

SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D Arbeitsvorbereitung

Programmierhandbuch

Gültig für

Steuerung
SINUMERIK 840D sl / 840DE sl
SINUMERIK 828D

Software
CNC-Software

Version
4.7 SP2


10/2015
6FC5398-2BP40-5AA3


Vorwort	
Grundlegende Sicherheitshinweise	1
Flexible NC-Programmierung	2
Datei- und Programmverwaltung	3
Schutzbereiche	4
Spezielle Wegbefehle	5
Koordinatentransformationen (Frames)	6
Transformationen	7
Kinematische Ketten	8
Kollisionsvermeidung mit kinematischen Ketten	9
Werkzeugkorrekturen	10
Bahnverhalten	11
Achskopplungen	12
Synchronaktionen	13
Pendeln	14
Stanzen und Nibbeln	15
Schleifen	16
Weitere Funktionen	17
Eigene Abspanprogramme	18
Zyklen extern programmieren	19
Tabellen	20
Anhang	A


Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten wird , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten kann , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 VORSICHT
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.


Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Vorwort

SINUMERIK-Dokumentation

Die SINUMERIK-Dokumentation ist in folgende Kategorien gegliedert:

- Allgemeine Dokumentation
- Anwender-Dokumentation
- Hersteller/Service-Dokumentation

Weiterführende Informationen

Unter dem Link www.siemens.com/motioncontrol/docu finden Sie Informationen zu folgenden Themen:

- Dokumentation bestellen / Druckschriftenübersicht
- Weiterführende Links für den Download von Dokumenten
- Dokumentation online nutzen (Handbücher/Informationen finden und durchsuchen)

Bei Fragen zur Technischen Dokumentation (z. B. Anregungen, Korrekturen) senden Sie bitte eine E-Mail an folgende Adresse:

docu.motioncontrol@siemens.com

My Documentation Manager (MDM)

Unter folgendem Link finden Sie Informationen, um auf Basis der Siemens Inhalte eine OEM-spezifische Maschinen-Dokumentation individuell zusammenzustellen:

www.siemens.com/mdm

Training

Informationen zum Trainingsangebot finden Sie unter:

- www.siemens.com/sitrain
SITRAIN - das Training von Siemens für Produkte, Systeme und Lösungen der Automatisierungstechnik
- www.siemens.com/sinustrain
SinuTrain - Trainingssoftware für SINUMERIK

FAQs

Frequently Asked Questions finden Sie in den Service&Support Seiten unter Produkt Support.
<http://support.automation.siemens.com>

SINUMERIK

Informationen zu SINUMERIK finden Sie unter folgendem Link:

www.siemens.com/sinumerik

Zielgruppe

Die vorliegende Druckschrift wendet sich an:

- Programmierer
- Projektueure

Nutzen

Das Programmierhandbuch befähigt die Zielgruppe, Programme und Software-Oberflächen zu entwerfen, zu schreiben, zu testen und Fehler zu beheben.

Standardumfang

In dem vorliegenden Programmierhandbuch ist die Funktionalität des Standardumfangs beschrieben. Ergänzungen oder Änderungen, die durch den Maschinenhersteller vorgenommen werden, werden vom Maschinenhersteller dokumentiert.

Es können in der Steuerung weitere, in dieser Dokumentation nicht erläuterte Funktionen ablauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei der Neulieferung bzw. im Servicefall.

Ebenso enthält diese Dokumentation aus Gründen der Übersichtlichkeit nicht sämtliche Detailinformationen zu allen Typen des Produkts und kann auch nicht jeden denkbaren Fall der Aufstellung, des Betriebs und der Instandhaltung berücksichtigen.

Technical Support

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet unter <http://www.siemens.com/automation/service&support>

Informationen zu Struktur und Inhalt

Programmierhandbuch Grundlagen / Arbeitsvorbereitung

Die Beschreibungen zur NC-Programmierung sind auf zwei Handbücher verteilt:

1. Grundlagen

Das Programmierhandbuch "Grundlagen" dient dem Maschinenfacharbeiter und setzt entsprechende Kenntnisse für Bohr-, Fräs- und Drehbearbeitungen voraus. An einfachen Programmierbeispielen werden die auch nach DIN 66025 bekannten Befehle und Anweisungen erläutert.

2. Arbeitsvorbereitung

Das Programmierhandbuch "Arbeitsvorbereitung" dient dem Technologen mit Kenntnissen über die gesamten Programmiermöglichkeiten. Die SINUMERIK-Steuerung ermöglicht mit einer speziellen Programmiersprache die Programmierung eines komplexen Werkstückprogramms (z. B. Freiformflächen, Kanalkoordinierung, ...) und erleichtert dem Technologen eine aufwendige Programmierung.

Verfügbarkeit der beschriebenen NC-Sprachelemente

Alle im vorliegenden Handbuch beschriebenen NC-Sprachelemente stehen für SINUMERIK 840D sl zur Verfügung. Die Verfügbarkeit bezüglich SINUMERIK 828D ist der Tabelle "Anweisungen: Verfügbarkeit bei SINUMERIK 828D (Seite 771)" zu entnehmen.

Inhaltsverzeichnis

	Vorwort.....	3
1	Grundlegende Sicherheitshinweise.....	17
1.1	Allgemeine Sicherheitshinweise.....	17
1.2	Industrial Security.....	18
2	Flexible NC-Programmierung.....	19
2.1	Variablen.....	19
2.1.1	Systemdaten.....	19
2.1.2	Vordefinierte Anwendervariablen: Rechenparameter.....	22
2.1.2.1	Kanalspezifische Rechenparameter (R).....	22
2.1.2.2	Globale Rechenparameter (RG).....	23
2.1.3	Vordefinierte Anwendervariablen: Link-Variablen.....	25
2.1.4	Definition von Anwendervariablen (DEF).....	27
2.1.5	Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF).....	33
2.1.6	Attribut: Initialisierungswert.....	36
2.1.7	Attribut: Grenzwerte (LLI, ULI).....	39
2.1.8	Attribut: Physikalische Einheit (PHU).....	40
2.1.9	Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB).....	43
2.1.10	Attribut: Datenklasse (DCM, DCI, DCU) - nur SINUMERIK 828D.....	48
2.1.11	Übersicht definierbarer und redefinierbarer Attribute.....	49
2.1.12	Definition und Initialisierung von Feldvariablen (DEF, SET, REP).....	50
2.1.13	Definition und Initialisierung von Feldvariablen (DEF, SET, REP): Weitere Informationen.....	54
2.1.14	Datentypen.....	56
2.1.15	Vorhandensein einer Variablen prüfen (ISVAR).....	57
2.1.16	Attributwerte/Datentyp lesen (GETVARPHU, GETVARAP, GETVARLIM, GETVARDFT, GETVARTYP).....	58
2.2	Indirekte Programmierung.....	63
2.2.1	Indirekte Programmierung von Adressen.....	63
2.2.2	Indirekte Programmierung von G-Codes.....	65
2.2.3	Indirekte Programmierung von Positionsattributen (GP).....	67
2.2.4	Indirekte Programmierung von Teileprogrammzeilen (EXECSTRING).....	69
2.3	Rechenfunktionen.....	70
2.4	Vergleichs- und logische Operationen.....	72
2.5	Genauigkeitskorrektur bei Vergleichsfehlern (TRUNC).....	74
2.6	Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND).....	76
2.7	Priorität der Operationen.....	77
2.8	Mögliche Typenkonvertierungen.....	78
2.9	Stringoperationen.....	79
2.9.1	Typenkonvertierung nach STRING (AXSTRING).....	80
2.9.2	Typenkonvertierung von STRING (NUMBER, ISNUMBER, AXNAME).....	80
2.9.3	Verkettung von Strings (<<).....	81

2.9.4	Wandlung in Klein-/Großbuchstaben (TOWER, TOWERUPPER).....	83
2.9.5	Länge eines Strings bestimmen (STRLEN).....	83
2.9.6	Zeichen/String in String suchen (INDEX, RINDEX, MINDEX, MATCH).....	84
2.9.7	Auswahl eines Teilstrings (SUBSTR).....	85
2.9.8	Lesen und Schreiben von einzelnen Zeichen.....	86
2.9.9	String formatieren (SPRINT).....	87
2.10	Programmsprünge und -verzweigungen.....	95
2.10.1	Rücksprung auf Programmanfang (GOTOS).....	95
2.10.2	Programmsprünge auf Sprungmarken (GOTOB, GOTOF, GOTO, GOTOC).....	96
2.10.3	Programmverzweigung (CASE ... OF ... DEFAULT ...).....	99
2.11	Programmteilwiederholung (REPEAT, REPEATB, ENDLABEL, P).....	101
2.12	Kontrollstrukturen.....	106
2.12.1	Bedingte Anweisung und Verzweigung (IF, ELSE, ENDIF).....	108
2.12.2	Endlos-Programmschleife (LOOP, ENDLOOP).....	110
2.12.3	Zählschleife (FOR ... TO ..., ENDFOR).....	110
2.12.4	Programmschleife mit Bedingung am Schleifenanfang (WHILE, ENDWHILE).....	112
2.12.5	Programmschleife mit Bedingung am Schleifenende (REPEAT, UNTIL).....	112
2.12.6	Programmbeispiel mit verschachtelten Kontrollstrukturen.....	113
2.13	Programmkoordinierung (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM).....	114
2.14	Interruptroutine (ASUP).....	120
2.14.1	Funktion einer Interruptroutine.....	120
2.14.2	Interruptroutine erstellen.....	121
2.14.3	Interruptroutine zuordnen und starten (SETINT, PRIO, BLSYNC).....	122
2.14.4	Zuordnung einer Interruptroutine deaktivieren/reactivieren (DISABLE, ENABLE).....	124
2.14.5	Zuordnung einer Interruptroutine löschen (CLRINT).....	124
2.14.6	Schnellabheben von der Kontur (SETINT LIFTFAST, ALF).....	125
2.14.7	Verfahrerrichtung beim Schnellabheben von der Kontur	127
2.14.8	Bewegungsablauf bei Interruptroutinen.....	130
2.15	Achstausch, Spindeltausch (RELEASE, GET, GETD).....	131
2.16	Achse einem anderen Kanal übergeben (AXTOCHAN).....	135
2.17	Maschinendaten wirksam setzen (NEWCONF).....	137
2.18	Datei schreiben (WRITE).....	138
2.19	Datei löschen (DELETE).....	141
2.20	Zeilen in Datei lesen (READ).....	142
2.21	Vorhandensein einer Datei prüfen (ISFILE).....	144
2.22	Datei-Informationen auslesen (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO).....	145
2.23	Aufrunden (ROUNDUP).....	147
2.24	Unterprogrammtechnik.....	148
2.24.1	Allgemeines.....	148
2.24.1.1	Unterprogramm.....	148
2.24.1.2	Unterprogrammnamen.....	150
2.24.1.3	Schachtelung von Unterprogrammen.....	151
2.24.1.4	Suchpfad.....	152
2.24.1.5	Formal- und Aktualparameter.....	152

2.24.1.6	Parameterübergabe.....	153
2.24.2	Definition eines Unterprogramms.....	154
2.24.2.1	Unterprogramm ohne Parameterübergabe.....	154
2.24.2.2	Unterprogramm mit Parameterübergabe Call-by-Value (PROC).....	155
2.24.2.3	Unterprogramm mit Parameterübergabe Call-by-Reference (PROC, VAR).....	157
2.24.2.4	Modale G-Funktionen sichern (SAVE).....	159
2.24.2.5	Einzelsatzbearbeitung unterdrücken (SBLOF, SBLON).....	160
2.24.2.6	Aktuelle Satzanzeige unterdrücken (DISPLOF, DISPLON, ACTBLOCNO).....	166
2.24.2.7	Unterprogramme mit Vorbereitung kennzeichnen (PREPRO).....	169
2.24.2.8	Unterprogrammrücksprung M17.....	170
2.24.2.9	Unterprogrammrücksprung RET.....	171
2.24.2.10	Parametrierbarer Unterprogrammrücksprung (RET ...).....	172
2.24.2.11	Parametrierbarer Unterprogrammrücksprung (RETB ...).....	178
2.24.3	Aufruf eines Unterprogramms.....	182
2.24.3.1	Unterprogrammaufruf ohne Parameterübergabe.....	182
2.24.3.2	Unterprogrammaufruf mit Parameterübergabe (EXTERN).....	183
2.24.3.3	Anzahl der Programmwiederholungen (P).....	186
2.24.3.4	Modaler Unterprogrammaufruf (MCALL).....	187
2.24.3.5	Indirekter Unterprogrammaufruf (CALL).....	189
2.24.3.6	Indirekter Unterprogrammaufruf mit Angabe des auszuführenden Programmteils (CALL BLOCK ... TO ...).....	190
2.24.3.7	Indirekter Aufruf eines in ISO-Sprache programmierten Programms (ISOCALL).....	192
2.24.3.8	Unterprogramm mit Pfadangabe und Parametern aufrufen (PCALL).....	192
2.24.3.9	Suchpfad bei Unterprogrammaufrufen erweitern (CALLPATH).....	193
2.24.3.10	Externes Unterprogramm abarbeiten (840D sl) (EXTCALL).....	194
2.24.3.11	Externes Unterprogramm abarbeiten (828D) (EXTCALL).....	198
2.25	Makrotechnik (DEFINE ... AS).....	202
3	Datei- und Programmverwaltung.....	205
3.1	Programmspeicher.....	205
3.1.1	Programmspeicher im NCK.....	205
3.1.2	Externe Programmspeicher.....	207
3.1.3	Adressierung von Dateien des Programmspeichers.....	209
3.1.4	Suchpfad bei Unterprogrammaufruf.....	214
3.1.5	Abfrage von Pfad und Dateiname.....	215
3.2	Arbeitsspeicher (CHANDATA, COMPLETE, INITIAL).....	216
4	Schutzbereiche.....	221
4.1	Festlegung der Schutzbereiche (CPROTDEF, NPROTDEF).....	221
4.2	Schutzbereiche aktivieren/deaktivieren (CPROT, NPROT).....	224
4.3	Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldebegrenzung und Softwareendschalter (CALCPOSI).....	228
5	Spezielle Wegbefehle.....	233
5.1	Codierte Positionen anfahren (CAC, CIC, CDC, CACP, CACN).....	233
5.2	Spline-Interpolation (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL).....	234
5.3	Spline-Verbund (SPLINEPATH).....	244

5.4	NC-Satz-Kompression ein-/ausschalten (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF).....	245
5.5	Polynom-Interpolation (POLY, POLYPATH, PO, PL).....	246
5.6	Einstellbarer Bahnbezug (SPATH, UPATH).....	251
5.7	Messen mit schaltendem Taster (MEAS, MEAW).....	254
5.8	Axiales Messen (MEASA, MEAWA, MEAC) (Option).....	257
5.9	Spezielle Funktionen für den OEM-Anwender (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829).....	267
5.10	Vorschubreduzierung mit Eckenverzögerung (FENDNORM, G62, G621)	268
5.11	Programmierbares Bewegungsendekriterium (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA).....	269
6	Koordinatentransformationen (Frames).....	273
6.1	Koordinatentransformation über Framevariable.....	273
6.1.1	Vordefinierte Framevariable (\$P_CHBFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME).....	275
6.2	Wertzuweisungen an Frames.....	278
6.2.1	Direkte Werte zuweisen (Achswert, Winkel, Maßstab).....	278
6.2.2	Framekomponenten lesen und verändern (TR, FI, RT, SC, MI).....	280
6.2.3	Rechnen mit Frames.....	282
6.2.4	Definition von Framevariablen (DEF FRAME).....	283
6.3	Grob- und Feinverschiebung (CTRANS, CFINE).....	284
6.4	Externe Nullpunktverschiebung (\$AA_ETRANS).....	285
6.5	Istwertsetzen mit Verlust des Referenzierstatus (PRESETON).....	287
6.6	Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS).....	288
6.7	Frame-Berechnung aus 3 Messpunkten im Raum (MEAFRAME).....	289
6.8	NCU-globale Frames.....	293
6.8.1	Kanalspezifische Frames (\$P_CHBFR, \$P_UBFR).....	294
6.8.2	Im Kanal wirksame Frames.....	295
7	Transformationen.....	301
7.1	Allgemeine Programmierung der Transformationsarten.....	301
7.1.1	Orientierungsbewegungen bei den Transformationen.....	303
7.1.2	Übersicht der Orientierungstransformation TRAORI.....	307
7.2	Drei-, Vier- und Fünf-Achs-Transformation (TRAORI).....	308
7.2.1	Allgemeine Zusammenhänge Kardanischer Werkzeugkopf.....	308
7.2.2	Drei, Vier, und Fünf- Achs-Transformation (TRAORI).....	311
7.2.3	Varianten der Orientierungsprogrammierung und Grundstellung (ORIRESET).....	313
7.2.4	Programmierung der Werkzeugorientierung (A..., B..., C..., LEAD, TILT).....	314
7.2.5	Stirnfräsen (A4, B4, C4, A5, B5, C5).....	321
7.2.6	Bezug der Orientierungsachsen (ORIWKS, ORIMKS).....	322
7.2.7	Programmierung der Orientierungsachsen (ORIXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2).....	324

7.2.8	Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO).....	326
7.2.9	Orientierungsvorgabe zweier Kontaktpunkte (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=).....	329
7.3	Orientierungspolynome (PO[Winkel], PO[Koordinate]).....	331
7.4	Drehungen der Werkzeugorientierung (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA).....	333
7.5	Bahnrelative Orientierungen.....	335
7.5.1	Orientierungsarten relativ zur Bahn.....	335
7.5.2	Bahnrelative Drehung der Werkzeugorientierung (ORIPATH, ORIPATHS, Drehwinkel)	336
7.5.3	Bahnrelative Interpolation der Werkzeugdrehung (ORIROTC, THETA).....	338
7.5.4	Glättung des Orientierungsverlaufs (ORIPATHS A8=, B8=, C8=).....	340
7.6	Komprimierung der Orientierung (COMPON, COMPCURV, COMPCAD, COMPSURF)	341
7.7	Glättung des Orientierungsverlaufs (ORISON, ORISOF).....	344
7.8	Kinematische Transformation.....	346
7.8.1	Stirnseitentransformation einschalten (TRANSMIT).....	346
7.8.2	Zylindermanteltransformation einschalten (TRACYL).....	346
7.8.3	Schiefwinkeltransformation einschalten (TRAANG).....	349
7.8.4	Schräges Einstechen an Schleifmaschinen (G5, G7).....	350
7.9	Verkettete Transformation einschalten (TRACON).....	351
7.10	Kartesisches PTP-Fahren.....	353
7.10.1	Kartesisches PTP-Fahren ein-/ausschalten (PTP, PTPG0, PTPWOC, CP).....	353
7.10.2	Stellung der Gelenke angeben (STAT).....	354
7.10.3	Vorzeichen der Achswinkel angeben (TU).....	357
7.10.4	Beispiel 1: PTP und TRAORI.....	359
7.10.5	Beispiel 2: PTPG0 und TRANSMIT.....	360
7.11	Randbedingungen bei der Anwahl einer Transformation.....	362
7.12	Transformation abwählen (TRAFOOF).....	363
8	Kinematische Ketten.....	365
8.1	Löschen von Komponenten (DELOBJ).....	365
8.2	Indexermittlung per Namen (NAMETOINT).....	368
9	Kollisionsvermeidung mit kinematischen Ketten.....	369
9.1	Prüfen auf Kollisionspaar (COLLPAIR).....	369
9.2	Neuberechnung des Kollisionsmodells anfordern (PROTA).....	370
9.3	Schutzbereichszustand setzen (PROTS).....	371
9.4	Abstandsbestimmung zweier Schutzbereiche (PROTD).....	372
10	Werkzeugkorrekturen.....	375
10.1	Korrekturspeicher.....	375
10.2	Additive Korrekturen.....	378
10.2.1	Additive Korrekturen anwählen (DL).....	378
10.2.2	Verschleiß- und Einrichtewerte festlegen (\$TC_SCPxy[t,d], \$TC_ECPxy[t,d]).....	379
10.2.3	Additive Korrekturen löschen (DELDL).....	380

10.3	Werkzeugkorrektur - Sonderbehandlung.....	381
10.3.1	Werkzeuglängen spiegeln.....	383
10.3.2	Vorzeichenbewertung Verschleiß.....	383
10.3.3	Koordinatensystem der aktiven Bearbeitung (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS).....	384
10.3.4	Werkzeuglänge und Ebenenwechsel.....	387
10.4	Online-Werkzeugkorrektur.....	388
10.4.1	Polynom-Funktion definieren (FCTDEF).....	388
10.4.2	Online-Werkzeugkorrektur schreiben, kontinuierlich (PUTFTOCF).....	390
10.4.3	Online-Werkzeugkorrektur schreiben, diskret (PUTFTOC).....	391
10.4.4	Online-Werkzeugkorrektur ein-/ausschalten (FTOCON/FTOCOF).....	391
10.5	Aktivierung 3D-Werkzeugkorrekturen (CUT3DC..., CUT3DF...).....	392
10.5.1	Aktivierung von 3D-Werkzeugkorrekturen (CUT3DC, CUT3DCD, CUT3DF, CUT3DFS, CUT3DFF, ISD).....	392
10.5.2	3D-Werkzeugkorrektur: Umfangfräsen, Stirnfräsen.....	395
10.5.3	3D-Werkzeugkorrektur: Werkzeugformen und Werkzeugdaten für Stirnfräsen.....	397
10.5.4	3D-Werkzeugkorrektur: Korrektur auf der Bahn, Bahnkrümmung, Eintauchtiefe (CUT3DC, CUT3DCD, ISD).....	398
10.5.5	3D-Werkzeugkorrektur: Innenecken/Außenecken und Schnittpunktverfahren (G450/G451).....	400
10.5.6	3D-Werkzeugkorrektur: 3D-Umfangsfräsen mit Begrenzungsflächen.....	402
10.5.7	3D-Werkzeugkorrektur: Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD).....	402
10.6	Werkzeugorientierung (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST).....	406
10.7	Freie D-Nummernvergabe, Schneidenummer.....	412
10.7.1	Freie D-Nummernvergabe, Schneidenummer (Adresse CE).....	412
10.7.2	Freie D-Nummernvergabe: D-Nummern prüfen (CHKDNO).....	412
10.7.3	Freie D-Nummernvergabe: D-Nummern umbenennen (GETDNO, SETDNO).....	413
10.7.4	Freie D-Nummernvergabe: T-Nummer zur vorgegebenen D-Nummer ermitteln (GETACTTD).....	414
10.7.5	Freie D-Nummernvergabe: D-Nummern ungültig setzen (DZERO).....	414
10.8	Werkzeugträgerkinematik.....	415
10.9	Werkzeuglängenkorrektur für orientierbare Werkzeugträger (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ).....	421
10.10	Online-Werkzeuglängenkorrektur (TOFFON, TOFFOF).....	423
10.11	Schneidendaten-Modifikation bei drehbaren Werkzeugen (CUTMOD).....	427
11	Bahnverhalten.....	433
11.1	Tangentialsteuerung.....	433
11.1.1	Kopplung definieren (TANG).....	433
11.1.2	Zwischensatzerzeugung einschalten (TLIFT).....	434
11.1.3	Kopplung einschalten (TANGON).....	436
11.1.4	Kopplung ausschalten (TANGOF).....	437
11.1.5	Kopplung löschen (TANGDEL).....	437
11.2	Vorschubverlauf (FNORM, FLIN, FCUB, FPO).....	438
11.3	Beschleunigungsverhalten.....	444
11.3.1	Beschleunigungsmodus (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA).....	444

11.3.2	Beeinflussung der Beschleunigung bei Folgeachsen (VELOLIMA, ACCLIMA, JERKLIMA).....	446
11.3.3	Aktivierung von Technologie-spezifischen Dynamikwerten (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH).....	448
11.4	Fahren mit Vorsteuerung (FFWON, FFWOF).....	449
11.5	Programmierbare Konturgenauigkeit (CPRECON, CPRECOF).....	450
11.6	Programmablauf mit Vorlaufspeicher (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)	452
11.7	Beeinflussung von Stopp-Ereignissen durch Stop-Delay-Bereiche (DELAYFSTON, DELAYFSTOF).....	454
11.8	Programmstelle für SERUPRO verhindern (IPTRLOCK, IPTRUNLOCK).....	456
11.9	Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)	458
11.10	Beeinflussung der Bewegungsführung.....	467
11.10.1	Prozentuale Ruckkorrektur (JERKLIM).....	467
11.10.2	Prozentuale Geschwindigkeitskorrektur (VELOLIM).....	468
11.10.3	Programmbeispiel für JERKLIM und VELOLIM.....	470
11.11	Programmierbare Kontur-/Orientierungstoleranz (CTOL, OTOL, ATOL).....	471
11.12	Toleranz bei G0-Bewegungen (STOLF).....	474
11.13	Satzwechselverhalten bei aktiver Kopplung (CPBC).....	476
12	Achskopplungen.....	477
12.1	Mitschleppen (TRAILON, TRAILOF).....	477
12.2	Kurventabellen (CTAB).....	481
12.2.1	Kurventabellen definieren (CTABDEF, CATBEND).....	481
12.2.2	Vorhandensein einer Kurventabelle prüfen (CTABEXISTS).....	488
12.2.3	Kurventabellen löschen (CTABDEL).....	488
12.2.4	Kurventabellen gegen Löschen und Überschreiben sperren (CTABLOCK, CTABUNLOCK).....	489
12.2.5	Kurventabellen: Tabelleneigenschaften ermitteln (CTABID, CTABISLOCK, CTABMENTYP, CTABPERIOD).....	490
12.2.6	Kurventabellenwerte lesen (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX).....	492
12.2.7	Kurventabellen: Ressourcennutzung prüfen (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL).....	496
12.3	Axiale Leitwertkopplung (LEADON, LEADOF).....	497
12.4	Elektronisches Getriebe (EG).....	503
12.4.1	Elektronisches Getriebe definieren (EGDEF).....	503
12.4.2	Elektronisches Getriebe einschalten (EGON, EGONSYN, EGONSYNE).....	504
12.4.3	Elektronisches Getriebe ausschalten (EGOFS, EGOFC).....	507
12.4.4	Definition eines Elektronischen Getriebes löschen (EGDEL).....	508
12.4.5	Umdrehungsvorschub (G95) / Elektronisches Getriebe (FPR).....	509
12.5	Synchronspindel.....	509
12.5.1	Synchronspindel: Programmierung (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC).....	510


12.6	Generische Kopplung (CP...)	520
12.7	Master/Slave-Kopplung (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)	528
13	Synchronaktionen	531
13.1	Definition einer Synchronaktion	531
14	Pendeln	533
14.1	Asynchrones Pendeln (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)	533
14.2	Über Synchronaktionen gesteuertes Pendeln (OSCILL)	538
15	Stanzen und Nibbeln	545
15.1	Aktivierung, Deaktivierung	545
15.1.1	Stanzen und Nibbeln ein oder aus (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)	545
15.2	Automatische Wegaufteilung	550
15.2.1	Wegaufteilung bei Bahnachsen	552
15.2.2	Wegaufteilung bei Einzelachsen	554
16	Schleifen	557
16.1	Schleifenspezifische Werkzeugüberwachung ein-/ausschalten (TMON, TMOF)	557
17	Weitere Funktionen	559
17.1	Achsfunktionen (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)	559
17.2	Umschaltbare Geometrieachsen (GEOAX)	561
17.3	Achscontainer (AXCTSWE, AXCTSWED, AXCTSWEC)	565
17.4	Warten auf gültige Achsposition (WAITENC)	567
17.5	Programmierbare Parametersatzumschaltung (SCPARA)	568
17.6	Vorhandenen NC-Sprachumfang prüfen (STRINGIS)	570
17.7	Fenster aus dem Teileprogramm interaktiv aufrufen (MMC)	573
17.8	Programmlaufzeit / Werkstückzähler	575
17.8.1	Programmlaufzeit / Werkstückzähler (Übersicht)	575
17.8.2	Programmlaufzeit	575
17.8.3	Werkstückzähler	579
17.9	Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE)	580
17.10	Alarmer (SETAL)	584
17.11	Erweitertes Stillsetzen und Rückziehen (ESR)	586
17.11.1	NC-geführtes ESR	587
17.11.1.1	NC-geführtes Rückziehen (POLF, POLFA, POLFMASK, POLFMLIN)	587
17.11.1.2	NC-geführtes Stillsetzen	591
17.11.2	Antriebsautarkes ESR	591
17.11.2.1	Antriebsautarkes Stillsetzen projektieren (ESRS)	591
17.11.2.2	Antriebsautarkes Rückziehen projektieren (ESRR)	592


18	Eigene Abspanprogramme.....	595
18.1	Unterstützende Funktionen für das Abspannen.....	595
18.2	Konturtabelle erstellen (CONTPRON).....	595
18.3	Codierte Konturtabelle erstellen (CONTDCON).....	601
18.4	Schnittpunkt zwischen zwei Konturelementen ermitteln (INTERSEC).....	605
18.5	Konturelemente einer Tabelle satzweise abfahren (EXECTAB).....	606
18.6	Kreisdaten berechnen (CALCDAT).....	607
18.7	Konturaufbereitung ausschalten (EXECUTE).....	609
19	Zyklen extern programmieren.....	611
19.1	Technologische Zyklen.....	611
19.1.1	Einleitung.....	611
19.1.2	Technologie-spezifische Übersicht.....	612
19.1.3	HOLES1 - Lochreihe.....	614
19.1.4	HOLES2 - Lochkreis.....	614
19.1.5	POCKET3 - Rechtecktasche fräsen.....	616
19.1.6	POCKET4 - Kreistasche fräsen.....	619
19.1.7	SLOT1 - Längsnut.....	621
19.1.8	SLOT2 - Kreisnut.....	624
19.1.9	LONGHOLE - Langloch.....	626
19.1.10	CYCLE60 - Gravurzyklus.....	628
19.1.11	CYCLE61 - Planfräsen.....	631
19.1.12	CYCLE62 - Konturaufruf.....	633
19.1.13	CYCLE63 - Konturtasche fräsen.....	634
19.1.14	CYCLE64 - Konturtasche vorbohren.....	636
19.1.15	CYCLE70 - Gewindefräsen.....	637
19.1.16	CYCLE72 - Bahnfräsen.....	639
19.1.17	CYCLE76 - Rechteckzapfen fräsen.....	643
19.1.18	CYCLE77 - Kreiszapfen fräsen.....	645
19.1.19	CYCLE78 - Bohrgewinde fräsen.....	647
19.1.20	CYCLE79 - Mehrkant.....	649
19.1.21	CYCLE81 - Bohren, Zentrieren.....	651
19.1.22	CYCLE82 - Bohren, Plansenken.....	652
19.1.23	CYCLE83 - Tieflochbohren.....	655
19.1.24	CYCLE84 - Gewindebohren ohne Ausgleichsfutter.....	658
19.1.25	CYCLE85 - Reiben.....	661
19.1.26	CYCLE86 - Ausdrehen.....	662
19.1.27	CYCLE92 - Abstich.....	663
19.1.28	CYCLE95 - Konturabspannen.....	665
19.1.29	CYCLE98 - Gewindekette.....	667
19.1.30	CYCLE99 - Gewindedrehen.....	671
19.1.31	CYCLE435 - Abrichterposition berechnen.....	676
19.1.32	CYCLE495 - Profilieren.....	676
19.1.33	CYCLE800 - Schwenken.....	678
19.1.34	CYCLE801 - Gitter oder Rahmen.....	681
19.1.35	CYCLE802 - Beliebige Positionen.....	683
19.1.36	CYCLE830 - Tieflochbohren 2.....	686
19.1.37	CYCLE832 - High Speed Settings.....	692

19.1.38	CYCLE840 - Gewindebohren mit Ausgleichsfutter.....	695
19.1.39	CYCLE899 - Offene Nut fräsen.....	698
19.1.40	CYCLE930 - Einstich.....	701
19.1.41	CYCLE940 - Freistich Formen.....	703
19.1.42	CYCLE951 - Abspannen.....	706
19.1.43	CYCLE952 - Konturstechen.....	709
19.1.44	CYCLE4071 - Längsschleifen mit Zustellung am Umkehrpunkt.....	715
19.1.45	CYCLE4072 - Längsschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal.....	716
19.1.46	CYCLE4073 - Längsschleifen mit kontinuierlicher Zustellung.....	720
19.1.47	CYCLE4074 - Längsschleifen mit kontinuierlicher Zustellung und Abbruchsignal.....	721
19.1.48	CYCLE4075 - Flachsleifen mit Zustellung am Umkehrpunkt.....	724
19.1.49	CYCLE4077 - Flachsleifen mit Zustellung am Umkehrpunkt und Abbruchsignal.....	727
19.1.50	CYCLE4078 - Flachsleifen mit kontinuierlicher Zustellung.....	730
19.1.51	CYCLE4079 - Flachsleifen mit intermittierender Zustellung.....	732
19.1.52	Randbedingungen.....	734
19.1.52.1	Technologieskalierung innerhalb von Zyklenmasken.....	734
19.2	Messzyklen.....	736
20	Tabellen.....	737
20.1	Anweisungen.....	737
20.2	Anweisungen: Verfügbarkeit bei SINUMERIK 828D	771
20.3	Aktuelle Sprache im HMI.....	796
A	Anhang.....	799
A.1	Liste der Abkürzungen.....	799
A.2	Dokumentationsübersicht.....	808
	Glossar.....	809
	Index.....	831

Grundlegende Sicherheitshinweise

1.1 Allgemeine Sicherheitshinweise

 WARNUNG
Lebensgefahr durch Nichtbeachtung von Sicherheitshinweisen und Restrisiken
Durch Nichtbeachtung der Sicherheitshinweise und Restrisiken in der zugehörigen Hardware-Dokumentation können Unfälle mit schweren Verletzungen oder Tod auftreten.
<ul style="list-style-type: none">• Halten Sie die Sicherheitshinweise der Hardware-Dokumentation ein.• Berücksichtigen Sie bei der Risikobeurteilung die Restrisiken.

 WARNUNG
Lebensgefahr durch Fehlfunktionen der Maschine infolge fehlerhafter oder veränderter Parametrierung
Durch fehlerhafte oder veränderte Parametrierung können Fehlfunktionen an Maschinen auftreten, die zu Körperverletzungen oder Tod führen können.
<ul style="list-style-type: none">• Schützen Sie die Parametrierungen vor unbefugtem Zugriff.• Beherrschen Sie mögliche Fehlfunktionen durch geeignete Maßnahmen (z. B. NOT-HALT oder NOT-AUS).

1.2 Industrial Security

Hinweis

Industrial Security

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Lösungen, Maschinen, Geräten und/oder Netzwerken unterstützen. Sie sind wichtige Komponenten in einem ganzheitlichen Industrial Security-Konzept. Die Produkte und Lösungen von Siemens werden unter diesem Gesichtspunkt ständig weiterentwickelt. Siemens empfiehlt, sich unbedingt regelmäßig über Produkt-Updates zu informieren.

Für den sicheren Betrieb von Produkten und Lösungen von Siemens ist es erforderlich, geeignete Schutzmaßnahmen (z. B. Zellschutzkonzept) zu ergreifen und jede Komponente in ein ganzheitliches Industrial Security-Konzept zu integrieren, das dem aktuellen Stand der Technik entspricht. Dabei sind auch eingesetzte Produkte von anderen Herstellern zu berücksichtigen. Weitergehende Informationen über Industrial Security finden Sie unter dieser Adresse (<http://www.siemens.com/industrialsecurity>).

Um stets über Produkt-Updates informiert zu sein, melden Sie sich für unseren produktspezifischen Newsletter an. Weitere Informationen hierzu finden Sie unter dieser Adresse (<http://support.automation.siemens.com>).



WARNUNG

Gefahr durch unsichere Betriebszustände wegen Manipulation der Software

Manipulationen der Software (z. B. Viren, Trojaner, Malware, Würmer) können unsichere Betriebszustände in Ihrer Anlage verursachen, die zu Tod, schwerer Körperverletzung und zu Sachschäden führen können.

- Halten Sie die Software aktuell.
Informationen und Newsletter hierzu finden Sie unter dieser Adresse (<http://support.automation.siemens.com>).
- Integrieren Sie die Automatisierungs- und Antriebskomponenten in ein ganzheitliches Industrial Security-Konzept der Anlage oder Maschine nach dem aktuellen Stand der Technik.
Weitergehende Informationen finden Sie unter dieser Adresse (<http://www.siemens.com/industrialsecurity>).
- Berücksichtigen Sie bei Ihrem ganzheitlichen Industrial Security-Konzept alle eingesetzten Produkte.

Flexible NC-Programmierung

2.1 Variablen

Durch die Verwendung von Variablen aus den Bereichen Systemdaten und Anwenderdaten, insbesondere in Verbindung mit Rechenfunktionen und Kontrollstrukturen, können NC-Programme und Zyklen flexibel gestaltet werden.

- **Systemdaten**
In den Systemdaten liegen die im System vordefinierten Variablen. Diese Variablen haben eine definierte Bedeutung. Sie werden in erster Linie von der Systemsoftware verwendet. Vom Anwender können diese Variablen in NC-Programmen und Zyklen gelesen und geschrieben werden. Beispiel: Maschinendaten, Settingdaten, Systemvariablen. Obwohl die Bedeutung eines Systemdatums fest vorgegeben ist, können die Eigenschaften vom Anwender durch Redefinition in gewissem Umfang verändert werden. Siehe "Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF) (Seite 33)"
- **Anwenderdaten**
In den Anwenderdaten liegen die vom Anwender definierten Variablen, deren Bedeutung ausschließlich durch den Anwender festgelegt wird. Sie werden vom System nicht ausgewertet.
Die Anwenderdaten sind unterteilt in:
 - **Vordefinierte Anwendervariablen**
Vordefinierte Anwendervariablen sind im System bereits definierte Variablen, deren Anzahl über Maschinendaten parametrisiert wird. Die Eigenschaften dieser Variablen können vom Anwender angepasst werden. Siehe "Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF) (Seite 33)".
 - **Anwenderdefinierte Variablen**
Anwenderdefinierte Variablen sind Variablen, die vom Anwender definiert und vom System zur Laufzeit angelegt werden. Ihre Anzahl, Datentyp, Sichtbarkeit und alle weiteren Eigenschaften werden ausschließlich durch den Anwender festgelegt. Siehe "Definition von Anwendervariablen (DEF) (Seite 27)"

2.1.1 Systemdaten

Die Systemdaten enthalten die im System vordefinierten Variablen, die in NC_Programmen und Zyklen Zugriff auf die aktuelle Parametrierung der Steuerung sowie auf Maschinen-, Steuerungs- und Prozesszustände ermöglichen.

Vorlaufvariablen

Vorlaufvariablen sind Systemdaten, die im Kontext des Vorlaufs, d. h. zum Zeitpunkt der Interpretation des Satzes, in dem die Variable programmiert ist, gelesen und geschrieben werden. Vorlaufvariable lösen keinen Vorlaufstopp aus.

Hauptlaufvariablen

Hauptlaufvariablen sind Systemdaten, die im Kontext des Hauptlaufs, d. h. zum Zeitpunkt der Ausführung des Satzes, in dem die Variable programmiert ist, gelesen oder geschrieben werden. Hauptlaufvariablen sind:

- Variable, die in Synchronaktionen programmiert werden können (Lesen/Schreiben)
- Variable, die im NC_Programm programmiert werden können und Vorlaufstopp auslösen (Lesen/Schreiben)
- Variable, die im NC_Programm programmiert werden können und deren Wert im Vorlauf ermittelt, aber erst im Hauptlauf geschrieben wird (Hauptlauf-synchron: nur Schreiben)

Präfix-Systematik

Zur besonderen Kennzeichnung von Systemdaten ist dem Namen im Normalfall ein Präfix vorangestellt, der sich aus dem \$-Zeichen, gefolgt von einem oder zwei Buchstaben und einem Unterstrich, zusammensetzt:

\$ + 1. Buchstabe	Bedeutung: Datenart
Systemdaten, die im Vorlauf gelesen / geschrieben werden	
\$M	Maschinendaten ¹⁾
\$S	Settingdaten, Schutzbereiche ¹⁾
\$T	Werkzeugverwaltungsdaten
\$P	Programmierte Werte
\$C	Zyklusvariablen der ISO-Hüllzyklen
\$O	Optionsdaten
R	R-Parameter (Rechenparameter) ²⁾
Systemdaten, die im Hauptlauf gelesen / geschrieben werden	
\$\$M	Maschinendaten ¹⁾
\$\$S	Settingdaten ¹⁾
\$A	Aktuelle Hauptlaufdaten
\$V	Servo-Daten
\$R	R-Parameter (Rechenparameter) ²⁾
¹⁾ Ob Maschinen- und Settingdaten als Vor- oder Hauptlaufvariable behandelt werden, hängt davon ab, ob sie mit einem oder zwei \$-Zeichen geschrieben werden. Die Schreibweise ist anwendungsspezifisch frei wählbar.	
²⁾ Bei der Verwendung eines R-Parameters im Teileprogramm / Zyklus als Vorlaufvariable wird kein Präfix geschrieben, z. B. R10. Bei der Verwendung in einer Synchronaktion als Hauptlaufvariable wird als Präfix ein \$-Zeichen geschrieben, z. B. \$R10.	

2. Buchstabe	Bedeutung: Sichtbarkeit
N	NCK-globale Variable (NCK)
C	kanalspezifische Variable (Channel)
A	achsspezifische Variable (Axis)

Randbedingungen

Ausnahmen in der Präfix-Systematik

Folgende Systemvariablen weichen von der oben genannten Präfix-Systematik ab:

- \$TC_...: Der 2. Buchstabe C verweist hier nicht auf kanalspezifische, sondern auf Werkzeughalter-spezifische Systemvariablen (TC = Tool Carrier)
- \$P_ ...: Kanalspezifische Systemvariablen

Verwendung von Maschinen- und Settingdaten in Synchronaktionen

Bei der Verwendung von Maschinen- und Settingdaten in Synchronaktionen kann durch den Präfix bestimmt werden, ob das Maschinen- oder Settingdatum vorlauf- oder hauptlaufsynchron gelesen/geschrieben wird.

Bleibt das Datum während der Bearbeitung unverändert, kann vorlaufsynchron gelesen werden. Der Präfix des Maschinen- oder Settingdatums wird dazu mit einem \$-Zeichen geschrieben:

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

Wird das Datum während der Bearbeitung verändert, muss hauptlaufsynchron gelesen / geschrieben werden. Der Präfix des Maschinen- oder Settingdatums wird dazu mit zwei \$-Zeichen geschrieben:

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

Hinweis

Schreiben von Maschinen- und Settingdaten

Beim Schreiben eines Maschinen- oder Settingdatums ist darauf zu achten, dass die aktive Zugriffsstufe beim Ausführen des Teileprogramms / Zyklus den Schreibzugriff erlaubt und die Wirksamkeit des Datums "IMMEDIATE" ist.

Literatur

Eine Auflistung der Eigenschaften aller Systemvariablen findet sich in:

Listenhandbuch Systemvariablen

Siehe auch

Variablen (Seite 19)

2.1.2 Vordefinierte Anwendervariablen: Rechenparameter

2.1.2.1 Kanalspezifische Rechenparameter (R)

Kanalspezifische Rechenparameter oder R-Parameter sind vordefinierte Anwendervariablen mit der Bezeichnung R, definiert als Feld vom Datentyp REAL. Aus historischen Gründen ist für R-Parameter neben der Schreibweise mit Feldindex z. B. R[10], auch die Schreibweise ohne Feldindex z. B. R10, erlaubt.

Bei der Verwendung in Synchronaktionen muss der Buchstabe \$ vorangestellt werden, z. B. \$R10.

Syntax

Bei Verwendung als Vorlaufvariable:

R<n>

R[<Ausdruck>]

Bei Verwendung als Hauptlaufvariable:

\$R<n>

\$R[<Ausdruck>]

Bedeutung

R:	Bezeichner bei Verwendung als Vorlaufvariable, z. B. im Teileprogramm	
\$R:	Bezeichner bei Verwendung als Hauptlaufvariable z. B. in Synchronaktionen	
	Typ:	REAL
	Wertebereich:	Bei nicht-exponentieller Schreibweise: ± (0.000 0001 ... 9999 9999) Hinweis: Es sind maximal 8 Dezimalstellen erlaubt
		Bei exponentieller Schreibweise: ± (1*10 ⁻³⁰⁰ ... 1*10 ⁺³⁰⁰) Hinweis: • Schreibweise: <Mantisse>EX<exponent> z.B. 8.2EX-3 • Es sind maximal 10 Zeichen einschließlich Vorzeichen und Dezimalpunkt erlaubt.
<n>:	Nummer des R-Parameters	
	Typ:	INT
	Wertebereich:	0 - MAX_INDEX Hinweis MAX_INDEX ergibt sich aus der parametrisierten Anzahl an R-Parametern: MAX_INDEX = (MD28050 \$MN_MM_NUM_R_PARAM) - 1
<Ausdruck>:	Feldindex Als Feldindex kann ein beliebiger Ausdruck angegeben werden, solange das Ergebnis des Ausdrucks in den Datentyp INT gewandelt werden kann (INT, REAL, BOOL, CHAR)	

Beispiel

Zuweisungen an R-Parameter und Verwendung von R-Parametern in mathematischen Funktionen:

Programmcode	Kommentar
R0=3.5678	; Zuweisung im Vorlauf
R[1]=-37.3	; Zuweisung im Vorlauf
R3=-7	; Zuweisung im Vorlauf
\$R4=-0.1EX-5	; Zuweisung im Hauptlauf: R4 = -0.1 * 10 ⁻⁵
\$R[6]=1.874EX8	; Zuweisung im Hauptlauf: R6 = 1.874 * 10 ⁸
R7=SIN(25.3)	; Zuweisung im Vorlauf
R[R2]=R10	; Indirekte Adressierung über R-Parameter
R[(R1+R2)*R3]=5	; Indirekte Adressierung über math. Ausdruck
X=(R1+R2)	; Verfahre Achse X auf die Position die sich aus der Summe von R1 und R2 ergibt
Z=SQRT(R1*R1+R2*R2)	; Verfahre Achse Z auf Position Quadratwurzel(R1 ² + R2 ²)

Siehe auch

Variablen (Seite 19)

2.1.2.2 Globale Rechenparameter (RG)**Funktion**

Neben den kanalspezifischen R-Parametern stehen dem Anwender auch globale R-Parameter zur Verfügung. Sie existieren innerhalb der Steuerung einmal und können von allen Kanälen aus gelesen/geschrieben werden.

Globale R-Parameter werden z. B. verwendet, um Informationen von einem Kanal in den nächsten zu bekommen. Ein anderes Beispiel sind globale Einstellungen, die für alle Kanäle ausgewertet werden sollen, wie z. B. die Auskrägung des Rohteils aus der Spindel.

Lesen und Schreiben der globalen R-Parameter erfolgt über die Bedienoberfläche oder im NC-Programm im Vorlauf. Eine Verwendung in Synchronaktionen oder Technologiezyklen ist nicht möglich.

Hinweis

Beim Lesen und Schreiben von globalen R-Parametern erfolgt **keine** Synchronisation zwischen den Kanälen.

Da das Lesen und Schreiben im Vorlauf erfolgt, ist der Zeitpunkt, zu dem ein geschriebener Wert von einem Kanal aus in einem anderen Kanal wirksam wird, nicht definiert.

Beispiel:

Im Kanal1 läuft eine Schleife mit einem globalen R-Parameter als Schleifenzähler. Kanal 2 schreibt einen Wert in diesen globalen R-Parameter, der zu einem Schleifenabbruch in Kanal 1 führt. Alle bis zu diesem Zeitpunkt im Kanal1 im Vorlauf interpretierten Schleifen werden jedoch noch ausgeführt. Wie viele Schleifen das sind ist nicht definiert und hängt unter anderem von der Auslastung des Kanals ab.

Eine Synchronisation zwischen den Kanälen muss der Anwender applikativ, z. B. mit WAIT-Marken, selbst realisieren!

Syntax

Schreiben im NC-Programm

```
RG [<n>] = <Wert>
RG [<Ausdruck>] = <Wert>
```

Lesen im NC-Programm

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

Bedeutung

RG:	Default-Name der NC-Adresse für globale R-Parameter Hinweis: Der Name der NC-Adresse ist einstellbar über MD15800 \$MN_R_PARM_NCK_NAME	
<n>:	Nummer des globalen R-Parameters	
	Typ:	INT
	Wertebereich:	0 ... MAX_INDEX Hinweis MAX_INDEX ergibt sich aus der parametrisierten Anzahl an globalen R-Parametern: MAX_INDEX = (MD18156 \$MN_MM_NUM_R_PARM_NCK) - 1
<Ausdruck>:	Als Feldindex kann ein beliebiger Ausdruck angegeben werden, solange das Ergebnis des Ausdrucks in den Datentyp INT gewandelt werden kann (INT, REAL, BOOL, CHAR)	

<Wert>:	Wert des globalen R-Parameters	
	Typ:	REAL
	Wertebereich:	Bei nicht-exponentieller Schreibweise: $\pm (0.000\ 0001 \dots 9999\ 9999)$ Hinweis: Es sind maximal 8 Dezimalstellen erlaubt Bei exponentieller Schreibweise: $\pm (1*10^{-300} \dots 1*10^{+300})$ Hinweis: <ul style="list-style-type: none"> • Schreibweise: <Mantisse>EX<exponent> z.B. 8.2EX-3 • Es sind maximal 10 Zeichen einschließlich Vorzeichen und Dezimalpunkt erlaubt.

2.1.3 Vordefinierte Anwendervariablen: Link-Variablen

Über Link-Variablen können im Rahmen der Funktion "NCU-Link" zyklisch Daten zwischen NCUs, die in einem Netzwerk miteinander verbunden sind, ausgetauscht werden. Sie ermöglichen dabei einen Datenformat-spezifischen Zugriff auf den Link-Variablen-Speicher. Der Link-Variablen-Speicher wird sowohl bezüglich der Größe und als auch der Datenstruktur vom Anwender / Maschinenhersteller anlagenspezifisch festgelegt.

Link-Variablen sind systemglobale Anwendervariablen, die bei projektierter Link-Kommunikation von allen NCUs des Link-Verbundes in Teileprogrammen und Zyklen gelesen und geschrieben werden können. Im Gegensatz zu globalen Anwendervariablen (GUD) können Link-Variablen auch in Synchronaktionen verwendet werden.

Bei Anlagen ohne aktiven NCU-Link können Link-Variablen Steuerungs-lokal neben den globalen Anwendervariablen (GUD) als zusätzliche globale Anwendervariablen verwendet werden.

Syntax

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

Bedeutung

\$A_DLB:	Link-Variable für Datenformat BYTE (1 Byte)	
	Datentyp:	UINT
	Wertebereich:	0 ... 255
\$A_DLW:	Link-Variable für Datenformat WORD (2 Bytes)	
	Datentyp:	INT
	Wertebereich:	-32768 ... 32767

2.1 Variablen

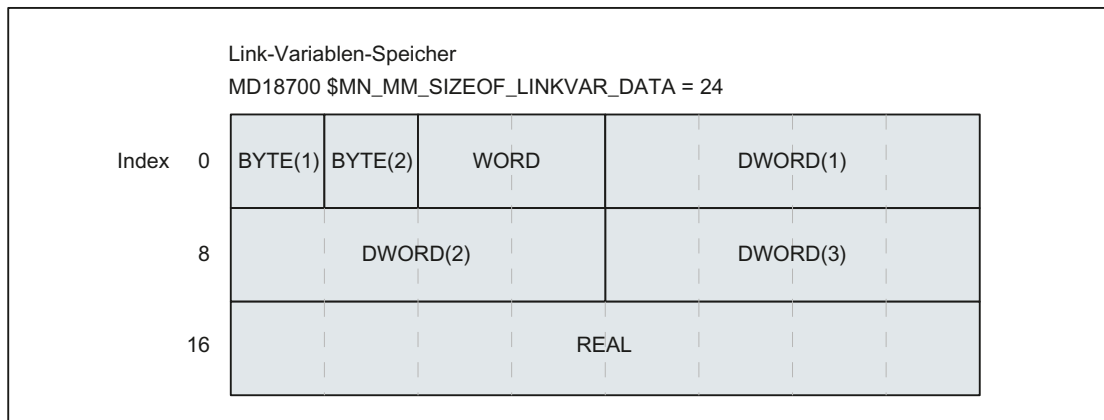
\$A_DLD:	Link-Variablen für Datenformat DWORD (4 Bytes)	
	Datentyp:	INT
	Wertebereich:	-2147483648 ... 2147483647
\$A_DLR:	Link-Variablen für Datenformat REAL (8 Bytes)	
	Datentyp:	REAL
	Wertebereich:	$\pm(2,2 \cdot 10^{-308} \dots 1,8 \cdot 10^{308})$
<Index>:	Adressindex in Byte, gerechnet vom Anfang des Link-Variablen-Speichers	
	Datentyp:	INT
	Wertebereich:	0 - MAX_INDEX Hinweis <ul style="list-style-type: none"> • MAX_INDEX ergibt sich aus der parametrisierten Größe des Link-Variablen-Speichers: $MAX_INDEX = (MD18700 \\$MN_MM_SIZEOF_LINKVAR_DATA) - 1$ • Es dürfen nur Indizes programmiert werden, so dass die im Link-Variablen-Speicher adressierten Bytes auf einer Datenformatgrenze liegen \Rightarrow Index = n * Bytes, mit n = 0, 1, 2, ... <ul style="list-style-type: none"> - \$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, ...

Beispiel

In der Automatisierungsanlage sind 2 NCUs (NCU1 und NCU2) vorhanden. An NCU1 ist Maschinenachse AX2 angeschlossen, die als Link-Achse von NCU2 verfahren wird.

NCU1 schreibt zyklisch den Stromistwert (\$VA_CURR) der Achse AX2 in den Link-Variablen-Speicher. NCU2 liest zyklisch den per Link-Kommunikation übertragenen Stromistwert und zeigt bei Überschreitung des Grenzwertes Alarm 61000 an.

Die Datenstruktur im Link-Variablen-Speicher ist im folgenden Bild dargestellt. Der Stromistwert wird über den REAL-Wert übertragen.



NCU1

NCU1 schreibt in einer statischen Synchronaktion zyklisch im IPO-Takt den Stromistwert der Achse AX2 über die Link-Variable \$A_DLR[16] in den Link-Variablen-Speicher.

Programmcode

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

NCU2

NCU2 liest in einer statischen Synchronaktion zyklisch im IPO-Takt den Stromistwert der Achse AX2 über die Link-Variable \$A_DLR[16] aus dem Link-Variablen-Speicher. Ist der Stromistwert größer als 23.0 A, wird der Alarm 61000 angezeigt.

Programmcode

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

Siehe auch

Variablen (Seite 19)

2.1.4 Definition von Anwendervariablen (DEF)

Mit dem Befehl `DEF` können anwenderspezifische Variablen, oder Anwendervariablen (User Data), definiert und mit Werten belegt werden.

Entsprechend dem Gültigkeitsbereich, d. h. dem Bereich in dem die Variable sichtbar ist, gibt es folgende Kategorien von Anwendervariablen:

- Lokale Anwendervariablen (LUD)
Lokale Anwendervariablen (LUD) sind Variablen, die in einem NC-Programm definiert sind, das zum Zeitpunkt der Abarbeitung nicht das Hauptprogramm ist. Sie werden beim Aufruf des NC-Programms angelegt und mit dem Ende des NC-Programms bzw. NC-Reset gelöscht. Auf LUD kann nur innerhalb des NC-Programms zugegriffen werden, in dem sie definiert sind.
- Programmglobale Anwendervariablen (PUD)
Programmglobale Anwendervariablen (PUD) sind Variablen, die in einem als Hauptprogramm verwendeten NC-Programm definiert sind. Sie werden mit NC-Programmstart angelegt und mit NC-Programmende bzw. NC-Reset gelöscht. Auf PUD kann im Hauptprogramm und in allen Unterprogrammen zugegriffen werden.
- Globale Anwendervariablen (GUD)
Globale Anwendervariablen (GUD) sind NC- bzw. Kanal-globale Variablen, die in einem Datenbaustein (SGUD, MGUD, UGUD, GUD4 ... GUD9) definiert sind und auch über Power On hinaus erhalten bleiben. Auf GUD kann in allen NC-Programmen zugegriffen werden.

Anwendervariablen müssen vor ihrer Verwendung (Lesen / Schreiben) definiert worden sein. Folgende Regeln sind dabei zu beachten:

- GUD müssen in einer Definitionsdatei, z. B. `_N_DEF_DIR/_N_MGUD_DEF`, definiert werden.
- PUD und LUD müssen im Definitionsteil eines NC-Programms definiert werden.
- Die Datendefinition muss in einem eigenen Satz erfolgen.
- Pro Datendefinition darf nur ein Datentyp verwendet werden.
- Pro Datendefinition können mehrere Variable des gleichen Datentyps definiert werden.

Syntax

LUD und PUD

```
DEF <Typ> <Phys_Einheit> <Grenzwerte> <Name>[<Wert_1>, <Wert_2>, <Wert_3>]=<Init_Wert>
```

GUD

```
DEF <Bereich> <VL_Stop> <Zugriffsrechte> <Datenklasse> <Typ> <Phys_Einheit> <Grenzwerte> <Name>[<Wert_1>, <Wert_2>, <Wert_3>]=<Init_Wert>
```

Bedeutung

DEF:	Befehl zur Definition von Anwendervariablen GUD, PUD, LUD	
<Bereich>:	Gültigkeitsbereich, nur relevant für GUD:	
	NCK:	NC-globale Anwendervariable
	CHAN:	Kanal-globale Anwendervariable
<VL_Stop>:	Vorlaufstop, nur relevant für GUD (optional)	
	SYNR:	Vorlaufstop beim Lesen
	SYNW:	Vorlaufstop beim Schreiben
	SYNRW:	Vorlaufstop beim Lesen/Schreiben
<Zugriffsrechte>:	Schutzstufe für das Lesen / Schreiben von GUD über NC-Programm oder BTSS (optional)	
	APRP <Schutzstufe>:	Lesen: NC-Programm
	APWP <Schutzstufe>:	Schreiben: NC-Programm
	APRB <Schutzstufe>:	Lesen: BTSS
	APWB <Schutzstufe>:	Schreiben: BTSS
	<Schutzstufe>:	Wertebereich: 0 ... 7
	Siehe "Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB) (Seite 43)"	
<Datenklasse>:	Datenklassenzuordnung (nur SINUMERIK 828D!)	
	DCM:	Datenklasse M (= Manufacturer)
	DCI:	Datenklasse I (= Individual)
	DCU:	Datenklasse U (= User)
	Siehe "Attribut: Datenklasse (DCM, DCI, DCU) - nur SINUMERIK 828D (Seite 48)".	

<Typ>:	Datentyp:	
	INT:	Ganzzahliger Wert mit Vorzeichen
	REAL:	Real-Zahl (LONG REAL nach IEEE)
	BOOL:	Wahrheitswert TRUE (1) / FALSE (0)
	CHAR:	ASCII-Zeichen
	STRING[<MaxLänge>]:	Zeichenkette definierter Länge
	AXIS:	Achs-/Spindelbezeichner
	FRAME:	Geometrische Angaben für eine statische Koordinatentransformation
Siehe "Datentypen (Seite 56)"		
<Phys_Einheit>:	Physikalische Einheit (optional)	
	PHU <Einheit>:	physikalische Einheit
	Siehe "Attribut: Physikalische Einheit (PHU) (Seite 40)"	
<Grenzwerte>:	unterer / oberer Grenzwert (optional)	
	LLI <Grenzwert>:	unterer Grenzwert (lower limit)
	ULI <Grenzwert>:	oberer Grenzwert (upper limit)
	Siehe "Attribut: Grenzwerte (LLI, ULI) (Seite 39)"	
<Name>:	Name der Variablen Hinweis <ul style="list-style-type: none"> • Maximal 31 Zeichen • Die beiden ersten Zeichen müssen ein Buchstabe und/oder ein Unterstrich sein. • Das "\$"-Zeichen ist für Systemvariablen reserviert und darf nicht verwendet werden. 	
[<Wert_1>, <Wert_2>, <Wert_3>]:	Angabe der Feldgrößen für 1- bis max. 3-dimensionale Feldvariablen (optional) Zur Initialisierung von Feldvariablen siehe "Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 50)"	
<Init_Wert>:	Initialisierungswert (optional) siehe "Attribut: Initialisierungswert (Seite 36)" Zur Initialisierung von Feldvariablen siehe "Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 50)"	

Beispiele

Beispiel 1: Definitionen von Anwendervariablen im Datenbaustein für Maschinenhersteller

Programmcode	Kommentar
%_N_MGUD_DEF	; GUD-Baustein: Maschinenhersteller
\$PATH=/_N_DEF_DIR	
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2	
; Beschreibung	
; Definition zweier GUD: STROM_1, STROM_2	
; Gültigkeitsbereich: Kanalweit	

2.1 Variablen

Programmcode	Kommentar
<pre> ; Datentyp: REAL ; VL-Stop: nicht programmiert => Defaultwert = kein VL-Stop ; Phys. Einheit: 24 = [A] ; Grenzwerte: Low = 0.0, High = 10.0 ; Zugriffsrechte: nicht programmiert => Defaultwert = 7 = Schüsselschalterstellung 0 ; Initialisierungswert: nicht programmiert => Defaultwert = 0.0 DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45 ; Beschreibung ; Definition zweier GUD: ZEIT_1, ZEIT_2 ; Gültigkeitsbereich: NCK-weit ; Datentyp: REAL ; VL-Stop: nicht programmiert => Defaultwert = kein VL-Stop ; Phys. Einheit: 13 = [s] ; Grenzwerte: Low = 10.0, High = nicht programmiert => obere Definitionsbereichsgrenze ; Zugriffsrechte: ; NC-Programm: Schreiben/Lesen = 3 = Endanwender ; BTSS: Schreiben = 0 = Siemens, Lesen = 3 = Endanwender ; Initialisierungswert: ZEIT_1 = 12.0, ZEIT_2 = 45.0 DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER" ; Beschreibung ; Definition eines GUD: GUD5_NAME ; Gültigkeitsbereich: NCK-weit ; Datentyp: STRING, max. 5 Zeichen ; VL-Stop: nicht programmiert => Defaultwert = kein VL-Stop ; Phys. Einheit: nicht programmiert => Defaultwert = 0 = keine phys. Einheit ; Grenzwerte: nicht programmiert => Definitionsbereichsgrenzen: Low = 0, High = 255 ; Zugriffsrechte: ; NC-Programm: Schreiben/Lesen = 3 = Endanwender ; BTSS: Schreiben = 0 = Siemens, Lesen = 3 = Endanwender ; Initialisierungswert: "COUNTER" M30 </pre>	

Beispiel 2: Programm-globale und -lokale Anwendervariablen (PUD / LUD)

Programmcode	Kommentar
PROC MAIN	; Hauptprogramm
DEF INT VAR1	; PUD-Definition
...	
SUB2	; Unterprogrammaufruf
...	
M30	

Programmcode	Kommentar
PROC SUB2	; Unterprogramm SUB2
DEF INT VAR2	; LUD-DEFINITION
...	
IF (VAR1==1)	; PUD lesen
VAR1=VAR1+1	; PUD lesen und schreiben
VAR2=1	; LUD schreiben
ENDIF	
SUB3	; Unterprogrammaufruf
...	
M17	

Programmcode	Kommentar
PROC SUB3	; Unterprogramm SUB3
...	
IF (VAR1==1)	; PUD lesen
VAR1=VAR1+1	; PUD lesen und schreiben
VAR2=1	; Fehler: LUD aus SUB2 nicht bekannt
ENDIF	
...	
M17	

Beispiel 3: Definition und Verwendung von Anwendervariablen vom Datentyp AXIS

Programmcode	Kommentar
DEF AXIS ABSZISSE	; 1. Geometrieachse
DEF AXIS SPINDLE	; Spindel
...	
IF ISAXIS(1) == FALSE GOTO WEITER	
ABSZISSE = \$P_AXN1	
WEITER:	
...	
SPINDLE=(S1)	; 1. Spindel
OVRA[SPINDLE]=80	; Spindeloverride = 80%
SPINDLE=(S3)	; 3. Spindel

Randbedingungen

Globale Anwendervariablen (GUD)

Im Rahmen der Definition von globalen Anwendervariablen (GUD) sind folgende Maschinendaten zu berücksichtigen:

Nr.	Bezeichner: \$MN_	Bedeutung
11140	GUD_AREA_SAVE_TAB	zusätzliche Sicherung für GUD-Bausteine
18118 ¹⁾	MM_NUM_GUD_MODULES	Anzahl GUD-Dateien im aktiven Filesystem
18120 ¹⁾	MM_NUM_GUD_NAMES_NCK	Anzahl der globalen GUD-Namen
18130 ¹⁾	MM_NUM_GUD_NAMES_CHAN	Anzahl der kanalspez. GUD-Namen
18140 ¹⁾	MM_NUM_GUD_NAMES_AXIS	Anzahl der achsspez. GUD-Namen
18150 ¹⁾	MM_GUD_VALUES_MEM	Speicherplatz für globale GUD-Werte
18660 ¹⁾	MM_NUM_SYNACT_GUD_REAL	Anzahl projektierbare GUD Datentyp REAL
18661 ¹⁾	MM_NUM_SYNACT_GUD_INT	Anzahl projektierbare GUD Datentyp INT
18662 ¹⁾	MM_NUM_SYNACT_GUD_BOOL	Anzahl projektierbare GUD Datentyp BOOL
18663 ¹⁾	MM_NUM_SYNACT_GUD_AXIS	Anzahl projektierbare GUD Datentyp AXIS
18664 ¹⁾	MM_NUM_SYNACT_GUD_CHAR	Anzahl projektierbare GUD Datentyp CHAR
18665 ¹⁾	MM_NUM_SYNACT_GUD_STRING	Anzahl projektierbare GUD Datentyp STRING

¹⁾ MD bei SINUMERIK 828D nur lesbar!

Programmglobale Anwendervariablen (PUD)

Hinweis

Sichtbarkeit von programmglobalen Anwendervariablen (PUD)

Im Hauptprogramm definierte programmglobale Anwendervariablen (PUD) sind nur dann auch in den Unterprogrammen sichtbar, wenn folgendes Maschinendatum gesetzt ist:

```
MD11120 $MN_LUD_EXTENDED_SCOPE = 1
```

Mit MD11120 = 0 sind die im Hauptprogramm definierten programmglobalen Anwendervariablen nur im Hauptprogramm sichtbar.

Kanalübergreifende Verwendung einer NCK-globalen Anwendervariablen vom Datentyp AXIS

Eine NCK-globale Anwendervariable vom Datentyp `AXIS`, die bei der Definition im Datenbaustein mit einem Achsbezeichner initialisiert wurde, kann nur dann in unterschiedlichen Kanälen der NC verwendet werden, wenn die Achse in diesen Kanälen die gleiche Kanalachsnummer hat.

Ist dies nicht der Fall, muss die Variable am NC-Programmanfang geladen oder, wie im folgenden Beispiel, die Funktion `AXNAME(...)` (siehe "Achsfunktionen (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)" (Seite 559)) verwendet werden.

Programmcode	Kommentar
DEF NCK STRING[5] ACHSE="X"	; Definition im Datenbaustein
...	
N100 AX[AXNAME(ACHSE)]=111 G00	; Verwendung im NC-Programm

2.1.5 Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF)

Mit dem Befehl `REDEF` können die Attribute von Systemdaten, Anwenderdaten und NC-Sprachbefehle geändert werden. Grundvoraussetzung für eine Redefinition ist, dass sie zeitlich nach der entsprechenden Definition ausgeführt wird.

Bei einer Redefinition können nicht mehrere Attribute gleichzeitig geändert werden. Für jedes zu ändernde Attribut muss eine eigene `REDEF`-Anweisung programmiert werden.

Werden mehrere konkurrierende Attributänderungen programmiert, wird immer die letzte Änderung aktiv.

Attributwerte zurücksetzen

Die mit `REDEF` geänderten Attribute für Zugriffsrechte und Initialisierungszeitpunkt können durch erneute Programmierung von `REDEF`, gefolgt von dem Namen der Variablen oder des NC-Sprachbefehls, auf ihren Defaultwert zurückgesetzt werden:

- Zugriffsrechte: Schutzstufe 7
- Initialisierungszeitpunkt: keine Initialisierung bzw. Beibehalten des aktuellen Werts

Redefinierbare Attribute

Siehe "Übersicht definierbarer und redefinierbarer Attribute (Seite 49)".

Lokale Anwendervariablen (PUD / LUD)

Für lokale Anwendervariablen (PUD / LUD) dürfen keine Redefinitionen vorgenommen werden.

Syntax

```
REDEF <Name> <VL_Stop>
REDEF <Name> <Phys_Einheit>
REDEF <Name> <Grenzwerte>
REDEF <Name> <Zugriffsrechte>
REDEF <Name> <Init_Zeitpunkt>
REDEF <Name> <Init_Zeitpunkt> <Init_Wert>
REDEF <Name> <Datenklasse>
REDEF <Name>
```

Bedeutung

REDEF:	Befehl zur Redefinition eines bestimmten Attributs bzw. zum Zurücksetzen der Attribute "Zugriffsrechte" und/oder "Initialisierungszeitpunkt" von Systemvariablen, Anwendervariablen und NC-Sprachbefehlen
<Name>:	Name einer bereits definierten Variablen oder eines NC-Sprachbefehls

<VL-Stop>:	Vorlaufstop	
	SYNR:	Vorlaufstop beim Lesen
	SYNW:	Vorlaufstop beim Schreiben
	SYNRW:	Vorlaufstop beim Lesen/Schreiben
<Phys_Einheit>:	Physikalische Einheit	
	PHU <Einheit>:	physikalische Einheit
	Siehe "Attribut: Physikalische Einheit (PHU) (Seite 40)". Hinweis Nicht redefinierbar für: <ul style="list-style-type: none"> • Systemvariablen • Globale Anwenderdaten (GUD) der Datentypen: BOOL, AXIS, STRING, FRAME 	
<Grenzwerte>:	unterer / oberer Grenzwert	
	LLI <Grenzwert>:	unterer Grenzwert (lower limit)
	ULI <Grenzwert>:	oberer Grenzwert (upper limit)
	Siehe "Attribut: Grenzwerte (LLI, ULI) (Seite 39)". Hinweis Nicht redefinierbar für: <ul style="list-style-type: none"> • Systemvariablen • Globale Anwenderdaten (GUD) der Datentypen: BOOL, AXIS, STRING, FRAME 	
<Zugriffsrechte>:	Zugriffsrechte für das Lesen / Schreiben über Teileprogramm oder BTSS	
	APX <Schutzstufe>:	Ausführen: NC-Sprachelement
	APRP <Schutzstufe>:	Lesen: Teileprogramm
	APWP <Schutzstufe>:	Schreiben: Teileprogramm
	APRB <Schutzstufe>:	Lesen: BTSS
	APWB <Schutzstufe>:	Schreiben: BTSS
	<Schutzstufe>:	Wertebereich: 0 ... 7
	Siehe "Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB) (Seite 43)".	
<Init_Zeitpunkt>:	Zeitpunkt zu dem die Variable reinitialisiert wird	
	INIPO:	Power On
	INIRE:	Hauptprogrammende, NC-Reset oder Power On
	INICF:	NewConfig oder Hauptprogrammende, NC-Reset oder Power On
	PRLOC:	Hauptprogrammende, NC-Reset nach lokaler Änderung oder Power On
	Siehe "Attribut: Initialisierungswert (Seite 36)".	

Randbedingungen

Granularität

Eine Redefinition bezieht sich immer auf die gesamte, durch ihren Namen eindeutig gekennzeichnete Variable. Es ist nicht möglich z. B. bei Feldvariablen für einzelne Feldelemente unterschiedliche Attributwerte zuzuweisen.

2.1.6 Attribut: Initialisierungswert

Definition (**DEF**) von Anwendervariablen

Bei der Definition kann für folgende Anwendervariablen ein Initialisierungswert vorgegeben werden:

- globale Anwendervariablen (GUD)
- programmglobale Anwendervariablen (PUD)
- lokale Anwendervariablen (LUD)

Redefinition (**REDEF**) von System- und Anwendervariablen

Bei der Redefinition kann für folgende Variablen ein Initialisierungswert vorgegeben werden:

- Systemdaten
 - Settingdaten
- Anwenderdaten
 - R-Parameter
 - Synchronaktionsvariable (\$AC_MARKER, \$AC_PARAM, \$AC_TIMER)
 - Synchronaktions-GUD (SYG_xy[], mit x=R, I, B, A, C, S und y=S, M, U, 4, ..., 9)
 - EPS-Parameter
 - Werkzeugdaten-OEM
 - Magazindaten-OEM
 - globale Anwendervariablen (GUD)

Reinitialisierungszeitpunkt

Bei der Redefinition kann der Zeitpunkt angegeben werden, zu dem die Variable reinitialisiert, d. h. wieder auf den Initialisierungswert gesetzt werden soll:

- **INIPO (Power On)**
Die Variable wird bei Power On reinitialisiert.
- **INIRE (Reset)**
Die Variable wird bei NC-Reset, BAG-Reset, Teileprogrammende (M02 / M30) oder Power On reinitialisiert.

- **INICF (NewConfig)**
Die Variable wird bei NewConf-Anforderung über HMI, Teileprogramm-Befehl `NEWCONFIG` oder NC-Reset, BAG-Reset, Teileprogrammende (M02 / M30) oder Power On reinitialisiert.
- **PRLOC (programmlokale Änderung)**
Die Variable wird nur dann bei NC-Reset, BAG-Reset oder Teileprogrammende (M02 / M30) reinitialisiert, wenn sie im Rahmen des aktuellen Teileprogramms verändert worden ist.
Das Attribut `PRLOC` darf nur in Zusammenhang mit programmierbaren Settingdaten (siehe folgende Tabelle) verwendet werden.

Tabelle 2-1 Programmierbare Settingdaten

Nummer	Bezeichner	G-Befehl ¹⁾
42000	<code>\$SC_THREAD_START_ANGLE</code>	SF
42010	<code>\$SC_THREAD_RAMP_DISP</code>	DITS / DITE
42400	<code>\$SA_PUNCH_DWELLTIME</code>	PDELAYON
42800	<code>\$SA_SPIND_ASSIGN_TAB</code>	SETMS
43210	<code>\$SA_SPIND_MIN_VELO_G25</code>	G25
43220	<code>\$SA_SPIND_MAX_VELO_G26</code>	G26
43230	<code>\$SA_SPIND_MAX_VELO_LIMS</code>	LIMS
43300	<code>\$SA_ASSIGN_FEED_PER_REV_SOURCE</code>	FPRON
43420	<code>\$SA_WORKAREA_LIMIT_PLUS</code>	G26
43430	<code>\$SA_WORKAREA_LIMIT_MINUS</code>	G25
43510	<code>\$SA_FIXED_STOP_TORQUE</code>	FXST
43520	<code>\$SA_FIXED_STOP_WINDOW</code>	FXSW
43700	<code>\$SA_OSCILL_REVERSE_POS1</code>	OSP1
43710	<code>\$SA_OSCILL_REVERSE_POS2</code>	OSP2
43720	<code>\$SA_OSCILL_DWELL_TIME1</code>	OST1
43730	<code>\$SA_OSCILL_DWELL_TIME2</code>	OST2
43740	<code>\$SA_OSCILL_VELO</code>	FA
43750	<code>\$SA_OSCILL_NUM_SPARK_CYCLES</code>	OSNSC
43760	<code>\$SA_OSCILL_END_POS</code>	OSE
43770	<code>\$SA_OSCILL_CTRL_MASK</code>	OSCTRL
43780	<code>\$SA_OSCILL_IS_ACTIVE</code>	OS
43790	<code>\$SA_OSCILL_START_POS</code>	OSB
1) mit diesem G-Befehl wird das Settingdatum angesprochen		

Randbedingungen

Initialisierungswert: globale Anwendervariablen (GUD)

- Für globale Anwendervariable (GUD) mit dem Gültigkeitsbereich `NCK` kann als Initialisierungszeitpunkt nur `INIPO` (Power On) vorgegeben werden.
- Für globale Anwendervariablen (GUD) mit dem Gültigkeitsbereich `CHAN` kann als Initialisierungszeitpunkt neben `INIPO` (Power On) auch `INIRE` (Reset) oder `INICF` (NewConfig) vorgegeben werden.
- Bei globalen Anwendervariablen (GUD) mit dem Gültigkeitsbereich `CHAN` und Initialisierungszeitpunkt `INIRE` (Reset) oder `INICF` (NewConfig) werden bei NC-Reset, BAG-Reset und NewConfig die Variablen nur in den Kanälen neu initialisiert, in denen die genannten Ereignisse ausgelöst wurden.

Initialisierungswert: Datentyp `FRAME`

Für Variablen vom Datentyp `FRAME` darf kein Initialisierungswert angegeben werden. Variablen vom Datentyp `FRAME` werden implizit immer mit dem Defaultframe initialisiert.

Initialisierungswert: Datentyp `CHAR`

Für Variablen vom Datentyp `CHAR` kann statt des ASCII-Codes (0...255) auch das entsprechende ASCII-Zeichen in Anführungszeichen programmiert werden, z.B. "A"

Initialisierungswert: Datentyp `STRING`

Bei Variablen vom Datentyp `STRING` muss die Zeichenkette in Anführungszeichen gesetzt werden z.B.: ...= "MASCHINE_1"

Initialisierungswert: Datentyp `AXIS`

Für Variablen vom Datentyp `AXIS` muss bei erweiterter Adressschreibweise der Achsbezeichner in Klammern gesetzt werden, z.B.: ...=(X3)

Initialisierungswert: Systemvariable

Für Systemvariable können durch Redefinition keine anwenderspezifischen Initialisierungswerte vorgegeben werden. Die Initialisierungswerte der Systemvariablen sind vom System fest vorgegeben. Durch Redefinition kann aber der Zeitpunkt (`INIRE`, `INICF`) zu dem die Systemvariable reinitialisiert wird geändert werden.

Impliziter Initialisierungswert: Datentyp `AXIS`

Für Variablen vom Datentyp `AXIS` wird folgender implizite Initialisierungswert verwendet:

- Systemdaten: "erste Geometrieachse"
- Synchronaktions-GUD (Bezeichnung: SYG_A*), PUD, LUD:
Achsbezeichner aus Maschinendatum: MD20082
\$MC_AXCONF_CHANAX_DEFAULT_NAME

Impliziter Initialisierungswert: Werkzeug- und Magazindaten

Für Werkzeug- und Magazindaten können Initialisierungswerte über folgendes Maschinendatum vorgegeben werden: MD17520 \$MN_TOOL_DEFAULT_DATA_MASK

Hinweis**Synchronisation**

Die Synchronisation von Ereignissen die eine Reinitialisierung einer globalen Variable auslösen mit dem Lesen dieser Variable an anderer Stelle, liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers.

Siehe auch

Variablen (Seite 19)

2.1.7 Attribut: Grenzwerte (LLI, ULI)

Ein oberer und unterer Grenzwert des Definitionsbereichs kann nur für folgende Datentypen vorgegeben werden:

- INT
- REAL
- CHAR

Definition (DEF) von Anwendervariablen: Grenzwerte und implizite Initialisierungswerte

Wird bei der Definition einer Anwendervariablen von einem der oben genannten Datentypen kein expliziter Initialisierungswert definiert, wird die Variable auf den impliziten Initialisierungswert des Datentyps gesetzt:

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

Liegt der implizite Initialisierungswert außerhalb des durch die programmierten Grenzwerte festgelegten Definitionsbereichs, wird die Variable mit dem Grenzwert initialisiert, der dem impliziten Initialisierungswert am nächsten liegt:

- impliziter Initialisierungswert < unterer Grenzwert (LLI) ⇒
Initialisierungswert = unterer Grenzwert
- impliziter Initialisierungswert > oberer Grenzwert (ULI) ⇒
Initialisierungswert = oberer Grenzwert

Beispiele:

Programmcode	Kommentar
DEF REAL GUD1	; unterer Grenzwert = Definitionsbereichsgrenze ; oberer Grenzwert = Definitionsbereichsgrenze ; kein Initialisierungswert programmiert ; => impliziter Initialisierungswert = 0.0
DEF REAL LLI 5.0 GUD2	; unterer Grenzwert = 5.0 ; oberer Grenzwert = Definitionsbereichsgrenze ; => Initialisierungswert = 5.0
DEF REAL ULI -5 GUD3	; unterer Grenzwert = Definitionsbereichsgrenze ; oberer Grenzwert = -5.0 ; => Initialisierungswert = -5.0

Redefinition (REDEF) von Anwendervariablen: Grenzwerte und aktuelle Istwerte

Werden bei der Redefinition der Grenzwerte einer Anwendervariablen diese so geändert, dass der aktuelle Istwert außerhalb des neuen Definitionsbereichs liegt, erfolgt ein Alarm und die Grenzwerte werden nicht übernommen.

Hinweis

Redefinition (REDEF) von Anwendervariablen

Bei der Redefinition der Grenzwerte einer Anwendervariablen ist auf das konsistente Ändern der folgenden Werte zu achten:

- Grenzwerte
 - Istwert
 - Initialisierungswert beim Redefinieren und beim automatischen Reinitialisieren aufgrund von INIPO, INIRE oder INICF
-

Siehe auch

Variablen (Seite 19)

2.1.8 Attribut: Physikalische Einheit (PHU)

Eine physikalische Einheit kann nur für Variablen von folgende Datentypen vorgegeben werden:

- INT
- REAL

Programmierbare physikalische Einheiten (PHU)

Die Angabe der physikalische Einheit erfolgt als Festkommazahl: PHU <Einheit>

Folgende physikalische Einheiten können programmiert werden:

<Einheit>	Bedeutung	Physikalische Einheit
0	keine physikalische Einheit	-
1	Linear- oder Winkel-Position ¹⁾²⁾	[mm], [inch], [Grad]
2	Linear-Position ²⁾	[mm], [inch]
3	Winkel-Position	[Grad]
4	Linear- oder Winkel-Geschwindigkeit ¹⁾²⁾	[mm/min], [inch/min], [U/min]
5	Linear-Geschwindigkeit ²⁾	[mm/min]
6	Winkel-Geschwindigkeit	[U/min]
7	Linear- oder Winkel-Beschleunigung ¹⁾²⁾	[m/s ²], [inch/s ²], [U/s ²]
8	Linear-Beschleunigung ²⁾	[m/s ²], [inch/s ²]
9	Winkel-Beschleunigung	[U/s ²]
10	Linear- oder Winkel-Ruck ¹⁾²⁾	[m/s ³], [inch/s ³], [U/s ³]
11	Linear-Ruck ²⁾	[m/s ³], [inch/s ³]
12	Winkel-Ruck	[U/s ³]
13	Zeit	[s]
14	Lageregler-Verstärkung	[16.667/s]
15	Umdrehungsvorschub ²⁾	[mm/U], [inch/U]
16	Temperaturkompensation ¹⁾²⁾	[mm], [inch]
18	Kraft	[N]
19	Masse	[kg]
20	Trägheitsmoment ³⁾	[kgm ²]
21	Prozent	[%]
22	Frequenz	[Hz]
23	Spannung	[V]
24	Strom	[A]
25	Temperatur	[°C]
26	Winkel	[Grad]
27	KV	[1000/min]
28	Linear- oder Winkel-Position ³⁾	[mm], [inch], [Grad]
29	Schnittgeschwindigkeit ²⁾	[m/min], [feet/min]
30	Umfangsgeschwindigkeit ²⁾	[m/s], [feet/s]
31	Widerstand	[Ohm]
32	Induktivität	[mH]
33	Drehmoment ³⁾	[Nm]
34	Drehmomentkonstante ³⁾	[Nm/A]
35	Stromreglerverstärkung	[V/A]
36	Drehzahlreglerverstärkung ³⁾	[Nm/(rad*s)]
37	Drehzahl	[U/min]
42	Leistung	[kW]
43	Strom, klein	[µA]
46	Drehmoment, klein ³⁾	[µNm]
48	Promille	-

2.1 Variablen

<Einheit>	Bedeutung	Physikalische Einheit
49	-	[Hz/s]
65	Durchfluss	[l/min]
66	Druck	[bar]
67	Volumen ³⁾	[cm ³]
68	Streckenverstärkung ³⁾	[mm/(V*min)]
69	Streckenverstärkung Kraftregler	[N/V]
155	Gewindesteigung ³⁾	[mm/U], [inch/U]
156	Gewindesteigungsänderung ³⁾	[mm/U / U], [inch/U / U]

- 1) Die physikalische Einheit ist abhängig vom Achstyp: Linear- oder Rundachse
- 2) Maßsystem-Umschaltung
 G70/G71(inch/metrisch)
 Nach einer Umschaltung des Grundsystems (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) mit G70/G71 erfolgt bei Schreib/Lesezugriffen auf längenbehaftete System- und Anwendervariablen **keine** Umrechnung der Werte (Istwert, Defaultwert und Grenzwerte)
 G700/G710(inch/metrisch)
 Nach einer Umschaltung des Grundsystems (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) mit G700/G710 erfolgt bei Schreib/Lesezugriffen auf längenbehaftete System- und Anwendervariablen **eine** Umrechnung der Werte (Istwert, Defaultwert und Grenzwerte)
- 3) Die Variable wird **nicht** automatisch in das aktuelle Maßsystem der NC (inch/metrisch) umgerechnet. Die Umrechnung liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers.

Hinweis

Ebenenüberlauf durch Formatumrechnung

Das interne Ablageformat für alle Anwendervariablen (GUD / PUD / LUD) mit längenbehafteten physikalischen Einheiten ist metrisch. Eine exzessive Verwendung derartiger Variablen im Hauptlauf des NCK, z. B. in Synchronaktionen, kann bei einer Maßsystemumschaltung zu einem Rechenzeitüberlauf der Interpolatorebene, Alarm 4240, führen.

Hinweis

Kompatibilität von Einheiten

Bei der Verwendung von Variablen (Zuweisung, Vergleich, Berechnung etc.) erfolgt keine Prüfung auf Kompatibilität der beteiligten Einheiten. Eine gegebenenfalls erforderliche Umrechnung liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers.

Siehe auch

Variablen (Seite 19)

2.1.9 Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB)

Den Zugriffsrechten entsprechen folgende bei der Programmierung anzugebende Schutzstufen:

Zugriffsrecht	Schutzstufe
Kennwort System	0
Kennwort Maschinenhersteller	1
Kennwort Service	2
Kennwort Endanwender	3
Schlüsselschalter Stellung 3	4
Schlüsselschalter Stellung 2	5
Schlüsselschalter Stellung 1	6
Schlüsselschalter Stellung 0	7

Definition (DEF) von Anwenderdaten

Zugriffsrechte (APR... / APW...) können für folgende Daten definiert werden:

- globale Anwenderdaten (GUD)

Redefinition (**REDEF**) von System- und Anwenderdaten

Zugriffsrechte (**APR...** / **APW...**) können für folgende Daten redefiniert werden:

- Systemdaten
 - Maschinendaten
 - Settingdaten
 - Systemvariable
 - Prozessdaten
 - Magazindaten
 - Werkzeugdaten
- Anwenderdaten
 - R-Parameter
 - Synchronaktionsvariable (**\$AC_MARKER**, **\$AC_PARAM**, **\$AC_TIMER**)
 - Synchronaktions-GUD (**SYG_xy[]**, mit x=R, I, B, A, C, S und y=S, M, U, 4, ..., 9)
 - EPS-Parameter
 - Werkzeugdaten-OEM
 - Magazindaten-OEM
 - globale Anwendervariablen (GUD)

Hinweis

Bei der Redefinition kann das Zugriffsrecht auf eine Variable zwischen der niedrigsten Schutzstufe 7 und der eigenen Schutzstufe, z. B. 1 (Maschinenhersteller), frei vergeben werden.

Redefinition (**REDEF**) von NC-Sprachbefehlen

Das Zugriffs- bzw. Ausführungsrecht (**APX**) kann für folgende NC-Sprachbefehle redefiniert werden:

- G-Funktionen / Wegbedingungen
 - Literatur
 - Programmierhandbuch Grundlagen; Kapitel: G-Funktionen / Wegbedingungen
- Vordefinierte Funktionen
 - Literatur
 - Programmierhandbuch Grundlagen; Kapitel: Vordefinierte Funktionen
- Vordefinierte Unterprogrammaufrufe
 - Literatur
 - Programmierhandbuch Grundlagen; Kapitel: Vordefinierte Unterprogrammaufrufe
- Anweisung **DO** bei Synchronaktionen
- Programmbezeichner von Zyklen
 - Der Zyklus muss in einem Zyklenverzeichnis abgelegt sein und eine **PROC**-Anweisung enthalten.

Zugriffsrechte bezüglich NC-Programmen und Zyklen (APRP, APWP)

Die unterschiedlichen Zugriffsrechte haben für den Zugriff aus einem NC-Programm bzw. Zyklus folgende Auswirkungen:

- APRP 0 / APWP 0
 - Beim Abarbeiten des NC-Programms muss das System-Kennwort gesetzt sein
 - Der Zyklus muss im Verzeichnis `_N_CST_DIR` (System) abgelegt sein
 - Für das Verzeichnis `_N_CST_DIR` muss im MD11160 `$MN_ACCESS_EXEC_CST` das Ausführungsrecht auf System eingestellt sein
- APRP 1 / APWP 1 bzw. APRP 2 / APWP 2
 - Beim Abarbeiten des NC-Programms muss das Maschinenhersteller- bzw. Service-Kennwort gesetzt sein
 - Der Zyklus muss im Verzeichnis `_N_CMA_DIR` (Maschinenhersteller) oder `_N_CST_DIR` abgelegt sein
 - Für die Verzeichnisse `_N_CMA_DIR` bzw. `_N_CST_DIR` müssen in den Maschinendaten MD11161 `$MN_ACCESS_EXEC_CMA` bzw. MD11160 `$MN_ACCESS_EXEC_CST` die Ausführungsrechte mindestens auf Maschinenhersteller eingestellt sein
- APRP 3 / APWP 3
 - Beim Abarbeiten des NC-Programms muss das Endanwender-Kennwort gesetzt sein
 - Der Zyklus muss im Verzeichnis `_N_CUS_DIR` (Anwender), `_N_CMA_DIR` oder `_N_CST_DIR` abgelegt sein
 - Für die Verzeichnisse `_N_CUS_DIR`, `_N_CMA_DIR` bzw. `_N_CST_DIR` müssen in den Maschinendaten MD11162 `$MN_ACCESS_EXEC_CUS`, MD11161 `$MN_ACCESS_EXEC_CMA` bzw. MD11160 `$MN_ACCESS_EXEC_CST` die Ausführungsrechte mindestens auf Endanwender eingestellt sein
- APRP 4...7 / APWP 4...7
 - Beim Abarbeiten des NC-Programms muss Schüsselschalterstellung 3 ... 0 eingestellt sein
 - Der Zyklus muss im Verzeichnis `_N_CUS_DIR`, `_N_CMA_DIR` oder `_N_CST_DIR` abgelegt sein
 - Für die Verzeichnisse `_N_CUS_DIR`, `_N_CMA_DIR` bzw. `_N_CST_DIR` müssen in den Maschinendaten MD11162 `$MN_ACCESS_EXEC_CUS`, MD11161 `$MN_ACCESS_EXEC_CMA` bzw. MD11160 `$MN_ACCESS_EXEC_CST` die Ausführungsrechte mindestens auf die entsprechende Schüsselschalterstellung eingestellt sein

Zugriffsrechte bezüglich BTSS (APRB, APWB)

Die Zugriffsrechte (APRB, APWB) beschränken den Zugriff auf System- und Anwendervariablen über BTSS für alle Systemkomponenten (HMI, PLC, externe Rechner, EPS-Dienste, etc.) gleichermaßen.

