SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D
Przygotowanie do pracy

Podręcznik programowania

Obowiązuje dla

Sterowanie
SINUMERIK 840D sl / 840DE sl / 828D

Wersja oprogramowania CNC 4.8 SP3

08/2018
6FC5398-2BP40-6NA2

Słowo wstępne
Podstawowe wskazówki bezpieczeństwa 1
Elastyczne programowanie NC 2
Zarządzanie plikami i programami 3
Obszary ochrony 4
Specjalne polecenia wykonania ruchu 5
Transformacje współrzędnych (Frame) 6
Transformacje 7
Łańcuchy kinematyczne 8
Uniknięcie kolizji za pomocą łańcuchów kinematycznych 9
Transformacje za pomocą łańcuchów kinematycznych 10
Korekcje narzędzi 11
Zachowanie się w ruchu po torze 12
Sprzężenia osi 13
Akcje synchroniczne 14
Ruch wahliwy 15
Tłoczenie i cięcie 16
Szlifowanie 17
Dalsze funkcje 18
Własne programy obróbki 19
Zewnętrzne programowanie cykli 20
Tablice 21
Aneks A

Промэлектроника, sales@prom-elec.com

Wskazówki prawne

Koncepcja wskazówek ostrzeżeń

Podręcznik zawiera wskazówki, które należy bezwzględnie przestrzegać dla zachowania bezpieczeństwa oraz w celu uniknięcia szkód materialnych. Wskazówki dot. bezpieczeństwa oznaczono trójkątnym symbolem, ostrzeżenia o możliwości wystąpienia szkód materialnych nie posiadają trójkątnego symbolu ostrzegawczego. W zależności od opisywanego stopnia zagrożenia, wskazówki ostrzegawcze podzielono w następujący sposób.

⚠ NIEBEZPIECZEŃSTWO
oznacza, że nieprzestrzeganie tego typu wskazówek ostrzegawczych **grozi** śmiercią lub odniesieniem ciężkich obrażeń ciała.

⚠ OSTRZEŻENIE
oznacza, że nieprzestrzeganie tego typu wskazówek ostrzegawczych **może** grozić śmiercią lub odniesieniem ciężkich obrażeń ciała.

⚠ OSTROŻNIE
oznacza, że nieprzestrzeganie tego typu wskazówek ostrzegawczych może spowodować lekkie obrażenia ciała.

UWAGA
oznacza, że nieprzestrzeganie tego typu wskazówek ostrzegawczych może spowodować szkody materialne.

W wypadku możliwości wystąpienia kilku stopni zagrożenia, wskazówkę ostrzegawczą oznaczono symbolem najwyższego z możliwych stopnia zagrożenia. Wskazówka oznaczona symbolem ostrzegawczym w postaci trójkąta, informująca o istniejącym zagrożeniu dla osób, może być również wykorzystana do ostrzeżenia przed możliwością wystąpienia szkód materialnych.

Wykwalifikowany personel

Produkt /system przynależny do niniejszej dokumentacji może być obsługiwany wyłącznie przez **personel wykwalifikowany** do wykonywania danych zadań z uwzględnieniem stosownej dokumentacji, a zwłaszcza zawartych w niej wskazówek dotyczących bezpieczeństwa i ostrzegawczych. Z uwagi na swoje wykształcenie i doświadczenie wykwalifikowany personel potrafi podczas pracy z tymi produktami / systemami rozpoznać ryzyka i unikać możliwych zagrożeń.

Zgodne z przeznaczeniem używanie produktów firmy Siemens

Przestrzegać następujących wskazówek:

⚠ OSTRZEŻENIE
Produkty firmy Siemens mogą być stosowane wyłącznie w celach, które zostały opisane w katalogu oraz w załączonej dokumentacji technicznej. Polecenie lub zalecenie firmy Siemens jest warunkiem użycia produktów bądź komponentów innych producentów. Warunkiem niezawodnego i bezpiecznego działania tych produktów są prawidłowe transport, przechowywanie, ustawienie, montaż, instalacja, uruchomienie, obsługa i konserwacja. Należy przestrzegać dopuszczalnych warunków otoczenia. Należy przestrzegać wskazówek zawartych w przynależnej dokumentacji.

Znaki towarowe

Wszystkie produkty oznaczone symbolem ® są zarejestrowanymi znakami towarowymi firmy Siemens AG. Pozostałe produkty posiadające również ten symbol mogą być znakami towarowymi, których wykorzystywanie przez osoby trzecie dla własnych celów może naruszać prawa autorskie właściciela danego znaku towarowego.

Wykluczenie od odpowiedzialności

Treść drukowanej dokumentacji została sprawdzona pod kątem zgodności z opisywanym w niej sprzętem i oprogramowaniem. Nie można jednak wykluczyć pewnych rozbieżności i dlatego producent nie jest w stanie zagwarantować całkowitej zgodności. Informacje i dane w niniejszej dokumentacji poddawane są ciągłej kontroli. Poprawki i aktualizacje ukazują się zawsze w kolejnych wydaniach.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
NIEMCY

Numer zamówieniowy dokumentacji: 6FC5398-2BP40-6NA2
℗ 08/2018 Prawo do dokonywania zmian zastrzeżone

Copyright © Siemens AG 1995 - 2018.
Wszelkie prawa zastrzeżone

Słowo wstępne

Dokumentacja SINUMERIK

Dokumentacja SINUMERIK jest podzielona na następujące kategorie:

- Dokumentacja ogólna/katalogi
- Dokumentacja użytkownika
- Dokumentacja producenta/serwisowa

Dalsze informacje

Pod adresem (https://support.industry.siemens.com/cs/de/en/view/108464614) znajdują się informacje na temat:

- Zamawiania dokumentacji/Wykaz publikacji
- Dodatkowe linki do pobierania dokumentacji
- Korzystania z dokumentacji online (szukanie i przeglądanie podręczników/informacji)

W przypadku pytań do dokumentacji technicznej (np. propozycje, korekty) należy wysłać e-mail na następujący adres (mailto:docu.motioncontrol@siemens.com).

mySupport/Dokumentation

Pod adresem (https://support.industry.siemens.com/My/ww/en/documentation) znajdują się informacje, w jaki sposób można indywidualnie zestawiać dokumentacje na bazie zasobów Siemensa i dopasowywać je do własnej dokumentacji maszyny.

Szkolenie

Pod adresem (http://www.siemens.com/sitrain) znajdują się informacje centrum szkoleniowego SITRAIN - szkolenia w zakresie produktów, systemów i rozwiązań techniki automatyzacyjnej Siemens.

FAQs

Najczęściej zadawane pytania można znaleźć na stronach Service&Support w zakładce Produkt Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

Informacje dot. SINUMERIK można znaleźć pod następującym adresem (http://www.siemens.com/sinumerik).

Grupa docelowa

Niniejszy podręcznik jest przeznaczony dla:

- programistów
- projektantów

Korzyści

Podręcznik programowania pomaga w projektowaniu, pisaniu, testowaniu programów i softwareowych interfejsów graficznych oraz usuwania błędów.

Zakres standardowy

W niniejszym podręczniku programowania opisano funkcje zakresu standardowego. Uzupełnienia albo zmiany, które zostały dokonane przez producenta maszyny, są przez niego dokumentowane.

W sterowaniu mogą być możliwe do realizacji dalsze funkcje, nie opisane w niniejszej dokumentacji. Nie można jednak zgłaszać roszczenia do tych funkcji w przypadku dostawy nowego urządzenia lub w przypadku serwisowania.

Ze względu na przejrzystość dokumentacja nie zawiera szczegółowych informacji dotyczących wszystkich typów produktu i nie uwzględnia każdego możliwego przypadku ustawienia, obsługi i konserwacji.

Wsparcie techniczne

Właściwe dla krajów numery telefonów doradztwa technicznego można znaleźć w internecie na stronie (https://support.industry.siemens.com/sc/ww/en/sc/2090) w zakładce „Kontakt”.

Informacje odnośnie struktury i treści

Podręcznik programowania "Podstawy” / "Przygotowanie do pracy"

Opisy do programowania NC są podzielone na dwa podręczniki:

1. **Podstawy**
   Podręcznik programowania "Podstawy" przeznaczony jest dla operatorów maszyn, którzy posiadają odpowiednią wiedzę w zakresie obróbki wiertłem, frezarskiej i tokarskiej. Na prostych przykładach programowania zostaną objaśnione polecenia i instrukcje znane również z DIN66025.
2. **Przygotowanie do pracy**
   Podręcznik programowania "Przygotowanie do pracy" przeznaczony jest dla technologów znających wszystkie możliwości programowania. Sterowanie SINUMERIK umożliwia przy pomocy specjalnego języka programowania sporządzenie skomplikowanego programu obróbki (np. powierzchnie o dowolnym kształcie, koordynacja kanałów, ...) i ułatwia technologom pracochłonne programowanie.

Dostępność opisanych elementów językowych NC

Wszystkie elementy językowe opisane w niniejszym podręczniku są dostępne dla SINUMERIK 840D sl. Dostępność dla SINUMERIK 828D przedstawiona jest w tabeli "Instrukcje: Dostępność w przypadku SINUMERIK 828D (Strona 897)„.

Spis treści

Słowo wstępne 3

1 Podstawowe wskazówki bezpieczeństwa 19

1.1 Ogólne wskazówki bezpieczeństwa 19

1.2 Gwarancja i odpowiedzialność za przykłady aplikacyjne 20

1.3 Industrial Security 21

2 Elastyczne programowanie NC 23

2.1 Zmienne 23
2.1.1 Dane systemowe 23
2.1.2 Wstępnie zdefiniowane zmienne użytkownika: Parametry obliczeniowe 26
2.1.2.1 Specyficzne dla kanału parametry obliczeniowe (R) 26
2.1.2.2 Globalne parametry obliczeniowe (RG) 27
2.1.3 Predefiniowane zmienne użytkownika: zmienne link 29
2.1.4 Definicja zmiennych użytkownika (DEF) 31
2.1.5 Redefinicja danych systemowych, danych użytkownika i poleceń językowych NC (REDEF) 37
2.1.6 Atrybut: wartość inicjalizacyjna 40
2.1.7 Atrybut: wartości graniczne (LLI, ULI) 43
2.1.8 Atrybut: jednostka fizyczna (PHU) 45
2.1.9 Atrybut: prawa dostępu (APR, APW, APRP, APWP, APRB, APWB) 47
2.1.10 Atrybut: Klasa danych (DCM, DCI, DCU) - tylko SINUMERIK 828D 53
2.1.11 Przegląd atrybutów definiowalnych i redefiniowalnych 54
2.1.12 Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP) 55
2.1.13 Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP): Dalsze informacje 60
2.1.14 Typy danych 61
2.1.15 Sprawdzenie istnienia zmiennej (ISVAR) 62
2.1.16 Wartości atrybutów/Odczyt typu danych (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP) 64

2.2 Programowanie pośrednie 70
2.2.1 Pośrednie programowanie adresów 70
2.2.2 Pośrednie programowanie poleceń G 72
2.2.3 Programowanie pośrednie atrybutów pozycji (GP) 73
2.2.4 Pośrednie programowanie wierszy programu obróbki (EXECSTRING) 76

2.3 Funkcje arytmetyczne 78

2.4 Operacje porównania i operacje logiczne 81

2.5 Korekcja dokładności w przypadku błędów porównania (TRUNC) 83

2.6 Minimum, maksimum i zakres zmiennych (MINVAL, MAXVAL i BOUND) 85

2.7 Priorytet operacji 87

2.8 Możliwe konwersje typu 88

2.9 Operacja na łańcuchach znaków 89

2.9.1 Konwersja typu na STRING (AXSTRING)....89
2.9.2 Konwersja typu z STRING (NUMBER, ISNUMBER, AXNAME)....90
2.9.3 Powiązanie łańcuchów znaków (<<)....91
2.9.4 Zmiana na litery małe/duże (TOLOWER, TOUPPER)....92
2.9.5 Określenie długości łańcucha znaków (STRLEN)....93
2.9.6 Szukanie znaku/łańcucha znaków w łańcuchu znaków (INDEX, RINDEX, MINDEX, MATCH)....94
2.9.7 Wybór częściowego łańcucha znaków (SUBSTR)....95
2.9.8 Odczyt i zapis pojedynczych znaków....96
2.9.9 Formatowanie łańcucha znaków (SPRINT)....97

2.10 Skoki i rozgałęzienia w programie....106
2.10.1 Skok do początku programu (GOTOS)....106
2.10.2 Skoki w programie do znaczników skoku (GOTOB, GOTOF, GOTO, GOTOC)....107
2.10.3 Rozgałęzienie programu (CASE ... OF ... DEFAULT ...)....110

2.11 Powtórzenie programu (REPEAT, REPEATB, ENDLABEL, P)....112

2.12 Struktury kontrolne....119
2.12.1 Warunkowa instrukcja i rozgałęzienie (IF, ELSE, ENDIF)....120
2.12.2 Pętla programowa bez końca (LOOP, ENDLOOP)....122
2.12.3 Pętla FOR (FOR ... TO ..., ENDFOR)....122
2.12.4 Pętla programowa z warunkiem na początku pętli (WHILE, ENDWHILE)....124
2.12.5 Pętla programowa z warunkiem na końcu (REPEAT, UNTIL)....125
2.12.6 Przykład programu z kaskadowanymi strukturami kontrolnymi....125

2.13 Polecenia koordynacji (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)....127

2.14 Procedura przerwania (ASUP)....132
2.14.1 Funkcja procedury przerwania....132
2.14.2 Sporządzenie procedury przerwania....133
2.14.3 Przyporządkowanie i uruchomienie procedury przerwania (SETINT, PRIO, BLSYNC). 134
2.14.4 Wyłączenie aktywności / reaktywowanie przyporządkowania procedury przerwania (DISABLE, ENABLE)....136
2.14.5 Skasowanie przyporządkowania procedury przerwania (CLRINT)....136
2.14.6 Szybkie cofnięcie od konturu (SETINT LIFTFAST, ALF)....137
2.14.7 Kierunek ruchu przy szybkim cofnięciu od konturu 139
2.14.8 Przebieg ruchów w przypadku procedur przerwania....142

2.15 Zamiana osi, zamiana wrzecion (RELEASE, GET, GETD)....144

2.16 Przekazanie osi do innego kanału (AXTOCHAN)....149

2.17 Ustawienie działania danych maszynowych (NEWCONF)....151

2.18 Zapisanie pliku (WRITE)....152

2.19 Skasowanie pliku (DELETE)....156

2.20 Odczyt wierszy w pliku (READ)....157

2.21 Sprawdzenie istnienia pliku (ISFILE)....159

2.22 Odczyt informacji o pliku (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)....160

2.23 Zaokrąglenie do góry (ROUNDUP)....162

2.24 Technika podprogramów....163
2.24.1 Informacje ogólnie....163
2.24.1.1 Podprogram....163

2.24.1.2 Nazwy podprogramów....164
2.24.1.3 Kaskadowanie podprogramów....165
2.24.1.4 Ścieżka szukania....166
2.24.1.5 Parametry formalne i aktualne....166
2.24.1.6 Przekazanie parametrów....167
2.24.2 Definicja podprogramu....169
2.24.2.1 Podprogram bez przekazania parametrów....169
2.24.2.2 Podprogram z przekazaniem parametrów Call-by-Value (PROC)....170
2.24.2.3 Podprogram z przekazaniem parametrów Call-by-Reference (PROC, VAR)....171
2.24.2.4 Zapisywanie modalnych funkcji G (SAVE)....173
2.24.2.5 Blokowanie wykonywania pojedynczymi blokami (SBLOF, SBLON)....174
2.24.2.6 Blokowanie wyświetlania aktualnego bloku (DISPLOF, DISPLON, ACTBLOCNO)....180
2.24.2.7 Oznakowanie podprogramów z przygotowaniem (PREPRO)....183
2.24.2.8 Powrót z podprogramu M17....184
2.24.2.9 Skok powrotny z podprogramu RET....185
2.24.2.10 Parametryzowalny powrót z podprogramu (RET ...)....186
2.24.2.11 Parametryzowany powrót z podprogramu (RETB ...)....192
2.24.3 Wywołanie podprogramu....196
2.24.3.1 Wywołanie podprogramu bez przekazania parametrów....196
2.24.3.2 Wywołanie podprogramu z przekazaniem parametrów (EXTERN)....198
2.24.3.3 Liczba powtórzeń programu (P)....200
2.24.3.4 Modalne wywołanie podprogramu (MCALL)....201
2.24.3.5 Pośrednie wywołanie podprogramu (CALL)....203
2.24.3.6 Pośrednie wywołanie podprogramu z podaniem części programu do wykonania (CALL BLOCK ... TO ...)....204
2.24.3.7 Pośrednie wywołanie programu programowanego w języku ISO (ISOCALL)....206
2.24.3.8 Wywołanie podprogramu z podaniem ścieżki i parametrami (PCALL)....206
2.24.3.9 Rozszerzenie ścieżki szukania przy wywołaniach podprogramu (CALLPATH)....207
2.24.3.10 Wykonywanie podprogramu zewnętrznego (840D sl) (EXTCALL)....208
2.24.3.11 Wykonywanie podprogramu zewnętrznego (828D) (EXTCALL)....212
2.25 Technika makr (DEFINE ... AS)....217

3 Zarządzanie plikami i programami....221

3.1 Pamięć programów....221
3.1.1 Pamięć programów w NC....221
3.1.2 Zewnętrzna pamięć programów....223
3.1.3 Adresowanie plików pamięci programów....225
3.1.4 Ścieżka szukania przy wywołaniu podprogramu....230
3.1.5 Odpytanie ścieżki i nazwy pliku....231
3.2 Pamięć robocza (CHANDATA, COMPLETE, INITIAL)....233

4 Obszary ochrony....237

4.1 Definiowanie obszarów ochrony (CPROTDEF, NPROTDEF)....237
4.2 Uaktywnienie / wyłączenie aktywności obszarów ochrony (CPROT, NPROT)....241
4.3 Sprawdzenie naruszenia obszaru ochrony, ograniczenia obszaru pracy oraz programowych wyłączników krańcowych (CALCPOSI)....245

5 Specjalne polecenia wykonania ruchu....257

5.1 Ruch do pozycji kodowanych (CAC, CIC, CDC, CACP, CACN)....257

5.2 Interpolacja Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)....258
5.3 Zespół spline (SPLINEPATH)....269
5.4 Włączenie/Wyłączenie kompresji bloków NC (COMPON / COMPCURV / COMPCAD / COMPSURF)....271
5.5 Interpolacja wielomianowa (POLY, POLYPATH, PO, PL)....272
5.6 Ustawiane odniesienie do toru (SPATH, UPATH)....278
5.7 Pomiar przy pomocy sondy przełączającej, MEAS, MEAW....280
5.8 Pomiary specyficzne dla osi (MEASA, MEAWA, MEAC) (opcja)....283
5.9 Funkcje specjalne dla użytkownika OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)....293
5.10 Zmniejszenie posuwu ze zwłoką na narożach (FENDNORM, G62, G621) 294
5.11 Programowane kryterium końca ruchu (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)....295

6 Transformacje współrzędnych (Frame)....299

6.1 Transformacja współrzędnych przez zmienną frame....299
6.1.1 Predefiniowana zmienna frame ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)....301
6.2 Przyporządkowania wartości do Frames....305
6.2.1 Bezpośrednie przyporządkowanie wartości (wartość osi, kąt, skala)....305
6.2.2 Odczyt i zmiana komponentów frame (TR, FI, RT, SC, MI)....307
6.2.3 Obliczanie z frame....308
6.2.4 Definicja zmiennych frame (DEF FRAME)....310
6.3 Przesunięcie zgrubne i dokładne (CTRANS, CFINE)....311
6.4 Zewnętrzne przesunięcie punktu zerowego ($AA_ETRANS)....313
6.5 Ustawienie wartości rzeczywistej z utratą statusu bazowania (PRESETON)....315
6.6 Ustawienie wartości rzeczywistej bez utraty statusu bazowania (PRESETONS)....317
6.7 Obliczenie frame z 3 punktów pomiarowych w przestrzeni (MEAFRAME)....319
6.8 Frame globalne dla NCU....323
6.8.1 Frame kanałowe ($P_CHBFR, $P_UBFR)....324
6.8.2 Frame działające w kanale....324

7 Transformacje....331

7.1 Ogólne programowanie rodzajów transformacji....331
7.1.1 Ruchy orientacji przy transformacjach....333
7.1.2 Przegląd transformacji orientacji TRAORI....337
7.2 Transformacje trzy-, cztero- i pięcioosiowe (TRAORI)....339
7.2.1 Zależności ogólne głowicy narzędziowej Kardana....339
7.2.2 Transformacje trzy-, cztero- i pięcioosiowe (TRAORI)....342
7.2.3 Warianty programowania orientacji i położenie podstawowe (OTIRESET)....343
7.2.4 Programowanie orientacji narzędzia (A..., B..., C..., LEAD, TILT)....345
7.2.5 Frezowanie czołowe (A4, B4, C4, A5, B5, C5)....351

7.2.6 Odniesienie osi orientacji (ORIWKS, ORIMKS)....352
7.2.7 Programowanie osi orientacji (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)....354
7.2.8 Programowanie orientacji wzdłuż pobocznicy stożka (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)....356
7.2.9 Zadanie orientacji dwóch punktów kontaktowych (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)....359
7.3 Wielomiany orientacji (PO[kąt], PO[współrzędna])....362
7.4 Obroty orientacji narzędzia (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)....364
7.5 Orientacje w stosunku do toru....367
7.5.1 Rodzaje orientacji w stosunku do toru....367
7.5.2 Odniesiony do toru obrót orientacji narzędzia (ORIPATH, ORIPATHS, kąt obrotu)....368
7.5.3 Interpolacja obrotu narzędzia względem toru (ORIROTC, THETA)....369
7.5.4 Wygładzanie przebiegu orientacji (ORIPATHS A8=, B8=, C8=)....372
7.6 Kompresja orientacji (COMPON, COMPCURV, COMPCAD, COMPSURF)....373
7.7 Wygładzanie przebiegu orientacji, załączenie/wyłączenie (ORISON, ORISOF)....376
7.8 Transformacja kinematyczna....378
7.8.1 Włączenie transformacji strony czołowej (TRANSMIT)....378
7.8.2 Włączenie transformacji pobocznicy walca (TRACYL)....378
7.8.3 Włączenie transformacji kąta skośnego z kątem programowalnym (TRAANG)....381
7.8.4 Wcinanie skośne na szlifierkach (G5, G7)....382
7.9 Włączenie transformacji powiązanej (TRACON)....384
7.10 Ruch kartezjański PTP....386
7.10.1 Włączenie/Wyłączenie ruchu kartezjańskiego PTP (PTP, PTPG0, PTPWOC, CP)....386
7.10.2 Podanie położenia przegubów (STAT)....387
7.10.3 Podanie znaku kąta osi (TU)....391
7.10.4 Przykład 1: Ruch PTP robota 6-osiowego z transformacją ROBX....394
7.10.5 Przykład 2: Ruch PTP przy rodzajowej transformacji 5-osiowej....395
7.10.6 Przykład 3: PTPG0 i TRANSMIT....396
7.11 Warunki brzegowe przy wyborze transformacji....398
7.12 Cofnięcie wyboru transformacji (TRAFOOF)....399
8 Łańcuchy kinematyczne....401
8.1 Kasowanie komponentów (DELOBJ)....401
8.2 Określanie indeksu poprzez nazwy (NAMETOINT)....404
9 Uniknięcie kolizji za pomocą łańcuchów kinematycznych....407
9.1 Sprawdzenie pod kątem pary kolizyjnej (COLLPAIR)....408
9.2 Żądanie ponowne obliczenie modelu maszyny dla unikania kolizji (PROTA):....409
9.3 Ustawienie stanu obszaru ochrony (PROTS)....410
9.4 Określenie odległości dwóch obszarów ochrony (PROTD)....411
10 Transformacje za pomocą łańcuchów kinematycznych....413
10.1 Włączenie transformacji (TRAFOON)....413
10.2 Modyfikacja transformacji orientacji po pomiarze maszyny (CORRTRAFO)....415

11 Korekcje narzędzi....423

11.1 Pamięć korekcji....423

11.2 Korekcje addytywne....426
11.2.1 Wybranie korekcji addytywnych (DL)....426
11.2.2 Ustalenie zużycia i wartości ustawczych ($TC_SCPxy[t,d], $TC_ECPxy[t,d])....427
11.2.3 Skasowanie korekcji addytywnych (DELDL)....428

11.3 Traktowanie specjalne korekcji narzędzia....430
11.3.1 Lustrzane odbicie długości narzędzia....432
11.3.2 Reakcja na znak zużycia....432
11.3.3 Układ współrzędnych aktywnej obróbki (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)....433
11.3.4 Długość narzędzia i zmiana płaszczyzny....436

11.4 Korekcja narzędzia online....438
11.4.1 Definicja funkcji wielomianowej (FCTDEF)....438
11.4.2 Korekcja narzędzia online zapis, nieprzerwany (PUTFTOCF)....439
11.4.3 Zapis korekcji narzędzia online, nieciągły (PUTFTOC)....440
11.4.4 Włączenie/wyłączenie korekcji narzędzia online (FTOCON/FTOCOF)....441

11.5 Korekcja promienia narzędzia 3D....442
11.5.1 Wybór korekcji promienia narzędzia 3D dla frezowania obwodowego (CUT3DC, CUT3DCD, ISD)....442
11.5.2 Wybór korekcji promienia narzędzia 3D dla frezowania czołowego (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)....446
11.5.3 Frezowanie obwodowe 3D z uwzględnieniem powierzchni ograniczającej (CUT3DCC, CUT3DCCD)....452

11.6 Orientacja narzędzia (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)....457

11.7 Dowolne nadawanie numerów D, numer ostrza....463
11.7.1 Dowolne nadawanie numerów D , numer ostrza (adres CE)....463
11.7.2 Dowolne nadawanie numerów D: sprawdzenie numerów D (CHKDNO)....463
11.7.3 Dowolne nadawanie numerów D: Zmiana nazw numerów D (GETDNO, SETDNO)....464
11.7.4 Dowolne nadawanie numerów D: Określenie numeru T do zadanego numeru D (GETACTTD)....465
11.7.5 Dowolne nadawanie numerów D: ustawienie nie obowiązywania numerów D (DZERO)....465

11.8 Kinematyka nośnika narzędzi....466

11.9 Korekcja długości narzędzia dla orientowalnych nośników narzędzi (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)....472

11.10 Korekcja długości narzędzia online (TOFFON, TOFFOF)....475

11.11 Modyfikacja danych korekcji dla obrotowych narzędzi....478
11.11.1 Obliczenie orientacji (ORISOLH)....478
11.11.2 Aktywacja modyfikacji danych korekcji dla narzędzi obrotowych (CUTMOD, CUTMODK)....487

11.12 Praca w środowiskach narzędziowych....494
11.12.1 Zapisanie środowiska narzędziowego (TOOLENV)....494
11.12.2 Kasowanie środowiska narzędziowego (DELTOOLENV)....497
11.12.3 Odczyt numerów T, D i DL (GETTENV)....498
11.12.4 Odczyt informacji do zapisanych środowisk narzędziowych ($P_TOOLENVN, ($P_TOOLENV)....499
11.12.5 Odczyt długości narzędzia lub składowych długości narzędzia (GETTCOR)....499

11.12.6 Zmiana składowych narzędzia (SETTCOR)....506
11.13 Odczyt przyporządkowania długości narzędzia L1, L2, L3 do osi współrzędnych (LENTOAX)....519
12 Zachowanie się w ruchu po torze....523
12.1 Sterowanie styczne....523
12.1.1 Definicja sprzężenia (TANG)....523
12.1.2 Włączenie tworzenia bloku pośredniego (TLIFT)....525
12.1.3 Włączenie sprzężenia (TANGON)....526
12.1.4 Wyłączenie sprzężenia (TANGOF)....527
12.1.5 Kasowanie sprzężenia (TANGDEL)....527
12.2 Przebieg posuwu (FNORM, FLIN, FCUB, FPO)....530
12.3 Przyśpieszenie....535
12.3.1 Tryb przyśpieszenia (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)....535
12.3.2 Sterowanie przyśpieszeniem w przypadku osi nadążnych (VELOLIMA, ACCLIMA, JERKLIMA)....537
12.3.3 Uaktywnienie specyficznych dla technologii wartości dynamiki (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)....539
12.4 Ruch ze sterowaniem wyprzedzającym (FFWON, FFWOF)....541
12.5 Programowalna dokładność konturu (CPRECON, CPRECOF)....542
12.6 Przebieg programu z pamięcią przebiegu wyprzedzającego (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)....544
12.7 Definiowanie obszarów Stop-Delay (DELAYFSTON, DELAYFSTOF)....547
12.8 Zablokowanie miejsca w programie dla SERUPRO (IPTRLOCK, IPTRUNLOCK)....550
12.9 Dosunięcie przywracające do konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)....553
12.10 Sterowanie prowadzeniem ruchu....562
12.10.1 Procentowa korekcja przyśpieszenia drugiego stopnia (JERKLIM)....562
12.10.2 Procentowa korekcja prędkości (VELOLIM)....563
12.10.3 Przykład programu dla JERKLIM i VELOLIM....565
12.11 Programowanie tolerancji konturu/orientacji (CTOL, OTOL, ATOL)....566
12.12 Zmiana bloku przy aktywnym sprzężeniu (CPBC)....571
13 Sprzężenia osi....573
13.1 Nadążanie (TRAILON, TRAILOF)....573
13.2 Tablice krzywych (CTAB)....578
13.2.1 Definiowanie tablic krzywych (CTABDEF, CTABEND)....578
13.2.2 Sprawdzenie istnienia pliku tablicy krzywych (CTABEXISTS)....585
13.2.3 Kasowanie tablic krzywych (CTABDEL)....585
13.2.4 Zablokowanie tablic krzywych przed skasowaniem i zastąpieniem (CTABLOCK, CTABUNLOCK)....586
13.2.5 Tablice krzywych: określenie właściwości tablicy (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)....587
13.2.6 Odczyt wartości tablicy krzywych (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)....589

13.2.7 Tablice krzywych: sprawdzić wykorzystanie zasobów (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)....594
13.3 Osiowe sprzężenie wartości wiodącej (LEADON, LEADOF)....596
13.4 Przekładnia elektroniczna (EG)....602
13.4.1 Zdefiniowanie przekładni elektronicznej (EGDEF)....602
13.4.2 Włączenie przekładni elektronicznej (EGON, EGONSYN, EGONSYNE)....603
13.4.3 Wyłączenie przekładni elektronicznej (EGOFS, EGOFC)....606
13.4.4 Skasowanie definicji przekładni elektronicznej (EGDEL)....607
13.4.5 Posuw na obrót (G95) / przekładnia elektroniczna (FPR)....608
13.5 Wrzeciono synchroniczne....609
13.5.1 Wrzeciono synchroniczne: Programowanie (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)....609
13.6 Sprzężenie rodzajowe (CP...)....620
13.7 Sprzężenie Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)....628
14 Akcje synchroniczne....631
14.1 Definicja akcji synchronicznej....631
15 Ruch wahliwy....633
15.1 Ruch wahliwy asynchroniczny (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)....633
15.2 Ruch wahliwy sterowany przez akcje synchroniczne (OSCILL)....639
16 Tłoczenie i cięcie....647
16.1 Aktywacja/Dezaktywacja....647
16.1.1 Włączenie/Wyłączenie tłoczenie i cięcia (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)....647
16.2 Automatyczny podział drogi....652
16.2.1 Podział drogi w przypadku osi uczestniczących w tworzeniu konturu....655
16.2.2 Podział drogi w przypadku pojedynczych osi....656
17 Szlifowanie....659
17.1 Włączenie/wyłączenie specyficznego dla szlifowania nadzoru nad narzędziem (TMON, TMOF)....659
18 Dalsze funkcje....661
18.1 Funkcje osi (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)....661
18.2 Przełączalne osie geometryczne (GEOAX)....664
18.3 Pojemnik osi (AXCTSWE, AXCTSWED, AXCTSWEC)....669
18.4 Czekanie na obowiązującą pozycję osi (WAITENC)....671
18.5 Programowalne przełączenie zestawu parametrów (SCPARA)....672
18.6 Sprawdzenie występującego zakresu językowego NC (STRINGIS)....674
18.7 Interaktywne wywołanie okna z programu obróbki (MMC)....678
18.8 Czas przebiegu programu / licznik obrabianych przedmiotów....683

18.8.1 Czas przebiegu programu....683
18.8.2 Licznik obrabianych przedmiotów....687
18.9 Process DataShare - Wyprowadzenie na zewnętrzne urządzenie/plik (EXTOPEN, WRITE, EXTCLOSE)....689
18.10 Alarmy (SETAL)....694
18.11 Rozszerzone zatrzymanie i wycofanie (ESR)....695
18.11.1 ESR prowadzone przez NC....696
18.11.1.1 Wycofanie prowadzone przez NC (POLF, POLFA, POLFMASK, POLFMLIN)....696
18.11.1.2 Zatrzymanie prowadzone przez NC....700
18.11.2 ESR wystarczające dla napędu....701
18.11.2.1 Zaprojektowanie zatrzymania niezależnego dla napędu (ESRS)....701
18.11.2.2 Zaprojektowanie wycofania niezależnego dla napędu (ESRR)....701
18.12 Definiowanie półfabrykatu (WORKPIECE)....703
18.13 Przełączanie trybu językowego (G290, G291)....707
19 Własne programy obróbki....709
19.1 Funkcje wspierające skrawanie....709
19.2 Sporządzenie tablicy konturu (CONTPRON)....710
19.3 Sporządzenie kodowanej tablicy konturu (CONTDCON)....716
19.4 Obliczenie punktu przecięcia między dwoma elementami konturu (INTERSEC)....720
19.5 Przejście pojedynczymi blokami elementów konturu w tablicy (EXECTAB)....722
19.6 Obliczenie danych okręgu (CALCDAT)....723
19.7 Wyłączenie przygotowania konturu (EXECUTE)....725
20 Zewnętrzne programowanie cykli....727
20.1 Cykle technologiczne....727
20.1.1 Wprowadzenie....727
20.1.2 Przegląd technologii....728
20.1.3 HOLES1 - Szereg otworów....730
20.1.4 HOLES2 - Okrąg otworów....731
20.1.5 POCKET3 - frezowanie kieszeni prostokątnej....733
20.1.6 POCKET4 - Frezowanie kieszeni kołowej....735
20.1.7 SLOT1 - rowek wzdłużny....738
20.1.8 SLOT2 - Rowek kołowy....740
20.1.9 LONGHOLE - otwór podłużny....743
20.1.10 CYCLE60 - Cykl grawerowania....745
20.1.11 CYCLE61 - Frezowanie płaszczyzny....747
20.1.12 CYCLE62 - Wywołanie konturu....749
20.1.13 CYCLE63 - Frezowanie konturu kieszeni....750
20.1.14 CYCLE64 - Wiercenie wstępne konturu kieszeni....752
20.1.15 CYCLE70 - Frezowanie gwintu....754
20.1.16 CYCLE72 - Frezowanie konturu....756
20.1.17 CYCLE76 - Frezowanie czopa prostokątnego....760
20.1.18 CYCLE77 - Frezowanie czopa kołowego....762
20.1.19 CYCLE78 - Wiercenie otworu z frezowaniem gwintu....764
20.1.20 CYCLE79 - Wielobok....767
20.1.21 CYCLE81 - wiercenie, nawiercanie....769

20.1.22 CYCLE82 - Wiercenie, pogłębianie czołowe....770
20.1.23 CYCLE83 - Wiercenie otworu głębokiego....773
20.1.24 CYCLE84 - Gwintowanie otworu bez oprawki kompensacyjnej....776
20.1.25 CYCLE85 - Rozwiercanie....779
20.1.26 CYCLE86 - Wytaczanie....780
20.1.27 CYCLE92 - Przecięcie....781
20.1.28 CYCLE95 - Skrawanie konturu....783
20.1.29 CYCLE98 - Łańcuch gwintów....785
20.1.30 CYCLE99 - Toczenie gwintu....789
20.1.31 CYCLE435 - Ustawienie współrzędnych obciągacza....794
20.1.32 CYCLE495 - Profilowanie....794
20.1.33 CYCLE800 - skręt....796
20.1.34 CYCLE801 - Siatka albo ramka....800
20.1.35 CYCLE802 - Dowolne pozycje....801
20.1.36 CYCLE830 - Wiercenie otworu głębokiego 2....804
20.1.37 CYCLE832 - High Speed Settings....810
20.1.38 CYCLE840 - Gwintowanie otworu z oprawką kompensacyjną....813
20.1.39 CYCLE899 - Frezowanie rowka otwartego....816
20.1.40 CYCLE930 - Rowek....819
20.1.41 CYCLE940 - Podcięcie....821
20.1.42 CYCLE951- Skrawanie warstwowe....824
20.1.43 CYCLE952 - Toczenie wcinające konturu (poprzeczne)....827
20.1.44 CYCLE4071 - Szlifowanie wzdłużne z dosuwem w punkcie nawrotu....833
20.1.45 CYCLE4072 -Szlifowanie wzdłużne z dosuwem w punkcie nawrotu i sygnał anulowania....834
20.1.46 CYCLE4073 - Szlifowanie wzdłużne z dosuwem ciągłym....838
20.1.47 CYCLE4074 - Szlifowanie wzdłużne z dosuwem ciągłym i sygnałem przerwania....839
20.1.48 CYCLE4075 - Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu....842
20.1.49 CYCLE4077 - Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu i przerwania sygnału....845
20.1.50 CYCLE4078 -Szlifowanie płaszczyzny z dosuwem ciągłym....848
20.1.51 CYCLE4079 -Szlifowanie płaszczyzny z dosuwem przerywanym....850
20.1.52 GROUP_BEGIN - Początek bloku programu....852
20.1.53 GROUP_END - Koniec bloku programu....853
20.1.54 GROUP_ADDEND - Koniec dodatkowego dosuwu....853
20.1.55 Warunki brzegowe....853
20.1.55.1 Skalowanie technologiczne w ramach okien cykli....853
20.2 Cykle pomiarowe....856
21 Tablice....857
21.1 Instrukcje....857
21.2 Instrukcje: Dostępność w przypadku SINUMERIK 828D....897
21.2.1 Warianty sterowania - frezowanie / toczenie....897
21.2.2 Warianty sterowania - szlifowanie....924
21.3 Aktualny język w HMI....952
A Aneks....953
A.1 Lista skrótów....953
A.2 Przegląd dokumentacji....962

Glosariusz....963
Indeks....985

Podstawowe wskazówki bezpieczeństwa 1

1.1 Ogólne wskazówki bezpieczeństwa

⚠ OSTRZEŻENIE

Zagrożenie życia przez nieprzestrzeganie wskazówek bezpieczeństwa i pozostałego ryzyka

W przypadku nieprzestrzegania wskazówek bezpieczeństwa i pozostałego ryzyka w przynależnej dokumentacji sprzętu mogą wystąpić wypadki z ciężkimi uszkodzeniami ciała albo skutkiem śmiertelnym.

- Należy przestrzegać wskazówek bezpieczeństwa zawartych w dokumentacji sprzętu.
- Przy ocenie ryzyka należy uwzględnić pozostałe ryzyko.

⚠ OSTRZEŻENIE

Nieprawidłowe działania maszyny wskutek błędnej albo zmienionej parametryzacji

W wyniku błędnej lub zmienionej parametryzacji mogą wystąpić w maszynie błędne funkcje, które mogą prowadzić do obrażeń ciała lub śmierci.

- Należy chronić parametryzację przed nieuprawnionym dostępem.
- Należy skontrolować możliwe awarie przez odpowiednie środki (np. EMERGENCY-STOP lub EMERGENCY-OFF).

1.2 Gwarancja i odpowiedzialność za przykłady aplikacyjne

Przykłady aplikacyjne nie mają charakteru wiążącego i nie są kompletne pod względem konfiguracji i wyposażenia oraz wszelkich innych ewentualności. Przykłady aplikacyjne nie przedstawiają rozwiązań specyficznych dla klienta, lecz stanowią jedynie pomoc w przypadku standardowych zadań.

Użytkownik odpowiada za stosowanie opisanego produktu zgodnie z przeznaczeniem. Przykłady aplikacyjne nie zwalniają z obowiązku bezpiecznego obchodzenie się podczas używania, instalacji, obsługi i serwisowania.

1.3 Industrial Security

Uwaga

Industrial Security

Siemens oferuje produkty i rozwiązania w zakresie funkcji Industrial Security, które pomagają w zapewnieniu bezpiecznego działania urządzeń, systemów, maszyn i sieci.

W celu zabezpieczenia urządzeń, systemów, maszyn i sieci przed cyberniebezpieczeństwami konieczne jest globalne wdrożenie koncepcji Industrial Security (i jej utrzymanie), która odpowiada aktualnemu stanowi techniki. Produkty i rozwiązania firmy Siemens stanowią tylko jeden element takiej koncepcji.

Klient jest odpowiedzialny za zapobieganie przed nieuprawnionym dostępem do swoich urządzeń, systemów, maszyn i sieci. Systemy, maszyny i komponenty powinny być połączone z siecią firmową lub internetem tylko, jeśli jest to konieczne; należy wówczas zastosować odpowiednie środki ochrony (np. firewall lub segmentacja sieci).

Należy także przestrzegać zaleceń firmy Siemens dotyczących odpowiednich środków ochrony. Dalsze informacje dotyczące Industrial Security można znaleźć pod adresem:

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

Produkty i rozwiązania firmy Siemens są stale rozwijane, by uczynić je jeszcze bezpieczniejszymi. Firma Siemens wyraźnie zaleca przeprowadzanie aktualizacji, w razie ich dostępności, i korzystanie zawsze tylko z aktualnej wersji produktu. Korzystanie z nieaktualnych lub nieobsługiwanych już wersji zwiększa ryzyko cyberniebiezpieczeństw.

Aby otrzymywać na bieżąco informacje o aktualizacjach produktu, należy zgłosić chęć otrzymywania Siemens Industrial Security RSS Feed na stronie:

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

Dalsze informacje znajdują się w internecie na stronie:

Podręcznik projektowania Industrial Security (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ OSTRZEŻENIE

Niebezpieczne warunki pracy w wyniku manipulacji w oprogramowaniu

Manipulacje w oprogramowaniu np. wirusy, konie trojańskie, programy szpiegujące lub robaki mogą powodować niebezpieczne warunki pracy urządzenia, które mogą prowadzić do śmierci, ciężkich obrażeń ciała i szkód materialnych.

- Należy aktualizować oprogramowanie.
- Należy zintegrować komponenty automatyki i komponenty napędowe w całościową koncepcję Industrial Security urządzenia lub maszyny zgodnie z najnowszym stanem techniki.
- Wziąć pod uwagę wszystkie produkty używane w całościowej koncepcji Industrial Security.
- Należy chronić pliki na wymiennych nośnikach pamięci przed złośliwym oprogramowaniem podejmując odpowiednie środki ochrony, np.: program antywirusowy.
- Zabezpieczyć napęd przed nieautoryzowanymi zmianami, aktywując funkcję przekształtnika „Ochrona Know-how".

Elastyczne programowanie NC

2

2.1 Zmienne

Dzięki zastosowaniu zmiennych z obszaru danych systemowych i użytkownika, w szczególności w połączeniu z funkcjami obliczeniowymi i strukturami kontrolnymi, można ekstremalnie elastycznie kształtować programy obróbki NC i cykle.

- Dane systemowe
  Dane systemowe zawierają wstępnie zdefiniowane w systemie zmienne. Zmienne te mają zdefiniowane znaczenie. Są one głównie wykorzystywane przez oprogramowanie systemowe. Użytkownik może je czytać i zapisywać w programach NC i cyklach. Przykład: Dane maszynowe, dane ustawcze, zmienne systemowe.
  Mimo, że znaczenie danych systemowych jest określone, to właściwości mogą być w pewnym stopniu zmieniane przez użytkownika za pomocą redefiniowania.
  Patrz "Redefinicja danych systemowych, danych użytkownika i poleceń językowych NC (REDEF) (Strona 37)"
- Dane użytkownika
  Dane użytkownika zawierają zdefiniowane przez użytkownika zmienne, których znaczenie jest określane wyłącznie przez użytkownika. Nie są one wykorzystywane przez system. Dane użytkownika są podzielone na:
  - Wstępnie zdefiniowane zmienne użytkownika
    Wstępnie zdefiniowanymi zmiennymi użytkownika są zmienne już zdefiniowane w systemie, których liczbę należy określić w danych maszynowych. Właściwości tych zmiennych mogą być dopasowywane przez użytkownika.
    Patrz "Redefinicja danych systemowych, danych użytkownika i poleceń językowych NC (REDEF) (Strona 37)".
  - Zmienne definiowane przez użytkownika
    Zmienne definiowane przez użytkownika są zmiennymi, które są definiowane przez użytkownika i tworzone przez system w czasie przebiegu. Ich liczbę, typ danych, widoczność i wszystkie dalsze właściwości ustala wyłącznie użytkownik.
    Patrz "Definicja zmiennych użytkownika (DEF) (Strona 31)"

2.1.1 Dane systemowe

Dane systemowe zawierają zmienne wstępnie zdefiniowane w systemie, które w programach NC i cyklach zapewniają dostęp do aktualnej parametryzacji sterowania, jak też do stanów maszyny, sterowania i procesu.

Zmienne przebiegu wyprzedzającego

Jako zmienne przebiegu wyprzedzającego są określane zmienne systemowe, które są czytane i zapisywane w kontekście przebiegu wyprzedzającego, np. w chwili interpretacji bloku programu obróbki, w którym jest zaprogramowana zmienna systemowa. Zmienne przebiegu wyprzedzającego nie wyzwalają zatrzymania tego przebiegu.

Zmienne przebiegu głównego

Jako zmienne przebiegu głównego są określane zmienne systemowe, które są czytane i zapisywane w kontekście przebiegu głównego, np. w chwili bloku programu obróbki, w którym jest zaprogramowana zmienna systemowa. Zmiennymi przebiegu głównego są:

- zmienne, które mogą być programowane w akcjach synchronicznych (odczyt/zapis)
- zmienne, które mogą być programowane w programie NC i wyzwalają zatrzymanie przebiegu wyprzedzającego (odczyt/zapis)
- zmienne, które mogą być programowane w programie NC, a których wartość jest określana w przebiegu wyprzedzającym, ale zapisywana dopiero w przebiegu głównym (synchronicznie z przebiegiem głównym: tylko zapis)

Klasyfikacja prefiksów

W celu specjalnego oznaczenia danych systemowych nazwa jest poprzedzona prefiksem, który składa się ze znaku $, po którym następuje jedna lub dwie litery i podkreślenie:

$ + 1. Litera	**Znaczenie: Rodzaj danych**
Dane przebiegu wyprzedzającego (dane systemowe, które są czytane / zapisywane w przebiegu wyprzedzającym)	
$M	Dane maszynowe [1)]
$S	Dane ustawcze, obszary ochrony [1)]
$T	Dane zarządzania narzędziami
$P	Wartości programowane
$C	Zmienne cykli ISO
$O	Dane opcji
R	Parametry R (parametry obliczeniowe) [2)]
Dane przebiegu głównego (dane systemowe, które są czytane / zapisywane w przebiegu głównym)	
$$M	Dane maszynowe [1)]
$$S	Dane ustawcze [1)]
$A	Aktualne dane przebiegu głównego
$V	Dane położenia regulatora
$R	Parametry R (parametry obliczeniowe) [2)]
[1)] To czy dane maszynowe i ustawcze są traktowane, jako zmienne przebiegu wyprzedzającego, czy jako zmienne przebiegu głównego zależy od tego, czy są pisane z jednym czy z dwoma znakami $. Sposób zapisu można wybrać odpowiednio do zastosowania. [2)] Przy zastosowaniu parametru R w programie obróbki / cyklu jako zmienna przebiegu wyprzedzającego prefiks nie jest pisany, np. R10. Przy zastosowaniu w akcji synchronicznej jako zmienną przebiegu głównego prefiks określony jest znakiem $, np. $R10.	

2. litera	**Znaczenie: Widoczność**
N	Zmienna globalna dla NC (**N**C)
C	Zmienna kanałowa (**C**hannel)
A	Zmienna osiowa (**A**xis)

Warunki brzegowe

Wyjątki w usystematyzowaniu prefiksów

Następujące zmienne systemowe odbiegają od podanej klasyfikacji prefiksów:

- $TC_...: 2. litera C wskazuje tutaj nie na kanałowe, lecz na specyficzne dla uchwytu narzędzia zmienne systemowe (TC = Tool Carrier)
- $P_ ...: Specyficzne dla kanału zmienne systemowe

Zastosowanie danych maszynowych i ustawczych w akcjach synchronicznych

Przy zastosowaniu danych maszynowych i ustawczych w akcjach synchronicznych można określić poprzez prefiks, czy dane te są czytane/zapisywane synchronicznie z przebiegiem wyprzedzającym czy głównym.

Gdy dana nie zmienia się podczas obróbki, odczyt może następować synchronicznie z przebiegiem wyprzedzającym. W tym celu prefiks danej maszynowej lub ustawczej jest pisany ze znakiem $:

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

Gdy dana zmienia się podczas obróbki, odczyt / zapis musi następować synchronicznie z przebiegiem głównym. W tym celu prefiks danej maszynowej lub ustawczej pisany jest z dwoma znakami $:

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

Uwaga

Zapis danych maszynowych i ustawczych

Przy zapisie danej maszynowej lub ustawczej należy zwracać uwagę, by aktywny poziom dostępu przy wykonywaniu programu obróbki / cyklu pozwalał na dostęp w celu zapisu i by działanie danej było "IMMEDIATE".

Literatura

Szczegółowy przegląd wszystkich zmiennych systemowych znajduje się w:

Podręcznik Lista parametrów - zmienne systemowe

Patrz również

Zmienne (Strona 23)

2.1.2 Wstępnie zdefiniowane zmienne użytkownika: Parametry obliczeniowe

2.1.2.1 Specyficzne dla kanału parametry obliczeniowe (R)

Specyficzne dla kanału parametry obliczeniowe lub inaczej parametry R są predefiniowanymi zmiennymi użytkownika o określeniu R, definiowanymi jako tablica typu danych REAL. Ze względów historycznych dla parametrów R oprócz pisowni z indeksem tablicy np. `R[10]` jest również dozwolona pisownia bez indeksu tablicy np. `R10`.

Przy zastosowaniu w akcjach synchronicznych musi nastąpić poprzedzenie literą $ np. `$R10`.

Składnia

Przy zastosowaniu jako zmienna przebiegu wyprzedzającego:
`R<n>`
`R[<wyrażenie>]`

Przy zastosowaniu jako zmienna przebiegu głównego:
`$R<n>`
`$R[<wyrażenie>]`

Znaczenie

`R`:	Identyfikator przy zastosowaniu, jako zmienna przebiegu wyprzedzającego np. w programie obróbki	
`$R`:	Identyfikator przy zastosowaniu, jako zmienna przebiegu głównego, np. w akcjach synchronicznych	
	Typ:	REAL
	Zakres wartości:	Przy zapisie nie wykładniczym: ± (0.000 0001 ... 9999 9999) **Wskazówka:** Jest dozwolonych maksymalnie 8 miejsc dziesiętnych
		Przy zapisie wykładniczym: ± ($1*10^{-300}$ ... $1*10^{+300}$) **Wskazówka:** • Notacja: <mantysa>EX<wykładnik> np. 8.2EX-3 • Jest dozwolonych maksymalnie 10 znaków łącznie ze znakiem liczby i kropką dziesiętną.
`<n>`:	Numer parametru R	
	Typ:	INT
	Zakres wartości:	0 - MAX_INDEX **Wskazówka:** MAX_INDEX wynika ze sparametryzowanej liczby parametrów R: MAX_INDEX = (MD28050 $MN_MM_NUM_R_PARAM) - 1
`<wyrażenie>`:	Indeks tablicy Jako indeks tablicy można podawać dowolne wyrażenie, jak długo wynik wyrażenia można zamienić na typ danych INT (INT, REAL, BOOL, CHAR)	

Przykład

Przyporządkowania do parametrów R i zastosowanie parametrów R w funkcjach matematycznych:

Kod programu	**Komentarz**
R0=3.5678	; Przyporządkowanie w przebiegu wyprzedzającym
R[1]=-37.3	; Przyporządkowanie w przebiegu wyprzedzającym
R3=-7	; Przyporządkowanie w przebiegu wyprzedzającym
$R4=-0.1EX-5	; Przyporządkowanie w przebiegu głównym: R4 = -0.1 * 10^-5
$R[6]=1.874EX8	; Przyporządkowanie w przebiegu głównym: R6 = 1.874 * 10^8
R7=SIN(25.3)	; Przyporządkowanie w przebiegu wyprzedzającym
R[R2]=R10	; Adresowanie pośrednie przez parametry R
R[(R1+R2)*R3]=5	Adresowanie pośrednie przez wyrażenie matematyczne
X=(R1+R2)	; Wykonaj ruch w osi X do pozycji wynikającej z sumy R1 i R2
Z=SQRT(R1*R1+R2*R2)	; Wykonaj ruch w osi Z do pozycji pierwiastek kwadratowy z (R1^2 + R2^2)

Patrz również

Zmienne (Strona 23)

2.1.2.2 Globalne parametry obliczeniowe (RG)

Funkcja

Oprócz specyficznych dla kanału parametrów R do dyspozycji są również globalne parametry R. Występują one w ramach sterowania jeden raz i mogą być czytane/zapisywane ze wszystkich kanałów.

Globalne parametry R są np. stosowane do przenoszenia informacji między kanałami. Innym przykładem są ustawienia globalne, które mają być poddawane ewaluacji dla wszystkich kanałów, jak np. wysięg półfabrykatu z wrzeciona.

Odczyt i zapis globalnych parametrów R następuje poprzez interfejs graficzny albo w programie NC w przebiegu wyprzedzającym. Zastosowanie w akcjach synchronicznych albo cyklach technologicznych jest niemożliwe.

Uwaga

Przy odczycie i zapisie globalnych parametrów R **nie** następuje synchronizacja między kanałami.

Ponieważ odczyt i zapis następuje w przebiegu wyprzedzającym, chwila, w której wartość zapisana w jednym kanale będzie działać w innym kanale, nie jest zdefiniowana.

Przykład:

W kanale1 jest wykonywana pętla z globalnym parametrem R jako licznikiem pętli. Kanał 2 zapisuje wartość do tego globalnego parametru R, która prowadzi do przerwania pętli w kanale 1. Wszystkie pętle zinterpretowane do tej chwili w kanale 1 w przebiegu wyprzedzającym zostaną jednak jeszcze wykonane. Ile jest to pętli jest niezdefiniowane i zależy między innymi od obciążenia kanału.

Synchronizację między kanałami użytkownik musi sam zrealizować aplikatywnie, tzn. np. przy użyciu znaczników WAIT!

Składnia

Zapis w programie NC

```
RG[<n>]=<wartość>
RG[<Wyrażenie>]=<Wartość>
```

Odczyt w programie NC

```
R...=RG[<n>]
R...=RG[<wyrażenie>]
```

Znaczenie

RG:	Nazw domyślna adresu NC dla **globalnych** parametrów R **Uwaga:** Nazwa adresu NC jest ustawiana za pomocą MD15800 $MN_R_PARAM_NCK_NAME	
<n>:	Numer globalnego parametru R	
	Typ:	INT
	Zakres wartości:	0 ... MAX_INDEX **Wskazówka:** MAX_INDEX wynika ze sparametryzowanej liczby globalnych parametrów R: MAX_INDEX = (MD18156 $MN_MM_NUM_R_PARAM_NCK) - 1
<wyrażenie>:	Jako indeks tablicy można podawać dowolne wyrażenie, jak długo wynik wyrażenia można zamienić na typ danych INT (INT, REAL, BOOL, CHAR)	

<Wartość>:	Wartość globalnego parametru R	
	Typ:	REAL
	Zakres wartości:	Przy zapisie nie wykładniczym: ± (0.000 0001 ... 9999 9999) **Wskazówka:** Jest dozwolonych maksymalnie 8 miejsc dziesiętnych
		Przy zapisie wykładniczym: ± ($1*10^{-300}$ ... $1*10^{+300}$) **Wskazówka:** • Notacja: <mantysa>EX<wykładnik> np. 8.2EX-3 • Dozwolonych jest maksymalnie 10 znaków łącznie ze znakiem liczby i kropką dziesiętną.

2.1.3 Predefiniowane zmienne użytkownika: zmienne link

Poprzez zmienne link mogą w ramach funkcji "NCU-Link" być wymieniane dane cykliczne między NCU, które są ze sobą połączone w sieci. Umożliwiają one przy tym specyficzny dla formatu danych dostęp do pamięci zmiennych link. Pamięć zmiennych link jest odnośnie zarówno wielkości jak też struktury danych ustalana przez użytkownika / producenta maszyny specyficznie dla urządzenia.

Zmienne link są globalnymi dla systemu zmiennymi użytkownika, które przy zaprojektowanej komunikacji link mogą być czytane i zapisywane w programach obróbki i cyklach przez wszystkie NCU zespołu link. W przeciwieństwie do globalnych zmiennych użytkownika (GUD) zmienne link mogą być stosowane również w akcjach synchronicznych.

W przypadku urządzeń bez aktywnej NCU-Link, zmienne Link mogą lokalnie w sterowaniu być stosowane oprócz globalnych zmiennych użytkownika (GUD), jako dodatkowe globalne zmienne użytkownika.

Składnia

```
$A_DLB[<indeks>]
$A_DLW[<indeks>]
$A_DLB[<indeks>]
$A_DLW[<indeks>]
```

Znaczenie

$A_DLB:	Zmienna link dla formatu danych BYTE (1 bajt)	
	Typ danych:	UINT
	Zakres wartości:	0 ... 255
$A_DLW:	Zmienna link dla formatu danych WORD (2 bajty)	
	Typ danych:	INT
	Zakres wartości:	-32768 ... 32767

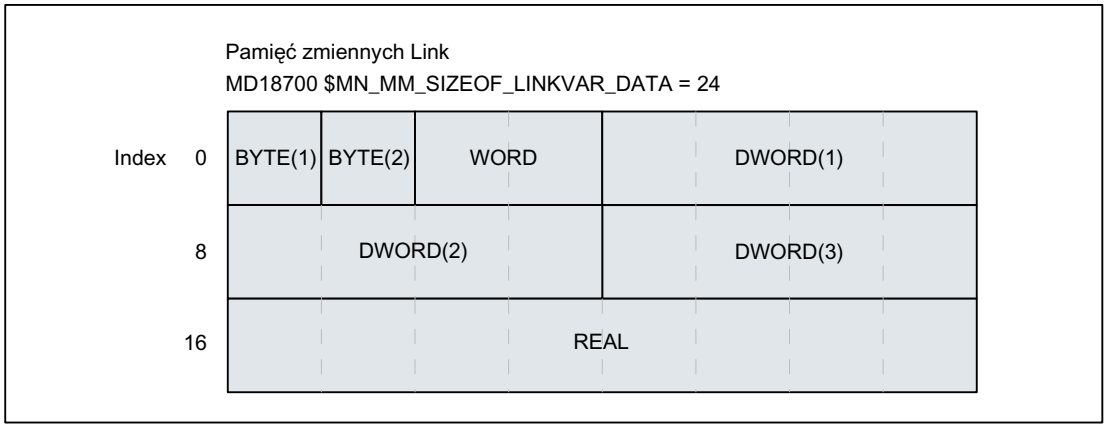
`$A_DLD`:	Zmienna link dla formatu danych DWORD (4 bajty)	
	Typ danych:	INT
	Zakres wartości:	-2147483648 ... 2147483647
`$A_DLR`:	Zmienna link dla formatu danych REAL (8 bajtów)	
	Typ danych:	REAL
	Zakres wartości:	$\pm(2{,}2*10^{-308} \ldots 1{,}8*10^{+308})$
`<indeks>`:	Indeks adresowy w bajtach, licząc od początku pamięci zmiennych link	
	Typ danych:	INT
	Zakres wartości:	0 - MAX_INDEX **Wskazówka** • MAX_INDEX wynika ze sparametryzowanej wielkości pamięci zmiennych link: MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1 • Wolno programować indeksy tylko tak, by bajty adresowane w pamięci zmiennych link leżały na granicy formatu danych ⇒ indeks = n * bajtów, gdzie n = 0, 1, 2, ... – `$A_DLB[i]`: i = 0, 1, 2, ... – `$A_DLW[i]`: i = 0, 2, 4, ... – `$A_DLD[i]`: i = 0, 4, 8, ... – `$A_DLR[i]`: i = 0, 8, 16, ...

Przykład

W urządzeniu automatyzacyjnym są 2 NCU: NCU1 i NCU2. Do NCU1 jest przyłączona oś maszyny AX2, ruch w której jest wykonywany z NCU2, jako w osi link.

NCU1 zapisuje cyklicznie wartość zadaną prądu ($VA_CURR) osi AX2 w pamięci zmiennych link. NCU2 czyta cyklicznie wartość rzeczywistą prądu przesyłaną przez komunikację link i przy przekroczeniu wartości granicznej wyświetla alarm 61000.

Struktura danych w pamięci zmiennych link jest przedstawiona na następującym rysunku. Wartość rzeczywista prądu jest przenoszona przez wartość REAL.



NCU1

NCU1 zapisuje w statycznej akcji synchronicznej, cyklicznie w takcie IPO, wartość rzeczywistą prądu osi AX2 przez zmienną link $A_DLR[16] w pamięci zmiennych link.

Kod programu

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

NCU2

NCU2 czyta w statycznej akcji synchronicznej, cyklicznie w takcie IPO, wartość rzeczywistą prądu osi AX2 przez zmienną link $A_DLR[16] z pamięci zmiennych link. Jeżeli wartość rzeczywista prądu jest większa niż 23.0 A, jest wyświetlany alarm 61000.

Kod programu

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

Patrz również

Zmienne (Strona 23)

2.1.4 Definicja zmiennych użytkownika (DEF)

Przy pomocy polecenia `DEF` można definiować własne zmienne lub zmienne użytkownika (User Data) i określać ich wartości.

Odpowiednio do zakresu obowiązywania, tzn. zakresu w którym zmienna jest widoczna, są następujące kategorie zmiennych użytkownika:

- Lokalne zmienne użytkownika (LUD)
  Lokalnymi zmiennymi użytkownika (LUD) są zmienne, które są zdefiniowane w programie NC, który w chwili wykonywania nie jest programem głównym. Są one tworzone przy wywołaniu programu NC i kasowane z końcem tego programu lub następnym załączeniem sterowania. Do LUD można mieć dostęp tylko w ramach tego programu NC, w którym są one zdefiniowane.
- Dane użytkownika globalne dla programu (PUD)
  Globalne dla programu zmienne użytkownika (PUD) są zmiennymi, które są definiowane w programie NC, będącym programem głównym. Są one tworzone przy wywołaniu programu NC i kasowane z końcem tego programu lub następnym załączeniem sterowania. Do PUD można mieć dostęp w programie głównym i we wszystkich podprogramach.

  Uwaga

  Dostępność globalnych w programie zmiennych użytkownika (PUD)

  Zdefiniowane w programie głównym zmienne użytkownika (PUD) są tylko wtedy dostępne w podprogramach, gdy jest ustawiona następująca dana maszynowa:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Za pomocą MD11120 = 0 zdefiniowane w programie głównym zmienne użytkownika są dostępne tylko w programie głównym.

- Globalne zmienne użytkownika (GUD)
  Globalne zmienne użytkownika (GUD) są zmiennymi globalnymi dla NC lub kanału, które są zdefiniowane w module danych (SGUD, MGUD, UGUD, GUD4 ... GUD9) i pozostają zachowane po resecie na końcu programu lub po następnym załączeniu sterowania. Do GUD można sięgać we wszystkich programach NC.

Zmienne użytkownika muszą być zdefiniowane przed ich zastosowaniem (odczyt / zapis). Należy przy tym przestrzegać następujących zasad:

- GUD muszą być zdefiniowane w pliku definicji, np. _N_DEF_DIR/_N_UGUD_DEF.
- PUD i LUD muszą być zdefiniowane w części definicyjnej programu NC.
- Definicja danych musi nastąpić we oddzielnym bloku.
- W definicji danych wolno zastosować tylko jeden typ danych.
- W definicji można zdefiniować wiele zmiennych o takim samym typie danych.

Składnia

LUD i PUD

```
DEF <typ> <jednostka_fizyczna> <wartości graniczne>
<nazwa>[<wartość_1>, <wartość_2>,
<wartość_3>]=<wartość_inicjalizacyjna>
```

GUD

```
DEF <zakres> <stop_przebiegu_wyprzedz.> <prawa dostępu> <klasa
danych> <typ> <jednostka_fizyczna> <wartości graniczne>
```

```
<nazwa>[<wartość_1>, <wartość_2>,
<wartość_3>]=<wartość_inicjalizacyjna>
```

Znaczenie

DEF:	Polecenie do definicji zmiennych użytkownika GUD, PUD, LUD	
<zakres>:	Zakres obowiązywania, ma znaczenie tylko dla GUD:	
	NC:	Zmienna użytkownika globalna dla NC
	CHAN:	Zmienna użytkownika globalna dla kanału
<stop_przebiegu_wyprzedz.>:	Zatrzymanie przebiegu wyprzedzającego, ma znaczenie tylko dla GUD (opcjonalnie)	
	SYNR:	Zatrzymanie przebiegu wyprzedzającego przy odczycie
	SYNW:	Zatrzymanie przebiegu wyprzedzającego przy zapisie
	SYNRW:	Zatrzymanie przebiegu wyprzedzającego przy odczycie/ zapisie
<Prawa dostępu>:	Poziom dostępu dla odczytu / zapisu GUD przez program NC lub BTSS-panel obsługi (opcjonalnie)	
	APRP <poziom dostępu>:	Odczyt: Program NC
	APWP <poziom dostępu>:	Zapis: Program NC
	APRB <poziom dostępu>:	Odczyt: BTSS
	APWB <poziom dostępu>:	Zapis: BTSS
	<poziom dostępu>:	Zakres wartości: 0 ... 7
	Patrz "Atrybut: prawa dostępu (APR, APW, APRP, APWP, APRB, APWB) (Strona 47)"	
<Klasa danych>:	Przyporządkowanie klasy danych (tylko SINUMERIK 828D!)	
	DCM:	Klasa danych M (= Manufacturer)
	DCI:	Klasa danych I (= Individual)
	DCU:	Klasa danych U (= User)
	Patrz "Atrybut: Klasa danych (DCM, DCI, DCU) - tylko SINUMERIK 828D (Strona 53)".	
<Typ>:	Typ danych:	
	INT:	Wartość całkowitoliczbowa ze znakiem
	REAL:	Liczba rzeczywista (LONG REAL według IEEE)
	BOOL:	Wartość logiczna TRUE (1) / FALSE (0)
	CHAR:	Znak ASCII
	STRING[<max długość>]:	Łańcuch znaków o zdefiniowanej długości
	AXIS:	Identyfikator osi/wrzeciona
	FRAME:	Dane geometryczne dla statycznej transformacji współrzędnych
	Patrz "Typy danych (Strona 61)"	
<Jednostka_fizyczna>:	Jednostka fizyczna (opcjonalnie)	
	PHU <jednostka>:	jednostka fizyczna
	Patrz "Atrybut: jednostka fizyczna (PHU) (Strona 45)"	

<Wartości graniczne>:	dolna / górna wartość graniczna (opcjonalnie)	
	LLI <wartość graniczna>:	Dolna wartość graniczna (lower limit)
	ULI <wartość graniczna>:	Górna wartość graniczna (upper limit)
	Patrz "Atrybut: wartości graniczne (LLI, ULI) (Strona 43)"	
<nazwa>:	Nazwa zmiennej **Wskazówka** • Maksymalnie 31 znaków • Obydwa pierwsze znaki muszą być literą i/lub podkreśleniem. • Znak "$" jest zarezerwowany dla zmiennych systemowych i nie wolno go stosować.	
[<wartość_1>, <wartość_2>, <wartość_3>]:	Podanie wielkości tablicy dla zmiennych tablicowych 1- do maks. 3-wymiarowych (opcjonalnie) Odnośnie inicjalizacji zmiennych tablicowych patrz "Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP) (Strona 55)"	
<Wartość_inicj>:	Wartość inicjalizacyjna (opcjonalnie) Patrz "Atrybut: wartość inicjalizacyjna (Strona 40)" Odnośnie inicjalizacji zmiennych tablicowych patrz "Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP) (Strona 55)"	

Przykłady

Przykład 1: Definicje zmiennych użytkownika w module danych dla producenta maszyny

Kod programu	**Komentarz**
%_N_MGUD_DEF	; Moduł GUD: Producent maszyny
$PATH=/_N_DEF_DIR	
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2	
; Opis	
; Definicja dwóch GUD: STROM_1, STROM_2	
; Zakres obowiązywania: W kanale	
; Typ danych: REAL	
; Stop przeb. wyprz. nie zaprogramowano => wartość domyślna = brak VL-Stop	
; Jednostka fizyczna: 24 = [A]	
; Wartości graniczne: Low = 0.0, High = 10.0	
; Prawa dostępu: nie zaprogramowane => wartość domyślna = 7 = położenie 0 przełącznika z kluczykiem	
; Wartość inicjalizacyjna: nie zaprogramowana => wartość domyślna = 0.0	
DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45	
; Opis	
; Definicja dwóch GUD: ZEIT_1, ZEIT_2	
; Zakres obowiązywania: szerokość w NC	
; Typ danych: REAL	
; Stop przeb. wyprz. nie zaprogramowano => wartość domyślna = brak VL-Stop	
; Jednostka fizyczna: 13 = [s]	

Kod programu	Komentarz
; Wartości graniczne: Low = 10.0, High = nie zaprogramowano => górna granica zakresu definicji	
; Prawa dostępu:	
; Program NC: zapis/odczyt = 3 = użytkownik końcowy	
; BTSS: zapis = 0 = Siemens, odczyt = 3 = użytkownik końcowy	
; Wartość inicjalizacyjna: ZEIT_1 = 12.0, ZEIT_2 = 45.0	
DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER"	
; Opis	
; Definicja GUD: GUD5_NAME	
; Zakres obowiązywania: szerokość w NC	
; Typ danych: STRING, max 5 znaków	
; Stop przeb. wyprz. nie zaprogramowano => wartość domyślna = brak VL-Stop	
; Jednostka fizyczna: nie zaprogramowano => wartość domyślna = 0 = bez jednostki fizycznej	
; Wartości graniczne: nie zaprogramowano => granice zakresu definicji: Low = 0, High = 255	
; Prawa dostępu:	
; Program NC: zapis/odczyt = 3 = użytkownik końcowy	
; BTSS: zapis = 0 = Siemens, odczyt = 3 = użytkownik końcowy	
; Wartość inicjalizacyjna: "COUNTER"	
M30	

Przykład 2: Globalne i lokalne zmienne użytkownika w programie (PUD / LUD)

Kod programu	Komentarz
PROC MAIN	; Program główny
DEF INT VAR1	; Definicja PUD
...	
SUB2	; Wywołanie podprogramu
...	
M30	

Kod programu	Komentarz
PROC SUB2	; Podprogram SUB2
DEF INT VAR2	; DEFINICJA LUD
...	
IF (VAR1==1)	; Odczyt PUD
VAR1=VAR1+1	; Odczyt i zapis PUD
VAR2=1	; Zapis LUD
ENDIF	
SUB3	; Wywołanie podprogramu
...	
M17	

Kod programu	Komentarz
PROC SUB3	; Podprogram SUB3
...	
IF (VAR1==1)	; Odczyt PUD
VAR1=VAR1+1	; Odczyt i zapis PUD
VAR2=1	; Błąd: LUD z SUB2 nie jest znana
ENDIF	
...	
M17	

Przykład 3: Definicja i zastosowanie zmiennych użytkownika o typie danych AXIS

Kod programu	Komentarz
DEF AXIS ABSZISSE	; 1. oś geometryczna
DEF AXIS SPINDLE	; Wrzeciono
...	
IF ISAXIS(1) == FALSE GOTOF WEITER	
ABSZISSE = $P_AXN1	
WEITER:	
...	
SPINDLE=(S1)	; 1. Wrzeciono
OVRA[SPINDLE]=80	; Korekcja wrzeciona = 80%
SPINDLE=(S3)	; 3. Wrzeciono

Warunki brzegowe

Globalne zmienne użytkownika (GUD)

W ramach definicji globalnych zmiennych użytkownika (GUD) należy uwzględnić następujące dane maszynowe:

Nr	Identyfikator: $MN_	Znaczenie
11140	GUD_AREA_ SAVE_TAB	Dodatkowe zabezpieczenie dla modułów GUD
18118 [1)]	MM_NUM_GUD_MODULES	Liczba plików GUD w aktywnym systemie plików
18120 [1)]	MM_NUM_GUD_NAMES_NCK	Liczba globalnych nazw GUD
18130 [1)]	MM_NUM_GUD_NAMES_CHAN	Liczba specyficznych dla kanału nazw GUD
18140 [1)]	MM_NUM_GUD_NAMES_AXIS	Liczba specyficznych dla osi nazw GUD
18150 [1)]	MM_GUD_VALUES_MEM	Miejsce w pamięci dla globalnych wartości GUD
18660 [1)]	MM_NUM_SYNACT_GUD_REAL	Liczba projektowanych GUD typ danych REAL
18661 [1)]	MM_NUM_SYNACT_GUD_INT	Liczba projektowanych GUD typ danych INT
18662 [1)]	MM_NUM_SYNACT_GUD_BOOL	Liczba projektowanych GUD typ danych BOOL
18663 [1)]	MM_NUM_SYNACT_GUD_AXIS	Liczba projektowanych GUD typ danych AXIS

Nr	Identyfikator: $MN_	Znaczenie
18664 [1)]	MM_NUM_SYNACT_GUD_CHAR	Liczba projektowanych GUD typ danych CHAR
18665 [1)]	MM_NUM_SYNACT_GUD_STRING	Liczba projektowanych GUD typ danych STRING

[1)] MD w przypadku SINUMERIK 828D tylko do odczytu!

Wykraczające poza kanał zastosowanie globalnej zmiennej użytkownika w NC o typie danych AXIS

Globalna zmienna użytkownika w NC o typie danych `AXIS`, która przy definicji w module danych została zainicjalizowana z identyfikatorem osi, może być tylko wtedy stosowana w różnych kanałach NC, gdy oś w tych kanałach ma taki sam numer osi kanałowej.

Jeżeli tak nie jest, zmienna musi zostać załadowana na początku programu NC albo, jak w poniższym przykładzie, musi zostać zastosowana funkcja AXNAME(...) (patrz "Funkcje osi (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Strona 661)").

Kod programu	Komentarz
`DEF NCK STRING[5] ACHSE="X"`	`; Definicja w module danych`
`...`	
`N100 AX[AXNAME(ACHSE)]=111 G00`	`; Zastosowanie w programie NC`

2.1.5 Redefinicja danych systemowych, danych użytkownika i poleceń językowych NC (REDEF)

Przy pomocy polecenia `REDEF` można zmienić atrybuty danych systemowych, danych użytkownika i polecenia językowe NC. Podstawowym warunkiem redefinicji jest to, by pod względem czasu została wykonana po odpowiedniej definicji.

Przy redefinicji nie można równocześnie zmienić wielu atrybutów. Dla każdego zmienianego atrybutu musi zostać zaprogramowana oddzielna instrukcja `REDEF`.

Jeżeli zostanie zaprogramowanych wiele konkurujących zmian atrybutów, wówczas jest zawsze aktywna ostatnia zmiana.

Cofnięcie wartości atrybutów

Zmienione za pomocą `REDEF` atrybuty dla praw dostępu oraz chwili inicjalizacji mogą zostać cofnięte do ich wartości domyślnej poprzez ponowne zaprogramowanie polecenia `REDEF` poprzedzonego nazwą zmiennej bądź polecenia językowego NC.

- Prawa dostępu: Poziom dostępu 7
- Chwila inicjalizacji: brak inicjalizacji lub też zachowanie aktualnej wartości

Atrybuty redefiniowalne

Patrz "Przegląd atrybutów definiowalnych i redefiniowalnych (Strona 54)".

Lokalne zmienne użytkownika (PUD / LUD)

Dla lokalnych zmiennych użytkownika (PUD / LUD) nie wolno dokonywać żadnych redefinicji.

Składnia

```
REDEF <nazwa> <VL_Stop>
REDEF <nazwa> <jednostka_fizyczna>
REDEF <nazwa> <wartości graniczne>
REDEF <nazwa> <prawa dostępu>
REDEF <nazwa> <chwila_inicjalizacji>
REDEF <nazwa> <chwila_inicjalizacji> <wartość_inicjalizacyjna>
REDEF <nazwa> <klasa danych>
REDEF <nazwa>
```

Znaczenie

REDEF:	Rozkaz redefinicji danego atrybutu lub też cofnięcia atrybutów "Prawo dostępu" i/lub "Chwila inicjalizacji" zmiennych systemu, użytkownika oraz poleceń języka NC.	
<nazwa>:	Nazwa już zdefiniowanej zmiennej albo polecenia języka NC	
<Zatrzymanie przebiegu wyprzedzającego>:	Zatrzymanie przebiegu wyprzedzającego	
	SYNR:	Zatrzymanie przebiegu wyprzedzającego przy odczycie
	SYNW:	Zatrzymanie przebiegu wyprzedzającego przy zapisie
	SYNRW:	Zatrzymanie przebiegu wyprzedzającego przy odczycie/zapisie
<Jednostka_fizyczna>:	Jednostka fizyczna	
	PHU <jednostka>:	jednostka fizyczna
	Patrz "Atrybut: jednostka fizyczna (PHU) (Strona 45)". **Wskazówka** Nieredefiniowana przy: • Zmienne systemowe • Globalne dane użytkownika (GUD) dot. typów danych: BOOL, AXIS, STRING, FRAME	
<Wartości graniczne>:	dolna / górna wartość graniczna	
	LLI <wartość graniczna>:	Dolna wartość graniczna (lower limit)
	ULI <wartość graniczna>:	Górna wartość graniczna (upper limit)
	Patrz "Atrybut: wartości graniczne (LLI, ULI) (Strona 43)". **Wskazówka** Nieredefiniowana przy: • Zmienne systemowe • Globalne dane użytkownika (GUD) dot. typów danych: BOOL, AXIS, STRING, FRAME	

<Prawa dostępu>:	Prawa dostępu dla odczytu / zapisu przez program obróbki lub BTSS	
	APRP <poziom dostępu>:	Wykonanie: Element języka NC
	APRP <poziom dostępu>:	Odczyt: Program obróbki
	APWP <poziom dostępu>:	Zapis: Program obróbki
	APRB <poziom dostępu>:	Odczyt: BTSS
	APWB <poziom dostępu>:	Zapis: BTSS
	<poziom dostępu>:	Zakres wartości: 0 ... 7
	Patrz "Atrybut: prawa dostępu (APR, APW, APRP, APWP, APRB, APWB) (Strona 47)".	
<Chwila_inicjalizacji>:	Chwila, w której zmienna jest inicjalizowana	
	INIPO:	Power On
	INIRE:	Koniec programu głównego, NC-Reset lub Power On
	INICF:	NEWCONF lub koniec programu głównego, NC-Reset lub Power On
	PRLOC:	Koniec programu głównego, NC-Reset po lokalnej zmianie lub Power On
	Patrz "Atrybut: wartość inicjalizacyjna (Strona 40)".	
<Wartość_inicj>:	Wartość inicjalizacyjna Przy redefinicji wartości inicjalizacyjnej należy zawsze podać również chwilę inicjalizacji (patrz <chwila_inicjalizacji>. Patrz "Atrybut: wartość inicjalizacyjna (Strona 40)". Odnośnie inicjalizacji zmiennych tablicowych patrz "Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP) (Strona 55)". **Uwaga** Nieredefiniowana dla zmiennych systemowych oprócz danych ustawczych.	
<Klasa danych>:	Przyporządkowanie klasy danych (tylko SINUMERIK 828D!)	
	DCM:	Klasa danych M (= Manufacturer)
	DCI:	Klasa danych I (= Individual)
	DCU:	Klasa danych U (= User)
	Patrz "Atrybut: Klasa danych (DCM, DCI, DCU) - tylko SINUMERIK 828D (Strona 53)".	

Przykład

Redefinicja zmiennej systemowej $TC_DPCx w module danych dla producenta maszyny

Kod programu

```
%_N_MGUD_DEF                                ; Moduł GUD: Producent maszyny
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)
```

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

do N100 Prawo dostępu w celu zapisu: BTSS = poziom dostępu 2, program obróbki = poziom dostępu 3

do N200 Jednostka fizyczna [%]

do N300 dolna wartość graniczna = 0, górna wartość graniczna = 200

do N400 Zmienne tablicowe są inicjalizowane podczas PowerOn z czterema wartościami.

do N800/N900 Cofnięcie wartości atrybutów "Prawo dostępu" i/lub "Chwila inicjalizacji"

Uwaga

Zastosowanie plików ACCESS

Przy zastosowaniu plików ACCESS przedefiniowanie praw dostępu musi zostać przeniesione z _N_MGUD_DEF do _N_MACCESS_DEF.

Warunki brzegowe

Podzielność

Redefinicja odnosi się zawsze do całej zmiennej, jednoznacznie określonej przez nazwę. Nie jest możliwe np. w przypadku zmiennych tablicowych, przyporządkowanie różnych wartości atrybutów poszczególnym elementom tablicy.

2.1.6 Atrybut: wartość inicjalizacyjna

Definicja (DEF) zmiennych użytkownika

Przy definicji można dla następujących zmiennych użytkownika zadać wartość inicjalizacyjną:

- Globalne zmienne użytkownika (GUD)
- Dane użytkownika globalne dla programu (PUD)
- Lokalne zmienne użytkownika (LUD)

Redefinicja (REDEF) zmiennych systemowych i zmiennych użytkownika

Przy redefinicji można dla następujących zmiennych zadać wartość inicjalizacyjną:

- Dane systemowe
  - Dane ustawcze
- Dane użytkownika
  - Parametry R
  - Zmienna akcji synchronicznej ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD akcji synchronicznej (SYG_xy[ ], gdzie x=R, I, B, A, C, S, a y=S, M, U, 4, ..., 9)
  - Parametry EPS
  - Dane narzędzia-OEM
  - Dane magazynu-OEM
  - Globalne zmienne użytkownika (GUD)

Chwila reinicjalizacji

Przy redefinicji może zostać podana chwila, w której zmienna ma zostać zainicjalizowana, tzn. ponownie ustawiona na wartość inicjalizacyjną:

- `INIPO` (Power On)
  Zmienna jest reinicjalizowana przy PowerOn.
- `INIRE` (Reset)
  Zmienna jest reinicjalizowana przy NC-Reset, BAG-Reset, końcu programu obróbki (`M02` / `M30`) lub przy PowerOn.
- `INICF` (NEWCONF)
  Zmienna jest reinicjalizowana przy funkcji „Ustaw działanie danych maszynowych” poprzez HMI, polecenie programu obróbki `NEWCONFIG` lub przy NC-Reset, BAG-Reset, koniec programu obróbki (M02 / M30) lub Power On.
- `PRLOC` (zmiana lokalna w programie)
  Zmienna jest tylko wtedy reinicjalizowana przy NC-Reset, BAG-Reset lub końcu programu obróbki (M02 / M30), gdy została zmieniona w ramach aktualnego programu obróbki. Atrybut `PRLOC` wolno stosować tylko w związku z programowalnymi danymi ustawczymi (patrz poniższa tablica).

Tabela 2-1 Programowane dane ustawcze

Numer	Identyfikator	Polecenie G [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`
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	`FPRAON`

Numer	Identyfikator	Polecenie G [1)]
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) Przy pomocy tego polecenia G następuje dostęp do danej ustawczej		

Warunki brzegowe

Wartość inicjalizacyjna: Globalne zmienne użytkownika (GUD)

- Dla globalnej zmiennej użytkownika (GUD) o zakresie obowiązywania NC, można jako chwilę inicjalizacji zadać tylko INIPO (Power On).
- Dla globalnych zmiennych użytkownika (GUD) o zakresie obowiązywania CHAN można jako chwilę inicjalizacji oprócz INIPO (Power On) zadać również INIRE (Reset) lub INICF (NEWCONF).
- W przypadku globalnych zmiennych użytkownika (GUD) o zakresie obowiązywania CHAN i chwili inicjalizacji INIRE (Reset) lub INICF (NEWCONF), przy NC-Reset, BAG-Reset i „Ustaw działanie danych maszynowych” zmienne są ponownie inicjalizowane tylko w kanałach, w których podane zdarzenia zostały wyzwolone.

Wartość inicjalizacyjna: Typ danych FRAME

Dla zmiennych o typie danych FRAME nie wolno podawać wartości inicjalizacyjnej. Zmienne o typie danych FRAME są inicjalizowane z frame domyślnym.

Wartość inicjalizacyjna: Typ danych CHAR

Dla zmiennych o typie danych CHAR można zamiast kodu ASCII (0...255) programować również odpowiednie znaki ASCII w cudzysłowie, np. "A"

Wartość inicjalizacyjna: Typ danych STRING

W przypadku zmiennych o typie danych STRING łańcuch znaków musi zostać umieszczony w cudzysłowie np.: ...= "MASZYNA_1"

Wartość inicjalizacyjna: Typ danych AXIS

Dla zmiennych o typie danych AXIS przy rozszerzonym sposobie pisania adresów identyfikator osi musi zostać umieszczony w nawiasach, np.: ...=(X3)

Wartość inicjalizacyjna: Zmienna systemowa

Dla zmiennej systemowej nie można przez redefinicję zadać wartości inicjalizacyjnych specyficznych dla użytkownika. Wartości inicjalizacyjne zmiennych systemowych są na stałe zadane przez system. Przez redefinicję można jednak zmienić chwilę (`INIRE`, `INICF`) w której zmienna systemowa jest reinicjalizowana.

Wartość domyślna inicjalizacji: Typ danych AXIS

Dla zmiennych o typie danych `AXIS` jest domyślnie stosowana następująca wartość inicjalizacyjna:

- Dane systemowe: "pierwsza oś geometryczna"
- GUD akcji synchronicznej (określenie: SYG_A*), PUD, LUD:
  Identyfikator osi z danej maszynowej: MD20082
  $MC_AXCONF_CHANAX_DEFAULT_NAME

Wartość domyślna inicjalizacji: Dane narzędzi i magazynu

Dla danych narzędzi i magazynu wartości inicjalizacyjne mogą być zadawane przez następującą daną maszynową: MD17520 $MN_TOOL_DEFAULT_DATA_MASK

Uwaga

Synchronizacja

Synchronizacja zdarzeń, które wyzwalają reinicjalizację zmiennej globalnej z odczytem tej zmiennej w innym miejscu, pozostaje wyłącznie w zakresie odpowiedzialności użytkownika / producenta maszyny.

Patrz również

Zmienne (Strona 23)

2.1.7 Atrybut: wartości graniczne (LLI, ULI)

Górną i dolną wartość graniczną zakresu definicji można zadać tylko dla następujących typów danych:

- INT
- REAL
- CHAR

Definicja (DEF) zmiennych użytkownika: Wartości graniczne i domyślne wartości inicjalizacyjne

Jeżeli przy definicji zmiennej użytkownika w jednym z wyżej wymienionych typów danych nie zostanie bezpośrednio zdefiniowana wartość inicjalizacyjna, zmienna zostanie domyślnie ustawiona na wartość inicjalizacyjną danego typu danych:

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

Jeżeli domyślna wartość graniczna leży poza zakresem definicji ustalonym przez zaprogramowane wartości graniczne, zmienna jest inicjalizowana z wartością graniczną, która jest najbliższej domyślnej wartości inicjalizacyjnej:

- Domyślna wartość inicjalizacyjna < dolna wartość graniczna (LLI) ⇒ wartość inicjalizacyjna = dolna wartość graniczna
- Domyślna wartość inicjalizacyjna > górna wartość graniczna (ULI) ⇒ wartość inicjalizacyjna = górna wartość graniczna

Przykłady:

Kod programu	**Komentarz**

`DEF REAL GUD1`	`; Dolna wartość graniczna = granica zakresu definicji`
	`; Górna wartość graniczna = granica zakresu definicji`
	`; Nie zaprogramowano wartości inicjalizacyjnej`
	`; => Domyślna wartość inicjalizacyjna = 0.0`

`DEF REAL LLI 5.0 GUD2`	`; Dolna wartość graniczna = 5.0`
	`; Górna wartość graniczna = granica zakresu definicji`
	`; => wartość inicjalizacyjna = 5.0`

`DEF REAL ULI -5 GUD3`	`; Dolna wartość graniczna = granica zakresu definicji`
	`; Górna wartość graniczna = -5.0`
	`; => wartość inicjalizacyjna = -5.0`

Redefinicja (REDEF) zmiennych użytkownika: Wartości graniczne i aktualne wartości rzeczywiste

Jeżeli przy redefinicji wartości granicznych zmiennej użytkownika zostaną one tak zmienione, że aktualna wartość rzeczywista znajdzie się poza nowym zakresem definicji, następuje alarm i wartości graniczne nie są przejmowane.

Uwaga

Redefinicja (REDEF) zmiennych użytkownika

Przy redefinicji wartości granicznych zmiennej użytkownika należy zwracać uwagę na spójną zmianę następujących wartości:

- Wartości graniczne
- Wartość rzeczywista
- Wartość inicjalizacyjna przy redefinicji, a przy automatycznej reinicjalizacji na podstawie INIPO, INIRE lub INICF

Patrz również

Zmienne (Strona 23)

2.1.8 Atrybut: jednostka fizyczna (PHU)

Jednostka fizyczna może zostać zadana tylko dla zmiennych o następujących typach danych:

- INT
- REAL

Programowane jednostki fizyczne (PHU)

Podanie jednostki fizycznej następuje jako liczba stałoprzecinkowa: `PHU <jednostka>`

Można programować następujące jednostki fizyczne:

<jednostka>	Znaczenie	Jednostka fizyczna
0	Nie jednostka fizyczna	-
1	Pozycja liniowa lub kątowa [1)2)]	[mm], [cal], [stopień]
2	Pozycja liniowa [2)]	[mm], [cal]
3	Pozycja kątowa	[stopień]
4	Prędkość liniowa lub kątowa [1)2)]	[mm/min], [cali/min], [obr/min]
5	Prędkość liniowa [2)]	[mm/min]
6	Prędkość kątowa	[obr/min]
7	Przyspieszenie liniowe lub kątowe [1)2)]	[m/s^2], [$cali/s^2$], [obr/s^2]
8	Przyspieszenie liniowe [2)]	[m/s^2], [$cali/s^2$]
9	Przyspieszenie kątowe	[obr/s^2]
10	Przyspieszenie drugiego stopnia liniowe lub kątowe [1)2)]	[m/s^3], [$cali/s^3$], [obr/s^3]

<jednostka>	Znaczenie	Jednostka fizyczna
11	Przyspieszenie drugiego stopnia liniowe [2]	[m/s³], [cali/s³]
12	Przyspieszenie drugiego stopnia kątowe	[obr/s³]
13	Czas	[s]
14	Wzmocnienie regulatora położenia	[16.667/s]
15	Posuw na obrót [2]	[mm/obr], [cali/obr]
16	Kompensacja temperatury [1][2]	[mm], [cal]
18	Siła	[N]
19	Masa	[kg]
20	Moment bezwładności [3]	[kgm²]
21	Procent	[%]
22	Częstotliwość	[Hz]
23	Napięcie	[V]
24	Prąd	[A]
25	Temperatura	[°C]
26	Kąt	[stopień]
27	KV	[1000/min]
28	Pozycja liniowa lub kątowa [3]	[mm], [cal], [stopień]
29	Prędkość skrawania [2]	[m/min], [stóp/min]
30	Prędkość obwodowa [2]	[m/s], [stóp/s]
31	Oporność	[Ohm]
32	Indukcyjność	[mH]
33	Moment obrotowy [3]	[Nm]
34	Stała momentu obrotowego [3]	[Nm/A]
35	Wzmocnienie regulatora prądu	[V/A]
36	Wzmocnienie regulatora prędkości obrotowej [3]	[Nm/(rad*s)]
37	Prędkość obrotowa	[obr/min]
42	Moc	[kW]
43	Prąd, mały	[µA]
46	Moment obrotowy, mały [3]	[µNm]
48	Promil	-
49	-	[Hz/s]
65	Przepływ	[l/min]
66	Ciśnienie	[bar]
67	Objętość [3]	[cm³]
68	Wzmocnienie obiektowe [3]	[mm/(V*min)]
69	Wzmocnienie obiektowe regulator siły	[N/V]
155	Skok gwintu [3]	[mm/obr], [cali/obr]
156	Zmiana skoku gwintu [3]	[mm/obr / obr], [cali/obr / obr]
1) Jednostka fizyczna jest zależna od typu osi: liniowa lub obrotowa		

<jednostka>	Znaczenie	Jednostka fizyczna
2) Przełączenie systemu miar G70/G71(calowy/metryczny) Po przełączeniu systemu podstawowego (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) przy pomocy G70/G71 przy dostępie w celu zapisu/odczytu do związanych z długością zmiennych systemowych i zmiennych użytkownika **nie** następuje przeliczenie wartości (wartość rzeczywista, wartość domyślna i wartości graniczne) G700/G710 (calowy/metryczny) Po przełączeniu systemu podstawowego (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) przy pomocy G700/G710 przy dostępie w celu zapisu/odczytu do związanych z długością zmiennych systemowych i zmiennych użytkownika **następuje** przeliczenie wartości (wartość rzeczywista, wartość domyślna i wartości graniczne)		
3) Zmienna **nie** jest automatycznie przeliczana na aktualny system miar NC (calowy/metryczny). Przeliczenie leży wyłącznie w zakresie odpowiedzialności użytkownika / producenta maszyny.		

Uwaga

Przekroczenie poziomu w wyniku przeliczenia formatu

Wewnętrzny format zapisu wszystkich zmiennych użytkownika (GUD / PUD / LUD) o związanych z długością jednostkach fizycznych jest formatem metrycznym. Nadmierne stosowanie tego rodzaju zmiennych w przebiegu głównym NCK, np. w akcjach synchronicznych, może przy przełączeniu systemu miar prowadzić do przekroczenia czasu obliczeniowego poziomu interpolatora, alarm 4240.

Uwaga

Kompatybilność jednostek

Przy stosowaniu zmiennych (przyporządkowanie, porównanie itd.) nie następuje kontrola na kompatybilność jednostek. Ewentualnie wymagane przeliczenie leży wyłącznie w zakresie odpowiedzialności użytkownika / producenta maszyny.

Patrz również

Zmienne (Strona 23)

2.1.9 Atrybut: prawa dostępu (APR, APW, APRP, APWP, APRB, APWB)

Określenie

Określenie atrybutu dostępu `AP`... składa się z:

1. A: **A**ccess
2. P: **P**rotection
3. R / W: **R**ead / **W**rite
4. P / B: **P**rogram / **B**TSS

Prawo dostępu / Poziomy dostępu

Prawom dostępu odpowiadają następujące poziomy dostępu podawane przy programowaniu:

Prawo dostępu	Poziom dostępu
Hasło "System"	0
Hasło „Producent"	1
Hasło „Serwis"	2
Hasło „Użytkownik końcowy"	3
Położenie 3 przełącznika z kluczykiem	4
Położenie 2 przełącznika z kluczykiem	5
Położenie 1 przełącznika z kluczykiem	6
Położenie 0 przełącznika z kluczykiem	7

Definicja (DEF) zmiennych użytkownika

Prawa dostępu (APR... / APW...) mogą być definiowane dla następujących danych:

- Globalne dane użytkownika (GUD)

Redefinicja (REDEF) danych systemowych i danych użytkownika

Prawa dostępu (APR... / APW...) mogą być redefiniowane dla następujących danych:

- Dane systemowe
  - Dane maszynowe

    Uwaga

    Redefinicja praw odczytu danych maszynowych

    Poziom dostępu do odczytu danych maszynowych może zostać ustawiony tylko za pomocą słowa kluczowego APR zarówno dla programu obróbki, jak i BTSS.

    Słowa kluczowe APRP i APRB nie są obsługiwane przy redefinicji praw odczytu i prowadzą do alarmu 12490 „Prawo dostępu APRP/APRB <poziom dostępu> nie został ustawiony”.

  - Dane ustawcze
  - Zmienna systemowa
  - Dane procesowe
  - Dane magazynu
  - Dane narzędzi
- Dane użytkownika
  - Parametry R
  - Zmienna akcji synchronicznej ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD akcji synchronicznej (SYG_xy[ ], gdzie x=R, I, B, A, C, S, a y=S, M, U, 4, ..., 9)
  - Parametry EPS
  - Dane narzędzia-OEM
  - Dane magazynu-OEM
  - Globalne zmienne użytkownika (GUD)

    Uwaga

    Przy redefinicji prawo dostępu do zmiennej może zostać dowolnie nadane między najniższym poziomem dostępu 7 i własnym poziomem dostępu, np. 1 (producent maszyny).

Redefinicja (REDEF) poleceń języka NC

Prawo dostępu lub wykonania (APX) można redefiniować dla następujących poleceń języka NC:

- Polecenia G / Warunek drogowy
  Literatura:
  Podręcznik programowania Podstawy; Rozdział: Polecenia G / Warunek drogowy
- Funkcje wstępnie zdefiniowane
  Literatura:
  Podręcznik programowania Podstawy; Rozdział: Funkcje wstępnie zdefiniowane
- Wstępnie zdefiniowane wywołania podprogramów
  Literatura:
  Podręcznik programowania Podstawy; Rozdział: Wstępnie zdefiniowane wywołania podprogramów
- Instrukcja DO w przypadku akcji synchronicznych
- Identyfikatory programowe cykli
  Cykl musi być zapisany w katalogu cykli i zawierać instrukcję PROC.

Prawa dostępu odnośnie programów NC i cykli (APRP, APWP)

Różne prawa dostępu mają następujące działanie dla dostępu w programie NC lub cyklu:

- APRP 0 / APWP 0
  - Przy wykonywaniu programu NC musi być ustawione hasło systemowe
  - Cykl musi być zapisany w katalogu _N_CST_DIR (System)
  - Dla katalogu _N_CST_DIR prawo wykonania musi być ustawione na system w MD11160 $MN_ACCESS_EXEC_CST
- APRP 1 / APWP 1 lub APRP 2 / APWP 2
  - Przy wykonywaniu programu NC musi być ustawione hasło „producent" lub „serwis"
  - Cykl musi być zapisany w katalogu _N_CMA_DIR (producenta) lub _N_CST_DIR
  - Dla katalogów _N_CMA_DIR lub _N_CST_DIR w danych maszynowych MD11161 $MN_ACCESS_EXEC_CMA lub MD11160 $MN_ACCESS_EXEC_CST prawa wykonania muszą być ustawione co najmniej na poziomie „producent"

- APRP 3 / APWP 3
  - Przy wykonywaniu programu NC musi być ustawione hasło „użytkownik końcowy"
  - Cykl musi być zapisany w katalogu _N_CUS_DIR (użytkownik), _N_CMA_DIR lub _N_CST_DIR
  - Dla katalogów _N_CUS_DIR, _N_CMA_DIR lub _N_CST_DIR w danych maszynowych MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA lub MD11160 $MN_ACCESS_EXEC_CST prawa wykonania muszą być ustawione co najmniej na „użytkownik końcowy"
- APRP 4...7 / APWP 4...7
  - Przy wykonywaniu programu NC przełącznik z kluczykiem musi być ustawiony w położeniu 3 ... 0
  - Cykl musi być zapisany w katalogu _N_CUS_DIR, _N_CMA_DIR lub _N_CST_DIR
  - Dla katalogów _N_CUS_DIR, _N_CMA_DIR lub _N_CST_DIR w danych maszynowych MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA lub MD11160 $MN_ACCESS_EXEC_CST prawa wykonania muszą być ustawione na odpowiednie położenie przełącznika z kluczykiem

Prawa dostępu odnośnie BTSS (APRB, APWB)

Prawa dostępu (APRB, APWB) ograniczają dostęp do zmiennych systemowych i zmiennych użytkownika przez BTSS dla wszystkich komponentów systemu (HMI, PLC, komputer zewnętrzny, usługa EPS, itd.) w tym samym stopniu.

Uwaga

Prawa dostępu lokalne w HMI

Przy zmianie praw dostępu do danych systemowych należy zwrócić uwagę, by nastąpiło to zgodnie z prawami dostępu ustalonymi przez HMI.

Atrybuty dostępu APR / APW

Z powodu kompatybilności atrybuty APR i APW są domyślnie odwzorowywane na atrybuty APRP / APRB i APWP / APWB:

- APR x ⇒ APRP x APRB x
- APW y ⇒ APWP y APWB y

Prawa dostępu za pomocą plików ACCESS

Przy zastosowaniu plików ACCESS do nadawania praw dostępu, redefinicje praw dostępu dla danych systemowych, danych użytkownika i poleceń języka NC można programować tylko w tych plikach ACCESS. Wyjątek stanowią globalne dane użytkownika (GUD). Dla nich, jeżeli okaże się to konieczne, redefinicja praw dostępu musi być nadal programowana w odpowiednich plikach definicji *_DEF.

Aby zapewnić ciągłą ochronę dostępu, dane maszynowe dotyczące uprawnień do wykonywania i ochrony dostępu do odpowiednich katalogów muszą być konsekwentnie dostosowywane.

Wynika z tego następujący sposób postępowania:

1. Sporządzenie potrzebnych plików definicji:
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. Parametryzacja prawa zapisu dla plików definicji na wartość wymaganą dla redefinicji:
   - MD11170 $MN_ACCESS_WRITE_SACCESS = <poziom dostępu>
   - MD11171 $MN_ACCESS_WRITE_MACCESS = <poziom dostępu>
   - MD11172 $MN_ACCESS_WRITE_UACCESS = <poziom dostępu>
3. Dla dostępów do chronionych elementów cykli muszą zostać dopasowane prawa wykonywania i zapisu katalogów cykli _N_CST_DIR, _N_CMA_DIR i _N_CST_DIR:
   Prawa wykonania
   - MD11160 $MN_ACCESS_EXEC_CST = <poziom dostępu>
   - MD11161 $MN_ACCESS_EXEC_CMA = <poziom dostępu>
   - MD11162 $MN_ACCESS_EXEC_CUS = <poziom dostępu>

   Prawa zapisu
   - MD11165 $MN_ACCESS_WRITE_CST = <poziom dostępu>
   - MD11166 $MN_ACCESS_WRITE_CMA = <poziom dostępu>
   - MD11167 MN_ACCESS_WRITE_CUS = <poziom dostępu>

   Prawo wykonywania musi zostać ustawione na co najmniej taki sam poziom dostępu, jak najwyższy poziom dostępu zastosowanego elementu.
   Prawo zapisu musi zostać ustawione co najmniej na taki sam poziom dostępu, jak prawo wykonywania.
4. Prawa zapisu lokalnych katalogów cykli w HMI muszą zostać ustawione na taki sam poziom dostępu jak w przypadku lokalnych katalogów cykli w NC.
   Literatura:
   Podręcznik obsługi

Wywołania podprogramów w plikach ACCESS

W celu dalszej strukturyzacji ochrony przed dostępem w plikach ACCESS mogą być również wywoływane podprogramy (rozszerzenie SPF lub MPF). Podprogramy dziedziczą przy tym prawa wykonywania wywołującego pliku ACCESS.

Uwaga

W plikach ACCESS mogą być redefiniowane tylko prawa dostępu. Wszystkie inne atrybuty muszą być nadal programowane lub redefiniowane w odpowiednich plikach definicji.

Patrz również

Zmienne (Strona 23)

2.1.10 Atrybut: Klasa danych (DCM, DCI, DCU) - tylko SINUMERIK 828D

W celu uproszczenia posługiwania się danymi w ramach uruchamiania, uruchamiania seryjnego i aktualizacji maszyn i szeregów maszyn, wszystkie dane systemowe i dane użytkownika w NC są podzielone na klasy.

Klasa danych	Dane
S = system	Udostępniane przez firmę Siemens dane systemowe jak dane maszynowe i ustawcze, cykle standardowe i pomiarowe, definicje (SGUD) i makra (SMAC) itd.
M = Manufacturer (producent maszyny)	Specyficzne dla szeregu maszyn dane uruchomieniowe jak cykle producenta, definicje (MGUD) i makra (MMAC) i dane maszynowe, które ustalają zakres działania maszyny.
I = Individual (specyficzne dla maszyny)	Specyficzne dla maszyny dane uruchomieniowe jak dane kompensacyjne i przesunięcia punktu odniesienia.
U = User (użytkownik)	Dane specyficzne dla maszyny, które są wytwarzane w trakcie pracy maszyny, jak dane narzędzi, dane ustawcze, programy obróbki, cykle użytkownika, definicje (UGUD) i makra (UMAC).

Literatura:
SINUMERIK 828D Podręcznik uruchomienia Toczenie i frezowanie; punkt: "Wprowadzenie i zastosowanie klas danych"

Definicja (DEF) zmiennych użytkownika

Przez klasę danych pliku albo katalogu, w którym jest definiowana dana użytkownika, jest domyślnie ustalona klasa danej. Klasy danej nie można zmienić.

Przy definicji (`DEF`) danej użytkownika można jednak dla **wartości danej** ustalić własną klasę danych, odmienną klasy danej.

Dla klasy danych wartości danych musi obowiązywać:

Priorytet klasy wartości danych **≤** priorytet klasy danej

Przykład:

Definicja GUD, która opisuje sondę pomiarową, powinna znajdować się w klasie danych M (= Manufacturer), ponieważ jest to konieczne dla przebiegu cykli producenta. Wartość danej powinna jednak, ponieważ typ sondy pomiarowej może być różny w różnych maszynach, należeć do klasy danych I (= Individual).

MGUD.DEF (klasa danych M)

```
...
DEF CHAN DCI INT CALIPER
...
```

Redefinicja (REDEF) danych systemowych

Klasę danych systemowych można zmienić przez redefinicję (REDEF). Redefinicja musi nastąpić w pliku definicji o klasie danych S albo M.

Przy stosowaniu plików ACCESS redefinicje są dozwolone tylko w ramach plików ACCESS.

Każdorazowa klasa danych maszynowych, ustawczych i danych opcji jak też zmiennych systemowych znajduje się w:

- Podręcznik listowy Szczegółowy opis danych maszynowych, Parametry: "Klasa"
- Podręcznik lista parametrów - zmienne systemowe

2.1.11 Przegląd atrybutów definiowalnych i redefiniowalnych

Poniższe tablice pokazują w przypadku których rodzajów danych które atrybuty mogą być definiowane (DEF) i/lub redefiniowane (REDEF).

Dane systemowe

Rodzaj danych	Wartość inicjalizacyjna	Wartości graniczne	Jednostka fizyczna	Prawa dostępu	Klasa danych (tylko 828D)
Dane maszynowe	---	---	---	REDEF	REDEF
Dane ustawcze	REDEF	---	---	REDEF	---
Dane FRAME	---	---	---	REDEF	---
Dane procesowe	---	---	---	REDEF	---
Kompensacja błędu skoku śruby pociągowej (EEC)	---	---	---	REDEF	---
Kompensacja ugięcia (CEC)	---	---	---	REDEF	---
Kompensacja błędu ćwiartki koła (QEC)	---	---	---	REDEF	---
Dane magazynu	---	---	---	REDEF	---
Dane narzędzi	---	---	---	REDEF	---
Obszary ochrony	---	---	---	REDEF	---
Orientowane nośniki narzędzi	---	---	---	REDEF	---
Łańcuchy kinematyczne	---	---	---	REDEF	---
Obszary ochrony 3D	---	---	---	REDEF	---
Ograniczenie obszaru pracy	---	---	---	REDEF	---

Dane użytkownika

Rodzaj danych	Wartość inicjalizacyjna	Wartości graniczne	Jednostka fizyczna	Prawa dostępu	Klasa danych
Parametry R	REDEF	REDEF	REDEF	REDEF	---
Zmienna akcji synchronicznej ($AC_...)	REDEF	REDEF	REDEF	REDEF	---
GUD akcji synchronicznej (SYG_...)	REDEF	REDEF	REDEF	REDEF	---

Rodzaj danych	Wartość inicjalizacyjna	Wartości graniczne	Jednostka fizyczna	Prawa dostępu	Klasa danych
Parametry EPS	REDEF	REDEF	REDEF	REDEF	---
Dane narzędzia OEM	REDEF	REDEF	REDEF	REDEF	---
Dane magazynu OEM	REDEF	REDEF	REDEF	REDEF	---
Globalne zmienne użytkownika (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF	DEF / REDEF
Lokalne zmienne użytkownika (PUD / LUD)	DEF	DEF	DEF	---	---

Patrz również

Zmienne (Strona 23)

2.1.12 Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP)

Zmienna użytkownika może zostać zdefiniowana jako 1- do maksymalnie 3-wymiarowa tablica (array):

- 1-wymiarowa: `DEF <typ danych> <nazwa zmiennej>[<n>]`
- 2-wymiarowa: `DEF <typ danych> <nazwa zmiennej>[<n>,<m>]`
- 3-wymiarowa: `DEF <typ danych> <nazwa zmiennej>[<n>,<m>,<o>]`

Uwaga

Zmienne użytkownika o typie danych STRING mogą być definiowane maksymalnie jako tablica 2-wymiarowa.

Typy danych

Zmienne użytkownika mogą być definiowane jako tablice dla następujących typów danych: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Przyporządkowanie wartości do elementów tablicy

Przyporządkowania wartości do elementów tablicy mogą być dokonywane w następującym czasie:

- Przy definicji tablicy (wartości inicjalizacyjne)
- Podczas przebiegu programu

Przyporządkowanie wartości może przy tym nastąpić przez:

- bezpośrednie podanie elementu tablicy
- bezpośrednie podanie elementu tablicy jako elementu startowego i podanie listy wartości (SET)
- bezpośrednie podanie elementu tablicy jako elementu startowego i podanie wartości i częstości jej powtarzania (REP)

Uwaga

Zmiennym użytkownika o typie danych FRAME nie można przyporządkować wartości inicjalizacyjnych.

Składnia (DEF)

```
DEF <typ danych> <nazwa zmiennej>[<n>,<m>,<o>]
DEF  STRING[<długość łańcucha znaków>] <nazwa zmiennej>[<n>,<m>]
```

Składnia (DEF...=SET...)

Zastosowanie listy wartości:

- Przy definicji:
  ```
  DEF <typ danych> <nazwa zmiennej>[<n>,<m>,<o>] =
  SET(<wartość1>,<wartość2>,...)
  ```
  Równoznaczne z:
  ```
  DEF <typ danych> <nazwa zmiennej>[<n>,<m>,<o>] =
  (<wartość1>,<wartość2>,...)
  ```

  Uwaga

  Przy inicjalizacji przez listę wartości podanie SET jest opcjonalne.

- Przy przyporządkowaniu wartości:
  ```
  <nazwa zmiennej>[<n>,<m>,<o>] = SET(<wartość1>,<wartość2>,...)
  ```

Składnia (DEF...=REP...)

Zastosowanie wartości z powtórzeniem

- Przy definicji:
  ```
  DEF <typ danych> <nazwa zmiennej>[<n>,<m>,<o>] = REP(<wartość>)
  DEF <typ danych> <nazwa zmiennej>[<n>,<m>,<o>] =
  REP(<wartość>,<liczna_elementów_tablicy>)
  ```
- Przy przyporządkowaniu wartości:
  ```
  <nazwa zmiennej>[<n>,<m>,<o>]=REP(<wartość>)
  <nazwa
  zmiennej>[<n>,<m>,<o>]=REP(<wartość>,<liczba_elementów_tablicy>)
  ```

Znaczenie

DEF:	Polecenie dot. definicji zmiennych	
<typ danych>:	Typ danych zmiennej	
	Zakres wartości: • W przypadku zmiennych systemowych: BOOL, CHAR, INT, REAL, STRING, AXIS • W przypadku zmiennych GUD lub LUD: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME	
<długość łańcucha znaków>:	Maksymalna liczba znaków w przypadku typu danych STRING	
<nazwa zmiennej>:	Nazwa zmiennej	
[<n>,<m>,<o>]:	Wielkości lub indeksy tablicy	
<n>:	Wielkość lub indeks tablicy dla 1. wymiaru	
	Typ:	INT (w przypadku zmiennej systemowej również AXIS)
	Zakres wartości:	Max wielkość tablicy: 65535 Indeks tablicy: 0 ≤ n ≤ 65534
<m>:	Wielkość lub indeks tablicy dla 2. wymiaru	
	Typ:	INT (w przypadku zmiennej systemowej również AXIS)
	Zakres wartości:	Max wielkość tablicy: 65535 Indeks tablicy: 0 ≤ m ≤ 65534
<o>:	Wielkość lub indeks tablicy dla 3. wymiaru	
	Typ:	INT (w przypadku zmiennej systemowej również AXIS)
	Zakres wartości:	Max wielkość tablicy: 65535 Indeks tablicy: 0 ≤ o ≤ 65534
SET:	Przyporządkowanie wartości przez podaną listę wartości	
(<wartość1>,<wartość2>,...):	Lista wartości	
REP:	Przyporządkowanie wartości przez podaną <wartość>	

<wartość>:	Wartość, którą mają zostać zapisane elementy tablicy przy inicjalizacji z REP.
<Liczba_elementów_tablicy>:	Liczba elementów tablicy, które mają zostać zapisane podaną <wartością>. Dla pozostałych elementów tablicy obowiązuje zależnie od czasu: • Inicjalizacja przy definicji tablicy: → Pozostałe elementy tablicy są zapisywane wartością zero • Przyporządkowanie podczas przebiegu programu: → Aktualne wartości elementów tablicy pozostają bez zmian. Jeżeli parametr nie jest zaprogramowany, wszystkie elementy tablicy są zapisywane wartością <wartość>. Jeżeli parametr jest równy zeru, obowiązuje zależnie od czasu: • Inicjalizacja przy definicji tablicy: → Wszystkie elementy są wyposażane w wartość domyślną zero • Przyporządkowanie podczas przebiegu programu: → Aktualne wartości elementów tablicy pozostają bez zmian.

Indeks tablicy

Kolejność domyślnych elementów tablicy np. przy przyporządkowaniu wartości przez SET lub REP następuje przez iterację indeksów tablicy od prawej do lewej.

Przykład: Inicjalizacja tablicy 3-wymiarowej o 24 elementach:

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

Odpowiednio:

```
FOR n=0 TO 1
  FOR m=0 TO 2
```

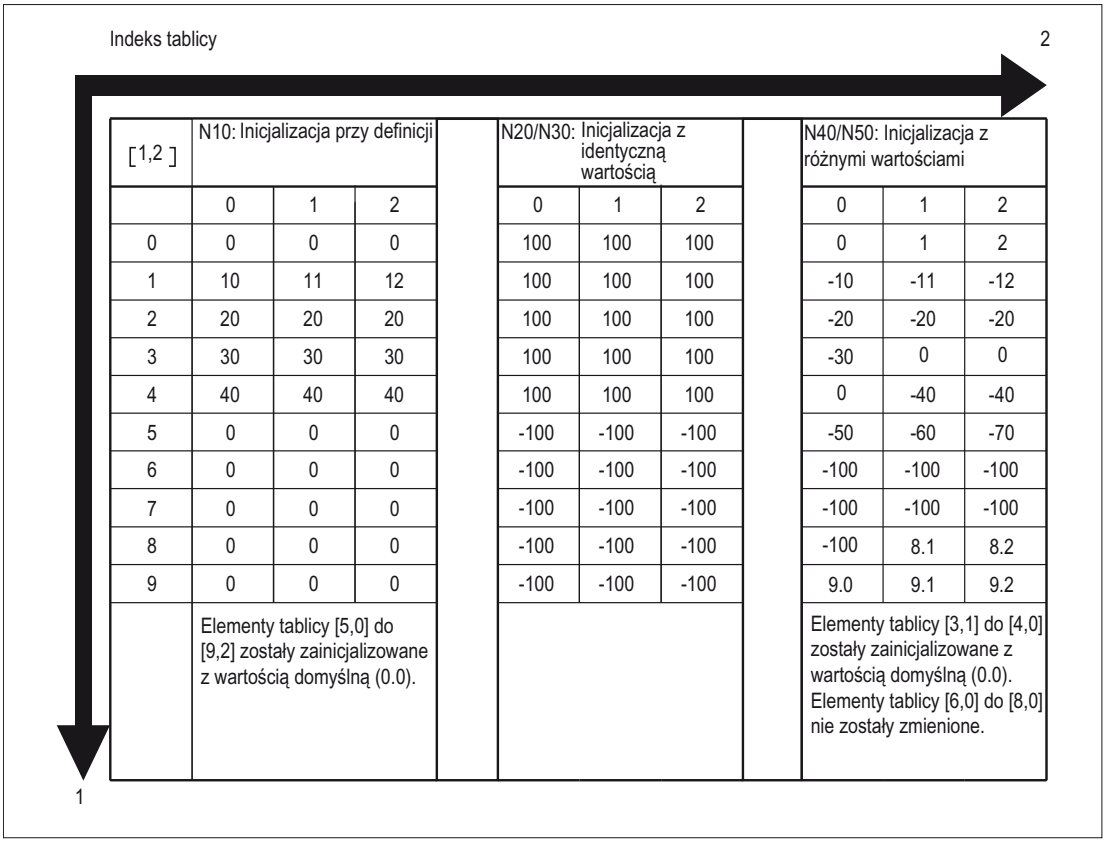
```
    FOR o=0 TO 3
      FELD[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

Przykład: Inicjalizacja kompletnych tablic zmiennych

Aktualna zajętość patrz rysunek.

Kod programu

```
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)
```



[1,2]	N10: Inicjalizacja przy definicji			N20/N30: Inicjalizacja z identyczną wartością			N40/N50: Inicjalizacja z różnymi wartościami		
	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
	Elementy tablicy [5,0] do [9,2] zostały zainicjalizowane z wartością domyślną (0.0).						Elementy tablicy [3,1] do [4,0] zostały zainicjalizowane z wartością domyślną (0.0). Elementy tablicy [6,0] do [8,0] nie zostały zmienione.		

Patrz również

Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP): Dalsze informacje (Strona 60)

Zmienne (Strona 23)

2.1.13 Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP): Dalsze informacje

Dalsze informacje (SET)

Inicjalizacja przy definicji

- Jest, rozpoczynając od 1. elementu tablicy, inicjalizowanych tyle elementów z wartościami z listy wartości, ile elementów jest zaprogramowanych na liście wartości.
- Elementy tablicy bez explicite podanych wartości na liście wartości (luki na liście) otrzymują wartość 0.
- W przypadku zmiennych o typie danych AXIS luki na liście wartości są niedopuszczalne.
- Jeżeli lista wartości zawiera więcej wartości, niż zdefiniowano elementów tablicy, zostanie wyświetlony alarm.

Przyporządkowanie wartości w przebiegu programu

Przy przyporządkowywaniu wartości w przebiegu programu obowiązują zasady opisane wyżej przy definicji. Dodatkowo są następujące możliwości:

- Jako elementy na liście wartości są dozwolone również wyrażenia.
- Przyporządkowanie wartości rozpoczyna się od zaprogramowanego indeksu tablicy. Przez to można w sposób celowy wyposażać w wartości tablice częściowe.

Przykład:

Kod programu	**Komentarz**
`DEF INT FELD[5,5]`	`; Definicja tablicy`
`FELD[0,0]=SET(1,2,3,4,5)`	`; Przyporządkowanie wartości do pierwszych 5 elementów tablicy [0,0] - [0,4]`
`FELD[0,0]=SET(1,2, , ,5)`	`; Przyporządkowanie wartości z luką do pierwszych 5 elementów tablicy [0,0] - [0,4], elementy tablicy [0,2] i [0,3] = 0`

`FELD[2,3]=SET(VARIABLE,4*5.6)`	`; Przyporządkowanie wartości ze zmienną i wyrażeniem od indeksu tablicy [2,3]:`
	`[2,3] = VARIABLE`
	`[2,4] = 4 * 5.6 = 22.4`

Dalsze informacje (REP)

Inicjalizacja przy definicji

- Wszystkie lub opcjonalnie podana liczba elementów tablicy są inicjalizowane z podaną wartością (stała).
- Zmienne o typie danych FRAME nie mogą być inicjalizowane.

Przykład:

Kod programu	Komentarz
`DEF REAL varName[10]=REP(3.5,4)`	`; Inicjalizacja definicji tablicy i elementów tablicy [0] do [3] o wartości 3,5`

Przyporządkowanie wartości w przebiegu programu

Przy przyporządkowywaniu wartości w przebiegu programu obowiązują zasady opisane wyżej przy definicji. Dodatkowo są następujące możliwości:

- Jako elementy na liście wartości są dozwolone również wyrażenia.
- Przyporządkowanie wartości rozpoczyna się od zaprogramowanego indeksu tablicy. Przez to można w sposób celowy wyposażać w wartości tablice częściowe.

Przykłady:

Kod programu	Komentarz
`DEF REAL varName[10]`	`; Definicja tablicy`
`varName[5]=REP(4.5,3)`	`; Elementy tablicy [5] do [7] = 4,5`
`R10=REP(2.4,3)`	`; Parametry R10 do R12 = 2,4`
`DEF FRAME FRM[10]`	`; Definicja tablicy`
`FRM[5]=REP(CTRANS(X,5))`	`; Elementy tablicy [5] do [9] = CTRANS(X,5)`

Patrz również

Definicja i inicjalizacja zmiennych tablicowych (DEF, SET, REP) (Strona 55)

2.1.14 Typy danych

Następujące typy danych są do dyspozycji w NC:

Typ danych	Znaczenie	Zakres wartości
INT	Wartość całkowitoliczbowa ze znakiem	-2147483648 ... +2147483647
REAL	Liczba rzeczywista (LONG REAL według IEEE)	±(~2,2*10^{-308} … ~1,8*10^{+308})
BOOL	Wartość logiczna TRUE (1) i FALSE (0)	1, 0
CHAR	Znak ASCII	Kod ASCII 0 ... 255
STRING	Łańcuch znaków o zdefiniowanej długości	Maksymalnie 200 znaków (bez znaków specjalnych)

AXIS	Identyfikator osi/wrzeciona	Identyfikator kanału
FRAME	Dane geometryczne dla statycznej transformacji współrzędnych (przesunięcie, obrót, skalowanie, lustrzane odbicie)	---

Domyślne zmiany typów danych

Następujące zmiany typów danych są możliwe i są domyślnie dokonywane przy przyporządkowaniach i przekazaniach parametrów:

Z ↓/ na →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x
x: Możliwe bez ograniczeń o: Możliwa utrata danych w wyniku przekroczenia zakresu wartości ⇒ alarm; Zaokrąglenie: wartość po przecinku ≥ 0,5 ⇒ zaokrąglenie do góry, wartość po przecinku < 0,5 ⇒ zaokrąglenie do dołu &: Wartość ≠ 0 ⇒ TRUE, wartość == 0 ⇒ FALSE			

Patrz również

Zmienne (Strona 23)

2.1.15 Sprawdzenie istnienia zmiennej (ISVAR)

Za pomocą predefiniowanej funkcji ISVAR można sprawdzić, czy zmienna systemowa/ użytkownika (np. dana maszynowa, dana ustawcza, zmienna systemowa, zmienna ogólna jak GUD) istnieje w NC.

Zmienna

Zmienne te są tworzone w następujący sposób:

Zmienna bezwymiarowa: <zmienna>
Zmienna jednowymiarowa bez indeksu tablicy:<zmienna>[ ]
Zmienna jednowymiarowa z indeksem tablicy n: <zmienna>[<n>]
Zmienna dwuwymiarowa bez indeksu tablicy: <zmienna>[,]
Zmienna dwuwymiarowa z indeksami tablicy n i m: <zmienna>[<n>,<m>]

Składnia

```
<wynik>=ISVAR(<zmienna>[<n>,<m>])
```

Znaczenie

<wynik>:	Wartość zwrotna		
	Typ danych:	BOOL	
	Zakres wartości:	1	Zmienna istnieje
		0	Zmienna nieznana
ISVAR:	Sprawdza, czy podane zmienne systemowe/użytkownika istnieją w NC.		
<zmienna>:	Nazwa zmiennej systemowej/użytkownika		
	Typ danych:	STRING	
<n>:	Indeks tablicy dla pierwszego wymiaru (**opcjonalnie**)		
	Typ danych:	INT	
<m>:	Indeks tablicy dla drugiego wymiaru (**opcjonalnie**)		
	Typ danych:	INT	

Odpowiednio do przekazywanego parametru są przeprowadzane następujące kontrole:

- Czy nazwa jest znana
- Czy zmienna jest tablicą
- Czy jest to tablica jedno- czy dwuwymiarowa
- Czy każdy indeks tablicy znajduje się w dozwolonym zakresie

Gdy wszystkie kontrole mają wynik pozytywny, zwracany jest TRUE (1).

Gdy kontrola ma wynik ujemny lub wystąpił błąd składni, zwracany jest FALSE (0).

Przykłady

```
Kod programu                          Komentarz
DEF INT VAR1
DEF BOOL IS_VAR=FALSE
N10 IS_VAR=ISVAR("VAR1")              ; IS_VAR jest w tym przypadku TRUE.
```

```
 Kod programu                         Komentarz
DEF REAL VARARRAY[10,10]
DEF BOOL IS_VAR=FALSE
N10 IS_VAR=ISVAR("VARARRAY[,]")       ; IS_VAR jest w tym przypadku TRUE, tablica
                                      jest dwuwymiarowa.
N20 IS_VAR=ISVAR("VARARRAY")          ; IS_VAR jest TRUE, zmienna istnieje.
N30 IS_VAR=ISVAR("VARARRAY[8,11]")    ; IS_VAR jest FALSE, indeks tablicy niedo-
                                      zwolony.
N40 IS_VAR=ISVAR("VARARRAY[8,8")      ; IS_VAR jest FALSE, brakuje "]" (błąd
                                      składni).
N50 IS_VAR=ISVAR("VARARRAY[,8]")      ; IS_VAR jest TRUE, indeks tablicy jest do-
                                      zwolony.
N60 IS_VAR=ISVAR("VARARRAY[8,]")      ; IS_VAR jest TRUE, indeks tablicy jest do-
                                      zwolony.
```

Kod programu	Komentarz
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"	; Przekazywany parametr jest daną maszynową, IS_VAR jest TRUE.

Kod programu	Komentarz
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("$P_EP")	; IS_VAR jest w tym przypadku TRUE.
N20 IS_VAR=ISVAR("$P_EP[X]")	; IS_VAR jest w tym przypadku TRUE.

2.1.16 Wartości atrybutów/Odczyt typu danych (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

Za pomocą predefiniowanych funkcji GETVARPHU, GETVARAP, GETVARLIM i GETVARDFT można odczytać wartości atrybutów zmiennych systemowych/użytkownika, przy pomocy GETVARTYP - typ danych zmiennej systemowej/użytkownika.

Odczyt jednostki fizycznej

Składnia:
`<wynik>=GETVARPHU(<nazwa>)`

Znaczenie:

`<wynik>`:	Wartość liczbowa jednostki fizycznej		
	Typ danych:	INT	
	Zakres wartości:	patrz tablica w "Atrybut: jednostka fizyczna (PHU) (Strona 45)"	
		W przypadku błędu:	
		- 2	Podana `<nazwa>` nie jest przyporządkowana do żadnego parametru systemowego i żadnej zmiennej użytkownika.
GETVARPHU:	Odczyt jednostki fizycznej zmiennej systemowej/użytkownika		
`<nazwa>`:	Nazwa zmiennej systemowej/użytkownika		
	Typ danych:	STRING	

Przykład:

NC zawiera następujące zmienne GUD:

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

Kod programu	Komentarz
DEF INT result=0	
result=GETVARPHU("electric")	; Odczytuje jednostkę fizyczną zmiennej GUD.
IF (result < 0) GOTOF error	

Jako wynik jest zwracana wartość 42. Odpowiada to jednostce fizycznej [kW].

Uwaga

Przy pomocy GETVARPHU można np. sprawdzić, czy dla przyporządkowania zmiennej a = b obydwie zmienne mają oczekiwane jednostki fizyczne.

Odczyt prawa dostępu

Składnia:
`<wynik>=GETVARAP(<nazwa>,<dostęp>)`

Znaczenie:

`<wynik>`:	Poziom ochrony dla podanego <dostępu>		
	Typ danych:	INT	
	Zakres wartości:	0 ... 7	Patrz "Atrybut: prawa dostępu (APR, APW, APRP, APWP, APRB, APWB) (Strona 47)".
		W przypadku błędu:	
		- 1	Możliwy tylko do zapisania (istotny tylko przy <dostęp> "`WP`" i "`WB`")
		- 2	Podana `<nazwa>` nie jest przyporządkowana do żadnego parametru systemowego i żadnej zmiennej użytkownika.
		- 3	nieprawidłowa wartość dla `<dostęp>`
`GETVARAP`:	Odczyt prawa dostępu do zmiennej systemowej/użytkownika		
`<nazwa>`:	Nazwa zmiennej systemowej/użytkownika		
	Typ danych:	STRING	
`<dostęp>`:	Rodzaj dostępu		
	Typ danych:	STRING	
	Zakres wartości:	`"RP"`	Odczyt poprzez program obróbki
		`"WP"`	Zapis poprzez program obróbki
		`"RB"`	Odczyt poprzez BTSS
		`"WB"`	Zapis poprzez BTSS

Przykład:

Kod programu	**Komentarz**
`DEF INT result=0`	
`result=GETVA-RAP("$TC_MAP8","WB")`	`; Określa ochronę dostępu dla parametru systemowego "pozycja magazynu" w odniesieniu do zapisu za pomocą BTSS.`
`IF (result < 0) GOTOF error`	

Jako wynik jest zwracana wartość 7. Odpowiada to pozycji 0 przełącznika z kluczykiem (= brak ochrony przed dostępem).

Uwaga

Przy pomocy GETVARAP można np. realizować program sprawdzania, który sprawdza prawa dostępu oczekiwane przez aplikację.

Odczyt wartości granicznych

Składnia:
`<status>=GETVARLIM(<nazwa>,<wartość graniczna>,<wynik>)`

Znaczenie:

`<status>`:	Status funkcji		
	Typ danych:	INT	
	Zakres wartości:	1	o. k.
		-1	Żadna wartość graniczna nie jest zdefiniowana (w przypadku zmiennych typu AXIS, STRING, FRAME)
		-2	Podana `<nazwa>` nie jest przyporządkowana do żadnego parametru systemowego i żadnej zmiennej użytkownika.
		-3	Nieprawidłowa wartość dla `<wartość graniczna>`
`GETVARLIM`:	Odczyt dolnej/górnej wartości granicznej zmiennej systemowej/użytkownika		
`<nazwa>`:	Nazwa zmiennej systemowej/użytkownika		
	Typ danych:	STRING	
`<wartość graniczna>`:	Podaje, która wartość graniczna ma zostać odczytana		
	Typ danych:	CHAR	
	Zakres wartości:	`"L"`:	= dolna wartość graniczna
		`"U"`:	= górna wartość graniczna
`<wynik>`:	Zwrot wartości granicznej		
	Typ danych:	VAR REAL	

Przykład:

Kod programu	**Komentarz**
`DEF INT state=0`	
`DEF REAL result=0`	
`state=GETVARLIM("$MA_MAX_AX_VELO","L",result)`	`; Określa dolną wartość graniczną dla MD32000 $MA_MAX_AX_VELO.`
`IF (result < 0) GOTOF error`	

Atrybuty/Odczyt typu danych

Składnia:
`<wynik>=GETVARDIM( <nazwa>, indeks)`

Znaczenie:

<wynik>:	Wymiary/Liczba tablic <indeks>	
	Typ danych:	INT
GETVARDIM:	Odczytuje dolną/górną wartość graniczną zmiennej systemowej/użytkownika	
<nazwa>:	Odczytuje liczbę elementów tablicy	
	Typ danych:	STRING
<indeks>:	Numer tablicy, maks. 3.	
	Typ danych:	INT

Przykład:

Kod programu	**Komentarz**
N5 DEF REAL myReal[5,4]	
N10 R1 = GETVATDIM(„myReal“,1)	R1 = 5
N15 R2 = GETVATDIM(„myReal“,2)	R2 = 4

Odczyt wartości standardowej

Składnia:

```
<status>=GETVARDFT(<nazwa>,<wynik>[,<indeks_1>,<indeks_2>,<indeks_3>
])
```

Znaczenie:

<status>:	Status funkcji		
	Typ danych:	INT	
	Zakres wartości:	1	o. k.
		-1	Wartość standardowa nie jest dostępna (np. ponieważ <wynik> ma nieprawidłowy typ dla <nazwa>)
		-2	Podana <nazwa> nie jest przyporządkowana do żadnego parametru systemowego i żadnej zmiennej użytkownika.
		-3	Nieprawidłowa wartość dla <indeks_1>, wymiar mniejszy od jedności (= nie tablica = skalar)
		-4	Nieprawidłowa wartość dla <indeks_2>
		-5	Nieprawidłowa wartość dla <indeks_3>
GETVARDFT:	Odczyt wartości standardowej zmiennej systemowej/użytkownika		
<nazwa>:	Nazwa zmiennej systemowej/użytkownika		
	Typ danych:	STRING	

<wynik>:	Zwrot wartości standardowej	
	Typ danych:	VAR REAL (przy odczycie wartości standardowej zmiennych typów INT, REAL, BOOL, AXIS)
		VAR STRING (przy odczycie wartości standardowej zmiennych typów STRING i CHAR)
		VAR FRAME (przy odczycie wartości standardowej zmiennych typu FRAME)
<indeks_1>:	Indeks dla pierwszego wymiaru (opcjonalnie)	
	Typ danych:	INT
	Nie zaprogramowanie oznacza = 0	
<indeks_2>:	Indeks dla drugiego wymiaru (opcjonalnie)	
	Typ danych:	INT
	Nie zaprogramowanie oznacza = 0	
<indeks_3>:	Indeks dla trzeciego wymiaru (opcjonalnie)	
	Typ danych:	INT
	Nie zaprogramowanie oznacza = 0	

Przykład:

```
Kod programu                                    Komentarz
DEF INT state=0
DEF REAL resultR=0                              ; Zmienna dla ujęcia wartości standardo-
                                                wych typów INT, REAL, BOOL, AXIS.
DEF FRAME resultF=0                             ; Zmienna dla ujęcia wartości standardo-
                                                wych typu FRAME


IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4)
GOTOF error
state=GETVARDFT("$MA_MAX_AX_VELO",              ; Określa wartość standardową
resultR, AXTOINT(X))                            osi "X".
IF (resultR < 0) GOTOF error


IF (GETVARTYP("$TC_TP8") <> 3) GOTOF error
state=GETVARDFT("$TC_TP8", resultR)


IF (GETVARTYP("$P_UBFR")  <> 7) GOTOF error
state=GETVARDFT("$P_UBFR", resultF )
```

Odczyt typu danych

Składnia:

```
<wynik>=GETVARTYP(<nazwa>)
```

Znaczenie:

<wynik>:	Typ danych podanej zmiennej systemowej/użytkownika		
	Typ danych:	INT	
	Zakres wartości:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		W przypadku błędu:	
		< 0	Podana <nazwa> nie jest przyporządkowana do żadnego parametru systemowego i żadnej zmiennej użytkownika.
GETVARTYP:	Odczyt typu danych zmiennej systemowej/użytkownika		
<nazwa>:	Nazwa zmiennej systemowej/użytkownika		
	Typ danych:	STRING	

Przykład:

Kod programu	**Komentarz**
`DEF INT result=0`	
`DEF STRING name="R"`	
`result=GETVARTYP(name)`	`; Określa typ parametru R.`
`IF (result < 0) GOTOF error`	

Jako wynik jest zwracana wartość 4. Odpowiada to typowi danych REAL.

2.2 Programowanie pośrednie

2.2.1 Pośrednie programowanie adresów

Przy pośrednim programowaniu adresów adres rozszerzony (<indeks>) jest zastępowany przez zmienną odpowiedniego typu.

Uwaga

Programowanie pośrednie adresów jest niemożliwe przy:

- N (numer bloku)
- L (podprogram)
- Adresy ustawiane
  (np. X[1] zamiast X1 jest niedopuszczalne)

Składnia

```
<ADRES>[<indeks>]
```

Znaczenie

<ADRES>[...]:	Adres stały z rozszerzeniem (indeks)
<indeks>:	Zmienna np. dla numeru wrzeciona, osi, ...

Przykłady

Przykład 1: Pośrednie programowanie numeru wrzeciona

Programowanie bezpośrednie:

Kod programu	Komentarz
S1=300	; Prędkość obrotowa 300 obr/min dla wrzeciona o numerze 1.

Programowanie pośrednie:

Kod programu	Komentarz
DEF INT SPINU=1	; Definicja zmiennej typu INT i przyporządkowanie wartości.
S[SPINU]=300	; Prędkość obrotowa 300 obr/min dla wrzeciona, którego numer jest zapisany w zmiennej SPINU (w tym przykładzie wrzeciono o numerze 1).

Przykład 2: Pośrednie programowanie jednej osi

Programowanie bezpośrednie:

Kod programu	Komentarz
FA[U]=300	; Posuw 300 dla osi "U".

Programowanie pośrednie:

Kod programu	Komentarz
DEF AXIS AXVAR2=U	; Definicja jednej zmiennej typu AXIS i przyporządkowanie wartości.
FA[AXVAR2]=300	; Posuw 300 dla osi, której nazwa adresowa jest zapisana w zmiennej o nazwie AXVAR2.

Przykład 3: Pośrednie programowanie jednej osi

Programowanie bezpośrednie:

Kod programu	Komentarz
$AA_MM[X]	; Odczyt wartości zmierzonej sondą pomiarową (MKS) osi "X".

Programowanie pośrednie:

Kod programu	Komentarz
DEF AXIS AXVAR3=X	; Definicja jednej zmiennej typu AXIS i przyporządkowanie wartości.
$AA_MM[AXVAR3]	; Odczyt wartości zmierzonej przez sondą pomiarową (MKS) dla osi, której nazwa jest zapisana w zmiennej AXVAR3.

Przykład 4: Pośrednie programowanie jednej osi

Programowanie bezpośrednie:

Kod programu
X1=100 X2=200

Programowanie pośrednie:

Kod programu	Komentarz
DEF AXIS AXVAR1 AXVAR2	; Definicja dwóch zmiennych typu AXIS
AXVAR1=(X1) AXVAR2=(X2)	; Przyporządkowanie nazw osi.
AX[AXVAR1]=100 AX[AXVAR2]=200	; Ruch w osiach, których nazwy adresowe są zapisane w zmiennych o nazwach AXVAR1 i AXVAR2.

Przykład 5: Pośrednie programowanie jednej osi

Programowanie bezpośrednie:

Kod programu
G2 X100 I20

Programowanie pośrednie:

Kod programu	Komentarz
DEF AXIS AXVAR1=X	; Definicja jednej zmiennej typu AXIS i przyporządkowanie wartości.
G2 X100 IP[AXVAR1]=20	; Programowanie pośrednie podania punktu środkowego dla osi, której nazwa adresowa jest zapisana w zmiennej o nazwie AXVAR1

Przykład 6: Programowanie pośrednie elementów tablicy

Programowanie bezpośrednie:

Kod programu	Komentarz
`DEF INT FELD1[4,5]`	`; Definicja pola 1.`

Programowanie pośrednie:

Kod programu	Komentarz
`DEFINE DIM1 AS 4`	`; W przypadku wymiarów pól ich wielkości muszą zostać podane jako stałe wartości.`
`DEFINE DIM2 AS 5`	
`DEF INT FELD[DIM1,DIM2]`	
`FELD[DIM1-1,DIM2-1]=5`	

Przykład 7: Pośrednie wywołanie podprogramu

Kod programu	Komentarz
`CALL "L" << R10`	`; Wywołanie programu, którego numer znajduje się w R10 (powiązanie łańcuchów znaków).`

2.2.2 Pośrednie programowanie poleceń G

Programowanie pośrednie poleceń G umożliwia efektywne programowanie cykli.

Składnia

```
G[<grupa>]=<numer>
```

Znaczenie

`G[...]`:	Polecenie G z rozszerzeniem (indeks)	
`<grupa>`:	Parametr indeksowy: Grupa G	
	Typ:	INT
`<numer>`:	Zmienna dla numeru polecenia G	
	Typ:	INT lub REAL

Uwaga

Z reguły można pośrednio programować tylko w poleceniach G nieokreślających składni.

Z poleceń G określających składnię są możliwe tylko te z grupy nr 1 funkcji G.
Określające składnię polecenia G grup nr 2, 3 i 4 funkcji G nie są możliwe.

Uwaga

W pośrednim programowaniu poleceń G nie są dozwolone żadne funkcje arytmetyczne. Konieczne obliczenie numeru polecenia G musi nastąpić w oddzielnym wierszu programu obróbki przed pośrednim programowaniem polecenia G.

Przykłady

Przykład 1: Ustawiane przesunięcie punktu zerowego (grupa nr 8 funkcji G)

Kod programu	Komentarz
N1010 DEF INT INT_VAR	
N1020 INT_VAR=2	
...	
N1090 G[8]=INT_VAR G1 X0 Y0	; G54
N1100 INT_VAR=INT_VAR+1	; Obliczenie polecenia G
N1110 G[8]=INT_VAR G1 X0 Y0	; G55

Przykład 2: Wybór płaszczyzny (grupa nr 6 funkcji G)

Kod programu	Komentarz
N2010 R10=$P_GG[6]	; Odczyt aktywnego polecenia G grupy nr 6
...	
N2090 G[6]=R10	

Literatura

Informacje dot. grup funkcji G patrz:
Podręcznik programowania Podstawy; rozdział "Grupy funkcji G"

2.2.3 Programowanie pośrednie atrybutów pozycji (GP)

Atrybuty pozycji, jak np. przyrostowe albo absolutne programowanie pozycji osi, mogą być programowane w połączeniu ze słowem kluczowym `GP` pośrednio jako zmienne.

Zastosowanie

Programowanie pośrednie atrybutów pozycji znajduje zastosowanie w **cyklach podstawienia**, ponieważ ma tu miejsce następująca zaleta w stosunku do programowania atrybutów pozycji jako słowo kluczowe (np. `IC`, `AC`, ...):

Dzięki programowaniu pośredniemu jako zmienne **nie** jest potrzebna instrukcja `CASE`, która rozgałęzia na wszystkie możliwe atrybuty pozycji.

Składnia

```
<POLECENIE POZYCJONOWANIA>[<oś/wrzeciono>]=
GP(<pozycja>,<atrybut pozycji)
<oś/wrzeciono>=GP(<pozycja>,<atrybut pozycji)
```

Znaczenie

`<POLECENIE POZYCJONOWANIA>[]`:	Następujące polecenia pozycjonowania mogą być programowane razem ze słowem kluczowym GP:
	`POS`, `POSA`, `SPOS`, `SPOSA`
	Poza tym możliwe:
	• Wszystkie występujące w kanale identyfikatory osi/wrzeciona
	`<oś/wrzeciono>`
	• Zmienny identyfikator osi/wrzeciona `AX`

`<oś/wrzeciono>`:	Oś/wrzeciono, które ma być pozycjonowane
`GP()`:	Słowo kluczowe do pozycjonowania
`<pozycja>`:	Parametr 1 Pozycja osi/wrzeciona jako stała lub zmienna
`<atrybut pozycji>`:	Parametr 2 Atrybut pozycji (np. tryb ruchu do pozycji) jako zmienna (np. $P_SUB_SPOSMODE) lub jako słowo kluczowe (IC, AC, ...)

Wartości dawane przez zmienne mają następujące znaczenie:

Wartość	Znaczenie	Dopuszczalny przy:
0	Bez zmiany atrybutu pozycji	
1	AC	`POS`, `POSA`, `SPOS`, `SPOSA`, `AX`, Adres osi
2	IC	`POS`, `POSA`, `SPOS`, `SPOSA`, `AX`, Adres osi
3	DC	`POS`, `POSA`, `SPOS`, `SPOSA`, `AX`, Adres osi
4	ACP	`POS`, `POSA`, `SPOS`, `SPOSA`, `AX`, Adres osi
5	ACN	`POS`, `POSA`, `SPOS`, `SPOSA`, `AX`, Adres osi
6	OC	-
7	PC	-
8	DAC	`POS`, `POSA`, `AX`, Adres osi
9	DIC	`POS`, `POSA`, `AX`, Adres osi
10	RAC	`POS`, `POSA`, `AX`, Adres osi
11	RIC	`POS`, `POSA`, `AX`, Adres osi
12	CAC	`POS`, `POSA`
13	CIC	`POS`, `POSA`
14	CDC	`POS`, `POSA`
15	CACP	`POS`, `POSA`
16	CACN	`POS`, `POSA`

Przykład

Przy aktywnym sprzężeniu wrzeciona synchronicznego między wrzecionem wiodącym S1 i wrzecionem nadążnym S2 jest przez polecenie `SPOS` w programie głównym wywoływany następujący cykl podstawienia do pozycjonowania wrzecion.

Pozycjonowanie następuje poprzez instrukcję w `N2230`:
`SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE) SPOS[2]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)`

Pozycja, do której ma nastąpić ruch, jest odczytywana ze zmiennej systemowej $P_SUB_SPOSIT, tryb ruchu - ze zmiennej systemowej $P_SUB_SPOSMODE.

```
N1000 PROC LANG_SUB DISPLOF SBLOF
...
N2100 IF($P_SUB_AXFCT==2)
N2110                                   ; Zastąpienie polecenia SPOS / SPOSA /
                                        M19 przy aktywnym sprzężeniu wrzecion syn-
                                        chronicznych
N2185 DELAYFSTON                        ; Początek zakresu stopp-delay
N2190 COUPOF(S2,S1)                     ; Wyłączenie aktywności sprzężenia wrze-
                                        cion synchronicznych
N2200                                   ; Pozycjonowanie wrzeciona wiodącego i
                                        nadążnego
N2210 IF($P_SUB_SPOS==TRUE) OR
($P_SUB_SPOSA==TRUE)
N2220                                   ; Pozycjonowanie wrzeciona przy pomocy
                                        SPOS:
N2230    SPOS[1]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
         SPOS[2]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
N2250 ELSE
N2260                                   ; Pozycjonowanie wrzeciona przy pomocy
                                        M19:
N2270 M1=19 M2=19                       ; Pozycjonowanie wrzeciona wiodącego i
                                        nadążnego
N2280 ENDIF
N2285 DELAYFSTOF                        ; Koniec zakresu stopp-delay
N2290 COUPON(S2,S1)                     ; Uaktywnienie sprzężenia wrzeciona syn-
                                        chronicznego
N2410 ELSE
N2420                                   ; Odpytanie na dalsze zastąpienia
...
N3300 ENDIF
...
N9999 RET
```

Warunki brzegowe

- W akcjach synchronicznych pośrednie programowanie atrybutów pozycji jest niemożliwe.

Literatura

Podręcznik działania Funkcje podstawowe; BAG, Kanał, Praca programowa, Zachowanie się przy zresetowaniu (K1),
punkt: Zastępowanie funkcji NC przez podprogramy

2.2.4 Pośrednie programowanie wierszy programu obróbki (EXECSTRING)

Przy pomocy polecenia programu obróbki `EXECSTRING` jest możliwe wykonanie przedtem utworzonej zmiennej string, jako wiersza programu obróbki.

Składnia

`EXECSTRING` jest programowany we własnym wierszu programu obróbki:
`EXECSTRING (<zmienna string>)`

Znaczenie

`EXECSTRING`:	Polecenie do wykonania zmiennej string, jako wiersza programu obróbki
`<zmienna string>`:	Zmienna typu STRING, która zawiera będący właściwie do wykonania wiersz programu obróbki

Uwaga

Przy pomocy `EXECSTRING` mogą za wyjątkiem struktur kontrolnych (Strona 119) być osadzane wszystkie układy, które mogą być programowane w **części programowej programu obróbki**. Wykluczone są przez to instrukcje `PROC` i `DEF`, jak też generalnie stosowanie w plikach INI i DEF.

Przykład

```
Kod programu                          Komentarz
N100 DEF STRING[100] MY_BLOCK         ; Definicja zmiennej string do ujęcia będącego do
                                      wykonania wiersza programu obróbki.
N110 DEF STRING[10] MFCT1="M7"
...
N200 EXECSTRING(MFCT1 <<              ; Wykonanie wiersza programu obróbki "M7 M4711".
"M4711")
...
N300 R10=1
N310 MY_BLOCK="M3"
```

Kod programu	**Komentarz**
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	; Wykonanie wiersza programu obróbki "M3 M7".

2.3 Funkcje arytmetyczne

Operator / funkcja obliczeniowa	Znaczenie
+	Dodawanie
–	Odejmowanie
*	Mnożenie
/ [1]	Dzielenie [1]
DIV [1]	Dzielenie całkowitoliczbowe [1]
MOD [1]	Dzielenie modulo (daje resztę dzielenia całkowitoliczbowego [1]
:	Operator powiązania w przypadku zmiennych FRAME
SIN()	Sinus
COS()	Cosinus
TAN()	Tangens
ASIN()	Arcus sinus
ACOS()	Arcus cosinus
ATAN2(,) [1]	Arcus tangens2 [1]
SQRT()	Pierwiastek kwadratowy
ABS()	Wartość bezwzględna
POT()	2. potęga (kwadrat)
TRUNC()	Część całkowitoliczbowa Dokładności w przypadku poleceń porównań, ustawiane przy pomocy TRUNC (patrz "Korekcja dokładności w przypadku błędów porównania (TRUNC) (Strona 83)")
ROUND()	Zaokrąglenie do liczby całkowitej
LN()	Logarytm naturalny
EXP()	Funkcja wykładnicza
MINVAL()	Mniejsza wartość z dwóch zmiennych (patrz "Minimum, maksimum i zakres zmiennych (MINVAL, MAXVAL i BOUND) (Strona 85)")
MAXVAL()	Większa wartość z dwóch zmiennych (patrz "Minimum, maksimum i zakres zmiennych (MINVAL, MAXVAL i BOUND) (Strona 85)")
BOUND()	Wartość zmiennej, która leży w zdefiniowanym zakresie (patrz "Minimum, maksimum i zakres zmiennych (MINVAL, MAXVAL i BOUND) (Strona 85)")
CTRANS()	Przesunięcie
CROT()	Obrót
CSCALE()	Zmiana skali
CMIRROR()	Lustrzane odbicie
1) Patrz punkt "Przykłady"	

Programowanie

W przypadku funkcji obliczeniowych obowiązuje zwykły matematyczny sposób zapisu. Priorytety wykonywania są ustawiane przez nawiasy okrągłe. Dla funkcji trygonometrycznych i ich funkcji odwrotnych obowiązuje podawanie w stopniach (kąt prosty = 90°).

Przykłady

Dzielenie: /

(typ REAL) = (typ INT albo typ REAL) / (typ INT albo typ REAL);

Przykład: 3 / 4 = 0.75

Dzielenie całkowitoliczbowe: DIV

(typ INT) = (typ INT albo REAL) / (typ INT albo REAL);

Przykład: 7 DIV 4.1 = 1

Dzielenie modulo (daje resztę dzielenia całkowitoliczbowego): MOD

(typ REAL) = (typ INT albo REAL) MOD (typ INT albo REAL);

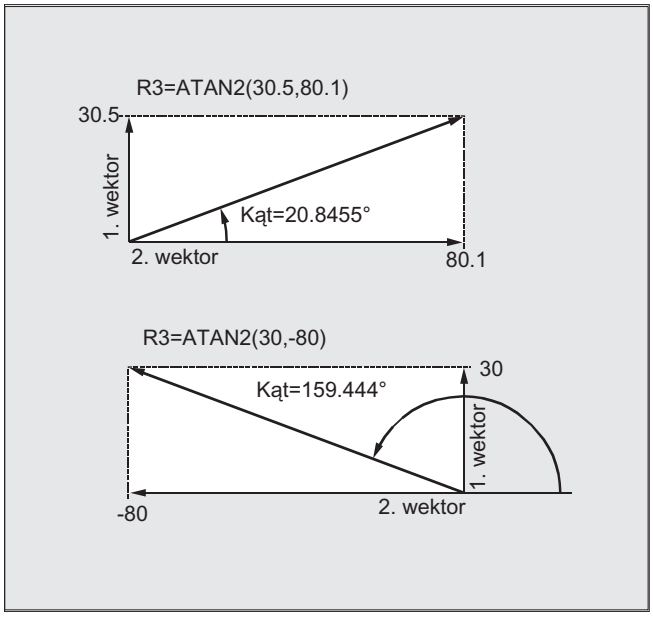
Przykład: 7 MOD 4.1 = 2.9

Arcus tangens2 ATAN2

Z dwóch prostopadłych do siebie wektorów funkcja obliczeniowa ATAN2 oblicza kąt wektora sumarycznego.

Wynik leży w zakresie czterech ćwiartek koła (-180° < 0 < +180°).

Bazą dla odniesienia kąta jest zawsze 2. wartość w kierunku dodatnim.



Przykłady programowania

Kod programu	**Komentarz**
R1=R1+1	; Nowe R1 = stare R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; Mnożenie i dzielenie przez dodawaniem i odejmowaniem.
R14=(R1+R2)*R3	; Wyrażenia w nawiasach są obliczane najpierw.
R15=SQRT(POT(R1)+POT(R2))	Najpierw następuje obliczenie w nawiasach wewnętrznych: R15 = pierwiastek kwadratowy((R1^2 + R2^2))
RESFRAME=FRAME1:FRAME2	; Powiązanie FRAME operatorem powiązania
FRAME3=CTRANS(…):CROT(…)	Przyporządkowanie wartości do składowej frame

2.4 Operacje porównania i operacje logiczne

Operacje porównania mogą np. być używane do formułowania warunku skoku. Porównywane mogą być przy tym również skomplikowane wyrażenia.

Operacje porównania można stosować do zmiennych typu `CHAR, INT, REAL` i `BOOL` . W przypadku typu `CHAR` jest porównywana wartość kodowa.
W przypadku typów `STRING`, `AXIS` i `FRAME` są możliwe == i <>, które dla operacji typu `STRING` mogą być stosowane również w akcjach synchronicznych.

Wynik operacji porównania jest zawsze typu `BOOL`.

Operatory logiczne służą do powiązania wartości logicznych.

Operacje logiczne dają się stosować tylko do zmiennych typu `BOOL`. Przez wewnętrzną konwersję typu dają się one stosować również do typów danych `CHAR, INT` i `REAL`.

W przypadku operacji logicznych (boolowskich) obowiązuje dla typów danych `BOOL`, `CHAR`, `INT` i `REAL`:

- 0 odpowiada: FALSE
- Nierówne 0 odpowiada: TRUE

Bitowe operatory logiczne

Na zmiennych typu `CHAR` i `INT` mogą również pojedynczymi bitami być wykonywane operacje logiczne. Ewentualnie konwersja typu następuje automatycznie.

Programowanie

Operator porównania	Znaczenie
==	Równe
<>	Nierówne
>	Większe niż
<	Mniejsze niż
>=	Większe lub równe
<=	Mniejsze lub równe

Operator logiczny	Znaczenie
AND	I
OR	LUB
NOT	Negacja
XOR	ALBO

Bitowy operator logiczny	Znaczenie
B_AND	Bitowe I
B_OR	Bitowe LUB
B_NOT	Bitowa negacja
B_XOR	Bitowe ALBO

Uwaga

W wyrażeniach arytmetycznych można przy pomocy nawiasów okrągłych ustalić kolejność wykonywania wszystkich operatorów, a przez to czynić odstępstwa od normalnych zasad pierwszeństwa.

Uwaga

Między BOOLOWSKIMI argumentami i operatorami muszą być pisane spacje.

Uwaga

Operator `B_NOT` odnosi się do tylko jednego argumentu. Znajduje się on za operatorem.

Przykłady

Przykład 1: Operatory porównania

```
IF R10>=100 GOTOF CEL
```

lub

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

Wynik porównania `R10>=100` jest najpierw poddawany pośredniemu zapisaniu w `R11`.

Przykład 2: Operatory logiczne

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

lub

```
IF NOT R10 GOTOB START
```

`NOT` odnosi się tylko do jednego argumentu.

Przykład 3: Bitowe operatory logiczne

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

2.5 Korekcja dokładności w przypadku błędów porównania (TRUNC)

Polecenie TRUNC odcina argument pomnożony przez współczynnik dokładności.

Ustawiana dokładność w przypadku błędów porównania

Dane programu obróbki typu REAL są wewnętrznie przedstawiane w formacie IEEE przy pomocy 64 bitów. Ze względu na tę formę przedstawienia liczby dziesiętne mogą być odwzorowywane niedokładnie, co przy porównywaniu z idealnie obliczonymi wartościami może prowadzić do nieoczekiwanych wyników.

Równość względna

Aby niedokładności wywołane przez formę przedstawienia nie zniekształcały przebiegu programu, w przypadku poleceń porównania jest sprawdzana nie równość absolutna lecz równość względna.

Składnia

Korekcja dokładności w przypadku błędów porównania

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

Znaczenie

TRUNC:	Odcięcie miejsc po przecinku

Uwzględniona równość względna 10^{-12} przy

- Równość: (==)
- Równość: (<>)
- Większe lub równe: (>=)
- Mniejsze lub równe: (<=)
- Większe niż/mniejsze niż: (><) z równością absolutną
- Większe od: (>)
- Mniejsze od: (<)

Kompatybilność

Z powodów kompatybilności można wyłączyć aktywność kontroli na równość względną przy (>) i (<) przez ustawienie danej maszynowej MD10280 $MN_ PROG_FUNCTION_MASK Bit0 = 1.

Uwaga

Porównania z danymi typu REAL są z wymienionych powodów generalnie obciążone pewną niedokładnością. Przy nie akceptowalnych odchyleniach konieczne jest przejście na rachunek INTEGER przez mnożenie argumentów przez współczynnik dokładności, a następnie obcinanie przy pomocy TRUNC.

Akcje synchroniczne

Opisane zachowanie się poleceń porównania obowiązuje również w przypadku akcji synchronicznych.

Przykłady

Przykład 1: Rozważania dot. dokładności

Kod programu	**Komentarz**
`N40 R1=61.01 R2=61.02 R3=0.01`	`; Przyporządkowanie wartości początkowych`
`N41 IF ABS(R2-R1) > R3 GOTOF BLAD`	`; Skok byłby dotychczas wykonany`
`N42 M30`	`; Koniec programu`
`N43 BŁĄD: SETAL(66000)`	
`R1=61.01 R2=61.02 R3=0.01`	`; Przyporządkowanie wartości początkowych`
`R11=TRUNC(R1*1000) R12=TRUNC(R2*1000) R13=TRUNC(R3*1000)`	`; Korekcja dokładności`
`IF ABS(R12-R11) > R13 GOTOF BLAD`	`; Skok nie zostanie już wykonany`
`M30`	`; Koniec programu`
`BŁĄD: SETAL(66000)`	

Przykład 2: Utworzenie i ewaluacja ilorazu obydwu argumentów

Kod programu	**Komentarz**
`R1=61.01 R2=61.02 R3=0.01`	`; Przyporządkowanie wartości początkowych`
`IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF BLAD`	`; Skok nie zostanie wykonany`
`M30`	`; Koniec programu`
`BŁĄD: SETAL(66000)`	

2.6 Minimum, maksimum i zakres zmiennych (MINVAL, MAXVAL i BOUND)

Przy pomocy poleceń MINVAL i MAXVAL można porównać ze sobą wartości dwóch zmiennych. Jako wynik jest zwracana mniejsza (w przypadku MINVAL) lub większa wartość (w przypadku MAXVAL).

Przy pomocy polecenia BOUND można sprawdzić, czy wartość sprawdzanej zmiennej leży w zdefiniowanym zakresie.

Składnia

```
<mniejsza wartość>=MINVAL(<zmienna1>,<zmienna2>)
<większa wartość>=MAXVAL(<zmienna1>,<zmienna2>)
<zwracana wartość>=<BOUND>(<minimum>,<maksimum>,<sprawdzana
zmienna>)
```

Znaczenie

MINVAL:	Określa **mniejszą** wartość z dwóch zmiennych (<zmienna1>, <zmienna2>)
<mniejsza wartość>:	Zmienna wynikowa dla polecenia MINVAL Jest ustawiana na mniejszą wartość zmiennej.
MAXVAL:	Określa **większą** wartość z dwóch zmiennych (<zmienna1>, <zmienna2>)
<większa wartość>:	Zmienna wynikowa dla polecenia MAXVAL Jest ustawiana na większą wartość zmiennej.
BOUND:	Sprawdza, czy zmienna (<zmienna sprawdzana>) leży w zdefiniowanym zakresie wartości.
<minimum>:	Zmienna, która definiuje wartość minimalną zakresu
<maksimum>:	Zmienna, która definiuje wartość maksymalną zakresu
<zwracana wartość>:	Zmienna wynikowa dla polecenia BOUND Jeżeli wartość sprawdzanej zmiennej leży w zdefiniowanym zakresie wartości, wówczas zmienna wynikowa jest ustawiana na wartość tej zmiennej. Jeżeli wartość sprawdzanej zmiennej jest większa, niż wartość maksymalna, wówczas zmienna wynikowa jest ustawiana na maksymalną wartość zdefiniowanego zakresu. Jeżeli wartość sprawdzanej zmiennej jest mniejsza, niż wartość minimalna, wówczas zmienna wynikowa jest ustawiana na minimalną wartość zdefiniowanego zakresu.

Uwaga

MINVAL, MAXVAL i BOUND można programować również w akcjach synchronicznych.

Uwaga

Zachowanie się w przypadku równości

W przypadku równości jest przy `MINVAL/MAXVAL` dawana jest równa wartość. W przypadku `BOUND` jest zwracana wartość sprawdzanej zmiennej.

Przykład

Kod programu	**Komentarz**
`DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4,`	`rVar5, rValMin, rValMax, rRetVar`
`rValMin=MINVAL(rVar1,rVar2)`	`; rValMin jest ustawiana na wartość 10.5.`
`rValMax=MAXVAL(rVar1,rVar2)`	`; rValMax jest ustawiana na wartość 33.7.`
`rVar3=19.7`	
`rRetVar=BOUND(rVar1,rVar2,rVar3)`	`; rVar3 leży w ramach granic, rRetVar jest ustawiana na 19.7.`
`rVar3=1.8`	
`rRetVar=BOUND(rVar1,rVar2,rVar3)`	`; rVar3 leży poniżej granicy minimum, rRetVar jest ustawiana na 10,5.`
`rVar3=45.2`	
`rRetVar=BOUND(rVar1,rVar2,rVar3)`	`; rVar3 leży powyżej granicy maksimum, rRetVar jest ustawiana na 33,7.`

2.7 Priorytet operacji

Do każdego operatora jest przyporządkowany priorytet. Prze ewaluacji wyrażenia są zawsze najpierw stosowane operatory o wyższym priorytecie. W przypadku operatorów o takiej samej randze ewaluacja następuje od lewej do prawej.

W wyrażeniach arytmetycznych można przy pomocy nawiasów okrągłych ustalić kolejność wykonywania wszystkich operatorów, a przez to czynić odstępstwa od normalnych zasad pierwszeństwa.

Kolejność operatorów

Od najwyższego do najniższego priorytetu

1.	NOT, B_NOT	Negacja, negacja bitowa
2.	*, /, DIV, MOD	Mnożenie, dzielenie
3.	+, –	Dodawanie, odejmowanie
4.	B_AND	Bitowe I
5.	B_XOR	Bitowe ALBO
6.	B_OR	Bitowe LUB
7.	AND	I
8.	XOR	ALBO
9.	OR	LUB
10.	<<	Powiązanie łańcuchów znaków, typ wyniku STRING
11.	==, <>, >, <, >=, <=	Operatory porównania

Uwaga

Operator powiązania ":" dla Frame nie może występować w jednym wyrażeniu z innymi operatorami. Dlatego dla tego operatora nie wymaga się ustalenia priorytetu.

Przykład instrukcji If

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

2.8 Możliwe konwersje typu

Funkcja

Konwersja typu przy przyporządkowaniu

Stała wartość liczbowa, zmienna albo wyrażenie, które jest przyporządkowywane do zmiennej, musi tolerować się z typem tej zmiennej. Jeżeli tak jest, wówczas przy przyporządkowywaniu typ jest automatycznie zmieniany.

Możliwe konwersje typu

na	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
z							
REAL	tak	tak*	tak[1])	tak*	–	–	–
INT	tak	tak	tak[1])	tak[2])	–	–	–
BOOL	tak	tak	tak	tak	tak	–	–
CHAR	tak	tak	tak[1])	tak	tak	–	–
STRING	–	–	tak[4])	tak[3])	tak	–	–
AXIS	–	–	–	–	–	tak	–
FRAME	–	–	–	–	–	–	tak

Objaśnienia

* Przy zmianie typu z REAL na INT następuje w przypadku ułamka >=0.5 zaokrąglenie do góry, w przeciwnym przypadku następuje zaokrąglenie do dołu (por. funkcja ROUND)

1) Wartość <>0 odpowiada TRUE, wartość==0 odpowiada FALSE

2) Gdy wartość leży w dopuszczalnym zakresie liczbowym

3) Gdy tylko 1 znak

4) Długość łańcucha znaków 0=>FALSE, w innym przypadku TRUE

Uwaga

Jeżeli przy konwersji wartość jest większa, niż zakres docelowy, następuje komunikat błędu.

Jeżeli w wyrażeniu występują mieszane typy, wówczas ich dopasowanie następuje automatycznie. Konwersje typu są możliwe również w akcjach synchronicznych, patrz punkt "Akcje synchroniczne ruchu, domyślna konwersja typu".

2.9 Operacja na łańcuchach znaków

Operacje na łańcuchach znaków

Oprócz klasycznych operacji "przyporządkowanie" i "porównanie" są możliwe następujące operacje na łańcuchach znaków:

- Konwersja typu na STRING (AXSTRING) (Strona 89)
- Konwersja typu z STRING (NUMBER, ISNUMBER, AXNAME) (Strona 90)
- Powiązanie łańcuchów znaków (<<) (Strona 91)
- Zmiana na litery małe/duże (TOLOWER, TOUPPER) (Strona 92)
- Określenie długości łańcucha znaków (STRLEN) (Strona 93)
- Szukanie znaku/łańcucha znaków w łańcuchu znaków (INDEX, RINDEX, MINDEX, MATCH) (Strona 94)
- Wybór częściowego łańcucha znaków (SUBSTR) (Strona 95)
- Odczyt i zapis pojedynczych znaków (Strona 96)
- Formatowanie łańcucha znaków (SPRINT) (Strona 97)

Znaczenie specjalne znaku 0

Znak 0 jest wewnętrznie interpretowany jako oznaczenie końca łańcucha znaków. Jeżeli znak zostanie zastąpiony przez znak 0, łańcuch znaków zostanie przez to skrócony.

Przykład:

```
Kod programu                                    Komentarz
DEF STRING[20] STRG="Oś . stoi"
STRG[6]="X"
MSG(STRG)                                       ; Daje komunikat "Oś X stoi".
STRG[6]=0
MSG(STRG)                                       ; Daje komunikat "Oś".
```

2.9.1 Konwersja typu na STRING (AXSTRING)

Dzięki funkcji "konwersja typu na STRING" można wykorzystywać zmienne różnego typu, jako część składową komunikatu (MSG).

Następuje przy domyślnym zastosowaniu operatora << dla typów danych INT, REAL, CHAR i BOOL (patrz " Powiązanie łańcuchów znaków (<<) (Strona 91) ").

Wartość INT jest zmieniana w normalnie czytelną formę. W przypadku wartości REAL jest podawanych do 10 miejsc po przecinku.

Przy pomocy polecenia `AXSTRING` zmienne typu AXIS mogą być konwertowane na STRING.

Składnia

```
<STRING_ERG> = << <dowolny_typ>
<STRING_ERG> = AXSTRING(<identyfikator osi>)
```

Znaczenie

<STRING_ERG>:	Zmienna dla wyniku konwersji typu	
	Typ:	STRING
<dowolny_typ>:	Typy zmiennych INT, REAL, CHAR, STRING i BOOL	
AXSTRING:	Polecenie AXSTRING daje podany identyfikator osi jako łańcuch znaków.	
<identyfikator osi>:	Zmienna dla identyfikatora osi	
	Typ:	AXIS

Uwaga

Zmienne FRAME nie mogą być konwertowane.

2.9.2 Konwersja typu z STRING (NUMBER, ISNUMBER, AXNAME)

Przy pomocy polecenia NUMBER następuje konwersja ze STRING na REAL. Konwertowalność można sprawdzić poleceniem ISNUMBER.

Przy pomocy polecenia AXNAME łańcuch znaków jest konwertowany na typ danych AXIS.

Składnia

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

Znaczenie

NUMBER:	Polecenie NUMBER zwraca liczbę przedstawioną przez <łańcuch znaków> jako wartość REAL.	
<łańcuch znaków>:	Konwertowana zmienna typu STRING	
<REAL_ERG>:	Zmienna dla wyniku konwersji typu przy pomocy NUMBER	
	Typ:	REAL
ISNUMBER:	Przy pomocy polecenia ISNUMBER można sprawdzić, czy <łańcuch znaków> można przekonwertować na poprawną liczbę.	

<BOOL_ERG>:	Zmienna dla wyniku odpytania przy pomocy ISNUMBER		
	Typ:	BOOL	
	Wartość:	TRUE	ISNUMBER daje wartość TRUE, gdy <łańcuch znaków> przedstawia liczbę REAL poprawną według zasad języka.
		FALSE	Jeżeli ISNUMBER da wartość FALSE, jest przy wywołaniu NUMBER z takim samym <łańcuchem znaków> wyzwalany alarm.
AXNAME:	Polecenie AXNAME konwertuje podany <łańcuch znaków> na identyfikator osi. **Wskazówka:** Jeżeli <łańcuch znaków> nie może zostać przyporządkowany do żadnego z zaprojektowanych identyfikatorów osi, jest wyzwalany alarm.		
<AXIS_ERG>:	Zmienna dla wyniku konwersji typu przy pomocy AXNAME		
	Typ:	AXIS	

Przykład

```
Kod programu                                    Komentarz
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 Powiązanie łańcuchów znaków (<<)

Funkcja "Powiązanie łańcuchów znaków" stwarza możliwość składania łańcucha znaków z poszczególnych części składowych.

Powiązanie jest realizowane przez operator "<<". Ten operator ma typ docelowy STRING dla wszystkich kombinacji typów bazowych CHAR, BOOL, INT, REAL i STRING. Ewentualnie niezbędna konwersja jest dokonywana według istniejących zasad.

Składnia

```
<dowolny_typ> << <dowolny_typ>
```

Znaczenie

<dowolny_typ>:	Zmienna typu CHAR, BOOL, INT, REAL lub STRING
<< :	Operator do powiązania zmiennych (<dow._typ>) w złożony łańcuch znaków (typ STRING). Ten operator jest dostępny również jako jedyny tzw. wariant "unarny". W ten sposób jest możliwe wykonanie bezpośredniej konwersji typu na STRING (nie dla FRAME i AXIS): << <dow._typ>

Na przykład można w ten sposób złożyć komunikat albo polecenie z list tekstowych i wstawić parametr (na przykład nazwę modułu):
`MSG(STRG_TAB[LOAD_IDX]<<MODUŁ_NAZWA)`

Uwaga

Wyniki pośrednie przy wiązaniu łańcuchów znaków nie mogą przekraczać maksymalnej długości łańcucha znaków.

Uwaga

Typy FRAME i AXIS nie mogą być stosowane razem z operatorem "<<".

Przykłady

Przykład 1: Powiązanie łańcuchów znaków

```
Kod programu                                          Komentarz
DEF INT IDX=2
DEF REAL VALUE=9.654
DEF STRING[20] STRG="INDEX:2"
IF STRG=="Index:"<<IDX GOTOF NO_MSG
MSG ("indeks:" <<IDX <<"/wartość:" <<VALUE)        ; Wyświetlenie:
                                                    "indeks: 2/wartość: 9.654"
NO_MSG:
```

Przykład 2: Bezpośrednia konwersja typu za pomocą <<

```
Kod programu              Komentarz
DEF REAL VALUE=3.5
<<VALUE                   ; Podana zmienna typu REAL jest konwertowana na typ
                          STRING.
```

2.9.4 Zmiana na litery małe/duże (TOLOWER, TOUPPER)

Funkcja "zmiana na litery małe/duże" pozwala zmienić wszystkie litery łańcucha znaków na jednolite ich przedstawienie.

Składnia

```
<STRING_ERG>=TOUPPER("<łańcuch znaków>")
<STRING_ERG>=TOLOWER("<łańcuch znaków>")
```

Znaczenie

TOUPPER:	Przy pomocy polecenia TOUPPER wszystkie litery łańcucha znaków są zamieniane na litery **duże**.	
TOLOWER:	Przy pomocy polecenia TOLOWER wszystkie litery łańcucha znaków są zamieniane na litery **małe**.	
<łańcuch znaków>:	Łańcuch znaków, który ma zostać przekonwertowany	
	Typ:	STRING
<STRING_ERG>:	Zmienna dla wyniku konwersji	
	Typ:	STRING

Przykład

Ponieważ jest również możliwe zainicjalizowanie wprowadzeń użytkownika na interfejsie graficznym, można uzyskać jednolitą prezentację z literami małymi lub dużymi:

Kod programu

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

2.9.5 Określenie długości łańcucha znaków (STRLEN)

Przy pomocy polecenia STRLEN jest możliwe określenie długości łańcucha znaków.

Składnia

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

Znaczenie

STRLEN:	Przy pomocy polecenia STRLEN jest określana długość podanego łańcucha znaków. Jest zwracana liczba znaków, które - licząc od początku łańcucha znaków - nie są znakiem 0.	
<łańcuch znaków>:	Łańcuch znaków, którego długość ma zostać określona	
	Typ:	STRING
<INT_ERG>:	Zmienna dla wyniku określenia	
	Typ:	INT

Przykład

Funkcja w związku z dostępem do pojedynczych znaków umożliwia określenie końca łańcucha znaków:

```
Kod programu
IF(STRLEN (MODUŁ_NAZWA) > 10) GOTOF BŁĄD
```

2.9.6 Szukanie znaku/łańcucha znaków w łańcuchu znaków (INDEX, RINDEX, MINDEX, MATCH)

Funkcja ta pozwala na poszukiwanie pojedynczych znaków wzgl. ich łańcucha w innym łańcuchu. Wyniki funkcji podają, w jakiej pozycji łańcucha znaleziono znak/łańcuch.

Składnia

`INT_ERG=INDEX(STRING,CHAR)` ; **Typ wyniku: INT**

`INT_ERG=RINDEX(STRING,CHAR)` ; **Typ wyniku: INT**

`INT_ERG=MINDEX(STRING,STRING)` ; **Typ wyniku: INT**

`INT_ERG=MATCH(STRING,STRING)` ; **Typ wyniku: INT**

Semantyka

Funkcje szukania: Zwracają one pozycję w łańcuchu znaków (pierwszy parametr), w którym poszukiwanie zakończyło się wynikiem pozytywnym. Jeżeli znaku/łańcucha nie można znaleźć, jest zwracana wartość -1. Pierwszy znak ma przy tym pozycję 0.

Znaczenie

INDEX:	Poszukuje w pierwszym parametrze znaku podanego jako drugi parametr (od przodu).
RINDEX:	Poszukuje w pierwszym parametrze znaku podanego jako drugi parametr (od tyłu).
MINDEX:	Odpowiada funkcji INDEX, oprócz tego, że jest przekazywana lista znaków (jako łańcuch znaków), z których jest zwracany indeks pierwszego znalezionego znaku.
MATCH:	Szuka łańcucha znaków w łańcuchu znaków.

W ten sposób łańcuchy znaków dają się dzielić według określonych kryteriów, na przykład w pozycjach ze spacją albo znakiem rozdzielającym przy podawaniu ścieżki ("/").

Przykład

Podział wprowadzenia na nazwy ścieżek i modułów

Kod programu	Komentarz
DEF INT PFADIDX, PROGIDX	
DEF STRING[26] WPROWADZENIE	
DEF INT LISTIDX	

Kod programu	**Komentarz**
WPROWADZENIE = "/_N_MPF_DIR/_N_EXE-CUTE_MPF"	
LISTIDX = MINDEX (WPROWADZENIE, "M,N,O,P") + 1	; Jako wartość LISTIDX jest zwracane 3; ponieważ "N" jest w parametrze WPROWADZENIE pierwszym od przodu znakiem z listy wyboru.
PFADIDX = INDEX (WPROWADZENIE, "/") +1	; przez to obowiązuje: PFADIDX = 1
PROGIDX = RINDEX (WPROWADZENIE, "/") +1	; przez to obowiązuje: PROGIDX = 12
	; Przy pomocy wprowadzonej w następnym punkcie funkcji SUBSTR zmienna WPROWADZENIE daje się rozłożyć na komponenty "ścieżka" i "moduł":
VARIABLE = SUBSTR (WPROWADZENIE, PFADIDX, PROGIDX-PFADIDX-1)	; daje wówczas "_N_MPF_DIR"
ZMIENNA = SUBSTR (WPROWADZENIE, PRO-GIDX)	; daje wówczas "_N_EXECUTE_MPF"

2.9.7 Wybór częściowego łańcucha znaków (SUBSTR)

Przy pomocy funkcji SUBSTRING można czytać dowolne części w ramach łańcucha znaków.

Składnia

```
<STRING_ERG>=SUBSTR(<łańcuch_znaków>,<indeks>,<długość>)
<STRING_ERG>=SUBSTR(<łańcuch_znaków>,<indeks>)
```

Znaczenie

SUBSTR:	Funkcja daje z <łańcucha znaków> łańcuch częściowy, rozpoczynając od <indeksu> o podanej <długości>. Gdy parametr <długości> nie jest podany, funkcja daje częściowy łańcuch znaków rozpoczynając od <indeksu> do końca łańcucha znaków.
<indeks>:	Pozycja początkowa częściowego łańcucha znaków w ramach łańcucha znaków. Jeżeli pozycja początkowa znajduje się za końcem łańcucha znaków, jest zwracany łańcuch pusty (""). Pierwszy znak łańcucha znaków: Indeks = 0 Zakres wartości: 0 ... (długość łańcucha znaków - 1)
<długość>:	Długość częściowego łańcucha znaków. Gdy zostanie podana za duża długość, jest zwracany tylko częściowy łańcuch znaków do końca łańcucha znaków. Zakres wartości: 1 ... (długość łańcucha znaków - 1)

Przykład

Kod programu	**Komentarz**
DEF STRING[29] ERG	

Kod programu	Komentarz
; 1	
; 0123456789012345678	
ERG = SUBSTR("POKWITOWANIE: 10 do 99", 10, 2)	; ERG == "10"
ERG = SUBSTR("POKWITOWANIE: 10 do 99", 10)	; ERG == "10 do 99"

2.9.8 Odczyt i zapis pojedynczych znaków

W ramach łańcucha znaków można czytać i zapisywać poszczególne znaki.

Należy przy tym przestrzegać następujących warunków brzegowych:

- możliwe tylko w przypadku zmiennych definiowanych przez użytkownika, nie w przypadku zmiennych systemowych
- poszczególne znaki łańcucha są przy wywołaniach podprogramów przekazywane tylko jako "call by value"

Składnia

```
<znak>=<łańcuch znaków>[<indeks>]
<znak>=<łańcuch_znaków_tablica>[<tablica_indeks>,<indeks>]
<łańcuch_znaków>[<indeks>]=<znak>
<łańcuch_znaków_tablica>[<tablica_indeks>,<indeks>]=<znak>
```

Znaczenie

<łańcuch znaków>:	Dowolny łańcuch znaków
<znak>:	Zmienna typu CHAR
<indeks>:	Pozycja znaku w ramach łańcucha znaków. Pierwszy znak łańcucha znaków: Indeks = 0 Zakres wartości: 0 ... (długość łańcucha znaków - 1)

Przykłady

Przykład 1: Zmienna komunikat

Kod programu	Komentarz
; 0123456789	
DEF STRING[50] KOMUNIKAT = "Oś n doszła do pozycji"	
KOMUNIKAT[6] = "X"	
MSG(KOMUNIKAT)	; "Oś X doszła do pozycji"

Przykład 2: Ewaluacja zmiennej systemowej

Kod programu	Komentarz
DEF STRING[50] STRG	; Pamięć pośrednia dla zmiennej systemowej

Kod programu	Komentarz
...	
STRG = $P_MMCA	; Załadowanie zmiennej systemowej
IF STRG[0] == "E" GOTO ...	; Ewaluacja zmiennej systemowej

Przykład 3: Przekazanie parametrów "call by value" i "call by reference"

Kod programu	Komentarz
; 0123456	
DEF STRING[50] STRG = "Achse X"	
DEF CHAR CHR	
...	
EXTERN UP_VAL(ACHSE)	; Definicja podprogramu z parametrem "call by value"
EXTERN UP_REF(VAR ACHSE)	; Definicja podprogramu z parametrem "call by reference"
...	
UP_VAL(STRG[6])	; Przekazanie parametrów "by value"
...	
CHR = STRG[6]	; Zapisywanie pośrednie
UP_REF(CHR)	; Przekazanie parametrów "by reference"

2.9.9 Formatowanie łańcucha znaków (SPRINT)

Przy pomocy predefiniowanej funkcji SPRINT mogą być formatowane łańcuchy znaków i np. przygotowywane do wyprowadzenia do urządzeń zewnętrznych (patrz też "Process DataShare - Wyprowadzenie na zewnętrzne urządzenie/plik (EXTOPEN, WRITE, EXTCLOSE) (Strona 689)").

Składnia

```
"<Wynik_String>"=SPRINT("<Format_String>",<Wartość_1>,<Wartość_2>,...
, <Wartość_n>)
```

Znaczenie

SPRINT:	Identyfikator funkcji predefiniowanej, która daje wartość typu STRING.
"<Format_String>":	Łańcuch znaków, który zawiera części stałe i zmienne. Części zmienne są ustalane poprzez formatyzator % i następujący po nim opis formatu.
< Wartość_1>,< Wartość_2>,…,< Wartość_n>:	Wartość w formie stałej albo zmiennej NC, która w miejscu, w którym znajduje się n-ty formatyzator %, jest wstawiana do <Format_String> odpowiednio do opisu formatu.
"<Wynik_String>":	Formatowany łańcuch znaków (maksymalnie 400 bajtów)

Dostępne opisy formatów

%B:	Konwersja na łańcuch znaków "TRUE", gdy konwertowana wartość:
	• jest nierówna 0.
	• nie jest pustym łańcuchem znaków (w przypadku wartości string).
	Konwersja na łańcuch znaków "FALSE", gdy konwertowana wartość:
	• jest równa 0.
	• jest pustym łańcuchem znaków.
	Przykład:
	`N10 DEF BOOL BOOL_VAR=1`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF BOOL_VAR:TRUE".

%C:	Konwersja na znak ASCII.
	Przykład:
	`N10 DEF CHAR CHAR_VAR="X"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF CHAR_VAR:X".

%D:	Konwersja na łańcuch znaków wartości całkowitoliczbowej (INTEGER).
	Przykład:
	`N10 DEF INT INT_VAR=123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF INT_VAR:123".

%<m>D:	Konwersja na łańcuch znaków wartości całkowitoliczbowej (INTEGER). Łańcuch znaków ma długość minimalna <m> znaków. Brakujące miejsca są wypełniane do lewej spacjami.
	Przykład:
	`N10 DEF INT INT_VAR=-123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF INT_VAR:xx-123" ("x" w przykładzie zastępuje spację).

%F:	Konwersja na łańcuch znaków liczby dziesiętnej z 6 miejscami po przecinku. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1.2341234EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR)`
	Wynik: Zmienna RESULT typu string jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR: -1234.123400".

%<m>F:	Konwersja na łańcuch znaków liczby dziesiętnej o 6 miejscach po przecinku i długości całkowitej co najmniej <m> znaków. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami. Brakujące znaki są wypełniane spacjami do długości całkowitej <m> z wyrównaniem do lewej.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1.23412345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR: xxx-1234.123457" ("x" w przykładzie zastępuje spację).

%.<n>F:	Konwersja na łańcuch znaków liczby dziesiętnej z <n> miejscami po przecinku. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1.2345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR: -1234.568".

%<m>.<n>F:	Konwersja na łańcuch znaków liczby dziesiętnej o <n> miejscach po przecinku i długości całkowitej co najmniej <m> znaków. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami. Brakujące znaki są wypełniane spacjami do długości całkowitej <m> z wyrównaniem do lewej.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1.2341234567890EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF REAL_VAR:xx-1234.12" ("x" w przykładzie zastępuje spację).

%E:	Konwersja na łańcuch znaków liczby dziesiętnej w formie wykładniczej. Mantysa jest zapisywana w znormalizowaniu z jednym miejscem przed przecinkiem i 6 miejscach po przecinku. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami. Wykładnik rozpoczyna się od słowa kluczowego "EX". Dalej następuje znak liczby ("+" albo "-") i liczba dwu- lub trzycyfrowa.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1234.567890`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR:-1.234568EX+03".

%<m>E:	Konwersja na łańcuch znaków liczby dziesiętnej w formie wykładniczej o długości całkowitej co najmniej <m> znaków. Brakujące znaki są wypełniane spacjami z wyrównaniem do lewej. Mantysa jest zapisywana w znormalizowaniu z jednym miejscem przed przecinkiem i 6 miejscach po przecinku. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami. Wykładnik rozpoczyna się od słowa kluczowego "EX". Dalej następuje znak liczby ("+" albo "-") i liczba dwu- lub trzycyfrowa.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1234.5`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" ("x" w przykładzie zastępuje spację)

%.<n>E:	Konwersja na łańcuch znaków liczby dziesiętnej w formie wykładniczej. Mantysa jest zapisywana w znormalizowaniu z jednym miejscem przed przecinkiem i <n> miejscach po przecinku. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami. Wykładnik rozpoczyna się od słowa kluczowego "EX". Dalej następuje znak liczby ("+" albo "-") i liczba dwu- lub trzycyfrowa.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR:-1.23EX+03".

%<m>.<n>E:	Konwersja na łańcuch znaków liczby dziesiętnej w formie wykładniczej o długości całkowitej co najmniej <m> znaków. Brakujące znaki są wypełniane spacjami z wyrównaniem do lewej. Mantysa jest zapisywana w znormalizowaniu z jednym miejscem przed przecinkiem i <n> miejscach po przecinku. Miejsca po przecinku są ew. zaokrąglane albo wypełniane zerami. Wykładnik rozpoczyna się od słowa kluczowego "EX". Dalej następuje znak liczby ("+" albo "-") i liczba dwu- lub trzycyfrowa.
	Przykład:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF REAL_VAR:xx-1.23EX+03" ("x" w przykładzie zastępuje spację).

%G:	Konwersja na łańcuch znaków liczby dziesiętnej zależnie od zakresu wartości w formie dziesiętnej albo wykładniczej. jeżeli wartość bezwzględna przedstawianej wartości jest mniejsza niż 1.0EX-04 albo większa/równa 1.0EX+06 jest wybierana forma wykładnicza, w przeciwnym przypadku forma dziesiętna. Jest wyświetlanych maksymalnie sześć miejsc znaczących, ew. następuje zaokrąglenie.
	Przykład z przedstawieniem dziesiętnym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR: 0.000123457".
	Przykład z przedstawieniem wykładniczym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR:1.23457EX+06".

%<m>G:	Konwersja na łańcuch znaków liczby dziesiętnej zależnie od zakresu wartości w formie dziesiętnej albo wykładniczej (jak %G). Łańcuch znaków ma długość całkowitą co najmniej <m> znaków. Brakujące znaki są wypełniane spacjami z wyrównaniem do lewej.
	Przykład z przedstawieniem dziesiętnym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF REAL_VAR:xxxx0.000123457" ("x" w przykładzie zastępuje spację).
	Przykład z przedstawieniem wykładniczym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF REAL_VAR:xxx1.23457EX+06" ("x" w przykładzie zastępuje spację).

%.<n>G:	Konwersja na łańcuch znaków liczby dziesiętnej zależnie od zakresu wartości w formie dziesiętnej albo wykładniczej. Jest wyświetlanych maksymalnie <n> miejsc znaczących, ew. następuje zaokrąglenie. Jeżeli wartość bezwzględna przedstawianej wartości jest mniejsza, niż 1.0EX-04 albo większa/równa 1.0EX(+<n>) jest wybierana forma wykładnicza, w przeciwnym przypadku forma dziesiętna.
	Przykład z przedstawieniem dziesiętnym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR: 0.000123".
	Przykład z przedstawieniem wykładniczym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF REAL_VAR:1.23EX+03".

%<m>.<n>G:	Konwersja na łańcuch znaków liczby dziesiętnej zależnie od zakresu wartości w formie dziesiętnej albo wykładniczej (jak %.<n>G). Łańcuch znaków ma długość całkowitą co najmniej <m> znaków. Brakujące znaki są wypełniane spacjami z wyrównaniem do lewej.
	Przykład z przedstawieniem dziesiętnym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF REAL_VAR:xxx0.0001235" ("x" w przykładzie zastępuje spację).
	Przykład z przedstawieniem wykładniczym:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF REAL_VAR:xx1.235EX+06" ("x" w przykładzie zastępuje spację).

%.<n>P:	Konwersja wartości REAL na wartość INTEGER przy uwzględnieniu <n> miejsc po przecinku. Wartość INTEGER jest wyprowadzana, jako liczba binarna 32-bitowa. Jeżeli konwertowanej wartości nie można przedstawić przy pomocy 32 bitów, wykonywanie jest przerwane z alarmem.
	Ponieważ ciąg bajtów utworzony przy pomocy instrukcji formatu %.<n>P może zawierać również zera binarne, tak utworzony całkowity łańcuch znaków nie odpowiada już konwersjom typu danych NC STRING. Dlatego, ani nie można go zapisać w zmiennej typu STRING, ani dalej przetwarzać przy użyciu poleceń string języka NC. Jedynie możliwym zastosowaniem jest przekazanie parametrów do polecenia WRITE z wyprowadzeniem do odpowiedniego urządzenia zewnętrznego (patrz poniższy przykład).
	Gdy tylko <Format_String> zawiera opis formatu typu %P, cały łańcuch znaków, za wyjątkiem liczby binarnej wygenerowanej przy pomocy %.<n>P, jest odpowiednio do MD10750 $MN_SPRINT_FORMAT_P_CODE wyprowadzany w kodzie ASCII, ISO (DIN6024) albo EIA (RS244). Gdy zostanie zaprogramowany znak niekonwertowalny, wykonywanie jest przerywane z alarmem.
	Przykład:
	`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`
	`... ; Postępowanie z błędami`
	`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`
	`N210 IF ERROR <> 0`
	`… ; Postępowanie z błędami`
	Wynik: Łańcuch znaków "INTEGER BINARY CODED: 'H0001E23A'" jest przenoszony na urządzenie wyjścia /ext/dev/1. Wartość szesnastkowa 0x0001E23A odpowiada wartości dziesiętnej 123450.

%<m>.<n>P:	Łańcuch znaków z:
	• jedną liczbą całkowitą <m> + <n> miejsc **albo**
	• liczbą dziesiętną z maksymalnie <m> miejscami przed przecinkiem i dokładnie <n> miejscami po przecinku.
	Jak w przypadku opisu formatu %.<n>P cały łańcuch znaków jest zapisywany w kodzie znaków ustalonym przez MD10750 $MN_SPRINT_FORMAT_P_CODE.
	Konwersja przy MD10751 = 0:
	Wartość REAL jest konwertowana na łańcuch znaków z liczbą całkowitą o <m> + <n> miejscach. Ew. miejsca po przecinku są zaokrąglane do <n> miejsc albo wypełniane przez 0. Brakujące miejsca przed przecinkiem są wypełniane spacjami. Znak minus jest dołączany z wyrównaniem do lewej, w miejscu znaku plus jest umieszczana spacja.
	Przykład:
	`N10 DEF REAL REAL_VAR=-123.45`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "PUNCHED TAPE FORMAT:-xx123450" ("x" w przykładzie zastępuje spację).
	Konwersja przy MD10751 = 1:
	Wartość REAL jest konwertowana na łańcuch znaków z liczbą dziesiętną o maksymalnie <m> miejscach przed przecinkiem i dokładnie <n> miejscach po przecinku. Ew. miejsca przed przecinkiem są obcinane, a miejsca po przecinku zaokrąglane albo wypełniane przez 0. Jeżeli <n> jest równe 0, odpada również kropka dziesiętna.
	Przykład:
	`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)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "PUNCHED TAPE FORMAT:-123.450 VAR2:23".

%S:	Wstawienie łańcucha znaków.
	Przykład:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF STRING_VAR:ABCDEFG".

%<m>S:	Wstawienie łańcucha znaków o co najmniej <m> znakach. Brakujące miejsca są wypełniane spacjami.
	Przykład:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF STRING_VAR:xxxABCDEFG" ("x" w przykładzie zastępuje spację).

%.<n>S:	Wstawienie <n> znaków łańcucha znaków (rozpoczynając od pierwszego znaku).
	Przykład:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "CONTENT OF STRING_VAR:ABC".

%<m>.<n>S:	Wstawienie <n> znaków łańcucha znaków (rozpoczynając od pierwszego znaku). Utworzony łańcuch znaków ma długość całkowitą co najmniej <m> znaków. Brakujące miejsca są wypełniane spacjami.
	Przykład:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przez łańcuch znaków "CONTENT OF STRING_VAR:xxxxxABCDE" ("x" w przykładzie zastępuje spację).

%X:	Konwersja wartości INTEGER na łańcuch znaków w formie szesnastkowej.
	Przykład:
	`N10 DEF INT INT_VAR='HA5B8'`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR)`
	Wynik: Zmienna string RESULT jest opisywana przy pomocy łańcucha znaków "INTEGER HEXADECIMAL:A5B8".

Uwaga

Właściwość języka NC, nie rozróżniania dużych i małych liter odnośnie identyfikatorów i słów kluczowych, obowiązuje również dla opisów formatu. Dlatego można je bez funkcjonalnej różnicy programować małymi albo dużymi literami.

Możliwości kombinacji

Poniższa tabela informuje o tym, które typy danych NC z którym opisem formatu można zestawiać. Obowiązują reguły konwersji domyślnej typu danych (patrz "Typy danych (Strona 61)").

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

Uwaga

Tabela pokazuje, że typów danych NC AXIS i FRAME nie można bezpośrednio stosować w funkcji SPRINT. Jest jednak możliwe:

- przekonwertowanie typu danych AXIS funkcją AXSTRING na łańcuch znaków, który następnie można przetwarzać przy pomocy SPRINT.
- czytanie poszczególnych wartości typu danych FRAME poprzez dostęp do komponentów frame. Przez to uzyskuje się daną typu REAL, którą można przetwarzać dalej przy pomocy SPRINT.

2.10 Skoki i rozgałęzienia w programie

2.10.1 Skok do początku programu (GOTOS)

Przy pomocy polecenia GOTOS jest możliwe, w celu powtórzenia programu, przeskoczenie z powrotem na początek programu głównego albo podprogramu.

Poprzez dane maszynowe można ustawić, że przy każdym skoku powrotnym na początek programu:

- czas przebiegu programu zostanie ustawiony na "0".
- stan licznik obrobionych przedmiotów jest zwiększany o wartość "1".

Składnia

```
GOTOS
```

Znaczenie

GOTOS:	Instrukcja skoku z celem początek programu.	
	Wykonanie jest sterowane przez sygnał interfejsowy NC/PLC: DB21, ... DBX384.0 (sterowanie rozgałęzieniem programu)	
	Wartość:	Znaczenie:
	0	Bez skoku powrotnego do początku programu. Wykonywanie programu jest kontynuowane od następnego bloku programu obróbki po GOTOS.
	1	Skok powrotny do początku programu. Program obróbki jest powtarzany.

Warunki brzegowe

- GOTOS wyzwala wewnętrznie STOPRE (zatrzymanie przebiegu wyprzedzającego).
- W przypadku programu obróbki z definicjami danych (zmienne LUD) następuje przy pomocy GOTOS skok do pierwszego bloku programu za segmentem definicji, tzn. definicje danych nie są ponownie wykonywane. Definiowane zmienne zachowują dlatego wartość uzyskaną w bloku GOTOS i nie są cofane do wartości standardowych zaprogramowanych w segmencie definicji.
- W akcjach synchronicznych i cyklach technologicznych polecenie GOTOS nie jest dostępne.

Przykład

Kod programu	**Komentarz**
N10 ...	; Początek programu.
...	
N90 GOTOS	; Skok do początku programu.
...	

2.10.2 Skoki w programie do znaczników skoku (GOTOB, GOTOF, GOTO, GOTOC)

W programie mogą być ustawiane znaczniki skoku (etykiety), do których można wykonywać skoki z innych miejsc tego samego programu przy pomocy poleceń `GOTOF`, `GOTOB`, `GOTO` lub `GOTOC`. Wykonywanie programu jest wówczas kontynuowane od instrukcji, która następuje bezpośrednio po znaczniku skoku. Przez to mogą być realizowane rozgałęzienia w ramach programu.

Oprócz znaczników skoku są jako cele skoku możliwe również numery bloków głównych i pomocniczych.

Gdy przed instrukcją skoku jest sformułowany warunek skoku (`IF ...`), wówczas skok w programie następuje tylko wtedy, gdy ten warunek jest spełniony.

Składnia

```
GOTOB <cel skoku>
IF <warunek skoku> = TRUE GOTOB <cel skoku>

GOTOF <cel skoku>
IF <warunek skoku> = TRUE GOTOF <cel skoku>

GOTO <cel skoku>
IF <warunek skoku> = TRUE GOTO <cel skoku>

GOTOC <cel skoku>
IF <warunek skoku> = TRUE GOTOC <cel skoku>
```

Znaczenie

`GOTOB`:	Instrukcja skoku z celem w kierunku początku programu.
`GOTOF`:	Instrukcja skoku z celem w kierunku końca programu.
`GOTO`:	Instrukcja skoku z szukaniem celu. Szukanie następuje w kierunku końca programu, następnie w kierunku początku programu.
`GOTOC`:	Działanie GOTO z tą różnicą, że alarm 14080 "Cel skoku nie został znaleziony" jest ukrywany. Oznacza to, że w przypadku bezowocnego szukania celu skoku wykonywanie programu nie jest przerywane, lecz jest kontynuowane od wiersza programu następującego po poleceniu GOTOC.

<cel skoku>:	Parametr celu skoku Możliwymi danymi są:	
	<Znacznik skoku>:	Celem skoku jest umieszczony w programie znacznik skoku o nazwie zdefiniowanej przez użytkownika: <znacznik skoku>:
	<Numer bloku>:	Celem skoku jest numer bloku głównego lub pomocniczego (np.: 200, N300)
	Zmienna typu STRING:	Zmienny cel skoku. Zmienna oznacza znacznik skoku albo numer bloku.
IF:	Słowo kluczowe do sformułowania warunku skoku. Warunek skoku dopuszcza wszystkie operacje porównań i logiczne (wynik: TRUE albo FALSE). Skok w programie jest wykonywany, gdy wynik tej operacji jest TRUE.	

Uwaga

Znaczniki skoku (etykiety)

Znaczniki skoku zawsze znajdują się na początku bloku. Gdy dostępny jest numer bloku, znacznik skoku znajduje się bezpośrednio za tym numerem.

Dla nazywania znaczników skoku obowiązują następujące zasady:

- Maksymalna liczba znaków:
  - Co najmniej 2
  - Co najwyżej 32
- Dopuszczalnymi znakami są:
  - Litery
  - Cyfry
  - Podkreślenia
- Pierwsze dwa znaki muszą być literami albo podkreśleniami.
- Po nazwie znacznika skoku następuje dwukropek (":").

Warunki brzegowe

- Celem skoku może być tylko blok ze znacznikiem skoku albo numer bloku, który leży **w ramach** programu.
- Instrukcja skoku bez warunku skoku musi zostać zaprogramowana w oddzielnym bloku. W przypadku instrukcji skoku z warunkami skoku to ograniczenie nie obowiązuje. Tutaj można sformułować wiele instrukcji skoku w jednym bloku.
- W przypadku programów z instrukcjami skoku bez warunków skoku koniec programu M2/M30 nie musi koniecznie znajdować się na końcu programu.

Przykłady

Przykład 1: Skoki do znaczników skoku

```
Kod programu                        Komentarz
N10 …
N20 GOTOF Etykieta_1                ; Skok w kierunku końca programu do
                                    ; Znacznik skoku "Etykieta_1”.
N30 …
N40 Etykieta_0: R1=R2+R3            ; Znacznik skoku "Etykieta_0" ustawiony.
N50 …
N60 Etykieta_1:                     ; Znacznik skoku "Etykieta_1" ustawiony.
N70 …
N80 GOTOB Etykieta_0                ; skok w kierunku początku programu
                                    ; do znacznika skoku "Etykieta_0”.
N90 …
```

Przykład 2: Skok pośredni do numeru bloku

```
Kod programu                        Komentarz
IF <warunek> == TRUE
   R10=100                          ; przyporządkowanie celu skoku
ELSE
   R10=110                          ; przyporządkowanie celu skoku
ENDIF
; Skok w kierunku końca programu do bloku, którego numer znajduje się w R10
N10 GOTOF "N"<<R10
...
N90 …
N100 ...                            ; Cel skoku
N110 ...
...
```

Przykład 3: Skok do zmiennego celu skoku

```
Kod programu                        Komentarz
DEF STRING[20] CEL
IF <warunek> == TRUE
   CEL = "Znacznik2"                ; przyporządkowanie celu skoku
ELSE
   CEL = "Znacznik2"                ; przyporządkowanie celu skoku
ENDIF
; Skok w kierunku końca programu do zmiennego celu skoku „Treść CELU.
GOTOF CEL
Znacznik1: T="wiertło1"             ; Cel skoku 1
...
Znacznik2: T="wiertło2"             ; Cel skoku 2
```

Kod programu	Komentarz
...	

Przykład 4: Skok z warunkiem skoku

Kod programu	Komentarz
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; przyporządkowanie wartości początkowych
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; Znacznik skoku LA1
N42 R1=R1+R3 R4=R4-1	
; IF warunek skoku= = TRUE	
; THEN skok w kierunku początku programu do znacznika LA1	
N43 IF R4>0 GOTOB LA1	
N44 M30	; Koniec programu

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

Funkcja CASE stwarza możliwość sprawdzenia aktualnej wartości (typ: INT) zmiennej lub funkcji obliczeniowej i wykonania skoku do różnych miejsc w programie zależności od wartości wyniku.

Składnia

```
CASE(<wyrażenie>) OF <stała_1> GOTOF <cel_skoku_1> <stała_2> GOTOF
<cel_skoku_2> ... DEFAULT GOTOF <cel skoku_n>
```

Znaczenie

CASE:	Instrukcja skoku	
<wyrażenie>:	Zmienna lub funkcja obliczeniowa	
OF:	Słowo kluczowe do formułowania warunkowych rozgałęzień programu	
<stała_1>:	Pierwsza podana stała wartość dla zmiennej lub funkcji obliczeniowej	
	Typ:	INT
<stała_2>:	Druga podana stała wartość dla zmiennej lub funkcji obliczeniowej	
	Typ:	INT
DEFAULT:	Dla przypadków, w których zmienna albo funkcja obliczeniowa nie przyjmuje żadnej z podanych stałych wartości, można przy pomocy instrukcji DEFAULT określić cel skoku. **Wskazówka:** W przypadku gdy instrukcja DEFAULT nie jest zaprogramowana, celem skoku jest blok następujący po instrukcji CASE.	
GOTOF:	Instrukcja skoku z celem w kierunku końca programu. Zamiast GOTOF można programować również wszystkie inne polecenia GOTO (patrz temat "Skoki w programie do znaczników skoku").	

<cel_skoku_1>:	Na ten cel skoku następuje rozgałęzienie, gdy wartość zmiennej albo funkcji obliczeniowej odpowiada pierwszej podanej stałej. Cel skoku może zostać podany następująco:	
	<Znacznik skoku>:	Celem skoku jest umieszczony w programie znacznik skoku o nazwie zdefiniowanej przez użytkownika: <znacznik skoku>:
	<Numer bloku>:	Celem skoku jest numer bloku głównego lub pomocniczego (np.: 200, N300)
	Zmienna typu STRING:	Zmienny cel skoku. Zmienna oznacza znacznik skoku albo numer bloku.
<cel_skoku_2>:	Na ten cel skoku następuje rozgałęzienie, gdy wartość zmiennej albo funkcji obliczeniowej odpowiada drugiej podanej stałej.	
<cel_skoku_n>:	Na ten cel skoku następuje rozgałęzienie, gdy wartość zmiennej nie odpowiada żadnej z podanych wartości stałych.	

Przykład

Kod programu

```
...
N20 DEF INT VAR1 VAR2 VAR3
N30 CASE(VAR1+VAR2-VAR3) OF 7 GOTOF Etykieta_1 9 GOTOF
Etykieta_2 DEFAULT GOTOF Etykieta_3
N40 Etykieta_1: G0 X1 Y1
N50 Etykieta_2: G0 X2 Y2
N60 Etykieta_3: G0 X3 Y3
...
```

Instrukcja CASE z N30 definiuje następujące możliwości rozgałęziania programów:

1. Gdy wartość funkcji obliczeniowej VAR1+VAR2-VAR3 = 7, wówczas przeskocz do bloku z definicją znacznika skoku "Etykieta_1" (→ N40).
2. Gdy wartość funkcji obliczeniowej VAR1+VAR2-VAR3 = 9, wówczas przeskocz do bloku z definicją znacznika skoku "Etykieta_2" (→ N50).
3. Gdy wartość funkcji obliczeniowej VAR1+VAR2-VAR3 nie wynosi ani 7 ani 9, wówczas przeskocz do bloku z definicją znacznika skoku "Etykieta_3" ((→ N60).

2.11 Powtórzenie programu (REPEAT, REPEATB, ENDLABEL, P)

Powtarzanie części programu umożliwia powtórzenie już napisanych części programu w ramach jednego programu w dowolnym układzie.

Będące do powtórzenia wiersze wzgl. części programu są zaznaczane znacznikami skoku (etykietami).

Uwaga

Znaczniki skoku (etykiety)

Znaczniki skoku zawsze znajdują się na początku bloku. Gdy jest numer bloku, znacznik skoku znajduje się bezpośrednio za tym numerem.

Dla nazywania znaczników skoku obowiązują następujące zasady:

- Maksymalna liczba znaków:
  - Co najmniej 2
  - Co najwyżej 32
- Dopuszczalnymi znakami są:
  - Litery
  - Cyfry
  - Podkreślenia
- Pierwsze dwa znaki muszą być literami albo podkreśleniami.
- Po nazwie znacznika skoku następuje dwukropek (":").

Składnia

1. Powtórzenie pojedynczego wiersza programu:

```
<Znacznik skoku>: ...
...
REPEATB <znacznik skoku> P=<n>
...
```

2. Powtórzenie części programu między znacznikiem skoku i instrukcją REPEAT:

```
<Znacznik skoku>: ...
...
REPEAT <znacznik skoku> P=<n>
...
```

3. Powtórzenie części między dwoma znacznikami skoku:

```
<Startowy znacznik skoku>: ...
...
<Końcowy znacznik skoku>: ...
...
```

```
REPEAT <startowy znacznik skoku> <końcowy znacznik sko-
ku> P=<n>
...
```

Uwaga

Wzięcie w nawias instrukcji `REPEAT` z obydwoma znacznikami skoku jest niemożliwe. Jeżeli `<startowy znacznik skoku>` zostanie znaleziony przed instrukcją `REPEAT`, a dojście do `<końcowego znacznika skoku>` nie nastąpi przed instrukcją `REPEAT`, wówczas jest przeprowadzane powtórzenie między `<startowym znacznikiem skoku>` i instrukcją `REPEAT`.

4. Powtórzenie części między znacznikiem skoku i ETYKIETĄ KOŃCOWĄ:

```
<Znacznik skoku>: ...
...
ETYKIETA KONCOWA: ...
...
REPEAT <znacznik skoku> P=<n>
...
```

Uwaga

Wzięcie w nawias instrukcji `REPEAT` ze `<znacznikiem skoku>` i `ETYKIETĄ KOŃCOWĄ` jest niemożliwe. Jeżeli `<startowy znacznik skoku>` zostanie znaleziony przed instrukcją `REPEAT`, a dojście do `<ETYKIETY KOŃCOWEJ>` nie nastąpi przed instrukcją `REPEAT`, wówczas jest przeprowadzane powtórzenie między `<znacznikiem skoku>` i instrukcją `REPEAT`.

Znaczenie

`REPEATB`:	Polecenie do powtórzenia wiersza programu
`REPEAT`:	Polecenie do powtórzenia części programu

<Znacznik skoku>:	<Znacznik skoku> oznacza: • będący do powtórzenia wiersz programu (w przypadku REPEATB) wzgl. • początek będącej do powtórzenia części programu (w przypadku REPEAT)
	Wiersz programu oznaczony przy pomocy <znacznika skoku> może znajdować się przed lub za instrukcję REPEAT/REPEATB. Szukanie następuje najpierw w kierunku początku programu. Gdy etykieta nie zostanie w tym kierunku znaleziona, wówczas szukanie następuje w kierunku końca programu **Wyjątek:** Gdy ma zostać powtórzona część programu między znacznikiem skoku i instrukcją REPEAT (patrz 2. pod "Składnia"), wówczas wiersz programu oznaczony przez <znacznik skoku> musi znajdować się **przed** instrukcjąREPEAT, ponieważ w tym przypadku szukanie odbywa się **tylko** w kierunku do początku programu. Jeżeli blok ze <znacznikiem skoku> zawiera dalsze instrukcje, wówczas są one ponownie wykonywane przy każdym powtórzeniu.
ENDLABEL:	Słowo kluczowe, które zaznacza koniec będącej do powtórzenia części programu Jeżeli blok z ENDLABE zawiera dalsze instrukcje, wówczas są one ponownie wykonywane przy każdym powtórzeniu. ENDLABE może być wielokrotnie powtarzana w programie.
P:	Adres do podania liczby powtórzeń
<n>:	Liczba powtórzeń części programu
	Typ: INT
	Będąca do powtarzania część programu jest powtarzana <n> razy. Po ostatnim powtórzeniu program jest kontynuowany od wiersza następującego po wierszu z REPEAT/REPEATB. **Wskazówka:** Jeżeli nie podano P=<n>, będąca do powtórzenia część programu jest powtarzana dokładnie jeden raz.

Przykłady

Przykład 1: Powtórzenie pojedynczego wiersza programu:

```
Kod programu                    Komentarz
N10 POZYCJA1: X10 Y20
N20 POZYCJA2: CYCLE(0,,9,8)     ; Cykl pozycji
N30 ...
N40 REPEATB POSITION1 P=5       ; Wykonaj BLOK N10 pięć razy.
N50 REPEATB POSITION2           ; Wykonaj blok N20 jeden raz.
N60 ...
N70 M30
```

Przykład 2: Powtórzenie części programu między znacznikiem skoku i instrukcją REPEAT

```
Kod programu                Komentarz
N5 R10=15
```

Kod programu	Komentarz
N10 Początek: R10=R10+1	; Szerokość
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	; Wykonaj cztery razy zakres N10 do N70.
N90 Z10	
N100 M30	

Przykład 3: Powtórzenie części między dwoma znacznikami skoku

Kod programu	Komentarz
N5 R10=15	
N10 Początek: R10=R10+1	; Szerokość
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	; Wykonaj trzy razy zakres N10 do N70.
N110 Z10	
N120 M30	

Przykład 4: Powtórzenie części między znacznikiem skoku i ETYKIETĄ KOŃCOWĄ

Kod programu	Komentarz
N10 G1 F300 Z-10	
N20 BEGIN1:	
N30 X10	
N40 Y10	
N50 BEGIN2:	
N60 X20	
N70 Y30	
N80 ETYKIETA KOŃCOWA: Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3: X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	; Wykonaj trzy razy zakres N110 do N120.
N140 REPEAT BEGIN2 P=2	; Wykonaj dwa razy zakres N50 do N80.

Kod programu	Komentarz
N150 M100	
N160 REPEAT BEGIN1 P=2	; Wykonaj dwa razy zakres N20 do N80.
N170 Z10	
N180 X0 Y0	
N190 M30	

Przykład 5: Obróbka frezarska, obróbka w pozycjach wiercenia przy pomocy różnych technologii

Kod programu	Komentarz
N10 NAWIERTAK()	; Wprowadzenie nawiertaka do pozycji roboczej.
N20 POS_1:	; Pozycje wiercenia 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	
N80 POS_2:	; Pozycje wiercenia 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 WIERTLO()	; Wprowadzenie wiertła do pozycji roboczej i cykl wiercenia.
N140 GWINT(6)	; wprowadzenie do pozycji roboczej gwintownika M6 i cykl gwintowania.
N150 REPEAT POS_1	; Powtórz segment programu od POS_1 jeden raz do ETYKIETA KOŃCOWA.
N160 WIERTLO()	; Wprowadzenie wiertła do pozycji roboczej i cykl wiercenia.
N170 GWINT(8)	; Wprowadzenie do pozycji roboczej gwintownika M8 i cykl gwintowania.
N180 REPEAT POS_2	; Powtórz segment programu od POS_2 jeden raz do ETYKIETA KOŃCOWA.
N190 M30	

Dodatkowe informacje

- Powtarzanie części programu można wywoływać w sposób kaskadowy. Każde wywołanie zajmuje jeden poziom podprogramu.
- Jeżeli podczas wykonywania powtarzania części programu jest zaprogramowane `M17` lub `RET`, wówczas powtarzanie części programu jest przerywane. Program jest kontynuowany od bloku następującym po wierszu `REPEAT`.

- W aktualnym wyświetleniu programu powtórzenie części programu jest wyświetlane jako oddzielny poziom podprogramu.
- Jeżeli podczas wykonywania części programu zostanie wyzwolone anulowanie poziomu, wówczas program jest kontynuowany po wywołaniu wykonywania części programu.
  Przykład:

```
Kod programu                        Komentarz
N5 R10=15
N10 POCZĄTEK: R10=R10+1             ; Szerokość
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10                           ; Anulowanie płaszczyzny
N50 X=-R10
N60 Y=-R10
N70 END: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3
N120 Z10                            ; Kontynuowanie wykonywania programu.
N130 M30
```

- Struktur kontrolnych i powtarzania części programu można używać w sposób kombinowany Nie powinno być jednak krzyżowania się. Powtórzenie części programu powinno leżeć w ramach jednej gałęzi struktury kontrolnej wzgl. jedna struktura kontrolna w ramach jednego powtórzenia części programu.
- Przy pomieszaniu skoków i powtórzenia części programu bloki są wykonywane czysto sekwencyjnie. Jeżeli np. skok następuje z powtórzenia programu, wówczas wykonywanie następuje tak długo, aż zostanie znaleziony koniec części programu.
  Przykład:

```
Kod programu
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
```

Kod programu
`N160 M30`

Uwaga

Instrukcja `REPEAT` powinna znajdować się za blokami ruchu.

2.12 Struktury kontrolne

Sterowanie standardowo wykonuje bloki NC w zaprogramowanej kolejności.

Ta kolejność może być zmieniana przez programowanie alternatywnych bloków programu i pętli programowych. Programowanie struktur kontroli następuje za pomocą słów kluczowych `IF`, `ELSE`, `ENDIF`, `LOOP`, `FOR`, `WHILE` i `REPEAT`.

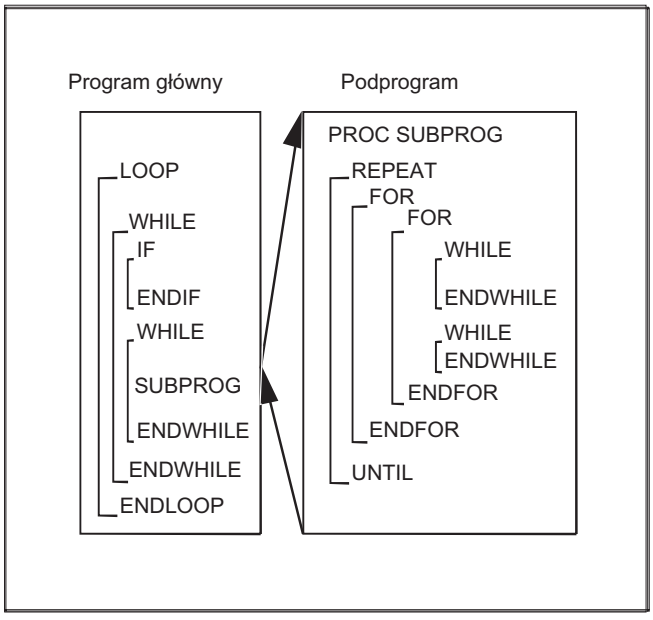
UWAGA
Błąd programowania Struktury kontrolne są możliwe tylko w ramach części instrukcyjnej programu. Definicje w nagłówku programu nie mogą być powtarzane warunkowo albo powtarzalnie. Na słowa kluczowe dla struktur kontrolnych nie wolno, tak samo jak na cele skoku, nakładać makropoleceń. Sprawdzenie w przypadku definicji makra nie następuje.

Działanie

Struktura kontrolna nie może być stosowana w rozszerzeniu na inne programy.

Głębokość kaskadowania

W ramach każdej płaszczyzny podprogramu jest możliwa głębokość kaskadowania wynosząca do 16 struktur kontrolnych.



Zachowanie się pod względem czasu przebiegu

W standardowo aktywnej pracy interpretera można poprzez zastosowanie skoków w programie uzyskać szybszy przebieg programu, niż ze strukturami kontrolnymi.

W cyklach prekompilowanych nie ma żadnej różnicy między skokami w programie i strukturami kontrolnymi.

Aktualne wyświetlenie bloku przy pętlach w programie

Jeżeli podczas pętli w programie wykonywane są tylko bloki przebiegu wyprzedzającego, na wyświetlaczu bloku wyświetli się ostatni **przed** pętlą w programie blok przebiegu głównego.

Aby np. celu diagnozy na wyświetlaczu bloku widoczne były również wykonywane bloki przebiegu wyprzedzającego, musi zostać aktywowany dekodowany pojedynczy blok SBL2.

Literatura

Podręcznik działania Funkcje podstawowe, Rozdział: BAG, kanał, tryb pracy programu, zachowanie się przy zresetowaniu (K1) > blok pojedynczy > dekodowany blok pojedynczy SBL2 z domyślnym zatrzymaniem przebiegu wyprzedzającego

Pętla bez przebiegu głównego

Jeżeli w ramach pętli programowej nie zaprogramowano przebiegu głównego, pętla jest tak długo przetwarzana w przebiegu wyprzedzającym, aż spełniony zostanie warunek pętli.

W ten sposób powstaje wysokie obciążenie (obliczeniami), które może wpływać na wyświetlanie.

Środek zaradczy - **do** pętli należy wstawić polecenie STOPRE lub czas oczekiwania G04 o wartości 0 sekund.

Warunki brzegowe

- Bloki z elementami struktur kontrolnych nie mogą zostać ukryte.
- Znaczniki skoku (etykiety) są niedozwolone w blokach z elementami struktur kontrolnych.
- Struktury kontrolne są wykonywane interpretacyjnie. Przy rozpoznaniu końca pętli jest przy uwzględnieniu znalezionych przy tym struktur kontrolnych szukany początek pętli. Dlatego w pracy interpretera struktura bloków programu nie jest kompletnie sprawdzana.
- Zasadniczo zaleca się nie stosowanie w sposób mieszany struktur kontrolnych i skoków w programie.
- Przy wyprzedzającym przetwarzaniu cykli może zostać sprawdzone prawidłowe kaskadowanie struktur kontrolnych.

2.12.1 Warunkowa instrukcja i rozgałęzienie (IF, ELSE, ENDIF)

Instrukcja warunkowa: IF - blok programu - ENDIF

W przypadku instrukcji warunkowej, jeśli warunek jest spełniony, zostanie wykonany blok programu znajdujący się pomiędzy `IF` oraz `ENDIF`.

Rozgałęzienie: IF - blok programu_1 - ELSE - blok programu_2 - ENDIF

Przy rozgałęzieniu będzie zawsze wykonywany jeden z dwóch bloków programu.

Jeżeli warunek jest spełniony, blok programu_1 znajdujący się pomiędzy `IF` i `ELSE` zostanie wykonany.

Jeżeli warunek **nie** jest spełniony, zostanie wykonany blok programu_2 znajdujący się pomiędzy `ELSE` i `ENDIF`.

Składnia

Instrukcja warunkowa

```
IF <warunek>
   Blok programu                          ; Wykonanie przy: <warunek> == TRUE
ENDIF
```

Rozgałęzienie

```
IF <warunek>
   Blok_programu_1                        ; Wykonanie przy: <warunek> == TRUE
ELSE
   Blok_programu_2                        ; Wykonanie przy: <warunek> == FALSE
ENDIF
```

Znaczenie

`IF`:	Rozpoczyna instrukcję warunkową lub rozgałęzienie.
`ELSE`:	Rozpoczyna alternatywny blok programu.
`ENDIF`:	Zaznacza koniec instrukcji warunkowej lub rozgałęzienia.
`<warunek>`:	Wyrażenie logiczne, którego ewaluacja wykazuje TRUE lub FALSE.

Przykład: Podprogram zmiany narzędzia

Kod programu	**Komentarz**
`PROC L6`	`Procedura zmiany narzędzia`
`N500 DEF INT TNR_AKTUALNE`	`Zmienna dla aktywnego numeru T`
`N510 DEF INT TNR_PRESELEKCJA`	`Zmienna dla wstępnie wybranego numeru T`
	`Określenie aktualnego narzędzia`
`N520 STOPRE`	
`N530 IF $P_ISTEST`	`W trybie testu programu ...`
`N540 TNR_AKTUALNE = $P_TOOLNO`	`... odczytane z kontekstu programu „aktualne” narzędzie.`
`N550 ELSE`	`W przeciwnym przypadku ...`
`N560 TNR_AKTUALNE = $TC_MPP6[9998,1]`	`... odczytane narzędzie wrzeciona.`
`N570 ENDIF`	
`N580 GETSELT(TNR_PRESELEKCJA)`	`Odczyt numeru T wstępnie wybranego narzędzia we wrzecionie.`

Kod programu	Komentarz
N590 IF TNR_AKTUALNY <> TNR_PRESELEKCJA	Gdy wstępnie wybrane narzędzie nie jest jeszcze aktualnym narzędziem, wówczas ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... ruch do punktu zmiany narzędzia ...
N610 M206	... i wykonać wymianę narzędzia.
N620 ENDIF	
N630 M17	

2.12.2 Pętla programowa bez końca (LOOP, ENDLOOP)

Pętla bez końca znajduje zastosowanie w programach bez końca. Na końcu pętli następuje zawsze przeskok do jej początku.

Składnia

```
LOOP
...
ENDLOOP
```

Znaczenie

LOOP:	Rozpoczyna pętlę bez końca.
ENDLOOP:	Zaznacza koniec pętli i powoduje powrót do jej początku.

Przykład

Kod programu
...
LOOP
MSG ("Żadne ostrze narzędzia nie jest aktywne")
M0
STOPRE
ENDLOOP
...

2.12.3 Pętla FOR (FOR ... TO ..., ENDFOR)

Pętla FOR jest stosowana, gdy przebieg pracy ma być powtarzany ze stałą liczbą przebiegów.

Składnia

```
FOR <zmienna> = <wartość początkowa> TO <wartość końco-
wa>
...
ENDFOR
```

Znaczenie

FOR:	Rozpoczyna pętlę FOR.	
ENDFOR:	Zaznacza koniec pętli i powoduje skok powrotny do jej początku, jak długo wartość końcowa liczenia nie jest jeszcze osiągnięta.	
<zmienna>:	Zmienna sterująca, która jest zwiększana od wartości początkowej do końcowej i przy każdym przebiegu zwiększa się o wartość "1".	
	Typ	INT lub REAL **Wskazówka:** Typ REAL jest wybierany, gdy np. są programowane parametry R dla pętli FOR. Gdy zmienna sterująca jest typu REAL, jej wartość jest zaokrąglana do wartości całkowitoliczbowej.
<wartość początkowa>:	Wartość początkowa liczenia Warunek: Wartość początkowa musi być mniejsza, niż wartość końcowa.	
<Wartość końcowa>:	Wartość końcowa liczenia	

Przykłady

Przykład 1: Zmienna INTEGER lub parametr R, jako zmienna sterująca

Zmienna INTEGER jako zmienna sterująca:

```
DEF INT iVARIABLE1
R10=R12-R20*R1 R11=6
FOR iVARIABLE1= R10 TO R11          ; Zmienna sterująca = zmienna INTEGER
 R20=R21*R22+R33
ENDFOR
M30
```

Parametr R jako zmienna sterująca:

```
R11=6
FOR R10=R12-R20*R1 TO R11           ; Zmienna sterująca = parametr R (zmienna real)
 R20=R21*R22+R33
ENDFOR
M30
```

Przykład 2: Wykonanie stałej liczby części

Kod programu	Komentarz
DEF INT LICZBA_SZTUK	; Definiuje zmienną typu INT o nazwie "LICZBA_SZTUK"
FOR LICZBA_SZTUK = 0 TO 100	; Rozpoczyna pętlę FOR. Zmienna "LICZBA_SZTUK" jest inkrementowana od wartości początkowej "0" do wartości końcowej "100".
G01 …	
ENDFOR	; Koniec pętli FOR.
M30	

2.12.4 Pętla programowa z warunkiem na początku pętli (WHILE, ENDWHILE)

W przypadku pętli WHILE warunek znajduje się na jej początku. Jak długo warunek jest spełniony, pętla WHILE będzie wykonywana.

Składnia

```
WHILE <warunek>
...
ENDWHILE
```

Znaczenie

WHILE:	Rozpoczyna pętlę programową.
ENDWHILE:	Zaznacza koniec pętli i powoduje powrót do jej początku.
<warunek>:	Warunek, który musi być spełniony, aby pętla WHILE była wykonywana.

Przykład

Kod programu	Komentarz
...	
WHILE $AA_IW[OŚ WIERCENIA] > -10	Wywołanie pętli WHILE pod następującym warunkiem: Aktualna wartość zadana w WKS dla osi wiercenia musi być większa, niż -10.
G1 G91 F250 AX[OŚ WIERCENIA] = -1	
ENDWHILE	
...	

2.12.5 Pętla programowa z warunkiem na końcu (REPEAT, UNTIL)

W przypadku pętli REPEAT warunek znajduje się na jej końcu. Pętla REPEAT jest wykonywana jeden raz i powtarzana tak długo, aż warunek zostanie spełniony.

Składnia

```
REPEAT
...
UNTIL <warunek>
```

Znaczenie

REPEAT:	Rozpoczyna pętlę programową.
UNTIL:	Zaznacza koniec pętli i powoduje powrót do jej początku.
<warunek>:	Warunek, który musi być spełniony, aby pętla REPEAT nie była już wykonywana.

Przykład

```
Kod programu                    Komentarz
...
REPEAT                          ; Wywołanie pętli REPEAT.
...
UNTIL ...                       ; Sprawdzenie, czy warunek jest spełniony.
...
```

2.12.6 Przykład programu z kaskadowanymi strukturami kontrolnymi

```
Kod programu                                    Komentarz
LOOP
   IF NOT $P_SEARCH                             ; IF bez szukania bloku
      G1 G90 X0 Z10 F1000
      WHILE $AA_IM[X] <= 100                    ; WHILE (wartość zadana osi X<=100)
         G1 G91 X10 F500                        ; Układ otworów
         Z-5 F100
         Z5
      ENDWHILE
   ELSE                                         ; ELSE Szukanie bloku
      MSG ("W trakcie szukania wiercenie nie
następuje")
   ENDIF                                        ; ENDIF
   $A_OUT[1] = 1                                ; następna płyta wiertarska
   G4 F2
```

Kod programu	Komentarz
ENDLOOP	
M30	

2.13 Polecenia koordynacji (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Zasadniczo kanał w NC może wykonywać uruchomiony w nim program niezależnie od innych kanałów swojej grupy trybów pracy (BAG). Jednakże, jeżeli kilka programów w kilku kanałach BAG uczestniczy w tym samym czasie w obróbce przedmiotu, przebiegi programów muszą być skoordynowane z kolejnymi poleceniami koordynacji w różnych kanałach.

