		PGsI

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
AS	K	Definice makra		+		PGAsl
ASCALE	G	Programovatelná změna měřítka	s	+		PGsl
ASIN	F	Matematická funkce, arkus sinus		+	+	PGAsl
ASPLINE	G	Akimovy spliny	m	+		PGAsl
ATAN2	F	arkus tangens2		+	+	PGAsl
ATOL	A	Tolerance pro specifickou osu pro funkce kompresoru, vyhlazení orientace a druhy zaoblování konturových přechodů	m	+		PGAsl
ATRANS	G	Aditivní programovatelné posunutí počátku	s	+		PGsl
AUXFUDEL	P	Vymazání pomocné kanálové funkce z globálního seznamu		+	-	FB1sl (H2)
AUXFUDELG	P	Vymazání všech pomocných kanálových funkcí z dané skupiny z globálního seznamu		+	-	FB1sl (H2)
AUXFUMSEQ	P	Zjištění posloupnosti při výstupu pomocných M-funkcí		+	-	FB1sl (H2)
AUXFUSYNC	P	Vygenerování řetězce, který reprezentuje kompletní blok výrobního programu pro kanálový příkaz SERUPRO-konec-ASUP, z globálního seznamu pomocných funkcí		+	-	FB1sl (H2)
AX	K	Proměnný identifikátor osy	m/s	+		PGAsl
AXCTSWE	P	Otočení osového zásobníku		+	-	PGAsl
AXCTSWEC	P	Uvolnění pro reset otočení osového zásobníku		+	+	PGAsl
AXCTSWED	P	Otočení osového zásobníku (varianta příkazu pro uvádění do provozu!)		+	-	PGAsl
AXIS	K	Identifikátor osy, adresa osy		+		PGAsl
AXNAME	F	Konvertovaný vstupní řetězec v identifikátoru osy		+	-	PGAsl
AXSTRING	F	Konvertuje řetězec čísla vřetena		+	-	PGAsl
AXTOCHAN	P	Vyžádání osy pro určitý kanál. Je možné z NC programu a ze synchronních akcí.		+	+	PGAsl
AXTOSPI	F	Konvertuje identifikátor osy na index vřetena		+	-	PGAsl
B	A	Název osy	m/s	+		PGAsl
B2	A	Orientace nástroje: Úhel RPY nebo Eulerův úhel	s	+		PGAsl
B3	A	Orientace nástroje: Složka směrového vektoru	s	+		PGAsl
B4	A	Orientace nástroje: 2. Složka normálového vektoru plochy na začátku bloku	s	+		PGAsl
B5	A	Orientace nástroje: 2. Složka normálového vektoru plochy na konci bloku	s	+		PGAsl
B6	A	Orientace nástroje: 2. Složka směrového vektoru pro osu kužele	s	+		PGAsl
B7	A	Orientace nástroje: 2. Vektorová složka orientace ve vnitřním bodě na plášťové ploše kužele	s	+		PGAsl
B_AND	O	Bitové logické A		+		PGAsl
B_OR	O	Bitové logické NEBO		+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
B_NOT	O	Bitová negace		+		PGAsI
B_XOR	O	Bitové logické XOR		+		PGAsI
BAUTO	G	Definice prvního splinového úseku prostřednictvím následujících tří bodů	m	+		PGAsI
BLOCK	K	Spolu s klíčovým slovem TO definuje úsek programu v nepřímo volaném podprogramu, který se má zpracovat.		+		PGAsI
BLSYNC	K	Zpracování rutiny přerušení má být zahájeno teprve s následující změnou bloku.		+		PGAsI
BNAT ⁶⁾	G	Přirozený přechod na první splinový blok	m	+		PGAsI
BOOL	K	Datový typ: Logické hodnoty TRUE/FALSE, příp. 1/0		+		PGAsI
BOUND	F	Zkontroluje, zda hodnota leží v rámci definovaného rozsahu hodnot. V případě rovnosti se vrací zpět zkoušená hodnota.		+	+	PGAsI
BRISK ⁶⁾	G	Skokové změny zrychlení po dráze	m	+		PGAsI
BRISKA	P	Aktivování skokových změn zrychlení při pohybu po dráze pro naprogramované osy		+	-	PGAsI
BSPLINE	G	B-Spline	m	+		PGAsI
BTAN	G	Tangenciální přechod na první splinový blok	m	+		PGAsI
C	A	Název osy	m/s	+		PGAsI
C2	A	Orientace nástroje: Úhel RPY nebo Eulerův úhel	s	+		PGAsI
C3	A	Orientace nástroje: 3. Složka směrového vektoru	s	+		PGAsI
C4	A	Orientace nástroje: 3. Složka normálového vektoru plochy na začátku bloku	s	+		PGAsI
C5	A	Orientace nástroje: 3. Složka normálového vektoru plochy na konci bloku	s	+		PGAsI
C6	A	Orientace nástroje: 3. Složka směrového vektoru pro osu kužele	s	+		PGAsI
C7	A	Orientace nástroje: 3. Vektorová složka orientace ve vnitřním bodě na plášťové ploše kužele	s	+		PGAsI
CAC	K	Absolutní najíždění na pozici		+		PGAsI
CACN	K	Na hodnotu uloženou v tabulce se bude najíždět v záporném směru a hodnota bude považována za absolutní.		+		PGAsI
CACP	K	Na hodnotu uloženou v tabulce se bude najíždět v kladném směru a hodnota bude považována za absolutní.		+		PGAsI
CALCDAT	F	Tato funkce vypočítá rádius a střed kruhu ne základě 3 nebo 4 bodů.		+	-	PGAsI
CALCPOSI	F	Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarové mezní hodnoty		+	-	PGAsI
CALL	K	Nepřímé volání podprogramu		+		PGAsI

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CALLPATH	P	Programovatelné vyhledávání cesty při volání podprogramu		+	-	PGAsl
CANCEL	P	Přerušení modální synchronní akce		+	-	FBSYsl
CASE	K	Podmíněné větvení programu		+		PGAsl
CDC	K	Přímé najíždění na pozici		+		PGAsl
CDOF ⁶⁾	G	Deaktivování monitorování kolize	m	+		PGsl
CDOF2	G	Deaktivování monitorování kolize, při 3D obvodovém frézování	m	+		PGsl
CDON	G	Aktivování monitorování kolize	m	+		PGsl
CFC ⁶⁾	G	Konstantní posuv na kontuře	m	+		PGsl
CFIN	G	Konstantní posuv, jen na vnitřních zakřiveních, ne na vnějších zakřiveních	m	+		PGsl
CFINE	F	Přiřazení jemného posunutí proměnné typu FRAME		+	-	PGAsl
CFTCP	G	Konstantní posuv na vztažném bodu bříty nástroje, dráha středu nástroje	m	+		PGsl
CHAN	K	Specifikace rozsahu platnosti dat		+		PGAsl
CHANDATA	P	Nastavení čísla kanálu pro přístup ke kanálovým datům		+	-	PGAsl
CHAR	K	Datový typ: Znaky ASCII		+		PGAsl
CHF	A	Faseta; hodnota = délka fasety	s	+		PGsl
CHKDM	F	Zkouška jednoznačnosti v rámci zásobníku		+	-	FBWsl
CHKDNO	F	Kontrola jednoznačnosti D-čísel		+	-	PGAsl
CHR	A	Faseta; Hodnota = délka fasety ve směru pohybu		+		PGsl
CIC	K	Inkrementální najíždění na pozici		+		PGAsl
CIP	G	Kruhová interpolace přes vnitřní bod	m	+		PGsl
CLEARM	P	Vymazání jedné/více značek pro přiřazení kanálu		+	+	PGAsl
CLRINT	P	Deaktivování přerušení		+	-	PGAsl
CMIRROR	F	Zrcadlové převrácení souřadné osy.		+	-	PGAsl
COARSEA	K	Konec pohybu při dosažení hrubého okna přesného najetí	m	+		PGAsl
COLLPAIR	F	Kontrola příslušnosti k protikolizní dvojici		+		PGAsl
COMPCAD	G	Aktivování funkce kompresoru COMPCAD	m	+		PGAsl
COMPCURV	G	Aktivování funkce kompresoru COMPCURV	m	+		PGAsl
COMPLETE		Řídící příkaz pro odesílání a příjem dat		+		PGAsl
COMPOF ⁶⁾	G	Deaktivování komprese NC bloků	m	+		PGAsl
COMPON	G	Aktivování funkce kompresoru COMPON	m	+		PGAsl
COMPSURF	G	Aktivování funkce kompresoru COMPSURF	m	+		PGAsl
CONTDCON	P	Aktivování dekodování kontury v tabulkové formě		+	-	PGAsl
CONTPRON	P	Aktivování referenční přípravy		+	-	PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CORROF	P	Všechny aktivní superponované pohyby jsou deaktivovány.		+	-	PGsl
CORRTRAF0	F	Změna offsetových vektorů nebo směrových vektorů orientačních os v kinematickém modelu stroje		+	-	PGAsl
COS	F	kosinus (goniometr. funkce)		+	+	PGAsl
COUPDEF	P	Definice vazby ELG / vazby synchron. vřeten		+	-	PGAsl
COUPDEL	P	Zrušení vazby ELG		+	-	PGAsl
COUPOF	P	Deaktivování vazby ELG / dvojice synchronních vřeten		+	-	PGAsl
COUPOFS	P	Vypnutí vazby ELG / dvojice synchronních vřeten se zastavením vlečného vřetena		+	-	PGAsl
COUPON	P	Aktivování vazby ELG / dvojice synchronních vřeten		+	-	PGAsl
COUPONC	P	Zapnutí vazby ELG / dvojice synchronních vřeten, převzít předešlé naprogramování		+	-	PGAsl
COUPRES	P	Reset vazby ELG		+	-	PGAsl
CP ⁶⁾	G	Pohyb po dráze	m	+		PGAsl
CPBC	K	Generická vazba: Kritérium přechodu na další blok		+	+	FB3sl (M3)
CPDEF	K	Generická vazba: Definice režimu vazby		+	+	FB3sl (M3)
CPDEL	K	Generická vazba: Vymazání režimu vazby		+	+	FB3sl (M3)
CPFMOF	K	Generická vazba: Chování vlečné osy při úplném deaktivování		+	+	FB3sl (M3)
CPFMON	K	Generická vazba: Chování vlečné vazby při aktivování		+	+	FB3sl (M3)
CPFMS0N	K	Generická vazba: Režim synchronní akce		+	+	FB3sl (M3)
CPFPOS	K	Generická vazba: Synchronní pozice vlečné osy		+	+	FB3sl (M3)
CPFRS	K	Generická vazba: Vztažný souřadný systém		+	+	FB3sl (M3)
CPLA	K	Generická vazba: Definice řídicí osy		+	-	FB3sl (M3)
CPLCTID	K	Generická vazba: Číslo tabulky křivky		+	+	FB3sl (M3)
CPLDEF	K	Generická vazba: Definice řídicí osy a definice režimu osy		+	+	FB3sl (M3)
CPLDEL	K	Generická vazba: Vymazání řídicí osy a režimu vazby		+	+	FB3sl (M3)
CPLDEN	K	Generická vazba: Jmenovatel faktoru vazby		+	+	FB3sl (M3)
CPLINSC	K	Generická vazba: Faktor měřítka pro vstupní hodnotu řídicí osy		+	+	FB3sl (M3)
CPLINTR	K	Generická vazba: Hodnota posunutí pro vstupní hodnotu řídicí osy		+	+	FB3sl (M3)
CPLNUM	K	Generická vazba: Číselník faktoru vazby		+	+	FB3sl (M3)
CPLOF	K	Generická vazba: Deaktivování řídicí osy režimu vazby		+	+	FB3sl (M3)

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CPLON	K	Generická vazba: Aktivování řídicí osy režimu vazby		+	+	FB3sl (M3)
CPLOUTSC	K	Generická vazba: Faktor měřítka pro výstupní hodnotu vazby		+	+	FB3sl (M3)
CPLOUTTR	K	Generická vazba: Hodnota posunutí pro výstupní hodnotu vazby		+	+	FB3sl (M3)
CPLPOS	K	Generická vazba: Synchronní pozice řídicí osy		+	+	FB3sl (M3)
CPLSETVAL	K	Generická vazba: Vztažený objekt vazby		+	+	FB3sl (M3)
CPMALARM	K	Generická vazba: Potlačení výstupu speciálních alarmů vztahujících se k vazbě		+	+	FB3sl (M3)
CPMBRAKE	K	Generická vazba: Chování vlečné osy při určitých signálech a příkazech pro zastavení		+	-	FB3sl (M3)
CPMPRT	K	Generická vazba: Chování vazby při spuštění výrobního programu pomocí hledání bloku prostřednictvím zkušebního zpracování programu		+	+	FB3sl (M3)
CPMRESET	K	Generická vazba: Chování vazby při resetu		+	+	FB3sl (M3)
CPMSTART	K	Generická vazba: Chování vazby při spuštění výrobního programu		+	+	FB3sl (M3)
CPMVDI	K	Generická vazba: Chování vlečné osy při určitých signálech rozhraní NC/PLC		+	+	FB3sl (M3)
CPOF	K	Generická vazba: Deaktivování režimu vazby		+	+	FB3sl (M3)
CPON	K	Generická vazba: Aktivování režimu vazby		+	+	FB3sl (M3)
CPRECOF ⁶⁾	G	Deaktivování programovatelné přesnosti kontury	m	+		PGAsl
CPRECON	G	Aktivování programovatelné přesnosti kontury	m	+		PGAsl
CPRES	K	Generická vazba: Aktivování dat vazby synchronních vřeten nastavených v konfiguraci		+	-	FB3sl (M3)
CPROT	P	Aktivování/deaktivování chráněné oblasti pro specifický kanál		+	-	PGAsl
CPROTDEF	P	Definice chráněné oblasti pro specifický kanál		+	-	PGAsl
CPSETTYPE	K	Generická vazba: Typ vazby		+	+	FB3sl (M3)
CPSYNCOF	K	Generická vazba: Prahová hodnota pro "hrubé" synchronní najíždění na určitou pozici		+	+	FB3sl (M3)
CPSYNCOF2	K	Generická vazba: Prahová hodnota pro "hrubé" synchronní najíždění 2 na určitou pozici		+	+	FB3sl (M3)
CPSYNCOV	K	Generická vazba: Mezní hodnota pro "hrubé" synchronní najíždění na určitou rychlost		+	+	FB3sl (M3)
CPSYNFIP	K	Generická vazba: Prahová hodnota pro "jemné" synchronní najíždění na určitou pozici		+	+	FB3sl (M3)
CPSYNFIP2	K	Generická vazba: Prahová hodnota pro "jemné" synchronní najíždění 2 na určitou pozici		+	+	FB3sl (M3)
CPSYNFIV	K	Generická vazba: Mezní hodnota pro "jemné" synchronní najíždění na určitou rychlost		+	+	FB3sl (M3)
CR	A	Rádus kruhu	s	+		PGsl
CROT	F	Otočení aktuálního souřadného systému		+	-	PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CROTS	F	Programovatelné otočení framu o prostorový úhel (otáčení v uvedených osách)	s	+	-	PGsl
CRPL	F	Otočení framu v libovolné rovině		+	-	FB1sl (K2)
CSCALE	F	Faktor změny měřítka pro více os		+	-	PGAsl
CSPLINE	F	Kubické spliny	m	+		PGAsl
CT	G	Kruh s tangenciálním přechodem	m	+		PGsl
CTAB	F	Zjistit polohu vlečné osy z tabulky křivek na základě polohy řídicí osy		+	+	PGAsl
CTABDEF	P	Aktivování definice tabulky		+	-	PGAsl
CTABDEL	P	Vymazání tabulky křivek		+	-	PGAsl
CTABEND	P	Deaktivování definice tabulky		+	-	PGAsl
CTABEXISTS	F	Kontrola tabulky křivek s číslem n		+	+	PGAsl
CTABFNO	F	Počet ještě možných tabulek křivek v paměti		+	+	PGAsl
CTABFPOL	F	Počet ještě možných polynomů v paměti		+	+	PGAsl
CTABFSEG	F	Počet ještě možných křivkových segmentů v paměti		+	+	PGAsl
CTABID	F	Zjištění čísla tabulky n-té tabulky křivek		+	+	PGAsl
CTABINV	F	Zjistit polohu řídicí osy z tabulky křivek na základě polohy vlečné osy		+	+	PGAsl
CTABISLOCK	F	Deaktivování zablokování tabulky křivek s číslem n		+	+	PGAsl
CTABLOCK	P	Zablokování mazání a přepisování		+	+	PGAsl
CTABMEMTYP	F	Zjištění paměti, ve které je uložena tabulka křivek s číslem n.		+	+	PGAsl
CTABMPOL	F	Maximální možný počet polynomů v paměti		+	+	PGAsl
CTABMSEG	F	Maximální možný počet křivkových segmentů v paměti		+	+	PGAsl
CTABNO	F	Počet definovaných křivkových tabulek v paměti SRAM nebo DRAM		+	+	FB3sl (M3)
CTABNOMEM	F	Počet definovaných křivkových tabulek v paměti SRAM nebo DRAM		+	+	PGAsl
CTABPERIOD	F	Zjištění periodicity tabulky křivek s číslem n		+	+	PGAsl
CTABPOL	F	Počet již používaných polynomů v paměti		+	+	PGAsl
CTABPOLID	F	Počet křivkových polynomů používaných tabulkou křivek s číslem n		+	+	PGAsl
CTABSEG	F	Počet již používaných křivkových segmentů v paměti		+	+	PGAsl
CTABSEGID	F	Počet křivkových segmentů používaných tabulkou křivek s číslem n		+	+	PGAsl
CTABSEV	F	Zjištění koncové hodnoty vlečné osy segmentu křivkové tabulky		+	+	PGAsl
CTABSSV	F	Zjištění počáteční hodnoty vlečné osy segmentu křivkové tabulky		+	+	PGAsl
CTABTEP	F	Zjištění hodnoty řídicí osy na konci křivkové tabulky		+	+	PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CTABTEV	F	Zjištění hodnoty vlečné osy na konci křivkové tabulky		+	+	PGAsl
CTABTMAX	F	Zjištění maximální hodnoty vlečné osy křivkové tabulky		+	+	PGAsl
CTABTMIN	F	Zjištění minimální hodnoty vlečné osy křivkové tabulky		+	+	PGAsl
CTABTSP	F	Zjištění hodnoty řídící osy na začátku křivkové tabulky		+	+	PGAsl
CTABTSV	F	Zjištění hodnoty vlečné osy na začátku křivkové tabulky		+	+	PGAsl
CTABUNLOCK	P	Odstranění blokování mazání a přepisování		+	+	PGAsl
CTOL	A	Tolerance kontury pro funkce kompresoru, vyhlazení orientace a druhy zaoblování konturových přechodů	m	+		PGAsl
CTTRANS	F	Posunutí počátku pro více os.		+	-	PGAsl
CUT2D ⁶⁾	G	2D korekce rádiusu nástroje	m	+		PGsl
CUT2DD	G	2½-D-korekce rádiusu nástroje vztažená na diferenční nástroj	m	+		PGsl
CUT2DF	G	2D korekce rádiusu nástroje, vzhledem k aktuálnímu framu (šikmá rovina)	m	+		PGsl
CUT2DFD	G	2½-D-korekce rádiusu nástroje vztažená na diferenční nástroj, vzhledem k aktuálnímu framu (šikmá rovina)	m	+		PGsl
CUT3DC	G	3D korekce rádiusu nástroje pro obvodové frézování	m	+		PGAsl
CUT3DCC	G	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s 3D korekcí rádiusu: Kontura na obráběné ploše	m	+		PGAsl
CUT3DCCD	G	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s diferenčním nástrojem na dráze středu nástroje: Přísuv k omezující ploše.	m	+		PGAsl
CUT3DCD	G	3D korekce rádiusu nástroje pro obvodové frézování vztažená na diferenční nástroj	m	+		PGAsl
CUT3DF	G	3D korekce rádiusu nástroje pro frézování na čelní ploše se změnou orientace	m	+		PGAsl
CUT3DFD	G	3D korekce rádiusu nástroje pro frézování na čelní ploše se změnou orientace vztažená na diferenční nástroj	m	+		PGAsl
CUT3DFF	G	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a v případě potřeby může být pomocí framu otočena do jiného směru.	m	+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CUT3DFS	G	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a nemůže být ovlivňována pomocí framů.	m	+		PGAsI
CUTCONOF ⁶⁾	G	Deaktivování konstantní korekce rádiusu	m	+		PGsI
CUTCONON	G	Aktivování konstantní korekce rádiusu	m	+		PGsI
CUTMOD	A	Aktivování funkce "Modifikace korekčních parametrů u točivých nástrojů" (ve spojení s orientovatelnými držáky nástrojů)	m	+		PGAsI
CUTMODK	A	Aktivování funkce "Modifikace korekčních parametrů u točivých nástrojů" (ve spojení s transformacemi orientace, které byly definovány prostřednictvím kinematického řetězce)	m	+		PGAsI
CYCLE60	C (T)	Gravírovací cyklus		+		PGAsI
CYCLE61	C (T)	Rovinné frézování		+		PGAsI
CYCLE62	C (T)	Volání kontury		+		PGAsI
CYCLE63	C (T)	Frézování konturové kapsy		+		PGAsI
CYCLE64	C (T)	Předvrtání konturové kapsy		+		PGAsI
CYCLE70	C (T)	Frézování závitů		+		PGAsI
CYCLE72	C (T)	Frézování po dráze		+		PGAsI
CYCLE76	C (T)	Frézování pravoúhlého čepu		+		PGAsI
CYCLE77	C (T)	Frézování kruhového čepu		+		PGAsI
CYCLE78	C (T)	Frézování vrtaných závitů		+		PGAsI
CYCLE79	C (T)	Mnohohran		+		PGAsI
CYCLE81	C (T)	Vrtání, navrtávání středících důlků		+		PGAsI
CYCLE82	C (T)	Vrtání, čelní zahlubování		+		PGAsI
CYCLE83	C (T)	Vrtání hlubokých děr		+		PGAsI
CYCLE84	C (T)	Vrtání závitů bez vyrovnávací hlavičky		+		PGAsI
CYCLE85	C (T)	Vystružování		+		PGAsI
CYCLE86	C (T)	Vyvrtávání		+		PGAsI
CYCLE92	C (T)	Upichování		+		PGAsI
CYCLE95	C (T)	Obrábění kontury		+		PGAsI
CYCLE98	C (T)	Řetězec závitů		+		PGAsI
CYCLE99	C (T)	Soustružení závitů		+		PGAsI
CYCLE150	C (M)	Vypisování/ukládání do protokolu výsledků měření		+		BNMsI
CYCLE435	C (T)	Výpočet polohy orovnávače		+		PGAsI
CYCLE495	C (T)	Profilování		+		PGAsI
CYCLE750	C (A)	Interní pracovní cyklus pro cykly CYCLE751 ... CYCLE759 (obsahuje příkaz MMC pro vlastní volání funkce)		-		FB3sI (T4)
CYCLE751	C (A)	Zahájení / uskutečnění / ukončení optimalizační procedury		M		FB3sI (T4)
CYCLE752	C (A)	Připojení osy do optimalizační procedury		M		FB3sI(T4)

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CYCLE753	C (A)	Aktivování režimu optimalizace		M		FB3sl (T4)
CYCLE754	C (A)	Vložení / odstranění datového bloku		M		FB3sl (T4)
CYCLE755	C (A)	Uložit / opětovné načtení datového bloku		M		FB3sl (T4)
CYCLE756	C (A)	Aktivování výsledků optimalizace		M		FB3sl (T4)
CYCLE757	C (A)	Uložení optimalizačních dat		M		FB3sl (T4)
CYCLE758	C (A)	Změna hodnoty parametru		M		FB3sl (T4)
CYCLE759	C (A)	Načtení hodnoty parametru		M		FB3sl (T4)
CYCLE800	C (T)	Nakládění		+		PGAsl
CYCLE801	C (T)	Mřížka nebo obdélník		+		PGAsl
CYCLE802	C (T)	Libovolné polohy		+		PGAsl
CYCLE830	C (T)	Vrtání hlubokých děr 2		+		PGAsl
CYCLE832	C (T)	Parametry pro vysokorychlostní obrábění		+		PGAsl
CYCLE840	C (T)	Vrtání závitů s vyrovnávací hlavičkou		+		PGAsl
CYCLE899	C (T)	Frézování otevřené drážky		+		PGAsl
CYCLE930	C (T)	Zápich		+		PGAsl
CYCLE940	C (T)	Odlehčovací zápich určitého tvaru		+		PGAsl
CYCLE951	C (T)	Oddělování třísky		+		PGAsl
CYCLE952	C (T)	Zápichy na kontuře		+		PGAsl
CYCLE961	C (M)	Stanovení polohy rohu obrobku (vnitřního nebo vnějšího) a její dosazení jako posunutí počátku.		+		BNMsl
CYCLE971	C (M)	Kalibrace nástrojové měřicí sondy, měření délky a/nebo rádiusu nástroje (jen pro technologii frézování)		+		BNMsl
CYCLE973	C (M)	Kalibrace obrobkové měřicí sondy na ploše na obrobku nebo v drážce (pouze pro technologii soustružení)		+		BNMsl
CYCLE974	C (M)	Stanovení počátku souřadného systému obrobku ve zvolené měřicí ose, stanovení korekce nástroje pomocí měření 1 bodu (pouze pro technologii soustružení)		+		BNMsl
CYCLE976	C (M)	Uskutečňování kalibrace obrobkové měřicí sondy v kalibračním kruhu, příp. na kalibrační kuličce, zcela v pracovní rovině nebo na hraně pro určitou osu a směr		+		BNMsl
CYCLE977	C (M)	Stanovení středu v rovině, jakož i šířky, příp. průměru		+		BNMsl
CYCLE978	C (M)	Měření polohy hrany v souřadném systému obrobku		+		BNMsl
CYCLE979	C (M)	Stanovení středu v rovině a měření rádiusu kruhového segmentu		+		BNMsl
CYCLE982	C (M)	Kalibrace nástrojové měřicí sondy, měření soustružnických, vrtacích a frézovacích nástrojů (jen pro technologii soustružení)		+		BNMsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
CYCLE994	C (M)	Stanovení počátku souřadného systému obrobku ve zvolené měřicí ose pomocí měření 2 bodů (pouze pro technologii soustružení)		+		BNMsl
CYCLE995	C (M)	Měření úhlové chyby vřetena na obráběcím stroji		+		BNMsl
CYCLE996	C (M)	Stanovení důležitých parametrů pro kinematické transformace s kruhovými osami		+		BNMsl
CYCLE997	C (M)	Určování středu a průměru koule, měření středů tří libovolně umístěných koulí		+		BNMsl
CYCLE998	C (M)	Určování úhlové polohy plochy (roviny) vzhledem k pracovní rovině, stanovení úhlu hran v souřadném systému obrobku		+		BNMsl
CYCLE4071	C (T)	Podélné broušení s přísuvem na bod obratu		+		PGAsl
CYCLE4072	C (T)	Podélné broušení s přísuvem na bod obratu a se signálem pro přerušení		+		PGAsl
CYCLE4073	C (T)	Podélné broušení se spojitým přísuvem		+		PGAsl
CYCLE4074	C (T)	Podélné broušení se spojitým přísuvem a se signálem pro přerušení		+		PGAsl
CYCLE4075	C (T)	Plošné broušení s přísuvem na bod obratu		+		PGAsl
CYCLE4077	C (T)	Plošné broušení s přísuvem na bod obratu a se signálem pro přerušení		+		PGAsl
CYCLE4078	C (T)	Plošné broušení se spojitým přísuvem		+		PGAsl
CYCLE4079	C (T)	Plošné broušení s přerušovaným přísuvem		+		PGAsl