Hinweis

HMI-lokale Zugriffsrechte

Bei Änderungen von Zugriffsrechten von Systemdaten muss darauf geachtet werden, dass diese konsistent zu den über HMI-Mechanismen festgelegten Zugriffsrechten erfolgt.

Zugriffsattribute APR / APW

Aus Kompatibilitätsgründen werden die Attribute APR und APW implizit auf die Attribute APRP / APRB und APWP / APWB abgebildet:

- APR x \Rightarrow APRP x APRB x
- APW y \Rightarrow APWP y APWB y

Zugriffsrechte über ACCESS-Dateien

Bei der Verwendung von ACCESS-Dateien für die Vergabe von Zugriffsrechten, dürfen Redefinitionen von Zugriffsrechten für Systemdaten, Anwenderdaten und NC-Sprachbefehlen ausschließlich in ACCESS-Dateien erfolgen. Eine Ausnahme bilden globale Anwenderdaten (GUD). Für diese muss eine Redefinition der Zugriffsrechte weiterhin in den entsprechenden Definitionsdateien *_DEF erfolgen.

Für einen durchgehenden Zugriffsschutz müssen die Maschinendaten für die Ausführungsrechte und den Zugriffsschutz der entsprechenden Verzeichnisse konsistenten angepasst werden.

Daraus ergibt sich folgende prinzipielle Vorgehensweise:

1. Erstellen der benötigten Definitionsdateien:
 - _N_DEF_DIR/_N_SACCESS_DEF
 - _N_DEF_DIR/_N_MACCESS_DEF
 - _N_DEF_DIR/_N_UACCESS_DEF
2. Parametrieren des Schreibrechts für die Definitionsdateien auf den für die Redefinition erforderlichen Wert:
 - MD11170 \$MN_ACCESS_WRITE_SACCESS = <Schutzstufe>
 - MD11171 \$MN_ACCESS_WRITE_MACCESS = <Schutzstufe>
 - MD11172 \$MN_ACCESS_WRITE_UACCESS = <Schutzstufe>

3. Für Zugriffe auf geschützte Elemente aus Zyklen heraus müssen die Ausführungs- und Schreibrechte der Zyklenverzeichnisse `_N_CST_DIR`, `_N_CMA_DIR` und `_N_CST_DIR` angepasst werden:

Ausführungsrechte

- MD11160 `$MN_ACCESS_EXEC_CST` = <Schutzstufe>
- MD11161 `$MN_ACCESS_EXEC_CMA` = <Schutzstufe>
- MD11162 `$MN_ACCESS_EXEC_CUS` = <Schutzstufe>

Schreibrechte

- MD11165 `$MN_ACCESS_WRITE_CST` = <Schutzstufe>
- MD11166 `$MN_ACCESS_WRITE_CMA` = <Schutzstufe>
- MD11167 `MN_ACCESS_WRITE_CUS` = <Schutzstufe>

Das Ausführungsrecht muss mindestens auf die gleiche Schutzstufe wie die höchste Schutzstufe des verwendeten Elements gesetzt werden.

Das Schreibrecht muss mindestens auf die gleiche Schutzstufe wie das Ausführungsrecht gesetzt werden.

4. Die Schreibrechte der HMI-lokalen Zyklenverzeichnisse müssen auf die gleiche Schutzstufe wie die der NC-lokalen Zyklenverzeichnisse gesetzt werden.

Literatur

Bedienhandbuch

Unterprogrammaufrufe in ACCESS-Dateien

Für die weitere Strukturierung des Zugriffsschutzes können in den ACCESS-Dateien auch Unterprogramme (Kennung SPF oder MPF) aufgerufen werden. Die Unterprogramme erben dabei die Ausführungsrechte der aufrufenden ACCESS-Datei.

Hinweis

In den ACCESS-Dateien können nur die Zugriffsrechte redefiniert werden. Alle anderen Attribute müssen weiterhin in den entsprechenden Definitionsdateien programmiert bzw. redefiniert werden.

Siehe auch

Variablen (Seite 19)

2.1.10 Attribut: Datenklasse (DCM, DCI, DCU) - nur SINUMERIK 828D

Zur Vereinfachung des Datenhandlings im Rahmen der Inbetriebnahme, Serieninbetriebnahme und Hochrüstung von Maschinen und Maschinenreihen sind alle System- und Anwenderdaten der NC in Datenklassen unterteilt.

Datenklasse	Daten
S = System	Von Siemens bereitgestellte Systemdaten wie Maschinen- und Settingdaten, Standard- und Messzyklen, Definitionen (SGUD) und Makros (SMAC) etc.
M = Manufacturer (Maschinenhersteller)	Maschinenreihen-spezifische Inbetriebnahmedaten wie Herstellerzyklen, Definitionen (MGUD) und Makros (MMAC) und Maschinendaten, die den Funktionsumfang der Maschine festlegen.
I = Individual (Maschinen-spezifisch)	Maschinen-spezifische Inbetriebnahmedaten wie Kompensationsdaten und Referenzpunktverschiebungen.
U = User (Anwender)	Maschinen-spezifische Daten, die im Betrieb der Maschine erzeugt werden wie Werkzeugdaten, Settingdaten, Teileprogramme, Anwenderzyklen, Definitionen (UGUD) und Makros (UMAC).

Literatur:

SINUMERIK 828D Inbetriebnahmehandbuch Drehen und Fräsen; Kap.: "Einführung und Anwendung von Datenklassen"

Definition (DEF) von Anwenderdaten

Durch die Datenklasse der Datei oder des Verzeichnisses, in dem ein Anwenderdatum definiert wird, ist implizit die Datenklasse des Datums festgelegt. Die Datenklasse des Datums kann nicht geändert werden.

Bei der Definition (DEF) des Anwenderdatums kann aber für den **Datenwert** eine eigene, von der Datenklasse des Datums abweichende Datenklasse, vorgegeben werden.

Für die Datenklasse des Datenwerts muss gelten:

Priorität der Datenklasse des Datenwerts ≤ Priorität der Datenklasse des Datums

Beispiel:

Die Definition des GUD, das einen Messtaster beschreibt, soll sich in der Datenklasse M (= Manufacturer) befinden, da es für den Ablauf der Herstellerzyklen notwendig ist. Der Wert des Datums soll aber, da der Typ des Messtasters von Maschine zu Maschine unterschiedlich sein kann, der Datenklasse I (= Individual) angehören.

MGUD.DEF (Datenklasse M)

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


Redefinition (REDEF) von Systemdaten

Die Datenklasse von Systemdaten kann durch Redefinition (REDEF) geändert werden. Die Redefinition muss in einer Definitionsdatei mit Datenklasse S oder M erfolgen.

Bei Verwendung von ACCESS-Dateien dürfen Redefinitionen nur innerhalb der ACCESS-Dateien erfolgen.

Die jeweilige Datenklasse der Maschinen-, Setting- und Optionsdaten sowie Systemvariablen findet sich in:

- Listenhandbuch Ausführliche Maschinendatenbeschreibung, Parameter: "Klasse"
- Listenhandbuch Systemvariablen

2.1.11 Übersicht definierbarer und redefinierbarer Attribute

Die folgenden Tabellen zeigen bei welchen Datenarten welche Attribute definiert (DEF) und/oder redefiniert (REDEF) werden können.

Systemdaten

Datenart	Init.Wert	Grenzwerte	phys. Einheit	Zugriffsrechte	Datenklasse (nur 828D)
Maschinendaten	---	---	---	REDEF	REDEF
Settingdaten	REDEF	---	---	REDEF	---
FRAME-Daten	---	---	---	REDEF	---
Prozessdaten	---	---	---	REDEF	---
Spindelsteigungsfehlerkomp. (EEC)	---	---	---	REDEF	---
Durchhangkompensation (CEC)	---	---	---	REDEF	---
Quadrantenfehlerkompensation (QEC)	---	---	---	REDEF	---
Magazindaten	---	---	---	REDEF	---
Werkzeugdaten	---	---	---	REDEF	---
Schutzbereiche	---	---	---	REDEF	---
orientierbare Werkzeugträger	---	---	---	REDEF	---
kinematische Ketten	---	---	---	REDEF	---
3D-Schutzbereiche	---	---	---	REDEF	---
Arbeitsfeldbegrenzung	---	---	---	REDEF	---

Anwenderdaten

Datenart	Init.Wert	Grenzwerte	phys. Einheit	Zugriffsrechte	Datenklasse
R-Parameter	REDEF	REDEF	REDEF	REDEF	---
Synchronaktionsvariable (\$AC_...)	REDEF	REDEF	REDEF	REDEF	---
Synchronaktions-GUD (SYG_...)	REDEF	REDEF	REDEF	REDEF	---
EPS-Parameter	REDEF	REDEF	REDEF	REDEF	---

Datenart	Init.Wert	Grenzwerte	phys. Einheit	Zugriffsrechte	Datenklasse
Werkzeugdaten–OEM	REDEF	REDEF	REDEF	REDEF	---
Magazindaten–OEM	REDEF	REDEF	REDEF	REDEF	---
globale Anwendervariablen (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF	DEF / REDEF
lokale Anwendervariablen (PUD / LUD)	DEF	DEF	DEF	---	---

Siehe auch

Variablen (Seite 19)

2.1.12 Definition und Initialisierung von Feldvariablen (DEF, SET, REP)

Eine Anwendervariable kann als 1- bis maximal 3-dimensionales Feld (Array) definiert werden:

- 1-dimensional: DEF <Datentyp> <Variablenname> [<n>]
- 2-dimensional: DEF <Datentyp> <Variablenname> [<n>, <m>]
- 3-dimensional: DEF <Datentyp> <Variablenname> [<n>, <m>, <o>]

Hinweis

Anwendervariable vom Datentyp STRING können maximal als 2-dimensionales Feld definiert werden.

Datentypen

Anwendervariable können als Felder für folgende Datentypen definiert werden: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Wertzuzuweisung an Feldelemente

Wertzuzuweisungen an Feldelemente können zu folgenden Zeitpunkten vorgenommen werden:

- bei der Felddefinition (Initialisierungswerte)
- während des Programmablaufs

Wertzuzuweisung können dabei erfolgen über:

- explizite Angabe eines Feldelements
- explizite Angabe eines Feldelements als Startelement und Angabe einer Werteliste (SET)
- explizite Angabe eines Feldelements als Startelement und Angabe eines Wertes und der Häufigkeit seiner Wiederholung (REP)

Hinweis

Anwendervariablen vom Datentyp FRAME können keine Initialisierungswerte zugewiesen werden.

Syntax (DEF)

```
DEF <Datentyp> <Variablenname>[<n>,<m>,<o>]
DEF STRING[<Stringlänge>] <Variablenname>[<n>,<m>]
```

Syntax (DEF...=SET...)

Verwendung einer Werteliste:

- bei der Definition:


```
DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] =
SET(<Wert1>,<Wert2>,...)
gleichbedeutend mit:
DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] =
(<Wert1>,<Wert2>,...)
```

Hinweis

Bei der Initialisierung über eine Werteliste ist die Angabe von SET optional.

- bei einer Wertzuweisung:


```
<Variablenname>[<n>,<m>,<o>] = SET(<WERT1>,<Wert2>,...)
```

Syntax (DEF...=REP...)

Verwendung eines Werte mit Wiederholung

- bei der Definition:


```
DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] = REP(<Wert>)
DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] =
REP(<Wert>,<Anzahl_Feldelemente>)
```
- bei einer Wertzuweisung:


```
<Variablenname>[<n>,<m>,<o>] = REP(<Wert>)
<Variablenname>[<n>,<m>,<o>] = REP(<Wert>,<Anzahl_Feldelemente>)
```

Bedeutung

DEF:	Befehl zur Definition von Variablen
<Datentyp>:	Datentyp der Variablen
	Wertebereich: <ul style="list-style-type: none"> • bei Systemvariablen: BOOL, CHAR, INT, REAL, STRING, AXIS • bei GUD- oder LUD-Variablen: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME
<Stringlänge>:	Maximale Anzahl der Zeichen beim Datentyp STRING
<Variablenname>:	Variablenname
[<n>,<m>,<o>]:	Feldgrößen bzw. Feldindizes

<n>:	Feldgröße bzw. Feldindex für 1. Dimension	
	Typ:	INT (bei Systemvariablen auch AXIS)
	Wertebereich:	Max. Feldgröße: 65535 Feldindex: $0 \leq n \leq 65534$
<m>:	Feldgröße bzw. Feldindex für 2. Dimension	
	Typ:	INT (bei Systemvariablen auch AXIS)
	Wertebereich:	Max. Feldgröße: 65535 Feldindex: $0 \leq m \leq 65534$
<o>:	Feldgröße bzw. Feldindex für 3. Dimension	
	Typ:	INT (bei Systemvariablen auch AXIS)
	Wertebereich:	Max. Feldgröße: 65535 Feldindex: $0 \leq o \leq 65534$
SET:	Wertzuzuweisung über die angegebenen Werteliste	
(<Wert1>, <Wert2>, ...):	Werteliste	
REP:	Wertzuzuweisung über den angegebenen <Wert>	
<Wert>:	Wert, mit dem die Feldelemente bei der Initialisierung mit REP beschrieben werden sollen.	
<Anzahl_Feldelemente>:	<p>Anzahl der Feldelemente, die mit dem angegebenen <Wert> beschrieben werden sollen. Für die restlichen Feldelemente gilt abhängig vom Zeitpunkt:</p> <ul style="list-style-type: none"> Initialisierung bei der Felddefinition: → Die restlichen Feldelemente werden mit Null beschrieben Zuzuweisung während des Programmlaufs: → Die aktuellen Werte der Feldelemente bleiben unverändert. <p>Ist der Parameter nicht programmiert, werden alle Feldelemente mit <Wert> beschrieben.</p> <p>Ist der Parameter gleich Null, gilt abhängig vom Zeitpunkt:</p> <ul style="list-style-type: none"> Initialisierung bei der Felddefinition: → Alle Elemente werden mit Null vorbelegt Zuzuweisung während des Programmlaufs: → Die aktuellen Werte der Feldelemente bleiben unverändert. 	

Feldindex

Die implizite Reihenfolge der Feldelemente z. B. bei einer Wertzuzuweisung über SET oder REP erfolgt durch Iteration der Feldindizes von rechts nach links.

Beispiel: Initialisierung eines 3-dimensionalen Feldes mit 24 Feldelementen:

```

DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1      1. Feldelement
  FELD[0,0,1] = 1      2. Feldelement
  FELD[0,0,2] = 1      3. Feldelement
  FELD[0,0,3] = 1      4. Feldelement
    
```

```

...
FELD[0,1,0] = 1      5. Feldelement
FELD[0,1,1] = 1      6. Feldelement
...
FELD[0,2,3] = 1      12. Feldelement
FELD[1,0,0] = 1      13. Feldelement
FELD[1,0,1] = 1      14. Feldelement
...
FELD[1,2,3] = 1      24. Feldelement

```

entsprechend:

```

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

```

Beispiel: Initialisierung kompletter Variablenfelder

Aktuelle Belegung siehe Abbildung.

Programmcode

```

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)

```

Feldindex

	N10: Initialisierung bei Definition			N20/N30: Initialisierung mit identischem Wert			N40/N50: Initialisierung mit verschiedenen Werten		
[1,2]	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
	Die Feldelemente [5,0] bis [9,2] wurden mit dem Defaultwert (0.0) initialisiert.						Die Feldelemente [3,1] bis [4,0] wurden mit dem Defaultwert (0.0) initialisiert. Die Feldelemente [6,0] bis [8,0] wurden nicht verändert.		

Siehe auch

Definition und Initialisierung von Feldvariablen (DEF, SET, REP): Weitere Informationen (Seite 54)

Variablen (Seite 19)

2.1.13 Definition und Initialisierung von Feldvariablen (DEF, SET, REP): Weitere Informationen

Weitere Informationen (SET)

Initialisierung bei der Definition

- Es werden, beginnend beim 1. Feldelement, so viele Feldelemente mit den Werten aus der Werteliste initialisiert, wie Elemente in der Werteliste programmiert sind.
- Feldelemente ohne explizit angegebene Werte in der Werteliste (Lücken in der Werteliste) werden mit 0 belegt.
- Bei Variablen vom Datentyp AXIS sind Lücken in der Werteliste nicht zugelassen.
- Enthält die Werteliste mehr Werte als Feldelemente definiert sind, wird ein Alarm angezeigt.

Wertzuweisung im Programmablauf

Bei der Wertzuweisung im Programmablauf gelten die oben bei der Definition beschriebenen Regeln. Zusätzlich gibt es folgende Möglichkeiten:

- Als Elemente in der Werteliste sind auch Ausdrücke erlaubt.
- Die Wertzuweisung beginnt bei dem programmierten Feldindex. Hierdurch lassen sich gezielt Teilfelder mit Werten belegen.

Beispiel:

Programmcode	Kommentar
DEF INT FELD[5,5]	; Felddefinition
FELD[0,0]=SET(1,2,3,4,5)	; Wertzuweisung an die ersten 5 Feldelemente [0,0] - [0,4]
FELD[0,0]=SET(1,2, , ,5)	; Wertzuweisung mit Lücke an die ersten 5 Feldelemente [0,0] - [0,4], Feldelemente [0,2] und [0,3] = 0
FELD[2,3]=SET(VARIABLE,4*5.6)	; Wertzuweisung mit Variable und Ausdruck ab Feldindex [2,3]: [2,3] = VARIABLE [2,4] = 4 * 5.6 = 22.4

Weitere Informationen (REP)

Initialisierung bei der Definition

- Alle oder die optional angegebene Anzahl an Feldelementen werden mit dem angegebenen Wert (Konstante) initialisiert.
- Variablen vom Datentyp FRAME können nicht initialisiert werden.

Beispiel:

Programmcode	Kommentar
DEF REAL varName[10]=REP(3.5,4)	; Felddefinition und Feldelemente [0] bis [3] mit Wert 3,5 initialisieren

Wertzuweisung im Programmablauf

Bei der Wertzuweisung im Programmablauf gelten die oben bei der Definition beschriebenen Regeln. Zusätzlich gibt es folgende Möglichkeiten:

- Als Elemente in der Werteliste sind auch Ausdrücke erlaubt.
- Die Wertzuweisung beginnt bei dem programmierten Feldindex. Hierdurch lassen sich gezielt Teilfelder mit Werten belegen.

Beispiele:

Programmcode	Kommentar
DEF REAL varName[10]	; Felddefinition
varName[5]=REP(4.5,3)	; Feldelemente [5] bis [7] = 4,5
R10=REP(2.4,3)	; R-Parameter R10 bis R12 = 2,4
DEF FRAME FRM[10]	; Felddefinition

2.1 Variablen

Programmcode	Kommentar
FRM[5]=REP(CTTRANS(X,5))	; Feldelemente [5] bis [9] = CTRANS(X,5)

Siehe auch

Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 50)

2.1.14 Datentypen

Folgende Datentypen stehen in der NC zur Verfügung:

Datentyp	Bedeutung	Wertebereich
INT	ganzzahliger Wert mit Vorzeichen	-2147483648 ... +2147483647
REAL	Real-Zahl (LONG REAL nach IEEE)	$\pm(\sim 2,2 \cdot 10^{-308} \dots \sim 1,8 \cdot 10^{+308})$
BOOL	Wahrheitswert TRUE (1) und FALSE (0)	1, 0
CHAR	ASCII-Zeichen	ASCII-Code 0 ... 255
STRING	Zeichenkette definierter Länge	maximal 200 Zeichen (keine Sonderzeichen)
AXIS	Achs-/Spindelbezeichner	Kanalachsbezeichner
FRAME	Geometrische Angaben für eine statische Koordinatentransformation (Verschieben, Drehen, Skalieren, Spiegeln)	---

Implizite Datentypwandlungen

Folgende Datentypwandlungen sind möglich und werden bei Zuweisungen und Parameterübergaben implizit vorgenommen:

von ↓ / nach →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x

x: ohne Einschränkungen möglich
o: Datenverlust durch Überschreitung des Wertebereichs möglich ⇒ Alarm;
Rundung: Nachkommawert ≥ 0,5 ⇒ aufrunden, Nachkommawert < 0,5 ⇒ abrunden
&: Wert ≠ 0 ⇒ TRUE, Wert == 0 ⇒ FALSE

Siehe auch

Variablen (Seite 19)

2.1 Variablen

Programmcode	Kommentar
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	; IS_VAR ist in diesem Fall TRUE, ist ein zweidimensionales Feld.
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR ist TRUE, Variable existiert.
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR ist FALSE, Feldindex ist nicht erlaubt.
N40 IS_VAR=ISVAR("VARARRAY[8,8]")	; IS_VAR ist FALSE, "]" fehlt (Syntaxfehler).
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR ist TRUE, Feldindex ist erlaubt.
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; IS_VAR ist TRUE, Feldindex ist erlaubt.

Programmcode	Kommentar
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("\$MC_GCODE_RESET_VALUES[1]")	; Übergabeparameter ist ein Maschindatum, IS_VAR ist TRUE.

Programmcode	Kommentar
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("\$P_EP")	; IS_VAR ist in diesem Fall TRUE.
N20 IS_VAR=ISVAR("\$P_EP[X]")	; IS_VAR ist in diesem Fall TRUE.

2.1.16 Attributwerte/Datentyp lesen (GETVARPHU, GETVARAP, GETVARLIM, GETVARDFT, GETVARTYP)

Mit den vordefinierten Funktionen GETVARPHU, GETVARAP, GETVARLIM und GETVARDFT können die Attributwerte von System-/Anwendervariablen gelesen werden, mit GETVARTYP der Datentyp einer System-/Anwendervariablen.

Physikalische Einheit lesen

Syntax:

<Ergebnis>=GETVARPHU (<Name>)

Bedeutung:

<Ergebnis>:	Zahlenwert der physikalischen Einheit	
	Datentyp:	INT
	Wertebereich:	siehe Tabelle in "Attribut: Physikalische Einheit (PHU) (Seite 40)"
		Im Fehlerfall:
	- 2	Der angegebene <Name> ist keinem Systemparameter und keiner Anwendervariablen zugeordnet.

GETVARPHU:	Lesen der physikalischen Einheit einer System- /Anwendervariablen	
<Name>:	Name der System- /Anwendervariablen	
	Datentyp:	STRING

Beispiel:

Der NCK enthalte folgende GUD-Variable:

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

Programmcode	Kommentar
DEF INT result=0	
result=GETVARPHU("elect- ric")	; Ermittle die physikalische Einheit der GUD-Variablen.
IF (result < 0) GOTOF error	

Als Ergebnis wird der Wert 42 zurückgeliefert. Dies entspricht der physikalischen Einheit [kW].

Hinweis

Mit GETVARPHU lässt sich z. B. prüfen, ob bei einer Variablenzuweisung a = b beide Variablen die erwarteten physikalischen Einheiten haben.

Zugriffsrecht lesen**Syntax:**

```
<Ergebnis>=GETVARAP (<Name>, <Zugriff>)
```

Bedeutung:

<Ergebnis>:	Schutzstufe für den angegebenen <Zugriff>		
	Datentyp:	INT	
	Wertebereich:	0 ... 7	Siehe "Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB) (Seite 43)".
		Im Fehlerfall:	
		- 2	Der angegebene <Name> ist keinem Systemparameter und keiner Anwendervariablen zugeordnet.
	- 3	falscher Wert für <Zugriff>	
GETVARAP:	Lesen des Zugriffsrechts auf eine System- /Anwendervariable		
<Name>:	Name der System- /Anwendervariablen		
	Datentyp:	STRING	
<Zugriff>:	Art des Zugriffs		
	Datentyp:	STRING	
	Wertebereich:	"RP":	Lesen über Teileprogramm
		"WP":	Schreiben über Teileprogramm
		"RB":	Lesen über BTSS
"WB":		Schreiben über BTSS	

Beispiel:

Programmcode	Kommentar
DEF INT result=0	
result=GETVARAP	; Ermittle den Zugriffsschutz für den Systemparameter
RAP("\$TC_MAP8", "WB")	"Magazinposition" bezüglich Schreiben über BTSS.
IF (result < 0) GOTOF error	

Als Ergebnis wird der Wert 7 zurückgeliefert. Dies entspricht der Schlüsselschalter-Stellung 0 (= kein Zugriffsschutz).

Hinweis

Mit GETVARAP kann z. B. ein Prüfprogramm realisiert werden, das die von der Applikation erwarteten Zugriffsrechte prüft.

Grenzwerte lesen

Syntax:

<Status>=GETVARLIM(<Name>, <Grenzwert>, <Ergebnis>)

Bedeutung:

<Status>:	Funktionsstatus			
	Datentyp:	INT		
	Wertebereich:	1	o. k.	
		-1	kein Grenzwert definiert (bei Variablen vom Typ AXIS, STRING, FRAME)	
		-2	Der angegebene <Name> ist keinem Systemparameter und keiner Anwendervariablen zugeordnet.	
-3		falscher Wert für <Grenzwert>		
GETVARLIM:	Lesen des unteren/oberen Grenzwerts einer System- /Anwendervariablen			
<Name>:	Name der System- /Anwendervariablen			
	Datentyp:	STRING		
<Grenzwert>:	Gibt an, welcher Grenzwert ausgelesen werden soll			
	Datentyp:	CHAR		
	Wertebereich:	"L"	= unterer Grenzwert	
		":		
":	"U"	= oberer Grenzwert		
	":			
<Ergebnis>:	Rückgabe des Grenzwerts			
	Datentyp:	VAR REAL		

Beispiel:

Programmcode	Kommentar
DEF INT state=0	

Programmcode	Kommentar
<pre> DEF REAL result=0 state=GETVARLIM("\$MA_MAX_AX_VE- LO","L",result) IF (result < 0) GOTOF error </pre>	<pre> ; Ermittle den unteren Grenzwert für MD32000 \$MA_MAX_AX_VELO. </pre>

Standardwert lesen

Syntax:

```
<Status>=GETVARDFT(<Name>,<Ergebnis>[,<Index_1>,<Index_2>,<Index_3>]
)
```

Bedeutung:

<Status>:	Funktionsstatus		
	Datentyp:	INT	
	Wertebereich:	1	o. k.
		-1	kein Standardwert verfügbar (z. B. weil <Ergebnis> den falschen Typ zu <Name> hat)
		-2	Der angegebene <Name> ist keinem Systemparameter und keiner Anwendervariablen zugeordnet.
		-3	falscher Wert für <Index_1>, Dimension kleiner eins (= kein Array = Skalar)
		-4	falscher Wert für <Index_2>
-5	falscher Wert für <Index_3>		
GETVARDFT:	Lesen des Standardwerts einer System- /Anwendervariablen		
<Name>:	Name der System-/Anwendervariablen		
	Datentyp:	STRING	
<Ergebnis>:	Rückgabe des Standardwerts		
	Datentyp:	VAR REAL (beim Lesen des Standardwerts von Variablen der Typen INT, REAL, BOOL, AXIS)	
		VAR STRING (beim Lesen des Standardwerts von Variablen der Typen STRING und CHAR)	
		VAR FRAME (beim Lesen des Standardwerts von Variablen des Typs FRAME)	
<Index_1>:	Index auf erste Dimension (optional)		
	Datentyp:	INT	
	Nicht programmiert bedeutet = 0		
<Index_2>:	Index auf zweite Dimension (optional)		
	Datentyp:	INT	
	Nicht programmiert bedeutet = 0		
<Index_3>:	Index auf dritte Dimension (optional)		
	Datentyp:	INT	
	Nicht programmiert bedeutet = 0		

Beispiel:

Programmcode	Kommentar
DEF INT state=0	
DEF REAL resultR=0	; Variable zur Aufnahme der Standardwerte der Typen INT, REAL, BOOL, AXIS.
DEF FRAME resultF=0	; Variable zur Aufnahme der Standardwerte des Typs FRAME
IF (GETVARTYP("\$MA_MAX_AX_VELO") <> 4) GOTO error	
state=GETVARDFT("\$MA_MAX_AX_VELO", resultR, AXTOINT(X))	; Ermittle den Standardwert der "X"-Achse.
IF (resultR < 0) GOTO error	
IF (GETVARTYP("\$TC_TP8") <> 3) GOTO error	
state=GETVARDFT("\$TC_TP8", resultR)	
IF (GETVARTYP("\$P_UBFR") <> 7) GOTO error	
state=GETVARDFT("\$P_UBFR", resultF)	

Datentyp lesen

Syntax:

<Ergebnis>=GETVARTYP (<Name>)

Bedeutung:

<Ergebnis>:	Datentyp der angegebenen System- /Anwendervariablen			
	Datentyp:	INT		
	Wertebereich:	1	= BOOL	
		2	= CHAR	
		3	= INT	
		4	= REAL	
		5	= STRING	
		6	= AXIS	
		7	= FRAME	
	Im Fehlerfall:			
< 0	Der angegebene <Name> ist keinem Systemparameter und keiner Anwendervariablen zugeordnet.			
GETVARTYP:	Lesen des Datentyps einer System- /Anwendervariablen			
<Name>:	Name der System- /Anwendervariablen			
	Datentyp:	STRING		

Beispiel:

Programmcode	Kommentar
DEF INT result=0	
DEF STRING name="R"	
result=GETVARTYP(name)	; Ermittle den Typ des R-Parameters.
IF (result < 0) GOTOF error	

Als Ergebnis wird der Wert 4 zurückgeliefert. Dies entspricht dem Datentyp REAL.

2.2 Indirekte Programmierung

2.2.1 Indirekte Programmierung von Adressen

Bei der indirekten Programmierung von Adressen wird die erweiterte Adresse (<Index>) durch eine Variable geeigneten Typs ersetzt.

Hinweis

Die indirekte Programmierung von Adressen ist nicht möglich bei:

- N (Satznummer)
- L (Unterprogramm)
- Einstellbaren Adressen
(z. B. X[1] anstelle von X1 ist nicht zulässig)

Syntax

<ADRESSE>[<Index>]

Bedeutung

<ADRESSE>[...]:	Feste Adresse mit Erweiterung (Index)
<Index>:	Variable z. B. für Spindelnummer, Achse, ...

Beispiele**Beispiel 1: Indirekte Programmierung einer Spindelnummer**

Direkte Programmierung:

Programmcode	Kommentar
S1=300	; Drehzahl 300 U/min für die Spindel mit Nummer 1.

Indirekte Programmierung:

Programmcode	Kommentar
DEF INT SPINU=1	; Definition der Variablen vom Typ INT und Wertzuweisung.
S[SPINU]=300	; Drehzahl 300 U/min für die Spindel, deren Nummer in der Variablen SPINU abgelegt ist (in diesem Beispiel die Spindel mit Nummer 1).

Beispiel 2: Indirekte Programmierung einer Achse

Direkte Programmierung:

Programmcode	Kommentar
FA[U]=300	; Vorschub 300 für die Achse "U".

Indirekte Programmierung:

Programmcode	Kommentar
DEF AXIS AXVAR2=U	; Definition einer Variablen vom Typ AXIS und Wertzuweisung.
FA[AXVAR2]=300	; Vorschub 300 für die Achse, deren Adressname in der Variablen mit dem Namen AXVAR2 abgelegt ist.

Beispiel 3: Indirekte Programmierung einer Achse

Direkte Programmierung:

Programmcode	Kommentar
\$AA_MM[X]	; Messtaster-Messwert (MKS) der Achse "X" lesen.

Indirekte Programmierung:

Programmcode	Kommentar
DEF AXIS AXVAR3=X	; Definition einer Variablen vom Typ AXIS und Wertzuweisung.
\$AA_MM[AXVAR3]	; Messtaster-Messwert (MKS) lesen für die Achse, deren Name in der Variablen AXVAR3 abgelegt ist.

Beispiel 4: Indirekte Programmierung einer Achse

Direkte Programmierung:

Programmcode	Kommentar
X1=100 X2=200	

Indirekte Programmierung:

Programmcode	Kommentar
DEF AXIS AXVAR1 AXVAR2	; Definition zweier Variablen vom Typ AXIS.
AXVAR1=(X1) AXVAR2=(X2)	; Zuweisung der Achsnamen.
AX[AXVAR1]=100 AX[AXVAR2]=200	; Verfahren der Achsen, deren Adressnamen in den Variablen mit den Namen AXVAR1 und AXVAR2 abgelegt sind.

Beispiel 5: Indirekte Programmierung einer Achse

Direkte Programmierung:

Programmcode
G2 X100 I20

Indirekte Programmierung:

Programmcode	Kommentar
DEF AXIS AXVAR1=X	; Definition einer Variablen vom Typ AXIS und Wertzuweisung.
G2 X100 IP[AXVAR1]=20	; Indirekte Programmierung der Mittelpunktsangabe für die Achse, deren Adressname in der Variablen mit dem Namen AXVAR1 abgelegt ist

Beispiel 6: Indirekte Programmierung von Feldelementen

Direkte Programmierung:

Programmcode	Kommentar
DEF INT FELD1[4,5]	; Definition von Feld 1.

Indirekte Programmierung:

Programmcode	Kommentar
DEFINE DIM1 AS 4	; Bei Felddimensionen müssen Feldgrößen als feste Werte angegeben werden.
DEFINE DIM2 AS 5	
DEF INT FELD[DIM1,DIM2]	
FELD[DIM1-1,DIM2-1]=5	

Beispiel 7: Indirekter Unterprogrammaufruf

Programmcode	Kommentar
CALL "L" << R10	; Aufruf des Programms, dessen Nummer in R10 steht (Stringverkettung).

2.2.2 Indirekte Programmierung von G-Codes

Die indirekte Programmierung von G-Codes ermöglicht eine effektive Zyklenprogrammierung.

Syntax

G[<Gruppe>]=<Nummer>

Bedeutung

G[...]:	G-Befehl mit Erweiterung (Index)	
<Gruppe>:	Index-Parameter: G-Funktionsgruppe	
	Typ:	INT
<Nummer>:	Variable für die G-Code-Nummer	
	Typ:	INT oder REAL

Hinweis

Es können i. d. R. nur nicht-syntaxbestimmende G-Codes indirekt programmiert werden.
 Von den syntaxbestimmenden G-Codes sind nur die der G-Funktionsgruppe 1 möglich.
 Die syntaxbestimmenden G-Codes der G-Funktionsgruppen 2, 3 und 4 sind nicht möglich.

Hinweis

In der indirekten G-Code-Programmierung sind keine Arithmetik-Funktionen erlaubt. Eine notwendige Berechnung der G-Code-Nummer muss in einer eigenen Teileprogrammzeile vor der indirekten G-Code-Programmierung erfolgen.

Beispiele

Beispiel 1: Einstellbare Nullpunktverschiebung (G-Funktionsgruppe 8)

Programmcode	Kommentar
N1010 DEF INT INT_VAR	
N1020 INT_VAR=2	
...	
N1090 G[8]=INT_VAR G1 X0 Y0	; G54
N1100 INT_VAR=INT_VAR+1	; G-Code-Berechnung
N1110 G[8]=INT_VAR G1 X0 Y0	; G55

Beispiel 2: Ebenenanwahl (G-Funktionsgruppe 6)

Programmcode	Kommentar
N2010 R10=\$P_GG[6]	; Aktive G-Funktion der G-Funktions-Gruppe 6 lesen
...	
N2090 G[6]=R10	

Literatur

Informationen zu den G-Funktionsgruppen siehe:
 Programmierhandbuch Grundlagen; Kapitel "G-Funktionsgruppen"

2.2.3 Indirekte Programmierung von Positionsattributen (GP)

Positionsattribute, wie z. B. die inkrementelle oder absolute Programmierung der Achsposition, können in Verbindung mit dem Schlüsselwort `GP` indirekt als Variablen programmiert werden.

Anwendung

Die indirekte Programmierung von Positionsattributen findet Verwendung in **Ersetzungszyklen**, da hier folgender Vorteil gegenüber der Programmierung von Positionsattributen als Schlüsselwort (z. B. `IC`, `AC`, ...) besteht:

Durch die indirekte Programmierung als Variablen wird **keine** `CASE`-Anweisung benötigt, die über alle möglichen Positionsattribute verzweigt.

Syntax

```
<POSITIONIERBEFEHL> [<Achse/Spindel>]=
GP (<Position>, <Positionsattribut>)
<Achse/Spindel>=GP (<Position>, <Positionsattribut>)
```

Bedeutung

<POSITIONIERBEFEHL> []:	Folgende Positionierbefehle können zusammen mit dem Schlüsselwort <code>GP</code> programmiert werden: <code>POS</code> , <code>POSA</code> , <code>SPOS</code> , <code>SPOSA</code> Außerdem möglich: <ul style="list-style-type: none"> alle im Kanal vorhandenen Achs-/Spindelbezeichner: <Achse/Spindel> variabler Achs-/Spindelbezeichner <code>AX</code>
<Achse/Spindel>:	Achse/Spindel, die positioniert werden soll
<code>GP ()</code> :	Schlüsselwort zur Positionierung
<Position>:	Parameter 1 Achs-/Spindelposition als Konstante oder Variable
<Positionsattribut>:	Parameter 2 Positionsattribut (z. B. Positionsanfahrmodus) als Variable (z. B. <code>\$P_SUB_SPOSMODE</code>) oder als Schlüsselwort (<code>IC</code> , <code>AC</code> , ...)

Die von den Variablen gelieferten Werte haben folgende Bedeutung:

Wert	Bedeutung	Zulässig bei:
0	Keine Änderung des Positionsattributs	
1	<code>AC</code>	<code>POS</code> , <code>POSA</code> , <code>SPOS</code> , <code>SPOSA</code> , <code>AX</code> , Achsadresse
2	<code>IC</code>	<code>POS</code> , <code>POSA</code> , <code>SPOS</code> , <code>SPOSA</code> , <code>AX</code> , Achsadresse
3	<code>DC</code>	<code>POS</code> , <code>POSA</code> , <code>SPOS</code> , <code>SPOSA</code> , <code>AX</code> , Achsadresse
4	<code>ACP</code>	<code>POS</code> , <code>POSA</code> , <code>SPOS</code> , <code>SPOSA</code> , <code>AX</code> , Achsadresse
5	<code>ACN</code>	<code>POS</code> , <code>POSA</code> , <code>SPOS</code> , <code>SPOSA</code> , <code>AX</code> , Achsadresse
6	<code>OC</code>	-

Wert	Bedeutung	Zulässig bei:
7	PC	-
8	DAC	POS, POSA, AX, Achsadresse
9	DIC	POS, POSA, AX, Achsadresse
10	RAC	POS, POSA, AX, Achsadresse
11	RIC	POS, POSA, AX, Achsadresse
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Beispiel

Bei einer aktiven Synchronspindelkopplung zwischen der Leitspindel S1 und der Folgespindel S2 wird durch den SPOS-Befehl im Hauptprogramm der folgende Ersetzungszyklus zur Positionierung der Spindeln aufgerufen.

Die Positionierung erfolgt über die Anweisung in N2230:

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

Die anzufahrende Position wird aus der Systemvariablen \$P_SUB_SPOSIT, der Positionsanfahrmodus wird aus der Systemvariablen \$P_SUB_SPOSMODE gelesen.

Programmcode	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
...	
N2100 IF(\$P_SUB_AXFCT==2)	
N2110	; Ersetzung des SPOS / SPOSA / M19-Befehls bei aktiver Synchronspindelkopplung
N2185 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N2190 COUPOF(S2,S1)	; Synchronspindelkopplung deaktivieren
N2200	; Leit- und Folgespindel positionieren
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220	; Spindel mit SPOS positionieren:
N2230 SPOS[1]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
SPOS[2]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
N2250 ELSE	
N2260	; Spindel mit M19 positionieren:
N2270 M1=19 M2=19	; Leit- und Folgespindel positionieren
N2280 ENDIF	
N2285 DELAYFSTOF	; Ende Stopp-Delay-Bereich
N2290 COUPON(S2,S1)	; Synchronspindelkopplung aktivieren
N2410 ELSE	

Programmcode	Kommentar
N2420	; Abfrage auf weitere Ersetzungen
...	
N3300 ENDIF	
...	
N9999 RET	

Randbedingungen

- In Synchronaktionen ist die indirekte Programmierung von Positionsattributen nicht möglich.

Literatur

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1), Kapitel: Ersetzung von NC-Funktionen durch Unterprogramme

2.2.4 Indirekte Programmierung von Teileprogrammzeilen (EXECSTRING)

Mit dem Teileprogrammbehl `EXECSTRING` ist es möglich, eine zuvor erzeugte String-Variablen als Teileprogrammzeile auszuführen.

Syntax

`EXECSTRING` wird in einer eigenen Teileprogrammzeile programmiert:
`EXECSTRING(<String-Variablen>)`

Bedeutung

<code>EXECSTRING:</code>	Befehl zur Ausführung einer String-Variablen als Teileprogrammzeile
<code><String-Variablen>:</code>	Variablen vom Typ <code>STRING</code> , die die eigentlich auszuführende Teileprogrammzeile enthält

Hinweis

Mit `EXECSTRING` können mit Ausnahme von Kontrollstrukturen (Seite 106) alle Teileprogramm-Konstrukte abgesetzt werden, die **im Programmteil eines Teileprogramms** programmiert werden können. Ausgeschlossen sind damit `PROC-` und `DEF-`Anweisungen sowie generell die Verwendung in `INI-` und `DEF-`Dateien.

Beispiel

Programmcode	Kommentar
N100 DEF STRING[100] MY_BLOCK	; Definition der String-Variablen zur Aufnahme der auszuführenden Teileprogrammzeile.
N110 DEF STRING[10] MFCT1="M7"	

2.3 Rechenfunktionen

Programmcode	Kommentar
...	
N200 EXECSTRING(MFCT1 << "M4711")	; Teileprogrammzeile "M7 M4711" ausführen.
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	; Teileprogrammzeile "M3 M7" ausführen.

2.3 Rechenfunktionen

Operator / Rechenfunktion	Bedeutung
+	Addition
-	Subtraktion
*	Multiplikation
/ ¹⁾	Division ¹⁾
DIV ¹⁾	Ganzzahl-Division ¹⁾
MOD ¹⁾	Modulo-Division (Liefert den Rest der Ganzzahl-Division) ¹⁾
:	Kettungsoperator für FRAME-Variablen
SIN ()	Sinus
COS ()	Cosinus
TAN ()	Tangens
ASIN ()	Arcussinus
ACOS ()	Arcuscosinus
ATAN2 (,) ¹⁾	Arcustangens2 ¹⁾
SQRT ()	Quadratwurzel
ABS ()	Betrag
POT ()	2. Potenz (Quadrat)
TRUNC ()	ganzzahliger Teil Genauigkeiten bei Vergleichsbefehlen einstellbar mit TRUNC (siehe "Genauigkeitskorrektur bei Vergleichsfehlern (TRUNC) (Seite 74)")
ROUND ()	Runden auf ein Ganzzahliges
LN ()	natürlicher Logarithmus
EXP ()	Exponentialfunktion
MINVAL ()	kleinerer Wert zweier Variablen (siehe "Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND) (Seite 76)")
MAXVAL ()	größerer Wert zweier Variablen (siehe "Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND) (Seite 76)")

BOUND ()	Variablenwert, der im definierten Wertebereich liegt (siehe "Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND) (Seite 76)")
CTRANS ()	Verschiebung
CROT ()	Drehung
CSCALE ()	Maßstabsveränderung
CMIRROR ()	Spiegeln
1) Siehe Absatz "Beispiele"	

Programmierung

Bei den Rechenfunktionen gilt die übliche mathematische Schreibweise. Prioritäten in der Abarbeitung werden durch runde Klammern gesetzt. Für die trigonometrischen und deren inverse Funktionen gilt die Gradangabe (rechter Winkel = 90°).

Beispiele

Division: /

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

Beispiel: $3 / 4 = 0.75$

Ganzzahl-Division: DIV

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

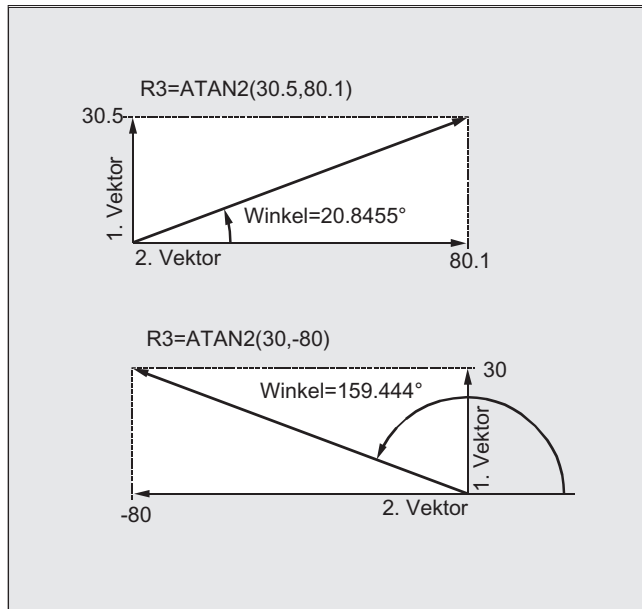
Beispiel: $7 \text{ DIV } 4.1 = 1$

Modulo-Division (Liefert den Rest der Ganzzahl-Division): MOD

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

Beispiel: $7 \text{ MOD } 4.1 = 2.9$

Arcustangens2: ATAN2



Die Rechenfunktion ATAN2 berechnet aus zwei aufeinander senkrecht stehenden Vektoren den Winkel des Summenvektors.

Das Ergebnis liegt im Bereich von vier Quadranten (-180° < 0 < +180°).

Basis für den Winkelbezug ist immer der 2. Wert in positiver Richtung.

Programmierbeispiele

Programmcode	Kommentar
R1=R1+1	; Neues R1 = altes R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; Punktrechnung geht vor Strichrechnung.
R14=(R1+R2)*R3	; Geclammerte Ausdrücke werden zuerst berechnet.
R15=SQRT(POT(R1)+POT(R2))	; Innere Klammern werden zuerst aufgelöst: R15 = Quadratwurzel((R1^2 + R2^2))
RESFRAME=FRAME1:FRAME2	; FRAME-Verknüpfung mit Kettungsoperator
FRAME3=CTRANS(...):CROT(...)	Wertzuzuweisung an eine FRAME-Komponente

2.4 Vergleichs- und logische Operationen

Vergleichsoperationen können z. B. zur Formulierung einer Sprungbedingung benutzt werden. Vergleichbar sind dabei auch komplexe Ausdrücke.

Die Vergleichsoperationen sind für Variable vom Typ CHAR, INT, REAL und BOOL anwendbar. Beim Typ CHAR wird der Codewert verglichen.

Bei den Typen STRING, AXIS und FRAME sind möglich: == und <>, die für Operationen vom Typ STRING auch in Synchronaktionen angewendet werden können.

Das Ergebnis von vergleichenden Operationen ist immer vom Typ BOOL.

Logische Operatoren dienen zur Verknüpfung von Wahrheitswerten.

Die logischen Operationen sind nur auf Variable vom Typ BOOL anwendbar. Über interne Typenkonvertierung sind sie auch auf die Datentypen CHAR, INT, und REAL anwendbar.

Bei den logischen (boolschen) Operationen gilt für die Datentypen `BOOL`, `CHAR`, `INT` und `REAL`:

- 0 entspricht: `FALSE`
- ungleich 0 entspricht: `TRUE`

Bitweise logische Operatoren

Mit den Variablen vom Typ `CHAR` und `INT` können auch bitweise logische Operationen vorgenommen werden. Gegebenenfalls erfolgt eine Typkonvertierung automatisch.

Programmierung

Vergleichsoperator	Bedeutung
<code>==</code>	gleich
<code><></code>	ungleich
<code>></code>	größer
<code><</code>	kleiner
<code>>=</code>	größer oder gleich
<code><=</code>	kleiner oder gleich

Logischer Operator	Bedeutung
<code>AND</code>	UND
<code>OR</code>	ODER
<code>NOT</code>	Negation
<code>XOR</code>	Exklusiv-ODER

Bitweise logischer Operator	Bedeutung
<code>B_AND</code>	bitweises UND
<code>B_OR</code>	bitweises ODER
<code>B_NOT</code>	bitweise Negation
<code>B_XOR</code>	bitweises Exklusiv-ODER

Hinweis

In arithmetischen Ausdrücken kann durch runde Klammern die Abarbeitungsreihenfolge aller Operatoren festgelegt und damit von den normalen Prioritätsregeln abgewichen werden.

Hinweis

Zwischen BOOLSCHEN Operanden und Operatoren müssen Zwischenräume geschrieben werden.

Hinweis

Der Operator `B_NOT` bezieht sich auf nur einen Operanden. Dieser steht nach dem Operator.

Beispiele

Beispiel 1: Vergleichsoperatoren

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

oder

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

Das Ergebnis des Vergleichs `R10 >= 100` wird zunächst in `R11` zwischengespeichert.

Beispiel 2: Logische Operatoren

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

oder

```
IF NOT R10 GOTOB START
```

`NOT` bezieht sich nur auf einen Operanden.

Beispiel 3: Bitweise logische Operatoren

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

2.5 Genauigkeitskorrektur bei Vergleichsfehlern (TRUNC)

Der `TRUNC`-Befehl schneidet den mit einem Genauigkeitsfaktor multiplizierten Operanden ab.

Einstellbare Genauigkeit bei Vergleichsbefehlen

Teileprogrammdateien vom Typ `REAL` werden intern im `IEEE`-Format mit 64 Bit dargestellt. Aufgrund dieser Darstellungsform können Dezimalzahlen ungenau abgebildet werden, die bei einem Vergleich mit ideal gerechneten Werten zu unerwarteten Ergebnissen führen können.

Relative Gleichheit

Damit die durch die Darstellungsform hervorgerufenen Ungenauigkeiten den Programmfluß nicht verfälschen, wird bei den Vergleichsbefehlen nicht auf absolute Gleichheit, sondern auf eine relative Gleichheit geprüft.

Syntax

Genauigkeitskorrektur bei Vergleichsfehlern

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

Bedeutung

TRUNC:	Abschneiden der Nachkommastellen
--------	----------------------------------

Berücksichtigte relative Gleichheit von 10^{-12} bei

- Gleichheit: (==)
- Ungleichheit: (<>)
- Größer-Gleich: (>=)
- Kleiner-Gleich: (<=)
- Größer/Kleiner: (><) mit absoluter Gleichheit
- Größer: (>)
- Kleiner: (<)

Kompatibilität

Aus Kompatibilitätsgründen kann die Prüfung auf relative Gleichheit bei (>) und (<) durch Setzen von Maschinendatum MD10280 \$MN_PROG_FUNCTION_MASK Bit0 = 1 deaktiviert werden.

Hinweis

Vergleiche mit Daten vom Typ REAL sind aus den genannten Gründen generell mit einer gewissen Ungenauigkeit behaftet. Bei nicht akzeptablen Abweichungen muss auf INTEGER-Rechnung ausgewichen werden, indem die Operanden mit einem Genauigkeitsfaktor multipliziert und danach mit TRUNC abgeschnitten werden.

Synchronaktionen

Das beschriebene Verhalten der Vergleichsbefehle gilt auch bei Synchronaktionen.

Beispiele**Beispiel 1: Genauigkeitsbetrachtungen**

Programmcode	Kommentar
N40 R1=61.01 R2=61.02 R3=0.01	; Zuweisung der Anfangswerte
N41 IF ABS(R2-R1) > R3 GOTOF FEHLER	; Sprung würde bisher ausgeführt werden
N42 M30	; Programmende
N43 FEHLER: SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	; Zuweisung der Anfangswerte
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)	; Genauigkeitskorrektur
R13=TRUNC(R3*1000)	
IF ABS(R12-R11) > R13 GOTOF FEHLER	; Sprung wird nicht mehr ausgeführt
M30	; Programmende
FEHLER: SETAL(66000)	

Beispiel 2: Quotient beider Operanden bilden und auswerten

Programmcode	Kommentar
R1=61.01 R2=61.02 R3=0.01	; Zuweisung der Anfangswerte

Programmcode	Kommentar
IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF FEHLER	; Sprung wird nicht ausgeführt
M30	; Programmende
FEHLER: SETAL(66000)	

2.6 Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND)

Mit den Befehlen `MINVAL` und `MAXVAL` können die Werte zweier Variablen miteinander verglichen werden. Als Ergebnis wird der kleinere Wert (bei `MINVAL`) bzw. größere Wert (bei `MAXVAL`) zurückgeliefert.

Mit dem Befehl `BOUND` kann geprüft werden, ob der Wert einer Prüfvariablen innerhalb eines definierten Wertebereichs liegt.

Syntax

```
<Kleinerer Wert>=MINVAL(<Variable1>,<Variable2>)
<Größerer Wert>=MAXVAL(<Variable1>,<Variable2>)
<Rückgabewert>=<BOUND>(<Minimum>,<Maximum>,<Prüfvariable>)
```

Bedeutung

MINVAL:	Ermittelt den kleineren Wert zweier Variablen (<Variable1>, <Variable2>)
<Kleinerer Wert>:	Ergebnisvariable für den Befehl <code>MINVAL</code> Wird auf den kleineren Variablenwert gesetzt.
MAXVAL:	Ermittelt den größeren Wert zweier Variablen (<Variable1>, <Variable2>)
<Größerer Wert>:	Ergebnisvariable für den Befehl <code>MAXVAL</code> Wird auf den größeren Variablenwert gesetzt.
BOUND:	Prüft, ob eine Variable (<Prüfvariable>) innerhalb eines definierten Wertebereichs liegt.
<Minimum>:	Variable, die den Minimalwert des Wertebereichs definiert
<Maximum>:	Variable, die den Maximalwert des Wertebereichs definiert
<Rückgabewert>:	Ergebnisvariable für den Befehl <code>BOUND</code> Wenn der Wert der Prüfvariablen innerhalb des definierten Wertebereichs liegt, dann wird die Ergebnisvariable auf den Wert der Prüfvariablen gesetzt. Wenn der Wert der Prüfvariablen größer als der Maximalwert ist, dann wird die Ergebnisvariable auf den Maximalwert des Definitionsbereichs gesetzt. Wenn der Wert der Prüfvariablen kleiner als der Minimalwert ist, dann wird die Ergebnisvariable auf den Minimalwert des Definitionsbereichs gesetzt.

Hinweis

MINVAL, MAXVAL und BOUND können auch in Synchronaktionen programmiert werden.

Hinweis**Verhalten bei Gleichheit**

Bei Gleichheit wird bei MINVAL/MAXVAL dieser gleiche Wert geliefert. Bei BOUND wird der Wert der zu prüfenden Variablen wieder zurückgegeben.

Beispiel

Programmcode	Kommentar
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar	
rValMin=MINVAL(rVar1,rVar2)	; rValMin wird auf den Wert 10.5 gesetzt.
rValMax=MAXVAL(rVar1,rVar2)	; rValMax wird auf den Wert 33.7 gesetzt.
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 liegt innerhalb der Grenzen, rRetVar wird auf 19.7 gesetzt.
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 liegt unterhalb der Minimumgrenze, rRetVar wird auf 10.5 gesetzt.
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 liegt oberhalb der Maximumgrenze, rRetVar wird auf 33.7 gesetzt.

2.7 Priorität der Operationen

Jedem Operator ist eine Priorität zugeordnet. Bei der Auswertung eines Ausdrucks werden stets die Operatoren höherer Priorität zuerst angewandt. Bei gleichrangigen Operatoren erfolgt die Auswertung von links nach rechts.

In arithmetischen Ausdrücken kann durch runde Klammern die Abarbeitungsreihenfolge aller Operatoren festgelegt und damit von den normalen Prioritätsregeln abgewichen werden.

Reihenfolge der Operatoren**Von der höchsten zur niedrigsten Priorität**

1.	NOT, B_NOT	Verneinung, bitweise Verneinung
2.	*, /, DIV, MOD	Multiplikation, Division
3.	+, -	Addition, Subtraktion
4.	B_AND	bitweises UND
5.	B_XOR	bitweises exklusives ODER
6.	B_OR	bitweises ODER

7.	AND	UND
8.	XOR	exklusives ODER
9.	OR	ODER
10.	<<	Verkettung von Strings, Ergebnistyp STRING
11.	==, <>, >, <, >=, <=	Vergleichsoperatoren

Hinweis

Der Kettungsoperator ":" für Frames darf nicht mit anderen Operatoren in einem Ausdruck vorkommen. Eine Prioritätseinstufung für diesen Operator ist deshalb nicht erforderlich.

Beispiel If-Anweisung

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

2.8 Mögliche Typenkonvertierungen

Funktion

Typkonvertierung bei Zuweisung

Der konstante Zahlenwert, die Variable oder der Ausdruck, der einer Variablen zugewiesen wird, muss mit dem Typ dieser Variablen verträglich sein. Ist dies gegeben, so wird bei der Zuweisung der Typ automatisch umgewandelt.

Mögliche Typkonvertierungen

nach	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
von							
REAL	ja	ja*	ja ¹⁾	ja*	-	-	-
INT	ja	ja	ja ¹⁾	ja ²⁾	-	-	-
BOOL	ja	ja	ja	ja	ja	-	-
CHAR	ja	ja	ja ¹⁾	ja	ja	-	-
STRING	-	-	ja ⁴⁾	ja ³⁾	ja	-	-
AXIS	-	-	-	-	-	ja	-
FRAME	-	-	-	-	-	-	ja

Erklärungen

- * Bei Typumwandlung von REAL nach INT wird bei gebrochenem Wert >= 0.5 aufgerundet, ansonsten wird abgerundet (vgl. Funktion ROUND)
- 1) Wert <> 0 entspricht TRUE, Wert == 0 entspricht FALSE
- 2) Wenn der Wert im zulässigen Zahlenbereich liegt
- 3) Wenn nur 1 Zeichen
- 4) Stringlänge 0 = >FALSE, ansonsten TRUE

Hinweis

Ist beim Konvertieren ein Wert größer als der Zielbereich, erfolgt eine Fehlermeldung.

Treten in einem Ausdruck gemischte Typen auf, so wird eine Typanpassung automatisch durchgeführt. Typumwandlungen sind auch in Synchronaktionen möglich, siehe Kapitel "Bewegungssynchronaktionen, Implizite Typwandlung".

2.9 Stringoperationen

Stringoperationen

Neben den klassischen Operationen "Zuweisung" und "Vergleich" sind folgende Stringoperationen möglich:

- Typenkonvertierung nach STRING (AXSTRING) (Seite 80)
- Typenkonvertierung von STRING (NUMBER, ISNUMBER, AXNAME) (Seite 80)
- Verkettung von Strings (<<) (Seite 81)
- Wandlung in Klein-/Großbuchstaben (TOLOWER, TOUPPER) (Seite 83)
- Länge eines Strings bestimmen (STRLEN) (Seite 83)
- Zeichen/String in String suchen (INDEX, RINDEX, MINDEX, MATCH) (Seite 84)
- Auswahl eines Teilstrings (SUBSTR) (Seite 85)
- Lesen und Schreiben von einzelnen Zeichen (Seite 86)
- String formatieren (SPRINT) (Seite 87)

Sonderbedeutung des 0-Zeichens

Das 0-Zeichen wird intern als Enderkennung eines Strings interpretiert. Wird ein Zeichen durch das 0-Zeichen ersetzt, wird der String damit verkürzt.

Beispiel:

Programmcode	Kommentar
DEF STRING[20] STRG="Achse . steht"	
STRG[6]="X"	
MSG(STRG)	; Liefert die Meldung "Achse X steht".
STRG[6]=0	
MSG(STRG)	; Liefert die Meldung "Achse".

2.9.1 Typenkonvertierung nach STRING (AXSTRING)

Durch die Funktion "Typkonvertierung nach STRING" lassen sich Variablen unterschiedlichen Typs als Bestandteil einer Meldung (MSG) nutzen.

Erfolgt bei Verwendung des Operators << implizit für die Datentypen INT, REAL, CHAR und BOOL (siehe "Verkettung von Strings (<<) (Seite 81)").

Ein INT-Wert wird in die normal lesbare Form umgewandelt. Bei REAL-Werten werden bis zu 10 Nachkommastellen angegeben.

Mit dem Befehl AXSTRING können Variable vom Typ AXIS nach STRING gewandelt werden.

Syntax

```
<STRING_ERG> = << <bel._Typ>
<STRING_ERG> = AXSTRING(<Achsbezeichner>)
```

Bedeutung

<STRING_ERG>:	Variable für das Ergebnis der Typkonvertierung	
	Typ:	STRING
<bel._Typ>:	Variablen-Typen INT, REAL, CHAR, STRING und BOOL	
AXSTRING:	Der Befehl AXSTRING liefert den angegebenen Achsbezeichner als String.	
<Achsbezeichner>:	Variable für Achsbezeichner	
	Typ:	AXIS

Hinweis

FRAME-Variablen können nicht konvertiert werden.

2.9.2 Typenkonvertierung von STRING (NUMBER, ISNUMBER, AXNAME)

Mit dem Befehl NUMBER wird von STRING nach REAL konvertiert. Die Konvertierbarkeit kann mit dem Befehl ISNUMBER überprüft werden.

Mit dem Befehl AXNAME wird ein String in den Datentyp AXIS konvertiert.

Syntax

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


Bedeutung

NUMBER:	Der Befehl NUMBER liefert die durch den <String> dargestellte Zahl als REAL-Wert zurück.			
<String>:	Zu konvertierende Variable vom Typ STRING			
<REAL_ERG>:	Variable für das Ergebnis der Typkonvertierung mit NUMBER			
	Typ:	REAL		
ISNUMBER:	Mit dem Befehl ISNUMBER kann überprüft werden, ob der <String> in eine gültige Zahl gewandelt werden kann.			
<BOOL_ERG>:	Variable für das Ergebnis der Abfrage mit ISNUMBER			
	Typ:	BOOL		
	Wert:	TRUE	ISNUMBER liefert den Wert TRUE, wenn der <String> eine nach den Regeln der Sprache gültige REAL-Zahl darstellt.	
		FALSE	Liefert ISNUMBER den Wert FALSE, wird bei Aufruf von NUMBER mit dem gleichen <String> Alarm ausgelöst.	
AXNAME:	Der Befehl AXNAME wandelt den angegebenen <String> in einen Achsbezeichner. Hinweis: Kann der <String> keinem projizierten Achsbezeichner zugeordnet werden, wird ein Alarm ausgelöst.			
<AXIS_ERG>:	Variable für das Ergebnis der Typkonvertierung mit AXNAME			
	Typ:	AXIS		

Beispiel

Programmcode	Kommentar
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 Verkettung von Strings (<<)

Die Funktion "Verkettung von Strings" schafft die Möglichkeit, einen String aus einzelnen Bestandteilen zusammensetzen zu können.

Realisiert wird die Verkettung über den Operator "<<". Dieser Operator hat für alle Kombinationen der Basistypen CHAR, BOOL, INT, REAL und STRING als Zieltyp STRING. Eine eventuell notwendige Konvertierung wird nach den bestehenden Regeln vorgenommen.

Syntax

<bel._Typ> << <bel._Typ>

Bedeutung

<bel._Typ>:	Variable vom Typ CHAR, BOOL, INT, REAL oder STRING
<< :	Operator für die Verkettung von Variablen (<bel._Typ>) zu einer zusammengesetzten Zeichenkette (Typ STRING). Dieser Operator ist auch alleinig als sog. "unäre" Variante verfügbar. So ist es möglich, eine explizite Typwandlung nach STRING auszuführen (nicht für FRAME und AXIS): << <bel._Typ>

Beispielsweise lässt sich so eine Meldung oder ein Kommando aus Textlisten zusammensetzen und Parameter (etwa ein Bausteinname) einfügen:
MSG (STRG_TAB [LOAD_IDX] <<BAUSTEIN_NAME)

Hinweis

Die Zwischenergebnisse bei der Stringverkettung dürfen die maximale Stringlänge nicht überschreiten.

Hinweis

Die Typen FRAME und AXIS können nicht zusammen mit dem Operator "<<" verwendet werden.

Beispiele

Beispiel 1: Verkettung von Strings

Programmcode	Kommentar
DEF INT IDX=2	
DEF REAL VALUE=9.654	
DEF STRING[20] STRG="INDEX:2"	
IF STRG=="Index:"<<IDX GOTOF NO_MSG	
MSG ("Index:"<<IDX<<"/Wert:"<<VALUE)	; Anzeige: "Index:2/Wert:9.654"
NO_MSG:	

Beispiel 2: Explizite Typkonvertierung mit <<

Programmcode	Kommentar
DEF REAL VALUE=3.5	
<<VALUE	; Die angegebene Variable vom Typ REAL wird in den Typ STRING konvertiert.

2.9.4 Wandlung in Klein-/Großbuchstaben (TOLOWER, TOUPPER)

Die Funktion "Wandlung in Klein-/Großbuchstaben" erlaubt es, alle Buchstaben einer Zeichenkette in eine einheitliche Darstellung zu wandeln.

Syntax

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

Bedeutung

TOUPPER:	Mit dem Befehl <code>TOUPPER</code> werden alle Buchstaben einer Zeichenkette in Großbuchstaben umgewandelt.	
TOLOWER:	Mit dem Befehl <code>TOLOWER</code> werden alle Buchstaben einer Zeichenkette in Kleinbuchstaben umgewandelt.	
<String>:	Zeichenkette, die umgewandelt werden soll	
	Typ:	STRING
<STRING_ERG>:	Variable für das Ergebnis der Umwandlung	
	Typ:	STRING

Beispiel

Da es auch möglich ist, Benutzereingaben an der Bedienoberfläche anzustoßen, kann eine einheitliche Darstellung mit Klein- oder Großbuchstaben erreicht werden:

Programmcode

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

2.9.5 Länge eines Strings bestimmen (STRLEN)

Mit dem Befehl `STRLEN` ist es möglich, die Länge einer Zeichenkette zu bestimmen.

Syntax

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

Bedeutung

STRLEN:	Mit dem Befehl <code>STRLEN</code> wird die Länge der angegebenen Zeichenkette bestimmt. Es wird die Anzahl der Zeichen zurückgegeben, die - vom Anfang der Zeichenkette an gezählt - keine 0-Zeichen sind.	
<String>:	Zeichenkette, deren Länge bestimmt werden soll	
	Typ:	STRING

<INT_ERG>:	Variable für das Ergebnis der Bestimmung	
	Typ:	INT

Beispiel

Die Funktion im Zusammenhang mit dem Einzelzeichenzugriff ermöglicht es, das Ende einer Zeichenkette zu bestimmen:

Programmcode

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

2.9.6 Zeichen/String in String suchen (INDEX, RINDEX, MINDEX, MATCH)

Diese Funktionalität erlaubt es, einzelne Zeichen bzw. einen String in einem weiteren String zu suchen. Die Funktionsergebnisse geben an, an welcher Position des Strings das Zeichen/der String im zu untersuchenden String gefunden wurde.

Syntax

INT_ERG=INDEX (STRING, CHAR) ; **Ergebnistyp: INT**

INT_ERG=RINDEX (STRING, CHAR) ; **Ergebnistyp: INT**

INT_ERG=MINDEX (STRING, STRING) ; **Ergebnistyp: INT**

INT_ERG=MATCH (STRING, STRING) ; **Ergebnistyp: INT**

Semantik

Suchfunktionen: Sie liefern die Position im String (erster Parameter) zurück, wo die Suche erfolgreich war. Kann das Zeichen/der String nicht gefunden werden, wird der Wert -1 zurückgegeben. Das erste Zeichen hat dabei die Position 0.

Bedeutung

INDEX:	sucht das als zweiten Parameter angegebene Zeichen (von vorne) im ersten Parameter.
RINDEX:	sucht das als zweiten Parameter angegebene Zeichen (von hinten) im ersten Parameter.
MINDEX:	entspricht der Funktion INDEX, außer, dass eine Liste von Zeichen (als String) übergeben wird, von denen der Index des ersten gefundenen Zeichens zurückgegeben wird.
MATCH:	sucht einen String in einem String.

So lassen sich Strings nach bestimmten Kriterien zerlegen, etwa an Positionen mit Leerzeichen oder Pfadtrennzeichen ("/").

Beispiel

Zerlegen einer Eingabe in Pfad- und Bausteinamen

Programmcode	Kommentar
DEF INT PFADIDX, PROGIDX	
DEF STRING[26] EINGABE	
DEF INT LISTIDX	
EINGABE = "/_N_MPF_DIR/_N_EXECU- TE_MPF"	
LISTIDX = MINDEX (EINGABE, "M,N,O,P") + 1	; Als Wert in LISTIDX wird 3 zurückgeliefert; da "N" das erste Zeichen im Parameter EINGABE, aus der Auswahlliste von vorne, ist.
PFADIDX = INDEX (EINGABE, "/") +1	; damit gilt: PFADIDX = 1
PROGIDX = RINDEX (EINGABE, "/") +1	; damit gilt: PROGIDX = 12
	; mit Hilfe der im nächsten Abschnitt eingeführ- ten Funktion SUBSTR läßt sich die Variable EINGABE in die Komponen- ten ;"Pfad";und "Baustein" zerlegen:
VARIABLE = SUBSTR (EINGABE, PFADIDX, PROGIDX-PFADIDX-1)	; liefert dann "_N_MPF_DIR"
VARIABLE = SUBSTR (EINGABE, PROGIDX)	; liefert dann "_N_EXECUTE_MPF"

2.9.7 Auswahl eines Teilstrings (SUBSTR)

Mit der Funktion SUBSTRING können beliebige Teile innerhalb eines Strings gelesen werden.

Syntax

```
<STRING_ERG>=SUBSTR(<String>,<Index>,<Länge>)
```

```
<STRING_ERG>=SUBSTR(<String>,<Index>)
```

Bedeutung

SUBSTR:	Die Funktion liefert aus <String> einen Teilstring, ausgehend von <Index> mit der angegebenen <Länge>. Ist der Parameter <Länge> nicht angegeben, liefert die Funktion einen Teilstring ausgehend von <Index> bis zum Stringende.
<Index>:	Anfangsposition des Teilstrings innerhalb des Strings. Liegt die Anfangsposition hinter dem Stringende, wird ein Leerstring ("") zurückgegeben. Erstes Zeichen des Strings: Index = 0 Wertebereich: 0 ... (Stringlänge - 1)
<Länge>:	Länge des Teilstrings. Wird eine zu große Länge angegeben, wird nur der Teilstring bis Stringende zurückgegeben. Wertebereich: 1 ... (Stringlänge - 1)

< Wert_1>,< Wert_2>,...,< Wert_n>:	Wert in Form einer Konstanten oder NC-Variablen, der an der Stelle, an der das n-te Formatsteuerzeichen % steht, entsprechend der Formatbeschreibung in den <Format_String> eingefügt wird.
"<Ergebnis_String>":	Formatierte Zeichenkette (maximal 400 Bytes)

Verfügbare Formatbeschreibungen

%B:	<p>Wandlung in den String "TRUE", wenn der zu wandelnde Wert:</p> <ul style="list-style-type: none"> • ungleich 0 ist. • kein Leerstring ist (bei String-Werten). <p>Wandlung in den String "FALSE", wenn der zu wandelnde Wert:</p> <ul style="list-style-type: none"> • gleich 0 ist. • ein Leerstring ist. <p>Beispiel:</p> <pre>N10 DEF BOOL BOOL_VAR=1 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF BOOL_VAR:TRUE" beschrieben.</p>
%C:	<p>Wandlung in ein ASCII-Zeichen.</p> <p>Beispiel:</p> <pre>N10 DEF CHAR CHAR_VAR="X" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C", CHAR_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF CHAR_VAR:X" beschrieben.</p>
%D:	<p>Wandlung in einen String mit einem ganzzahligen Wert (INTEGER).</p> <p>Beispiel:</p> <pre>N10 DEF INT INT_VAR=123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D", INT_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF INT_VAR:123" beschrieben.</p>
%<m>D:	<p>Wandlung in einen String mit einem ganzzahligen Wert (INTEGER). Der String hat eine Mindestlänge von <m> Zeichen. Fehlende Stellen werden linksbündig mit Leerzeichen aufgefüllt.</p> <p>Beispiel:</p> <pre>N10 DEF INT INT_VAR=-123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D", INT_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF INT_VAR:xx-123" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>

%F:	<p>Wandlung in einen String mit einer Dezimalzahl mit 6 Nachkommastellen. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt.</p> <p>Beispiel: N10 DEF REAL REAL_VAR=-1.2341234EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F", REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR: -1234.123400" beschrieben.</p>
%<m>F:	<p>Wandlung in einen String mit einer Dezimalzahl mit 6 Nachkommastellen und einer Gesamtlänge von mindestens <m> Zeichen. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Fehlende Zeichen auf die Gesamtlänge <m> werden linksbündig mit Leerzeichen aufgefüllt.</p> <p>Beispiel: N10 DEF REAL REAL_VAR=-1.23412345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F", REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR: xxx-1234.123457" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>
%.<n>F:	<p>Wandlung in einen String mit einer Dezimalzahl mit <n> Nachkommastellen. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt.</p> <p>Beispiel: N10 DEF REAL REAL_VAR=-1.2345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F", REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:-1234.568" beschrieben.</p>
%<m>.<n>F:	<p>Wandlung in einen String mit einer Dezimalzahl mit <n> Nachkommastellen und einer Gesamtlänge von mindestens <m> Zeichen. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Fehlende Zeichen auf die Gesamtlänge <m> werden linksbündig mit Leerzeichen aufgefüllt.</p> <p>Beispiel: N10 DEF REAL REAL_VAR=-1.2341234567890EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F", REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xx-1234.12" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>
%E:	<p>Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung. Die Mantisse wird normalisiert mit einer Vorkommastelle und 6 Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.</p> <p>Beispiel: N10 DEF REAL REAL_VAR=-1234.567890 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E", REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:-1.234568EX+03" beschrieben.</p>

<p>%<m>E:</p>	<p>Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung mit einer Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt. Die Mantisse wird normalisiert mit einer Vorkommastelle und 6 Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.</p> <p>Beispiel:</p> <pre>N10 DEF REAL REAL_VAR=-1234.5 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen)</p>
<p>%.<n>E:</p>	<p>Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung. Die Mantisse wird normalisiert mit einer Vorkommastelle und <n> Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.</p> <p>Beispiel:</p> <pre>N10 DEF REAL REAL_VAR=-1234.5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:-1.23EX+03" beschrieben.</p>
<p>%<m>.<n>E:</p>	<p>Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung mit einer Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt. Die Mantisse wird normalisiert mit einer Vorkommastelle und <n> Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.</p> <p>Beispiel:</p> <pre>N10 DEF REAL REAL_VAR=-1234.5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E",REAL_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xx-1.23EX+03" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>

%G:	<p>Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung: ist der darzustellende Wert betragsmäßig kleiner als 1.0EX-04 oder größer/gleich 1.0EX+06 wird die Exponentialdarstellung gewählt, sonst die Dezimaldarstellung. Es werden maximal sechs signifikante Stellen angezeigt, ggf. wird gerundet.</p> <p>Beispiel mit Dezimaldarstellung: N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:0.000123457" beschrieben.</p> <p>Beispiel mit Exponentialdarstellung: N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:1.23457EX+06" beschrieben.</p>
%<m>G:	<p>Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung (wie %G). Der String hat eine Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt.</p> <p>Beispiel mit Dezimaldarstellung: N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xxxx0.000123457" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p> <p>Beispiel mit Exponentialdarstellung: N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xxx1.23457EX+06" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>
%.<n>G:	<p>Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung. Es werden maximal <n> signifikante Stellen angezeigt, ggf. wird gerundet. Ist der darzustellende Wert betragsmäßig kleiner als 1.0EX-04 oder größer/gleich 1.0EX(+<n>) wird die Exponentialdarstellung gewählt, sonst die Dezimaldarstellung.</p> <p>Beispiel mit Dezimaldarstellung: N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:0.000123" beschrieben.</p> <p>Beispiel mit Exponentialdarstellung: N10 DEF REAL REAL_VAR=1.234567890123456EX+03 N20 DEF STRING[80] RESULT N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</p> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:1.23EX+03" beschrieben.</p>

<p>%<m>.<n>G:</p>	<p>Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung (wie %.<n>G). Der String hat eine Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt.</p> <p>Beispiel mit Dezimaldarstellung:</p> <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> <p>Ergebnis: Die Stringvariable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xx0.0001235" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p> <p>Beispiel mit Exponentialdarstellung:</p> <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xx1.235EX+06" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>
<p>%.<n>P:</p>	<p>Wandlung eines REAL-Werts in einen INTEGER-Wert unter Berücksichtigung von <n> Nachkommastellen. Der INTEGER-Wert wird als 32-Bit-Binärzahl ausgegeben. Kann der zu wandelnde Wert nicht mit 32 Bit dargestellt werden, wird die Bearbeitung mit Alarm abgebrochen.</p> <p>Da eine mit der Formatanweisung %.<n>P erzeugte Byte-Folge auch binäre Nullen enthalten kann, entspricht der so erzeugte Gesamt-String nicht mehr den Konventionen des NC-Datentyps STRING. Er kann daher weder in einer Variablen vom Typ STRING gespeichert werden, noch mit den String-Befehlen der NC-Sprache weiter bearbeitet werden. Die einzig mögliche Verwendung ist die Parameterübergabe an den WRITE-Befehl mit Ausgabe an ein entsprechendes externes Gerät (siehe folgendes Beispiel).</p> <p>Sobald der <Format_String> eine Formatbeschreibung vom Typ %P enthält, wird der gesamte String, mit Ausnahme der mit %.<n>P generierten Binärzahl, entsprechend dem MD10750 \$MN_SPRINT_FORMAT_P_CODE im Zeichen-Code ASCII, ISO (DIN6024) oder EIA (RS244) ausgegeben. Wird ein nicht umsetzbares Zeichen programmiert, wird die Bearbeitung mit Alarm abgebrochen.</p> <p>Beispiel:</p> <pre>N10 DEF REAL REAL_VAR=123.45 N20 DEF INT ERROR N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1" ... N100 EXTOPEN(ERROR,EXT_DEVICE) N110 IF ERROR <> 0 ... ; Fehlerhandling N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)) N210 IF ERROR <> 0 ... ; Fehlerhandling</pre> <p>Ergebnis: Der String "INTEGER BINARY CODED: 'H0001E23A'" wird an das Ausgabegerät /ext/dev/1 übertragen. Der hexadezimale Wert 0x0001E23A entspricht dem dezimalen Wert 123450.</p>

%<m>.<n>P:	<p>Wandlung eines REAL-Werts entsprechend der Einstellung im Maschinendatum MD10751 \$MN_SPRINT_FORMAT_P_DECIMAL in einen String mit:</p> <ul style="list-style-type: none"> • einer Ganzzahl von <m> + <n> Stellen oder • einer Dezimalzahl mit maximal <m> Vorkommastellen und exakt <n> Nachkommastellen. <p>Wie bei der Formatbeschreibung %.<n>P wird der gesamte String in dem durch MD10750 \$MN_SPRINT_FORMAT_P_CODE festgelegten Zeichen-Code abgelegt.</p> <p>Wandlung bei MD10751 = 0:</p> <p>Der REAL-Wert wird in einen String mit einer Ganzzahl von <m> + <n> Stellen gewandelt. Ggf. werden Nachkommastellen auf <n> Stellen gerundet oder mit 0 aufgefüllt. Fehlende Vorkommastellen werden mit Leerzeichen aufgefüllt. Das Minus-Vorzeichen wird linksbündig angefügt, an Stelle des Plus-Vorzeichens wird ein Leerzeichen abgesetzt.</p> <p>Beispiel:</p> <pre>N10 DEF REAL REAL_VAR=-123.45 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)</pre> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "PUNCHED TAPE FORMAT:-xx123450" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p> <p>Wandlung bei MD10751 = 1:</p> <p>Der REAL-Wert wird in einen String mit einer Dezimalzahl mit maximal <m> Vorkommastellen und exakt <n> Nachkommastellen gewandelt. Ggf. werden die Vorkommastellen abgeschnitten und die Nachkommastellen gerundet oder mit 0 aufgefüllt. Ist <n> gleich 0 entfällt auch der Dezimalpunkt.</p> <p>Beispiel:</p> <pre>N10 DEF REAL REAL_VAR1=-123.45 N20 DEF REAL REAL_VAR2=123.45 N30 DEF STRING[80] RESULT N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)</pre> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "PUNCHED TAPE FORMAT:-123.450 VAR2:23" beschrieben.</p>
%S:	<p>Einfügen eines Strings.</p> <p>Beispiel:</p> <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)</pre> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:ABCDEFGH" beschrieben.</p>
%<m>S:	<p>Einfügen eines Strings mit mindestens <m> Zeichen. Fehlende Stellen werden mit Leerzeichen aufgefüllt.</p> <p>Beispiel:</p> <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)</pre> <p>Ergebnis: Die String-Variablen RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:xxxABCDEFGH" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>

%.<n>S:	<p>Einfügen von <n> Zeichen eines Strings (beginnend mit dem ersten Zeichen).</p> <p>Beispiel:</p> <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:ABC" beschrieben.</p>
%<m>.<n>S:	<p>Einfügen von <n> Zeichen eines Strings (beginnend mit dem ersten Zeichen). Die Gesamtlänge des erzeugten Strings hat mindestens <m> Zeichen. Fehlende Stellen werden mit Leerzeichen aufgefüllt.</p> <p>Beispiel:</p> <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:xxxxxABCDE" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).</p>
%X:	<p>Wandlung eines INTEGER-Werts in einen String mit Hexadezimal-Darstellung.</p> <p>Beispiel:</p> <pre>N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR)</pre> <p>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "INTEGER HEXADECIMAL:A5B8" beschrieben.</p>

Hinweis

Die Eigenschaft der NC-Sprache, bezüglich Bezeichnern und Schlüsselworten nicht zwischen Groß- und Kleinbuchstaben zu unterscheiden, gilt auch für die Formatbeschreibungen. Sie können daher ohne funktionalen Unterschied mit Klein- oder Großbuchstaben programmiert werden.

Kombinationsmöglichkeiten

Die folgende Tabelle gibt Auskunft darüber, welche NC-Datentypen mit welcher Formatbeschreibung kombiniert werden können. Es gelten die Regeln zur impliziten Datentypwandlung (siehe "Datentypen (Seite 56)").

	NC-Datentypen						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-

	NC-Datentypen						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

Hinweis

Die Tabelle zeigt, dass die NC-Datentypen AXIS und FRAME nicht direkt in der SPRINT-Funktion verwendet werden können. Es ist aber möglich:

- den Datentyp AXIS mit der Funktion AXSTRING in einen String zu wandeln, der dann mit SPRINT weiterverarbeitet werden kann.
- die einzelnen Werte des Datentyps FRAME per Frame-Komponentenzugriff zu lesen. Dadurch bekommt man ein Datum vom Typ REAL, das mit SPRINT weiterverarbeitet werden kann.

2.10 Programmsprünge und -verzweigungen

2.10.1 Rücksprung auf Programmanfang (GOTOS)

Mit dem Befehl `GOTOS` ist es möglich, zur Programmwiederholung an den Anfang eines Haupt- oder Unterprogramms zurückzuspringen.

Über Maschinendaten kann eingestellt werden, dass bei jedem Rücksprung auf den Programmanfang:

- die Programmlaufzeit auf "0" gesetzt wird.
- die Werkstückzählung um den Wert "1" erhöht wird.

Syntax

`GOTOS`

Bedeutung

GOTOS:	Sprunganweisung mit Sprungziel Programmanfang.	
	Die Ausführung wird gesteuert über das NC/PLC-Nahtstellensignal: DB21, ... DBX384.0 (Programmverzweigung steuern)	
	Wert:	Bedeutung:
	0	Kein Rücksprung auf den Programmanfang. Die Programmbearbeitung wird mit dem nächsten Teileprogrammsatz nach <code>GOTOS</code> fortgeführt.
1	Rücksprung auf den Programmanfang. Das Teileprogramm wird wiederholt.	

Randbedingungen

- GOTOS löst intern ein STOPRE (Vorlaufstopp) aus.
- Bei einem Teileprogramm mit Datendefinitionen (LUD-Variablen) wird mit GOTOS auf den ersten Programmsatz nach dem Definitionsabschnitt gesprungen, d. h. die Datendefinitionen werden nicht erneut ausgeführt. Die definierten Variablen behalten daher den im GOTOS-Satz erreichten Wert und werden nicht auf die im Definitionsabschnitt programmierten Standardwerte zurückgesetzt.
- In Synchronaktionen und Technologie-Zyklen steht der Befehl GOTOS nicht zur Verfügung.

Beispiel

Programmcode	Kommentar
N10 ...	; Programmanfang.
...	
N90 GOTOS	; Sprung an den Programmanfang.
...	

2.10.2 Programmsprünge auf Sprungmarken (GOTOB, GOTOF, GOTO, GOTOC)

In einem Programm können Sprungmarken (Labels) gesetzt werden, auf die von anderen Stellen innerhalb desselben Programms mit dem Befehlen GOTOF, GOTOB, GOTO bzw. GOTOC gesprungen werden kann. Die Programmbearbeitung wird dann mit der Anweisung fortgesetzt, die unmittelbar nach der Sprungmarke folgt. Dadurch sind Verzweigungen innerhalb des Programms realisierbar.

Neben den Sprungmarken sind als Sprungziele auch Haupt- und Nebensatznummern möglich.

Wenn vor der Sprunganweisung eine Sprungbedingung (IF ...) formuliert ist, dann erfolgt der Programmsprung nur dann, wenn die Sprungbedingung erfüllt ist.

Syntax

```
GOTOB <Sprungziel>
IF <Sprungbedingung> == TRUE GOTOB <Sprungziel>
```

```
GOTOF <Sprungziel>
IF <Sprungbedingung> == TRUE GOTOF <Sprungziel>
```

```
GOTO <Sprungziel>
IF <Sprungbedingung> == TRUE GOTO <Sprungziel>
```

```
GOTOC <Sprungziel>
IF <Sprungbedingung> == TRUE GOTOC <Sprungziel>
```


Bedeutung

GOTOB:	Sprunganweisung mit Sprungziel in Richtung Programmanfang.	
GOTOF:	Sprunganweisung mit Sprungziel in Richtung Programmende.	
GOTO:	Sprunganweisung mit Sprungzielsuche. Die Suche erfolgt erst in Richtung Programmende, dann in Richtung Programmanfang.	
GOTOC:	Wirkung wie GOTO mit dem Unterschied, dass der Alarm 14080 "Sprungziel nicht gefunden" unterdrückt wird. Das bedeutet, dass die Programmbearbeitung im Falle einer ergebnislosen Sprungzielsuche nicht abgebrochen wird, sondern mit der auf den Befehl GOTOC folgenden Programmzeile fortgesetzt wird.	
<Sprungziel>:	Sprungzielparameter Mögliche Angaben sind:	
	<Sprungmarke>:	Sprungziel ist die im Programm gesetzte Sprungmarke mit benutzerdefiniertem Namen: <Sprungmarke>:
	<Satznummer>:	Sprungziel ist eine Haupt- oder Nebensatznummer (z. B.: 200, N300)
	Variable vom Typ STRING:	Variables Sprungziel. Die Variable steht für eine Sprungmarke oder eine Satznummer.
IF:	Schlüsselwort zur Formulierung der Sprungbedingung. Die Sprungbedingung lässt alle Vergleichs- und logischen Operationen zu (Ergebnis: TRUE oder FALSE). Der Programmsprung wird ausgeführt, wenn das Ergebnis dieser Operation TRUE ist.	

Hinweis

Sprungmarken (Labels)

Sprungmarken stehen immer am Anfang eines Satzes. Wenn eine Programmnummer vorhanden ist, steht die Sprungmarke unmittelbar nach der Satznummer.

Für die Benennung von Sprungmarken gelten folgende Regeln:

- Anzahl an Zeichen:
 - mindestens 2
 - höchstens 32
- Erlaubte Zeichen sind:
 - Buchstaben
 - Ziffern
 - Unterstriche
- Die ersten beiden Zeichen müssen Buchstaben oder Unterstriche sein.
- Nach dem Namen der Sprungmarke folgt ein Doppelpunkt (":").

Randbedingungen

- Sprungziel kann nur ein Satz mit Sprungmarke oder Satznummer sein, der **innerhalb** des Programms liegt.
- Eine Sprunganweisung ohne Sprungbedingung muss in einem separaten Satz programmiert werden. Bei Sprunganweisungen mit Sprungbedingungen gilt diese Einschränkung nicht. Hier können mehrere Sprunganweisungen in einem Satz formuliert werden.
- Bei Programmen mit Sprunganweisungen ohne Sprungbedingungen muss das Programmende M2/M30 nicht zwangsläufig am Programmende stehen.

Beispiele

Beispiel 1: Sprünge auf Sprungmarken

Programmcode	Kommentar
N10 ...	
N20 GOTOF Label_1	; Sprung in Richtung Programmende zur Sprungmarke "Label_1".
N30 ...	
N40 Label_0: R1=R2+R3	; Sprungmarke "Label_0" gesetzt.
N50 ...	
N60 Label_1:	; Sprungmarke "Label_1" gesetzt.
N70 ...	
N80 GOTOB Label_0	; Sprung in Richtung Programmanfang zur Sprungmarke "Label_0".
N90 ...	

Beispiel 2: Indirekter Sprung auf Satznummer

Programmcode	Kommentar
IF <Bedingung> == TRUE	
R10=100	; Sprungziel zuweisen
ELSE	
R10=110	; Sprungziel zuweisen
ENDIF	
	; Sprung in Richtung Programmende auf den Satz, dessen Satznummer in R10 steht
N10 GOTOF "N"<<R10	
...	
N90 ...	
N100 ...	; Sprungziel
N110 ...	
...	

Beispiel 3: Sprung auf variables Sprungziel

Programmcode	Kommentar
DEF STRING[20] ZIEL	
IF <Bedingung> == TRUE	
ZIEL = "Marke1"	; Sprungziel zuweisen
ELSE	
ZIEL = "Marke2"	; Sprungziel zuweisen
ENDIF	
	; Sprung in Richtung Programmende zum variablen Sprungziel "Inhalt von ZIEL"
GOTOF ZIEL	
Marke1: T="Bohrer1"	; Sprungziel 1
...	
Marke2: T="Bohrer2"	; Sprungziel 2
...	

Beispiel 4: Sprung mit Sprungbedingung

Programmcode	Kommentar
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; Zuweisung der Anfangswerte
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; Sprungmarke LA1
N42 R1=R1+R3 R4=R4-1	
	; IF Sprungbedingung == TRUE
	; THEN Sprung in Richtung Programmanfang zur Sprungmarke LA1
N43 IF R4>0 GOTOB LA1	
N44 M30	; Programmende

2.10.3 Programmverzweigung (CASE ... OF ... DEFAULT ...)

Die CASE-Funktion bietet die Möglichkeit, den aktuellen Wert (Typ: INT) einer Variablen oder einer Rechenfunktion zu überprüfen und abhängig vom Ergebnis an unterschiedliche Stellen im Programm zu springen.

Syntax

```
CASE(<Ausdruck>) OF <Konstante_1> GOTOF <Sprungziel_1> <Konstante_2>
GOTOF <Sprungziel_2> ... DEFAULT GOTOF <Sprungziel_n>
```

Bedeutung

CASE:	Sprunganweisung	
<Ausdruck>:	Variable oder Rechenfunktion	
OF:	Schlüsselwort zur Formulierung der bedingten Programmverzweigungen	
<Konstante_1>:	Erster angegebener konstanter Wert für die Variable oder Rechenfunktion	
	Typ:	INT

<Konstante_2>:	Zweiter angegebener konstanter Wert für die Variable oder Rechenfunktion	
	Typ:	INT
DEFAULT:	Für die Fälle, in denen die Variable oder Rechenfunktion keinen der angegebenen konstanten Werte annimmt, kann mit der Anweisung DEFAULT ein Sprungziel bestimmt werden. Hinweis: Falls die DEFAULT-Anweisung nicht programmiert ist, wird in diesen Fällen der auf die CASE-Anweisung folgende Satz zum Sprungziel.	
GOTOF:	Sprunganweisung mit Sprungziel in Richtung Programmende. Statt GOTOF sind auch alle anderen GOTO-Befehle programmierbar (siehe Thema "Programmsprünge auf Sprungmarken").	
<Sprungziel_1>:	Auf dieses Sprungziel wird verzweigt, wenn der Wert der Variablen oder Rechenfunktion der ersten angegebenen Konstanten entspricht. Das Sprungziel kann wie folgt angegeben werden:	
	<Sprungmarke>:	Sprungziel ist die im Programm gesetzte Sprungmarke mit benutzerdefiniertem Namen: <Sprungmarke>:
	<Satznummer>:	Sprungziel ist eine Haupt- oder Nebensatznummer (z. B.: 200, N300)
	Variable vom Typ STRING:	Variables Sprungziel. Die Variable steht für eine Sprungmarke oder eine Satznummer.
<Sprungziel_2>:	Auf dieses Sprungziel wird verzweigt, wenn der Wert der Variablen oder Rechenfunktion der zweiten angegebenen Konstanten entspricht.	
<Sprungziel_n>:	Auf dieses Sprungziel wird verzweigt, wenn der Wert der Variablen keinen der angegebenen konstanten Werte annimmt.	