Warunek

Wszystkie kanały uczestniczące w koordynacji programu muszą należeć do tej **samej** grupy trybów pracy (BAG):

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<kanał>] = <numer BAG>

Nazwa kanału zamiast numeru kanału

Parametrami predefiniowanych procedur do koordynacji programu mogą być również stosowane wprowadzone nazwy kanałów w MD20000 $MC_CHAN_NAME[<indeks kanału>] zamiast numerów kanałów. Stosowanie nazw kanałów w programie NC musi być najpierw zwolnione:

MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE

Składnia

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

Znaczenie

`INIT()`:	Predefiniowana procedura do wyboru programu NC, który ma zostać wykonany w podanym kanale
`START()`:	Predefiniowana procedura do uruchomienia programu wybranego w danym kanale
`WAITM()`:	Predefiniowana procedura czekania na dojście do znacznika w podanych kanałach We własnym kanale podany znacznik jest ustawiany przez `WAITM`. Blok poprzedzający ulega zakończeniu z zatrzymaniem dokładnym. Znacznik czekania jest kasowany po synchronizacji. Równocześnie można ustawić maks. 10 znaczników na kanał.
`WAITE()`:	Predefiniowana procedura dla czekania na zakończenie programu w jednym lub wielu innych kanałach
`WAITMC()`:[1)]	Predefiniowana procedura czekania na dojście do znacznika w podanych kanałach W przeciwieństwie do `WAITM` wyhamowywanie osi do zatrzymania dokładnego rozpoczyna się tylko wtedy, gdy inne kanały jeszcze nie doszły do znacznika czekania.

SETM(): [1]	Predefiniowana procedura dla ustawienia jednego lub wielu znaczników czekania w celu koordynacji kanałów Nie wpływa to na wykonywanie we własnym kanale. SETM zachowuje obowiązywanie po resecie kanału i NC-Start.		
CLEARM(): [1]	Predefiniowana procedura do kasowania jednego lub wielu znaczników czekania do koordynacji kanałów Nie wpływa to na wykonywanie we własnym kanale. CLEARM() kasuje wszystkie znaczniki czekania w kanale. CLEARM(0) kasuje tylko znaczniki czekania o wartości "0". CLEARM zachowuje obowiązywanie po resecie kanału i NC-Start.		
<ChanNr>:	Numer kanału Numer własnego kanału nie musi być podawany.		
	Typ:	INT	
<Prog>:	Absolutne lub względne podanie ścieżki (**opcjonalnie**) + nazwy programu		
	Typ:	STRING	
	W celu podania ścieżki patrz: **Literatura** Podręcznik programowania, Przygotowanie do pracy; rozdział „Zarządzanie plikami i programami" > „Pamięć programów" > „Adresowanie plików pamięci programów" (Strona 225)		
<AckMode>:	Tryb kwitowania (**opcjonalnie**)		
	Typ:	CHAR	
	Wartości:	"N"	Bez kwitowania Wykonywanie programu jest kontynuowane po wysłaniu rozkazu. Nadawca nie jest powiadamiany, gdy rozkazu nie można pomyślnie wykonać.
		"S"	Kwitowanie synchroniczne Wykonywanie programu jest tak długo zatrzymywane, aż komponent odbierający pokwituje rozkaz. Przy pozytywnym pokwitowaniu jest wykonywane następne polecenie. Przy negatywnym pokwitowaniu jest wyprowadzany komunikat błędu.
<MarkNr>:	Numer znacznika czekania **Wskazówka** W systemie wielokanałowym dostępnych jest maksymalnie 100 znaczników czekania (znacznik czekania 0 ... 99). W systemie jednokanałowym dostępny jest tylko znacznik czekania 0.		
1) Do komunikacji użytkownika i / lub do koordynacji w kanałach, znaczniki czekania mogą być wprowadzane za pomocą SETM / CLEARM oraz bez użycia warunkowego rozkazu czekania WAITMC. Znaczniki czekania zachowują przy tym swoje wartości przy resecie kanału i NC-Start.			

Przykłady

START za pomocą nazwy kanału z MD20000

- Parametryzowanie
```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "BEARBEITUNG"  ; Nazwa kanału 1
$MC_CHAN_NAME[ 1 ] = "ZUFUEHRUNG"   ; Nazwa kanału 2
```
- Programowanie

Kod programu	Komentarz
START(BEARBEITUNG)	; Start z kanału 1
START(ZUFUEHRUNG)	; Start z kanału 2

START poprzez lokalną „nazwę kanału" i zmienną użytkownika

Kod programu	Komentarz
DEF INT MASCHINE = 1	; Definicja zmiennej użytkownika dla kanału 1
DEF INT LADER = 2	; Definicja zmiennej użytkownika dla kanału 2
...	
START(MASCHINE)	; Start z kanału 1
START(LADER)	; Start z kanału 2

START poprzez lokalne „nazwy kanałów", zmienną użytkownika i sparametryzowane nazwy kanałów

Kod programu	Komentarz
DEF INT chanNo1	; Definicja zmiennej użytkownika dla kanału 1
DEF INT chanNo2	; Definicja zmiennej użytkownika dla kanału 2
chanNo1 = CHAN_1	; Przyporządkowanie sparametryzowanej nazwy kanału, kanał 1
chanNo2 = CHAN_2	; Przyporządkowanie sparametryzowanej nazwy kanału, kanał 2
...	
START(chanNo1)	; Start z kanału 1
START(chanNo2)	; Start z kanału 2

Polecenie INIT z absolutnym podaniem ścieżki

Wybór programu /_N_MPF_DIR/_N_ABSPAN1_MPF w kanale 2.

Kod programu
```
INIT(2,"/_N_WKS_DIR/_N_WAŁEK1_WPD/_N_ABSPAN1_MPF")
```

Polecenie INIT z nazwą programu

Wybór programu o nazwie "MYPROG". Sterowanie szuka programu na podstawie ścieżki.

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

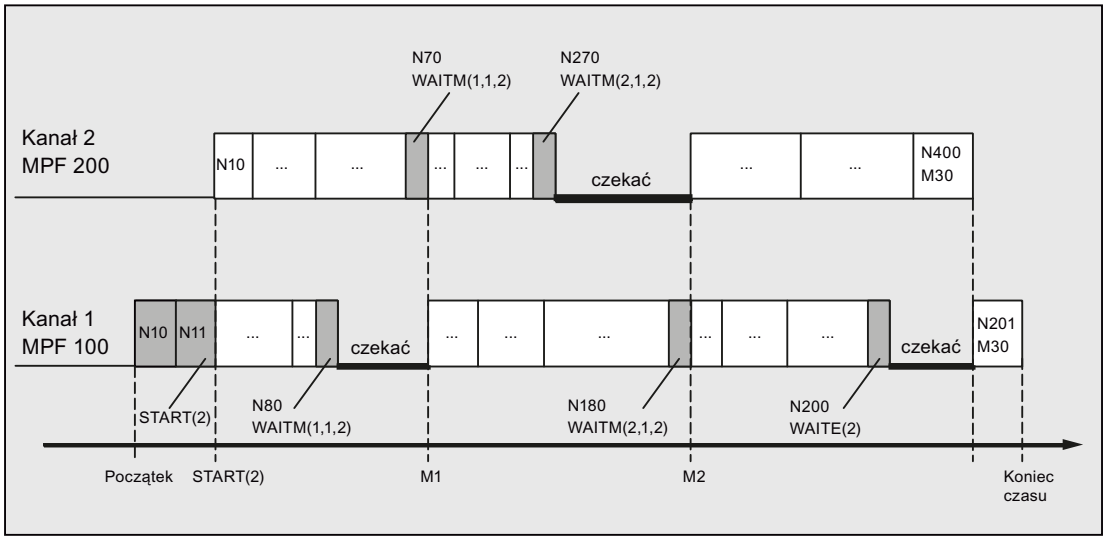
Koordynowanie programu za pomocą WAITM

- Kanał 1: Program /_N_MPF_DIR/_N_MPF100_MPF jest już wybrany i uruchomiony.

Kod programu	**Komentarz**
	; Program MPF100
N10 INIT(2,"MPF200","N")	; Wybór programu MPF200, kanał 2
N11 START(2)	; Start z kanału 2
...	
N80 WAITM(1,1,2)	; Czekanie na znacznik WAIT 1 w kanale 1 i 2
N81 ...	; Kanał 1, N81 i kanał 2, N71 ; zostaną rozpoczęte synchronicznie
...	
N180 WAITM(2,1,2)	; Czekanie na znacznik WAIT 2 w kanale 1 i 2
N181 ...	; Kanał 1, N181 i kanał 2, N271 ; zostaną rozpoczęte synchronicznie
...	
N200 WAITE(2)	; Czekanie na koniec programu w kanale 2
N201 ...	; N201 zostanie rozpoczęty dopiero po zakończeniu programu ; MPF200 uruchomiono w kanale 2
N201 M30	; Koniec programu kanał 1

- Kanał 2: W kanale 1 wybierany jest program MPF200_MPF dla kanału 2 i uruchamiany za pomocą bloku N10 i N20.

Kod programu	**Komentarz**
;$PATH=/_N_MPF_DIR	; Program MPF200
...	
N70 WAITM(1,1,2)	; Czekanie na znacznik WAIT 1 w kanale 1 i 2
N71 ...	; Kanał 1, N81 i kanał 2, N71 ; zostaną rozpoczęte synchronicznie
...	
N270 WAITM(2,1,2)	; Czekanie na znacznik WAIT 2 w kanale 1 i 2
N271 ...	; Kanał 1, N181 i kanał 2, N271 ; zostaną rozpoczęte synchronicznie
...	
N400 M30	; Koniec programu w kanale 2



Warunki brzegowe

Niesynchroniczne rozpoczęcie wykonywania kolejnych bloków po znacznikach WAIT

Podczas koordynacji kanałów za pomocą znacznika WAIT może dojść do niesynchronicznego rozpoczęcia wykonywania kolejnych bloków. To zachowanie występuje, gdy bezpośrednio przed dojściem do wspólnego znacznika WAIT w jednym z synchronizowanych kanałów zostanie wyzwolona akcja, która skutkuje skasowaniem pozostałej drogi z domyślnym repozycjonowaniem (REPOSA).

Założenie: Aktualne przyporządkowanie osi w kanałach 1 i 2

- Kanał 1: Osie X1 i U
- Kanał 2: Oś X2

Tabela 2-2 Przebieg w czasie w kanałach 1 i 2

Kanał 1	Kanał 2	Opis
...	...	Dowolne wykonywanie w kanale 1 i kanale 2
`N100 WAITM(20,1,2)`		Kanał 1: Dochodzi do znacznika `WAIT` i czeka na synchronizację z kanałem 2
Początek wykonywania GETD(U): • *Zamiana osi* • *Skasowanie pozostałej drogi* • *REPOSA* *Koniec*	`N200 GETD(U)`	Kanał 2: Żądanie osi U z kanału 1 Kanał 1: Wykonywanie `GET(U)` w tle
	`N210 WAITM(20,1,2)`	Kanał 2: Dochodzi do znacznika `WAIT`. ⇒ Synchronizacja kanałów 1 i 2 została zakończona
	`N220 G0 X2=100`	Kanał 2: Początek wykonywania N220
`N110 G0 X1=100`		Kanał 1: Przesunięty w czasie początek wykonywania N110

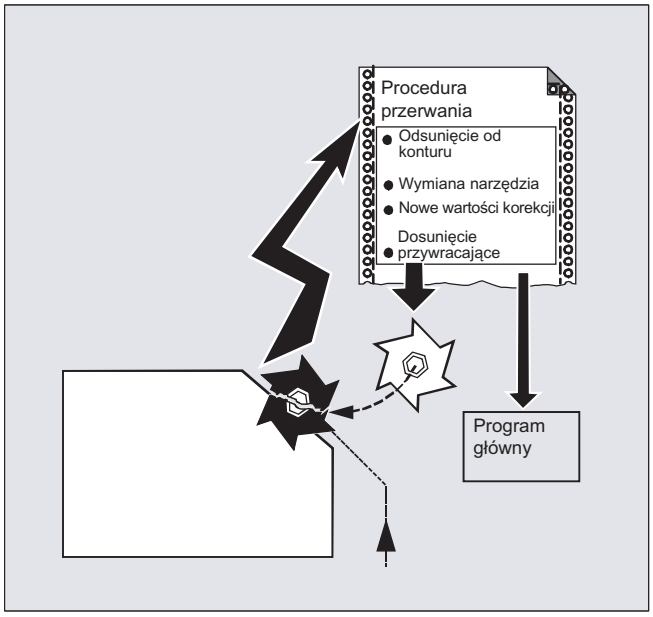
2.14 Procedura przerwania (ASUP)

2.14.1 Funkcja procedury przerwania

Uwaga

Występujące w poniższym opisie na przemian pojęcia "Podprogram asynchroniczny (ASUP)" i "Procedura przerwania" oznaczają tą samą funkcjonalność.

Wytłumaczmy funkcję procedury przerwania na podstawie typowego przykładu:



Podczas obróbki następuje pęknięcie narzędzia. Przez to jest wyzwalany sygnał, który zatrzymuje bieżący przebieg obróbki i równocześnie uruchamia podprogram - tak zwaną procedurę przerwania. W tym podprogramie znajdują się wszystkie instrukcje, które w takim przypadku mają zostać wykonane.

Gdy podprogram jest wykonany (i przez to jest przywrócona gotowość do pracy), sterowanie przeskakuje z powrotem do programu głównego i kontynuuje obróbkę od miejsca przerwania - zależnie od polecenia `REPOS` - patrz "Dosunięcie przywracające do konturu (Strona 553) ").

⚠ OSTROŻNIE

Niebezpieczeństwo kolizji

Gdy w podprogramie nie jest zaprogramowane polecenie `REPOS`, wówczas pozycjonowanie następuje na koniec bloku, po którym następuje blok przerwany.

Literatura

Podręcznik działania Funkcje podstawowe; BAG, Kanał, Praca programowa, Zachowanie się przy zresetowaniu (K1), punkt: "Asynchroniczne podprogramy (ASUPs), rutyny przerwań"

2.14.2 Sporządzenie procedury przerwania

Sporządzenie procedury przerwania jako podprogramu

Procedura przerwania jest oznaczana podczas definicji jak podprogram.

Przykład:

```
Kod programu        Komentarz
PROC ABHEB_Z        ; Nazwa programu "ABHEB_Z"
N10 ...             ; Następnie następują bloki NC.
...
N50 M17             ; Na zakończenie koniec programu i powrót do programu głównego.
```

Zapis modalnych poleceń G (SAVE)

Procedura przerwania może w definicji zostać oznaczona przez `SAVE`.

Atrybut `SAVE` powoduje, że aktywne przed wywołaniem procedury przerwania modalne polecenia G są zapisywane, a po zakończeniu procedury przerwania ponownie reaktywowane (patrz " Podprogramy z mechanizmem SAVE (SAVE) (Strona 173) ").

Przez to jest możliwe kontynuowanie obróbki w miejscu przerwania po zakończeniu procedury przerwania.

Przykład:

```
Kod programu
PROC ABHEB_Z SAVE
N10 ...
...
N50 M17
```

Przyporządkowanie dalszych procedur przerwania (SETINT)

W ramach procedury przerwania mogą być programowane i uaktywniane przez dalsze procedury przerwania instrukcje `SETINT` (patrz "Przyporządkowanie i uruchomienie procedury przerwania (SETINT)" (Strona 134)). Wyzwolenie następuje dopiero przez wejście.

Literatura

Odnośnie dalszych informacji dot. sporządzania podprogramów patrz punkt "Technika podprogramów, technika makr".

2.14.3 Przyporządkowanie i uruchomienie procedury przerwania (SETINT, PRIO, BLSYNC)

Sterowanie dysponuje wieloma szybkimi wejściami (wejście 1 ... 8), z których każde wyzwala jedno przerwanie (1 ... 8). Do każdego przerwania można poprzez polecenie `SETINT` przyporządkować priorytet i procedurę przerwania. Jeżeli przerwanie zostanie wyzwolone przez ustawienie szybkiego wejścia, aktualne wykonywanie w kanale zostanie przerwane i zostanie uruchomiona procedura przerwania.

Priorytet przerwania

Jeżeli w programie obróbki zostaną wielu wejściom przyporządkowane przerwania, muszą do przerwań zostać przyporządkowane różne priorytety.

Do przerwania można przyporządkować wartość priorytetu w zakresie 1 ... 128. Wartość priorytetu 1 odpowiada najwyższemu a 128 najniższemu priorytetowi.

Składnia

```
SETINT(<n>) <NAZWA>
SETINT(<n>) PRIO=<wartość> <NAZWA>
SETINT(<n>) PRIO=<wartość> <NAZWA> BLSYNC
SETINT(<n>) PRIO=<wartość> <NAZWA> LIFTFAST
```

Znaczenie

`SETINT(<n>)`:	Sygnał przerwania <n> jest przyporządkowywany do programu NC. Przyporządkowana procedura przerwania jest uruchamiana, gdy tyko nastąpi rozpoznanie sygnału przerwania <n> == 1. **Uwaga:** Jeżeli sygnał przerwania <n> zostanie przyporządkowana do innej procedury przerwania, poprzednie przyporządkowane przestaje działać.	
`<n>`:	Numer sygnału przerwania	
	Typ:	INT
	Zakres wartości:	1 ... 32
`PRIO=`:	Priorytet przerwania (opcjonalnie)	
`<Wartość>`:	(**opcjonalnie**) Wartość priorytetu	
	Typ:	INT
	Zakres wartości:	1 ... 128 (1 ⇒ najwyższy priorytet)
`<NAZWA>`:	Nazwa programu NC (ASUP)	
`BLSYNC`:	(**opcjonalnie**) `BLSYNC` powoduje, że po wyzwoleniu przerwania najpierw następuje czekanie, aż aktualny blok zostanie wykonany. Dopiero wtedy wykonywana jest procedura przerwania.	
`LIFTFAST`:	(**opcjonalnie**) `LIFTFAST` powoduje, że po wyzwoleniu przerwania najpierw następuje szybkie wycofanie narzędzia (patrz rozdział „Szybkie cofnięcie od konturu (SETINT LIFTFAST, ALF) (Strona 137)„). Dopiero wtedy wykonywana jest procedura przerwania.	

Warunki brzegowe

Zasady przerwania

1. Dla każdego przerwania, które nie może być natychmiast wykonane, albo aktualnie jest już wykonywane, jest zapisywane kolejne żądanie przerwania. Wykraczające ponadto żądania przerwania są dla tego przerwania tracone.
2. Jeżeli aktualnie jest wykonywane przerwanie i zostanie wyzwolone kolejne przerwanie o wyższym priorytecie, przerywa ono przerwanie o niższym priorytecie. Po zakończeniu przerwania o wyższym priorytecie jest kontynuowane przerwanie i niższym priorytecie. Jeżeli podczas wykonywania przerwania o wyższym priorytecie wpłyną dalsze żądania dla przerwania o niższym priorytecie, żądanie to jest zapisywane w pamięci. Dalsze ulegną utraceniu.
3. Jeżeli aktualnie jest wykonywane przerwanie i zostanie wyzwolone kolejne przerwanie o wyższym priorytecie, przerywa ono przerwanie o niższym priorytecie. Jest wykonywane przerwanie o wyższym priorytecie. Jeżeli zostanie ponownie wyzwolone przerwanie o wyższym priorytecie, aktualne przerwanie jest przerywane i jest wykonywane przerwanie o wyższym priorytecie. Maksymalnie jest możliwych sześć aktywnych poziomów przerwania. Jeden aktualnie wykonywany poziom przerwania i pięć poziomów czekających. Dla każdego aktywnego poziomu przerwania jest zapisywane maksymalnie jedno kolejne żądanie przerwania. Wszystkie dalsze żądania przerwania są tracone. Tak samo żądania przerwania są tracone, gdy żądanie nastąpi dla dalszych poziomów przerwania (poziom przerwania ≥ 7).

Przykłady

Przykład 1: Przyporządkowanie procedur przerwania i ustalenie priorytetu

Kod programu	Komentarz
`N20 SETINT(3) PRIO=1 ABHEB_Z`	`; IF Wejście 3 == 1`
	`; THEN uruchom procedurę przerwania "ABHEB_Z"`
`N30 SETINT(2) PRIO=2 ABHEB_X`	`; IF Wejście 2 == 1`
	`; THEN uruchom procedurę przerwania "ABHEB_Z"`

Procedury przerwania wykonywane są w kolejności wartości priorytetów, gdy sygnał na wejściach występuje równocześnie: najpierw "ABHEB_Z", następnie "ABHEB_X”.

Przykład 2: Ponowne przyporządkowanie procedury przerwania

Kod programu	Komentarz
`N20 SETINT(3) PRIO=2 ABHEB_Z`	`; IF Wejście 3 == 1`
	`; THEN uruchom procedurę przerwania "ABHEB_Z"`
`...`	
`N80 SETINT(3) PRIO=1 ABHEB_X`	`; IF Wejście 3 == 1`
	`; THEN uruchom procedurę przerwania "ABHEB_Z"`

2.14.4 Wyłączenie aktywności / reaktywowanie przyporządkowania procedury przerwania (DISABLE, ENABLE)

Aktywność instrukcji SETINT może zostać wyłączona przy pomocy DISABLE i ponownie włączona przy pomocy ENABLE, bez utraty przyporządkowania wejście → procedura przerwania.

Składnia

```
DISABLE(<n>)
ENABLE(<n>)
```

Znaczenie

DISABLE(<n>):	Polecenie: **Wyłączenie aktywności** przyporządkowania procedury przerwania od wejścia <n>	
ENABLE(<n>):	Polecenie: **Reaktywowanie** przyporządkowania procedury przerwania od wejścia <n>	
<n>:	Parametr: Numer sygnału przerwania	
	Typ:	INT
	Zakres wartości:	1 ... 32

Przykład

Kod programu	**Komentarz**
N20 SETINT(3) PRIO=1 ABHEB_Z	; Gdy wejście 3 przełączy, wówczas powinna zostać
	; uruchomiona procedura przerwania "ABHEB_Z”.
...	
N90 DISABLE(3)	; Aktywność instrukcji SETINT z N20 jest wyłączana.
...	
N130 ENABLE(3)	; Instrukcja SETINT z N20 jest ponownie uaktywniana.
...	

2.14.5 Skasowanie przyporządkowania procedury przerwania (CLRINT)

Zdefiniowane przy pomocy SETINT przyporządkowanie sygnału przerwania do programu NC (ASUP) może zostać skasowane przy pomocy CLRINT.

Składnia

```
CLRINT(<n>)
```

Znaczenie

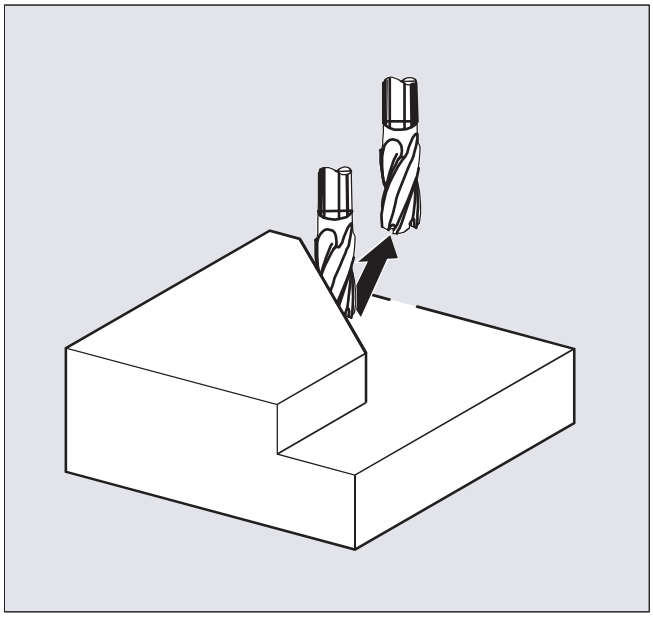
CLRINT(<n>):	Polecenie: Kasowanie przyporządkowania sygnału alarmowego <n> do zdefiniowanego za pomocą SETINT programu NC (ASUP) <n>	
<n>:	Parametr: Numer sygnału przerwania	
	Typ:	INT
	Zakres wartości:	1 ... 32

Przykład

Kod programu	Komentarz
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; Przyporządkowanie między wejściem "3" i procedurą przerwania "ABHEB_Z" jest skasowane.

2.14.6 Szybkie cofnięcie od konturu (SETINT LIFTFAST, ALF)

W przypadku instrukcji SETINT z LIFTFAST przy przełączeniu wejścia narzędzie jest odsuwane od konturu przez szybkie cofnięcie.



Dalszy przebieg jest zależny od tego, czy instrukcja SETINT oprócz LIFTFAST zawiera procedurę przerwania:

Z procedurą przerwania: **Po** szybkim cofnięciu jest wykonywana procedura przerwania.

Bez procedury przerwania: Po szybkim cofnięciu obróbka jest zatrzymywana z alarmem.

Składnia

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

Znaczenie

SETINT(<n>):	Polecenie: Wejście <n> należy przyporządkować do procedury przerwania. Przyporządkowana procedura przerwania ulega uruchomieniu, gdy przełączy się wejście <n>.	
<n>:	Parametr: Numer wejścia	
	Typ:	INT
	Zakres wartości:	1 ... 8
PRIO= :	Ustalenie priorytetu	
<wartość>:	Wartość priorytetu	
	Zakres wartości:	1 ... 128
	Priorytet 1 odpowiada najwyższemu priorytetowi.	
<NAZWA>:	Nazwa podprogramu (procedury przerwania), który ma zostać wykonany.	
LIFTFAST:	Polecenie: Szybkie cofnięcie od konturu	
ALF=… :	Polecenie: Programowany kierunek ruchu (znajduje się w bloku ruchu) Odnośnie możliwości programowania z ALF patrz temat "Kierunek ruchu przy szybkim cofnięciu od konturu (Strona 139)".	

Warunki brzegowe

Zachowanie się przy aktywnym frame z lustrzanym odbiciem

Przy określaniu kierunku odsunięcia następuje sprawdzenie, czy jest aktywny frame zawierający lustrzane odbicie. W tym przypadku przy kierunku cofnięcia odniesionym do kierunku stycznej następuje zamiana prawej i lewej strony. Składowe kierunków w kierunku narzędzia nie są poddawane lustrzanemu odbiciu. Takie zachowanie jest uaktywniane przez ustawienie MD:

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

Przykład

Wyłamane narzędzie ma zostać automatycznie zastąpione przez narzędzie siostrzane. Obróbka jest następnie kontynuowana przy użyciu nowego narzędzia.

Program główny:

Program główny	**Komentarz**
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; Gdy zastąpi zmiana sygnału na wejściu 1, narzędzie jest natychmiast w drodze cofnięcia szybkiego (kod nr 7 dla korekcji promienia narzędzia G41) odsuwane od konturu. Następnie jest wykonywana procedura przerwania "W_WECHS".
N20 G0 Z100 G17 T1 ALF=7 D1	

Program główny	**Komentarz**
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

Podprogram:

Podprogram	**Komentarz**
PROC W_WECHS SAVE	; Podprogram z zapisaniem aktualnego stanu roboczego
N10 G0 Z100 M5	; Pozycja wymiany narzędzia, stop wrzeciona
N20 T11 M6 D1 G41	; Wymiana narzędzia
N30 REPOSL RMBBL M3	; Ruch przywracający do konturu o powrót do programu głównego (programowanie następuje w jednym bloku)

2.14.7 Kierunek ruchu przy szybkim cofnięciu od konturu

Ruch wycofania

Kierunek ruchu wycofania jest określany przez następujące polecenia G:

- LFTXT
  Płaszczyzna ruchu wycofania jest określana ze stycznej do toru i kierunku narzędzia (ustawienie standardowe).
- LFWP
  Płaszczyzna ruchu wycofania jest aktywną płaszczyzną roboczą, która jest wybierana przy pomocy poleceń G G17, G18 albo G19. Kierunek ruchu wycofania jest niezależny od stycznej do toru. Przez to jest możliwe programowanie szybkiego cofnięcia równoległego do osi.
- LFPOS
  Wycofanie osi podanej przy pomocy POLFMASK / POLFMLIN do absolutnej pozycji osi zaprogramowanej przy pomocy POLF.
  ALF nie ma wpływu na kierunek wycofania dla wielu osi, jak też dla wielu osi w zależności liniowej.
  Literatura:
  Podręcznik programowania Podstawy; Rozdział: "Szybkie wycofanie się podczas nacinania gwintu"

Programowany kierunek ruchu (ALF=...)

W płaszczyźnie ruchu wycofania jest przy pomocy `ALF` programowany kierunek w krokach dyskretnych co 45 stopni.

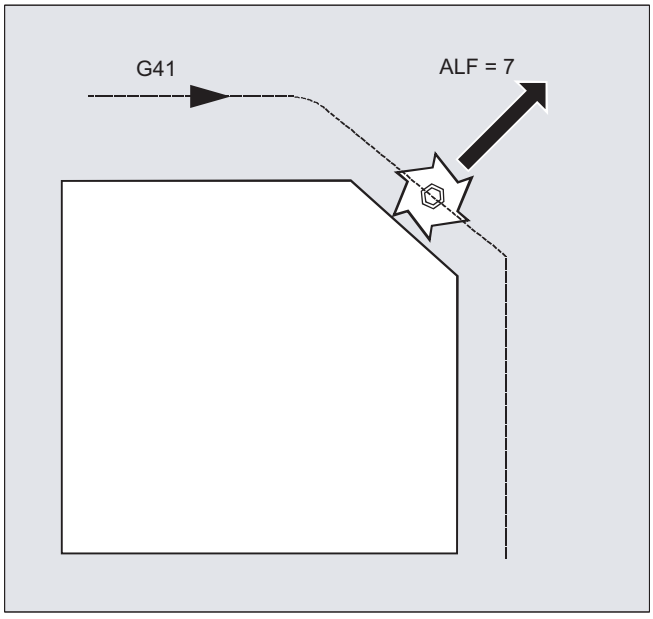
Możliwe kierunku ruchu są zapisane w sterowaniu pod specjalnymi numerami kodowymi i mogą być pod tymi numerami wywoływane.

Przykład:

Kod programu

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

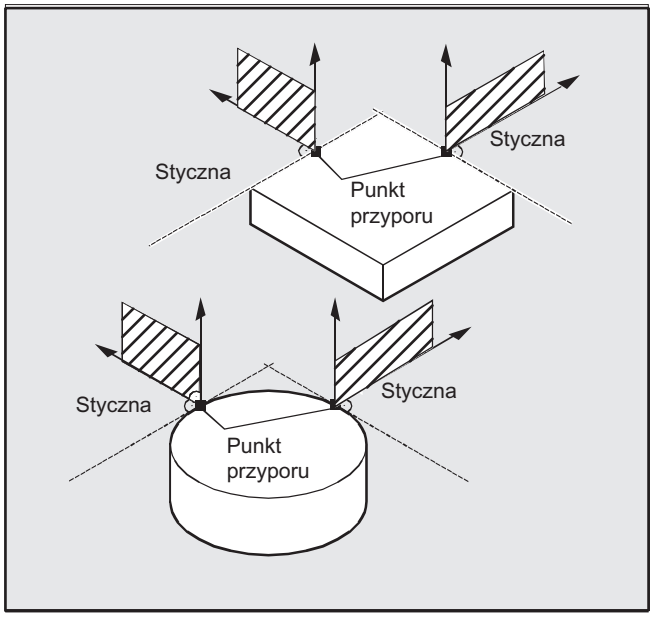
Narzędzie przy włączonym `G41` (kierunek obróbki na lewo od konturu) odsuwa się prostopadle od konturu.



Płaszczyzna odniesienia dla opisu kierunków ruchu w przypadku LFTXT

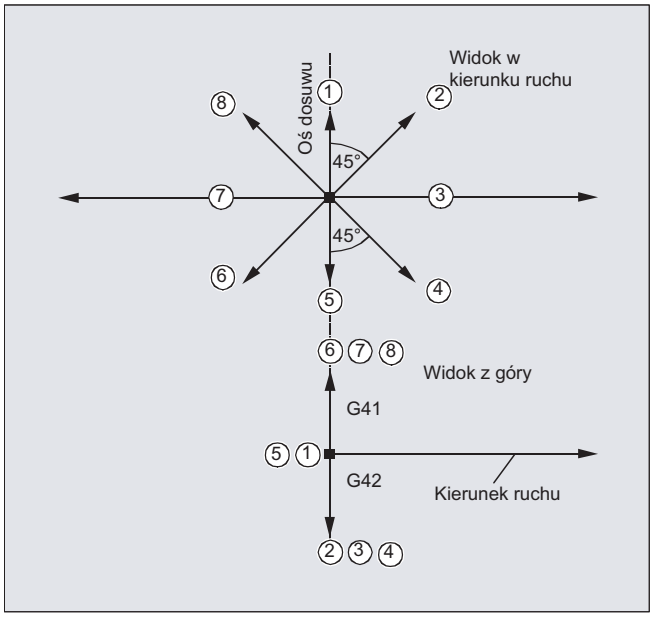
W punkcie styku narzędzia z zaprogramowanym konturem jest tworzona płaszczyzna, która służy jako odniesienie dla podania ruchu wycofania przy pomocy odpowiedniego numeru kodowego.

Płaszczyzna odniesienia jest tworzona z osi wzdłużnej narzędzia (kierunek dosuwu) i wektora na konturze, który jest prostopadły do niej i do stycznej w punkcie styku narzędzia.



Numery kodowe z kierunkami ruchu w przypadku LFTXT

Wychodząc od płaszczyzny odniesienia można na poniższym rysunku znaleźć numery kodowe z kierunkami ruchu.



Dla `ALF=1` jest ustalone wycofanie w kierunku narzędzia.

Przy pomocy `ALF=0` funkcja "szybkie cofnięcie" jest wyłączona.

⚠ OSTROŻNIE

Niebezpieczeństwo kolizji

Przy włączonej korekcji promienia narzędzia:

- przy `G41` kodowania 2, 3, 4
- przy `G42` kodowania 6, 7, 8

nie powinny być stosowane, gdyż w tych przypadkach narzędzie wykonałoby ruch w kierunku konturu i nastąpiłaby kolizja z obrabianym przedmiotem.

Numery kodowe z kierunkami ruchu w przypadku LFWP

W przypadku `LFWP` kierunek na płaszczyźnie roboczej wynika z następującego przyporządkowania:

- `G17`: Płaszczyzna X/Y
  `ALF=1`: Wycofanie w kierunku X
  `ALF=3`: wycofanie w kierunku Y
- `G18`: płaszczyzna Z/X
  `ALF=1`: Wycofanie w kierunku Z
  `ALF=3`: Wycofanie w kierunku X
- `G19`: Płaszczyzna Y/Z
  `ALF=1`: Wycofanie w kierunku Y
  `ALF=3`: Wycofanie w kierunku Z

2.14.8 Przebieg ruchów w przypadku procedur przerwania

Procedura przerwania bez LIFTFAST

Ruchy w osiach są hamowane na torze, aż do stanu zatrzymanego. Następnie jest uruchamiana procedura przerwania.

Pozycja stanu zatrzymanego jest zapisywana jako pozycja przerwania i dosunięcie do niej w ptrzypadku `REPOS` następuje za pomocą `RMI` na końcu procedury przerwania.

Procedura przerwania z LIFTFAST

Ruchy w osiach są hamowane na torze. Równocześnie jest wykonywany ruch `LIFTFAST`, jako ruch nałożony. Gdy ruch po torze i ruch `LIFTFAST` zatrzymały się, następuje uruchomienie procedury przerwania.

Jako pozycja przerwania jest zapisywana pozycja na konturze, w której jest uruchamiany ruch `LIFTFAST`, a przez to nastąpiło wyjście z toru.

Procedura przerwania z `LIFTFAST` i `ALF=0` zachowuje się identycznie jak procedura przerwania bez `LIFTFAST`.

Uwaga

Wielkość bezwzględna, o którą osie geometryczne odsuwają się od konturu przy szybkim cofnięciu, jest ustawiana przez daną maszynową.

2.15 Zamiana osi, zamiana wrzecion (RELEASE, GET, GETD)

Jedna albo wiele osi lub wrzecion może zawsze interpolować tylko w jednym kanale. Jeżeli oś musi pracować na przemian w dwóch różnych kanałach (np. w wymieniaczu palet), wówczas musi ona najpierw zostać udostępniona w aktualnym kanale, a następnie przejęta do drugiego kanału. Oś jest wymieniana między kanałami.

Rozszerzenia zamiany osi

Oś/wrzeciono może być poddawane zamianie z zatrzymaniem przebiegu wyprzedzającego i synchronizacją między przebiegiem wyprzedzającym i przebiegiem głównym lub również bez zatrzymania przebiegu wyprzedzającego. Poza tym zamiana osi jest również możliwa przez

- Obrót pojemnika osi AXCTSWE lub AXCTWED przy pomocy implicite `GET/GETD`.
- Frame z rotacją, gdy ta oś jest przez to powiązana z innymi osiami.
- Akcje synchroniczne, patrz akcje synchroniczne ruchu, "zamiana osi `RELEASE`, `GET`".

Producent maszyny

Należy przestrzegać danych producenta maszyny. Poprzez projektowane dane maszynowe oś musi być w celu zamiany osi jednoznacznie zdefiniowana we wszystkich kanałach, a zachowanie się pod względem zamiany osi jest również przez dane maszynowe ustawiane zmiennie.

Składnia

`RELEASE(nazwa osi, nazwa osi, ...)` lub `RELEASE(S1)`

`GET(nazwa osi, nazwa osi, ...)` lub `GET(S2)`

`GETD(nazwa osi, nazwa osi, ...)` lub `GETD(S3)`

Przy pomocy GETD (GET Directly) oś jest pobierana bezpośrednio z innego kanału. Oznacza to, że do tego GETD nie musi być programowany pasujący RELEASE w innym kanale. Oznacza to jednak również, że teraz musi zostać zbudowana inna komunikacja kanałowa (np. znaczniki Wait).

Znaczenie

`RELEASE (nazwa osi, nazwa osi, …):`	Zwolnienie osi
`GET (nazwa osi, nazwa osi, …):`	Przejęcie osi
`GETD (nazwa osi, nazwa osi, …):`	Bezpośrednie przejęcie osi
`Nazwa osi:`	Przyporządkowanie osi w systemie: AX1, AX2, ... lub podanie nazw osi maszyny
`RELEASE(S1):`	Zwolnienie wrzeciona S1, S2, …
`GET(S2):`	Przejęcie wrzeciona S1, S2, ...
`GETD(S3):`	Bezpośrednie przejęcie wrzeciona S1, S2, …

Zażądanie GET bez zatrzymania przebiegu wyprzedzającego

Jeżeli po zażądaniu GET **bez** zatrzymania przebiegu wyprzedzającego oś zostanie ponownie zwolniona przy pomocy `RELEASE(oś)` lub `WAITP(oś)`, wówczas następne `GET` prowadzi do `GET` **z** zatrzymaniem przebiegu wyprzedzającego.

⚠ OSTROŻNIE

Zmiana przyporządkowania osi

Oś wzgl. wrzeciono przejęte przy pomocy GET pozostaje przyporządkowane do tego kanału również po naciśnięciu RESET albo z programu.

Przy ponownym uruchomieniu programu przyporządkowanie zamienionych osi lub wrzecion musi nastąpić programowo, w przypadku gdy oś jest potrzebna w swoim podstawowym kanale.

W przypadku POWER ON zostaje ona przyporządkowana do kanału zapisanego w danej maszynowej.

Przykłady

Przykład 1: Wymiana osi pomiędzy kanałami

6 osi kanału 1 używa się do obróbki: 1., 2., 3. i 4. osi.
5. i 6. oś są używane w kanale 2 do wymiany obrabianego przedmiotu.

Oś 2 ma być możliwa do zamiany między obydwoma kanałami, a po załączeniu zasilania być przyporządkowana do kanału 1.

Program „MAIN“ w kanale 1:

Kod programu	**Komentarz**
`INIT (2,"TAUSCH2")`	`; Należy wybrać program TAUSCH2 w kanale 2.`
`N… START (2)`	`; Start programu w kanale 2.`
`N… GET (AX2)`	`; Przejęcie osi AX2.`
`...`	
`N… RELEASE (AX2)`	`; Zwolnienie osi AX2.`
`N… WAITM (1,1,2)`	`; Czekanie na znacznik WAIT w kanale 1 i 2 w celu synchronizacji w obydwu kanałach.`
`...`	`; Dalszy przebieg po zamianie osi.`
`N… M30`	

Program "TAUSCH2" w kanale 2:

Programowanie	**Komentarz**
`N… RELEASE(AX2)`	
`N160 WAITM(1,1,2)`	`; Czekanie na znacznik WAIT w kanale 1 i 2 w celu synchronizacji w obydwu kanałach.`
`N150 GET(AX2)`	`; Przejęcie osi AX2.`
`...`	`; Dalszy przebieg po zamianie osi.`
`N… M30`	

Przykład 2: Zamiana osi bez synchronizacji

Gdy oś nie musi być synchronizowana, GET nie wytwarza zatrzymania przebiegu wyprzedzającego.

Programowanie	**Komentarz**
N01 G0 X0	
N02 RELEASE(AX5)	
N03 G64 X10	
N04 X20	
N05 GET(AX5)	; Gdy synchronizacja nie jest konieczna, nie jest to wykonywalny blok.
N06 G01 F5000	; Nie jest to wykonywalny blok.
N07 X20	; Nie jest to wykonywalny blok, ponieważ pozycja X jest taka jak w N04.
N08 X30	; Pierwszy wykonywalny blok po N05.
...	

Przykład 3: Uaktywnienie zamiany osi bez zatrzymania przebiegu wyprzedzającego

Warunek: Zamiana osi bez zatrzymania przebiegu wyprzedzającego musi zostać zaprogramowana przez daną maszynową.

Programowanie	**Komentarz**
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	; Oś B staje się osią neutralną.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; Oś nie wyzwala zatrzymania przebiegu wyprzedzającego/REORG.
N041 G4 F2	
N050 M5	
N099 M30	

Jeżeli jest wykonywany ruch wrzeciona lub osi B bezpośrednio po bloku N023 jako **osi PLC**, np. na 180 stopni i z powrotem na 1 stopień, wówczas ta oś staje się z powrotem osią neutralną i w bloku `N40` nie wyzwala zatrzymania przebiegu wyprzedzającego.

Dodatkowe informacje

Warunki zamiany osi

- Oś musi być przez dane maszynowe zdefiniowana we wszystkich kanałach, które chcą tej osi używać.
- Poprzez specyficzną dla **osi** daną maszynową musi być ustalone, któremu kanałowi ma być przyporządkowana oś po załączeniu zasilania.

Opis

Zwolnienie osi: RELEASE

Przy zwolnieniu osi należy przestrzegać:

1. Oś nie może uczestniczyć w żadnej transformacji.
2. Przy sprzężeniach osi (sterowanie styczne), muszą zostać zwolnione wszystkie osie zespołu.
3. Konkurującej osi pozycjonowania nie wolno w tym stanie zamienić.
4. W przypadku osi wiodącej gantry są zamieniane również wszystkie osie nadążne.
5. W przypadku sprzężeń osi (nadążanie, sprzężenie wartości wiodącej, przekładnia elektroniczna) można zwolnić tylko oś wiodącą systemu.

Przejęcie osi GET

Przy pomocy tego polecenia przeprowadza się właściwą zamianę osi. Odpowiedzialność za oś leży całkowicie w tym kanale, w którym zaprogramowano polecenie.

Skutki GET:

Zamiana osi z synchronizacją:

Oś musi być synchronizowana zawsze wtedy, gdy w międzyczasie była przyporządkowana do innego kanału albo do PLC, a przed GET nie nastąpiła synchronizacja przez "WAITP", G74 albo skasowanie pozostałej drogi.

- Zatrzymanie przebiegu wyprzedzającego następuje (jak przy STOPRE).
- Obróbka zostaje przerwana na tak długo, aż zamiana zostanie całkowicie wykonana.

Automatyczne "GET"

Jeżeli oś jest zasadniczo dostępna w kanale, ale obecnie nie występuje jako "oś kanału", następuje automatyczne wykonanie GET. Gdy oś(sie) jest(są) już zsynchronizowana(e), zatrzymanie przebiegu wyprzedzającego nie jest wytwarzane.

Ustawienie zmiennego zachowania się przy zamianie osi

Moment oddania osi daje się ustawić poprzez daną maszynową jak następuje:

- Automatyczna zamiana odbywa się między dwoma kanałami również wtedy, gdy oś została przez WAITP doprowadzona do stanu neutralnego (zachowanie się jak dotychczas)
- Przy zażądaniu obrotu zasobnika osi wszystkie przyporządkowywalne do kanału wykonującego osie zasobnika są przy pomocy następującego domyślnego GET lub GETD pobierane do kanału. Następna zamiana osi jest dozwolona dopiero po zakończeniu obrotu zasobnika osi.
- Po wstawionym bloku pośrednim w przebiegu głównym następuje sprawdzenie, czy reorganizacja jest konieczna czy nie. Tylko gdy stany osi tego bloku **nie** są zgodne z aktualnymi stanami osi, jest wymagana reorganizacja.
- Zamiast bloku GET z zatrzymaniem przebiegu wyprzedzającego i synchronizacją między przebiegiem wyprzedzającym i głównym zamiana osi może nastąpić również bez zatrzymania przebiegu wyprzedzającego. Jest wówczas tworzony tylko blok pośredni z zażądaniem GET. W przebiegu głównym przy wykonywaniu tego bloku następuje sprawdzenie, czy stany osi w bloku są zgodne z aktualnymi stanami osi.

Dalsze informacje dot. działania zamiany osi i wrzecion patrz
Podręcznik działania Funkcje rozszerzone; BAG, kanały, zamiana osi (K5).

2.16 Przekazanie osi do innego kanału (AXTOCHAN)

Przy pomocy polecenia językowego AXTOCHAN może zostać zażądana oś przekazana do innego kanału. Oś może zostać pobrana do odpowiedniego kanału zarówno z programu obróbki NC, jak też z akcji synchronicznej.

Składnia

```
AXTOCHAN(nazwa osi,numer kanału[,nazwa osi,numer kanału[,...]])
```

Znaczenie

Element	Opis
AXTOCHAN:	Zażądanie osi dla określonego kanału
Nazwa osi:	Przyporządkowanie osi w systemie: X, Y, ... lub podanie nazw uczestniczących osi maszyny. Wykonującym kanałem nie musi być własny kanał i nie musi to być też ten kanał, który posiada aktualnie prawo interpolacji dla osi
numer kanału:	Numer kanału, któremu oś ma zostać przyporządkowana

Uwaga

Konkurująca oś pozycjonowania i oś kontrolowana wyłącznie przez PLC

Oś PLC nie może jako konkurująca oś pozycjonowania zmienić kanału. Oś kontrolowana wyłącznie przez PLC nie może zostać przyporządkowana do programu NC.

Literatura:
Podręcznik działania Funkcje rozszerzone; Osie pozycjonowania (P2)

Przykład

AXTOCHAN w programie NC

Osie X i Y są znane w 1. kanale i w 2. kanale. Aktualnie kanał 1 ma prawo interpolacji i w kanale 1 jest uruchamiany następujący program:

```
Kod programu                    Komentarz
N110 AXTOCHAN(Y,2)              ; Przesunięcie osi Y do 2. kanału.
N111 M0
N120 AXTOCHAN(Y,1)              ; Sprowadzenie osi Y z powrotem (neutralna).
N121 M0
N130 AXTOCHAN(Y,2,X,2)          ; Przesunięcie osi X do 2. kanału (osie neutralne).
N131 M0
N140 AXTOCHAN(Y,2)              ; Przesunięcie osi Y do 2. kanału (program NC).
N141 M0
```

Dodatkowe informacje

AXTOCHAN w programie NC

Przy tym jest tylko przy zażądaniu osi dla programu NC we własnym kanale przeprowadzane `GET`, a przez to również następuje czekanie na rzeczywistą zmianę stanu. Jeżeli oś zostanie zażądana dla innego kanału albo ma stać się osią neutralną we własnym kanale, wówczas jest tylko odpowiednio generowane żądanie.

AXTOCHAN z akcji synchronicznej

Gdy oś zostanie zażądana dla własnego kanału, wówczas `AXTOCHAN` z akcji synchronicznej zostanie odwzorowane na `GET` z akcji synchronicznej. W tym przypadku oś staje się neutralna przy pierwszym zażądaniu dla własnego kanału. Przy drugim zażądaniu oś jest przyporządkowywana do programu NC analogicznie do zażądania GET w programie NC. Odnośnie zażądania GET z akcji synchronicznej patrz punkt "Akcje synchroniczne ruchu".

2.17 Ustawienie działania danych maszynowych (NEWCONF)

Przy pomocy polecenia NEWCONF jest ustawiane działanie wszystkich danych maszynowych. Funkcja może zostać również uaktywniona na interfejsie obsługi HMI przez naciśnięcie przycisku programowego "Ustaw działanie MD".

Przy wykonywaniu funkcji NEWCONF następuje domyślne zatrzymanie przebiegu wyprzedzającego, tzn. ruch po torze jest przerywany.

Składnia

```
NEWCONF
```

Znaczenie

NEWCONF:	Polecenie do ustawienia działania wszystkich danych maszynowych stopnia działania "NEW_CONFIG"

Wykonanie NEWCONF z programu obróbki poza granice kanału

Jeżeli osiowe dane maszynowe zostaną zmienione z programu obróbki, a następnie uaktywnione przy pomocy NEWCONF, wówczas polecenie NEWCONF uaktywnia tylko te dane maszynowe, które powodują zmiany w kanale programu obróbki.

Uwaga

Aby spowodować niezawodne działanie wszystkich zmian, polecenie NEWCONF musi zostać wykonane w każdym kanale, w którym są aktualnie liczone odnośne osie albo funkcje zmienione przez dane maszynowe.

Przy NEWCONF nie są ustawiane żadne osiowe dane maszynowe.

Dla osi kontrolowanych przez PLC musi zostać wykonany RESET osiowy.

Przykład

Obróbka frezarska: Obróbka w pozycji wiercenia przy pomocy różnych technologii

```
Kod programu                          Komentarz
N10 $MA_CONTOUR_TOL[AX]=1.0           ; Zmiana danej maszynowej.
N20 NEWCONF                           ; Ustawienie działania danych maszynowych.
...
```

2.18 Zapisanie pliku (WRITE)

Przy pomocy polecenia `WRITE` bloki/dane z programu NC można zapisać na końcu pliku znajdującego się w pasywnym systemie plików albo w zewnętrznej pamięci programów (plik protokołu). Może to być również właśnie wykonywany program.

Uwaga

Plik, w którym ma nastąpić zapisanie przez polecenie `WRITE`, ulega utworzeniu, jeżeli nie istnieje w pamięci programów.

Warunek

Aktualnie ustawiony poziom dostępu musi być równy albo większy od prawa WRITE pliku. Jeżeli tak nie jest, dostęp jest odrzucany z komunikatem błędu (zwracana wartość zmiennej błędu = 13).

Składnia

```
DEF INT <błąd>
...
WRITE(<błąd>,"<nazwa pliku>"/<urządzenie zewnętrzne>","<blok/dane>")
```

Znaczenie

`WRITE`:	Polecenie do dołączenia bloku lub danych na końcu podanego pliku		
`<błąd>`:	**Parametr 1:** Zmienna do zwrotu wartości błędu		
	Typ:	INT	
	Wartość:	0	Nie ma błędu
		1	Ścieżka niedozwolona
		2	Ścieżka nie znaleziona
		3	Plik nie znaleziony
		4	Nieprawidłowy typ pliku
		10	Plik jest zapełniony
		11	Plik jest używany
		12	Brak wolnych zasobów
		13	Brak praw dostępu
		14	Brakujące lub nieudane EXTOPEN dla urządzenia wyjściowego
		15	Błąd przy zapisie na urządzeniu zewnętrznym
		16	Zaprogramowano niepoprawną ścieżkę zewnętrzna

<nazwa pliku>:	**Parametr 2:** Nazwa pliku, w którym podany blok wzgl. podane dane mają zostać dołączone	
	Typ:	STRING
	Przed właściwą nazwą pliku można podać ścieżkę absolutną. Bez podania ścieżki plik jest szukany w aktualnym katalogu (= katalogu wybranego programu). Zasady podawania ścieżek patrz "Adresowanie plików pamięci programów (Strona 225)".	
<urządzenie zewnętrzne>:	Jeżeli dane mają zostać wyprowadzone poprzez funkcję "Process DataShare" na urządzenie zewnętrzne / do pliku zewnętrznego, musi zamiast nazwy pliku zostać podany symboliczny identyfikator otwieranego zewnętrznego urządzenia/pliku.	
	Typ:	STRING
	Dalsze informacje patrz "Process DataShare - Wyprowadzenie na zewnętrzne urządzenie/plik (EXTOPEN, WRITE, EXTCLOSE) (Strona 689)". **Wskazówka** Identyfikator musi być identyczny z identyfikatorem podanym w poleceniu EXTOPEN.	
<blok/dane>:	**Parametr 3:** Plik lub dane, które mają zostać dołączone w podanym pliku.	
	Typ:	STRING

Uwaga

Przy zapisie do pasywnego systemu plików albo w zewnętrznej pamięci programów polecenie WRITE dołącza domyślnie znak "LF" (LINE FEED = nowy wiersz) na końcu wyprowadzanego łańcucha znaków.

Dla wyprowadzania na urządzenie zewnętrzne / do pliku zewnętrznego poprzez funkcję "Process DataShare" to zachowanie się nie obowiązuje. Jeżeli równocześnie ma być wprowadzane "LF", musi zostać bezpośrednio podane w wyprowadzanym łańcuchu znaków.

→ Patrz do niniejszego przykład 3: Domyślnie/bezpośrednio "LF"!

Warunki brzegowe

- **Maksymalna wielkość pliku (→ producent maszyny!)**
  Maksymalnia możliwa wielkość plików protokołowych w pasywnym systemie plików jest ustawiana w danej maszynowej:
  MD11420 $MN_LEN_PROTOCOL_FILE
  Maksymalna wielkość pliku obowiązuje dla wszystkich plików, które zostaną utworzone w pasywnym systemie plików przy pomocy polecenia WRITE. Przy przekroczeniu jest wyprowadzany komunikat błędu i blok wzgl. dane nie są zapisywane. O ile pamięć jest wystarczająca, można utworzyć nowy plik.

Przykłady

Przykład 1: Polecenie WRITE do pasywnego systemu plików bez absolutnego podania ścieżki

Kod programu	Komentarz
N10 DEF INT ERROR	; Definicja zmiennej błędu.
N20 WRITE(ERROR,"PROT","PROTOKOL Z 7.2.97")	; Zapisz tekst z "PROTOKÓŁ Z 7.2.97" do pliku _N_PROT_MPF.

Kod programu	**Komentarz**
N30 IF ERROR	; Ewaluacja błędu.
N40 MSG ("Błąd przy poleceniu WRITE:" <<ERROR)	
N50 M0	
N60 ENDIF	
...	

Przykład 2: Polecenie WRITE do pasywnego systemu plików z absolutnym podaniem ścieżki

```
Kod programu
...
WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","PROTOKOL Z 7.2.97")
...
```

Przykład 3: Domyślnie/bezpośrednio "LF"

a, zapis do pasywnego systemu plików z tworzonym domyślnym "LF"

```
Kod programu
...
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
```

Wyprowadzany wynik:

MY_STRING

MY_STRING

b, zapis do pliku zewnętrznego bez tworzonego domyślnie "LF"

```
Kod programu
...
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
```

Wyprowadzany wynik:

MY_STRINGMY_STRING

c, zapis w urządzeniu zewnętrznym z zaprogramowanym bezpośrednio "LF"

Aby uzyskać ten sam wynik co przy a, musi zostać zaprogramowane co następuje:

Kod programu

```
...
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
```

Wyprowadzany wynik:

MY_STRING

MY_STRING

2.19 Skasowanie pliku (DELETE)

Przy pomocy polecenia `DELETE` mogą być kasowane wszystkie pliki, wszystko jedno czy powstały poprzez polecenie `WRITE` czy nie. Również pliki, które zostały sporządzone pod najwyższym poziomem dostępu, można skasować przy pomocy `DELETE`.

Składnia

```
DEF INT <błąd>
DELETE(<błąd>,"<nazwa pliku>")
```

Znaczenie

`DELETE`:	Polecenie do skasowania podanego pliku		
`<błąd>`:	Zmienna do zwrotu wartości błędu		
	Typ.	INT	
	Wartość:	0	Nie ma błędu
		1	Ścieżka niedozwolona
		2	Ścieżka nie znaleziona
		3	Plik nie znaleziony
		4	Nieprawidłowy typ pliku
		11	Plik jest używany
		12	Brak wolnych zasobów
		20	Inny błąd
`<nazwa pliku>`:	Nazwa pliku do skasowania		
	Typ:	STRING	
	Przed właściwą nazwą pliku można podać ścieżkę absolutną. Bez podania ścieżki plik jest szukany w aktualnym katalogu (= katalogu wybranego programu). Zasady podawania ścieżek patrz "Adresowanie plików pamięci programów (Strona 225)".		

Przykład

```
N10 DEF INT ERROR                                   ; Definicja zmiennej błędu.
N15 STOPRE                                          ; Zatrzymanie przebiegu wyprzedzającego.
N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")        ; Skasuj plik TEST1 w katalogu podprogramu.
N30 IF ERROR                                        ; Ewaluacja błędu.
N40 MSG ("Błąd przy poleceniu DELETE:" <<ERROR)
N50 M0
N60 ENDIF
```

2.20 Odczyt wierszy w pliku (READ)

Polecenie `READ` czyta w podanym pliku jeden albo wiele wierszy i zapisuje przeczytane informacje w tablicy typu STRING. Każdy przeczytany wiersz zajmuje w tej tablicy jeden element.

Warunek

Aktualnie ustawiony poziom dostępu musi być równy albo większy od uprawnienia READ pliku. Jeżeli tak nie jest, dostęp jest odrzucany z komunikatem błędu (zwracana wartość zmiennej błędu = 13).

Składnia

```
DEF INT <błąd>
DEF STRING[<długość łańcucha znaków>] <wynik>[<n>,<m>]
READ(<błąd>,"<nazwa pliku>",<wiersz początkowy>,<liczba
wierszy>,<wynik>)
```

Znaczenie

`READ`:	Polecenie do odczytu wierszy podanego pliku i do zapisania tych wierszy w tablicy zmiennych.		
`<błąd>`:	Zmienna do zwrotu wartości błędu (parametr Call-By-Reference)		
	Typ.	INT	
	Wartość:	0	Nie ma błędu
		1	Ścieżka niedozwolona
		2	Ścieżka nie znaleziona
		3	Plik nie znaleziony
		4	Nieprawidłowy typ pliku
		11	Plik jest używany
		13	Prawa dostępu niewystarczające
		21	Wiersza nie ma (parametr `<wiersz początkowy>` lub `<liczba wierszy>` większy, niż liczba wierszy w podanym pliku).
		22	Długość tablicy zmiennych wynikowych (`<wynik>`) jest za mała.
		23	Zakres wiersza za duży (parametr `<liczba wierszy>` został wybrany tak duży, że odczyt następuje poza końcem pliku).
`<nazwa pliku>`:	Nazwa czytanego pliku (parametr Call-By-Value)		
	Typ:	STRING	
	Przed właściwą nazwą pliku można podać ścieżkę absolutną. Bez podania ścieżki plik jest szukany w aktualnym katalogu (= katalogu wybranego programu). Zasady podawania ścieżek patrz "Adresowanie plików pamięci programów (Strona 225)".		

<wiersz początkowy>:	Wiersz początkowy czytanej części pliku (parametr Call-By-Value)		
	Typ:	INT	
	Wartość:	0	Jest czytana liczba wierszy przed końcem pliku podana przy pomocy parametru <Liczba wierszy>.
		1 … n	Numer pierwszego wiersza do odczytania.
<liczba wierszy>:	Liczba czytanych wierszy (parametr Call-By-Value)		
	Typ:	INT	
<wynik>:	Zmienna wynikowa (parametr Call-By-Reference) Tablica zmiennych, w której jest zapisywany czytany tekst.		
	Typ:	STRING (maks. długość: 255)	
	Jeżeli w parametrze <liczba wierszy> jest podanych mniej wierszy, niż wynosi wielkość tablicy zmiennych wynikowych [<n>,<m>], wówczas pozostałe elementy tablicy nie są zmieniane. Zakończenie wiersza przez znak sterujący "LF" (line feed) albo "CR LF" (Carrige Return Line Feed) **nie** jest zapisywane w zmiennej wynikowej. Przeczytane wiersze są obcinane, gdy wiersz jest dłuższy, niż zdefiniowana długość łańcucha znaków. Nie następuje komunikat błędu.		

Uwaga

Pliki binarne nie mogą być wczytywane. Jest wyprowadzany błąd "Nieprawidłowy typ pliku" (zwracana wartość zmiennej błędu = 4). Następujące typy plików nie dają się czytać: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Przykład

Kod programu	**Komentarz**
N10 DEF INT ERROR	; Definicja zmiennej błędu.
N20 DEF STRING[255] RESULT[5]	; Definicja zmiennej wynikowej.
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF", 1,5,RESULT)	; Nazwa pliku z identyfikatorem domeny, rozszerzeniem pliku i podaniem ścieżki.
N40 IF ERROR <>0	; Ewaluacja błędu.
N50 MSG("BŁĄD"<<ERROR<<"W PRZYPADKU POLECENIA READ")	
N60 M0	
N70 ENDIF	
...	

2.21 Sprawdzenie istnienia pliku (ISFILE)

Za pomocą polecenia `ISFILE` można sprawdzić, czy plik jest w pamięci programów.

Składnia

```
<Wynik>=ISFILE("<Nazwa pliku>")
```

Znaczenie

`ISFILE`:	Polecenie do sprawdzenia istnienia pliku		
`<Nazwa pliku>`:	Nazwa pliku, którego istnienie ma zostać sprawdzone.		
	Typ:	STRING	
	Przed właściwą nazwą pliku można podać ścieżkę absolutną. Bez podania ścieżki plik jest szukany w aktualnym katalogu (= katalogu wybranego programu). Zasady podawania ścieżek patrz "Adresowanie plików pamięci programów (Strona 225)".		
`<Wynik>`:	Zmienna wynikowa do przypisania wyniku kontroli		
	Typ.	BOOL	
	Wartość:	TRUE	Plik istnieje
		FALSE	Plik nie istnieje

Przykłady

Przykład 1

Kod programu	**Komentarz**
`N10 DEF BOOL RESULT`	`; Definicja zmiennej wynikowej.`
`N20 RESULT=ISFILE("TESTFILE")`	
`N30 IF(RESULT==FALSE)`	
`N40  MSG("PLIK NIE ISTNIEJE")`	
`N50  M0`	
`N60 ENDIF`	
`...`	

Przykład 2

Kod programu	**Komentarz**
`N10 DEF BOOL RESULT`	`; Definicja zmiennej wynikowej.`
`N20 RESULT=ISFILE("TESTFILE")`	
`N30 IF(NOT ISFILE("TESTFILE"))`	
`N40  MSG("PLIK NIE ISTNIEJE")`	
`N50  M0`	
`N60 ENDIF`	
`...`	

2.22 Odczyt informacji o pliku (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Za pomocą poleceń `FILEDATE`, `FILETIME`, `FILESIZE`, `FILESTAT` i `FILEINFO` można odczytać określone informacje o pliku, jak data / godzina dostępu w celu zapisu, aktualna wielkość pliku, status pliku lub suma tych informacji.

Warunek

Aktualnie ustawiony poziom dostępu musi być równy albo większy od praw dostępu katalogu nadrzędnego. Jeżeli tak nie jest, dostęp jest odrzucany z komunikatem błędu (zwracana wartość zmiennej błędu = 13).

Składnia

```
FILE....(<Błąd>,"<Nazwa pliku>",<Wynik>)
```

Znaczenie

`FILEDATE`:	Podaje **datę** ostatniego dostępu do pliku w celu zapisu		
`FILETIME`:	Podaje **czas zegarowy** ostatniego dostępu do pliku w celu zapisu		
`FILESIZE`:	Podaje **aktualną wielkość** pliku		
`FILESTAT`:	Podaje **status** pliku odnośnie następujących **praw**: • Odczyt (r: read) • Zapis (w: write) • Wykonanie (x: execute) • Wyświetlanie (s: show) • Kasowanie (d: delete) **Wskazówka:** Te poziomy ochrony są specjalnymi właściwościami pasywnego systemu plików. Przy dostępie do zewnętrznej pamięci programów `FILESTAT` podaje tylko domyślne prawa dostępu (77777).		
`FILEINFO`:	Podaje dla pliku **sumę informacji**, które można odczytać za pomocą `FILEDATE`, `FILETIME`, `FILESIZE` i `FILESTAT`.		
`<Błąd>`:	Zmienna do zwrotu wartości błędu (parametr Call-By-Reference)		
	Typ.	VAR INT	
	Wartość:	0	Brak błędu
		1	Ścieżka niedozwolona
		2	Ścieżka nie znaleziona
		3	Plik nie znaleziony
		4	Nieprawidłowy typ pliku
		13	Prawa dostępu niewystarczające
		22	Długość łańcucha znaków zmiennej wynikowej (`<Wynik>`) jest za mała.

<Nazwa pliku>:	Nazwa pliku, z którego mają zostać odczytane informacje o pliku			
	Typ:	CHAR[160]		
	Przed właściwą nazwą pliku można podać ścieżkę absolutną. Bez podania ścieżki plik jest szukany w aktualnym katalogu (= katalogu wybranego programu). Zasady podawania ścieżek patrz "Adresowanie plików pamięci programów (Strona 225)".			
<Wynik>:	Zmienna wynikowa (Parametr Call-By-Reference) Zmienna, w której jest zapisywana zażądana informacja o pliku.			
	Typ:	VAR CHAR[8]	przy	FILEDATE Format: "dd.mm.yy"
		VAR CHAR[8]	przy	FILETIME Format: "hh:mm.ss"
		VAR INT	przy	FILESIZE Wielkość pliku jest wyprowadzana w bajtach.
		VAR CHAR[5]	przy	FILESTAT Format: "rwxsd" (r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
		VAR CHAR[32]	przy	FILEINFO Format: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

Przykład

Kod programu	**Komentarz**
N10 DEF INT ERROR	; Definicja zmiennej błędu.
N20 STRING[32] RESULT	; Definicja zmiennej wynikowej.
N30 FILEINFO(ERROR,"/_N_MPF_DIR/_N_TESTFILE_MPF",RESULT)	; Nazwa pliku z identyfikatorem domeny, rozszerzeniem pliku i podaniem ścieżki.
N40 IF ERROR <> 0	; Ocena błędu
N50 MSG("BLAD"<<ERROR<<"DLA POLECENIA FILEINFO")	
N60 M0	
N70 ENDIF	
...	

Przykład mógłby w zmiennej wynikowej RESULT dać np. następujący wynik:

"77777 12345678 26.05.00 13:51:30"

2.23 Zaokrąglenie do góry (ROUNDUP)

Przy pomocy funkcji "ROUNDUP" wprowadzane wartości typu REAL (liczby ułamkowe z kropką dziesiętną) mogą być zaokrąglane do góry do najbliższej liczby całkowitej.

Składnia

```
ROUNDUP(<wartość>)
```

Znaczenie

ROUNDUP:	Polecenie zaokrąglenia wprowadzonej wartości do góry
<wartość>:	Wprowadzona wartość typu REAL

Uwaga

Wprowadzane wartości typu INTEGER (liczba całkowita) są zwracane bez zmiany.

Przykłady

Przykład 1: Różne wprowadzane wartości i wyniki ich zaokrąglenia

Przykład	Wynik zaokrąglenia
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

Przykład 2: ROUNDUP w programie NC

Kod programu

```
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 podprogramów

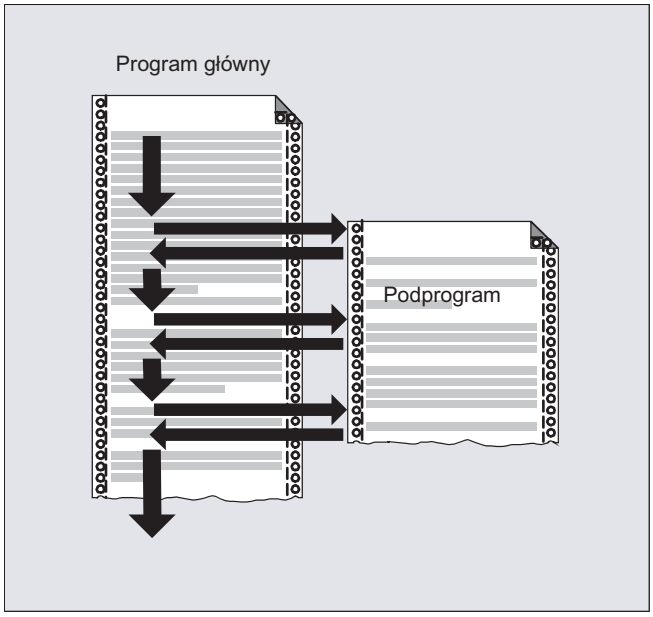
2.24.1 Informacje ogólnie

2.24.1.1 Podprogram

Określenie "podprogram" pochodzi jeszcze z okresu, gdy programy obróbki były na stałe podzielone na programy główne i podprogramy. Programami głównymi były przy tym programy obróbki, które były w sterowaniu wybierane do wykonania i następnie uruchamiane. Podprogramami były programy obróbki, które były wywoływane z programu głównego.

Tego stałego podziału już nie ma w dzisiejszym języku SINUMERIK NC. Każdy program obróbki może zasadniczo zostać wybrany i uruchomiony jako program główny, albo zostać wywołany z innego programu obróbki jako podprogram.

Stąd dalej jako podprogram jest określany program obróbki, który jest wywoływany z innego programu obróbki.



Zastosowanie

Tak, jak we wszystkich językach programowania wyższego poziomu, również w języku NC podprogramy są stosowane do tego, by części programu, które są wykorzystywane wielokrotnie, wyłączyć jako programy stanowiące oddzielną całość.

Podprogramy mają następujące zalety:

- Zwiększają przejrzystość i czytelność programów
- Zwiększają jakość dzięki ponownemu stosowaniu przetestowanych części programów

- Umożliwiają tworzenie specyficznych bibliotek obróbkowych
- Oszczędzają miejsce w pamięci

2.24.1.2 Nazwy podprogramów

Reguły nazewnictwa

Nazwę podprogramu można dowolnie wybierać zgodnie z następującymi zasadami:

- Dozwolone znaki:
  - Litery: A ... Z, a ... z
  - Cyfry: 0 ... 9
  - Podkreślenie: _
- Pierwsze dwa znaki powinny składać się z dwóch liter lub podkreślenia poprzedzonego literą.

  Uwaga

  Gdy ten warunek zostanie spełniony, wówczas program NC może zostać wywołany jako podprogram z innego programu wyłącznie poprzez podanie nazwy programu. Jeżeli natomiast nazwa programu rozpoczyna się od cyfr, wówczas wywołanie podprogramu jest możliwe tylko za pomocą instrukcji CALL.

- Maksymalna długość: 24 znaki

Uwaga

Duże / małe litery

W języku NC SINUMERIK **nie** rozróżnia się pisania dużymi i małymi literami.

Uwaga

Niedozwolone nazwy programów

Aby uniknąć problemów z aplikacjami systemu Windows, **nie** można używać następujących nazw programów:

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

Rozszerzenia wewnętrzne sterowania

Nazwa podprogramu nadana przy sporządzaniu programu jest wewnętrznie w sterowaniu rozszerzana o prefiks i postfiks:

- Prefiks: `_N_`
- Postfiks: `_SPF`

Zastosowanie nazwy programu

Przy stosowaniu nazwy programu, np. przy wywołaniu podprogramu, są możliwe wszystkie kombinacje prefiksu, nazwy programu i postfiksu.

Przykład:

Podprogram o nazwie `SUB_PROG` można uruchomić przez następujące identyfikatory:

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

Taka sama nazwa programu głównego i podprogramu

Jeżeli istnieje program główny (.MPF) i podprogram (.SPF) o tej samej nazwie, należy podać odpowiednie rozszerzenie pliku w celu jednoznacznej identyfikacji podczas stosowania nazwy programu w programie NC. W przeciwnym razie zostanie użyty program znaleziony na początku ścieżki wyszukiwania z podaną nazwą programu.

2.24.1.3 Kaskadowanie podprogramów

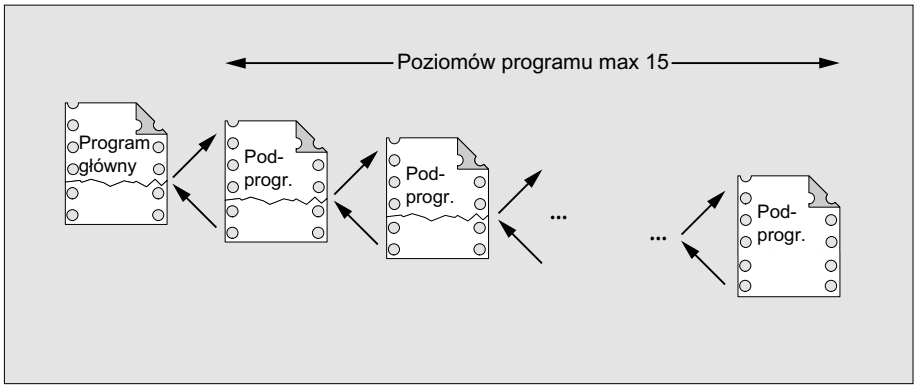
Program główny może wywoływać podprogramy, które wywołują kolejne podprogramy. Przebiegi programów są przez to w sobie zagnieżdżane. Każdy program jest przy tym wykonywany na własnym poziomie programu.

Głębokość kaskadowania

Język NC udostępnia aktualnie 16 poziomów programu. Program główny jest wykonywany zawsze na najwyższym poziomie programu 0. Podprogram jest zawsze wykonywany na kolejnym niższym poziomie programu. Poziom programu 1 jest przez to pierwszym poziomem podprogramu.

Podział poziomów programu:

- Poziom programu 0: Poziom programu głównego
- Poziom programu 1 - 15: poziom podprogramu 1 - 15



Procedury przerwania (ASUP)

Jeżeli w ramach procedury przerwania zostanie wywołany podprogram, jest on wykonywany nie na poziomie programu aktualnie aktywnym w kanale, lecz również na następnym niższym poziomie (n + 1). Aby było to możliwe również na najniższym poziomie programu, są w związku z procedurami przerwania do dyspozycji 2 dodatkowe poziomy (16 i 17).

Jeżeli będą potrzebne więcej, niż 2 poziomy programu, musi to zostać explicite uwzględnione w strukturyzacji programu obróbki wykonywanego w kanale. Tzn. wolno wówczas wykorzystać maksymalnie tylko tyle poziomów programu, by pozostały jeszcze wystarczające poziomy do wykonywania przerwań.

Jeżeli wykonanie przerwania wymaga np. 4 poziomów programu, wówczas program obróbki musi być tak strukturyzowany, by zajmował maksymalnie 13. poziom programu. Jeżeli wówczas nastąpi przerwanie, będą do dyspozycji 4 poziomy programu (14 do 17).

Cykle Siemens

Cykle Siemens wymagają 3 poziomów programu. Wywołanie cyklu Siemens musi dlatego nastąpić co najpóźniej w:

- Wykonywanie programu obróbki: poziom programu 12
- Procedura przerwania: poziom programu 14

2.24.1.4 Ścieżka szukania

Przy wywołaniu podprogramu bez podania ścieżki sterowanie przeszukuje istniejące pamięci programów wg predefiniowanej kolejności szukania (patrz "Ścieżka szukania przy wywołaniu podprogramu (Strona 230)").

2.24.1.5 Parametry formalne i aktualne

O parametrach formalnych i aktualnych mówimy w związku z definicją i wywoływaniem podprogramów z przekazaniem parametrów.

Parametry formalne

Przy definicji podprogramu muszą zostać zdefiniowane parametry będące do przekazania podprogramowi, tak zwane parametry formalne, z typem i nazwą parametru.

Parametry formalne definiują przez to interfejs podprogramu.

Przykład:

Kod programu	**Komentarz**
`PROC KONTUR (REAL X, REAL Y)`	`; Parametry formalne: X i Y obydwa typu REAL`
`N20 X1=X Y1=Y`	`; Ruch osi X1 do pozycji X i osi Y1 do pozycji Y`
`...`	
`N100 RET`	

Parametry aktualne

Przy wywołaniu podprogramu muszą zostać przekazane do podprogramu absolutne wartości lub zmienne, tak zwane parametry aktualne.

Przez to aktualne parametry przy wywołaniu wyposażają interfejs podprogramu w aktualne wartości.

Przykład:

Kod programu	**Komentarz**
N10 DEF REAL SZEROKOSC	; Definicja zmiennej
N20 SZEROKOSC=20.0	; Przyporządkowanie zmiennych
N30 KONTUR(5.5, SZEROKOSC)	; Wywołanie podprogramu z aktualnymi parametrami: 5.5 i SZEROKOŚĆ
...	
N100 M30	

2.24.1.6 Przekazanie parametrów

Definicja podprogramu z przekazaniem parametrów

Definicja podprogramu z przekazaniem parametrów następuje przy pomocy słowa kluczowego `PROC` i kompletnego wyszczególnienia wszystkich parametrów oczekiwanych od podprogramu.

Niekompletne przekazanie parametrów

Przy wywołaniu podprogramu nie zawsze wszystkie parametry zdefiniowane w interfejsie podprogramu muszą zostać bezpośrednio przekazane. Gdy parametr zostanie pominięty, jest dla tego parametru przekazywana wartość standardowa "0".

Do jednoznacznego oznaczenia kolejności parametrów muszą w każdym razie zawsze być podawane przecinki jako znaki rozdzielające parametry. Wyjątek stanowi ostatni parametr. Gdy zostanie on przy wywołaniu pominięty, można również pominąć ostatni przecinek.

Przykład:

Podprogram:

Kod programu	**Komentarz**
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; Parametry formalne: X, Y i Z
...	
N100 RET	

Program główny:

Kod programu	**Komentarz**
PROC MAIN_PROG	
...	

Kod programu	**Komentarz**
N30 SUB_PROG(1.0,2.0,3.0)	; Wywołanie podprogramu z kompletnym przekazaniem parametrów: X=1.0, Y=2.0, Z=3.0
…	
N100 M30	

Przykłady wywołania podprogramu w N30 z niekompletnym przekazaniem parametrów:

```
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
```

UWAGA

Przekazanie parametrów Call-by-Reference

Parametrów, które są przekazywane przez Call-by-Reference, nie wolno pomijać przy wywołaniu podprogramu.

UWAGA

Typ danych AXIS

Parametrów o typie danych AXIS nie wolno pomijać przy wywołaniu podprogramu.

Sprawdzenie przekazywanych parametrów

Przez zmienną systemową $P_SUBPAR [n] z n = 1, 2, ... można sprawdzić w podprogramie, czy parametr został bezpośrednio przekazany czy pominięty. Indeks n odnosi się do kolejności parametrów formalnych. Indeks n = 1 odnosi się do 1. parametru formalnego, indeks n = 2 do 2. parametru formalnego, itd.

Poniższy wycinek programu pokazuje na przykład dla 1. parametru formalnego, jak można realizować sprawdzenie:

Programowanie	**Komentarz**
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; Parametry formalne: X, Y i Z
N20 IF $P_SUBPAR[1]==TRUE	; Sprawdzenie 1. parametru formalnego X.
...	; Te akcje są przeprowadzane, gdy parametr formalny X został bezpośrednio przekazany.
N40 ELSE	
...	; Te akcje są przeprowadzane, gdy parametr formalny X nie został przekazany.
N60 ENDIF	
...	; Akcje ogólne

Programowanie	Komentarz
N100 RET	

2.24.2 Definicja podprogramu

2.24.2.1 Podprogram bez przekazania parametrów

Przy definicji podprogramów bez przekazania parametrów można pominąć wiersz definicji na początku programu.

Składnia

```
[PROC <nazwa programu>]
...
```

Znaczenie

PROC:	Polecenie definiujące na początku programu
<nazwa programu>:	Nazwa programu

Przykład

Przykład 1: Podprogram z instrukcją PROC

Kod programu	Komentarz
PROC SUB_PROG	; Wiersz definicji
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; Powrót z podprogramu

Przykład 2: Podprogram bez instrukcji PROC

Kod programu	Komentarz
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; Powrót z podprogramu

Patrz również

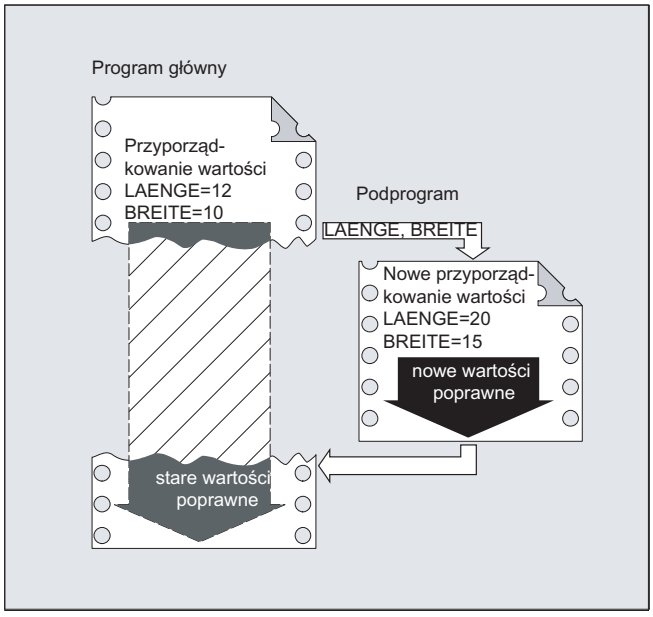
Wywołanie podprogramu bez przekazania parametrów (Strona 196)

2.24.2.2 Podprogram z przekazaniem parametrów Call-by-Value (PROC)

Definicja podprogramu z przekazaniem parametrów Call-by-Value następuje za pomocą słowa kluczowego `PROC`, po którym następuje nazwa programu i kompletne wyszczególnienie wszystkich parametrów z typem i nazwą. Polecenie definiujące musi znajdować się w pierwszym wierszu programu.

Call-by-value

Podczas przekazywania parametrów Call-by-Value program wywołujący przekazuje do podprogramu jedynie wartość zmiennej, dlatego podprogram nie ma bezpośredniego dostępu do zmiennych. Dzięki temu podczas zmiany wartości parametrów zmienia się tylko widoczna w podprogramie wartość. Wartość zdefiniowanych zmiennych w wywołanym programie nie ulega zmianie. Przekazanie parametrów Call-by-Value nie ma żadnego wpływu na program wywołujący.



Składnia

```
PROC <Nazwa programu> (<Typ parametru> <Nazwa
parametru>=<Wartość_inicj>, ...)
```

Uwaga

Może zostać przekazanych maksymalnie 127 parametrów.

Znaczenie

`PROC`:	Polecenie definiujące na początku programu
`<Nazwa programu>`:	Nazwa programu
`<Typ parametru>`:	Typ danych parametru (np. REAL, INT, BOOL)

<Nazwa parametru>:	Nazwa parametru
<Wartość_inicj>:	Wartość opcjonalna do inicjalizacji parametru (**opcjonalnie**) Jeżeli przy wywołaniu podprogramu parametr nie zostanie podany, zostanie mu przyporządkowana wartość inicjalizacyjna.

Przykłady

Przykład 1

Definicja podprogramu SUB_PROG z trzema parametrami typu REAL z wartościami domyślnymi:

Kod programu

```
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

Przykład 2

Różne warianty wywołania

Kod programu

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; warianty wywołania
  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
```

Patrz również

Wywołanie podprogramu z przekazaniem parametrów (EXTERN) (Strona 198)

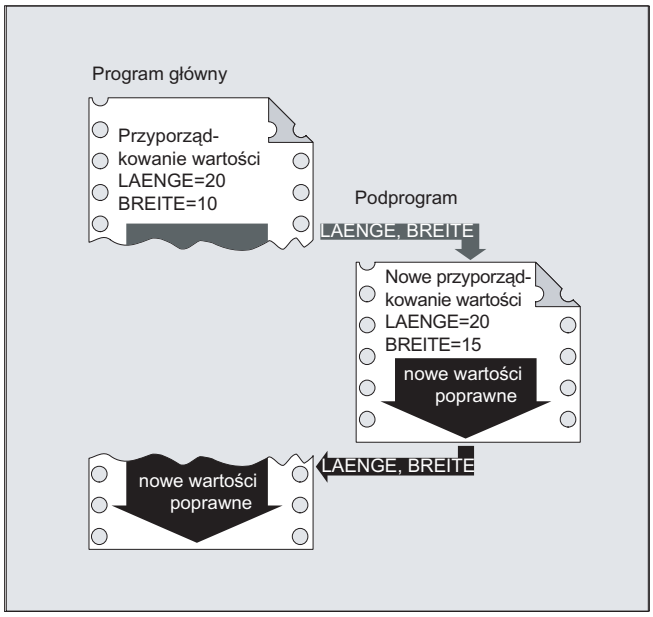
2.24.2.3 Podprogram z przekazaniem parametrów Call-by-Reference (PROC, VAR)

Definicja podprogramu z przekazaniem parametrów Call-by-Reference następuje za pomocą słowa kluczowego `PROC`, po którym następuje nazwa programu i kompletne wyszczególnienie wszystkich parametrów ze słowem kluczowym `VAR`, typem i nazwą. Polecenie definiujące musi znajdować się w pierwszym wierszu programu. Przy przekazaniu parametrów Call-by-Reference mogą być również przekazywane odniesienia do tablicy.

Call-By-Reference

Podczas przekazywania parametrów Call-by-Reference wywołany program nie przekazuje do podprogramu wartości zmiennej, tylko jej odniesienie (wskaźnik). W ten sposób podprogram ma bezpośredni dostępu do zmiennych. Dzięki temu podczas zmiany wartości parametrów zmienia się nie tylko widoczna wartość, ale także wartość zdefiniowanej zmiennej w

wywołanym programie. Przekazanie parametrów Call-by-Reference po zakończeniu podprogramu ma wpływ na program wywołujący.



Uwaga

Przekazanie parametrów Call-by-Reference jest wymagane tylko wtedy, gdy przekazywana zmienna została lokalnie zdefiniowana w programie wywołującym (LUD). Zmienne globalne dla kanału lub globalne dla NC nie muszą być przekazywane, gdyż dostęp do nich jest możliwy bezpośrednio z podprogramu.

Składnia

```
PROC <Nazwa programu> (VAR <Typ parametru> <Nazwa parametru>, ...)
PROC <Nazwa programu> (VAR <Typ tablicy<Nazwa
tablicy[<m>,<n>,<o>], ...)
```

Uwaga

Może zostać przekazanych maksymalnie 127 parametrów.

Znaczenie

PROC:	Polecenie definiujące na początku programu
VAR:	Słowo kluczowe dla przekazania parametrów przez odniesienie
<Nazwa programu>:	Nazwa programu
<Typ parametru>:	Typ danych parametru (np. REAL, INT, BOOL)
<Nazwa parametru>:	Nazwa parametru
<Typ tablicy:	Typ danych elementów tablicy (np. REAL, INT, BOOL)

<Nazwa tablicy>:	Nazwa tablicy	
[<m>,<n>,<o>]:	Wielkość tablicy Aktualnie są możliwe tablice maksymalnie 3-wymiarowe:	
	<m>:	Wielkość tablicy dla 1. wymiaru
	<n>:	Wielkość tablicy dla 2. wymiaru
	<o>:	Wielkość tablicy dla 3. wymiaru

Uwaga

- Nazwa programu podana po słowie kluczowym `PROC` musi być zgodna z nazwą programu nadaną w interfejsie obsługi.
- Za pomocą nieokreślonej długości pola jako parametru formalnego, podprogramy mogą wykonywać pola o zmiennej długości. Za pomocą nieokreślonej długości tablicy jako parametru formalnego, podprogramy mogą przetwarzać tablice o zmiennej długości. Przecinek musi być jednak pisany.
  Przykład: `PROC <Nazwa_programu> (VAR REAL TABLICA[,5]) (VAR REAL FELD[ ,5])`

Przykład

Definicja podprogramu z dwoma parametrami jako odniesienie do typu REAL:

Kod programu

```
; Parametr 1: Odniesienie do typu: REAL, nazwa: LANGE
; Parametr 2: Odniesienie do typu: REAL, nazwa: BREITE
PROC SUB_PROG(VAR REAL LAENGE, VAR REAL BREITE)
```

Patrz również

Wywołanie podprogramu z przekazaniem parametrów (EXTERN) (Strona 198)

2.24.2.4 Zapisywanie modalnych funkcji G (SAVE)

Atrybut `SAVE` powoduje, że aktywne przed wywołaniem podprogramu modalne polecenia G są zapisywane i ponownie uaktywniane po zakończeniu podprogramu.

UWAGA
Przerwanie trybu przechodzenia płynnego Jeżeli przy aktywnym trybie przechodzenia płynnego zostanie wywołany podprogram z atrybutem `SAVE`, tryb przechodzenia płynnego jest przerywany na końcu podprogramu (skok powrotny).

Składnia

```
PROC <Nazwa podprogramu> SAVE
```

Znaczenie

SAVE:	Zapisanie modalnych poleceń G przed wywołaniem podprogramu i przywrócenie po zakończeniu podprogramu

Przykład

W podprogramie KONTUR działa modalne polecenie G91 (wymiar przyrostowy). W programie głównym działa modalne polecenie G90 (wymiar absolutny). Przez definicję podprogramu z SAVE po zakończeniu podprogramu działa ponownie G90.

Definicja podprogramu:

Kod programu	Komentarz
PROC KONTUR (REAL WERT1) SAVE	; Definicja podprogramu z parametrem SAVE
N10 G91 ...	; Modalne polecenie G91: Wymiar przyrostowy
N100 M17	; Koniec podprogramu

Program główny:

Kod programu	Komentarz
N10 G0 X... Y... G90	; Modalne polecenie G90: Wymiar absolutny
N20 ...	
...	
N50 KONTUR (12.4)	; Wywołanie podprogramu
N60 X... Y...	; Modalne polecenie G90 reaktywowane przez SAVE

Warunki brzegowe

Frame

Zachowanie się frame odnośnie podprogramów z atrybutem SAVE jest zależne od typu frame i może zostać ustawione przez dane maszynowe.

Literatura

Podręcznik działania Funkcje podstawowe; Osie, Układy współrzędnych frame (K2), punkt: "Powrót z podprogramu z SAVE"

2.24.2.5 Blokowanie wykonywania pojedynczymi blokami (SBLOF, SBLON)

Blokowanie wykonywania pojedynczymi blokami dla całego programu

Programy oznaczone przez SBLOF są przy aktywnym wykonywaniu pojedynczymi blokami wykonywane jak jeden kompletny blok, tzn. dla całego programu wykonywanie pojedynczymi blokami jest zablokowane.

SBLOF znajduje się w wierszu PROC i działa, aż do końca albo anulowania podprogramu. Poleceniem powrotu następuje decyzja, czy na końcu podprogramu następuje zatrzymanie czy nie:

Powrót z M17:	Stop na końcu podprogramu
Powrót z RET:	Bez zatrzymania na końcu podprogramu

Blokowanie wykonywania pojedynczymi blokami w ramach programu

SBLOF musi znajdować się w oddzielnym bloku. Od tego bloku wykonywanie pojedynczymi blokami jest wyłączane do:

- następnego SBLON
  lub
- do końca aktywnego poziomu podprogramu

Składnia

Blokowanie wykonywania pojedynczymi blokami dla całego programu:
PROC ... SBLOF

Blokowanie wykonywania pojedynczymi blokami w ramach programu:

```
SBLOF
...
SBLON
```

Znaczenie

PROC:	Pierwsza instrukcja programu
SBLOF:	Polecenie wyłączenia wykonywania pojedynczymi blokami SBLOF może znajdować się w bloku PROC albo w oddzielnym bloku.
SBLON:	Polecenie włączenia wykonywania pojedynczymi blokami SBLON musi znajdować się w oddzielnym bloku.

Warunki brzegowe

- **Blokowanie wykonywania pojedynczymi blokami i wyświetlenie bloku**
  Aktualne wyświetlanie bloku może w cyklach/podprogramach być blokowane przy pomocy `DISPLOF`. Jeżeli `DISPLOF` zostanie zaprogramowane razem z `SBLOF`, wówczas przy zatrzymaniu wykonywania pojedynczymi blokami w ramach cyklu/podprogramu wyświetlanie następuje, jak przed wywołaniem cyklu/podprogramu.
- **Blokowanie wykonywania pojedynczymi blokami w ASUP systemowym i ASUP użytkownika**
  Gdy zatrzymanie w wykonywaniu pojedynczymi blokami jest blokowane w ASUP systemowym lub ASUP użytkownika przez ustawienia w danej maszynowej MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (Bit0 = 1 wzgl. Bit1 = 1), wówczas może ono zostać ponownie uaktywnione przez zaprogramowanie `SBLON` w ASUP.
  Gdy zatrzymywanie w wykonywaniu pojedynczymi blokami jest blokowane w ASUP użytkownika przez ustawienie w danej maszynowej MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP, **nie** można ponownie uaktywnić zatrzymywania przez zaprogramowanie `SBLON` w ASUP.
- **Cechy szczególne blokowania wykonywania pojedynczymi blokami przy różnych typach wykonywania pojedynczymi blokami**
  Przy aktywnym wykonywaniu pojedynczymi blokami SBL2 (stop po każdym bloku programu obróbki) **nie** następuje zatrzymanie w bloku `SBLON`, gdy w MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (uniemożliwienie zatrzymywania w wykonywaniu pojedynczymi blokami) bit 12 jest ustawiony na "1".
  Przy aktywnym wykonywaniu pojedynczymi blokami SBL3 (stop po każdym bloku programu obróbki również w cyklu) polecenie `SBLOF` jest blokowane.

Przykłady

Przykład 1: Blokowanie wykonywania pojedynczymi blokami w ramach programu

Kod programu	Komentarz
`N10 G1 X100 F1000`	
`N20 SBLOF`	`; Wyłączenie wykonywania pojedynczymi blokami`
`N30 Y20`	
`N40 M100`	
`N50 R10=90`	
`N60 SBLON`	`; Ponowne włączenie wykonywania pojedynczymi blokami`
`N70 M110`	
`N80 ...`	

Zakres między `N20` i `N60` jest przy wykonywaniu pojedynczymi blokami wykonywany jako jeden krok.

Przykład 2: Cykl powinien dla użytkownika działać jak jedno polecenie

Program główny:

Kod programu
`N10 G1 X10 G90 F200`

Kod programu
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30

Cykl CYCLE1:

Kod programu	**Komentarz**
N100 PROC CYCLE1 DISPLOF SBLOF	; Blokowanie wykonywania pojedynczymi blokami
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

Cykl CYCLE1 jest wykonywany przy aktywnym wykonywaniu pojedynczymi blokami, tzn. w celu wykonania CYCLE1 musi zostać naciśnięty przycisk Start.

Przykład 3: ASUP uruchomiony z PLC w celu uaktywnienia zmienionego przesunięcia punktu zerowego i korekcji narzędzia powinien być niewidoczny.

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

Przykład 4: Przy pomocy MD10702 Bit 12 = 1 nie następuje zatrzymanie

Sytuacja wyjściowa:

- Wykonywanie pojedynczymi blokami jest aktywne.
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

Program główny:

Kod programu	**Komentarz**
N10 G0 X0	; Zatrzymać w tym wierszu programu obróbki.
N20 X10	; Zatrzymać w tym wierszu programu obróbki.
N30 CYCLE	; Blok ruchu wygenerowany przez cykl.
N50 G90 X20	; Zatrzymać w tym wierszu programu obróbki.
M30	

Cykl CYCLE:

Kod programu	**Komentarz**
PROC CYCLE SBLOF	; Blokowanie zatrzymywania w wykonywaniu pojedynczymi blokami
N100 R0 = 1	
N110 SBLON	; Z powodu MD10702 bit12=1 w tym wierszu programu obróbki nie następuje zatrzymanie.
N120 X1	; W tym wierszu programu obróbki następuje zatrzymanie.
N140 SBLOF	
N150 R0 = 2	
RET	

Przykład 5: Blokowanie wykonywania pojedynczymi blokami przy kaskadowaniu programu

Sytuacja wyjściowa:

Wykonywanie pojedynczymi blokami jest aktywne.

Kaskadowanie programu:

Kod programu	**Komentarz**
N10 X0 F1000	; W tym bloku następuje zatrzymanie.
N20 UP1(0)	
PROC UP1(INT _NR) SBLOF	; Blokowanie zatrzymywania w wykonywaniu pojedynczymi blokami
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	

Kod programu	**Komentarz**
N210 SBLON	; Włączenie zatrzymywania w wykonywaniu pojedynczymi blokami
N220 X22	; W tym bloku następuje zatrzymanie.
N230 UP3(0)	
PROC UP3(INT _NR)	
N300 SBLOF	; Blokowanie zatrzymywania w wykonywaniu pojedynczymi blokami
N305 X30	
N310 SBLON	; Włączenie zatrzymywania w wykonywaniu pojedynczymi blokami
N320 X32	; W tym bloku następuje zatrzymanie.
N330 SBLOF	; Blokowanie zatrzymywania w wykonywaniu pojedynczymi blokami
N340 X34	
N350 M17	; SBLOF jest aktywne.
N240 X24	; W tym bloku następuje zatrzymanie. SBLON jest aktywne.
N250 M17	; W tym bloku następuje zatrzymanie. SBLON jest aktywne.
N120 X12	
N130 M17	; W tym bloku powrotu następuje zatrzymanie. SBLOF instrukcji PROC jest aktywne.
N30 X0	; W tym bloku następuje zatrzymanie.
N40 M30	; W tym bloku następuje zatrzymanie.

Dodatkowe informacje

Blokada wykonywania pojedynczymi blokami dla podprogramów asynchronicznych

Aby wykonać ASUP w jednym kroku przy wykonywaniu pojedynczymi blokami, musi w ASUP zostać zaprogramowana instrukcja `PROC` z `SBLOF`. Obowiązuje to również dla funkcji "edytowalny ASUP systemowy" (MD11610 $MN_ASUP_EDITABLE).

Przykład dla edytowalnego ASUP systemowego:

Kod programu	**Komentarz**
N10 PROC ASUP1 SBLOF DISPLOF	
N20 IF $AC_ASUP=='H200'	
N30 RET	; Bez REPOS przy zmianie trybu pracy
N40 ELSE	
N50 REPOSA	; REPOS we wszystkich pozostałych przypadkach.
N60 ENDIF	

Sterowanie programem w wykonywaniu pojedynczymi blokami

W trybie wykonywania pojedynczymi blokami użytkownik może wykonywać program obróbki pojedynczymi blokami. Istnieją następujące rodzaje ustawienia:

- SBL1: IPO pojedynczymi blokami z zatrzymaniem po każdym bloku zawierającym funkcje maszynowe.
- SBL2: Wykonywanie pojedynczymi blokami z zatrzymaniem po każdym bloku.
- SBL3: Zatrzymanie w cyklu (przez wybór SBL3 jest wyłączane polecenie `SBLOF`).

Blokowanie wykonywania pojedynczymi blokami przy kaskadowaniu programu

Jeżeli w podprogramie zaprogramowano `SBLOF` w instrukcji `PROC`, wówczas następuje zatrzymanie do powrotu z podprogramu przy pomocy `M17`. Zapobiega to temu, że w programie wywołującym będzie już wykonywany następny blok. Jeżeli w podprogramie z `SBLOF`, bez `SBLOF` w instrukcji `PROC`, zostanie uaktywnione blokowanie wykonywania pojedynczymi blokami, zatrzymanie następuje dopiero po następnym bloku zawierającym funkcję maszynową w programie wywołującym. Jeżeli jest to niepożądane, musi w podprogramie jeszcze przed powrotem (`M17`) ponownie zostać zaprogramowane `SBLON`. Przy powrocie z `RET` do programu nadrzędnego nie następuje zatrzymanie.

2.24.2.6 Blokowanie wyświetlania aktualnego bloku (DISPLOF, DISPLON, ACTBLOCNO)

Na wyświetlaniu bloku jest standardowo wyświetlany aktualny blok programu. W cyklach wzgl. podprogramach może przy pomocy polecenia `DISPLOF` być blokowane wyświetlanie aktualnego bloku. Zamiast aktualnego bloku jest wówczas wyświetlane wywołanie cyklu lub podprogramu. Poleceniem `DISPLON` można usunąć blokowanie wyświetlania bloku.

`DISPLOF` wzgl. `DISPLON` jest programowane w wierszu programu z instrukcją `PROC` i działa dla całego podprogramu i domyślnie dla wszystkich podprogramów wywoływanych z tego podprogramu, które nie zawierają polecenia `DISPLON` wzgl. `DISPLOF`. To zachowanie się działa również dla ASUP.

Składnia

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

Znaczenie

`DISPLOF`:	Polecenie blokowania aktualnego wyświetlania bloku.	
	Usytuowanie:	Na końcu wiersza programu z instrukcją `PROC`
	Działanie:	Aż do powrotu z podprogramu albo do końca programu.
	Wskazówka: Gdy z podprogramu z poleceniem `DISPLOF` są wywoływane dalsze podprogramy, wówczas również w tych podprogramach jest blokowane wyświetlanie aktualnego bloku, o ile nie jest w nich bezpośrednio zaprogramowane `DISPLON`.	

DISPLON:	Polecenie do wyłączenia blokowania wyświetlania aktualnego bloku.	
	Usytuowanie:	Na końcu wiersza programu z instrukcją PROC
	Działanie:	Aż do powrotu z podprogramu albo do końca programu.
	Wskazówka: Gdy z podprogramu z poleceniem DISPLON są wywoływane dalsze podprogramy, wówczas również w tych podprogramach jest wyświetlany aktualny blok, o ile nie jest w nich bezpośrednio zaprogramowane DISPLOF.	
ACTBLOCNO:	DISPLOF razem z atrybutem ACTBLOCNO powoduje, że w przypadku alarmu jest wyprowadzany numer aktualnego bloku, w którym wystąpił alarm. Obowiązuje to również wtedy, gdy na niższym poziomie programu jest zaprogramowane tylko DISPLOF. W przypadku DISPLOF bez ACTBLOCNO jest natomiast wyświetlany numer bloku wywołania cyklu wzgl. podprogramu z ostatniego poziomu programu nie oznaczonego przez DISPLOF.	

Przykłady

Przykład 1: Blokowanie aktualnego wyświetlania bloku w cyklu

```
Kod programu                              Komentarz
PROC CYCLE(AXIS TOMOV, REAL POSITION)     ; Blokowanie wyświetlania aktualnego bloku.
SAVE DISPLOF                              Zamiast tego ma być wyświetlane wywołanie
                                          cyklu, np.: CYCLE(X,100.0)
DEF REAL DIFF                             ; Treść cykli
G01 ...
...
RET                                       ; Powrót z podprogramu. Na wyświetleniu blo-
                                          ku jest wyświetlany blok następujący po wy-
                                          wołaniu cyklu.
```

Przykład 2: Wyświetlanie bloku przy wyprowadzeniu alarmu

Podprogram SUBPROG1 (z ACTBLOCNO):

```
Kod programu                    Komentarz
PROC SUBPROG1 DISPLOF ACTBLOC-
NO
N8000 R10 = R33 + R44
...
N9040 R10 = 66 X100             ; Wyzwolenie alarmu 12080
...
N10000 M17
```

Podprogram SUBPROG2 (bez ACTBLOCNO):

```
Kod programu                    Komentarz
PROC SUBPROG2 DISPLOF
```

Kod programu	Komentarz
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; Wyzwolenie alarmu 12080
...	
N7000 M17	

Program główny:

Kod programu	Komentarz
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; Wyprowadzenie alarmu = "12080 kanał K1 blok N9040 błąd składni w tekście R10="
N2060 ...	
N2350 SUBPROG2	; Wyprowadzenie alarmu = "12080 kanał K1 blok N2350 błąd składni w tekście R10="
...	
N3000 M30	

Przykład 3: Wyłączenie blokowania wyświetlania aktualnego bloku

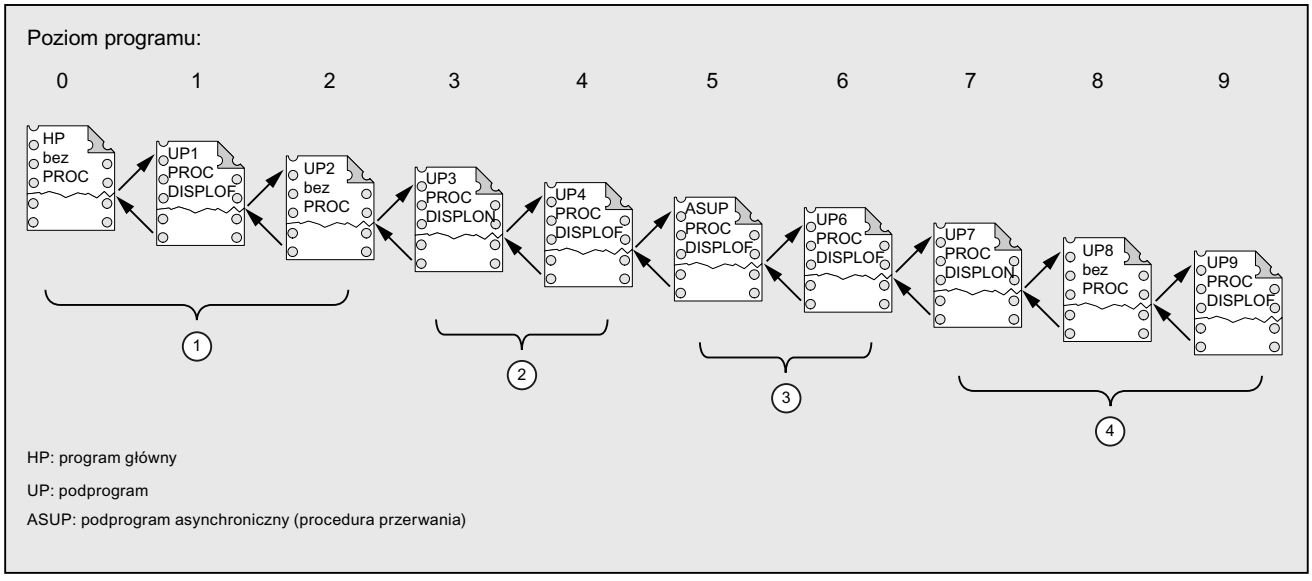
Podprogram SUB1 z blokowaniem:

Kod programu	Komentarz
PROC SUB1 DISPLOF	; Blokowanie wyświetlania aktualnego bloku w podprogramie SUB1. Zamiast tego ma być wyświetlany blok z wywołaniem SUB1.
...	
N300 SUB2	; Wywołanie podprogramu SUB2.
...	
N500 M17	

Podprogram SUB2 bez blokowania:

Kod programu	Komentarz
PROC SUB2 DISPLON	; Zniesienie blokowania wyświetlania aktualnego bloku w podprogramie SUB2.
...	
N200 M17	; Powrót do podprogramu SUB1. W SUB1 wyświetlanie aktualnego bloku jest ponownie blokowane.

Przykład 4: Zachowanie się wyświetlania przy różnych kombinacjach DISPLON/DISPLOF



① Na wyświetlaniu aktualnego bloku są wyświetlane wiersze programu obróbki z poziomu programu 0.
② Na wyświetlaniu aktualnego bloku są wyświetlane wiersze programu obróbki z poziomu programu 3.
③ Na wyświetlaniu aktualnego bloku są wyświetlane wiersze programu obróbki z poziomu programu 3.
④ Na wyświetlaniu aktualnego bloku są wyświetlane wiersze programu obróbki z poziomu programu 7/8.

2.24.2.7 Oznakowanie podprogramów z przygotowaniem (PREPRO)

Przy pomocy słowa kluczowego `PREPRO` mogą w rozruchu na końcu wiersza instrukcji `PROC` zostać oznakowane wszystkie pliki.

Uwaga

Ten rodzaj przygotowania programu jest zależny od odpowiednio ustawionej danej maszynowej. Należy przestrzegać danych producenta maszyny.

Literatura:
Podręcznik działania Funkcje specjalne; Przygotowanie (V2)

Składnia

```
PROC … PREPRO
```

Znaczenie

`PREPRO`:	Słowo kluczowe do oznakowania wszystkich plików przygotowanych w rozruchu, programów NC zapisanych w katalogach cykli

Wczytanie podprogramów z przygotowaniem i wywołanie podprogramu

Zarówno w rozruchu przygotowanych podprogramów z parametrami, jak też przy wywołaniu podprogramu katalogi cykli są traktowane w tej samej kolejności:

1. _N_CUS_DIR cykle użytkownika
2. _N_CMA_DIR cykle producenta
3. _N_CST_DIR cykle standardowe

W przypadku programów NC o tej samej nazwie, a różnym wykonaniu jest uaktywniana instrukcja `PROC` znaleziona jako pierwsza, a inna instrukcja `PROC` jest pomijana bez komunikatu alarmowego.

2.24.2.8 Powrót z podprogramu M17

Na końcu podprogramu znajduje się polecenie powrotu `M17` (wzgl. polecenie programu obróbki `M30`). Powoduje ono skok powrotny do programu wywołującego do bloku następującego po wywołaniu podprogramu.

Uwaga

`M17` i `M30` są w języku NC traktowane tak samo.

Składnia

```
PROC <nazwa programu>
...
M17/M30
```

Warunki brzegowe

Wpływ powrotu z podprogramu na tryb przechodzenia płynnego

Jeżeli `M17` (wzgl. `M30`) jest jedynym poleceniem w bloku programu obróbki, przerywa to aktywny w kanale tryb przechodzenia płynnego..

Aby uniknąć przerwania trybu przechodzenia płynnego, należy napisać `M17` (wzgl. `M30`) w ostatnim bloku ruchu postępowego. Dodatkowo następująca dana maszynowa musi być ustawiona na "0":

MD20800 $MC_SPF_END_TO_VDI = 0 (bez wyprowadzenia M30/M17 do interfejsu NC/PLC)

Przykład

1. Podprogram z `M17` w oddzielnym bloku

Kod programu	**Komentarz**
`N10 G64 F2000 G91 X10 Y10`	
`N20 X10 Z10`	

Kod programu	Komentarz
N30 M17	; Powrót z przerwaniem trybu przechodzenia płynnego.

2. Podprogram z M17 w ostatnim bloku ruchu postępowego

Kod programu	Komentarz
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; Powrót bez przerwania trybu przechodzenia płynnego.

2.24.2.9 Skok powrotny z podprogramu RET

Zamiast polecenia powrotu M17 można w podprogramie stosować również polecenie RET. RET musi być programowane w oddzielnym bloku programu obróbki. Tak jak M17 RET powoduje skok powrotny do programu wywołującego do bloku następującym po wywołaniu podprogramu.

Uwaga

Przez zaprogramowanie parametrów można zmienić zachowanie się RET pod względem skoku powrotnego (patrz "Parametryzowalny powrót z podprogramu (RET ...) (Strona 186)").

Zastosowanie

Instrukcji RET należy użyć wtedy, gdy praca z przechodzeniem płynnym G64 (G641 ..., G645) nie powinna być przerywana przez powrót.

Warunek

Polecenie RET można stosować tylko w podprogramach, które nie zostały zdefiniowane z atrybutem SAVE.

Składnia

```
PROC <nazwa programu>
...
RET
```

Przykład

Program główny:

Kod programu	Komentarz
PROC MAIN_PROGRAM	; Początek programu
...	

Kod programu	Komentarz
N50 SUB_PROG	; Wywołanie podprogramu: SUB_PROG
N60 ...	
...	
N100 M30	; Koniec programu

Podprogram:

Kod programu	Komentarz
PROC SUB_PROG	
...	
N100 RET	; Powrót następuje do bloku N60 w programie głównym.

2.24.2.10 Parametryzowalny powrót z podprogramu (RET ...)

Powszechnie przy pomocy polecenia `RET` wykonuje się skok powrotny z podprogramu do programu wywołującego. Wykonywanie jest kontynuowane od wiersza programu następującego po wywołaniu podprogramu. Jeżeli wykonywanie programu ma być kontynuowane w innym miejscu, dostępne są następujące opcje:

- Kontynuacja wykonywania programu po wywołaniu cykli skrawania w trybie dialektu ISO (według opisu konturu).
- Powrót do programu głównego z dowolnej płaszczyzny podprogramu (również po ASUP) w przypadku reakcji na błąd.
- Skok powrotny przez wiele poziomów programu dla zastosowań specjalnych w cyklach kompilacyjnych i w trybie dialektu ISO.

W tym celu należy zaprogramować polecenie `RET` z kolejnymi parametrami.

Kierunek szukania

Jeżeli podany jest parametr `<blok docelowy>` powrót następuje najpierw do bloku po bloku wywołania. Następnie cel jest wyszukiwany w kierunku **końca** programu, do którego następuje skok powrotny. Gdy wynik szukania nie powiedzie się, następuje szukanie w kierunku początku programu.

Składnia

```
RET("<blok docelowy>")
RET("<blok docelowy>",<blok po bloku docelowym>)
RET("<blok docelowy>",<blok po bloku docelowym>,<liczba poziomów
powrotu>)
RET("<blok docelowy>", ,<liczba poziomów powrotu>)
RET("<blok docelowy>",<blok po bloku docelowym>,<liczba poziomów
powrotu>,
<powrót do początku programu>)
RET( , ,<liczba płaszczyzn powrotu>,<powrót do początku programu>)
```

Znaczenie

RET:	Koniec podprogramu		
<blok docelowy>:	W parametrze blok, do którego ma być kontynuowane wykonywanie programu, jest określony jako miejsce docelowe skoku. Gdy parametr <liczba płaszczyzn powrotu> nie jest zaprogramowany, wówczas cel skoku znajduje się w tym programie, z którego wywołano bieżący podprogram. Możliwymi danymi są:		
	<numer bloku>	Numer bloku docelowego. Szukanie numeru bloku w programie, do którego nastąpił powrót, następuje najpierw w kierunku końca programu.	
	<znacznik skoku>	Znacznik skoku, do którego następuje powrót, musi być w programie. Szukanie znacznika skoku w programie, do którego nastąpił powrót, odbywa się najpierw w kierunku końca programu.	
	<łańcuch znaków>	Łańcuch znaków, do którego następuje powrót, musi być w programie (np. nazwa programu albo zmiennej). Szukanie łańcucha znaków w programie, do którego nastąpił powrót, odbywa się najpierw w kierunku końca programu. Dla programowania łańcucha znaków w bloku docelowym obowiązują następujące zasady: • **Spacja na końcu** (w odróżnieniu od znacznika skoku, który jest oznaczony przez ":" na końcu). • **Przed łańcuchem znaków** można ustawić tylko jeden numer bloku i/albo jeden znacznik skoku, natomiast **żadnego polecenia programowego**.	
<blok po bloku docelowym>:	W parametrze podaje się, czy wykonywanie programu ma być kontynuowane od bloku podanego w parametrze <blok docelowy> czy od bloku następnego.		
	Typ:	INT	
	Wartość:	0	Skok powrotny następuje do bloku podanego w parametrze <blok docelowy>.
		> 0	Skok powrotny następuje do następnego bloku określonego w parametrze <blok docelowy>.

<Liczba płaszczyzn powrotu>:	Parametr określa liczbę poziomów programu, które mają zostać przeskoczone, aby znaleźć blok docelowy i kontynuować wykonywanie programu.		
	Typ:	INT	
	Wartość:	1	Program jest kontynuowany na "aktualnym poziomie programu - 1" (jak RET bez parametrów).
		2	Program jest kontynuowany na "aktualnym poziomie programu - 2", tzn. jedna płaszczyzna jest przeskakiwana.
		3	Program jest kontynuowany na "aktualnym poziomie programu - 3", tzn. są przeskakiwane dwie płaszczyzny.
		...	
	Zakres wartości:	1 ... 15	
<Skok powrotny do początku programu>:	Parametr ten określa, czy program powinien być kontynuowany od początku po powrocie do programu głównego i przy aktywnym **trybie dialektu ISO**.		
	Typ:	BOOL	
	Wartość:	1	Gdy skok powrotny następuje do programu głównego, w którym jest aktywny **tryb dialektu ISO**, następuje rozgałęzienie na początek programu.

Uwaga

Przy powrocie z podprogramu z łańcuchem znaków jako informacją dla szukania bloku docelowego, program wywołujący zawsze szuka najpierw znacznika skoku.

Gdy cel skoku ma być jednoznacznie zdefiniowany przez łańcuch znaków, łańcuch znaków nie może być zgodny z nazwą znacznika skoku, gdyż w przeciwnym razie powrót z podprogramu będzie zawsze wykonywany na znacznik skoku, a nie na łańcuchu znaków (patrz przykład 2).

Warunki brzegowe

W przypadku powrotu przez wiele poziomów programu są uwzględniane instrukcje SAVE poszczególnych poziomów programu.

Jeśli podprogram modalny jest aktywny po powrocie z kilku poziomów programu i jeśli polecenie odwołania MCALL jest zaprogramowane dla modalnego podprogramu w jednym z pominiętych podprogramów, modalny podprogram pozostaje aktywny.

UWAGA

Błąd programowania

W przypadku skoku powrotnego przez wiele płaszczyzn programowych użytkownik jest odpowiedzialny za zapewnienie, że wymagane ustawienia modalne będą kontynuowane. Można to uzyskać np. przez zaprogramowanie odpowiedniego bloku głównego.

Przykłady

Przykład 1: Kontynuacja w programie głównym po wykonaniu ASUP

Programowanie	**Komentarz**
N10010 CALL "UP1"	; Poziom programu 0 (program główny)
N11000 PROC UP1	; Poziom programu 1
N11010 CALL "UP2"	
N12000 PROC UP2	; Poziom programu 2
...	
N19000 PROC ASUP	; Poziom programu 3 (obróbka ASUP)
...	
N19100 RET("N10900", ,$P_STACK)	; Powrót z podprogramu do programu głównego
	; $P_STACK: Aktualna płaszczyzna programu
N10900	; Blok docelowy w programie głównym
N10910 MCALL	; Wyłączenie modalnego wywołania podprogramu
N10920 G0 G60 G40 M5	; Inicjalizacja dalszych ustawień modalnych

Przykład 2: Łańcuch znaków (<String>) jako informacja dla szukania bloku docelowego

Program główny:

Kod programu	**Komentarz**
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; Wywołanie podprogramu "subProg1"
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; Wywołanie podprogramu "subProg2"
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; "subProg1" jest tutaj definiowany jako znacznik skoku.
N1560 ...	
N1600 subProg3	; Wywołanie podprogramu "subProg3"
N1610 ...	
N1900 M30	

Podprogram subProg1:

Kod programu	**Komentarz**
PROC subProg1	

Kod programu	Komentarz
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; Powrót do programu głównego do bloku N1400

Podprogram subProg2:

Kod programu	Komentarz
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; Powrót do programu głównego do bloku N1500

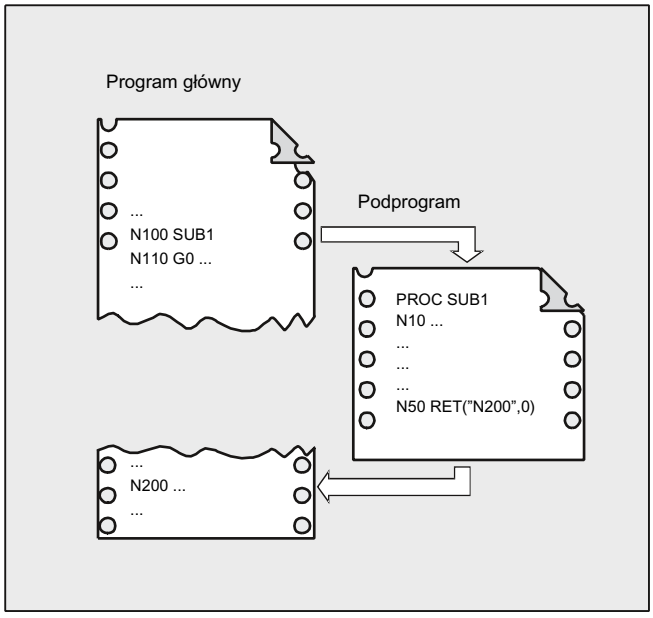
Podprogram subProg3:

Kod programu	Komentarz
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; Powrót do programu głównego do bloku N1550

Dodatkowe informacje

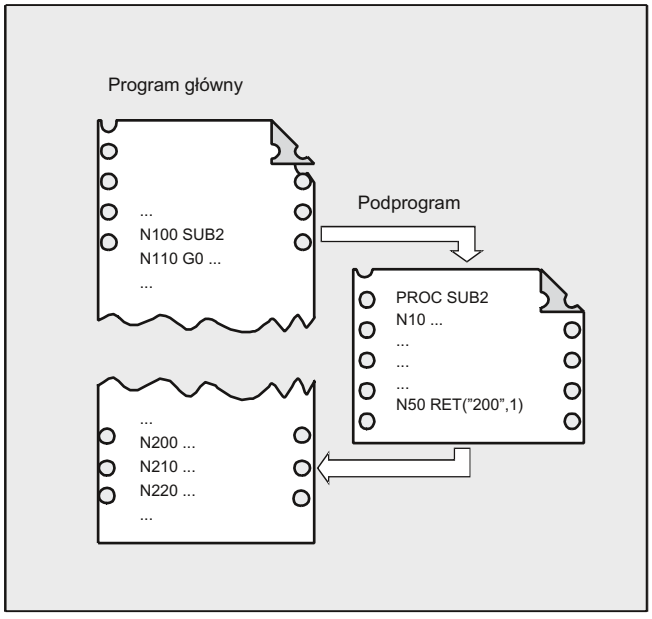
Poniższe grafiki pokazują różne działania parametrów powrotu

1. <blok docelowy> = "N200", <blok po bloku docelowym> = 0



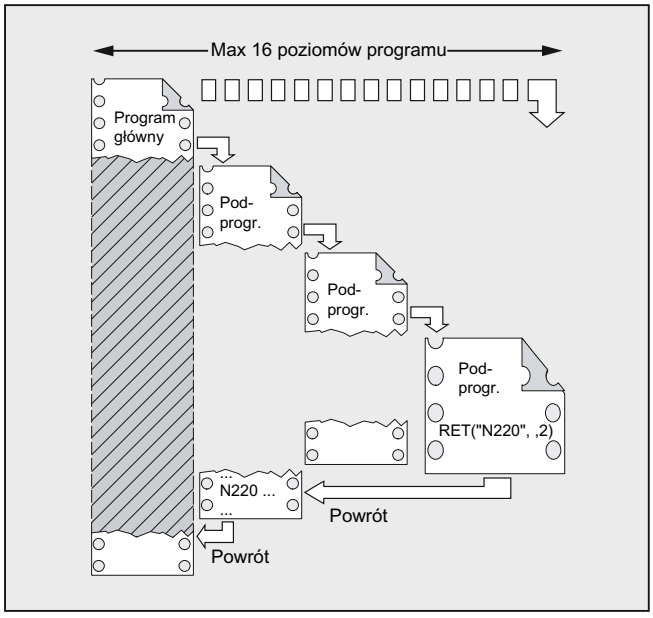
Po poleceniu `RET` wykonywanie jest kontynuowane od bloku `N200` w programie głównym.

2. <blok docelowy> = "N200", <blok po bloku docelowym> = 1



Po poleceniu `RET` wykonywanie programu jest kontynuowane od bloku `N210`, który w programie głównym następuje po bloku `N200`.

3. <blok docelowy> = "N220", <liczba płaszczyzn powrotu> = 2



Po poleceniu `RET` następuje skok powrotny o dwa poziomy programu i wykonywanie programu jest kontynuowane od bloku `N220`.

2.24.2.11 Parametryzowany powrót z podprogramu (RETB ...)

Zazwyczaj za pomocą polecenia `RETB` wykonuje się skok powrotny z podprogramu do programu wywołującego. Wykonywanie jest następnie kontynuowane od wiersza programu następującego po wywołaniu podprogramu. Jeżeli wykonywanie programu ma być kontynuowane w innym miejscu, dostępne są następujące opcje:

- Kontynuacja wykonywania programu po wywołaniu cykli skrawania w trybie dialektu ISO (według opisu konturu).
- Powrót do programu głównego z dowolnej płaszczyzny podprogramu (również po ASUP) w przypadku reakcji na błąd.
- Skok powrotny przez wiele poziomów programu dla zastosowań specjalnych w cyklach kompilacyjnych i w trybie dialektu ISO.

W tym celu należy zaprogramować polecenie `RETB` z kolejnymi parametrami.

Kierunek szukania

Jeżeli podany jest parametr `<blok docelowy>` powrót następuje najpierw do bloku po bloku wywołania. Następnie cel jest wyszukiwany w kierunku **końca** programu, do którego następuje skok powrotny. Gdy wynik szukania nie powiedzie się, następuje szukanie w kierunku początku program.

Składnia

```
RETB("<blok docelowy>")
RETB("blok docelowy>,<blok po bloku docelowym>
RETB("<blok docelowy>",<blok po bloku docelowym>,<liczba płaszczyzn
powrotu>)
RETB("<blok docelowy>", ,<liczba płaszczyzn powrotu>)
RETB("<blok docelowy>",<blok po bloku docelowym>,<liczba płaszczyzn
powrotu>,
<powrót do początku programu>)
RETB( , ,<liczba płaszczyzn powrotu>,<powrót do początku programu>)
```

Znaczenie

RETB:	Koniec podprogramu		
<blok docelowy>:	W parametrze blok, do którego ma być kontynuowane wykonywanie programu, jest określony jako miejsce docelowe skoku. Gdy parametr <liczba płaszczyzn powrotu> nie jest zaprogramowany, wówczas cel skoku znajduje się w tym programie, z którego wywołano bieżący podprogram. Możliwymi danymi są:		
	<Numer bloku>	Numer bloku docelowego. Szukanie numeru bloku w programie, odbywa się najpierw w kierunku początku programu, do którego nastąpił powrót.	
	<Znacznik skoku>	Znacznik skoku, do którego następuje powrót, musi być w programie. Szukanie znacznika skoku w programie odbywa się najpierw w kierunku początku programu, do którego nastąpił powrót.	
	<Łańcuch znaków>	Łańcuch znaków, do którego następuje powrót, musi być w programie (np. nazwa programu albo zmiennej). Szukanie łańcucha znaków w programie odbywa się najpierw w kierunku początku programu do którego nastąpił powrót. Poniższe zasady odnoszą się do programowania łańcucha znaków w bloku docelowym: • **Spacja na końcu** (w odróżnieniu od znacznika skoku, który jest oznaczony przez ":" na końcu). • **Przed łańcuchem znaków** można ustawić tylko jeden numer bloku i/albo jeden znacznik skoku, natomiast **żadnego polecenia programowego**.	
<blok po bloku docelowym>:	Parametr określa, czy wykonywanie programu ma być kontynuowane w bloku określonym w parametrze <blok docelowy> czy w następnym bloku.		
	Typ:	INT	
	Wartość:	0	Skok powrotny następuje do bloku określonego w parametrze <blok docelowy>.
		> 0	Skok powrotny następuje do następnego bloku określonego w parametrze <blok docelowy>.

<Liczba płaszczyzn powrotu>:	Parametr określa liczbę poziomów programu, które mają zostać przeskoczone, aby znaleźć blok docelowy i kontynuować wykonywanie programu.		
	Typ:	INT	
	Wartość:	1	Program jest kontynuowany na "aktualnym poziomie programu - 1" (jak RET bez parametrów).
		2	Program jest kontynuowany na "aktualnym poziomie programu - 2", tzn. jedna płaszczyzna jest pomijana.
		3	Program jest kontynuowany na "aktualnym poziomie programu - 3", tzn. dwie płaszczyzny są pomijane.
		...	
	Zakres wartości:	1 ... 15	
<Skok powrotny do początku programu>:	Parametr ten określa, czy program powinien być kontynuowany od początku po powrocie do programu głównego i przy aktywnym **trybie dialektu ISO**.		
	Typ:	BOOL	
	Wartość:	1	Gdy skok powrotny następuje do programu głównego, w którym jest aktywny **tryb dialektu ISO**, następuje rozgałęzienie na początek programu.

Uwaga

Przy powrocie z podprogramu z łańcuchem znaków jako informacją dla szukania bloku docelowego, program wywołujący zawsze szuka najpierw znacznika skoku.

Dlatego, gdy cel skoku ma być jednoznacznie zdefiniowany przez łańcuch znaków, łańcuch znaków nie może być zgodny z nazwą znacznika skoku, gdyż w przeciwnym razie powrót z podprogramu będzie zawsze wykonywany na znacznik skoku, a nie na łańcuchu znaków (patrz przykład 2).

Warunki brzegowe

W przypadku powrotu przez wiele poziomów programu wykorzystywane są instrukcje SAVE poszczególnych poziomów programu.

Jeśli podprogram modalny jest aktywny po powrocie z kilku poziomów programu i jeśli polecenie odwołania MCALL jest zaprogramowane dla modalnego podprogramu w jednym z pominiętych podprogramów, modalny podprogram pozostaje aktywny.

UWAGA
Błąd programowania
W przypadku skoku powrotnego przez wiele płaszczyzn programowych użytkownik jest odpowiedzialny za zapewnienie, że wymagane ustawienia modalne będą kontynuowane. Można to uzyskać np. przez zaprogramowanie odpowiedniego bloku głównego.

Przykład

Kod programu	**Komentarz**
BEISPIEL.MPF	
…	
N3000 START_CYC(parm1, 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	

Kod programu	**Komentarz**
PROC END_CYC(…)	; Wywołanie w programie głównym wiersz N3040
N10000...	
N15000 if status == 1	
N15010 RETB(“START_CYC”)	; Powrót do programu wywołującego PRZYKŁAD.MPF
	; Szukaj wg łańcucha znaków "START_CYC"
	; Kierunek szukania: do tyłu kierunek do początku programu
	; Kontynuacja wykonywania programu od wiersza N3000
N15020 endif	
N15030 if status == 0	
N15040 RET	; Powrót do programu wywołującego PRZYKŁAD.MPF
	; Kontynuacja wykonywania programu od wiersza N3050
N15050 endif	
N16000 RET(“START_CYC”)	; Powrót do programu wywołującego PRZYKŁAD.MPF
	; Szukaj wg łańcucha znaków "START_CYC"
	; Kierunek szukania: do przodu kierunek do końca programu
	; Kontynuacja wykonywania programu od wiersza N4500
N17060 RETB	; Powrót do programu wywołującego PRZYKŁAD.MPF
	; Kontynuacja wykonywania programu od wiersza N3050
	; RETB bez parametru jest identyczne z RET

2.24.3 Wywołanie podprogramu

2.24.3.1 Wywołanie podprogramu bez przekazania parametrów

Wywołanie podprogramu następuje albo z adresem L i numerem podprogramu albo przez podanie nazwy programu.

Również program główny może zostać wywołany jako podprogram. Wstawiony do programu głównego koniec programu `M2` albo `M30` jest w tym przypadku traktowany jak `M17` (koniec programu z powrotem do programu wywołującego).

Uwaga

Odpowiednio podprogram może również zostać uruchomiony jako program główny

Strategia szukania przez sterowanie:

Czy istnieje *_MPF?

Czy istnieje *_SPF?

Z tego wynika: Jeżeli nazwa podprogramu wywoływanego jest identyczna z nazwą programu głównego, wówczas jest ponownie wywoływany program wywołujący. Tego z reguły nie pożądanego zjawiska należy uniknąć przez jednoznaczny wybór nazw programów i podprogramów.

Uwaga

Podprogramy, które nie wymagają przekazania parametrów, mogą też być wywoływane z pliku inicjalizacyjnego.

Składnia

```
L<numer>/<nazwa programu>
```

Uwaga

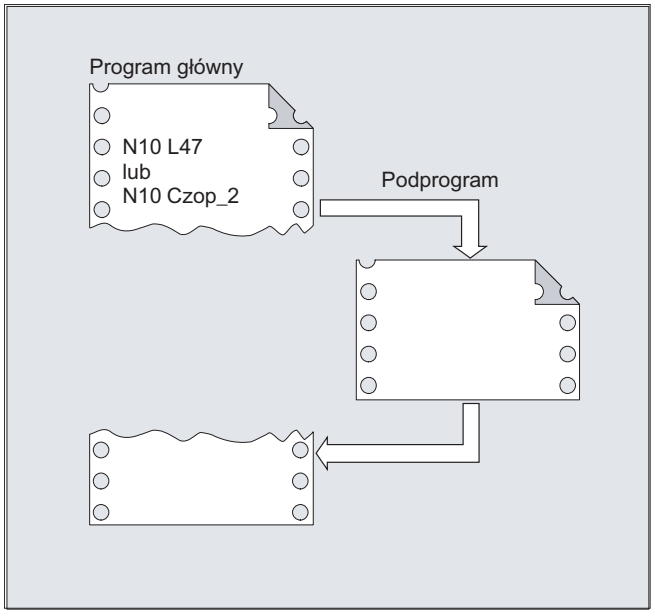
Wywołanie podprogramu musi zawsze być programowane w oddzielnym bloku NC.

Znaczenie

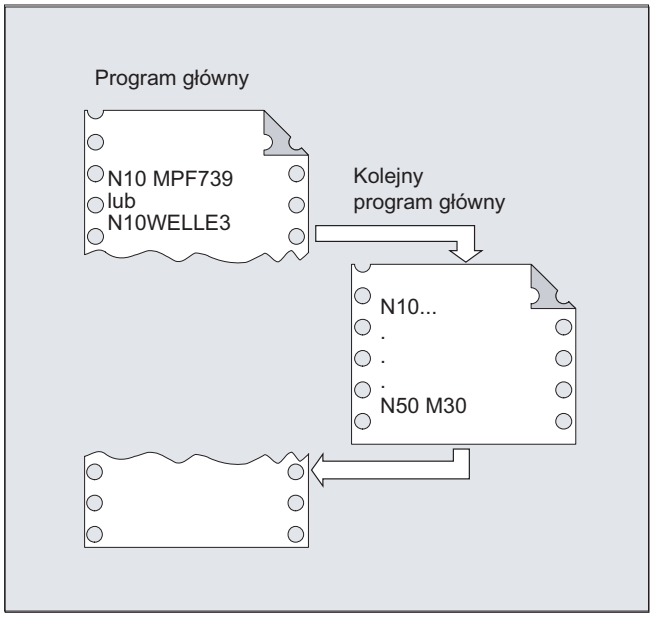
`L`:	Adres dla wywołania podprogramu	
`<numer>`:	Numer podprogramu	
	Typ:	INT
	Wartość:	Maksymalnie 7 miejsc dziesiętnych **Uwaga:** Zera na początku mają znaczenie przy nazewnictwie (⇒ L123, L0123 i L00123 są trzema różnymi podprogramami).
`<nazwa programu>`:	Nazwa podprogramu (lub programu głównego)	

Przykłady

Przykład 1: Wywołanie podprogramu bez przekazania parametrów



Przykład 2: Wywołanie programu głównego jako podprogramu



Patrz również

Podprogram bez przekazania parametrów (Strona 169)

2.24.3.2 Wywołanie podprogramu z przekazaniem parametrów (EXTERN)

Przy wywołaniu podprogramu z przekazaniem parametrów można bezpośrednio przekazywać zmienne albo wartości (nie w przypadku parametrów `VAR`).

Podprogramy z przekazaniem parametrów muszą przed wywołaniem zostać podane w programie głównym przez `EXTERN` (np. na początku programu). Podawane są przy tym nazwa podprogramu i typy zmiennych w kolejności ich przekazania.

UWAGA
Niebezpieczeństwo pomyłki Zarówno typy zmiennych jak i kolejność przekazania muszą być zgodne z definicjami, które zostały uzgodnione w nazwie podprogramu pod `PROC`. Nazwy parametrów mogą być różne w programie głównym i podprogramie.

Składnia

```
EXTERN <nazwa programu>(<typ_par1>,<typ_par2>,<typ_par3>)
...
<nazwa_programu>(<wartość_par1>,<wartość_par2>,<wartość_par3>)
```

Uwaga

Podprogram musi zawsze być programowany w oddzielnym bloku NC.

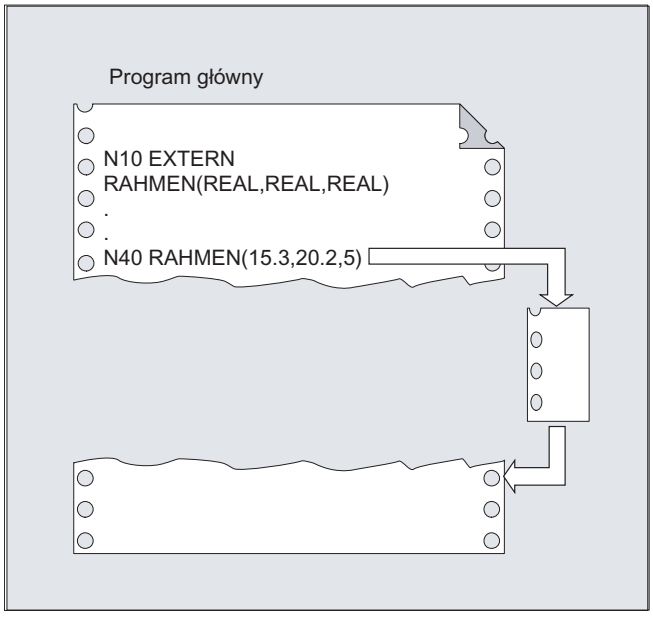
Znaczenie

`<nazwa programu>`:	Nazwa podprogramu
`EXTERN`:	Słowo kluczowe do podania podprogramu z przekazaniem parametrów. **Uwaga:** `EXTERN` musi zostać podane tylko wtedy, gdy podprogram znajduje się w katalogu obrabianego przedmiotu albo w globalnym katalogu podprogramów. Cykle nie muszą być deklarowane jako `EXTERN`.
`<typ_par1>,<typ_par2>,<typ_par3>`:	Typy zmiennych przekazywanych parametrów w kolejności przekazania
`<wartość_par1>,<wartość_par2>,<wartość_par3>`:	Wartości zmiennych dla przekazywanych parametrów

Przykłady

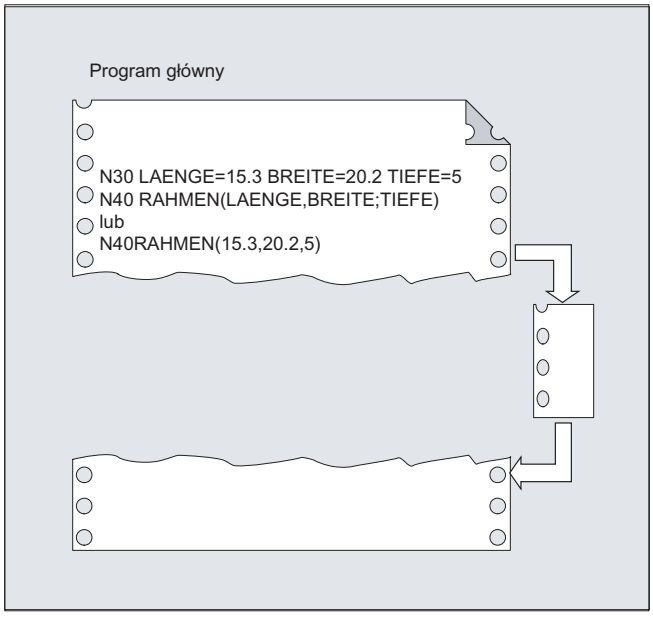
Przykład 1: Wywołanie podprogramu z uprzednim poinformowaniem

Kod programu	**Komentarz**
N10 EXTERN RAMKA(REAL,REAL,REAL)	; Podanie podprogramu.
...	
N40 RAMKA(15.3,20.2,5)	; Wywołanie podprogramu z przekazaniem parametrów



Przykład 2: Wywołanie podprogramu bez poinformowania

Kod programu	**Komentarz**
N10 DEF REAL DLUGOSC, SZEROKOSC, GŁRBOKOSC	
N20 …	
N30 DŁUGOSC=15.3 SZEROKOSC=20.2 GŁEBOKOSC=5	
N40 RAMKA(DŁUGOSC,SZEROKOSC,GŁEBOKOSC)	; lub: N40 RAMKA(15.3,20.2,5)



Patrz również

Podprogram z przekazaniem parametrów Call-by-Value (PROC) (Strona 170)

Podprogram z przekazaniem parametrów Call-by-Reference (PROC, VAR) (Strona 171)

2.24.3.3 Liczba powtórzeń programu (P)

Jeżeli podprogram ma być wykonywany kolejno wiele razy, można w bloku z wywołaniem podprogramu pod adresem `P` zaprogramować żądaną liczbę powtórzeń programu.

⚠ OSTROŻNIE
Wywołanie podprogramu z powtórzeniem programu i przekazaniem parametrów Parametry są przekazywane tylko przy wywołaniu programu wzgl. pierwszym przebiegu. Dla dalszych powtórzeń parametry pozostają bez zmian. W przypadku, gdy przy powtórzeniach programu chcemy zmieniać parametry, należy w podprogramie ustalić odpowiednie uzgodnienia.

Składnia

```
<nazwa programu> P<wartość>
```

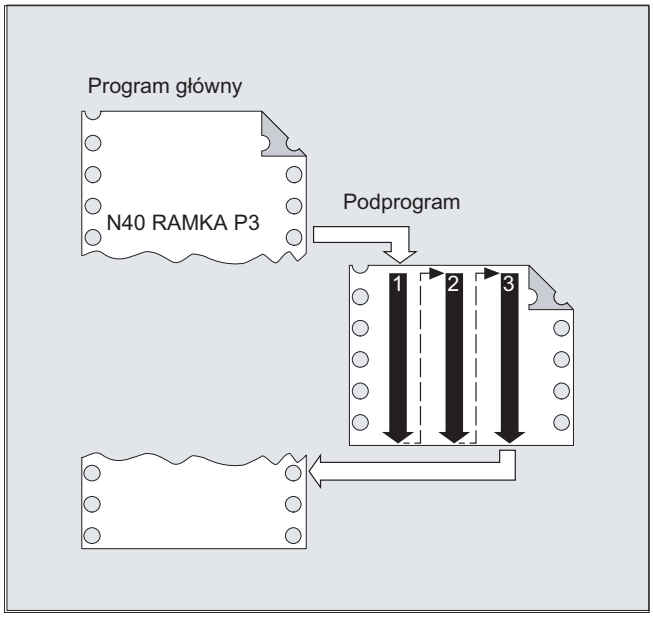
Znaczenie

`<nazwa programu>`:	Wywołanie podprogramu
`P`:	Adres dla programowania powtórzeń programu

<wartość>:	Liczba powtórzeń programu	
	Typ:	INT
	Zakres wartości:	1 … 9999 (bez znaku liczby)

Przykład

```
Kod programu          Komentarz
...
N40 RAMKA P3          ; Podprogram RAMKA ma być wykonywany kolejno 3 razy.
...
```



2.24.3.4 Modalne wywołanie podprogramu (MCALL)

Przez modalne wywołanie podprogramu `MCALL(<Nazwa programu>)` podany podprogram nie jest natychmiast wywoływany. Zamiast tego od tej chwili następuje automatyczne jego wywołanie w programie obróbki po każdym bloku ruchu postępowego z ruchami po torze. Również poza granice płaszczyzn programowych.

Uwaga

W przebiegu programu działa zawsze tylko ostatnie modalne wywołanie podprogramu `MCALL(<Nazwa programu>)`. Aktualne modalne wywołanie podprogramu zastępuje dotychczas aktywne.

Gdy parametry są przekazywane do podprogramu, przekazanie parametrów następuje tylko przy wywołaniu `MCALL(<Nazwa programu>(Par1, Par2, ...))`.

UWAGA
Modalne wywołania podprogramu bez ruchu po torze. Podprogram modalny jest wywoływany w następujących sytuacjach, również bez programowania ruchu po torze: • Programowanie adresów `S` albo `F`, gdy jest aktywne `G0` albo `G1` • Gdy `G0` lub `G1` zostało zaprogramowane samo w bloku albo z dalszymi poleceniami G.

Składnia

```
MCALL <Nazwa programu>
...
MCALL
```

Znaczenie

`MCALL <Nazwa programu>`:	Włączenie funkcji "Modalne wywołanie podprogramu"
`<Nazwa programu>`:	Nazwa podprogramu
`MCALL`:	Przy pomocy `MCALL` bez podania nazwy programu jest wyłączana funkcja "Modalne wywołanie podprogramu".

Warunki brzegowe

ASUP

Gdy wykonywanie programu obróbki zostanie przerwane przez ASUP (patrz punkt "Procedura przerwania (ASUP) (Strona 132)"), nie są w tym ASUP wykonywane żadne modalne wywołania podprogramu.

Gdy start ASUP nastąpi w stanie „Reset" kanału, zachowuje się on względem modalnych wywołań podprogramów jak normalny program obróbki.

Cykl wymiany narzędzia

Gdy w cyklu wymiany narzędzia następuje odwołanie funkcji "Modalne wywołanie podprogramu", wówczas należy pamiętać, że cykl wymiany narzędzia jest ew. wywoływany również po szukaniu bloku, pośrednio poprzez szukanie ASUP lub ręcznie przez pamięć pośrednią. W tej sytuacji nie wolno cofnąć wyboru funkcji "Modalne wywołanie podprogramu", gdyż w przeciwnym przypadku wynik szukania zostanie zafałszowany. Dlatego zaleca się cofnięcie wyboru funkcji "Modalne wywołanie podprogramu" w cyklu wymiany narzędzia programować następująco:

```
Kod programu                Komentarz
...
  IF $AC_ASUP == 0          ; Wywołanie następuje nie poprzez szukanie ASUP lub pamięć
                            pośrednią.
```

Kod programu	Komentarz
MCALL	; Wyłączenie funkcji "Modalne wywołanie podprogramu".
ENDIF	
...	

Przykłady

Przykład 1

Kod programu	Komentarz
N10 G0 X0 Y0	
N20 MCALL L70	; Włączenie modalnego wywołania podprogramu dla L70.
N30 X10 Y10	; Następuje ruch do X10 Y10 a następnie wywołanie L70.
N40 X20 Y20	; Następuje ruch do X20 Y20 a następnie wywołanie L70.
...	
N100 MCALL	; Wyłączenie funkcji "Modalne wywołanie podprogramu".
N110 X0 Y0	; Następuje ruch do X0 Y0, **nie** następuje wywołanie L70.

Przykład 2

Kod programu
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

W tym przykładzie następne bloki NC z zaprogramowanymi osiami biorącymi udział w tworzeniu konturu znajdują się w podprogramie L80. L70 jest wywoływany przez L80.

2.24.3.5 Pośrednie wywołanie podprogramu (CALL)

W zależności od danych warunków mogą w jednym miejscu być wywoływane różne podprogramy. W tym celu nazwa podprogramu jest zapisywana w zmiennej typu STRING. Wywołanie podprogramu następuje przy pomocy `CALL` i nazwy zmiennej.

Uwaga

Pośrednie wywołanie podprogramu jest możliwe tylko dla podprogramów bez przekazania parametrów. W celu bezpośredniego wywołania podprogramu należy zapisać nazwę w stałej STRING.

Składnia

```
CALL <nazwa programu>
```

Znaczenie

CALL:	Polecenie pośredniego wywołania podprogramu	
<nazwa programu>:	Nazwa podprogramu (zmienna albo stała)	
	Typ:	STRING

Przykład

Wywołanie bezpośrednie ze stałą STRING:

Kod programu	Komentarz
…	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_CZESC1_SPF"	; Bezpośrednie wywołanie podprogramu TEIL1 za pomocą CALL.
…	

Wywołanie pośrednie przez zmienną:

Kod programu	Komentarz
…	
DEF STRING[100] PROGNAME	; Definicja zmiennej.
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_CZESC1_SPF"	; Przyporządkowanie podprogramu TEIL1 do zmiennej PROGNAME.
CALL PROGNAME	; Pośrednie wywołanie podprogramu TEIL1 poprzez CALL i zmienną PROGNAME.
…	

2.24.3.6 Pośrednie wywołanie podprogramu z podaniem części programu do wykonania (CALL BLOCK ... TO ...)

Przy pomocy CALL i kombinacji słów kluczowych BLOCK... TO podprogram jest pośrednio wywoływany i wykonywana jego część oznaczona znacznikiem startowym i znacznikiem końcowym.

Składnia

```
CALL <nazwa programu> BLOCK <znacznik startowy> TO <znacznik końcowy>
CALL BLOCK <znacznik startowy> TO <znacznik końcowy>
```

Znaczenie

CALL:	Polecenie pośredniego wywołania podprogramu	
<nazwa programu>:	Nazwa podprogramu (zmienna lub stała), która zawiera będącą do wykonania część programu (**podanie opcjonalne**).	
	Typ:	STRING
	Wskazówka: Gdy <nazwa programu> nie jest zaprogramowana, jest w aktualnym programie szukana i wykonywana jego część oznaczona <znacznikiem startowym> i <znacznikiem końcowym>.	
BLOCK ... TO ... :	Kombinacja słów kluczowych dla pośredniego wykonania części programu	
<znacznik startowy>:	Zmienna, która odsyła do początku będącej do wykonania części programu.	
	Typ:	STRING
<znacznik końcowy>:	Zmienna, która odsyła do końca będącej do wykonania części programu.	
	Typ:	STRING

Przykład

Program główny:

Kod programu	**Komentarz**
...	
DEF STRING[20] ETYKIETA_STARTOWA, ETYKIETA_KONCOWA	; Definicja zmiennych dla znacznika startowego i znacznika końcowego.
ETYKIETA_STARTOWA="ETYKIETA_1"	
ETYKIETA_KONCOWA="ETYKIETA_2"	
...	
CALL "CONTUR_1" SEGMENT ETYKIETA_STARTOWA DO ETYKIETA_KONCOWA	; Pośrednie wywołanie podprogramu i oznaczenie będącej do wykonania części programu.
...	

Podprogram:

Kod programu	**Komentarz**
PROC CONTUR_1 ...	
ETYKIETA_1	; Znacznik startowy: Początek wykonywania części programu
N1000 G1 ...	
...	
ETYKIETA_2	; Znacznik końcowy; Koniec wykonywania części programu
...	

2.24.3.7 Pośrednie wywołanie programu programowanego w języku ISO (ISOCALL)

Przy pomocy pośredniego wywołania programu `ISOCALL` może zostać wywołany program zaprogramowany w języku ISO. Jest przy tym uaktywniany tryb ISO ustawiony w danych maszynowych. Na końcu programu działa ponownie pierwotny tryb obróbki. Jeżeli w danych maszynowych nie jest ustawiony tryb ISO, wywołanie podprogramu następuje w trybie Siemens.

Dalsze informacje dot. trybu ISO patrz:
Literatura:
Opis funkcji Dialekty ISO

Składnia

```
ISOCALL <Nazwa programu>
```

Znaczenie

`ISOCALL`:	Słowo kluczowe do pośredniego wywołania podprogramu, przy pomocy którego jest uaktywniany tryb ISO ustawiony w danych maszynowych
`<Nazwa programu>`:	Nazwa programu zaprogramowanego w języku ISO (zmienna lub stała typu STRING)

Przykład: Wywołanie konturu z cyklem zaprogramowanym w trybie ISO

```
Kod programu                                  Komentarz
0122_SPF                                      ; Opis konturu w trybie ISO
N1010 G1 X10 Z20
N1020 X30 R5
N1030 Z50 C10
N1040 X50
N1050 M99
N0010 DEF STRING[5] PROGNAME = "0122"         ; Program obróbki (cykl) Siemens
...
N2000 R11 = $AA_IW[X]
N2010 ISOCALL PROGNAME
N2020 R10 = R10+1                             ; Wykonanie programu 0122.spf w trybie
                                              ISO
...
N2400 M30
```

2.24.3.8 Wywołanie podprogramu z podaniem ścieżki i parametrami (PCALL)

Przy pomocy `PCALL` można wywoływać podprogramy z absolutnym podaniem ścieżki i przekazaniem parametrów.

Składnia

```
PCALL <ścieżka/nazwa_programu>(<parametr 1>,…,<parametr n>)
```

Znaczenie

PCALL:	Słowo kluczowe do wywołania podprogramu z absolutnym podaniem ścieżki.
<ścieżka/nazwa programu>:	Absolutne podanie ścieżki łącznie z nazwą podprogramu. Zasady podawania ścieżek patrz "Adresowanie plików pamięci programów (Strona 225)". Jeżeli nie podano ścieżki absolutnej, PCALL zachowuje się jak standardowe wywołanie podprogramu z identyfikatorem programu. Nazwa programu jest podawana bez przedrostka i bez rozszerzenia pliku. Jeżeli nazwa programu ma być programowana z przedrostkiem i rozszerzeniem, wówczas musi ona explicite zostać zadeklarowana z przedrostkiem i rozszerzeniem przy pomocy polecenia EXTERN.
<parametr 1>, ...:	Aktualny parametr odpowiednio do instrukcji PROC podprogramu.

Przykład

Kod programu
PCALL /_N_WKS_DIR/_N_WALEK_WPD/WALEK(parametr1,parametr2,…)

2.24.3.9 Rozszerzenie ścieżki szukania przy wywołaniach podprogramu (CALLPATH)

Przy pomocy polecenia CALLPATH można rozszerzyć ścieżkę szukania dla wywołań podprogramów. Przez to mogą być wywoływane również podprogramy z nie wybranego katalogu obrabianego przedmiotu, bez podania kompletnej, absolutnej nazwy ścieżki podprogramu.

Kolejną możliwością zastosowania jest tryb pracy EES "EES bez GDIR, gdy do zapisywania podprogramów globalnych jest używany katalog w zewnętrznej pamięci programów. W tym przypadku można przy pomocy CALLPATH rozszerzyć ścieżkę szukania o ten katalog podprogramów.

Rozszerzenie ścieżki szukania następuje przed wpisem dla cykli użytkownika (_N_CUS_DIR).

Następujące zdarzenia ponownie cofają wybór rozszerzenia ścieżki szukania:

- CALLPATH ze spacją
- CALLPATH bez parametrów
- Koniec programu obróbki
- Reset

Składnia

```
CALLPATH("<nazwa ścieżki>")
```

Znaczenie

CALLPATH:	Słowo kluczowe do programowanego rozszerzenia ścieżki szukania. Jest programowane we oddzielnym wierszu programu obróbki.
<nazwa ścieżki>:	Stała albo zmienna typu STRING. Zawiera absolutne podanie ścieżki katalogu, o który ścieżka szukania ma zostać rozszerzona. Zasady podawania ścieżek patrz "Adresowanie plików pamięci programów (Strona 225)".

Przykład

Ścieżka szukania ma zostać rozszerzona o określony katalog obrabianego przedmiotu:

Kod programu

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

Jest przez to ustawiana następująca ścieżka szukania (pozycja 5. jest nowa):

1. Aktualny katalog/*nazwa*
2. Aktualny katalog/*nazwa_SPF*
3. Aktualny katalog /*nazwa_MPF*
4. //NC:/_N_SPF_DIR/*nazwa_SPF*
5. **/_N_WKS_DIR/_N_MYWPD_WPD/nazwa_SPF**
6. /N_CUS_DIR/*nazwa_SPF*
7. /_N_CMA_DIR/*nazwa_SPF*
8. /_N_CST_DIR/*nazwa_SPF*

Warunki brzegowe

- CALLPATH sprawdza, czy zaprogramowana nazwa ścieżki rzeczywiście istnieje. W przypadku błędu wykonywanie programu obróbki jest przerywane z alarmem bloku korekcyjnego 14009.
- CALLPATH można również programować w plikach INI. Działa ono wówczas przez okres wykonywania pliku INI (plik WPD-INI albo program inicjalizacyjny dla aktywnych danych NC np. frame w 1. kanale _N_CH1_UFR_INI). Następnie ścieżka szukania jest ponownie cofana.

2.24.3.10 Wykonywanie podprogramu zewnętrznego (840D sl) (EXTCALL)

Przy pomocy polecenia EXTCALL można wykonywać program obróbki doładowując go z pamięci zewnętrznej.

Jako pamięć zewnętrzna dostępna jest:

- Dysk lokalny
- Dysk sieciowy
- Dyski USB

Uwaga

Jako interfejs do wykonywania programu zewnętrznego znajdującego się na dysku USB wolno stosować tylko interfejsy USB na pulpicie obsługi wzgl. TCU.

UWAGA
Uszkodzenie narzędzia/obrabianego przedmiotu przez USB-FlashDrive
Jest zalecane, by do wykonywania podprogramu zewnętrznego nie stosować USB-FlashDrive. Przerwanie komunikacji z USB-FlashDrive podczas wykonywania programu obróbki w wyniku problemów na stykach, wypadnięcia, przerwanie w wyniku uderzenia albo pomyłkowe wyciągnięcie, prowadzi do natychmiastowego zatrzymania obróbki. Może przy tym nastąpić uszkodzenie narzędzia i/albo obrabianego przedmiotu.

Ustawienie domyślne ścieżki programu zewnętrznego

Ścieżka do zewnętrznego katalogu programów może zostać domyślnie ustawiona przy pomocy danej ustawczej:

SD42700 $SC_EXT_PROG_PATH

Ze ścieżki i identyfikatora programu, podanego przy wywołaniu `EXTCALL` wynika ścieżka całkowita wywoływanego programu.

Uwaga

Parametry

Przy wywoływaniu programu zewnętrznego nie można mu przekazać żadnych parametrów.

Składnia

```
EXTCALL(„<Ścieżka/><Nazwa programu>")
```

Znaczenie

EXTCALL:	Polecenie do wywołania podprogramu zewnętrznego	
„<Ścieżka/><Nazwa programu>":	Stała/Zmienna typu STRING	
	<Ścieżka/>:	Absolutne lub względne podanie ścieżki (**opcjonalnie**)
	<Nazwa programu>:	Nazwa programu jest podawana bez przedrostka "_N_". Rozszerzenie pliku ("MPF", "SPF") można dołączyć do nazwy programu przy pomocy znaku "_" lub "." (**opcjonalnie**). Przykład: "WALEK" "WALEK_SPF" "WALEK.SPF"

Podanie ścieżki: Skróty

Następujące oznaczenia skrótowe można stosować przy podaniu ścieżki:

- Dysk lokalny: "**LOCAL_DRIVE:**"
- Karta CF: "**CF_CARD:**"
- Dysk USB (pulpit obsługi): "**USB:**"

Oznaczenia skrótowe "CF_CARD:" i "LOCAL_DRIVE:" można stosować alternatywnie.

Przykład

Wykonywanie ze dysku lokalnego

Program główny "MAIN.MPF" znajduje się w pamięci NC i jest wybrany do wykonania.

Podprogram "SP_1"

Podprogram zewnętrzny "SP_1.SPF" wzgl. "SP_1.MPF" znajduje się na dysku lokalnym w katalogu "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Ścieżkę do zewnętrznego katalogu programów należy ustawić przy pomocy:

SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

Uwaga

Podanie ścieżki do wywołania podprogramu zewnętrznego:

- Bez zastosowania ustawienia domyślnego: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- Z zastosowaniem ustawienia domyślnego: "SP_1"

Podprogram "SP_2"

Podprogram zewnętrzny "SP_2.SPF" wzgl. "SP_2.MPF" znajduje się w katalogu WKS.DIR/WST1.WPD na dysku USB. Ustawienie domyślne ścieżki do zewnętrznego katalogu

programów jest stosowane dla ścieżki podprogramu "SP_1" i w programie głównym również nie jest przepisywane. Dlatego przy wywołaniu podprogramu "SP_2" musi zostać podana kompletna ścieżka.

Program główny "MAIN"

```
Kod programu
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

Dodatkowe informacje

Wywołanie EXTCALL z absolutnym podaniem ścieżki

Gdy podprogram pod podaną ścieżką istnieje, wówczas jest wykonywany po wywołaniu `EXTCALL`. Gdy podprogram pod podaną ścieżką nie istnieje, wówczas po wywołaniu `EXTCALL` wykonywanie programu jest przerywane.

Wywołanie EXTCALL ze względnym podaniem ścieżki / bez podania ścieżki

W przypadku wywołania `EXTCALL` ze względnym podaniem ścieżki lub bez jej podania istniejące pamięci programów są przeszukiwane według następującego wzorca:

1. Gdy w SD42700 $SC_EXT_PROG_PATH jest domyślnie ustawiona ścieżka, następuje najpierw szukanie rozpoczynając od tej ścieżki według danej w wywołaniu `EXTCALL` (nazwa programu ew. ze względnym podaniem ścieżki). Ścieżka absolutna wynika wówczas przez powiązanie znaków:
   - z domyślnie ustawionego podania ścieżki w SD42700 $SC_EXT_PROG_PATH
   - Znak rozdzielający "/"
   - Podanie ścieżki i nazwa podprogramu w poleceniu `EXTCALL`
2. Gdy podprogram nie został znaleziony pod 1., są przeszukiwane katalogi pamięci użytkownika.

Szukanie kończy się, gdy podprogram został po raz pierwszy znaleziony. Jeżeli podprogram nie zostanie znaleziony, wykonywanie programu zostanie przez wywołanie `EXTCALL` przerwane.

Ustawiana pamięć doładowcza (bufor FIFO)

Do wykonywania podprogramu zewnętrznego jest potrzebna pamięć doładowania. Wielkość pamięci doładowania jest domyślnie ustawiona na 30 kByte i może zostać zmieniona tylko przez producenta maszyny (za pomocą MD18360 MM_EXT_PROG_BUFFER_SIZE).

Uwaga

Podprogramy z instrukcjami skoku

W przypadku podprogramów zewnętrznych, które zawierają instrukcje skoku (`GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF` itd.), cele skoku muszą być położone w ramach pamięci doładowania.

Uwaga

Programy ShopMill/ShopTurn

Programy ShopMill i ShopTurn z powodu dołączonych na końcu pliku opisów konturu muszą być zapisywane w pamięci doładowania w sposób kompletny.

Dla równolegle wykonywanych podprogramów zewnętrznych jest potrzebna własna pamięć doładowania.

Reset / koniec programu / POWER ON

Przez Reset i POWER ON zewnętrzne wywołania podprogramów są przerywane, a poszczególne pamięci doładowania kasowane.

Program wybrany do "Wykonywania ze źródła zewnętrznego" pozostaje po reset, końcu podprogramu albo POWER ON nadal wybrany dla "wykonywania ze źródła zewnętrznego". Zachowanie się nie różni się od wewnętrzne wybranych programów, o ile zewnętrzna pamięć programów jest nadal dostępna.

Literatura

Dalsze informacje dot. "Wykonywania ze źródła zewnętrznego" patrz:

Podręcznik działania Funkcje podstawowe; BAG, Kanał, Praca programowa, Zachowanie się przy zresetowaniu (K1)

2.24.3.11 Wykonywanie podprogramu zewnętrznego (828D) (EXTCALL)

Przy pomocy polecenia `EXTCALL` można wykonywać program obróbki doładowując go z pamięci zewnętrznej.

Jako pamięć zewnętrzna dostępna jest:

- Karta użytkownika CF
- Dysk sieciowy
- Dyski USB

Uwaga

Jako interfejs do wykonywania programu zewnętrznego znajdującego się na dysku USB wolno stosować tylko interfejs USB na pulpicie obsługi (PPU).

UWAGA
Uszkodzenie narzędzia/obrabianego przedmiotu przez USB-FlashDrive
Jest zalecane, by do wykonywania podprogramu zewnętrznego nie stosować USB-FlashDrive. Przerwanie komunikacji z USB-FlashDrive podczas wykonywania programu obróbki w wyniku problemów na stykach, wypadnięcia, przerwanie w wyniku uderzenia albo pomyłkowe wyciągnięcie, prowadzi do natychmiastowego zatrzymania obróbki. Może przy tym nastąpić uszkodzenie narzędzia i/albo obrabianego przedmiotu.

Ustawienie domyślne ścieżki programu zewnętrznego

Ścieżka do zewnętrznego katalogu programów może zostać domyślnie ustawiona przy pomocy danej ustawczej:

SD42700 $SC_EXT_PROG_PATH

Ze ścieżki i identyfikatora programu, podanego przy wywołaniu `EXTCALL` wynika ścieżka całkowita wywoływanego programu.

Uwaga

Parametry

Przy wywoływaniu programu zewnętrznego nie można mu przekazać żadnych parametrów.

Składnia

```
EXTCALL(„<Ścieżka/><Nazwa programu>")
```

Znaczenie

EXTCALL:	Polecenie do wywołania podprogramu zewnętrznego	
„<Ścieżka/><Nazwa programu>":	Stała/Zmienna typu STRING	
	<Ścieżka/>:	Absolutne lub względne podanie ścieżki (**opcjonalnie**)
	<Nazwa programu>:	Nazwa programu jest podawana bez przedrostka "_N_". Rozszerzenie pliku ("MPF", "SPF") można dołączyć do nazwy programu przy pomocy znaku "_" lub "." (**opcjonalnie**). Przykład: "WALEK" "WALEK_SPF" "WALEK.SPF"

Podanie ścieżki: Skróty

Następujące oznaczenia skrótowe można stosować przy podaniu ścieżki:

- Karta użytkownika CF: "**CF_CARD:**"
- Dysk USB (pulpit obsługi): "**USB:**"

Przykład

Program główny "MAIN.MPF" znajduje się w pamięci NC i jest wybrany do wykonania.

Podprogram "SP_1"

Podprogram zewnętrzny "SP_1.SPF" wzgl. "SP_1.MPF" znajduje się na karcie użytkownika CF w katalogu "/WKS.DIR/WST1.WPD".

Ścieżkę do zewnętrznego katalogu programów należy ustawić przy pomocy:

SD42700 $SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

Uwaga

Podanie ścieżki do wywołania podprogramu zewnętrznego:

- Bez zastosowania ustawienia domyślnego: "CF_CARD:WKS.DIR/WST1.WPD/SP_1"
- Z zastosowaniem ustawienia domyślnego: "SP_1"

Podprogram "SP_2"

Podprogram zewnętrzny "SP_2.SPF" wzgl. "SP_2.MPF" znajduje się w katalogu WKS.DIR/WST1.WPD na dysku USB. Ustawienie domyślne ścieżki do zewnętrznego katalogu programów jest stosowane dla ścieżki podprogramu "SP_1" i w programie głównym również nie jest przepisywane. Dlatego przy wywołaniu podprogramu "SP_2" musi zostać podana kompletna ścieżka.

Program główny "MAIN"

```
Kod programu
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

Dodatkowe informacje

Wywołanie EXTCALL z absolutnym podaniem ścieżki

Gdy podprogram pod podaną ścieżką istnieje, wówczas jest wykonywany po wywołaniu EXTCALL. Gdy podprogram pod podaną ścieżką nie istnieje, wówczas po wywołaniu EXTCALL wykonywanie programu jest przerywane.

Wywołanie EXTCALL ze względnym podaniem ścieżki / bez podania ścieżki

W przypadku wywołania EXTCALL ze względnym podaniem ścieżki lub bez jej podania istniejące pamięci programów są przeszukiwane według następującego wzorca:

1. Gdy w SD42700 $SC_EXT_PROG_PATH jest domyślnie ustawiona ścieżka, następuje najpierw szukanie rozpoczynając od tej ścieżki według danej w wywołaniu EXTCALL (nazwa programu ew. ze względnym podaniem ścieżki). Ścieżka absolutna wynika wówczas przez powiązanie znaków:
   - z domyślnie ustawionego podania ścieżki w SD42700 $SC_EXT_PROG_PATH
   - Znak rozdzielający "/"
   - Podanie ścieżki i nazwa podprogramu w poleceniu EXTCALL
2. Gdy podprogram nie został znaleziony pod 1., są przeszukiwane katalogi pamięci użytkownika.

Szukanie kończy się, gdy podprogram został po raz pierwszy znaleziony. Jeżeli podprogram nie zostanie znaleziony, wykonywanie programu zostanie przez wywołanie EXTCALL przerwane.

Ustawiana pamięć doładowcza (bufor FIFO)

Do wykonywania podprogramu zewnętrznego jest potrzebna pamięć doładowania. Wielkość pamięci doładowania jest ustawiona domyślnie (patrz MD18360 MM_EXT_PROG_BUFFER_SIZE).

Uwaga

Podprogramy z instrukcjami skoku

W przypadku podprogramów zewnętrznych, które zawierają instrukcje skoku (GOTOF, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDIF itd.), cele skoku muszą być położone w ramach pamięci doładowania.

Uwaga

Programy ShopMill/ShopTurn

Programy ShopMill i ShopTurn z powodu dołączonych na końcu pliku opisów konturu muszą być zapisywane w pamięci doładowania w sposób kompletny.

Dla równolegle wykonywanych podprogramów zewnętrznych jest potrzebna własna pamięć doładowania.

Reset / koniec programu / POWER ON

Przez Reset i POWER ON zewnętrzne wywołania podprogramów są przerywane, a poszczególne pamięci doładowania kasowane.

Program wybrany do "Wykonywania ze źródła zewnętrznego" pozostaje po reset, końcu podprogramu albo POWER ON nadal wybrany dla "wykonywania ze źródła zewnętrznego". Zachowanie się nie różni się od wewnętrzne wybranych programów, o ile zewnętrzna pamięć programów jest nadal dostępna.

Literatura

Dalsze informacje dot. "Wykonywania ze źródła zewnętrznego" patrz:

Podręcznik działania Funkcje podstawowe; BAG, Kanał, Praca programowa, Zachowanie się przy zresetowaniu (K1)

2.25 Technika makr (DEFINE ... AS)

UWAGA
Technika makr zwiększa kompleksowość programowania Przy zastosowaniu techniki makr język programowania sterowania może zostać bardzo zmieniony. Technika makr powinna być stosowana z dużą ostrożnością.

Jako makro określa się połączenie poszczególnych instrukcji w nową instrukcję łączną z własną nazwą. Przy wywołaniu makra w przebiegu programu instrukcje zaprogramowane pod nazwą makra są kolejno wykonywane.

Odpowiednio do zakresu obowiązywania, tzn. zakresu w którym działa definicja makra, wyróżniamy następujące kategorie makr:

- Makra lokalne
  Makra lokalne to makra, które zdefiniowane są na początku programu NC, który w chwili wykonywania nie jest programem głównym. Są one tworzone przy wywołaniu programu NC i kasowane z końcem tego programu lub następnym załączeniem sterowania. Do makr lokalnych można mieć dostęp tylko w ramach programu NC, w którym są one zdefiniowane.
- Makra globalne dla programu
  Makra globalne dla programu są to makra, które definiowano na początku programu NC stosowanego jako program główny. Są one tworzone przy wywołaniu programu NC i kasowane z końcem tego programu lub następnym załączeniem sterowania. Do makr globalnych dla programu można mieć dostęp w programie głównym i we wszystkich podprogramach.

  Uwaga

  Dostępność makr globalnych dla programu

  Zdefiniowane w programie głównym makra globalne dla programu są tylko wtedy dostępne w podprogramach, gdy ustawiona jest następująca dana maszynowa:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Za pomocą MD11120 = 0 zdefiniowane w programie głównym makra globalne dla programu są tylko dostępne w programie głównym.

- Makra globalne
  Makra globalne są globalnymi makrami dla NC lub kanału, które są zdefiniowane w pliku definicji (plik makr) i pozostają zachowane po resecie na końcu programu lub po następnym załączeniu sterowania. Makra globalne można następnie wywołać i wykonać w dowolnym programie głównym albo podprogramie.

  Uwaga

  Aby móc stosować makra **zewnętrznego** pliku makr w programie NC, plik makr musi zostać załadowany do NC.

Makra muszą być zdefiniowane przed ich zastosowaniem. Należy przy tym przestrzegać następujących zasad:

- W makropoleceniu mogą być definiowane dowolne identyfikatory, funkcje G, M, H i nazwy podprogramów L.
- Definicja makra może nastąpić na początku programu lub we własnym pliku definicji (plik makr).
- Lokalne i globalne makra dla programu definiowane są na początku programu.
- Makra globalne należy definiować w pliku makr, np. _N_DEF_DIR/_N_UMAC_DEF.
- Makra poleceń G mogą być definiowane tylko jako makra globalne.
- Funkcje H i L mogą być programowane 2-cyfrowo.
- Polecenia M i G mogą być programowane 3-cyfrowo.

Uwaga

Słów kluczowych i zarezerwowanych nazw nie wolno nadpisywać za pomocą makr.

Składnia

Definicja makra:
```
DEFINE <Macro_Name> AS <Operation_1> <Operation_2> ...
```

Wywołanie w programie NC:
```
<Macro_Name>
```

Znaczenie

`DEFINE ... AS`:	Kombinacja słów kluczowych do definicji makra
`<Macro_Name>`:	Nazwa makra Jako nazwy makr są dopuszczalne tylko identyfikatory. Przy pomocy nazwy makra makro jest wywoływane z programu NC.
`<Operation_1>`:	Pierwsze polecenie zaprogramowane w makro
`<Operation_2>`:	Drugie polecenie zaprogramowane w makro

Warunki brzegowe

Kaskadowanie

Kaskadowanie makr jest niemożliwe.

Przykłady

Przykład 1: Definicja makra na początku programu

Kod programu	Komentarz
`DEFINE LINIA AS G1 G94 F300`	`; Definicja makra`
`...`	
`N70 LINIA X10 Y20`	`; Wywołanie makra`
`...`	

Przykład 2: Definicje makr w pliku makr

Kod programu	Komentarz
`DEFINE M6 AS L6`	`; Przy wymianie narzędzia wywoływany jest podprogram, który przejmuje niezbędną transmisję danych. W podprogramie jest wyprowadzana właściwa funkcja M wymiany narzędzia (np. M106).`
`DEFINE G81 AS DRILL(81)`	`; Odwzorowanie funkcji G DIN.`
`DEFINE G33 AS M333 G333`	`; Przy nacinaniu gwintu następuje żądanie synchronizacji z PLC. Pierwotne polecenie G33 została przez MD przemianowane na G333, programowanie pozostaje dla użytkownika takie samo.`

Przykład 3: Zewnętrzny plik makr

Po wczytaniu zewnętrznego pliku makr do sterowania, plik makr musi zostać załadowany do NC. Dopiero wówczas makra mogą być zastosowane w programie NC.

Kod programu	Komentarz
`%_N_UMAC_DEF`	
`;$PATH=/_N_DEF_DIR`	`; Makra specyficzne dla klienta`
`DEFINE PI AS 3.14`	
`DEFINE TC1 AS M3 S1000`	
`DEFINE M13 AS M3 M7`	`; Wrzeciono w prawo, załączenie chłodziwa`
`DEFINE M14 AS M4 M7`	`; Wrzeciono w lewo, załączenie chłodziwa`
`DEFINE M15 AS M5 M9`	`; Zatrzymanie wrzeciona, wyłączenie chłodziwa`
`DEFINE M6 AS L6`	`; Wywołanie programu wymiany narzędzia`
`DEFINE G80 AS MCALL`	`; Odwołanie cyklu wiercenia`
`M30`	

Zarządzanie plikami i programami 3

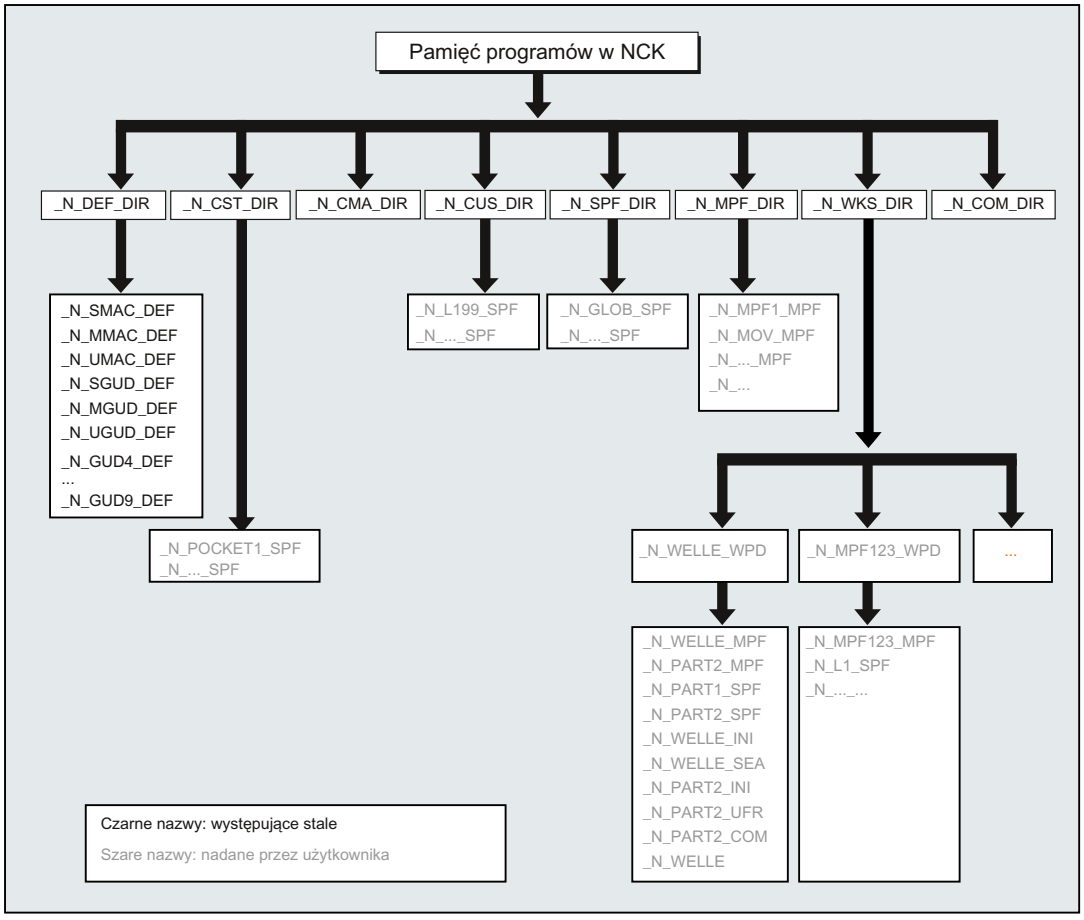
3.1 Pamięć programów

3.1.1 Pamięć programów w NC

W pamięci programów są trwale zapisywane pliki i programy (np. programy główne i podprogramy, definicje makr) (→ pasywny system plików).

Literatura:
Podręcznik działania Funkcje rozszerzone; Konfiguracja pamięci (S7)

Oprócz tego jest pewna liczba typów plików, które mogą być tutaj zapisywane tymczasowo i w razie potrzeby (np. przy obróbce określonego przedmiotu) powinny być przenoszone do pamięci roboczej (np. do celów inicjalizacyjnych).



Katalogi standardowe

Standardowo są zawarte następujące katalogi:

Wykaz	Treść
_N_DEF_DIR	Moduły danych i moduły makr
_N_CST_DIR	Cykle standardowe
_N_CMA_DIR	Cykle producenta
_N_CUS_DIR	Cykle użytkownika
_N_WKS_DIR	Obrabiane przedmioty
_N_SPF_DIR	Podprogramy globalne
_N_MPF_DIR	Programy główne
_N_COM_DIR	Komentarze

Typy plików

W pamięci programów mogą być umieszczane następujące typy plików:

Typ pliku	Opis
<Nazwa>_MPF	Program główny
<Nazwa>_SPF	Podprogram
<Nazwa>_TEA	Dane maszynowe
<Nazwa>_SEA	Dane ustawcze
<Nazwa>_TOA	Korekcje narzędzi
<Nazwa>_UFR	Przesunięcia punktu zerowego/frame
<Nazwa>_INI	Plik inicjalizacyjny
<Nazwa>_GUD	Globalne dane użytkownika
<Nazwa>_RPA	Parametry R
<Nazwa>_COM	Komentarz
<Nazwa>_DEF	Definicje globalnych danych użytkownika i makr

Katalog główny obrabianych przedmiotów (_N_WKS_DIR)

Katalog główny obrabianych przedmiotów jest standardowo utworzony w pamięci programów pod nazwą _N_WKS_DIR. Katalog główny obrabianych przedmiotów zawiera dla wszystkich obrabianych przedmiotów, które zostały zaprogramowane, odpowiednie katalogi obrabianego przedmiotu.

Katalogi obrabianego przedmiotu (..._WPD)

Katalog obrabianego przedmiotu zawiera wszystkie pliki, które są konieczne do obróbki tego przedmiotu. Mogą to być programy główne, podprogramy, dowolne programy inicjalizacyjne i pliki komentarzy.

Programy inicjalizacyjne są jednorazowo wykonywane z pierwszym startem programu po jego wybraniu (odpowiednio do danej maszynowej MD11280 $MN_WPD_INI_MODE).

Przykład:

Katalog obrabianego przedmiotu _N_WALEK_WPD, który został utworzony dla obrabianego przedmiotu WALEK, zawiera następujące pliki:

Plik	Opis
_N_WALEK_MPF	Program główny
_N_PART2_MPF	Program główny
_N_PART1_SPF	Podprogram
_N_PART2_SPF	Podprogram
_N_WALEK_INI	Ogólny program inicjalizujący dane obrabianego przedmiotu
_N_WALEK_SEA	Program inicjalizujący dane ustawcze
_N_PART2_INI	Ogólny program inicjalizujący dane dla programu Part 2
_N_PART2_UFR	Program inicjalizujący dane frame dla programu Part 2
_N_WALEK_COM	Plik komentarzy

Dodatkowo w katalogu obrabianego przedmiotu można zapisać również dane, które nie są bezpośrednio potrzebne do wykonywania przez NC. Mogą to oprócz plików ASCII być również pliki binarne jak np. obrazy w formacie JPG albo opisy w formacie PDF. Aby mogły one być interpretowane przez NC jako pliki binarne, rozszerzenia nazw plików muszą być znane w NC (ustawienie przy uruchamianiu poprzez MD17000 $MN_ EXTENSIONS_OF_BIN_FILES; w ustawieniu podstawowym znajdują się następujące rozszerzenia: JPG, GIF, PNG, BMP, PDF, ICO, HTM).

Wybranie obrabianego przedmiotu do obróbki

Katalog obrabianego przedmiotu może zostać wybrany w celu wykonania w kanale. Jeżeli w tym katalogu znajduje się program główny **o tej samej nazwie** albo tylko jeden program główny (_MPF), wówczas jest on automatycznie wybierany do wykonania.

Literatura:
Podręcznik obsługi

3.1.2 Zewnętrzna pamięć programów

Oprócz pasywnego systemu plików w NC mogą na maszynie być dostępne również zewnętrzne pamięci programów (np. na dysku lokalnym albo dysku sieciowym).

Poprzez funkcje „Wykonywanie ze źródła zewnętrznego" albo "EES (Execution from External Storage)" programy obróbki można wykonywać **bezpośrednio** z zewnętrznych pamięci programu.

Literatura:
Podręcznik działania Funkcje podstawowe: K1: BAG, Kanał, Praca programowa, Zachowanie się przy zresetowaniu

Globalna pamięć programów obróbki (GDIR)

Przy uzgadnianiu dysków jeden z nich można określić jako globalną pamięć programów obróbki (GDIR).

Literatura:
Podręcznik obsługi; rozdział: "Zarządzanie programami" > "Ustawianie dysków"

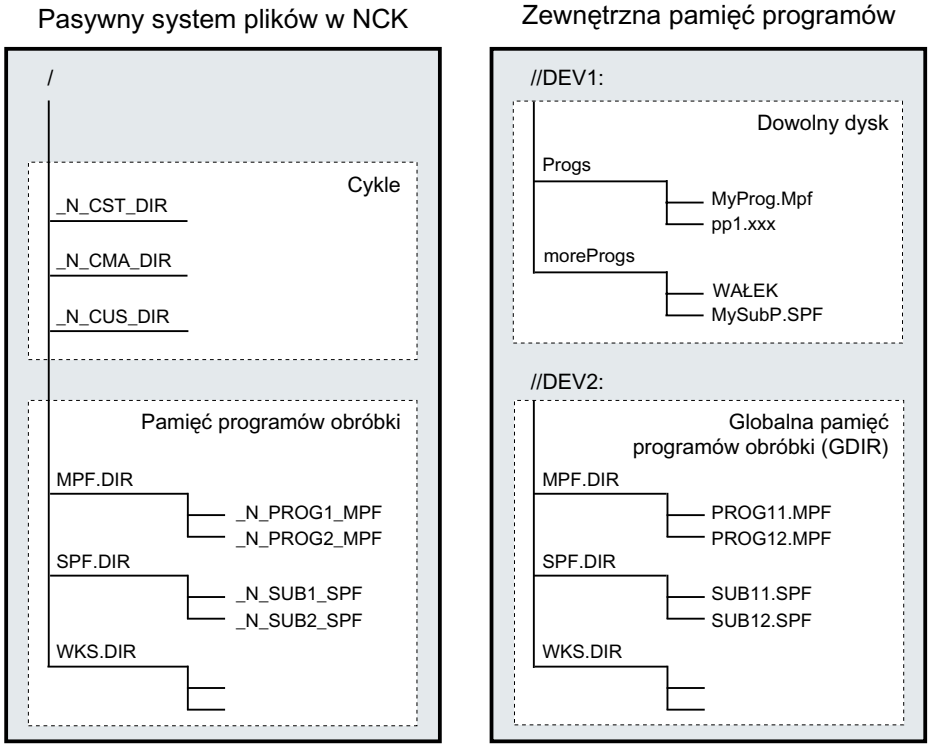
Wówczas system automatycznie tworzy na dysku katalogi MPF.DIR, SPF.DIR i WKS.DIR. Te trzy katalogi tworzą GDIR.

GDIR odgrywa rolę wyłącznie dla funkcjonowania EES. Zależnie od konfiguracji dysku, GDIR zastępuje lub rozszerza pamięć programów obróbki NC. Ustawienie GDIR nie jest jednak dla pracy EES koniecznie wymagane.

Katalogi i pliki GDIR można adresować w programie obróbki w taki sam sposób jak w pasywnym systemie plików. Dzięki temu jest możliwe kompatybilne przemieszczenie programu NC z podaniem ścieżki z pasywnego systemu plików do GDIR. Katalog SPF.DIR w GDIR jest zawarty na ścieżce szukania podprogramu.

Organizacja programu

Poniższy rysunek ma za zadanie zilustrować organizację programów na zewnętrznych pamięciach programów:



System plików case-insensitive

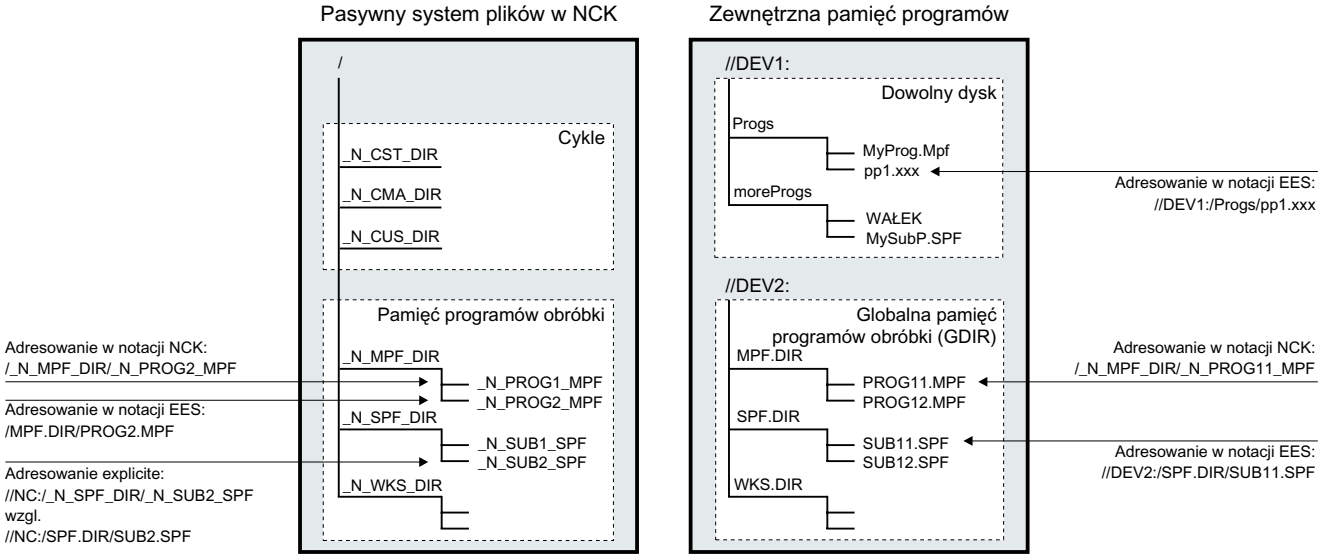
Uwaga

Aby uniknąć problemów w wyniku pisania dużymi i małymi literami przy adresowaniu pliku (patrz "Adresowanie plików pamięci programów (Strona 225)"), **case-insensitive** systemy plików powinny być stosowane jako zewnętrzna pamięć programów.

3.1.3 Adresowanie plików pamięci programów

Odniesienie do pliku w pamięci programów, który zaadresowany został przez polecenie obsługi plików (np. WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO), następuje albo przez ścieżkę absolutną i dodatkowo nazwę pliku albo przez samą nazwę pliku. W drugim przypadku ścieżka wybranego programu jest używana jako ścieżka pliku.

Adresowanie w notacji NC/EES



Adresowanie plików pasywnego systemu plików

Adresowanie plików pasywnego systemu plików następuje zazwyczaj w **notacji NC** (nazwy katalogu i pliku rozpoczynają się od oznaczenia domeny "_N_", znakiem rozdzielającym dla rozszerzenia katalogu/pliku jest "_") bez podania nazwy stacji. Adresowanie w **notacji EES** (bez oznaczenia domeny "_N_", znakiem rozdzielającym dla rozszerzenia katalogu/pliku jest ".") jest także dopuszczalne.

Przykład:

- Notacja NC: "/_N_SPF_DIR/_N_SUB1_SPF"
- Notacja EES: "/SPF.DIR/SUB1.SPF"

Uwaga

Adresowania plików pasywnego systemu plików w notacji EES są wewnętrznie zamieniane w notację NC zgodnie z następującymi zasadami:

- Nazwy katalogów i plików są rozszerzane o oznaczenie domeny "_N_".
- Jeżeli czwartym od końca znakiem w nazwie katalogu lub pliku jest kropka ("."), jest ona zamieniana na podkreślnik ("_").

Domyślnie zdefiniowana nazwa dysku "//NC:" może również odpowiednio adresować pasywny system plików.

Przykład:

- Notacja NC: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- Notacja EES: "//NC:/SPF.DIR/SUB1.SPF"

Adresowanie plików zewnętrznej pamięci programów

Adresowanie plików zewnętrznej pamięci programów, która nie jest wyróżniona jako GDIR, musi następować w notacji EES. Na początku ścieżki adresowej musi zostać podana nazwa dysku (np. "//DEV1:"). Są dopuszczalne wszystkie symboliczne nazwy sprzętowe zaprojektowane w /user/sinumerik/hmi/cfg/logdrive.ini.

Przykład:

- Notacja EES: "//DEV1:/MyProgDir/pp1.xxx"
- Notacja NC: niedopuszczalna

Adresowanie plików globalnej pamięci programów obróbki (GDIR)

Podczas adresowania plików GDIR możliwe jest również określenie ścieżki w zapisie NC oprócz ścieżki określonej w notacji EES.

Przykład:

- Notacja EES: "//DEV2:/MPF.DIR/PROG11.MPF"
- Notacja NC: "/_N_MPF_DIR/_N_PROG11_MPF"

Uwaga

Adresowania plików GDIR w notacji NC są wewnętrznie przekształcane w notację EES zgodnie z następującymi zasadami:

- Oznaczenie domeny "_N_" w nazwach katalogu i pliku jest usuwane.
- Jeżeli czwartym od końca znakiem w nazwie katalogu lub pliku jest podkreślnik ("_"), jest on zamieniany na kropkę (".").

Zasady podawania ścieżek

Kompletne podanie ścieżki składa się z nazwy dysku, ścieżki katalogu i nazwy pliku.

Nazwa dysku

Dla podania nazwy dysku obowiązują następujące zasady:

- Są dopuszczalne wszystkie symboliczne nazwy sprzętowe zaprojektowane w /user/sinumerik/hmi/cfg/logdrive.ini.
- Na początku znajduje się znak "//", po którym następuje co najmniej jedna litera albo jedna cyfra.
- Następnymi znakami mogą być w dowolnej kombinacji litery, cyfry, "_" i spacja.
- Nazwa kończy się literą lub cyfrą, po której następuje ":".
- Inne znaki specjalne są niedozwolone.

Uwaga

Dla pasywnego systemu plików jest zdefiniowana domyślnie nazwa dysku "//NC:".

Przykłady:

- Zewnętrzna pamięć programów:
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

Ścieżka katalogu

Aby określić ścieżkę katalogu, obowiązują następujące reguły:

- Na początku i na końcu ścieżki katalogu i jako znak rozdzielający poszczególnych części ścieżki znajduje się "/".

  Uwaga

  Podwójny ukośnik ("//") w ramach ścieżki katalogu **nie** jest dozwolony!

- Nazwa katalogu:
  - Nazwa katalogu musi rozpoczynać się od litery lub cyfry. Tylko przy adresowaniu w notacji NC nazwy katalogów rozpoczynają się od oznaczenia domeny "_N_".
  - Następnymi znakami mogą być litery, cyfry i "_" w dowolnej kombinacji.

    Uwaga

    Przy zewnętrznych pamięciach programów również spacje są dopuszczalne w nazwie katalogu. Nie obowiązuje to jednak, gdy zewnętrzna pamięć programów jest ustawiona jako globalna pamięć programów obróbki (GDIR).

  - Inne znaki specjalne są niedozwolone.
- Rozszerzenia katalogów:
  - Rozszerzenia katalogów muszą składać się dokładnie z trzech liter/cyfr.
  - Za pomocą "_" (notacja NC) lub "." (notacja EES) są one oddzielane od nazwy katalogu.

  Uwaga

  W pasywnym systemie plików występują tylko rozszerzenia katalogów _DIR i _WPD.

Przykłady:

- Pasywny system plików lub GDIR:
  - Notacja NC: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - Notacja EES: WKS.DIR/MYPROGS.WPD/...
- Zewnętrzna pamięć programów:
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

Nazwa pliku

Dla nazw plików obowiązują następujące zasady:

- Tylko przy adresowaniu w notacji NC nazwy plików rozpoczynają się od oznaczenia domeny "_N_".
- Dwa kolejne znaki powinny składać się z dwóch liter lub podkreślenia poprzedzonego literą.

  Uwaga

  Gdy ten warunek zostanie spełniony, wówczas program NC może zostać wywołany jako podprogram z innego programu wyłącznie poprzez podanie nazwy programu. Jeżeli natomiast nazwa programu rozpoczyna się od cyfr, wówczas wywołanie podprogramu jest możliwe tylko za pomocą instrukcji CALL.

- Następnymi znakami mogą być litery, cyfry i "_" w dowolnej kombinacji.
- Rozszerzenie pliku:
  - Rozszerzenie pliku musi składać się dokładnie z trzech liter/cyfr.

    Uwaga

    Dopuszczalne rozszerzenia plików w pasywnym systemie plików patrz "Pamięć programów w NC (Strona 221)".

  - Za pomocą "_" (notacja NC) lub "." (notacja EES) są one oddzielane od nazwy pliku.

Przykłady:

- Pasywny system plików lub GDIR:
  - Notacja NC: _N_SUB1_SPF
  - Notacja EES: SUB1.SPF
- Zewnętrzna pamięć programów:
  - Teil1
  - _Teil1
  - Teil_1.spf
  - Teil1.mpf

Podprogramy DIN

Dla nazw podprogramów DIN obowiązują następujące zasady:

- Pierwszym znakiem jest litera "L".
- Następnymi znakami są cyfry (co najmniej jedna).
- Rozszerzenie pliku:
  - Rozszerzenie pliku musi składać się dokładnie z trzech liter.
  - Za pomocą "_" (notacja NC) lub "." (notacja EES) są one oddzielane od nazwy pliku.

Przykłady:

- L123
- L1_SPF (notacja NC) lub L1.SPF (notacja EES)

Maksymalna długość ścieżki

Dla podania nazwy dysku i ścieżki katalogu dostępnych jest maksymalnie 128 bajtów, długość nazwy pliku nie może przekraczać 31 bajtów. Dla całej ścieżki daje to długość maksymalną 159 bajtów.

3.1.4 Ścieżka szukania przy wywołaniu podprogramu

Przy przy wywołaniach podprogramu bez podania ścieżki jest określana ścieżka absolutna przez realizację stałej ścieżki szukania.

Pamięci programów są przy tym przeszukiwane w następującej kolejności:

		Katalog	Opis
1		aktualny katalog / *nazwa*	Aktualnym katalogiem jest katalog, w którym nastąpił wybór programu. Może to być: • Katalog obrabianego przedmiotu albo katalog standardowy _N_MPF_DIR w pamięci programów obróbki w NC wzgl. globalnej pamięci programów obróbki. lub • dowolny katalog zewnętrznej pamięci programów
2		aktualny katalog / *nazwa_SPF*	
3		aktualny katalog / *nazwa_MPF*	
4	a	//NC:/_N_SPF_DIR / *nazwa_SPF*	Katalog podprogramów w pamięci programów obróbki w NC
	b	//DEV2:/_N_SPF_DIR / nazwa_SPF [1]	Katalog podprogramów w globalnej pamięci programów obróbki **Wskazówka:** Ten krok szukania nie jest uwzględniany, gdy globalna pamięć programów obróbki nie jest ustawiona albo nastąpił wybór programu w pamięci programów obróbki w NC.
5		Programowane przy pomocy `CALLPATH` rozszerzenie ścieżki szukania (patrz "Rozszerzenie ścieżki szukania przy wywołaniach podprogramu (CALLPATH) (Strona 207)"). **Wskazówka:** Ten krok szukania nie jest uwzględniany, gdy `CALLPATH` nie jest zaprogramowany.	
6		/_N_CUS_DIR / *nazwa_SPF*	Katalog cykli użytkownika
7		/_N_CMA_DIR / *nazwa_SPF*	Katalog cykli producenta
8		/_N_CST_DIR / *nazwa_SPF*	Katalog cykli standardowych

[1] //DEV2:" oznacza na przykład dysk, na którym jest ustawiona globalna pamięć programów obróbki.

Dla szukania obowiązują następujące zasady:

- Przejście ścieżki szukania następuje dla każdego wywołania podprogramu, tzn. nie ma znaczenia, gdzie znajduje się program nadrzędny.
- W zależności od katalogu uwzględniane są różne typy plików.
- Zasadniczo w katalogu nie następuje szukanie w katalogach podrzędnych, tzn. kaskadowanych.

3.1.5 Odpytanie ścieżki i nazwy pliku

Do odpytania ścieżki i nazwy pliku programu NC są do dyspozycji następujące zmienne systemowe odczytywalne w programie obróbki;

Zmienna systemowa	Typ	Znaczenie
$P_STACK	INT	Podaje poziom programu, na którym aktualny program NC jest wykonywany.
$P_PATH[<n>]	STRING	Podaje ścieżkę programu NC, który jest wykonywany na poziomie wybranym przez indeks tablicy <n>. Przykłady: $P_PATH[0] podaje ścieżkę dla programu głównego, np. "/_N_WKS_DIR/_N_WAŁEK_WPD/". $P_PATH[$P_STACK - 1] podaje ścieżkę programu wywołującego. Jeżeli ścieżka odnosi się do programu NC, który jest zapisany w pasywnym systemie plików w NC albo w globalnej pamięci programów obróbki (GDIR), ścieżka jest podawana w notacji NC. Jeżeli ścieżka odnosi się do programu NC, który jest wykonywany z innej zewnętrznej pamięci programów jako globalnej pamięci programów obróbki, $P_PATH podaje ścieżkę w notacji EES.
$P_PROG[<n>]	STRING	Podaje nazwę programu NC, który jest wykonywany na poziomie wybranym przez indeks tablicy <n>. Jeżeli programu NC jest zapisany w pasywnym systemie plików w NC albo w globalnej pamięci programów obróbki, nazwa programu jest podawana w notacji NC. Jeżeli program NC jest wykonywany z innego dysku zewnętrznego niż z globalnej pamięci programów obróbki, $P_PROG podaje nazwę w notacji EES.
$P_PROGPATH	STRING	Podaje ścieżkę programu NC, który jest właśnie wykonywany. Wywołanie $P_PROGPATH jest identyczne z $P_PATH[$P_STACK].

Zmienna systemowa	Typ	Znaczenie	
$P_IS_EES_PATH[<n>]	BOOL	Odpytanie, czy ścieżka podana przez $P_PATH[<n>] wzgl. nazwa programu podana przez $P_PROG[<n>] odpowiada notacji NC czy notacji EES.	
		= FALSE	$P_PATH[<n>] i $P_PROG[<n>] podaje notacje NC. Tzn. każdy identyfikator jest poprzedzony prefiksem "_N_". Znakiem rozdzielającym dla rozszerzenia pliku jest "_". Przykłady: • Ścieżka w notacji NC: "/_N_WKS_DIR/_N_MYWPD_WPD/" • Program w notacji NC:"_N_MYPROG_MPF" Ścieżka w notacji NC może odnosić się zarówno do pasywnego systemu plików w NC jak też do globalnej pamięci programów obróbki.
		= TRUE	$P_PATH[<n>] i $P_PROG[<n>] dają notację EES. Tzn. identyfikatory nie są poprzedzone prefiksem "_N_". Znakiem rozdzielającym dla rozszerzenia pliku jest ".". Przykłady: • Ścieżka w notacji EES: "//DEV1:/WKS.DIR/MYWPD.WPD/" • Nazwa programu w notacji EES: MYPROG.MPF

<n>: Indeks <n> definiuje poziom programu, z którego informacja o ścieżce ma być czytana (zakres wartości: 0 ... 17)

Uwaga

W trybie EES, poza globalną pamięcią programów obróbki (GDIR), zmienne systemowe $P_PROG, $P_PATH, i $P_PROGPATH dają nazwy ścieżek w notacji EES. Programy użytkownika, które te nazwy ścieżek analizują i przetwarzają, muszą dlatego dla trybu EES zostać tak rozbudowane, by mogły przetwarzać również nazwy ścieżek w notacji EES.

3.2 Pamięć robocza (CHANDATA, COMPLETE, INITIAL)

Funkcja

Pamięć robocza zawiera aktualne dane systemowe i użytkownika, z którymi sterowanie jest eksploatowane (aktywny system plików), np.:

- Aktywne dane maszynowe
- Dane korekcyjne narzędzia
- Przesunięcia punktu zerowego
- ...

Programy inicjalizacyjne

Chodzi tutaj o programy, przy pomocy których dane pamięci roboczej są domyślnie ustawiane (inicjalizowane). Mogą w tym celu być stosowane następujące typy plików:

Typ pliku	Opis
nazwa_TEA	Dane maszynowe
nazwa_SEA	Dane ustawcze
nazwa_TOA	Korekcje narzędzi
nazwa_UFR	Przesunięcia punktu zerowego/frame
nazwa_INI	Plik inicjalizacyjny
nazwa_GUD	Globalne dane użytkownika
nazwa_RPA	Parametry R

Zakresy danych

Dane mogą być włączone do różnych zakresów, w których mają obowiązywać. Na przykład sterowanie może dysponować wieloma kanałami albo jak zazwyczaj wieloma osiami.

Są:

Oznaczenie	Zakresy danych
NC	Dane specyficzne dla NC
CH<n>	Dane specyficzne dla kanału (<n> podaje numer kanału)
AX<n>	Dane specyficzne dla osi (n podaje numer osi maszyny)
TO	Dane narzędzi
COMPLETE	Wszystkie dane

Utworzenie programu inicjalizacyjnego na zewnętrznym PC

Przy pomocy identyfikacji zakresów danych i identyfikacji typów danych mogą zostać określone obszary, które przy zapisywaniu danych są traktowane jako jednostka:

_N_AX5_TEA_INI	Dane maszynowe dla 5-ej osi
_N_CH2_UFR_INI	Frame kanału 2
_N_COMPLETE_TEA_INI	Wszystkie dane maszynowe

Po uruchomieniu sterowania w pamięci roboczej jest zestaw danych, który gwarantuje jego należytą pracę.

Sposób postępowania w przypadku sterowań wielokanałowych (CHANDATA)

`CHANDATA(<numer kanału>)` dla wielu kanałów jest dopuszczalne tylko w pliku _N_INITIAL_INI. Jest to plik uruchomieniowy, przy pomocy którego są inicjalizowane wszystkie dane sterowania.

Kod programu	**Komentarz**
`%_N_INITIAL_INI`	
`CHANDATA(1)`	
	`; Przyporządkowanie osi maszyny kanał 1:`
`$MC_AXCONF_MACHAX_USED[0]=1`	
`$MC_AXCONF_MACHAX_USED[1]=2`	
`$MC_AXCONF_MACHAX_USED[2]=3`	
`CHANDATA(2)`	
	`; Przyporządkowanie osi maszyny kanał 2:`
`$MC_AXCONF_MACHAX_USED[0]=4`	
`$MC_AXCONF_MACHAX_USED[1]=5`	
`CHANDATA(1)`	
	`; Osiowe dane maszynowe:`
	`; Okno zatrzymania dokładnego zgrubnie:`
`$MA_STOP_LIMIT_COARSE[AX1]=0.2`	`; Oś 1`
`$MA_STOP_LIMIT_COARSE[AX2]=0.2`	`; Oś 2`
	`; Okno zatrzymania dokładnego dokładnie:`
`$MA_STOP_LIMIT_FINE[AX1]=0.01`	`; Oś 1`
`$MA_STOP_LIMIT_FINE[AX1]=0.01`	`; Oś 2`

UWAGA

Instrukcja CHANDATA

W programie obróbki wolno umieścić instrukcję `CHANDATA` tylko dla tego kanału, w którym jest wykonywany program NC. Tzn. instrukcja może zostać użyta do tego, by chronić programy przed wykonywaniem w nieprzewidzianym kanale.

W przypadku błędu wykonywanie programu jest przerywane.

Uwaga

Pliki INI na listach zadań nie zawierają instrukcji `CHANDATA`.

Wykonanie kopii zapasowej programów inicjalizacyjnych (COMPLETE, INITIAL)

Pliki pamięci roboczej mogą zostać zapisane na zewnętrznym PC i być stamtąd wczytywane.

- Pliki są zapisywane przy pomocy `COMPLETE`.
- Przy pomocy `INITIAL` jest dla wszystkich obszarów tworzony plik INI (_N_INITIAL_INI).

Wczytanie programów inicjalizacyjnych

UWAGA
Utrata danych Gdy zostanie wczytany plik o nazwie "INITIAL_INI", wówczas wszystkie dane, które nie są dostarczane w tym pliku, są inicjalizowane jako dane standardowe. Wyłączone z tego są tylko dane maszynowe. Jako standardowe są więc dostarczane **dane ustawcze, dane narzędzi, PPZ, wartości GUD, ...** (zazwyczaj "ZERO").

Do wczytywania poszczególnych danych maszynowych nadaje się np. plik COMPLETE_TEA_INI. W tym pliku sterowanie oczekuje tylko danych maszynowych. Inne obszary danych pozostają w tym przypadku nienaruszone.

Ładowanie programów inicjalizacyjnych

Programy INI mogą być również wybierane i wywoływane jako programy obróbki, gdy stosują tylko dane jednego kanału. W ten sposób jest również możliwe inicjalizowanie danych sterowanych programem.

Obszary ochrony

4

4.1 Definiowanie obszarów ochrony (CPROTDEF, NPROTDEF)

Obszary ochrony, które powinny ochraniać elementy maszyny przed kolizją, definiowane są w programie obróbki za każdym razem w blokach. Zawierają one następujące elementy:

1. Określenie płaszczyzny roboczej
   Przed właściwą definicją obszaru ochrony należy wybrać płaszczyznę roboczą, do której będzie się odnosił opis konturu obszaru ochrony.
2. Początek definicji
   W zależności od polecenia NC tworzony jest kanałowy lub maszynowy obszar ochrony.
3. Opis konturu obszaru ochrony
   Kontur obszaru ochrony jest opisywany za pomocą ruchów postępowych. Nie są one wykonywane i nie mają żadnego związku z poprzednimi lub następnymi opisami geometrii. Definiowany jest tylko obszar ochrony.
4. Koniec definicji

Składnia

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/... X/Y/Z...
...
EXECUTE(<Var>)
```

Znaczenie

`DEF INT <Var>`:	Definicja lokalnej zmiennej pomocniczej o typie danych INTEGER	
`<Var>`:	Nazwa zmiennej pomocniczej	
`G17/G18/G19`:	Płaszczyzna robocza **Wskazówka:** Płaszczyzny roboczej nie wolno zmieniać przed końcem definicji. Programowanie aplikaty między początkiem i końcem definicji jest niedozwolone.	
`CPROTDEF()`:	Wstępnie zdefiniowana procedura definicji obszaru ochrony specyficznego dla **kanału**	
`NPROTDEF()`:	Wstępnie zdefiniowana procedura definicji obszaru ochrony specyficznego dla **maszyny**	
`<n>`:	Numer zdefiniowanego obszaru ochrony	
	Typ danych:	INT

<t>:	Typ obszaru ochrony:		
	Typ danych:	BOOL	
	Wartość:	TRUE	Obszar ochrony odniesiony do **narzędzia**
		FALSE	Obszar ochrony odniesiony do **obrabianego przedmiotu**
<AppLim>:	Rodzaj ograniczenia w 3. wymiarze		
	Typ danych:	INT	
	Wartość:	0	Nie ma ograniczenia
		1	Ograniczenie w kierunku dodatnim
		2	Ograniczenie w kierunku ujemnym
		3	Ograniczenie w kierunku dodatnim i ujemnym
<AppPlus>:	Wartość ograniczenia w kierunku dodatnim 3. wymiaru		
	Typ danych:	REAL	
<AppMinus>:	Wartość ograniczenia w kierunku ujemnym 3. wymiaru		
	Typ danych:	REAL	

`G0/G1/... X/Y/Z... ...:`	Kontur obszaru ochrony jest opisany maksymalnie 11 ruchami postępowymi na wybranej płaszczyźnie. Pierwszym ruchem jest ruch do konturu. Ostatni punkt opisu konturu musi zawsze pokrywać się z pierwszym punktem opisu konturu. Jako obszar ochrony obowiązuje obszar na lewo od konturu: • Wewnętrzny obszar ochrony Kontur dla wewnętrznego obszaru ochrony należy opisać przeciwnie do ruchu wskazówek zegara. • Zewnętrzny obszar ochrony (dozwolony tylko dla obszarów ochrony odniesionych do przedmiotu obrabianego!) Kontur dla zewnętrznego obszaru ochrony należy opisać zgodnie z ruchem wskazówek zegara. Dopuszczalne są następujące elementy konturu: • `G0`, `G1` dla prostych elementów konturu • `G2` dla segmentów kołowych w kierunku ruchu wskazówek zegara Dozwolone tylko dla obszarów ochrony odniesionych do przedmiotu obrabianego! Niedozwolone w obszarach ochrony odniesionych do narzędzia, ponieważ mogą być tylko wypukłe. • `G3` dla segmentów kołowych przeciwnie do ruchu wskazówek zegara **Wskazówka:** Obszar ochrony nie może być opisywany za pomocą pełnego okręgu. Pełny okrąg należy podzielić na dwa półokręgi. **Wskazówka:** Sekwencja `G2`, `G3` wzgl. `G3`, `G2` jest niedozwolona. Między dwoma blokami kołowymi należy wstawić krótszy blok `G1`.
`EXECUTE(<Var>):`	Wstępnie zdefiniowana procedura, która oznacza koniec definicji. Za pomocą `EXECUTE` następuje powrót do normalnej obróbki programu.

Przykład

Patrz przykład „Uaktywnienie / wyłączenie aktywności obszarów ochrony (CPROT, NPROT) (Strona 241)".

Dodatkowe informacje

Obszary ochrony specyficzne dla maszyny

Obszar ochrony specyficzny dla maszyny lub jego kontur jest definiowany za pomocą osi geometrycznych, tzn. w odniesieniu do bazowego układu współrzędnych (BKS) kanału. Aby mogła się odbywać prawidłowa kontrola obszaru ochrony we wszystkich kanałach w których aktywny jest obszar ochrony specyficzny dla maszyny, bazowy układ współrzędnych (BKS) wszystkich kanałów musi być taki sam.

- Położenie środka układu współrzędnych w odniesieniu do punktu zerowego maszyny
- Orientowanie osi współrzędnych

Punkt odniesienia opisu konturu

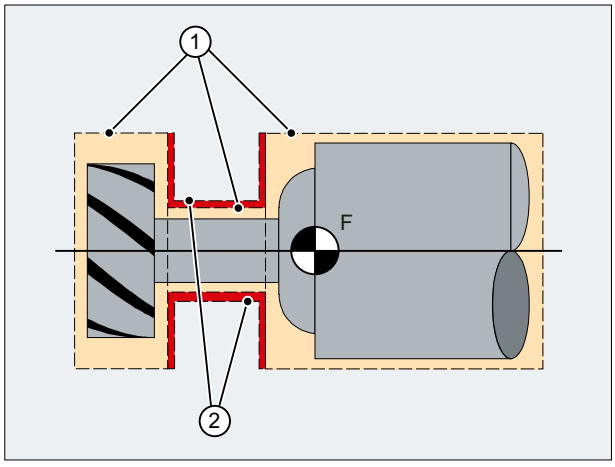
- Obszary ochrony odniesione do narzędzia:
  Współrzędne obszarów ochrony odniesionych do **narzędzia** są podawane absolutnie, w odniesieniu do **punktu odniesienia nośnika narzędzia F**.
- Obszary ochrony odniesione do obrabianego przedmiotu
  Współrzędne obszarów ochrony odniesionych do **przedmiotu obrabianego** podawane są absolutnie, w odniesieniu do **bazowego układu współrzędnych** (BKS).

Obszary ochrony obrotowo-symetryczne

W przypadku obrotowo-symetrycznych obszarów ochrony (np. uchwyt tokarski) należy opisać cały kontur, a nie tylko kontur do osi toczenia.

Obszary ochrony odniesione do narzędzia:

Obszary ochrony odniesione do narzędzia muszą być zawsze wypukłe. Jeżeli jest żądany wklęsły obszar ochrony, należy go podzielić na wiele obszarów wypukłych.



① Wypukłe obszary ochrony
② Wklęsłe obszary ochrony (**niedozwolone**)
F Punkt odniesienia nośnika narzędzi

Warunki brzegowe

Podczas definiowania obszaru ochrony następujące funkcje nie mogą być aktywne lub w użyciu:

- Korekcja promienia narzędzia (korekcja promienia frezu, korekcja promienia ostrza)
- Transformacja
- Najazd na punkt referencyjny (`G74`)
- Najazd na punkt stały (`G75`)
- Czas oczekiwania (`G4`)
- Zatrzymanie przebiegu wyprzedzającego (`STOPRE`)
- Koniec programu (`M17`, `M30`)
- Funkcje M: `M0`, `M1`, `M2`

4.2 Uaktywnienie / wyłączenie aktywności obszarów ochrony (CPROT, NPROT)

Wcześniej zdefiniowane w programie obróbki obszary ochrony mogą być w każdym momencie aktywowane lub wstępnie aktywowane w celu późniejszej ich aktywacji przez program użytkownika PLC. Aktywne obszary ochrony mogą być w każdej chwili dezaktywowane.

Podczas aktywacji lub aktywacji wstępnej istnieje możliwość względnego przesunięcia punktu odniesienia obszaru ochrony.

Uwaga

Obszar ochrony jest uwzględniany dopiero po wybazowniu wszystkich osi geometrycznych kanału, w których był on aktywny.

Uwaga

Kontrola obszarów ochrony

Jeżeli żaden obszar ochrony odniesiony do narzędzia nie jest aktywny, to wówczas tor ruchu narzędzia jest sprawdzany w stosunku do obszarów ochrony odniesionych do obrabianego przedmiotu.

Jeżeli żaden obszar ochrony odniesiony do obrabianego przedmiotu nie jest aktywny, to nie następuje nadzór obszarów ochrony.

Składnia

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

Znaczenie

CPROT:	Wstępnie zdefiniowana procedura aktywacji obszaru ochrony specyficznego dla **kanału**		
NPROT:	Wstępnie zdefiniowana procedura aktywacji obszaru ochrony specyficznego dla **maszyny**		
<n>:	Numer obszaru ochrony		
	Typ danych:	INT	
<Status>:	Za pomocą tego parametru jest ustawiany specyficzny dla kanału status aktywacji.		
	Typ danych:	INT	
	Wartość:	0	Wyłączenie aktywności obszaru ochrony
		1	Wstępne uaktywnienie obszaru ochrony
		2	Aktywowanie obszaru ochrony
		3	Wstępne uaktywnienie obszaru ochrony z warunkowym zatrzymaniem

<XMov>,<YMov>,<ZMov>:	Addytywne wartości przesunięcia w kierunku X/Y/Z Przesunięcie może nastąpić w 1, 2 albo 3 wymiarach. Wartości przesunięcia odnoszą się do: • punktu zerowego maszyny dla obszaru ochrony odniesionego do **przedmiotu obrabianego** • punktu odniesienia nośnika narzędzi F dla obszarów ochrony odniesionych do **narzędzia**	
	Typ danych:	REAL

Przykład

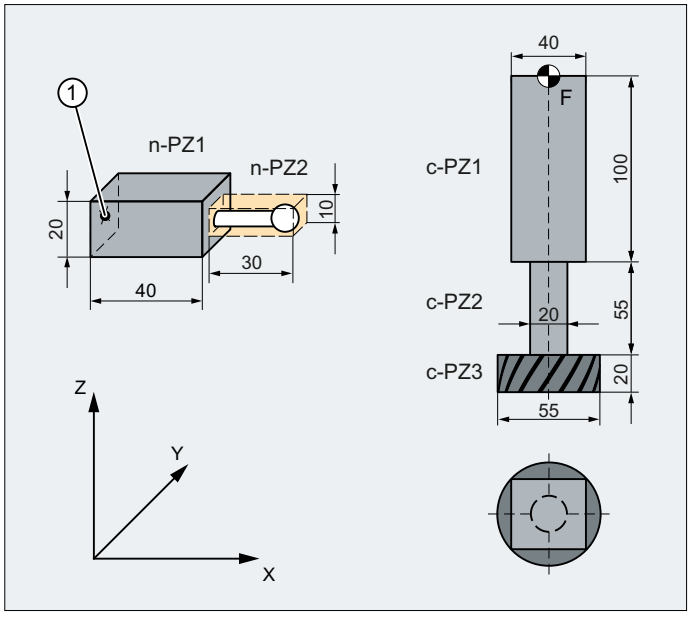
Dla frezarki ma być prowadzony nadzór na możliwą kolizję frezu z sondą pomiarową. Położenie sondy pomiarowej powinno przy uaktywnieniu zostać podane przez przesunięcie.

W tym celu definiowane są następujące obszary ochrony:

- Każdorazowo jeden specyficzny dla maszyny i odniesiony do obrabianego przedmiotu obszar ochrony dla uchwytu sondy pomiarowej (n-PZ1) i dla samej sondy pomiarowej (n-PZ2).
- Każdorazowo specyficzny dla kanału i odniesiony do narzędzia obszar ochrony dla uchwytu frezu (c-PZ1), chwytu frezu (c-PZ2) i dla samego frezu (c-PZ3).

Wszystkie obszary ochrony są zorientowane w kierunku Z.

Położenie punktu odniesienia czujnika pomiarowego przy uaktywnieniu niech wynosi X = -120, Y = 60 i Z = 80.



① Punkt odniesienia obszaru ochrony czujnika pomiarowego

F Punkt odniesienia nośnika narzędzi

Kod programu	**Komentarz**
DEF INT PROTZONE	; Definicja zmiennej pomocniczej
G17	; Płaszczyzna robocza X/Y
; Definicja obszarów ochrony:	
NPROTDEF(1,FALSE,3,10,-10)	; Obszar ochrony n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE(PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	; Obszar ochrony n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE(PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	; Obszar ochrony c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE(PROTZONE)	
CPROTDEF(2,TRUE,3,-100,-150)	; Obszar ochrony c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; Obszar ochrony c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE(PROTZONE)	
; Aktywacja obszarów ochrony:	
NPROT(1,2,-120,60,80)	; Obszar ochrony n-PZ1 z aktywnym przesunięciem
NPROT(2,2,-120,60,80)	; Obszar ochrony n-PZ2 z aktywnym przesunięciem
CPROT(1,2,0,0,0)	; Aktywacja obszaru ochrony c-PZ1
CPROT(2,2,0,0,0)	; Aktywacja obszaru ochrony c-PZ2
CPROT(3,2,0,0,0)	; Aktywacja obszaru ochrony c-PZ3

Dodatkowe informacje

Status aktywacji po rozruchu sterowania

Obszar ochrony może być aktywny zaraz po rozruchu sterowania i bazowaniu osi. Dzieję się tak wtedy, gdy dla obszaru ochrony następująca zmienna systemowa jest ustawiona na TRUE:

- $SN_PA_ACTIV_IMMED[<n>] (dla obszarów ochrony specyficznych dla maszyny) lub
- $SC_PA_ACTIV_IMMED[<n>] (dla obszarów ochrony specyficznych dla kanału)
  Indeks "<n>" odpowiada numerowi obszaru ochrony: 0 = 1. Obszar ochrony

Obszar ochrony jest aktywowany ze statusem = 2 i bez przesunięcia.

Wielokrotna aktywacja obszaru ochrony

Obszar ochrony specyficzny dla maszyny może działać równocześnie w wielu kanałach (np. obszar ochrony pinoli w przypadku dwóch naprzeciwległych suportów). Nadzór obszarów ochrony następuje tylko wtedy, gdy nastąpiło bazowanie wszystkich osi geometrycznych do punktu odniesienia.

Obszaru ochrony nie da się aktywować w tym samym czasie w jednym kanale z różnymi przesunięciami.

Nadzór obszaru ochrony przy aktywnej korekcji promienia narzędzia

Przy aktywnej korekcji promienia narzędzia działanie nadzoru obszaru ochrony jest możliwe tylko wtedy, gdy płaszczyzna korekcji promienia narzędzia jest identyczna z płaszczyzną definicji obszaru ochrony.

4.3 Sprawdzenie naruszenia obszaru ochrony, ograniczenia obszaru pracy oraz programowych wyłączników krańcowych (CALCPOSI).

Funkcja

Funkcja CALCPOSI sprawdza w układzie współrzędnych przedmiotu obrabianego (WKS), czy od danego punktu początkowego **osie geometryczne** mogą wykonać ruch po zadanej drodze bez naruszenia aktywnych ograniczeń. W przypadku, gdy droga ruchu nie może zostać całkowicie przebyta z powodu ograniczeń, zwracana jest dodatnia, dziesiętnie kodowana wartość statusu oraz maksymalna możliwa droga ruchu.

Definicja

```
INT CALCPOSI(VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

Składnia

```
<Status> = CALCPOSI(VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

Znaczenie

`CALCPOSI(...):`	Wstępnie zdefiniowana funkcja do testu naruszenia ograniczeń dla osi geometrycznych	
	Zatrzymanie przebiegu wyprzedzającego:	nie
	W oddzielnym bloku:	tak

<status>: (Część 1)	Wartość zwrotna funkcji. Wartości ujemne wskazują stany błędu.		
	Typ danych:		INT
	Zakres wartości:		-8 ≤ x ≤ 100000
	Wartość:	0	Można całkowicie przebyć drogę ruchu
		-1	W <Limit> przynajmniej jedna składowa jest ujemna
		-2	Błędy przy obliczaniu transformacji. Przykład: Droga ruchu prowadzi przez punkt osobliwy, tak że pozycje osi nie są zdefiniowane.
		-3	Podana droga ruchu <Dist> i maksymalna możliwa droga ruchu do przebycia <MaxDist> są zależne liniowo. **Uwaga** Może wystąpić tylko razem z <TestLim>, Bit 4 == 1.
		-4	Rzutem kierunku ruchu znajdującego się w <Dist> na powierzchnię ograniczającą jest wektor zerowy lub kierunek ruchu prostopadły względem naruszonej powierzchni ograniczającej. **Uwaga** Może wystąpić tylko razem z <TestLim>, Bit 5 == 1.
		-5	W <TestLim> są Bit 4 == 1 AND Bit 5 == 1
		-6	Co najmniej jedna oś maszyny, która służy do kontroli granic ruchu, nie jest wybazowana.
		-7	Funkcja unikania kolizji: Nieważna definicja łańcucha kinematycznego bądź obszarów ochronnych.
		-8	Funkcja unikania kolizji: Funkcja nie może być wykonana z powodu braku pamięci.
<status>: (Część 2)	Miejsce jednostek: **Uwaga** Jeśli równocześnie naruszono kilka granic, zgłaszana jest ta, która prowadzi do największego ograniczenia wyznaczonej drogi ruchu.		
	Wartość:	1	Krańcowe wyłączniki programowe ograniczają drogę ruchu.
		2	Ograniczenie obszaru pracy ogranicza drogę ruchu
		3	Obszary ochrony ograniczają drogę ruchu.
		4	Funkcja unikania kolizji: Obszary ochrony ograniczają drogę ruchu.
	Miejsce dziesiątek		
	Wartość:	1x	Wartość początkowa narusza granicę
		2x	Zadana prosta narusza granicę. Wartość ta jest również zwracana wtedy, jeśli sam punkt końcowy nie narusza granicy, ale naruszenie wartości granicznej nastąpiłoby na drodze od punktu początkowego do końcowego (np. przejście przez obszar ochrony, zakrzywione krańcowe wyłączniki programowe w WKS przy transformacjach nieliniowych, np. Transmit).