Příkazy D ... F

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
D	A	Číslo korekčních parametrů nástroje		+		PGsl
D0	A	Je-li použito D0, jsou korekční parametry daného nástroje deaktivovány.		+		PGsl
DAC	K	Absolutní blokové programování průměrů pro specifickou osu	s	+		PGsl
DC	K	Údaj absolutního rozměru pro kruhové osy, na pozici se najíždí přímo	s	+		PGsl
DCI	K	Přiřazení datové třídy I (= Individuální) (jen SI-NUMERIK 828D!)		+		PGAsl
DCM	K	Přiřazení datové třídy M (= Výrobce) (jen SI-NUMERIK 828D!)		+		PGAsl
DCU	K	Přiřazení datové třídy U (= Uživatel) (jen SI-NUMERIK 828D!)		+		PGAsl
DEF	K	Definice proměnných		+		PGAsl
DEFAULT	K	Větev v příkazu větvení CASE		+		PGAsl
DEFINE	K	Klíčové slovo pro definice maker		+		PGAsl
DELAYFSTOF	P	Definice konce oblasti zastavení-zpoždění	m	+	-	PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
DELAYFSTON	P	Definice začátku oblasti zastavení-zpoždění	m	+	-	PGAsl
DELDL	F	Vymazání aditivních korekcí		+	-	PGAsl
DELDTG	P	Vymazání zbytkové dráhy		-	+	FBSYsl
DELETE	P	Vymazání zadaného souboru. Název souboru může být zadán pomocí cesty a identifikace souboru.		+	-	PGAsl
DELMLOWNER	F	Vymazání místa v zásobníku, které je napevno přiřazeno nástroji		+	-	FBWsl
DEMLRES	F	Vymazání rezervace místa v zásobníku		+	-	FBWsl
DELMT	P	Vymazat modul Multitool		+	-	FBWsl
DELOBJ	F	Mazání prvků z kinematických řetězců, chráněných oblastí, prvků chráněných oblastí, dvojic oblastí chráněných kvůli kolizi a dat transformací		+		PGAsl
DELT	P	Vymazání nástroje		+	-	FBWsl
DELTC	P	Vymazání datového bloku držáku nástroje		+	-	FBWsl
DELTOLENV	F	Vymazání datových bloků pro popis nástrojového prostředí		+	-	PGAsl
DIACYCOFA	K	Modální programování průměrů pro specifickou osu: V cyklech deaktivováno	m	+		FB1sl (P1)
DIAM90	G	Programování průměrů pro G90, programování rádiusů pro G91	m	+		PGAsl
DIAM90A	K	Modální programování průměrů pro G90 a AC, programování rádiusů pro G91 a IC pro specifickou osu	m	+		PGsl
DIAMCHAN	K	Převzetí všech os ze strojních parametrů týkajících se funkcí os v kanálovém stavu programování průměrů		+		PGsl
DIAMCHANA	K	Převzetí programování průměrů ze specifického kanálu		+		PGsl
DIAMCYCOF	G	Programování průměrů pro specifický kanál: V cyklech deaktivováno	m	+		FB1sl (P1)
DIAMOF ⁶⁾	G	Programování průměrů: Vypnuto Základní nastavení viz údaje od výrobce stroje	m	+		PGsl
DIAMOFA	K	Zapnutí modálního programování průměrů pro specifickou osu: Vypnuto Základní nastavení viz údaje od výrobce stroje	m	+		PGsl
DIAMON	G	Programování průměrů: Zapnuto	m	+		PGsl
DIAMONA	K	Zapnutí modálního programování průměrů pro specifickou osu: Zapnuto Aktivování viz údaje od výrobce stroje	m	+		PGsl
DIC	K	Relativní blokové programování průměrů pro specifickou osu	s	+		PGsl
DILF	A	Návratová dráha (délka)	m	+		PGsl
DISABLE	P	Vypnutí přerušení		+	-	PGAsl
DISC	A	Převýšení přechodové kružnice – korekce rádiusu nástroje	m	+		PGsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
DISCL	A	Vzdálenost koncového bodu rychlého přísuvného pohybu od roviny obrábění		+		PGsl
DISPLOF	PA	Potlačení vypisování aktuálního bloku		+		PGAsl
DISPLON	PA	Odstranění potlačení vypisování aktuálního bloku		+		PGAsl
DISPR	A	Rozdíl dráhy pro zpětné polohování	s	+		PGAsl
DISR	A	Vzdálenost pro zpětné polohování	s	+		PGAsl
DISRP	A	Vzdálenost návratové roviny od roviny opracování při měkkém najíždění a odjíždění		+		PGsl
DITE	A	Výběr závitu	m	+		PGsl
DITS	A	Náběh závitu	m	+		PGsl
DIV	K	Celočíselné dělení		+		PGAsl
DL	A	Aktivování lokálně závislé aditivní korekce nástroje (DL, součtová a seřizovací korekce)	m	+		PGAsl
DO	A	Klíčové slovo pro synchronní akci, ke spuštění dojde, když je podmínka splněna.		-	+	FBSYsl
DRFOF	P	Vypnutí posunutí ručním kolečkem (DRF)	m	+	-	PGsl
DRIVE	G	Zrychlení po dráze závislé na rychlosti	m	+		PGAsl
DRIVEA	P	Zapnutí lomené charakteristiky zrychlení pro naprogramované osy		+	-	PGAsl
DYNFINISH	G	Dynamika pro jemné obrábění načisto	m	+		PGAsl
DYNNORM ⁶⁾	G	Normální dynamika	m	+		PGAsl
DYNPOS	G	Dynamika pro režim polohování, vrtání závitů	m	+		PGAsl
DYNROUGH	G	Dynamika pro obrábění nahrubo	m	+		PGAsl
DYNSEMIFIN	G	Dynamika pro obrábění načisto	m	+		PGAsl
DZERO	P	Označení všech D-čísel dané jednotky TO jako neplatné		+	-	PGAsl
EAUTO	G	Definice posledního splinového úseku prostřednictvím posledních 3 bodů	m	+		PGAsl
EGDEF	P	Definice elektronické převodovky		+	-	PGAsl
EGDEL	P	Vymazání definice vazby pro vlečnou osu		+	-	PGAsl
EGOFC	P	Kontinuální vypínání elektronické převodovky		+	-	PGAsl
EGOFS	P	Selektivní vypnutí elektronické převodovky		+	-	PGAsl
EGON	P	Zapnutí elektronické převodovky		+	-	PGAsl
EGONSYN	P	Zapnutí elektronické převodovky		+	-	PGAsl
EGONSYNE	P	Zapnutí elektronické převodovky, spolu se zadáním režimu najíždění		+	-	PGAsl
ELSE	K	Větvení programu, pokud podmínka IF není splněna		+		PGAsl
ENABLE	P	Zapnutí přerušení		+	-	PGAsl
ENAT ⁶⁾	G	Přirozený přechod na následující blok posuvu	m	+		PGAsl
ENDFOR	K	Koncový řádek smyčky FOR se zadáním počtem průchodů		+		PGAsl
ENDIF	K	Koncový řádek větvení IF		+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
ENDLABEL	K	Koncové návěští pro opakování výrobního programu pomocí příkazu REPEAT		+		PGAsl, FB1sl (K1)
ENDLOOP	K	Koncový řádek nekonečné programové smyčky LOOP		+		PGAsl
ENDPROC	K	Koncový řádek programu, na jehož prvním řádku je příkaz PROC.		+		
ENDWHILE	K	Koncový řádek smyčky WHILE		+		PGAsl
ESRR	P	Nastavení parametrů pro odjíždění nezávislé na pohonu (ESR)		+		PGAsl
ESRS	P	Nastavení parametrů pro zastavení nezávislé na pohonu (ESR)		+		PGAsl
ETAN	G	Tangenciální křivkový přechod na následující blok posuvu na začátku splinu	m	+		PGAsl
EVERY	K	Synchronní akce se uskuteční v případě změny stavu podmínky z FALSE na TRUE		-	+	FBSYsl
EX	K	Klíčové slovo pro přiřazení hodnoty v exponenciálním způsobu zápisu		+		PGAsl
EXECSTRING	P	Předání proměnné typu String se zpracovatelným řádkem výrobního programu		+	-	PGAsl
EXECTAB	P	Spuštění zpracování prvku z tabulky pohybů		+	-	PGAsl
EXECUTE	P	Zapnutí zpracovávání programu		+	-	PGAsl
EXP	F	Exponenciální funkce ex		+	+	PGAsl
EXTCALL	A	Zpracovávání externího podprogramu		+	+	PGAsl
EXTCLOSE	P	Zavření externího přístroje/souboru otevřeného pro zápis		+	-	PGAsl
EXTERN	K	Identifikace podprogramu s předáváním parametrů		+		PGAsl
EXTOPEN	P	Otevření externího přístroje/souboru pro daný kanál pro zápis		+	-	PGAsl
F	A	Hodnota posuvu (ve spojení s G4 se pomocí F programuje také doba prodlevy)		+	+	PGsl
FA	K	Posuv osy	m	+	+	PGsl
FAD	A	Posuv pro přísuv při měkkém najíždění a odjíždění		+		PGsl
FALSE	K	Logická konstanta: FALSE		+	+	PGAsl
FB	A	Blokový posuv		+		PGsl
FCTDEF	P	Definice polynomické funkce		+	-	PGAsl
FCUB	G	Posuv proměnný podle kubického splinu	m	+		PGAsl
FD	A	Posuv po dráze pro korekci pomocí ručního kolečka	s	+		PGsl
FDA	K	Posuv osy při korekci ručním kolečkem	s	+		PGsl
FENDNORM ⁶⁾	G	Vypnutí zpoždování v rozích	m	+		PGAsl
FFWOF ⁶⁾	G	Vypnutí dopředné regulace	m	+		PGAsl
FFWON	G	Zapnutí dopředné regulace	m	+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
FGREF	K	Vztažný rádius u kruhových os nebo vztažný faktor dráhy u orientovaných os (vektorová interpolace)	m	+		PGsl
FGROUP	P	Definice os s posuvem po dráze		+	-	PGsl
FI	K	Parametr pro přístup k datům framu: Jemné posunutí		+		PGAsl
FIFOCTRL	G	Ovládání zásobníku dopředné regulace	m	+		PGAsl
FILEDATE	P	Zjištění data posledního přístupu za účelem zápisu do daného souboru		+	-	PGAsl
FILEINFO	P	Zjištění parametrů FILEDATE, FILESIZE, FILESTAT a FILETIME najednou		+	-	PGAsl
FILESIZE	P	Zjištění aktuální velikosti souboru		+	-	PGAsl
FILESTAT	P	Zjištění stavových informací o souboru, jako jsou oprávnění pro čtení, zápis, spouštění, vypisování a mazání (rwxs)		+	-	PGAsl
FILETIME	P	Zjištění času posledního přístupu za účelem zápisu do daného souboru		+	-	PGAsl
FINEA	K	Konec pohybu při dosažení jemného okna přesného najetí	m	+		PGAsl
FL	K	Mezní hodnota rychlosti pro synchronní osy	m	+		PGsl
FLIN	G	Lineárně proměnný posuv	m	+		PGAsl
FMA	K	Větší počet axiálních posuvů	m	+		PGsl
FNORM ⁶⁾	G	Normální posuv podle DIN 66025	m	+		PGAsl
FOC	K	Omezení momentu/síly s blokovanou platností	s	-	+	FBSYsl
FOCOF	K	Deaktivování modálního omezení momentu/síly	m	-	+	FBSYsl
FOCON	K	Aktivování modálního omezení momentu/síly	m	-	+	FBSYsl
FOR	K	Smyčka s pevně daným počtem průchodů		+		PGAsl
FP	A	Pevný bod: Číslo pevného bodu, na který se má najíždět	s	+		PGsl
FPO	K	Průběh posuvu naprogramovaný pomocí polynomu		+		PGAsl
FPR	P	Označení kruhové osy		+	-	PGsl
FPRAOF	P	Deaktivování otáčkového posuvu		+	-	PGsl
FPRAON	P	Aktivování otáčkového posuvu		+	-	PGsl
FRAME	K	Datový typ určený pro definici souřadných systémů		+		PGAsl
FRC	A	Posuv pro rádius a fasetu	s	+		PGsl
FRCM	A	Modální posuv pro rádius a fasetu	m	+		PGsl
FROM	K	Akce se uskuteční, jestliže je daná podmínka jednorázově splněna a pokud je aktivní synchronní akce.		-	+	FBSYsl
FTOC	P	Změna jemné korekce nástroje		-	+	FBSYsl
FTOCOF ⁶⁾	G	Vypnutí on-line působící jemné korekce nástroje	m	+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
FTOCON	G	Zapnutí on-line působící jemné korekce nástroje	m	+		PGsl
FXS	K	Najíždění na pevný doraz	m	+	+	PGsl
FXST	K	Mezní hodnota momentu pro najíždění na pevný doraz	m	+	+	PGsl
FXSW	K	Monitorovací okno pro najíždění na pevný doraz		+	+	PGsl
FZ	K	Posuv na zub	m	+		PGsl

Příkazy G ... L

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
G0	G	Lineární interpolace rychlým posuvem (pohyb rychlým posuvem)	m	+		PGsl
G1 ⁶⁾	G	Lineární interpolace s pracovním posuvem (přímková interpolace)	m	+		PGsl
G2	G	Kruhová interpolace ve směru hodinových ručiček	m	+		PGsl
G3	G	Kruhová interpolace proti směru hodinových ručiček	m	+		PGsl
G4	G	Doba prodlevy určená časově	s	+		PGsl
G5	G	Šikmé zapichovací broušení	s	+		PGAsl
G7	G	Vyrovňovací pohyb při šikmém zapichovacím broušení	s	+		PGAsl
G9	G	Přesné najetí – snižování rychlosti	s	+		PGsl
G17 ⁶⁾	G	Volba pracovní roviny X/Y	m	+		PGsl
G18	G	Volba pracovní roviny Z/X	m	+		PGsl
G19	G	Volba pracovní roviny Y/Z	m	+		PGsl
G25	G	Dolní ohraničení pracovního pole	s	+		PGsl
G26	G	Horní ohraničení pracovního pole	s	+		PGsl
G33	G	Řezání závitů s konstantním stoupáním	m	+		PGsl
G34	G	Řezání závitu s lineárně narůstajícím stoupáním	m	+		PGsl
G35	G	Řezání závitu s lineárně klesajícím stoupáním	m	+		PGsl
G40 ⁶⁾	G	Vypnutí korekce rádiusu nástroje	m	+		PGsl
G41	G	Korekce rádiusu nástroje vlevo od kontury	m	+		PGsl
G42	G	Korekce rádiusu nástroje vpravo od kontury	m	+		PGsl
G53	G	Potlačení aktuálního posunutí počátku (blokové)	s	+		PGsl
G54	G	1. nastavitelné posunutí počátku	m	+		PGsl
G55	G	2. nastavitelné posunutí počátku	m	+		PGsl
G56	G	3. nastavitelné posunutí počátku	m	+		PGsl
G57	G	4. nastavitelné posunutí počátku	m	+		PGsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
G58 (840D sl)	G	Absolutní programovatelné posunutí počátku (hrubé posunutí)	s	+		PGsl
G58 (828D)	G	5. nastavitelné posunutí počátku	m	+		PGsl
G59 (840D sl)	G	Aditivní programovatelné posunutí počátku (jemné posunutí)	s	+		PGsl
G59 (828D)	G	6. nastavitelné posunutí počátku	m	+		PGsl
G60 ⁶⁾	G	Přesné najetí – snižování rychlosti	m	+		PGsl
G62	G	Zpoždění na vnitřních rozích při aktivní korekci rádiusu nástroje (G41, G42)	m	+		PGAsl
G63	G	Vrtání závitů s vyrovnávací hlavičkou	s	+		PGsl
G64	G	Režim řízení pohybu po dráze	m	+		PGsl
G70	G	Rozměrové údaje pro geometrické parametry v palcích (délky)	m	+	+	PGsl
G71 ⁶⁾	G	Rozměrové údaje pro geometrické parametry v metrických jednotkách (délky)	m	+	+	PGsl
G74	G	Najíždění na referenční bod	s	+		PGsl
G75	G	Najíždění na pevný bod	s	+		PGsl
G90 ⁶⁾	G	Zadávání absolutních rozměrů	m/s	+		PGsl
G91	G	Zadávání inkrementálních rozměrů	m/s	+		PGsl
G93	G	Časově reciproční posuv v jednotkách 1/min	m	+		PGsl
G94 ⁶⁾	G	Lineární posuv F v mm/min, v palcích/min nebo stupních/min	m	+		PGsl
G95	G	Otáčkový posuv F v mm/ot nebo palcích/ot	m	+		PGsl
G96	G	Otáčkový posuv (stejně jako u G95) a konstantní řezná rychlost	m	+		PGsl
G97	G	Otáčkový posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	m	+		PGsl
G110	G	Programování pólu vztahované na naposled naprogramovanou požadovanou pozici	s	+		PGsl
G111	G	Programování pólu vzhledem k počátku aktuální souřadné soustavy obrobku	s	+		PGsl
G112	G	Programování pólu vzhledem k poslednímu platnému pólu	s	+		PGsl
G140 ⁶⁾	G	Směr najíždění WAB definován příkazy G41/ G42	m	+		PGsl
G141	G	Směr najíždění WAB vlevo od kontury	m	+		PGsl
G142	G	Směr najíždění WAB vpravo od kontury	m	+		PGsl
G143	G	Směr najíždění WAB v závislosti na tečně	m	+		PGsl
G147	G	Měkké najíždění po přímce	s	+		PGsl
G148	G	Měkké odjíždění po přímce	s	+		PGsl
G153	G	Potlačení aktuálního framu včetně základního framu	s	+		PGsl
G247	G	Měkké najíždění po čtvrtkruhu	s	+		PGsl
G248	G	Měkké odjíždění po čtvrtkruhu	s	+		PGsl
G290 ⁶⁾	G	Aktivování přepnutí na režim SINUMERIK	m	+		FBWsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
G291	G	Aktivování přepnutí na režim ISO2/3	m	+		FBWsl
G331	G	Vrtání závitů bez vyrovnávací hlavičky, kladné stoupání, směr otáčení vpravo	m	+		PGsl
G332	G	Vrtání závitů bez vyrovnávací hlavičky, záporné stoupání, směr otáčení vlevo	m	+		PGsl
G335	G	Soustružení sudovitého závitu ve směru hodinových ručiček	m	+		PGsl
G336	G	Soustružení sudovitého závitu proti směru hodinových ručiček	m	+		PGsl
G340 ⁶⁾	G	Prostorový najížděcí blok (hloubka a v rovině stejné – spirála)	m	+		PGsl
G341	G	Napřed přísuv v kolmé ose (z), pak najíždění v rovině	m	+		PGsl
G347	G	Měkké najíždění po půlkruhu	s	+		PGsl
G348	G	Měkké odjíždění po půlkruhu	s	+		PGsl
G450 ⁶⁾	G	Přechodový kruh	m	+		PGsl
G451	G	Průsečík ekvidistančních drah	m	+		PGsl
G460 ⁶⁾	G	Aktivování protikolizního monitorování pro blok najíždění a odjíždění	m	+		PGsl
G461	G	Vložení kruhu do bloku korekce rádiusu nástroje	m	+		PGsl
G462	G	Vložení přímky do bloku korekce rádiusu nástroje	m	+		PGsl
G500 ⁶⁾	G	Deaktivování všech nastavitelných framů, základní frame je aktivní	m	+		PGsl
G505 ... G599	G	5. ... 99. Nastavitelné posunutí počátku	m	+		PGsl
G601 ⁶⁾	G	Přechod na další blok při jemném přesném najetí	m	+		PGsl
G602	G	Přechod na další blok při hrubém přesném najetí	m	+		PGsl
G603	G	Přechod na další blok při konci bloku IPO	m	+		PGsl
G621	G	Snížení rychlosti na všech rozích	m	+		PGAsl
G641	G	Režim řízení pohybu po dráze s přechodovými zaobleními podle kritéria dráhy (= programovatelná vzdálenost zaoblení)	m	+		PGsl
G642	G	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí	m	+		PGsl
G643	G	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí (uvnitř bloku)	m	+		PGsl
G644	G	Režim řízení pohybu po dráze s přechodovými zaobleními s maximální možnou dynamikou	m	+		PGsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
G645	G	Režim řízení pohybu po dráze s přechodovými zaobleními v rozích a s tangenciálními přechody mezi bloky při dodržení definovaných tolerancí	m	+		PGsl
G700	G	Rozměrové údaje pro geometrické a technologické parametry v palcích (délky, posuv)	m	+	+	PGsl
G710 ⁶⁾	G	Rozměrové údaje pro geometrické a technologické parametry v metrických jednotkách (délky, posuv)	m	+	+	PGsl
G810 ⁶⁾ , ..., G819	G	G-skupina vyhrazená pro uživatele OEM		+		PGAsl
G820 ⁶⁾ , ..., G829	G	G-skupina vyhrazená pro uživatele OEM		+		PGAsl
G931	G	Zadání posuvu pomocí doby posuvu, konstantní rychlost po dráze vypnuta	m	+		
G942	G	Zmrazení lineárního posuvu a konstantní řezné rychlosti nebo otáček vřetena	m	+		
G952	G	Zmrazení otáčkového posuvu a konstantní řezné rychlosti nebo otáček vřetena	m	+		
G961	G	Lineární posuv (stejně jako u G94) a konstantní řezná rychlost	m	+		PGsl
G962	G	Lineární posuv nebo otáčkový posuv a konstantní řezná rychlost	m	+		PGsl
G971	G	Lineární posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	m	+		PGsl
G972	G	Lineární posuv nebo konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	m	+		PGsl
G973	G	Otáčkový posuv bez omezení otáček vřetena a konstantní otáčky vřetena (G97 bez LIMS pro režim ISO)	m	+		PGsl
GEOAX	P	Nové přiřazení geometrických os 1 - 3 kanálovým osám		+	-	PGAsl
GET	P	Výměna uvolněné osy mezi kanály		+	+	PGAsl
GETACTT	F	Stanovení aktivního nástroje ze skupiny stejnojmenných nástrojů.		+	-	FBWsl
GETACTTD	F	Stanovení odpovídajícího T-čísla k absolutnímu D-číslu		+	-	PGAsl
GETD	P	Výměna uvolněné osy mezi kanály		+	-	PGAsl
GETDNO	F	Zjištění D-čísla břitu (CE) daného nástroje (T)		+	-	PGAsl
GETEXET	P	Načtení T-čísla vyměňovaného nástroje		+	-	FBWsl
GETFREELOC	P	Vyhledání volného místa v zásobníku pro zadáný nástroj		+	-	FBWsl
GETSELT	P	Zjištění předem zvoleného T-čísla		+	-	FBWsl
GETT	F	Přiřazení T-čísla názvu nástroje		+	-	FBWsl
GETTCOR	F	Načtení délek nástroje, příp. složek délky nástroje		+	-	PGAsl
GETTENV	F	Načtení T-čísla, D-čísla a DL-čísla		+	-	PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
GETVARAP	F	Načtení přístupových oprávnění k systémové/uživatelské proměnné		+	-	PGAsl
GETVARDFT	F	Načtení standardní hodnoty systémové/uživatelské proměnné		+	-	PGAsl
GETVARLIM	F	Načtení mezní hodnoty systémové/uživatelské proměnné		+	-	PGAsl
GETVARPHU	F	Načtení fyzikální jednotky systémové/uživatelské proměnné		+	-	PGAsl
GETVARTYP	F	Načtení datového typu systémové/uživatelské proměnné		+	-	PGAsl
GFRAME0 ... GFRAME100	G	Aktivování framu pro broušení <n> ve správě dat v kanálu	m	+		PGsl
GOTO	K	Příkaz skoku napřed dopředu a potom zpátky (napřed ve směru konce programu a potom směrem k začátku programu)		+		PGAsl
GOTOB	K	Příkaz skoku směrem dozadu (směrem k začátku programu)		+		PGAsl
GOTOC	K	Stejně jako GOTO, ale s potlačením alarmu 14080 „Cíl skoku nenalezen“.		+		PGAsl
GOTOF	K	Příkaz skoku směrem dopředu (směrem ke konci programu)		+		PGAsl
GOTOS	K	Skok zpátky na začátek programu		+		PGAsl
GP	K	Klíčové slovo pro nepřímé programování atributů polohy		+		PGAsl
GROUP_ ADDEND	C (T)	Konec doplňkového bloku najíždění		+		PGAsl
GROUP_BEGIN	C (T)	Začátek programového bloku		+		PGAsl
GROUP_END	C (T)	Konec programového bloku		+		PGAsl
GWPSOF	P	Deaktivování konstantní obvodové rychlosti brusného kotouče	s	+	-	PGsl
GWPSON	P	Aktivování konstantní obvodové rychlosti brusného kotouče	s	+	-	PGsl
H...	A	Výstup pomocných funkcí do PLC		+	+	PGsl/FB1sl (H2)
HOLES1	C (T)	Řada děr		+		PGAsl
HOLES2	C (T)	Díry na kruhovém oblouku		+		PGAsl
I	A	Interpolační parametr	s	+		PGsl
I1	A	Souřadnice vnitřního bodu	s	+		PGsl
IC	K	Zadávání inkrementálních rozměrů	s	+		PGsl
ICYCOF	P	Všechny bloky technologického cyklu za příkazem ICYCOF zpracovávat v taktu IPO		+	+	FBSYsl
ICYCON	P	Každý blok technologického cyklu za příkazem ICYCON zpracovávat v samostatném taktu IPO		+	+	FBSYsl
ID	K	Identifikace pro modální synchronní akce	m	-	+	FBSYsl
IDS	K	Identifikace pro modální statické synchronní akce		-	+	FBSYsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
IF	K	Úvodní příkaz pro podmíněné skoky ve výrobním programu / technologickém cyklu		+	+	PGAsI
INDEX	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PGAsI
INICF	K	Inicializace proměnných při aktivování NE-WCONF		+		PGAsI
INIPO	K	Inicializace proměnných při zapnutí systému (Power-On)		+		PGAsI
INIRE	K	Inicializace proměnných při resetu		+		PGAsI
INIT	P	Aktivování určitého NC programu pro zpracovávání v určitém kanálu.		+	-	PGAsI
INITIAL		Vytvoření souboru INI pro všechny oblasti		+		PGAsI
INT	K	Datový typ: Celočíselné hodnoty se znaménkem		+		PGAsI
INTERSEC	F	Výpočet průsečíku mezi dvěma konturovými prvky		+	-	PGAsI
INVCCW	G	Pohyb po evolventě, proti směru hodinových ručiček	m	+		PGsI
INVCW	G	Pohyb po evolventě, ve směru hodinových ručiček	m	+		PGsI
INVFRAME	F	Výpočet inverzního framu z daného framu		+	-	FB1sI (K2)
IP	K	Proměnný interpolační parametr		+		PGAsI
IPOBRKA	P	Kritérium pro řízení pohybu od počátečního bodu skokové změny brzdné charakteristiky	m	+	+	
IPOENDA	K	Konec pohybu při dosažení "IPO Stop"	m	+		PGAsI
IPTRLOCK	P	Napevno nastavit začátek úseku programu, který může být prohledáván, na následující blok s funkcemi stroje	m	+	-	PGAsI
IPTRUNLOCK	P	Nastavení konce úseku programu, který může být prohledáván, na aktuální blok v okamžiku přerušení	m	+	-	PGAsI
IR	A	Souřadnice středu kruhu (směr X) při soustružení sudovitého závitu		+		PGsI
ISAXIS	F	Kontrola, zda geometrická osa 1 zadaná jako parametr existuje		+	-	PGAsI
ISD	A	Hloubka zajiždění nástroje	m	+		PGAsI
ISFILE	F	Kontrola, zda je daný soubor v uživatelské paměti NC k dispozici		+	-	PGAsI
ISNUMBER	F	Kontrola, zda je možno převést vstupní řetězec na číslo.		+	-	PGAsI
ISOCALL	K	Nepřímé volání programu naprogramovaného v jazyce ISO		+		PGAsI
ISVAR	F	Kontrola, zda předávaný parametr obsahuje proměnnou, která je NC systému známá.		+	-	PGAsI
J	A	Interpolační parametr	s	+		PGsI
J1	A	Souřadnice vnitřního bodu	s	+		PGsI

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
JERKA	P	Aktivování chování při zrychlení, které je nastaveno prostřednictvím MD, pro naprogramované osy		+	-	
JERKLIM	K	Snížení nebo zvýšení maximálního osového ryvu	m	+		PGAsl
JERKLIMA	K	Snížení nebo zvýšení maximálního osového ryvu	m	+	+	PGAsl
JR	A	Souřadnice středu kruhu (směr Y) při soustružení sudovitého závitu		+		PGsl
K	A	Interpolační parametr	s	+		PGsl
K1	A	Souřadnice vnitřního bodu	s	+		PGsl
KONT	G	Objíždění kontury s korekcí nástroje	m	+		PGsl
KONTC	G	Najíždění/odjíždění se spojitým polynomi- kým zakřivením	m	+		PGsl
KONTT	G	Najíždění/odjíždění se spojitým tangenciál- ním polynomm	m	+		PGsl
KR	A	Souřadnice středu kruhu (směr Z) při soustružení sudovitého závitu		+		PGsl
L	A	Číslo podprogramu	s	+	+	PGAsl
LEAD	A	Předozadní úhel 1. orientace nástroje 2. polynom orientace	m	+		PGAsl
LEADOF	P	Deaktivování osové vazby pomocí řídicí hod- noty		+	+	PGAsl
LEADON	P	Aktivování osové vazby pomocí řídicí hodnoty		+	+	PGAsl
LENTOAX	F	Vyvolání informací o přiřazení délek nástroje L1, L2 a L3 aktivního nástroje abscise, ordi- nátě a aplikátě.		+	-	PGAsl
LFOF ⁶⁾	G	Deaktivování rychlého zpětného pohybu při řezání závitů	m	+		PGsl
LFON	G	Aktivování rychlého zpětného pohybu při řezá- ní závitů	m	+		PGsl
LFPOS	G	Návrat osy stanovené příkazem POLFMASK nebo POLFLIN na absolutní pozici naprogramo- vanou pomocí příkazu POLF.	m	+		PGsl
LFTXT ⁶⁾	G	Rovina zpětného pohybu při rychlém pozved- nutí nástroje se bude určovat na základě teč- ny k dráze a momentálního směru nástroje.	m	+		PGsl
LFWP	G	Rovina zpětného pohybu při rychlém pozved- nutí nástroje je určena prostřednictvím aktuál- ní pracovní roviny (G17/G18/G19).	m	+		PGsl
LIFTFAST	K	Rychlé pozvednutí		+		PGsl
LIMS	K	Omezení otáček u příkazů G96/G961 a G97	m	+		PGsl
LLI	K	Spodní mezní hodnota pro proměnné		+		PGAsl
LN	F	Přirozený logaritmus		+	+	PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
LOCK	P	Zablokování synchronní akce s identifikací ID (zastavení technologického cyklu)		-	+	FBSYsl
LONGHOLE	C (T)	Podlouhlá díra		+		PGAsl
LOOP	K	Úvodní příkaz nekonečné smyčky		+		PGAsl

Příkazy M ... R

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
M0		Programovatelné zastavení		+	+	PGsl
M1		Volitelné zastavení		+	+	PGsl
M2		Konec hlavního programu (jako v případě M30)		+	+	PGsl
M3		Vřeteno se otáčí vpravo		+	+	PGsl
M4		Vřeteno se otáčí vlevo		+	+	PGsl
M5		zastavení vřetena		+	+	PGsl
M6		Výměna nástroje		+	+	PGsl
M17		Konec podprogramu		+	+	PGsl
M19		Nastavení pozice vřetena na polohu zadanou v SD 43240		+	+	PGsl
M30		Konec hlavního programu (jako v případě M2)		+	+	PGsl
M40		Automatické přepínání stupňů převodovky		+	+	PGsl
M41 ... M45		Stupně převodovky 1 ... 5		+	+	PGsl
M70		Přechod do osového režimu		+	+	PGsl
MASLDEF	P	Definice spojení os typu Master/Slave		+	+	PGAsl
MASLDEL	P	Zrušení spojení os typu Master/Slave a vymazání definice tohoto spojení		+	+	PGAsl
MASLOF	P	Deaktivování dočasného spojení		+	+	PGAsl
MASLOFS	P	Deaktivování dočasného spojení s automatickým zastavením osy typu Slave		+	+	PGAsl
MASLON	P	Aktivování dočasného spojení		+	+	PGAsl
MATCH	F	Vyhledávání řetězce v rámci jiného řetězce		+	-	PGAsl
MAXVAL	F	Větší z hodnot ve dvou proměnných (aritmetická funkce)		+	+	PGAsl
MCALL	K	Modální volání podprogramu		+		PGAsl
MEAC	K	Kontinuální měření jednou osou bez vymazání zbytkové dráhy	s	+	+	PGAsl
MEAFRAME	F	Výpočet framu na základě změřených bodů		+	-	PGAsl
MEAS	A	Měření s vymazáním zbytkové dráhy	s	+		PGAsl
MEASA	K	Měření jednou osou s vymazáním zbytkové dráhy	s	+	+	PGAsl
MEASURE	F	Metoda výpočtu pro měření obrobku a nástroje		+	-	FB1sl (M5)

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
MEAW	A	Měření bez mazání zbytkové dráhy	s	+		PGAsl
MEAWA	K	Měření jednou osou bez vymazání zbytkové dráhy	s	+	+	PGAsl
MI	K	Přístup k datům framu: Zrcadlové převrácení		+		PGAsl
MINDEX	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PGAsl
MINVAL	F	Menší z hodnot ve dvou proměnných (aritmetická funkce)		+	+	PGAsl
MIRROR	G	Programovatelné zrcadlové převrácení	s	+		PGAsl
MMC	P	Vyvolání interaktivního dialogového okna na HMI z výrobního programu		+	-	PGAsl
MOD	K	Dělení MODULO		+		PGAsl
MODAXVAL	F	Zjištění pozice modulo kruhové osy modulo		+	-	PGAsl
MOV	K	Spuštění polohovací osy		-	+	FBSYsl
MOVT	A	Zadání koncového bodu pohybu pracovním posuvem ve směru nástroje				FB1(K2)
MSG	P	Programovatelná hlášení	m	+	-	PGsl
MVTOOL	P	Příkaz jazyka pro pohyb nástroje		+	-	FBWsl
N	A	Číslo vedlejšího bloku v NC programu		+		PGsl
NAMETOINT	F	Zjištění indexu systémové proměnné		+		PGAsl
NC	K	Specifikace rozsahu platnosti dat		+		PGAsl
NEWCONF	P	Převzetí změněných strojních parametrů (odpovídá příkazu "Nastavit strojní parametr jako platný")		+	-	PGAsl
NEWMT	F	Založení nového modulu Multitool		+	-	FBWsl
NEWT	F	Založení nového nástroje		+	-	FBWsl
NORM ⁶⁾	G	Normální nastavení v počátečním a koncovém bodě při korekci nástroje	m	+		PGsl
NOT	K	Logické NE (negace)		+		PGAsl
NPROT	P	Zapnutí/vypnutí chráněné oblasti pro specifický stroj		+	-	PGAsl
NPROTDEF	P	Definice chráněné oblasti pro specifický stroj		+	-	PGAsl
NUMBER	F	Převedení vstupního řetězce na číslo		+	-	PGAsl
OEMIPO1	G	Interpolace OEM 1	m	+		PGAsl
OEMIPO2	G	Interpolace OEM 2	m	+		PGAsl
OF	K	Klíčové slovo v příkazu větvení CASE		+		PGAsl
OFFN	A	Přídavek rozměru pro naprogramovanou konturu	m	+		PGsl
OMA1	A	Adresa OEM 1	m	+		PGAsl
OMA2	A	Adresa OEM 2	m	+		PGAsl
OMA3	A	Adresa OEM 3	m	+		PGAsl
OMA4	A	Adresa OEM 4	m	+		PGAsl
OMA5	A	Adresa OEM 5	m	+		PGAsl
OR	K	Logický operátor, spojení typu NEBO		+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
ORIXES	G	Lineární interpolace os stroje nebo orientačních os	m	+		PGAsI
ORIXPOS	G	Úhel orientace pomocí virtuálních orientačních os s polohováním kruhové osy	m	+		PGAsI
ORIC ⁶⁾	G	Změny orientace na vnějších rozích jsou superponovány s vkládaným kruhovým blokem	m	+		PGAsI
ORICONCCW	G	Interpolace po ploše pláště kužele proti směru hodinových ručiček.	m	+		PGAsI/FB3sl (F3)
ORICONCW	G	Interpolace po ploše pláště kužele ve směru hodinových ručiček.	m	+		PGAsI/FB3sl (F4)
ORICONIO	G	Interpolace po ploše pláště kužele s udáním pomocné meziorientace	m	+		PGAsI/FB3sl (F4)
ORICONTO	G	Interpolace po ploše pláště kužele s tangenciálním přechodem (zadání koncové orientace)	m	+		PGAsI/FB3sl (F5)
ORICURVE	G	Interpolace orientace s udáním pohybu dvou kontaktních bodů nástroje	m	+		PGAsI/FB3sl (F6)
ORID	G	Změna orientace se bude provádět před kruhovým blokem.	m	+		PGAsI
ORIEULER ⁶⁾	G	Úhel orientace pomocí Eulerova úhlu	m	+		PGAsI
ORIMKS	G	Orientace nástroje v souřadném systému stroje	m	+		PGAsI
ORIPATH	G	Orientace nástroje vztažená na dráhu	m	+		PGAsI
ORIPATHS	G	Orientace nástroje vztažená na dráhu, zlom v průběhu orientace se vyhladí	m	+		PGAsI
ORIPANE	G	Interpolace v rovině (odpovídá ORIVECT) Interpolace pomocí největší kružnice koule	m	+		PGAsI
ORIRESET	P	Základní poloha orientace nástroje s až 3 orientačními osami		+	-	PGAsI
ORIROTA ⁶⁾	G	Úhel otočení vztažen ke směru otáčení zadanému absolutně	m	+		PGAsI
ORIROTC	G	Tangenciální vektor otočení k tečně dráhy	m	+		PGAsI
ORIROTR	G	Úhel otočení relativně vůči rovině mezi počáteční a koncovou orientací	m	+		PGAsI
ORIROTT	G	Úhel otočení relativně vůči změně vektoru orientace	m	+		PGAsI
ORIRPY	G	Úhel orientace prostřednictvím úhlu RPY (XYZ)	m	+		PGAsI
ORIRPY2	G	Úhel orientace pomocí úhlu RPY (ZYX)	m	+		PGAsI
ORIS	A	Změna orientace	m	+		PGAsI
ORISOF ⁶⁾	G	Vypnutí vyhlazování charakteristiky orientace	m	+		PGAsI
ORISOLH	F	Výpočet orientace		+		PGAsI
ORISON	G	Zapnutí vyhlazování charakteristiky orientace	m	+		PGAsI
ORIVECT ⁶⁾	G	Interpolace pomocí největší kružnice koule (identická s ORIPANE)	m	+		PGAsI