Beispiel

```

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

Die CASE-Anweisung aus N30 definiert folgende Programmverzeigungsmöglichkeiten:

1. Wenn der Wert der Rechenfunktion $VAR1+VAR2-VAR3 = 7$, dann springe zu dem Satz mit Sprungmarkendefinition "Label_1" (→ N40).
2. Wenn der Wert der Rechenfunktion $VAR1+VAR2-VAR3 = 9$, dann springe zu dem Satz mit Sprungmarkendefinition "Label_2" (→ N50).
3. Wenn der Wert der Rechenfunktion $VAR1+VAR2-VAR3$ weder 7 noch 9 beträgt, dann springe zu dem Satz mit Sprungmarkendefinition "Label_3" (→ N60).

2.11 Programmteiwiederholung (REPEAT, REPEATB, ENDLABEL, P)

Die Programmteiwiederholung ermöglicht die Wiederholung bereits geschriebener Programmteile innerhalb eines Programms in beliebiger Zusammensetzung.

Die zu wiederholenden Programmzeilen bzw. Programmbereiche werden durch Sprungmarken (Labels) gekennzeichnet.

Hinweis

Sprungmarken (Labels)

Sprungmarken stehen immer am Anfang eines Satzes. Wenn eine Programmnummer vorhanden ist, steht die Sprungmarke unmittelbar nach der Satznummer.

Für die Benennung von Sprungmarken gelten folgende Regeln:

- Anzahl an Zeichen:
 - mindestens 2
 - höchstens 32
 - Erlaubte Zeichen sind:
 - Buchstaben
 - Ziffern
 - Unterstriche
 - Die ersten beiden Zeichen müssen Buchstaben oder Unterstriche sein.
 - Nach dem Namen der Sprungmarke folgt ein Doppelpunkt (":").
-

Syntax

1. Einzelne Programmzeile wiederholen:

```
<Sprungmarke>: ...
...
REPEATB <Sprungmarke> P=<n>
...
```

2. Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholen:

```
<Sprungmarke>: ...
...
REPEAT <Sprungmarke> P=<n>
...
```

3. Bereich zwischen zwei Sprungmarken wiederholen:

```
<Start-Sprungmarke>: ...
...
<End-Sprungmarke>: ...
...
REPEAT <Start-Sprungmarke> <End-Sprungmarke> P=<n>
```

| ...

Hinweis

Die REPEAT-Anweisung mit den beiden Sprungmarken zu klammern, ist nicht möglich. Wird die <Start-Sprungmarke> vor der REPEAT-Anweisung gefunden und wird die <End-Sprungmarke> nicht vor der REPEAT-Anweisung erreicht, dann wird die Wiederholung zwischen <Start-Sprungmarke> und REPEAT-Anweisung durchgeführt.

4. Bereich zwischen Sprungmarke und ENDLABEL wiederholen:

```
<Sprungmarke>: ...
...
ENDLABEL: ...
...
REPEAT <Sprungmarke> P=<n>
...
```

Hinweis

Die REPEAT-Anweisung mit der <Sprungmarke> und dem ENDLABEL zu klammern, ist nicht möglich. Wird die <Sprungmarke> vor der REPEAT-Anweisung gefunden und wird ENDLABEL nicht vor der REPEAT-Anweisung erreicht, dann wird die Wiederholung zwischen <Sprungmarke> und REPEAT-Anweisung durchgeführt.

Bedeutung

REPEATB:	Befehl zum Wiederholen einer Programmzeile
REPEAT:	Befehl zum Wiederholen eines Programmbereichs
<Sprungmarke>:	<p>Die <Sprungmarke> kennzeichnet:</p> <ul style="list-style-type: none"> • die zu wiederholende Programmzeile (bei REPEATB) bzw. • den Beginn des zu wiederholenden Programmbereichs (bei REPEAT) <p>Die mit der <Sprungmarke> gekennzeichnete Programmzeile kann vor oder nach der REPEAT/REPEATB-Anweisung stehen. Gesucht wird zunächst in Richtung Programmanfang. Wird die Sprungmarke in dieser Richtung nicht gefunden, dann wird in Richtung Programmende gesucht.</p> <p>Ausnahme: Wenn der Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholt werden soll (siehe 2. unter Syntax), dann muss die mit der <Sprungmarke> gekennzeichnete Programmzeile vor der REPEAT-Anweisung stehen, da in diesem Fall nur in Richtung Programmanfang gesucht wird. Enthält die Zeile mit der <Sprungmarke> weitere Anweisungen, so werden diese bei jeder Wiederholung erneut ausgeführt.</p>

ENDLABEL:	Schlüsselwort, welches das Ende eines zu wiederholenden Programmbe- reichs markiert Enthält die Zeile mit dem ENDLABEL weitere Anweisungen, so werden diese bei jeder Wiederholung erneut ausgeführt. ENDLABEL kann mehrfach im Programm verwendet werden.
P:	Adresse zur Angabe der Wiederholungsanzahl
<n>:	Anzahl an Programmteiwiederholungen
	Typ: INT
	Der zu wiederholende Programmteil wird <n> mal wiederholt. Nach der letz- ten Wiederholung wird das Programm mit der auf die REPEAT-/REPEATB-Zei- le folgenden Zeile fortgesetzt. Hinweis: Ist kein P=<n> angegeben, wird der zu wiederholende Programmteil genau einmal wiederholt.

Beispiele

Beispiel 1: Einzelne Programmzeile wiederholen

Programmcode	Kommentar
N10 POSITION1: X10 Y20	
N20 POSITION2: CYCLE(0,,9,8)	; Positionszyklus
N30 ...	
N40 REPEATB POSITION1 P=5	; Führe SATZ N10 fünfmal aus.
N50 REPEATB POSITION2	; Führe Satz N20 einmal aus.
N60 ...	
N70 M30	

Beispiel 2: Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholen

Programmcode	Kommentar
N5 R10=15	
N10 Begin: R10=R10+1	; Breite
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	; Führe Bereich N10 bis N70 viermal aus.
N90 Z10	
N100 M30	

Beispiel 3: Bereich zwischen zwei Sprungmarken wiederholen

Programmcode	Kommentar
N5 R10=15	

2.11 Programmteiwiederholung (REPEAT, REPEATB, ENDLABEL, P)

Programmcode	Kommentar
N10 Begin: R10=R10+1	; Breite
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	; Führe Bereich N10 bis N70 dreimal aus.
N110 Z10	
N120 M30	

Beispiel 4: Bereich zwischen Sprungmarke und ENDLABEL wiederholen

Programmcode	Kommentar
N10 G1 F300 Z-10	
N20 BEGIN1:	
N30 X10	
N40 Y10	
N50 BEGIN2:	
N60 X20	
N70 Y30	
N80 ENDLABEL: Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3: X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	; Führe Bereich N110 bis N120 dreimal aus.
N140 REPEAT BEGIN2 P=2	; Führe Bereich N50 bis N80 zweimal aus.
N150 M100	
N160 REPEAT BEGIN1 P=2	; Führe Bereich N20 bis N80 zweimal aus.
N170 Z10	
N180 X0 Y0	
N190 M30	

Beispiel 5: Fräsbearbeitung, Bohrposition mit verschiedenen Technologien bearbeiten

Programmcode	Kommentar
N10 ZENTRIERBOHRER()	; Zentrierbohrer einwechseln.
N20 POS_1:	; Bohrpositionen 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	

Programmcode	Kommentar
N70 ENDLABEL:	
N80 POS_2:	; Bohrpositionen 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 BOHRER()	; Bohrer wechseln und Bohrzyklus.
N140 GEWINDE(6)	; Gewindebohrer M6 einwechseln und Gewindezyklus.
N150 REPEAT POS_1	; Wiederhole Programmabschnitt ab POS_1 einmal bis ENDLABEL.
N160 BOHRER()	; Bohrer wechseln und Bohrzyklus.
N170 GEWINDE(8)	; Gewindebohrer M8 einwechseln und Gewindezyklus.
N180 REPEAT POS_2	; Wiederhole Programmabschnitt ab POS_2 einmal bis ENDLABEL.
N190 M30	

Weitere Informationen

- Programmteiwiederholung kann geschachtelt aufgerufen werden. Jeder Aufruf belegt eine Unterprogrammebene.
- Ist während der Bearbeitung einer Programmteiwiederholung `M17` oder `RET` programmiert, so wird die Programmteiwiederholung abgebrochen. Das Programm wird mit dem auf die `REPEAT`-Zeile folgenden Satz fortgesetzt.
- In der aktuellen Programm-Anzeige wird die Programmteiwiederholung als eigene Unterprogrammebene angezeigt.
- Wird während der Programmteil-Bearbeitung Ebenenabbruch ausgelöst, so wird das Programm nach dem Aufruf der Programmteibearbeitung fortgesetzt.
Beispiel:

Programmcode	Kommentar
N5 R10=15	
N10 BEGIN: R10=R10+1	; Breite
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	; Ebenenabbruch
N50 X=-R10	
N60 Y=-R10	
N70 END: Z10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	; Programmbearbeitung fortsetzen.
N130 M30	

- Kontrollstrukturen und Programmteilwiederholung können kombiniert genutzt werden. Es sollte jedoch keine Überschneidungen geben. Eine Programmteilwiederholung sollte innerhalb eines Kontrollstruktur-Zweigs liegen bzw. eine Kontrollstruktur innerhalb einer Programmteilwiederholung.
- Bei der Mischung von Sprüngen und Programmteilwiederholung werden die Sätze rein sequentiell abgearbeitet. Erfolgt z. B. ein Sprung aus einer Programmteilwiederholung, so wird solange bearbeitet, bis das programmierte Programmteilende gefunden wird.
Beispiel:

Programmcode

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

Hinweis

Die REPEAT-Anweisung sollte hinter den Verfahrssätzen stehen.

2.12 Kontrollstrukturen

Die Steuerung arbeitet die NC-Sätze standardmäßig in der programmierten Reihenfolge ab.

Diese Reihenfolge kann durch die Programmierung von alternativen Programmblöcken und Programmschleifen variiert werden. Die Programmierung dieser Kontrollstrukturen erfolgt mit den Schlüsselwörtern IF, ELSE, ENDIF, LOOP, FOR, WHILE und REPEAT.

ACHTUNG

Programmierfehler

Kontrollstrukturen sind nur innerhalb des Anweisungsteils eines Programms möglich. Definitionen im Programmkopf können nicht bedingt oder wiederholt ausgeführt werden.

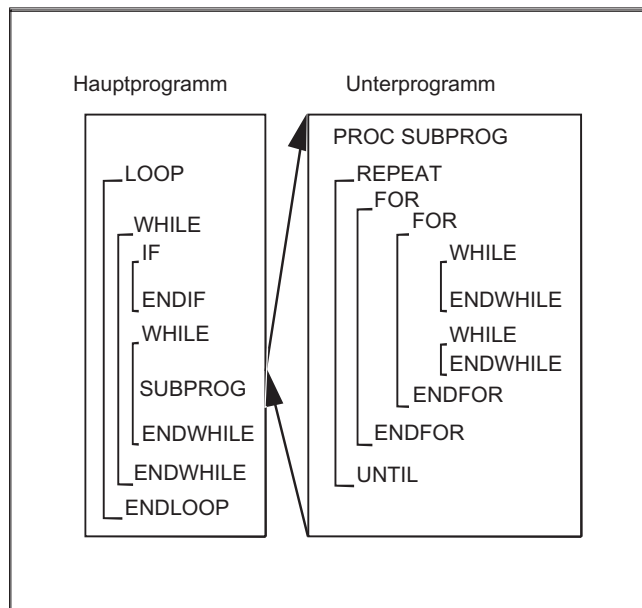
Schlüsselwörter für Kontrollstrukturen dürfen ebenso wie Sprungziele nicht mit Makros überlagert werden. Eine Abprüfung bei der Makrodefinition findet nicht statt.

Wirksamkeit

Eine Kontrollstruktur kann nicht programmübergreifend verwendet werden.

Schachtelungstiefe

Innerhalb jeder Unterprogrammebene ist eine Schachtelungstiefe von bis zu 16 Kontrollstrukturen möglich.



Laufzeitverhalten

Im standardmäßig aktiven Interpreterbetrieb kann durch Verwendung von Programmsprüngen ein schnellerer Programmablauf als mit Kontrollstrukturen erreicht werden.

In vorkompilierten Zyklen ist kein Unterschied zwischen Programmsprüngen und Kontrollstrukturen vorhanden.

Aktuelle Satzanzeige bei Programmschleifen

Werden innerhalb einer Programmschleife nur Vorlaufsätze ausgeführt, wird in der aktuellen Satzanzeige der letzte Hauptaufsatz **vor** der Programmschleife angezeigt.

Damit z.B. zu Diagnosezwecken auch die abgearbeiteten Vorlaufsätze in der aktuellen Satzanzeige sichtbar werden, muss der Decodier-Einzelsatz SBL2 aktiviert werden.

Literatur

Funktionshandbuch Grundfunktionen, Kapitel: BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1) > Einzelsatz > Decodier-Einzelsatz SBL2 mit impliziten Vorlaufstop

Randbedingungen

- Sätze mit Kontrollstrukturelementen können nicht ausgeblendet werden.
- Sprungmarken (Labels) sind in Sätzen mit Kontrollstrukturelementen nicht erlaubt.
- Kontrollstrukturen werden interpretativ abgearbeitet. Bei Erkennen eines Schleifenendes wird unter Berücksichtigung der dabei gefundenen Kontrollstrukturen nach dem Schleifenanfang gesucht. Daher wird im Interpreterbetrieb die Blockstruktur eines Programms nicht komplett geprüft.
- Grundsätzlich empfiehlt sich, Kontrollstrukturen und Programmsprünge nicht gemischt zu verwenden.
- Bei Vorverarbeitung von Zyklen kann die korrekte Schachtelung von Kontrollstrukturen überprüft werden.

2.12.1 Bedingte Anweisung und Verzweigung (IF, ELSE, ENDIF)

Bedingten Anweisung: IF - Programmblock - ENDIF

Bei einer bedingten Anweisung wird der zwischen `IF` und `ENDIF` stehende Programmblock nur ausgeführt, wenn die Bedingung erfüllt ist.

Verzweigung: IF - Programmblock_1 - ELSE - Programmblock_2 - ENDIF

Bei einer Verzweigung wird immer einer von zwei Programmblöcken ausgeführt.

Ist die Bedingung erfüllt, wird der zwischen `IF` und `ELSE` stehende Programmblock_1 ausgeführt.

Ist die Bedingung **nicht** erfüllt, wird der zwischen `ELSE` und `ENDIF` stehende Programmblock_2 ausgeführt.

Syntax

Bedingte Anweisung

```
IF <Bedingung>  
    Programmblock  
ENDIF  
; Ausführung bei: <Bedingung> == TRUE
```

Verzweigung

```

IF <Bedingung>
    Programmblock_1                ; Ausführung bei: <Bedingung> == TRUE
ELSE
    Programmblock_2                ; Ausführung bei: <Bedingung> == FALSE
ENDIF

```

Bedeutung

IF:	Leitet die bedingte Anweisung bzw. Verzweigung ein.
ELSE:	Leitet den alternativen Programmblock ein.
ENDIF:	Markiert das Ende der bedingte Anweisung bzw. Verzweigung.
<Bedingung>:	Logischer Ausdruck, dessen Auswertung TRUE oder FALSE ergibt.

Beispiel: Werkzeugwechselunterprogramm

Programmcode	Kommentar
PROC L6	Werkzeugwechselroutine
N500 DEF INT TNR_AKTUELL	Variable für aktive T-Nummer
N510 DEF INT TNR_VORWAHL	Variable für vorgewählte T-Nummer
	Aktuelles Werkzeug ermitteln
N520 STOPRE	
N530 IF \$P_ISTEST	Im Programmtest-Betrieb wird ...
N540 TNR_AKTUELL = \$P_TOOLNO	... aus dem Programmkontext das "aktuelle" Werkzeug gelesen.
N550 ELSE	Andernfalls wird ...
N560 TNR_AKTUELL = \$TC_MPP6[9998,1]	... das Werkzeug der Spindel ausgelesen.
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	T-Nummer des vorgewählten Werkzeugs auf der Spindel lesen.
N590 IF TNR_AKTUELL <> TNR_VORWAHL	Wenn das vorgewählte Werkzeug noch nicht das aktuelle Werkzeug ist, dann ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... Werkzeugwechsellpunkt anfahren ...
N610 M206	... und Werkzeugwechsel ausführen.
N620 ENDIF	
N630 M17	

2.12.2 Endlos-Programmschleife (LOOP, ENDLOOP)

Die Endlos-Schleife findet Verwendung in Endlos-Programmen. Am Schleifenende findet immer wieder der Rücksprung zum Schleifenanfang statt.

Syntax

```
LOOP  
...  
ENDLOOP
```

Bedeutung

LOOP:	Leitet die Endlosschleife ein.
ENDLOOP:	Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang.

Beispiel

Programmcode

```
...  
LOOP  
MSG("keine Werkzeugschneide aktiv")  
M0  
STOPRE  
ENDLOOP  
...
```

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

Die Zählschleife wird verwendet, wenn ein Arbeitsablauf mit einer festen Anzahl von Durchläufen wiederholt werden soll.

Syntax

```
FOR <Variable> = <Anfangswert> TO <Endwert>  
...  
ENDFOR
```

Bedeutung

FOR:	Leitet die Zählschleife ein.
ENDFOR:	Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang, solange der Endwert der Zählung noch nicht erreicht ist.

<Variable>:	Zählvariable, die vom Anfangs- bis zum Endwert hochgezählt wird und sich bei jedem Durchlauf um den Wert "1" erhöht.	
	Typ	INT oder REAL Hinweis: Der Typ REAL wird genommen, wenn z. B. R-Parameter für eine Zählschleife programmiert werden. Ist die Zählvariable vom Typ REAL, wird ihr Wert auf einen ganzzahligen Wert gerundet.
<Anfangswert>:	Anfangswert der Zählung Bedingung: Der Anfangswert muss kleiner sein als der Endwert.	
<Endwert>:	Endwert der Zählung	

Beispiele

Beispiel 1: INTEGER-Variable oder R-Parameter als Zählvariable

INTEGER-Variable als Zählvariable:

Programmcode	Kommentar
DEF INT iVARIABLE1 R10=R12-R20*R1 R11=6 FOR iVARIABLE1= R10 TO R11 R20=R21*R22+R33 ENDFOR M30	; Zählvariable = INTEGER-Variable

R-Parameter als Zählvariable:

Programmcode	Kommentar
R11=6 FOR R10=R12-R20*R1 TO R11 R20=R21*R22+R33 ENDFOR M30	; Zählvariable = R-Parameter (Realvariable)

Beispiel 2: Fertigung einer festen Teilstückzahl

Programmcode	Kommentar
DEF INT STUECKZAHL FOR STUECKZAHL = 0 TO 100 G01 ... ENDFOR M30	; Definiert Variable vom Typ INT mit Namen "STUECKZAHL". ; Leitet die Zählschleife ein. Die Variable "STUECKZAHL" wird vom Anfangswert "0" bis zum Endwert "100" hochgezählt. ; Ende der Zählschleife.

2.12.4 Programmschleife mit Bedingung am Schleifenanfang (WHILE, ENDWHILE)

Bei einer WHILE-Schleife steht die Bedingung am Schleifenanfang. Solange die Bedingung erfüllt ist, wird die WHILE-Schleife durchlaufen.

Syntax

```
WHILE <Bedingung>  
...  
ENDWHILE
```

Bedeutung

WHILE:	Leitet die Programmschleife ein.
ENDWHILE:	Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang.
<Bedingung>:	Bedingung, die erfüllt sein muss, damit die WHILE-Schleife durchlaufen wird.

Beispiel

Programmcode	Kommentar
...	
WHILE \$AA_IW[BOHRACHSE] > -10	; Aufruf der WHILE-Schleife unter folgender Bedingung: der aktuelle WKS-Sollwert für die Bohrachse muss größer -10 sein.
G1 G91 F250 AX[BOHRACHSE] = -1	
ENDWHILE	
...	

2.12.5 Programmschleife mit Bedingung am Schleifenende (REPEAT, UNTIL)

Bei einer REPEAT-Schleife steht die Bedingung am Schleifenende. Die REPEAT-Schleife wird einmal durchlaufen und solange wiederholt, bis die Bedingung erfüllt ist.

Syntax

```
REPEAT  
...  
UNTIL <Bedingung>
```


Bedeutung

REPEAT:	Leitet die Programmschleife ein.
UNTIL:	Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang.
<Bedingung>:	Bedingung, die erfüllt sein muss, damit die REPEAT-Schleife nicht mehr durchlaufen wird.

Beispiel

Programmcode	Kommentar
...	
REPEAT	; Aufruf der REPEAT-Schleife.
...	
UNTIL ...	; Prüfung, ob Bedingung erfüllt ist.
...	

2.12.6 Programmbeispiel mit verschachtelten Kontrollstrukturen

Programmcode	Kommentar
LOOP	
IF NOT \$P_SEARCH	; IF kein Satzsuchlauf
G1 G90 X0 Z10 F1000	
WHILE \$AA_IM[X] <= 100	; WHILE (Sollwert X-Achse <= 100)
G1 G91 X10 F500	; Bohrbild
Z-5 F100	
Z5	
ENDWHILE	
ELSE	; ELSE Satzsuchlauf
MSG("Im Suchlauf wird nicht gebohrt")	
ENDIF	; ENDIF
\$A_OUT[1] = 1	; nächste Bohrplatte
G4 F2	
ENDLOOP	
M30	

2.13 Programmkoordinierung (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Ein Kanal der NC kann prinzipiell das in ihm gestartete Programm unabhängig von anderen Kanälen seiner Betriebsartengruppe (BAG) abarbeiten. Sind aber gleichzeitig mehrere Programme in mehreren Kanälen der BAG an der Fertigung eines Werkstücks beteiligt, müssen die Programmabläufe in den unterschiedlichen Kanälen koordiniert werden.

Hinweis

Betriebsartengruppe

Eine Programmkoordinierung ist nur zwischen Kanälen möglich, die zur **gleichen Betriebsartengruppe (BAG)** gehören.

Syntax

```
INIT (<Kanal-Nr.>, <Programm>, <Quittungsmodus>)  
START (<Kanal-Nr.>, <Kanal-Nr.>, ...)  
WAITM (<Marken-Nr.>, <Kanal-Nr.>, <Kanal-Nr.>, ...)  
WAITMC (<Marken-Nr.>, <Kanal-Nr.>, <Kanal-Nr.>, ...)  
WAITE (<Kanal-Nr.>, <Kanal-Nr.>, ...)  
SETM (<Marken-Nr.>, <Marken-Nr.>, ...)  
CLEARM (<Marken-Nr.>, <Marken-Nr.>, ...)
```

Hinweis

Eine Befehl der Programmkoordinierung muss alleine in einem eigenen Satz stehen.

Bedeutung


INIT:	Vordefinierte Prozedur zur Anwahl des NC-Programms, das im angegebenen Kanal abgearbeitet werden soll		
	<Kanal-Nr.>:	Kanalnummer	
		Typ:	INT
	<Programm>:	Optionale Pfadangabe (absolut oder relativ) + Programmname	
		Typ:	STRING
	Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)".		
<Quittungsmodus>:	Parameter vom Typ CHAR (optional)		
	Werte:	"N"	ohne Quittung Die Programmbearbeitung wird nach Absenden des Kommandos fortgeführt. Der Absender wird nicht benachrichtigt, wenn das Kommando nicht erfolgreich ausgeführt werden kann.
		"S"	synchrone Quittung Die Programmabarbeitung wird solange angehalten, bis die Empfängerkomponente das Kommando quittiert hat. Bei positiver Quittung wird der nächste Befehl abgearbeitet. Bei negativer Quittung wird eine Fehlermeldung ausgegeben.
	Hinweis: Standardeinstellung: "S" (synchrone Quittung)		
START:	Vordefinierte Prozedur zum Starten des im jeweiligen Kanal angewählten Programms		
	<Kanal-Nr.>, ...:	Aufzählung der Kanalnummern	
		Typ:	INT
WAITM:	Vordefinierte Prozedur zum Warten auf das Erreichen einer Marke in den angegebenen Kanälen		
	Im eigenen Kanal wird die angegebene Marke durch WAITM gesetzt. Der vorhergehenden Satz wird mit Genauhalt beendet. Die Marke wird nach Synchronisation gelöscht.		
	Gleichzeitig können max. 10 Marken pro Kanal gesetzt werden.		
	<Marken-Nr.>:	Nummer der Marke	
		Typ:	INT
<Kanal-Nr.>, ...:	Kanalnummer, ... (die Nummer des eigenen Kanals muss nicht angegeben werden)		
		Typ:	INT
WAITE:	Vordefinierte Prozedur zum Warten auf das Programmende in einem oder mehreren anderen Kanälen		
	<Kanal-Nr.>, ...:	Aufzählung der Kanalnummern	
		Typ:	INT

WAITMC:	Vordefinierte Prozedur zum Warten auf das Erreichen einer Marke in den angegebenen Kanälen Im Ggs. zu WAITM wird Genauhalt nur eingeleitet, wenn die anderen Kanäle die Marke noch nicht erreicht haben. Parameter wie bei WAITM.	
SETM:	Vordefinierte Prozedur zum Setzen einer oder mehrerer Marken für die Kanalkoordinierung Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst. SETM behält Gültigkeit über Reset und NC-Start hinweg.	
	<Marken-Nr.>, ...:	Aufzählung der Markennummern
CLEARM:	Vordefinierte Prozedur zum Löschen einer oder mehrerer Marken der Kanalkoordinierung Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst. Alle Marken im Kanal können mit CLEARM() gelöscht werden. CLEARM(0) löscht die Marke "0". CLEARM behält Gültigkeit über Reset und NC-Start hinweg.	
	<Marken-Nr.>, ...:	Aufzählung der Markennummern

Hinweis

Kanalnummer


Kanalnamen müssen über Variablen in Nummern gewandelt werden.

 VORSICHT
Kanalnummer
Die Nummernzuordnung ist vor leichtfertiger Änderung zu sichern.

Hinweis

Kanalname

Anstelle von Kanalnummern können auch die über \$MC_CHAN_NAME definierten Kanalnamen (Bezeichner oder Schlüsselwort) programmiert werden (Typ: STRING).

 VORSICHT
Kanalname
Die Namen dürfen nicht bereits in der NC in anderer Bedeutung wie z. B. als Schlüsselwort, Sprachbefehl, Achsname etc. vorhanden sein.

Hinweis

Zum Datenaustausch zwischen den Programmen können die Variablen benutzt werden, über die Kanäle gemeinsam verfügen (NCK-spezifische globale Variablen). Ansonsten wird die Programmerstellung für jeden Kanal getrennt vorgenommen.

Beispiele

START über Maschinendatum parametrisierte Kanalnamen

Parametrierung:

```
$MC_CHAN_NAME[ 0 ] = "CHAN_1" ; Name von Kanal 1
$MC_CHAN_NAME[ 1 ] = "CHAN_2" ; Name von Kanal 2
```

Die im Maschinendatums MD20000 \$MC_CHAN_NAME parametrisierten Namen "CHAN_1" und "CHAN_2" repräsentieren Steuerungs-intern die Kanalnummer 1 und 2.

Programmcode	Kommentar
START(CHAN_1)	; Start von Kanal 1
START(CHAN_2)	; Start von Kanal 2

START über lokale "Kanalnamen" und Anwendervariable

Festlegung:

- Kanal 1: Programmlokaler Name "MASCHINE"
- Kanal 2: Programmlokaler Name "LADER"

Programmcode	Kommentar
DEF INT MASCHINE = 1	; Definition Anwendervariable für Kanal 1
DEF INT LADER = 2	; Definition Anwendervariable für Kanal 2
...	
START(MASCHINE)	; Start von Kanal 1
START(LADER)	; Start von Kanal 2

START über lokale "Kanalnamen", Anwendervariable und parametrisierte Kanalnamen

Programmcode	Kommentar
DEF INT chanNo1	; Definition Anwendervariable für Kanal 1
DEF INT chanNo2	; Definition Anwendervariable für Kanal 2
chanNo1 = CHAN_1	; Zuweisung parametrisierter Kanalnamen Kanal 1
chanNo2 = CHAN_2	; Zuweisung parametrisierter Kanalnamen Kanal 2
...	
START(chanNo1)	; Start von Kanal 1
START(chanNo2)	; Start von Kanal 2

INIT-Befehl mit absoluter Pfadangabe

Anwahl von Programm /_N_MPF_DIR/_N_ABSPAN1_MPF in Kanal 2.

Programmcode

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

INIT-Befehl mit Programmnamen

Anwahl des Programms mit dem Namen "MYPROG". Die Steuerung sucht das Programm anhand des Suchpfades.

```

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

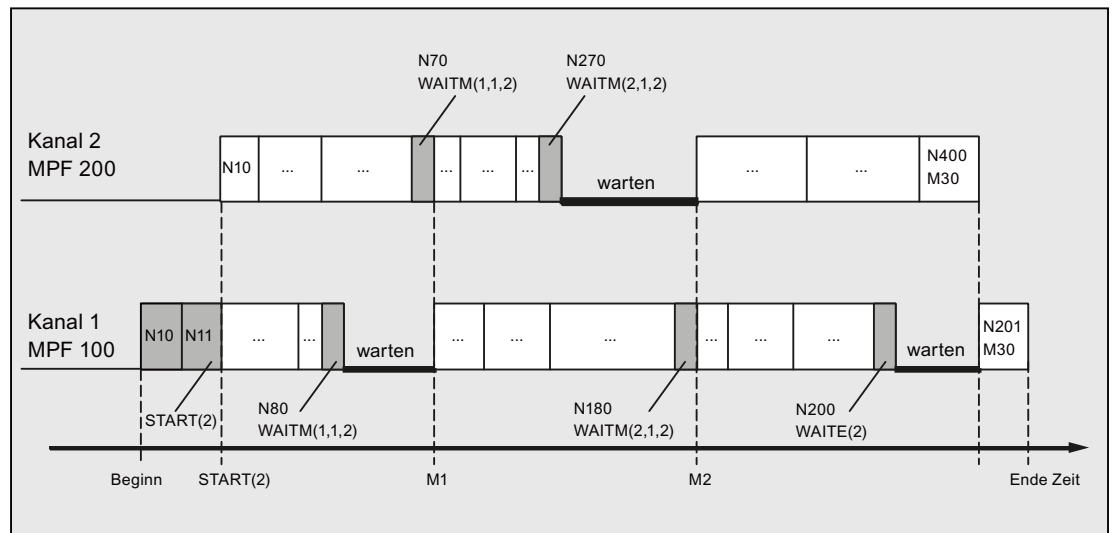
Programmkoordinierung mit WAITM

Kanal 1: Das Programm /_N_MPF_DIR/_N_MPF100_MPF ist bereits angewählt. und gestartet.

Programmcode	Kommentar
	; Programm MPF100
N10 INIT(2,"MPF200","N")	; Anwahl Programm MPF200, Kanal 2
N11 START(2)	; Start von Kanal 2
...	
N80 WAITM(1,1,2)	; Warten auf WAIT-Marke 1 in Kanal 1 und 2
N81 ...	; Kanal 1, N81 und Kanal 2, N71 werden ; synchron begonnen
...	
N180 WAITM(2,1,2)	; Warten auf WAIT-Marke 2 in Kanal 1 und 2
N181 ...	; Kanal 1, N181 und Kanal 2, N271 werden ; synchron begonnen
...	
N200 WAITE(2)	; Warten auf Programmende in Kanal 2
N201 ...	; N201 wird erst nach dem Programmende von ; MPF200 in Kanal 2 begonnen
N201 M30	; Programmende Kanal 1

Kanal 2: In Kanal 1 wird über Satz N10 und N20 das Programm MPF200_MPF für Kanal 2 angewählt und gestartet.

Programmcode	Kommentar
;\$PATH=/_N_MPF_DIR	; Programm MPF200
...	
N70 WAITM(1,1,2)	Warten auf WAIT-Marke 1 in Kanal 1 und 2
N71 ...	; Kanal 1, N81 und Kanal 2, N71 werden ; synchron begonnen
...	
N270 WAITM(2,1,2)	Warten auf WAIT-Marke 2 in Kanal 1 und 2
N271 ...	; Kanal 1, N181 und Kanal 2, N271 werden ; synchron begonnen
...	
N400 M30	Programmende Kanal 2



Randbedingungen

Nicht synchroner Beginn des Abarbeitens von Folgesätzen nach WAIT-Marken

Bei einer Kanalkoordinierung mittels WAIT-Marken kann es zu einem nicht synchronen Beginn des Abarbeitens der Folgesätze kommen. Dieses Verhalten tritt auf, wenn unmittelbar vor Erreichen der gemeinsamen WAIT-Marke in einem der zu synchronisierenden Kanäle eine Aktion ausgelöst wird, die in diesem Restweglöschen mit implizitem Repositionieren (REPOSA) zur Folge hat.

Annahme: Aktuelle Achszuordnung in den Kanälen 1 und 2

- Kanal 1: Achsen X1 und U
- Kanal 2: Achse X2

Tabelle 2-2 Zeitlicher Ablauf in den Kanälen 1 und 2

Kanal 1	Kanal 2	Beschreibung
...	...	Beliebige Bearbeitung in Kanal 1 und 2
N100 WAITM(20, 1, 2)		Kanal 1: erreicht die WAIT-Marke und wartet auf Synchronisation mit Kanal 2
<i>Beginn der GETD(U) Bearbeitung:</i>	N200 GETD(U)	Kanal 2: fordert die Achse U aus Kanal 1 an Kanal 1: Bearbeitung von GET(U) im Hintergrund
• Achstausch	N210 WAITM(20, 1, 2)	Kanal 2: erreicht die WAIT-Marke. ⇒ Die Synchronisation der Kanäle 1 und 2 ist damit abgeschlossen
• Restweglöschen	N220 G0 X2=100	Kanal 2: Beginn des Abarbeitens von N220
• REPOSA		
<i>Ende</i>		
N110 G0 X1=100		Kanal 1: Zeitversetzter Beginn des Abarbeitens von N110

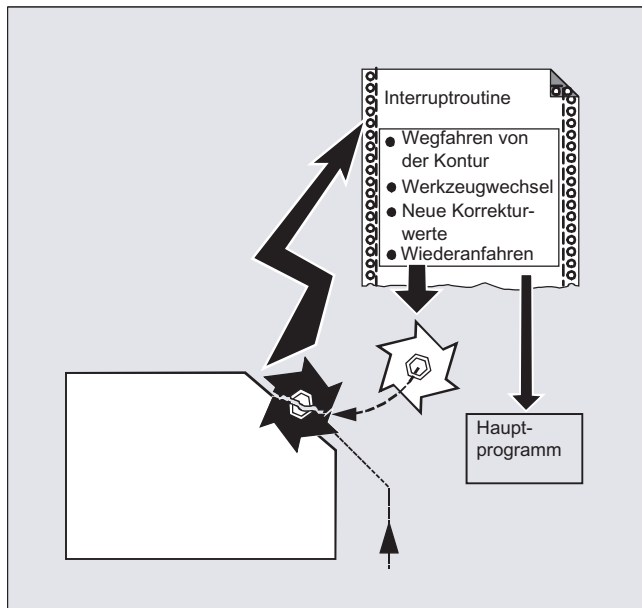
2.14 Interruptroutine (ASUP)

2.14.1 Funktion einer Interruptroutine

Hinweis

Die in der folgenden Beschreibung abwechselnd vorkommenden Begriffe "Asynchrones Unterprogramm (ASUP)" und "Interruptroutine" kennzeichnen die gleiche Funktionalität.

Die Funktion einer Interruptroutine soll anhand eines typischen Beispiels verdeutlicht werden:



Während der Bearbeitung bricht das Werkzeug. Hierdurch wird ein Signal ausgelöst, das den laufenden Bearbeitungsablauf stoppt und gleichzeitig ein Unterprogramm – die sogenannte Interruptroutine – startet. In diesem Unterprogramm stehen alle Anweisungen, die in diesem Fall ausgeführt werden sollen.

Ist das Unterprogramm abgearbeitet (und hierdurch die Betriebsbereitschaft hergestellt), springt die Steuerung in das Hauptprogramm zurück und setzt die Bearbeitung – je nach `REPOS`-Befehl – an der Unterbrechungsstelle fort (siehe "Wiederauffahren an Kontur (Seite 458)").

VORSICHT

Kollisionsgefahr

Wenn im Unterprogramm kein `REPOS`-Befehl programmiert ist, dann wird auf den Endpunkt des Satzes positioniert, der auf den unterbrochenen Satz folgt.

Literatur

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1), Kapitel: "Asynchrone Unterprogramme (ASUPs), Interruptroutinen"

2.14.2 Interruptroutine erstellen**Interruptroutine als Unterprogramm erstellen**

Die Interruptroutine wird bei der Definition wie ein Unterprogramm gekennzeichnet.

Beispiel:

Programmcode	Kommentar
PROC ABHEB_Z	; Programmname "ABHEB_Z"
N10 ...	; Danach folgen die NC-Sätze.
...	
N50 M17	; Zum Schluss Programmende und Rückkehr ins Hauptprogramm.

Modale G-Funktionen sichern (SAVE)

Die Interruptroutine kann bei der Definition mit `SAVE` gekennzeichnet werden.

Das Attribut `SAVE` bewirkt, dass die vor dem Aufruf der Interruptroutine aktiven modalen G-Funktionen gesichert und nach dem Ende der Interruptroutine wieder reaktiviert werden (siehe "Unterprogramme mit SAVE-Mechanismus (SAVE) (Seite 159)").

Dadurch ist es möglich, die Bearbeitung nach Ablauf der Interruptroutine an der Unterbrechungsstelle fortzusetzen.

Beispiel:

Programmcode
PROC ABHEB_Z SAVE
N10 ...
...
N50 M17

Weitere Interruptroutinen zuordnen (SETINT)

Innerhalb der Interruptroutine können `SETINT`-Anweisungen (siehe "Interruptroutine zuordnen und starten (SETINT)" (Seite 122)) programmiert und hierdurch weitere Interruptroutinen scharf geschaltet werden. Das Auslösen erfolgt erst durch den Eingang.

Literatur

Für weitere Informationen zur Erstellung von Unterprogrammen siehe Kapitel "Unterprogrammtechnik, Makrotechnik".

2.14.3 Interruptroutine zuordnen und starten (SETINT, PRIO, BLSYNC)

Die Steuerung verfügt über mehrere schnelle Eingänge (Eingang 1 ... 8), die jeweils einen Interrupt (1 ... 8) auslösen. Jedem Interrupt kann über den Befehl `SETINT` eine Priorität und eine Interrupt-Routine zugeordnet werden. Wird der Interrupt durch Setzen des schnellen Eingangs ausgelöst, wird die aktuelle Bearbeitung im Kanal unterbrochen und die Interrupt-Routine gestartet.

Interrupt-Priorität

Werden in einem Teileprogramm mehreren Eingängen Interrupts zugeordnet, müssen den Interrupts unterschiedliche Prioritäten zugewiesen werden.

Einem Interrupt kann ein Prioritätswerte von 1 ... 128 zugeordnet werden. Prioritätswert 1 entspricht der höchsten, 128 der niedrigsten Priorität.

Syntax

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

Bedeutung

SETINT (<n>):	Dem Interruptsignal <n> wird NC-Programm (ASUP) <Name> zuordnen. Die zugeordnete Interruptroutine wird gestartet, sobald Interruptsignal <n> == 1 erkannt wird. Hinweis: Wird einem Interruptsignal <n> eine andere Interrupt-Routine zugeordnet, wird die vorhergehende Zuordnung unwirksam.	
<n>:	Nummer des Interruptsignals	
	Typ:	INT
	Wertebereich:	1 ... 32
PRIO=:	Priorität des Interrupts (optional)	
<Wert>:	(optional) Prioritätswert	
	Typ:	INT
	Wertebereich:	1 ... 128 (1 ⇒ höchste Priorität)
<NAME>:	Name des NC-Programms (ASUP)	
BLSYNC:	(optional) BLSYNC bewirkt, dass nach dem Auslösen des Interrupts zuerst gewartet wird, bis der aktuelle Satz abgearbeitet ist. Erst danach wird die Interrupt-Routine ausgeführt.	
LIFTFAST:	(optional) LIFTFAST bewirkt, dass nach dem Auslösen des Interrupts zuerst ein Schnellabheben erfolgt (siehe Kapitel "Schnellabheben von der Kontur (SETINT LIFTFAST, ALF) (Seite 125)"). Erst danach wird die Interrupt-Routine ausgeführt.	

Randbedingungen

Interrupt-Regeln

1. Für jeden Interrupt, der nicht sofort abgearbeitet werden kann, oder aktuell schon in Bearbeitung ist, wird eine weitere Interrupt-Anforderung gespeichert. Darüber hinaus gehende Interrupt-Anforderungen für diesen Interrupt verloren.
2. Wird aktuell ein Interrupt bearbeitet und es wird ein weiterer Interrupt mit höherer Priorität ausgelöst, unterbricht dieser den niederprioren Interrupt. Nach Abschluss des höherprioren Interrupts, wird der niederpriore Interrupt fortgesetzt. Treffen während der höherpriore Interrupt bearbeitet wird, weiter Anforderungen für den niederprioren Interrupt ein, wird eine Anforderung gespeichert. Weitere gehen verloren.
3. Wird aktuell ein Interrupt bearbeitet und es wird ein weiterer Interrupt mit höherer Priorität ausgelöst, unterbricht dieser den niederprioren Interrupt. Der höherpriore Interrupt wird bearbeitet. Wird wiederum ein höherpriore Interrupt ausgelöst, wird der aktuelle Interrupt unterbrochen und der höherpriore Interrupt bearbeitet. Maximal sind sechs aktive Interrupt-Ebenen möglich. Eine aktuell bearbeitete Interrupt-Ebene und fünf wartende Interrupt-Ebenen. Für jede aktive Interrupt-Ebene wird maximal eine weitere Interrupt-Anforderung gespeichert. Alle weiteren Interrupt-Anforderungen gehen verloren. Ebenso gehen Interrupt-Anforderung verloren, wenn diese für weitere Interrupt-Ebenen (Interrupt-Ebene ≥ 7) angefordert werden.

Beispiele

Beispiel 1: Interruptroutinen zuordnen und Priorität festlegen

Programmcode	Kommentar
N20 SETINT(3) PRIO=1 ABHEB_Z	; IF Eingang 3 == 1
	; THEN Interruptroutine "ABHEB_Z" starten
N30 SETINT(2) PRIO=2 ABHEB_X	; IF Eingang 2 == 1
	; THEN Interruptroutine "ABHEB_X" starten.

Die Interruptroutinen werden in der Reihenfolge der Prioritätswerte nacheinander abgearbeitet, wenn die Eingänge gleichzeitig anstehen: zuerst "ABHEB_Z", dann "ABHEB_X".

Beispiel 2: Interruptroutine neu zuordnen

Programmcode	Kommentar
N20 SETINT(3) PRIO=2 ABHEB_Z	; IF Eingang 3 == 1
	; THEN Interruptroutine "ABHEB_Z" starten
...	
N80 SETINT(3) PRIO=1 ABHEB_X	; IF Eingang 3 == 1
	; THEN Interruptroutine "ABHEB_X" starten

2.14.4 Zuordnung einer Interruptroutine deaktivieren/reaktivieren (DISABLE, ENABLE)

Eine SETINT-Anweisung kann mit DISABLE deaktiviert und mit ENABLE wieder aktiviert werden, ohne dass die Zuordnung Eingang → Interruptroutine verloren geht.

Syntax

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

Bedeutung

DISABLE (<n>):	Befehl: Deaktivieren der Interruptroutinen-Zuordnung von Eingang <n>	
ENABLE (<n>):	Befehl: Reaktivieren der Interruptroutinen-Zuordnung von Eingang <n>	
<n>:	Parameter: Nummer des Interruptsignals	
	Typ:	INT
	Wertebereich:	1 ... 32

Beispiel

Programmcode	Kommentar
N20 SETINT(3) PRIO=1 ABHEB_Z	; Wenn Eingang 3 schaltet, dann soll die ; Interruptroutine "ABHEB_Z" starten.
...	
N90 DISABLE(3)	; Die SETINT-Anweisung aus N20 wird deaktiviert.
...	
N130 ENABLE(3)	; Die SETINT-Anweisung aus N20 wird wieder akti- viert.
...	

2.14.5 Zuordnung einer Interruptroutine löschen (CLRINT)

Eine mit SETINT definierte Zuordnung eines Interruptsignals zu einem NC-Programm (ASUP) kann mit CLRINT gelöscht werden.

Syntax

CLRINT (<n>)

Bedeutung

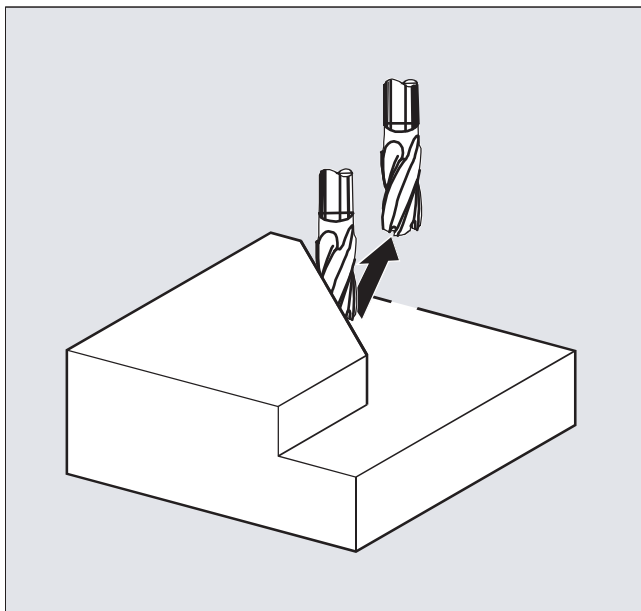
CLRINT (<n>):	Befehl: Löschen der Zuordnung des Interruptsignals <n> zum mit SETINT definierten NC-Programm (ASUP) <n>	
<n>:	Parameter: Nummer des Interruptsignals	
	Typ:	INT
	Wertebereich:	1 ... 32

Beispiel

Programmcode	Kommentar
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; Die Zuordnung zwischen Eingang "3" und der Interruptroutine "ABHEB_Z" ist gelöscht.

2.14.6 Schnellabheben von der Kontur (SETINT LIFTFAST, ALF)

Bei einer SETINT-Anweisung mit LIFTFAST wird beim Schalten des Eingangs das Werkzeug durch schnelles Abheben von der Werkstückkontur weggefahren.



Der weitere Ablauf ist davon abhängig, ob die SETINT-Anweisung neben LIFTFAST eine Interruptroutine enthält:

Mit Interruptroutine:	Nach dem Schnellabheben wird die Interruptroutine ausgeführt.
Ohne Interruptroutine:	Die Bearbeitung wird nach dem Schnellabheben mit Alarm gestoppt.

Syntax

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

Bedeutung

SETINT (<n>):	Befehl: Eingang <n> einer Interruptroutine zuordnen. Die zugeordnete Interruptroutine startet, wenn Eingang <n> schaltet.	
<n>:	Parameter: Nummer des Eingangs	
	Typ:	INT
	Wertebereich:	1 ... 8
PRIO= :	Festlegung der Priorität	
<Wert>:	Prioritätswert	
	Wertebereich:	1 ... 128
	Priorität 1 entspricht der höchsten Priorität.	
<NAME>:	Name des Unterprogramms (Interruptroutine), das abgearbeitet werden soll.	
LIFTFAST:	Befehl: Schnellabheben von der Kontur	
ALF=... :	Befehl: Programmierbare Verfahrrichtung (steht im Bewegungssatz) Zu den Programmiermöglichkeiten mit ALF siehe Thema "Verfahrrichtung beim Schnellabheben von der Kontur (Seite 127)".	

Randbedingungen

Verhalten bei aktivem Frame mit Spiegelung

Bei der Bestimmung der Abheberichtung wird geprüft, ob ein Frame mit Spiegelung aktiv ist. In diesem Fall werden bei der Abheberichtung bezogen auf die Tangentenrichtung rechts und links vertauscht. Die Richtungsanteile in Werkzeugrichtung werden nicht gespiegelt. Aktiviert wird dieses Verhalten durch die MD-Einstellung:

```
MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE
```

Beispiel

Ein abgebrochenes Werkzeug soll automatisch durch ein Schwesterwerkzeug ersetzt werden. Die Bearbeitung wird dann mit dem neuen Werkzeug fortgesetzt.

Hauptprogramm:

Hauptprogramm	Kommentar
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; Wenn Eingang 1 schaltet, wird sofort das Werkzeug mit Schnellabheben (Code Nr. 7 für Werkzeugradiuskorrektur G41) von der Kontur weg gefahren. Dann wird die Interruptroutine "W_WECHS" abgearbeitet.
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	

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

Unterprogramm:

Unterprogramm	Kommentar
PROC W_WECHS SAVE	; Unterprogramm mit Speicherung des aktuellen Betriebszustandes
N10 G0 Z100 M5	; Werkzeugwechselposition, Spindelstopp
N20 T11 M6 D1 G41	; Werkzeug wechseln
N30 REPOS L RMBBL M3	; Kontur wiederanfahren und Rücksprung ins Hauptprogramm (wird in einem Satz programmiert)

2.14.7 Verfahrrichtung beim Schnellabheben von der Kontur

Rückzugsbewegung

Die Ebene der Rückzugsbewegung wird durch folgende G-Codes bestimmt:

- **LFTXT**
Die Ebene der Rückzugsbewegung wird aus der Bahntangente und der Werkzeugrichtung bestimmt (Standardeinstellung).
- **LFWP**
Die Ebene der Rückzugsbewegung ist die aktive Arbeitsebene, die mit den G-Codes G17, G18 oder G19 ausgewählt wird. Die Richtung der Rückzugsbewegung ist unabhängig von der Bahntangente. Damit ist ein achsparalleles Schnellabheben programmierbar.
- **LFPOS**
Rückzug der mit POLFMASK / POLFMLIN bekannt gemachten Achse auf die mit POLF programmierte absolute Achsposition.
ALF hat keinen Einfluss auf die Abheberichtung für mehrere Achsen sowie für mehrere Achsen im linearen Zusammenhang.
Literatur:
Programmierhandbuch Grundlagen; Kapitel: "Schnellrückzug während Gewindeschneiden"

Programmierbare Verfahrrichtung (ALF=...)

In der Ebene der Rückzugsbewegung wird mit ALF die Richtung in diskreten Schritten von 45 Grad programmiert.

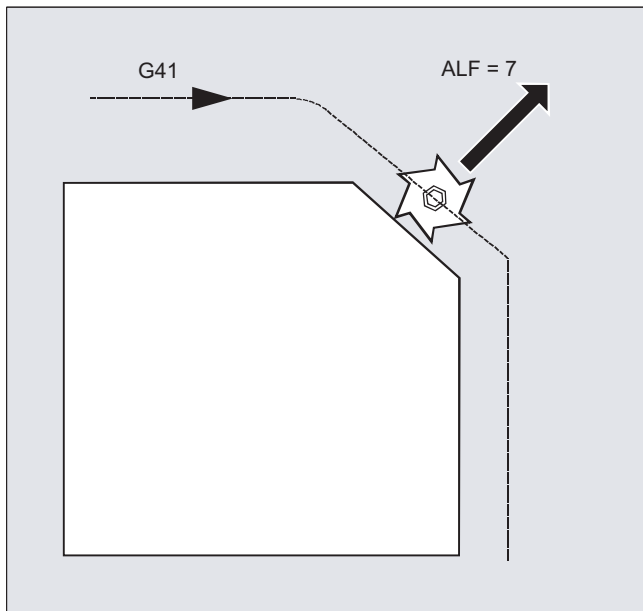
Die möglichen Verfahrrichtungen sind in der Steuerung unter speziellen Code-Nummern gespeichert und unter dieser Nummer abrufbar.

Beispiel:

Programmcode

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

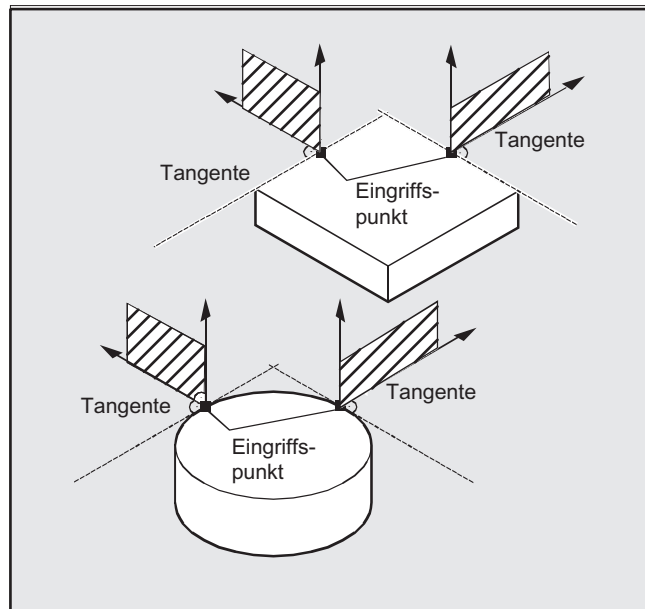
Das Werkzeug fährt bei eingeschaltetem G41 (Bearbeitungsrichtung links von der Kontur) senkrecht von der Kontur weg.



Bezugsebene für die Beschreibung der Verfahrrichtungen bei LFTXT

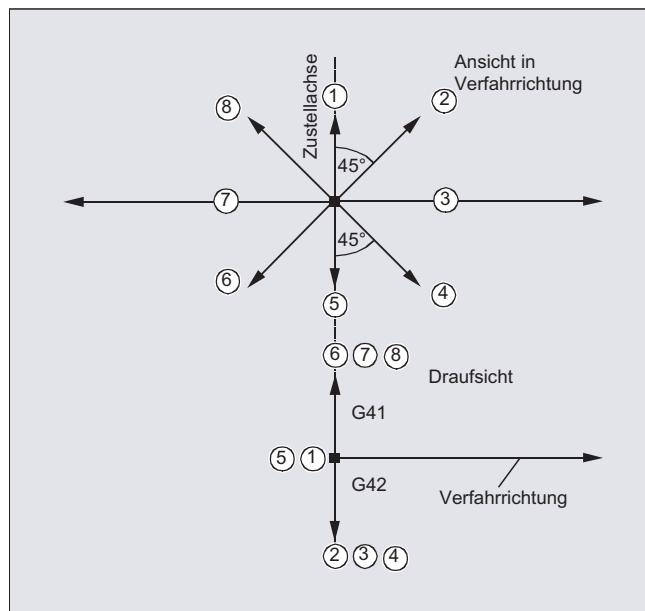
Im Eingriffspunkt des Werkzeugs an der programmierten Kontur wird eine Ebene aufgespannt, die als Bezug für die Angabe der Abhebebewegung mit der entsprechenden Code-Nummer dient.

Die Bezugsebene wird aufgespannt aus der Werkzeuglängsachse (Zustellrichtung) und einem Vektor, der zu dieser und senkrecht zur Tangente im Eingriffspunkt des Werkzeugs an der Kontur steht.




Code-Nummern mit Verfahrrichtungen bei LFTXT

Ausgehend von der Bezugsebene finden Sie in folgender Abbildung die Code-Nummern mit Verfahrrichtungen.



Für $ALF=1$ ist der Rückzug in Werkzeugrichtung festgelegt.

Mit `ALF=0` ist die Funktion "Schnellabheben" ausgeschaltet.

 VORSICHT
Kollisionsgefahr
Bei eingeschalteter Werkzeugradiuskorrektur sollten:
<ul style="list-style-type: none">• bei G41 die Codierungen 2, 3, 4• bei G42 die Codierungen 6, 7, 8
nicht verwendet werden, da in diesen Fällen das Werkzeug zur Kontur hinfahren und mit dem Werkstück kollidieren würde.

Code-Nummern mit Verfahrrichtungen bei LFWP

Bei `LFWP` ergibt sich die Richtung in der Arbeitsebene nach folgender Zuordnung:

- G17: X/Y-Ebene
ALF=1: Rückzug in X-Richtung
ALF=3: Rückzug in Y-Richtung
- G18: Z/X-Ebene
ALF=1: Rückzug in Z-Richtung
ALF=3: Rückzug in X-Richtung
- G19: Y/Z-Ebene
ALF=1: Rückzug in Y-Richtung
ALF=3: Rückzug in Z-Richtung

2.14.8 Bewegungsablauf bei Interruptroutinen

Interruptroutine ohne LIFTFAST

Die Achsbewegungen werden auf der Bahn bis zum Stillstand abgebremst. Anschließend startet die Interruptroutine.

Die Stillstandsposition wird als Unterbrechungsposition abgespeichert und wird bei `REPOS` mit `RMIBL` am Ende der Interruptroutine angefahren.

Interruptroutine mit LIFTFAST

Die Achsbewegungen werden auf der Bahn abgebremst. Gleichzeitig wird die `LIFTFAST`-Bewegung als überlagerte Bewegung ausgeführt. Wenn die Bahnbewegung und `LIFTFAST`-Bewegung zum Stillstand gekommen sind, wird die Interruptroutine gestartet.

Als Unterbrechungsposition wird die Position auf der Kontur abgespeichert, bei der die `LIFTFAST`-Bewegung gestartet und dadurch die Bahn verlassen wurde.

Die Interruptroutine verhält sich mit LIFTFAST und ALF=0 identisch wie die Interruptroutine ohne LIFTFAST.

Hinweis

Der Betrag, um den die Geometrieachsen beim Schnellabheben von der Kontur wegfahren, ist über ein Maschinendatum einstellbar.

2.15 Achstausch, Spindeltausch (RELEASE, GET, GETD)

Eine oder mehrere Achsen bzw. Spindeln können immer nur in einem Kanal interpoliert werden. Muss eine Achse wechselweise in zwei verschiedenen Kanälen arbeiten (z. B. Palettenwechsler), so muss sie zunächst im aktuellen Kanal freigegeben und dann in den anderen Kanal übernommen werden. Die Achse wird zwischen den Kanälen getauscht.

Achstauscherweiterungen

Eine Achse/Spindel kann mit Vorlaufstopp und Synchronisation zwischen Vorlauf und Hauptlauf oder alternativ auch ohne Vorlaufstopp getauscht werden. Außerdem ist ein Achstausch auch möglich über

- Achscontainer-Drehung AXCTSWE bzw. AXCTWED mittels impliziten GET/GETD.
- Frame mit Rotation, wenn diese Achse hierüber mit anderen Achsen verknüpft ist.
- Synchronaktionen, siehe Bewegungssynchronaktionen, "Achstausch RELEASE, GET".

Maschinenhersteller

Bitte beachten Sie die Angaben des Maschinenherstellers. Über projektierbare Maschinendaten muss eine Achse für den Achstausch in allen Kanälen eindeutig definiert sein und das Achstauschverhalten ist auch über Maschinendaten veränderbar einstellbar.

Syntax

RELEASE (Achsname, Achsname, ...) oder RELEASE (S1)

GET (Achsname, Achsname, ...) oder GET (S2)

GETD (Achsname, Achsname, ...) oder GETD (S3)

Mit GETD (GET Directly) wird eine Achse direkt aus einem anderen Kanal geholt. Das bedeutet, dass zu diesem GETD kein passendes RELEASE in einem anderen Kanal programmiert sein muss. Es bedeutet aber auch, dass jetzt eine andere Kanalkommunikation aufgebaut werden muss (z. B. Waitmarken).


Bedeutung

RELEASE (Achsname, Achsname, ...):	Freigeben der Achse(n)
GET (Achsname, Achsname, ...):	Übernehmen der Achse(n)
GETD (Achsname, Achsname, ...):	Direktes Übernehmen der Achse(n)
Achsname:	Achszuordnung im System: AX1, AX2, ... oder Angabe der Maschinenachsenamen

RELEASE (S1):	Freigeben der Spindel S1, S2, ...
GET (S2):	Übernehmen der Spindel S1, S2, ...
GETD (S3):	Direktes Übernehmen der Spindel S1, S2, ...

GET-Anforderung ohne Vorlaufstopp

Wird nach einer GET-Anforderung ohne Vorlaufstopp die Achse mit RELEASE (Achse) oder WAITP (Achse) wieder freigegeben, so führt ein nachfolgender GET zu einem GET mit Vorlaufstopp.

	<p>VORSICHT</p> <p>Achszuordnung geändert</p> <p>Eine mit GET übernommene Achse bzw. Spindel bleibt auch nach einem Tasten- oder Programm-RESET diesem Kanal zugeordnet.</p> <p>Bei neuem Programmstart muss die Zuordnung der getauschten Achsen bzw. Spindeln programmtechnisch erfolgen, falls die Achse in ihrem Grundkanal benötigt wird.</p> <p>Bei POWER ON wird sie dem im Maschinendatum hinterlegten Kanal zugeordnet.</p>
---	--

Beispiele

Beispiel 1: Achstausch zwischen zwei Kanälen

Von 6 Achsen werden in Kanal 1 zur Bearbeitung benutzt: 1., 2., 3. und 4. Achse. 5. und 6. Achse werden in Kanal 2 zum Werkstückwechsel benutzt.

Achse 2 soll zwischen beiden Kanälen getauscht werden können und nach POWER ON dem Kanal 1 zugeordnet sein.

Programm "MAIN" in Kanal 1:

Programmcode	Kommentar
INIT (2,"TAUSCH2")	; Programm TAUSCH2 im Kanal 2 anwählen.
N... START (2)	; Programm in Kanal 2 starten.
N... GET (AX2)	; Achse AX2 übernehmen.
...	
N... RELEASE (AX2)	; Achse AX2 freigeben.
N... WAITM (1,1,2)	; Warten auf WAIT-Marke in Kanal 1 und 2 zur Synchronisation in den beiden Kanälen.
...	; Weiterer Ablauf nach Achstausch.
N... M30	

Programm "TAUSCH2" in Kanal 2:

Programmierung	Kommentar
N... RELEASE (AX2)	

Programmierung	Kommentar
N160 WAITM(1,1,2)	; Warten auf WAIT-Marke in Kanal 1 und 2 zur Synchronisation in den beiden Kanälen.
N150 GET(AX2)	; Achse AX2 übernehmen.
...	; Weiterer Ablauf nach Achstausch.
N... M30	

Beispiel 2: Achstausch ohne Synchronisierung

Wenn die Achse nicht synchronisiert werden muss, wird durch GET kein Vorlaufstopp erzeugt.

Programmierung	Kommentar
N01 G0 X0	
N02 RELEASE(AX5)	
N03 G64 X10	
N04 X20	
N05 GET(AX5)	; Wenn keine Synchronisation nötig, wird dies kein ausführbarer Satz.
N06 G01 F5000	; Kein ausführbarer Satz.
N07 X20	; Kein ausführbarer Satz, da X-Position wie in N04.
N08 X30	; Erster ausführbarer Satz nach N05.
...	

Beispiel 3: Aktivierung eines Achstausches ohne Vorlaufstopp

Voraussetzung: Der Achstausch ohne Vorlaufstopp muss über ein Maschinendatum projiziert werden.

Programmierung	Kommentar
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	; Achse B wird zur neutralen Achse.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; Achse löst kein Vorlaufstopp/REORG aus.
N041 G4 F2	
N050 M5	
N099 M30	

Wird die Spindel bzw. Achse B unmittelbar nach dem Satz N023 als **PLC-Achse** z. B. auf 180 Grad und zurück auf 1 Grad verfahren, dann wird diese Achse wieder zur neutralen Achse und löst im Satz N40 keinen Vorlaufstopp auf.

Weitere Informationen

Voraussetzungen für den Achstausch

- Die Achse muss über Maschinendaten in allen Kanälen definiert sein, die Achse verwenden wollen.
- Über das **achs**spezifische Maschinendatum muss festgelegt sein, welchem Kanal die Achse nach POWER ON zugeordnet werden soll.

Beschreibung

Achse freigeben: RELEASE

Bei der Achsfreigabe ist zu beachten:

1. Die Achse darf an keiner Transformation beteiligt sein.
2. Bei Achskopplungen (Tangentialsteuerung), müssen alle Achsen des Verbands freigegeben werden.
3. Eine konkurrierende Positionierachse kann in diesem Zustand nicht getauscht werden.
4. Bei einer Gantry-Masterachse werden auch alle Folgeachsen getauscht.
5. Bei Achskopplungen (Mitschleppen, Leitwertkopplung, Elektronisches Getriebe) kann nur die Leitachse des Verbandes freigegeben werden.

Achse übernehmen: GET

Mit diesem Befehl wird der eigentliche Achstausch durchgeführt. Die Verantwortung für die Achse liegt vollständig bei dem Kanal, in dem der Befehl programmiert wurde.

Auswirkungen von GET:

Achstausch mit Synchronisierung:

Eine Achse muss immer dann synchronisiert werden, wenn sie zwischenzeitlich in einem anderen Kanal oder der PLC zugeordnet war, und vor dem GET keine Synchronisierung durch "WAITP", G74 oder Restweglöschen stattgefunden hat.

- Ein Vorlaufstopp erfolgt (wie bei STOPRE).
- Die Bearbeitung wird so lange unterbrochen, bis der Tausch vollständig ausgeführt ist.

Automatisches "GET"

Wenn eine Achse prinzipiell im Kanal verfügbar, jedoch derzeit nicht als "Kanal-Achse" vorhanden ist, wird automatisch ein GET ausgeführt. Falls die Achse(n) schon synchronisiert ist (sind), wird kein Vorlaufstopp erzeugt.

Achstauschverhalten veränderbar einstellen

Der Abgabezeitpunkt von Achsen lässt sich über ein Maschinendatum wie folgt einstellen:

- Automatischer Achstausch findet zwischen zwei Kanälen auch dann statt, wenn die Achse durch WAITP in einen neutralen Zustand gebracht wurde (Verhalten wie bisher)
- Bei der Anforderung einer Achs-Containerdrehung werden alle dem ausführenden Kanal zuordenbaren Achsen des Achs-Containers mittels impliziten GET bzw. GETD in den Kanal geholt. Ein anschließender Achstausch ist erst nach dem Abschluss der Achs-Containerdrehung wieder erlaubt.
- Nach einem eingeschobenen Zwischensatz im Hauptlauf wird geprüft, ob ein Reorganisieren erforderlich ist oder nicht. Nur wenn die Achszustände dieses Satzes mit den aktuellen Achszuständen **nicht** übereinstimmen, ist ein Reorganisieren erforderlich.
- Statt eines GET-Satzes mit Vorlaufstopp und Synchronisation zwischen Vorlauf und Hauptlauf kann ein Achstausch auch ohne Vorlaufstopp erfolgen. Es wird dann nur ein Zwischensatz mit der GET-Anforderung erzeugt. Im Hauptlauf wird bei Abarbeitung dieses Satzes überprüft, ob die Zustände der Achse im Satz mit den aktuellen Achszuständen übereinstimmen.

Weitere Informationen zur Funktionalität eines Achs- oder Spindeltausches siehe Funktionshandbuch Erweiterungsfunktionen; BAGs, Kanäle, Achstausch (K5).

2.16 Achse einem anderen Kanal übergeben (AXTOCHAN)

Mit dem Sprachbefehl `AXTOCHAN` kann eine Achse angefordert werden, um diese Achse einem anderen Kanal zu übergeben. Die Achse kann sowohl vom NC-Teilprogramm als auch aus einer Synchronaktion heraus in den entsprechenden Kanal gebracht werden.

Syntax

```
AXTOCHAN (Achsname, Kanalnummer [, Achsname, Kanalnummer [, ... ] ] )
```

Bedeutung

Element	Beschreibung
AXTOCHAN:	Achse für einen bestimmten Kanal anfordern
Achsname:	Achszuordnung im System: X, Y, ... oder Angabe der beteiligten Maschinenachsennamen. Der auszuführende Kanal muss nicht der eigene Kanal sein und es muss auch nicht der Kanal sein, der aktuell das Interpolationsrecht für die Achse besitzt
Kanalnummer:	Nummer des Kanals, dem die Achse zugeordnet werden soll

Hinweis

Konkurrierende Positionierachse und ausschließlich PLC kontrollierte Achse

Eine PLC-Achse kann als konkurrierende Positionierachse den Kanal nicht wechseln. Eine ausschließlich von der PLC kontrollierte Achse kann nicht dem NC-Programm zugeordnet werden.

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

Beispiel

AXTOCHAN im NC-Programm

Die Achsen X und Y sind im 1. Kanal und im 2. Kanal bekannt. Aktuell hat der Kanal 1 das Interpolationsrecht und im Kanal 1 wird folgendes Programm gestartet:

Programmcode	Kommentar
N110 AXTOCHAN(Y,2)	; Y-Achse in den 2. Kanal schieben.
N111 M0	
N120 AXTOCHAN(Y,1)	; Y-Achse wieder zurückholen (neutral).
N121 M0	
N130 AXTOCHAN(Y,2,X,2)	; Y-Achse und X-Achse in den 2. Kanal schieben (Achsen neutral).
N131 M0	
N140 AXTOCHAN(Y,2)	; Y-Achse in den 2. Kanal schieben (NC-Programm).
N141 M0	

Weitere Informationen

AXTOCHAN im NC-Programm

Dabei wird nur bei einer Anforderung der Achse für das NC-Programm im eigenen Kanal ein GET durchgeführt und damit auch auf die tatsächliche Zustandsänderung gewartet. Wird die Achse für einen anderen Kanal angefordert oder soll sie zur neutralen Achse im eigenen Kanal werden, dann nur wird die Anforderung entsprechend abgesetzt.

AXTOCHAN aus einer Synchronaktion

Wird eine Achse für den eigenen Kanal angefordert so wird AXTOCHAN aus einer Synchronaktion auf ein GET aus einer Synchronaktion abgebildet. In diesem Fall wird die Achse bei der ersten Anforderung für den eigenen Kanal zur neutralen Achse. Bei der zweiten Anforderung wird die Achse dem NC-Programm analog zur GET-Anforderung im NC-Programm zugeordnet. Zur GET-Anforderung aus einer Synchronaktion siehe Kapitel "Bewegungssynchronaktionen".

2.17 Maschinendaten wirksam setzen (NEWCONF)

Mit dem Befehl `NEWCONF` werden alle Maschinendaten der Wirksamkeitsstufe "NEW_CONFIG" wirksam gesetzt. Die Funktion kann auch in der Bedienoberfläche HMI durch Betätigen des Softkeys "MD wirksam setzen" aktiviert werden.

Bei der Ausführung der Funktion "NEWCONF" erfolgt ein impliziter Vorlaufstopp, d. h. die Bahnbewegung wird unterbrochen.

Syntax

```
NEWCONF
```

Bedeutung

NEWCONF:	Befehl zum Wirksamsetzen aller Maschinendaten der Wirksamkeitsstufe "NEW_CONFIG"
----------	--

NEWCONF aus dem Teileprogramm kanalübergreifend ausführen

Werden axiale Maschinendaten aus dem Teileprogramm heraus verändert und anschließend mit `NEWCONF` aktiviert, so setzt der Befehl `NEWCONF` nur die Maschinendaten aktiv, die Änderungen für den Kanal des Teileprogramms bewirken.

Hinweis

Um alle Änderungen sicher wirksam werden zu lassen, muss der Befehl `NEWCONF` in jedem Kanal ausgeführt werden, in dem auch die von den veränderten Maschinendaten betroffenen Achsen oder Funktionen aktuell gerechnet werden.

Bei `NEWCONF` werden keine axialen Maschinendaten wirksam gesetzt.

Für PLC-kontrollierte Achsen muss ein axialer RESET ausgeführt werden.

Beispiel

Fräsbearbeitung: Bohrposition mit verschiedenen Technologien bearbeiten

Programmcode	Kommentar
N10 \$MA_CONTOUR_TOL[AX]=1.0	; Maschinendatum ändern.
N20 NEWCONF	; Maschinendaten wirksam setzen.
...	

2.18 Datei schreiben (WRITE)

Mit dem WRITE-Befehl können Sätze/Daten aus dem NC-Programm an das Ende einer sich im passiven Filesystem oder auf einem externen Programmspeicher befindlichen Datei (Protokolldatei) geschrieben werden. Dies kann auch das gerade in Abarbeitung befindliche Programm sein.

Hinweis

Eine per WRITE-Befehl zu beschreibende Datei wird neu angelegt, wenn sie nicht im Programmspeicher existiert.

Voraussetzung

Die aktuell eingestellte Schutzstufe muss gleich oder größer dem WRITE-Recht der Datei sein. Ist dies nicht der Fall, wird der Zugriff mit Fehlermeldung (Rückgabewert der Fehlervariablen = 13) abgelehnt.

Syntax

```
DEF INT <Fehler>
...
WRITE (<Fehler>, "<Dateiname>"/"<ExtG>", "<Satz/Daten>")
```

Bedeutung

WRITE:	Befehl zum Anfügen eines Satzes bzw. von Daten an das Ende der angegebenen Datei		
<Fehler>:	Parameter 1: Variable für die Rückgabe des Fehlerwerts		
	Typ:	INT	
	Wert:	0	kein Fehler
		1	Pfad nicht erlaubt
		2	Pfad nicht gefunden
		3	Datei nicht gefunden
		4	falscher Dateityp
		10	Datei ist voll
		11	Datei wird benutzt
		12	keine Ressourcen frei
		13	keine Zugriffsrechte
		14	fehlendes oder fehlgeschlagenes EXTOPEN für das Ausgabegerät
15		Fehler beim Schreiben auf externes Gerät	
16	ungültiger externer Pfad programmiert		

<Dateiname>:	Parameter 2: Name der Datei, in der der angegebene Satz bzw. die angegebenen Daten angefügt werden sollen	
	Typ:	STRING
	Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht. Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)".	
<ExtG>:	Sollen Daten über die Funktion "Process DataShare" auf ein externes Gerät/Datei ausgegeben werden, muss statt des Dateinamens der symbolische Bezeichner für das zu öffnende externe Gerät/Datei angegeben werden.	
	Typ:	STRING
	Weitere Informationen siehe "Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE) (Seite 580)". Hinweis: Der Bezeichner muss identisch zu dem im EXTOPEN-Befehl angegebenen Bezeichner sein.	
<Satz/Daten>:	Parameter 3: Satz bzw. Daten, die in der angegebenen Datei angefügt werden sollen.	
	Typ:	STRING

Hinweis

Beim Schreiben ins passive Dateisystem oder auf einen externen Programmspeicher fügt der WRITE-Befehl implizit ein "LF"-Zeichen (LINE FEED = neue Zeile) an das Ende des Ausgabestrings an.

Für die Ausgabe auf ein externes Gerät/Datei über die Funktion "Process DataShare" gilt dieses Verhalten nicht. Soll ein "LF" mit ausgegeben werden, muss das explizit im Ausgabestring mit angegeben werden.

→ Siehe hierzu Beispiel 3: Implizites/explicites "LF"!

Randbedingungen

- **Maximale Dateigröße (→ Maschinenhersteller!)**
Die maximal mögliche Dateigröße von Protokolldateien im passiven Filesystem wird eingestellt mit dem Maschinendatum:
MD11420 \$MN_LEN_PROTOCOL_FILE
Die maximale Dateigröße gilt für alle Dateien, die mit dem WRITE-Befehl im passiven Filesystem angelegt werden. Bei Überschreitung wird eine Fehlermeldung ausgegeben und der Satz bzw. die Daten werden nicht abgespeichert. Sofern der Speicher ausreicht, kann eine neue Datei angelegt werden.

Beispiele**Beispiel 1: WRITE-Befehl ins passive Filesystem ohne absolute Pfadangabe**

Programmcode	Kommentar
N10 DEF INT ERROR	; Definition der Fehlervariablen.

Programmcode	Kommentar
N20 WRITE(ERROR,"PROT","PROTOKOLL VOM 7.2.97")	; Schreibe den Text "PROTOKOLL VOM 7.2.97" in die Datei _N_PROT_MPF.
N30 IF ERROR	; Fehlerauswertung.
N40 MSG ("Fehler bei WRITE-Befehl:" << ERROR)	
N50 M0	
N60 ENDIF	
...	

Beispiel 2: WRITE-Befehl ins passive Filesystem mit absoluter Pfadangabe

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

Beispiel 3: Implizites/explizites "LF"

a, Schreiben ins passive Dateisystem mit implizit erzeugtem "LF"

Programmcode
...
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

Ausgabeergebnis:

MY_STRING

MY_STRING

b, Schreiben in externe Datei ohne implizit erzeugtem "LF"

Programmcode
...
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

Ausgabeergebnis:

MY_STRINGMY_STRING

c, Schreiben in externe Datei mit explizit programmiertem "LF"

Um dasselbe Ergebnis wie bei a, zu erzielen, muss Folgendes programmiert werden:

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

Ausgabeergebnis:

MY_STRING

MY_STRING

2.19 Datei löschen (DELETE)

Mit dem DELETE-Befehl können alle Dateien gelöscht werden, egal, ob diese per WRITE-Befehl entstanden sind oder nicht. Auch Dateien, die unter höherer Zugriffsstufe erstellt wurden, können mit DELETE gelöscht werden.

Syntax

```
DEF INT <Fehler>
DELETE(<Fehler>,"<Dateiname>")
```

Bedeutung

DELETE:	Befehl zum Löschen der angegebenen Datei			
<Fehler>:	Variable für die Rückgabe des Fehlerwerts			
	Typ:	INT		
	Wert:	0	kein Fehler	
		1	Pfad nicht erlaubt	
		2	Pfad nicht gefunden	
		3	Datei nicht gefunden	
		4	falscher Dateityp	
		11	Datei wird benutzt	
		12	keine Ressourcen frei	
20		sonstiger Fehler		
<Dateiname>:	Name der zu löschenden Datei			
	Typ:	STRING		
	Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht. Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)".			

Beispiel

Programmcode	Kommentar
N10 DEF INT ERROR	; Definition der Fehlervariablen.
N15 STOPRE	; Vorlaufstopp.
N20 DELETE (ERROR, "/_N_SPF_DIR/_N_TEST1_SPF")	; Lösche die Datei TEST1 im Unterprogrammverzeichnis.
N30 IF ERROR	; Fehlerauswertung.
N40 MSG("Fehler bei DELETE-Befehl:" <<ERROR)	
N50 M0	
N60 ENDIF	

2.20 Zeilen in Datei lesen (READ)

Der READ-Befehl liest in der angegebenen Datei eine oder mehrere Zeilen und legt die gelesenen Informationen in einem Feld vom Typ STRING ab. Jede gelesene Zeile belegt in diesem Feld ein Feldelement.

Voraussetzung

Die aktuell eingestellte Schutzstufe muss gleich oder größer dem READ-Recht der Datei sein. Ist dies nicht der Fall, wird der Zugriff mit Fehlermeldung (Rückgabewert der Fehlervariablen = 13) abgelehnt.

Syntax

```
DEF INT <Fehler>
DEF STRING[<Stringlänge>] <Ergebnis>[<n>,<m>]
READ(<Fehler>,"<Dateiname>",<Anfangszeile>,<Zeilenanzahl>,<Ergebnis>
)
```

Bedeutung

READ:	Befehl zum Lesen von Zeilen der angegebenen Datei und zur Ablage dieser Zeilen in einem Variablenfeld.		
<Fehler>:	Variable für die Rückgabe des Fehlerwerts (Call-By-Reference-Parameter)		
	Typ:	INT	
	Wert:	0	kein Fehler
		1	Pfad nicht erlaubt
		2	Pfad nicht gefunden
		3	Datei nicht gefunden
		4	falscher Dateityp
		11	Datei wird benutzt
		13	Zugriffsrechte nicht ausreichend
		21	Zeile nicht vorhanden (Parameter <Anfangszeile> oder <Zeilenanzahl> größer als Anzahl der Zeilen in der angegebenen Datei).
22		Feldlänge der Ergebnisvariablen (<Ergebnis>) ist zu klein.	
23	Zeilenbereich zu groß (Parameter <Zeilenanzahl> so groß gewählt, dass über das Dateiende hinausgelesen wird).		
<Dateiname>:	Name der zu lesenden Datei (Call-By-Value-Parameter)		
	Typ:	STRING	
Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht. Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)".			
<Anfangszeile>:	Anfangszeile des zu lesenden Dateibereichs (Call-By-Value-Parameter)		
	Typ:	INT	
	Wert:	0	Es werden die mit dem Parameter <Zeilenanzahl> angegebene Anzahl an Zeilen vor dem Dateiende gelesen.
1 ... n		Nummer der ersten zu lesenden Zeile.	
<Zeilenanzahl>:	Anzahl der zu lesenden Zeilen (Call-By-Value-Parameter)		
	Typ:	INT	

2.21 Vorhandensein einer Datei prüfen (ISFILE)

<Ergebnis>:	Ergebnisvariable (Call-By-Reference-Parameter) Variablenfeld, in dem der gelesene Text abgelegt wird.	
	Typ:	STRING (max. Länge: 255)
Wenn im Parameter <Zeilenanzahl> weniger Zeilen angegeben sind als die Feldgröße [<n> , <m>] der Ergebnisvariablen beträgt, dann werden die restlichen Feldelemente nicht verändert. Der Abschluss einer Zeile durch die Steuerzeichen "LF" (Line Feed) oder "CR LF" (Carriage Return Line Feed) wird nicht in der Ergebnisvariablen abgelegt. Gelesene Zeilen werden abgeschnitten, wenn die Zeile länger ist als die definierte Stringlänge. Es erfolgt keine Fehlermeldung.		

Hinweis

Binäre Files können nicht eingelesen werden. Es wird der Fehler "falscher Dateityp" (Rückgabewert der Fehlervariablen = 4) ausgegeben. Folgenden Dateitypen sind nicht lesbar: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Beispiel

Programmcode	Kommentar
N10 DEF INT ERROR	; Definition der Fehlervariablen.
N20 DEF STRING[255] RESULT[5]	; Definition der Ergebnisvariablen.
N30 READ(ERROR, "/_N_CST_DIR/_N_TESTFILE_MPF", 1,5,RESULT)	; Dateiname mit Domain-, Dateikennung und Pfadangabe.
N40 IF ERROR <>0	; Fehlerauswertung.
N50 MSG("FEHLER"<<ERROR<<"BEI READ-BEFEHL")	
N60 M0	
N70 ENDIF	
...	

2.21 Vorhandensein einer Datei prüfen (ISFILE)

Mit dem ISFILE-Befehl kann geprüft werden, ob eine Datei im Programmspeicher existiert.

Syntax

`<Ergebnis>=ISFILE("<Dateiname>")`

Bedeutung

ISFILE:	Befehl zum Prüfen des Vorhandenseins einer Datei		
<Dateiname>:	Name der Datei, deren Vorhandensein geprüft werden soll.		
	Typ:	STRING	
<Ergebnis>:	Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht. Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)".		
	Ergebnisvariable zur Aufnahme des Prüfergebnisses		
	Typ.	BOOL	
	Wert:	TRUE	Datei vorhanden
		FALSE	Datei nicht vorhanden

Beispiel

Programmcode	Kommentar
N10 DEF BOOL RESULT	; Definition der Ergebnisvariablen.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF (RESULT==FALSE)	
N40 MSG("DATEI NICHT VORHANDEN")	
N50 M0	
N60 ENDIF	
...	

oder:

Programmcode	Kommentar
N10 DEF BOOL RESULT	; Definition der Ergebnisvariablen.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF (NOT ISFILE("TESTFILE"))	
N40 MSG("DATEI NICHT VORHANDEN")	
N50 M0	
N60 ENDIF	
...	

2.22 Datei-Informationen auslesen (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Über die Befehle FILEDATE, FILETIME, FILESIZE, FILESTAT und FILEINFO können bestimmte Datei-Informationen wie Datum/Uhrzeit des letzten schreibenden Zugriffs, aktuelle Dateigröße, Datei-Status oder die Summe dieser Informationen ausgelesen werden.

Voraussetzung

Die aktuell eingestellte Schutzstufe muss gleich oder größer dem Show-Recht des übergeordneten Verzeichnisses sein. Ist dies nicht der Fall, wird der Zugriff mit Fehlermeldung (Rückgabewert der Fehlervariablen = 13) abgelehnt.

Syntax

```
FILE....(<Fehler>,"<Dateiname>",<Ergebnis>)
```

Bedeutung

FILEDATE:	Liefert das Datum des letzten Schreibzugriffs auf eine Datei		
FILETIME:	Liefert die Uhrzeit des letzten Schreibzugriffs auf eine Datei		
FILESIZE:	Liefert die aktuelle Größe einer Datei		
FILESTAT:	Liefert für den Status einer Datei bezüglich folgender Rechte : <ul style="list-style-type: none"> • Lesen (r: read) • Schreiben (w: write) • Ausführen (x: execute) • Anzeigen (s: show) • Löschen (d: delete) Hinweis: Diese Schutzstufen sind spezielle Eigenschaften des passiven Filesystems. Beim Zugriff auf externe Programmspeicher liefert FILESTAT daher nur Default-Zugriffsrechte (77777).		
FILEINFO:	Liefert für eine Datei die Summe der Informationen , die über FILEDATE, FILETIME, FILESIZE und FILESTAT auslesbar sind		
<Fehler>:	Variable für die Rückgabe des Fehlerwerts (Call-By-Reference-Parameter)		
	Typ:	VAR INT	
	Wert:	0	kein Fehler
		1	Pfad nicht erlaubt
		2	Pfad nicht gefunden
		3	Datei nicht gefunden
		4	falscher Dateityp
		13	Zugriffsrechte nicht ausreichend
22		Stringlänge der Ergebnisvariablen (<Ergebnis>) ist zu klein.	
<Dateiname>:	Name der Datei, von der Datei-Information(en) ausgelesen werden soll(en)		
	Typ:	CHAR[160]	
	Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht. Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)".		

<Ergebnis>:	Ergebnisvariable (Call-By-Reference-Parameter) Variable, in der die angeforderte Datei-Information abgelegt wird.			
	Typ:	VAR CHAR[8]	bei	FILEDATE Format: "dd.mm.yy"
		VAR CHAR[8]	bei	FILETIME Format: "hh:mm:ss"
		VAR INT	bei	FILESIZE Die Dateigröße wird in Byte ausgegeben.
		VAR CHAR[5]	bei	FILESTAT Format: "rwxsd" (r: read, w: write, x: execute, s: show, d: delete)
		VAR CHAR[32]	bei	FILEINFO Format: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

Beispiel

Programmcode	Kommentar
N10 DEF INT ERROR	; Definition der Fehlervariablen.
N20 STRING[32] RESULT	; Definition der Ergebnisvariablen.
N30 FILEINFO(ERROR, "/_N_MPF_DIR/_N_TESTFILE_MPF", RESULT)	; Dateiname mit Domain-, Dateikennung und Pfadangabe.
N40 IF ERROR <>0	; Fehlerauswertung
N50 MSG("FEHLER"<<ERROR<<"BEI FILEINFO-BEFEHL")	
N60 M0	
N70 ENDIF	
...	

Das Beispiel könnte in der Ergebnisvariablen RESULT z. B. folgendes Ergebnis liefern:

```
"77777 12345678 26.05.00 13:51:30"
```

2.23 Aufrunden (ROUNDUP)

Mit der Funktion "ROUNDUP" können Eingabewerte vom Typ REAL (gebrochene Zahlen mit Dezimalpunkt) auf die nächste größere ganze Zahl aufgerundet werden.

Syntax

```
ROUNDUP(<Wert>)
```

Bedeutung

ROUNDUP:	Befehl zum Aufrunden eines Eingabewerts
<Wert>:	Eingabewert vom Typ REAL

Hinweis

Eingabewerte vom Typ INTEGER (eine ganze Zahl) werden unverändert zurückgeliefert.