<status>: (Część 3)	Miejsce setek		
	Wartość:	1xx	AND miejsce jednostek == 1 lub 2: Naruszono dodatnią wartość graniczną.
			AND miejsce jednostek == 3 [1]: Naruszono specyficzny dla NC obszar ochrony.
		2xx	AND miejsce jednostek == 1 lub 2: Naruszono ujemną wartość graniczną.
			AND miejsce jednostek == 3 [1]: Jest naruszony obszar ochrony specyficzny dla kanału.
<status>: (Część 4)	Miejsce tysięcy		
	Wartość:	1xxx	AND miejsce jednostek == 1 lub 2: Współczynnik, przez który jest mnożony numer osi, która narusza granicę. Liczenie osi zaczyna się od 1. Odniesienie: • Programowy wyłącznik krańcowy: Osie maszyny • Ograniczenie obszaru pracy: Osie geometryczne AND miejsce jednostek == 3 [1]: Współczynnik, przez który mnożony jest numer naruszonego obszaru ochrony.
<status>: (Część 5)	Miejsce setek tysięcy		
	Wartość:	0xxxxx	Miejsce setek tysięcy == 0: <Dist> pozostaje niezmieniona
		1xxxxx	W <Dist> cofnięty zostaje jeden wektor kierunku, który definiuje dalszy kierunek ruchu na powierzchni ograniczającej. Może występować tylko z następującymi ograniczeniami: • Naruszenie programowych wyłączników krańcowych lub ograniczenia obszaru pracy (nie w punkcie startu) • Transformacja **nie** jest aktywna. • <TestID>, Bit 4 lub Bit 5 == 1
<Start>:	Odniesienie dla wektora z pozycjami startu: • <Start> [0]: 1. oś geometryczna • <Start> [1]: 2. oś geometryczna • <Start> [2]: 3. oś geometryczna		
	Typ parametru:		Wejście
	Typ danych:		VAR REAL [3]:
	Zakres wartości:		-maks. wartość REAL ≤ x[n] ≤ + maks. wartość REAL

<Dist>:	Odniesienie do wektora.	
	Wejście: przyrostowa droga ruchu • <Dist> [0]: 1. oś geometryczna • <Dist> [1]: 2. oś geometryczna • <Dist> [2]: 3. oś geometryczna	
	Wyjście](tylko przy ustalonej pozycji stu tysięcy w <status>):	
	<Dist> zawiera wektor jednostkowy **v** jako wartość wyjściową, który definiuje dalszy kierunek ruchu w WKS.	
	Przypadek 1: Tworzenie wektora **v** przy <TestID>, Bit 4 == 1 Wektory sterowania <Dist> i <MaxDist> tworzą płaszczyznę ruchu. Płaszczyzna ta zostanie przecięta naruszoną powierzchnią ograniczającą. Linia przecięcia obu płaszczyzn określa kierunek wektora **v**. Orientacja (znak) jest wybierana tak, aby kąt pomiędzy wektorem wejściowym <MaxDist> i **v** był nie większy, niż 90 stopni.	
	Przypadek 2: Tworzenie wektora **v** przy <TestID>, Bit 5 == 1 Wektor **v** jest wektorem jednostkowym w kierunku rzutu wektora ruchu zawartego w <Dist> na powierzchnię ograniczającą. Jeżeli rzut wektora ruchu na powierzchnię ograniczającą jest wektorem zerowym, zostanie zwrócony błąd.	
	Typ parametru:	Wejście/wyjście
	Typ danych:	VAR REAL [3]:
	Zakres wartości:	-Maks. wartość REAL ≤ x[n] ≤ + maks. wartość REAL
<Limit>:	Odniesienie do tablicy o wymiarze 5. • <Limit> [0 - 2]: Minimalne odległości osi geometrycznych do ograniczeń: – <Limit> [0]: 1. oś geometryczna – <Limit> [1]: 2. oś geometryczna – <Limit> [2]: 3. oś geometryczna Minimalne odległości są zachowywane przy: – Ograniczenie obszaru pracy: Brak ograniczeń – Programowy wyłącznik krańcowy: Gdy żadna transformacja nie jest aktywna lub aktywna jest transformacja, w której możliwe jest jednoznaczne przyporządkowanie osi geometrycznych do osi liniowych maszyny, np. transformacje 5-osiowe. • <Limit> [3]: Zawiera minimalny odstęp dla liniowych osi maszyny, które np. z powodu transformacji nieliniowej nie można przypisać do żadnej osi geometrycznej. Ta wartość jest również używana jako wartość graniczna przy nadzorze konwencjonalnych obszarów ochrony oraz obszarów ochrony służących do unikania kolizji. • <Limit> [4]: Zawiera minimalny odstęp dla obrotowych osi maszyny, które np. z powodu transformacji nieliniowej nie można przypisać żadnej osi geometrycznej. **Uwaga** Wartość ta działa tylko w przypadku nadzoru programowych włączników krańcowych transformacji specjalnych.	
	Typ parametru:	Wejście
	Typ danych:	VAR REAL [5]:
	Zakres wartości:	-maks. wartość REAL ≤ x[n] ≤ + maks. wartość REAL

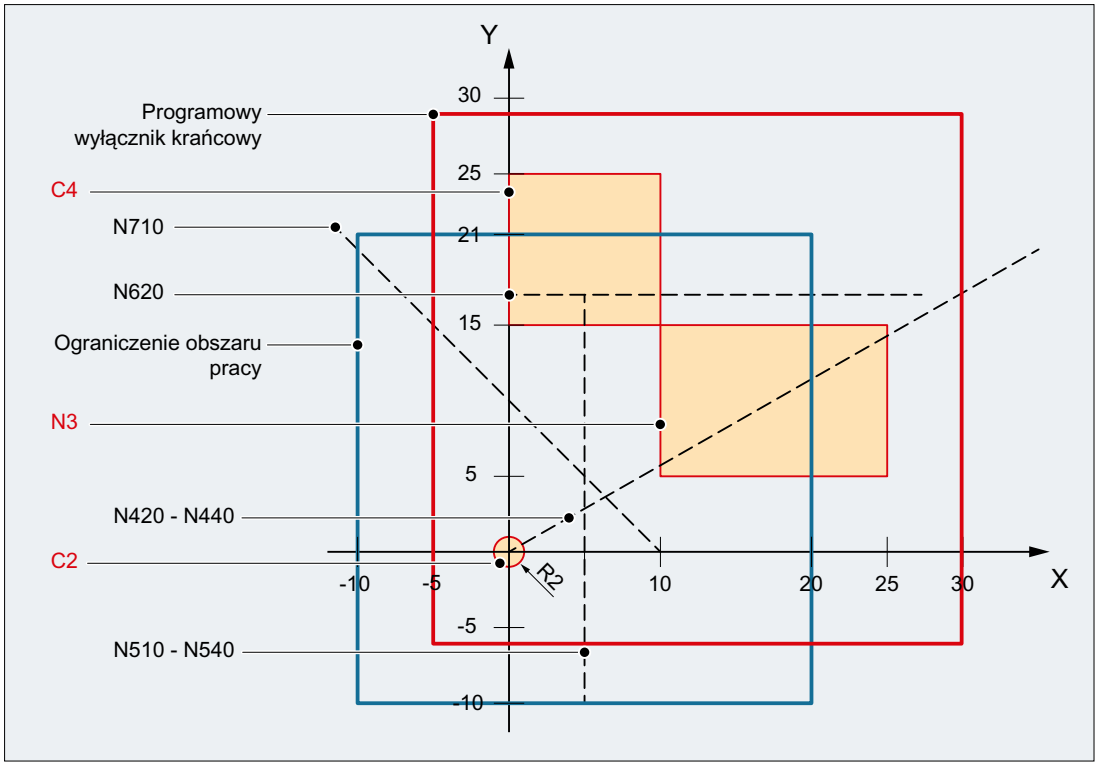
<MaxDist>:	Odniesienie do wektora z przyrostową drogą ruchu, w przypadku którego zadany minimalny odstęp od granicy osi wszystkich uczestniczących osi maszyny nie został osiągnięty: • <Dist> [0]: 1. oś geometryczna • <Dist> [1]: 2. oś geometryczna • <Dist> [2]: 3. oś geometryczna Jeżeli droga ruchu nie jest ograniczona, zawartość tego parametru zwrotnego jest równa zawartości <Dist>.		
	Przy <TestID>, Bit 4 == 1: <Dist> i <MaxDist> <MaxDist> i <Dist> jako wartości wejściowe muszą zawierać wektory, które definiują płaszczyznę ruchu. Oba wektory muszą być liniowo niezależne od siebie. Wartość <MaxDist> jest dowolna. W celu obliczenia kierunku ruchu patrz opis dla <Dist>.		
	Typ parametru:		Wyjście
	Typ danych:		VAR REAL [3]:
	Zakres wartości:		-maks. wartość REAL ≤ x[n] ≤ + maks. wartość REAL
<MeasSys>:	System miar (calowy / metryczny) dla danych pozycji i długości (**opcjonalnie**)		
	Typ danych:		BOOL
	Wartość:	FALSE (domyślnie)	System miar odpowiednio do aktualnie aktywnego polecenia w grupie G nr 13 (G70, G71, G700, G710). **Uwaga** Przy aktywnym G70 i metrycznym systemie podstawowym lub aktywnym G71 i calowym systemie podstawowym zmienne systemowe $AA_IW i $AA_MW są podawane w systemie podstawowym i w celu zastosowania przez funkcję CALCPOSI muszą zostać przeliczone.
		TRUE	System miar zgodnie z ustawionym systemem podstawowym: MD52806 $MN_ISO_SCALING_SYSTEM

<TestLim>:	Kodowany bitowo wybór nadzorowanych ograniczeń (**opcjonalnie**)		
	Typ danych:		INT
	Wartość domyślna:		Bit 0, 1, 2, 3, 6, 7 == 1 (207)
	Bit	**Dziesiętnie**	**Znaczenie**
	0	1	Programowy wyłącznik krańcowy
	1	2	Ograniczenie obszaru pracy
	2	4	Aktywne konwencjonalne obszary ochrony
	3	8	Wstępnie aktywowane konwencjonalne obszary ochrony
	4	16	Przy uszkodzonych programowych wyłącznikach krańcowych lub ograniczeniach obszaru pracy w <Dist> kierunek ruchu należy zwrócić odpowiednio do **przypadku 1** (patrz wyżej).
	5	32	Przy uszkodzonych programowych wyłącznikach krańcowych lub ograniczeniach obszaru pracy w <Dist> kierunek ruchu należy zwrócić odpowiednio do **przypadku 2** (patrz wyżej).
	6	64	Aktywne obszary ochrony dla unikania kolizji
	7	128	Wstępnie aktywowane obszary ochrony dla unikania kolizji
	8	256	Pary aktywnych oraz wstępnie aktywowanych obszarów ochrony dla unikania kolizji

[1] Jeśli naruszonych jest wiele obszarów ochrony, zgłoszony zostanie obszar ochrony, który prowadzi do największego ograniczenia zadanej drogi ruchu.

Przykład

Ograniczenia



W przykładzie przedstawiono skuteczny programowy wyłącznik krańcowy i ograniczenia obszaru pracy w płaszczyźnie X-Y oraz następujące trzy obszary ochrony:

- C2: Obszar ochrony odniesiony do narzędzia, specyficzny dla kanału, aktywny, kołowy, promień = 2 mm
- C4: Obszar ochrony odniesiony do przedmiotu obrabianego, specyficzny dla kanału, wstępnie aktywowany, kwadratowy, długości boków = 10 mm
- N3: Obszar ochrony specyficzny dla maszyny, aktywny, prostokątny, długości boków = 10 mm x 15 mm

Program NC

W programie NC definiowane są najpierw obszary ochrony i ograniczenia obszaru pracy. Następnie będzie wywołana funkcja `CALCPOSI()` z różnymi parametrami.

Kod programu

```
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
```

Kod programu

```
; Obszar ochrony odniesiony do narzędzia C2
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; Obszar ochrony odniesiony do przedmiotu obrabianego C4
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; Obszar ochrony specyficzny dla maszyny N3
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
; Aktywacja lub wstępna aktywacja obszarów ochrony
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; Zdefiniowanie ograniczeń obszaru pracy
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.
```

Kod programu

```
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)
; Frame z N690 i N700 ponownie skasować
N720 TRANS
N730 _START[0] =  0.
N740 _START[1] = 10.
N750 _START[2] =  0.
; Wektory _DIST i _MAXDIST definiują płaszczyznę ruchu
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
```

Wynik CALCPOSI()

N...	<Status>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	Uwagi
420	3123	8.040	4.594	N3 jest naruszony.
430	1122	20.000	11.429	Brak nadzoru nad obszarem ochrony, ograniczenie obszaru obróbki jest naruszone.
440	1121	30.000	17.143	Jest aktywny już tylko nadzór programowego wyłącznika krańcowego.
510	4213	0.000	0.000	Punkt startu narusza C4
520	0000	0.000	-27.000	Wstępnie aktywowany C4 nie jest nadzorowany. Zadaną drogę można całkowicie przebyć.
540	2222	0.000	-25.000	Z powodu _LIMIT[1]=2 droga ruchu jest ograniczona przez ograniczenie obszaru pracy.

N...	<Status>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	Uwagi
620	4223	-13.000	0.000	Odstęp od C4 z powodu C2 i _LIMIT[3] wynosi 4 mm. Odstęp C2 → N3 wynoszący 0,1 mm nie prowadzi do ograniczenia drogi ruchu.
710	1221	0.000	21.213	Aktywne frame z przesunięciem i obrotem Dopuszczalna droga ruchu w _DIST obowiązuje w przesuniętym i obróconym WKS.
820	102121	18.000	18.000	Programowy wyłącznik krańcowy osi Y jest naruszony. Za pomocą <_TESTLIM> = 17 żądane jest obliczenie kierunku dalszego ruchu. Kierunek jest w <_DIST> (0.707, 0.0, 0.707). Jest on obowiązujący, ponieważ w <_STATUS> jest ustawiona pozycja stutysięczna.

Dodatkowe informacje

Status osi „Bazowania”

Wszystkie rozpatrywane osie maszyny w `CALCPOSI()` muszą być wybazowane.

Okręgi z podaniem drogi

Wszystkie okręgi z podaniem drogi są **zawsze** interpretowane jak z podaniem promienia. Należy to uwzględnić w szczególności w przypadku osi poprzecznych z aktywnym programowaniem na średnicy (`DIAMON` / `DIAM90`).

Redukcja drogi ruchu

Jeżeli określona droga ruchu osi jest ograniczona, w wartości zwrotnej `<MaxDist>` drogi ruchu innych osi są proporcjonalnie zmniejszane. W rezultacie wynikowy punkt końcowy leży nadal na zaprogramowanym torze.

Osie obrotowe

Osie obrotowe są nadzorowane tylko wtedy, gdy nie są osiami obrotowymi modulo.

Dopuszczalne jest, aby żadne programowe wyłączniki krańcowe, ograniczenia obszaru obróbki lub obszary ochrony nie były zdefiniowane dla jednej lub wielu uczestniczących osi.

Status programowego wyłącznika krańcowego i ograniczenia obszarów pracy

Programowy wyłącznik krańcowy i ograniczenia obszaru obróbki są uwzględniane tylko wtedy, gdy są one aktywne przy wywołaniu `CALCPOSI()`. Na status mogą mieć wpływ:

- Dane maszynowe: MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- Dane ustawcze: $AC_WORKAREA_CS_...
- Sygnał interfejsu NC/PLC: DB31, ... DBX12.2 / 3
- Polecenia: `WALIMON` / `WALIMOF`

Programowy wyłącznik krańcowy i transformacje

W przypadku `CALCPOSI()` z powodu niejednoznaczności w określonych miejscach zakresu ruchu w różnych transformacjach kinematycznych (np. `TRANSMIT`), pozycje osi maszyny

(MKS) nie zawsze można jednoznacznie określić na podstawie pozycji osi geometrycznych (WKS). Przy normalnym wykonywaniu ruchu postępowego jednoznaczność wynika z reguły z wcześniejszej historii i warunku, że ciągłemu ruchowi w WKS musi odpowiadać ciągły ruch osi maszyny MKS. Przy nadzorze ograniczeń programowych przy pomocy funkcji `CALCPOSI()` w celu wyeliminowania niejednoznaczności wykorzystywana jest aktualna pozycja maszyny.

Uwaga

Zatrzymanie przebiegu wyprzedzającego

W przypadku użycia `CALCPOSI()` w połączeniu z transformacjami, wyłączna odpowiedzialność za zaprogramowanie zatrzymania przebiegu wyprzedzającego (`STOPRE`) w celu synchronizacji pozycji maszyny z przebiegiem wyprzedzającym przed `CALCPOSI()` leży po stronie użytkownika.

Odstęp obszaru ochrony i konwencjonalne obszary ochrony

W przypadku konwencjonalnych obszarów ochrony **nie** można zagwarantować, że odstęp bezpieczeństwa podany w parametrze `<Limit>[3]` zostanie zachowany względem wszystkich obszarów ochrony przy ruchu postępowym na zaprogramowanej drodze ruchu. Zagwarantowane jest tylko to, że jeżeli dla zwróconego punktu końcowego określonego w `<Dist>` przy przedłużeniu o odstęp bezpieczeństwa w kierunku ruchu, nie zostanie naruszony żaden obszar ochrony. Prosta może jednak przechodzić dowolnie blisko obszaru ochrony.

Odległość obszaru ochrony i obszaru ochrony dla unikania kolizji

W przypadku obszarów ochrony unikanie kolizji zapewnia, że przy ruchu postępowym na zaprogramowanej drodze ruchu odstęp bezpieczeństwa określony w parametrze `<Limit>[3]` zostanie zachowany względem wszystkich obszarów ochrony.

Odstęp bezpieczeństwa określony w parametrze `<Limit>[3]` jest skuteczny tylko wtedy, gdy obowiązuje:

`<Limit>[3]` **>** (MD10619 $MN_COLLISION_TOLERANCE)

Jeżeli w parametrze `<TestLim>` ustawiono Bit 4 (obliczenie kolejnych kierunków przejazdu), to zawarty w `<DIST>` wektor kierunku jest ważny tylko wtedy, gdy miejsce tysięcy ustawione jest w wartości zwrotnej funkcji (`<Status>`). Taki kierunek nie może zostać określony, ponieważ albo obszar ochrony został naruszony albo aktywna jest transformacja, wartość wejściowa w `<DIST>` pozostaje niezmieniona. Inne komunikaty błędów nie występują.

Specjalne polecenia wykonania ruchu 5

5.1 Ruch do pozycji kodowanych (CAC, CIC, CDC, CACP, CACN)

Przez poniższe polecenia można wykonywać ruchy w osiach liniowych i obrotowych do stałych pozycji w osiach, zapisanych w tablicach danych maszynowych. Ten rodzaj programowania jest określany jako "ruchy do pozycji kodowanych".

Składnia

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

Znaczenie

`CAC(<n>)`:	Ruch do pozycji kodowanej o numerze n
`CIC(<n>)`:	Ruch do pozycji kodowanej, wychodząc od aktualnego numeru pozycji, o n miejsc do przodu (+n) albo do tyłu (-n)
`CDC(<n>)`:	Ruch do pozycji kodowanej o numerze n po najkrótszej drodze (tylko dla osi obrotowych)
`CACP(<n>)`:	Ruch do pozycji kodowanej o numerze n w kierunku dodatnim (tylko dla osi obrotowych)
`CACN(<n>)`:	Dosunięcie do pozycji kodowanej o numerze n w kierunku ujemnym (tylko dla osi obrotowych)
`<n>`:	Numer pozycji w ramach tablicy danych maszynowych Zakres wartości: 0, 1, ... (max liczba miejsc w tablicy - 1)

Przykład: Ruch do kodowanych pozycji osi pozycjonowania

Kod programu	**Komentarz**
`N10 FA[B]=300`	`; Posuw dla osi pozycjonowania B`
`N20 POS[B]=CAC(10)`	`; Ruch do pozycji kodowanej o numerze 10`
`N30 POS[B]=CIC(-4)`	`; Ruch do pozycji kodowanej od "aktualnego numeru pozycji" - 4`

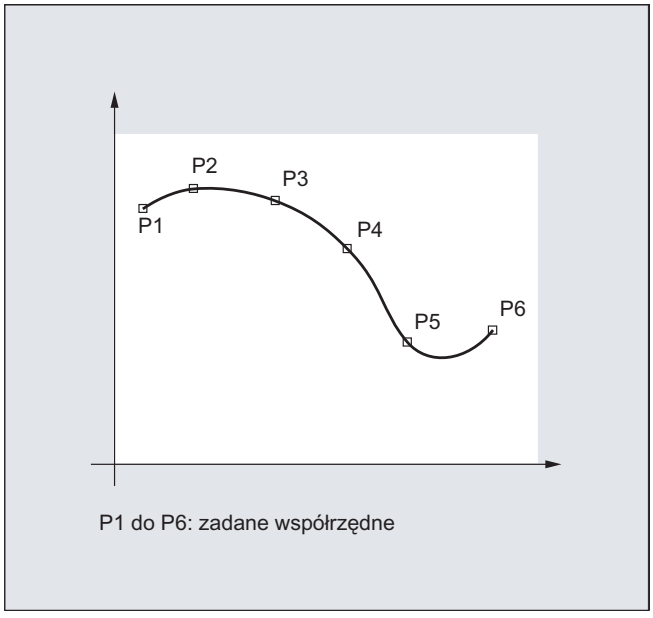
Literatura

- Podręcznik działania, Funkcje rozszerzające; Osie podziałowe (T1)
- Podręcznik działania Akcje synchroniczne

5.2 Interpolacja Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Dowolnie zakrzywione kontury na obrabianych przedmiotach nie mogą zostać dokładnie opisane analitycznie. Tego rodzaju kontury są dlatego przybliżane przez ograniczoną liczbę punktów oparcia, np. przy digitalizacji powierzchni. W celu utworzenia powierzchni dygitalizowanej na obrabianym przedmiocie punkty oparcia muszą zostać połączone w opis konturu. Umożliwia to interpolacja Spline.

Spline definiuje krzywą, która składa się z wielomianów 2. albo 3. stopnia. Właściwości w punktach oparcia spline są definiowane **zależnie od zastosowanego typu spline**.



W przypadku SINUMERIK solution line są do dyspozycji następujące typy spline:

- A-Spline
- B-Spline
- C-Spline

Składnia

Informacje ogólne:

```
ASPLINE X... Y... Z... A... B... C...
ASPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

Dodatkowo programowane w przypadku B-Spline:

```
PW=<n>
SD=2
PL=<wartość>
```

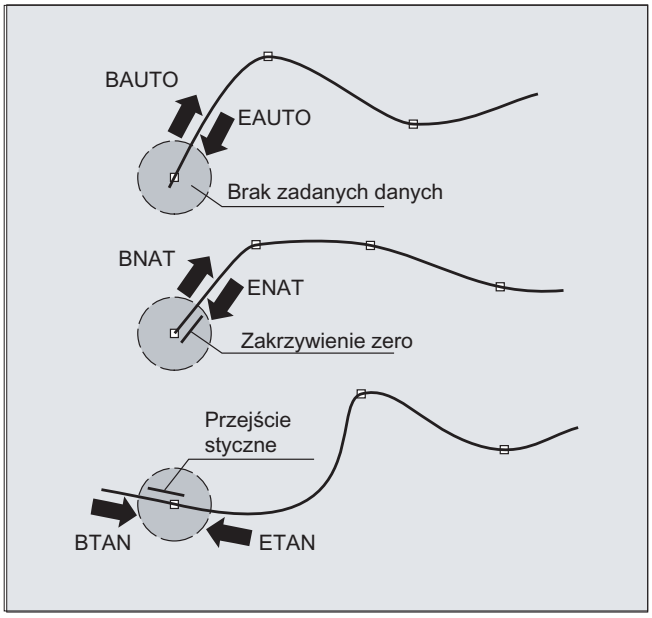
Dodatkowo programowane w przypadku A-Spline i C-Spline:

```
BAUTO / BNAT / BTAN
```

EAUTO / ENAT / ETAN

Znaczenie

Typ interpolacji spline:			
ASPLINE:	Polecenie do włączenia interpolacji A-Spline		
BSPLINE:	Polecenie do włączenia interpolacji B-Spline		
CSPLINE:	Polecenie do włączenia interpolacji C-Spline		
	Polecenia ASPLINE, BSPLINE i CSPLINE działają modalnie i należą do grupy poleceń drogowych.		
Punkty oparcia wzgl. punkty kontrolne:			
X... Y... Z... A... B... C...	Pozycje we współrzędnych kartezjańskich		
Waga punktu (tylko B-Spline):			
PW:	Przy pomocy polecenia PW jest dla każdego punktu oparcia możliwe zaprogramowanie tak zwanej "wagi punktu".		
<n>:	"Waga punktu"		
	Zakres wartości:	0 ≤ n ≤ 3	
	Długość kroku:	0.0001	
	Działanie:	n > 1	Krzywa jest silniej przyciągana przez punkt kontrolny.
		n < 1	Krzywa jest słabiej przyciągana przez punkt kontrolny.
Stopień spline (tylko B-Spline):			
SD:	Standardowo jest stosowany wielokąt 3. stopnia. Przez zaprogramowanie SD=2 można jednak użyć również wielokąt 2. stopnia.		
Odstęp węzłów (tylko B-Spline):			
PL:	Odstępy węzłów są wewnętrznie odpowiednio obliczane. Sterowanie może jednak realizować również zadane odstępy węzłów, które przy pomocy polecenia PL są podawane jako tzw. długość przedziału między parametrami.		
<wartość>:	Parametr - przedział - długość		
	Zakres wartości:	jak wymiar drogi	
Zachowanie się na przejściu na początku krzywej spline (tylko A-Spline lub C-Spline):			
BAUTO:	Brak zadania zachowania się na przejściu. Początek wynika z położenia pierwszego punktu.		
BNAT:	Zakrzywienie zero		
BTAN:	Przejście styczne do poprzedniego bloku (położenie kasowania)		
Zachowanie się na przejściu na końcu krzywej spline (tylko A-Spline lub C-Spline):			
EAUTO:	Brak zadania zachowania się na przejściu. Koniec wynika z położenia ostatniego punktu.		
ENAT:	Zakrzywienie zero		

ETAN:	Przejście styczne do poprzedniego bloku (położenie kasowania)
	

Uwaga

Programowane zachowanie się na przejściu nie ma wpływu na B-Spline. B-Spline jest w punkcie startowym i końcowym zawsze styczny do wielokąta kontrolnego.

Warunki brzegowe

- Korekcję promienia narzędzia można stosować.
- Nadzór na kolizję następuje w rzucie na płaszczyznę.

Przykłady

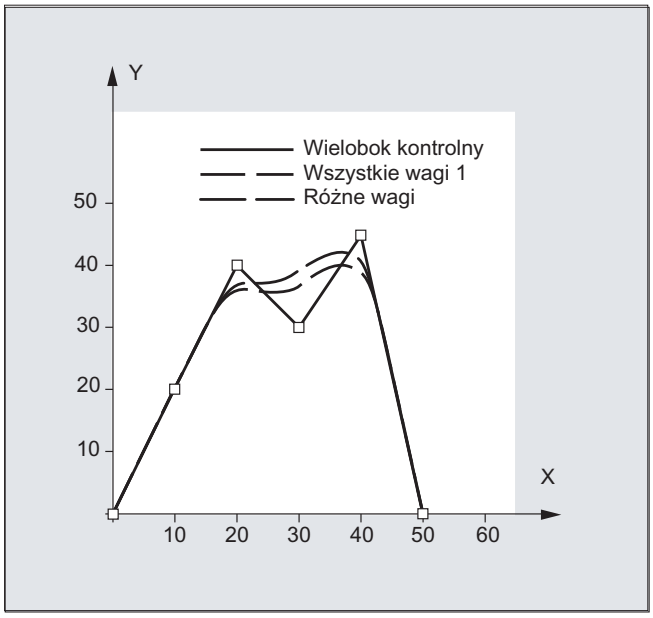
Przykład 1: B-Spline

```
Kod programu 1 (wszystkie wagi 1)
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

Kod programu 2 (różne wagi)

```
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
```

Kod programu 3 (wielokąt kontrolny)	**Komentarz**
N10 G1 X0 Y0 F300 G64	
N20	; Odpada
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	

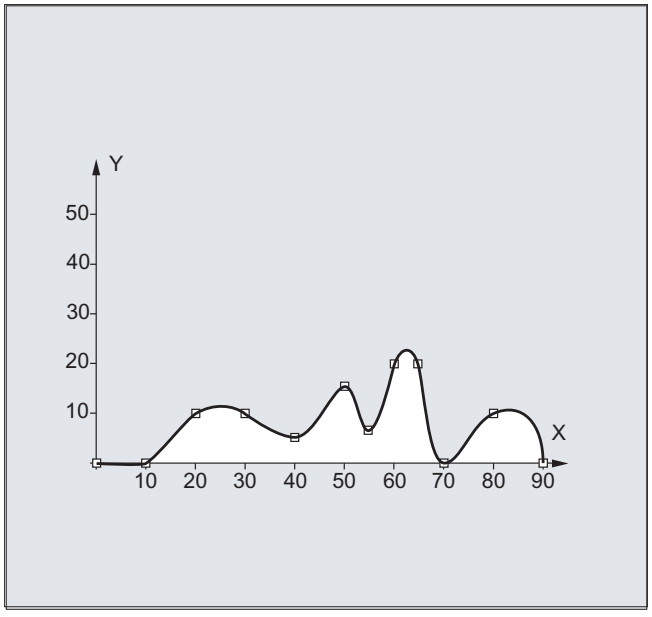


Przykład 2: C-Spline, na początku i na końcu zakrzywienie zero

Kod programu

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
```

```
Kod programu
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
```



Przykład 3: Interpolacja spline (A-Spline) i transformacja współrzędnych (ROT)

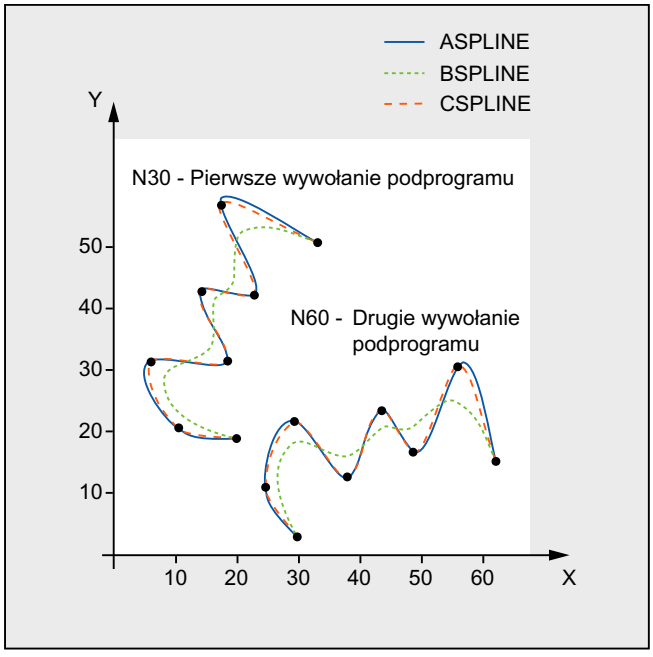
Program główny:

Kod programu	Komentarz
N10 G00 X20 Y18 F300 G64	; Ruch do punktu startowego.
N20 ASPLINE	; Uaktywnienie typu interpolacji A-Spline.
N30 KONTUR	; Pierwsze wywołanie podprogramu.
N40 ROT Z-45	; Transformacja współrzędnych: Obrót WKS o -45° wokół osi Z.
N50 G00 X20 Y18	; Ruch do punktu początkowego konturu.
N60 KONTUR	; Drugie wywołanie podprogramu.
N70 M30	; Koniec programu

Podprogram "Kontur" (zawiera współrzędne punktu oparcia):

```
Kod programu
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
```

Na poniższym rysunku są oprócz krzywej spline, która wynika z przykładu programu (ASPLINE), zawarte również krzywe spline, które wynikłyby przy uaktywnieniu interpolacji B-Spline albo C-Spline



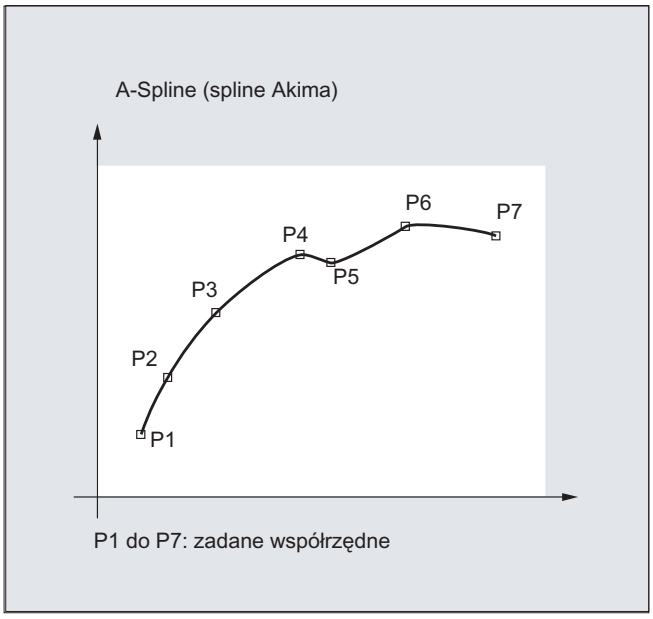
Dodatkowe informacje

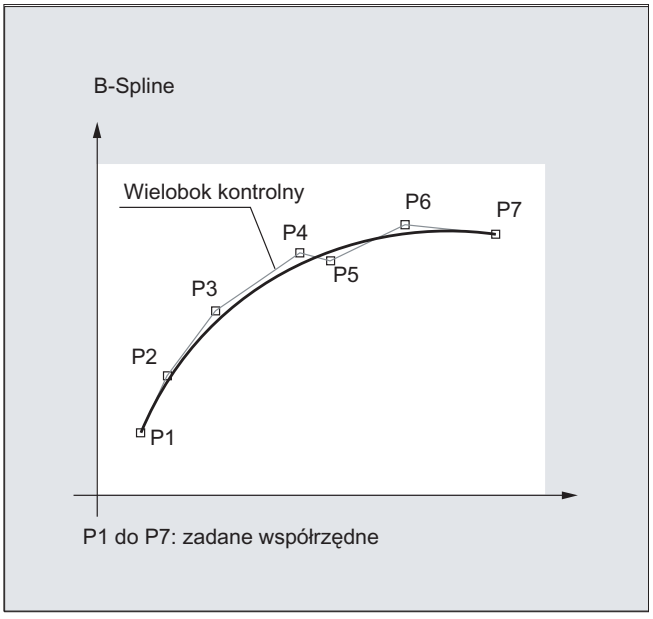
Zalety interpolacji Spline

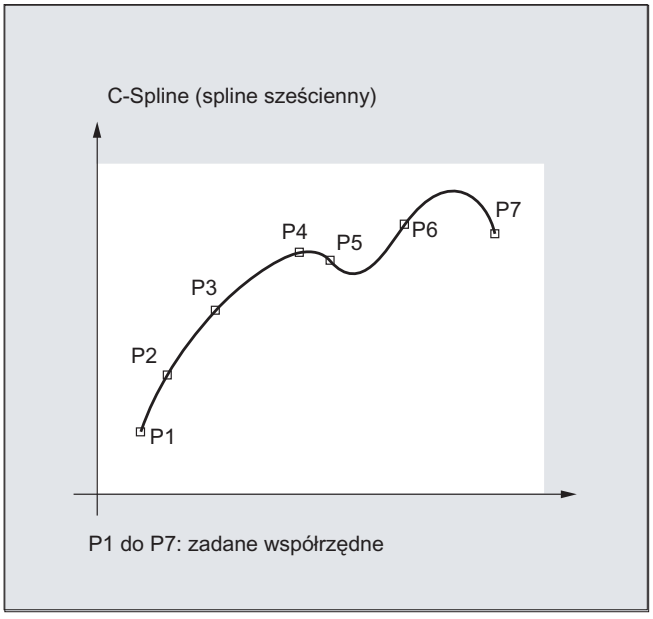
Poprzez zastosowanie interpolacji Spline można, w przeciwieństwie do zastosowania bloków prostoliniowych `G01`, osiągnąć następujące zalety:

- Zmniejszenie liczby potrzebnych bloków programu obróbki do opisania konturu
- Miękki, oszczędzający mechanikę, przebieg krzywej przy przejściach między blokami programu obróbki

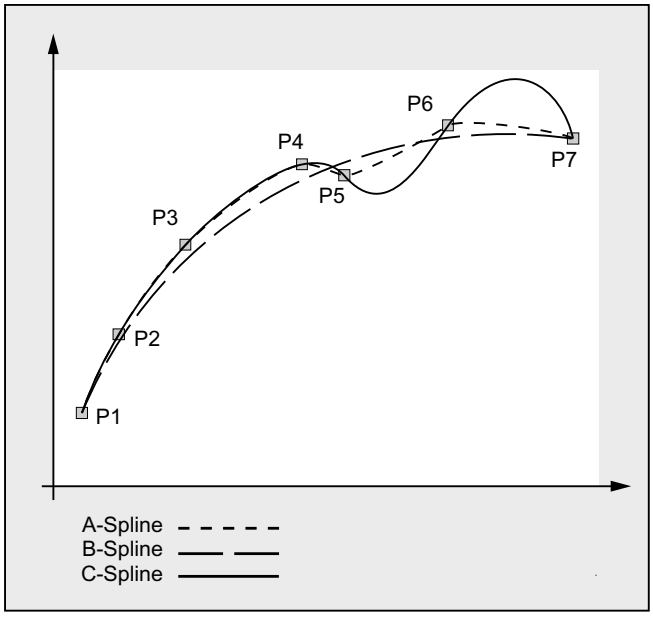
Właściwości i zastosowanie różnych typów Spline

Typ Spline	Właściwości i zastosowanie
A-Spline	 **Właściwości:** • Przebiega ściśle przez zadane punkty oparcia. • Przebieg krzywej ma ciągłą pochodną, ale nie stałe zakrzywienie. • Nie wytwarza niepotrzebnych drgań. • Obszar wpływu zmian punktów oparcia jest lokalny, tzn. zmiana punktu oparcia ma wpływ na max 6 sąsiednich punktów oparcia. **Zastosowanie:** A-Spline jest przede wszystkim odpowiednia do interpolacji przebiegów krzywych z dużymi zmianami nachylenia (np. krzywe schodkowe).

Typ Spline	Właściwości i zastosowanie
B-Spline	 **Właściwości:** • Przebiega nie przez zadane punkty oparcia lecz tylko w ich pobliżu. Krzywa jest przyciągana przez punkty oparcia. Poprzez współczynniki wagi dla punktów oparcia można dodatkowo wpływać na przebieg krzywej. • Przebieg krzywej ma stałą pochodną i stałe zakrzywienie. • Nie wytwarza niepotrzebnych drgań. • Obszar wpływu zmian punktów oparcia jest lokalny, tzn. zmiana punktu oparcia ma wpływ na max 6 sąsiednich punktów oparcia. **Zastosowanie:** B-Spline stanowi pierwotnie interfejs do systemów CAD.

Typ Spline	Właściwości i zastosowanie
C-Spline	 **Właściwości:** • Przebiega ściśle przez zadane punkty oparcia. • Przebieg krzywej ma stałą pochodną i stałe zakrzywienie. • Wytwarza często niepotrzebne drgania, w szczególności w miejscach o dużych zmianach nachylenia. • Zakres wpływu zmian punktów oparcia jest globalny, tzn. zmiana jednego punktu oparcia ma wpływ na cały przebieg krzywej. **Zastosowanie:** C-Spline można tylko wtedy dobrze zastosować, gdy punkty oparcia leżą na analitycznie znanej krzywej (okrąg, parabola, hiperbola)

Porównanie trzech typów Spline przy takich samych punktach oparcia



Liczba minimalna bloków Spline

Polecenie G `ASPLINE`, `BSPLINE` i `CSPLINE` łączą punkty końcowe bloków przy pomocy Spline. W tym celu musi w przebiegu wyprzedzającym być równocześnie obliczanych szereg bloków (punktów końcowych). Wielkość bufora do obliczania wynosi standardowo 10 bloków. Nie każda informacja zawarta w bloku jest punktem końcowym Spline. Sterowanie potrzebuje jednak z 10 bloków określonej liczby bloków z punktem końcowym Spline.

Typ Spline	Liczba minimalna bloków Spline
A-Spline:	Z każdych 10 bloków co najmniej **4** muszą być blokami Spline. Nie wlicza się do tego bloków komentarzy i obliczeń parametrów.
B-Spline:	Z każdych 10 bloków co najmniej **6** muszą być blokami Spline. Nie wlicza się do tego bloków komentarzy i obliczeń parametrów.
C-Spline:	Potrzebna minimalna liczba bloków Spline wynika z następującej sumy: Wartość z MD20160 $MC_CUBIC_SPLINE_BLOCKS + 1 W MD20160 jest wpisywana liczba punktów, z których jest obliczany segment Spline. Ustawienie standardowe wynosi 8. Z 10 bloków musi dlatego w przypadku standardowym być **9** bloków Spline.

Uwaga

Przy zejściu poniżej tolerowalnej wartości jest wyprowadzany alarm, tak samo gdy oś uczestnicząca w Spline zostanie zaprogramowana jako oś pozycjonowania.

Połączenie krótkich bloków Spline

W przypadku interpolacji Spline mogą powstawać krótkie bloki Spline, które prowadzą do niepotrzebnego zmniejszenia prędkości ruchu po torze. Za pomocą funkcji „Połączenie krótkich bloków Spline" bloki te mogą zostać tak połączone, by wynikowa długość bloku była wystarczająco duża i nie prowadziła do zmniejszenia prędkości ruchu po torze.

Funkcja jest uaktywniana poprzez specyficzną dla kanału daną maszynową:

MD20488 $MC_SPLINE_MODE (ustawienie dla interpolacji Spline)

Literatura:
Podręcznik działania Funkcje podstawowe; Tryb przechodzenia płynnego, Zatrzymanie dokładne, LookAhead (B1),
punkt: Połączenie krótkich bloków Spline

5.3 Zespół spline (SPLINEPATH)

Osie, które mają interpolować w zespole spline, są wybierane poleceniem `SPLINEPATH`. Przy interpolacji spline jest możliwych do ośmiu osi uczestniczących w tworzeniu konturu.

Uwaga

Jeżeli `SPLINEPATH` nie zostanie explicite zaprogramowane, wówczas jako zespół spline wykonują ruch pierwsze trzy osie kanału.

Składnia

Ustalenie zespołu spline następuje w oddzielnym bloku:

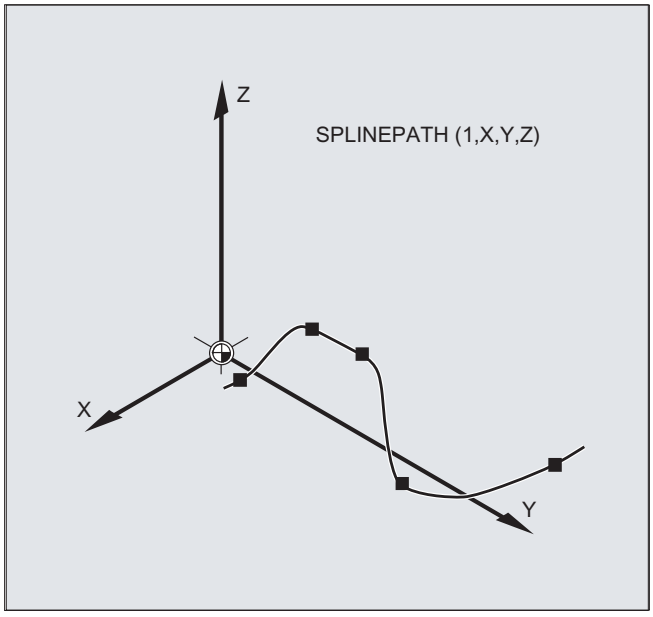
```
SPLINEPATH(n,X,Y,Z,…)
```

Znaczenie

`SPLINEPATH`:	Polecenie do ustalenia zespołu spline
`n`:	=1 (stała wartość)
`X,Y,Z,…` :	Identyfikatory osi uczestniczących w tworzeniu konturu, które mają interpolować w zespole spline

Przykład: Zespół spline z trzema osiami uczestniczącymi w tworzeniu konturu

Kod programu	**Komentarz**
`N10 G1 X10 Y20 Z30 A40 B50 F350`	
`N11 SPLINEPATH(1,X,Y,Z)`	`; Zespół spline`
`N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60`	`; C-Spline`
`N14 X30 Y40 Z50 A60 B70`	`; Punkty oparcia`
`…`	
`N100 G1 X… Y…`	`; Cofnięcie wyboru interpolacji spline`


Z
SPLINEPATH (1,X,Y,Z)
X
Y

5.4 Włączenie/Wyłączenie kompresji bloków NC (COMPON / COMPCURV / COMPCAD / COMPSURF)

Funkcje do kompresji w blokach liniowych (i w zależności od parametryzacji również w blokach okręgu i/lub ruchu szybkiego) włącza się i wyłącza za pomocą poleceń G z grupy G nr 30. Polecenia działają modalnie.

Składnia

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

Znaczenie

COMPON:	Włączenie funkcji kompresora COMPON
COMPCURV:	Włączenie funkcji kompresora COMPCURV
COMPCAD:	Włączenie funkcji kompresora COMPCAD
COMPSURF	Włączenie funkcji kompresora COMPSURF
COMPOF:	Wyłączenie aktualnie aktywnej funkcji kompresora.

Uwaga

W celu dodatkowego polepszenia jakości powierzchni można użyć funkcji wygładzania G642 i ograniczenia przyspieszenia drugiego stopnia SOFT. Te polecenia należy pisać na początku programu.

Przykład: COMPCAD

```
Kod programu                    Komentarz
N10 G00 X30 Y6 Z40
N20 G1 F10000 G642              ; Włączenie: Funkcja wygładzania G642
N30 SOFT                        ; Włączenie: Ograniczenie przyspieszenia drugiego
                                stopnia SOFT
N40 COMPCAD                     ; Włączenie: Funkcja kompresora COMPCAD
N50 STOPFIFO
N24050 Z32.499                  ; 1. blok ruchu postępowego
N24051 X41.365 Z32.500          ; 2. blok ruchu postępowego
...
N99999 X... Z...                ; Ostatni blok ruchu postępowego
COMPOF                          ; Funkcja kompresora wył.
...
```

5.5 Interpolacja wielomianowa (POLY, POLYPATH, PO, PL)

W przypadku interpolacji wielomianowej (POLY) chodzi właściwie nie o rodzaj interpolacji spline. Jest ona w pierwszym rzędzie pomyślana jako interfejs do programowania zewnętrznie wytwarzanych krzywych spline. Przy tym odcinki spline mogą być programowane bezpośrednio.

Ten rodzaj interpolacji odciąża NC od obliczania współczynników wielomianu. Daje się ona optymalnie stosować wtedy, gdy współczynniki przychodzą bezpośrednio od systemu CAD albo postprocesora.

Składnia

Wielomian 3. stopnia:

```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

Wielomian 5. stopnia i nowa składnia wielomianu:

```
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")
```

Uwaga

Suma zaprogramowanych w jednym bloku współczynników wielomianu i osi nie może przekraczać maksymalnej dozwolonej w bloku liczby osi.

Znaczenie

POLY :	Włączenie interpolacji wielomianowej przy pomocy bloku z POLY.
POLYPATH :	Interpolacja wielomianowa wybierana dla obydwu grup osi AXIS lub VECT
PO[identyfikator osi/zmienna] :	Punkty końcowe i współczynniki wielomianu
X, Y, Z :	Identyfikatory osi
xe, ye, ze :	Podanie pozycji końcowej dla każdorazowej osi; zakres wartości jak miara drogi
a2, a3, a4, a5 :	Współczynniki a_2, a_3, a_4, i a_5 są pisane z ich wartościami; zakres wartości jak miara drogi. Każdorazowo ostatni współczynnik można pominąć, gdy ma on wartość zero.
PL :	Długość przedziału parametru, na którym wielomiany są zdefiniowane (zakres definicji funkcji f(p)). Przedział rozpoczyna się zawsze od 0, p może przyjmować wartości od 0 do PL. Teoretyczny zakres wartości dla PL: 0,0001 … 99 999,9999 **Wskazówka:** Wartość PL obowiązuje dla bloku, w którym się znajduje. Gdy PL nie zaprogramowano, działa PL=1.

Włączenie/wyłączenie interpolacji wielomianowej

Interpolacja wielomianowa jest w programie obróbki włączana poleceniem G `POLY`.

Polecenie G `POLY` należy razem z `G0`, `G1`, `G2`, `G3`, `ASPLINE`, `BSPLINE` i `CSPLINE` do 1. grupy G.

Osie, które są zaprogramowane przez podanie tylko nazwy i punktu końcowego (np. `X10`), wykonują ruch liniowy. Gdy wszystkie osie bloku NC są tak zaprogramowane, sterowanie zachowuje się tak, jak w przypadku G1.

Interpolacja wielomianowa jest przez zaprogramowanie innego polecenia z 1. grupy G (np. `G0`, `G1`) implicite ponownie wyłączana.

Współczynnik wielomianu

Wartość `PO (PO[])` lub `...=PO(...)` podaje wszystkie współczynniki wielomianu dla osi. Odpowiednio do stopnia wielomianu podaje się wiele wartości rozdzielonych przecinkiem. W ramach jednego bloku są możliwe różne stopnie wielomianów dla różnych osi.

Wywołanie podprogramu POLYPATH

Przy pomocy `POLYPATH(...)` może zostać udostępniona interpolacja wielomianowa selektywnie dla określonych grup osi:

Tylko osie uczestniczące w tworzeniu konturu i osie dodatkowe:	`POLYPATH("AXES")`
Tylko osie orientacji: (przy wykonywaniu ruchu z transformacją orientacji)	`POLYPATH("VECT")`

Każdorazowo nie udostępnione osie wykonują ruch liniowy.

Standardowo jest udostępniona interpolacja wielomianowa dla obydwu grup osi.

Przez zaprogramowanie bez parametru `POLYPATH( )` jest wyłączana interpolacja wielomianowa dla wszystkich osi.

Przykład

```
Kod programu                                          Komentarz
N10 G1 X… Y… Z… F600
N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2)          ; Interpolacja wielomianowa wł.
PL=1.5
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                       ; Dane mieszane dla osi
N27 PO[X]=(10,2.5) PO[Y]=(2.3)                        ; PL nie zaprogramowano; działa
                                                      PL=1
N30 G1 X… Y… Z.                                       ; Interpolacja wielomianowa wył.
…
```

Przykład: Nowa składnia wielomianu

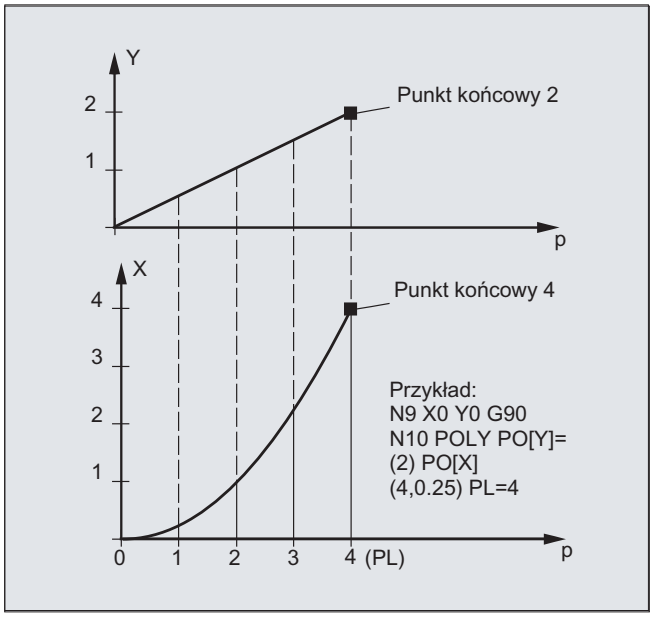
Dalej obowiązująca składnia wielomianowa	Nowa składnia wielomianu
`PO[identyfikator osi]=(.. , ..)`	`Identyfikator osi=PO(.. , ..)`
`PO[PHI]=(.. , ..)`	`PHI=PO(.. , ..)`
`PO[PSI]=(.. , ..)`	`PSI=PO(.. , ..)`
`PO[THT]=(.. , ..)`	`THT=PO(.. , ..)`
`PO[]=(.. , ..)`	`PO(.. , ..)`
`PO[zmienna]=IC(.. , ..)`	`zmienna=PO IC(.. , ..)`

Przykład: Krzywa na płaszczyźnie X/Y

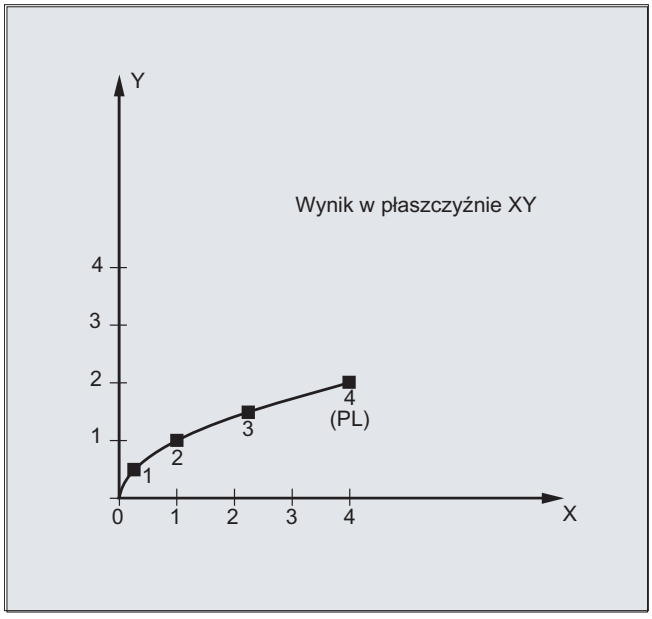
Programowanie

Kod programu
`N9 X0 Y0 G90 F100`
`N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4`

Przebieg krzywych X(p) i Y(p)



Przebieg krzywej w płaszczyźnie XY



Opis

Ogólna postać funkcji wielomianowej brzmi:

$f(p)= a_0 + a_1p + a_2p^2 + . . . + a_np^n$

gdzie: a_i: stałe współczynniki (i = 0, 1, ..., n)
p: Parametry

W sterowaniu można programować maksymalnie wielomiany 5. stopnia:

$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$

Przez wyposażanie współczynników w konkretne wartości można tworzyć różne przebiegi krzywych, jak funkcje prostoliniowe, paraboliczne, potęgowe.

Prosta jest tworzona przez $a_2 = a_3 = a_4 = a_5 = 0$:

$f(p)= a_0 + a_1p$

Dalej obowiązuje:

$a_{0:}$ Pozycja osi na końcu bloku poprzedzającego

p = PL

$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3) / p$

Jest możliwe programowanie wielomianów, **bez** uaktywnienia interpolacji wielomianowej poleceniem G `POLY`. W tym przypadku nie są interpolowane zaprogramowane wielomiany, lecz następuje ruch liniowy do zaprogramowanych punktów końcowych osi (`G1`). Dopiero po bezpośrednim uaktywnieniu interpolacji wielomianowej w programie obróbki (`POLY`) ruch dla zaprogramowanych wielomianów jest wykonywany, jak dla wielomianów.

Cecha szczególna: Wielomian mianownikowy

Dla osi geometrycznych można przy pomocy `PO[]=(…)` bez podania nazwy osi zaprogramować również wspólny wielomian mianownikowy, tzn. ruch w osiach geometrycznych jest wykonywany jako iloraz dwóch wielomianów.

W ten sposób dają się dokładnie przedstawiać np. krzywe stożkowe (okrąg, elipsa, parabola, hiperbola).

Przykład:

Kod programu	**Komentarz**
`POLY G90 X10 Y0 F100`	`; Osie geometryczne wykonują ruch liniowy do pozycji X10 Y0.`
`PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)`	`; Osie geometryczne wykonują ruch po ćwierćokręgu do X0 Y10.`

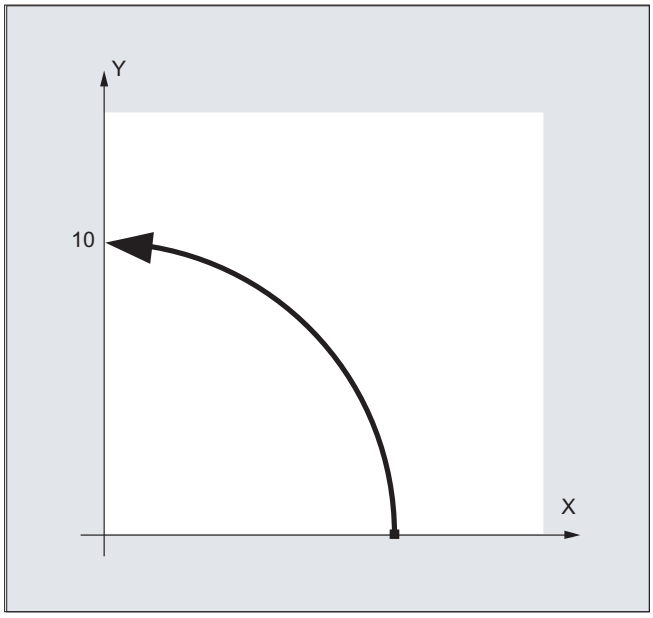
Stały współczynnik (a_0) wielomianu mianownikowego jest stale przyjmowany o wielkości 1. Zaprogramowany punkt końcowy jest niezależny od `G90` / `G91`.

Z zaprogramowanych wartości następuje obliczenie X(p) i Y(p):

$X(p) = (10 - 10 * p^2) / (1 + p^2)$
$Y(p) = 20 * p / (1 + p^2)$
gdzie $0 \le p \le 1$

Z zaprogramowanych punktów początkowych, punktów końcowych, współczynnika a_2 i PL=1 wynikają następujące wartości pośrednie:

Licznik (X) = $10 + 0 * p - 10 * p^2$
Licznik (Y) = $0 + 20 * p + 0 * p^2$
Mianownik = $1 + p^2$



Przy włączonej interpolacji wielomianowej zaprogramowanie wielomianu mianownikowego z miejscami zerowymi w ramach przedziału `[0,PL]` jest odrzucane z alarmem. Na ruch osi dodatkowych wielomian mianownikowy nie ma wpływu.

Uwaga

Korekcja promienia narzędzia daje się przy interpolacji wielomianowej włączyć przy pomocy `G41`, `G42` i stosować jak dla interpolacji prostoliniowej albo kołowej.

5.6 Ustawiane odniesienie do toru (SPATH, UPATH)

W przypadku interpolacji wielomianowej (POLY, ASPLINE, BSPLINE, CSPLINE, COMPON, COMPCURV) pozycje osi uczestniczących w tworzeniu konturu i są zadawane przez wielomiany $p_i(U)$. Parametr krzywej U porusza się przy tym w ramach bloku NC od 0 do 1.

Przez FGROUP są wybierane osie (osie FGROUP), do których odnosi się posuw po torze F. Interpolacja ze stałą prędkością na drodze S osi FGROUP oznacza podczas interpolacji wielomianowej jednak z reguły nie stałą zmianę parametru krzywej U. Dla osi nie zawartych w FGROUP można dlatego wybierać między dwoma możliwościami, jak powinny one podążać za osiami FGROUP:

- Synchronicznie do drogi S (SPATH)
- Synchronicznie do parametru U krzywej (UPATH)

Składnia

```
SPATH
UPATH
```

Znaczenie

SPATH:	Osie nie zawarte w FGROUP wykonują ruch w odniesieniu do drogi S ruchu po torze
UPATH:	Osie nie zawarte w FGROUP wykonują ruch w odniesieniu do parametru U krzywej

Uwaga

UPATH i SPATH określają również współzależność wielomianu słowa F (FPOLY, FCUB, FLIN) z ruchem po torze.

Warunki brzegowe

SPATH wzgl. UPATH nie mają znaczenia przy:

- Interpolacja liniowa (G1)
- Interpolacja kołowa: (G2, G3)
- Bloki gwintowana (G33, G34, G35, G33x, G63)
- Gdy wszystkie osie uczestniczące w tworzeniu konturu są zawarte w FGROUP.

Przykład

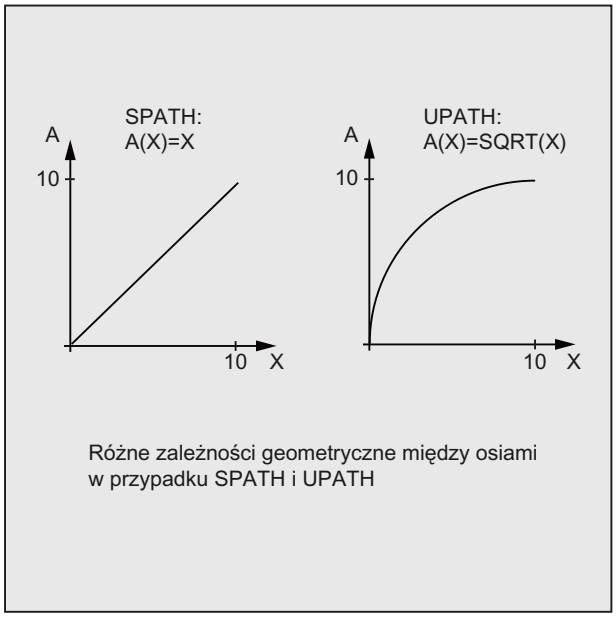
Poniższy przykład pokazuje różnicę między obydwoma rodzajami prowadzenia ruchu.

Kod programu

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
N20 POLY PO[X]=(10,10) A10
```

```
Kod programu
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

W obydwu segmentach programu w `N20` droga S osi FGROUP zależy od kwadratu parametru krzywej U. Dlatego wzdłuż drogi X wynikają różne pozycje osi synchronicznej A, w zależności od tego, czy jest aktywne `SPATH` czy `UPATH`.



Różne zależności geometryczne między osiami w przypadku SPATH i UPATH

Dodatkowe informacje

Zachowanie się sterowania w przypadku resetu i danych maszynowych / danych opcji

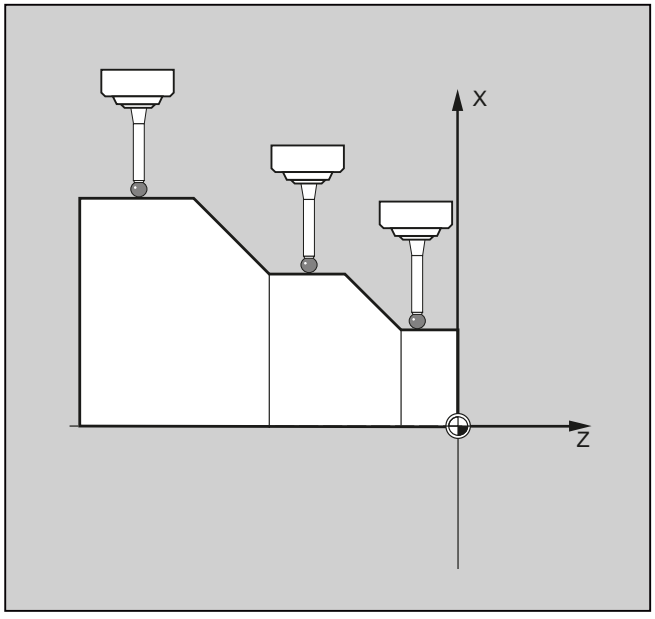
Po resecie działa polecenie G określone przez MD20150 $MC_GCODE_RESET_VALUES[44] (45. grupa kodów G).

Wartość ustawienia podstawowego dla rodzaju ścięcia jest ustalana za pomocą MD20150 $MC_GCODE_RESET_VALUES[9] (10. grupa kodów G).

Specyficzna dla osi dana maszynowa MD33100 $MA_COMPRESS_POS_TOL[<n>] ma rozszerzone znaczenie: zawiera tolerancje dla funkcji kompresora i dla ścinania narożników przy pomocy G642.

5.7 Pomiar przy pomocy sondy przełączającej, MEAS, MEAW

Przy pomocy funkcji "pomiar sondą przełączającą" są wykonywane ruchy do pozycji rzeczywistych na obrabianym przedmiocie i przy zboczu przełączenia sondy są mierzone pozycje wszystkich osi zaprogramowanych w bloku pomiarowym i dla każdej osi zapisywane w odpowiedniej komórce pamięci.



Dla programowania funkcji są do dyspozycji obydwa następujące adresy stałe:

- MEAS
  Przy pomocy MEAS jest kasowana pozostała droga między pozycją rzeczywistą i zadaną.
- MEAW
  MEAW jest stosowane do zadań pomiarowych, przy których w każdym przypadku powinno nastąpić dosunięcie do zaprogramowanej pozycji.

MEAS i MEAW działają pojedynczymi blokami i są programowane razem z instrukcjami ruchu. Posuw i rodzaj interpolacji (G0, G1, …), tak samo jak liczba osi, muszą przy tym zostać dopasowane do zadania pomiarowego.

Składnia

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Znaczenie

MEAS:	Polecenie: Pomiar **z** kasowaniem pozostałej drogi	
	Działanie:	Pojedynczymi blokami
MEAW:	Polecenie: Pomiar **bez** kasowania pozostałej drogi	
	Działanie:	pojedynczymi blokami

<TE>:	Zdarzenie przerzutnikowe do wyzwolenia pomiaru		
	Typ:	INT	
	Zakres wartości:	-2, -1, 1, 2	
	Znaczenie:		
	(+)1	Zbocze rosnące od sondy pomiarowej 1 (na wejście pomiarowe 1)	
	-1	Zbocze opadające od sondy pomiarowej 1 (na wejście pomiarowe 1)	
	(+)2	Zbocze rosnące od sondy pomiarowej 2 (na wejście pomiarowe 2)	
	-2	Zbocze opadające od sondy pomiarowej 2 (na wejście pomiarowe 2)	
	Wskazówka Istnieją maksymalnie 2 sondy pomiarowe (zależnie od stopnia rozbudowy).		
G...:	Rodzaj interpolacji, np. G0, G1, G2 lub G3		
X... Y... Z...:	Punkty końcowe we współrzędnych kartezjańskich		

Przykład

Kod programu	**Komentarz**
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; Blok pomiarowy z sonda pomiarową pierwszego wejścia pomiarowego i interpolacją prostoliniową. Zatrzymanie przebiegu wyprzedzającego jest wytwarzane automatycznie.
...	

Dodatkowe informacje

Status zlecenia pomiarowego

Jeżeli w programie jest wymagana ocena, czy sonda pomiarowa przełączyła czy nie, można odczytać zmienną stanu $AC_MEA\[<n>] (<n> = numer sondy pomiarowej):

Wartość	Znaczenie
0	Zlecenie pomiarowe nie jest spełnione
1	Zlecenie pomiarowe zakończone pomyślnie (sonda pomiarowa przełączyła)

Uwaga

Jeżeli sonda pomiarowa jest wychylana w programie, zmienna jest ustawiana na 1. Przy starcie bloku pomiarowego zmienna jest automatycznie ustawiana na stan początkowy.

Rejestracja zmierzonej wartości

Są rejestrowane pozycje wszystkich osi uczestniczących w tworzeniu konturu i osi pozycjonowania (maksymalna liczba osi zależnie od konfiguracji sterowania). W przypadku `MEAS` ruch jest po przełączeniu sondy w sposób zdefiniowany hamowany.

Uwaga

Gdy w bloku pomiarowym jest zaprogramowana oś geometryczna, zostaną zapisane wartości pomiarowe dla wszystkich aktualnych osi geometrycznych.

Jeżeli w bloku pomiaru jest zaprogramowana oś uczestnicząca w transformacji, zostaną zapisane wartości pomiarowe wszystkich osi uczestniczących w tej transformacji

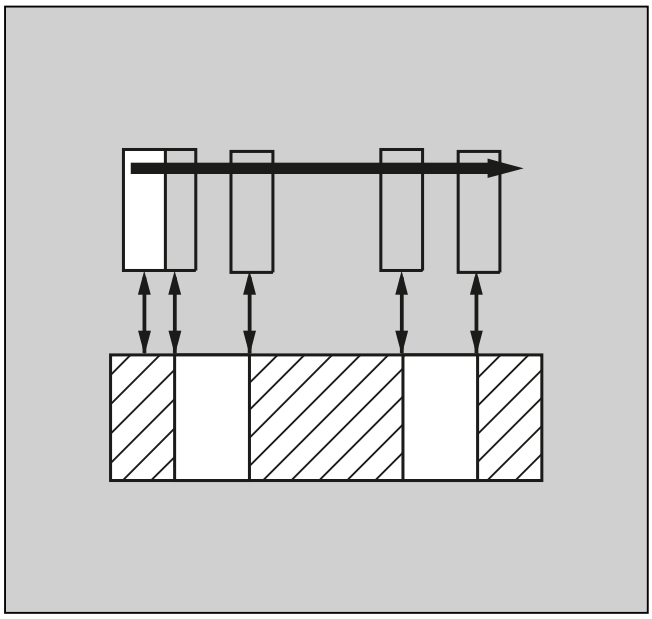
Odczyt wyników pomiarów

Wyniki pomiaru dla osi zmierzonych przy pomocy sondy pomiarowej można przeczytać poprzez następujące zmienne systemowe:

- $AA_MM[<oś>]
  Wyniki pomiaru w układzie współrzędnych maszyny
- $AA_MW[<oś>]
  Wyniki pomiaru w układzie współrzędnych obrabianego przedmiotu

5.8 Pomiary specyficzne dla osi (MEASA, MEAWA, MEAC) (opcja)

Przy pomiarze specyficznym dla osi można używać wielu sond pomiarowych i wielu systemów pomiarowych.



Dla programowania funkcji są do dyspozycji słowa kluczowe MEASA, MEAWA i MEAC.

Przy pomocy MEASA lub MEAWA jest dla każdorazowo zaprogramowanej osi odczytywanych do czterech wartości na pomiar i zapisywanych w zmiennych systemowych odpowiednio do zdarzenia przerzutnikowego.

Ciągłe zlecenia pomiaru mogą być przeprowadzane przy pomocy MEAC. W tym przypadku wyniki pomiaru są zapisywane w zmiennych FIFO.

Składnia

```
MEASA[<oś>]=(<tryb>,<TE1>,...,<TE4>)
MEAWA[<oś>]=(<tryb>,<TE1>,...,<TE4>)
MEAC[<oś>]=(<tryb>,<pamięć pomiarowa>,<TE1>,...,<TE4>)
```

Uwaga

MEASA i MEAWA działają pojedynczymi blokami i można je programować w jednym bloku. Jeżeli natomiast MEASA/MEAWA zostanie zaprogramowane w jednym bloku z MEAS/MEAW, nastąpi komunikat błędu.

Znaczenie

MEASA:	Słowo kluczowe: Pomiar osiowy z kasowaniem pozostałej drogi	
	Działanie:	pojedynczymi blokami

MEAWA:	Słowo kluczowe: Pomiar osiowy bez kasowania pozostałej drogi	
	Działanie:	pojedynczymi blokami
MEAC:	Słowo kluczowe: Ciągły pomiar osiowy bez kasowania pozostałej drogi	
	Działanie:	pojedynczymi blokami
<oś>:	Nazwa osi kanałowej użytej do pomiaru	
<tryb>:	Liczba dwucyfrowa do podania trybu pracy (tryb i system pomiaru)	
	Dekada jednostek (tryb pomiaru):	
	0	Anulowanie zlecenia pomiarowego.
	1	Maks. 4 różne równocześnie uaktywniane zdarzenia przerzutnikowe.
	2	Maks. 4 kolejno uaktywniane zdarzenia przerzutnikowe.
	3	Maks. 4 kolejno uaktywniane zdarzenia przerzutnikowe, ale nie ma nadzoru zdarzenia przerzutnikowego 1 przy starcie (alarmy 21700/21703 są ukrywane). **Wskazówka** Ten tryb jest przy MEAC niemożliwy.
	Dekada dziesiątek (system pomiarowy):	
	0 (albo brak informacji)	Aktywny system pomiarowy
	1	System pomiarowy 1
	2	System pomiarowy 2
	3	Obydwa systemy pomiarowe
<TE>:	Zdarzenie przerzutnikowe do wyzwolenia pomiaru	
	Typ:	INT
	Zakres wartości:	-2, -1, 1, 2
	Znaczenie:	
	(+)1	Zbocze rosnące od sondy pomiarowej 1
	-1	Zbocze opadające od sondy pomiarowej 1
	(+)2	Zbocze rosnące od sondy pomiarowej 2
	-2	Zbocze opadające od sondy pomiarowej 2
<Pamięć pomiarowa>:	Numer pamięci FIFO	

Przykłady

Przykład 1: Pomiar osiowy z kasowaniem pozostałej drogi w trybie 1 (ocena w kolejności czasowej)

a) z 1 systemem pomiarowym

Kod programu	Komentarz
...	
N100 MEASA[X]=(1,1,-1) G01 X100 F100	; Pomiar w trybie 1 z aktywnym systemem pomiarowym. Czekanie na sygnał pomiarowy o zboczu rosnącym/opadającym od sondy pomiarowej 1 na drodze ruchu do X=100.
N110 IF $AC_MEA[1]==FALSE GOTOF ENDE	; Skontrolowanie pomyślnego zakończenia pomiaru.

Kod programu	Komentarz
N120 R10=$AA_MM1[X]	; Zapisanie wartości zmierzonej należącej do pierwszego zaprogramowanego zdarzenia przerzutnikowego (zbocze rosnące).
N130 R11=$AA_MM2[X]	; Zapisanie wartości zmierzonej należącej do drugiego zaprogramowanego zdarzenia przerzutnikowego (zbocze opadające).
N140 ENDE:	

b) z 2 systemami pomiarowymi

Kod programu	Komentarz
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; Pomiar w trybie 1 z obydwoma systemami pomiarowymi. Czekanie na sygnał pomiarowy o zboczu rosnącym/opadającym od sondy pomiarowej 1 na drodze ruchu do X=100.
N210 IF $AC_MEA[1]==FALSE GOTOF ENDE	; Skontrolowanie pomyślnego zakończenia pomiaru.
N220 R10=$AA_MM1[X]	; Zapisanie wartości zmierzonej systemu pomiarowego 1 przy zboczu rosnącym.
N230 R11=$AA_MM2[X]	; Zapisanie wartości zmierzonej systemu pomiarowego 2 przy zboczu rosnącym.
N240 R12=$AA_MM3[X]	; Zapisanie wartości zmierzonej systemu pomiarowego 1 przy zboczu opadającym.
N250 R13=$AA_MM4[X]	; Zapisanie wartości zmierzonej systemu pomiarowego 2 przy zboczu opadającym.
N260 ENDE:	

Przykład 2: Pomiar osiowy z kasowaniem pozostałej drogi w trybie 2 (ocena w kolejności zaprogramowanej)

Kod programu	Komentarz
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; Pomiar w trybie 2 z aktywnym systemem pomiarowym. Czekanie na sygnał pomiarowy w kolejności zbocze rosnące od sondy pomiarowej 1, zbocze opadające sonda 1, zbocze rosnące od sondy 2, zbocze opadające sonda 2 na drodze ruchu do X = 100.
N110 IF $AC_MEA[1]==FALSE GOTOF MESSTASTER2	; Skontrolowanie pomyślnego pomiaru sondą pomiarową 1.
N120 R10=$AA_MM1[X]	; Zapisanie wartości zmierzonej należącej do pierwszego zaprogramowanego zdarzenia przerzutnikowego (zbocze rosnące sonda pomiarowa 1).

Kod programu	Komentarz
N130 R11=$AA_MM2[X]	; Zapisanie wartości pomiarowej należącej do drugiego zaprogramowanego zdarzenia przerzutnikowego (zbocze rosnące sonda pomiarowa 1).
N140 MESSTASTER2:	
N150 IF $AC_MEA[2]==FALSE GOTOF ENDE	; Skontrolowanie pomyślnego pomiaru sondą pomiarową 2.
N160 R12=$AA_MM3[X]	; Zapisanie wartości pomiarowej należącej do trzeciego zaprogramowanego zdarzenia przerzutnikowego (zbocze rosnące sonda pomiarowa 2).
N170 R13=$AA_MM4[X]	Zapisanie wartości pomiarowej należącej do czwartego zaprogramowanego zdarzenia przerzutnikowego (zbocze rosnące sonda pomiarowa 2).
N180 ENDE:	

Przykład 3: Pomiar ciągły osiowy w trybie 1 (ocena w kolejności czasowej)

a) Pomiar do 100 wartości

Kod programu	Komentarz
...	
N110 DEF REAL MESSWERT[100]	
N120 DEF INT Schleife=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; Pomiar w trybie 1 z aktywnym systemem pomiarowym, zapisanie wartości zmierzonych pod $AC_FIFO1, czekanie na sygnał pomiarowy o zboczu opadającym od sondy pomiarowej 1 na drodze ruchu do X=1000.
N135 STOPRE	
N140 MEAC[X]=(0)	; Przerwanie pomiaru po osiągnięciu pozycji osi.
N150 R1=$AC_FIFO1[4]	; Zapisanie liczby zebranych wartości zmierzonych w parametrze R1.
N160 FOR Schleife=0 TO R1-1	
N170 MESSWERT[Schleife]=$AC_FIFO1[0]	; Odczytanie i zapisanie wartości pomiarowych z $AC_FIFO1.
N180 ENDFOR	

b) Pomiar z kasowaniem pozostałej drogi po 10 wartościach zmierzonych

Kod programu	Komentarz
...	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; Skasowanie pozostałej drogi.
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	

```
Kod programu                                    Komentarz
N30 MEAC[X]=(0)
N40 R1=$AC_FIFO1[4]                             ; Liczba wartości zmierzonych.
...
```

c) Pomiar "opadającego/rosnącego" boku zęba przy użyciu 2 sond pomiarowych

```
Kod programu                                    Komentarz
...
N110 DEF REAL MESSWERT[16]
N120 DEF INT Schleife=0
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100           ; Pomiar w trybie 1 z aktywnym systemem pomiarowym, zapisanie wartości pomiarowych pod $AC_FIFO1, czekanie na sygnał pomiarowy w kolejności: zbocze opadające od sondy pomiarowej 1, zbocze rosnące sondy pomiarowej 2, na drodze ruchu do X=100.
N140 STOPRE                                     ; Zatrzymanie przebiegu wyprzedzającego.
N150 MEAC[X]=(0)                                ; Przerwanie pomiaru po osiągnięciu pozycji osi.
N160 R1=$AC_FIFO1[4]                            ; Zapisanie liczby zebranych wartości zmierzonych w parametrze R1.
N170 FOR Schleife=0 TO R1-1
N180 MESSWERT[Schleife]=$AC_FIFO1[0]            ; Odczytanie i zapisanie wartości pomiarowych z $AC_FIFO1.
N190 ENDFOR
```

Dodatkowe informacje

Zlecenie pomiarowe

Zaprogramowanie zlecenia pomiarowego może nastąpić w programie obróbki albo z akcji synchronicznej (patrz rozdział "Akcje synchroniczne (Strona 631)"). Na oś może przy tym w jednej i tej samej chwili być aktywne tylko jedno zlecenie pomiarowe.

Uwaga

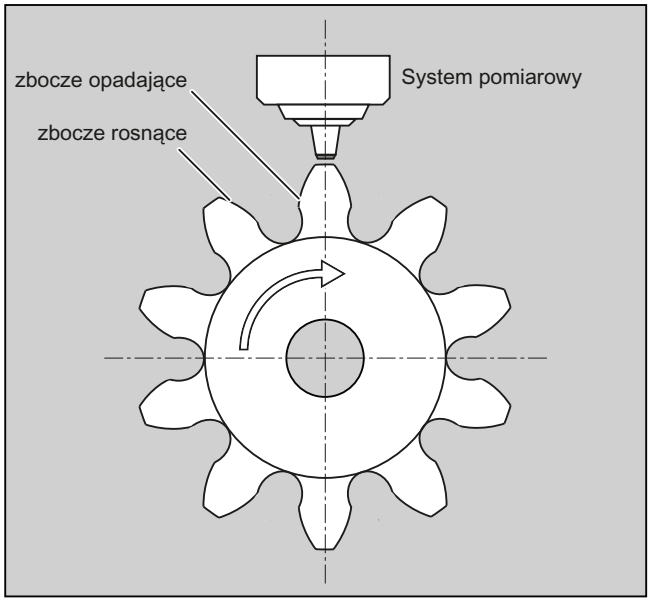
Posuw należy dopasować do każdorazowego zadania pomiarowego.

W przypadku `MEASA` i `MEAWA` prawidłowe wyniki mogą zostać zagwarantowane tylko w przypadku posuwów, przy których następuje nie więcej, niż jedno takie samo i nie więcej, niż 4 różne zdarzenia przerzutnikowe na takt regulacji położenia.

Przy pomiarze ciągłym przy pomocy `MEAC` stosunek między taktem interpolatora i taktem regulatora położenia nie może być większy, niż 1:8.

Zdarzenie przerzutnikowe

Zdarzenie przerzutnikowe składa się z numeru sondy pomiarowej i kryterium wyzwalania (zbocze rosnące albo malejące) sygnału pomiarowego.



Dla każdego pomiaru można przetwarzać każdorazowo do 4 zdarzeń przerzutnikowych uaktywnionej sondy pomiarowej, a więc do dwóch sond pomiarowych po dwa zbocza pomiarowe. Kolejność przetwarzania jak też maksymalna liczba zdarzeń przerzutnikowych, są przy tym zależne od wybranego trybu.

Uwaga

Dla trybu pomiaru 1 obowiązuje: Takie samo zdarzenie przerzutnikowe wolno tylko jeden raz zaprogramować w zleceniu pomiarowym!

W przypadku `MEAC` liczba wartości pomiarowych na zdarzenie przerzutnikowe może zostać podwyższona przez zastosowanie telegramu PROFIBUS 395 do łącznie 8 wartości pomiarowych przy rosnącym i 8 przy opadającym zboczu na zdarzenie przerzutnikowe i takt regulatora położenia.

- Jedna sonda pomiarowa: 8 wartości pomiarowych przy rosnącym i 8 przy opadającym zboczu
- Dwie sondy pomiarowe: 4 wartości pomiarowe przy rosnącym i 4 przy opadającym zboczu na sondę

W taki sposób da się realizować wyższe posuwy lub prędkości obrotowe przy zastosowaniu telegramu PROFIBUS 395.

Literatura

Podręcznik działania, Funkcje rozszerzone; Pomiar (M5), Rozdział: Pomiar osiowy

Tryb pracy

Przy pomocy pierwszej cyfry (dziesiątki) trybu pracy jest wybierany żądany system pomiarowy. Gdy jest tylko jeden system pomiarowy, a został jednak zaprogramowany, drugi, zostaje automatycznie zastosowany system istniejący.

Przy pomocy drugiej cyfry (jednostki) jest wybierany żądany tryb pomiaru. Przez to proces pomiaru jest dopasowywany do możliwości każdorazowego sterowania:

- **Tryb 1**
  Ocena zdarzeń przerzutnikowych następuje w czasowej kolejności ich wystąpienia. W tym trybie przy zastosowaniu sześcioosiowych grup osi możliwe jest zaprogramowanie tylko jednego zdarzenia przerzutnikowego wzgl. przy podaniu wielu zdarzeń przerzutnikowych następuje automatyczne przejście na tryb 2 (bez komunikatu).
- **Tryb 2**
  Ocena zdarzeń przerzutnikowych następuje w kolejności zaprogramowanej.
- **Tryb 3**
  Ocena zdarzeń przerzutnikowych następuje w kolejności zaprogramowanej, ale nie ma nadzoru zdarzenia przerzutnikowego 1 przy STARCIE.

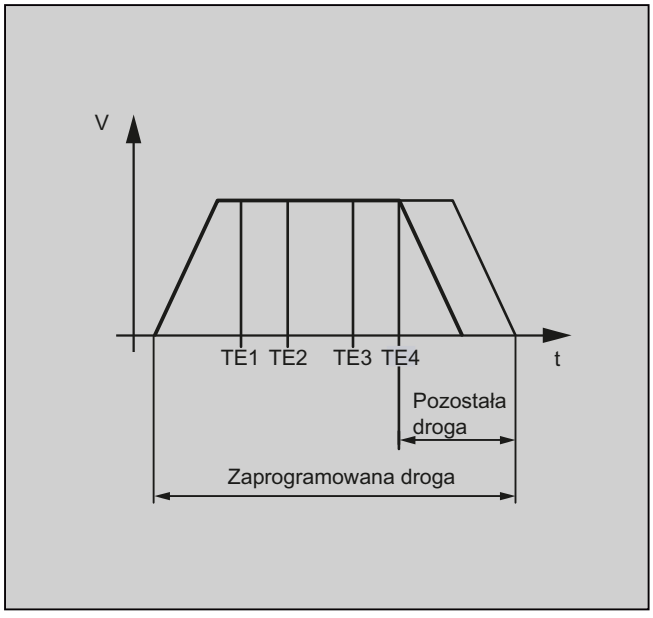
Uwaga

Przy zastosowaniu 2 systemów pomiarowych dają się zaprogramować tylko dwa zdarzenia przerzutnikowe.

Pomiar z i bez kasowania pozostałej drogi

Przy zaprogramowaniu `MEASA` kasowanie pozostałej drogi jest przeprowadzane dopiero po zarejestrowaniu wszystkich wymaganych wartości pomiarowych.

Dla specjalnych zadań pomiarowych, w przypadku których w każdym przypadku ma zostać dokonane dosunięcie do zaprogramowanej pozycji, stosuje się `MEAWA`.



Uwaga

`MEASA` nie daje się programować w akcjach synchronicznych. Zastępczo można zaprogramować `MEAWA` plus skasowanie pozostałej drogi jako akcja synchroniczna.

Jeżeli zlecenie pomiaru jest uruchamiane przy pomocy `MEAWA` z akcji synchronicznych, wartości pomiarowe są dostępne tylko w układzie współrzędnych maszyny.

Wyniki pomiaru dla MEASA, MEAWA

Wyniki pomiaru są do dyspozycji pod następującymi zmiennymi systemowymi:

- W układzie współrzędnych maszyny:

`$AA_MM1[<oś>]`	Wartość pomiarowa zaprogramowanego systemu pomiarowego przy zdarzeniu przerzutnikowym 1
`...`	...
`$AA_MM4[<oś>]`	Wartość pomiarowa zaprogramowanego systemu pomiarowego przy zdarzeniu przerzutnikowym 4

- W układzie współrzędnych obrabianego przedmiotu:

`$AA_MW1[<oś>]`	Wartość pomiarowa zaprogramowanego systemu pomiarowego przy zdarzeniu przerzutnikowym 1
`...`	...
`$AA_MW4[<oś>]`	Wartość pomiarowa zaprogramowanego systemu pomiarowego przy zdarzeniu przerzutnikowym 4

Osie geometryczne / transformacje

Jeżeli ma zostać uruchomiony pomiar osiowy dla osi geometrycznej, to samo zlecenie pomiaru musi zostać bezpośrednio zaprogramowane dla wszystkich pozostałych osi geometrycznych. To samo dotyczy osi, które uczestniczą w transformacji.

Przykłady:
```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

lub
```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Zlecenie pomiarowe z 2 systemami pomiarowymi

Jeżeli jest przeprowadzane zlecenie pomiaru z dwoma systemami pomiarowymi, każde z obydwu możliwych zdarzeń przerzutnikowych jest rejestrowane przez obydwa systemy każdej osi. Zajętość zarezerwowanych zmiennych jest przez to zadana:

`$AA_MM1[<oś>]`	wzgl.	`$AA_MW1[<oś>]`	Wartość pomiarowa od systemu pomiarowego 1 przy zdarzeniu przerzutnikowym 1
`$AA_MM2[<oś>]`	wzgl.	`$AA_MW2[<oś>]`	Wartość pomiarowa od systemu pomiarowego 2 przy zdarzeniu przerzutnikowym 1

`$AA_MM3[<oś>]`	wzgl.	`$AA_MW3[<oś>]`	Wartość pomiarowa od systemu pomiarowego 1 przy zdarzeniu przerzutnikowym 2
`$AA_MM4[<oś>]`	wzgl.	`$AA_MW4[<oś>]`	Wartość pomiarowa od systemu pomiarowego 2 przy zdarzeniu przerzutnikowym 2

Zmienne systemowe

Status sondy pomiarowej jest do dyspozycji pod następującymi zmiennymi systemowymi:

$A_PROBE[<n>]

Wartość	Znaczenie
1	Sonda pomiarowa wychylona
0	Sonda pomiarowa nie wychylona

Ograniczenie sondy pomiarowej jest do określenia pod następującymi zmiennymi systemowymi:

$A_PROBE_LIMITED[<n>]

Wartość	Znaczenie
1	Ograniczenie sondy pomiarowej aktywne
0	Ograniczenie sondy pomiarowej nieaktywne

<n> = sonda pomiarowa

Literatura:
Podręcznik lista parametrów - zmienne systemowe

Status zlecenia pomiarowego w przypadku MEASA, MEAWA

Jeżeli w programie jest wymagana ocena, wówczas można dokonać odczytania statusu zlecenia pomiarowego przez $AC_MEA\[<n>], gdzie n = numer sondy pomiarowej. Gdy wszystkie zaprogramowane w bloku zdarzenia przełączające sond pomiarowych <n> nastąpiły, zmienna ta daje wartość 1. W przeciwnym przypadku wartość wynosi 0.

Uwaga

Jeżeli pomiar jest uruchamiany z akcji synchronicznych, $AC_MEA nie jest już aktualizowane. W tym przypadku należy odczytać nowy sygnał interfejsu DB31, … DBX62.3 wzgl. równoważną zmienną $AA_MEAACT[<oś>].

Znaczenie:

$AA_MEAACT==1: Pomiar aktywny

$AA_MEAACT==0: Pomiar nieaktywny

Pomiar ciągły (MEAC)

Wartości pomiarowe w przypadku `MEAC` są zapisywane w podanej pamięci FIFO\[n] w układzie współrzędnych maszyny. Jeżeli dla pomiarów są zaprojektowane dwie sondy pomiarowe,

wartości pomiarowe drugiej sondy są zapisywane oddzielnie w zaprojektowanej w tym celu (ustawianej przez MD) pamięci FIFO[n+1].

Pamięć FIFO jest pamięcią obiegową, do której wartości pomiarowe są wpisywane na zasadzie obiegu w zmiennych $AC_FIFO, patrz rozdział "Akcje synchroniczne (Strona 631)".

Uwaga

Treść pamięci FIFO może być odczytana tylko jeden raz. W celu wielokrotnego użycia danych pomiarowych muszą one być poddane zapisaniu pośredniemu w danych użytkownika.

Gdy liczba wartości pomiarowych przekracza liczbę maksymalną ustaloną w danej maszynowej dla pamięci FIFO, wówczas pomiar ulega automatycznemu zakończeniu.

Pomiar bez końca daje się zrealizować przez cykliczny odczyt wartości pomiarowych. Odczyt musi przy tym następować co najmniej z taką samą częstotliwością co wpływ nowych wartości pomiarowych.

Literatura:

- Podręcznik działania, Akcje synchroniczne; Opis szczegółowy,
  Rozdział: Parametry ($AC_FIFO)
- Podręcznik działania, Funkcje rozszerzone; Pomiar (M5), Rozdział: Pomiar osiowy

Ochrona przed błędnym programowaniem

Następujące błędy programowania są rozpoznawane i wyświetlane jako błąd:

- `MEASA`/`MEAWA` zaprogramowano w jednym bloku razem z `MEAS`/`MEAW`
  Przykład:
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- `MEASA`/`MEAWA` z liczbą parametrów <2 lub >5
  Przykład:
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```
- `MEASA`/`MEAWA` ze zdarzeniem przerzutnikowym nierównym 1/ -1/ 2/ -2
  Przykład:
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- `MEASA`/`MEAWA` z nieprawidłowym trybem
  Przykład:
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- `MEASA`/`MEAWA` z podwójnie zaprogramowanym zdarzeniem przerzutnikowym
  Przykład:
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```
- `MEASA`/`MEAWA` i brakująca oś geometryczna
  Przykład:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;oś GEO X/Y/
  Z
  ```
- Niejednolite zadanie pomiarowe w przypadku osi geometrycznych
  Przykład:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50
  F100
  ```

5.9 Funkcje specjalne dla użytkownika OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

Adresy OEM

Znaczenie adresów OEM określa użytkownik OEM. Działanie jest określane przez cykle kompilowane. 5 adresów OEM jest zarezerwowanych (`OMA1` ... `OMA5`). Identyfikatory adresów można ustawiać. Adresy OEM są dopuszczalne w każdym bloku.

Zarezerwowane wywołania funkcji G

Dla użytkownika OEM są zarezerwowane następujące wywołania funkcji G:

- `OEMIPO1`, `OEMIPO2` (z grupy G nr 1)
- `G810` ... `G819` (grupa G nr 31)
- `G820` ... `G829` (grupa G nr 32)

Działanie jest określane przez cykle kompilowane.

Funkcje i podprogramy

Dodatkowo użytkownicy OEM mogą również tworzyć predefiniowane funkcje i podprogramy z przekazaniem parametrów.

Uwaga

Symulacja obrabianego przedmiotu

Do w. opr. 4.4 nie są w symulacji obrabianego przedmiotu obsługiwane żadne cykle, a od w. opr. 4.4 tylko niektóre cykle kompilowane (CC).

Polecenia językowe w programie obróbki nie obsługiwanych cykli kompilowanych (`OMA1` ... `OMA5`, `OEMIPO1/2`, `G810` ... `G829`, własne procedury i funkcje) prowadzą do braku indywidualnego ich traktowania oraz do komunikatu alarmowego i do anulowania symulacji.

Rozwiązanie: Należy indywidualnie traktować specyficzne dla CC elementy językowe w programie obróbki (odpytanie $P_SIM).
Przykład:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;podczas symulacji CC nieaktywne
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 Zmniejszenie posuwu ze zwłoką na narożach (FENDNORM, G62, G621)

Przy automatycznej zwłoce na narożach posuw jest zmniejszany (kształt dzwona) na krótko przed odnośnym narożnikiem. Poza tym mające znaczenie dla obróbki zachowanie się narzędzia może być parametryzowane przez dane ustawcze. Są to:

- Początek i koniec zmniejszenia posuwu
- Korekcja, z którym posuw jest zmniejszany
- Rozpoznanie znaczącego narożnika

Jako znaczące narożniki są uwzględniane te narożniki, których kąt wewnętrzny jest mniejszy niż narożnik sparametryzowany przez daną ustawczą.

Przy pomocy wartości domyślnej `FENDNORM` funkcja automatycznej korekcji na narożnikach jest wyłączana.

Literatura:
/FBFA/ Opis działania Dialekty ISO

Składnia

```
FENDNORM
G62 G41
G621
```

Znaczenie

`FENDNORM`:	Automatyczna zwłoka na narożnikach wył.
`G62`:	Zwłoka w narożnikach wewnętrznych przy aktywnej korekcji promienia narzędzia
`G621`:	Zwłoka na wszystkich narożnikach przy aktywnej korekcji promienia narzędzia

G62 działa tylko na narożnikach wewnętrznych z

- aktywną korekcją promienia narzędzia `G41`, `G42` i
- aktywnym przechodzeniu płynnym `G64`, `G641`

Ruch do odpowiedniego narożnika następuje ze zmniejszonym posuwem, który wynika z:

F * (korekcja do redukcji posuwu) * korekcja posuwu

Maksymalnie możliwe obniżenie posuwu jest uzyskiwane dokładnie wtedy, gdy narzędzie, w odniesieniu do toru punktu środkowego, ma dokonać zmiany kierunku na odnośnym narożniku.

G621 działa analogicznie do G62 na każdym narożniku, osi ustalonych przez `FGROUP`.

5.11 Programowane kryterium końca ruchu (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Podobnie do kryterium zmiany bloku przy interpolacji ruchu po torze (G601, G602 i G603), kryterium końca ruchu przy interpolacji pojedynczej osi może zostać zaprogramowane w programie obróbki wzgl. w akcjach synchronicznych dla osi rozkazowych/PLC.

Zależnie od tego, które kryterium końca ruchu jest ustawione, bloki programu obróbki wzgl. bloki cyklu technologicznego ulegają w różny sposób szybkiemu zakończeniu. To samo obowiązuje dla PLC przez FC15/16/18.

Składnia

```
FINEA[<oś>]
COARSEA[<oś>]
IPOENDA[<oś>]
IPOBRKA(<oś>[,<chwila>])
ADISPOSA(<oś>[,<tryb>,<wielkość okna>])
```

Znaczenie

FINEA:	Kryterium końca ruchu: "Zatrzymanie dokładne dokładnie"		
	Działanie:	Modalnie	
COARSEA:	Kryterium końca ruchu: "Zatrzymanie dokładne zgrubnie"		
	Działanie:	Modalnie	
IPOENDA:	Kryterium końca ruchu:"Interpolator-Stopp"		
	Działanie:	Modalnie	
IPOBRKA:	Kryterium zmiany bloku: Rampa hamowania		
	Działanie:	Modalnie	
ADISPOSA:	Okno tolerancji dla kryterium końca ruchu		
	Działanie:	Modalnie	
<oś>:	Nazwa osi kanału (X, Y, ...)		
<chwila>:	Chwila zmiany bloku, odniesiona do charakterystyki hamowania w %: • 100% = początek charakterystyki hamowania • 0% = koniec charakterystyki hamowania, równoznaczny z IPOENDA		
	Typ:	REAL	
<tryb>:	Odniesienie okna tolerancji		
	Zakres wartości:	0	Okno tolerancji nie aktywne
		1	Okno tolerancji odnośnie pozycji zadanej
		2	Okno tolerancji odnośnie pozycji rzeczywistej
	Typ:	INT	
<wielkość okna>:	Wielkość okna tolerancji		
	Typ:	REAL	

Przykłady

Przykład 1: Kryterium końca ruchu: „Zatrzymanie interpolatora”

```
Kod programu
; Wykonaj ruch w osi pozycjonowania X na 100, prędkość
200 m/min, przyspieszenie 90%,
; Kryterium końca ruchu: Zatrzymanie interpolatora
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]

; Akcja synchroniczna:
; ZAWSZE GDY: Jest ustawione wejście 1
; WÓWCZAS Wykonaj ruch w osi pozycjonowania X na 50,
prędkość 200 m/min, przyspieszenie 140%,
;      Kryterium końca ruchu: Zatrzymanie interpolatora
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

Przykład 2: Kryterium zmiany bloku: „Rampa hamowania”

```
Kod programu          Komentarz
                      ; Ustawienie domyślne działa
N40 POS[X]=100        ; Ruch pozycjonowania w X na pozycję 100
                      Kryterium zmiany bloku: Zatrzymanie dokładne dokładnie
N20 IPOBRKA(X,100)    ; Kryterium zmiany bloku „Rampa hamowania”,
                      100% = początek charakterystyki hamowania
N30 POS[X]=200        ; Zmiana bloku następuje, gdy tylko oś X zaczyna hamować
N40 POS[X]=250        ; Oś X nie hamuje na pozycji 200, lecz wykonuje nadal ruch
                      dalej na pozycję 250.
                      Gdy tylko oś rozpocznie hamowanie, następuje zmiana bloku.
N50 POS[X]=0          ; Oś X hamuje i wykonuje ruch z powrotem do pozycji 0
                      Zmiana bloku następuje na pozycji 0 i "zatrzymaniu dokładnym
                      dokładnie"
N60 X10 F100          ; Oś X wykonuje ruch do pozycji 10, jako oś uczestnicząca w
                      tworzeniu konturu
```

Dalsze informacje

Zmienna systemowa dla kryterium końca ruchu

Działające kryterium końca ruchu można odczytać z zmiennej systemowej $AA_MOTEND.

Literatura: /LIS2sl/ Podręcznik lista, podręcznik 2

Kryterium zmiany bloku: "Charakterystyka hamowania" (IPOBRKA)

Jeżeli przy uaktywnieniu kryterium zmiany bloku "charakterystyka hamowania" jest dla opcji: czas zmiany bloku zaprogramowana wartość, działa ona dla następnego ruchu pozycjonowania i zostanie synchronicznie z przebiegiem głównym zapisana w danej

ustawczej. Jeżeli nie podano wartości czasu zmiany bloku, działa aktualna wartość danej ustawczej.

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Z następnym zaprogramowaniem osiowego kryterium końca ruchu (`FINEA`, `COARSEA` , `IPOENDA`) następuje wyłączenie aktywności `IPOBRKA` dla odpowiedniej osi.

Kryterium zmiany bloku: "okno tolerancji" (ADISPOSA)

Przy pomocy `ADISPOSA` można, jako dodatkowe kryterium zmiany bloku, zdefiniować okno tolerancji wokół punktu końcowego bloku (do wyboru pozycja rzeczywista albo zadana). Dla zmiany bloku muszą być wówczas spełnione obydwa warunki:

- Kryterium zmiany bloku: "Charakterystyka hamowania"
- Kryterium zmiany bloku: "Okno tolerancji"

Literatura

Dalsze informacje dot. kryterium zmiany bloku osi pozycjonowania patrz:

- Podręcznik działania Funkcje rozszerzające; Osie pozycjonowania (P2)
- Podręcznik programowania Podstawy; punkt "Regulacja posuwu"

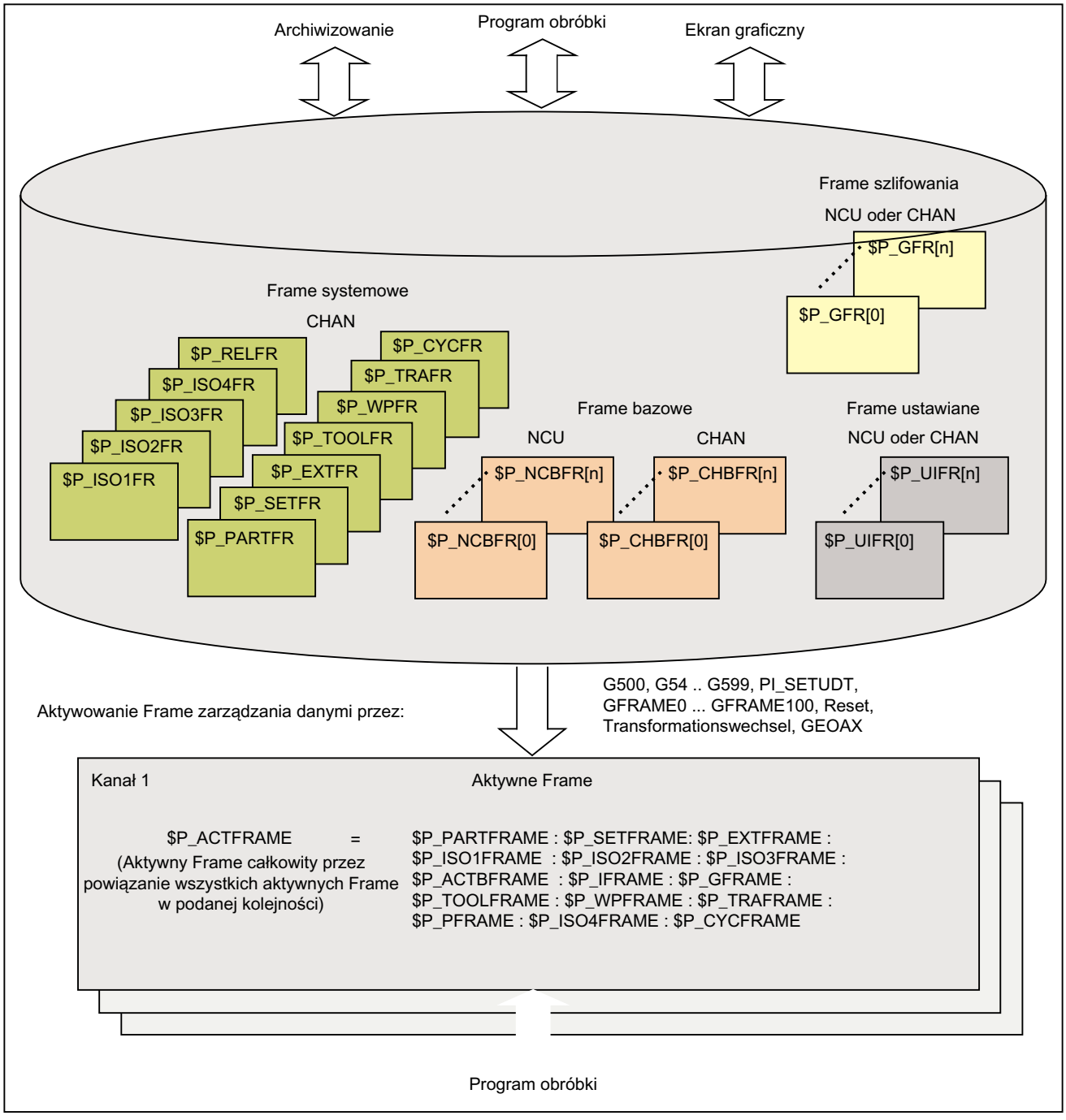
Transformacje współrzędnych (Frame) 6

6.1 Transformacja współrzędnych przez zmienną frame

Oprócz poleceń opisanych w Podręczniku programowania Podstawy, punkt "Transformacje współrzędnych (frame)", np. `ROT`, `AROT`, `SCALE`, itd. układ współrzędnych obrabianego przedmiotu (WKS) może zostać poddany transformacji również poprzez zmienne frame $P_...FR (frame zarządzania danymi) i $P_...FRAME (aktywne frame).

Poniższy rysunek daje przegląd strukturyzacji zmiennych frame:

- Frame zarządzania danymi
- Aktywne frame
- Aktywny frame łączny: Powiązanie wszystkich aktywnych frame
- Frame globalne dla NCU
- Frame specyficzne dla kanału



Rysunek 6-1 Przegląd zmiennych frame

6.1.1 Predefiniowana zmienna frame ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

Aktywne: Specyficzne dla kanału Frame bazowe $P_CHBFRAME[<n>] ($P_BFRAME)

Uwaga

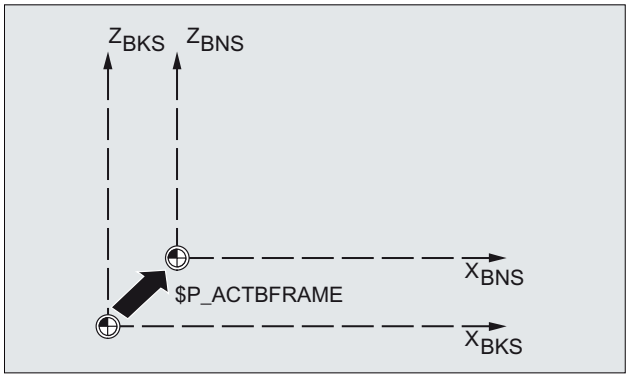
Aktualny frame bazowy`$P_BFRAME` i frame bazowy zarządzania danymi `$P_UBFR` są zachowywane z powodów kompatybilności.

- `$P_BFRAME` □ `$P_CHBFRAME[0]`
- `$P_UBFR` □ `$P_CHBFR[0]`.

Zmienne frame $P_CHBFRAME[<n>], definiują odniesienie między bazowym układem współrzędnych (BKS) i bazowym układem współrzędnych punktu zerowego (BNS).

Jeżeli aktualny, specyficzny dla kanału frame bazowy $P_CHBFRAME[`<n>`] ma natychmiast działać w programie NC, są do dyspozycji następujące możliwości

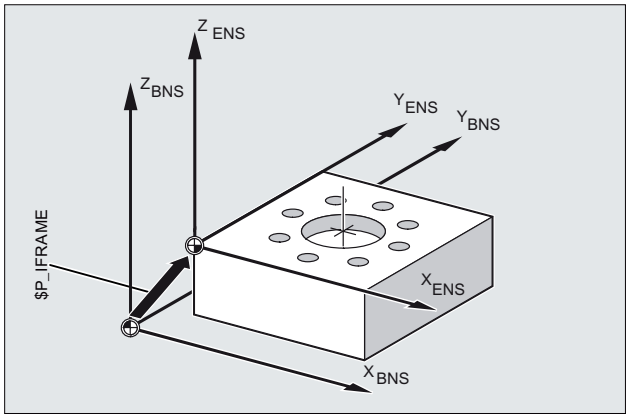
- Polecenia:
  - `G500` (wyłączenie wszystkich frame ustawianych, frame bazowe pozostają aktywne)
  - `G54` ... `G599` (ustawiane przesunięcia punktu zerowego)
- Przyporządkowanie specyficznego dla kanału bazowego frame zarządzania danymi do aktualnego specyficznego dla kanału frame bazowego:
  `$P_CHBFRAME[<n>] = $P_CHBFR[<m>]`



Aktywne: Specyficzny dla kanału frame ustawiany $P_IFRAME

Zmienna frame $P_IFRAME, definiuje odniesienie między bazowym układem punktu zerowego (BNS) i ustawianym układem punktu zerowego (ENS).

- `$P_IFRAME` odpowiada `$P_UIFR[$P_IFRNUM]`
- `$P_IFRAME` zawiera po zaprogramowaniu np. `G54` zdefiniowane przez `G54` przesunięcie, obrót, skalowanie i lustrzane odbicie.

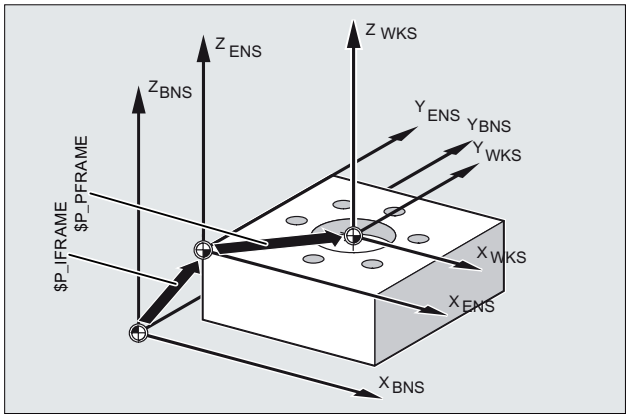


Aktywne: Specyficzny dla kanału frame programowany $P_PFRAME

Zmienna frame $P_PFRAME, definiuje odniesienie między ustawianym układem punktu zerowego (ENS) i układem współrzędnych obrabianego przedmiotu (WKS).

`$P_PFRAME` zawiera wynikający frame, który wynika

- **z zaprogramowania** `TRANS/ATRANS`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` wzgl.
- **z przyporządkowania** `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` do FRAME programowanego

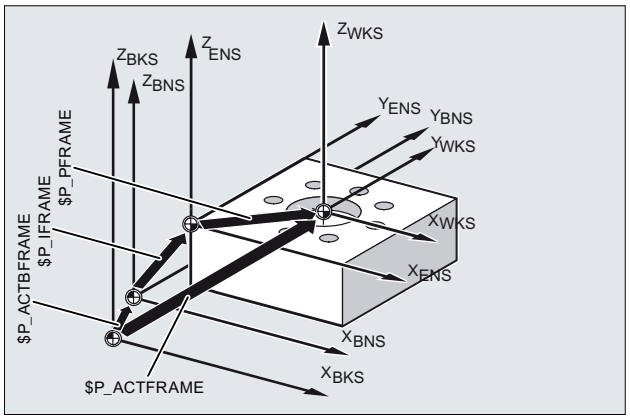


Aktywne: Frame łączny $P_ACTFRAME

Działający w kanale frame łączny, wynika z powiązania wszystkich frame działających w kanale.

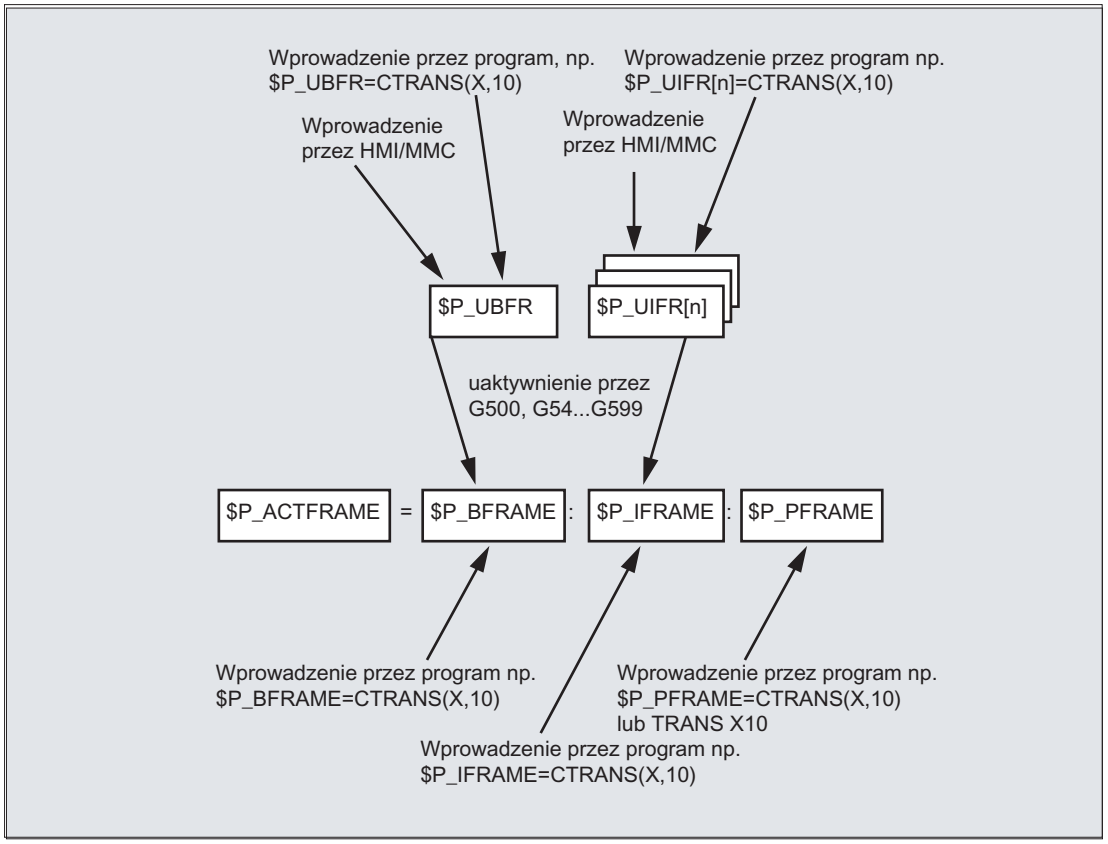
```
$P_ACTFRAME =    $P_PARTFRAME : $P_SETFRAME  : $P_EXTFRAME :
                 $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
                 $P_ACTBFRAME : $P_IFRAME    : $P_GFRAME :
                 $P_TOOLFRAME : $P_WPFRAME   : $P_TRAFRAME :
                 $P_PFRAME    : $P_ISO4FRAME : $P_CYCFRAME
```

$P_ACTFRAME opisuje punkt zerowy aktualnego układu współrzędnych obrabianego przedmiotu.



Rysunek 6-2 Zmienna frame $P_ACTFRAME

Jeżeli jeden z następujących frame $P_BFRAME / $P_CHBFRAME[<n>], $P_IFRAME albo $P_PFRAME zostanie zmieniony, aktualny frame łączny$P_ACTFRAME zostanie obliczony na nowo.



Frame bazowy i frame ustawiany działają po reset, gdy MD 20110 RESET_MODE_MASK jest ustawiona następująco:

Bit0=1, Bit14=1 --> $P_UBFR (frame bazowy) działa

Bit0=1, Bit5=1 --> `$P_UIFR[$P_UIFRNUM]` (frame ustawiany) działa

Zarządzanie danymi: Kanałowe frame bazowe $P_CHBFR[<n>]

Poprzez zmienne frame $P_CHBFR[<n>] frame bazowe są czytane/zapisywane w zarządzaniu danymi. Frame zarządzania danymi nie staje się aktywny w kanale natychmiast po zapisaniu. Uaktywnienie zapisanego frame następuje przy:

- Reset kanału i MD20110 $MC_RESET_MODE_MASK,Bit0 == 1 i Bit14 == 1
- Polecenie `G500`, `G54` ... `G57`, `G505` ... `G599` (wł./wył. frame bazowych z następnym ponownym obliczeniem aktualnego frame łącznego)

Zarządzanie danymi: Specyficzne dla kanału frame ustawiane $P_UIFR[<n>]

Poprzez zmienne frame $P_UIFR[<n>] frame ustawiane są czytane/zapisywane w zarządzaniu danymi. Frame nie staje się aktywny w kanale natychmiast po zapisaniu. Wliczenia zapisanego frame w kanale następuje przy:

- Polecenie `G500` (wyłączenie wszystkich frame ustawianych wzgl. przesunięć punktu zerowego)
- Polecenie `G54` ... `G57`, `G505` ... `G599` (włączenie frame ustawianego wzgl. przesunięcia punktu zerowego)

Aktywny frame ustawiany	Frame zarządzania danymi	(odpowiada poleceniu)
`$P_IFRAME` =	`$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 Przyporządkowania wartości do Frames

6.2.1 Bezpośrednie przyporządkowanie wartości (wartość osi, kąt, skala)

W programie NC można frame albo zmienne frame bezpośrednio wyposażać w wartości.

Składnia

Składnia

```
$P_PFRAME = CTRANS(X, <wartość przesunięcia>, Y, <wartość
przesunięcia>, Z, <wartość przesunięcia>, ...)
$P_PFRAME = ROT(X, <kąt>, Y, <kąt>, Z, <kąt>, ...)
$P_UIFR[..] = CROT(X, <kąt>, Y, <kąt>, Z, <kąt>,  ...
$P_PFRAME = CSCALE(X, <skala>, Y, <skala>, Z, <skala>, ...)
$P_PFRAME = CMIRROR(X, Y, Z)
```

Składnia `$P_CHBFRAME[<n>]` jest identyczna z `$P_PFRAME`.

Znaczenie

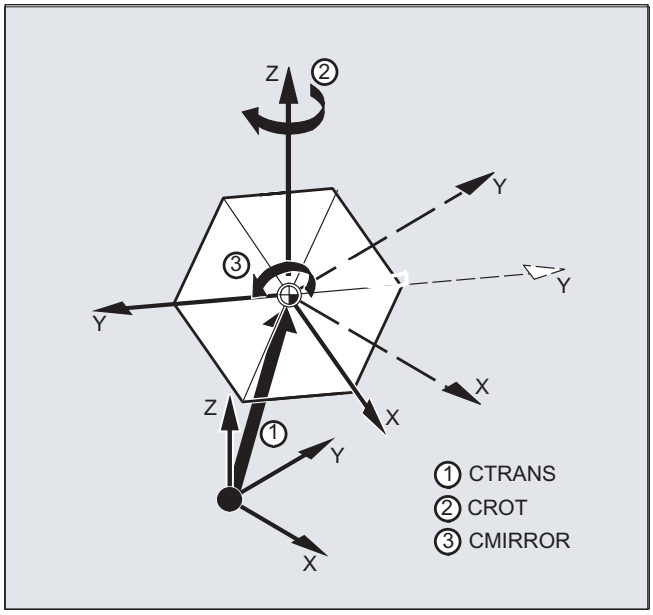
`CTRANS`:	Przesunięcie w podanych osiach
`CROT`:	Obrót wokół podanych osi
`CSCALE`:	Zmiana skali w podanych osiach
`CMIRROR`:	Odwrócenie kierunku podanej osi
`X, Y, Z`:	Wartość przesunięcia w kierunku podanej osi geometrycznej
`<wartość przesunięcia>`:	Wartość przesunięcia
`<kąt>`:	Kąt, o który następuje obrót
`<skala>`:	Podanie skali

Przykłady

Przyporządkowania wartości do komponentów frame aktualnego frame programowanego

Przyporządkowanie wartości do komponentów frame translacja, obrót i lustrzane odbicie.

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```



Zapisanie składowych obrotu frame

Przyporządkowanie wartości do wszystkich trzech osi składowej obrotu ustawianego frame zarządzania danymi `$P_UIFR` z `CROT`:

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

Alternatywnie do tego bezpośrednie przyporządkowanie poszczególnych wartości bezpośrednio do każdorazowej osi składowej obrotu frame zarządzania danymi.

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

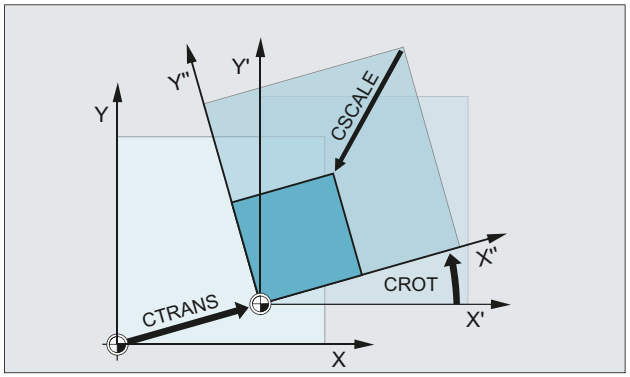
Opis

Wiele operacji na jeden frame można ze sobą powiązać przez operator powiązania `:`. Operacje są wówczas wykonywane kolejno od lewej ku prawej.

Przykład

Operacje powiązane na $P_PFRAME z przesunięciem, obrotem i skalowaniem:

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```



6.2.2 Odczyt i zmiana komponentów frame (TR, FI, RT, SC, MI)

Istnieje możliwość sięgania do **pojedynczych** danych danego frame, np. do określonej wartości przesunięcia albo kąta obrotu. Te wartości można zmienić albo przyporządkować do innej zmiennej.

Składnia

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	Kąt obrotu RT wokół osi X z aktualnie obowiązującego ustawianego przesunięcia punktu zerowego $P_UIFRNUM ma zostać przyporządkowany do zmiennej R10.
`R12=$P_UIFR[25,Z,TR]`	Wartość przesunięcia TR w Z z zestawu danych ustawionego frame nr 25 ma zostać przyporządkowana do zmiennej R12.
`R15=$P_PFRAME[Y,TR]`	Wartość przesunięcia TR w Y aktualnego programowanego frame ma zostać przyporządkowana do zmiennej R15.
`$P_PFRAME[X,TR]=25`	Wartość przesunięcia TR w X aktualnego programowanego frame ma zostać zmieniona. Od teraz obowiązuje X25.

Znaczenie

`$P_UIFRNUM`:	`Przy pomocy tej zmiennej jest automatycznie tworzone odniesienie do aktualnie obowiązującego ustawianego przesunięcia punktu zerowego.`
`P_UIFR[n,…,…]`:	`Poprzez podanie numeru frame n sięgamy do ustawianego frame numer n.`
	`Podanie składowej, który ma być czytana albo zmieniona:`
`TR`:	`TR translacja`
`FI`:	`FI translacja dokładnie`
`RT`:	`RT obrót`

SC:	SC Scale zmiana skali
MI:	MI lustrzane odbicie
X, Y, Z:	Dodatkowo (patrz przykłady) jest podawana odpowiednia oś X, Y, Z.

Zakres wartości dla obrotu RT

Obrót wokół 1. osi geometrycznej: -180° do +180°
Obrót wokół 2. osi geometrycznej: -90° bis +90°
Obrót wokół 3. osi geometrycznej: -180° do +180°

Opis

Wywołanie frame

Określając zmienną systemową $P_UIFRNUM można uzyskać bezpośredni dostęp do przesunięcia punktu zerowego aktualnie ustawionego przy pomocy $P_UIFR lub G54, G55, … ($P_UIFRNUM zawiera numer aktualnie ustawionego frame).

Wszystkie inne zapisane w pamięci ustawiane frame $P_UIFR wywołujemy przez podanie odpowiedniego numeru $P_UIFR[n].

Dla predefiniowanej zmiennej frame z frame zdefiniowanych przez użytkownika należy podać nazwę, np. $P_IFRAME.

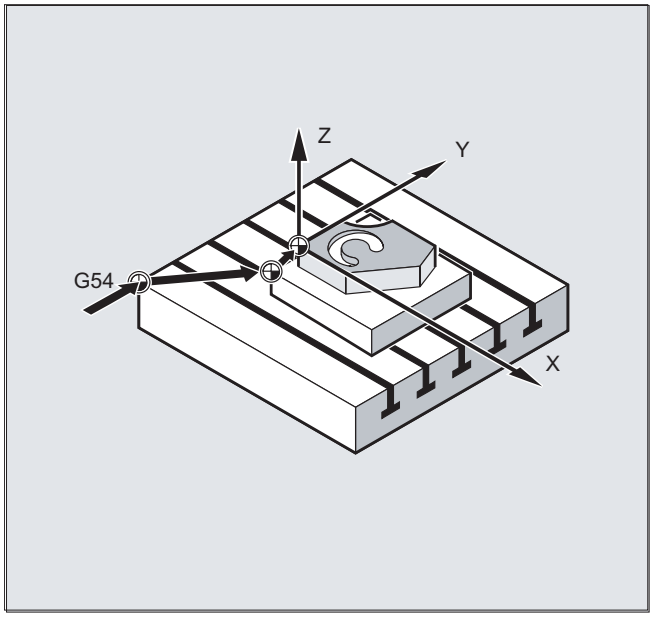
Wywołanie danych

Nawiasy kwadratowe zawierają nazwę osi i składową frame wartości, do której chcemy uzyskać dostęp lub którą chcemy zmienić, np. [X, RT] albo [Z, MI].

6.2.3 Obliczanie z frame

W programie NC frame może zostać przyporządkowany do innego frame albo frame mogą zostać ze sobą powiązane.

Powiązania frame nadają się np. do opisu wielu obrabianych przedmiotów, które są umieszczone na palecie i mają być obrabiane w jednym przebiegu.



Dla opisu zadań obróbki na paletach składowe frame mogą zawierać tylko określone wartości częściowe, przez których powiązanie są generowane różne punkty zerowe obrabianego przedmiotu.

Przykłady

Przyporządkowania

Kod programu	**Komentarz**
`DEF FRAME USTAWIENIE_1`	`; Definicja lokalnej zmiennej frame`
`USTAWIENIE_1 = CTRANS(X,10)`	`; Przyporządkowanie wyniku funkcji do zmiennej frame`
`$P_PFRAME = USTAWIENIE_1`	`; Przyporządkowanie zmiennej frame do aktualnego frame`
`DEF FRAME USTAWIENIE_4`	`; Definicja lokalnej zmiennej frame`
`USTAWIENIE_4 = $P_PFRAME`	`; Zapisanie pośrednie aktualnego frame w zmiennych frame`
`...`	
`$P_PFRAME = USTAWIENIE_4`	`; Odczyt zwrotny aktualnego frame ze zmiennych frame`

Powiązania

Powiązanie frame ze sobą następuje w zaprogramowanej kolejności przez operator : . Składowe frame jak np. przesunięcia, obroty itd., są wykonywane kolejno addytywnie.

Kod programu	Komentarz

`$P_IFRAME = $P_UIFR[15] : $P_UIFR[16]`	`; Przyporządkowanie frame wynikowego z powiązania`
	`; obydwu ustawianych frame zarządzania danymi do aktywnego`
	`; ustawianego frame łącznego.`
	`; Przykład zastosowania:`
	`; $P_UIFR[15]: Przesunięcie`
	`; $P_UIFR[16]: Obrót`

`$P_UIFR[3] = $P_UIFR[4] : $P_UIFR[5]`	`; Przyporządkowanie frame wynikowego z powiązania`
	`; obydwu ustawianych frame zarządzania danymi do`
	`; innego ustawianego frame zarządzania danymi`

6.2.4 Definicja zmiennych frame (DEF FRAME)

Oprócz predefiniowanych zmiennych frame można również definiować własne zmienne frame. Te definiowane samodzielnie zmienne frame są zmiennymi użytkownika typu FRAME. Nazwę frame można dowolnie nadać w ramach zasad dla zmiennych użytkownika.

Przy pomocy funkcji CTRANS, CROT, CSCALE i CMIRROR można samodzielnie definiowanym zmiennym frame przyporządkowywać wartości.

Składnia

```
DEF FRAME <nazwa>
```

Znaczenie

`DEF FRAME`:	Definiowanie zmiennych użytkownika typu FRAME.
<nazwa>:	Nazwa zmiennej frame

Przykład

Definicja zmiennej frame „PALETA” i przyporządkowanie wartości przesunięcia i obrotu:

Kod programu	Komentarz
`DEF FRAME PALETA`	`; Definicja zmiennej frame PALETA`
`PALETTE = CTRANS(...) : CROT(...)`	`; Przyporządkowanie frame wynikowego powiązania` `; przesunięcia i obrotu do zmiennej frame PALETA`

6.3 Przesunięcie zgrubne i dokładne (CTRANS, CFINE)

Przesunięcie dokładne

Przesunięcie dokładne `CFINE(...)` można stosować do następujących frame:

- Frame ustawiane: $P_UIFR wzgl. $P_IFRAME
- Frame bazowe: $P_NCBFR[<n>], $P_CHBFR[<n>] wzgl. $P_CHBFRAMES[<n>] albo $P_ACTBFRAME
- Frame programowany: $P_PFRAME

Przesunięcie dokładne frame jest programowane przy pomocy polecenia `CFINE(...)`.

Przesunięcie zgrubne

Przesunięcie zgrubne `CTRANS(...)` można stosować do wszystkich frame.

Przesunięcie całkowite

Przesunięcie całkowite jest sumą przesunięcia zgrubnego i dokładnego.

Dane maszynowe

Udostępnienie przesunięcia dokładnego

Przesunięcie dokładne jest udostępniane przy pomocy danej maszynowej:

MD18600 $MN_MM_FRAME_FINE_TRANS = 1

Składnia

Przesunięcie dokładne

- Frame łączny
  - `<frame> = CFINE(<K_1>,<wartość>)`
  - `<frame> = CFINE(<K_1>,<wartość>, <K_2>, <wartość>)`
  - `<frame> = CFINE(<K_1>,<wartość>, <K_2>, <wartość>, <K_3>, <wartość>)`
- Komponent Frame
  - `<frame>[<n>, <K_1>, FI] = <wartość>`

Przesunięcie zgrubne

- Frame łączny
  - `<frame> = CTRANS(<K_1>,<wartość>)`
  - `<frame> = CTRANS(<K_1>,<wartość>, <K_2,<wartość>)`
  - `<frame> = CTRANS(<K_1>,<wartość>, <K_2,<wartość>, <K_3,<wartość>)`
- Komponent Frame
  - `<frame>[<n>,<K_1>,TR] = <wartość>`

Specjalnie dla frame programowanego `$P_PFRAME`:

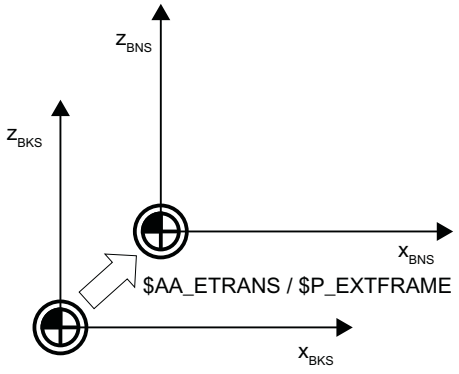
- `TRANS <K_1> <wartość>`
- `TRANS <K_1> <wartość> <K_2> <wartość>`
- `TRANS <K_1> <wartość> <K_2> <wartość> <K_3> <wartość>`

Znaczenie

`<Frame>`:	Frame np. Ustawiany frame zarządzania danymi `$P_UIFR[<n>]`
`CFINE`:	Przesunięcie dokładne, przesunięcie addytywne.
`CTRANS`:	Przesunięcie zgrubne, przesunięcie absolutne
`TRANS`:	Tylko frame programowany: Przesunięcie zgrubne, przesunięcie absolutne
`<K_n>`:	Osie współrzędnych X, Y, Z
`<wartość>`:	Wartość przesunięcia

6.4 Zewnętrzne przesunięcie punktu zerowego ($AA_ETRANS)

Zewnętrznym przesunięciem punktu zerowego jest przesunięcie liniowe między układem współrzędnych bazowym (BKS) i układem bazowego przesunięcia punktu zerowego (BNS).



Zewnętrzne przesunięcie punktu zerowego przy pomocy $AA_ETRANS działa, zależnie od sparametryzowania danych maszynowych, na dwa sposoby:

1. Zmienna systemowa $AA_ETRANS działa po uaktywnieniu przez sygnał interfejsu NC/PLC bezpośrednio jako wartość przesunięcia
2. Wartość zmiennej systemowej $AA_ETRANS jest po uaktywnieniu przez sygnał interfejsu NC/PLC przejmowana do aktywnych frame systemowych $P_EXTFRAME i obszaru danych frame $P_EXTFR. Następnie aktywne frame łączne $P_ACTFRAME jest obliczane na nowo.

Dane maszynowe

W związku ze zmienną systemową $AA_ETRANS należy rozróżnić dwa sposoby postępowania, które wybiera się poprzez następującą daną maszynową:

MD28082 $MC_MM_SYSTEM_FRAME_MASK,Bit1 = <wartość>

<wartość>	Znaczenie
0	Funkcja: Bezpośredni zapis $AA_ETRANS[<oś>] przez PLC, HMI albo program NC. Zezwolenie na realizację przesunięcia punktu zerowego od $AA_ETRANS[<oś>] w najbliższym możliwym bloku ruchu postępowego: DB31, ... DBX3.0
1	Funkcja: Uaktywnienie aktywnego frame systemowego $P_EXTFRAME i obszaru danych frame $P_EXTFR Zezwolenie na realizację przesunięcia punktu zerowego od $AA_ETRANS[<oś>] przez: DB31, ... DBX3.0 Potem następuje w kanale: • Zatrzymanie wszystkich ruchów postępowych w kanale (oprócz osi rozkazowych i osi PLC) • Zatrzymanie przebiegu wyprzedzającego z następną reorganizacją (STOPRE) • Przesunięcie zgrubne aktywnych frame $P_EXTFRAME[<oś>] = $AA_ETRANS[<oś>] • Przesunięcie zgrubne dla obszaru danych frame $P_EXTFR\[<oś>] = $AA_ETRANS\[<oś>] • Nowe obliczenie aktywnego frame łącznego $P_ACTFRAME • Realizacja przesunięcia w zaprogramowanych osiach. • Kontynuacja przerwanego ruchu postępowego wzgl. programu NC

Programowanie

- Składnia
  `$AA_ETRANS[<oś>] = <wartość>`
- Znaczenie

`$AA_ETRANS`:	Zmienna systemowa do zapisywania pośredniego zewnętrznego przesunięcia punktu zerowego
`<oś>`:	Oś kanałowa
`<wartość>`:	Wartość przesunięcia

Sygnał interfejsu NC/PLC

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<oś>] = $P_EXTFR[<oś>] = $AA_ETRANS[<oś>]

6.5 Ustawienie wartości rzeczywistej z utratą statusu bazowania (PRESETON)

Procedura PRESETON() ustawia dla jednej lub wielu osi nową wartość rzeczywistą w układzie współrzędnych maszyny (MKS). Odpowiada to przesunięciu punktu zerowego MKS osi. Oś X nie wykonuje wskutek tego ruchu.

PRESETON wyzwala zatrzymanie przebiegu wyprzedzającego z synchronizacją. Pozycja rzeczywista jest przyporządkowywana do osi dopiero w stanie zatrzymanym.

Jeżeli przy PRESETON oś nie jest przyporządkowana do kanału, dalsze postępowanie jest zależne od specyficznego dla osi zaprojektowania zachowania się przy zamianie osi:

MD30552 $MA_AUTO_GET_TYPE

Status bazowania

Przez ustawienie nowej wartości rzeczywistej w układzie współrzędnych maszyny status bazowania osi maszyny jest cofany:

DB31, ... DBX60.4 / .5 = 0 (bazuje / synchronizuje system pomiarowy 1 / 2)

Jest zalecane stosowanie PRESETON tylko w przypadku osi bez obowiązku bazowania.

W celu przywrócenia pierwotnego układu współrzędnych maszyny system pomiarowy osi maszyny musi zostać ponownie wybazowany, np. przez bazowanie z programu obróbki (G74).

⚠ OSTROŻNIE

Utrata statusu bazowania

Przez ustawienie nowej wartości rzeczywistej w układzie współrzędnych maszyny przy pomocy PRESETON status bazowania osi maszyny jest cofany na "nie bazowana / synchronizowana".

Programowanie

Składnia

```
PRESETON(<oś_1>, <wartość_1> [, <oś_2>, <wartość_2>, ... <oś_8>,
<wartość_8>])
```

Znaczenie

PRESETON:	Ustawienie wartości rzeczywistej z utratą statusu bazowania	
	Zatrzymanie przebiegu wyprzedzającego:	tak
	W oddzielnym bloku:	tak
<oś_x>:	Nazwa osi maszyny	
	Typ:	AXIS
	Zakres wartości:	Nazwy osi maszyny definiowane w kanale

<wartość_x>:	Nowa wartość rzeczywista osi maszyny w układzie współrzędnych maszyny (MKS) Wprowadzanie następuje w aktualnie obowiązującym systemie miar (calowy / metryczny) Aktywne programowanie po średnicy (DIAMON) jest uwzględniane	
	Typ:	REAL

Literatura

PRESETONS w programach NC

Szczegółowy opis PRESETON w programach NC znajduje się w:

Podręcznik działania Funkcje podstawowe; Rozdział "K2: Osie, układy współrzędnych, frame" > „Układy współrzędnych" > „Układ współrzędnych maszyny (MKS)" > "Ustawienie wartości rzeczywistej z utratą statusu bazowania (PRESETON)"

PRESETONS w akcjach synchronicznych

Szczegółowy opis PRESETON w akcjach synchronicznych znajduje się w:

Podręcznik działania Akcje synchroniczne; Rozdział: „Opis szczegółowy" > „Akcje w akcjach synchronicznych" > „Ustawienie wartości rzeczywistej bez utraty statusu bazowania (PRESETON)"

6.6 Ustawienie wartości rzeczywistej bez utraty statusu bazowania (PRESETONS)

Procedura PRESETONS() ustawia dla jednej lub wielu osi nową wartość rzeczywistą w układzie współrzędnych maszyny (MKS). Odpowiada to przesunięciu punktu zerowego MKS osi. Oś X nie wykonuje wskutek tego ruchu.

PRESETONS wyzwala zatrzymanie przebiegu wyprzedzającego z synchronizacją. Pozycja rzeczywista jest przyporządkowywana do osi dopiero w stanie zatrzymanym.

Jeżeli przy PRESETONS oś nie jest przyporządkowana do kanału, dalsze postępowanie jest zależne od specyficznego dla osi zaprojektowania zachowania się przy zamianie osi:

MD30552 $MA_AUTO_GET_TYPE

Status bazowania

Przez ustawienie nowej wartości rzeczywistej w układzie współrzędnych maszyny (MKS) przy pomocy PRESETONS status bazowania osi maszyny **nie** ulega zmianie.

Warunki

- **Typ przetwornika**
  PRESETONS jest możliwa tylko w przypadku następujących typów przetworników aktywnego systemu pomiarowego:
  - MD30240 $MA_ENC_TYPE[<system pomiarowy>] = 0 (przetwornik symulowany)
  - MD30240 $MA_ENC_TYPE[<system pomiarowy>] = 1 (przetwornik o sygnale surowym)
- **Tryb bazowania**
  PRESETONS jest możliwa tylko w przypadku następujących trybów bazowania aktywnego systemu pomiarowego:
  - MD34200 $MA_ENC_REFP_MODE[<system pomiarowy>] = 0 (bazowanie do punktu odniesienia niemożliwe)
  - MD34200 $MA_ENC_REFP_MODE[<system pomiarowy>] = 1 (bazowanie przyrostowych, obrotowych albo liniowych systemów pomiarowych: Impuls zerowy na ścieżce przetwornika)

Programowanie

Składnia

```
PRESETONS(<oś_1>, <wartość_1> [, <oś_2>, <wartość_2>, ... <oś_8>,
<wartość_8>])
```

Znaczenie

PRESETONS:	Ustawienie wartości rzeczywistej bez utraty statusu bazowania	
	Zatrzymanie przebiegu wyprzedzającego:	tak
	W oddzielnym bloku:	tak

<oś_x>:	Nazwa osi maszyny	
	Typ:	AXIS
	Zakres wartości:	Nazwy osi maszyny definiowane w kanale
<wartość_x>:	Nowa aktualna wartość rzeczywista osi maszyny w układzie współrzędnych maszyny (MKS) Wprowadzanie następuje w aktywnym systemie miar (calowy / metryczny) Aktywne programowanie po średnicy (DIAMON) jest uwzględniane	
	Typ:	REAL

Literatura

PRESETONS w programach NC

Szczegółowy opis PRESETONS w programach NC znajduje się w:

Podręcznik działania Funkcje podstawowe; Rozdział "K2: Osie, układy współrzędnych, frame" > „Układy współrzędnych" > „Układ współrzędnych maszyny (MKS)» > „Ustawienie wartości rzeczywistej bez utraty statusu bazowania (PRESETONS)"

PRESETONS w akcjach synchronicznych

Szczegółowy opis PRESETONS w akcjach synchronicznych znajduje się w:

Podręcznik działania Akcje synchroniczne; Rozdział: „Opis szczegółowy" > „Akcje w akcjach synchronicznych" > „Ustawienie wartości rzeczywistej bez utraty statusu bazowania (PRESETONS)"

6.7 Obliczenie frame z 3 punktów pomiarowych w przestrzeni (MEAFRAME)

Funkcja MEAFRAME jest stosowana do wspierania cykli pomiarowych. Oblicza ona Frame z trzech idealnych i korespondujących zmierzonych punktów.

Gdy obrabiany przedmiot jest pozycjonowany w celu obróbki, jego rzeczywista pozycja jest względem do kartezjańskiego układu współrzędnych maszyny (pozycja idealna) zazwyczaj zarówno przesunięta, jak też obrócona. W celu dokładnej obróbki albo pomiaru jest konieczne albo kosztowne ustawienie fizyczne albo zmiana ruchów w programie obróbki.

Frame można ustalić przez odczyt trzech punktów w przestrzeni, których idealne pozycje są znane. Odczyt następuje przy pomocy sondy dotykowej lub optycznej, która dotyka specjalnych otworów, precyzyjnie ustalonych na płycie nośnej, albo kulek pomiarowych.

Składnia

```
MEAFRAME(<punkty idealne>,<punkty pomiarowe>,<jakość>)
```

Znaczenie

MEAFRAME:	Wywołanie funkcji		
<punkty idealne>:	2-wym. Tablica danych REAL, zawierająca trzy współrzędne punktów idealnych		
<punkty pomiarowe>:	2-wym. Tablica danych REAL, zawierająca trzy współrzędne punktów zmierzonych		
<jakość>:	Zmienna, przy pomocy której są zwracane informacje dot. jakości obliczenia FRAME		
	Typ:	VAR REAL	
	Wartość:	-1	Punkty idealne leżą prawie na jednej prostej: Frame nie można było obliczyć. Zwracana zmienna FRAME zawiera neutralny Frame.
		-2	Punkty pomiarowe leżą prawie na jednej prostej: Frame nie można było obliczyć. Zwracana zmienna FRAME zawiera neutralny Frame.
		-4	Obliczenie matrycy obrotu nie udaje się z innego powodu.
		≥ 0.0	Suma zniekształceń (odstępy między punktami), które są potrzebne do zamiany zmierzonego trójkąta na trójkąt zgodny z trójkątem idealnym.

Uwaga

Jakość pomiaru

Aby współrzędne zmierzone mogły zostać przyporządkowane do współrzędnych idealnych przy pomocy kombinacji obrót/translacja, trójkąt wyznaczony przez zmierzone punkty musi być zgodny z trójkątem idealnym. Jest to realizowane przy pomocy algorytmu kompensacji, który minimalizuje sumę kwadratów odchyleń między trójkątem zmierzonym i idealnym, zmieniających trójkąt zmierzony na idealny.

Efektywnie zniekształcenie punktów pomiarowych może służyć, jako wskaźnik jakości pomiaru i jest dlatego wyprowadzane przez MEAFRAME, jako dodatkowa zmienna.

Uwaga

Frame utworzony przez MEAFRAME może poprzez funkcję ADDFRAME zostać przetransformowany na inny Frame w łańcuchu Frame (patrz przykład "Powiązanie przy pomocy ADDFRAME").

Przykłady

Przykład 1:

Program obróbki 1:

```
Kod programu
...
DEF FRAME CORR_FRAME
```

Ustawienie punktów pomiarowych:

```
Kod programu                                          Komentarz
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]=                             ; Dla testu.
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                            ; Pozwala na obrót pozycji przed-
                                                      miotu o max 5 stopni.
DEF REAL FIT_QUALITY_LIMIT=3                          ; Pozwala na przesunięcie o max 3
                                                      mm między
                                                      trójkątem idealnym i trójkątem
                                                      zmierzonym.
DEF REAL SHOW_MCS_POS1[3]
DEF REAL SHOW_MCS_POS2[3]
DEF REAL SHOW_MCS_POS3[3]
```

Kod programu	Komentarz
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	; Ograniczenie 1. kąta RPY.
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; Ograniczenie 2. kąta RPY.
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; Ograniczenie 3. kąta RPY.
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 $P_IFRAME=CORR_FRAME	; Uaktywnienie Frame próbkowania z Frame ustawianym. ; Sprawdzenie Frame przez pozycjonowanie osi geometrycznych na punkty idealne.
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=$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]	

Kod programu	Komentarz
N700 G500	; Wyłączenie aktywności frame ustawianego, ponieważ z frame zerowym (ustawienie wstępne braku wpisu wartości).
No_FRAME	; Wyłączenie Frame ustawianego, ponieważ jest wstępnie ustawiony Frame zerowy (wartość nie jest wpisana).
M0	
M30	

Przykład 2: Powiązanie frame

Powiązanie MEAFRAME dla korekcji

Funkcja MEAFRAME daje Frame korekcyjny. Jeżeli ten Frame zostanie powiązany z nastawnym Frame $P_UIFR[1], który był aktywny przy wywołaniu funkcji, (np. G54), wówczas uzyskuje się nastawny Frame dla dalszych przeliczeń w celu wykonywania ruchów albo obróbki.

Powiązanie przy pomocy ADDFRAME

Jeżeli ten Frame korekcyjny ma w łańcuchu Frame działać w innym miejscu albo gdy przed ustawianym Frame są aktywne jeszcze inne Frame, wówczas funkcja ADDFRAME( ) może zostać użyta do powiązania w jednym z Frame bazowych kanału albo we Frame systemowym.

We frame nie może być przy tym aktywne:

- lustrzane odbicie przy pomocy MIRROR
- skalowanie przy pomocy SCALE

Parametrami wejściowymi dla wartości zadanych i rzeczywistych są współrzędne obrabianego przedmiotu. W systemie podstawowym sterowania współrzędne te należy stale podawać metrycznie albo calowo (G71/G70) i jako wymiar odniesiony do promienia (DIAMOF).

Literatura:
Dalsze informacje dot. ADDFRAME patrz:
Podręcznik działania Funkcje podstawowe; K2: Osie, układy współrzędnych, frame

6.8 Frame globalne dla NCU

Frame globalne dla NCU występują w NCU tylko jeden raz dla wszystkich kanałów. Frame globalne dla NCU mogą być czytane i zapisywane ze wszystkich kanałów. Uaktywnienie frame globalnych dla NCU następuje w każdorazowym kanale.

Przez frame globalne **osie kanału i osie maszyny** z przesunięciami mogą być skalowane i poddawane lustrzanemu odbiciu.

Zależności geometryczne i łańcuchy frame

W przypadku frame globalnych nie istnieje zależność geometryczna między osiami. Dlatego nie mogą być wykonywane żadne obroty, ani programowanie identyfikatorów osi geometrycznych.

- Do frame globalnych nie dają się użyć żadne rotacje. Zaprogramowanie obrotu jest odrzucane z alarmem: "18310 Kanał %1 blok %2 Frame: obrót niedopuszczalny".
- Powiązanie frame globalnych i frame kanałowych jest możliwe. Frame wynikowy zawiera wszystkie składowe frame łącznie z rotacją dla wszystkich osi. Przyporządkowanie Frame ze składowymi obrotowymi do Frame globalnego jest odrzucane z alarmem "Frame: obrót niedopuszczalny".

Frame globalne dla NCU

Globalne dla NCU frame bazowe $P_NCBFR[n]

Można zaprojektować do 8 frame bazowych globalnych dla NCU:

Równocześnie mogą występować kanałowe frame bazowe.

Frame globalne mogą być czytane i zapisywane ze wszystkich kanałów NCU. Przy zapisie frame globalnych użytkownik powinien zatroszczyć się o koordynację kanałów. Można to realizować np. przez znaczniki Wait (WAITMC).

Producent maszyny

Liczba globalnych Frame bazowych jest projektowana poprzez dane maszynowe.

Literatura:
Podręcznik działania Funkcje podstawowe; Osie, Układy współrzędnych, frame (K2)

Globalne dla NCU Frame ustawiane $P_UIFR[n]

Wszystkie Frame ustawiane G500, G54...G599 mogą być projektowane jako globalne dla NCU, lub jako specyficzne dla kanału.

Producent maszyny

Wszystkie Frame ustawiane mogą przy pomocy danej maszynowej MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES zostać przeprojektowane na Frame globalne.

Jako identyfikatory osi w przypadku poleceń programowania frame mogą być stosowane identyfikatory kanału i identyfikatory osi maszyny. Programowanie identyfikatorów osi geometrycznych jest odrzucane z alarmem.

6.8.1 Frame kanałowe ($P_CHBFR, $P_UBFR)

Frame ustawiane albo Frame bazowe mogą poprzez program obróbki i poprzez BTSS być zapisywane i czytane przez obsługę i przez PLC.

Przesunięcie dokładne jest możliwe również dla frame globalnych. Wyłączanie Frame globalnych następuje tak, jak w przypadku Frame specyficznych dla kanału poprzez G53, G153, SUPA i G500.

Producent maszyny

Przez daną maszynową MD28081 $MC_MM_NUM_BASE_FRAMES można zaprojektować liczbę Frame bazowych w kanale. Konfiguracja standardowa jest tak zaprojektowana, że jest co najmniej jeden frame bazowy na kanał. Maksymalnie jest możliwych 8 frame bazowych na kanał. Dodatkowo do 8 frame bazowych w kanale może być jeszcze 8 frame globalnych NCU.

Frame specyficzne dla kanału

$P_CHBFR[n]

Poprzez zmienną systemową $P_CHBFR[n] mogą być czytane i zapisywane Frame bazowe. Przy zapisaniu Frame bazowego powiązany łączny Frame bazowy nie jest uaktywniany, lecz uaktywnienie następuje dopiero przy wykonaniu instrukcji G500, G54...G599. Zmienna służy przeważnie jako pamięć dla procesów zapisu na frame bazowym z HMI lub PLC. Te zmienne frame są zabezpieczane przez wykonanie kopii zapasowej danych.

Pierwszy frame bazowy w kanale

Zapisanie na predefiniowaną zmienną $P_UBFR uaktywnia Frame bazowy z indeksem tablicy 0 nie równocześnie, lecz uaktywnienie następuje dopiero z wykonaniem instrukcji G500, G54 ... G599. Zmienna może być zapisywana i czytana również w programie.

$P_UBFR

$P_UBFR jest identyczna z $P_CHBFR[0]. Standardowo jest zawsze jeden frame bazowy w kanale, tak że zmienna systemowa jest kompatybilna ze starszymi wersjami. Jeżeli nie ma specyficznego dla kanału frame bazowego, jest przy zapisie albo odczycie wyprowadzany alarm "Frame: Instrukcja niedopuszczalna".

6.8.2 Frame działające w kanale

Frame działające w kanale są wprowadzane z programu obróbki przez odnośne zmienne systemowe tych frame. Należą do tego również frame systemowe. Przez te zmienne systemowe można w programie obróbki czytać i zapisywać aktualny frame systemowy.

Aktualne frame działające w kanale

Przegląd

Aktualne frame systemowe	dla:
$P_PARTFRAME	TCARR i PAROT

$P_SETFRAME	Ustawienie wartości rzeczywistej i zadrapanie
$P_EXTFRAME	Zewnętrzne przesunięcie punktu zerowego
$P_NCBFRAME[n]	Aktualne frame bazowe globalne dla NCU
$P_CHBFRAME[n]	Aktualne kanałowe frame bazowe
$P_BFRAME	Aktualny 1. frame bazowy w kanale
$P_ACTBFRAME	Całkowity frame bazowy
$P_CHBFRMASK i $P_NCBFRMASK	Całkowity frame bazowy
$P_IFRAME	Aktualny frame ustawiany
Aktualne frame systemowe	dla:
$P_TOOLFRAME	TOROT i TOFRAME
$P_WPFRAME	Punkty odniesienia narzędzia
$P_TRAFRAME	Transformacje
$P_PFRAME	Aktualny programowany frame.
Aktualny frame systemowy	dla:
$P_CYCFRAME	Cykle
P_ACTFRAME	Aktualny frame całkowity
Powiązanie FRAME	Frame aktualny składa się z całkowitego frame bazowego

$P_NCBFRAME[n] Aktualne Frame bazowe globalne dla NCU

Poprzez zmienną systemową $P_NCBFRAME[n] mogą być czytane i zapisywane aktualne globalne elementy tablicy Frame bazowego. Wynikający całkowity frame bazowy jest wliczany przez proces zapisu w kanale.

Zmieniony frame będzie aktywny tylko w kanale, w którym został zaprogramowany. Jeżeli frame ma zostać zmieniony dla wszystkich kanałów NCU, konieczne jest równoczesne zapisanie $P_NCBFR[n] i $P_NCBFRAME[n]. Inne kanały muszą wówczas jeszcze uaktywnić Frame przy pomocy np. G54. Przy zapisie frame bazowego całkowity frame bazowy jest obliczany na nowo.

$P_CHBFRAME[n] Aktualne frame bazowe kanału

Poprzez zmienną systemową $P_CHBFRAME[n] mogą być czytane i zapisywane aktualne elementy tablicy Frame bazowego kanału. Wynikający całkowity frame bazowy jest wliczany przez proces zapisu w kanale. Przy zapisie frame bazowego całkowity frame bazowy jest obliczany na nowo.

$P_BFRAME aktualny 1. frame bazowy w kanale

Poprzez predefiniowaną zmienną Frame $P_BFRAME aktualny Frame bazowy z indeksem tablicy 0, który obowiązuje w kanale, może być czytany i zapisywany w programie obróbki. Zapisany frame bazowy jest natychmiast wliczany.

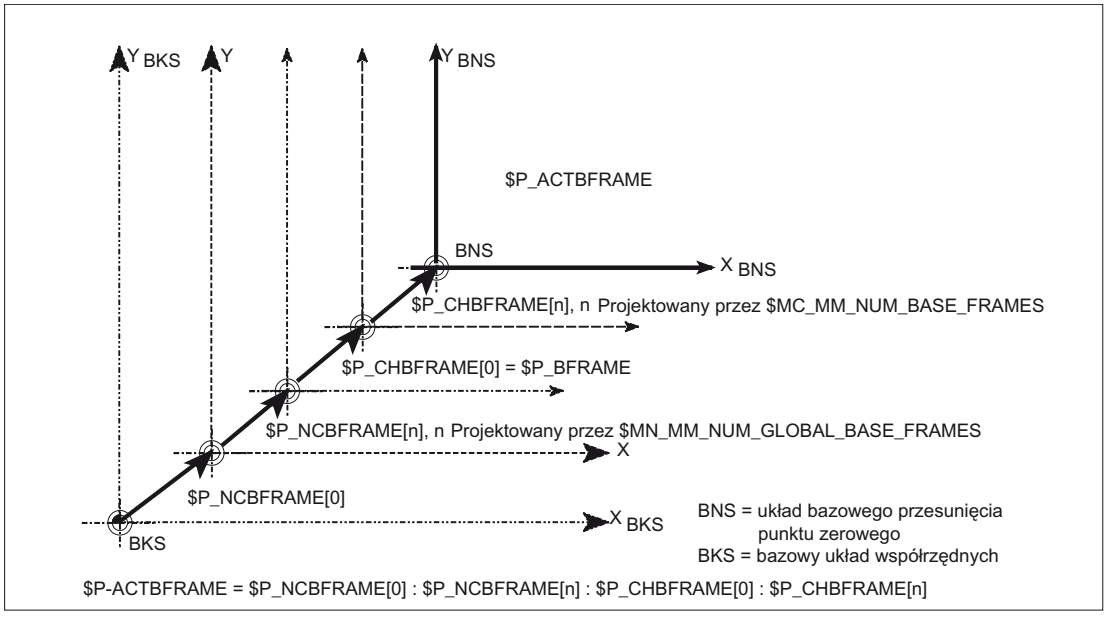
$P_BFRAME jest identyczna z $P_CHBFRAME[0]. Zmienna systemowa ma standardowo zawsze poprawną wartość. Jeżeli nie ma specyficznego dla kanału frame bazowego, jest przy zapisie albo odczycie wyprowadzany alarm "Frame: Instrukcja niedopuszczalna".

$P_ACTBFRAME Całkowity frame bazowy

Zmienna $P_ACTFRAME oblicza powiązany całkowity Frame bazowy. Zmienna może być tylko czytana.

$P_ACTFRAME odpowiada:

$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n].



$P_CHBFRMASK i $P_NCBFRMASK Całkowity frame bazowy

Poprzez zmienne systemowe $P_CHBFRMASK i $P_NCBFRMASK użytkownik może wybrać, jakie Frame bazowe chce włączyć do obliczenia "całkowitego" Frame bazowego. Zmienne mogą być programowane tylko w programie i czytane przez BTSS. Wartość zmiennych jest interpretowana jako maska bitowa i podaje, który element tablicy Frame bazowego $P_ACTBFAME jest brany do obliczenia.

Przy pomocy $P_CHBFRMASK można zadać, które specyficzne dla kanału Frame bazowe, a przy pomocy $P_NCBFRMASK, które globalne dla NCU Frame bazowe są wliczane.

Przez zaprogramowanie zmiennych całkowity Frame bazowy i Frame całkowity jest obliczany na nowo. Po Reset i w ustawieniu podstawowym wartość $P_CHBFRMASK i $P_NCBFRMASK jest następująca:

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

Przykład:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

$P_IFRAME Aktualny ustawiany frame

Poprzez predefiniowaną zmienną Frame $P_IFRAME aktualny ustawiany Frame, który obowiązuje w kanale, może być czytany i zapisywany w programie obróbki. Zapisany ustawiany frame jest wliczany natychmiast.

W przypadku globalnych dla NCU frame ustawianych zmieniony frame działa tylko w tym kanale, w którym jest zaprogramowany. Jeżeli Frame ma zostać zmieniony dla wszystkich kanałów NCU, musi równocześnie nastąpić opisanie $P_UIFR[n] i $P_IFRAME. Inne kanały muszą wówczas jeszcze uaktywnić odpowiedni Frame przy pomocy np. G54.

$P_PFRAME Aktualny frame programowany

$P_PFRAME jest to Frame programowany, który wynika z zaprogramowania TRANS/ ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR wzgl. z przyporządkowania CTRANS, CROT, CMIRROR, CSCALE do programowanego FRAME.

Aktualna, programowana zmienna frame, która tworzy odniesienie między układem ustawianego przesunięcia punktu zerowego (ENS) i układem współrzędnych obrabianego przedmiotu (WKS).

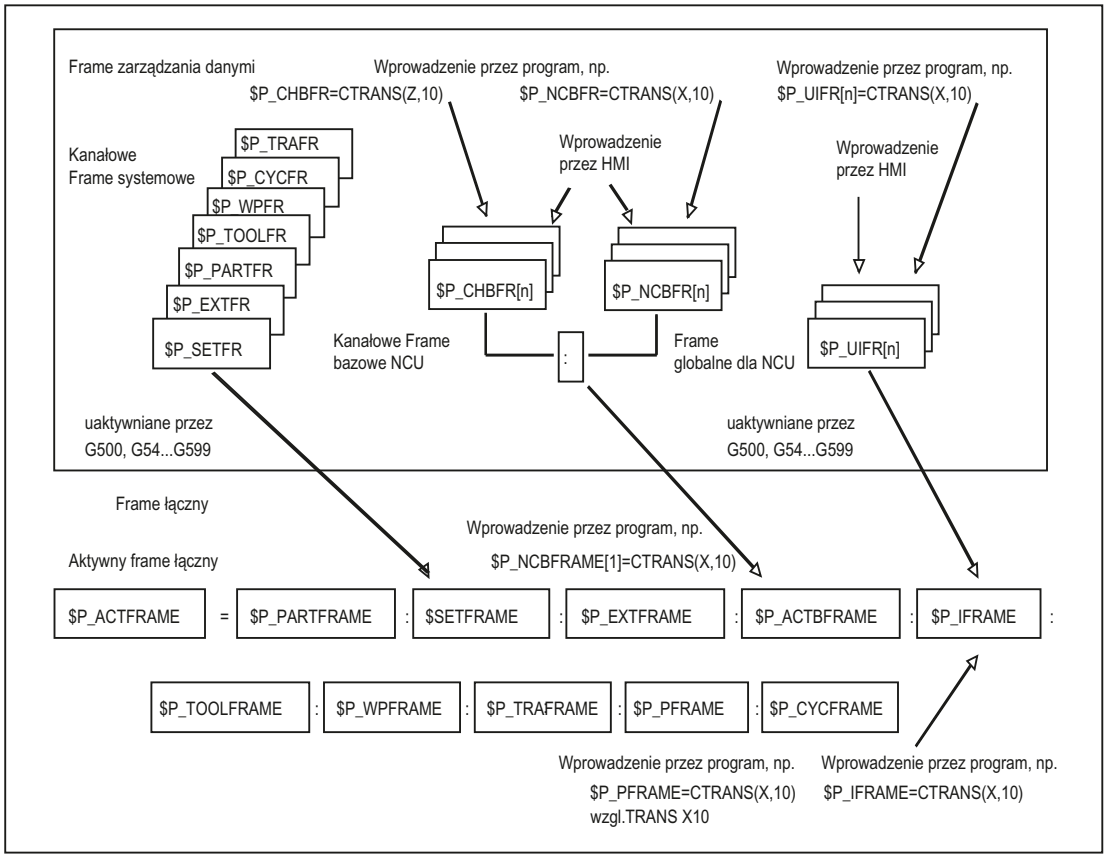
P_ACTFRAME Aktualny frame całkowity

Aktualny wynikający frame łączny $P_ACTFRAME wynika teraz jako powiązanie wszystkich frame bazowych, aktualnego frame ustawianego i frame programowanego. Aktualny frame jest uaktualniany zawsze wtedy, gdy zmienia się jego składowa.

`$P_ACTFRAME` odpowiada:

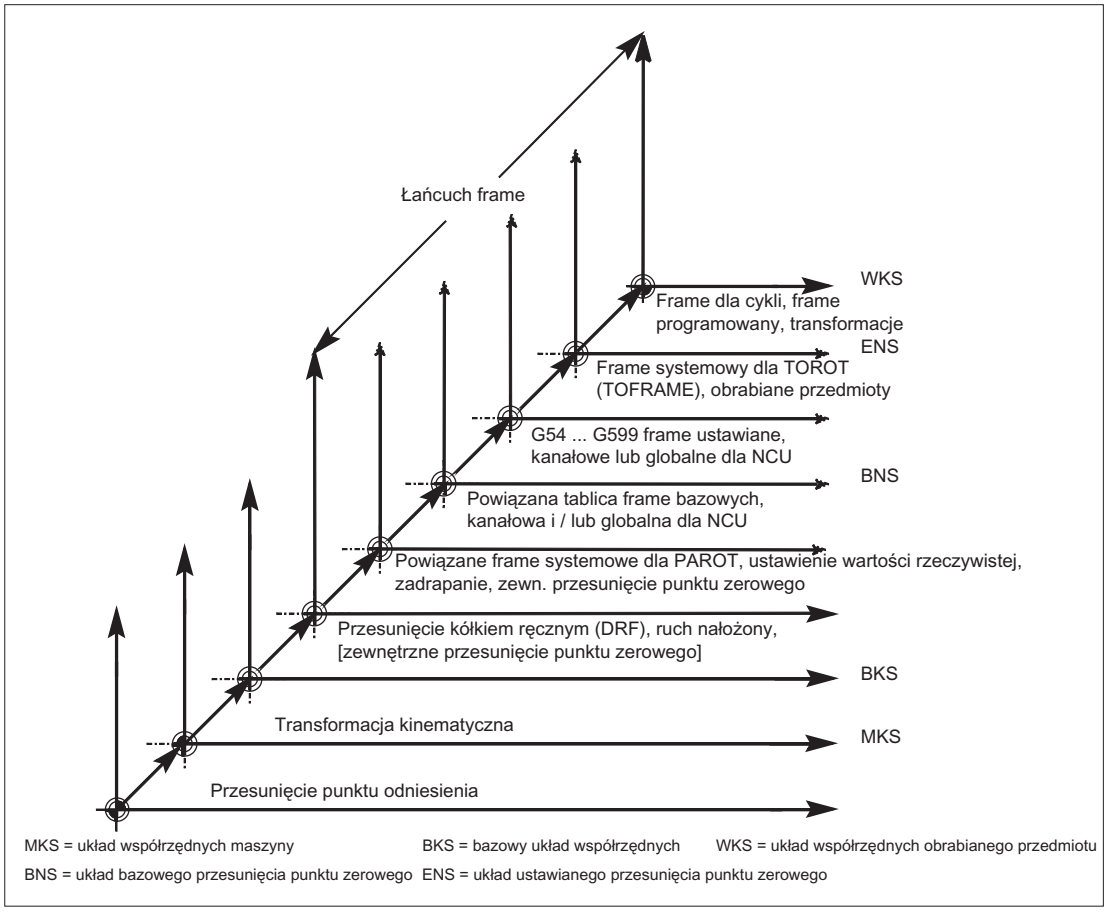
`$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :`

`$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME`


Frame zarządzania danymi
Wprowadzenie przez program, np.
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
Wprowadzenie przez program, np.
$P_UIFR[n]=CTRANS(X,10)
Kanałowe
Frame systemowe
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
Wprowadzenie
przez HMI
Wprowadzenie
przez HMI
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
Kanałowe Frame
bazowe NCU
:
Frame
globalne dla NCU
uaktywniane przez
G500, G54...G599
uaktywniane przez
G500, G54...G599
Frame łączny
Wprowadzenie przez program, np.
$P_NCBFRAME[1]=CTRANS(X,10)
Aktywny frame łączny
$P_ACTFRAME
=
$P_PARTFRAME
:
$SETFRAME
:
$P_EXTFRAME
:
$P_ACTBFRAME
:
$P_IFRAME
:
$P_TOOLFRAME
:
$P_WPFRAME
:
$P_TRAFRAME
:
$P_PFRAME
:
$P_CYCFRAME
Wprowadzenie przez program, np.
$P_PFRAME=CTRANS(X,10)
wzgl.TRANS X10
Wprowadzenie przez program, np.
$P_IFRAME=CTRANS(X,10)

Powiązanie frame

Aktualny frame składa się z całkowitego frame bazowego, frame ustawianego, frame systemowego i frame programowanego według wyżej podanego frame całkowitego.



Transformacje 7

7.1 Ogólne programowanie rodzajów transformacji

Funkcja ogólna

W celu dopasowywania sterowania do różnych kinematyk maszyny można wybrać programowanie rodzajów transformacji przy pomocy odpowiednich parametrów. Przez te parametry można dla wybranej transformacji odpowiednio uzgodnić zarówno orientację narzędzia w przestrzeni, jak też ruchy orientacji.

Przy transformacjach trzy-, cztero- i pięcioosiowych zaprogramowane dane dot. pozycji odnoszą się zawsze do wierzchołka narzędzia, który jest ortogonalnie aktualizowany do znajdującej się w przestrzeni powierzchni obróbki. Współrzędne kartezjańskie są przeliczane z bazowego układu współrzędnych na układ współrzędnych maszyny i odnoszą się do osi geometrycznych. Te opisują punkt roboczy. Wirtualne osie obrotowe opisują orientacje narzędzia w przestrzeni i są programowane przy pomocy TRAORI.

Przy transformacji kinematycznej pozycje mogą być programowane w kartezjańskim układzie współrzędnych. Sterowanie transformuje zaprogramowane przy pomocy TRANSMIT, TRACYL i TRAANG ruchy kartezjańskiego układu współrzędnych na ruchy realnych osi maszyny.

Programowanie

Transformacje trzy-, cztero- i pięcioosiowe TRAORI

Uzgodniona transformacja orientacji jest uaktywniana przy pomocy polecenia TRAORI i trzech możliwych parametrów dla numeru transformacji, wektora orientacji i offsetów osi obrotowych.

```
TRAORI(numer transformacji, wektor orientacji, offsety osi
obrotowych)
```

Transformacje kinematyczne

Do transformacji kinematycznych należą uzgodnione transformacje `TRANSMIT (numer transformacji)`

```
TRACYL(średnica robocza, numer transformacji)
```

```
TRAANG(kąt osi ustawionej skośnie, numer transformacji)
```

Wyłączenie aktywnej transformacji

Przy pomocy `TRAFOOF` można wyłączyć właśnie aktywną transformację.

Transformacja orientacji

Transformacje trzy-, cztero- i pięcioosiowe TRAORI

Do optymalnej obróbki powierzchni ukształtowanych przestrzennie w przestrzeni roboczej maszyny, obrabiarki potrzebują oprócz trzech osi liniowych X, Y i Z, jeszcze osi dodatkowych.

Dodatkowe osie opisują orientację w przestrzeni i są dalej nazywane osiami orientacji. Są one do dyspozycji jako osie obrotu w przypadku czterech typów maszyn z różną kinematyką.

1. Dwuosiowa głowica skrętna, np. głowica narzędziowa Kardana z osią obrotową równolegle do osi liniowej przy nieruchomym stole narzędziowym.
2. Dwuosiowy stół obrotowy, np. nieruchoma głowica skrętna ze stołem narzędziowym obracanym wokół dwóch osi.
3. Jednoosiowa głowica skrętna i jednoosiowy stół obrotowy, np. obrotowa głowica skrętna z narzędziem obróconym przy stole narzędziowym obracanym wokół jednej osi.
4. Dwuosiowa głowica skrętna i jednoosiowy stół obrotowy, np. przy stole narzędziowym obracanym wokół jednej osi i obrotowa głowica skrętna z narzędziem obracanym wokół własnej osi.

Transformacje **3- i 4-osiowe są specjalnymi** specjalnymi formami transformacji 5-osiowej i są programowane analogicznie do transformacji 5-osiowych.

"Transformacja rodzajowa 3-/4-/5-/6-osiowa pokrywa swoim zakresem funkcji prostopadle umieszczone osie obrotowe jak też transformacje dla głowicy frezarskiej Kardana i może jak każda inna transformacja orientacji być uaktywniana przy pomocy TRAORI również dla tych czterech typów maszyn. W przypadku rodzajowej transformacji 5/6 osiowej orientacja narzędzia ma kolejny trzeci stopień swobody, przy którym oprócz dowolnego kierunku narzędzia w przestrzeni narzędzie może być obracane wokół własnej osi.

Literatura: /FB3/ Podręcznik działania Funkcje specjalne; Transformacja 3- do 5-osiowej (F2)

Niezależne od kinematyki położenie podstawowe orientacji narzędzia

ORIRESET

Jeżeli przy pomocy TRAORI jest aktywna transformacja orientacji, wówczas można przy pomocy ORIRESET podać położenia podstawowe do 3 osi orientacji z opcjonalnymi parametrami A, B, C. Przyporządkowanie kolejności zaprogramowanych parametrów do osi obrotowych następuje według kolejności osi orientacji ustalonej przez transformację. Zaprogramowanie ORIRESET(A, B, C) powoduje, że osie orientacji wykonują ruch liniowy i synchroniczny od swojej aktualnej pozycji do podanej pozycji położenia podstawowego.

Transformacje kinematyczne

TRANSMIT i TRACYL

Przy obróbkach frezarskich na tokarkach można dla uzgodnionej transformacji programować albo

1. obróbkę czołową w zamocowaniu obrotowym przy pomocy TRANSMIT albo
2. obróbkę dowolnie przebiegających rowków na elementach walcowych przy pomocy TRACYL

.

TRAANG

Jeżeli oś dosuwu, np. dla technologii szlifowania, ma być dosuwana również skośnie, wówczas można przy pomocy TRAANG dla uzgodnionej transformacji zaprogramować parametryzowalny kąt.

Ruch kartezjański PTP

Do transformacji kinematycznej należy również "ruch kartezjański PTP" przy którym można programować do 8 różnych położeń przegubu STAT=. Pozycje są programowane w kartezjańskim układzie współrzędnych, przy czym ruch maszyny następuje we współrzędnych maszyny.

Literatura:
/FB2/ Podręcznik działania Funkcje rozszerzające; Transformacja kinematyczna (M1)

Transformacje powiązane

Każdorazowo można połączyć dwie transformacje jedna za drugą. W przypadku powiązanej w ten sposób drugiej transformacji składowe ruchu osi są przejmowane z pierwszej transformacji.

Jako pierwsza transformacja są możliwe:

- Transformacja orientacji TRAORI
- Transformacja biegunowa TRANSMIT
- Transformacja walcowa TRACYL
- Transformacja osi skośnej TRAANG

Druga transformacja musi być skośną osią TRAANG

7.1.1 Ruchy orientacji przy transformacjach

Ruchy postępowe i ruchy orientacji

Ruchy programowanych orientacji zależą pierwotnie od typu maszyny. Przy transformacji trzy-, cztero- i pięcioosiowej z TRAORI osie obrotowe albo osie skrętne osie, osie liniowe opisują ruchy orientacji narzędzia.

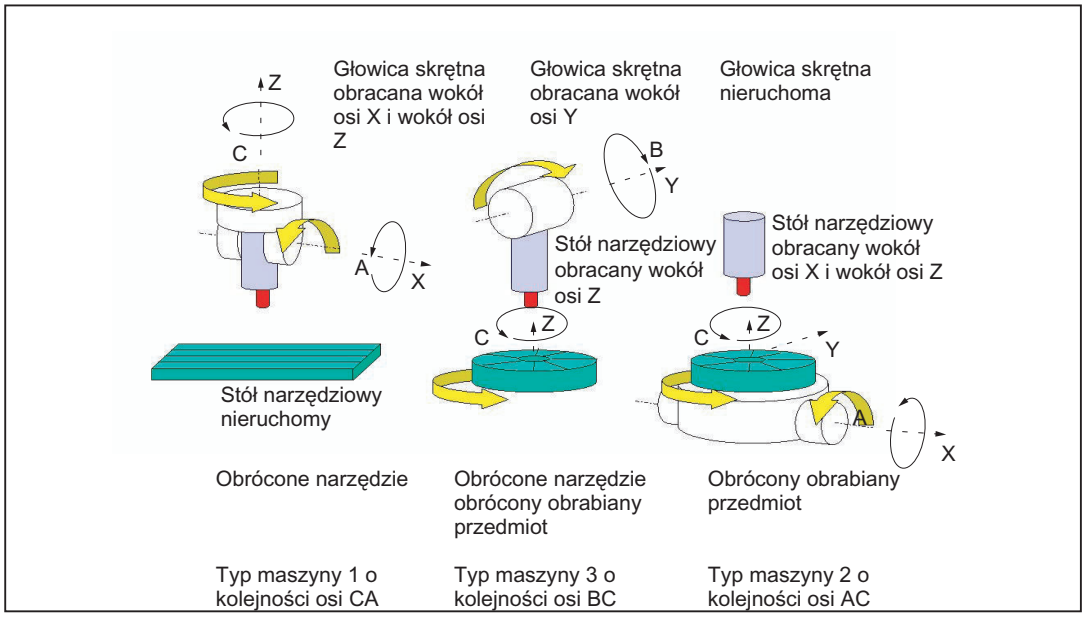
Zmiany pozycji osi obrotowych uczestniczących w transformacji orientacji prowadzą do ruchów wyrównawczych pozostałych osi maszyny. Położenie wierzchołka narzędzia pozostaje przy tym niezmienione.

Ruchy orientacji narzędzia mogą być programowane przez identyfikatory osi obrotowych A..., B..., C..., osi wirtualnych zależnie od zastosowania albo przez podanie kątów Eulera wzgl. kątów RPY albo wektorów kierunku wzgl. wektorów normalnych do powierzchni, wektorów unormowanych dla osi obrotu stożka albo dla orientacji pośredniej na pobocznicy stożka.

Przy transformacji kinematycznej z TRANSMIT, TRACYL i TRAANG sterowanie transformuje zaprogramowane ruchy kartezjańskiego układu współrzędnych na ruchy realnych osi maszyny.

Kinematyka maszyny przy transformacji trzy-, cztero- i pięcioosiowej

Obrotowym może być albo narzędzie albo stół narzędziowy z maksymalnie dwoma osiami obrotowymi. Kombinacja każdorazowo jednej osi głowicy skrętnej i stołu obrotowego jest również możliwa.



Typ maszyny	Programowanie orientacji
Transformacja trzyosiowa typy maszyn 1 i 2	Programowanie orientacji narzędzia tylko w płaszczyźnie, która jest prostopadła do osi obrotowej. Istnieją **dwie** osie ruchu postępowego (liniowe) i **jedna** oś rotacyjna (obrotowa).
Transformacja czteroosiowa typy maszyn 1 i 2	Programowanie orientacji narzędzia tylko w płaszczyźnie, która jest prostopadła do osi obrotowej. Istnieją **trzy** osie ruchu postępowego (liniowe) i **jedna** oś rotacyjna (obrotowa).
Transformacja pięcioosiowa typ maszyny 3 jednoosiowa głowica skrętna i jednoosiowy stół obrotowy	Programowanie transformacji orientacji. Kinematyka z **trzema** osiami liniowymi i **dwoma** ortogonalnymi osiami obrotowymi. Osie obrotowe są równoległe do dwóch z trzech osi liniowych. Pierwsza oś obrotowa jest poruszana od dwóch kartezjańskich osi liniowych. Obraca ona trzecią osią liniową z narzędziem. Druga oś obrotowa obraca obrabianym przedmiotem.

Transformacje rodzajowe 5/6-osiowe

Typ maszyny	Programowanie transformacji orientacji
Transformacja rodzajowa pięcio-/sześcioosiowa typy maszyn 4 Dwuosiowa głowica skrętna z obrotowym narzędziem i jednoosiowy stół obrotowy	Programowanie transformacji orientacji. Kinematyka z **trzema** osiami liniowymi i **trzema** ortogonalnymi osiami obrotowymi. Osie obrotowe są równoległe do dwóch z trzech osi liniowych. Pierwsza oś obrotowa jest poruszana od dwóch kartezjańskich osi liniowych. Obraca ona trzecią osią liniową z narzędziem. Druga oś obrotowa obraca obrabianym przedmiotem. Podstawowa orientacja narzędzia może zostać zaprogramowana przez dodatkowy obrót wokół niego o kąt THETA.

Przy wywołaniu "transformacji rodzajowej trzy-, cztero- i pięcio-/sześcioosiowej" można dodatkowo przekazać orientację podstawową narzędzia. Ograniczenia odnośnie kierunków osi obrotowych już nie obowiązują. Jeżeli osie obrotowe nie są dokładnie prostopadłe do siebie lub występujące osie obrotowe nie są ustawione dokładnie równolegle do osi liniowych, "rodzajowa transformacja pięcio-/sześcioosiowa" może dać lepsze wyniki orientacji narzędzia.

Transformacje kinematyczne TRANSMIT, TRACYL i TRAANG

Dla obróbek frezarskich na tokarkach albo skośnie dosuwanej osi przy szlifowaniu obowiązują zależnie od transformacji w przypadku standardowym następujące układy osi:

TRANSMIT	Uaktywnienie transformacji biegunowej
Obróbka czołowa w zamocowaniu tokarskim	jedna oś obrotowa jedna oś dosuwu prostopadła do osi toczenia jedna oś podłużna równoległa do osi toczenia

TRACYL	Uaktywnienie transformacji pobocznicy walca
Obróbka dowolnie przebiegających rowków na elementach cylindrycznych	jedna oś obrotowa jedna oś dosuwu prostopadła do osi toczenia jedna oś podłużna równoległa do osi toczenia

TRAANG	Uaktywnienie transformacji z osią skośną
Obróbka ze skośną osią dosuwu	jedna oś obrotowa jedna oś dosuwu z parametryzowanym kątem jedna oś podłużna równoległa do osi toczenia

Ruch kartezjański PTP

Ruch maszyny następuje we współrzędnych maszyny i jest programowany przy pomocy:

TRAORI	Uaktywnienie transformacji
PTP - ruch punkt do punktu	Ruch do pozycji w kartezjańskim układzie współrzędnych (MKS)
CP	Ruch po torze osi kartezjańskich w (BKS)
STAT	Położenie przegubów jest zależne od transformacji
TU	Kąt, o który osie wykonują ruch po najkrótszej drodze

Ruch PTP przy rodzajowej transformacji 5/6-osiowej

Ruch maszyny następuje we współrzędnych maszyny, a orientacja narzędzia może zostać zaprogramowana zarówno przy pomocy pozycji osi obrotowych, jak również przy pomocy niezależnych od kinematyki wektorów kąt Eulera wzgl. kąt RPY albo wektorów kierunku.

Jest przy tym możliwa interpolacja osi obrotowej, interpolacja wektora z interpolacją wielkiego okręgu lub interpolacja wektora orientacji na pobocznicy stożka.

Przykład, transformacja trzy- do pięcioosiowej w przypadku głowicy frezarskiej Kardana

Obrabiarka ma co najmniej 5 osi, z tego

- Trzy osie ruchu prostoliniowego, które przesuwają punkt roboczy do dowolnej pozycji w przestrzeni roboczej.
- Dwie obrotowe osie skrętne, które są umieszczone pod projektowanym kątem (najczęściej 45 stopni), umożliwiają narzędziu przyjmowanie orientacji w przestrzeni, które przy umieszczeniu pod kątem 45 stopni ograniczają się do półkuli.

7.1.2 Przegląd transformacji orientacji TRAORI

Możliwe rodzaje programowania w związku z TRAORI

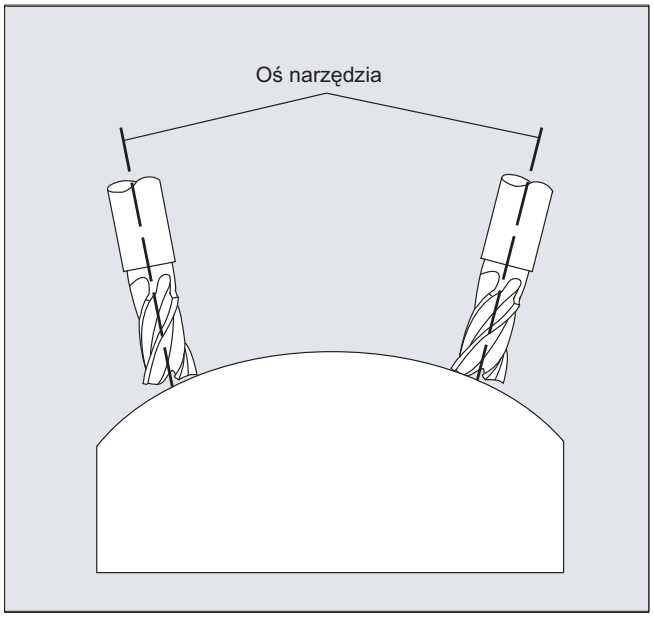
Typ maszyny	Programowanie przy aktywnej transformacji TRAORI
Typy maszyn 1, 2 albo 3, dwuosiowa głowica skrętna albo dwuosiowy stół obrotowy albo kombinacja jednoosiowej głowicy skrętnej i jednoosiowego stołu obrotowego.	Kolejność osi orientacji i kierunek orientacji narzędzia można projektować albo **w odniesieniu do maszyny** przez dane maszynowe zależnie od kinematyki maszyny albo **w odniesieniu do obrabianego przedmiotu** z programowalną orientacją niezależnie od kinematyki maszyny Kierunki obrotu osi orientacji w systemie odniesienia są programowane przy pomocy: - ORIMKS układ odniesienia = układ współrzędnych maszyny - ORIWKS układ odniesienia = układ współrzędnych obrabianego przedmiotu Położeniem podstawowym jest ORIWKS. Programowanie osi orientacji przy pomocy: A, B, C pozycji osi maszyny bezpośrednio A2, B2, C2 programowanie kąta osi wirtualnych przy pomocy - ORIEULER przez kąt Eulera (standard) - ORIRPY przez kąt RPY - ORIVIRT1 przez wirtualne osie orientacji 1. definicja - ORIVIRT2 przez wirtualne osie orientacji 2. definicja z rozróżnieniem rodzaju interpolacji: **interpolacja liniowa** - ORIAXES osi orientacji lub osi maszyny **interpolacja dużego okręgu** (interpolacja wektora orientacji) - ORIVECT osi orientacji Programowanie osi orientacji przez podanie A3, B3, C3 komponentów wektorowych (normalna kierunkowa/powierzchni) Programowanie wynikowej orientacji narzędzia A4, B4, C4 wektora normalnej powierzchni na początku bloku A5, B5, C5 wektora normalnej powierzchni na końcu bloku LEAD kąt wyprzedzenia dla orientacji narzędzia TILT kąt boczny dla orientacji narzędzia

Typ maszyny	Programowanie przy aktywnej transformacji TRAORI
	Interpolacja wektora orientacji na pobocznicy stożka Zmiany orientacji na znajdującej się **dowolnie w przestrzeni** pobocznicy stożka przez interpolację: - ORIPLANE w płaszczyźnie (interpolacja dużego okręgu) - ORICONCW na pobocznicy stożka w kierunku ruchu wskazówek zegara - ORICONCCW na pobocznicy stożka przeciwnie do ruchu wskazówek zegara A6, B6, C6 wektora kierunkowego (oś obrotu stożka) -OICONIO interpolacja na pobocznicy stożka z: A7, B7, C7 wektory pośrednie (orientacja początkowa i końcowa) albo - ORICONTO na pobocznicy stożka przejście styczne Zmiany orientacji odniesione **do toru** przy pomocy - ORICURVE zadanie ruchu dwóch punktów kontaktowych przez PO[XH]=(xe, x2, x3, x4, x5) wielomiany orientacji max 5. stopnia PO[YH]=(ye, y2, y3, y4, y5) wielomiany orientacji max 5. stopnia PO[ZH]=(ze, z2, z3, z4, z5) wielomiany orientacji max 5. stopnia - ORIPATHS wygładzanie przebiegu orientacji przy pomocy A8, B8, C8. Fazie przeorientowania narzędzia odpowiada: Kierunek i długość drogi narzędzia przy cofnięciu
Typy maszyn 1 i 3 Dalsze typy maszyn z dodatkowym obrotem narzędzia wokół siebie samego wymagają 3. osi obrotowej Transformacja orientacji, jak np. rodzajowa transformacja 6-osiowa. Obroty wektora orientacji.	Programowanie **obrotów** orientacji narzędzia przy pomocy LEAD kąt wyprzedzenia w stosunku do wektora normalnej powierzchni PO[PHI] programowanie wielomianu max 5. stopnia TILT kąt boczny, obrót wokół stycznej do toru (kierunek Z) PO[PSI] programowanie wielomianu max 5. stopnia THETA kąt obrotu (obrót wokół kierunku narzędzia w Z) THETA= wartość, która jest uzyskiwana na końcu bloku THETA=AC(...) przełączenie pojedynczymi blokami na absolutne podawanie wymiarów THETA=IC(...) przełączenie pojedynczymi blokami na przyrostowe podawanie wymiarów THETA=Θ_e Interpolowanie kąta programowanego G90/G91 PO[THT]=(..) Programowanie wielomianu max 5. stopnia Programowanie wektora obrotu - ORIROTA obrót absolutny - ORIROTR względny wektor obrotu - ORIROTT styczny wektor obrotu
Odniesiona do toru orientacja dla zmian orientacji w stosunku do toru albo obrót wektora obrotu stycznie do toru	Zmiany orientacji **w stosunku do toru** przy pomocy - ORIPATH orientacja narzędzia w odniesieniu do toru - ORIPATHS dodatkowo w przypadku załamania w przebiegu orientacji Programowanie wektora obrotu - ORIROTC styczny wektor obrotu, obrót do stycznej do toru

7.2 Transformacje trzy-, cztero- i pięcioosiowe (TRAORI)

7.2.1 Zależności ogólne głowicy narzędziowej Kardana

Aby uzyskać optymalne warunki skrawania przy obrabianiu powierzchni przestrzennie zakrzywionych, kąt ustawienia narzędzia musi się dawać zmieniać.

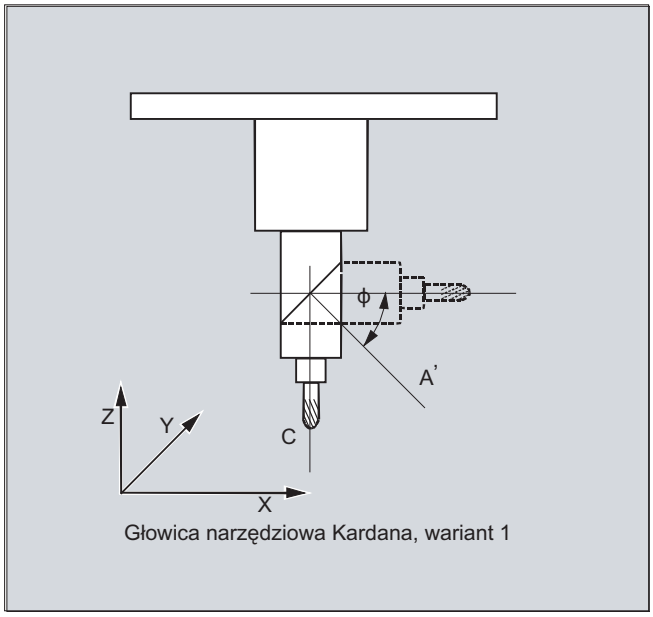


Przy pomocy jakiej konstrukcji maszyny jest to uzyskiwane, jest zapisane w danych osi.

Transformacja 5-osiowa

Głowica narzędziowa Kardana

Tutaj trzy osie liniowe (X, Y, Z) i dwie osie orientacji ustalają kąt ustawienia i punkt roboczy narzędzia. Jedna z obydwu osi orientacji jest przyłożona jako oś skośna, tutaj w przykładzie A' - w wielu przypadkach jako układ 45°.

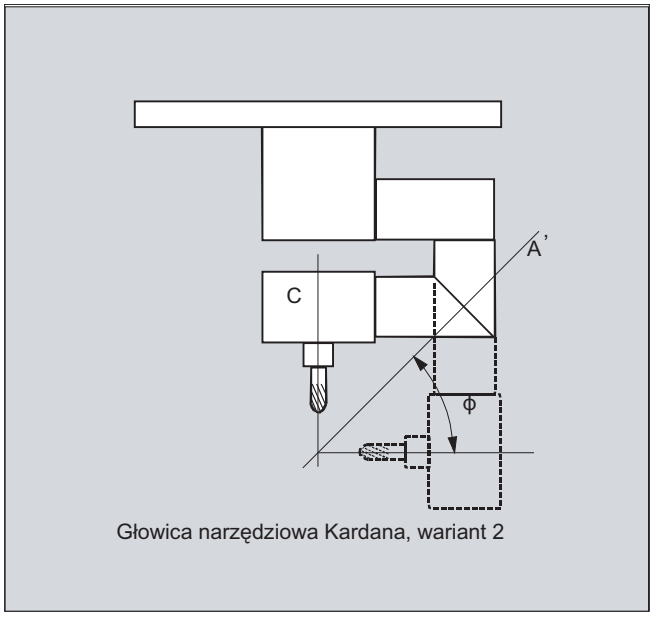


Głowica narzędziowa Kardana, wariant 1

W pokazanych tutaj przykładach widać układy na przykładzie z głowicą narzędziową Kardana kinematyki maszyny CA!

Producent maszyny

Kolejność osi orientacji i kierunek orientacji narzędzia są zależnie od kinematyki maszyny ustawiane przez dane maszynowe.



Głowica narzędziowa Kardana, wariant 2

W niniejszym przykładzie A' leży pod kątem φ względem osi X

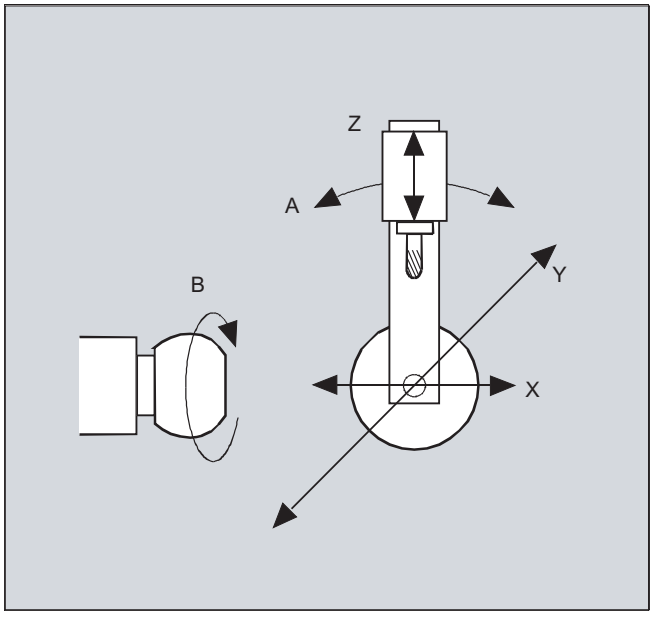
Ogólnie obowiązują następujące możliwe zależności:

A' leży pod kątem φ względem	osi X
B' leży pod kątem φ względem	osi Y
C' leży pod kątem φ względem	osi Z

Kąt φ można projektować w zakresie 0° do +89° przez dane maszynowe.

Ze skrętną osią liniową

Chodzi przy tym o układ z poruszanym obrabianym przedmiotem i poruszanym narzędziem. Kinematyka składa się z trzech osi liniowych (X, Y, Z) i dwóch prostopadle usytuowanych osi obrotowych. Pierwsza oś obrotowa jest poruszana np. przez sanie krzyżowe o dwóch osiach liniowych, narzędzie jest ustawione równolegle do trzeciej osi liniowej. Druga oś obrotu obraca obrabiany przedmiot. Trzecia oś liniowa (oś skrętna) leży w płaszczyźnie sani krzyżowych.



Kolejność osi rotacyjnych i kierunek orientacji narzędzia są zależnie od kinematyki maszyny ustawiane przez dane maszynowe.

Obowiązują następujące możliwe zależności:

Osie:	Kolejności osi:
1. Oś obrotowa	A A B B C C
2. Oś obrotowa	B C A C A B
Skręcona oś liniowa	Z Y Z X Y X

Dalsze objaśnienia dot. konfigurowalnych kolejności osi dla kierunku orientacji narzędzia patrz

Literatura: /FB3/ Podręcznik działania Funkcje specjalne; transformacja 3- do 5-osiowej (F2), punkt Głowica frezarska Kardana, "Parametryzacja".

7.2.2 Transformacje trzy-, cztero- i pięcioosiowe (TRAORI)

Użytkownik może projektować dwie wzgl. trzy osie ruchu postępowego i jedną oś obrotową. Transformacje zakładają, że oś obrotowa jest ustawiona ortogonalnie na płaszczyźnie orientacji.

Orientacja narzędzia jest możliwa tylko w płaszczyźnie, która jest prostopadła do osi obrotowej. Transformacja obsługuje typy maszyn o ruchomym narzędziu i ruchomym obrabianym przedmiocie.

Projektowanie i programowanie transformacji trzy- i czteroosiowych jest analogiczne do transformacji pięcioosiowych.

Literatura:
Podręcznik działania Funkcje specjalne; Transformacje wieloosiowe (F2)

Składnia

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
TRAFOOF
```

Znaczenie

`TRAORI`:	Uaktywnia pierwszą uzgodnioną transformację orientacji	
`TRAORI(<n>)`:	Uaktywnia transformację orientacji uzgodnioną przy pomocy n	
`<n>`:	Numer transformacji	
	Wartość:	1 albo 2
	Przykład: TRAORI(1) uaktywnia transformację orientacji 1	
`<X>,<Y>,<Z>`:	Składowa wektora orientacji, na którą wskazuje narzędzie.	
`<A>,<B>`:	Programowalne przesunięcie dla osi obrotowych	
`TRAFOOF`:	Wyłączenie transformacji	

Orientacja narzędzia

Zależnie od wybranego kierunku orientacji narzędzia aktywna płaszczyzna robocza (G17, G18, G19) musi być tak ustawiona w programie NC, by korekcja długości narzędzia działała w kierunku orientacji narzędzia.

Uwaga

Po włączeniu transformacji dane dot. pozycji odnoszą się zawsze do wierzchołka narzędzia (X, Y, Z). Zmiany pozycji osi obrotowych uczestniczących w transformacji prowadzą do ruchów wyrównawczych pozostałych osi maszyny, dzięki czemu pozycja wierzchołka narzędzia pozostaje nie zmieniona.

Transformacja orientacji jest zawsze skierowana od wierzchołka narzędzia do jego uchwytu.

Przesunięcie dla osi orientacji

Przy uaktywnieniu transformacji orientacji można bezpośrednio zaprogramować dodatkowe przesunięcie dla osi orientacji.

Wolno pominąć parametry, gdy przy programowaniu zostanie zachowana prawidłowa kolejność.

Przykład:

`TRAORI(, , , ,A,B)` ; gdy ma zostać wprowadzony tylko jedno przesunięcie

Alternatywnie bezpośredni do programowania można to dodatkowe przesunięcie dla osi orientacji również przejąć automatycznie z aktualnie aktywnego przesunięcia punktu zerowego. Przejęcie jest projektowane przez dane maszynowe.

Przykłady

```
TRAORI(1,0,0,1)          ; Podstawowa orientacja narzędzia wskazuje w kierunku Z
TRAORI(1,0,1,0)          ; Podstawowa orientacja narzędzia wskazuje w kierunku Z
TRAORI(1,0,1,1)          ; Podstawowa orientacja narzędzia wskazuje w kierunku Y/Z
                         (odpowiada położeniu -45°)
```

7.2.3 Warianty programowania orientacji i położenie podstawowe (OTIRESET)

Programowanie orientacji narzędzia przy TRAORI

W połączeniu z programowaną transformacją orientacji TRAORI mogą dodatkowo do osi liniowych X, Y, Z być przez identyfikatory osi obrotowych A..., B..., C... programowane pozycje osi albo kąty osi wirtualnych albo składowe wektora. Dla osi orientacji i osi maszyny są możliwe różne rodzaje interpolacji. Niezależnie od tego, jakie wielomiany orientacji PO[kąt] i wielomiany osi PO[oś] są właśnie aktywne, może być zaprogramowanych wiele różnych rodzajów wielomianów jak np. G1, G2, G3, CIP lub POLY.

Zmiana orientacji narzędzia może zostać zaprogramowana przez wektory orientacji. Przy tym orientacja końcowa każdego bloku może nastąpić albo przez programowanie bezpośrednie wektora albo przez programowanie pozycji osi obrotowych.

Warianty programowania orientacji przy transformacji trzy- do pięcioosiowej

Następujące warianty programowania orientacji wzajemnie się wyktuczają.

`A, B, C`	Bezpośrednie podanie pozycji osi obrotowych
`A2, B2, C2`	Programowanie kąta osi wirtualnych poprzez kąt Eulera albo kąt RPY
`A3, B3, C3`	Podanie składowych wektora
`LEAD I TILT`	Podanie kąta wyprzedzenia i kąta bocznego w odniesieniu do toru i powierzchni
`A4, B4, C4` `A5, B5, C5`	Wektor normalny do powierzchni na początku i na końcu bloku

A6, B6, C6 A7, B7, C7	Interpolacja wektora orientacji na pobocznicy stożka
A8, B8, C8	Zorientowanie narzędzia, kierunku i długości drogi dla ruchu wycofania

Najazd na położenie podstawowe orientacji narzędzia (ORIRESET)

Poprzez ORIRESET(...) osie orientacji odpowiedniej kinematyki maszyny będą wykonywały ruch liniowo i synchroniczne z ich aktualnej pozycji na zaprogramowaną pozycję podstawową. Jeżeli dla osi nie jest zaprogramowana żadna pozycja położenia podstawowego, wówczas jest stosowana zdefiniowana pozycja z przynależnej danej maszynowej $MC_TRAFO5_ROT_AX_OFFSET_1/2.

Aktywne frame osi obrotowych nie są przy tym uwzględniane.

Przykłady kinematyki maszyny CA (nazwy osi kanału C, A)

Polecenia	Opis
ORIRESET(90, 45)	Oś C: 90° Oś A: 45°
ORIRESET(, 30)	Oś C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] Oś A: 30°
ORIRESET( )	Oś C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] Oś A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

Przykłady dla kinematyki maszyny CAB (nazwy osi kanału C, A, B)

Polecenia	Opis
ORIRESET(90, 45, 90)	Oś C: 90° Oś A: 45° Oś B: 90°
ORIRESET( )	Oś C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] Oś A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1] Oś B: $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]

Uwaga

Najazd na położenie podstawowe orientacji narzędzia za pomocą ORIRESET(...) dozwolony jest wyłącznie, gdy transformacja orientacji TRAORI(...) jest aktywna.

Programowanie obrotów LEAD, TILT i THETA

Kąt wyprzedzenia LEAD i kąt boczny TILT

Obroty orientacji narzędzia w przypadku transformacji trzy- do pięcioosiowej są programowane przy pomocy kątów wyprzedzenia LEAD i kątów bocznych TILT.

Kąt obrotu THETA

W przypadku transformacji **z trzecią osią obrotową** , zarówno dla orientacji z składowymi wektora, jak również z programowaniem kąta LEAD, TILT, może zostać zaprogramowany obrót narzędzia wokół swojej osi z kątem obrotu THETA.

7.2.4 Programowanie orientacji narzędzia (A..., B..., C..., LEAD, TILT)

Do programowania orientacji narzędzia dostępne są następujące opcje:

1. Bezpośrednie programowanie ruchu osi obrotowych. Zmiana orientacji następuje zawsze w bazowym układzie współrzędnych lub układzie współrzędnych maszyny. Osie orientacji wykonują zawsze ruch, jako osie synchroniczne.
2. Programowanie w kątach Eulera albo RPY według definicji kąta przez `A2, B2, C2`.
3. Programowanie wektora kierunku przez `A3, B3, C3`. Wektor kierunku jest skierowany od wierzchołka narzędzia w kierunku uchwytu.
4. Programowanie wektora normalnego do powierzchni na początku bloku za pomocą `A4, B4, C4` i na końcu bloku za pomocą `A5, B5, C5` (frezowanie czołowe).
5. Programowanie przez kąt wyprzedzenia `LEAD` i kąt boczny `TILT`
6. Programowanie osi obrotu stożka jako wektor znormalizowany przez `A6, B6, C6`, albo orientacji pośredniej na pobocznicy stożka przez `A7, B7, C7`,
   patrz punkt "Programowanie orientacji wzdłuż pobocznicy stożka (ORIPLANE, ORICONxx)".
7. Programowanie zorientowania, kierunku i długości drogi narzędzia podczas ruchu odsunięcia przez `A8, B8, C8`,
   patrz punkt "Wygładzanie przebiegu orientacji" (ORIPATHS A8=, B8=, C8=)"

Uwaga

We wszystkich przypadkach programowanie orientacji jest dopuszczalne tylko wtedy, gdy transformacja orientacji jest włączona.

Zaleta: Te programy dają się przenosić dla dowolnej kinematyki maszyny.

Definicja orientacji narzędzia przez polecenie G

Uwaga

Producent maszyny

Za pomocą danej maszynowej można przełączać między kątami Eulera i kątami RPY. Przy odpowiednich ustawieniach danych maszynowych jest możliwe przełączanie zarówno zależnie, jak i niezależnie od aktywnego polecenia grupy G nr 50. Dostępne są następujące opcje ustawień:

1. Gdy obie dane maszynowe do definicji osi orientacji i definicji kąta orientacji przez polecenie G są ustawione na zero:
   Kąty zaprogramowane za pomocą `A2, B2, C2` są interpretowane, albo jako kąty Eulera albo kąty RPY **w zależności od danej maszynowej** do definicji kąta programowania orientacji.
2. Gdy dana maszynowa do definicji osi orientacji przez polecenie G jest ustawiona na jeden, przełączenie następuje
   zależnie od aktywnego polecenia grupy G nr 50:
   Kąty zaprogramowane za pomocą `A2, B2, C2` są interpretowane zgodnie z jednym z aktywnych poleceń G `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` i `ORIPY2`. Wartości zaprogramowane za pomocą osi orientacji są również interpretowane jako kąt orientacji zgodnie z aktywnym poleceniem grupy G nr 50.
3. Gdy dana maszynowa dla definicji kątów orientacji przez polecenie G jest ustawiona na jeden, a dana maszynowa dla definicji osi orientacji przez polecenie G jest ustawiona na zero, przełączenie następuje
   niezależnie od aktywnego polecenia grupy G nr 50:
   Kąty zaprogramowane przy pomocy `A2, B2, C2` są interpretowane zgodnie z jednym z aktywnych poleceń G `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` i `ORIPY2` . Wartości zaprogramowane z osiami orientacji są niezależnie od aktywnego polecenia grupy G nr 50 zawsze interpretowane jako pozycje osi obrotowych.

Składnia

Pozycje osi obrotowych

```
G1 X<Wartość> Y<Wartość> Z<Wartość> A<Wartość> B<Wartość> C<Wartość>
```

Kąt Eulera

```
G1 X<Wartość> Y<Wartość> Z<Wartość> A2=<Wartość> B2=<Wartość>
C2=<Wartość>
```

Wektor kierunkowy

```
G1 X<Wartość> Y<Wartość> Z<Wartość> A3=<Wartość> B3=<Wartość>
C3=<Wartość>
```

Wektor normalny do powierzchni na początku bloku

```
G1 X<Wartość> Y<Wartość> Z<Wartość> A4=<Wartość> B4=<Wartość>
C4=<Wartość>
```

Wektor normalny do powierzchni na końcu bloku

```
G1 X<Wartość> Y<Wartość> Z<Wartość> A5=<Wartość> B5=<Wartość>
C5=<Wartość>
```

Kąt wyprzedzenia

```
LEAD=<Wartość>
```

Kąt boczny

```
TILT=<Wartość>
```

Znaczenie

`G1` :	Interpolacja liniowa
`X, Y, Z`:	Pozycje osi liniowych
`A, B, C`:	Pozycje osi obrotowych
`A2=, B2=, C2=`:	Programowanie kąta (kąt Eulera albo RPY)
`A3=, B3=, C3=`:	Wektor kierunkowy we współrzędnych X, Y i Z WKS
`A4=, B4=, C4=`:	Wektor normalny do powierzchni na początku bloku we współrzędnych X, Y i Z WKS
`A5=, B5=, C5=`:	Wektor normalny do powierzchni na końcu bloku we współrzędnych X, Y i Z WKS
`LEAD=` :	Kąt wyprzedzenia [1)]
`TILT=` :	Kąt boczny [1)]
1) Interpretacja danych kąta zależy od ustawienia w MD21094 $MC_ORIPATH_MODE	

Dodatkowe informacje

Z reguły programy 5-osiowe są tworzone w systemach CAD/CAM i nie są wprowadzane do sterowania. Dlatego poniższe objaśnienia są skierowane głównie do programistów postprocesorów.

W celu programowania orientacji do dyspozycji są następujące polecenia:

Polecenie	Znaczenie
`ORIEULER`:	Kąt Eulera z kolejnością obrotów ZX'Z''
`ORIRPY`:	Kąt RPY z kolejnością obrotów ZY'X''
`ORIRPY2`:	Kąt RPY z kolejnością obrotów XY'Z''
`ORIVIRT1`:	Wirtualne osie orientacji z dowolnie zdefiniowaną kolejnością obrotów poprzez: MD21120 $MC_ORIAX_TURN_TAB_1
`ORIVIRT2`:	Wirtualne osie orientacji z dowolnie zdefiniowaną kolejnością obrotów poprzez: MD21130 $MC_ORIAX_TURN_TAB_2
`ORIAXPOS`:	Wirtualne osie orientacji z pozycjami osi obrotowych

Uwaga

Przez dane maszynowe producent maszyny może definiować różne warianty. Należy przestrzegać danych producenta maszyny.

Programowanie w kątach Eulera ORIEULER, kolejność obrotów Z X' Z"

Wartości zaprogramowane podczas programowania orientacji `ORIEULER` za pomocą `A2`, `B2`, `C2` są interpretowane jako kąty Eulera (w stopniach).

Nowy wektor orientacji wynika z następujących trzech obrotów pierwotnego wektora orientacji

1. poprzez oś obrotową `A2` wokół osi współrzędnych Z
2. poprzez oś obrotową `B2` wokół nowej osi współrzędnych X'
3. poprzez oś obrotową `C2` wokół nowej osi współrzędnych Z"

W tym przypadku wartość `C2` (obrót wokół nowej osi Z) jest bez znaczenia i nie musi być programowana.

Programowanie w kątach RPY ORIRPY, kolejność obrotów ZY'X"

Wartości zaprogramowane podczas programowania orientacji `ORIRPY` za pomocą `A2`, `B2`, `C2` są interpretowane jako kąty RPY (w stopniach) z kolejnością obrotów Z Y' X".

Uwaga

W przeciwieństwie do programowania przy pomocy `ORIEULER`, przy `ORIRPY` wszystkie trzy wartości mają wpływ na wektor orientacji.

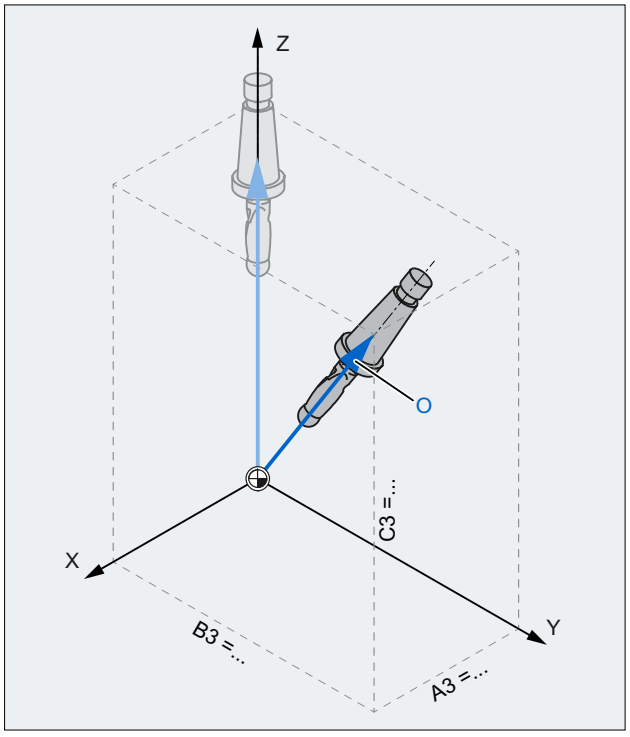
Nowy wektor orientacji wynika z następujących trzech obrotów pierwotnego wektora orientacji

1. poprzez oś obrotową `A2` wokół nowej osi współrzędnych Z
2. poprzez oś obrotową `B2` wokół nowej osi współrzędnych Y'
3. poprzez oś obrotową `C2` wokół osi współrzędnych X"

Programowanie wektora kierunkowego

Składowe wektora kierunkowego są programowane za pomocą `A3, B3, C3`. Wektor jest skierowany w kierunku uchwytu narzędzia; długość wektora jest przy tym bez znaczenia.

Niezaprogramowane składowe wektora są ustawiane na zero.



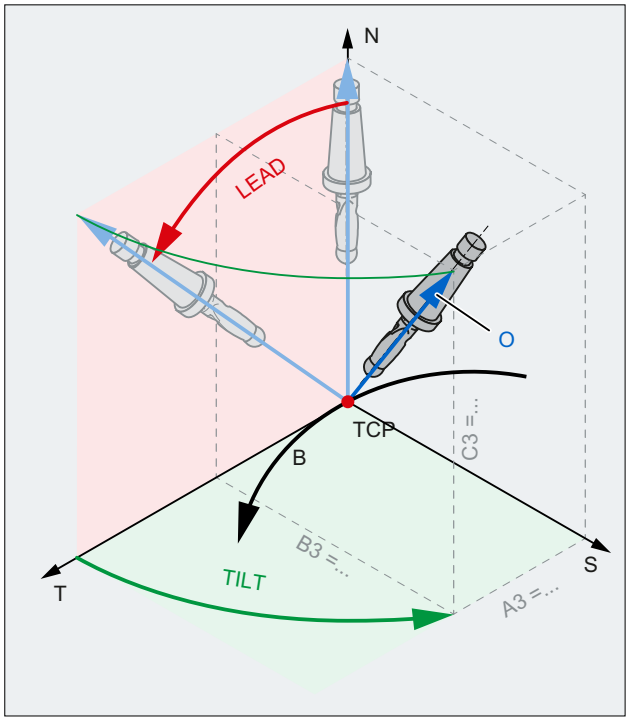
X, Y, Z	Osie współrzędnych WKS
A3, B3, C3	Składowa wektora kierunku
O	Wektor orientacji

Rysunek 7-1 Programowanie wektora kierunku

Programowanie orientacji narzędzia przy pomocy LEAD i TILT

Wynikająca orientacja narzędzia jest obliczana z:

- Stycznej do toru
- Wektora normalnego do powierzchni
  Na początku bloku `A4, B4, C4` i na końcu bloku `A5, B5, C5`
- Kąta wyprzedzenia `LEAD`
  Kąta wyprzedzenia w płaszczyźnie wyznaczonej przez styczną do toru i wektor normalny do powierzchni
- Kąta bocznego `TILT` na końcu bloku
  Kąt względem stycznej do toru w płaszczyźnie prostopadłej do wektora normalnego do powierzchni



T	Styczna do toru
S	Prostopadła do stycznej toru
N	Normalna do powierzchni
B	Tor
TCP	Tool Center Point
O	Wektor orientacji

Rysunek 7-2 Programowanie LEAD TILT

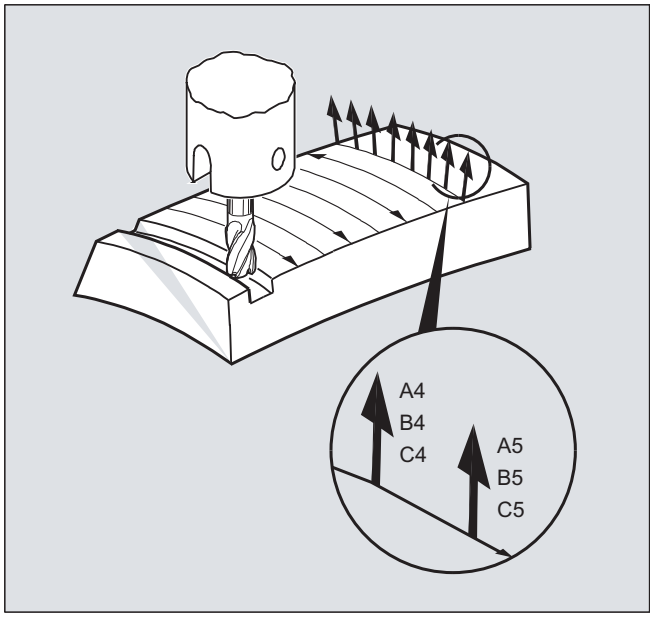
Uwaga

Zachowanie się w narożnikach wewnętrznych przy korekcji narzędzia 3D

Gdy blok jest skracany w narożniku wewnętrznym, zaprogramowana orientacja narzędzia jest przyjmowana również na końcu bloku.

7.2.5 Frezowanie czołowe (A4, B4, C4, A5, B5, C5)

Frezowanie czołowe służy do obróbki powierzchni dowolnie zakrzywionych.



Dla tego rodzaju frezowania trójwymiarowego potrzeba wierszowego opisu torów 3D na powierzchni obrabianego przedmiotu.

Obliczenia są przeprowadzane przy uwzględnieniu kształtu narzędzia i jego wymiarów - zazwyczaj w CAM. Obliczone na gotowo bloki NC są następnie przez postprocesory wczytywane do sterowania.

Programowanie zakrzywienia toru

Opis powierzchni

Opis zakrzywienia toru następuje przez wektory normalnych powierzchni przy pomocy następujących składowych:

`A4, B4, C4` wektor startowy na początku bloku

`A5, B5, C5` wektor końcowy na końcu bloku

Jeżeli w bloku jest podany tylko wektor startowy, wektor normalnej powierzchni pozostaje stały przez cały blok. Gdy w bloku jest podany tylko wektor końcowy, wówczas interpolacja odbywa się od wartości końcowej poprzedniego bloku przez interpolację dużego okręgu do zaprogramowanej wartości końcowej.

Gdy jest zaprogramowany wektor startowy i końcowy, wówczas między obydwoma kierunkami interpolacja następuje również przez interpolację dużego okręgu. Przez to dają się wytwarzać w sposób ciągły gładkie drogi po torze.

W położeniu podstawowym wektory normalnej powierzchni są niezależnie od aktywnej płaszczyzny G17 do G19 skierowane w kierunku Z.

Długość wektora jest bez znaczenia.

Nie zaprogramowane składowe wektora są ustawiane na zero.

Przy aktywnym ORIWKS (patrz "Odniesienie osi orientacji (ORIWKS, ORIMKS) (Strona 352)") wektory normalnej powierzchni odnoszą się do aktywnego frame i przy obrocie frame są równocześnie obracane.

Producent maszyny

Wektor normalnej powierzchni musi być prostopadły do stycznej do toru w ramach wartości granicznej ustawionej przez daną maszynową, w przeciwnym przypadku jest wyprowadzany alarm.

7.2.6 Odniesienie osi orientacji (ORIWKS, ORIMKS)

Przy programowaniu orientacji w układzie współrzędnych obrabianego przedmiotu przez

- kąt Eulera wzgl. kąt RPY albo
- wektor orientacji

można ustawić przebieg ruchu obrotowego przez `ORIMKS`/`ORIWKS`.

Uwaga

Producent maszyny

Rodzaj interpolacji orientacji jest ustalany daną maszynową:

MD21104 $MC_ORI_IPO_WITH_G_CODE

= FALSE: Odniesieniem są polecenia G ORIWKS i ORIMKS

= TRUE: Odniesieniem są polecenia G grupy nr 51 (ORIAXES, ORIVECT, ORIPLANE, ...)

Składnia

```
ORIMKS=...
ORIWKS=...
```

Znaczenie

`ORIMKS`:	Obrót w układzie współrzędnych maszyny
`ORIWKS`:	Obrót w układzie współrzędnych obrabianego przedmiotu

Uwaga

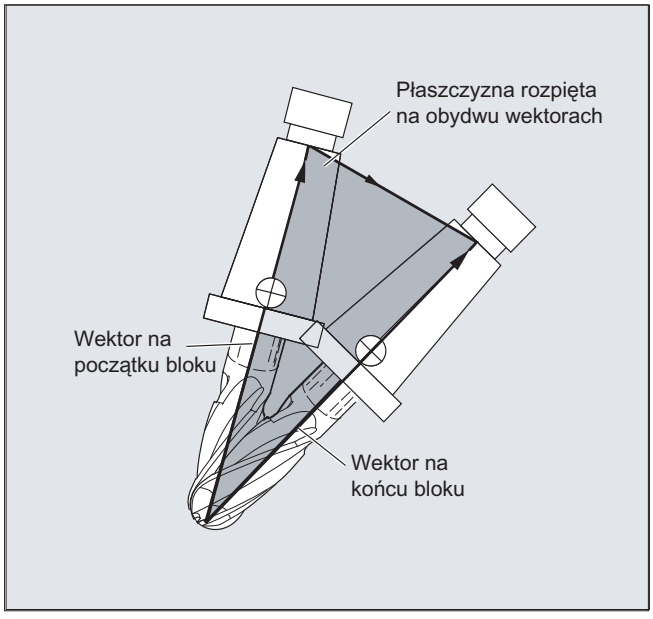
ORIWKS jest ustawieniem podstawowym. Jeżeli w przypadku programu 5-osiowego nie jest z góry jasne, na której maszynie ma on być wykonywany, należy z zasady wybrać ORIWKS. Jakie ruchy maszyna rzeczywiście wykonuje, zależy od jej kinematyki.

Przy pomocy `ORIMKS` można programować rzeczywiste ruchy maszyny, np. aby uniknąć kolizji z przyrządami itp.

Dodatkowe informacje

W przypadku ORIMKS wykonywany ruch narzędzia jest **zależny** od kinematyki maszyny. Przy zmianie orientacji z wierzchołkiem narzędzia stałym w przestrzeni następuje interpolacja liniowa między pozycjami osi obrotowej.

W przypadku ORIWKS wykonywany ruch narzędzia jest **niezależny** od kinematyki maszyny. Przy zmianie orientacji z wierzchołkiem narzędzia stałym w przestrzeni narzędzie porusza się w płaszczyźnie wyznaczonej przez wektor początkowy i wektor końcowy.



Położenia osobliwe

Uwaga

ORIWKS

Ruchy orientacji w zakresie położenia osobliwego maszyny 5-osiowej wymagają dużych ruchów w osiach maszyny. (Na przykład w przypadku skrętnej głowicy obrotowo-skrętnej z C, jako osią obrotu i A jako osią skrętu wszystkie położenia z A=0 są położeniami osobliwymi.)

Producent maszyny

Aby nie przeciążyć osi maszyny, układ prowadzenia prędkości bardzo zmniejsza prędkość po torze w pobliżu miejsc osobliwych.

Przy pomocy

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

transformacja może zostać tak sparametryzowana, by ruchy orientacji w pobliżu bieguna były prowadzone przez biegun i by była możliwa nieprzerwana obróbka.

Miejsca szczególne są traktowane tylko przy pomocy MD $MC_TRAFO5_POLE_LIMIT.

Literatura:
/FB3/ Podręcznik działania Funkcje specjalne; transformacja 3- do 5-osiowej (F2), punkt "Miejsca osobliwe i ich traktowanie".

7.2.7 Programowanie osi orientacji (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Funkcja "Osie orientacji" opisuje orientację narzędzia w przestrzeni i jest osiągalna przez zaprogramowanie przesunięcie dla osi obrotowych. Dalszy trzeci stopień swobody można osiągnąć przez dodatkowy obrót narzędzia wokół siebie samego. Ta orientacja narzędzia następuje dowolnie w przestrzeni przez trzecią oś obrotową i wymaga transformacji sześcioosiowej. Obrót narzędzia wokół siebie jest ustalany zależnie od rodzaju interpolacji wektorów obrotu z kątem obrotu THETA (patrz "Obroty orientacji narzędzia (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Strona 364)".

Osie orientacji są programowane przez identyfikatory osi A2, B2, C2.

Składnia

```
N... ORIAXES/ORIVECT            ; interpolacja liniowa lub interpolacja
                                dużego okręgu
N... G1 X Y Z A B C

N... ORIPLANE                   ; interpolacja orientacji płaszczyzny

N... ORIEULER/ORIRPY/ORIRPY2    ; kąt orientacji, kąt Eulera/RPY
N... G1 X Y Z A2= B2= C2=       ; programowanie kąta osi wirtualnych

N... ORIVIRT1/ORIVIRT2          ; Definicja wirtualnych osi orientacji 1/2
N... G1 X Y Z A3= B3= C3=       ; programowanie wektora kierunku
```

Uwaga

Dla zmian orientacji wzdłuż znajdującej się w przestrzeni pobocznicy stożka mogą być programowane kolejne przesunięcia obrotowej osi orientacji, patrz "Programowanie orientacji wzdłuż pobocznicy stożka (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Strona 356)".

Znaczenie

ORIAXES:	Interpolacja liniowa osi maszyny albo osi orientacji
ORIVECT:	Interpolacja dużego okręgu (identyczna z ORIPLANE)
ORIMKS: ORIWKS:	Obrót w układzie współrzędnych maszyny Obrót w układzie współrzędnych obrabianego przedmiotu Opis patrz "Odniesienie osi orientacji (ORIWKS, ORIMKS) (Strona 352)".

A= B= C=:	Programowanie pozycji osi maszyny
ORIEULER:	Programowanie orientacji przez kąt Eulera
ORIRPY:	Programowanie orientacji przez kąt RPY Kolejnością obrotów jest XYZ, przy czym obowiązuje: • A2 jest kątem obrotu wokół X • B2 jest kątem obrotu wokół Y • C2 jest kątem obrotu wokół Z
ORIRPY2:	Programowanie orientacji przez kąt RPY Kolejnością obrotów jest ZYX, przy czym obowiązuje: • A2 jest kątem obrotu wokół Z • B2 jest kątem obrotu wokół Y • C2 jest kątem obrotu wokół X
A2= B2= C2=:	Programowanie kąta osi wirtualnych
ORIVIRT1/ORIVIRT2:	Programowanie orientacji przez wirtualne osie orientacji Definicja 1: Ustalenie według MD21120 $MC_ORIAX_TURN_TAB_1 Definicja 2: Ustalenie według MD21130 $MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Programowanie wektora biegunowego osi kierunkowej

Dalsze informacje

Producent maszyny

MD21102 $ MC_ORI_DEF_WITH_G_CODE określa, w jaki sposób zdefiniowane są zaprogramowane kąty A2, B2, C2:

Definicja następuje według MD21100 $MC_ORIENTATION_IS_EULER (standard) albo według grupy G 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

MD21104 $ MC_ORI_IPO_WITH_G_CODE określa, który typ interpolacji jest aktywny: ORIWKS/ORIMKS czy ORIAXES/ORIVECT.

Tryb pracy JOG

Kąty orientacji są w tym trybie pracy zawsze interpolowane liniowo. Przy ruchu ciągłym i przyrostowym za pomocą przycisków ruchu można wykonywać ruch tylko jednej osi orientacji. Osie orientacji można przesuwać jednocześnie za pomocą kółek ręcznych.

Dla ruchu ręcznego osi orientacji działa specyficzny dla kanału przełącznik korekcji posuwu lub przełącznik korekcji posuwu szybkiego przy nałożeniu tego posuwu.

Przy pomocy następujących danych maszynowych jest możliwe oddzielne zadanie prędkości:

MD21160 $MC_JOG_VELO_RAPID_GEO

MD21165 $MC_JOG_VELO_GEO

MD21150 $MC_JOG_VELO_RAPID_ORI

MD21155 $MC_JOG_VELO_ORI

Uwaga

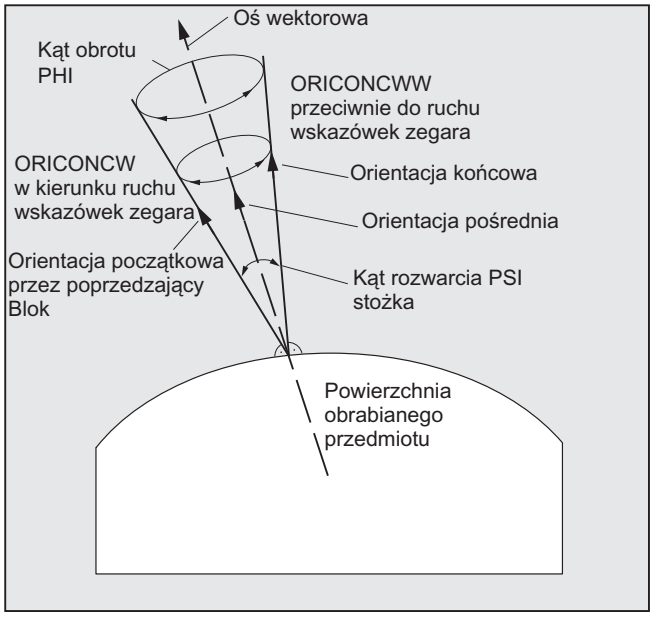
SINUMERIK 840D sl z "Pakietem transformacji Handling"

Za pomocą funkcji "Kartezjański ruch ręczny" przesunięcie osi geometrycznych w systemach odniesienia MKS, WKS i TCS można ustawić niezależnie od siebie w trybie JOG.

Literatura:
Opis funkcji, Funkcje rozszerzone; Transformacja kinematyczna (M1)

7.2.8 Programowanie orientacji wzdłuż pobocznicy stożka (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

Przy pomocy orientacji rozszerzonej jest możliwe wykonywanie zmian orientacji wzdłuż powierzchni pobocznicowej stożka znajdującej się w przestrzeni. Interpolacja wektora orientacji na pobocznicy stożka następuje przy pomocy poleceń modalnych ORICONxx. Dla interpolacji w płaszczyźnie można programować orientację końcową przy pomocy ORIPLANE. Generalnie orientacja startowa jest ustalana przez poprzedzające bloki.



Programowanie

Orientacja końcowa jest ustalana albo przez podanie zaprogramowania kąta jako kąt Eulera lub kąt RPY przy pomocy A2, B2, C2, albo przez zaprogramowanie pozycji osi obrotowych przy pomocy A, B, C. Dla osi orientacji wzdłuż pobocznicy stożka są wymagane dalsze dane programowe:

- Oś obrotu stożka jako wektor z `A6, B6, C6`
- Kąt rozwarcia `PSI` z identyfikatorem `NUT`
- Orientacja pośrednia w pobocznicy stożka z `A7, B7, C7`

Uwaga

Programowanie wektora biegunowego A6, B6, C6 dla osi obrotu stożka

Zaprogramowanie orientacji końcowej nie jest bezwarunkowo wymagane. Gdy orientacja końcowa nie jest podana, wówczas jest interpolowana pełna pobocznica stożka 360 stopni.

Programowanie kąta rozwarcia stożka z NUT=kąt

Podanie orientacji końcowej jest bezwzględnie wymagane.

Kompletna pobocznica stożka 360 stopni nie może być w ten sposób interpolowana.