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
ORIVIRT1	G	Úhel orientace pomocí virtuálních orientačních os (definice 1)	m	+		PGAsl
ORIVIRT2	G	Úhel orientace pomocí virtuálních orientačních os (definice 1)	m	+		PGAsl
ORIWKS ⁶⁾	G	Orientace nástroje v souřadném systému obrobku	m	+		PGAsl
OS	K	Oscilační pohyb ZAP/VYP		+		PGAsl
OSB	K	Oscilace: Počáteční bod	m	+		FB1sl (P5)
OSC	G	Konstantní vyhlazení orientace nástroje	m	+		PGAsl
OSCILL	K	osa: 1 - 3 příslušná osa	m	+		PGAsl
OSCTRL	K	Možnosti oscilačního pohybu	m	+		PGAsl
OSD	G	Vyhlazování orientace nástroje zadáním délky zaoblení pomocí parametru SD	m	+		PGAsl
OSE	K	Koncový bod oscilačního pohybu	m	+		PGAsl
OSNSC	K	Oscilace: Počet vyjiskření	m	+		PGAsl
OSOF ⁶⁾	G	Vypnutí vyhlazování orientace nástroje	m	+		PGAsl
OSP1	K	Oscilace: levý bod obratu	m	+		PGAsl
OSP2	K	Oscilace: pravý bod obratu	m	+		PGAsl
OSS	G	Vyhlazování orientace nástroje na konci bloku	m	+		PGAsl
OSSE	G	Vyhlazování orientace nástroje na počátku a na konci bloku	m	+		PGAsl
OST	G	Vyhlazování orientace nástroje zadáním úhlové tolerance ve stupních pomocí parametru SD (maximální odchylka od naprogramovaného průběhu orientace)	m	+		PGAsl
OST1	K	Oscilace: stop v levém bodu obratu	m	+		PGAsl
OST2	K	Oscilace: stop v pravém bodu obratu	m	+		PGAsl
OTOL	A	Tolerance orientace pro funkce kompresoru, vyhlazení orientace a druhy zaoblování konturových přechodů	m	+		PGAsl
OVR	K	Korekce otáček	m	+		PGAsl
OVRA	K	Korekce otáček pro osu	m	+	+	PGAsl
OVRRAP	K	Korekce rychlého posuvu	m	+		PGAsl
P	A	Počet průchodů podprogramem		+		PGAsl
PAROT	G	Srovnání souřadného systému obrobku podle obrobku	m	+		PGsl
PAROTOF ⁶⁾	G	Vypnutí otáčení framu vztahující se na obrobek	m	+		PGsl
PCALL	K	Volání podprogramu s absolutním udáním cesty a předáváním parametrů		+		PGAsl
PDELAYOF	G	Deaktivování zpoždění při lisování	m	+		PGAsl
PDELAYON ⁶⁾	G	Aktivování zpoždění při lisování	m	+		PGAsl
PHI	K	Úhel otočení orientace okolo směrové osy kužele		+		PGAsl
PHU	K	Fyzikální jednotka proměnné		+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
PL	A	1. B-Spline: Vzdálenost uzlů 2. Polynomická interpolace: Délka intervalu parametru při polynomické interpolaci	s	+		PGAsI
PM	K	za minutu		+		PGsI
PO	K	Koeficient polynomu při polynomické interpolaci	s	+		PGAsI
POCKET3	C (T)	Frézování pravoúhlé kapsy		+		PGAsI
POCKET4	C (T)	Frézování kruhové kapsy		+		PGAsI
POLF	K	Poloha pro zpětný pohyb LIFTFAST	m	+		PGsI/PGAsI
POLFA	P	Poloha pro zpětný pohyb jednotlivými osami se spouštěním pomocí parametru \$AA_ESR_TRIGGER	m	+	+	PGsI
POLFMASK	P	Uvolnění os pro zpětný pohyb bez souvislostí mezi jednotlivými osami	m	+	-	PGsI
POLFMLIN	P	Uvolnění os pro zpětný pohyb s lineární souvislostí mezi jednotlivými osami	m	+	-	PGsI
POLY	G	Polynomická interpolace	m	+		PGAsI
POLYPATH	P	Polynomická interpolace může být vybrána pro skupiny os AXIS nebo VECT	m	+	-	PGAsI
PON	G	Zapnutí lisování	m	+		PGAsI
PONS	G	Zapnutí lisování v taktu IPO	m	+		PGAsI
POS	K	Polohování osy		+	+	PGsI
POSA	K	Polohování osy přes hranici bloku		+	+	PGsI
POSM	P	Polohování zásobníku		+	-	FBWsl
POSMT	P	Najetí modulem Moltitool v držáku nástroje na číslo místa		+	-	FBWsl
POSP	K	Polohování v úsecích (oscilace)		+		PGsI
POSRANGE	F	Zjištění, zda se požadovaná poloha osy, jejíž interpolace právě probíhá, nachází v okně okolo předem zadané referenční pozice.		+	+	FBSYsl
POT	F	Druhá mocnina (aritmetická funkce)		+	+	PGAsI
PR	K	na otáčku		+		PGsI
PREPRO	PA	Označení podprogramů s přípravou		+		PGAsI
PRESETON	P	Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod		+	+	PGAsI
PRESETONS	P	Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod		+	+	PGAsI
PRIO	K	Klíčové slovo pro nastavení priority při spravování přerušení		+		PGAsI
PRLOC	K	Inicializace proměnných po resetu pouze po lokální změně		+		PGAsI
PROC	K	První příkaz programu		+		PGAsI
PROTA	P	Vyžádání nového výpočtu protikolizního modelu		+		PGAsI

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
PROTD	F	Výpočet vzdálenosti mezi dvěma chráněnými oblastmi		+		PGAsl
PROTS	P	Nastavení stavu chráněné oblasti		+		PGAsl
PSI	K	Úhel kužele		+		PGAsl
PTP	G	Pohyb od bodu k bodu (pohyb PTP)	m	+		PGAsl
PTPG0	G	Pohyb od bodu k bodu jen při G0, jinak pohyb po dráze s CP	m	+		PGAsl
PTPWOC	G	Pohyb od bodu k bodu bez kompenzačních pohybů, které jsou způsobovány změnami orientace	m	+		PGAsl
PUNCHACC	P	Zrychlení při prostřihování závislé na dráze		+	-	PGAsl
PUTFTOC	P	Jemná korekce nástroje pro paralelní orovnávaní		+	-	PGAsl
PUTFTOCF	P	Jemná korekce nástroje v závislosti na funkci určené příkazem FCTDEF pro paralelní orovnávaní		+	-	PGAsl
PW	A	B-Spline, váha uzlového bodu	s	+		PGAsl
QU	K	Výstup rychlých doplňkových (pomocných) funkcí		+		PGsl
R...	A	Početní parametr, také jako nastavitelný adresový identifikátor a s numerickým rozšířením		+		PGAsl
RAC	K	Absolutní blokové programování rádiusů pro specifickou osu	s	+		PGsl
RDISABLE	P	Zablokování načítání		-	+	FBSYsl
READ	P	Načtení jednoho nebo více řádků ze zadaného souboru a uložení načtených informací do pole		+	-	PGAsl
REAL	K	Datový typ: Proměnná s plovoucí řádovou čárkou a se znaménkem (reálná čísla)		+		PGAsl
REDEF	K	Opětovná definice systémových proměnných, uživatelských proměnných a příkazů NC jazyka		+		PGAsl
RELEASE	P	Odblokování os stroje za účelem výměny os		+	+	PGAsl
REP	K	Klíčové slovo pro inicializaci všech prvků pole se stejnou hodnotou		+		PGAsl
REPEAT	K	Opakování programové smyčky		+		PGAsl
REPEATB	K	Opakování programového řádku		+		PGAsl
REPOSA	G	Zpětné najíždění na konturu lineárně všemi osami	s	+		PGAsl
REPOSH	G	Najetí zpět na konturu po půlkruhu	s	+		PGAsl
REPOSHA	G	Najetí zpět na konturu všemi osami; geometrické osy po půlkruhu	s	+		PGAsl
REPOSL	G	Najíždění na konturu po čtvrtkruhu	s	+		PGAsl
REPOSQ	G	Najetí zpět na konturu po čtvrtkruhu	s	+		PGAsl
REPOSQA	G	Najetí zpět na konturu lineárně všemi osami; geometrické osy po čtvrtkruhu	s	+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
RESETMON	P	Příkaz jazyka pro aktivování požadované hodnoty		+	-	FBWsl
RET	P	Konec podprogramu		+	+	PGAsl
RETB	P	Konec podprogramu		+	+	PGAsl
RIC	K	Relativní blokové programování rádiusů pro specifickou osu	s	+		PGsl
RINDEX	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PGAsl
RMB	G	Zpětné najíždění na začátek bloku	m	+		PGAsl
RMBBL	G	Zpětné najíždění na začátek bloku	s	+		PGAsl
RME	G	Zpětné najíždění na konec bloku	m	+		PGAsl
RMEBL	G	Zpětné najíždění na konec bloku	s	+		PGAsl
RMI ⁶⁾	G	Zpětné najíždění na místo přerušení	m	+		PGAsl
RMIBL ⁶⁾	G	Zpětné najíždění na místo přerušení	s	+		PGAsl
RMN	G	Zpětné najíždění na nejbližší blok s bodem dráhy	m	+		PGAsl
RMNBL	G	Zpětné najíždění na nejbližší blok s bodem dráhy	s	+		PGAsl
RND	A	Zaoblení rohů kontury	s	+		PGsl
RNDM	A	Modální zaoblení	m	+		PGsl
ROT	G	Programovatelné otočení	s	+		PGsl
ROTS	G	Programovatelná otáčení framu o prostorový úhel	s	+		PGsl
ROUND	F	Zaokrouhlení desetinných míst		+	+	PGAsl
ROUNDUP	F	Zaokrouhlování vstupní hodnoty		+	+	PGAsl
RP	A	Polární rádius	m/s	+		PGsl
RPL	A	Rotace v rovině	s	+		PGsl
RT	K	Parametr pro přístup k datům framu: Otočení		+		PGAsl
RTLIOF	G	G0 bez lineární interpolace (interpolace jednotlivých os)	m	+		PGsl
RTLION ⁶⁾	G	G0 s lineární interpolací	m	+		PGsl

Příkazy S ... Z

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
S	A	Otáčky vřetena (u G4, G96/G961 jiný význam)	m/s	+	+	PGsl
SAVE	PA	Atribut pro záchranu informací při vyvolávání podprogramů		+		PGAsl
SBLOF	P	Potlačení zpracovávání blok po bloku		+	-	PGAsl
SBLON	P	Odstranění blokování zpracovávání blok po bloku		+	-	PGAsl
SC	K	Parametr pro přístup k datům framu: Změna měřítka		+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
SCALE	G	Programovatelná změna měřítka	s	+		PGsl
SCC	K	Selektivní přiřazení příčné osy příkazu G96/G961/G962. Identifikátorem osy mohou být geometrická, kanálová nebo strojní osa.		+		PGsl
SCPARA	K	Programování bloku parametrů servomechanismu		+	+	PGAsl
SD	A	Stupeň splinu	s	+		PGAsl
SET	K	Klíčové slovo pro inicializaci všech prvků pole se zadanými hodnotami		+		PGAsl
SETAL	P	Aktivování alarmu		+	+	PGAsl
SETDNO	F	Přiřazení D-čísla bříty (CE) daného nástroje (T)		+	-	PGAsl
SETINT	K	Stanovení, která rutina přerušení má být aktivována, když se aktivuje daný vstup NC.		+		PGAsl
SETM	P	Nastavování značek ve vlastním kanálu		+	+	PGAsl
SETMS	P	Zpětné přepnutí na řídicí vřeteno určené ve strojním parametru		+	-	PGsl
SETMS(n)	P	Vřeteno n má platit jako řídicí vřeteno		+		PGsl
SETMTH	P	Definice čísla držáku hlavního nástroje		+	-	FBWsl
SETPIECE	P	Nastavení počtu kusů pro všechny nástroje, které jsou přiřazeny danému vřetenu.		+	-	FBWsl
SETTA	P	Aktivování nástroje ze skupiny opotřebení		+	-	FBWsl
SETTCOR	F	Změna komponent nástroje; při této změně jsou zohledňovány všechny okrajové podmínky		+	-	PGAsl
SETTIA	P	Deaktivování nástroje ze skupiny opotřebení		+	-	FBWsl
SF	A	Úhlové posunutí počátečního bodu pro řezání závitů	m	+		PGsl
SIN	F	Sinus (trigonometrická funkce)		+	+	PGAsl
SIRELAY	F	Aktivování bezpečnostních funkcí stanovených pomocí parametrů SIRELILN, SIRELOUT, SIRELTIME		-	+	FBSlsl
SIRELIN	P	Inicializace vstupních veličin funkčního modulu		+	-	FBSlsl
SIRELOUT	P	Inicializace výstupních veličin funkčního modulu		+	-	FBSlsl
SIRELTIME	P	Inicializace časovače funkčního modulu		+	-	FBSlsl
SLOT1	C (T)	Podélná drážka		+		PGAsl
SLOT2	C (T)	Kruhová drážka		+		PGAsl
SOFT	G	Zrychlení po dráze s omezením ryvu	m	+		PGAsl
SOFTA	P	Aktivování změn zrychlení pro naprogramované osy s omezením trhavých pohybů		+	-	PGAsl
SON	G	Aktivování prostřihování	m	+		PGAsl
SONS	G	Zapnutí prostřihování v taktu IPO	m	+		PGAsl
SPATH ⁶⁾	G	Referenční dráha pro osy v FGROUPE je délka oblouku	m	+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
SPCOF	P	Přepnutí řídicího vřetena nebo vřetena (n) z režimu polohové regulace do režimu regulace otáček	m	+	-	PGsl
SPCON	P	Přepnutí řídicího vřetena nebo vřetena (n) z režimu regulace otáček do režimu regulace polohy	m	+	-	PGAsl
SPI	F	Převedení čísla vřetena v identifikátoru osy		+	-	PGAsl
SPIF1 ⁶⁾	G	Rychlé vstupy/výstupy NC pro lisování/prostřihování, byte 1	m	+		FB2sl (N4)
SPIF2	G	Rychlé vstupy/výstupy NC pro lisování/prostřihování, byte 2	m	+		FB2sl (N4)
SPLINEPATH	P	Definice pásma hodnot pro spliny		+	-	PGAsl
SPN	A	Počet úseků na blok	s	+		PGAsl
SPOF ⁶⁾	G	Vypnutí zdvihu, vypnutí lisování, prostřihování	m	+		PGAsl
SPOS	K	Poloha vřetena	m	+	+	PGsl
SPOSA	K	Polohování vřetena přes hranice bloků	m	+		PGsl
SPP	A	Délka úseku	m	+		PGAsl
SPRINT	F	Vrací zpět vstupní řetězec formátovaný		+		PGAsl
SQRT	F	Druhá odmocnina (aritmetická funkce) (square root)		+	+	PGAsl
SR	A	Zpětná dráha oscilačního pohybu pro synchronní akci	s	+		PGsl
SRA	K	Zpětná dráha oscilačního pohybu osy při externím vstupu pro synchronní akci	m	+		PGsl
ST	A	Doba vyjiskřování s oscilačním pohybem pro synchronní akci	s	+		PGsl
STA	K	Doba vyjiskřování s oscilačním pohybem osy pro synchronní akci	m	+		PGsl
START	P	Spuštění zvoleného programu současně ve více kanálech z momentálně zpracovávaného programu		+	-	PGAsl
STARTFIFO ⁶⁾	G	Zpracovávat; souběžně s plněním paměti preprocesoru	m	+		PGAsl
STAT		Poloha kloubu	s	+		PGAsl
STOLF	K	Toleranční faktor G0	m	+		PGAsl
STOPFIFO	G	Zastavení zpracování; plnění paměti preprocesoru, dokud není zjištěn příkaz STARTFIFO, naplnění paměti preprocesoru nebo konec programu	m	+		PGAsl
STOPRE	P	Zastavení předběžného zpracování, dokud nejsou zpracovány všechny připravené bloky z hlavního zpracování programu.		+	-	PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
STOPREOF	P	Odblokování zastavení předběžného zpracování		-	+	FBSYsl
STRING	K	Datový typ: Řetězec znaků		+		PGAsl
STRINGIS	F	Kontrola existujícího rozsahu jazyka NC systému a speciálně u tohoto příkazu odpovídajících názvů NC-cyklů, uživatelských proměnných, maker a názvů návěští, zda existují, zda jsou platné, definované nebo aktivní.		+	-	PGAsl
STRLEN	F	Zjištění délky řetězce		+	-	PGAsl
SUBSTR	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PGAsl
SUPA	G	Potlačení aktuálního posunutí počátku, včetně programovatelných posunutí, systémových framů, posunutí ručním kolečkem (DRF), externích posunutí a superponovaných pohybů	s	+		PGsl
SVC	K	Řezná rychlost nástroje	m	+		PGsl
SYNFCT	P	Vyhodnocování polynomu v závislosti na podmínce v pohybové synchronní akci		-	+	FBSYsl
SYNR	K	Načtení proměnné se uskutečňuje synchronně, tzn. v okamžiku zpracovávání		+		PGAsl
SYNRW	K	Načtení a zápis do proměnné se uskutečňuje synchronně, tzn. v okamžiku zpracovávání		+		PGAsl
SYNW	K	Zápis do proměnné se uskutečňuje synchronně, tzn. v okamžiku zpracovávání		+		PGAsl
T	A	Vyvolání nástroje (výměna jen tehdy, je-li nastaveno strojním parametrem, jinak je třeba příkaz M6)		+		PGsl
TAN	F	Tangens (trigonometrická funkce)		+	+	PGAsl
TANG	P	Tangenciální řízení: Definice vazby		+	-	PGAsl
TANGDEL	P	Tangenciální řízení: Vymazání vazby		+	-	PGAsl
TANGOF	P	Tangenciální řízení: Deaktivování vazby		+	-	PGAsl
TANGON	P	Tangenciální řízení: Aktivování vazby		+	-	PGAsl
TCA (828D: _TCA)	P	Volba nástroje / výměna nástroje nezávisle na jeho stavu		+	-	FBWsl
TCARR	A	Vyžádání držáku nástroje (s číslem „m“)		+		PGAsl
TCI	P	Výměna nástroje ze schránky do zásobníku		+	-	FBWsl
TCOABS ⁶⁾	G	Stanovení délkových složek nástroje z aktuální orientace nástroje	m	+		PGAsl
TCOFR	G	Stanovení složek délky nástroje z orientace aktivního framu	m	+		PGAsl
TCOFRX	G	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose X	m	+		PGAsl
TCOFRY	G	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose Y	m	+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
TCOFRZ	G	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose Z	m	+		PGAsI
THETA	A	Úhel otočení	s	+		PGAsI
TILT	A	Úhel bočního naklonění	m	+		PGAsI
TLIFT	P	Tangenciální řízení: Aktivování vytváření pomocných bloků		+	-	PGAsI
TML	P	Volba nástroje spolu s jeho číslem místa v zásobníku		+	-	FBWsl
TMOF	P	Deaktivování monitorování nástroje		+	-	PGAsI
TMON	P	Aktivování monitorování nástroje		+	-	PGAsI
TO	K	Příkaz pro zadání koncové hodnoty ve smyčce FOR s počítadlem		+		PGAsI
TOFF	K	Offset délky nástroje ve směru délkové složky nástroje, která se uplatňuje rovnoběžně s geometrickou osou zadanou v indexu.	m	+		PGsl
TOFFL	K	Offset délky nástroje ve směru délkové složky nástroje L1, L2, příp. L3.	m	+		PGsl
TOFFOF	P	Vynulování on-line korekce délky nástroje		+	-	PGAsI
TOFFON	P	Aktivování on-line korekce délky nástroje		+	-	PGAsI
TOFFR	A	Offset radiusu nástroje	m	+		PGsl
TOFRAME	G	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PGsl
TOFRAMEX	G	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PGsl
TOFRAMEY	G	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PGsl
TOFRAMEZ	G	Stejně jako příkaz TOFRAME	m	+		PGsl
TOLOWER	F	Přeměna všech písmen v řetězci na malá písmena		+	-	PGAsI
TOOLENV	F	Uložení do paměti všech aktuálních stavů, které jsou důležité pro vyhodnocování parametrů nástroje uchovávaných v paměti.		+	-	PGAsI
TOOLGNT	F	Zjištění počtu nástrojů ve skupině nástrojů		+	-	FBWsl
TOOLGT	F	Zjištění T-čísla nástroje z dané skupiny nástrojů		+	-	FBWsl
TOROT	G	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PGsl
TOROTOF ⁶⁾	G	Zrušení otočení framu ve směru nástroje	m	+		PGsl
TOROTX	G	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PGsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
TOROTY	G	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PGsl
TOROTZ	G	Stejně jako příkaz TOROT	m	+		PGsl
TOUPPER	F	Přeměna všech písmen v řetězci na velká písmena		+	-	PGAsl
TOWBCS	G	Hodnoty opotřebení v základním souřadném systému (BCS)	m	+		PGAsl
TOWKCS	G	Hodnoty opotřebení v souřadném systému hlavy nástroje při kinetické transformaci (liší se od MCS otočením nástroje)	m	+		PGAsl
TOWMCS	G	Hodnoty opotřebení v souřadném systému stroje (MCS)	m	+		PGAsl
TOWSTD ⁶⁾	G	Základní nastavení pro korekce hodnoty ve směru délky nástroje	m	+		PGAsl
TOWTCS	G	Hodnoty opotřebení v souřadném systému nástroje (vztažný bod držáku nástroje T na nástrojovém sklíčidle)	m	+		PGAsl
TOWWCS	G	Hodnoty opotřebení v souřadném systému obrobku (WCS)	m	+		PGAsl
TR	K	Složka posunutí proměnné typu FRAME		+		PGAsl
TRAANG	P	Transformace šikmé osy		+	-	PGAsl
TRACON	P	Kaskádová transformace		+	-	PGAsl
TRACYL	P	Válec: Transformace plášťové plochy		+	-	PGAsl
TRAFOOF	P	Vypnutí aktivních transformací v kanálu		+	-	PGAsl
TRAFOON	P	Aktivování transformace definované prostřednictvím kinematického řetězce		+	-	PGAsl
TRAILOF	P	Vypnutí asynchronního vlečení		+	+	PGAsl
TRAILON	P	Zapnutí asynchronního vlečení		+	+	PGAsl
TRANS	G	Absolutní programovatelné posunutí počátku	s	+		PGsl
TRANSMIT	P	Polární transformace (obrábění šikmých ploch)		+	-	PGAsl
TRAORI	P	4-, 5-osá transformace, generická transformace		+	-	PGAsl
TRUE	K	Logická konstanta: TRUE		+		PGAsl
TRUNC	F	Odříznutí míst za desetinnou čárkou		+	+	PGAsl
TU		Úhel osy	s	+		PGAsl
TURN	A	Počet závitů u spirály	s	+		PGsl
ULI	K	Horní mezní hodnota pro proměnné		+		PGAsl
UNLOCK	P	Odblokování synchronní akce s identifikací ID (pokračování technologického cyklu)		-	+	FBSYsl
UNTIL	K	Podmínka pro ukončení smyčky REPEAT		+		PGAsl
UPATH	G	Referenční dráha pro osy v FGROUP je křivkový parametr	m	+		PGAsl
VAR	K	Klíčové slovo: Druh předávání parametrů		+		PGAsl

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
1) 2) 3) 4) 5) Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
VELOLIM	K	Omezení maximální rychlosti osy	m	+		PGAsI
VELOLIMA	K	Snížení nebo zvýšení maximální rychlosti vlečné osy	m	+	+	PGAsI
WAITC	P	Čekání, až bude pro osu/vřeteno splněno kritérium pro změnu bloku vazby.		+	-	PGAsI
WAITE	P	Čekání na konec programu v jiném kanálu		+	-	PGAsI
WAITENC	P	Čekání na synchronizované, příp. restaurované polohy os		+	-	PGAsI
WAITM	P	Čekání na značku v uvedeném kanálu; konec předešlého bloku s přesným najetím.		+	-	PGAsI
WAITMC	P	Čekání na značku v uvedeném kanálu; přesné najetí jen tehdy, pokud ostatní kanály značky ještě nedosáhly.		+	-	PGAsI
WAITP	P	Čekání na konec posuvu polohovací osy		+	-	PGsI
WAITS	P	Čekání na dosažení pozice vřetena		+	-	PGsI
WALCS0 ⁶⁾	G	Deaktivování ohraničení pracovního pole ve WCS	m	+	-	PGsI
WALCS1	G	Skupina 1 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS2	G	Skupina 2 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS3	G	Skupina 3 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS4	G	Skupina 4 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS5	G	Skupina 5 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS6	G	Skupina 6 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS7	G	Skupina 7 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS8	G	Skupina 8 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS9	G	Skupina 9 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALCS10	G	Skupina 10 ohraničení pracovního pole ve WCS aktivní	m	+	-	PGsI
WALIMOF	G	Vypnutí ohraničení pracovního pole BCS	m	+	-	PGsI
WALIMON ⁶⁾	G	Zapnutí ohraničení pracovního pole BCS	m	+	-	PGsI
WHEN	K	Pokud je podmínka jednou splněna, daná akce se jednou uskuteční.		-	+	FBSYsI
WHENEVER	K	Pokud je podmínka splněna, v každém taktu interpolátoru se bude daná akce cyklicky provádět.		-	+	FBSYsI
WHILE	K	Začátek programové smyčky WHILE		+		PGAsI

Příkaz	Typ ¹⁾	Význam	W ²⁾	TP ³⁾	SA ⁴⁾	Popis viz ⁵⁾
^{1) 2) 3) 4) 5)} Pokud budete potřebovat vysvětlení, viz Legenda (Strana 884).						
WRITE	P	Zápis textu do systému souborů. Vložení bloku na konec zadaného souboru.		+	-	PGsl
WRTPR	P	Zápis řetězce znaků do proměnné BTSS		+	-	PGsl
X	A	Název osy	m/s	+	+	PGsl
XOR	O	Logické XOR		+		PGAsl
Y	A	Název osy	m/s	+	+	PGsl
Z	A	Název osy	m/s	+	+	PGsl

Legenda

1) Druh příkazu:

A Adresa

Identifikátor, kterému je hodnota přiřazována (např. OVR=10). Existují také některé adresy, které aktivují nebo deaktivují funkci i bez přiřazení nějaké hodnoty (např. CPLON a CPLOF).

C (A) Cyklus AST

Předem definovaný NC program pro automatickou dodatečnou optimalizaci s AST (= automatická optimalizace servomechanismu) Přizpůsobování podmínkám konkrétní situace při optimalizaci se uskutečňuje prostřednictvím parametrů, které jsou předávány při volání cyklu.

C (M) Měřicí cyklus

Předem definovaný NC-program, ve kterém je naprogramována určitá všeobecně platná měřicí operace, jako např. určování vnitřního průměru válcového obrobku. Přizpůsobování podmínkám konkrétní situace při měření se uskutečňuje prostřednictvím parametrů, které jsou předávány při volání cyklu.

C (T) Technologický cyklus

Předem definovaný NC program, ve kterém je naprogramována určitá obráběcí operace, jako je například vrtání závitu nebo frézování kapsy tak, aby byl všeobecně platný. Přizpůsobování podmínkám konkrétní situace při obrábění se uskutečňuje prostřednictvím parametrů, které jsou předávány při volání cyklu.

F Předdefinovaná funkce (poskytuje výslednou (vracenou) hodnotu)

Volání předem definované funkce se může nacházet i jako operand ve výrazu.

G G-příkaz

G-příkazy jsou rozděleny do skupin. V jednom bloku může být napsán jen jeden příkaz G-funkce z určité skupiny. Příkaz G-funkce může mít modální platnost (do odvolání jiným příkazem ze stejné skupiny) nebo platí jen pro blok, ve kterém se nachází (bloková působnost).

K Klíčové slovo

Identifikátor, který určuje syntaxi daného bloku. Klíčovému slovu nemůže být přiřazena žádná hodnota a pomocí klíčového slova nemůže být aktivována/deaktivována žádná funkce NC systému.

Příklady: Řídící struktury (IF, ELSE, ENDIF, WHEN, ...), zpracování programu (GOTO, GOTO, RET ...)

O Operátor

Operátor pro matematické, porovnávací nebo logické operace

P Předdefinovaná procedura (neposkytuje žádnou výslednou (vracenou) hodnotu)

PA Atribut programu

Atributy programu se nacházejí na konci definičního řádku podprogramu:

PROC <název programu> (...) <Atribut programu>

Určují chování při zpracovávání podprogramu.

2) Platnost příkazu:

m modální

s bloková

- 3) Možnosti programování ve výrobním programu:
- + programovatelné
 - nelze naprogramovat
 - M Může být naprogramováno jen výrobcem stroje
- 4) Lze naprogramovat v synchronních akcích:
- + programovatelné
 - nelze naprogramovat
 - T Lze naprogramovat jen v technologických cyklech
- 5) Odkaz na dokument, který obsahuje podrobný popis příkazu:
- | | |
|------------------|--|
| <i>PGsl</i> | Programovací příručka, Základy |
| <i>PGAsl</i> | Programovací příručka, Pro pokročilé |
| <i>BNMsl</i> | Programovací příručka, Měřicí cykly |
| <i>BHDsl</i> | Příručka pro obsluhu, Soustružení |
| <i>BHFsl</i> | Příručka pro obsluhu, Frézování |
| <i>FB1sl ()</i> | Příručka k funkcím, Základní funkce (s alfanumerickou zkratkou popisu příslušné funkce v závorkách) |
| <i>FB2sl ()</i> | Příručka k funkcím, Rozšiřovací funkce (s alfanumerickou zkratkou popisu příslušné funkce v závorkách) |
| <i>FB3sl ()</i> | Příručka k funkcím, Speciální funkce (s alfanumerickou zkratkou popisu příslušné funkce v závorkách) |
| <i>FBSlsl</i> | Příručka k funkcím, Safety Integrated |
| <i>FBSYsl</i> | Příručka k funkcím, Synchronní akce |
| <i>FBWsl</i> | Příručka k funkcím, Správa nástrojů |
- 6) Standardní nastavení na začátku programu (stav systému při dodávce řídicího systému, pokud není naprogramováno nic jiného).

Obrázek 21-1 Význam poznámek pod čarou v tabulkách příkazů

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Poznámka

Cykly

Cykly byly označeny jako „volitelné“, jestliže jsou připojeny k následujícím volitelným doplňkům, pro které je nutno mít licenci:

- Rozšířené technologické funkce (číslo produktu: 6FC5800-0AP58-0YB0)
- Měřicí cykly (číslo produktu: 6FC5800-0AP28-0YB0)
- Měření kinematiky (číslo produktu: 6FC5800-0AP18-0YB0)
- SINUMERIK Schleifen Advanced (Pokročilé broušení) (číslo produktu: 6FC5800-0AS35-0YB0)

Nebylo nijak označeno, pokud je v cyklech kvůli volitelnému doplňku „Rozšířené technologické funkce“ na určitých systémech obsažena pouze část funkcí.