Beispiele

Beispiel 1: Verschiedene Eingabewerte und deren Rundungsergebnisse

Beispiel	Rundungsergebnis
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

Beispiel 2: ROUNDUP im NC-Programm

```
Programmcode
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 Unterprogrammtechnik

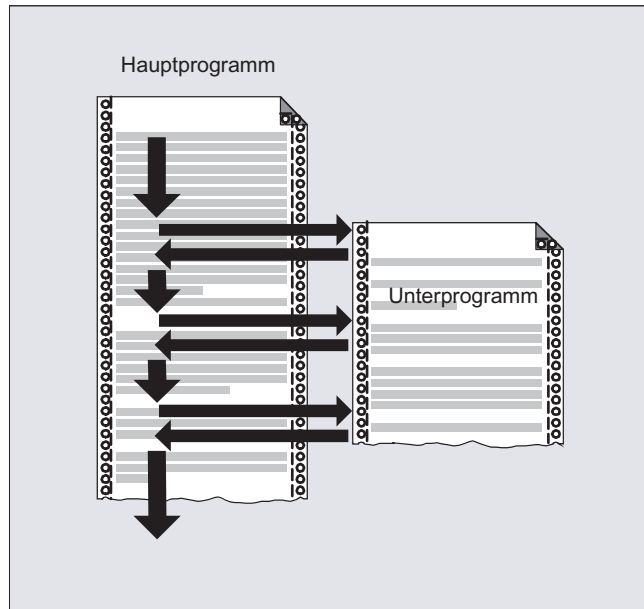
2.24.1 Allgemeines

2.24.1.1 Unterprogramm

Die Bezeichnung "Unterprogramm" stammt noch aus der Zeit, als Teileprogramme fest in Haupt- und Unterprogramme unterteilt waren. Hauptprogramme waren dabei die Teileprogramme, die an der Steuerung zum Abarbeiten angewählt und dann gestartet wurden. Unterprogramme waren die Teileprogramme, die vom Hauptprogramm aus aufgerufen wurden.

Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Jedes Teileprogramm kann prinzipiell als Hauptprogramm angewählt und gestartet oder als Unterprogramm von einem anderen Teileprogramm aus aufgerufen werden.

Somit wird im weiteren Verlauf mit Unterprogramm ein Teileprogramm bezeichnet, das von einem anderen Teileprogramm aus aufgerufen wird.



Anwendung

Wie in allen höheren Programmiersprachen werden auch in der NC-Sprache Unterprogramme dazu angewandt, um Programmteile, die mehrfach verwendet werden, in eigenständige, in sich abgeschlossene Programme auszulagern.

Unterprogrammen bieten folgende Vorteile:

- Erhöhen die Übersichtlichkeit und Lesbarkeit der Programme
- Erhöhen die Qualität durch Wiederverwendung getesteter Programmteile
- Bieten die Möglichkeit zur Schaffung spezifischer Bearbeitungsbibliotheken
- Sparen Speicherplatz

2.24.1.2 Unterprogrammnamen

Benennungsregeln

Der Unterprogrammname kann unter Einhaltung folgender Regeln frei gewählt werden:

- Erlaubte Zeichen:
 - Buchstaben: A ... Z, a ... z
 - Ziffern: 0 ... 9
 - Unterstrich: _
- Die ersten beiden Zeichen **müssen** zwei Buchstaben oder ein Unterstrich gefolgt von einem Buchstabe sein.
- Maximale Länge: 24 Zeichen

Groß- / Kleinbuchstaben

In der SINUMERIK NC-Sprache wird **nicht** zwischen Groß- und Kleinbuchstaben unterschieden.

Steuerungsinterne Erweiterungen

Der bei der Programmerstellung vergebene Unterprogrammname wird steuerungsintern mit einem Pre- und Postfix erweitert:

- Prefix: `_N_`
- Postfix: `_SPF`

Verwendung des Programmnamens

Bei der Verwendung des Programmnamens, z. B. bei einem Unterprogrammaufruf, sind alle Kombinationen von Prefix, Programmnamen und Postfix möglich.

Beispiel:

Das Unterprogramm mit dem Programmnamen `SUB_PROG` kann über folgende Bezeichner aufgerufen werden:

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

Namensgleichheit bei Haupt- und Unterprogrammen

Existiert ein Hauptprogramme (.MPF) und ein Unterprogramme (.SPF) mit gleichem Programmnamen, muss bei der Verwendung des Programmnamens im NC-Programm die entsprechende Dateierweiterung zur eindeutigen Kennzeichnung mit angegeben werden. Ansonsten wird das Programm verwendet, das entlang des Suchpfades als erstes mit dem angegebenen Programmnamen gefunden wird.

2.24.1.3 Schachtelung von Unterprogrammen

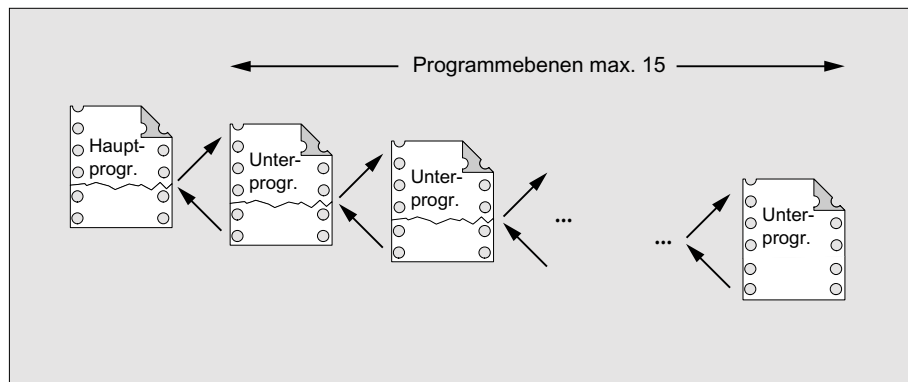
Ein Hauptprogramm kann Unterprogramme aufrufen, die wiederum Unterprogramme aufrufen. Die Abläufe der Programme sind somit ineinander geschachtelt. Jedes Programm läuft dabei in einer eigenen Programmebene.

Schachtelungstiefe

Die NC-Sprache stellt aktuell 16 Programmebenen zur Verfügung. Das Hauptprogramm läuft immer in der obersten Programmebene 0. Ein Unterprogramm läuft immer in der dem Aufruf folgenden nächstniedrigeren Programmebene. Die Programmebene 1 ist somit die erste Unterprogrammebene.

Unterteilung der Programmebenen:

- Programmebene 0: Hauptprogrammebene
- Programmebene 1 - 15: Unterprogrammebene 1 - 15



Interruptroutinen (ASUP)

Wird im Rahmen einer Interruptroutine ein Unterprogramm aufgerufen, wird dieses nicht in der aktuellen im Kanal aktiven Programmebene (n), sondern ebenfalls in der nächstniedrigeren Programmebene (n+1) abgearbeitet. Damit dies auch in der untersten Programmebene noch möglich ist, stehen im Zusammenhang mit Interruptroutinen 2 zusätzliche Programmebenen (16 und 17) zur Verfügung.

Werden mehr als 2 Programmebenen benötigt, muss dies explizit in der Strukturierung des im Kanal abgearbeiteten Teileprogramms berücksichtigt werden. D. h. es darf dann maximal nur so viele Programmebenen beanspruchen, dass noch ausreichend Programmebenen für die Interruptbearbeitung zur Verfügung stehen.

Benötigt die Interruptbearbeitung z. B. 4 Programmebenen, muss das Teileprogramm so strukturiert werden, dass es maximal Programmebene 13 belegt. Erfolgt dann ein Interrupt, stehen diesem die benötigten 4 Programmebenen (14 bis 17) zur Verfügung.

Siemens-Zyklen

Siemens-Zyklen benötigen 3 Programmebenen. Der Aufruf eines Siemens-Zyklus muss daher spätestens erfolgen in:

- Teileprogrammbearbeitung: Programmebene 12
- Interruptroutine: Programmebene 14

2.24.1.4 Suchpfad

Beim Aufruf eines Unterprogramms ohne Pfadangabe durchsucht die Steuerung nach einer vordefinierten Suchreihenfolge (siehe "Suchpfad bei Unterprogrammaufruf (Seite 214)") die vorhandenen Programmspeicher.

2.24.1.5 Formal- und Aktualparameter

Von Formal- und Aktualparameter spricht man im Zusammenhang mit der Definition und dem Aufruf von Unterprogrammen mit Parameterübergabe.

Formalparameter

Bei der Definition eines Unterprogramms müssen die dem Unterprogramm zu übergebenden Parameter, die sogenannten Formalparameter, mit Typ und Parameternamen definiert werden.

Die Formalparameter definieren somit die Schnittstelle des Unterprogramms.

Beispiel:

Programmcode	Kommentar
PROC KONTUR (REAL X, REAL Y)	; Formalparameter: X und Y beide vom Typ REAL
N20 X1=X Y1=Y	; Verfahren der Achse X1 auf Position X und der Achse Y1 auf Position Y
...	
N100 RET	

Aktualparameter

Beim Aufruf eines Unterprogramms müssen dem Unterprogramm absolute Werte oder Variablen, die sogenannten Aktualparameter, übergeben werden.

Die Aktualparameter befüllen somit beim Aufruf die Schnittstelle des Unterprogramms mit aktuellen Werten.

Beispiel:

Programmcode	Kommentar
N10 DEF REAL BREITE	; Variablendefinition
N20 BREITE=20.0	; Variablenzuweisung
N30 KONTUR(5.5, BREITE)	; Unterprogrammaufruf mit Aktualparametern: 5.5 und BREITE

Programmcode	Kommentar
...	
N100 M30	

2.24.1.6 Parameterübergabe

Definition eines Unterprogramms mit Parameterübergabe

Die Definition eines Unterprogramms mit Parameterübergabe erfolgt mit dem Schlüsselwort PROC und einer vollständigen Auflistung aller vom Unterprogramm erwarteten Parameter.

Unvollständige Parameterübergabe

Beim Aufruf des Unterprogramms müssen nicht immer alle in der Unterprogrammchnittstelle definierten Parameter explizit übergeben werden. Wird ein Parameter weggelassen, wird für diesen Parameter der Standardwert "0" übergeben.

Zur eindeutigen Kennzeichnung der Reihenfolge der Parameter müssen allerdings die Kommas als Trennzeichen der Parameter immer mit angegeben werden. Eine Ausnahme bildet der letzte Parameter. Wird dieser beim Aufruf weggelassen, kann auch das letzte Komma entfallen.

Beispiel:

Unterprogramm:

Programmcode	Kommentar
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; Formalparameter: X, Y und Z
...	
N100 RET	

Hauptprogramm:

Programmcode	Kommentar
PROC MAIN_PROG	
...	
N30 SUB_PROG(1.0,2.0,3.0)	; Unterprogrammaufruf mit vollständiger Parameterübergabe: X=1.0, Y=2.0, Z=3.0
...	
N100 M30	

Beispiele für den Unterprogrammaufruf in N30 mit unvollständiger Parameterübergabe:

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
```

ACHTUNG

Parameterübergabe Call-by-Reference

Parameter, die über Call-by-Reference übergeben werden, dürfen beim Unterprogramm-Aufruf nicht weggelassen werden.

ACHTUNG

Datentyp AXIS

Parameter vom Datentyp AXIS dürfen beim Unterprogramm-Aufruf nicht weggelassen werden.

Überprüfung der Übergabeparameter

Über die Systemvariable \$P_SUBPAR [n] mit n = 1, 2, ... kann im Unterprogramm überprüft werden, ob ein Parameter explizit übergeben oder weggelassen wurde. Der Index n bezieht sich auf die Reihenfolge der Formalparameter. Index n = 1 bezieht sich auf den 1. Formalparameter, Index n = 2 auf den 2. Formalparameter usw.

Der folgende Programmausschnitt zeigt beispielhaft für den 1. Formalparameter, wie eine Überprüfung realisiert werden kann:

Programmierung	Kommentar
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; Formalparameter: X, Y und Z
N20 IF \$P_SUBPAR[1]==TRUE	; Überprüfung des 1.Formalparameters X.
...	; Diese Aktionen werden ausgeführt, wenn der Formalparameter X explizit übergeben wurde.
N40 ELSE	
...	; Diese Aktionen werden ausgeführt, wenn der Formalparameter X nicht übergeben wurde.
N60 ENDIF	
...	; Allgemeine Aktionen
N100 RET	

2.24.2 Definition eines Unterprogramms

2.24.2.1 Unterprogramm ohne Parameterübergabe

Bei der Definition von Unterprogrammen ohne Parameterübergabe kann die Definitionszeile am Programmanfang entfallen.

Syntax

```
[PROC <Programmname>]
...
```

Bedeutung

PROC:	Definitionsanweisung am Anfang eines Programms
<Programmname>:	Name des Programms

Beispiel

Beispiel 1: Unterprogramm mit PROC-Anweisung

Programmcode	Kommentar
PROC SUB_PROG	; Definitionszeile
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; Unterprogrammrücksprung

Beispiel 2: Unterprogramm ohne PROC-Anweisung

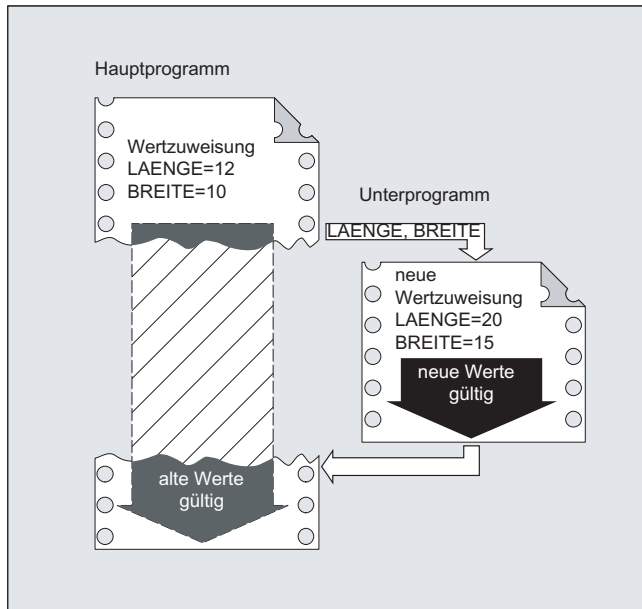
Programmcode	Kommentar
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; Unterprogrammrücksprung

2.24.2.2 Unterprogramm mit Parameterübergabe Call-by-Value (PROC)

Die Definition eines Unterprogramms mit Parameterübergabe Call-by-Reference erfolgt mit dem Schlüsselwort `PROC`, gefolgt vom Programmnamen und einer vollständigen Auflistung aller Parameter mit Typ und Namen. Die Definitionsanweisung muss in der ersten Programmzeile stehen.

Call-by-Value

Das aufrufende Programm übergibt bei einer Parameterübergabe Call-by-Reference dem Unterprogramm nur den Wert einer Variablen. Damit bekommt das Unterprogramm keinen direkten Zugriff auf die Variable. Dadurch wird bei einer Änderung des Parameterwertes nur der im Unterprogramm sichtbare Wert geändert. Der Wert der im aufrufenden Programm definierten Variablen bleibt unverändert. Die Parameterübergabe Call-by-Reference hat somit keine Rückwirkungen auf das aufrufende Programm.



Syntax

PROC <Programmname> (<Parametertyp> <Parametername>=<Init_Wert>, ...)

Hinweis

Es können bis zu 127 Parameter übergeben werden.

Bedeutung

PROC:	Definitionsanweisung am Anfang eines Programms
<Programmname>:	Name des Programms
<Parametertyp>:	Datentyp des Parameters (z. B. REAL, INT, BOOL)
<Parametername>:	Name des Parameters
<Init_Wert>:	Optionaler Wert zur Initialisierung des Parameters (optional) Wird beim Aufruf des Unterprogramms kein Parameter angegeben, erhält der Parameter den Initialisierungswert zugewiesen.

Beispiel

Definition eines Unterprogramms SUB_PROG mit drei Parametern vom Typ REAL mit Defaultwerten:

```

Programmcode
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL
HIGHT=30.0)
...
N100 RET
    
```

Verschieden Aufrufvarianten:

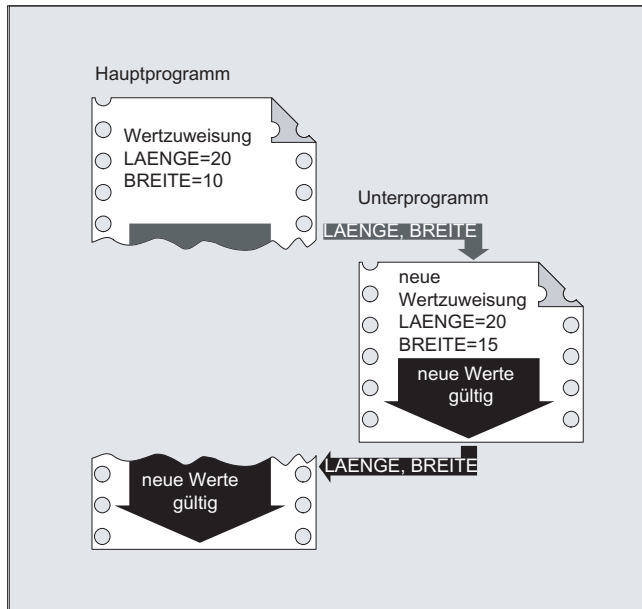
```
Programmcode
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; Aufrufvarianten
  SUB_PROG
  SUB_PROG(PAR_1, PAR_2, PAR_3)
  SUB_PROG(PAR_1)
  SUB_PROG(PAR_1, , PAR_3)
  SUB_PROG( , , PAR_3)
N100 RET
```

2.24.2.3 Unterprogramm mit Parameterübergabe Call-by-Reference (PROC, VAR)

Die Definition eines Unterprogramms mit Parameterübergabe Call-by-Reference erfolgt mit dem Schlüsselwort `PROC`, gefolgt vom Programmnamen und einer vollständigen Auflistung aller Parameter mit Schlüsselwort `VAR`, Typ und Namen. Die Definitionsanweisung muss in der ersten Programmzeile stehen. Als Parameter können auch Referenzen auf Felder übergeben werden.

Call-by-Reference

Das aufrufende Programm übergibt bei einer Parameterübergabe Call-by-Reference dem Unterprogramm nicht den Wert einer Variablen, sondern eine Referenz (Zeiger) auf die Variable. Damit bekommt das Unterprogramm direkten Zugriff auf die Variable. Dadurch wird bei einer Änderung des Parameterwertes nicht nur der im Unterprogramm sichtbare Wert geändert, sondern der Wert der im aufrufenden Programm definierten Variablen. Die Parameterübergabe Call-by-Reference hat somit auch nach Beendigung des Unterprogramms Rückwirkungen auf das aufrufende Programm.



Hinweis

Die Parameterübergabe Call-by-Reference ist nur dann erforderlich, wenn die übergebene Variable lokal in einem aufrufenden Programm definiert ist (LUD). Kanal-globale oder NC-globale Variablen müssen nicht übergeben werden, da auf diese auch direkt vom Unterprogramm aus zugegriffen werden kann.

Syntax

```
PROC <Programmname> (VAR <Parametertyp> <Parametername>, ...)
PROC <Programmname> (VAR <Feldtyp> <Feldname> [<m>,<n>,<o>], ...)
```

Hinweis

Es können bis zu 127 Parameter übergeben werden.

Bedeutung

PROC:	Definitionsanweisung am Anfang eines Programms
VAR:	Schlüsselwort für die Parameterübergabe per Referenz
<Programmname>:	Name des Programms
<Parametertyp>:	Datentyp des Parameters (z. B. REAL, INT, BOOL)
<Parametername>:	Name des Parameters
<Feldtyp>:	Datentyp der Feldelemente (z. B. REAL, INT, BOOL)
<Feldname>:	Name des Feldes

[<m>, <n>, <o>]:	Feldgröße	
	Aktuell sind maximal 3-dimensionale Felder möglich:	
	<m>:	Feldgröße für 1. Dimension
	<n>:	Feldgröße für 2. Dimension
	<o>:	Feldgröße für 3. Dimension

Hinweis

- Der nach dem Schlüsselwort `PROC` angegebene Programmname muss mit dem an der Bedienoberfläche vergebenen Programmnamen übereinstimmen.
- Mit Feldern unbestimmter Feldlänge als Formalparameter können Unterprogramme Felder variabler Länge bearbeiten. Dazu wird bei der Definition z. B. eines zweidimensionalen Feldes als Formalparameter die Länge der 1. Dimension nicht angegeben. Das Komma aber muss geschrieben werden.
Beispiel: `PROC <Programmname> (VAR REAL FELD[,5])`

Beispiel

Definition eines Unterprogramms mit 2 Parameter als Referenz auf Typ REAL:

Programmcode	Kommentar
PROC SUB_PROG(VAR REAL LAENGE, VAR REAL BREITE)	; Parameter 1: Referenz auf Typ: REAL, Name: LAENGE
	Parameter 2: Referenz auf Typ: REAL, Name: BREITE
...	
N100 RET	

2.24.2.4 Modale G-Funktionen sichern (SAVE)

Das Attribut `SAVE` bewirkt, dass die vor dem Unterprogrammaufruf aktiven modalen G-Funktionen gesichert und nach dem Unterprogrammende wieder reaktiviert werden.

ACHTUNG
Unterbrechung des Bahnsteuerbetriebs
Wird bei aktivem Bahnsteuerbetrieb ein Unterprogramme mit Attribut <code>SAVE</code> aufgerufen, wird der Bahnsteuerbetrieb am Ende des Unterprogramms (Rücksprung) unterbrochen.

Syntax

```
PROC <Unterprogrammname> SAVE
```

Bedeutung

SAVE:	Sichern der modalen G-Funktionen vor dem Unterprogrammaufruf und Wiederherstellen nach Unterprogrammende
-------	--

Beispiel

Im Unterprogramm KONTUR wirkt die modale G-Funktion G91 (Kettenmaß). Im Hauptprogramm wirkt die modale G-Funktion G90 (Absolutmaß). Durch die Unterprogrammdefinition mit SAVE wirkt nach dem Unterprogrammende im Hauptprogramm wieder G90.

Unterprogramm-Definition:

Programmcode	Kommentar
PROC KONTUR (REAL WERT1) SAVE	; Unterprogramm-Definition mit Parameter SAVE
N10 G91 ...	; Modale G-Funktion G91: Kettenmaß
N100 M17	; Unterprogrammende

Hauptprogramm:

Programmcode	Kommentar
N10 G0 X... Y... G90	; Modale G-Funktion G90: Absolutmaß
N20 ...	
...	
N50 KONTUR (12.4)	; Unterprogrammaufruf
N60 X... Y...	; Modale G-Funktion G90 durch SAVE reaktiviert

Randbedingungen

Frames

Das Verhalten von Frames bezüglich Unterprogrammen mit dem Attribut SAVE ist abhängig vom Typ des Frames und kann über Maschinendaten eingestellt werden.

Literatur

Funktionshandbuch Grundfunktionen; Achsen, Koordinatensysteme, Frames (K2), Kapitel: "Unterprogrammrücksprung mit SAVE"

2.24.2.5 Einzelsatzbearbeitung unterdrücken (SBLOF, SBLON)

Einzelsatzunterdrückung für das gesamte Programm

Mit SBLOF gekennzeichnete Programme werden bei aktiver Einzelsatzbearbeitung wie ein Satz komplett abgearbeitet, d. h. für das gesamte Programm wird die Einzelsatzbearbeitung unterdrückt.

SBLOF steht in der PROC-Zeile und gilt bis zum Ende oder Abbruch des Unterprogramms. Mit dem Rücksprung-Befehl wird entschieden, ob am Ende des Unterprogramms angehalten wird oder nicht:

Rücksprung mit M17: Stopp am Ende des Unterprogramms

Rücksprung mit RET: Kein Stopp am Ende des Unterprogramms

Einzelsatzunterdrückung innerhalb des Programms

SBLOF muss allein im Satz stehen. Ab diesem Satz wird Einzelsatz ausgeschaltet bis:

- zum nächsten SBLON
oder
- zum Ende der aktiven Unterprogrammebene

Syntax

Einzelsatzunterdrückung für das gesamte Programm:

```
PROC ... SBLOF
```

Einzelsatzunterdrückung innerhalb des Programms:

```
SBLOF
```

```
...
```

```
SBLON
```

Bedeutung

PROC:	Erste Anweisung eines Programms
SBLOF:	Befehl zum Ausschalten der Einzelsatzbearbeitung SBLOF kann in einem PROC-Satz oder allein im Satz stehen.
SBLON:	Befehl zum Einschalten der Einzelsatzbearbeitung SBLON muss in einem eigenen Satz stehen.

Randbedingungen

- Einzelatzunterdrückung und Satzanzeige**
 Die aktuelle Satzanzeige kann in Zyklen/Unterprogrammen mit `DISPLOF` unterdrückt werden. Wird `DISPLOF` zusammen mit `SBLOF` programmiert, so wird bei Einzelsatz-Stopps innerhalb des Zyklus/Unterprogramms nach wie vor der Aufruf des Zyklus/Unterprogramms angezeigt.
- Einzelatzunterdrückung im System-ASUP oder Anwender-ASUP**
 Wenn der Einzelsatz-Stopp im System- oder Anwender-ASUP über die Einstellungen im Maschinendatum MD10702 `$MN_IGNORE_SINGLEBLOCK_MASK` unterdrückt wird (Bit0 = 1 bzw. Bit1 = 1), dann kann der Einzelsatz-Stopp durch Programmierung von `SBLON` im ASUP wieder aktiviert werden.
 Wird der Einzelsatz-Stopp im Anwender-ASUP über die Einstellung im Maschinendatum MD20117 `$MC_IGNORE_SINGLEBLOCK_ASUP` unterdrückt, dann kann der Einzelsatz-Stopp durch Programmierung von `SBLON` im ASUP **nicht** wieder aktiviert werden.
- Besonderheiten der Einzelatzunterdrückung bei den verschiedenen Einzelatzbearbeitungstypen**
 Bei aktiver Einzelatzbearbeitung SBL2 (Stopp nach jedem Teileprogrammssatz) wird im `SBLON`-Satz **nicht** angehalten, wenn im MD10702 `$MN_IGNORE_SINGLEBLOCK_MASK` (Einzelatzstopp verhindern) Bit 12 auf "1" gesetzt ist.
 Bei aktiver Einzelatzbearbeitung SBL3 (Stopp nach jedem Teileprogrammssatz auch im Zyklus) wird der Befehl `SBLOF` unterdrückt.

Beispiele

Beispiel 1: Einzelatzunterdrückung innerhalb eines Programms

Programmcode	Kommentar
N10 G1 X100 F1000	
N20 SBLOF	; Einzelsatz ausschalten
N30 Y20	
N40 M100	
N50 R10=90	
N60 SBLON	; Einzelsatz wieder einschalten
N70 M110	
N80 ...	

Der Bereich zwischen N20 und N60 wird im Einzelatzbetrieb als ein Schritt bearbeitet.

Beispiel 2: Zyklus soll für den Anwender wie ein Befehl wirken

Hauptprogramm:

Programmcode
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0

Programmcode

N50 M30
Zyklus CYCLE1:

Programmcode	Kommentar
N100 PROC CYCLE1 DISPLOF SBLOF	; Einzelsatz unterdrücken
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

Der Zyklus CYCLE1 wird bei aktiver Einzelsatzbearbeitung abgearbeitet, d. h. es muss für die Bearbeitung von CYCLE1 einmal die Start-Taste gedrückt werden.

Beispiel 3: Ein von der PLC gestartetes ASUP zum Aktivieren von geänderten Nullpunktverschiebung und Werkzeugkorrekturen soll nicht sichtbar sein.

Programmcode

```

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

```

Beispiel 4: Mit MD10702 Bit 12 = 1 wird nicht angehalten

Ausgangssituation:

- Einzelsatzbearbeitung ist aktiv.
- MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

Hauptprogramm:

Programmcode	Kommentar
N10 G0 X0	; In dieser Teileprogrammzeile stoppen.
N20 X10	; In dieser Teileprogrammzeile stoppen.
N30 CYCLE	; Vom Zyklus generierter Verfahrssatz.
N50 G90 X20	; In dieser Teileprogrammzeile stoppen.
M30	

Zyklus CYCLE:

Programmcode	Kommentar
PROC CYCLE SBLOF	; Einzelsatz-Stopp unterdrücken
N100 R0 = 1	
N110 SBLON	; Wegen MD10702 Bit12=1 wird in dieser Teileprogrammzeile nicht gestoppt.
N120 X1	; In dieser Teileprogrammzeile wird gestoppt.
N140 SBLOF	
N150 R0 = 2	
RET	

Beispiel 5: Einzelsatzunterdrückung bei Programmschachtelung

Ausgangssituation:

Einzelsatzbearbeitung ist aktiv.

Programmverschachtelung:

Programmcode	Kommentar
N10 X0 F1000	; In diesem Satz wird gestoppt.
N20 UP1(0)	
PROC UP1(INT _NR) SBLOF	; Einzelsatz-Stopp unterdrücken.
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 SBLON	; Einzelsatz-Stopp einschalten.
N220 X22	; In diesem Satz wird gestoppt.
N230 UP3(0)	
PROC UP3(INT _NR)	
N300 SBLOF	; Einzelsatz-Stopp unterdrücken.
N305 X30	
N310 SBLON	; Einzelsatz-Stopp einschalten.

Programmcode	Kommentar
N320 X32	; In diesem Satz wird gestoppt.
N330 SBLOF	; Einzelsatz-Stopp unterdrücken.
N340 X34	
N350 M17	; SBLOF ist aktiv.
N240 X24	; In diesem Satz wird gestoppt. SBLON ist aktiv.
N250 M17	; In diesem Satz wird gestoppt. SBLON ist aktiv.
N120 X12	
N130 M17	; In diesem Rücksprungssatz wird gestoppt. SBLOF der PROC-Anweisung ist aktiv.
N30 X0	; In diesem Satz wird gestoppt.
N40 M30	; In diesem Satz wird gestoppt.

Weitere Informationen

Einzelsatzsperrung für asynchrone Unterprogramme

Um ein ASUP im Einzelsatz in einem Schritt abzuarbeiten, muss im ASUP eine PROC-Anweisung mit SBLOF programmiert werden. Dies gilt auch für die Funktion "Editierbares System-ASUP" (MD11610 \$MN_ASUP_EDITABLE).

Beispiel für ein editierbares System-ASUP:

Programmcode	Kommentar
N10 PROC ASUP1 SBLOF DISPLOF	
N20 IF \$AC_ASUP=='H200'	
N30 RET	; Kein REPOS bei BA-Wechsel.
N40 ELSE	
N50 REPOSA	; REPOS in allen übrigen Fällen.
N60 ENDIF	

Programmbeeinflussungen im Einzelsatz

In der Einzelsatzbearbeitung kann der Anwender das Teileprogramm satzweise abarbeiten. Es existieren folgende Einstellungsarten:

- SBL1: IPO-Einzelsatz mit Stopp nach jedem Maschinenfunktionssatz.
- SBL2: Einzelsatz mit Stopp nach jedem Satz.
- SBL3: Halt im Zyklus (durch die Anwahl von SBL3 wird der SBLOF-Befehl unterdrückt).

Einzelsatzunterdrückung bei Programmschachtelung

Wurde in einem Unterprogramm SBLOF in der PROC-Anweisung programmiert, so wird auf den Unterprogrammrücksprung mit M17 angehalten. Damit wird verhindert, dass im aufrufenden Programm bereits der nächste Satz ausgeführt wird. Wird in einem Unterprogramm mit SBLOF, ohne SBLOF in der PROC-Anweisung, eine Einzelsatzunterdrückung aktiviert, wird erst nach dem nächsten Maschinenfunktionssatz des

aufzuführenden Programm angehalten. Ist dies nicht erwünscht, muss im Unterprogramm noch vor dem Rücksprung (M17) wieder `SBLON` programmiert werden. Bei einem Rücksprung mit `RET` in ein übergeordnetes Programm wird nicht angehalten.

2.24.2.6 Aktuelle Satzanzeige unterdrücken (DISPLOF, DISPLON, ACTBLOCNO)

In der Satzanzeige wird standardmäßig der aktuelle Programmsatz angezeigt. In Zyklen bzw. Unterprogrammen kann die Anzeige des aktuellen Satzes mit dem Befehl `DISPLOF` unterdrückt werden. Anstelle des aktuellen Satzes wird dann der Aufruf des Zyklus bzw. Unterprogramms angezeigt. Mit dem Befehl `DISPLON` kann die Unterdrückung der Satzanzeige wieder aufgehoben werden.

`DISPLOF` bzw. `DISPLON` wird in der Programmzeile mit der `PROC`-Anweisung programmiert und wirkt für das gesamte Unterprogramm und implizit für alle von diesem Unterprogramm aufgerufenen Unterprogramme, die keinen `DISPLON`- bzw. `DISPLOF`-Befehl enthalten. Dieses Verhalten gilt auch für ASUPs.

Syntax

```
PROC ... DISPLOF
PROC ... DISPLOF ACTBLOCNO
PROC ... DISPLON
```

Bedeutung

DISPLOF:	Befehl zum Unterdrücken der aktuellen Satzanzeige.	
	Platzierung:	Am Ende der Programmzeile mit der <code>PROC</code> -Anweisung
	Wirksamkeit:	Bis zum Rücksprung aus dem Unterprogramm oder Programmende.
	Hinweis: Wenn aus dem Unterprogramm mit dem <code>DISPLOF</code> -Befehl weitere Unterprogramme aufgerufen werden, dann wird auch in diesen Unterprogrammen die aktuelle Satzanzeige unterdrückt, sofern in diesen nicht explizit <code>DISPLON</code> programmiert ist.	
DISPLON:	Befehl zum Aufheben der Unterdrückung der aktuellen Satzanzeige	
	Platzierung:	Am Ende der Programmzeile mit der <code>PROC</code> -Anweisung
	Wirksamkeit:	Bis zum Rücksprung aus dem Unterprogramm oder Programmende.
	Hinweis: Wenn aus dem Unterprogramm mit dem <code>DISPLON</code> -Befehl weitere Unterprogramme aufgerufen werden, dann wird auch in diesen Unterprogrammen der aktuelle Programmsatz angezeigt, sofern in diesen nicht explizit <code>DISPLOF</code> programmiert ist.	
ACTBLOCNO:	<p><code>DISPLOF</code> zusammen mit dem Attribut <code>ACTBLOCNO</code> bewirkt, dass im Falle eines Alarms die Nummer des aktuellen Satzes ausgegeben wird, in dem der Alarm aufgetreten ist. Dies gilt auch dann, wenn in einer niedrigeren Programmebene nur <code>DISPLOF</code> programmiert ist.</p> <p>Bei <code>DISPLOF</code> ohne <code>ACTBLOCNO</code> wird dagegen die Satznummer des Zyklus- bzw. Unterprogrammaufrufs aus der letzten nicht mit <code>DISPLOF</code> gekennzeichneten Programmebene angezeigt.</p>	

Beispiele

Beispiel 1: Aktuelle Satzanzeige im Zyklus unterdrücken

Programmcode	Kommentar
PROC CYCLE(Axis TOMOV, REAL POSITION) SAVE DISPLOF	; Aktuelle Satzanzeige unterdrücken. Stattdessen soll der Zyklus-Aufruf angezeigt werden, z. B.: CYCLE(X,100.0)
DEF REAL DIFF	; Zyklen-Inhalt
G01 RET	; Unterprogramm-Rücksprung. In der Satzanzeige wird der auf den Zyklus-Aufruf folgende Satz angezeigt.

Beispiel 2: Satzanzeige bei der Alarmausgabe

Unterprogramm SUBPROG1 (mit ACTBLOCNO):

Programmcode	Kommentar
PROC SUBPROG1 DISPLOF ACTBLOCNO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	; Alarm 12080 auslösen
...	
N10000 M17	

Unterprogramm SUBPROG2 (ohne ACTBLOCNO):

Programmcode	Kommentar
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; Alarm 12080 auslösen
...	
N7000 M17	

Hauptprogramm:

Programmcode	Kommentar
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; Alarmausgabe = "12080 Kanal K1 Satz N9040 Syntaxfehler bei Text R10="

Programmcode	Kommentar
N2060 ...	
N2350 SUBPROG2	; Alarmausgabe = "12080 Kanal K1 Satz N2350 Syntaxfehler bei Text R10="
...	
N3000 M30	

Beispiel 3: Unterdrückung der aktuellen Satzanzeige aufheben

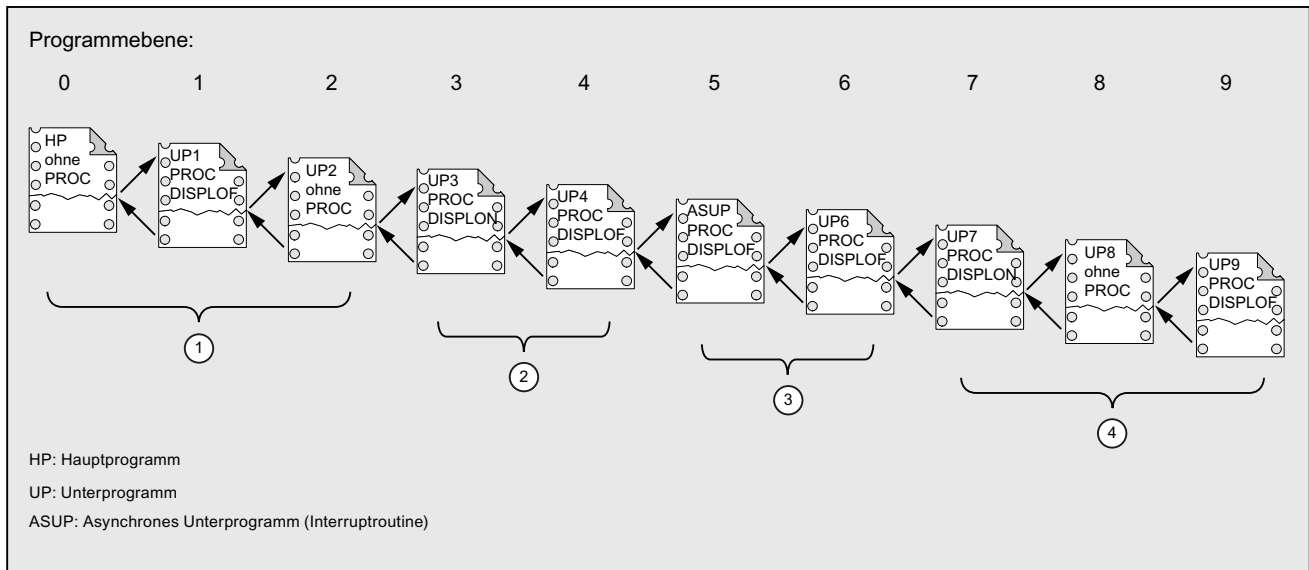
Unterprogramm SUB1 mit Unterdrückung:

Programmcode	Kommentar
PROC SUB1 DISPLOF	; Aktuelle Satzanzeige im Unterprogramm SUB1 unterdrücken. Stattdessen soll der Satz mit dem SUB1-Aufruf angezeigt werden.
...	
N300 SUB2	; Unterprogramm SUB2 aufrufen.
...	
N500 M17	

Unterprogramm SUB2 ohne Unterdrückung:

Programmcode	Kommentar
PROC SUB2 DISPLON	; Unterdrückung der aktuellen Satzanzeige im Unterprogramm SUB2 aufheben.
...	
N200 M17	; Rücksprung ins Unterprogramm SUB1. In SUB1 wird die aktuelle Satzanzeige wieder unterdrückt.

Beispiel 4: Anzeigeverhalten bei unterschiedlichen DISPLON/DISPLOF-Kombinationen



- ① In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 0 angezeigt.
- ② In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 3 angezeigt.
- ③ In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 3 angezeigt.
- ④ In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 7/8 angezeigt.

2.24.2.7 Unterprogramme mit Vorbereitung kennzeichnen (PREPRO)

Mit dem Schlüsselwort `PREPRO` können im Hochlauf am Ende der `PROC`-Anweisungszeile alle Dateien gekennzeichnet werden.

Hinweis

Diese Art der Programmvorbereitung ist vom entsprechend eingestellten Maschinendatum abhängig. Bitte Angaben des Maschinenherstellers beachten.

Literatur:

Funktionshandbuch Sonderfunktionen; Vorverarbeitung (V2)

Syntax

```
PROC ... PREPRO
```

Bedeutung

PREPRO:	Schlüsselwort für Kennzeichnung aller im Hochlauf vorbereiteten Dateien, der in Zyklenverzeichnissen abgelegten NC-Programme
---------	--

Unterprogramme mit Vorbereitung einlesen und Unterprogrammaufruf

Sowohl im Hochlauf vorbereiteter Unterprogramme mit Parametern als auch beim Unterprogrammaufruf werden die Zyklenverzeichnissen in der gleichen Reihenfolge behandelt:

1. `_N_CUS_DIR` Anwenderzyklen
2. `_N_CMA_DIR` Herstellerzyklen
3. `_N_CST_DIR` Standardzyklen

Im Falle gleichnamiger NC-Programme mit unterschiedlicher Ausprägung wird die zuerst gefundene `PROC`-Anweisung aktiviert und die andere `PROC`-Anweisung wird ohne Alarmmeldung überlesen.

2.24.2.8 Unterprogrammrücksprung M17

Am Ende eines Unterprogramms steht der Rücksprung-Befehl `M17` (bzw. der Teileprogrammende-Befehl `M30`). Er bewirkt den Rücksprung in das aufrufende Programm auf den Teileprogrammsatz nach dem Unterprogrammaufruf.

Hinweis

`M17` und `M30` werden in der NC-Sprache gleichwertig behandelt.

Syntax

```
PROC <Programmname>  
...  
M17/M30
```

Randbedingungen

Auswirkung des Unterprogrammrücksprungs auf den Bahnsteuerbetrieb

Steht `M17` (bzw. `M30`) alleine im Teilprogrammsatz, wird dadurch ein im Kanal aktiver Bahnsteuerbetrieb unterbrochen.

Um zu vermeiden, dass der Bahnsteuerbetrieb unterbrochen wird, ist `M17` (bzw. `M30`) mit in den letzten Verfahrssatz zu schreiben. Zusätzlich muss folgendes Maschinendatum auf "0" gesetzt sein:

`MD20800 $MC_SPF_END_TO_VDI = 0` (keine `M30/M17`-Ausgabe an die NC/PLC-Nahtstelle)

Beispiel

1. Unterprogramm mit `M17` im eigenen Satz

Programmcode	Kommentar
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	

Programmcode	Kommentar
N30 M17	; Rücksprung mit Unterbrechung des Bahnsteuerbetriebs.

2. Unterprogramm mit M17 im letzten Verfahrssatz

Programmcode	Kommentar
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; Rücksprung ohne Unterbrechung des Bahnsteuerbetriebs.

2.24.2.9 Unterprogrammrücksprung RET

Als Ersatz für den Rücksprungsbefehl M17 kann im Unterprogramm auch der Befehl `RET` verwendet werden. `RET` muss in einem eigenen Teileprogrammsatz programmiert werden. Wie M17 bewirkt `RET` den Rücksprung in das aufrufende Programm auf den Teileprogrammsatz nach dem Unterprogrammaufruf.

Hinweis

Durch die Programmierung von Parametern kann das Rücksprungverhalten von `RET` geändert werden (siehe "Parametrierbarer Unterprogrammrücksprung (RET ...) (Seite 172)").

Anwendung

Die `RET`-Anweisung ist dann zu benutzen, wenn ein G64-Bahnsteuerbetrieb (G641 ... G645) durch den Rücksprung nicht unterbrochen werden soll.

Voraussetzung

Der Befehl `RET` kann nur in Unterprogrammen verwendet werden, die nicht mit dem Attribut `SAVE` definiert wurden.

Syntax

```
PROC <Programmname>
...
RET
```

Beispiel

Hauptprogramm:

Programmcode	Kommentar
PROC MAIN_PROGRAM	; Programmanfang
...	

Programmcode	Kommentar
N50 SUB_PROG	; Unterprogrammaufruf: SUB_PROG
N60 ...	
...	
N100 M30	; Programmende

Unterprogramm:

Programmcode	Kommentar
PROC SUB_PROG	
...	
N100 RET	; Rücksprung erfolgt auf Satz N60 im Hauptprogramm.

2.24.2.10 Parametrierbarer Unterprogrammrücksprung (RET ...)

Im Allgemeinen wird mit dem Befehl `RET` aus einem Unterprogramm in das aufrufende Programm zurückgesprungen. Die Bearbeitung wird dann mit der auf den Unterprogrammaufruf folgenden Programmzeile fortgesetzt. Soll die Programmbearbeitung an einer anderen Stelle fortgesetzt werden, stehen folgende Möglichkeiten zur Verfügung:

- Fortsetzung der Programmbearbeitung nach Aufruf der Abspanzyklen im ISO-Dialekt-Modus (nach der Konturbeschreibung).
- Rücksprung ins Hauptprogramm aus einer beliebigen Unterprogrammebene (auch nach ASUP) beim Fehlerhandling.
- Rücksprung über mehrere Programmebenen für spezielle Anwendungen in Compilezyklen und im ISO-Dialekt-Modus.

Dazu ist der Befehl `RET` mit weiteren Parametern zu programmieren.

Suchrichtung

Bei Angabe des Parameters `<Zielsatz>` erfolgt der Rücksprung zuerst auf den Satz nach dem Aufrufsatz. Anschließend wird das Ziel in Richtung des Programmendes des Programms gesucht, in das zurückgesprungen wird. Ist die Suche nicht erfolgreich, wird als nächstes in Richtung des Programmanfangs gesucht.

Syntax

```
RET("<Zielsatz>")
RET("<Zielsatz>",<Satz nach Zielsatz>)
RET("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>)
RET("<Zielsatz>",<Anzahl der Rücksprungebenen>)
RET("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>,<Rücksprung auf Programmanfang>)
RET( , ,<Anzahl der Rücksprungebenen>,<Rücksprung auf Programmanfang> )
```

Bedeutung

RET:	Unterprogrammende				
<Zielsatz>:	Im Parameter wird als Sprungziel der Satz angegeben, an dem die Programmbearbeitung fortgesetzt werden soll. Ist der Parameter <Anzahl der Rücksprungebenen> nicht programmiert, dann befindet sich das Sprungziel in dem Programm, aus dem das aktuelle Unterprogramm aufgerufen wurde. Mögliche Angaben sind:				
	<Satznummer>	Nummer des Zielsatzes. Die Suche nach der Satznummer erfolgt im Programm auf das zurückgesprungen wird zuerst in Richtung Programmende.			
	<Sprungmarke>	Sprungmarke, die im Programm in das zurückgesprungen wird, vorhanden sein muss. Die Suche nach der Sprungmarke erfolgt im Programm auf das zurückgesprungen wird zuerst in Richtung Programmende.			
	<Zeichenkette>	Zeichenkette, die im Programm in das zurückgesprungen wird, vorhanden sein muss (z. B. Programm- oder Variablenname). Die Suche nach der Zeichenkette erfolgt im Programm auf das zurückgesprungen wird zuerst in Richtung Programmende. Für die Programmierung der Zeichenkette im Zielsatz gelten folgende Regeln: <ul style="list-style-type: none"> • Leerzeichen am Ende (im Unterschied zur Sprungmarke, die durch einen ":" am Ende gekennzeichnet ist). • Vor der Zeichenkette dürfen nur eine Satznummer und/oder eine Sprungmarke gesetzt sein, keine Programmbeefhle. 			
<Satz nach Zielsatz>:	Im Parameter wird angegeben, ob die Programmbearbeitung im unter Parameter <Zielsatz> angegebenen Satz oder im darauf folgenden Satz fortgeführt werden soll.				
	Typ:	INT			
	Wert:	<table border="1"> <tr> <td>0</td> <td>Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz.</td> </tr> <tr> <td>> 0</td> <td>Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen nachfolgenden Satz.</td> </tr> </table>	0	Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz.	> 0
0	Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz.				
> 0	Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen nachfolgenden Satz.				

<Anzahl der Rücksprungebene- nen>:	Im Parameter wird die Anzahl der Programmebenen angegeben, die zurückzuspringen sind, um dort den Zielsatz zu suchen und die Programmbearbeitung fortzusetzen.		
	Typ:	INT	
	Wert:	1	Das Programm wird in der "aktuellen Programmebene - 1" fortgesetzt (wie RET ohne Parameter).
		2	Das Programm wird in der "aktuellen Programmebene - 2" fortgesetzt, d. h. es wird eine Ebene übersprungen.
		3	Das Programm wird in der "aktuellen Programmebene - 3" fortgesetzt, d. h. es werden zwei Ebenen übersprungen.
...			
Wertebereich:	1 ... 15		
<Rücksprung auf Programmanfang>:	Im Parameter wird angegeben ob bei einem Rücksprung in das Hauptprogramm und dort aktivem ISO-Dialekt-Modus das Programm am Programmanfang fortgesetzt werden soll.		
	Typ:	BOOL	
	Wert:	1	Wenn der Rücksprung ins Hauptprogramm erfolgt und dort ein ISO-Dialekt-Modus aktiv ist, wird auf den Programmanfang verzweigt.

Hinweis

Bei einem Unterprogrammrückprung mit einer Zeichenkette als Angabe für die Zielsatzsuche wird im aufrufenden Programm immer zuerst nach einer Sprungmarke gesucht.

Wenn ein Sprungziel durch eine Zeichenkette eindeutig definiert sein soll, darf die Zeichenkette daher nicht mit dem Namen einer Sprungmarke übereinstimmen, da sonst der Unterprogrammrückprung immer auf die Sprungmarke und nicht auf die Zeichenkette ausgeführt wird (siehe Beispiel 2).

Randbedingungen

Beim Rücksprung über mehrere Programmebenen werden die SAVE-Anweisungen der einzelnen Programmebenen ausgewertet.

Ist bei einem Rücksprung über mehrere Programmebenen ein modales Unterprogramm aktiv und ist in einem der übersprungenen Unterprogramme der Abwahlbefehl MCALL für das modale Unterprogramm programmiert, bleibt das modale Unterprogramm weiterhin aktiv.

<p>ACHTUNG</p> <p>Programmierfehler</p> <p>Bei einem Rücksprung über mehrere Programmebenen liegt es in der alleinigen Verantwortung des Anwenders dafür zu sorgen, dass mit den erforderlichen modalen Einstellungen fortgesetzt wird. Dies kann z. B. durch Programmierung eines entsprechenden Hauptsatzes erreicht.</p>

Beispiele

Beispiel 1: Wiederaufsetzen im Hauptprogramm nach ASUP-Bearbeitung

Programmierung	Kommentar
N10010 CALL "UP1"	; Programmebene 0 (Hauptprogramm)
N11000 PROC UP1	; Programmebene 1
N11010 CALL "UP2"	
N12000 PROC UP2	; Programmebene 2
...	
N19000 PROC ASUP	; Programmebene 3 (ASUP-Bearbeitung)
...	
N19100 RET("N10900", , \$P_STACK)	; Unterprogrammrücksprung ins Hauptprogramm
	; \$P_STACK: aktuelle Programmebene
N10900	; Zielsatz im Hauptprogramm
N10910 MCALL	; Modalen Unterprogrammaufruf ausschalten
N10920 GO G60 G40 M5	; Weitere modale Einstellungen initialisieren

Beispiel 2: Zeichenkette (<String>) als Angabe für die Zielsatzsuche

Hauptprogramm:

Programmcode	Kommentar
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; Aufruf von Unterprogramm "subProg1"
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; Aufruf von Unterprogramm "subProg2"
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; "subProg1" ist hier als Sprungmarke definiert.
N1560 ...	
N1600 subProg3	; Aufruf von Unterprogramm "subProg3"
N1610 ...	
N1900 M30	

Unterprogramm subProg1:

Programmcode	Kommentar
PROC subProg1	

Programmcode	Kommentar
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; Rücksprung ins Hauptprogramm auf den Satz N1400

Unterprogramm subProg2:

Programmcode	Kommentar
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; Rücksprung ins Hauptprogramm auf den Satz N1500

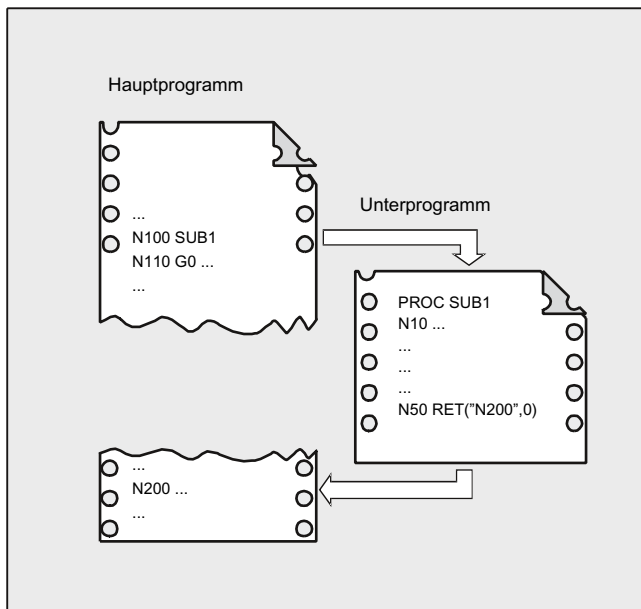
Unterprogramm subProg3:

Programmcode	Kommentar
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; Rücksprung ins Hauptprogramm auf den Satz N1550

Weitere Informationen

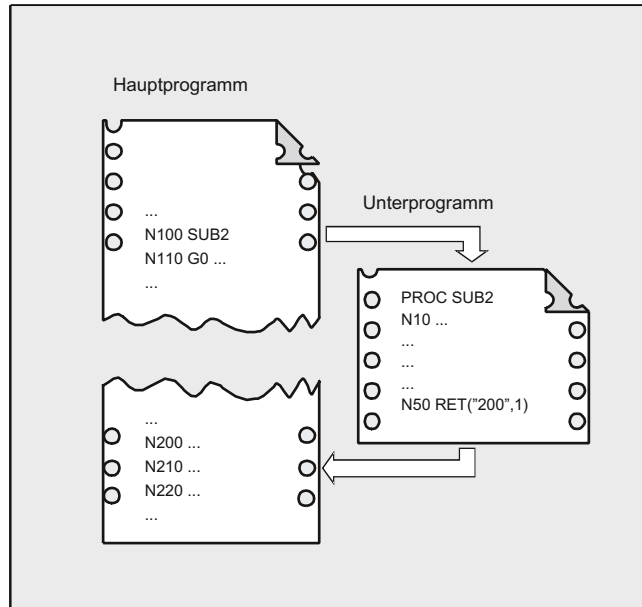
Die folgenden Grafiken zeigen die unterschiedlichen Wirkungen der Rücksprungsparameter

1. <Zielsatz> = "N200", <Satz nach Zielsatz> = 0



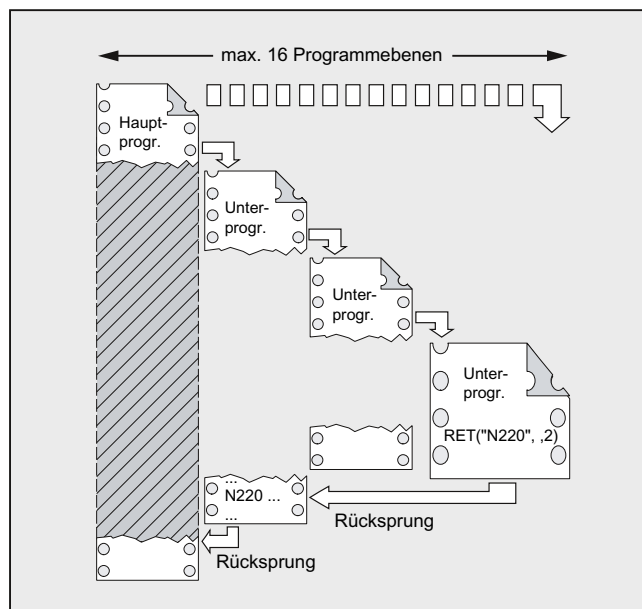
Nach dem `RET`-Befehl wird die Programmbearbeitung mit dem Satz `N200` im Hauptprogramm fortgesetzt.

2. <Zielsatz> = "N200", <Satz nach Zielsatz> = 1



Nach dem `RET`-Befehl wird die Programmbearbeitung mit dem Satz (`N210`) fortgesetzt, der auf den Satz `N200` im Hauptprogramm folgt.

3. <Zielsatz> = "N220", <Anzahl der Rücksprungebenen> = 2



Nach dem `RET`-Befehl wird zwei Programmebenen zurückgesprungen und die Programmbearbeitung wird mit dem Satz `N220` fortgesetzt.

2.24.2.11 Parametrierbarer Unterprogrammrücksprung (RETB ...)

Im Allgemeinen wird mit dem Befehl `RETB` aus einem Unterprogramm in das aufrufende Programm zurückgesprungen. Die Bearbeitung wird dann mit der auf den Unterprogrammaufruf folgenden Programmzeile fortgesetzt. Soll die Programmbearbeitung an einer anderen Stelle fortgesetzt werden, stehen folgende Möglichkeiten zur Verfügung:

- Fortsetzung der Programmbearbeitung nach Aufruf der Abspanzyklen im ISO-Dialekt-Modus (nach der Konturbeschreibung).
- Rücksprung ins Hauptprogramm aus einer beliebigen Unterprogrammebene (auch nach ASUP) beim Fehlerhandling.
- Rücksprung über mehrere Programmebenen für spezielle Anwendungen in Compilezyklen und im ISO-Dialekt-Modus.

Dazu ist der Befehl `RETB` mit weiteren Parametern zu programmieren.

Suchrichtung

Bei Angabe des Parameters `<Zielsatz>` erfolgt der Rücksprung zuerst auf den Satz nach dem Aufrufsatz. Anschließend wird das Ziel in Richtung des Programm**anfangs** des Programms gesucht, in das zurückgesprungen wird. Ist die Suche nicht erfolgreich, wird als nächstes in Richtung des Programm**endes** gesucht.

Syntax

```
RETB("<Zielsatz>")
RETB("<Zielsatz>",<Satz nach Zielsatz>)
RETB("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>)
RETB("<Zielsatz>",<Anzahl der Rücksprungebenen>)
RETB("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>,<Rücksprung auf Programmanfang>)
RETB( , ,<Anzahl der Rücksprungebenen>,<Rücksprung auf Programmanfang>)
```

Bedeutung

RETB:	Unterprogrammende				
<Zielsatz>:	<p>Im Parameter wird als Sprungziel der Satz angegeben, an dem die Programmbearbeitung fortgesetzt werden soll.</p> <p>Ist der Parameter <Anzahl der Rücksprungebenen> nicht programmiert, dann befindet sich das Sprungziel in dem Programm, aus dem das aktuelle Unterprogramm aufgerufen wurde.</p> <p>Mögliche Angaben sind:</p>				
	<Satznummer>	<p>Nummer des Zielsatzes.</p> <p>Die Suche nach der Satznummer erfolgt im Programm auf das zurückgesprungen wird zuerst in Richtung Programmanfang.</p>			
	<Sprungmarke>	<p>Sprungmarke, die im Programm in das zurückgesprungen wird, vorhanden sein muss.</p> <p>Die Suche nach der Sprungmarke erfolgt im Programm auf das zurückgesprungen wird zuerst in Richtung Programmanfang.</p>			
	<Zeichenkette>	<p>Zeichenkette, die im Programm in das zurückgesprungen wird, vorhanden sein muss (z. B. Programm- oder Variablenname).</p> <p>Die Suche nach der Zeichenkette erfolgt im Programm auf das zurückgesprungen wird zuerst in Richtung Programmanfang.</p> <p>Für die Programmierung der Zeichenkette im Zielsatz gelten folgende Regeln:</p> <ul style="list-style-type: none"> • Leerzeichen am Ende (im Unterschied zur Sprungmarke, die durch einen ":" am Ende gekennzeichnet ist). • Vor der Zeichenkette dürfen nur eine Satznummer und/oder eine Sprungmarke gesetzt sein, keine Programmbeefhle. 			
<Satz nach Zielsatz>:	<p>Im Parameter wird angegeben, ob die Programmbearbeitung im unter Parameter <Zielsatz> angegebenen Satz oder im darauf folgenden Satz fortgeführt werden soll.</p>				
	Typ:	INT			
	Wert:	<table border="1"> <tr> <td>0</td> <td>Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz.</td> </tr> <tr> <td>> 0</td> <td>Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen nachfolgenden Satz.</td> </tr> </table>	0	Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz.	> 0
0	Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz.				
> 0	Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen nachfolgenden Satz.				

<Anzahl der Rücksprungebene- nen>:	Im Parameter wird die Anzahl der Programmebenen angegeben, die zurückzuspringen sind, um dort den Zielsatz zu suchen und die Programmbearbeitung fortzusetzen.		
	Typ:	INT	
	Wert:	1	Das Programm wird in der "aktuellen Programmebene - 1" fortgesetzt (wie RET ohne Parameter).
		2	Das Programm wird in der "aktuellen Programmebene - 2" fortgesetzt, d. h. es wird eine Ebene übersprungen.
		3	Das Programm wird in der "aktuellen Programmebene - 3" fortgesetzt, d. h. es werden zwei Ebenen übersprungen.
...			
Wertebereich:	1 ... 15		
<Rücksprung auf Programmanfang>:	Im Parameter wird angegeben ob bei einem Rücksprung in das Hauptprogramm und dort aktivem ISO-Dialekt-Modus das Programm am Programmanfang fortgesetzt werden soll.		
	Typ:	BOOL	
	Wert:	1	Wenn der Rücksprung ins Hauptprogramm erfolgt und dort ein ISO-Dialekt-Modus aktiv ist, wird auf den Programmanfang verzweigt.

Hinweis

Bei einem Unterprogrammrückprung mit einer Zeichenkette als Angabe für die Zielsatzsuche wird im aufrufenden Programm immer zuerst nach einer Sprungmarke gesucht.

Wenn ein Sprungziel durch eine Zeichenkette eindeutig definiert sein soll, darf die Zeichenkette daher nicht mit dem Namen einer Sprungmarke übereinstimmen, da sonst der Unterprogrammrückprung immer auf die Sprungmarke und nicht auf die Zeichenkette ausgeführt wird (siehe Beispiel 2).

Randbedingungen

Beim Rücksprung über mehrere Programmebenen werden die SAVE-Anweisungen der einzelnen Programmebenen ausgewertet.

Ist bei einem Rücksprung über mehrere Programmebenen ein modales Unterprogramm aktiv und ist in einem der übersprungenen Unterprogramme der Abwahlbefehl MCALL für das modale Unterprogramm programmiert, bleibt das modale Unterprogramm weiterhin aktiv.

<p>ACHTUNG</p> <p>Programmierfehler</p> <p>Bei einem Rücksprung über mehrere Programmebenen liegt es in der alleinigen Verantwortung des Anwenders dafür zu sorgen, dass mit den erforderlichen modalen Einstellungen fortgesetzt wird. Dies kann z. B. durch Programmierung eines entsprechenden Hauptsatzes erreicht.</p>

Beispiel

Programmcode	Kommentar
BEISPIEL.MPF	
...	
N3000 START_CYC(param1, param2, ...)	
N3010 TECH_CYC1(param1, param2, ...)	
N3020 TECH_CYC2(param1, param2, ...)	
N3030 TECH_CYC3(param1, param2, ...)	
N3040 END_CYC(param1, param2, ...)	
N3040 END_CYC (param1, param2, ...)	
N3050 ...	
N4500 START_CYC(param11, param12, ...)	
N4510 ...	
N4590 END_CYC(param11, param12, ..)	
N5000 ...	
...	
N6000 M30	

Programmcode	Kommentar
PROC END_CYC(...)	; Aufruf im Hauptprogramm Zeile N3040
N10000 ...	
N15000 if status == 1	
N15010 RETB("START_CYC")	; Rücksprung auf aufrufendes Programm BEISPIEL.MPF ; Suche nach Zeichenkette "START_CYC" ; Suchrichtung: rückwärts Richtung Programmstart ; Fortsetzung der Programmbearbeitung mit Zeile N3000
N15020 endif	
N15030 if status == 0	
N15040 RET	; Rücksprung auf aufrufendes Programm BEISPIEL.MPF ; Fortsetzung der Programmbearbeitung mit Zeile N3050
N15050 endif	
N16000 RET("START_CYC")	; Rücksprung auf aufrufendes Programm BEISPIEL.MPF ; Suche nach Zeichenkette "START_CYC" ; Suchrichtung: vorwärts Richtung Programmende ; Fortsetzung der Programmbearbeitung mit Zeile N4500
N17060 RETB	; Rücksprung auf aufrufendes Programm BEISPIEL.MPF ; Fortsetzung der Programmbearbeitung mit Zeile N3050 ; RETB ohne Parameter ist identisch mit RET

2.24.3 Aufruf eines Unterprogramms

2.24.3.1 Unterprogrammaufruf ohne Parameterübergabe

Der Aufruf eines Unterprogramms erfolgt entweder mit Adresse L und Unterprogrammnummer oder durch Angabe des Programmnamens.

Auch ein Hauptprogramm kann als Unterprogramm aufgerufen werden. Das im Hauptprogramm gesetzte Programmende M2 oder M30 wird in diesem Fall wie M17 (Programmende mit Rücksprung ins aufrufende Programm) gewertet.

Hinweis

Entsprechend kann ein Unterprogramm auch als Hauptprogramm gestartet werden.

Suchstrategie der Steuerung:

Gibt es *_MPF?

Gibt es *_SPF?

Daraus folgt: Falls der Name des aufzurufenden Unterprogramms mit dem Namen des Hauptprogramms identisch ist, dann wird das aufrufende Hauptprogramm wieder aufgerufen. Dieser in der Regel nicht gewünschte Effekt muss durch eindeutige Namenswahl über Unterprogramme und Hauptprogramme vermieden werden.

Hinweis

Unterprogramme, die keine Parameterübergabe erfordern, können auch aus einer Initialisierungsdatei aufgerufen werden.

Syntax

L<Nummer>/<Programmname>

Hinweis

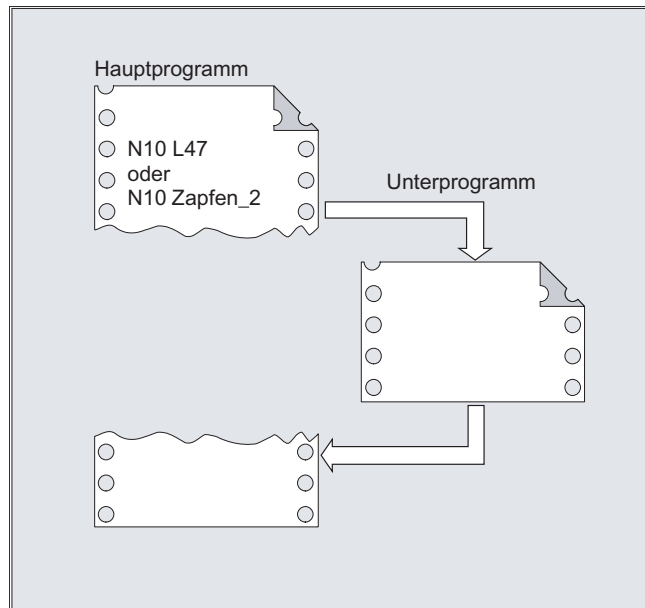
Der Aufruf eines Unterprogramms muss immer im eigenen NC-Satz programmiert werden.

Bedeutung

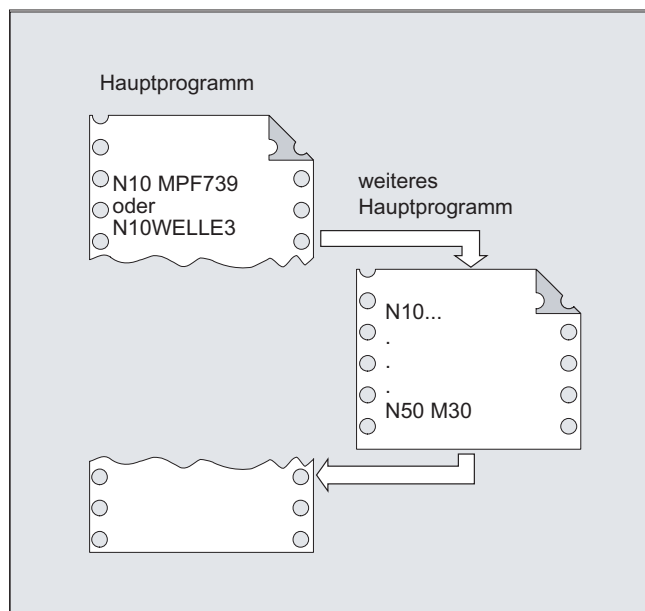
L:	Adresse für den Unterprogrammaufruf	
<Nummer>:	Nummer des Unterprogramms	
	Typ:	INT
	Wert:	Maximal 7 Dezimalstellen Achtung: Führende Nullen sind bei der Namensgebung von Bedeutung (⇒ L123, L0123 und L00123 sind drei verschiedene Unterprogramme).
<Programmname>:	Name des Unterprogramms (oder Hauptprogramms)	

Beispiele

Beispiel 1: Aufruf eines Unterprogramms ohne Parameterübergabe



Beispiel 2: Aufruf eines Hauptprogramms als Unterprogramm



2.24.3.2 Unterprogrammaufruf mit Parameterübergabe (EXTERN)

Beim Unterprogrammaufruf mit Parameterübergabe können Variablen oder Werte direkt übergeben werden (nicht bei `VAR`-Parametern).

Unterprogramme mit Parameterübergabe müssen vor dem Aufruf im Hauptprogramm mit `EXTERN` bekannt gemacht werden (z. B. am Programmanfang). Angegeben werden dabei der Name des Unterprogramms und die Variablentypen in der Reihenfolge der Übergabe.

ACHTUNG

Verwechslungsgefahr

Sowohl die Variablentypen als auch die Reihenfolge der Übergabe muss mit den Definitionen, die im Unterprogramm unter `PROC` vereinbart wurden, übereinstimmen. Die Parameternamen können in Haupt- und Unterprogramm unterschiedlich sein.

Syntax

```
EXTERN <Programmname>(<Typ_Par1>,<Typ_Par2>,<Typ_Par3>)
...
<Programmname>(<Wert_Par1>,<Wert_Par2>,<Wert_Par3>)
```

Hinweis

Der Unterprogrammaufruf muss immer im eigenen NC-Satz programmiert werden.

Bedeutung

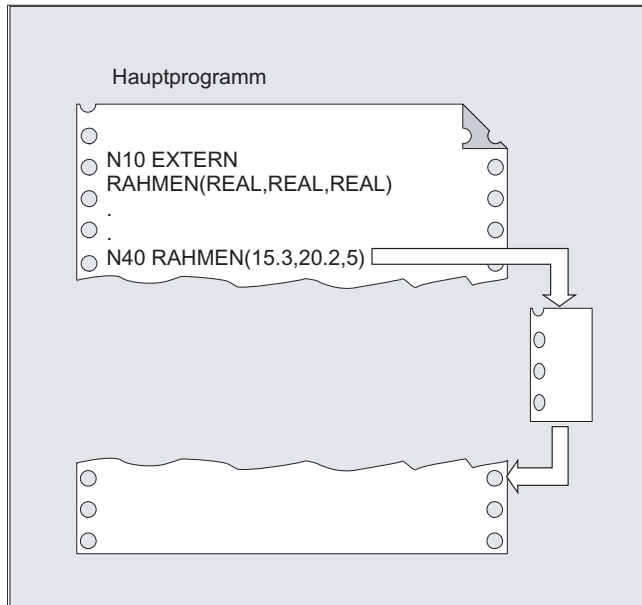
<Programmname>:	Name des Unterprogramms
EXTERN:	Schlüsselwort für die Bekanntmachung eines Unterprogramms mit Parameterübergabe Hinweis: EXTERN muss nur dann angegeben werden, wenn das Unterprogramm im Werkstück- oder im globalen Unterprogrammverzeichnis steht. Zyklen müssen nicht als EXTERN erklärt werden.
<Typ_Par1>,<Typ_Par2>,<Typ_Par3>:	Variablentypen der zu übergebenden Parameter in der Reihenfolge der Übergabe
<Wert_Par1>,<Wert_Par2>,<Wert_Par3>:	Variablenwerte für die zu übergebenden Parameter

Beispiele

Beispiel 1: Unterprogrammaufruf mit vorhergehender Bekanntmachung

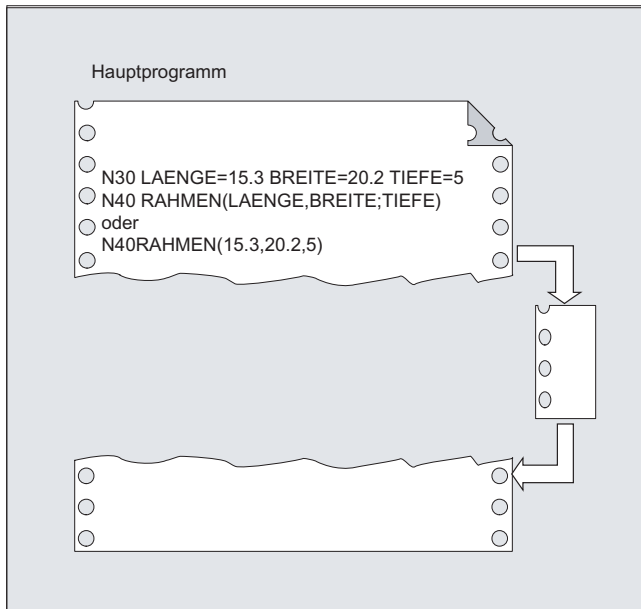
Programmcode	Kommentar
N10 EXTERN RAHMEN (REAL, REAL, REAL)	; Angabe des Unterprogramms.
...	

Programmcode	Kommentar
N40 RAHMEN(15.3,20.2,5)	; Aufruf des Unterprogramms mit Parameterübergabe.




Beispiel 2: Unterprogrammaufruf ohne Bekanntmachung

Programmcode	Kommentar
N10 DEF REAL LAENGE, BREITE, TIEFE	
N20 ...	
N30 LAENGE=15.3 BREITE=20.2 TIEFE=5	
N40 RAHMEN(LAENGE,BREITE,TIEFE)	; oder: N40 RAHMEN(15.3,20.2,5)



2.24.3.3 Anzahl der Programmwiederholungen (P)

Soll ein Unterprogramm mehrfach hintereinander abgearbeitet werden, kann im Satz mit dem Unterprogrammaufruf unter der Adresse P die gewünschte Anzahl der Programmwiederholungen programmiert werden.

	VORSICHT
<p>Unterprogrammaufruf mit Programmwiederholung und Parameterübergabe</p> <p>Parameter werden nur beim Programmaufruf bzw. ersten Durchlauf übergeben. Für die weiteren Wiederholungen bleiben die Parameter unverändert. Falls Sie bei Programmwiederholungen die Parameter verändern wollen, müssen Sie im Unterprogramm entsprechende Vereinbarungen festlegen.</p>	

Syntax

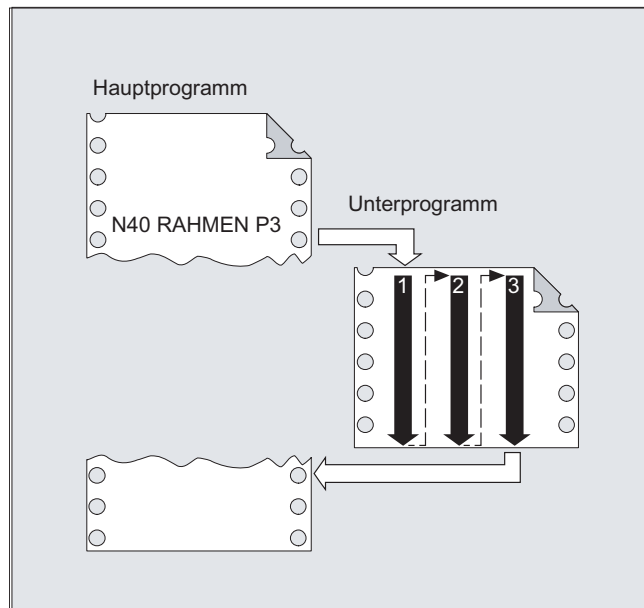
<Programmname> P<Wert>

Bedeutung

<Programmname>:	Unterprogrammaufruf	
P:	Adresse für die Programmierung von Programmwiederholungen	
<Wert>:	Anzahl der Programmwiederholungen	
	Typ:	INT
	Wertebereich:	1 ... 9999 (ohne Vorzeichen)

Beispiel

Programmcode	Kommentar
...	
N40 RAHMEN P3	; Das Unterprogramm RAHMEN soll dreimal hintereinander abgearbeitet werden.
...	



2.24.3.4 Modaler Unterprogrammaufruf (MCALL)

Durch den modalen Unterprogrammaufruf `MCALL(<Programmname>)` wird das angegebene Unterprogramm nicht sofort aufgerufen. Stattdessen erfolgt der Aufruf ab diesem Zeitpunkt im Teileprogramm automatisch nach jedem Verfahrssatz mit Bahnbewegungen. Auch über Programmebenen hinweg.

Hinweis

In einem Programmablauf wirkt immer nur der letzte modale Unterprogrammaufruf `MCALL(<Programmname>)`. Der aktuelle modale Unterprogrammaufruf ersetzt den bisher aktiven.

Werden Parameter an das Unterprogramm übergeben, erfolgt die Parameterübergabe nur beim Aufruf `MCALL(<Programmname>(Par1, Par2, ...))`.

ACHTUNG

Modale Unterprogrammaufrufe ohne Bahnbewegung

Das modale Unterprogramm wird in folgenden Situationen auch ohne Programmierung einer Bahnbewegung aufgerufen:

- Programmierung der Adressen S oder F, wenn G0 oder G1 aktiv ist
- Wenn G0 oder G1 allein im Satz steht oder zusammen mit weiteren G-Codes programmiert wurde.

Syntax

```

MCALL <Programmname>
...
MCALL
    
```

Bedeutung

MCALL <Programmname>:	Funktion "Modaler Unterprogrammaufruf" einschalten
<Programmname>:	Name des Unterprogramms
MCALL:	Mit MCALL ohne Angabe eines Programmnamens wird die Funktion "Modaler Unterprogrammaufruf" ausgeschaltet.

Randbedingungen

ASUP

Wird die Bearbeitung eines Teileprogramms durch ein ASUP (siehe Kapitel "Interruptroutine (ASUP) (Seite 120)") unterbrochen, werden in diesem ASUP keine modalen Unterprogrammaufrufe ausgeführt.

Wird ein ASUP im Kanalzustand "Reset" gestartet, verhält es sich bezüglich der modalen Unterprogrammaufrufe wie ein normales Teileprogramm.

Werkzeugwechsel-Zyklus

Erfolgt im Werkzeugwechsel-Zyklus eine Abwahl der Funktion "Modaler Unterprogrammaufruf", so ist zu beachten, dass der Werkzeugwechsel-Zyklus ggf. auch nach Satzsuchlauf implizit über das Suchlauf-ASUP oder manuell per Überspeichern aufgerufen wird. In dieser Situation darf die Funktion "Modaler Unterprogrammaufruf" nicht abgewählt werden, weil sonst das Suchlauf-Ergebnis verfälscht wird. Es wird deshalb empfohlen, die Abwahl der Funktion "Modaler Unterprogrammaufruf" im Werkzeugwechsel-Zyklus wie folgt zu programmieren:

Programmcode	Kommentar
...	

Programmcode	Kommentar
IF \$AC_ASUP == 0	; Aufruf erfolgt nicht über Suchlauf-ASUP oder Überspeichern.
MCALL	; Funktion "Modaler Unterprogrammaufruf" ausschalten.
ENDIF	
...	

Beispiele

Beispiel 1

Programmcode	Kommentar
N10 G0 X0 Y0	
N20 MCALL L70	; Modalen Unterprogrammaufruf für L70 einschalten.
N30 X10 Y10	; X10 Y10 wird angefahren, anschließend wird L70 aufgerufen.
N40 X20 Y20	; X20 Y20 wird angefahren, anschließend wird L70 aufgerufen.
...	
N100 MCALL	; Funktion "Modaler Unterprogrammaufruf" ausschalten.
N110 X0 Y0	; X0 Y0 wird angefahren, L70 wird nicht aufgerufen.

Beispiel 2

Programmcode
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

In diesem Beispiel stehen die nachfolgenden NC-Sätze mit programmierten Bahnachsen in Unterprogramm L80. L70 wird durch L80 aufgerufen.

2.24.3.5 Indirekter Unterprogrammaufruf (CALL)

In Abhängigkeit von den gegebenen Bedingungen können an einer Stelle unterschiedliche Unterprogramme aufgerufen werden. Hierzu wird der Name des Unterprogramms in einer Variablen vom Typ STRING hinterlegt. Der Unterprogrammaufruf erfolgt mit `CALL` und dem Variablennamen.

Hinweis

Der indirekte Unterprogrammaufruf ist nur für Unterprogramme ohne Parameterübergabe möglich. Für den direkten Aufruf eines Unterprogramms hinterlegen Sie den Namen in einer STRING-Konstanten.

Syntax

```
CALL <Programmname>
```

Bedeutung

CALL:	Befehl für den indirekten Unterprogrammaufruf	
<Programmname>:	Name des Unterprogramms (Variable oder Konstante)	
	Typ:	STRING

Beispiel

Direkter Aufruf mit STRING-Konstante:

Programmcode	Kommentar
...	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Unterprogramm TEIL1 mit CALL direkt aufrufen.
...	

Indirekter Aufruf über Variable:

Programmcode	Kommentar
...	
DEF STRING[100] PROGNAME	; Variable definieren.
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Unterprogramm TEIL1 der Variablen PROGNAME zuordnen.
CALL PROGNAME	; Unterprogramm TEIL1 über CALL und die Variable PROGNAME indirekt aufrufen.
...	

2.24.3.6 Indirekter Unterprogrammaufruf mit Angabe des auszuführenden Programmteils (CALL BLOCK ... TO ...)

Mit CALL und der Schlüsselwortkombination BLOCK ... TO wird ein Unterprogramm indirekt aufgerufen und der mit Start- und Endmarke gekennzeichnete Programmteil ausgeführt.

Syntax

```
CALL <Programmname> BLOCK <Startmarke> TO <Endmarke>
CALL BLOCK <Startmarke> TO <Endmarke>
```

Bedeutung

CALL:	Befehl für den indirekten Unterprogrammaufruf	
<Programmname>:	Name des Unterprogramms (Variable oder Konstante), das den zu bearbeitenden Programmteil enthält (Angabe optional).	
	Typ:	STRING
	Hinweis: Ist kein <Programmname> programmiert, wird der mit <Startmarke> und <Endmarke> gekennzeichnete Programmteil im aktuellen Programm gesucht und ausgeführt.	
BLOCK ... TO ... :	Schlüsselwortkombination für indirekte Programmteilausführung	
<Startmarke>:	Variable, die auf den Beginn des zu bearbeitenden Programmteils verweist.	
	Typ:	STRING
<Endmarke>:	Variable, die auf das Ende des zu bearbeitenden Programmteils verweist.	
	Typ:	STRING

Beispiel

Hauptprogramm:

Programmcode	Kommentar
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; Variablendefinition für die Start- und Endmarke.
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; Indirekter Unterprogrammaufruf und Kennzeichnung des auszuführenden Programmteils.
...	

Unterprogramm:

Programmcode	Kommentar
PROC CONTUR_1 ...	
LABEL_1	; Startmarke: Beginn der Programmteilausführung
N1000 G1 ...	
...	
LABEL_2	; Endmarke: Ende der Programmteilausführung
...	

2.24.3.7 Indirekter Aufruf eines in ISO-Sprache programmierten Programms (ISOCALL)

Mit dem indirekten Programmaufruf `ISOCALL` kann ein in einer ISO-Sprache programmiertes Programm aufgerufen werden. Dabei wird der in den Maschinendaten eingestellte ISO-Modus aktiviert. Am Programmende wird wieder der ursprüngliche Bearbeitungsmodus wirksam. Ist in den Maschinendaten kein ISO-Modus eingestellt, erfolgt der Aufruf des Unterprogramms im Siemens-Modus.

Weitere Informationen zum ISO-Modus siehe:

Literatur:

Funktionsbeschreibung ISO-Dialekte

Syntax

`ISOCALL <Programmname>`

Bedeutung

ISOCALL:	Schlüsselwort für indirekten Unterprogrammaufruf, mit dem der in den Maschinendaten eingestellte ISO-Modus aktiviert wird
<Programmname>:	Name des in einer ISO-Sprache programmierten Programms (Variable oder Konstante vom Typ STRING)

Beispiel: Kontur mit Zyklenprogrammierung aus dem ISO-Modus heraus aufrufen

Programmcode	Kommentar
0122_SPF	; Konturbeschreibung im ISO-Modus
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; Siemens-Teileprogramm (-Zyklus)
...	
N2000 R11 = \$AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; Programm 0122.spf im ISO-Modus bearbeiten
...	
N2400 M30	

2.24.3.8 Unterprogramm mit Pfadangabe und Parametern aufrufen (PCALL)

Mit `PCALL` können Unterprogramme mit absoluter Pfadangabe und Parameterübergabe aufgerufen werden.

Syntax

```
PCALL <Pfad/Programmname>(<Parameter 1>, ..., <Parameter n>)
```

Bedeutung

PCALL:	Schlüsselwort für Unterprogrammaufruf mit absoluter Pfadangabe.
<Pfad/Programmname>:	Absolute Pfadangabe einschließlich Unterprogrammnamen. Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)". Wurde kein absoluter Pfad angegeben, verhält sich PCALL wie ein Standard-Unterprogrammaufruf mit Programmbezeichner. Der Programmname wird ohne Präfix und ohne Datei-Kennung angegeben. Soll der Programmname mit Präfix und Datei-Kennung programmiert werden, so muss er explizit mit Präfix und Datei-Kennung mit dem Befehl EXTERN erklärt werden.
<Parameter 1>, ...:	Aktual-Parameter entsprechend der PROC-Anweisung des Unterprogramms.

Beispiel

Programmcode
PCALL/_N_WKS_DIR/_N_WELLE_WPD/WELLE (parameter1,parameter2,...)

2.24.3.9 Suchpfad bei Unterprogrammaufrufen erweitern (CALLPATH)

Mit dem Befehl CALLPATH kann der Suchpfad für Unterprogrammaufrufe erweitert werden. Damit können auch Unterprogramme aus einem nicht ausgewählten Werkstückverzeichnis aufgerufen werden, ohne den vollständigen, absoluten Pfadnamen des Unterprogramms anzugeben.

Eine weitere Anwendungsmöglichkeit bietet sich im EES-Betriebsmodus "EES ohne GDIR", wenn ein Verzeichnis auf einem externen Programmspeicher zur Ablage globaler Unterprogramme genutzt wird. In diesem Fall kann mit CALLPATH der Suchpfad um dieses Unterprogrammverzeichnis erweitert werden.

Die Suchpfaderweiterung erfolgt vor dem Eintrag für Anwenderzyklen (_N_CUS_DIR).

Durch folgende Ereignisse wird die Suchpfaderweiterung wieder abgewählt:

- CALLPATH mit Leerzeichen
- CALLPATH ohne Parameter
- Teileprogrammende
- Reset

Syntax

```
CALLPATH ("<Pfadname>")
```

Bedeutung

CALLPATH:	Schlüsselwort für die programmierbare Suchpfaderweiterung. Wird in einer eigenen Teileprogrammzeile programmiert.
<Pfadname>:	Konstante oder Variable vom Typ STRING. Enthält die absolute Pfadangabe des Verzeichnisses, um das der Suchpfad erweitert werden soll. Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 209)".

Beispiel

Der Suchpfad soll um ein bestimmtes Werkstückverzeichnis erweitert werden:

```

Programmcode
...
CALLPATH("/_N_WKS_DIR/_N_MYWPD_WPD")
...

```

Damit wird folgender Suchpfad eingestellt (Position 5. ist neu):

1. Aktuelles Verzeichnis/*name*
2. Aktuelles Verzeichnis/*name_SPF*
3. Aktuelles Verzeichnis/*name_MPF*
4. //NC:/_N_SPF_DIR/*name_SPF*
5. /_N_WKS_DIR/_N_MYWPD_WPD/*name_SPF*
6. /N_CUS_DIR/*name_SPF*
7. /_N_CMA_DIR/*name_SPF*
8. /_N_CST_DIR/*name_SPF*

Randbedingungen

- CALLPATH prüft, ob der programmierte Pfadname tatsächlich vorhanden ist. Im Fehlerfall wird die Teileprogrammbearbeitung mit Korrektursatz-Alarm 14009 abgebrochen.
- CALLPATH kann auch in INI-Dateien programmiert werden. Er wirkt dann für die Bearbeitungsdauer der INI-Datei (WPD-INI-Datei oder Initialisierungsprogramm für NC-aktive Daten, z. B. Frames im 1. Kanal _N_CH1_UFR_INI). Danach wird der Suchpfad wieder zurückgesetzt.

2.24.3.10 Externes Unterprogramm abarbeiten (840D sl) (EXTCALL)

Mit dem Befehl `EXTCALL` kann ein Teileprogramm von einem externen Speicher nachgeladen und abgearbeitet werden.

Als externer Speicher stehen zur Verfügung:

- Lokales Laufwerk
- Netzlaufwerk
- USB-Laufwerk

Hinweis

Als Schnittstelle zum Abarbeiten eines auf einem USB-Laufwerk befindlichen externen Programms dürfen nur die USB-Schnittstellen an der Bedientafelfront bzw. TCU verwendet werden.

ACHTUNG**Werkzeug-/Werkstückbeschädigung durch USB-FlashDrive**

Es wird empfohlen zum Abarbeiten eines externen Unterprogramms kein USB-FlashDrive zu verwenden. Ein Kommunikationsabbruch zum USB-FlashDrive während der Abarbeitung des Teileprogramms durch Kontaktschwierigkeiten, Herausfallen, Abbrechen durch Anstoßen oder versehentliches Abziehen führt zum sofortigen Stop der Bearbeitung. Werkzeug und/oder Werkstück könnten dabei beschädigt werden.

Voreinstellung des externen Programmpfades

Der Pfad zum externen Programmverzeichnis kann voreingestellt mit dem Settingdatum:

SD42700 \$SC_EXT_PROG_PATH

Zusammen mit dem beim `EXTCALL`-Aufruf angegebenen Programmpfad und -bezeichner ergibt sich daraus der Gesamtpfad des aufzurufenden Teileprogramms.

Hinweis**Parameter**

Beim Aufruf eines externen Programms können diesem keine Parameter übergeben werden.

Syntax

```
EXTCALL("<Pfad/><Programmname>")
```

Bedeutung

EXTCALL:	Befehl zum Aufrufen eines externen Unterprogramms	
"<Pfad/><Programmname>":	Konstante/Variable vom Typ STRING	
	<Pfad/>:	Absolute oder relative Pfadangabe (optional)
	<Programmname>:	Der Programmname wird ohne Präfix "_N_" angegeben. Die Dateierweiterung ("MPF", "SPF") kann mit dem Zeichen "_" oder "." am Programmnamen angefügt werden (optional). Beispiel: "WELLE" "WELLE_SPF" "WELLE.SPF"

Pfadangabe: Kurzbezeichnungen

Folgende Kurzbezeichnungen können bei der Pfadangabe verwendet werden:

- Lokales Laufwerk: "LOCAL_DRIVE:"
- CF-Karte: "CF_CARD:"
- USB-Laufwerk (Bedientafelfront): "USB:"

Die Kurzbezeichnungen "CF_CARD:" und "LOCAL_DRIVE:" sind alternativ verwendbar.

Beispiel

Abarbeiten von lokalem Laufwerk

Das Hauptprogramm "MAIN.MPF" befindet sich im NC-Speicher und ist zur Abarbeitung ausgewählt.

Unterprogramm "SP_1"

Das externe Unterprogramm "SP_1.SPF" bzw. "SP_1.MPF" befindet sich auf dem lokalen Laufwerk in dem Verzeichnis "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Der Pfad zum externen Programmverzeichnis ist einzustellen mit:

```
SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD
```

Hinweis

Pfadangabe für den Aufruf des externen Unterprogramms:

- Ohne Verwendung der Voreinstellung: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
 - Mit Verwendung der Voreinstellung: "SP_1"
-

Unterprogramm "SP_2"

Das externe Unterprogramm "SP_2.SPF" bzw. "SP_2.MPF" befindet sich im Verzeichnis WKS.DIR/WST1.WPD des USB-Laufwerks. Die Voreinstellung des Pfads zum externen

Programmverzeichnis wird für den Pfad des Unterprogramms "SP_1" verwendet und wird im Hauptprogramm auch nicht umgeschrieben. Daher muss beim Aufruf des Unterprogramms "SP_2" der vollständige Pfad angegeben werden.

Hauptprogramm "MAIN"

Programmcode
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30

Weitere Informationen

EXTCALL-Aufruf mit absoluter Pfadangabe

Ist das Unterprogramm unter dem angegebenen Pfad vorhanden, wird es mit dem EXTCALL-Aufruf ausgeführt. Ist das Unterprogramm unter dem angegebenen Pfad nicht vorhanden, wird die Programmausführung mit dem EXTCALL-Aufruf abgebrochen.

EXTCALL-Aufruf mit relativer Pfadangabe / ohne Pfadangabe

Bei einem EXTCALL-Aufruf mit relativer Pfadangabe bzw. ohne Pfadangabe werden die vorhandenen Programmspeicher nach folgendem Muster durchsucht:

1. Ist in SD42700 \$SC_EXT_PROG_PATH eine Pfadangabe voreingestellt, wird zuerst ausgehend von diesem Pfad nach der Angabe im EXTCALL-Aufruf (Programmname ggf. mit relativer Pfadangabe) gesucht. Der absolute Pfad ergibt sich dann durch Zeichenverkettung aus:
 - Voreingestellten Pfadangabe im SD42700 \$SC_EXT_PROG_PATH
 - Trennzeichen "/"
 - Pfadangabe und Unterprogrammname im Befehl EXTCALL
2. Wurde das Unterprogramm unter 1. nicht gefunden, werden die Verzeichnisse des Anwenderspeichers durchsucht.

Die Suche endet, wenn das Unterprogramm erstmalig gefunden wurde. Wird das Unterprogramm nicht gefunden, wird die Programmausführung mit dem EXTCALL-Aufruf abgebrochen.

Einstellbarer Nachladespeicher (FIFO-Puffer)

Für das Abarbeiten eines externen Unterprogramms wird ein Nachladespeicher benötigt. Die Größe des Nachladespeichers ist mit 30 kByte voreingestellt und kann nur vom Maschinenhersteller verändert werden (über MD18360 MM_EXT_PROG_BUFFER_SIZE).

Hinweis

Unterprogramme mit Sprunganweisungen

Bei externen Unterprogrammen, die Sprunganweisungen enthalten (GOTO, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDF etc.), müssen die Sprungziele innerhalb des Nachladespeichers liegen.

Hinweis

ShopMill-/ShopTurn-Programme

ShopMill- und ShopTurn-Programme müssen wegen der am Dateiende angefügten Konturbeschreibungen vollständig im Nachladespeicher abgelegt sein.