Programowanie orientacji pośredniej A7, B7, C7 w pobocznicy stożka

Podanie orientacji końcowej jest bezwzględnie wymagane. Zmiana orientacji i kierunek obrotu są jednoznacznie ustalane przez trzy wektory orientacji startowej, końcowej i pośredniej. Wszystkie trzy wektory muszą przy tym być różne. Jeżeli zaprogramowana orientacja pośrednia jest równoległa do orientacji startowej albo końcowej, jest przeprowadzana liniowa interpolacja dużego okręgu orientacji w płaszczyźnie, która jest wyznaczana przez wektor startowy i końcowy

Rozszerzona interpolacja orientacji na pobocznicy stożka

```
N... ORICONCW albo ORICONCCW
N... A6= B6= C6= A3= B3= C3=
lub
N... ORICONTO
N... G1 X Y Z A6= B6= C6=
lub
N... ORICONIO
N... G1 X Y Z A7= B7= C7=
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Interpolacja na pobocznicę stożka z

wektorem biegunowym stożka w kierunku / przeciwnie do ruchu wskazówek zegara i orientacją końcową albo

przejściem stycznym i podaniem orientacji końcowej

albo

podaniem orientacji końcowej i orientacji pośredniej na pobocznicy stożka z

wielomianami kąta obrotu i wielomianami kąta rozwarcia

Parametry

`ORIPLANE`:	Interpolacja na płaszczyźnie (Interpolacja dużego okręgu)
`ORICONCW`:	Interpolacja na pobocznicy stożka w kierunku ruchu wskazówek zegara

ORICONCCW:	Interpolacja na pobocznicy stożka przeciwnie do ruchu wskazówek zegara
ORICONTO:	Interpolacja na pobocznicy stożka przejście styczne
A6= B6= C6=:	Programowanie osi obrotu stożka (wektor znormalizowany)
NUT=kąt:	Kąt rozwarcia stożka w stopniach
NUT=+179:	Kąt ruchu mniejszy albo równy 180 stopni
NUT=-181:	Kąt ruchu większy albo równy 180 stopni
ORICONIO:	Interpolacja na pobocznicy stożka
A7= B7= C7=:	Orientacja pośrednia (programowanie jako wektor znormalizowany)
PHI:	Kąt obrotu orientacji wokół osi kierunkowej stożka
PSI:	Kąt rozwarcia stożka
Możliwe wielomiany PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Oprócz poszczególnych kątów można programować również wielomiany maksymalnie 5. stopnia.

Przykład: Różne zmiany orientacji

Kod programu	**Komentarz**
…	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; Transformacja orientacji wł.
N30 ORIVECT	; Interpolowanie orientacji narzędzia jako wektor.
…	; Orientacja narzędzia na płaszczyźnie.
N40 ORIPLANE	; Wybór interpolacji wielkiego okręgu.
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Orientacja na płaszczyźnie Y/Z obrócona o 45 stopni, na końcu bloku jest osiągana orientacja (0,1/√2,1/√2).
…	
N70 ORICONCW	; Programowanie orientacji na pobocznicy stożka:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	; Kąt orientacji jest interpolowany na pobocznicy stożka z kierunkiem (0,0,1), aż do orientacji (1/√2,0,1/√2) w kierunku ruchu wskazówek zegara, kąt obrotu wynosi przy tym 270 stopni.
N90 A6=0 B6=0 C6=1	; Orientacja narzędzia przechodzi pełny obrót na tej samej pobocznicy stożka.

Dodatkowe informacje

Jeżeli zmiany orientacji mają zostać opisane na pobocznicy stożka leżącego dowolnie w przestrzeni, wówczas musi być znany wektor, wokół którego orientacja narzędzia ma wykonać obrót. Poza tym musi zostać zadana orientacja startowa i orientacja końcowa. Orientacja startowa wynika z bloku poprzedzającego, a orientacja końcowa musi albo zostać zaprogramowana albo ustalona przez inne warunki.

Programowanie w płaszczyźnie ORIPLANE odpowiada ORIVECT

Programowanie interpolacji wielkiego okręgu razem z wielomianami kąta odpowiada interpolacji liniowej i wielomianowej konturów. Orientacja narzędzia jest interpolowana w płaszczyźnie, która jest wyznaczona przez orientację startową i końcową. Gdy zostaną dodatkowo zaprogramowane wielomiany, wówczas wektor orientacji może ulec odchyleniu od płaszczyzny.

Programowanie okręgów w płaszczyźnie G2/G3, CIP i CT

Orientacja rozszerzona odpowiada interpolacji okręgów na płaszczyźnie. Odnośnie odpowiednich możliwości programowania okręgów z podaniem punktu środkowego albo promienia jak G2/G3, okręgu przez punkt pośredni CIP i okręgów stycznych CT patrz

Literatura: Podręcznik programowania Podstawy, "Programowanie poleceń dot. drogi".

Programowanie orientacji

Interpolacja wektora orientacji na pobocznicy stożka ORICONxx

Dla interpolacji orientacji na pobocznicy stożka mogą być wybierane cztery różne rodzaje interpolacji z grupy G nr 51:

1. Interpolacja na pobocznicy stożka w kierunku ruchu wskazówek zegara `ORICONCW` z podaniem orientacji końcowej i kierunku stożka albo kąta rozwarcia. Wektor kierunkowy jest programowany przy pomocy identyfikatorów `A6, B6, C6`, a kąt rozwarcia stożka przy pomocy identyfikatora NUT= zakres wartości w przedziale 0 do 180 stopni.
2. Interpolacja na pobocznicy stożka przeciwnie do ruchu wskazówek zegara `ORICONCWW` z podaniem orientacji końcowej i kierunku stożka lub kąta rozwarcia. Wektor kierunkowy jest programowany przy pomocy identyfikatorów `A6, B6, C6`, a kąt rozwarcia stożka przy pomocy identyfikatora NUT= zakres wartości w przedziale 0 do 180 stopni.
3. Interpolacja na pobocznicy stożka `ORICONIO` z podaniem orientacji końcowej i orientacji pośredniej, która jest programowana przy pomocy identyfikatorów `A7, B7, C7`.
4. Interpolacja na pobocznicy stożka `ORICONTO` z przejściem stycznym i podaniem orientacji końcowej. Wektor kierunkowy jest programowany przy pomocy identyfikatorów `A6, B6, C6`.

7.2.9 Zadanie orientacji dwóch punktów kontaktowych (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Programowanie zmiany orientacji przez drugą krzywą w przestrzeni ORICURVE

Kolejna możliwość programowania zmian orientacji polega na tym, by oprócz wierzchołka narzędzia wzdłuż krzywej w przestrzeni, programować również ruch drugiego punktu styku narzędzia za pomocą `ORICURVE`. Przez to mogą zostać jednoznacznie ustalone zmiany orientacji narzędzia, jak przy programowaniu samego wektora narzędzia.

Producent maszyny

Należy przestrzegać wskazówek producenta maszyny dot. identyfikatorów osi ustawianych przez daną maszynową odnośnie programowania 2. toru orientacji narzędzia.

Programowanie

Przy tym rodzaju interpolacji mogą dla obydwu krzywych w przestrzeni być programowane punkty przy pomocy `G1` wzgl. wielomiany przy pomocy `POLY`. Okręgi i ewolwenty są niedopuszczalne. Dodatkowo może zostać uaktywniona interpolacja Spline przy pomocy `BSPLINE` i funkcja "Połączenie krótkich bloków Spline".

Literatura:
Podręcznik działania Funkcje podstawowe; Przechodzenie płynne, Zatrzymanie dokładne, LookAhead (B1), Rozdział: Połączenie krótkich bloków Spline

Inne rodzaje Spline `ASPLINE` i `CSPLINE`, jak też uaktywnienie kompresora przy pomocy `COMPON`, `COMPCURV` lub `COMPCAD` są niedopuszczalne.

Ruch dwóch punktów styku narzędzia można zadać przy programowaniu dla współrzędnych wielomianów orientacji do maksymalnie 5. stopnia.

Rozszerzona interpolacja orientacji z dodatkową krzywą w przestrzeni i wielomiany dla współrzędnych

```
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)
```

Podanie ruchu drugiego punktu styku narzędzia i dodatkowe wielomiany każdorazowych współrzędnych

Parametry

`ORICURVE`	`Interpolacja orientacji z zadaniem ruchu dwóch punktów styku narzędzia.`
`XH YH ZH`	`Identyfikator współrzędnych drugiego punktu styku narzędzia dodatkowego konturu jako krzywa w przestrzeni`
`Możliwe wielomiany PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)`	`Oprócz każdorazowych punktów końcowych można dodatkowo programować krzywe w przestrzeni przy pomocy wielomianów.`
`xe, ye, ze`	`Punkty końcowe krzywej w przestrzeni`
`xi, yi, zi`	`Współczynniki wielomianów maksymalnie 5. stopnia`

Uwaga

Identyfikatory XH YH ZH do programowania 2. toru orientacji

Identyfikatory muszą zostać tak wybrane, by nie powstał konflikt z innymi identyfikatorami osi liniowych

Osie X Y Z

i osiami obrotowymi jak

A2 B2 C2 kąt Eulera wzgl. kąt RPY

A3 B3 C3 wektory kierunkowe

A4 B4 C4 wzgl. A5 B5 C5 wektory normalne do powierzchni

A6 B6 C6 wektory obrotu wzgl. A7 B7 C7 współrzędne punktu pośredniego

lub innymi parametrami interpolacji.

7.3 Wielomiany orientacji (PO[kąt], PO[współrzędna])

Niezależnie od tego, jaka interpolacja wielomianowa grupy nr G1 jest właśnie aktywna, mogą być programowane dwa różne typy wielomianów orientacji do maksymalnie 5. stopnia przy transformacji trzy- do pięcioosiowej.

1. Wielomiany dla **kątów:** kąt wyprzedzenia LEAD, kąt boczny TILT w odniesieniu do płaszczyzny, która jest wyznaczana przez orientację początkową i końcową.
2. Wielomiany dla **współrzędnych:** XH, YH, ZH drugiej krzywej w przestrzeni dla orientacji narzędzia punktu odniesienia na narzędziu.

Przy transformacji sześcioosiowej można w celu zorientowania narzędzia dodatkowo zaprogramować obrót wektora obrotu THT z wielomianami do maksymalnie 5. stopnia dla obrotów samego narzędzia.

Składnia

Wielomiany orientacji typu 1 dla **kąta**

```
N… PO[PHI]=(a2, a3, a4, a5)
N… PO[PSI]=(b2, b3, b4, b5)
```

Transformacja trzy- do pięcioosiowa

Wielomiany orientacji typu 2 dla **współrzędnych**

```
N… PO[XH]=(xe, x2, x3, x4, x5)
N… PO[YH]=(ye, y2, y3, y4, y5)
N… PO[ZH]=(ze, z2, z3, z4, z5)
```

Identyfikatory dla współrzędnych drugiego toru orientacji dla orientacji narzędzia

Dodatkowo można w obydwu przypadkach programować wielomian dla **obrotu** przy transformacji sześcioosiowej z

```
N… PO[THT]=(c2, c3, c4, c5)
```

lub

```
N… PO[THT]=(d2, d3, d4, d5)
```

odniesiona do toru interpolacja obrotu narzędzia

absolutna, względna i styczna interpolacja do zmiany orientacji

wektora orientacji. Jest to możliwe wtedy, gdy transformacja obsługuje wektor obrotu z przesunięciem programowalnym i interpolowanym przez kąt obrotu THETA.

Znaczenie

`PO[PHI]`	Kąt w płaszczyźnie między orientacją startową i końcową
`PO[PSI]`	Kąt, który opisuje odchylenie orientacji od płaszczyzny między orientacją startową i końcową
`PO[THT]`	Kąt obrotu zaprogramowanej przez obrót wektora obrotu przy pomocy THETA polecenia G grupy nr 54

PHI	Kąt wyprzedzenia LEAD
PSI	Kąt boczny TILT
THETA	Obrót wokół kierunku narzędzia w Z
PO[XH]	Współrzędna X punktu odniesienia na narzędziu
PO[YH]	Współrzędna Y punktu odniesienia na narzędziu
PO[ZH]	Współrzędna Z punktu odniesienia na narzędziu

Dodatkowe informacje

Wielomiany orientacji nie mogą być programowane

- gdy interpolacje spline ASPLINE, BSPLINE, CSPLINE są aktywne.
  Wielomiany typu 1 dla kąta orientacji są
  możliwe dla każdego rodzaju interpolacji oprócz Spline tzn. przy interpolacji liniowej z posuwem szybkim G0 wzgl. z posuwem G1
  przy interpolacji wielomianowej z POLY i
  przy interpolacji okręgu wzgl. ewolwenty z G2, G3, CIP, CT, INVCW i INCCCW.
  Wielomiany typu 2 dla współrzędnych orientacji są natomiast możliwe tylko wtedy, gdy jest aktywna interpolacja liniowa z posuwem szybkim G0 wzgl. z posuwem G1 albo interpolacja wielomianowa z POLY.
- gdy orientacja jest interpolowana przy pomocy interpolacji osi ORIAXES. W tym przypadku mogą być programowane bezpośrednio wielomiany z PO[A] i PO[B] dla osi orientacji A i B.

Wielomiany orientacji typu 1 z ORIVECT, ORIPLANE i ORICONxx

Przy interpolacji dużego okręgu i interpolacji na pobocznicy stożka z ORIVECT, ORIPLANE i ORICONxx są możliwe tylko wielomiany orientacji typu 1.

Wielomiany orientacji typu 2 z ORICURVE

Gdy jest aktywna interpolacja z dodatkową krzywą w przestrzeni ORICURVE, kartezjańskie składowe wektora orientacji są interpolowane i są możliwe tylko wielomiany orientacji typu 2.

7.4 Obroty orientacji narzędzia (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

Jeżeli w przypadku typów maszyn z ruchomym narzędziem również musi być możliwa zmiana orientacji narzędzia, wówczas każdy blok jest programowany z orientacją końcową. Zależnie od kinematyki maszyny można programować albo kierunek orientacji osi orientacji albo kierunek obrotu wektora orientacji THETA. Dla tych wektorów obrotu można programować różne rodzaje interpolacji:

- ORIROTA: Kąt obrotu do kierunku obrotu zadanego absolutnie.
- ORIROTR: Kąt obrotu w stosunku do płaszczyzny między orientacją startową i końcową.
- ORIROTT: Kąt obrotu w stosunku do zmiany wektora orientacji.
- ORIROTC: Styczny kąt obrotu do stycznej do toru.

Składnia

Tylko gdy jest aktywny rodzaj interpolacji ORIROTA, można programować kąt obrotu albo wektor obrotu na cztery możliwe sposoby:

1. Bezpośrednio pozycje osi obrotowych `A, B, C`
2. Kąt Eulera (w stopniach) przez `A2, B2, C2`
3. Kąt RPY (w stopniach) przez `A2, B2, C2`
4. Wektor kierunku przez `A3, B3, C3` (kąt obrotu przy pomocy `THETA=<wartość>`)

W przypadku gdy ORIOTR albo ORIOTT są aktywne, kąt obrotu można programować już tylko bezpośrednio przy pomocy `THETA`.

Obrót można również zaprogramować w oddzielnym bloku, przy czym nie następuje zmiana orientacji. Przy tym ORIROTR i ORIROTT nie mają znaczenia. W tym przypadku kąt obrotu jest zawsze interpretowany w odniesieniu do kierunku absolutnego (ORIROTA).

`N... ORIROTA`	Ustalenie interpolacji wektora obrotu
`N... ORIROTR`	
`N... ORIROTT`	
`N... ORIROTC`	
`N... A3= B3= C3= THETA=<wartość>`	Ustalenie obrotu wektora orientacji
`N... PO[THT]=(`d_2`, `d_3`, `d_4`, `d_5`)`	Interpolacja kąta obrotu z wielomianem 5. stopnia

Znaczenie

`ORIROTA`:	Kąt obrotu do kierunku obrotu zadanego absolutnie
`ORIROTR`:	Kąt obrotu w stosunku do płaszczyzny między orientacją startową i końcową
`ORIROTT`:	Kąt obrotu jako styczny wektor obrotu do zmiany orientacji
`ORIROTC`:	Kąt obrotu jako styczny wektor obrotu do stycznej do toru

THETA:	Obrót wektora orientacji
THETA=<wartość>:	Kąt obrotu w stopniach, który jest osiągany na końcu bloku
THETA=Θe:	Kąt obrotu z kątem końcowym Θ_e wektora obrotu
THETA=AC(…):	Przełączanie pojedynczymi blokami na absolutne podawanie wymiaru
THETA=AC(…):	Przełączanie pojedynczymi blokami na przyrostowe podawanie wymiaru
Θe:	Kąt końcowy wektora obrotu zarówno absolutny z G90, jak też względny z G91 (przyrostowe podawanie wymiarów) jest aktywny
PO[THT]=(....):	Wielomian dla kąta obrotu

Przykład: Obroty orientacji

```
Kod programu                              Komentarz
N10 TRAORI                                ; Uaktywnienie transformacji orientacji
N20 G1 X0 Y0 Z0 F5000                     ; Orientacja narzędzia
N30 A3=0 B3=0 C3=1 THETA=0                ; W kierunku Z z kątem obrotu 0
N40 A3=1 B3=0 C3=0 THETA=90               ; W kierunku X i obrót o 90 stopni
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)       ; Orientacja
N60 A3=0 B3=1 C3=0 THETA=IC(-90)          ; W kierunku Y i obrót na 180 stopni
N70 ORIROTT                               ; Pozostaje stała i obrót na 90 stopni
N80 A3=1 B3=0 C3=0 THETA=30               ; Kąt obrotu w stosunku do zmiany orientacji
                                          ; Kąt obrotu pod kątem 30 stopni do płaszczyz-
                                          ny X-Y
```

Przy interpolacji od bloku N40 kąt obrotu jest interpolowany liniowo od wartości startowej 0 do wartości końcowej 90 stopni. W bloku N50 kąt obrotu zmienia się z 90 stopni do 180 stopni według paraboli θ(u) = +90u2. W N60 może również zostać wykonany obrót, bez zmiany orientacji.

W przypadku N80 orientacja narzędzia jest obracana z kierunku Y do kierunku X. Przy tym zmiana orientacji leży w płaszczyźnie X-Y, a wektor obrotu tworzy z tą płaszczyzną kąt 30 stopni.

Dodatkowe informacje

ORIROTA

Kąt obrotu THETA jest interpolowany względem absolutnie ustalonego kierunku w przestrzeni. Podstawowy kierunek obrotu jest definiowany przez dane maszynowe

ORIROTR

Kąt obrotu THETA jest interpretowany względem płaszczyzny, która jest wyznaczana przez orientację początkową i orientację końcową.

ORIROTT

Kąt obrotu THETA jest interpretowany względem zmiany orientacji. Dla THEA=0 wektor obrotu jest interpolowany stycznie do zmiany orientacji i różni się tylko wtedy od ORIROTR, gdy dla orientacji został zaprogramowany co najmniej jeden wielomian dla "kąta wychylenia PSI". Z tego wynika zmiana orientacji, która nie przebiega w płaszczyźnie. Przez dodatkowo

zaprogramowany kąt obrotu `THETA` można wówczas np. tak interpolować wektor obrotu, by tworzył on zawsze określoną wartość do zmiany orientacji.

ORIROTC

Wektor obrotu jest interpolowany w stosunku do stycznej do toru z przesunięciem programowalnym przez kąt `THETA` . Dla kąta przesunięcia może przy tym być również programowany wielomian `PO[THT]=(c2, c3, c4, c5)` maksymalnie 5. stopnia.

7.5 Orientacje w stosunku do toru

7.5.1 Rodzaje orientacji w stosunku do toru

Przy pomocy tej rozszerzonej funkcji orientacja względna jest osiągana nie tylko na końcu bloku, lecz na całym przebiegu toru. Orientacja osiągnięta w poprzedzającym bloku jest przy pomocy interpolacji dużego okręgu prowadzona do zaprogramowanej orientacji końcowej. Zasadniczo są dwie możliwości programowania żądanej orientacji w stosunku do toru:

1. Orientacja narzędzia jak również obrót narzędzia jest przy pomocy ORIPATH, ORPATHTS interpolowana w stosunku do toru.
2. Wektor orientacji jest programowany i interpolowany zazwyczaj, jak dotychczas. Przy pomocy ORIROTC jest inicjalizowany obrót wektora orientacji w stosunku do stycznej do toru.

Składnia

Rodzaj interpolacji orientacji i obrotu narzędzia jest programowany przy pomocy:

`N... ORIPATH`	Orientacja w stosunku do toru
`N... ORIPATHS`	Orientacja w stosunku do toru z wygładzaniem przebiegu orientacji
`N... ORIROTC`	Odniesiona do toru interpolacja wektora obrotu

Załamanie orientacji wywołane przez naroże w przebiegu toru można wygładzić przy pomocy `ORIPATHS`. Kierunek i długość drogi ruchu cofnięcia są programowane przez wektor o składowych `A8=X, B8=Y, C8=Z`.

Za pomocą `ORIPATH/ORIPATHS` można programować różne odniesienia do stycznej do toru przez trzy kąty

- `LEAD=` podanie kąta wyprzedzenia w odniesieniu do toru i powierzchni
- `TILT=` podanie kąta bocznego w odniesieniu do toru i powierzchni
- `THETA=` kąt obrotu

dla całego przebiegu toru. Do kąta obrotu `THETA` mogą za pomocą `PO[THT]=(...)` być dodatkowo programowane wielomiany maksymalnie 5. stopnia.

Uwaga

Producent maszyny

Należy przestrzegać danych producenta maszyny. Przez projektowalne dane maszynowe i ustawcze mogą dla rodzaju orientacji w stosunku do toru zostać dokonane dalsze ustawienia. Dalsze objaśnienia patrz

Literatura:
/FB3/ Podręcznik działania Funkcje specjalne; Transformacje 3- do 5-osiowe (F2), Rozdział: "Orientacja"

Znaczenie

Interpolację kątów LEAD i TILT można przez daną maszynową różnie ustawiać:

- Odniesienie orientacji narzędzia zaprogramowane przy pomocy LEAD i TILT jest utrzymywane przez cały blok.
- Kąt wyprzedzenia LEAD: Obrót wokół kierunku prostopadłego do stycznej i wektora normalnego TILT: Obrót orientacji wokół wektora normalnego.
- Kąt wyprzedzenia LEAD: Obrót wokół kierunku prostopadłego do stycznej i wektora normalnego, kąt boczny TILT: Obrót orientacji wokół kierunku stycznego do toru.
- Kąt obrotu THETA: Obrót narzędzia wokół swojej osi z dodatkową trzecią osią obrotową jako oś orientacji przy transformacji sześcioosiowej.

Uwaga

Odniesiona do toru orientacja razem z OSC, OSS, OSSE, OSD, OST jest niedopuszczalna

Odniesiona do toru interpolacja orientacji ORIPATH lub ORIPATHS i ORIOTC nie może być programowana razem z wygładzaniem przebiegu orientacji za pomocą jednego z poleceń G z grupy nr 34. W tym celu musi być aktywne OSOF.

7.5.2 Odniesiony do toru obrót orientacji narzędzia (ORIPATH, ORIPATHS, kąt obrotu)

W przypadku transformacji sześcioosiowej można w celu orientacji narzędzia dowolnie w przestrzeni również obrócić narzędzie wokół siebie samego przy pomocy trzeciej osi obrotowej. Przy odniesionym do toru obrocie orientacji narzędzia przy pomocy ORIPATH lub ORIPATHS można zaprogramować dodatkowy obrót przez kąt obrotu THETA. Alternatywnie do tego kąty LEAD i TILT mogą być programowane przez wektor, który leży na płaszczyźnie prostopadle do kierunku narzędzia.

Producent maszyny

Należy przestrzegać danych producenta maszyny. Przez daną maszynową można różnie ustawić interpolację kątów LEAD i TILT.

Składnia

Obrót orientacji narzędzia i narzędzia

Rodzaj orientacji narzędzia w stosunku do toru jest uaktywniany przy pomocy ORIPATH albo ORIPATHS.

N... ORIPATH	Uaktywnienie rodzaju orientacji w odniesieniu do toru
N... ORIPATHS	Uaktywnienie rodzaju orientacji w odniesieniu do toru z wygładzaniem przebiegu orientacji

Uaktywnienie trzech możliwych kątów z działaniem obrotowym:

N... LEAD=	Kąt dla zaprogramowanej orientacji w stosunku do wektora normalnej powierzchni

`N... TILT=`	Kąt dla zaprogramowanej orientacji w płaszczyźnie prostopadle do stycznej do toru w stosunku do wektora normalnej powierzchni
`N... THETA=`	Kąt obrotu w stosunku do zmiany orientacji wokół kierunku narzędzia trzeciej osi obrotowej

Wartości kątów na końcu bloku są programowane przy pomocy `LEAD=wartość`, `TILT=wartość` wzgl. `THETA=wartość`. Dodatkowo do stałych kątów mogą dla wszystkich trzech kątów być programowane wielomiany maksymalnie 5. stopnia.

`N... PO[PHI]=(a2, a3, a4, a5)`	Wielomian dla kąta wyprzedzenia LEAD
`N... PO[PSI]=(b2, b3, b4, b5)`	Wielomian dla kąta bocznego TILT
`N... PO[THT]=(d2, d3, d4, d5)`	Wielomian dla kąta obrotu THETA

Przy programowaniu wyższe współczynniki wielomianu, które wynoszą zero, można pominąć. Przykład `PO[PHI]=a2` daje parabolę dla kąta wyprzedzenia `LEAD`.

Znaczenie

Orientacja narzędzia odniesiona do toru

`ORIPATH`:	Orientacja narzędzia odniesiona do toru
`ORIPATHS`:	Orientacja narzędzia w odniesieniu do toru, załamanie w przebiegu orientacji jest wygładzane
`LEAD`:	Kąt w stosunku do wektora normalnej powierzchni, w płaszczyźnie wyznaczonej przez styczną do toru i wektor normalnej powierzchni
`TILT`:	Obrót orientacji wokół kierunku Z lub obrót wokół stycznej do toru
`THETA`:	Obrót wokół kierunku narzędzia do Z
`PO[PHI]`:	Wielomian orientacji dla kąta wyprzedzenia LEAD
`PO[PSI]`:	Wielomian orientacji dla kąta bocznego TILT
`PO[THT]`:	Wielomian orientacji dla kąta obrotu THETA

Uwaga

Kąt obrotu THETA

Do obrotu narzędzia z trzecią osią obrotową jako osią orientacji wokół siebie samego jest wymagana transformacja sześcioosiowa.

7.5.3 Interpolacja obrotu narzędzia względem toru (ORIROTC, THETA)

Interpolacja z wektorami obrotu

Do zaprogramowanego przy pomocy ORIROTC obrotu narzędzia w stosunku do stycznej do toru można interpolować wektor obrotu również z przesunięciem programowanym przez kąt

obrotu THETA. Przy tym można dla kąta przesunięcie z PO[THT] programować wielomian do maksymalnie 5. stopnia.

Składnia

`N... ORIROTC`	Inicjalizacja obrotu narzędzia względem stycznej do toru
`N... A3= B3= C3= THETA=wartość`	Ustalenie obrotu wektora orientacji
`N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	Interpolacja kąta przesunięcia z wielomianem maksymalnie 5. stopnia

Obrót można również zaprogramować w oddzielnym bloku, przy czym nie następuje zmiana orientacji.

Znaczenie

Odniesiona do toru interpolacja obrotu narzędzia przy transformacji sześcioosiowej

`ORIROTC`:	Ustawienie stycznego wektora obrotu do stycznej do toru
`THETA=wartość`:	Kąt obrotu w stopniach, który jest osiągany na końcu bloku
`THETA=θe`:	Kąt obrotu z kątem końcowym Θ_e wektora obrotu
`THETA=AC(…)`:	Przełączanie pojedynczymi blokami na absolutne podawanie wymiaru
`THETA=IC(…)`:	Przełączanie pojedynczymi blokami na przyrostowe podawanie wymiaru
`PO[THT]=(c2, c3, c4, c5)`:	Interpolacja kąta przesunięcia maksymalnie 5. stopnia

Uwaga

Interpolacja wektora obrotu ORIROTC

Ustawienie również obrotu narzędzia w stosunku do stycznej do toru przeciwnie do kierunku jego orientacji jest możliwe tylko w przypadku transformacji sześcioosiowej.

Przy aktywnym ORIROTC

Wektora obrotu ORIROTA nie można zaprogramować. W przypadku zaprogramowania jest wyprowadzany ALARM 14128 "Absolutne zaprogramowanie obrotu narzędzia przy aktywnym ORIROTC".

Kierunek orientacji narzędzia przy transformacji trzy- do pięcioosiowej

Kierunek orientacji narzędzia może być programowany jak zazwyczaj przy transformacji trzy- do pięcioosiowej przez kąt Eulera wzgl. kąt RPY albo wektory biegunowe. Również są możliwe zmiany orientacji narzędzia w przestrzeni przez programowanie interpolacji dużego okręgu ORIVECT, interpolacji liniowej osi orientacji ORIAXES, wszystkie interpolacje na pobocznicy

stożka ORICONxx, jak też interpolacji dodatkowo do krzywej w przestrzeni z dwoma punktami kontaktowymi narzędzia ORICURVE.

G....:	Podanie rodzaju ruchu osi obrotowych
X, Y, Z:	Podanie osi liniowych
ORIAXES:	Interpolacja liniowa osi maszyny albo osi orientacji
ORIVECT:	Interpolacja dużego okręgu (identyczna z ORIPLANE)
ORIMKS: ORIWKS:	Obrót w układzie współrzędnych maszyny Obrót w układzie współrzędnych obrabianego przedmiotu Opis patrz punkt Obroty orientacji narzędzia
A= B= C=:	Programowanie pozycji osi maszyny
ORIEULER:	Programowanie orientacji przez kąt Eulera
ORIRPY:	Programowanie orientacji przez kąt RPY
A2= B2= C2=:	Programowanie kąta osi wirtualnych
ORIVIRT1: ORIVIRT2:	Programowanie orientacji przez wirtualne osie orientacji (Definicja 1), ustalenie według MD $MC_ORIAX_TURN_TAB_1 (Definicja 2), ustalenie według MD $MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Programowanie wektora biegunowego osi kierunkowej
ORIPLANE:	Interpolacja na płaszczyźnie (Interpolacja dużego okręgu)
ORICONCW:	Interpolacja na pobocznicy stożka w kierunku ruchu wskazówek zegara
ORICONCCW:	Interpolacja na pobocznicy stożka przeciwnie do ruchu wskazówek zegara
ORICONTO:	Interpolacja na pobocznicy stożka przejście styczne
A6= B6= C6=:	Programowanie osi obrotu stożka (wektor znormalizowany)
NUT=kąt	Kąt rozwarcia stożka w stopniach
NUT=+179	Kąt ruchu mniejszy albo równy 180 stopni
NUT=-181	Kąt ruchu większy albo równy 180 stopni
ORICONIO:	Interpolacja na pobocznicy stożka
A7= B7= C7=:	Orientacja pośrednia (programowanie jako wektor znormalizowany)
ORICURVE XH YH ZH np. z wielomianami PO[XH]=(xe, x2, x3, x4, x5)	Interpolacja orientacji z zadaniem ruchu dwóch punktów kontaktowych narzędzia. Oprócz każdorazowych punktów końcowych można programować dodatkowe wielomiany krzywych w przestrzeni.

Uwaga

Gdy orientacja narzędzia jest interpolowana z aktywnym ORIAXES przez osie orientacji, wówczas odniesiona do toru inicjalizacja kąta obrotu jest spełniana tylko na końcu bloku.

7.5.4 Wygładzanie przebiegu orientacji (ORIPATHS A8=, B8=, C8=)

Przy wykazujących stałe przyśpieszenie zmianach orientacji na konturze, przerwania ruchów po torze, które szczególnie mogą występować na narożnikach konturu, są niepożądane. Wynikające z tego załamanie w przebiegu orientacji można wygładzić przez wstawienie własnego bloku pośredniego. Zmiana orientacji następuje wówczas ze stałym przyspieszeniem, gdy podczas przeorientowania jest aktywne również ORIPATHS. W tej fazie może zostać przeprowadzony ruch cofnięcia narzędzia.

Producent maszyny

Należy przestrzegać wskazówek producenta maszyny dot. ewentualnie predefiniowanych danych maszynowych i ustawczych, przy pomocy których ta funkcja jest uaktywniana.

Przez daną maszynową można ustawić, jak wektor cofnięcia jest interpretowany:

1. W układzie współrzędnych narzędzia współrzędna Z jest definiowana przez kierunek narzędzia.
2. W układzie współrzędnych obrabianego przedmiotu współrzędna Z jest definiowana przez aktywną płaszczyznę.

Dalsze objaśnienia do funkcji "Orientacja odniesiona do toru" patrz
Literatura:
Podręcznik działania Funkcje specjalne; Transformacje wieloosiowe (F2

Składnia

Dla stałych orientacji narzędzia w odniesieniu do całego toru są na narożniku konturu wymagane dalsze dane programowe. Kierunek i długość drogi tego ruchu jest programowany przez wektor ze składowymi A8=X, B8=Y, C8=Z:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Znaczenie

`ORIPATHS`:	Orientacja narzędzia w odniesieniu do toru, załamanie w przebiegu orientacji jest wygładzane.
`A8= B8= C8=`:	Składowe wektora dla kierunku i długości drogi
`X, Y, Z`:	Ruch cofnięcia w kierunku narzędzia

Uwaga

Programowanie wektora kierunkowego A8, B8, C8

Gdy długość wektora jest równa zeru, ruch cofnięcia nie następuje.

ORIPATHS

Odniesiona do toru orientacja narzędzia staje się aktywna z ORIPATHS. W przeciwnym przypadku orientacja jest przy pomocy liniowej interpolacji dużego okręgu transferowana z orientacji startowej do końcowej.

7.6 Kompresja orientacji (COMPON, COMPCURV, COMPCAD, COMPSURF)

Programy NC, w których jest aktywna transformacja orientacji (TRAORI) i są programowane orientacje narzędzi (obojętnie jakiego rodzaju), mogą być kompresowane przy zachowaniu zadanych tolerancji.

Programowanie

Orientacja narzędzia

W przypadku gdy jest aktywna transformacja orientacji (TRAORI), w przypadku maszyn 5-osiowych orientacja narzędzia może być programowana następująco (niezależnie od kinematyki):

- Programowanie **wektora** kierunkowego przez:
  `A3=<...> B3=<...> C3=<...>`
- Programowanie **kątów** Eulera wzgl. **kątów** RPY przez:
  `A2=<...> B2=<...> C2=<...>`

Obrót narzędzia

W przypadku maszyn **6-osiowych** można dodatkowo do orientacji narzędzia zaprogramować jeszcze jego obrót.

Programowanie kąta obrotu następuje przy pomocy:

`THETA=<...>`

Patrz " Obroty orientacji narzędzia (Strona 364) ".

Uwaga

Bloki NC, w których dodatkowo jest zaprogramowany obrót, dają się kompresować tylko wtedy, gdy kąt obrotu zmienia się **liniowo**. Tzn. dla kąta obrotu nie może być zaprogramowany wielomian przy pomocy PO[THT]=(...).

Ogólna forma kompresowalnego bloku NC

Ogólna forma kompresowalnego bloku NC może dlatego wyglądać następująco:

```
N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

wzgl.

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

Uwaga

Wartości pozycji mogą być podawane bezpośrednio (np. X90) lub pośrednio przez przyporządkowanie parametrów (np. X=R1*(R2+R3)).

Programowanie orientacji narzędzia przez pozycje osi obrotowych

Orientacja narzędzia może być również podana przez pozycje osi obrotowych, np. w formie:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

W tym przypadku kompresja jest przeprowadzana na dwa różne sposoby, zależnie od tego, czy interpolacja dużego okręgu jest przeprowadzana czy nie. Gdy interpolacja wielkiego okręgu nie następuje, wówczas skompresowana zmiana orientacji jest przedstawiana w zwykły sposób przez wielomiany osiowe dla osi obrotowych.

Dokładność konturu

Zależnie od ustawionego trybu kompresji (MD20482 $MC_COMPRESSOR_MODE) przy kompresji działają dla osi geometrycznych i osi orientacji albo zaprojektowane tolerancje specyficzne dla osi (MD33100 $MA_COMPRESS_POS_TOL) albo następujące tolerancje specyficzne dla kanału, ustawiane przez dane ustawcze:

SD42475 $SC_COMPRESS_CONTUR_TOL (maksymalne odchylenie konturu)

SD42476 $SC_COMPRESS_ORI_TOL (maksymalne odchylenie kątowe orientacji narzędzia)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (maksymalne odchylenie kątowe dla kąta obrotu narzędzia) (dostępne tylko w przypadku maszyn 6-osiowych)

Literatura:
Podręcznik działania Funkcje podstawowe; Transformacja 3- do 5-osiowa (F2),
punkt: "Kompresowanie orientacji"

Aktywacja / Dezaktywacja

Funkcje kompresora są włączane przez modalne polecenia G `COMPON`, `COMPCURV`, `COMPCAD` lub `COMPSURF`.

Funkcja kompresora wyłącza się za pomocą `COMPOF`.

Patrz "Kompresja bloków NC (COMPON, COMPCURV, COMPCAD) (Strona 271) ".

Uwaga

Ruch orientacji jest kompresowany tylko przy aktywnej interpolacji dużego okręgu (tzn. zmiana orientacji narzędzia następuje w płaszczyźnie, która jest wyznaczana przez orientację startową i końcową).

Interpolacja dużego okręgu jest przeprowadzana pod następującymi warunkami:

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  ORIWKS jest aktywne a
  orientacja jest programowana przy pomocy wektorów (z A3, B3, C3 wzgl. A2, B2, C2).
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 i
  ORIVECT lub ORIPLANE jest aktywne.
  Orientacja narzędzia może być zaprogramowana jako wektor kierunkowy lub z pozycjami osi obrotowych. Jeżeli jest aktywne jedno z poleceń G `ORICONxx` albo `ORICURVE` lub są zaprogramowane wielomiany dla kąta orientacji (`PO[PHI]` i `PO[PSI]`), nie jest przeprowadzana interpolacja dużego okręgu.

Przykład

W poniższym przykładzie programowania jest kompresowany okrąg, który jest przybliżany przez ciąg prostych. Orientacja narzędzia porusza się przy tym synchronicznie do niego na pobocznicy stożka. Chociaż kolejne zaprogramowane zmiany orientacji przebiegają niestale, funkcja kompresora generuje gładki przebieg orientacji.

```
Programowanie                              Komentarz
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               ; Maksymalne odchylenie konturu = 0.05 mm
$SC_COMPRESS_ORI_TOL=5                     ; Maksymalne odchylenie orientacji = 5 stopni

TRAORI
COMPCURV                                   Następuje ruch po okręgu, który jest tworzo-
                                           ny z wieloboków. Orientacja porusza się przy
                                           tym po stożku wokół osi Z z kątem rozwarcia
                                           45 stopni.
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 Wygładzanie przebiegu orientacji, załączenie/wyłączenie (ORISON, ORISOF)

„Wygładzanie przebiegu orientacji" jest załączane/wyłączane w programie obróbki za pomocą polecenia G z grupy nr 61. Polecenia działają modalnie.

Warunki

- System z transformacja 5/6-osiową
- Funkcja kompresora COMPCAD jest aktywana.

Składnia

```
ORISON
...
ORISOF
```

Znaczenie

ORISON:	Wygładzanie przebiegu orientacji załączone
ORISOF:	Wygładzanie przebiegu orientacji wyłączone

Przykład

```
Kod programu                    Komentarz
...
TRAORI()                        ; Włączenie transformacji orientacji.
COMPCAD                         ; Załączenie funkcji kompresora COMPCAD.
ORISON                          ; Włączenie wygładzania orientacji.
$SC_ORISON_TOL=1.0              ; Maksymalne odchylenie kątowe orientacji narzędzia =
                                1,0 stopnia.
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                          ; Wyłączenie wygładzania orientacji.
...
```

Orientacja ulega skrętowi o 90 stopni w płaszczyźnie XZ, od -45 do +45 stopni. W wyniku wygładzenia przebiegu orientacji nie osiąga już ona maksymalnych wartości kątowych -45 wzgl. +45 stopni.

7.8 Transformacja kinematyczna

7.8.1 Włączenie transformacji strony czołowej (TRANSMIT)

Transformacja strony czołowej (TRANSMIT) jest włączana w programie obróbki albo akcji synchronicznej instrukcją `TRANSMIT`.

Składnia

```
TRANSMIT
TRANSMIT(<n>)
```

Znaczenie

`TRANSMIT`:	Włączyć TRANSMIT z pierwszym zestawem danych TRANSMIT
`TRANSMIT(n)`:	Włączyć TRANSMIT z <n>-tym zestawem danych TRANSMIT

Uwaga

Aktywna w kanale transformacja TRANSMIT jest wyłączana przez:

- Wyłączenie transformacji: `TRAFOOF`
- Włączenie innej transformacji: np. `TRACYL`, `TRAANG`, `TRAORI`

7.8.2 Włączenie transformacji pobocznicy walca (TRACYL)

Transformacja pobocznicy walca (TRACYL) jest włączana w programie obróbki albo akcji synchronicznej instrukcją `TRACYL`.

Składnia

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

Znaczenie

`TRACYL(<d>)`:	Włączenie TRACYL z pierwszym zestawem danych TRACYL oraz średnicy roboczej <d>
`TRACYL (<d>,<n>)`:	Włączenie TRACYL z <n>-tym zestawem danych TRACYL oraz średnicy roboczej <d>
`<d>`:	Średnica odniesienia lub robocza Zaprogramowana wartość F powinna być większa od 1.

<n>:	Numer zestawu danych TRACYL (opcjonalnie)	
	Zakres wartości:	1, 2
<k>:	Parametr <k> dotyczy tylko typu transformacji 514	
	k = 0:	bez korekcji ścianki rowka
	k = 1:	Z korekcją ścianki rowka
	Jeśli nie podano parametru, działa sparametryzowane położenie podstawowe: $MC_TRACYL_DEFAULT_MODE_<n> gdzie <n> = numer zestawu danych TRACYL	

Uwaga

Aktywna w kanale transformacja TRACYL jest wyłączana przez:

- Wyłączenie transformacji: `TRAFOOF`
- Włączenie innej transformacji: np. `TRAANG`, `TRANSMIT`, `TRAORI`

Przykład

Kod programu	Komentarz
...	
N40 TRACYL(40.)	; Włączanie TRACYL z pierwszym zestawem danych TRACYL oraz średnicy roboczej 40 mm
...	

Dodatkowe informacje

Struktura programu

Program obróbki do frezowania rowka z transformacją TRACYL 513 (TRACYL z korekcją ścianki rowka składa się z reguły z następujących kroków:

1. Wybór narzędzia.
2. Wybór TRACYL.
3. Wybór odpowiedniego przesunięcia współrzędnych (FRAME)
4. Pozycjonowanie.
5. Programowanie OFFN.
6. Wybór korekcji promienia narzędzia (WRK).
7. Blok dosuwu (zrealizowanie WRK i dosunięcie do ścianki rowka).
8. Kontur linii środkowej rowka.
9. Cofnięcie wyboru WRK.
10. Blok odsunięcia (cofnięcie realizacji korekcji promienia narzędzia i odsunięcie od ścianki rowka).
11. Pozycjonowanie.

12. TRAFOOF.

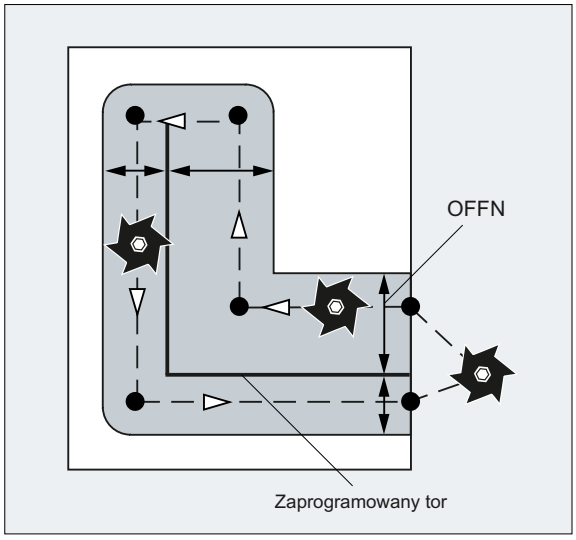
13. Ponowny wybór pierwotnego przesunięcia współrzędnych (FRAME).

Przesunięcie konturu (OFFN)

Żeby frezować rowki za pomocą TRACYL 513, w programie programowana jest linia środkowa rowka, a poprzez adres OFFN **połowa jego szerokości**.

Aby uniknąć uszkodzenia ścianki rowka, OFFN działa dopiero z wybraną korekcją promienia narzędzia.

Zmiana OFFN w ramach programu obróbki jest możliwa. W ten sposób linia środkowa rowka może zostać przesunięta ze środka:



Uwaga

OFFN powinien mieć co najmniej taką wielkość jak promień narzędzia, aby wykluczyć uszkodzenie przeciwległej ścianki rowka.

Uwaga

OFFN z TRACYL działa inaczej, niż bez TRACYL. Ponieważ OFFN jest uwzględniane przy aktywnej korekcji promienia narzędzia również bez TRACYL, OFFN powinno po TRAFOOF ponownie zostać ustawione na zero.

UWAGA
Działanie OFFN zależy od typu transformacji
W przypadku transformacji TRACYL 513 (TRACYL z korekcją ścianki rowka) jest dla OFFN programowana połowa szerokości rowka.
W przypadku transformacji TRACYL 512 (TRACYL bez korekcji ścianki rowka) wartość OFFN działa natomiast jako naddatek do WRK.

Korekcja promienia narzędzia (WRK)

W przypadku transformacji TRACYL 513 WRK nie jest wliczana w stosunku do ścianki rowka, lecz do zaprogramowanego środka rowka. Żeby narzędzie mogło przejechać na lewo od ścianki rowka, zamiast G41 należy zaprogramować instrukcję G42 albo podać wartość OFFN ze znakiem ujemnym.

Średnica narzędzia

Za pomocą TRACYL oraz narzędzia, którego średnica jest mniejsza niż szerokość rowka, nie jest wykonywana taka sama geometria ścianki rowka, jak w przypadku narzędzia, którego średnica jest równa szerokości rowka. W celu poprawy dokładności zaleca się wybrać średnicę narzędzia tylko nieco mniejszą od szerokości rowka.

Korzystanie z osi

Uwaga

Następujące osie nie mogą być stosowane jako osie pozycjonowania, wzgl. osie ruchu wahadłowego:

- oś geometryczna w kierunku obwodowym pobocznicy walca (oś Y)
- dodatkowa oś liniowa w przypadku korekcji ścianki rowka (oś Z)

7.8.3 Włączenie transformacji kąta skośnego z kątem programowalnym (TRAANG)

Transformacja kąta skośnego (TRAANG) z programowalnym kątem jest załączana w programie obróbki albo akcji synchronicznej za pomocą instrukcji `TRAANG`.

Składnia

```
TRAANG
TRAANG()
TRAANG(,<n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

Znaczenie

`TRAANG`: `TRAANG()`:	Włączenie TRAANG z pierwszym zestawem danych TRAANG i ostatnio obowiązującym kątem <α>
`TRAANG(,<n>)`:	Włączenie TRAANG z <n>-tym zestawem danych TRAANG i ostatnio obowiązującym kątem <α>
`TRAANG(<α>)`:	Włączenie TRAANG z pierwszym zestawem danych TRAANG i kątem <α>
`TRAANG(<α>,<n>)`:	Włączenie TRAANG z <n>-tym zestawem danych TRAANG i kątem <α>

<α>:	Kąt osi skośnej (opcjonalnie)	
	Zakres wartości:	-90° < α < + 90°
	Gdy nie został podany żaden kąt działa ustawienie podstawowe sparametryzowane w danej maszynowej: MD2xxxx $MC_TRAANG_ANGLE_<n>	
<n>:	Numer zestawu danych TRAANG (opcjonalnie)	
	Zakres wartości:	1, 2

Uwaga

Aktywna w kanale transformacja `TRAANG` jest wyłączana przez:

- Wyłączenie transformacji: `TRAFOOF`
- Załączenie innej transformacji: np. `TRACYL`, `TRANSMIT`, `TRAORI`

Przykład

Kod programu	Komentarz
`N20 TRAANG(45)`	`; Załączenie TRAANG z pierwszym zestawem danych TRAANG i kątem 45°.`

7.8.4 Wcinanie skośne na szlifierkach (G5, G7)

Funkcje G7 oraz G5 służą do ułatwienia programowania obróbki wcinającej skośnej na szlifierkach z transformacją "Oś skośna" (TRAANG) w ten sposób, że przy wcinaniu ruch wykonuje oś skośna.

Należy zaprogramować przy tym tylko żądaną pozycję końcową ruchu wcinającego w X i Z. Wychodząc od aktualnej pozycji osi X, zaprogramowanej pozycji końcowej oraz kąta α osi skośnej NC obliczy przynależną pozycję startową dla G7 i nastąpi dosunięcie.

Pozycja startowa wynika z punktu przecięcia obu prostych:

- Prosta równoległa do osi Z w odstępnie od aktualnej pozycji osi X
- Prosta równoległa do osi skośnej zaprogramowanej przez pozycję docelową

Za pomocą następującej G5 nastąpi ruch osi skośnej na zaprogramowaną pozycję docelową.

Składnia

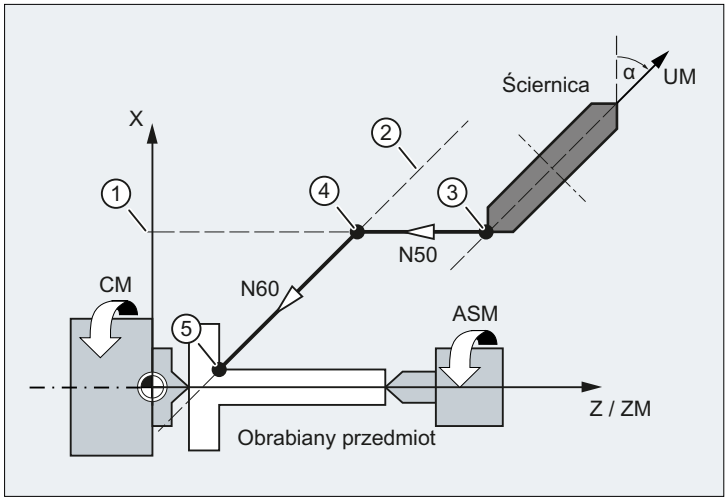
```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

Znaczenie

`G7`:	Obliczenie oraz dosunięcie do pozycji startowej wcinania skośnego
`G5`:	Ruch w osiach skośnych w kierunku zaprogramowanej pozycji końcowej

<Endpos_X>:	Pozycja końcowa osi X
<Endpos_X>:	Pozycja końcowa osi Z

Przykład



① Równolegle do osi Z w odstępnie od aktualnej pozycji osi X
② Równolegle do osi skośnej poprzez zaprogramowaną pozycję końcową
③ Pozycja wyjściowa
④ Wcinanie: Pozycja startowa
⑤ Wcinanie: Pozycja końcowa
X oś geometryczna
Z Oś geometryczna
ZM Oś maszyny
UM Oś maszyny

Kod programu	**Komentarz**
N... G18	; Wybór płaszczyzny XZ.
N40 TRAANG(45,0)	; Włączenie transformacji TRAANG , kąt = 45°.
N50 G7 X40 Z70 F4000	; Obliczenie oraz dosunięcie do pozycji startowej.
N60 G5 X40 F100	; Ruch w osi skośnej do pozycji końcowej.
N70 …	

7.9 Włączenie transformacji powiązanej (TRACON)

Transformacja powiązana jest włączana w programie obróbki albo akcji synchronicznej instrukcją `TRACON`.

Składnia

```
TRACON(<Trafo_Nr>,<Par_1>,<Par_2,...)
...
TRAFOOF
```

Znaczenie

`TRACON`:	Załączenie transformacji powiązanej Aktywna wcześniej inna transformacja jest przez `TRACON()` pośrednio wyłączana.		
`<Trafo_Nr>`:	Numer powiązanej transformacji		
	Typ:	INT	
	Zakres wartości:	0 ... 2	
	Wartość:	0, 1	pierwsza/jedyna transformacja powiązana
		2	druga transformacja powiązana
		brak wprowadzenia	równoznaczny z 0 wzgl. 1
		Wskazówka: Wartości nierówne 0, 1, 2 wytwarzają komunikat błędu.	
`<Par_1>,<Par_2,...`:	Parametry dla powiązanych ze sobą transformacji (np. kąt osi skośnej) Przy nie ustawionych parametrach działają ustawienia domyślne albo parametry ostatnio używane. Przez podanie przecinków należy zadbać o to, by podane parametry były poddawane ocenie w takiej kolejności, w jakiej są oczekiwane, jeżeli dla przedtem umieszczonych parametrów mają działać ustawienia domyślne. W szczególności przy podaniu co najmniej jednego parametru musi przed nim znajdować się przecinek, również gdy podanie `<Trafo_Nr>` nie jest konieczne, a więc na przykład `TRACON( , 3.7)`. W przypadku TRACON z Transmit lub TRAORI (obróbka 5-osiowa) drugi parametr Par_2 nie działa jako kąt dla osi skośnej. W przypadku TRACON z TRACYL (obróbka powierzchni bocznej) stosowany jest Par dla średnicy jednostkowej.		
`TRAFOOF`:	Wyłączenie ostatnio włączonej (powiązanej) transformacji.		

Przykład

Kod programu	**Komentarz**
`...`	

Kod programu	**Komentarz**
N230 TRACON(1,45.)	; Włączenie pierwszej transformacji powiązanej.
	; Wybór aktywnej wcześniej transformacji jest automatycznie cofany.
	; Parametrem dla osi skośnej jest 45°.
...	
N330 TRACON(2,40.)	; Włączenie drugiej transformacji powiązanej.
	; Parametrem dla osi skośnej jest 40°.
...	
N380 TRAFOOF	; Wyłączenie drugiej transformacji powiązanej.
...	

7.10 Ruch kartezjański PTP

7.10.1 Włączenie/Wyłączenie ruchu kartezjańskiego PTP (PTP, PTPG0, PTPWOC, CP)

Ruch kartezjański z punktu do punktu lub ruch PTP włącza się i wyłącza w programie NC za pomocą poleceń z grupy nr G49.

Polecenia działają modalnie. Ustawienie domyślne jest to przejazd z ruchem kartezjańskim po torze (CP).

Inaczej niż w przypadku CP podczas aktywnego ruchu PTP wykonywana jest tylko transformacja kinematyczna punktu docelowego, a osie maszyny wykonują ruch synchroniczny.

Aby kartezjański punkt docelowy można było jednoznacznie przeliczyć w wartościach osi maszyny potrzebne są dodatkowe informacje dotyczące pozycji i parametrów kąta, które charakteryzują położenie osi. Dane te podawane są za pomocą ustawialnych adresów STAT (Strona 387) i TU (Strona 391).

Warunek

Transformacja TRAORI, TRANSMIT, RCTRA lub ROBX jest aktywna.

Składnia

```
PTP / PTPG0 / PTPWOC
...
CP
```

Znaczenie

PTP:	Włączenie ruchu punkt do punktu PTP Najazd na zaprogramowaną pozycję kartezjańską w blokach G0 i G1 następuje za pomocą ruchu osi synchronicznych.
PTPG0:	Włączenie ruchu punkt do punktu PTPG0 Najazd na zaprogramowaną pozycję kartezjańską za pomocą ruchu osi synchronicznych tylko w G0. W G1 nastąpi przełączenie na ruch po torze CP.
PTPWOC:	Włączenie ruchu punkt do punktu PTPWOC (możliwy tylko przy aktywnej transformacji orientacji) Tak jak w przypadku PTP, jednak bez ruchów wyrównawczych, które są wywoływane przez ruchy osi obrotowych i osi orientacji.
CP:	Wyłączenie ruchu punkt do punktu i włączenie ruchu po torze CP Za pomocą CP realizowany jest ruch kartezjański po torze.

Uwaga
PTPWOC

W połączeniu z transformacją RCTRA lub ROBX zastosowanie PTPWOC nie ma znaczenia!

Przykłady

Patrz:

- Przykład 1: Ruch PTP robota 6-osiowego z transformacją ROBX (Strona 394)
- Przykład 2: Ruch PTP przy rodzajowej transformacji 5-osiowej (Strona 395)
- Przykład 1: Ruch PTP robota 6-osiowego z transformacją ROBX (Strona 394)

Patrz również

Przykład 3: PTPG0 i TRANSMIT (Strona 396)

7.10.2 Podanie położenia przegubów (STAT)

Podanie pozycji we współrzędnych kartezjańskich i danych orientacji narzędzia nie jest wystarczające, aby precyzyjnie określić ustawienia maszyny, ponieważ przy takiej samej orientacji narzędzia możliwych jest wiele położeń przegubów. W zależności od rodzaju kinematyki, istnieje do 8 różnych położeń przegubu. Położenia przegubu są specyficzne dla transformacji.

Aby uniknąć dwuznaczności położenie przegubów jest podane pod adresem STAT.

Uwaga

Sterowanie uwzględnia zaprogramowane wartości STAT tylko przy ruchach PTP. Przy ruchach CP są one ignorowane, ponieważ podczas przejazdu z aktywną transformacją zmiana położenia jest niemożliwa. Przy przejeździe z aktywnym CP położenie dla punktu docelowego jest przejmowane z punktu startowego.

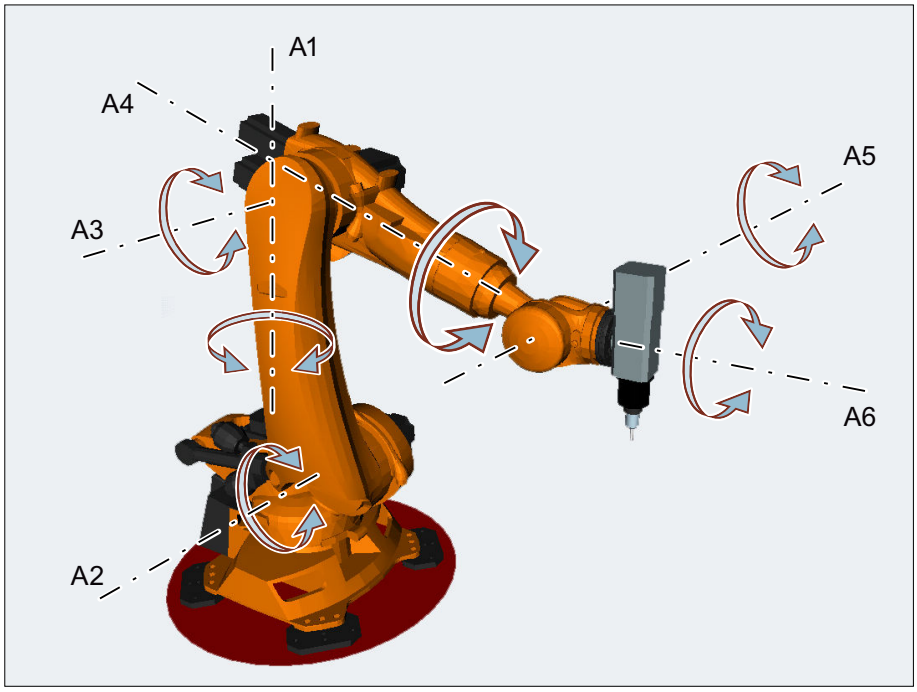
Składnia

```
STAT=<Wartość>
```

Znaczenie

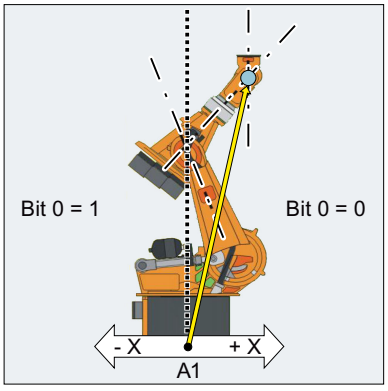
`STAT=`:	Ustawialny adres do podania położenia przegubów
`<Wartość>`:	Wartość binarna lub dziesiętna Zawiera jeden bit dla każdego z możliwych położeń. Znaczenie bitu jest ustalane przez różne transformacje.

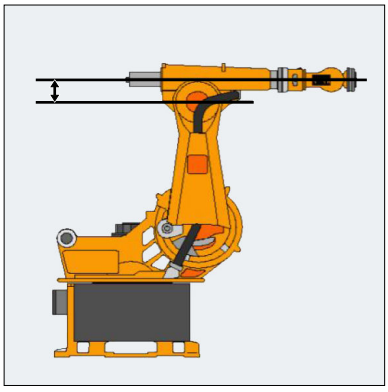
Zastosowanie STAT można zilustrować na przykładzie 6-osiowego robota z wrzecionem frezarskim. Transformacja kinematyczna powinna odbywać się za pomocą transformacji robota ROBX (warunek: cykl kompilowany "RMCC/ROBX rozszerzona transformacja robotyki" jest załadowany i aktywny).



Osie A1, A2 i A3 są osiami głównymi robota. Za pomocą osi głównych oznaczane są osie A4, A5 i A6 jako osie głowicy lub ramienia i pozycjonowane w przestrzeni roboczej. Za pomocą dodatkowych ruchów osi ramienia wrzeciono frezarskie może być tak orientowane w przestrzeni, jak jest to wymagane przy obróbce. Przy takiej samej orientacja narzędzia możliwe są różne położenia przegubu.

Wybór położenia przegubu do zastosowania w obróbce następuje przez zaprogramowanie bitu 0 ... 2 ustawialnego adresu STAT:

<table>
<tr><td>Bit 0</td><td colspan="2">A4 A5 A6</td></tr>
<tr><td rowspan="2"></td><td>= 0</td><td>Obszar podstawowy (Shoulder Right)
Robot znajduje się w obszarze podstawowym, gdy wartość X punktu przecięcia osi w odniesieniu do układu współrzędnych A1, jest dodatnia.</td></tr>
<tr><td>= 1</td><td>Obszar nad głową (Shoulder Left)
Robot znajduje się w obszarze nad głową, gdy wartość X punktu przecięcia osi w odniesieniu do układu współrzędnych A1, jest ujemna.</td></tr>
<tr><td colspan="3"> 

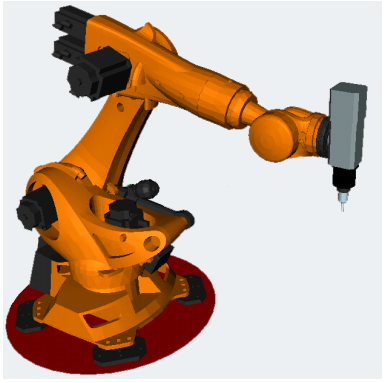
Przykład: Punkt przecięcia osi ramienia znajduje się w obszarze podstawowym</td></tr>
<tr><td>Bit 1</td><td colspan="2">Położenie osi 3
Kąt, w którym zmienia się wartość bitu 1 zależy od typu robota. Dla robota, którego osie 3 i 4 przecinają się obowiązuje:</td></tr>
<tr><td></td><td>= 0</td><td>A3 < 0° (Elbow Down)</td></tr>
<tr><td></td><td>= 1</td><td>A3 ≥ 0° (Elbow Up)</td></tr>
<tr><td></td><td colspan="2">Uwaga:
W przypadku robotów z offsetem między osią 3 i 4 jest kąt, przy którym zmienia się wartość bitu 1 w zależności od wielkości offsetu.
 
Offset między A3 i A4</td></tr>
<tr><td>Bit 2</td><td colspan="2">Położenie osi 5</td></tr>
<tr><td></td><td>= 0</td><td>A5 ≤ 0° (No Handflip)</td></tr>
<tr><td></td><td>= 1</td><td>A5 > 0° (Handflip)</td></tr>
</table>

Przykład programu:

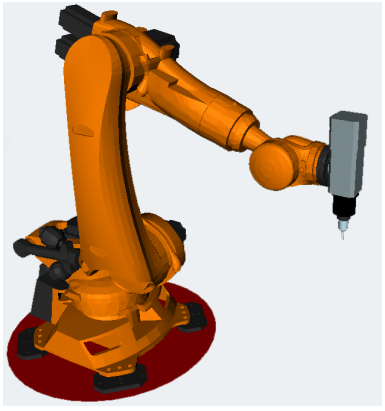
Kod programu	**Komentarz**
...	
N14 T="T8MILLD20" D1	; $TC_DP3[1,1]=132.95
N16 ORIMKS	

Kod programu	**Komentarz**
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 **STAT=...** F2000	; Wartość STAT określa położenie przegubu
...	

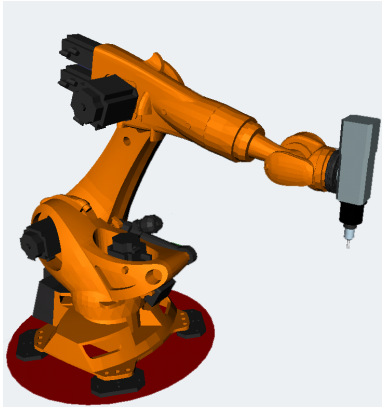
STAT=1 ('B001') → Shoulder Left
→ Elbow Down
→ No Handflip



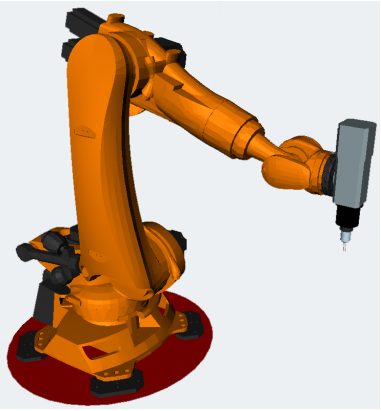
STAT=2 ('B010') → Shoulder Right
→ Elbow Up
→ No Handflip



STAT=5 ('B101') → Shoulder Left
→ Elbow Down
→ Handflip



`STAT=6 ('B110')` → Shoulder Right
→ Elbow Up
→ Handflip



TRANSMIT

Aby uniknąć dwuznaczności w odniesieniu do biegunów, przy TRANSMIT wykorzystuje się adres STAT.

Jeśli oś obrotowa musi obróci się o 180º lub kontur przy CP jest wykonywany przez biegun obowiązuje:

Bit 0	Dotyczy tylko $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 lub 2:	
	= 0	Oś obrotowa wykonuje ruch o +180º lub obraca się zgodnie z ruchem wskazówek zegara.
	= 1	Oś obrotowa wykonuje ruch o -180º lub obraca się przeciwnie do ruchu wskazówek zegara.
Bit 1	Dotyczy tylko $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0:	
	= 0	Wykonywanie przez biegun. Oś obrotowa nie obraca się.
	= 1	Obrót wokół bieguna. Przy tym istotny jest bit 0 STAT.

7.10.3 Podanie znaku kąta osi (TU)

W przypadku osi obrotowych kąt osi, który jest większy niż +180° lub mniejszy niż -180°, aby wykonać ruch bez określania szczególnej strategii przejazdu (np. punkty pośrednie) należy pod ustawialnym adresem TU podać znaki kąta osi.

Uwaga

Sterowanie uwzględnia zaprogramowane wartości TU tylko przy ruchach PTP. Przy ruchach CP są one ignorowane.

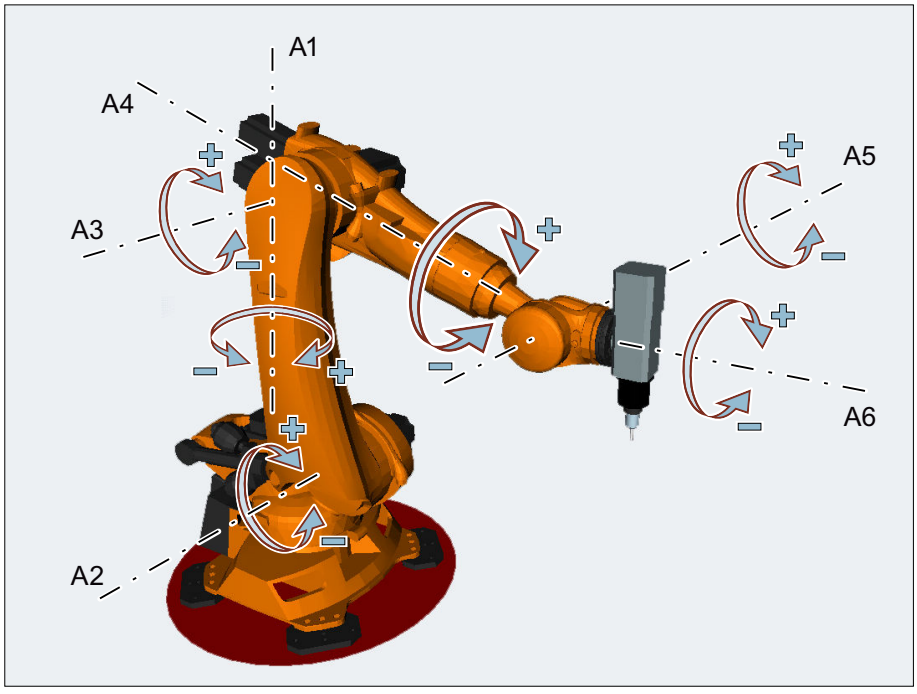
Składnia

```
TU=<Wartość>
```

Znaczenie

TU:	Ustawialny adres do podania znaków kąta osi			
<Wartość>:	Wartość binarna lub dziesiętna Zmienna TU zawiera dla każdej osi, która wchodzi do transformacji, jeden bit, który sygnalizuje kierunek ruchu.			
	Bit	= 0	Znak kąta osi: +	Zakres kąta osi: 0° ≤ θ < 360°
		= 1	Znak kąta osi: -	Zakres kąta osi: - 360° < θ < 0°

Przykład: Robot 6-osiowy



Bit	Znaczenie	Wartość	Znak kąta osi	Kąt osi
Bit 0 [1)]	Znak dla kąta osi A1	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 1 [1)]	Znak dla kąta osi A2	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 2 [1)]	Znak dla kąta osi A3	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 3 [1)]	Znak dla kąta osi A4	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 4 [1)]	Znak dla kąta osi A5	= 0	+	≥ 0°
		= 1	-	< 0°

Bit	Znaczenie	Wartość	Znak kąta osi	Kąt osi
Bit 5 [1)]	Znak dla kąta osi A6	= 0	+	≥ 0°
		= 1	-	< 0°

[1)] Rzeczywisty numer bitu TU równa się numerowi osi kanału osi robota! W przykładzie z osiami robota (A1 do A6) jest sześć pierwszych osi w kanale, następnie będą zastosowane bity TU 0 ...5. W przypadku innego przyporządkowania osi kanału w osiach robota zmieniłyby się odpowiednio numery bitów TU osi robota (np.: osiami robota są osie kanału od 3 do 8 tzn. bity TU 2 ... 7 zostaną zastosowane dla osi robota).

TU=19 (odpowiada TU='B010011) oznaczałoby to:

Bit	Wartość		Kąt osi
0	= 1	○	$\theta_{A1} < 0°$
1	= 1	○	$\theta_{A2} < 0°$
2	= 0	○	$\theta_{A3} \geq 0°$
3	= 0	○	$\theta_{A4} \geq 0°$
4	= 1	○	$\theta_{A5} < 0°$
5	= 0	○	$\theta_{A6} \geq 0°$

Uwaga

W przypadku osi z zakresem ruchu > ±360° przejazd odbywa się zawsze najkrótszą drogą, ponieważ położenie osi nie jest wystarczająco precyzyjnie określone w informacjach TU.

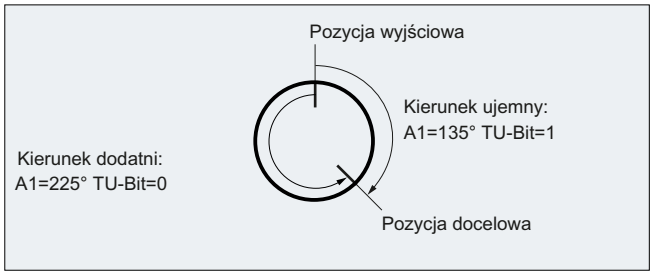
Jeżeli w przypadku jednej pozycji nie zaprogramowano żadnego TU, w zależności od MD30455 $MA_MISC_FUNCTION_MASK jest wykonywany krótszy lub dłuższy ruch (patrz rozdział „Uwzględnienie limitów SW przy ruchach PTP„ w Podręczniku działania, Funkcje rozszerzone).

TRANSMIT

W przypadku ruchów PTP przy aktywnym TRANSMIT adres TU nie jest istotny.

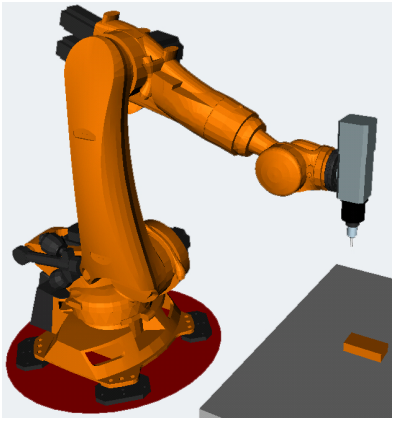
Przykład

W podanych na poniższym rysunku położenia osi obrotowej, ruch może nastąpić w kierunku ujemnym albo dodatnim. Pod adresem A1 jest programowana jest pozycja kąta. Dopiero dzięki danym TU kierunek ruchu staje się jednoznaczny.



7.10.4 Przykład 1: Ruch PTP robota 6-osiowego z transformacją ROBX

W przykładzie poniżej przedstawiono kartezjański ruch PTP i związane z nim polecenia NC.



Rysunek 7-3 Robot 6-osiowy z wrzecionem frezarskim

```
N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
;$P_UIFR[1]=CTRANS(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 Przykład 2: Ruch PTP przy rodzajowej transformacji 5-osiowej

Założenie: Podstawą jest prostokątna kinematyka CA.

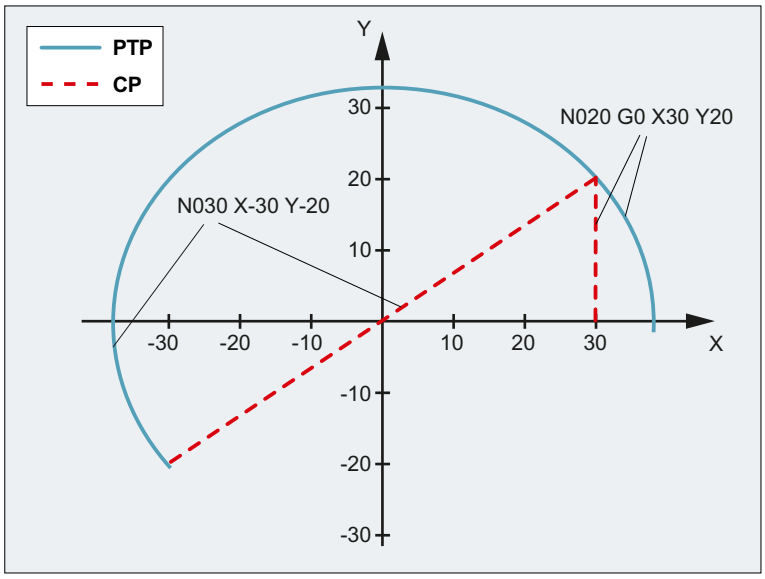
Kod programu	Komentarz
TRAORI	; Transformacja kinematyki CA wł.
PTP	; Włączenie ruchu PTP
N10 A3=0 B3=0 C3=1	; Pozycje osi obrotowych C=0 A=0
N20 A3=1 B3=0 C3=1	; Pozycje osi obrotowych C=90 A=45
N30 A3=1 B3=0 C3=0	; Pozycje osi obrotowych C=90 A=90
N40 A3=1 B3=0 C3=1 **STAT=1**	; Pozycje osi obrotowych C=270 A=-45

Wybór jednoznaczny ruchu do pozycji osi obrotowej:

W bloku N40 osie obrotowe w wyniku zaprogramowania **STAT = 1** wykonują dłuższą drogę od swojego punktu startowego (C=90, A=90) do punktu końcowego (C=270, A=-45). Natomiast w przypadku **STAT =0** osie obrotowe wykonałyby krótszą drogę do punktu końcowego (C=90, A=45).

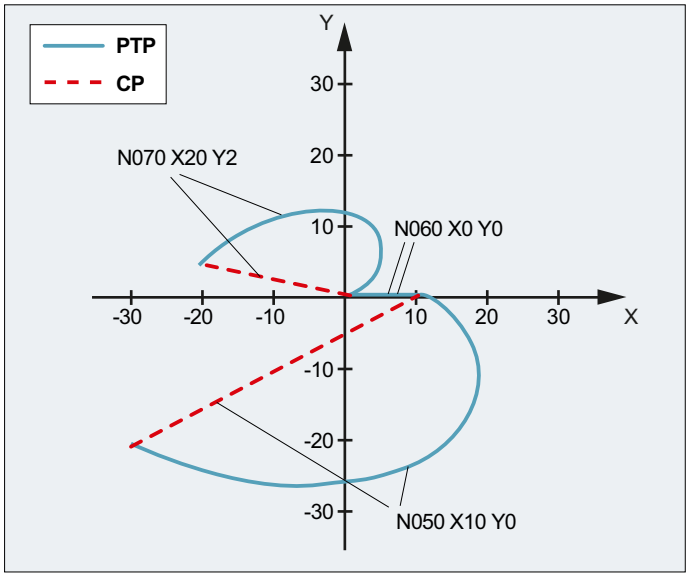
7.10.6 Przykład 3: PTPG0 i TRANSMIT

Obejście bieguna za pomocą PTPG0 i TRANSMIT



Kod programu	**Komentarz**
N001 G0 X30 Z0 F10000 T1 D1 G90	; Położenie wyjściowe, wymiar absolutny
N002 SPOS=0	
N003 **TRANSMIT**	; Transformacja TRANSMIT
N010 **PTPG0**	; Do każdego bloku G0 automatycznie PTP, a następnie ponownie CP
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Wyjście z bieguna przy pomocy PTP i TRANSMIT



Programowanie	**Komentarz**
N001 G0 X90 Z0 F10000 T1 D1 G90	; Położenie wyjściowe
N002 SPOS=0	
N003 **TRANSMIT**	; Transformacja TRANSMIT
N010 **PTPG0**	; Do każdego bloku G0 automatycznie PTP, a następnie ponownie 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 Warunki brzegowe przy wyborze transformacji

Funkcja

Wybór transformacji jest możliwy przez program obróbki lub MDA. Należy przy tym uwzględnić:

- Nie jest wstawiany blok pośredni ruchu (fazy/zaokrąglenia).
- Ciąg bloków Spline musi być zakończony; jeżeli nie, ukazuje się komunikat.
- Musi być cofnięty wybór korekcji dokładnej narzędzia (FTOCOF); jeżeli nie, ukazuje się komunikat.
- Musi być cofnięty wybór korekcji promienia narzędzia (G40); jeżeli nie, ukazuje się komunikat.
- Uaktywniona korekcja długości narzędzia jest przejmowana przez sterowanie do transformacji.
- Sterowanie cofa wybór aktualnego frame, działającego przed transformacją.
- Aktywne ograniczenie obszaru pracy jest dla osi objętych transformacją cofane przez sterowanie (odpowiada WALIMOF).
- Nadzór obszaru ochrony jest cofany.
- Praca z przechodzeniem płynnym i ścinanie narożników są przerywane.
- Wszystkie osie podane w danej maszynowej muszą być zsynchronizowane w odniesieniu do bloku.
- Zamienione osie są zamieniane z powrotem, jeżeli nie, ukazuje się komunikat.
- W przypadku osi zależnych jest wyprowadzany komunikat.

Wymiana narzędzia

Wymiana narzędzia jest dopuszczalna tylko przy cofniętym wyborze korekcji promienia narzędzia.

Zmiana korekcji długości narzędzia i wybór/cofnięcie wyboru korekcji promienia narzędzia nie mogą być zaprogramowane w tym samym bloku.

Zmiana frame

Wszystkie instrukcje, które odnoszą się tylko do bazowego układu współrzędnych, są dozwolone (FRAME, korekcja promienia narzędzia). Zmiana frame w przypadku G91 (wymiar przyrostowy) nie jest jednak - inaczej niż przy nie aktywnej transformacji - traktowana oddzielnie. Będący do przebycia przyrost jest poddawany ocenie w układzie współrzędnych obrabianego przedmiotu - niezależnie od tego, jaki frame działał w poprzedzającym bloku.

Wykluczenia

Osie objęte transformacją nie mogą być stosowane:

- jako oś Preset (alarm),
- dla dosunięcia do punktu stałego (alarm),
- dla bazowania (alarm).

7.12 Cofnięcie wyboru transformacji (TRAFOOF)

Przy pomocy wstępnie zdefiniowanego polecenia TRAFOOF wyłączane są wszystkie aktywne transformacje i frame.

Uwaga

Dla cofnięcia wyboru transformacji obowiązują te same warunki brzegowe (Strona 398) jak przy wyborze.

Aktywność później potrzebnych frame musi zostać włączona przez ponowne zaprogramowanie.

Składnia

```
...
TRAFOOF
```

Znaczenie

TRAFOOF:	Wyłączenie wszystkich aktywnych transformacji/frame

Łańcuchy kinematyczne

8

8.1 Kasowanie komponentów (DELOBJ)

Funkcja `DELOBJ()` „kasuje" komponenty za pomocą zresetowania przyporządkowanych zmiennych systemowych do ich wartości domyślnych:

- Elementy łańcuchów kinematycznych
- Obszary ochrony, elementy obszaru ochrony i pary kolizyjne
- Dane transformacji

Składnia

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

Znaczenie

DELOBJ	Usuwanie elementów łańcuchów kinematycznych, obszarów ochrony oraz elementów obszarów ochrony, par kolizyjnych i danych transformacji	
<CompType>:	Rodzaj komponentów do usunięcia	
	Typ danych:	STRING
	Wartość: "KIN_CHAIN_ELEM" **znaczenie**: Zmienne systemowe wszystkich elementów kinematycznych: $NK_...	
	Wartość: "KIN_CHAIN_SWITCH" **znaczenie**: Zmienna systemowa $NK_SWITCH[<i>]	
	Wartość: "KIN_CHAIN_ALL" **Znaczenie:** Wszystkie elementy i łączniki kinematyczne Równoznacznie z kolejnym wywołaniem DELOBJ z "KIN_CHAIN_ELEM" i "KIN_CHAIN_SWITCH"	
	Wartość: "PROT_AREA" **Znaczenie**: Zmienne systemowe obszarów ochrony: • $NP_PROT_NAME • $NP_CHAIN_NAME • $NP_CHAIN_ELEM • $NP_1ST_PROT	
	Wartość: "PROT_AREA_ELEM" **Znaczenie**: Zmienne systemowe elementów obszaru ochrony dla obszarów ochrony maszyny i/lub automatycznych obszarów ochrony narzędzi: • $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	
	Wartość: "PROT_AREA_COLL_PAIRS" **Znaczenie**: Zmienne systemowe par kolizyjnych • $NP_COLL_PAIR • $NP_SAFETY_DIST	
	Wartość: "PROT_AREA_ALL" **Znaczenie**: Wszystkie obszary ochrony, elementy obszaru ochrony i pary kolizyjne (zmienne systemowe $NP_...) Równoznacznie z kolejnym wywołaniem DELOBJ z "PROT_AREA", "PROT_AREA_ELEM" i "PROT_AREA_COLL_PAIRS"	
	Wartość: "TRAFO_DATA" **Znaczenie**: Zmienne systemowe wszystkich transformacji $NT_...	

<Index1>:	Indeks (pierwszych) usuwanych komponentów (**opcjonalnie**)	
	Typ danych:	INT
	Wartość domyślna:	-1
	Zakres wartości:	-1 ≤ x ≤ (maksymalna liczba zaprojektowanych komponentów -1)
	Wartość	**Znaczenie**
	0, 1, 2	Indeks komponentów do usunięcia.
	-1	Wszystkie komponenty podanego typu zostaną usunięte. <Index2> nie jest uwzględniany.
<Index2>:	Indeks ostaniego z usuwanych komponentów (**opcjonalnie**) Jeśli nie zaprogramowano <Index2> , usunięte zostaną tylko zmienne systemowe bazowanych w <Index1> komponentów	
	Typ danych:	INT
	Wartość domyślna:	Tylko zmienne systemowe komponentów bazowanych w <Index1> zostaną usunięte.
	Zakres wartości:	<Index1> < x ≤ (maksymalna liczba zaprojektowanych komponentów -1)
<NoAlarm>:	Wyłączenie alarmu (**opcjonalnie**)	
	Typ danych:	BOOL
	Wartość domyślna:	FALSE
	Wartość	**Znaczenie**
	FALSE	W przypadku błędu (<RetVal> < 0) obróbka programowa zostanie zatrzymana i wyświetli się alarm.
	TRUE	W przypadku błędu obróbka programu nie zostanie zatrzymana i nie wyświetli się żaden alarm. Przypadek zastosowania: Specyficzna dla użytkownika reakcja odpowiednio do wartości zwrotnej.
<RetVal>:	Wartość zwrotu funkcji	
	Typ danych:	INT
	Zakres wartości:	0, -1, -2, ... -7
	Wartość	**Znaczenie**
	0	Żaden błąd nie wystąpił.
	-1	Wywołanie funkcji bez parametrów. Minimum jeden parametr <CompType> musi zostać podany.
	-2	<CompType> oznacza nieznane komponenty
	-3	<Index1> jest mniejszy niż -1
	-4	<Index1> jest większy niż zaprojektowana liczba komponentów
	-5	<Index1> przy usuwaniu **grupy** komponentów ma wartość różną od -1
	-6	<Index2> jest mniejszy niż <Index1>
	-7	<Index2> jest większy niż zaprojektowana liczba komponentów

8.2 Określanie indeksu poprzez nazwy (NAMETOINT)

W polach zmiennych systemowych typu STRING wprowadzono specyficzne dla użytkownika nazwy. Na podstawie oznaczenia zmiennych systemowych i nazw funkcja `NAMETOINT()` określa się wartość indeksu przynależnego do nazw, pod którym jest zapisany w zmiennych systemowych.

Składnia

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

Znaczenie

`NAMETOINT`:	Określanie indeksu zmiennych systemowych	
`<SysVar>`:	Nazwa pola zmiennych systemowych typu STRING	
	Typ danych:	STRING
	Zakres wartości:	Nazwy wszystkich pól zmiennych systemowych NC typu STRING
`<Name>`:	Łańcuch znaków lub nazwa, dla których powinien zostać określony indeks zmiennych systemowych.	
	Typ danych:	STRING
`<NoAlarm>`:	Wyłączenie alarmu (**opcjonalnie**)	
	Typ danych:	BOOL
	Wartość domyślna:	FALSE
	Wartość	**Znaczenie**
	TRUE	W przypadku błędu obróbka programu nie zostanie zatrzymana i nie wyświetli się żaden alarm. Przypadek zastosowania: Specyficzna dla użytkownika reakcja odpowiednio do wartości zwrotnej.
	FALSE	W przypadku błędu (<RetVal> < 0) obróbka programu zostanie zatrzymana i wyświetli się alarm.
`<RetVal>`:	Indeks zmiennych systemowych albo błąd	
	Typ danych:	INT
	Zakres wartości:	-1 ≤ x ≤ (maks. liczba zaprojektowanych komponentów -1)
	Wartość	**Znaczenie**
	≥ 0	Szukana nazwa została znaleziona w podanym indeksie zmiennych systemowych.
	-1	Nie znaleziono szukanej nazwy lub wystąpił błąd.

Przykład

Kod programu	**Komentarz**
`DEF INT INDEX`	
`$NP_PROT_NAME[27] = „Pokrycie"`	

Kod programu	Komentarz
...	
INDEX = NAMETOINT(„$NP_PROT_NAME”,”Pokrycie")	; Indeks == 27

Uniknięcie kolizji za pomocą łańcuchów kinematycznych

9

Uwaga

Obszary ochrony

Podane w następnych rozdziałach obszary ochrony odnoszą się do funkcji "Geometryczne modelowanie maszyny"

Literatur:
Podręcznik działania Funkcje specjalne, Rozdział "Geometryczne modelowanie maszyny"

9.1 Sprawdzenie pod kątem pary kolizyjnej (COLLPAIR)

Funkcja `COLLPAIR(...)` określa, czy dwa obszary ochrony tworzą parę kolizyjną.

Składnia

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>)])
```

Znaczenie

`COLLPAIR`:	Kontrola pod kątem przynależności do pary kolizyjnej		
`<RetVal>`:	Zwracana wartość funkcji		
	Typ danych:	INT	
	Wartość:	≥ 0	Oba obszary ochronne tworzą jedną parę kolizyjną. Wartość zwrotna == Indeks pary kolizyjnej m (patrz $NP_COLL_PAIR)
		-1	Albo podano mniej niż dwa łańcuchy, albo minimum jeden z nich stanowi łańcuch zerowy.
		-2	Podany w pierwszym parametrze obszar ochrony nie został znaleziony.
		-3	Podany w drugim parametrze obszar ochrony nie został znaleziony.
		-4	Żaden z obu podanych obszarów ochrony nie został znaleziony.
		-5	Oba podane obszary ochrony zostały znalezione, ale nie tworzą one jednej pary kolizyjnej.
`<Name_1>`:	Nazwa pierwszego obszaru ochrony		
	Typ danych:	STRING	
	Zakres wartości:	Sparametryzowane nazwy obszarów ochrony	
`<Name_2>`:	Nazwa drugiego obszaru ochrony		
	Typ danych:	STRING	
	Zakres wartości:	Sparametryzowane nazwy obszarów ochrony	
`<NoAlarm>`:	Wyłączenie alarmu **(opcjonalnie)**		
	Typ danych:	BOOL	
	Wartość:	FALSE (domyślnie)	W przypadku błędu (<RetVal> < 0) program obróbki zostanie zatrzymany i wyświetli się alarm.
		TRUE	W przypadku błędu program obróbki nie zostanie zatrzymany i nie wyświetli się żaden alarm. Przypadek zastosowania: Specyficzna dla użytkownika reakcja odpowiednio do wartości zwrotnej

9.2 Żądanie ponowne obliczenie modelu maszyny dla unikania kolizji (PROTA):

Jeżeli zmienne systemowe łańcucha kinematycznego $NK_..., geometrycznego modelowania maszyny lub unikania kolizji $NP_... zostały zapisane w programie obróbki, to należy następnie wywołać procedurę `PROTA`, aby zmiana w wewnętrznym modelu maszyny unikania kolizji była skuteczna.

Składnia

```
PROTA[(<Par>)]
```

Znaczenie

`PROTA`:	Żądanie ponowne obliczenie modelu maszyny unikania kolizji		
	• Wyzwala zatrzymanie przebiegu wyprzedzającego. • Musi znajdować się w oddzielnym bloku.		
`<Par>`:	Parametry (**opcjonalnie**)		
	Typ danych:	STRING	
	Wartość:	---	Bez parametru. Przeprowadzane jest ponowne obliczenie modelu maszyny. Stany stref ochrony zostaną zachowane.
		"R"	Przeprowadzane jest ponowne obliczenie modelu maszyny. Obszary ochrony zostaną przesunięte do statusu inicjalizacji odpowiednio do $NP_INIT_STAT.

Warunki brzegowe

Symulacja

Procedura `PROTA` nie może być stosowana w programach obróbki w związku z symulacją (simNC).

Przykład: Unikanie wywołania `PROTA`, podczas gdy symulacja jest aktywna.

```
Kod programu                    Komentarz
...
IF $P_SIM == FALSE              ; IF Symulacja nieaktywna
   PROTA                        ; THEN Nowe obliczenie modelu kolizji
ENDIF                           ; ENDIF
...
```

Patrz również

Ustawienie stanu obszaru ochrony (PROTS) (Strona 410)

9.3 Ustawienie stanu obszaru ochrony (PROTS)

Procedura `PROTS(...)` ustawia status obszarów ochrony na podane wartości.

Składnia

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

Znaczenie

`PROTS`:	Ustawia status obszarów ochrony		
	• Musi znajdować się w oddzielnym bloku.		
`<State>`:	Status, w jakim należy ustawić podane obszary ochrony		
	Typ danych:	CHAR	
	Wartość:	"A" lub "a"	Status: Aktywny
		"I" lub "i"	Status: Nieaktywny
		"P" lub "p"	Status: Wstępnie aktywowane lub sterowane przez PLC [1]
		"R" lub "r"	Status: Wartość wewnętrzna NC statusu inicjalizacyjnego [2]
`<Name_1> ... <Name_n>`:	Nazwa jednego lub kilku obszarów ochrony, jakie należy ustawić dla podanego statusu (**opcjonalnie**) Jeżeli nie podano nazwy, podany status zostanie ustawiony dla wszystkich zdefiniowanych obszarów ochrony.		
	Typ danych:	STRING	
	Zakres wartości:	Sparametryzowane nazwy obszarów ochrony	
	Wskazówka: Maksymalna liczba obszarów ochrony, które mogą zostać podane jako parametry jest zależna tylko od maksymalnej możliwej liczby znaków przypadających na wiersz programu.		
[1] Aktywacja / Dezaktywacja następuje za pomocą: DB10.DBX234.0 - DBX241.7 [2] Status jest ustawiany na wewnętrzną wartość statusu inicjalizacji NC, tzn. na wartość, którą miała zmienna systemowa $NP_INIT_STAT w chwili ostatniego wywołania PROTA(...) (Strona 409).			

9.4 Określenie odległości dwóch obszarów ochrony (PROTD)

Funkcja `PROTD(...)` oblicza odstęp od dwóch obszarów ochrony.

Właściwości funkcji:

- Obliczenie odległości następuje niezależnie od statusu obszarów ochrony (aktywny, nieaktywny, wstępnie aktywowany).
- Do obliczenia odległości dwóch obszarów ochrony stosuje się tylko takie elementy obszaru ochrony, które są oznaczone za pomocą $NP_USAGE = "C" lub "A”. Elementy obszaru ochrony oznaczone za pomocą $NP_USAGE = "V” nie są brane pod uwagę.
- Obszary ochrony, w przypadku których wszystkie elementy obszaru ochrony oznaczone są za pomocą $NP_USAGE = "V” nie mogą być zastosowane do obliczenia odległości.
- Obliczenie odległości następuje za pomocą pozycji obowiązujących na końcu bloku poprzedzającego.
- Nałożenia, które zostaną uwzględnione w przebiegu głównym, np. przesunięcie DRF albo zewnętrzne przesunięcia punktu zerowego zapisują się wraz z wartościami obowiązującymi do **momentu interpretacji** funkcji w obliczeniu odległości.

Uwaga

Synchronizacja

Przy stosowaniu funkcji `PROTD(...)` wyłącznie w gestii użytkownika leży synchronizacja przebiegu głównego oraz wyprzedzającego wykonana ewentualnie za pomocą zatrzymania przebiegu wyprzedzającego `STOPRE`.

Kolizja

Jeśli zachodzi kolizja pomiędzy podanymi obszarami ochrony, funkcja podaje odległość 0,0. Kolizja występuje, gdy oba obszary ochrony stykają się lub przenikają.

Odstęp bezpieczeństwa dla kontroli kolizji (MD10622 $MN_COLLISION_SAFETY_DIST) nie jest uwzględniany podczas obliczania odległości.

Składnia

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

Znaczenie

`PROTD`:	Oblicza odległość dwóch podanych obszarów ochrony	
	• Musi znajdować się w oddzielnym bloku.	
`<RetVal>`:	Zwracana wartość funkcji: Wartość absolutna odległości obu obszarów ochrony lub 0,0 w przypadku kolizji (patrz wyżej: akapit Kolizja)	
	Typ danych:	REAL
	Zakres wartości:	0,0 ≤ x ≤ +maks. Wartość REAL

<Name_1>, <Name_2>:	Nazwy obu obszarów ochrony, których wzajemna odległość ma zostać obliczona. (**opcjonalnie**)		
	Typ danych:	STRING	
	Zakres wartości:	Sparametryzowane nazwy obszarów ochrony	
	Wartość domyślna:	"" (pusty łańcuch znaków) Jeżeli nie podano żadnych obszarów ochrony, funkcja oblicza aktualną odległość najmniejszą ze wszystkich zawartych w modelu kolizji uaktywnionych bądź wstępnie aktywowanych obszarów ochrony.	
<Vector>:	Wartość zwrotna: 3-wymiarowy wektor odstępu od obszaru ochrony <Name_2> do obszaru ochrony <Name_1> z: • <Vector>[0]: Współrzędna X w układzie współrzędnych globalnych • <Vector>[1]: Współrzędna Y w układzie współrzędnych globalnych • <Vector>[2]: Współrzędna Z w układzie współrzędnych globalnych Przy kolizji: <Vector> == wektor zerowy		
	Typ danych:	VAR REAL [3]	
	Zakres wartości:	<Vector> [n]: 0,0 ≤ x ≤ +maks. wartość REAL	
<System>:	System miar (calowy/metryczny) dla odstępu i wektora odstępu (**opcjonalnie**)		
	Typ danych:	BOOL	
	Wartość:	FALSE (domyślnie)	System miar odpowiednio do aktualnie aktywnego polecenia z grupy G nr 13 (G70, G71, G700, G710).
		TRUE	System miar odpowiednio do ustawionego systemu podstawowego: MD52806 $MN_ISO_SCALING_SYSTEM

Transformacje za pomocą łańcuchów kinematycznych 10

10.1 Włączenie transformacji (TRAFOON)

Zdefiniowane transformacje za pomocą łańcuchów kinematycznych są aktywowane za pomocą predefiniowanej procedury TRAFOON. Wywołanie musi znajdować się w oddzielnym bloku.

Uwaga

Alternatywnie transformację zdefiniowaną za pomocą łańcuchów kinematyki można również aktywować za pomocą konwencjonalnych poleceń języka jak np. TRAORI lub TRANSMIT. W tym przypadku w zmiennej $NT_TRAFO_INDEX należy wprowadzić odpowiednią niezerową wartość.

Dodatkowe informacje do $NT_TRAFO_INDEX - patrz „Lista parametrów zmiennych systemowych".

Składnia

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

Znaczenie

TRAFOON:	Procedura aktywacji transformacji zdefiniowanej za pomocą łańcuchów kinematyki		
<Trafoname>:	Nazwa zestawu danych transformacji		
	Typ danych:	STRING	
	Zakres wartości:	Wszystkie nazwy zestawów danych transformacji zdefiniowanych za pomocą $NT_NAME	
	Uwaga: Nazwa zestawu danych transformacji musi być jednoznaczna. W $NT_NAME może wystąpić tylko jeden raz.		
<Diameter>:	Średnica odniesienia lub robocza (tylko TRACYL)		
	Typ danych:	REAL	
	Wartość musi być większa od 1.		
<k>:	Definiuje zastosowanie korekcji ścianki rowka (tylko TRACYL).		
	Typ danych:	BOOL	
	Wartość:	FALSE	Bez korekcji ścianki rowka
		TRUE	Z korekcją ścianki rowka
	Odpowiada transformacji typu TRACYL 514 (programowalna korekcja ścianki rowka). Jeżeli nie podano <k>, działa sparametryzowane ustawienie Bitu 10 w $NT_CNTRL[<n>].		

Przykład

Kod programu	Komentarz
`TRAFOON["Trans_1"]`	`Aktywuje transformację z nazwą Trans_1.`

10.2 Modyfikacja transformacji orientacji po pomiarze maszyny (CORRTRAFO)

W przypadku transformacji orientacji, które zostały zdefiniowane za pomocą łańcuchów kinematyki, użytkownik może zastosować predefiniowaną funkcję CORRTRAFO w celu modyfikacji wektorów offsetu lub wektorów kierunku osi orientacji modelu kinematycznego maszyny po pomiarze maszyny.

Składnia

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode> [,
<No_Alarm>])
```

Znaczenie

CORRTRAFO:	Wywołanie funkcji

<Corr_Status>:	Wartość zwrotna funkcji		
	Typ danych:	INT	
	Wartości:	0	Funkcja została wykonana bez błędu.
		1	Żadna transformacja nie jest aktywna.
		2	Aktualnie aktywna transformacja nie jest transformacją orientacji.
		3	Aktywna transformacja orientacji nie została zdefiniowana za pomocą łańcuchów kinematyki.
		10	Parametr wywołania <Corr_Index> jest ujemny.
		11	Parametr wywołania <Corr_Mode> jest ujemny.
		12	Niepoprawne odesłanie do łańcucha częściowego (miejsce jednostek w <Corr_Index>). Wartość nie może być większa niż liczba osi orientacji w łańcuchu częściowym.
		13	Niepoprawne odesłanie do osi orientacji łańcucha częściowego (miejsce jednostek w <Corr_Index>). Wartość musi być mniejsza niż liczba osi orientacji w łańcuchu częściowym.
		14	Niepoprawne odesłanie do łańcucha częściowego (miejsca dziesiątek w <Corr_Index>). Dozwolone są tylko wartości 0 i 1 (odesłanie do łańcucha Part lub Tool). Ten numer błędu występuje również wtedy, gdy łańcuch częściowy, do którego odsyła <Corr_Index> nie istnieje.
		15	W segmencie, do którego odwołuje się parametr <Corr_Index>, nie zdefiniowano żadnego elementu korekcyjnego ($NT_CORR_ELEM_P lub $NT_CORR_ELEM_T).
		20	Niepoprawny tryb korekcji (miejsce jednostek <Corr_Mode>). Dozwolone są tylko wartości 0 i 1.
		21	Niepoprawny tryb korekcji (miejsce dziesiątek i/albo setek w <Corr_Mode>). Przy zapisie kierunku osi tylko miejsce jednostek może wynosić zero.
		30	Miejsce setek w <Corr_Mode> jest niepoprawne. Dozwolone są tylko wartości 0 i 1.
		31	Miejsce tysięcy w <Corr_Mode> jest niepoprawne. Dozwolone są tylko wartości 0 i 1.
		40	Wektor kierunku, który ma zostać przejęty jako kierunek osi jest wektorem zerowym. Ten błąd może wystąpić tylko wtedy, gdy miejsce tysięcy <Corr_Mode> jest równe 0. Jeżeli miejsce tysięcy parametru jest równe 1 (nieaktywna kontrola maksymalnej korekcji) można również zapisać wektor zerowy.
		41	Przy korekcji wektora offsetu odchylenie względem aktualnych wartości jest w przynajmniej jednej współrzędnej większe niż wartość maksymalna zadana przez daną ustawczą SD41610 $SN_CORR_TRAFO_LIN_MAX. Parametr <Corr_Vect> jest zastępowany przez wektor błędu. Dotyczy to również sytuacji, gdy obróbka została przerwana wraz z alarmem (patrz parametr <No_Alarm>). W składowych, w których wartość korekcji przekroczyła dopuszczalny limit, wektor błędu zawiera poprawną dla znaku różnicę między określoną wartością korekcji a limitem. Zawartość składowych, w których limit nie został przekroczony wynosi 0.

		42	Przy korekcji wektora kierunku odchylenie kątowe względem aktualnego kierunku jest większe niż wartość maksymalna zadana przez daną ustawczą SD41611 $SN_CORR_TRAFO_DIR_MAX.
		43	Próba zapisania zmiennej systemowej została odrzucona z powodu braku uprawnień do zapisu.
<Corr_Vect>:	Wektor korekcji Zawartość wektora kierunku jest definiowana za pomocą następujących parametrów **<**Corr_Index**>** i <Corr_Mode>. Jeżeli <Corr_Status> = 41, zawartość wektora jest zastępowana (patrz wyżej).		
	Typ danych:	REAL	
<Corr_Index**>**:	Segment, w którym jest modyfikowany element korekcji / Indeks osi orientacji, w której jest modyfikowany wektor kierunku		
	Typ danych:	INT	
	Parametr <Corr_Index> jest kodowany dziesiętnie (miejsce jednostek do dziesiątek):		
	Miejsce jednostek:	Zawiera indeks segmentu lub osi orientacji w łańcuchu częściowym.	
	Miejsce dziesiątek:	Odniesienie do łańcucha częściowego.	
		0x	Łańcuch przedmiotu obrabianego
		1x	Łańcuch narzędzia

<Corr_Mode>:	Tryb korekcji		
	Typ danych:	INT	
	Parametr <Corr_Mode> jest kodowany dziesiętnie (miejsce jednostek do tysięcy):		
	Miejsce jednostek:	Określa, jaki element ma być korygowany.	
		xxx**0**	Korekcja liniowego wektora offsetu
		xxx**1**	Korekcja wektora kierunku osi orientacji
	Miejsce dziesiątek:	Określa sposób modyfikacji elementu korekcji, do którego odwołuje się zawartość <Corr_Index>.	
		xx**0**x	Wektor korekcji zapisywany jest bezpośrednio w elemencie korekcji. Ten wariant może być użyty do bezpośredniego opisania elementu korekcji bez znajomości indeksu <n> odpowiedniej danej systemowej ($NK_OFF_DIR[<n>, ...]).
		xx**1**x	Jak 0, ale z tą różnicą, że przekazana wartość korekcji jest interpretowana w globalnym układzie współrzędnych. Różnica między wariantem 0 i 1 powstaje zwykle, gdy łańcuch kinematyki w ustawieniu podstawowym (pozycje wszystkich osi orientacji są równe 0) zawiera kolejne obroty.
		xx**2**x	Jak 1, ale z tą różnicą, że wartość korekcji odnosi się do całego segmentu, tzn. do elementu korekcji jest wpisywana taka wartość, że cały segment osiąga długość zdefiniowaną przez wartość korekcji.
		Uwaga: Przy opisywaniu wektora kierunku osi orientacji wartości 1 i 2 są niedozwolone.	
	Miejsce setek:	Określa, w jaki sposób należy interpretować treść parametru <Corr_Vect>.	
		x**0**xx	Wyznaczony wektor korekcji <Corr_Vect> zawiera całkowitą nową długość elementu korekcji lub segmentu, do którego odnosi się <Corr_Index> wraz z miejscem dziesiętnym <Corr_Mode> (korekcja absolutna).
		x**1**xx	Wyznaczony wektor korekcji <Corr_Vect> zawiera tylko różnicę względem aktualnej długości elementu korekcji lub segmentu, do którego odnosi się <Corr_Index> wraz z miejscem dziesiętnym <Corr_Mode> (korekcja przyrostowa).
		Wskazówka: W przypadku korekcji wektora kierunku osi orientacji zawartość miejsca setek musi wynosić 0.	
	Miejsce tysięcy:	Określa, czy korekcja ma być ograniczona za pomocą następującej wartości maksymalnej: • SD41610 $SN_CORR_TRAFO_LIN_MAX lub • SD41611 $SN_CORR_TRAFO_DIR_MAX	
		0xxx	Kontrola korekcji maksymalnej jest aktywna.
		1xxx	Kontrola korekcji maksymalnej nie jest aktywna.

<No_Alarm>:	Zachowanie się w przypadku błędu (wartość zwrotna > 0) (**opcjonalnie**)		
	Typ danych:	BOOL	
	Wartość:	FALSE (domyślne)	W przypadku błędu wykonywanie programu jest przerywane oraz wyświetlany jest alarm 14103.
		TRUE	W przypadku błędu obróbka programu nie jest zatrzymana i nie jest wyświetlany żaden alarm. Przypadek zastosowania: Specyficzna dla użytkownika reakcja zgodnie z wartością zwrotną.

Uwaga

Jeśli przy wywołaniu funkcji wystąpi błąd, to wyzwolony zostanie alarm lub zwrócony zostanie numer błędu (patrz parametr <No_Alarm>), aby sam użytkownik mógł odpowiednio zareagować na stan błędu. Przyczynę błędu określa parametr alarmu. Numer błędu zwrócony w miejsce alarmu jest identyczny z parametrem alarmu.

Dalsze informacje o CORRTRAFO

Struktura kinematyczna maszyny z transformacją orientacji jest opisywana za pomocą jednego lub dwóch łańcuchów kinematyki (łańcuchy częściowe), które wynikają z punktu zerowego globalnego układu współrzędnych. Jeden z dwóch łańcuchów, **łańcuch narzędzia**, kończy się w punkcie odniesienia narzędzia, drugi **łańcuch przedmiotu obrabianego** w punkcie zerowym bazowego układu współrzędnych.

Funkcja CORRTRAFO mierzy długości ramienia dźwigni i kierunki osi na maszynie z transformacją orientacji oraz zapisuje te wartości w specjalnym elemencie korekcji. Łańcuch kinematyki jest opisywany między innymi za pomocą elementu typu OFFSET, który definiowany jest poprzez $NK_TYPE.

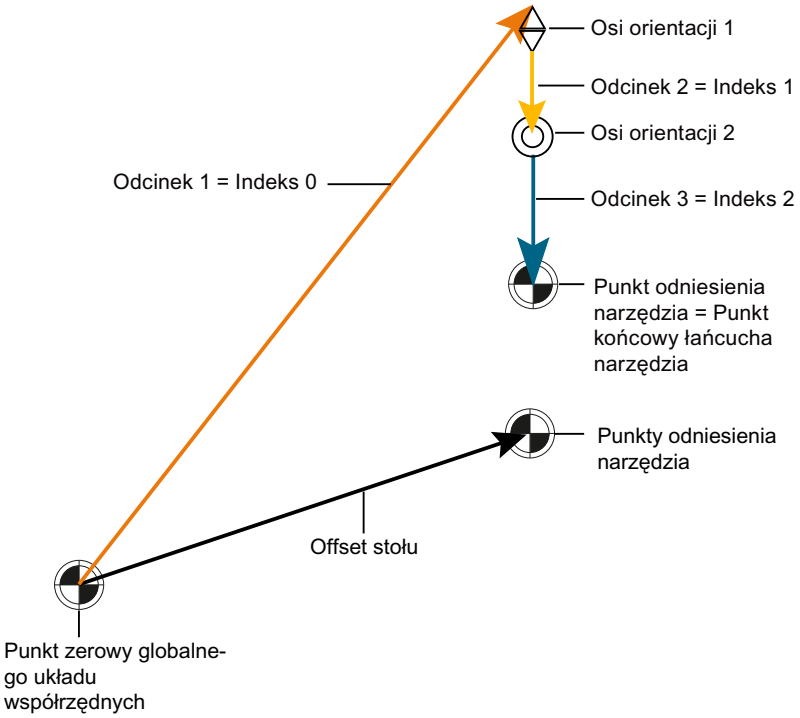
CORRTRAFO pracuje z segmentami

Oba łańcuchy częściowe mogą dzielić się na maksymalnie cztery segmenty:

- Segment 1 rozpoczyna się w punkcie początkowym łańcucha i kończy się na pierwszej osi orientacji.
- Segment 2 jest segmentem pomiędzy osią orientacji 1 i osią orientacji 2.
- Segment 3 jest segmentem pomiędzy osią orientacji 2 i osią orientacji 3.
- Segment 4 jest segmentem pomiędzy osią orientacji 3 i końcem łańcucha przedmiotu obrabianego lub łańcucha narzędzia.

Segmenty mogą zawierać stałe elementy łańcuchów typu OFFSET lub ROT_CONST.

Poniższa grafika przedstawia transformację orientacji z 2 osiami orientacji.



Rysunek 10-1 Przykład CORRTRAFO

Segmenty są dokładnie zdefiniowane: Jeżeli przebieg kinematyki łańcucha częściowego następują od jego punktu początkowego do punktu końcowego, to pierwszy segment oznaczany jest indeksem 0, drugi indeksem 1 itd. Indeks ostatniego segmentu jest wtedy zawsze równy liczbie osi orientacji.

Elementy korekcji

W każdym przypadku stały element łańcucha kinematyki (element łańcucha typu $ NK_TYPE [<n>] = "OFFSET") w każdym segmencie można ustawić za pomocą zmiennej systemowej $NT_CORR_ELEM_T[<n>, 0 ... 3] lub $NT_CORR_ELEM_P[<n>, 0 ... 3]. W tak oznaczonym elemencie, wartości korekcji są opisywane za pomocą funkcji CORRTRAFO, która określana jest podczas pomiaru maszyny.

Przykład z indeksem transformacji = 1:

- $NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"; Offset osi C (osi orientacji 1) w segmencie 1 jest definiowany jako element korekcji.
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"; Offset osi C (osi orientacji 2) w segmencie 2 jest definiowany jako element korekcji.
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"; Offset osi B do punktu odniesienia narzędzia w segmencie 3 jest definiowany jako element korekcji.

Kolejność odsyłaczy w $NT_CORR_ELEM_T/P[<n>, 0 ... 3] musi korespondować z opisanymi wyżej segmentami, tzn. w $NT_CORR_ELEM_T/P [<n>, 0] może znajdować się tylko jeden element łańcucha, który leży przed pierwszą osią orientacji, itd.

W tak zdefiniowanych elementach korekcji funkcja CORRTRAFO opisuje wartości, które określone zostają poprzez pomiar maszyny. Modyfikacja wartości korekcji jest definiowana w CORRTRAFO za pomocą parametru <Corr_Mode>.

Zamknięcie łańcucha

Jeżeli w zmiennych systemowych $NT_CNTRL[<n>] ustawiono Bit 7 i Bit 8, to na końcu łańcucha przedmiotu obrabianego (Bit 7) lub przed punktem początkowym łańcucha narzędzia (Bit 8) automatycznie wstawiane są wewnętrznie dodatkowe stałe elementy łańcucha, które tworzą połączenie punktu końcowego łańcucha z punktem zerowym maszyny („zamknięcie łańcucha").

Te automatycznie wstawione elementy nie mogą być opisywane z zewnątrz, są tylko odczytywane (patrz zmienne systemowe $AC_TRAFO_CORR_ELEM_P/T).

Punkt do zamknięcia łańcucha narzędzia

Jeżeli zmienna systemowa $NT_CLOSE_CHAIN_T nie jest pusta, łańcuch narzędzia nie zamyka się w punkcie końcowym łańcucha, lecz w punkcie końcowym oznaczonego elementu łańcucha. Kolejne elementy łańcucha, które znajdują się za tym punktem prowadzą do odpowiedniego przesunięcia punktu zerowego przy aktywacji transformacji.

Indeks osi orientacji

Poza stałymi offsetami pomiędzy osiami orientacji, funkcja CORRTRAFO może być również wykorzystywana do opisu wektorów kierunku osi orientacji. Indeks osi orientacji oznacza tutaj indeks, który powstaje, gdy kinematyczny łańcuch częściowy przebiega od początku do końca, a liczba rozpoczyna się od zera. Dlatego indeks osi orientacji jest zawsze taki sam jak indeks poprzedniego segmentu.

Indeks osi orientacji może być również określony za pomocą zmiennej systemowej $AC_TRAFO_ORIAX_LOC.

Maksymalna dopuszczalna zmiana elementu łańcucha

Maksymalna dopuszczalna zmiana elementu łańcucha może być ograniczana za pomocą obu danych ustawczych SD41610 $SN_CORR_TRAFO_LIN_MAX dla wektorów offsetu i SD41611 $SN_CORR_TRAFO_DIR_MAX dla wektorów kierunku osi orientacji. SD41610 $SN_CORR_TRAFO_LIN_MAX podaje, o jaką maksymalną wartość można zmienić każdą poszczególną składową wektora względem swojej wartości odniesienia. SD41611 $SN_CORR_TRAFO_DIR_MAX podaje, o jaki maksymalny kąt może się zmienić kierunek wektora osi względem jego wartości odniesienia. Wartością odniesienia jest zawsze odpowiednia wartość, która działa w aktywnej transformacji wywołanej przez CORRTRAFO. Oznacza to, że ewentualnie zmienione treści danych kinematyki w zarządzaniu danymi po aktywacji transformacji nie mają żadnego wpływu na sposób działania funkcji CORRTRAFO.

Korekcje narzędzi 11

11.1 Pamięć korekcji

Budowa korekcji pamięci

Każde pole danych można wywołać przy pomocy numeru T i numeru D (oprócz "płaskiego nr D") i zawiera ono oprócz danych geometrycznych dla narzędzia jeszcze dalsze wpisy, np. typ narzędzia.

Płaska struktura numerów D

"Płaska struktura numerów D" jest stosowana, gdy zarządzanie narzędziami następuje poza NC. W tym przypadku numery D z przynależnymi zestawami korekt narzędzia są tworzone bez przyporządkowania do narzędzi

W programie obróbki można dalej programować T. To T nie ma jednak odniesienia do zaprogramowanego numeru D.

Dane ostrzy nadawane przez użytkownika

Przez daną maszynową mogą być konfigurowane dane użytkownika dot. ostrza. Należy przestrzegać danych producenta maszyny.

Parametry narzędzia

Uwaga

Poszczególne wartości w pamięci korekcji

Poszczególne wartości pamięci korekcji P1 do P25 dają się czytać i zapisywać z programu przez zmienne systemowe. Wszystkie pozostałe parametry są zarezerwowane.

Parametry narzędzi $TC_DP6 do $TC_DP8, $TC_DP10 i $TC_DP11, jak też $TC_DP15 do $TC_DP17, $TC_DP19 i $TC_DP20 mają w zależności od typu narzędzia inne znaczenie.

Parametr narzędzia numer (DP)	Znaczenie zmiennej systemowej	Wskazówka
$TC_DP1	Typ narzędzia	Przegląd patrz lista
$TC_DP2	Położenie ostrza	Tylko dla narzędzi tokarskich
Geometria	**Korekcja długości**	
$TC_DP3	Długość 1	Obliczenie według
$TC_DP4	Długość 2	Typ i płaszczyzna
$TC_DP5	Długość 3	
Geometria	**Promień**	

Parametr narzędzia numer (DP)	Znaczenie zmiennej systemowej	Wskazówka
$TC_DP6 [1)] $TC_DP6 [2)]	Promień 1 / długość 1 Średnica d	Narz. frezarskie/tokarskie/szlifierskie Piła do rowków
$TC_DP7 [1)] $TC_DP7 [2)]	Długość 2 / promień naroża frezów stożkowych Szerokość rowka b promień naroża	Narzędzia frezarskie Piła do rowków
$TC_DP8 [1)] $TC_DP8 [2)]	Promień zaokrąglenia 1 dla narzędzi frezarskich Występ k	Narzędzia frezarskie Piła do rowków
$TC_DP9 [1) 3)]	Promień zaokrąglenia 2	Zarezerwowano
$TC_DP10 [1)]	Kąt 1 strona czołowa narzędzia	Stożkowe narzędzia frezarskie
$TC_DP11 [1)]	Kąt 2 oś wzdłużna narzędzia	Stożkowe narzędzia frezarskie
Zużycie	**Korekcja długości i promienia**	
$TC_DP12	Długość 1	
$TC_DP13	Długość 2	
$TC_DP14	Długość 3	
$TC_DP15 [1)] $TC_DP15 [2)]	Promień 1 / długość 1 Średnica d	Narz. frezarskie/tokarskie/szlifierskie Piła do rowków
$TC_DP16 [1)] $TC_DP16 [3)]	Długość 2 / Promień naroża frezów stożkowych, szerokość rowka b promień naroża	Narzędzia frezarskie Piła do rowków
$TC_DP17 [1)] $TC_DP17 [2)]	Promień zaokrąglenia 1 dla narzędzi frezarskich Występ k	Frezowanie / frezowanie czołowe 3D Piła do rowków
$TC_DP18 [1) 3)]	Promień zaokrąglenia 2	Zarezerwowano
$TC_DP19[1)]	Kąt 1 strona czołowa narzędzia	Stożkowe narzędzia frezarskie
$TC_DP20[1)]	Kąt 2 oś wzdłużna narzędzia	Stożkowe narzędzia frezarskie
Wymiar bazowy/przystawka	**Korekcje długości**	
$TC_DP21	Długość 1	
$TC_DP22	Długość 2	
$TC_DP23	Długość 3	
Technologia		
$TC_DP24	Kąt przystawienia	Tylko dla narzędzi tokarskich
$TC_DP25		Zarezerwowano

1) Obowiązuje również dla narzędzi frezarskich do frezowania czołowego 3D

2) W przypadku piły do rowków typ narzędzia

3) zarezerwowano (nie jest obsługiwane przez SINUMERIK 840D sl)

Uwagi

Dla wielkości geometrycznych (np. długość 1 albo promień) jest wiele wpisywanych składowych. Są one sumowane tworząc wielkość wynikową (np. długość całkowita 1, promień całkowity), która następnie działa.

Niepotrzebne korekcje należy wyposażyć w wartość zero.

Parametry narzędzia $TC-DP1 do $TC-DP23 z narzędziami kształtowymi

Uwaga

Parametry narzędzi, które nie są wymienione w tablicy, jak np. $TC_DP7, nie są poddawane ocenie, tzn. ich treść jest bez znaczenia.

Parametr narzędzia numer (DP)	Znaczenie	Ostrza Dn		Wskazówka
$TC_DP1	Typ narzędzia			400 do 599
$TC_DP2	Położenie ostrza			
Geometria	**Korekcja długości**			
$TC_DP3	Długość 1			
$TC_DP4	Długość 2			
$TC_DP5	Długość 3			
Geometria	**Promień**			
$TC_DP6	Promień			
Geometria	**Kąt graniczny**			
$TC_DP10	Minimalny kąt graniczny			
$TC_DP11	Maksymalny kąt graniczny			
Zużycie	**Korekcja długości i promienia**			
$TC_DP12	Zużycie długość 1			
$TC_DP13	Zużycie długość 2			
$TC_DP14	Zużycie długość 3			
$TC_DP15	Zużycie promień			
Zużycie	**Kąt graniczny**			
$TC_DP19	Zużycie min. kąt graniczny			
$TC_DP20	Zużycie maks. kąt graniczny			
Wymiar bazowy/przystawka	**Korekcje długości**			
$TC_DP21	Długość 1			
$TC_DP22	Długość 2			
$TC_DP23	Długość 3			

Wartość podstawowa i wartość zużycia

Wielkości wynikowe wynikają każdorazowo z sumy wartości podstawowej i wartości zużycia (np. $TC_DP6 + $TC_DP15 dla promienia). Do długości narzędzia pierwszego ostrza jest poza tym jeszcze dodawany wymiar bazowy ($TC_DP21 – $TC_DP23). Dodatkowo na tę długość narzędzia działają wszystkie inne wielkości, które również w przypadku zwykłego narzędzia mogą mieć wpływ na jego efektywną długość (przystawka, orientowany nośnik narzędzi, dane ustawcze).

Kąty graniczne 1 i 2

Kąty graniczne 1 lub 2 odnoszą się każdorazowo do wektora od punktu środkowego ostrza do punktu odniesienia ostrza i są liczone przeciwnie do kierunku ruchu wskazówek zegara.

11.2 Korekcje addytywne

11.2.1 Wybranie korekcji addytywnych (DL)

Korekcje addytywne mogą być traktowane jako korekty procesu programowane w czasie obróbki. Odnoszą się one do danych geometrycznych ostrza i są przez to częścią składową danych ostrza narzędzia.

Dostęp do danych korekcji addytywnej jest uzyskiwany przez numer DL (DL: Location dependent; korekcje odnośnie każdorazowego miejsca użycia) i wprowadzane przez interfejs graficzny.

Zastosowanie

Przez korekcje addytywne mogą być wyrównywane błędy uwarunkowane miejscem użycia.

Składnia

```
DL=<numer>
```

Znaczenie

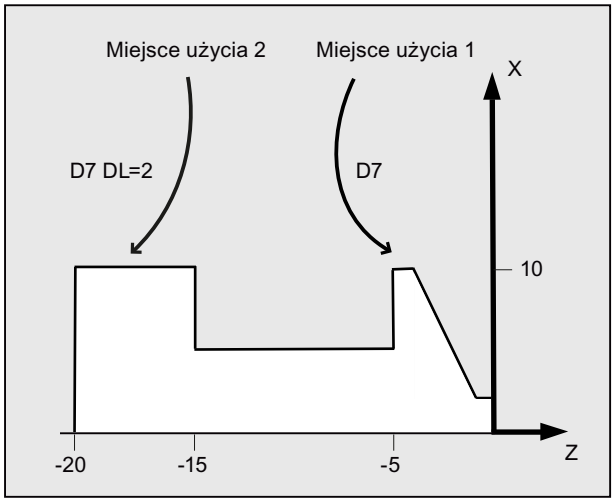
DL:	Polecenie do uaktywnienia korekcji addytywnej
<numer>:	Poprzez parametr <numer> jest podawany będący do uaktywnienia addytywny zestaw danych korekcyjnych narzędzia.

Uwaga

Ustalenie liczby i uaktywnienie korekcji addytywnych następuje przez dane maszynowe (→ przestrzegać danych producenta maszyny!).

Przykład

To samo ostrze jest stosowane do dwóch gniazd łożyskowych:



Kod programu	Komentarz
N110 T7 D7	; Głowica rewolwerowa jest pozycjonowana na miejscu 7. D7 i DL=1 są uaktywniane i realizowane w następnym bloku.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; Dodatkowo do D7 jest uaktywniane DL=2 i realizowane w następnym bloku.
N150 G1 Z-21	
N160 G0 X200 Z200	; Ruch do punktu wymiany narzędzia
...	

11.2.2 Ustalenie zużycia i wartości ustawczych ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

Wartości zużycia i ustawcze mogą być czytane i zapisywane poprzez zmienne systemowe. Logika orientuje się przy tym według logiki odpowiednich zmiennych systemowych dla narzędzi i ostrzy.

Zmienne systemowe

`$TC_SCPxy[<t>,<d>]`:	Wartości zużycia, które przez xy są przyporządkowane do każdorazowego parametru geometrycznego, przy czym x odpowiada numerowi wartości zużycia, a y tworzy odniesienie do parametru geometrycznego.
`$TC_ECPxy[<t>,<d>]`:	Wartości ustawcze, które przez xy są przyporządkowane do każdorazowego parametru geometrycznego, przy czym x odpowiada numerowi wartości ustawczej, a y tworzy odniesienie do parametru geometrycznego.
`<t>`: Numer T narzędzia `<d>`: Numer D ostrza narzędzia	

Uwaga

Ustalone wartości zużycia i ustawcze są dodawane do parametrów geometrycznych i pozostałych parametrów korekcyjnych (numery D).

Przykład

Wartość zużycia długości 1 dla ostrza <d> narzędzia <t> jest ustalona na wartość 1.0.

Parametr: $TC_DP**3** (długość 1, w przypadku narzędzi tokarskich)

Wartości zużycia: $TC_SCP1**3** do $TC_SCP6**3**

Wartości ustawcze: $TC_ECP1**3** bis $TC_ECP6**3**

$TC_SCP4**3** [<t>,<d>] = 1.0

11.2.3 Skasowanie korekcji addytywnych (DELDL)

Przy pomocy polecenia `DELDL`są kasowane addytywne korekcje ostrza narzędzia (zwolnienie pamięci). Są przy tym kasowane zarówno ustalone wartości zużycia, jak też wartości ustawcze.

Składnia

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<Status>=DELDL[<t>,<d>]
```

Znaczenie

`DELDL`:	Polecenie do kasowania korekcji addytywnych
`<t>`:	Numer T narzędzia
`<d>`:	Numer D ostrza narzędzia
`DELDL[<t>,<d>]`:	Są kasowane wszystkie addytywne korekcje ostrza `<d>` narzędzia `<t>`.

DELDL[<t>]:	Są kasowane wszystkie addytywne korekcje wszystkich ostrzy narzędzia <t>.	
DELDL:	Są kasowane wszystkie korekcje addytywne wszystkich ostrzy wszystkich narzędzi jednostki TO (dla kanału, w którym zostanie zaprogramowane polecenie).	
<status>:	Status kasowania	
	Wartość:	Znaczenie:
	0	Skasowanie zostało przeprowadzone z powodzeniem.
	-	Skasowanie nie zostało przeprowadzone (gdy parametryzowanie określa dokładnie jedno ostrze), albo jest niekompletne (gdy parametryzowanie określa wiele ostrzy).

Uwaga

Wartości zużycia i wartości ustawczych aktywnych narzędzi nie można skasować (zachowuje się analogicznie do zachowania się D wzgl. danych narzędzia przy kasowaniu).

11.3 Traktowanie specjalne korekcji narzędzia

Przy pomocy danych ustawczych SD42900 do SD42960 można sterować ewaluacją znaków długości narzędzia i zużycia.

Dotyczy to również zachowania się składowych zużycia podczas lustrzanego odbicia osi geometrycznych lub zmiany płaszczyzny obróbki, a także kompensacji temperatury w kierunku narzędzia.

Wartości zużycia

Jeżeli podana poniżej wartość zużycia odnosi się do sumy rzeczywistych wartości zużycia ($TC_DP12 do $TC_DP20), korekcji sumarycznych z wartościami zużycia ($SCPX3 do $SCPX11) i wartości ustawczych ($ECPX3 do $ECPX11).

Bliższe dane dot. korekcji sumarycznych patrz:
Literatura:
Podręcznik działania Zarządzanie narzędziami

Dane ustawcze

SD42900 $SC_MIRROR_TOOL_LENGTH	Lustrzane odbicie komponentów długości narzędzia i komponentów wymiaru bazowego.
SD42910 $SC_MIRROR_TOOL_WEAR	Lustrzane odbicie wartości zużycia komponentów długości narzędzia.
SD42920 $SC_WEAR_SIGN_CUTPOS	Reakcja na znak składowej zużycia w zależności od położenia ostrza.
SD42930 $SC_WEAR_SIGN	Odwraca znaki wymiarów zużycia.
SD42935 $SC_WEAR_TRANSFORM	Transformacja wartości zużycia.
SD42940 $SC_TOOL_LENGTH_CONST	Przyporządkowanie komponentów długości narzędzia do osi geometrycznych.
SD42950 $SC_TOOL_LENGTH_TYPE	Przyporządkowanie komponentów długości narzędzia niezależnie od typu narzędzia.
SD42960 $SC_TOOL_TEMP_COMP	Wartość kompensacji temperatury w kierunku narzędzia. Działa również przy istniejącej orientacji narzędzia.

Literatura

Podręcznik działania Funkcje podstawowe; Korekcja narzędzia (W1)

Dodatkowe informacje

Uaktywnienie zmienionych danych ustawczych

Nowa reakcja na składowe narzędzia przy zmianie opisanych danych ustawczych działa dopiero wtedy, gdy następny raz zostanie wybrane ostrze narzędzia Jeżeli narzędzie jest już aktywne i ma zacząć działać zmieniona reakcja na dane tego narzędzia, narzędzie to musi zostać wybrane ponownie

Odpowiednio jest w przypadku, gdy zmienia się wynikowa długość narzędzia, ponieważ zmieniony został stan lustrzanego odbicia osi. Narzędzie musi po poleceniu lustrzanego odbicia zostać ponownie zostać wybrane, aby zaczęły działać zmienione komponenty długości narzędzia.

Orientowalne nośniki narzędzi i nowe dane ustawcze

Dane ustawcze SD42900 do SD42940 nie działają na komponenty ewentualnie aktywnego orientowalnego nośnika narzędzi. Narzędzie wchodzi jednak zawsze ze swoją całą wynikową długością (długość narzędzia + zużycie + wymiar bazowy) do obliczenia z orientowanym nośnikiem narzędzi. Przy obliczaniu wynikającej długości całkowitej są uwzględniane wszystkie zmiany, które zostały spowodowane przez dane ustawcze; tzn. wektory orientowanego nośnika narzędzi są niezależne od płaszczyzny obróbki.

Uwaga

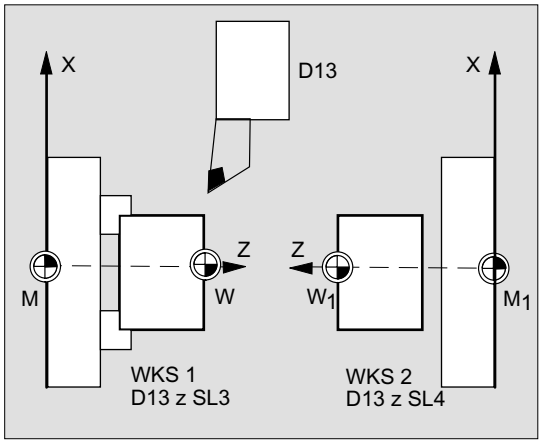
Często przy zastosowaniu orientowalnych nośników narzędzi ma sens zdefiniowanie wszystkich narzędzi dla systemu podstawowego nie poddanego lustrzanemu odbiciu, również tych, które są stosowane tylko przy obróbce w lustrzanym odbiciu. Przy obróbce z osiami poddanymi lustrzanemu odbiciu nośnik narzędzi jest wówczas tak obracany, że rzeczywiste położenie narzędzia jest prawidłowo opisywane. Wszystkie składowe długości narzędzia działają wówczas automatycznie we właściwym kierunku, tak że zbędne staje się sterowanie reakcją poszczególnych składowych przez dane ustawcze zależnie od stanu lustrzanego odbicia poszczególnych osi.

Dalsze możliwości zastosowania

Zastosowanie funkcji orientowanych nośników narzędzi może mieć sens również wtedy, gdy w maszynie nie jest fizycznie przewidziana żadna możliwość obracania narzędzi, ale są one na stałe zainstalowane z różnym zorientowaniem. Zwymiarowania narzędzi można wówczas jednolicie dokonać w zorientowaniu podstawowym, a wymiary mające znaczenie dla obróbki wynikają z obrotu wirtualnego nośnika narzędzi.

11.3.1 Lustrzane odbicie długości narzędzia

Przy pomocy danych ustawczych SD42900 $SC_MIRROR_TOOL_LENGTH i SD42910 $SC_MIRROR_TOOL_WEAR nierównych zeru można poddać lustrzanemu odbiciu komponenty długości narzędzia i komponenty wymiarów bazowych z wartościami zużycia w ich przynależnych osiach.



SD42900 $SC_MIRROR_TOOL_LENGTH

Dana ustawcza **nierówna** zeru:

Lustrzanemu odbiciu podlegają również komponenty długości narzędzia ($TC_DP3, $TC_DP4 i $TC_DP5) i komponenty wymiarów bazowych ($TC_DP21, $TC_DP22 i $TC_DP23), których przynależne osie są poddane lustrzanemu odbiciu, - przez zmianę znaku liczby.

Wartości zużycia **nie** podlegają równoczesnemu lustrzanemu odbiciu. Gdy mają one również zostać poddane lustrzanemu odbiciu, musi być ustawiona dana ustawcza SD42910 $SC_MIRROR_TOOL_WEAR.

SD42910 $SC_MIRROR_TOOL_WEAR

Dana ustawcza **nierówna** zeru:

Wartości zużycia komponentów długości narzędzia, których przynależne osie są poddane lustrzanemu odbiciu, są również poddawane lustrzanemu odbiciu - przez odwrócenie znaku liczby.

11.3.2 Reakcja na znak zużycia

Przy pomocy ustawionych danych ustawczych SD42920 $SC_WEAR_SIGN_CUTPOS i SD42930 $SC_WEAR_SIGN nierówne zeru, można poddać odwróceniu ewaluację znaku komponentów zużycia.

SD42920 $SC_WEAR_SIGN_CUTPOS

Dana ustawcza **nierówna** zeru:

W przypadku narzędzi o mającym znaczeniu położeniu ostrza (narzędzia tokarskie i szlifierskie, typy narzędzi 400) reakcja na znak liczby składowych zużycia w płaszczyźnie

obróbki zależy od położenia ostrza. W przypadku typów narzędzi bez mającego znaczenie położenia ostrza ta dana ustawcza nie ma znaczenia

W poniższej tabeli przez X zaznaczone są wymiary, których znak jest odwracany przez SD42920 (nierówna 0):

Położenie ostrza	Długość 1	Długość 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Uwaga

Ewaluacja znaku przez SD42920 i SD42910 jest niezależna od siebie. Gdy np. znak podania wymiaru zostanie zmieniony przez obydwie dane ustawcze, znak wynikowy pozostanie bez zmian.

SD42930 $SC_WEAR_SIGN

Dana ustawcza **nierówna** zeru:

Znak wszystkich wymiarów zużycia jest odwracany. Działa on zarówno na długość narzędzia jak również na pozostałe wielkości promienia narzędzia, promień zaokrąglenia, itd.

Gdy zostanie wprowadzona dodatnia miara zużycia, narzędzie stanie się przez to "krótsze" i "cieńsze", patrz punkt "Korekcja narzędzia, traktowanie specjalne", początek działania zmienionych danych ustawczych".

11.3.3 Układ współrzędnych aktywnej obróbki (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

Zależnie od kinematyki maszyny albo od występowania orientowanego nośnika narzędzi wartości zużycia zmierzone w jednym z tych układów współrzędnych są przenoszone lub transformowane na odpowiedni układ współrzędnych.

Układy współrzędnych aktywnej obróbki

Z poniższych układów współrzędnych mogą wynikać przesunięcia długości narzędzia, które uwzględniają zużycie jako składową długości narzędzia przez odpowiednie polecenie G z grupy nr 56 do aktywnego narzędzia.

- Układ współrzędnych maszyny (MKS)
- Bazowy układ współrzędnych (BKS)
- Układ współrzędnych przedmiotu obrabianego (WKS)

- Układ współrzędnych narzędzia (TCS)
- Układ współrzędnych narzędzia transformacji kinematycznej (KCS)

Składnia

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

Znaczenie

TOWSTD:	Wartość ustawienia podstawowego korekcji w długości narzędzia, wartość zużycia
TOWMCS:	Korekcje w długości narzędzia w MKS
TOWWCS:	Korekcje w długości narzędzia w WKS
TOWBCS:	Korekcje w długości narzędzia w BKS
TOWTCS:	Korekcje długości narzędzia w punkcie odniesienia nośnika narzędzi (orientowany nośnik narzędzi)
TOWKCS:	Korekcje długości narzędzia głowicy narzędziowej (transformacja kinematyczna)

Dalsze informacje

Cechy wyróżniające

Poniższa tabela pokazuje główne cechy wyróżniające:

Polecenie G	Wartość zużycia	Aktywny orientowany nośnik narzędzi
TOWSTD:	Wartość ustawienia podstawowego, długość narzędzia	Wartości zużycia podlegają obrotowi.
TOWMCS:	Wartość zużycia w MKS. TOWMCS jest identyczny z TOWSTD, gdy nie jest aktywny orientowany nośnik narzędzi.	Obraca się tylko wektor wynikowej długości narzędzia bez uwzględnienia zużycia.
TOWWCS:	Wartość zużycia jest przeliczana w układzie współrzędnych przedmiotu obrabianego na układ współrzędnych maszyny.	Wektor narzędzia jest obliczany bez uwzględnienia zużycia, jak w przypadku TOWMCS.
TOWBCS:	Wartość zużycia jest przeliczana w bazowym układzie współrzędnych na układ współrzędnych maszyny.	Wektor narzędzia jest obliczany bez uwzględnienia zużycia, jak w przypadku TOWMCS.
TOWTCS:	Wartość zużycia jest przeliczana w układzie współrzędnych narzędzia na układ współrzędnych maszyny.	Wektor narzędzia jest obliczany bez uwzględnienia zużycia, jak w przypadku TOWMCS.

TOWWCS, TOWBCS, TOWTCS: Wektor zużycia jest dodawany do wektora narzędzia.

Transformacja liniowa

Zdefiniowanie długości narzędzia w układzie współrzędnych maszyny ma sens, gdy układ ten wynika z bazowego układu współrzędnych przez transformację liniową.

Transformacja nieliniowa

Jeżeli transformacja nieliniowa jest aktywowana np. za pomocą TRANSMIT, wówczas po określeniu MKS, automatycznie stosowany jest BKS jako żądany układ współrzędnych.

Bez transformacji kinematycznej i bez zorientowanego nośnika narzędzi

Jeżeli nie jest aktywna ani transformacja kinematyczna ani orientowany nośnik narzędzi, to aż do WKS wszystkie pozostałe cztery układy współrzędnych pokrywają się. Tak więc tylko układ współrzędnych przedmiotu obrabianego różni się od pozostałych. Ponieważ należy poddać ewaluacji wyłącznie długości narzędzi, translacje między układami współrzędnych nie mają żadnego znaczenia.

Literatura:
Dalsze informacje dot. korekcji narzędzi patrz:
Opis funkcji, Funkcje podstawowe; Korekcja narzędzia (W1)

Uwzględnienie wartości zużycia

Dana ustawcza **SD42935 $SC_WEAR_TRANSFORM** określa, która z trzech składowych zużycia:

- zużycie,
- korekcja sumaryczna dokładna,
- korekcja sumaryczna zgrubna,

ma podlegać obrotowi przez transformację przystawki albo orientowany nośnik narzędzi, gdy jest aktywne jedno z następujących poleceń G:

- `TOWSTD`
  Położenie podstawowe. Dla korekcji w długości narzędzia
- `TOWMCS`
  Wartości zużycia w układzie współrzędnych maszyny (MKS).
- `TOWWCS`
  Wartości zużycia w układzie współrzędnych przedmiotu obrabianego (WKS).
- `TOWBCS`
  Wartości zużycia w bazowym układzie współrzędnych (BKS).

- TOWTCS
  Wartości zużycia w układzie współrzędnych narzędzia na zamocowaniu uchwytu narzędzia (T odniesienie nośnika narzędzi).
- TOWKCS
  Wartości zużycia w układzie współrzędnych głowicy narzędziowej przy transformacji kinetycznej.

Uwaga

Ocena poszczególnych składowych zużycia (przyporządkowanie do osi geometrycznych, ocena znaku) ma wpływ na:

- Aktywną płaszczyznę
- Transformację przystawki
- Dane ustawcze:
  - 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 Długość narzędzia i zmiana płaszczyzny

Przy pomocy ustawionych danych ustawczych SD42940 $SC_TOOL_LENGTH_CONST nierównych zero można dla narzędzi tokarskich i szlifierskich przy zmianie płaszczyzny przyporządkować komponenty długości narzędzia, jak długość, zużycie i wymiar bazowy do osi geometrycznych.

SD42940 $SC_TOOL_LENGTH_CONST

Dana ustawcza **nierówna** zeru:

Przyporządkowanie składowych długości narzędzia (długość, zużycie i wymiar bazowy) do osi geometrycznych przy zmianie płaszczyzny obróbki (G17 - G18) nie ulega zmianie.

Poniższa tabela pokazuje przyporządkowanie składowych długości narzędzia do osi geometrii dla narzędzi tokarskich i szlifierskich (typ narzędzia 400 do 599):

Treść	Długość 1	Długość 2	Długość 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Każda wartość nierówna 0, która jest nierówna jednej z sześciu wymienionych wartości, jest traktowana jako wartość 18.

Poniższa tabela pokazuje przyporządkowanie składowych długości narzędzia do osi geometrii dla wszystkich innych narzędzi (typ narzędzia < 400 lub > 599):

Płaszczyzna obróbki	Długość 1	Długość 2	Długość 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Każda wartość nierówna 0, która jest nierówna jednej z sześciu wymienionych wartości, jest traktowana jako wartość 17.

Uwaga

Przy przedstawianiu w tabelach zakładamy, że osie geometryczne do 3 mają identyfikatory X, Y i Z. Dla przyporządkowania korekcji do osi jest miarodajny nie identyfikator osi lecz ich kolejność.

11.4 Korekcja narzędzia online

11.4.1 Definicja funkcji wielomianowej (FCTDEF)

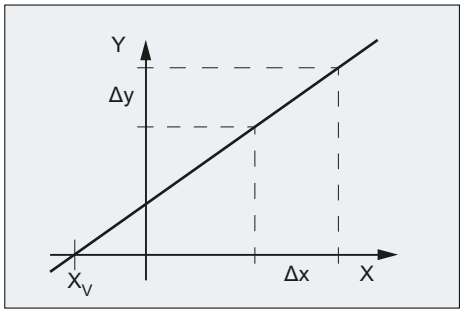
Określone strategie obciągacza (np. obciągacz rolkowy) charakteryzują się tym, że wraz z dosunięciem obciągacza rolkowego do promienia ściernica zmniejsza się w sposób stały (liniowo). W tym celu potrzebna jest funkcja liniowa pomiędzy dosunięciem obciągacza rolkowego i zapisem wartości zużycia każdej długości. Definiowanie funkcji liniowych następuje za pomocą wstępnie zdefiniowanej procedury FCTDEF(...) dla funkcji wielomianowej maksymalnie trzeciego stopnia.

Równanie liniowe

$y = f(x) = a_0 + a_1 * x_1$

a_1: Nachylenie prostej, za pomocą $a_1 = \Delta x / \Delta y$

a_0: Przesunięcie prostej wzdłuż osi X za pomocą $a_0 = -a1 * X_V$



Składnia

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

Znaczenie

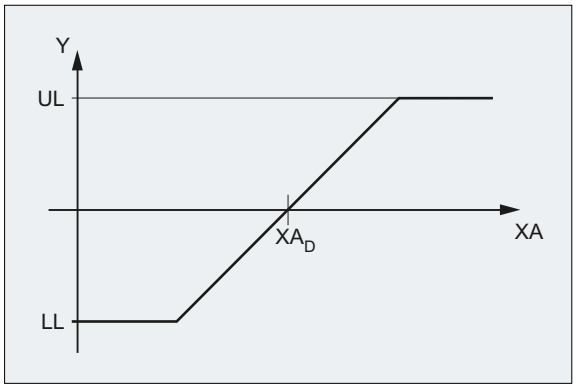
FCTDEF(...):	Definiowanie funkcji wielomianowej dla PUTFTOCF(...): $y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$	
<Func>:	Numer funkcji	
	Typ danych:	INT
	Zakres wartości:	1, 2, 3
<LLimit>:	Dolna wartość ograniczenia	
	Typ danych:	REAL
<ULimit>:	Górna wartość ograniczenia	
	Typ danych:	REAL
<a0>,<a1>,<a2>,<a3>:	Współczynniki funkcji wielomianowej	
	Typ danych:	REAL

Przykład

Ustalenia

- Numer funkcji 1
- Dolna i górna wartość ograniczenia: -100, 100
- Nachylenie charakterystyki $a_1 = 1$
- Punkt pracy powinien znajdować się w środku charakterystyki. Przy tym charakterystyka na podstawie pozycji zadanej osi XA w WKS w chwili definicji funkcji w programie NC musi być przesunięta w kierunku ujemnym Y. $a_0 = -a_1 * XA_D = -1 * \$AA\_IW$
- $a_2 = a_3 = 0$

Charakterystyka



UL	Górna wartość ograniczenia
LL	Dolna wartość ograniczenia
XA_D	Wartość zadana osi XA w chwili definicji funkcji w programie NC

Programowanie

Kod programu	Komentarz
`FCTDEF(1,-100,100,-$AA_IW[XA],1)`	`; Definicja funkcji`

11.4.2 Korekcja narzędzia online zapis, nieprzerwany (PUTFTOCF)

Za pomocą wstępnie zdefiniowanej procedury PUTFTOCF(...) przeprowadzona zostanie korekcja narzędzia online za pomocą wcześniej zdefiniowanej funkcji wielomianowej FCTDEF(...) (Strona 438).

Uwaga

Korekcja narzędzia online może również nastąpić za pomocą akcji synchronicznych.

Dodatkowe informacje patrz Podręcznik funkcjonowania Akcje synchroniczne.

Składnia

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

Znaczenie

`PUTFTOCF(...)`:	Zapis korekcji narzędzia online w sposób ciągły pojedynczymi blokami na podstawie zdefiniowanej funkcji wielomianowej FCTDEF(...).	
`<Func>`:	Numer funkcji, określony podczas definiowania funkcji za pomocą FCTDEF(...)	
	Typ danych:	INT
	Zakres wartości:	1, 2, 3
`<RefVal>`:	Wartość odniesienia, od której ma zostać wyprowadzona korekcja (np. wartość zadana osi).	
	Typ danych:	VAR REAL
`<ToolPar>`:	Numer parametru zużycia (długość 1, 2 lub 3), w którym wartość korekcji ma zostać obliczona.	
	Typ danych:	INT
`<Chan>`:	Numer kanału, w którym korekcja narzędzia online ma działać. **Uwaga**: Podanie jest wymagane tylko wtedy, gdy korekcja ma nie działać w aktywnym kanale.	
	Typ danych:	INT
`<Sp>`:	Numer wrzeciona, dla którego korekcja narzędzia online ma działać. **Uwaga:** Podanie jest wymagane tylko wtedy, gdy zamiast aktywnego, pracującego narzędzia ma zostać skorygowana nieaktywna ściernica.	
	Typ danych:	INT

11.4.3 Zapis korekcji narzędzia online, nieciągły (PUTFTOC)

Funkcja

Za pomocą wstępnie zdefiniowanej procedury PUTFTOC(...) przeprowadzona zostanie korekcja narzędzia online ze stałą wartością korekcji.

Składnia

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

Znaczenie

`PUTFTOC(...)`:	Zapisanie korekcji narzędzia online:	
`<CorrVal>`:	Wartość korekcji, która ma być dodana do parametru zużycia.	
	Typ danych:	REAL

<ToolPar>:	Numer parametru zużycia (długość 1, 2 lub 3), w którym wartość korekcji ma zostać obliczona.	
	Typ danych:	INT
<Chan>:	Numer kanału, w którym korekcja narzędzia online ma działać. **Uwaga**: Podanie jest wymagane tylko wtedy, gdy korekcja ma nie działać w aktywnym kanale.	
	Typ danych:	INT
<Sp>:	Numer wrzeciona, dla którego korekcja narzędzia online ma działać. **Uwaga:** Podanie jest wymagane tylko wtedy, gdy zamiast aktywnego, pracującego narzędzia ma zostać skorygowana nieaktywna ściernica.	
	Typ danych:	INT

11.4.4 Włączenie/wyłączenie korekcji narzędzia online (FTOCON/FTOCOF)

Za pomocą polecenia G FTOCON i FTOCOF korekcja narzędzia online zostanie włączona lub wyłączona.

Składnia

```
FTOCON
...
FTOCOF
```

Znaczenie

FTOCON:	Włączenie korekcji narzędzia online Polecenie musi być programowane w tym kanale, w którym korekcja narzędzia online ma być włączona.
FTOCOF:	Wyłączenie korekcji narzędzia online Polecenie musi być programowane w tym kanale, w którym korekcja narzędzia online ma być wyłączona. **Wskazówka:** Za pomocą FTOCOF korekcja narzędzie nie jest dalej realizowana. W specyficznych dla ostrza danych korekcyjnych pozostaje jednak skorygowana kompletna wartość napisana przy pomocy PUTFTOC/PUTFTOCF. W celu ostatecznego wyłączenia aktywności korekcji narzędzia online musi po FTOCOF nastąpić jeszcze wybór/cofnięcie wyboru narzędzia (T...).

11.5 Korekcja promienia narzędzia 3D

11.5.1 Wybór korekcji promienia narzędzia 3D dla frezowania obwodowego (CUT3DC, CUT3DCD, ISD)

Korekcja promienia narzędzia 3D (KPN-3D) dla frezowania obwodowego bez uwzględnienia powierzchni ograniczających jest wybierana za pomocą działających w sposób modalny poleceń G CUT3DC lub CUT3DCD.

Właściwa aktywacja następuje za pomocą G41 lub G42. Natomiast wyłączenie korekcji promienia narzędzia za pomocą G40.

Składnia

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Znaczenie

CUT3DC:	Korekcja promienia narzędzia 3D dla frezowania obwodowego (tylko przy aktywnej transformacji 5-osiowej)	
CUT3DCD:	Korekcja promienia narzędzia 3D dla frezowania obwodowego w odniesieniu do narzędzia o zmienionych wymiarach (tylko przy aktywnej transformacji 5-osiowej) Różnica promienia jest określana za pomocą parametru narzędzia $TC_DP15.	
G41/G42 X... Y... Z...:	Włącza korekcję promienia narzędzia	
	G41:	Korekcja promienia narzędzia na lewo od konturu
	G42:	Korekcja promienia narzędzia na prawo od konturu
	Uwaga: Włączenie musi nastąpić w bloku liniowym (G0/G1).	
CDOF2:	Wyłącza nadzór na kolizję dla frezowania obwodowego 3D	
ORIC/ORID:	Za pomocą poleceń G ORIC i ORID określane jest zachowanie przy zmianach orientacji na narożach zewnętrznych.	
	ORIC:	Zmiany orientacji na narożach zewnętrznych są nakładane na wstawiony blok okręgu.
	ORID:	Zmiany orientacji na narożach zewnętrznych są wykonywane przed wstawionym blokiem okręgu.

ISD=<Value>:	Za pomocą adresu ISD można zmieniać głębokość zagłębiania się narzędzia przy frezowaniu obwodowym i **aktywnej** korekcji promienia narzędzia 3D.
	<Value>: Wartość zagłębienia
G40 X... Y... Z...:	Wyłączenie korekcji promienia narzędzia **Uwaga:** Włączenie musi nastąpić w bloku liniowym (G0/G1) za pomocą ruchów osi geometrycznych.

Uwaga

Polecenia G do wyboru korekcji promienia narzędzia 3D są wykorzystywane w bloku dosuwu, tzn. zazwyczaj w bloku, który zawiera G41 lub G42.

G41 i G42 mogą być również programowane w blokach bez ruchów postępowych, w których są korekcje ważnych osi geometrycznych. W tym przypadku blok dosunięcia jest pierwszym blokiem ruchu następującym po takim bloku.

Zmiana wariantu korekcji promienia narzędzia 3D przy aktywnej korekcji promienia narzędzia jest ignorowana bez alarmu.

Przykład

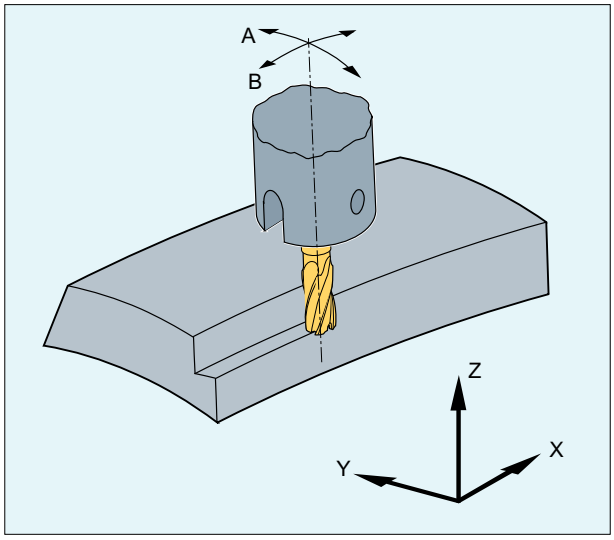
```
Kod programu                          Komentarz
                                      ; Definicja narzędzia D1:
$TC_DP1[1,1]=120                      ; Typ (frez palcowy)
$TC_DP3[1,1]=20                       ; Wektor korekcji długości
$TC_DP6[1,1]=8                        ; Promień

N10 X0 Y0 Z0 T1 D1 F12000             ; Wybór narzędzia.
N20 TRAORI(1)                         ; Włączenie transformacji.
N30 G42 ORIC ISD=10 CUT3DC G64 X30    ; Aktywacja frezowania obwodowego 3D.
                                      ; Zmiana orientacji na narożach zewnętrznych w sposób ciągły.
                                      ; Głębokość zagłębienia: 10 mm
N40 ORIWKS A30 B15                    ; Zmiana orientacji na narożu poprzez podanie pozycji osi.
N50 Y20 A3=1 C3=1                     ; Blok ruchu ze zmianą orientacji.
                                      ; Podanie orientacji za pomocą wektora kierunku.
N60 X50 Y30                           ; Blok ruchu ze stałą orientacją.
N70 Y50 A3=0.5 B3=1 C3=5              ; Blok ruchu ze zmianą orientacji.
N80 M63                               ; Blok bez informacji o ruchu.
N90 X0 ISD=20                         ; Blok ruchu ze zmianą głębokości zagłębienia.
N100 G40 Y0                           ; Wyłączenie korekcji promienia narzędzia.
N110 M30
```

Dalsze informacje

Tor i orientacja

Używany tutaj wariant frezowania obwodowego jest realizowany poprzez zadanie toru (linii prowadzącej) i przynależnej orientacji. W przypadku tego rodzaju obróbki, kształt narzędzia na torze nie ma znaczenia. Decydujący jest tylko promień w punkcie przyporu narzędzia.



Rysunek 11-1 Frezowanie obwodowe

Sposób dosunięcia

Sposób dosunięcia w przypadku korekcji promienia narzędzia 3D jest zawsze NORM.

Zachowanie się na narożach zewnętrznych

Na narożach zewnętrznych polecenia z grupy G nr 18 wykorzystywane są do frezowania obwodowego z korekcją promienia narzędzia 3D, analogicznie do warunków dla korekcji promienia narzędzia 2½D:

- G450: Okrąg przejściowy (narzędzie obchodzi naroże przedmiotu obrabianego po torze kołowym).
  W przeciwieństwie do rozwiązania dla korekcji promienia narzędzia 2½D, wstawiany element konturu na narożu zewnętrznym jest zawsze okręgiem o promieniu 0, na który korekcja promienia narzędzia działa w taki sam sposób jak na każdy inny zaprogramowany tor. W miejsce okręgu nie można wstawić elementów stożka. Adres DISC nie ma w związku z tym żadnego znaczenia i nie jest wykorzystywany.
- G451: Punkt przecięcia równoległych (narzędzie wychodzi z materiału w narożu)
  Punkt przecięcia jest określany przez przedłużenie krzywych offsetu obydwu bloków i określenie punktu ich przecięcia w płaszczyźnie prostopadłej do orientacji narzędzia na narożu.

Metoda punktu przecięcia (G451) nie jest stosowana, gdy pomiędzy uczestniczącymi blokami ruchu wstawiany jest co najmniej jeden blok, który zawiera zmianę orientacji narzędzia. W takim przypadku w narożu jest zawsze wstawiany okrąg.

Zachowanie przy zmianie orientacji na narożach zewnętrznych

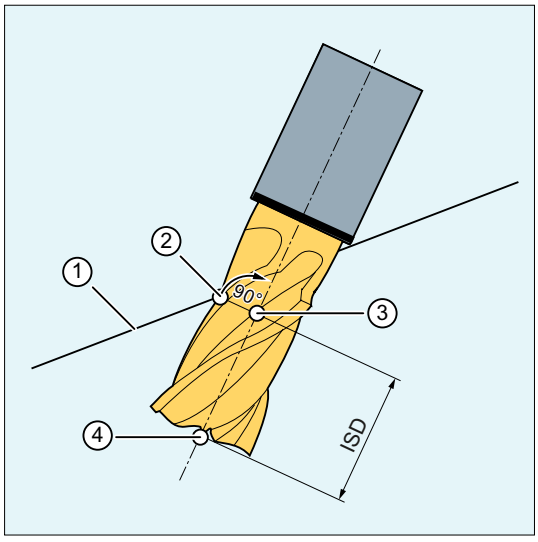
Za pomocą poleceń G - ORIC lub ORID - można ustalać, czy zmiany orientacji, które zostały zaprogramowane pomiędzy dwoma narożami tworzącymi blok ruchu, mają być wykonywane przed rozpoczęciem wstawionego bloku okręgu (ORID) czy równocześnie z nim (ORIC).

Głębokość zagłębienia

Głębokością zagłębienia frezu jest odstęp punktu pomocniczego frezu od wierzchołka narzędzia.

Punkt pomocniczy frezu jest rzutem prostopadłym punktu obróbki frezu na zaprogramowanym torze na oś wzdłużną narzędzia.

Przy głębokości zagłębiania położenie punktu obróbki ustawiane jest na pobocznicy narzędzia.



① Zaprogramowany tor
② Punkt obróbki frezem
③ Punkt pomocniczy frezu
④ Wierzchołek frezu
ISD Głębokość zagłębienia (InSertion Depth)

Rysunek 11-2 Głębokość zagłębienia

Korekcja promienia narzędzia w odniesieniu do narzędzia o zmienionych wymiarach

Korekcja promienia narzędzia 3D do frezowania obwodowego odnosząca się do narzędzia o zmienionych wymiarach jest aktywowana za pomocą polecenia CUT3DCD. Stosuje się ją wtedy, gdy programowany kontur odnosi się do toru punktu środkowego narzędzia znormalizowanego, a obróbka odbywa się za pomocą jednego z różniących się narzędzi. Podczas obliczania korekcji promienia narzędzia 3D uwzględniana będzie tylko wartość zużycia promienia aktywnego narzędzia ($TC_DP15) i ewentualnie zaprogramowane przesunięcia korekcji narzędzia OFFN i TOFFR. Promień podstawowy ($TC_DP6) aktywnego narzędzia **nie** jest uwzględniany.

Frezowanie kieszeni o skośnych ściankach w przypadku frezowania obwodowego za pomocą CUT3DC

Dla tej korekcji promienia narzędzia 3D, odchyłka promienia frezu jest kompensowana przez dokonanie dosunięcia w kierunku normalnej do powierzchni obrabianej. W tym przypadku płaszczyzna, w której leży powierzchnia czołowa frezu pozostaje niezmieniona, gdy głębokość zagłębienia ISD pozostaje taka sama. Frez np. o mniejszym promieniu względem narzędzia normalnego nie osiągnąłby wówczas dna wnęki, które również stanowi płaszczyznę ograniczającą. W celu automatycznego dosuwu narzędzia sterowanie musi znać płaszczyznę ograniczającą, patrz rozdział "Frezowanie obwodowe 3D z uwzględnieniem powierzchni ograniczającej (CUT3DCC, CUT3DCCD) (Strona 452)".

Advanced Surface / Top Surface

Uwaga

Podczas korzystania z korekcji promienia narzędzia CUT3DCD w połączeniu z licencjonowaną opcją „Advanced Surface” lub „Top Surface” należy przestrzegać zaleceń dotyczących ustawień "Advanced Surface" / "Top Surface”!

Aby sprawdzić ustawione dane, specjalne programy testowe są dostępne za pośrednictwem portalu SIOS:

- Program testowy dla Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392)
- Program testowy dla Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)

11.5.2 Wybór korekcji promienia narzędzia 3D dla frezowania czołowego (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

Korekcja promienia narzędzia 3D dla frezowania czołowego 3D jest wybierana za pomocą działających w sposób modalny poleceń G - CUT3DF, CUT3DFS, CUT3DFF lub CUT3DFD.

Właściwa aktywacja następuje za pomocą G41 lub G42.

Aby móc obliczyć korekcję promienia narzędzia, w przypadku frezowania czołowego 3D wymagana jest definicja normalnej do powierzchni obrabianej. Musi to nastąpić w bloku za pomocą G41 lub G42 poprzez adresy A4, B4, C4 i A5, B5, C5.

Natomiast wyłączenie korekcji promienia narzędzia za pomocą G40.

Składnia

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

Znaczenie

CUT3DFS:	Korekcja promienia narzędzia 3D dla frezowania czołowego ze stałą orientacją. Orientacja narzędzia jest określona przez G17 - G19 i Frame nie mają na nią wpływu.	
CUT3DFF:	Korekcja promienia narzędzia 3D dla frezowania czołowego ze stałą orientacją. Orientacja narzędzia jest określona przez G17 - G19 lub przez kierunek obrócony przez Frame.	
CUT3DF:	Korekcja promienia narzędzia 3D dla frezowania czołowego ze zmianą orientacji (tylko przy aktywnej transformacji 5-osiowej).	
CUT3DFD:	Korekcja promienia narzędzia 3D dla frezowania czołowego ze zmianą orientacji w odniesieniu do narzędzia o zmienionych wymiarach (tylko przy aktywnej transformacji 5-osiowej). Różnica promienia jest określana za pomocą parametru narzędzia $TC_DP15. **Uwaga**: CUT3DFD jest możliwe tylko w połączeniu z funkcją „wygładzania normalnej do powierzchni przy frezowaniu czołowym 3D". Jest ona aktywowana poprzez wywołanie licencjonowanej funkcji „Top Surface" za pomocą cyklu CYCLE832(...) (Strona 810).	
G41/G42 X... Y... Z...:	Włącza korekcję narzędzia Zachowanie przy G41 i przy G42 jest identyczne w przypadku frezowania czołowego 3D. **Uwaga:** Włączenie musi nastąpić w bloku liniowym (G0/G1).	
A4/5=... B4/5=... C4/5=...:	Definicja normalnej do powierzchni obrabianej	
	A4=... B4=... C4=...:	Definicja na początku bloku
	A5=... B5=... C5=...:	Definicja na końcu bloku
ORIC/ORID:	Za pomocą poleceń G ORIC i ORID określane jest zachowanie przy zmianach orientacji na narożach zewnętrznych.	
	ORIC:	Zmiany orientacji na narożach zewnętrznych są nakładane na wstawiony blok okręgu.
	ORID:	Zmiany orientacji są wykonywane przed blokiem okręgu.
G40 X... Y... Z...:	Wyłącza korekcję promienia narzędzia **Uwaga:** Włączenie musi nastąpić w bloku liniowym (G0/G1) za pomocą ruchów osi geometrycznych.	

Uwaga

G41 i G42 mogą być również programowane w blokach bez ruchów postępowych, w których są korekcje ważnych osi geometrycznych. W tym przypadku blok dosunięcia jest pierwszym blokiem ruchu następującym po takim bloku.

Zmiana wariantu korekcji promienia narzędzia 3D przy aktywnej korekcji promienia narzędzia jest ignorowana bez alarmu.

Przykłady

Przykład 1: Frezowanie czołowe 3D za pomocą CUT3DF

Kod programu	Komentarz
N10	; Definicja narzędzia D1:
N20 $TC_DP1[1,1]=121	; Typ narzędzia (frez torusowy)
N30 $TC_DP3[1,1]=20	; Korekcja długości
N40 $TC_DP6[1,1]=5	; Promień
N50 $TC_DP7[1,1]=3	; Promień zaokrąglenia
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; Wybór narzędzia.
N90 TRAORI(1)	; Wybór transformacji orientacji.
N100 B4=-1 C4=1	; Definicja płaszczyzny.
N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0	; Włączenie korekcji narzędzia.
N120 X30	
N130 Y20 A4=1 C4=1	; Naroże zewnętrzne, nowe zdefiniowanie powierzchni.
N140 B3=1 C3=5	; Zmiana orientacji za pomocą ORID.
N150 B3=1 C3=1	; Zmiana orientacji za pomocą ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; Zmiana orientacji za pomocą ORIC
N180 A3=-1 C3=1	; Zmiana orientacji za pomocą ORIC
N190 Y-10 A4=-1 C4=3	; Nowe zdefiniowanie płaszczyzny.
N200 X-20 Y-20 Z10	; Naroże wewnętrzne z blokiem poprzedzającym.
N210 X-30 Y10 A4=1 C4=1	; Naroże wewnętrzne, nowe zdefiniowanie płaszczyzny.
N220 A3=1 B3=0.5 C3=1.7	; Zmiana orientacji za pomocą ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; Zmiana orientacji.
N250 X0 Y30 C4=1	; Ruch po torze, nowa płaszczyzna.
	; Orientacja z programowaniem względnym.
N260 BSPLINE X20 Z15	; Rozpoczęcie Spline, programowanie względne orientacji
N270 X30 Y25 Z18	; Pozostaje aktywny podczas Spline.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; Linia spiralna, orientacja z programowaniem osi
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; Ruch po torze, orientacja, płaszczyzna niestała.
N330 X-20 G40	; Wyłączenie korekcji promienia narzędzia.
N340 M30	

Przykład 2: Fragment programu NC z systemu CAD utworzony za pomocą CUT3DFD

Kod programu	**Komentarz**
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
; Dojazd do pozycji startowej w MKS z uwzględnieniem długości narzędzia.	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; Koniec pozycjonowania w MKS.	
;	
TRAORI(1)	; Wybór transformacji orientacji.
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; Wywołanie cyklu CYCLE832 z: ; tolerancją konturu = 0,01 mm, ; rodzajem obróbki: Top Surface z wygładzaniem ; obróbką wykańczającą z wprowadzeniem tolerancji orientacji ; tolerancją orientacji = 1 stopień
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; Bloki od N09 do N10 się szybką składową ruchu dosuwu do przedmiotu obrabianego ze stałą orientacją.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; W blokach N11 do N21 realizowane są wolne składowe ruchu dosuwu. Narzędzie znajduje się już blisko przedmiotu obrabianego, poza tym aktywne są teraz ustawienia do obróbki form (np. COMPSURF) z cyklu CYCLE832 (z powodu aktywnego G01). Droga po torze tej tak zwanej fazy ustalania przy obróbce form powinna być ok. 1000 razy większa od tolerancji konturu (w tym przykładzie 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	

Kod programu	**Komentarz**
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	
N22 **G41** X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; Od N22 normalna powierzchni jest po raz pierwszy w pełni zdefiniowana poprzez cały blok (tzn. normalna powierzchni jest dostępna na końcu bloku poprzedzającego N21, na początku N22 i na końcu bloku N22). To spełnia warunek konieczny do włączenia korekcji narzędzia za pomocą G41/G42.
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	
...	

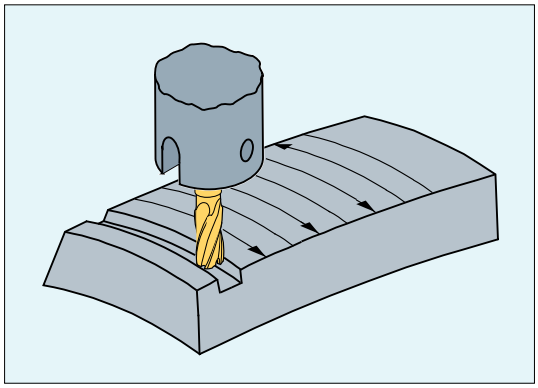
Uwaga

Do włączenia korekcji promienia narzędzia za pomocą G41/G42 w przypadku frezowania czołowego 3D za pomocą CUT3DFD wymagana jest definicja normalnej do powierzchni. Programowanie G41/G42 bez definicji normalnej do powierzchni prowadzi do wyzwolenia alarmu.

Dalsze informacje

Frezowanie czołowe 3D

W przypadku tego rodzaju frezowania 3D wymagany jest wierszowy opisu torów 3D na powierzchni przedmiotu obrabianego. Obliczenia są przeprowadzane zazwyczaj w CAM przy uwzględnieniu kształtu narzędzia i jego wymiarów. Postprocesor zapisuje do programu obróbki – oprócz bloków NC – orientacje narzędzi (przy aktywnej transformacji 5-osiowej) i polecenie G dla pożądanej korekcji narzędzia 3D. Tym samym operator maszyny ma możliwość - w przeciwieństwie do narzędzia używanego do obliczania torów NC - użycia nieznacznie mniejszych narzędzi.



Rysunek 11-3 Frezowanie czołowe

Sposób dosunięcia

Sposób dosunięcia w przypadku korekcji promienia narzędzia 3D jest zawsze NORM.

Zachowanie się na narożach zewnętrznych

Naroża zewnętrzne przy frezowaniu czołowym są traktowane jako okręgi o promieniu 0, przy czym płaszczyzna okręgu wyznaczana jest przez styczną końcową pierwszego bloku i styczną startową drugiego bloku. Dlatego możliwa jest także zmiana orientacji przy przejściu bloku. Na narożu zewnętrznym jest więc zawsze wstawiany okrąg jako element konturu. Metoda punktu przecięcia nie jest dostępna przy frezowaniu czołowym.

Zachowanie przy zmianie orientacji na narożach zewnętrznych

Za pomocą poleceń G - ORIC lub ORID - można ustalać, czy zmiany orientacji, które zostały zaprogramowane pomiędzy dwoma narożami tworzącymi blok ruchu, mają być wykonywane przed rozpoczęciem wstawionego bloku okręgu (ORID) czy równocześnie z nim (ORIC).

Korekcja promienia narzędzia w odniesieniu do narzędzia o zmienionych wymiarach

Korekcja promienia narzędzia 3D odnosząca się do narzędzia o zmienionych wymiarach jest wywoływana za pomocą polecenia CUT3DFD. Stosuje się ją wtedy, gdy programowany kontur odnosi się do toru punktu środkowego narzędzia znormalizowanego, a obróbka odbywa się za pomocą jednego z różniących się narzędzi. Podczas obliczania korekcji promienia narzędzia 3D uwzględniana będzie tylko wartość zużycia promienia aktywnego narzędzia ($TC_DP15) i ewentualnie zaprogramowane przesunięcia korekcji narzędzia OFFN i TOFFR. Promień podstawowy ($TC_DP6) aktywnego narzędzia **nie** jest uwzględniany.

Frezowanie czołowe 3D za pomocą CUT3DFD jest możliwe tylko w połączeniu z funkcją „Wygładzania normalnej do powierzchni przy frezowaniu czołowym 3D”. Jest ona aktywowana poprzez wywołanie funkcji wymagającej licencji „Top Surface” za pomocą cyklu CYCLE832(...) (Strona 810). Aktywacja musi nastąpić **przed** włączeniem korekcji narzędzia za pomocą G41/G42, a zatem nie bezpośrednio przed ingerencją narzędzia, lecz z wyprzedzeniem o długość toru, która odpowiada ok. 1000-krotności tolerancji konturu (np. 1000 x 0,01 mm = 10 mm). Dezaktywacja musi być w odwrotnej kolejności: Najpierw wyłączyć korekcję narzędzia za pomocą G40, a następnie, po długości toru, który odpowiada ok. 1000-krotności tolerancji konturu, dezaktywować przez np. CUT2D (lub podobny).

Aby móc zastosować funkcję „Wygładzania normalnej powierzchni przy frezowaniu czołowym 3D”, wymagane jest dodatkowo udostępnienie funkcji „Interpolacji normalnej powierzchni za pomocą wielomianów”:

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Uwaga

Do frezowania czołowego 3D za pomocą CUT3DFD w połączeniu z „Top Surface” należy przestrzegać zaleceń dotyczących ustawienia "Top Surface”!

Aby sprawdzić ustawione dane, specjalne programy testowe są dostępne (https://support.industry.siemens.com/cs/ww/en/view/109738423) za pośrednictwem portalu SIOS:

11.5.3 Frezowanie obwodowe 3D z uwzględnieniem powierzchni ograniczającej (CUT3DCC, CUT3DCCD)

Przy frezowaniu obwodowym 3D z ciągłą albo stałą zmianą orientacji narzędzia jest często programowany tor punktu środkowego narzędzia dla zdefiniowanego narzędzia znormalizowanego. Ponieważ w praktyce często nie są dostępne odpowiednie narzędzia znormalizowane, można użyć narzędzi, które nie różnią się zbytnio od narzędzia znormalizowanego (≤ 5 %).

Za pomocą CUT3DCCD powierzchnia ograniczająca opisująca zaprogramowane narzędzie znormalizowane uwzględniana jest dla rzeczywistego narzędzia o zmienionych wymiarach. Program NC opisuje tor punktu środkowego narzędzia znormalizowanego.

Za pomocą CUT3DCC uwzględniana jest powierzchnia ograniczająca, którą osiągnęłoby zaprogramowane narzędzie znormalizowane przy zastosowaniu narzędzi cylindrycznych. Program NC opisuje kontur na powierzchni obróbki.

Wektor normalny powierzchni ograniczającej określa się tak jak w przypadku frezowania czołowego 3D z A4, B4, C4 i A5, B5, C5.

Składnia

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Znaczenie

CUT3DCCD:	Korekcja promienia narzędzia 3D dla frezowania obwodowego z uwzględnieniem powierzchni ograniczającej z narzędziem o zmienionych wymiarach po torze punktu środkowego narzędzia: Dosunięcie do powierzchni ograniczającej
CUT3DCC:	Korekcja promienia narzędzia 3D dla frezowania obwodowego z uwzględnieniem powierzchni ograniczającej z korekcją promienia 3D: Kontur na powierzchni obróbki

G41/G42 X... Y... Z...:	Włącza korekcję promienia narzędzia	
	G41:	Korekcja promienia narzędzia na lewo od konturu
	G42:	Korekcja promienia narzędzia na prawo od konturu
	Uwaga: Włączenie musi nastąpić w bloku liniowym (G0/G1).	
CDOF2:	Wyłącza nadzór na kolizję dla frezowania obwodowego 3D	
A4/5=... B4/5=... C4/5=...:	Definicja normalnej do powierzchni obrabianej	
	A4=... B4=... C4=...:	Definicja na początku bloku
	A5=... B5=... C5=...:	Definicja na końcu bloku
G40 X... Y... Z...:	Wyłącza korekcję promienia narzędzia **Uwaga:** Włączenie musi nastąpić w bloku liniowym (G0/G1) za pomocą ruchów osi geometrycznych.	

Uwaga

Polecenia G do wyboru korekcji promienia narzędzia 3D są wykorzystywane w bloku dosuwu, tzn. zazwyczaj w bloku, który zawiera G41 lub G42.

G41 i G42 mogą być również programowane w blokach bez ruchów postępowych, w których są korekcje ważnych osi geometrycznych. W tym przypadku blok dosunięcia jest pierwszym blokiem ruchu następującym po takim bloku.

Zmiana wariantu korekcji promienia narzędzia 3D przy aktywnej korekcji promienia narzędzia jest ignorowana bez alarmu.

Przykład

```
Kod programu                                    Komentarz
N10 $TC_DP1[1,1]=120                            ; Frez walcowy
N20 $TC_DP6[1,1]=10
N30 $TC_DP15[1,1]=-3
...
; Obróbka za pomocą frezu walcowego oraz CUT3DCCD
N110 TRAORI                                     ; Włączenie transformacji.
N120 A4=0 B4=0 C4=1                             ; Definicja normalnej do powierzchni ob-
                                                  rabianej na początku bloku.
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64           ; Aktywacja frezowania obwodowego 3D z
                                                  uwzględnieniem powierzchni ograniczają-
                                                  cej + wyłączenie nadzoru na kolizję.
N150 X20
N160 X30 A45                                    ; Kąt rozwarty ==> brak dosuwu
N170 X40 A-45                                   ; Kąt ostry ==> dosuw
N180 X55
N190 Y10 Z10                                    ; Ruch w kierunku narzędzia
N200 Y20
```

Kod programu	Komentarz
N210 C45	; Tylko zmiana orientacji.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; Zmiana powierzchni.
N240 Y50 G40	; Wyłączenie korekcji promienia narzędzia.
...	

Dalsze informacje

Typ narzędzia

Wykorzystywany jest typ narzędzia (parametr narzędzia $TC_DP1). Dopuszczone są tylko typy frezów o walcowej części trzpieniowej (frez walcowy albo palcowy, jak też frezy torusowe, a w szczególnym przypadku cylindryczny frez do matryc). To odpowiada typom narzędzi o numerach 1 - 399 za wyjątkiem numerów 111 i 155 do 157.

Narzędzia znormalizowane z zaokrągleniem naroża

Zaokrąglenie naroża narzędzia znormalizowanego jest opisywane przez parametr narzędzia $TC_DP7. Z parametru narzędzia $TC_DP16 wynika odchylenie zaokrąglenia naroża rzeczywistego narzędzia w stosunku do narzędzia znormalizowanego.

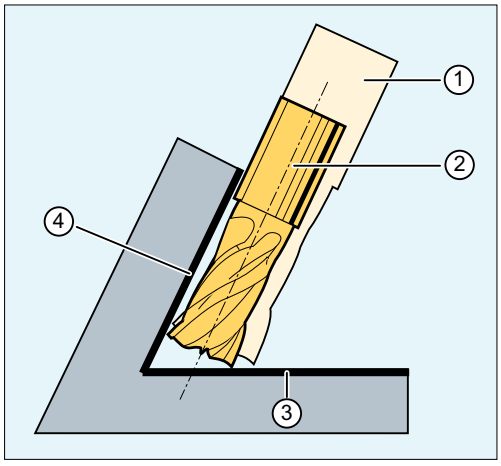
Przykład: Frez torusowy o promieniu zmniejszonym w stosunku do promienia normalnego.

Typ narzędzia	Promień części trzpieniowej (R)	Promień naroża (r)
Narzędzie znormalizowane z zaokrągleniem naroża	R = $TC_DP6	r = $TC_DP7
Narzędzie rzeczywiste z zaokrągleniem naroża Typy narzędzi 121 i 131 frez torusowy (frez palcowy z zaokrągleniem naroża)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16

W tym przykładzie zarówno $TC_DP15 + OFFN, jak też $TC_DP16 są ujemne.

Korekcja promienia narzędzia 3D za pomocą CUT3DCCD: Tor punktu środkowego narzędzia z dosuwem do płaszczyzny ograniczającej

Gdy w porównaniu z pasującym narzędziem znormalizowanym używane jest narzędzie o mniejszym promieniu, wówczas frez jest dosuwany w kierunku wzdłużnym na tyle daleko, aż ponownie dotknie dna wnęki. Dzięki temu z naroża, które jest tworzone przez płaszczyznę obróbki i płaszczyznę ograniczającą, materiał jest wybierany na tyle, na ile pozwala na to narzędzie. Jest to mieszany sposób obróbki, frezowania obwodowego i czołowego. Analogicznie do narzędzia o zmniejszonym promieniu, w przypadku narzędzia o powiększonym promieniu następuje odpowiednie dosunięcie w kierunku przeciwnym.



① Narzędzie znormalizowane
② Dosunięcie narzędzia z mniejszym promieniem do powierzchni ograniczającej
③ Powierzchnia ograniczająca
④ Powierzchnia obróbki

W stosunku do wszystkich innych korekcji narzędzi z grupy G nr 22 podany dla CUT3DCCD parametr narzędzia $TC_DP6 nie ma żadnego znaczenia dla promienia narzędzia i nie wpływa na uzyskiwaną korekcję. Offset korekcji określany jest z sumy wartości zużycia promienia narzędzia (parametr narzędzia $TC_DP15) i offsetu narzędzia OFFN zaprogramowanego w celu obliczenia przesunięcia prostopadłego do powierzchni ograniczającej.

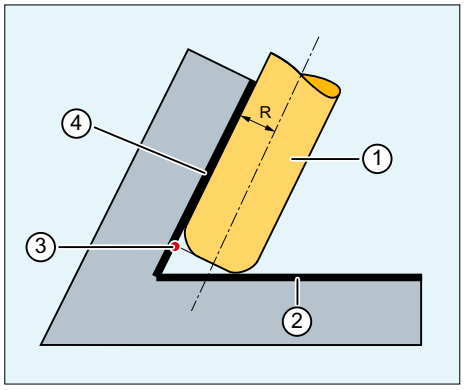
To, czy obrabiana powierzchnia leży na lewo czy na prawo od toru, nie można odczytać ze sporządzonego programu obróbki. Dlatego zakłada się dodatni promień i ujemną wartość zużycia oryginalnego narzędzia. Ujemna wartość zużycia opisuje zawsze narzędzie o zmniejszonej średnicy.

Zastosowanie narzędzi walcowych

Przy zastosowaniu narzędzi walcowych dosuw jest wymagany tylko wtedy, gdy powierzchnia obróbki i płaszczyzna ograniczająca tworzą kąt ostry (mniejszy niż 90 stopni). Gdy są stosowane frezy torusowe (frez palcowy z zaokrąglonym narożem), wówczas zarówno w przypadku kątów ostrych jak i rozwartych wymaga to dosuwu w kierunku wzdłużnym narzędzia.

Korekcja promienia narzędzia 3D za pomocą CUT3DCC: Kontur na powierzchni obróbki

Jeżeli jest aktywne CUT3DCC z frezem torusowym, to zaprogramowany wówczas tor odnosi się do fikcyjnego frezu palcowego o takiej samej średnicy. Wynikający stąd punkt odniesienia toru jest przy zastosowaniu frezu torusowego przedstawiony na poniższym rysunku.



① Frez torusowy
② Powierzchnia ograniczająca
③ Punkt odniesienia toru
④ Powierzchnia obróbki
R Promień części trzpieniowej (promień narzędzia)

Dopuszcza się, aby kąt między powierzchnią obrabianą i powierzchnią ograniczającą również w ramach bloku przechodził z kąta ostrego w rozwarty lub na odwrót.

W stosunku do narzędzia znormalizowanego stosowane rzeczywiste narzędzie może być albo większe albo mniejsze. Przy czym wynikający promień naroża nie może stać się ujemny, a znak wynikającego promienia narzędzia musi pozostać zachowany.

W przypadku CUT3DCC program obróbki NC odnosi się do konturu na obrabianej powierzchni. Podobnie jak w przypadku zwykłej korekcji promienia narzędzia, stosuje się promień sumaryczny, który składa się z następujących elementów:

- Promienia narzędzia (parametr narzędzia $TC_DP6)
- Wartości zużycia (parametr narzędzia $TC_DP15)
- Offsetu narzędzia OFFN zaprogramowanego w celu obliczenia przesunięcia prostopadłego do powierzchni ograniczającej.

Położenie powierzchni ograniczającej określane jest z następującej różnicy:

Wymiary narzędzia znormalizowanego - promień narzędzia (parametr narzędzia $TC_DP6)

Advanced Surface / Top Surface

Uwaga

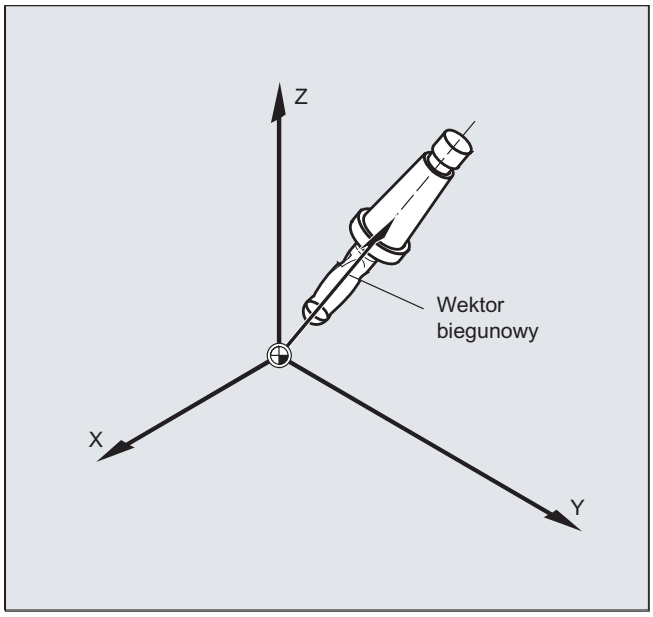
Podczas korzystania z korekcji promienia narzędzia CUT3DCC / CUT3DCCD w połączeniu z licencjonowaną funkcją „Advanced Surface” lub „Top Surface” należy przestrzegać zaleceń dotyczących ustawień "Advanced Surface" / "Top Surface”!

Aby sprawdzić ustawione dane, specjalne programy testowe są dostępne za pośrednictwem portalu SIOS:

- Program testowy dla Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392)
- Program testowy dla Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)

11.6 Orientacja narzędzia (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Pod orientacją narzędzia rozumie się geometryczne ustawienie narzędzia w przestrzeni. W przypadku 5-osiowej maszyny obróbkowej orientacja osi jest ustawialna przez polecenia programowe.



Ruchy ścinania naroży dla orientacji uaktywnione za pomocą `OSD` i `OST` są w zależności od rodzaju interpolacji dla orientacji narzędzia tworzone różnie:

Przy aktywnej interpolacji wektorowej wygładzony przebieg orientacji jest interpolowany również za pomocą interpolacji wektorowej. Natomiast przy aktywnej interpolacji osi obrotowych orientacja jest wygładzana bezpośrednio za pomocą ruchów osi obrotowych.

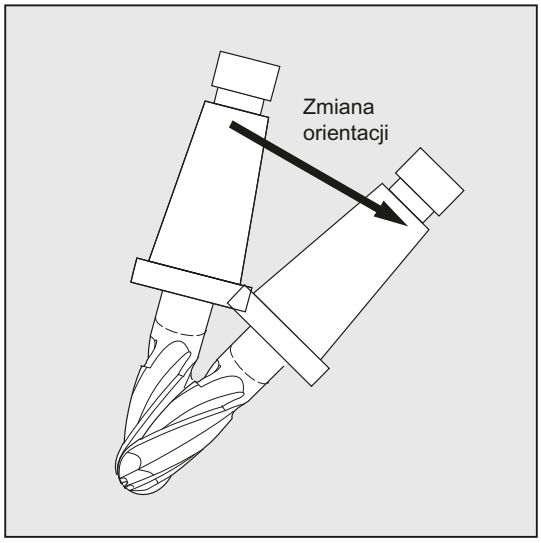
Programowanie

Programowanie zmiany orientacji:

Zmiana orientacji narzędzia może być programowana przez:

- bezpośrednie programowanie osi obrotowych `A`, `B`, `C` (interpolacja osi obrotowej)
- kąt Eulera lub kąt RPY
- wektor kierunkowy (interpolacja wektorowa przez podanie `A3` lub `B3` lub `C3`)
- `LEAD/TILT` (frezowanie czołowe)

Układem współrzędnych odniesienia jest albo układ współrzędnych maszyny (`ORIMKS`) albo aktualny układ współrzędnych obrabianego przedmiotu (`ORIWKS`).



Programowanie orientacji narzędzia:

ORIC:	Orientacja i ruch po torze równolegle
ORID:	Orientacja i ruch po torze kolejno
OSOF:	Brak wygładzania orientacji
OSC:	Orientacja stała
OSS:	Wygładzanie orientacji tylko na początku bloku
OSSE:	Wygładzanie orientacji na początku i końcu bloku
ORIS:	Prędkość zmiany orientacji przy włączonym wygładzaniu orientacji w stopniach na mm (obowiązuje dla OSS i OSSE)
OSD:	Wygładzanie orientacji przez zadanie drogi wygładzania przy pomocy danej ustawczej: SD42674 $SC_ORI_SMOOTH_DIST
OST:	Wygładzanie orientacji przez zadanie tolerancji kątowej w stopniach przy interpolacji wektorowej za pomocą danej ustawczej: SD42676 $SC_ORI_SMOOTH_TOL Przy interpolacji osi obrotowych zadana tolerancja jest przyjmowana, jako maksymalne odchylenie osi orientacji.

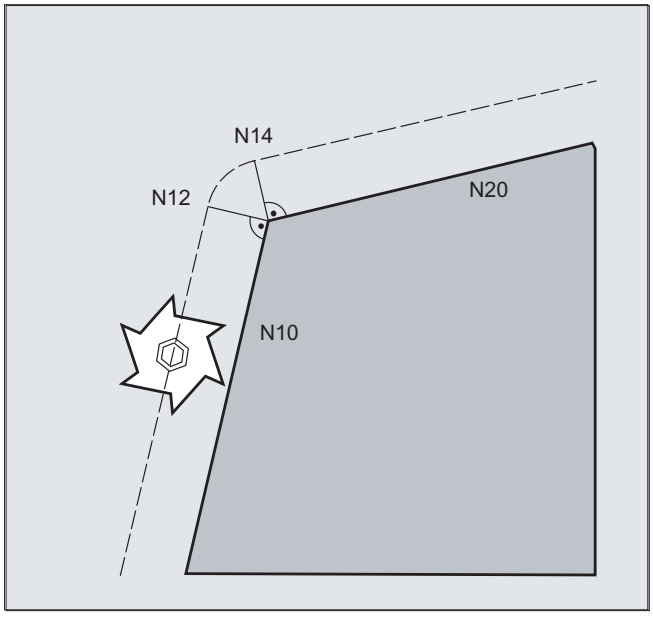
Uwaga

Wszystkie polecenia do wygładzania orientacji narzędzia (OSOF, OSC, OSS, OSSE, OSD i OST) są ujęte w grupie G nr 34. Działają one modalnie, tzn. zawsze może działać tylko jedno z tych poleceń.

Przykłady

Przykład 1: ORIC

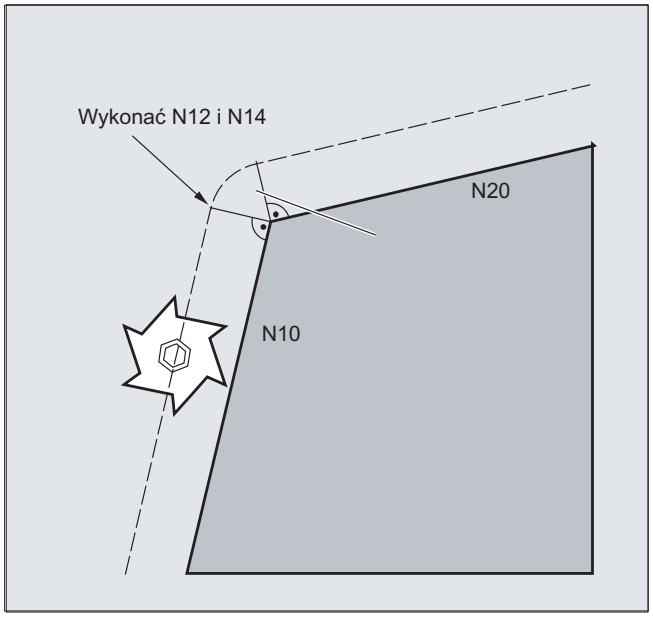
Jeżeli między blokami ruchu N10 i N20 jest zaprogramowanych dwa albo więcej bloków ze zmianami orientacji (np. A2=... B2=... C2=...) i jest aktywne ORIC, wówczas wstawiony blok ruchu kołowego jest odpowiednio do wielkości zmian kąta dzielony na te bloki pośrednie.



Kod programu	Komentarz
ORIC	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 C2=… B2=…	; Blok okręgu, który jest wstawiany na narożu zewnętrznym, dzieli się na N12 i N14, odpowiednio do zmiany orientacji. Ruch kołowy i zmiana orientacji są przy tym wykonywane równolegle.
N14 C2=… B2=…	
N20 X =…Y=… Z=… G1 F200	

Przykład 2: ORID

Gdy jest aktywne ORID, wówczas wszystkie bloki są wykonywane między obydwoma blokami ruchu postępowego na końcu pierwszego bloku ruchu postępowego. Blok okręgu ze stałą orientacją jest wykonywany bezpośrednio przed drugim blokiem ruchu postępowego.



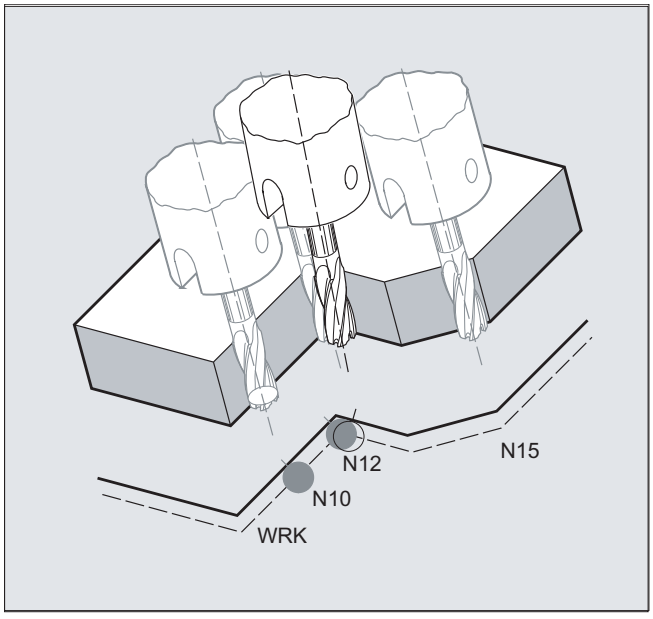
Kod programu	Komentarz
ORID	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 A2=… B2=… C2=…	; Bloki N12 i N14 są wykonywane na końcu N10. Następnie jest realizowany blok okręgu z aktualną orientacją.
N14 M20	; Funkcje pomocnicze itd.
N20 X… Y… Z…	

Uwaga

Dla rodzaju zmiany orientacji na narożu zewnętrznym właściwe jest polecenie programowe, które jest aktywne w pierwszym bloku ruchu postępowego naroża zewnętrznego.

Bez zmiany orientacji: Jeżeli orientacja nie jest na końcu bloku zmieniana, wówczas przekrój narzędzia jest okręgiem, który dotyka obydwu konturów.

Przykład 3: Zmiana orientacji na narożu wewnętrznym



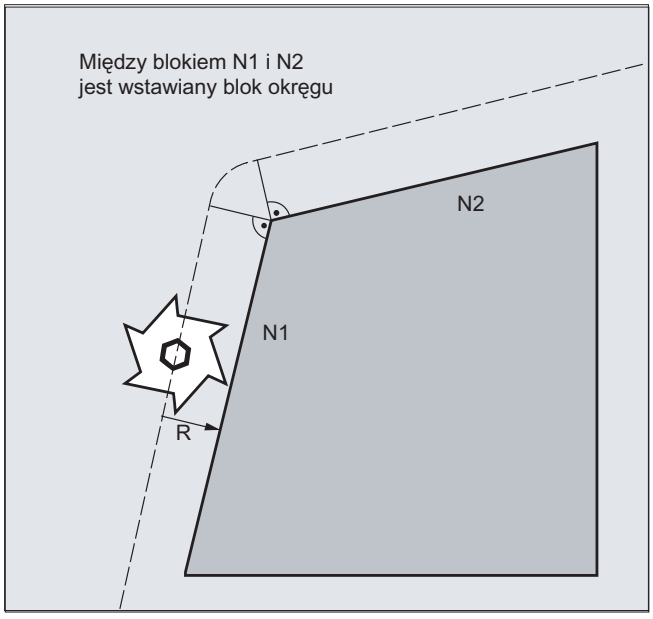
Kod programu
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…

Dodatkowe informacje

Zachowanie się na narożnikach zewnętrznych

Na narożniku zewnętrznym jest zawsze wstawiany blok okręgu o promieniu frezu.

Przy pomocy poleceń programowych `ORIC` lub `ORID` można ustalić, czy zmiany orientacji, które zostały zaprogramowane między blokiem `N1` i `N2`, mają być wykonywane przed rozpoczęciem wstawionego bloku okręgu czy równocześnie z nim.



Jeżeli na narożach zewnętrznych jest konieczna zmiana orientacji, wówczas może ona nastąpić do wyboru równolegle do interpolacji albo oddzielnie od ruchu po torze.

W przypadku `ORID` są najpierw wykonywane wstawione bloki bez ruchu po torze. Blok okręgu jest wstawiany bezpośrednio przed drugim z obydwu bloków ruchu, przy pomocy których jest tworzone naroże.

Jeżeli na narożniku zewnętrznym jest wstawionych wiele bloków orientacji i jest wybrane `ORIC`, wówczas ruch kołowy jest dzielony na poszczególne wstawione bloki odpowiednio do zawartych w nich wielkości zmian orientacji.

Wygładzanie orientacji przy pomocy OSD lub OST

Przy wygładzaniu przy pomocy `G642` maksymalne odchylenie dla osi konturowych i osi orientacji nie może być bardzo różne. Mniejsza tolerancja z obydwu określa kształt ruchu wygładzania wzgl. tolerancję kąta, stosunkowo duże wygładzanie przebiegu orientacji, bez konieczności większych odchyleń od konturu.

Przez uaktywnienie `OSD` lub `OST` jest możliwe, z zadaną drogą wygładzania lub tolerancją kątową dokonywanie "znakomitego" wygładzania bardzo małych odchyleń przebiegu orientacji bez dużych odchyleń od konturu.

Uwaga

W odróżnieniu od wygładzania konturu (i od przebiegu orientacji) przy pomocy `G642` nie jest przy wygładzaniu orientacji przy pomocy `OSD` lub `OST` tworzony własny blok, lecz ruch wygładzania jest wstawiany bezpośrednio dla zaprogramowanych oryginalnych bloków.

Przy pomocy `OSD` lub `OST` nie mogą być wygładzane przejścia między blokami, przy których następuje zmiana rodzaju interpolacji dla orientacji narzędzia (wektor → oś obrotowa, oś obrotowa → wektor). Te przejścia między blokami mogą ewentualnie być wygładzane przy pomocy zwykłych funkcji ścinania naroży `G641`, `G642` wzgl. `G643`.

11.7 Dowolne nadawanie numerów D, numer ostrza

11.7.1 Dowolne nadawanie numerów D , numer ostrza (adres CE)

Numer D

Numery D mogą być stosowane jako numery korekcji. Dodatkowo można przez adres CE adresować numer ostrza. Przez parametr systemowy $TC_DPCE można zapisać numer ostrza.

Ustawienie domyślne: nr korekcji == nr ostrza

Przez dane maszynowe ustala się maksymalną liczbę numerów D (numerów ostrza) i maksymalną liczbę ostrzy na narzędzie (→ producent maszyny). Kolejne polecenia mają sens tylko wtedy, gdy maksymalny numer ostrza (MD18105) został ustalony większy, niż liczba ostrzy na narzędzie (MD18106). Proszę przestrzegać danych producenta maszyny.

Uwaga

Oprócz względnego nadawania numerów D, numery D mogą być również nadawane jako "płaskie" wzgl. "absolutne" numery D (1-32000) bez odniesienia do numeru T (w ramach funkcji "Płaska struktura numerów D").

Literatura

Podręcznik działania Funkcje podstawowe; Korekcja narzędzia (W1)

11.7.2 Dowolne nadawanie numerów D: sprawdzenie numerów D (CHKDNO)

Przy pomocy polecenia `CKKDNO` sprawdzamy, czy istniejące numery D są jednoznacznie nadane. Numery D wszystkich narzędzi zdefiniowanych w ramach jednej jednostki TO mogą wystąpić tylko jeden raz. Narzędzia zamienne nie są przy tym uwzględniane.

Składnia

```
state=CHKDNO(Tno1,Tno2,Dno)
```

Znaczenie

`state`:	= TRUE:	Numery D zostały jednoznacznie nadane dla sprawdzonego zakresu.
	=FALSE:	Nastąpiła kolizja numerów D albo parametryzacja jest niepoprawna. Przez Tno1, Tno2 i Dno są przekazywane parametry, które prowadziły do kolizji. Ewaluacja tych danych jest możliwa w programie obróbki.
`CHKDNO(Tno1,Tno2)`:	Są sprawdzane wszystkie numery D wymienionych narzędzi.	

CHKDNO(Tno1):	Są sprawdzane wszystkie numery D dla Tno1 w stosunku do wszystkich innych narzędzi.
CHKDNO:	Są sprawdzane wszystkie numery D wszystkich narzędzi w stosunku do wszystkich innych narzędzi.

11.7.3 Dowolne nadawanie numerów D: Zmiana nazw numerów D (GETDNO, SETDNO)

Numery D muszą być nadawane jednoznacznie. Dwa różne ostrza jednego narzędzia nie mogą mieć tego samego numeru D.

GETDNO

To polecenie daje numer D określonego ostrza (ce) narzędzia o T-numerze t. Jeżeli nie istnieje numer D dla podanych parametrów, wówczas następuje ustawienie d=0. Jeżeli numer D jest nie poprawny, jest zwracana wartość większa, niż 32000.

SETDNO

Przy pomocy tego polecenia przyporządkowuje się wartość d do numeru D ostrza ce narzędzia t. Przez state jest zwracany wynik tej instrukcji (TRUE albo FALSE). Jeżeli nie istnieje zestaw danych do wprowadzonych parametrów, jest zwracane FALSE. Błędy składni powodują alarm. Numeru D nie można bezpośrednio ustawić na 0.

Składnia

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Znaczenie

d:	Numer D ostrza narzędzia
t:	Numer T narzędzia
ce:	Numer ostrza (numer CE) narzędzia
state:	Podaje, czy można było bezbłędnie wykonać polecenie (TRUE albo FALSE).

Przykład, zmiana nazwy numeru D

Programowanie	Komentarz
$TC_DP2[1,2]=120	
$TC_DP3[1,2] = 5.5	
$TC_DPCE[1,2] = 3	; Numer ostrza CE
...	
N10 def int DNrStary, DNrNowy = 17	
N20 DNrStary = GETDNO(1,3)	
N30 SETDNO(1,3,DNrNowy)	

Przez to do ostrza CE=3 jest przyporządkowywana nowa wartość D 17. Teraz sięganie do danych tego ostrza następuje przez numer D 17; zarówno przez parametry systemowe, jak też w programowaniu przy pomocy adresu NC.

11.7.4 Dowolne nadawanie numerów D: Określenie numeru T do zadanego numeru D (GETACTTD)

Przy pomocy polecenia `GETACTTD` określa się do absolutnego numeru D przynależny numer T. Nie ma badania na jednoznaczność. Gdy jest wiele takich samych numerów D w ramach jednej jednostki TO, jest zwracany numer T pierwszego znalezionego narzędzia. Przy stosowaniu "płaskich" numerów D stosowanie tego polecenia nie ma sensu, ponieważ zawsze zostanie zwrócona wartość "1" (w zarządzaniu danymi nie ma numeru T).

Składnia

```
status=GETACTTD(Tnr,Dnr)
```

Znaczenie

`Dnr`:	Numer D, dla którego ma być szukany numer T.	
`Tnr`:	Znaleziony numer T	
`status`:	Wartość:	Znaczenie:
	0	Numer T został znaleziony. Tnr otrzymuje wartość numeru T.
	-1	Do podanego numeru D nie istnieje numer T; Tnr=0.
	-2	Numer D nie jest absolutny. Tnr zawiera wartość pierwszego znalezionego narzędzia, który zawiera numer o wartości Dnr.
	-5	Funkcja nie mogła zostać wykonana z innego powodu.

11.7.5 Dowolne nadawanie numerów D: ustawienie nie obowiązywania numerów D (DZERO)

Polecenie `DZERO` służy do wsparcia podczas przezbrajania. Tak zaznaczone zestawy danych korekcyjnych nie są już sprawdzane przez polecenie `CHKDNO`. Aby je ponownie udostępnić, numer D musi zostać ponownie ustawiony przy pomocy `SETDNO`.

Składnia

```
DZERO
```

Znaczenie

`DZERO`:	Zaznacza wszystkie numery D jednostki TO jako nie obowiązujące.

11.8 Kinematyka nośnika narzędzi

Warunki

Nośnik narzędzi może orientować narzędzie tylko wówczas we wszystkich możliwych kierunkach w przestrzeni, gdy

- są dwie osie obrotowe V_1 i V_2.
- osie obrotowe są prostopadłe w stosunku do siebie.
- oś wzdłużna narzędzia jest prostopadła do drugiej osi obrotu V_2.

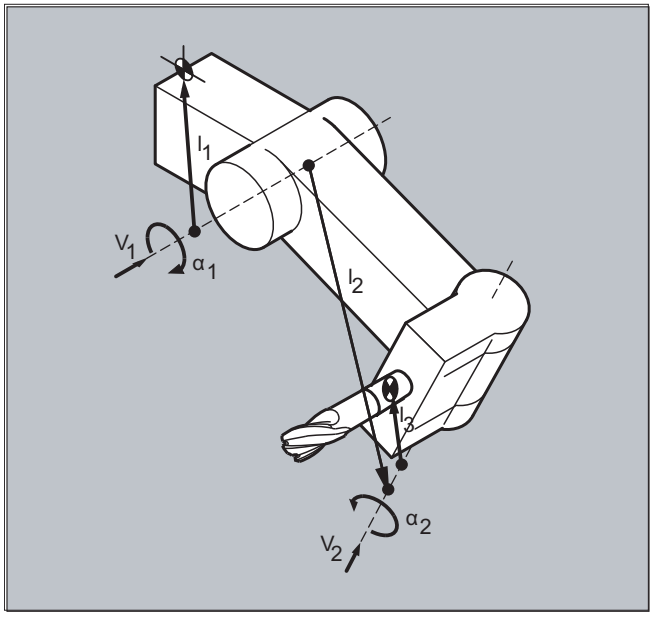
Dodatkowo w przypadku maszyn, przy których muszą dawać się ustawiać wszystkie możliwe orientacje, obowiązuje następujący wymóg:

- orientacja narzędzia musi być prostopadła do pierwszej osi obrotowej V_1.

Funkcja

Kinematyka nośnika narzędzi o maksymalnie dwóch osiach obrotu v_1 lub v_2 jest opisywana przez 17 zmiennych systemowych $TC_CARR1[m] do $TC_CARR17[m]. Opis nośnika narzędzi składa się z:

- wektorowej odległości od pierwszej osi obrotu do punktu odniesienia nośnika narzędzi I_1, wektorowej odległości od pierwszej do drugiej osi obrotu I_2, wektorowej odległości od drugiej osi obrotu do punktu odniesienia narzędzia I_3.
- wektorów biegunowych obydwu osi obrotowych V_1, V_2.
- kątów obrotu α1, α2 wokół obydwu osi. Kąty obrotu są liczone w kierunku ruchu wskazówek zegara patrząc w kierunku wektorów osi obrotu.



Dla maszyn z **rozdzieloną kinematyką** (obrotowe jest zarówno narzędzie, jak i obrabiany przedmiot) zmienne systemowe są rozszerzone o wpisy $TC_CARR18[m] do $TC_CARR23[m].

Parametry

Funkcja zmiennych systemowych dla orientowanych nośników narzędzi			
Określenie	**Składowa x**	**Składowa y**	**Składowa z**
l_1 wektor przesunięcia	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
l_2 wektor przesunięcia	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
v_1 oś obrotowa	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 oś obrotowa	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
α_1 kąt obrotu α_2 kąt obrotu	$TC_CARR13[m] $TC_CARR14[m]		
l_3 wektor przesunięcia	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

Rozszerzenia zmiennych systemowych dla orientowanych nośników narzędzi			
Określenie	**Składowa x**	**Składowa y**	**Składowa z**
l_4 wektor przesunięcia	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
Identyfikator osi oś obrotowa v_1 oś obrotowa v_2	Identyfikatory osi obrotowych v_1 i v_2 (ustawieniem domyślnym jest zero) $TC_CARR21[m] $TC_CARR22[m]		
Typ kinematyki **T**ool **P**art **M**ixed mode	$TC_CARR23[m]		
	Typ kinematyki T ->	Typ kinematyki P ->	Typ kinematyki M
	Tylko narzędzie jest obrotowe (ustawienie domyślne)	Tylko obrabiany przedmiot jest obrotowy	Obrabiany przedmiot & narzędzie są obrotowe
Przesunięcie osi obrotowej v_1 osi obrotowej v_2	Kąty w stopniach osi obrotowych v_1 i v_2 przy przyjęciu położenia podstawowego $TC_CARR24[m] $TC_CARR25[m]		
Przesunięcie kątowe osi obrotowej v_1 osi obrotowej v_2	Przesunięcie połączenia Hirtha w stopniach osi obrotowych v_1 i v_2 $TC_CARR26[m] $TC_CARR27[m]		
Przyrost kątowy v_1 oś obrotowa v_2 oś obrotowa	Przyrost połączenia Hirtha w stopniach osi obrotowych v_1 i v_2 $TC_CARR28[m] $TC_CARR29[m]		
Pozycja min Oś obrotowa v_1 Oś obrotowa v_2	Limit softwareowy pozycji minimalnej osi obrotowych v_1 i v_2 $TC_CARR30[m] $TC_CARR31[m]		
Pozycja max Oś obrotowa v_1 Oś obrotowa v_2	Limity softwareowe pozycji maksymalnej osi obrotowych v_1 i v_2 $TC_CARR32[m] $TC_CARR33[m]		
Nazwa nośnika narzędzi	Zamiast liczby nośnik narzędzi może otrzymać nazwę. $TC_CARR34[m]		

Rozszerzenia zmiennych systemowych dla orientowanych nośników narzędzi			
Użytkownik: Nazwa osi 1 Nazwa osi 2 Oznaczenie	Zamierzone zastosowanie przez użytkownika w ramach cykli pomiarowych. $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
Pozycja	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
Przesunięcie dokładne	Parametry, które **mogą być dodawane do wartości w parametrach bazowych**.		
l_1 wektor przesunięcia	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 wektor przesunięcia	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 wektor przesunięcia	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 wektor przesunięcia	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 oś obrotowa	$TC_CARR64[m]		
v_2 oś obrotowa	$TC_CARR65[m]		

Uwaga

Objaśnienia do parametrów

Przy pomocy "m" jest każdorazowo podawany numer opisywanego nośnika narzędzi

$TC_CARR47 do $TC_CARR54, jak też $TC_CARR61 do $TC_CARR63 nie są zdefiniowane i przy próbie dostępu w celu zapisu albo odczytu prowadzą do alarmu.

Punkty początkowe wzgl. końcowe wektorów odległości na osiach mogą być dowolnie wybierane. Kąty obrotu α_1, α_2 wokół obydwu osi są w podstawowym stanie nośnika narzędzi zdefiniowane jako 0°. Kinematyka nośnika narzędzi może w ten sposób być opisywana na dowolnie wiele sposobów.

Nośniki narzędzi o tylko jednej albo nie posiadające osi obrotowych mogą być opisywane przez ustawienie na zero wektorów kierunkowych jednej albo obydwu osi obrotowych.
W przypadku nośnika narzędzi bez osi obrotowej wektory odległości działają, jak dodatkowe korekcje narzędzia, na których składowe nie wpływa przełączanie płaszczyzn obróbki (G17 do G19).

Rozszerzenia parametrów

Parametry osi obrotowych

Zmienne systemowe zostały rozszerzone o wpisy $TC_CARR24[m] do $TC_CARR33[m] i opisane następująco:

Przesunięcie osi obrotowych v_1, v_2	Zmiana pozycji osi obrotowej v_1 lub v_2 przy podstawowym położeniu orientowanego nośnika narzędzi.
Przesunięcie kątowe/ przyrost kątowy Osie obrotowe v_1, v_2	Przesunięcie lub przyrost kątowy połączenia Hirtha osi obrotowych v_1 i v_2. Zaprogramowany albo obliczony kąt jest zaokrąglany do najbliższej wartości, która wynika przy n całkowitym z phi = s + n * d.
Pozycja minimalna i maksymalna Osie obrotowe v_1, v_2	Pozycja minimalna/maksymalna osi obrotowej, kąt graniczny (Software-Limit) osi obrotowej v1 i v2.

Parametry dla użytkownika

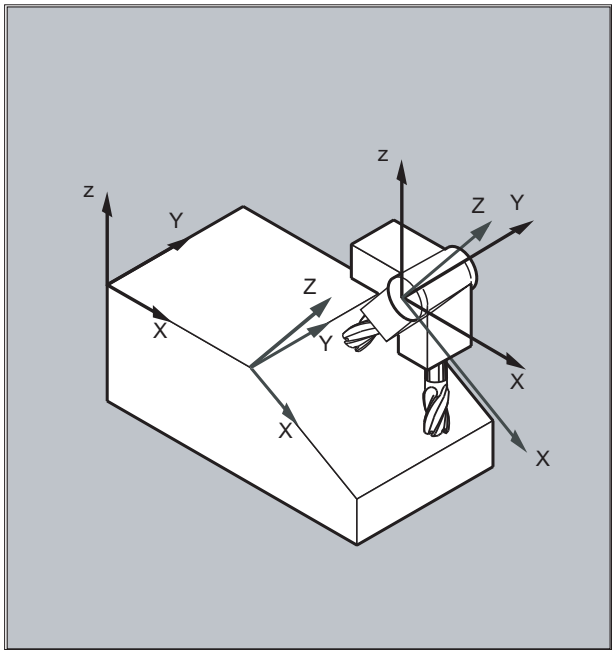
$TC_CARR34 do $TC_CARR40 zawierają parametry, które są do dowolnej dyspozycji użytkowników i do w. opr. 6.4 standardowo nie są dalej wykorzystywane w ramach NCK albo nie mają znaczenia.

Parametry przesunięcia dokładnego

$TC_CARR41 bis $TC_CARR65 zawierają parametry przesunięcia dokładnego, które mogą być dodawane do wartości w parametrach bazowych. Wartość przesunięcia dokładnego przyporządkowana do parametru bazowego wynika, gdy do numeru parametru zostanie dodana wartość 40.

Przykład

Nośnik narzędzi zastosowany w poniższym przykładzie daje się kompletnie opisać przez obrót wokół osi Y.



Kod programu	Komentarz
N10 $TC_CARR8[1]=1	; Definicja składowej Y pierwszej osi obrotu nośnika narzędzi 1.
N20 $TC_DP1[1,1]= 120	; Definicja frezu trzpieniowego.
N30 $TC_DP3[1,1]=20	; Definicja frezu trzpieniowego o długości 20 mm.
N40 $TC_DP6[1,1]=5	; Definicja frezu trzpieniowego o promieniu 5 mm.
N50 ROT Y37	; Definicja frame z obrotem 37° wokół osi Y.
N60 X0 Y0 Z0 F10000	; Dosunięcie do pozycji wyjściowej
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; Ustawienie korekcji promienia, korekcji długości narzędzia w obróconym frame, wybór nośnik narzędzi 1, narzędzie 1.
N80 X40	; Przeprowadzenie obróbki przy obrocie 37°.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Dodatkowe informacje

Kinematyka rozłączona

Dla maszyn z rozdzieloną kinematyką (obrotowe jest zarówno narzędzie, jak i obrabiany przedmiot) zmienne systemowe są rozszerzone o wpisy `$TC_CARR18[m]` do `$TC_CARR23[m]`.

Obrotowy stół narzędziowy składający się z:

- odległości wektorowej drugiej osi obrotu V_2 od punktu odniesienia obrotowego stołu narzędziowego I_4 trzeciej osi obrotowej.

Osie obrotowe składające się z:

- obydwu identyfikatorów kanału dla odniesienia osi obrotowych V_1 i V_2, do których pozycji ewentualnie następuje dostęp przy określaniu orientacji orientowanego nośnika narzędzi.

Typ kinematyki z jedną z wartości T, P albo M:

- Typ kinematyki T: Tylko narzędzie jest obrotowe.
- Typ kinematyki P: Tylko obrabiany przedmiot jest obrotowy.
- Typ kinematyki M: Obrotowe jest narzędzie i obrabiany przedmiot.

Skasowanie danych nośnika narzędzi

Przy pomocy `$TC_CARR1[0]=0` mogą być kasowane dane wszystkich zestawów danych nośników narzędzi.

Typ kinematyki `$TC_CARR23[T]=T` musi zostać podany jako jedna z trzech dopuszczalnych liter dużych albo małych (T,P,M) i z tego powodu nie powinien być kasowany.

Zmiana danych nośnika narzędzi

Każdą z opisanych wartości można zmienić przez przyporządkowanie nowej wartości w programie obróbki. Każdy znak inny niż T, P albo M prowadzi do alarmu przy próbie uaktywnienia orientowalnego nośnika narzędzi.

Odczyt danych nośnika narzędzi

Każda z zapisanych wartości może zostać przeczytana przez przyporządkowanie do zmiennej w programie obróbki.

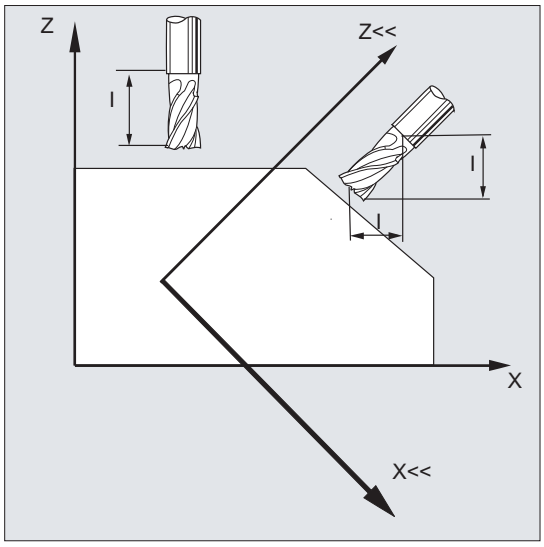
Przesunięcia dokładne

Niedopuszczalna wartość przesunięcia dokładnego zostaje rozpoznana dopiero wtedy, gdy zostanie uaktywniony orientowalny nośnik narzędzi, który zawiera taką wartość i jednocześnie dana nastawcza SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE.

Wielkość dopuszczalnego przesunięcia dokładnego jest przez dane maszynowe ograniczana do maksymalnie dopuszczalnej wartości.

11.9 Korekcja długości narzędzia dla orientowalnych nośników narzędzi (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Wraz ze zmienioną orientacją narzędzia w przestrzeni zmieniają się również składowe jego długości.



Po przezbrojeniu, np. przez ręczne ustawienie albo wymianę nośnika narzędzi ze stałą orientacją w przestrzeni, komponenty długości narzędzia muszą zostać ponownie zdefiniowane. Następuje to przy pomocy poleceń drogowych `TCOABS` i `TCOFR`.

W przypadku orientowalnego nośnika narzędzi aktywnego Frame można przy wyborze narzędzia przy pomocy `TCOFRZ`, `TCOFRY` i `TCOFRX` określić kierunek, w którym narzędzie ma być zwrócone.

Składnia

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

Znaczenie

`TCARR=[<m>]`:	Zażądanie nośnika narzędzi o numerze "m"
`TCOABS`:	Obliczenie składowych długości narzędzia z aktualnego zorientowania nośnika narzędzi
`TCOFR`:	Określenie składowych długości narzędzia ze zorientowania aktywnego Frame
`TCOFRZ`:	Orientowalny nośnik narzędzi z aktywnego Frame, którego narzędzie wskazuje w kierunku Z

TCOFRY:	Orientowalny nośnik narzędzi z aktywnego Frame, którego narzędzie wskazuje w kierunku Y
TCOFRX:	Orientowalny nośnik narzędzi z aktywnego Frame, którego narzędzie wskazuje w kierunku X

Dodatkowe informacje

Korekcja długości narzędzia z orientacji nośnika (TCOABS)

TCOABS oblicza korekcję promienia narzędzia z aktualnych kątów orientacji nośnika narzędzi; zapisanych w zmiennych systemowych $TC_CARR13 i $TC_CARR14.

Odnośnie definicji kinematyki nośnika narzędzi przy pomocy zmiennych systemowych patrz "Kinematyka nośnika narzędzi (Strona 466) ".

W celu ponownego obliczenia korekcji długości narzędzia przy zmianie frame, wrzeciono musi zostać jeszcze raz wybrane.

Kierunek narzędzia z aktywnego frame

Orientowalny nośnik narzędzi można tak ustawić, że narzędzie będzie zwrócone w następujących kierunkach:

- przy pomocy TCOFR lub TCOFRZ w kierunku Z
- przy pomocy TCOFRY w kierunku Y
- przy pomocy TCOFRX w kierunku X

Przełączenie między TCOFR i TCOABS powoduje nowe obliczenie korekcji długości narzędzia.

Zażądanie nośnika narzędzi (TCARR)

Przy pomocy TCARR następuje z numerem nośnika narzędzi m zażądanie jego danych geometrycznych (pamięć korekcji).

Przy pomocy m=0 następuje cofnięcie wyboru aktywnego nośnika narzędzi.

Dane geometryczne nośnika narzędzi stają się aktywne dopiero po wywołaniu narzędzia. Wybrane narzędzie pozostaje aktywne po zmianie nośnika narzędzi.

Aktualne dane geometryczne nośnika narzędzi mogą być również definiowane w programie obróbki przez odpowiednie zmienne systemowe.

Nowe obliczenie korekcji długości narzędzia (TCOABS) przy zmianie frame

W celu ponownego obliczenia korekcji długości narzędzia przy zmianie frame, wrzeciono musi zostać jeszcze raz wybrane.

Uwaga

Orientacja narzędzia musi zostać ręcznie dopasowana do aktywnego frame.

Przy obliczaniu korekcji długości narzędzia są w ramach kroku pośredniego obliczane również kąty obrotu nośnika narzędzi. Ponieważ w przypadku nośników narzędzi o dwóch osiach obrotu zazwyczaj istnieją dwie pary kątów obrotu, przy pomocy których orientacja narzędzia

może zostać dopasowana do aktywnego frame, wartości zapisane w zmiennych systemowych muszą przynajmniej w przybliżeniu odpowiadać mechanicznie ustawionym kątom obrotu.

Uwaga

Orientacja narzędzia

Kątów skrętu obliczonych przez zorientowanie frame sterowanie nie jest w stanie sprawdzić pod względem możliwości ustawienia na maszynie

Jeżeli osie obrotu nośnika narzędzi są konstrukcyjnie tak usytuowane, że orientacja narzędzia, obliczona przez orientację frame, nie może zostać osiągnięta, zostanie wyprowadzony alarm.

Kombinacja korekcji dokładnej narzędzia i funkcji dotyczących korekcji długości narzędzia przy ruchomych nośnikach narzędzi jest niedopuszczalna. Przy próbie równoczesnego wywołania obydwu funkcji następuje komunikat błędu.

Przy pomocy `TOFRAME` jest możliwe zdefiniowanie frame na podstawie kierunku zorientowania wybranego nośnika narzędzi. Dokładniejsze informacje patrz punkt "Frame".

Przy aktywnej transformacji orientacji (transformacja 3-, 4-, 5-osiowa) można wybrać nośnik narzędzi o orientacji odbiegającej od położenia zerowego, bez wyprowadzenia przy tym alarmu.

Przekazywane parametry cykli standardowych i pomiarowych

Dla przekazywanych parametrów cykli standardowych i pomiarowych obowiązują zdefiniowane zakresy wartości.

W przypadku wartości kątowych zakres wartości jest ustalony następująco:

- Obrót wokół 1. osi geometrycznej: -180 stopni do +180 stopni
- Obrót wokół 2. osi geometrycznej: -90 stopni do +90 stopni
- Obrót wokół 3. osi geometrycznej: -180 stopni do +180 stopni

Patrz punkt Frame "Obrót programowany (ROT, AROT, RPL)".

Uwaga

Przy przekazywaniu wartości kątowych do cyklu standardowego lub pomiarowego należy przestrzegać:

Wartości mniejsze, niż dokładność obliczania przez NC należy zaokrąglić do zera!

Dokładność obliczania przez NC dla pozycji końcowej jest ustalona w danej maszynowej:

MD10210 $MN_INT_INCR_PER_DEG

11.10 Korekcja długości narzędzia online (TOFFON, TOFFOF)

Poprzez zmienną systemową $AA_TOFF[<n>] można trójwymiarowo w czasie rzeczywistym nałożyć efektywne długości narzędzia odpowiednio do trzech jego kierunków.

Jako indeks <n> są stosowane trzy identyfikatory osi geometrycznych. Przez to liczba aktywnych kierunków korekcji jest ustalona przez osie geometryczne aktywne w tym samym czasie.

Wszystkie korekcje mogą być równocześnie aktywne.

Funkcja korekcji długości narzędzia online daje się stosować przy:

- Transformacja orientacji TRAORI
- Orientowalne nośniki narzędzi TCARR

Uwaga

Korekcja długości narzędzia online jest **opcją,** która przedtem musi zostać udostępniona. Funkcja ta ma sens tylko w połączeniu z aktywną transformacją orientacji albo aktywnym orientowalnym nośnikiem narzędzi.

Składnia