21.2.1 Varianty řídicího systému pro frézování / soustružení

Příkazy A ... C

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
:	•	•	•	•	•	•	•	•
*	•	•	•	•	•	•	•	•
+	•	•	•	•	•	•	•	•
-	•	•	•	•	•	•	•	•
<	•	•	•	•	•	•	•	•
<<	•	•	•	•	•	•	•	•
<=	•	•	•	•	•	•	•	•
=	•	•	•	•	•	•	•	•
>=	•	•	•	•	•	•	•	•
/	•	•	•	•	•	•	•	•
/0 ... /7	•	•	•	•	•	•	•	•
A	•	•	•	•	•	•	•	•
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-
A5	-	-	-	-	-	-	-	-

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
• Standardní								
○ Volitelný do- plněk								
- není k dispozici								
A6	-	-	-	-	-	-	-	-
A7	-	-	-	-	-	-	-	-
ABS	•	•	•	•	•	•	•	•
AC	•	•	•	•	•	•	•	•
ACC	•	•	•	•	•	•	•	•
ACCLIMA	•	•	•	•	•	•	•	•
ACN	•	•	•	•	•	•	•	•
ACOS	•	•	•	•	•	•	•	•
ACP	•	•	•	•	•	•	•	•
ACTBLOCNO	•	•	•	•	•	•	•	•
ADDFRAME	•	•	•	•	•	•	•	•
ADIS	•	•	•	•	•	•	•	•
ADISPOS	•	•	•	•	•	•	•	•
ADISPOSA	•	•	•	•	•	•	•	•
ALF	•	•	•	•	•	•	•	•
AMIRROR	•	•	•	•	•	•	•	•
AND	•	•	•	•	•	•	•	•
ANG	•	•	•	•	•	•	•	•
AP	•	•	•	•	•	•	•	•
APR	•	•	•	•	•	•	•	•
APRB	•	•	•	•	•	•	•	•
APRP	•	•	•	•	•	•	•	•
APW	•	•	•	•	•	•	•	•
APWB	•	•	•	•	•	•	•	•
APWP	•	•	•	•	•	•	•	•
APX	•	•	•	•	•	•	•	•
AR	•	•	•	•	•	•	•	•
AROT	•	•	•	•	•	•	•	•
AROTS	•	•	•	•	•	•	•	•
AS	•	•	•	•	•	•	•	•
ASCALE	•	•	•	•	•	•	•	•
ASIN	•	•	•	•	•	•	•	•
ASPLINE	○	○	○	○	○	○	○	○
ATAN2	•	•	•	•	•	•	•	•
ATOL	•	•	•	•	•	•	•	•
ATRANS	•	•	•	•	•	•	•	•
AUXFUDEL	•	•	•	•	•	•	•	•
AUXFUDELG	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
AUXFUMSEQ	•	•	•	•	•	•	•	•
AUXFUSYNC	•	•	•	•	•	•	•	•
AX	•	•	•	•	•	•	•	•
AXCTSWE	-	-	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-	-	-
AXIS	•	•	•	•	•	•	•	•
AXNAME	•	•	•	•	•	•	•	•
AXSTRING	•	•	•	•	•	•	•	•
AXTOCHAN	•	•	•	•	•	•	•	•
AXTOSPI	•	•	•	•	•	•	•	•
B	•	•	•	•	•	•	•	•
B2	-	-	-	-	-	-	-	-
B3	-	-	-	-	-	-	-	-
B4	-	-	-	-	-	-	-	-
B5	-	-	-	-	-	-	-	-
B6	-	-	-	-	-	-	-	-
B7	-	-	-	-	-	-	-	-
B_AND	•	•	•	•	•	•	•	•
B_OR	•	•	•	•	•	•	•	•
B_NOT	•	•	•	•	•	•	•	•
B_XOR	•	•	•	•	•	•	•	•
BAUTO	○	○	○	○	○	○	○	○
BLOCK	•	•	•	•	•	•	•	•
BLSYNC	•	•	•	•	•	•	•	•
BNAT	○	○	○	○	○	○	○	○
BOOL	•	•	•	•	•	•	•	•
BOUND	•	•	•	•	•	•	•	•
BRISK	•	•	•	•	•	•	•	•
BRISKA	•	•	•	•	•	•	•	•
BSPLINE	○	○	○	○	○	○	○	○
BTAN	○	○	○	○	○	○	○	○
C	•	•	•	•	•	•	•	•
C2	-	-	-	-	-	-	Název kanálové osy	Název kanálové osy
C3	-	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	-
C5	-	-	-	-	-	-	-	-

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
• Standardní ○ Volitelný do- plněk - není k dispozici								
C6	-	-	-	-	-	-	-	-
C7	-	-	-	-	-	-	-	-
CAC	•	•	•	•	•	•	•	•
CACN	•	•	•	•	•	•	•	•
CACP	•	•	•	•	•	•	•	•
CALCDAT	•	•	•	•	•	•	•	•
CALCPOSI	•	•	•	•	•	•	•	•
CALL	•	•	•	•	•	•	•	•
CALLPATH	•	•	•	•	•	•	•	•
CANCEL	•	•	•	•	•	•	•	•
CASE	•	•	•	•	•	•	•	•
CDC	•	•	•	•	•	•	•	•
CDOF	-	-	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-	-	-
CDON	-	-	-	-	-	-	-	-
CFC	•	•	•	•	•	•	•	•
CFIN	•	•	•	•	•	•	•	•
CFINE	•	•	•	•	•	•	•	•
CFTCP	•	•	•	•	•	•	•	•
CHAN	•	•	•	•	•	•	•	•
CHANDATA	•	•	•	•	•	•	•	•
CHAR	•	•	•	•	•	•	•	•
CHF	•	•	•	•	•	•	•	•
CHKDM	•	•	•	•	•	•	•	•
CHKDNO	•	•	•	•	•	•	•	•
CHR	•	•	•	•	•	•	•	•
CIC	•	•	•	•	•	•	•	•
CIP	•	•	•	•	•	•	•	•
CLEARM	-	-	-	-	-	○	-	•
CLRINT	•	•	•	•	•	•	•	•
CMIRROR	•	•	•	•	•	•	•	•
COARSEA	•	•	•	•	•	•	•	•
COLLPAIR	-	-	-	-	-	-	-	-
COMPCAD	•	-	•	-	•	•	-	-
COMPCURV	•	-	•	-	•	•	-	-
COMPLETE	•	•	•	•	•	•	•	•
COMPOF	•	-	•	-	•	•	-	-
COMPON	•	-	•	-	•	•	-	-

Příkaz	SINUMERIK 828D							
● Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
COMPSURF	-	-	○	-	○	○	-	-
CONTDCON	●	●	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●	●	●
CORROF	●	●	●	●	●	●	●	●
CORRTRAFO	-	-	-	-	-	-	-	-
COS	●	●	●	●	●	●	●	●
COUPDEF	-	○	-	○	-	-	○	○
COUPDEL	-	○	-	○	-	-	○	○
COUPOF	-	○	-	○	-	-	○	○
COUPOFS	-	○	-	○	-	-	○	○
COUPON	-	○	-	○	-	-	○	○
COUPONC	-	○	-	○	-	-	○	○
COUPRES	-	○	-	○	-	-	○	○
CP	●	●	●	●	●	●	●	●
CPBC	○	○	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○	○	○
CPLA	○	○	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○	○	○
CPLON	○	○	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○	○	○

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
CPMPRT	○	○	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○	○	○
CPOF	○	○	○	○	○	○	○	○
CPON	○	○	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●	●	●
CPRES	○	○	○	○	○	○	○	○
CPROT	●	●	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○	○	○
CPSYNCOF	○	○	○	○	○	○	○	○
CPSYNCOF2	○	○	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○	○	○
CR	●	●	●	●	●	●	●	●
CROT	●	●	●	●	●	●	●	●
CROTS	●	●	●	●	●	●	●	●
CRPL	●	●	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○	○	○
CT	●	●	●	●	●	●	●	●
CTAB	-	-	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-	-	-
CTABID	-	-	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-	-	-

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
CTABMPOL	-	-	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-	-	-
CTOL	•	•	•	•	•	•	•	•
CTRANS	•	•	•	•	•	•	•	•
CUT2D	•	•	•	•	•	•	•	•
CUT2DD	•	•	•	•	•	•	•	•
CUT2DF	•	•	•	•	•	•	•	•
CUT2DFD	•	•	•	•	•	•	•	•
CUT3DC	-	-	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-	-	-
CUTCONOF	•	•	•	•	•	•	•	•
CUTCONON	•	•	•	•	•	•	•	•
CUTMOD	•	•	•	•	•	•	•	•
CUTMODK	-	-	-	-	-	-	-	-
CYCLE60	○	○	•	•	•	•	•	•
CYCLE61	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
CYCLE62	•	•	•	•	•	•	•	•
CYCLE63	○	○	•	•	•	•	•	•
CYCLE64	○	○	•	•	•	•	•	•
CYCLE70	○	○	•	•	•	•	•	•
CYCLE72	•	•	•	•	•	•	•	•
CYCLE76	•	•	•	•	•	•	•	•
CYCLE77	•	•	•	•	•	•	•	•
CYCLE78	○	○	•	•	•	•	•	•
CYCLE79	○	○	•	•	•	•	•	•
CYCLE81	•	•	•	•	•	•	•	•
CYCLE82	•	•	•	•	•	•	•	•
CYCLE83	•	•	•	•	•	•	•	•
CYCLE84	•	•	•	•	•	•	•	•
CYCLE85	•	•	•	•	•	•	•	•
CYCLE86	•	•	•	•	•	•	•	•
CYCLE92	•	•	•	•	•	•	•	•
CYCLE95	•	•	•	•	•	•	•	•
CYCLE98	•	•	•	•	•	•	•	•
CYCLE99	•	•	•	•	•	•	•	•
CYCLE150	○	○	○	○	○	○	○	○
CYCLE435	-	-	-	-	-	-	-	-
CYCLE495	-	-	-	-	-	-	-	-
CYCLE750	•	•	•	•	•	•	•	•
CYCLE751	•	•	•	•	•	•	•	•
CYCLE752	•	•	•	•	•	•	•	•
CYCLE753	•	•	•	•	•	•	•	•
CYCLE754	•	•	•	•	•	•	•	•
CYCLE755	•	•	•	•	•	•	•	•
CYCLE756	•	•	•	•	•	•	•	•
CYCLE757	•	•	•	•	•	•	•	•
CYCLE758	•	•	•	•	•	•	•	•
CYCLE759	•	•	•	•	•	•	•	•
CYCLE800	-	-	•	•	•	•	•	•
CYCLE801	•	•	•	•	•	•	•	•
CYCLE802	•	•	•	•	•	•	•	•
CYCLE830	○	○	•	•	•	•	•	•
CYCLE832	•	•	•	•	•	•	•	•
CYCLE840	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
CYCLE899	○	○	•	•	•	•	•	•
CYCLE930	•	•	•	•	•	•	•	•
CYCLE940	•	•	•	•	•	•	•	•
CYCLE951	•	•	•	•	•	•	•	•
CYCLE952	○	○	•	•	•	•	•	•
CYCLE961	○	○	○	○	○	○	○	○
CYCLE971	○	○	○	○	○	○	○	○
CYCLE973	○	○	○	○	○	○	○	○
CYCLE974	○	○	○	○	○	○	○	○
CYCLE976	○	○	○	○	○	○	○	○
CYCLE977	○	○	○	○	○	○	○	○
CYCLE978	○	○	○	○	○	○	○	○
CYCLE979	○	○	○	○	○	○	○	○
CYCLE982	○	○	○	○	○	○	○	○
CYCLE994	○	○	○	○	○	○	○	○
CYCLE995	○	○	○	○	○	○	○	○
CYCLE996	○	-	○	-	○	○	-	-
CYCLE997	○	○	○	○	○	○	○	○
CYCLE998	○	○	○	○	○	○	○	○
CYCLE4071	-	-	-	-	-	-	-	-
CYCLE4072	-	-	-	-	-	-	-	-
CYCLE4073	-	-	-	-	-	-	-	-
CYCLE4074	-	-	-	-	-	-	-	-
CYCLE4075	-	-	-	-	-	-	-	-
CYCLE4077	-	-	-	-	-	-	-	-
CYCLE4078	-	-	-	-	-	-	-	-
CYCLE4079	-	-	-	-	-	-	-	-

Příkazy D ... F

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
D	•	•	•	•	•	•	•	•
D0	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
DAC	•	•	•	•	•	•	•	•
DC	•	•	•	•	•	•	•	•
DCI	•	•	•	•	•	•	•	•
DCM	•	•	•	•	•	•	•	•
DCU	•	•	•	•	•	•	•	•
DEF	•	•	•	•	•	•	•	•
DEFINE	•	•	•	•	•	•	•	•
DEFAULT	•	•	•	•	•	•	•	•
DELAYFSTON	•	•	•	•	•	•	•	•
DELAYFSTOF	•	•	•	•	•	•	•	•
DELDL	•	•	•	•	•	•	•	•
DELDTG	•	•	•	•	•	•	•	•
DELETE	•	•	•	•	•	•	•	•
DELMLOWNER	•	•	•	•	•	•	•	•
DELMRES	•	•	•	•	•	•	•	•
DELMT	•	•	•	•	•	•	•	•
DELOBJ	-	-	-	-	-	-	-	-
DELT	•	•	•	•	•	•	•	•
DELTC	•	•	•	•	•	•	•	•
DELTOOLNV	•	•	•	•	•	•	•	•
DIACYCOFA	•	•	•	•	•	•	•	•
DIAM90	•	•	•	•	•	•	•	•
DIAM90A	•	•	•	•	•	•	•	•
DIAMCHAN	•	•	•	•	•	•	•	•
DIAMCHANA	•	•	•	•	•	•	•	•
DIAMCYCOF	•	•	•	•	•	•	•	•
DIAMOF	•	•	•	•	•	•	•	•
DIAMOFA	•	•	•	•	•	•	•	•
DIAMON	•	•	•	•	•	•	•	•
DIAMONA	•	•	•	•	•	•	•	•
DIC	•	•	•	•	•	•	•	•
DILF	•	•	•	•	•	•	•	•
DISABLE	•	•	•	•	•	•	•	•
DISC	•	•	•	•	•	•	•	•
DISCL	•	•	•	•	•	•	•	•
DISPLOF	•	•	•	•	•	•	•	•
DISPLON	•	•	•	•	•	•	•	•
DISPR	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
DISR	•	•	•	•	•	•	•	•
DISRP	•	•	•	•	•	•	•	•
DITE	•	•	•	•	•	•	•	•
DITS	•	•	•	•	•	•	•	•
DIV	•	•	•	•	•	•	•	•
DL	-	-	-	-	-	-	-	-
DO	•	•	•	•	•	•	•	•
DRFOF	•	•	•	•	•	•	•	•
DRIVE	•	•	•	•	•	•	•	•
DRIVEA	•	•	•	•	•	•	•	•
DYNFINISH	•	•	•	•	•	•	•	•
DYNNORM	•	•	•	•	•	•	•	•
DYNPOS	•	•	•	•	•	•	•	•
DYNROUGH	•	•	•	•	•	•	•	•
DYNSEMIFIN	•	•	•	•	•	•	•	•
DZERO	•	•	•	•	•	•	•	•
EAUTO	○	○	○	○	○	○	○	○
EGDEF	-	○	-	○	-	-	○	○
EGDEL	-	○	-	○	-	-	○	○
EGOFC	-	○	-	○	-	-	○	○
EGOFS	-	○	-	○	-	-	○	○
EGON	-	○	-	○	-	-	○	○
EGONSYN	-	○	-	○	-	-	○	○
EGONSYNE	-	○	-	○	-	-	○	○
ELSE	•	•	•	•	•	•	•	•
ENABLE	•	•	•	•	•	•	•	•
ENAT	○	○	○	○	○	○	○	○
ENDFOR	•	•	•	•	•	•	•	•
ENDIF	•	•	•	•	•	•	•	•
ENDLABEL	•	•	•	•	•	•	•	•
ENDLOOP	•	•	•	•	•	•	•	•
ENDPROC	•	•	•	•	•	•	•	•
ENDWHILE	•	•	•	•	•	•	•	•
ESRR	○	○	○	○	○	○	○	○
ESRS	○	○	○	○	○	○	○	○
ETAN	○	○	○	○	○	○	○	○
EVERY	•	•	•	•	•	•	•	•
EX	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
EXECSTRING	•	•	•	•	•	•	•	•
EXECTAB	•	•	•	•	•	•	•	•
EXECUTE	•	•	•	•	•	•	•	•
EXP	•	•	•	•	•	•	•	•
EXTCALL	•	•	•	•	•	•	•	•
EXTCLOSE	•	•	•	•	•	•	•	•
EXTERN	•	•	•	•	•	•	•	•
EXTOPEN	•	•	•	•	•	•	•	•
F	•	•	•	•	•	•	•	•
FA	•	•	•	•	•	•	•	•
FAD	•	•	•	•	•	•	•	•
FALSE	•	•	•	•	•	•	•	•
FB	•	•	•	•	•	•	•	•
FCTDEF	•	•	•	•	•	•	•	•
FCUB	•	•	•	•	•	•	•	•
FD	•	•	•	•	•	•	•	•
FDA	•	•	•	•	•	•	•	•
FENDNORM	•	•	•	•	•	•	•	•
FFWOF	•	•	•	•	•	•	•	•
FFWON	•	•	•	•	•	•	•	•
FGREF	•	•	•	•	•	•	•	•
FGROUP	•	•	•	•	•	•	•	•
FI	•	•	•	•	•	•	•	•
FIFOCTRL	•	•	•	•	•	•	•	•
FILEDATE	•	•	•	•	•	•	•	•
FILEINFO	•	•	•	•	•	•	•	•
FILESIZE	•	•	•	•	•	•	•	•
FILESTAT	•	•	•	•	•	•	•	•
FILETIME	•	•	•	•	•	•	•	•
FINEA	•	•	•	•	•	•	•	•
FL	•	•	•	•	•	•	•	•
FLIN	•	•	•	•	•	•	•	•
FMA	•	•	•	•	•	•	•	•
FNORM	•	•	•	•	•	•	•	•
FOCOF	○	○	○	○	○	○	○	○
FOCON	○	○	○	○	○	○	○	○
FOR	•	•	•	•	•	•	•	•
FP	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
FPO	-	-	-	-	-	-	-	-
FPR	•	•	•	•	•	•	•	•
FPRAOF	•	•	•	•	•	•	•	•
FPRAON	•	•	•	•	•	•	•	•
FRAME	•	•	•	•	•	•	•	•
FRC	•	•	•	•	•	•	•	•
FRCM	•	•	•	•	•	•	•	•
FROM	•	•	•	•	•	•	•	•
FTOC	•	•	•	•	•	•	•	•
FTOCOF	•	•	•	•	•	•	•	•
FTOCON	•	•	•	•	•	•	•	•
FXS	•	•	•	•	•	•	•	•
FXST	•	•	•	•	•	•	•	•
FXSW	•	•	•	•	•	•	•	•
FZ	•	•	•	•	•	•	•	•

Příkazy G ... L

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
G0	•	•	•	•	•	•	•	•
G1	•	•	•	•	•	•	•	•
G2	•	•	•	•	•	•	•	•
G3	•	•	•	•	•	•	•	•
G4	•	•	•	•	•	•	•	•
G5	•	•	•	•	•	•	•	•
G7	•	•	•	•	•	•	•	•
G9	•	•	•	•	•	•	•	•
G17	•	•	•	•	•	•	•	•
G18	•	•	•	•	•	•	•	•
G19	•	•	•	•	•	•	•	•
G25	•	•	•	•	•	•	•	•
G26	•	•	•	•	•	•	•	•
G33	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
G34	•	•	•	•	•	•	•	•
G35	•	•	•	•	•	•	•	•
G40	•	•	•	•	•	•	•	•
G41	•	•	•	•	•	•	•	•
G42	•	•	•	•	•	•	•	•
G53	•	•	•	•	•	•	•	•
G54	•	•	•	•	•	•	•	•
G55	•	•	•	•	•	•	•	•
G56	•	•	•	•	•	•	•	•
G57	•	•	•	•	•	•	•	•
G58	→ G505							
G59	→ G506							
G60	•	•	•	•	•	•	•	•
G62	•	•	•	•	•	•	•	•
G63	•	•	•	•	•	•	•	•
G64	•	•	•	•	•	•	•	•
G70	•	•	•	•	•	•	•	•
G71	•	•	•	•	•	•	•	•
G74	•	•	•	•	•	•	•	•
G75	•	•	•	•	•	•	•	•
G90	•	•	•	•	•	•	•	•
G91	•	•	•	•	•	•	•	•
G93	•	•	•	•	•	•	•	•
G94	•	•	•	•	•	•	•	•
G95	•	•	•	•	•	•	•	•
G96	•	•	•	•	•	•	•	•
G97	•	•	•	•	•	•	•	•
G110	•	•	•	•	•	•	•	•
G111	•	•	•	•	•	•	•	•
G112	•	•	•	•	•	•	•	•
G140	•	•	•	•	•	•	•	•
G141	•	•	•	•	•	•	•	•
G142	•	•	•	•	•	•	•	•
G143	•	•	•	•	•	•	•	•
G147	•	•	•	•	•	•	•	•
G148	•	•	•	•	•	•	•	•
G153	•	•	•	•	•	•	•	•
G247	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
G248	•	•	•	•	•	•	•	•
G290	•	•	•	•	•	•	•	•
G291	•	•	•	•	•	•	•	•
G331	•	•	•	•	•	•	•	•
G332	•	•	•	•	•	•	•	•
G335	•	•	•	•	•	•	•	•
G336	•	•	•	•	•	•	•	•
G340	•	•	•	•	•	•	•	•
G341	•	•	•	•	•	•	•	•
G347	•	•	•	•	•	•	•	•
G348	•	•	•	•	•	•	•	•
G450	•	•	•	•	•	•	•	•
G451	•	•	•	•	•	•	•	•
G460	•	•	•	•	•	•	•	•
G461	•	•	•	•	•	•	•	•
G462	•	•	•	•	•	•	•	•
G500	•	•	•	•	•	•	•	•
G505 ... G599	•	•	•	•	•	•	•	•
G601	•	•	•	•	•	•	•	•
G602	•	•	•	•	•	•	•	•
G603	•	•	•	•	•	•	•	•
G621	•	•	•	•	•	•	•	•
G641	•	•	•	•	•	•	•	•
G642	•	•	•	•	•	•	•	•
G643	•	•	•	•	•	•	•	•
G644	•	•	•	•	•	•	•	•
G645	•	•	•	•	•	•	•	•
G700	•	•	•	•	•	•	•	•
G710	•	•	•	•	•	•	•	•
G810 ... G819	-	-	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-	-	-
G931	•	•	•	•	•	•	•	•
G942	•	•	•	•	•	•	•	•
G952	•	•	•	•	•	•	•	•
G961	•	•	•	•	•	•	•	•
G962	•	•	•	•	•	•	•	•
G971	•	•	•	•	•	•	•	•
G972	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
G973	•	•	•	•	•	•	•	•
GEOAX	•	•	•	•	•	•	•	•
GET	•	•	•	•	•	•	•	•
GETACTT	•	•	•	•	•	•	•	•
GETACTTD	•	•	•	•	•	•	•	•
GETD	-	-	-	-	-	○	-	•
GETDNO	•	•	•	•	•	•	•	•
GETEXET	•	•	•	•	•	•	•	•
GETFREELOC	•	•	•	•	•	•	•	•
GETSELT	•	•	•	•	•	•	•	•
GETT	•	•	•	•	•	•	•	•
GETTCOR	•	•	•	•	•	•	•	•
GETTENV	•	•	•	•	•	•	•	•
GETVARAP	•	•	•	•	•	•	•	•
GETVARDFT	•	•	•	•	•	•	•	•
GETVARLIM	•	•	•	•	•	•	•	•
GETVARPHU	•	•	•	•	•	•	•	•
GETVARTYP	•	•	•	•	•	•	•	•
GFRAME0 ... GFRAME100	-	-	-	-	-	-	-	-
GOTO	•	•	•	•	•	•	•	•
GOTOB	•	•	•	•	•	•	•	•
GOTOC	•	•	•	•	•	•	•	•
GOTOF	•	•	•	•	•	•	•	•
GOTOS	•	•	•	•	•	•	•	•
GP	•	•	•	•	•	•	•	•
GWPSOF	•	•	•	•	•	•	•	•
GROUP_ ADDEND	•	•	•	•	•	•	•	•
GROUP_BEGIN	•	•	•	•	•	•	•	•
GROUP_END	•	•	•	•	•	•	•	•
GWPSO	•	•	•	•	•	•	•	•
H...	•	•	•	•	•	•	•	•
HOLES1	•	•	•	•	•	•	•	•
HOLES2	•	•	•	•	•	•	•	•
I	•	•	•	•	•	•	•	•
I1	•	•	•	•	•	•	•	•
IC	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
ICYCOF	•	•	•	•	•	•	•	•
ICYCON	•	•	•	•	•	•	•	•
ID	•	•	•	•	•	•	•	•
IDS	•	•	•	•	•	•	•	•
IF	•	•	•	•	•	•	•	•
INDEX	•	•	•	•	•	•	•	•
INIPO	•	•	•	•	•	•	•	•
INIRE	•	•	•	•	•	•	•	•
INICF	•	•	•	•	•	•	•	•
INIT	-	-	-	-	-	○	-	•
INITIAL								
INT	•	•	•	•	•	•	•	•
INTERSEC	•	•	•	•	•	•	•	•
INVCCW	-	-	-	-	-	-	-	-
INVCW	-	-	-	-	-	-	-	-
INVFRAME	•	•	•	•	•	•	•	•
IP	•	•	•	•	•	•	•	•
IPOBRKA	•	•	•	•	•	•	•	•
IPOENDA	•	•	•	•	•	•	•	•
IPTRLOCK	•	•	•	•	•	•	•	•
IPTRUNLOCK	•	•	•	•	•	•	•	•
IR	•	•	•	•	•	•	•	•
ISAXIS	•	•	•	•	•	•	•	•
ISD	-	-	-	-	-	-	-	-
ISFILE	•	•	•	•	•	•	•	•
ISNUMBER	•	•	•	•	•	•	•	•
ISOCALL	•	•	•	•	•	•	•	•
ISVAR	•	•	•	•	•	•	•	•
J	•	•	•	•	•	•	•	•
J1	•	•	•	•	•	•	•	•
JERKA	•	•	•	•	•	•	•	•
JERKLIM	•	•	•	•	•	•	•	•
JERKLIMA	•	•	•	•	•	•	•	•
JR	•	•	•	•	•	•	•	•
K	•	•	•	•	•	•	•	•
K1	•	•	•	•	•	•	•	•
KONT	•	•	•	•	•	•	•	•
KONTC	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
KONTT	•	•	•	•	•	•	•	•
KR	•	•	•	•	•	•	•	•
L	•	•	•	•	•	•	•	•
LEAD								
Orientace nástroje	-	-	-	-	-	-	-	-
Polynom orientace	-	-	-	-	-	-	-	-
LEADOF	-	-	-	•	-	-	•	•
LEADON	-	-	-	•	-	-	•	•
LENTOAX	•	•	•	•	•	•	•	•
LFOF	•	•	•	•	•	•	•	•
LFON	•	•	•	•	•	•	•	•
LFPOS	•	•	•	•	•	•	•	•
LFTXT	•	•	•	•	•	•	•	•
LFWP	•	•	•	•	•	•	•	•
LIFTFAST	•	•	•	•	•	•	•	•
LIMS	•	•	•	•	•	•	•	•
LLI	•	•	•	•	•	•	•	•
LN	•	•	•	•	•	•	•	•
LOCK	•	•	•	•	•	•	•	•
LONGHOLE	•	•	•	•	•	•	•	•
LOOP	•	•	•	•	•	•	•	•

Příkazy M ... R

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
M0	•	•	•	•	•	•	•	•
M1	•	•	•	•	•	•	•	•
M2	•	•	•	•	•	•	•	•
M3	•	•	•	•	•	•	•	•
M4	•	•	•	•	•	•	•	•
M5	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
M6	•	•	•	•	•	•	•	•
M17	•	•	•	•	•	•	•	•
M19	•	•	•	•	•	•	•	•
M30	•	•	•	•	•	•	•	•
M40	•	•	•	•	•	•	•	•
M41 ... M45	•	•	•	•	•	•	•	•
M70	•	•	•	•	•	•	•	•
MASLDEF	-	-	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-	-	-
MASLON	-	-	-	-	-	-	-	-
MATCH	•	•	•	•	•	•	•	•
MAXVAL	•	•	•	•	•	•	•	•
MCALL	•	•	•	•	•	•	•	•
MEAC	-	-	-	-	-	-	-	-
MEAFRAME	•	•	•	•	•	•	•	•
MEAS	•	•	•	•	•	•	•	•
MEASA	-	-	-	-	-	-	-	-
MEASURE	•	•	•	•	•	•	•	•
MEAW	•	•	•	•	•	•	•	•
MEAWA	-	-	-	-	-	-	-	-
MI	•	•	•	•	•	•	•	•
MINDEX	•	•	•	•	•	•	•	•
MINVAL	•	•	•	•	•	•	•	•
MIRROR	•	•	•	•	•	•	•	•
MMC	•	•	•	•	•	•	•	•
MOD	•	•	•	•	•	•	•	•
MODAXVAL	•	•	•	•	•	•	•	•
MOV	•	•	•	•	•	•	•	•
MOVTT	•	•	•	•	•	•	•	•
MSG	•	•	•	•	•	•	•	•
MVTOOL	•	•	•	•	•	•	•	•
N	•	•	•	•	•	•	•	•
NAMETOINT	•	•	•	•	•	•	•	•
NC	•	•	•	•	•	•	•	•
NEWCONF	•	•	•	•	•	•	•	•
NEWMT	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
• Standardní								
○ Volitelný do- plněk								
- není k dispozici								
NEWT	•	•	•	•	•	•	•	•
NORM	•	•	•	•	•	•	•	•
NOT	•	•	•	•	•	•	•	•
NPROT	•	•	•	•	•	•	•	•
NPROTDEF	•	•	•	•	•	•	•	•
NUMBER	•	•	•	•	•	•	•	•
OEMIPO1	-	-	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-	-	-
OF	•	•	•	•	•	•	•	•
OFFN	•	•	•	•	•	•	•	•
OMA1	-	-	-	-	-	-	-	-
OMA2	-	-	-	-	-	-	-	-
OMA3	-	-	-	-	-	-	-	-
OMA4	-	-	-	-	-	-	-	-
OMA5	-	-	-	-	-	-	-	-
OR	•	•	•	•	•	•	•	•
ORIXES	-	-	-	-	-	-	-	-
ORIXPOS	-	-	-	-	-	-	-	-
ORIC	-	-	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-	-	-
ORID	-	-	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-	-	-
ORIPANE	-	-	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-	-	-
ORIS	-	-	-	-	-	-	-	-

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
ORISOF	-	-	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-	-	-
ORISON	-	-	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-	-	-
OS	-	-	-	-	-	-	-	-
OSB	-	-	-	-	-	-	-	-
OSC	-	-	-	-	-	-	-	-
OSCILL	-	-	-	-	-	-	-	-
OSCTRL	-	-	-	-	-	-	-	-
OSD	-	-	-	-	-	-	-	-
OSE	-	-	-	-	-	-	-	-
OSNSC	•	•	•	•	•	•	•	•
OSOF	-	-	-	-	-	-	-	-
OSP1	•	•	•	•	•	•	•	•
OSP2	•	•	•	•	•	•	•	•
OSS	-	-	-	-	-	-	-	-
OSSE	-	-	-	-	-	-	-	-
OST	-	-	-	-	-	-	-	-
OST1	•	•	•	•	•	•	•	•
OST2	•	•	•	•	•	•	•	•
OTOL	•	•	•	•	•	•	•	•
OVR	•	•	•	•	•	•	•	•
OVRA	•	•	•	•	•	•	•	•
OVRRAP	•	•	•	•	•	•	•	•
P	•	•	•	•	•	•	•	•
PAROT	•	•	•	•	•	•	•	•
PAROTOF	•	•	•	•	•	•	•	•
PCALL	•	•	•	•	•	•	•	•
PDELAYOF	-	-	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-	-	-
PHI	-	-	-	-	-	-	-	-
PHU	•	•	•	•	•	•	•	•
PL	-	-	-	-	-	-	-	-
PM	•	•	•	•	•	•	•	•
PO	-	-	-	-	-	-	-	-