Für parallel abgearbeitete externe Unterprogramme wird jeweils ein eigener Nachladespeicher benötigt.

Reset / Programmende / POWER ON

Durch Reset und POWER ON werden externe Unterprogrammaufrufe abgebrochen und die jeweiligen Nachladespeicher gelöscht.

Ein für "Abarbeiten von Extern" selektiertes Programm bleibt über Reset, Programmende oder POWER ON hinaus weiter für "Abarbeiten von Extern" angewählt. Das Verhalten unterscheidet sich nicht zu intern angewählten Programmen, sofern der externe Programmspeicher weiterhin zur Verfügung steht.

Literatur

Weitere Informationen zu "Abarbeiten von Extern" finden sich in:

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

2.24.3.11 Externes Unterprogramm abarbeiten (828D) (EXTCALL)

Mit dem Befehl `EXTCALL` kann ein Teileprogramm von einem externen Speicher nachgeladen und abgearbeitet werden.

Als externer Speicher stehen zur Verfügung:

- Anwender CF-Karte
- Netzlaufwerk
- USB-Laufwerk

Hinweis

Als Schnittstelle zum Abarbeiten eines auf einem USB-Laufwerk befindlichen externen Programms darf nur die USB-Schnittstelle der Bedientafelfront (PPU) verwendet werden.

ACHTUNG**Werkzeug-/Werkstückbeschädigung durch USB-FlashDrive**

Es wird empfohlen zum Abarbeiten eines externen Unterprogramms kein USB-FlashDrive zu verwenden. Ein Kommunikationsabbruch zum USB-FlashDrive während der Abarbeitung des Teileprogramms durch Kontaktschwierigkeiten, Herausfallen, Abbrechen durch Anstoßen oder versehentliches Abziehen führt zum sofortigen Stop der Bearbeitung. Werkzeug und/oder Werkstück könnten dabei beschädigt werden.

Voreinstellung des externen Programmpfades

Der Pfad zum externen Programmverzeichnis kann voreingestellt mit dem Settingdatum:

SD42700 \$SC_EXT_PROG_PATH

Zusammen mit dem beim `EXTCALL`-Aufruf angegebenen Programmpfad und -bezeichner ergibt sich daraus der Gesamtpfad des aufzurufenden Teileprogramms.

Hinweis**Parameter**

Beim Aufruf eines externen Programms können diesem keine Parameter übergeben werden.

Syntax

```
EXTCALL("<Pfad/><Programmname>")
```

Bedeutung

EXTCALL:	Befehl zum Aufrufen eines externen Unterprogramms	
"<Pfad/><Programmname>":	Konstante / Variable vom Typ STRING	
	<Pfad/>:	Absolute oder relative Pfadangabe (optional)
	<Programmname>:	Der Programmname wird ohne Präfix "_N_" angegeben. Die Dateierweiterung ("MPF", "SPF") kann mit dem Zeichen "_" oder "." am Programmnamen angefügt werden (optional). Beispiel: "WELLE" "WELLE_SPF" "WELLE.SPF"

Pfadangabe: Kurzbezeichnungen

Folgende Kurzbezeichnungen können bei der Pfadangabe verwendet werden:

- Anwender CF-Karte: "CF_CARD:"
- USB-Laufwerk (Bedientafelfront): "USB:"

Beispiel

Das Hauptprogramm "MAIN.MPF" befindet sich im NC-Speicher und ist zur Abarbeitung ausgewählt.

Unterprogramm "SP_1"

Das externe Unterprogramm "SP_1.SPF" bzw. "SP_1.MPF" befindet sich auf der Anwender CF-Karte im Verzeichnis "/WKS.DIR/WST1.WPD".

Der Pfad zum externen Programmverzeichnis ist einzustellen mit:

SD42700 \$SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

Hinweis

Pfadangabe für den Aufruf des externen Unterprogramms:

- Ohne Verwendung der Voreinstellung: "CF_CARD:WKS.DIR/WST1.WPD/SP_1"
 - Mit Verwendung der Voreinstellung: "SP_1"
-

Unterprogramm "SP_2"

Das externe Unterprogramm "SP_2.SPF" bzw. "SP_2.MPF" befindet sich im Verzeichnis WKS.DIR/WST1.WPD des USB-Laufwerks. Die Voreinstellung des Pfads zum externen Programmverzeichnis wird für den Pfad des Unterprogramms "SP_1" verwendet und wird im Hauptprogramm auch nicht umgeschrieben. Daher muss beim Aufruf des Unterprogramms "SP_2" der vollständige Pfad angegeben werden.

Hauptprogramm "MAIN"

Programmcode
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30

Weitere Informationen

EXTCALL-Aufruf mit absoluter Pfadangabe

Ist das Unterprogramm unter dem angegebenen Pfad vorhanden, wird es mit dem EXTCALL-Aufruf ausgeführt. Ist das Unterprogramm unter dem angegebenen Pfad nicht vorhanden, wird die Programmausführung mit dem EXTCALL-Aufruf abgebrochen.

EXTCALL-Aufruf mit relativer Pfadangabe / ohne Pfadangabe

Bei einem EXTCALL-Aufruf mit relativer Pfadangabe bzw. ohne Pfadangabe werden die vorhandenen Programmspeicher nach folgendem Muster durchsucht:

1. Ist in SD42700 \$SC_EXT_PROG_PATH eine Pfadangabe voreingestellt, wird zuerst ausgehend von diesem Pfad nach der Angabe im EXTCALL-Aufruf (Programmname ggf. mit relativer Pfadangabe) gesucht. Der absolute Pfad ergibt sich dann durch Zeichenverkettung aus:
 - Voreingestellten Pfadangabe im SD42700 \$SC_EXT_PROG_PATH
 - Trennzeichen "/"
 - Pfadangabe und Unterprogrammname im Befehl EXTCALL
2. Wurde das Unterprogramm unter 1. nicht gefunden, werden die Verzeichnisse des Anwenderspeichers durchsucht.

Die Suche endet, wenn das Unterprogramm erstmalig gefunden wurde. Wird das Unterprogramm nicht gefunden, wird die Programmausführung mit dem EXTCALL-Aufruf abgebrochen.

Einstellbarer Nachladespeicher (FIFO-Puffer)

Für das Abarbeiten eines externen Unterprogramms wird ein Nachladespeicher benötigt. Die Größe des Nachladespeichers ist voreingestellt (siehe MD18360 MM_EXT_PROG_BUFFER_SIZE).

Hinweis

Unterprogramme mit Sprunganweisungen

Bei externen Unterprogrammen, die Sprunganweisungen enthalten (GOTO, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDF etc.), müssen die Sprungziele innerhalb des Nachladespeichers liegen.

Hinweis

ShopMill-/ShopTurn-Programme

ShopMill- und ShopTurn-Programme müssen wegen der am Dateiende angefügten Konturbeschreibungen vollständig im Nachladespeicher abgelegt sein.

Für parallel abgearbeitete externe Unterprogramme wird jeweils ein eigener Nachladespeicher benötigt.

Reset / Programmende / POWER ON

Durch Reset und POWER ON werden externe Unterprogrammaufrufe abgebrochen und die jeweiligen Nachladespeicher gelöscht.

Ein für "Abarbeiten von Extern" selektiertes Programm bleibt über Reset, Programmende oder POWER ON hinaus weiter für "Abarbeiten von Extern" angewählt. Das Verhalten unterscheidet sich nicht zu intern angewählten Programmen, sofern der externe Programmspeicher weiterhin zur Verfügung steht.

Literatur

Weitere Informationen zu "Abarbeiten von Extern" finden sich in:

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

2.25 Makrotechnik (DEFINE ... AS)

ACHTUNG
Erschwerte Programmierung
Mit Makrotechnik kann die Programmiersprache der Steuerung stark verändert werden. Makrotechnik darf nur mit großer Sorgfalt eingesetzt werden.

Als Makro bezeichnet man die Zusammenfassung von einzelnen Anweisungen zu einer neuen Gesamtanweisung mit eigenem Namen. Auch G-, M- und H-Funktionen oder L-Unterprogrammnamen können als Makros angelegt werden. Bei Aufruf des Makros im Programmablauf werden die unter dem Makronamen programmierten Anweisungen nacheinander abgearbeitet.

Anwendung

Anweisungsfolgen, die sich wiederholen, programmiert man nur einmal als Makro in einem eigenen Makrobaustein (Makrodatei) oder einmal am Programmanfang. Das Makro kann dann in jedem beliebigen Haupt- oder Unterprogramm aufgerufen und abgearbeitet werden.

Aktivierung

Um die Makros einer Makrodatei im NC-Programm verwenden zu können, muss die Makrodatei in die NC geladen werden.

Syntax

Makro-Definition:

```
DEFINE <Makroname> AS <Anweisung 1> <Anweisung 2> ...
```

Aufruf im NC-Programm:

```
<Makroname>
```

Bedeutung

DEFINE ... AS:	Schlüsselwort-Kombination zur Definition eines Makros
<Makroname>:	Name des Makros Als Makronamen sind nur Bezeichner zulässig. Mit dem Makronamen wird das Makro aus dem NC-Programm heraus aufgerufen.
<Anweisung>:	Programmieranweisung, die im Makro enthalten sein soll.

Regeln zur Makro-Definition

- Im Makro können beliebige Bezeichner, G-, M-, H-Funktionen und L-Programmnamen definiert werden.
- Makros können auch im NC-Programm definiert werden.
- G-Funktions-Makros können nur steuerungsglobal im Makrobaustein definiert werden.
- H- und L-Funktionen sind 2-stellig programmierbar.
- M- und G-Funktionen können 3-stellig programmiert werden.

Hinweis

Schlüsselworte und reservierte Namen dürfen nicht mit Makros überdefiniert werden.

Randbedingungen

Eine Schachtelung von Makros ist nicht möglich.

Beispiele

Beispiel 1: Makrodefinition am Programmanfang

Programmcode	Kommentar
DEFINE LINIE AS G1 G94 F300	; Makro-Definition
...	
N70 LINIE X10 Y20	; Makro-Aufruf

Programmcode	Kommentar
...	

Beispiel 2: Makrodefinitionen in einer Makrodatei

Programmcode	Kommentar
DEFINE M6 AS L6	; Beim Werkzeugwechsel wird ein Unterprogramm aufgerufen, das den nötigen Datentransfer übernimmt. Im Unterprogramm wird die eigentliche Werkzeugwechsel-M-Funktion ausgegeben (z. B. M106).
DEFINE G81 AS DRILL(81)	; Nachbildung der DIN-G-Funktion.
DEFINE G33 AS M333 G333	; Beim Gewindeschneiden wird Synchronisation mit der PLC angefordert. Die ursprüngliche G-Funktion G33 wurde per MD in G333 umbenannt, die Programmierung bleibt für den Anwender gleich.

Beispiel 3: Externe Makrodatei

Nach dem Einlesen der externen Makrodatei in die Steuerung muss die Makrodatei in die NC geladen werden. Erst dann können die Makros im NC-Programm verwendet werden.

Programmcode	Kommentar
%_N_UMAC_DEF	
;\$PATH=/_N_DEF_DIR	; Kundenspezifische Makros
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Spindel rechts, Kühlmittel ein
DEFINE M14 AS M4 M7	; Spindel links, Kühlmittel ein
DEFINE M15 AS M5 M9	; Spindel Halt, Kühlmittel aus
DEFINE M6 AS L6	; Aufruf des Werkzeugwechselprogramms
DEFINE G80 AS MCALL	; Abwahl Bohrzyklus
M30	

Datei- und Programmverwaltung

3.1 Programmspeicher

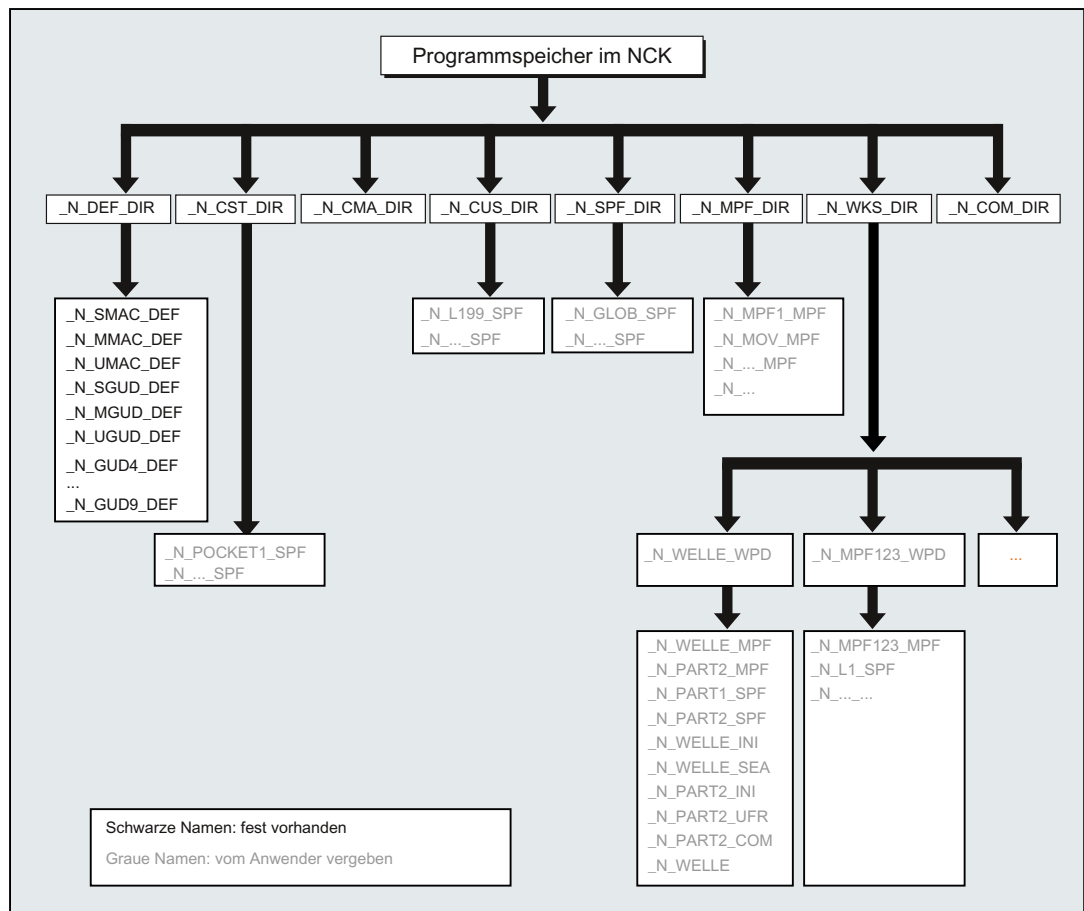
3.1.1 Programmspeicher im NCK

Im Programmspeicher werden Dateien und Programme (z. B. Haupt- und Unterprogramme, Makro-Definitionen) persistent gespeichert (→ Passives Filesystem).

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Speicherkonfiguration (S7)

Daneben gibt es eine Anzahl von Dateitypen, die hier zwischengespeichert werden können und bei Bedarf (z. B. bei Bearbeitung eines bestimmten Werkstücks) in den Arbeitsspeicher zu übertragen sind (z. B. für Initialisierungszwecke).



Standard-Verzeichnisse

Folgende Verzeichnisse sind standardmäßig vorhanden:

Verzeichnis	Inhalt
_N_DEF_DIR	Datenbausteine und Makrobausteine
_N_CST_DIR	Standard-Zyklen
_N_CMA_DIR	Hersteller-Zyklen
_N_CUS_DIR	Anwender-Zyklen
_N_WKS_DIR	Werkstücke
_N_SPF_DIR	Globale Unterprogramme
_N_MPF_DIR	Hauptprogramme
_N_COM_DIR	Kommentare

Dateitypen

Im Programmspeicher können folgende Dateitypen eingebracht werden:

Dateityp	Beschreibung
<Name>_MPF	Hauptprogramm
<Name>_SPF	Unterprogramm
<Name>_TEA	Maschinendaten
<Name>_SEA	Settingdaten
<Name>_TOA	Werkzeugkorrekturen
<Name>_UFR	Nullpunktverschiebungen/Frame
<Name>_INI	Initialisierungsdatei
<Name>_GUD	Globale Anwenderdaten
<Name>_RPA	R-Parameter
<Name>_COM	Kommentar
<Name>_DEF	Definitionen für globale Anwenderdaten und Makros

Werkstück-Hauptverzeichnis (_N_WKS_DIR)

Das Werkstück-Hauptverzeichnis ist standardmäßig unter der Bezeichnung `_N_WKS_DIR` im Programmspeicher eingerichtet. Das Werkstück-Hauptverzeichnis enthält für alle Werkstücke, die Sie programmiert haben, die entsprechenden Werkstückverzeichnisse.

Werkstückverzeichnisse (..._WPD)

Ein Werkstückverzeichnis enthält alle Dateien, die zum Bearbeiten eines Werkstücks notwendig sind. Dies können Hauptprogramme, Unterprogramme, beliebige Initialisierungsprogramme und Kommentar-Dateien sein.

Initialisierungsprogramme werden nach der Programmanwahl mit dem ersten Teileprogrammstart einmalig ausgeführt (entsprechend Maschinendatum MD11280 \$MN_WPD_INI_MODE).

Beispiel:

Das Werkstückverzeichnis `_N_WELLE_WPD`, das für das Werkstück WELLE angelegt wurde, enthält folgende Dateien:

Datei	Beschreibung
<code>_N_WELLE_MPF</code>	Hauptprogramm
<code>_N_PART2_MPF</code>	Hauptprogramm
<code>_N_PART1_SPF</code>	Unterprogramm
<code>_N_PART2_SPF</code>	Unterprogramm
<code>_N_WELLE_INI</code>	Allgemeines Initialisierungsprogramm der Daten für das Werkstück
<code>_N_WELLE_SEA</code>	Initialisierungsprogramm Settingdaten
<code>_N_PART2_INI</code>	Allgemeines Initialisierungsprogramm der Daten für Programm Part 2
<code>_N_PART2_UFR</code>	Initialisierungsprogramm für Frame-Daten für Programm Part 2
<code>_N_WELLE_COM</code>	Kommentardatei

Zusätzlich können in einem Werkstückverzeichnis auch Daten abgelegt werden, welche nicht unmittelbar für die Bearbeitung durch den NCK benötigt werden. Dies können neben ASCII-Dateien auch Binärdateien wie z. B. Bilder im JPG-Format oder Beschreibungen im PDF-Format sein. Damit diese vom NCK als Binärdateien interpretiert werden können, müssen die Dateierweiterungen im NCK bekannt sein (Einstellung bei der Inbetriebnahme über MD17000 `$MN_EXTENSIONS_OF_BIN_FILES`; in der Grundeinstellung sind folgende Dateierweiterungen vorbelegt: JPG, GIF, PNG, BMP, PDF, ICO, HTM).

Werkstück für die Bearbeitung anwählen

Ein Werkstückverzeichnis kann für die Abarbeitung in einem Kanal angewählt werden. Befindet sich in diesem Verzeichnis ein Hauptprogramm **gleichen Namens** oder nur ein einziges Hauptprogramm (`_MPF`), so wird dieses automatisch für die Abarbeitung angewählt.

Literatur:

Bedienhandbuch

3.1.2 Externe Programmspeicher

Neben dem passiven Filesystem im NCK können an einer Maschine auch externe Programmspeicher verfügbar sein (z. B. auf dem lokalen Laufwerk oder auf einem Netzlaufwerk).

Über die Funktionen "Abarbeiten von Extern" oder "EES (Execution from External Storage)" können Teilprogramme **direkt** von externen Programmspeichern abgearbeitet werden.

Literatur:

Funktionshandbuch Grundfunktionen; K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten

Globaler Teileprogrammspeicher (GDIR)

Bei der Vereinbarung der Laufwerke kann eines der Laufwerke als globaler Teileprogrammspeicher (GDIR) ausgezeichnet werden.

Literatur:

Bedienhandbuch; Kapitel: "Programme verwalten" > "Laufwerke einrichten"

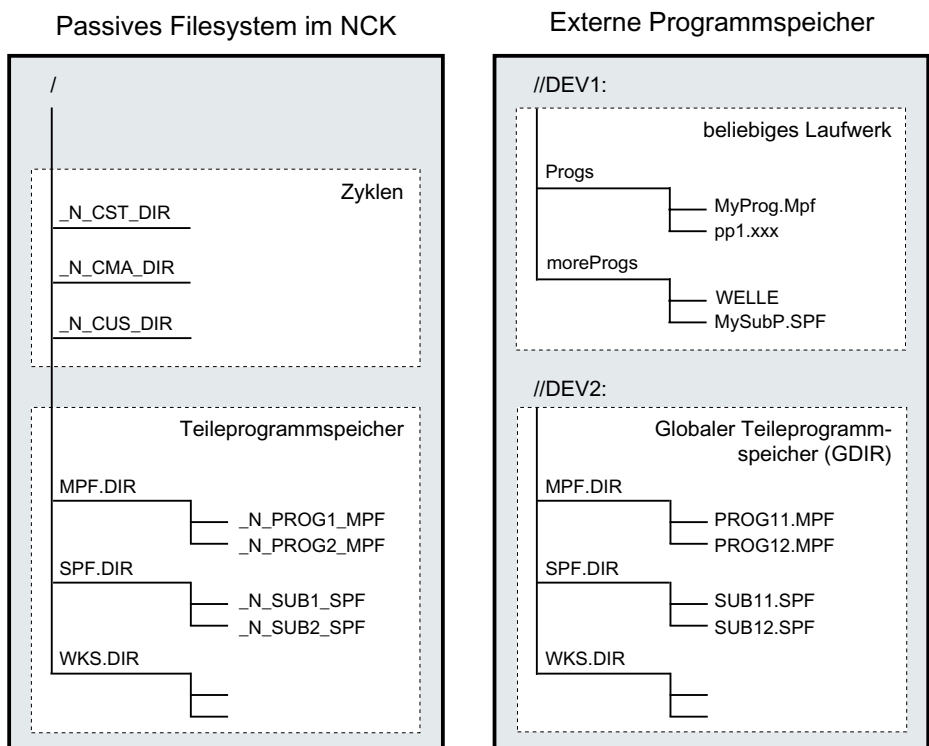
Vom System werden dann automatisch die Verzeichnisse MPF.DIR, SPF.DIR und WKS.DIR auf dem Laufwerk angelegt. Diese drei Verzeichnisse bilden das GDIR.

Das GDIR spielt ausschließlich für die Funktion EES eine Rolle. Je nach Laufwerkskonfiguration ersetzt oder erweitert das GDIR den NC-Teilprogrammspeicher. Das Einrichten eines GDIR ist für den EES-Betrieb allerdings nicht zwingend erforderlich.

Die Verzeichnisse und Dateien des GDIR können im Teilprogramm auf dieselbe Art wie im passiven Filesystem adressiert werden. Damit ist eine kompatible Verlagerung eines NC-Programms mit Pfadangaben aus dem passiven Filesystem ins GDIR möglich. Das Verzeichnis SPF.DIR des GDIR ist im Suchpfad für Unterprogramme enthalten.

Programmorganisation

Die folgende Abbildung soll die Programmorganisation auf externen Programmspeichern veranschaulichen:



Case-insensitive Dateisysteme

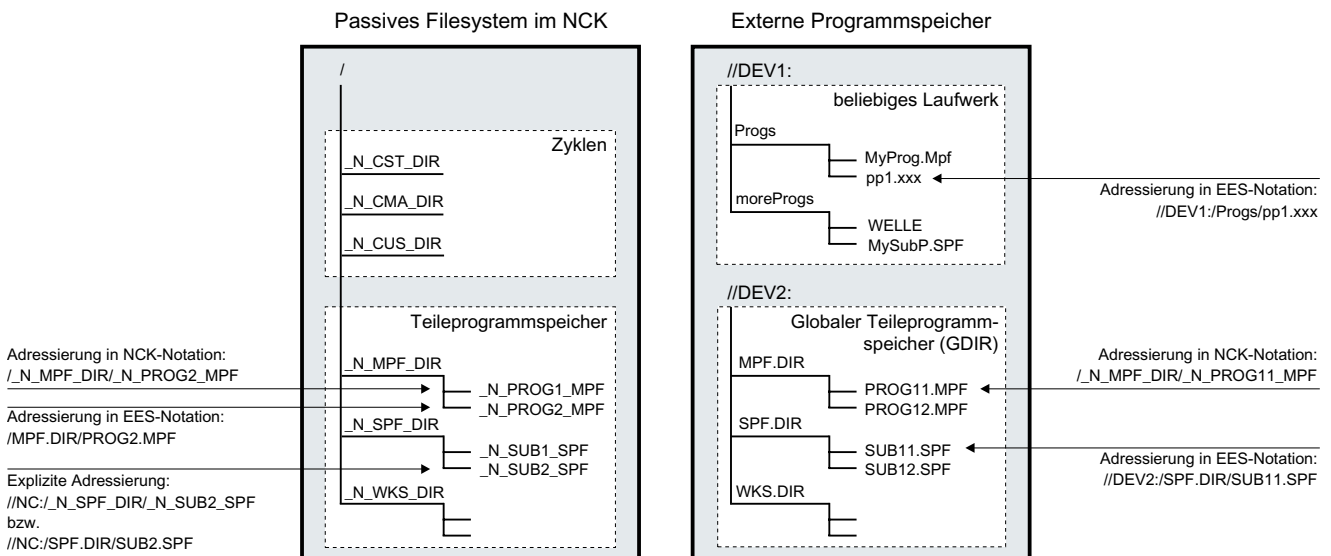
Hinweis

Um Probleme durch die Groß- und Kleinschreibung bei der Datei-Adressierung (siehe "Adressierung von Dateien des Programmspeichers (Seite 209)") zu vermeiden, sollten **case-insensitive** Dateisysteme als externe Programmspeicher verwendet werden.

3.1.3 Adressierung von Dateien des Programmspeichers

Eine Datei im Programmspeicher, die durch einen Dateihandlingsbefehl (z. B. WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) angesprochen wird, wird entweder durch einen absoluten Pfad plus Dateinamen oder nur durch den Dateinamen allein referenziert. Im zweiten Fall wird der Pfad des angewählten Programms als Dateipfad verwendet.

Adressierung in NCK-/EES-Notation



Adressierung von Dateien des passiven Filesystems

Die Adressierung von Dateien des passiven Filesystems erfolgt üblicherweise in **NCK-Notation** (Verzeichnis- und Dateinamen beginnen mit der Domainkennung "_N_", Trennzeichen für die Verzeichnis-/Dateierweiterung ist "_") ohne Angabe des Laufwerknamens. Eine Adressierung in **EES-Notation** (ohne Domain-Kennung "_N_", Trennzeichen für die Verzeichnis-/Dateierweiterung ist ".") ist aber auch zulässig.

Beispiel:

- NCK-Notation: "/_N_SPF_DIR/_N_SUB1_SPF"
- EES-Notation: "/SPF.DIR/SUB1.SPF"

Hinweis

Adressierungen von Dateien des passiven Filesystems in EES-Notation werden intern nach folgenden Regeln in NCK-Notation umgewandelt:

- Verzeichnis- und Dateinamen werden mit der Domainkennung "_N_" erweitert.
 - Ist das viertletzte Zeichen im Verzeichnis- bzw. Dateinamen ein Punkt ("."), wird es in einen Unterstrich ("_") umgewandelt.
-

Über den vordefinierten Laufwerksnamen "//NC:" kann das passive Filesystem auch gezielt adressiert werden.

Beispiel:

- NCK-Notation: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- EES-Notation: "//NC:/SPF.DIR/SUB1.SPF"

Adressierung von Dateien eines externen Programmspeichers

Die Adressierung von Dateien eines externen Programmspeichers, der nicht als GDIR ausgezeichnet ist, muss in EES-Notation erfolgen. Am Anfang des Adressierungspfades muss der Laufwerksname (z. B. "//DEV1:") angegeben werden. Es sind alle in /user/sinumerik/hmi/cfg/logdrive.ini projektierten symbolischen Gerätenamen zulässig.

Beispiel:

- EES-Notation: "//DEV1:/MyProgDir/pp1.xxx"
- NCK-Notation: nicht zulässig

Adressierung von Dateien des globalen Teileprogrammspeichers (GDIR)

Bei der Adressierung von Dateien des GDIR ist neben der Pfadangabe in EES-Notation auch eine Pfadangabe in NCK-Notation zulässig.

Beispiel:

- EES-Notation: "//DEV2:/MPF.DIR/PROG11.MPF"
 - NCK-Notation: "/_N_MPF_DIR/_N_PROG11_MPF"
-

Hinweis

Adressierungen von Dateien des GDIR in NCK-Notation werden intern nach folgenden Regeln in EES-Notation umgewandelt:

- Die Domainkennung "_N_" in Verzeichnis- und Dateinamen wird entfernt.
 - Ist das viertletzte Zeichen im Verzeichnis- bzw. Dateinamen ein Unterstrich ("_"), wird es in einen Punkt (".") umgewandelt.
-

Regeln zur Pfadangabe

Eine vollständige Pfadangabe besteht aus Laufwerksnamen, Verzeichnispfad und Dateinamen.

Laufwerksname

Für die Angabe des Laufwerksnamens gelten folgende Regeln:

- Es sind alle in /user/sinumerik/hmi/cfg/logdrive.ini projektierten symbolischen Gerätenamen zulässig.
- Am Anfang steht das Zeichen "/", gefolgt von mindestens einem Buchstaben oder einer Ziffer.
- Die folgenden Zeichen können in beliebiger Kombination Buchstaben, Ziffern, "_" und Leerzeichen sein.
- Abgeschlossen wird der Name mit einem Buchstaben oder einer Ziffer, gefolgt von einem ":".
- Weitere Sonderzeichen sind nicht erlaubt.

Hinweis

Für das passive Filesystem ist der Laufwerksname "//NC:" vordefiniert.

Beispiele:

- Externe Programmspeicher:
 - //Drive1:
 - //Drive_1:
 - //Drive 1:
 - //A B:
 - //1 B C 2:

Verzeichnispfad

Für die Angabe des Verzeichnispfades gelten folgende Regeln:

- Am Anfang und am Ende des Verzeichnispfades und als Trennzeichen für die einzelnen Pfadanteile steht "/".

Hinweis

Ein doppelter Schrägstrich ("//") innerhalb des Verzeichnispfades ist **nicht** zulässig!

- Verzeichnisnamen:
 - Verzeichnisnamen müssen mit einem Buchstaben oder einer Ziffer beginnen. Nur bei der Adressierung in NCK-Notation beginnen die Verzeichnisnamen mit der Domäinkennung "_N_".
 - Die folgenden Zeichen können in beliebiger Kombination Buchstaben, Ziffern und "_" sein.

Hinweis

Bei externen Programmspeichern sind auch Leerzeichen im Verzeichnisnamen zulässig. Dies gilt jedoch nicht, wenn der externe Programmspeicher als globaler Teileprogrammspeicher (GDIR) eingerichtet ist.

- Weitere Sonderzeichen sind nicht erlaubt.
- Verzeichniserweiterungen:
 - Verzeichniserweiterungen müssen aus genau drei Buchstaben/Ziffern bestehen.
 - Sie werden mit "_" (NCK-Notation) bzw. "." (EES-Notation) vom Verzeichnisnamen getrennt.

Hinweis

Im passiven Filesystem gibt es nur die Verzeichniserweiterungen _DIR und _WPD.

Beispiele:

- Passives Filesystem bzw. GDIR:
 - NCK-Notation: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
 - EES-Notation: WKS.DIR/MYPROGS.WPD/...
- Externe Programmspeicher:
 - /abc
 - /ab_c.def
 - /ab c1.def
 - /a b c .d11
 - /abc.def/ghi.klm

Dateiname

Für Dateinamen gelten folgende Regeln:

- Nur bei der Adressierung in NCK-Notation beginnen die Dateinamen mit der Domäinkennung "_N_".
- Die beiden folgenden Zeichen müssen zwei Buchstaben oder ein Unterstrich und ein Buchstabe sein.
- Die folgenden Zeichen können in beliebiger Kombination Buchstaben, Ziffern und "_" sein.
- Dateierweiterung:
 - Die Dateierweiterung muss aus genau drei Buchstaben/Ziffern bestehen.

Hinweis

Zulässige Dateierweiterungen im passiven Filesystem siehe "Programmspeicher im NCK (Seite 205)".

- Sie wird mit "_" (NCK-Notation) bzw. "." (EES-Notation) vom Dateinamen getrennt.

Beispiele:

- Passives Filesystem bzw. GDIR:
 - NCK-Notation: _N_SUB1_SPF
 - EES-Notation: SUB1.SPF
- Externe Programmspeicher:
 - Teil1
 - _Teil1
 - Teil_1.spf
 - Teil1.mpf

DIN Unterprogrammname

Für DIN Unterprogrammnamen gelten folgende Regeln:

- Das erste Zeichen ist der Buchstabe "L".
- Die folgenden Zeichen sind Ziffern (mindestens eine).
- Dateierweiterung:
 - Die Dateierweiterung muss aus genau drei Buchstaben bestehen.
 - Sie wird mit "_" (NCK-Notation) bzw. "." (EES-Notation) vom Dateinamen getrennt.

Beispiele:

- L123
- L1_SPF (NCK-Notation) bzw. L1.SPF (EES-Notation)

Maximale Pfadlänge

Für die Angabe von Laufwerksname und Verzeichnispfad stehen maximal 128 Bytes zur Verfügung, die Länge des Dateinamens darf maximal 31 Bytes betragen. Für den gesamten Pfad gibt es eine maximale Länge von 159 Bytes.

3.1.4 Suchpfad bei Unterprogrammaufruf

Bei Unterprogrammaufrufen ohne Pfadangabe wird der absolute Pfad durch Abarbeiten eines festen Suchpfades ermittelt.

Die Programmspeicher werden dabei in der folgenden Reihenfolge durchsucht:

	Verzeichnis	Beschreibung
1	aktuelles Verzeichnis / <i>name</i>	Das aktuelle Verzeichnis ist das Verzeichnis, in dem die Programmanwahl erfolgt ist. Dies kann sein: <ul style="list-style-type: none"> ein Werkstückverzeichnis oder das Standard-Verzeichnis <code>_N_MPF_DIR</code> im NCK-Teileprogrammspeicher bzw. globalen Teileprogrammspeicher oder ein beliebiges Verzeichnis eines externen Programmspeichers
2	aktuelles Verzeichnis / <i>name_SPF</i>	
3	aktuelles Verzeichnis / <i>name_MPF</i>	
4	a //NC:/_N_SPF_DIR / <i>name_SPF</i>	Unterprogramm-Verzeichnis im NCK-Teileprogrammspeicher
	b //DEV2:/_N_SPF_DIR / <i>name_SPF</i> ¹⁾	Unterprogramm-Verzeichnis im globalen Teileprogrammspeicher Hinweis: Dieser Suchschritt entfällt, wenn kein globaler Teileprogrammspeicher eingerichtet ist oder die Programmanwahl im NCK-Teileprogrammspeicher erfolgt ist.
5	Mit <code>CALLPATH</code> programmierte Suchpfaderweiterung (siehe "Suchpfad bei Unterprogrammaufrufen erweitern (<code>CALLPATH</code>) (Seite 193)"). Hinweis: Dieser Suchschritt entfällt, wenn kein <code>CALLPATH</code> programmiert ist.	
6	<i>/_N_CUS_DIR / name_SPF</i>	Anwenderzyklen-Verzeichnis
7	<i>/_N_CMA_DIR / name_SPF</i>	Herstellerzyklen-Verzeichnis
8	<i>/_N_CST_DIR / name_SPF</i>	Standardzyklen-Verzeichnis

¹⁾ //DEV2:" steht beispielhaft für das Laufwerk, auf dem der globale Teileprogrammspeicher eingerichtet ist.

Für die Suche gelten folgende Regeln:

- Der Suchpfad wird für jeden einzelnen Unterprogrammaufruf durchlaufen, d. h. es ist irrelevant, wo sich das übergeordnete Programm befindet.
- Je nach Verzeichnis werden unterschiedliche Dateitypen berücksichtigt.
- Grundsätzlich wird in einem Verzeichnis nicht in unterlagerten, d. h. geschachtelten Verzeichnissen gesucht.

3.1.5 Abfrage von Pfad und Dateiname

Zur Abfrage des Pfades und des Dateinamens eines NC-Programms stehen folgende im Teileprogramm lesbare Systemvariablen zur Verfügung:

Systemvariable	Typ	Bedeutung
\$P_STACK	INT	Liefert die Programmebene, in der das aktuelle NC-Programm bearbeitet wird.
\$P_PATH[<n>]	STRING	Liefert den Pfad des NC-Programms, welches in der durch den Feldindex <n> selektierten Programmebene bearbeitet wird. Beispiele: \$P_PATH[0] liefert den Pfad für das Hauptprogramm, z. B. "/_N_WKS_DIR/_N_WELLE_WPD/". \$P_PATH[\$P_STACK - 1] liefert den Pfad des aufrufenden Programms. Bezieht sich der Pfad auf ein NC-Programm, das im passiven Filesystem des NCK oder im globalen Teileprogrammspeicher (GDIR) abgelegt ist, wird der Pfad in NCK-Notation geliefert. Bezieht sich der Pfad auf ein NC-Programm, das von einem anderen externen Programmspeicher als dem globalen Teileprogrammspeicher abgearbeitet wird, liefert \$P_PATH den Pfad in EES-Notation.
\$P_PROG[<n>]	STRING	Liefert den Namen des NC-Programms, welches in der durch den Feldindex <n> selektierten Programmebene bearbeitet wird. Ist das NC-Programm im passiven Filesystem des NCK oder im globalen Teileprogrammspeicher abgelegt, wird der Programmname in NCK-Notation geliefert. Wird das NC-Programm von einem anderen externen Laufwerk als dem globalen Teileprogrammspeicher abgearbeitet, liefert \$P_PROG den Namen in EES-Notation.
\$P_PROGPATH	STRING	Liefert den Pfad des NC-Programms, das gerade bearbeitet wird. Der Aufruf von \$P_PROGPATH ist identisch zu \$P_PATH[\$P_STACK].

Systemvariable	Typ	Bedeutung
\$P_IS_EES_PATH[<n>]	BOOL	Abfrage, ob der von \$P_PATH[<n>] gelieferte Pfad bzw. der von \$P_PROG[<n>] gelieferte Programmname der NCK-Notation oder der EES-Notation entspricht.
		= FALSE \$P_PATH[<n>] und \$P_PROG[<n>] liefern NCK-Notation. D. h. jedem Bezeichner ist eine Präfix "_N_" vorangestellt. Das Trennzeichen für die Dateikennung ist "_". Beispiele: <ul style="list-style-type: none"> • Pfad in NCK-Notation: "/_N_WKS_DIR/_N_MYWPD_WPD/" • Programmname in NCK-Notation: "_N_MYPROG_MPF" Ein Pfad in NCK-Notation kann sich sowohl auf das passive Filesystem im NCK als auch auf den globalen Teileprogrammspeicher beziehen.
		= TRUE \$P_PATH[<n>] und \$P_PROG[<n>] liefern EES-Notation. D. h. den Bezeichnern ist kein Präfix "_N_" vorangestellt. Das Trennzeichen für die Dateikennung ist ".". Beispiele: <ul style="list-style-type: none"> • Pfad in EES-Notation: "//DEV1:/WKS.DIR/MYWPD.WPD/" • Programmname in EES-Notation: "MYPROG.MPF"

<n>: Der Index <n> definiert die Programmebene, aus der die Pfadinformation gelesen werden soll (Wertebereich: 0 ... 17)

Hinweis

Im EES-Betrieb werden außerhalb des globalen Teileprogrammspeichers (GDIR) von den Systemvariablen \$P_PROG, \$P_PATH, und \$P_PROGPATH Pfadnamen in EES-Notation geliefert. Anwenderprogramme, die diese Pfadnamen auswerten und weiterverarbeiten, müssen daher für den EES-Betrieb so erweitert werden, dass sie auch Pfadnamen in EES-Notation verarbeiten können.

3.2 Arbeitsspeicher (CHANDATA, COMPLETE, INITIAL)

Funktion

Der Arbeitsspeicher enthält die aktuellen System- und Anwenderdaten, mit denen die Steuerung betrieben wird (aktives Filesystem), z. B.:

- Aktive Maschinendaten
- Werkzeugkorrekturdaten

- Nullpunktverschiebungen
- ...

Initialisierungsprogramme

Hierbei handelt es sich um Programme, mit denen die Daten des Arbeitsspeichers vorbesetzt (initialisiert) werden. Hierfür können folgende Dateitypen verwendet werden:

Dateityp	Beschreibung
name_TEA	Maschinendaten
name_SEA	Settingdaten
name_TOA	Werkzeugkorrekturen
name_UFR	Nullpunktverschiebungen/Frame
name_INI	Initialisierungsdatei
name_GUD	Globale Anwenderdaten
name_RPA	R-Parameter

Datenbereiche

Die Daten können in unterschiedliche Bereiche eingegliedert werden, in denen sie gelten sollen. Beispielsweise kann eine Steuerung über mehrere Kanäle verfügen oder gewöhnlich auch über mehrere Achsen.

Es gibt:

Kennung	Datenbereiche
NCK	NCK-spezifische Daten
CH<n>	Kanalspezifische Daten (<n> gibt die Kanalnummer an)
AX<n>	Achsspezifische Daten (<n> gibt die Nummer der Maschinenachse an)
TO	Werkzeugdaten
COMPLETE	Alle Daten

Initialisierungsprogramm am externen PC erzeugen

Mit Hilfe von Datenbereichskennung und Datentypenkennung können die Bereiche bestimmt werden, die bei der Datensicherung als Einheit betrachtet werden:

_N_AX5_TEA_INI	Maschinendaten für Achse 5
_N_CH2_UFR_INI	Frames des Kanals 2
_N_COMPLETE_TEA_INI	Alle Maschinendaten

Nach Inbetriebnahme der Steuerung ist ein Datensatz im Arbeitsspeicher vorhanden, der den ordnungsgemäßen Betrieb der Steuerung gewährleistet.

Vorgehensweise bei mehrkanaligen Steuerungen (CHANDATA)

CHANDATA (<Kanalnummer>) für mehrere Kanäle ist nur in der Datei _N_INITIAL_INI zulässig. Das ist die Inbetriebnahmedatei, mit der alle Daten der Steuerung initialisiert werden.

Programmcode	Kommentar
%_N_INITIAL_INI	
CHANDATA (1)	
	; Maschinenachszuordnung Kanal 1:
\$MC_AXCONF_MACHAX_USED[0]=1	
\$MC_AXCONF_MACHAX_USED[1]=2	
\$MC_AXCONF_MACHAX_USED[2]=3	
CHANDATA (2)	
	; Maschinenachszuordnung Kanal 2:
\$MC_AXCONF_MACHAX_USED[0]=4	
\$MC_AXCONF_MACHAX_USED[1]=5	
CHANDATA (1)	
	; Axiale Maschinendaten:
	; Genauhaltfenster grob:
\$MA_STOP_LIMIT_COARSE[AX1]=0.2	; Achse 1
\$MA_STOP_LIMIT_COARSE[AX2]=0.2	; Achse 2
	; Genauhaltfenster fein:
\$MA_STOP_LIMIT_FINE[AX1]=0.01	; Achse 1
\$MA_STOP_LIMIT_FINE[AX1]=0.01	; Achse 2

ACHTUNG

CHANDATA-Anweisung

Im Teileprogramm darf die CHANDATA-Anweisung nur für den Kanal gesetzt werden, auf dem das NC-Programm abgearbeitet wird. D. h. die Anweisung kann dazu benutzt werden, NC-Programme davor zu schützen, dass sie auf einem nicht vorgesehenen Kanal abgearbeitet werden.

Im Fehlerfall wird die Programmabarbeitung abgebrochen.

Hinweis

INI-Dateien in Joblisten enthalten keine CHANDATA-Anweisungen.

Initialisierungsprogramme sichern (COMPLETE, INITIAL)

Die Dateien des Arbeitsspeichers können auf einem externen PC gesichert und von dort wieder eingelesen werden.

- Die Dateien werden mit COMPLETE gesichert.
- Mit INITIAL wird über alle Bereiche eine INI-Datei (_N_INITIAL_INI) erzeugt.

Initialisierungsprogramme einlesen

ACHTUNG

Datenverlust

Wird die Datei mit dem Namen "INITIAL_INI" eingelesen, so werden alle Daten, die in der Datei nicht versorgt werden, mit Standarddaten initialisiert. Ausgenommen davon sind nur die Maschinendaten. Es werden also **Settingdaten, Werkzeugdaten, NPV, GUD-Werte, ...** mit Standarddaten (normalerweise "NULL") versorgt.

Zum Einlesen von einzelnen Maschinendaten eignet sich z. B. die Datei COMPLETE_TEA_INI. In dieser Datei erwartet die Steuerung nur Maschinendaten. Damit bleiben die anderen Datenbereiche in diesem Fall unberührt.

Initialisierungsprogramme laden

Die INI-Programme können auch als Teileprogramme angewählt und aufgerufen werden, wenn sie nur Daten eines Kanals verwenden. So ist es auch möglich, programmgesteuerte Daten zu initialisieren.

Schutzbereiche

4.1 Festlegung der Schutzbereiche (CPROTDEF, NPROTDEF)

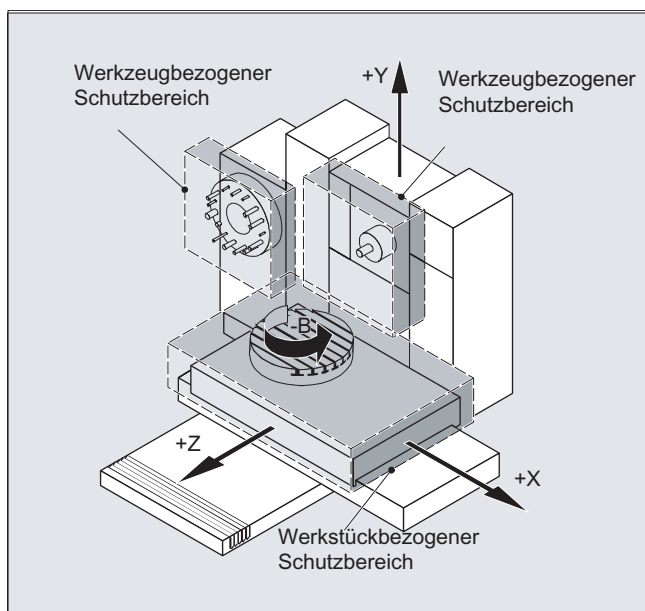
Mit Hilfe von Schutzbereichen lassen sich verschiedene Elemente an der Maschine, die Ausrüstung sowie das Werkstück vor falschen Bewegungen schützen.

Werkzeugbezogener Schutzbereich:

Für Teile, die zum Werkzeug gehören (z. B. Werkzeug, Werkzeugträger).

Werkstückbezogener Schutzbereich:

Für Teile, die zum Werkstück gehören (z. B. Teile des Werkstücks, Aufspanntisch, Spannpratzen, Spindelfutter, Reitstock).



Syntax

```
DEF INT NOT_USED
G17/G18/G19
CPROTDEF/NPROTDEF (<n>, <t>, <applim>, <applus>, <appminus>)
G0/G1/... X/Y/Z...
...
EXECUTE (NOT_USED)
```

Bedeutung

DEF INT NOT_USED:	Lokale Variable vom Datentyp INTEGER definieren
G17/G18/G19:	Die gewünschte Ebene wird vor CPROTDEF bzw. NPROTDEF mit G17/G18/G19 angewählt und darf vor EXECUTE nicht geändert werden. Eine Programmierung der Applikate zwischen CPROTDEF bzw. NPROTDEF und EXECUTE ist nicht zulässig.
CPROTDEF:	Kanalspezifische Schutzbereiche definieren
NPROTDEF:	Maschinenspezifische Schutzbereiche definieren
G0/G1/... X/Y/Z... ...:	Die Kontur der Schutzbereiche wird mit maximal 11 Verfahrbewegungen in der angewählten Ebene angegeben. Dabei ist die erste Verfahrbewegung die Bewegung an die Kontur. Als Schutzbereich gilt dabei der Bereich links von der Kontur. Hinweis: Die zwischen CPROTDEF bzw. NPROTDEF und EXECUTE stehenden Verfahrbewegungen werden nicht ausgeführt, sondern definieren den Schutzbereich.
EXECUTE:	Definition beenden
<n>:	Nummer des definierten Schutzbereichs
<t>:	Typ des Schutzbereichs
	TRUE: Werkzeugbezogener Schutzbereich
	FALSE: Werkstückbezogener Schutzbereich
<aplim>:	Art der Begrenzung in der 3. Dimension
	0: keine Begrenzung
	1: Begrenzung in Plus-Richtung
	2: Begrenzung in Minus-Richtung
	3: Begrenzung in Plus- und Minus-Richtung
<applus>:	Wert der Begrenzung in Plus-Richtung der 3. Dimension
<appminus>:	Wert der Begrenzung in Minus-Richtung der 3. Dimension
NOT_USED:	Die Fehlervariable ist bei Schutzbereichen mit EXECUTE wirkungslos

Randbedingungen

Während der Definition der Schutzbereiche darf:

- keine Fräserradius- bzw. Schneidenradiuskorrektur aktiv sein.
- keine Transformation aktiv sein.
- kein Frame aktiv sein.

Es darf auch nicht Referenzpunktfahren (G74), Festpunktfahren (G75), Satzvorlauf-Stopp oder Programmende programmiert sein.

Weitere Informationen

Definition von Schutzbereichen

Zur Definition von Schutzbereichen gehören:

- CPROTDEF für kanalspezifische Schutzbereiche
- NPROTDEF für maschinenspezifische Schutzbereiche
- Konturbeschreibung des Schutzbereichs
- Abschluss der Definition mit EXECUTE

Bei Aktivierung des Schutzbereichs im NC-Teileprogramm können Sie den Bezugspunkt des Schutzbereichs relativ verschieben.

Bezugspunkt der Konturbeschreibung

Die werkstückbezogenen Schutzbereiche werden im Basiskoordinatensystem definiert.

Die werkzeugbezogenen Schutzbereiche werden bezogen auf den Werkzeugträgerbezugspunkt F angegeben.

Zulässige Konturelemente

Für die Konturbeschreibung des Schutzbereichs sind zulässig:

- G0, G1 für gerade Konturelemente
- G2 für Kreisabschnitte im Uhrzeigersinn (nur für werkstückbezogene Schutzbereiche)
- G3 für Kreisabschnitte gegen den Uhrzeigersinn

Hinweis

Soll ein Vollkreis den Schutzbereich beschreiben, so ist er in zwei Teilkreise aufzuteilen. Die Folge G2, G3 bzw. G3, G2 ist nicht zulässig. Hier ist ggf. ein kurzer G1-Satz einzuschieben.

Der letzte Punkt der Konturbeschreibung muss mit dem ersten Punkt zusammenfallen.

Außenschutzbereiche

Außenschutzbereiche (nur bei werkstückbezogenen Schutzbereichen möglich) sind im Uhrzeigersinn zu definieren.

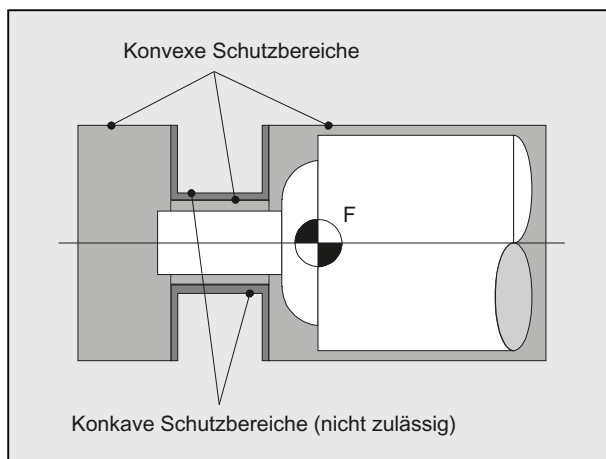
Rotationssymmetrische Schutzbereiche

Bei rotationssymmetrischen Schutzbereichen (z. B. Spindelfutter) muss die Gesamtkontur beschrieben werden (nicht nur bis zur Drehmitte!).

Werkzeugbezogene Schutzbereiche

Werkzeugbezogene Schutzbereiche müssen immer konvex sein. Falls ein konkaver Schutzbereich gewünscht ist, ist dieser in mehrere konvexe Schutzbereiche zu zerlegen.

4.2 Schutzbereiche aktivieren/deaktivieren (CPROT, NPROT)



4.2 Schutzbereiche aktivieren/deaktivieren (CPROT, NPROT)

Vorher definierte Schutzbereiche zur Kollisionsüberwachung aktivieren, voraktivieren oder aktive Schutzbereiche deaktivieren.

Die maximale Anzahl der gleichzeitig in einem Kanal aktiven Schutzbereiche wird über Maschinendatum festgelegt.

Ist kein werkzeugbezogener Schutzbereich aktiv, so wird die Werkzeugbahn gegen die werkstückbezogenen Schutzbereiche geprüft.

Hinweis

Ist kein werkstückbezogener Schutzbereich aktiv, so findet keine Schutzbereichsüberwachung statt.

Syntax

CPROT (<n>, <state>, <xMov>, <yMov>, <zMov>)
 NPROT (<n>, <state>, <xMov>, <yMov>, <zMov>)

Bedeutung

CPROT:	Aufruf kanalspezifischer Schutzbereich	
NPROT:	Aufruf maschinenspezifischer Schutzbereich	
<n>:	Nummer des Schutzbereichs	
<state>:	Statusangabe	
	0:	Schutzbereich deaktivieren
	1:	Schutzbereich voraktivieren
	2:	Schutzbereich aktivieren
3:	Schutzbereich voraktivieren mit bedingtem Stopp	
<xMov>, <yMov>, <zMov>:	Bereits definierten Schutzbereich in den Geometrieachsen verschieben	

Randbedingungen

Schutzbereichsüberwachung bei aktiver Werkzeugradiuskorrektur

Bei aktiver Werkzeugradiuskorrektur ist eine funktionsfähige Schutzbereichsüberwachung nur möglich, wenn die Ebene der Werkzeugradiuskorrektur identisch ist mit der Ebene der Schutzbereichsdefinitionen.

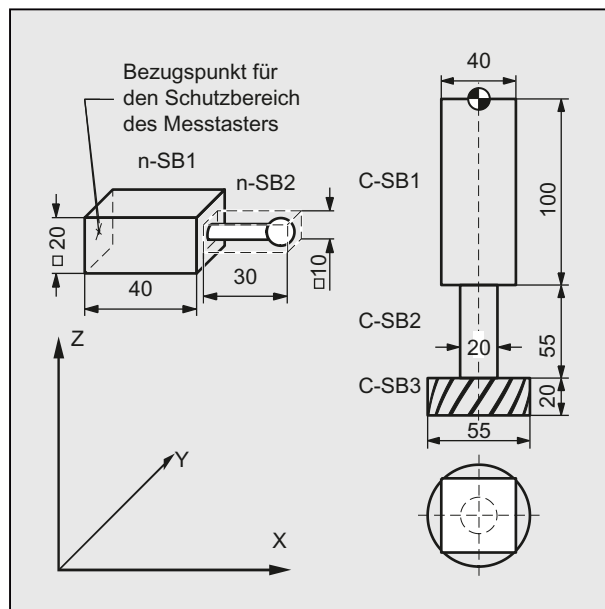
Beispiel

Für eine Fräsmaschine soll eine mögliche Kollision des Fräasers mit dem Messtaster überwacht werden. Die Lage des Messtasters soll bei der Aktivierung durch eine Verschiebung angegeben werden. Es werden dafür folgende Schutzbereiche definiert:

- Jeweils ein maschinenspezifischer und werkstückbezogener Schutzbereich für den Messtasterhalter (n-SB1) und für den Messtaster selbst (n-SB2).
- Jeweils ein kanalspezifischer und werkzeugbezogener Schutzbereich für den Fräserhalter (c-SB1), den Fräferschaft (c-SB2) und für den Fräser selbst (c-SB3).

Die Orientierung aller Schutzbereiche liegt in Z-Richtung.

Die Lage des Bezugspunkts des Messtasters bei der Aktivierung soll bei $X = -120$, $Y = 60$ und $Z = 80$ liegen.



Programmcode	Kommentar
DEF INT SCHUTZB	; Definition einer Hilfsvariablen
Definition der SchutzbereicheG17	; Orientierung einstellen

4.2 Schutzbereiche aktivieren/deaktivieren (CPROT, NPROT)

Programmcode	Kommentar
NPROTDEF (1, FALSE, 3, 10, -10) G01 X0 Y-10 X40 Y10 X0 Y-10 EXECUTE (SCHUTZB)	; Schutzbereich n-SB1
NPROTDEF (2, FALSE, 3, 5, -5) G01 X40 Y-5 X70 Y5 X40 Y-5 EXECUTE (SCHUTZB)	; Schutzbereich n-SB2
CPROTDEF (1, TRUE, 3, 0, -100) G01 X-20 Y-20 X20 Y20 X-20 Y-20 EXECUTE (SCHUTZB)	; Schutzbereich c-SB1
CPROTDEF (2, TRUE, 3, -100, -150) G01 X0 Y-10 G03 X0 Y10 J10 X0 Y-10 J-10 EXECUTE (SCHUTZB)	; Schutzbereich c-SB2
CPROTDEF (3, TRUE, 3, -150, -170) G01 X0 Y-27,5 G03 X0 Y27,5 J27,5 X0 Y27,5 J-27,5 EXECUTE (SCHUTZB)	; Schutzbereich c-SB3
Aktivierung der Schutzbereiche:	
NPROT (1, 2, -120, 60, 80)	; Schutzbereich n-SB1 m. Versch. aktivieren
NPROT (2, 2, -120, 60, 80)	; Schutzbereich n-SB2 m. Versch. aktivieren
CPROT (1, 2, 0, 0, 0)	; Schutzbereich c-SB1 m. Versch. aktivieren
CPROT (2, 2, 0, 0, 0)	; Schutzbereich c-SB2 m. Versch. aktivieren
CPROT (3, 2, 0, 0, 0)	; Schutzbereich c-SB3 m. Versch. aktivieren

Weitere Informationen

Aktivierungsstatus (<state>)

- **<state>=2**
Ein Schutzbereich wird im Allgemeinen im Teileprogramm mit Status = 2 aktiviert. Der Status ist immer kanalspezifisch, auch bei maschinenbezogenen Schutzbereichen.
- **<state>=1**
Wenn durch das PLC-Anwenderprogramm vorgesehen ist, dass ein Schutzbereich durch das PLC-Anwenderprogramm wirksam gesetzt werden kann, so erfolgt die dafür erforderliche Voraktivierung durch den Status = 1.
- **<state>=3**
Bei der Voraktivierung mit bedingtem Stopp wird nicht grundsätzlich vor einem verletzten, voraktivierten Schutzbereich angehalten. Der Stopp erfolgt nur dann, wenn der Schutzbereich wirksam gesetzt worden ist. Dies ermöglicht eine unterbrechungsfreie Bearbeitung, wenn die Schutzbereiche nur in besonderen Fällen wirksam gesetzt werden. Zu beachten ist, dass infolge der Bremsrampe ggf. in einen Schutzbereich gefahren wird, falls der Schutzbereich erst unmittelbar vor dem Einfahren wirksam gesetzt worden ist. Die Voraktivierung mit bedingtem Stopp erfolgt durch den Status = 3.
- **<state>=0**
Die Deaktivierung und damit das Ausschalten der Schutzbereiche erfolgt durch den Status = 0. Es ist dabei keine Verschiebung notwendig.

Verschiebung von Schutzbereichen beim (Vor-)Aktivieren

Die Verschiebung kann in 1, 2 oder 3 Dimensionen erfolgen. Die Angabe der Verschiebung bezieht sich auf:

- den Maschinennullpunkt bei werkstückspezifischen Schutzbereichen.
- den Werkzeugträgerbezugspunkt F bei werkzeugspezifischen Schutzbereichen.

Status nach dem Hochlaufen

Schutzbereiche können bereits nach dem Hochlaufen und anschließendem Referenzpunktanfahren aktiviert sein. Es muss dafür die Systemvariable \$SN_PA_ACTIV_IMMED[<n>] bzw. \$SC_PA_ACTIV_IMMED[<n>] auf TRUE gesetzt sein. Sie werden immer mit dem Status = 2 aktiviert und haben keine Verschiebung.

Mehrfache Aktivierung von Schutzbereichen

Ein Schutzbereich kann gleichzeitig auch in mehreren Kanälen wirksam sein (z. B. Pinole bei zwei gegenüberliegenden Schlitten). Die Überwachung der Schutzbereiche erfolgt nur, wenn alle Geometrieachsen referiert sind.

Dabei gilt:

- Der Schutzbereich ist in einem Kanal nicht gleichzeitig mehrfach mit verschiedenen Verschiebungen aktivierbar.
- Maschinenbezogene Schutzbereiche müssen in beiden Kanälen die gleiche Orientierung aufweisen.

4.3 Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldebegrenzung und Softwareendschalter (CALCPOSI)

Die Funktion `CALCPOSI()` überprüft im Werkstückkoordinatensystem (WKS) bezüglich der **Geometrieachsen**, ob, ausgehend von der Startposition, entlang des Verfahrwegs aktive Begrenzungen verletzt würde.

Kann der Verfahrweg aufgrund von Begrenzungen nicht vollständig abgefahren werden, wird ein positiver, dezimal codierter Statuswert und der maximal mögliche Verfahrweg zurückgegeben.

Syntax

```
<Status> = CALCPOSI (VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <System>, <TestLim>)
```

Bedeutung

CALCPOSI:	Test auf Begrenzungsverletzungen bezüglich der Geometrieachsen.	
	Vorlaufstopp:	nein
	Alleine im Satz:	ja
<Status>: (Teil 1)	Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an.	
	Datentyp:	INT
	Wertebereich:	$-8 \leq x \leq 100000$
	Werte	Bedeutung
	0	Der Verfahrweg kann vollständig abgefahren werden
	Fehlerzustände	
	-1	In <Limit> ist mindestens eine Komponente negativ.
	-2	Fehler bei einer Transformationsberechnung. Beispiel: Der Verfahrweg führt durch eine Singularität, so dass die Achspositionen nicht definiert sind.
	-3	Der angegebene Verfahrweg <Dist> und der maximal mögliche Verfahrweg <MaxDist> sind linear abhängig. Hinweis Kann nur im Zusammenhang mit <TestLim>, Bit 4 == 1 auftreten.
	-4	Die Projektion der in <Dist> enthaltenen Verfahrrichtung auf die Begrenzungsfläche ist der Nullvektor bzw. die Verfahrrichtung steht senkrecht auf der verletzten Begrenzungsfläche. Hinweis Kann nur im Zusammenhang mit <TestLim>, Bit 5 == 1 auftreten.
	-5	In <TestLim> sind Bit 4 == 1 UND Bit 5 == 1
	-6	Mindestens eine Maschinenachse, die für die Überprüfung der Verfahrgrenzen betrachtet werden muß, ist nicht referenziert.
-7	Funktion Kollisionsvermeidung: Ungültige Definition der kinematischen Kette oder der Schutzbereiche.	
-8	Funktion Kollisionsvermeidung: Die Funktion kann wegen Speichermangel nicht ausgeführt werden.	

4.3 Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldebegrenzung und Softwareendschalter (CALCPOSI)

<Status>: (Teil 2)	Einerstelle	
	Hinweis Sind gleichzeitig mehrere Begrenzungen verletzt, wird diejenige gemeldet, die zur stärksten Einschränkung des vorgegebenen Verfahrenweges führt.	
	1	Softwareendschalter begrenzen den Verfahrenweg
	2	Arbeitsfeldebegrenzung begrenzt den Verfahrenweg
	3	Schutzbereiche begrenzen den Verfahrenweg
	4	Funktion Kollisionsvermeidung: Schutzbereiche begrenzen den Verfahrenweg
	Zehnerstelle	
1x	Der Anfangswert verletzt die Begrenzung.	
2x	Die vorgegebene Gerade verletzt die Begrenzung. Dieser Wert wird auch dann zurückgegeben, wenn der Endpunkt selbst keine Begrenzung verletzt, auf dem Weg vom Start-zum Endpunkt aber eine Verletzung eines Grenzwertes auftreten würde (z. B. Durchfahren eines Schutzbereiches, gekrümmte Softwareendschalter im WKS bei nichtlinearen Transformationen, z. B. Transmit).	
<Status>: (Teil 3)	Hunderterstelle	
	1xx	UND Einerstelle == 1 oder 2: Der positive Grenzwert ist verletzt.
		UND Einerstelle == 3 ¹⁾ : Es ist ein NC-spez. Schutzbereich verletzt.
	2xx	UND Einerstelle == 1 oder 2: Der negative Grenzwert ist verletzt.
		UND Einerstelle == 3 ¹⁾ : Es ist ein kanalspez. Schutzbereich verletzt.
<Status>: (Teil 4)	Tausenderstelle	
	1xxx	UND Einerstelle == 1 oder 2: Faktor, mit dem die Achsnmmer multipliziert wird, die die Begrenzung verletzt. Die Zählung der Achsen beginnt bei 1. Bezug: <ul style="list-style-type: none"> • Softwareendschalter: Maschinenachsen • Arbeitsfeldebegrenzung: Geometrieachsen UND Einerstelle == 3 ¹⁾ : Faktor, mit dem die Nummer des verletzten Schutzbereichs multipliziert wird.
<Status>: (Teil 5)	Einhunderttausenderstelle	
	0xxxx	Einhunderttausenderstelle == 0: <Dist> bleibt unverändert
	1xxxx	In <Dist> wird ein Richtungsvektor zurückgegeben, der die weitere Bewegungsrichtung auf der Begrenzungsfläche definiert. Kann nur bei folgenden Randbedingungen auftreten: <ul style="list-style-type: none"> • Softwareendschalter oder Arbeitsfeldebegrenzung verletzt (nicht im Startpunkt) • Eine Transformation ist nicht aktiv • <TestID>, Bit 4 oder Bit 5 == 1
	1) Sind mehrere Schutzbereiche verletzt, wird der Schutzbereich gemeldet, der zur stärksten Einschränkung des vorgegebenen Verfahrenweges führt.	

4.3 Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldebegrenzung und Softwareendschalter (CALCPOSI)

<Start>:	Referenz auf einen Vektor mit den Startpositionen:	
	<ul style="list-style-type: none"> • <Start> [0]: Abszisse • <Start> [1]: Ordinate • <Start> [2]: Applikate 	
	Parametertyp:	Eingang
	Datentyp:	VAR REAL[3]
	Wertebereich:	-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert
<Dist>:	Referenz auf einen Vektor mit dem inkrementellen Verfahrensweg:	
	<ul style="list-style-type: none"> • <Dist> [0]: Abszisse • <Dist> [1]: Ordinate • <Dist> [2]: Applikate 	
	Bei gesetzter Einhunderttausenderstelle im <Status>:	
	<Dist> enthält als Ausgangswert einen Einheitsvektor v , der die weitere Fahr- richtung im WCS definiert.	
	Fall 1: Bildung des Vektors v bei <TestID>, Bit 4 == 1 Die Eingangsvektoren <Dist> und <MaxDist> spannen die Bewegungsebene auf. Diese Ebene wird mit der verletzten Begrenzungsfläche geschnitten. Die Schnittgerade der beiden Ebenen definiert die Richtung des Vektors v . Dabei wird die Orientierung (Vorzeichen) so gewählt, daß der Winkel zwischen dem Eingangsvektor <MaxDist> und v nicht größer als 90 Grad ist.	
	Fall 2: Bildung des Vektors v bei <TestID>, Bit 5 == 1 Der Vektor v ist der Einheitsvektor in Richtung der Projektion des in <Dist> enthaltenen Fahrvektors auf die Begrenzungsfläche. Ist die Projektion des Fahrvektors auf die Begrenzungsfläche der Nullvektor, wird ein Fehler zu- rückgeben.	
Parametertyp:	Ein/Ausgang	
Datentyp:	VAR REAL[3]	
	Wertebereich:	-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert

4.3 Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldebegrenzung und Softwareendschalter (CALCPOSI)

<Limit>:	<p>Referenz auf ein Feld der Länge 5.</p> <ul style="list-style-type: none"> • <Limit> [0 - 2]: Mindestabstand der Geometrieachsen, Abszisse, Ordinate, Applikate Die ersten drei Elemente enthalten die Mindestabstände der Geometrieachse, die gegenüber den überwachten Begrenzungen eingehalten werden müssen. Sie werden bezüglich der Arbeitsfeldebegrenzung immer und bezüglich der Softwareendschalter verwendet, wenn entweder keine Transformation, oder aber eine Transformation aktiv ist, bei der eine eindeutige Zuordnung der Geometrieachsen zu linearen Maschinenachsen möglich ist (z.B. 5-Achs-Transformationen) • <Limit> [3]: Enthält den Mindestabstand für lineare Maschinenachsen, die z.B. auf Grund einer nichtlinearen Transformation keiner Geometrieachse zugeordnet werden können. Dieser Wert wird außerdem als Grenzwert bei der Überwachung der konventionellen Schutzbereiche und der Schutzbereiche der Kollisionsvermeidung verwendet. • <Limit> [4]: Enthält den Mindestabstand für rotatorische Maschinenachsen, die z.B. auf Grund einer nichtlinearen Transformation keiner Geometrieachse zugeordnet werden können . <p>Hinweis Dieser Wert wird nur bei der Überwachung der Softwareendschalter von speziellen Transformationen wirksam.</p> <table border="1" data-bbox="571 938 1481 1049"> <tr> <td>Parametertyp:</td> <td>Eingang</td> </tr> <tr> <td>Datentyp:</td> <td>VAR REAL[5]</td> </tr> <tr> <td>Wertebereich:</td> <td>-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert</td> </tr> </table>	Parametertyp:	Eingang	Datentyp:	VAR REAL[5]	Wertebereich:	-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert
Parametertyp:	Eingang						
Datentyp:	VAR REAL[5]						
Wertebereich:	-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert						
<MaxDist>:	<p>Referenz auf einen Vektor mit dem inkrementellen Fahrweg, bei dem der vorgegebene Mindestabstand von einer Achsgrenze von allen beteiligten Maschinenachsen nicht unterschritten wird:</p> <ul style="list-style-type: none"> • <Dist> [0]: Abszisse • <Dist> [1]: Ordinate • <Dist> [2]: Applikate <p>Ist der Fahrweg nicht eingeschränkt, ist der Inhalt dieses Rückgabeparameters gleich dem Inhalt von <Dist>.</p> <p>Bei <TestID>, Bit 4 == 1: <Dist> und <MaxDist> <MaxDist> und <Dist> müssen als Eingangswerte Vektoren enthalten, die eine Bewegungsebene aufspannt. Die beiden Vektoren müssen voneinander linear unabhängig sein. Der Betrag von <MaxDist> ist beliebig. Zur Berechnung der Bewegungsrichtung siehe die Beschreibung zu <Dist>.</p> <table border="1" data-bbox="571 1502 1481 1610"> <tr> <td>Parametertyp:</td> <td>Ausgang</td> </tr> <tr> <td>Datentyp:</td> <td>VAR REAL[3]</td> </tr> <tr> <td>Wertebereich:</td> <td>-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert</td> </tr> </table>	Parametertyp:	Ausgang	Datentyp:	VAR REAL[3]	Wertebereich:	-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert
Parametertyp:	Ausgang						
Datentyp:	VAR REAL[3]						
Wertebereich:	-max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert						

4.3 Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldbegrenzung und Softwareendschalter (CALCPOSI)

<System>:	Maßsystem (inch / metrisch) für Positions- und Längenangaben (optional)		
	Datentyp:	BOOL	
	Defaultwert:	FALSE	
	Wert	Bedeutung	
	FALSE	Maßsystem entsprechend der aktuell aktiven G-Funktion aus G-Gruppe 13 (G70, G71, G700, G710). Hinweis Bei aktivem G70 und Grundsystem metrische oder aktivem G71 und Grundsystem inch, werden die Systemvariablen \$AA_IW und \$AA_MW im Grundsystem geliefert und müssen bei Verwendung für CALCPOSI () gegebenenfalls umgerechnet werden.	
TRUE	Maßsystem entsprechend des eingestellten Grundsystems: MD52806 \$MN_ISO_SCALING_SYSTEM		
<TestLim>:	Bitcodierte Auswahl der zu überwachenden Begrenzungen (optional)		
	Datentyp:	INT	
	Defaultwert:	Bit 0, 1, 2, 3 == 1 (15)	
	Bit	Dezimal	Bedeutung
	0	1	Softwareendschalter
	1	2	Arbeitsfeldbegrenzung
	2	4	Aktivierte konventionelle Schutzbereiche
	3	8	Voraktivierte konventionelle Schutzbereiche
	4	16	Bei verletzten Softwareendschaltern bzw. Arbeitsfeldbegrenzungen in <Dist> die Verfahrrichtung entsprechend Fall 1 zurückliefern.
	5	32	Bei verletzten Softwareendschaltern bzw. Arbeitsfeldbegrenzungen in <Dist> die Verfahrrichtung entsprechend Fall 2 zurückliefern.
	6	64	Aktivierte Schutzbereiche der Kollisionsvermeidung
	7	128	Voraktivierte Schutzbereiche der Kollisionsvermeidung
8	256	Paare von aktivierten und voraktivierten Schutzbereiche der Kollisionsvermeidung	

Literatur

Funktionshandbuch Grundfunktionen, (A3) Achsüberwachungen, Schutzbereiche, Kapitel "Schutzbereiche"

Spezielle Wegbefehle

5.1 Codierte Positionen anfahren (CAC, CIC, CDC, CACP, CACN)

Über die folgenden Befehle können Sie Linear- und Rundachsen über Positionsnummern auf in Maschinendaten-Tabellen hinterlegte feste Achspositionen verfahren. Diese Art der Programmierung wird als "Anfahren von codierten Positionen" bezeichnet.

Syntax

```
CAC (<n>)
CIC (<n>)
CACP (<n>)
CACN (<n>)
```

Bedeutung

CAC (<n>):	Codierte Position von Positionsnummer n anfahren
CIC (<n>):	Codierte Position, ausgehend von der aktuellen Positionsnummer, n-Positionsplätze vor (+n) oder zurück (-n) anfahren
CDC (<n>):	Codierte Position von Positionsnummer n auf kürzestem Weg anfahren (nur für Rundachsen)
CACP (<n>):	Codierte Position von Positionsnummer n in positiver Richtung anfahren (nur für Rundachsen)
CACN (<n>):	Codierte Position von Positionsnummer n in negativer Richtung anfahren (nur für Rundachsen)
<n>:	Positionsnummer innerhalb der Maschinendaten-Tabelle Wertebereich: 0, 1, ... (max. Anzahl Tabellenplätze - 1)

Beispiel: Anfahren von codierten Positionen einer Positionierachse

Programmiercode	Kommentar
N10 FA[B]=300	; Vorschub für Positionierachse B
N20 POS[B]=CAC(10)	; Codierte Position von Positionsnummer 10 anfahren
N30 POS[B]=CIC(-4)	; Codierte Position von "aktuelle Positionsnummer" - 4 anfahren

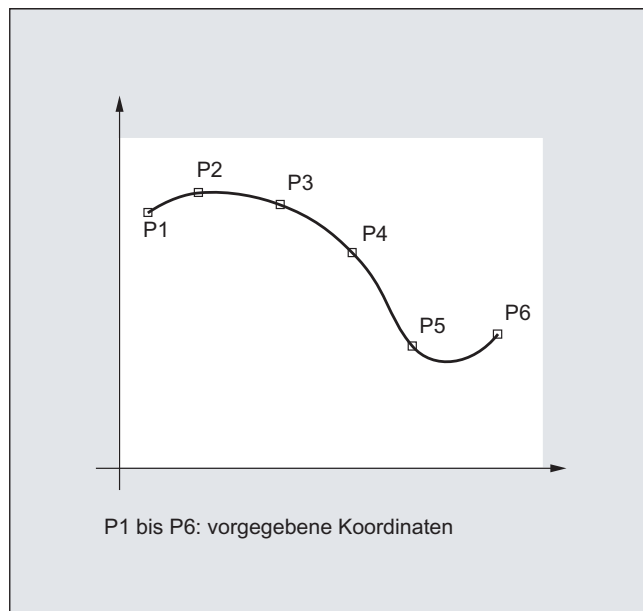
Literatur

- Funktionshandbuch Erweiterungsfunktionen; Teilungsachsen (T1)
- Funktionshandbuch Synchronaktionen

5.2 Spline-Interpolation (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Beliebig gekrümmte Konturen an Werkstücken können nicht analytisch exakt beschrieben werden. Derartige Konturen werden daher durch eine begrenzte Anzahl von Stützpunkten, z. B. beim Digitalisieren von Oberflächen, angenähert. Zur Erzeugung der digitalisierten Oberfläche an einem Werkstück müssen die Stützpunkte zu einer Konturbeschreibung verbunden werden. Dies ermöglicht die Spline-Interpolation.

Ein Spline definiert eine Kurve, die aus Polynomen 2. oder 3. Grades zusammengesetzt wird. Die Eigenschaften an den Stützpunkten eines Splines sind **abhängig vom verwendeten Spline-Typ** definierbar.



Folgende Spline-Typen stehen bei SINUMERIK solution line zur Verfügung:

- A-Spline
- B-Spline
- C-Spline

Syntax

Allgemein:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

Bei B-Spline zusätzlich programmierbar:

```
PW=<n>
SD=2
PL=<Wert>
```

Bei A- und C-Spline zusätzlich programmierbar:

```
BAUTO / BNAT / BTAN
```

5.2 Spline-Interpolation (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

EAUTO / ENAT / ETAN

Bedeutung

Spline-Interpolationstyp:					
ASPLINE:	Befehl zum Einschalten der A-Spline-Interpolation				
BSPLINE:	Befehl zum Einschalten der B-Spline-Interpolation				
CSPLINE:	Befehl zum Einschalten der C-Spline-Interpolation				
	Die Befehle ASPLINE, BSPLINE und CSPLINE sind modal wirksam und gehören zur Gruppe der Wegbefehle.				
Stützpunkte bzw. Kontrollpunkte:					
X... Y... Z... A... B... C...	Positionen in kartesischen Koordinaten				
Punktgewicht (nur B-Spline):					
PW:	Mit dem Befehl PW ist für jeden Stützpunkt die Programmierung eines sogenannten "Punktgewichts" möglich.				
<n>:	"Punktgewicht"				
	Wertebereich:	$0 \leq n \leq 3$			
	Schrittweite:	0.0001			
	Wirkung:	<table border="1"> <tr> <td>$n > 1$</td> <td>Die Kurve wird vom Kontrollpunkt stärker angezogen.</td> </tr> <tr> <td>$n < 1$</td> <td>Die Kurve wird vom Kontrollpunkt weniger stark angezogen.</td> </tr> </table>	$n > 1$	Die Kurve wird vom Kontrollpunkt stärker angezogen.	$n < 1$
$n > 1$	Die Kurve wird vom Kontrollpunkt stärker angezogen.				
$n < 1$	Die Kurve wird vom Kontrollpunkt weniger stark angezogen.				
Spline-Grad (nur B-Spline):					
SD:	Standardmäßig wird ein Polygon 3. Grades verwendet. Durch Programmierung von SD=2 kann aber auch ein Polygon 2. Grades verwendet werden.				
Knotenabstand (nur B-Spline):					
PL:	Die Knotenabstände werden intern geeignet berechnet. Die Steuerung kann aber auch vorgegebene Knotenabstände verarbeiten, die mit dem Befehl PL als sog. Parameter-Intervall-Länge angegeben werden.				
<Wert>:	Parameter-Intervall-Länge				
	Wertebereich:	wie Wegmaß			
Übergangsverhalten am Beginn der Spline-Kurve (nur A- oder C-Spline):					
BAUTO:	Keine Vorgabe für das Übergangsverhalten. Der Anfang ergibt sich aus der Lage des ersten Punkts.				
BNAT:	Krümmung Null				
BTAN:	Tangentialer Übergang zum Satz vorher (Löschstellung)				
Übergangsverhalten am Ende der Spline-Kurve (nur A- oder C-Spline):					
EAUTO:	Keine Vorgabe für das Übergangsverhalten. Das Ende ergibt sich aus der Lage des letzten Punkts.				

ENAT:	Krümmung Null
ETAN:	Tangentialer Übergang zum Satz vorher (Löschstellung)
<p>Das Diagramm zeigt drei Beispiele für Spline-Interpolationen an einem Kontrollpolygonpunkt (Knoten). 1. BAUTO/EAUTO: Die Spline-Kurve beginnt am Knoten ohne feste Tangentialrichtung. Beschriftung: "keine Vorgabe". 2. BNAT/ENAT: Die Spline-Kurve beginnt am Knoten mit einer Krümmung Null. Beschriftung: "Krümmung Null". 3. BTAN/ETAN: Die Spline-Kurve beginnt am Knoten mit einem tangentialen Übergang von der vorherigen Kurve. Beschriftung: "Übergang tangential".</p>	

Hinweis

Das programmierbare Übergangsverhalten hat keinen Einfluss auf den B-Spline. Der B-Spline ist in Start- und Endpunkt immer tangential zum Kontrollpolygon.

Randbedingungen

- Die Werkzeugradiuskorrektur ist einsetzbar.
- Kollisionsüberwachung erfolgt in der Projektion auf die Ebene.