```
TRAORI
TOFFON(<kierunek korekcji>[,<wartość przesunięcia])
WHEN TRUE DO $AA_TOFF[<kierunek korekcji>]              ; W akcjach synchronicznych.
...
TOFFOF(<kierunek korekcji>)
```

Dalsze objaśnienia dot. programowania korekcji długości narzędzia online w akcjach synchronicznych ruchu patrz "Akcje synchroniczne (Strona 631)".

Znaczenie

TOFFON:	**Uaktywnienie** korekcji długości narzędzia online	
	<kierunek korekcji>:	Kierunek narzędzia (X, Y, Z), w którym ma działać korekcja długości narzędzia online.
	<wartość przesunięcia:	Przy uaktywnieniu można dla odpowiedniego kierunku korekcji podać przesunięcie, które jest natychmiast realizowany.
TOFFOF:	**Cofnięcie** korekcji długości narzędzia online Wartości korekcji w podanym kierunku korekcji są cofane i jest wyzwalane zatrzymanie przebiegu wyprzedzającego.	

Przykłady

Przykład 1: Wybór korekcji długości narzędzia

Kod programu	Komentarz
MD21190 $MC_TOFF_MODE =1	; Następuje ruch do wartości absolutnych.
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)	; Transformacja wł.
N20 TOFFON(Z)	; Uaktywnienie korekcji długości narzędzia online dla kierunku Z narzędzia.
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; Dla kierunku Z narzędzia jest interpolowana korekcja długości narzędzia wynosząca 10.
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	; Przyporządkowanie aktualnej korekcji w kierunku X.
N120 TOFFON(X,-XOFFSET) G4 F5	; Dla kierunku X narzędzia korekcja długości narzędzia jest ponownie cofana na 0:

Przykład 2: Cofnięcie wyboru korekcji długości narzędzia

Kod programu	Komentarz
N10 TRAORI(1)	; Transformacja wł.
N20 TOFFON(X)	; Uaktywnienie korekcji długości narzędzia online dla kierunku X narzędzia.
N30 WHEN TRUE DO $AA_TOFF[X]=10 G4 F5	; Dla kierunku X narzędzia jest interpolowana korekcja długości narzędzia wynosząca 10.
...	
N80 TOFFOF(X)	; Przesunięcie pozycji kierunku X narzędzia jest kasowany: ...$AA_TOFF[X]=0 Nie jest wykonywany ruch w żadnej osi. Do aktualnej pozycji w WKS jest doliczane przesunięcie pozycji odpowiednio do aktualnej orientacji.

Dodatkowe informacje

Przygotowanie bloków

Przy przetwarzaniu bloków w przebiegu wyprzedzającym jest uwzględniany działający w przebiegu głównym aktualne przesunięcie długości narzędzia. Aby móc w dużym stopniu korzystać z maksymalnie dopuszczalnych prędkości w osiach, jest wymagane zatrzymanie przygotowywania bloków przez zatrzymanie przebiegu wyprzedzającego `STOPRE`, podczas gdy jest budowane przesunięcie narzędzia.

Przesunięcie narzędzia jest w momencie przebiegu wyprzedzającego znany też zawsze wtedy, gdy korekcje długości narzędzia po starcie programu nie są już zmieniane, albo gdy po zmianie korekcji długości narzędzia zostało przetworzonych więcej bloków, niż może pomieścić bufor IPO między przebiegiem wyprzedzającym i przebiegiem głównym.

Zmienna $AA_TOFF_PREP_DIFF

Wielkość różnicy między aktualną korekcją działającą w interpolatorze i korekcją, która działała w chwili przetwarzania bloku, można odczytać w zmiennej $AA_TOFF_PREP_DIFF[<n>].

Ustawienie danych maszynowych i danych ustawczych

Dla korekcji długości narzędzia online są do dyspozycji następujące dane systemowe:

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (rezerwa przyśpieszenia dla ruchu nałożonego)
- MD21190 $MC_TOFF_MODE
  Zawartość zmiennej systemowej $AA_TOFF[<n>] jest realizowana albo całkowana, jako wartość absolutna.
- MD21194 $MC_TOFF_VELO (prędkość korekcji długości narzędzia online)
- MD21196 $MC_TOFF_ACCEL (przyśpieszenie korekcji długości narzędzia online)
- Dana nastawcza do zadania wartości granicznych:
  SD42970 $SC_TOFF_LIMIT (górna granica wartości korekcji długości narzędzia)

Literatura:
Podręcznik działania Funkcje specjalne (F2): Transformacje wieloosiowe

11.11 Modyfikacja danych korekcji dla obrotowych narzędzi

11.11.1 Obliczenie orientacji (ORISOLH)

Predefiniowana funkcja ORISOLH wspiera użytkownika w dostosowywaniu pozycji osi obrotowej maszyny w taki sposób, że narzędzie to zostaje ustawione w określonym, niezależnym od kinematyki położeniu względem obrabianego przedmiotu. Warunkiem jest, aktywna transformacja 6-osiowa, która jest parametryzowana za pomocą łańcuchów kinematycznych.

Dostępne są dwie funkcje podstawowe:

- Zorientowanie narzędzia
  Oba kąty β i γ są podane. Funkcja oblicza odpowiedni kąt trzech osi orientacji.
- Bezpośrednie zorientowanie narzędzia
  Kąt drugiej i trzeciej osi orientacji jest podany. Funkcja oblicza z nich odpowiednie kąty β i γ, a także brakujący jeszcze kąt pierwszej osi orientacji.

Uwaga

Kolejność osi orientacji

Jeżeli łańcuch kinematyczny, który opisuje budowę maszyny, przebiega od przedmiotu obrabianego do narzędzia, to w celu określenia kolejności trzech osi orientacji transformacji 6-osiowej mają zastosowanie następujące ustalenia:

- Oś orientacji, która leży najbliżej **przedmiotu** obrabianego jest **pierwszą** osią orientacji.
- Oś orientacji, która leży najbliżej na**rzędzia** jest **trzecią** osią orientacji.

Pierwszą osią orientacji jest z reguły wrzeciono, a odpowiedni obrót jest realizowany w tym przypadku za pomocą obrotowego Frame.

Składnia

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

Znaczenie

ORISOLH:	Wywołanie funkcji		
<RetVal>:	Wartość zwrotna funkcji		
	Typ danych:	INT	
	Zakres wartości:	0, -2, -3, ..., -17	
	Wartości:	0	Funkcja została wykonana bez błędu.
		-2	Aktualnie nie jest aktywna żadna transformacja (6-osiowa transformacja orientacji).
		-3	Parametr pierwszy (<Cntrl>) jest ujemny.
		-4	Miejsce jednostek parametru pierwszego (<Cntrl>) jest nieprawidłowe. Dozwolone są tylko wartości 0 i 1.
		-5	Miejsce dziesiątek parametru pierwszego (<Cntrl>) jest nieprawidłowe. Dozwolone są tylko wartości od 0 do 3.
		-6	Miejsce setek parametru pierwszego (<Cntrl>) jest nieprawidłowe. Dozwolone są tylko wartości 0 i 1.
		-7	Miejsce tysięcy parametru pierwszego (<Cntrl>) jest nieprawidłowe. Dozwolone są tylko wartości od 0 do 3.
		-8	W przypadku funkcji „bezpośredniego zorientowania narzędzia" kąt γ jest za duży.
		-9	W przypadku funkcji „bezpośredniego zorientowania narzędzia” naruszana jest co najmniej jedna granica określonej pozycji osi.
		-10	Żadne narzędzie nie jest aktywne.
		-11	Wymagana orientacja nie jest ustawiana.
		-12	Dopasowanie wolnych kątów osi w przypadku sprzęgła Hirtha nie jest możliwe dla pierwszego i jedynego rozwiązania.
		-13	Dopasowanie wolnych kątów osi w przypadku sprzęgła Hirtha nie jest możliwe dla drugiego rozwiązania.
		-14	Dopasowanie wolnych kątów osi w przypadku sprzęgła Hirtha nie jest możliwe dla żadnego z dwóch rozwiązań.
		-15	Pierwsza oś orientacji jest parametryzowana jako oś Hirtha.
		-16	Zarówno druga jak i trzecia oś obrotowa jest parametryzowana jako oś Hirtha. Osią Hirtha może być maksymalnie tylko jedna z dwóch osi.
		-17	W przypadku funkcji „skręt bezpośredni” co najmniej jedna z pozycji osi nie jest kompatybilna z odpowiednim sprzęgłem Hirtha.

<Cntrl>:	Steruje zachowaniem się funkcji		
	Typ danych:	INT	
	Parametr <Cntrl> jest kodowany dziesiętnie (miejsce jednostek do tysięcy):		
	Miejsce jednostek:	Miejsce jednostek steruje zachowaniem błędu.	
		xxx**0**	W przypadku błędu (wartość zwrotna < 0) wykonywanie programu jest przerywane i wyprowadzany jest alarm 14106. **Uwaga:** Alarm jest wyprowadzany niezależnie od wartości miejsca jedności również wtedy, gdy parametr <Cntrl> ma wartość ujemną.
		xxx**1**	W przypadku błędu (wartość zwrotna < 0) nie jest wyprowadzany **żaden** alarm. Użytkownik ma możliwość odpowiedniego reagowania w programie.
	Miejsce dziesiątek:	Steruje zachowaniem przy obecności osi orientacji ze sprzęgłem Hirtha. **Uwaga:** Ten parametr jest wykorzystywany tylko w przypadku funkcji „zorientuj ” (tzn. gdy miejsce setek ma wartość „0”).	
		xx**0**x	Pozycja osi jest zaokrąglana do najbliższej pozycji.
		xx**1**x	Pozycje osi są tak zaokrąglane, że odchylenie kąta β od swojej zaprogramowanej wartości jest minimalne.
		xx**2**x	Pozycje osi są tak zaokrąglane, że kąt β jest równy najwyższej możliwej wartości, która jest mniejsza niż zaprogramowana wartość (β jest zaokrąglany w dół).
		xx**3**x	Pozycje osi są tak zaokrąglane, że kąt β jest równy najmniejszej możliwej wartości, która jest większa niż zaprogramowana wartość (β jest zaokrąglany w górę).
	Miejsce setek:	Określa, jakie funkcje mają być wykonywane lub jakie znaczenie mają dwa następujące parametry <W1> i <W2>.	
		x**0**xx	Funkcja „zorientowania narzędzia” Parametry <W1> i <W2> mają następujące znaczenie: • <W1> = β • <W2> = γ Obliczane będą odpowiednie kąty osi orientacji.
		x**1**xx	Funkcja „bezpośrednie zorientowanie narzędzia” <W1> jest zadaną pozycją dla drugiej osi orientacji, <W2> jest zadaną pozycją dla trzeciej osi orientacji transformacji 6-osiowej. Określane są pozycje pierwszej osi orientacji i kąty β i γ, które są kompatybilne z obiema zadanymi pozycjami. Jeżeli nie wystąpił żaden błąd, to w zmiennych systemowych $P_ORI_POS[<n>, <m>] będą zawsze wyprowadzane dwa rozwiązania. Przy czym indeks pierwszy <n> (0 lub 1) odsyła do rozwiązania, natomiast indeks drugi <m> (0 ... 2) do osi orientacji: • $P_ORI_POS[0/1, **0**]: Pozycja pierwszej osi orientacji • $P_ORI_POS[0/1, **1**]: Kąt β • $P_ORI_POS[0/1, **2**]: Kąt γ Sprawdzane jest, czy zadane pozycje <W1> i <W2> są kompatybilne z aktywnym sprzęgłem Hirtha lub aktywnym ogra-

			niczeniem softwareowym. Jeśli tak nie jest, zwracany jest odpowiedni numer błędu (patrz parametr <RetVal>). Przy dowolnym wyborze obu kątów <W1> i <W2> ostrze narzędzia zasadniczo nie leży w płaszczyźnie obróbki. Kąt γ, o który obracane jest ostrze z płaszczyzny obróbki, nie może być większy niż wartość graniczna, która jest określana przez daną ustawczą SD42999 $SC_ORISOLH_INCLINE_TOL.
	Miejsce tysięcy:	Określa, jak należy właściwie zmodyfikować pozycje rozwiązania, gdy miejsce setek ma wartość „0”, tzn. w przypadku funkcji „orientowania narzędzia”.	
		0xxx	Obliczone pozycje osi powinny znajdować się możliwie blisko aktualnych pozycji osi maszynowych.
		1xxx	Obliczone pozycje osi w przypadku osi modulo powinny znajdować się możliwie blisko środka zakresu modulo, a w przypadku innych osi możliwie blisko 0. Dla osi nie będących osiami modulo oznacza to, że pozycje osi będą redukowane w zakresie -180° … +180°.
		2xxx	Obliczone pozycje osi niezależnie od typu osi powinny być redukowane w zakresie -180° … +180°.
<W1>:	Kąt pierwszy Znaczenie wynika z miejsca setek parametru <Cntrl>.		
	Typ danych:	REAL	
<W1>:	Kąt drugi Znaczenie wynika z miejsca setek parametru <Cntrl>.		
	Typ danych:	REAL	

Uwaga

Parametry niezaprogramowane mają wartość standardową "0".

Dalsze informacje

Liczba znalezionych rozwiązań wraz z kolejnymi informacjami o statusie przy wykonywaniu funkcji ORISOLH mogą być czytane przez następujące zmienne systemowe:

Zmienna systemowa	Znaczenie		
$P_ORI_POS [<n>, <m>]	Podaje kąt osi orientacji, który określany jest podczas programowania orientacji.		
	<n>:	Indeks rozwiązania	
		Zakres wartości:	0, 1
	<m>:	Indeks osi orientacji	
		Zakres wartości:	0 ... 2
			Kolejność osi orientacji (1 ... 3) odnosi się do definicji osi w $NT_ROT_AX_NAME.
	Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia”, zmienne $P_ORI_POS[0/1, 1] i P_ORI_POS[0/1, 2] zawierają wartości obu kątów β i γ należących do obu rozwiązań. Pierwsze rozwiązanie wpisane do $P_ORI_POS[<n>, <m>], tzn. z indeksem <n> = 0, jest zawsze tym, które jest wybierane przez sterowanie, gdy żądana orientacja zostanie najechana. Drugi indeks <m> odnosi się przy tym do osi orientacji, tzn. do $NT_ROT_AX_NAME. Pozycje osi wpisane do $P_ORI_POS[<n>, <m>] uwzględniają offsety, które są wprowadzone do $NK_OFF i $NK_OFF_FINE, tzn. ten kąt osi w kolejnych blokach może być zastosowany bez dalszych modyfikacji w celu ustawienia żądanej orientacji. Jeżeli oś obrotowa jest osią Hirtha, to pozycje rozwiązania są zaokrąglane do najbliższego skoku sprzędła Hirtha. W zmiennej systemowej $P_ORI_DIFF mogą być odczytywane różnice w pozycjach osi dla dokładnych rozwiązań oraz rozwiązań dopasowanych do podziału rastra Hirtha dla osi obrotowych sprzęgła Hitha.		
$P_ORI_DIFF [<n>, <m>]	Podaje różnicę pomiędzy pozycjami dokładnymi i pozycjami osi orientacji dostępnymi w $P_ORI_POS, które określane są podczas programowania orientacji.		
	<n>:	Indeks rozwiązania	
		Zakres wartości:	0, 1
	<m>:	Indeks osi orientacji	
		Zakres wartości:	0 ... 2
			Kolejność osi orientacji (1 ... 3) odnosi się do definicji osi w $NT_ROT_AX_NAME.
	Treść może być niezerowa tylko wtedy, gdy pozycje są z rastra (sprzęgło Hirtha), tzn. jeśli dana systemowa $ NT_HIRTH_INCR danej osi jest niezerowa i jeśli ta oś jest ręczną osią obrotową.		

Zmienna systemowa	Znaczenie		
$P_ORI_SOL	Jeżeli kąt osi podczas transformacji orientacji obliczany jest z więcej niż jedną osią orientacji, która prowadzić ma do określonej orientacji, istnieje zasadniczo więcej niż jedno rozwiązanie. Zmienna systemowa $P_ORI_SOL zawiera liczbę poprawnych rozwiązań wraz z dodatkowymi informacjami o statusie. Treść $P_ORI_SOL jest kodowana w następujący sposób:		
	Wartości < 0	Ogólne stany błędów	
		-1	Dla aktywnej transformacji nie zostały jeszcze obliczone żadne rozwiązania (brak wywołania ORISOLH).
		-2	Żadna transformacja nie jest aktywna lub aktywną transformacją nie jest żadna transformacja orientacji (transformacja 6-osiowa), która może dostarczać pozycje do określonego programowania orientacji.
		-4	Żądana orientacja z daną kinematyką nie jest ustawiana.
		-5	Przy wywołaniu funkcji ORISOLH w trybie bezpośredniego zorientowania narzędzia: nie znaleziono żadnego rozwiązania.
		-6	Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia" kąt γ jest za duży.
		-7	Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia”, podany został kąt, który z uwagi na sprzęgło Hitha nie jest ustawiany.
		-8	Pierwsza oś orientacji (oś Frame) nie może być parametryzowana jako oś Hirtha.
		-9	Zarówno druga jak i trzecia oś obrotowa jest parametryzowana jako oś Hirtha. Osią Hirtha może być maksymalnie tylko jedna z dwóch osi.
		-10	Nie znaleziono żadnego dopasowania rozwiązania/rozwiązań do sprzęgła Hirtha.
	Wartości > 0 Miejsce jednostek:	Liczba możliwych rozwiązań matematycznych bez uwzględnienia granic osi lub ewentualnie warunków błędów.	
		0	Nie występuje żadne rozwiązanie, tzn. wymagana orientacja nie jest ustawiana. Taki przypadek może mieć trzy różne przyczyny: • Ze względu na kinematykę maszyny (osie orientacji nie są ustawione pod kątem prostym), wymagana orientacja jest zasadniczo nieosiągalna, nawet przy dowolnym zakresie ruchu osi orientacji. W tym przypadku miejsca dziesiątek i setek $P_ORI_SOL mają wartość zero, przyporządkowana zmienna statusu $P_ORI_STAT do osi orientacji zawiera wartość „-4”. • Obliczone rozwiązania nie są osiągalne, ponieważ naruszają one ograniczenia osi. Pozycje osi orientacji, wynikające bez ograniczeń osi, mogłyby być odczytywane w $P_ORI_POS. • Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia”, pozycje osi zostały podane w taki sposób, że albo wektor orientacji albo wektor normalnej orientacji narzędzia zorientowany

Zmienna systemowa	Znaczenie		
			jest równolegle do pierwszej osi, której pozycja ma być obliczona. Pozycja tej osi w takich przypadkach nie jest definiowana.
		1	Istnieje jedno rozwiązanie. Taki przypadek może mieć trzy różne przyczyny: • Z uwagi na ustaloną orientację i kinematykę maszyny również bez uwzględnienia granic osi istnieje tylko jedno rozwiązanie (pod względem matematycznym chodzi o dwa zbieżne rozwiązania). Ten przypadek występuje w przypadku kinematyki nieprostokątnej na krawędzi zakresu orientacji. $P_ORI_POS zawiera oba (identyczne) rozwiązania. • Jest tylko jedno rozwiązanie, ponieważ drugie rozwiązanie jest niepoprawne z uwagi na naruszenie granic osi. Poprawnym rozwiązaniem jest wtedy zawsze pierwsze rozwiązanie w $P_ORI_POS. Drugie rozwiązanie, wynikające bez uwzględnienia granic osi, mogłyby być również odczytywane w $P_ORI_POS. • Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia" jest to normalny przypadek. Do określonych pozycji drugiej osi orientacji istnieje zasadniczo tylko jedna poprawna pozycja brakującej do obliczenia osi orientacji.
		2	Istnieją dwa rozwiązania.
		8	Istnieje nieskończenie wiele rozwiązań tzn. pozycja osi orientacji (osi biegunowej) jest dowolna. Z dwóch możliwych położeń drugiej osi wyklucza się jednak ograniczenia związane z naruszeniem granic osi.
		9	Istnieje nieskończenie wiele rozwiązań tzn. pozycja osi orientacji (osi biegunowej) jest nieokreślona. Nieokreślone osie mogą być ustalane na podstawie miejsca setek lub zmiennej systemowej $P_ORI_STAT.
	Wartości > 0 Miejsce dziesiątek:	Wyświetlanie kodowane bitowo dla naruszonych granic osi. Dokładne przyczyny błędów mogą być określane na podstawie zmiennej systemowej $P_ORI_STAT.	
		Bit 0 (wartość 10):	Dla przynajmniej jednego rozwiązania jest naruszona co najmniej jedna granica 1. osi orientacji.
		Bit 1 (wartość 20):	Dla przynajmniej jednego rozwiązania jest naruszona co najmniej jedna granica 2. osi orientacji.
		Bit 2 (wartość 40):	Dla przynajmniej jednego rozwiązania jest naruszona co najmniej jedna granica 3. osi orientacji.
	Wartości > 0 Miejsce setek:	Wyświetlenie kodowane bitowo dla niezdefiniowanych pozycji osi (może wystąpić tylko wtedy, gdy istnieje nieskończenie wiele rozwiązań, tzn. gdy miejsce jednostek jest równe „9”).	
		Bit 0 (wartość 100):	Pozycja 1. osi orientacji nie jest zdefiniowana.
		Bit 1 (wartość 200):	Pozycja 2. osi orientacji nie jest zdefiniowana.

Zmienna systemowa	Znaczenie		
		Bit 2 (wartość 400):	Pozycja 3. osi orientacji nie jest zdefiniowana.
	Oznaczenie 1., 2. i 3. osi orientacji odnosi się do definicji osi w $NT_ROT_AX_NAME.		

Zmienna systemowa	Znaczenie			
$P_ORI_STAT [<n>]	Podaje status dla każdej z maksymalnie 3 osi orientacji po wywołaniu OR.			
	<n>:	Indeks osi orientacji (odpowiada indeksowi określonej osi orientacji w $NT_ROT_AX_NAME)		
		Zakres wartości:	0 ... 2	
			Kolejność osi orientacji (1 ... 3) odnosi się do definicji osi w $NT_ROT_AX_NAME.	
	Treść $P_ORI_STAT jest kodowana w następujący sposób:			
	Wartości < 0	Ogólne stany błędów		
		-1	Status nie jest zdefiniowany (brakujące wywołanie ORISOLH).	
		-2	Żadna transformacja nie jest aktywna lub aktywną transformacją nie jest żadna transformacja orientacji (transformacja 6-osiowa), która może dostarczać pozycje do określonego programowania orientacji.	
		-3	Oś nie znajduje się w aktywnej transformacji.	
		-4	Pozycja osi nie może zostać obliczona, ponieważ przy danej kinematyce nie można osiągnąć wymaganej orientacji, nawet przy dowolnym przyjętym zakresie ruchu osi.	
		-5	Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia”, pozycje osi zostały podane w taki sposób, że albo wektor orientacji albo wektor normalnej orientacji narzędzia zorientowany jest równolegle do pierwszej osi, której pozycja ma być obliczona. Pozycja tej osi w takich przypadkach nie jest definiowana.	
		-6	Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia" kąt γ jest za duży.	
		-7	Przy wywołaniu funkcji ORISOLH w trybie „bezpośredniego zorientowania narzędzia”, podany został kąt, który z uwagi na sprzęgło Hitha nie jest ustawiany.	
		-8	Pierwsza oś orientacji (oś Frame) nie może być parametryzowana jako oś Hirtha.	
		-9	Zarówno druga jak i trzecia oś obrotowa jest parametryzowana jako oś Hirtha. Osią Hirtha może być maksymalnie tylko jedna z dwóch osi.	
		-10	Nie znaleziono żadnego dopasowania rozwiązania/rozwiązań do sprzęgła Hirtha.	
	Wartości > 0 Miejsce jednostek:	Wyświetlanie kodowane bitowo dla naruszonych granic osi pierwszego rozwiązania.		
		Bit 0 (wartość 1):	Pierwsze rozwiązanie narusza dolną granicę osi.	
		Bit 1 (wartość 2):	Pierwsze rozwiązanie narusza górną granicę osi.	
	Wartości > 0 Miejsce dziesiątek:	Wyświetlanie kodowane bitowo dla naruszonych granic osi drugiego rozwiązania.		
		Bit 0 (wartość 10):	Drugie rozwiązanie narusza dolną granicę osi.	
		Bit 1 (wartość 20):	Drugie rozwiązanie narusza górną granicę osi.	
	Wartości > 0 Miejsce setek:	Wyświetlenie niezdefiniowanych pozycji osi.		
		Bit 0 (wartość 100):	Pozycja osi orientacji nie jest zdefiniowana, tzn. żądana orientacja jest osiągana w każdym dowolnym ustawieniu osi obrotowej (pozycja biegu-	

Zmienna systemowa	Znaczenie		
			na). Ta informacja jest również zawarta w zmiennej systemowej $P_ORI_SOL.
	Spośród numerów błędów, które wskazują na naruszenie granic osi, kilka może wystąpić jednocześnie. Ponieważ przy naruszeniu granic osi próbuje się osiągnąć pozycję w dopuszczalnych granicach osi poprzez dodanie lub odjęcie wielokrotności 360 ° - jeżeli nie jest to możliwe - nie jest jasno określone, czy naruszona została dolna czy granica osi. Jeśli nie ma rozwiązania dla wymaganej orientacji ($ P_ORI_SOL = 0), status osi orientacji zawartych w transformacji wynosi "0".		

Uwaga

$NT_ROT_AX_NAME

Ta zmienna systemowa odnosi się do maksymalnie 3 osi, które służą do ustawienia orientacji. Zawiera ona nazwy ogniw łańcucha ($ NK_NAME), które definiują osie maszyny (osie obrotowe), które muszą wykonywać ruchy orientacji wynikające z transformacji kinematycznej. Kolejność, w której maksymalnie trzy osie obrotowe są zawarte w tej zmiennej systemowej jest bez znaczenia dla kinematyki maszyny, ponieważ pochodzi ona ze struktury łańcuchów kinematycznych. Ponieważ jednak definiuje ona kolejność, w jakiej inne zmienne uzyskują dostęp do osi obrotowych, kolejność osi orientacji w NT_ROT_AX_NAME musi być zgodna z opisem kinematyki.

Uwaga

Informacje o statusie

Informacje o statusie, które na przykład wskazują na to, że orientacja jest nieosiągalna lub może zostać osiągnięta tylko z naruszeniem odpowiednich granic osi, nie prowadzą do alarmu NC. Odpowiedzialność za reagowanie na wymienione warunki spoczywa na użytkowniku.

11.11.2 Aktywacja modyfikacji danych korekcji dla narzędzi obrotowych (CUTMOD, CUTMODK)

Modyfikacja danych korekcji dla narzędzi obrotowych jest aktywowana w programie NC za pomocą polecenia językowego CUTMOD (w połączeniu z orientowanymi nośnikami narzędzi) lub CUTMODK (dla transformacji orientacji zdefiniowanej przy użyciu łańcuchów kinematycznych).

Uwaga

Ponieważ orientowalne nośniki narzędzi i transformacje orientacji zdefiniowane poprzez łańcuchy kinematyczne nie mogą być aktywne w tym samym czasie, nie powstają żadne konflikty między tymi dwoma wariantami.

Składnia

```
CUTMOD = <Value>
```

lub

```
CUTMODK = <Command>
```

Znaczenie

CUTMOD:	Wywołanie funkcji w połączeniu z orientowanymi nośnikami narzędzi		
<Value>:	Przypisana wartość		
	Typ danych:	INT	
	Wartość:	0	Funkcja nie jest aktywna. Wartości dostarczane przez zmienne systemowe $ P_AD ... są takie same jak odpowiadające im parametry narzędzia.
		> 0	Funkcja jest aktywowana, jeśli aktywny jest nośnik narzędzia z określonym numerem, tzn. aktywacja jest powiązana z konkretnym orientowanym nośnikiem narzędzia. Wartości dostarczone przez zmienne systemowe $ P_AD ... mogły zostać zmodyfikowane względem odpowiednich parametrów narzędzia w zależności od aktywnego obrotu. Wyłączenie aktywności określonego orientowanego nośnika narzędzi tymczasowo dezaktywuje funkcję, aktywacja innego orientowanego nośnika narzędzi trwale wyłącza jej aktywność. W pierwszym przypadku funkcja zostaje ponownie aktywowana, gdy ponownie wybierany jest ten sam orientowany nośnik narzędzi, w drugim przypadku konieczne jest dokonanie nowego wyboru, nawet jeśli orientowany nośnik narzędzi z określoną liczbą zostanie ponownie uaktywniony w późniejszym czasie. Reset nie wpływa na funkcję.
		-1	Funkcja jest aktywowana zawsze wtedy, gdy orientowany nośnik narzędzi jest aktywny. Przy zmianie nośnika narzędzi lub przy jego odwołaniu i późniejszym ponownym wybraniu, CUTMOD nie musi być ponownie ustawiany.
		-2	Funkcja jest aktywowana zawsze wtedy, gdy aktywowany jest orientowany nośnik narzędzi, którego numer jest taki sam jak numer aktualnie aktywnego orientowanego nośnika narzędzi. Gdy żaden z orientowanych nośników narzędzi nie jest aktywny, oznacza to, że CUTMOD=0. Gdy orientowany nośnik narzędzi jest aktywny, jest to jednoznaczne z bezpośrednim podaniem aktualnego numeru nośnika narzędzi.
		< -2	Wartości mniejsze od 2 są ignorowane, tzn. ten przypadek jest traktowany tak, jakby CUTMOD nie został zaprogramowany. **Uwaga:** Ten zakres wartości nie powinien być stosowany, ponieważ jest on zarezerwowany dla ewentualnych późniejszych rozszerzeń.
CUTMODK:	Wywołanie funkcji związane z transformacjami orientacji zdefiniowanymi przez łańcuchy kinematyczne		

<Command>:	Przypisany rozkaz		
	Typ danych:	STRING	
	Wartość:	"NEW"	Zapisywany są stany istotne dla "modyfikacji danych korekcji" aktywnej transformacji zdefiniowanej przy pomocy łańcuchów kinematycznych, nazwy transformacji i aktualnego Frame konturu. **Uwaga:** Ten rozkaz jest dozwolony tylko wtedy, gdy aktywna jest odpowiednia transformacja (TRAORI_DYN, TRAORI_STAT lub TRAANG_K).
		"OFF"	Wyłącza aktywną „modyfikację danych korekcji”. Dane zapisane wcześniej z "NEW" są zachowane. **Uwaga:** Ten rozkaz jest również dozwolony, gdy CUTMODK nie jest aktywny. Nie ma to żadnego wpływu. Istniejący zestaw danych dla "modyfikacji danych korekcji" zostaje zachowany.
		"ON"	Za pomocą tego rozkazu, "modyfikacja danych korekcji" jest ponownie aktywowana poprzez zestaw danych zapisany wcześniej za pomocą rozkazu "NEW". Jeśli transformacja z nazwą zapisanego zestawu danych jest aktywna podczas wykonywania tego rozkazu, "modyfikacja danych korekcji" zaczyna działać natychmiast. W przeciwnym razie aktywacja zostanie opóźniona do momentu aktywowania aktywnej transformacji.
		"CLEAR"	Wyłącza "modyfikację danych korekcji", podobnie jak rozkaz „OFF", a także dodatkowo kasuje zapisany zestaw danych. **Uwaga:** Ten rozkaz jest również dozwolony, gdy CUTMODK nie jest aktywny.

Uwaga

SD42984 $SC_CUTDIRMOD

Funkcja aktywowana przez polecenie CUTMOD lub CUTMODK zastępuje funkcję aktywowaną przez daną ustawczą SD42984 $SC_CUTDIRMOD. Jednak ta funkcja jest nadal dostępna. Ponieważ nie ma sensu używać równolegle obu funkcji, można je aktywować tylko wtedy, gdy CUTMOD jest równa zero, a CUTMODK jest równa łańcuchowi zerowemu.

Dalsze informacje

Czytanie zmodyfikowanych danych korekcji

Zmodyfikowane dane korekcji są dostarczane w następujących zmiennych systemowych i zmiennych BTSS:

Znaczenie	Zmienna systemowa	Zmienne BTSS
Położenie ostrza	$P_AD[2]	cuttEdgeParam2
Kąt uchwytu	$P_AD[10]	cuttEdgeParam10

Znaczenie	Zmienna systemowa	Zmienne BTSS
Kierunek skrawania	$P_AD[11]	cuttEdgeParam11
Kąt przyłożenia	$P_AD[24]	cuttEdgeParam24

Dane są zawsze modyfikowane w porównaniu do odpowiednich parametrów narzędzia ($ TC_DP2 [..., ...], ...), jeśli funkcja „modyfikacji danych korekcji dla narzędzi obrotowych" została aktywowana za pomocą polecenia `CUTMOD` lub `CUTMOD`, a narzędzie zostało obrócone przez orientowany nośnik narzędzi lub odpowiednią transformację orientacji.

Inne funkcje ważne pod względem funkcjonalnym

Zmienna systemowa	Znaczenie
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	Zwraca kąt, o który narzędzie zostało obrócone w aktywnej płaszczyźnie obróbki i opiera się na określeniu zmodyfikowanych danych ostrzy w funkcjach CUTMOD lub CUTMODK.
$P_CUTMOD / $AC_CUTMOD	Odczytuje aktualną wartość, która została ostatnio zaprogramowana za pomocą polecenia CUTMOD (numer nośnika narzędzi, dla którego ma zostać aktywowana modyfikacja danych korekcji). Jeśli ostatnią zaprogramowaną wartością była `CUTMOD = -2` (aktywacja z aktualnie aktywnym orientowanym nośnikiem narzędzi), zmienna systemowa nie zwraca wartości "-2", ale numer orientowanego nośnika narzędzia aktywnego w czasie programowania.
$P_CUTMODK / $AC_CUTMODK	Odczytuje nazwę transformacji, pod którą utworzono aktualny zestaw danych dla "modyfikacji danych korekcji".
$P_CUT_INV / $AC_CUT_INV	Daje wartość TRUE, gdy narzędzie jest tak obracane, że kierunek obrotów wrzeciona musi zostać odwrócony. W tym celu w bloku, do którego odnosi się dana operacja odczytu, muszą być spełnione następujące cztery warunki: 1. Jest aktywne narzędzie tokarskie albo szlifierskie (typy narzędzi 400 do 599 i / albo SD42950 $SC_TOOL_LENGTH_TYPE = 2). 2. Modyfikacja danych korekcji została aktywowana za pomocą polecenia `CUTMOD` lub `CUTMODK`. 3. Jest aktywny orientowany nośnik narzędzi lub transformacja orientacji zdefiniowana za pomocą łańcuchów kinematycznych, który/która została wybrana za pomocą polecenia CUTMOD lub CUTMODK. 4. Narzędzie jest obracane przez orientowany nośnik narzędzi lub przez kinematyczną transformację orientacji, tak że wynikowa normalna ostrza narzędzia jest obracana o więcej niż 90° (zazwyczaj 180°) względem położenia początkowego. Jeśli co najmniej jeden z czterech powyższych warunków nie jest spełniony, zmienna zwraca wartość FALSE. Dla narzędzi, których położenie ostrza nie jest zdefiniowane, wartość zmiennej jest zawsze FALSE.

Zmienna systemowa	Znaczenie	
$P_CUTMOD_ERR	Stan błędu po ostatnim wywołaniu funkcji CUTMOD Funkcja CUTMOD może być również bezpośrednio wywoływana podczas zmiany narzędzi. Po resecie zmienna jest ustawiana na wartość zero. Jest ona najpierw cofana przy każdej zmianie narzędzia i w stosownych przypadkach ponownie opisywana. Zmienna jest kodowana w sposób bitowy. Bity mają zastępujące znaczenie:	
	Bit 0:	Dla aktywnego narzędzia nie zdefiniowano prawidłowego kierunku skrawania.
	Bit 1:	Kąty skrawania (kąt przyłożenia i kąt uchwytu) aktywnego narzędzia są równe zeru.
	Bit 2:	Kąt przyłożenia aktywnego narzędzia ma nieprawidłową wartość (<0 ° lub > 180 °).
	Bit 3:	Kąt uchwytu aktywnego narzędzia ma nieprawidłową wartość (<0 ° lub > 90 °).
	Bit 4:	Kąt płytki aktywnego narzędzia ma nieprawidłową wartość (<0 ° lub > 90 °).
	Bit 5:	Kombinacja położenia ostrza - kąt uchwytu aktywnego narzędzia jest niedopuszczalny (dla położenia ostrzy od 1 do 4 kąt uchwytu musi wynosić ≤ 90 °, dla pozycji ostrza od 5 do 8 musi to być ≥ 90 °).
	Bit 6:	Niedopuszczalny obrót aktywnego narzędzia. Narzędzie zostało obrócone o ± 90 ° (z tolerancją około 1 °) poza aktywną płaszczyznę obróbki. W rezultacie położenie ostrza na płaszczyźnie obróbki nie jest już zdefiniowane.
	Bit 7:	Płytka skrawająca nie znajduje się w płaszczyźnie obróbki, a kąt między płytką skrawającą a płaszczyzną obróbki przekracza górną granicę określoną przez daną nastawczą SD42998 $ SC_CUTMOD_PLANE_TOL.
	Bit 8:	Płytka skrawająca nie znajduje się w płaszczyźnie obróbki. Wielkość kąta α jest większa niż 1 °. Kąt α to kąt obrotu wokół osi współrzędnych, który jest prostopadły zarówno do osi obrotu kąta β, jak i do osi obrotu kąta γ (dla G18 w osi X).

$P_ ...: Zmienne przebiegu wyprzedzającego

$AC_...: Zmienne przebiegu głównego

Wszystkie zmienne przebiegu głównego można odczytać w akcjach synchronicznych. Dostęp do odczytu z przebiegu wyprzedzającego generuje zatrzymanie przebiegu wyprzedzającego.

Zmiana płaszczyzny

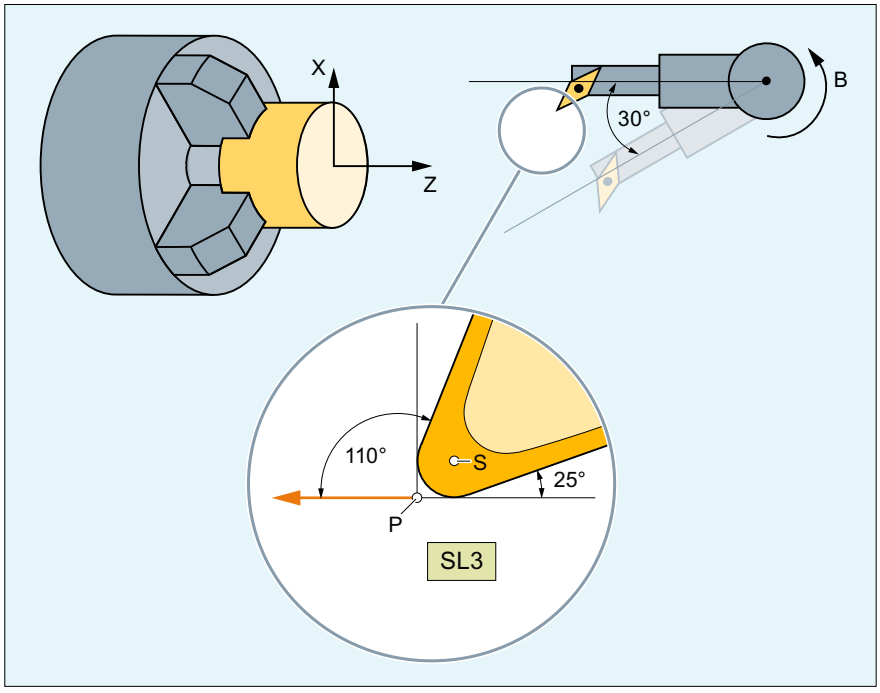
W celu określenia zmodyfikowanego położenia ostrza, kierunku skrawania, kąta uchwytu lub kąta przyłożenia decydujące znaczenie ma uwzględnienie ostrza w odpowiedniej aktywnej płaszczyźnie (G17 - G19).

Jeżeli jednak dana ustawcza SD42940 $SC_TOOL_LENGTH_CONST (zmiana składowych długości narzędzia przy zmianie płaszczyzny) zawiera poprawną wartość zerową (plus lub minus 17, 18 lub 19), wówczas jej treść określa płaszczyznę, w której rozpatrywane są odpowiednie wielkości.

Ta nadrzędna reguła danych ustawczych za pomocą kodu G może zostać uchylona przez ustawienie Bitu nr 18 danej maszynowej MC_TOOL_PARAMETER_DEF_MASK. Oznacza to, że po ustawieniu tego bitu zawsze obowiązuje płaszczyzna zdefiniowana przez polecenie grupy G nr 6.

Działanie zmodyfikowanych danych ostrza

Zmodyfikowane położenie ostrza i zmodyfikowany punkt odniesienia ostrza są natychmiast skuteczne przy zaprogramowaniu nawet dla już aktywnego narzędzia. Wybór nowego narzędzia nie jest konieczny.



S: Punkt środkowy ostrza

P: Punkt odniesienia ostrza

SL: Położenie ostrza

Rysunek 11-4 Narzędzie z położeniem ostrza 3 i orientowanym nośnikiem narzędzi, który może obracać narzędziem wokół osi B

Kod programu	**Komentarz**
N10 $TC_DP1[1,1]=500	
N20 $TC_DP2[1,1]=3	; Położenie ostrza
N30 $TC_DP3[1,1]=12	
N40 $TC_DP4[1,1]=1	
N50 $TC_DP6[1,1]=6	
N60 $TC_DP10[1,1]=110	; Kąt uchwytu
N70 $TC_DP11[1,1]=3	; Kierunek skrawania
N80 $TC_DP24[1,1]=25	; Kąt przyłożenia
N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0	; Oś B
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1	; Oś C
N110 $TC_CARR13[2]=0	
N120 $TC_CARR14[2]=0	
N130 $TC_CARR21[2]=X	

Kod programu	Komentarz
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	

Wartości liczbowe w komentarzach wskazują końcowe pozycje bloków we współrzędnych maszyny (MKS) w kolejności X → Y → Z.

Objaśnienia

W bloku N180 narzędzie wybiera się najpierw za pomocą CUTMOD=0 i przy nieobróconym orientowanym nośniku narzędzi. Ponieważ wszystkie wektory przesunięcia orientowanego nośnika narzędzi wynoszą 0, następuje ruch do pozycji, która odpowiada długościom narzędzi określonych w $TC_DP3[1,1] i $TC_DP4[1,1].

W bloku N200 orientowany nośnik narzędzi aktywowany jest z obrotem 30° wokół osi B. Ponieważ położenie ostrza nie jest zmodyfikowane z powodu CUTMOD=0, nadal obowiązuje stary punkt odniesienia ostrza. Dlatego w bloku N210 następuje ruch do pozycji, która zachowuje stary punkt odniesienia ostrza w punkcie zerowym (tzn. wektor (1, 12) obraca się w płaszczyźnie Z/X o 30°).

W bloku N260 w przeciwieństwie do bloku N200 działa CUTMOD=2. Ze względu na obrót orientowanego nośnika narzędzi, zmodyfikowanym położeniem ostrza staje się 8. Powoduje to również różne pozycje osi.

W blokach N220 lub N270 aktywowana jest korekcja promienia narzędzia (WRK). Różne położenie ostrza w obu fragmentach programu nie ma wpływu na pozycje końcowe bloków, w których jest aktywna WRK, w związku z czym odpowiednie pozycje są identyczne. Dopiero w blokach odwołania N260 lub N300 znów oddziałują różne położenia ostrza.

11.12 Praca w środowiskach narzędziowych

Przegląd funkcji

- Zapisanie środowiska narzędziowego (TOOLENV) (Strona 494)
- Kasowanie środowiska narzędziowego (DELTOOLENV) (Strona 497)
- Odczyt numerów T, D i DL (GETTENV) (Strona 498)
- Odczyt długości narzędzia lub składowych długości narzędzia (GETTCOR) (Strona 499)
- Zmiana składowych narzędzia (SETTCOR) (Strona 506)

Przegląd zmiennych systemowych

- Odczyt informacji do zapisanych środowisk narzędziowych ($P_TOOLENVN, ($P_TOOLENV) (Strona 499)

11.12.1 Zapisanie środowiska narzędziowego (TOOLENV)

Funkcja TOOLENV służy do zapisywania wszystkich aktualnych stanów, które mają znaczenie dla interpretacji danych narzędzia zapisanych w pamięci.

Są to następujące dane:

- Aktywne polecenia grupy G:
  - nr 6 (G17, G18, G19)
  - nr 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- Aktywna oś poprzeczna
- Dane maszynowe:
  - MD18112 $MN_MM_KIND_OF_SUMCORR (właściwości korekcji sumarycznej w obszarze TO)
  - MD20360 $MC_TOOL_PARAMETER_DEF_MASK (definicja parametrów narzędzia)

- Dane ustawcze:
  - SD42900 $SC_MIRROR_TOOL_LENGTH (zmiana znaku długości narzędzia przy odbiciu lustrzanym)
  - SD42910 $SC_MIRROR_TOOL_WEAR (zmiana znaku zużycia narzędzia przy lustrzanym odbiciu)
  - SD42920 $SC_WEAR_SIGN_CUTPOS (znak zużycia dla narzędzi z położeniami ostrzy)
  - SD42930 $SC_WEAR_SIGN (znak zużycia)
  - SD42935 $SC_WEAR_TRANSFORM (transformacje dla składowych narzędzia)
  - SD42940 $SC_LENGTH_CONST (zmiana składowych długości narzędzia przy zmianie płaszczyzny)
  - SD42942 $SC_TOOL_LENGTH_CONST_T (zmiana składowych długości narzędzia dla narzędzi tokarskich przy zmianie płaszczyzny)
  - SD42950 $SC_TOOL_LENGTH_TYPE (przyporządkowanie składowych długości narzędzia niezależnie od typu narzędzia)
  - SD42954 $SC_TOOL_ORI_CONST_M (zmiana składowych orientacji narzędzia dla narzędzi frezarskich przy zmianie płaszczyzny)
  - SD42956 $SC_TOOL_ORI_CONST_T (zmiana składowych orientacji narzędzia dla narzędzi tokarskich przy zmianie płaszczyzny)
- Składowa orientacji aktualnego frame łącznego (obroty i lustrzane odbicia, bez przesunięć punktu zerowego lub skalowań)
- Składowa orientacji i wynikowe długości aktywnych orientowanych nośników narzędzi
- Składowa orientacji i wynikowe długości aktywnych transformacji

Poza wymienionymi danymi, które opisują środowisko narzędzia, będzie zapisywany również numer T, D i DL aktywnego narzędzia, aby możliwy był późniejszy dostęp do tego narzędzia w tym samym środowisku jak przy wywołaniu TOOLENV bez konieczności ponownego określania narzędzia.

Składnia

```
<Status> = TOOLENV(<Name>)
```

Znaczenie

`TOOLENV(...)`:	Wstępnie zdefiniowana funkcja do zapisania środowiska narzędziowego	
	W oddzielnym bloku:	tak

<Status>:	Wartość zwrotna funkcji. Wartości ujemne wskazują stany błędu.		
	Typ danych:		INT
	Wartość:	0	Funkcja OK
		-1	Brak zarezerwowanego miejsca w pamięci dla środowisk narzędziowych: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkcjonalność „Środowiska narzędziowe” nie jest dostępna.
		-2	Brak wolnego miejsca w pamięci dla środowisk narzędziowych.
		-3	Zerowy łańcuch znaków, jako nazwa środowiska narzędzia nie jest dopuszczalna.
		-4	Brak podanego parametru (<nazwa>)
Parametry			
1 <nazwa>:	Nazwa, pod którą powinien być zapisany aktualny zestaw danych. Jeżeli zestaw danych o tej samej nazwie już istnieje jest on nadpisywany. W tym przypadku status wynosi „0”.		
	Typ danych:		STRING

Dodatkowe informacje

Wymiar bazowy/Wymiar przystawki - korekcja długości narzędzia

To czy długość przystawki czy wymiar bazowy są użyte do obliczenia długości narzędzia (parametry specyficzne dla ostrzy $TC_DP21, $TC_DP22 i $TC_DP23) decyduje wartość w danej maszynowej aktywnego zarządzania magazynem narzędzi (dostępna tylko z opcją „Zarządzanie narzędziami”!).

MD18104 $MN_MM_NUM_TOOL_ADAPTER (adapter narzędzia w obszarze TO)

Ponieważ zmiana tej danej maszynowej może być skuteczna tylko przy rozruchu sterowania, nie jest ona zapisywana w środowisku narzędziowym.

Wynikowa długość orientowanego nośnika narzędzi i transformacji

Uwaga

Zarówno przy orientowanych nośnikach narzędzi jak również przy transformacjach istnieją zmienne systemowe lub dane maszynowe, które działają jak dodatkowe składowe długości narzędzia i są całkowicie lub częściowo zależne od wykonywanego obrotu. Wynikające poprzez to dodatkowe składowe długości narzędzia, muszą być przy wywołaniu TOOLENV razem zapisane, ponieważ tworzą one część środowiska, w którym jest stosowane narzędzie.

Transformacja przystawki

Transformacja przystawki jest właściwością przystawki narzędzia i tym samym narzędzia całkowitego. Dlatego nie jest składnikiem środowiska narzędziowego, które może być zastosowane dla innego narzędzia.

Przez zapisanie wszystkich danych koniecznych do określenia całkowitej długości narzędzia jest możliwe obliczenie w późniejszym czasie rzeczywistej długości narzędzia, również wtedy, gdy w tym czasie nie jest ono aktywne lub zmieniły się warunki środowiska (np. polecenia G

lub dane ustawcze). Tak samo można obliczyć rzeczywistą długość innego narzędzia przyjmując, że zostało ono utworzone na tych samych warunkach co narzędzie, dla którego został zapisany status.

Maksymalna liczba zestawów danych dla środowisk narzędziowych

Za pomocą danej maszynowej MD18116 $MN_MM_NUM_TOOL_ENV określa się, ile zestawów danych do opisania środowisk narzędziowych można zapisać. Dane znajdują się w obszarze TOA. Pozostają one zachowane również po wyłączeniu sterowania.

Zabezpieczenie danych nie jest możliwe. Oznacza to, że te dane nie mogą być przenoszone między różnymi sterowaniami.

11.12.2 Kasowanie środowiska narzędziowego (DELTOOLENV)

Za pomocą funkcji DELTOOLENV można kasować zestawy danych do opisu środowisk narzędziowych. Kasowanie oznacza, że zestaw danych zapisany pod określoną nazwą nie będzie już dostępny (próba uzyskania dostępu prowadzi do alarmu).

Uwaga

Zestawy danych można kasować tylko za pomocą funkcji DELTOOLENV, poprzez INITIAL.INI-Download lub zimny start (rozruch NC ze standardowymi danymi maszynowymi). Nie ma innej automatycznej procedury kasowania.

Składnia

```
<Status> = DELTOOLENV(<Name>)
<Status> = DELTOOLENV()
```

Znaczenie

DELTOOLENV(...):		Wstępnie zdefiniowana funkcja do kasowania środowiska narzędziowego		
		W oddzielnym bloku:		tak
<Status>:		Wartość zwrotna funkcji. Wartości ujemne wskazują stany błędu.		
		Typ danych:		INT
		Wartość:	0	Funkcja OK
			-1	Brak zarezerwowanego miejsca w pamięci dla środowisk narzędziowych: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkcjonalność „Środowiska narzędziowe” nie jest dostępna.
			-2	Środowisko narzędziowe o podanej nazwie nie istnieje.
Parametry				
1	<Name>:	Nazwa zestawu danych do skasowania		
		Typ danych:		STRING

DELTOOLENV():	DELTOOLENV() bez podania nazwy kasuje wszystkie zestawy danych do opisu środowiska narzędziowego

11.12.3 Odczyt numerów T, D i DL (GETTENV)

Funkcja GETTENV służy do odczytu zapisanych w środowisku narzędziowym numerów T, D i DL.

Składnia

```
<Status> = GETTENV(<Name>, <TDDL>)
```

Znaczenie

GETTENV(...):	Funkcja wstępnie zdefiniowana do odczytu numerów T, D i DL w zestawie danych do opisu środowiska narzędziowego		
	W oddzielnym bloku:		tak
<Status>:	Wartość zwrotna funkcji. Wartości ujemne wskazują stany błędu.		
	Typ danych:		INT
	Wartość:	0	Funkcja OK
		-1	Brak zarezerwowanego miejsca w pamięci dla środowisk narzędziowych: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkcjonalność „Środowiska narzędziowe” nie jest dostępna.
		-2	Środowisko narzędziowe o podanej nazwie nie istnieje.
Parametry			
1 <Name>:	Nazwa zestawów danych, z których powinien nastąpić odczyt numeru T, D i DL.		
	Typ danych:		STRING
2 <TDDL>:	Pole tego parametru wyników zawiera numer T, D i DL narzędzia, którego środowisko narzędziowe jest przechowywane w określonym zestawie danych: • <TDDL> [0]: numer T • <TDDL> [1]: numer D • <TDDL> [2]: numer DL		
	Typ danych:		INT[3]
GETTENV(,<TDDL>), GETTENV("",<TDDL>):	Przy wywołaniu funkcji GETTENV dopuszcza się pominięcie pierwszego parametru lub przekazanie łańcucha zerowego jako pierwszego parametru. W tych obu szczególnych przypadkach zwracane są w <TDDL> numery T, D i DL **aktywnego** narzędzia.		

11.12.4 Odczyt informacji do zapisanych środowisk narzędziowych ($P_TOOLENVN, ($P_TOOLENV)

Informacje do zapisanych środowisk narzędziowych są do oczytania poprzez następujące zmienne systemowe:

$P_TOOLENVN:	Podaje liczbę zestawów danych zdefiniowanych za pomocą TOOLENV (jeszcze nieskasowanych) do opisu środowisk narzędzia			
	Składnia:	<n> = $P_TOOLENVN		
	Znaczenie:	<n>:	Liczba zdefiniowanych zestawów danych	
			Typ danych:	INT
			Zakres wartości:	0 ... MD18116 $MN_MM_NUM_TOOL_ENV
	Dostęp do tych zmiennych systemowych jest dozwolony, gdy nie są możliwe żadne środowiska narzędziowe (MD18116 = 0). W tym przypadku wartością zwrotną jest „0”.			
$P_TOOLENV:	Podaje nazwę zestawu danych <i> do opisu środowiska narzędziowego			
	Składnia:	<Name> = $P_TOOLENV[<i>]		
	Znaczenie:	<Name>:	Nazwa zestawu danych z numerem <i>	
			Typ danych:	STRING
		<i>:	Numer zestawu danych	
			Typ danych:	INT
			Zakres wartości:	1 ... $P_TOOLENVN
	Przyporządkowanie numerów do zestawów danych nie jest stałe, lecz może się zmienić poprzez skasowanie lub utworzenie nowych zestawów danych. Numeracja zestawów danych jest przeprowadzana wewnętrznie. Jeżeli <i> wskazuje na niezdefiniowany zestaw danych, zwracany jest zerowy łańcuch znaków. Jeżeli indeks <i> jest nieważny, tzn. <i> jest mniejszy niż 1 lub większy niż maksymalna liczba zestawów danych dla środowisk narzędziowych (MD18116 $MN_MM_NUM_TOOLENV), wyświetlany jest następujący alarm: Alarm 17020 „Niedozwolony indeks1 tablicy"			

11.12.5 Odczyt długości narzędzia lub składowych długości narzędzia (GETTCOR)

Funkcja GETTCOR służy do odczytywania długości narzędzia lub składowych długości narzędzia.

Poprzez parametryzację można podać jakie składowe powinny zostać uwzględnione i pod jakimi warunkami należy rozpatrywać narzędzie.

Składnia

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

Znaczenie

GETTCOR(...):	Wstępnie zdefiniowana funkcja do odczytu długości narzędzia lub składowych długości narzędzia		
	W oddzielnym bloku:		tak
<Status>:	Wartość zwrotna funkcji. Wartości ujemne wskazują stany błędu.		
	Typ danych:		INT
	Wartość:	0	Funkcja OK
		-1	Brak zarezerwowanego miejsca w pamięci dla środowisk narzędziowych: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkcjonalność „Środowiska narzędziowe” nie jest dostępna.
		-2	Środowisko narzędziowe o podanej nazwie pod <Stat> nie występuje.
		-3	Nieprawidłowy łańcuch znaków w parametrze <Comp>. Przyczyną tego błędu może być nieprawidłowy lub podwójnie zaprogramowany wiersz.
		-4	Nieprawidłowy numer T
		-5	Nieprawidłowy numer D
		-6	Nieprawidłowy numer DL
		-7	Próba uzyskania dostępu do nieistniejącego modułu pamięci.
		-8	Próba uzyskania dostępu do nieistniejącej opcji (np. programowalna orientacja narzędzia, zarządzanie narzędziami).
		-9	Łańcuch znaków <Comp> zawiera dwukropek (znacznik do specyfikacji układu współrzędnych), jednakże nie zawiera ważnych znaków do oznaczenia żądanego układu współrzędnych.
Parametry			

1	<Len>:	Wektor wynikowy	
		Typ danych:	REAL[11]
		Składowe wektora są umieszczone w następującej kolejności: • <Len> [0]: Typ narzędzia • <Len> [1]: Położenie ostrza • <Len> [2]: Odcięta • <Len> [3]: Rzędna • <Len> [4]: Aplikata • <Len> [5]: Promień narzędzia Układem współrzędnych odniesienia dla składowych długości jest układ współrzędnych zdefiniowany w <Comp> i <Stat>. Jeżeli w <Comp> nie zdefiniowano żadnego układu współrzędnych, to długości narzędzi będą przedstawiane w układzie współrzędnych maszyny. Przyporządkowanie odciętej, rzędnej i aplikaty do osi geometrycznych zależy od aktywnej płaszczyzny zastosowanej w środowisku narzędziowym. Tzn. dla G17 odcięta jest równoległa do X, dla G18 do Z itd. Składowe <Len>\[6] do <Len>\[10] zawierają dodatkowe parametry, które mogą zostać podane do opisu geometrii narzędzia (np. $TC_DP7 do $TC_DP11 dla geometrii lub odpowiednie składowe dla zużycia i korekcji sumarycznych i ustawczych). Tych 5 dodatkowych elementów i promień narzędzia są zdefiniowane tylko dla składowych E, G, S i W. Ich ocena nie zależy od <Stat>. Odpowiednie wartości w <Len>[6] do <Len>[10] mogą mieć wartość inną niż zero, gdy przy obliczaniu długości narzędzia uczestniczy co najmniej jedna z czterech wymienionych składowych. Pozostałe składowe nie mają wpływu na wynik. Wymiary odnoszą się do systemu podstawowego sterowania (calowy lub metryczny).	

2	<Comp>:	Składowe długości narzędzia (opcjonalnie)			
		Typ danych:	STRING		
		Łańcuch znaków składa się z dwóch częściowych łańcuchów znaków, które oddzielone są od siebie dwukropkiem. Ogól. formuła: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**Pierwszy łańcuch częściowy** określa składowe długości narzędzia, które powinny zostać uwzględnione przy obliczaniu długości narzędzia. Kolejność znaków w łańcuchu częściowym oraz jego sposób zapisu jest dowolny. Między znakami może zostać wstawiona dowolna liczba znaków spacji lub tabulatora (white spaces). **Wskazówka:** Znaki w łańcuchu częściowym nie mogą być programowane **dwukrotnie**!		
			Znak:	–	Znak ujemny (dozwolony tylko jako pierwszy znak!) Obliczana jest całkowita długość narzędzia minus składowe określone w poniższym łańcuchu znaków.
				C	Wymiar przystawki lub wymiar bazowy (obie alternatywnie istniejące składowe, które są aktywne dla zastosowanego narzędzia)
				E	Korekcje ustawcze
				G	Geometria
				K	Transformacja kinematyczna (uwzględniana tylko dla transformacji rodzajowej 3-, 4- i 5-osiowej!).
				S	Korekcje sumaryczne
				T	Zorientowany nośnik narzędzi
				W	Zużycie
			Jeżeli pierwszy łańcuch częściowy jest pusty (za wyjątkiem white spaces), oznacza to, że obliczona ma zostać całkowita długość narzędzia z uwzględnieniem wszystkich składowych. To samo ma również zastosowanie, gdy parametr <Comp> nie jest podany.		
		<Substr_2>:	Opcjonalny **drugi łańcuch częściowy** określa układ współrzędnych, w którym powinna zostać wyświetlana długość narzędzia. Drugi łańcuch częściowy składa się tylko z pojedynczych, istotnych znaków.		
			Znaki:	A	Ustawiany układ współrzędnych (ACS)
				B	Układ współrzędnych bazowy (BKS)
				K	Układ współrzędnych narzędzia transformacji kinematycznej (KCS)
				M	Układ współrzędnych maszyny (MKS)
				T	Układ współrzędnych narzędzia (TCS)
				W	Układ współrzędnych przedmiotu obrabianego (WKS)
			Jeżeli nie podano żadnego układu współrzędnych, ocena następuje w układzie współrzędnych maszyny (MKS). Obroty, któ-		

			re mogą wymagać uwzględnienia są definiowane poprzez środowisko narzędziowe zdefiniowane w <Stat>.
3	<_Stat>:	Nazwa zestawu danych do opisu środowiska narzędziowego (opcjonalnie)	
		Typ danych:	STRING
		Jeżeli wartością tego parametru jest łańcuch zerowy ("") lub wartość nie jest podana, to stosowany jest aktualny stan. Jeśli nie określono żadnego narzędzia, używane jest aktualne narzędzie.	
4	<T>:	Wewnętrzny numer T narzędzia (opcjonalnie)	
		Typ danych:	INT
		Jeżeli parametr ten nie jest podany lub jego wartość wynosi „0”, to stosowane jest narzędzie utworzone w <Stat>. Jeżeli wartość tego parametru wynosi „-1”, to stosowany jest numer T aktywnego narzędzia. Bezpośrednie wprowadzanie numeru aktywnego narzędzia nie jest dozwolone. **Wskazówka:** Jeżeli nie podano <Stat>, to jako środowisko narzędziowe stosowany jest aktualny stan. Ponieważ <T> = 0 odnosi się do numeru T zapisanego w środowisku narzędziowym, używane jest aktywne narzędzie, tzn. dane <T> = 0 i <T> = -1 są w tym szczególnym przypadku równoznaczne.	
5	<D>:	Numer ostrza narzędzia (opcjonalnie)	
		Typ danych:	INT
		Jeżeli parametr ten nie jest podany lub jego wartość wynosi „0”, to zastosowany numer D zależy od źródła numeru T. Jeżeli zastosowany jest numer T ze środowiska narzędziowego, to odczytywany jest także numer D środowiska narzędziowego, w przeciwnym razie numer aktualnego aktywnego narzędzia.	
6	<DL>:	Numer korekcji zależnej od miejsca (opcjonalnie)	
		Typ danych:	INT
		Jeżeli parametr nie jest podany, to zastosowany numer DL zależy od źródła numeru T. Jeżeli zastosowany jest numer T ze środowiska narzędziowego, to odczytywany jest także numer D środowiska narzędziowego, w przeciwnym razie numer aktualnego aktywnego narzędzia.	

Przykłady

`GETTCOR(_LEN)`	Oblicza długość narzędzia aktualnie aktywnego narzędzia w układzie współrzędnych maszyny z uwzględnieniem wszystkich składowych.
`GETTCOR(_LEN,"CGW:W")`	Długość narzędzia jest obliczana na podstawie wymiaru przystawki lub wymiaru bazowego, geometrii i zużycia dla aktywnego narzędzia. Dalsze składowe jak np. zorientowany nośnik narzędzi lub transformacje kinematyczne nie są uwzględniane. Wyświetlenie następuje w układzie współrzędnych przedmiotu obrabianego.

`GETTCOR (_LEN,"-K:B")`	Całkowita długość aktywnego narzędzia jest obliczana bez uwzględnienia składowych długości aktywnej transformacji kinematycznej. Wyświetlenie następuje w podstawowym układzie współrzędnych.
`GETTCOR (_LEN,":M","Testenv1",,3)`	Całkowita długość narzędzia dla narzędzia utworzonego w środowisku narzędziowym o nazwie „Testenv1” jest obliczana w układzie współrzędnym maszyny. Obliczenie następuje jednak niezależnie od utworzonego numeru ostrza dla numeru ostrza D3.

Dodatkowe informacje

Transformacja przystawki / Orientowane nośniki narzędzi / Transformacja kinematyczna

Obroty i zmiana składowych, które są wykonywane przez transformacje przystawki, orientowanych nośników narzędzi i transformację kinematyczną, są częściami składowymi środowiska narzędziowego. Są one dlatego zawsze wykonywane, również wtedy, gdy odpowiednie składowe długości nie powinny zostać uwzględniane. Jeżeli nie jest to wymagane, należy definiować środowiska narzędziowe, dla których odpowiednie transformacje nie są aktywne. W wielu przypadkach (zawsze wtedy, gdy na maszynie nie ustawiono żadnej transformacji lub orientowanego nośnika narzędzi), zapisane zestawy danych dla środowisk narzędziowych wypełniają automatycznie te warunki, i dlatego użytkownik nie musi specjalnie zwracać na nie uwagi.

Narzędzia tokarskie i szlifierskie: Obliczenie długości narzędzia w zależności od MD20360 $MC_TOOL_PARAMETER_DEF_MASK

To jak oszacować zużycie lub długość narzędzia w istniejącej osi średnicy w przypadku narzędzi tokarskich i szlifierskich określone jest w danej maszynowej:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK (definicja parametrów narzędzia)

Bit	Wartość	
0	W przypadku narzędzi tokarskich i szlifierskich wliczany jest **parametr zużycia** osi poprzecznej jako wartość średnicy.	
	= 0 (domyślnie)	nie
	= 1	tak
1	W przypadku narzędzi tokarskich i szlifierskich wliczana jest **składowa długości narzędzia** osi poprzecznej jako wartość średnicy.	
	= 0 (domyślnie)	nie
	= 1	tak

Jeżeli określone bity zostały ustalone, przynależny wpis jest szacowany za pomocą współczynnika 0,5. To oszacowanie odzwierciedla podane długości narzędzia w GETTCOR.

Przykład:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 $MC_DIAMETER_AX_DEF (oś geometryczna z funkcją osi poprzecznej) = "X"

X jest osią średnicy (standardowa konfiguracja tokarki)

Kod programu	**Komentarz**
N30 $TC_DP1[1,1]=500	
N40 $TC_DP2[1,1]=2	
N50 $TC_DP3[1,1]=3.0	; Geometria L1
N60 $TC_DP4[1,1]=4.0	
N70 $TC_DP5[1,1]=5.0	
N80 $TC_DP12[1,1]=12.0	; Zużycie L1
N90 $TC_DP13[1,1]=13.0	
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	

Składowe długości transformacji kinematycznej i orientowanego nośnika narzędzi

Jeżeli przy obliczeniu długości narzędzia uwzględniony jest **orientowany nośnik narzędzia**, to następujące wektory są uwzględniane w obliczeniu długości narzędzia.

Typ	Wektory
M	l1 i l2
T	l1, l2 i l3
P	Długość narzędzia nie podlega wpływowi orientowanego nośnika narzędzi.

W przypadku **5-osiowej transformacji** rodzajowej, następujące dane maszynowe, dla typu transformacji 24 i 56, uwzględniane są w obliczeniu długości narzędzia:

Typ transformacji	Dane maszynowe
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

Transformacja 56 (ruchome narzędzia i przedmioty obrabiane) odpowiada typowi M w przypadku orientowanych nośników narzędzi.

W dotychczasowych wersjach oprogramowania, przy tej transformacji 5-osiowej wektor MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 (wektor przesunięcia kinematycznego 1./2. 5-osiowej transformacji w kanale) odpowiada sumie obu wektorów l_1 i l_3 w przypadku orientowanych nośników narzędzia typu M.

Dla transformacji w obu przypadkach istotna jest tylko suma. Rodzaj złożenia z obu pojedynczych składowych jest bez znaczenia. Przy obliczaniu długości narzędzia istotne jest, która składowa narzędzia i który stół narzędziowy jest przyporządkowany. Dlatego wstawiono daną maszynową MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 (wektor

przesunięcia kinematycznego w stole). Odpowiada ona wektorowi l3. Dana maszynowa MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 nie odpowiada już sumie z l1 i l3, lecz jedynie wektorowi l1. Jeżeli dana maszynowa MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 ma wartość zero, to działanie jest identyczne z dotychczasowym.

Kompatybilność

Funkcja GETTCOR wraz z funkcją TOOLENV i SETTCOR służą między innymi do tego, aby zastąpić część funkcjonalności, która dotychczas realizowana była zewnętrznie w cyklach pomiarowych.

W cyklach pomiarowych uwzględniono tylko część parametrów, które ostatecznie określają rzeczywistą długość narzędzia. Wymienione funkcje można tak sparametryzować, że działanie cykli pomiarowych związane z obliczeniem długości narzędzia może zostać odtworzone.

11.12.6 Zmiana składowych narzędzia (SETTCOR)

Funkcja SETTCOR służy do zmiany składowych narzędzia przy uwzględnieniu wszystkich warunków brzegowych, które można uwzględnić w ocenie poszczególnych składowych.

Uwaga

Do terminologii: W odniesieniu do długości składowych narzędzia, należy rozumieć składowe wektorowe, które składają się na całkowitą długość narzędzia (np. geometria lub zużycie). Takie komponenty składają się z trzech wartości jednostkowych (L1, L2, L3), które poniżej zostały określone jako wartości współrzędnych.

Składowa narzędzia „Geometria” składa się więc przykładowo z trzech wartości współrzędnych $TC_DP3 do $TC_DP5.

Składnia

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>,
<Stat>, <T>, <D>, <DL>])
```

Znaczenie

SETTCOR(...):	Wstępnie zdefiniowane funkcje do zmiany składowych narzędzia	
	W oddzielnym bloku:	tak

<Status>:	Wartość zwrotna funkcji. Wartości ujemne wskazują stany błędu.		
	Typ danych:		INT
	Wartość:	0	Funkcja OK
		-1	Brak zarezerwowanego miejsca w pamięci dla środowisk narzędziowych: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkcjonalność „Środowiska narzędziowe” nie jest dostępna.
		-2	Środowisko narzędziowe o podanej nazwie pod <Stat> nie występuje.
		-3	Nieprawidłowy łańcuch znaków w parametrze <Comp>. Przyczyną tego błędu może być nieprawidłowy lub podwójnie zaprogramowany wiersz.
		-4	Nieprawidłowy numer T
		-5	Nieprawidłowy numer D
		-6	Nieprawidłowy numer DL
		-7	Próba uzyskania dostępu do nieistniejącego modułu pamięci.
		-8	Próba uzyskania dostępu do nieistniejącej opcji (np. programowalna orientacja narzędzia, zarządzanie narzędziami).
		-9	Niedozwolona wartość liczbowa dla parametru <CorComp>.
		-10	Niedozwolona wartość liczbowa dla parametru <CorMode>.
		-11	Zawartości parametrów <Comp> i <CorComp> są niespójne.
		-12	Zawartości parametrów <Comp> i <CorMode> są niespójne.
		-13	Zawartość parametru <GeoAx> nie określa osi geometrycznej.
		-14	Próba uzyskania dostępu do nieistniejącej korekcji ustawiania.
Parametry			
1	<CorVal>:	Wektor korekcji W układzie współrzędnych obrabianego przedmiotu (WKS) zdefiniowanym przez <Stat> obowiązuje następujące przyporządkowanie: • <CorVal> [0]: Odcięta • <CorVal> [1]: Rzędna • <CorVal> [2]: Aplikata Jeśli ma być skorygowana tylko jedna składowa (tj. bez korekcji wektorowej, patrz parametr <CorMode>), wartość korekcji jest zawsze w <CorVal> [0], niezależnie od tego na którą oś działa. Zawartość obu pozostałych składowych nie jest wtedy wykorzystywana. Jeżeli <CorVal> lub składowa <CorVal> odnosi się do osi poprzecznej, to wykorzystywane jest podawanie **wymiarów na promieniu**. Oznacza to, że narzędzie jest np. o podany wymiar „dłuższe”, co prowadzi do odpowiednio dwukrotnie większej zmiany średnicy przedmiotu obrabianego. Wymiary odnoszą się do **systemu podstawowego** sterowania (calowy lub metryczny).	
		Typ danych:	REAL[3]

2	<Comp>:	Składowa(e) narzędzia			
		Typ danych:	STRING		
		Łańcuch znaków składa się z dwóch częściowych łańcuchów znaków, które oddzielone są od siebie dwukropkiem. Ogól. formuła: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**Pierwszy łańcuch** musi być zawsze dostępny i może składać się z jednego lub dwóch znaków. Pierwszy lub jedyny znak oznacza 1. składową (Val_1) i drugi znak 2. składową (Val_2), które są przetwarzane zgodnie z następującymi parametrami <CorComp> i <CorMode>.		
			Znaki:	C	Wymiar przystawki lub wymiar bazowy (obie alternatywnie istniejące składowe, które są aktywne dla zastosowanego narzędzia)
				E	Korekcje ustawcze
				G	Geometria
				S	Korekcje sumaryczne
				W	Zużycie
		<Substr_2>:	**Drugi łańcuch częściowy** jest opcjonalny. Alternatywnie może on składać się (jedynie) z liter „W” lub „T”.		
			Znaki:	W	Jeżeli drugi łańcuch częściowy jest pusty lub zawiera literę „W”, to wartości korekcji będą tak obliczane, jakby były mierzone w układzie współrzędnych **przedmiotu** obrabianego (WKS).
				T	Jeżeli drugi łańcuch częściowy zawiera literę „T”, wartości korekcji będą tak obliczane, jakby były mierzone w układzie współrzędnych narz**ędzia** (**T**ool Coordinate System, TCS).
		Sposób zapisu znaków w łańcuchu (małe lub duże) jest dowolny. Można wstawić dowolną liczbę znaków spacji lub tabulatora (white spaces).			

3	<Comp>:	Określa komponenty zestawu danych narzędzia, które powinny zostać opisane (opcjonalnie).		
		Typ danych:		INT
		Wartość:	0	Wartość korekcji <CorVal>[0] odnosi się do osi geometrycznej przekazanej w parametrze <GeoAx> w układzie współrzędnych przedmiotu obrabianego lub w układzie współrzędnych narzędzia (patrz opis parametru <Comp>). Tzn. wartość korekcji musi być tak wliczana do określonych składowych narzędzia, że uwzględniając wszystkie parametry, które mogą wpłynąć na obliczenie długości narzędzia, wynikiem jest zmiana całkowitej długości narzędzia w podanym kierunku osi o zadaną wartość. Ta zmiana zostanie osiągnięta poprzez korekcję podanej pod <Comp> składowej i podanej symbolicznej zasady obliczania pod <CorMode> (patrz następne parametry). Korekcja wynikowa może zatem wpłynąć na wszystkie trzy składowe osi.
			1	Jak „0”, jednakże wektorowo. Zawartość wektora <CorVal> odnosi się do odciętej, rzędnej i aplikaty w układzie współrzędnych przedmiotu obrabianego lub układu współrzędnych narzędzi (patrz opis parametru <Comp>). Następny parametr <GeoAx> nie jest uwzględniany.
			2	Korekcja wektorowa, tzn. L1, L2 i L3 może być zmieniana symultanicznie. W przeciwieństwie do wariantów „0” i „1” wartości korekcji zawartych w <CorVal> odnoszą się do współrzędnych składowej Val_1 narzędzia (patrz następny parametr <CorMode>). Ewentualne położenie skośne narzędzia względem układu współrzędnych obrabianego przedmiotu nie ma wpływu na korekcję.
			3 - 5	Korekcja długości narzędzia L1 do L3 ($TC_DP3 do $TC_DP5) lub odpowiednich wartości dla zużycia, korekcji sumarycznej i ustawczej. Wartość korekcji jest zawarta w <CorVal>[0]. Jest ona mierzona we współrzędnych składowej narzędzia Val_1 (patrz następny parametr <CorMode>). Ewentualne położenie skośne narzędzia względem układu współrzędnych obrabianego przedmiotu nie ma wpływu na korekcję.
			6	Korekcja promienia narzędzia ($TC_DP6) lub odpowiednich wartości dla zużycia, korekcji sumarycznej i ustawianej. Bity 10 i 11 (szacowanie danych średnicy lub danych zużycia średnicy opcjonalnie jako dane na promieniu lub na średnicy) są uwzględniane w danej maszynowej MD20360 $MC_TOOL_PARAMETER_DEF_MASK.
			7 – 11	Korekcja $TC_DP7 do $TC_DP11 lub odpowiednie wartości dla zużycia, korekcji sumarycznej i ustawczej. Te parametry są traktowane jak promień narzędzia.
		Jeżeli parametr ten nie jest podany, jego wartość wynosi „0”.		

<table>
<tr><td>4</td><td><CorMode>:</td><td colspan="3">Określa rodzaj wykonywanych operacji zapisu (opcjonalnie)</td></tr>
<tr><td></td><td></td><td colspan="2">Typ danych:</td><td>INT</td></tr>
<tr><td></td><td></td><td rowspan="4">Wartość:</td><td>0</td><td>Val_{1nowy} = <CorVal></td></tr>
<tr><td></td><td></td><td>1</td><td>Val_{1nowy} = Val_{1stary} + <CorVal></td></tr>
<tr><td></td><td></td><td>2</td><td>Val_{1nowy} = <CorVal>
Val_{2nowy} = 0</td></tr>
<tr><td></td><td></td><td>3</td><td>Val_{1nowy} = Val_{1stary} + Val_{2stary} + <CorVal>
Val_{2nowy} = 0</td></tr>
<tr><td></td><td></td><td colspan="3">Sposób zapisu Val_{1stary} + Val_{2stary} należy rozumieć symbolicznie. Jeżeli obie składowe są szacowane w różny sposób (z powodu stanu w <_Stat>), tzn. między dwoma składowymi jest aktywny obrót, to Val_{2stary} jest tak przekształcany przed dodaniem, że wynikowe długości narzędzia pozostają niezmienione po skasowaniu Val_{2nowy} i przed dodaniem <CorVal>.
<CorVal> odnosi się zawsze do Val_1. <CorVal> jest wartością, która w zależności od drugiej części parametru <Comp> jest mierzona w układzie współrzędnych obrabianego przedmiotu (WKS) lub układzie współrzędnych narzędzia (TCS). Dlatego jest ona już przekształcona względem składowej narzędzia, w której powinna być utworzona. Nie może być ona zatem uwzględniona bezpośrednio za pomocą zapisanej wartości, lecz musi zostać przed dodaniem ponownie przekształcona do Val_1 lub Val_2. Może to spowodować, że korekcja będzie miała wpływ w innej osi niż ta zdefiniowana przez <CorComp> lub że wpłynie na kilka osi.
W przypadku <CorComp> = 0 tzn., gdy <CorVal> nie zawiera wektora, lecz tylko wartość jednostkową, wykonywane będą opisane operacje w układzie współrzędnych, w którym nastąpił pomiar <CorVal> (WKS / TCS). Dotyczy to w szczególności ustawiania Val_{2nowy} na zero w wariancie 2 i 3. Wynik jest wtedy ponownie przekształcany w układzie współrzędnych narzędzia. W efekcie może być tak, że żadna z wartości współrzędnych ustawiona na zero (L1, L2, L3) nie jest zerem lub wartości współrzędnych, które wcześniej były zerem teraz nie są równe zeru. Jeżeli jednak odpowiednie operacje są wykonywane kolejno dla wszystkich trzech osi geometrycznych, zapewnia się, że wszystkie trzy wartości współrzędnych skasowanego komponentu stają się zerowe. Jeżeli narzędzie nie obraca się względem układu współrzędnych przedmiotu obrabianego lub tak się obraca, że wszystkie składowe narzędzia pozostają równolegle do osi współrzędnych (zmiany osi), zapewnia się, że zmienia się tylko współrzędna narzędzia.
Wykonywanie takiej samej operacji (<CorMode>) z <CorComp> = 0 dla wszystkich osi współrzędnych w dowolnej kolejności jest identyczne, jak jednorazowe wykonanie tej samej operacji za pomocą <CorComp>=1.
Dla wartości parametrów „0” i „1”, parametr <Comp> musi zawierać jeden, a dla wartości parametrów „2” i „3” dwa znaki.
Przykład:
<Comp> zawiera łańcuch znaków "ES", <CorMode> wartość "2"
⇒ Korekcja ustawiania$_{nowa}$ = <CorVal>, korekcja sumaryczna$_{nowa}$ = 0
Jeżeli parametr <CorMode> nie jest podany, jego wartość wynosi „0”.</td></tr>
<tr><td>5</td><td><GeoAx>:</td><td colspan="3">Podaje indeks osi geometrycznej, w której wartość korekcji <CorVal>[0] została zmierzona (opcjonalnie).</td></tr>
<tr><td></td><td></td><td colspan="2">Typ danych:</td><td>INT</td></tr>
<tr><td></td><td></td><td colspan="2">Zakres wartości:</td><td>0 ... 2</td></tr>
<tr><td></td><td></td><td colspan="3">Indeksy od 0 do 2 odnoszą się do odciętej, rzędnej i aplikaty w działającej płaszczyźnie ((G17/G18/G19) aktualnego środowiska narzędziowego.
Zawartość tego parametru jest tylko wtedy szacowana, gdy parametr <CorComp> ma wartość „0”.</td></tr>
</table>

6	<Stat>:	Nazwa zestawu danych do opisu środowiska narzędziowego (opcjonalnie)	
		Typ danych:	STRING
		Jeżeli wartością tego parametru jest zerowy łańcuch znaków ("") lub wartość nie jest podana, to stosowany jest aktualny stan. Jeśli nie określono żadnego narzędzia, używane jest aktualne narzędzie.	
7	<T>:	Wewnętrzny numer T narzędzia (opcjonalnie)	
		Typ danych:	INT
		Jeżeli parametr ten nie jest podany lub jego wartość wynosi „0”, to stosowane jest narzędzie utworzone w <Stat>. Jeżeli wartość tego parametru wynosi „-1”, to stosowany jest numer T aktywnego narzędzia. Bezpośrednie wprowadzanie numeru aktywnego narzędzia nie jest dozwolone. **Wskazówka:** Jeżeli nie podano <Stat>, to jako środowisko narzędziowe stosowany jest aktualny stan. Ponieważ <T> = 0 odnosi się do numeru T zapisanego w środowisku narzędziowym, używane jest aktywne narzędzie, tzn. dane <T> = 0 i <T> = -1 są w tym szczególnym przypadku równoznaczne.	
8	<D>:	Numer ostrza narzędzia (opcjonalnie)	
		Typ danych:	INT
		Jeżeli parametr ten nie jest podany lub jego wartość wynosi „0”, to zastosowany numer D zależy od źródła numeru T. Jeżeli zastosowany jest numer T ze środowiska narzędziowego, to odczytywany jest także numer D środowiska narzędziowego, w przeciwnym razie numer aktualnego aktywnego narzędzia.	
9	<TL>:	Numer korekcji zależnej od miejsca (opcjonalnie)	
		Typ danych:	INT
		Jeżeli parametr nie jest podany, to zastosowany numer DL zależy od źródła numeru T. Jeżeli zastosowany jest numer T ze środowiska narzędziowego, to odczytywany jest także numer D środowiska narzędziowego, w przeciwnym razie numer aktualnego aktywnego narzędzia. Określone numery T, D i DL narzędzia bez korekcji zależnej od miejsca, nie mogą mieć wyspecyfikowanych w parametrze <Comp> korekcji sumarycznych lub ustawczych (kod błędu w <Status>).	

Uwaga

Nie wszystkie możliwe kombinacje trzech parametrów <Comp>, <CorComp> i <CorMode> mają znaczenie. Przykładowo reguła obliczania 3 w <CorComp> wymaga podania dwóch znaków w <Comp>. Jeżeli wprowadzono niepoprawną kombinację parametrów, w <Status> zwracany jest odpowiedni kod błędu.

Przykłady

Przykład 1

```
Kod programu                                Komentarz
N10 DEF REAL _CORVAL[3]
N20 $TC_DP1[1,1]= 120                       ; Narzędzie frezarskie
N30 $TC_DP3[1,1]=10.0                       ; Geometria L1
N40 $TC_DP12[1,1]=1.0                       ; Zużycie L1
```

Kod programu	Komentarz
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"G",0,0,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> Pozycja MKS X0.000 Y0.000 **Z1.333**
N90 M30	

<CorComp> wynosi „0”, dlatego działająca w kierunku Z wartość współrzędnych składowej geometrii jest zastępowana przez wartość korekcji 0,333.

Tak więc wynikowa całkowita długość narzędzia wynosi: L1 = 0,333 + 1,000 = 1,333

Przykład 2

Kod programu	Komentarz
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]= 120	; Narzędzie frezarskie
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Zużycie 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	; ==> Pozycja MKS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> wynosi „1”, dlatego działająca w kierunku Z wartość korekcji 0,333 jest dodawana do wartości zużycia 1,0.

Tak więc wynikowa całkowita długość narzędzia wynosi: L1 = 10,0 + 1,333 = 11,333

Przykład 3

Kod programu	Komentarz
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]= 120	; Narzędzie frezarskie
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Zużycie 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	; ==> Pozycja MKS X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> wynosi „2”, dlatego działająca w kierunku Z korekcja jest wpisywana do składowej geometrii (stara wartość jest nadpisywana), a wartość zużycia jest kasowana.

Tak więc wynikowa całkowita długość narzędzia wynosi: L1 = 0,333 + 0,0 = 0,333

Przykład 4

Kod programu	Komentarz
N10 DEF REAL _CORVAL[3]	

Kod programu	Komentarz
N20 $TC_DP1[1,1]= 120	; Narzędzie frezarskie
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Zużycie 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	; ==> Pozycja MKS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> wynosi „3”, dlatego wartość zużycia i wartość korekcji jest dodawana do składowej geometrii, a składowa zużycia jest kasowana.

Tak więc wynikowa całkowita długość narzędzia wynosi: L1 = 11,333 + 0,0 = 11,333.

Przykład 5

Kod programu	Komentarz
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]= 120	; Narzędzie frezarskie
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Zużycie 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	; ==> Pozycja MKS X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> wynosi 3 jak w poprzednim przykładzie, jednakże korekcja działa teraz na oś geometryczną z indeksem „0” (oś X), która z powodu G17 dla narzędzia frezarskiego jest przyporządkowana do składowej narzędzia L3. Dlatego wywołanie SETTCOR nie wpływa na parametry narzędzia $TC_DP3 i $TC_DP12. Wartość korekcji jest zamiast tego wpisywana w $TC_DP5.

Przykład 6

Kod programu	Komentarz
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Narzędzie tokarskie
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP4[1,1]=15.0	; Geometria L2
N50 $TC_DP12[1,1]=10.0	; Zużycie L1
N60 $TC_DP13[1,1]=0.0	; Zużycie L2
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	; ==> Pozycja MKS X24.330 Y0.000 Z17.500
N120 M30	

Narzędziem jest narzędzie tokarskie. W bloku N80 jest aktywny obrót frame, dlatego bazowy układ współrzędnych (BKS) jest obracany względem układ współrzędnych obrabianego przedmiotu (WKS). Wartość korekcji (N70) działa w WSK na oś geometryczną z indeksem „1”, tzn. ponieważ funkcja G18 jest aktywna na osi X. Ponieważ <CorMode> ma wartość 3, zużycie narzędzia w WKS w kierunku osi X, po wykonaniu N100 musi mieć wartość zero.

Dlatego zawartość istotnych parametrów narzędzia jest na końcu programu:

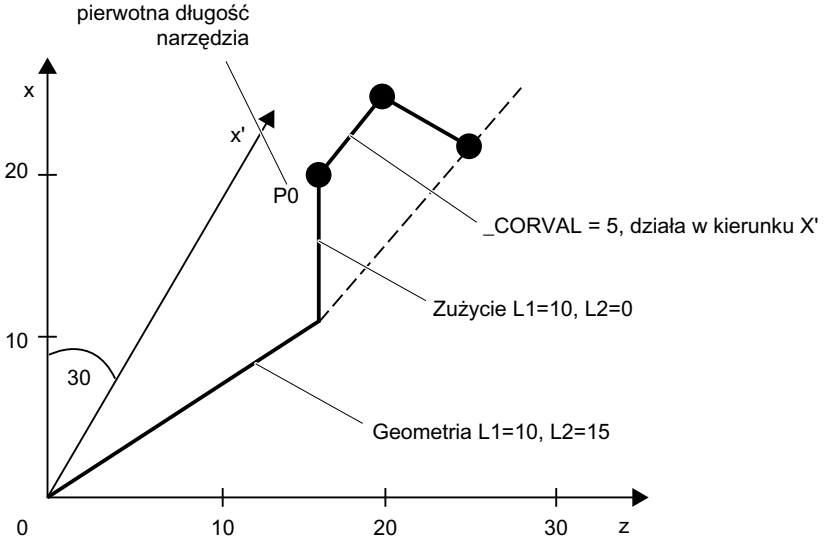
$TC_DP3[1,1] : 21.830 ; Geometria L1

$TC_DP4[1,1] : 21.830 ; Geometria L2

$TC_DP12[1,1] : 2.500 ; Zużycie L1

$TC_DP13[1,1] : -4.330 ; Zużycie L2

Relacje geometryczne przedstawiono na ilustracji poniżej: Całkowite zużycie łącznie z _CORVAL jest przedstawione w kierunku X' w WKS. Daje to punkt P2. Współrzędne tego punktu (mierzone we współrzędnych X-Y) są wpisywane do składowej geometrycznej narzędzia. W zużyciu zostaje wektor różnicy P_2 - P_1. Tym samym zużycie nie ma już żadnej składowej w kierunku _CORVAL.



Jeżeli przykładowy program według N110 jest kontynuowany z poniższymi instrukcjami, pozostałe zużycie jest w całości przejmowane do geometrii, ponieważ korekcja działa teraz w osi Z (parametr <GeoAx> = 0):

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> Pozycja MKS X24.330 Y0.000 Z17.500
```

Ponieważ nowa wartość korekcji wynosi „0”, całkowita długość narzędzia i tym samym przejmowane pozycje w N140 nie mogą się zmienić. Jeśli _CORVAL w N120 nie byłby równy „0”, to wynikiem byłaby nowa długość całkowita narzędzia, a zatem i zmieniona pozycja w N140. Składowa zużycia długości narzędzia wynosiłaby w każdym przypadku zero, tzn. całkowita długość narzędzia jest następnie uwzględniana we wszystkich przypadkach w składowej geometrii narzędzia.

Taki sam wynik, jak za pomocą dwukrotnego wywołania funkcji SETTCOR z parametrem <CorComp> = 0, osiąga się również poprzez jednokrotne wywołanie za pomocą <CorComp> = 1 (korekcja wektorowa):

```
Kod programu                                Komentarz
N10 DEF REAL _CORVAL[3]
N20 $TC_DP1[1,1]=500                        ; Narzędzie tokarskie
N30 $TC_DP3[1,1]=10.0                       ; Geometria L1
N40 $TC_DP4[1,1]=15.0                       ; Geometria L2
N50 $TC_DP12[1,1]=10.0                      ; Zużycie L1
N60 $TC_DP13[1,1]=0.0                       ; Zużycie 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                         ; ==> Pozycja MKS X24.330 Y0.000 Z17.500
N120 M30
```

W tym przypadku wszystkie składowe zużycia narzędzia będą zerowane natychmiast po pierwszym wywołaniu SETTCOR w N100.

Przykład 7

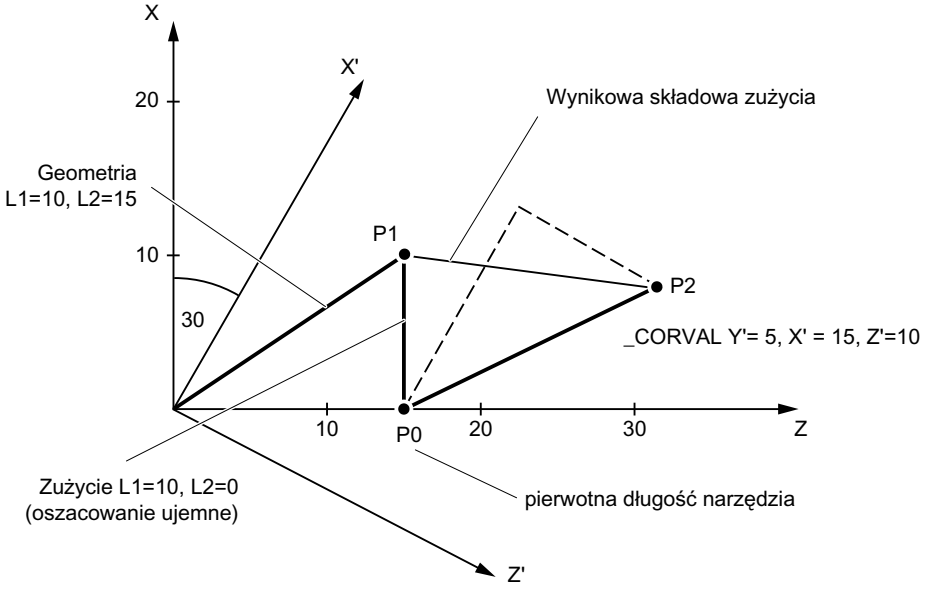
```
Kod programu                                Komentarz
N10 DEF REAL _CORVAL[3]
N20 $TC_DP1[1,1]=500                        ; Narzędzie tokarskie
N30 $TC_DP3[1,1]=10.0                       ; Geometria L1
N40 $TC_DP4[1,1]=15.0                       ; Geometria L2
N50 $TC_DP12[1,1]=10.0                      ; Zużycie L1
N60 $TC_DP13[1,1]=0.0                       ; Zużycie 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                         ; ==> Pozycja MKS X25.000 Y0.000 Z15.000
N120 M30
```

W porównaniu do przykładu 6 parametr <CorComp> wynosi tu 3, dlatego można pominąć podanie parametru <GeoAx>. Wartość zawarta w _CORVAL[0] działa teraz bezpośrednio na składową narzędzia L1, obrót w N80 nie ma wpływu na wynik, składowe zużycia w $TC_DP12 są razem z _CORVAL[0] przejmowane do składowych geometrii, dlatego w związku z $TC_DP13 całkowita długość narzędzia już po pierwszym wywołaniu SETTCOR w N100 w składowej geometrii.

Przykład 8

```
N10 DEF REAL _CORVAL[3]
N20 $TC_DP1[1,1]=500                    ; Narzędzie tokarskie
N30 $TC_DP3[1,1]=10.0                   ; Geometria L1
N40 $TC_DP4[1,1]=15.0                   ; Geometria L2
N50 $TC_DP5[1,1]=20.0                   ; Geometria L3
N60 $TC_DP12[1,1]=10.0                  ; Zużycie L1
N70 $TC_DP13[1,1]=0.0                   ; Zużycie L2
N80 $TC_DP14[1,1]=0.0                   ; Zużycie 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                     ; ==> Pozycja MKS X7.990 Y25.000 Z31.160
N170 M30
```

W N90 ustawiona jest dana ustawcza SD42930 $SC_WEAR_SIGN, tzn. zużycie musi być oszacowane ze znakiem ujemnym. Korekcja jest wektorowa (<CorComp> = 1), a wektor korekcji należy dodać do zużycia (<CorMode> = 1). Stosunki geometryczne w płaszczyźnie Z-X przedstawiono na ilustracji poniżej:



Ze względu na wartość <CorMode> równą 1, składowa geometrii narzędzia pozostaje niezmieniona. Wektor korekcji zdefiniowany w WKS (obrót wokół osi y) musi zostać przejęty do składowej zużycia w taki sposób, że całkowita długość narzędzia na rysunku 3 wskazuje na punkt P_2. Wynikowa składowa zużycia narzędzia jest zatem określana przez odległość obu

punktów P_1 i P_2.
Ponieważ zużycie z powodu danej ustawczej SD42930 $SC_WEAR_SIGN jest szacowane ujemnie, określona w ten sposób korekcja ze znakiem ujemnym musi zostać zapisana w pamięci korekcji. Dlatego zawartość istotnych parametrów narzędzia jest na końcu programu:

$TC_DP3[1,1] : 10.000 ; Geometria L1 (stała)

$TC_DP4[1,1] : 15.000 ; Geometria L2 (stała)

$TC_DP5[1,1] : 10.000 ; Geometria L3 (stała)

$TC_DP12[1,1] : 2.010 ; Zużycie L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1] : -16.160 ; Zużycie L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1] : -5.000 ; Zużycie L3

Efekt ustawienia danej ustawczej SD42930 $SC_WEAR_SIGN można rozpoznać na składowej L3 w kierunku Y bez dodatkowej komplikacji poprzez obrót frame.

Dodatkowe informacje

Narzędzia tokarskie / szlifierskie: Obliczenie długości narzędzia w zależności od MD20360 $MC_TOOL_PARAMETER_DEF_MASK

Podobnie jak w przypadku narzędzi tokarskich / szlifierskich, zużycie lub długość narzędzia w istniejącej osi średnicy określone jest w danej maszynowej:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <wartość>

<Bit>	<Wartość>	Znaczenie
0	0	W przypadku narzędzi tokarskich / szlifierskich **parametr zużycia** osi poprzecznej zawarty jest w **wartości promienia**.
	1	W przypadku narzędzi tokarskich / szlifierskich **parametr zużycia** osi poprzecznej zawarty jest w **wartości średnicy**.
1	0	W przypadku narzędzi tokarskich / szlifierskich **składowa długości narzędzia** osi poprzecznej zawarta jest w **wartości promienia**.
	1	W przypadku narzędzi tokarskich / szlifierskich **składowa długości narzędzia** osi poprzecznej zawarta jest w **wartości średnicy**.

Jeżeli określone bity zostały ustalone, przynależny wpis jest szacowany za pomocą współczynnika 0,5. Korekcja za pomocą SETTCOR jest wykonywana w ten sposób, że całkowita rzeczywista zmiana długości narzędzia jest równa wartości przekazanej w <CorVal>. Jeżeli długość, ze względu na daną maszynową MD20360 $MC_TOOL_PARAMETER_DEF_MASK, jest szacowana przy obliczaniu długości za pomocą współczynnika 0,5, to korekcja tej składowej musi nastąpić za pomocą podwójnie przekazanej wartości.

Przykład

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 2 (długość narzędzia należy oszacować w osi średnicy za pomocą współczynnika 0,5)

Oś X jest osią średnicy.

```
N10 DEF REAL _LEN[11]
N20 DEF REAL _CORVAL[3]
N30 $TC_DP1[1,1]=500                    ; Typ narzędzia
N40 $TC_DP2[1,1]=2                      ; Położenie ostrza
N50 $TC_DP3[1,1]=3.                     ; Geometra - długość 1
N60 $TC_DP4[1,1]=4.                     ; Geometra - długość 2
N70 $TC_DP5[1,1]=5.                     ; Geometra - długość 3
N80 _CORVAL[0]=1.
N90 _CORVAL[1]=1.
N100 _CORVAL[2]=1.
N110 T1 D1 G18 G0 X0 Y0 Z0              ; ==> Pozycja MKS X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)
N130 T1 D1 X0 Y0 Z0                     ; ==> Pozycja MKS X2.5 Y6 Z5
N140 R3=$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
```

Korekcja długości narzędzia powinna w każdej osi wynosić 1 mm (N80 do N100). Dlatego w długościach L2 i L3 dodaje się 1 mm do pierwotnej długości. Do pierwotnej długości narzędzia w L1 jest zaś dodawana podwójna wartość korekcji (2 mm), aby długość całkowita narzędzia zmieniła się zgodnie z wymaganiami o 1 mm. Przejmowane pozycje w blokach N110 i N130 rozpoznaje się po tym, że każda pozycja osi zmieniła się o 1 mm.

11.13 Odczyt przyporządkowania długości narzędzia L1, L2, L3 do osi współrzędnych (LENTOAX)

Funkcja LENTOAX podaje informacje o przyporządkowaniu długości **aktywnego** narzędzia L1, L2 i L3 do odciętej, rzędnej i aplikaty. Na przyporządkowanie odciętej, rzędnej i aplikaty do osi geometrycznych mają wpływ frame i aktywna płaszczyzna (G17 - G19).

Rozpatrywana jest przy tym tylko składowa geometrii narzędzia ($TC_DP3[<t>,<d>] do $TC_DP5[<t>,<d>]), tzn. ewentualnie odmienne od tego przyporządkowanie osi dalszych składowych (np. zużycie) nie wpływa na wynik.

Składnia

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

Zasada

LENTOAX(...):	Wstępnie zdefiniowana funkcja do odczytu przyporządkowania długości narzędzia L1, L2 i L3 aktywnego narzędzia do osi współrzędnych		
	W oddzielnym bloku:		tak
<Status>:	Wartość zwrotna funkcji. Wartości ujemne wskazują stany błędu.		
	Typ danych:		INT
	Wartość:	0	Funkcja OK Informacja w <AxInd> wystarcza do opisu (wszystkie składowe długości narzędzia są równoległe do osi geometrycznych).
		1	Funkcja jest OK, dla do poprawnego opisu należy oszacować zawartość <Matrix> (składowe długości narzędzia nie są równoległe do osi geometrycznych).
		-1	Nieważny łańcuch znaków w parametrze <Coord>.
		-2	Brak aktywnego narzędzia.
Parametry			

1	<AxInd>:	Jeżeli składowe długości narzędzia są równoległe do osi geometrycznych, to indeksy osi, które są przyporządkowane do składowych długości L1 do L3, są zwracane w polu **<**AxInd**>**. • **<**AxInd**>** [0]: Odcięta • **<**AxInd**>** [1]: Rzędna • **<**AxInd**>** [2]: Aplikata		
		Typ danych:		INT[3]
		Wartość:	0	Przyporządkowanie nie występuje (brak osi)
			1 ... 3 lub -1 ... -3	Numer długości, która działa w odpowiedniej osi współrzędnych. Znak jest ujemny, gdy składowa długości narzędzia wskazuje ujemny kierunek współrzędnych.
		Jeżeli nie wszystkie składowe długości są równoległe lub przeciwrównoległe do osi geometrycznych, to w **<**AxInd**>** jest zwracany indeks osi, który zawiera większy udział składowych długości komponentu. W tym przypadku (gdy funkcje nie dostarczą błędu z innego powodu) jest zwracana wartość **<**Status**>** = 1. Odwzorowanie składowych długości narzędzia L1 do L3 na osiach geometrycznych 1 do 3 jest następnie w całości opisywane poprzez zawartość 2. parametru **<**Matrix**>**.		
2	<Matrix>:	Macierz, która przedstawia wektor długości narzędzia (L1=1, L2=1, L3=1) w wektorze osi współrzędnych (odcięta, rzędna, aplikata), tzn. do kolumn są przyporządkowane składowe długości narzędzia w kolejności L1, L2, L3, a do wierszy, osie w kolejności odcięta, rzędna, aplikata.		
		Typ danych:		REAL
		W macierzy obowiązują zawsze wszystkie elementy, nawet jeśli oś geometryczna należąca do osi współrzędnych nie istnieje, tzn. gdy odpowiedni wpis w **<**AxInd**>** jest pusty.		
3	<Coord>:	Układ współrzędnych, dla którego obowiązuje przyporządkowanie (opcjonalnie)		
		Typ danych:		STRING
		Znaki:	MCS M	Odwzorowanie długości narzędzia w układzie współrzędnych maszyny
			BCS B	Odwzorowanie długości narzędzia w bazowym układzie współrzędnych
			WCS W	Odwzorowanie długości narzędzia w układzie współrzędnych przedmiotu obrabianego (domyślnie)
			KCS K	Odwzorowanie długości narzędzia w układzie współrzędnych narzędzia transformacji kinematycznej
			TCS T	Odwzorowanie długości narzędzia w układzie współrzędnych narzędzia
		Sposób zapisu znaków w łańcuchu (małe lub duże) jest dowolny. Jeżeli parametr **<**Coord**>** nie jest podany, to stosowany jest WKS (domyślnie).		

Uwaga

W TCS, wszystkie składowe długości narzędzia są zawsze równoległe lub przeciwrównoległe do osi.

Składowe mogą być przeciwrównoległe, gdy aktywne jest odbicie lustrzane i ustawiona jest następująca dana ustawcza:

SD42900 $SC_MIRROR_TOOL_LENGTH (zmiana znaku długości narzędzia przy odbiciu lustrzanym)

Przykład

Standardowy przypadek - narzędzie frezarskie w G17.

L1 działa w Z (aplikata), L2 w Y (rzędna), L3 w X (odcięta).

Wywołanie funkcji w formie:
`<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")`

Parametr wyników <AxInd> zawiera wtedy wartości:

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

lub krócej: (3, 2, 1)

W tym przypadku przynależna macierz wynosi:

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Zmiana G17 na G18 lub G19 nie zmienia wyniku, ponieważ przyporządkowanie składowych komponentów długości do osi geometrycznych zmienia się w taki sam sposób jak przyporządkowanie odciętej, rzędnej i aplikaty.

Przy aktywnym G17 jest teraz programowany obrót frame wokół Z o wartość 60 stopni, np.:
`ROT Z60`

Kierunek aplikaty (kierunek Z) pozostaje niezmieniony, składowa główna L2 leży teraz w kierunku nowej osi X, składowa główna L1 w kierunku osi ujemnej Y. Dlatego wartość zwrotna (<Status>) wynosi „1”, <AxInd> zawiera wartości (2, -3, 1).

W tym przypadku przynależna macierz wynosi:

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

Zachowanie się w ruchu po torze 12

12.1 Sterowanie styczne

12.1.1 Definicja sprzężenia (TANG)

Za pomocą wstępnie zdefiniowanej procedury TANG (...) definiowane jest sprzężenie styczne między osią obrotową (osią nadążną) i dwiema osiami geometrycznymi (osiami wiodącymi). Oś nadążna jest ustawiana przy tym w sposób ciągły stycznie do toru osi wiodącej.

Uwaga

Współczynnik sprzężenia

Współczynnik sprzężenia wynoszący 1 nie musi być bezpośrednio programowany.
Za pomocą współczynnika sprzężenia -1 obracany jest kierunek osi stycznej.

Składnia

```
TANG(<oś nadążna>, <oś wiodąca_1>, <oś wiodąca_2>, <współ.sprz.>,
<układ współrzędnych>, <optymalizacja>)
```

Znaczenie

TANG(...):	Definiowanie sprzężenia stycznego	
<Oś nadążna>:	Nazwa osi nadążnej (oś obrotowa)	
	Typ danych:	AXIS
	Zakres wartości:	Nazwy osi kanału
<Oś wiodąca_1> <Oś wiodąca_2>	Nazwy osi wiodących (osie geometryczne) [1)]	
	Typ danych:	AXIS
	Zakres wartości:	Nazwy osi geometrycznych kanału.
<Współczynnik sprzężenia>:	Współczynnik n zmiany kąta osi nadążnej do zmiany stycznej do toru osi wiodącej: Zmiana kąta$_{\text{oś nadążna}}$ = zmiana kąta$_{\text{styczna do toru}}$ * n	
	Typ danych:	REAL
	Wartość domyślna:	1,0

<Układ współrzędnych>:	Działający układ współrzędnych[2]		
	Typ danych:	CHAR	
	Wartość:	"B":	Bazowy układ współrzędnych (wartość domyślna)
		"W":	Układ współrzędnych obrabianego przedmiotu (niedostępny)
<Optymalizacja>	Rodzaj optymalizacji		
	Typ danych:	CHAR	
	Wartość:	"S":	Standard (wartość domyślna) Dynamika osi obrotowej nie ma żadnego oddziaływania zwrotnego na oś wiodącą. Jeżeli dynamika osi obrotowej jest wyższa niż wymagana dla funkcji śledzenia proces ten jest wystarczająco dokładny. Jeżeli dynamika osi obrotowej nie jest wystarczająco wysoka, aby śledzić zmiany stycznej do toru zorientowanie osi obrotowej wzdłuż niezdefiniowanej drogi wygładzania odbiega od zorientowania zadanego.
		"P":	Dynamika osi obrotowej jest uwzględniana podczas planowania toru osi wiodącej. Przy czym podczas włączania sprzężenia stycznego za pomocą TANGON() należy podać dwa dodatkowe parametry: • Drogę wygładzania • Tolerancję kątową Patrz Rozdział "Włączenie sprzężenia (TANGON) (Strona 526)". **Uwaga** W związku z transformacjami kinematycznymi zalecamy stosowanie rodzaju optymalizacji "P".

Wskazówka

Wartości domyślne nie muszą być szczegółowo programowane.

[1] **Wskazówka**

Jako osie wiodące w sprzężeniu stycznym należy zastosować te osie geometryczne, które w układzie współrzędnym maszyny (MKS) odsuwają zaprogramowany tor w odniesieniu do położenia zasadniczego maszyny. Jeśli np. na frezarce z głowicą skrętną zastosowany zostanie cykl CYCLE800, to w zależności od konfiguracji cyklu następuje interpolacja w WKS np. osiami geometrycznymi X i Y. Sprzężenie styczne musi zostać zdefiniowane za pomocą osi geometrycznych będącymi osiami wiodącymi, które odsuwają zaprogramowany tor w MKS. W tym celu jako osie wiodące należy zastosować osie geometryczne w stanie **nieskręconym** maszyny.

[2] **Wskazówka**

Bazowy układ współrzędnych (BKS) nie może się obracać względem MKS. Jeżeli za pomocą polecenia ROT lub cyklu skrętu CYCLE800 BKS obraca się sterowanie styczne nie pracuje właściwie.

12.1.2 Włączenie tworzenia bloku pośredniego (TLIFT)

Jeżeli w miejscu zaprogramowanego toru osi wiodącej zmiana stycznej osi nadążnej przekracza wartość graniczną zaprogramowaną w danej maszynowej MD37400 $MA_EPS_TLIFT_TANG_STEP, to następuje dalsze planowanie toru w zależności od ustawionego zachowania się na narożach. Bez zastosowania wstępnie zdefiniowanej procedury TLIFT(...) ruch nastąpi odpowiednio do zaprogramowanego za pomocą TANG(...) (Strona 523) i TANGON(...) (Strona 526)zachowania ścinania naroży.

Włączenie tworzenia bloku pośredniego

Za pomocą programowania TLIFT (...) w połączeniu z TANG(...) przed przebiegiem rozpoznania naroża w miejscu toru zostanie wstawiony automatycznie wygenerowany przez sterowanie blok pośredni.

W trakcie wykonywania programu w chwili osiągnięcia bloku pośredniego nastąpi zatrzymanie osi wiodących. W bloku pośrednim oś nadążna obraca się z maksymalną dynamiką w kierunku stycznym do toru następnego bloku. Następnie osie wiodące poruszają się dalej na zaprogramowanym torze.

Wyłączenie tworzenia bloku pośredniego

Do wyłączenia tworzenia bloku pośredniego należy na nowo zdefiniować sprzężenie styczne za pomocą TANG(...), ale bez kolejnego włączenia tworzenia bloku pośredniego za pomocą TLIFT(...).

Składnia

```
TLIFT(<oś nadążna>)
```

Znaczenie

`TLIFT(...)`:	Włączenie rozpoznawania naroży za pomocą obliczenia bloku pośredniego	
`<Oś nadążna>`:	Nazwa osi nadążnej (oś obrotowa)	
	Typ danych:	AXIS
	Zakres wartości:	Nazwy osi kanału

Prędkość obrotowa osi nadążnej

Oś uczestnicząca w tworzeniu konturu

Jeśli oś nadążna przed aktywacją sprzężenia stycznego wykonuje ruch już jako oś uczestnicząca w tworzeniu konturu, ruch obrotowy następują w bloku pośrednim jako oś uczestnicząca w tworzeniu konturu.

Poprzez wprowadzenie promienia odniesienia za pomocą `FGREF[<Achse>]=0.001` ruchy obrotowe odbywają się z maksymalną ustaloną prędkością osi.

MD32000 $MA_MAX_AX_VELO[<oś nadążna>]

Oś pozycjonowania

Jeśli oś nadążna przed aktywacją sprzężenia stycznego nie wykonuje jeszcze ruchu jako oś uczestnicząca w tworzeniu konturu, obrót następuje w bloku pośrednim jako oś pozycjonująca.

Ruchy obrotowe odbywają się z ustaloną prędkością osi pozycjonującej:

MD32060 $MA_POS_AX_VELO[<oś nadążna>]

12.1.3 Włączenie sprzężenia (TANGON)

Poprzez wstępnie zdefiniowaną procedurę TANGON(...) zostanie włączone sprzężenie styczne zdefiniowane wcześniej za pomocą TANG(...) (Strona 523). Oś nadążna przy następnym przejeździe osi wiodącej jest w sposób ciągły orientowana do stycznej do toru.

Kąt osi nadążnej

Kąt, który przyjmuje oś nadążna w odniesieniu do stycznej do toru jest zależny od zadanego w TANG(...) stosunku przełożenia, kąta przesunięcia sparametryzowanego w danej maszynowej MD37402 $MA_TANG_OFFSET oraz zadanego kąta przesunięcia działającego addytywnie przy TANGON(...).

Optymalizacja „P"

Jeżeli podczas definiowania sprzężenia stycznego (TANG(...)) jako parametr optymalizacji podano wartość „P”, przy włączaniu sprzężenia należy podać parametr „droga wygładzania” i opcjonalnie parametr „tolerancja kąta”.

Jeżeli jako tolerancję kąta poda się zadaną wartość 0 zadziała jedynie parametr „droga wygładzania”.

Jeżeli jako tolerancję kąta poda się zadaną wartość większą niż 0 to aktywna droga wygładzania utworzy się z minimum sparametryzowanej drogi wygładzania i drogi wygładzania na podstawie sparametryzowanej tolerancji kąta.

Jeżeli dynamika osi nadążnej nie jest wystarczająca, aby podążać za sparametryzowanymi warunkami, odpowiednio do osi wiodących zredukowana zostanie prędkość toru.

Składnia

```
TANGON(<Oś nadążna>, <Kąt przesunięcia>, <Droga wygładzania>,
<Tolerancja kąta>)
```

Znaczenie

`TANGON(...)`:	Włączenie sprzężenia stycznego	
`<Oś nadążna>`:	Nazwa osi nadążnej (oś obrotowa)	
	Typ danych:	AXIS
	Zakres wartości:	Nazwy osi kanału
`<Kąt przesunięcia>`:	Kąt przesunięcia osi nadążnej do stycznej do toru Punkt odniesienia to punkt zerowy osi obrotowej.	
	Typ danych:	REAL

<Droga wygładzania>:	Dopuszczalna maksymalna wartość drogi wygładzania. Jeżeli droga wygładzania jest większa z powodu warunków dynamicznych, prędkość ruchu po torze osi wiodących zmniejsza się.	
	Typ danych:	REAL
<Tolerancja kątowa>:	Maksymalnie dopuszczalna tolerancja w odniesieniu do zadanego kąta pomiędzy ustawieniem zerowym osi nadążna a styczną do toru.	
	Typ danych:	REAL

12.1.4 Wyłączenie sprzężenia (TANGOF)

Za pomocą wstępnie zdefiniowanej procedury TANGOF(...) zdefiniowane przez TANG(...) (Strona 523) sprzężenie styczne i włączone przez TANGON(...) (Strona 526) zostanie wyłączone. Oś nadążna nie będzie zorientowana na styczną do toru osi wiodącej. Sprzężenie osi nadążnej z osią wiodącą pozostanie bez zmian nawet po wyłączeniu co uniemożliwia działanie następujących funkcji:

- Zmiana płaszczyzny
- Przełączenie osi geometrycznych
- Definiowanie nowego sprzężenia dla osi nadążnej

Całkowite anulowanie połączenia osi nadążnej z osią wiodącą następuje dopiero po usunięciu sprzężenia za pomocą TANGDEL(...) (Strona 527).

Programowanie

```
TANGOF(<oś nadążna>)
```

Znaczenie

TANGOF(...):	Wyłączenie sprzężenia stycznego	
<Oś nadążna>:	Nazwa osi nadążnej (oś obrotowa)	
	Typ danych:	AXIS
	Zakres wartości:	Nazwy osi kanału

12.1.5 Kasowanie sprzężenia (TANGDEL)

Sprzężenie styczne zdefiniowane za pomocą TANG(...) (Strona 523) również pozostaje bez zmian po wyłączeniu sprzężenia stycznego za pomocą TANGOF(...) (Strona 527). Bieżące sprzężenie styczne uniemożliwia w dalszym ciągu działanie następujących funkcji:

- Zmiana płaszczyzny
- Przełączenie osi geometrycznych
- Definiowanie nowego sprzężenia dla osi nadążnej

Za pomocą wstępnie zdefiniowanej procedury TANGDEL(...) po wyłączeniu sprzężenia stycznego za pomocą TANGOF(...) istniejące sprzężenie zostanie skasowane.

Sytax

```
TANGDEL(<oś nadążna>)
```

Znaczenie

TANGDEL(...):	Kasuje zdefiniowane sprzężenie styczne za pomocą TANG().	
	Działanie:	Pojedynczymi blokami
<Oś nadążna>:	Nazwa osi nadążnej, której sprzężenie styczne powinno zostać skasowane.	
	Typ danych:	AXIS
	Zakres wartości:	Nazwy osi kanału

Przykłady

Zmiana osi wiodącej

Zanim będzie można zdefiniować dla osi nadążnej nowe sprzężenie styczne za pomocą osi wiodącej należy najpierw skasować aktualne sprzężenie.

```
Kod programu                    Komentarz
N10 TANG(A, X, Y, 1)            ; Definiowanie sprzężenia stycznego dla osi nadążnej A:
                                A do X i Y
N20 TANGON(A)                   ; Włączenie sprzężenia stycznego dla osi nadążnej A:
N30 X10 Y20
...
N80 TANGOF(A)                   ; Wyłączanie sprzężenia stycznego dla osi nadążnej A:
N90 TANGDEL(A)                  ; Kasowanie sprzężenia stycznego dla osi nadążnej A:
...
N120 TANG(A, X, Z)              ; Definiowanie nowego sprzężenia stycznego dla osi nad-
                                ążnej A:
N130 TANGON(A)                  ; Włączenie nowego sprzężenia stycznego dla osi nadąż-
                                nej A:
...
```

Przełączenie osi geometrycznych

Zanim będzie można przełączyć osie geometryczne dla aktualnego sprzężenia należy je najpierw skasować.

```
Kod programu                    Komentarz
N10 GEOAX(2, Y1)                ; 2. oś geometryczna = oś maszyny Y1
N20 TANG(A, X, Y)               ; Definiowanie sprzężenia stycznego dla osi nadążnej
                                A:
N30 TANGON(A, 90)               ; Włączenie sprzężenia stycznego dla osi nadążnej A:
N40 G2 F8000 X0 Y0 I0 J50       ; Blok ruchu
N50 TANGOF(A)                   ; Wyłączanie sprzężenia stycznego dla osi nadążnej A:
N60 TANGDEL(A)                  ; Kasowanie sprzężenia stycznego dla osi nadążnej A:
N70 GEOAX(2, Y2)                ; 2. oś geometryczna = oś maszyny Y2
```

Kod programu	**Komentarz**
N80 TANG(A, X, Y)	; Definiowanie nowego sprzężenia stycznego dla osi nadążnej A:
N90 TANGON(A, 90)	; Włączenie nowego sprzężenia stycznego dla osi nadążnej A:
...	

12.2 Przebieg posuwu (FNORM, FLIN, FCUB, FPO)

W celu bardziej elastycznego zadawania przebiegu posuwu jego programowanie według DIN 66025 jest rozszerzane o przebiegi liniowe i sześcienne.

Przebiegi sześcienne mogą być programowane bezpośrednio albo jako spline interpolujące. Przez to dają się - zależnie od zakrzywienia obrabianego przedmiotu - programować ciągle gładkie przebiegi prędkości.

Te przebiegi prędkości umożliwiają wolne od szarpnięć zmiany przyśpieszenia, a przez to wykonywanie równomiernych powierzchni obrabianych przedmiotów.

Składnia

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

Znaczenie

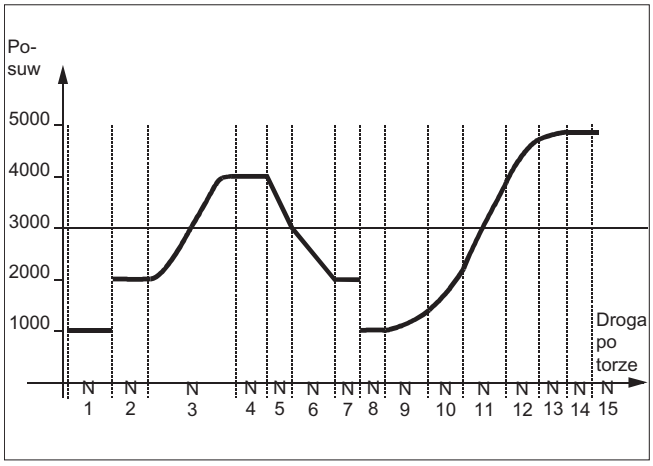
FNORM:	Ustawienie podstawowe. Wartość posuwu jest zadawana przez drogę ruchu po torze w bloku i obowiązuje następnie jako wartość modalna.
FLIN:	Profil prędkości ruchu po torze **liniowy:** Wartość posuwu jest uzyskiwana liniowo z aktualnej wartości na początku bloku do końca bloku na drodze ruchu po torze i następnie obowiązuje jako wartość modalna. To zachowanie się można kombinować z G93 i G94.
FCUB:	Profil prędkości ruchu po torze **sześcienny:** Programowane pojedynczymi blokami wartości F - w odniesieniu do punktu końcowego bloku - są łączone przez spline. Spline rozpoczyna się i kończy stycznie do poprzedzającego wzgl. następnego podania posuwu i działa z G93 i G94. Jeżeli w bloku brak jest adresu F, wówczas jest przyjmowana ostatnio zaprogramowana wartość F.
F=FPO… :	Profil prędkości ruchu po torze **przez wielomian:** Adres F określa przebieg posuwu przez wielomian od aktualnej wartości do końca bloku. Wartość końcowa obowiązuje następnie jako wartość modalna.

Optymalizacja posuwu na zakrzywionych torach ruchu

Wielomian posuwu `F=FPO` i spline posuwu `FCUB` powinny być zawsze wykonywane ze stałą prędkością skrawania `CFC`. Przez to daje się wytworzyć wykazujący stałe przyśpieszenie profil posuwu zadanego.

Przykład: Różne profile posuwu

W tym przykładzie można znaleźć programowanie i przedstawienie graficzne różnych profili posuwu.



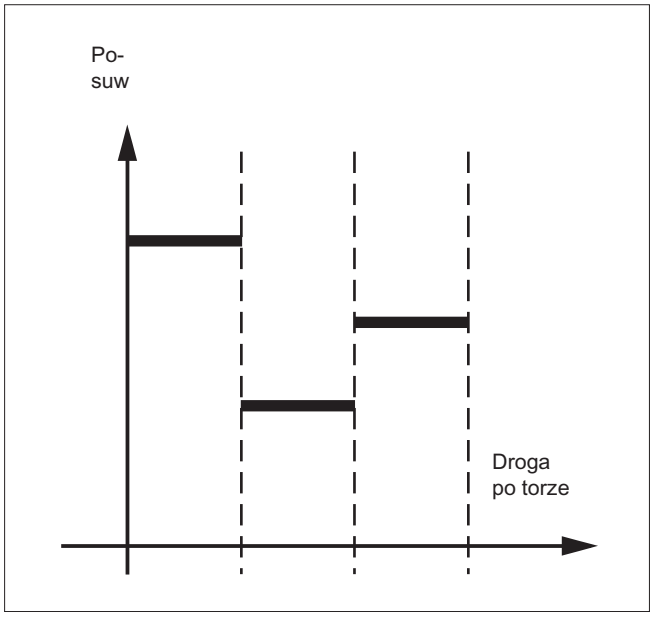
Kod programu	**Komentarz**
N1 F1000 FNORM G1 X8 G91 G64	; Stały profil posuwu, podanie w wymiarze przyrostowym
N2 F2000 X7	; Skokowa zmiana prędkości zadanej
N3 F=FPO(4000, 6000, -4000)	; Profil posuwu przez wielomian z posuwem 4000 na końcu bloku
N4 X6	; Posuw wielomianowy 4000 obowiązuje jako wartość modalna.
N5 F3000 FLIN X5	; Liniowy profil posuwu
N6 F2000 X8	; Liniowy profil posuwu
N7 X5	; Posuw liniowy obowiązuje jako wartość modalna
N8 F1000 FNORM X5	; Stały profil posuwu ze skokową zmianą przyśpieszenia.
N9 F1400 FCUB X8	; Wszystkie dalsze programowane pojedynczymi blokami wartości F są łączone ze spline.
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; Wyłączenie profilu spline
N14 FNORM X5	
N15 X20	

Dodatkowe informacje

FNORM

Adres posuwu F określa posuw ruchu po torze jako wartość stałą według DIN 66025.

Więcej informacji na ten temat można znaleźć w podręczniku programowania "Podstawy".

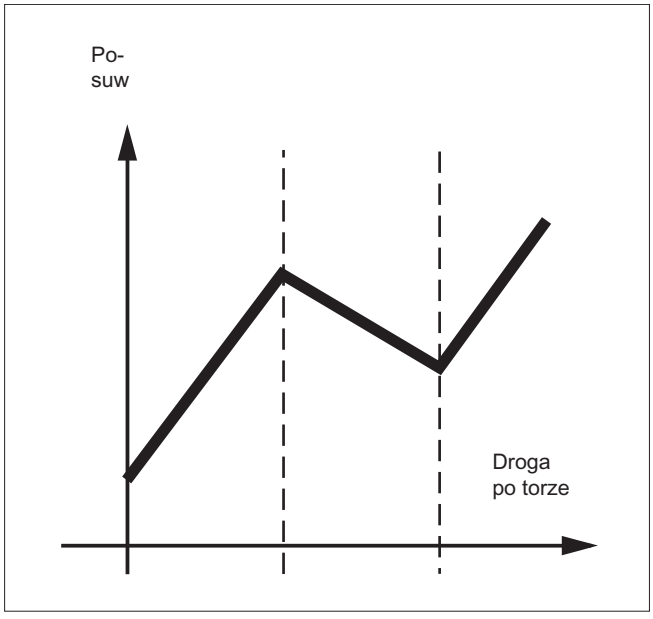


FLIN

Przebieg posuwu jest uzyskiwany od aktualnej wartości posuwu do zaprogramowanej wartości F liniowo do końca bloku.

Przykład:

```
N30 F1400 FLIN X50
```



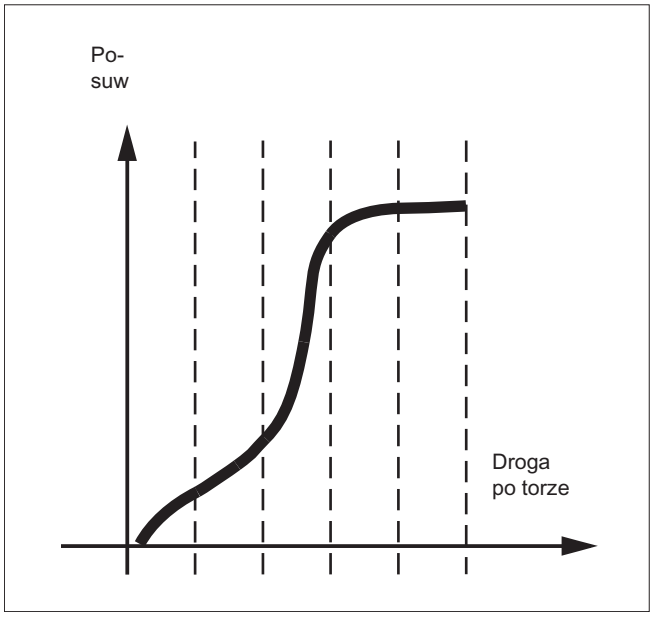
FCUB

Posuw jest uzyskiwany od aktualnej wartości posuwu do zaprogramowanej wartości F do końca bloku w przebiegu sześciennym. Sterowanie łączy przez spline wszystkie

zaprogramowane pojedynczymi blokami wartości posuwu z aktywnym FCUB. Wartości posuwu służą tutaj jako punkty oparcia do obliczenia interpolacji spline.

Przykład:

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```



F=FPO(...,...,...)

Przebieg posuwu jest programowany bezpośrednio przez wielomian. Podanie współczynników wielomianu następuje analogicznie do interpolacji wielomianowej.

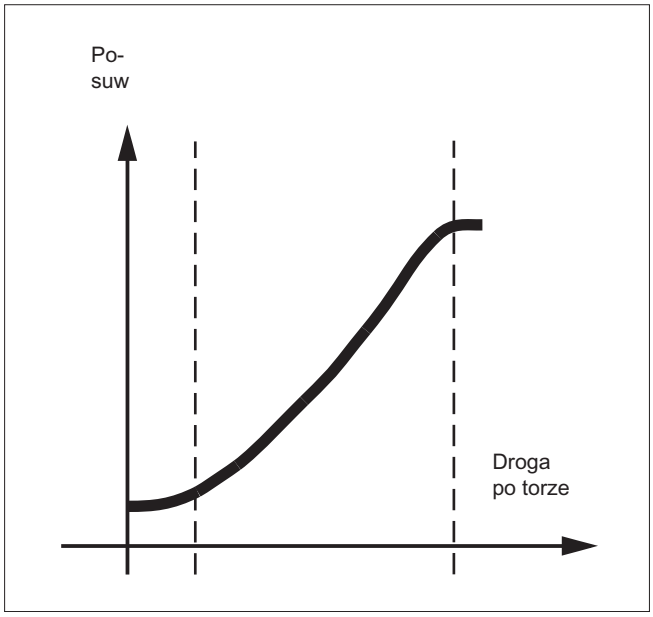
Przykład:

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf i cubf są przedtem zdefiniowanymi zmiennymi.

endfeed:	posuw na końcu bloku
quadf:	kwadratowy współczynnik wielomianowy
cubf:	sześcienny współczynnik wielomianowy

Przy aktywnym `FCUB` spline dołącza na początku i końcu bloku do przebiegu ustalonego przez `FPO`.



Warunki brzegowe

- Niezależnie od zaprogramowanego przebiegu posuwu obowiązują funkcje do programowania zachowania się w ruchu po torze.
- Programowany przebieg posuwu obowiązuje zasadniczo bezwzględnie - niezależnie od `G90` albo `G91`.
- Przebieg posuwu `FLIN` i `FCUB` działa z `G93` i `G94`, **nie** przy `G95`, `G96/G961` i `G97/G971`.
- Przy aktywnym kompresorze `COMPON` obowiązuje przy połączeniu wielu bloków w jeden segment Spline:

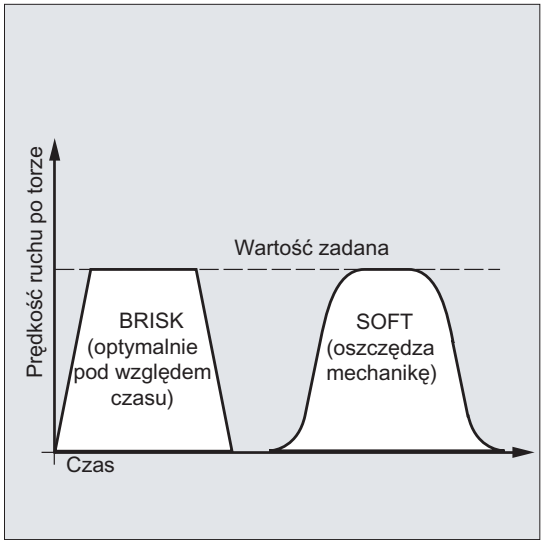
`FNORM`:	Dla segmentu spline obowiązuje słowo F ostatniego przynależnego bloku.
`FLIN`:	Dla segmentu spline obowiązuje słowo F ostatniego przynależnego bloku. Zaprogramowana wartość F obowiązuje do końca segmentu i następnie jest uzyskiwana liniowo.
`FCUB`:	Utworzony Spline posuwu odbiega maksymalnie o wartość zdefiniowaną w danej maszynowej MD20172 $MC_COMPESS_VELO_TOL od zaprogramowanych punktów końcowych.
`F=FPO(…,…,…)`:	Te bloki nie są kompresowane.

12.3 Przyśpieszenie

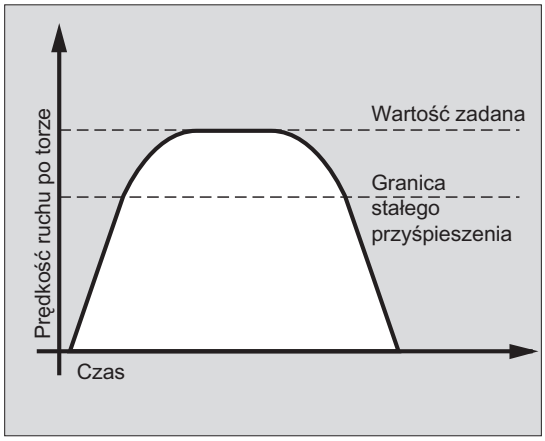
12.3.1 Tryb przyśpieszenia (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Do programowania trybu przyśpieszenia są do dyspozycji następujące polecenia programu obróbki:

- "BRISK, BRISKA"
  Poszczególne osie wzgl. osie uczestniczące w tworzeniu konturu wykonują ruch z max przyspieszeniem, aż do osiągnięcia zaprogramowanej prędkości posuwu (**bez ograniczenia przyspieszenia drugiego stopnia**).
- "SOFT, SOFTA"
  Poszczególne osie wzgl. osie uczestniczące w tworzeniu konturu wykonują ruch ze stałym przyspieszeniem, aż do osiągnięcia zaprogramowanej prędkości posuwu (**z ograniczeniem przyśpieszenia drugiego stopnia**).
- "DRIVE, DRIVEA"
  Poszczególne osie wzgl. osie uczestniczące w tworzeniu konturu wykonują ruch z maksymalnym przyspieszeniem, aż do zaprojektowanej granicy prędkości (ustawienie MD!). Potem następuje zmniejszenie przyspieszenia (ustawienie MD!), aż do osiągnięcia zaprogramowanej prędkości posuwu.



Rysunek 12-1 Przebieg prędkości ruchu po torze przy `BRISK` i `SOFT`



Rysunek 12-2 Przebieg prędkości ruchu po torze przy DRIVE.

Składnia

```
BRISK
BRISKA(<oś1>,<oś2>,…)
SOFT
SOFTA(<oś1>,<oś2>,…)
DRIVE
DRIVEA(<oś1>,<oś2>,…)
```

Znaczenie

BRISK:	Polecenie do włączenia "przyśpieszenia bez ograniczenia przyśpieszenia drugiego stopnia" dla osi uczestniczących w tworzeniu konturu.
BRISKA:	Polecenie do włączenia „przyspieszenia bez ograniczenia przyspieszenia drugiego stopnia" dla ruchów w pojedynczych osiach (JOG, JOG/INC, oś pozycjonowania, oś ruchu wahadłowego, itd.).
SOFT:	Polecenie do włączenia "przyśpieszenia z ograniczeniem przyśpieszenia drugiego stopnia" dla osi uczestniczących w tworzeniu konturu.
SOFTA:	Polecenie do włączenia „przyspieszenia z ograniczeniem przyspieszenia drugiego stopnia" dla ruchów w pojedynczych osiach (JOG, JOG/INC, oś pozycjonowania, oś ruchu wahadłowego, itd.).
DRIVE:	Polecenie do włączenia zmniejszonego przyśpieszenia powyżej zaprojektowanej granicy prędkości (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) dla osi uczestniczących w tworzeniu konturu.
DRIVEA:	Polecenie do włączenia zmniejszonego przyspieszenia powyżej zaprojektowanej granicy prędkości MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) dla ruchów w pojedynczych osiach (JOG, JOG/INC, oś pozycjonowania, oś ruchu wahadłowego, itd.).
(<oś1>,<oś2>,…):	Pojedyncze osie, dla których ma obowiązywać wywołany tryb przyspieszenia.

Warunki brzegowe

Zmiana trybu przyspieszenia podczas obróbki

Gdy w programie obróbki tryb przyspieszenia zostanie zmieniony podczas obróbki (`BRISK` ↔ `SOFT`), wówczas również w trybie przechodzenia płynnego następuje na przejściu zmiana bloku z zatrzymaniem dokładnym na końcu bloku.

Przykłady

Przykład 1: SOFT i BRISKA

```
Kod programu
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
...
```

Przykład 2: DRIVE i DRIVEA

```
Kod programu
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
...
```

Literatura

Podręcznik działania Funkcje podstawowe; Przyśpieszenie (B2)

12.3.2 Sterowanie przyśpieszeniem w przypadku osi nadążnych (VELOLIMA, ACCLIMA, JERKLIMA)

W przypadkach sprzężeń osi (aktualizacja styczna, holowanie, sprzężenie wartości wiodącej, przekładnia elektroniczna; patrz Sprzężenia osi (Strona 573)) osie/wrzeciona nadążne wykonują ruch zależnie od jednej lub wielu osi/wrzecion wiodących.

Na ograniczenia dynamiki osi/wrzecion nadążnych można wpływać przy pomocy funkcji VELOLIMA, ACCLIMA i JERKLIMA z programu obróbki lub z akcji synchronicznych, również przy już aktywnym sprzężeniu osi.

Uwaga

Funkcja JERKLIMA jest dostępna nie dla wszystkich rodzajów sprzężeń.

Literatura:

- Podręcznik działania Funkcje specjalne; Sprzężenia osi (M3)
- Podręcznik działania Funkcje rozszerzone; wrzeciono synchroniczne (S3)

Uwaga

Dostępność w przypadku SINUMERIK 828D

Funkcje VELOLIMA, ACCLIMA i JERKLIMA mogą w przypadku SINUMERIK 828D być stosowane tylko w połączeniu z funkcją „nadążania"!

Składnia

```
VELOLIMA(<oś>)=<wartość>
ACCLIMA(<oś>)=<wartość>
JERKLIMA(<oś>)=<wartość>
```

Znaczenie

VELOLIMA:	Polecenie do korekcji sparametryzowanej **prędkości**maksymalnej
ACCLIMA:	Polecenie do korekcji sparametryzowanego **przyspieszenia**maksymalnego
JERKLIMA:	Polecenie do korekcji sparametryzowanego maksymalnego **przyspieszenia drugiego stopnia**
<oś>:	Oś nadążna, której ograniczenia dynamiki mają zostać skorygowane
<wartość>:	Procentowa wartość korekcji

Przykłady

Przykład 1: Korekcja ograniczeń dynamiki dla osi nadążnej (AX4)

Kod programu	Komentarz
...	
VELOLIMA[AX4]=75	; Korekcja ograniczenia do 75% zapisanej w danej maszynowej prędkości maksymalnej w osi.
ACCLIMA[AX4]=50	; Korekcja ograniczenia do 50% zapisanego w danej maszynowej osiowego przyspieszenia maksymalnego.
JERKLIMA[AX4]=50	; Korekcja ograniczenia do 50% zapisanego w danej maszynowej osiowego maksymalnego przyspieszenia drugiego stopnia.
...	

Przykład 2: Przekładnia elektroniczna

Oś 4 jest przez sprzężenie "przekładnia elektroniczna" sprzężona z osią X. Zdolność przyspieszenia osi nadążnej jest ograniczona do 70% przyspieszenia maksymalnego. Maksymalna dopuszczalna prędkość jest ograniczona do 50% prędkości maksymalnej. Po dokonanym włączeniu sprzężenia maksymalna dopuszczalna prędkość jest ponownie ustawiana na 100%.

Kod programu	Komentarz
...	
N120 ACCLIMA[AX4]=70	; Zmniejszone przyspieszenie maksymalne.
N130 VELOLIMA[AX4]=50	; Zmniejszona prędkość maksymalna.
...	

Kod programu	Komentarz
N150 EGON(AX4,"FINE",X,1,2)	; Włączenie sprzężenia przekładni elektronicznej.
...	
N200 VELOLIMA[AX4]=100	; Pełna maksymalna prędkość.
...	

Przykład 3: Wpływ na sprzężenie wartości wiodącej przez statyczną akcję synchroniczną

Oś 4 jest sprzęgana z osią X przez sprzężenie z wartością wiodącą. Przyspieszenie jest poprzez statyczną akcję synchroniczną 2 od pozycji 100 ograniczone do 80 %.

Kod programu	Komentarz
...	
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	; Akcja synchroniczna
N130 LEADON(AX4, X, 2)	; Sprzężenie wartości wiodącej wł.
...	

12.3.3 Uaktywnienie specyficznych dla technologii wartości dynamiki (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)

Przy pomocy grupy G "technologia" można dla 5 różnych technologicznych kroków obróbki uaktywnić pasującą dynamikę.

Wartości dynamiki i polecenia G dają się projektować, a przez to są zależne od ustawień danych maszynowych (→ producenta maszyny!).

Literatura:
Podręcznik działania Funkcje podstawowe; Tryb przechodzenia płynnego, Zatrzymanie dokładne, LookAhead (B1)

Składnia

Uaktywnienie wartości dynamiki:
```
DYNNORM
DYNPOS
DYNROUGH
DYNSEMIFIN
DYNFINISH
```

Uwaga

Wartości dynamiki działają już w tym bloku, w którym zostało zaprogramowane przynależne polecenie G. Nie następuje zatrzymanie obróbki.

Zapis albo odczyt określonego elementu tablicy:
```
R<m>=$MA...[n,X]
```

```
$MA...[n,X]=<wartość>
```

Znaczenie

DYNNORM:	Polecenie G do uaktywnienia **normalnej dynamiki**	
DYNPOS:	Polecenie G do uaktywnienia dynamiki dla **pozycjonowania**, **gwintowania otworu**	
DYNROUGH:	Polecenie G do uaktywnienia dynamiki dla **obróbki zgrubnej**	
DYNSEMIFIN:	Polecenie G do uaktywnienia dynamiki dla **obróbki wykańczającej**	
DYNFINISH:	Polecenie G do uaktywnienia dynamiki dla **obróbki wykańczającej**	
R<m>:	Parametr obliczeniowy o numerze <m>	
$MA...[n,X]:	Dana maszynowa z elementem tablicy określającym dynamikę	
<n>:	Indeks tablicy	
	Zakres wartości:	0 ... 4
	0	Normalna dynamika (DYNNORM)
	1	Dynamika dla pozycjonowania (DYNPOS)
	2	Dynamika dla obróbki zgrubnej (DYNROUGH)
	3	Dynamika dla obróbki wykańczającej (DYNSEMIFIN)
	4	Dynamika dla obróbki wykańczającej (DYNFINISH)
<X>:	Adres osi	
<wartość>:	Wartość dynamiki	

Przykłady

Przykład 1: Uaktywnienie wartości dynamiki

Kod programu	**Komentarz**
DYNNORM G1 X10	; Położenie podstawowe
DYNPOS G1 X10 Y20 Z30 F…	; Tryb pozycjonowania, gwintowanie otworu
DYNROUGH G1 X10 Y20 Z30 F10000	; Obróbka zgrubna
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; Obróbka wykańczająca
DYNFINISH G1 X10 Y20 Z30 F1000	; Wygładzanie wykańczające

Przykład 2: Zapis albo odczyt określonego elementu tablicy

Przyśpieszenie maksymalne dla obróbki zgrubnej, oś X.

Kod programu	**Komentarz**
R1=$MA_MAX_AX_ACCEL[2,X]	; Odczyt
$MA_MAX_AX_ACCEL[2,X]=5	; Zapis

12.4 Ruch ze sterowaniem wyprzedzającym (FFWON, FFWOF)

Dzięki sterowaniu wyprzedzającemu zależna od prędkości droga wybiegu ruchu po torze jest redukowana do zera. Ruch ze sterowaniem wyprzedzającym umożliwia większą dokładność ruchu po torze, a przez to lepsze wyniki produkcji.

Składnia

```
FFWON
FFWOF
```

Znaczenie

FFWON:	Polecenie do **włączenia** sterowania wyprzedzającego
FFWOF:	Polecenie do **wyłączenia** sterowania wyprzedzającego

Uwaga

Poprzez dane maszynowe ustala się rodzaj sterowania wyprzedzającego i które osie uczestniczące w tworzeniu konturu mają wykonywać ruch z takim sterowaniem.

Standard: Sterowanie wyprzedzające zależne od prędkości

Opcja: Sterowanie wyprzedzające zależne od przyspieszenia

Przykład

Kod programu

```
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

12.5 Programowalna dokładność konturu (CPRECON, CPRECOF)

Funkcja "Programowalna dokładność konturu" zmniejsza błąd toru na konturach zakrzywionych, dzięki automatycznemu dopasowaniu prędkości.

Będącą do utrzymania dokładność konturu zadaje się w zależności od zaprojektowania maszyny (MD20470 $MC_MC_CPREC_WITH_FFW; patrz dane producenta maszyny) albo poprzez daną ustawczą $SC_CONTPREC lub poprzez zaprogramowaną tolerancję konturu CTOL. Im mniejsza jest wartość i im mniejszy współczynnik K_V osi geometrycznych, tym bardziej jest zmniejszany posuw po torze na konturach zakrzywionych.

Funkcja "Programowalna dokładność konturu" jest włączana i wyłączana poprzez instrukcje `CPRECON` i `CPRECOF` w programie NC.

Składnia

```
CPRECON
...
CPRECOF
```

Znaczenie

`CPRECON`:	Wywołanie polecenia G: **włączyć** "Programowalną dokładność konturu"	
	Działanie:	modalnie
`CPRECOF`:	Wywołanie polecenia G: **Wyłączyć** "programowalną dokładność konturu"	
	Działanie:	modalnie

CPRECON i CPRECOF tworzą razem grupę funkcji G nr 39 (programowalna dokładność konturu).

Uwaga

Poprzez daną ustawczą $SC_MINFEED (minimalny posuw po torze przy CPRECON) użytkownik może zadać prędkość minimalną posuwu po torze.

Posuw nie jest ograniczany poniżej tej wartości, chyba że została zaprogramowana niższa wartość F lub dynamiczne ograniczenia osi wymuszają mniejszą prędkość.

Uwaga

Funkcja "Programowalna dokładność konturu" dotyczy tylko osi geometrycznych toru. Na prędkość osi pozycjonowania nie ma ona wpływu.

Przykład

Kod programu	**Komentarz**
`N10 G0 X0 Y0`	
`N20 CPRECON`	`; Włączenie "Programowalnej dokładności konturu".`

Kod programu	**Komentarz**
N30 G1 G64 X100 F10000	; Obróbka z 10 m/min w trybie przechodzenia płynnego.
N40 G3 Y20 J10	; Automatyczne ograniczenie posuwu w bloku ruchu kołowego.
N50 G1 X0	; Posuw ponownie bez ograniczenia (10 m/min).
...	
N100 CPRECOF	; Wyłączenie "Programowalnej dokładności konturu".
N110 G0 ...	

Literatura

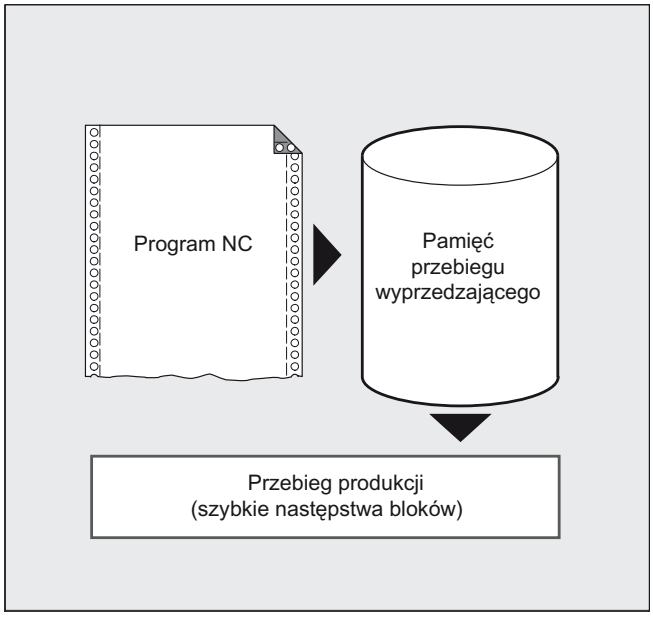
Odnośnie programowania CTOL patrz "Programowanie tolerancji konturu/orientacji (CTOL, OTOL, ATOL) (Strona 566)"

Dla uzyskania bardziej szczegółowych informacji dot. funkcji "Programowalna dokładność konturu" patrz:

Podręcznik działania, Funkcje specjalne; Nadzór tunelu konturu (K6), Rozdział: „Programowana dokładność konturu”

12.6 Przebieg programu z pamięcią przebiegu wyprzedzającego (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

W zależności od stopnia rozbudowy sterowanie dysponuje określoną wielkością tzw. pamięci przebiegu wyprzedzającego, która zapisuje przygotowane bloki przed ich wykonaniem i w trakcie obróbki wyprowadza jako szybkie sekwencje bloków. Dają się przez to wykonywać krótkie drogi z dużymi prędkościami. O ile pozostały czas sterowania to dopuszcza, pamięć przebiegu wyprzedzającego jest z zasady napełniana.



Oznakowanie segmentu obróbki

Segment obróbki, który ma zostać poddany pośredniemu zapisaniu w pamięci przebiegu wyprzedzającego, jest w programie obróbki oznakowywany na początku przez „STOPFIFO” i na końcu przez „STARTFIFO”. Wykonywanie przygotowanych i poddanych pośredniemu zapisaniu bloków rozpoczyna się dopiero po poleceniu „STARTFIFO” albo gdy pamięć przebiegu wyprzedzającego jest zapełniona.

Automatyczne sterowanie pamięcią przebiegu wyprzedzającego

Automatyczne sterowanie pamięcią przebiegu wyprzedzającego jest wywoływane poleceniem "FIFOCTRL". "FIFOCTRL działa najpierw dokładnie tak, jak "STOPFIFO". Przy każdym programowaniu następuje czekanie, aż pamięć przebiegu wyprzedzającego zostanie wypełniona, następnie rozpoczyna się wykonywanie. Różne jest natomiast zachowanie się przy opróżnianiu pamięci przebiegu wyprzedzającego: Za pomocą FIFOCTRL prędkość po torze jest coraz bardziej zmniejszana od poziomu napełnienia 2/3, żeby zapobiec kompletnemu opróżnieniu pamięci i hamowaniu aż do stanu zatrzymania.

Zatrzymanie przebiegu wyprzedzającego

Przygotowywanie i pośrednie zapisywanie bloków jest zatrzymywane, gdy w bloku jest zaprogramowane polecenie „STOPRE”. Kolejny blok jest wykonywany dopiero wtedy, gdy

wszystkie przedtem przygotowane i zapisane bloki są całkowicie wykonane. Poprzedni blok jest zatrzymywany w zatrzymaniu dokładnym (jak G9).

UWAGA
Anulowanie programu Przy włączonej korekcji narzędzia i w przypadku interpolacji Spline nie należy programować „STOPRE”, ponieważ spowoduje to przerwanie przynależnych do siebie ciągów bloków.

Składnia

Tabela 12-1 Oznakowanie segmentu obróbki:

```
STOPFIFO
...
STARTFIFO
```

Tabela 12-2 Automatyczne sterowanie pamięcią przebiegu wyprzedzającego:

```
...
FIFOCTRL
...
```

Tabela 12-3 Zatrzymanie przebiegu wyprzedzającego:

```
...
STOPRE
...
```

Uwaga

Polecenia „STOPFIFO”, „STARTFIFO”, „FIFOCTRL” i „STOPRE” muszą być programowane w oddzielnym bloku.

Znaczenie

STOPFIFO:	„STOPFIFO oznakowuje początek segmentu obróbki, który ma zostać poddany pośredniemu zapisaniu w pamięci przebiegu wyprzedzającego. Przy pomocy „STOPFIFO" wykonywanie jest zatrzymywane i pamięć pośrednia jest napełniana aż: • nastąpi rozpoznanie „STARTFIFO" albo „STOPRE" lub • pamięć przebiegu wyprzedzającego zostanie wypełniona lub • nastąpi dojście do końca programu.
STARTFIFO:	Przy pomocy „STARTFIFO" następuje szybkie wykonywanie segmentu obróbki, równolegle z tym następuje napełnianie pamięci przebiegu wyprzedzającego
FIFOCTRL:	Włączenie automatycznego sterowania pamięcią przebiegu wyprzedzającego
STOPRE:	Zatrzymanie przebiegu wyprzedzającego

Uwaga

Napełnianie pamięci przebiegu wyprzedzającego nie jest wykonywane wzgl. jest przerywane, gdy segment obróbki zawiera polecenia, które wymuszają pracę nie buforowaną (bazowanie do punktu odniesienia, funkcje pomiarowe, ...).

Uwaga

Przy dostępie do danych stanu maszyny ($SA...) sterowanie wytwarza wewnętrzne zatrzymanie przebiegu wyprzedzającego.

Przykład: Zatrzymanie przebiegu wyprzedzającego

Kod programu	**Komentarz**
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; Blok pomiarowy z sonda pomiarową pierwszego wejścia pomiarowego i interpolacją prostoliniową.
N40 STOPRE	; Zatrzymanie przebiegu wyprzedzającego.
...	

12.7 Definiowanie obszarów Stop-Delay (DELAYFSTON, DELAYFSTOF)

W celu zdefiniowania warunkowo przerywanego obszaru (obszar Stopp-Delay) w programie obróbki służą wstępnie zdefiniowane procedury DELAYFSTON i DELAYFSTOF.

Uwaga

DELAYFSTON i DELAYFSTOF **nie** są dozwolone w akcjach synchronicznych!

Składnia

```
DELAYFSTON
...
DELAYFSTOF
```

Znaczenie

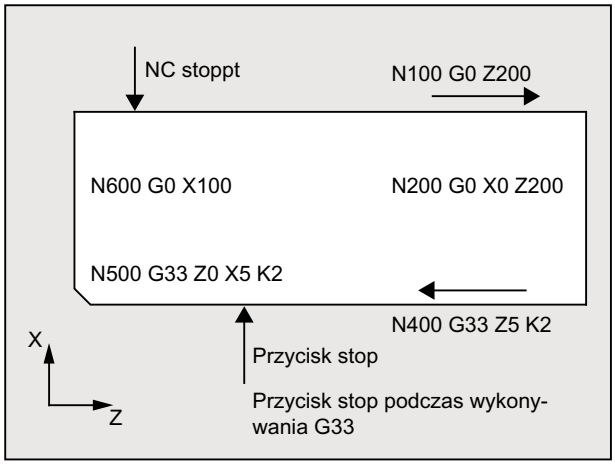
DELAYFSTON:	Definiuje początek obszaru Stop-Delay	
	W oddzielnym bloku:	tak
DELAYFSTOF:	Definiuje koniec obszaru Stop-Delay	
	W oddzielnym bloku:	tak

Przykład programowania

Następujący blok programu jest powtarzany w pętli:

```
Kod programu
...
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
...
```

Na rysunku poniżej widać, że użytkownik w obszarze Stop-Delay naciska "Stop", a NC rozpoczyna proces hamowana poza obszarem Stop-Delay, tzn. w bloku N100. W wyniku tego NC zatrzymuje się w początkowym obszarze N100.



Dodatkowe informacje

Koniec podprogramu

Wraz z końcem podprogramu, w którym wywołano DELAYFSTON następuje domyślne uaktywnienie DELAYFSTOF.

Kaskadowanie

Jeżeli podprogram 1 w obszarze Stop-Delay wywoła podprogram 2, wówczas podprogram 2 mieści się w całości w obszarze Stop-Delay. W szczególności nie działa DELAYFSTOF w podprogramie 2.

Przykład:

Kod programu	Komentarz
N10010 **DELAYFSTON**	; Bloki z N10xxx płaszczyzna programowa 1.
N10020 R1 = R1 + 1	
N10030 G4 F1	; Obszar Stop-Delay rozpoczyna się.
...	
N10040 Podprogram2	
...	
...	; Interpretacja podprogramu 2.
N20010 **DELAYFSTON**	; Nie działa, powtórne rozpoczęcie, 2. płaszczyzna.
...	
N20020 **DELAYFSTOF**	; Nie działa, koniec na innym poziomie.
N20030 RET	
N10050 **DELAYFSTOF**	; Koniec obszaru Stop-Delay w tej samej płaszczyźnie.
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; Obszar Stop-Delay kończy się. Zatrzymania działają od teraz w sposób bezpośredni.

Zmienne systemowe

To, czy proces obróbki programów aktualnie znajduje się w obszarze Stop-Delay, można sprawdzić za pomocą następujących zmiennych systemowych:

- w programie obróbki za pomocą $P_DELAYFST
- w akcjach synchronicznych za pomocą $AC_DELAYFST

Wartość	Znaczenie
0	Obszar Delay-Stop nie jest aktywny
1	Obszar Delay-Stop jest aktywny

12.8 Zablokowanie miejsca w programie dla SERUPRO (IPTRLOCK, IPTRUNLOCK)

Dla określonych skomplikowanych sytuacji mechanicznych na maszynie jest konieczne uniemożliwienie szukania bloku SERUPRO.

Za pomocą programowanego wskaźnika przerwania jest możliwość ingerencji, by przy "szukaniu miejsca przerwania" rozpocząć obróbkę przed miejscem nienadającym się do przeszukiwania.

Jest możliwość również definiowania obszarów niezdatnych do przeszukiwania w obszarach programu obróbki, w których NC jeszcze nie może ponownie rozpocząć obróbki. Wraz z przerwaniem programu NC zapamiętuje ostatnio wykonany blok, którego można szukać poprzez interfejs obsługi HMI.

Składnia

```
IPTRLOCK
IPTRUNLOCK
```

Polecenia znajdują się w oddzielnych wierszach programu obróbki i umożliwiają programowany wskaźnik przerwania

Znaczenie

IPTRLOCK:	Początek segmentu programu niezdatnego do przeszukiwania
IPTRUNLOCK:	Koniec segmentu programu niezdatnego do przeszukiwania

Obydwa polecenia są dopuszczalne tylko w programach obróbki, ale **nie** w akcjach synchronicznych.

Przykład

Kaskadowanie niezdatnych do przeszukiwania fragmentów programu na dwóch poziomach programowych za pomocą „IPTRUNLOCK”. Implicite „IPTRUNLOCK” w podprogramie 1 kończy obszar niezdatny do przeszukiwania.

Kod programu	**Komentarz**
N10010 IPTRLOCK()	
N10020 R1 = R1 + 1	
N10030 G4 F1	; Blok zatrzymania, który rozpoczyna niezdatny do przeszukiwania segment programu.
...	
N10040 Podprogram2	
...	; Interpretacja podprogramu 2.
N20010 IPTRLOCK ()	; Nie działa, powtórne rozpoczęcie.
...	
N20020 IPTRUNLOCK ()	; Nie działa, koniec na innym poziomie.
N20030 RET	
...	
N10060 R2 = R2 + 2	

Kod programu	Komentarz
N10070 RET	; Koniec segmentu programu niezdatnego do przeszukiwania.
N100 G4 F2	; Program główny jest kontynuowany.

Przerwanie na 100 daje wówczas znów wskaźnik przerwania.

Dodatkowe informacje

Ujęcie i szukanie obszarów niezdatnych do przeszukiwania

Fragmenty programu niezdatne do przeszukiwania programu są zaznaczane przy pomocy poleceń językowych „IPTRLOCK” i „IPTRUNLOCK”.

Polecenie „IPTRLOCK” zamraża wskaźnik przerwania na pojedynczym bloku wykonywalnym w przebiegu głównym (SBL1). Ten blok jest dalej nazywany blokiem zatrzymania. Gdy po „IPTRLOCK” nastąpi anulowanie programu, wówczas na interfejsie graficznym HMI można szukać tego tzw. bloku zatrzymania.

Ponowne ustawienie na aktualnym bloku

Wskaźnik przerwania jest przy pomocy „IPTRUNLOCK” dla następnego segmentu programu ustawiany na aktualnym bloku w punkcie przerwania.

Po znalezieniu celu szukania można z tym samym blokiem zatrzymania powtórzyć nowy cel szukania.

Edytowany przez użytkownika wskaźnik przerwania, musi zostać ponownie usunięty poprzez HMI.

Zasady przy kaskadowaniu

Następujące punkty regulują współpracę poleceń językowych „IPTRLOCK” i „IPTRUNLOCK” z kaskadowaniem i końcem podprogramu:

1. Z końcem podprogramu, w którym wywołano „IPTRLOCK” następuje implicite uaktywnienie „IPTRUNLOCK”.
2. „IPTRLOCK” w obszarze niezdatnym do przeszukiwania pozostaje bez działania.
3. Jeżeli podprogram1 w obszarze niezdolnym do przeszukiwania wywoła podprogram2, wówczas cały podprogram2 pozostaje niezdatny do przeszukiwania. W szczególności nie działa „IPTRUNLOCK” w podprogramie2.

Dalsze informacje na ten temat patrz
/FB1/ Podręcznik działania Funkcje podstawowe; BAG, Kanał, praca programowa (K1).

Zmienna systemowa

Obszar niezdatny do przeszukiwania można rozpoznać przy pomocy „$S_IPRTLOCK” w programie obróbki.

Automatyczny wskaźnik przerwania

Funkcja automatycznego wskaźnika przerwania automatycznie ustala przedtem ustalony rodzaj sprzężenia jako niezdatny do przeszukiwania. Przy pomocy danej maszynowej jest dla

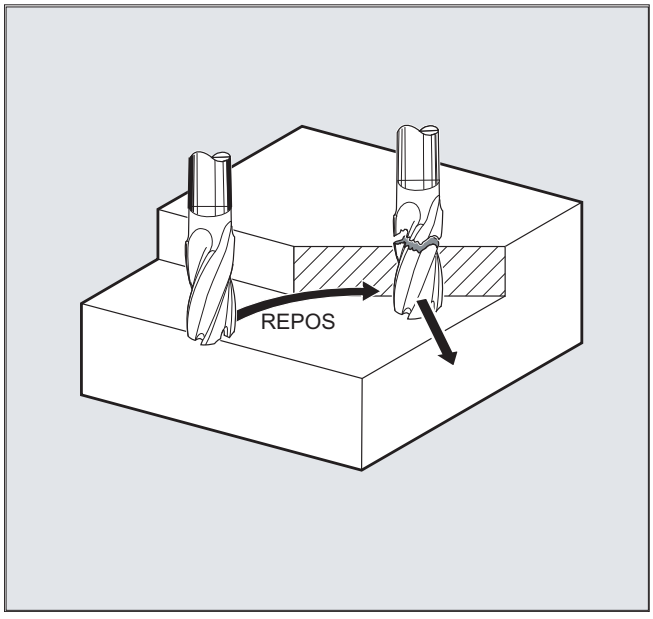
- Przekładni elektronicznej przy „EGON”
- Osiowego sprzężenia wartości wiodącej przy „LEADON”

uaktywniany automatyczny wskaźnik przerwania. Jeżeli zaprogramowane i uaktywniane poprzez daną maszynową automatyczne wskaźniki przerwania zachodzą na siebie, wówczas jest tworzony największy możliwy obszar niezdatny do przeszukiwania.

12.9 Dosunięcie przywracające do konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Gdy podczas obróbki nastąpi przerwanie bieżącego programu i odsunięcie narzędzia - na przykład z powodu jego pęknięcia albo w celu przeprowadzenia pomiaru - można dokonać ponownego sterowanego wg programu dosunięcia narzędzia do konturu w wybranym punkcie.

Polecenie `REPOS` działa w ASUP jak powrót z podprogramu (np. M17). Kolejne bloki nie są już wykonywane. Na temat przerwania przebiegu programu patrz też "Procedura przerwania (ASUP) (Strona 132)".



Składnia

```
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=…
REPOSQ RMBBL DISR=…
REPOSH RMEBL DISR=…
REPOSHA DISR=…
```

Znaczenie

Wybór drogi dosunięcia

REPOSA:	Dosunięcie przywracające do konturu za pomocą osi geometrycznych po prostej. Wszystkie pozostałe osie kanałowe będą także ponownie pozycjonowane.
REPOSL:	Dosunięcie przywracające do konturu za pomocą osi geometrycznych po prostej. Wszystkie inne osie muszą być bezpośrednio programowane.
REPOSQ DISR=… :	Dosunięcie przywracające do konturu za pomocą osi geometrycznych po ćwierćokręgu za pomocą promienia DISR. Wszystkie inne osie muszą być bezpośrednio programowane.
REPOSQA DISR=… :	Dosunięcie przywracające do konturu za pomocą osi geometrycznych po ćwierćokręgu za pomocą promienia DISR. Wszystkie pozostałe osie kanałowe będą także ponownie pozycjonowane.
REPOSH DISR=… :	Dosunięcie przywracające do konturu za pomocą osi geometrycznych po półokręgu za pomocą średnicy DISR. Wszystkie inne osie muszą być bezpośrednio programowane.
REPOSHA DISR=… :	Dosunięcie przywracające do konturu za pomocą osi geometrycznych po półokręgu za pomocą promienia DISR. Wszystkie pozostałe osie kanałowe będą także ponownie pozycjonowane.

Wybór punktu ponownego dosunięcia

RMIBL:	Dosunięcie do punktu przerwania
RMIBL DISPR=…:	Punkt wejścia w odstępie DISPR w mm/calach przed punktem przerwania
RMBBL:	Dosunięcie do punktu początkowego bloku
RMEBL:	Dosunięcie do punktu końcowego bloku
RMEBL DISPR=… :	Dosunięcie do punktu końcowego bloku w odstępie DISPR przed punktem końcowym
RMNBL:	Dosunięcie do najbliżej położonego punktu toru
A0 B0 C0 :	Osie, w których ma nastąpić dosunięcie

Uwaga

Kompatybilność

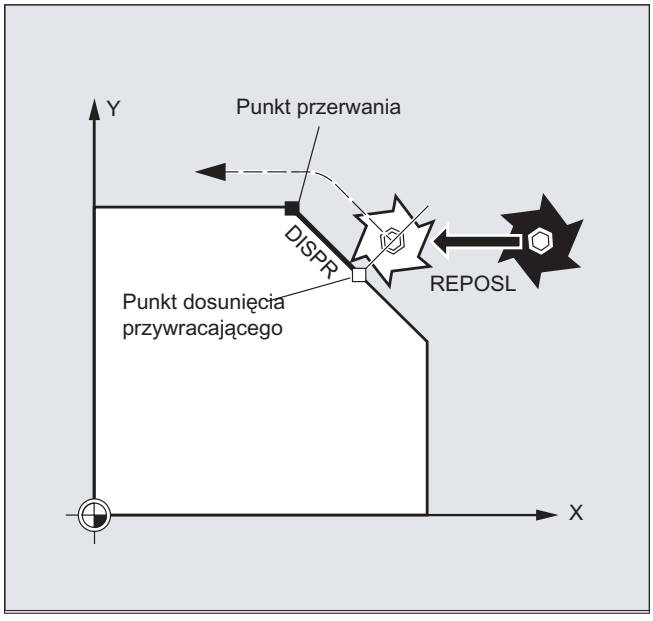
Aby utrzymać kompatybilność ze starszymi wersjami oprogramowania, można zainstalować tryb dosunięcia REPOS, a także nadal programować poprzez modalne polecenia RMI, RMB, RME i RMN. Przy zastosowaniu w ramach ASUP powinien zostać wyposażony atrybut SAVE w instrukcji PROC. W przeciwnym przypadku działa stosowany w ASUP modalny tryb dosunięcia REPOS, gdy odbiega on od ustawienia domyślnego RMI, również w przypadku kolejnych procesów REPOS.

Dosunięcie przywracające do konturu po prostej, REPOSA, REPOSL

Narzędzie wykonuje ruch do punktu dosunięcia przywracającego bezpośrednio po prostej.

Przykład

```
REPOSL RMIBL DISPR=6 F400
```

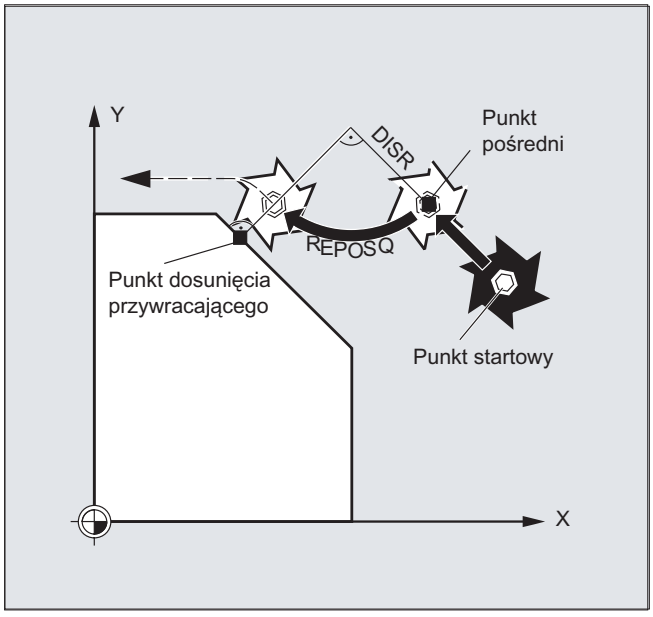


Dosunięcie przywracające do konturu po ćwierćokręgu, REPOSQ, REPOSQA

Narzędzie wykonuje ruch do punktu dosunięcia przywracającego po ćwierćokręgu o promieniu `DISR=...`. Niezbędny punkt pośredni między punktem startowym i punktem dosunięcia przywracającego sterowanie oblicza automatycznie.

Przykład

```
REPOSQ RMIBL DISR=10 F400
```

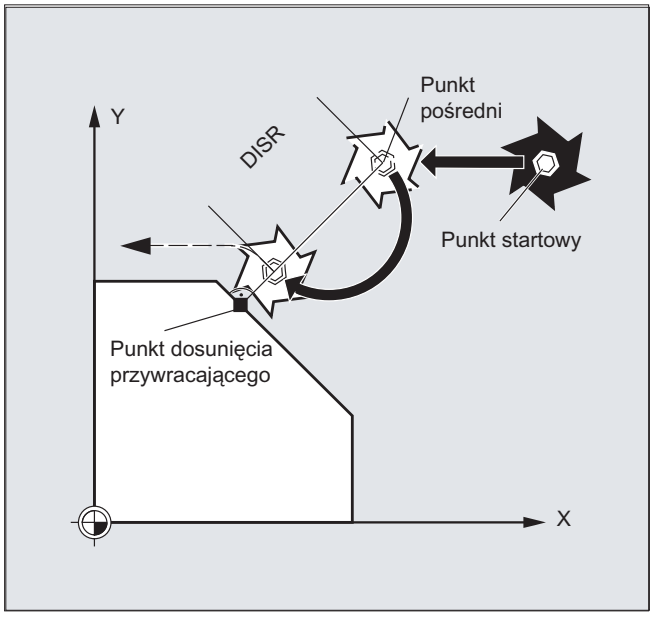


Dosunięcie przywracające do konturu po półokręgu, REPOSH, REPOSHA

Narzędzie wykonuje ruch do punktu dosunięcia przywracającego po półokręgu o średnicy `DISR=...`. Niezbędny punkt pośredni między punktem startowym i punktem dosunięcia przywracającego sterowanie oblicza automatycznie.

Przykład

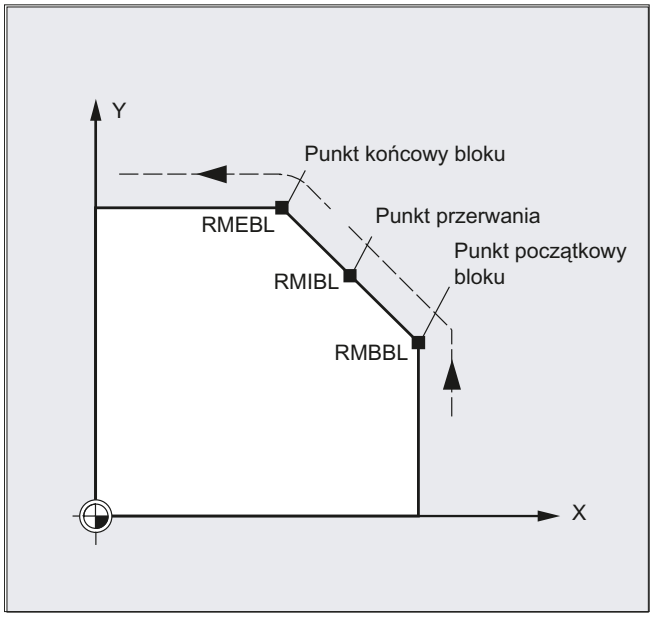
```
REPOSH RMIBL DISR=20 F400
```



Ustalenie punktu dosunięcia przywracającego (nie dla dosunięcia SERUPRO przy pomocy RMNBL)

W odniesieniu do bloku NC, w którym przebieg programu został przerwany, możecie wybierać między trzema punktami dosunięcia przywracającego:

- RMIBL, punkt przerwania
- RMBBL, punkt początkowy bloku wzgl. ostatni punkt końcowy
- RMEBL, punkt końcowy bloku



Przy pomocy RMIBL DISPR=… wzgl. RMEBL DISPR=… można ustalić punkt dosunięcia przywracającego, który leży przed punktem przerwania wzgl. przed punktem końcowym bloku.

Przy pomocy DISPR=… opisujemy drogę po konturze w mm/calach, o którą punkt dosunięcia przywracającego leży przed punktem przerwania wzgl. punktem końcowym. Punkt ten może leżeć maksymalnie w punkcie początkowym bloku - również dla większych wartości.

Jeżeli DISPR=… nie zostanie zaprogramowane, obowiązuje DISPR=0, a przez to punkt przerwania (w przypadku RMIBL) wzgl. punkt końcowy bloku (w przypadku RMEBL).

Znak DISPR

Znak DISPR podlega ewaluacji. W przypadku znaku dodatniego zachowanie jest jak dotychczas.

W przypadku znaku ujemnego dosunięcie następuje za punktem przerwania wzgl. w przypadku RMBBL za punktem startowym.

Odstęp punkt przerwania - punkt rozpoczęcia wynika z wartości bezwzględnej DISPR. Ten punkt może również dla większych wartości absolutnych leżeć maksymalnie w punkcie końcowym bloku.

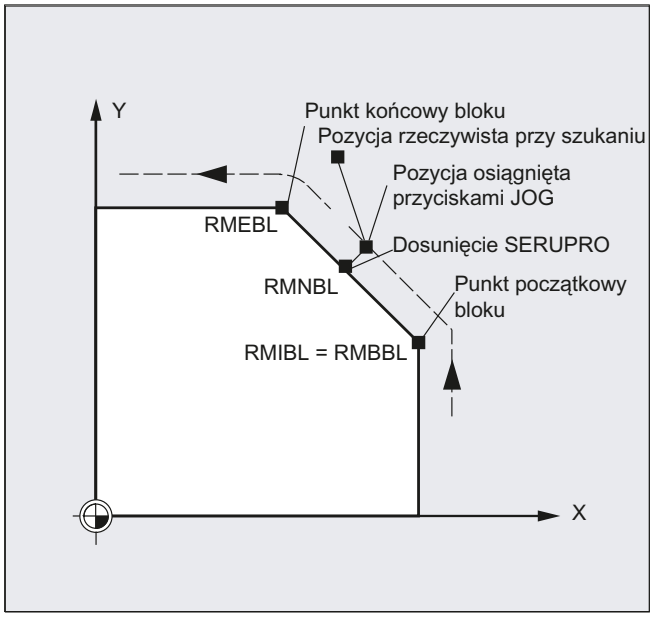
Przykład zastosowania:

Poprzez czujnik zostaje rozpoznane zbliżenie się do łapy dociskowej. Zostaje wyzwolone ASUP, przy pomocy którego łapa dociskowa jest omijana.

Następnie z ujemnym DISPR następuje repozycjonowanie na punkt za łapą i program jest kontynuowany.

Dosuw SERUPRO przy pomocy RMNBL

Jeżeli przy obróbce zostanie w dowolnym miejscu wymuszone przerwanie, wówczas następuje przy pomocy dosunięcia SERUPRO pod RMNBL dosunięcie po najkrótszej drodze od miejsca przerwania, aby następnie przeprowadzić obróbkę tylko po pozostałej drodze. W tym celu użytkownik uruchamia proces SERUPRO do bloku przerwania i przyciskami JOG pozycjonuje przed wadliwym miejscem bloku docelowego.



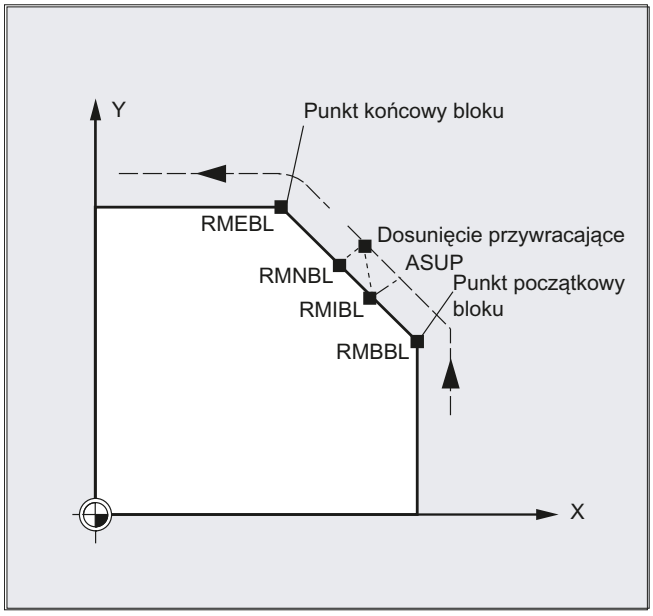
Uwaga

SERUPRO

Dla SERUPRO są identyczne RMIBL i RMBBL. RMNBL nie jest ograniczone tylko do SERUPRO lecz obowiązuje ogólnie.

Dosunięcie do najbliżej położonego punktu toru RMNBL

W momencie interpretacji przez REPOSA po przerwaniu blok dosunięcia z RMNBL nie jest jeszcze raz rozpoczynany od początku, lecz jest wykonywana tylko pozostała droga. Następuje dosunięcie do najbliżej położonego punktu na torze w przerwanym bloku.



Status dla obowiązującego trybu REPOS

Obowiązujący tryb REPOS przerwanego bloku można odczytać poprzez akcje synchroniczne przy pomocy zmiennej $AC_REPOS_PATH_MODE:

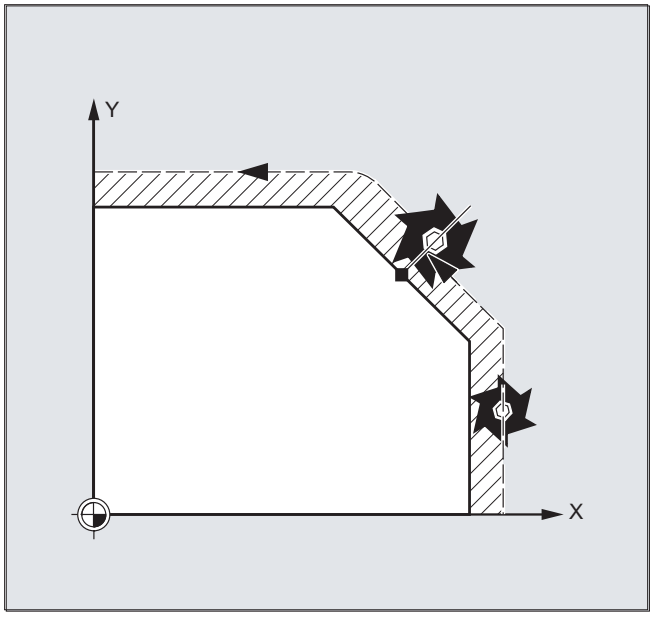
0 Dosunięcie nie zdefiniowane
1 RMBBL: Dosunięcie do początku
2 RMIBL: Dosunięcie do punktu przerwania
3 RMEBL: Dosunięcie do punktu końcowego bloku
4 RMNBL: Dosunięcie do najbliżej położonego punktu toru przerwanego bloku.

Dosunięcie z nowym narzędziem

W przypadku gdy nastąpiło zatrzymanie przebiegu programu z powodu pęknięcia narzędzia:

Z zaprogramowaniem nowego numeru D program jest kontynuowany od punktu dosunięcia przywracającego ze zmienionymi wartościami korekcji narzędzia.

Przy zmienionych wartościach korekcji narzędzia możliwe jest, że nie będzie już można dokonać ponownego dosunięcia do punktu przerwania. W tym przypadku dosunięcie następuje do punktu położonego najbliżej punktu przerwania na nowym konturze (ewentualnie zmodyfikowanego o DISPR).



Dosunięcie do konturu

Ruch, którym narzędzie ponownie zbliża się do konturu, daje się programować. Adresy osi ruchu podajemy z wartością zero.

Przy pomocy poleceń REPOSA, REPOSQA i REPOSHA są automatycznie repozycjonowane wszystkie osie. Podanie osi nie jest konieczne.

Przy zaprogramowaniu REPOSL, REPOSQ i REPOSH wszystkie osie geometryczne wykonują ruch automatycznie, również bez podania w poleceniu. Wszystkie inne osie muszą zostać podane w poleceniu.

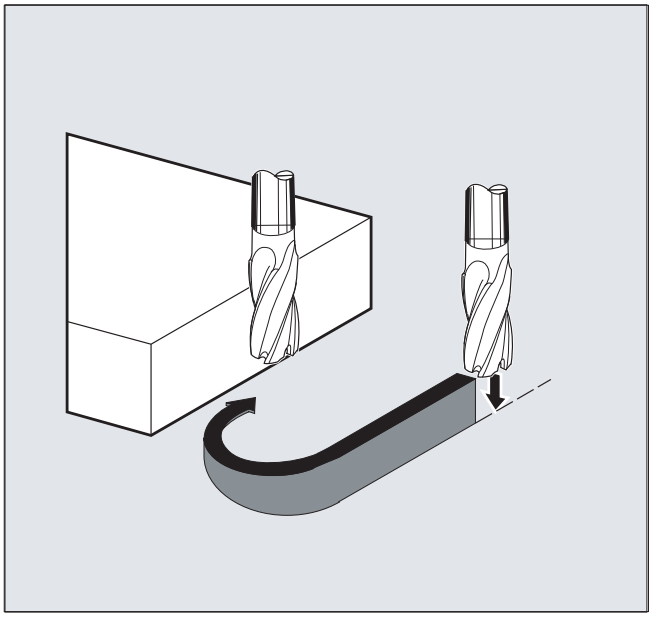
Dla ruchów kołowych REPOSH i REPOSQ obowiązuje:

Ruch po okręgu jest wykonywany w podanej płaszczyźnie obróbki G17 do G19.

W przypadku gdy w bloku dosuwu zostanie podana trzecią oś geometryczna (kierunek dosuwu), ruch do punktu dosunięcia przywracającego na wypadek, że pozycja narzędzia i zaprogramowana pozycja w kierunku dosuwu nie są zgodne, jest wykonywany po linii śrubowej.

W następujących przypadkach następuje automatyczne przełączenie na dosunięcie liniowe REPOSL:

- Nie podano wartości DISR.
- Nie ma zdefiniowanego kierunku dosunięcia (przerwanie programu w bloku bez informacji o ruchu).
- W przypadku kierunku dosunięcia prostopadle do aktualnej płaszczyzny roboczej.



12.10 Sterowanie prowadzeniem ruchu

12.10.1 Procentowa korekcja przyśpieszenia drugiego stopnia (JERKLIM)

Przy pomocy polecenia NC „JERKLIM” można zmniejszyć albo zwiększyć ustawione poprzez daną maszynową maksymalnie możliwe przyspieszenie drugiego stopnia osi przy ruchu po torze w krytycznych fragmentach programu.

Warunek

Tryb przyspieszenia SOFT musi być aktywny.

Działanie

Funkcja działa:

- w trybach pracy AUTOMATYKA.
- tylko na osie uczestniczące w tworzeniu konturu.

Składnia

```
JERKLIM[<oś>]=<wartość>
```

Znaczenie

JERKLIM:	Polecenie do korekcji przyspieszenia drugiego stopnia	
<oś>:	Oś maszyny, której wartość graniczna przyśpieszenia drugiego stopnia ma zostać dopasowana.	
<wartość>:	Procentowa wartość korekcji, odniesiona do zaprojektowanego maksymalnego przyspieszenia drugiego stopnia w osi przy ruchu po torze (MD32431 $MA_MAX_AX_JERK).	
	Zakres wartości:	1 ... 200
	Wartość 100 powoduje brak sterowania przyśpieszeniem drugiego stopnia.	

Uwaga

Zachowanie się JERKLIM na końcu programu obróbki i przy zresetowaniu kanału jest projektowane przy pomocy bitu 0 w danej maszynowej MD32320 $MA_DYN_LIMIT_RESET_MASK:

- Bit 0 = 0:
  Zaprogramowana wartość JERKLIM jest przez zresetowanie kanału / M30 cofana do 100%.
- Bit 0 = 1:
  Zaprogramowana wartość JERKLIM pozostaje zachowana po zresetowaniu kanału / M30.

Przykład

Kod programu	Komentarz
...	
N60 JERKLIM[X]=75	; Sanie osi w kierunku X mogą przyspieszać/zwalniać tylko z max 75% dopuszczalnego dla osi przyspieszenia drugiego stopnia.
...	

12.10.2 Procentowa korekcja prędkości (VELOLIM)

Przy pomocy polecenia VELOLIM można w programie obróbki albo akcji synchronicznej zmniejszyć ustawioną poprzez daną maszynową maksymalnie możliwą szybkość osi wzgl. maksymalnie możliwą, zależną od stopnia przekładni, prędkość obrotową wrzeciona.

Działanie

Funkcja działa:

- w trybach pracy AUTOMATYKA.
- na osie uczestniczące w tworzeniu konturu i osie pozycjonowania.
- na wrzeciona w trybie wrzeciona/osi

Składnia

```
VELOLIM[<oś/wrzeciono>]=<wartość>
```

Znaczenie

VELOLIM:	Polecenie do korekcji prędkości
<oś/wrzeciono>:	Oś albo wrzeciono, którego wartość graniczna prędkości albo prędkości obrotowej ma zostać dopasowana. **VELOLIM dla wrzecion** Poprzez daną maszynowa (MD30455 $MA_MISC_FUNCTION_MASK, Bit 6) można dla programowania ustawić w programie obróbki, czy „VELOLIM" działa niezależnie od aktualnego zastosowania jako wrzeciono lub oś (bit 6 = 1), czy ma być programowalne oddzielnie dla każdego trybu pracy (bit 6 = 0). Jeżeli jest zaprojektowane oddzielne działanie, wówczas wybór jest dokonywany przez identyfikator przy programowaniu: • Identyfikator S<n> trybów pracy wrzeciona • Identyfikator osi, np. "C", dla trybu pracy jako oś

<wartość>:	Procentowa wartość korekcji Wartość korekcji odnosi się: • w przypadku osi / wrzecion w trybie pracy jako oś (gdy MD30455 bit 6 == 0): do zaprojektowanej maksymalnej prędkości osi (MD32000 $MA_MAX_AX_VELO). • w przypadku wrzecion w trybie pracy jako wrzeciono albo oś (gdy MD30455 bit 6 == 1): do maksymalnej prędkości obrotowej aktywnego stopnia przekładni (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])	
	Zakres wartości:	1 ... 100
	Wartość 100 nie powoduje wpływu na prędkość wzgl. prędkość obrotową.	

Uwaga

Zachowanie się na końcu programu obróbki i zresetowaniu kanału

Zachowanie się „VELOLIM” na końcu programu obróbki i przy zresetowaniu kanału jest ustawiane poprzez daną maszynową: MD32320 $MA_DYN_LIMIT_RESET_MASK, bit 0

Rozpoznanie aktywnego ograniczenia prędkości obrotowej w trybie pracy jako wrzeciono

Ograniczenie prędkości obrotowej przez „VELOLIM” (mniejsze niż 100%) można w trybie pracy jako wrzeciono rozpoznać poprzez następującą zmienną systemową:

- $AC_SMAXVELO (maksymalna możliwa prędkość obrotowa wrzeciona)
- $AC_SMAXVELO_INFO (oznaczenie przyczyny ograniczenia prędkości obrotowej)

Przykłady

Przykład 1: Ograniczenie prędkości osi maszyny

Kod programu	Komentarz
...	
N70 VELOLIM[X]=80	; Ruch sań osi w kierunku X powinien być wykonywany tylko z max 80% prędkości dopuszczalnej dla osi.
...	

Przykład 2: Ograniczenie prędkości obrotowej wrzeciona

Kod programu	Komentarz
N05 VELOLIM[S1]=90	; Ograniczenie maksymalnej prędkości obrotowej wrzeciona 1 do 90% z 1000 obr/min.
...	
N50 VELOLIM[C]=45	; Ograniczenie prędkości obrotowej do 45% z 1000 obr/min, niech C będzie identyfikatorem osi S1.
...	

Ustawienia danych maszynowych dla wrzeciona 1 (AX5)

- Maksymalna prędkość obrotowa stopnia przekładni 1 = 1000 obr/min:
  MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- Zaprogramowanie „VELOLIM” działa wspólnie dla pracy wrzeciona i osi niezależnie od zaprogramowanego identyfikatora:
  MD30455 $MA_MISC_FUNCTION_MASK[AX5], Bit 6 = 1

12.10.3 Przykład programu dla JERKLIM i VELOLIM

Poniższy program przedstawia przykład zastosowania procentowego ograniczenia przyspieszenia drugiego stopnia i prędkości:

Kod programu	**Komentarz**
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20	
ACC[Y]=30	
N1200 G1 Y20 VELOLIM[X]=5	; Ruch sań osi w kierunku X powinien być wykonywany tylko z max 5% prędkości dopuszczalnej dla osi.
JERKLIM[Y]=200	; Sanie osi w kierunku Y mogą przyspieszać/zwalniać z max 200% dopuszczalnego dla osi przyspieszenia drugiego stopnia.
N1300 G1 X0 JERKLIM[X]=2	; Sanie osi w kierunku X mogą przyspieszać/zwalniać tylko z max 2% dopuszczalnego dla osi przyspieszenia drugiego stopnia.
N1400 G1 Y0	
M30	

12.11 Programowanie tolerancji konturu/orientacji (CTOL, OTOL, ATOL)

Za pomocą adresów CTOL, OTOL i ATOL można, poprzez dane maszynowe i ustawcze, zaprogramowane tolerancje obróbki dopasować do funkcji kompresora, ścinania narożników i wygładzania orientacji w programie obróbki części.

Zaprogramowane wartości tolerancji obowiązują, aż zostaną na nowo zaprogramowane albo skasowane przez przyporządkowanie ujemnej wartości. Ponadto są one kasowane na końcu programu lub przy resecie. Po skasowaniu działają ponownie sparametryzowane wartości tolerancji.

Składnia

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

Znaczenie

CTOL:	Adres do programowania **tolerancji konturu**			
	Obszar zastosowania:	• wszystkie funkcje kompresora • wszystkie rodzaje ścinania naroży poza G641 i G644		
	Zatrzymanie przebiegu wyprzedzającego:	nie		
	Działanie:	modalnie		
	<Value>:	Wartością tolerancji konturu jest podana długość.		
		Typ:	REAL	
		Jednostka:	cal/mm (zależnie od aktualnego ustawienia podawania wymiarów)	
		Zakres wartości:	≥ 0:	Wartość tolerancji
			< 0:	Kasuje zaprogramowaną wartość tolerancji ⇒ Ponownie obowiązuje wartość tolerancji sparametryzowana w danej maszynowej lub ustawczej.

OTOL:	Adres do programowania **tolerancji orientacji**			
	Obszar zastosowania:	• wszystkie funkcje kompresora • wygładzanie orientacji ORISON • wszystkie rodzaje ścinania naroży poza G641, G644 i OSD		
	Zatrzymanie przebiegu wyprzedzającego:	nie		
	Działanie:	modalnie		
	<Value>:	Wartością tolerancji orientacji jest podany kąt.		
		Typ:	REAL	
		Jednostka:	stopień	
		Zakres wartości:	≥ 0:	Wartość tolerancji
			< 0:	Kasuje zaprogramowaną wartość tolerancji ⇒ Ponownie obowiązuje wartość tolerancji sparametryzowana w danej maszynowej lub ustawczej.
ATOL:	Adres do programowania **tolerancji specyficznej dla osi**			
	Obszar zastosowania:	• wszystkie funkcje kompresora • wygładzanie orientacji ORISON • wszystkie rodzaje ścinania naroży poza G641, G644 i OSD		
	Zatrzymanie przebiegu wyprzedzającego:	nie		
	Działanie:	modalnie		
	<Axis>:	Nazwa osi kanału, dla której powinna działać zaprogramowana tolerancja.		
	<Value>:	Wartością tolerancji osi jest zależnie od typu osi (liniowa lub obrotowa) podana długość lub podany kąt.		
		Typ:	REAL	
		Jednostka:	Dla osi liniowych:	cal/mm (zależnie od aktualnego ustawienia podawania wymiarów)
			Dla osi obrotowych:	stopień
		Zakres wartości:	≥ 0:	Wartość tolerancji
			< 0:	Kasuje zaprogramowaną wartość tolerancji ⇒ Ponownie obowiązuje wartość tolerancji sparametryzowana w danej maszynowej lub ustawczej.

Uwaga

Wartości tolerancji specyficzne dla kanału zaprogramowane za pomocą CTOL i OTOL mają wyższy priorytet niż wartości tolerancji zaprogramowane za pomocą ATOL.

Uwaga

Frame skalujące

Frame skalujące działają na zaprogramowane tolerancje w taki sam sposób jak na pozycje w osiach, tzn. tolerancja względna pozostaje taka sama.

Przykład

Kod programu	Komentarz
COMPCAD G645 G1 F10000	; Aktywacja funkcji kompresora COMPCAD.
X... Y... Z...	; Tutaj działają dane maszynowe i ustawcze.
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; Od tego miejsca działa tolerancja konturu 0,02 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; Od tego miejsca działa tolerancja konturu 0,005 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; Od tego miejsca działają dane maszynowe i ustawcze.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Zmienne systemowe

Odczyt za pomocą zatrzymania przebiegu wyprzedzającego

Aktualnie działające tolerancje są odczytywane w programie obróbki i akcji synchronicznej za pomocą następujących zmiennych systemowych:

- $AC_CTOL
  Tolerancja konturu specyficzna dla kanału, która działała przy przygotowywaniu aktualnego bloku przebiegu głównego.
  Jeżeli nie działa żadna tolerancja konturu, $AC_CTOL daje pierwiastek sumy kwadratów tolerancji osi geometrycznych.
- $AC_CTOL
  Tolerancja orientacji specyficzna dla kanału, która działała przy przygotowywaniu aktualnego bloku przebiegu głównego.
  Jeżeli żadna tolerancja orientacji nie działa, $AC_OTOL podczas aktywnej transformacji orientacji daje pierwiastek sumy kwadratów tolerancji osi orientacji, w przeciwnym przypadku wartość "-1".
- $AA_ATOL[<oś>]
  Tolerancja konturu specyficzna dla osi, która działała przy przygotowywaniu aktualnego bloku przebiegu głównego.
  Jeżeli aktywna jest tolerancja konturu, $AA_ATOL[<Oś geometryczna>] daje tolerancję konturu podzieloną przez pierwiastek z liczby osi geometrycznych.
  Jeżeli aktywna jest tolerancja orientacji i transformacja orientacji, $AA_ATOL[<Oś orientacji>] daje tolerancję orientacji podzieloną przez pierwiastek z liczby osi orientacji.

Uwaga

Gdy nie zostały zaprogramowane wartości tolerancji, wówczas zmienne $A nie są dostatecznie zróżnicowane, aby rozróżnić możliwie różne tolerancje poszczególnych funkcji.

Takie przypadki mogą wystąpić, gdy dane maszynowe i ustawcze ustawiają różne tolerancje dla funkcji kompresora, ścinania naroży i wygładzania orientacji. Zmienne systemowe dają wówczas największą wartość, która występuje przy aktywnych w tym momencie funkcjach. Gdy jest np. aktywna funkcja kompresora z tolerancją orientacji o wartości 0,1° i wygładzanie orientacji ORISON o wartości 1°, zmienna $AC_OTOL daje wartość "1". Gdy wygładzanie orientacji zostanie wyłączone, $AC_OTOL daje wartość „0,1”.

Odczyt bez zatrzymania przebiegu wyprzedzającego

Aktualnie działające tolerancje są odczytywane w programie obróbki za pomocą następujących zmiennych systemowych:

- $P_CTOL
  Aktualnie działająca tolerancja konturu specyficzna dla kanału.
- $P_OTOL
  Aktualnie działająca tolerancja orientacji specyficzna dla kanału.
- $PA_ATOL
  Aktualnie działająca tolerancja konturu specyficzna dla osi.

Warunki brzegowe

Tolerancje zaprogramowane za pomocą CTOL, OTOL i ATOL działają również na funkcje, które w sposób pośredni zależą od tej tolerancji.

- Ograniczenie błędu cięciwy przy obliczaniu wartości zadanej
- Funkcje podstawowe trybu powierzchni dowolnych

Zaprogramowanie CTOL, OTOL i ATOL **nie** ma wpływu na następujące funkcje ścinania narożników:

- Wygładzanie orientacji przy pomocy OSD
  OSD nie stosuje żadnej tolerancji, lecz odległość do przejścia między blokami.
- Ścinanie narożników za pomocą G644
  G644 nie służy do obróbki, lecz do optymalizacji wymiany narzędzia i innych ruchów bez obróbki.
- Ścinanie narożników za pomocą G645
  G645 zachowuje się prawie zawsze jak G642 i stosuje zaprogramowane tolerancje. Tylko w przypadku przejść między blokami po krzywej wielomianowej o stałej pochodnej ze skokiem łukowym, np. skośne przejście okrąg-koło, stosowana jest wartość tolerancji z danej maszynowej MD33120 $MA_PATH_TRANS_POS_TOL. Droga wygładzania może znajdować się w tych miejscach, to jest również na stronie zewnętrznej programowanego konturu, gdzie wiele zastosowań jest mniej tolerancyjnych. Poza tym wystarczające jest z reguły mała, stała ustawiona tolerancja do wyrównania zmian o charakterze łuku, o którym program NC nie musi myśleć.

12.12 Zmiana bloku przy aktywnym sprzężeniu (CPBC)

Przy pomocy polecenia CPBC zadawane jest kryterium zmiany bloku, które musi być spełnione, aby w programie obróbki przy aktywnym sprzężeniu została przeprowadzona zmiana bloku.

Składnia

```
CPBC[<oś nadążna>] = <kryterium>
```

Znaczenie

CPBC:	Kryterium zmiany bloku przy aktywnym sprzężeniu	
<oś nadążna>:	Identyfikator osi nadążnej	
<kryterium>:	Kryterium zmiany bloku	
	Typ:	STRING
	Wartość	**Znaczenie:** Nastąpi zmiana bloku
	"NOC"	niezależnie od stanu sprzężenia
	"IPOSTOP"	przy ruchu synchronicznym po stronie wartości zadanej
	"COARSE"	przy ruchu synchronicznym "zgrubnie" po stronie wartości rzeczywistej
	"FINE"	przy ruchu synchronicznym "dokładnie" po stronie wartości rzeczywistej

Przykład

Kod programu

```
; Zmiana bloku następuje przy:
;  - sprzężenie z osią nadążną X2 == aktywne
;  - ruch synchroniczny po stronie wartości zadanej ==
aktywny
CPBC[X2]="IPOSTOP"
```

Sprzężenia osi

13

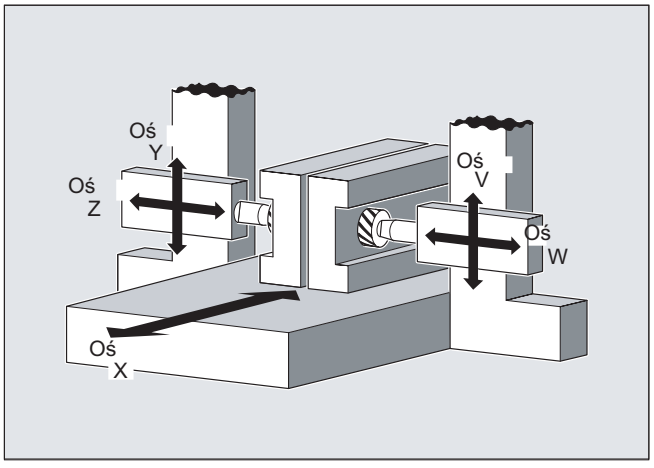
13.1 Nadążanie (TRAILON, TRAILOF)

Przy ruchu zdefiniowanej osi wiodącej przyporządkowane jej osie nadążne wykonują, przy uwzględnieniu współczynnika sprzężenia, drogi ruchu wyprowadzone od osi wiodącej.

Oś wiodąca i osie nadążne tworzą razem zespół nadążania.

Zakresy zastosowania

- Ruch w osi za pośrednictwem osi symulowanej. Oś wiodąca jest osią symulowaną, a oś nadążna osią rzeczywistą. W ten sposób można wykonywać ruch w osi rzeczywistej z uwzględnieniem współczynnika sprzężenia.
- Obróbka dwustronna z dwoma zespołami nadążania:
  1. Oś wiodąca Y, oś nadążna V
  2. Oś wiodąca Z, oś nadążna W



Składnia

```
TRAILON(<oś nadążna>,<oś wiodąca>,<współczynnik sprzężenia>)
TRAILOF(<oś nadążna>,<oś wiodąca>,<oś wiodąca 2>)
TRAILOF(<oś nadążna>)
```

Znaczenie

TRAILON:	Polecenie do włączenia i definicji zespołu nadążania	
	Działanie:	Modalnie
<oś nadążna>:	Parametr 1: Określenie osi nadążnej **Wskazówka:** Oś nadążna może także być osią wiodącą dla dalszych osi nadążnych. W ten sposób mogą być budowane różne zespoły nadążania.	
<oś wiodąca>:	Parametr 2: Określenie osi wiodącej	

<współczynnik sprzężenia>:	Parametr 3: Współczynnik sprzężenia Współczynnik sprzężenia podaje pożądany stosunek dróg osi nadążnej i osi wiodącej: <Współczynnik sprzężenia> = droga osi nadążnej/droga osi wiodącej	
	Typ:	REAL
	Ustawienie domyślne:	1
	Wprowadzenie wartości ujemnej powoduje przeciwny kierunek ruchu osi wiodącej i osi nadążnej. Jeżeli przy programowaniu współczynnik sprzężenia nie zostanie podany, wówczas automatycznie obowiązuje współczynnik 1.	
TRAILOF:	Polecenie do wyłączenia zespołu nadążania	
	Działanie:	Modalnie
	TRAILOF z 2 parametrami wyłącza tylko sprzężenie z podaną osią wiodącą: TRAILOF(<oś nadążna>,<oś wiodąca>) Jeżeli oś nadążna ma 2 osie wiodące, może nastąpić wezwanie do wyłączenia obydwu sprzężeń TRAILOF z 3 parametrami: TRAILOF(<oś nadążna>,<oś wiodąca>,<oś wiodąca 2>) Ten sam wynik daje zaprogramowanie TRAILOF bez podania osi wiodącej: TRAILOF(<oś nadążna>)	

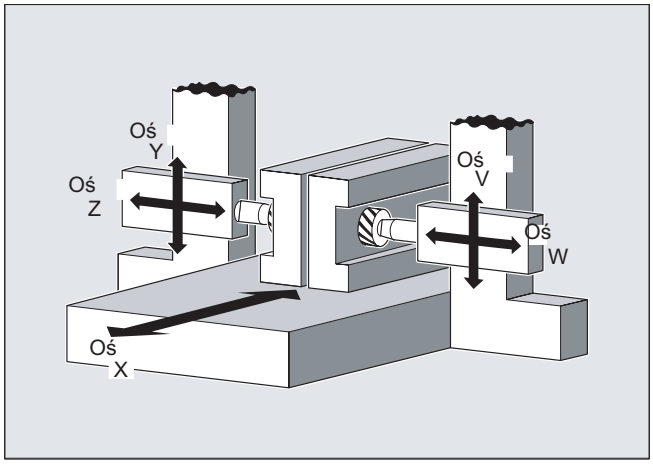
Uwaga

Nadążanie następuje zawsze w bazowym układzie współrzędnych (BKS).

Liczba zespołów nadążania, możliwych do równoczesnego uaktywnienia, jest ograniczona tylko przez możliwości kombinacji osi istniejących w maszynie.

Przykład

Przedmiot ma być obrabiany dwustronnie z przedstawioną konstelacją osi. W tym celu należy tworzyć 2 zespoły nadążania



Kod programu	**Komentarz**
…	
N100 TRAILON(V,Y)	; Włączenie 1. zespołu nadążania
N110 TRAILON(W,Z,-1)	; Włączenie 2. zespołu nadążania. Współczynnik sprzężenia ujemny: Oś nadążna wykonuje ruch w kierunku przeciwnym względem osi wiodącej.
N120 G0 Z10	; Dosuw w osi Z i W w przeciwnym kierunku osi.
N130 G0 Y20	; Dosuw w osi Y i V w tym samym kierunku osi.
…	
N200 G1 Y22 V25 F200	; Nałożenie zależnego i niezależnego ruchu osi nadążnej V.
…	
TRAILOF(V,Y)	; Wyłączenie 1. zespołu nadążania.
TRAILOF(W,Z)	; Wyłączenie 2. zespołu nadążania.

Dalsze informacje

Typy osi

Zespół osi nadążnych może składać się z dowolnych kombinacji osi liniowych i obrotowych. Jako oś wiodąca może przy tym również zostać zdefiniowana również oś symulowana.

Osie nadążne

Osi nadążnej mogą równocześnie zostać przyporządkowane maksymalnie 2 osie wiodące. Przyporządkowanie następuje w różnych zespołach nadążania.

Oś nadążna może być programowana przy pomocy wszystkich będących do dyspozycji poleceń ruchu (G0, G1, G2, G3, ...). Dodatkowo oprócz niezależnie definiowanych dróg oś nadążna wykonuje ruchy po drogach wyprowadzonych z jej osi wiodących ze współczynnikami sprzężenia.

Ograniczenie dynamiki

Ograniczenie dynamiki jest zależne od rodzaju uaktywnienia zespołu nadążania:

- Uaktywnienie w programie obróbki
  Jeżeli uaktywnienie nastąpi w programie obróbki i wszystkie osie wiodące znajdują się jako osie programowe w kanale uaktywniającym, wówczas przy wykonywaniu ruchów osi wiodących dynamika wszystkich osi nadążnych jest tak uwzględniana, by żadna z nich nie została przeciążona.
  Jeżeli uaktywnienie następuje w programie obróbki z osiami wiodącymi, które nie są aktywne jako osie programowe w kanale uaktywniającym ($AA_TYP ≠ 1), przy wykonywaniu ruchów osi wiodących dynamika osi nadążnej nie jest uwzględniana. W wyniku tego w przypadku osi nadążnych może dojść do przeciążenia z dynamika mniejszą, niż potrzebna dla sprzężenia.
- Uaktywnienie w akcji synchronicznej
  Jeżeli uaktywnienie nastąpi w akcji synchronicznej, przy wykonywaniu ruchów w osiach wiodących dynamika osi nadążych nie jest uwzględniana. W wyniku tego w przypadku osi nadążnych może dojść do przeciążenia z dynamika mniejszą, niż potrzebna dla sprzężenia.

⚠ OSTROŻNIE

Przeciążenie osi

Jeżeli zespół nadążania zostanie uaktywniony

- w akcjach synchronicznych
- w programie obróbki z osiami wiodącymi, które nie są osiami programowymi w kanale osi nadążnej,

, wówczas w zakresie szczególnej odpowiedzialności użytkownika/producenta maszyny leży podjęcie odpowiednich środków, by w wyniku ruchów postępowych osi wiodącej nie doszło do przeciążenia osi nadążnych.

Status sprzężenia

Status sprzężenia osi można odpytać w programie obróbki przy pomocy zmiennych systemowych:

$AA_COUP_ACT[<oś>]

Wartość	Znaczenie
0	Nie jest aktywne żadne sprzężenie
8	Nadążanie aktywne

Wyświetlenie osi nadążnej w przypadku osi obrotowych modulo

Jeżeli oś wiodąca i oś nadążna są osiami obrotowymi modulo, sumują się ruchy postępowe osi wiodącej n * 360°, gdzie n = 1, 2, 3... , na wyświetleniu pozostałej drogi osi holowanej, aż do wyłączenia sprzężenia.

Przykład: Segment programu z TRAILON, oś wiodąca B i oś nadążna C

Kod programu	Komentarz
`TRAILON(C,B,1)`	`; Włączenie sprzężenia`
`G0 B0`	`; Pozycja wyjściowa`

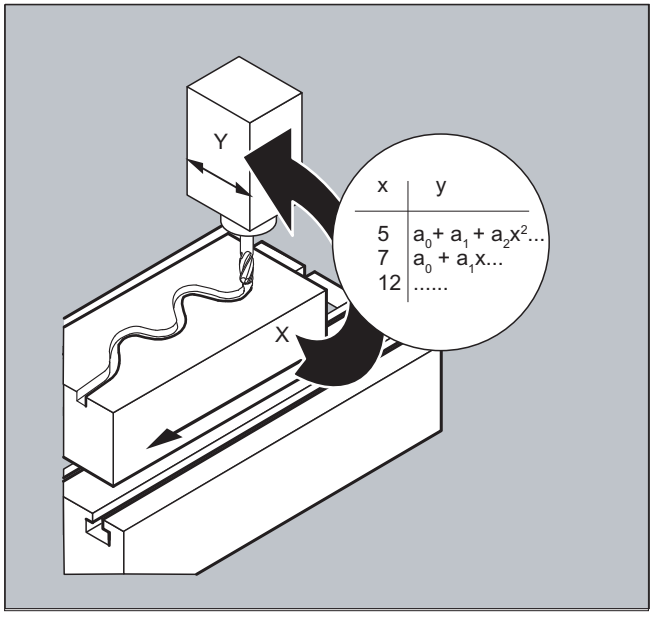
Kod programu	**Komentarz**
	; Wyświetlenie pozostałej drogi na początku bloku:
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

13.2 Tablice krzywych (CTAB)

Przy pomocy tablic krzywych można programować zależności pozycji i prędkości między dwoma osiami (wiodącą i nadążną). Definicja tablic krzywych następuje w programie obróbki.

Zastosowanie

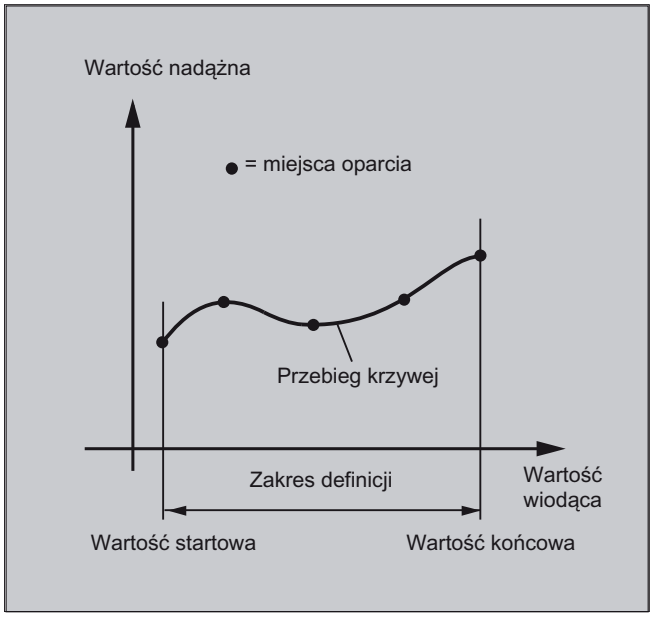
Tablice krzywych zastępują mechaniczne tarcze krzywkowe. Tablica krzywych tworzy przy tym podstawę osiowego sprzężenia wartości wiodącej w ten sposób, że stwarza funkcjonalną zależność między wartością wiodącą i nadążną: Przy odpowiednim zaprogramowaniu sterowanie z przyporządkowanych do siebie pozycji osi wiodącej i nadążnej oblicza wielomian, który odpowiada tarczy krzywkowej.



13.2.1 Definiowanie tablic krzywych (CTABDEF, CTABEND)

Tablica krzywych stanowi program obróbki albo jego segment, który charakteryzuje się poleceniem CTABDEF na początku i CTABEND na końcu.

W ramach tego segmentu programu obróbki są przez polecenia ruchu do poszczególnych pozycji osi wiodącej przyporządkowywane jednoznaczne pozycje osi nadążnej, które służą jako węzły interpolacji do obliczania przebiegu krzywej w formie wielomianu do maksymalnie 5. stopnia.



Warunek

Dla definicji tablic krzywych musi przez odpowiednie zaprojektowanie MD być zarezerwowane miejsce w pamięci (→ producent maszyny!).

Składnia

```
CTABDEF(<oś nadążna>,<oś wiodąca>,<n>,<okresowość>[,<miejsce
zapisania>])
...
CTABEND
```

Znaczenie

`CTABDEF( )`:	Początek definicji tablicy krzywych	
`CTABEND`:	Koniec definicji tablicy krzywych	
`<oś nadążna>`:	Oś, której ruch ma zostać obliczony poprzez tablicę krzywych	
`<oś wiodąca>`:	Oś, która daje wartości wiodące do obliczania ruchu osi nadążnej	
`<n>`:	Numer (**ID**) tablicy krzywych Numer tablicy krzywych jest jednoznaczny i niezależny od miejsca w pamięci. Nie może być tablic o takim samym numerze w pamięci statycznej i pamięci dynamicznej NC.	
`<okresowość>`:	Okresowość tablicy	
	`0`	Tablica nie jest okresowa (jest wykonywana tylko jeden raz, również w przypadku osi obrotowych)
	`1`	Tablica jest periodyczna względem osi wiodącej
	`2`	Tablica jest periodyczna względem osi wiodącej i osi nadążnej

<miejsce zapisania>:	Podanie miejsca zapisania (opcjonalne)	
	"SRAM"	Tablica krzywych jest tworzona w **statycznej** pamięci NC.
	"DRAM"	Tablica krzywych jest tworzona w **dynamicznej** pamięci NC.
	Wskazówka: Gdy dla tego parametru nie zostanie zaprogramowana żadna wartość, wówczas zostanie użyte standardowe miejsce w pamięci, ustawione przy pomocy MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	

Uwaga

Zastąpienie

Tablica krzywych jest zastępowana, gdy tylko przy ponownej definicji tablicy zostanie użyty jej numer (<n>) (wyjątek: tablica krzywych jest aktywna w sprzężeniu osi albo zablokowana przy pomocy CTABLOCK). **Przy zastępowaniu tablic krzywych nie jest wyprowadzane odpowiednie ostrzeżenie!**

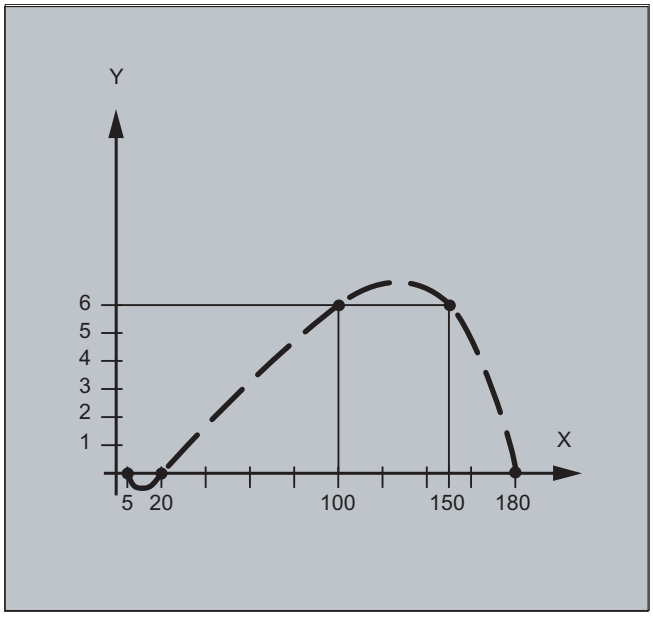
Przykłady

Przykład 1: Fragment programu jako definicja tablicy krzywych

Segment programu ma bez zmian być używany do definicji tablicy krzywych. Występujące tam polecenie do zatrzymania przebiegu wyprzedzającego STOPRE może pozostać i natychmiast staje się ponownie aktywne, gdy tylko segment programu nie jest już używany do definicji tablicy, a CTABDEF i CTABEND zostały usunięte.

```
Kod programu                    Komentarz
…
CTABDEF(Y,X,1,1)                ; Definicja tablicy krzywych.
…
IF NOT ($P_CTABDEF)
STOPRE
ENDIF
…
CTABEND
```

Przykład 2: Definicja nieperiodycznej tablicy krzywych



Kod programu	Komentarz
N100 CTABDEF(Y,X,3,0)	; Początek definicji nieperiodycznej tablicy krzywych o numerze 3.
N110 X0 Y0	; 1. instrukcja ruchu, ustala wartości startowe i 1. węzeł interpolacji:
	Wartość wiodąca: 0, wartość nadążna: 0
N120 X20 Y0	; 2. węzeł interpolacji:
	Wartość wiodąca: 0…20, wartość nadążna: Wartość startowa...0
N130 X100 Y6	; 3. węzeł interpolacji:
	Wartość wiodąca: 20…100, wartość nadążna 0…6
N140 X150 Y6	; 4. węzeł interpolacji:
	Wartość wiodąca: 100…150, wartość nadążna: 6…6
N150 X180 Y0	; 5. węzeł interpolacji:
	Wartość wiodąca: 150…180, wartość nadążna: 6…0
N200 CTABEND	; Koniec definicji. Tablica krzywych jest w swoim wewnętrznym przedstawieniu wytwarzana jako wielomian maksymalnie 5. stopnia. Obliczenie przebiegu krzywej z podanymi węzłami interpolacji jest zależne od modalnie wybranego rodzaju interpolacji (interpolacja kołowa, liniowa, spline). Jest odtwarzany stan programu obróbki przed początkiem definicji.

Przykład 3: Definicja periodycznej tablicy krzywych

Definicja periodycznej tablicy krzywych o numerze 2, zakres wartości prowadzącej od 0 do 360, ruch osi nadążnej od 0 do 45 i z powrotem do 0:

```
Kod programu                                          Komentarz
N10 DEF REAL DEPPOS
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1)                                  ; Początek definicji.
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                                          ; Koniec definicji.
;Test krzywej przez sprzężenie Y z X:
N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)

N170 DEPPOS=CTAB(75.0,2,GRADIENT)                     ; Odczyt funkcji tabeli przy war-
                                                      tości wiodącej 75.0.
N180 G0 X75 Y=DEPPOS                                  ; Pozycjonowanie osi wiodącej i
                                                      nadążnej.
;Po włączeniu sprzężenia nie jest konieczna synchronizacja osi nadążnej.
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30
```

Dodatkowe informacje

Wartość startowa i wartość końcowa tablicy krzywych

Wartością startową początku obszaru definicji tablicy krzywych jest pierwsze podanie przynależnych do siebie pozycji osi (pierwsza instrukcja ruchu) w ramach definicji tablicy krzywych. Wartość końcowa obszaru definicji tablicy krzywych jest odpowiednio określana przez ostatnie polecenie ruchu.

Dostępny zakres językowy

W ramach definicji tablicy krzywych jest do dyspozycji cały zakres języka NC.

Uwaga

Następujące dane są niedopuszczalne w definicjach tablic krzywych:

- Zatrzymanie przebiegu wyprzedzającego
- Skoki w ruchu osi wiodącej (np. przy zmianie transformacji)
- instrukcja ruchu tylko dla osi nadążnej
- Odwrócenie kierunku ruchu osi wiodącej, tzn. pozycja osi wiodącej musi być zawsze jednoznaczna
- Instrukcja CTABDEF i CTABEND na różnych poziomach programu

Działanie instrukcji modalnych

Wszystkie modalnie działające instrukcje, które nastąpią w ramach definicji tablicy krzywych, nie obowiązują z zakończeniem definicji tablicy krzywych. Program obróbki, w którym następuje definicja tablicy, znajduje się przez to w takim samym stanie przed i po definicji.

Przyporządkowania do parametrów R

Przyporządkowania do parametrów R w ramach definicji tablicy są cofane po CTABEND.

Przykład:

Kod programu	**Komentarz**
...	
R10=5 R11=20	; R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	; R10=25
X=R10	
CTABEND	
...	; R10=5

Uaktywnienie ASPLINE, BSPLINE, CSPLINE

Jeżeli w ramach definicji tablicy krzywych CTABDEF ... CTABEND nastąpi uaktywnienie ASPLINE, BSPLINE lub CSPLINE, wówczas przed tym uaktywnieniem spline powinien zostać zaprogramowany co najmniej jeden punkt startowy. Natychmiastowego uaktywnienia po CTABDEF należałoby unikać, ponieważ wówczas spline zależy od aktualnej pozycji osi przed definicją tablicy krzywych.

Przykład:

Kod programu
...
CTABDEF(Y,X,1,0)
X0 Y0

Kod programu
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND

Powtórne zastosowanie tablic krzywych

Zależność funkcjonalna osi wiodącej i nadążnej, obliczona poprzez tablicę krzywych, pozostaje zachowana pod wybranym numerem tablicy po zakończeniu programu i po POWER OFF, w przypadku gdy tablica jest zapisana w statycznej pamięci NC (SRAM).

Tablica, która została zapisana w pamięci dynamicznej (DRAM), jest przy POWER ON kasowana i musi ewentualnie zostać utworzona jeszcze raz.

Raz sporządzona tablica krzywych daje się zastosować do dowolnych kombinacji osi wiodącej i nadążnej i jest niezależna od tego, których osi użyto do sporządzenia tablicy.

Zastąpienie tablicy krzywych

Tablica krzywych jest zastępowana, gdy tylko przy ponownej definicji tablicy zostanie użyty jej numer.

Wyjątek: Tablica krzywych jest aktywna w sprzężeniu osi albo zablokowana przy pomocy CTABLOCK.

Uwaga

Przy zastępowaniu tablic krzywych nie jest wyprowadzane odpowiednie ostrzeżenie!

Czy definicja tablicy krzywych jest aktywna?

Przy pomocy zmiennej systemowej `$P_CTABDEF` można z programu obróbki w każdym czasie odczytać, czy definicja tablicy krzywych jest aktywna.

Zniesienie definicji tablicy krzywych

Segment programu obróbki po wyłączeniu instrukcji do definicji tablic krzywych przed nawias może być ponownie stosowany jako realny program obróbki.

Ładowanie tablic krzywych przez "wykonywanie ze źródła zewnętrznego"

Przy wykonywaniu tablic krzywych ze źródła zewnętrznego wielkość bufora doładowywania (DRAM) musi zostać tak dobrana przez MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE , by cała definicja tablicy mogła zostać zapisana w buforze. Wykonywanie programu obróbki zostanie w przeciwnym przypadku przerwane z alarmem 15150.

Skoki osi nadążnej

Zależnie od ustawienia danej maszynowej:
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
mogą być tolerowane skoki osi nadążnej przy braku ruchów w osi wiodącej.

13.2.2 Sprawdzenie istnienia pliku tablicy krzywych (CTABEXISTS)

Przy pomocy polecenia CTABEXISTS można sprawdzić, czy określony numer tablicy krzywych istnieje w pamięci NC.

Składnia

```
CTABEXISTS(<n>)
```

Znaczenie

CTABEXISTS:	Sprawdza, czy tablica krzywych o numerze <n> występuje w statycznej albo dynamicznej pamięci NC	
	0	Tablica nie istnieje
	1	Tablica istnieje
<n>:	Numer (**ID**) tablicy krzywych	

13.2.3 Kasowanie tablic krzywych (CTABDEL)

Przy pomocy CTABDEL można kasować tablice krzywych.

Uwaga

Tablice krzywych, które są aktywne w sprzężeniu osi, nie mogą zostać skasowane.

Składnia

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<miejsce zapisania>)
CTABDEL()
CTABDEL(,,<miejsce zapisania>)
```

Znaczenie

CTABDEL:	Polecenie do kasowania tablic krzywych
<n>:	Numer (**ID**) kasowanej tablicy krzywych Przy kasowaniu zakresu tablic krzywych CTABDEL(<n>,<m>) następuje przez <n> podanie numeru pierwszej tablicy zakresu.
<m>:	Przy kasowaniu zakresu tablic krzywych CTABDEL(<n>,<m>) następuje przez <m> podanie numeru ostatniej tablicy zakresu. <m> musi być większe od <n>!

<miejsce zapisania>:	Podanie miejsca zapisania (opcjonalne) Przy kasowaniu **bez** podania miejsca zapisania są kasowane podane tablice krzywych w pamięci statycznej i dynamicznej. Przy kasowaniu **z** podaniem miejsca zapisania są z podanych tablic krzywych kasowane te, które znajdują się w podanej pamięci. Pozostałe pozostają.	
	"SRAM"	Skasowanie w **statycznej** pamięci NC
	"DRAM"	Skasowanie w **dynamicznej** pamięci NC

Gdy CTABDEL zostanie zaprogramowane bez podania kasowanej tablicy krzywych, wówczas zostaną skasowane **wszystkie** tablice krzywych wzgl. tablice krzywych w podanej pamięci:

CTABDEL():	Kasuje wszystkie tablice krzywych w statycznej i dynamicznej pamięci NC
CTABDEL(,,"SRAM"):	Kasuje wszystkie tablice krzywych w statycznej pamięci NC
CTABDEL(,,"DRAM"):	Kasuje wszystkie tablice krzywych w dynamicznej pamięci NC

Uwaga

Gdy przy kasowaniu wielokrotnym CTABDEL(<n>,<m>) lub CTABDEL() co najmniej jedna z kasowanych tablic krzywych jest aktywna w sprzężeniu, wówczas polecenie skasowania nie jest wykonywane, tzn. **żadna** z zaadresowanych tablic nie zostanie skasowana.

13.2.4 Zablokowanie tablic krzywych przed skasowaniem i zastąpieniem (CTABLOCK, CTABUNLOCK)

Tablice krzywych mogą przez ustawienie blokad być chronione przed nie zamierzonym skasowaniem i zastąpieniem. Ustawioną blokadę można w każdym czasie wyłączyć.

Składnia

Ustawienie blokady:

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<miejsce zapisania>)
CTABLOCK()
CTABLOCK(,,<miejsce zapisania>)
```

Wyłączenie blokady:

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<miejsce zapisania>)
CTABUNLOCK()
CTABUNLOCK(,,<miejsce zapisania>)
```

Znaczenie

CTABLOCK:	Polecenie do **ustawienia** blokady przed skasowaniem/zastąpieniem
CTABUNLOCK:	Polecenie do **wyłączenia** blokady przed skasowaniem/zastąpieniem CTABUNLOCK zwalnia tablice krzywych zablokowane przy pomocy CTABLOCK. Tablice krzywych, które działają w aktywnym sprzężeniu, pozostają nadal zablokowane i nie mogą zostać skasowane. Blokada przy pomocy CTABLOCK zostanie wyłączona, gdy zostanie zniesiona blokada przez aktywne sprzężenie z wyłączeniem aktywności sprzężenia. Przez to tablica może zostać skasowana. Ponowne wywołanie CTABUNLOCK nie jest konieczne.
<n>:	Numer (**ID**) blokowanej/zwalnianej tablicy krzywych Przy blokowaniu/zwalnianiu zakresu tablic krzywych CTABLOCK(<n>,<m>)/ CTABUNLOCK(<n>,<m>) następuje przez <n> podanie numeru pierwszej tablicy zakresu.
<m>:	Przy blokowaniu/zwalnianiu zakresu tablic krzywych CTABLOCK(<n>,<m>)/ CTABUNLOCK(<n>,<m>) następuje przez <m> podanie numeru ostatniej tablicy zakresu. <m> musi być większe od <n>!
<miejsce zapisania>:	Podanie miejsca zapisania (opcjonalne) Przy ustawieniu/wyłączeniu blokady **bez** podania miejsca zapisania są blokowane/zwalniane podane tablice krzywych w pamięci statycznej i dynamicznej. Przy ustawieniu/wyłączeniu blokady **z** podaniem miejsca zapisania są z podanych tablic krzywych blokowane/zwalniane te, które znajdują się w podanej pamięci. Pozostałe nie są blokowane/zwalniane.
"SRAM"	Ustawienie/wyłączenie blokady w **statycznej** pamięci NC
"DRAM"	Ustawienie/wyłączenie blokady w **dynamicznej** pamięci NC

Gdy CTABLOCK/CTABUNLOCK zostanie zaprogramowane bez podania blokowanej/ zwalnianej tablicy krzywych, wówczas zostaną zablokowane/zwolnione **wszystkie** tablice krzywych wzgl. tablice krzywych w podanej pamięci:

CTABLOCK():	Blokuje wszystkie tablice krzywych w statycznej i dynamicznej pamięci NC
CTABLOCK(,,"SRAM"):	Blokuje wszystkie tablice krzywych w statycznej pamięci NC
CTABLOCK(,,"DRAM"):	Blokuje wszystkie tablice krzywych w dynamicznej pamięci NC
CTABUNLOCK():	Odblokowuje wszystkie tablice krzywych w statycznej i dynamicznej pamięci NC
CTABUNLOCK(,,"SRAM"):	Odblokowuje wszystkie tablice krzywych w statycznej pamięci NC
CTABUNLOCK(,,"DRAM"):	Odblokowuje wszystkie tablice krzywych w dynamicznej pamięci NC

13.2.5 Tablice krzywych: określenie właściwości tablicy (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Przy pomocy tych poleceń mogą być odpytywane ważne właściwości tablicy krzywych (numer tablicy, stan zablokowania, miejsce zapisania, okresowość).

Składnia

```
CTABID(<p>)
CTABID(<p>,<miejsce zapisania>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

Znaczenie

CTABID:	Daje **numer tablicy**, który w podanej pamięci jest wpisany jako <p>-ta tablica krzywych. Przykład: CTABID(1,"SRAM") daje numer pierwszej tablicy krzywych w statycznej pamięci NC. Pierwsza tablica krzywych odpowiada przy tym tablicy krzywych posiadającej najwyższy numer. **Wskazówka:** Jeżeli między kolejnymi wywołaniami CTABID zostanie zmieniona kolejność tablic krzywych w pamięci, np. przez skasowanie tablic krzywych przez CTABDEL, CTABID(<p>,...) o tym samym numerze <p> może dać inną tablicę krzywych, niż poprzednio.	
CTABISLOCK:	Zwraca **stan zablokowania** tablicy krzywych o numerze <n>:	
	0	Tablica nie jest zablokowana
	1	Tablica jest zablokowana przez CTABLOCK
	2	Tablica jest zablokowana przez aktywne sprzężenie
	3	Tablica jest zablokowana przez CTABLOCK i aktywne sprzężenie
	-1	Tablica nie istnieje
CTABMEMTYP:	Daje **miejsce zapisania** tablicy krzywych o numerze <n>:	
	0	Skasowanie w statycznej pamięci NC
	1	Tablica w dynamicznej pamięci NC
	-1	Tablica nie istnieje
CTABPERIOD:	Daje **okresowość** tablicy krzywych o numerze <n>:	
	0	Tablica jest nieperiodyczna
	1	Tablica jest periodyczna w osi wiodącej
	2	Tablica jest periodyczna w osi wiodącej i nadążnej
	-1	Tablica nie istnieje
<p>:	Numer wpisu w pamięci	
<n>:	Numer (**ID**) tablicy krzywych	
<miejsce zapisania>:	Podanie miejsca zapisania (opcjonalne)	
	"SRAM"	**statyczna** pamięć NC
	"DRAM"	**dynamiczna** pamięć NC
	Wskazówka: Gdy dla tego parametru nie zostanie zaprogramowana żadna wartość, wówczas zostanie użyte standardowe miejsce w pamięci, ustawione przy pomocy MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	

13.2.6 Odczyt wartości tablicy krzywych (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Następujące wartości tablicy krzywych można czytać w programie obróbki:

- Wartości osi nadążnej i osi wiodącej na początku i na końcu tablicy krzywych
- Wartości osi nadążnej na początku i na końcu segmentu krzywej
- Wartość osi nadążnej do wartości osi wiodącej
- Wartość osi wiodącej do wartości osi nadążnej
- Minimalna i maksymalna wartość osi nadążnej
  - w całym zakresie definicji tablicy krzywych
    lub
  - w zdefiniowanym przedziale tablicy krzywych

Składnia

```
CTABTSV(<n>,<gradient>[,<oś nadążna>])
CTABTEV(<n>,<gradient>[,<oś nadążna>])
CTABTSP(<n>,<gradient>[,<oś wiodąca>])
CTABTEP(<n>,<gradient>[,<oś wiodąca>])
CTABSSV(<wartość wiodąca>,<n>,<gradient>[,<oś nadążna>])
CTABSEV(<wartość wiodąca>,<n>,<gradient>[,<oś nadążna>])
CTAB(<wartość wiodąca>,<n>,<gradient>[,<oś nadążna>,<oś wiodąca>]
CTABINV(<wartość nadążna>,<wartość przybliżona>,<n>,<gradient>[,<oś
nadążna>,<oś wiodąca>]
CTABTMIN(<n>[,<oś nadążna>])
CTABTMAX(<n>[,<oś nadążna>])
CTABTMIN(<n>,<a>,<b>[,<oś nadążna>,<oś wiodąca>])
CTABTMAX(<n>,<a>,<b>[,<oś nadążna>,<oś wiodąca>])
```

Znaczenie

CTABTSV:	Odczyt wartości osi nadążnej na **początku** tablicy krzywych nr <n>
CTABTEV:	Odczyt wartości osi nadążnej na **końcu** tablicy krzywych nr <n>
CTABTSP:	Odczyt wartości osi wiodącej na **początku** tablicy krzywych nr <n>
CTABTEP:	Odczyt wartości osi wiodącej na **końcu** tablicy krzywych nr <n>
CTABSSV:	Odczyt wartości osi nadążnej na **początku** segmentu krzywej należącego do podanej wartości osi wiodącej (<wartość wiodąca>)
CTABSEV:	Odczyt wartości osi nadążnej na **końcu** segmentu krzywej należącego do podanej wartości osi wiodącej (<wartość wiodąca>)
CTAB:	Odczyt wartości osi nadążnej do podanej wartości osi wiodącej (<wartość wiodąca>)
CTABINV:	Odczyt wartości osi wiodącej do podanej wartości osi nadążnej (<wartość nadążna>)

CTABTMIN:	Określenie **minimalnej wartości** osi nadążnej: • w całym zakresie definicji tablicy krzywych lub • w zdefiniowanym przedziale <a> ... <b>
CTABTMAX:	Określenie **maksymalnej wartości** osi nadążnej: • w całym zakresie definicji tablicy krzywych lub • w zdefiniowanym przedziale <a> ... <b>
<n>:	Numer (**ID**) tablicy krzywych
<gradient>:	W parametrze <gradient> jest zwracane **nachylenie** funkcji tablicy krzywych w określonej pozycji
<oś nadążna>:	Oś, której ruch ma zostać obliczony poprzez tablicę krzywych (opcjonalnie)
<oś wiodąca>:	Oś, która daje wartości wiodące do obliczania ruchu osi nadążnej (opcjonalnie)
<wartość nadążna>:	Wartość osi nadążnej do odczytu przynależnej wartości osi wiodącej CTABINV
<wartość wiodąca>:	Wartość osi wiodącej: • do odczytu przynależnej wartości osi nadążnej przy CTAB lub • do wyboru segmentu krzywej przy CTABSSV/CTABSEV
<wartość przybliżona>:	Przyporządkowanie wartości osi wiodącej do wartości osi nadążnej przy CTABINV nie zawsze musi być jednoznaczne. CTABINV potrzebuje dlatego jako parametru wartości przybliżonej oczekiwanej wartości osi wiodącej.
<a>:	**Dolna** granica przedziału wartości wiodącej przy CTABTMIN/CTABTMAX
<b>:	**Górna** granica przedziału wartości wiodącej przy CTABTMIN/CTABTMAX
	Wskazówka: Przedział wprowadzanej wartości <a> ... <b> musi leżeć w ramach zakresu definicji tablicy krzywych.