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
POCKET3	•	•	•	•	•	•	•	•
POCKET4	•	•	•	•	•	•	•	•
POLF	•	•	•	•	•	•	•	•
POLFA	•	•	•	•	•	•	•	•
POLFMASK	•	•	•	•	•	•	•	•
POLFMLIN	•	•	•	•	•	•	•	•
POLY	-	-	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-	-	-
PON	-	-	-	-	-	-	-	-
PONS	-	-	-	-	-	-	-	-
POS	•	•	•	•	•	•	•	•
POSA	•	•	•	•	•	•	•	•
POSM	•	•	•	•	•	•	•	•
POSMT	•	•	•	•	•	•	•	•
POSP	•	•	•	•	•	•	•	•
POSRANGE	•	•	•	•	•	•	•	•
POT	•	•	•	•	•	•	•	•
PR	•	•	•	•	•	•	•	•
PREPRO	•	•	•	•	•	•	•	•
PRESETON	•	•	•	•	•	•	•	•
PRESETONS	•	•	•	•	•	•	•	•
PRIO	•	•	•	•	•	•	•	•
PRLOC	•	•	•	•	•	•	•	•
PROC	•	•	•	•	•	•	•	•
PROTA	•	•	•	•	•	•	•	•
PROTD	•	•	•	•	•	•	•	•
PROTS	•	•	•	•	•	•	•	•
PSI	-	-	-	-	-	-	-	-
PTP	•	•	•	•	•	•	•	•
PTPG0	•	•	•	•	•	•	•	•
PTPWOC	•	•	•	•	•	•	•	•
PUNCHACC	-	-	-	-	-	-	-	-
PUTFTOC	•	•	•	•	•	•	•	•
PUTFTOCF	•	•	•	•	•	•	•	•
PW	○	○	○	○	○	○	○	○
QU	•	•	•	•	•	•	•	•
R...	•	•	•	•	•	•	•	•
RAC	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
• Standardní								
◦ Volitelný do- plněk								
- není k dispozici								
RDISABLE	•	•	•	•	•	•	•	•
READ	•	•	•	•	•	•	•	•
REAL	•	•	•	•	•	•	•	•
RELEASE	•	•	•	•	•	•	•	•
REP	•	•	•	•	•	•	•	•
REPEAT	•	•	•	•	•	•	•	•
REPEATB	•	•	•	•	•	•	•	•
REPOSA	•	•	•	•	•	•	•	•
REPOSH	•	•	•	•	•	•	•	•
REPOSHA	•	•	•	•	•	•	•	•
REPOSL	•	•	•	•	•	•	•	•
REPOSQ	•	•	•	•	•	•	•	•
REPOSQA	•	•	•	•	•	•	•	•
RESETMON	•	•	•	•	•	•	•	•
RET	•	•	•	•	•	•	•	•
RETB	•	•	•	•	•	•	•	•
RIC	•	•	•	•	•	•	•	•
RINDEX	•	•	•	•	•	•	•	•
RMB	•	•	•	•	•	•	•	•
RME	•	•	•	•	•	•	•	•
RMI	•	•	•	•	•	•	•	•
RMN	•	•	•	•	•	•	•	•
RND	•	•	•	•	•	•	•	•
RNDM	•	•	•	•	•	•	•	•
ROT	•	•	•	•	•	•	•	•
ROTS	•	•	•	•	•	•	•	•
ROUND	•	•	•	•	•	•	•	•
ROUNDUP	•	•	•	•	•	•	•	•
RP	•	•	•	•	•	•	•	•
RPL	•	•	•	•	•	•	•	•
RT	•	•	•	•	•	•	•	•
RTLIOF	•	•	•	•	•	•	•	•
RTLION	•	•	•	•	•	•	•	•

Příkazy S ... Z

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
S	•	•	•	•	•	•	•	•
SAVE	•	•	•	•	•	•	•	•
SBLOF	•	•	•	•	•	•	•	•
SBLON	•	•	•	•	•	•	•	•
SC	•	•	•	•	•	•	•	•
SCALE	•	•	•	•	•	•	•	•
SCC	•	•	•	•	•	•	•	•
SCPARA	•	•	•	•	•	•	•	•
SD	○	○	○	○	○	○	○	○
SET	•	•	•	•	•	•	•	•
SETAL	•	•	•	•	•	•	•	•
SETDNO	•	•	•	•	•	•	•	•
SETINT	•	•	•	•	•	•	•	•
SETM	-	-	-	-	-	○	-	•
SETMS	•	•	•	•	•	•	•	•
SETMS (n)	•	•	•	•	•	•	•	•
SETMTH	•	•	•	•	•	•	•	•
SETPIECE	•	•	•	•	•	•	•	•
SETTA	•	•	•	•	•	•	•	•
SETTCOR	•	•	•	•	•	•	•	•
SETTIA	•	•	•	•	•	•	•	•
SF	•	•	•	•	•	•	•	•
SIN	•	•	•	•	•	•	•	•
SIRELAY	-	-	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-	-	-
SLOT1	•	•	•	•	•	•	•	•
SLOT2	•	•	•	•	•	•	•	•
SOFT	•	•	•	•	•	•	•	•
SOFTA	•	•	•	•	•	•	•	•
SON	-	-	-	-	-	-	-	-
SONS	-	-	-	-	-	-	-	-
SPATH	•	•	•	•	•	•	•	•
SPCOF	•	•	•	•	•	•	•	•
SPCON	•	•	•	•	•	•	•	•

Příkaz	SINUMERIK 828D							
● Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
SPI	●	●	●	●	●	●	●	●
SPIF1	-	-	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○	○	○
SPN	-	-	-	-	-	-	-	-
SPOF	-	-	-	-	-	-	-	-
SPOS	●	●	●	●	●	●	●	●
SPOSA	●	●	●	●	●	●	●	●
SPP	-	-	-	-	-	-	-	-
SPRINT	●	●	●	●	●	●	●	●
SQRT	●	●	●	●	●	●	●	●
SR	●	●	●	●	●	●	●	●
SRA	●	●	●	●	●	●	●	●
ST	●	●	●	●	●	●	●	●
STA	●	●	●	●	●	●	●	●
START	-	-	-	-	-	○	-	●
STARTFIFO	●	●	●	●	●	●	●	●
STAT	●	●	●	●	●	●	●	●
STOLF	●	●	●	●	●	●	●	●
STOPFIFO	●	●	●	●	●	●	●	●
STOPRE	●	●	●	●	●	●	●	●
STOPREOF	●	●	●	●	●	●	●	●
STRING	●	●	●	●	●	●	●	●
STRINGFELD	●	●	●	●	●	●	●	●
STRINGIS	●	●	●	●	●	●	●	●
STRLEN	●	●	●	●	●	●	●	●
SUBSTR	●	●	●	●	●	●	●	●
SUPA	●	●	●	●	●	●	●	●
SVC	●	●	●	●	●	●	●	●
SYNFCT	●	●	●	●	●	●	●	●
SYNR	●	●	●	●	●	●	●	●
SYNRW	●	●	●	●	●	●	●	●
SYNW	●	●	●	●	●	●	●	●
T	●	●	●	●	●	●	●	●
TAN	●	●	●	●	●	●	●	●
TANG	-	-	-	-	-	-	-	-
TANGDEL	-	-	-	-	-	-	-	-
TANGOF	-	-	-	-	-	-	-	-

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
TANGON	-	-	-	-	-	-	-	-
TCA (828D: _TCA)	•	•	•	•	•	•	•	•
TCARR	•	-	•	-	•	•	-	-
TCI	•	•	•	•	•	•	•	•
TCOABS	•	-	•	-	•	•	-	-
TCOFR	•	-	•	-	•	•	-	-
TCOFRX	•	-	•	-	•	•	-	-
TCOFRY	•	-	•	-	•	•	-	-
TCOFRZ	•	-	•	-	•	•	-	-
THETA	-	-	-	-	-	-	-	-
TILT	-	-	-	-	-	-	-	-
TLIFT	-	-	-	-	-	-	-	-
TML	•	•	•	•	•	•	•	•
TMOF	•	•	•	•	•	•	•	•
TMON	•	•	•	•	•	•	•	•
TO	•	•	•	•	•	•	•	•
TOFF	•	•	•	•	•	•	•	•
TOFFL	•	•	•	•	•	•	•	•
TOFFOF	•	•	•	•	•	•	•	•
TOFFON	•	•	•	•	•	•	•	•
TOFFR	•	•	•	•	•	•	•	•
TOFRAME	•	•	•	•	•	•	•	•
TOFRAMEX	•	•	•	•	•	•	•	•
TOFRAMEY	•	•	•	•	•	•	•	•
TOFRAMEZ	•	•	•	•	•	•	•	•
TOLOWER	•	•	•	•	•	•	•	•
TOOLENV	•	•	•	•	•	•	•	•
TOOLGNT	•	•	•	•	•	•	•	•
TOOLGT	•	•	•	•	•	•	•	•
TOROT	•	•	•	•	•	•	•	•
TOROTOF	•	•	•	•	•	•	•	•
TOROTX	•	•	•	•	•	•	•	•
TOROTY	•	•	•	•	•	•	•	•
TOROTZ	•	•	•	•	•	•	•	•
TOUPPER	•	•	•	•	•	•	•	•
TOWBCS	•	-	•	-	•	•	-	-
TOWKCS	•	-	•	-	•	•	-	-

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustru- žení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustru- žení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustru- žení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
TOWMCS	•	-	•	-	•	•	-	-
TOWSTD	•	-	•	-	•	•	-	-
TOWTCS	•	-	•	-	•	•	-	-
TOWWCS	•	-	•	-	•	•	-	-
TR	•	•	•	•	•	•	•	•
TRAANG	-	-	-	-	-	-	○	○
TRACON	-	-	-	-	-	-	○	○
TRACYL	○	○	○	○	○	○	○	○
TRAFOOF	•	•	•	•	•	•	•	•
TRAFOON	-	-	-	-	-	-	-	-
TRAILOF	•	•	•	•	•	•	•	•
TRAILON	•	•	•	•	•	•	•	•
TRANS	•	•	•	•	•	•	•	•
TRANSMIT	○	○	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-	-	-
TRUE	•	•	•	•	•	•	•	•
TRUNC	•	•	•	•	•	•	•	•
TU	•	•	•	•	•	•	•	•
TURN	•	•	•	•	•	•	•	•
ULI	•	•	•	•	•	•	•	•
UNLOCK	•	•	•	•	•	•	•	•
UNTIL	•	•	•	•	•	•	•	•
UPATH	•	•	•	•	•	•	•	•
VAR	•	•	•	•	•	•	•	•
VELOLIM	•	•	•	•	•	•	•	•
VELOLIMA	•	•	•	•	•	•	•	•
WAITC	•	•	•	•	•	•	•	•
WAITE	-	-	-	-	-	○	-	•
WAITENC	•	•	•	•	•	•	•	•
WAITM	-	-	-	-	-	○	-	•
WAITMC	-	-	-	-	-	○	-	•
WAITP	•	•	•	•	•	•	•	•
WAITS	•	•	•	•	•	•	•	•
WALCS0	•	•	•	•	•	•	•	•
WALCS1	•	•	•	•	•	•	•	•
WALCS2	•	•	•	•	•	•	•	•
WALCS3	•	•	•	•	•	•	•	•
WALCS4	•	•	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D							
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW Frézování Export (me42)	SW24x(5) CNC-SW Soustružení Export (te42)	SW26x(3) CNC-SW Frézování Export (me62)	SW26x(3) CNC-SW Soustružení Export (te62)	SW28x(2) CNC-SW Frézování Export (me821)	SW28x(1) CNC-SW Frézování Adv. Export (me822)	SW28x(2) CNC-SW Soustružení Export (te821)	SW28x(1) CNC-SW Soustružení Adv. Export (te822)
WALCS5	•	•	•	•	•	•	•	•
WALCS6	•	•	•	•	•	•	•	•
WALCS7	•	•	•	•	•	•	•	•
WALCS8	•	•	•	•	•	•	•	•
WALCS9	•	•	•	•	•	•	•	•
WALCS10	•	•	•	•	•	•	•	•
WALIMOF	•	•	•	•	•	•	•	•
WALIMON	•	•	•	•	•	•	•	•
WHEN	•	•	•	•	•	•	•	•
WHENEVER	•	•	•	•	•	•	•	•
WHILE	•	•	•	•	•	•	•	•
WORKPIECE	•	•	•	•	•	•	•	•
WRITE	•	•	•	•	•	•	•	•
WRTPR	•	•	•	•	•	•	•	•
X	•	•	•	•	•	•	•	•
XOR	•	•	•	•	•	•	•	•
Y	•	•	•	•	•	•	•	•
Z	•	•	•	•	•	•	•	•

21.2.2 Varianty řídicího systému pro broušení

Příkazy A ... C

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
:	•	•	•	•	•	•
*	•	•	•	•	•	•
+	•	•	•	•	•	•
-	•	•	•	•	•	•
<	•	•	•	•	•	•
<<	•	•	•	•	•	•
<=	•	•	•	•	•	•

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
=	•	•	•	•	•	•
>=	•	•	•	•	•	•
/	•	•	•	•	•	•
/0 ... /7	•	•	•	•	•	•
A	•	•	•	•	•	•
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-
A5	-	-	-	-	-	-
A6	-	-	-	-	-	-
A7	-	-	-	-	-	-
ABS	•	•	•	•	•	•
AC	•	•	•	•	•	•
ACC	•	•	•	•	•	•
ACCLIMA	•	•	•	•	•	•
ACN	•	•	•	•	•	•
ACOS	•	•	•	•	•	•
ACP	•	•	•	•	•	•
ACTBLOCNO	•	•	•	•	•	•
ADDFRAME	•	•	•	•	•	•
ADIS	•	•	•	•	•	•
ADISPOS	•	•	•	•	•	•
ADISPOSA	•	•	•	•	•	•
ALF	•	•	•	•	•	•
AMIRROR	•	•	•	•	•	•
AND	•	•	•	•	•	•
ANG	•	•	•	•	•	•
AP	•	•	•	•	•	•
APR	•	•	•	•	•	•
APRB	•	•	•	•	•	•
APRP	•	•	•	•	•	•
APW	•	•	•	•	•	•
APWB	•	•	•	•	•	•
APWP	•	•	•	•	•	•
APX	•	•	•	•	•	•
AR	•	•	•	•	•	•
AROT	•	•	•	•	•	•
AROTS	•	•	•	•	•	•
AS	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
ASCALE	•	•	•	•	•	•
ASIN	•	•	•	•	•	•
ASPLINE	○	○	○	○	○	○
ATAN2	•	•	•	•	•	•
ATOL	•	•	•	•	•	•
ATRANS	•	•	•	•	•	•
AUXFUDEL	•	•	•	•	•	•
AUXFUDELG	•	•	•	•	•	•
AUXFUMSEQ	•	•	•	•	•	•
AUXFUSYNC	•	•	•	•	•	•
AX	•	•	•	•	•	•
AXCTSWE	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-
AXIS	•	•	•	•	•	•
AXNAME	•	•	•	•	•	•
AXSTRING	•	•	•	•	•	•
AXTOCHAN	•	•	•	•	•	•
AXTOSPI	•	•	•	•	•	•
B	•	•	•	•	•	•
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B6	-	-	-	-	-	-
B7	-	-	-	-	-	-
B_AND	•	•	•	•	•	•
B_OR	•	•	•	•	•	•
B_NOT	•	•	•	•	•	•
B_XOR	•	•	•	•	•	•
BAUTO	○	○	○	○	○	○
BLOCK	•	•	•	•	•	•
BLSYNC	•	•	•	•	•	•
BNAT	○	○	○	○	○	○
BOOL	•	•	•	•	•	•
BOUND	•	•	•	•	•	•
BRISK	•	•	•	•	•	•
BRISKA	•	•	•	•	•	•
BSPLINE	○	○	○	○	○	○

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	Název kanálo- vé osy	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
CAC	●	●	●	●	●	●
CACN	●	●	●	●	●	●
CACP	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●
CALCPOSI	●	●	●	●	●	●
CALL	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●
CASE	●	●	●	●	●	●
CDC	●	●	●	●	●	●
CDOF	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-
CDON	-	-	-	-	-	-
CFC	●	●	●	●	●	●
CFIN	●	●	●	●	●	●
CFINE	●	●	●	●	●	●
CFTCP	●	●	●	●	●	●
CHAN	●	●	●	●	●	●
CHANDATA	●	●	●	●	●	●
CHAR	●	●	●	●	●	●
CHF	●	●	●	●	●	●
CHKDM	●	●	●	●	●	●
CHKDNO	●	●	●	●	●	●
CHR	●	●	●	●	●	●
CIC	●	●	●	●	●	●
CIP	●	●	●	●	●	●
CLEARM	-	-	●	-	-	●
CLRINT	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
COLLPAIR	-	-	-	-	-	-
COMPCAD	•	•	•	•	•	•
COMPCURV	•	•	•	•	•	•
COMPLETE	•	•	•	•	•	•
COMPOF	•	•	•	•	•	•
COMPON	•	•	•	•	•	•
COMPSURF	-	-	-	-	-	-
CONTDCON	•	•	•	•	•	•
CONTPRON	•	•	•	•	•	•
CORROF	•	•	•	•	•	•
CORRTRAFO	-	-	-	-	-	-
COS	•	•	•	•	•	•
COUPDEF	○	○	○	○	○	○
COUPDEL	○	○	○	○	○	○
COUPOF	○	○	○	○	○	○
COUPOFS	○	○	○	○	○	○
COUPON	○	○	○	○	○	○
COUPONC	○	○	○	○	○	○
COUPRES	○	○	○	○	○	○
CP	•	•	•	•	•	•
CPBC	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○
CPLA	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○
CPLON	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
CPLOUTTR	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○
CPOF	○	○	○	○	○	○
CPON	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●
CPRES	○	○	○	○	○	○
CPROT	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○
CPSYNCOF	○	○	○	○	○	○
CPSYNCOF2	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○
CR	●	●	●	●	●	●
CROT	●	●	●	●	●	●
CROTS	●	●	●	●	●	●
CRPL	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○
CT	●	●	●	●	●	●
CTAB	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-
CTABID	-	-	-	-	-	-

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
CTABINV	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-
CTOL	•	•	•	•	•	•
CTRANS	•	•	•	•	•	•
CUT2D	•	•	•	•	•	•
CUT2DD	•	•	•	•	•	•
CUT2DF	•	•	•	•	•	•
CUT2DFD	•	•	•	•	•	•
CUT3DC	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-
CUTCONOF	•	•	•	•	•	•
CUTCONON	•	•	•	•	•	•
CUTMOD	•	•	•	•	•	•

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
CUTMODK	-	-	-	-	-	-
CYCLE60	-	-	-	-	-	-
CYCLE61	-	-	-	-	-	-
CYCLE62	•	•	•	•	•	•
CYCLE63	-	-	-	-	-	-
CYCLE64	-	-	-	-	-	-
CYCLE70	-	-	-	-	-	-
CYCLE72	-	-	-	-	-	-
CYCLE76	-	-	-	-	-	-
CYCLE77	-	-	-	-	-	-
CYCLE78	-	-	-	-	-	-
CYCLE79	-	-	-	-	-	-
CYCLE81	-	-	-	-	-	-
CYCLE82	-	-	-	-	-	-
CYCLE83	-	-	-	-	-	-
CYCLE84	-	-	-	-	-	-
CYCLE85	-	-	-	-	-	-
CYCLE86	-	-	-	-	-	-
CYCLE92	-	-	-	-	-	-
CYCLE95	-	-	-	-	-	-
CYCLE98	-	-	-	-	-	-
CYCLE99	-	-	-	-	-	-
CYCLE150	-	-	-	-	-	-
CYCLE435	○	○	○	○	○	○
CYCLE495	○	○	○	○	○	○
CYCLE750	•	•	•	•	•	•
CYCLE751	•	•	•	•	•	•
CYCLE752	•	•	•	•	•	•
CYCLE753	•	•	•	•	•	•
CYCLE754	•	•	•	•	•	•
CYCLE755	•	•	•	•	•	•
CYCLE756	•	•	•	•	•	•
CYCLE757	•	•	•	•	•	•
CYCLE758	•	•	•	•	•	•
CYCLE759	•	•	•	•	•	•
CYCLE800	○	○	○	○	○	○
CYCLE801	-	-	-	-	-	-
CYCLE802	-	-	-	-	-	-
CYCLE830	-	-	-	-	-	-

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
CYCLE832	•	•	•	•	•	•
CYCLE840	-	-	-	-	-	-
CYCLE899	-	-	-	-	-	-
CYCLE930	-	-	-	-	-	-
CYCLE940	-	-	-	-	-	-
CYCLE951	-	-	-	-	-	-
CYCLE952	-	-	-	-	-	-
CYCLE961	-	-	-	-	-	-
CYCLE971	-	-	-	-	-	-
CYCLE973	-	-	-	-	-	-
CYCLE974	-	-	-	-	-	-
CYCLE976	-	-	-	-	-	-
CYCLE977	-	-	-	-	-	-
CYCLE978	-	-	-	-	-	-
CYCLE979	-	-	-	-	-	-
CYCLE982	-	-	-	-	-	-
CYCLE994	-	-	-	-	-	-
CYCLE995	-	-	-	-	-	-
CYCLE996	-	-	-	-	-	-
CYCLE997	-	-	-	-	-	-
CYCLE998	-	-	-	-	-	-
CYCLE4071	•	•	•	•	•	•
CYCLE4072	•	•	•	•	•	•
CYCLE4073	•	•	•	•	•	•
CYCLE4074	•	•	•	•	•	•
CYCLE4075	•	•	•	•	•	•
CYCLE4077	•	•	•	•	•	•
CYCLE4078	•	•	•	•	•	•
CYCLE4079	•	•	•	•	•	•

Příkazy D ... F

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
D	•	•	•	•	•	•
D0	•	•	•	•	•	•
DAC	•	•	•	•	•	•
DC	•	•	•	•	•	•
DCI	•	•	•	•	•	•
DCM	•	•	•	•	•	•
DCU	•	•	•	•	•	•
DEF	•	•	•	•	•	•
DEFINE	•	•	•	•	•	•
DEFAULT	•	•	•	•	•	•
DELAYFSTON	•	•	•	•	•	•
DELAYFSTOF	•	•	•	•	•	•
DELDL	•	•	•	•	•	•
DELDTG	•	•	•	•	•	•
DELETE	•	•	•	•	•	•
DELMLOWNER	•	•	•	•	•	•
DEMLRES	•	•	•	•	•	•
DELMT	-	-	-	-	-	-
DELOBJ	-	-	-	-	-	-
DELT	•	•	•	•	•	•
DELTC	•	•	•	•	•	•
DELTOOLNV	•	•	•	•	•	•
DIACYCOFA	•	•	•	•	•	•
DIAM90	•	•	•	•	•	•
DIAM90A	•	•	•	•	•	•
DIAMCHAN	•	•	•	•	•	•
DIAMCHANA	•	•	•	•	•	•
DIAMCYCOF	•	•	•	•	•	•
DIAMOF	•	•	•	•	•	•
DIAMOFA	•	•	•	•	•	•
DIAMON	•	•	•	•	•	•
DIAMONA	•	•	•	•	•	•
DIC	•	•	•	•	•	•
DILF	•	•	•	•	•	•
DISABLE	•	•	•	•	•	•
DISC	•	•	•	•	•	•
DISCL	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
DISPLOF	•	•	•	•	•	•
DISPLON	•	•	•	•	•	•
DISPR	•	•	•	•	•	•
DISR	•	•	•	•	•	•
DISRP	•	•	•	•	•	•
DITE	•	•	•	•	•	•
DITS	•	•	•	•	•	•
DIV	•	•	•	•	•	•
DL	-	-	-	-	-	-
DO	•	•	•	•	•	•
DRFOF	•	•	•	•	•	•
DRIVE	•	•	•	•	•	•
DRIVEA	•	•	•	•	•	•
DYNFINISH	•	•	•	•	•	•
DYNNORM	•	•	•	•	•	•
DYNPOS	•	•	•	•	•	•
DYNROUGH	•	•	•	•	•	•
DYNSEMIFIN	•	•	•	•	•	•
DZERO	•	•	•	•	•	•
EAUTO	○	○	○	○	○	○
EGDEF	○	○	○	○	○	○
EGDEL	○	○	○	○	○	○
EGOFC	○	○	○	○	○	○
EGOFS	○	○	○	○	○	○
EGON	○	○	○	○	○	○
EGONSYN	○	○	○	○	○	○
EGONSYNE	○	○	○	○	○	○
ELSE	•	•	•	•	•	•
ENABLE	•	•	•	•	•	•
ENAT	○	○	○	○	○	○
ENDFOR	•	•	•	•	•	•
ENDIF	•	•	•	•	•	•
ENDLABEL	•	•	•	•	•	•
ENDLOOP	•	•	•	•	•	•
ENDPROC	•	•	•	•	•	•
ENDWHILE	•	•	•	•	•	•
ESRR	○	○	○	○	○	○
ESRS	○	○	○	○	○	○
ETAN	○	○	○	○	○	○

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní						
○ Volitelný doplněk						
- není k dispozici						
EVERY	•	•	•	•	•	•
EX	•	•	•	•	•	•
EXECSTRING	•	•	•	•	•	•
EXECTAB	•	•	•	•	•	•
EXECUTE	•	•	•	•	•	•
EXP	•	•	•	•	•	•
EXTCALL	•	•	•	•	•	•
EXTCLOSE	•	•	•	•	•	•
EXTERN	•	•	•	•	•	•
EXTOPEN	•	•	•	•	•	•
F	•	•	•	•	•	•
FA	•	•	•	•	•	•
FAD	•	•	•	•	•	•
FALSE	•	•	•	•	•	•
FB	•	•	•	•	•	•
FCTDEF	•	•	•	•	•	•
FCUB	•	•	•	•	•	•
FD	•	•	•	•	•	•
FDA	•	•	•	•	•	•
FENDNORM	•	•	•	•	•	•
FFWOF	•	•	•	•	•	•
FFWON	•	•	•	•	•	•
FGREF	•	•	•	•	•	•
FGROUP	•	•	•	•	•	•
FI	•	•	•	•	•	•
FIFOCTRL	•	•	•	•	•	•
FILEDATE	•	•	•	•	•	•
FILEINFO	•	•	•	•	•	•
FILESIZE	•	•	•	•	•	•
FILESTAT	•	•	•	•	•	•
FILETIME	•	•	•	•	•	•
FINEA	•	•	•	•	•	•
FL	•	•	•	•	•	•
FLIN	•	•	•	•	•	•
FMA	•	•	•	•	•	•
FNORM	•	•	•	•	•	•
FOCOF	○	○	○	○	○	○
FOCON	○	○	○	○	○	○
FOR	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
FP	•	•	•	•	•	•
FPO	-	-	-	-	-	-
FPR	•	•	•	•	•	•
FPRAOF	•	•	•	•	•	•
FPRAON	•	•	•	•	•	•
FRAME	•	•	•	•	•	•
FRC	•	•	•	•	•	•
FRCM	•	•	•	•	•	•
FROM	•	•	•	•	•	•
FTOC	•	•	•	•	•	•
FTOCOF	•	•	•	•	•	•
FTOCON	•	•	•	•	•	•
FXS	•	•	•	•	•	•
FXST	•	•	•	•	•	•
FXSW	•	•	•	•	•	•
FZ	•	•	•	•	•	•

Příkazy G ... L

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
G0	•	•	•	•	•	•
G1	•	•	•	•	•	•
G2	•	•	•	•	•	•
G3	•	•	•	•	•	•
G4	•	•	•	•	•	•
G5	•	•	•	•	•	•
G7	•	•	•	•	•	•
G9	•	•	•	•	•	•
G17	•	•	•	•	•	•
G18	•	•	•	•	•	•
G19	•	•	•	•	•	•
G25	•	•	•	•	•	•
G26	•	•	•	•	•	•
G33	•	•	•	•	•	•

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
G34	•	•	•	•	•	•
G35	•	•	•	•	•	•
G40	•	•	•	•	•	•
G41	•	•	•	•	•	•
G42	•	•	•	•	•	•
G53	•	•	•	•	•	•
G54	•	•	•	•	•	•
G55	•	•	•	•	•	•
G56	•	•	•	•	•	•
G57	•	•	•	•	•	•
G58	→ G505					
G59	→ G506					
G60	•	•	•	•	•	•
G62	•	•	•	•	•	•
G63	•	•	•	•	•	•
G64	•	•	•	•	•	•
G70	•	•	•	•	•	•
G71	•	•	•	•	•	•
G74	•	•	•	•	•	•
G75	•	•	•	•	•	•
G90	•	•	•	•	•	•
G91	•	•	•	•	•	•
G93	•	•	•	•	•	•
G94	•	•	•	•	•	•
G95	•	•	•	•	•	•
G96	•	•	•	•	•	•
G97	•	•	•	•	•	•
G110	•	•	•	•	•	•
G111	•	•	•	•	•	•
G112	•	•	•	•	•	•
G140	•	•	•	•	•	•
G141	•	•	•	•	•	•
G142	•	•	•	•	•	•
G143	•	•	•	•	•	•
G147	•	•	•	•	•	•
G148	•	•	•	•	•	•
G153	•	•	•	•	•	•
G247	•	•	•	•	•	•
G248	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
G290	•	•	•	•	•	•
G291	-	-	-	-	-	-
G331	•	•	•	•	•	•
G332	•	•	•	•	•	•
G335	•	•	•	•	•	•
G336	•	•	•	•	•	•
G340	•	•	•	•	•	•
G341	•	•	•	•	•	•
G347	•	•	•	•	•	•
G348	•	•	•	•	•	•
G450	•	•	•	•	•	•
G451	•	•	•	•	•	•
G460	•	•	•	•	•	•
G461	•	•	•	•	•	•
G462	•	•	•	•	•	•
G500	•	•	•	•	•	•
G505 ... G599	•	•	•	•	•	•
G601	•	•	•	•	•	•
G602	•	•	•	•	•	•
G603	•	•	•	•	•	•
G621	•	•	•	•	•	•
G641	•	•	•	•	•	•
G642	•	•	•	•	•	•
G643	•	•	•	•	•	•
G644	•	•	•	•	•	•
G645	•	•	•	•	•	•
G700	•	•	•	•	•	•
G710	•	•	•	•	•	•
G810 ... G819	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-
G931	•	•	•	•	•	•
G942	•	•	•	•	•	•
G952	•	•	•	•	•	•
G961	•	•	•	•	•	•
G962	•	•	•	•	•	•
G971	•	•	•	•	•	•
G972	•	•	•	•	•	•
G973	•	•	•	•	•	•
GEOAX	•	•	•	•	•	•

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
GET	•	•	•	•	•	•
GETACTT	•	•	•	•	•	•
GETACTTD	•	•	•	•	•	•
GETD	-	-	•	-	-	•
GETDNO	•	•	•	•	•	•
GETEXET	•	•	•	•	•	•
GETFREELOC	•	•	•	•	•	•
GETSELT	•	•	•	•	•	•
GETT	•	•	•	•	•	•
GETTCOR	•	•	•	•	•	•
GETTENV	•	•	•	•	•	•
GETVARAP	•	•	•	•	•	•
GETVARDFT	•	•	•	•	•	•
GETVARLIM	•	•	•	•	•	•
GETVARPHU	•	•	•	•	•	•
GETVARTYP	•	•	•	•	•	•
GFRAME0 ... GFRA- ME100	< 50	< 100	< 100	< 50	< 100	< 100
GOTO	•	•	•	•	•	•
GOTOB	•	•	•	•	•	•
GOTOC	•	•	•	•	•	•
GOTOF	•	•	•	•	•	•
GOTOS	•	•	•	•	•	•
GP	•	•	•	•	•	•
GWPSOF	•	•	•	•	•	•
GROUP_ADDEND	•	•	•	•	•	•
GROUP_BEGIN	•	•	•	•	•	•
GROUP_END	•	•	•	•	•	•
GWPSON	•	•	•	•	•	•
H...	•	•	•	•	•	•
HOLES1	-	-	-	-	-	-
HOLES2	-	-	-	-	-	-
I	•	•	•	•	•	•
I1	•	•	•	•	•	•
IC	•	•	•	•	•	•
ICYCOF	•	•	•	•	•	•
ICYCON	•	•	•	•	•	•
ID	•	•	•	•	•	•
IDS	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
IF	•	•	•	•	•	•
INDEX	•	•	•	•	•	•
INIPO	•	•	•	•	•	•
INIRE	•	•	•	•	•	•
INICF	•	•	•	•	•	•
INIT	-	-	•	-	-	•
INITIAL						
INT	•	•	•	•	•	•
INTERSEC	•	•	•	•	•	•
INVCCW	-	-	-	-	-	-
INVCW	-	-	-	-	-	-
INVFRAME	•	•	•	•	•	•
IP	•	•	•	•	•	•
IPOBRKA	•	•	•	•	•	•
IPOENDA	•	•	•	•	•	•
IPTRLOCK	•	•	•	•	•	•
IPTRUNLOCK	•	•	•	•	•	•
IR	•	•	•	•	•	•
ISAXIS	•	•	•	•	•	•
ISD	-	-	-	-	-	-
ISFILE	•	•	•	•	•	•
ISNUMBER	•	•	•	•	•	•
ISOCALL	-	-	-	-	-	-
ISVAR	•	•	•	•	•	•
J	•	•	•	•	•	•
J1	•	•	•	•	•	•
JERKA	•	•	•	•	•	•
JERKLIM	•	•	•	•	•	•
JERKLIMA	•	•	•	•	•	•
JR	•	•	•	•	•	•
K	•	•	•	•	•	•
K1	•	•	•	•	•	•
KONT	•	•	•	•	•	•
KONTC	•	•	•	•	•	•
KONTT	•	•	•	•	•	•
KR	•	•	•	•	•	•
L	•	•	•	•	•	•