Beispiele

Beispiel 1: B-Spline

Programmcode 1 (alle Gewichte 1)

```

N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
    
```

5.2 Spline-Interpolation (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

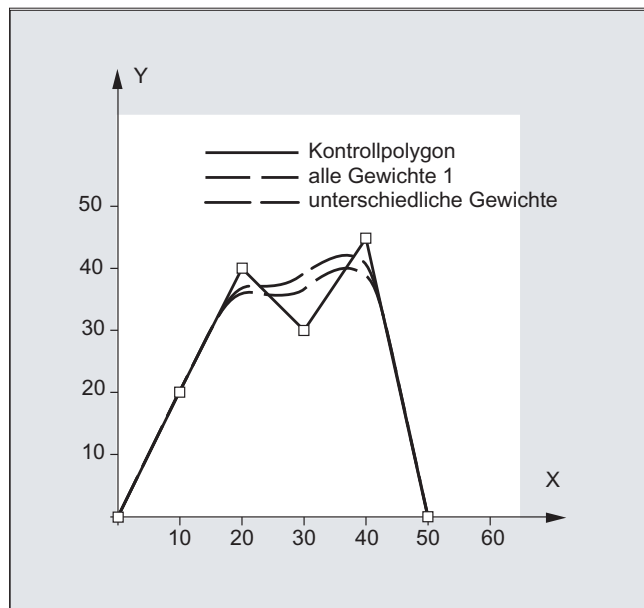
Programmcode 2 (unterschiedliche Gewichte)

```
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
```

Programmcode 3 (Kontrollpolygon)

Kommentar

Programmcode 3 (Kontrollpolygon)	Kommentar
N10 G1 X0 Y0 F300 G64	
N20	; entfällt
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	



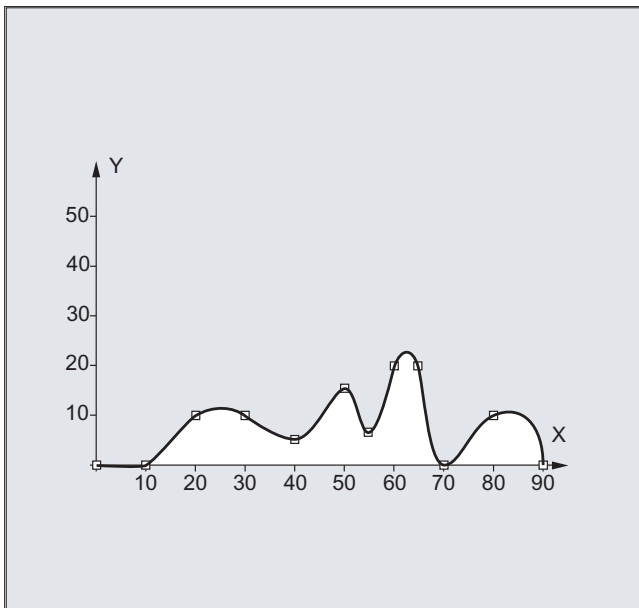
Beispiel 2: C-Spline, am Anfang und am Ende Krümmung Null

Programmcode

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
```

Programmcode

```
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
```



Beispiel 3: Spline-Interpolation (A-Spline) und Koordinatentransformation (ROT)

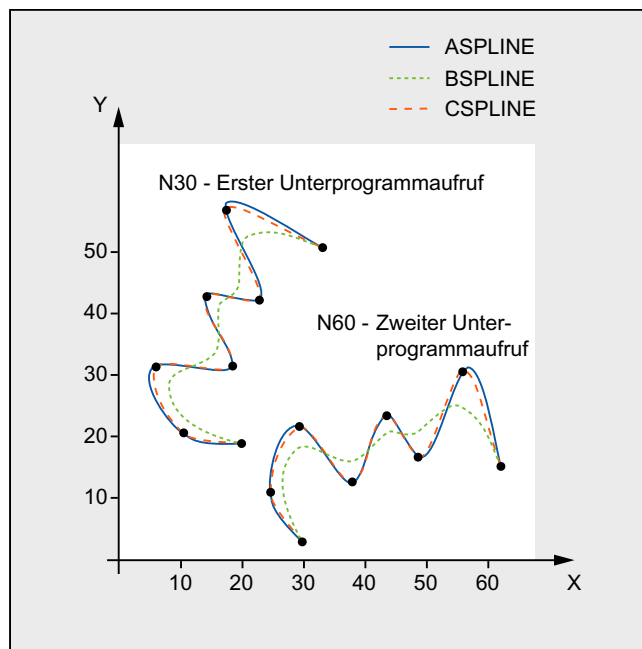
Hauptprogramm:

Programmcode	Kommentar
N10 G00 X20 Y18 F300 G64	; Startpunkt anfahren.
N20 ASPLINE	; Interpolationstyp A-Spline aktivieren.
N30 KONTUR	; Erster Aufruf des Unterprogramms.
N40 ROT Z-45	; Koordinatentransformation: Drehung des WKS um -45° um die Z-Achse.
N50 G00 X20 Y18	; Konturstartpunkt anfahren.
N60 KONTUR	; Zweiter Aufruf des Unterprogramms.
N70 M30	; Programmende

Unterprogramm "Kontur" (enthält die Stützpunkt-Koordinaten):

Programmcode
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

In der folgenden Abbildung sind neben der Spline-Kurve, die aus dem Programmbeispiel resultiert (ASPLINE), auch die Spline-Kurven enthalten, die sich bei Aktivierung einer B- oder C-Spline-Interpolation ergeben hätten (BSPLINE, CSPLINE):



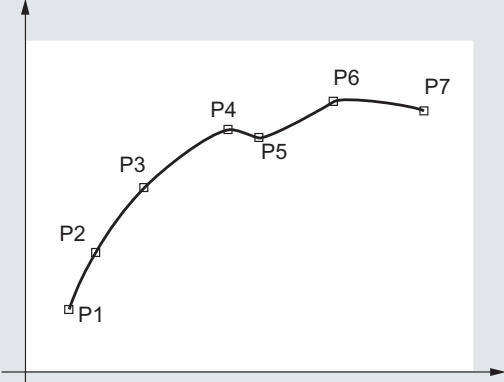
Weitere Informationen

Vorteile der Spline-Interpolation

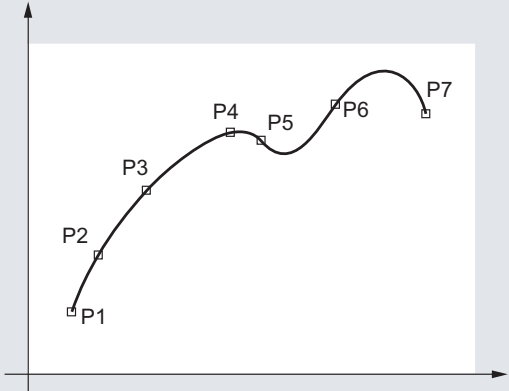
Durch Verwendung der Spline-Interpolation lassen sich, im Gegensatz zur Verwendung von Geradensätzen G01, folgende Vorteile erzielen:

- Reduzierung der Anzahl von benötigten Teileprogrammssätzen zur Beschreibung der Kontur
- Weicher, mechanischschonender Kurvenverlauf beim Übergang zwischen den Teileprogrammssätzen

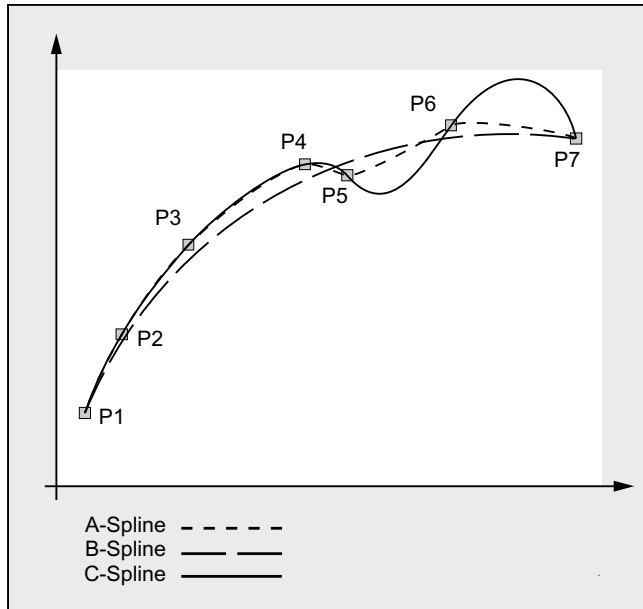
Eigenschaften und Anwendung der verschiedenen Spline-Typen

Spline-Typ	Eigenschaften und Anwendung
<p>A-Spline</p>	<div data-bbox="572 385 1222 991" style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <p style="text-align: center;">A-Spline (Akima-Spline)</p>  <p style="text-align: center;">P1 bis P7: vorgegebene Koordinaten</p> </div> <p>Eigenschaften:</p> <ul style="list-style-type: none"> • Verläuft exakt durch die vorgegebenen Stützpunkte. • Der Kurvenverlauf ist tangential- aber nicht krümmungsstetig. • Erzeugt kaum ungewollte Schwingungen. • Der Einflussbereich von Stützpunktänderungen ist lokal, d. h. Veränderung eines Stützpunkts wirkt sich nur auf bis zu max. 6 benachbarte Stützpunkte aus. <p>Anwendung:</p> <p>Der A-Spline eignet sich vor allem für die Interpolation von Kurvenverläufen mit großen Steigungsänderungen (z. B. treppenförmige Kurvenverläufe).</p>

Spline-Typ	Eigenschaften und Anwendung
<p>B-Spline</p>	<div data-bbox="611 325 1259 934" style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> </div> <p>Eigenschaften:</p> <ul style="list-style-type: none"> • Verläuft nicht durch die vorgegebenen Stützpunkte, sondern nur in deren Nähe. Die Kurve wird durch die Stützpunkte angezogen. Durch Gewichtung der Stützpunkte mit einem Faktor, kann der Kurvenverlauf zusätzlich beeinflusst werden. • Der Kurvenverlauf ist tangential- und krümmungsstetig. • Erzeugt keine ungewollten Schwingungen. • Der Einflussbereich von Stützpunktänderungen ist lokal, d. h. Veränderung eines Stützpunkts wirkt sich nur auf bis zu max. 6 benachbarte Stützpunkte aus. <p>Anwendung:</p> <p>Der B-Spline ist primär als Schnittstelle zu CAD-Systemen gedacht.</p>

Spline-Typ	Eigenschaften und Anwendung
<p>C-Spline</p>	<div data-bbox="571 323 1222 932" style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <p style="text-align: center;">C-Spline (kubischer Spline)</p>  <p style="text-align: center;">P1 bis P7: vorgegebene Koordinaten</p> </div> <p>Eigenschaften:</p> <ul style="list-style-type: none"> • Verläuft exakt durch die vorgegebenen Stützpunkte. • Der Kurvenverlauf ist tangential- und krümmungsstetig. • Erzeugt häufig ungewollten Schwingungen, besonders an Stellen mit großen Steigungsänderungen. • Der Einflussbereich von Stützpunktänderungen ist global, d. h. Veränderung eines Stützpunkts wirkt sich auf den gesamten Kurvenverlauf aus. <p>Anwendung:</p> <p>Der C-Spline kann dann gut eingesetzt werden, wenn die Stützpunkte auf einer analytisch bekannten Kurve liegen (Kreis, Parabel, Hyperbel)</p>

Gegenüberstellung der drei Spline-Typen bei gleichen Stützpunkten



Mindestanzahl an Spline-Sätzen

Die G-Codes `ASPLINE`, `BSPLINE` und `CSPLINE` verbinden Satzendpunkte mit Splines. Dazu müssen im Vorlauf eine Reihe von Sätzen (Endpunkte) gleichzeitig berechnet werden. Die Größe des Puffers für die Berechnung beträgt standardmäßig 10 Sätze. Nicht jede Satzinformation ist ein Spline-Endpunkt. Die Steuerung benötigt jedoch von 10 Sätzen eine bestimmte Anzahl an Spline-Endpunkt-Sätzen:

Spline-Typ	Mindestanzahl an Spline-Sätzen
A-Spline:	Von je 10 Sätzen müssen mindestens 4 Spline-Sätze sein. Kommentarsätze und Parameterrechnungen zählen hierbei nicht.
B-Spline:	Von je 10 Sätzen müssen mindestens 6 Spline-Sätze sein. Kommentarsätze und Parameterrechnungen zählen hierbei nicht.
C-Spline:	Die benötigte Mindestanzahl an Spline-Sätzen ergibt sich aus folgender Summe: Wert aus MD20160 <code>\$MC_CUBIC_SPLINE_BLOCKS + 1</code> Im MD20160 wird die Anzahl der Punkte eingetragen, über die der Spline-Abschnitt berechnet wird. Die Standardeinstellung beträgt 8. Von je 10 Sätzen müssen daher im Standardfall mindestens 9 Spline-Sätze sein.

Hinweis

Bei Unterschreitung des tolerierbaren Werts wird ein Alarm ausgegeben, ebenso, wenn eine am Spline beteiligte Achse als Positionierachse programmiert wird.

Zusammenfassung kurzer Spline-Sätze

Bei der Spline-Interpolation können kurze Spline-Sätze entstehen, die zu einer unnötigen Reduzierung der Bahngeschwindigkeit führen. Mit der Funktion "Zusammenfassung kurzer Spline-Sätze" können diese Sätze so zusammengefasst werden, dass die resultierende Satzlänge ausreichend groß ist und nicht zu einer Verringerung der Bahngeschwindigkeit führt.

Die Funktion wird aktiviert über das kanalspezifische Maschinendatum:

MD20488 \$MC_SPLINE_MODE (Einstellung für Spline-Interpolation)

Literatur:

Funktionshandbuch Grundfunktionen; Bahnsteuerbetrieb, Genauhalt, LookAhead (B1), Kapitel: Zusammenfassung kurzer Spline-Sätze

5.3 Spline-Verbund (SPLINEPATH)

Die im Spline-Verbund zu interpolierenden Achsen werden mit dem Befehl `SPLINEPATH` ausgewählt. Bis zu acht Bahnachsen sind bei der Spline-Interpolation möglich.

Hinweis

Wird `SPLINEPATH` nicht explizit programmiert, so werden die ersten drei Achsen des Kanals als Spline-Verbund verfahren.

Syntax

Die Festlegung des Spline-Verbundes erfolgt in einem gesonderten Satz:

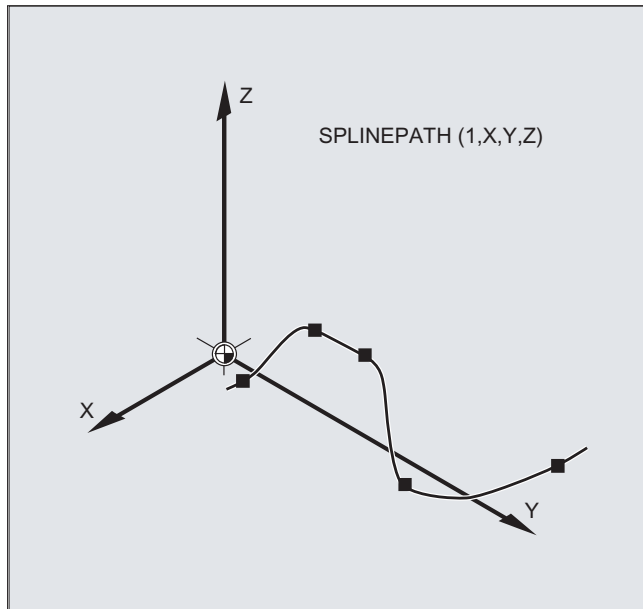
`SPLINEPATH (n, X, Y, Z, ...)`

Bedeutung

<code>SPLINEPATH:</code>	Befehl zur Festlegung eines Spline-Verbundes
<code>n:</code>	=1 (fester Wert)
<code>X, Y, Z, ... :</code>	Bezeichner der im Spline-Verbund zu interpolierenden Bahnachsen

Beispiel: Spline-Verbund mit drei Bahnachsen

Programmcode	Kommentar
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1, X, Y, Z)	; Spline-Verbund
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; C-Spline
N14 X30 Y40 Z50 A60 B70	; Stützpunkte
...	
N100 G1 X... Y...	; Abwahl Spline-Interpolation



5.4 NC-Satz-Kompression ein-/ausschalten (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

Die Kompressor-Funktionen werden mit den G-Befehlen der G-Gruppe 30 ein-/ausgeschaltet. Die Befehle sind modal wirksam.

Syntax

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

Bedeutung

COMPON:	Einschalten der Kompressor-Funktion COMPON
COMPCURV:	Einschalten der Kompressor-Funktion COMPCURV
COMPCAD:	Einschalten der Kompressor-Funktion COMPCAD
COMPSURF:	Einschalten der Kompressor-Funktion COMPSURF
COMPOF:	Ausschalten der aktuell aktiven Kompressor-Funktion

Hinweis

Zur zusätzlichen Verbesserung der Oberflächengüte kann die Überschleiffunktion G642 und die Ruckbegrenzung SOFT verwendet werden. Diese Befehle sind am Programmumfang zu schreiben.

Beispiel: COMPCAD

Programmcode	Kommentar
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	; Einschalten: Überschleiffunktion G642
N30 SOFT	; Einschalten: Ruckbegrenzung SOFT
N40 COMPCAD	; Einschalten: Kompressor-Funktion COMPCAD
N50 STOPFIFO	
N24050 Z32.499	; 1. Verfahrersatz
N24051 X41.365 Z32.500	; 2. Verfahrersatz
...	
N99999 X... Z...	; letzter Verfahrersatz
COMPOF	; Kompressor-Funktion aus.
...	

5.5 Polynom-Interpolation (POLY, POLYPATH, PO, PL)

Im eigentlichen Sinn handelt es sich bei der Polynom-Interpolation (POLY) nicht um eine Spline-Interpolationsart. Sie ist in erster Linie als Schnittstelle für die Programmierung extern erzeugter Spline-Kurven gedacht. Hierbei können die Spline-Abschnitte direkt programmiert werden.

Diese Interpolationsart entlastet die NC von der Berechnung der Polynom-Koeffizienten. Sie ist dann optimal einsetzbar, wenn die Koeffizienten direkt von einem CAD-System oder Post-Prozessor kommen.

Syntax

Polynom 3. Grades:

POLY PO[X]=(xe, a2, a3) PO[Y]=(ye, b2, b3) PO[Z]=(ze, c2, c3) PL=n

Polynome 5.Grades und neue Polynomsyntax:

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")

Hinweis

Die Summe der in einem NC-Satz programmierten Polynom-Koeffizienten und Achsen darf die maximal erlaubte Achszahl pro Satz nicht überschreiten.

Bedeutung

POLY :	Einschalten der Polynom-Interpolation mit einem Satz mit POLY.
POLYPATH :	Polynom-Interpolation selektierbar für die beiden Achsgruppen AXIS oder VECT

PO[Achsbezeichner/Variable] :	Endpunkte und Polynom-Koeffizienten
X, Y, Z :	Achsbezeichner
xe, ye, ze :	Angabe der Endposition für die jeweilige Achse; Wertebereich wie Wegmaß
a2, a3, a4, a5 :	Die Koeffizienten $a_2, a_3, a_4,$ und a_5 werden mit ihrem Wert geschrieben; Wertebereich wie Wegmaß. Der jeweils letzte Koeffizient kann entfallen, wenn er den Wert Null hat.
PL :	Länge des Parameterintervalls, auf dem die Polynome definiert sind (Definitionsbereich der Funktion $f(p)$). Das Intervall beginnt immer bei 0, p kann Werte von 0 bis PL annehmen. Theoretischer Wertebereich für PL: 0,0001 ... 99 999,9999 Hinweis: Der PL-Wert gilt für den Satz, in dem er steht. Ist kein PL programmiert, wirkt PL=1.

Ein-/Ausschalten der Polynom-Interpolation

Die Polynom-Interpolation wird im Teileprogramm durch den G-Befehl POLY eingeschaltet.

Der G-Befehl POLY gehört zusammen mit G0, G1, G2, G3, ASPLINE, BSPLINE und CSPLINE zur 1. G-Gruppe.

Achsen, die nur mit Namen und Endpunkt programmiert sind (z.B. X10), werden linear verfahren. Sind alle Achsen eines NC-Satzes so programmiert, verhält sich die Steuerung wie bei G1.

Die Polynom-Interpolation wird durch die Programmierung eines anderen Befehls der 1. G-Gruppe (z. B. G0, G1) implizit wieder ausgeschaltet.

Polynomkoeffizient

Der PO-Wert (PO[]=) bzw. ...=PO(...) gibt alle Polynom-Koeffizienten für eine Achse an. Entsprechend dem Grad des Polynoms werden mehrere Werte durch Kommata getrennt angegeben. Innerhalb eines Satzes sind unterschiedliche Polynomgrade für verschiedene Achsen möglich.

Unterprogramm POLYPATH

Mit POLYPATH(...) kann die Polynom-Interpolation selektiv für bestimmte Achsgruppen freigegeben werden:

Nur Bahnachsen und Zusatzachsen:	POLYPATH ("AXES")
Nur Orientierungsachsen: (beim Verfahren mit Orientierungs-Transformation)	POLYPATH ("VECT")

Die jeweils nicht freigegebenen Achsen werden linear verfahren.

Standardmäßig ist die Polynom-Interpolation für beide Achsgruppen freigegeben.

5.5 Polynom-Interpolation (POLY, POLYPATH, PO, PL)

Durch Programmierung ohne Parameter POLYPATH () wird die Polynom-Interpolation für alle Achsen deaktiviert.

Beispiel

Programmcode	Kommentar
N10 G1 X... Y... Z... F600	
N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2) PL=1.5	; Polynom-Interpolation ein
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	; gemischte Angaben für die Achsen
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	; kein PL programmiert; es wirkt PL=1
N30 G1 X... Y... Z.	; Polynom-Interpolation aus
...	

Beispiel: Neue Polynomsyntax

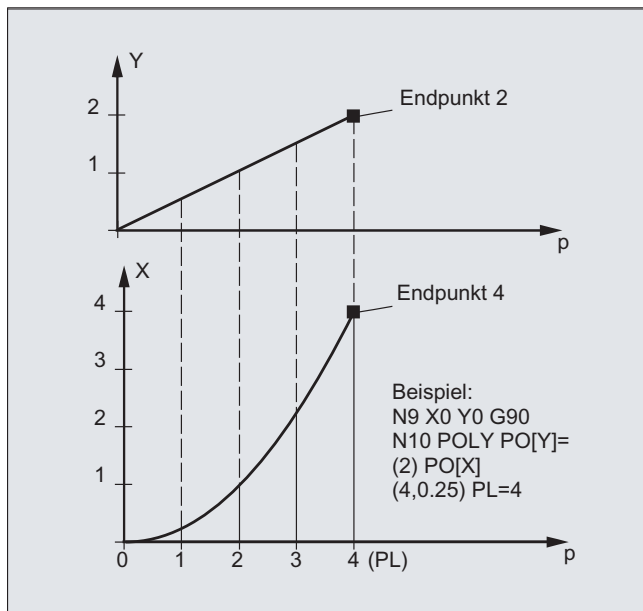
Weiterhin gültige Polynomsyntax	Neue Polynomsyntax
PO[Achsbezeichner]=(.. , ..)	Achsbezeichner=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[variable]=IC(.. , ..)	variable=PO IC(.. , ..)

Beispiel: Kurve in der X/Y-Ebene

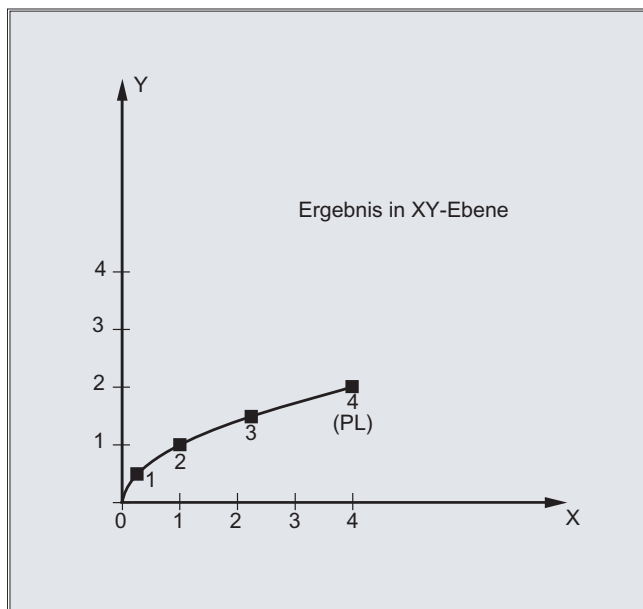
Programmierung

Programmcode
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

Verlauf der Kurven X(p) und Y(p)



Verlauf der Kurve in der XY-Ebene



Beschreibung

Die allgemeine Form der Polynom-Funktion lautet:

$$f(p) = a_0 + a_1p + a_2p^2 + \dots + a_np^n$$

mit: a_n : konstante Koeffizienten
 p: Parameter

In der Steuerung können maximal Polynome 5. Grades programmiert werden:

$$f(p) = a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

Durch Belegen der Koeffizienten mit konkreten Werten sind verschiedenen Kurvenverläufe, wie Geraden, Parabeln und Potenzfunktionen, erzeugbar.

Eine Gerade wird erzeugt durch $a_2 = a_3 = a_4 = a_5 = 0$:

$$f(p) = a_0 + a_1p$$

Weiter gilt:

a_0 : Achsposition am Ende des vorangehenden Satzes

$p = PL$

$$a_1 = (x_E - a_0 - a_2 * p^2 - a_3 * p^3) / p$$

Es ist möglich Polynome zu programmieren, **ohne** dass die Polynom-Interpolation durch den G-Befehl POLY aktiviert wurde. In diesem Fall werden nicht die programmierten Polynome interpoliert, sondern die programmierten Endpunkte der Achsen linear angefahren (G1). Erst nach expliziter Aktivierung der Polynom-Interpolation im Teileprogramm (POLY) werden die programmierten Polynome auch als solche verfahren.

Besonderheit: Nenner-Polynom

Für die Geometrieachsen kann mit $PO[] = (...)$ ohne Angabe eines Achsnamens auch ein gemeinsames Nenner-Polynom programmiert werden, d. h. die Bewegung der Geometrieachsen wird als Quotient zweier Polynome interpoliert.

Damit lassen sich z. B. Kegelschnitte (Kreis, Ellipse, Parabel, Hyperbel) exakt darstellen.

Beispiel:

Programmcode	Kommentar
POLY G90 X10 Y0 F100	; Geometrieachsen verfahren linear auf die Position X10 Y0.
PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)	; Geometrieachsen verfahren im Viertelkreis auf X0 Y10.

Der konstante Koeffizient (a_0) des Nenner-Polynoms wird stets mit 1 angenommen. Der programmierte Endpunkt ist unabhängig von G90 / G91.

Aus den programmierten Werten berechnen sich $X(p)$ und $Y(p)$ zu:

$$X(p) = (10 - 10 * p^2) / (1 + p^2)$$

$$Y(p) = 20 * p / (1 + p^2)$$

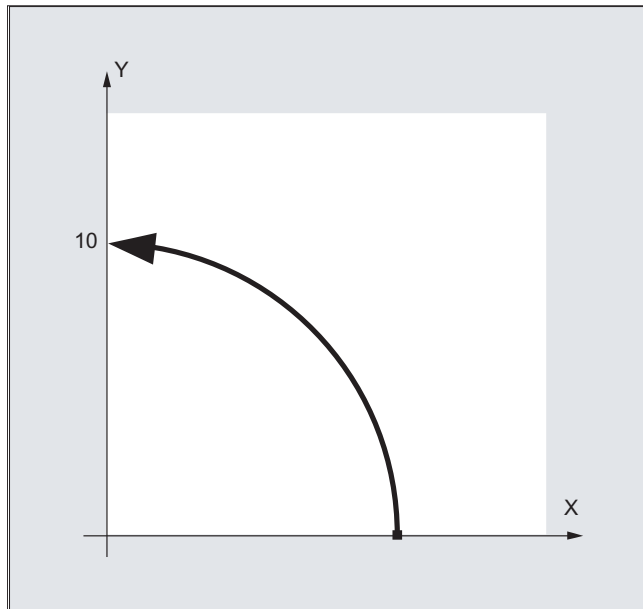
mit $0 \leq p \leq 1$

Aufgrund der programmierten Anfangspunkte, Endpunkte, Koeffizient a_2 und $PL=1$ ergeben sich folgende Zwischenergebnisse:

$$\text{Zähler (X)} = 10 + 0 * p - 10 * p^2$$

$$\text{Zähler (Y)} = 0 + 20 * p + 0 * p^2$$

$$\text{Nenner} = 1 + p^2$$



Bei eingeschalteter Polynom-Interpolation wird die Programmierung eines Nenner-Polynoms mit Nullstellen innerhalb des Intervalls $[0, PL]$ mit einem Alarm abgelehnt. Auf die Bewegung von Zusatzachsen hat das Nenner-Polynom keinen Einfluss.

Hinweis

Eine Werkzeugradiuskorrektur ist bei der Polynom-Interpolation mit G41, G42 einschaltbar und wie für Geraden- oder Kreisinterpolation verwendbar.

5.6 Einstellbarer Bahnbezug (SPATH, UPATH)

Bei Polynominterpolation (POLY, ASPLINE, BSPLINE, CSPLINE, COMCON, COMPCURV) werden die Positionen der Bahnachsen i durch Polynome $p_i(U)$ vorgegeben. Der Kurvenparameter U bewegt sich dabei innerhalb eines NC-Satzes von 0 bis 1.

Durch FGROUPE werden die Achsen (FGROUPE-Achsen) ausgewählt, auf die sich der Bahnvorschub F bezieht. Eine Interpolation mit konstanter Geschwindigkeit auf dem Bahnweg S der FGROUPE-Achsen bedeutet während der Polynominterpolation jedoch in der Regel eine nicht konstante Änderung des Kurvenparameters U . Für die nicht in FGROUPE enthaltenen Achsen kann daher zwischen zwei Möglichkeiten gewählt werden, wie diese den FGROUPE-Achsen folgen sollen:

- Synchron zum Bahnweg S (SPATH)
- Synchron zum Kurvenparameter U (UPATH)

Syntax

```
SPATH
UPATH
```

Bedeutung

SPATH:	Die nicht in FGROUPE enthaltenen Achsen werden bezogen auf den Bahnweg S verfahren
UPATH:	Die nicht in FGROUPE enthaltenen Achsen werden bezogen auf den Kurvenparameter U verfahren

Hinweis

UPATH und SPATH bestimmen auch den Zusammenhang des F-Wort-Polynoms (FPOLY, FCUB, FLIN) mit der Bahnbewegung.

Randbedingungen

SPATH bzw. UPATH haben keine Bedeutung bei:

- Linearinterpolation (G1)
- Kreisinterpolation (G2, G3)
- Gewindesätzen (G33, G34, G35, G33x, G63)
- Alle Bahnachsen sind in FGROUPE enthalten

Beispiel

Das folgende Beispiel zeigt den Unterschied zwischen den beiden Arten der Bewegungsführung.

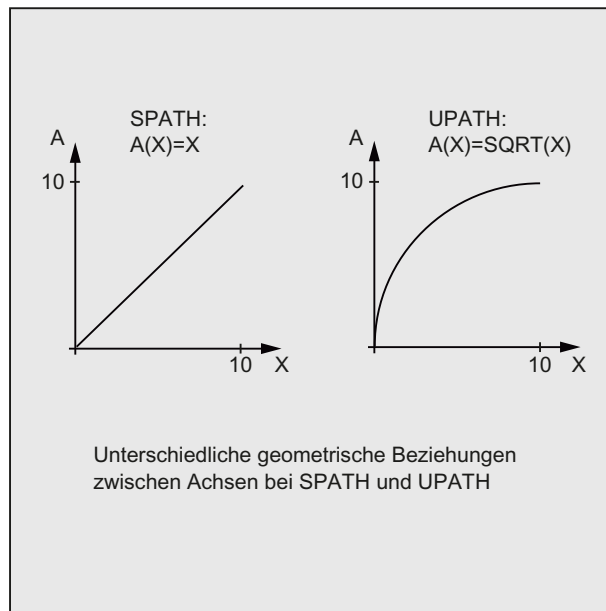
Programmcode

```
N10 FGROUPE (X, Y, Z)
N15 G1 X0 A0 F1000 SPATH ; SPATH
N20 POLY PO[X]=(10,10) A10
```

Programmcode

```
N10 FGROUPE (X, Y, Z)
N15 G1 X0 A0 F1000 UPATH ; UPATH
N20 POLY PO[X]=(10,10) A10
```

In beiden Programmausabschnitten hängt in N20 der Weg S der FGROUPE-Achsen vom Quadrat des Kurvenparameters U ab. Daher ergeben sich entlang des Wegs von X unterschiedliche Positionen der Synchronachse A, je nachdem, ob SPATH oder UPATH aktiv ist.



Weitere Informationen

Steuerungsverhalten bei Reset und Maschinen-/Optionsdaten

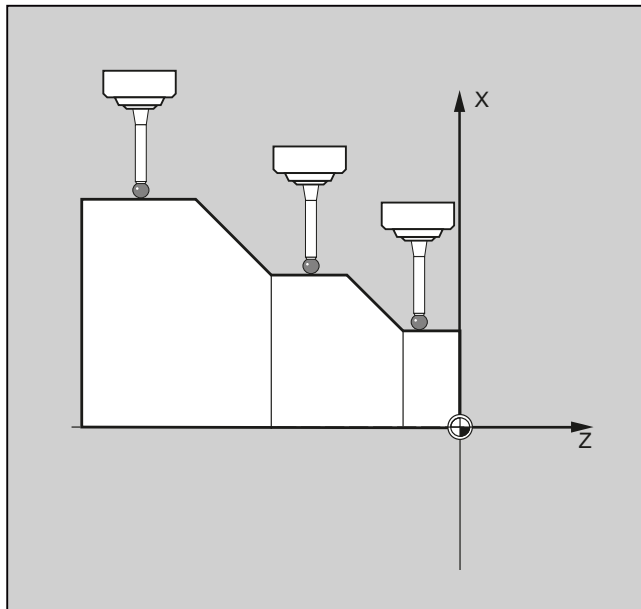
Nach Reset ist die durch MD20150 \$MC_GCODE_RESET_VALUES[44] bestimmte G-Funktion wirksam (45. G-Code-Gruppe).

Der Grundstellungswert für die Art des Überschleifens wird mit MD20150 \$MC_GCODE_RESET_VALUES[9] festgelegt (10. G-Code-Gruppe).

Das achsspezifische Maschinendatum MD33100 \$MA_COMPRESS_POS_TOL[<n>] hat eine erweiterte Bedeutung: es enthält die Toleranzen für die Kompressorfunktion und für das Überschleifen mit G642.

5.7 Messen mit schaltendem Taster (MEAS, MEAW)

Mit der Funktion "Messen mit schaltendem Taster" werden Istpositionen am Werkstück angefahren und bei der Schaltflanke des Messtasters werden für alle im Messatz programmierten Achsen die Positionen gemessen und für jede Achse in die entsprechende Speicherzelle geschrieben.



Für die Programmierung der Funktion stehen die beiden folgenden festen Adressen zur Verfügung:

- MEAS
Mit MEAS wird der Restweg zwischen Ist- und Sollposition gelöscht.
- MEAW
MEAW wird für Messaufgaben eingesetzt, bei denen in jedem Fall die programmierte Position angefahren werden soll.

MEAS und MEAW sind satzweise wirksam und werden zusammen mit Bewegungsanweisungen programmiert. Vorschub und Interpolationsart (G0, G1, ...), ebenso wie die Anzahl der Achsen, müssen dabei dem jeweiligen Messproblem angepasst sein.

Syntax

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Bedeutung

MEAS:	Befehl: Messen mit Restweglöschen	
	Wirksamkeit:	satzweise
MEAW:	Befehl: Messen ohne Restweglöschen	
	Wirksamkeit:	satzweise

<TE>:	Trigger-Ereignis zur Auslösung der Messung	
	Typ:	INT
	Wertebereich:	-2, -1, 1, 2
	Bedeutung:	
	(+)1	steigende Flanke von Messtaster 1 (auf Messeingang 1)
	-1	fallende Flanke von Messtaster 1 (auf Messeingang 1)
	(+)2	steigende Flanke von Messtaster 2 (auf Messeingang 2)
	-2	fallende Flanke von Messtaster 2 (auf Messeingang 2)
Hinweis: Es existieren maximal 2 Messtaster (je nach Ausbaustufe).		
G...:	Interpolationsart, z. B. G0, G1, G2 oder G3	
X... Y... Z...:	Endpunkte in kartesischen Koordinaten	

Beispiel

Programmcode	Kommentar
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; Messsatz mit Messtaster des ersten Messeingangs und Geradeninterpolation. Vorlaufstopp wird automatisch erzeugt.
...	

Weitere Informationen

Messauftragsstatus

Ist im Programm eine Auswertung erforderlich, ob der Messtaster geschaltet hat oder nicht, kann die Zustandsvariable \$AC_MEA[<n>] (<n> = Nummer des Messtasters) abgefragt werden:

Wert	Bedeutung
0	Messauftrag nicht erfüllt
1	Messauftrag erfolgreich beendet (Messtaster hat geschaltet)

Hinweis

Wird der Messtaster im Programm ausgelenkt, wird die Variable auf 1 gesetzt. Beim Start eines Messsatzes wird die Variable automatisch auf den Anfangszustand des Tasters gesetzt.

Messwerteaufnahme

Es werden die Positionen aller verfahrenen Bahn- und Positionierachsen des Satzes (maximale Anzahl an Achsen je nach Steuerungskonfiguration) erfasst. Bei MEAS wird die Bewegung nach dem Schalten des Messtasters definiert abgebremst.

Hinweis

Ist in einem Messsatz eine Geometrieachse programmiert, werden die Messwerte für alle aktuellen Geometrieachsen abgelegt.

Ist in einem Messsatz eine an einer Transformation beteiligte Achse programmiert, werden die Messwerte aller an dieser Transformation beteiligten Achsen abgelegt.

Messergebnisse lesen

Die Messergebnisse für die mit Messtaster erfassten Achsen können über die folgenden Systemvariablen gelesen werden:

- \$AA_MM[<Achse>]
Messergebnisse im Maschinenkoordinatensystem
- \$AA_MW[<Achse>]
Messergebnisse im Werkstückkoordinatensystem

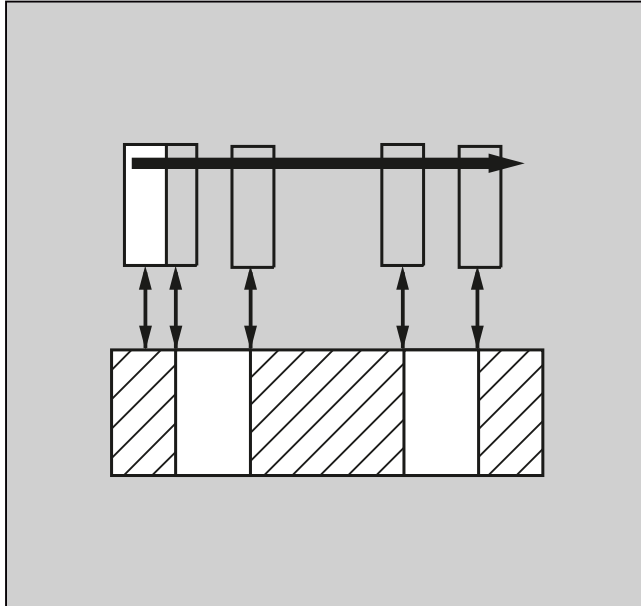
Beim Lesen dieser Variablen wird intern kein Vorlaufstopp erzeugt.

Hinweis

Mit STOPRE muss im NC-Programm an geeigneter Stelle ein Vorlaufstopp programmiert werden. Ansonsten werden falsche Werte gelesen.

5.8 Axiales Messen (MEASA, MEAWA, MEAC) (Option)

Beim axialen Messen können mehrere Messtaster und mehrere Messsysteme benutzt werden.



Für die Programmierung der Funktion stehen die Schlüsselwörter `MEASA`, `MEAWA` und `MEAC` zur Verfügung.

Mit `MEASA` bzw. `MEAWA` werden für die jeweils programmierte Achse bis zu vier Messwerte pro Messung erfasst und passend zum Trigger-Ereignis in Systemvariablen abgelegt.

Kontinuierliche Messaufträge können mit `MEAC` durchgeführt werden. In diesem Fall werden die Messergebnisse in FIFO-Variablen abgelegt.

Syntax

```
MEASA [<Achse>] = (<Modus>, <TE1>, ..., <TE4>)
MEAWA [<Achse>] = (<Modus>, <TE1>, ..., <TE4>)
MEAC [<Achse>] = (<Modus>, <Messspeicher>, <TE1>, ..., <TE4>)
```

Hinweis

`MEASA` und `MEAWA` sind satzweise wirksam und können zusammen in einem Satz programmiert werden. Wird dagegen `MEASA/MEAWA` zusammen mit `MEAS/MEAW` in einem Satz programmiert, kommt es zu einer Fehlermeldung.

Bedeutung

MEASA:	Schlüsselwort: Axiales Messen mit Restweglöschen	
	Wirksamkeit:	satzweise
MEAWA:	Schlüsselwort: Axiales Messen ohne Restweglöschen	
	Wirksamkeit:	satzweise

MEAC:	Schlüsselwort: Axiales kontinuierliches Messen ohne Restweglöschen	
	Wirksamkeit:	satzweise
<Achse>:	Name der zur Messung verwendeten Kanalachse	
<Modus>:	Zweistellige Ziffer zur Angabe des Betriebsmodus (Messmodus und Messsystem)	
	Einerdekade (Messmodus):	
	0	Messauftrag abbrechen.
	1	Bis zu 4 verschiedene gleichzeitig aktivierbare Trigger-Ereignisse.
	2	Bis zu 4 nacheinander aktivierbare Trigger-Ereignisse.
	3	Bis zu 4 nacheinander aktivierbare Trigger-Ereignisse, jedoch keine Überwachung von Trigger-Ereignis 1 beim Start (Alarmer 21700/21703 werden unterdrückt). Hinweis: Dieser Modus ist bei MEAC nicht möglich.
	Zehnerdekade (Messsystem):	
	0 (oder keine Angabe)	aktives Messsystem
	1	Messsystem 1
	2	Messsystem 2
3	beide Messsysteme	
<TE>:	Trigger-Ereignis zur Auslösung der Messung	
	Typ:	INT
	Wertebereich:	-2, -1, 1, 2
	Bedeutung:	
	(+)1	steigende Flanke von Messtaster 1
	-1	fallende Flanke von Messtaster 1
	(+)2	steigende Flanke von Messtaster 2
	-2	fallende Flanke von Messtaster 2
<Messspeicher>:	Nummer des FIFO (Umlaufspeichers)	

Beispiele

Beispiel 1: Axiales Messen mit Restweglöschen im Modus 1 (Auswertung in zeitlicher Reihenfolge)

a) mit 1 Messsystem

Programmcode	Kommentar
...	
N100 MEASA[X]=(1,1,-1) G01 X100 F100	; Messen im Modus 1 mit aktivem Messsystem. Warten auf Messsignal mit steigender/fallender Flanke von Messtaster 1 auf dem Fahrweg nach X=100.
N110 STOPRE	; Vorlaufstopp
N120 IF \$AC_MEA[1]==FALSE GOTO ENDE	; Erfolg der Messung kontrollieren.

5.8 Axiales Messen (MEASA, MEAWA, MEAC) (Option)

Programmcode	Kommentar
N130 R10=\$AA_MM1[X]	; Zum ersten programmierten Trigger-Ereignis (steigende Flanke) gehörigen Messwert speichern.
N140 R11=\$AA_MM2[X]	; Zum zweiten programmierten Trigger-Ereignis (fallende Flanke) gehörigen Messwert speichern.
N150 ENDE:	

b) mit 2 Messsystemen

Programmcode	Kommentar
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; Messen im Modus 1 mit beiden Messsystemen. Warten auf Messsignal mit steigender/fallender Flanke von Messtaster 1 auf dem Verfahrensweg nach X=100.
N210 STOPRE	; Vorlaufstopp
N220 IF \$AC_MEA[1]==FALSE GOTOF ENDE	; Erfolg der Messung kontrollieren.
N230 R10=\$AA_MM1[X]	; Messwert des Messsystems 1 bei steigender Flanke speichern.
N240 R11=\$AA_MM2[X]	; Messwert des Messsystems 2 bei steigender Flanke speichern.
N250 R12=\$AA_MM3[X]	; Messwert des Messsystems 1 bei fallender Flanke speichern.
N260 R13=\$AA_MM4[X]	; Messwert des Messsystems 2 bei fallender Flanke speichern.
N270 ENDE:	

Beispiel 2: Axiales Messen mit Restweglöschern im Modus 2 (Auswertung in programmierter Reihenfolge)

Programmcode	Kommentar
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; Messen im Modus 2 mit aktivem Messsystem. Warten auf Messsignal in der Reihenfolge steigende Flanke von Messtaster 1, fallende Flanke Messtaster 1, steigende Flanke von Messtaster 2, fallende Flanke Messtaster 2 auf dem Verfahrensweg nach X=100.
N110 STOPRE	; Vorlaufstopp
N120 IF \$AC_MEA[1]==FALSE GOTOF MESSTASTER2	; Erfolg der Messung mit Messtaster 1 kontrollieren.
N130 R10=\$AA_MM1[X]	; Zum ersten programmierten Trigger-Ereignis (steigende Flanke Messtaster 1) gehörigen Messwert speichern.

5.8 Axiales Messen (MEASA, MEAWA, MEAC) (Option)

Programmcode	Kommentar
N140 R11=\$AA_MM2[X]	; Zum zweiten programmierten Trigger-Ereignis (steigende Flanke Messtaster 1) gehörigen Messwert speichern.
N150 MESSTASTER2:	
N160 IF \$AC_MEA[2]==FALSE GOTOF ENDE	; Erfolg der Messung mit Messtaster 2 kontrollieren.
N170 R12=\$AA_MM3[X]	; Zum dritten programmierten Trigger-Ereignis (steigende Flanke Messtaster 2) gehörigen Messwert speichern.
N180 R13=\$AA_MM4[X]	; Zum vierten programmierten Trigger-Ereignis (steigende Flanke Messtaster 2) gehörigen Messwert speichern.
N190 ENDE:	

Beispiel 3: Axiales kontinuierliches Messen im Modus 1 (Auswertung in zeitlicher Reihenfolge)

a) Messen von bis zu 100 Messwerten

Programmcode	Kommentar
...	
N110 DEF REAL MESSWERT[100]	
N120 DEF INT Schleife=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; Messen im Modus 1 mit aktivem Messsystem, Speichern der Messwerte unter \$AC_FIFO1, Warten auf Messsignal mit fallender Flanke von Messtaster 1 auf dem Verfahrweg nach X=1000.
N135 STOPRE	
N140 MEAC[X]=(0)	; Messung nach Erreichen der Achsposition abbrechen.
N150 R1=\$AC_FIFO1[4]	; Anzahl aufgelaufener Messwerte in Parameter R1 speichern.
N160 FOR Schleife=0 TO R1-1	
N170 MESSWERT[Schleife]=\$AC_FIFO1[0]	; Messwerte aus dem \$AC_FIFO1 auslesen und abspeichern.
N180 ENDFOR	

b) Messen mit Restweglöschen nach 10 Messwerten

Programmcode	Kommentar
...	
N10 WHEN \$AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; Restweg löschen.
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC[X]=(0)	
N40 R1=\$AC_FIFO1[4]	; Anzahl Messwerte.
...	

c) Messen einer fallenden/steigenden Zahnflanke mit 2 Messtastern

Programmcode	Kommentar
...	
N110 DEF REAL MESSWERT[16]	
N120 DEF INT Schleife=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; Messen im Modus 1 mit aktivem Messsystem, Speichern der Messwerte unter \$AC_FIFO1, Warten auf Messsignal in der Reihenfolge fallende Flanke von Messtaster 1, steigende Flanke Messtaster 2, auf dem Verfahrensweg nach X=100.
N140 STOPRE	; Vorlaufstopp
N150 MEAC[X]=(0)	; Messung nach Erreichen der Achsposition abrechnen.
N160 R1=\$AC_FIFO1[4]	; Anzahl aufgelaufener Messwerte in Parameter R1 speichern.
N170 FOR Schleife=0 TO R1-1	
N180 MESSWERT[Schleife]=\$AC_FIFO1[0]	; Messwerte aus dem \$AC_FIFO1 auslesen und abspeichern.
N190 ENDFOR	

Weitere Informationen

Messauftrag

Die Programmierung eines Messauftrags kann im Teileprogramm oder aus einer Synchronaktion (siehe Kapitel "Synchronaktionen (Seite 531)") heraus erfolgen. Pro Achse kann dabei zu ein- und demselben Zeitpunkt nur ein Messauftrag aktiv sein.

Hinweis

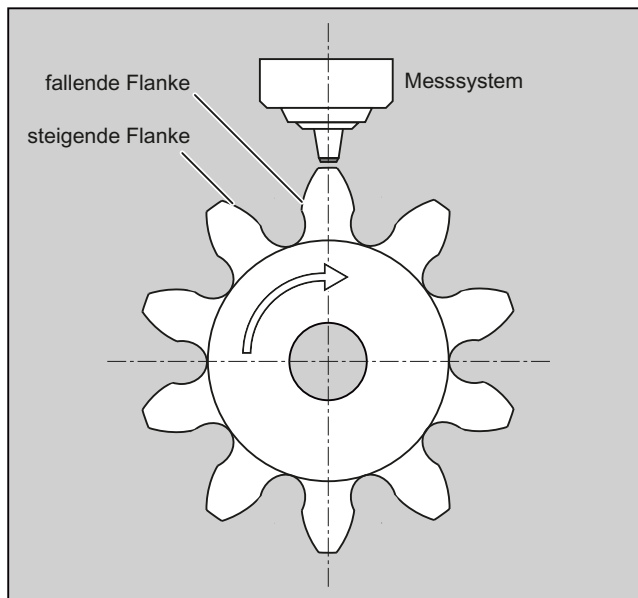
Der Vorschub ist dem jeweiligen Messproblem anzupassen.

Bei MEASA und MEAWA können korrekte Ergebnisse nur bei Vorschüben gewährleistet werden, bei denen nicht mehr als ein gleiches und nicht mehr als 4 verschiedene Trigger-Ereignisse pro Lageregler-Takt eintreffen.

Beim kontinuierlichen Messen mit MEAC darf das Verhältnis zwischen Interpolatortakt und Lageregler-Takt nicht größer als 1:8 werden.

Trigger-Ereignis

Ein Trigger-Ereignis setzt sich zusammen aus der Nummer des Messtasters und dem Auslösekriterium (steigende oder fallende Flanke) des Messsignals.



Für jede Messung können jeweils bis zu 4 Trigger-Ereignisse der angesprochenen Messtaster verarbeitet werden, also bis zu zwei Messtaster mit je zwei Messflanken. Die Reihenfolge der Verarbeitung sowie die maximale Anzahl der Trigger-Ereignisse sind dabei abhängig vom gewählten Modus.

Hinweis

Für Messmodus 1 gilt: Ein gleiches Trigger-Ereignis darf nur einmal in einem Messauftrag programmiert werden!

Bei MEAC kann die Anzahl der Messwerte pro Lageregler-Takt auf 8 von steigender Flanke und 8 von fallender Flanke je Messtaster erhöht werden. Damit lassen sich höhere Vorschübe und Drehzahlen realisieren.

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Messen (M5), Kapitel: Axiales Messen

Betriebsmodus

Mit der ersten Ziffer (Zehnerdekade) des Betriebsmodus wird das gewünschte Messsystem angewählt. Ist nur ein Messsystem vorhanden, jedoch das zweite programmiert, wird automatisch das vorhandene eingesetzt.

Mit der zweiten Ziffer (Einerdekade) wird der gewünschte Messmodus angewählt. Damit wird der Messvorgang an die Möglichkeiten der jeweiligen Steuerung angepasst:

- **Modus 1**
Die Auswertung der Trigger-Ereignisse erfolgt in der zeitlichen Reihenfolge ihres Auftretens. In diesem Modus ist bei Einsatz von Sechssachsbaugruppen nur ein Trigger-Ereignis programmierbar bzw. wird bei Angabe mehrerer Trigger-Ereignisse automatisch in Modus 2 umgesetzt (ohne Meldung).
- **Modus 2**
Die Auswertung der Trigger-Ereignisse erfolgt in der programmierten Reihenfolge.
- **Modus 3**
Die Auswertung der Trigger-Ereignisse erfolgt in der programmierten Reihenfolge, jedoch keine Überwachung von Trigger-Ereignis 1 beim START.

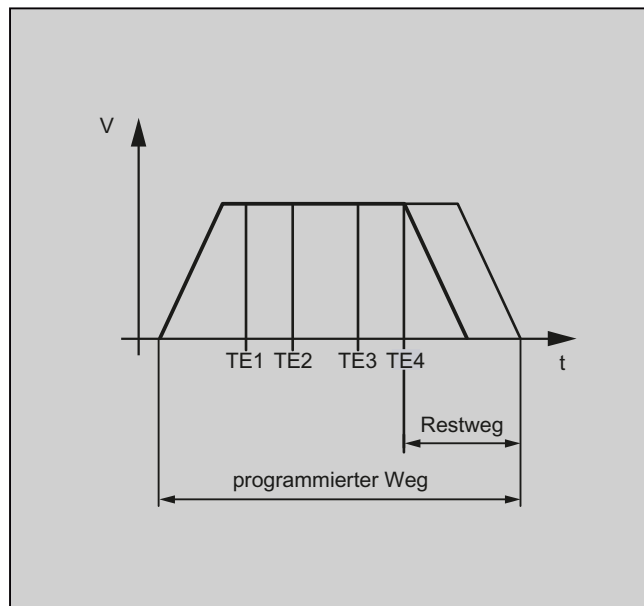
Hinweis

Bei Einsatz von 2 Messsystemen sind nur zwei Trigger-Ereignisse programmierbar.

Messen mit und ohne Restweglöschchen

Bei der Programmierung von MEASA wird Restweglöschchen erst nach der Erfassung aller geforderten Messwerte durchgeführt.

Für spezielle Messaufgaben, bei denen in jedem Fall die programmierte Position angefahren werden soll, wird MEAWA eingesetzt.



Hinweis

MEASA ist nicht in Synchronaktionen programmierbar. Ersatzweise kann MEAWA plus Restweglöschchen als Synchronaktion programmiert werden.

Wird der Messauftrag mit MEAWA aus den Synchronaktionen gestartet, sind die Messwerte nur im Maschinen-Koordinatensystem verfügbar.

5.8 Axiales Messen (MEASA, MEAWA, MEAC) (Option)

Messergebnisse für MEASA, MEAWA

Die Messergebnisse stehen unter folgenden Systemvariablen zur Verfügung:

- im Maschinen-Koordinatensystem:

\$AA_MM1 [<Achse>] Messwert des programmierten Messsystems bei Trigger-Ereignis 1

 \$AA_MM4 [<Achse>] Messwert des programmierten Messsystems bei Trigger-Ereignis 4

- im Werkstück-Koordinatensystem:

\$AA_WM1 [<Achse>] Messwert des programmierten Messsystems bei Trigger-Ereignis 1

 \$AA_WM4 [<Achse>] Messwert des programmierten Messsystems bei Trigger-Ereignis 4

Hinweis

Beim Lesen dieser Variablen wird intern kein Vorlaufstopp erzeugt. Mit `STOPRE` muss an geeigneter Stelle ein Vorlaufstopp programmiert werden. Ansonsten werden falsche Werte eingelesen.

Geometrieachsen / Transformationen

Soll das axiale Messen für eine Geometrieachse gestartet werden, muss der gleiche Messauftrag explizit für alle restlichen Geometrieachsen programmiert werden. Das Gleiche gilt für Achsen, die an einer Transformation beteiligt sind.

Beispiele:

N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100

oder

N10 MEASA[Z]=(1,1) POS[Z]=100

Messauftrag mit 2 Messsystemen

Wird ein Messauftrag mit zwei Messsystemen durchgeführt, wird jedes der beiden möglichen Trigger-Ereignisse von beiden Messsystemen der jeweiligen Achse erfasst. Die Belegung der reservierten Variablen ist damit vorgegeben:

\$AA_MM1 [<Achse>]	bzw.	\$AA_MW1 [<Achse>]	Messwert von Messsystem 1 bei Trigger-Ereignis 1
\$AA_MM2 [<Achse>]	bzw.	\$AA_MW2 [<Achse>]	Messwert von Messsystem 2 bei Trigger-Ereignis 1
\$AA_MM3 [<Achse>]	bzw.	\$AA_MW3 [<Achse>]	Messwert von Messsystem 1 bei Trigger-Ereignis 2
\$AA_MM4 [<Achse>]	bzw.	\$AA_MW4 [<Achse>]	Messwert von Messsystem 2 bei Trigger-Ereignis 2

Systemvariablen

Der Messtasterstatus steht unter der folgenden Systemvariablen zur Verfügung:

\$A_PROBE[<n>]

Wert	Bedeutung
1	Messtaster ausgelenkt
0	Messtaster nicht ausgelenkt

Die Messtasterbegrenzung steht unter der folgenden Systemvariablen zur Verfügung:

\$A_PROBE_LIMITED[<n>]

Wert	Bedeutung
1	Messtasterbegrenzung aktiv
0	Messtasterbegrenzung inaktiv

<n> = Messtaster

Literatur:

Listenhandbuch Systemvariablen

Messauftragsstatus bei MEASA, MEAWA

Ist im Programm eine Auswertung erforderlich, so kann der Messauftragsstatus über \$AC_MEA[<n>], mit <n> = Nummer des Messtasters, abgefragt werden. Sobald alle in einem Satz programmierten Trigger-Ereignisse der Messtaster <n> erfolgt sind, liefert diese Variable den Wert 1. Anderenfalls ist der Wert 0.

Hinweis

Wird Messen aus Synchronaktionen gestartet, wird \$AC_MEA nicht mehr aktualisiert. In diesem Fall sind das neue PLC-Nahtstellensignal DB31, ... DBX62.3 bzw. die gleichwertige Variable \$AA_MEA ACT[<Achse>] abzufragen.

Bedeutung:

\$AA_MEA ACT==1: Messen aktiv

\$AA_MEA ACT==0: Messen nicht aktiv

Kontinuierliches Messen (MEAC)

Die Messwerte liegen bei MEAC im Maschinenkoordinatensystem vor und werden im angegebenen FIFO[n]-Speicher (Umlaufspeicher) abgelegt. Sind für die Messung zwei Messtaster projektiert, werden die Messwerte des zweiten Messtasters getrennt im zusätzlich dafür projektierten (über MD einstellbar) FIFO[n+1]-Speicher abgelegt.

Der FIFO-Speicher ist ein Umlaufspeicher, in den Messwerte im Umlaufprinzip in \$AC_FIFO-Variablen eingetragen werden (siehe Kapitel "Synchronaktionen (Seite 531)").

Hinweis

Der FIFO-Inhalt kann nur einmal aus dem Umlaufspeicher ausgelesen werden. Zur Mehrfachverwendung der Messdaten müssen diese in den Anwenderdaten zwischengespeichert werden.

Überschreitet die Anzahl der Messwerte für den FIFO-Speicher die im Maschinendatum festgelegte Höchstzahl, so wird die Messung automatisch beendet.

Endloses Messen lässt sich durch zyklisches Auslesen von Messwerten realisieren. Das Auslesen muss dabei mindestens in der gleichen Häufigkeit wie der Eingang von neuen Messwerten erfolgen.

Literatur:

- Funktionshandbuch Synchronaktionen; Ausführliche Beschreibung, Kapitel: Parameter (\$AC_FIFO)
- Funktionshandbuch Erweiterungsfunktionen; Messen (M5), Kapitel: Axiales Messen

Schutz vor Fehlerprogrammierungen

Folgende Fehlprogrammierungen werden erkannt und mit einem Fehler angezeigt:

- MEASA/MEAWA zusammen mit MEAS/MEAW in einem Satz programmiert
Beispiel:
N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
- MEASA/MEAWA mit Parameteranzahl <2 oder >5
Beispiel:
N01 MEAWA[X]=(1) G01 F100 POS[X]=100
- MEASA/MEAWA mit Trigger-Ereignis ungleich 1/ -1/ 2/ -2
Beispiel:
N01 MEASA[B]=(1,1,3) B100
- MEASA/MEAWA mit falschem Modus
Beispiel:
N01 MEAWA[B]=(4,1) B100
- MEASA/MEAWA mit doppelt programmiertem Trigger-Ereignis
Beispiel:
N01 MEASA[B]=(1,1,-1,2,-1) B100
- MEASA/MEAWA und fehlende Geometrieachse
Beispiel:
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;GEO-Achse X/Y/Z
- Uneinheitlicher Messauftrag bei Geometrieachsen
Beispiel:
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100

5.9 Spezielle Funktionen für den OEM-Anwender (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

OEM-Adressen

Die Bedeutung der OEM-Adressen bestimmt der OEM-Anwender. Die Funktionalität wird über Compile-Zyklen eingebracht. 5 OEM-Adressen sind reserviert (OMA1 ... OMA5). Die Adressbezeichner sind einstellbar. OEM-Adressen sind in jedem Satz zulässig.

Reservierte G-Funktionsaufrufe

Für den OEM-Anwender sind folgende G-Funktionsaufrufe reserviert:

- OEMIPO1, OEMIPO2 (aus G-Funktionsgruppe 1)
- G810 ... G819 (G-Funktionsgruppe 31)
- G820 ... G829 (G-Funktionsgruppe 32)

Die Funktionalität wird über Compile-Zyklen eingebracht.

Funktionen und Unterprogramme

Zusätzlich können OEM-Anwender auch vordefinierte Funktionen und Unterprogramme mit Parameterübergabe anlegen.

Hinweis

Werkstücksimulation

Bis SW 4.4 werden keine, ab SW 4.4 nur ausgesuchte Compile-Zyklen (CC) bei der Werkstücksimulation unterstützt.

Sprachbefehle im Teileprogramm von nicht unterstützten Compile-Zyklen (OMA1 ... OMA5, OEMIPO1/2, G810 ... G829, eigene Prozeduren und Funktionen) führen daher ohne individuelle Behandlung zur Alarmmeldung und zum Abbruch der Simulation.

Lösung: Die fehlenden CC-spezifischen Sprachelemente im Teileprogramm individuell behandeln (\$P_SIM-Abfrage).

Beispiel:

```
N1 G01 X200 F500
IF (1== $P_SIM)
N5 X300 ;bei Simulation CC nicht aktiv
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 Vorschubreduzierung mit Eckenverzögerung (FENDNORM, G62, G621)

Bei der automatischen Eckenverzögerung wird der Vorschub glockenförmig kurz vor der betreffenden Ecke abgesenkt. Außerdem kann das Ausmaß des für die Bearbeitung relevanten Werkzeugverhaltens über Settingdaten parametrisiert werden. Dies sind:

- Beginn und Ende der Vorschubreduzierung
- Override, mit dem der Vorschub reduziert wird
- Erkennung der relevanten Ecke

Als relevante Ecken werden diejenigen Ecken berücksichtigt, deren Innenwinkel kleiner als die über Settingdatum parametrisierte Ecke ist.

Mit dem Defaultwert `FENDNORM` wird die Funktion des automatischen Eckenoverride ausgeschaltet.

Literatur:

/FBFA/ Funktionsbeschreibung ISO-Dialekte

Syntax

`FENDNORM`

`G62 G41`

`G621`

Bedeutung

<code>FENDNORM:</code>	Automatischer Eckenverzögerung aus
<code>G62:</code>	Eckenverzögerung an Innenecken bei aktiver Werkzeugradiuskorrektur
<code>G621:</code>	Eckenverzögerung an allen Ecken bei aktiver Werkzeugradiuskorrektur

G62 wirkt nur an den Innenecken mit

- aktiver Werkzeugradiuskorrektur `G41`, `G42` und
- aktiven Bahnsteuerbetrieb `G64`, `G641`

Die entsprechende Ecke wird mit dem abgesenkten Vorschub angefahren, der sich ergibt aus:

$F * (\text{Override zur Vorschubreduzierung}) * \text{Vorschuboverride}$

Die maximal mögliche Vorschubabsenkung wird genau dann erreicht, wenn das Werkzeug, bezogen auf die Mittelpunktbahn, den Richtungswechsel an der betreffenden Ecke vornehmen soll.

G621 wirkt analog zu G62 an jeder Ecke, der durch `FGROUP` festgelegten Achsen.

5.11 Programmierbares Bewegungsendekriterium (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Ähnlich dem Satzwechselkriterium bei Bahninterpolation (G601, G602 und G603) kann das Bewegungsendekriterium bei Einzelachsinterpolation in einem Teileprogramm bzw. in Synchronaktionen für Kommando-/PLC-Achsen programmiert werden.

Je nachdem, welches Bewegungsendekriterium eingestellt ist, werden Teileprogrammsätze bzw. Technologiezyklussätze mit Einzelachsbewegungen unterschiedlich schnell beendet. Gleiches gilt für PLC über FC15/16/18.

Syntax

```

FINEA [<Achse>]
COARSEA [<Achse>]
IPOENDA [<Achse>]
IPOBRKA (<Achse> [, <Zeitpunkt>])
ADISPOSA (<Achse> [, <Modus>, <Fenstergröße>])
    
```

Bedeutung

FINEA:	Bewegungsendekriterium: "Genauhalt fein"	
	Wirksamkeit:	modal
COARSEA:	Bewegungsendekriterium: "Genauhalt grob"	
	Wirksamkeit:	modal
IPOENDA:	Bewegungsendekriterium: "Interpolator-Stopp"	
	Wirksamkeit:	modal
IPOBRKA:	Satzwechselkriterium: Bremsrampe	
	Wirksamkeit:	modal
ADISPOSA:	Toleranzfenster zum Bewegungsendekriterium	
	Wirksamkeit:	modal
<Achse>:	Kanalachsname (X, Y, ...)	
<Zeitpunkt>:	Zeitpunkt des Satzwechsels, bezogen auf die Bremsrampe in %:	
	<ul style="list-style-type: none"> • 100% = Beginn der Bremsrampe • 0% = Ende der Bremsrampe, gleichbedeutend mit IPOENDA 	
	Typ:	REAL
<Modus>:	Bezug des Toleranzfensters	
	Wertebereich:	0 Toleranzfenster nicht aktiv
		1 Toleranzfenster bezüglich Sollposition
		2 Toleranzfenster bezüglich Istposition
	Typ:	INT
<Fenstergröße>:	Größe des Toleranzfensters	
	Typ:	REAL

Beispiele

Beispiel 1: Bewegungsendekriterium: "Interpolator-Stopp"

Programmcode
<pre> ; Fahre Positionierachse X auf 100, Geschwindigkeit 200 m/min, Beschleunigung 90%, ; Bewegungsendekriterium: Interpolator-Stopp N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X] ; Synchronaktion: ; IMMER WENN: Eingang 1 gesetzt ist ; DANN Fahre Positionierachse X auf 50, Geschwindigkeit 200 m/min, Beschleunigung 140%, ; Bewegungsendekriterium: Interpolator-Stopp N120 EVERY \$A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140 IPOENDA[X] </pre>

Beispiel 2: Satzwechselkriterium: "Bremsrampe"

Programmcode	Kommentar
	; Defaulteinstellung wirksam
N40 POS[X]=100	; Positionierbewegung von X auf Position 100 Satzwechselkriterium: Genauhalt fein
N20 IPOBRKA(X,100)	; Satzwechselkriterium: "Bremsrampe", 100% = Beginn der Bremsrampe
N30 POS[X]=200	; Satzwechsel erfolgt, sobald Achse X zu bremsen beginnt
N40 POS[X]=250	; Achse X brems nicht weiter auf Position 200, sondern fährt weiter auf Position 250. Sobald die Achse zu bremsen beginnt erfolgt der Satzwechsel.
N50 POS[X]=0	; Achse X brems und fährt auf Position 0 zurück. Satzwechsel erfolgt bei Position 0 und "Genauhalt fein"
N60 X10 F100	; Achse X fährt als Bahnachse auf Position 10

Weitere Informationen

Systemvariable für Bewegungsendekriterium

Das wirksame Bewegungsendekriterium kann über die Systemvariable \$AA_MOTEND gelesen werden.

Literatur: /LIS2sl/ Listenhandbuch, Buch 2

Satzwechselkriterium: "Bremsrampe" (IPOBRKA)

Ist bei der Aktivierung des Satzwechselkriteriums "Bremsrampe" für den optionalen Satzwechselzeitpunkt ein Wert programmierte, wird dieser für die nächste Positionierbewegung wirksam und hauptlaufsynchron in das Settingdatum geschrieben. Ist für

5.11 Programmierbares Bewegungsendekriterium (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

den Satzwechselzeitpunkt kein Wert angegeben, wird der aktuelle Wert des Settingdatums wirksam.

SD43600 \$SA_IPOBRAKE_BLOCK_EXCHANGE

Mit der nächsten Programmierung eines axialen Bewegungsendekriteriums (FINEA, COARSEA , IPOENDA) wird IPOBRKA für die entsprechende Achse deaktiviert.

Zusätzliches Satzwechselkriterium: "Toleranzfenster" (ADISPOSA)

Mit ADISPOSA kann als zusätzliches Satzwechselkriterium ein Toleranzfenster um den Satzpunkt (wahlweise Ist- oder Sollposition) definiert werden. Für den Satzwechsel müssen dann beide Bedingungen erfüllt sein:

- Satzwechselkriterium: "Bremsrampe"
- Satzwechselkriterium: "Toleranzfenster"

Literatur

Weitere Informationen zum Satzwechselkriterium von Positionierachsen siehe:

- Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)
- Programmierhandbuch Grundlagen; Kapitel "Vorschubregelung"

Koordinatentransformationen (Frames)

6.1 Koordinatentransformation über Framevariable

Neben den im Programmierhandbuch Grundlagen, Kapitel "Koordinatentransformationen (Frames)" beschriebenen Befehlen wie z.B. `ROT`, `AROT`, `SCALE`, etc. kann das Werkstück-Koordinatensystem (WKS) auch über die Framevariablen `$P_...FR` (Datenhaltungsframes) und `$P_...FRAME` (Aktive Frames) transformiert werden.

Das folgende Bild gibt einen Überblick über die Strukturierung der Framevariablen:

- Datenhaltungsframes
- Aktive Frames
- Aktiver Gesamtframe: Verkettung aller aktiven Frames
- NCU-globale Frames
- Kanalspezifische Frames

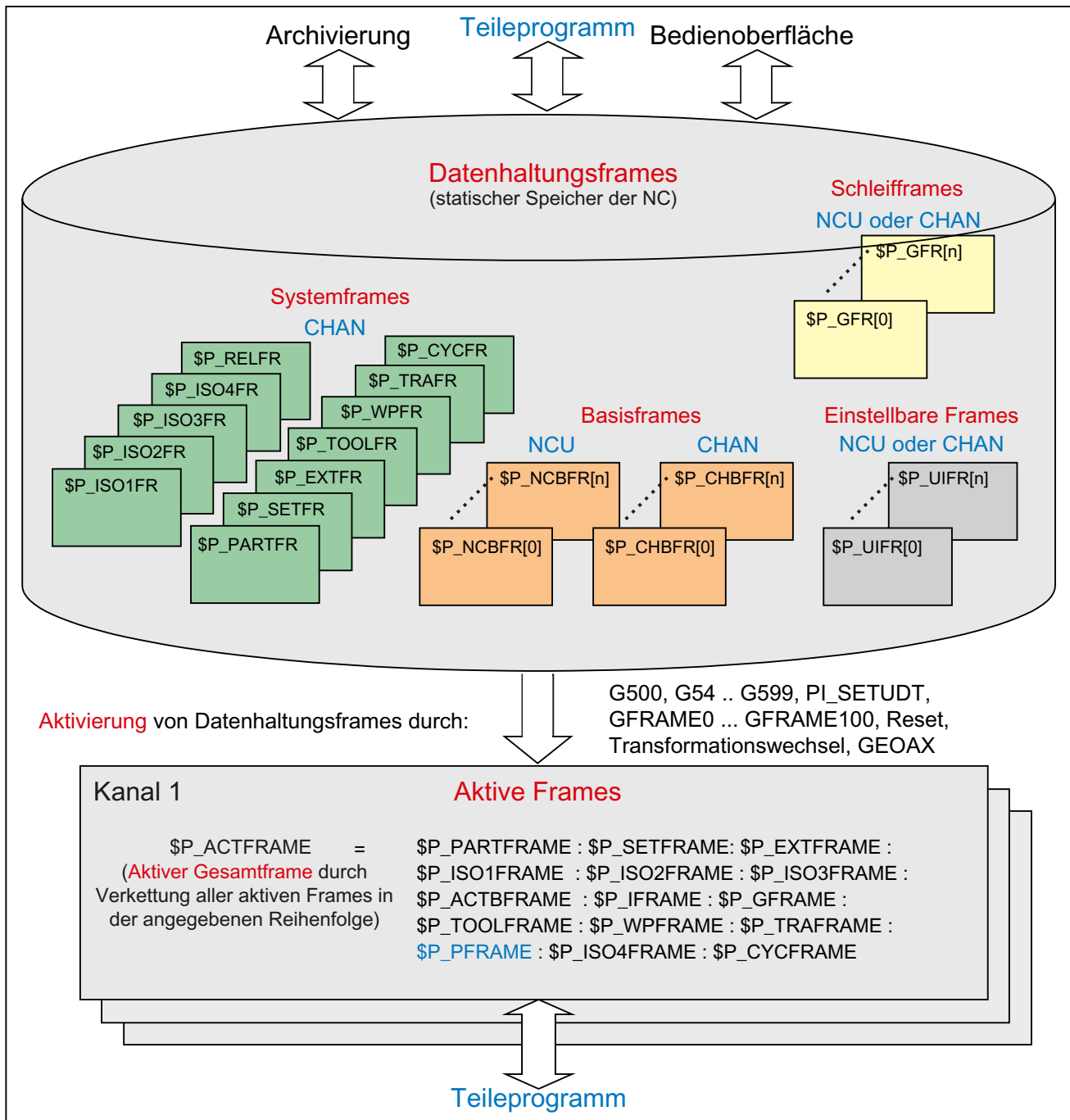


Bild 6-1 Übersicht der Framevariablen

6.1.1 Vordefinierte Framevariable (\$P_CHBFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME)

Aktiv: kanalspezifische Basisframes \$P_CHBFRAME[<n>] (\$P_BFRAME)

Hinweis

Die der aktuelle Basisframe \$P_BFRAME und der Datenhaltungs-Basisframe \$P_UBFR werden aus Kompatibilitätsgründen beibehalten.

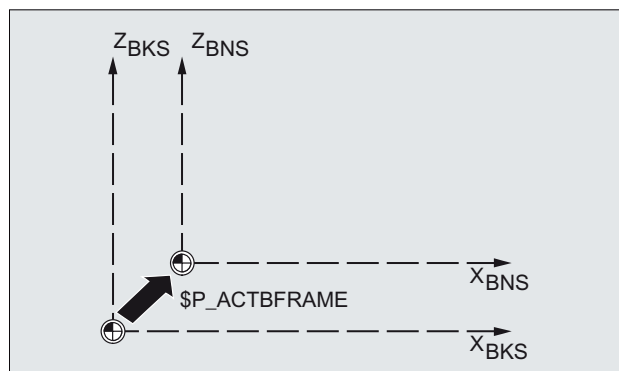
- \$P_BFRAME □ \$P_CHBFRAME[0]
- \$P_UBFR □ \$P_CHBFR[0].

Die Framevariablen \$P_CHBFRAME[<n>] definieren den Bezug zwischen Basiskoordinatensystem (BKS) und Basis-Nullpunktsystem (BNS).

Soll der aktuelle kanalspezifische Basisframe \$P_CHBFRAME[<n>] sofort im NC-Programm wirksam werden, stehen folgende Möglichkeiten zu Verfügung

- Befehle:
 - G500 (Ausschalten aller Einstellbaren Frames, die Basisframes bleiben aktiv)
 - G54 ... G599 (Einstellbare Nullpunktverschiebungen)
- Zuweisung eines kanalspezifischen Basisframes der Datenhaltung an einen aktuellen kanalspezifischen Basisframe:

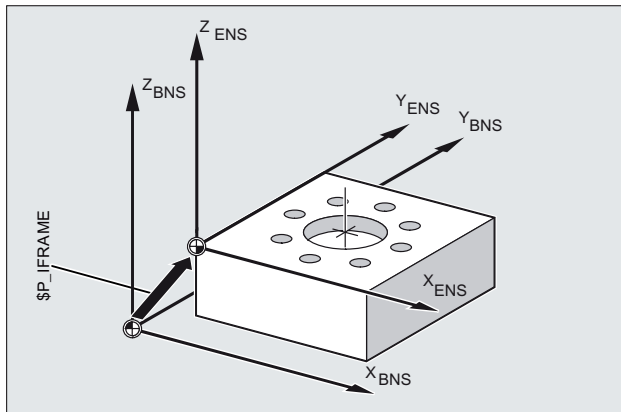
$$\$P_CHBFRAME[<n>] = \$P_CHBFR[<m>]$$



Aktiv: kanalspezifischer Einstellbarer Frame \$P_IFRAME

Die Framevariablen \$P_IFRAME, definiert den Bezug zwischen Basis-Nullpunktsystem (BNS) und Einstellbarem Nullpunktsystem (ENS).

- \$P_IFRAME entspricht \$P_UIFR[\$P_IFRNUM]
- \$P_IFRAME enthält nach Programmierung von z. B. G54 die durch G54 definierte Translation, Rotation, Skalierung und Spiegelung.

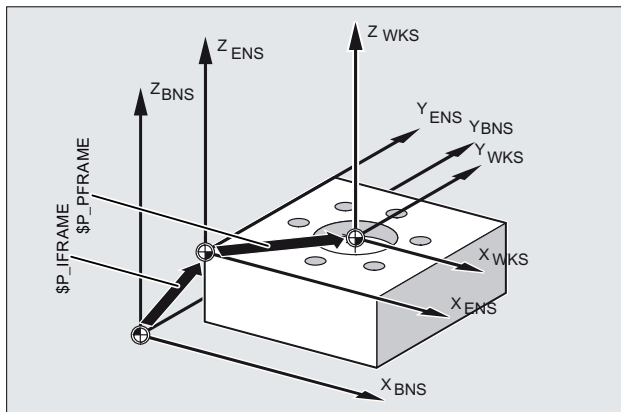


Aktiv: kanalspezifischer Programmierbarer Frame \$P_PFRAME

Die Framevariable \$P_PFRAME, definiert den Bezug zwischen dem Einstellbaren Nullpunktsystem (ENS) und dem Werkstückkoordinatensystem (WKS).

\$P_PFRAME enthält den resultierenden Frame, der sich

- aus der Programmierung von TRANS/ATRANS, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR bzw.
- aus der Zuweisung von CTRANS, CROT, CMIRROR, CSCALE an den programmierbaren FRAME ergibt



Aktiv: Gesamtframe \$P_ACTFRAME

Der im Kanal wirksame Gesamtframe, ergibt sich aus der Verkettung aller im Kanal wirksamen Frames.

$$\begin{aligned}
 \$P_ACTFRAME = & \quad \$P_PARTFRAME : \$P_SETFRAME : \$P_EXTFRAME : \\
 & \quad \$P_ISO1FRAME : \$P_ISO2FRAME : \$P_ISO3FRAME : \\
 & \quad \$P_ACTBFRAME : \$P_IFRAME : \$P_GFRAME : \\
 & \quad \$P_TOOLFRAME : \$P_WPFRAME : \$P_TRAFRAME : \\
 & \quad \$P_PFRAME : \$P_ISO4FRAME : \$P_CYCFRAME
 \end{aligned}$$

\$P_ACTFRAME beschreibt den Nullpunkt des aktuellen Werkstückkoordinatensystems.

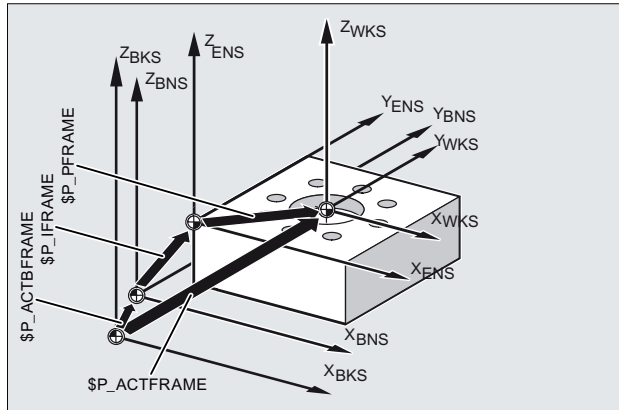
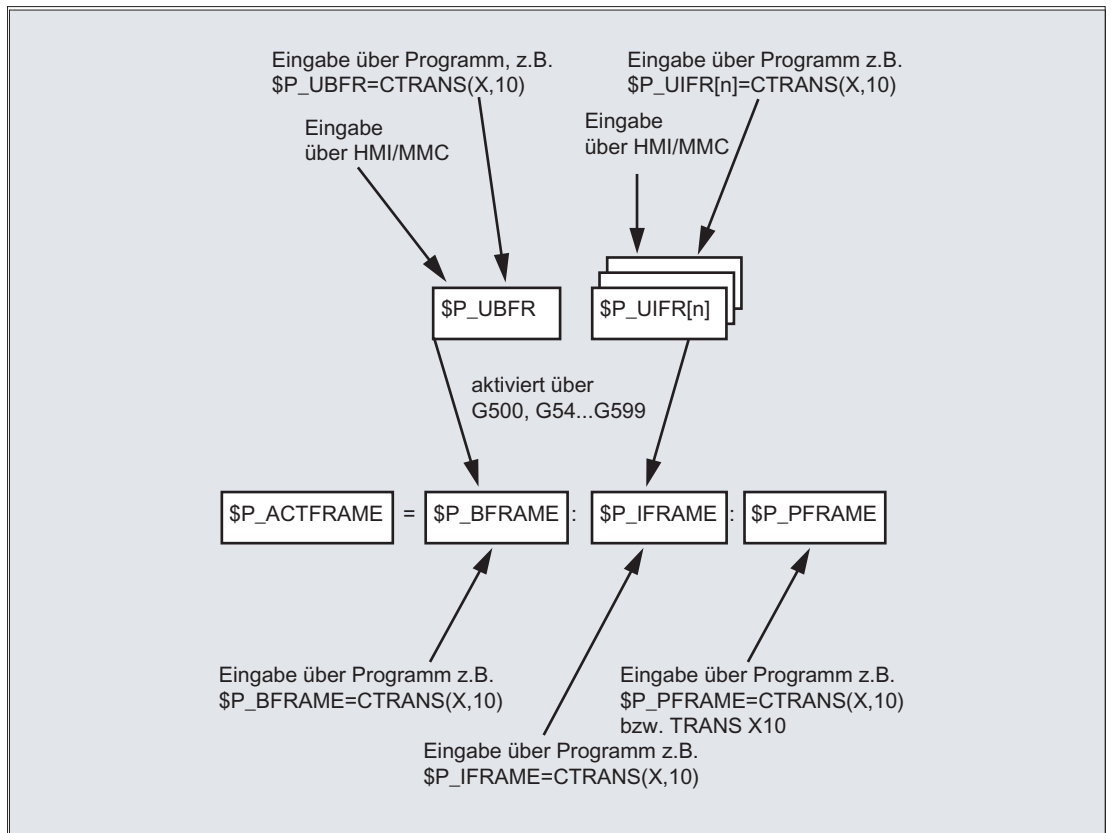


Bild 6-2 Framevariable \$P_ACTFRAME

Wird einer der folgenden Frames \$P_BFRAME / \$P_CHBFRAME [<n>], \$P_IFRAME oder \$P_PFRAME verändert, wird der aktuelle Gesamtframe \$P_ACTFRAME neu berechnet.



Basisframe und Einstellbarer Frame wirken nach Reset, wenn das MD 20110 RESET_MODE_MASK folgendermaßen eingestellt ist:

Bit0=1, Bit14=1 --> \$P_UBFR (Basisframe) wirkt

Bit0=1, Bit5=1 --> \$P_UIFR[\$P_UIFRNUM] (einst. Frame) wirkt

Datenhaltung: Kanalspezifische Basisframes \$P_CHBFR[<n>]

Über die Framevariablen \$P_CHBFR[<n>] werden die Basisframes in der Datenhaltung gelesen / geschrieben. Der Datenhaltungsframe wird durch das Schreiben nicht sofort im Kanal aktiv. Das Aktivieren des geschriebenen Frames erfolgt bei:

- Kanal-Reset und MD20110 \$MC_RESET_MODE_MASK, Bit0 == 1 und Bit14 == 1
- Befehl G500, G54 ... G57, G505 ... G599 (Ein/Ausschalten von Basisframes mit anschließender Neuberechnung des Aktuellen Gesamtframes)

Datenhaltung: Kanalspezifische Einstellbare Frames \$P_UIFR[<n>]

Über die Framevariablen \$P_UIFR[<n>] werden die Einstellbaren Frames in der Datenhaltung gelesen / geschrieben. Der Frame wird durch das Schreiben nicht sofort im Kanal aktiv. Das Einrechnen des geschriebenen Frames im Kanal erfolgt bei:

- Befehl G500 (Ausschalten aller Einstellbaren Frames bzw. Nullpunktverschiebungen)
- Befehl G54 ... G57, G505 ... G599 (Einschalten eines Einstellbaren Frames bzw. Nullpunktverschiebung)

Aktiver Einstellbarer Frame	Datenhaltungsframe	(entspricht Befehl)
\$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 Wertzuweisungen an Frames

6.2.1 Direkte Werte zuweisen (Achswert, Winkel, Maßstab)

Im NC-Programm können Sie direkt Frames oder Framevariablen mit Werten belegen.

Syntax

Syntax

\$P_PFRAME = CTRANS(X, <Verschiebungswert>, Y, <Verschiebungswert>, Z, <Verschiebungswert>, ...)

\$P_PFRAME = ROT(X, <Winkel>, Y, <Winkel>, Z, <Winkel>, ...)

\$P_UIFR[..] = CROT(X, <Winkel>, Y, <Winkel>, Z, <Winkel>, ...)

```
$P_PFRAME = CSCALE(X, <Maßstab>, Y, <Maßstab>, Z, <Maßstab>, ...)
```

```
$P_PFRAME = CMIRROR(X, Y, Z)
```

Die Syntax für \$P_CHBFRAME [<n>] ist identisch zu \$P_PFRAME.

Bedeutung

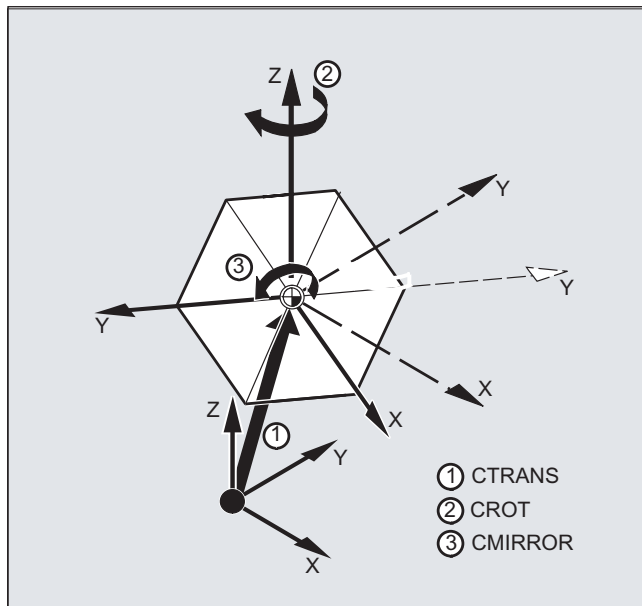
CTRANS:	Verschiebung in den angegebenen Achsen
CROT:	Drehung um die angegebenen Achsen
CSCALE:	Maßstabsveränderung in den angegebenen Achsen
CMIRROR:	Richtungsumkehr der angegebenen Achse
X, Y, Z:	Verschiebewert in Richtung der angegebenen Geometrieachse
<Verschiebungswert>:	Verschiebungswert
<Winkel>:	Winkel um den gedreht wird
<Maßstab>:	Maßstabsangabe

Beispiele

Wertzuweisungen an Framekomponenten des aktuellen Programmierbaren Frames

Wertzuweisung an die Framekomponenten Translation, Drehung und Spiegelung des aktuellen Programmierbaren Frames:

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```



Rotationskomponenten eines Frames schreiben

Zuweisen von Werten an alle drei Achsen der Drehkomponente des Einstellbaren Datenhaltungsframes \$P_UIFR mit CROT :

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

Alternativ dazu die direkte Zuweisung der Einzelwerte direkt an die jeweilige Achse der Drehkomponente des Datenhaltungsframes:

```
$P_UIFR[5, Y, RT]=0  
$P_UIFR[5, X, RT]=0  
$P_UIFR[5, Z, RT]=0
```

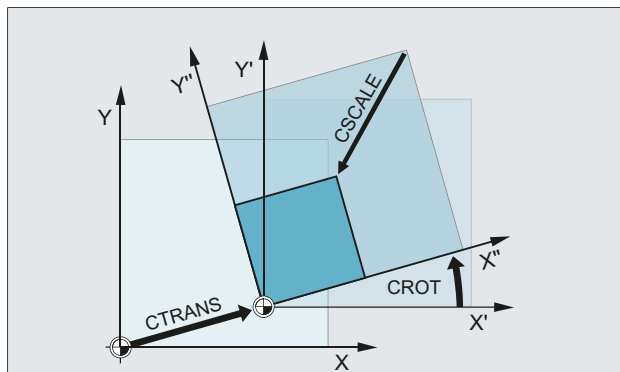
Beschreibung

Mehrere Operationen auf einen Frame können durch den Kettungsoperator : miteinander verbunden werden. Die Operationen werden dann nacheinander von links nach rechts ausgeführt.

Beispiel

Verkettete Operationen auf \$P_PFRAME mit Verschiebung, Rotation und Skalierung:

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```



6.2.2 Framekomponenten lesen und verändern (TR, FI, RT, SC, MI)

Sie haben die Möglichkeit, auf **einzelne** Daten eines Frames, z. B. auf einen bestimmten Verschiebewert oder Drehwinkel zuzugreifen. Diese Werte können Sie verändern oder einer anderen Variablen zuweisen.

Syntax

```
R10=$P_UIFR[$P_UIFNUM, X, RT]
```

Der Drehwinkel RT um die X-Achse aus der aktuell gültigen einstellbaren Nullpunktverschiebung \$P_UIFRNUM soll der Variablen R10 zugewiesen werden.

```
R12=$P_UIFR[25, Z, TR]
```

Der Verschiebewert TR in Z aus dem Datensatz des eingestellten Frames Nr. 25 soll der Variablen R12 zugewiesen werden.

R15=\$P_PFRAME[Y, TR]

Der Verschiebewert TR in Y des aktuellen programmierbaren Frames soll der Variablen R15 zugewiesen werden.

\$P_PFRAME[X, TR]=25

Der Verschiebewert TR in X des aktuellen programmierbaren Frames soll verändert werden. Ab sofort gilt X25.

Bedeutung

\$P_UIFRNUM:	Mit dieser Variablen wird automatisch der Bezug zur aktuell gültigen einstellbaren Nullpunktverschiebung hergestellt.
P_UIFR[n, ..., ...] :	Durch Angabe der Framenummer n greifen Sie auf den einstellbaren Frame Nr. n zu.
	Angabe der Komponente, die gelesen oder verändert werden soll:
TR:	TR Translation
FI:	FI Translation Fine
RT:	RT Rotation
SC:	SC Scale Maßstabsveränderung
MI:	MI Spiegelung
X, Y, Z:	Zusätzlich (siehe Beispiele) wird die entsprechende Achse X, Y, Z angegeben.

Wertebereich für Drehung RT

Drehung um 1. Geometrieachse: -180° bis +180°

Drehung um 2. Geometrieachse: -90° bis +90°

Drehung um 3. Geometrieachse: -180° bis +180°

Beschreibung

Frame aufrufen

Durch Angabe der Systemvariablen \$P_UIFRNUM können Sie direkt auf die mit \$P_UIFR bzw. G54, G55, ... aktuell eingestellte Nullpunktverschiebung zugreifen (\$P_UIFRNUM enthält die Nummer des aktuell eingestellten Frames).

Alle anderen gespeicherten einstellbaren Frames \$P_UIFR rufen Sie durch Angabe der entsprechenden Nummer \$P_UIFR[n] auf.

Für vordefinierte Framevariable und eigendefinierte Frames geben Sie den Namen an, z. B. \$P_IFRAME.

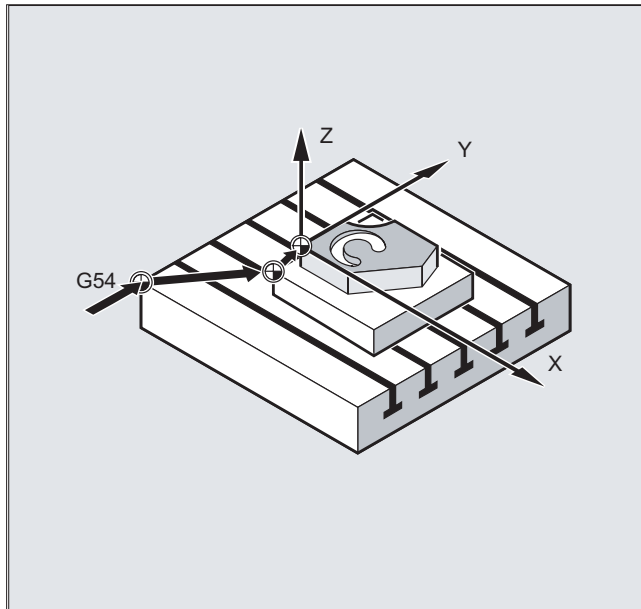
Daten aufrufen

In den eckigen Klammern stehen Achsname und Framekomponente des Wertes, auf den Sie zugreifen oder den Sie verändern wollen, z. B. [X, RT] oder [Z, MI].

6.2.3 Rechnen mit Frames

Im NC-Programm kann ein Frame einem anderen Frame zugewiesen oder Frames miteinander verkettet werden.

Frameverkettungen eignen sich z. B. für die Beschreibung mehrerer Werkstücke, die auf einer Palette angeordnet sind und in einem Fertigungsablauf bearbeitet werden sollen.



Für die Beschreibung von Palettenaufgaben könnten die Framekomponenten z. B. nur bestimmte Teilwerte enthalten, durch deren Verkettung verschiedene Werkstücknullpunkte generiert werden.

Beispiele

Zuweisungen

Programmcode	Kommentar
DEF FRAME EINSTELLUNG_1	; Definition einer lokalen Framevariablen
EINSTELLUNG_1 = CTRANS(X,10)	; Zuweisung des Ergebnisses einer Funktion an die Framevariable
\$P_PFRAME = EINSTELLUNG_1	; Zuweisung der Framevariablen an den Aktuellen Frame
DEF FRAME EINSTELLUNG_4	; Definition einer lokalen Framevariablen
EINSTELLUNG_4 = \$P_PFRAME	; Zwischenspeicherung des Aktuellen Frames in der Framevariablen
...	
\$P_PFRAME = EINSTELLUNG_4	; Rücklesen des Aktuellen Frames aus der Framevariablen

Verkettungen

Die Frames werden in der programmierten Reihenfolge durch den Operator `:` miteinander verkettet. Die Framekomponenten wie z. B. Verschiebungen, Drehungen usw. werden nacheinander additiv ausgeführt.

Programmcode	Kommentar
\$P_IFRAME = \$P_UIFR[15] : \$P_UIFR[16]	; Zuweisung des Ergebnis-Frames aus der Verkettung der ; beiden Einstellbaren Datenhaltungsframes an den aktiven ; Einstellbaren Gesamt-Frame. ; Anwendungsbeispiel: ; \$P_UIFR[15]: Verschiebung ; \$P_UIFR[16]: Drehung
\$P_UIFR[3] = \$P_UIFR[4] : \$P_UIFR[5]	; Zuweisung des Ergebnis-Frames aus der Verkettung der ; beiden Einstellbaren Datenhaltungsframes an ei- nen ; anderen Einstellbaren Datenhaltungsframe

6.2.4 Definition von Framevariablen (DEF FRAME)

Neben den vordefinierten Framevariablen können auch eigene Framevariablen definiert werden. Die selbstdefinierten Framevariablen sind Anwendervariablen vom Typ FRAME. Der Name des Frames kann im Rahmen der Regeln für Anwendervariablen frei vergeben werden.

Mit den Funktionen CTRANS, CROT, CSCALE, CMIRROR können die selbstdefinierten Framevariablen Werten zugewiesen werden.

Syntax

```
DEF FRAME <Name>
```

Bedeutung

DEF FRAME:	Anwendervariablen vom Typ FRAME definieren.
<Name>:	Name der Framevariablen

Beispiel

Definition einer Framevariablen "PALETTE" und Zuweisung von Verschiebungs- und Drehungswerten:

Programmcode	Kommentar
DEF FRAME PALETTE	; Framevariable PALETTE definieren

Programmcode	Kommentar
PALETTE = CTRANS(...) : CROT(...)	; Zuweisung des Ergebnisframes der Verkettung von
	; Verschiebung und Drehung an die Framevariable
	PALETTE

6.3 Grob- und Feinverschiebung (CTRANS, CFINE)

Feinverschiebung

Eine Feinverschiebung `CFINE(...)` kann auf folgende Frames angewandt werden:

- Einstellbare Frames: `$P_UIFR` bzw. `$P_IFRAME`
- Basisframes: `$P_NCBFR[<n>]`, `$P_CHBFR[<n>]` bzw. `$P_CHBFRAMES[<n>]` oder `$P_ACTBFRAME`
- Programmierbares Frame: `$P_PFRAME`

Die Feinverschiebung eines Frames wird mit dem Befehl `CFINE(...)` programmiert.

Grobverschiebung

Die Grobverschiebung `CTRANS(...)` kann auf alle Frames angewandt werden.

Gesamtverschiebung

Die Gesamtverschiebung ergibt sich aus der Addition von Grob- und Feinverschiebung.

Maschinendaten

Freigabe der Feinverschiebung

Die Feinverschiebung wird freigegeben mit dem Maschinendatum:

MD18600 \$MN_MM_FRAME_FINE_TRANS = 1

Syntax

Feinverschiebung

- Gesamtframe
 - `<Frame> = CFINE(<K_1>, <Wert>)`
 - `<Frame> = CFINE(<K_1>, <Wert>, <K_2>, <Wert>)`
 - `<Frame> = CFINE(<K_1>, <Wert>, <K_2>, <Wert>, <K_3>, <Wert>)`
- Frame-Komponente
 - `<Frame>[<n>, <K_1>, FI] = <Wert>`

Grobverschiebung

• Gesamtframe

- $\langle \text{Frame} \rangle = \text{CTRANS}(\langle K_1 \rangle, \langle \text{Wert} \rangle)$
- $\langle \text{Frame} \rangle = \text{CTRANS}(\langle K_1 \rangle, \langle \text{Wert} \rangle, \langle K_2 \rangle, \langle \text{Wert} \rangle)$
- $\langle \text{Frame} \rangle = \text{CTRANS}(\langle K_1 \rangle, \langle \text{Wert} \rangle, \langle K_2 \rangle, \langle \text{Wert} \rangle, \langle K_3 \rangle, \langle \text{Wert} \rangle)$

• Frame-Komponente

- $\langle \text{Frame} \rangle[\langle n \rangle, \langle K_1 \rangle, \text{TR}] = \langle \text{Wert} \rangle$

Speziell für den Programmierbaren Frame $\$P_PFRAME$:

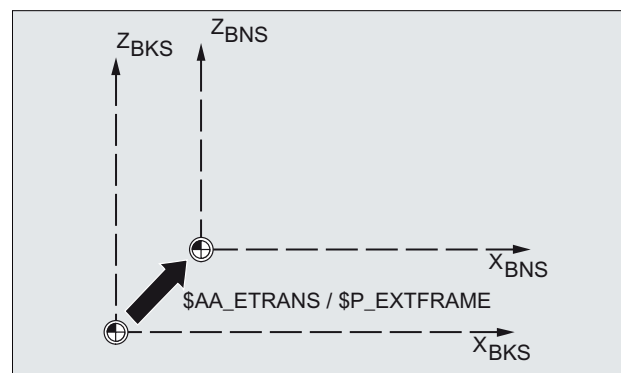
- $\text{TRANS} \langle K_1 \rangle \langle \text{Wert} \rangle$
- $\text{TRANS} \langle K_1 \rangle \langle \text{Wert} \rangle \langle K_2 \rangle \langle \text{Wert} \rangle$
- $\text{TRANS} \langle K_1 \rangle \langle \text{Wert} \rangle \langle K_2 \rangle \langle \text{Wert} \rangle \langle K_3 \rangle \langle \text{Wert} \rangle$

Bedeutung

$\langle \text{Frame} \rangle$:	Frame z.B. Einstellbarer Frame der Datenhaltung $\$P_UIFR[\langle n \rangle]$
CFINE:	Feinverschiebung, additive Verschiebung.
CTRANS:	Grobverschiebung, absolute Verschiebung
TRANS:	Nur Programmierbarer Frame: Grobverschiebung, absolute Verschiebung
$\langle K_n \rangle$:	Koordinatenachsen X, Y, Z
$\langle \text{Wert} \rangle$:	Verschiebungswert

6.4 Externe Nullpunktverschiebung (\$AA_ETRANS)

Die Externe Nullpunktverschiebung ist eine lineare Verschiebung zwischen Basiskoordinatensystem (BKS) und Basisnullpunktsystem (BNS).



6.4 Externe Nullpunktverschiebung (\$AA_ETRANS)

Die Externen Nullpunktverschiebung mittels \$AA_ETRANS wirkt, abhängig von der Maschinendaten-Parametrierung, auf zwei Arten:

1. Die Systemvariablen \$AA_ETRANS wirkt nach Aktivierung durch das NC/PLC-Nahtstellensignal direkt als Verschiebungswert
2. Der Wert der Systemvariablen \$AA_ETRANS wird nach Aktivierung durch das NC/PLC-Nahtstellensignal in die aktiven Systemframes \$P:EXTFRAME und den Datenhaltungsframe \$P_EXTFR übernommen. Anschließend wird der aktive Gesamtframe \$P_ACTFRAME neu berechnet.