Przykłady

Przykład 1:

Określenie wartości osi nadążnej i osi wiodącej na początku i na końcu tablicy krzywych, jak też minimalnej i maksymalnej wartości osi nadążnej w całym zakresie definicji tablicy krzywych.

```
Kod programu                          Komentarz
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)                 ; Początek definicji tablicy
N110 X0 Y10                           ; Pozycja startowa 1. segment tablicy
```

```
Kod programu                                Komentarz
N120 X30 Y40                                ; Pozycja końcowa 1. segment tablicy =
                                            pozycja startowa 2. segment tablicy
N130 X60 Y5                                 ; Pozycja końcowa 2. segment tablicy = ...
N140 X70 Y30
N150 X80 Y20
N160 CTABEND                                ; Koniec definicji tablicy.
...
N200 STARTPOS=CTABTSV(1,GRADIENT)           ; Wartość osi nadążnej na początku tablicy
                                            krzywych = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)             ; Wartość osi nadążnej na końcu tablicy krzy-
                                            wych = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)          ; Wartość osi wiodącej na początku tablicy
                                            krzywych = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)            ; Wartość osi wiodącej na końcu tablicy krzy-
                                            wych = 80
N240 MINVAL=CTABTMIN(1)                     ; Wartość minimalna osi nadążnej przy Y=5
N250 MAXVAL=CTABTMAX(1)                     ; Wartość maksymalna osi nadążnej przy Y=40
```

Przykład 2:

Określenie wartości osi nadążnej na początku i na końcu segmentu krzywej należącego do wartości osi wiodącej X=30.

```
Kod programu                                 Komentarz
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                        ; Początek definicji tablicy.
N110 X0 Y0                                   ; Pozycja startowa 1. segment tablicy
N120 X20 Y10                                 ; Pozycja końcowa 1. segment tablicy =
                                             pozycja startowa 2. segment tablicy
N130 X40 Y40                                 ; Pozycja końcowa 2. segment tablicy = ...
N140 X60 Y10
N150 X80 Y0
N160 CTABEND                                 ; Koniec definicji tablicy.
...
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)  ; Pozycja startowa Y w 2. segmencie = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)    ; Pozycja końcowa Y w 2. segmencie = 40
```

Dodatkowe informacje

Zastosowanie w akcjach synchronicznych

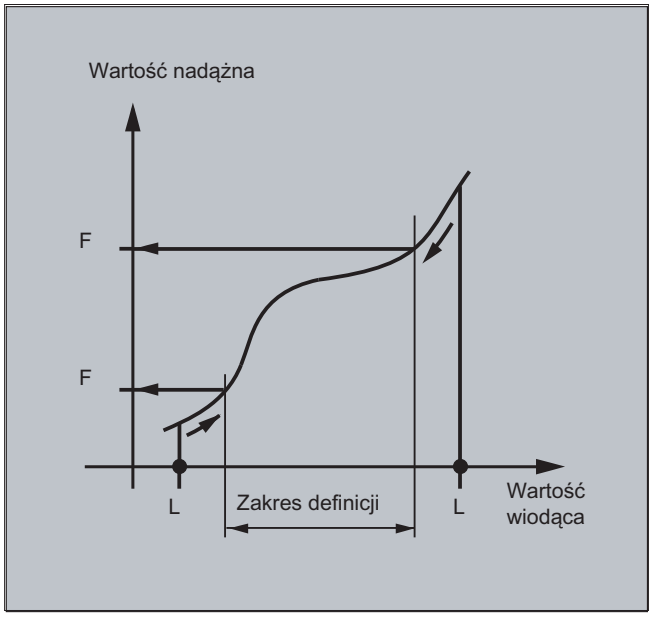
Wszystkie polecenia do odczytu wartości tablic krzywych mogą być stosowane również w akcjach synchronicznych (patrz też punkt "akcje synchroniczne ruchu").

Przy stosowaniu poleceń CTABINV, CTABTMIN i CTABTMAX należy zwracać uwagę, by:

- w chwili wykonania była dostępna wystarczająca moc NC
  lub
- przed wywołaniem została odpytana liczba segmentów tablicy krzywych, aby ewentualnie móc podzielić odnośna tablicę

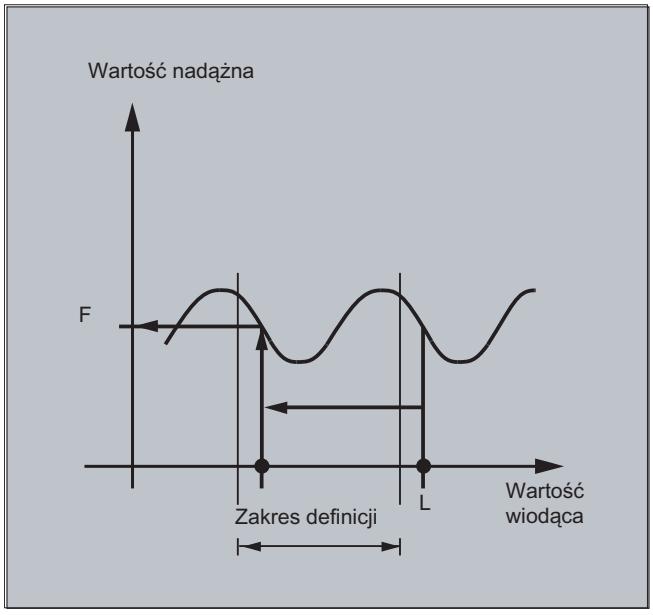
CTAB przy nieperiodycznych tablicach krzywych

Gdy podana `<wartość wiodąca>` leży poza zakresem definicji, jest jako wartość nadążna wyprowadzana granica górna wzgl. dolna.



CTAB przy periodycznych tablicach krzywych

Jeżeli podana `<wartość wiodąca>` leży poza zakresem definicji, jest interpretowana wartość wiodąca modulo zakresu definicji i wyprowadzana odpowiednia wartość nadążna:



Wartość przybliżona CTABINV

Polecenie CTABINV potrzebuje przybliżonej wartości oczekiwanej wartości wiodącej. CTABINV zwraca wartość wiodącą, która jest najbliższa wartości przybliżonej. Wartością przybliżoną może być np. wartość wiodąca z poprzedniego taktu interpolacji.

Nachylenie funkcji tablicy krzywych

Wyprowadzenie nachylenia (`<gradient>`) umożliwia obliczenie prędkości osi wiodącej lub nadążnej w odpowiedniej pozycji.

Podanie osi wiodącej lub nadążnej

Opcjonalne podanie osi wiodącej i/albo nadążnej jest ważne, w przypadku gdy oś wiodąca i oś nadążna są zaprojektowane w różnych jednostkach długości.

CTABSSV, CTABSEV

Polecenia CTABSSV i CTABSEV w następujących przypadkach **nie** nadają się**,** do odpytania zaprogramowanych segmentów:

- Są zaprogramowane okręgi albo ewolwenty.
- Jest aktywne fazowanie wzgl. zaokrąglanie przy pomocy `CHF`/`RND`.
- Jest aktywne ścinanie narożników przy pomocy `G643`.
- Jest aktywna kompresja bloków NC przy pomocy `COMPON`/`COMPCURV`/`COMPCAD`.

13.2.7 Tablice krzywych: sprawdzić wykorzystanie zasobów (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Przy pomocy tych poleceń programista ma możliwość poinformowania się o aktualnym wykorzystaniu zasobów dla tablic krzywych, segmentów tablicy i wielomianów.

Składnia

```
CTABNO
CTABNOMEM(<miejsce zapisania>)
CTABFNO(<miejsce zapisania>)
CTABSEGID(<n>,<miejsce zapisania>)
CTABSEG(<miejsce zapisania>,<rodzaj segmentu>)
CTABFSEG(<miejsce zapisania>,<rodzaj segmentu>)
CTABMSEG(<miejsce zapisania>,<rodzaj segmentu>)
CTABPOLID(<n>)
CTABPOL(<miejsce zapisania>)
CTABFPOL(<miejsce zapisania>)
CTABMPOL(<miejsce zapisania>)
```

Znaczenie

CTABNO:	Określenie liczby całkowitej **zdefiniowanych** tablic krzywych (w statycznej i dynamicznej pamięci NC)
CTABNOMEM:	Określenie liczby **zdefiniowanych** tablic krzywych w podanym <miejscu zapisania>
CTABFNO:	Określenie liczby **jeszcze możliwych** tablic krzywych w podanym <miejscu zapisania>
CTABSEGID:	Określenie liczby segmentów krzywej o podanym <rodzaju segmentu>, które są stosowane przez tablicę krzywych o numerze <n>
CTABSEG:	Określenie liczby **zastosowanych** segmentów krzywej o podanym <rodzaju segmentu> w podanym <miejscu zapisania>
CTABFSEG:	Określenie liczby **jeszcze możliwych** segmentów krzywej o podanym <rodzaju segmentu> w podanym <miejscu zapisania>
CTABMSEG:	Określenie liczby **maksymalnie możliwych** segmentów krzywej o podanym <rodzaju segmentu> w podanym <miejscu zapisania>
CTABPOLID:	Określenie liczby wielomianów krzywej, które są stosowane przez tablicę krzywych o numerze <n>
CTABPOL:	Określenie liczby **zastosowanych** wielomianów krzywych w podanym <miejscu zapisania>
CTABFPOL:	Określenie liczby **jeszcze możliwych** wielomianów krzywych w podanym <miejscu zapisania>
CTABMPOL:	Określenie liczby **maksymalnie możliwych** wielomianów krzywych w podanym <miejscu zapisania>
<n>:	Numer (**ID**) tablicy krzywych

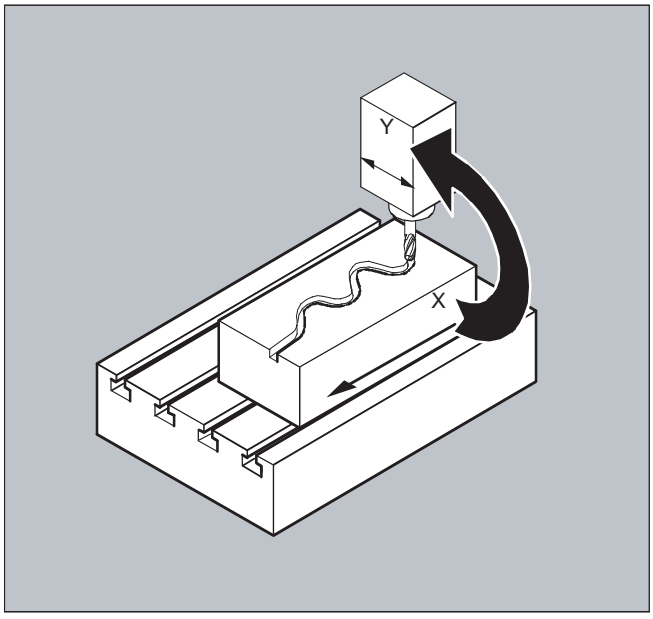
<miejsce zapisania>:	Podanie miejsca zapisania (opcjonalne)	
	"SRAM"	**statyczna** pamięć NC
	"DRAM"	**dynamiczna** pamięć NC
	Wskazówka: Gdy dla tego parametru nie zostanie zaprogramowana żadna wartość, wówczas zostanie użyte standardowe miejsce w pamięci, ustawione przy pomocy MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	
<rodzaj segmentu>:	Podanie rodzaju segmentu (opcjonalne)	
	"L"	Segmenty liniowe
	"P"	Segmenty wielomianowe
	Wskazówka: Jeżeli dla tego parametru nie zostanie zaprogramowana żadna wartość, wówczas zostanie wyprowadzona suma segmentów liniowych i wielomianowych.	

13.3 Osiowe sprzężenie wartości wiodącej (LEADON, LEADOF)

Uwaga

Ta funkcja nie jest dostępna dla SINUMERIK 828D!

W przypadku osiowego sprzężenia wartości wiodącej oś wiodąca i oś nadążna wykonują ruch synchroniczny. Przy tym każdorazowa pozycja osi nadążnej jest przez tablicę krzywej wzgl. obliczony z niej wielomian jednoznacznie przyporządkowana do - ew. symulowanej - pozycji osi wiodącej.



Osią wiodącą nazywa się ta oś, która daje wartości wejściowe dla tablicy krzywych. **Osią nadążną** nazywa się ta oś, która przyjmuje pozycje obliczone poprzez tablicę krzywej.

Sprzężenie wartości rzeczywistej i zadanej

Jako wartości wiodące, a więc wartości wyjściowe do określenia pozycji osi nadążnej mogą być stosowane:

- Wartości rzeczywiste pozycji osi wiodącej: Sprzężenie wartości rzeczywistej
- Wartości zadane pozycji osi wiodącej: Sprzężenie wartości zadanej

Sprzężenie wartości wiodącej obowiązuje zawsze w bazowym układzie współrzędnych.

Odnośnie sporządzania tablic krzywych patrz rozdział "Tablice krzywych".

Składnia

```
LEADON(<oś nadążna>,<oś wiodąca>,<n>)
LEADOF(<oś nadążna>,<oś wiodąca>)
```

albo wyłączenie bez podania osi wiodącej:

```
LEADOF(oś nadążna)
```

Sprzężenie wartości wiodącej może zostać włączone i wyłączone zarówno z programu obróbki, jak i podczas ruchu z akcji synchronicznych.

Znaczenie

LEADON:	Włączenie sprzężenia wartości wiodącej
LEADOF:	Wyłączenie sprzężenia wartości wiodącej
<oś nadążna>:	Oś nadążna
<oś wiodąca>:	Oś wiodąca
<n>:	Numer tablicy krzywych
$SA_LEAD_TYPE:	Przełączenie między sprzężeniem wartości zadanej i rzeczywistej

Wyłączenie sprzężenia wartości wiodącej, LEADOF

Z wyłączeniem sprzężenia wartości wiodącej oś nadążna staje się ponownie normalną osią rozkazową!

Osiowe sprzężenie wartości wiodącej i różne stany robocze, RESET

Zależnie od ustawienia w danej maszynowej sprzężenia wartości wiodącej są wyłączane przez RESET.

Przykład, sprzężenie wartości wiodącej z akcji synchronicznej

W przypadku prasy zwykłe mechaniczne sprzężenie między osią wiodącą (wał stempla) i osiami systemu transportowego składającego się z osi transportowych i osi pomocniczych ma zostać zastąpione przez elektroniczny system sprzęgający.

Przykład pokazuje, jak w przypadku prasy mechaniczny system transportowy jest zastępowany przez system elektroniczny. Procesy sprzęgania i rozprzęgania są realizowane jako **statyczne akcje synchroniczne**.

Od osi wiodącej LW (wał stempla) są sterowane osie transportowe i osie pomocnicze definiowane, jako osie nadążne poprzez tablice krzywych.

Osie nadążne

X oś posuwu wzgl. wzdłużna
YL oś zamykania wzgl. poprzeczna
ZL oś skoku
U posuw wałków, oś pomocnicza
V głowica prostująca, oś pomocnicza
W smarowanie, oś pomocnicza

Akcje

Jako akcje występują w akcjach synchronicznych np.:

- Sprzężenie, `LEADON(<oś nadążna>,<oś wiodąca>,<numer tablicy krzywych>)`
- Rozłączenie sprzężenia, `LEADOF(<oś nadążna>,<oś wiodąca>)`
- Ustawienie wartości rzeczywistej, `PRESETON(<oś>, <wartość>)`

- Ustawienie znacznika, `$AC_MARKER[i]=<wartość>`
- Rodzaj sprzężenia: rzeczywista/wirtualna wartość wiodąca
- Ruch do pozycji osi, `POS[<oś>]=<wartość>`

Warunki

Jako warunki podlegają ewaluacji szybkie wejścia cyfrowe, zmienne czasu rzeczywistego `$AC_MARKER` i porównania pozycji, powiązane operatorem logicznym AND.

Uwaga

W poniższym przykładzie zmiany wiersza, wcięcia i **tłusty** druk zastosowano wyłącznie w tym celu, by zwiększyć czytelność programowania. Dla sterowania wszystko, co jest pod jednym numerem wiersza, jest jednym wierszem.

Komentarz

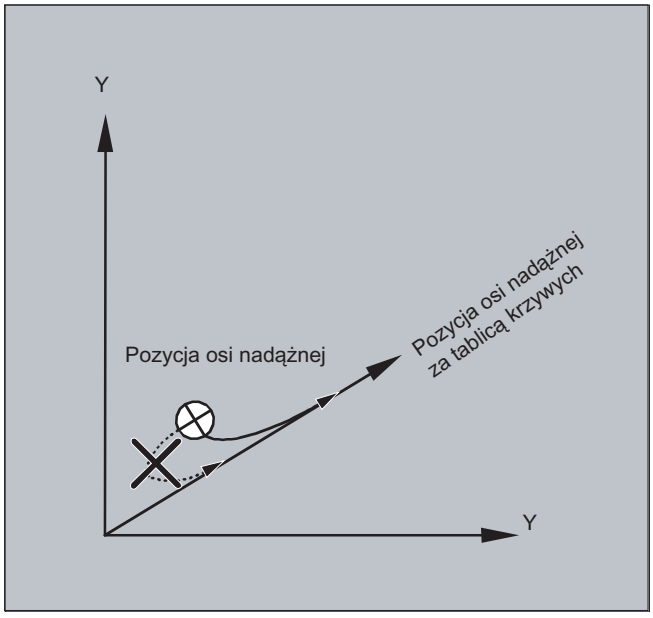
Kod programu	**Komentarz**
	; Definiuje wszystkie statyczne akcje synchroniczne.
	; ****cofnięcie znacznika
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
	; *** E1 0=>1 zesprzęglenie transfer WŁ
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_MARKER[0]=1	
	; **** E1 0=>1 zesprzęglenie posuw wałków WŁ
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1	
	; **** E1 0->1 zesprzęglenie głowica prostująca WŁ
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1	
	; **** E1 0->1 zesprzęglenie smarowanie WŁ
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0) DO LEADON(W,LW,4) PRESETON(W,0) $AC_MARKER[7]=1	
	; **** E2 0=>1 zesprzęglenie WYŁ
N30 IDS=3 EVERY ($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	

Opis

Sprzężenie wartości wiodącej wymaga synchronizacji osi wiodącej i osi nadążnej. Tę synchronizację można uzyskać tylko wtedy, gdy oś nadążna przy włączeniu sprzężenia wartości wiodącej znajduje się w zakresie tolerancji przebiegu krzywej obliczonego z tablicy krzywych.

Zakres tolerancji pozycji osi nadążnej jest zdefiniowany przez daną maszynową `MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE .`

Jeżeli oś nadążna przy włączeniu sprzężenia wartości wiodącej nie znajduje się jeszcze w odpowiedniej pozycji, wówczas praca synchroniczna jest podejmowana automatycznie, gdy tylko obliczona wartość zadana pozycji osi nadążnej zbliży się do rzeczywistej pozycji tej osi. Oś nadążna wykonuje przy tym podczas procesu synchronizacji ruch w kierunku, który jest zdefiniowany przez prędkość zadaną osi nadążnej (obliczona z prędkości osi wiodącej i według tablicy krzywych CTAB).

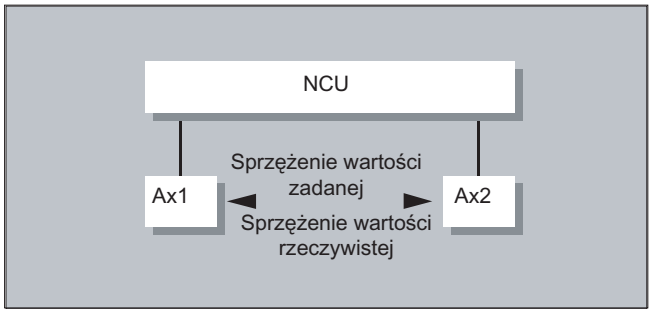


Bez pracy synchronicznej

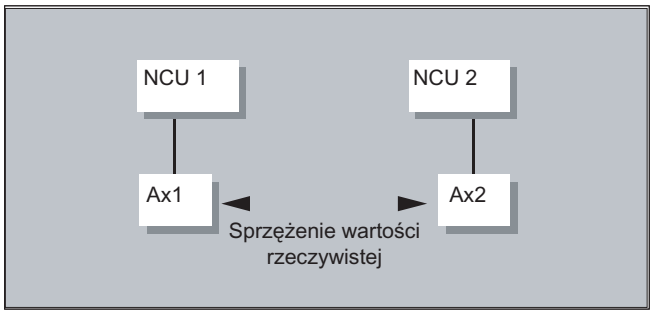
Jeżeli obliczona pozycja osi nadążnej oddala się z włączeniem sprzężenia wartości wiodącej od aktualnej pozycji osi nadążnej, praca synchroniczna nie jest podejmowana.

Sprzężenie wartości rzeczywistej i zadanej

Sprzężenie wartości zadanej daje w porównaniu ze sprzężeniem wartości rzeczywistej polepszony przebieg synchronizacji między osią wiodącą i osią nadążną i dlatego jest standardowo ustawione jako domyślne.



Sprzężenie wartości zadanej jest możliwe tylko wtedy, gdy oś wiodąca i nadążna są interpolowane przez tą samą NCU. W przypadku zewnętrznej osi wiodącej oś nadążna może zostać sprzężona z osią wiodącą tylko przez wartości rzeczywiste.



Przełączenie jest możliwe przez daną ustawczą $SA_LEAD_TYPE.

Przełączanie między sprzężeniem wartości rzeczywistej i zadanej powinno następować zawsze przy zatrzymanej osi nadążnej. Ponieważ tylko na postoju następuje po przełączeniu ponowna synchronizacja.

Przykład zastosowania

Czytanie wartości rzeczywistych może przy dużych wstrząsach maszyny następować z błędami. Przy zastosowaniu sprzężenia wartości wiodącej na linii pras może dlatego w operacjach roboczych o większych wstrząsach być konieczne przełączenie ze sprzężenia wartości rzeczywistej na sprzężenie wartości zadanej.

Symulacja wartości wiodącej przy sprzężeniu wartości zadanej

Przez daną maszynową interpolator osi wiodącej daje się oddzielić od serwonapędu. Przez to przy sprzężeniu wartości zadanej wartości zadane mogą być wytwarzane bez rzeczywistego ruchu osi wiodącej.

Wartości wiodące wytwarzane przez sprzężenie wartości zadanej są w celu użycia np. w akcjach synchronicznych możliwe do odczytania z następujących zmiennych:

- $AA_LEAD_P	Wartość wiodąca, pozycja
- $AA_LEAD_V	Wartość wiodąca, prędkość

Utworzenie wartości wiodących

Wartości wiodące mogą być wytwarzane do wyboru przy pomocy innych samemu programowanych metod. Tak wytworzone wartości wiodące są zapisywane do zmiennych

- $AA_LEAD_SP	Wartość wiodąca, pozycja
- $AA_LEAD_SV	Wartość wiodąca, prędkość

i z nich czytane. W celu użycia tych zmiennych musi być ustawiona dana ustawcza $SA_LEAD_TYPE = 2.

Status sprzężenia

W programie obróbki NC można odpytać na status sprzężenia przy pomocy następującej zmiennej systemowej:

`$AA_COUP_ACT[oś]`
0: Nie jest aktywne żadne sprzężenie
16: Sprzężenie wartości wiodącej aktywne

Zarządzanie statusem w przypadku akcji synchronicznych

Procesy łączenia i sprzęgania są zarządzane poprzez zmienne czasu rzeczywistego:

`$AC_MARKER[i] = n`
gdzie:
i numer znacznika
n wartość statusu

13.4 Przekładnia elektroniczna (EG)

Przy pomocy funkcji "przekładnia elektroniczna" jest możliwe sterowanie ruchem **osi nadążnej** po liniowym bloku ruchu w zależności od maksymalnie pięciu **osi wiodących**. Zależności między osiami wiodącymi i osią nadążną są dla każdej osi wiodącej definiowane poprzez współczynnik sprzężenia.

Obliczona składowa ruchu osi nadążnej jest tworzona przez dodawanie z poszczególnych składowych ruchu osi wiodących pomnożonych przez każdorazowy współczynnik sprzężenia. Przy uaktywnieniu zespołu osi EG można spowodować synchronizację osi nadążnej na zdefiniowaną pozycję. Zespół przekładni można z programu obróbki:

- zdefiniować,
- włączyć,
- wyłączyć,
- skasować

.

Ruch osi nadążnej można do wyboru wyprowadzić z

- wartości zadanych osi wiodących, jak też
- wartości rzeczywistych osi wiodących.

Jako rozszerzenie mogą być również realizowane nieliniowe zależności między osiami wiodącymi i osią nadążną poprzez **tablice krzywych** (patrz punkt "Zachowanie się w ruchu po torze"). Przekładnie elektroniczne mogą być kaskadowane, tzn. oś nadążna przekładni elektronicznej może być osią wiodącą dla kolejnej przekładni elektronicznej.

13.4.1 Zdefiniowanie przekładni elektronicznej (EGDEF)

Zespół osi przekładni elektronicznej jest ustalany przez podanie osi nadążnej i co najmniej jednej, ale co najwyżej pięciu osi wiodących z każdorazowym typem sprzężenia.

Warunek

Warunek dla definicji zespołu osi przekładni elektronicznej:

Dla osi nadążnej nie może być jeszcze zdefiniowane żadne sprzężenie osi (wzgl. istniejące musi zostać przedtem skasowane przy pomocy EGDEL).

Składnia

```
EGDEF (oś nadążna, oś wiodąca1,typ sprzężenia1,oś wiodąca ,typ
sprzężenia2,...)
```

Znaczenie

EGDEF:	Definicja przekładni elektronicznej
Oś nadążna:	Oś, na którą wpływają osie wiodące

`Oś wiodąca1`
`,...,`
`Oś wiodąca5`
	Osie, które wpływają na oś nadążną	

`Typ sprzężenia1`
`,...,`
`Typ sprzężenia5`
	Typ sprzężenia
	Typ sprzężenia nie musi być taki sam dla wszystkich osi wiodących i dlatego dla każdej osi wiodącej należy go podać oddzielnie.	

	Wartość:	**Znaczenie:**
	0	Na oś nadążną wpływa **wartość rzeczywista** odpowiedniej osi wiodącej.
	1	Na oś nadążną wpływa **wartość zadana** odpowiedniej osi wiodącej.

Uwaga

Współczynniki sprzężenia są przy definicji zespołu sprzężenia przekładni elektronicznej domyślnie ustawione na zero.

Uwaga

EGDEF wyzwala zatrzymanie przebiegu wyprzedzającego. Definicję przekładni przy pomocy EGDEF należy również wtedy stosować bez zmian, gdy w przypadku systemów jedna lub wiele osi wiodących wpływa poprzez **tablicę krzywych** na oś nadążną.

Przykład

Kod programu	**Komentarz**
`EGDEF(C,B,1,Z,1,Y,1)`	`; Definicja zespołu osi przekładni elektronicznej. Osie wiodące B, Z, Y wpływają na oś nadążną C poprzez wartość zadaną.`

13.4.2 Włączenie przekładni elektronicznej (EGON, EGONSYN, EGONSYNE)

Są trzy warianty włączenia zespołu osi przekładni elektronicznej.

Składnia

Wariant 1:

Zespół osi przekładni elektronicznej jest włączany bez synchronizacji selektywnie przy pomocy:
`EGON(FA,"tryb zmiany bloku",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)`

Wariant 2:

Zespół osi przekładni elektronicznej jest włączany z synchronizacją selektywnie przy pomocy:
`EGONSYN(FA,"tryb zmiany bloku",SynPosFA,[,LAi,SynPosLAi,Zi,Ni])`

Wariant 3:

Zespół osi przekładni elektronicznej jest z synchronizacją selektywnie włączany, a tryb dosunięcia zadawany przy pomocy:
`EGONSYNE(FA,"tryb zmiany bloku",SynPosFA,tryb dosunięcia[,LAi,SynPosLAi,Zi,Ni])`

Znaczenie

Wariant 1:

FA	Oś nadążna	
`Tryb zmiany bloku`:	Mogą być używane następujące tryby:	
	`"NOC"`	Zmiana bloku następuje natychmiast
	`"FINE"`	Zmiana bloku następuje przy "ruchu synchronicznym dokładnie"
	`"COARSE"`	Zmiana bloku następuje przy "ruchu synchronicznym zgrubnie"
	`"IPOSTOP"`	Zmiana bloku następuje przy pracy synchronicznej po stronie wartości zadanej
LA1, ... LA5	Osie wiodące	
Z1, ... Z5	Licznik dla współczynnika sprzężenia i	
N1, ... N5	Mianownik dla współczynnika sprzężenia i Współczynnik sprzężenia i = licznik i / mianownik i	

Wolno programować tylko osie wiodące, które przedtem zostały wyspecyfikowane przy pomocy EGDEF. Musi zostać zaprogramowana co najmniej jedna oś wiodąca.

Wariant 2:

`FA`	Oś nadążna	
`Tryb zmiany bloku`:	Mogą być używane następujące tryby:	
	`"NOC"`	Zmiana bloku następuje natychmiast
	`"FINE"`	Zmiana bloku następuje przy "ruchu synchronicznym dokładnie"
	`"COARSE"`	Zmiana bloku następuje przy "ruchu synchronicznym zgrubnie"
	`"IPOSTOP"`	Zmiana bloku następuje przy pracy synchronicznej po stronie wartości zadanej
`[,LAi,SynPosLAi,Zi,Ni]`	(nawiasów kwadratowych nie należy pisać) Min. 1, max 5 sekwencji:	
`LA1, ... LA5`	Osie wiodące	
`SynPosLAi`	Pozycja synchronizacji dla i-tej osi wiodącej	
`Z1, ... Z5`	Licznik dla współczynnika sprzężenia i	
`N1, ... N5`	Mianownik dla współczynnika sprzężenia i Współczynnik sprzężenia i = licznik i / mianownik i	

Wolno programować tylko osie wiodące, które przedtem zostały wyspecyfikowane przy pomocy EGDEF. Poprzez zaprogramowane "pozycje synchroniczne" dla osi nadążnej (SynPosFA) dla osi wiodących (SynPosLA) są definiowane pozycje, w których zespół

sprzężenia jest uważany za *synchroniczny*. O ile przekładnia elektroniczna przy włączeniu nie znajduje się w stanie synchronicznym, oś nadążna wykonuje ruch do swojej zdefiniowanej pozycji synchronicznej.

Wariant 3:

Parametry odpowiadają tym z wariantu 2 plus:

`Tryb dosunięcia`:	Mogą być używane następujące tryby:	
	`"NTGT"`	Ruch w optymalnym czasie do najbliższej luki międzyzębnej
	`"NTGP"`	Ruch po optymalnej drodze do najbliższej luki międzyzębnej
	`"ACN"`	Ruch osi obrotowej w kierunku ujemnym absolutnie
	`"ACP"`	Ruch osi obrotowej w kierunku dodatnim absolutnie
	`"DCT"`	W optymalnym czasie do zaprogramowanej pozycji synchronicznej
	`"DCP"`	Po optymalnej drodze do zaprogramowanej pozycji synchronicznej

Wariant 3 ma wpływ tylko na osie nadążne modulo, które są sprzężone z osiami wiodącymi modulo. Optymalizacja czasu uwzględnia granice prędkości osi nadążnej.

Dodatkowe informacje

Opis wariantów włączenia

Wariant 1:

Pozycje osi wiodących, jak też osi nadążnej w chwili włączenia są zapisywane jako "pozycje synchroniczne". "Pozycje synchroniczne" mogą być czytane przy pomocy zmiennych systemowych `$AA_EG_SYN`.

Wariant 2:

Gdy w zespole sprzężenia są osie modulo, ich pozycje podlegają redukcji modulo. Jest przez to zagwarantowany ruch do najbliższej możliwej pozycji synchronicznej (tzw. *synchronizacja względna*: np. najbliższy wrąb międzyzębny). Jeżeli dla osi nadążnej nie zostało udzielone "zezwolenie nałożenie osi nadążnej" sygnał interfejsowy DB(30 + numer osi), DBX 26 bit 4, nie następuje ruch do pozycji synchronicznej. Zamiast tego program jest zatrzymywany na bloku EGONSYN i jest wyświetlany samokasujący się alarm 16771, aż zostanie ustawiony w/ w sygnał.

Wariant 3:

Odstęp zębów (w stopniach) wynika z: 360 * Zi/Ni. W przypadku gdy oś nadążna stoi w chwili wywołania, optymalizacja drogi daje takie samo zachowanie się, jak optymalizacja czasu.

W przypadku osi nadążnej będącej już w ruchu następuje przy pomocy NTGP niezależnie od aktualnej prędkości osi nadążnej synchronizacja na najbliższy wrąb międzyzębny. W przypadku osi nadążnej będącej już w ruchu następuje przy pomocy NTGT zależnie od aktualnej prędkości osi nadążnej synchronizacja na najbliższy wrąb międzyzębny. Oś jest w tym celu ew. również hamowana.

Tablice krzywych

Jeżeli dla jednej z osi wiodących jest stosowana **tablica krzywych**, wówczas:

Ni mianownik współczynnika sprzężenia liniowego musi zostać ustawiony na 0. (mianownik 0 byłby dla sprzężeń liniowych niedopuszczalny). Mianownik jest dla sterowania znakiem, że

Zi należy interpretować jako numer będącej do zastosowania tablicy krzywych. Tablica krzywych o podanym numerze musi w chwili włączenia być już zdefiniowana.

LAi Podanie osi wiodącej odpowiada podaniu osi wiodącej w przypadku sprzężenia poprzez współczynnik sprzężenia (sprzężenie liniowe).

Dalsze wskazówki dot. użycia tablic krzywych i kaskadowania przekładni elektronicznych i ich synchronizacji można znaleźć w:
Literatura:
Podręcznik działania Funkcje specjalne; Sprzężenia osi i ESR (M3), punkt "Nadążanie i sprzężenie wartości wiodącej".

Zachowanie się przekładni elektronicznej przy włączeniu zasilania, RESECIE, zmianie trybu pracy, szukaniu bloku

- Po włączeniu zasilania **żadne** sprzężenie nie jest aktywne.
- Aktywne sprzężenia pozostają zachowane po RESECIE i zmianie trybu pracy.
- Przy szukaniu bloku polecenia przełączenia, kasowania, definicji przekładni elektronicznej nie są wykonywane i nie są gromadzone, lecz pomijane.

Zmienne systemowe przekładni elektronicznej

Przy pomocy zmiennych systemowych przekładni elektronicznej program obróbki może odczytywać aktualne stany zespołu osi przekładni elektronicznej i ew. reagować na nie.

Zmienne systemowe przekładni elektronicznej są oznaczone następująco:

$AA_EG_ ...

lub

$VA_EG_ ...

Literatura:
Podręcznik zmiennych systemowych

13.4.3 Wyłączenie przekładni elektronicznej (EGOFS, EGOFC)

Dla wyłączenia aktywnego zespołu osi przekładni elektronicznej istnieją 3 warianty.

Programowanie

Wariant 1:

Składnia	Znaczenie
`EGOFS(oś nadążna)`	Przekładnia elektroniczna jest wyłączana Oś nadążna jest hamowana do stanu zatrzymanego. Wywołanie wyzwala zatrzymanie przebiegu wyprzedzającego.

Wariant 2:

Składnia	Znaczenie
`EGOFS(oś nadążna,oś wiodąca1, …,oś wiodąca5)`	Ta parametryzacja polecenia pozwana na **selektywny** wpływ poszczególnych osi wiodących na ruch osi nadążnej.

Musi zostać podana co najmniej jedna oś wiodąca. Wpływ podanych osi wiodących na oś nadążną zostaje w sposób celowy wyłączony. Wywołanie wyzwala zatrzymanie przebiegu wyprzedzającego. Jeżeli pozostają jeszcze aktywne osie wiodące, wówczas oś nadążna pod ich wpływem nadal wykonuje ruch. Gdy wszystkie oddziaływania osi wiodących są w ten sposób wyłączone, oś nadążna jest hamowana do stanu zatrzymanego.

Wariant 3:

Składnia	Znaczenie
`EGOFC(wrzeciono nadążne1)`	Przekładnia elektroniczna jest wyłączana Wrzeciono nadążne wykonuje nadal ruch z prędkością obrotową / prędkością aktualną w chwili wyłączenia. Wywołanie wyzwala zatrzymanie przebiegu wyprzedzającego.

Uwaga

Ten wariant jest dozwolony tylko dla wrzecion.

13.4.4 Skasowanie definicji przekładni elektronicznej (EGDEL)

Zespół osi przekładni elektronicznej musi być wyłączony, zanim definicja będzie mogła zostać skasowana.

Programowanie

Składnia	Znaczenie
`EGDEL(oś nadążna)`	Definicja sprzężenia zespołu osi jest kasowana. Aż do osiągnięcia maksymalnej liczby równocześnie uaktywnionych zespołów osi, jest ponownie możliwe zdefiniowanie dalszych zespołów osi przy pomocy `EGDEF`. Wywołanie wyzwala zatrzymanie przebiegu wyprzedzającego.

13.4.5 Posuw na obrót (G95) / przekładnia elektroniczna (FPR)

Poleceniem FPR można również oś holowaną przekładni elektronicznej podać jako określającą posuw oś posuwu na obrót. Dla tego przypadku obowiązuje następujące zachowanie się:

- Posuw jest zależny od prędkości zadanej osi nadążnej przekładni elektronicznej.
- Prędkość zadana jest obliczana z prędkości wrzecion wiodących i osi wiodących modulo (które nie są osiami uczestniczącymi w tworzeniu konturu) i im przyporządkowanych współczynników sprzężenia.
- Składowe prędkości osi liniowych wzgl. osi wiodących nie modulo i ruchy nałożone osi nadążnej nie są uwzględniane.

13.5 Wrzeciono synchroniczne

W pracy synchronicznej jest wrzeciono wiodące (LS) i wrzeciono nadążne (FS), tzw. **para wrzecion synchronicznych**. Wrzeciono nadążne podąża przy aktywnym sprzężeniu (praca synchroniczna) za ruchami wrzeciona wiodącego odpowiednio do ustalonej zależności funkcyjnej.

Pary wrzecion synchronicznych dają się dla każdej maszyny zarówno projektować na stałe przy pomocy kanałowych danych maszynowych albo definiować specyficznie dla zastosowania przez program obróbki CNC. W kanale NC może równocześnie pracować do 2 par wrzecion synchronicznych.

Sprzężenie może zostać z programu obróbki

- zdefiniowane lub zmienione
- włączone
- wyłączone
- skasowane

.

Ponadto można zależnie od wersji oprogramowania

- czekać na warunek pracy synchronicznej
- zmienić zachowanie się pod względem zmiany bloku
- wybrać rodzaj sprzężenia, albo sprzężenie wartości zadanej albo sprzężenie wartości rzeczywistej, lub zadać przesunięcie kątowe między wrzecionem wiodącym i wrzecionem nadążnym
- przy włączeniu sprzężenia przejąć poprzednie zaprogramowanie wrzeciona nadążnego
- skorygować albo zmierzone albo już znane odchylenie ruchu synchronicznego

.

13.5.1 Wrzeciono synchroniczne: Programowanie (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

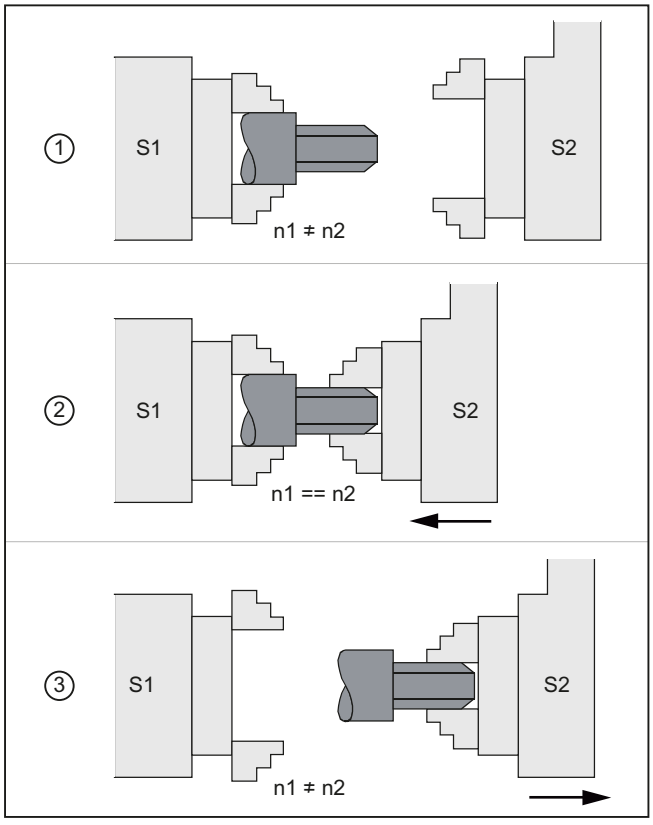
Funkcja "Wrzeciono synchroniczne" umożliwia synchroniczny pod względem liczby obrotów ruch wrzeciona nadążnego (FS) oraz wiodącego (LS) za pomocą zaprogramowanego stosunku przełożenia.

Funkcja udostępnia następujące tryby:

- Synchronizacja prędkości obrotowych ($n_{FS} = n_{LS}$)
- Synchronizacja położenia ($\phi_{FS} = \phi_{LS}$)
- Synchronizacja położenia z przesunięciem kątowym ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

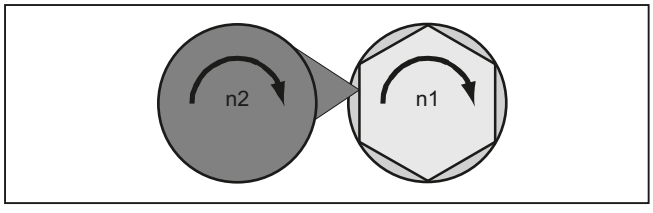
Przykłady zastosowania:

- Przekazanie przedmiotu obrabianego podczas ruchu np. do obróbki tylnej strony, stosunku przełożenia: 1:1



① Synchronizacja liczby obrotów
② Przekazanie przedmiotu obrabianego
③ Obróbka tylnej strony

- Obróbka konturów wielobocznych (toczenie wieloboku), synchroniczność prędkości obrotowej, stosunek przełożenia: n_1:n_2



Składnia

```
COUPDEF(<FS>,<LS>,<ÜFS>,<ÜLS>,<zmiana bloku>,<rodzaj sprzężenia>)
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>,<zmiana bloku>,<LS>,<zmiana bloku>)
```

Uwaga

Skrócony sposób zapisu

W przypadku instrukcji COUPOF, COUPOFS, COUPRES i COUPDEL jest możliwy skrócony sposób zapisu bez podania wrzeciona wiodącego.

Znaczenie

COUPDEF:	Definicja/zmiana sprzężenia specyficzna dla użytkownika
COUPON:	Włączenie sprzężenia. Wychodząc od aktualnej prędkości obrotowej wrzeciono nadążne synchronizuje się na wrzeciono wiodące
COUPONC:	Przejęcie sprzężenia z poprzednim zaprogramowaniem od M3 S... albo M4 S.... Różnica prędkości obrotowej wrzeciona nadążnego jest przejmowana natychmiast.
COUPOF:	Wyłączenie sprzężenia. • Z natychmiastową zmianą bloku: COUPOF(<S2>,<S1>) • Zmiana bloku dopiero po przejściu pozycji wyłączenia <POSFS> wzgl. <POSLS>: COUPOF(<S2>,<S1>,<POSFS>) COUPOF(<S2>,<S1>,<POSFS>,<POSLS>)
COUPOFS:	Wyłączenie sprzężenia z zatrzymaniem wrzeciona nadążnego. Zmiana bloku jak najszybciej z natychmiastową zmianą bloku: COUPOFS(<S2>,<S1>) Zmiana bloku dopiero po przejściu pozycji wyłączenia: COUPOFS(<S2>,<S1>,<POSFS>)
COUPRES:	Cofnięcie parametrów sprzężenia do zaprojektowanych MD i SD
COUPDEL:	Skasowanie sprzężenia definiowanego przez użytkownika
WAITC:	Czekanie na warunek pracy synchronicznej (NOC są anulowane na IPO przy zmianie bloku)
<FS>:	Określenie wrzeciona nadążnego
Parametry opcjonalne:	
<LS>:	Określenie wrzeciona wiodącego Podanie z numerem wrzeciona: z. B. S2, S1
<ZFS>, <NLS>:	Stosunek przełożenia między FS i LS. <ZFS> / <NLS> = Licznik/ Mianownik Ustawienie domyślne: <ZFS> / <NLS> = 1.0 ; Podanie mianownika opcjonalnie

<zmiana bloku>:	Zachowanie się pod względem zmiany bloku Zmiana bloku następuje:	
	"NOC"	natychmiast
	"FINE"	z osiągnięciem "ruchu synchronicznego dokładnie"
	"COARSE"	z osiągnięciem "ruchu synchronicznego zgrubnie"
	"IPOSTOP"	z osiągnięciem IPOSTOP, tzn. po uzyskaniu ruchu synchronicznego po stronie wartości zadanej (ustawienie domyślne)
	Zachowanie się pod względem zmiany bloku działa modalnie.	
<rodzaj sprzężenia>:	Rodzaj sprzężenia: Sprzężenie między FS i LS.	
	"DV"	Sprzężenie wartości zadanej (ustawienie domyślne)
	"AV"	Sprzężenie wartości rzeczywistej
	"VV"	Sprzężenie prędkości
	Rodzaj sprzężenia działa modalnie.	
<POSFS>:	Przesunięcie kątowe między wrzecionem wiodącym i nadążnym	
	Zakres wartości:	0°… 359,999°
<POSFS>,<POSLS>:	Pozycje wyłączenia wrzeciona nadążnego i wrzeciona wiodącego "Zmiana bloku jest zwalniana po przejściu POS_{FS}, POS_{LS}"	
	Zakres wartości:	0°… 359,999°

Przykłady

Praca z wrzecionem wiodącym i wrzecionem nadążnym

Kod programu	**Komentarz**
	Wrzeciono wiodące = wrzeciono główne = wrzeciono 1
	Wrzeciono nadążne = wrzeciono 2
N05 M3 S3000 M2=4 S2=500	Wrzeciono wiodące obraca się z 3000 obr/min, wrzeciono nadążne obraca się z 500 obr/min.
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	Definicja sprzężenia (może być też projektowane).
...	
N70 SPCON	Wrzeciono wiodące na regulację położenia (sprzężenie wartości zadanej).
N75 SPCON(2)	Wrzeciono nadążne na regulację położenia.
N80 COUPON(S2,S1,45)	Sprzężenie w ruchu na pozycję przesunięcia = 45 stopni.
...	
N200 FA[S2]=100	Prędkość pozycjonowania = 100 stopni/min
N205 SPOS[2]=IC(-90)	Nałożenie ruchu 90 stopni w kierunku ujemnym.
N210 WAITC(S2,"Fine")	Czekanie na ruch synchroniczny "dokładnie".
N212 G1 X... Y... F...	Obróbka
...	
N215 SPOS[2]=IC(180)	Nałożenie ruchu 180 stopni w kierunku dodatnim.
N220 G4 S50	Czas oczekiwania = 50 obrotów wrzeciona głównego
N225 FA[S2]=0	Uaktywnienie zaprojektowanej prędkości (MD).

Kod programu	Komentarz
N230 SPOS[2]=IC(-7200)	20 obrotów. Ruch z zaprojektowaną prędkością w kierunku ujemnym.
...	
N350 COUPOF(S2,S1)	Odsprzężenie w ruchu, S=S2=3000
N355 SPOSA[2]=0	Zatrzymanie wrzeciona nadążnego na zero stopni.
N360 G0 X0 Y0	
N365 WAITS(2)	Czekanie na wrzeciono 2.
N370 M5	Zatrzymanie wrzeciona nadążnego.
N375 M30	

Programowanie różnicy prędkości obrotowej

Kod programu	Komentarz
	Wrzeciono wiodące = wrzeciono główne = wrzeciono 1
	Wrzeciono nadążne = wrzeciono 2
N01 M3 S500	Wrzeciono wiodące obraca się z 500 obr/min.
N02 M2=3 S2=300	Wrzeciono nadążne obraca się z 300 obr/min.
...	
N10 G4 F1	Czas oczekiwania wrzeciona głównego.
N15 COUPDEF (S2,S1,-1)	Współczynnik sprzężenia ze stosunkiem przełożenia -1:1
N20 COUPON (S2,S1)	Uaktywnienie sprzężenia. Prędkość obrotowa wrzeciona nadążnego wynika z prędkości obrotowej wrzeciona wiodącego i współczynnika sprzężenia
...	
N26 M2=3 S2=100	Programowanie różnicy prędkości obrotowej.

Przykłady przejęcia ruchu do różnicy prędkości obrotowej

1. Włączenie sprzężenia przy uprzednim zaprogramowaniu wrzeciona nadążnego za pomocą COUPON

Kod programu	Komentarz
	Wrzeciono wiodące = wrzeciono główne = wrzeciono 1
	Wrzeciono nadążne = wrzeciono 2
N05 M3 S100 M2=3 S2=200	Wrzeciono wiodące obraca się z prędkością 100 obr/min, wrzeciono nadążne obraca się z prędkością 200 obr/min.
N10 G4 F5	Czas oczekiwania wrzeciona głównego = 5 sekund
N15 COUPDEF(S2,S1,1)	Stosunek przełożenia wrzeciona nadążnego do wrzeciona wiodącego wynosi 1,0 (ustawienie domyślne).
N20 COUPON(S2,S1)	Sprzężenie w ruchu na wrzeciono wiodące.
N10 G4 F5	Wrzeciono nadążne obraca się ze 100 obr/min.

2. Włączenie sprzężenia przy uprzednim zaprogramowaniu wrzeciona nadążnego za pomocą COUPON

Kod programu	Komentarz
	Wrzeciono wiodące = wrzeciono główne = wrzeciono 1
	Wrzeciono nadążne = wrzeciono 2
N05 M3 S100 M2=3 S2=200	Wrzeciono wiodące obraca się z prędkością 100 obr/min, wrzeciono nadążne obraca się z prędkością 200 obr/min.
N10 G4 F5	Czas oczekiwania wrzeciona głównego = 5 sekund
N15 COUPDEF(S2,S1,1)	Stosunek przełożenia wrzeciona nadążnego do wrzeciona wiodącego wynosi 1,0 (ustawienie domyślne).
N20 COUPONC(S2,S1)	Sprzężenie w ruchu na wrzeciono wiodące i przejęcie poprzedniej prędkości obrotowej do S2.
N10 G4 F5	S2 obraca się ze 100obr/min + 200obr/min = 300obr/min

3. Włączenie sprzężenia przy zatrzymanym wrzecionie nadążnym za pomocą COUPON

Kod programu	Komentarz
	Wrzeciono wiodące = wrzeciono główne = wrzeciono 1
	Wrzeciono nadążne = wrzeciono 2
N05 SPOS=10 SPOS[2]=20	Wrzeciono nadążne S2 w trybie pozycjonowania.
N15 COUPDEF(S2,S1,1)	Stosunek przełożenia wrzeciona nadążnego do wrzeciona wiodącego wynosi 1,0 (ustawienie domyślne).
N20 COUPON(S2,S1)	Sprzężenie w ruchu na wrzeciono wiodące.
N10 G4 F1	Sprzężenie jest zamykane, S2 zatrzymuje się na 20 stopniach.

4. Włączenie sprzężenia przy zatrzymanym wrzecionie nadążnym za pomocą COUPON

Uwaga

Tryb pozycjonowania lub tryb osi

Jeżeli wrzeciono nadążne przed sprzężeniem znajduje się w trybie pozycjonowania albo pracy jako oś, wówczas wrzeciono nadążne przy COUPON(<FS>,<LS>) i COUPONC(<FS>,<LS>) zachowuje się tak samo.

Uwaga

Wrzeciono wiodące i praca w trybie osi

Jeżeli wrzeciono wiodące przed definicją sprzężenia znajduje się w trybie pracy jako oś, również po włączeniu sprzężenia działa wartość graniczna prędkości z danej maszynowej:

MD32000 $MA_MAX_AX_VELO (maksymalna prędkość osi)

W celu uniknięcia tego zachowania się oś musi przed definicją sprzężenia zostać przełączona na tryb wrzeciona (M3 S... lub M4 S...).

Dodatkowe informacje

Projektowane sprzężenie

Przy projektowaniu sprzężenia wrzeciono wiodące i nadążne są ustalane przez daną maszynową. Zaprojektowanych wrzecion nie można zmienić w programie obróbki. Parametryzacja sprzężenia może nastąpić w programie obróbki przy pomocy `COUPDEF` (warunek: nie jest ustalona ochrona przed zapisem).

Sprzężenie definiowane przez użytkownika

Przy pomocy `COUPDEF` można na nowo zdefiniować lub zmienić sprzężenie w programie obróbki. Jeżeli jest już aktywne sprzężenie, musi ono przed definicją nowego sprzężenia zostać najpierw skasowane przy pomocy `COUPDEL`.

Sprzężenie jest kompletnie definiowane przez:

```
COUPDEF(<FS>,<LS>,<ÜFS>,<ÜLS>, zachowanie się przy zmianie bloku,
rodzaj sprzężenia)
```

Wrzeciono nadążne (FS) i wrzeciono wiodące (LS)

Przy pomocy nazw osi wrzeciona nadążnego (FS) i wrzeciona wiodącego (LS) sprzężenie jest określane jednoznacznie. Nazwy osi muszą być programowane z każdą instrukcją `COUPDEF`. Inne parametry sprzężenia działają modalnie i muszą być programowane tylko wtedy, gdy są zmieniane.

Przykład:

```
COUPDEF(S2,S1)
```

Stosunek przełożenia

Stosunek przełożenia podawany jest jako stosunek prędkości obrotowej wrzeciona nadążnego i wrzeciona wiodącego:

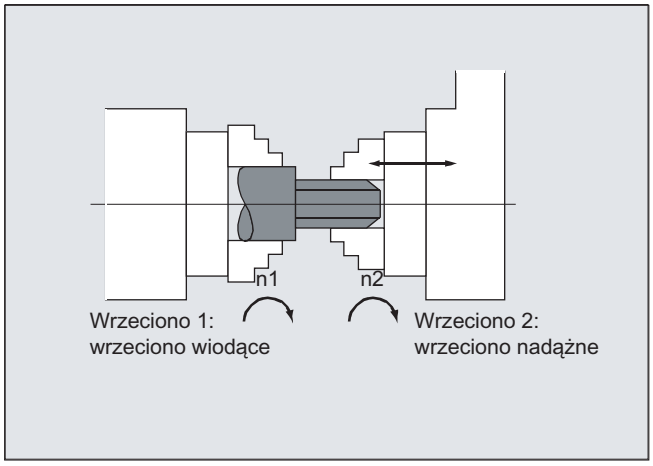
Wrzeciono nadążne/Wrzeciono wiodące = Licznik/Mianownik

Licznik musi zostać zaprogramowany. Mianownik musi zostać zaprogramowany. Dla mianownika ustalona wartość domyślna wynosi 1.0.

Przykład:

Wrzeciono nadążne S2 i wrzeciono wiodące S1, stosunek przełożenia = 1 / 1

```
COUPDEF(S2, S1, 1.0)
```



Uwaga

Stosunek przełożenia można również zmienić w ruchu podczas włączonego sprzężenia.

Zachowanie się przy zmianie bloku NOC, FINE, COARSE, IPOSTOP

Przy programowaniu zachowania się przy zmianie bloku jest możliwy następujący skrócony sposób zapisu:

- `"NO"`: natychmiast (ustawienie domyślne)
- `"FI"`: z osiągnięciem "ruchu synchronicznego dokładnie"
- `"CO"`: z osiągnięciem "ruchu synchronicznego zgrubnie"
- `"IP"`: z osiągnięciem IPOSTOP, tzn. po uzyskaniu ruchu synchronicznego po stronie wartości zadanej

Rodzaj sprzężenia

Uwaga

Rodzaj sprzężenia wolno jest zmienić tylko przy wyłączonym sprzężeniu.

Włączenie pracy synchronicznej COUPON, <POSFS>

- Włączenie sprzężenia z dowolnym przesunięciem kątowym między LS i FS:
  - `COUPON(S2,S1)`
  - `COUPON(S2)`
- Włączenie sprzężenia z zależnością kątową <`POSFS`>
  <`POSFS`> odnosi się do pozycji 0° wrzeciona wiodącego w dodatnim kierunku obrotów obszar wartości <`POSFS`>: 0°… 359,999°
  - `COUPON(S2,S1,30)`

Uwaga

Przesunięcie kątowe może zostać zmienione również przy aktywnym sprzężeniu.

Pozycjonowanie wrzeciona nadążnego

Również przy włączonym sprzężeniu wrzeciona synchronicznego wrzeciono nadążne można pozycjonować niezależnie od wrzeciona wiodącego w obszarze ±180°.

- Pozycjonowanie wrzeciona nadążnego za pomocą `SPOS`
  Przykład: `SPOS[2]=IC(-90)`
  Dalsze informacje dot. `SPOS` znajdują się w:
  Literatura:
  Podręcznik programowania Podstawy

Różnica prędkości obrotowej

Różnica prędkości obrotowej powstaje podczas pracy ze sterowaniem prędkością obrotową oraz aktywnego sprzężenia wrzecion synchronicznych poprzez podlegające ewaluacji nałożenie prędkości obrotowej wrzeciona nadążnego na podstawie ruchu wrzeciona wiodącego oraz prędkości obrotowej wrzeciona wiodącego na podstawie zaprogramowania wrzeciona:

- Sprzężenie wrzeciona synchronicznego za pomocą `COUPONC`
- `S<FS>=<prędkość obrotowa>` [`M<FS>=<kierunek obrotu>]`

  Uwaga

  Warunki brzegowe
  - Dla kierunku obrotu `M3` / `M4` należy również na nowo zaprogramować prędkość obrotową `S...`.
  - Nałożenie prędkości obrotowej wrzeciona (`M<kierunek obrotu> S<FS>`) przez ruch wrzeciona wiodącego podczas sprzężenia wrzecion synchronicznych `COUPONC` działa tylko wtedy, gdy jest zezwolenie dla nałożenia.
  - Dynamika wrzeciona wiodącego musi zostać na tyle ograniczona, by przy nałożeniu wrzeciona nadążnego jego wartości graniczne dynamiki nie zostały przekroczone.

  Odnośnie dalszych informacji dotyczących różnicy prędkości obrotowej patrz: :
  Literatura:
  Podręcznik działania Funkcje rozszerzające; Wrzeciono synchroniczne (S3)

Prędkość, przyspieszenie FA, ACC, OVRA, VELOLIMA

Osiową prędkość i przyspieszenie wrzeciona nadążnego można programować przy pomocy:

- `FA[SPI(S<n>)]` wzgl. `FA[S<n>]` (prędkość osiowa)
- `ACC[SPI(S<n>)]` wzgl. `ACC[S<n>]` (przyśpieszenie osiowe)
- `OVRA[SPI(S<n>)]` wzgl. `OVRA[S<n>]` (korekcja osiowa)
- `VELOLIMA[SPI(S<n>)]` wzgl. `VELOLIMA[S<n>]` (osiowe zwiększenie lub zmniejszenie prędkości)

Przy pomocy `<n>` = 1, 2, 3, ... (numery wrzecion nadążnych)

Literatura
Podręcznik programowania Podstawy

Uwaga

Zmniejszenie albo zwiększenie maksymalnego osiowego przyspieszenia nie działa w przypadku wrzecion.

Odnośnie dalszych informacji dotyczących dynamiki osiowej znajdują się w :
Literatura:
Podręcznik funkcjonowania Funkcje rozszerzone; Osie obrotowe (R2)

Programowane zachowanie się przy zmianie bloku WAITC

Przy pomocy `WAITC` może zostać zadane zachowanie się przy zmianie bloku, np. po zmianie parametrów sprzężenia albo procesów pozycjonowania, z różnymi warunkami ruchu synchronicznego (zgrubnie, dokładnie, `IPOSTOP`). Jeżeli warunki ruchu synchronicznego nie są podane, obowiązuje zachowanie się podane przy definicji `COUPDEF`.

Przykłady

- Czekanie na osiągnięcie warunku ruchu synchronicznego `FINE` w przypadku wrzeciona nadążnego S2 i `COARSE` w przypadku wrzeciona nadążnego S4:
  `WAITC(S2,"FINE",S4,"COARSE")`
- Czekanie na osiągnięcie warunku ruchu synchronicznego zgodnie z `COUPDEF`: `WAITC( )`

Wyłączenie sprzężenia COUPOF

Przy pomocy `COUPOF` można zadać zachowanie się sprzężenia przy wyłączeniu:

- Wyłączenie sprzężenia z natychmiastową zmiana bloku:
  - `COUPOF(S2,S1)` (z podaniem wrzeciona wiodącego)
  - `COUPOF(S2)` (bez podania wrzeciona wiodącego)
- Wyłączenie sprzężenia po przejściu pozycji wyłączenia. Zmiana bloku następuje po przejściu pozycji wyłączenia.
  - `COUPOF(S2,S1,150)` (pozycja wyłączenia FS: 150°)
  - `COUPOF(S2,S1,150,30)` (pozycja wyłączenia FS: 150°, LS: 30°)

Wyłączenie sprzężenia z zatrzymaniem wrzeciona nadążnego COUPOFS

Przy pomocy `COUPOFS` można zadać zachowanie się sprzężenia przy wyłączeniu z zatrzymaniem wrzeciona nadążnego:

- Wyłączenie sprzężenia z zatrzymaniem wrzeciona nadążnego z natychmiastową zmianą bloku:
  - `COUPOFS(S2,S1)` (z podaniem wrzeciona wiodącego)
  - `COUPOFS(S2)` (bez podania wrzeciona wiodącego)
- Wyłączenie sprzężenia po przejściu pozycji wyłączenia z zatrzymaniem wrzeciona nadążnego. Zmiana bloku następuje po przejściu pozycji wyłączenia.
  - `COUPOFS(S2,S1,150)` (pozycja wyłączenia FS: 150°)

Skasowanie sprzężeń COUPDEL

Przy pomocy `COUPDEL` sprzężenie jest kasowane:

- `COUPDEL(S2,S1)` (z podaniem wrzeciona wiodącego)
- `COUPDEL(S2)` (bez podania wrzeciona wiodącego)

Cofnięcie parametrów sprzężenia COUPRES

Przy pomocy `COUPRES` są uaktywniane wartości sprzężenia, sparametryzowane w danych maszynowych i ustawczych:

- `COUPRES(S2,S1)` (z podaniem wrzeciona wiodącego)
- `COUPRES(S2)` (bez podania wrzeciona wiodącego)

Zmienne systemowe

- Aktualny stan sprzężenia wrzeciona nadążnego
  Aktualny stan sprzężenia wrzeciona nadążnego może zostać przeczytany w sposób kodowany bitowo poprzez:
  `<wartość> = $AA_COUP_ACT[<FS>]`

Bit	<wartość>	Znaczenie
-	0	Nie jest aktywne żadne sprzężenie
2	4	Sprzężenie wrzeciona synchronicznego aktywne

Wskazówka

- Wszystkie inne wartości odnoszą się do pracy osi
- Jeśli wrzeciono stanowi wrzeciono nadążne o kilku sprzężeniach, jako stan sumaryczny zostanie zwrócona wartość stanu sprzężenia wszystkich sprzężeń.

- Aktualne przesunięcie kątowe
  Aktualne przesunięcie kątowe wrzeciona nadążnego do wrzeciona wiodącego może zostać odczytane poprzez:
  - `$AA_COUP_OFFS[<FS>]` (przesunięcie kątowe po stronie wartości zadanej)
  - `$VA_COUP_OFFS[<FS>]` (przesunięcie kątowe po stronie wartości rzeczywistej)

  Przykład zastosowania
  Korekcja różnicy przesunięcia kątowego w programie NC po wyłączeniu trybu śledzenia:
  Różnica przesunięcia kątowego = Zaprogramowane przesunięcie kątowe - zmienna systemowa

Literatura

Szczegółowe informacje dot. zmiennych systemowych znajdują się w:

Podręcznik lista parametrów - zmienne systemowe

13.6 Sprzężenie rodzajowe (CP...)

"Sprzężenie ogólne" jest ogólną funkcją sprzężenia, w której zebrane są wszystkie właściwości istniejących rodzajów sprzężenia (nadążanie, sprzężenie wartości wiodącej, przekładnia elektroniczna oraz wrzeciono synchroniczne).

Funkcja umożliwia elastyczny rodzaj programowania:

- Użytkownik może dla swoich aplikacji w sposób losowy wybrać właściwości sprzężenia (zasada modułów).
- Każda właściwość sprzężenia jest pojedynczo programowana.
- Możliwe są zmiany właściwości zdefiniowanego sprzężenia (np. współczynnik sprzężenia).
- Późniejsze używanie dalszych właściwości sprzężenia jest możliwe.
- Układ współrzędnych odniesienia osi nadążnej (bazowy układ współrzędnych lub układ współrzędnych maszyny) jest programowany.
- Określone właściwości sprzężenia mogą być również programowane w akcjach synchronicznych.
  Literatura: Podręcznik działania Akcje synchroniczne

Uwaga

Dotychczasowe wywołania sprzężenia dla nadążania (TRAIL*), sprzężenia wartości wiodącej (LEAD*), napędu elektronicznego (EG*) oraz wrzeciona synchronicznego (COUP*) są dalej wspierane poprzez cykle dopasowania.

Przegląd wszystkich słów kluczowych oraz właściwości sprzężenia

Następująca tabela przedstawia przegląd wszystkich słów kluczowych dla sprzężenia ogólnego, a tym samym programowanych właściwości sprzężenia:

Słowo kluczowe	Właściwość sprzężenia/Znaczenie	Składnia
CPDEF	Tworzenie trybu sprzężenia	CPDEF=(<FAx>)
CPDEL	Kasowanie trybu sprzężenia	CPDEL=(<FAx>)
CPLA	Definicja osi wiodącej	CPLA[<FAx>]=(<LAx>)
CPLDEF	Definicja osi wiodącej i tworzenie trybu sprzężenia (również możliwe za pomocą CPDEF + CPLA)	CPLDEF[<FAx>]=(<LAx>) lub CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>)
CPLDEL	Kasowanie osi wiodącej modułu sprzężenia (również możliwe za pomocą CPDEF + CPLA)	CPLDEL[<FAx>]=(<LAx>) lub CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>)
CPON	Włączenie modułu sprzężenia	CPON=(<FAx>)
CPOF	Wyłączenie modułu sprzężenia	CPOF=(<FAx>)

Słowo kluczowe	Właściwość sprzężenia/Znaczenie	Składnia		
CPLON	Włączenie osi wiodącej modułu sprzężenia	CPLON[<FAx>]=<LAx>		
CPLOF	Wyłączenie osi wiodącej modułu sprzężenia	CPLOF[<FAx>]=<LAx>		
CPLNUM	Licznik współczynnika sprzężenia	CPLNUM[FAx,LAx]=<wartość>		
CPLDEN	Mianownik współczynnika sprzężenia	CPLDEN[FAx,LAx]=<wartość>		
CPLCTID	Numer tablicy krzywych	CPLCTID[FAx,LAx]=<wartość>		
CPLSETVAL	Odniesienie sprzężenia	CPLSETVAL[FAx,LAx]="<Odniesienie sprzężenia>"		
		"<Odniesienie sprzężenia>":	"CMDPOS"	Sprzężenie wartości zadanej
			"CMDVEL"	Sprzężenie prędkości
			"ACTPOS"	Sprzężenie wartości rzeczywistej
CPFRS	Układ odniesienia współrzędnych	CPFRS[FAx]="<Odniesienie sprzężenia>"		
		"<Odniesienie sprzężenia>":	"BCS"	Bazowy układ współrzędnych
			"MCS"	Układ współrzędnych maszyny
CPBC	Kryterium zmiany bloku	CPBC[FAx]="<Kryterium zmiany bloku>"		
		"<Kryterium zmiany bloku>":	"NOC"	Zmiana bloku następuje niezależnie od stanu sprzężenia.
			"IPOSTOP"	Zmiana bloku następuje przy pracy synchronicznej po stronie wartości zadanej.
			"COARSE"	Zmiana bloku następuje przy ruchu synchronicznym "zgrubnie" po stronie wartości rzeczywistej.
			"FINE"	Zmiana bloku następuje przy ruchu synchronicznym "dokładnie" po stronie wartości rzeczywistej.
CPFPOS + CPON	Pozycja synchroniczna osi nadążnej przy włączeniu	CPON=FAx CPFPOS[FAx]=<wartość>		
CPLPOS + CPON	Pozycja synchroniczna osi wiodącej przy włączeniu	CPLPOS[FAx,LAx]=<wartość>		

Słowo kluczowe	Właściwość sprzężenia/Znaczenie	Składnia		
CPFMSON	Tryb synchronizacji	CPFMSON[FAx]="<Tryb synchronizacji>"		
		"<Tryb synchronizacji>":	"CFAST"	Sprzężenie zostanie zamknięte w optymalnym czasie.
			"CCOARSE"	Sprzężenie zostanie dopiero wtedy włączone, gdy zgodnie z regułą sprzężenia żądana pozycja osi nadążnej będzie się znajdowała w obszarze aktualnej pozycji osi nadążnej.
			"NTGT"	Nastąpi ruch w optymalnym czasie do najbliższej luki międzyzębnej.
			"NTGP"	Nastąpi ruch po optymalnej drodze do najbliższej luki międzyzębnej.
			"NRGT"	Ruch w optymalnym czasie do segmentu zgodnie ze stosunkiem ilości przekładni do liczby zębów.
			"NRGP"	Ruch w optymalnym czasie do segmentu zgodnie ze stosunkiem ilości przekładni do liczby zębów.
			"ACN"	Tylko dla osi obrotowych! Oś obrotowa wykonuje ruch w ujemnym kierunku osi do pozycji synchronicznej. Synchronizacja następuje od razu.
			"ACP"	Tylko dla osi obrotowych! Oś obrotowa wykonuje ruch w dodatnim kierunku osi do pozycji synchronicznej. Synchronizacja następuje od razu.
			"DCT"	Tylko dla osi obrotowych! Oś obrotowa wykonuje ruch w optymalnym czasie do zaprogramowanej pozycji synchronicznej. Synchronizacja następuje od razu.
			"DCP"	Tylko dla osi obrotowych! Oś obrotowa wykonuje ruch po optymalnej drodze do zaprogramowanej pozycji synchronicznej. Syn-

Słowo kluczowe	Właściwość sprzężenia/Znaczenie	Składnia		
				chronizacja następuje od razu.
CPFMON	Zachowanie się osi nadążnej przy włączeniu	CPFMON[FAx]="<Podtrzymanie włączenia>"		
		"<Podtrzymanie włączenia>":	"STOP"	Tylko przy wrzecionach! Aktywny ruch wrzeciona nadążnego zostanie zatrzymany przed włączeniem.
			"CONT"	Tylko dla wrzecion oraz osi przebiegu głównego! Aktualny ruch osi nadążnej/wrzeciona nadążnego zostanie przejęty do sprzężenia jako ruch początkowy.
			"ADD"	Tylko przy wrzecionach! Składowe ruchu sprzężenia działają dodatkowo dla aktualnego ruchu nałożonego, tzn. aktualny ruch osi nadążnej/wrzeciona nadążnego zostanie zachowany jako ruch nałożony.
CPFMOF	Zachowanie się osi nadążnej przy kompletnym wyłączeniu	CPFMOF[FAx]="<Podtrzymanie wyłączenia>"		
		"<Podtrzymanie wyłączenia>":	"STOP"	Zatrzymanie wrzeciona nadążnego/osi nadążnej. Aktywny ruch nałożony zostanie zahamowany do stanu zatrzymania. Następnie zostanie otwarte sprzężenie
			"CONT"	Tylko dla wrzecion oraz osi przebiegu głównego! Wrzeciono nadążne wykonuje nadal ruch z prędkością obrotową / prędkością aktualną w chwili wyłączenia.
CPFPOS + CPOF	Pozycja wyłączenia osi nadążnej przy wyłączeniu	CPOF=(FAx) CPFPOS[FAx]=<wartość>		

Słowo kluczowe	Właściwość sprzężenia/Znaczenie	Składnia		
CPMRESET	Zachowanie się sprzężenia przy RESET	CPMRESET[FAx]="<Podtrzymanie Reset>"		
		"<Podtrzymanie Reset>":	"NONE"	Aktualny stan sprzężenia pozostaje zachowany.
			"ON"	Jeżeli zostanie utworzony odpowiedni moduł sprzężenia, zostanie ono włączone. Wszystkie zdefiniowane powiązania osi wiodących zostaną aktywowane. Następuje to również wtedy, gdy wszystkie albo część tych powiązań jest już aktywnych, tzn. również przy całkowicie aktywnym sprzężeniu następuje nowa synchronizacja.
			"OF"	Aktywny ruch nałożony zostanie zahamowany do stanu zatrzymania. Następnie sprzężenie zostanie wyłączone. Jeśli dany moduł sprzężenia zostanie utworzony bez wyraźnej definicji (CPDEF), zostanie on usunięty. W innym przypadku pozostaje nadal tworzony, tzn. może być dalej używany.
			"OFC"	Możliwe tylko przy wrzecionach! Wrzeciono nadążne wykonuje nadal ruch z prędkością obrotową / prędkością aktualną w chwili wyłączenia. Sprzężenie zostanie wyłączone. Jeśli dany moduł sprzężenia zostanie utworzony bez wyraźnej definicji (CPDEF), zostanie on usunięty. W innym przypadku pozostaje nadal tworzony, tzn. może być dalej używany.
			"DEL"	Aktywny ruch nałożony zostanie zahamowany do stanu zatrzymania. Potem sprzężenie zostanie dezaktywowane, a następnie usunięte.
			"DELC"	Możliwe tylko przy wrzecionach!

Słowo kluczowe	Właściwość sprzężenia/Znaczenie	Składnia		
				Wrzeciono nadążne wykonuje nadal ruch z prędkością obrotową / prędkością aktualną w chwili wyłączenia. Sprzężenie zostanie dezaktywowane, a następnie usunięte.
CPMSTART	Zachowanie się sprzężenia przy starcie programu obróbki	CPMSTART[FAx]="<Podtrzymanie Reset>"		
		"<Podtrzymanie Start>":	"NONE"	Aktualny stan sprzężenia pozostaje zachowany.
			"ON"	Sprzężenie włączone. Wszystkie zdefiniowane powiązania osi wiodących zostaną aktywowane. Następuje to również wtedy, gdy wszystkie albo część tych powiązań jest już aktywnych, tzn. również przy całkowicie aktywnym sprzężeniu następuje nowa synchronizacja.
			"OF"	Sprzężenie zostanie wyłączone. Jeśli dany moduł sprzężenia zostanie utworzony bez wyraźnej definicji (CPDEF), zostanie on usunięty. W innym przypadku pozostaje nadal tworzony, tzn. może być dalej używany.
			"DEL"	Sprzężenie zostanie dezaktywowane, a następnie usunięte.
CPMPRT	Zachowanie się sprzężenia przy starcie programu obróbki w trakcie wyszukiwania bloku za pomocą testu programu	CPMPRT[FAx]="<Podtrzymanie Start>"		
		"<Podtrzymanie Start>":	patrz CPMSTART	
CPLINTR	Wartość przesunięcia wartości wejściowej dla osi wiodącej	CPLINTR[FAx,LAx]=<wartość>		
CPLINSC	Współczynnik skalowania wartości wejściowej dla osi wiodącej	CPLINSC[FAx,LAx]=<wartość>		
CPLOUTTR	Wartość przesunięcia wartości wyjściowej sprzężenia	CPLOUTTR[FAx,LAx]=<wartość>		
CPLOUTSC	Współczynnik skalowania wartości wyjściowej sprzężenia	CPLOUTSC[FAx,LAx]=<wartość>		
CPSYNCOP	Wartość progowa dla ruchu synchronicznego pozycjonowania "zgrubnie"	CPSYNCOP[FAx]=<wartość>		

Słowo kluczowe	Właściwość sprzężenia/Znaczenie	Składnia		
CPSYNFIP	Wartość progowa dla ruchu synchronicznego pozycjonowania "dokładnie"	CPSYNFIP[FAx]=<wartość>		
CPSYNCOP2	Druga wartość progowa dla ruchu synchronicznego pozycjonowania "zgrubnie"	CPSYNCOP2[FAx]=<wartość>		
CPSYNFIP2	Druga wartość progowa dla ruchu synchronicznego pozycjonowania "dokładnie"	CPSYNFIP2[FAx]=<wartość>		
CPSYNCOV	Wartość progowa dla ruchu synchronicznego szybkiego "zgrubnie"	CPSYNCOV[FAx]=<wartość>		
CPSYNFIV	Wartość progowa dla ruchu synchronicznego szybkiego "dokładnie"	CPSYNFIV[FAx]=<wartość>		
CPMBRAKE	Zachowanie się osi nadążnej przy określonych sygnałach i rozkazach Stop	CPMBRAKE[FAx]=<wartość kodowana bitowo>		
CPMVDI	Zachowanie się osi nadążnej na określone sygnały interfejsu NC/PLC	CPMVDI[FAx]=<wartość kodowana bitowo>		
CPMALARM	Ukrywanie specjalnych, dotyczących sprzężenia wyprowadzeń alarmów	CPMALARM[FAx]=<wartość kodowana bitowo>		
CPSETTYPE	Typ sprzężenia	CPSETTYPE[FAx]="<Typ sprzężenia>"		
		"<Typ sprzężenia>":	"CP"	Dowolne programowanie
			"TRAIL"	Typ sprzężenia "Nadążanie"
			"LEAD"	Typ sprzężenia "Sprzężenie wartości wiodącej"
			"EG"	Typ sprzężenia "Przekładnia elektroniczna"
			"COUP"	Typ sprzężenia "Sprzężenie synchroniczne"

FAx: Oś nadążna/wrzeciono nadążne

LAx: Oś wiodąca/wrzeciono wiodące

Uwaga

Właściwości sprzężenia, które nie zostały wyraźnie zaprogramowane (w programie obróbki lub w akcjach synchronicznych), będą działały w oparciu o ich ustawienia standardowe.

W zależności od ustawienia słowa kluczowego CPSETTYPE zamiast ustawień standardowych (CPSETTYPE="CP") mogą działać również ustawione domyślnie właściwości sprzężenia.

Literatura

Szczegółowe informacje odnośnie sprzężenia ogólnego patrz:

- Podręcznik działania Funkcje specjalne; M3: Sprzężenia osi, rozdział: "Sprzężenie ogólne"

13.7 Sprzężenie Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

"Sprzężenie Master/Slave" umożliwia:

- Sprzężenie osi Slave z osią Master w czasie postoju uczestniczących osi.
- Sprzężenie i rozłączenie wrzecion **obracających się**, o sterowanej prędkości obrotowej.
- Projektowanie dynamiczne.

Uwaga

Tryb pozycjonowania

W przypadku osi i wrzecion w trybie pozycjonowania sprzężenie jest załączane i rozłączane tylko w czasie postoju.

Składnia

```
MASLON(<Slave_1>,<Slave_2>,...)
MASLOF(<Slave_1>,<Slave_2>,...)
MASLOFS(<Slave_1>,<Slave_2>,...)
```

Projektowanie dynamiczne:

```
MASLDEF(<Slave_1>,<Slave_2>,...,<Master>)
MASLDEL(<Slave_1>,<Slave_2>,...)
```

Znaczenie

MASLON:	Włączenie sprzężenia tymczasowego Master/Slave	
	<Slave_x>,...:	Oś Slave 1 ... n
MASLOF:	Rozłączenie aktywnego sprzężenia Master/Slave	
	<Slave_1>,...:	Oś Slave 1 ... n
MASLOFS:	Rozłączenie sprzężenia Master/Slave i automatyczne wyhamowanie wrzecion Slave (patrz wskazówka "Zachowanie się sprzężonych wrzecion w pracy ze sterowaną prędkością obrotową"!)	
	<Slave_1>,...:	Oś Slave 1 ... n
MASLDEF:	Utworzenie/zmiana zespołu Master/Slave z programu obróbki	
	<Slave_1>,...:	Oś Slave 1 ... n
	<Master>:	Oś Master
MASLDEL:	Rozłączenie sprzężenia Master/Slave i kasowanie definicji zespołu	
	<Slave_1>,...:	Oś Slave 1 ... n
	Wskazówka Definicje Master/Slave, zaprojektowane w danych maszynowych, pozostają zachowane.	

Uwaga

Zachowanie się sprzężenia w przypadku wrzecion w trybie sterowania prędkością obrotową

W przypadku wrzecion w trybie sterowania prędkością obrotową zachowanie się MASLON, MASLOF, MASLOFS i MASLDEL pod względem sprzężenia jest ustalane bezpośrednio przez następującą daną maszynową:

MD37263 $MA_MS_SPIND_COUPLING_MODE

W ustawieniu standardowym przy pomocy MD37263 = 0 sprzężenie i rozłączenie osi slave następuje wyłącznie w stanie zatrzymania uczestniczących osi. MASLOFS odpowiada MASLOF.

Przy MD37263 = 1 instrukcja sprzężenia jest wykonywana bezpośrednio, a przez to również w ruchu. Sprzężenie jest w przypadku MASLON natychmiast tworzone, a przy MASLOFS albo MASLOF natychmiast rozłączane. Obracające się w tym momencie wrzeciona Slave zachowują przy MASLOF swoje prędkości obrotowe, aż do ponownego zaprogramowania prędkości obrotowej. W przypadku MASLOFS są natomiast automatycznie hamowane.

Uwaga

W przypadku MASLOF/MASLOFS odpada domyślne zatrzymanie przebiegu wyprzedzającego. Ze względu na brak zatrzymania przebiegu wyprzedzającego zmienne systemowe $P nie dają dla osi Slave zaktualizowanych wartości, aż do chwili ponownego zaprogramowania.

Uwaga

Dla osi Slave wartość rzeczywista może przez PRESETON być synchronizowana na taką samą wartość osi Master. W tym celu trwałe sprzężenie Master/Slave musi zostać na krótki czas rozłączone, aby wartość rzeczywistą nie bazowanej osi Slave przy pomocy POWER ON ustawić na wartość osi Master. Następnie trwałe sprzężenie jest ponownie tworzone.

Sprzężenie trwałe master/slave jest uaktywniane przez następujące ustawienie:
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1
Nie ma ono wpływu na polecenia językowe sprzężenia tymczasowego.

Przykłady

Przykład 1: Ustawienie wartości rzeczywistej w przypadku osi slave sprzężenia master/slave

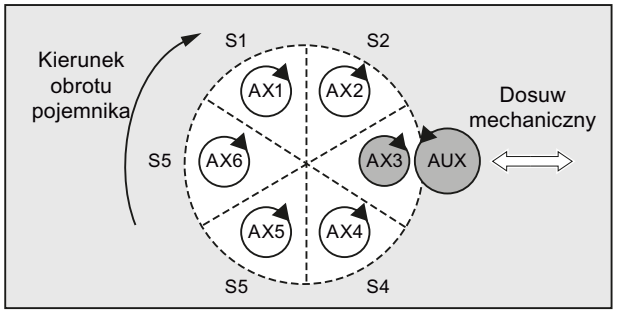
W przypadku permanentnego sprzężenia master/slave wartość rzeczywista osi slave jest przy pomocy PRESETON ustawiana na wartość osi master.

Kod programu	Komentarz
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; Wyłączenie stałego sprzężenia osi slave
NEWCONF	; Aktywacja zmiany danych maszynowych
STOPRE	
MASLOF(Y1)	; Wyłączenie sprzężenia tymczasowego
PRESETON(AX2,$VA_IM(M_AX))	; Wartość rzeczywista osi slave = wartość rzeczywista osi master

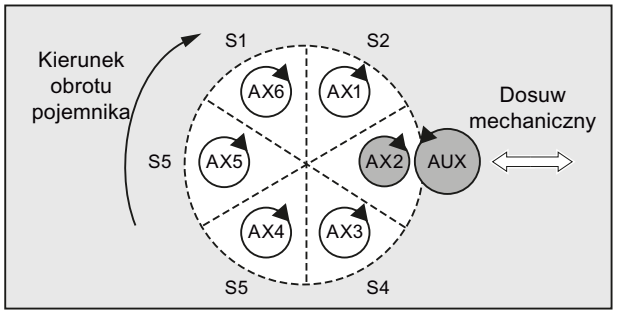
Kod programu	Komentarz
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; Włączenie stałego sprzężenia osi slave
NEWCONF	; Aktywacja zmiany danych maszynowych

Przykład 2: Dynamiczne projektowanie sprzężenia Master/Slave

Aby po obróceniu zasobnika osi mogło zostać dokonane sprzężenie z innym wrzecionem, przedtem stare sprzężenie musi zostać rozłączone, zaprojektowanie skasowane i zaprojektowane nowe sprzężenie.



Rysunek 13-1 Przed obrotem zasobnika osi.



Rysunek 13-2 Po obrocie zasobnika osi wokół slotu

Kod programu	Komentarz
MASLDEF(AUX,S3)	; AUX: Slave, S3: Master = AX3
MASLON(AUX)	; Sprzężenie wł.
M3=3 S3=4000	; Master obrócić
MASLDEL(AUX)	; Sprzężenie rozłączyć i skasować
AXCTSWE(CT1)	; Udostępnienie obrotu zasobnika osi
MASLDEF(AUX,S3)	; AUX: Slave, S3: Master = AX2

Literatura

- Podręcznik działania Funkcje specjalne; Rozdział „TE3: Sprzężenie prędkość obrotowa/ moment obrotowy, master-slave”
- Podręcznik działania Funkcje rozszerzone, rozdział „B3: Systemy decentralne - tylko 840D sl" > "NCU-Link" > „Zasobnik osi”

Akcje synchroniczne 14

14.1 Definicja akcji synchronicznej

Akcja synchroniczna jest definiowana w bloku programu obróbki. W ramach tego bloku nie wolno programować żadnych innych poleceń, które nie są częścią składową akcji synchronicznej.

Akcja synchroniczna składa się z następujących elementów:

Obowiązywanie, Nr ident. t (opcjonalnie)	Część warunkowa (opcjonalnie)			Część akcyjna		
	Częstość	Polecenie G (opcjonalnie)	Warunek	Słowo kluczowe	Polecenie G (opcjonalnie)	Akcje
--- [1] ID=<Nr> IDS=<Nr>	--- [1] WHENEVER FROM WHEN EVERY	G...	Wyrażenie logiczne	DO	G...	Akcja 1 ... Akcja n

[1] nie zaprogramowano

Składnia

```
DO Akcja 1> ... Akcja n>
<Częstość> [<Funkcja G>] <Warunek> DO Akcja 1> ... Akcja n>
ID=<Nr> <Częstość> [<Funkcja G>] <Warunek> DO Akcja 1> ... Akcja n>
IDS=<Nr> <Częstość> [<Funkcja G>] <Warunek> DO Akcja 1> ... Akcja n>
```

Literatura

Szczegółowy opis funkcjonalności akcji synchronicznych znajduje się w:

Podręcznik funkcjonowania Akcje synchroniczne

Ruch wahliwy 15

15.1 Ruch wahliwy asynchroniczny (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Oś ruchu wahadłowego wykonuje ruch między dwoma punktami nawrotu 1 i 2 z podanym posuwem w jedną i drugą stronę, aż ruch ten zostanie wyłączony.

Inne osie mogą podczas ruchu wahadłowego dowolnie interpolować. Poprzez ruch po torze albo przy pomocy osi pozycjonowania można uzyskać ciągły dosuw. Nie ma jednak przy tym **żadnego związku** między ruchem wahadłowym i ruchem dosuwu.

Właściwości asynchronicznego ruchu wahadłowego

- Asynchroniczny ruch wahliwy działa specyficznie dla osi poza granice bloku.
- Poprzez program obróbki jest zagwarantowane synchroniczne z blokiem włączenie ruchu wahliwego.
- Wspólna interpolacja wielu osi i nałożenie odcinków ruchu wahliwego są niemożliwe.

Programowanie

Poprzez następujące polecenia jest możliwe odpowiadające wykonywaniu programu NC włączanie i sterowanie asynchronicznym ruchem wahadłowym z programu obróbki.

Zaprogramowane wartości są synchronicznie do bloków zapisywane w przebiegu głównym w odpowiednich danych ustawczych i działają, aż do następnej zmiany.

Składnia

```
OSP1[<oś>]=<wartość> OSP2[<oś>]=<wartość>
OST1[<oś>]=<wartość> OST2[<oś>]=<wartość>
FA[<oś>]=<wartość>
OSCTRL[<oś>]=(<opcja ustawiona>,<opcja cofnięta>)
OSNSC[<oś>]=<wartość>
OSE[<oś>]=<wartość>
OSB[<oś>]=<wartość>
OS[<oś>]=1
OS[<oś>]=0
```

Znaczenie

<oś>:	Nazwa osi wahadłowej		
OS:	Włączenie, wyłączenie ruchu wahadłowego: OS		
	Wartość:	1	**Włączenie** ruchu wahadłowego
		0	**Wyłączenie** ruchu wahadłowego
OSP1:	Ustalenie pozycji ruchu nawrotnego 1		

OSP2:	Ustalenie pozycji ruchu nawrotnego 2 **Wskazówka:** Jeżeli jest aktywny ruch przyrostowy, wówczas pozycja jest obliczana przyrostowo względem ostatniej zaprogramowanej w programie NC, odpowiedniej pozycji nawrotu.		
OST1:	Ustalenie czasu zatrzymania w punkcie nawrotu 1 w [s]		
OST2:	Ustalenie czasu zatrzymania w punkcie nawrotu 2 w [s]		
	<wartość>:	-2	Interpolacja jest kontynuowana bez czekania na zatrzymanie dokładne
		-1	Czekanie na zatrzymanie dokładne zgrubnie
		0	Czekanie na zatrzymanie dokładne dokładnie
		>0	Czekanie na zatrzymanie dokładne dokładnie, a następnie odczekanie podanego czasu zatrzymania **Wskazówka:** Jednostka czasu zatrzymania jest identyczna z czasem zatrzymania zaprogramowanym poprzez G4.
FA:	Ustalenie prędkości posuwu Jako prędkość posuwu obowiązuje zdefiniowana prędkość posuwu osi pozycjonowania. Gdy prędkość posuwu nie jest zdefiniowana, obowiązuje wartość zapisana w danej maszynowej.		

OSCTRL:	Podanie opcji ustawienia i cofnięcia Wartości opcji 0 - 3 kodują zachowanie się w punktach nawrotu przy wyłączeniu. Można wybrać jeden z wariantów 0 - 3. Pozostałe ustawienia dają się według potrzeby kombinować z wybranym wariantem Wiele opcji jest ze sobą łączonych przy pomocy znaku plus (+).		
	<wartość>:	0	Przy wyłączeniu ruchu wahadłowego zatrzymać w najbliższym punkcie nawrotu (ustawienie domyślne) **Wskazówka:** Możliwe tylko przez cofnięcie wartości 1 i 2.
		1	Przy wyłączeniu ruchu wahadłowego zatrzymanie w punkcie nawrotu 1
		2	Przy wyłączeniu ruchu wahadłowego zatrzymanie w punkcie nawrotu 2
		3	Przy wyłączeniu ruchu wahadłowego nie ma ruchu do punktu nawrotnego, w przypadku gdy skoki wyiskrzania nie są zaprogramowane
		4	Po wyiskrzaniu ruch do pozycji końcowej
		8	Jeżeli ruch wahadłowy zostanie przerwany przez skasowanie pozostałej drogi, powinny następnie zostać wykonane skoki wyiskrzania i ew. ruch do pozycji końcowej.
		16	Jeżeli ruch wahadłowy zostanie przerwany przez skasowanie pozostałej drogi, powinien jak przy wyłączeniu nastąpić ruch do odpowiedniej pozycji nawrotu.
		32	Zmieniony posuw jest aktywny dopiero od następnego punktu nawrotu
		64	FA równe 0, FA = 0: Nałożenie drogi jest aktywne FA nierówne 0, FA <>0: Nałożenie prędkości jest aktywne
		128	W przypadku osi obrotowej DC (najkrótsza droga)
		256	Skok wyiskrzania jest wykonywany jako skok podwójny.(standard) 1=skok wyiskrzania jest wykonywany jako skok pojedynczy.
		512	Najpierw wykonać ruch do pozycji startowej
OSNSC:	Ustalenie liczby skoków wyiskrzania		
OSE:	Ustalenie pozycji końcowej (w WKS), do której ma nastąpić ruch po wyłączeniu ruchu wahliwego **Wskazówka:** Przy zaprogramowaniu OSE działa domyślnie dla „OSCTRL” opcja 4.		
OSB:	Ustalenie pozycji startowej (w WKS), do której ma nastąpić ruch przed włączeniem ruchu wahadłowego Ruch do pozycji startowej następuje przed punktem nawrotu 1. Jeżeli pozycja startowa jest zgodna z pozycją nawrotu 1, wówczas następnie następuje ruch do pozycji nawrotu 2. Przy osiągnięciu pozycji startowej nie działa czas zatrzymania, również gdy pozycja startowa jest zgodna z pozycją nawrotu 1, zamiast tego następuje czekanie na zatrzymanie dokładne dokładnie. Ustawiony warunek zatrzymania dokładnego jest dotrzymywany. **Wskazówka:** Aby nastąpił ruch do pozycji startowej, musi być ustawiona dana ustawcza SD43770 $SA_OSCILL_CTRL_MASK bit 9.		

Przykłady

Przykład 1: Oś ruchu wahadłowego ma wykonywać ruch wahadłowy między dwoma punktami nawrotu

Oś ruchu wahadłowego Z ma wykonywać ruch wahadłowy między 10 i 100. Ruch do punktu nawrotu 1 z zatrzymaniem dokładnym dokładnie, ruch do punktu nawrotu 2 z zatrzymaniem dokładnym zgrubnie. Posuw dla osi ruchu wahadłowego ma wynosić 250. Na końcu obróbki mają nastąpić 3 skoki wyiskrzania i oś ma dojść do pozycji końcowej 200. Posuw dla osi dosuwu ma wynosić 1, koniec dosuwu w kierunku X ma być osiągnięty przy pozycji 15.

Kod programu	Komentarz
WAITP(X,Y,Z)	; Położenie wyjściowe.
G0 X100 Y100 Z100	; Przełączenie na tryb osi pozycjonowania.
WAITP(X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; Punkt nawrotu 1, punkt nawrotu 2.
OSE[Z]=200	; Pozycja końcowa.
OST1[Z]=0 OST2[Z]=-1	: Czas zatrzymania na U1: Zatrzymanie dokładne dokładnie
	: Czas zatrzymania na U2: Zatrzymanie dokładne zgrubnie
FA[Z]=250 FA[X]=1	; Posuw w osi ruchu wahadłowego, posuw w osi dosuwu.
OSCTRL[Z]=(4,0)	; Opcje ustawiania.
OSNSC[Z]=3	; 3 skoki wyiskrzania.
OS[Z]=1	; Wystartowanie ruchu wahadłowego.
WHEN $A_IN[3]==TRUE DO DELDTG(X)	; Skasowanie pozostałej drogi.
POS[X]=15	; Położenia wyjściowe oś X
POS[X]=50	; Położenie końcowe oś X.
OS[Z]=0	; Zatrzymanie ruchu wahadłowego.
M30	

Uwaga

Sekwencję poleceń „OSP1[Z]=...” do „OSNCS[Z]=...” można również zaprogramować w jednym bloku.

Przykład 2: Ruch wahadłowy ze zmianą online pozycji nawrotu

Dane ustawcze wymagane dla asynchronicznego ruchu wahadłowego mogą zostać ustawione w programie obróbki.

Gdy dane ustawcze zostaną zapisane bezpośrednio w programie obróbki, wówczas zmiana działa już w chwili przebiegu wyprzedzającego. Synchroniczne zachowanie się może zostać osiągnięte przez zatrzymanie przebiegu wyprzedzającego (STOPRE).

Kod programu	Komentarz
$SA_OSCILL_REVERSE_POS1[Z]=-10	
$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	

Kod programu	Komentarz
WAITP(Z)	
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0	; Gdy wartość rzeczywista osi ruchu wahadłowego przekroczyła punkt nawrotny, oś dosuwu jest zatrzymywana.
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; Włączenie ruchu wahadłowego.
OS[Z]=0	; Wyłączenie ruchu wahadłowego.
M30	

Dodatkowe informacje

Oś ruchu wahadłowego

Dla ruchu wahadłowego obowiązuje:

- Każda oś może zostać użyta jako oś ruchu wahadłowego.
- Równocześnie może być aktywnych wiele osi ruchu wahadłowego (maksymalnie: liczba osi pozycjonowania).
- Dla osi ruchu wahadłowego jest zawsze - niezależnie od polecenia G aktualnie obowiązującego w programie - aktywna interpolacja liniowa `G1`.

Oś ruchu wahadłowego może:

- być osią wejściową dla transformacji dynamicznej
- być osią wiodącą w przypadku osi gantry i osi nadążnych
- wykonywać ruch:
  - bez ograniczenia przyspieszenia drugiego stopnia „BRISK”
    lub
  - z ograniczeniem przyspieszenia drugiego stopnia „SOFT”
    lub
  - z załamaną charakterystyką przyspieszenia (jak osie pozycjonowania)

Punkty nawrotu ruchu wahadłowego

Przy ustalaniu pozycji ruchu wahadłowego należy uwzględnić aktualne przesunięcia:

- Podanie absolutne
  „OSP1[Z]=<wartość>"
  Pozycja punkt nawrotny = suma przesunięć + wartość zaprogramowana
- Podanie względne
  „OSP1[Z]=IC(<wartość>)"
  Pozycja punktu nawrotnego = punkt nawrotny 1 + zaprogramowana wartość

Przykład:

Kod programu
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]=IC(3)

WAITP

Gdy ruch wahadłowy ma być wykonywany przy pomocy osi geometrycznej, musi ona przy pomocy „WAITP” zostać zwolniona dla tego ruchu.

Po zakończonym ruchu wahadłowym oś ruchu wahadłowego jest przy pomocy WAITP ponownie wpisywana jako oś pozycjonowania i może być ponownie normalnie stosowana.

Ruch wahadłowy z akcjami synchronicznymi ruchu i czasami zatrzymania

Po upływie ustawionych czasów zatrzymania następuje przy ruchu wahadłowym wewnętrzna zmiana bloku (widać po nowych pozostałych drogach w osi). Przy zmianie bloku jest sprawdzana funkcja wyłączenia. Funkcja wyłączenia jest przy tym ustalana odpowiednio do ustawień sterujących przebiegiem ruchu (`OSCTRL`). *Na to zachowanie się pod względem czasu można wpływać przez korekcję posuwu.*

Ewentualnie jest następnie wykonywany jeszcze jeden skok ruchu wahadłowego, zanim zostaną wystartowane skoki wyiskrzania albo nastąpi ruch do pozycji końcowej. *Powstaje przy tym wrażenie, że zmienia się zachowanie się pod względem wyłączenia. Tak jednak nie jest.*

15.2 Ruch wahliwy sterowany przez akcje synchroniczne (OSCILL)

Przy tego rodzaju ruchu wahadłowego, ruch dosuwu jest dozwolony tylko w punktach nawrotu lub w ramach zdefiniowanych obszarów nawrotu.

Zależnie od wymogu ruch wahadłowy może podczas dosuwu

- być kontynuowany lub
- zostać zatrzymany, aż dosuw zostanie kompletnie wykonany.

Składnia

1. **Ustalenie parametrów ruchu wahadłowego**
2. **Zdefiniowanie akcji synchronicznych ruchu**
3. **Przyporządkowanie osi, ustalenie dosuwu**

Znaczenie

`OSP1[<oś ruchu wahadłowego>]=`	Pozycja punktu nawrotu 1
`OSP2[<oś ruchu wahadłowego>]=`	Pozycja punktu nawrotu 2
`OST1[<oś ruchu wahadłowego>]=`	Czas zatrzymania w punkcie nawrotu 1 w sekundach
`OST2[<oś ruchu wahadłowego>]=`	Czas zatrzymania w punkcie nawrotu 2 w sekundach
`FA[<oś ruchu wahadłowego>]=`	Posuw w osi ruchu wahadłowego
`OSCTRL[<oś ruchu wahadłowego>]=`	Opcje ustawienia wzgl. cofnięcia
`OSNSC[<oś ruchu wahadłowego>]=…`	Liczba skoków wyiskrzania
`OSE[<oś ruchu wahadłowego>]=`	Pozycja końcowa
`WAITP(<oś ruchu wahadłowego>)`	Zezwolenie na ruch wahadłowy osi

Przyporządkowanie osi, dosuw

```
OSCILL[<oś ruchu wahadłowego>]=(<oś dosuwu 1>,<oś dosuwu 2>,<oś dosuwu 3>)
POSP[<oś dosuwu>]=(<pozycja końcowa>,<długość częściowa>,<tryb>)
```

`OSCILL`:	Przyporządkowanie osi dosuwu do osi ruchu wahadłowego
`POSP`:	Ustalenie dosuwów całkowitych i częściowych (patrz rozdział "Zarządzanie plikami i programami")
`Pozycja końcowa`:	Pozycja końcowa dla osi dosuwu, po tym jak wszystkie dosuwy są wykonane.

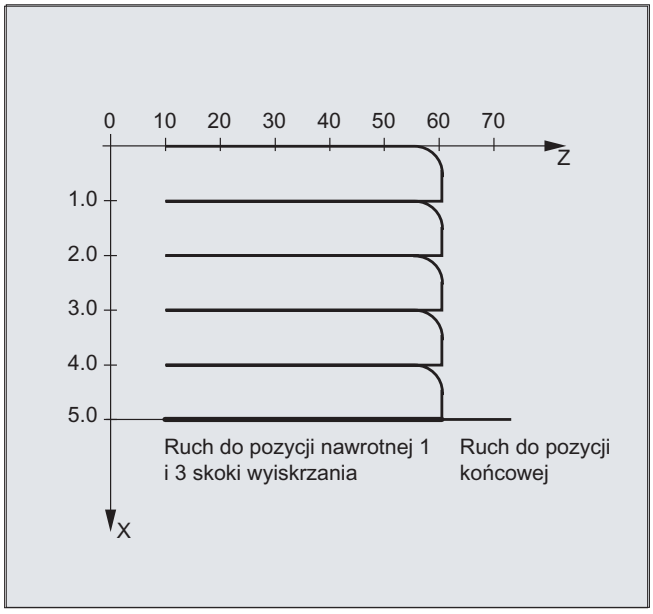
`Długość częściowa:`	Wielkość dosuwu częściowego w punkcie nawrotnym / obszarze nawrotu
`Tryb:`	Podział dosuwu całkowitego na dosuwy częściowe = dwa skrawy resztowe o takiej samej wielkości (ustawienie domyślne); = wszystkie dosuwy częściowe o takiej samej wielkości

Akcje synchroniczne ruchu

`WHEN… … DO`	jeżeli..., wówczas...
`WHENEVER … DO`	zawsze gdy..., wówczas...

Przykład

W punkcie nawrotnym 1 dosuw nie powinien następować. W punkcie nawrotnym 2 dosuw powinien następować już w odstępie ii2 przed tym punktem 2 a oś ruchu wahadłowego nie powinna w punkcie nawrotnym czekać na zakończenie dosuwu częściowego. Oś Z jest osią ruchu wahadłowego, a oś X osią dosuwu.



1. parametr dla ruchu wahadłowego

Kod programu	**Komentarz**
`DEF INT ii2`	`; Zdefiniowanie zmiennej dla obrębu nawrotu 2`
`OSP1[Z]=10 OSP2[Z]=60`	`; Zdefiniowanie punktu nawrotu 1 i 2`
`OST1[Z]=0 OST2[Z]=0`	`; Punkt nawrotu 1: Zatrzymanie dokładne dokładnie` `Punkt nawrotu 2: Zatrzymanie dokładne dokładnie`
`FA[Z]=150 FA[X]=0,5`	`; Posuw oś ruchu wahadłowego Z, posuw oś dosuwu X`
`OSCTRL[Z]=(2+8+16,1)`	`; Wyłączenie ruchu wahadłowego w punkcie nawrotu 2; po RWL wyiskrzanie i ruch do pozycji końcowej; po RWL ruch do odpowiedniej pozycji nawrotu`
`OSNC[Z]=3`	`; Skoki wyiskrzania`

Kod programu	Komentarz
OSE[Z]=70	; Pozycja końcowa = 70
ii2=2	; Ustawienie obszaru nawrotu
WAITP(Z)	Zezwól na ruch wahadłowy osi Z

2. Akcja synchroniczna ruchu

Kod programu	Komentarz
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z] DO -> $AA_OVR[X]=0 $AC_MARKER[0]=0	; Zawsze, gdy aktualna pozycja osi ruchu wahadłowego Z w MKS jest mniejsza niż początek obszaru nawrotu 2, to osiową korekcję osi dosuwu X ustawia na 0%, a znacznik o indeksie 0 na wartość 0.
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0	; Zawsze gdy aktualna pozycja osi ruchu wahadłowego Z w MKS jest większa albo równa pozycji nawrotu 2, to osiową korekcję osi ruchu wahadłowego Z ustawia na 0%.
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1	; Zawsze gdy pozostała droga dosuwu częściowego jest równa 0, to znacznik o indeksie 0 ustawia na wartość 1.
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100	; Zawsze gdy znacznik o indeksie 0 jest równy 1, to osiową korekcję osi dosuwu X ustawia na 0%. Przez to zapobiega się zbyt wczesnemu dosuwowi (oś ruchu wahadłowego Z jeszcze nie opuściła obszaru nawrotu 2, oś dosuwu X jest jednak gotowa do ponownego dosuwu). Ustaw osiowy korektor osi ruchu wahadłowego Z z 0% (akcja 2. akcji synchronicznej) w celu wykonania ruchu ponownie na 100%.

-> musi zostać zaprogramowane w jednym bloku

3. Uruchomienie ruchu wahadłowego

Kod programu	Komentarz
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; Wystartowanie osi Do osi ruchu wahadłowego Z jest przyporządkowywana oś X jako oś dosuwu. Oś X powinna do pozycji 5 wykonywać ruch krokami po 1.
M30	; Koniec programu

Dodatkowe informacje

1. **Ustalenie parametrów ruchu wahadłowego**
   Przed blokiem ruchu, który zawiera przyporządkowanie osi dosuwu i osi ruchu wahadłowego, jak też ustalenie dosuwu, należy ustalić parametry ruchu wahadłowego (patrz "Asynchroniczny ruch wahadłowy").
2. **Ustalenie akcji synchronicznych ruchu**
   Poprzez warunki synchronizacji następuje:
   Zatrzymanie dosuwu, aż oś ruchu wahadłowego znajdzie się w ramach obszaru nawrotu (ii1, ii2) albo w punkcie nawrotu (U1, U2).
   Zatrzymanie ruchu wahadłowego podczas dosuwu w punkcie nawrotu.
   Ponowne uruchomienie ruchu wahadłowego po zakończonym dosuwie częściowym.
   Ustalenie **startu następnego dosuwu częściowego**.
3. **Przyporządkowanie osi ruchu wahadłowego i osi dosuwu** jak też ustalenie **dosuwu całkowitego i częściowego**.

Ustalenie parametrów ruchu wahadłowego

Przyporządkowanie osi ruchu wahadłowego i osi dosuwu: OSCILL

```
OSCILL[<oś ruchu wahadł.>]=(<oś dosuwu1>,<oś dosuwu2>,<oś dosuwu3>)
```

Przy pomocy polecenia „OSCILL” następują przyporządkowania osi i uruchomienie ruchu wahadłowego.

Maksymalnie do jednej osi ruchu wahadłowego mogą zostać przyporządkowane 3 osie dosuwu.

Uwaga

Przed uruchomieniem ruchu wahadłowego muszą być ustalone warunki synchronizacji dla zachowania się osi.

Ustalenie dosuwów: POSP

```
POSP[<oś dosuwu>]=(<pozycja końcowa>,<długość częściowa>,<tryb>)
```

Przy pomocy polecenia „POSP” sterowanie jest informowane o:

- Dosuw całkowity (przez pozycję końcową)
- Wielkość każdorazowego dosuwu częściowego w punkcie nawrotu lub w obszarze nawrotu
- Zachowanie się pod względem dosuwu częściowego przy osiągnięciu pozycji końcowej (przez tryb)

Tryb = 0	Dla obydwu ostatnich dosuwów częściowych następuje podział pozostałej drogi do punktu docelowego na 2 pozostałe kroki o takiej samej wielkości (ustawienie domyślne).
Tryb = 1	Wszystkie dosuwy częściowe mają taką samą wielkość. Są one obliczane z dosuwu całkowitego.

Ustalenie akcji synchronicznych ruchu

Wymienione poniżej akcje synchroniczne ruchu są najogólniej używane do wykonywania ruchu wahadłowego.

Są tutaj przykładowe rozwiązania dla poszczególnych wymogów, które posłużą jako moduły do sporządzania specyficznych dla użytkownika ruchów wahadłowych.

Uwaga

W indywidualnym przypadku warunki synchronizacji mogą być również inaczej zaprogramowane.

Słowa kluczowe

`WHEN … DO …`	jeżeli..., wówczas...
`WHENEVER … DO`	zawsze gdy..., wówczas...

Funkcje

Przy pomocy poniżej opisanych szczegółowo środków językowych możecie zrealizować następujące funkcje:

1. Dosuw w punkcie nawrotu.
2. Dosuw w obszarze nawrotu.
3. Dosuw w obydwu punktach nawrotu
4. Zatrzymanie ruchu wahadłowego w punkcie nawrotu.
5. Ponownie uruchomienie ruchu wahadłowego.
6. Niezbyt wczesny start dosuwu częściowego.

Dla wszystkich akcji synchronicznych przedstawionych tutaj jako przykład obowiązują założenia:

- Punkt nawrotu 1 < punkt nawrotu 2
- Z = oś ruchu wahadłowego
- X = oś dosuwu

Uwaga

Dla bliższych objaśnień patrz punkt "Akcje synchroniczne ruchu".

Przyporządkowanie osi ruchu wahliwego i osi dosuwu, jak też ustalenie dosuwu całkowitego i częściowego

Dosuw w obszarze nawrotu

Ruch dosuwu powinien rozpoczynać się w ramach obszaru nawrotu, zanim punkt nawrotu zostanie osiągnięty.

Te akcje synchroniczne zapobiegają ruchowi dosuwu, aż oś ruchu wahliwego znajdzie się w obszarze nawrotu.

Przy danych założeniach (patrz wyżej) wynikają następujące instrukcje:

Obszar nawrotu 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X]=0
```

Zawsze gdy aktualna pozycja osi ruchu wahadłowego w MKS jest większa, niż początek obszaru nawrotu 1, ustaw korekcję osi dosuwu na 0%.

Obszar nawrotu 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X]=0
```

Zawsze gdy aktualna pozycja osi ruchu wahadłowego w MKS jest mniejsza, niż początek obszaru nawrotu 2, ustaw korekcję osi dosuwu na 0%.

Dosuw w punkcie nawrotu

Dopóki oś ruchu wahadłowego nie doszła do punktu nawrotu, nie następuje żaden ruch osi dosuwu.

Przy danych założeniach (patrz wyżej) wynikają następujące instrukcje:

Obszar nawrotu 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Zawsze gdy aktualna pozycja osi ruchu wahadłowego Z w MKS jest większa albo mniejsza, niż pozycja punktu nawrotu 1, ustaw korekcję osi dosuwu X na 0% i korekcję osi ruchu wahadłowego Z na 100%.

Obszar nawrotu 2:

Dla punktu nawrotu2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Zawsze gdy aktualna pozycja osi ruchu wahadłowego Z w MKS jest większa albo mniejsza, niż pozycja punktu nawrotu 2, ustaw korekcję osi dosuwu X na 0% i korekcję osi ruchu wahadłowego Z na 100%.

Zatrzymanie ruchu wahadłowego w punkcie nawrotu

Oś ruchu wahadłowego jest zatrzymywana w punkcie nawrotu, równocześnie rozpoczyna się ruch dosuwu . Ruch wahadłowy jest kontynuowany, gdy ruch dosuwu jest całkowicie wykonany.

Równocześnie ta akcja synchroniczna może zostać użyta do tego, by uruchomić ruch dosuwu, w przypadku gdy został on zatrzymany przez poprzednią akcję synchroniczną, która jeszcze działa.

Przy danych założeniach (patrz wyżej) wynikają następujące instrukcje:

Obszar nawrotu 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Zawsze gdy aktualna pozycja osi ruchu wahadłowego w MKS jest równa pozycji nawrotu 1, ustaw korekcję osi ruchu wahadłowego na 0% i korekcję osi dosuwu na 100%

Obszar nawrotu 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Zawsze, gdy aktualna pozycja osi ruchu wahadłowego Z w MKS jest równa pozycji nawrotu 2, ustaw korekcję osi ruchu wahadłowego X na 0% i korekcję osi dosuwu na 100%.

Ewaluacja online punktu nawrotu

Jeżeli po prawej stronie porównania znajduje się zmienna przebiegu głównego oznaczona przez `$$`, wówczas obydwie zmienne są na bieżąco ewaluowane w takcie IPO i porównywane ze sobą.

Uwaga

Więcej informacji na ten temat patrz punkt "Akcje synchroniczne ruchu".

Ponownie uruchomienie ruchu wahadłowego

Ta akcja synchroniczna ruchu jest używana do tego, by kontynuować ruch osi ruchu wahadłowego, gdy ruch dosuwu częściowego jest zakończony.

Przy danych założeniach (patrz wyżej) wynikają następujące instrukcje:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]=100
```

Zawsze, gdy pozostała droga dla dosuwu częściowego w osi dosuwu X w WKS jest równa zero, ustaw korekcję osi ruchu wahadłowego na 100%.

Następny dosuw częściowy

Po dokonanym dosuwie musi zostać uniemożliwiony zbyt wczesny start następnego dosuwu częściowego.

W tym celu jest stosowany specyficzny dla kanału znacznik (`$AC_MARKER[indeks]`), który jest ustawiany na końcu dosuwu częściowego (częściowa pozostała droga ≡ 0) i kasowany przy opuszczaniu obszaru nawrotu. Następnie przy pomocy akcji synchronicznej następuje uniemożliwienie następnego ruchu dosuwu.

Przy danych założeniach (patrz wyżej) wynikają np. dla punktu nawrotu 1 następujące instrukcje:

1. Ustawienie znacznika:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1]=1
```

Zawsze gdy pozostała droga dla dosuwu częściowego w osi dosuwu X w WKS jest równa zero, ustaw znacznik o indeksie 1 na 1.

2. Skasowanie znacznika

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1]=0
```

Zawsze, gdy aktualna pozycja osi ruchu wahadłowego Z w MKS jest większa, albo mniejsza niż pozycja punktu nawrotu 1, ustaw znacznik 1 na 0.

3. Uniemożliwienie dosuwu

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

Zawsze gdy znacznik 1 jest taki sam, ustaw korekcję osi dosuwu X na 0%.

Tłoczenie i cięcie 16

16.1 Aktywacja/Dezaktywacja

16.1.1 Włączenie/Wyłączenie tłoczenie i cięcia (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Aktywacja/Dezaktywacja tłoczenia i cięcia

Przy pomocy PON i SON jest aktywowana funkcja tłoczenia lub cięcia. SPOF powoduje zakończenie wszystkich funkcji specyficznych dla tłoczenia i cięcia. Modalnie działające polecenia PON i SON wykluczają się wzajemnie, tzn., PON powoduje wyłączenie aktywności SON i na odwrót.

Tłoczenie/cięcie z wstępnym naprężeniem

Funkcje SONS i PONS również włączają funkcje tłoczenie lub cięcia.

W przeciwieństwie do działającego w przypadku SON/PON sterowanie skokiem na płaszczyźnie interpolacji, w przypadku tych funkcji następuje sygnałowo-techniczne sterowanie wyzwalaniem skoku na płaszczyźnie serwo. Dzięki temu można pracować z większymi częstotliwościami skoków, a przez to z wyższą wydajnością wykrawania.

Podczas oceny sygnału w wstępnym naprężeniu są zablokowane wszystkie funkcje, które prowadzą do zmiany zmiany pozycji osi cięcia albo tłoczenia (np. ruch kółkiem ręcznym, zmiana frame przez PLC, funkcje pomiarowe).

Tłoczenie ze zwłoką

PDELAYON powoduje opóźnione wyprowadzenie skoku tłoczenia. Modalnie działające polecenie spełnia funkcję przygotowawczą i dlatego z reguły znajduje się przed PON. Po PDELAYOF następuje normalne dalsze tłoczenie.

Uwaga

Czas zwłoki jest ustawiany w danej ustawczej SD42400 $SC_PUNCH_DWELLTIME.

Przyśpieszenie zależne od drogi

Przy pomocy PUNCHACC można ustalić charakterystykę przyspieszenia, która zależnie od odstępu otworów definiuje różne przyspieszenia.

Drugi interfejs tłoczenia

Maszyny, które mają na przemian używać drugiego interfejsu tłoczenia (druga jednostka tłoczenia lub podobne urządzenie), można przełączyć na drugą parę szybkich cyfrowych wejść i wyjść sterowania (para I/O). Dla obydwu interfejsów tłoczenia są dostępne wszystkie funkcje

tłoczenia/cięcia. Przełączanie między pierwszym i drugim interfejsem tłoczenia następuje przez polecenia `SPIF1` i `SPIF2`.

Uwaga

Warunek: Poprzez dane maszynowe musi być zdefiniowana druga para I/O dla funkcji tłoczenia (→ patrz dane producenta maszyny!).

Składnia

```
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
```

Znaczenie

`PON`:	Aktywacja tłoczenia	
`SON`:	Aktywacja cięcia	
`PONS`:	Aktywacja tłoczenia z rekordem początkowym	
`SONS`:	Aktywacja cięcia z rekordem początkowym	
`SPOF`:	Dezaktywacja tłoczenia/cięcia	
`PDELAYON`:	Aktywacja tłoczenia ze zwłoką	
`PDELAYOF`:	Dezaktywacją tłoczenia ze zwłoką	
`PUNCHACC`:	Aktywacja przyspieszenia zależnego od drogi Parametr:	
	`<Smin>`	Najmniejszy odstęp otworów
	`<Amin>`	Przyspieszenie początkowe `<Amin>` może być większe, niż `<Amax>`.
	`<Smax>`	Największy odstęp otworów
	`<Amax>`	Przyspieszenie końcowe `<Amax>` może być większe, niż `<Amin>`.
`SPIF1`:	Uaktywnienie **pierwszego** interfejsu tłoczenia Sterowanie skokami następuje przez pierwszą parę szybkich I/O.	
`SPIF2`:	Uaktywnienie **drugiego** interfejsu tłoczenia Sterowanie skokami następuje przez drugą parę szybkich I/O.	
	Wskazówka: Po RESET albo rozruchu sterowania jest aktywny pierwszy interfejs tłoczenia. Gdy jest używany tylko jeden interfejs tłoczenia, nie musi być on programowany.	

Przykłady

Przykład 1: Aktywacja cięcia

Kod programu	Komentarz
...	
N70 X50 SPOF	; Pozycjonowanie bez wyzwolenia tłoczenia.
N80 X100 SON	; Uaktywnienie cięcia, wyzwolenie skoku przed ruchem (X=50) i na końcu zaprogramowanego ruchu (X=100).
...	

Przykład 2: Tłoczenie ze zwłoką

Kod programu	Komentarz
...	
N170 PDELAYON X100 SPOF	; Pozycjonowanie bez wyzwolenia tłoczenia, uaktywnienie zwłoki wyzwalania tłoczenia.
N180 X800 PON	; Aktywacja tłoczenia Po dojściu do pozycji końcowej skok tłoczenia jest wyprowadzany ze zwłoką.
N190 PDELAYOF X700	; Wyłączenie aktywności tłoczenia ze zwłoką, normalne wyzwolenie tłoczenia na końcu zaprogramowanego ruchu.
...	

Przykład 3: Tłoczenie z dwoma interfejsami stanu

Kod programu	Komentarz
...	
N170 SPIF1 X100 PON	; Na końcu bloku następuje wyzwolenie skoku na pierwszym szybkim wyjściu. Sygnał "skok aktywny" jest nadzorowany na pierwszym wejściu.
N180 X800 SPIF2	; Drugie wyzwolenie skoku następuje na drugim szybkim wyjściu. Sygnał "skok aktywny" jest nadzorowany na drugim wejściu.
N190 SPIF1 X700	; Sterowanie wszystkimi dalszymi skokami następuje przy pomocy pierwszego interfejsu.
...	

Dodatkowe informacje

Tłoczenie i cięcie z wstępnym naprężeniem (PONS/SONS)

Tłoczenie i cięcie z wstępnym naprężeniem nie jest możliwe równocześnie w wielu kanałach. PONS wzgl. SONS można uaktywnić tylko w jednym kanale.

Przyspieszenie zależne od drogi (PUNCHACC)

Przykład:
PUNCHACC(2,50,10,100)

Odstępy otworów poniżej 2 mm:

Ruch odbywa się z przyspieszeniem 50% przyspieszenia maksymalnego.

Odstępy otworów od 2 mm do 10 mm:

Przyspieszenie rośnie do 100% proporcjonalnie do odstępu.

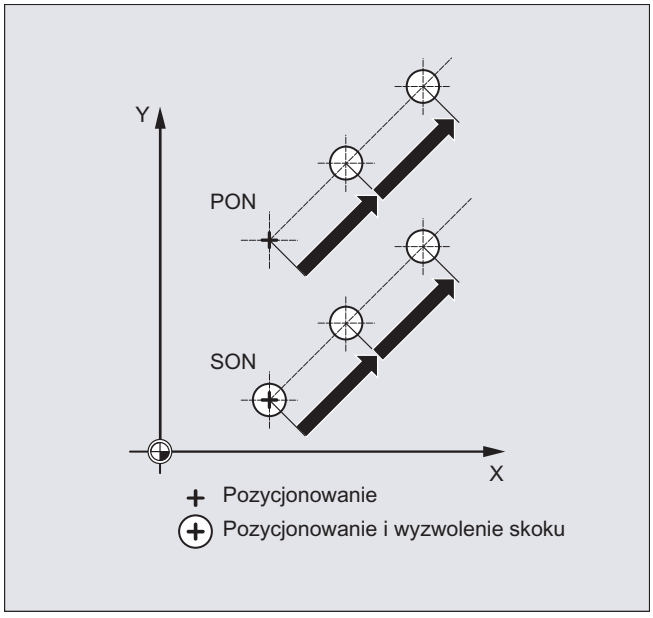
Odstępy otworów większe, niż 10 mm:

Ruch z przyspieszeniem 100%.

Wyzwolenie pierwszego skoku

Wyzwolenie pierwszego skoku po uaktywnieniu funkcji następuje w przypadku cięcia i tłoczenia różnie pod względem czasowym:

- `PON/PONS`:
  - Wszystkie skoki – również skok w pierwszym bloku po uaktywnieniu – następują na końcu bloku.
- `SON/SONS`:
  - Pierwszy skok po uaktywnieniu cięcia następuje, już na początku bloku.
  - Wszystkie dalsze skoki są wyzwalane każdorazowo na końcu bloku.



Tłoczenie i cięcie na miejscu

Wyzwolenie skoku następuje tylko wtedy, gdy blok zawiera informację o ruchu dla osi tłoczenia albo cięcia (osie aktywnej płaszczyzny).

Aby mimo to wyzwolić skok w tym samym miejscu, programuje się jedną z osi tłoczenia/cięcia z drogą ruchu 0.

Praca z użyciem narzędzi obrotowych

Uwaga

Aby narzędzia obrotowe wprowadzić stycznie na zaprogramowany tor, należy stosować sterowanie styczne.

Zastosowanie poleceń M

Przy pomocy techniki makr jest jak dotychczas możliwe używanie specjalnych funkcji M zamiast poleceń językowych (kompatybilność). Przy tym obowiązują następujące zgodności ze starymi systemami:

```
M20, M23   ≙  SPOF
M22        ≙  SON
M25        ≙  PON
M26        ≙  PDELAYON
```

Przykład pliku makr:

Kod programu	**Komentarz**
`DEFINE M25 AS PON`	`; Tłoczenie wł.`
`DEFINE M125 AS PONS`	`; Tłoczenie z wstępnym naprężeniem wł.`
`DEFINE M22 AS SON`	`; Cięcie wł.`
`DEFINE M122 AS SONS`	`; Cięcie z wstępnym naprężeniem wł.`
`DEFINE M26 AS PDELAYON`	`; Tłoczenie ze zwłoką wł.`
`DEFINE M20 AS SPOF`	`; Tłoczenie, cięcie wył.`
`DEFINE M23 AS SPOF`	`; Tłoczenie, cięcie wył.`

Przykład programowania:

Kod programu	**Komentarz**
`...`	
`N100 X100 M20`	`; Pozycjonowanie bez wyzwolenia tłoczenia.`
`N110 X120 M22`	`; Aktywacja cięcia, przed i po ruchu wyzwolenia skoku.`
`N120 X150 Y150 M25`	`; Aktywacja tłoczenia, wyzwolenie skoku na końcu ruchu.`
`...`	

16.2 Automatyczny podział drogi

Podział w odcinkach częściowych

Przy aktywnym tłoczeniu lub cięciu zarówno SPP, jak też SPN powodują podział całkowitego odcinka ruchu, zaprogramowanego dla osi uczestniczących w tworzeniu konturu, na pewną liczbę odcinków częściowych o takiej samej długości (równoległy podział drogi). Wewnętrznie każdy odcinek częściowy odpowiada jednemu blokowi.

Liczba skoków

Przy tłoczeniu pierwszy skok następuje w punkcie końcowym pierwszego odcinka częściowego, przy cięciu natomiast - w punkcie startowym pierwszego odcinka częściowego. Na całym odcinku ruchu wynikają przez to następujące liczby:

Tłoczenie: Liczba skoków = liczba odcinków częściowych

Cięcie: Liczba skoków = liczba odcinków częściowych + 1

Funkcje pomocnicze

Funkcje pomocnicze są wykonywane w pierwszym z utworzonych bloków.

Składnia

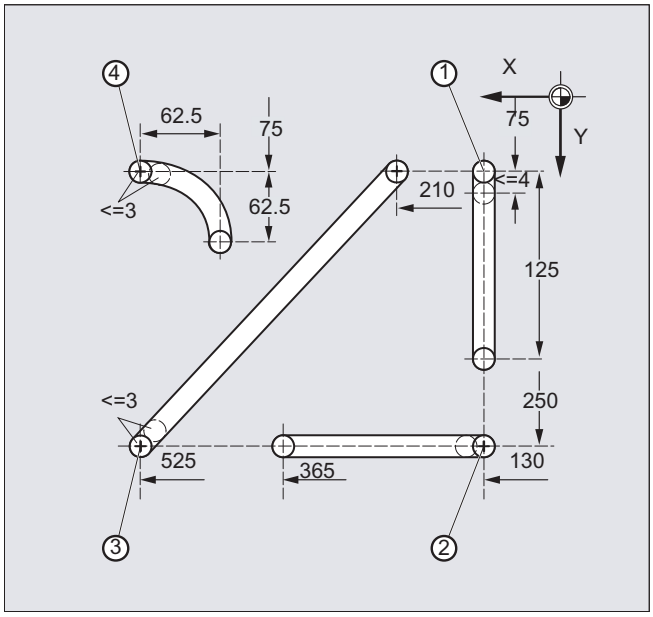
```
SPP=
SPN=
```

Znaczenie

SPP:	Wielkość odcinka częściowego (maksymalny odstęp skoków); działa modalnie
SPN:	Liczba odcinków częściowych na blok; działa pojedynczymi blokami

Przykład 1

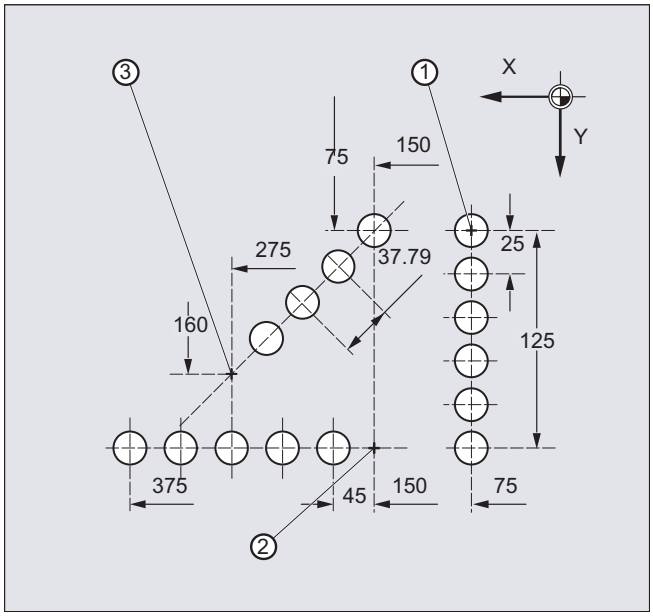
Zaprogramowane odcinki cięcia powinny automatycznie zostać podzielone na odcinki częściowe o takiej samej wielkości.



Kod programu	Komentarz
N100 G90 X130 Y75 F60 SPOF	; Pozycjonowanie na punkt startowy 1
N110 G91 Y125 SPP=4 SON	; Cięcie wł.; maksymalna długość odcinka podziałowego dla automatycznego podziału drogi: 4 mm
N120 G90 Y250 SPOF	; Cięcie wył.; pozycjonowanie na punkt startowy 2
N130 X365 SON	; Cięcie wł.; maksymalna długość odcinka podziałowego dla automatycznego podziału drogi: 4 mm
N140 X525 SPOF	; Cięcie wył.; pozycjonowanie na punkt startowy 3
N150 X210 Y75 SPP=3 SON	; Cięcie wł.; maksymalna długość odcinka podziałowego dla automatycznego podziału drogi: 3 mm
N160 X525 SPOF	; Cięcie wył.; pozycjonowanie na punkt startowy 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; Cięcie wł.; maksymalna długość odcinka podziałowego dla automatycznego podziału drogi: 3 mm
N180 G00 G90 Y300 SPOF	; Cięcie wył.

Przykład 2

Dla poszczególnych szeregów otworów powinien nastąpić automatyczny podział drogi. Dla podziału jest każdorazowo podawana maksymalna długość odcinka częściowego (wartość SPP).



Kod programu	**Komentarz**
N100 G90 X75 Y75 F60 PON	; Pozycjonowanie na punkt startowy 1; tłoczenie pojedynczego otworu
N110 G91 Y125 SPP=25	; Maksymalna długość odcinka podziałowego dla automatycznego podziału drogi: 25 mm
N120 G90 X150 SPOF	; Tłoczenie wył.; pozycjonowanie na punkt startowy 2
N130 X375 SPP=45 PON	; Tłoczenie wł.; maksymalna długość odcinka podziało- wego dla automatycznego podziału drogi: 45 mm
N140 X275 Y160 SPOF	; Tłoczenie wył.; pozycjonowanie na punkt startowy 3
N150 X150 Y75 SPP=40 PON	Tłoczenie wł.; zamiast zaprogramowanej długości odcin- ka podziałowego 40 mm jest stosowana obliczona długość odcinka 37,79 mm.
N160 G00 Y300 SPOF	; Tłoczenie wył.; pozycjonowanie

16.2.1 Podział drogi w przypadku osi uczestniczących w tworzeniu konturu

Długość odcinka częściowego SPP

Za pomocą `SPP` podaje się maksymalny odstęp skoków, a przez to maksymalną długość odcinków częściowych, na które całkowity odcinek ruchu ma zostać podzielony. Wyłączenie polecenia następuje przy pomocy `SPOF` albo `SPP=0`.

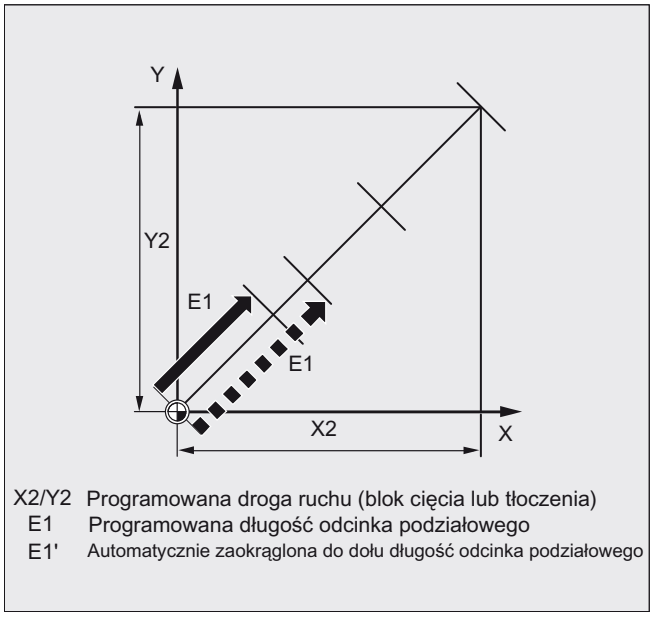
Przykład:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

Całkowita droga ruchu 10 mm jest dzielona na 5 odcinków po 2 mm (`SPP=2`).

Uwaga

Podział drogi przy pomocy `SPP` następuje zawsze równolegle: wszystkie odcinki częściowe mają taką samą długość. Oznacza to, że zaprogramowana wielkość odcinka częściowego (wartość `SPP`) obowiązuje tylko wtedy, gdy iloraz drogi całkowitej i wartości SPP jest liczbą całkowitą. Jeżeli tak nie jest, wówczas wielkość odcinka częściowego jest tak redukowana, że jest uzyskiwany iloraz całkowitoliczbowy.



X2/Y2 Programowana droga ruchu (blok cięcia lub tłoczenia)
E1 Programowana długość odcinka podziałowego
E1' Automatycznie zaokrąglona do dołu długość odcinka podziałowego

Przykład:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

W przypadku drogi całkowitej 15 mm i odcinka częściowego 3,5 mm uzyskuje się iloraz nie całkowitoliczbowy (4.28). Przez to następuje zmniejszenie wartości SPP do najbliższego możliwego ilorazu całkowitoliczbowego. W tym przypadku uzyskuje się długość odcinka częściowego wynoszącą 3 mm.

Liczba odcinków częściowych SPN

Przy pomocy `SPN` definiuje się liczbę odcinków częściowych, która ma zostać utworzona z całkowitej drogi ruchu. Długość odcinków częściowych jest obliczana automatycznie. Ponieważ `SPN` działa pojedynczymi blokami, musi przedtem przy pomocy `PON` albo `SON` zostać uaktywnione tłoczenie albo cięcie.

16.2.2 Podział drogi w przypadku pojedynczych osi

Jeżeli oprócz osi uczestniczących w tworzeniu konturu również pojedyncze osie są zdefiniowane jako oś tłoczenia-cięcia, wówczas również one mogą podlegać automatycznemu podziałowi drogi.

Zachowanie się pojedynczej osi w przypadku SPP

Zaprogramowana długość odcinka częściowego (`SPP`) odnosi się w zasadzie do osi uczestniczących w tworzeniu konturu. Dlatego w bloku, w którym oprócz ruchu pojedynczej osi i wartości SPP nie jest zaprogramowana żadna oś uczestnicząca w tworzeniu konturu, wartość SPP jest ignorowana.

Jeżeli w bloku jest zaprogramowana zarówno pojedyncza oś, jak też oś uczestnicząca w tworzeniu konturu, wówczas zachowanie się pojedynczej osi zależy od ustawienia odpowiedniej danej maszynowej.

1. Ustawienie standardowe
   Droga pojedynczej osi jest dzielona równomiernie na bloki pośrednie wytwarzane przez `SPP`.

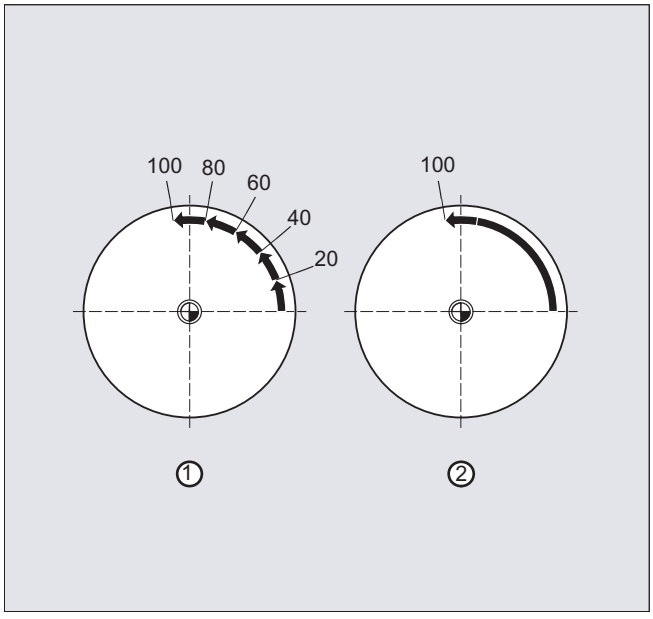
Przykład:

```
N10 G1 SON X10 A0

N20 SPP=3 X25 A100
```

Poprzez odcinek skoku 3 mm jest przy całkowitej drodze ruchu osi X (oś uczestnicząca w tworzeniu konturu) wynoszącej 15 mm wytwarzanych 5 bloków.

Oś A obraca się przez to w każdym bloku o 20°.



1. Pojedyncza oś bez podziału drogi
   Pojedyncza oś przebywa swoją całkowitą drogę w pierwszym z wytworzonych bloków.
2. Różny podział drogi
   Zachowanie się pojedynczej osi jest zależne od interpolacji osi uczestniczących w tworzeniu konturu:

- Interpolacja kołowa: Podział drogi
- Interpolacja liniowa: Bez podziału drogi

Zachowanie się w przypadku SPN

Zaprogramowana liczba odcinków częściowych obowiązuje również, gdy nie jest równocześnie zaprogramowana oś uczestnicząca w tworzeniu konturu.

Warunek: pojedyncza oś jest zdefiniowana jako oś tłoczenia-cięcia.

Szlifowanie 17

17.1 Włączenie/wyłączenie specyficznego dla szlifowania nadzoru nad narzędziem (TMON, TMOF)

Za pomocą wstępnie zdefiniowanych procedur TMON...) i TMOF...) włączany lub wyłączany jest specyficzny dla szlifowania nadzór nad narzędziem (geometria i nadzór nad prędkością obrotową).

Warunek

Specyficzne parametry narzędzia $TC_TPG1 do $TC_TPG9 muszą być ustawione.

Składnia

```
TMON(<TNr>)
...
TMOF(<TNr>)
```

Znaczenie

`TMON(...)`:	Włączenie specyficznego dla szlifowania nadzoru nad narzędziem Polecenie musi być programowane w tym kanale, w którym specyficzny dla szlifowania nadzór nad narzędziem ma być włączony.
`TMOF(...)`:	Wyłączenie specyficznego dla szlifowania nadzoru nad narzędziem Polecenie musi być programowane w tym kanale, w którym specyficzny dla szlifowania nadzór nad narzędziem ma być wyłączony.
`<TNr>`:	numer T **Uwaga:** Podanie jest wymagane tylko wtedy, gdy zamiast dla aktywnego, pracującego narzędzia nadzór ma zostać włączony lub wyłączony dla nieaktywnej ściernicy.
`TMOF(0)`:	Wyłączenie nadzoru nad wszystkimi narzędziami

Dalsze funkcje 18

18.1 Funkcje osi (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

„AXNAME” jest stosowane np. przy sporządzaniu ogólnie obowiązujących cykli, gdy nazwy osi nie są znane.

„AX” jest stosowane do pośredniego programowania osi geometrycznych i synchronicznych. Identyfikator osi jest przy tym zapisywany w zmiennej typu AXIS lub dawany przez polecenie jak „AXNAME” albo „SPI”.

"SPI" jest stosowane, gdy są programowane funkcje osi dla wrzeciona, np. wrzeciona synchronicznego.

"AXTOSPI" jest stosowane, aby zamienić identyfikator osi na indeks wrzeciona (funkcja odwrotna do "SPI").

"AXSTRING" jest stosowane, aby zamienić identyfikator osi (typ danych AXIS) na łańcuch znaków (funkcja odwrotna do "AXNAME").

"ISAXIS" jest stosowane w powszechnie obowiązujących cyklach, aby zapewnić, że określona oś geometryczna występuje, a przez to następne wywołanie $P_AXNX nie zostanie przerwane z błędem.

"MODAXVAL" jest stosowane, aby w przypadku osi obrotowych modulo określić pozycję modulo.

Składnia

```
AXNAME("String")
AX[AXNAME("String")]
SPI(n)

AXTOSPI(A) albo AXTOSPI(B) albo AXTOSPI(C)
AXSTRING(SPI(n))
ISAXIS(<numer osi geometrycznej>)
<pozycja modulo>=MODAXVAL(<oś>,<pozycja osi>)
```

Znaczenie

AXNAME:	Konwertuje wejściowy ciąg znaków na identyfikator osi; wejściowy ciąg znaków musi zawierać obowiązującą nazwę osi.
AX:	Zmienny identyfikator osi
SPI:	Konwertuje numer wrzeciona na identyfikator osi; przekazany parametr musi zawierać poprawny numer wrzeciona.
n:	Numer wrzeciona
AXTOSPI:	Konwertuje identyfikator osi na indeks wrzeciona typu Integer. "AXTOSPI" odpowiada funkcji odwrotnej do "SPI".
X, Y, Z:	Identyfikator osi typu AXIS jako zmienna albo stała

AXSTRING:	Jest wyprowadzany łańcuch znaków z przynależnym numerem wrzeciona.
ISAXIS:	Sprawdza, czy istnieje podana oś geometryczna.
MODAXVAL:	W przypadku osi obrotowych modulo określa pozycję modulo; odpowiada ona ona reszcie modulo względem sparametryzowanego zakresu modulo (wynosi w ustawieniu standardowym 0 do 360 stopni; przez MD30340 MODULO_RANGE_START i MD30330 $MA_MODULO_RANGE można zmienić początek i wielkość zakresu modulo).

Uwaga

Rozszerzenia SPI

Funkcja osi SPI(n) może być teraz stosowana również do odczytu i zapisu komponentów frame. Dzięki temu mogą być zapisywane frame np. o składni
`$P_PFRAME[SPI(1),TR]=2.22`.

Przez dodatkowe zaprogramowanie pozycji osi przez adres `AX[SPI(1)]=<pozycja osi>` może zostać wykonany ruch w osi. Warunkiem jest, by wrzeciono znajdowało się w trybie pozycjonowania albo pracy jako oś.

Przykłady

Przykład 1: AXNAME, AX, ISAXIS

Kod programu	Komentarz
`OVRA[AXNAME("oś poprzeczna")]=10`	`; Korekcja osi poprzecznej`
`AX[AXNAME("oś poprzeczna")]=50.2`	`; Pozycja końcowa dla osi poprzecznej`
`OVRA[SPI(1)]=70`	`; Korekcja wrzeciona 1`
`AX[SPI(1)]=180`	`; Pozycja końcowa dla wrzeciona 1`
`IF ISAXIS(1) == FALSE GOTOF DALEJ`	`; Czy jest odcięta?`
`AX[$P_AXN1]=100`	`; Ruch w odciętej wykonany`
`DALEJ:`	

Przykład 2: AXSTRING

Przy programowaniu przy pomocy AXSTRING[SPI(n)] nie jest już jako numer wrzeciona wyprowadzany indeks osi, której wrzeciono jest przyporządkowane, lecz łańcuch znaków "Sn".

Kod programu	Komentarz
`AXSTRING[SPI(2)]`	`; Jest wyprowadzany łańcuch znaków "S2".`

Przykład 3: MODAXVAL

Ma zostać określona pozycja modulo osi obrotowej modulo A.

Wartością wyjściową dla obliczenia jest pozycja osi 372.55.

Sparametryzowany zakres modulo wynosi 0 do 360 stopni:

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

Kod programu	Komentarz
R10=MODAXVAL(A,372.55)	; Obliczona pozycja modulo R10 = 12.55.

Przykład 4: MODAXVAL

Jeżeli programowany identyfikator osi nie odnosi się do osi obrotowej modulo, wówczas będąca do konwersji wartość (<pozycja osi>) jest zwracana bez zmian.

Kod programu	Komentarz
R11=MODAXVAL(X,372.55)	; X jest osią liniowa; R11 = 372.55.

18.2 Przełączalne osie geometryczne (GEOAX)

Przy pomocy funkcji "Przełączane osie geometryczne" osie geometryczne sparametryzowane poprzez dane maszynowe można zastąpić innymi osiami kanałowymi.

Składnia

```
GEOAX(<n>,<oś kanałowa>,<n>,<oś kanałowa>,<n>,<oś kanałowa>)
GEOAX()
```

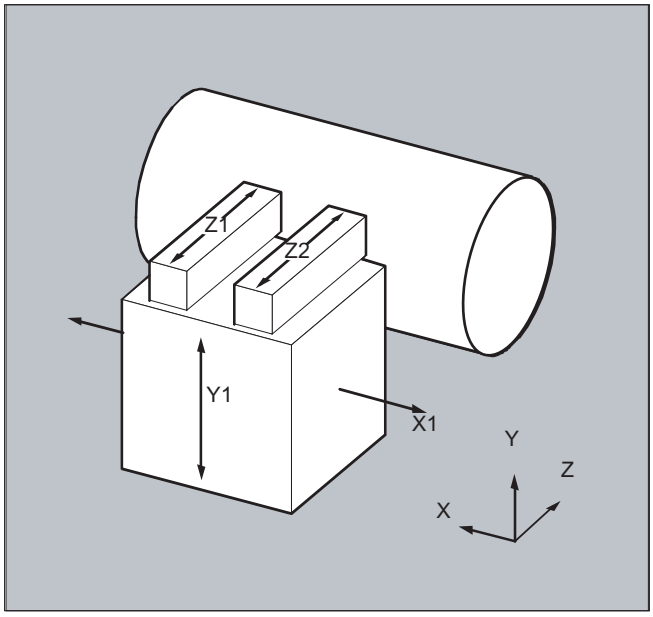
Znaczenie

`GEOAX(...)`	Funkcja do przełączania osi geometrycznych. **Wskazówka**: `GEOAX)` bez podania parametrów ponownie uaktywnia sparametryzowaną w danych maszynowych podstawową konfigurację osi geometrycznych.
`<n>`	Numer osi geometrycznej, która jest zastępowana przez podaną oś kanałową. Zakres wartości: 0, 1, 2, 3 **Wskazówka**: 0: Podana oś kanałowa jest bez jej zastąpienia usuwana z zespołu osi geometrycznych. 1: 1. oś geometryczna ≙Oś współrzędnych X (odcięta) WKS 2: 2. oś geometryczna ≙Oś współrzędnych Y (rzędna) WKS 3: 3. oś geometryczna ≙Oś współrzędnych Z (aplikata) WKS
`<oś kanałowa>`	Nazwa osi kanałowej, która ma zostać ujęta w zespole osi geometrycznych.

Przykłady

Przykład 1: Włączanie dwóch osi na przemian jako oś geometryczna

Ruchy saniami narzędziowymi mogą być wykonywane przez osie kanałowe X1, Y1, Z1, Z2:



Osie geometryczne są tak zaprojektowane, że po włączeniu najpierw działa Z1 jako 3. oś geometryczna pod nazwą "Z" i razem z X1 i Y1 tworzy zespół osi geometrycznych.

W programie obróbki teraz osie Z1 i Z2 mają na przemian być używane jako oś geometryczna Z:

Kod programu	Komentarz
...	
N100 GEOAX(3,Z2)	; Jako 3. oś geometryczna (Z) działa oś kanałowa Z2.
N110 G1 ...	
N120 GEOAX(3,Z1)	; Jako 3. oś geometryczna (Z) działa oś kanałowa Z1.
...	

Przykład 2: Przełączanie osi geometrycznych przy 6 osiach kanału

Maszyna posiada 6 osi kanałowych o nazwach XX, YY, ZZ, U, V, W.

Podstawowe ustawienie konfiguracji osi geometrycznych przez dane maszynowe jest następujące:

Oś kanałowa XX = 1. oś geometryczna (oś X)

Oś kanałowa YY = 2. oś geometryczna (oś Y)

Oś kanałowa ZZ = 3. oś geometryczna (oś Z)

Kod programu	Komentarz
N10 GEOAX()	; Konfiguracja podstawowa osi geometrycznych działa.
N20 G0 X0 Y0 Z0 U0 V0 W0	; Wszystkie osie przesuwem szybkim do pozycji 0.
N30 GEOAX(1,U,2,V,3,W)	Oś kanałowa U staje się pierwszą (X), V drugą (Y)
	; a W trzecią osią geometryczną (Z).
N40 GEOAX(1,XX,3,ZZ)	; Oś kanałowa XX staje się pierwszą (X), ZZ trzecią
	; osią geometryczną (Z). Oś kanału V pozostaje drugą
	; osią geometryczną (Y).

Kod programu	**Komentarz**
N50 G17 G2 X20 I10 F1000	; Okrąg w płaszczyźnie X/Y. Ruch wykonują
	; osie kanałowe XX i V.
N60 GEOAX(2,W)	; Oś kanałowa W staje się drugą osią geometryczną (Y).
N80 G17 G2 X20 I10 F1000	; Okrąg w płaszczyźnie X/Y. Ruch wykonują
	; osie kanałowe XX i W.
N90 GEOAX()	; Cofnięcie do stanu podstawowego.
N100 GEOAX(1,U,2,V,3,W)	; Oś kanałowa U staje się pierwszą (X), V drugą
	; (Y) a W trzecią osią geometryczną (Z).
N110 G1 X10 Y10 Z10 XX=25	; Osie kanałowe U, V, W wykonują każdorazowo ruch do
	; pozycji 10. XX jako oś dodatkowa wykonuje ruch do pozycji 25.
N120 GEOAX(0,V)	; V jest wyłączana z zespołu osi geometrycznych.
	; U i W są nadal pierwszą (X) i trzecią
	; osią geometryczną (Z).
	; Druga oś geometryczna (Y) pozostaje nie zajęta.
N130 GEOAX(1,U,2,V,3,W)	; Oś kanałowa U pozostaje pierwszą (X), V staje się
	; drugą (Y), W pozostaje trzecią osią geometryczną (Z).
N140 GEOAX(3,V)	; V staje się trzecią osią geometryczną (Z), przy czym W
	; jest zastępowana, a przez to usuwana z zespołu
	; osi geometrycznych. Druga oś geometryczna (Y)
	; jest jak dotychczas nie zajęta.

Dane maszynowe

Konfiguracja osi

Przyporządkowanie osi geometrycznych, dodatkowych i osi maszyny do osi kanałowych:

- MD10000 $MN_AXCONF_MACHAX_NAME_TAB
- MD20050 $MC_AXCONF_GEOAX_ASIGN_TAB
- MD20060 $MC_AXCONF_GEOAX_NAME_TAB
- MD20070 $MC_AXCONF_MACHAX_USED
- MD20080 $MC_AXCONF_CHANAX_NAME_TAB
- MD35000 $MA_SPIND_ASSIGN_TO_MACHAX

Zachowanie się przy reset

Zachowanie się przy zresetowaniu zmienionych przyporządkowań osi geometrycznych:

- MD20110 $MC_RESET_MODE_MASK, Bit 12
- MD20118 $MC_GEOAX_CHANGE_RESET

Zachowanie się przy NC-Start

- MD20112 $MC_START_MODE_MASK, Bit 12

Informacja dla programu użytkownika PLC

Możliwość parametryzacji polecenia M, które przy przełączeniu osi geometrycznych jest wyprowadzane na złącze NC/PLC:

- MD22532 $MC_GEOAX_CHANGE_M_CODE

Warunki brzegowe

Bez przełączenia osi geometrycznych

- Jeżeli jest aktywna jedna z następujących funkcji, przełączenie osi geometrycznych jest niemożliwe:
  - Transformacja
  - Interpolacja spline
  - Korekcja promienia narzędzia
  - Korekcja dokładna narzędzia
- Oś geometryczna i jeszcze jedna oś kanałowa mają taką samą nazwę.
- Jedna z osi uczestniczących w przełączeniu osi geometrycznych uczestniczy w akcji, która wykracza poza granice bloku. Np. oś pozycjonowania wykraczająca poza zakres bloku albo oś nadążna sprzężenia osi.

Osie obrotowe

Osie obrotowe nie mogą być czynione osiami geometrycznymi.

Stan osi po zastąpieniu

Oś zastąpiona przez przełączenie w zespole osi geometrycznych jest po procesie przełączenia poprzez swoją nazwę osi kanałowej możliwa do programowania jako oś dodatkowa.

Frame, obszary ochrony, ograniczenia obszaru pracy

Z przełączeniem osi geometrycznych są kasowane wszystkie frame, obszary ochrony i ograniczenia obszaru pracy.

Współrzędne biegunowe

Zamiana osi geometrycznych przy pomocy `GEOAX` ustawia analogicznie do zmiany płaszczyzny przy pomocy `G17-G19` modalne współrzędne biegunowe na wartość 0.

DRF, NPV

Ewentualne przesunięcie kółkiem ręcznym (DRF) albo zewnętrzne przesunięcie punktu zerowego (NPV) nadal działa po przełączeniu.

Konfiguracja podstawowa osi geometrycznych

Polecenie `GEOAX ()` wywołuje podstawową konfigurację zespołu osi geometrycznych.

Po POWER ON i przy przełączeniu na tryb pracy "bazowanie do punktu odniesienia" następuje automatycznie przełączenie z powrotem na konfigurację podstawową.

Korekcja długości narzędzia

Aktywna korekcja długości narzędzia działa również po procesie przełączenia. Obowiązuje ona jednak dla nowo przyjętych albo zmienionych pod względem pozycji osi geometrycznych jako jeszcze nie zrealizowana. Przy pierwszym poleceniu ruchu dla tej osi geometrycznej wynikowa droga ruchu składa się w związku z tym z sumy korekcji długości narzędzia i zaprogramowanej drogi ruchu.

Osie geometryczne, które przy przełączeniu zachowują swoją pozycję w zespole osi, zachowują również swój status odnośnie korekcji długości narzędzia.

Konfiguracja osi geometrycznych przy aktywnej transformacji

- Konfiguracja osi geometrycznych, sparametryzowana poprzez dane maszynowe transformacji, nie daje się zmienić poprzez funkcję "przełączane osie geometryczne".
- Przy różniącej się konfiguracji osi geometrycznych dla transformacji muszą w danych maszynowych transformacji zostać sparametryzowane różniące się zestawy danych.
- Zmieniona przez `GEOAX` konfiguracja osi geometrycznych jest kasowana przez uaktywnienie transformacji.
- Odnośnie osi geometrycznych specyficzne dla transformacji parametryzacje osi geometrycznych aktywnej transformacji mają pierwszeństwo przed parametryzacjami dotyczącymi przełączenia osi geometrycznych.
  Przykład: Transformacja jest aktywna. Wg danych maszynowych przy resecie kanału transformacja ma zostać zachowana. Równocześnie jednak w przypadku resetu kanału powinna zostać stworzona podstawowa konfiguracja osi geometrycznych. Pozostaje zachowana konfiguracja osi geometrycznych, która jest ustalona dla transformacji.
- Z wyłączeniem transformacji ponownie będzie działać sparametryzowane ustawienie podstawowe konfiguracji osi geometrycznych.

Tryb pracy JOG, funkcja maszynowa REF

Przy przełączeniu na tryb pracy JOG, funkcja maszynowa REF (najazd na punk referencyjny), będzie działać konfiguracja osi geometrycznych sparametryzowana w danych maszynowych.

18.3 Pojemnik osi (AXCTSWE, AXCTSWED, AXCTSWEC)

Poprzez polecenia "AXCTSWE" wzgl. "AXCTSWED" jest zwalniany obrót podanego zasobnika osi.

Poprzez polecenie "AXCTSWEC" można cofnąć już udzielone zezwolenie na obrót zasobnika osi.

Składnia

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

Znaczenie

AXCTSWE:	Zezwolenie na obrót zasobnika osi Wykonywanie programu nie jest zatrzymywane przez "AXCTSWE". Obrót następuje, gdy tylko wszystkie kanały uczestniczące w zasobniku osi udzieliły zezwolenia.	
AXCTSWED:	Zezwolenie na obrót zasobnika osi bez uwzględnienia innych kanałów uczestniczących w zasobniku. **Wskazówka** • Wariant polecenia do uproszczenia uruchamiania programu obróbki wzgl. akcji synchronicznej. • Zachowanie się odnośnie innych kanałów uczestniczących w zasobniku osi można zadać przez: MD12760 $MN_ AXCT_FUNCTION_MASK, Bit 0	
AXCTSWEC:	Cofnięcie zezwolenia na obrót zasobnika osi **Wskazówka** Zezwolenie na obrót zasobnika osi może zostać cofnięte tylko wtedy, gdy obrót jeszcze nie rozpoczął się: $AN_AXCTSWA[zasobnik osi>] == 0 Na temat zmiennej systemowej patrz "Pojemnik osi (AXCTSWE, AXCTSWED, AXCTSWEC) (Strona 669)"	
<ID>:	Identyfikator zasobnika osi albo osi pojemnikowej:	
	CT<numer>:	Identyfikator domyślny zasobnika osi: MD12750 $MN_AXCT_NAME_TAB Przykład: "CT1"
	zasobnik:	Specyficzny dla użytkownika identyfikator zasobnika osi: MD12750 $MN_AXCT_NAME_TAB Przykład: "POJEMNIK_1"
	<oś>:	Identyfikator osi zasobnika znanej w kanale

Uwaga

Długość kroku

Wielkość kroku obrotu zasobnika osi jest ustawiana poprzez daną ustawczą:

SD41700 $SN_AXCT_SWWIDTH

Dodatkowe informacje

Diagnoza

Aktualny status zasobnika osi można przeczytać poprzez następującą zmienną systemową:

Zmienna systemowa	Typ	Opis
$AC_AXCTSWA[<nazwa>]	BOOL	Specyficzny dla kanału status zasobnika osi
$AN_AXCTSWA[zasobnik osi>]	BOOL	Specyficzny dla NCU status zasobnika osi
$AN_AXCTSWE[zasobnik osi>]	INT	Specyficzny dla slotu status obrotu zasobnika osi Zmienna systemowa daje **bitami** status slotów zasobnika osi. Każdy bit odpowiada jednemu slotowi.
$AN_AXCTAS[zasobnik osi>]	INT	Liczba miejsc (slotów) o które zasobnik osi został aktualnie przełączony.

Obrót zasobnika osi z domyślnym GET / GETD

Poprzez poniższą daną maszynową można ustawić, że poleceniem "AXCTSWE" wszystkie osie zasobnika kanału przy pomocy domyślnego "GET / GETD" są przywoływane do kanału. Zamiana osi jest możliwa dopiero po dokonanym obrocie zasobnika.

MD10722 $MN_AXCHANGE_MASK, Bit 1 = 1

Uwaga

Obrót zasobnika osi przy pomocy domniemanego "GET / GETD" **nie** jest przeprowadzany dla osi w stanie "oś przebiegu głównego" (np. oś PLC), ponieważ oś do obrotu zasobnika musiałaby opuścić ten stan.

18.4 Czekanie na obowiązującą pozycję osi (WAITENC)

Przy pomocy polecenia językowego "WAITENC" można w programie NC czekać, aż dla osi zaprojektowanych przy pomocy MD34800 $MA_WAIT_ENC_VALID = 1 będą do dyspozycji synchronizowane wzgl. restaurowane pozycje.

W stanie czekania może nastąpić przerwanie, np. przez start ASUP lub przez zmianę trybu pracy na JOG. Z kontynuowaniem programu jest ew. znów przyjmowany stan czekania.

Uwaga

Stan czekania jest wyświetlany na interfejsie graficznym przez stan zatrzymania "Czekanie na system pomiarowy".

Składnia

"WAITENC" można programować w części programowej dowolnego programu NC.

Programowanie musi nastąpić w oddzielnym bloku:

```
...
WAITENC
...
```

Przykład

"WAITENC" jest stosowane np. w sterowanym zdarzeniami programie użytkownika .../_N_CMA_DIR/_N_PROG_EVENT_SPF, jak pokazuje to poniższy przykład zastosowania.

Przykład zastosowania: Wycofanie narzędzia po POWER OFF z transformacją orientacji

Obróbka z orientacją narzędzia została przerwana przez zanik napięcia.
Przy następnym rozruchu jest wywoływany sterowany zdarzeniami program użytkownika .../_N_CMA_DIR/_N_PROG_EVENT_SPF.

W sterowanym zdarzeniami programie użytkownika następuje przy pomocy "WAITENC" czekanie na synchronizowane wzgl. restaurowane pozycje osi, aby następnie móc obliczyć frame, który ustawia WKS w kierunku narzędzia.

Kod programu	Komentarz
...	
IF $P_PROG_EVENT == 4	; Rozruch.
IF $P_TRAFO <> 0	; Transformacja została wybrana.
WAITENC	; Czekanie na obowiązujące pozycje osi orientacji.
TOROTZ	; Obrót osi Z WKS w kierunku osi narzędzia.
ENDIF	
M17	
ENDIF	
...	

Następnie narzędzie może zostać odsunięte w trybie pracy JOG przez ruch wycofania w kierunku osi narzędzia.

18.5 Programowalne przełączenie zestawu parametrów (SCPARA)

Przy pomocy polecenia „SCPARA” można dla osi zażądać przełączenia na określony zestaw parametrów.

Uwaga

Bez przełączenia parametrów podczas obróbki gwintu

Przy nacinaniu gwintu G33 i gwintowaniu otworu G331 / G332 zestaw parametrów jest wybierany przez sterowanie i nie można go zmienić.

Zablokowane przełączenie zestawu parametrów

Przełączenia zestawu parametrów można zażądać również poprzez interfejs NC/PLC. Aby uniknąć konfliktów przy przełączaniu, można poprzez interfejs NC/PLC zablokować przełączenie parametrów NC (SCPARA):

DB31, ... DBX9.3 (zadanie zestawu parametrów zablokowane przez NC))

Uwaga

Gdy zostanie przez „SCPARA” zażądane przełączenie zestawu parametrów, podczas gdy jest zablokowane przez interfejs NC/PLC, przełączenie nie jest akceptowane bez wyprowadzenia komunikatu błędu.

Składnia

```
SCPARA [<oś>]= <wartość>
```

Znaczenie

SCPARA:	Polecenie: Przełącz zestaw parametrów	
<oś>:	Identyfikator osi (oś kanału)	
	Typ:	AXIS
<wartość>:	Numer zestawu parametrów: 1, 2, 3, ... max numer zestawu parametrów	

Przykład

Kod programu	Komentarz
...	
N110 SCPARA[X]= 3	; Wybór: Oś X, 3. zestaw parametrów
...	

Dodatkowe informacje

Zezwolenie na przełączenie zestawu parametrów

Przełączenie zestawu parametrów osi musi być bezpośrednio zwolnione:

MD35590 $MA_PARAMSET_CHANGE_ENABLE[<oś>]

Odczyt zestawu parametrów

Numer wybranego zestawu parametrów (zadany zestaw parametrów) można przeczytać przez zmienną systemową $AA_SCPAR.

Literatura

Szczegółowe informacje dot. zestawów parametrów znajdują się w:

Podręcznik działania Funkcje podstawowe; Rozdział "Prędkości, systemy wartości zadanej/ rzeczywistej, Regulacja (G2)" > "Regulacja" > "Zestawy parametrów regulatora położenia"

18.6 Sprawdzenie występującego zakresu językowego NC (STRINGIS)

Przy pomocy funkcji "STRINGIS(...)" można sprawdzić, czy podany łańcuch znaków jest dostępny jako element języka programowania NC w aktualnym zakresie językowym.

Definicja

```
INT STRINGIS(STRING <nazwa>)
```

Składnia

```
STRINGIS(<nazwa>)
```

Znaczenie

`STRINGIS`:	Funkcja z wartością zwrotną
`<nazwa>`:	Nazwa sprawdzanego elementu języka programowania NC
Wartość zwrotna:	Formatem wartości zwrotnej jest yxx (dziesiętny).

Elementy języka programowania NC

Następujące elementy języka programowania NC mogą zostać sprawdzone:

- Polecenia G wszystkich istniejących grup G, np. "G0", "INVCW", "POLY", "ROT", "KONT", "SOFT", "CUT2D", "CDON", "RMBBL", "SPATH"
- Adresy DIN lub NC jak np. "ADIS", "RNDM", "SPN", "SR", "MEAS"
- Funkcje np. "TANG(...)" lub "GETMDACT"
- Procedury np. "SBLOF".
- Słowa kluczowe np. "ACN", "DEFINE" lub "SETMS"
- Dane systemowe np. dane maszynowe $M... , dane ustawcze $S... albo dane opcji $O...
- Zmienne systemowe $A... , $V... , $P...
- Parametry obliczeniowe R...
- Nazwy uaktywnionych cykli
- Zmienne GUD i LUD
- Nazwy makr
- Nazwy etykiet

Wartość zwrotna

Wartość zwrotna ma znaczenie tylko w pierwszych 3 miejscach dziesiętnych. Formatem wartości zwrotnej jest yxx, gdzie y = informacja bazowa, a xx = informacja szczegółowa.

Wartość zwrotna	Znaczenie
000	Łańcuch znaków ’name’ nie jest znany w niniejszym systemie [1)]
100	Łańcuch znaków ’name’ jest elementem języka programowania NC, ale jego programowanie jest aktualnie niemożliwe (opcja/funkcja jest nieaktywna)

Wartość zwrotna	Znaczenie	
2xx	Łańcuch znaków 'name' jest programowalnym elementem języka programowania NC (opcja/funkcja jest aktywna). Informacja szczegółowa xx zawiera dalsze informacje o rodzaju elementu:	
	xx	**Znaczenie**
	01	Adres DIN lub adres NC [2)]
	02	Polecenie G (np. G04, INVCW)
	03	Funkcja z wartością zwrotną
	04	Funkcja bez wartości zwrotnej
	05	Słowo kluczowe (np. DEFINE)
	06	Dana maszynowa ($M...), ustawcza ($S...) lub dana opcji ($O...)
	07	Parametry systemowe, np. zmienne systemowe ($...) albo parametry obliczeniowe (R...)
	08	Cykl (cykl musi być załadowany w NC, a programy cykli muszą być aktywne [3)])
	09	Zmienna GUD (zmienna GUD musi być zdefiniowana w pliku definicji GUD, a zmienne GUD muszą być aktywne)
	10	Nazwa makra (makro musi być zdefiniowane w plikach definicji makr i makra muszą być aktywne) [4)]
	11	Zmienna LUD aktualnego programu obróbki
	12	ISO polecenia G (tryb językowy ISO musi być aktywny)
400	Łańcuch znaków 'name' jest adresem NC, który nie został rozpoznany jako xx == 01 albo xx == 10 i który nie jest G lub R [2)]	
y00	Specyficzne przyporządkowanie nie jest możliwe	

1) Zależnie od sterowania jest ewentualnie znany tylko podzbiór poleceń językowych NC Siemens, np. SINUMERIK 802D sl. Na tych sterowaniach jest dla łańcuchów znaków, które zasadniczo są poleceniami językowymi NC Siemens, zwracana wartość 0. To zachowanie się może zostać zmienione przez MD10711 $MN_NC_LANGUAGE_CONFIGURATION. Przy MD10711 = 1 jest wówczas dla poleceń językowych NC Siemens zawsze zwracana wartość 100.

2) Adresami NC są następujące litery: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Te adresy NC mogą być również programowane z rozszerzeniem. Rozszerzenie adresu może zostać podane przy sprawdzaniu przy pomocy STRINGIS. Przykład: 201 == STRINGIS("A1").
Litery: D, F, H, L, M, N, O, P, S, T są adresami NC albo funkcjami pomocniczymi, które są stosowane jako definiowane przez użytkownika. Dla nich jest zawsze zwracana wartość 400. Przykład: 400 == STRINGIS("D"). Te adresy NC nie mogą przy sprawdzaniu przy pomocy STRINGIS być podawane z rozszerzeniem.
Przykład: 000 == STRINGIS("M02"), ale 400 == STRINGIS("M").

3) Nazwy parametrów cykli nie mogą być sprawdzane przy pomocy STRINGIS.
4) Adresy zdefiniowane jako makro, np. G, H, M, L są identyfikowane jako makro.

Przykłady

W poniższych przykładach przyjmuje się, że element językowy NC podany jako łańcuch znaków, o ile szczególnie nie zaznaczono, zasadniczo daje się programować w sterowaniu.

1. Łańcuch znaków "T" jest definiowany jako funkcja pomocnicza:
```
400 == STRINGIS("T")
000 == STRINGIS ("T3")
```
2. Łańcuch znaków "X" jest definiowany jako oś:
```
201 == STRINGIS("X")
201 == STRINGIS("X1")
```
3. Łańcuch znaków "A2" jest definiowany jako adres NC z rozszerzeniem:
```
201 == STRINGIS("A")
201 == STRINGIS("A2")
```

4. Łańcuch znaków "INVCW" jest definiowany jak nazwane polecenie G:
   `202 == STRINGIS("INVCW")`
5. Łańcuch znaków "$MC_GCODE_RESET_VALUES" jest definiowany jako dana maszynowa:
   `206 == STRINGIS("$MC_GCODE_RESET_VALUES")`
6. Łańcuch znaków "GETMDACT" jest funkcją językową NC:
   `203 == STRINGIS("GETMDACT ")`
7. Łańcuch znaków "DEFINE" jest słowem kluczowym:
   `205 == STRINGIS("DEFINE")`
8. Łańcuch znaków "$TC_DP3" jest parametrem systemowym (komponent długości narzędzia):
   `207 == STRINGIS("$TC_DP3")`
9. Łańcuch znaków "$TC_TP4" jest parametrem systemowym (wielkość narzędzia):
   `207 == STRINGIS("$TC_TP4")`
10. Łańcuch znaków "$TC_MPP4" jest parametrem systemowym (stan miejsca w magazynie):
    - Zarządzanie magazynem narzędzi jest aktywne: `207 == STRINGIS("$TC_MPP4") ;`
    - Zarządzanie magazynem narzędzi nie jest aktywne: `000 == STRINGIS("$TC_MPP4")`

    Patrz też w punkcie: Zarządzanie magazynem narzędzi.
11. Łańcuch znaków "MACHINERY_NAME" jest definiowany jako zmienna GUD:
    `209 == STRINGIS("MACHINERY_NAME")`
12. Łańcuch znaków "LONGMACRO" jest definiowany jako makro:
    `210 == STRINGIS("LONGMACRO")`
13. Łańcuch znaków "MYVAR" jest definiowany jako zmienna LUD:
    `211 == STRINGIS("MYVAR")`
14. Łańcuch znaków "XYZ" nie jest poleceniem znanym w NC, zmienną GUD, nazwą makra albo cyklu:
    `000 == STRINGIS("XYZ")`

Zarządzanie magazynem narzędzi

Jeżeli funkcja zarządzania magazynem narzędzi nie jest aktywna, STRINGIS daje dla parametrów systemowych zarządzania magazynem narzędzi, niezależnie od danej maszynowej

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

zawsze wartość 000.

Tryb ISO

Gdy jest aktywna funkcja "Tryb ISO":

- MD18800 $MN_MM_EXTERN_LANGUAGE (aktywowanie zewnętrznych języków NC)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (system sterowania do zaadaptowania)

STRINGIS najpierw sprawdza podany łańcuch znaków jako polecenie G SINUMERIK. Jeżeli łańcuch znaku nie jest poleceniem G SINUMERIK, następuje sprawdzenie go jako polecenia G ISO.

Programowane przełączenia (`G290` (SINUMERIK Mode), `G291` (ISO Mode)) nie mają żadnego wpływu na STRINGIS.

Przykład

Dane maszynowe mające znaczenie dla funkcji STRINGIS(...) mają następujące wartości:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (Jako znane są traktowane tylko te polecenia NC, których opcje są ustawione)
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0' (opcja: Transformacje)
- MD10700 $MN_PREPROCESSING_LEVEL='H43' (przetwarzanie wstępne cykli aktywne)

Następujący program przykładowy jest wykonywany bez komunikatu błędu:

Kod programu	**Komentarz**
`N1 R1=STRINGIS("TRACYL")`	`; R1 == 0, ponieważ TRACYL z powodu brakującej opcji transformacji jest rozpoznawany jako „nieznany"`
`N2 IF STRINGIS("TRACYL") == 204`	
`N3  TRACYL(1,2,3)`	`; N3 jest pomijane`
`N4 ELSE`	
`N5  G00`	`; i zamiast tego wykonywanie N5`
`N6 ENDIF`	
`N7 M30`	

18.7 Interaktywne wywołanie okna z programu obróbki (MMC)

Poprzez wstępnie zdefiniowany podprogram MMC(...), wychodzące z programu NC okna dialogowe specyficzne dla użytkownika mogą być wyświetlana na interfejsie obsługi np. SINUMERIK Operate.

Projektowanie dialogów może następować za pomocą następujących rodzajów dialogów:

- Run MyScreens
- Easy XML
- Użytkownik XML

Literatura

- Podręcznik programowania Run MyScreens
- Podręcznik programowania Easy XML

Składnia

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

Znaczenie

MMC(...):	identyfikator podprogramu Parametry są podawane jako kodowane na stałe miejsca i oddzielane za pomocą przecinka w obrębie dwóch łańcuchów danych - łańcuchem rozkazowym a łańcuchem kwitowania.	
Parametry w obrębie łańcucha rozkazowego:		
<ADDRESS>:	Obszar obsługi, w którym są wykonywane zaprojektowane okna dialogowe użytkownika.	
	Funkcja	**Obszary obsługi**
	Okno dialogowe użytkownika "Run MyScreens"	CYCLES
	Okno dialogowe użytkownika "Easy XML"	CYCLES
	Użytkownik XML	XML
	Wyskakujące okno "Run MyScreens"	POPUPDLG
	Wyskakujące okno "Easy XML"	POPUPDLG
<COMMAND>:	Wykonywany rozkaz	
	Funkcja	**Rozkazy**
	Okno dialogowe użytkownika "Run MyScreens"	PICTURE_ON, PICTURE_OFF
	Okno dialogowe użytkownika "Easy XML"	PICTURE_ON, PICTURE_OFF
	Użytkownik XML	XML_ON, XML_OFF
	Wyskakujące okno "Run MyScreens"	PICTURE_ON, PICTURE_OFF
	Wyskakujące okno "Easy XML"	PICTURE_ON, PICTURE_OFF

<FILE>:	Nazwa pliku, w którym jest zaprogramowane okno dialogowe	
	Funkcja	**Pliki**
	Okno dialogowe użytkownika "Run MyScreens"	<name>.com
	Okno dialogowe użytkownika "Easy XML"	<name>.xml
	Użytkownik XML	<name>.xml
	Wyskakujące okno "Run MyScreens"	<name>.com
	Wyskakujące okno "Easy XML"	<name>.xml
	Wyskakujące okno "Easy XML" z projektowanie bezpośrednio w programie NC (patrz przykład 2)	xmldial_emb.xml
<DIALOG>:	Nazwa wyświetlanego okna dialogowego	
	Funkcja	**Nazwa okna dialogowego**
	Wszystkie funkcje za wyjątkiem wyskakującego okna "Easy XML" z projektowaniem bezpośrednio w programie NC	W pliku <FILE> projektowana nazwa okna dialogowego
	Wyskakujące okno "Easy XML" z projektowanie bezpośrednio w programie NC (patrz przykład 3)	main
Parametry w obrębie łańcucha kwitowania:		
<QUIT>:	Typ kwitowania	
	N:	Bez kwitowania. Wykonywanie programu jest kontynuowane po wysłaniu rozkazu. Brak informacji zwrotnej, gdy rozkaz nie może być wykonany pomyślnie. **Wskazówka** Typ kwitowania „N” jest stosowany, gdy czas wyświetlania (czas oczekiwania) programowany jest w programie NC (patrz poniżej przykład 2).
	A:	Kwitowanie asynchroniczne Wykonywanie programu jest kontynuowane po wysłaniu polecenia. Wartość zwrotna jest zapisana w zdefiniowanych specyficznych dla użytkownika zmiennych kwitowania w ramach projektowania okna dialogowego (zmienne GUD) i mogą być odczytywane przez program NC.

Przykład

Przykład 1

Wyświetlenie okna dialogowego i reakcja na warunek użytkownika w programie NC.

```
Kod programu                                    Komentarz
; Zmienna kwitowania QUIT została już utworzona jako globalna zmienna użytkownika
(GUD)
; typu STRING podczas projektowania okna dialogowego:
; DEF NCK STRING[20] QUIT
QUIT = "XXX"                                   ; Inicjalizacja zmiennej kwitowania
G4 F5
```

Kod programu	**Komentarz**
MMC("CYCLES,PICTURE_ON,test.com,test1","A")	; Wyświetlenie okna dialogowego ; -Obszar obsługi: CYCLES ; - Status obrazu: PICTURE_ON (wyświetlenie) ; - Plik obraz okna dialogowego: test.com ; - Obraz okna dialogowego: test1
INPUT:	; Czekanie na warunek użytkownika
STOPRE	; Zatrzymanie przebiegu wyprzedzającego
IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK	; Przycisk "RUN"
IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK	; Przycisk "CHK"
GOTOB INPUT	; => Czekanie
WORK:	; Przycisk "RUN" naciśnięty
MSG(„Kontynuacja obróbki -> NC-Start")	; Wyprowadzenie komunikatu
MMC("CYCLES,PICTURE_OFF","N")	; Zamknięcie okna dialogowego
M0	; Czekanie na start NC
GOTOF END	; => Koniec programu
CHECK:	; Przycisk "CHK" naciśnięty
MSG(„Najazd na pozycję -> NC-Start")	; Wyprowadzenie komunikatu
MMC("CYCLES,PICTURE_OFF","N")	; Zamknięcie okna dialogowego
M0	; Czekanie na start NC
GOTOF END	; => Koniec programu
END:	
...	

Przykład 2

Czas wyświetlania okna dialogowego jest określony w programie NC, np. za pomocą czasu oczekiwania.

Kod programu	**Komentarz**
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N")	; Wyświetlenie okna dialogowego
X200	
Z40	
MMC("POPUPDLG,PICTURE_**OFF**","N")	; Zamknięcie okna dialogowego

Przykład 3

Wprowadzenie skryptu PopUp do programu NC i jego zastosowanie.

Kod programu

```
PROC POPUP_TEST
; ----------------------------- Script -----------------------------
```

Kod programu

```
; <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>
;             <op>
```

Kod programu

```
;                ypos = ypos + 24;
;                count = count + 1;
;             </op>
;          </while>
;       </paint>
;    </form>
; </main_dialog>
; ========================= Część programu============================
...
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
G4 F2
X0
...
```

Warunki brzegowe

- Pliki definicji *.com okna dialogowego należy zapisać w katalogu „proj”.
- Pliki definicji Easy XML *.xml okna dialogowego należy zapisać w katalogu „appl”. Jeżeli pliki definicji zostały zapisane w innym katalogu, podanie ścieżki musi nastąpić w sposób pośredni w oparciu o katalog „appl”.
- Okna dialogowe zdefiniowane przez użytkownika nie mogą być równocześnie wyświetlane z różnych kanałów.

18.8 Czas przebiegu programu / licznik obrabianych przedmiotów

W celu wsparcia operatora obrabiarki dostępne są informacje dot. czasu przebiegu programu i liczby obrabianych przedmiotów.

Te informacje mogą być przetwarzane jako zmienne systemowe w programie NC i/albo w PLC. Równocześnie te informacje są do dyspozycji w celu wyświetlenia na interfejsie obsługi.

18.8.1 Czas przebiegu programu

Funkcja „Czas przebiegu programu" udostępnia wewnętrzny zegar w NC do nadzoru procesów technologicznych, które mogą być odczytywane przez zmienne systemowe specyficzne dla NC i specyficzne dla kanału, w programie obróbki i w akcjach synchronicznych.

Triger do pomiaru czasu przebiegu ($AC_PROG_NET_TIME_TRIGGER) jest jedyną zapisywalną zmienną systemową funkcji i służy do selektywnego pomiaru segmentów programu. Tzn. przez zapisywanie trigera w programie NC pomiar czasu może być włączany i wyłączany.

Zmienna systemowa	Znaczenie	Aktywność
Specyficzna dla NC		
$AN_SETUP_TIME	Czas od ostatniego rozruchu sterowania z wartościami standardowymi ("start zimny") w minutach Jest przy każdym rozruchu sterowania z wartościami standardowymi automatycznie cofany na "0".	• Zawsze aktywny
$AN_POWERON_TIME	Czas od ostatniego normalnego rozruchu sterowania ("start ciepły") w minutach Jest przy każdym normalnym rozruchu sterowania automatycznie cofany na "0".	
Specyficzna dla kanału		

Zmienna systemowa	Znaczenie	Aktywność
$AC_OPERATING_TIME	Całkowity czas przebiegu programów NC w trybie pracy Automatyka w sekundach Wartość jest przy każdym rozruchu sterowania automatycznie cofana na "0".	• Uaktywnienie przez MD27860 • Tylko tryb pracy AUTOMATYKA
$AC_CYCLE_TIME	Czas przebiegu wybranego programu NC w sekundach Wartość jest ze startem nowego programu NC automatycznie cofana na "0".	
$AC_CUTTING_TIME	Czas obróbki w sekundach Mierzony jest czas ruchu osi uczestniczących w tworzeniu konturu (jest aktywna co najmniej jedna) bez aktywnego posuwu szybkiego we wszystkich programach NC między NC-Start i końcem programu / reset NC. Pomiar jest dodatkowo przerywany przy aktywnym czasie oczekiwania. Jest przy każdym rozruchu sterowania z wartościami standardowymi wartość jest automatycznie cofana na "0".	
$AC_ACT_PROG_NET_TIME	Aktualny czas przebiegu netto aktualnego programu NC w sekundach. Jest ze startem nowego programu NC automatycznie cofany na "0".	• Zawsze aktywny • Tylko tryb pracy AUTOMATYKA
$AC_OLD_PROG_NET_TIME	Czas przebiegu netto dopiero co zakończonego poprawnie programu za pomocą M30 w sekundach.	
$AC_OLD_PROG_NET_TIME_COUNT	Zmiany na $AC_OLD_PROG_NET_TIME Po POWER ON zmienna $AC_OLD_PROG_NET_TIME_COUNT jest ustawiona na "0". $AC_OLD_PROG_NET_TIME_COUNT jest zwiększana zawsze wtedy, gdy sterowanie zapisało $AC_OLD_PROG_NET_TIME na nowo.	

Zmienna systemowa	Znaczenie		Aktywność
$AC_PROG_NET_TIME_TRIGGER	Triger pomiaru czasu przebiegu:		• Tylko tryb pracy AUTOMATYKA
	0	Stan neutralny Triger nie jest aktywny.	
	1	Zakończenie Kończy pomiar i kopiuje wartość z $AC_ACT_PROG_NET_TIME do $AC_OLD_PROG_NET_TIME. AC_ACT_PROG_NET_TIME jest ustawiana na "0" i następnie mierzy dalej.	
	2	Start Uruchamia pomiar i ustawia przy tym $AC_ACT_PROG_NET_TIME na "0". $AC_OLD_PROG_NET_TIME nie jest zmieniana.	
	3	Stop Zatrzymuje pomiar. Nie zmienia $AC_OLD_PROG_NET_TIME i utrzymuje stałą $AC_ACT_PROG_NET_TIME, aż do kontynuacji.	
	4	Kontynuacja Kontynuacja pomiaru, tzn. wcześniej zatrzymany pomiar jest ponownie podejmowany. $AC_ACT_PROG_NET_TIME biegnie dalej. $AC_OLD_PROG_NET_TIME nie jest zmieniana.	
Przez POWER ON wszystkie zmienne systemowe są cofane na "0"!			

Uwaga

Producent maszyny

Włączenie uaktywnionego zegara następuje przez daną maszynową MD27860 $MC_PROCESSTIMER_MODE.

Zachowanie się aktywnych pomiarów czasu w przypadku określonych funkcji (np. GOTOS, korekcja = 0%, aktywny posuw w pracy próbnej, test programu, ASUP, PROG_EVENT, …) jest konfigurowane przez dane maszynowe MD27850 $MC_PROG_NET_TIMER_MODE i MD27860 $MC_PROCESSTIMER_MODE.

Literatura:
Podręcznik działania Funkcje podstawowe; BAG, Kanał, Praca programowa, Zachowanie się przy zresetowaniu (K1), punkt: Czas przebiegu programu

Uwaga

Pozostały czas dla obrabianego przedmiotu

Gdy kolejno obrabiane są takie same przedmioty, można za pomocą następującej wartości zegara określić czas pozostały do obróbki przedmiotu:

- czas obróbki ostatnio obrobionego przedmiotu (patrz $AC_OLD_PROG_NET_TIME)
- Aktualny czas obróbki (patrz $AC_ACT_PROG_NET_TIME)

Pozostały czas jest jest oprócz aktualnego czasu obróbki wyświetlany na otoczce graficznej.

Uwaga

Zastosowanie STOPRE

Zmienne systemowe $AC_OLD_PROG_NET_TIME i $AC_OLD_PROG_NET_TIME_COUNT nie wytwarzają domyślnego zatrzymania przebiegu wyprzedzającego. Przy zastosowaniu w programie obróbki nie stanowi to problemu, gdy wartość zmiennej systemowej pochodzi z poprzedniego przebiegu programu. Gdy jednak triger jest w celu pomiaru czasu przebiegu ($AC_PROG_NET_TIME_TRIGGER) zapisywany z wysoką częstotliwością i przez to $AC_OLD_PROG_NET_TIME bardzo często zmienia się, wówczas powinno w programie obróbki być domyślnie stosowane `STOPRE`.

Warunki brzegowe

- **Szukanie bloku**
  Przy szukaniu bloku nie są określane czasy przebiegu programu.
- **REPOS**
  Czas trwania procesu REPOS jest doliczany do aktualnego czasu obróbki ($AC_ACT_PROG_NET_TIME).

Przykłady

Przykład 1: Pomiar czasu trwania "mySubProgrammA"

Kod programu

```
...
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
```

Gdy program wykonał wiersz N80, w $AC_OLD_PROG_NET_TIME znajduje się czas przebiegu netto dla "mySubProgrammA".

Wartość $AC_OLD_PROG_NET_TIME:

- Pozostaje zachowana po M30.
- Jest aktualizowana po każdym przejściu pętli.

Przykład 2: Pomiar czasu trwania "mySubProgrammA" i "mySubProgrammC

Kod programu

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
```

Kod programu
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 Licznik obrabianych przedmiotów

Funkcja "Licznik obrabianych przedmiotów" udostępnia różne liczniki, które mogą być stosowane w szczególności do wewnętrznego w sterowaniu liczenia obrabianych przedmiotów.

Liczniki istnieją jako specyficzne dla kanału zmienne systemowe z dostępem w celu zapisu i odczytu w zakresie wartości 0 do 999 999 999.

Zmienna systemowa	Znaczenie
$AC_REQUIRED_PARTS	Liczba przedmiotów do wykonania (zadana liczba obrabianych przedmiotów) W tym liczniku można zdefiniować liczbę obrabianych przedmiotów, przy osiągnięciu której rzeczywista liczba obrobionych przedmiotów $AC_ACTUAL_PARTS jest zerowana.
$AC_TOTAL_PARTS	Całkowita liczba wykonanych przedmiotów (całkowita rzeczywista liczba przedmiotów obrabianych) Ten licznik podaje liczbę przedmiotów wykonanych od chwili startu. Wartość jest tylko przy rozruchu sterowania z wartościami standardowymi automatycznie cofana na "0".
$AC_ACTUAL_PARTS	Liczba wykonanych przedmiotów (rzeczywista liczba przedmiotów obrabianych) W tym liczniku jest rejestrowana liczba wszystkich przedmiotów wykonanych od chwili startu. Przy osiągnięciu zadanej liczby obrobionych przedmiotów ($AC_REQUIRED_PARTS) licznik jest automatycznie cofany na "0" ($AC_REQUIRED_PARTS > 0).
$AC_SPECIAL_PARTS	Liczba wykonanych przedmiotów liczonych przez użytkownika Ten licznik pozwala użytkownikowi na liczenie wykonanych przedmiotów według własnej definicji. Można zdefiniować wyprowadzenie alarmu przy osiągnięciu zadanej liczby sztuk ($AC_REQUIRED_PARTS). Zerowanie licznika użytkownik musi przeprowadzić sam.

Uwaga

Wszystkie liczniki są przy rozruchu sterowania z wartościami standardowymi ustawiane na „0” i mogą być niezależnie od ich uaktywnienia czytane/zapisywane.

Uwaga

Przez kanałowe dane maszynowe można wpływać na uaktywnienie licznika, czas zerowania i algorytm liczenia.

Uwaga

Liczenie obrabianych przedmiotów za pomocą polecenia M definiowanego przez użytkownika

Poprzez dane maszynowe można ustawić, by impulsy liczenia dla różnych liczników obrabianych przedmiotów zamiast przez koniec programu `M2`/`M30` były wyzwalane przez definiowane przez użytkownika polecenia M.

18.9 Process DataShare - Wyprowadzenie na zewnętrzne urządzenie/plik (EXTOPEN, WRITE, EXTCLOSE)

Zapis danych z programu obróbki na zewnętrzne urządzenie/plik następuje w trzech krokach:

1. Otwarcie zewnętrznego urządzenia/pliku
   Przy pomocy polecenia EXTOPEN następuje otwarcie zewnętrznego urządzenia/pliku dla kanału w celu zapisu.
2. Zapis danych
   Wyprowadzana dana może zostać przygotowana przy pomocy funkcji string języka NC (""), np. SPRINT. Sam zapis następuje poprzez polecenie WRITE.
3. Zamknięcie zewnętrznego urządzenia/pliku
   Przy pomocy polecenia EXTCLOSE albo poprzez dojście do końca programu (M30), jak też przy zresetowaniu kanału jest zwalniane zajęte w kanale zewnętrzne urządzenie/plik.

Składnia

```
DEF INT <Result>
DEF STRING[<n>] <Output>
…
EXTOPEN(<Result>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
…
<Output>="wyprowadzane dane"
WRITE(<Result>,<ExtDev>,<Output>)
…
EXTCLOSE(<Result>,<ExtDev>)
```

Znaczenie

EXTOPEN:	Wstępnie zdefiniowana procedura do otwarcia zewnętrznego urządzenia/pliku		
<Result>:	**Parametr 1:** Zmienna wynikowa Na podstawie wartości zmiennej wynikowej można ocenić w programie powodzenie operacji i odpowiednio postąpić.		
	Typ:	INT	
	Wartości:	0	Nie ma błędu
		1	Urządzenia zewnętrznego nie można otworzyć
		2	Urządzenie zewnętrzne nie jest zaprojektowane
		3	Urządzenie zewnętrzne zaprojektowane z niepoprawną ścieżką
		4	Brak praw dostępu do urządzenia zewnętrznego
		5	Tryb użycia: Urządzenie zewnętrzne już jest zajęte "na wyłączność"
		6	Tryb użycia: Urządzenie zewnętrzne już jest zajęte "do podziału"
		7	Długość pliku większa, niż LOCAL_DRIVE_MAX_FILESIZE
		8	Maksymalna liczba urządzeń zewnętrznych jest przekroczona
		9	Opcja dla LOCAL_DRIVE nie ustawiona
		11	Interfejs V.24 jest już zajęty poprzez funkcję Easy-Message (tylko 828D)
		12	Tryb zapisu: Podanie sprzeczne z extdev.ini
		16	Zaprogramowano niepoprawną ścieżkę zewnętrzna
		22	Urządzenie zewnętrzne nie zamontowane

<ExtDev>:	**Parametr 2:** Identyfikator symboliczny otwieranego zewnętrznego urządzenia/ pliku	
	Typ: STRING	
	Identyfikator symboliczny składający się z: 1. logicznej nazwy urządzenia 2. po której ew. następuje ścieżka pliku (dołączona przy pomocy "/"). Są zdefiniowane następujące **logiczne nazwy urządzeń**:	
	"LOCAL_DRIVE":	Lokalna karta CompactFlash (wstępnie zdefiniowana)
	"CYC_DRIVE":	zarezerwowane podanie stacji do zastosowania w cyklach SIEMENSA (wstępnie zdefiniowane)
	"/dev/ext/1",... "/dev/ext/9":	Dostępne dyski sieciowe **Wskazówka:** Wymagane zaprojektowanie w pliku extdev.ini!
	"/dev/cyc/1", "/dev/cyc/2":	zarezerwowane podanie dysku do zastosowania w cyklach SIEMENSA **Wskazówka:** Wymagane zaprojektowanie w pliku extdev.ini!
	"/dev/v24":	Interfejs V.24 **Wskazówka:** Wymagane zaprojektowanie w pliku extdev.ini!
	Ścieżka pliku: • Do "LOCAL_DRIVE" i "CYC_DRIVE" musi zostać podana ścieżka pliku, np.: "LOCAL_DRIVE/my_dir/my_file.txt" • Logiczne nazwy urządzeń "/dev/ext/1...9" i "/dev/cyc/1...2" mogą poprzez zaprojektowanie: – już wskazywać na plik, wówczas wolno podać tylko logiczną nazwę urządzenia, np.: "/dev/ext/4" – albo na katalog, wówczas musi zostać podana ścieżka pliku, np. "/dev/ext/5/my_dir/my_file.txt" • Do "/dev/v24" nie może być podwieszona ścieżka pliku.	
	Wskazówka: Dla logicznych nazw urządzeń "/dev/ext/1...9", "/dev/v24" i "/dev/cyc/1...2" pisanie dużymi/małymi literami jest ignorowane, przy podaniu ścieżki do pliku pisanie dużymi/małymi literami ma znaczenie. Dla "LOCAL_DRIVE" i "CYC_DRIVE" jest dopuszczalne pisanie tylko dużymi literami.	

<SyncMode>:	**Parametr 3:** Tryb obróbki dla poleceń WRITE do tego urządzenia/pliku		
	Typ:	STRING	
	Wartości:	"SYN":	Zapis synchroniczny Wykonywanie programu jest zatrzymywane, aż proces zapisu będzie zakończony. Pomyślne zakończenie zapisu synchronicznego można sprawdzić poprzez ocenę zmiennej błędu polecenia WRITE.
		"ASYN":	Zapis asynchroniczny Wykonywanie programu nie jest przerywane poprzez polecenie WRITE. **Uwaga:** Zmienna wynikowa polecenia WRITE w tym trybie ma zawsze wartość 0 (nie ma błędu) i nie ma wartości informacyjnej. W tym trybie nie ma pewności, że polecenie WRITE zostało wykonane pomyślnie.
<AccessMode>:	**Parametr 4:** Tryb użycia dla tego urządzenia/pliku		
	Typ:	STRING	
	Wartości:	"SHARED":	Następuje zażądanie urządzenia/pliku w trybie "shared". Inne kanały mogą współużywać urządzenie, tzn. również w tym trybie otwierać.
		"EXCL":	Stosowanie urządzenia/pliku następuje w kanale na wyłączność, żaden inny kanał nie może współużywać urządzenia.
<WriteMode>:	**Parametr 5:** Tryb zapisu dla poleceń WRITE do tego urządzenia/pliku (opcjonalnie)		
	Typ:	STRING	
	Wartości:	"APP":	Zawieszenie Plik pozostaje zachowany w swojej treści, wywołania zapisu powodują dołączenie na końcu.
		"OVR":	Zastąpienie Treść pliku jest kasowana i sporządzana na nowo poprzez następujące dalej wywołania zapisu.
	Wskazówka: Przy pomocy tego parametru nie można zastąpić trybu zapisu zaprojektowanego w extdev.ini. W przypadku konfliktu wywołanie EXTOPEN jest kwitowane z błędem.		

WRITE:	Wstępnie zdefiniowana procedura zapisu wyprowadzanych danych

EXTCLOSE:	Wstępnie zdefiniowana procedura do zamknięcia otwartego zewnętrznego urządzenia/pliku		
<Result>:	**Parametr 1:** Zmienna wynikowa		
	Typ:	INT	
	Wartości:	0	Nie ma błędu
		16	Zaprogramowano niepoprawną ścieżkę zewnętrzna
		21	Błąd przy zamknięciu urządzenia zewnętrznego
<ExtDev>:	**Parametr 2:** Identyfikator symboliczny zamykanego zewnętrznego urządzenia/pliku **Uwaga** Identyfikator musi być identyczny z identyfikatorem podanym w wywołanym EXTOPEN!		

Przykład

Kod programu

```
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("Błąd przy 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
…
```

Patrz również

Operacja na łańcuchach znaków (Strona 89)

Zapisanie pliku (WRITE) (Strona 152)

18.10 Alarmy (SETAL)

W programie NC mogą być ustawiane alarmy. Są one przedstawiane na interfejsie graficznym w specjalnym polu. Z alarmem jest każdorazowo związana reakcja sterowania odpowiednio do kategorii alarmu.

Literatura:
Bardziej szczegółowe informacje dot. reakcji na alarm patrz podręcznik uruchomienia.

Składnia

```
SETAL(<numer alarmu>[,<łańcuch znaków>])
```

Znaczenie

SETAL:	Słowo kluczowe do zaprogramowania alarmu. SETAL musi być programowane w oddzielnym bloku NC.	
<numer alarmu>:	Zmienna typu INT. Zawiera numer alarmu. Obowiązujący zakres numerów alarmów leży między 60000 i 69999, z czego 60000 do 64999 zarezerwowano dla cykli SIEMENS, a numery 65000 do 69999 są do dyspozycji użytkownika.	
<łańcuch znaków>:	Przy programowaniu alarmów cykli użytkownika może dodatkowo zostać podany łańcuch znaków z max 4 parametrami. W tych parametrach mogą być definiowane zmienne teksty użytkownika. Są jednak do dyspozycji również następujące predefiniowane parametry:	
	Parametry	**Znaczenie**
	%1	Numer kanału
	%2	Numer bloku, etykieta
	%3	Indeks tekstu dla alarmów cykli
	%4	Dodatkowy parametr alarmu

Uwaga

Teksty alarmów muszą być projektowane na interfejsie graficznym.

Uwaga

Jeżeli wyprowadzenie alarmu ma nastąpić w języku aktywnym na interfejsie graficznym, użytkownik potrzebuje informacji o języku aktualnie ustawionym na HMI. Tę informację można odczytać w programie obróbki i w akcjach synchronicznych poprzez zmienną systemową $AN_LANGUAGE_ON_HMI (patrz Aktualny język w HMI (Strona 952)").

Przykład

```
Kod programu                          Komentarz
...
N100 SETAL(65000)                     ; Ustawienie alarmu nr 65000
...
```

18.11 Rozszerzone zatrzymanie i wycofanie (ESR)

Funkcja "Rozszerzone zatrzymanie i wycofanie", dalej określana jako ESR, stwarza możliwość elastycznego reagowania na błędy w zależności od procesu:

- **Rozszerzone zatrzymanie**
  Jeżeli specyficzna sytuacja błędu na to pozwala, wszystkie osie zwolnione do zatrzymania rozszerzonego są zatrzymywane w sposób uporządkowany.
- **Wycofanie**
  Skrawające narzędzie jest jak najszybciej wycofywane od obrabianego przedmiotu.
- **Tryb generatora (funkcja napędowa SINAMICS "Regulacja Vdc")**
  Przy spadku poniżej sparametryzowanej wartości napięcia obwodu pośredniego, np. w przypadku zaniku napięcia sieciowego, energia elektryczna potrzebna do wycofania jest wytwarzana przez odzysk energii hamowania przewidzianego w tym celu napędu (tryb generatora).

Źródła wyzwalania

Źródła ogólne (zewnętrzne dla NC / globalne albo specyficzne dla BAG/kanału)

- Wejścia cyfrowe (np. na zespole konstrukcyjnym NCU) wzgl. wewnętrzne w sterowaniu, odczytywalne odwzorowanie wyjść cyfrowych ($A_IN, $A_OUT)
- Stan kanału ($AC_STAT)
- Sygnały VDI ($A_DBB)
- Komunikaty zbiorcze liczby alarmów ($AC_ALARM_STAT)

Źródła osiowe

- Próg wycofania awaryjnego osi holowanej (praca synchroniczna sprzężenia elektronicznego, $VC_EG_SYNCDIFF[<oś nadążna>])
- Napęd: Próg ostrzegania dla obwodu pośredniego (grożący niedobór napięcia), $AA_ESR_STAT[<oś>]
- Napęd: Próg minimalnej prędkości obrotowej generatora (nie ma już zdatnej do zwrotu energii ruchu obrotowego), $AA_ESR_STAT[<oś>].

Logika powiązania statycznych akcji synchronicznych: Powiązanie źródeł/reakcji

Elastyczne możliwości powiązania statycznych akcji synchronicznych są używane do tego, aby na podstawie źródeł w stosunkowo krótkim czasie wyzwalać określone reakcje.

Powiązanie wszystkich odnośnych źródeł przy pomocy statycznych akcji synchronicznych leży w rękach użytkownika. Może on reagować na źródłowe zmienne systemowe jako całość, albo też selektywnie przy pomocy masek bitowych i dokonać odpowiedniego powiązania z pożądanymi reakcjami. Statyczne akcje synchroniczne działają we wszystkich tryby pracy.

Literatura:
Opis działania Akcje synchroniczne

Uaktywnienie

Zwolnienie funkcji

Funkcje pracy generatora, zatrzymania, wycofania, uzyskują zezwolenie przez ustawienie odnośnego sygnału sterującego $AA_ESR_ENABLE. Sygnał ten może być zmieniany z akcji synchronicznych.

Wyzwolenie funkcji

ESR jest wyzwalane wspólnie dla wszystkich zwolnionych osi przez ustawienie zmiennej systemowej $AC_ESR_TRIGGER.

Praca generatora staje się "automatycznie" aktywna w napędzie przy rozpoznaniu grożącego zbyt małego napięcia w obwodzie pośrednim.

Wystarczające dla napędu zatrzymanie i/albo wycofanie staje się aktywne przy rozpoznaniu przerwania komunikacji (między NC i napędem), jak też przy rozpoznaniu niedomiaru napięcia obwodu pośredniego w napędzie (zakładając konfigurację i zezwolenie).

Wystarczające dla napędu zatrzymanie i/albo wycofanie może dodatkowo zostać wyzwolone również ze strony NC przez ustawienie odpowiedniego sygnału sterującego $AN_ESR_TRIGGER" (polecenie Broadcast dla wszystkich napędów).

Literatura

Informacje szczegółowe dot. ESR patrz:

Podręcznik działania Funkcje specjalne; Rozszerzone zatrzymanie i wycofanie (R3)

18.11.1 ESR prowadzone przez NC

18.11.1.1 Wycofanie prowadzone przez NC (POLF, POLFA, POLFMASK, POLFMLIN)

Dla wycofania prowadzonego przez NC są wymagane określone warunki wyjściowe (patrz "Wycofanie prowadzone przez NC (POLF, POLFA, POLFMASK, POLFMLIN) (Strona 696)"). Gdy te warunki są spełnione, wówczas jest uaktywniane szybkie wycofanie (LIFTFAST) dla skonfigurowanych w kanale osi wycofania, przez ustawienie zmiennej systemowej $AC_ESR_TRIGGER (lub $AA_ESR_TRIGGER dla pojedynczych osi).

Składnia

```
POLF[<oś>]=<pozycja>
POLFA(<oś>,<typ>,<wartość>)
POLFMASK(<oś_1>,<oś_2>,...)
POLFMLIN(<oś_1>,<oś_2>,...)
```

Dla `POLFA` są dozwolone następujące formy skrócone:

```
POLFA(<oś>,<typ>) ; forma skrócona dla wycofania w pojedynczej osi
POLFA(oś,0/1/2) ; szybkie wyłączenie aktywności albo uaktywnienie
```

```
POLFA(oś,0,$AA_POLFA[oś]) ; powoduje zatrzymanie przebiegu
wyprzedzającego
POLFA(oś,0) ; nie powoduje zatrzymania przebiegu wyprzedzającego
```

Znaczenie

POLF:	Adres dla podania pozycji docelowej osi wycofania POLF działa modalnie.			
	<oś>:	Nazwa osi geometrycznej albo osi kanału/maszyny, w której następuje wycofanie		
	<pozycja>:	Pozycja wycofania		
		Typ:	REAL	
		Dla osi geometrycznej obowiązuje WKS, w innym przypadku MKS. W przypadku takich samych identyfikatorów dla osi geometrycznej i osi kanału/maszyny wycofanie następuje w układzie współrzędnych obrabianego przedmiotu.		
POLFA:	Predefiniowane wywołanie podprogramu dla podania pozycji wycofania pojedynczych osi			
	<oś>:	Identyfikator kanału		
	<Typ>:	Tryb podania pozycji		
		Typ:	INT	
		Wartość:	0:	zaznaczenie wartości pozycji jako nie obowiązującej
			1:	wartość pozycji jest absolutna
			2:	wartość pozycji jest przyrostowa (odległość)
		Uwaga: Jeżeli oś nie jest osią pojedynczą albo brakuje typu wzgl. typ=0, wówczas jest generowany odpowiedni alarm.		
	<pozycja>:	Pozycja wycofania (patrz wyżej)		
		Uwaga: Wartość pozycji jest przejmowana również z typem=0. Wartość ta jest wówczas jednak zaznaczona jako nie obowiązująca i musi dla wycofania zostać zaprogramowana na nowo.		
POLFMASK:	Predefiniowane wywołanie podprogramu dla wyboru osi, w których po wyzwoleniu szybkiego cofnięcia ma nastąpić wycofanie **niezależnie od siebie**.			
	<oś_1>,...:	Nazwy osi, które w przypadku szybkiego cofnięcia powinny wykonać ruch do swoich pozycji zdefiniowanych przy pomocy POLF. Wszystkie podane osie muszą znajdować się w tym samym układzie współrzędnych.		
	POLFMASK() bez podania osi wyłącza aktywność szybkiego odsunięcia dla wszystkich osi, które są wycofywane niezależnie od siebie.			
POLFMLIN:	Predefiniowane wywołanie podprogramu do wyboru osi, w których po wyzwoleniu szybkiego cofnięcia ma nastąpić wycofanie **w zależności liniowej**.			
	<oś_1>,...:	patrz wyżej		
	POLFMLIN() bez podania osi wyłącza aktywność szybkiego odsunięcia dla wszystkich osi, które są wycofywane w zależności liniowej.			

Uwaga

Zanim poprzez POLFMASK albo POLFMLIN będzie można udzielić zezwolenia na szybkie cofnięcie do stałej pozycji, musi dla wybranych osi zostać zaprogramowana pozycja przy pomocy POLF.

Uwaga

Jeżeli osie uzyskają kolejno zezwolenie przy pomocy POLFMASK, POLFMLIN albo POLFMLIN, POLFMASK, obowiązuje dla danej osi każdorazowo ostatnie ustalenie.

Uwaga

Pozycje zaprogramowane przy pomocy POLF i uaktywnianie przez POLFMASK albo POLFMLIN są kasowane przy starcie programu obróbki. Oznacza to, że użytkownik musi w każdym programie obróbki na nowo zaprogramować wartości dla POLF i wybrane osie w POLFMASK wzgl. POLFMLIN.

Uwaga

Jeśli przy stosowaniu form skróconych POLFA zostanie zmieniony tylko typ, wówczas użytkownik musi zagwarantować, by pozycja wycofania lub droga wycofania zawierała sensowną wartość. W szczególności należy na nowo nastawić pozycję wycofania i drogę wycofania po POWER ON.

Przykład

Wycofanie pojedynczej osi:

Kod programu	Komentarz
MD37500 $MA_ESR_REACTION[AX1]=21	; Wycofanie prowadzone przez NC.
...	
$AA_ESR_ENABLE[AX1]=1	
POLFA(AX1,1,20.0)	; Do AX1 jest przydzielana osiowa pozycja wycofania 20.0 (absolutna).
$AA_ESR_TRIGGER[AX1]=1	; Od tego miejsca zaczyna się wycofanie.

Dalsze informacje

Warunki dla wycofania prowadzonego przez NC

- W Kanale jest zaprojektowana oś wycofania dla wycofania prowadzonego przez NC:
  MD37500 $MA_ESR_REACTION = 21
- ESR musi być zwolnione dla tej osi:
  $AA_ESR_ENABLE = 1
- Czasy zwłoki są zdefiniowane:
  MD21380 $MC_ESR_DELAY_TIME1
  MD21381 $MC_ESR_DELAY_TIME2

- W programie obróbki są zaprogramowane specyficzne dla osi pozycje wycofania z POLF.
- Osie są przy pomocy POLFMASK/POLFMLIN wybrane dla wycofania prowadzonego przez NC.
- Dla ruchu wycofania sygnały zezwoleń muszą być, jak i pozostać ustawione.

Zwolnienie i wystartowanie wycofania prowadzonego przez NC

Gdy jest ustawiona zmienna systemowa $AC_ESR_TRIGGER = 1 i gdy w tym kanale jest skonfigurowana oś wycofania (tzn. MD37500: $MA_ESR_REACTION = 21) i dla tej osi jest ustawiona $AA_ESR_ENABLE = 1, wówczas jest w tym kanale uaktywniane szybkie wycofanie (LIFTFAST).

Skonfigurowany przy pomocy POLF (wzgl. LFPOS) ruch wycofania osi wybranej(ych) przy pomocy POLFMASK albo POLFMLIN zastępuje ruch po torze ustalony dla tej(tych) osi w programie obróbki.

Dla wycofania jest do dyspozycji maksymalnie suma czasów MD21380 $MC_ESR_DELAY_TIME1 i MD21381 $MC_ESR_DELAY_TIME2. Po upływie tego czasu jest również dla osi wycofania rozpoczynane szybkie hamowanie z następną aktualizacją.

Uwaga

Rozszerzony ruch wycofania (tzn. LIFTFAST/LFPOS) wyzwolony przez $AC_ESR_TRIGGER) jest **nieprzerywalny** i może zostać przedwcześnie zakończony tylko poprzez zatrzymanie awaryjne.

Uwaga

Wycofanie wyzwolone przez $AC_ESR_TRIGGER jest zablokowane przed wielokrotnym wycofaniem.

Wycofanie w pojedynczej osi

Przy wycofaniu w pojedynczej osi musi przy pomocy POLFA być zaprogramowana pozycja wycofania pojedynczej osi i muszą zostać dotrzymane następujace warunki:

- $AA_ESR_ENABLE = 1
- <oś> musi w chwili wyzwolenia ($AA_ESR_TRIGGER = 1) być osią pojedynczą.
- <typ> musi być albo 1 albo 2.

Kierunek wycofania przy szybkim wycofaniu

W chwili uaktywnienia szybkiego wycofania jest uaktywniany obowiązujący Frame.

Uwaga

Frame z obrotem wpływają poprzez POLF również na kierunek, w którym następuje wycofanie.

Zamiana osi

Osie wycofania należy zawsze przyporządkowywać dokładnie jednemu kanałowi NC i nie mogą one być zamieniane pomiędzy kanałami. Przy próbie zamiany osi wycofania z innym kanałem, jest generowany alarm. Dopiero gdy przy pomocy $AA_ESR_ENABLE[AX] = 0

aktywność tej osi została wyłączona, można ją przełączyć na nowy kanał. Po dokonanej zamianie osi można je ponownie obciążyć przy pomocy $AA_ESR_ENABLE[AX] = 1.

Osie neutralne

Osie neutralne nie mogą wykonywać ESR prowadzonego przez NC.

18.11.1.2 Zatrzymanie prowadzone przez NC

Dla zatrzymania skonfigurowanego w kanale jest przez ustawienie zmiennej systemowej $AC_ESR_TRIGGER (wzgl. $AA_ESR_TRIGGER dla pojedynczych osi) uaktywniane zatrzymanie prowadzone przez NC.

Warunki

- W Kanale jest zaprojektowana oś zatrzymania dla zatrzymania prowadzonego przez NC:
  MD37500 $MA_ESR_REACTION = 22
- ESR musi być zwolnione dla tej osi:
  $AA_ESR_ENABLE = 1
- Czasy zwłoki są zdefiniowane:
  MD21380 $MC_ESR_DELAY_TIME1 (czas zwłoki osi ESR)
  MD21381 $MC_ESR_DELAY_TIME2 (czas ESR dla hamowania interpolacji)

Przebieg

Poprzez czas w MD21380 oś interpoluje bez przeszkód dalej, jak zaprogramowano. Po upływie czasu w MD21380 rozpoczynane jest interpolacyjnie prowadzone hamowanie (zatrzymanie według charakterystyki). Dla hamowania prowadzonego interpolacyjnie jest do dyspozycji maksymalnie przedział czasu w MD21381. Po upływie tego czasu jest rozpoczynane szybkie hamowanie z następną aktualizacją.

Przykład

Zatrzymanie pojedynczej osi:

Kod programu	**Komentarz**
`MD37500 $MC_ESR_REACTION[AX1]=22`	`; Zatrzymanie prowadzone przez NC.`
`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 tego miejsca rozpoczyna się zatrzymanie.`

18.11.2 ESR wystarczające dla napędu

18.11.2.1 Zaprojektowanie zatrzymania niezależnego dla napędu (ESRS)

Przy pomocy funkcji `ESRS(...)` są projektowane parametry napędu dla "zatrzymania" autarkicznej dla napędu funkcji ESR.

Składnia

```
ESRS(<oś_1>,<czas zatrzymania_1>[,...,<oś_n>,<czas zatrzymania_n>])
```

Znaczenie

`ESRS(...)`:	Funkcja dla zapisu parametrów napędu dla funkcji ESR "zatrzymanie" Funkcja: • musi znajdować się w oddzielnym bloku. • wyzwala zatrzymanie przebiegu wyprzedzającego. • nie może być stosowana w akcjach synchronicznych.	
`<oś_1>`, `...`, `<oś_n>`:	Oś, która powinna zostać zaprojektowana dla zatrzymania wystarczającego dla napędu W napędzie jest dla tej osi pisany parametr napędu p0888 (konfiguracja): p0888 = 1	
	Typ:	AXIS
	Zakres wartości:	Identyfikator kanału
`<czas zatrzymania_1>`, `...`, `<czas zatrzymania_n>`:	Okres czasu, w którym napęd po wystąpieniu błędu nadal wykonuje ruch z aktualną wartością zadaną prędkości obrotowej W napędzie jest dla podanej osi pisany parametr napędu p0892 (stopień czasowy): p0892 = <czas zatrzymania>	
	Jednostka:	s
	Typ:	REAL
	Zakres wartości:	0.00 - 20.00
W jednym wywołaniu funkcji można zaprogramować maksymalnie 5 osi; n = 5		

18.11.2.2 Zaprojektowanie wycofania niezależnego dla napędu (ESRR)

Przy pomocy funkcji `ESRR(...)` są projektowane parametry napędu dla "wycofania" autarkicznej dla napędu funkcji ESR.

Składnia

```
ESRR(<oś_1>,<droga wycofania_1>,<prędkość
wycofania_1>[,...,<oś_n>,<droga wycofania_n>,<prędkość
wycofania_n>])
```

Znaczenie

`ESRR(...)`:	Funkcja do zapisu parametrów napędu dla funkcji ESR "wycofanie" Funkcja: • musi znajdować się w oddzielnym bloku. • wyzwala zatrzymanie przebiegu wyprzedzającego. • nie może być stosowana w akcjach synchronicznych.	

`<oś_1>,`
`...,`
`<oś_n>`:
	Oś, dla której ma zostać zaprojektowane wycofanie wystarczające dla napędu
	W napędzie jest dla tej osi pisany parametr napędu p0888 (konfiguracja):
	p0888 = 2	

	Typ:	AXIS
	Zakres wartości:	Identyfikator kanału

`<droga wycofania_1>,`
`...,`
`<droga wycofania_n>`:
	Droga wycofania jest dla napędu przeliczana na prędkość obrotową wycofania. Wartość jest dla podanej osi zapisywana w parametrze napędu p0893 (prędkość obrotowa):
	p0893 = (<droga wycofania _n> przeliczona na prędkość obrotową wycofania)	

	Jednostka:	mm/min, cale/min, stopni/min (zależnie od jednostki osi)
	Typ:	REAL
	Zakres wartości:	MIN - MAX

`<Prędkość wycofania_1>,`
`...,`
`<Prędkość wycofania_n>`:
	Prędkość wycofania jest dla napędu przeliczana na czas trwania. Wartość jest dla podanej osi zapisywana w parametrze napędu p0892 (stopień czasowy) [s]:
	p0892 = <droga wycofania_n> / <prędkość wycofania _n>	

	Jednostka:	mm/min, cale/min, stopni/min (zależnie od jednostki osi)
	Typ:	REAL
	Zakres wartości:	0.00 - MAX
W jednym wywołaniu funkcji można zaprogramować maksymalnie 5 osi; n = 5		

18.12 Definiowanie półfabrykatu (WORKPIECE)

Sterowanie musi znać kształt i wielkość półfabrykatu, aby można go było wyświetlić w symulacji graficznej. Użytkownik ma możliwość definiowania półfabrykatu poprzez interfejs obsługi lub bezpośrednio w programie NC. Ponadto definicje półfabrykatów pozostają zachowane przez reset (końca programu/Kanału/BAG). Wraz z następnym rozruchem sterowania są one automatycznie kasowane.

Składnia

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

Znaczenie

WORKPIECE(...):		Wstępnie zdefiniowana procedura do definiowania półfabrykatu		
		Zatrzymanie przebiegu wyprzedzającego:	tak	
		W oddzielnym bloku:	tak	
Parametry				
1	"<WP>":	Nazwa przedmiotu obrabianego (opcjonalnie)		
		Typ danych:	STRING	
		Podanie jest konieczne tylko wtedy, gdy w jednym kanale można podać wiele przedmiotów obrabianych. Bez podania przyjmowane jest „WORKP<n>”, gdzie <n> oznacza numer zadeklarowanego kanału.		
2	"<RefP>":	Zamocowanie (opcjonalnie; tylko dla frezarek)		
		Typ danych:	STRING	
		Zakres wartości:	„Stół”	Zamocowanie na stole stałym
			"A"	Zamocowanie na osi obrotowej A
			"B"	Zamocowanie na osi obrotowej B
			"C"	Zamocowanie na osi obrotowej C
		Warunek: Stół lub oś obrotowa musi być odblokowana poprzez odpowiednią daną maszynową dla zamocowania półfabrykatu (patrz Podręcznik uruchomieniowy SINUMERIK Operate).		
3	"<ZeroOffset>":	Ustawiane przesunięcie punktu zerowego do pozycji półfabrykatu (nieprogramowalne). Wybór ustawianego przesunięcia punktu zerowego do pozycji półfabrykatu jest proponowana tylko przy wprowadzania półfabrykatu za pomocą interfejsu obsługi. Podczas bezpośredniej definicji półfabrykatu w programie obróbki półfabrykat odnosi się natomiast zawsze do aktualnie obowiązującego przesunięcia punktu zerowego.		

4	"<Type>":	Kształt półfabrykatu		
		Typ danych:	STRING	
		Zakres wartości:	"CYLINDER":	Walec
			"PIPE":	Tuleja
			"RECTANGLE":	Prostopadłościan współosiowy
			"BOX":	Prostopadłościan
			"N_CORNER":	Wielobok
5 ... 12	<Par5> ... <Par12>:	Parametry do opisu kształtu półfabrykatu		
		Typ danych:	REAL	
		Liczba niezbędnych parametrów i ich znaczenie zależy od kształtu półfabrykatu i wartości parametru bit. Patrz: • „Tablica parametrów do opisu kształtu półfabrykatu” • „Tablica bitów dla parametrów”		
WORPIECE():		Wywołanie WORKPIECE bez parametru kasuje wszystkie definicje półfabrykatu.		
WORPIECE(<WP>):		Wywołanie WORKPIECE z nazwą przedmiotu obrabianego kasuje tylko tą definicję półfabrykatu.		

Tabela 18-1 Parametry do opisu kształtu półfabrykatu

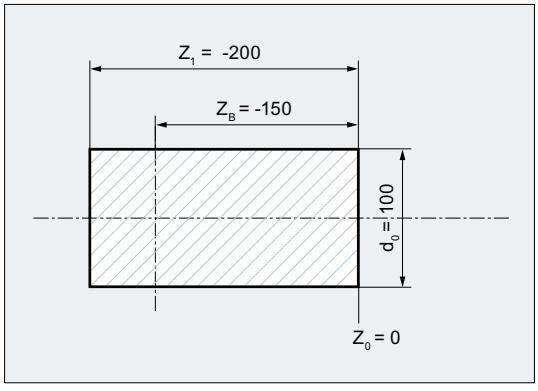
Kształt półfabrykatu	Parametry							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
Walec	Bity dla parametrów Wartość real (rzeczywista), która jest interpretowana jako kodowana bitowo liczba całkowita. Bity definiują znaczenie następujących parametrów (patrz „Tablica bitów dla parametrów”).	Punkt odniesienia Z_0	Długość Z_1	Wymiar obróbkowy Z_B	Średnica zewnętrzna d_0	-	Obrót wokół osi obrotowej	-
Tuleja		Punkt odniesienia Z_0	Długość Z_1	Wymiar obróbkowy Z_B	Średnica zewnętrzna d_0	Grubość ścianki (przyr.) / średnica wewnętrzna d_1 (abs.)	Obrót wokół osi obrotowej	-
Prostopadłościan współosiowy		Punkt odniesienia Z_0	Długość Z_1	Wymiar obróbkowy Z_B	Szerokość W	Długość L	Obrót wokół osi obrotowej	-
Prostopadłościan		Punkt odniesienia Z_0	Długość Z_1	Wymiar obróbkowy Z_B	X_0	Y_0	X_1	Y_1
Wielobok		Punkt odniesienia Z_0	Długość Z_1	Wymiar obróbkowy Z_B	Liczba narożników	Rozwartość klucza	Obrót wokół osi obrotowej	-

Tabela 18-2 Bity dla parametrów

Bit	Znaczenie	
4 (0x0010)	Prostopadłościan: X_1	
	= 0	ink
	= 1	abs.
5 (0x0020)	Prostopadłościan: Y_1	
	= 0	ink
	= 1	abs.
6 (0x0040)	Długość Z_1 (wymiar końcowy)	
	= 0	ink
	= 1	abs.
Bit 7 (0x0080)	Wymiar obróbkowy Z_B	
	= 0	ink
	= 1	abs.
Bit 8 (0x0100)	Tuleja: Grubość ścianki / Średnica wewnętrzna	
	= 0	ink
	= 1	abs.
9 (0x0200)	Wielobok	
	= 0	Rozwartość klucza
	= 1	Długość krawędzi
12 (0x1000)	Zamocowanie dla tokarek	
	= 0	Wrzeciono główne
	= 1	Wrzeciono przechwytujące
13 (0x2000)	Wrzeciono przechwytujące	
	= 0	z odbiciem lustrzanym
	= 1	bez odbicia lustrzanego

Przykłady

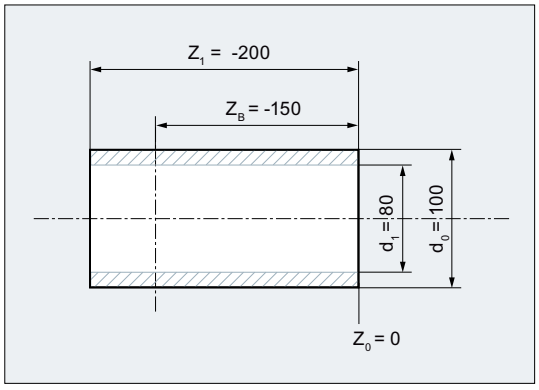
Przykład 1: Półfabrykaty cylindryczne na tokarce




```
Kod programu                                    Komentarz
...
```

Kod programu	Komentarz
WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100)	; **Definicja półfabrykatu:** ; Kształt półfabrykatu: Walec ; Bit parametru=0(brak ustawionego bitu) → Wartości dla długości i wymiaru obróbkowego są przyrostowe, półfabrykat na wrzecionie głównym ; Punkt odniesienia(Z0)=0 ; Długość(Z1)=-200 ; Wymiar obróbkowy(ZB)=-150 ; Średnica zewnętrzna(d0)=100
...	

Przykład 2: Półfabrykaty w kształcie tulei na tokarce



Kod programu	Komentarz
...	
WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80)	; **Definicja półfabrykatu:** ; Kształt półfabrykatu: Tuleja ; Bit parametru=256(Bit8=1) → Średnica wewnętrzna jest absolutnie, długość i wymiar obróbkowy są inkrementalnie, półfabrykat na wrzecionie głównym ; Punkt odniesienia(Z0)=0 ; Długość(Z1)=-200 ; Wymiar obróbkowy(ZB)=-150 ; Średnica zewnętrzna(d0)=100 ; Średnica wewnętrzna(d1)=80
...	