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
LEAD						
Orientace nástroje	-	-	-	-	-	-
Polynom orientace	-	-	-	-	-	-
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	•	•	•	•	•	•
LFOF	•	•	•	•	•	•
LFON	•	•	•	•	•	•
LFPOS	•	•	•	•	•	•
LFTXT	•	•	•	•	•	•
FWP	•	•	•	•	•	•
LIFTFAST	•	•	•	•	•	•
LIMS	•	•	•	•	•	•
LLI	•	•	•	•	•	•
LN	•	•	•	•	•	•
LOCK	•	•	•	•	•	•
LONGHOLE	-	-	-	-	-	-
LOOP	•	•	•	•	•	•

Příkazy M ... R

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
M0	•	•	•	•	•	•
M1	•	•	•	•	•	•
M2	•	•	•	•	•	•
M3	•	•	•	•	•	•
M4	•	•	•	•	•	•
M5	•	•	•	•	•	•
M6	•	•	•	•	•	•
M17	•	•	•	•	•	•
M19	•	•	•	•	•	•
M30	•	•	•	•	•	•
M40	•	•	•	•	•	•
M41 ... M45	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
M70	•	•	•	•	•	•
MASLDEF	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-
MASLON	-	-	-	-	-	-
MATCH	•	•	•	•	•	•
MAXVAL	•	•	•	•	•	•
MCALL	•	•	•	•	•	•
MEAC	-	○	○	-	○	○
MEAFRAME	•	•	•	•	•	•
MEAS	•	•	•	•	•	•
MEASA	-	○	○	-	○	○
MEASURE	•	•	•	•	•	•
MEAW	•	•	•	•	•	•
MEAWA	-	○	○	-	○	○
MI	•	•	•	•	•	•
MINDEX	•	•	•	•	•	•
MINVAL	•	•	•	•	•	•
MIRROR	•	•	•	•	•	•
MMC	•	•	•	•	•	•
MOD	•	•	•	•	•	•
MODAXVAL	•	•	•	•	•	•
MOV	•	•	•	•	•	•
MOVT	•	•	•	•	•	•
MSG	•	•	•	•	•	•
MVTOOL	•	•	•	•	•	•
N	•	•	•	•	•	•
NAMETOINT	•	•	•	•	•	•
NCK	•	•	•	•	•	•
NEWCONF	•	•	•	•	•	•
NEWMT	•	•	•	•	•	•
NEWT	-	-	-	-	-	-
NORM	•	•	•	•	•	•
NOT	•	•	•	•	•	•
NPROT	•	•	•	•	•	•
NPROTDEF	•	•	•	•	•	•
NUMBER	•	•	•	•	•	•
OEMIPO1	-	-	-	-	-	-

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
OEMIPO2	-	-	-	-	-	-
OF	•	•	•	•	•	•
OFFN	•	•	•	•	•	•
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
OR	•	•	•	•	•	•
ORIXES	-	-	-	-	-	-
ORIXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-
ORIS	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-
ORISON	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-
OS	○	○	○	○	○	○

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
OSB	○	○	○	○	○	○
OSC	-	-	-	-	-	-
OSCILL	○	○	○	○	○	○
OSCTRL	○	○	○	○	○	○
OSD	-	-	-	-	-	-
OSE	○	○	○	○	○	○
OSNSC	●	●	●	●	●	●
OSOF	-	-	-	-	-	-
OSP1	●	●	●	●	●	●
OSP2	●	●	●	●	●	●
OSS	-	-	-	-	-	-
OSSE	-	-	-	-	-	-
OST	-	-	-	-	-	-
OST1	●	●	●	●	●	●
OST2	●	●	●	●	●	●
OTOL	●	●	●	●	●	●
OVR	●	●	●	●	●	●
OVRA	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●
P	●	●	●	●	●	●
PAROT	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●
PCALL	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-
PHI	-	-	-	-	-	-
PHU	●	●	●	●	●	●
PL	-	-	-	-	-	-
PM	●	●	●	●	●	●
PO	-	-	-	-	-	-
POCKET3	-	-	-	-	-	-
POCKET4	-	-	-	-	-	-
POLF	●	●	●	●	●	●
POLFA	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●
POLY	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-
PON	-	-	-	-	-	-

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní						
○ Volitelný doplněk						
- není k dispozici						
PONS	-	-	-	-	-	-
POS	•	•	•	•	•	•
POSA	•	•	•	•	•	•
POSM	•	•	•	•	•	•
POSMT	-	-	-	-	-	-
POSP	•	•	•	•	•	•
POSRANGE	•	•	•	•	•	•
POT	•	•	•	•	•	•
PR	•	•	•	•	•	•
PREPRO	•	•	•	•	•	•
PRESETON	•	•	•	•	•	•
PRESETONS	•	•	•	•	•	•
PRIO	•	•	•	•	•	•
PRLOC	•	•	•	•	•	•
PROC	•	•	•	•	•	•
PROTA	•	•	•	•	•	•
PROTD	•	•	•	•	•	•
PROTS	•	•	•	•	•	•
PSI	-	-	-	-	-	-
PTP	•	•	•	•	•	•
PTPG0	•	•	•	•	•	•
PTPWOC	•	•	•	•	•	•
PUNCHACC	-	-	-	-	-	-
PUTFTOC	•	•	•	•	•	•
PUTFTOCF	•	•	•	•	•	•
PW	○	○	○	○	○	○
QU	•	•	•	•	•	•
R...	•	•	•	•	•	•
RAC	•	•	•	•	•	•
RDISABLE	•	•	•	•	•	•
READ	•	•	•	•	•	•
REAL	•	•	•	•	•	•
RELEASE	•	•	•	•	•	•
REP	•	•	•	•	•	•
REPEAT	•	•	•	•	•	•
REPEATB	•	•	•	•	•	•
REPOSA	•	•	•	•	•	•
REPOSH	•	•	•	•	•	•
REPOSHA	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
REPOSL	•	•	•	•	•	•
REPOSQ	•	•	•	•	•	•
REPOSQA	•	•	•	•	•	•
RESETMON	•	•	•	•	•	•
RET	•	•	•	•	•	•
RETB	•	•	•	•	•	•
RIC	•	•	•	•	•	•
RINDEX	•	•	•	•	•	•
RMB	•	•	•	•	•	•
RME	•	•	•	•	•	•
RMI	•	•	•	•	•	•
RMN	•	•	•	•	•	•
RND	•	•	•	•	•	•
RNDM	•	•	•	•	•	•
ROT	•	•	•	•	•	•
ROTS	•	•	•	•	•	•
ROUND	•	•	•	•	•	•
ROUNDUP	•	•	•	•	•	•
RP	•	•	•	•	•	•
RPL	•	•	•	•	•	•
RT	•	•	•	•	•	•
RTLIOF	•	•	•	•	•	•
RTLION	•	•	•	•	•	•

Příkazy S ... Z

Příkaz	SINUMERIK 828D					
• Standardní ○ Volitelný doplněk - není k dispozici	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
S	•	•	•	•	•	•
SAVE	•	•	•	•	•	•
SBLOF	•	•	•	•	•	•
SBLON	•	•	•	•	•	•
SC	•	•	•	•	•	•
SCALE	•	•	•	•	•	•
SCC	•	•	•	•	•	•

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
SCPARA	•	•	•	•	•	•
SD	○	○	○	○	○	○
SET	•	•	•	•	•	•
SETAL	•	•	•	•	•	•
SETDNO	•	•	•	•	•	•
SETINT	•	•	•	•	•	•
SETM	-	-	•	-	-	•
SETMS	•	•	•	•	•	•
SETMS (n)	•	•	•	•	•	•
SETMTH	•	•	•	•	•	•
SETPIECE	•	•	•	•	•	•
SETTA	•	•	•	•	•	•
SETTCOR	•	•	•	•	•	•
SETTIA	•	•	•	•	•	•
SF	•	•	•	•	•	•
SIN	•	•	•	•	•	•
SIRELAY	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-
SLOT1	-	-	-	-	-	-
SLOT2	-	-	-	-	-	-
SOFT	•	•	•	•	•	•
SOFTA	•	•	•	•	•	•
SON	-	-	-	-	-	-
SONS	-	-	-	-	-	-
SPATH	•	•	•	•	•	•
SPCOF	•	•	•	•	•	•
SPCON	•	•	•	•	•	•
SPI	•	•	•	•	•	•
SPIF1	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○
SPN	-	-	-	-	-	-
SPOF	-	-	-	-	-	-
SPOS	•	•	•	•	•	•
SPOSA	•	•	•	•	•	•
SPP	-	-	-	-	-	-
SPRINT	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
SQRT	•	•	•	•	•	•
SR	•	•	•	•	•	•
SRA	•	•	•	•	•	•
ST	•	•	•	•	•	•
STA	•	•	•	•	•	•
START	-	-	•	-	-	•
STARTFIFO	•	•	•	•	•	•
STAT	•	•	•	•	•	•
STOLF	•	•	•	•	•	•
STOPFIFO	•	•	•	•	•	•
STOPRE	•	•	•	•	•	•
STOPREOF	•	•	•	•	•	•
STRING	•	•	•	•	•	•
STRINGFELD	•	•	•	•	•	•
STRINGIS	•	•	•	•	•	•
STRLEN	•	•	•	•	•	•
SUBSTR	•	•	•	•	•	•
SUPA	•	•	•	•	•	•
SVC	•	•	•	•	•	•
SYNFCT	•	•	•	•	•	•
SYNR	•	•	•	•	•	•
SYNRW	•	•	•	•	•	•
SYNW	•	•	•	•	•	•
T	•	•	•	•	•	•
TAN	•	•	•	•	•	•
TANG	○	○	○	○	○	○
TANGDEL	○	○	○	○	○	○
TANGOF	○	○	○	○	○	○
TANGON	○	○	○	○	○	○
TCA (828D: _TCA)	•	•	•	•	•	•
TCARR	•	•	•	•	•	•
TCI	•	•	•	•	•	•
TCOABS	•	•	•	•	•	•
TCOFR	•	•	•	•	•	•
TCOFRX	•	•	•	•	•	•
TCOFRY	•	•	•	•	•	•
TCOFRZ	•	•	•	•	•	•
THETA	-	-	-	-	-	-

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
TILT	-	-	-	-	-	-
TLIFT	○	○	○	○	○	○
TML	•	•	•	•	•	•
TMOF	•	•	•	•	•	•
TMON	•	•	•	•	•	•
TO	•	•	•	•	•	•
TOFF	•	•	•	•	•	•
TOFFL	•	•	•	•	•	•
TOFFOF	•	•	•	•	•	•
TOFFON	•	•	•	•	•	•
TOFFR	•	•	•	•	•	•
TOFRAME	•	•	•	•	•	•
TOFRAMEX	•	•	•	•	•	•
TOFRAMEY	•	•	•	•	•	•
TOFRAMEZ	•	•	•	•	•	•
TOLOWER	•	•	•	•	•	•
TOOLENV	•	•	•	•	•	•
TOOLGNT	•	•	•	•	•	•
TOOLGT	•	•	•	•	•	•
TOROT	•	•	•	•	•	•
TOROTOF	•	•	•	•	•	•
TOROTX	•	•	•	•	•	•
TOROTY	•	•	•	•	•	•
TOROTZ	•	•	•	•	•	•
TOUPPER	•	•	•	•	•	•
TOWBCS	•	•	•	•	•	•
TOWKCS	•	•	•	•	•	•
TOWMCS	•	•	•	•	•	•
TOWSTD	•	•	•	•	•	•
TOWTCS	•	•	•	•	•	•
TOWWCS	•	•	•	•	•	•
TR	•	•	•	•	•	•
TRAANG	○	○	○	-	-	-
TRACON	○	○	○	-	-	-
TRACYL	○	○	○	○	○	○
TRAFOOF	•	•	•	•	•	•
TRAFOON	-	-	-	-	-	-
TRAILOF	•	•	•	•	•	•
TRAILON	•	•	•	•	•	•

21.2 Příkazy: Použitelnost u systému SINUMERIK 828D

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní						
○ Volitelný doplněk						
- není k dispozici						
TRANS	•	•	•	•	•	•
TRANSMIT	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-
TRUE	•	•	•	•	•	•
TRUNC	•	•	•	•	•	•
TU	•	•	•	•	•	•
TURN	•	•	•	•	•	•
ULI	•	•	•	•	•	•
UNLOCK	•	•	•	•	•	•
UNTIL	•	•	•	•	•	•
UPATH	•	•	•	•	•	•
VAR	•	•	•	•	•	•
VELOLIM	•	•	•	•	•	•
VELOLIMA	•	•	•	•	•	•
WAITC	•	•	•	•	•	•
WAITE	-	-	•	-	-	•
WAITENC	•	•	•	•	•	•
WAITM	-	-	•	-	-	•
WAITMC	-	-	•	-	-	•
WAITP	•	•	•	•	•	•
WAITS	•	•	•	•	•	•
WALCS0	•	•	•	•	•	•
WALCS1	•	•	•	•	•	•
WALCS2	•	•	•	•	•	•
WALCS3	•	•	•	•	•	•
WALCS4	•	•	•	•	•	•
WALCS5	•	•	•	•	•	•
WALCS6	•	•	•	•	•	•
WALCS7	•	•	•	•	•	•
WALCS8	•	•	•	•	•	•
WALCS9	•	•	•	•	•	•
WALCS10	•	•	•	•	•	•
WALIMOF	•	•	•	•	•	•
WALIMON	•	•	•	•	•	•
WHEN	•	•	•	•	•	•
WHENEVER	•	•	•	•	•	•
WHILE	•	•	•	•	•	•
WORKPIECE	•	•	•	•	•	•
WRITE	•	•	•	•	•	•

Příkaz	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
• Standardní ○ Volitelný doplněk - není k dispozici						
WRTPR	•	•	•	•	•	•
X	•	•	•	•	•	•
XOR	•	•	•	•	•	•
Y	•	•	•	•	•	•
Z	•	•	•	•	•	•

21.3 Aktuální jazyk v HMI

Následující tabulka obsahuje všechny jazyky, které jsou na uživatelském rozhraní k dispozici.

Momentálně nastavený jazyk je možné ve výrobním programu a v synchronních akcích zjistit pomocí následující systémové proměnné:

\$AN_LANGUAGE_ON_HMI = <hodnota>

<hodnota>	Jazyk	Zkratka jazyka
1	Němčina (Německo)	DEU
2	Francouzština	FRA
3	Angličtina (Spojené království)	ENG
4	Španělština	ESP
6	Italština	ITA
7	Holandština	NLD
8	Čínština (zjednodušená)	CHS
9	Švédština	SVE
18	Maďarština	HUN
19	Finština	FIN
28	Čeština	CSY
50	Portugalština (Brazílie)	PTB
53	Polština	PLK
55	Dánština	DAN
57	Ruština	RUS
68	Slovenština	SKY
72	Rumunština	ROM
80	Čínština (tradiční)	CHT
85	Korejština	KOR
87	Japonština	JPN
89	Turečtina	TRK

Poznámka

Aktualizace proměnné \$AN_LANGUAGE_ON_HMI se uskutečňuje:

- po náběhu systému
- po resetu NC systémů a/nebo PLC
- po přepnutí na jiný NC systém v rámci M2N
- po přepnutí jazyka na HMI

Přílohy

A.1 Seznam zkratek

A	
A	Výstup
ADI4	Analogové rozhraní pohonu pro 4 osy
AC	Adaptivní regulace
ALM	Modul Active Line
ARM	Asynchronní motor
AS	Automatizační systém
ASCII	American Standard Code for Information Interchange: Americká norma pro kódy při výměně informací.
ASIC	Application Specific Integrated Circuit: Integrovaný obvod pro specifickou aplikaci
ASUP	Asynchronní podprogram
AUXFU	Auxiliary Function: Pomocná funkce
AWL	Seznam příkazů
AWP	Uživatelský program

B	
BA	Provozní režim
BAG	Skupiny provozních režimů
BCD	Binary Coded Decimals: Desítková čísla vyjádřená v binárním kódu.
BERO	Bezdotykový snímač přiblížení
BI	Binektorový vstup
BICO	Binektorový konektor
BIN	Binary Files: Binární soubory
BIOS	Basic Input Output System
BCS	Základní souřadný systém
BO	Binektorový výstup
BTSS	Rozhraní ovládacího panelu

C	
CAD	Computer-Aided Design (konstrukce s podporou počítače)
CAM	Výroba s podporou počítače
CC	Compile Cycle: Cykly překladače
CEC	Cross Error Compensation (kompenzace křížové chyby)
CI	Konektorový vstup
CF karta	Kompaktní flash karta

C	
CNC	Computerized Numerical Control: Numerické řízení s počítačovou podporou
CO	Konektorový výstup
CoL	Osvědčení o licenci
COM	Komunikace
CPA	Compiler Projecting Data: Konfigurační data překladače
CRT	Cathode Ray Tube: obrazovka
CSB	Central Service Board: modul PLC
CU	Řídící jednotka
CP	Komunikační procesor
CPU	Central Processing Unit: Centrální procesorová jednotka
CR	Carriage Return
CTS	Clear To Send: Hlášení o připravenosti k odesílání u sériových datových rozhraní.
CUTCOM	Cutter Radius Compensation: Korekce rádiusu nástroje

D	
DAU	Digitálně-analogový převodník
DB	Datový modul (PLC)
DBB	Byte datového modulu (PLC)
DBD	Dvojitě slovo datového modulu (PLC)
DBW	Slovo datového modulu (PLC)
DBX	Bit datového modulu (PLC)
DDE	Dynamic Data Exchange (dynamická výměna dat)
DDS	Drive Data Set: Datový blok pohonu
DIN	Deutsche Industrie Norm (Německá průmyslová norma)
DIO	Data Input/Output: Obrazovka s informacemi o přenosu dat
DIR	Directory: Adresář
DLL	Dynamic Link Library (dynamická knihovna)
DO	Objekt pohonu
DPM	Paměť se dvěma porty
DPR	Paměť RAM se dvěma porty
DRAM	Dynamická paměť (nezálohovaná)
DRF	Differential Resolver Function: Funkce diferenčního otočného snímače (ruční kolečko)
DRIVE-CLiQ	Drive Component Link with IQ (Inteligentní propojení součástí pohonu)
DRY	Dry Run: Posuv při zkušebním zpracování
DSB	Decoding Single Block: dekódování blok po bloku
DSC	Dynamická servoregulace / Dynamická regulace tuhosti
DW	Datové slovo
DWORD	Dvojitě slovo (momentálně 32 bitů)

E	
E	Vstup
EES	Zpracování z externího paměťového média
E/A	Vstupy / výstupy
ENC	Encoder: snímač skutečné polohy
EFM	Jednoduchý periferní modul (modul V/V PLC)
EGB	Moduly/součásti ohrožené elektrostatickým výbojem
EMC	Elektromagnetická slučitelnost
EN	Evropská norma
ENC	Encoder: snímač skutečné polohy
EnDat	Rozhraní snímače
EPROM	Erasable Programmable Read Only Memory: Mazatelná, elektricky programovatelná paměť jen pro čtení
Síťové služby ePS	Služby pro dálkovou údržbu strojů s podporou internetu
EQN	Typové označení snímače absolutní hodnoty s 2048 sinusovými signály na otáčku
ES	Engineering System
ESR	Rozšířené zastavování a odjíždění
ETC	Tlačítko ETC ">" pro vyvolání rozšíření pruhu programových tlačítek v tomtéž menu

F	
FB	Funkční modul (PLC)
FC	Function Call: Funkční modul (PLC)
FEPROM	Flash-EPROM: Paměť s možností čtení a zápisu
FIFO	First In First Out: Paměť, která pracuje bez zadávání adres. Data, která jsou do ní uložena, jsou čtena ve stejné posloupnosti, v jaké byla uložena.
FIPO	Jemný interpolátor
FPU	Floating Point Unit: jednotka pracující v plovoucí řádové čárce
FRK	Korekce rádiusu frézy
FST	Feed Stop: Zastavení posuvu
FUP	Funkční schéma (metoda programování pro PLC)
FW	Firmware

G	
GC	Global Control (PROFIBUS: Vysílání telegramu)
GDIR	Globální paměť výrobních programů
GEO	Geometrie, např. geometrická osa
GIA	Gear Interpolation Data: Interpolační data převodovky
GND	Uzemnění signálu
GP	Základní program (PLC)
GS	Stupeň převodovky
GSD	Kmenový soubor zařízení pro popis jednotek typu PROFIBUS Slave

G	
GSDML	Generic Station Description Markup Language: Jazyk pro popis založený na formátu XML a určený pro sestavování souborů GSD
GUD	Global User Data: Globální uživatelská data

H	
HEX	Zkratka pro hexadecimální formát
HiFu	Pomocná funkce
HLA	Hydraulický lineární pohon
HMI	Human Machine Interface: Uživatelské rozhraní systému SINUMERIK
HSA	Pohon hlavního vřetena
HW	Hardware

I	
IBN	Uvádění do provozu
IKA	Interpolační kompenzace
IM	Interface-Modul: modul rozhraní
IMR	Interface-Modul Receive: modul rozhraní pro přijímací režim
IMS	Interface-Modul Send: modul rozhraní pro režim odesílání
INC	Increment: Velikost kroku
INI	Initializing Data: Inicializační data
IPO	Interpolátor
ISA	Mezinárodní standardní architektura
ISO	International Standard Organization (mezinárodní organizace pro normy)

J	
JOG	Jogging: Seřizovací režim

K	
K_v	Faktor zesílení regulační smyčky
K_p	Proporcionální zesílení
$K_{\bar{U}}$	Převodový poměr
KOP	Kontaktní schéma (metoda programování pro PLC)

L	
LAI	Logic Machine Axis Image: Logické zobrazení os stroje
LAN	Local Area Network
LCD	Liquid-Crystal Display: displej z tekutých krystalů
LED	Light Emitting Diode: Světelná dioda
LF	Line Feed (konec řádku)

L	
LMS	Systém pro měření polohy
LR	Regulátor polohy
LSB	Least Significant Bit: Nejméně významný bit
LUD	Local User Data: Uživatelská data (lokální)

M	
MAC	Media Access Control (Řízení přístupu k médiu)
MAIN	Main program: Hlavní program (OB1, PLC)
MB	Megabyte
MCI	Motion Control Interface (Rozhraní pro řízení pohybů)
MCIS	Informační systém pro řízení pohybů
MCP	Machine Control Panel: Řídící panel stroje
MD	Strojní parametr, příp. strojní parametry
MDA	Manual Data Automatic: Manuální zadávání
MDS	Motor Data Set: Datový blok motoru
MELDW	Slovo hlášení
MCS	Souřadný systém stroje
MM	Modul motoru
MPF	Main Program File: Hlavní program (NC)
MSTT	Řídící panel stroje

N	
NC	Numerical Control: Numerický řídicí systém pro přípravu bloků, řízením posuvů atd.
NCU	Numerical Control Unit: Hardwarová jednotka NC systému
NRK	Název operačního systému v NC systému
NST	Signál rozhraní
NURBS	Neuniformní racionální B-spliny
NV	Posunutí počátku
NX	Numerical Extension: Rozšiřovací modul osy

O	
OB	Organizační modul v PLC
OEM	Original Equipment Manufacturer (původní výrobce zařízení)
OP	Operation Panel: Zařízení pro obsluhu systému
OPI	Operation Panel Interface: Rozhraní ovládacího panelu
OPT	Options: volitelné doplňky
OLP	Optical Link Plug: Konektor pro optický kabel
OSI	Open Systems Interconnection: Spojení otevřených systémů – norma pro komunikaci mezi počítači

P	
PAA	Procesní zobrazení výstupů
PAE	Procesní zobrazení vstupů
PC	Osobní počítač
PCIN	Název programového vybavení pro výměnu dat s řídicím systémem
PCMCIA	Personal Computer Memory Card International Association: Normy pro paměťové karty počítačů
PCU	Jednotka PC: skříň PC (výpočetní jednotka)
PG	Programovací přístroj
PKE	Rozpoznávání parametrů: Součást PKW
PKW	Rozpoznávání parametrů: Hodnota (součást parametru PPO)
PLC	Programmable Logic Control: Programovatelné logické řízení
PN	PROFINET
PNO	Organizace uživatelů sběrnice PROFIBUS
PO	POWER ON
POE	Programová organizační jednotka
POS	Poloha/pozice
POSMO A	Positioning Motor Actuator: Motorový polohovací akční člen
POSMO CA	Positioning Motor Compact AC: Kompletní pohonná jednotka s integrovaným výkonovým a regulačním modulem, ale i s polohovací jednotkou a programovou pamětí; střídavé napájení
POSMO CD	Positioning Motor Compact DC: Stejně jako CA, ale se stejnosměrným napájením
POSMO SI	Positioning Motor Servo Integrated: Servomotor pro polohování, stejnosměrné napájení
PPO	Objekt parametrů a procesních dat; cyklický datový telegram při komunikaci po sběrnici PROFIBUS DP a s profilem "Pohony s měnitelnými otáčkami"
PPU	Panel Processing Unit (centrální hardware panelově orientovaného řídicího CNC systému, jako je např. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Sériová datová sběrnice
PRT	Testování programu
PSW	Programové řídicí slovo
PTP	Point to Point: od bodu k bodu
PUD	Program Global User Data: Programově globální uživatelské proměnné
PZD	Data procesů: Složka procesních dat PPO

Q	
QFK	Korekce chyby kvadrantu

R	
RAM	Random Access Memory (dynamická paměť RAM): Paměť pro čtení/zápis
REF	Funkce najíždění na referenční bod
REPOS	Funkce najíždění na původní polohu

R	
RISC	Reduced Instruction Set Computer: Typ procesor s malým instrukčním souborem a rychlým zpracováním příkazů.
ROV	Rapid Override: korekce rychlého posuvu
RP	R-Parametry, početní parametry, předem definované uživatelské proměnné
RPA	R-Parameter Active: Paměťová oblast v NC systému pro čísla R-parametrů
RPY	Roll Pitch Yaw: Způsob otáčení souřadného systému
RTLI	Rapid Traverse Linear Interpolation: Lineární interpolace při rychlém posuvu
RTS	Request To Send: Požadavek na odeslání, řídicí signál sériového rozhraní pro přenos dat.
RTCP	Řídicí protokol v reálném čase

S	
SA	Synchronní akce
SBC	Safe Break Control: Bezpečné řízení brždění
SBL	Single Block: Zpracování blok po bloku
SBR	Subroutine: Podprogram (PLC)
SD	Nastavovaný parametr, příp. nastavované parametry
SDB	Systémový datový modul
SEA	Setting Data Active: Identifikace (datový typ) pro nastavované parametry
SERUPRO	Search-Run by Program Test: Vyhledávání bloku pomocí zkušebního zpracování programu
SFB	Systémový funkční modul
SFC	Systémové volání funkce
SGE	Vstup související s bezpečností
SGA	Výstup související s bezpečností
SH	Bezpečné zastavení
SIM	Modul typu Single in Line
SK	Programové tlačítko
SKP	Skip: Funkce pro přeskokování bloků výrobního programu
SLM	Synchronní lineární motor
SM	Krokový motor
SMC	Modul snímačů namontovaný ve skříni
SME	Modul snímačů namontovaný externě
SMI	Integrovaný modul snímače
SPF	Sub Program File: Podprogram (NC)
SPS	Řídicí systém s programovatelnou pamětí = PLC
SRAM	Statická paměť (se zálohovaným napájením)
SRK	Korekce rádiusu břitu
SRM	Synchronní motor
SSFK	Korekce chyby stoupání vřetena
SSI	Serial Synchron Interface: Synchronní sériové rozhraní
SSL	Vyhledávání bloku
STW	Řídicí slovo

S	
SUG	Obvodová rychlost kotouče
SW	Software
SYF	System Files: Systémové soubory
SYNACT	Synchronized Action: Synchronní akce

T	
TB	Připojovací deska (SINAMICS)
TCP	Tool Center Point: Špička nástroje
TCP/IP	Protokol pro řízení přenosu / Internetový protokol
TCU	Thin Client Unit (Jednotka klienta)
TEA	Testing Data Active: Identifikace strojních parametrů
TIA	Totally Integrated Automation (Úplně integrovaná automatizace)
TM	Připojovací modul (SINAMICS)
TO	Tool Offset: Korekční parametry nástroje
TOA	Tool Offset Active: Označení (typ souboru) pro korekční parametry nástroje
TRANSMIT	Transform Milling Into Turning: Transformace souřadného systému pro obrábění frézováním na soustruhu
TTL	Transistor–Transistor–Logik (typ rozhraní)
TZ	Technologický cyklus

U	
UFR	User Frame: Posunutí počátku
UP	Podprogram
USB	Universal Serial Bus (Univerzální sériové rozhraní)
USV	Nepřerušitelný zdroj napájení

V	
VDI	Interní komunikační rozhraní mezi NC systémem a PLC
VDI	Verein Deutscher Ingenieure (Sdružení německých techniků)
VDE	Verband Deutscher Elektrotechniker (Svaz německých elektrotechniků)
VI	Napěťový vstup
VO	Napěťový výstup
VSA	Pohon posuvu

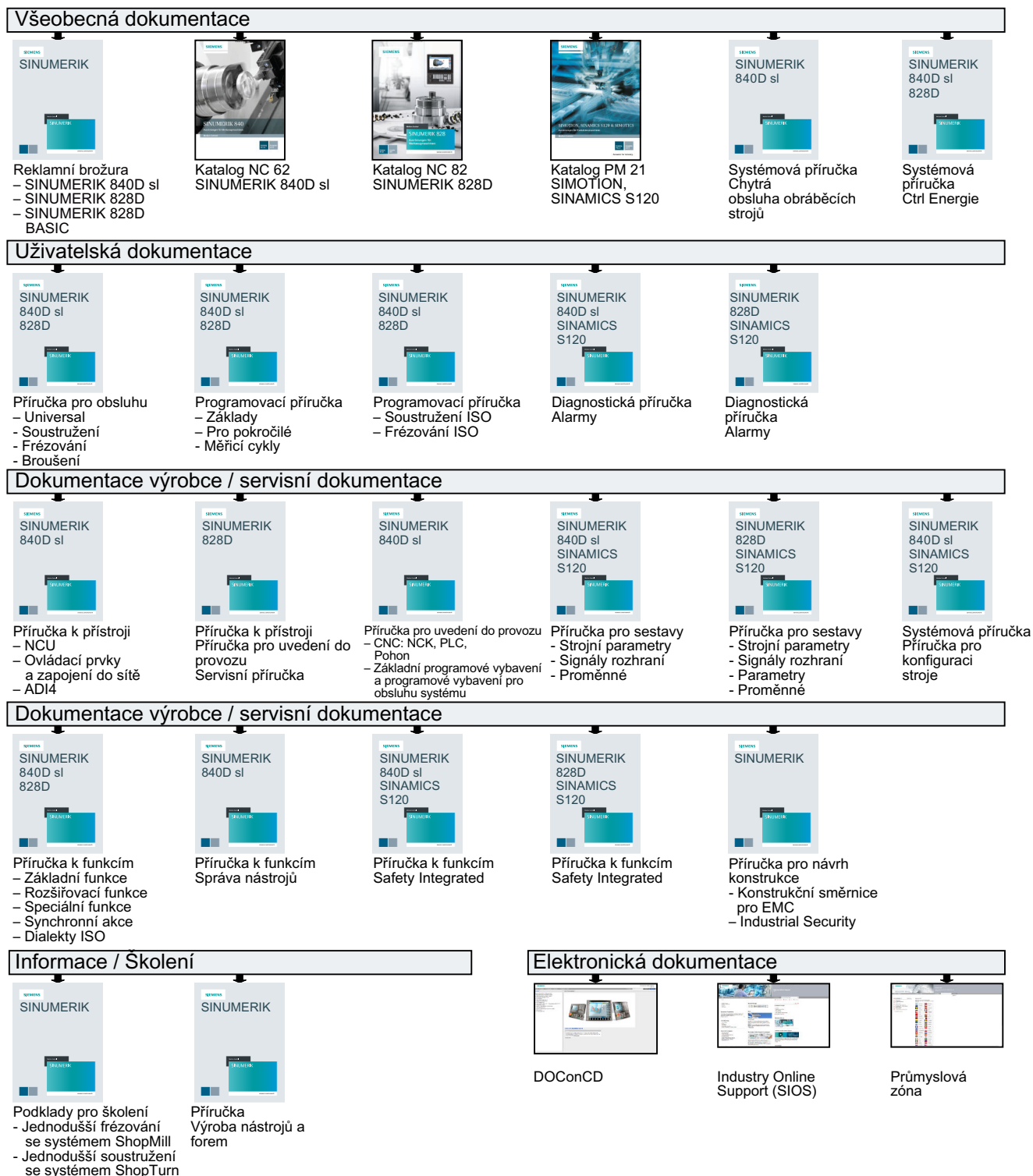
W	
WAB	Funkce měkkého najíždění a odjíždění
WCS	Souřadný systém obrobku
WKZ	Nástroj
WLK	Korekce délky nástroje
WOP	Dílensky orientované programování

W	
WPD	Work Piece Directory: Adresář obrobku
WRK	Korekce rádiusu nástroje
WZ	Nástroj
WZK	Korekční parametry nástroje
WZV	Správa nástrojů
WZW	Výměna nástroje

X	
XML	Extensible Markup Language (Počítačový jazyk)

Z	
ZOA	Zero Offset Active: Identifikace posunutí počátku
ZSW	Stavové slovo (pohonu)

A.2 Přehled dokumentace



Glosář

Absolutní rozměry

Udání cíle pohybu osy prostřednictvím údaje, který je vztažen na počátek momentálně platného souřadného systému. Viz -> Inkrementální rozměr

Adresa

Adresa je identifikátor pro určitý operand nebo rozsah operandů, např. vstup, výstup atd.