Maschinendaten

Im Zusammenhang mit der Systemvariablen \$AA_ETRANS sind zwei Vorgehensweisen zu unterscheiden, die über das folgende Maschinendatum ausgewählt werden:

MD28082 \$MC_MM_SYSTEM_FRAME_MASK, Bit1 = <Wert>

<Wert>	Bedeutung
0	Funktion: Direktes Schreiben von \$AA_ETRANS[<Achse>] durch PLC, HMI oder NC-Programm. Freigabe zum Herausfahren der Nullpunktverschiebung von \$AA_ETRANS[<Achse>] im nächstmöglichen Verfahrsatz: DB31, ... DBX3.0
1	Funktion: Aktivierung des aktiven Systemframes \$P:EXTFRAME und des Datenhaltungsframes \$P_EXTFR Freigabe zum Herausfahren der Nullpunktverschiebung von \$AA_ETRANS[<Achse>] durch: DB31, ... DBX3.0. Daraufhin erfolgt im Kanal: <ul style="list-style-type: none"> • Stop aller Verfahrbewegungen im Kanal (außer Kommando- und PLC-Achsen) • Vorlaufstop mit anschließendem Reorganisieren (STOPRE) • Grobverschiebung aktiver Frame \$P_EXTFRAME[<Achse>] = \$AA_ETRANS[<Achse>] • Grobverschiebung Datenhaltungsframe \$P_EXTFR[<Achse>] = \$AA_ETRANS[<Achse>] • Neuberechnung des aktiven Gesamtframes \$P_ACTFRAME • Herausfahren der Verschiebung in den programmierten Achsen. • Fortsetzen der unterbrochenen Verfahrbewegung bzw. des NC-Programms

Programmierung

- Syntax
\$AA_ETRANS[<Achse>] = <Wert>
- Bedeutung

\$AA_ETRANS:	Systemvariable zum Zwischenspeichern der externen Nullpunktverschiebung
<Achse>:	Kanalachse
<Wert>:	Verschiebungswert

NC/PLC-Nahtstellensignal

DB31, ... DBX3.0 = 0 → 1 ⇒ \$P_EXTFRAME[<Achse>] = \$P_EXTFR[<Achse>] = \$AA_ETRANS[<Achse>]

6.5 Istwertsetzen mit Verlust des Referenzierstatus (PRESETON)

Die Prozedur `PRESETON()` setzt für eine oder mehrere Achsen einen neuen Istwert im Maschinenkoordinatensystem (MKS). Dies entspricht einer Nullpunktverschiebung des MKS der Achse. Die Achse wird dadurch nicht verfahren.

Durch `PRESETON` wird ein Vorlaufstopp mit Synchronisation ausgelöst. Die Istposition wird der Achse erst im Stillstand zugewiesen.

Ist die Achse bei `PRESETON` dem Kanal nicht zugeordnet, ist das weitere Vorgehen abhängig von der achsspezifischen Projektierung des Achstausverhaltens:

MD30552 \$MA_AUTO_GET_TYPE


Referenzierstatus

Durch das Setzen eines neuen Istwertes im Maschinenkoordinatensystem wird der Referenzierstatus der Maschinenachse zurückgesetzt:

DB31, ... DBX60.4 / .5 = 0 (Referenziert / Synchronisiert Messsystem 1 / 2)

Es wird empfohlen, `PRESETON` nur bei Achsen ohne Referenzpunktspflicht zu verwenden.

Zum Wiederherstellen des ursprünglichen Maschinenkoordinatensystems muss das Messsystem der Maschinenachse, z.B. durch Referenzpunktfahren aus dem Teileprogramm (G74), erneut referenziert werden.

 VORSICHT
Verlust des Referenzierstatus
Durch das Setzen eines neuen Istwertes im Maschinenkoordinatensystem mit <code>PRESETON</code> wird der Referenzierstatus der Maschinenachse auf "nicht referenziert / synchronisiert" zurückgesetzt.

Programmierung

Syntax

`PRESETON(<Achse_1>, <Wert_1> [, <Achse_2>, <Wert_2>, ... <Achse_8>, <Wert_8>])`

Bedeutung

PRESETON:	Istwertsetzen mit Verlust des Referenzierstatus	
	Vorlaufstopp:	ja
	Alleine im Satz:	ja
<Achse_x>:	Maschinenachsname	
	Typ:	AXIS
	Wertebereich:	im Kanal definierte Maschinenachsenamen
<Wert_x>:	Neuer Istwert der Maschinenachse im Maschinenkoordinatensystem (MKS) Die Eingabe erfolgt im aktuell gültigen Maßsystem (inch / metrisch) Eine aktive Durchmesserprogrammierung (<code>DIAMON</code>) wird berücksichtigt	
	Typ:	REAL

Literatur

PRESETONS in NC-Programmen

Eine ausführliche Beschreibung von PRESETON in NC-Programmen findet sich in:

Funktionshandbuch Grundfunktionen; Kapitel "K2: Achsen, Koordinatensysteme, Frames" > "Koordinatensysteme" > "Maschinenkoordinatensystem (MKS)" > "Istwertsetzen mit Verlust des Referenzierstatus (PRESETON)"

PRESETONS in Synchronaktionen

Eine ausführliche Beschreibung von PRESETON in Synchronaktionen findet sich in:

Funktionshandbuch Synchronaktionen; Kapitel: "Ausführliche Beschreibng" > "Aktionen in Synchronaktionen" > "Istwertsetzen mit Verlust des Referenzierstatus (PRESETON)"

6.6 Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS)

Die Prozedur PRESETONS () setzt für eine oder mehrere Achsen einen neuen Istwert im Maschinenkoordinatensystem (MKS). Dies entspricht einer Nullpunktverschiebung des MKS der Achse. Die Achse wird dadurch nicht verfahren.

Durch PRESETONS wird ein Vorlaufstop mit Synchronisation ausgelöst. Die Istposition wird der Achse erst im Stillstand zugewiesen.

Ist die Achse bei PRESETONS dem Kanal nicht zugeordnet, ist das weitere Vorgehen abhängig von der achsspezifischen Projektierung des Achstauschverhaltens:

MD30552 \$MA_AUTO_GET_TYPE

Referenzierstatus

Durch das Setzen eines neuen Istwerts im Maschinenkoordinatensystem (MKS) mit PRESETONS wird der Referenzierstatus der Maschinenachse **nicht** verändert.

Voraussetzungen

- **Gebertyp**

PRESETONS ist nur bei folgenden Gebertypen des aktiven Messsystems möglich:

- MD30240 \$MA_ENC_TYPE[<Messsystem>] = 0 (Simulierter Geber)
- MD30240 \$MA_ENC_TYPE[<Messsystem>] = 1 (Rohsignalgeber)

- **Referenzier-Mode**

PRESETONS ist nur bei folgenden Referenzier-Modes des aktiven Messsystems möglich:

- MD34200 \$MA_ENC_REFP_MODE[<Messsystem>] = 0 (kein Referenzpunktfahren möglich)
- MD34200 \$MA_ENC_REFP_MODE[<Messsystem>] = 1 (Referenzieren von inkrementellen, rotatorischen oder linearen Messsystemen: Nullimpuls auf der Geberspur)

Programmierung

Syntax

```
PRESETONS (<Achse_1>, <Wert_1> [, <Achse_2>, <Wert_2>, ... <Achse_8>, <Wert_8>])
```

Bedeutung

PRESETONS:	Istwertsetzen ohne Verlust des Referenzierstatus	
	Vorlaufstopp:	ja
	Alleine im Satz:	ja
<Achse_x>:	Maschinenachsname	
	Typ:	AXIS
	Wertebereich:	im Kanal definierte Maschinenachsenamen
<Wert_x>:	Neuer aktueller Istwert der Maschinenachse im Maschinenkoordinatensystem (MKS)	
	Die Eingabe erfolgt im aktiven Maßsystem (inch / metrisch)	
	Eine aktive Durchmesserprogrammierung (DIAMON) wird berücksichtigt	
	Typ:	REAL

Literatur

PRESETONS in NC-Programmen

Eine ausführliche Beschreibung von PRESETONS in NC-Programmen findet sich in:

Funktionshandbuch Grundfunktionen; Kapitel "K2: Achsen, Koordinatensysteme, Frames" > "Koordinatensysteme" > "Maschinenkoordinatensystem (MKS)" > "Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS)"

PRESETONS in Synchronaktionen

Eine ausführliche Beschreibung von PRESETONS in Synchronaktionen findet sich in:

Funktionshandbuch Synchronaktionen; Kapitel: "Ausführliche Beschreibung" > "Aktionen in Synchronaktionen" > "Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS)"

6.7 Frame-Berechnung aus 3 Messpunkten im Raum (MEAFRAME)

Die Funktion MEAFRAME wird zur Unterstützung der Messzyklen eingesetzt. Sie berechnet den Frame aus drei idealen und den korrespondierenden gemessenen Punkten.

Wird ein Werkstück für die Bearbeitung positioniert, ist seine Position relativ zum kartesischen Maschinenkoordinatensystem bezüglich seiner Idealposition i. A. sowohl verschoben als auch gedreht. Für exakte Bearbeitung oder Messung ist entweder eine kostspielige physikalische Justierung oder Änderung der Bewegungen im Teileprogramm nötig.

Ein Frame kann durch Abtasten dreier Punkte im Raum festgelegt werden, deren Idealpositionen bekannt sind. Abgetastet wird mit einem Berührungs- oder optischen Sensor, der spezielle, auf der Trägerplatte präzise fixierte Löcher oder Messkugeln berührt.

Syntax

MEAFRAME (<Ideal-Punkte>, <Messpunkte>, <Qualität>)

Bedeutung

MEAFRAME:	Funktionsaufruf		
<Ideal-Punkte>:	2-dim. REAL-Feld, das die drei Koordinaten der Ideal-Punkte enthält		
<Messpunkte>:	2-dim. REAL-Feld, das die drei Koordinaten der gemessenen Punkte enthält		
<Qualität>:	Variable, mit der Informationen zur Qualität der FRAME-Berechnung zurückgegeben werden		
	Typ:	VAR REAL	
	Wert:	-1	Die idealen Punkte liegen nahezu auf einer Geraden: Der Frame konnte nicht berechnet werden. Die zurückgegebene FRAME-Variable enthält einen neutralen Frame.
		-2	Die Messpunkte liegen nahezu auf einer Geraden: Der Frame konnte nicht berechnet werden. Die zurückgegebene FRAME-Variable enthält einen neutralen Frame.
		-4	Die Berechnung der Rotationsmatrix schlägt aus einem anderen Grund fehl.
≥ 0.0		Summe der Verzerrungen (Abstände zwischen den Punkten), die zur Überführung des gemessenen Dreiecks in ein zum idealen Dreieck kongruentes benötigt wird.	

Hinweis

Qualität der Messung

Damit die gemessenen den idealen Koordinaten mit einer kombinierten Rotation/Translation zugeordnet werden können, muss das von den Messpunkten aufgespannte Dreieck kongruent zum idealen Dreieck sein. Dies wird bewerkstelligt von einem Kompensationsalgorithmus, der die Summe der Quadrate der Abweichungen minimiert, die das gemessene in das ideale Dreieck überführen.

Die effektiv benötigte Verzerrung der Messpunkte kann als Indikator für die Qualität der Messung dienen und wird deshalb als zusätzliche Variable von MEAFRAME ausgegeben.

Hinweis

Das von MEAFRAME erzeugte Frame kann durch die Funktion ADDFRAME in ein anderes Frame in der Frame-Kette transformiert werden (siehe Beispiel "Verkettung mit ADDFRAME").

Beispiele

Beispiel 1:

Teileprogramm 1:

Programmcode

```
...
DEF FRAME CORR_FRAME
```

Setzen von Messpunkten:

Programmcode

Kommentar

```
DEF REAL IDEAL_POINT[3,3]=
SET(10.0,0.0,0.0,0.0,10.0,0.0,0.0,0.0,10.0)

DEF REAL MEAS_POINT[3,3]=
SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8)
; Für Test.

DEF REAL FIT_QUALITY=0
DEF REAL ROT_FRAME_LIMIT=5
; Erlaubt max. 5 Grad-Verdrehung
der Teileposition.

DEF REAL FIT_QUALITY_LIMIT=3
; Erlaubt max. 3 mm-Verschiebung
zwischen
dem idealen und dem gemessenen
Dreieck.

DEF REAL SHOW_MCS_POS1[3]
DEF REAL SHOW_MCS_POS2[3]
DEF REAL SHOW_MCS_POS3[3]
```

Programmcode

Kommentar

```
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
SETAL(65020)
GOTOF NO_FRAME
ENDIF
; Begrenzung des 1. RPY- Winkels.
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT
; Begrenzung des 2. RPY-Winkels.
```

6.7 Frame-Berechnung aus 3 Messpunkten im Raum (MEAFRAME)

Programmcode	Kommentar
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; Begrenzung des 3. RPY- Winkels.
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 \$P_IFRAME=CORR_FRAME	; Abtast-Frame mit einem setzbaren Frame aktivieren.
	; Frame prüfen durch Positionieren der Geometrieachsen auf die idealen Punkte.
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=\$AA_IM[X]	
N420 SHOW_MCS_POS1[1]=\$AA_IM[Y]	
N430 SHOW_MCS_POS1[2]=\$AA_IM[Z]	
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]	
N510 SHOW_MCS_POS2[0]=\$AA_IM[X]	
N520 SHOW_MCS_POS2[1]=\$AA_IM[Y]	
N530 SHOW_MCS_POS2[2]=\$AA_IM[Z]	
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]	
N610 SHOW_MCS_POS3[0]=\$AA_IM[X]	
N620 SHOW_MCS_POS3[1]=\$AA_IM[Y]	
N630 SHOW_MCS_POS3[2]=\$AA_IM[Z]	
N700 G500	; Setzbaren Frame deaktivieren, da mit Nullframe (kein Wert eingetragen) vorbesetzt).
No_FRAME	; Setzbaren Frame deaktivieren, da mit Nullframe (kein Wert eingetragen) vorbesetzt).
M0	
M30	

Beispiel 2: Verkettung von Frames

Verkettung von MEAFRAME für Korrekturen

Die Funktion MEAFRAME liefert ein Korrektur-Frame. Wird dieser Korrektur-Frame mit dem einstellbaren Frame \$P_UIFR[1] verkettet, der bei Aufruf der Funktion aktiv war (z. B. G54), so erhält man einen einstellbaren Frame für weitere Umrechnungen zum Verfahren oder Bearbeiten.

Verkettung mit ADDFRAME

Soll dieser Korrektur-Frame in der Frame-Kette an einer anderen Stelle wirken oder sind vor dem einstellbaren Frame noch andere Frames aktiv, dann kann die Funktion ADDFRAME zum Einketten in einem der Kanal-Basisframes oder einem Systemframe genutzt werden.

In den Frames darf hierbei nicht aktiv sein:

- Spiegelung mit MIRROR
- Skalierung mit SCALE

Die Eingangsparameter für Soll- und Istwerte sind die Werkstückkoordinaten. Im Grundsystem der Steuerung sind diese Koordinaten stets metrisch oder in Inch (G71/G70) und als radiusbezogenes (DIAMOF) Maß anzugeben.

Literatur:

Weitere Informationen zu ADDFRAME siehe:

Funktionshandbuch Grundfunktionen; K2: Achsen, Koordinatensysteme, Frames

6.8 NCU-globale Frames

NCU-globale Frames gibt es pro NCU nur einmal für alle Kanäle. NCU-globale Frames können von allen Kanälen aus geschrieben und gelesen werden. Die Aktivierung der NCU-globalen Frames erfolgt im jeweiligen Kanal.

Durch globale Frames können **Kanalachsen und Maschinenachsen** mit Verschiebungen, skaliert und gespiegelt werden.

Geometrische Zusammenhänge und Frameketten

Bei globalen Frames existiert kein geometrischer Zusammenhang zwischen den Achsen. Deshalb können keine Drehungen und keine Programmierung von Geometrie-Achsbezeichnern ausgeführt werden.

- Auf globale Frames lassen sich keine Rotationen anwenden. Die Programmierung einer Rotation wird mit dem Alarm: "18310 Kanal %1 Satz %2 Frame: Rotation unzulässig", abgelehnt.
- Die Verkettung von globalen Frames und kanalspezifischen Frames ist möglich. Der resultierende Frame enthält alle Frameanteile inklusive der Rotationen für alle Achsen. Die Zuweisung eines Frames mit Rotationsanteilen an einen globalen Frame wird mit dem Alarm "Frame: Rotation unzulässig" abgelehnt.

NCU-globale Frames

NCU-globale Basisframes \$P_NCBFR[n]

Es können bis zu 8 NCU-globale Basisframes projiziert werden:

Gleichzeitig können kanalspezifische Basisframes vorhanden sein.

Globale Frames können von allen Kanälen einer NCU geschrieben und gelesen werden. Beim Schreiben von globalen Frames ist vom Anwender für eine Kanalkoordinierung Sorge zu tragen. Dies kann z. B. durch Wait-Marken (WAITMC) realisiert werden.

Maschinenhersteller

Die Anzahl von globalen Basisframes wird über Maschinendaten projiziert.

Literatur:

Funktionshandbuch Grundfunktionen; Achsen, Koordinatensysteme, Frames (K2)

NCU-globale einstellbare Frames \$P_UIFR[n]

Alle einstellbaren Frames G500, G54...G599 können entweder NCU-global oder kanalspezifisch projiziert werden.

Maschinenhersteller

Alle einstellbaren Frames können mit Hilfe des Maschinendatums MD18601 \$MN_MM_NUM_GLOBAL_USER_FRAMES zu globalen Frames umprojiziert werden.

Als Achsbezeichner bei den Frame-Programmbefehlen können Kanalachsbezeichner und Maschinenachsbezeichner verwendet werden. Die Programmierung von Geometrieachsbezeichnern wird mit einem Alarm abgelehnt.

6.8.1 Kanalspezifische Frames (\$P_CHBFR, \$P_UBFR)

Einstellbare Frames oder Basisframes können über das Teileprogramm und über BTSS von der Bedienung und von der PLC geschrieben und gelesen werden.

Die Feinverschiebung ist auch für die globalen Frames möglich. Die Unterdrückung von globalen Frames erfolgt ebenso, wie bei kanalspezifischen Frames über G53, G153, SUPA und G500.

Maschinenhersteller

Über das Maschinendatum MD28081 \$MC_MM_NUM_BASE_FRAMES kann die Anzahl der Basisframes im Kanal projiziert werden. Die Standardkonfiguration ist so ausgelegt, dass es mindestens ein Basisframe pro Kanal gibt. Maximal sind 8 Basisframes pro Kanal möglich. Zusätzlich zu den 8 Basisframes im Kanal kann es noch 8 NCU-globale Basisframes geben.

Kanalspezifische Frames

\$P_CHBFR[n]

Über die Systemvariable \$P_CHBFR[n] können die Basisframes gelesen und geschrieben werden. Beim Schreiben eines Basisframes wird der verkettete Gesamt-Basisframe nicht aktiviert, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54 ... G599-Anweisung. Die Variable dient vorwiegend als Speicher für Schreibvorgänge auf das Basisframe von HMI oder PLC. Diese Frame-Variablen werden über die Datensicherung gesichert.

Erster Basisframe im Kanal

Ein Schreiben auf die vordefinierte Variable \$P_UBFR aktiviert den Basisframe mit dem Feldindex 0 nicht gleichzeitig, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54 ... G599-Anweisung. Die Variable kann auch im Programm geschrieben und gelesen werden.

\$P_UBFR

\$P_UBFR ist identisch mit \$P_CHBFR[0]. Standardmäßig gibt es immer einen Basisframe im Kanal, so dass die Systemvariable kompatibel zu älteren Ständen ist. Gibt es kein kanalspezifisches Basisframe, wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.

6.8.2 Im Kanal wirksame Frames

Im Kanal wirksame Frames werden vom Teileprogramm über die betreffenden Systemvariablen dieser Frames eingegeben. Hierzu gehören auch Systemframes. Über diese Systemvariablen kann im Teileprogramm das aktuelle Systemframe gelesen und geschrieben werden.

Aktuelle im Kanal wirksame Frames

Übersicht

Aktuelle Systemframes	für:
\$P_PARTFRAME	TCARR und PAROT
\$P_SETFRAME	Istwertsetzen und Ankratzen
\$P_EXTFRAME	Externe Nullpunktverschiebung
\$P_NCBFRAME[n]	Aktuelle NCU-globale Basisframes
\$P_CHBFRAME[n]	Aktuelle Kanal-Basisframes
\$P_BFRAME	Aktueller 1. Basisframe im Kanal
\$P_ACTBFRAME	Gesamt-Basisframe
\$P_CHBFRMASK und \$P_NCBFRMASK	Gesamt-Basisframe
\$P_IFFRAME	Aktueller einstellbarer Frame
Aktuelle Systemframes	für:
\$P_TOOLFRAME	TOROT und TOFRAME
\$P_WPFRAME	Werkstückbezugspunkte
\$P_TRAFRAME	Transformationen
\$P_PFRAME	Aktueller programmierbarer Frame
Aktuelles Systemframe	für:
\$P_CYCFRAME	Zyklen
P_ACTFRAME	Aktueller Gesamtframe
FRAME-Kettung	Aktuelles Frame setzt sich aus dem Gesamt-Basisframe zusammen

\$P_NCBFRAME [n] Aktuelle NCU-globale Basisframes

Über die Systemvariable \$P_NCBFRAME[n] können die aktuellen globalen Basisframe-Feldelemente gelesen und geschrieben werden. Das resultierende Gesamt-Basisframe wird durch den Schreibvorgang im Kanal eingerechnet.

Der geänderte Frame wird nur in dem Kanal, in dem der Frame programmiert wurde, aktiv. Soll der Frame für alle Kanäle einer NCU geändert werden, muss gleichzeitig \$P_NCBFR[n] und \$P_NCBFRAME[n] beschrieben werden. Die anderen Kanäle müssen dann noch den Frame mit z. B. G54 aktivieren. Beim Schreiben eines Basisframes wird der Gesamt-Basisframe neu berechnet.

\$P_CHBFRAME[n] Aktuelle Kanal-Basisframes

Über die Systemvariable \$P_CHBFRAME[n] können die aktuellen Kanal-Basisframe-Feldelemente gelesen und geschrieben werden. Der resultierende Gesamt-Basisframe wird

durch den Schreibvorgang im Kanal eingerechnet. Beim Schreiben eines Basisframes wird der Gesamt-Basisframe neu berechnet.

\$P_BFRAME Aktueller 1. Basisframe im Kanal

Über die vordefinierte Framevariable \$P_BFRAME kann der aktuelle Basisframe mit dem Feldindex 0, der im Kanal gültig ist, im Teileprogramm gelesen und geschrieben werden. Der geschriebene Basisframe wird sofort eingerechnet.

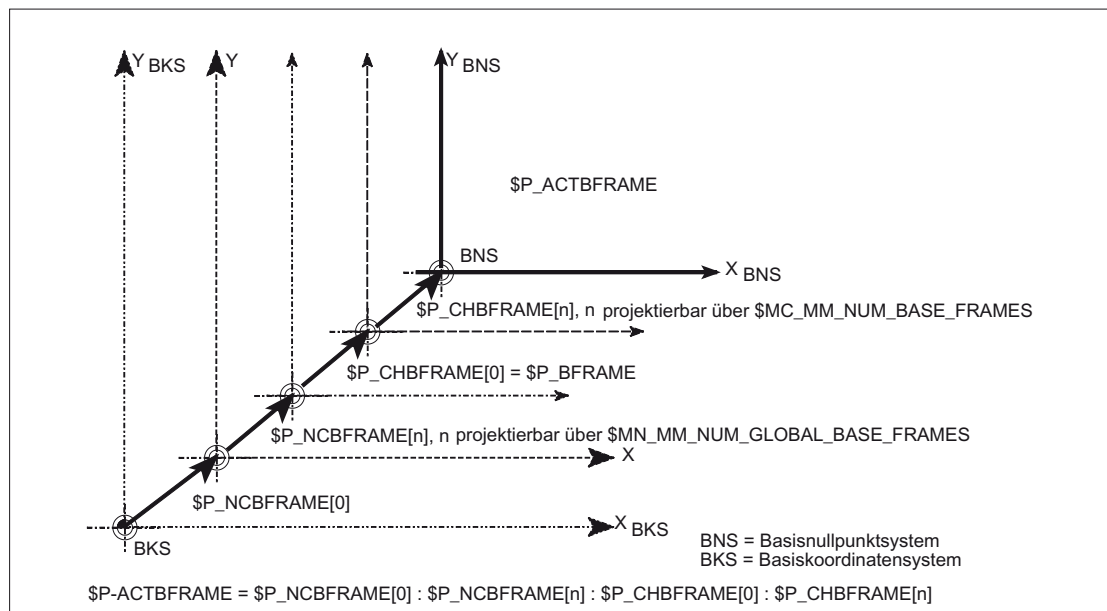
\$P_BFRAME ist identisch mit \$P_CHBFRAME[0]. Die Systemvariable hat standardmäßig immer einen gültigen Wert. Gibt es kein kanalspezifisches Basisframe, wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.

\$P_ACTBFRAME Gesamt-Basisframe

Die Variable \$P_ACTBFRAME ermittelt das verkettete Gesamt-Basisframe. Die Variable ist nur lesbar.

\$P_ACTBFRAME entspricht:

\$P_NCBFRAME[0] : ... : \$P_NCBFRAME[n] : \$P_CHBFRAME[0] : ... : \$P_CHBFRAME[n].



\$P_CHBFRMASK und \$P_NCBFRMASK Gesamt-Basisframe

Über die Systemvariable \$P_CHBFRMASK und \$P_NCBFRMASK kann der Anwender auswählen, welche Basisframes er in die Berechnung des "Gesamt"-Basisframes mit einbeziehen möchte. Die Variablen können nur im Programm programmiert werden und über BTSS gelesen werden. Der Wert der Variablen wird als Bitmaske interpretiert und gibt an, welches Basisframe-Feldelement von \$P_ACTBFRAME in die Berechnung einfließt.

Mit \$P_CHBFRMASK kann vorgegeben werden, welche kanalspezifischen Basisframes, und mit \$P_NCBFRMASK, welche NCU-globalen Basisframes eingerechnet werden.

Mit der Programmierung der Variablen werden der Gesamt-Basisframe und der Gesamt-Frame neu berechnet. Nach Reset und in der Grundeinstellung ist der Wert von \$P_CHBFRMASK und \$P_NCBFRMASK wie folgt:

```
$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK
```

```
$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK
```

Beispiel:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
```

```
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

\$P_IFRAME Aktueller einstellbarer Frame

Über die vordefinierte Framevariable \$P_IFRAME kann der aktuelle einstellbare Frame, welcher im Kanal gültig ist, im Teileprogramm gelesen und geschrieben werden. Der geschriebene einstellbare Frame wird sofort eingerechnet.

Bei NCU-globalen einstellbaren Frames wirkt der geänderte Frame nur in dem Kanal, in dem der Frame programmiert wurde. Soll der Frame für alle Kanäle einer NCU geändert werden, muss gleichzeitig \$P_UIFR[n] und \$P_IFRAME beschrieben werden. Die anderen Kanäle müssen dann noch den entsprechenden Frame mit z. B. G54 aktivieren.

\$P_PFRAME Aktueller programmierbarer Frame

\$P_PFRAME ist der programmierbare Frame, der sich aus der Programmierung von TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR bzw. aus der Zuweisung von CTRANS, CROT, CMIRROR, CSCALE an den programmierbaren FRAME ergibt.

Aktuelle, programmierbare Framevariable, die den Bezug zwischen dem einstellbaren Nullpunktsystem (ENS) und dem Werkstückkoordinatensystem (WKS) herstellt.

P_ACTFRAME Aktueller Gesamtframe

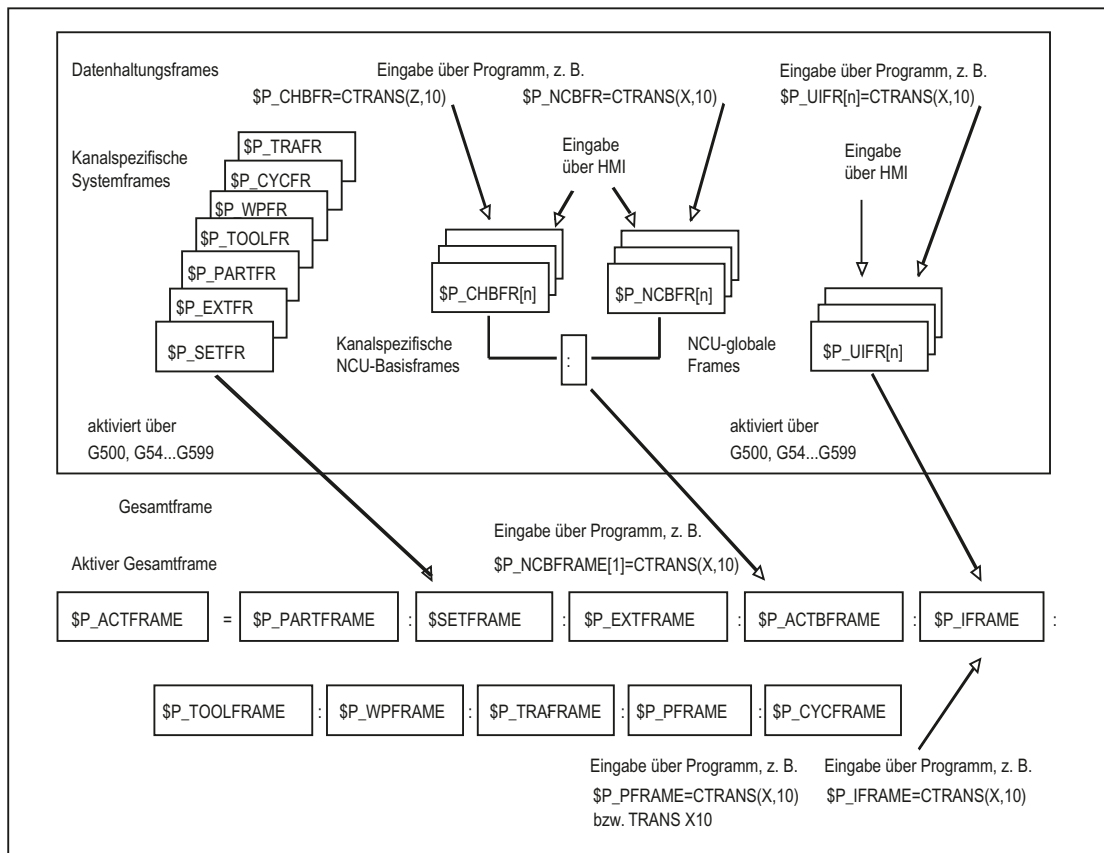
Der aktuelle resultierende Gesamtframe \$P_ACTFRAME ergibt sich nun als Verkettung aller Basisframes, dem aktuellen einstellbaren Frame und dem programmierbaren Frame. Der aktuelle Frame wird immer dann aktualisiert, wenn sich ein Frameanteil ändert.

\$P_ACTFRAME entspricht:

```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :
```

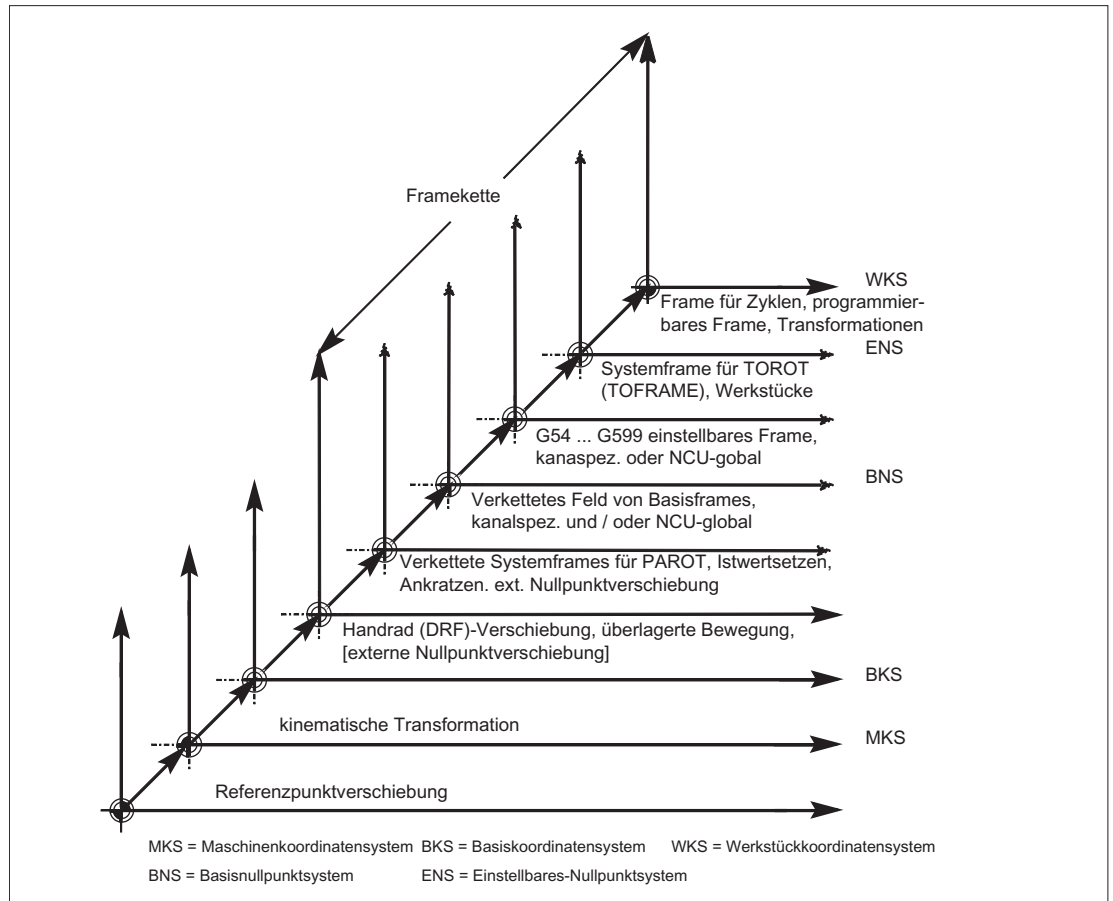
```
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```

6.8 NCU-globale Frames



Frame-Kettung

Der aktuelle Frame setzt sich aus dem Gesamt-Basisframe, dem einstellbaren Frame, dem Systemframe und dem programmierbaren Frame gemäß oben angegebenen aktuellen Gesamtframe zusammen.



Transformationen

7.1 Allgemeine Programmierung der Transformationsarten

Allgemeine Funktion

Zur Anpassung der Steuerung an verschiedene Maschinenkinematiken besteht die Auswahl Transformationsarten mit geeigneten Parametern zu programmieren. Über diese Parameter kann für die ausgewählte Transformation sowohl die Orientierung des Werkzeugs im Raum als auch die Orientierungsbewegungen der Rundachsen entsprechend vereinbart werden.

Bei den Drei-, Vier- und Fünf-Achs-Transformationen beziehen sich die programmierten, Positionsangaben immer auf die Spitze des Werkzeugs, welches orthogonal zur im Raum befindlichen Bearbeitungsfläche nachgeführt wird. Die kartesischen Koordinaten werden vom Basiskoordinatensystem ins Maschinenkoordinatensystem umgerechnet und beziehen sich auf die Geometrieachsen. Diese beschreiben den Arbeitspunkt. Virtuelle Rundachsen beschreiben die Orientierungen des Werkzeugs im Raum und werden mit TRAORI programmiert.

Bei der kinematischen Transformation können Positionen im kartesischen Koordinatensystem programmiert werden. Die Steuerung transformiert die mit TRANSMIT, TRACYL und TRAANG programmierten Verfahrbewegungen des kartesischen Koordinatensystems auf die Verfahrbewegungen der realen Maschinenachsen.

Programmierung

Drei-, Vier- und Fünf-Achs-Transformationen TRAORI

Die vereinbarte Orientierungstransformation wird mit dem Befehl TRAORI und den drei möglichen Parametern für Trafonummer, Orientierungsvektor und Rundachsoffsets aktiviert.

```
TRAORI(Trafonummer, Orientierungsvektor, Rundachsoffsets)
```

Kinematische Transformationen

Zu den Kinematischen Transformationen gehören die vereinbarten Transformationen

```
TRANSMIT(Trafonummer)
```

```
TRACYL(Arbeitsdurchmesser, Trafonummer)
```

```
TRAANG(Winkel der schräg stehenden Achse, Trafonummer)
```

Aktive Transformation ausschalten

Mit TRAF00F kann die gerade aktive Transformation ausgeschaltet werden.

Orientierungstransformation

Drei-, Vier- und Fünf- Achs-Transformationen TRAORI

Zur optimalen Bearbeitung räumlich geformter Flächen im Arbeitsraum der Maschine, benötigen Werkzeugmaschinen außer den drei Linearachsen X, Y und Z noch zusätzliche

Achsen. Die zusätzlichen Achsen beschreiben die Orientierung im Raum und werden nachfolgend Orientierungsachsen genannt. Sie stehen als Drehachsen bei vier Maschinentypen mit verschiedener Kinematik zur Verfügung.

1. Zweiachsen-Schwenkkopf, z. B. Kardanischer Werkzeugkopf mit einer Rundachse parallel zu einer Linearachse bei festem Werkzeugschwenktisch.
2. Zweiachsen-Drehtisch, z. B. fester Schwenkkopf mit drehbarem Werkzeugschwenktisch um zwei Achsen.
3. Einachs-Schwenkkopf und Einachs-Drehtisch, z. B. ein drehbarer Schwenkkopf mit gedrehtem Werkzeug bei drehbarem Werkzeugschwenktisch um eine Achse.
4. Zweiachsen-Schwenkkopf und Einachs-Drehtisch, z. B. bei drehbarem Werkzeugschwenktisch um eine Achse und ein drehbarer Schwenkkopf mit drehbarem Werkzeug um sich selbst.

Die **3- und 4-Achs-Transformationen** sind Sonderformen der 5-Achs-Transformation und werden analog zu den 5-Achs-Transformationen programmiert.

Die "**Generische 3-/4-/5-/6-Achs-Transformation**" deckt mit ihrem Funktionsumfang für rechtwinklig angeordnete Rundachsen sowie die Transformationen für den Kardanischen Fräskopf ab und kann wie jede andere Orientierungstransformation auch für diese vier Maschinentypen mit TRAORI aktiviert werden. Bei der generischen 5/6-Achs-Transformation hat die Werkzeugorientierung einen weiteren dritten Freiheitsgrad, bei dem zur Werkzeugrichtung beliebig im Raum, das Werkzeug um die eigene Achse gedreht werden kann.

Literatur: /FB3/ Funktionshandbuch Sonderfunktionen; 3- bis 5-Achstransformation (F2)

Kinematikunabhängige Grundstellung der Werkzeugorientierung

ORIRESET

Ist mit TRAORI eine Orientierungstransformation aktiv, dann können mit ORIRESET die Grundstellungen von bis zu 3 Orientierungsachsen mit optionalen Parametern A, B, C angegeben werden. Die Zuordnung der Reihenfolge der programmierten Parameter zu den Rundachsen erfolgt gemäß der durch die Transformation festgelegten Reihenfolge der Orientierungsachsen. Die Programmierung von ORIRESET(A, B, C) bewirkt, dass die Orientierungsachsen linear und synchron von ihrer momentanen Position zu der angegebenen Grundstellungsposition fahren.

Kinematische Transformationen

TRANSMIT und TRACYL

Bei Fräsbearbeitungen an Drehmaschinen kann für die vereinbarte Transformation entweder

1. eine stirnseitige Bearbeitung in der Drehaufspannung mit TRANSMIT oder
2. eine Bearbeitung von beliebig verlaufenden Nuten an zylindrischen Körpern mit TRACYL programmiert werden.

TRAANG

Soll die Zustellachse z. B. für die Technologie Schleifen auch schräg zustellbar sein, so kann mit TRAANG für die vereinbarte Transformation ein parametrierbarer Winkel programmiert werden.

Kartesisches PTP-Fahren

Zur kinematischen Transformation gehört auch das "Kartesische PTP-Fahren" bei dem bis zu 8 unterschiedliche Gelenkstellungen STAT= programmiert werden können. Die Positionen werden im kartesischen Koordinatensystem programmiert, wobei die Bewegung der Maschine in Maschinenkoordinaten erfolgt.

Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformation (M1)

Verkettete Transformationen

Es können jeweils zwei Transformationen hintereinander geschaltet werden. Bei der hierdurch verketteten zweiten Transformation werden die Bewegungsanteile der Achsen aus der ersten Transformation übernommen.

Als erste Transformation sind möglich:

- Orientierungstransformation TRAORI
- Polartransformation TRANSMIT
- Zylindertransformation TRACYL
- Transformation Schräge Achse TRAANG

Die zweite Transformation muss Schräge Achse TRAANG sein

7.1.1 Orientierungsbewegungen bei den Transformationen

Verfahrbewegungen und Orientierungsbewegungen

Die Verfahrbewegungen der programmierbaren Orientierungen hängen primär vom Maschinentyp ab. Bei der Drei-, Vier- und Fünf-Achs-Transformation mit TRAORI beschreiben die rotatorischen Achsen oder die schwenkbaren Linearachsen die Orientierungsbewegungen des Werkzeugs.

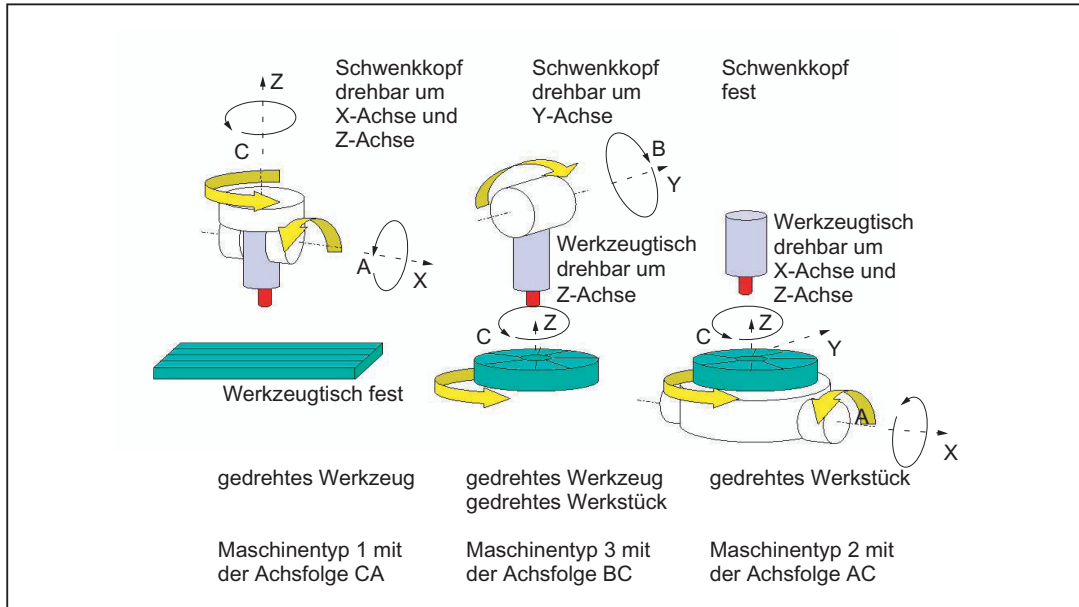
Änderungen der Positionen der an der Orientierungstransformation beteiligten Rundachsen führen zu Ausgleichsbewegungen der übrigen Maschinenachsen. Die Position der Werkzeugspitze bleibt dabei unverändert.

Orientierungsbewegungen des Werkzeugs können über die Rundachsbezeichner A..., B..., C... der virtuellen Achsen je nach Anwendung entweder durch Angabe von Euler- bzw. RPY-Winkeln oder Richtungs- bzw. Flächennormalenvektoren, Normierte Vektoren für die Drehachse eines Kegels oder für die Zwischenorientierung auf einer Kegelmantelfläche programmiert werden.

Bei der Kinematischen Transformation mit TRANSMIT, TRACYL und TRAANG transformiert die Steuerung die programmierten Verfahrbewegungen des kartesischen Koordinatensystems auf die Verfahrbewegungen der realen Maschinenachsen.

Maschinenkinematik bei Drei-, Vier- und Fünf-Achs-Transformation TRAORI

Es kann entweder das Werkzeug oder der Werkzeuggestisch mit bis zu zwei Rundachsen drehbar sein. Eine Kombination von jeweils einachsigem Schwenkkopf und Drehtisch ist auch möglich.



Maschinentyp	Programmierung der Orientierung
Drei-Achs-Transformation Maschinentypen 1 und 2	Programmierung der Werkzeugorientierung nur in der Ebene, die senkrecht zu der rotatorischen Achse ist. Es existieren zwei translatorischen Achsen (Linearachsen) und einer rotatorischen Achse (Rundachse).
Vier-Achs-Transformation Maschinentypen 1 und 2	Programmierung der Werkzeugorientierung nur in der Ebene, die senkrecht zu der rotatorischen Achse ist. Es existieren drei translatorischen Achsen (Linearachsen) und einer rotatorischen Achse (Rundachse).
Fünf-Achs-Transformation Maschinentypen 3 Einachs-Schwenkkopf und Einachs-Drehtisch	Programmierung der Orientierungstransformation. Kinematik mit drei Linearachsen und zwei orthogonalen Rundachsen. Die Rundachsen sind parallel zu zwei der drei Linearachsen. Die erste Rundachse wird von zwei kartesischen Linearachsen bewegt. Sie dreht die dritte Linearachse mit dem Werkzeug. Die zweite Rundachse dreht das Werkstück.

Generische 5/6-Achs Transformationen

Maschinentyp	Programmierung der Orientierungstransformation
Generische Fünf-/Sechs-Achs Transformation Maschinentypen 4 Zweiachs-Schwenkkopf mit drehbarem Werkzeug um sich selbst und Einachs-Drehtisch	Programmierung der Orientierungstransformation. Kinematik mit drei Linearachsen und drei orthogonalen Rundachsen. Die Rundachsen sind parallel zu zwei der drei Linearachsen. Die erste Rundachse wird von zwei kartesischen Linearachsen bewegt. Sie dreht die dritte Linearachse mit dem Werkzeug. Die zweite Rundachse dreht das Werkstück. Die Grundorientierung des Werkzeugs kann durch eine zusätzliche Drehung um sich selbst mit dem Drehwinkel THETA programmiert werden.

Beim Aufruf der "Generischen Drei-, Vier- und Fünf-/Sechs-Achs Transformation" kann zusätzlich die Grundorientierung des Werkzeugs übergeben werden. Es gelten die Einschränkungen bezüglich der Richtungen der Rundachsen nicht mehr. Wenn die Rundachsen nicht exakt senkrecht aufeinander stehen oder vorhandene Rundachsen nicht exakt parallel zu den Linearachsen stehen, kann die "Generische Fünf-/Sechs-Achs Transformation" bessere Ergebnisse der Werkzeugorientierung liefern.

Kinematische Transformationen TRANSMIT, TRACYL und TRAANG

Für Fräsbearbeitungen an Drehmaschinen oder einer schräg zustellbaren Achse beim Schleifen gelten abhängig von der Transformation im Standardfall folgende Achsanordnungen:

TRANSMIT	Aktivierung der Polar-Transformation
stirnseitige Bearbeitung in der Drehaufspannung	eine Rundachse eine Zustellachse senkrecht zur Drehachse eine Längsachse parallel zur Drehachse
TRACYL	Aktivierung der Zylindermanteltransformation
Bearbeitung von beliebig verlaufenden Nuten an den zylindrischen Körper	eine Rundachse eine Zustellachse senkrecht zur Drehachse eine Längsachse parallel zur Drehachse
TRAANG	Aktivierung der Transformation Schräge Achse
Bearbeitung mit schräger Zustellachse	eine Rundachse eine Zustellachse mit parametrierbaren Winkel eine Längsachse parallel zur Drehachse

Kartesisches PTP-Fahren

Die Bewegung der Maschine erfolgt in Maschinenkoordinaten und wird programmiert mit:

TRAORI	Aktivierung der Transformation
PTP Punkt zu Punkt fahren CP	Position im kartesischen Koordinatensystem (MKS) anfahren Bahnbewegung der kartesischen Achsen im (BKS)

TRAORI	Aktivierung der Transformation
STAT	Stellung der Gelenke ist abhängig von der Transformation
TU	Um welchen Winkel die Achsen auf den kürzesten Weg verfahren

PTP-Fahren bei generischer 5/6-AchsTransformation

Die Bewegung der Maschine erfolgt in Maschinenkoordinaten und die Werkzeugorientierung kann sowohl mit Rundachspositionen als auch mit von der Kinematik unabhängigen Vektoren Euler bzw. RPY-Winkel oder den Richtungsvektoren programmiert werden.

Dabei sind Rundachsinterpolation, Vektoreninterpolation mit Großkreisinterpolation oder Interpolation des Orientierungsvektors auf einer Kegelmantelfläche möglich.

Beispiel Drei-, bis Fünf-Achs-Transformation bei einen Kardanischen Fräskopf

Die Werkzeugmaschine hat mindestens 5 Achsen, davon

- Drei translatorische Achsen für geradlinige Bewegungen, die den Arbeitspunkt an jede beliebige Position im Arbeitsraum bewegen.
- Zwei rotatorische Schwenkachsen, die unter einem projektierbaren Winkel (meist 45 Grad) angeordnet sind, ermöglichen dem Werkzeug Orientierungen im Raum einzunehmen, die sich bei 45 Grad Anordnung auf eine Halbkugel beschränken.

7.1.2 Übersicht der Orientierungstransformation TRAORI

Mögliche Programmierungsarten im Zusammenhang mit TRAORI

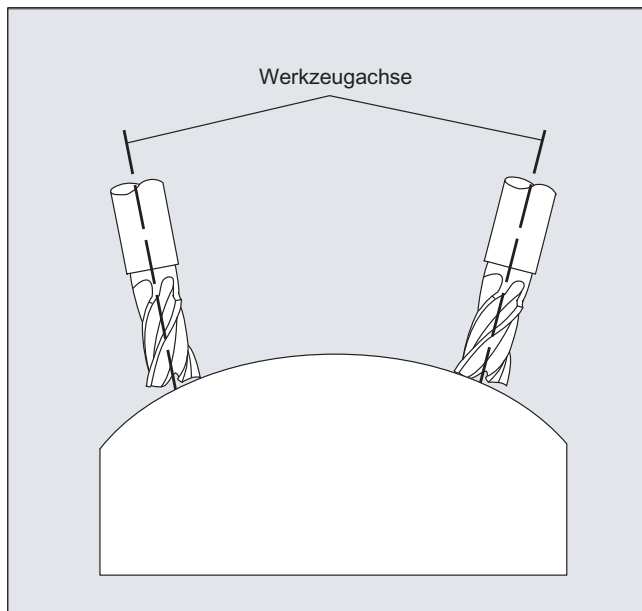
Maschinentyp	Programmierung bei aktiver Transformation TRAORI
<p>Maschinentypen 1, 2 oder 3 Zweiachs-Schwenkkopf oder Zweiachs-Drehtisch oder eine Kombination von jeweils ein- achsigen Schwenkkopf und Drehtisch.</p>	<p>Achsfolge der Orientierungsachsen und die Orientierungsrichtung des Werkzeugs ist und entweder maschinenbezogen projektierbar über Maschinendaten abhängig von der Maschinenkinematik oder werkstückbezogen mit programmierbarer Orientierung unabhängig von der Maschinenkinematik Die Drehrichtungen der Orientierungsachsen im Bezugssystem wird programmiert mit: - ORIMKS Bezugssystem = Maschinenkoordinatensystem - ORIWKS Bezugssystem = Werkstückkoordinatensystem Die Grundeinstellung ist ORIWKS. Programmierung der Orientierungsachsen mit: A, B, C der Maschinenachspalten direkt A2, B2, C2 Winkelprogrammierung virtueller Achsen mit - ORIEULER über Euler-Winkel (Standard) - ORIRPY über RPY-Winkel - ORIVIRT1 über virtuelle Orientierungsachsen 1. Definition - ORIVIRT2 über virtuelle Orientierungsachsen 2. Definition mit Unterscheidung der Interpolationsart: lineare Interpolation - ORIAxes von Orientierungsachsen oder Maschinenachsen Großkreisinterpolation (Interpolation des Orientierungsvektors) - ORIVECT von Orientierungsachsen Programmierung der Orientierungsachsen durch Angabe A3, B3, C3 der Vektorkomponenten (Richtung-/Flächennormale) Programmierung der resultierenden Werkzeugorientierung A4, B4, C4 des Flächennormalvektors am Satzbeginn A5, B5, C5 des Flächennormalvektors am Satzende LEAD Voreilwinkel für die Werkzeugorientierung TILT Seitwärtswinkel für die Werkzeugorientierung</p>
	<p>Interpolation des Orientierungsvektors auf einer Kegelmantelfläche Orientierungsänderungen auf einer beliebig im Raum befindlichen Kegelmantelfläche durch Interpolation: - ORIPLANE in der Ebene (Großkreisinterpolation) - ORICONCW auf einer Kegelmantelfläche im Uhrzeigersinn - ORICONCCW auf einer Kegelmantelfläche gegen Uhrzeigersinn A6, B6, C6 Richtungsvektors (Drehachse des Kegels) - ORICONIO Interpolation auf einer Kegelmantelfläche mit: A7, B7, C7 Zwischenvektoren (Start- und Endorientierung) oder - ORICONTA auf einer Kegelmantelfläche tangentialer Übergang Orientierungsänderungen bezogen auf eine Bahn mit - ORICURVE Vorgabe der Bewegung zweier Kontaktpunkte über PO[XH]=(xe, x2, x3, x4, x5) Orientierungspolynome bis 5.Grades PO[YH]=(ye, y2, y3, y4, y5) Orientierungspolynome bis 5.Grades PO[ZH]=(ze, z2, z3, z4, z5) Orientierungspolynome bis 5.Grades - ORIPATHS Glättung des Orientierungsverlaufs mit A8, B8, C8 Umorientierungsphase des Werkzeugs entspricht: Richtung und Weglänge des Werkzeugs bei der Abhebebewegung</p>

Maschinentyp	Programmierung bei aktiver Transformation TRAORI
<p>Maschinentypen 1 und 3</p> <p>Weitere Maschinentypen mit zusätzlicher Drehung des Werkzeugs um sich selbst erfordern eine 3. Rundachse</p> <p>Orientierungstransformation, wie z.B. generische 6-Achs-Transformation. Drehungen des Orientierungsvektors.</p>	<p>Programmierung der Drehungen der Werkzeugorientierung mit LEAD Voreilwinkel Winkel relativ zum Flächennormalenvektor PO[PHI] Programmierung eines Polynoms bis 5.Grades TILT Seitwärtswinkel Drehung um Bahntangente (Z-Richtung) PO[PSI] Programmierung eines Polynoms bis 5.Grades THETA Drehwinkel (Drehung um die Werkzeugrichtung in Z) THETA= Wert der am Satzende erreicht wird THETA=AC(...) Satzweise auf Maßangabe absolut umschalten THETA=IC(...) Satzweise auf Kettenmaßangabe umschalten THETA=Θ₀ Programmierter Winkel G90/G91 interpolieren PO[THT]=(..) Programmierung eines Polynoms bis 5.Grades Programmierung des Drehvektors - ORIROTA Drehung absolut - ORIROTR relativer Drehvektor - ORIROTT tangentialer Drehvektor</p>
<p>Bahnrelative Orientierung für Orientierungsänderungen relativ zur Bahn oder Drehung des Drehvektors tangential zur Bahn</p>	<p>Orientierungsänderungen relativ zur Bahn mit - ORIPATH Werkzeugorientierung bezogen auf die Bahn - ORIPATHS zusätzlich bei einen Knick im Orientierungsverlauf Programmierung des Drehvektors - ORIROTC tangentialer Drehvektor, Drehung zur Bahntangente</p>

7.2 Drei-, Vier- und Fünf-Achs-Transformation (TRAORI)

7.2.1 Allgemeine Zusammenhänge Kardanischer Werkzeugkopf

Um optimale Schnittbedingungen beim Bearbeiten räumlich gekrümmter Flächen zu erzielen, muss der Anstellwinkel des Werkzeugs veränderbar sein.

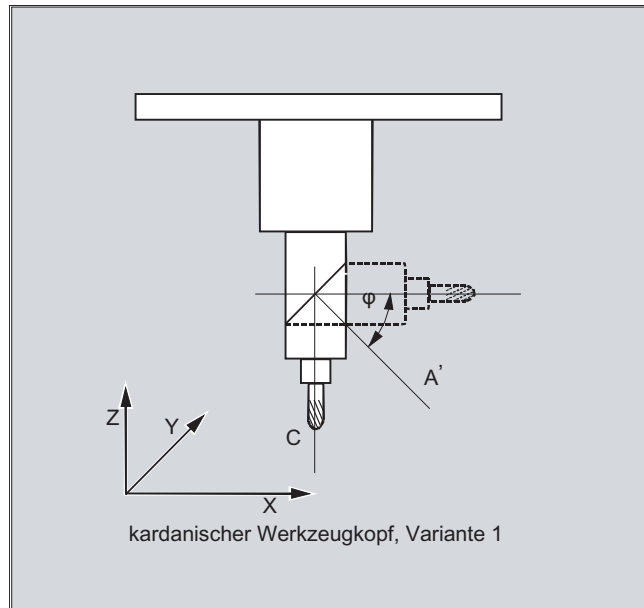


Mit welcher Maschinenkonstruktion dies erreicht wird, ist in den Achsdaten hinterlegt.

5-Achs-Transformation

Kardanischer Werkzeugkopf

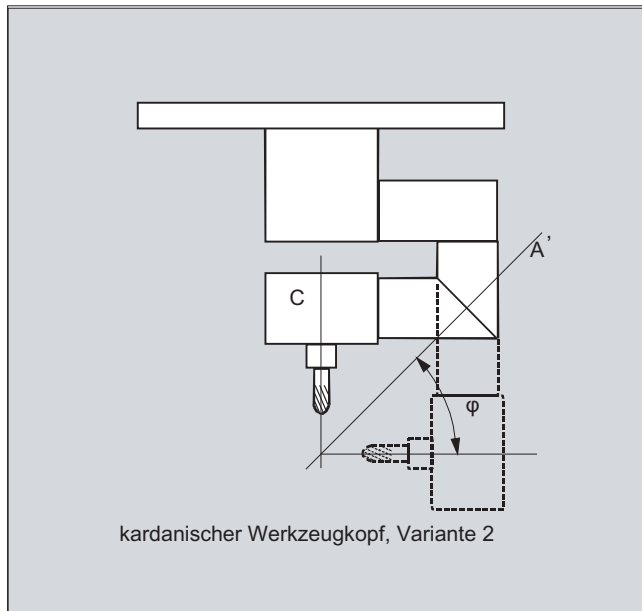
Hier legen drei Linearachsen (X, Y, Z) und zwei Orientierungsachsen (C, A) den Anstellwinkel und Arbeitspunkt des Werkzeugs fest. Eine der beiden Orientierungsachsen ist als Schrägachse angelegt, hier im Beispiel A' - in vielen Fällen als 45°-Anordnung.



In den hier gezeigten Beispielen sehen Sie die Anordnungen am Beispiel mit dem Kardanischen Werkzeugkopf der Maschinenkinematik CA!

Maschinenhersteller

Die Achsfolge der Orientierungsachsen und die Orientierungsrichtung des Werkzeugs kann abhängig von der Maschinenkinematik über Maschinendaten eingestellt werden.



In diesem Beispiel liegt A' unter dem Winkel φ zur X-Achse

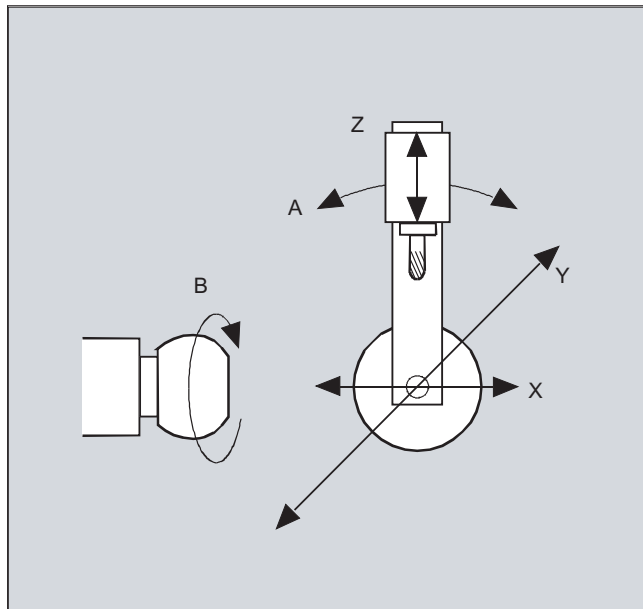
Allgemein gelten folgende mögliche Zusammenhänge:

A' liegt unter dem Winkel φ zur	X-Achse
B' liegt unter dem Winkel φ zur	Y-Achse
C' liegt unter dem Winkel φ zur	Z-Achse

Der Winkel φ kann im Bereich 0° bis $+89^\circ$ über Maschinendaten projiziert werden.

Mit schwenkbarer Linearachse

Hierbei handelt es sich um eine Anordnung mit bewegtem Werkstück und bewegtem Werkzeug. Die Kinematik setzt sich aus drei Linearachsen (X, Y, Z) und zwei rechtwinklig angeordneten Drehachsen zusammen. Die erste Rundachse wird z. B. über einen Kreuzschlitten von zwei Linearachsen bewegt, das Werkzeug steht parallel zur dritten Linearachse. Die zweite Drehachse dreht das Werkstück. Die dritte Linearachse (Schwenkachse) liegt in der Ebene des Kreuzschlittens.



Die Achsfolge der rotatorischen Achsen und die Orientierungsrichtung des Werkzeugs kann abhängig von der Maschinenkinematik über Maschinendaten eingestellt werden.

Es gelten folgende mögliche Zusammenhänge:

Achsen:	Achsfolgen:
1. Rundachse	A A B B C C
2. Rundachse	B C A C A B
Geschwenkte Linearachse	Z Y Z X Y X

Weitere Erläuterungen zu konfigurierbaren Achsfolgen für die Orientierungsrichtung des Werkzeugs siehe

Literatur: /FB3/ Funktionshandbuch Sonderfunktionen; 3- bis 5-Achs-Transformationen (F2), Kapitel Kardanischer Fräskopf, "Parametrierung".

7.2.2 Drei-, Vier-, und Fünf- Achs-Transformation (TRAORI)

Der Anwender kann zwei bzw. drei translatorische Achsen und eine rotatorische Achse projektieren. Die Transformationen gehen davon aus, dass die rotatorische Achse orthogonal auf der Orientierungsebene steht.

Die Orientierung des Werkzeugs ist nur in der Ebene möglich, die senkrecht zur rotatorischen Achse ist. Die Transformation unterstützt die Maschinentypen mit beweglichem Werkzeug und beweglichem Werkstück.

Die Projektierung und Programmierung der Drei- und Vier-Achs-Transformationen sind analog zu den Fünf-Achs-Transformationen.

Literatur:

Funktionshandbuch Sonderfunktionen; Mehrachstransformationen (F2)

Syntax

TRAORI (<n>)
 TRAORI (<n>, <X>, <Y>, <Z>, <A>,)
 TRAFOOF

Bedeutung

TRAORI:	Aktiviert die erste vereinbarte Orientierungstransformation	
TRAORI (<n>):	Aktiviert die mit n vereinbarte Orientierungstransformation	
<n>:	Nummer der Transformation	
	Wert:	1 oder 2
	Beispiel: TRAORI(1) aktiviert Orientierungstransformation 1	
<X>, <Y>, <Z>:	Komponente des Orientierungsvektors, in die das Werkzeug zeigt.	
<A>, :	Programmierbarer Offset für die Rundachsen	
TRAFOOF:	Transformation ausschalten	

Werkzeugorientierung

Abhängig von der gewählten Orientierungsrichtung des Werkzeugs muss im NC-Programm die aktive Arbeitsebene (G17, G18, G19) so eingestellt werden, dass die Werkzeuglängtenkorrektur in Richtung der Werkzeugorientierung wirkt.

Hinweis

Nach dem Einschalten der Transformation beziehen sich Positionsangaben (X, Y, Z) immer auf die Spitze des Werkzeugs. Änderung der Positionen der an der Transformation beteiligten Rundachsen führen zu Ausgleichsbewegungen der übrigen Maschinenachsen, wodurch die Position der Werkzeugspitze unverändert bleibt.

Die Orientierungstransformation ist immer von der Werkzeugspitze zur Werkzeugaufnahme gerichtet.

Offset für Orientierungsachsen

Bei Aktivierung der Orientierungstransformation kann ein zusätzlicher Offset für Orientierungsachsen direkt programmiert werden.

Es dürfen Parameter weggelassen werden, wenn bei der Programmierung die richtige Reihenfolge eingehalten wird.

Beispiel:

TRAORI (, , , , A, B) ; wenn nur ein einziger Offset eingegeben werden soll

Alternativ zur direkten Programmierung kann der zusätzliche Offset für Orientierungsachsen auch aus der momentan aktiven Nullpunktverschiebung automatisch übernommen werden. Die Übernahme wird über Maschinendaten projiziert.

Beispiele

TRAORI (1, 0, 0, 1)	; Die Grundorientierung des Werkzeugs zeigt in Z-Richtung
TRAORI (1, 0, 1, 0)	; Die Grundorientierung des Werkzeugs zeigt in Z-Richtung
TRAORI (1, 0, 1, 1)	; Die Grundorientierung des Werkzeugs zeigt in Y/Z-Richtung (entspricht Stellung -45°)

7.2.3 Varianten der Orientierungsprogrammierung und Grundstellung (ORIRESET)

Orientierungsprogrammierung der Werkzeugorientierung bei TRAORI

In Verbindung mit einer programmierbaren Orientierungstransformation TRAORI können zusätzlich zu den Linearachsen X, Y, Z auch über die Rundachsbezeichner A..., B..., C... Achspositionen oder virtuelle Achsen mit Winkeln oder Vektorkomponenten programmiert werden. Für Orientierungs- und Maschinenachsen sind verschiedene Interpolationsarten möglich. Unabhängig davon, welche Orientierungspolynome PO[Winkel] und Achspolynome PO[Achse] gerade aktiv sind, können mehrere unterschiedliche Polynomarten wie z.B. G1, G2, G3, CIP oder POLY programmiert sein.

Die Änderung der Orientierung des Werkzeuges kann auch über Orientierungsvektoren programmiert werden. Hierbei kann die Endorientierung jedes Satzes entweder durch direkte Programmierung des Vektors oder durch Programmierung der Rundachspositionen erfolgen.

Hinweis

Varianten der Orientierungsprogrammierung bei Drei- bis Fünf-Achs-Transformationen

Bei der Drei- bis Fünf-Achs-Transformation schließen sich die Varianten

1. A, B, C direkte Angabe der Maschinenachspositionen
2. A2, B2, C2 Winkelprogrammierung virtueller Achsen über Eulerwinkel oder RPY-Winkel
3. A3, B3, C3 Angabe der Vektorkomponenten
4. LEAD, TILT Angabe der Voreil- und Seitwärtswinkel bezogen auf die Bahn und Oberfläche
5. A4, B4, C4 und A5, B5, C5 Flächennormalenvektor am Satzanfang und am Satzende
6. A6, B6, C6 und A7, B7, C7 Interpolation des Orientierungsvektors auf einer Kegelmantelfläche
7. A8, B8, C8 Umorientierung des Werkzeuges, Richtung und Weglänge der Abhebebewegung gegenseitig aus.

Gemischt programmierte Werte werden durch Alarmmeldungen verhindert.

Grundstellung der Werkzeugorientierung ORIRESET

Durch Programmierung von ORIRESET(A, B, C) werden Orientierungsachsen linear und synchron von ihrer momentanen Position zu der angegebenen Grundstellungsposition gefahren.

Wird für eine Achse keine Grundstellungsposition programmiert, dann wird definierte Position aus dem dazugehörigen Maschinendatum \$MC_TRAFO5_ROT_AX_OFFSET_1/2 verwendet. Eventuell aktive Frames der Rundachsen werden dabei nicht berücksichtigt.

Hinweis

Nur wenn eine Orientierungstransformation mit TRAORI(...) aktiv ist, kann eine Grundstellung der Werkzeugorientierung kinematikunabhängig mit ORIRESET(...) ohne Alarm 14101 programmiert werden.

Beispiele

```
1. Beispiel für Maschinenkinematik CA (Kanalachsnamen C, A)
ORIRESET(90, 45)      ; C auf 90 Grad, A auf 45 Grad
ORIRESET(, 30)       ; C auf $MC_TRAFO5_ROT_AX_OFFSET_1/2[0], A auf 30 Grad
ORIRESET( )          ; C auf $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                    ; A auf $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

2. Beispiel für Maschinenkinematik CAC (Kanalachsnamen C, A, B)
ORIRESET(90, 45, 90) ; C auf 90 Grad, A auf 45 Grad, B auf 90 Grad
ORIRESET( )          ; C auf $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                    ; A auf $MC_TRAFO5_ROT_AX_OFFSET_1/2[1],
                    ; B auf $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]
```

Programmierung der Drehungen LEAD, TILT und THETA

Die Drehungen der Werkzeugorientierung werden bei der Drei- bis Fünf-Achs-Transformation mit den Voreilwinkel LEAD und den Seitwärtswinkel TILT programmiert.

Bei einer Transformation mit dritter Rundachse sind sowohl für die Orientierung mit Vektorkomponenten als auch mit Angabe der Winkel LEAD, TILT zusätzliche Programmierungen von C2 (Verdrehungen des Orientierungsvektors) erlaubt.

Mit einer zusätzlichen dritten Rundachse kann die Drehung des Werkzeugs um sich selbst mit dem Drehwinkel THETA programmiert werden.

7.2.4 Programmierung der Werkzeugorientierung (A..., B..., C..., LEAD, TILT)

Für die Programmierung der Orientierung des Werkzeugs gibt es folgende Möglichkeiten:

1. Direkte Programmierung der Bewegung der Rundachsen. Die Orientierungsänderung erfolgt immer im Basis- bzw. Maschinen-Koordinatensystem. Die Orientierungsachsen werden als Synchronachsen verfahren.
2. Programmierung in Euler- oder RPY-Winkeln gemäß Winkeldefinition über A2, B2, C2.
3. Programmierung des Richtungsvektors über A3, B3, C3. Der Richtungsvektor zeigt von der Werkzeugspitze in Richtung Werkzeugaufnahme.
4. Programmierung des Flächennormalenvektors am Satzanfang mit A4, B4, C4 und am Stazende mit A5, B5, C5 (Stirnfräsen).

5. Programmierung über Voreilwinkel LEAD und Seitwärtswinkel TILT
6. Programmierung der Drehachse des Kegels als normierter Vektor über A6, B6, C6 oder der Zwischenorientierung auf der Kegelmantelfläche über A7, B7, C7, siehe Kapitel "Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONxx)".
7. Programmierung der Umoorientierung, Richtung und Weglänge des Werkzeugs während der Abhebebewegung über A8, B8, C8, siehe Kapitel "Glättung des Orientierungsverlaufs (ORIPATHS A8=, B8=, C8=)"

Hinweis

In allen Fällen ist die Orientierungsprogrammierung nur zulässig, wenn eine Orientierungstransformation eingeschaltet ist.

Vorteil: Diese Programme sind auf jede Maschinenkinematik portierbar.

Definition der Werkzeugorientierung über G-Code

Hinweis

Maschinenhersteller

Über Maschinendatum kann zwischen Euler- oder RPY-Winkeln umgeschaltet werden. Bei entsprechenden Maschinendaten Einstellungen ist eine Umschaltung sowohl abhängig als auch unabhängig vom aktiven G-Code der Gruppe 50 möglich. Folgende Einstellmöglichkeiten stehen zur Auswahl:

1. Wenn beide Maschinendaten für die Definition der Orientierungsachsen und Definition der Orientierungswinkel über G-Code auf Null gesetzt sind:
Die mit A2, B2, C2 programmierten Winkel werden **abhängig vom Maschinendatum** Winkeldefinition der Orientierungsprogrammierung entweder als Euler- oder RPY-Winkeln interpretiert.
 2. Wenn das Maschinendatum für die Definition der Orientierungsachsen über G-Code auf Eins gesetzt ist, erfolgt Umschaltung **abhängig** vom aktiven G-Code der Gruppe 50:
Die mit A2, B2, C2 programmierten Winkel werden gemäß eines der aktiven G-Codes ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS und ORIPY2 interpretiert. Die mit den Orientierungsachsen programmierten Werte werden entsprechend dem aktiven G-Code der Gruppe 50 auch als Orientierungswinkel interpretiert.
 3. Wenn das Maschinendatum für die Definition der Orientierungswinkel über G-Code auf Eins und das Maschinendatum für die Definition der Orientierungsachsen über G-Code auf Null gesetzt ist erfolgt Umschaltung **unabhängig** vom aktiven G-Code der Gruppe 50:
Die mit A2, B2, C2 programmierten Winkel werden gemäß eines der aktiven G-Codes ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS und ORIPY2 interpretiert. Die mit den Orientierungsachsen programmierten Werte werden unabhängig vom aktiven G-Code der Gruppe 50 immer als Rundachspositionen interpretiert.
-

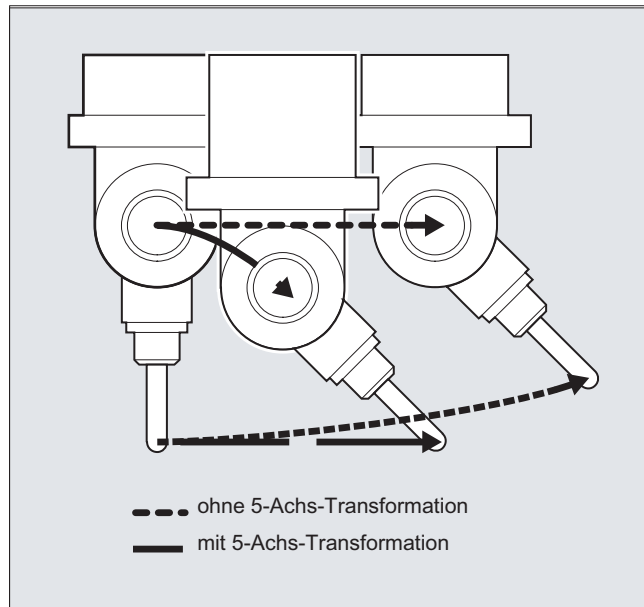
Programmierung

G1 X... Y... Z... A... B... C...	Programmierung der Bewegung der Rundachsen
G1 X... Y... Z... A2= B2= C2=	Programmierung in Eulerwinkeln
G1 X... Y... Z... A3== B3== C3==	Programmierung des Richtungsvektors
G1 X... Y... Z... A4== B4== C4==	Programmierung des Flächennormalenvektors am Satzanfang
G1 X... Y... Z... A5== B5== C5==	Programmierung des Flächennormalenvektors am Satzende
LEAD=	Voreilwinkel für die Programmierung der Werkzeugorientierung
TILT=	Seitwärtswinkel für die Programmierung der Werkzeugorientierung

Parameter

G...:	Angabe der Bewegungsart der Rundachsen
X... Y... Z...:	Angabe der Linearachsen
A... B... C...:	Angabe der Maschinenachspositionen der Rundachsen
A2= B2= C2= :	Winkelprogrammierung (Euler- oder RPY-Winkel) virtueller Achsen bzw. Orientierungsachsen
A3== B3== C3== :	Angabe der Vektorkomponenten Richtungsvektors
A4== B4== C4== :	Angabe z. B. beim Stirnfräsen die Komponente des Flächennormalenvektors am Satzanfang
A5== B5== C5== :	Angabe z. B. beim Stirnfräsen die Komponente des Flächennormalenvektors am Satzende
LEAD= :	Winkel relativ zum Flächennormalenvektor, in der von Bahntangente und Flächennormalenvektor aufgespannten Ebene
TILT= :	Winkel in der Ebene, senkrecht zur Bahntangente relativ zum Flächennormalenvektor

Beispiel: Gegenüberstellung ohne und mit 5-Achs-Transformation



Weitere Informationen

In der Regel werden 5-Achs-Programme von CAD/CAM-Systemen erzeugt und nicht an der Steuerung eingegeben. Deshalb wenden sich die folgenden Erklärungen hauptsächlich an Programmierer von Postprozessoren.

Die Art der Orientierungsprogrammierung wird in der G-Code Gruppe 50 festgelegt:

G-Funktion	Orientierungsprogrammierung
ORIEULER:	über Euler-Winkel
ORIRPY:	über RPY-Winkel (Drehreihenfolge ZYX)
ORIVIRT1:	über virtuelle Orientierungsachsen (Definition 1)
ORIVIRT2:	über virtuelle Orientierungsachsen (Definition 2)
ORIAPOS:	über virtuelle Orientierungsachsen mit Rundachspositionen
ORIPY2:	über RPY-Winkel (Drehreihenfolge XYZ)

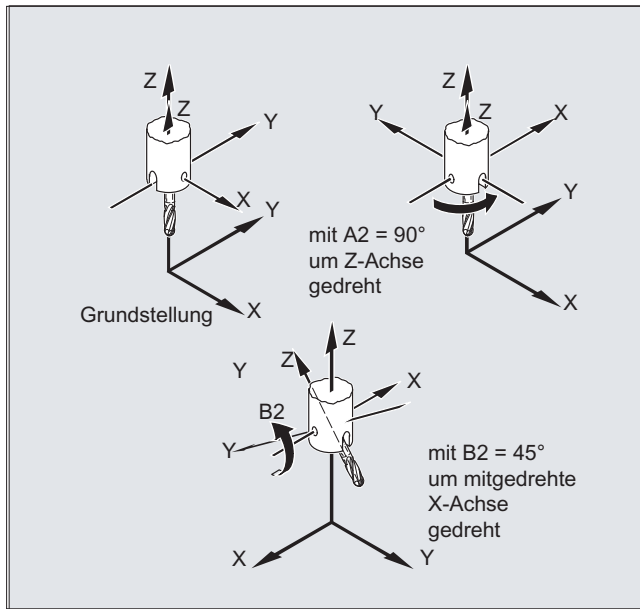
Hinweis

Über Maschinendaten können vom Maschinenhersteller verschiedene Varianten definiert werden. Bitte beachten Sie die Angaben des Maschinenherstellers.

Programmierung in Eulerwinkeln ORIEULER

Die bei der Orientierungsprogrammierung mit A_2 , B_2 , C_2 programmierten Werte werden als Eulerwinkel (in Grad) interpretiert.

Der Orientierungsvektor ergibt sich, indem ein Vektor in Z-Richtung zunächst mit A_2 um die Z-Achse, dann mit B_2 um die neue X-Achse und zuletzt mit C_2 um die neue Z-Achse gedreht wird.



In diesem Fall ist der Wert von $C2$ (Drehung um neue Z-Achse) bedeutungslos und muss nicht programmiert werden.

Programmierung in RPY-Winkeln ORIRPY

Die bei der Orientierungsprogrammierung mit $A2$, $B2$, $C2$ programmierten Werte werden als RPY-Winkel (in Grad) interpretiert.

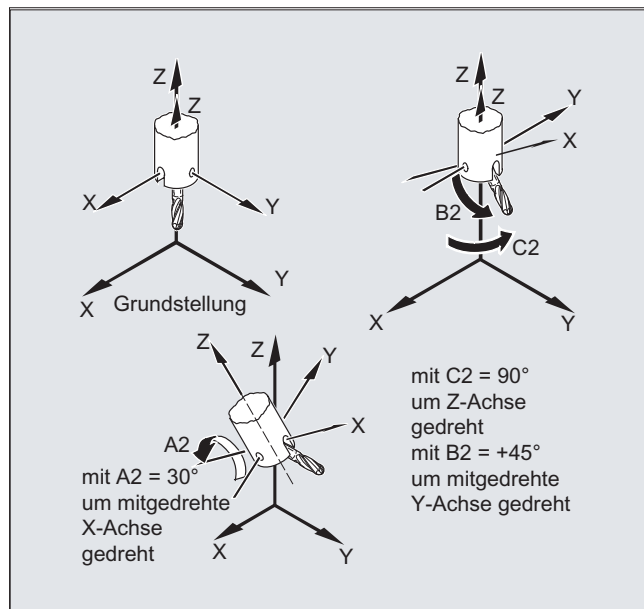
Hinweis

Im Gegensatz zur Eulerwinkel-Programmierung haben hier alle drei Werte Einfluss auf den Orientierungsvektor.

Bei Winkeldefinition mit Orientierungswinkel über RPY-Winkel gilt für die Orientierungsachsen:

`$MC_ORI_DEF_WITH_G_CODE = 0`

Der Orientierungsvektor ergibt sich, indem ein Vektor in Z-Richtung zunächst mit $C2$ um die Z-Achse, dann mit $B2$ um die neue Y-Achse und zuletzt mit $A2$ um die neue X-Achse gedreht wird.



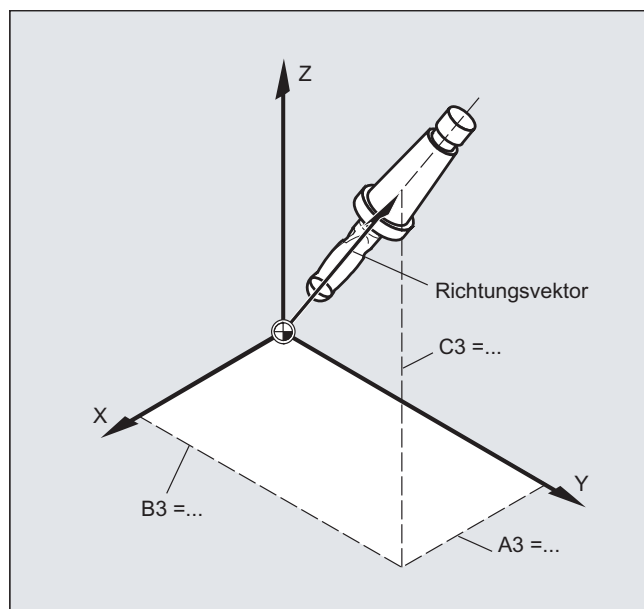
Ist das Maschinendatum zur über Definition der Orientierungsachsen über G-Code $\$MC_ORI_DEF_WITH_G_CODE = 1$, dann gilt:

Der Orientierungsvektor ergibt sich, indem ein Vektor in Z-Richtung zunächst mit $A2$ um die Z-Achse, dann mit $B2$ um die neue Y-Achse und zuletzt mit $C2$ um die neue X-Achse gedreht wird.

Programmierung des Richtungsvektors

Die Komponenten des Richtungsvektors werden mit $A3$, $B3$, $C3$ programmiert. Der Vektor zeigt in Richtung Werkzeugaufnahme; die Länge des Vektors ist dabei ohne Bedeutung.

Nicht programmierte Vektorkomponenten werden gleich Null gesetzt.



Programmierung der Werkzeugorientierung mit LEAD= und TILT=

Die resultierende Werkzeugorientierung wird ermittelt aus:

- Bahntangente
- Flächennormalenvektor
am Satzanfang A_4, B_4, C_4 und am Satzende A_5, B_5, C_5
- Voreilwinkel $LEAD$
in der von Bahntangente und Flächennormalenvektor aufgespannten Ebene
- Seitwärtswinkel $TILT$ am Satzende
senkrecht zur Bahntangente und relativ zum Flächennormalenvektor

Verhalten bei Innenecken (bei 3D-WZK):

Wenn der Satz an einer Innenecke verkürzt wird, wird die resultierende Werkzeugorientierung ebenso am Satzende erreicht.

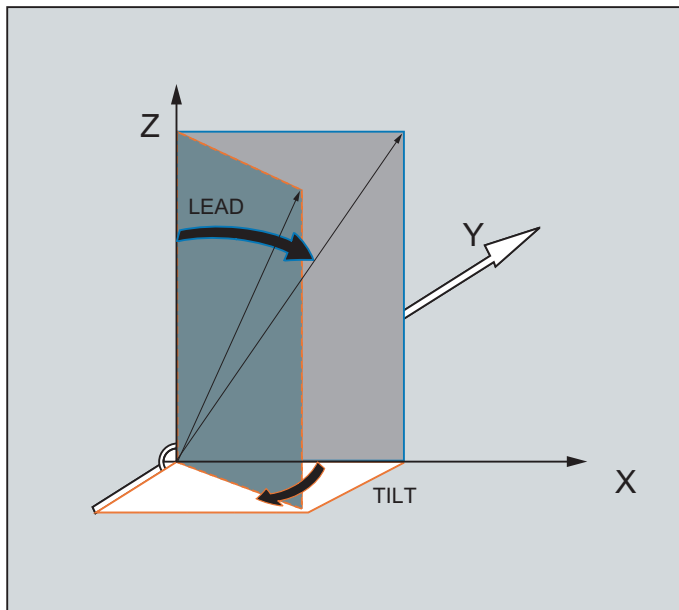
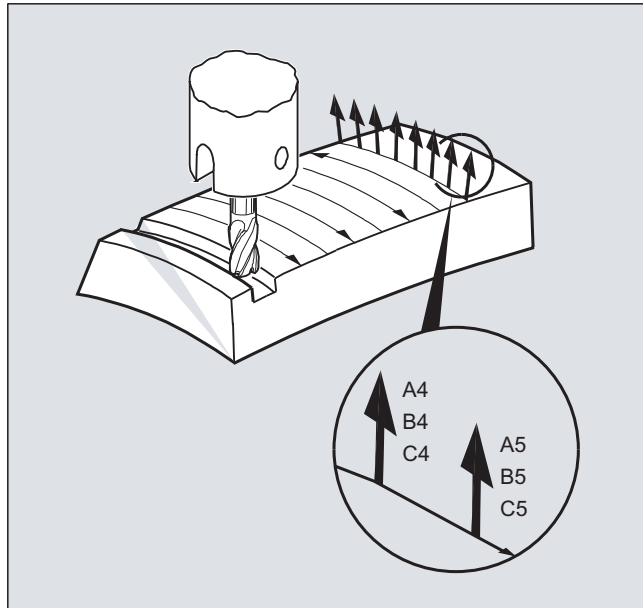


Bild 7-1 Definition der Werkzeugorientierung mit LEAD= und TILT=

7.2.5 Stirnfräsen (A4, B4, C4, A5, B5, C5)

Stirnfräsen dient zur Bearbeitung beliebig gekrümmter Oberflächen.



Für diese Art des 3D-Fräsen benötigen Sie die zeilenweise Beschreibung der 3D-Bahnen auf der Werkstückoberfläche.

Die Berechnungen werden unter Berücksichtigung der Werkzeugform und Werkzeugabmessungen üblicherweise im CAM durchgeführt. Die fertig berechneten NC-Sätze werden dann über Postprozessoren in die Steuerung eingelesen.

Programmierung der Bahnkrümmung

Beschreibung der Flächen

Die Beschreibung der Bahnkrümmung erfolgt über Flächennormalenvektoren mit folgenden Komponenten:

A4, B4, C4 Startvektor am Satzanfang

A5, B5, C5 Endvektor am Satzende

Steht in einem Satz nur der Startvektor, bleibt der Flächennormalenvektor über den ganzen Satz konstant. Steht in einem Satz nur der Endvektor, so wird vom Endwert des vorherigen Satzes über Großkreisinterpolation zum programmierten Endwert interpoliert.

Sind Start- und Endvektor programmiert, so wird zwischen beiden Richtungen ebenfalls über Großkreisinterpolation interpoliert. Hierdurch lassen sich kontinuierlich glatte Bahnwege erzeugen.

In der Grundstellung zeigen Flächennormalenvektoren unabhängig von der aktiven Ebene G17 bis G19 in Z-Richtung.

Die Länge eines Vektors ist ohne Bedeutung.

Nicht programmierte Vektorkomponenten werden zu Null gesetzt.

Bei aktivem ORIWKS (siehe "Bezug der Orientierungsachsen (ORIWKS, ORIMKS) (Seite 322)") beziehen sich die Flächennormalenvektoren auf den aktiven Frame und werden bei Frame-Drehung mitgedreht.

Maschinenhersteller

Der Flächennormalenvektor muss innerhalb eines über Maschinendatum einstellbaren Grenzwertes senkrecht zur Bahntangente stehen, ansonsten wird Alarm ausgegeben.

7.2.6 Bezug der Orientierungsachsen (ORIWKS, ORIMKS)

Bei Orientierungsprogrammierung im Werkstückkoordinatensystem über

- Euler- bzw. RPY-Winkel oder
- Orientierungsvektor

kann der Verlauf der Drehbewegung über ORIMKS/ORIWKS eingestellt werden.

Hinweis

Maschinenhersteller

Die Interpolationsart für die Orientierung wird festgelegt mit dem Maschinendatum:

MD21104 \$MC_ORI_IPO_WITH_G_CODE

= FALSE: Bezug sind die G-Funktionen ORIWKS und ORIMKS

= TRUE: Bezug sind die G- Funktionen der 51. Gruppe (ORIXES, ORIVECT, ORIPLANE, ...)

Syntax

ORIMKS= . . .

ORIWKS= . . .

Bedeutung

ORIMKS:	Drehung im Maschinenkoordinatensystem
ORIWKS:	Drehung im Werkstückkoordinatensystem

Hinweis

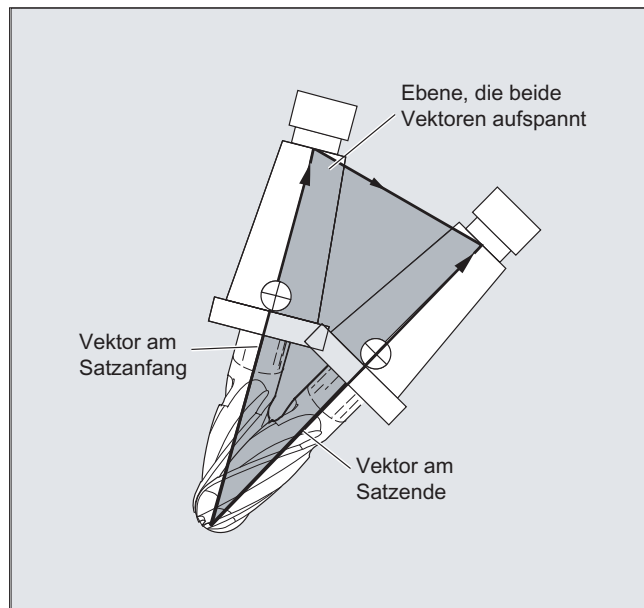
ORIWKS ist Grundeinstellung. Ist bei einem Fünf-Achs-Programm nicht von vornherein klar, auf welcher Maschine es ablaufen soll, so ist grundsätzlich ORIWKS zu wählen. Welche Bewegungen die Maschine tatsächlich ausführt, hängt von der Maschinenkinematik ab.

Mit ORIMKS können tatsächliche Maschinenbewegungen programmiert werden, z. B. um Kollisionen mit Vorrichtungen o. ä. zu vermeiden.

Weitere Informationen

Bei **ORIMKS** ist die ausgeführte Werkzeugbewegung von der Maschinenkinematik **abhängig**. Bei Orientierungsänderung mit raumfester Werkzeugspitze wird zwischen den Rundachspositionen linear interpoliert.

Bei **ORIWKS** ist die Werkzeugbewegung von der Maschinenkinematik **unabhängig**. Bei Orientierungsänderung mit raumfester Werkzeugspitze bewegt sich das Werkzeug in der vom Anfangs- und Endvektor aufgespannten Ebene.



Singuläre Stellungen

Hinweis

ORIWKS

Orientierungsbewegungen im Bereich der singulären Stellung der Fünf-Achs-Maschine erfordern große Bewegungen der Maschinenachsen. (Beispielsweise sind bei einem Drehschwenkkopf mit C als Drehachse und A als Schwenkachse alle Stellungen mit $A=0$ singulär.)

Maschinenhersteller

Um die Maschinenachsen nicht zu überlasten, senkt die Geschwindigkeitsführung die Bahngeschwindigkeit in der Nähe der singulären Stellen stark ab.

Mit den Maschinendaten