18.13 Przełączanie trybu językowego (G290, G291)

Sterowanie oferuje możliwość odczytywania i wykonywania programów obróbki z zewnętrznych systemów CNC. Warunkiem jest ustalenie podczas uruchamiania odpowiedniego trybu językowego NC (dialekt ISO).

Literatura:
Podręcznik działania, Dialekty ISO

Tryb dialektu ISO można aktywować dla każdego kanału oddzielnie. Na przykład tryb dialektu ISO może pracować w kanale 1, podczas gdy w kanale 2 aktywny jest tryb SINUMERIK.

Przełączenie między trybem SINUMERIK i dialektem ISO następuje za pomocą poleceń z grupy nr 47 w programie NC. Przełączenie nie ma wpływu na aktywne narzędzie, korekcje narzędzi i przesunięcia punktu zerowego.

Składnia

```
G291
...
G290
```

Znaczenie

G290:	Aktywacja trybu językowego SINUMERIK	
	W oddzielnym bloku:	tak
	Działanie:	Modalnie
G291:	Aktywacja trybu językowego ISO	
	W oddzielnym bloku:	tak
	Działanie:	Modalnie

Warunki

Tryb SINUMERIK

- Ustawienie domyślne plecenia G może zostać ustalone dla każdego kanału za pomocą danej maszynowej.
- W trybie SINUMERIK nie można programować poleceń językowych z dialektu ISO.

Tryb dialektu ISO

- Tryb dialektu ISO jako ustawienie podstawowe sterowania można ustawić za pomocą danych maszynowych. Sterowanie uruchamia się następnie w trybie dialektu ISO.
- Programowane mogą być tylko polecenia G z dialektu ISO. Programowanie funkcji SINUMERIK w trybie dialektu ISO nie jest możliwe.
- Łączenie dialektu ISO i języka SINUMERIK w tym samym bloku NC nie jest możliwe.
- Przełączanie poprzez polecenia G między dialektem ISO M (frezowanie) i dialektem ISO T (toczenie) nie jest możliwe.

- Mogą zostać wywołane podprogramy, które są zaprogramowane w trybie SINUMERIK.
- Jeżeli funkcje SINUMERIK powinny zostać zastosowane, należy najpierw przełączyć w tryb SINUMERIK (patrz przykład).

Przykład

Kompresja bloków liniowych w trybie dialektu ISO.

Kod programu	**Komentarz**
N5 G290	; Aktywuje tryb językowy SINUMERIK
N10 COMPON	; COMPON jest poleceniem języka Siemens, które aktywuje funkcje kompresora, która koleje bloki liniowe zastępują blokami wielomianu z możliwie dużą długością toru.
N15 G291	; Aktywuje tryb językowy ISO
N20 G01 X100 Y100 F1000	; Ponieważ COMPON jest aktywny w trybie SINUMERIK, można za pomocą tej funkcji także kompresować bloki liniowe w trybie dialektu ISO.
...	

Własne programy obróbki 19

19.1 Funkcje wspierające skrawanie

Do skrawania są udostępniane gotowe cykle obróbkowe Ponadto jest możliwość sporządzania własnych programów skrawania, przy pomocy niżej wymienionych funkcji.

- Sporządzenie tablicy konturu (CONTPRON)
- Sporządzenie kodowanej tablicy konturu (CONTDCON)
- Wyłączenie przygotowania konturu (EXECUTE)
- Obliczenie punktu przecięcia między dwoma elementami konturu (INTERSEC)
  (Tylko dla tablic, które zostały sporządzone przez CONTPRON.)
- Wykonywanie pojedynczymi blokami elementów konturowych tablicy (EXECTAB)
  (Tylko dla tablic, które zostały sporządzone przez CONTPRON.)
- Obliczenie danych okręgu (CALCDAT)

Uwaga

Można stosować te funkcje nie tylko do skrawania, lecz uniwersalnie.

Warunki

Przed wywołaniem funkcji CONTPRON albo CONTDCON musi:

- nastąpić ruch do punktu startowego, który pozwala na bezkolizyjna obróbkę.
- być wyłączona korekcja promienia ostrza przy pomocy `G40`.

19.2 Sporządzenie tablicy konturu (CONTPRON)

Przy pomocy `CONTPRON` włączamy przygotowanie konturu. Następnie wywoływane bloki NC nie są wykonywane, lecz dzielone na poszczególne ruchy i zapisywane w tablicy konturu. Każdemu elementowi konturu odpowiada jeden wiersz w dwuwymiarowej tablicy konturu. Jest zwracana liczba obliczonych podcięć.

Składnia

Włączenie przygotowania konturu:
`CONTPRON(<tablica konturu>,<rodzaj obróbki>,<podcięcia>, <kierunek obróbki>)`

Wyłączenie przygotowania konturu i powrót do normalnego trybu pracy:
`EXECUTE(<BŁĄD>)`

Patrz " Wyłączenie przygotowania konturu (EXECUTE) (Strona 725) "

Znaczenie

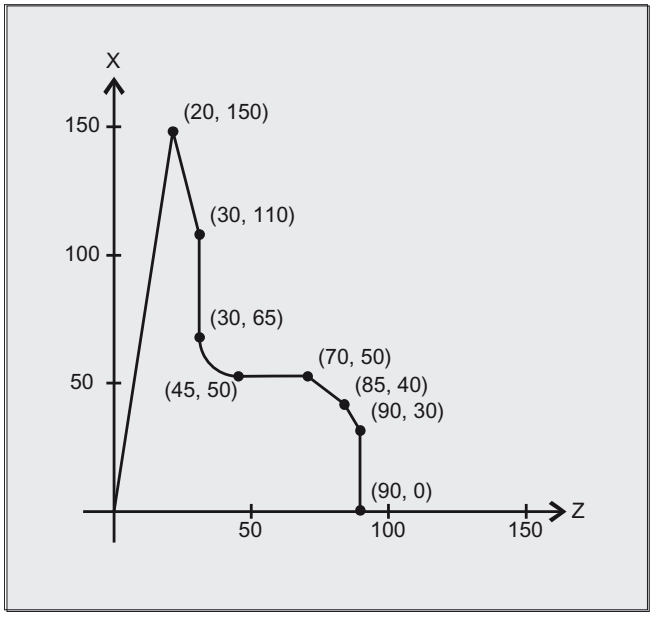
`CONTPRON`:	Predefiniowana procedura do włączenia przygotowania konturu do sporządzenia tablicy konturu		
`<tablica konturu>`:	Nazwa tablicy konturu		
`<rodzaj obróbki>`:	Parametr rodzaju obróbki		
	Typ:	CHAR	
	Wartość:	`"G"`:	Toczenie wzdłużne: Obróbka wewnętrzna
		`"L"`:	Toczenie wzdłużne: Obróbka zewnętrzna
		`"N"`:	Toczenie poprzeczne: Obróbka wewnętrzna
		`"P"`:	Toczenie poprzeczne: Obróbka zewnętrzna
`<podcięcia>`:	Zmienna wynikowa liczby występujących elementów podcięć		
	Typ:	INT	
`<kierunek obróbki>`:	Parametr kierunku obróbki		
	Typ:	INT	
	Wartość:	0	Przygotowanie konturu do przodu (wartość standardowa)
		1	Przygotowanie konturu w obydwu kierunkach

Przykład 1

Sporządzenie tablicy konturu z:

- nazwą "KTAB"
- max 30 elementami konturu (okręgi, proste)

- zmienną wynikową liczby występujących elementów podcięć
- zmienną komunikatów błędów



Program NC:

Kod programu	**Komentarz**
N10 DEF REAL KTAB[30,11]	; Tablica konturu o nazwie KTAB i max 30 elementami konturu, wartość parametru 11 (liczba kolumn tablicy) jest wielkością stałą.
N20 DEF INT ANZHINT	; Zmienna liczby elementów podcięć o nazwie ANZHINT.
N30 DEF INT BLAD	; Zmienna dla zwrotnego komunikatu błędu (0=nie ma błędu, 1=błąd).
N40 G18	
N50 CONTPRON(KTAB,"G",ANZHINT)	; Włączenie przygotowania konturu.
N60 G1 X150 Z20	; N60 do N120: Opis konturu
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(BŁĄD)	; Zakończenie wypełniania tablicy konturu, przełączenie na normalną pracę programową.
N140 …	; Dalsze opracowywanie tablicy.

Tablica konturu KTAB:

Indeks Wiersz	Kolumna									
(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

Objaśnienie treści kolumn:

(0) Wskaźnik na następny element konturu (na jego numer wiersza)

(1) Wskaźnik na poprzedni element konturu

(2) Kodowanie trybu konturu dla ruchu

Możliwe wartości dla 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) Punkt początkowy elementów konturu

(3) = odcięta, (4) = rzędna w aktualnej płaszczyźnie

(5), (6) Punkt końcowy elementów konturu

(5) = odcięta, (6) = rzędna w aktualnej płaszczyźnie

(7) Wskaźnik max/min: Oznacza lokalne maksima i minima w konturze

(8) Wartość maksymalna między elementem konturu i odciętą (przy obróbce wzdłużnej) wzgl. odcięta (przy obróbce poprzecznej). Kąt jest zależny od zaprogramowanego rodzaju obróbki.

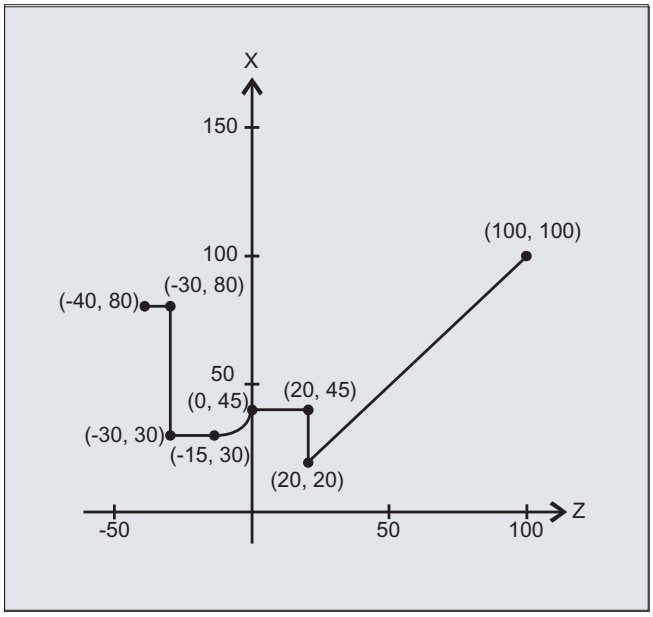
(9), (10) Współrzędne punktu środkowego elementu konturu, gdy jest to blok kołowy.

(9) = odcięta, (10) = rzędna

Przykład 2

Sporządzenie tablicy konturu z

- nazwą KTAB
- max 92 elementami konturu (okręgi, proste)
- Tryb pracy: Toczenie wzdłużne, obróbka zewnętrzna
- przygotowaniem do przodu i do tyłu



Program NC:

Kod programu	**Komentarz**
N10 DEF REAL KTAB[92,11]	; Tablica konturu o nazwie KTAB z max 92 elementami konturu, wartość parametru 11 jest wielkością stałą.
N20 DEF CHAR BT="L"	; Tryb pracy dla CONTPRON: Toczenie wzdłużne, obróbka zewnętrzna
N30 DEF INT HE=0	; Liczba elementów podcięcia =0
N40 DEF INT MODE=1	; Przygotowanie do przodu i do tyłu
N50 DEF INT ERR=0	; Zwrotna sygnalizacja błędu
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; Włączenie przygotowania konturu.
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)	; Zakończenie wypełniania tablicy konturu, przełączenie na normalną pracę programową.
...	

Tablica konturu KTAB:

Po zakończeniu przygotowania konturu, kontur jest do dyspozycji w obydwu kierunkach.

Indeks	Kolumna										
Wiersz	(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

Objaśnienie treści kolumn i uwagi do wierszy 0, 1, 6, 8, 83, 85 i 91

Obowiązują wyjaśnienia treści kolumn wymienione w przykładzie 1.

Zawsze w wierszu 0 tablicy:

1) Poprzedzający: Wiersz n zawiera koniec konturu do przodu

2) Następny: Wiersz n jest końcem tablicy konturu do przodu

Po jednym razie w ramach elementów konturu do przodu:

3) Poprzedzający: Początek konturu (do przodu)

4) Następny: Koniec konturu (do przodu)

Zawsze na wiersz końca tablicy konturu (do przodu) +1:

5) Poprzedzający: Liczba podcięć do przodu

6) Następny: Liczba podcięć do tyłu

Po jednym razie w ramach elementów konturu do tyłu:

7) Następny: Koniec konturu (w kierunku do tyłu)

8) Poprzedzający: Początek konturu (do tyłu)

Zawsze w ostatnim wierszu tablicy:

9) Poprzedzający: Wiersz n jest początkiem tablicy konturu (do tyłu)

10) Następny: Wiersz n zawiera początek konturu (do tyłu)

Dodatkowe informacje

Dozwolone polecenia ruchu, układ współrzędnych

Dla programowania konturu są dopuszczalne następujące polecenia G:

- Grupa G1: `G0`, `G1`, `G2`, `G3`

Dodatkowo możliwe są:

- Zaokrąglenie i faza
- Programowanie okręgu przez `CIP` i `CT`

Funkcje spline, wielomian i gwint prowadzą do błędów.

Zmiany układu współrzędnych przez włączenie frame są między `CONTPRON` i `EXECUTE` niedopuszczalne. To samo dotyczy zmiany między `G70` i `G71` wzgl. `G700` i `G710`.

Zamiana osi geometrycznych przy pomocy `GEOAX` podczas przygotowywania tablicy konturu prowadzi do alarmu.

Elementy podcięć

Opis konturu poszczególnych elementów podcięć może nastąpić do wyboru w podprogramie albo w poszczególnych blokach.

Skrawanie niezależne od zaprogramowanego kierunku konturu

Przygotowanie konturu z `CONTPRON` zostało tak rozszerzone, że po jego wywołaniu tablica konturu jest do dyspozycji niezależnie od zaprogramowanego kierunku.

19.3 Sporządzenie kodowanej tablicy konturu (CONTDCON)

Po włączeniu przygotowania konturu za pomocą CONTDCON, kolejne wywołane bloki NC są zapisywane w 6-kolumnowej tablicy konturu w kodowanej formie. Każdemu elementowi konturu odpowiada jeden wiersz tablicy. Znając niżej podane kodowania można np. dla cykli zestawiać programy DIN z wierszy tablicy. W wierszu tablicy o numerze 0 są zapisywane dane punktu wyjściowego.

Składnia

Włączenie przygotowania konturu:
`CONTDCON(<tablica konturu>,<kierunek obróbki>)`

Wyłączenie przygotowania konturu i powrót do normalnego trybu pracy:
`EXECUTE(<BŁĄD>)`

Patrz " Wyłączenie przygotowania konturu (EXECUTE) (Strona 725) "

Znaczenie

CONTDCON:	Wstępnie zdefiniowana procedura do włączenia przygotowania konturu do sporządzenia kodowanej tablicy konturu		
<tablica konturu>:	Nazwa tablicy konturu		
<kierunek obróbki>:	Parametr kierunku obróbki		
	Typ:	INT	
	Wartość:	0	Przygotowanie konturu według sekwencji bloków konturu (wartość standardowa)
		1	Jest niedozwolona

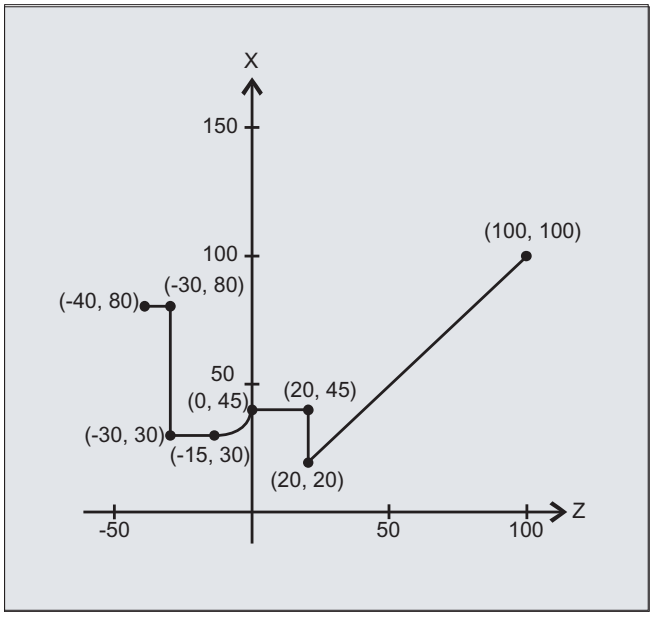
Uwaga

Polecenia G dopuszczone dla CONTDCON w tablicowanym fragmencie programu są bardziej obszerne, niż w przypadku CONTPRON. Ponadto są równocześnie zapisywane posuwy i typ posuwu dla fragmentu konturu.

Przykład

Sporządzenie tablicy konturu z:

- Nazwą "KTAB"
- Elementami konturu (okręgi, proste)
- Trybem pracy: Toczenie
- Kierunkiem obróbki: Do przodu



Program NC:

Kod programu	Komentarz
N10 DEF REAL KTAB[9,6]	; Tablica konturu o nazwie KTAB i 9 wierszach. Pozwalają one na 8 bloków konturu. Wartość parametru 6 (liczba kolumn tablicy) jest stałą wielkością.
N20 DEF INT MODE = 0	; Zmienna kierunku obróbki. Wartość standardowa 0: Tylko w zaprogramowanym kierunku konturu.
N30 DEF INT ERROR = 0	; Zmienna dla zwrotnego komunikatu błędu.
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; Wywołanie przygotowania konturu (MODE można pominąć).
N110 G1 Z20 X20 F200	; Opis konturu.
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)	Zakończenie wypełniania tablicy konturu, przełączenie na normalną pracę programu.
...	

Tablica konturu KTAB:

	Indeks kolumny					
	0	1	2	3	4	5
Indeks wiersza	Tryb konturu	Punkt końcowy odcięta	Punkt końcowy rzędna	Punkt środkowy odcięta	Punkt środkowy rzędna	Posuw
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

Objaśnienie treści kolumn:

Wiersz 0: Kodowanie dla **punktu startowego:**

Kolumna 0: 10^0 (miejsce jednostek): G0 = 0
10^1 (miejsce dziesiątek): G70 = 0, G71 = 1, G700 = 2, G710 = 3
Kolumna 1: Punkt startowy odcięta
Kolumna 2: Punkt startowy rzędna
Kolumna 3-4: 0
Kolumna 5: Indeks wiersza ostatniego elementu konturu w tablicy

Wiersze 1-n: Wpisy **elementów konturu**

Kolumna 0: 10^0 (miejsce jednostek): G0 = 0, G1 = 1, G2 = 2, G3 = 3
10^1 (miejsce dziesiątek): G70 = 0, G71 = 1, G700 = 2, G710 = 3
10^2 (miejsce setek): G90 = 0, G91 = 1
10^3 (miejsce tysięcy): G93 = 0, G94 = 1, G95 = 2, G96 = 3
10^4 (miejsce dziesiątek tysięcy): G60 = 0, G44 = 1, G641 = 2, G642 = 3
10^5 (miejsce setek tysięcy): G9 = 1
Kolumna 1: Punkt końcowy odcięta
Kolumna 2: Punkt końcowy rzędna
Kolumna 3: Punkt środkowy odcięta przy interpolacji okręgu
Kolumna 4: Punkt środkowy rzędna przy interpolacji okręgu
Kolumna 5: Posuw

Dodatkowe informacje

Dozwolone polecenia ruchu, układ współrzędnych

Dla programowania konturu są dopuszczalne następujące grupy G i polecenia G:

Grupa G1:	`G0`, `G1`, `G2`, `G3`
Grupa G 10:	`G60`, `G64`, `G641`, `G642`
Grupa G 11:	`G9`
Grupa G 13:	`G70`, `G71`, `G700`, `G710`
Grupa G 14:	`G90`, `G91`
Grupa G15:	`G93`, `G94`, `G95`, `G96`, `G961`

Dodatkowo możliwe są:

- Zaokrąglenie i faza
- Programowanie okręgu przez `CIP` i `CT`

Funkcje spline, wielomian i gwint prowadzą do błędów.

Zmiany układu współrzędnych przez włączenie frame są między `CONTDCON` i `EXECUTE` niedopuszczalne. To samo dotyczy zmiany między `G70` i `G71` wzgl. `G700` i `G710`.

Zamiana osi geometrycznych przy pomocy `GEOAX` podczas przygotowywania tablicy konturu prowadzi do alarmu.

Kierunek obróbki

Tablica konturu utworzona przy pomocy `CONTDCON` jest przewidziana do skrawania w zaprogramowanym kierunku konturu.

19.4 Obliczenie punktu przecięcia między dwoma elementami konturu (INTERSEC)

`INTERSEC` określa punkt przecięcia dwóch znormalizowanych elementów konturu z tablicy konturu utworzonej przy pomocy `CONTPRON`.

Składnia

```
<Status>=INTERSEC(<tablica konturu_1>[<element konturu_1>],
<tablica konturu_2>[<element konturu_2>],<punkt przecięcia>,<rodzaj
obróbki>)
```

Znaczenie

`INTERSEC`:	Predefiniowana funkcja do obliczenia punktu przecięcia dwóch elementów konturu z tablicy konturu utworzonej przy pomocy `CONTPRON`		
`<status>`:	Zmienna statusu punktu przecięcia		
	Typ:	BOOL	
	Wartość:	TRUE	Punkt przecięcia znaleziony
		FALSE	Nie znaleziono punktu przecięcia
`<tablica_konturu_1>`:	Nazwa pierwszej tablicy konturu		
`<element_konturu_1>`:	Numer elementu konturu pierwszej tablicy konturu		
`<tablica_konturu_2>`:	Nazwa drugiej tablicy konturu		
`<element_konturu_2>`:	Numer elementu konturu drugiej tablicy konturu		
`<punkt przecięcia>`:	Współrzędne punktu przecięcia w aktywnej płaszczyźnie (`G17` / `G18` / `G19`)		
	Typ:	REAL	
`<rodzaj obróbki>`:	Parametr rodzaju obróbki		
	Typ:	INT	
	Wartość:	0	Obliczenie punktu przecięcia w płaszczyźnie aktywnej z parametrem 2 (wartość standardowa)
		1	Obliczenie punktu przecięcia niezależnie od przekazanej płaszczyzny

Uwaga

Należy pamiętać, że zmienne przed ich zastosowaniem muszą być zdefiniowane.

Przekazanie konturów wymaga dotrzymania wartości zdefiniowanych przy pomocy `CONTPRON`:

Parametry	Znaczenie
2	Kodowanie trybu konturu dla ruchu
3	Punkt początkowy konturu odcięta
4	Punkt początkowy konturu rzędna
5	Punkt końcowy konturu odcięta

Parametry	Znaczenie
6	Punkt końcowy konturu rzędna
9	Współrzędna punktu środkowego dla odciętej (tylko w przypadku konturu kołowego)
10	Współrzędna punktu środkowego dla rzędnej (tylko w przypadku konturu kołowego)

Przykład

Określenie punktu przecięcia elementu 3 konturu tablicy TABNAME1 i elementu 7 konturu tablicy TABNAME2. Współrzędne punktu przecięcia w aktywnej płaszczyźnie są zapisywane w zmiennej ISCOORD (1. element = odcięta, 2. element = rzędna). Jeżeli punkt przecięcia nie istnieje, następuje przeskok do KEINSCH (nie znaleziono punktu przecięcia).

```
Kod programu                                          Komentarz
DEF REAL TABNAME1[12,11]                              ; Tablica 1 konturu
DEF REAL TABNAME2[10,11]                              ; Tablica 2 konturu
DEF REAL ISCOORD[2]                                   ; Zmienna współrzędnych punktu przecięcia.
DEF BOOL ISPOINT                                      ; Zmienna statusu punktu przecięcia.
DEF INT MODE                                          ; Zmienna rodzaju obróbki.
…
MODE=1                                                ; Obliczenie niezależnie od aktywnej płaszczyzny.
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7],         ; Wywołanie punktu przecięcia elementów konturu.
ISCOORD,MODE)
N20 IF ISPOINT==FALSE GOTOF KEINSCH                   ; Skok do KEINSCH.
…
```

19.5 Przejście pojedynczymi blokami elementów konturu w tablicy (EXECTAB)

Przy pomocy EXECTAB można pojedynczymi blokami wykonywać elementy konturowe tablicy, które zostały utworzone np. poleceniem CONTPRON.

Składnia

```
EXECTAB(<tablica konturu>[<element konturu>])
```

Znaczenie

EXECTAB:	Predefiniowana procedura wykonania ruchu po elemencie konturu
<tablica konturu>:	Nazwa tablicy konturu
<element konturu>:	Numer elementu konturu

Przykład

Ruchy po elementach konturu 0 do 2 tablicy KTAB mają być wykonywane pojedynczymi blokami.

Kod programu	**Komentarz**
N10 EXECTAB(KTAB[0])	; Ruch po elemencie 0 tablicy KTAB.
N20 EXECTAB(KTAB[1])	; Ruch po elemencie 1 tablicy KTAB.
N30 EXECTAB(KTAB[2])	; Ruch po elemencie 2 tablicy KTAB.

19.6 Obliczenie danych okręgu (CALCDAT)

Przy pomocy `CALCDAT` można z trzech albo czterech znanych punktów okręgu obliczyć promień i współrzędne punktu środkowego okręgu. Podane punkty muszą być różne.

W przypadku 4 punktów, które nie leżą dokładnie na okręgu, jest dla punktu środkowego okręgu i promienia wybierana wartość średnia.

Uwaga

Instrukcja utworzenia wartości średniej

Obliczenie łuku koła jest wykonywane 4 x:

1. z punktem na okręgu 1, 2, 3
2. z punktem na okręgu 1, 2, 4
3. z punktem na okręgu 1, 3, 4
4. z punktem na okręgu 2, 3, 4

Współrzędne punktu środkowego okręgu, wartość odciętej i wartość rzędnej, są obliczane w ten sposób, że wartości odciętej wzgl. rzędnej z 4 obliczeń łuku koła są dodawane i dzielone przez cztery.

Promień jest obliczany w ten sposób, że jest tworzony pierwiastek sumy czterech promieni z obliczeń łuku koła a wynik mnożony przez 0,5.

Składnia

```
<Status>=CALCDAT(<punkty
okręgu>[<liczba>,<rodzaj>],<liczba>,<wynik>)
```

Znaczenie

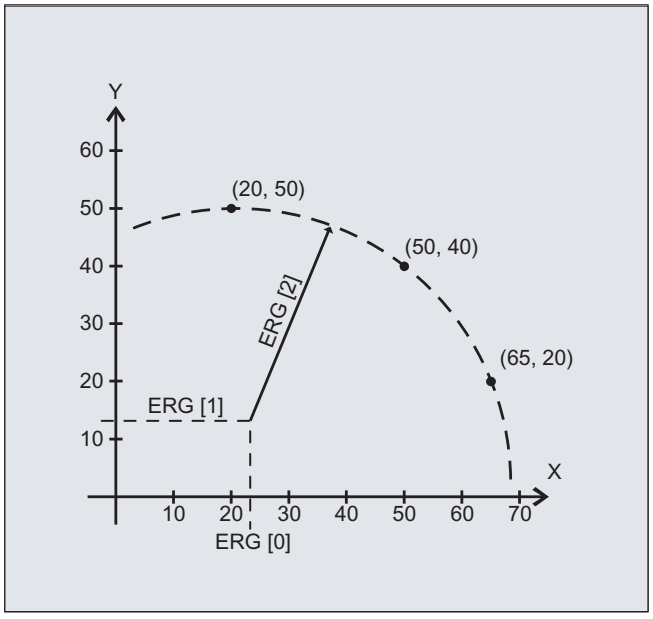
`CALCDAT`:	Predefiniowana funkcja do obliczania promienia i współrzędnych punktu środkowego okręgu z 3 albo 4 punktów.		
`<status>`:	Zmienna statusu obliczenia okręgu		
	Typ:	BOOL	
	Wartość:	TRUE	Podane punkty leżą na okręgu.
		FALSE	Podane punkty **nie** leżą na okręgu.
`<punkty na okręgu>[]`:	Zmienna do podania punktów okręgu z parametrami:		
	`<liczba>`:	Liczba punktów na okręgu (3 albo 4)	
	`<rodzaj>`:	Rodzaj podania współrzędnych, np. 2 dla współrzędnych 2-punktowych	
`<liczba>`:	Parametr liczby punktów stosowanych do obliczenia (3 albo 4)		
`<wynik>[3]`:	Zmienna dla wyniku: Podanie współrzędnych punktu środkowego okręgu i promienia		
	0	Współrzędna punktu środkowego okręgu: Wartość odciętej	
	1	Współrzędna punktu środkowego okręgu: Wartość rzędnej	
	2	Promień	

Uwaga

Należy pamiętać, że zmienne przed ich zastosowaniem muszą być zdefiniowane.

Przykład

Z trzech punktów ma nastąpić obliczenie, czy leżą one na jednym segmencie okręgu.



Kod programu	**Komentarz**
`N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)`	`; Zmienna do podania punktów okręgu.`
`N20 DEF REAL ERG[3]`	`; Zmienna wyniku.`
`N30 DEF BOOL STATUS`	`; Zmienna statusu.`
`N40 STATUS=CALCDAT(PKT,3,ERG)`	`; Wywołanie obliczonych danych okręgu.`
`N50 IF STATUS == FALSE GOTOF ERROR`	`; Skok do błędu.`

19.7 Wyłączenie przygotowania konturu (EXECUTE)

Przy pomocy `EXECUTE` przygotowanie konturu jest wyłączane i równocześnie następuje przełączenie z powrotem na normalny tryb pracy.

Składnia

```
EXECUTE(BŁĄD>)
```

Znaczenie

`EXECUTE`:	Predefiniowana procedura do zakończenia przygotowania konturu	
`<BŁĄD>`:	Zmienna sygnalizacji zwrotnej błędu	
	Typ:	INT
	Wartość zmiennej pokazuje, czy można było bezbłędnie przygotować kontur:	
	0	Błąd
	1	Nie ma błędu

Przykład

Kod programu

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

Zewnętrzne programowanie cykli 20

20.1 Cykle technologiczne

20.1.1 Wprowadzenie

Treść

Ten rozdział zawiera opis cykli dla technologii toczenia, frezowania, szlifowania.

Budowa

Opis cyklu jest zbudowany następująco:

- **Składnia**
  Nazwa cyklu i kolejność wywoływania przekazywanych parametrów
- **Parametry**
  Tablica do objaśnienia poszczególnych parametrów

Opis parametrów

W tablicy podane są do parametru następujące dane: nazwa, opis, zakres wartości i zależności z innymi parametrami.

Kolumna odniesienia do parametru w oknie służy ułatwieniu odnalezienia wartości zaprogramowanych w sterowaniu, przy dekompilacji generowanych zewnętrznie wywołań cykli.

Parametr "tylko dla powierzchni"

W tablicy parametry są oznakowane przez "tylko dla powierzchni". Nie mają one znaczenia dla funkcji cyklu. Są one potrzebne tylko do tego, aby wywołania cykli móc poddać kompletnemu przetwarzaniu wstecznemu. Jeżeli nie są one zaprogramowane, można mimo to poddać cykl przetwarzaniu wstecznemu, pola są wówczas odpowiednio zaznaczone kolorem i muszą zostać wypełnione w oknie.

Parametr "zarezerwowano"

Parametry, które są opisane przez "zarezerwowano", muszą być programowane z 0 albo spacją między przecinkami, aby przyporządkowanie następnych wywoływanych parametrów pasowało do wewnętrznych parametrów cyklu. Wyjątek: W przypadku parametrów string wartość "" albo spacja.

Powtarzanie cykli na szablonie pozycji

Cykle wiercenia i frezowania mogą być powtarzane na szablonie pozycji (wywołania modalne). Przed cyklem należy wówczas w tym samym wierszu napisać `MCALL`, np. `MCALL CYCLE83(...)`.

Uwaga

Jeżeli określone przekazywane parametry (np. `<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>`) są zaprogramowane pośrednio jako parametry, jest przy dekompilacji otwierane okno wprowadzania, nie może jednak zostać zapisane, ponieważ do określonych pól wyboru nie ma jednoznacznego przyporządkowania.

20.1.2 Przegląd technologii

W poniższej tabeli podsumowującej wymienione są wszystkie dostępne programowane zewnętrznie i przyporządkowane do technologii cykle technologiczne.

Technologia	Cykl technologiczny
Wiercenie	• CYCLE81 - wiercenie, nawiercanie (Strona 769) • CYCLE82 - Wiercenie, pogłębianie czołowe (Strona 770) • CYCLE85 - Rozwiercanie (Strona 779) • CYCLE86 - Wytaczanie (Strona 780) • CYCLE83 - Wiercenie otworu głębokiego (Strona 773) • CYCLE830 - Wiercenie otworu głębokiego 2 (Strona 804) • CYCLE84 - Gwintowanie otworu bez oprawki kompensacyjnej (Strona 776) • CYCLE840 - Gwintowanie otworu z oprawką kompensacyjną (Strona 813) • CYCLE78 - Wiercenie otworu z frezowaniem gwintu (Strona 764) • CYCLE802 - Dowolne pozycje (Strona 801) • HOLES1 - Szereg otworów (Strona 730) • CYCLE801 - Siatka albo ramka (Strona 800) • HOLES2 - Okrąg otworów (Strona 731)
Toczenie	• CYCLE951- Skrawanie warstwowe (Strona 824) • CYCLE930 - Rowek (Strona 819) • CYCLE940 - Podcięcie (Strona 821) • CYCLE99 - Toczenie gwintu (Strona 789) • CYCLE98 - Łańcuch gwintów (Strona 785) • CYCLE92 - Przecięcie (Strona 781)

Technologia	Cykl technologiczny
Toczenie konturu	• CYCLE62 - Wywołanie konturu (Strona 749) • CYCLE952 - Toczenie wcinające konturu (poprzeczne) (Strona 827)
Frezowanie	• CYCLE61 - Frezowanie płaszczyzny (Strona 747) • POCKET3 - frezowanie kieszeni prostokątnej (Strona 733) • POCKET4 - Frezowanie kieszeni kołowej (Strona 735) • CYCLE76 - Frezowanie czopa prostokątnego (Strona 760) • CYCLE77 - Frezowanie czopa kołowego (Strona 762) • CYCLE79 - Wielobok (Strona 767) • SLOT1 - rowek wzdłużny (Strona 738) • SLOT2 - Rowek kołowy (Strona 740) • CYCLE899 - Frezowanie rowka otwartego (Strona 816) • LONGHOLE - otwór podłużny (Strona 743) • CYCLE70 - Frezowanie gwintu (Strona 754) • CYCLE60 - Cykl grawerowania (Strona 745)
Frezowanie konturu	• CYCLE62 - Wywołanie konturu (Strona 749) • CYCLE72 - Frezowanie konturu (Strona 756) • CYCLE63 - Frezowanie konturu kieszeni (Strona 750) • CYCLE64 - Wiercenie wstępne konturu kieszeni (Strona 752)
Szlifowanie	• CYCLE495 - Profilowanie (Strona 794) • CYCLE435 - Ustawienie współrzędnych obciągacza (Strona 794) • CYCLE4071 - Szlifowanie wzdłużne z dosuwem w punkcie nawrotu (Strona 833) • CYCLE4072 -Szlifowanie wzdłużne z dosuwem w punkcie nawrotu i sygnał anulowania (Strona 834) • CYCLE4073 - Szlifowanie wzdłużne z dosuwem ciągłym (Strona 838) • CYCLE4074 - Szlifowanie wzdłużne z dosuwem ciągłym i sygnałem przerwania (Strona 839) • CYCLE4075 - Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu (Strona 842) • CYCLE4077 - Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu i przerwania sygnału (Strona 845) • CYCLE4078 -Szlifowanie płaszczyzny z dosuwem ciągłym (Strona 848) • CYCLE4079 -Szlifowanie płaszczyzny z dosuwem przerywanym (Strona 850)
Pozostałe	• CYCLE800 - skręt (Strona 796) • CYCLE832 - High Speed Settings (Strona 810)
Wszystkie	• GROUP_BEGIN - Początek bloku programu (Strona 852) • GROUP_END - Koniec bloku programu (Strona 853) • GROUP_ADDEND - Koniec dodatkowego dosuwu (Strona 853)

20.1.3 HOLES1 - Szereg otworów

Składnia

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	X0	<SPCA>	REAL	Punkt odniesienia dla szeregu otworów w 1. osi (abs)		
2	Y0	<SPCO>	REAL	Punkt odniesienia dla szeregu otworów w 2. osi (abs)		
3	α0	<STA1>	REAL	Podstawowy kąt obrotu (kąt w stosunku do 1. osi)		
4	L0	<FDIS>	REAL	Odstęp 1. otworu od punktu odniesienia		
5	L	<DBH>	REAL	Odstęp między otworami		
6	N	<NUM>	INT	Liczba otworów		
7		<_VARI>	INT	Zarezerwowano		
8		<_UMODE>	INT	Zarezerwowano		
9		<_HIDE>	STRING [200]	Ukryte pozycje • max 198 znaków • Podanie kolejnych numerów pozycji, np. "1,3" (pozycje 1 i 3 nie będą wykonywane)		
10		<_NSP>	INT	Zarezerwowano		
11		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)

20.1.4 HOLES2 - Okrąg otworów

Składnia

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie			
1	X0	<CPA>	REAL	Punkt środkowy dla okręgu otworów w 1. osi (abs) Punkt odniesienia w 1. osi			(w przypadku XY) (w przypadku XA, YB, ZC)
2	Y0	<CPO>	REAL	Punkt środkowy dla okręgu otworów w 2. osi (abs) Punkt odniesienia w 2. osi			(w przypadku XY) (w przypadku XA, YB, ZC)
3	R	<RAD>	REAL	Promień okręgu otworów			(w przypadku XY)
4	α0	<STA1>	REAL	Kąt początkowy albo 1. pozycja osi obrotowej			(w przypadku XY) (w przypadku XA, YB, ZC)
5	α1	<INDA>	REAL	Kąt przełączania (tylko w przypadku łuku koła)			(w przypadku XY, XA, YB, ZC)
					< 0 =	Zgodnie z ruchem wskazówek zegara	
					> 0 =	Przeciwnie do ruchu wskazówek zegara	
6	N	<NUM>	INT	Liczba pozycji			
7		<_VARI>	INT	Rodzaj obróbki			
				JEDNOSTKI:	Zarezerwowano		
				DZIESIĄTKI:	Rodzaj pozycjonowania		
					0 =	ruch do pozycji - liniowy	
					1 =	ruch do pozycji - po torze kołowym	
				SETKI:	Zarezerwowano		
				TYSIĄCE:	Szablon okręgu		
					0 =	tryb kompatybilności, gdy INDA = 0 wówczas pełny okrąg, INDA <> 0 wówczas łuk koła)	
					1 =	Okrąg	
					2 =	Łuk koła	
				DZIESIĄTKI TYSIĘCY:	Szablon pozycji z osią obrotową		
					0 =	XY (bez osi obrotowej)	(w przypadku XY)
					1 =	XA (oś X i oś obrotowa wokół X)	(tylko w przypadku XA)
					2 =	YB (oś Y i oś obrotowa wokół Y)	(w przypadku YB)
					3 =	ZC (oś Z i oś obrotowa wokół C)	(tylko w przypadku ZC)
				JEDOSTKI MILIONÓW + SETKI TYSIĘCY:	Przesunięcie (przy wielu osiach obrotowych wokół tej samej osi; gdy indeks jest za duży, wówczas 1. oś)		
					00 =	1. Oś A, B albo C	
					01 =	2. Oś A, B albo C	
					...		
					10 =	20. Oś A, B albo C	

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
8		<_UMODE>	INT	Zarezerwowano		
9		<_HIDE>	STRING [200]	Zarezerwowano		
10		<_NSP>	INT	Zarezerwowano		
11		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)

20.1.5 POCKET3 - frezowanie kieszeni prostokątnej

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)
4	Z1	<_DP>	REAL	Głębokość kieszeni (abs./ink.), patrz <_AMODE>)
5	L	<_LENG>	REAL	Długość kieszeni (ink., wprowadzić ze znakiem)
6	W	<_WID>	REAL	Szerokość kieszeni (ink., wprowadzić ze znakiem)
7	R	<_CRAD>	REAL	Promień naroża kieszeni
8	X0	<_PA>	REAL	Punkt odniesienia 1. osi (abs)
9	YO	<_PO>	REAL	Punkt odniesienia 2. osi (abs)
10	α0	<_STA>	REAL	Kąt obrotu, kąt między osią wzdłużną (L) i 1. osią
11	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość
12	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie
13	UZ	<_FALD>	REAL	Naddatek na głębokości
14	F	<_FFP1>	REAL	Posuw w płaszczyźnie
15	FZ	<_FFD>	REAL	Posuw dla dosuwu na głębokość

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
16		<_CDIR>	INT	Kierunek frezowania:	0 =	Ruch współbieżny
					1 =	Ruch przeciwbieżny
17		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:		
					1 =	Obróbka zgrubna
					2 =	Obróbka wykańczająca
					4 =	Obróbka wykańczająca obrzeża
					5 =	Fazowanie
				DZIESIĄTKI:		
					0 =	Wiercona wstępnie, dosuw z G0
					1 =	Prostopadły dosuw z G1
					2 =	Spiralnie
					3 =	Ruch wahadłowy po wzdłużnej osi kieszeni
				SETKI:	Zarezerwowano	
18	DXY	<_MIDA>	REAL	Maksymalny dosuw w płaszczyźnie, jednostka patrz <_AMODE>		
19	L1	<_AP1>	REAL	Długość kieszeni półfabrykatu (ink.)		
20	W1	<_AP2>	REAL	Szerokość kieszeni półfabrykatu (ink.)		
21	AZ	<_AD>	REAL	Głębokość kieszeni półfabrykatu (ink.)		
22	ER	<_RAD1>	REAL	Promień toru spiralnego przy zagłębianiu		
	EW			Maksymalny kąt zagłębiania dla ruchu wahadłowego		
23	EP	<_DP1>	REAL	Skok przy zagłębianiu po linii spiralnej		
24		<_UMODE>	INT	Zarezerwowano		
25	FS	<_FS>	REAL	Szerokość fazy (ink.)		
26	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
27		<_GMODE>	INT	Tryb geometrii (ocena zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka/tylko obliczenie punktu startowego	
					0 =	Tryb kompatybilności
					1 =	Normalna obróbka
				TYSIĄCE:	Zwymiarowanie przez środek/naroże	
					0 =	Tryb kompatybilności
					1 =	Zwymiarowanie przez środek
					2 =	Zwymiarowanie punkt narożny, położenie kieszeni +LENG/+WID
					3 =	Zwymiarowanie punkt narożny, położenie kieszeni -LENG/+WID
					4 =	Zwymiarowanie punkt narożny, położenie kieszeni +LENG/-WID
					5 =	Zwymiarowanie punkt narożny, położenie kieszeni -LENG/-WID
				DZIESIĄTKI TYSIĘCY:	Obróbka kompletna/obróbka poprawkowa	
					0 =	Tryb kompatybilności (<_AP1>, <_AP2> i <_AD> traktować jak dotychczas)
					1 =	Obróbka kompletna
					2 =	Obróbka poprawkowa

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
28		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Rodzaj posuwu: Grupa G (G94/G95) dla posuwu dla powierzchni i na głębokość	
					0 =	Tryb kompatybilności
					1 =	Polecenie G jak przed wywołaniem cyklu. G94/G95 dla posuwu dla powierzchni i na głębokość takie same
				SETKI:	---	Zarezerwowano
				TYSIĄCE:	---	Zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	Wprowadzenie: uproszczone
29		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość kieszeni (Z1)	
					0 =	Absolutnie (tryb kompatybilności)
					1 =	Przyrostowo
				DZIESIĄTKI:	Jednostka dla dosuwu w płaszczyźnie (DXY)	
					0 =	mm
					1 =	% średnicy narzędzia
				SETKI:	Głębokość zagłębienia przy fazowaniu (ZFS)	
					0 =	Absolutnie
					1 =	Przyrostowo

20.1.6 POCKET4 - Frezowanie kieszeni kołowej

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)		
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Głębokość kieszeni (abs/ink), patrz <_AMODE>		
5	∅	<_CDIAM>	REAL	Średnica albo promień kieszeni, patrz <_DMODE>		
6	X0	<_PA>	REAL	Punkt odniesienia 1. osi (abs)		
7	Y0	<_PO>	REAL	Punkt odniesienia 2. osi (abs)		
8	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość, patrz <_VARI> = płaszczyznami Maksymalny skok linii spiralnej, patrz <_VARI> = linia spiralna		
9	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie		
10	UZ	<_FALD>	REAL	Naddatek na głębokości		
11	F	<_FFP1>	REAL	Posuw dla obróbki płaszczyzny		
12	FZ	<_FFD>	REAL	Posuw dla dosuwu na głębokość		
13		<_CDIR>	INT	Kierunek frezowania	0 =	Ruch współbieżny
					1 =	Ruch przeciwbieżny
14		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Obróbka	
					1 =	Obróbka zgrubna
					2 =	Obróbka wykańczająca
					4 =	Obróbka wykańczająca obrzeża
					5 =	Fazowanie
				DZIESIĄTKI:	Rodzaj dosuwu (obróbka zgrubna i wykańczająca)	
					0 =	Wiercona wstępnie, dosuw z G0 (kieszeń jest po obróbce wstępnej)
					1 =	Prostopadły dosuw z G1
					2 =	Spiralnie
				SETKI:	Zarezerwowano	
				TYSIĄCE:		
					0 =	Warstwami
					1 =	Spiralnie
15	DXY	<_MIDA>	REAL	Maksymalny dosuw w płaszczyźnie, patrz <_AMODE>, 0 = 0,8 x średnica WZG		
16	∅	<_AP1>	REAL	Średnica/promień kieszeni półfabrykatu (przyr.)		
17	AZ	<_AD>	REAL	Głębokość kieszeni półfabrykatu (przyr.)		
18	ER	<_RAD1>	REAL	Promień toru spiralnego przy zagłębianiu		
19	EP	<_DP1>	REAL	Skok przy zagłębianiu po linii spiralnej		
20		<_UMODE>	INT	Zarezerwowano		
21	FS	<_FS>	REAL	Szerokość fazy (ink.)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
22	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>		
23		<_GMODE>	INT	Tryb geometrii (ocena zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Obróbka/obliczenie punktu startowego	
					0 =	Tryb kompatybilności
					1 =	Normalna obróbka
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Obróbka kompletna/obróbka poprawkowa	
					0 =	Tryb kompatybilności (<_AP1> i <_AD> traktować jak dotychczas)
					1 =	Obróbka kompletna
					2 =	Obróbka poprawkowa
24		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Rodzaj posuwu: Grupa G (G94/G95) dla posuwu dla powierzchni i na głębokość	
					0 =	Tryb kompatybilności
					1 =	Polecenie G jak przed wywołaniem cyklu. G94/G95 dla posuwu dla powierzchni i na głębokość takie same
				SETKI:		
					0 =	Tryb kompatybilności (<_CDIAM>/<_AP1> wprowadzić jako promień)
					1 =	<_CDIAM>/<_AP1> wprowadzić jako średnicę
				TYSIĄCE:	---	Zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	Wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
25		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość kieszeni (Z1)	
					0 =	Absolutnie (tryb kompatybilności)
					1 =	Przyrostowo
				DZIESIĄTKI:	Jednostka szerokości dosuwu (DXY)	
					0 =	mm
					1 =	% średnicy narzędzia
				SETKI:	Głębokość zagłębienia przy fazowaniu (ZFS)	
					0 =	Absolutnie
					1 =	Przyrostowo

20.1.7 SLOT1 - rowek wzdłużny

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<RFP>	REAL	Punkt odniesienia osi narzędzia (abs)
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)
4	Z1	<_DP>	REAL	Głębokość rowka (abs.)
5		<_DPR>	REAL	Głębokość rowka (przyr.), w odniesieniu do Z0 (wprowadzić bez znaku)
6		<NUM>	INT	Liczba rowków = 1
7	L	<LENG>	REAL	Długość rowka
8	W	<WID>	REAL	Szerokość rowka
9	X0	<_CPA>	REAL	Punkt odniesienia 1. oś płaszczyzny
10	Y0	<_CPO>	REAL	Punkt odniesienia, 2. osi płaszczyzny
11		<RAD>	REAL	Zarezerwowano
12	α	<STA1>	REAL	Kąt obrotu
13		<INDA>	REAL	Zarezerwowano
14	FZ	<FFD>	REAL	Posuw dla dosuwu na głębokość
15	F	<FFP1>	REAL	Posuw
16	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
17		<CDIR>	INT	Kierunek frezowania	0 =	ruch współbieżny
					1 =	ruch przeciwbieżny
18	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie albo na obrzeżu rowka		
19		<VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:		
					0 =	Zarezerwowano
					1 =	obróbka zgrubna
					2 =	Obróbka wykańczająca
					4 =	obróbka wykańczająca obrzeża (tylko obrzeża)
					5 =	fazowanie
				DZIESIĄTKI:	Dosunięcie	
					0 =	z nawiercaniem, dosuw z G0 (rowek jest po obróbce wstępnej)
					1 =	prostopadle, dosuw z G1
					2 =	spiralnie
					3 =	ruchem wahadłowym
				SETKI:	Zarezerwowano	
20	DZF	<_MIDF>	REAL	Zarezerwowano		
21	FF	<FFP2>	REAL	Zarezerwowano		
22	SF	<SSF>	REAL	Zarezerwowano		
23	UZ	<_FALD>	REAL	Naddatek na głębokości		
24	ER	<_STA2>	REAL	Promień toru spiralnego przy zagłębianiu		
	EW			Maksymalny kąt wgłębiania dla ruchu wahadłowego		
25	EP	<_DP1>	REAL	Głębokość zagłębienia na obrót dla toru spiralnego		
26		<_UMODE>	INT	Zarezerwowano		
27	FS	<_FS>	REAL	Szerokość fazy (przyr.) tylko przy fazowaniu		
28	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>		
29		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka albo tylko obliczenie punktu startowego	
					1 =	normalna obróbka
				TYSIĄCE:	Wymiarowanie punkt odniesienia, położenie rowka	
					0 =	środek
					1 =	po lewej wewnątrz +L
					2 =	po prawej wewnątrz -L
					3 =	lewe obrzeże +L
					4 =	prawe obrzeże -L

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
30		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Oznaczenie wersji oprogramowania	
					1 =	rozszerzenie funkcji SLOT1
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone
31		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa Z1 (abs/ink)	
					0 =	kompatybilność
					1 =	Z1 (ink)
					2 =	Z1 (abs)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Głębokość zagłębienia przy fazowaniu ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (ink)

Uwaga

Cykl jest w porównaniu z wcześniejszymi wersjami oprogramowania wyposażony w nowe funkcje. Skutkuje to tym, że określone parametry nie są już wyświetlane w oknie wprowadzania (<NUM>, <RAD>, <INDA>NUM, RAD, INDA). Wiele rowków na jednym szablonie pozycji można programować przy pomocy "MCALL" i wywołanie pożądanego szablonu, np. HOLES2.

20.1.8 SLOT2 - Rowek kołowy

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<RFP>	REAL	Punkt odniesienia osi narzędzia (abs)		
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Głębokość rowka (abs.)		
5		<_DPR>	REAL	Głębokość rowka (przyr.), w odniesieniu do Z0 (wprowadzić bez znaku)		
6	N	<NUM>	INT	Liczba rowków		
7	α1	<AFSL>	REAL	Kąt rozwarcia rowka		
8	W	<WID>	REAL	Szerokość rowka		
9	X0	<_CPA>	REAL	Punkt odniesienia = punkt środkowy okręgu, 1. osi płaszczyzny		
10	Y0	<_CPO>	REAL	Punkt odniesienia = punkt środkowy okręgu, 2. osi płaszczyzny		
11	R	<RAD>	REAL	Promień okręgu		
12	α0	<STA1>	REAL	Kąt początkowy		
13	α2	<INDA>	REAL	Kąt przełączania		
14	FZ	<FFD>	REAL	Posuw dla dosuwu na głębokość		
15	F	<FFP1>	REAL	Posuw		
16	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość		
17		<CDIR>	INT	Kierunek frezowania	0 =	ruch współbieżny
					1 =	ruch przeciwbieżny
18	UXY	<_FAL>	REAL	Naddatek na obróbkę wykańczającą w płaszczyźnie albo na obrzeżu		
19		<VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:		
					0 =	obróbka kompletna
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
					3 =	obróbka wykańczająca obrzeża
					5 =	fazowanie
				DZIESIĄTKI:		
					0 =	pozycjonowanie pośrednie z prostą G0
					1 =	pozycjonowanie pośrednie na torze kołowym
				SETKI:	Zarezerwowano	
				TYSIĄCE:		
					0 =	tryb kompatybilności, gdy <INDA> = 0 wówczas pełny okrąg, <INDA> <> 0 wówczas łuk koła)
					1 =	Okrąg
					2 =	łuk koła
20	DZF	<_MIDF>	REAL	Zarezerwowano		
21		<FFP2>	REAL	Zarezerwowano		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
22		<SSF>	REAL	Zarezerwowano		
23	FF	<_FFCP>	REAL	Zarezerwowano		
24		<_UMODE>	INT	Zarezerwowano		
25	FS	<_FS>	REAL	Szerokość fazy (przyr.)		
26	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>		
27		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka albo tylko obliczenie punktu startowego	
					0 =	tryb kompatybilności
					1 =	normalna obróbka
28		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Oznaczenie wersji oprogramowania	
					1 =	funkcje SLOT2 od w. opr. 2.5
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone
29		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa Z1 (abs/ink)	
					0 =	kompatybilność
					1 =	Z1 (ink)
					2 =	Z1 (abs)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Głębokość zagłębienia przy fazowaniu ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (ink)

20.1.9 LONGHOLE - otwór podłużny

Składnia

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>,
<_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>,
<_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)		
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Głębokość otworu podłużnego (abs)		
5		<_DPR>	REAL	Głębokość otworu podłużnego (ink), w odniesieniu do Z0 (wprowadzić bez znaku)		
6		<NUM>	INT	Liczba otworów podłużnych = 1		
7	L	<LENG>	REAL	Długość otworu podłużnego		
8	X0	<_CPA>	REAL	Punkt odniesienia 1. osi płaszczyzny		
9	Y0	<_CPO>	REAL	Punkt odniesienia, 2. osi płaszczyzny		
10		<RAD>	REAL	Zarezerwowano		
11	α0	<STA1>	REAL	Kąt obrotu		
12		<INDA>	REAL	Zarezerwowano		
13	FZ	<FFD>	REAL	Posuw dla dosuwu na głębokość		
14	F	<FFP1>	REAL	Posuw		
15	DZ	<MID>	REAL	Maksymalny dosuw na głębokość		
16		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Rodzaj dosuwu	
					1 =	Prostopadły z G1
					3 =	Ruchem wahadłowym
				SETKI:	Zarezerwowano	
17		<_UMODE>	INT	Zarezerwowano		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
18		<_GMODE>	INT	Tryb geometrii (ocena zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka albo tylko obliczenie punktu startowego	
					0 =	Tryb kompatybilności
					1 =	Normalna obróbka
				TYSIĄCE:	Wymiarowanie punkt odniesienia, położenie rowka	
					0 =	Środek
					1 =	Po lewej wewnątrz +L
					2 =	Po prawej wewnątrz -L
					3 =	Lewe obrzeże +L
					4 =	Prawe obrzeże -L
19		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Rodzaj posuwu: Grupa G (G94/G95) dla posuwu dla powierzchni i na głębokość	
					0 =	Tryb kompatybilności
					1 =	Polecenie G jak przed wywołaniem cyklu. G94/G95 dla posuwu dla powierzchni i na głębokość takie same
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Oznaczenie wersji oprogramowania	
					1 =	Rozszerzenie funkcji LONGHOLE (wymiarowanie punktu odniesienia)
20		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa Z1 (abs/ink)	
					0 =	Kompatybilność
					1 =	Z1 (ink)
					2 =	Z1 (abs)

Uwaga

Cykl jest w porównaniu z wcześniejszymi wersjami oprogramowania wyposażony w nowe funkcje. Skutkuje to tym, że określone parametry nie są już wyświetlane w oknie wprowadzania (<NUM>, <RAD>, <INDA>NUM, RAD, INDA). Wiele rowków na jednym szablonie pozycji można programować przy pomocy "MCALL" i wywołania pożądanego szablonu, np. HOLES2.

20.1.10 CYCLE60 - Cykl grawerowania

Składnia

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1		<_TEXT>	STRING [200]	Tekst do wygrawerowania (maksymalnie 100 znaków)
2	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)
3	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)
4	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do płaszczyzny odniesienia, wprowadzić bez znaku)
5	Z1	<_DP>	REAL	Głębokość (abs), patrz <_AMODE>
6	Z1	<_DPR>	REAL	Głębokość (ink), patrz <_AMODE>
7	X0	<_PA>	REAL	Punkt odniesienia 1. oś płaszczyzny (abs) - prostopadła, patrz <_VARI>
	R			Punkt odniesienia, długość (promień) - w układzie biegunowym, patrz <_VARI>
8	Y0	<_PO>	REAL	Punkt odniesienia, 2. Oś płaszczyzny (abs) - prostopadła, patrz <_VARI>
	α0			Punkt odniesienia, kąt w odniesieniu do 1. osi - w układzie biegunowym, patrz <_VARI>
9	α1	<_STA>	REAL	Kierunek tekstu, kąt linii tekstu w odniesieniu do 1. osi), patrz <_VARI>
10	XM	<_CP1>	REAL	Punkt środkowy okręgu tekstu, 1. oś płaszczyzny (abs) - prostopadła, patrz <_VARI>
	LM			Punkt środkowy okręgu tekstu, długość (promień) w odniesieniu do WNP - w układzie biegunowym, patrz <_VARI>
11	YM	<_CP2>	REAL	Punkt środkowy okręgu tekstu, 2. oś płaszczyzny (abs) - prostopadła, patrz <_VARI>
	αM			Punkt środkowy okręgu tekstu, kąt w odniesieniu do 1. osi - w układzie biegunowym, patrz <_VARI>

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
12	W	<_WID>	REAL	Wysokość znaków (wprowadzić bez znaku)		
13	DX1 DX2	<_DF>	REAL	Odstęp znaków / szerokość całkowita, patrz <_VARI>		
	α2			Kąt rozwarcia, patrz <_VARI>		
14	FZ	<_FFD>	REAL	Posuw dla dosuwu na głębokość, patrz <_DMODE>		
15	F	<_FFP1>	REAL	Posuw dla obróbki płaszczyzny		
16		<_VARI>	INT	Obróbka (orientacja i punkt odniesienia grawerowanego tekstu)		
				JEDNOSTKI:	Punkt odniesienia	
					0 =	Prostokątny
					1 =	Biegunowy
				DZIESIĄTKI:	Zorientowanie tekstu	
					0 =	Tekst na linii
					1 =	Tekst na łuku koła u góry
					2 =	Tekst na łuku koła u dołu
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Punkt odniesienia tekstu poziomo	
					0 =	Z lewej
					1 =	Współśrodkowo
					2 =	Z prawej
				DZIESIĄTKI TYSIĘCY:	Punkt odniesienia tekstu pionowo	
					0 =	Z dołu
					1 =	Współśrodkowo
					2 =	U góry
				SETKI TYSIĘCY:	Długość tekstu	
					0 =	Odstęp znaków
					1 =	Całkowita szerokość tekstu (tylko w przypadku tekstu liniowego)
					2 =	Kąt rozwarcia (tylko w przypadku tekstu na łuku koła)
				MILIONY:	Punkt środkowy okręgu	
					0 =	Prostokątny (kartezjański)
					1 =	Biegunowy
				DZIESIĄTKI MILIONÓW:	Pismo w lustrzanym odbiciu	
					0 =	Kompatybilność
					1 =	Pismo w lustrzanym odbiciu WŁ.
					2 =	Pismo w lustrzanym odbiciu WYŁ.
17		<_CODEP>	INT	Numer strony kodowej pisma (obecnie tylko 1252)		
18		<_UMODE>	INT	Zarezerwowano		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
19		_GMODE>	INT	Tryb geometrii (ocena zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka / tylko obliczenie punktu startowego	
					0 =	Tryb kompatybilności
					1 =	Normalna obróbka
20		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Rodzaj posuwu: Grupa G (G94/G95) dla posuwu dla powierzchni i na głębokość	
					0 =	Tryb kompatybilności
					1 =	Polecenie G jak przed wywołaniem cyklu. G94/G95 dla posuwu dla powierzchni i na głębokość takie same
21		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa (<_DP>, <_DPR>)	
					0 =	Kompatybilność
					1 =	Przyrostowo (<_DPR>)
					2 =	absolutne (<_DP>)

20.1.11 CYCLE61 - Frezowanie płaszczyzny

Składnia

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia, wysokość półfabrykatu (abs)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Wysokość części gotowej (abs/ink), patrz <_AMODE>		
5	X0	<_PA>	REAL	Punkt narożny 1 w 1. osi (abs)		
6	Y0	<_PO>	REAL	Punkt narożny 1 w 2. osi (abs)		
7	X1	<_LENG>	REAL	Punkt narożny 2 w 1. osi (abs/ink), patrz <_AMODE>		
8	Y1	<_WID>	REAL	Punkt narożny 2 w 2. osi (abs/ink), patrz <_AMODE>		
9	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość		
10	DXY	<_MIDA>	REAL	Maksymalny dosuw w płaszczyźnie (jednostka patrz <_AMODE>)		
11	UZ	<_FALD>	REAL	Naddatek na głębokości		
12	F	<_FFP1>	REAL	Posuw roboczy		
13		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Obróbka	
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
				DZIESIĄTKI:	Kierunek obróbki	
					1 =	równolegle do 1. osi, jeden kierunek
					2 =	równolegle do 2. osi, jeden kierunek
					3 =	równolegle do 1. osi, zmienny kierunek
					4 =	równolegle do 2. osi, zmienny kierunek
14		<_LIM>	INT	Ograniczenia		
				JEDNOSTKI:	Ograniczenie 1. oś minus	
					0 =	nie
					1 =	tak
				DZIESIĄTKI:	Ograniczenie 1. oś plus	
					0 =	nie
					1 =	tak
				SETKI:	Ograniczenie 2. oś minus	
					0 =	nie
					1 =	tak
				TYSIĄCE:	Ograniczenie 2. oś plus	
					0 =	nie
					1 =	tak

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
15		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
16		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa (<_DP>)	
					0 =	absolutnie
					1 =	przyrostowo
				DZIESIĄTKI:	Jednostka dla dosuwu w płaszczyźnie (<_MIDA>)	
					0 =	mm
					1 =	% średnicy narzędzia
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Długość płaszczyzny	
					0 =	przyrostowo
					1 =	absolutne
				DZIESIĄTKI TYSIĘCY:	Szerokość płaszczyzny	
					0 =	przyrostowo
					1 =	absolutne

20.1.12 CYCLE62 - Wywołanie konturu

Składnia

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	PRG/CON	<_KNAME>	STRING [140]	Nazwa konturu albo podprogramu, nie musi być programowana w przypadku _TYPE = 2		
2		<_TYPE>	INT	Określenie wprowadzenia konturu		
					0 =	podprogram
					1 =	nazwa konturu
					2 =	etykiety
					3 =	etykiety w podprogramie

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
3	LAB1	<_LAB1>	STRING[32]	Etykieta 1, początek konturu
4	LAB2	<_LAB2>	STRING[32]	Etykieta 2, koniec konturu

20.1.13 CYCLE63 - Frezowanie konturu kieszeni

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	PRG	<_PRG>	STRING [100]	Nazwa programu wybierania materiału		
2		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Technologia obróbki	
					1 =	obróbka zgrubna
					3 =	obróbka wykańczająca dna
					4 =	Obróbka wykańczająca obrzeża
					5 =	fazowanie
				DZIESIĄTKI:	Rodzaj dosuwu	
					0 =	Zagłębianie środkowo
					1 =	Zagłębianie spiralnie
					2 =	Zagłębianie ruchem wahliwym
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Tryb wycofania	
					0 =	wycofanie na płaszczyznę wycofania
					1 =	wycofanie na punkt odniesienia + odstęp bezpieczeństwa
				DZIESIĄTKI TYSIĘCY:	Punkt startowy przy obróbce zgrubnej i wykańczającej dna	
					0 =	auto
					1 =	ręcznie
3	RP	<_RP>	REAL	Płaszczyzna wycofania (abs)		
4	Z0	<_Z0>	REAL	Punkt odniesienia osi narzędzia (abs)		
5	SC	<_SC>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
6	Z1	<_Z1>	REAL	Głębokość końcowa, (patrz <_AMODE> JEDNOSTKI)		
7	F	<_F>	REAL	Posuw w płaszczyźnie obróbka zgrubna/wykańczająca		
8	FZ	<_FZ>	REAL	Posuw dla dosuwu na głębokość		
9	DXY	<_DXY>	REAL	Dosuw w płaszczyźnie - jednostka, (patrz <_AMODE> DZIESIĄTKI)		
10	DZ	<_DZ>	REAL	Dosuw na głębokość		
11	UXY	<_UXY>	REAL	Naddatek w płaszczyźnie		
12	UZ	<_UZ>	REAL	Naddatek na głębokości		
13		<_CDIR>	INT	Kierunek frezowania	0 =	ruch współbieżny
					1 =	ruch przeciwbieżny
14	XS	<_XS>	REAL	Punkt startowy X, absolutnie		
15	YS	<_YS>	REAL	Punkt startowy Y, absolutnie		
16	ER	<_ER>	REAL	Zagłębianie spiralne: Promień		
17	EP	<_EP>	REAL	Zagłębianie spiralne: Skok		
18	EW	<_EW>	REAL	Wgłębianie ruchem wahadłowym: maksymalny kąt zagłębiania		
19	FS	<_FS>	REAL	Szerokość fazy (przyr.) tylko przy fazowaniu		
20	ZFS	<_ZFS>	REAL	Głębokość zagłębienia wierzchołka narzędzia przy fazowaniu, (patrz <_AMODE> SETKI)		
21	TR	<_TR>	STRING[32]	Nazwa narzędzia odniesienia przy obróbce pozostałego materiału		
22	DR	<_DR>	INT	Numer D narzędzia odniesienia przy obróbce pozostałego materiału		
23		<_UMODE>	INT	Zarezerwowano		
24		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka/tylko obliczenie punktu startowego	
					0 =	obróbka normalna (tryb kompatybilności nie jest wymagany)
					1 =	normalna obróbka
					2 =	zarezerwowano

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
25		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Rodzaj technologii	
					1 =	kieszeń
					2 =	czop
				TYSIĄCE:	Obróbka pozostałego materiału	
					0 =	nie
					1 =	tak
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone
26		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa (Z1)	
					0 =	absolutnie (tryb kompatybilności)
					1 =	przyrostowo
				DZIESIĄTKI:	Jednostka dla dosuwu w płaszczyźnie (DXY)	
					0 =	mm
					1 =	% średnicy narzędzia
				SETKI:	Głębokość zagłębienia przy fazowaniu (ZFS)	
					0 =	absolutnie
					1 =	przyrostowo
				TYSIĄCE:	---	zarezerwowano

20.1.14 CYCLE64 - Wiercenie wstępne konturu kieszeni

Składnia

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	PRG	<_PRG>	STRING [100]	Nazwa programu wiercenia/nawiercania		
2		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Tryb wycofania	
					0 =	wycofanie na płaszczyznę wycofania
					1 =	wycofanie na punkt odniesienia + odstęp bezpieczeństwa
3	RP	<_RP>	REAL	Płaszczyzna wycofania (abs)		
4	Z0	<_Z0>	REAL	Punkt odniesienia (abs)		
5	SC	<_SC>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
6	Z1	<_Z1>	REAL	Głębokość wiercenia/nawiercanie, (patrz <_AMODE> JEDNOSTKI)		
7	F	<_F>	REAL	Posuw wiercenia/nawiercania		
8	DXY	<_DXY>	REAL	Dosuw w płaszczyźnie - jednostka, (patrz <_AMODE> DZIESIĄTKI)		
9	UXY	<_UXY>	REAL	Naddatek w płaszczyźnie		
10	UZ	<_UZ>	REAL	Naddatek na głębokości		
11		<_CDIR>	INT	Kierunek frezowania	0 =	ruch współbieżny
					1 =	ruch przeciwbieżny
12	TR	<_TR>	STRING[20]	Nazwa narzędzia odniesienia		
13	DR	<_DR>	INT	Numer D narzędzia odniesienia		
14		<_UMODE>	INT	Zarezerwowano		
15		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka/tylko obliczenie punktu startowego	
					0 =	obróbka normalna (tryb kompatybilności nie jest wymagany)
					1 =	normalna obróbka
					2 =	zarezerwowano

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
25		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	tryb technologii)	
					1 =	wiercenie wstępne
					2 =	nawiercanie
26		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość wiercenia/nawiercania Z1	
					0 =	absolutnie (tryb kompatybilności)
					1 =	przyrostowo
				DZIESIĄTKI:	Jednostka dla dosuwu w płaszczyźnie (DXY)	
					0 =	mm
					1 =	% średnicy narzędzia

20.1.15 CYCLE70 - Frezowanie gwintu

Składnia

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)
4	Z1	<_DP>	REAL	Długość gwintu (abs, ink), patrz <_AMODE> Uwzględnienie wybiegu na dnie otworu (minimum pół skoku)
5	∅	<_DIATH>	REAL	Średnica nominalna gwintu
6	H1	<_H1>	REAL	Głębokość gwintu
7	U	<_FAL>	REAL	Naddatek na obróbkę wykańczającą

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
8	P	<_PIT>	REAL	Skok gwintu (wybór <_PITA>: mm, cale, MODUL, zwojów/cal)		
9	NT	<_NT>	INT	Liczba zębów na płytce skrawającej Długość narzędzia zawsze odniesiona do dolnego zęba!		
10	DXY	<_MID>	REAL	Dosuw maksymalny na przejście <_MID> > <_H1>: wszystko za jednym przejściem		
11	F	<_FFR>	REAL	Posuw frezowania		
12		<_TYPTH>	INT	Typ gwintu	0 =	gwint wewnętrzny
					1 =	gwint zewnętrzny
13	X0	<_PA>	REAL	Punkt środkowy okręgu, 1. oś (abs)		
14	Y0	<_PO>	REAL	Punkt środkowy okręgu, 2. oś (abs)		
15	αS	<_NSP>	REAL	Kąt startowy (Gwint wielozwojny)		
16		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:		
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
				DZIESIĄTKI:		
					1 =	od góry do dołu
					2 =	od dołu do góry
				SETKI:		
					0 =	gwint prawy
					1 =	gwint lewy
17		<_PITA>	INT	Oszacowanie skoku gwintu		
					0 =	tryb kompatybilności
					1 =	skok w mm
					2 =	skok w zwojach na cal (TPI)
					3 =	skok w calach
					4 =	skok jako MODUŁ
18		<_PITM>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu (tylko dla powierzchni)		
19		<_PTAB>	STRING[20]	Łańcuch znaków dla tablicy gwintów ("", "ISO", "BSW", "BSP", "UNC") (tylko dla powierzchni)		
20		<_PTABA>	STRING[20]	Łańcuch znaków dla wyboru w tablicy gwintów (np. "M 10", "M 12", ...) (tylko dla powierzchni)		
21		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Obróbka/obliczenie punktu startowego	
					0 =	tryb kompatybilności
					1 =	normalna obróbka

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
22		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
23		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Długość gwintu (<_DP>)	
					0 =	absolutnie
					1 =	przyrostowo

20.1.16 CYCLE72 - Frezowanie konturu

Składnia

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1		<_KNAME>	STRING [141]	Nazwa podprogramu konturu
2	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)
3	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)
4	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)
5	Z1	<_DP>	REAL	Punkt końcowy, głębokość końcowa (abs/ink) patrz <_AMODE>
6	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość (ink, wprowadzić bez znaku)
7	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie (ink), naddatek na obrzeżu konturu
8	UZ	<_FALD>	REAL	Naddatek na głębokości (ink), naddatek na dnie wprowadzić bez znaku)
9	FX	<_FFP1>	REAL	Posuw po konturze
10	FZ	<_FFD>	REAL	Posuw dla dosuwu na głębokość (lub dosuwu przestrzennego)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
11		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Obróbka	
					1 =	Obróbka zgrubna
					2 =	Obróbka wykańczająca
					5 =	Fazowanie
				DZIESIĄTKI:		
					0 =	Drogi pośrednie z G0
					1 =	Drogi pośrednie z G1
				SETKI:	Wycofanie na końcu konturu	
					0 =	Wycofanie na końcu konturu na punkt odniesienia
					1 =	Wycofanie na końcu konturu na punkt odniesienia + <_SDIS>
					2 =	Wycofanie na końcu konturu o <_SDIS>
					3 =	Bez wycofania na końcu konturu, ruch do następnego punktu startowego następuje z posuwem po konturze
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Obróbka konturu	
					0 =	Obróbka konturu do przodu
					1 =	Obróbka konturu do tyłu Ograniczenia przy obróbce w tył: • max 170 elementów konturu (łącznie z fazami i zaokrągleniami) • są poddawane ocenie tylko wartości w płaszczyźnie (X/Y) i F
12		<_RL>	INT	Kierunek obróbki		
					40 =	Na środku konturu (G40, dosunięcie i odsunięcie: prosta albo prostopadle)
					41 =	Na lewo od konturu (G41, dosunięcie i odsunięcie: prosta albo okrąg)
					42 =	Na prawo od konturu (G42, dosunięcie i odsunięcie: prosta albo okrąg)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
13		<_AS1>	INT	Ruch dosunięcia do konturu		
				JEDNOSTKI:		
					1 =	Prosta
					2 =	Ćwierćokrąg
					3 =	Półokrąg
					4 =	Prostopadłe dosunięcie i odsunięcie
				DZIESIĄTKI:		
					0 =	Ostatni ruch, w płaszczyźnie
					1 =	Ostatni ruch, przestrzennie
14	L1	<_LP1>	REAL	Droga dosunięcia, albo promień dosunięcia (ink, wprowadzić bez znaku)		
15	FZ	<_FF3>	REAL	Posuw dla dróg pośrednich (G94/G95 jak na konturze)		
16		<_AS2>	INT	Ruch odsunięcia od konturu (nie przy prostopadłym dosunięciu/odsunięciu)		
				JEDNOSTKI:		
					1 =	Prosta
					2 =	Ćwierćokrąg
					3 =	Półokrąg
				DZIESIĄTKI:		
					0 =	Ostatni ruch, w płaszczyźnie
					1 =	Ostatni ruch, przestrzennie
17	L2	<_LP2>	REAL	Droga odsunięcia, albo promień odsunięcia (ink, wprowadzić bez znaku)		
18		<_UMODE>	INT	Zarezerwowano		
19	FS	<_FS>	REAL	Szerokość fazy (ink)		
20	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>		
21		<_GMODE>	INT	Tryb geometrii (ocena zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka/tylko obliczenie punktu startowego	
					0 =	Tryb kompatybilności
					1 =	Normalna obróbka

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
22		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Rodzaj posuwu: Grupa G (G94/G95) dla posuwu dla powierzchni i na głębokość	
					0 =	Tryb kompatybilności
					1 =	Polecenie G jak przed wywołaniem cyklu. G94/G95 dla posuwu dla powierzchni i na głębokość takie same
				TYSIĄCE:		
					0 =	Tryb kompatybilności: Nazwa konturu znajduje się w <_KNAME>
					1 =	Nazwa konturu jest programowana w CYCLE62 i przekazywana w _SC_CONT_NAME
23		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Punkt końcowy Z1 (<_DP>)	
					0 =	Absolutnie (tryb kompatybilności)
					1 =	Przyrostowo
				DZIESIĄTKI:	Jednostka dla dosuwu w płaszczyźnie	
					0 =	mm, cale
					1 =	Zarezerwowano
				SETKI:	Głębokość zagłębienia przy fazowaniu (<_ZFS>)	
					0 =	Absolutnie
					1 =	Przyrostowo

Uwaga

Gdy przekazywane parametry poniżej są programowane pośrednio (jako parametry), okno wprowadzania nie jest poddawane przetwarzaniu wstecznemu:

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

20.1.17 CYCLE76 - Frezowanie czopa prostokątnego

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)		
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Głębokość czopa (abs.)		
5		<_DPR>	REAL	Głębokość czopa (przyr.), w odniesieniu do Z0 (wprowadzić bez znaku)		
6	L	<_LENG>	REAL	Długość czopa, patrz <_GMODE> (wprowadzić bez znaku)		
7	W	<_WID>	REAL	Szerokość czopa, patrz <_GMODE> (wprowadzić bez znaku)		
8	R	<_CRAD>	REAL	Promień naroża czopa (wprowadzić bez znaku)		
9	X0	<_PA>	REAL	Punkt odniesienia czopa, 1. oś płaszczyzny (abs)		
10	Y0	<_PO>	REAL	Punkt odniesienia czopa, 2. oś płaszczyzny (abs)		
11	α0	<_STA>	REAL	Kąt obrotu, kąt między osią wzdłużną (L) i 1. osią płaszczyzny		
12	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość (przyr., wprowadzić bez znaku)		
13	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie (przyr.), naddatek na obrzeżu konturu		
14	UZ	<_FALD>	REAL	Naddatek na głębokości (przyr.), naddatek na dnie (wprowadzić bez znaku)		
15	FX	<_FFP1>	REAL	Posuw po konturze		
16	FZ	<_FFD>	REAL	Posuw dla dosuwu na głębokość		
17		<_CDIR>	INT	Kierunek frezowania (wprowadzić bez znaku)		
				JEDNOSTKI:		
					0 =	ruch współbieżny
					1 =	ruch przeciwbieżny
18		<_VARI>	INT	Obróbka		
				JEDNOSTKI:		
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
					5 =	fazowanie
19	L1	<_AP1>	REAL	Długość czopa półfabrykatu		
20	W1	<_AP2>	REAL	Szerokość czopa półfabrykatu		
21	FS	<_FS>	REAL	Szerokość fazy (przyr.)		
22	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs, ink), patrz <_AMODE>		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
23		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka albo tylko obliczenie punktu startowego	
					0 =	tryb kompatybilności
					1 =	normalna obróbka
				TYSIĄCE:	Zwymiarowanie czopa przez środek albo naroże	
					0 =	tryb kompatybilności
					1 =	zwymiarowanie przez środek
					2 =	zwymiarowanie punktu narożnego, czop +L +W
					3 =	zwymiarowanie punktu narożnego, czop -L +W
					4 =	zwymiarowanie punktu narożnego, czop +L -W
					5 =	zwymiarowanie punktu narożnego, czop -L -W
				DZIESIĄTKI TYSIĘCY:	Obróbka kompletna albo obróbka poprawkowa	
					0 =	tryb kompatybilności
					1 =	obróbka kompletna
					2 =	obróbka końcowa
24		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	---	zarezerwowano
				SETKI:	---	zarezerwowano
				TYSIĄCE:	---	zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
25		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa Z1 (DP)	
					0 =	kompatybilność
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Głębokość zagłębienia przy fazowaniu (ZFS)	
					0 =	absolutnie
					1 =	przyrostowo

20.1.18 CYCLE77 - Frezowanie czopa kołowego

Składnia

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)		
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Głębokość czopa (abs.)		
5		<_DPR>	REAL	Głębokość czopa (przyr.), w odniesieniu do Z0 (wprowadzić bez znaku)		
6	∅	<_CDIAM>	REAL	Średnica czopa (wprowadzić bez znaku)		
7	X0	<_PA>	REAL	Punkt odniesienia czopa, 1. oś płaszczyzny (abs)		
8	Y0	<_PO>	REAL	Punkt odniesienia czopa, 2. oś płaszczyzny (abs)		
9	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość (przyr., wprowadzić bez znaku)		
10	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie (przyr.), naddatek na obrzeżu konturu		
11	UZ	<_FALD>	REAL	Naddatek na głębokości (przyr.), naddatek na dnie (wprowadzić bez znaku)		
12	FX	<_FFP1>	REAL	Posuw po konturze		
13	FZ	<_FFD>	REAL	Posuw dla dosuwu na głębokość		
14		<_CDIR>	INT	Kierunek frezowania (wprowadzić bez znaku)		
				JEDNOSTKI:		
					0 =	ruch współbieżny
					1 =	ruch przeciwbieżny

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
15		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Obróbka	
					1 =	obróbka zgrubna aż do naddatku
					2 =	obróbka wykańczająca (naddatek X/Y/Z = 0)
					5 =	fazowanie
16	∅1	<_AP1>	REAL	Średnica czopa półfabrykatu		
17	FS	<_FS>	REAL	Szerokość fazy (przyr.)		
18	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>		
19		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka/tylko obliczenie punktu startowego	
					0 =	tryb kompatybilności
					1 =	normalna obróbka
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Obróbka kompletna/obróbka poprawkowa	
					0 =	Tryb kompatybilności (<_AP1> traktować jak dotychczas)
					1 =	obróbka kompletna
					2 =	obróbka końcowa
20		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	---	zarezerwowano
				SETKI:	---	zarezerwowano
				TYSIĄCE:	---	zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
21		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa Z1 (DP)	
					0 =	absolutnie (tryb kompatybilności)
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Głębokość zagłębienia przy fazowaniu (ZFS)	
					0 =	absolutnie
					1 =	przyrostowo

20.1.19 CYCLE78 - Wiercenie otworu z frezowaniem gwintu

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)
4	Z1	<_DP>	REAL	Końcowa głębokość wiercenia (abs/ink), patrz <_AMODE>
5		<_ADPR>	REAL	Głębokość nawiercania ze zmniejszonym posuwem po torze (ink), działa z <_VARI> DZIESIĄTKI TYSIĘCY
6	D	<_FDPR>	REAL	maksymalny dosuw na głębokość (przyr.) D ≥ Z1 ⇒ dosuw na końcową głębokość wiercenia D < Z1 ⇒ cykl głębokiego wiercenia z wieloma dosuwami i usuwaniem wiórów
7	ZR	<_LDPR>	REAL	Pozostała głębokość wiercenia przy wierceniu przelotowym (przyr.), z posuwem FR
8	∅	<_DIAM>	REAL	Średnica nominalna gwintu
9	P	<_PIT>	REAL	Skok gwintu jako wartość liczbowa

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
10		<_PITA>	INT	Interpretacja skoku gwintu P		
					1 =	skok w mm/obr
					2 =	skok w zwojach/cal
					3 =	skok w calach/obr.
					4 =	skok jako MODUŁ
11	DF	<_DAM>	REAL	Wartość bezwzględna/procentowa dla każdego kolejnego dosuwu (degresja), patrz <_AMODE>		
12	V1	<_MDEP>	REAL	dosuw minimalny (przyr.), działa tylko przy degresji		
13		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Usunięcie wiórów przed frezowaniem gwintu	
					0 =	bez usuwania wiórów przed frezowaniem gwintu (działa tylko na końcową głębokość wiercenia)
					1 =	usuwanie wiórów przed frezowaniem gwintu (działa tylko na końcową głębokość wiercenia)
				SETKI:	Gwint prawy/lewy	
					0 =	gwint prawy
					1 =	gwint lewy
				TYSIĄCE:	Pozostała głębokość wiercenia z posuwem wiercenia	
					0 =	bez pozostałej głębokości wiercenia z posuwem wiercenia FR
					1 =	pozostała głębokość wiercenia z posuwem wiercenia FR
				DZIESIĄTKI TYSIĘCY:	Nawiercanie ze zmniejszonym posuwem	
					0 =	bez nawiercania ze zmniejszonym posuwem
					1 =	Nawiercanie ze zmniejszonym posuwem Posuw nawiercania = 0,3 F1, gdy F1 < 0,15 mm/obr Posuw nawiercania = 0,1 mm/obr, gdy F1 ≥0,15 mm/obr
14		<_CDIR>	INT	Kierunek frezowania	0 =	ruch współbieżny
					1 =	ruch przeciwbieżny
					4 =	ruch przeciwbieżny + współbieżny (kombinacja obróbki zgrubnej + wykańczającej)
15	Z2	<_GE>	REAL	Wielkość wycofania przed frezowaniem gwintu (przyr.)		
16	F1	<_FFD>	REAL	Posuw wiercenia (mm/min wzgl. cali/min albo mm/obr.)		
17	FR	<_FRDP>	REAL	Posuw wiercenia dla pozostałej głębokości wiercenia (mm/min albo mm/obr)		
18	F2	<_FFR>	REAL	Posuw dla frezowania gwintu (mm/min albo mm/ostrze)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
19	FS	<_FFP2>	REAL	Posuw obróbki wykańczającej dla <_CDIR> =4 (mm/min albo mm/ostrze)		
20		<_FFA>	INT	Interpretacja posuwów		
				JEDNOSTKI:	Posuw po torze F1	
				DZIESIĄTKI:	Posuw dla pozostałej głębokości wiercenia FR	
				SETKI:	Posuw dla frezowania gwintu F2	
				TYSIĄCE:	Posuw obróbki wykańczającej FS	
21		<_PITM>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu (tylko dla powierzchni)[1)]		
22		<_PTAB>	STRING[20]	Łańcuch znaków dla tablicy gwintów ("", "ISO", "BSW", "BSP", "UNC") (tylko dla powierzchni)[1)]		
23		<_PTABA>	STRING[20]	Łańcuch znaków dla wyboru w tablicy gwintów (np. "M 10", "M 12", ...) (tylko dla powierzchni)[1)]		
24		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych), zarezerwowano		
25		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
26		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość gwintowania = końcowa głębokość gwintowania Z1 abs/ink	
					0 =	absolutnie
					1 =	przyrostowo
				DZIESIĄTKI:	Wartość bezwzględna/procentowa DF dla każdego kolejnego dosuwu (degresja)	
					0 =	wartość bezwzględna
					1 =	wartość procentowa (0.001 do 100 %)

Uwaga

[1)] Parametry 21, 22 i 23 są stosowane tylko przy wyborze gwintu w tablicach gwintów okna wprowadzania. Dostęp do tablic gwintów poprzez definicję cyklu w czasie wykonywania cyklu jest niemożliwy.

20.1.20 CYCLE79 - Wielobok

Składnia

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)		
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Głębokość wieloboku (abs/ink) patrz <_AMODE>		
5	N	<_NUM>	INT	Liczba krawędzi (1...n)		
6	SW/L	<_SWL>	REAL	Rozstaw klucza albo długość krawędzi (zależnie od <_VARI>) ("SW" w przypadku rozstawu klucza, "L" w przypadku długości krawędzi) Rozstaw klucza tylko przy parzystej liczbie krawędzi i jednej krawędzi		
7	X0	<_PA>	REAL	Punkt odniesienia czopa 1. oś (abs)		
8	Y0	<_PO>	REAL	Punkt odniesienia czopa 2. oś (abs)		
9	α0	<_STA>	REAL	Kąt obrotu środka krawędzi w stosunku do 1. osi (oś X)		
10	R1/FS1	<_RC>	REAL	Zaokrąglenie naroża przy <_NUM> > 2 (promień/faza, patrz <_AMODE>) (ink, wprowadzić bez znaku) ("R1" w przypadku promienia, "FS1" w przypadku fazy)		
11	∅	<_AP1>	REAL	Średnica półfabrykatu czopa		
12	DXY	<_MIDA>	REAL	Maksymalna szerokość dosuwu (jednostka, patrz <_AMODE>)		
13	DZ	<_MID>	REAL	Maksymalny dosuw na głębokość		
14	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie		
15	UZ	<_FALD>	REAL	Naddatek na głębokości		
16	F	<_FFP1>	REAL	Posuw roboczy		
17		<_CDIR>	INT	Kierunek frezowania	0 =	ruch współbieżny
					1 =	ruch przeciwbieżny

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
18		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Obróbka	
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
					3 =	obróbka wykańczająca obrzeża
					5 =	fazowanie
				DZIESIĄTKI:	Rozstaw klucza lub długość krawędzi	
					0 =	rozwartość klucza
					1 =	długość krawędzi
19	FS	<_FS>	REAL	Szerokość fazy (przyr.)		
20	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>		
21		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka albo tylko obliczenie punktu startowego	
					1 =	normalna obróbka
22		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/18/19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	---	zarezerwowano
				SETKI:	---	zarezerwowano
				TYSIĄCE:	---	zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
23		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość końcowa (<_DP>)	
					0 =	absolutnie
					1 =	przyrostowo
				DZIESIĄTKI:	Jednostka dla dosuwu w płaszczyźnie (<_MIDA>)	
					0 =	mm
					1 =	% średnicy narzędzia
				SETKI:	Głębokość zagłębienia przy fazowaniu (<_ZFS>)	
					0 =	absolutnie
					1 =	przyrostowo
				TYSIĄCE:	Zaokrąglenie naroża (<_RC>)	
					0 =	promień
					1 =	faza

20.1.21 CYCLE81 - wiercenie, nawiercanie

Składnia

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)		
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1/∅	<DP>	REAL	Głębokość wiercenia (abs)/średnica nawiercania (abs), patrz <_GMODE>		
5	Z1	<DPR>	REAL	Głębokość wiercenia (przyr.)		
6	DT	<DTB>	REAL	Czas oczekiwania na końcowej głębokości wiercenia, patrz <_AMODE>		
7		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Nawiercanie odniesione do głębokości/średnicy	
					0 =	kompatybilność, głębokość
					1 =	średnica

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
8		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
9		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość wiercenia Z1 (abs/ink)	
					0 =	kompatybilność, z programowania DP/DPR
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Czas oczekiwania DT na końcowej głębokości wiercenia w sekundach/obrotach	
					0 =	kompatybilność, ze znaku DTB (> 0 sekund lub < 0 obrotów)
					1 =	w sekundach
					2 =	w obrotach

20.1.22 CYCLE82 - Wiercenie, pogłębianie czołowe

Składnia

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)
4	Z1	<DP>	REAL	Głębokość wiercenia (abs), patrz <_AMODE>
5	Z1	<DPR>	REAL	Głębokość wiercenia (ink), patrz <_AMODE>
6	DT	<DTB>	REAL	Czas oczekiwania na końcowej głębokości wiercenia, patrz <_AMODE>

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
7		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Głębokość wiercenia odniesiona do wierzchołka/trzonu	
					0 =	kompatybilność, wierzchołek
					1 =	trzon
8		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	zarezerwowano	
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
9		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość wiercenia Z1 (abs./przyr.)	
					0 =	kompatybilność z programowania DP/DPR
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Czas oczekiwania DT na końcowej głębokości wiercenia w sekundach/obrotach	
					0 =	kompatybilność, ze znaku DT (> 0 sekund albo < 0 obrotów)
					1 =	w sekundach
					2 =	w obrotach
				SETKI:	Głębokość nawiercania ZA abs/ink	
					0 =	przyrostowo
					1 =	absolutnie
				TYSIĄCE:	Ewaluacja posuwu nawiercania	
					0 =	w % posuwu wiercenia
					1 =	stóp/min
					2 =	stóp/obr
				DZIESIĄTKI TYSIĘCY:	Pozostała głębokość wiercenia ZD abs/ink	
					0 =	przyrostowo
					1 =	absolutnie
				SETKI TYSIĘCY:	Ewaluacja posuwu pozostałego wiercenia	
					0 =	w % posuwu wiercenia
					1 =	stóp/min
					2 =	stóp/obr
10		<_VARI>	INT	Rodzaj obróbki nawiercanie/przewiercanie		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	zarezerwowano	
				TYSIĄCE:	Wiercenie przelotowe	
					0 =	wiercenie przelotowe "nie"
					1 =	wiercenie przelotowe "tak"
				DZIESIĄTKI TYSIĘCY:	Nawiercanie	
					0 =	nawiercanie „nie”
					1 =	nawiercanie „tak”
11	ZA	<S_ZA>	REAL	Głębokość nawiercania przyrostowo odniesiona do punktu odniesienia albo absolutnie (patrz <_AMODE> SETKI)		
12	FA	<S_FA>	REAL	Posuw nawiercania jako wartość albo w % (w połączeniu z <_AMODE> TYSIĄCE)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
13	ZD	<S_ZD>	REAL	Pozostała głębokość wiercenia odniesiona do końcowej głębokości wiercenia (w połączeniu z <_AMODE> DZIESIĄTKI TYSIĘCY)
14	FD	<S_FD>	REAL	Posuw pozostałego wiercenia jako wartość albo w % (w połączeniu z <_AMODE> SETKI TYSIĘCY)

20.1.23 CYCLE83 - Wiercenie otworu głębokiego

Składnia

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>,
<DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>,
<_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)		
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<DP>	REAL	Końcowa głębokość wiercenia (abs), patrz <_AMODE>		
5	Z1	<DPR>	REAL	Końcowa głębokość wiercenia (ink), patrz <_AMODE>		
6	D	<FDEP>	REAL	1. Głębokość wiercenia (abs), patrz <_AMODE>		
7	D	<FDPR>	REAL	1. Głębokość wiercenia (ink), patrz <_AMODE>		
8	DF	<_DAM>	REAL	Wartość degresywna/procentowa dla każdego kolejnego dosuwu, patrz <_AMODE>		
9	DTB	<DTB>	REAL	Czas oczekiwania na głębokości wiercenia, patrz <_AMODE>		
10	DTS	<DTS>	REAL	Czas oczekiwania w punkcie początkowym (tylko przy usuwaniu wiórów), patrz <_AMODE>		
11	FD1	<FRF>	REAL	Wielkość procentowa posuwu dla pierwszego dosuwu, patrz <_AMODE>		
12		<VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Łamanie wiórów / usuwanie wiórów	
					0 =	łamanie wiórów
					1 =	usuwanie wiórów
13		<_AXN>	INT	Oś narzędzia		
					0 =	3. oś geometryczna
					1 =	1. oś geometryczna
					2 =	2. oś geometryczna
					> 2	3. oś geometryczna
14	V1	<_MDEP>	REAL	Dosuw minimalny (tylko przy wielkości procentowej dla degresji)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
15	V2	<_VRT>	REAL	Wartość wycofania po każdym kroku obróbki (tylko przy łamaniu wiórów)		
					> 0	zmienna wielkość wycofania
					0 =	wartość standardowa 1 mm
16	DT	<_DTD>	REAL	Czas oczekiwania na końcowej głębokości wiercenia, patrz <_AMODE>		
17	V3	<_DIS1>	REAL	Odstęp wcześniejszego zatrzymania (tylko przy usuwaniu wiórów), patrz <_AMODE>		
18		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Głębokość wiercenia odniesiona do wierzchołka/trzonu	
					0 =	wierzchołek
					1 =	trzon
19		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	---	Zarezerwowano
				SETKI:	---	Zarezerwowano
				TYSIĄCE:	---	Zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	Wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
20		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość wiercenia = końcowa głębokość wiercenia Z1 (abs/ink)	
					0 =	kompatybilność, z programowania <DP>/<DPR>
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Czas oczekiwania na głębokości wiercenia DTB w sekundach/obrotach	
					0 =	Kompatybilność ze znaku DTB (> 0 sekund lub < 0 obrotów)
					1 =	w sekundach
					2 =	w obrotach
				SETKI:	Czas oczekiwania w punkcie początkowym DTS w sekundach/obrotach	
					0 =	kompatybilność ze znaku DTS (> 0 sekund albo < 0 obrotów)
					1 =	w sekundach
					2 =	w obrotach
				TYSIĄCE:	Czas czekania na końcowej głębokości wiercenia DTD w sekundach/obrotach	
					0 =	Kompatybilność ze znaku DTD (> 0 sekund albo < 0 obrotów)
					1 =	w sekundach
					2 =	w obrotach
				DZIESIĄTKI TYSIĘCY:	1. Głębokość wiercenia D (abs/ink)	
					0 =	Kompatybilność, z programowania <FDEPF>/<DPR>
					1 =	przyrostowo
					2 =	absolutnie
				SETKI TYSIĘCY:	Wartość degresywna/procentowa <_DAM> dla każdego kolejnego dosuwu	
					0 =	Kompatybilność, ze znaku <_DAM> (> 0 wartość degresywna lub < 0 współczynnik 0,001 do 1,0)
					1 =	Wartość degresywna
					2 =	Wartość procentowa (0,001 do 100 %)
				MILIONY:	Odstęp wcześniejszego zatrzymania V3 automatycznie/ręcznie	
					0 =	kompatybilność ze znaku <_DIS1> (= 0 automatycznie lub > 0 ręcznie)
					1 =	automatycznie (jest obliczane w cyklu)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
					2 =	ręcznie (zaprogramowana wartość)
				DZIESIĄTKI MILIONÓW:	Współczynnik posuwu dla pierwszego dosuwu <FRF> jako współczynnik/wartość procentowa	
					0 =	kompatybilność, jako współczynnik (0,001 do 1,0, FRF = 0 oznacza 100%)
					1 =	Wartość procentowa (0,001 do 999,999 %)

20.1.24 CYCLE84 - Gwintowanie otworu bez oprawki kompensacyjnej

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)		
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<DP>	REAL	Głębokość gwintowania = końcowa głębokość gwintowania (abs), patrz <_AMODE>		
5	Z1	<DPR>	REAL	Głębokość gwintowania = końcowa głębokość gwintowania (ink), patrz <_AMODE>		
6	DT	<DTB>	REAL	Czas oczekiwania na głębokości gwintowania w sekundach		
7	SDE	<SDAC>	INT	Kierunek obrotów po zakończeniu cyklu		
8		<MPIT>	REAL	Wielkość gwintu tylko dla "ISO metrycznego zwykłego" (skok jest obliczany wewnętrznie podczas wykonywania)		
9	P	<PIT>	REAL	Skok gwintu jako wartość, jednostka miary patrz <_PITA>		
10	αS[1)]	<POSS>	REAL	Pozycja wrzeciona dla zorientowanego zatrzymania wrzeciona		
11	S	<SST>	REAL	Prędkość obrotowa wrzeciona dla gwintowania otworu		
12	SR	<SST1>	REAL	Prędkość obrotowa wrzeciona dla wycofania		
13		<_AXN>	INT	Oś gwintowania	0 =	3. Oś geometryczna
					1 =	1. Oś geometryczna
					2 =	2. Oś geometryczna
					≥ 3 =	3. Oś geometryczna

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie			
14		<_PITA>	INT	Jednostka miary skoku gwintu (ewaluacja <PIT> i <MPIT>)			
					0 =	skok w mm	- Ewaluacja <MPIT>/<PIT>
					1 =	skok w mm	- Ewaluacja <PIT>
					2 =	skok w TPI	- Ewaluacja <PIT> (zwojów gwintu na cal)
					3 =	skok w calach	- Ewaluacja <PIT>
					4 =	MODUŁ	- Ewaluacja <PIT>
15		<_TECHNO>	INT	Technologia[1)]			
				JEDNOSTKI:	Zatrzymanie dokładne		
					0 =	Zatrzymanie dokładne aktywne, jak przed wywołaniem cyklu	
					1 =	zatrzymanie dokładne G601	
					2 =	zatrzymanie dokładne G602	
					3 =	zatrzymanie dokładne G603	
				DZIESIĄTKI:	Sterowanie wyprzedzające		
					0 =	z/bez sterowania wyprzedzającego aktywne, jak przed wywołaniem cyklu	
					1 =	ze sterowaniem wyprzedzającym FFWON	
					2 =	bez sterowania wyprzedzającego FFWOF	
				SETKI:	Przyspieszenie		
					0 =	SOFT/BRISK/DRIVE aktywne, jak przed wywołaniem cyklu	
					1 =	z ograniczeniem przyspieszenia drugiego stopnia SOFT	
					2 =	bez ograniczenia przyspieszenia drugiego stopnia BRISK	
					3 =	zredukowane przyspieszenie DRIVE	
				TYSIĄCE:	MCALL praca jako wrzeciono		
					0 =	przy MCALL ponowne uaktywnienie trybu wrzeciona	
					1 =	przy MCALL pozostać w regulacji położenia	

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
16		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	0 =	1 przejście
					1 =	łamanie wiórów (gwintowanie otworów głębokich)
					2 =	usuwanie wiórów (gwintowanie otworów głębokich)
				TYSIĄCE:	ISO/SIEMENS tryb bez znaczenia dla okna wprowadzania	
					0 =	wywołanie z kompatybilności ISO
					1 =	wywołanie z kontekstu SIEMENS
17	D	<_DAM>	REAL	Maksymalny dosuw na głębokość (tylko przy usuwaniu/łamaniu wiórów)		
18	V2	<_VRT>	REAL	Wartość wycofania po każdej obróbce (tylko przy łamaniu wiórów), patrz <_AMODE>		
19		<_PITM>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu[2)]		
20		<_PTAB>	STRING[5]	Łańcuch znaków dla tablicy gwintów ("", "ISO", "BSW", "BSP", "UNC")[2)]		
21		<_PTABA>	STRING[20]	Łańcuch znaków dla wyboru w tablicy gwintów (np. "M 10", "M 12", ...)[2)]		
22		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
23		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Tryb zgodności (tylko dla okna wprowadzania do przetwarzania wstecznego), gdy MD 52216 Bit0 = 1[1)]	
					0 =	Parametry technologii zostaną wyświetlone (kompatybilność): Parametry TECHNOLOGII działają
					1 =	Parametry technologii nie zostaną wyświetlone: Technologia "jak zaprogramowano przed wywołaniem cyklu" działa
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: Uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
24		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość wiercenia = końcowa głębokość wiercenia Z1 (abs/ink)	
					0 =	Kompatybilność, z programowania <DP>/<DPR>
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Kierunek gwintu w prawo/w lewo	
					0 =	kompatybilność, ze znaku PIT/MPIT
					1 =	w prawo
					2 =	w lewo
				DZIESIĄTKI TYSIĘCY:	Zarezerwowano	
				SETKI TYSIĘCY:	Zarezerwowano	
				MILIONY:	Wartość wycofania po każdym dosuwie V2 ręcznie/automatycznie	
					0 =	kompatybilność, z programowania <_VRT> (> 0 wartość zmienna lub ≤ 0 wartość standardowa 1 mm/0.0394 cala)
					1 =	automatycznie (wartość standardowa 1 mm/0,0394 cala)
					2 =	ręcznie (jak zaprogramowano pod V2)

[1] Tablice technologii mogą być ukryte w zależności od danej ustawczej SD52216 $MCS_FUNCTION_MASK_DRILL

[2]Parametry 19, 20 i 21 są stosowane tylko przy wyborze gwintu w tablicach gwintów okna wprowadzania. Dostęp do tablic gwintów poprzez definicję cyklu w czasie wykonywania cyklu jest niemożliwy.

20.1.25 CYCLE85 - Rozwiercanie

Składnia

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
4	Z1	<DP>	REAL	Głębokość wiercenia (abs), patrz <_AMODE>		
5	Z1	<DPR>	REAL	Głębokość wiercenia (ink), patrz <_AMODE>		
6	DT	<DTB>	REAL	Czas oczekiwania na końcowej głębokości wiercenia, patrz <_AMODE>		
7	F	<FFR>	REAL	Posuw		
8	FR	<RFF>	REAL	Posuw przy wycofaniu		
9		<_GMODE>	INT	Zarezerwowano		
10		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
11		<_AMODE>	INT	Tryb alternatywny (wiercenie)		
				JEDNOSTKI:	Głębokość wiercenia Z1 (abs/ink)	
					0 =	kompatybilność, z programowania DP/DPR
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Czas oczekiwania DT na końcowej głębokości wiercenia w sekundach/obrotach	
					0 =	kompatybilność, ze znaku DT (> 0 sekund albo < 0 obrotów)
					1 =	w sekundach
					2 =	w obrotach

20.1.26 CYCLE86 - Wytaczanie

Składnia

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
4	Z1	<DP>	REAL	Głębokość wiercenia (abs), patrz <_AMODE>		
5	Z1	<DPR>	REAL	Głębokość wiercenia (ink), patrz <_AMODE>		
6	DT	<DTB>	REAL	Czas czekania na końcowej głębokości wiercenia, patrz <_AMODE>		
7	DIR	<SDIR>	INT	Kierunek obrotów wrzeciona	3 =	M3
					4 =	M4
8	DX	<RPA>	REAL	Wartość wycofania w kierunku X		
9	DY	<RPO>	REAL	Wartość wycofania w kierunku Y		
10	DZ	<RPAP>	REAL	Wartość wycofania w kierunku Z		
11	SPOS	<POSS>	REAL	Pozycja wrzeciona dla wycofania (zorientowane zatrzymanie wrzeciona, w stopniach)		
12		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Tryb wycofania	
					0 =	wycofanie, kompatybilność
					1 =	bez wycofania
13		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
14		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość wiercenia Z1 (abs/ink)	
					0 =	kompatybilność, z programowania <DP>/<DPR>
					1 =	przyrostowo
					2 =	absolutnie
				DZIESIĄTKI:	Czas oczekiwania DT na końcowej głębokości wiercenia w sekundach/obrotach	
					0 =	kompatybilność, ze znaku DT (> 0 sekund albo < 0 obrotów)
					1 =	w sekundach
					2 =	w obrotach

20.1.27 CYCLE92 - Przecięcie

Składnia

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>,
<_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>,
<_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	X0	<_SPD>	REAL	Punkt odniesienia (abs., zawsze średnica)		
2	Y0	<_SPL>	REAL	Punkt odniesienia (abs)		
3	X1	<_DIAG1>	REAL	Głębokość dla zmniejszenia prędkości obrotowej, patrz <_AMODE> (JEDNOSTKI)		
4	X2	<_DIAG2>	REAL	Głębokość końcowa, patrz <_AMODE> (DZIESIĄTKI)		
5	R/FS	<_RC>	REAL	Promień zaokrąglenia albo szerokość fazy, patrz <_AMODE> (TYSIĄCE)		
6	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
7	S	<_SV1>	REAL	Stała prędkość obrotową wrzeciona (patrz <_AMODE> DZIESIĄTKI TYSIĘCY)		
	V			Stała prędkość skrawania		
8	SV	<_SV2>	REAL	Maksymalna prędkość obrotowa przy stałej prędkości skrawania		
9	DIR	<_SDAC>	INT	Kierunek obrotów wrzeciona	3 =	dla M3
					4 =	dla M4
10	F	<_FF1>	REAL	Posuw do głębokości zmniejszenia prędkości obrotowej		
11	FR	<_FF2>	REAL	Zmniejszony posuw do głębokości końcowej		
12	SR	<_SS2>	REAL	Zmniejszona prędkość obrotowa do głębokości końcowej		
13	XM	<_DIAGM>	REAL	Głębokość wysunięcia chwytaka obrabianych przedmiotów (abs., zawsze średnica)		
14		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Wycofanie	
					0 =	odniesienie do <_SPD> + <_SDIS>
					1 =	bez wycofania na końcu
				DZIESIĄTKI:	Chwytak obrabianych przedmiotów	
					0 =	nie, nie wykonywać polecenia M
					1 =	tak, wywołanie wysunięcia szuflady CUST_TECHCYC(101), zamknięcia szuflady CUST_TECHCYC(102)
15		<_DN>	INT	Numer D dla 2. ostrza, gdy nie zaprogramowano ⇒ D+1		
20		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
21		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość dla zmniejszenia prędkości obrotowej (<_DIAG1>)	
					0 =	absolutnie, wartość osi poprzecznej na średnicy
					1 =	przyrostowo, wartość osi poprzecznej na promieniu
				DZIESIĄTKI:	Głębokość końcowa (<_DIAG2>)	
					0 =	absolutnie, wartość osi poprzecznej na średnicy
					1 =	przyrostowo, wartość osi poprzecznej na promieniu
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Zaokrąglenie/faza (<_RC>)	
					0 =	promień
					1 =	faza
				DZIESIĄTKI TYSIĘCY:	Prędkość obrotowa wrzeciona/prędkość skrawania (<_SV1>)	
					0 =	stała prędkość obrotowa wrzeciona
					1 =	stała prędkość skrawania

20.1.28 CYCLE95 - Skrawanie konturu

Składnia

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	CON	<NPP>	STRING [140]	Nazwa konturu
2	D	<MID>	REAL	Maksymalny dosuw na głębokość przy obróbce zgrubnej, patrz <_GMODE>
3	UZ	<FALZ>	REAL	Naddatek w Z
4	UX	<FALX>	REAL	Naddatek w X
5	U	<FAL>	REAL	Naddatek równolegle do konturu (działa w obu osiach)
6	F	<FF1>	REAL	Posuw dla obróbki zgrubnej
7	FY	<FF2>	REAL	Posuw zagłębiania dla podcięcia
8	FS	<FF3>	REAL	Posuw obróbki wykańczającej

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
9		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI i DZIESIĄTKI:		
					1 =	obróbka zgrubna, wzdłużna, zewnętrzna
					2 =	obróbka zgrubna, poprzeczna, zewnętrzna
					3 =	obróbka zgrubna, wzdłużna, wewnętrzna
					4 =	obróbka zgrubna, poprzeczna, wewnętrzna
					5 =	obróbka wykańczająca, wzdłużna, zewnętrzna
					6 =	6 = obróbka wykańczająca, poprzeczna, zewnętrzna
					7 =	obróbka wykańczająca, wzdłużna, wewnętrzna
					8 =	obróbka wykańczająca, poprzeczna, wewnętrzna
					9 =	obróbka kompletna, wzdłużna, zewnętrzna
					10 =	obróbka kompletna, poprzeczna, zewnętrzna
					11 =	obróbka kompletna, wzdłużna, wewnętrzna
					12 =	obróbka kompletna, poprzeczna, wewnętrzna
				SETKI:		
					0 =	z dociąganiem po konturze, bez pozostających naroży
					1 =	bez dociągania po konturze
					2 =	dociąganie tylko do poprzedniego punktu przecięcia, mogą powstawać naroża
10	DT	<DT>	REAL	Czas oczekiwania przy przerywaniu posuwu		
11	DI	<DAM>	REAL	Ostęp przerw posuwu		
12	VRT	<_VRT>	REAL	Droga cofnięcia od konturu		
					0 =	wewnętrznie jest stosowana droga wycofania 1 mm niezależnie od systemu miar calowego/metrycznego
					> 0 =	droga wycofania

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
13		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Oszacowanie głębokości dosuwu	
					0 =	głębokość dosuwu zostanie przeliczona odpowiednio do Grupy G DIAMON/DIAMOF
					1 =	głębokość dosuwu działa jako wartość promienia (niezależnie od DIAMON/DIAMOF)
14		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				TYSIĄCE:		
					0 =	Tryb kompatybilności: Nazwa konturu znajduje się w NPP
					1 =	Nazwa konturu jest programowana w CYCLE62 i przekazywana w _SC_CONT_NAME

20.1.29 CYCLE98 - Łańcuch gwintów

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	Z0	<_PO1>	REAL	Punkt odniesienia w Z (abs.)
2	X0	<_DM1>	REAL	Punkt odniesienia w X (abs.), na średnicy
3	Z1	<_PO2>	REAL	Punkt pośredni 1 w Z (abs/ink), patrz <_AMODE> (JEDNOSTKI)
4	X1	<_DM2>	REAL	Punkt pośredni 1 w X (abs/ink), patrz <_AMODE> (DZIESIĄTKI) albo
	X1α			Skos gwintu 1 (-90° do 90°) abs. zawsze średnica, przyr. zawsze promień
5	Z2	<_PO3>	REAL	Punkt pośredni 2 w Z (abs/ink), patrz <_AMODE> (SETKI)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
6	X2	<_DM3>	REAL	Punkt pośredni 2 w X (abs/ink), patrz <_AMODE> (TYSIĄCE) albo		
	X2α			Skos gwintu 2 (-90° do 90°) abs. zawsze średnica, przyr. zawsze promień		
7	Z3	<_PO4>	REAL	Punkt końcowy w Z (abs/ink), patrz <_AMODE> (DZIESIĄTKI TYSIĘCY)		
8	X3	<_DM4>	REAL	Punkt końcowy w X (abs/ink), patrz <_AMODE> (SETKI TYSIĘCY) albo		
	X3α			Skos gwintu 3 (-90° do 90°) abs. zawsze średnica, przyr. zawsze promień		
9	LW	<APP>	REAL	Dobieg gwintu (przyr., wprowadzić bez znaku)		
10	LR	<ROP>	REAL	Dobieg gwintu (przyr., wprowadzić bez znaku)		
11	H1	<TDEP>	REAL	Głębokość gwintu (przyr., wprowadzić bez znaku)		
12	U	<FAL>	REAL	Naddatek w X i Z		
13	DP	<_IANG>	REAL	Skos dosuwu jako odstęp albo kąt, patrz <_AMODE> (JEDNOSTKI MILIONÓW)		
	αP			Skos dosuwu działa odpowiednio do ustawienia parametru <_VARI> (SETKI).		
					Definicja trybu kompatybilności _VARI_SETKI = 0:	
					> 0 =	dosuw po jednym zboczu
					0 =	dosuw prostopadle do gwintu
					< 0 =	dosuw po zboczach naprzemiennie
					Definicja dla _VARI_SETKI<>0:	
					> 0 =	dosuw po zboczu dodatnim
					0 =	dosuw po środku
					< 0 =	dosuw po zboczu ujemnym
14	α0	<NSP>	REAL	Przemieszczenie kąta startowego dla 1. zwoju gwintu		
15		<NRC>	INT	Liczba zgrubnych przejść narzędzia, patrz <_VARI> (DZIESIĄTKI TYSIĘCY)		
16	NN	<NID>	INT	Liczba przejść jałowych narzędzia		
17	P0	<_PP1>	REAL	Skok gwintu 1. odcinek gwintu, patrz <_PITA>		
18	P1	<_PP2>	REAL	Skok gwintu 2. odcinek gwintu, patrz <_PITA>		
19	P2	<_PP3>	REAL	Skok gwintu 3. odcinek gwintu, patrz <_PITA>		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
20		<_VARI>	INT	Obróbka		
				JEDNOSTKI:	Technologia	
					1 =	gwint zewnętrzny z dosuwem liniowym
					2 =	gwint wewnętrzny z dosuwem liniowym
					3 =	gwint zewnętrzny z dosuwem degresywnym, przekrój skrawania pozostaje stały
					4 =	gwint wewnętrzny z dosuwem degresywnym, przekrój skrawania pozostaje stały
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Rodzaj dosuwu	
					0 =	Tryb kompatybilności dla <_IANG>
					1 =	dosuw jednostronny
					2 =	dosuw naprzemienny
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Alternatywny dosuw na głębokość	
					0 =	kompatybilność, zadanie liczby przejść zgrubnych (<_NRC>)
					1 =	zadanie wartości dla 1. dosuwu (<_MID>)
				SETKI TYSIĘCY:	Rodzaj obróbki	
					0 =	kompatybilność (obróbka zgrubna i wykańczająca)
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
					3 =	obróbka zgrubna i wykańczająca
				MILIONY:	Kolejność obróbki przy gwincie wielozwojnym	
					0 =	kolejność zwojów rosnąco
					1 =	kolejność zwojów naprzeciwko
21	N	<_NUMTH>	INT	Liczba zwojów gwintu		
22		<_VRT>	REAL	Odstęp ruchu wstecznego (ink)		
					0 =	wewnętrznie jest stosowana droga wycofania 1 mm niezależnie od systemu miar calowego/metrycznego
					> 0 =	droga wycofania
23	D1	<_MID>	REAL	pierwszy dosuw, patrz <_VARI> (DZIESIĄTKI TYSIĘCY)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
24	DA	<_GDEP>	REAL	Zmiana głębokości zwoju (działa tylko przy "wielokrotny")		
					0 =	nie uwzględniać zmiany głębokości zwoju
					> 0 =	uwzględniać zmiany głębokości zwoju
25		<_IFLANK>	REAL	Dosuw po skosie jako szerokość (tylko dla powierzchni)		
26		<_PITA>	INT	Oszacowanie skoku gwintu		
					0 =	tryb kompatybilności dla skoku gwintu, ewaluacja <_PP1> do <_PP3> jak dotychczas odpowiednio do aktywnego systemu metrycznego/calowego
					1 =	skok w mm
					2 =	skok w TPI (zwojów gwintu na cal)
					3 =	skok w calach
					4 =	MODUŁ
27		<_PITM1>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu (tylko dla powierzchni)		
28		<_PITM2>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu (tylko dla powierzchni)		
29		<_PITM3>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu (tylko dla powierzchni)		
30		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	---	zarezerwowano
				SETKI:	---	zarezerwowano
				TYSIĄCE:	---	zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
31		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	1. Punkt pośredni w Z (Z1)	
					0 =	absolutnie
					1 =	przyrostowo
				DZIESIĄTKI:	1. Punkt pośredni w X (X1)	
					0 =	absolutnie
					1 =	przyrostowo
					2 =	α
				SETKI:	2. Punkt pośredni w Z (Z2)	
					0 =	absolutnie
					1 =	przyrostowo
				TYSIĄCE:	2. Punkt pośredni w X (X2)	
					0 =	absolutnie
					1 =	przyrostowo
					2 =	α
				DZIESIĄTKI TYSIĘCY:	Punkt końcowy w Z (Z3)	
					0 =	absolutnie
					1 =	przyrostowo
				SETKI TYSIĘCY:	Punkt końcowy w X (X3)	
					0 =	absolutnie
					1 =	przyrostowo
					2 =	α
				MILIONY:	Wybór dosuwu po skosie jako kąt albo szerokość	
					0 =	kąt dosuwu <_IANG>
					1 =	skos dosuwu <_IFLANK>
				DZIESIĄTKI MILIONÓW:	Jednozwojny/wielozwojny	
					0 =	Tryb kompatybilności (kąt startowy <_NSP> jest uwzględniany)
					1 =	jednozwojny (z przesunięciem kąta startowego <_NSP>)
					2 =	wielozwojny

20.1.30 CYCLE99 - Toczenie gwintu

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	Z0	<_SPL>	REAL	Punkt odniesienia (abs)		
2	X0	<_SPD>	REAL	Punkt odniesienia (abs., zawsze średnica)		
3	Z1	<_FPL>	REAL	Punkt końcowy w połączeniu z <_AMODE> (JEDNOSTKI)		
4	X1	<_FPD>	REAL	Punkt końcowy w połączeniu <_AMODE> (DZIESIĄTKI)		
5	LW/LW2	<_APP>	REAL	Wyprzedzenie gwintu w połączeniu z <_AMODE> (SETKI) lub Dobieg gwintu = wybieg gwintu w połączeniu z <_AMODE> (SETKI)		
6	LR	<_ROP>	REAL	Wybieg gwintu		
7	H1	<_TDEP>	REAL	Głębokość gwintu		
8	U	<_FAL>	REAL	Naddatek w X i Z		
9	DP	<_IANG>	REAL	Skos dosuwu jako odstęp albo kąt, w połączeniu z <_AMODE> (TYSIĄCE)		
	αP				> 0 =	dosuw po zboczu dodatnim
					< 0 =	dosuw po zboczu ujemnym
					0 =	dosuw po środku
10	α0	<_NSP>	REAL	Przemieszczenie kąta startowego (działa tylko przy 1. zwoju)		
11	ND	<_NRC>	INT	Liczba przejść narzędzia przy obróbce zgrubnej, w połączeniu z <_VARI> (DZIESIĄTKI TYSIĘCY)		
12	NN	<_NID>	INT	Liczba przejść jałowych narzędzia		
13	P	<_PIT>	REAL	Skok gwintu jako wartość, w połączeniu z <_PITA>		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
14		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Technologia	
					1 =	gwint zewnętrzny z dosuwem liniowym
					2 =	gwint wewnętrzny z dosuwem liniowym
					3 =	gwint zewnętrzny z dosuwem degresywnym, przekrój skrawania pozostaje stały
					4 =	gwint wewnętrzny z dosuwem degresywnym, przekrój skrawania pozostaje stały
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Rodzaj dosuwu	
					1 =	dosuw jednostronny
					2 =	dosuw naprzemienny
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Alternatywny dosuw na głębokość	
					0 =	zadanie liczby przejść zgrubnych (<_NRC>)
					1 =	zadanie wartości dla 1. dosuwu (<_MID>)
				SETKI TYSIĘCY:	Rodzaj obróbki	
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
					3 =	obróbka zgrubna i wykańczająca
				MILIONY:	Kolejność obróbki przy gwincie wielozwojnym	
					0 =	kolejność zwojów rosnąco
					1 =	kolejność zwojów naprzeciwko
15	N	<_NUMTH>	INT	Liczba zwojów gwintu		
16	VR	<_SDIS>	REAL	Odstęp ruchu powrotnego, przyr.		
17	D1	<_MID>	REAL	Pierwsza głębokość dosuwu, w połączeniu z <_VARI> (DZIESIĄTKI TYSIĘCY)		
18	DA	<_GDEP>	REAL	Zmiana głębokości zwoju (działa tylko przy "wielozwojny")		
					0 =	nie uwzględniać zmiany głębokości zwoju
					> 0 =	uwzględniać zmiany głębokości zwoju
19	G	<_PIT1>	REAL	Zmiana skoku na obrót		
					0 =	skok gwintu jest stały (G33)
					> 0 =	skok gwintu staje się większy (G34)
					< 0 =	skok gwintu staje się mniejszy (G35)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
20		<_FDEP>	REAL	Wartość zagłębiania (wprowadzić bez znaku)		
21	N1	<_GST>	INT	Zwój startowy N1 = 1...N, w połączeniu z <_AMODE> (SETKI TYSIĘCY)		
22		<_GUD>	INT	Zarezerwowano		
23		<_IFLANK>	REAL	Dosuw po skosie jako szerokość (tylko dla powierzchni)		
24		<_PITA>	INT	Jednostka miary skoku gwintu (oszacowanie PIT i/albo MPIT)		
					0 =	skok w mm - ewaluacja MPIT/PIT
					1 =	skok w mm - ewaluacja PIT
					2 =	skok w TPI - ewaluacja PIT (zwojów gwintu na cal)
					3 =	skok w calach - ewaluacja PIT
					4 =	MODUŁ - ewaluacja PIT
25		<_PITM>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu (tylko dla powierzchni)[1)]		
26		<_PTAB>	STRING[20]	Łańcuch znaków dla tablicy gwintów (tylko dla powierzchni)[1)]		
27		<_PTABA>	STRING[20]	Łańcuch znaków dla wyboru w tablicy gwintów (tylko dla powierzchni)[1)]		
28		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Rodzaj gwintu	
					0 =	gwint wzdłużny
					1 =	gwint poprzeczny
					2 =	gwint stożkowy
				SETKI:	---	zarezerwowano
				TYSIĄCE:	---	zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
29		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Długość gwintu w Z	
					0 =	absolutnie
					1 =	przyrostowo
				DZIESIĄTKI:	Długość gwintu w X	
					0 =	absolutnie, wartość osi poprzecznej na średnicy
					1 =	przyrostowo, wartość osi poprzecznej na promieniu
					2 =	α
				SETKI:	Ewaluacja wyprzedzenia/dobiegu <_APP>	
					0 =	wyprzedzenie gwintu <_APP>
					1 =	dobieg gwintu = wybieg gwintu <_APP> = -<_ROP>
					2 =	zadanie drogi dobiegu gwintu <_APP> = -<_APP>
				TYSIĄCE:	Wybór dosuwu po skosie jako kąt albo szerokość	
					0 =	kąt dosuwu <_IANG>
					1 =	skos dosuwu <_IFLANK>
				DZIESIĄTKI TYSIĘCY:	Jednozwojny/wielozwojny	
					0 =	jednozwojny (z przesunięciem kąta startowego <_NSP>)
					1 =	wielozwojny
				SETKI TYSIĘCY:	zwój startowy <_GST>	
					0 =	obróbka kompletna
					1 =	rozpocząć obróbkę od tego zwoju
					2 =	obrobić tylko ten zwój
				MILIONY:	kompensacja ugięcia w przypadku gwintu wzdłużnego	
					0 =	Wysokość segmentu cylindra gwintu XS
					1 =	Promień cylindra gwintu RS
30	XS / RS	<_S_XRS>	REAL	Kompensacja ugięcia w przypadku gwintu wzdłużnego w połączeniu z <_AMODE> JEDNOSTKI MILIONÓW		

Uwaga

[1] Parametry <_PITM>, <_PTAB> i <_PTABA> są stosowane tylko przy wyborze gwintu w tablicach gwintów okna wprowadzania.
Dostęp do tablic gwintów poprzez definicję cyklu w czasie wykonywania cyklu jest niemożliwy.

20.1.31 CYCLE435 - Ustawienie współrzędnych obciągacza

Składnia

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1		<_T>	STRING[32]	Nazwa narzędzia ściernicy		
2		<_DD>	INT	Numer ostrza ściernicy		
3		<S_TA>	STRING[32]	Punkt odniesienia obciągacza - nazwa obciągacza		
4		<S_DA>	INT	Numer ostrza obciągacza		
5		<S_AD>	REAL	Wielkość obciągania średnica		
6		<S_AL>	REAL	Wielkość obciągania poprzecznie		
7		<S_PVD>	REAL	Przesunięcie na profilowanie: średnica		
8		<S_PVL>	REAL	Przesunięcie na profilowanie: poprzecznie		
9		<S_PD>	REAL	Naddatek na profilowanie: średnica		
10		<S_PL>	REAL	Naddatek na profilowanie: poprzecznie		
11		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	aktywne narzędzie na końcu cyklu	
					0 =	Obciągacz aktywny
					1 =	Ściernica aktywna

20.1.32 CYCLE495 - Profilowanie

Składnia

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1		<_T>	STRING[20]	Nazwa narzędzia ściernicy
2		<_DD>	INT	Numer ostrza ściernicy
3		<_SC>	REAL	Wielkość cofnięcia do obchodzenia przeszkód, przyrostowo
4		<_F>	REAL	Posuw profilowania

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
5		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Rodzaj profilowania	
					1 =	równolegle do osi
					2 =	równolegle do konturu
				DZIESIĄTKI:	Kierunek obróbki	
					0 =	ciągnąc możliwe z położeniem ostrza 1 do 4:
					1 =	popychając możliwe z położeniem ostrza 1 do 4:
					2 =	zmienny możliwe z położeniem ostrza 1 do 8:
					3 =	początek → koniec możliwe z położeniem ostrza 1 do 8:
					4 =	koniec → początek możliwe z położeniem ostrza 1 do 8:
				SETKI:	Kierunek dosuwu	
					1 =	Dosuw X- przy G18 lub Y- przy G19
					2 =	Dosuw X+ przy G18 lub Y+ przy G19
					3 =	Dosuw Z- przy G18 i przy G19
					4 =	Dosuw Z+ przy G18 i przy G19
6		<_D>	REAL	Wielkość obciągania dla profilowania: równolegle do osi		
7		<_DX>	REAL	Wartość obciągania X przy G18 lub Y przy G19 dla profilowania: równolegle do krawędzi		
8		<_DZ>	REAL	Wielkość obciągania Z przy G18 lub przy G19 dla profilowania: równolegle do konturu		
9		<S_PA>	REAL	Naddatek na profilowanie		
10		<S_N>	INT	Liczba skoków w programie profilowania		
11		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
12		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Wybór profilowanie nowe/kontynuacja	
					1 =	nowy
					2 =	kontynuacja
				DZIESIĄTKI:	Wybór naddatku na profilowanie	
					0 =	od konturu surowego do najniższego punktu konturu
					1 =	od konturu surowego do najwyższego punktu konturu
13		<S_FW>	REAL	Kąt przystawienia obciągacza		
14		<S_HW>	REAL	Kąt uchwytu obciągacza		

20.1.33 CYCLE800 - skręt

Składnia

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1		<_FR>	INT	Tryb odsunięcia:	0 =	Bez odsunięcia
					1 =	Odsunięcie w osi maszyny Z
					2 =	Odsunięcie w osi maszyny Z, a następnie XY
					3 =	Zarezerwowano
					4 =	Odsunięcie w kierunku narzędzia maksymalne
					5 =	Odsunięcie w kierunku narzędzia przyrostowe
2		<_TC>	STRING[32]	Nazwa zestawu danych skrętu:	""	"" (bez nazwy) gdy jest tylko 1 zestaw danych skrętu
					"0"	Cofnięcie wyboru zestawu danych skrętu (skasowanie frame skrętu)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
3		<_ST>	INT	Status transformacji		
				JEDNOSTKI:		
					0 =	Nowa, płaszczyzna skrętu jest kasowana i obliczana na nowo z aktualnymi parametrami
					1 =	Addytywnia, płaszczyzna skrętu dodaje się addytywnie do aktywnej płaszczyzny skrętu
				DZIESIĄTKI:	Aktualizacja wierzchołka narzędzia tak/nie (aktywna tylko wtedy, gdy w URUCHOMIENIE jest ustawiona funkcja SKRĘT)	
					0 =	Nie aktualizować wierzchołka narzędzia
					1 =	Aktualizować wierzchołek narzędzia (TRAORI)
				SETKI:	Ustawienie / zorientowanie narzędzia (funkcja jest wyświetlana w oknie wprowadzania SKRĘT narzędzia)	
					0 =	Nie ustawiać narzędzia
					1 =	Ustawić narzędzie (preferowany frez kulisty)
					2 =	Zorientować narzędzie **tokarskie** (gdy jest ustawiona kinematyka osi B dla technologii toczenia w IBN skręt)
					3 =	Zorientować narzędzie **frezarskie** (gdy jest ustawiona kinematyka osi B dla technologii toczenia w IBN skręt)
				TYSIĄCE:	Wewnętrzny parametr skrętu w JOG	
				DZIESIĄTKI TYSIĘCY:	Patrz parametr kierunek <_DIR>	
					0 =	Skręt "tak"
					1 =	skręt "nie" kierunek "minus" [3]
					2 =	Skręt "nie" kierunek "plus" [3]
				SETKI TYSIĘCY:	Patrz parametr kierunek <_DIR>	
					0 =	Kompatybilność
					1 =	Wybór kierunku „Minus" zoptymalizowany (tylko dla interfejsu obsługi) [4]
					2 =	Wybór kierunku „Plus" zoptymalizowany (tylko dla interfejsu obsługi) [4]

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
4		<_MODE> 5)	INT	Tryb skrętu: Oszacowanie kąta skrętu i kolejności skrętu (kodowanie bitowe!)		
				Bit: 7 6	0 0:	Kąt skrętu pojedynczymi osiami -> patrz parametry <_A>, <_B>, <_C>
					0 1:	Kąt przestrzenny -> patrz parametry <_A>, <_B> 1)
					1 0:	Kąt projekcji -> patrz parametry <_A>, <_B>, <_C> 1)
					1 1:	Tryb skrętu osie obrotowe bezpośrednio -> patrz parametry <_A>, <_B> 1)
				Bit: 5 4 3 2 1 0 (w przypadku kątów przestrzennych bez znaczenia!)	x x x x 0 1	1.obrót _A wokół X
					x x x x 1 0	1.obrót _A wokół Y
					x x x x 1 1	1.obrót _A wokół Z
					x x 0 1 x x	2.obrót _B wokół X
					x x 1 0 x x	2.obrót _B wokół Y
					x x 1 1 x x	2.obrót _B wokół Z
					0 1 x x x x	3.obrót _C wokół X
					1 0 x x x x	3.obrót _C wokół Y
					1 1 x x x x	3.obrót _C wokół Z
5	X0	<_X0>	REAL	Punkt odniesienia X przed obrotem		
6	Y0	<_Y0>	REAL	Punkt odniesienia Y przed obrotem		
7	Z0	<_Z0>	REAL	Punkt odniesienia Z przed obrotem		
8	X(A)	<_A>	REAL	1. obrót według ustawienia w parametrze <_MODE>		
9	Y(B)	<_B>	REAL	2. obrót według ustawienia w parametrze <_MODE>		
10	Z(C)	<_C>	REAL	3. obrót według ustawienia w parametrze <_MODE>		
11	X1	<_X1>	REAL	Punkt odniesienia X po obrocie		
12	Y1	<_Y1>	REAL	Punkt odniesienia Y po obrocie		
13	Z1	<_Z1>	REAL	Punkt odniesienia Z po obrocie		
14	- albo +	<_DIR>	INT	Wyzwolić ruch postępowy osi obrotowych (domyślnie = -1!)		
					-1 =	Pozycjonować na mniejszą wartość osi obrotowej 1 albo 2 2)
					+1 =	Pozycjonować na większą wartość osi obrotowej 1 albo 2 2)
					0 =	Skręt nie (tylko obliczenie frame skrętu) 1) 3)
15	FR	<_FR_I>	REAL	Wartość (przyr.) odsunięcia w kierunku narzędzia przyrostowo		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
16		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Przedstawienie wartości beta przy orientacji narzędzia	
					0 =	Wartość
					1 =	Strzałka

Uwaga

Gdy przekazywane parametry poniżej są programowane pośrednio (jako parametry), okno wprowadzania nie jest poddawane przetwarzaniu wstecznemu: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

[1)] Wybór jest możliwy, gdy w URUCHOMIENIE jest ustawiona funkcja SKRĘT.

[2)] Wybór jest możliwy, gdy w URUCHOMIENIE SKRĘT jest ustawione odniesienie kierunku do osi obrotowej 1 albo 2.

bez pola wyboru przy odniesieniu kierunku nie

[3)] Wybór skręt "nie" może być ukryty SD 55221 bit 0

Skręt "nie" w kierunku "minus" odpowiada <_DIR> = 0 i _ST DZIESIĄTKI TYSIĘCY = 1

Skręt "nie" w kierunku "plus" odpowiada <_DIR> = 0 i _ST DZIESIĄTKI TYSIĘCY = 2

[4)] Wybór kierunku osi obrotowej 1 albo 2 następuje również, gdy oś obrotowa z odniesieniem kierunku znajduje się w położeniu biegunowym (wartość pozycji równa zero).

[5)] Przykład kodowania: Obrót pojedynczymi osiami, kolejność obrotów ZYX

Binarny: 00011011; dziesiętnie: 27

Identyfikatory osi XYZ odpowiadają osiom geometrycznym kanału NC. Obroty wokół osi XYZ wolno wykonywać pojedynczo. Np. kolejność obrotów ZXZ jest w wywołaniu cyklu CYCLE800 niedozwolona

20.1.34 CYCLE801 - Siatka albo ramka

Składnia

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	X0	<_SPCA>	REAL	Punkt odniesienia dla szablonu pozycji (siatka/ramka) w 1. osi (abs)		
2	Y0	<_SPCO>	REAL	Punkt odniesienia dla szablonu pozycji (siatka/ramka) w 2. osi (abs)		
3	α0	<_STA>	REAL	Podstawowy kąt obrotu (kąt w stosunku do 1. osi)	< 0 =	obrót w kierunku ruchu wskazówek zegara
					> 0 =	obrót przeciwnie do ruchu wskazówek zegara
4	L1	<_DIS1>	REAL	Odstęp kolumn (odstęp pozycji 1. osi, wprowadzić bez znaku liczby)		
5	L2	<_DIS2>	REAL	Odstęp wierszy (odstęp pozycji 2. osi, wprowadzić bez znaku liczby)		
6	N1	<_NUM1>	INT	Liczba kolumn		
7	N2	<_NUM2>	INT	Liczba wierszy		
8		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Szablon pozycji	
					0 =	siatka
					1 =	ramka
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
9		<_UMODE>	INT	Zarezerwowano		
10	αX	<_ANG1>	REAL	Kąt ścięcia w stosunku do 1. osi (położenie skośne wierszy w odniesieniu do 1. osi)		
					< 0 =	pomiar w kierunku ruchu wskazówek zegara (0 do -90 stopni)
					> 0 =	pomiar przeciwnie do kierunku ruchu wskazówek zegara (0 do 90 stopni)
11	αY	<_ANG2>	REAL	Kąt ścięcia w stosunku do 2. osi (położenie skośne kolumn w odniesieniu do 2. osi)		
					< 0 =	pomiar w kierunku ruchu wskazówek zegara (0 do -90 stopni)
					> 0 =	pomiar przeciwnie do kierunku ruchu wskazówek zegara (0 do 90 stopni)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
12		<_HIDE>	STRING [200]	Ukryte pozycje • max 198 znaków • Podanie kolejnych numerów pozycji, np. "1,3" (pozycje 1 i 3 nie będą wykonywane)		
13		<_NSP>	INT	Zarezerwowano		
14		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)

20.1.35 CYCLE802 - Dowolne pozycje

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1		<_XA>	INT	Alternatywa dla wszystkich pozycji X (9-cyfrowa wartość dziesiętna) Liczba miejsc: 876543210 (Miejsce odpowiada pozycji gwintowania Xn)		
				Wartość miejsc:	1 =	absolutnie (1. programowana pozycja zawsze absolutnie)
					2 =	przyrostowo
2		<_YA>	INT	Alternatywa dla wszystkich pozycji Y (9-cyfrowa wartość dziesiętna) Liczba miejsc: 876543210 (Miejsce odpowiada pozycji gwintowania Yn)		
				Wartość miejsc:	1 =	absolutnie (1. programowana pozycja zawsze absolutnie)
					2 =	przyrostowo
3	X0	<_X0>	REAL	1. Pozycja X		
4	Y0	<_Y0>	REAL	1. Pozycja Y		
5	X1	<_X1>	REAL	2. Pozycja X		
6	Y1	<_Y1>	REAL	2. Pozycja Y		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
7	X2	<_X2>	REAL	3. Pozycja X
8	Y2	<_Y2>	REAL	3. Pozycja Y
9	X3	<_X3>	REAL	4. Pozycja X
10	Y3	<_Y3>	REAL	4. Pozycja Y
11	X4	<_X4>	REAL	5. Pozycja X
12	Y4	<_Y4>	REAL	5. Pozycja Y
13	X5	<_X5>	REAL	6. Pozycja X
14	Y5	<_Y5>	REAL	6. Pozycja Y
15	X6	<_X6>	REAL	7. Pozycja X
16	Y6	<_Y6>	REAL	7. Pozycja Y
17	X7	<_X7>	REAL	8. Pozycja X
18	Y7	<_Y7>	REAL	8. Pozycja Y
19	X8	<_X8>	REAL	9. Pozycja X
20	Y8	<_Y8>	REAL	9. Pozycja Y

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
21		<_VARI>	INT	Obróbka		
				SETKI:	(tylko do wywołania z Jobshop) (na razie uwzględniane tylko 0 i 2)	
					0 =	bez zaciskania wrzeciona
					1 =	zaciskać wrzeciono tylko przy zagłębianiu prostopadłym z G00 albo G01
					2 =	zaciskać wrzeciono podczas całej obróbki
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Szablon pozycji z/bez osi obrotowej – kombinacja osi (z <_VARI> SETKI TYSIĘCY)	
					0 =	XY (tylko XY bez osi obrotowej, kompatybilność)
					1 =	X,Y albo Z i oś obrotowa: XA, YB, ZC (1 oś obrotowa z osią geom., wokół której oś obrotowa wykonuje ruch)
					2 =	XY i oś obrotowa: XYA, XYB, XYC (1 oś obrotowa z 1. i 2. osią geometryczną, bez TRACYL)
				SETKI TYSIĘCY:	Oś obrotowa	
					0 =	bez osi obrotowej (nur XY, kompatybilność)
					1 =	oś A (oś obrotowa wokół X)
					2 =	oś B (oś obrotowa wokół Y)
					3 =	oś C (oś obrotowa wokół Z)
				DZIESIĄTKI MILIONÓW + JEDNOSTKI MILIONÓW:	Szablon pozycji z osią obrotową - przesunięcie (przy wielu osiach obrotowych wokół tej samej osi; gdy indeks jest za duży, wówczas 1. osi)	
					00 =	1. oś A, B albo C albo przy kompatybilności
					01 =	2. Oś A, B albo C
					...	
					19 =	20. oś A, B albo C
22		<_UMODE>	INT	Wybór zaciskanego wrzeciona: (tylko dla wywołania z Jobshop) (wywołanie cyklu użytkownika CUST_TECHCYC)		
					3 =	zaciśnięcie/zwolnienie wrzeciona głównego
					23 =	Zaciśnięcie/zwolnienie wrzeciona przechwytującego.

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
23		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
24		<S_ABA>	INT	Alternatywa dla wszystkich pozycji AB (9-cyfrowa wartość dziesiętna) Liczba miejsc: 876543210 (Miejsce odpowiada pozycji ABn)		
				Wartość miejsc:	1 =	absolutnie (1. programowana pozycja zawsze absolutnie)
					2 =	przyrostowo
25	A0	<S_AB0>	REAL	1. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową (w połączeniu z<_VARI>)		
26	A1	<S_AB1>	REAL	2. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		
27	A2	<S_AB2>	REAL	3. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		
28	A3	<S_AB3>	REAL	4. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		
29	A4	<S_AB4>	REAL	5. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		
30	A5	<S_AB5>	REAL	6. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		
31	A6	<S_AB6>	REAL	7. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		
32	A7	<S_AB7>	REAL	8. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		
33	A8	<S_AB8>	REAL	9. pozycja osi obrotowej w przypadku szablonu pozycji z osią obrotową		

Uwaga

Niepotrzebne pozycje parametrów X1/Y1/A1 do X8/Y8/A8 można pominąć. Wartości alternatywne dla <_XA>, <_YA> i <S_ABA> należy jednak podać kompletnie dla wszystkich 9 pozycji.

W przypadku szablonu pozycji XA YB albo ZC (oś geometryczna i oś obrotowa), oś płaszczyzny obróbki, która w szablonie pozycji nie wykonuje ruchu (w przypadku G17 i XA jest to Y) należy wypozycjonować przed wywołaniem cyklu.

20.1.36 CYCLE830 - Wiercenie otworu głębokiego 2

Składnia

```
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>,
<S_SV3>, <S_CON>, <S_COFF>, <_GMODE>, <_DMODE>, <_AMODE>,
<S_AMODE2>, <S_AMODE3>, <S_ZPV>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)		
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, bez znaku)		
4	Z1	<_DP>	REAL	Końcowa głębokość wiercenia (abs/przyr), patrz <_AMODE> JEDNOSTKI)		
5	D	<FDEP>	REAL	Głębokość wiercenia dla łamania wiórów/usuwania wiórów absolutna lub przyrostowa w odniesieniu do punktu odniesienia przy nawiercaniu/ bez nawiercania lub odniesiona do głębokości otworu pilotowego (patrz <_AMODE> DZIESIĄTKI TYSIĘCY)		
6	DF	<_DAM>	REAL	Wartość bezwzględna/procentowa dla każdego kolejnego dosuwu, wartość degresywna albo procentowa (patrz <_AMODE> SETKI TYSIĘCY)		
7	DTB	<DTB>	REAL	Czas oczekiwania na każdej głębokości wiercenia (patrz <_AMODE> DZIESIĄTKI)		
8	DTS	<DTS>	REAL	Czas oczekiwania przy usuwaniu wiórów w punkcie początkowym (patrz <_AMODE> SETKI)		
9	FD1	<FRF>	REAL	Wielkość procentowa posuwu dla pierwszego dosuwu, patrz <_AMODE> DZIESIĄTKI MILIONÓW)		
10		<VARI>	INT	Obróbka		
				JEDNOSTKI:	Łamanie wiórów/usuwanie wiórów	
					0 =	w jednym przejściu
					1 =	łamanie wiórów
					2 =	usuwanie wiórów
					3 =	łamanie i usuwanie wiórów
				DZIESIĄTKI:	Wycofanie przy usuwaniu wiórów	
					0 =	na głębokość otworu pilotowego
					1 =	na odstęp bezpieczeństwa
				SETKI:	Miękkie wejście w materiał	
					0 =	nie
					1 =	tak
				TYSIĄCE:	Wiercenie przelotowe	
					0 =	nie
					1 =	tak
				DZIESIĄTKI TYSIĘCY:	Nawiercenie/otwór pilotowy	
					0 =	bez nawiercenia
					1 =	z nawierceniem
					2 =	z otworem pilotowym
				SETKI TYSIĘCY:	Wycofanie	
					0 =	na głębokość otworu pilotowego
					1 =	na płaszczyznę wycofania
11	V1	<_MDEP>	REAL	Dosuw minimalny przyrostowo (tylko przy wielkości procentowej dla degresji)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
12	V2	<_VRT>	REAL	Wartość wycofania po każdej obróbce przyrostowo - (tylko przy łamaniu wiórów)		
					0 =	wartość standardowa 1 mm
					> 0 =	zmienna wielkość wycofania
13	DT	<_DTD>	REAL	Czas oczekiwania na końcowej głębokości wiercenia, (patrz <_AMODE> TYSIĄCE)		
14	V3	<_DIS1>	REAL	Odstęp wcześniejszego zatrzymania tylko przy usuwaniu wiórów, (patrz <_AMODE> JEDNOSTKI MILIONÓW)		
15	FP	<S_FP>	REAL	Posuw wejścia do otworu pilotowego jako wartość albo w % (w połączeniu z <S_AMODE2> SETKI)		
16		<S_SDAC2>	INT	Kierunek obrotów wrzeciona przy dosunięciu		
					3 =	M3
					4 =	M4
					5 =	M5 (domyślnie)
17	SP	<S_SV2>	REAL	Dosunięcie ze	stałą prędkością obrotową wrzeciona (patrz <S_AMODE2> DZIESIĄTKI MILIONÓW)	
	V4				stałą prędkością skrawania	
				Prędkość obrotowa wrzeciona w % prędkości obrotowej wiercenia		
18	F	<S_FB>	REAL	Posuw wiercenia (patrz <S_AMODE2> JEDNOSTKI)		
19		<_SDAC>	REAL	Kierunek obrotów wrzeciona przy wierceniu		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	Wiercenie ze	stałą prędkością obrotową wrzeciona (patrz <S_AMODE2> JEDNOSTKI MILIONÓW)	
	V5				stałą prędkością skrawania	
21	SPOS	<S_SPOS>	REAL	Pozycja wrzeciona, tylko gdy dosuw z M5		
22	ZA	<S_ZA>	REAL	Głębokość nawiercania przyrostowo odniesiona do punktu odniesienia albo absolutnie (patrz <S_AMODE3> JEDNOSTKI)		
23	FA	<S_FA>	REAL	Posuw nawiercania jako wartość albo w % (w połączeniu z <S_AMODE2> DZIESIĄTKI)		
24	ZP	<S_ZP>	REAL	Otwór pilotowy przyrostowo odniesiony do punktu odniesienia, absolutnie albo współczynnik średnicy otworu (patrz <S_AMODE3> DZIESIĄTKI)		
25	FS	<S_FS>	REAL	Posuw wejścia w materiał jako wartość albo w % (w połączeniu z <S_AMODE2> TYSIĄCE)		
26	ZS1	<S_ZS1>	REAL	Głębokość każdego wejścia w materiał ze stałym posuwem (przyr)		
27	ZS2	<S_ZS2>	REAL	Głębokość każdego wejścia w materiał dla zwiększenia posuwu (przyr)		
28	N	<S_N>	INT	Liczba skoków łamania wiórów przed każdym usunięciem wiórów		
29	ZD	<S_ZD>	REAL	Pozostała głębokość wiercenia przyrostowo odniesiona do końcowej głębokości wiercenia albo absolutnie (w połączeniu z <S_AMODE3> SETKI)		
30	FD	<S_FD>	REAL	Posuw pozostałego wiercenia jako wartość albo w % (w połączeniu z <S_AMODE2> DZIESIĄTKI TYSIĘCY)		
31	FR	<S_FR>	REAL	Posuw wycofania (w połączeniu z <S_AMODE2> SETKI TYSIĘCY)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
32		<S_SDAC3>	INT	Kierunek obrotów wrzeciona przy wycofaniu		
					3 =	M3
					4 =	M4
					5 =	M5
33	SR	<S_SV3>	REAL	Wycofanie ze	stałą prędkością obrotową wrzeciona (patrz <S_AMODE2> SETKI MILIONÓW)	
	V6				stałą prędkością skrawania	
				Prędkość obrotowa wrzeciona w % prędkości obrotowej wiercenia		
34	Chłodziwo wł.	<S_CON>	STRING[10]	Chłodziwo wł., polecenie M albo wywołanie podprogramu		
35	Chłodziwo wył.	<S_COFF>	STRING[10]	Chłodziwo wył., polecenie M albo wywołanie podprogramu		
36		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Głębokość wiercenia odniesiona do wierzchołka/trzonu	
					0 =	wierzchołek
					1 =	trzon
37		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	Wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
38		<_AMODE>	INT	Tryb alternatywny 1		
				JEDNOSTKI:	Głębokość gwintowania = końcowa głębokość gwintowania Z1 abs/przyr	
					0 =	przyrostowo
					1 =	absolutne
				DZIESIĄTKI:	Czas oczekiwania na każdej głębokości wiercenia DTB w sekundach/obrotach	
					0 =	w sekundach
					1 =	w obrotach
				SETKI:	Czas oczekiwania w celu usunięcia wiórów w sekundach/obrotach	
					0 =	w sekundach
					1 =	w obrotach
				TYSIĄCE:	Czas oczekiwania DT na końcowej głębokości wiercenia w sekundach/obrotach	
					0 =	w sekundach
					1 =	w obrotach
				DZIESIĄTKI TYSIĘCY:	1. Głębokość wiercenia D abs/przyr	
					0 =	przyrostowo
					1 =	absolutnie
				SETKI TYSIĘCY:	Wartość bezwzględna/procentowa DF dla każdego kolejnego dosuwu (degresja)	
					0 =	wartość bezwzględna
					1 =	wartość procentowa (0,001 do 100 %)
				MILIONY:	Odstęp wcześniejszego zatrzymania V3 automatycznie/ręcznie	
					0 =	automatycznie (jest obliczane w cyklu)
					1 =	ręcznie (zaprogramowana wartość)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
39		<S_AMODE2>	INT	Tryb alternatywny 2		
				JEDNOSTKI:	JEDNOSTKI: Posuw wiercenia F	
					0 =	stóp/min
					1 =	STÓP/OBR
				DZIESIĄTKI:	Ewaluacja posuwu nawiercania FA	
					0 =	w % posuwu wiercenia
					1 =	stóp/min
					2 =	STÓP/OBR
				SETKI:	Ewaluacja posuwu wejścia w otwór pilotowy FP	
					0 =	w % posuwu wiercenia
					1 =	stóp/min
					2 =	STÓP/OBR
				TYSIĄCE:	Ewaluacja posuwu wejścia w materiał FS	
					0 =	w % posuwu wiercenia
					1 =	stóp/min
					2 =	stóp/obr
				DZIESIĄTKI TYSIĘCY:	Ewaluacja posuwu wiercenia przelotowego FD	
					0 =	w % posuwu wiercenia
					1 =	stóp/min
					2 =	STÓP/OBR
				SETKI TYSIĘCY:	Posuw wycofania FR	
					0 =	stóp/min
					1 =	Posuw szybki
				MILIONY:	Wiercenie - prędkość obrotowa wrzeciona/prędkość skrawania (S/V5)	
					0 =	stała prędkość obrotowa wrzeciona
					1 =	stała prędkość skrawania
				DZIESIĄTKI MILIONÓW:	Dosunięcie z prędkością obrotową wrzeciona/prędkością skrawania (SP/V4)	
					0 =	stała prędkość obrotowa wrzeciona
					1 =	stała prędkość skrawania
					2 =	Prędkość obrotowa wrzeciona w % prędkości obrotowej wiercenia
				SETKI MILIONÓW:	Wycofanie - prędkość obrotowa wrzeciona/prędkość skrawania (SR/V6)	
					0 =	stała prędkość obrotowa wrzeciona
					1 =	stała prędkość skrawania
					2 =	Prędkość obrotowa wrzeciona w % prędkości obrotowej wiercenia

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
40		<S_AMODE3>	INT	Tryb alternatywny 3		
				JEDNOSTKI:	Głębokość nawiercania ZA abs/przyr	
					0 =	przyrostowo
					1 =	absolutne
				DZIESIĄTKI:	Głębokość otworu pilotowego ZP	
					0 =	przyrostowo
					1 =	absolutne
					2 =	współczynnik średnicy otworu
				SETKI:	Pozostała głębokość wiercenia ZD abs/ przyr	
					0 =	przyrostowo
					1 =	absolutne
41	ZPV	<S_ZPV>	REAL	Odstęp wcześniejszego zatrzymania odniesiona przyrostowo do głębokości otworu		

20.1.37 CYCLE832 - High Speed Settings

Składnia

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

Uwaga

CYCLE832 nie zwalnia producenta maszyny od niezbędnych zadań optymalizacyjnych przy uruchamianiu maszyny. Dotyczy to optymalizacji osi uczestniczących w obróbce i ustawień NCU (sterowanie wyprzedzające, ograniczenie przyspieszenia drugiego stopnia, itd.).

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	Tolerancja	<S_TOL>	REAL	Tolerancja konturu Tolerancja konturu odpowiada tolerancji osi geometrycznych.		
2		<S_TOLM>	INT	Rodzaj obróbki (technologia)		
				JEDNOSTKI:		
					0 =	Cofnięcie wyboru
					1 =	Obróbka wykańczająca (finish)
					2 =	Obróbka półwykańczająca (semifinish)
					3 =	Obróbka zgrubna (rough)
				DZIESIĄTKI:		
					0 =	Kompatybilność[1] lub bez tolerancji orientacji
					1 =	Tolerancja orientacji w parametrze <S_OTOL>
				SETKI ... DZIESIĄTKI TYSIĘCY	zajęte z powodu kompatybilności	
				MILIONY:		
					0 =	Kompatybilność. Automatycznie zastosowane zostaną najlepsze będące do dyspozycji funkcje wytwarzania form: • Opcja Top Surface nie aktywna: ⇒ Advanced Surface • Opcja Top Surface nie aktywna: ⇒ Top Surface z wygładzaniem
					1 =	Top Surface bez wygładzania
					2 =	Top Surface z wygładzaniem
3	Tolerancja ORI	<S_OTOL>	REAL	Tolerancja orientacji wzgl. oznaczenie wersji CYCLE832 Parametr tolerancji dla orientacji narzędzia. Jest potrzebny przy wykonywaniu programu obróbki charakteryzującego się dużą prędkością na maszynach z dynamiczną transformacją orientacji (np. obróbka 5-osiowa). Parametr <S_OTOL> **musi** zostać zaprogramowany. Dotyczy to również zastosowań na maszynach 3-osiowych w przypadku programów bez orientacji narzędzia (<S_OTOL> = 1).		

[1] Tolerancja orientacji wyprowadzona z tolerancji konturu pomnożonej przez współczynnik z danych ustawczych cykli SD55441 do SD55443.
Literatura:
Podręcznik uruchomienia Oprogramowanie podstawowe i oprogramowanie obsługowe; SINUMERIK Operate (IM9), rozdział "Konfiguracja funkcji High Speed Setting (CYCLE832)"

Wprowadzanie tekstem jawnym

Aby polepszyć czytelność wywołania cyklu, parametr <S_TOLM> (rodzaj obróbki) można wprowadzić również tekstem jawnym. Teksty jawne nie są zależne od języka. Są dopuszczalne następujące wprowadzenia:

_OFF	dla:	0	Cofnięcie wyboru
_FINISH	dla:	1	Obróbka wykańczająca
_SEMIFIN	dla:	2	Obróbka półwykańczająca
_ROUGH	dla:	3	Obróbka zgrubna
_ORI_FINISH	dla:	11	Obróbka wykańczająca z wprowadzeniem tolerancji orientacji
_ORI_SEMIFIN	dla:	12	Obróbka półwykańczająca z wprowadzeniem tolerancji orientacji
_ORI_ROUGH	dla:	13	Obróbka zgrubna z wprowadzeniem tolerancji orientacji
_TOP_SURFACE_SMOOTH_OFF	dla:	1000000	Top Surface bez wygładzania
_TOP_SURFACE_SMOOTH_ON	dla:	2000000	Top Surface z wygładzaniem

Dla wprowadzenia tekstem jawnym przy Top Surface teksty jawne będą łączone jak w następującym przykładzie:

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

Uwaga

Teksty jawne są oparte o nazwę funkcji G nr 59 (tryb dynamiki dla interpolacji ruchu po torze). Przy pomocy tekstów jawnych jest jasny podział maszyn pod względem ich zastosowania na maszyny 3-osiowe i maszyny z wielokrotną transformacją orientacji (TRAORI).

Cofnięcie CYCLE832

Przy cofnięciu wyboru CYCLE832 parametr <S_TOL> musi zostać przekazany z wartością zero.

Przykład: `CYCLE832(0,0,1)`

Składnia `CYCLE832()` jest dla cofnięcia wyboru CYCLE832 również dozwolona.

Przykłady

Przykład 1: CYCLE832 na maszynie 3-osiowej bez transformacji orientacji

a) Wywoływanie cykli z wprowadzeniem tekstu jawnego

Kod programu	Komentarz
G710	; Systemem miar jest system metryczny.
CYCLE832(0.004,_FINISH,1)	Wywołanie CYCLE832 za pomocą: Tolerancja konturu = 0,004 mm, rodzaj obróbki: Obróbka wykańczająca

Kod programu	Komentarz
...	; Wykonywanie programu obróbki o dużej prędkości

b) Wywoływanie cykli bez wprowadzenia tekstu jawnego

Kod programu	Komentarz
G710	; patrz wyżej
CYCLE832(0.004,1,1)	; patrz wyżej
...	; patrz wyżej

Przykład 2: CYCLE832 na maszynie 5-osiowej z transformacją orientacji

a) Wywoływanie cykli i cofnięcie wyboru z wprowadzeniem tekstu jawnego

Kod programu	Komentarz
G710	; Systemem miar jest system metryczny.
TRAORI	; Uaktywnienie transformacji orientacji.
CYCLE832(0.3,_ORI_ROUGH,0.8)	; Wywołanie CYCLE832 za pomocą: Tolerancja konturu = 0,3 mm, rodzaj obróbki: Obróbka zgrubna z wprowadzeniem tolerancji orientacji, tolerancja orientacji = 0,8 stopnia
...	; Wykonywanie programu obróbki o dużej prędkości
CYCLE832(0,_OFF,1)	; Tolerancja konturu = 0, rodzaj obróbki: cofnięcie wyboru CYCLE832, tolerancja orientacji = 0 stopni

b) Wywoływanie cykli i cofnięcie wyboru bez wprowadzenia tekstu jawnego

Kod programu	Komentarz
G710	; patrz wyżej
TRAORI	; patrz wyżej
CYCLE832(0.3,13,0.8)	; patrz wyżej
...	; patrz wyżej
CYCLE832(0,0,1)	; patrz wyżej

20.1.38 CYCLE840 - Gwintowanie otworu z oprawką kompensacyjną

Składnia

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie			
1	RP	<RTP>	REAL	Płaszczyzna wycofania (abs)			
2	Z0	<RFP>	REAL	Punkt odniesienia (abs)			
3	SC	<SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)			
4	Z1	<DP>	REAL	Głębokość wiercenia (abs), patrz <_AMODE>			
5	Z1	<DPR>	REAL	Głębokość wiercenia (ink), patrz <_AMODE>			
6	DT	<DTB>	REAL	Czas oczekiwania na głębokości wiercenia/na odstępie bezpieczeństwa po wycofaniu, w sekundach, patrz <ENC>			
7		<SDR>	INT	Kierunek obrotów dla wycofania			
8	SDE	<SDAC>	INT	Kierunek obrotów po zakończeniu cyklu			
9		<ENC>	INT	Gwintowanie otworu z przetwornikiem wrzeciona (G33)/gwintowanie otworu bez przetwornika wrzeciona (G63)			
					0 =	z przetwornikiem wrzeciona	- Skok z <MPIT>/<PIT> - bez DT
					20 =	z przetwornikiem wrzeciona	- Skok z <MPIT>/<PIT> - z DT po wycofaniu na odstęp bezpieczeństwa
					11 =	bez przetwornika wrzeciona	- Skok z <MPIT>/<PIT> - z DT na głębokość gwintowania
					1 =	bez przetwornika wrzeciona	- Skok z zaprogramowanego posuwu - z DT na głębokość wiercenia (posuw = prędkość obrotowa skok)
10		<MPIT>	REAL	Wielkość gwintu tylko dla "ISO metrycznego zwykłego" (skok jest obliczany wewnętrznie podczas wykonywania) Zakres wartości: 3 do 48 (dla M3 do M48), alternatywnie do <PIT>			
11		<PIT>	REAL	Skok gwintu jako wartość, jednostka miary patrz <_PITA> Zakres wartości: > 0, alternatywnie do MPIT			
12		<_AXN>	INT	Oś gwintowania	0 =	3. Oś geometryczna	
					1 =	1. Oś geometryczna	
					2 =	2. Oś geometryczna	
					≥ 3 =	3. Oś geometryczna	

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie			
13		<_PITA>	INT	Jednostka miary skoku gwintu (ewaluacja <PIT> i <MPIT>)			
					0 =	skok w mm	- Ewaluacja <MPIT>/<PIT>
					1 =	skok w mm	- Ewaluacja <PIT>
					2 =	skok w TPI	- Ewaluacja <PIT> (zwojów gwintu na cal)
					3 =	skok w calach	- Ewaluacja <PIT>
					4 =	MODUŁ	- Ewaluacja <PIT>
14		<_TECHNO>	INT	Technologia[1]			
				JEDNOSTKI:	Zatrzymanie dokładne		
					0 =	Zatrzymanie dokładne aktywne, jak przed wywołaniem cyklu	
					1 =	zatrzymanie dokładne G601	
					2 =	zatrzymanie dokładne G602	
					3 =	zatrzymanie dokładne G603	
				DZIESIĄTKI:	Sterowanie wyprzedzające		
					0 =	z/bez sterowania wyprzedzającego aktywne, jak przed wywołaniem cyklu	
					1 =	ze sterowaniem wyprzedzającym FFWON	
					2 =	bez sterowania wyprzedzającego FFWOF	
15		<_PITM>	STRING[15]	Łańcuch znaków jako znacznik dla wprowadzenia skoku gwintu[2]			
16		<_PTAB>	STRING[5]	Łańcuch znaków dla tablicy gwintów ("", "ISO", "BSW", "BSP", "UNC")[2]			
17		<_PTABA>	STRING[20]	Łańcuch znaków dla wyboru w tablicy gwintów (np. "M 10", "M 12", ...)[2]			
18		<_GMODE>	INT	Zarezerwowano			

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
19		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Tryb zgodności (tylko dla okna wprowadzania do przetwarzania wstecznego), gdy MD 52216 Bit0 = 1[1)]	
					0 =	Parametry technologii zostaną wyświetlone (kompatybilność): Parametry TECHNOLOGII działają
					1 =	Parametry technologii nie zostaną wyświetlone: Technologia "jak zaprogramowano przed wywołaniem cyklu" działa
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone
20		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość wiercenia Z1 (abs/ink)	
					0 =	Kompatybilność, z programowania <DP>/<DPR>
					1 =	przyrostowo
					2 =	absolutne

[1)] Tablice technologii mogą być ukryte w zależności od danej ustawczej SD52216 $MCS_FUNCTION_MASK_DRILL

[2)]Parametry 15, 16 i 17 są stosowane tylko przy wyborze gwintu w tablicach gwintów okna wprowadzania. Dostęp do tablic gwintów poprzez definicję cyklu w czasie wykonywania cyklu jest niemożliwy!

20.1.39 CYCLE899 - Frezowanie rowka otwartego

Składnia

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	RP	<_RTP>	REAL	Płaszczyzna wycofania (abs)		
2	Z0	<_RFP>	REAL	Punkt odniesienia osi narzędzia (abs)		
3	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (addytywnie do punktu odniesienia, wprowadzić bez znaku)		
4	Z1	<_DP>	REAL	Głębokość rowka (abs/ink), patrz <_AMODE>		
5	L	<_LENG>	REAL	Długość rowka (przyr.)		
6	W	<_WID>	REAL	Szerokość rowka (przyr.)		
7	X0	<_PA>	REAL	Punkt odniesienia/pozycja startowa 1. osi (abs)		
8	Y0	<_PO>	REAL	Punkt odniesienia/pozycja startowa 2. osi (abs)		
9	α0	<_STA>	REAL	Kąt obrotu w stosunku do 1. osi		
10	DZ	<_MID>	REAL	Maksymalna głębokość dosuwu (przyr.), tylko dla frezowania trochoidalnego		
11	DXY	<_MIDA>	REAL	Maksymalny dosuw w płaszczyźnie, patrz <_AMODE>		
12	UXY	<_FAL>	REAL	Naddatek w płaszczyźnie		
13	UZ	<_FALD>	REAL	Naddatek na głębokości		
14	F	<_FFP1>	REAL	Posuw		
15		<_CDIR>	INT	Kierunek frezowania		
				JEDNOSTKI:		
					0 =	ruch współbieżny
					1 =	ruch przeciwbieżny
					4 =	Na przemian
16		<_VARI>	INT	Obróbka		
				JEDNOSTKI:		
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
					3 =	obróbka wykańczająca dna
					4 =	Obróbka wykańczająca obrzeża
					5 =	obróbka półwykańczająca
					6 =	fazowanie
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Zarezerwowano	
				TYSIĄCE:		
					1 =	frezowanie trochoidalne
					2 =	frezowanie posuwem wgłębnym

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
17		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka/tylko obliczenie punktu startowego	
					1 =	normalna obróbka
				TYSIĄCE:	Zwymiarowanie przez środek/krawędź	
					0 =	zwymiarowanie przez środek
					1 =	zwymiarowanie przez krawędź "lewą" ("-" kierunek 1. osi)
					2 =	zwymiarowanie przez krawędź "prawą" ("+" kierunek 1. osi)
18		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	---	zarezerwowano
				SETKI:	---	zarezerwowano
				TYSIĄCE:	---	zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	wprowadzenie: uproszczone
19		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Głębokość rowka Z1	
					0 =	absolutnie
					1 =	przyrostowo
				DZIESIĄTKI:	Jednostka dla dosuwu w płaszczyźnie (<_MIDA>) DXY	
					0 =	mm
					1 =	% średnicy narzędzia
				SETKI:	Głębokość zagłębienia przy fazowaniu ZFS	
					0 =	absolutnie
					1 =	przyrostowo
20		<_UMODE>	INT	Zarezerwowano		
21	FS	<_FS>	REAL	Szerokość fazy (przyr.)		
22	ZFS	<_ZFS>	REAL	Głębokość zagłębienia (wierzchołka narzędzia) przy fazowaniu (abs/ink), patrz <_AMODE>)		

20.1.40 CYCLE930 - Rowek

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1	X0	<_SPD>	REAL	Punkt odniesienia w osi poprzecznej (zawsze średnica)		
2	Z0	<_SPL>	REAL	Punkt odniesienia w osi wzdłużnej		
3	B1	<_WIDG>	REAL	Szerokość rowka na dnie		
4	B2	<_WIDG2>	REAL	Szerokość rowka u góry (tylko dla powierzchni)		
5	T1	<_DIAG>	REAL	Głębokość rowka w punkcie odniesienia, przy abs. i obróbka wzdłużna = średnica, i innym przypadku przyr.		
6	T2	<_DIAG2>	REAL	Głębokość zagłębienia w punkcie odniesienia (tylko dla powierzchni), przy abs. i obróbka wzdłużna = średnica, i innym przypadku przyr.		
7	α0	<_STA>	REAL	Kąt skosu (-180 ≤ <_STA> ≤ 180)		
8	α1	<_ANG1>	REAL	Kąt zbocza 1 (0 ≤ <_ANG1> < 90) na stronie rowka określonej przez punkt odniesienia		
9	α2	<_ANG2>	REAL	Kąt zbocza 2 (0 ≤ <_ANG2> < 90) po drugiej stronie w stosunku do punktu odniesienia		
10	R1/FS1	<_RCO1>	REAL	Promień zaokrąglenia albo szerokość fazy 1, na zewnątrz na punkcie odniesienia		
11	R2/FS2	<_RCI1>	REAL	Promień zaokrąglenia albo szerokość fazy 2, wewnątrz na punkcie odniesienia		
12	R3/FS3	<_RCI2>	REAL	Promień zaokrąglenia albo szerokość fazy 3, wewnątrz po drugiej stronie w stosunku do punktu odniesienia		
13	R4/FS4	<_RCO2>	REAL	Promień zaokrąglenia albo szerokość fazy 4, zewnątrz po drugiej stronie w stosunku do punktu odniesienia		
14	U	<_FAL>	REAL	Naddatek w X i Z, patrz <_VARI> (DZIESIĄTKI TYSIĘCY) (wprowadzić bez znaku)		
15	D	<_IDEP1>	REAL	Maksymalny dosuw na głębokość przy zagłębianiu (wprowadzić bez znaku)		
					0 =	1. Przejście narzędzia bezpośrednio na całej głębokości
					> 0 =	1. przejście <_IDEP1>, 2. przejście 2 · <_IDEP1> itd.
16	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa (wprowadzić bez znaku)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
17		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Technologia obróbki	
					1 =	obróbka zgrubna
					2 =	obróbka wykańczająca
					3 =	Obróbka zgrubna i wykańczająca
				SETKI:	Położenie wzdłużne/poprzeczne zewnętrzne/wewnętrzne +Z/+Z wzgl. +X/-X	
					1 =	wzdłużne/zewnętrzne +Z
					2 =	poprzeczne/wewnętrzne -X
					3 =	wzdłużne/wewnętrzne +Z
					4 =	poprzeczne/wewnętrzne +X
					5 =	wzdłużne/zewnętrzne - Z
					6 =	poprzeczne/zewnętrzne -X
					7 =	wzdłużne/wewnętrzne -Z
					8 =	poprzeczne/zewnętrzne +X
				TYSIĄCE:	Położenie punktu odniesienia	
					0 =	Punkt odniesienia u góry
					1 =	Punkt odniesienia u dołu
				DZIESIĄTKI TYSIĘCY:	Określenie zachowania się naddatków na obróbkę wykańczającą	
					0 =	naddatek na obr. wykańczającą U równolegle do konturu
					1 =	naddatki na obr. wykańczającą UX i UZ oddzielnie
18		<_DN>	INT	Numer D dla 2. ostrza narzędzia		
					> 0 =	numer D dla korekcji narzędzia 2. ostrza noża do toczenia poprzecznego
					0 =	nie zaprogramowano 2. ostrza
19	N	<_NUM>	INT	Liczba rowków (0 = 1 rowek)		
20	DP	<_DBH>	REAL	Liczba rowków (wymagana tylko wtedy, gdy <_NUM> > 1)		
21	F	<_FF1>	REAL	Posuw		
22		<_NR>	INT	Oznaczenie kształtu rowka odpowiada pionowym przyciskom programowym do wyboru kształtu		
					0 =	zbocza 90° bez faz/zaokrągleń
					1 =	skośne zbocza z fazami/zaokrągleniami (bez α0)
					2 =	jak 1, ale na stożku (z α0)
23	UX	<_FALX>	REAL	Naddatek w osi X, patrz <_VARI> (DZIESIĄTKI TYSIĘCY) (wprowadzić bez znaku)		
24	UZ	<_FALZ>	REAL	Naddatek w osi Z, patrz <_VARI> (DZIESIĄTKI TYSIĘCY) (wprowadzić bez znaku)		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
25		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
26		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Zwymiarowanie głębokości (tylko dla powierzchni)	
					0 =	w punkcie odniesienia
					1 =	po stronie przeciwnej do punktu odniesienia
				DZIESIĄTKI:	Głębokość	
					0 =	absolutnie
					1 =	przyrostowo
				SETKI:	Zwymiarowanie szerokości (tylko dla powierzchni)	
					0 =	na średnicy zewnętrznej (u góry)
					1 =	na średnicy wewnętrznej (u dołu)
				TYSIĄCE:	Promień/faza 1 (<_RCO1>)	
					0 =	promień
					1 =	faza
				DZIESIĄTKI TYSIĘCY:	Promień/faza 2 (<_RCI1>)	
					0 =	promień
					1 =	faza
				SETKI TYSIĘCY:	Promień/faza 3 (<_RCI2>)	
					0 =	promień
					1 =	faza
				MILIONY:	Promień/faza 4 (<_RCO2>)	
					0 =	promień
					1 =	faza

20.1.41 CYCLE940 - Podcięcie

Przy pomocy cyklu CYCLE940 można programować różne kształty podcięć. Te podcięcia znacznie różnią się one pod względem parametryzacji.

Dodatkowe kolumny tablicy pokazują, które parametry są potrzebne przy danym kształcie podcięcia. Odpowiadają one pionowym przyciskom programowym wyboru w oknie cykli:

- E: Podcięcie kształt E
- F: Podcięcie kształt F

- A-D: Podcięcie gwintu DIN (kształty A-D)
- T: Podcięcie gwintu (dowolna definicja kształtu)

Składnia

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>,
<_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>,
<_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Okno parametrów	Parametry wewnętrzne	Typ danych	Programowanie w przypadku kształtu				Znaczenie		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	Punkt odniesienia w osi poprzecznej (zawsze średnica)		
2	Z0	<_SPL>	REAL	x	x	x	x	Punkt odniesienia osi wzdłużnej (abs.)		
3	KSZTAŁT	<_FORM>	CHAR	x	x	x	x	Kształt podcięcia (duże litery, np. "T") Wybór, z której tablicy mają być brane podcięcia		
									A =	Zewnątrz, odniesienie DIN76, A = normalne
									B =	Zewnątrz, odniesienie DIN76, B = krótkie
									C =	Wewnątrz, odniesienie DIN76, C = normalne
									D =	Wewnątrz, odniesienie DIN76, D = skrócone
									E =	Odniesienie DIN509
									F =	Odniesienie DIN509
									T=	Dowolny kształt
4	POŁOŻENIE	<_LAGE>	INT	x	x	x	x	Położenie podcięcia (równolegle do Z)	0 =	Zewnętrzne +Z: ____|
									1 =	Zewnętrzne -Z: |____/
									2 =	Wewnętrzne +Z: /-----|
									3 =	Wewnętrzne -Z: |-----\
5	SC	<_SDIS>		x	x	x	x	Odstęp bezpieczeństwa (ink)		
6	F	<_FFP>		x	x	x	x	Posuw dla obróbki (mm/obr.)		

Nr	Okno parametrów	Parametry wewnętrzne	Typ danych	Programowanie w przypadku kształtu				Znaczenie			
7		<_VARI>	INT	-	-	x	x	Rodzaj obróbki			
								JEDNOSTKI:	Obróbka		
									1 =	Obróbka zgrubna	
									2 =	Obróbka wykańczająca	
									3 =	Obróbka zgrubna + obróbka wykańczająca	
								DZIESIĄTKI:	Strategia obróbki		
									0 =	Równolegle do konturu	
									1 =	Wzdłużnie	
				Podcięcia kształt E i F są zawsze obrabiane w jednym przejściu, jak przy obróbce wykańczającej.							
8	X1	<_EPD>		x	x	-	-	Naddatek X (abs/ink), patrz <_AMODE>			
				-	-	-	x	Głębokość podcięcia (abs/ink), patrz <_AMODE>			
9	Z1	<_EPL>		-	x	-	-	Naddatek Z			
				-	-	-	x	Szerokość podcięcia (abs/ink), patrz <_AMODE>			
10	R1	<_R1>		-	-	-	x	Promień zaokrąglenia na skosach			
11	R2	<_R2>		-	-	-	x	Promień zaokrąglenia w narożu			
12	α	<_STA>		-	-	x	x	Kąt zagłębiania			
13	VX	<_VRT>		x	x	-	-	Podcięcie X (abs/ink), patrz <_AMODE>			
				-	-	x	x	Podcięcie X przy obróbce wykańczającej, (abs/ink), patrz <_AMODE>			
14	D	<_MID>		-	-	x	x	Dosuw na głębokość			
15	U	<_FAL>		-	-	x	x	Naddatek równoległy do konturu, patrz <_AMODE>			
16	UX	<_FALX>		-	-	x	x	Naddatek na obróbkę wykańczającą X			
17	UZ	<_FALZ>		-	-	x	x	Naddatek na obróbkę wykańczającą Z			
18	P	<_PITI>	INT	-	-	x	-	Wybór skoku, kształt A-D, odpowiada 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	-	-	Wybór promień/głębokość, kształt 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]					Łańcuch znaków dla tablicy gwintów ("", "ISO", "BSW", "BSP", "UNC") (tylko dla powierzchni)			
20		<_PTABA>	STRING [20]					Łańcuch znaków dla wyboru w tablicy gwintów (np. "M 10", "M 12", ...) (tylko dla powierzchni)			

Nr	Okno parametrów	Parametry wewnętrzne	Typ danych	Programowanie w przypadku kształtu				Znaczenie		
21		<_DMODE>	INT					Tryb wyświetlania		
				x	x	x	x	JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
									0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
									1 =	G17 (aktywna tylko w cyklu)
									2 =	G18 (aktywna tylko w cyklu)
									3 =	G19 (aktywna tylko w cyklu)
22		<_AMODE>	INT					Tryb alternatywny		
				x	x	-	x	JEDNOSTKI:	Parametr <_EPD> naddatek X albo głębokość podcięcia	
									0 =	Absolutnie (zawsze średnica)
									1 =	Przyrostowo
				x	x	-	x	DZIESIĄTKI:	Parametr <_EPL> naddatek Z albo szerokość podcięcia	
									0 =	Absolutnie
									1 =	Przyrostowo
				x	x	x	x	SETKI:	Parametr <_VRT> podcięcie X	
									0 =	Absolutnie (zawsze średnica)
									1 =	Przyrostowo
				-	-	x	x	TYSIĄCE:	Naddatek na obróbkę wykańczającą	
									0 =	Naddatek na obróbkę wykańczającą równolegle do konturu (<_FAL>)
									1 =	Naddatek oddzielnie (<_FALX>/<_FALZ>)

20.1.42 CYCLE951- Skrawanie warstwowe

Składnia

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>,
<_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	X0	<_SPD>	REAL	Punkt odniesienia (abs., zawsze średnica)
2	Z0	<_SPL>	REAL	Punkt odniesienia (abs)
3	X1	<_EPD>	REAL	Punkt końcowy

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
4	Z1	<_EPL>	REAL	Punkt końcowy		
5	XM α1 α2	<_ZPD>	REAL	Punkt pośredni, patrz <_DMODE> (DZIESIĄTKI)		
6	ZM α1 α2	<_ZPL>	REAL	Punkt pośredni, patrz <_DMODE> (DZIESIĄTKI)		
7	Położenie	<_LAGE>	INT	Położenie obrabianego narożа	0 =	zewnątrz/z tyłu
					1 =	zewnątrz/z przodu
					2 =	zewnątrz/z tyłu
					3 =	wewnątrz/z przodu
8	D	<_MID>	REAL	Maksymalny dosuw na głębokość przy zagłębianiu		
9	UX	<_FALX>	REAL	Naddatek w X		
10	UZ	<_FALZ>	REAL	Naddatek w Z		
11		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Kierunek skrawania (wzdłużnie lub poprzecznie) w układzie współrzędnych	
					1 =	Wzdłużnie
					2 =	Poprzecznie
				DZIESIĄTKI:		
					1 =	Obróbka zgrubna do naddatku na obróbkę wykańczającą
					2 =	Obróbka wykańczająca
				SETKI:	Zarezerwowano	
				TYSIĄCE:	Zarezerwowano	
				DZIESIĄTKI TYSIĘCY:	Zarezerwowano	
12	R1/FS1	<_RF1>	REAL	Promień zaokrąglenia albo szerokość fazy 1, patrz <_AMODE> (DZIESIĄTKI TYSIĘCY)		
13	R2/FS2	<_RF2>	REAL	Promień zaokrąglenia albo szerokość fazy 2, patrz <_AMODE> (SETKI TYSIĘCY)		
14	R3/FS3	<_RF3>	REAL	Promień zaokrąglenia albo szerokość fazy 3, patrz <_AMODE> (JEDNOSTKI MILIONÓW)		
15	SC	<_SDIS>	REAL	Odstęp bezpieczeństwa		
16	F	<_FF1>	REAL	Posuw dla obróbki zgrubnej/wykańczającej		
17		<_NR>	INT	Oznaczenie rodzaju skrawania (odpowiada pionowym przyciskom programowym do wyboru kształtu):		
					0 =	skrawanie 1, naroże 90 stopni bez faz/zaokrągleń
					1 =	skrawanie 2, naroże 90 stopni z fazami/zaokrągleniami
					2 =	skrawanie 3, dowolne naroże z fazami/zaokrągleniami

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
18		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Forma wprowadzenia <_ZPD>/<_ZPL>	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
					5 =	α1/α2
21		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Punkt pośredni w X	
					0 =	absolutnie, wartość osi poprzecznej na średnicy
					1 =	przyrostowo, wartość osi poprzecznej na promieniu
				DZIESIĄTKI:	Punkt pośredni w Z	
					0 =	absolutnie
					1 =	przyrostowo
				SETKI:	Punkt końcowy w X	
					0 =	absolutnie, wartość osi poprzecznej na średnicy
					1 =	przyrostowo, wartość osi poprzecznej na promieniu
				TYSIĄCE:	Punkt końcowy w Z	
					0 =	absolutnie
					1 =	przyrostowo
				DZIESIĄTKI TYSIĘCY:	Promień/faza 1	
					0 =	promień
					1 =	faza
				SETKI TYSIĘCY:	Promień/faza 2	
					0 =	promień
					1 =	faza
				MILIONY:	Promień/faza 3	
					0 =	promień
					1 =	faza

20.1.43 CYCLE952 - Toczenie wcinające konturu (poprzeczne)

Składnia

```
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>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie
1	PRG	<_PRG>	STRING[100]	Nazwa programu skrawania warstwowego
2	CON	<_CON>	STRING[100]	Nazwa programu z którego jest odczytywany aktualny kontur półfabrykatu (przy obróbce pozostałego materiału)
3	CONR	<_CONR>	STRING[100]	Nazwa programu, w którym jest zapisywany aktualny kontur półfabrykatu (patrz <_AMODE> DZIESIĄTKI TYSIĘCY)

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
4		<_VARI>	INT	Rodzaj obróbki		
				JEDNOSTKI:	Rodzaj skrawania	
					1 =	Wzdłużnie
					2 =	Poprzecznie
					3 =	Równolegle do konturu
				DZIESIĄTKI:	Obróbka technologiczna (patrz <_GMODE> SETKI)	
					1 =	Obróbka zgrubna
					2 =	Obróbka wykańczająca
					3 =	Zarezerwowano
					4 =	Obróbka zgrubna dwukanałowa
					5 =	Obróbka wykańczająca dwukanałowa
				SETKI:	Kierunek obróbki	
					1 =	Kierunek obróbki X -
					2 =	Kierunek obróbki X +
					3 =	Kierunek obróbki Z -
					4 =	Kierunek obróbki Z +
				TYSIĄCE:	Kierunek dosuwu	
					1 =	Na zewnątrz X -
					2 =	Wewnątrz X +
					3 =	Strona czołowa Z -
					4 =	Strona tylna Z +
				DZIESIĄTKI TYSIĘCY:	Określenie zachowania się naddatków na obróbkę wykańczającą	
					0 =	Naddatki na obr. wykańczającą UX i UZ oddzielnie
					1 =	Naddatek na obr. wykańczającą U równolegle do konturu
				SETKI TYSIĘCY:	Dociąganie	
					0 =	Kompatybilność, dociąganie automatycznie
					1 =	Z dociąganiem po konturze
					2 =	Bez dociągania
					3 =	Dociąganie automatycznie
				MILIONY:	Podcięcia	
					0 =	Miejsce nie jest uwzględniane przy wcinaniu, - reszta i toczeniu wcinającym, - reszta
					1 =	Obróbka podcięć
					2 =	Nie obrabiać podcięć
				DZIESIĄTKI MILIONÓW:	Za/Przed osią toczenia	
					0 =	Obróbka przed osią toczenia
					1 =	Zarezerwowano

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
5	F	<_F>	REAL	Posuw dla obróbki zgrubnej/wykańczającej		
	FZ			Posuw w odciętej dla toczenia wcinającego		
6	FR	<_FR>	REAL	Posuw zagłębiania w podcięcie dla obróbki zgrubnej		
	FX			Posuw w rzędnej dla toczenia wcinającego		
7	RP	<_RP>	REAL	Płaszczyzna wycofania przy obróbce wewnętrznej (abs, zawsze średnica)		
8	D	<_D>	REAL	Dosuw obróbka zgrubna, (patrz <_AMODE> JEDNOSTKI)		
9	DX	<_DX>	REAL	Dosuw X (patrz <_AMODE> JEDNOSTKI)		
10	DZ	<_DZ>	REAL	Dosuw Z (patrz <_AMODE> JEDNOSTKI)		
11	UX	<_UX>	REAL	Naddatek X (patrz <_VARI> DZIESIĄTKI TYSIĘCY)		
12	UZ	<_UZ>	REAL	Naddatek Z (patrz <_VARI> DZIESIĄTKI TYSIĘCY)		
13	U	<_U>	REAL	Naddatek równoległy do konturu (patrz <_VARI> DZIESIĄTKI TYSIĘCY)		
14	U1	<_U1>	REAL	Dodatkowy naddatek przy obróbce wykańczającej (patrz <_AMODE> TYSIĄCE)		
15	BL	<_BL>	INT	Definicja półfabrykatu	1 =	Walec z naddatkiem
					2 =	Naddatek na konturze części gotowej
					3 =	Kontur półfabrykatu jest podany
16	XD	<_XD>	REAL	Definicja półfabrykatu X (patrz <_AMODE> SETKI TYSIĘCY)		
17	ZD	<_ZD>	REAL	Definicja półfabrykatu Z (patrz <_AMODE> JEDNOSTKI MILIONÓW)		
18	XA	<_XA>	REAL	Granica 1 X (abs, zawsze średnica)		
19	ZA	<_ZA>	REAL	Granica 1 Z (abs)		
20	XB	<_XB>	REAL	Granica 2 X (patrz <_AMODE> DZIESIĄTKI MILIONÓW		
21	ZB	<_ZB>	REAL	Granica 2 Z (patrz <_AMODE> SETKI MILIONÓW		
22	XDA	<_XDA>	REAL	Granica 1 wcinania dla 1. pozycji wcinania ma stronie czołowej (abs, zawsze średnica)		
23	XDB	<_XDB>	REAL	Granica 2 wcinania dla 1. pozycji wcinania ma stronie czołowej (abs, zawsze średnica)		
24	N	<_N>	INT	Liczba rowków		
25	DP	<_DP>	REAL	Odstęp pomiędzy rowkami	Rowek wzdłużny: równolegle do osi Z	
					Rowek poprzeczny: równolegle do osi X	
26	DI	<_DI>	REAL	Odstęp dla przerwania posuwu	0 =	Bez przerywania
					> 0 =	Z przerwaniem
27	SC	<_SC>	REAL	Odstęp bezpieczeństwa do obchodzenia przeszkód, przyrostowo		
28	D2	<_DN>	INT	Numer D dla 2. ostrza, gdy nie zaprogramowano ⇒ D+1		

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
29		<_GMODE>	INT	Tryb geometrii (ewaluacja zaprogramowanych wartości geometrycznych)		
				JEDNOSTKI:	Zarezerwowano	
				DZIESIĄTKI:	Zarezerwowano	
				SETKI:	Wybór obróbka / tylko obliczenie punktu startowego	
					0 =	Obróbka normalna (tryb kompatybilności nie jest wymagany)
					1 =	Normalna obróbka
					2 =	Obliczenie pozycji startowej - bez obróbki (tylko dla wywołania z ShopMill/ShopTurn)
				TYSIĄCE:	Ograniczenie	
					0 =	nie
					1 =	tak
				DZIESIĄTKI TYSIĘCY:	Wprowadzenie granicy 1 X	
					0 =	nie
					1 =	tak
				SETKI TYSIĘCY:	Wprowadzenie granicy 2 X	
					0 =	nie
					1 =	tak
				MILIONY:	Wprowadzenie granicy 1 Z	
					0 =	nie
					1 =	tak
				DZIESIĄTKI MILIONÓW:	Wprowadzenie granicy 2 Z	
					0 =	nie
					1 =	tak

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
30		<_DMODE>	INT	Tryb wyświetlania		
				JEDNOSTKI:	Płaszczyzna obróbki G17/G18/G19	
					0 =	Kompatybilność, aktywna pozostaje płaszczyzna działająca przed wywołaniem cyklu
					1 =	G17 (aktywna tylko w cyklu)
					2 =	G18 (aktywna tylko w cyklu)
					3 =	G19 (aktywna tylko w cyklu)
				DZIESIĄTKI:	Rodzaj technologii	
					1 =	Skrawanie konturu
					2 =	Toczenie wcinające konturu
					3 =	Toczenie wcinające
				SETKI:	Obróbka pozostałego materiału	
					0 =	nie
					1 =	tak
				TYSIĄCE:	---	Zarezerwowano
				DZIESIĄTKI TYSIĘCY:	Skalowanie technologiczne w ramach okien cykli (Strona 853)	
					0 =	Wprowadzenie: kompletne
					1 =	Wprowadzenie: uproszczone

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
31		<_AMODE>	INT	Tryb alternatywny		
				JEDNOSTKI:	Wybór dosuwu	
					0 =	Dosuw DX i DZ przy rodzaju skrawania równolegle do konturu
					1 =	dosuw D
				DZIESIĄTKI:	Strategia dosuwu	
					0 =	Zmienna głębokość skrawania (90 ... 100 %)
					1 =	Stała głębokość skrawania
				SETKI:	Podział na przejścia narzędzia	
					0 =	Równomiernie
					1 =	Zorientować na krawędziach
				TYSIĄCE:	Wybór naddatku na konturze U1, obróbka wykańczająca podwójna	
					0 =	nie
					1 =	tak
				DZIESIĄTKI TYSIĘCY:	Wybór aktualnego półfabrykatu	
					0 =	nie
					1 =	tak
				SETKI TYSIĘCY:	Wybór naddatku na konturze XD	
					0 =	Absolutnie, wartość osi poprzecznej na średnicy
					1 =	Przyrostowo, wartość osi poprzecznej na promieniu
				MILIONY:	Wybór naddatku na półfabrykacie ZD	
					0 =	Absolutnie
					1 =	Przyrostowo
				DZIESIĄTKI MILIONÓW:	Wybór granicy 2 XB	
					0 =	Absolutnie, wartość osi poprzecznej na średnicy
					1 =	Przyrostowo, wartość osi poprzecznej na promieniu
				SETKI MILIONÓW:	Wybór granicy 2 ZB	
					0 =	Absolutnie
					1 =	Przyrostowo
				JEDNOSTKI MILIARDÓW:		
					0 =	Kanał wiodący
					1 =	Kanał nadążny
32		<_PK>	INT	Numer kanału partnerskiego, gdy w maszynie są więcej, niż 2 kanały		
33	DCH	<_DCH>	REAL	Przesunięcie kanału		
34	FS	<_FS>	REAL	Posuw przy obróbce wykańczającej w ramach obróbki kompletnej		

20.1.44 CYCLE4071 - Szlifowanie wzdłużne z dosuwem w punkcie nawrotu

Składnia

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_A>	REAL	Głębokość dosuwu na początku
2	<S_B>	REAL	Głębokość dosuwu na końcu
3	<S_W>	REAL	Szerokość szlifowania
4	<S_U>	REAL	Czas wyiskrzania
5	<S_I>	REAL	Posuw dla dosuwu
6	<S_K>	REAL	Posuw dosuwu poprzecznego
7	<S_H>	INT	Liczba powtórzeń
8	<S_A1>	AXIS	Oś dosuwu (opcjonalnie) lub 1. oś geometryczna
9	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie) lub 2. oś geometryczna

Funkcja

Cykl służy do wykonania <S_H>dosunięć. Przy tym głębokość dosuwu na początku i na końcu może być różna. Między dosunięciami następuje ruch styczny.

Przebieg

1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego.
2. Ruch osi dosuwu na głębokość dosuwu na początku P1<S_A> z posuwem dla dosunięcia P5 <S_I>.
3. Wyiskrzenie z czasem wyiskrzenia P4 <S_U>.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P3 <S_W> jako droga ruchu i posuwu dla dosuwu poprzecznego P6 <S_K>
5. Ruch osi dosuwu na głębokość dosuwu na końcu P2<S_B> z posuwem dla dosunięcia P5 <S_I>.
6. Wyiskrzenie z czasem wyiskrzenia P4 <S_U>.
7. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu na punkt początkowy i posuwu dla dosuwu poprzecznego P6 <S_K>.

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.

Przebieg jest powtarzany odpowiednio do zaprogramowanej liczby powtórzeń P7 (<S_H>).

Przykład

Wykonywanie dwóch ruchów wahadłowych z następującymi parametrami cyklu:

- Głębokość dosuwu na początku 0.02mm
- Głębokość dosuwu na końcu 0.01mm
- Skok: 100 mm
- Czas wyiskrzania: 1 s
- Posuw dosuwu poprzecznego: 1 mm/min
- Posuw poprzeczny: 1000 mm/min
- Powtórzenia: 2
- Oś ruchu wahadłowego i oś dosuwu: Standardowe osie geometryczne

Kod programu

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.45 CYCLE4072 -Szlifowanie wzdłużne z dosuwem w punkcie nawrotu i sygnał anulowania

Składnia

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_GAUGE>	STRING	Warunki anulowania dla dosuwu: 1. Numer szybkiego wejścia 2. Wyrażenie logiczne
2	<S_A>	REAL	Głębokość dosuwu na początku
3	<S_B>	REAL	Głębokość dosuwu na końcu
4	<S_W>	REAL	Szerokość szlifowania
5	<S_U>	REAL	Czas wyiskrzania
6	<S_I>	REAL	Posuw dla dosuwu
7	<S_K>	REAL	Posuw dosuwu poprzecznego
8	<S_H>	INT	Liczba powtórzeń
9	<S_A1>	AXIS	Oś dosuwu (opcjonalnie) lub 1. oś geometryczna
10	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie) lub 2. oś geometryczna

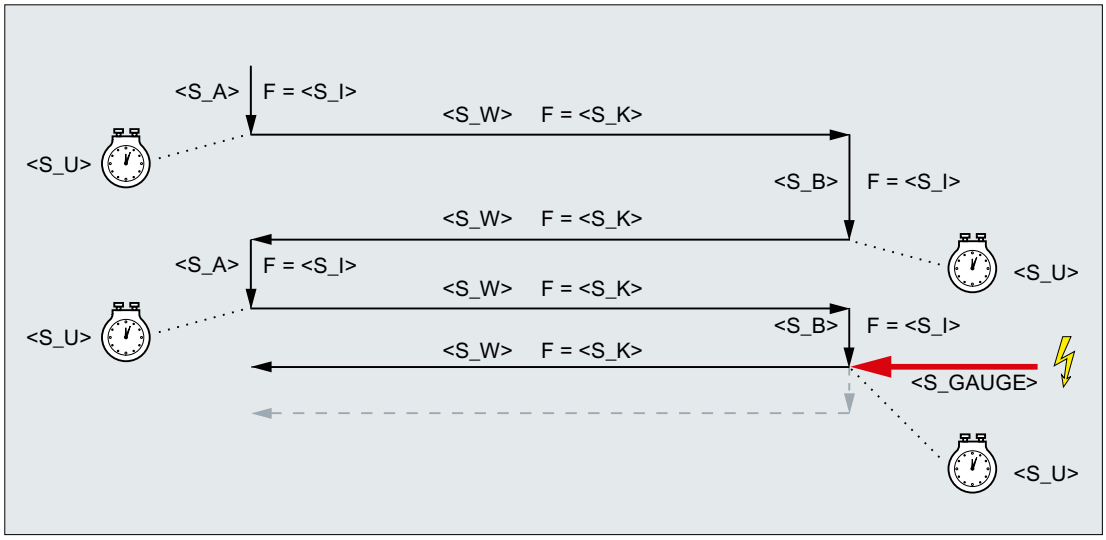
Funkcja

Cykl służy do wykonania <S_H> dosunięć przy uwzględnieniu zewnętrznego sygnału anulowania. Przy tym głębokość dosuwu na początku i na końcu może być różna. Między dosunięciami następuje ruch styczny. Dosuw na głębokość jest anulowany, gdy warunek przerwania jest spełniony. Po anulowaniu dosuwu na głębokość jest zawsze wykonywany kompletny skok.

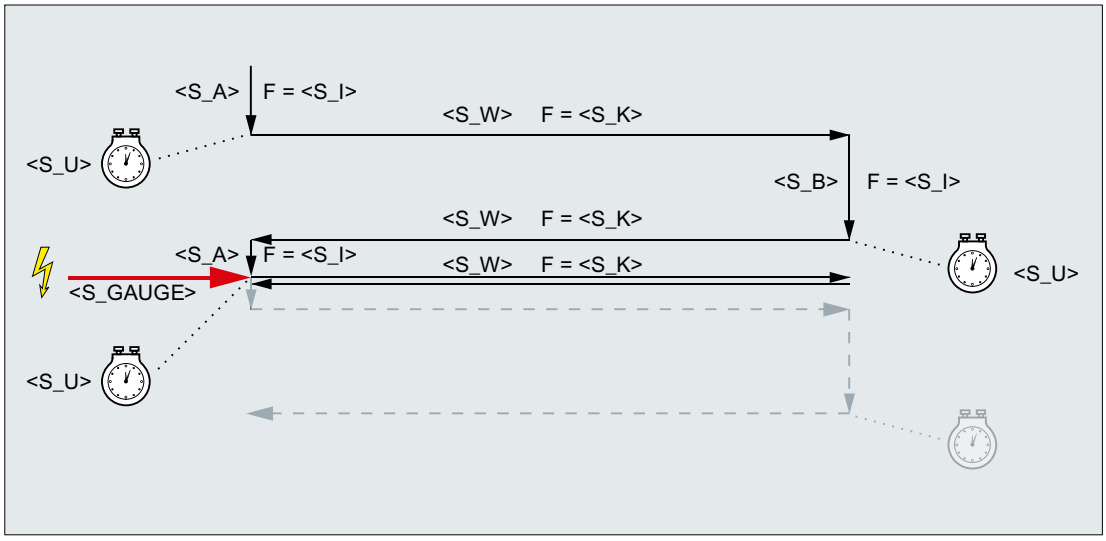
Przebieg

1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego.
2. Ruch osi dosuwu na głębokość dosuwu na początku P2 **<**S_A**>** z posuwem dla dosunięcia P6 <S_I>.
3. Wyiskrzenie z czasem wyiskrzenia P5 **<**S_U**>**.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 **<**S_W**>** jako droga ruchu i posuwu dla dosuwu poprzecznego P7 <S_K>
5. Ruch osi dosuwu na głębokość dosuwu na końcu P3**<**S_B**>** z posuwem dla dosunięcia P6 <S_I>.
6. Wyiskrzenie z czasem wyiskrzenia P5 **<**S_U**>**.
7. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 **<**S_W**>** jako droga ruchu na punkt początkowy i posuwu dla dosuwu poprzecznego P7 <S_K>
8. Bez przerwania: Opisany powyżej przebieg jest tak często powtarzany aż osiągnięta zostanie zaprogramowana liczba powtórzeń P7 (<S_H>).
   Z przerwaniem: obróbka jest przerywana wraz z osiągnięciem następnego punktu startowego.

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.



Rysunek 20-1 Anulowanie dosuwu na końcu



Rysunek 20-2 Anulowanie dosuwu na początku

Zasoby

Jako zasobów cykl używa wykraczającej poza granice bloku akcji synchronicznej i jednej zmiennej akcji synchronicznej. Akcja synchroniczna jest określana dynamicznie z wolnego zakresu pasma akcji synchronicznych (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Jako zmienna akcji synchronicznej jest stosowana SYG_IS[1].

Przykłady

Przykład 1: Oscylacja z dwoma skokami

Parametry cykli:

- Głębokość dosuwu na początku 0.02mm
- Głębokość dosuwu na końcu 0.01mm
- Skok: 100 mm
- Czas wyiskrzania: 1 s
- Posuw dosuwu poprzecznego: 1 mm/min
- Posuw poprzeczny: 1000 mm/min
- Powtórzenia: 2
- Oś ruchu wahadłowego i oś dosuwu: Standardowe osie geometryczne

Sygnał anulowania: szybkie wejście 1 ($A_IN[1])

Kod programu

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

Przykład 2: Oscylacja z dwoma skokami

Parametry cykli:

- Głębokość dosuwu na początku 0.02mm
- Głębokość dosuwu na końcu 0.01mm
- Skok: 100 mm
- Czas wyiskrzania: 1 s
- Posuw dosuwu poprzecznego: 1 mm/min
- Posuw poprzeczny: 1000 mm/min
- Powtórzenia: 2
- Oś ruchu wahadłowego i oś dosuwu: Standardowe osie geometryczne

Sygnał anulowania: Zmienna $A_DBR[20] < 0,01

Kod programu

```
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 - Szlifowanie wzdłużne z dosuwem ciągłym

Składnia

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_A>	REAL	Głębokość dosuwu na początku
2	<S_B>	REAL	Głębokość dosuwu na końcu
3	<S_W>	REAL	Szerokość szlifowania
4	<S_U>	REAL	Czas wyiskrzania
5	<S_K>	REAL	Posuw dosuwu poprzecznego
6	<S_H>	INT	Liczba powtórzeń
7	<S_A1>	AXIS	Oś dosuwu (opcjonalnie) lub 1. oś geometryczna
8	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie) lub 2. oś geometryczna

Funkcja

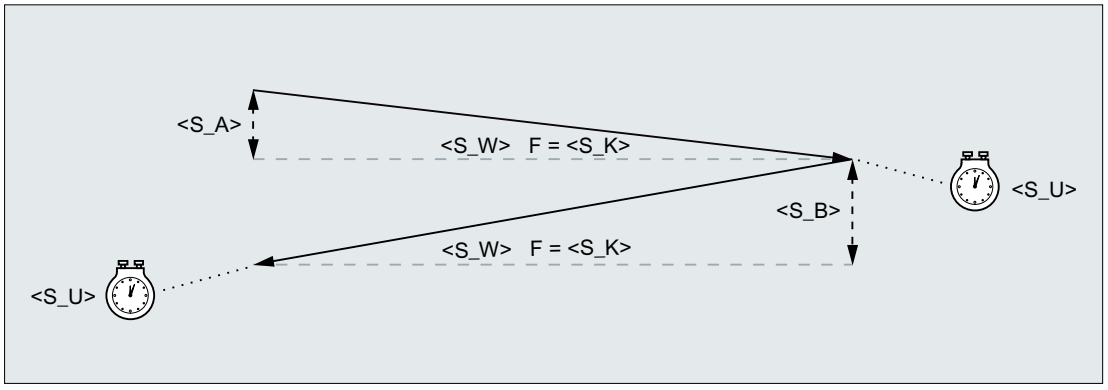
Cykl służy do wykonania dosunięć <S_H>. Przy tym dosunięcie może być różne od początku do końca i od końca do początku.

Przebieg

1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego z głębokością dosuwu 0.
2. Ruch osi ruchu wahadłowego z szerokością szlifowania P3 <S_W> jako droga ruchu i posuwu dla dosuwu poprzecznego P5 <S_K> przy ciągłym zwiększaniu głębokości dosuwu maksymalnie na głębokość dosuwu na początku P1 <S_A>.
3. Wyiskrzenie z czasem wyiskrzenia P4 <S_U>.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P3 <S_W> jako droga ruchu i posuwu dla dosuwu poprzecznego P5 S_K<S_B> przy ciągłym zwiększaniu głębokości dosuwu maksymalnie na głębokość dosuwu na końcu P2 <>.
5. Wyiskrzenie z czasem wyiskrzenia P4 <S_U>.

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.

Przebieg jest powtarzany odpowiednio do zaprogramowanej liczby powtórzeń P7 (<S_H>).



Przykład

Oscylacja z dwoma skokami

Parametry cykli:

- Głębokość dosuwu na początku 0.02mm
- Głębokość dosuwu na końcu 0.01mm
- Skok: 100 mm
- Czas wyiskrzania: 1 s
- Posuw poprzeczny: 1000 mm/min
- Powtórzenia: 2
- Oś ruchu wahadłowego i oś dosuwu: Standardowe osie geometryczne

Kod programu

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.47 CYCLE4074 - Szlifowanie wzdłużne z dosuwem ciągłym i sygnałem przerwania

Składnia

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_GAUGE>	STRING	Warunki anulowania dla dosuwu: 1. Numer szybkiego wejścia 2. Wyrażenie logiczne
2	<S_A>	REAL	Głębokość dosuwu na początku
3	<S_B>	REAL	Głębokość dosuwu na końcu
4	<S_W>	REAL	Szerokość szlifowania
5	<S_U>	REAL	Czas wyiskrzania
6	<S_K>	REAL	Posuw dosuwu poprzecznego
7	<S_H>	INT	Liczba powtórzeń
8	<S_A1>	AXIS	Oś dosuwu (opcjonalnie) lub 1. oś geometryczna
9	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie) lub 2. oś geometryczna

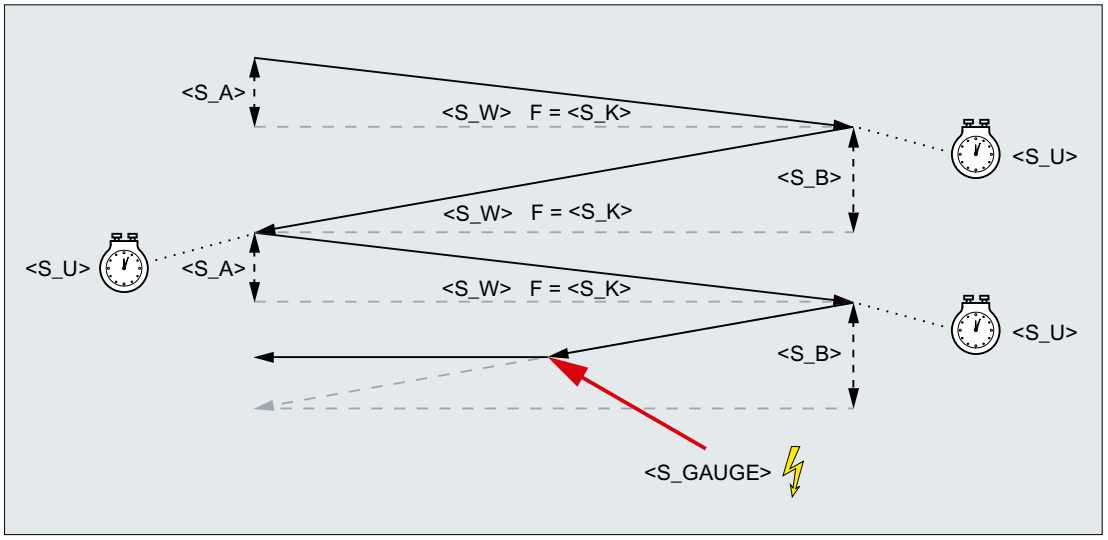
Funkcja

Cykl służy do wykonania <S_H> dosunięć przy uwzględnieniu np. zewnętrznego sygnału anulowania. Przy tym głębokość dosuwu na początku i na końcu może być różna. Dosuw na głębokość jest anulowany, gdy warunek przerwania jest spełniony. Po anulowaniu dosuwu na głębokość jest zawsze wykonywany kompletny skok.

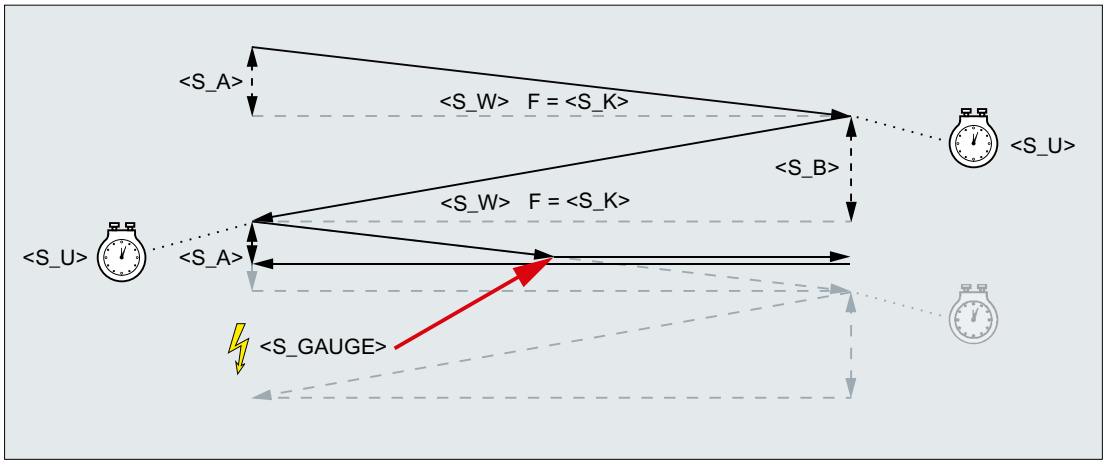
Przebieg

1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego z głębokością dosuwu 0.
2. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_W> jako droga ruchu i posuwu dla dosuwu poprzecznego P6 <S_A> przy ciągłym zwiększaniu głębokości dosuwu maksymalnie na głębokość dosuwu na początku P2 <S_A>.
3. Wyiskrzenie z czasem wyiskrzenia P5 <S_U>.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_W> jako droga ruchu i posuwu dla dosuwu poprzecznego P6 S_K<S_B> przy ciągłym zwiększaniu głębokości dosuwu maksymalnie na głębokość dosuwu na końcu P3 <>.
5. Wyiskrzenie z czasem wyiskrzenia P5 <S_U>.
6. Bez przerwania: Opisany powyżej przebieg jest tak często powtarzany aż osiągnięta zostanie zaprogramowana liczba powtórzeń P7 (<S_H>).
   Z przerwaniem: Dosuw na głębokość jest anulowany. Obróbka jest przerywana wraz z osiągnięciem następnego punktu startowego.

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.



Rysunek 20-3 Anulowanie dosuwu od końca do początku



Rysunek 20-4 Anulowanie dosuwu od początku do końca

Zasoby

Jako zasobów cykl używa wykraczającej poza granice bloku akcji synchronicznej i jednej zmiennej akcji synchronicznej. Akcja synchroniczna jest określana dynamicznie z wolnego zakresu pasma akcji synchronicznych (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Jako zmienna akcji synchronicznej jest stosowana SYG_IS[1].

Przykłady

Przykład 1: Oscylacja z dwoma skokami

Parametry cykli:

- Głębokość dosuwu na początku 0.02mm
- Głębokość dosuwu na końcu 0.01mm

- Skok: 100 mm
- Czas wyiskrzania: 1 s
- Posuw poprzeczny: 1000 mm/min
- Powtórzenia: 2
- Oś ruchu wahadłowego i oś dosuwu: Standardowe osie geometryczne

Sygnał anulowania: szybkie wejście 1 ($A_IN[1])

Kod programu

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

Przykład 2: Oscylacja z dwoma skokami

Parametry cykli:

- Głębokość dosuwu na początku 0.02mm
- Głębokość dosuwu na końcu 0.01mm
- Skok: 100 mm
- Czas wyiskrzania: 1 s
- Posuw poprzeczny: 1000 mm/min
- Powtórzenia: 2
- Oś ruchu wahadłowego i oś dosuwu: Standardowe osie geometryczne

Sygnał anulowania: Zmienna $A_DBR[20] < 0,01

Kod programu

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.48 CYCLE4075 - Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu

Składnia

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_I>	REAL	Głębokość dosuwu na początku
2	<S_J>	REAL	Głębokość dosuwu na końcu

Nr	Parametry	Typ danych	Znaczenie
3	<S_K>	REAL	Całkowita głębokość dosuwu
4	<S_A>	REAL	Szerokość szlifowania
5	<S_R>	REAL	Posuw dla dosuwu
6	<S_F>	REAL	Posuw dosuwu poprzecznego
7	<S_P>	REAL	Czas wyiskrzania
8	<S_A1>	AXIS	Oś dosuwu (opcjonalnie)
9	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie)

Funkcja

Cykl służy do obróbki na całkowitą głębokością dosuwu P3 <S_K> w krokach dosuwu. Głębokość dosuwu na początku P1 <S_I i na końcu P2 S_J może być różna. Między dosunięciami następuje ruch styczny.

Oznaczenia drogi P1 do P4 mogą mieć wartość ujemną lub dodatnią.

Oznaczenie osi dosuwu P8 <S_A1> i/lub osi ruchu wahadłowego P9 <S_A2> są opcjonalne. Jeżeli nie podano jednego lub obydwu parametrów, wówczas cykl stosuje pierwsze dwie osie geometryczne kanału.

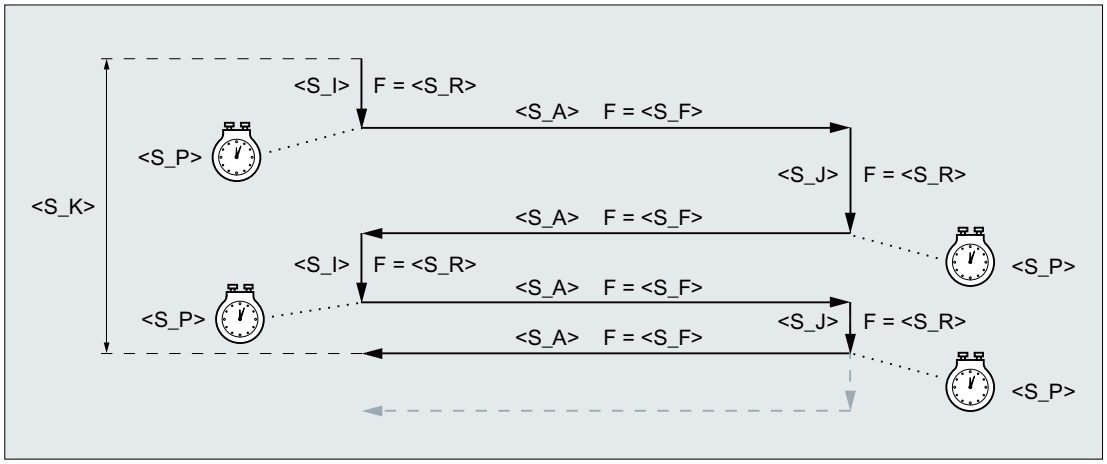
Jeżeli suma z głębokości dosuwu na początku P1 <S_I i na końcu P2 S_J jest równa 0 lub całkowita głębokość dosuwu P3 <S_K> jest równa 0 wykonywany jest tylko skok wyiskrzenia.

Przebieg

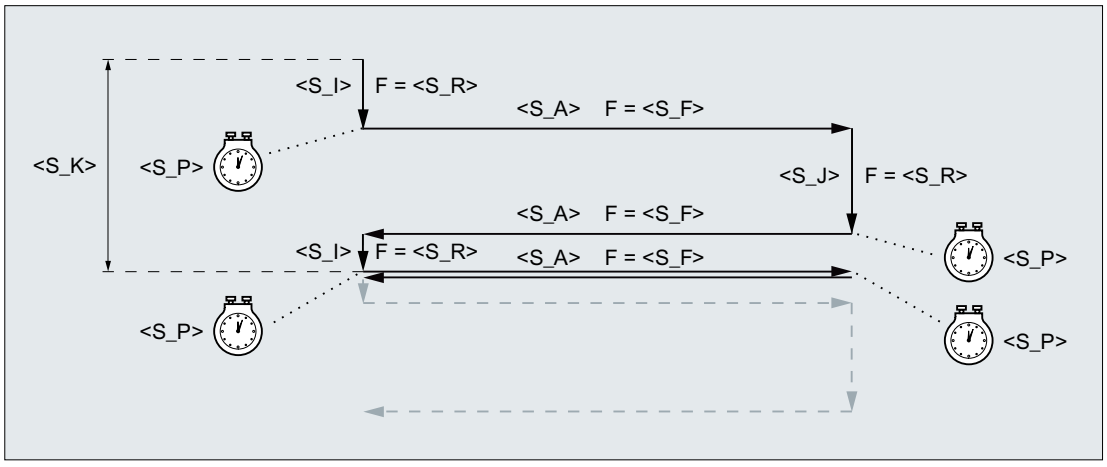
1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego.
2. Ruch osi dosuwu na głębokość dosuwu na początku P1<S_I> z posuwem dla dosunięcia P5 <S_R>.
3. Wyiskrzenie z czasem wyiskrzenia P7 <S_P>.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu i posuwu dla dosuwu poprzecznego P6 <S_F>.
5. Ruch osi dosuwu na głębokość dosuwu na końcu P2<S_J> z posuwem dla dosunięcia P5 <S_R>.
6. Wyiskrzenie z czasem wyiskrzenia P7 <S_P>.
7. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu na punkt początkowy i posuwu dla dosuwu poprzecznego P6 <S_F>.

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.

Przebieg ten jest tak często powtarzany aż osiągnięta zostanie całkowita głębokość dosuwu P3 <S_K>. Ostatni skok jest następnie dzielony nierównomiernie.



Rysunek 20-5 Całkowita głębokość dosuwu osiągnięta przy dosuwie w drugim punkcie nawrotu



Rysunek 20-6 Całkowita głębokość dosuwu osiągnięta przy dosuwie w pierwszym punkcie nawrotu

Przykład

Oscylacja z:

- Głębokością dosuwu 0,02 mm na początku
- Głębokością dosuwu 0,01 mm na końcu
- Całkowita głębokość dosuwu 1 mm
- Skok 100 mm
- Posuw dosuwu 1 mm/min
- Posuw poprzeczny 1000 mm/min
- Czas wyiskrzania 1 sekunda
- Standardowe osie geometryczne

Kod programu

```
N10 T1 D1
```

Kod programu

```
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.49 CYCLE4077 - Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu i przerwania sygnału

Składnia

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_GAUGE>	STRING	Warunek anulowania dla dosuwu: • Numer szybkiego wejścia • Wyrażenie logiczne
2	<S_I>	REAL	Głębokość dosuwu na początku
3	<S_J>	REAL	Głębokość dosuwu na końcu
4	<S_K>	REAL	Całkowita głębokość dosuwu
5	<S_A>	REAL	Szerokość szlifowania
6	<S_R>	REAL	Posuw dla dosuwu
7	<S_F>	REAL	Posuw dosuwu poprzecznego
8	<S_P>	REAL	Czas wyiskrzania
9	<S_A1>	AXIS	Oś dosuwu (opcjonalnie)
10	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie)

Funkcja

Cykl służy do obróbki na całkowitą głębokością dosuwu P4 <S_K> w krokach dosuwu. Głębokość dosuwu na początku P2 <S_I i na końcu P3 S_J może być różna. Między dosunięciami następuje ruch styczny. Dosuw na głębokość jest anulowany, gdy sygnał szybkiego wejścia jest równy 1 lub spełniony jest warunek anulowania. Po anulowaniu jest zawsze wykonywany kompletny skok.

Oznaczenia drogi P2 do P5 mogą mieć wartość ujemną lub dodatnią.

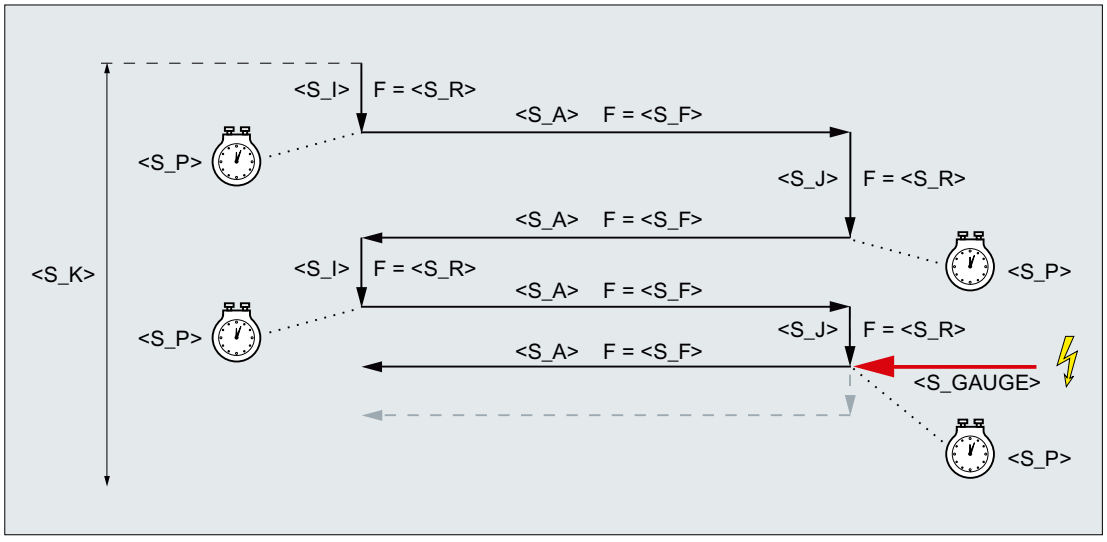
Oznaczenie osi dosuwu P9 <S_A1> i/lub osi ruchu wahadłowego P10 <S_A2> są opcjonalne. Jeżeli nie podano jednego lub obydwu parametrów, wówczas cykl stosuje pierwsze dwie osie geometryczne kanału.

Jeżeli suma z głębokości dosuwu na początku P2 <S_I i na końcu P3 S_J jest równa 0 lub całkowita głębokość dosuwu P4 <S_K> jest równa 0 wykonywany jest tylko skok wyiskrzenia.

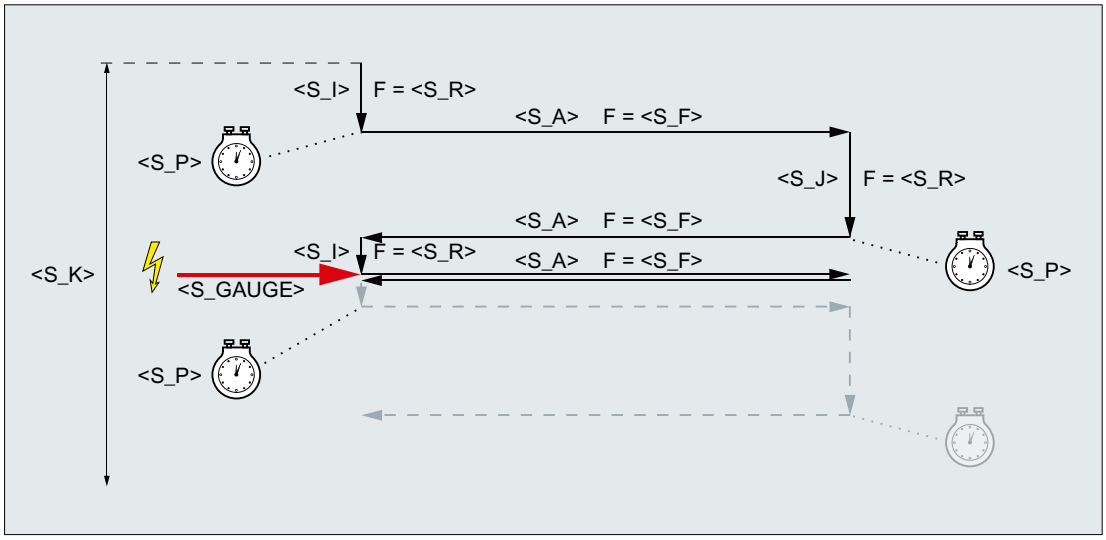
Przebieg

1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego.
2. Ruch osi dosuwu na głębokość dosuwu na początku P2 <S_I> z posuwem dla dosunięcia P6 <S_R>.
3. Wyiskrzenie z czasem wyiskrzenia P8 <S_P>.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P5 <S_A> jako droga ruchu i posuwu dla dosuwu poprzecznego P7 <S_F>
5. Ruch osi dosuwu na głębokość dosuwu na końcu P3 <S_J> z posuwem dla dosunięcia P6 <S_R>.
6. Wyiskrzenie z czasem wyiskrzenia P8 <S_P>.
7. Ruch osi ruchu wahadłowego z szerokością szlifowania P5 <S_A> jako droga ruchu na punkt początkowy i posuwu dla dosuwu poprzecznego P7 <S_F>
8. Bez przerwania: Opisany powyżej przebieg jest tak często powtarzany aż osiągnięta zostanie całkowita głębokość dosuwu P4 (<S_K>). Ostatni skok jest następnie dzielony nierównomiernie.
   Z przerwaniem: obróbka ulega zakończeniu w punkcie startowym.

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.



Rysunek 20-7 Anulowanie dosuwu na końcu



Rysunek 20-8 Anulowanie dosuwu na początku

Zasoby

Jako zasobów cykl używa wykraczającej poza granice bloku akcji synchronicznej i jednej zmiennej akcji synchronicznej. Akcja synchroniczna jest określana dynamicznie z wolnego zakresu pasma akcji synchronicznych (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Jako zmienna akcji synchronicznej jest stosowana SYG_IS[1].