Adresa osy

Viz --> Název osy

Alarmy

Všechna --> hlášení a alarmy se vypisují prostým textem na ovládacím panelu spolu s datem, časem a odpovídajícím 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 ke zobrazení prostým textem přímo z výrobního programu.
2. Alarmy a hlášení z PLC
Alarmy a hlášení stroje se mohou přenášet ke zobrazení prostým textem z programu PLC. Za tím účelem nejsou zapotřebí žádné doplňkové funkční moduly.

Archivace

Odesílání dat a/nebo adresářů do **externího** paměťového zařízení.

Asynchronní podprogram

Výrobní program, který může být spuštěn asynchronně, tedy nezávisle na aktuálním stavu jiného programu signálem přerušení (např. signál "rychlejší vstup NC systému").

Automatický režim

Provozní režim řídicího systému (režim zpracovávání posloupnosti bloků podle DIN): Provozní režim NC-systémů, ve kterém je zvolen --> výrobní program a ten je kontinuálně zpracováván.

Baudrate

Rychlost přenosu dat (bitů/s).

Bezpečnostní funkce

Řídicí 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 spustil alarm CNC.

Blok výrobního programu

Část --> výrobního programu, která je vymezena znaky Line Feed. Jsou rozlišovány --> hlavní bloky a --> vedlejší bloky.

Celkový reset

V případě celkového resetu jsou z --> CPU vymazány následující paměti:

- --> Pracovní paměť
- oblasti pro čtení a zápis --> paměti pro načítání
- --> Systémová paměť
- --> Zálohovaná paměť

CNC

Viz --> NC

Computerized Numerical Control: zahrnuje komponenty → NC, → PLC, HMI, → COM.

CNC

Viz --> NC

Computerized Numerical Control: zahrnuje komponenty → NC, → PLC, HMI, → COM.

COM

Součást řídicího systému NC pro uskutečňování a koordinaci komunikace.

CPU

Central Processor Unit, --> Centrální procesorová jednotka

C-Spline

C-spline je nejznámějším a nejčastěji používaným splinem. Přechody mezi uzlovými body mají spojitou tečnu a zakřivení. Používají se polynomy 3. stupně.

Cykly

Cykly jsou chráněné podprogramy pro uskutečňování opakovaně se vyskytujících obráběcích procesů na --> obrobcích.

Časově reciproční posuv

Namísto rychlosti posuvu pro pohyb osy může být naprogramován také čas, za jaký se má úsek dráhy v bloku urazit (G93).

Datové slovo

Datová jednotka o velikosti dva byty v --> datovém modulu.

Datový modul

1. Datová jednotka --> PLC, ke které mají přístup programy --> HIGHSTEP.
2. Datová jednotka --> NC systému: Datové moduly obsahují definice pro globální uživatelská data. Data mohou být při své definici přímo inicializována.

Definice proměnných

Definice proměnné zahrnuje stanovení datového typu a názvu proměnné. Pomocí názvu proměnné je přístup k hodnotě proměnné.

Diagnostika

1. Systémová oblast řídicího systému
2. Řídicí systém obsahuje jak program pro diagnostiku sebe sama, tak také zkušební nástroje pro servis: Stavové, alarmové a servisní obrazovky.

Dráhová osa

Dráhové osy jsou všechny osy podílející se na obrábění v --> kanálu, které jsou --> interpolátorem ovládány tak, aby byly současně spouštěny, urychlovány, zastavovány a naváděny do koncového bodu.

DRF

Differential Resolver Function: Funkce NC systému, která ve spojení s elektronickým ručním kolečkem vytváří v režimu "Auto" 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.

Editor

Editor umožňuje sestavování, upravování, doplňování, kompresi a vkládání programů/textů/programových bloků.

Externí posunutí počátku

Posunutí počátku specifikované --> PLC.

Frame

Frame představuje matematický předpis, který převádí jeden kartézský souřadný systém do jiného kartézského souřadného systému. Frame obsahuje tyto komponenty: --> posunutí počátku, --> otočení, --> změna měřítka, --> zrcadlové převrácení.

Geometrická osa

Geometrické osy tvoří 2-, příp. 3-rozměrný --> souřadný systém obrobku, ve kterém je ve --> výrobním programu naprogramována geometrie obrobku.

Geometrie

Popis --> obrobku v --> souřadném systému obrobku.

HIGHSTEP

Shrnutí programovacích možností pro --> PLC systému AS300/AS400.

Hlášení

Všechna hlášení naprogramovaná v programu pro výrobu součástí a systémem rozpoznané --> alarmy se vypisují na řídicím 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 začínající znakem „:“, který obsahuje všechny příkazy, které jsou zapotřebí pro spuštění pracovního postupu ve --> výrobním programu.

Hlavní program

Označení hlavní program pochází ještě z dob, kdy byly výrobní programy pevně rozděleny na hlavní programy a --> podprogramy. Toto pevné rozdělení v dnešním jazyku systému SINUMERIK už neexistuje. V principu může být kterýkoli výrobní program v kanálu zvolen a spuštěn. To se potom uskutečňuje na --> programové úrovni 0 (úroveň hlavního programu). V hlavním programu mohou být jako podprogramy vyvolávány další výrobní programy nebo --> cykly.

Hodnota kompenzace

Rozdíl mezi polohou osy zjištěnou měřicím snímačem a požadovanou naprogramovanou polohou osy.

Chráněný prostor

Trojrozměrný prostor v rámci --> pracovního prostoru, do kterého se špička nástroje nesmí dostat.

Identifikátor

Slova podle normy DIN 66025 jsou doplňována identifikátory (názvy) pro proměnné (početní proměnné, systémové proměnné, uživatelské proměnné), pro podprogramy, pro klíčová slova a slova s více adresovými písmeny. Tato doplnění mají při sestavování bloku stejný význam jako slova. Identifikátor musí být jednoznačný. Stejný identifikátor se nesmí používat pro různé objekty.

Interpolace spliny

Pomocí splinové interpolace je řídicí systém schopen pouze na základě několika předem zadaných opěrných bodů vytvořit požadovanou konturu s hladkým křivkovým průběhem.

Interpolační kompenzace

Prostřednictvím interpolačních kompenzací, jako jsou → kompenzace chyby stoupání vřetena, průhybu, úhlové odchylky a teplotní kompenzace jsou mechanické chyby stroje kompenzovány.

Interpolátor

Logická jednotka --> NC systému, která po zadání cílové pozice ve výrobním programu stanoví pomocné hodnoty pro jednotlivé osy odpovídající pohybu, který je potřeba uskutečnit.

Jednotka TOA

Každá --> oblast TOA může obsahovat větší počet jednotek TOA. Počet možných jednotek TOA je omezen maximálním možným počtem aktivních --> kanálů. Jednotka TOA obsahuje právě jeden datový modul nástrojů a jeden datový modul zásobníku. Kromě toho může obsahovat ještě i jeden datový modul držáku nástroje (volitelné).

JOG

Provozní režim řídicího systému (seřizování): V provozním režimu JOG je možné provádět seřizování stroje. Jednotlivými osami a vřeteny je možné pohybovat pomocí směrových tlačítek v tipovacím režimu. Dalšími funkcemi v provozním režimu JOG jsou --> najíždění na referenční bod, --> Repos a --> Preset (dosazení skutečné hodnoty).

Kanál

Kanál se vyznačuje tím, že může zpracovávat --> výrobní program nezávisle na jiných kanálech. Kanál řídí výlučně osy a vřetena, která mu byla přiřazena. Programové postupy různých kanálů mohou být prostřednictvím --> synchronizace koordinovány.

Kanál pro zpracování

Prostřednictvím kanálové struktury mohou být zkráceny jalové časy, neboť pohybové operace mohou probíhat paralelně, např. posuv podavače souběžně s obráběním. Na kanál CNC je přitom možno pohlížet jako na samostatný CNC řídicí systém s dekodováním, přípravou bloků a interpolací.

Klíč programátora

Znaky a posloupnosti znaků, které v programovacím jazyku pro --> výrobní programy mají pevně definovaný význam.

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 na 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

Vyrovňování mechanické nepřesnosti vřetena podílejícího se na posuvu prováděné řídicím systémem na základě změřených hodnot odchylek.

Kompenzace vůle

Vyrovňování mechanických vůlí stroje, např. 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 HW

Nástroj SIMATIC S7 pro nastavování konfigurace a parametrů hardwarových komponent v rámci projektů typu S7.

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 obrobku

Kontura nahotovo obrobeného obrobku. Viz --> Surový obrobek.

Kontura obrobku

Požadovaná kontura vyráběného/obráběného --> obrobku.

Korekce rádiusu bříty

Při programování kontury se vychází z toho, že nástroj je špičatý. Jelikož toto v praxi není realizovatelné, zadává se do řídicího 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.

Korekce rádiusu nástroje

Abyste mohli požadovanou --> konturu obrobku přímo naprogramovat, musí řídicí systém pohybovat nástrojem po ekvidistanční dráze vzhledem ke kontuře, přičemž musí znát přesný rádius použitého nástroje (G41/G42).

Korekční parametry nástroje

Zohledňování rozměrů nástroje při výpočtu dráhy.

Kostrá

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

Kruhová osa uskutečňuje otočení obrobku nebo nástroje do předem definované úhlové polohy.

KÜ

Převodový poměr

KV

Faktor zesílení smyčky, regulační charakteristika regulačního obvodu.

Lineární osa

Lineární osa je osa, která oproti kruhové ose opisuje přímku.

Look Ahead

Pomocí funkce **Look Ahead** řídicí systém vyhodnocuje několik bloků dopředu (tento počet lze nastavit pomocí parametru), čímž se dosahuje optimální rychlosti při zpracování.

MDA

Provozní režim řídicího 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) se řídicí systém převede na základní systém.

Měřicí systém využívající palce

Měřicí systém, který vzdálenosti udává v „palcích“ a jejich zlomcích.

Metrický systém měřicích jednotek

Normovaný systém využívající jednotky: pro délky např. mm (milimetr), m (metr).

Mez přesného najetí

Pokud všechny dráhové osy dosáhnou své meze přesného najetí, řídicí systém se chová, jako by bylo cílového bodu přesně dosaženo. Uskuteční se přechod na další blok --> výrobního programu.

Mezní hodnota otáček

Maximální/minimální otáčky (vřetena): Zadáním strojních parametrů, parametrů --> PLC, nebo --> nastavovaných parametrů mohou být maximální otáčky vřetena omezeny.

Modul

Pojmem "moduly" jsou označovány všechny soubory, které jsou zapotřebí pro vytváření a zpracovávání programů.

Najíždění na pevný bod

Obráběcí stroje mohou definovaným způsobem najíždět na pevné body, jako je např. bod pro výměnu nástroje, základací bod, bod pro výměnu palety atd. Souřadnice těchto bodů jsou uloženy v řídicím systému. Pokud je to možné, řídicí systém pohybuje příslušnými osami --> rychlým posuvem.

Nastavované parametry

Parametry, které definovaným způsobem systémovým programovým vybavením zprostředkovávají řídicímu systému NC vlastnosti obráběcího stroje.

Nástroj

Pracovní součást na obráběcím stroji, která způsobuje obrábění, např. soustružnický nůž, fréza, vrták, laserový paprsek ...).

Název osy

Kvůli jednoznačné identifikaci musí mít všechny kanálové a --> strojní osy v rámci řídicího systému přiřazeny názvy, které jsou jednoznačné v rámci příslušného kanálu, resp. řídicího systému. --> Geometrické osy mají označení X, Y, Z. --> Kruhové osy, které se okolo těchto geometrických os otáčejí, mají názvy A, B, C.

NC

Součást Numerical Control (číslicové řízení) → CNC systému, která zpracovává → výrobní program a koordinuje pohybové operace obráběcího stroje.

NRK

Numerický robotický kernel (operační systém --> NC).

NURBS

V rámci řídicího systému prováděné interní vedení pohybu a dráhová interpolace se uskutečňují na základě NURBS (**Ne**Uniformní **R**acionální **B**-Spliny). Díky tomu je v řídicím systému k dispozici jednotné chování pro všechny interpolace.

Oblast TOA

V oblasti TOA jsou soustředěna všechna data nástrojů a zásobníků. Pokud jde o dosažitelnost těchto dat, standardně se oblast kryje s oblastí --> kanálu. Pomocí strojních parametrů je však možné nastavit, že několik kanálů --> jednotku TOA sdílí, takže tyto kanály potom mohou mít k dispozici společná data nástrojů.

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.

OEM

Pro výrobce stroje, který si přeje v řídicím systému instalovat své vlastní uživatelské rozhraní nebo specifické technologické funkce, existuje prostor pro individuální řešení (aplikace OEM).

Ohraničení pracovního pole

Navíc kromě koncových spínačů může být rozsah pohybu os dále omezen pomocí ohraničení pracovního pole. Pro každou osu může existovat dvojice hodnot, která chráněný pracovní prostor popisuje.

Orientované zastavení vřetena

Zastavení vřetena obrobku v předem definované úhlové poloze, např. aby bylo možné uskutečnit další obrábění na určitém místě.

Osa C

Osa, okolo které se uskutečňuje řízený otočný pohyb a polohování s nástrojovým vřetenem.

Osy

Osy CNC jsou v závislosti na spektru svých funkcí rozděleny do těchto kategorií:

- Osy: interpolační dráhové osy
- Pomocné osy: Přířuvné a polohovací osy bez interpolace a se specifickým osovým posuvem. Pomocné osy se nepodílejí na vlastním obrábění, např. jsou to podavače nástroje, zásobník nástrojů atd.

Osy stroje

Fyzicky existující osy v obráběcím stroji.

Otočení

Složka --> framu, která definuje otočení souřadného systému o určitý úhel.

Override

Manuální, příp. programovatelná možnost zásahu, která obsluhujícímu pracovníkovi umožňuje změnit naprogramované posuvy nebo otáčky, aby je bylo možné přizpůsobit určitému obrobku či materiálu.

Override posuvu

Naprogramovaná rychlost je nahrazena aktuálním nastavením rychlosti uskutečněným pomocí --> řídicího panelu stroje nebo na --> PLC (0-200%). Rychlost posuvu může být dodatečně měněna v programu pro opracování součásti prostřednictvím programovatelného procentuálního faktoru (1 – 200 %).

Paměť korekcí

Datová oblast řídicího systému, ve které jsou uloženy korekční parametry nástroje.

Paměť pro načítání

Paměť pro načítání se u CPU 314 systému --> SPS rovná --> pracovní paměti.

Periferní modul

Periferní moduly vytvářejí spojení mezi CPU a procesem.

Jedná se o následující:

- --> Moduly digitálních vstupů/výstupů
- --> Moduly analogových vstupů/výstupů
- --> Simulační moduly

Pevný bod stroje

Bod jednoznačně definovaný obráběcím strojem, např. referenční bod stroje.

PLC

Programmable Logic Control: --> Řídicí systém s programovatelnou pamětí. Komponenty řídicího --> NC systému: Přizpůsobení řídicího systému pro řídicí logiku obráběcího stroje.

Počátek souřadného systému obrobku

Počátek (nula) --> souřadného systému obrobku tvoří výchozí bod této soustavy. Je definován vzdáleností od počátku --> souřadné soustavy stroje.

Počátek souřadného systému stroje

Pevný bod obráběcího stroje, na který jsou vztaženy všechny (odvozené) měřicí systémy.

Podprogram

Označení podprogram pochází ještě z dob, kdy byly výrobní programy pevně rozděleny na --> hlavní programy a podprogramy. Toto pevné rozdělení v dnešním jazyku systému SINUMERIK už neexistuje. V principu může být každý výrobní program nebo každý --> cyklus vyvolán jako podprogram v rámci nějakého jiného výrobního programu. To se potom uskutečňuje na --> programové úrovni (x+1) (úroveň podprogramu (x+1)).

Pohon

Pohon je tou jednotkou CNC systému, která na základě dat z NC systému uskutečňuje regulaci otáček a momentu.

Polární souřadnice

Souřadný systém, ve kterém je poloha bodu v rovině dána vzdáleností od počátku a úhlem, který svírá vektor radiusu 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.

Polynomická interpolace

Pomocí polynomické interpolace mohou být konstruovány křivky rozmanitých průběhů, jako jsou **přímka**, **parabola**, **mocnná funkce** atd. (SINUMERIK 840D sl).

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 řídicí 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 (nulu) a --> frame.

1. Nastavitelné
K dispozici je určitý v konfiguraci definovaný počet nastavitelných posunutí počátku pro každou CNC osu. Alternativně lze používat posunutí aktivovaná pomocí příkazů G-funkcí.
2. Externí
Navíc na všechna posunutí, jež definují polohu souřadného systému obrobku, může být aplikována korekce externím posunutím počátku pomocí ručního kolečka (posunutí DRF) nebo 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.

Pracovní paměť

Pracovní paměť je paměť typu RAM v --> CPU, do níž má přístup procesor během zpracování uživatelského programu.

Pracovní prostor

Trojrozměrný prostor, v němž se na základě konstrukce obráběcího stroje může pohybovat špička nástroje. Viz také --> Chráněný prostor

Program pro přenos dat PCIN

PCIN je pomocný program pro odesílání a přijímání uživatelských dat CNC přes sériové rozhraní, jako jsou např. výrobní programy, korekční parametry nástroje atd. Program PCIN se může spouštět pod MS-DOSem na standardních průmyslových PC.

Programová paměť PLC

SINUMERIK 840D sl: V uživatelské paměti PLC jsou společně uloženy uživatelský program PLC a uživatelská data a základní program PLC.

Programová úroveň

Výrobní program spuštěný v kanálu je zpracováván jako --> hlavní program na programové úrovni 0 (úroveň hlavního programu). Každý výrobní program, který je vyvoláván v hlavním programu, je zpracováván jako --> podprogram na své vlastní programové úrovni 1 ... n.

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 pole

Ohraničení pracovního 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.

Provozní režim

Pojem označující způsob fungování řídicího systému SINUMERIK. Jsou definovány provozní režimy --> Jog, --> MDA, --> Auto.

Předběžná koincidence

K přechodu na další blok dochází už tehdy, když se pohyb po dráze dostane do blízkosti koncového bodu, takže vzdálenost od něj je menší než předem zadaná hodnota delta.

Přepínač na klíč

Přepínač na klíč na ovládacím panelu stroje má 4 polohy, které jsou obsazeny funkcemi operačního systému řídicího systému. K přepínači na klíč patří tři různě barevné klíče, které je možné vytáhnout v dále uvedených polohách.

Přesné najetí

Při programovatelném příkazu přesného najetí se na pozici uvedenou v bloku najíždí přesně a v případě potřeby velmi pomalu. Pro zkrácení doby přibližování jsou pro rychlý a pracovní posuv definovány --> meze přesného najetí.

Přímková interpolace

Nástroj se pohybuje po přímkách k cílovému bodu a přitom opracovává obrobek.

Referenční bod

Bod obráběcího stroje, na který je vztažen měřicí systém os stroje.

Režim řízení pohybu po dráze

Cílem řízení pohybu po dráze je zabránit velkým bržděním --> dráhových os na hranicích bloků ve výrobním programu a přecházet do následujícího bloku pokud možno se stejnou rychlostí pohybu po dráze.

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-Parametry

Početní parametry, mohou být programátorem --> výrobního programu použity pro libovolné účely v programu nebo mohou být zjišťovány jejich hodnoty.

RS-232

Sériové rozhraní pro vstup/výstup dat. Pomocí tohoto rozhraní můžete načítat, odesílat a zálohovat výrobní programy, jakož i data výrobce a uživatelská data.

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.

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

Rychlost pohybu po dráze

Maximální naprogramovatelná dráhová rychlost závisí na jemnosti zadávané hodnoty. Například při rozlišení 0,1 mm činí maximální programovatelná dráhová rychlost 1000 m/min.

Rychlý posuv

Nejvyšší rychlost pohybu osy. Použije se např. tehdy, je-li potřeba nástrojem v klidu najet na --> konturu obrobku nebo od kontury obrobku odjet. Rychlost rychlého posuvu je nastavena prostřednictvím strojního parametru specificky pro daný stroj.

Řetězové kótování

Též inkrementální rozměr: Stanovení cíle pohybu osy pomocí dráhy a směru, které je potřeba urazit, vztažené na již dosažený bod. Viz --> Absolutní rozměr.

Řídící osa

Řídící osa je osa --> gantry, která je z pohledu obsluhujícího pracovníka a programátora k dispozici a která může být v důsledku toho odpovídajícím způsobem ovlivňována stejně jako normální osa NC systému.

Řídící panel stroje

Řídící panel obráběcího stroje s ovládacími prvky, jako jsou tlačítka, otočné přepínače atd. a s jednoduchými signalizačními prvky, jako jsou světelné diody. Slouží k bezprostřednímu ovlivňování obráběcího stroje pomocí PLC.

Řídící systém s programovatelnou pamětí

Paměťové programovatelné řídicí systémy (SPS) jsou elektronické řídicí systémy, jejichž funkce je uložena ve formě programu v paměťovém zařízení. Konstrukce a zapojení zařízení tedy nezávisí na funkci řídicího systému. Paměťové programovatelné řídicí systémy mají konstrukci počítače: skládají se z CPU (centrální modul) s pamětí, modulů vstupů/výstupů a interního sběrnicevého systému. Periferie a programovací jazyk jsou podřízeny potřebám řízení.

Řízení podle 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 (funkce --> Look Ahead).

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.

Skupiny provozních režimů

Osy a vřetena, které k sobě po technologické stránce patří, mohou být soustředěny do skupiny provozních režimů (BAG). Osy/vřetena v jedné skupině provozních režimů mohou být řízeny prostřednictvím jednoho nebo více --> kanálů. Kanálům ve skupině provozních režimů je vždy přiřazen stejný --> 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 (nulou) vztažen na obrobek. Při programování v souřadném systému obrobku jsou rozměry a směry vztaženy na tento systém.

Souřadný systém stroje

Souřadný systém, který je vztažen 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:

- Kruhový pohyb v rovině
- Lineární pohyb kolmo na tuto rovinu

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 alfanumerických znaků.

Standardní cykly

Pro často se opakující obráběcí operace jsou k dispozici standardní cykly:

- Pro technologie vrtání/frézování
- Pro technologii soustružení

V systémové oblasti „Program“ pod menu „Podpora cyklů“ se nachází seznam cyklů, které jsou Vám k dispozici. Po aktivování požadovaného obráběcího cyklu se srozumitelným textem vypíše potřebné parametry, jimž je potřeba přiřadit odpovídající hodnoty.

Surový obrobek

Součást, na které má být zahájeno opracovávání obrobku.

Synchronizace

Příkazy na určitých místech ve --> výrobním programu pro koordinaci operací v různých --> kanálech.

Synchronizovaná osa

Synchronizovaná osa je osa --> gantry, jejíž požadovaná poloha je neustále odvozena od pracovních posuvových pohybů --> vodící osy a která se proto pohybuje synchronizovaně. Z pohledu obsluhujícího pracovníka a programátora "není" synchronizovaná osa "k dispozici".

Synchronizované osy

Synchronní osy potřebují pro provedení svého pohybu stejný čas, jaký potřebuje geometrická osa pro svůj pohyb po dráze.

Synchronní akce

1. Výstup pomocné funkce
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. Rychlé výstupy pomocných funkcí
Pro časově kritické spínací funkce mohou být minimalizovány potvrzovací časy (--> pomocné funkce). Zbytečné body pozastavení jsou z obráběcího procesu odstraněny.

Systémová paměť

Systémová paměť je paměť v NCU, do které se ukládají následující data:

- Data, která potřebuje řídicí systém
- Operandy časovačů, počítadel a ukazatelů

Systémové proměnné

Proměnná, která existuje bez přičinění programátora výrobního programu. Je definována svým datovým typem a názvem proměnné, který začíná znakem \$. Viz také --> Uživatelská proměnná.

Technika maker

Shrnutí většího počtu příkazů do jednoho identifikátoru. Tento identifikátor v programu reprezentuje tento daný počet soustředěných příkazů.

Textový editor

Viz --> Editor

Transformace

Aditivní nebo absolutní posunutí počátku v jedné ose.

Uživatelem definované proměnné

Uživatel může pro libovolné využití ve --> výrobním programu nebo v datovém modulu (globální uživatelská data) definovat uživatelské proměnné. Definice obsahuje udání datového typu a název proměnné. Viz také --> Systémová proměnná.

Uživatelská paměť

Všechny programy a data, jako jsou výrobní programy, podprogramy, komentáře, korekční parametry nástroje, posunutí počátku/frame, jakož i kanálová a programová uživatelská data, mohou být společně uloženy v uživatelské paměti CNC systému.

Uživatelské rozhraní

Pracovní plocha je zobrazovací médium CNC řídicího systému představovaná displejem. Zobrazuje se s programovými tlačítky ve vodorovném a svislém pruhu.

Uživatelský program

Uživatelské programy pro automatizační systémy S7-300 jsou vytvářeny v programovacím jazyku STEP 7. Uživatelský program je strukturovaný a skládá se z jednotlivých modulů.

Základní typy modulů jsou:

- Modul kódů
Tyto moduly obsahují příkazy jazyka STEP 7.
- Datový modul
Tyto moduly obsahují konstanty a proměnné pro programy v jazyce STEP 7.

Vedlejší blok

Blok začínající „N“ a obsahující informace pro krok pracovního postupu, např. udání polohy.

Velikost kroku

Udání délky posuvu pomocí počtu inkrementů (velikost kroku). Počet inkrementů může být uložen jako nastavovaný parametr, příp. může být zvolen pomocí tlačítek s odpovídajícím popisem 10, 100, 1000, 10000.

Vrtání závitů bez vyrovnávací hlavičky

Pomocí této funkce můžete vyrábět 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).

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 NC řídicí systém, který zabezpečí celkové opracování určitého obrobku. Také uskutečnění určitého opracování na zadaném --> surovém obrobku.

Vyšší programovací jazyk CNC

Vyšší programovací jazyk slouží k zapisování NC programů, → synchronních akcí a → cyklů. Nabízí: Řídicí struktury, --> uživatelem definované proměnné, --> systémové proměnné, --> makra.

WinSCP

WinSCP je Open Source Program pro Windows sloužící pro přenášení souborů, který je volně k dispozici.

Základní osa

Osa, na kterou jsou vztaženy požadovaná nebo skutečná hodnota za účelem výpočtu hodnoty kompenzace.

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, existuje paralelně k --> souřadnému systému stroje. Liší se od něho v --> názvech os.

Zakřivení

Zakřivení k kontury je inverzní hodnota radiusu r oskulační kružnice v daném bodě kontury ($k = 1/r$).

Záložní baterie

Záložní baterie zaručuje, že --> uživatelský program v --> CPU je chráněn proti výpadku napájení a že definované datové oblasti a značky, časy a čísla zůstanou nezměněny.

Zaokrouhlovací osa

Zaokrouhlovací osa uskutečňuje otočení obrobku nebo nástroje do odpovídající úhlové polohy v dělicí mřížce. Při dosažení mřížky je zaokrouhlovací osa "na svém místě".

Zavádění

Načítání systémových programů po zapnutí.

Změna měřítka

Komponent --> framu, který způsobuje změnu měřítka pro určitou osu.

Zrcadlové převrácení

Při zrcadlovém převrácení jsou znaménka hodnot souřadnic osy vztahující se k dané kontuře vyměněna. Současně je možné zrcadlově převrátit i několik os.

Zrychlení s omezením ryvu

Aby bylo dosaženo optimálního průběhu zrychlení stroje a aby se současně šetřila jeho mechanika, je možné ve výrobním programu přepínat mezi skokovým a spojitým (bez trhnutí) zrychlováním.