Przykłady

Przykład 1

Oscylacja z:

- Głębokością dosuwu 0,02 mm na początku
- Głębokością dosuwu 0,01 mm na końcu
- Całkowita głębokość dosuwu 1 mm
- Skok 100 mm
- Posuw dosuwu 1 mm/min
- Posuw poprzeczny 1000 mm/min
- Czas wyiskrzania 1 sekunda
- Standardowe osie geometryczne

Sygnał anulowania: szybkie wejście 1 ($A_IN[1])

Kod programu

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

Przykład 2

Oscylacja z:

- Głębokością dosuwu 0,02 mm na początku
- Głębokością dosuwu 0,01 mm na końcu
- Całkowita głębokość dosuwu 1 mm
- Skok 100 mm
- Posuw dosuwu 1 mm/min
- Posuw poprzeczny 1000 mm/min
- Czas wyiskrzania 1 sekunda
- Standardowe osie geometryczne

Sygnał anulowania: Zmienna Dualport-RAM 20 mniejsza niż 0,01 ($A_DBR[20] < 0,01)

Kod programu

```
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 -Szlifowanie płaszczyzny z dosuwem ciągłym

Składnia

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_I>	REAL	Głębokość dosuwu od początku do końca
2	<S_J>	REAL	Głębokość dosuwu od końca do początku
3	<S_K>	REAL	Całkowita głębokość dosuwu
4	<S_A>	REAL	Szerokość szlifowania
5	<S_F>	REAL	Posuw
6	<S_P>	REAL	Czas wyiskrzania
7	<S_A1>	AXIS	Oś dosuwu (opcjonalnie)
8	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie)

Funkcja

Cykl służy do obróbki na całkowitą głębokością dosuwu P3 <S_K> z wykorzystaniem kroków dosuwu. Głębokość dosuwu od początku do końca P1 <S_I i od końca do początku P2 <S_J> może być różna.

Oznaczenia drogi P1 do P4 mogą mieć wartość ujemną lub dodatnią.

Oznaczenie osi dosuwu P8 <S_A1> i/lub osi ruchu wahadłowego P9 <S_A2> są opcjonalne. Jeżeli nie podano jednego lub obydwu parametrów, wówczas cykl stosuje pierwsze dwie osie geometryczne kanału.

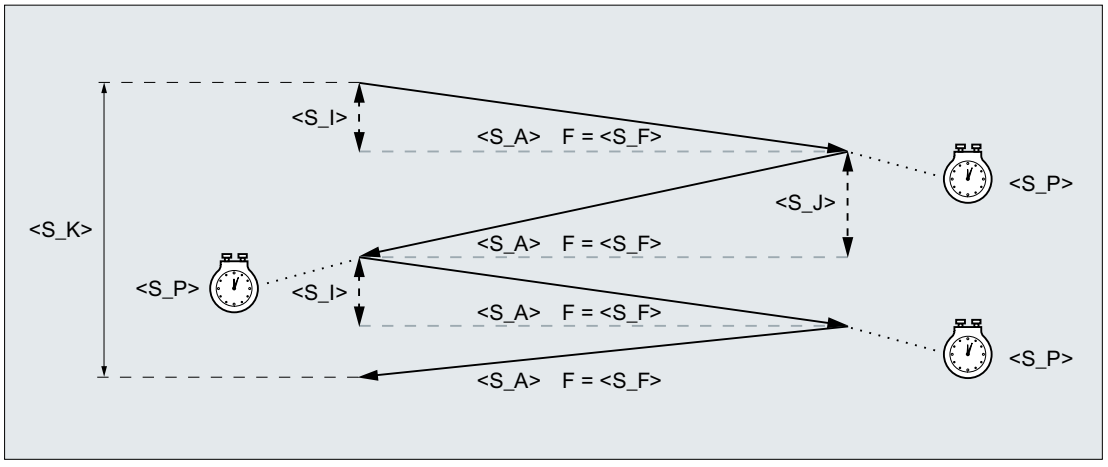
Jeżeli suma z głębokości dosuwu P1 <S_I i P2 S_J jest równa 0 lub całkowita głębokość dosuwu P3 <S_K> jest równa 0 wykonywany jest tylko skok wyiskrzenia.

Przebieg

1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego z głębokością dosuwu 0.
2. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu i posuwu dla dosuwu poprzecznego P5 <S_F> przy ciągłym zwiększaniu głębokości dosuwu maksymalnie na głębokość dosuwu na początku P1 <S_I>.
3. Wyiskrzenie z czasem wyiskrzenia P7 <S_P>.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu na punkt początkowy i posuw P5 <S_F> przy ciągłym zwiększaniu głębokości dosuwu maksymalnie na głębokość dosuwu na końcu P2 <S_J>.
5. Wyiskrzenie z czasem wyiskrzenia P7 <S_P>.
6. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu na punkt początkowy i posuw P5 <S_F>

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.

Przebieg ten jest tak często powtarzany aż osiągnięta zostanie całkowita głębokość dosuwu P3 <S_K>. Ostatni skok jest następnie dzielony nierównomiernie.



Przykład

Oscylacja z:

- Głębokością dosuwu 20 mm na początku
- Głębokością dosuwu 10 mm na początku
- Całkowita głębokość dosuwu 100 mm
- Skok 100 mm

- Posuw 1000 mm/min
- Czas wyiskrzania 1 sekunda
- Standardowe osie geometryczne

Kod programu

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

20.1.51 CYCLE4079 -Szlifowanie płaszczyzny z dosuwem przerywanym

Składnia

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Parametry

Nr	Parametry	Typ danych	Znaczenie
1	<S_I>	REAL	Głębokość dosuwu na początku
2	<S_J>	REAL	Głębokość dosuwu na końcu
3	<S_K>	REAL	Całkowita głębokość dosuwu
4	<S_A>	REAL	Szerokość szlifowania
5	<S_R>	REAL	Posuw dla dosuwu
6	<S_F>	REAL	Posuw dosuwu poprzecznego
7	<S_P>	REAL	Czas wyiskrzania
8	<S_A1>	AXIS	Oś dosuwu (opcjonalnie)
9	<S_A2>	AXIS	Oś ruchu wahadłowego (opcjonalnie)

Funkcja

Cykl służy do obróbki na całkowitą głębokością dosuwu P3 <S_K> w krokach dosuwu. Głębokość dosuwu na początku P1 <S_I i na końcu P2 S_J może być różna. Między dosunięciami następuje ruch styczny.

Oznaczenia drogi P1 do P4 mogą mieć wartość ujemną lub dodatnią.

Oznaczenie osi dosuwu P8 <S_A1> i/lub osi ruchu wahadłowego P9 <S_A2> są opcjonalne. Jeżeli nie podano jednego lub obydwu parametrów, wówczas cykl stosuje pierwsze dwie osie geometryczne kanału.

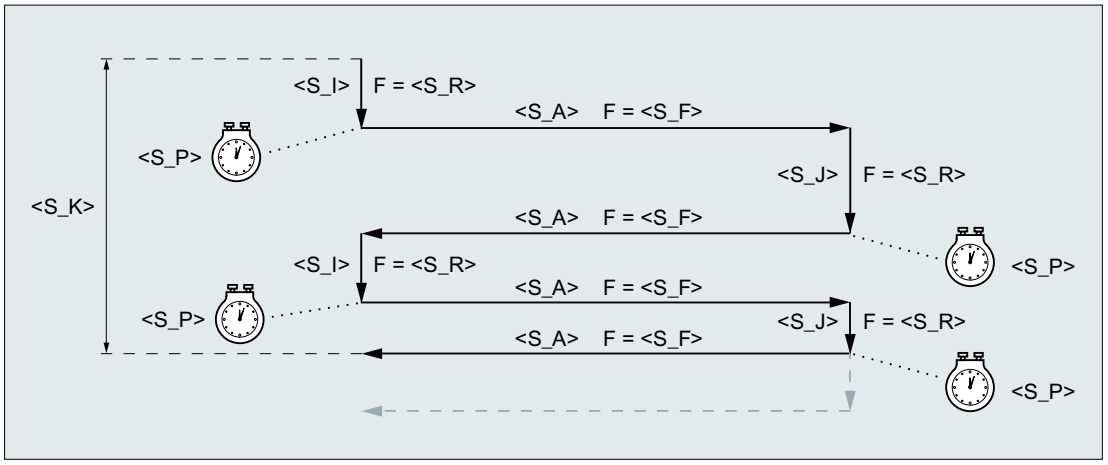
Jeżeli suma z głębokości dosuwu na początku P1 <S_I i na końcu P2 S_J jest równa 0 lub całkowita głębokość dosuwu P3 <S_K> jest równa 0 wykonywany jest tylko skok wyiskrzenia.

Przebieg

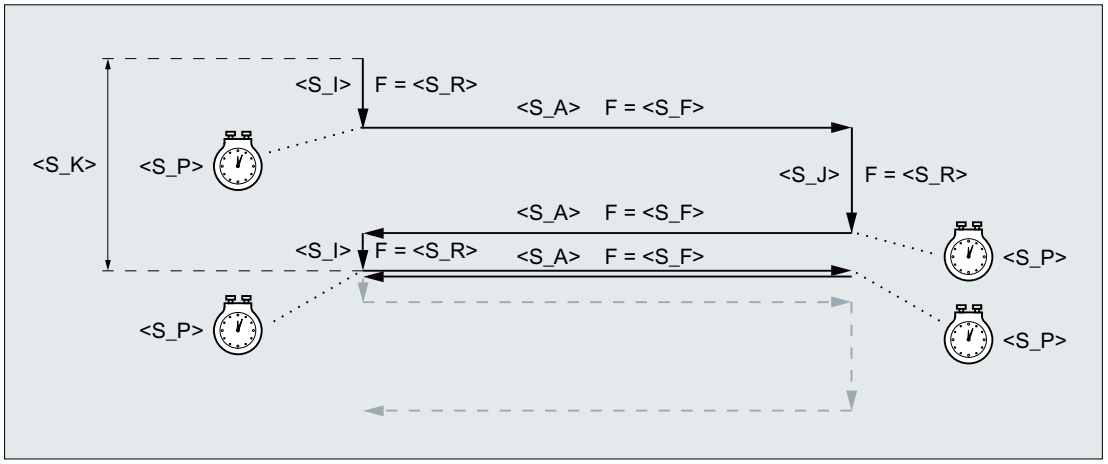
1. Uruchomienie cyklu w aktualnej pozycji osi ruchu wahadłowego.
2. Ruch osi dosuwu na głębokość dosuwu na początku P1<S_I> z posuwem dla dosunięcia P5 <S_R>.
3. Wyiskrzenie z czasem wyiskrzenia P7 <S_P>.
4. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu i posuw dla dosuwu poprzecznego P6 <S_F>.
5. Ruch osi dosuwu na głębokość dosuwu na końcu P2<S_J> z posuwem dla dosunięcia P5 <S_R>.
6. Wyiskrzenie z czasem wyiskrzenia P7 <S_P>.
7. Ruch osi ruchu wahadłowego z szerokością szlifowania P4 <S_A> jako droga ruchu na punkt początkowy i posuw dla dosuwu poprzecznego P6 <S_F>.

Przebieg nie jest przerywany przez wykonywanie pojedynczymi blokami.

Przebieg ten jest tak często powtarzany aż osiągnięta zostanie całkowita głębokość dosuwu P3 <S_K>. Ostatni skok jest następnie dzielony nierównomiernie.



Rysunek 20-9 Całkowita głębokość dosuwu osiągnięta przy dosuwie w drugim punkcie nawrotu



Rysunek 20-10 Całkowita głębokość dosuwu osiągnięta przy dosuwie w pierwszym punkcie nawrotu

Przykład

Oscylacja z:

- Głębokością dosuwu 0,02 mm na początku
- Głębokością dosuwu 0,01 mm na końcu
- Całkowita głębokość dosuwu 1 mm
- Skok 100 mm
- Posuw dosuwu 1 mm/min
- Posuw poprzeczny 1000 mm/min
- Czas wyiskrzania 1 sekunda
- Standardowe osie geometryczne

Kod programu

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.52 GROUP_BEGIN - Początek bloku programu

Składnia

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie		
1		<_LEVEL>	INT	Płaszczyzna		
				0 =	Poziom główny	
				1 =	1. Poziom niżej	
2		<_NAME>	STRING[128]	Nazwa bloku		
3		<_SP>	INT	Wrzeciono		
				0 =	Brak wrzeciona	
				1 =	Wrzeciono główne	
				2 =	Wrzeciono przechwytujące	
4		<_MODE>	INT	Tryb		
				Bit 0	= 1	GROUP_ADDEND występuje
				Bit 1	= 1	ShopTurn: automatyczne wycofanie (ruch na punkt wymiany narzędzia)
				Bit 12	Zarezerwowano	
				Bit 13	Zarezerwowano	
5		<S_ICON>	STRING[32]	Nazwa ikony (tylko dla interfejsu obsługi)		

20.1.53 GROUP_END - Koniec bloku programu

Składnia

```
GROUP_END(<_LEVEL>, <_SP>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie	
1		<_LEVEL>	INT	Płaszczyzna	
				0 =	Poziom główny
				1 =	1. Poziom niżej
2		<_SP>	INT	Wrzeciono	
				0 =	Brak wrzeciona
				1 =	Wrzeciono główne
				2 =	Wrzeciono przechwytujące

20.1.54 GROUP_ADDEND - Koniec dodatkowego dosuwu

Składnia

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

Parametry

Nr	Parametry okno	Parametry wewnętrzne	Typ danych	Znaczenie	
1		<_LEVEL>	INT	Płaszczyzna	
				0 =	Poziom główny
				1 =	1. Poziom niżej
2		<_SP>	INT	Wrzeciono	
				0 =	Brak wrzeciona
				1 =	Wrzeciono główne
				2 =	Wrzeciono przechwytujące

20.1.55 Warunki brzegowe

20.1.55.1 Skalowanie technologiczne w ramach okien cykli

Gdy aktywne jest skalowanie technologiczne, można przy różnych oknach cykli wybrać wprowadzanie uproszczone, przy którym są wyświetlane tylko najistotniejsze parametry cykli.

Przy następujących oknach cykli można np. wybrać wprowadzanie uproszczone:

Technologia	Okno cykli
Wiercenie	Wiercenie otworów głębokich
	Gwintowanie otworu
Frezowanie	Kieszeń prostokątna
	Frezowanie konturu: Kieszeń
Toczenie	Toczenie gwintu: Wzdłużne
	Toczenie konturu: Skrawanie warstwowe
	Toczenie konturu: Wcinanie
	Toczenie konturu: Toczenie wcinające

W przypadku odnośnych okien cykli jest na interfejsie graficznym dostępna możliwość wyboru "Wprowadzanie: **uproszczone**" i "wprowadzenie: **kompletne**".

Nie wyświetlane parametry cykli

Parametry cykli nie wyświetlane przy wprowadzaniu uproszczonym są wyposażane w stałe, technologicznie sensowne, ale niezmienialne wartości. Albo do parametrów cykli są przyporządkowywane wartości parametryzowane poprzez specyficzne dla kanału dane ustawcze cykli. Patrz niżej punkt "Uruchomienie" > "Kanałowe dane ustawcze dla cykli"

Przełączenie kompletne" > "Wprowadzenie: uproszczone"

Gdy okno cyklu zostanie wypełnione przy ustawieniu "Wprowadzanie kompletne” i przełączone na "Wprowadzanie uproszczone”, to przy generowaniu wywołania cyklu dla wówczas już niewyświetlanych parametrów, są stosowane wartości domyślne albo ustawcze.

Uruchomienie

Specyficzne dla kanału dane maszynowe konfiguracji

Przy pomocy danej maszynowej jest uaktywniana możliwość skalowania technologicznego w ramach okien cykli:

MD52210 $MCS_FUNCTION_MASK_DISP, Bit 9 = 1 (wyświetlenie wyboru "Wprowadzanie uproszczone")

Specyficzne dla kanału dane ustawcze dla cykli

Gdy jest aktywne wprowadzanie uproszczone w ramach okien cykli, wartości określonych parametrów cykli można zadać poprzez następujące dane ustawcze:

Numer	Identyfikator	Znaczenie
SD55300	$SCS_EASY_SAFETY_CLEARANCE	Odstęp bezpieczeństwa
SD55301	$SCS_EASY_DWELL_TIME	Czas oczekiwania
SD55305	$SCS_EASY_DRILL_DEEP_FD1	Wiercenie otworów głębokich: Wartość procentowa: 1. Posuw
SD55306	$SCS_EASY_DRILL_DEEP_DF	Wiercenie otworów głębokich: Wartość procentowa: Dosuw
SD55307	$SCS_EASY_DRILL_DEEP_V1	Wiercenie otworów głębokich: minimalny dosuw na głębokość

Numer	Identyfikator	Znaczenie
SD55308	$SCS_EASY_DRILL_DEEP_V2	Wiercenie otworów głębokich: Wartość wycofania
SD55309	$SCS_EASY_THREAD_RETURN_DIST	Toczenie gwintu: Pozycja powrotu

20.2 Cykle pomiarowe

Cykle pomiarowe są specjalnymi, udostępnionymi przez firmę Siemens, podprogramami do rozwiązywania określonych zadań pomiarowych. Jak to jest powszechne w cyklach, również cykle pomiarowe można dopasowywać poprzez parametry do konkretnego problemu.

Cykle pomiarowe są dostępne dla pomiarów w następujących obszarach i technologiach:

- Pomiary narzędzi toczenie / frezowanie
- Pomiary obrabianego przedmiotu toczenie / frezowanie

Literatura

Szczegółowy opis cykli pomiarowych znajduje się w:

Podręcznik programowania Cykle pomiarowe

Tablice

21

21.1 Instrukcje

Uwaga

Cykle

Lista poleceń zawiera wszystkie cykle, które mogą wystąpić w programie NC (kod G), tzn. mogą zostać zaprogramowane za pomocą edytora programu lub poprzez maski oraz w przypadku szlifowania muszą być programowane bez wsparcia programowania. Nie uwzględniono cykli, które z powodów kompatybilności są jeszcze dostępne w sterowaniu, ale nie mogą być programowane za pomocą edytora programów SINUMERIK Operate (cykle kompatybilności).

Polecenia A ... C

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
:	O	Numer bloku głównego NC, zakończenie etykiety skoku, operator powiązania		+		PGAsl
*	O	Operator mnożenia		+		PGAsl
+	O	Operator dodawania		+		PGAsl
-	O	Operator odejmowania		+		PGAsl
<	O	Operator porównania, mniejsze		+		PGAsl
<<	O	Operator powiązania dla łańcuchów znaków		+		PGAsl
<=	O	Operator porównania, mniejsze lub równe		+		PGAsl
=	O	Operator przypisania		+		PGAsl
>=	O	Operator porównania, większe lub równe		+		PGAsl
/	O	Operator dzielenia		+		PGAsl
/0 … … /7		Blok jest ukrywany(1. poziom ukrywania)° ... ... Blok jest ukrywany(8. poziom ukrywania)		+		PGsl
A	A	Nazwa osi	m/s	+		PGAsl
A2	A	Orientacja narzędzia: kąt RPY lub Eulera	s	+		PGAsl
A3	A	Orientacja narzędzia: 1. Składowa wektora kierunkowego	s	+		PGAsl
A4	A	Orientacja narzędzia: 1. Składowa wektora normalnego powierzchni na początku bloku	s	+		PGAsl
A5	A	Orientacja narzędzia: 1. Składowa wektora normalnego powierzchni na końcu bloku	s	+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
A6	A	Orientacja narzędzia: 1. Składowa wektora kierunkowego dla osi obrotu stożka	s	+		PGAsl
A7	A	Orientacja narzędzia: 1. Składowa wektora dla orientacji pośredniej na powierzchni bocznej stożka	s	+		PGAsl
ABS	F	Wartość absolutna (bezwzględna)		+	+	PGAsl
AC	K	Absolutne podanie wymiarów współrzędnych/ pozycji	s	+		PGsl
ACC	K	Wpływ na aktualne przyspieszenie osiowe	m	+	+	PGsl
ACCLIMA	K	Wpływ na aktualne maksymalne przyspieszenie osiowe	m	+	+	PGAsl
ACN	K	Absolutne podanie pozycji dla osi obrotowych, z dosunięciem do pozycji w kierunku ujemnym	s	+		PGsl
ACOS	F	Arcus-Cosinus (funkcja trygonometryczna)		+	+	PGAsl
ACP	K	Absolutne podanie pozycji dla osi obrotowych, z dosunięciem do pozycji w kierunku dodatnim	s	+		PGsl
ACTBLOCNO	P	Wyświetlenie aktualnego numeru bloku alarmowego, również gdy jest aktywne „Blokowanie wyświetlanie aktualnego bloku” (DISPLOF)!		+		PGAsl
ADDFRAME	F	Obliczenie i ew. uaktywnienie zmierzonego Frame		+	-	PGAsl, FB1sl (K2)
ADIS	A	Droga wygładzania dla funkcji ruchu po torze G1, G2, G3, ...	m	+		PGsl
ADISPOS	A	Droga wygładzania dla posuwu szybkiego G0	m	+		PGsl
ADISPOSA	P	Wielkość okna tolerancji dla IPOBRKA	m	+	+	PGAsl
ALF	A	Kąt szybkiego wycofania	m	+		PGAsl
AMIRROR	G	Programowane odbicie lustrzane	s	+		PGsl
AND	K	Logiczne AND		+		PGAsl
ANG	A	Kąt przebiegu konturu	s	+		PGsl
AP	A	Współrzędna kątowa	m/s	+		PGsl
APR	K	Ochrona dostępu do odczytu / wyświetlania		+		PGAsl
APRB	K	Prawo dostępu do odczytu, BTSS		+		PGAsl
APRP	K	Prawo dostępu do odczytu, program obróbki		+		PGAsl
APW	K	Ochrona dostępu do zapisu		+		PGAsl
APWB	K	Prawo dostępu do zapisu, BTSS		+		PGAsl
APWP	K	Prawo dostępu do zapisu, program obróbki		+		PGAsl
APX	K	Definicja ochrony dostępu do wykonania podanego elementu językowego		+		PGAsl
AR	A	Kąt rozwarcia	m/s	+		PGsl
AROT	G	Programowany obrót (addytywny)	s	+		PGsl
AROTS	G	Programowane obroty Frame z kątami przestrzennymi	s	+		PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
AS	K	Definicja makra		+		PGAsl
ASCALE	G	Programowane skalowanie (addytywne)	s	+		PGsl
ASIN	F	Funkcja obliczeniowa, arcus sinus		+	+	PGAsl
ASPLINE	G	Akima-Spline	m	+		PGAsl
ATAN2	F	Arcus tangens2		+	+	PGAsl
ATOL	A	Specyficzna dla osi tolerancja dla funkcji kompresora, wygładzania orientacji i rodzajów wygładzania	m	+		PGAsl
ATRANS	G	Addytywne programowane przesunięcie punktu zerowego	s	+		PGsl
AUXFUDEL	P	Kasowanie specyficznej dla kanału funkcji pomocniczej z globalnej listy funkcji pomocniczych		+	-	FB1sl (H2)
AUXFUDELG	P	Kasowanie wszystkich funkcji pomocniczych dla specyficznej dla kanału danej grupy funkcji z globalnej listy funkcji pomocniczych		+	-	FB1sl (H2)
AUXFUMSEQ	P	Określenie kolejności wyprowadzania funkcji pomocniczych M		+	-	FB1sl (H2)
AUXFUSYNC	P	Wygenerowanie jako łańcucha znaków kompletnego bloku programu obróbki dla kanałowego SERUPRO-End-ASUP, z globalnej listy funkcji pomocniczych		+	-	FB1sl (H2)
AX	K	Zmienny identyfikator osi	m/s	+		PGAsl
AXCTSWE	P	Obrót zasobnika osi		+	-	PGAsl
AXCTSWEC	P	Cofnięcie zezwolenia dla obrotu zasobnika osi		+	+	PGAsl
AXCTSWED	P	Obrót zasobnika osi (wariant polecenia do uruchomienia!)		+	-	PGAsl
AXIS	K	Identyfikator osi, adres osi		+		PGAsl
AXNAME	F	Konwertuje wejściowy łańcuch znaków na identyfikator osi		+	-	PGAsl
AXSTRING	F	Konwertuje łańcuch znaków na numer wrzeciona		+	-	PGAsl
AXTOCHAN	P	Żądanie przejęcia osi dla określonego kanału Możliwe z programu NC i z akcji synchronicznej.		+	+	PGAsl
AXTOSPI	F	Konwertuje identyfikator osi na indeks wrzeciona		+	-	PGAsl
B	A	Nazwa osi	m/s	+		PGAsl
B2	A	Orientacja narzędzia: kąt RPY lub Eulera	s	+		PGAsl
B3	A	Orientacja narzędzia: Składowa wektora kierunkowego	s	+		PGAsl
B4	A	Orientacja narzędzia: 2. Składowa wektora normalnego powierzchni na początku bloku	s	+		PGAsl
B5	A	Orientacja narzędzia: 2. Składowa wektora normalnego powierzchni na końcu bloku	s	+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
B6	A	Orientacja narzędzia: 2. Składowa wektora kierunkowego dla osi obrotu stożka	s	+		PGAsl
B7	A	Orientacja narzędzia: 2. Składowa wektora dla orientacji pośredniej na powierzchni bocznej stożka	s	+		PGAsl
B_AND	O	Bitowe AND		+		PGAsl
B_OR	O	Bitowe OR		+		PGAsl
B_NOT	O	Negacja bitowa		+		PGAsl
B_XOR	O	Wyłącznie bitowe OR		+		PGAsl
BAUTO	G	Definicja pierwszego segmentu spline przez następne 3 punkty	m	+		PGAsl
BLOCK	K	Razem ze słowem kluczowym TO definiuje się wykonywaną część podprogramu w pośrednim wywołaniu podprogramu		+		PGAsl
BLSYNC	K	Wykonywanie procedury przerwania powinno się rozpocząć dopiero wraz z następną zmianą bloku		+		PGAsl
BNAT [6]	G	Przejście naturalne do pierwszego bloku spline	m	+		PGAsl
BOOL	K	Typ danych: wartości logiczne TRUE/FALSE lub 1/0		+		PGAsl
BOUND	F	Sprawdza, czy dana wartość leży w zdefiniowanym zakresie. Równość zwraca sprawdzaną wartość.		+	+	PGAsl
BRISK [6]	G	Skok przyspieszenia po torze	m	+		PGAsl
BRISKA	P	Włącza skok przyspieszenia po torze dla zaprogramowanych osi		+	-	PGAsl
BSPLINE	G	B-Spline	m	+		PGAsl
BTAN	G	Przejście styczne do pierwszego bloku spline	m	+		PGAsl
C	A	Nazwa osi	m/s	+		PGAsl
C2	A	Orientacja narzędzia: kąt RPY lub Eulera	s	+		PGAsl
C3	A	Orientacja narzędzia: 3. Składowa wektora kierunkowego	s	+		PGAsl
C4	A	Orientacja narzędzia: 3. Składowa wektora normalnego powierzchni na początku bloku	s	+		PGAsl
C5	A	Orientacja narzędzia: 3. Składowa wektora normalnego powierzchni na końcu bloku	s	+		PGAsl
C6	A	Orientacja narzędzia: 3. Składowa wektora kierunkowego dla osi obrotu stożka	s	+		PGAsl
C7	A	Orientacja narzędzia: 3. Składowa wektora dla orientacji pośredniej na powierzchni bocznej stożka	s	+		PGAsl
CAC	K	Absolutne dosunięcie do pozycji		+		PGAsl
CACN	K	Ruch do wartości zapisanej w tablicy jest wykonywany absolutnie w kierunku ujemnym		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CACP	K	Ruch do wartości zapisanej w tablicy jest wykonywany absolutnie w kierunku dodatnim		+		PGAsl
CALCDAT	F	Obliczenie promienia i punktu środkowego okręgu z 3 lub 4 punktów		+	-	PGAsl
CALCPOSI	F	Sprawdzenie naruszenia obszaru ochrony, ograniczenia obszaru pracy i ograniczenia programowego		+	-	PGAsl
CALL	K	Pośrednie wywołanie podprogramu		+		PGAsl
CALLPATH	P	Programowana ścieżka szukania przy wywołaniach podprogramu		+	-	PGAsl
CANCEL	P	Anulowanie modalnej akcji synchronicznej		+	-	FBSYsl
CASE	K	Warunkowe rozgałęzienie programu		+		PGAsl
CDC	K	Bezpośrednie dosunięcie do pozycji		+		PGAsl
CDOF [6]	G	Wyłączanie nadzoru nad kolizją	m	+		PGsl
CDOF2	G	Wyłączanie nadzoru nad kolizją przy frezowaniu obwodowym 3D	m	+		PGsl
CDON	G	Włączenie nadzoru nad kolizją	m	+		PGsl
CFC [6]	G	Stały posuw po konturze	m	+		PGsl
CFIN	G	Stały posuw tylko przy zakrzywieniu wewnętrznym, nie przy zakrzywieniu zewnętrznym	m	+		PGsl
CFINE	F	Przyporządkowanie przesunięcia dokładnego do zmiennej FRAME		+	-	PGAsl
CFTCP	G	Posuw stały w punkcie odniesienia ostrza narzędzia, tor punktu środkowego	m	+		PGsl
CHAN	K	Specyfikacja zakresu obowiązywania danych		+		PGAsl
CHANDATA	P	Ustawienie numeru kanału dla dostępu do danych kanału		+	-	PGAsl
CHAR	K	Typ danych: Znak ASCII		+		PGAsl
CHF	A	Faza; Wartość = długość fazy	s	+		PGsl
CHKDM	F	Kontrola jednoznaczności w magazynie		+	-	FBWsl
CHKDNO	F	Badanie jednoznaczności numerów D		+	-	PGAsl
CHR	A	Faza; Wartość = długość fazy w kierunku ruchu		+		PGsl
CIC	K	Przyrostowe dosunięcie do pozycji		+		PGAsl
CIP	G	Interpolacja kołowa przez punkt pośredni	m	+		PGsl
CLEARM	P	Cofnięcie jednego/wielu znaczników do koordynacji kanałów		+	+	PGAsl
CLRINT	P	Cofnięcie wyboru przerwania		+	-	PGAsl
CMIRROR	F	Lustrzane odbicie w jednej osi współrzędnych		+	-	PGAsl
COARSEA	K	Koniec ruchu po osiągnięciu "zatrzymania dokładnego zgrubnie"	m	+		PGAsl
COLLPAIR	F	Kontrola pod kątem przynależności do pary kolizyjnej		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
COMPCAD	G	Włączenie funkcji kompresora bloków COMPCAD	m	+		PGAsl
COMPCURV	G	Włączenie funkcji kompresora bloków COMPCURV	m	+		PGAsl
COMPLETE		Polecenia sterownicze do zapisywania i wczytania danych		+		PGAsl
COMPOF [6]	G	Wyłączenie kompresji bloków NC	m	+		PGAsl
COMPON	G	Włączenie funkcji kompresora bloków COMPON	m	+		PGAsl
COMPSURF	G	Włączenie funkcji kompresora bloków COMPSURF	m	+		PGAsl
CONTDCON	P	Włączenie dekodowania konturu w formie tablicy		+	-	PGAsl
CONTPRON	P	Włączenie przygotowania konturu		+	-	PGAsl
CORROF	P	Odwołanie wszystkich aktywnych ruchów nałożonych.		+	-	PGsl
CORRTRAFO	F	Modyfikacja wektorów offsetu lub wektorów kierunkowych osi orientacji w modelu kinematycznym maszyny		+	-	PGAsl
COS	F	Cosinus (funkcja trygonometryczna)		+	+	PGAsl
COUPDEF	P	Definicja zespołu ELG / zespołu wrzeciona synchronicznego		+	-	PGAsl
COUPDEL	P	Usuwanie zespołu ELG		+	-	PGAsl
COUPOF	P	Wyłączanie zespołu ELG / pary wrzecion synchronicznych		+	-	PGAsl
COUPOFS	P	Wyłączenie zespołu ELG / pary wrzecion synchronicznych z zatrzymaniem wrzeciona nadążnego		+	-	PGAsl
COUPON	P	Włączanie zespołu ELG / pary wrzecion synchronicznych		+	-	PGAsl
COUPONC	P	Włączenie zespołu ELG / przejęcie pary wrzecion synchronicznych ze stanem poprzedzającym zaprogramowanie		+	-	PGAsl
COUPRES	P	Cofnięcie zespołu ELG		+	-	PGAsl
CP [6]	G	Ruch po torze	m	+		PGAsl
CPBC	K	Sprzężenie ogólne: kryterium zmiany bloku		+	+	FB3sl (M3)
CPDEF	K	Sprzężenie ogólne: dodaje tryb sprzężenia		+	+	FB3sl (M3)
CPDEL	K	Sprzężenie ogólne: usuwa tryb sprzężenia		+	+	FB3sl (M3)
CPFMOF	K	Sprzężenie ogólne: zachowanie się osi nadążnej przy kompletnym wyłączeniu		+	+	FB3sl (M3)
CPFMON	K	Sprzężenie ogólne: zachowanie się osi nadążnej przy włączeniu		+	+	FB3sl (M3)
CPFMSON	K	Sprzężenie ogólne: tryb synchronizacji		+	+	FB3sl (M3)
CPFPOS	K	Sprzężenie ogólne: pozycja synchronizacji osi nadążnej		+	+	FB3sl (M3)

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CPFRS	K	Sprzężenie ogólne: układ odniesienia współrzędnych		+	+	FB3sl (M3)
CPLA	K	Sprzężenie ogólne: definicja osi wiodącej		+	-	FB3sl (M3)
CPLCTID	K	Sprzężenie ogólne: numer tablicy krzywych		+	+	FB3sl (M3)
CPLDEF	K	Sprzężenie ogólne: definicja osi wiodącej i utworzenie trybu sprzężenia		+	+	FB3sl (M3)
CPLDEL	K	Sprzężenie ogólne: usuwa oś wiodącą trybu sprzężenia		+	+	FB3sl (M3)
CPLDEN	K	Sprzężenie ogólne: mianownik współczynnika sprzężenia		+	+	FB3sl (M3)
CPLINSC	K	Sprzężenie ogólne: współczynnik skalowania dla wartości wejściowej osi wiodącej		+	+	FB3sl (M3)
CPLINTR	K	Sprzężenie ogólne: wartość przesunięcia dla wartości nadążnej sprzężenia		+	+	FB3sl (M3)
CPLNUM	K	Sprzężenie ogólne: licznik współczynnika sprzężenia		+	+	FB3sl (M3)
CPLOF	K	Sprzężenie ogólne: wyłącza oś wiodącą trybu sprzężenia		+	+	FB3sl (M3)
CPLON	K	Sprzężenie ogólne: włącza oś wiodącą trybu sprzężenia		+	+	FB3sl (M3)
CPLOUTSC	K	Sprzężenie ogólne: współczynnik skalowania wartości wyjściowej sprzężenia		+	+	FB3sl (M3)
CPLOUTTR	K	Sprzężenie ogólne: wartość przesunięcia wstępnego wartości sprzężenia		+	+	FB3sl (M3)
CPLPOS	K	Sprzężenie ogólne: pozycja synchronizacji osi wiodącej		+	+	FB3sl (M3)
CPLSETVAL	K	Sprzężenie ogólne: odniesienie sprzężenia		+	+	FB3sl (M3)
CPMALARM	K	Sprzężenie ogólne: Naciśnięcie potwierdzenia specjalnego alarmu związanego ze sprzężeniem		+	+	FB3sl (M3)
CPMBRAKE	K	Sprzężenie ogólne: zachowanie się osi nadążnej przy określonych sygnałach i poleceniach Stop		+	-	FB3sl (M3)
CPMPRT	K	Sprzężenie ogólne: Zachowanie sprzężenia przy starcie programu obróbki podczas szukania i testu programu		+	+	FB3sl (M3)
CPMRESET	K	Sprzężenie ogólne: zachowanie sprzężenia podczas Reset		+	+	FB3sl (M3)
CPMSTART	K	Sprzężenie ogólne: zachowanie sprzężenia przy starcie programu obróbki		+	+	FB3sl (M3)
CPMVDI	K	Sprzężenie ogólne: zachowanie osi nadążnej po określonym sygnale interfejsu NC/PLC		+	+	FB3sl (M3)
CPOF	K	Sprzężenie ogólne: wyłącza tryb sprzężenia		+	+	FB3sl (M3)
CPON	K	Sprzężenie ogólne: włącza tryb sprzężenia		+	+	FB3sl (M3)
CPRECOF [6]	G	Wyłączenie programowanej dokładności konturu	m	+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CPRECON	G	Włączenie programowanej dokładności konturu	m	+		PGAsl
CPRES	K	Sprzężenie ogólne: uruchamia zaprojektowane dane sprzężenia wrzeciona synchronicznego		+	-	FB3sl (M3)
CPROT	P	Włączenie lub wyłączenie obszaru ochrony specyficznego dla kanału		+	-	PGAsl
CPROTDEF	P	Definicja obszaru ochrony specyficznego dla kanału		+	-	PGAsl
CPSETTYPE	K	Sprzężenie ogólne: typ sprzężenia		+	+	FB3sl (M3)
CPSYNCOP	K	Sprzężenie ogólne: wartość progowa dla synchronizacji pozycji „Zgubnie”		+	+	FB3sl (M3)
CPSYNCOP2	K	Sprzężenie ogólne: wartość progowa dla synchronizacji pozycji „Zgrubnie” 2		+	+	FB3sl (M3)
CPSYNCOV	K	Sprzężenie ogólne: Wartość progowa dla synchronizacji prędkości „Zgrubnie”		+	+	FB3sl (M3)
CPSYNFIP	K	Sprzężenie ogólne: wartość progowa dla synchronizacji prędkości „Dokładne"		+	+	FB3sl (M3)
CPSYNFIP2	K	Sprzężenie ogólne: wartość progowa dla synchronizacji pozycji „Zgubnie” 2		+	+	FB3sl (M3)
CPSYNFIV	K	Sprzężenie ogólne: wartość progowa dla synchronizacji pozycji "Dokładnie"		+	+	FB3sl (M3)
CR	A	Promień okręgu	s	+		PGsl
CROT	F	Obrót aktualnego układu współrzędnych		+	-	PGAsl
CROTS	F	Programowane obroty Frame z kątami przestrzennymi (obroty w podanych osiach)	s	+	-	PGsl
CRPL	F	Obrót Frame w dowolnej płaszczyźnie		+	-	FB1sl (K2)
CSCALE	F	Współczynnik skali dla wielu osi		+	-	PGAsl
CSPLINE	F	Spline sześcienny	m	+		PGAsl
CT	G	Okrąg z przejściem stycznym	m	+		PGsl
CTAB	F	Określona pozycja osi nadążnej na podstawie pozycji osi wiodącej z tablicy krzywych		+	+	PGAsl
CTABDEF	P	Włączenie definicji tablicy		+	-	PGAsl
CTABDEL	P	Kasowanie tablicy krzywych		+	-	PGAsl
CTABEND	P	Wyłączenie definicji tablicy		+	-	PGAsl
CTABEXISTS	F	Sprawdzanie tablicy krzywych o numerze n		+	+	PGAsl
CTABFNO	F	Liczba jeszcze możliwych tablic krzywych w pamięci		+	+	PGAsl
CTABFPOL	F	Liczba jeszcze możliwych wielomianów w pamięci		+	+	PGAsl
CTABFSEG	F	Liczba jeszcze możliwych segmentów krzywej w pamięci		+	+	PGAsl
CTABID	F	Wyświetla numer n-tej tablicy krzywych		+	+	PGAsl
CTABINV	F	Określona pozycja osi wiodącej na podstawie pozycji osi nadążnej z tablicy krzywych		+	+	PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CTABISLOCK	F	Podaje stan blokady tablicy krzywych z numerem n		+	+	PGAsl
CTABLOCK	P	Kasowanie, zastępowanie, blokowanie		+	+	PGAsl
CTABMEMTYP	F	Zwraca typ pamięci, w której jest zapisana tablica krzywych z numerem n		+	+	PGAsl
CTABMPOL	F	Maksymalna liczba możliwych wielomianów w pamięci		+	+	PGAsl
CTABMSEG	F	Maksymalna liczba możliwych segmentów krzywej w pamięci		+	+	PGAsl
CTABNO	F	Liczba zdefiniowanych tablic krzywych w SRAM lub DRAM		+	+	FB3sl (M3)
CTABNOMEM	F	Liczba zdefiniowanych tablic krzywych w SRAM lub DRAM		+	+	PGAsl
CTABPERIOD	F	Podaje okresowość tablicy krzywych z numerem n		+	+	PGAsl
CTABPOL	F	Liczba użytych wielomianów w pamięci		+	+	PGAsl
CTABPOLID	F	Liczba użytych wielomianów krzywej z tablicy krzywych z numerem n		+	+	PGAsl
CTABSEG	F	Liczba użytych segmentów krzywej w pamięci		+	+	PGAsl
CTABSEGID	F	Liczba użytych segmentów krzywej z tabeli krzywych z numerem n		+	+	PGAsl
CTABSEV	F	Podaje wartość końcową osi nadążnej segmentu tablicy krzywych		+	+	PGAsl
CTABSSV	F	Podaje wartość początkową osi nadążnej segmentu tablicy krzywych		+	+	PGAsl
CTABTEP	F	Podaje wartość osi wiodącej na końcu tablicy krzywych		+	+	PGAsl
CTABTEV	F	Podaje wartość osi nadążnej na końcu tablicy krzywych		+	+	PGAsl
CTABTMAX	F	Podaje wartość maksymalną osi nadążnej tablicy krzywych		+	+	PGAsl
CTABTMIN	F	Podaje wartość minimalną osi nadążnej tablicy krzywych		+	+	PGAsl
CTABTSP	F	Podaje wartość osi wiodącej na początku tablicy krzywych		+	+	PGAsl
CTABTSV	F	Podaje wartość osi nadążnej na początku tablicy krzywych		+	+	PGAsl
CTABUNLOCK	P	Usuwa blokadę kasowania i zastępowania		+	+	PGAsl
CTOL	A	Tolerancja konturu dla funkcji kompresora, wygładzanie orientacji i rodzaje wygładzania	m	+		PGAsl
CTRANS	F	Przesunięcie punktu zerowego dla wielu osi		+	-	PGAsl
CUT2D [6]	G	Korekcja narzędzia 2D	m	+		PGsl
CUT2DD	G	Korekcja narzędzia 2½ D w odniesieniu do narzędzia o zmienionych wymiarach	m	+		PGsl
CUT2DF	G	Korekcja narzędzia 2D względem aktualnego Frame (płaszczyzna skośna)	m	+		PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CUT2DFD	G	Korekcja narzędzia 2½-D w odniesieniu do narzędzia o zmienionych wymiarach względem aktualnego Frame (płaszczyzna skośna)	m	+		PGsl
CUT3DC	G	Korekcja narzędzia 3D dla frezowania obwodowego	m	+		PGAsl
CUT3DCC	G	Korekcja narzędzia 3D dla frezowania obwodowego uwzględniając powierzchnie ograniczające z korekcją promienia 3D: Kontur na powierzchni obróbki	m	+		PGAsl
CUT3DCCD	G	Korekcja narzędzia 3D dla frezowania obwodowego uwzględniając powierzchnie ograniczające z narzędziem o zmienionych wymiarach na torze punktu środkowego narzędzia: Dosunięcie do powierzchni ograniczającej	m	+		PGAsl
CUT3DCD	G	Korekcja narzędzia 3D w odniesieniu do narzędzia o zmienionych wymiarach dla frezowania obwodowego	m	+		PGAsl
CUT3DF	G	Korekcja narzędzia 3D dla frezowania czołowego ze zmianą orientacji	m	+		PGAsl
CUT3DFD	G	Korekcja narzędzia 3D w odniesieniu do narzędzia o zmienionych wymiarach dla frezowania czołowego ze zmianą orientacji	m	+		PGAsl
CUT3DFF	G	Korekcja narzędzia 3D dla frezowania czołowego ze stałą orientacją. Orientacją narzędzia jest kierunek ustalony przez G17 - G19 lub kierunek obrócony przez Frame.	m	+		PGAsl
CUT3DFS	G	Korekcja narzędzia 3D dla frezowania czołowego ze stałą orientacją. Orientacja narzędzia jest ustalona przez G17 - G19 i Frame nie mają na nią wpływu.	m	+		PGAsl
CUTCONOF [6]	G	Wyłączenie stałej korekcji promienia	m	+		PGsl
CUTCONON	G	Włączenie stałej korekcji promienia	m	+		PGsl
CUTMOD	A	Włącza modyfikację danych korekcji dla obrotowych narzędzi (w połączeniu z orientowanym nośnikiem narzędzi)	m	+		PGAsl
CUTMODK	A	Włącza modyfikację danych korekcji dla obrotowych narzędzi (w połączeniu z transformacjami orientacji, które definiowane są za pomocą łańcucha kinematycznego)	m	+		PGAsl
CYCLE60	C (T)	Cykl grawerowania		+		PGAsl
CYCLE61	C (T)	Frezowanie płaszczyzny		+		PGAsl
CYCLE62	C (T)	Wywołanie konturu		+		PGAsl
CYCLE63	C (T)	Frezowanie kieszeni kształtowej		+		PGAsl
CYCLE64	C (T)	Wiercenie wstępne kieszenie kształtowej		+		PGAsl
CYCLE70	C (T)	Frezowanie gwintu		+		PGAsl
CYCLE72	C (T)	Frezowanie konturu		+		PGAsl
CYCLE76	C (T)	Frezowanie czopa prostokątnego		+		PGAsl
CYCLE77	C (T)	Frezowanie czopa kołowego		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CYCLE78	C (T)	Wiercenie otworu z frezowaniem gwintu		+		PGAsl
CYCLE79	C (T)	Wielobok		+		PGAsl
CYCLE81	C (T)	Wiercenie, nawiercanie		+		PGAsl
CYCLE82	C (T)	Wiercenie, pogłębianie czołowe		+		PGAsl
CYCLE83	C (T)	Wiercenie głębokich otworów		+		PGAsl
CYCLE84	C (T)	Gwintowanie otworu bez oprawki kompensacyjnej		+		PGAsl
CYCLE85	C (T)	Rozwiercanie		+		PGAsl
CYCLE86	C (T)	Wytaczanie		+		PGAsl
CYCLE92	C (T)	Przecięcie		+		PGAsl
CYCLE95	C (T)	Skrawanie konturu		+		PGAsl
CYCLE98	C (T)	Łańcuch gwintów		+		PGAsl
CYCLE99	C (T)	Toczenie gwintu		+		PGAsl
CYCLE150	C (M)	Wyświetlanie / Protokołowanie wyników pomiaru narzędzia		+		BNMsl
CYCLE435	C (T)	Obliczanie pozycji obciągacza		+		PGAsl
CYCLE495	C (T)	Profilowanie		+		PGAsl
CYCLE750	C (A)	Wewnętrzny cykl pracy dla CYCLE751 ... CYCLE759 (zawiera polecenia MMC dla właściwego wywołania funkcji)		-		FB3sl (T4)
CYCLE751	C (A)	Otwieranie / Wykonywanie / Zamykanie sesji optymalizacyjnej		M		FB3sl (T4)
CYCLE752	C (A)	Dodanie osi w celu jej optymalizacji		M		FB3sl(T4)
CYCLE753	C (A)	Wybór trybu optymalizacji		M		FB3sl (T4)
CYCLE754	C (A)	Wstawianie / Kasowanie zestawu danych		M		FB3sl (T4)
CYCLE755	C (A)	Wykonanie / Przywracanie kopii zapasowej zestawu danych		M		FB3sl (T4)
CYCLE756	C (A)	Aktywowanie wyników optymalizacji		M		FB3sl (T4)
CYCLE757	C (A)	Zapisanie danych optymalizacyjnych		M		FB3sl (T4)
CYCLE758	C (A)	Zmiana wartości parametru		M		FB3sl (T4)
CYCLE759	C (A)	Odczytanie wartości parametru		M		FB3sl (T4)
CYCLE800	C (T)	Cykl Skrętu		+		PGAsl
CYCLE801	C (T)	Siatka lub ramka		+		PGAsl
CYCLE802	C (T)	Dowolne pozycje		+		PGAsl
CYCLE830	C (T)	Wiercenie głębokich otworów 2		+		PGAsl
CYCLE832	C (T)	High Speed Settings (nastawy obróbki szybkościowej)		+		PGAsl
CYCLE840	C (T)	Gwintowanie otworu z oprawką kompensacyjną		+		PGAsl
CYCLE899	C (T)	Frezowanie rowka otwartego		+		PGAsl
CYCLE930	C (T)	Rowek		+		PGAsl
CYCLE940	C (T)	Kształty podcięć		+		PGAsl
CYCLE951	C (T)	Skrawanie		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CYCLE952	C (T)	Toczenie konturu metodą wcinania		+		PGAsl
CYCLE961	C (M)	Określenie położenia przedmiotu obrabianego i zastosowanie go jako przesunięcie punktu zerowego		+		BNMsl
CYCLE971	C (M)	Kalibracja sondy do pomiaru narzędzi, pomiar długości narzędzia i/lub promienia narzędzia (tylko dla technologii frezowania)		+		BNMsl
CYCLE973	C (M)	Kalibracja sondy do pomiaru obrabianych przedmiotów na powierzchni obrabianego przedmiotu lub w rowku (tylko dla technologii toczenia)		+		BNMsl
CYCLE974	C (M)	Określenie punktu zerowego obrabianego przedmiotu w wybranej osi pomiaru lub korekcja narzędzia z pomiarem 1-punktowym (tylko dla technologii toczenia)		+		BNMsl
CYCLE976	C (M)	Kalibracja sondy do pomiaru obrabianych przedmiotów w pierścieniu kalibracyjnym lub na kuli kalibracyjnej, kompletnie w płaszczyźnie roboczej albo na krawędzi dla określonej osi i kierunku.		+		BNMsl
CYCLE977	C (M)	Określenie punktu środkowego w płaszczyźnie oraz szerokości lub średnicy		+		BNMsl
CYCLE978	C (M)	Pomiar pozycji krawędzi w układzie współrzędnych obrabianego przedmiotu		+		BNMsl
CYCLE979	C (M)	Pomiar punktu środkowego na płaszczyźnie i promienia fragmentów łuków.		+		BNMsl
CYCLE982	C (M)	Kalibracja sondy do pomiaru narzędzi, pomiar narzędzi tokarskich, wiertarskich i frezarskich (tylko w technologii toczenia)		+		BNMsl
CYCLE994	C (M)	Określenie punktu zerowego obrabianego przedmiotu w wybranej osi pomiaru z pomiarem 2-punktowym (tylko w technologii toczenia)		+		BNMsl
CYCLE995	C (M)	Pomiar odchylenia kątowego wrzeciona na obrabiarce		+		BNMsl
CYCLE996	C (M)	Określenie parametrów transformacji istotnych dla transformacji kinematycznej z osiami obrotowymi		+		BNMsl
CYCLE997	C (M)	Określenie punktu środkowego i średnicy kulki, pomiar punktów środkowych trzech rozmieszczonych kul		+		BNMsl
CYCLE998	C (M)	Określenie położenia kątowego powierzchni (płaszczyzny) w odniesieniu do płaszczyzny roboczej i kąta krawędzi w układzie współrzędnych obrabianego przedmiotu		+		BNMsl
CYCLE4071	C (T)	Szlifowanie wzdłużne z dosuwem w punkcie nawrotu		+		PGAsl
CYCLE4072	C (T)	Szlifowanie wzdłużne z dosuwem w punkcie nawrotu i sygnałem przerwania		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
CYCLE4073	C (T)	Szlifowanie wzdłużne z dosuwem ciągłym		+		PGAsl
CYCLE4074	C (T)	Szlifowanie wzdłużne z dosuwem ciągłym i sygnałem przerwania		+		PGAsl
CYCLE4075	C (T)	Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu		+		PGAsl
CYCLE4077	C (T)	Szlifowanie płaszczyzny z dosuwem w punkcie nawrotu i sygnałem przerwania		+		PGAsl
CYCLE4078	C (T)	Szlifowanie płaszczyzny z dosuwem ciągłym		+		PGAsl
CYCLE4079	C (T)	Szlifowanie płaszczyzny z przerywanym dosuwem		+		PGAsl

Polecenia D - F

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
D	A	Numer korekcji narzędzia		+		PGsl
D0	A	Przy zaprogramowaniu D0 korekcje narzędzia nie działają		+		PGsl
DAC	K	Absolutne, pojedynczymi blokami, specyficzne dla osi programowanie na średnicy	s	+		PGsl
DC	K	Podawanie wymiarów absolutnych dla osi obrotowych, z bezpośrednim najazdem na pozycję	s	+		PGsl
DCI	K	Przyporządkowanie klasy danych I (=indywidualnie) (tylko w SINUMERIK 828D!)		+		PGAsl
DCM	K	Przyporządkowanie klasy danych M (=producent) (tylko w SINUMERIK 828D!)		+		PGAsl
DCU	K	Przyporządkowanie klasy danych U (= Użytkownik) (tylko w SINUMERIK 828D!)		+		PGAsl
DEF	K	Definicja zmiennej		+		PGAsl
DEFAULT	K	Domyślna gałąź w rozgałęzieniu CASE		+		PGAsl
DEFINE	K	Słowo kluczowe do definicji makr		+		PGAsl
DELAYFSTOF	P	Zdefiniowanie końca obszaru Stopp-Delay	m	+	-	PGAsl
DELAYFSTON	P	Zdefiniowanie początku obszaru Stopp-Delay	m	+	-	PGAsl
DELDL	F	Kasowanie addytywnych korekcji		+	-	PGAsl
DELDTG	P	Kasowanie pozostałej drogi		-	+	FBSYsl
DELETE	P	Kasowanie podanego pliku. Nazwę pliku można podać ze ścieżką i rozszerzeniem pliku.		+	-	PGAsl
DELMLOWNER	F	Kasowanie miejsca narzędzia w magazynie		+	-	FBWsl
DELMLRES	F	Kasowanie rezerwacji miejsca w magazynie		+	-	FBWsl
DELMT	P	Kasowanie Multitool		+	-	FBWsl
DELOBJ	F	Kasowanie elementów łańcuchów kinematycznych, obszarów ochrony oraz elementów obszarów ochrony, par kolizyjnych i danych transformacji		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1][2][3][4][5] Objaśnienia patrz legenda (Strona 895).						
DELT	P	Kasowanie narzędzia		+	-	FBWsl
DELTC	P	Kasowanie zestawu danych nośnika narzędzi		+	-	FBWsl
DELTOOLENV	F	Kasowanie zestawów danych do opisu środowisk narzędziowych		+	-	PGAsl
DIACYCOFA	K	Specyficzne dla osi, modalne programowanie na średnicy: WYŁ. w cyklach	m	+		FB1sl (P1)
DIAM90	G	Programowanie na średnicy dla G90, programowanie na promieniu dla G91	m	+		PGAsl
DIAM90A	K	Specyficzne dla osi, modalne programowanie na średnicy dla G90 i AC, programowanie na promieniu dla G91 i IC	m	+		PGsl
DIAMCHAN	K	Przejęcie wszystkich osi z funkcji osi MD do stanu aktualnego kanału na programowanie na średnicy		+		PGsl
DIAMCHANA	K	Przejęcie stanu kanału do programowania na średnicy		+		PGsl
DIAMCYCOF	G	Specyficzne dla kanału programowanie na średnicy: WYŁ. w cyklach	m	+		FB1sl (P1)
DIAMOF [6]	G	Programowanie na średnicy: WYŁ. Położenie podstawowe patrz producent maszyny	m	+		PGsl
DIAMOFA	K	Specyficzne dla osi, modalne programowanie na średnicy: WYŁ. Położenie podstawowe patrz producent maszyny	m	+		PGsl
DIAMON	G	Programowanie na średnicy: WŁ	m	+		PGsl
DIAMONA	K	Specyficzne dla osi, modalne programowanie na średnicy: WŁ. Włączenie patrz producent maszyny	m	+		PGsl
DIC	K	Względne, pojedynczymi blokami, specyficzne dla osi programowanie na średnicy	s	+		PGsl
DILF	A	Droga wycofania (długość)	m	+		PGsl
DISABLE	P	Interrupt WYŁ.		+	-	PGAsl
DISC	A	Zwiększenie okręgu przejściowego, korekcja promienia narzędzia	m	+		PGsl
DISCL	A	Odstęp punktu końcowego szybkiego ruchu dosuwu od płaszczyzny obróbki		+		PGsl
DISPLOF	PA	Blokowanie wyświetlania aktualnego bloku		+		PGAsl
DISPLON	PA	Wyłączenie blokowania wyświetlania aktualnego bloku		+		PGAsl
DISPR	A	Różnica toru dla funkcji repos	s	+		PGAsl
DISR	A	Odstęp dla funkcji repos	s	+		PGAsl
DISRP	A	Odległość płaszczyzny wycofania od płaszczyzny obróbki przy miękkim dosunięciu i odsunięciu		+		PGsl
DITE	A	Droga wybiegu gwintu	m	+		PGsl
DITS	A	Droga dobiegu gwintu	m	+		PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1][2][3][4][5] Objaśnienia patrz legenda (Strona 895).						
DIV	K	Dzielenie bez reszty		+		PGAsl
DL	A	Wybór zależnej od miejsca addytywnej korekcji narzędzia (DL, korekcja sumaryczna i ustawcza)	m	+		PGAsl
DO	A	Słowo kluczowe dla akcji synchronicznej, przy spełnionym warunku wyzwala akcję		-	+	FBSYsl
DRFOF	P	Wyłączenie przesunięć kółkiem ręcznym (DRF)	m	+	-	PGsl
DRIVE	G	Przyspieszenie ruchu po torze w zależności od prędkości	m	+		PGAsl
DRIVEA	P	Włączenie załamanej charakterystyki przyspieszenia dla zaprogramowanych osi		+	-	PGAsl
DYNFINISH	G	Dynamika obróbki wykańczającej	m	+		PGAsl
DYNNORM [6]	G	Normalna dynamika	m	+		PGAsl
DYNPOS	G	Dynamika pozycjonowania, gwintowania otworu	m	+		PGAsl
DYNROUGH	G	Dynamika obróbki zgrubnej	m	+		PGAsl
DYNSEMIFIN	G	Dynamika obróbki wstępnie wykańczającej	m	+		PGAsl
DZERO	P	Oznacza wszystkie numery D jednostki TO jako nieobowiązujące		+	-	PGAsl
EAUTO	G	Definicja ostatniego segmentu Spline przez ostatnie 3 punkty	m	+		PGAsl
EGDEF	P	Definicja przekładni elektronicznej		+	-	PGAsl
EGDEL	P	Kasowanie definicji sprzężenia dla osi nadążnej		+	-	PGAsl
EGOFC	P	Wyłączenie przekładni elektronicznej w sposób ciągły		+	-	PGAsl
EGOFS	P	Wyłączenie przekładni elektronicznej w sposób selektywny		+	-	PGAsl
EGON	P	Włączenie przekładni elektronicznej		+	-	PGAsl
EGONSYN	P	Włączenie przekładni elektronicznej		+	-	PGAsl
EGONSYNE	P	Włączenie przekładni elektronicznej z zadaniem trybu dosunięcia		+	-	PGAsl
ELSE	K	Rozgałęzienie programu, gdy warunek IF nie jest spełniony		+		PGAsl
ENABLE	P	Interrupt WŁ.		+	-	PGAsl
ENAT [6]	G	Naturalne przejście krzywej do następnego bloku	m	+		PGAsl
ENDFOR	K	Wiersz końcowy pętli FOR		+		PGAsl
ENDIF	K	Wiersz końcowy rozgałęzienia IF		+		PGAsl
ENDLABEL	K	Znacznik końcowy dla powtórzeń programu obróbki przez REPEAT		+		PGAsl, FB1sl (K1)
ENDLOOP	K	Wiersz końcowy pętli programowej bez końca LOOP		+		PGAsl
ENDPROC	K	Wiersz końcowy programu z wierszem początkowym PROC		+		

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1][2][3][4][5] Objaśnienia patrz legenda (Strona 895).						
ENDWHILE	K	Wiersz końcowy z pętlą WHILE		+		PGAsl
ESRR	P	Parametryzacja niezależnego napędu dla wycofania ESR		+		PGAsl
ESRS	P	Parametryzacja niezależnego napędu dla zatrzymania ESR		+		PGAsl
ETAN	G	Styczne przejście krzywej do następnego bloku na końcu Spline	m	+		PGAsl
EVERY	K	Wykonanie akcji synchronicznej przy zmianie warunku z FALSE na TRUE		-	+	FBSYsl
EX	K	Słowo kluczowe dla przypisania wartości zapisie wykładniczym		+		PGAsl
EXECSTRING	P	Wykonanie wiersza bloków programu obróbki zawartych w zmiennej String		+	-	PGAsl
EXECTAB	P	Wykonanie elementu z tablicy ruchów		+	-	PGAsl
EXECUTE	P	Wykonanie programu WŁ.		+	-	PGAsl
EXP	F	Funkcja wykładnicza ex		+	+	PGAsl
EXTCALL	A	Wykonywanie podprogramu zewnętrznego		+	+	PGAsl
EXTCLOSE	P	Zamknięcie zewnętrznego urządzenia/pliku otwartego w celu zapisu		+	-	PGAsl
EXTERN	K	Deklaracja podprogramu z przekazaniem parametrów		+		PGAsl
EXTOPEN	P	Otwarcie zewnętrznego urządzenia/pliku kanału w celu zapisu		+	-	PGAsl
F	A	Wartość posuwu (w połączeniu z G4 pod parametrem F programowany jest również czas oczekiwania)		+	+	PGsl
FA	K	Posuw osiowy	m	+	+	PGsl
FAD	A	Posuw dosuwu dla miękkiego dosunięcia i odsunięcia		+		PGsl
FALSE	K	Stała logiczna: fałsz		+	+	PGAsl
FB	A	Posuw pojedynczymi blokami		+		PGsl
FCTDEF	P	Definicja funkcji wielomianowej		+	-	PGAsl
FCUB	G	Posuw zmienny według funkcji Spline trzeciego stopnia	m	+		PGAsl
FD	A	Posuw po torze z nałożeniem ruchu przy pomocy kółka ręcznego	s	+		PGsl
FDA	K	Posuw osiowy z nałożeniem ruchu przy pomocy kółka ręcznego	s	+		PGsl
FENDNORM [6]	G	Zwłoka na narożach WYŁ.	m	+		PGAsl
FFWOF [6]	G	Sterowanie wyprzedzające WYŁ.	m	+		PGAsl
FFWON	G	Sterowanie wyprzedzające WŁ.	m	+		PGAsl
FGREF	K	Promień odniesienia przy osiach obrotowych albo współczynniki odniesienia toru przy osiach orientacji (interpolacja wektorowa)	m	+		PGsl
FGROUP	P	Ustalenie osi z posuwem po torze		+	-	PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1][2][3][4][5] Objaśnienia patrz legenda (Strona 895).						
FI	K	Parametry dostępu do danych frame: przesunięcie dokładne		+		PGAsl
FIFOCTRL	G	Sterowanie buforem przebiegu wyprzedzającego	m	+		PGAsl
FILEDATE	P	Podaje datę ostatniego dostępu do pliku w celu zapisu		+	-	PGAsl
FILEINFO	P	Podaje status FILEDATE, FILESIZE, FILESTAT i FILETIME		+	-	PGAsl
FILESIZE	P	Podaje aktualny rozmiar pliku		+	-	PGAsl
FILESTAT	P	Podaje status pliku odnośnie praw do odczytu, zapisu, wykonania, wyświetlenia, usuwania (rwxsd)		+	-	PGAsl
FILETIME	P	Podaje godzinę ostatniego dostępu do pliku w celu zapisu		+	-	PGAsl
FINEA	K	Koniec ruchu przy osiągnięciu "zatrzymania dokładnego dokładnie"	m	+		PGAsl
FL	K	Prędkość graniczna dla osi synchronicznych	m	+		PGsl
FLIN	G	Posuw zmienny liniowo	m	+		PGAsl
FMA	K	Wiele posuwów osiowych	m	+		PGsl
FNORM [6]	G	Posuw normalny według normy DIN66025	m	+		PGAsl
FOC	K	Ograniczenia momentu/siły działające pojedynczymi blokami	s	-	+	FBSYsl
FOCOF	K	Wyłączenie modalnego ograniczenia momentu/siły	m	-	+	FBSYsl
FOCON	K	Włączenie modalnego ograniczenia momentu/siły	m	-	+	FBSYsl
FOR	K	Pętla FOR o stałej liczbie przebiegów		+		PGAsl
FP	A	Punkt stały: Numer punktu stałego, do którego ma zostać wykonany ruch	s	+		PGsl
FPO	K	Przebieg posuwu zaprogramowany przez wielomian		+		PGAsl
FPR	P	Oznaczenie osi obrotowej		+	-	PGsl
FPRAOF	P	Wyłączenie posuwu na obrót		+	-	PGsl
FPRAON	P	Włączenie posuwu na obrót		+	-	PGsl
FRAME	K	Typ danych do transformacji układu współrzędnych		+		PGAsl
FRC	A	Posuw dla zaokrąglenia i fazy	s	+		PGsl
FRCM	A	Modalny posuw dla zaokrąglenia i fazy	m	+		PGsl
FROM	K	Akcja jest wykonywana, gdy warunek zostanie raz spełniony i tak długo jak akcja synchroniczna jest aktywna		-	+	FBSYsl
FTOC	P	Zmiana korekcji dokładnej narzędzia		-	+	FBSYsl
FTOCOF [6]	G	Działająca online dokładna korekcja narzędzia WYŁ.	m	+		PGAsl
FTOCON	G	Działająca online dokładna korekcja narzędzia WŁ.	m	+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1][2][3][4][5] Objaśnienia patrz legenda (Strona 895).						
FXS	K	Ruch do twardego zderzaka wł.	m	+	+	PGsl
FXST	K	Granica momentu dla ruchu do zderzaka twardego	m	+	+	PGsl
FXSW	K	Okno nadzoru dla ruchu do zderzaka twardego		+	+	PGsl
FZ	K	Posuw na ostrze	m	+		PGsl

Polecenia G ... L

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
G0	G	Interpolacja liniowa z posuwem szybkim (ruch posuwem szybkim)	m	+		PGsl
G1 [6]	G	Interpolacja liniowa z posuwem (interpolacja prostoliniowa)	m	+		PGsl
G2	G	Interpolacja kołowa zgodnie z ruchem wskazówek zegara	m	+		PGsl
G3	G	Interpolacja kołowa przeciwnie do ruchu wskazówek zegara	m	+		PGsl
G4	G	Czas oczekiwania, wstępnie określony	s	+		PGsl
G5	G	Szlifowanie wcinające skośne	s	+		PGAsl
G7	G	Ruch wyrównawczy przy szlifowaniu wcinającym skośnym	s	+		PGAsl
G9	G	Zatrzymanie dokładne - zmniejszenie prędkości	s	+		PGsl
G17 [6]	G	Wybór płaszczyzny roboczej X/Y	m	+		PGsl
G18	G	Wybór płaszczyzny roboczej Z/X	m	+		PGsl
G19	G	Wybór płaszczyzny roboczej Y/Z	m	+		PGsl
G25	G	Dolne ograniczenie obszaru pracy	s	+		PGsl
G26	G	Górne ograniczenie obszaru pracy	s	+		PGsl
G33	G	Nacinanie gwintu o stałym skoku	m	+		PGsl
G34	G	Nacinanie gwintu o skoku rosnącym liniowo	m	+		PGsl
G35	G	Nacinanie gwintu o skoku malejącym liniowo	m	+		PGsl
G40 [6]	G	Korekcja promienia narzędzia WYŁ.	m	+		PGsl
G41	G	Korekcja promienia narzędzia na lewo od konturu	m	+		PGsl
G42	G	Korekcja promienia narzędzia na prawo od konturu	m	+		PGsl
G53	G	Blokowanie aktualnego przesunięcia punktu zerowego (pojedynczymi blokami)	s	+		PGsl
G54	G	Pierwsze ustawiane przesunięcie punktu zerowego	m	+		PGsl
G55	G	Drugie ustawiane przesunięcie punktu zerowego	m	+		PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
G56	G	Trzecie ustawiane przesunięcie punktu zerowego	m	+		PGsl
G57	G	Czwarte ustawiane przesunięcie punktu zerowego	m	+		PGsl
G58 (840D sl)	G	Absolutne programowalne przesunięcie punktu zerowego (przesunięcie zgrubne)	s	+		PGsl
G58 (828D)	G	Piąte ustawiane przesunięcie punktu zerowego	m	+		PGsl
G59 (840D sl)	G	Addytywne programowane przesunięcie punktu zerowego (przesunięcie dokładne)	s	+		PGsl
G59 (828D)	G	Szóste ustawiane przesunięcie punktu zerowego	m	+		PGsl
G60 [6]	G	Zatrzymanie dokładne - zmniejszenie prędkości	m	+		PGsl
G62	G	Zwłoka na narożach wewnętrznych przy aktywnej korekcji promieni narzędzia (G41, G42)	m	+		PGAsl
G63	G	Gwintowanie otworu z oprawką kompensacyjną	s	+		PGsl
G64	G	Tryb przechodzenia płynnego	m	+		PGsl
G70	G	Calowe podawanie wymiarów dla danych geometrycznych (długości)	m	+	+	PGsl
G71 [6]	G	Metryczne podawanie wymiarów dla danych geometrycznych (długości)	m	+	+	PGsl
G74	G	Najazd na punkt referencyjny	s	+		PGsl
G75	G	Dosunięcie do punktu stałego	s	+		PGsl
G90 [6]	G	Absolutne podanie wymiaru	m/s	+		PGsl
G91	G	Przyrostowe podanie wymiarów	m/s	+		PGsl
G93	G	Zależny od czasu posuw 1/min	m	+		PGsl
G94 [6]	G	Posuw liniowy F w mm/min lub calach/min i stopniach/min	m	+		PGsl
G95	G	Posuw na obrót F w mm/obr. lub calach/obr.	m	+		PGsl
G96	G	Posuw na obrót (jak przy G95) i stała prędkość skrawania	m	+		PGsl
G97	G	Posuw na obrót i stała prędkość obrotowa wrzeciona (stała prędkość skrawania WYŁ)	m	+		PGsl
G110	G	Zaprogramowanie bieguna w stosunku do ostatniej zaprogramowanej pozycji zadanej	s	+		PGsl
G111	G	Zaprogramowanie bieguna w stosunku do punktu zerowego aktualnego układu współrzędnych przedmiotu obrabianego	s	+		PGsl
G112	G	Zaprogramowanie bieguna w stosunku do ostatniego obowiązującego bieguna	s	+		PGsl
G140 [6]	G	Kierunek dosunięcia WAB ustalony przez G41/G42	m	+		PGsl
G141	G	Kierunek dosunięcia WAB na lewo od konturu	m	+		PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
G142	G	Kierunek dosunięcia WAB na prawo od konturu	m	+		PGsl
G143	G	Kierunek dosunięcia WAB zależny od stycznej	m	+		PGsl
G147	G	Miękkie dosunięcie po prostej	s	+		PGsl
G148	G	Miękkie odsunięcie po prostej	s	+		PGsl
G153	G	Blokowanie aktualnych frame łącznie z frame bazowym	s	+		PGsl
G247	G	Miękkie dosunięcie po ćwierćokręgu	s	+		PGsl
G248	G	Miękkie odsunięcie po ćwierćokręgu	s	+		PGsl
G290 [6]	G	Przełączenie na tryb SINUMERIK WŁ.	m	+		FBWsl
G291	G	Przełączenie na tryb ISO2/3 WŁ.	m	+		FBWsl
G331	G	Gwintowanie otworu bez oprawki kompensacyjnej, skok dodatni, obroty w prawo	m	+		PGsl
G332	G	Gwintowanie otworu bez oprawki kompensacyjnej, skok ujemny, obroty w lewo	m	+		PGsl
G335	G	Toczenie gwintów beczkowatych zgodnie z ruchem wskazówek zegara	m	+		PGsl
G336	G	Toczenie gwintów beczkowatych przeciwnie do ruchu wskazówek zegara	m	+		PGsl
G340 [6]	G	Blok dosunięcia przestrzennego (równocześnie na głębokości i w płaszczyźnie (linia spiralna))	m	+		PGsl
G341	G	Najpierw dosuw w osi prostopadłej (z), następnie dosunięcie w płaszczyźnie	m	+		PGsl
G347	G	Miękkie dosunięcie po półokręgu	s	+		PGsl
G348	G	Miękkie odsunięcie po półokręgu	s	+		PGsl
G450 [6]	G	Okrąg przejściowy	m	+		PGsl
G451	G	Punkt przecięcia równoległych	m	+		PGsl
G460 [6]	G	Włączenie nadzoru nad kolizją dla bloku dosunięcia i odsunięcia	m	+		PGsl
G461	G	Wstawienie okręgu w bloku WRK	m	+		PGsl
G462	G	Wstawienie prostej w bloku WRK	m	+		PGsl
G500 [6]	G	Wyłączenie wszystkich ustawianych frame, frame bazowe są aktywne	m	+		PGsl
G505 ... G599	G	5 ... 99. Ustawiane przesunięcie punktu zerowego	m	+		PGsl
G601 [6]	G	Zmiana bloku przy zatrzymaniu dokładnym dokładnie	m	+		PGsl
G602	G	Zmiana bloku przy zatrzymaniu dokładnym zgrubnie	m	+		PGsl
G603	G	Zmiana bloku na końcu bloku IPO	m	+		PGsl
G621	G	Zwłoka na wszystkich narożach	m	+		PGAsl
G641	G	Tryb przechodzenia płynnego ze ścinaniem naroży według kryterium drogi (= programowana droga wygładzania)	m	+		PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
G642	G	Tryb przechodzenia płynnego ze ścinaniem naroży przy zachowaniu zdefiniowanych tolerancji	m	+		PGsl
G643	G	Tryb przechodzenia płynnego ze ścinaniem naroży przy zachowaniu zdefiniowanych tolerancji (wewnątrz bloku)	m	+		PGsl
G644	G	Tryb przechodzenia płynnego ze ścinaniem naroży z maksymalną możliwą dynamiką	m	+		PGsl
G645	G	Tryb przechodzenia płynnego ze ścinaniem naroży i stycznymi przejściami między blokami przy zachowaniu zdefiniowanych tolerancji	m	+		PGsl
G700	G	Calowe podawanie wymiarów dla danych geometrycznych i technologicznych (długości, posuw)	m	+	+	PGsl
G710 [6]	G	Metryczne podawanie wymiarów dla danych geometrycznych i technologicznych (długości, posuw)	m	+	+	PGsl
G810 [6], ..., G819	G	Grupa G zarezerwowana dla użytkownika OEM		+		PGAsl
G820 [6], ..., G829	G	Grupa G zarezerwowana dla użytkownika OEM		+		PGAsl
G931	G	Zadanie posuwu przez czas ruchu z wyłączeniem stałej prędkości ruchu po torze	m	+		
G942	G	Zamrożenie posuwu liniowego i stałej prędkości skrawania albo prędkości obrotowej wrzeciona	m	+		
G952	G	Zamrożenie posuwu na obrót i stałej prędkości skrawania albo prędkości obrotowej wrzeciona	m	+		
G961	G	Posuw liniowy (jak przy G94) i stała prędkość skrawania	m	+		PGsl
G962	G	Posuw liniowy lub posuw na obrót i stała prędkość skrawania	m	+		PGsl
G971	G	Posuw liniowy i stała prędkość obrotowa wrzeciona (stała prędkość skrawania WYŁ)	m	+		PGsl
G972	G	Posuw liniowy lub posuw na obrót i stała prędkość obrotowa wrzeciona (stała prędkość skrawania WYŁ)	m	+		PGsl
G973	G	Posuw na obrót bez ograniczenia prędkości obrotowej wrzeciona i stała prędkość obrotowa wrzeciona (G97 bez LIMS dla trybu ISO)	m	+		PGsl
GEOAX	P	Przyporządkowanie nowych osi kanału do osi geometrycznych 1 - 3		+	-	PGAsl
GET	P	Zamiana udostępnionych osi między kanałami		+	+	PGAsl
GETACTT	F	Określa aktywne narzędzie z grupy narzędzi o takiej samej nazwie		+	-	FBWsl
GETACTTD	F	Określa przynależny numer T do absolutnego numeru D		+	-	PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
GETD	P	Zamiana osi bezpośrednio między kanałami		+	-	PGAsl
GETDNO	F	Podaje numer D ostrza (CE) narzędzia (T)		+	-	PGAsl
GETEXET	P	Odczyt założonego numeru T		+	-	FBWsl
GETFREELOC	P	Szukanie wolnego miejsca w magazynach dla danego narzędzia		+	-	FBWsl
GETSELT	P	Podanie wstępnie wybranego numeru T		+	-	FBWsl
GETT	F	Określenie numeru T do nazwy narzędzia		+	-	FBWsl
GETTCOR	F	Odczyt długości narzędzia lub składowych długości narzędzia		+	-	PGAsl
GETTENV	F	Odczyt numerów T, D i DL		+	-	PGAsl
GETVARAP	F	Odczyt prawa dostępu do zmiennej systemowej/użytkownika		+	-	PGAsl
GETVARDFT	F	Odczyt wartości standardowej zmiennej systemowej/użytkownika		+	-	PGAsl
GETVARLIM	F	Odczyt wartości granicznych zmiennej systemowej/użytkownika		+	-	PGAsl
GETVARPHU	F	Odczyt jednostki fizycznej zmiennej systemowej/użytkownika		+	-	PGAsl
GETVARTYP	F	Odczyt typu danych zmiennych systemowych/ użytkownika		+	-	PGAsl
GFRAME0 ... GFRAME100	G	Aktywowanie <n> frame dla szlifowania z obszaru danych w kanale	m	+		PGsl
GOTO	K	Polecenie skoku najpierw do przodu a następnie do tyłu (kierunek najpierw do końca a następnie do początku programu)		+		PGAsl
GOTOB	K	Polecenie skoku do tyłu (kierunek do początku programu)		+		PGAsl
GOTOC	K	Działa jak funkcja GOTO z tą różnicą, że alarm 14080 "Cel skoku nie został znaleziony" jest ukrywany		+		PGAsl
GOTOF	K	Polecenie skoku do przodu (kierunek do końca programu)		+		PGAsl
GOTOS	K	Skok powrotny do początku programu		+		PGAsl
GP	K	Słowo kluczowe do pośredniego programowania atrybutów pozycji		+		PGAsl
GROUP_ ADDEND	C (T)	Koniec sekcji dodatkowego dosuwu		+		PGAsl
GROUP_BEGIN	C (T)	Początek bloku zbiorczego programu		+		PGAsl
GROUP_END	C (T)	Koniec bloku zbiorczego programu		+		PGAsl
GWPSOF	P	Odwołanie wyboru stałej prędkości obwodowej ściernicy (SUG)	s	+	-	PGsl
GWPSON	P	Wybór stałej prędkości obwodowej ściernicy (SUG)	s	+	-	PGsl
H...	A	Wyprowadzenie funkcji pomocniczej do PLC		+	+	PGsl/FB1sl (H2)
HOLES1	C (T)	Szereg otworów		+		PGAsl
HOLES2	C (T)	Okrąg otworów		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
I	A	Parametr interpolacji	s	+		PGsl
I1	A	Współrzędne punktu pośredniego	s	+		PGsl
IC	K	Przyrostowe podawanie wymiarów	s	+		PGsl
ICYCOF	P	Wykonanie wszystkich bloków jednego cyklu technologicznego według ICYCOF w jednym takcie IPO		+	+	FBSYsl
ICYCON	P	Wykonanie każdego bloku cyklu technologicznego według ICYCON w oddzielnym takcie IPO		+	+	FBSYsl
ID	K	Oznaczenie modalnych akcji synchronicznych	m	-	+	FBSYsl
IDS	K	Oznaczenie modalnych statycznych akcji synchronicznych		-	+	FBSYsl
IF	K	Rozpoczęcie skoku warunkowego w programie obróbki / cyklu technologicznym		+	+	PGAsl
INDEX	F	Określenie indeksu znaku w wejściowym łańcuchu znaków		+	-	PGAsl
INICF	K	Inicjalizacja zmiennych przy użyciu NEWCONF		+		PGAsl
INIPO	K	Inicjalizacja zmiennych przy PowerOn		+		PGAsl
INIRE	K	Inicjalizacja zmiennych przy Reset		+		PGAsl
INIT	P	Wybór określonego programu NC do wykonywania w określonym kanale		+	-	PGAsl
INITIAL		Utworzenie pliku INI dla wszystkich zakresów		+		PGAsl
INT	K	Typ danych: Wartość całkowita ze znakiem		+		PGAsl
INTERSEC	F	Obliczenie punktu przecięcia między dwoma elementami konturu		+	-	PGAsl
INVCCW	G	Ruch po ewolwencie, przeciwnie do ruchu wskazówek zegara	m	+		PGsl
INVCW	G	Ruch po ewolwencie, zgodnie z ruchem wskazówek zegara	m	+		PGsl
INVFRAME	F	Obliczenie frame przeciwnego dla danego frame		+	-	FB1sl (K2)
IP	K	Zmienny parametr interpolacji		+		PGAsl
IPOBRKA	P	Kryterium ruchu od punktu początkowego charakterystyki hamowania	m	+	+	
IPOENDA	K	Koniec ruchu przy osiągnięciu "Stop IPO"	m	+		PGAsl
IPTRLOCK	P	Zamrożenie początku niezdatnego do szukania segmentu programu na najbliższy blok funkcjonalny maszyny	m	+	-	PGAsl
IPTRUNLOCK	P	Ustawienie końca niezdatnego do szukania segmentu programu na aktualny blok w chwili przerwania	m	+	-	PGAsl
IR	A	Współrzędne punktu środkowego okręgu (kierunek X) przy toczeniu gwintów beczkowatych		+		PGsl
ISAXIS	F	Sprawdzanie, czy podana jako parametr oś geometryczna 1 istnieje		+	-	PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
ISD	A	Głębokość zagłębienia	m	+		PGAsl
ISFILE	F	Sprawdzanie, czy plik istnieje w pamięci użytkownika NC		+	-	PGAsl
ISNUMBER	F	Sprawdzanie, czy wejściowy łańcuch znaków można zamienić w liczbę		+	-	PGAsl
ISOCALL	K	Pośrednie wywołanie programu w języku ISO		+		PGAsl
ISVAR	F	Sprawdzenie, czy przekazany parametr zawiera zmienną znaną w NC		+	-	PGAsl
J	A	Parametr interpolacji	s	+		PGsl
J1	A	Współrzędne punktu pośredniego	s	+		PGsl
JERKA	P	Uaktywnienie ustawionego przez MD przyspieszenia dla programowanych osi		+	-	
JERKLIM	K	Zmniejszenie lub zwiększenie maksymalnego osiowego przyspieszenia drugiego stopnia	m	+		PGAsl
JERKLIMA	K	Zmniejszenie lub zwiększenie maksymalnego osiowego przyspieszenia drugiego stopnia	m	+	+	PGAsl
JR	A	Współrzędne punktu środkowego okręgu (kierunek Y) przy toczeniu gwintów beczkowatych		+		PGsl
K	A	Parametr interpolacji	s	+		PGsl
K1	A	Współrzędne punktu pośredniego	s	+		PGsl
KONT	G	Obejście konturu przy korekcji narzędzia	m	+		PGsl
KONTC	G	Dosunięcie/odsunięcie po krzywej wielomianowej o stałej krzywiźnie	m	+		PGsl
KONTT	G	Dosunięcie/odsunięcie po krzywej wielomianowej o stałej pochodnej	m	+		PGsl
KR	A	Współrzędne punktu środkowego okręgu (kierunek Z) przy toczeniu gwintów beczkowatych		+		PGsl
L	A	Numer podprogramu	s	+	+	PGAsl
LEAD	A	Kąt wyprzedzenia 1. Orientacja narzędzia 2. Wielomiany orientacji	m	+		PGAsl
LEADOF	P	Sprzężenie osiowe wartości wiodącej WYŁ.		+	+	PGAsl
LEADON	P	Sprzężenie osiowe wartości wiodącej WŁ.		+	+	PGAsl
LENTOAX	F	Podaje informacje o przyporządkowaniu długości aktywnego narzędzia L1, L2 i L3 do odciętej, rzędnej i aplikaty		+	-	PGAsl
LFOF [6]	G	Szybkie wycofanie przy nacinaniu gwintu WYŁ.	m	+		PGsl
LFON	G	Szybkie wycofanie przy nacinaniu gwintu WŁ.	m	+		PGsl
LFPOS	G	Wycofanie osi podanej przy pomocy POLFMASK lub POLFMLIN do absolutnej pozycji osi zaprogramowanej przy pomocy POLF	m	+		PGsl
LFTXT [6]	G	Płaszczyzna ruchu wycofania przy szybkim cofnięciu jest określana ze stycznej do toru i aktualnego kierunku narzędzia	m	+		PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
LFWP	G	Płaszczyzna ruchu wycofania przy szybkim cofnięciu jest określana przez aktualną płaszczyznę roboczą (G17/G18/G19)	m	+		PGsl
LIFTFAST	K	Szybkie cofnięcie		+		PGsl
LIMS	K	Ograniczenie prędkości obrotowej przy G96/G961 i G97	m	+		PGsl
LLI	K	Dolna wartość graniczna zmiennych		+		PGAsl
LN	F	Logarytm naturalny		+	+	PGAsl
LOCK	P	Zablokowanie akcji synchronicznej z ID (zatrzymanie cyklu technologicznego)		-	+	FBSYsl
LONGHOLE	C (T)	Otwór podłużny		+		PGAsl
LOOP	K	Wprowadzenie pętli bez końca		+		PGAsl

Polecenia M ... R

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
M0		Programowany stop		+	+	PGsl
M1		Stop warunkowy		+	+	PGsl
M2		Koniec programu, program główny (jak M30)		+	+	PGsl
M3		Kierunek obrotów wrzeciona w prawo		+	+	PGsl
M4		Kierunek obrotów wrzeciona w lewo		+	+	PGsl
M5		Zatrzymanie wrzeciona		+	+	PGsl
M6		Wybór narzędzia		+	+	PGsl
M17		Koniec podprogramu		+	+	PGsl
M19		Pozycjonowanie wrzeciona na pozycję wpisaną w SD43240		+	+	PGsl
M30		Koniec programu, program główny (jak M2)		+	+	PGsl
M40		Automatyczne przełączanie przekładni		+	+	PGsl
M41 ... M45		Stopień przekładni 1 ... 5		+	+	PGsl
M70		Przejście wrzeciona na pracę jako oś		+	+	PGsl
MASLDEF	P	Zdefiniowanie zespołu osi master/slave		+	+	PGAsl
MASLDEL	P	Rozłączenie zespołu osi master/slave i kasowanie definicji zespołu		+	+	PGAsl
MASLOF	P	Wyłącza tymczasowe sprzężenie		+	+	PGAsl
MASLOFS	P	Wyłącza tymczasowe sprzężenie poprzez automatyczne zatrzymanie osi Slave		+	+	PGAsl
MASLON	P	Włącza tymczasowe sprzężenie		+	+	PGAsl
MATCH	F	Szukanie znaków w łańcuchu znaków		+	-	PGAsl
MAXVAL	F	Większa wartość z dwóch zmiennych (funkcja arytmetyczna)		+	+	PGAsl
MCALL	K	Modalne wywołanie podprogramu		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
MEAC	K	Ciągły pomiar w osi bez kasowania pozostałej drogi	s	+	+	PGAsl
MEAFRAME	F	Obliczenie Frame z punktów pomiarowych		+	-	PGAsl
MEAS	A	Pomiar z kasowaniem pozostałej drogi	s	+		PGAsl
MEASA	K	Pomiar w osi z kasowaniem pozostałej drogi	s	+	+	PGAsl
MEASURE	F	Metoda obliczania przy pomiarze obrabianego przedmiotu i narzędzia		+	-	FB1sl (M5)
MEAW	A	Pomiar bez kasowania pozostałej drogi	s	+		PGAsl
MEAWA	K	Pomiar osiowy bez kasowania pozostałej drogi	s	+	+	PGAsl
MI	K	Dostęp do danych Frame: lustrzane odbicie		+		PGAsl
MINDEX	F	Określenie indeksu znaku w wejściowym łańcuchu znaków		+	-	PGAsl
MINVAL	F	Mniejsza wartość z dwóch zmiennych (funkcja arytmetyczna)		+	+	PGAsl
MIRROR	G	Programowane odbicie lustrzane	s	+		PGAsl
MMC	P	Interaktywne wywołanie z programu obróbki okna dialogowego na HMI		+	-	PGAsl
MOD	K	Dzielenie modulo		+		PGAsl
MODAXVAL	F	Określenie pozycji modulo osi obrotowej modulo		+	-	PGAsl
MOV	K	Uruchomienie osi pozycjonowania		-	+	FBSYsl
MOVT	A	Podaje punkt zakończenia przejazdu w kierunku narzędzia				FB1(K2)
MSG	P	Komunikaty programowane	m	+	-	PGsl
MVTOOL	P	Polecenie językowe do wykonania ruchu narzędzia		+	-	FBWsl
N	A	Numer bloku pomocniczego NC		+		PGsl
NAMETOINT	F	Określenie indeksu zmiennych systemowych		+		PGAsl
NC	K	Specyfikacja zakresu obowiązywania danych		+		PGAsl
NEWCONF	P	Przejęcie zmienionych danych maszynowych (odpowiada „Aktywacji działania danej maszynowej")		+	-	PGAsl
NEWMT	F	Utworzenie nowego Multitool		+	-	FBWsl
NEWT	F	Utworzenie nowego narzędzia		+	-	FBWsl
NORM [6]	G	Normalne ustawienie w początkowym, końcowym punkcie przy korekcji narzędzia	m	+		PGsl
NOT	K	Logiczne NIE (negacja)		+		PGAsl
NPROT	P	Specyficzny dla maszyny obszar ochrony WŁ./WYŁ.		+	-	PGAsl
NPROTDEF	P	Definicja obszaru ochrony specyficznego dla maszyny		+	-	PGAsl
NUMBER	F	Zamiana wejściowego łańcucha znaków na liczbę		+	-	PGAsl
OEMIPO1	G	Interpolacja OEM 1	m	+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
OEMIPO2	G	Interpolacja OEM 2	m	+		PGAsl
OF	K	Słowo kluczowe w rozgałęzieniu CASE		+		PGAsl
OFFN	A	Naddatek do zaprogramowanego konturu	m	+		PGsl
OMA1	A	Adres OEM 1	m	+		PGAsl
OMA2	A	Adres OEM 2	m	+		PGAsl
OMA3	A	Adres OEM 3	m	+		PGAsl
OMA4	A	Adres OEM 4	m	+		PGAsl
OMA5	A	Adres OEM 5	m	+		PGAsl
OR	K	Operator logiczny, logiczne połączenie OR		+		PGAsl
ORIAXES	G	Interpolacja liniowa osi maszyny albo osi orientacji	m	+		PGAsl
ORIAXPOS	G	Kąt orientacji za pomocą wirtualnych osi orientacji z pozycją osi obrotowych	m	+		PGAsl
ORIC [6]	G	Zmiany orientacji na narożach zewnętrznych są nakładane na wstawiony blok okręgu	m	+		PGAsl
ORICONCCW	G	Interpolacja na kołowej pobocznicy stożka przeciwnie do ruchu wskazówek zegara	m	+		PGAsl/FB3sl (F3)
ORICONCW	G	Interpolacja na kołowej pobocznicy stożka w kierunku ruchu wskazówek zegara	m	+		PGAsl/FB3sl (F4)
ORICONIO	G	Interpolacja na kołowej pobocznicy stożka z podaniem orientacji pośredniej	m	+		PGAsl/FB3sl (F4)
ORICONTO	G	Interpolacja na kołowej pobocznicy stożka z przejściem stycznym (podanie orientacji końcowej)	m	+		PGAsl/FB3sl (F5)
ORICURVE	G	Interpolacja orientacji z zadaniem ruchu poprzez dwa punkty styku narzędzia	m	+		PGAsl/FB3sl (F6)
ORID	G	Zmiany orientacji są wykonywane przed blokiem okręgu	m	+		PGAsl
ORIEULER [6]	G	Kąt orientacji określany przez kąt Eulera	m	+		PGAsl
ORIMKS	G	Orientacja narzędzia w układzie współrzędnych maszyny	m	+		PGAsl
ORIPATH	G	Orientacja narzędzia odniesiona do toru	m	+		PGAsl
ORIPATHS	G	Orientacja narzędzia w odniesieniu do toru, załamanie w przebiegu orientacji jest wygładzane	m	+		PGAsl
ORIPLANE	G	Interpolacja w płaszczyźnie (odpowiada ORIVECT) interpolacja dużego okręgu	m	+		PGAsl
ORIRESET	P	Położenie podstawowe orientacji narzędzia z maks. 3 osiami orientacji		+	-	PGAsl
ORIROTA [6]	G	Kąt obrotu w kierunku obrotu zadanego absolutnie	m	+		PGAsl
ORIROTC	G	Styczny wektor obrotu względem stycznej do toru	m	+		PGAsl
ORIROTR	G	Kąt obrotu względem płaszczyzny określonej orientacją początkową i końcową	m	+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
ORIROTT	G	Kąt obrotu względem zmiany wektora orientacji	m	+		PGAsl
ORIRPY	G	Kąt orientacji za pomocą kąta RPY (XYZ)	m	+		PGAsl
ORIRPY2	G	Kąt orientacji za pomocą kąta RPY (ZYX)	m	+		PGAsl
ORIS	A	Zmiana orientacji	m	+		PGAsl
ORISOF [6]	G	Wygładzanie przebiegu orientacji WYŁ.	m	+		PGAsl
ORISOLH	F	Oblicza orientację		+		PGAsl
ORISON	G	Wygładzanie przebiegu orientacji WŁ.	m	+		PGAsl
ORIVECT [6]	G	Interpolacja dużego okręgu (identyczna z ORIPLANE)	m	+		PGAsl
ORIVIRT1	G	Kąt orientacji za pomocą wirtualnych osi orientacji (definicja 1)	m	+		PGAsl
ORIVIRT2	G	Kąt orientacji za pomocą wirtualnych osi orientacji (definicja 1)	m	+		PGAsl
ORIWKS [6]	G	Orientacja narzędzia w układzie współrzędnych przedmiotu obrabianego	m	+		PGAsl
OS	K	Ruch wahadłowy wł./wył.		+		PGAsl
OSB	K	Ruch wahadłowy: punkt startowy	m	+		FB1sl (P5)
OSC	G	Stałe wygładzanie orientacji narzędzia	m	+		PGAsl
OSCILL	K	Oś: 1 - 3 osi dosuwu	m	+		PGAsl
OSCTRL	K	Opcje ruchu wahadłowego	m	+		PGAsl
OSD	G	Wygładzanie orientacji narzędzia przez zadanie drogi wygładzania za pomocą SD	m	+		PGAsl
OSE	K	Ruch wahadłowy, punkt końcowy	m	+		PGAsl
OSNSC	K	Ruch wahadłowy: liczba wyiskrzeń	m	+		PGAsl
OSOF [6]	G	Wygładzanie orientacji narzędzia WYŁ.	m	+		PGAsl
OSP1	K	Ruch wahadłowy: lewy punkt nawrotu	m	+		PGAsl
OSP2	K	Ruch wahadłowy: prawy punkt nawrotu	m	+		PGAsl
OSS	G	Wygładzanie orientacji narzędzia na końcu bloku	m	+		PGAsl
OSSE	G	Wygładzanie orientacji narzędzia na początku i końcu bloku	m	+		PGAsl
OST	G	Wygładzanie orientacji narzędzia przez zadanie tolerancji kątowej w stopniach za pomocą SD (maksymalne odchylenie od zaprogramowanego przebiegu orientacji)	m	+		PGAsl
OST1	K	Ruch wahadłowy: punkt zatrzymania w lewym punkcie nawrotu	m	+		PGAsl
OST2	K	Ruch wahadłowy: punkt zatrzymania w prawym punkcie nawrotu	m	+		PGAsl
OTOL	A	Tolerancja orientacji dla funkcji kompresora, wygładzanie orientacji i rodzaje wygładzania	m	+		PGAsl
OVR	K	Korekcja prędkości obrotowej	m	+		PGAsl
OVRA	K	Osiowa korekcja prędkości obrotowej	m	+	+	PGAsl
OVRRAP	K	Korekcja posuwu szybkiego	m	+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
P	A	Liczba powtórzeń przebiegów podprogramu		+		PGAsl
PAROT	G	Zorientowanie układu współrzędnych przedmiotu obrabianego na przedmiocie obrabianym	m	+		PGsl
PAROTOF [6]	G	Wyłącza obrót Frame związany z obrabianym przedmiotem	m	+		PGsl
PCALL	K	Wywołanie podprogramów z absolutnym podaniem ścieżki i przekazaniem parametrów		+		PGAsl
PDELAYOF	G	Zwłoka przy „Tłoczenie WYŁ."	m	+		PGAsl
PDELAYON [6]	G	Zwłoka przy „Tłoczenie WŁ."	m	+		PGAsl
PHI	K	Kąt obrotu orientacji wokół osi kierunkowej stożka		+		PGAsl
PHU	K	Fizyczna jednostka zmiennej		+		PGAsl
PL	A	1. B-Spline: odstęp węzłów 2. Interpolacja wielomianowa: długość przedziału parametrów przy interpolacji wielomianowej	s	+		PGAsl
PM	K	Na minutę		+		PGsl
PO	K	Współczynnik wielomianu przy interpolacji wielomianowej	s	+		PGAsl
POCKET3	C (T)	Frezowanie kieszeni prostokątnej		+		PGAsl
POCKET4	C (T)	Frezowanie kieszeni kołowej		+		PGAsl
POLF	K	Pozycja wycofania LIFTFAST	m	+		PGsl/PGAsl
POLFA	P	Uruchamia pozycję wycofania dla pojedynczych osi przez $AA_ESR_TRIGGER	m	+	+	PGsl
POLFMASK	P	Udostępnia oś dla wycofania bez sprzężenia między osiami	m	+	-	PGsl
POLFMLIN	P	Udostępnia oś dla wycofania z zależnością liniową między osiami	m	+	-	PGsl
POLY	G	Interpolacja wielomianowa	m	+		PGAsl
POLYPATH	P	Interpolacja wielomianowa wybierana dla grup osi AXIS lub VECT	m	+	-	PGAsl
PON	G	Tłoczenie WŁ.	m	+		PGAsl
PONS	G	Tłoczenie WŁ. w takcie IPO	m	+		PGAsl
POS	K	Pozycjonowanie osi		+	+	PGsl
POSA	K	Pozycjonowanie osi poza granice bloku		+	+	PGsl
POSM	P	Pozycjonowanie magazynu		+	-	FBWsl
POSMT	P	Pozycjonowanie Multitool na uchwyt narzędziowy z numerem miejsca		+	-	FBWsl
POSP	K	Pozycjonowanie w częściach (ruch przerywany)		+		PGsl
POSRANGE	F	Określenie, czy aktualnie interpolowana pozycja zadana osi znajduje się w oknie wokół zadanej pozycji odniesienia		+	+	FBSYsl
POT	F	Kwadrat (funkcja arytmetyczna)		+	+	PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
PR	K	Na obrót		+		PGsl
PREPRO	PA	Oznakowanie podprogramów z przygotowaniem		+		PGAsl
PRESETON	P	Ustawienie wartości rzeczywistej ze stratą statusu referencyjnego		+	+	PGAsl
PRESETON	P	Ustawienie wartości rzeczywistej bez straty statusu referencyjnego		+	+	PGAsl
PRIO	K	Słowo kluczowe do ustawienia priorytetu przy wykonywaniu przerwań		+		PGAsl
PRLOC	K	Inicjalizacja zmiennej przy resecie tylko po zmianie lokalnej		+		PGAsl
PROC	K	Pierwsze polecenie podprogramu parametrycznego		+		PGAsl
PROTA	P	Żądanie nowego obliczenia modelu kolizji		+		PGAsl
PROTD	F	Obliczenie odstępu drugiego obszaru ochrony		+		PGAsl
PROTS	P	Ustawienie stanu obszaru ochrony		+		PGAsl
PSI	K	Kąt rozwarcia stożka		+		PGAsl
PTP	G	Ruch z punktu do punktu (ruch PTP)	m	+		PGAsl
PTPG0	G	Ruch punkt do punktu tylko przy G0, w przeciwnym razie ruch po torze CP	m	+		PGAsl
PTPWOC	G	Ruch z punktu do punktu bez ruchów wyrównawczych, które spowodowane są zmianami orientacji	m	+		PGAsl
PUNCHACC	P	Zależne od drogi przyspieszenie przy cięciu		+	-	PGAsl
PUTFTOC	P	Korekcja dokładna narzędzia dla obciągania równoległego		+	-	PGAsl
PUTFTOCF	P	Korekcja dokładna narzędzia w zależności od ustalonej przy pomocy FCTDEF funkcji obciągania równoległego		+	-	PGAsl
PW	A	B-Spline, waga punktu	s	+		PGAsl
QU	K	Szybkie wyprowadzenie funkcji dodatkowej (pomocniczej)		+		PGsl
R...	A	Parametr obliczeniowy również jako ustawiany identyfikator adresu z rozszerzeniem numerycznym		+		PGAsl
RAC	K	Absolutne, pojedynczymi blokami, specyficzne dla osi programowanie po promieniu	s	+		PGsl
RDISABLE	P	Blokada wczytywania		-	+	FBSYsl
READ	P	Wczytuje w podanym pliku jeden lub wiele wierszy i zapisuje w tablicy przeczytane informacje		+	-	PGAsl
REAL	K	Typ danych: Liczba zmiennoprzecinkowa ze znakiem (liczby rzeczywiste)		+		PGAsl
REDEF	K	Redefinicja zmiennych systemowych, zmiennych użytkownika i poleceń językowych NC		+		PGAsl
RELEASE	P	Udostępnienie osi maszyny do zamiany osi		+	+	PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
REP	K	Słowo kluczowe do inicjalizacji wszystkich elementów tablicy o tej samej wartości		+		PGAsl
REPEAT	K	Powtórzenie pętli programowej		+		PGAsl
REPEATB	K	Powtórzenie wiersza programu		+		PGAsl
REPOSA	G	Ponowny dojazd do konturu liniowo we wszystkich osiach	s	+		PGAsl
REPOSH	G	Ponowny dojazd do konturu po półokręgu	s	+		PGAsl
REPOSHA	G	Ponowny dojazd do konturu we wszystkich osiach; osie geometryczne po półokręgu	s	+		PGAsl
REPOSL	G	Ponowny dojazd do konturu liniowo	s	+		PGAsl
REPOSQ	G	Ponowny dojazd do konturu po ćwierćokręgu	s	+		PGAsl
REPOSQA	G	Ponowny dojazd do konturu liniowo we wszystkich osiach; osie geometryczne po ćwierćokręgu	s	+		PGAsl
RESETMON	P	Polecenie językowe do uaktywnienia wartości zadanej		+	-	FBWsl
RET	P	Koniec podprogramu		+	+	PGAsl
RETB	P	Koniec podprogramu		+	+	PGAsl
RIC	K	Względnie pojedynczymi blokami, specyficzne dla osi programowanie na promieniu	s	+		PGsl
RINDEX	F	Określenie indeksu znaku w wejściowym łańcuchu znaków		+	-	PGAsl
RMB	G	Ponowny dojazd do punktu początkowego bloku	m	+		PGAsl
RMBBL	G	Ponowny dojazd do punktu początkowego bloku	s	+		PGAsl
RME	G	Ponowny dojazd do punktu końcowego bloku	m	+		PGAsl
RMEBL	G	Ponowny dojazd do punktu końcowego bloku	s	+		PGAsl
RMI [6]	G	Ponowny dojazd do punktu przerwania	m	+		PGAsl
RMIBL [6]	G	Ponowny dojazd do punktu przerwania	s	+		PGAsl
RMN	G	Ponowny dojazd do najbliżej położonego punktu toru	m	+		PGAsl
RMNBL	G	Ponowny dojazd do najbliżej położonego punktu toru	s	+		PGAsl
RND	A	Zaokrąglenie naroża konturu	s	+		PGsl
RNDM	A	Zaokrąglenie modalne	m	+		PGsl
ROT	G	Programowany obrót	s	+		PGsl
ROTS	G	Programowane obroty Frame z kątami przestrzennymi	s	+		PGsl
ROUND	F	Zaokrąglanie miejsc po przecinku		+	+	PGAsl
ROUNDUP	F	Zaokrąglenie wprowadzonej wartości do góry		+	+	PGAsl
RP	A	Promień biegunowy	m/s	+		PGsl
RPL	A	Obrót w płaszczyźnie	s	+		PGsl
RT	K	Parametry dostępu do danych Frame: obrót		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
RTLIOF	G	G0 bez interpolacji liniowej (interpolacja pojedynczych osi)	m	+		PGsl
RTLION [6]	G	G0 z interpolacją liniową	m	+		PGsl

Polecenia S ... Z

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
S	A	Obroty wrzeciona (przy G4, G96/G961 inne znaczenie)	m/s	+	+	PGsl
SAVE	PA	Atrybut do zapamiętania funkcji G i Frame przy wywołaniu podprogramu		+		PGAsl
SBLOF	P	Blokowanie wykonywania pojedynczymi blokami		+	-	PGAsl
SBLON	P	Wyłączenie blokady wykonywania pojedynczymi blokami		+	-	PGAsl
SC	K	Parametry dostępu do danych Frame: Skalowanie		+		PGAsl
SCALE	G	Programowane skalowanie	s	+		PGsl
SCC	K	Selektywne przypisanie osi poprzecznej do G96/G961/G962. Identyfikatorami osi mogą być osie geometryczne, kanałowe lub maszyny.		+		PGsl
SCPARA	K	Programowanie zestawu parametrów serwo		+	+	PGAsl
SD	A	Stopień Spline	s	+		PGAsl
SET	K	Słowo kluczowe do inicjalizacji wszystkich elementów tablicy z wyszczególnionymi wartościami		+		PGAsl
SETAL	P	Ustawienie alarmu		+	+	PGAsl
SETDNO	F	Przyporządkowanie numeru D do ostrza (CE) narzędzia (T)		+	-	PGAsl
SETINT	K	Ustalenie, która procedura przerwania ma zostać uaktywniona, gdy jest aktywne wejście NC		+		PGAsl
SETM	P	Ustawienie znaczników synchronizacji we własnym kanale		+	+	PGAsl
SETMS	P	Przełączenie z powrotem na wrzeciono Master określone w danej maszynowej		+	-	PGsl
SETMS(n)	P	Wrzeciono n ma obowiązywać, jako wrzeciono Master		+		PGsl
SETMTH	P	Ustawienie numeru uchwytu narzędzia Master		+	-	FBWsl
SETPIECE	P	Uwzględnienie liczby sztuk dla wszystkich narzędzi, które są przyporządkowane do wrzeciona		+	-	FBWsl
SETTA	P	Ustawienie aktywności narzędzia z grupy zużycia		+	-	FBWsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
SETTCOR	F	Zmiana składowych narzędzia przy uwzględnieniu wszystkich ograniczeń		+	-	PGAsl
SETTIA	P	Wyłączenie aktywności narzędzia z grupy zużycia		+	-	FBWsl
SF	A	Przesunięcie punktu początkowego przy nacinaniu gwintu	m	+		PGsl
SIN	F	Sinus (funkcja trygonometryczna)		+	+	PGAsl
SIRELAY	F	Aktywacja funkcji bezpieczeństwa sparametryzowanych za pomocą SIRELIN, SIRELOUT i SIRELTIME.		-	+	FBSIsl
SIRELIN	P	Inicjalizacja wielkości wejściowych bloku funkcyjnego		+	-	FBSIsl
SIRELOUT	P	Inicjalizacja wielkości wyjściowych bloku funkcyjnego		+	-	FBSIsl
SIRELTIME	P	Inicjalizacja zegara bloku funkcyjnego		+	-	FBSIsl
SLOT1	C (T)	Rowek wzdłużny		+		PGAsl
SLOT2	C (T)	Rowek kołowy		+		PGAsl
SOFT	G	Przyspieszenie ruchu po torze z ograniczonym szarpnięciem	m	+		PGAsl
SOFTA	P	Włączenie przyspieszenia osi programowanych z ograniczonym szarpnięciem		+	-	PGAsl
SON	G	Cięcie WŁ.	m	+		PGAsl
SONS	G	Cięcie w takcie IPO WŁ.	m	+		PGAsl
SPATH [6]	G	Odniesieniem toru dla osi FGROUP jest długość łuku	m	+		PGAsl
SPCOF	P	Przełączenie wrzeciona Master lub wrzeciona (n) z regulacji położenia na regulację obrotów	m	+	-	PGsl
SPCON	P	Przełączenie wrzeciona Master lub wrzeciona (n) z regulacji obrotów na regulację położenia	m	+	-	PGAsl
SPI	F	Konwertuje numer wrzeciona na identyfikator osi		+	-	PGAsl
SPIF1 [6]	G	Szybkie wejścia/wyjścia NC dla tłoczenia/cięcia, Bajt 1	m	+		FB2sl (N4)
SPIF2	G	Szybkie wejścia/wyjścia NC dla tłoczenia/cięcia, Bajt 2	m	+		FB2sl (N4)
SPLINEPATH	P	Ustalenie zespołu Spline		+	-	PGAsl
SPN	A	Liczba odcinków częściowych na blok	s	+		PGAsl
SPOF [6]	G	Skok WYŁ., tłoczenia, cięcie, WYŁ	m	+		PGAsl
SPOS	K	Pozycja wrzeciona	m	+	+	PGsl
SPOSA	K	Pozycja wrzeciona poza granicami bloku	m	+		PGsl
SPP	A	Długość odcinka częściowego	m	+		PGAsl
SPRINT	F	Zwraca sformatowany wejściowy łańcuch znaków		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
SQRT	F	Pierwiastek kwadratowy (funkcja arytmetyczna) (square root)		+	+	PGAsl
SR	A	Droga wycofania ruchu oscylacyjnego dla akcji synchronicznej	s	+		PGsl
SRA	K	Droga wycofania ruchu oscylacyjnego przy zewnętrznym wejściu osi dla akcji synchronicznej	m	+		PGsl
ST	A	Czas wyiskrzania ruchu oscylacyjnego dla akcji synchronicznej	s	+		PGsl
STA	K	Osiowy czas wyiskrzania ruchu oscylacyjnego dla akcji synchronicznej	m	+		PGsl
START	P	Uruchomienie wybranych programów w kilku kanałach jednocześnie z bieżącego programu		+	-	PGAsl
STARTFIFO [6]	G	Wykonywanie; równolegle do przebiegu wypełnienia bufora przebiegu wyprzedzającego	m	+		PGAsl
STAT		Ustawienie przegubów	s	+		PGAsl
STOLF	K	Współczynnik tolerancji G0	m	+		PGAsl
STOPFIFO	G	Wstrzymanie wykonywania programu; wypełnienie bufora przebiegu wyprzedzającego, aż nastąpi STARTFIFO, bufor przebiegu wyprzedzającego zostanie wypełniony albo nastąpi koniec programu	m	+		PGAsl
STOPRE	P	Zatrzymanie przebiegu wyprzedzającego, aż wszystkie przygotowane bloki przebiegu głównego będą wykonane		+	-	PGAsl
STOPREOF	P	Anulowanie zatrzymania przebiegu wyprzedzającego		-	+	FBSYsl
STRING	K	Typ danych: Łańcuch znaków		+		PGAsl
STRINGIS	F	Sprawdza istniejący zakres języka NC, a szczególnie należące do niego polecenia, nazwy cykli NC, zmienne użytkownika, makra i nazwy etykiet: czy istnieją, czy są ważne, zdefiniowane lub aktywne.		+	-	PGAsl
STRLEN	F	Określenie długości łańcucha znaków		+	-	PGAsl
SUBSTR	F	Określenie indeksu znaku w wejściowym łańcuchu znaków		+	-	PGAsl
SUPA	G	Blokowanie aktualnego przesunięcia punktu zerowego, łącznie z przesunięciami programowanymi, Frame systemowymi, przesunięciami kółkiem ręcznym (DRF), zewnętrznym przesunięciem punktu zerowego i nałożonym ruchem	s	+		PGsl
SVC	K	Prędkość skrawania narzędzia	m	+		PGsl
SYNFCT	P	Ewaluacja wielomianu zależnie od warunku w akcji synchronicznej ruchu		-	+	FBSYsl
SYNR	K	Odczyt zmiennej następuje synchronicznie, tzn. w chwili wykonywania		+		PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
SYNRW	K	Odczyt i zapis zmiennej następuje synchronicznie, tzn. w chwili wykonywania		+		PGAsl
SYNW	K	Zapis zmiennej następuje synchronicznie, tzn. w chwili wykonywania		+		PGAsl
T	A	Wywołanie narzędzia (zmiana tylko wtedy, gdy ustalono to w danej maszynowej; poza tym konieczne jest polecenie M6)		+		PGsl
TAN	F	Tangens (funkcja trygonometryczna)		+	+	PGAsl
TANG	P	Sterowanie styczne: Definiowanie sprzężenia		+	-	PGAsl
TANGDEL	P	Sterowanie styczne: Kasowanie sprzężenia		+	-	PGAsl
TANGOF	P	Sterowanie styczne: Wyłączenie sprzężenia		+	-	PGAsl
TANGON	P	Sterowanie styczne: Włączenie sprzężenia		+	-	PGAsl
TCA (828D: _TCA)	P	Wybór narzędzia / Wymiana narzędzia niezależnie od jego statusu		+	-	FBWsl
TCARR	A	Zażądanie nośnika narzędzi (numer "m")		+		PGAsl
TCI	P	Wymień narzędzie z bufora do magazynu		+	-	FBWsl
TCOABS [6]	G	Określa składowe długości narzędzia na podstawie aktualnej orientacji narzędzia	m	+		PGAsl
TCOFR	G	Określa składowe długości narzędzia na podstawie zorientowania aktywnego Frame	m	+		PGAsl
TCOFRX	G	Określa orientację narzędzia na podstawie aktywnego Frame przy wyborze narzędzia, narzędzie skierowane w kierunku X	m	+		PGAsl
TCOFRY	G	Określa orientację narzędzia na podstawie aktywnego Frame przy wyborze narzędzia, narzędzie skierowane w kierunku Y	m	+		PGAsl
TCOFRZ	G	Określa orientację narzędzia na podstawie aktywnego Frame przy wyborze narzędzia, narzędzie skierowane w kierunku Z	m	+		PGAsl
THETA	A	Kąt obrotu	s	+		PGAsl
TILT	A	Kąt boczny	m	+		PGAsl
TLIFT	P	Sterowanie styczne: Włączenie tworzenia bloków pośrednich		+	-	PGAsl
TML	P	Wybór narzędzia z magazynu - numer miejsca		+	-	FBWsl
TMOF	P	Cofnięcie wyboru nadzoru narzędzia		+	-	PGAsl
TMON	P	Wybór nadzoru narzędzia		+	-	PGAsl
TO	K	Określa wartość końcową w pętli FOR		+		PGAsl
TOFF	K	Przesunięcie długości narzędzia w kierunku składowej długości narzędzia, działającego równolegle do osi geometrycznej podanej w indeksie	m	+		PGsl
TOFFL	K	Przesunięcie długości narzędzia w kierunku składowej długości narzędzia L1, L2 lub L3	m	+		PGsl
TOFFOF	P	Wyłączenie korekcji długości narzędzia online		+	-	PGAsl
TOFFON	P	Aktywacja korekcji długości narzędzia online		+	-	PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
TOFFR	A	Przesunięcie promienia narzędzia	m	+		PGsl
TOFRAME	G	Ustawienie osi Z przez obrót Frame w aktualnym WKS równolegle do orientacji narzędzia	m	+		PGsl
TOFRAMEX	G	Ustawienie osi X przez obrót Frame w aktualnym WKS równolegle do orientacji narzędzia	m	+		PGsl
TOFRAMEY	G	Ustawienie osi Y przez obrót Frame w aktualnym WKS równolegle do orientacji narzędzia	m	+		PGsl
TOFRAMEZ	G	Jak TOFRAME	m	+		PGsl
TOLOWER	F	Zamienia litery łańcucha znaków na małe litery		+	-	PGAsl
TOOLENV	F	Zapisuje aktualne stany, które mają znaczenie dla interpretacji danych narzędzia zapisanych w pamięci		+	-	PGAsl
TOOLGNT	F	Określa liczbę narzędzi w grupie		+	-	FBWsl
TOOLGT	F	Określa numer T narzędzia z grupy		+	-	FBWsl
TOROT	G	Ustawienie osi Z przez obrót Frame w aktualnym WKS równolegle do orientacji narzędzia	m	+		PGsl
TOROTOF [6]	G	Obroty Frame w kierunku narzędzia WYŁ.	m	+		PGsl
TOROTX	G	Ustawienie osi X przez obrót Frame w aktualnym WKS równolegle do orientacji narzędzia	m	+		PGsl
TOROTY	G	Ustawienie osi Y przez obrót Frame w aktualnym WKS równolegle do orientacji narzędzia	m	+		PGsl
TOROTZ	G	Jak TOROT	m	+		PGsl
TOUPPER	F	Zamienia litery łańcucha znaków na duże litery		+	-	PGAsl
TOWBCS	G	Wartości zużycia w bazowym układzie współrzędnych (BKS)	m	+		PGAsl
TOWKCS	G	Wartości zużycia w układzie współrzędnych głowicy narzędziowej przy transformacji kinematycznej (różni się od MKS przez obrót narzędzia)	m	+		PGAsl
TOWMCS	G	Wartości zużycia w układzie współrzędnych maszyny (MKS)	m	+		PGAsl
TOWSTD [6]	G	Wartość wyjściowa dla korekcji długości narzędzia	m	+		PGAsl
TOWTCS	G	Wartości zużycia w układzie współrzędnych narzędzia (punkt odniesienia uchwytu narzędzia T w mocowaniu oprawki narzędziowej)	m	+		PGAsl
TOWWCS	G	Wartości zużycia w układzie współrzędnych przedmiotu obrabianego (WKS)	m	+		PGAsl
TR	K	Składowa przesunięcia zmiennej Frame		+		PGAsl
TRAANG	P	Transformacja osi skośnej		+	-	PGAsl
TRACON	P	Transformacja kaskadowa		+	-	PGAsl
TRACYL	P	Walec: Transformacja pobocznicy		+	-	PGAsl
TRAFOOF	P	Wyłącza transformacje aktywne w kanale		+	-	PGAsl
TRAFOON	P	Włącza transformacje za pomocą łańcuchów kinematycznych		+	-	PGAsl
TRAILOF	P	Holowanie asynchroniczne WYŁ.		+	+	PGAsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
TRAILON	P	Holowanie asynchroniczne WŁ.		+	+	PGAsl
TRANS	G	Absolutne programowalne przesunięcie punktu zerowego	s	+		PGsl
TRANSMIT	P	Transformacja biegunowa (obróbka powierzchni czołowej)		+	-	PGAsl
TRAORI	P	Transformacja 4-, 5-osiowa, transformacja rodzajowa		+	-	PGAsl
TRUE	K	Stała logiczna: Prawda		+		PGAsl
TRUNC	F	Obcięcie miejsc po przecinku		+	+	PGAsl
TU		Kąt osi	s	+		PGAsl
TURN	A	Liczba zwojów linii śrubowej	s	+		PGsl
ULI	K	Górna wartość graniczna zmiennych		+		PGAsl
UNLOCK	P	Zezwolenie dla akcji synchronicznej z ID (kontynuowanie cyklu technologicznego)		-	+	FBSYsl
UNTIL	K	Warunek zakończenia pętli REPEAT		+		PGAsl
UPATH	G	Odniesieniem toru dla osi FGROUP jest parametr krzywej	m	+		PGAsl
VAR	K	Słowo kluczowe: Sposób przekazania parametrów		+		PGAsl
VELOLIM	K	Zmniejszenie maksymalnej prędkości osiowej	m	+		PGAsl
VELOLIMA	K	Zmniejszenie albo zwiększenie maksymalnej prędkości osiowej w osi nadążnej	m	+	+	PGAsl
WAITC	P	Czekanie, aż kryterium zmiany bloku sprzężenia dla osi/wrzecion będzie spełnione		+	-	PGAsl
WAITE	P	Czekanie na koniec programu w innym kanale		+	-	PGAsl
WAITENC	P	Czekanie na synchronizowane lub odtworzone pozycje osi		+	-	PGAsl
WAITM	P	Czekanie na znacznik w podanym kanale; zakończenie poprzedniego bloku z zatrzymaniem dokładnym.		+	-	PGAsl
WAITMC	P	Czekanie na znacznik w podanym kanale; zatrzymanie dokładne tylko wtedy, gdy inne kanały jeszcze nie doszły do znacznika.		+	-	PGAsl
WAITP	P	Czekanie na koniec ruchu osi pozycjonowania		+	-	PGsl
WAITS	P	Czekanie na osiągnięcie pozycji wrzeciona		+	-	PGsl
WALCS0 [6]	G	Wyłączenie ograniczenia obszaru pracy WKS	m	+	-	PGsl
WALCS1	G	Aktywuje grupę ograniczenia pola obróbki WKS 1	m	+	-	PGsl
WALCS2	G	Aktywuje grupę ograniczenia pola obróbki WKS 2	m	+	-	PGsl
WALCS3	G	Aktywuje grupę ograniczenia pola obróbki WKS 3	m	+	-	PGsl
WALCS4	G	Aktywuje grupę ograniczenia pola obróbki WKS 4	m	+	-	PGsl
WALCS5	G	Aktywuje grupę ograniczenia pola obróbki WKS 5	m	+	-	PGsl

Polecenie	Typ [1]	Znaczenie	W [2]	TP [3]	SA [4]	Patrz opis [5]
[1] [2] [3] [4] [5] Objaśnienia patrz legenda (Strona 895).						
WALCS6	G	Aktywuje grupę ograniczenia pola obróbki WKS 6	m	+	-	PGsl
WALCS7	G	Aktywuje grupę ograniczenia pola obróbki WKS 7	m	+	-	PGsl
WALCS8	G	Aktywuje grupę ograniczenia pola obróbki WKS 8	m	+	-	PGsl
WALCS9	G	Aktywuje grupę ograniczenia pola obróbki WKS 9	m	+	-	PGsl
WALCS10	G	Aktywuje grupę ograniczenia pola obróbki WKS 10	m	+	-	PGsl
WALIMOF	G	Ograniczenie pola obróbki BKS WYŁ.	m	+	-	PGsl
WALIMON [6]	G	Ograniczenie pola obróbki BKS WŁ.	m	+	-	PGsl
WHEN	K	Akcja jest wykonywana jeden raz, gdy warunek zostanie raz spełniony		-	+	FBSYsl
WHENEVER	K	Akcja jest wykonywana cyklicznie, gdy warunek jest spełniony		-	+	FBSYsl
WHILE	K	Początek pętli programowej WHILE		+		PGAsl
WRITE	P	Zapisanie tekstu do systemu plików. Wstawia blok na końcu podanego pliku.		+	-	PGAsl
WRTPR	P	Zapisuje łańcuch znaków w zmiennej BTSS		+	-	PGsl
X	A	Nazwa osi	m/s	+	+	PGsl
XOR	O	Wyłączne logiczne OR		+		PGAsl
Y	A	Nazwa osi	m/s	+	+	PGsl
Z	A	Nazwa osi	m/s	+	+	PGsl

Legenda

[1] Rodzaj polecenia:

A Adres

Identyfikator, któremu jest przyporządkowywana wartość (np. OVR=10). Jest również kilka adresów, które włączają lub wyłączają funkcję bez przyporządkowania wartości (np. CPLON i CPLOF).

C (A) Cykl AST

Wstępnie zdefiniowany program NC do automatycznej optymalizacji za pomocą AST (= automatyczna optymalizacja serwo). Dopasowanie do konkretnej sytuacji optymalizacji odbywa się za pomocą parametrów, które są przekazywane do cyklu podczas jego wywołania.

C (M) Cykl pomiarowy

Wstępnie zdefiniowany program NC, w którym zaprogramowany jest w odpowiedni sposób określony proces pomiaru, jak np. określenie średnicy wewnętrznej cylindrycznego przedmiotu obrabianego. Dopasowanie do konkretnej sytuacji pomiarowej odbywa się za pomocą parametrów, które są przekazywane do cyklu podczas jego wywołania.

C (T) Cykl technologiczny

Wstępnie zdefiniowany program NC, w którym zaprogramowany jest w odpowiedni sposób określony cykl, jak np. gwintowanie otworu lub frezowanie kieszeni. Dopasowanie do konkretnego procesu obróbki odbywa się poprzez parametry, które są przekazywane do cyklu podczas jego wywołania.

F Funkcje wstępnie zdefiniowane (wartość zwrotna)

Wywołanie funkcji wstępnie zdefiniowanej może znajdować się w wyrażeniu jako argument.

G Polecenie G

Polecenia G dzieli się na grupy. W jednym bloku można zaprogramować tylko jedno polecenie G z danej grupy. Polecenie G może działać modalnie (do momentu odwołania przez inne polecenie z tej samej grupy) lub może działać tylko w bloku, w którym się znajduje (pojedynczymi blokami).

K Słowo kluczowe

Identyfikator, który określa składnię bloku. Do słowa kluczowego nie jest przyporządkowywana wartość i przy jego pomocy nie można też włączyć/wyłączyć żadnej funkcji NC.

Przykłady: Struktury kontrolne (IF, ELSE, ENDIF, WHEN, ...), przebieg programu (GOTOB, GOTO, RET …)

O Operator

Operator operacji matematycznej, porównania albo logicznej

P Procedura wstępnie zdefiniowana (brak wartość zwrotnej)

PA Atrybut programu

Atrybuty programu są umieszczane na końcu wiersza definicji podprogramu:
`PROC <nazwa programu>(...) <atrybut programu>`

Określają zachowanie się podprogramu podczas wykonywania.

[2] Działanie polecenia:

m Modalnie

s Pojedynczymi blokami

[3] Możliwość zaprogramowania w programie obróbki:

\+ programowane

\- nieprogramowane

M programowane tylko przez producenta maszyny

[4] Możliwość programowania w akcjach synchronicznych:

+	programowane
-	nieprogramowane
T	programowane tylko w cyklach technologicznych

[5] Odesłanie do dokumentacji, która zawiera szczegółowy opis poleceń:

PGs/	Podręcznik programowania, Podstawy
PGAs/	Podręcznik programowania, Przygotowanie do pracy
BNMs/	Podręcznik programowania, Cykle pomiarowe
BHDs/	Podręcznik obsługi, Toczenie
BHFs/	Podręcznik obsługi, Frezowanie
FB1s/ ( )	Opis funkcji, Funkcje podstawowe (z alfanumerycznym skrótem opisu działania w nawiasach)
FB2s/ ( )	Opis funkcji, Funkcje rozszerzone (z alfanumerycznym skrótem opisu działania w nawiasach)
FB3s/ ( )	Opis funkcji, Funkcje specjalne (z alfanumerycznym skrótem opisu działania w nawiasach)
FBSIs/	Opis funkcji Safety Integrated
FBSYs/	Opis funkcji, Akcje synchroniczne
FBWs/	Opis funkcji, Zarządzanie narzędziami

[6] Ustawienie standardowe na początku programu (w stanie przy dostawie sterowania, o ile nie zaprogramowano inaczej).

Rysunek 21-1 Znaczenie przypisów w tabelach zestawień

21.2 Instrukcje: Dostępność w przypadku SINUMERIK 828D

Uwaga

Cykle

Cykle zostały oznaczone jako „opcjonalne”, gdy zależą od następujących opcji wymagających licencji:

- Rozszerzone funkcje technologiczne (numer artykułu: 6FC5800-0AP58-0YB0)
- Cykle pomiarowe (numer artykułu: 6FC5800-0AP28-0YB0)
- Pomiar kinematyki (numer artykułu: 6FC5800-0AP18-0YB0)
- SINUMERIK Szlifowanie Advanced (numer artykułu: 6FC5800-0AS35-0YB0)

Cykle zostały nieoznaczone, gdy zawierają one tylko częściową funkcjonalność na określonych systemach poprzez zastosowanie opcji „Rozszerzone funkcje technologiczne”.

21.2.1 Warianty sterowania - frezowanie / toczenie

Polecenia A ... C

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
:	●	●	●	●	●	●	●	●
*	●	●	●	●	●	●	●	●
+	●	●	●	●	●	●	●	●
-	●	●	●	●	●	●	●	●
<	●	●	●	●	●	●	●	●
<<	●	●	●	●	●	●	●	●
<=	●	●	●	●	●	●	●	●
=	●	●	●	●	●	●	●	●
>=	●	●	●	●	●	●	●	●
/	●	●	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●	●
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-
A5	-	-	-	-	-	-	-	-
A6	-	-	-	-	-	-	-	-

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	-	-	-	-	-	-	Nazwa osi kanału	Nazwa osi kanału
C3	-	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	-
C5	-	-	-	-	-	-	-	-
C6	-	-	-	-	-	-	-	-

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowa-nie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowa-nie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowa-nie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	-	●	-	●	●	-	-
COMPSURF	-	-	○	-	○	○	-	-

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	○	○	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○	○	○

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
CPMRESET	○	○	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○	○	○
CPOF	○	○	○	○	○	○	○	○
CPON	○	○	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●	●	●
CPRES	○	○	○	○	○	○	○	○
CPROT	●	●	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○	○	○
CR	●	●	●	●	●	●	●	●
CROT	●	●	●	●	●	●	●	●
CROTS	●	●	●	●	●	●	●	●
CRPL	●	●	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○	○	○
CT	●	●	●	●	●	●	●	●
CTAB	-	-	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-	-	-
CTABID	-	-	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-	-	-

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
CYCLE62	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowa-nie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowa-nie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowa-nie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
CYCLE899	○	○	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	-	-	-	-	-	-	-	-

Polecenia D ... F

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
D	●	●	●	●	●	●	●	●
D0	●	●	●	●	●	●	●	●
DAC	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
DC	●	●	●	●	●	●	●	●
DCI	●	●	●	●	●	●	●	●
DCM	●	●	●	●	●	●	●	●
DCU	●	●	●	●	●	●	●	●
DEF	●	●	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●	●	●
DELDL	●	●	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●	●	●
DELETE	●	●	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●	●	●
DELMLRES	●	●	●	●	●	●	●	●
DELMT	●	●	●	●	●	●	●	●
DELOBJ	-	-	-	-	-	-	-	-
DELT	●	●	●	●	●	●	●	●
DELTC	●	●	●	●	●	●	●	●
DELTOOLENV	●	●	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●	●	●
DIC	●	●	●	●	●	●	●	●
DILF	●	●	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●	●	●
DISC	●	●	●	●	●	●	●	●
DISCL	●	●	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●	●	●
DISPR	●	●	●	●	●	●	●	●
DISR	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
FPO	-	-	-	-	-	-	-	-

Polecenie	SINUMERIK 828D							
• Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
FPR	•	•	•	•	•	•	•	•
FPRAOF	•	•	•	•	•	•	•	•
FPRAON	•	•	•	•	•	•	•	•
FRAME	•	•	•	•	•	•	•	•
FRC	•	•	•	•	•	•	•	•
FRCM	•	•	•	•	•	•	•	•
FROM	•	•	•	•	•	•	•	•
FTOC	•	•	•	•	•	•	•	•
FTOCOF	•	•	•	•	•	•	•	•
FTOCON	•	•	•	•	•	•	•	•
FXS	•	•	•	•	•	•	•	•
FXST	•	•	•	•	•	•	•	•
FXSW	•	•	•	•	•	•	•	•
FZ	•	•	•	•	•	•	•	•

Polecenia G ... L

Polecenie	SINUMERIK 828D							
• Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
G0	•	•	•	•	•	•	•	•
G1	•	•	•	•	•	•	•	•
G2	•	•	•	•	•	•	•	•
G3	•	•	•	•	•	•	•	•
G4	•	•	•	•	•	•	•	•
G5	•	•	•	•	•	•	•	•
G7	•	•	•	•	•	•	•	•
G9	•	•	•	•	•	•	•	•
G17	•	•	•	•	•	•	•	•
G18	•	•	•	•	•	•	•	•
G19	•	•	•	•	•	•	•	•
G25	•	•	•	•	•	•	•	•
G26	•	•	•	•	•	•	•	•
G33	•	•	•	•	•	•	•	•
G34	•	•	•	•	•	•	•	•

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowa-nie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowa-nie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowa-nie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowa-nie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowa-nie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowa-nie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
GWPSON	●	●	●	●	●	●	●	●
H...	●	●	●	●	●	●	●	●
HOLES1	●	●	●	●	●	●	●	●
HOLES2	●	●	●	●	●	●	●	●
I	●	●	●	●	●	●	●	●
I1	●	●	●	●	●	●	●	●
IC	●	●	●	●	●	●	●	●
ICYCOF	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
KONTT	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
KR	●	●	●	●	●	●	●	●
L	●	●	●	●	●	●	●	●
LEAD								
Orientacja narzędzia	-	-	-	-	-	-	-	-
Wielomian orientacji	-	-	-	-	-	-	-	-
LEADOF	-	-	-	●	-	-	●	●
LEADON	-	-	-	●	-	-	●	●
LENTOAX	●	●	●	●	●	●	●	●
LFOF	●	●	●	●	●	●	●	●
LFON	●	●	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●	●	●
LFWP	●	●	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●	●	●
LIMS	●	●	●	●	●	●	●	●
LLI	●	●	●	●	●	●	●	●
LN	●	●	●	●	●	●	●	●
LOCK	●	●	●	●	●	●	●	●
LONGHOLE	●	●	●	●	●	●	●	●
LOOP	●	●	●	●	●	●	●	●

Polecenia M ... R

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
M0	●	●	●	●	●	●	●	●
M1	●	●	●	●	●	●	●	●
M2	●	●	●	●	●	●	●	●
M3	●	●	●	●	●	●	●	●
M4	●	●	●	●	●	●	●	●
M5	●	●	●	●	●	●	●	●
M6	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
MOVT	●	●	●	●	●	●	●	●
MSG	●	●	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●	●	●
N	●	●	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●	●	●
NC	●	●	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●	●	●
NEWT	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowa-nie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowa-nie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowa-nie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
NORM	●	●	●	●	●	●	●	●
NOT	●	●	●	●	●	●	●	●
NPROT	●	●	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-	-	-
OF	●	●	●	●	●	●	●	●
OFFN	●	●	●	●	●	●	●	●
OMA1	-	-	-	-	-	-	-	-
OMA2	-	-	-	-	-	-	-	-
OMA3	-	-	-	-	-	-	-	-
OMA4	-	-	-	-	-	-	-	-
OMA5	-	-	-	-	-	-	-	-
OR	●	●	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-	-	-
ORIC	-	-	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-	-	-
ORID	-	-	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-	-	-
ORIS	-	-	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-	-	-

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	-	-	-	-	-	-	-	-
POCKET3	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowa-nie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowa-nie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowa-nie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
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	●	●	●	●	●	●	●	●

Polecenia S ... Z

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie 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	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie 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	-	-	-	-	-	-	-	-

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie 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	●	-	●	-	●	●	-	-

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowanie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowanie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowanie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie 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	●	●	●	●	●	●	●	●

Polecenie	SINUMERIK 828D							
● Standard ○ Opcja - niedostępna	SW24x(5) CNC-SW Frezowa-nie Export (me42)	SW24x(5) CNC-SW Toczenie Export (te42)	SW26x(3) CNC-SW Frezowa-nie Export (me62)	SW26x(3) CNC-SW Toczenie Export (te62)	SW28x(2) CNC-SW Frezowa-nie Export (me821)	SW28x(1) CNC-SW Frezowanie Adv. Export (me822)	SW28x(2) CNC-SW Toczenie Export (te821)	SW28x(1) CNC-SW Toczenie Adv. Export (te822)
WALCS5	●	●	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●	●	●
WHEN	●	●	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●	●	●
WHILE	●	●	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●	●	●
WRITE	●	●	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●	●	●
X	●	●	●	●	●	●	●	●
XOR	●	●	●	●	●	●	●	●
Y	●	●	●	●	●	●	●	●
Z	●	●	●	●	●	●	●	●

21.2.2 Warianty sterowania - szlifowanie

Polecenia A ... C

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
:	●	●	●	●	●	●
*	●	●	●	●	●	●
+	●	●	●	●	●	●
-	●	●	●	●	●	●
<	●	●	●	●	●	●
<<	●	●	●	●	●	●
<=	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
=	●	●	●	●	●	●
>=	●	●	●	●	●	●
/	●	●	●	●	●	●
/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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	○	○	○	○	○	○

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	Nazwa osi kanału	-	-	-
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	○	○	○	○	○	○

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○
CR	●	●	●	●	●	●
CROT	●	●	●	●	●	●
CROTS	●	●	●	●	●	●
CRPL	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○
CT	●	●	●	●	●	●
CTAB	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-
CTABID	-	-	-	-	-	-

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	-	-	-	-	-	-

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenia D ... F

Polecenie	SINUMERIK 828D					
• Standard ○ Opcja - niedostępna	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	•	•	•	•	•	•
DELMLRES	•	•	•	•	•	•
DELMT	-	-	-	-	-	-
DELOBJ	-	-	-	-	-	-
DELT	•	•	•	•	•	•
DELTC	•	•	•	•	•	•
DELTOOLENV	•	•	•	•	•	•
DIACYCOFA	•	•	•	•	•	•
DIAM90	•	•	•	•	•	•
DIAM90A	•	•	•	•	•	•
DIAMCHAN	•	•	•	•	•	•
DIAMCHANA	•	•	•	•	•	•
DIAMCYCOF	•	•	•	•	•	•
DIAMOF	•	•	•	•	•	•
DIAMOFA	•	•	•	•	•	•
DIAMON	•	•	•	•	•	•
DIAMONA	•	•	•	•	•	•
DIC	•	•	•	•	•	•
DILF	•	•	•	•	•	•
DISABLE	•	•	•	•	•	•
DISC	•	•	•	•	•	•
DISCL	•	•	•	•	•	•

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	○	○	○	○	○	○

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	●	●	●	●	●	●

Polecenia G ... L

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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 ... GFRAME100	< 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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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						
Orientacja narzędzia	-	-	-	-	-	-
Wielomian orientacji	-	-	-	-	-	-
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

Polecenia M ... R

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	-	-	-	-	-	-

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
OEMIPO2	-	-	-	-	-	-
OF	●	●	●	●	●	●
OFFN	●	●	●	●	●	●
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
OR	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-
ORIS	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-
ORISON	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-
OS	○	○	○	○	○	○

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	-	-	-	-	-	-

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	●	●	●	●	●	●

Polecenia S ... Z

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	-	-	-	-	-	-

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
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	●	●	●	●	●	●

Polecenie	SINUMERIK 828D					
● Standard ○ Opcja - niedostępna	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)
WRTPR	●	●	●	●	●	●
X	●	●	●	●	●	●
XOR	●	●	●	●	●	●
Y	●	●	●	●	●	●
Z	●	●	●	●	●	●

21.3 Aktualny język w HMI

Poniższa tablica zawiera wszystkie języki dostępne w interfejsie graficznym.

Aktualnie ustawiony język można odczytać w programie obróbki i w akcjach synchronicznych poprzez następującą zmienną systemową:

$AN_LANGUAGE_ON_HMI = <wartość>

<Wartość>	Język	Skrót języków
1	Niemiecki (Niemcy)	DEU
2	Francuski	FRA
3	Angielski (Zjednoczone Królestwo)	ENG
4	Hiszpański	ESP
6	Włoski	ITA
7	Niderlandzki	NLD
8	Chiński uproszczony	CHS
9	Szwedzki	SVE
18	Węgierski	HUN
19	Fiński	FIN
28	Czeski	CSY
50	Portugalski (Brazylia)	PTB
53	Polski	PLK
55	Duński	DAN
57	Rosyjski	RUS
68	Słowacki	SKY
72	Rumuński	ROM
80	Chiński (tradycyjny)	CHT
85	Koreański	KOR
87	Japoński	JPN
89	Turecki	TRK

Uwaga

Aktualizacja $AN_LANGUAGE_ON_HMI następuje:

- Po rozruchu systemu.
- Po restarcie NC i/lub PLC.
- Po przełączeniu na inny NC w ramach M2N.
- Po przełączeniu języka na HMI.

Aneks

A

A.1 Lista skrótów

A	
A	Wyjście
ADI4	Analogowy interfejs napędu dla 4 osi
AC	Sterowanie adaptacyjne
ALM	Active Line Module
ARM	Asynchroniczny silnik obrotowy
AS	System automatyzacji
ASCII	American Standard Code for Information Interchange: Amerykańska norma kodów do wymiany informacji
ASIC	Application Specific Integrated Circuit: Obwód sterownia użytkownika
ASUP	Podprogram asynchroniczny
AUXFU	Auxiliary Function: Funkcja pomocnicza
AWL	Lista instrukcji
AWP	Program użytkownika

B	
BA	Tryb pracy
BAG	Grupa trybów pracy
BCD	Binary Coded Decimals: Liczby dziesiętne zakodowane w kodzie binarnym
BERO	Bezdotykowy łącznik zbliżeniowy
BI	Binector Input
BICO	Binector Connector
BIN	Binary Files: Dane binarne
BIOS	Basic Input Output System
BKS	Bazowy układ współrzędnych
BO	Binector Output
BTSS	Interfejs pulpitu obsługi

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Cykle kompilowane
CEC	Cross Error Compensation
CI	Connector Input
Karta CF	Karta Compact Flash

C	
CNC	Computerized Numerical Control: Komputerowe sterowanie numeryczne
CO	Connector Output
CoL	Certificate of License
COM	Communication
CPA	Compiler Projecting Data: Dane projektowe kompilatora
CRT	Cathode Ray Tube: Kineskop
CSB	Central Service Board: Moduły PLC
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: Jednostka centralna komputera
CR	Carriage Return
CTS	Clear To Send: Komunikat gotowości do wysyłania w przypadku szeregowych interfejsów danych
CUTCOM	Cutter Radius Compensation: Korekcja promienia narzędzia

D	
DAU	Przetwornik cyfrowo-analogowy
DB	Moduł danych (PLC)
DBB	Bajt modułu danych (PLC)
DBD	Podwójne słowo modułu danych (PLC)
DBW	Słowo modułu danych (PLC)
DBX	Bit modułu danych (PLC)
DDE	Dynamic Data Exchange
DDS	Drive Data Set: Zestaw danych napędu
DIN	Niemiecka Norma Przemysłowa
DIO	Data Input/Output: Wyświetlanie przesyłanych danych
DIR	Directory: Katalog
DLL	Dynamic Link Library
DO	Drive Object
DPM	Dual Port Memory
DPR	Dual Port RAM
DRAM	Pamięć dynamiczna (niebuforowana)
DRF	Differential Resolver Function: Funkcja selsynu obrotowego (kółko ręczne)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run: Posuw próbny
DSB	Decoding Single Block: Dekodowanie pojedynczymi blokami
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Słowo danych
DWORD	Podwójne słowo (aktualnie 32 bity)

E	
E	Wejście
EES	Execution from External Storage
E/A	Wprowadzenie/wyprowadzenie
ENC	Encoder: Enkoder wartości rzeczywistej
EFP	Zwykły moduł peryferyjny (moduł PLC–E/A)
EGB	Elektronicznie zagrożone moduły/elementy konstrukcyjne
EMV	Kompatybilność elektromagnetyczna
EN	Norma Europejska
ENC	Encoder: Enkoder wartości rzeczywistej
EnDat	Interfejs enkodera
EPROM	Erasable Programmable Read Only Memory: Kasowalna, elektrycznie programowana pamięć do odczytu
ePS Network Services	Zdalne usługi serwisowe poprzez łącza internetowe w celu zdalnej konserwacji maszyny
EQN	Określenie typu enkodera absolutnego o 2048 sygnałach sinusoidalnych/obrót
ES	Engineering System
ESR	Rozszerzone zatrzymanie i wycofanie
ETC	Przycisk ETC ">"; rozszerzenie paska przycisków w tym samym menu

F	
FB	Moduł funkcjonalny (PLC)
FC	Function Call: Moduł funkcjonalny (PLC)
FEPROM	Flash-EPROM: Pamięć do odczytu i zapisu
FIFO	First In First Out: Pamięć, która pracuje bez podania adresu i której dane są czytane w tej samej kolejności, w jakiej zostały zapisane
FIPO	Interpolator dokładny
FPU	Floating Point Unit: Jednostka zmiennoprzecinkowa
FRK	Korekcja promienia frezu
FST	Feed Stop: Posuw stop
FUP	Plan funkcji (metoda programowania PLC)
FW	Oprogramowanie sprzętowe

G	
GC	Global Control (PROFIBUS: Telegram Broadcast)
GDIR	Globalna pamięć programów obróbki
GEO	Geometria, np. oś geometryczna
GIA	Gear Interpolation Data: Dane interpolacyjne przekładni
GND	Signal Ground
GP	Program podstawowy (PLC)
GS	Stopień przekładni
GSD	Blik konfiguracyjny do opisu PROFIBUS Slave

G	
GSDML	Generic Station Description Markup Language: Bazujący na XML język opisowy do sporządzania pliku GSD
GUD	Global User Data: Globalne dane użytkownika

H	
HEX	Skrót liczby szesnastkowej
HiFu	Funkcja pomocnicza
HLA	Hydrauliczny napęd liniowy
HMI	Human Machine Interface: Interfejs obsługi SINUMERIK
HSA	Napęd wrzeciona głównego
HW	Hardware

I	
IBN	Uruchomienie
IKA	Kompensacja interpolacyjna
IM	Interface-Modul: Moduł przyłączeniowy
IMR	Interface-Modul Receive: Moduł przyłączeniowy do odbioru
IMS	Interface-Modul Send: Moduł przyłączeniowy do wysyłania
INC	Increment: Przyrost
INI	Initializing Data: Dane inicjalizacyjne
IPO	Interpolator
ISA	International Standard Architecture
ISO	International Standard Organization

J	
JOG	Jogging: Tryb ustawiania

K	
K_V	Współczynnik wzmocnienia obwodu regulacji
K_P	Wzmocnienie proporcjonalne
$K_Ü$	Stosunek przełożenia
KOP	Schemat drabinkowy (metoda programowania PLC)

L	
LAI	Logic Machine Axis Image: Logiczne odwzorowanie osi maszyny
LAN	Local Area Network
LCD	Liquid-Crystal Display: Wyświetlacz ciekłokrystaliczny
LED	Light Emitting Diode: Dioda LED
LF	Line Feed

L	
LMS	System pomiaru położenia
LR	Regulator położenia
LSB	Least Significant Bit: bit najmniej znaczący
LUD	Local User Data: Dane użytkownika (lokalnie)

M	
MAC	Media Access Control
MAIN	Main program: Program główny (OB1, PLC)
MB	Megabajt
MCI	Motion Control Interface
MCIS	Motion–Control–Information–System
MCP	Machine Control Panel: Pulpit sterowniczy maszyny
MD	Dana maszynowa lub dane maszynowe
MDA	Manual Data Automatic: Wprowadzanie ręczne
MDS	Motor Data Set: Zestaw danych silnika
MELDW	Słowo komunikatu
MKS	Układ współrzędnych maszyny
MM	Motor Module
MPF	Main Program File: Program główny (NC)
MSTT	Pulpit sterowniczy maszyny

N	
NC	Numerical Control: Sterowanie numeryczne z przygotowaniem bloków, zakresem ruchu, itd.
NCU	Numerical Control Unit: Jednostka sprzętowa NC
NRK	Określenie systemu operacyjnego NC
NST	Sygnał interfejsowy
NURBS	Non-Uniform Rational B-Spline
NV	Przesunięcie punktu zerowego
NX	Numerical Extension: Moduł rozszerzania osi

O	
OB	Moduł organizacyjny w PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Urządzenie obsługi
OPI	Operation Panel Interface: Przyłącze pulpitu obsługi
OPT	Options: Opcje
OLP	Optical Link Plug: Wtyczka magistralna światłowodowa
OSI	Open Systems Interconnection: Normalizacja dla komunikacji komputerowej

P	
PAA	Odwzorowanie procesu wyjść
PAE	Odwzorowanie procesu wejść
PC	Personal Computer
PCIN	Nazwa oprogramowania do wymiany danych ze sterowaniem
PCMCIA	Personal Computer Memory Card International Association: Normalizacja kart pamięci
PCU	PC Unit: PC-Box (jednostka komputerowa)
PG	Urządzenie do programowania
PKE	Identyfikator parametru: Część PKW
PKW	Identyfikator parametru: Wartość (część parametru PPO)
PLC	Programmable Logic Control: Programowalny sterownik logiczny
PN	PROFINET
PNO	Organizacja użytkowników PROFIBUS
PO	POWER ON
POE	Jednostka organizacyjna programu
POS	Pozycja/pozycjonowanie
POSMO A	Positioning Motor Actuator: silnik pozycjonujący
POSMO CA	Positioning Motor Compact AC: Kompletna jednostka napędowa ze zintegrowanym zespołem mocy i sterowania, jak też jednostką pozycjonowania i pamięcią programów; zasilanie prądem przemiennym
POSMO CD	Positioning Motor Compact DC: jak CA, jednak zasilanie prądem stałym
POSMO SI	Positioning Motor Servo Integrated: Silnik pozycjonujący; zasilanie prądem stałym
PPO	Parameter Prozessdaten Objekt ; Cykliczny telegram danych przy przesyłaniu przez PROFIBUS-DP oraz profilu "napędy ze zmienną prędkością obrotową"
PPU	Panel Processing Unit (sprzęt centralny sterowania CNC bazującego na pulpicie, np. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Szeregowa magistrala danych
PRT	Test programu
PSW	Słowo sterujące programu
PTP	Point to Point: Punkt do punktu
PUD	Program Global User Data: Zmienne użytkownika globalne dla programu
PZD	Dane procesowe: Część danych procesu PPO

Q	
QFK	Kompensacja błędu ćwiartki koła

R	
RAM	Random Access Memory: Pamięć do zapisu/odczytu
REF	Funkcja bazowania do punktu odniesienia
REPOS	Funkcja repozycjonowania
RISC	Reduced Instruction Set Computer: Typ procesora o małym zasobie poleceń i szybkim ich wykonywaniu

R	
ROV	Rapid Override: Korekcja posuwu szybkiego
RP	Parametry R, parametry obliczeniowe, predefiniowane zmienne użytkownika
RPA	R-Parameter Active: Obszar pamięci w NC dla numerów parametrów R
RPY	Roll Pitch Yaw: Rodzaj obrotu układu współrzędnych
RTLI	Rapid Traverse Linear Interpolation: Interpolacja liniowa przy posuwie szybkim
RTS	Request To Send: Włączenie wysyłania, sygnał sterujący dla szeregowych interfejsów danych
RTCP	Real Time Control Protocol

S	
SA	Akcja synchroniczna
SBC	Safe Brake Control: Bezpieczne sterowanie hamulcem
SBL	Single Block: Wykonywanie pojedynczymi blokami
SBR	Subroutine: Podprogram (PLC)
SD	Dana ustawcza lub dane ustawcze
SDB	System moduł danych
SEA	Setting Data Active: Oznaczenie (typ pliku) dla danych ustawczych
SERUPRO	Search–Run by Program Test: Szukanie bloku za pomocą funkcji "Test programu"
SFB	Systemowy moduł funkcji
SFC	System Function Call
SGE	Wejście dedykowane dla bezpieczeństwa
SGA	Wejście dedykowane dla bezpieczeństwa
SH	Bezpieczne zatrzymanie
SIM	Single in Line Module
SK	Przycisk programowy
SKP	Skip: Funkcja do pomijania bloku programu obróbki
SLM	Synchroniczny silnik liniowy
SM	Silnik krokowy
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Sub Program File: Podprogram (NC)
SPS	Sterowanie programowane w pamięci = PLC
SRAM	Pamięć statyczna (buforowana)
SRK	Korekcja promienia ostrza
SRM	Synchroniczny silnik obrotowy
SSFK	Kompensacja błędu skoku śruby pociągowej
SSI	Serial Synchron Interface: Szeregowy interfejs synchroniczny
SSL	Szukanie bloku
STW	Słowo sterowania
SUG	Prędkość obwodowa ściernicy
SW	Oprogramowanie

S	
SYF	System Files: Dane systemowe
SYNACT	Synchronized Action: Akcja synchroniczna

T	
TB	Moduł terminala (SINAMICS)
TCP	Tool Center Point: Wierzchołek narzędzia
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: Oznaczenie dla danych maszynowych
TIA	Totally Integrated Automation
TM	Moduł terminala (SINAMICS)
TO	Tool Offset: Korekcja narzędzia
TOA	Tool Offset Active: Oznaczenie (typ pliku) dla korekcji narzędzia
TRANSMIT	Transform Milling Into Turning: Transformacja współrzędnych dla obróbki frezarskiej na tokarce
TTL	Logika tranzystor-tranzystor (typ interfejsu)
TZ	Cykl technologiczny

U	
UFR	User Frame: Przesunięcie punktu zerowego
UP	Podprogram
USB	Universal Serial Bus
USV	Zasilanie bezprzerwowe

V	
VDI	Wewnętrzny interfejs komunikacyjny między NC i PLC
VDI	Stowarzyszenie Inżynierów Niemieckich
VDE	Związek Elektrotechników Niemieckich
VI	Voltage Input
VO	Voltage Output
VSA	Napęd posuwu

W	
WAB	Funkcja miękkie dosunięcie i odsunięcie
WKS	Układ współrzędnych obrabianego przedmiotu
WKZ	Narzędzie
WLK	Korekcja długości narzędzia
WOP	Programowanie zorientowane na warsztat
WPD	Work Piece Directory: Katalog obrabianego przedmiotu
WRK	Korekcja promienia narzędzia

W	
WZ	Narzędzie
WZK	Korekcja narzędzia
WZV	Zarządzanie narzędziami
WZW	Wymiana narzędzia

X	
XML	Extensible Markup Language

Z	
ZOA	Zero Offset Active: Oznaczenie przesunięć punktu zerowego
ZSW	Słowo stanu (napędu)

A.2 Przegląd dokumentacji

Dokumentacja ogólna

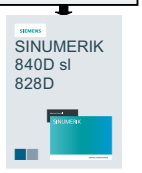
Materiały promocyjne
– SINUMERIK 840D sl
– SINUMERIK 828D
– SINUMERIK 828D BASIC

Katalog NC 62
SINUMERIK 840D sl

Katalog NC 82
SINUMERIK 828D

Katalog PM 21
SIMOTION, SINAMICS S120

Podręcznik systemu
Inteligentna obsługa obrabiarek



Podręcznik systemu
Ctrl-Energy

Dokumentacja użytkownika

Podręcznik obsługi
– Universal
– Toczenie
– Frezowanie
– Szlifowanie

Podręcznik programowania
– Podstawy
– Przygotowanie do pracy
– Cykle pomiarowe

Podręcznik programowania
– Toczenie ISO
– Frezowanie ISO

Podręcznik diagnostyczny
Alarmy

Podręcznik diagnostyczny
Alarmy

Dokumentacja producenta / serwisowa

Podręcznik urządzenia
– NCU
– Komponenty obsługowe i połączenie sieciowe
– ADI4

Podręcznik urządzenia
Podręcznik uruchomieniowy
Podręcznik serwisowy

Podręcznik uruchomieniowy
– CNC: NCK, PLC, Napęd
– oprogramowanie podstawowe i obsługowe

Lista parametrów
– Dane maszynowe
– Sygnały interfejsowe
– Zmienne

Lista parametrów
– Dane maszynowe
– Sygnały interfejsowe
– Parametry
– Zmienne

Podręcznik systemu
Podręcznik projektowania maszyn

Dokumentacja producenta / serwisowa

Opis funkcji
– Funkcje podstawowe
– Funkcje rozszerzone
– Funkcje specjalne
– Akcje synchroniczne
– Dialekty ISO

Opis funkcji
Zarządzanie narzędziami

Opis funkcji
Safety Integrated

Opis funkcji
Safety Integrated

Podręcznik projektowania
- Wytyczne dotyczące montażu zgodnie z EMC
– Industrial Security

Info / Szkolenia

Dokumentacja szkoleniowa
– Łatwiejsze frezowanie z ShopMill
– Łatwiejsze toczenie z ShopTurn

Podręcznik Wytwarzanie narzędzi i form

Dokumentacja elektroniczna

DOConCD

Industry Online Support (SIOS)

Industry Mall

Glosariusz

Adres

Adres jest identyfikatorem dla określonego argumentu albo zakresu argumentów, np. wejście, wyjście itd.

Adres osi

Patrz → Nazwa osi

Akcje synchroniczne

1. Wyprowadzenie funkcji pomocniczej
   Podczas obróbki mogą z programu CNC być wyprowadzane do PLC funkcje technologiczne (→ funkcje pomocnicze). Poprzez te funkcje są np. sterowane urządzenia dodatkowe obrabiarki, jak tuleja wrzecionowa, chwytak, uchwyt tokarski, itd.
2. Szybkie wyprowadzenie funkcji pomocniczej
   Dla krytycznych pod względem czasu funkcji łączeniowych można zminimalizować czasy kwitowania dla → funkcji pomocniczych i uniknąć niepotrzebnych punktów zatrzymania w procesie obróbki.

Alarmy

Wszystkie → komunikaty i alarmy wyświetlane są na pulpicie obsługi zwykłym tekstem z podaniem daty i godziny oraz odpowiedniego symbolu stanowiącego kryterium kasowania. Alarmy i komunikaty wyświetlane są osobno.

1. Alarmy i komunikaty w programie obróbki
   Alarmy i komunikaty mogą być wyświetlane zwykłym tekstem bezpośrednio z programu obróbki.
2. Alarmy i komunikaty PLC
   Alarmy i komunikaty maszyny mogą być wyświetlane zwykłym tekstem z programu PLC. Do tego nie są potrzebne żadne dodatkowe pakiety modułów funkcjonalnych.

Archiwizowanie

Zapisywanie plików i/lub katalogów na **dyski** zewnętrzne.

Automatyka

Tryb pracy sterowania (praca z ciągiem bloków według DIN): Rodzaj pracy w systemach NC, w których → program obróbki jest wybierany i wykonywany w sposób ciągły.

Bateria buforująca

Bateria buforująca gwarantuje, że → program użytkownika w → CPU jest zapisany w sposób odporny na przerwy zasilania a ustalone obszary danych i znaczniki, czasy i liczniki są utrzymywane.

Bazowy układ współrzędnych

Kartezjański układ współrzędnych, jest transformowany na układ współrzędnych maszyny.

W → programie obróbki programista używa nazw osi bazowego układu współrzędnych. Istnieje on również, gdy → transformacja nie jest aktywna, i jest równoległy do → układu współrzędnych maszyny. Różnica występuje w → nazwie osi.

Blok

Jako bloki są określane wszystkie dane, które są potrzebne do sporządzenia i wykonania programu.

Blok główny

Rozpoczynający się od ":" blok, który zawiera wszystkie dane, aby móc w bezpieczny sposób uruchomić przebieg pracy w → programie obróbki.

Blok pomocniczy

Rozpoczynający się od "N" blok zawierający informacje dot. kroku roboczego np. podanie pozycji.

Blok programu obróbki

Część → programu obróbki, ograniczona przez Line Feed. Rozróżnia się → bloki główne i → bloki pomocnicze.

Bloki pośrednie

Ruchy postępowe z wybraną → korekcją narzędzia (`G41`/`G42`) mogą być przerywane przez ograniczoną liczbę bloków pośrednich (bloki bez ruchów w osiach w płaszczyźnie korekcji), przy czym korekcja narzędzia może być jeszcze prawidłowo obliczana. Dopuszczalną liczbę bloków pośrednich, które sterowanie czyta wyprzedzająco, można ustawić przez parametry systemowe.

Calowy system miar

System miar, który definiuje odległości w calach i ich ułamkach.

CNC

Patrz → NC

Computerized Numerical Control: obejmuje komponenty → NC, → PLC, HMI, → COM.

CNC

Patrz → NC

Computerized Numerical Control: obejmuje komponenty → NC, → PLC, HMI, → COM.

COM

Komponent sterowania NC do realizacji i koordynacji komunikacji.

CPU

Central Processor Unit, patrz → sterowanie programowane w pamięci

C-Spline

C-Spline jest najbardziej znaną i najczęściej stosowaną funkcją spline. Przejścia w punktach oparcia są określone krzywą wielomianową o ciągłej pochodnej lub wykazują stałe zakrzywienie. Stosowane są wielomiany trzeciego stopnia.

Cykle

Chronione podprogramy do wykonywania powtarzających się operacji obróbkowej na → obrabianym przedmiocie.

Cykle standardowe

Dla często powtarzających się zadań obróbkowych są do dyspozycji cykle standardowe:

- dla technologii "wiercenie/frezowanie"
- dla technologii "toczenie"

W oknie obsługowym "Program" są w menu "Obsługa cykli" wyszczególnione dostępne cykle. Po wyborze żądanego cyklu obróbkowego wyświetlane są tekstem jawnym niezbędne parametry dla przyporządkowania wartości.

Dane ustawcze

Dane, które informują sterowanie NC o właściwościach obrabiarki w sposób zdefiniowany przez oprogramowanie systemowe.

Definicja zmiennej

Definicja zmiennej obejmuje ustalenie typu danych i nazwy zmiennej. Przy pomocy nazwy zmiennej można sięgać do jej wartości.

Diagnoza

1. Obszar obsługi sterowania
2. Sterowanie wyposażone jest zarówno w program do samodiagnozy oraz ekrany diagnostyczne: wyświetlanie statusu, alarmu, serwisowe

Dosunięcie do punktu stałego

Obrabiarki mogą w sposób zdefiniowany dokonywać dosunięcia do punktów stałych jak punkt zmiany narzędzia, punkt załadunku, punkt wymiany palety itd. Współrzędne tych punktów są zapisane w sterowaniu. Sterowanie wykonuje ruch w odpowiednich osiach, jeżeli to możliwe, → posuwem szybkim.

DRF

Differential Resolver Function: Funkcja NC, która w połączeniu z elektronicznym kółkiem ręcznym wytwarza przyrostowe przesunięcie punktu zerowego w trybie pracy automatycznej.

Edytor

Edytor umożliwia sporządzenie, zmianę, uzupełnienie, łączenie i wstawianie programów/ tekstów/bloków programu.

Edytor tekstów

Patrz → Edytor

Frame

Frame jest to instrukcja obliczeniowa, która transformuje kartezjański układ współrzędnych w inny kartezjański układ współrzędnych. Frame zawiera składowe → przesunięcie punktu zerowego, → obrót, → skalowanie, → lustrzane odbicie.

Frame programowane

Przy pomocy programowanych → frame mogą dynamicznie, w trakcie wykonywania programu obróbki, być definiowane nowe punkty wyjściowe układu współrzędnych. Rozróżnia się ustalenie bezwzględne na podstawie nowego frame i ustalenie addytywne w odniesieniu do istniejącego punktu wyjściowego.

Funkcje bezpieczeństwa

Sterowanie zawiera stale aktywne nadzory, które rozpoznają zakłócenia w → CNC, sterowaniu adaptacyjnym (→ PLC) i maszynie na tyle wcześnie, że są w dużym stopniu wykluczone uszkodzenia obrabianego przedmiotu, narzędzia albo maszyny. W przypadku zakłócenia przebieg obróbki jest przerywany a napędy są zatrzymywane, przyczyna zakłócenia jest zapisywana w pamięci i jest wyświetlany alarm. Równocześnie PLC otrzymuje informację, że jest aktywny alarm CNC.

Funkcje pomocnicze

Przy pomocy funkcji pomocniczych można w → programach obróbki przekazywać → parametry do → PLC, które wyzwalają reakcje zdefiniowane przez producenta maszyny.

Geometria

Opis → przedmiotu obrabianego w → układzie współrzędnych przedmiotu obrabianego.

Granica zatrzymania dokładnego

Gdy wszystkie osie uczestniczące w tworzeniu konturu osiągną swoją granicę zatrzymania dokładnego, wówczas sterowanie zachowuje się tak, jakby dokładnie osiągnęło punkt docelowy. Następuje przełączenie na następny blok → programu obróbki.

Graniczna prędkość obrotowa

Maksymalna/minimalna prędkość obrotowa (wrzeciona): Przez zadanie w danych maszynowych, → PLC lub → danych ustawczych można ograniczyć maksymalną prędkość obrotową wrzeciona.

Grupa trybów pracy

Technologicznie przynależne do siebie osie i wrzeciona mogą być połączone w grupę trybów pracy (BAG). Osie/wrzeciona jednej grupie BAG mogą być sterowane przez jeden lub kilka → kanałów. Kanałom grupy trybów pracy (BAG) jest zawsze przyporządkowany taki sam → tryb pracy.

Gwintowanie otworu bez oprawki kompensacyjnej

Przy pomocy tej funkcji można gwintować otwory bez użycia oprawki kompensacyjnej. Dzięki interpolującemu ruchowi wrzeciona, jako osi obrotowej i osi gwintowania gwint jest wykonywany dokładnie na głębokość końcową, np. gwint w otworze nieprzelotowym (warunek: praca wrzeciona jako oś obrotowa).

HIGHSTEP

Podsumowanie możliwości programowania dla → PLC systemu AS300/AS400.

HW-Config

Narzędzie SIMATIC S7 do konfiguracji i parametryzacji komponentów sprzętowych w ramach projektu S7.

Identyfikator

Słowa według DIN 66025 są uzupełniane przez identyfikatory (nazwy) dla zmiennych (zmienne obliczeniowe, zmienne systemowe, zmienne użytkownika), dla podprogramów, słów kluczowych i słów z wieloma literami adresowymi. Znaczenie tych uzupełnień jest pod

względem znaczenia równoważne słowom przy budowie bloków. Identyfikatory muszą być jednoznaczne. Tych samych identyfikatorów nie wolno stosować do różnych obiektów.

Inicjalizacja systemu

Ładowanie programu systemowego po Power On.

Interfejs graficzny

Interfejs graficzny (BOF) jest medium, mającym postać ekranu, do wyświetlania stanu sterowania CNC. Jest on wyposażony w poziome i pionowe przyciski programowe.

Interpolacja kołowa

Narzędzie → ma wykonać ruch po okręgu między ustalonymi punktami konturu z podanym posuwem, a przy tym wykonywać obróbkę przedmiotu obrabianego.

Interpolacja linii śrubowej

Interpolacja linii śrubowej nadaje się szczególnie do prostego wykonywania gwintów wewnętrznych i zewnętrznych przy pomocy frezów kształtowych i do frezowania rowków smarowych.

Linia śrubowa składa się przy tym z dwóch ruchów:

- Ruch kołowy w płaszczyźnie
- Ruch liniowy prostopadle do tej płaszczyzny

Interpolacja prostoliniowa

Ruch narzędzia następuje po prostej do punktu docelowego i jest przy tym prowadzona obróbka.

Interpolacja spline

Przy pomocy interpolacji spline sterowanie może utworzyć z niewielu zadanych punktów oparcia zadanego konturu gładki przebieg krzywej.

Interpolacja wielomianowa

Przy pomocy interpolacji wielomianowej mogą być wytwarzane najróżniejsze przebiegi krzywych, jak **funkcje prostoliniowe, paraboliczne, potęgowe** (SINUMERIK 840D sl).

Interpolator

Jednostka logiczna → NC, która po podaniu pozycji docelowych w programie obróbki określa wartości pośrednie dla ruchów wykonywanych w poszczególnych osiach.

Jednostka TOA

Każdy → zakres TOA może zawierać wiele jednostek TOA. Liczba możliwych jednostek TOA jest ograniczona przez maksymalną liczbę aktywnych → kanałów. Jedna jednostka TOA obejmuje dokładnie jeden moduł danych narzędzi i jeden moduł danych magazynu. Dodatkowo może jeszcze być zawarty moduł danych nośnika narzędzi (opcjonalnie).

Język wysokiego poziomu CNC

Język wysokiego poziomu służy do pisania programów NC, → akcji synchronicznych i → cykli. Zapewnia on: Struktury kontrolne, → zmienne definiowane przez użytkownika, → zmienne systemowe, → technologia makr.

JOG

Tryb pracy sterowania (ustawianie): W trybie pracy JOG można ustawiać maszynę. Poszczególnymi osiami i wrzecionami można przez przyciski kierunkowe wykonywać ruch impulsowy. Dalsze funkcje w trybie pracy JOG to → najazd na punkt referencyjny, → Repos, jak też → Preset (ustawienie wartości rzeczywistej).

Kanał

Kanał charakteryzuje się tym, że niezależnie od innych kanałów może on wykonywać → program obróbki. Kanał steruje wyłącznie przyporządkowanymi mu osiami i wrzecionami. Przebiegi programów obróbki w różnych kanałach mogą być koordynowane przez → synchronizację.

Kanał obróbki

Dzięki strukturze kanałowej można przez równoległe wykonywanie ruchów skrócić czasy dodatkowe, np. ruch manipulatora załadowczego odbywa się równocześnie z obróbką. Kanał CNC należy przy tym traktować jako oddzielne sterowanie CNC z dekodowaniem, przetwarzaniem bloków i interpolacją.

Klucz programowania

Znaki i ciągi znaków, które w języku programowania mają dla → programu obróbki ustalone znaczenie.

Koincydencja wyprzedzająca

Zmiana bloku już wtedy, gdy droga po torze zbliżyła się do pozycji końcowej o zadaną wartość delta.

Kompensacja błędu ćwiartki koła

Błędy konturu na przejściach między ćwiartkami, które powstają w wyniku zmieniających się warunków tarcia na prowadnicach, dają się w dużym stopniu wyeliminować przez

kompensację błędu ćwiartki. Parametryzowanie kompensacji błędu ćwiartki następuje w drodze testu kształtu kołowego.

Kompensacja błędu skoku śruby pociągowej

Wyrównywanie niedokładności mechanicznych uczestniczącej w posuwie śruby pociągowej tocznej przez sterowanie na podstawie zapisanych wartości pomiarowych odchyleń.

Kompensacja interpolacyjna

Za pomocą kompensacji interpolacyjnej, np. → kompensacji błędu skoku śruby pociągowej, ugięcia, odchylenia kątowego i temperatury mogą być kompensowane mechaniczne błędy maszyny.

Kompensacja luzów

Kompensacja mechanicznych luzów maszyny, np. luzy nawrotu w śrubach pociągowych tocznych. Kompensację luzu dla każdej osi można wprowadzić oddzielnie.

Komunikaty

Wszystkie komunikaty zaprogramowane w programie obróbki i → alarmy rozpoznane przez system są na pulpicie obsługi wyświetlane zwykłym tekstem z podaniem daty i godziny oraz odpowiedniego symbolu kryterium kasowania. Alarmy i komunikaty wyświetlane są osobno.

Kontur

Obrys → przedmiotu obrabianego

Kontur części gotowej

Kontur przedmiotu obrabianego na gotowo. Patrz → Półfabrykat.

Kontur obrabianego przedmiotu

Zadany kontur wykonywanego/obrabianego → obrabianego przedmiotu.

Korekcja

Ręczna lub programowana możliwość ingerencji, która pozwala operatorowi na zmianę programowanych posuwów lub prędkości obrotowych, aby dopasować je do określonego obrabianego przedmiotu lub materiału.

Korekcja narzędzia

Uwzględnienie wymiarów narzędzia przy obliczaniu toru.

Korekcja promienia narzędzia

Aby móc bezpośrednio programować pożądany → kontur obrabianego przedmiotu, sterowanie musi przy uwzględnieniu promienia użytego narzędzia wykonywać ruch po torze równoległym do zaprogramowanego konturu (G41/G42).

Korekcja promienia ostrza

Przy programowaniu konturu zakłada się, że narzędzie jest szpiczaste. Ponieważ jest to w praktyce nie do zrealizowania, promień zakrzywienia użytego narzędzia jest podawany sterowaniu i przez nie uwzględniany. Przy tym punkt środkowy zakrzywienia jest prowadzony wokół konturu z przesunięciem o promień zakrzywienia.

Korektor posuwu

Na zaprogramowaną prędkość jest nakładane aktualne ustawienie prędkości przez → pulpit sterowniczy maszyny albo z → PLC (0-200%). Prędkość posuwu może dodatkowo zostać skorygowana w programie obróbki albo przez programowany współczynnik procentowy (1-200%).

KÜ

Stosunek przełożenia

KV

Współczynnik wzmocnienia w pętli regulacji

Look-Ahead

Za pomocą funkcji **Look-Ahead** przez „Wyprzedzające czytanie" sparametryzowanej liczby bloków zawierających ruch postępowy uzyskiwana jest optymalna prędkość obróbki.

Lustrzane odbicie

Przy lustrzanym odbiciu są zamieniane znaki wartości współrzędnych konturu odnośnie osi. Lustrzanego odbicia można dokonać równocześnie wokół wielu osi.

Masa

Za masę uważa się całość połączonych ze sobą nieaktywnych części maszyny, które również w przypadku błędu nie mogą uzyskać niebezpiecznego napięcia.

MDA

Tryb pracy sterowania: Manual Data Automatic. W trybie pracy MDA poszczególne bloki programu albo ich ciągi mogą bez odniesienia do programu głównego albo podprogramu być wprowadzane, a następnie natychmiast wykonywane przez naciśnięcie przycisku NC-Start.

Metryczne i calowe podawanie wymiarów

W programie obróbki wartości pozycji i skoku gwintu można programować w calach. Niezależnie od programowanego podawania wymiarów (G70 / G71) sterowanie jest ustawiane na system podstawowy.

Metryczny system miar

Znormalizowany system jednostek: dla długości np. mm (milimetr), m (metr).

Moduł danych

1. Jednostka danych → PLC, do której mogą sięgać programy → HIGHSTEP.
2. Jednostka danych → NC: moduły danych zawierają definicje globalnych danych użytkownika. Dane mogą być inicjalizowane bezpośrednio podczas definicji.

Moduł programowy

Moduły programowe zawierają programy główne i podprogramy → programów obróbki.

Nadzór nad konturem

Jako miara zgodności z konturem jest nadzorowany uchyb nadążania w ramach definiowanego pasma tolerancji. Niedopuszczalnie wysoki uchyb nadążania może wynikać np. z przeciążenia napędu. W takim przypadku następuje alarm i osie są zatrzymywane.

Napęd

Napędem jest jednostka CNC, która wykonuje regulację prędkości obrotowej i momentu na podstawie danych NC.

Narzędzie

Działająca w obrabiarce część, która powoduje obróbkę, (np. nóż tokarski, frez, wiertło, promień lasera ...).

Nazwa osi

W celu identyfikacji wszystkie osie kanałowe → oraz osie maszynowe sterowania muszą zostać oznaczone jednoznaczną nazwą. Osie → geometryczne oznaczane są jako X, Y, Z. Obracające się wokół osi geometrycznych → osie obrotowe, oznaczone są jako A, B, C.

NC

Komponent NC sterowania → CNC, który wykonuje → programy obróbki i w istotnej części koordynuje przebiegi ruchów na obrabiarce.

NRK

Numeric Robotic Kernel (system operacyjny → NC)

NURBS

Sterowanie wewnętrzne prędkością oraz interpolacja ruchu po torze są przeprowadzane na podstawie funkcji NURBS (**N**on **U**niform **R**ational **B**-**S**plines). W ten sposób w sterowaniu dla wszystkich interpolowanych osi dostępna jest jednolita metoda.

Obrabiany przedmiot

Cześć wykonywana / obrabiana przez obrabiarkę.

Obróbka skosów

Obróbka wiertarska i frezarska powierzchni obrabianych przedmiotów, które nie leżą w płaszczyznach współrzędnych maszyny, może być komfortowo prowadzona przy wsparciu przez funkcję "obróbka skosów".

Obrót

Komponent → frame, który definiuje obrót układu współrzędnych o określony kąt.

OEM

Dla producentów maszyn, którzy chcą sporządzać swoje własne interfejsy graficzne albo umieszczać w sterowaniu funkcje specyficzne dla technologii, są przewidziane przestrzenie dla indywidualnych rozwiązań (aplikacje OEM).

Ograniczenie obszaru pracy

Za pomocą ograniczenia obszaru pracy można dodatkowo względem wyłączników krańcowych ograniczyć zakres ruchu osi. Dla każdej osi jedna para wartości służy do opisu chronionej przestrzeni roboczej.

Osie

Osie CNC w zależności od swojej funkcji dzieli się na:

- Osie: osie interpolowane biorące udział w tworzeniu konturu.
- Osie pomocnicze: nieinterpolowane, osie dosuwu i pozycjonowania poruszające się ze specyficznym dla nich posuwem. Osie pomocnicze nie biorą udziału we właściwej obróbce, np. manipulatory podające narzędzia, magazyn narzędzi.

Osie maszyny

Osie fizycznie istniejące w obrabiarce.

Osie synchroniczne

Osie synchroniczne potrzebują dla przebycia swojej drogi takiego samego czasu co osie geometryczne dla swojego ruchu po torze.

Oś bazowa

Oś, której wartość zadana lub rzeczywista jest brana do obliczenia wartości kompensacji.

Oś C

Oś, wokół której następuje sterowany ruch obrotowy i pozycjonowanie za pomocą wrzeciona obrabianego przedmiotu.

Oś geometryczna

Osie geometryczne tworzą dwu- lub trójwymiarowy układ współrzędnych przedmiotu obrabianego, w którym to → w programach obróbki programowana jest geometria przedmiotu obrabianego.

Oś kompensacji

Oś, której wartość zadana albo rzeczywista jest modyfikowana przez wartość kompensacji.

Oś liniowa

Oś liniowa jest to oś, która w przeciwieństwie do osi obrotowej wykonuje ruch po prostej.

Oś obrotowa

Osie obrotowe powodują obrót obrabianego przedmiotu albo narzędzia do zadanego położenia kątowego.

Oś pozycjonowania

Oś, która wykonuje ruch pomocniczy na obrabiarce. (np. magazyn narzędzi, transport palet). Osie pozycjonowania są to osie, które nie interpolują z → osiami uczestniczącymi w tworzeniu konturu.

Oś uczestnicząca w tworzeniu konturu

Takimi osiami są wszystkie osie → kanału uczestniczące w obróbce, które są przez → interpolator tak prowadzone, że równocześnie rozpoczynają ruch, przyspieszają, zatrzymują się i osiągają punkt końcowy.

Oś wiodąca

Oś wiodąca jest to → oś gantry, która z punktu widzenia operatora i programisty istnieje, a przez to można odpowiednio na nią wpływać jak na normalną oś NC.

Oś zaokrągleniowa

Osie zaokrągleniowe powodują obrót obrabianego przedmiotu albo narzędzia do położenia kątowego odpowiadającego rastrowi podziałowemu. Po osiągnięciu rastra oś zaokrągleniowa jest "w pozycji".

Oś zsynchronizowana

Oś zsynchronizowana jest → osią gantry, której pozycja zadana jest stale wyprowadzana od ruchu postępowego → osi wiodącej, a przez to wykonuje ruch synchroniczny. Z punktu widzenia operatora i programisty oś zsynchronizowana "nie istnieje".

Pamięć korekcji

Obszar danych w sterowaniu, w którym są zapisane dane korekcyjne narzędzi.

Pamięć ładowania

Pamięć ładowania jest w przypadku sterowania CPU 314 → SPS równa → pamięci roboczej.

Pamięć programów PLC

SINUMERIK 840D sl: W pamięci użytkownika PLC program użytkownika PLC i dane użytkownika są zapisywane razem z programem podstawowym PLC.

Pamięć robocza

Pamięć robocza jest pamięcią RAM w -> CPU, do której procesor podczas wykonywania programu wczytuje program użytkownika.

Pamięć systemowa

Pamięć systemowa jest pamięcią w CPU, w której są zapisywane następujące dane:

- dane, których potrzebuje system operacyjny
- argumenty, czasy, liczniki, znaczniki

Pamięć użytkownika

Wszystkie programy i dane takie jak programy obróbki, podprogramy, komentarze, korekcje narzędzi, przesunięcia punktu zerowego / frame, jak też dane użytkownika dla kanału i programu mogą być zapisywane we wspólnej pamięci użytkownika w CNC.

Parametry R

Parametr obliczeniowy, może być przez programistę → programu obróbki ustawiany i odpytywany w programie dla dowolnych celów.

Peryferyjny zespół konstrukcyjny

Peryferyjne zespoły konstrukcyjne stanowią połączenie między CPU i procesem.

Peryferyjnymi zespołami konstrukcyjnymi są:

- → Cyfrowe zespoły wprowadzania/wyprowadzania
- → Analogowe zespoły wprowadzania/wyprowadzania
- → Symulatorowe zespoły konstrukcyjne

PLC

Programmable **L**ogic **C**ontrol: → Programowalny sterownik logiczny. Komponenty → NC: Sterowanie adaptacyjne do realizacji logiki kontrolnej obrabiarki.

Podprogram

Określenie "Podprogram" pochodzi z okresu, gdy programy obróbki były na stałe podzielone na → programy główne i podprogramy. Tego podziału już nie ma w dzisiejszym języku SINUMERIK NC. Zasadniczo każdy program obróbki lub każdy → cykl można wywołać w ramach innego programu obróbki, jako podprogram. Jest on wówczas wykonywany na następnym → poziomie programu (x+1) (poziom podprogramu (x+1)).

Podprogram asynchroniczny

Program obróbki, który może zostać uruchomiony asynchronicznie (niezależnie) do aktualnego stanu programu przez sygnał przerwania (np. sygnał "szybkie wejście NC").

Posuw po torze

Posuw po torze działa na → osie uczestniczące w tworzeniu konturu. Stanowi on sumę geometryczną posuwów uczestniczących → osi geometrycznych.

Posuw szybki

Najszybszy ruch w osi. Jest on np. stosowany, gdy narzędzie jest dosuwane z położenia spoczynkowego do → konturu obrabianego przedmiotu lub odsuwane od konturu. Prędkość posuwu szybkiego jest ustawiana specyficznie dla maszyny przez daną maszynową.

Posuw zależny od czasu

Zamiast prędkości posuwu można dla osi zaprogramować czas, którego wymaga posuw po torze w ramach bloku (G93).

Poziom programu

Program obróbki uruchomiony w kanale jest wykonywany jako → program główny na poziomie 0 (poziom programu głównego). Każdy program obróbki wywołany w programie głównym jest wykonywany jako → podprogram na własnym poziomie 1 ... n.

Półfabrykat

Część, od której jest rozpoczynana obróbka.

Prędkość ruchu po torze

Maksymalna możliwa do zaprogramowania prędkość ruchu po torze jest zależna od dokładności wprowadzanej wartości. Przy rozdzielczości np. 0,1 mm maksymalna możliwa do zaprogramowania prędkość ruchu po torze wynosi 1000 m/min.

Procedura przerwania

Procedury przerwania są specjalnymi → podprogramami, które mogą być uruchamiane przez zdarzenia (sygnały zewnętrzne) z procesu obróbki. Wykonywany blok programu obróbki jest przerywany, pozycja przerwania w osiach jest automatycznie zapisywana w pamięci.

Program główny

Określenie "Program główny" pochodzi z okresu, gdy programy obróbki były na stałe podzielone na programy główne i podprogramy. Tego podziału już nie ma w dzisiejszym języku SINUMERIK NC. Zasadniczo każdy program obróbki można wybrać i uruchamiać w danym kanale. Jest on wówczas wykonywany na → poziomie programu 0 (poziom programu głównego). W programie głównym mogą być jako podprogramy wywoływane dalsze programy obróbki lub cykle

Program obróbki

Ciąg poleceń pod adresem sterowania NC, które w sumie powodują wykonanie określonego → obrabianego przedmiotu. Również podjęcie określonej obróbki na danym → półfabrykacie.

Program przesyłania danych PCIN

PCIN jest programem pomocniczym do wysyłania i odbierania danych użytkownika CNC poprzez interfejs szeregowy, jak np. programy obróbki, korekcje narzędzi, itd. Program PCIN może być wykonywany pod MS-DOS na standardowym przemysłowym PC.

Program użytkownika

Programy użytkownika dla systemów automatyki S7-300 są sporządzane za pomocą języka programowania w STEP 7. Program użytkownika ma budowę modułową i składa się z poszczególnych bloków.

Podstawowymi typami bloków są:

- Bloki programowe
  Bloki te zawierają polecenia STEP 7.
- Bloki danych
  Bloki te zawierają stałe i zmienne dla programu STEP 7.

Programowane ograniczenie pola roboczego

Ograniczenie przestrzeni ruchów narzędzia do przestrzeni zdefiniowanej przez programowane ograniczenia.

Programowanie PLC

PLC jest programowane przy pomocy oprogramowania **STEP 7**. Oprogramowanie do programowania STEP 7 bazuje na standardowym systemie operacyjnym **WINDOWS** i zawiera innowacyjnie rozwinięte funkcje programowania STEP 5.

Przełącznik z kluczykiem

Przełącznik z kluczykiem na pulpicie sterowniczym maszyny posiada 4 położenia, które mają funkcje ustalone w systemie operacyjnym sterowania. Do przełącznika z kluczykiem należą trzy kluczyki o różnych kolorach, które można wyjąć w podanych położeniach

Przestrzeń ochronna

Przestrzeń trójwymiarowa w ramach → przestrzeni roboczej, w którą nie może wejść wierzchołek narzędzia.

Przestrzeń robocza

Przestrzeń trójwymiarowa, w której może poruszać się wierzchołek narzędzia, ze względu na konstrukcję obrabiarki. Patrz → Obszar chroniony.

Przesunięcie punktu zerowego

Zadanie nowego punktu odniesienia układu współrzędnych poprzez odniesienie do istniejącego punktu zerowego i → frame.

1. Ustawialne
   Każda oś CNC dysponuje zaprojektowaną liczbą ustawialnych przesunięć punktu zerowego. Przesunięcia wybrane za pomocą poleceń G działają alternatywnie.
2. Zewnętrzne
   Na wszystkie przesunięcia, które ustalają położenie punktu zerowego obrabianego przedmiotu, można dodatkowo nałożyć przesunięcie zewnętrzne przy pomocy kółka ręcznego (przesunięcie DRF) albo z PLC.
3. Programowalne
   Za pomocą polecenia `TRANS` można programować przesunięcia punktu zerowego dla wszystkich osi uczestniczących w tworzeniu konturu i osi pozycjonowania.

Przycisk programowy

Przycisk, którego napis jest reprezentowany przez pole na ekranie, które dynamicznie dopasowuje się do aktualnej sytuacji obsługowej. Dowolnie wykorzystywane przyciski funkcyjne (przyciski programowe) są przyporządkowywane do funkcji definiowanych w oprogramowaniu.

Przyrost

Podanie długości ruchu poprzez liczbę przyrostową (przyrost). Liczba przyrostowa może być zapisana jako → dana ustawcza lub wybrana przez odpowiednio opisane przyciski 10, 100, 1000, 10000.

Przyspieszenie z ograniczeniem szarpnięcia drugiego stopnia

W celu uzyskania optymalnego zachowania się przy przyspieszeniu na maszynie przy jednoczesnym oszczędzaniu mechaniki, można w programie obróbki przełączać między skokiem przyspieszenia i stałym przyspieszeniem (bez szarpnięcia).

Pulpit sterowniczy maszyny

Pulpit obsługi obrabiarki z elementami obsługi jak przyciski, przełączniki obrotowe itd. i z prostymi elementami sygnalizacyjnymi jak diody. Służy on do bezpośredniego wpływania na maszynę poprzez PLC.

Punkt odniesienia

Punkt obrabiarki, do którego odnosi się system pomiarowy → osi maszyny.

Punkt stały maszyny

Punkt jednoznacznie definiowany przez obrabiarkę, np. punkt referencyjny maszyny.

Punkt zerowy maszyny

Stały punkt obrabiarki, do którego można sprowadzić wszystkie systemy pomiarowe.

Punkt zerowy obrabianego przedmiotu

Punkt zerowy obrabianego przedmiotu tworzy punkt wyjściowy dla → układu współrzędnych obrabianego przedmiotu. Jest on definiowany przez odstępy od → punktu zerowego maszyny.

Sieć

Sieć jest połączeniem wielu układów S7-300 i innych urządzeń końcowych, np. PG, poprzez → kabel łączący. Wymiana danych między przyłączonymi urządzeniami następuje poprzez sieć.

Skalowanie

Komponent → frame, który powoduje specyficzne dla osi zmiany skali.

Słowa kluczowe

Słowa o ustalonej pisowni, które w języku programowania mają dla → programów obróbki zdefiniowane znaczenie.

Słowo danych

Jednostka danych o wielkości dwóch bajtów w ramach → modułu danych.

Softwareowe wyłączniki krańcowe

Softwareowe wyłączniki krańcowe ograniczają zakres ruchu w osi i zapobiegają najechaniu sań na sprzętowy wyłącznik krańcowy. Na każdą oś można zadać 2 pary wartości, które można oddzielnie uaktywniać przez → PLC.

Sterowanie prędkością

Aby przy ruchach postępowych móc uzyskać akceptowalną prędkość ruchu o bardzo krótkich przemieszczeniach, można ustawić wyprzedzające oszacowanie wielu bloków (→ Look Ahead).

Sterowanie programowane w pamięci

Sterowania programowane w pamięci (SPS) są to sterowania elektroniczne, których działanie jest zapisane jako program w urządzeniu sterującym. Budowa i okablowanie urządzenia nie zależą więc od jego funkcji. Sterowanie programowane w pamięci ma strukturę komputera; składa się ono z CPU (centralny zespół konstrukcyjny) z pamięcią, zespołów wejścia/wyjścia i wewnętrznego systemu przewodów magistralnych. Peryferia i język programowania są dostosowane do wymogów techniki sterowania.

Sterowanie wyprzedzające, dynamiczne

Niedokładności → konturu, uwarunkowane uchybem nadążania, dają się prawie całkowicie wyeliminować przez dynamiczne, zależne od przyśpieszenia sterowanie wyprzedzające. Przez to również przy wysokich → prędkościach po torze uzyskuje się znakomitą dokładność obróbki. Sterowanie wyprzedzające można wybrać i cofnąć specyficznie dla osi przez → program obróbki.

Synchronizacja

Instrukcje w → programach obróbki służące do koordynacji przebiegów w różnych → kanałach w określonych miejscach obróbki.

Szukanie bloku

W celu testowania programów obróbki albo po przerwaniu obróbki można poprzez funkcję "szukanie bloku" wybrać dowolne miejsce w programie obróbki, od którego obróbkę można uruchomić albo kontynuować.

Szybkie cofnięcie od konturu

Przy przybyciu przerwania można przez program obróbkowy CNC spowodować ruch, który umożliwi szybkie cofnięcie narzędzia od właśnie obrabianego konturu. Dodatkowo można zaprogramować kąt wycofania i wielkość drogi. Po szybkim cofnięciu można dodatkowo wykonać procedurę przerwania.

Szybkie cyfrowe wejścia/wyjścia

Poprzez wejścia cyfrowe mogą np. być uruchamiane szybkie procedury programowe CNC (procedury przerwania). Przez wyjścia cyfrowe CNC mogą być uruchamiane szybkie, sterowane przez program funkcje łączeniowe.

Szybkość transmisji

Szybkość przesyłania danych (bitów/s).

Tablica kompensacji

Tablica punktów oparcia. Dla wybranych pozycji osi bazowej wyznacza ona wartości kompensacji w osi kompensacji.

Technologia makr

Połączenie pewnej liczby poleceń pod jednym identyfikatorem. Identyfikator reprezentuje w programie zbiór połączonych poleceń.

Transformacja

Addytywne lub absolutne przesunięcie punktu zerowego osi.

Tryb pracy

Koncepcja przebiegu pracy sterowania SINUMERIK. Zdefiniowane są tryby pracy → JOG, → MDA, → automatyczny.

Tryb przechodzenia płynnego

Celem płynnego przechodzenia jest uniknięcie większego hamowania → osi uczestniczących w tworzeniu konturu na granicach bloków programu obróbki i przełączanie z możliwie taką samą prędkością ruchu po torze na następny blok programu.

Układ współrzędnych

Patrz → układ współrzędnych maszyny, → układ współrzędnych przedmiotu obrabianego

Układ współrzędnych maszyny

Układ współrzędnych, który jest odniesiony do osi obrabiarki.

Układ współrzędnych obrabianego przedmiotu

Układ współrzędnych obrabianego przedmiotu ma swój punkt początkowy w → punkcie zerowym obrabianego przedmiotu. Przy programowaniu w układzie współrzędnych obrabianego przedmiotu wymiary i kierunki odnoszą się do tego układu.

V.24

Interfejs szeregowy do wprowadzania/wyprowadzania danych. Przez ten interfejs mogą być ładowane i zapisywane programy obróbki, jak też dane producenta i użytkownika.

Wartość kompensacji

Różnica między zmierzoną przez czujnik pomiarowy pozycją osi a pozycją żądaną, zaprogramowaną.

WinSCP

WinSCP jest powszechnie dostępnym programem Open Source dla Windows do przesyłania plików.

Współrzędne biegunowe

Układ współrzędnych, który ustala położenie punktu na płaszczyźnie przez jego odległość od punktu zerowego i kąt, który tworzą promień wodzący i ustaloną oś.

Wymiar absolutny

Podanie celu ruchu w osi przez wartość wymiaru odnoszącego się do punktu zerowego aktualnie obowiązującego układu współrzędnych. Patrz → Wymiar przyrostowy.

Wymiar przyrostowy

Również wymiar przyrostowy: Podanie celu ruchu w osi poprzez będący do przebycia odcinek drogi i kierunek w odniesieniu do aktualnej pozycji. Patrz → Wymiar absolutny

Zakres ruchu

Maksymalny dopuszczalny zakres ruchu w przypadku osi liniowych wynosi ±9 dekad. Wartość bezwzględna jest zależna od wybranej dokładności wprowadzania i regulacji położenia oraz systemu jednostek (calowy albo metryczny).

Zakres TOA

Zakres TOA obejmuje wszystkie dane narzędzi i magazynów. Standardowo zakres wzgl. zasięg danych pokrywa się z zakresem → "Kanał". Przez dane maszynowe można jednak ustalić, że wiele kanałów będzie dzielić się jedną → jednostką TOA, tak że dla tych kanałów będą wówczas dostępne wspólne dane zarządzania narzędziami.

Zakrzywienie

Zakrzywienie k konturu jest odwrotnością promienia r przylegającego okręgu w punkcie konturu (k = 1/r).

Zarządzanie programami obróbki

Zarządzanie programami obróbki może być zorganizowane według → obrabianych przedmiotów. Wielkość pamięci użytkownika określa liczbę zarządzanych programów i danych. Każdy plik (programy i dane) można wyposażyć w nazwę o maksymalnie 24 znakach alfanumerycznych.

Zatrzymanie dokładne

Przy zaprogramowanym poleceniu zatrzymania dokładnego dosunięcie do pozycji podanej w bloku jest dokonywane dokładnie i ew. bardzo powoli. W celu redukcji czasu zbliżania są zdefiniowane dla posuwu szybkiego i posuwu → granice zatrzymania dokładnego.

Zewnętrzne przesunięcie punktu zerowego

Przesunięcie punktu zerowego zadane z → PLC.

Zmienna systemowa

Zmienna istniejąca bez udziału programisty programującego → program obróbki. Jest ona zdefiniowana przez typ danych i nazwę, która rozpoczyna się od znaku **$**. Patrz → Zmienne definiowane przez użytkownika.

Zmienne definiowane przez użytkownika

Użytkownicy mogą dla dla swoich potrzeb w → programie obróbki albo module danych (globalne dane użytkownika) tworzyć własne zmienne. Definicja zawiera typ danych i nazwę zmiennej. Patrz → Zmienne systemowe.

Zorientowane zatrzymanie wrzeciona

Zatrzymanie wrzeciona obrabianego przedmiotu w zadanym położeniu kątowym, np. aby w określonym miejscu przeprowadzić dodatkową obróbkę.

Zresetowanie całkowite

Przy zresetowaniu całkowitym są kasowane następujące pamięci → CPU

- → Pamięć robocza
- Obszar zapisu / odczytu → pamięci ładowania
- → Pamięć systemowa
- → Pamięć backup

Indeks

$

$AA_ATOL, 569
$AA_COUP_ACT
  Przy nadążaniu, 576
  Przy osiowym sprzężeniu wartości wiodącej, 600
$AA_ESR_ENABLE, 696
$AA_LEAD_SP, 600
$AA_LEAD_SV, 600
$AC_ACT_PROG_NET_TIME, 684
$AC_ACTUAL_PARTS, 687
$AC_AXCTSWA, 670
$AC_AXCTSWE, 670
$AC_CTOL, 569
$AC_CUT_INV, 490
$AC_CUTMOD, 490
$AC_CUTMOD_ANG, 490
$AC_CUTMODK, 490
$AC_CUTTING_TIME, 684
$AC_CYCLE_TIME, 684
$AC_DELAYFST, 549
$AC_ESR_TRIGGER, 696
$AC_OLD_PROG_NET_TIME, 684
$AC_OLD_PROG_NET_TIME_COUNT, 684
$AC_OPERATING_TIME, 684
$AC_PROG_NET_TIME_TRIGGER, 685
$AC_REPOS_PATH_MODE, 559
$AC_REQUIRED_PARTS, 687
$AC_SMAXVELO, 564
$AC_SMAXVELO_INFO, 564
$AC_SPECIAL_PARTS, 687
$AC_TOTAL_PARTS, 687
$AC_TRAFO_CORR_ELEM_P, 422
$AC_TRAFO_CORR_ELEM_T, 422
$AC_TRAFO_ORIAX_LOC, 422
$AN_AXCTAS, 670
$AN_AXCTSWA, 670
$AN_LANGUAGE_ON_HMI, 952
$AN_POWERON_TIME, 683
$AN_SETUP_TIME, 683
$NT_CLOSE_CHAIN_T, 422
$NT_CNTRL, 422
$NT_CORR_ELEM_P, 421
$NT_CORR_ELEM_T, 421
$NT_NAME, 413
$NT_ROT_AX_NAME, 487
$NT_TRAFO_INDEX, 413
$P_ACTBFRAME, 326
$P_AD, 489
$P_BFRAME, 325
$P_CHBFRAME, 325
$P_CHBFRMASK, 326
$P_CTOL, 569
$P_CUT_INV, 490
$P_CUTMOD, 490
$P_CUTMOD_ANG, 490
$P_CUTMOD_ERR, 491
$P_CUTMODK, 490
$P_DELAYFST, 549
$P_IFRAME, 327
$P_IS_EES_PATH, 232
$P_NCBFRAME, 325
$P_NCBFRMASK, 326
$P_ORI_DIFF, 482
$P_ORI_POS, 482
$P_ORI_SOL, 483
$P_ORI_STAT, 486
$P_OTOL, 569
$P_PATH, 231
$P_PFRAME, 327
$P_PROG, 231
$P_PROGPATH, 231
$P_SIM, 293
$P_STACK, 231
$P_SUBPAR, 168
$P_TOOLENV, 499
$P_TOOLENVN, 499
$PA_ATOL, 569
$SA_LEAD_TYPE, 600
$SC_CONTPREC, 542
$SC_MINFEED, 542
$SC_PA_ACTIV_IMMED, 244
$SN_PA_ACTIV_IMMED, 244
$TC_CARR1...14, 466
$TC_CARR18...23, 466
$TC_CARR18[m], 471
$TC_DP1 ... 25, 423
$TC_ECPxy, 427
$TC_SCPxy, 427

*

* (funkcja obliczeniowa), 79

/

/ (funkcja obliczeniowa), 79

+

+ (funkcja obliczeniowa), 79

<

< (operator porównania), 81
<< (operator powiązania), 91
<= (operator porównania), 81
<> (operator porównania), 81

=

== (operator porównania), 81

>

> (operator porównania), 81
>= (operator porównania), 81

A

ABS, 79
ACCLIMA, 537
ACOS, 79
ACTBLOCNO, 180
ACTFRAME, 301
ADISPOSA, 295
Adresowanie, 225
Adresy OEM, 293
Alarmy
    Ustawienie w programie NC, 694
Alarmy cykli, 694
ALF
    dla szybkiego wycofania od konturu, 140
AND, 81
APR, 48
APRB, 48
APRP, 48
APW, 48
APWB, 48
APWP, 48
AS, 217
ASIN, 79
ASPLINE, 258
A-Spline, 264
Asynchroniczny ruch wahadłowy, 633
ATAN2, 79
ATOL, 566
Atrybuty pozycji
    Programowanie pośrednie, 73
Automatyczny podział drogi, 652
Automatyczny wskaźnik przerwania, 552
AV, 609
AX, 661
AXCTSWE, 669
AXCTSWEC, 669
AXCTSWED, 669
AXIS, 31
AXNAME, 90
AXSTRING, 661
AXTOCHAN, 149
AXTOSPI, 661

B

B_AND, 81
B_NOT, 81
B_OR, 81
B_XOR, 81
BAUTO, 258
BFRAME, 301
BLOCK, 204
Blok zatrzymania, 551
BLSYNC, 134
BNAT, 258
BOOL, 31
BOUND, 85
BRISK, 535
BRISKA, 535
BSPLINE, 258
B-Spline, 265
BTAN, 258

C

CAC, 257
CACN, 257
CACP, 257
CALL, 203
CALLPATH, 207
CASE, 110
case-insensitiv, 225
CDC, 257
CFINE, 311

CHAN, 31
CHANDATA, 233
CHAR, 31
CHKDNO, 463
CIC, 257
Cięcie, 647
CLEARM, 127
CLRINT, 136
COARSE, 609
COARSEA, 295
COLLPAIR, 408
COMPCAD, 271
COMPCURV, 271
COMPLETE, 233
COMPOF, 271
COMPON, 271
COMPSURF, 271
CONTDCON, 716
CONTPRON, 710
CORRTRAFO, 415
COS, 79
COUPDEF, 609
COUPDEL, 609
COUPOF, 609
COUPOFS, 609
COUPON, 609
COUPONC, 609
COUPRES, 609
CP, 386
CPBC, 621
CPDEF, 620
CPDEL, 620
CPFMOF, 623
CPFMON, 623
CPFMSON, 622
CPFPOS + CPOF, 623
CPFPOS + CPON, 621
CPFRS, 621
CPLA, 620
CPLCTID, 621
CPLDEF, 620
CPLDEL, 620
CPLDEN, 621
CPLINSC, 625
CPLINTR, 625
CPLNUM, 621
CPLOF, 621
CPLON, 621
CPLOUTSC, 625
CPLOUTTR, 625
CPLPOS, 621
CPLSETVAL, 621
CPMALARM, 626
CPMBRAKE, 626
CPMPRT, 625
CPMRESET, 624
CPMSTART, 625
CPMVDI, 626
CPOF, 620
CPON, 620
CPRECOF, 542
CPRECON, 542
CPROT, 241
CPROTDEF, 237
CPSETTYPE, 626
CPSYNCOP, 625
CPSYNCOP2, 626
CPSYNCOV, 626
CPSYNFIP, 626
CPSYNFIP2, 626
CPSYNFIV, 626
CSPLINE, 258
C-Spline, 266
CTAB, 589
CTABDEF, 578
CTABDEL, 585
CTABEND, 578
CTABEXISTS, 585
CTABFNO, 594
CTABFPOL, 594
CTABFSEG, 594
CTABID, 587
CTABINV, 589
CTABISLOCK, 587
CTABLOCK, 586
CTABMEMTYP, 587
CTABMPOL, 594
CTABMSEG, 594
CTABNO, 594
CTABNOMEM, 594
CTABPERIOD, 587
CTABPOL, 594
CTABPOLID, 594
CTABSEG, 594
CTABSEGID, 594
CTABSEV, 589
CTABSSV, 589
CTABTEP, 589
CTABTEV, 589
CTABTMAX, 589
CTABTMIN, 589
CTABTSP, 589
CTABTSV, 589
CTABUNLOCK, 586

CTOL, 566
CTRANS, 311
CUT3DC, 442
CUT3DCC, 452
CUT3DCCD, 452
CUT3DCD, 442
CUT3DF, 446
CUT3DFD, 446
CUT3DFF, 446
CUT3DFS, 446
CUTMOD, 487
CUTMODK, 487
CYCLE4071
programowanie zewnętrzne, 833
CYCLE4072
programowanie zewnętrzne, 834
CYCLE4073
programowanie zewnętrzne, 838
CYCLE4074
programowanie zewnętrzne, 839
CYCLE4075
programowanie zewnętrzne, 842
CYCLE4077
programowanie zewnętrzne, 845
CYCLE4078
programowanie zewnętrzne, 848
CYCLE4079
programowanie zewnętrzne, 850
CYCLE435 - Ustawienie współrzędnych obciągacza
programowanie zewnętrzne, 794
CYCLE495 - Profilowanie
programowanie zewnętrzne, 794
CYCLE60 - Grawerowanie
programowanie zewnętrzne, 745
CYCLE61 - Frezowanie płaszczyzny
programowanie zewnętrzne, 747
CYCLE62 - Wywołanie konturu
programowanie zewnętrzne, 749
CYCLE63 - Frezowanie konturu kieszeni
programowanie zewnętrzne, 750
CYCLE64 - Wiercenie wstępne konturu kieszeni
programowanie zewnętrzne, 752
CYCLE70 - Frezowanie gwintu
programowanie zewnętrzne, 754
CYCLE72 - Frezowanie konturu
programowanie zewnętrzne, 756
CYCLE76 - Czop prostokątny
programowanie zewnętrzne, 760
CYCLE77 - Czop kołowy
programowanie zewnętrzne, 762
CYCLE78 - Wiercenie otworu z frezowaniem gwintu
programowanie zewnętrzne, 764
CYCLE79 - Wielobok
programowanie zewnętrzne, 767
CYCLE800 - Skręt
programowanie zewnętrzne, 796
CYCLE801 - szablon pozycji siatka/ramka
programowanie zewnętrzne, 800
CYCLE802 - Dowolne pozycje
programowanie zewnętrzne, 801
CYCLE81 - nawiercanie
programowanie zewnętrzne, 769
CYCLE82 - Wiercenie
programowanie zewnętrzne, 770
CYCLE83 - Wiercenie otworu głębokiego
programowanie zewnętrzne, 773
CYCLE830 - Wiercenie otworu głębokiego 2
programowanie zewnętrzne, 804
CYCLE832 - High Speed Settings
programowanie zewnętrzne, 810
CYCLE84 - Gwintowanie otworu bez oprawki kompensacyjnej
programowanie zewnętrzne, 776
CYCLE840 - Gwintowanie otworu z oprawką kompensacyjną
programowanie zewnętrzne, 813
CYCLE85 - Rozwiercanie
programowanie zewnętrzne, 779
CYCLE86 - Wytaczanie
programowanie zewnętrzne, 780
CYCLE899 - Frezowanie rowka otwartego
programowanie zewnętrzne, 816
CYCLE92 - Przecięcie
programowanie zewnętrzne, 781
CYCLE930 - Rowek
programowanie zewnętrzne, 819
CYCLE940 - Podcięcie
programowanie zewnętrzne, 822
CYCLE95 - Skrawanie konturu
programowanie zewnętrzne, 783
CYCLE952 - Toczenie wcinające konturu
programowanie zewnętrzne, 827
CYCLE98 - Łańcuch gwintów
programowanie zewnętrzne, 785
CYCLE99 - Toczenie gwintu
programowanie zewnętrzne, 789
Czas obróbki, 684
Czas przebiegu
Zachowanie się struktur kontrolnych, 119
Część programu
Powtórzenie, 112
Czop kołowy - CYCLE77
programowanie zewnętrzne, 762

Czop prostokątny - CYCLE76
programowanie zewnętrzne, 760

D

Dane okręgu
Obliczenie, 723
DCI, 53
DCM, 53
DCU, 53
DEF, 31
DEFAULT, 110
DEFINE ... AS, 217
Definicja półfabrykatu, 703
DELAYFSTOF, 547
DELAYFSTON, 547
DELDL, 428
DELETE, 156
DELOBJ, 401
DELTOOLENV, 497
DISABLE, 136
DISPLOF, 180
DISPLON, 180
DISPR, 553
DIV, 79
DL, 426
DO, 631
Dokładność konturu
programowalna, 542
Dostępność
Zależne od systemu, 5
Dowolne pozycje - CYCLE802
programowanie zewnętrzne, 801
DRIVE, 535
DRIVEA, 535
DV, 609
DYNFINISH, 539
DYNNORM, 539
DYNPOS, 539
DYNROUGH, 539
DYNSEMIFIN, 539

E

Easy XML, 678
EAUTO, 258
EES, 223
EG
Przekładnia elektroniczna, 602
EGDEF, 602
EGDEL, 607
EGOFC, 606
EGOFS, 606
EGON, 603
EGONSYN, 603
EGONSYNE, 603
Element konturu
Odsunięcie, 722
ELSE, 120
ENABLE, 136
ENAT, 258
ENDFOR, 122
ENDIF, 120
ENDLABEL, 112
ENDLOOP, 122
ENDWHILE, 124
ESR, 695
ESRR, 701
ESRS, 701
ETAN, 258
Etykieta, 112
EVERY, 631
EXECSTRING, 76
EXECTAB, 722
EXECUTE, 725
EXP, 79
EXTCALL
dla SINUMERIK 828D, 212
dla SINUMERIK 840D sl, 208
EXTCLOSE, 689
EXTERN, 198
EXTOPEN, 689

F

FALSE, 31
FCTDEF, 438
FCUB, 530
FENDNORM, 294
FFWOF, 541
FFWON, 541
FIFOCTRL, 544
FILEDATE, 160
FILEINFO, 160
FILESIZE, 160
FILESTAT, 160
FILETIME, 160
FINE, 609
FINEA, 295
FLIN, 530
FNORM, 530
FOR, 122
FPO, 530

FRAME, 31
Globalne dla NCU,
Łańcuchy frame,
-powiązanie,
Przyporządkowanie,
Specyficzne dla kanału,
Systemowe-,
Wywołanie,
Frame systemowe, 324
Frezowanie czołowe, 351
Frezowanie gwintu - CYCLE70
programowanie zewnętrzne, 754
Frezowanie konturu - CYCLE72
programowanie zewnętrzne, 756
Frezowanie konturu kieszeni – CYCLE63
programowanie zewnętrzne, 750
Frezowanie płaszczyzny - CYCLE61
programowanie zewnętrzne, 747
Frezowanie rowka otwartego - CYCLE899
programowanie zewnętrzne, 816
FROM, 631
FTOCOF, 441
FTOCON, 441
Funkcje OEM, 293

G

G290, 707
G291, 707
G5, 382
G62, 294
G621, 294
G7, 382
G810 ... G819, 293
G820 ... G829, 293
GEOAX, 664
GET, 144
GETACTTD, 465
GETD, 144
GETDNO, 464
GETTCOR, 499
GETTENV, 498
GETVARAP, 65
GETVARDFT, 67
GETVARDIM, 66
GETVARLIM, 66
GETVARPHU, 64
GETVARTYP, 68
Globalna pamięć programów obróbki (GDIR), 223
Głębokość kaskadowania
Struktury kontrolne, 119
Głębokość zagłębienia, 445
GOTO, 107
GOTOB, 107
GOTOC, 107
GOTOF, 107
GOTOS, 106
GP, 73
Grawerowanie (CYCLE60)
programowanie zewnętrzne, 745
GROUP_ADDEND - Koniec dodatkowego dosuwu
programowanie zewnętrzne, 853
GROUP_BEGIN - Początek bloku programu
programowanie zewnętrzne, 852
GROUP_END - Koniec bloku programu
programowanie zewnętrzne, 853
Grupa G
Technologia, 539
GUD, 32
Gwintowanie otworu bez oprawki kompensacyjnej - CYCLE84
programowanie zewnętrzne, 776
Gwintowanie otworu z oprawką kompensacyjną - CYCLE840
programowanie zewnętrzne, 813

H

High Speed Settings - CYCLE832
programowanie zewnętrzne, 810
HOLES1 - Szablon pozycji Linia
programowanie zewnętrzne, 730
HOLES2 - Szablon pozycji Okrąg
programowanie zewnętrzne, 730

I

ID, 631
IDS, 631
IF, 120
IFRAME, 301
Indeks tablicy, 58
INDEX, 94
INICF, 31
Inicjalizacja
Tablic, 55
INIPO, 31
INIRE, 31
INIT, 127
INITIAL, 233
INITIAL_INI, 233
INT, 31
Interpolacja wektora obrotu, 364

Interpolacja wielomianowa, 272
INTERSEC, 720
IPOBRKA, 295
IPOENDA, 295
IPOSTOP, 609
IPTRLOCK, 550
IPTRUNLOCK, 550
ISAXIS, 661
ISFILE, 159
ISNUMBER, 90
ISOCALL, 206
ISVAR, 62

J

JERKLIM, 562
JERKLIMA, 537

K

Kąt Eulera, 348
Kąt RPY, 348
Kieszeń kołowa - POCKET4
  programowanie zewnętrzne, 735
Kieszeń prostokątna - POCKET3
  programowanie zewnętrzne, 732
Kinematyka
  Rozłączona, 471
Klasa danych, 53
Kody G
  Programowanie pośrednie, 72
Komponenty frame
  FI, 307
  MI, 307
  RT, 307
  SC, 307
  TR, 307
Koniec bloku programu - GROUP_END
  programowanie zewnętrzne, 853
Koniec dodatkowego dosuwu - GROUP_ADDEND
  programowanie zewnętrzne, 853
Kontur
  dosunięcie przywracające, 553
  Kodowanie, 716
  Przygotowanie, 710
  -tablica, 710
Korekcja długości narzędzia online, 475
Korekcja narzędzia
  Pamięć korekcji, 423
  Układ współrzędnych dla wartości zużycia, 433
Korekcja promienia narzędzia
  Zwłoka na narożnikach, 294
Korekcje narzędzi
  addytywne, 426
Kryterium końca ruchu
  Programowane, 295

L

L..., 196
LEAD, 345
LEADOF, 596
LEADON, 596
LENTOAX, 519
LIFTFAST, 137
LLI, 44
LN, 79
LONGHOLE - Otwór podłużny
  programowanie zewnętrzne, 743
LOOP, 122
LUD, 32

Ł

Łańcuch gwintów - CYCLE98
  programowanie zewnętrzne, 785
Łańcuch narzędzia, 420
Łańcuch przedmiotu obrabianego, 420
Łańcuch znaków
  Formatowanie, 97
  Operacje, 89
  Powiązanie, 91
Łańcuchy znaków
  Długość, 93

M

M17, 184
M30, 184
Makro, 217
MASLDEF, 628
MASLDEL, 628
MASLOF, 628
MASLOFS, 628
MASLON, 628
MATCH, 94
MAXVAL, 85
MCALL, 201
MD10010, 127
MD10280, 127
MD15800, 28

MD18104, 496
MD18116, 497
MD18156, 28
MD20360, 504
MD24558, 506
MD24658, 506
MEAC, 283
MEAFRAME, 319
MEAS, 280
MEASA, 283
MEAW, 280
MEAWA, 283
Miejsca osobliwe, 353
MINDEX, 94
MINVAL, 85
MMC, 678
MOD, 79
MODAXVAL, 661
MPF, 222

N

Nadążanie, 573
NAMETOINT, 404
Narzędzie
  -orientacja, 457
  Orientacja przy zmianie frame, 474
  parametry, 423
Nawiercanie - CYCLE81
  programowanie zewnętrzne, 769
Nazwa dysku, 227
Nazwa pliku, 229
NCK, 31
NEWCONF, 151
NOC, 609
Nośnik narzędzi
  Kinematyka, 466
  -Orientowalny, 472
NOT, 81
Notacja EES, 225
Notacja NCK, 225
NPROT, 241
NPROTDEF, 237
NUMBER, 90
Numer D
  Dowolne nadanie, 463
Numer ostrza, 463
Numery D
  Sprawdzenie, 463
  Zmiana nazwy, 464
NUT, 356

O

Obrabiany przedmiot
  Katalog główny, 222
  Katalogi, 222
  Licznik, 687
Obrót
  wektora orientacji, 364
Obszary ochrony, 237
OEMIPO1/2, 293
OMA1 ... OMA5, 293
Operatory logiczne, 81
Operatory porównania, 81
OR, 81
ORIAXES, 354
ORIC, 457
ORICONCCW, 356
ORICONCW, 356
ORICONIO, 356
ORICONTO, 356
ORICURVE, 359
ORID, 457
Orientacja narzędzia
  W stosunku do toru ruchu, 367
Orientacja narzędzia odniesiona do toru, 367
Orientowane nośniki narzędzi, 466
ORIEULER, 354
ORIMKS, 352
ORIPATH, 368
ORIPATHS, 368
ORIPLANE, 356
ORIRESET(A, B, C), 343
ORIROTA, 364
ORIROTC
  przy interpolacji obrotu narzędzia, 370
  przy obrocie orientacji narzędzia, 364
ORIROTR, 364
ORIROTT, 364
ORIRPY, 354
ORIRPY2, 354
ORIS, 457
ORISOF, 376
ORISOLH, 478
ORISON, 376
ORIVECT, 354
ORIVIRT1, 354
ORIVIRT2, 354
ORIWKS, 352
OS, 633
OSB, 633
OSC, 457

OSCILL, 639
OSCTRL, 633
OSD, 457
OSE, 633
Osie
  Nadążanie, 575
Osie nadążne, 575
Osie obrotu
  Kąt skrętu, 466
  Wektory kierunku, 466
  Wektory odległości, 466
Osie orientacji, 354
Osiowe sprzężenie wartości wiodącej, 596
OSNSC, 633
OSOF, 457
OSP1, 633
OSP2, 633
OSS, 457
OSSE, 457
OST, 457
OST1, 633
OST2, 633
Oś
  Zamiana, 144
Oś geometryczna
  Przełączenie, 664
Oś nadążna
  Przy osiowym sprzężeniu wartości wiodącej, 596
Oś wiodąca
  Przy osiowym sprzężeniu wartości wiodącej, 596
OTOL, 566
Otwór podłużny - LONGHOLE
  programowanie zewnętrzne, 743

P

P..., 200
P_ACTFRAME, 327
Pamięć
  Pamięć, 221
  Przebieg wyprzedzający, 544
  Robocza, 233
Pamięć korekcji, 423
Pamięć programów
  Katalogi standardowe, 222
  Typy plików, 222
Pamięć robocza, 233
Parametry
  Aktualne, 167
  Formalne, 166
  Narzędzie, 423
  Przekazanie przy wywołaniu podprogramu, 167, 198
Parametry obliczeniowe
  Globalny, 27
  Specyficzne dla kanału, 26
PCALL, 206
PDELAYOF, 647
PDELAYON, 647
Pętla bez końca, 122
Pętla FOR, 122
Pętla programowa
  Pętla bez końca, 122
  Pętla FOR, 122
  Pętla IF, 120
  Pętla REPEAT, 125
  Pętla WHILE, 124
PFRAME, 301
PHI
  przy orientacji wzdłuż pobocznicy stożka, 356
  Wielomiany orientacji, 363
PHU, 45
PL
  przy interpolacji wielomianowej, 272
  W przypadku interpolacji Spline, 258
Plik
  Informacje, 160
PO, 272
PO[PHI]
  przy obrocie orientacji narzędzia, 368
  przy orientacji wzdłuż pobocznicy stożka, 356
  Wielomiany orientacji, 362
PO[PSI]
  przy obrocie orientacji narzędzia, 368
  przy orientacji wzdłuż pobocznicy stożka, 356
  Wielomiany orientacji, 362
PO[THT]
  przy obrocie orientacji narzędzia, 368
  Wielomiany orientacji, 362
PO[XH]
  przy zadaniu orientacji dwóch punktów styku, 359
  Wielomiany orientacji, 363
PO[YH]
  przy zadaniu orientacji dwóch punktów styku, 359
  Wielomiany orientacji, 363
PO[ZH]
  przy zadaniu orientacji dwóch punktów styku, 359
  Wielomiany orientacji, 363
POCKET3 - Kieszeń prostokątna
  programowanie zewnętrzne, 732
POCKET4 - Kieszeń kołowa
  programowanie zewnętrzne, 735

Początek bloku programu - GROUP_BEGIN, 852
Podanie ścieżki, 226
Podcięcie - CYCLE940
programowanie zewnętrzne, 822
Podprogram
Nazwa, 164
-powrót, parametryzowany (RETB...), 192
Powtórzenie, 200
Programowana ścieżka szukania, 207
-skok powrotny, parametryzowany (RET ...), 186
Wywołanie bez przekazania parametrów, 196
Wywołanie z przekazaniem parametrów, 198
-wywołanie, modalne, 201
-wywołanie, pośrednie, 203
Zastosowanie, 163
Podprogramy DIN, 229
POLF
dla wycofania prowadzonego przez NC, 696
POLFA, 696
POLFMASK
dla wycofania prowadzonego przez NC, 696
POLFMLIN
dla wycofania prowadzonego przez NC, 696
POLY, 272
POLYPATH, 272
Położenie podstawowe orientacji narzędzia
ORIRESET, 344
PON, 656
PONS, 647
POSFS, 609
POT, 79
Powiązanie
Łańcuchy znaków, 91
Powtórzenie części programu
Z programowaniem pośrednim CALL, 204
Pozostały czas
Dla obrabianego przedmiotu, 685
PREPRO, 183
PRESETON, 315
PRESETONS, 317
PRIO, 134
PRLOC, 31
Procedura przerwania
Kasowanie, 136
Nowe przyporządkowanie, 135
Programowany kierunek ruchu, 140
Ruch wycofania, 139
Szybkie cofnięcie od konturu, 137
Wyłączenie/włączenie, 136
Process DataShare, 689
Profilowanie - CYCLE495
programowanie zewnętrzne, 794
Program
Czasy przebiegu, 683
Inicjalizacyjny, 233
Pamięć, 222
Powtórzenie, 200
programu, 225
Rozgałęzienie, 110
Skoki, 107
Program inicjalizacyjny, 233
Programowanie orientacji, 354
Programowanie pośrednie
Adresów, 70
atrybutów pozycji, 73
Kody G, 72
wierszy programów obróbki, 76
programowanie zewnętrzne, 852
PROTA, 409
PROTD, 411
PROTS, 410
Przebieg wyprzedzający
Pamięć, 544
Przecięcie - CYCLE92
programowanie zewnętrzne, 781
Przekładnia elektroniczna, 602
Przełączane osie geometryczne, 664
Przełączanie trybu językowego, 707
Przesunięcie punktu zerowego
Zewnętrzne, 313
Przygotowanie konturu
Zwrotna sygnalizacja błędu, 725
Przyspieszenie drugiego stopnia
Korekcja, 562
Ograniczenie, 535
PSI
przy orientacji wzdłuż pobocznicy stożka, 356
Wielomiany orientacji, 363
PTP, 386
PTPG0, 386
PTPWOC, 386
PUD, 32
PUNCHACC, 647
Punkt obróbki frezem, 445
Punkt pomocniczy frezu, 445
PUTFTOC, 440
PUTFTOCF, 439
PW, 258

R

READ, 157
REAL, 31
REDEF, 37

RELEASE, 144
REP, 55
REPEAT, 112
REPEATB, 112
REPOSA, 553
REPOSH, 553
REPOSHA, 553
REPOSL, 553
REPOSQ, 553
REPOSQA, 553
RET, 185
RET (parametryzowany), 186
RETB (parametryzowane), 192
RG, 27
RINDEX, 94
RMBBL, 553
RMEBL, 553
RMIBL, 553
RMNBL, 553
Rodzaj sprzężenia, 612
Rodzaje transformacji
  Funkcja ogólna, 331
ROUND, 79
ROUNDUP, 162
Rowek - CYCLE930
  programowanie zewnętrzne, 819
Rowek kołowy - SLOT2
  programowanie zewnętrzne, 740
Rowek wzdłużny - SLOT1
  programowanie zewnętrzne, 738
Rozwiercanie - CYCLE85
  programowanie zewnętrzne, 779
Ruch kartezjański PTP, 386
Ruch punkt do punktu, 386
Ruch synchroniczny
  dokładnie, 612
  Zgrubnie, 612
Ruch wahadłowy
  Asynchroniczny, 633
  Asynchroniczny ruch wahadłowy, 633
  Dosuw częściowy, 642
  Dosuw w punkcie nawrotu, 644
  Obszar nawrotu, 642
  Punkt nawrotu, 642
  Sterowanie przez akcję synchroniczną, 639
  Synchroniczny ruch wahadłowy, 639
Run MyScreens, 678

S

SAVE, 173
SBLOF, 175
SBLON, 175
SCPARA, 672
SD, 258
SD41610, 422
SD41611, 422
SD42475, 374
SD42476, 374
SD42477, 374
SD42900, 432
SD42910, 432
SD42920, 432
SD42930, 433
SD42935, 435
SD42940, 436
SD42984, 489
SET, 55
SETAL, 694
SETDNO, 464
SETINT, 134
SETM, 127
SETTCOR, 506
SIN, 79
Skasowanie pozostałej drogi, 289
Skok
  Do początku programu, 106
  do znaczników skoku, 107
Skrawanie
  Funkcje wspomagające, 709
Skrawanie konturu - CYCLE95
  programowanie zewnętrzne, 783
Skrawanie warstwowe - CYCLE951
  programowanie zewnętrzne, 824
Skręt - CYCLE800
  programowanie zewnętrzne, 796
SLOT1 - rowek wzdłużny
  programowanie zewnętrzne, 738
SLOT2 - Rowek kołowy
  programowanie zewnętrzne, 740
SOFT, 535
SOFTA, 535
SON, 647
SONS, 647
SPF, 222
SPI, 661
SPIF1, 647
SPIF2, 647
Spline
  Interpolacja, 258
  Typy, 264
SPLINEPATH, 269
SPN, 652
SPOF, 647

SPP, 652
SPRINT, 97
Sprzężenie
  Rodzajowe, 620
Sprzężenie prędkości, 612
Sprzężenie wartości rzeczywistej, 612
Sprzężenie wartości wiodącej
  Sprzężenie wartości rzeczywistej i zadanej, 599
  Synchronizacja osi wiodącej i osi nadążnej, 598
Sprzężenie wartości zadanej, 612
SQRT, 79
START, 127
STARTFIFO, 544
STAT, 387
Status sprzężenia
  Przy nadążaniu, 576
  Przy osiowym sprzężeniu wartości wiodącej, 600
Status zlecenia pomiarowego, 291
STOPFIFO, 544
STOPRE, 544
STRING, 31
STRINGIS, 674
STRLEN, 93
Struktury
  kontrolne, 119
SUBSTR, 95
Symulacja wartości wiodącej, 600
Synchroniczność liczby obrotów, 609
Synchroniczny ruch wahadłowy
  Akcje synchroniczne, 643
  Dosuw w obszarze nawrotu, 643
  Ewaluacja, takt IPO, 645
  Następny dosuw częściowy, 645
  Przyporządkowanie osi ruchu wahadłowego i osi dosuwu, 642
  Ruch dosuwu, 643
  Ustalenie dosuwów, 642
Synchronizacja położenia, 609
Synchronizacja położenia z zależnością kątową, 609
SYNR, 31
SYNRW, 31
SYNW, 31
System
  Zależna dostępność, 5
Szablon pozycji Linia (HOLES1)
  programowanie zewnętrzne, 730
Szablon pozycji Okrąg - HOLES2
  programowanie zewnętrzne, 730
Szablon pozycji siatka/ramka - CYCLE801
  programowanie zewnętrzne, 800
Szlifowanie wcinające skośne, 382
Szybkie cofnięcie od konturu, 137
Ścieżka katalogu, 228
Ścieżka szukania
  Programowana ścieżka szukania, 207
  Przy wywołaniu podprogramu, 230

T

Tablica, 55
  -definicja, 55
  Element, 55
TAN, 79
TANG, 523
TANGDEL, 527
TANGOF, 527
TANGON, 526
TCARR, 472
TCOABS, 472
TCOFR, 472
TCOFRX, 472
TCOFRY, 472
TCOFRZ, 472
THETA
  przy interpolacji obrotu narzędzia, 370
  przy obrocie orientacji narzędzia, 364
TILT, 345
TLIFT, 525
Tłoczenie, 647
TMOF, 659
TMON, 659
Toczenie gwintu - CYCLE99
  programowanie zewnętrzne, 789
Toczenie wcinające konturu - CYCLE952
  programowanie zewnętrzne, 827
TOFFOF, 475
TOFFON, 475
TOLOWER, 92
TOOLENV, 494
TOUPPER, 92
TOWBCS, 433
TOWKCS, 433
TOWMCS, 433
TOWSTD, 433
TOWTCS, 433
TOWWCS, 433
TRAANG
  z programowalnym katem, 381
TRACON, 384
TRACYL, 378
TRAFOOF, 399
TRAFOON, 413
TRAILOF, 573
TRAILON, 573

Transformacja biegunowa, 335
Transformacja kąta skośnego (TRAANG)
z programowalnym katem, 381
Transformacja orientacji TRAORI
Kinematyka maszyny, 334
Programowanie orientacji, 343
Ruchy orientacji, 333
Transformacja rodzajowa 5/6-osiowa, 334
Warianty programowania orientacji, 343
Transformacja pobocznicy walca, 335
Transformacja ze skrętną osią liniową, 341
Transformacje
Niezależne od kinematyki położenie podstawowe orientacji narzędzia, 332
Powiązane, 384
Transformacja orientacji, 331
Transformacja trzy-, cztero- i pięcioosiowa, 342
Transformacje kinematyczne, 332
Transformacje powiązane, 333
TRANSMIT, 378
TRAORI, 342
TRUE, 31
TRUNC, 79
Tryb pracy
Przy pomiarze, 289
Tryb przyspieszenia, 535
TU, 391
Typ kinematyki, 471

U

Ujęcie i szukanie obszarów niezdatnych do przeszukiwania, 551
ULI, 44
UNTIL, 125
Ustawienie współrzędnych obciągacza - CYCLE435
programowanie zewnętrzne, 794
Użytkownik XML, 678

V

VELOLIM, 563
VELOLIMA, 537

W

WAITC, 609
WAITE, 127
WAITENC, 671
WAITM, 127
WAITMC, 127
Wartość ustawcza, 427
Wartość zużycia, 427
Warunki brzegowe przy transformacjach, 398
Wektor kierunkowy, 348
Wektor orientacji THETA, 364
WHEN, 631
WHEN-DO, 643
WHENEVER, 631
WHENEVER-DO, 643
WHILE, 124
Wielobok - CYCLE79
programowanie zewnętrzne, 767
Wielomian mianownikowy, 276
Wiercenie - CYCLE82
programowanie zewnętrzne, 770
Wiercenie otworu głębokiego - CYCLE83
programowanie zewnętrzne, 773
Wiercenie otworu głębokiego 2 - CYCLE830
programowanie zewnętrzne, 804
Wiercenie otworu z frezowaniem gwintu - CYCLE78
programowanie zewnętrzne, 764
Wiercenie wstępne konturu kieszeni - CYCLE64
programowanie zewnętrzne, 752
Wierzchołek frezu, 445
WORKPIECE, 703
WRITE, 152
Wrzeciono
Zamiana, 144
Wrzeciono synchroniczne
Sprzężenie, 609
Ustalenie pary, 615
Współczynnik sprzężenia, 573
Współczynnik wielomianu, 273
Wycofanie
Prowadzone przez NC, 696
wystarczające dla napędu, 701
Wygładzanie
Przebieg orientacji, 376
Wykonywanie pojedynczymi blokami
Blokowanie, 175
Wyprowadzenie
na zewnętrznym urządzeniu/pliku, 689
Wytaczanie - CYCLE86
programowanie zewnętrzne, 780
Wywołanie konturu - CYCLE62
programowanie zewnętrzne, 749
Wywołanie podprogramu z podaniem ścieżki i parametrami, 206

X

XOR, 81

Z

Zablokowanie
wyświetlania bloków, 180
Zaokrąglenie do góry, 162
Zatrzymanie
Prowadzone przez NC, 700
wystarczające dla napędu, 701
Zdarzenie przerzutnikowe
Przy pomiarze, 288
Zespół nadążania, 573
Zespół spline, 269
Zewnętrzne przesunięcie punktu zerowego, 313
Zmienna
Konwersja typów, 88
Zmienna frame
Predefiniowana zmienna frame, 301
Przyporządkowanie wartości, 305
Wywoływanie transformacji współrzędnych, 299
Zmienne
Definiowane przez użytkownika, 31
Konwersja typu, 89
link, 29
Zmienne systemowe
Ograniczenie sondy pomiarowej, 291
Status sondy pomiarowej, 291
Znacznik skoku
przy powtórzeniach części programu, 112
W przypadku skoków w programie, 108
Znak 0, 89
Zwłoka na narożnikach wewnętrznych, 294
Zwłoka na wszystkich narożnikach, 294