Rejstřík

\$

\$AA_ATOL, 559
\$AA_COUP_ACT
 při vlečení, 566
 v případě axiální vazby řídicí hodnotou, 590
\$AA_ESR_ENABLE, 685
\$AA_LEAD_SP, 590
\$AA_LEAD_SV, 590
\$AC_ACT_PROG_NET_TIME, 673
\$AC_ACTUAL_PARTS, 676
\$AC_AXCTSWA, 658
\$AC_AXCTSWE, 658
\$AC_CTOL, 559
\$AC_CUT_INV, 483
\$AC_CUTMOD, 482
\$AC_CUTMOD_ANG, 482
\$AC_CUTMODK, 482
\$AC_CUTTING_TIME, 672
\$AC_CYCLE_TIME, 672
\$AC_DELAYFST, 540
\$AC_ESR_TRIGGER, 685
\$AC_OLD_PROG_NET_TIME, 673
\$AC_OLD_PROG_NET_TIME_COUNT, 673
\$AC_OPERATING_TIME, 672
\$AC_OTOL, 559
\$AC_PROG_NET_TIME_TRIGGER, 673
\$AC_REPOS_PATH_MODE, 549
\$AC_REQUIRED_PARTS, 676
\$AC_SMAXVELO, 554
\$AC_SMAXVELO_INFO, 554
\$AC_SPECIAL_PARTS, 676
\$AC_TOTAL_PARTS, 676
\$AC_TRAFO_CORR_ELEM_P, 414
\$AC_TRAFO_CORR_ELEM_T, 414
\$AC_TRAFO_ORIAX_LOC, 414
\$AN_AXCTAS, 658
\$AN_AXCTSWA, 658
\$AN_ESR_TRIGGER, 685
\$AN_LANGUAGE_ON_HMI, 941
\$AN_POWERON_TIME, 672
\$AN_SETUP_TIME, 672
\$NT_CLOSE_CHAIN_T, 414
\$NT_CNTRL, 414
\$NT_CORR_ELEM_P, 413
\$NT_CORR_ELEM_T, 413
\$NT_NAME, 405
\$NT_ROT_AX_NAME, 478

\$NT_TRAFO_INDEX, 405
\$P_ACTBFRAME, 320
\$P_AD, 482
\$P_BFRAME, 319
\$P_CTOL, 559
\$P_CUT_INV, 483
\$P_CUTMOD, 482
\$P_CUTMOD_ANG, 482
\$P_CUTMOD_ERR, 483
\$P_CUTMODK, 482
\$P_DELAYFST, 540
\$P_CHBFRAME, 319
\$P_CHBFRMASK, 320
\$P_IFRAME, 321
\$P_IS_EES_PATH, 226
\$P_NCBFRAME, 319
\$P_NCBFRMASK, 320
\$P_ORI_DIFF, 473
\$P_ORI_POS, 473
\$P_ORI_SOL, 474
\$P_ORI_STAT, 477
\$P_OTOL, 559
\$P_PATH, 225
\$P_PFRAME, 321
\$P_PROG, 225
\$P_PROGPATH, 225
\$P_SIM, 287
\$P_STACK, 225
\$P_SUBPAR, 164
\$P_TOOLENV, 492
\$P_TOOLENVN, 492
\$PA_ATOL, 559
\$SA_LEAD_TYPE, 590
\$SC_CONTPREC, 533
\$SC_MINFEED, 533
\$SC_PA_ACTIV_IMMED, 238
\$SN_PA_ACTIV_IMMED, 238
\$TC_CARR1...14, 458
\$TC_CARR18...23, 458
\$TC_CARR18[m], 462
\$TC_DP1 ... 25, 415
\$TC_ECPxy, 419
\$TC_SCPxy, 419

*

* (matematická funkce), 76

- /
/ (matematická funkce), 76
- +
- + (matematická funkce), 76
- <
- < (relační operátor), 78
<< (operátor zřetězení), 88
<= (relační operátor), 78
<> (relační operátor), 78
- =
- == (relační operátor), 78
- >
- > (relační operátor), 78
>= (relační operátor), 78
- A**
- ABS, 76
ACCLIMA, 528
ACOS, 76
ACTBLOCNO, 176
ACTFRAME, 295
ADISPOSA, 289
Adresování, 219
Adresy OEM, 287
Alarmy
 nastavení v NC programu, 682
Alarmy cyklů, 682
ALF
 pro rychlé pozvednutí od kontury, 135
AND, 78
APR, 45
APRB, 45
APRP, 45
APW, 45
APWB, 45
APWP, 45
Array, 52
AS, 212
ASIN, 76
ASPLINE, 252
A-Spline, 258
Asynchronní oscilační pohyby, 623
ATAN2, 76
ATOL, 556
Atributy polohy
 nepřímé programování, 70
Automatické rozdělení dráhy, 640
Automatický ukazatel vyhledávání, 542
AV, 599
AX, 649
AXCTSWE, 657
AXCTSWECD, 657
AXCTSWED, 657
AXIS, 29
AXNAME, 87
AXSTRING, 649
AXTOCHAN, 145
AXTOSPI, 649
- B**
- B_AND, 78
B_NOT, 78
B_OR, 78
B_XOR, 78
BAUTO, 252
BFRAME, 295
BLOCK, 200
Blok zastavení, 542
BLSYNC, 130
BNAT, 252
BOOL, 29
BOUND, 82
BRISK, 526
BRISKA, 526
BSPLINE, 252
B-Spline, 259
BTAN, 252
- C**
- CAC, 251
CACN, 251
CACP, 251
CALL, 199
CALLPATH, 203
CASE, 107
CDC, 251
Cesta do adresáře, 221

Cesta pro vyhledávání
 Programovatelná cesta pro vyhledávání, 203
 při volání podprogramu, 224
CFINE, 305
CIC, 251
CLEARARM, 123
CLRINT, 132
COARSE, 599
COARSEA, 289
COLLPAIR, 400
COMPCAD, 265
COMPCURV, 265
COMPLETE, 227
COMPOF, 265
COMPON, 265
COMPSURF, 265
CONTDCON, 706
CONTPRON, 700
CORRTRAFO, 407
COS, 76
COUPDEF, 599
COUPDEL, 599
COUPOF, 599
COUPOFS, 599
COUPON, 599
COUPONC, 599
COUPRES, 599
CP, 380
CPBC, 611
CPDEF, 610
CPDEL, 610
CPFMOF, 613
CPFMON, 613
CPFMSON, 612
CPFPOS + CPOF, 613
CPFPOS + CPON, 611
CPFRS, 611
CPLA, 610
CPLCTID, 611
CPLDEF, 610
CPLDEL, 610
CPLDEN, 611
CPLINSC, 615
CPLINTR, 615
CPLNUM, 611
CPLOF, 611
CPLON, 610
CPLOUTSC, 615
CPLOUTTR, 615
CPLPOS, 611
CPLSETVAL, 611
CPMALARM, 616
CPMBRAKE, 616
CPMPRT, 615
CPMRESET, 614
CPMSTART, 615
CPMVDI, 616
CPOF, 610
CPON, 610
CPRECOF, 533
CPRECON, 533
CPROT, 235
CPROTDEF, 231
CPSETTYPE, 616
CPSYNCOF, 615
CPSYNCOF2, 615
CPSYNCOV, 616
CPSYNFIP, 615
CPSYNFIP2, 615
CPSYNFIV, 616
CSPLINE, 252
C-Spline, 260
CTAB, 579
CTABDEF, 568
CTABDEL, 575
CTABEND, 568
CTABEXISTS, 575
CTABFNO, 584
CTABFPOL, 584
CTABFSEG, 584
CTABID, 578
CTABINV, 579
CTABISLOCK, 578
CTABLOCK, 576
CTABMEMTYP, 578
CTABMPOL, 584
CTABMSEG, 584
CTABNO, 584
CTABNOMEM, 584
CTABPERIOD, 578
CTABPOL, 584
CTABPOLID, 584
CTABSEG, 584
CTABSEGID, 584
CTABSEV, 579
CTABSSV, 579
CTABTEP, 579
CTABTEV, 579
CTABTMAX, 579
CTABTMIN, 579
CTABTSP, 579
CTABTSV, 579
CTABUNLOCK, 576
CTOL, 556

- CTTRANS, 305
- CUT3DC, 434
- CUT3DCC, 444
- CUT3DCCD, 444
- CUT3DCD, 434
- CUT3DF, 438
- CUT3DFD, 438
- CUT3DFF, 438
- CUT3DFS, 438
- CUTMOD, 479
- CUTMODK, 479
- CYCLE4071
 - externí programování, 822
- CYCLE4072
 - externí programování, 823
- CYCLE4073
 - externí programování, 827
- CYCLE4074
 - externí programování, 828
- CYCLE4075
 - externí programování, 831
- CYCLE4077
 - externí programování, 834
- CYCLE4078
 - externí programování, 837
- CYCLE4079
 - externí programování, 839
- CYCLE435 - Nastavení souřadného systému
orovnávače
 - externí programování, 783
- CYCLE495 - Profilování
 - externí programování, 783
- CYCLE60 - Gravírování
 - externí programování, 735
- CYCLE61 - Rovinné frézování
 - externí programování, 737
- CYCLE62 - Volání kontury
 - externí programování, 739
- CYCLE63 - Frézování konturové kapsy
 - externí programování, 740
- CYCLE64 - Předvrtání konturové kapsy
 - externí programování, 742
- CYCLE70 - Frézování závitů
 - externí programování, 744
- CYCLE72 - Frézování po dráze
 - externí programování, 745
- CYCLE76 - Pravoúhlý čep
 - externí programování, 750
- CYCLE77 - Kruhový čep
 - externí programování, 752
- CYCLE78 - Frézování vrtaných závitů
 - externí programování, 754
- CYCLE79 - Mnohohran
 - externí programování, 757
- CYCLE800 - Naklápění
 - externí programování, 785
- CYCLE801 - Polohovací vzor Mřížka/Obdélník
 - externí programování, 789
- CYCLE802 - Libovolné pozice
 - externí programování, 790
- CYCLE81 - Navrtávání středících důlků
 - externí programování, 759
- CYCLE82 - Vrtání
 - externí programování, 760
- CYCLE83 - Vrtání hlubokých děr
 - externí programování, 762
- CYCLE830 - Vrtání hlubokých děr 2
 - externí programování, 793
- CYCLE832 - Vysokorychlostní obrábění
 - externí programování, 799
- CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky
 - externí programování, 765
- CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou
 - externí programování, 802
- CYCLE85 - Vystružování
 - externí programování, 768
- CYCLE86 - Vyvrtávání
 - externí programování, 769
- CYCLE899 - Frézování otevřené drážky
 - externí programování, 805
- CYCLE92 - Upichování
 - externí programování, 771
- CYCLE930 - Zápich
 - externí programování, 808
- CYCLE940 - Odlehčovací zápich
 - externí programování, 811
- CYCLE95 - Oddělování třísky od kontury
 - externí programování, 772
- CYCLE951 - Oddělování třísky
 - externí programování, 813
- CYCLE952 - Zápichy na kontuře
 - externí programování, 816
- CYCLE98 - Řetězec závitů
 - externí programování, 774
- CYCLE99 - Soustružení závitů
 - externí programování, 778
- Čelní frézování, 345
- Číslo břitů, 455
- CHAN, 29
- CHANDATA, 227
- CHAR, 29
- CHKDNO, 455
- Chráněné oblasti, 231

D

Datová třída, 50
 DCI, 50
 DCM, 50
 DCU, 50
 D-čísla
 přejmenovat, 456
 zkontrolovat, 455
 D-číslo
 volné zadávání, 455
 DEF, 29
 DEFAULT, 107
 DEFINE ... AS, 212
 Definice surového obrobku, 692
 DELAYFSTOF, 538
 DELAYFSTON, 538
 DELDL, 420
 DELETE, 152
 DELOBJ, 393
 DELTOOLENV, 490
 DISABLE, 132
 DISPLOF, 176
 DISPLON, 176
 DISPR, 543
 DIV, 76
 DL, 418
 DO, 621
 Doba zpracování, 672
 -chování řídicích struktur, 115
 DRIVE, 526
 DRIVEA, 526
 Druh vazby, 602
 Druhy transformací
 Všeobecná funkce, 325
 Držák nástroje
 - Orientovatelný, 463
 -kinematika, 458
 DV, 599
 DYNFINISH, 530
 DYNNORM, 530
 DYNPOS, 530
 DYNROUGH, 530
 DYNSEMIFIN, 530

E

Easy XML, 667
 EAUTO, 252
 EES, 217

EG

Elektronická převodovka, 592
 EGDEF, 592
 EGDEL, 597
 EGOFC, 596
 EGOFS, 596
 EGON, 593
 EGONSYN, 593
 EGONSYNE, 593
 Elektronická převodovka, 592
 ELSE, 116
 ENABLE, 132
 ENAT, 252
 ENDFOR, 118
 ENDIF, 116
 ENDLABEL, 109
 ENDLOOP, 118
 ENDWHILE, 120
 ESR, 684
 ESRR, 690
 ESRS, 690
 ETAN, 252
 Eulerův úhel, 342
 EVERY, 621
 EXECSTRING, 73
 EXECTAB, 712
 EXECUTE, 715
 EXP, 76
 EXTCALL
 pro SINUMERIK 828D, 208
 pro SINUMERIK 840D sl, 204
 EXTCLOSE, 677
 EXTERN, 194
 Externí posunutí počátku, 307
 externí programování, 841
 EXTOPEN, 677

F

FALSE, 29
 FCTDEF, 430
 FCUB, 521
 FENDNORM, 288
 FFWOF, 532
 FFWON, 532
 FIFOCTRL, 535
 FILEDATE, 156
 FILEINFO, 156
 FILESIZE, 156
 FILESTAT, 156
 FILETIME, 156
 FINE, 599

FINEA, 289
 FLIN, 521
 FNORM, 521
 FOR, 118
 FPO, 521
 FRAME, 29
 - zřetězení,
 Vyvolat,
 Framy
 Globální v NCU, 317
 Kanálové, 318
 Přiřadit, 303
 Řetězce framů, 304
 Systémové, 318
 Frézování konturové kapsy - CYCLE63
 externí programování, 740
 Frézování otevřené drážky - CYCLE899
 externí programování, 805
 Frézování po dráze - CYCLE72
 externí programování, 745
 Frézování vrtaných závitů - CYCLE78
 externí programování, 754
 Frézování závitu - CYCLE70
 externí programování, 744
 FROM, 621
 FTOCOF, 433
 FTOCON, 433
 Funkce OEM, 287

G

G290, 696
 G291, 696
 G5, 376
 G62, 288
 G621, 288
 G7, 376
 G810 ... G819, 287
 G820 ... G829, 287
 GEOAX, 652
 Geometrická osa
 přepínání, 652
 GET, 140
 GETACTTD, 457
 GETD, 140
 GETDNO, 456
 GETTCOR, 492
 GETTENV, 491
 GETVARAP, 61
 GETVARDFT, 64
 GETVARDIM, 63
 GETVARLIM, 62

GETVARPHU, 60
 GETVARTYP, 65
 G-kód
 nepřímé programování, 69
 Globální paměť výrobních programů (GDIR), 217
 GOTO, 104
 GOTOB, 104
 GOTOC, 104
 GOTOF, 104
 GOTOS, 103
 GP, 70
 Gravírování - CYCLE60
 externí programování, 735
 GROUP_ADDEND - konec doplňkového bloku
 najíždění
 externí programování, 842
 GROUP_BEGIN - Začátek programového bloku
 externí programování, 841
 GROUP_END - konec programového bloku
 externí programování, 842
 G-skupina
 Technologie, 530
 GUD, 30

H

Hloubka vnoření
 řídících struktur, 115
 Hloubka zajiždění nástroje, 437
 Hodnota opotřebení, 419
 HOLES1 - Polohovací vzor Přímka
 externí programování, 720
 HOLES2 - Polohovací vzor Kružnice
 externí programování, 720

I

ID, 621
 IDS, 621
 IF, 116
 IFRAME, 295
 INDEX, 91
 Index pole, 55
 INICF, 29
 Inicializace
 polí, 52
 Inicializační program, 227
 INIPO, 29
 INIRE, 29
 INIT, 123
 INITIAL, 227

INITIAL_INI, 227
 INT, 29
 Interpolace vektoru otočení, 358
 INTERSEC, 710
 IPOBRKA, 289
 IPOENDA, 289
 IPOSTOP, 599
 IPTRLOCK, 541
 IPTRUNLOCK, 541
 ISAXIS, 649
 ISFILE, 155
 ISNUMBER, 87
 ISOCALL, 202
 ISVAR, 59

J

JERKLIM, 552
 JERKLIMA, 528

K

Kinematika
 S více stupni volnosti, 462
 Koeficienty polynomu, 267
 Konec doplňkového bloku najíždění -
 GROUP_ADDEND
 externí programování, 842
 Konec programového bloku - GROUP_END
 externí programování, 842
 Kontura
 -kódování, 706
 opětovné najetí, 543
 -příprava, 700
 -tabulka, 700
 Konturový prvek
 odjíždění, 712
 Korekce rádiusu nástroje
 Zpoždění v rozích, 288
 Korekční parametry nástroje
 aditivní, 418
 Paměť korekcí, 415
 Souřadný systém pro hodnoty opotřebení, 425
 Kritérium konce pohybu
 programovatelné, 289
 Kruhová drážka - SLOT2
 externí programování, 730
 Kruhová kapsa - POCKET4
 externí programování, 725
 Kruhové osy
 Směrové vektory, 458

Úhel natočení, 458
 Vektory vzdálenosti, 458
 Kruhový čep - CYCLE77
 externí programování, 752

L

L..., 192
 Label, 109
 LEAD, 339
 LEADOF, 586
 LEADON, 586
 LENTOAX, 512
 Libovolné pozice - CYCLE802
 externí programování, 790
 LIFTFAST, 133
 Linkové
 - proměnné, 27
 Lisování, 635
 LLI, 42
 LN, 76
 Logické operátory, 78
 LONGHOLE - Podlouhlá díra
 externí programování, 733
 LOOP, 118
 LUD, 30

M

M17, 180
 M30, 180
 Makro, 212
 MASLDEF, 617
 MASLDEL, 617
 MASLOF, 617
 MASLOFS, 617
 MASLON, 617
 MATCH, 91
 MAXVAL, 82
 MCALL, 197
 MD10010, 123
 MD10280, 123
 MD15800, 26
 MD18104, 489
 MD18116, 490
 MD18156, 26
 MD20360, 497
 MD24558, 498
 MD24658, 498
 MEAC, 277
 MEAFRAME, 313

MEAS, 274
 MEASA, 277
 MEAW, 274
 MEAWA, 277
 MINDEX, 91
 MINVAL, 82
 MMC, 667
 Mnohohran - CYCLE79
 externí programování, 757
 MOD, 76
 MODAXVAL, 649
 MPF, 216

N

Naklápění - CYCLE800
 externí programování, 785
 NAMETOINT, 396
 Nastavení souřadného systému orovnávače -
 CYCLE435
 externí programování, 783
 Nástroj
 - orientace, 449
 -orientace při změně framu, 465
 -parametry, 415
 Návěští skoku
 při opakování úseku programu, 109
 při programových skocích, 104
 Navrtávání středících důlků - CYCLE81
 externí programování, 759
 Název cesty, 220
 Název jednotky, 220
 Název souboru, 222
 NCK, 29
 Nekonečná smyčka, 118
 Nepřímé programování
 adres, 67
 atributů polohy, 70
 G-kódů, 69
 řádků výrobního programu, 73
 NEWCONF, 147
 NOC, 599
 NOT, 78
 NPROT, 235
 NPROTDEF, 231
 NUMBER, 87
 NUT, 350

O

Obráběcí bod frézy, 437

Obrobek
 -adresáře, 216
 -hlavní adresář, 216
 -počítadlo, 675
 Oddělování třísky
 podporované funkce, 699
 Oddělování třísky - CYCLE951
 externí programování, 813
 Oddělování třísky od kontury - CYCLE95
 externí programování, 772
 Odjíždění
 nezávislé na pohonu, 690
 řízené NC systémem, 685
 Odlehčovací zápich - CYCLE940
 externí programování, 811
 OEMIPO1/2, 287
 Okrajové podmínky u transformací, 391
 OMA1 ... OMA5, 287
 On-line korekce délky nástroje, 466
 Opakování částí programu
 s nepřímým programováním CALL, 200
 OR, 78
 ORIAXES, 348
 ORIC, 449
 ORICONCCW, 350
 ORICONCW, 350
 ORICONIO, 350
 ORICONTO, 350
 ORICURVE, 353
 ORID, 449
 Orientace nástroje
 vzhledem ke dráze, 361
 Orientace nástroje vztahující se ke dráze, 361
 Orientační osy, 348
 Orientovatelný držák nástroje, 458
 ORIEULER, 348
 ORIMKS, 346
 ORIPATH, 362
 ORIPATHS, 362
 ORIPLANE, 350
 ORIRESET(A, B, C), 337
 ORIROTA, 358
 ORIROTC
 při interpolaci otáčení nástroje, 364
 při otáčení orientace nástroje, 358
 ORIROTR, 358
 ORIROTT, 358
 ORIRPY, 348
 ORIRPY2, 348
 ORIS, 449
 ORISOF, 370
 ORISOLH, 469

ORISON, 370
 ORIVECT, 348
 ORIVIRT1, 348
 ORIVIRT2, 348
 ORIWKS, 346
 OS, 623
 OSB, 623
 OSC, 449
 Oscilační pohyby
 Bod obratu, 630
 Oblast obratu, 630
 Přísuv v bodě obratu, 632
 Oscilační pohyby
 Asynchronní, 623
 Asynchronní oscilační pohyby, 623
 Dílčí přísuv, 630
 řízení prostřednictvím synchronních akcí, 628
 Synchronní oscilační pohyby, 628
 OSCILL, 628
 OSCTRL, 623
 OSD, 449
 OSE, 623
 OSNSC, 623
 OSOF, 449
 Osová vazba řídicí hodnotou, 586
 OSP1, 623
 OSP2, 623
 OSS, 449
 OSSE, 449
 OST, 449
 OST1, 623
 OST2, 623
 Osy
 *vyměnit, 140
 Vlečení, 565
 Otočení
 vektoru orientace, 358
 OTOL, 556

P

P..., 196
 P_ACTFRAME, 321
 Paměť
 Pracovní-, 227
 Program, 215
 Předběžné zpracování, 535
 Paměť korekcí, 415
 Paměť programů
 Standardní adresáře, 216
 Typy souborů, 216

Parametr
 Nástroj, 415
 -předávání při volání podprogramu, 194
 Parametry
 Formální-, 163
 -předávání při volání podprogramu, 163
 Skutečné-, 163
 Parametry kruhu
 vypočítat, 713
 PCALL, 202
 PDELAYOF, 635
 PDELAYON, 635
 PFRAME, 295
 PHI
 Polynom orientace, 356
 při orientaci podél kuželové plášťové plochy, 350
 PHU, 43
 PL
 při polynomické interpolaci, 266
 při splinové interpolaci, 252
 PO, 266
 PO[PHI]
 Polynom orientace, 356
 při orientaci podél kuželové plášťové plochy, 350
 při otáčení orientace nástroje, 362
 PO[PSI]
 Polynom orientace, 356
 při orientaci podél kuželové plášťové plochy, 350
 při otáčení orientace nástroje, 362
 PO[THT]
 Polynom orientace, 356
 při otáčení orientace nástroje, 362
 PO[XH]
 Polynom orientace, 357
 při udání orientace pomocí dvou styčných bodů, 353
 PO[YH]
 Polynom orientace, 357
 při udání orientace pomocí dvou styčných bodů, 353
 PO[ZH]
 Polynom orientace, 357
 při udání orientace pomocí dvou styčných bodů, 353
 POCKET3 - Pravoúhlá kapsa
 externí programování, 722
 POCKET4 - Kruhová kapsa
 externí programování, 725
 Početní parametry
 Globální, 25
 Kanálové, 24

- Podélná drážka - SLOT1
 - externí programování, 728
- Podlouhlá díra - LONGHOLE
 - externí programování, 733
- Podprogram
 - návrat zpět, s nastavitelnými parametry (RET ...), 182
 - návrat zpět, s nastavitelnými parametry (RETB...), 188
 - název, 160
 - opakování, 196
 - Použití, 159
 - Programovatelná cesta pro vyhledávání, 203
 - volání bez předávání parametrů, 192
 - volání modální, 197
 - volání nepřímé, 199
 - volání s předáváním parametrů, 194
- Podprogramy podle normy DIN, 223
- Polární transformace, 329
- Pole
 - definice, 52
 - prvek, 52
- POLF
 - pro odjíždění řízené NC systémem, 685
- POLFA, 685
- POLFMASK
 - pro odjíždění řízené NC systémem, 685
- POLFMLIN
 - pro odjíždění řízené NC systémem, 685
- Polohovací vzor Kružnice - HOLES2
 - externí programování, 720
- Polohovací vzor Mřížka/Obdélník - CYCLE801
 - externí programování, 789
- Polohovací vzor Přímka - HOLES1
 - externí programování, 720
- POLY, 266
- Polynom jmenovatele, 270
- Polynomická interpolace, 266
- POLYPATH, 266
- Pomocný bod frézy, 437
- PON, 643
- PONS, 635
- POSFS, 599
- Posunutí počátku
 - Externí, 307
- Posuv od bodu k bodu, 380
- Posuv PTP v kartézských souřadnicích, 380
- POT, 76
- Pracovní paměť, 227
- Pravoúhlá kapsa - POCKET3
 - externí programování, 722
- Pravoúhlý čep - CYCLE76
 - externí programování, 750
- PREPRO, 179
- PRESETON, 309
- PRESETONS, 311
- PRIO, 130
- PRLOC, 29
- Proces DataShare, 677
- Profilování - CYCLE495
 - externí programování, 783
- Program
 - adresování, 219
 - doba zpracování, 672
 - Inicializační-, 227
 - opakování, 196
 - paměť, 216
 - skoky, 104
 - větvení, 107
- Programová smyčka
 - Konečná smyčka, 118
 - Smyčka IF, 116
 - Smyčka REPEAT, 121
 - Smyčka s počítadlem, 118
 - Smyčka WHILE, 120
- Programování orientace, 348
- Proměnná
 - Konvertování typu, 85
- Proměnná typu FRAME
 - Předem definované proměnné typu FRAME, 295
 - Přiřazení hodnot, 299
 - Vyvolání transformace souřadného systému, 293
- Proměnné
 - Konverze typu, 86
 - uživatelem definovaná, 29
- Prostřihování, 635
- PROTA, 401
- PROTD, 403
- PROTS, 402
- Provozní režim
 - při měření, 282
- Předběžné zpracování
 - paměť, 535
- Předvrtání konturové kapsy - CYCLE64
 - externí programování, 742
- Přepínatelné geometrické osy, 652
- Přesnost kontury
 - programovatelná, 533
- Převodový faktor, 563
- Příprava kontury
 - Chybové hlášení, 715

PSI
 Polynom orientace, 356
 při orientaci podél kuželové pláštěvé plochy, 350
 PTP, 380
 PTPG0, 380
 PTPWOC, 380
 PUD, 30
 PUNCHACC, 635
 PUTFTOC, 432
 PUTFTOCF, 431
 PW, 252

R

READ, 153
 REAL, 29
 REDEF, 35
 Relační operátory, 78
 RELEASE, 140
 REP, 52
 REPEAT, 109
 REPEATB, 109
 REPOSA, 543
 REPOSH, 543
 REPOSHA, 543
 REPOSL, 543
 REPOSQ, 543
 REPOSQA, 543
 RET, 181
 RET (s nastavitelnými parametry), 182
 RETB (s nastavitelnými parametry), 188
 Režim jazyka, 696
 Režim při změnách zrychlení, 526
 RG, 25
 RINDEX, 91
 RMBBL, 543
 RMEBL, 543
 RMIBL, 543
 RMNBL, 543
 ROUND, 76
 ROUNDUP, 158
 Rovinné frézování - CYCLE61
 externí programování, 737
 Run MyScreens, 667
 Rutina přerušení
 Aktivovat / deaktivovat, 132
 Nové přiřazení, 131
 Programovatelný směr pohybu, 135
 Rychlé pozvednutí od kontury, 133
 Vymazat, 132
 Zpětný pohyb, 135
 Rychlé pozvednutí od kontury, 133

Řetězec nástroje, 412
 Řetězec obrobku, 412
 Řetězec závitů - CYCLE98
 externí programování, 774
 Řídící
 -struktury, 115
 Řídící osa
 v případě axiální vazby řídicí hodnotou, 586

S

SAVE, 170
 SBLOF, 171
 SBLON, 171
 SCPARA, 661
 SD, 252
 SD41610, 414
 SD41611, 414
 SD42475, 368
 SD42476, 368
 SD42477, 368
 SD42900, 424
 SD42910, 424
 SD42920, 424
 SD42930, 425
 SD42935, 427
 SD42940, 428
 SD42984, 481
 Seřizovací hodnota, 419
 SET, 52
 SETAL, 682
 SETDNO, 456
 SETINT, 130
 SETM, 123
 SETTTCOR, 499
 Simulace řídicí hodnoty, 590
 SIN, 76
 Singulární polohy, 347
 skok
 na návěští skoku, 104
 na začátek programu, 103
 SLOT1 - Podélná drážka
 externí programování, 728
 SLOT2 - Kruhová drážka
 externí programování, 730
 Složka framu
 FI, 301
 MI, 301
 RT, 301
 SC, 301
 TR, 301
 Směrový vektor, 342

Smyčka s počítadlem, 118
 Snížení rychlosti na všech rozích, 288
 SOFT, 526
 SOFTA, 526
 SON, 635
 SONS, 635
 Soubor
 -informace, 156
 Soustružení závitu - CYCLE99
 externí programování, 778
 SPF, 216
 SPI, 649
 SPIF1, 635
 SPIF2, 635
 Spline
 -interpolace, 252
 -Typy, 258
 SPLINEPATH, 263
 Splinový svazek, 263
 SPN, 640
 SPOF, 635
 Spouštěcí událost
 při měření, 281
 SPP, 640
 SPRINT, 94
 SQRT, 76
 START, 123
 STARTFIFO, 535
 STAT, 381
 Stav měřicí úlohy, 285
 Stav vazby
 při vlečení, 566
 v případě axiální vazby řídicí hodnotou, 590
 STOPFIFO, 535
 STOPRE, 535
 STRING, 29
 -délka, 90
 formátovat, 94
 -operace, 86
 -Zřetězení, 88
 STRINGIS, 663
 STRLEN, 90
 SUBSTR, 92
 Synchronizace
 hrubý, 602
 jemný, 602
 Synchronizace otáček, 599
 Synchronizace polohy, 599
 Synchronizace polohy s úhlovým offsetem, 599
 Synchronní oscilační pohyby
 Definice přiřazení, 631
 Následující dílčí přísuv, 634

Přiřazení oscilační a přísuvné osy, 631
 Přísuv v oblasti obratu, 632
 Přísuvný pohyb, 632
 Synchronní akce, 631
 Vyhodnocování, takt IPO, 633
 Synchronní vřeteno
 -definice páru, 605
 Vazba, 599
 SYN, 29
 SYN, 29
 SYN, 29
 Systém
 - Využitelnost závisující na, 5
 Systémové framy, 318
 Systémové proměnné
 Omezení měřicí sondy, 285
 Stav měřicí sondy, 285
 Šikmé zapichovací broušení, 376
 Špička frézy, 437

T

TAN, 76
 TANG, 515
 TANGDEL, 519
 TANGOF, 519
 TANGON, 518
 TCARR, 463
 TCOABS, 463
 TCOFR, 463
 TCOFRX, 463
 TCOFRY, 463
 TCOFRZ, 463
 THETA
 při interpolaci otáčení nástroje, 364
 při otáčení orientace nástroje, 358
 TILT, 339
 TLIFT, 517
 TMOF, 647
 TMON, 647
 TOFFOF, 466
 TOFFON, 466
 TOLOWER, 89
 TOOLENV, 487
 TOUPPER, 89
 TOWBCS, 425
 TOWKCS, 425
 TOWMCS, 425
 TOWSTD, 425
 TOWTCS, 425
 TOWWCS, 425

- TRAANG
s naprogramovatelným úhlem, 375
- TRACON, 378
- TRACYL, 372
- TRAFOOF, 392
- TRAFOON, 405
- TRAILOF, 563
- TRAILON, 563
- Transformace
Kinematické transformace, 326
Transformace orientace, 325
Tří-, čtyř- a pětiosá transformace, 336
Základní nastavení orientace nástroje nezávislé na kinematice, 326
zřetězené, 378
Zřetězené transformace, 327
- Transformace orientace TRAORI
Generická 5/6-osá transformace, 328
Kinematika stroje, 328
Pohyby pro nastavení orientace, 327
Programování orientace, 337
Varianty programování orientace, 337
- Transformace s naklápěcí lineární osou, 335
- Transformace šikmého úhlu (TRAANG)
s naprogramovatelným úhlem, 375
- Transformace válcového pláště, 329
- TRANSMIT, 372
- TRAORI, 336
- Trhové pohyby
-korekce, 552
-omezení, 526
- TRUE, 29
- TRUNC, 76
- TU, 385
- Typ kinematiky, 462
- U**
- Úhel RPY, 342
- ULI, 42
- UNTIL, 121
- Upichování - CYCLE92
externí programování, 771
- Úsek programu
-opakování, 109
- Uživatelské XML, 667
- V**
- Vazba
Generická, 610
- Vazba pomocí požadované hodnoty, 602
- Vazba pomocí rychlosti, 602
- Vazba pomocí skutečné hodnoty, 602
- Vazba řídicí hodnotou
Synchronizace řídicí a vlečné osy., 588
Vazba pomocí skutečné a požadované hodnoty, 589
- Vektor orientace THETA, 358
- velká a malá písmena nejsou rozlišována, 218
- VELOLIM, 553
- VELOLIMA, 528
- Vlečené osy, 565
- Vlečení, 563
- Vlečná osa
v případě axiální vazby řídicí hodnotou, 586
- Vlečné spojení os, 563
- Volání kontury - CYCLE62
externí programování, 739
- Volání podprogramu s udáním cesty a s parametry, 202
- Vrtání - CYCLE82
externí programování, 760
- Vrtání hlubokých děr - CYCLE83
externí programování, 762
- Vrtání hlubokých děr 2 - CYCLE830
externí programování, 793
- Vrtání závitů bez vyrovnávací hlavičky - CYCLE84
externí programování, 765
- Vrtání závitů s vyrovnávací hlavičkou - CYCLE840
externí programování, 802
- Vřetenno
*vyměnit, 140
- Vyhazení
průběh orientace, 370
- Vymazání zbytkové dráhy, 283
- Vypisování bloku
potlačení, 176
- Výsledek
do externího zařízení/souboru, 677
- Vysokorychlostní obrábění - CYCLE832
externí programování, 799
- Vystružování - CYCLE85
externí programování, 768
- Využitelnost
V závislosti na systému, 5
- Vyvrtávání - CYCLE86
externí programování, 769
- W**
- WAITC, 599
- WAITE, 123

WAITENC, 659
WAITM, 123
WAITMC, 123
WHEN, 621
WHEN-DO, 631
WHENEVER, 621
WHENEVER-DO, 631
WHILE, 120
WORKPIECE, 692
WRITE, 148

X

XOR, 78

Z

Začátek programového bloku - GROUP_BEGIN, 841
Základní nastavení orientace nástroje -
ORIRESET, 338
Zaokrouhlování, 158
Zápich - CYCLE930
 externí programování, 808
Zápichy na kontuře - CYCLE952
 externí programování, 816
zastavení
 nezávislé na pohonu, 690
 řízené NC systémem, 689
Zbývající čas
 pro daný obrobek, 674
Zjišťování a prohledávání oblastí určených pro
vyhledávání, 542
Znak 0, 86
Zpoždění na vnitřních rozích, 288
Zpracování blok po bloku
 -potlačení, 171
Způsob zápisu typu EES, 219
Způsob zápisu typu NCK, 219
Zřetězení obráběcích operací
 proměnných typu String, 88