```
$MC_TRAFO5_NON_POLE_LIMIT
```

```
$MC_TRAFO5_POLE_LIMIT
```

kann die Transformation so parametrisiert werden, dass Orientierungsbewegungen in der Nähe des Pols durch den Pol gelegt werden und eine zügige Bearbeitung möglich ist.

Singuläre Stellen werden nur mit dem MD \$MC_TRAFO5_POLE_LIMIT behandelt.

Literatur:

/FB3/ Funktionshandbuch Sonderfunktionen; 3- bis 5-Achs-Transformation (F2), Kapitel "Singuläre Stellen und ihre Behandlung".

7.2.7 Programmierung der Orientierungsachsen (ORIXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Die Funktion "Orientierungsachsen" beschreibt die Orientierung des Werkzeugs im Raum und wird durch Programmierung der Offsets für die Rundachsen erreicht. Ein weiterer dritter Freiheitsgrad kann durch die zusätzliche Drehung des Werkzeugs um sich selbst erzielt werden. Diese Werkzeugorientierung erfolgt beliebig im Raum über eine dritte Rundachse und erfordert die Sechs-Achs-Transformation. Die Eigendrehung des Werkzeugs um sich selbst wird abhängig von der Interpolationsart der Drehvektoren mit dem Drehwinkel THETA festgelegt (siehe "Drehungen der Werkzeugorientierung (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Seite 333)".

Orientierungsachsen werden über die Achsbezeichner A2, B2, C2 programmiert.

Syntax

```
N... ORIXES/ORIVECT           ; Lineare oder Großkreisinterpolation
N... G1 X Y Z A B C

N... ORIPLANE                 ; Orientierungsinterpolation der Ebene

N... ORIEULER/ORIRPY/ORIRPY2 ; Orientierungswinkel Euler-/RPY-Winkel
N... G1 X Y Z A2= B2= C2=    ; Winkelprogrammierung virtueller Achsen

N... ORIVIRT1/ORIVIRT2       ; virtuelle Orientierungsachsen Def. 1/2
N... G1 X Y Z A3= B3= C3=    ; Richtungsvektorprogrammierung
```

Hinweis

Für Orientierungsänderungen entlang einer im Raum befindlichen Kegelmantelfläche können weitere Rundachsoffsets der Orientierungsachsen programmiert werden (siehe "Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Seite 326)".

Bedeutung

ORIXES:	Lineare Interpolation der Maschinen- oder Orientierungsachsen
ORIVECT:	Großkreisinterpolation (identisch mit ORIPLANE)

ORIMKS: ORIWKS:	Drehung im Maschinenkoordinatensystem Drehung im Werkstückkoordinatensystem Beschreibung siehe "Bezug der Orientierungsachsen (ORIWKS, ORIMKS) (Seite 322)".
A= B= C=:	Programmierung der Maschinenachspannung
ORIEULER:	Orientierungsprogrammierung über Euler-Winkel
ORIRPY:	Orientierungsprogrammierung über RPY-Winkel Die Drehreihenfolge ist XYZ, wobei gilt: <ul style="list-style-type: none"> • A2 ist der Drehwinkel um X • B2 ist der Drehwinkel um Y • C2 ist der Drehwinkel um Z
ORIRPY2:	Orientierungsprogrammierung über RPY-Winkel Die Drehreihenfolge ist ZYX, wobei gilt: <ul style="list-style-type: none"> • A2 ist der Drehwinkel um Z • B2 ist der Drehwinkel um Y • C2 ist der Drehwinkel um X
A2= B2= C2=:	Winkelprogrammierung virtueller Achsen
ORIVIRT1/ORIVIRT2:	Orientierungsprogrammierung über virtuelle Orientierungsachsen Definition 1: Festlegung nach MD21120 \$MC_ORIAX_TURN_TAB_1 Definition 2: Festlegung nach MD21130 \$MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Richtungsvektorprogrammierung der Richtungsachse

Weitere Informationen

Maschinenhersteller

Mit MD21102 \$MC_ORI_DEF_WITH_G_CODE wird festgelegt, wie die programmierten Winkel A2, B2, C2 definiert werden:

Definition erfolgt nach MD21100 \$MC_ORIENTATION_IS_EULER (Standard) oder Definition erfolgt nach G-Gruppe 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Mit MD21104 \$MC_ORI_IPO_WITH_G_CODE wird festgelegt, welche Interpolationsart wirksam ist: ORIWKS/ORIMKS oder ORIAXES/ORIVECT.

Betriebsart JOG

Die Orientierungswinkel werden in dieser Betriebsart immer linear interpoliert. Beim kontinuierlichen und inkrementellen Verfahren über Verfahrstasten kann nur eine Orientierungsachse verfahren werden. Über die Handräder können die Orientierungsachsen gleichzeitig verfahren werden.

Für das Handverfahren von Orientierungsachsen wirkt der kanalspezifische Vorschub-Korrekturschalter bzw. der Eilgang-Korrekturschalter bei Eilgangüberlagerung.

Mit folgenden Maschinendaten ist eine separate Geschwindigkeitsvorgabe möglich:

MD21160 \$MC_JOG_VELO_RAPID_GEO

MD21165 \$MC_JOG_VELO_GEO

MD21150 \$MC_JOG_VELO_RAPID_ORI

MD21155 \$MC_JOG_VELO_ORI

Hinweis

SINUMERIK 840D sl mit "Transformationspaket Handling"

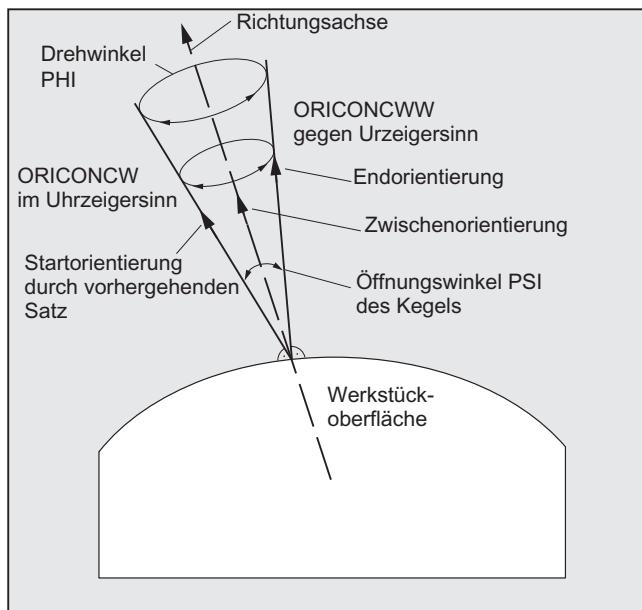
Mit der Funktion "Kartesisches Handverfahren" kann im JOG-Betrieb die Translation von Geometrieachsen in den Bezugssystemen MKS, WKS und TKS getrennt voneinander eingestellt werden.

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformation (M1)

7.2.8 Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

Mit der erweiterten Orientierung ist es möglich, Orientierungsänderungen entlang sich einer im Raum befindlichen Kegelmantelfläche auszuführen. Die Interpolation des Orientierungsvektors auf einer Kegelmantelfläche erfolgt mit den modalen Befehlen ORICONxx. Für die Interpolation in einer Ebene kann die Endorientierung mit ORIPLANE programmiert werden. Generell wird die Startorientierung durch die vorhergehenden Sätze festgelegt.



Programmierung

Die Endorientierung wird entweder durch Angabe der Winkelprogrammierung in Euler- oder RPY-Winkel mit A2, B2, C2 oder durch Programmierung der Rundachspositionen mit A, B, C festgelegt. Für die Orientierungsachsen entlang der Kegelmantelfläche sind weitere Programmierangaben erforderlich:

- Drehachse des Kegels als Vektor mit A6, B6, C6
- Öffnungswinkel PSI mit den Bezeichner NUT
- Zwischenorientierung im Kegelmantel mit A7, B7, C7

Hinweis

Programmierung des Richtungsvektor A6, B6, C6 für die Drehachse des Kegels

Die Programmierung einer Endorientierung ist nicht unbedingt erforderlich. Ist keine Endorientierung angegeben, dann wird ein voller Kegelmantel mit 360 Grad interpoliert.

Programmierung des Öffnungswinkel des Kegels mit NUT=winkel

Die Angabe einer Endorientierung ist zwingend erforderlich.

Ein vollständiger Kegelmantel mit 360 Grad kann auf diese Weise nicht interpoliert werden.

Programmierung der Zwischenorientierung A7, B7, C7 im Kegelmantel

Die Angabe einer Endorientierung ist zwingend erforderlich. Die Orientierungsänderung und Drehrichtung wird eindeutig durch die drei Vektoren Start-, End- und Zwischenorientierung festgelegt. Alle drei Vektoren müssen hierbei voneinander unterschiedlich sein. Ist die programmierte Zwischenorientierung parallel zur Start- oder Endorientierung, dann wird eine lineare Großkreisinterpolation der Orientierung in der Ebene, die von Start- und Endvektor aufgespannt wird, durchgeführt.

Erweiterte Orientierungsinterpolation auf einer Kegelmantelfläche

N... ORICONCW oder ORICONCCW

N... A6= B6= C6= A3= B3= C3=
oder

N... ORICONTO

N... G1 X Y Z A6= B6= C6=
oder

N... ORICONIO

N... G1 X Y Z A7= B7= C7=

N... PO[PHI]=(a2, a3, a4, a5)

N... PO[PSI]=(b2, b3, b4, b5)

Interpolation auf einen Kegelmantel mit Richtungsvektor im/gegen Uhrzeigersinn des Kegels und Endorientierung oder tangentialem Übergang und Angabe der Endorientierung oder Angabe der Endorientierung und einer Zwischenorientierung im Kegelmantel mit Polynome für Drehwinkel und Polynome für Öffnungswinkel

Parameter

ORIPLANE:	Interpolation in der Ebene (Großkreisinterpolation)
ORICONCW:	Interpolation auf einer Kegelmantelfläche im Uhrzeigersinn
ORICONCCW:	Interpolation auf einer Kegelmantelfläche gegen Uhrzeigersinn
ORICONTO:	Interpolation auf einer Kegelmantelfläche tangentialer Übergang

7.2 Drei-, Vier- und Fünf-Achs-Transformation (TRAORI)

A6= B6= C6=:	Programmierung der Drehachse des Kegels (normierter Vektor)
NUT=winkel:	Öffnungswinkel des Kegels in Grad
NUT=+179:	Verfahrwinkel kleiner oder gleich 180 Grad
NUT=-181:	Verfahrwinkel größer oder gleich 180 Grad
ORICONIO:	Interpolation auf einer Kegelmantelfläche
A7= B7= C7=:	Zwischenorientierung (Programmierung als normierter Vektor)
PHI:	Drehwinkel der Orientierung um die Richtungsachse des Kegels
PSI:	Öffnungswinkel des Kegels
mögliche Polynome PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Außer den jeweiligen Winkeln sind auch Polynome maximal 5. Grades programmierbar

Beispiel: Unterschiedliche Orientierungsänderungen

Programmcode	Kommentar
...	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; Orientierungstransformation ein.
N30 ORIVECT	; Werkzeug-Orientierung als Vektor interpolieren.
...	; Werkzeugorientierung in der Ebene.
N40 ORIPLANE	; Großkreisinterpolation auswählen.
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Orientierung in der Y/Z-Ebene um 45 Grad gedreht, am Satzende wird die Orientierung $(0, 1/\sqrt{2}, 1/\sqrt{2})$ erreicht.
...	
N70 ORICONCW	; Orientierungsprogrammierung auf Kegelmantel:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	; Der Orientierungsvektor wird auf einem Kegelmantel mit der Richtung $(0, 0, 1)$ bis zur Orientierung $(1/\sqrt{2}, 0, 1/\sqrt{2})$ im Uhrzeigersinn interpoliert, der Drehwinkel beträgt hierbei 270 Grad.
N90 A6=0 B6=0 C6=1	; Die Werkzeugorientierung durchläuft eine volle Umdrehung auf demselben Kegelmantel.

Weitere Informationen

Sollen Orientierungsänderungen auf einer beliebig im Raum liegenden Kegelmantelfläche beschrieben werden, dann muss der Vektor um den die Werkzeugorientierung gedreht werden soll, bekannt sein. Außerdem müssen die Start- und Endorientierung vorgegeben werden. Die Startorientierung ergibt sich aus den vorhergehenden Satz und die Endorientierung muss entweder programmiert oder durch andere Bedingungen festgelegt werden.

Programmierung in der Ebene ORIPLANE entspricht ORIVECT

Die Programmierung der Großkreisinterpolation zusammen mit Winkelpolynomen entspricht der Linear- und Polynominterpolation von Konturen. Die Werkzeugorientierung wird in einer Ebene interpoliert, die von der Start- und Endorientierung aufgespannt wird. Werden zusätzlich Polynome programmiert, dann kann der Orientierungsvektor auch aus der Ebene gekippt werden.

Programmierung von Kreisen in einer Ebene G2/G3, CIP und CT

Die erweiterte Orientierung entspricht der Interpolation von Kreisen in einer Ebene. Zu den entsprechenden Programmiermöglichkeiten von Kreisen mit Mittelpunktangabe oder Radiusangabe wie G2/G3, Kreis über Zwischenpunkt CIP und Tangentialkreise CT siehe

Literatur: Programmierhandbuch Grundlagen, "Wegbefehle programmieren".

Orientierungsprogrammierung**Interpolation des Orientierungsvektors auf einer Kegelmantelfläche ORICONxx**

Für die Interpolation von Orientierungen auf einer Kegelmantelfläche können vier verschiedene Interpolationsarten aus der G-Code Gruppe 51 ausgewählt werden:

1. Interpolation auf einen Kegelmantel im Uhrzeigersinn `ORICONCW` mit Angabe der Endorientierung und der Kegelrichtung oder des Öffnungswinkels. Der Richtungsvektor wird mit den Bezeichnern `A6`, `B6`, `C6` und der Öffnungswinkel des Kegels wird mit dem Bezeichner `NUT=` Wertebereich im Intervall 0 bis 180 Grad programmiert.
2. Interpolation auf einen Kegelmantel gegen Uhrzeigersinn `ORICONCWW` mit Angabe der Endorientierung und der Kegelrichtung oder des Öffnungswinkels. Der Richtungsvektor wird mit den Bezeichnern `A6`, `B6`, `C6` und der Öffnungswinkel des Kegels wird mit dem Bezeichner `NUT=` Wertebereich im Intervall 0 bis 180 Grad programmiert.
3. Interpolation auf einen Kegelmantel `ORICONIO` mit Angabe der Endorientierung und einer Zwischenorientierung, die mit den Bezeichnern `A7`, `B7`, `C7` programmiert wird.
4. Interpolation auf einen Kegelmantel `ORICONTO` mit tangentialem Übergang und Angabe der Endorientierung. Der Richtungsvektor wird mit den Bezeichnern `A6`, `B6`, `C6` programmiert.

7.2.9 Orientierungsvorgabe zweier Kontaktpunkte (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Programmierung der Orientierungsänderung durch die zweite Raumkurve ORICURVE

Eine weitere Möglichkeit der Programmierung von Orientierungsänderungen besteht darin, außer der Werkzeugspitze entlang einer Raumkurve auch die Bewegung eines zweiten Kontaktpunktes des Werkzeugs mit `ORICURVE` zu programmieren. Damit können Orientierungsänderungen des Werkzeugs, wie bei der Programmierung des Werkzeugvektors selber, eindeutig festgelegt werden.

Maschinenhersteller

Beachten Sie bitte die Hinweise des Maschinenherstellers zu über Maschinendatum einstellbare Achsbezeichner für die Programmierung der 2. Orientierungsbahn des Werkzeugs.

Programmierung

Bei dieser Interpolationsart können für die beiden Raumkurven Punkte mit G1 bzw. Polynome mit POLY programmiert werden. Kreise und Evolventen sind nicht zulässig. Zusätzlich kann eine Spline-Interpolation mit BSPLINE und die Funktion "Zusammenfassung kurzer Spline-Sätze" aktiviert werden.

Literatur:

Funktionshandbuch Grundfunktionen; Bahnsteuerbetrieb, Genauhalt, Look Ahead (B1), Kapitel: Zusammenfassung kurzer Spline-Sätze

Die anderen Splinearten ASPLINE und CSPLINE sowie die Aktivierung eines Kompressors mit COMPON, COMPCURV oder COMPCAD sind nicht zulässig.

Die Bewegung der zwei Kontaktpunkte des Werkzeugs kann bei der Programmierung der Orientierungspolynome für Koordinaten bis maximal 5.Grades vorgegeben werden.

Erweiterte Orientierungsinterpolation mit zusätzlicher Raumkurve und Polynome für Koordinaten

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)

Angabe der Bewegung des zweiten Kontaktpunktes des Werkzeugs und zusätzliche Polynome der jeweiligen Koordinaten

Parameter

ORICURVE	Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeuges.
XH YH ZH	Bezeichner der Koordinaten des zweiten Kontaktpunktes des Werkzeuges der zusätzlichen Kontur als Raumkurve
mögliche Polynome PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)	Außer den jeweiligen Endpunkten sind die Raumkurven zusätzlich mit Polynomen programmierbar.
xe, ye, ze	Endpunkte der Raumkurve
xi, yi, zi	Koeffizienten der Polynome maximal 5. Grades

Hinweis**Bezeichner XH YH ZH für die Programmierung einer 2. Orientierungsbahn**

Die Bezeichner müssen so gewählt werden, dass kein Konflikt mit anderen Bezeichnern der Linearachsen

X Y Z Achsen

und Rundachsen wie

A2 B2 C2 Eulerwinkel bzw. RPY-Winkel

A3 B3 C3 Richtungsvektoren

A4 B4 C4 bzw. A5 B5 C5 Flächennormalenvektoren

A6 B6 C6 Drehvektoren bzw. A7 B7 C7 Zwischenpunktkoordinaten

oder anderen Interpolationsparameter entsteht.

7.3 Orientierungspolynome (PO[Winkel], PO[Koordinate])

Unabhängig davon, welche Polynominterpolation der G-Code Gruppe 1 gerade aktiv ist, können zwei verschiedene Typen von Orientierungspolynomen bis maximal 5. Grades bei einer Drei- bis Fünf-Achs-Transformation programmiert werden.

1. Polynome für **Winkel**: Voreilwinkel LEAD, Seitwärtswinkel TILT in Bezug auf die Ebene, die von Start- und Endorientierung aufgespannt wird.
2. Polynome für **Koordinaten**: XH, YH, ZH der zweiten Raumkurve für die Werkzeugorientierung eines Bezugspunktes auf dem Werkzeug.

Bei einer Sechs-Achs-Transformation kann zur Werkzeugorientierung zusätzlich die Drehung des Drehvektors THT mit Polynomen bis maximal 5. Grades für Drehungen des Werkzeugs selbst programmiert werden.

Syntax

Orientierungspolynome vom Typ 1 für Winkel

N... PO[PHI] = (a2, a3, a4, a5)

Drei- bis Fünf-Achs-Transformation

N... PO[PSI] = (b2, b3, b4, b5)

Orientierungspolynome vom Typ 2 für Koordinaten

N... PO[XH] = (xe, x2, x3, x4, x5)

N... PO[YH] = (ye, y2, y3, y4, y5)

N... PO[ZH] = (ze, z2, z3, z4, z5)

Bezeichner für die Koordinaten der zweiten Orientierungsbahn für die Werkzeugorientierung

Zusätzlich kann in beiden Fällen ein Polynom für die **Drehung** bei Sechs-Achs-Transformationen mit

N... PO[THT]=(c2, c3, c4, c5)

Bahnrelative Interpolation der Drehung

oder

N... PO[THT]=(d2, d3, d4, d5)

absoluter, relative und tangentielle Interpolation zur Orientierungsänderung

des Orientierungsvektors programmiert werden. Dies ist dann möglich, wenn die Transformation einen Drehvektor mit einem durch den Drehwinkel THETA programmierbaren und interpolierbaren Offset unterstützt.

Bedeutung

PO[PHI]	Winkel in der Ebene zwischen Start- und Endorientierung
PO[PSI]	Winkel der die Auskippung der Orientierung aus der Ebene zwischen Start- und Endorientierung beschreibt
PO[THT]	Drehwinkel der durch Drehung des Drehvektors einer der mit THETA programmierten G-Codes der Gruppe 54
PHI	Voreilwinkel LEAD
PSI	Seitwärtswinkel TILT
THETA	Drehung um die Werkzeugrichtung in Z
PO[XH]	X-Koordinate des Bezugspunktes auf dem Werkzeug
PO[YH]	Y-Koordinate des Bezugspunktes auf dem Werkzeug
PO[ZH]	Z-Koordinate des Bezugspunktes auf dem Werkzeug

Weitere Informationen

Orientierungspolynomen können nicht programmiert werden

- wenn die Splineinterpolationen ASPLINE, BSPLINE, CSPLINE aktiv sind. Polynome vom Typ1 für Orientierungswinkel sind für jede Interpolationsart außer Spline d.h. bei Linearinterpolation mit Eilgang G00 bzw. mit Vorschub G01 bei Polynominterpolation mit POLY und bei Kreis- bzw. Evolventeninterpolation mit G02, G03, CIP, CT, INVCW und INCCCW möglich. Polynome vom Typ2 für Orientierungskordinaten sind dagegen nur möglich, wenn Linearinterpolation mit Eilgang G00 bzw. mit Vorschub G01 oder Polynominterpolation mit POLY aktiv ist.
- wenn die Orientierung mittels Achsinterpolation ORIAXES interpoliert wird. In diesem Fall können direkt Polynome mit PO[A] und PO[B] für die Orientierungsachsen A und B programmiert werden.

Orientierungspolynome vom Typ 1 mit ORIVECT, ORIPLANE und ORICONxx

Bei Großkreisinterpolation und Kegelmantelinterpolation mit ORIVECT, ORIPLANE und ORICONxx sind nur Orientierungspolynome vom Typ 1 möglich.

Orientierungspolynome vom Typ 2 mit ORICURVE

Ist die Interpolation mit zusätzlicher Raumkurve ORICURVE aktiv, werden die kartesischen Komponenten des Orientierungsvektors interpoliert und es sind nur Orientierungspolynome vom Typ 2 möglich.

7.4 Drehungen der Werkzeugorientierung (ORIROTA, ORIROT, ORIROTT, ORIROTC, THETA)

Soll bei Maschinentypen mit beweglichem Werkzeug auch die Orientierung des Werkzeugs veränderbar sein, so wird jeder Satz mit einer Endorientierung programmiert. Abhängig von der Maschinenkinematik können entweder die Orientierungsrichtung der Orientierungsachsen oder die Drehrichtung des Orientierungsvektors THETA programmiert werden. Für diese Drehvektoren sind verschiedene Interpolationsarten programmierbar:

- ORIROTA: Drehwinkel zu einer absolut vorgegebenen Drehrichtung.
- ORIROT: Drehwinkel relativ zur Ebene zwischen Start- und Endorientierung.
- ORIROTT: Drehwinkel relativ zur Änderung des Orientierungsvektors.
- ORIROTC: Tangentialer Drehwinkel zur Bahntangente.

Syntax

Nur wenn die Interpolationsart ORIROTA aktiv ist, kann der Drehwinkel oder der Drehvektor auf die vier möglichen Arten wie folgt programmiert werden:

1. Direkt die Rundachspositionen A, B, C
2. Eulerwinkel (in Grad) über $A2, B2, C2$
3. RPY-Winkel (in Grad) über $A2, B2, C2$
4. Richtungsvektor über $A3, B3, C3$ (Drehwinkel mittels $THETA=<Wert>$)

Falls ORIROT oder ORIROTT aktiv sind, kann der Drehwinkel nur noch direkt mit THETA programmiert werden.

Eine Drehung kann auch allein in einem Satz programmiert werden, ohne dass eine Orientierungsänderung stattfindet. Dabei haben ORIROT und ORIROTT keine Bedeutung. In diesem Fall wird der Drehwinkel immer in Bezug zur absoluten Richtung interpretiert (ORIROTA).

N... ORIROTA	Interpolation des Drehvektors festlegen
N... ORIROT	
N... ORIROTT	
N... ORIROTC	
N... A3= B3= C3= THETA=<Wert>	Drehung des Orientierungsvektors festlegen
N... PO[THT]=(d_2, d_3, d_4, d_5)	Drehwinkel mit Polynom 5. Grades interpolieren

Bedeutung

ORIOTA:	Drehwinkel zu einer absolut vorgegebenen Drehrichtung
ORIOTR:	Drehwinkel relativ zur Ebene zwischen Start- und Endorientierung
ORIOTT:	Drehwinkel als tangentialer Drehvektor zur Orientierungsänderung
ORIOTC:	Drehwinkel als tangentialer Drehvektor zur Bahntangente
THETA:	Drehung des Orientierungsvektors
THETA=<Wert>:	Drehwinkel in Grad, der am Satzende erreicht wird
THETA=Θ _e :	Drehwinkel mit Endwinkel Θ _e des Drehvektors
THETA=AC (...):	Satzweise auf Maßangabe absolut umschalten
THETA=AC (...):	Satzweise auf Kettenmaßangabe umschalten
Θ _e :	Endwinkel des Drehvektors sowohl absolut mit G90 als auch relativ mit G91 (Kettenmaßangabe) ist aktiv
PO[THT]=(...):	Polynom für den Drehwinkel

Beispiel: Drehungen der Orientierungen

Programmcode	Kommentar
N10 TRAORI	; Orientierungstransformation aktivieren
N20 G1 X0 Y0 Z0 F5000	; Orientierung des Werkzeugs
N30 A3=0 B3=0 C3=1 THETA=0	; in Z-Richtung mit Drehwinkel 0
N40 A3=1 B3=0 C3=0 THETA=90	; in X-Richtung und Drehung um 90 Grad
N50 A3=0 B3=1 C3=0 PO[THT]=(180, 90)	; Orientierung
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; in Y-Richtung und Drehung auf 180 Grad
N70 ORIOTT	; bleibt konstant und Drehung auf 90 Grad
N80 A3=1 B3=0 C3=0 THETA=30	; Drehwinkel relativ zur Orientierungsänderung ; Drehvektor im Winkel 30 Grad zur X-Y Ebene

Bei der Interpolation von Satz N40 wird der Drehwinkel vom Startwert 0 Grad zum Endwert 90 Grad linear interpoliert. Im Satz N50 ändert sich der Drehwinkel von 90 Grad auf 180 Grad gemäß der Parabel $\theta(u) = +90u^2$. In N60 kann auch eine Drehung ausgeführt werden, ohne dass eine Orientierungsänderung stattfindet.

Bei N80 wird die Werkzeugorientierung von der Y-Richtung in X-Richtung gedreht. Dabei liegt die Orientierungsänderung in der X-Y Ebene und der Drehvektor bildet zu dieser Ebene einen Winkel von 30 Grad.

Weitere Informationen

ORIOTA

Der Drehwinkel THETA wird bezüglich einer absolut festgelegten Richtung im Raum interpoliert. Die Grunddrehrichtung erfolgt über Maschinendaten

ORIOTR

Der Drehwinkel THETA wird relativ zur Ebene, die von der Start- und Endorientierung aufgespannt wird, interpretiert.

ORIROTT

Der Drehwinkel `THETA` wird relativ zur Orientierungsänderung interpretiert. Für `THETA=0` wird der Drehvektor tangential zur Orientierungsänderung interpoliert und unterscheidet sich nur dann zu `ORIROTR`, wenn für die Orientierung mindestens ein Polynom für den "Kippwinkel `PSI`" programmiert wurde. Damit ergibt sich eine Orientierungsänderung, die nicht in der Ebene abläuft. Durch einen zusätzlich programmierten Drehwinkel `THETA` kann dann z. B. der Drehvektor so interpoliert werden, dass er immer einen bestimmten Wert zur Orientierungsänderung bildet.

ORIROTC

Der Drehvektor wird relativ zur Bahntangente mit einem durch den Winkel `THETA` programmierbaren Offset interpoliert. Für den Offsetwinkel kann dabei auch ein Polynom `PO[THETA]=(c2, c3, c4, c5)` maximal 5. Grades programmiert werden.

7.5 Bahnrelative Orientierungen

7.5.1 Orientierungsarten relativ zur Bahn

Mit dieser erweiterten Funktion wird die relative Orientierung nicht nur am Satzende, sondern über den gesamten Bahnverlauf erreicht. Es wird die im Vorgängersatz erreichte Orientierung mittels Großkreisinterpolation in die programmierte Endorientierung überführt. Grundsätzlich gibt es zwei Möglichkeiten die gewünschte Orientierung relativ zur Bahn zu programmieren:

1. Die Werkzeugorientierung als auch die Drehung des Werkzeugs wird mit `ORIPATH`, `ORPATHS` relativ zur Bahn interpoliert.
2. Der Orientierungsvektor wird wie bisher üblich programmiert und interpoliert. Mit `ORIROTC` wird die Drehung des Orientierungsvektors relativ zur Bahntangente angestellt.

Syntax

Die Interpolationsart der Orientierung und der Drehung des Werkzeugs wird programmiert mit:

N... <code>ORIPATH</code>	Bahnrelative Orientierung
N... <code>ORIPATHS</code>	Bahnrelative Orientierung mit Glättung des Orientierungsverlaufs
N... <code>ORIROTC</code>	Bahnrelative Interpolation des Drehvektors

Ein durch eine Ecke im Bahnverlauf hervorgerufener Knick der Orientierung kann mit `ORIPATHS` geglättet werden. Die Richtung und Weglänge der Abhebebewegung wird durch den Vektor mit den Komponenten `A8=X`, `B8=Y` `C8=Z` programmiert.

Mit `ORIPATH/ORIPATHS` können verschiedene Bezüge zur Bahntangente über die drei Winkel

- `LEAD=` Angabe Vorwärtswinkel bezogen auf die Bahn und Oberfläche
- `TILT=` Angabe von Seitwärtswinkel bezogen auf die Bahn und Oberfläche
- `THETA=` Drehwinkel

für den gesamten Bahnverlauf programmiert werden. Zum Drehwinkel `THETA` können mit `PO[THET]= (. . .)` zusätzlich Polynome maximal 5. Grades programmiert werden.

Hinweis

Maschinenhersteller

Bitte beachten Sie die Angaben des Maschinenherstellers. Über projektierbare Maschinen- und Settingdaten können zur Bahnrelativen Orientierungsart weitere Einstellungen vorgenommen werden. Weitere Erläuterungen siehe

Literatur:

/FB3/ Funktionshandbuch Sonderfunktionen; 3- bis 5-Achs-Transformation (F2), Kapitel "Orientierung"

Bedeutung

Die Interpolation der Winkel `LEAD` und `TILT` ist über Maschinedatum unterschiedlich einstellbar:

- Der mit `LEAD` und `TILT` programmierte Bezug der Werkzeugorientierung wird über den ganzen Satz hinweg eingehalten.
- Vorwärtswinkel `LEAD`: Drehung um die Richtung senkrecht zur Tangente und Normalenvektor `TILT`: Drehung der Orientierung um den Normalenvektor.
- Vorwärtswinkel `LEAD`: Drehung um die Richtung senkrecht zur Tangente und Normalenvektor Seitwärtswinkel `TILT`: Drehung der Orientierung um die Richtung der Bahntangente.
- Drehwinkel `THETA`: Drehung des Werkzeugs um sich selbst mit einer zusätzlichen dritten Rundachse als Orientierungsachse bei Sechs-Achs-Transformation.

Hinweis

Bahnrelative Orientierung zusammen mit `OSC`, `OSS`, `OSSE`, `OSD`, `OST` unzulässig

Die bahnrelative Orientierungsinterpolation `ORIPATH` bzw. `ORIPATHS` und `ORIOTC` kann nicht zusammen mit der Glättung des Orientierungsverlaufs mit einen der G-Codes aus der Gruppe 34 programmiert werden. Hierfür muss `OSOF` aktiv sein.

7.5.2 Bahnrelative Drehung der Werkzeugorientierung (`ORIPATH`, `ORIPATHS`, Drehwinkel)

Bei einer Sechs-Achs-Transformation kann zur Werkzeugorientierung beliebig im Raum auch das Werkzeug mit einer dritten Rundachse um sich selbst gedreht werden. Bei bahnrelativer Drehung der Werkzeugorientierung mit `ORIPATH` bzw. `ORIPATHS` kann die zusätzliche Drehung über den Drehwinkel `THETA` programmiert werden. Alternativ hierzu können die Winkel `LEAD` und `TILT` durch einen Vektor, der in der Ebene senkrecht zur Werkzeugrichtung liegt, programmiert werden.

Maschinenhersteller

Bitte beachten Sie die Angaben des Maschinenherstellers. Über Maschinendatum kann die Interpolation der Winkel LEAD und TILT unterschiedlich eingestellt werden.

Syntax**Drehung der Werkzeugorientierung und des Werkzeugs**

Die Werkzeugorientierungsart relativ zur Bahn wird mit ORIPATH oder ORIPATHS aktiviert.

N... ORIPATH	Orientierungsart bezogen auf die Bahn aktivieren
N... ORIPATHS	Orientierungsart bezogen auf die Bahn mit Glättung des Orientierungsverlaufs aktivieren

Aktivierung der drei möglichen Winkel mit Drehwirkung:

N... LEAD=	Winkel für die programmierten Orientierung relativ zum Flächennormalenvektor
N... TILT=	Winkel für die programmierte Orientierung in der Ebene senkrecht zur Bahntangente relativ zum Flächennormalenvektor
N... THETA=	Drehwinkel relativ zur Orientierungsänderung um die Werkzeugrichtung der dritten Rundachse

Die Werte der Winkel am Satzende werden mit LEAD=Wert, TILT=Wert bzw. THETA=Wert programmiert. Zusätzlich zu den konstanten Winkeln können für alle drei Winkel Polynome maximal 5. Grades programmiert werden.

N... PO[PHI]=(a2, a3, a4, a5)	Polynom für den Voreilwinkel LEAD
N... PO[PSI]=(b2, b3, b4, b5)	Polynom für den Seitwärtswinkel TILT
N... PO[THT]=(d2, d3, d4, d5)	Polynom für den Drehwinkel THETA

Bei der Programmierung können die höheren Polynomkoeffizienten, die Null sind, weggelassen werden. Beispiel PO[PHI]=a2 ergibt für den Voreilwinkel LEAD eine Parabel.

Bedeutung**Bahnrelative Werkzeugorientierung**

ORIPATH:	Werkzeugorientierung bezogen auf die Bahn
ORIPATHS:	Werkzeugorientierung bezogen auf die Bahn, Knick im Orientierungsverlauf wird geglättet
LEAD:	Winkel relativ zum Flächennormalenvektor, in der von Bahntangente und Flächennormalenvektor aufgespannten Ebene
TILT:	Drehung der Orientierung um die Z-Richtung bzw. Drehung um die Bahntangente
THETA:	Drehung um die Werkzeugrichtung nach Z
PO[PHI]:	Orientierungspolynom für den Voreilwinkel LEAD
PO[PSI]:	Orientierungspolynom für den Seitwärtswinkel TILT
PO[THT]:	Orientierungspolynom für den Drehwinkel THETA

Hinweis

Drehwinkel THETA

Für die Drehung des Werkzeugs mit dritter Rundachse als Orientierungsachse um sich selbst, ist eine Sechs-Achs-Transformation erforderlich.

7.5.3 Bahnrelative Interpolation der Werkzeugdrehung (ORIROTC, THETA)

Interpolation mit Drehvektoren

Zur mit ORIROTC programmierten Drehung des Werkzeugs relativ zur Bahntangente kann der Drehvektor auch mit einem durch den Drehwinkel THETA programmierbaren Offset interpoliert werden. Dabei kann für den Offsetwinkel mit PO[THT] ein Polynom bis maximal 5. Grades programmiert werden.

Syntax

N... ORIROTC	Drehung des Werkzeugs relativ zur Bahntangente anstellen
N... A3= B3= C3= THETA=Wert	Drehung des Orientierungsvektors festlegen
N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)	Offsetwinkel mit Polynom maximal 5. Grades interpolieren

Eine Drehung kann auch allein in einem Satz programmiert werden, ohne dass eine Orientierungsänderung stattfindet.

Bedeutung

Bahnrelative Interpolation der Drehung des Werkzeugs bei Sechs-Achs-Transformation

ORIROTC:	tangentialem Drehvektor zur Bahntangente anstellen
THETA=Wert:	Drehwinkel in Grad, der am Satzende erreicht wird
THETA=θe:	Drehwinkel mit Endwinkel θ _e des Drehvektors
THETA=AC (...):	Satzweise auf Maßangabe absolut umschalten
THETA=IC (...):	Satzweise auf Kettenmaßangabe umschalten
PO[THT]=(c2, c3, c4, c5):	Offsetwinkel mit Polynom 5. Grades interpolieren

Hinweis

Interpolation des Drehvektors ORIOTC

Soll gegen die Orientierungsrichtung des Werkzeugs auch die Drehung des Werkzeugs relativ zur Bahntangente angestellt werden, dann ist dies nur bei einer Sechs-Achs-Transformation möglich.

Bei aktiven ORIOTC

Der Drehvektor ORIOTA kann nicht programmiert werden. Im Falle einer Programmierung wird der ALARM 14128 "Absolutprogrammierung der Werkzeugdrehung bei aktivem ORIOTC" ausgegeben.

Orientierungsrichtung des Werkzeugs bei Drei- bis Fünf-Achs-Transformation

Die Orientierungsrichtung des Werkzeugs kann wie bei der Drei- bis Fünf-Achs-Transformation gewohnt über Eulerwinkel bzw. RPY-Winkel oder des Richtungsvektoren programmiert werden. Auch sind Orientierungsänderungen des Werkzeugs im Raum durch Programmierung der Großkreisinterpolation ORIVECT, der linearen Interpolation der Orientierungsachsen ORIAXES, alle Interpolationen auf einer Kegelmantelfläche ORICONxx sowie der Interpolation zusätzlich zur Raumkurve mit zwei Kontaktpunkten des Werkzeugs ORICURVE möglich.

G :	Angabe der Bewegungsart der Rundachsen
X, Y, Z:	Angabe der Linearachsen
ORIAXES:	Lineare Interpolation der Maschinen- oder Orientierungsachsen
ORIVECT:	Großkreisinterpolation (identisch mit ORIPLANE)
ORIMKS:	Drehung im Maschinenkoordinatensystem
ORIWKS:	Drehung im Werkstückkoordinatensystem Beschreibung siehe Kap. Drehungen der Werkzeugorientierung
A= B= C=:	Programmierung der Maschinenachspoition
ORIEULER:	Orientierungsprogrammierung über Euler-Winkel
ORIRPY:	Orientierungsprogrammierung über RPY-Winkel
A2= B2= C2=:	Winkelprogrammierung virtueller Achsen
ORIVIRT1: ORIVIRT2:	Orientierungsprogrammierung über virtuelle Orientierungsachsen (Definition 1), Festlegung nach MD \$MC_ORIAX_TURN_TAB_1 (Definition 2), Festlegung nach MD \$MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Richtungsvektorprogrammierung der Richtungsachse
ORIPLANE:	Interpolation in der Ebene (Großkreisinterpolation)
ORICONCW:	Interpolation auf einer Kegelmantelfläche im Uhrzeigersinn
ORICONCCW:	Interpolation auf einer Kegelmantelfläche gegen Uhrzeigersinn
ORICONTO:	Interpolation auf einer Kegelmantelfläche tangentialer Übergang
A6= B6= C6=:	Programmierung der Drehachse des Kegels (normierter Vektor)
NUT=winkel	Öffnungswinkel des Kegels in Grad
NUT=+179	Verfahrwinkel kleiner oder gleich 180 Grad
NUT=-181	Verfahrwinkel größer oder gleich 180 Grad
ORICONIO:	Interpolation auf einer Kegelmantelfläche

A7= B7= C7=:	Zwischenorientierung (Programmierung als normierter Vektor)
ORICURVE XH YH ZH z.B. mit Polynome PO[XH]=(xe, x2, x3, x4, x5)	Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeuges. Außer den jeweiligen Endpunkten sind zusätzliche Raumkurven Polynome programmierbar.

Hinweis

Wird die Werkzeugorientierung mit aktiven ORIXES über die Orientierungsachsen interpoliert, dann wird die bahnrelative Anstellung des Drehwinkels nur am Satzende erfüllt.

7.5.4 Glättung des Orientierungsverlaufs (ORIPATHS A8=, B8=, C8=)

Bei beschleunigungsstetigen Orientierungsänderungen an der Kontur sind Unterbrechungen der Bahnbewegungen, die besonders an einer Ecke der Kontur auftreten können unerwünscht. Der sich hieraus ergebende Knick im Orientierungsverlauf kann durch Einfügen eines eigenen Zwischensatzes geglättet werden. Die Orientierungsänderung erfolgt dann beschleunigungsstetig, wenn während der Umorientierung auch ORIPATHS aktiv ist. In dieser Phase kann eine Abhebebewegung des Werkzeuges durchgeführt werden.

Maschinenhersteller

Beachten Sie bitte die Hinweise des Maschinenherstellers zu gegebenenfalls vordefinierten Maschinendaten und Settingdaten mit denen diese Funktion aktiviert wird.

Über Maschinendatum ist einstellbar, wie der Abhebevektor interpretiert wird:

1. Im Werkzeugkoordinatensystem wird die Z-Koordinate durch die Werkzeugrichtung definiert.
2. Im Werkstückkoordinatensystem wird die Z-Koordinate durch die aktive Ebene definiert.

Weitere Erläuterungen zur Funktion "Bahnrelative Orientierung" siehe

Literatur:

Funktionshandbuch Sonderfunktionen; Mehrachstransformationen (F2)

Syntax

Für stetige Werkzeugorientierungen bezogen auf die gesamte Bahn sind an einer Ecke der Kontur weitere Programmierangaben erforderlich. Die Richtung und die Weglänge dieser Bewegung wird durch den Vektor mit den Komponenten A8=X, B8=Y, C8=Z programmiert:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Bedeutung

ORIPATHS:	Werkzeugorientierung bezogen auf die Bahn, ein Knick im Orientierungsverlauf wird geglättet.
A8= B8= C8=:	Vektorkomponenten für Richtung und Weglänge
X, Y, Z:	Abhebebewegung in Werkzeugrichtung

Hinweis**Programmierung des Richtungsvektors A8, B8, C8**

Ist die Länge dieses Vektors gleich Null erfolgt keine Abhebebewegung.

ORIPATHS

Die bahnbezogene Werkzeugorientierung wird mit ORIPATHS aktiv. Anderenfalls wird die Orientierung mittels linearer Großkreisinterpolation von der Start- zur Endorientierung überführt.

7.6 Komprimierung der Orientierung (COMPON, COMPCURV, COMPCAD, COMPSURF)

NC-Programme, in denen eine Orientierungstransformation (TRAORI) aktiv ist und Werkzeugorientierungen (egal welcher Art) programmiert sind, können unter Einhaltung von vorgegeben Toleranzen komprimiert werden.

Programmierung**Werkzeugorientierung**

Falls eine Orientierungstransformation (TRAORI) aktiv ist, kann bei 5-Achs Maschinen die Werkzeugorientierung folgendermaßen (kinematikunabhängig) programmiert werden:

- Programmierung des Richtungs**v**ektors über:
A3=<...> B3=<...> C3=<...>
- Programmierung der Euler**w**inkel bzw. RPY-**W**inkel über:
A2=<...> B2=<...> C2=<...>

Drehung des Werkzeugs

Bei **6-Achs** Maschinen kann zusätzlich zur Werkzeugorientierung noch die Drehung des Werkzeugs programmiert werden.

Die Programmierung des Drehwinkels erfolgt mit:

THETA=<...>

Siehe " Drehungen der Werkzeugorientierung (Seite 333) ".

Hinweis

NC-Sätze, in denen zusätzlich eine Drehung programmiert ist, sind nur dann komprimierbar, falls sich der Drehwinkel **linear** ändert. D. h. für den Drehwinkel darf kein Polynom mit $PO[THT]=(\dots)$ programmiert sein.

Allgemeine Form eines komprimierbaren NC-Satzes

Die allgemeine Form eines komprimierbaren NC-Satzes kann daher wie folgt aussehen:

N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>

bzw.

N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>

Hinweis

Die Positionswerte können direkt (z. B. X90) oder indirekt über Parameterzuweisungen (z. B. $X=R1*(R2+R3)$) angegeben werden.

Programmierung der Werkzeugorientierung durch Rundachspositionen

Die Werkzeugorientierung kann auch durch Rundachspositionen angegeben sein, z. B. in der Form:

N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>

In diesem Fall wird die Komprimierung auf zwei unterschiedliche Arten durchgeführt, abhängig davon ob eine Großkreisinterpolation durchgeführt wird oder nicht. Wenn keine Großkreisinterpolation stattfindet, dann wird die komprimierte Orientierungsänderung durch axiale Polynome für die Rundachsen in üblicher Weise dargestellt.

Konturgenauigkeit

Abhängig vom eingestellten Kompressionsmodus (MD20482 \$MC_COMPRESSOR_MODE) werden für die Geometrieachsen und Orientierungsachsen bei der Komprimierung entweder die projizierten achsspezifischen Toleranzen (MD33100 \$MA_COMPRESS_POS_TOL) oder die folgenden über Settingdaten einstellbaren kanalspezifischen Toleranzen wirksam:

SD42475 \$SC_COMPRESS_CONTUR_TOL (Maximale Konturabweichung)

SD42476 \$SC_COMPRESS_ORI_TOL (Maximale Winkelabweichung für die Werkzeugorientierung)

SD42477 \$SC_COMPRESS_ORI_ROT_TOL (Maximale Winkelabweichung für den Drehwinkel des Werkzeugs) (nur bei 6-Achs Maschinen verfügbar)

Literatur:

Funktionshandbuch Grundfunktionen; 3- bis 5-Achs-Transformation (F2),
Kapitel: "Komprimierung der Orientierung"

Aktivierung / Deaktivierung

Kompressor-Funktionen werden eingeschaltet durch die modalen G-Befehle COMPON, COMPCURV, COMPCAD bzw. COMPSURF.

7.6 Komprimierung der Orientierung (COMPON, COMPCURV, COMPCAD, COMPSURF)

Beendet wird die Kompressor-Funktion mit COMPOF.

Siehe " NC-Satz-Kompression (COMPON, COMPCURV, COMPCAD) (Seite 245) ".

Hinweis

Die Orientierungsbewegung wird nur komprimiert bei aktiver Großkreisinterpolation (d. h. die Änderung der Werkzeugorientierung erfolgt in der Ebene, die von Start- und Endorientierung aufgespannt wird).

Eine Großkreisinterpolation wird unter den folgenden Bedingungen durchgeführt:

- MD21104 \$MC_ORI_IPO_WITH_G_CODE = 0,
ORIWKS ist aktiv und
Orientierung ist mittels Vektoren programmiert (mit A3, B3, C3 bzw. A2, B2, C2).
- MD21104 \$MC_ORI_IPO_WITH_G_CODE = 1 und
ORIVECT bzw. ORIPLANE ist aktiv.
Die Werkzeugorientierung kann entweder als Richtungsvektor oder mit
Rundachspalten programmiert sein. Ist einer der G-Codes ORICON_{xx} oder
ORICURVE aktiv oder sind Polynome für die Orientierungswinkel (PO[PHI] und PO[PSI])
programmiert, wird keine Großkreisinterpolation durchgeführt.

Beispiel

Im nachfolgenden Programmbeispiel wird ein Kreis, der durch einen Polygonzug angenähert ist, komprimiert. Die Werkzeugorientierung bewegt sich dabei synchron dazu auf einem Kegelmantel. Obwohl die aufeinanderfolgenden programmierten Orientierungsänderungen un stetig verlaufen, generiert die Kompressor-Funktion einen glatten Verlauf der Orientierung.

Programmierung	Kommentar
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	; Maximale Abweichung der Kontur = 0.05 mm
\$SC_COMPRESS_ORI_TOL=5	; Maximale Abweichung der Orientierung = 5 Grad
TRAORI	
COMPCURV	; Es wird ein Kreis gefahren, der aus Polygo- nen gebildet wird. Die Orientierung bewegt sich dabei auf einem Kegel um die Z-Achse mit einem Öffnungswinkel von 45 Grad.
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	

7.7 Glättung des Orientierungsverlaufs (ORISON, ORISOF)

Programmierung	Kommentar
N140 ENDFOR	

7.7 Glättung des Orientierungsverlaufs (ORISON, ORISOF)

Mit der Funktion "Glättung des Orientierungsverlaufs (ORISON)" können Schwankungen der Orientierung über mehrere Sätze hinweg geglättet werden. Dadurch wird ein glatter Verlauf sowohl der Orientierung als auch der Kontur erzielt.

Voraussetzung

Die Funktion "Glättung des Orientierungsverlaufs (ORISON)" ist nur in Systemen mit 5/6-Achs-Transformation verfügbar.

Syntax

```
ORISON
...
ORISOF
```

Bedeutung

ORISON:	Glättung des Orientierungsverlaufs EIN	
	Wirksamkeit:	modal
ORISOF:	Glättung des Orientierungsverlaufs AUS	
	Wirksamkeit:	modal

Settingdaten

Die Glättung des Orientierungsverlaufs erfolgt unter Einhaltung:

- einer vorgegebenen maximalen Toleranz (maximale Winkelabweichung der Werkzeugorientierung in Grad)
und
- eines vorgegebenen maximalen Bahnwegs.

Diese Vorgaben werden über Settingdaten definiert:

- SD42678 \$SC_ORISON_TOL (Toleranz für die Glättung des Orientierungsverlaufs)
- SD42680 O\$SC_ORISON_DIST (Bahnweg für die Glättung des Orientierungsverlaufs)

Beispiel

Programmcode	Kommentar
...	

Programmcode	Kommentar
TRAORI()	; Einschalten der Orientierungstransformation.
ORISON	; Einschalten der Orientierungsglättung.
\$SSC_ORISON_TOL=1.0	; Toleranz der Orientierungsglättung = 1,0 Grad.
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	; Ausschalten der Orientierungsglättung.
...	

Die Orientierung wird um 90 Grad in der XZ-Ebene von -45 bis +45 Grad geschwenkt. Durch die Glättung des Orientierungsverlaufs erreicht die Orientierung nicht mehr die maximalen Winkelwerte von -45 bzw. +45 Grad.

Weitere Informationen

Anzahl der Sätze

Die Glättung des Orientierungsverlaufs erfolgt über eine projektierte Anzahl von Sätzen, die im Maschinendatum MD28590 \$MC_MM_ORISON_BLOCKS hinterlegt ist.

Hinweis

Wird die Glättung des Orientierungsverlaufs mit ORISON aktiviert, ohne dass ausreichend Satzspeicher dafür projektiert wurde (MD28590 < 4), dann erfolgt eine Alarmmeldung und die Funktion kann nicht ausgeführt werden.

Maximale Satzweglänge

Der Orientierungsverlauf wird nur in solchen Sätzen geglättet, deren Verfahrensweg kleiner ist als die projektierte maximale Satzweglänge (MD20178 \$MC_ORISON_BLOCK_PATH_LIMIT). Sätze mit längeren Verfahrenswegen unterbrechen die Glättung und werden wie programmiert abgefahren.

7.8 Kinematische Transformation

7.8.1 Stirnseitentransformation einschalten (TRANSMIT)

Die Stirnseitentransformation (TRANSMIT) wird im Teileprogramm oder Synchronaktion über die Anweisung `TRANSMIT` eingeschaltet.

Syntax

`TRANSMIT`

`TRANSMIT (<n>)`

Bedeutung

<code>TRANSMIT:</code>	TRANSMIT mit erstem TRANSMIT-Datensatz einschalten
<code>TRANSMIT (n):</code>	TRANSMIT mit <n>-tem TRANSMIT-Datensatz einschalten

Hinweis

Eine im Kanal aktive Transformation `TRANSMIT` wird ausgeschaltet durch:

- Transformation ausschalten: `TRAFOOF`
- Einschalten einer anderen Transformation: z. B. `TRACYL`, `TRAANG`, `TRAORI`

7.8.2 Zylindermanteltransformation einschalten (TRACYL)

Die Zylindermanteltransformation (`TRACYL`) wird im Teileprogramm oder Synchronaktion über die Anweisung `TRACYL` eingeschaltet.

Syntax

`TRACYL (<d>)`

`TRACYL (<d>, <n>)`

`TRACYL (<d>, <n>, <k>)`

Bedeutung

<code>TRACYL (<d>):</code>	<code>TRACYL</code> mit erstem <code>TRACYL</code> -Datensatz und Arbeitsdurchmesser <code><d></code> einschalten
<code>TRACYL (<d>, <n>):</code>	<code>TRACYL</code> mit <n>-tem <code>TRACYL</code> -Datensatz und Arbeitsdurchmesser <code><d></code> einschalten
<code><d>:</code>	Bezugs- bzw. Arbeitsdurchmesser Der Wert muss größer 1 sein.

<n>:	TRACYL-Datensatznummer (optional)	
	Wertebereich:	1, 2
<k>:	Der Parameter <k> ist nur relevant für Transformationstyp 514	
	k = 0:	ohne Nutwandkorrektur
	k = 1:	mit Nutwandkorrektur
	Wird der Parameter nicht angegeben, wirkt die parametrisierte Grundstellung: \$MC_TRACYL_DEFAULT_MODE_<n> mit <n> = TRACYL-Datensatznummer	

Hinweis

Eine im Kanal aktive Transformation TRACYL wird ausgeschaltet durch:

- Transformation ausschalten: TRAFOOF
- Einschalten einer anderen Transformation: z. B. TRAANG, TRANSMIT, TRAORI

Beispiel

Programmcode	Kommentar
...	
N40 TRACYL(40.)	; TRACYL mit erstem TRACYL-Datensatz und Arbeitsdurchmesser 40 mm einschalten.
...	

Weitere Informationen

Programmstruktur

Ein Teileprogramm zum Fräsen einer Nut mit TRACYL-Transformation 513 (TRACYL mit Nutwandkorrektur) besteht in der Regel aus folgenden Schritten:

1. Werkzeug anwählen.
2. TRACYL anwählen.
3. Passende Koordinatenverschiebung (FRAME) anwählen.
4. Positionieren.
5. OFFN programmieren.
6. WRK anwählen.
7. Anfahrsatz (Einfahren der WRK und Anfahren der Nutwand).
8. Kontur der Nutmittenlinie.
9. WRK abwählen.
10. Abfahrsatz (Ausfahren der WRK und Wegfahren von der Nutwand).
11. Positionieren.

12. TRAFOOF.

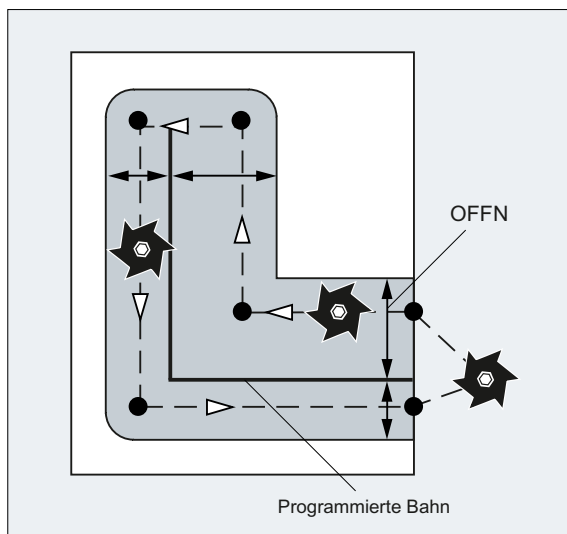
13. Ursprüngliche Koordinatenverschiebung (FRAME) wieder anwählen.

Kontur-Offset (OFFN)

Um mit der TRACYL-Transformation 513 Nuten zu fräsen, wird im Teileprogramm die Mittellinie der Nut und über die Adresse OFFN **die halbe Nutbreite** programmiert.

Um eine Beschädigung der Nutwand zu vermeiden, wird OFFN erst mit angewählter Werkzeugradiuskorrektur wirksam.

Eine Änderung von OFFN innerhalb des Teileprogramms ist möglich. Damit kann die Nutmittellinie aus der Mitte verschoben werden:



Hinweis

OFFN sollte mindestens so groß wie der Werkzeugradius sein, um eine Beschädigung der gegenüberliegenden Nutwand auszuschließen.

Hinweis

OFFN mit TRACYL wirkt sich anders aus als ohne TRACYL. Da OFFN auch ohne TRACYL bei aktiver WRK eingerechnet wird, sollte OFFN nach TRAFOOF wieder zu Null gesetzt werden.

<p>ACHTUNG</p> <p>Wirkung von OFFN ist vom Transformationstyp abhängig</p> <p>Bei der TRACYL-Transformation 513 (TRACYL mit Nutwandkorrektur) wird für OFFN die halbe Nutbreite programmiert.</p> <p>Bei der TRACYL-Transformation 512 (TRACYL ohne Nutwandkorrektur) wirkt der Wert von OFFN dagegen als Aufmass zur WRK.</p>
--

Werkzeugradiuskorrektur (WRK)

Bei der TRACYL-Transformation 513 wird die WRK nicht relativ zur Nutwand, sondern zur programmierten Mitte der Nut eingerechnet. Damit das Werkzeug links von der Nutwand fährt, ist statt G41 die Anweisung G42 zu programmieren oder der Wert von OFFN mit negativem Vorzeichen anzugeben.

Werkzeugdurchmesser

Mit TRACYL und einem Werkzeug, dessen Durchmesser kleiner als die Nutbreite ist, wird nicht dieselbe Nutwandgeometrie erzeugt, wie mit einem Werkzeug, dessen Durchmesser gleich der Nutbreite ist. Zur Verbesserung der Genauigkeit wird empfohlen, den Werkzeugdurchmesser nur wenig kleiner als die Nutbreite zu wählen.

Achsnutzung**Hinweis**

Folgende Achsen können nicht als Positionierachse bzw. Pendelachse verwendet werden:

- die Geometrieachse in Umfangsrichtung der Zylindermantelfläche (Y-Achse)
- die zusätzliche Linearachse bei Nutwandkorrektur (Z-Achse)

7.8.3 Schiefwinkeltransformation einschalten (TRAANG)

Die Schiefwinkeltransformation (TRAANG) wird im Teileprogramm oder Synchronaktion über die Anweisung TRAANG eingeschaltet.

Syntax

```
TRAANG
TRAANG ( )
TRAANG ( , <n> )
TRAANG ( <α> )
TRAANG ( <α> , <n> )
```

Bedeutung

TRAANG / TRAANG () :	TRAANG mit erstem TRAANG-Datensatz und zuletzt gültigem Winkel <α> einschalten
TRAANG (, <n>) :	TRAANG mit <n>-tem TRAANG-Datensatz und zuletzt gültigem Winkel <α> einschalten
TRAANG (<α>) :	TRAANG mit erstem TRAANG-Datensatz und Winkel <α> einschalten
TRAANG (<α> , <n>) :	TRAANG mit <n>-tem TRAANG-Datensatz und Winkel <α> einschalten

< α >:	Winkel der schrägstehenden Achse (optional)	
	Wertebereich:	$-90^\circ < \alpha < +90^\circ$
	Wird der Parameter nicht angegeben, wirkt die parametrisierte Grundstellung: \$MC_TRAANG_ANGLE_<n> mit <n> = TRAANG-Datensatznummer	
<n>:	TRAANG-Datensatznummer (optional)	
	Wertebereich:	1, 2

Hinweis

Eine im Kanal aktive Transformation TRAANG wird ausgeschaltet durch:

- Transformation ausschalten: TRAFOOF
- Einschalten einer anderen Transformation: z. B. TRACYL, TRANSMIT, TRAORI

Beispiel

Programmcode	Kommentar
...	
N20 TRAANG(45)	; TRAANG mit erstem TRAANG-Datensatz und Winkel 45° einschalten.
...	

7.8.4 Schräges Einstechen an Schleifmaschinen (G5, G7)

Die G-Befehle G7 und G5 dienen zur Vereinfachung der Programmierung für das schräge Einstechen an Schleifmaschinen mit Transformation "Schräge Achse (TRAANG)", so dass beim Einstechen ausschließlich die schräge Achse verfährt.

Zu programmieren ist dabei nur die gewünschte Endposition der Einstechbewegung in X und Z. Die zugehörige Startposition wird von der NC bei G7, ausgehend von der aktuellen Position der X-Achse, der programmierten Endposition und dem Winkel α der schrägen Achse, berechnet und angefahren.

Die Startposition ergibt sich aus dem Schnittpunkt der beiden Geraden:

- Gerade parallel zur Z-Achse im Abstand der aktuellen Position der X-Achse
- Gerade parallel zur schrägen Achse durch die programmierte Endposition

Mit dem nachfolgenden G5 wird die schräge Achse auf die programmierte Endposition verfahren.

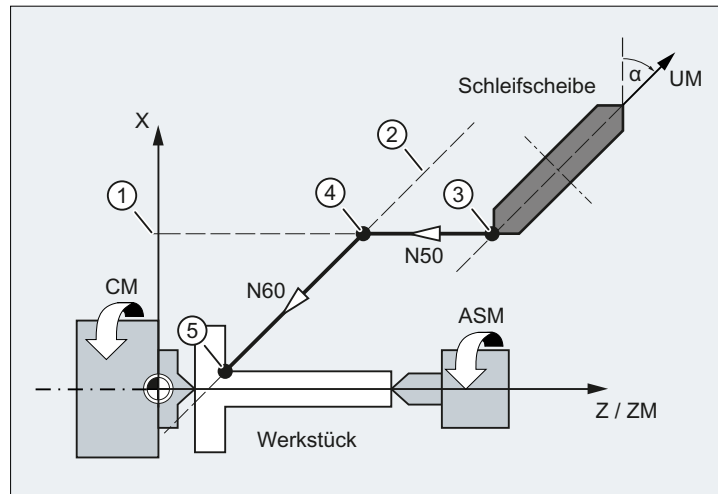
Syntax

```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

Bedeutung

G7:	Startposition zum schrägen Einstechen berechnen und anfahren
G5:	Verfahren der schrägen Achse auf die programmierte Endposition
<Endpos_X>:	Endposition der X-Achse
<Endpos_Z>:	Endposition der Z-Achse

Beispiel



- ① Parallele zur Z-Achse im Abstand der aktuellen Position der X-Achse
- ② Parallele zur schrägen Achse durch die programmierte Endposition
- ③ Ausgangsposition
- ④ Einstechen: Startposition
- ⑤ Einstechen: Endposition
- X Geometrieachse
- Z Geometrieachse
- ZM Maschinenachse
- UM Maschinenachse

Programmcode	Kommentar
N... G18	; XZ-Ebene anwählen.
N40 TRAANG(45.0)	; Transformation TRAANG einschalten, Winkel = 45°.
N50 G7 X40 Z70 F4000	; Startposition berechnen und anfahren.
N60 G5 X40 F100	; Schräge Achse auf Endposition verfahren.
N70 ...	

7.9 Verkettete Transformation einschalten (TRACON)

Eine verkettete Transformation wird im Teileprogramm oder Synchronaktion über die Anweisung `TRACON` eingeschaltet.

7.9 Verkettete Transformation einschalten (TRACON)

Syntax

```
TRACON(<Trafo_Nr>,<Par_1>,<Par_2>,...)
...
TRAFOOF
```

Bedeutung

TRACON:	Verkettete Transformation einschalten Eine zuvor aktivierte andere Transformation wird durch TRACON () implizit ausgeschaltet.		
<Trafo_Nr>:	Nummer der verketteten Transformation		
	Typ:	INT	
	Wertebe- reich:	0 ... 2	
	Wert:	0, 1	erste/einzige verkettete Transformation
		2	zweite verkettete Transformation
keine Angabe		gleichbedeutend mit 0 bzw. 1	
Hinweis: Werte ungleich 0, 1, 2 erzeugen einen Fehleralarm.			
<Par_1>,<Par_2>,...:	Parameter für die miteinander verketteten Transformationen (z. B. Winkel der schrägen Achse) Bei nicht gesetzten Parametern werden die Voreinstellungen oder die zuletzt benutzten Parameter wirksam. Durch Kommasetzung muss dafür gesorgt werden, dass die angegebenen Parameter in der Reihenfolge ausgewertet werden, in der sie erwartet werden, wenn für vorher stehende Parameter Voreinstellungen wirken sollen. Insbesondere muss bei Angabe mindestens eines Parameters vor diesem ein Komma stehen, auch wenn die Angabe von <Trafo_Nr> nicht notwendig ist, also beispielsweise TRACON (, 3.7).		
TRAFOOF:	Die zuletzt eingeschaltete (verkettete) Transformation ausschalten		

Beispiel

Programmcode	Kommentar
...	
N230 TRACON(1,45.)	; Erste verkettete Transformation einschalten. ; Die vorher aktive Transformation wird automatisch abgewählt. ; Der Parameter für die schräge Achse ist 45°.
...	
N330 TRACON(2,40.)	; Zweite verkettete Transformation einschalten. ; Der Parameter für die schräge Achse ist 40°.
...	
N380 TRAFOOF	; Zweite verkettete Transformation ausschalten.
...	

7.10 Kartesisches PTP-Fahren

7.10.1 Kartesisches PTP-Fahren ein-/ausschalten (PTP, PTPG0, PTPWOC, CP)

Das kartesische Punkt-zu-Punkt- bzw. PTP-Fahren wird im NC-Programm mit den Befehlen der G-Gruppe 49 ein-/ausgeschaltet.

Die Befehle sind modal wirksam. Voreinstellung ist das Verfahren mit einer kartesischen Bahnbewegung (CP).

Anders als bei CP wird bei aktivem PTP-Fahren nur noch eine Transformation des kartesischen Zielpunkts durchgeführt und die Maschinenachsen werden synchron verfahren.

Damit der kartesische Zielpunkt eindeutig in Maschinenachswerte umgerechnet werden kann, sind zusätzlich zu den Positions- und Winkelangaben noch Informationen notwendig, die die Achsstellungen kennzeichnen. Diese Angaben erfolgen über die einstellbaren Adressen STAT (Seite 354) und TU (Seite 357).

Voraussetzung

Transformation TRAORI/TRANSMIT ist aktiv.

Syntax

```
PTP / PTPG0 / PTPWOC
...
CP
```

Bedeutung

PTP:	Punkt-zu-Punkt-Bewegung PTP einschalten Die programmierte kartesische Position in G0- und G1-Sätzen wird mit einer Synchronachsbewegung angefahren.
PTPG0:	Punkt-zu-Punkt-Bewegung PTPG0 einschalten Nur in G0-Sätzen wird die programmierte kartesische Position mit einer Synchronachsbewegung angefahren. In G1-Sätzen wird auf die Bahnbewegung CP umgeschaltet.
PTPWOC:	Punkt-zu-Punkt-Bewegung PTPWOC einschalten Wie PTP, allerdings ohne Ausgleichsbewegungen, die durch die Bewegungen von Rundachsen und Orientierungsachsen verursacht sind.
CP:	Punkt-zu-Punkt-Bewegung ausschalten und Bahnbewegung CP einschalten Mit CP wird eine kartesische Bahnbewegung ausgeführt.

Beispiele

Siehe:

- Beispiel 1: PTP und TRAORI (Seite 359)
- Beispiel 2: PTPG0 und TRANSMIT (Seite 360)

7.10.2 Stellung der Gelenke angeben (STAT)

Die Positionsangabe mit kartesischen Koordinaten und die Angabe der Werkzeugorientierung reichen nicht aus, um die Maschinenstellung eindeutig festzulegen, da bei gleicher Werkzeugorientierung mehrere Gelenkstellungen möglich sind. Je nachdem, um welche Kinematik es sich handelt, existieren bis zu 8 unterschiedliche Gelenkstellungen. Diese unterschiedlichen Gelenkstellungen sind transformationspezifisch.

Um Mehrdeutigkeiten zu vermeiden, wird die Stellung der Gelenke unter der Adresse STAT angegeben.

Hinweis

Die Steuerung berücksichtigt programmierte STAT-Werte nur bei PTP-Bewegungen. Bei CP-Bewegungen werden sie ignoriert, da beim Verfahren mit aktiver Transformation ein Stellungswechsel normalerweise nicht möglich ist. Beim Verfahren mit aktivem CP wird die Stellung für den Zielpunkt vom Startpunkt übernommen.

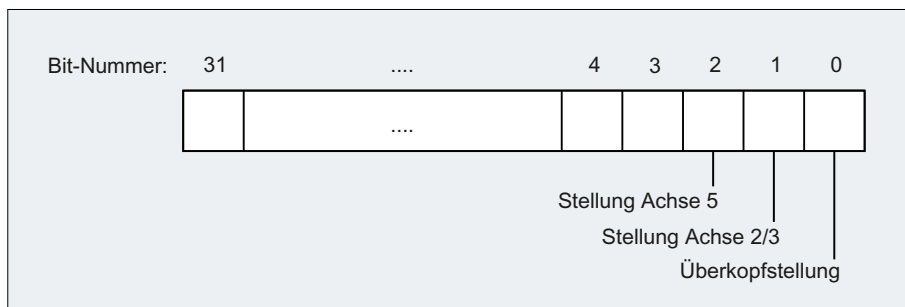
Syntax

STAT=<Wert>

Bedeutung

STAT:	Einstellbare Adresse zur Angabe der Gelenkstellungen
<Wert>:	Binär- oder Dezimalwert Enthält für jede der möglichen Stellungen ein Bit. Die Bedeutung der Bits wird von der jeweiligen Transformation festgelegt.

Für die Transformationen, die im "Transformationspaket Handling" enthalten sind, gilt folgende Zuordnung:



Am Beispiel eines Roboters mit einer 6-Achs-Gelenkinematik soll die Bedeutung der Bits veranschaulicht werden:

Bit 0	Position des Schnittpunkts der Handachsen (A4, A5, A6)	
	= 0	Grundbereich Der Roboter befindet sich im Grundbereich, wenn der X-Wert des Schnittpunkts der Handachsen, bezogen auf das A1-Koordinatensystem, positiv ist.
	= 1	Überkopfbereich Der Roboter befindet sich im Überkopfbereich, wenn der X-Wert des Schnittpunkts der Handachsen, bezogen auf das A1-Koordinatensystem, negativ ist.
Bit 1	Position von Achse 3 Der Winkel, bei dem sich der Wert von Bit 1 ändert, ist abhängig vom Robotertyp. Für Roboter, deren Achsen 3 und 4 sich schneiden, gilt:	
	= 0	$A3 < 0^\circ$
	= 1	$A3 \geq 0^\circ$
	Hinweis: Bei Robotern mit einem Offset zwischen Achse 3 und 4 ist der Winkel, bei dem sich der Wert von Bit 1 ändert, von der Größe dieses Offsets abhängig.	
Bit 2	Position von Achse 5	
	= 0	$A5 \leq 0^\circ$
	= 1	$A5 > 0^\circ$
Bit 3	Wird nicht genutzt. Wert ist immer 0.	
Bit 4	Gibt an, ob der Punkt mit einem absolutgenauen Roboter geteacht wurde oder nicht.	
	= 0	Der Punkt wurde nicht mit einem absolutgenauen Roboter geteacht.
	= 1	Der Punkt wurde mit einem absolutgenauen Roboter geteacht.
	Hinweis: Der Punkt kann unabhängig vom Wert des Bits sowohl mit absolutgenauen als auch mit nicht absolutgenauen Robotern abgefahren werden. Das Bit 4 dient nur der Information und hat keinen Einfluss darauf, wie die Robotersteuerung den Punkt berechnet. Bit 4 kann daher auch außer Acht gelassen werden, wenn ein Roboter offline programmiert wird.	

TRANSMIT

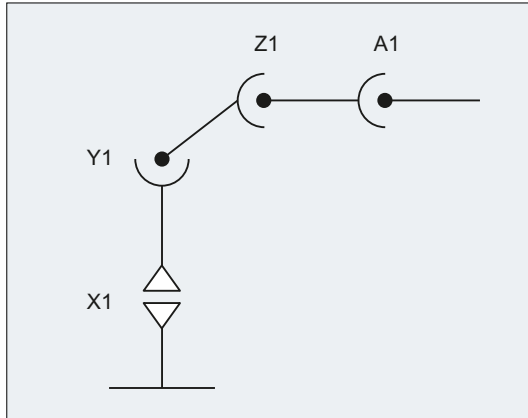
Bei TRANSMIT wird Adresse STAT benutzt, um die Mehrdeutigkeit hinsichtlich des Pols aufzulösen.

Wenn die Rundachse um 180° drehen muss bzw. die Kontur bei CP durch den Pol führen würde gilt:

Bit 0	Nur relevant bei $\$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1$ bzw. 2:	
	= 0	Rundachse verfährt um $+180^\circ$ bzw. dreht im Uhrzeigersinn.
	= 1	Rundachse verfährt um -180° bzw. dreht gegen den Uhrzeigersinn.
Bit 1	Nur relevant bei $\$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0$:	
	= 0	Es wird durch den Pol gefahren. Die Rundachse dreht nicht.
	= 1	Es wird um den Pol gedreht. Dabei ist Bit 0 von STAT relevant.

Beispiele

Die folgenden Abbildungen zeigen Beispiele für die Mehrdeutigkeit durch unterschiedliche Gelenkstellungen bei einer 6-Achs-Gelenkinematik. Erst durch die STAT-Angaben werden die Gelenkstellungen eindeutig.



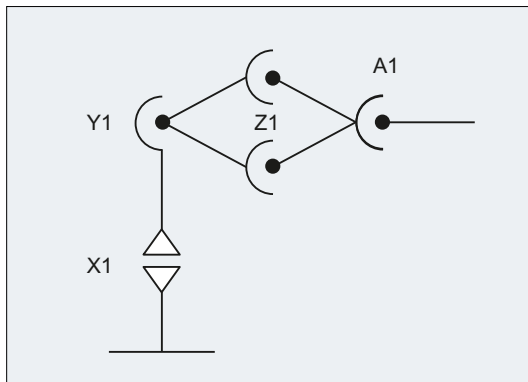
Stellung 1:

X1 = 0°
Y1 = 45°
Z1 = 45°

Stellung 2:

X1 = -180°
Y1 = -45°
Z1 = -45°
STAT = 'B01'

Mehrdeutigkeit Überkopfbereich



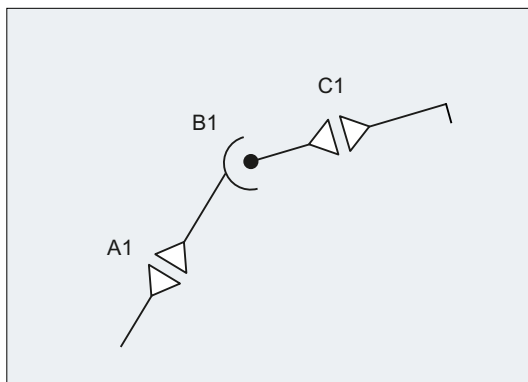
Stellung 1:

Y1 = -30°
Z1 = 60°
A1 = -30°
STAT = 'B00'

Stellung 2:

Y1 = 30°
Z1 = -60°
A1 = 30°
STAT = 'B10'

Mehrdeutigkeit Ellbogen oben oder unten



Stellung 1:

A1 = 0°
B1 = 40°
C1 = 0°
STAT = 'B00'

Stellung 2:

A1 = 180°
B1 = -40°
C1 = 180°
STAT = 'B100'

Mehrdeutigkeit Achse B1

7.10.3 Vorzeichen der Achswinkel angeben (TU)

Um bei rotatorischen Achsen auch Achswinkel, die größer $+180^\circ$ oder kleiner -180° sind, ohne besondere Verfahrensstrategie (z. B. Zwischenpunkte) anfahren zu können, müssen unter der einstellbaren Adresse TU die Vorzeichen der Achswinkel angegeben werden.

Hinweis

Die Steuerung berücksichtigt programmierte TU-Werte nur bei PTP-Bewegungen. Bei CP-Bewegungen werden sie ignoriert.

Syntax

TU=<Wert>

Bedeutung

TU:	Einstellbare Adresse zur Angabe der Achswinkelvorzeichen		
<Wert>:	Binär- oder Dezimalwert Für jede Achse, die in die Transformation eingeht, gibt es ein Bit, das das Vorzeichen des Achswinkels (θ) und damit die Fahrerrichtung anzeigt.		
Bit	= 0	$0^\circ \leq \theta < 360^\circ$	
	= 1	$-360^\circ < \theta < 0^\circ$	

Am Beispiel eines Roboters mit einer 6-Achs-Gelenkinematik soll die Bedeutung der Bits veranschaulicht werden:

Bit 0	Vorzeichen des Achswinkels A1	
	= 0	$A1 \geq 0^\circ$
	= 1	$A1 < 0^\circ$
Bit 1	Vorzeichen des Achswinkels A2	
	= 0	$A2 \geq 0^\circ$
	= 1	$A2 < 0^\circ$
Bit 2	Vorzeichen des Achswinkels A3	
	= 0	$A3 \geq 0^\circ$
	= 1	$A3 < 0^\circ$
Bit 3	Vorzeichen des Achswinkels A4	
	= 0	$A4 \geq 0^\circ$
	= 1	$A4 < 0^\circ$
Bit 4	Vorzeichen des Achswinkels A5	
	= 0	$A5 \geq 0^\circ$
	= 1	$A5 < 0^\circ$
Bit 5	Vorzeichen des Achswinkels A6	
	= 0	$A6 \geq 0^\circ$
	= 1	$A6 < 0^\circ$

TU=19 (entspricht TU='B010011) würde also bedeuten:

Bit	Wert		Achswinkel
0	= 1	⇒	A1 < 0°
1	= 1	⇒	A2 < 0°
2	= 0	⇒	A3 ≥ 0°
3	= 0	⇒	A4 ≥ 0°
4	= 1	⇒	A5 < 0°
5	= 0	⇒	A6 ≥ 0°

Hinweis

Bei Achsen mit einem Verfahrbereich > ±360° wird immer auf kürzestem Weg verfahren, da die Achsstellung durch die TU-Information nicht eindeutig bestimmbar ist.

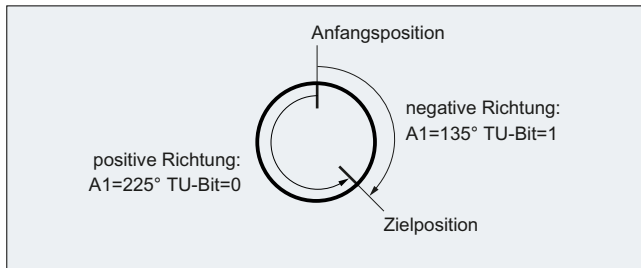
Wird bei einer Position kein TU programmiert, wird immer auf kürzestem Weg verfahren.

TRANSMIT

Beim PTP-Fahren bei aktivem TRANSMIT ist die Adresse TU ohne Bedeutung!

Beispiel

Die in der folgenden Abbildung angegebene Rundachsposition kann in negativer oder in positiver Richtung angefahren werden. Unter der Adresse A1 wird die Winkelposition programmiert. Erst durch die TU-Angabe wird die Verfahrrichtung eindeutig.



7.10.4 Beispiel 1: PTP und TRAORI

Beispiel 1

Nachfolgend ist eine Sequenz aus einem Beispielprogramm für folgende Kinematik aufgeführt:

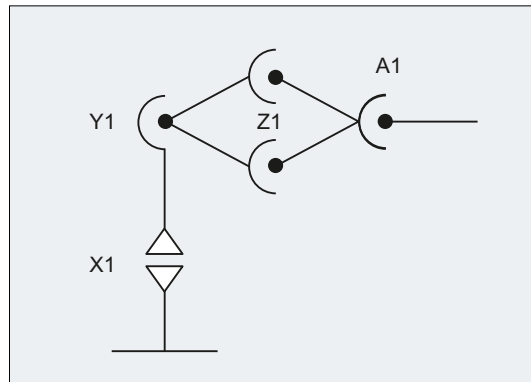


Bild 7-2 Ellbogen oben oder unten

Programmcode	Kommentar
N10 G0 X0 Y-30 Z60 A-30 F10000	; Ausgangsstellung → Ellbogen oben
N20 TRAORI(1)	; Transformation ein
N30 X1000 Y0 Z400 A0	
N40 X1000 Z500 A0 STAT='B10' TU='B100' PTP	; Umorientierung ohne Transformation ; → Ellbogen unten
N50 X1200 Z400 CP	; Transformation wieder aktiv
N60 X1000 Z500 A20	
N70 M30	

Beispiel 2

PTP-Fahren bei generischer 5-Achs-Transformation

Annahme: Es liegt eine rechtwinklige CA-Kinematik zu Grunde.

Programmcode	Kommentar
TRAORI	; Transformation CA-Kinematik ein
PTP	; PTP-Fahren einschalten
N10 A3=0 B3=0 C3=1	; Rundachspositionen C=0 A=0
N20 A3=1 B3=0 C3=1	; Rundachspositionen C=90 A=45
N30 A3=1 B3=0 C3=0	; Rundachspositionen C=90 A=90
N40 A3=1 B3=0 C3=1 STAT=1	; Rundachspositionen C=270 A=-45

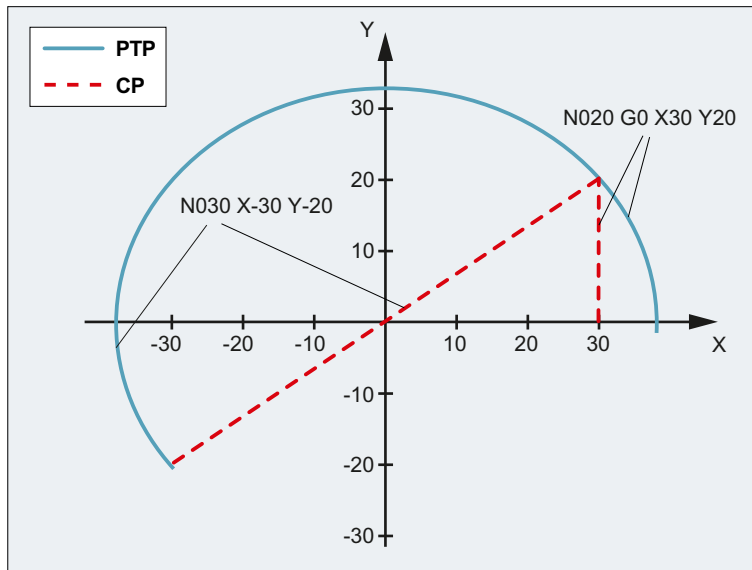
Eindeutige Anfahrstellung der Rundachsposition auswählen:

Im Satz N40 fahren dabei die Rundachsen durch die Programmierung von STAT=1 den längeren Weg von ihrem Startpunkt (C=90, A=90) zum Endpunkt (C=270, A=-45). Mit STAT=0 würden die Rundachsen dagegen den kürzeren Weg zum Endpunkt (C=90, A=45) fahren.

7.10.5 Beispiel 2: PTPG0 und TRANSMIT

Beispiel 1

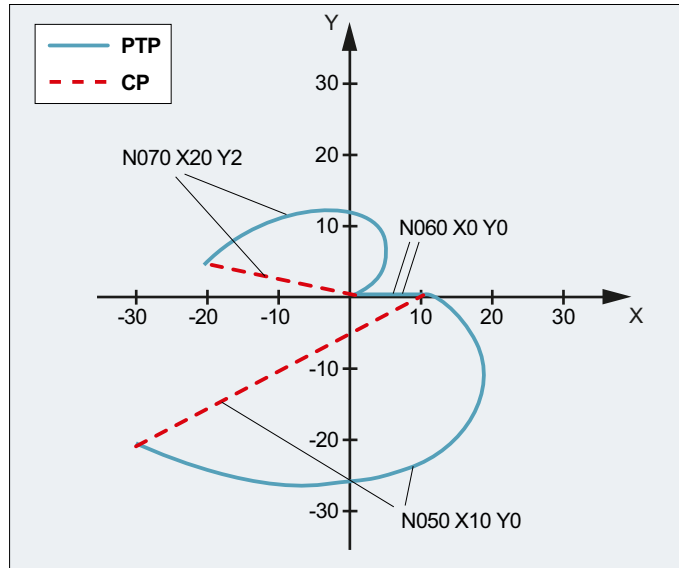
Umfahren des Pols mit PTPG0 und TRANSMIT



Programmcode	Kommentar
N001 G0 X30 Z0 F10000 T1 D1 G90	; Ausgangsstellung Absolutmaß
N002 SPOS=0	
N003 TRANSMIT	; Transformation TRANSMIT
N010 PTPG0	; Zu jedem G0-Satz automatisch PTP und danach wieder CP.
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Beispiel 2

Herausfahren aus dem Pol mit PTPG0 und TRANSMIT



Programmierung	Kommentar
N001 G0 X90 Z0 F10000 T1 D1 G90	; Ausgangsstellung
N002 SPOS=0	
N003 TRANSMIT	; Transformation TRANSMIT
N010 PTPG0	; Zu jedem G0-Satz automatisch PTP und danach wieder 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 Randbedingungen bei der Anwahl einer Transformation

Funktion

Die Anwahl von Transformationen ist über Teileprogramm bzw. MDA möglich. Dabei ist zu beachten:

- Ein Bewegungszwischensatz wird nicht eingefügt (Fasen/Radien).
- Eine Spline-Satzfolge muss abgeschlossen sein; wenn nicht, erscheint eine Meldung.
- Werkzeugfeinkorrektur muss abgewählt sein (FTOCOF); wenn nicht, erscheint eine Meldung.
- Werkzeugradiuskorrektur muss abgewählt sein (G40); wenn nicht, erscheint eine Meldung.
- Eine aktivierte Werkzeuglängenkorrektur wird von der Steuerung in die Transformation übernommen.
- Der vor der Transformation wirksame aktuelle Frame wird von der Steuerung abgewählt.
- Eine aktive Arbeitsfeldbegrenzung wird für die von der Transformation betroffenen Achsen von der Steuerung abgewählt (entspricht WALIMOF).
- Schutzbereichsüberwachung wird abgewählt.
- Bahnsteuerbetrieb und Überschleifen werden unterbrochen.
- Alle in dem Maschinendatum angegebenen Achsen müssen satzbezogen synchronisiert sein.
- Getauschte Achsen werden zurückgetauscht; wenn nicht, erscheint eine Meldung.
- Bei abhängigen Achsen wird eine Meldung ausgegeben.

Werkzeugwechsel

Ein Werkzeugwechsel ist nur bei abgewählter Werkzeugradiuskorrektur zulässig.

Ein Wechsel der Werkzeuglängenkorrektur und eine An-/Abwahl der Werkzeugradiuskorrektur darf nicht im selben Satz programmiert sein.

Framewechsel

Alle Anweisungen, die sich nur auf das Basis-Koordinatensystem beziehen, sind erlaubt (FRAME, Werkzeugradiuskorrektur). Ein Framewechsel bei G91 (Kettenmaß) wird aber – anders als bei inaktiver Transformation – nicht gesondert behandelt. Das zu fahrende Inkrement wird im Werkstück-Koordinatensystem des neuen Frames ausgewertet – unabhängig davon, welches Frame im Vorgängersatz wirkte.

Ausschlüsse

Von der Transformation betroffene Achsen können nicht verwendet werden:

- als Preset-Achse (Alarm),
- für das Fixpunktanfahren (Alarm),
- zum Referieren (Alarm).

7.12 Transformation abwählen (TRAFOOF)

Mit der vordefinierten Prozedur TRAFOOF werden alle aktiven Transformationen und Frames ausgeschaltet.

Hinweis

Für die Abwahl der Transformation gelten dieselben Randbedingungen (Seite 362) wie für die Anwahl.

Danach benötigte Frames müssen durch erneute Programmierung aktiv geschaltet werden.

Syntax

```
...  
TRAFOOF
```

Bedeutung

TRAFOOF:	Alle aktiven Transformationen/Frames ausschalten
----------	--

Kinematische Ketten

8.1 Löschen von Komponenten (DELOBJ)

Die Funktion `DELOBJ()` "löscht" Komponenten durch Zurücksetzen der zugeordneten Systemvariablen auf ihren Defaultwert:

- Elemente von kinematischen Ketten
- Schutzbereiche, Schutzbereichselemente und Kollisionspaare
- Transformationsdaten

Syntax

```
[<RetVal>=] DELOBJ (<CompType> [ , , , <NoAlarm> ] )  
[<RetVal>=] DELOBJ (<CompType> , <Index1> [ , , <NoAlarm> ] )  
[<RetVal>=] DELOBJ (<CompType> [ , <Index1> ] [ , <Index2> ] [ , <NoAlarm> ] )
```

Bedeutung

DELOBJ:	Löschen von Elementen von kinematischen Ketten, Schutzbereichen, Schutzbereichselementen, Kollisionspaaren und Transformationsdaten
<CompType>:	Typ der zu löschenden Komponente
	Datentyp: STRING
	Wert: "KIN_CHAIN_ELEM" Bedeutung: Systemvariablen aller kinematischen Elemente: \$NK_...
	Wert: "KIN_CHAIN_SWITCH" Bedeutung: Systemvariable \$NK_SWITCH[<i>]</i>
	Wert: "KIN_CHAIN_ALL" Bedeutung: Alle kinematischen Elemente und Schalter. Gleichbedeutend mit dem sukzessiven Aufruf von DELOBJ mit "KIN_CHAIN_ELEM" und "KIN_CHAIN_SWITCH"
	Wert: "PROT_AREA" Bedeutung: Systemvariablen der Schutzbereiche: <ul style="list-style-type: none"> • \$NP_PROT_NAME • \$NP_CHAIN_NAME • \$NP_CHAIN_ELEM • \$NP_1ST_PROT
	Wert: "PROT_AREA_ELEM" Bedeutung: Systemvariablen der Schutzbereichselemente von Maschinenschutzbereichen und/oder automatischen Werkzeugschutzbereichen: <ul style="list-style-type: none"> • \$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
	Wert: "PROT_AREA_COLL_PAIRS" Bedeutung: Systemvariablen der Kollisionspaare: <ul style="list-style-type: none"> • \$NP_COLL_PAIR • \$NP_SAFETY_DIST
	Wert: "PROT_AREA_ALL" Bedeutung: Alle Schutzbereiche, Schutzbereichselemente und Kollisionspaare (Systemvariablen \$NP_...) Gleichbedeutend mit dem sukzessiven Aufruf von DELOBJ mit "PROT_AREA", "PROT_AREA_ELEM" und "PROT_AREA_COLL_PAIRS"
	Wert: "TRAFO_DATA" Bedeutung: Systemvariablen aller Transformationen \$NT_...

<Index1>:	Index der ersten zu löschenden Komponente (optional)	
	Datentyp:	INT
	Defaultwert:	-1
	Wertebereich:	$-1 \leq x \leq$ (maximale Anzahl projektierter Komponenten -1)
	Wert	Bedeutung
	0, 1, 2,	Index der zu löschenden Komponente.
	-1	Alle Komponenten des angegebenen Typs werden gelöscht. <Index2> wird nicht ausgewertet.
<Index2>:	Index der letzten zu löschenden Komponenten (optional) Ist <Index2> nicht programmiert, werden nur die Systemvariablen der in <Index1> referenzierten Komponente gelöscht.	
	Datentyp:	INT
	Defaultwert:	Nur die Systemvariablen der in <Index1> referenzierten Komponente werden gelöscht.
	Wertebereich:	<Index1> < x ≤ (max. Anzahl projektierter Komponenten -1)
<NoAlarm>:	Alarmunterdrückung (optional)	
	Datentyp:	BOOL
	Defaultwert:	FALSE
	Wert	Bedeutung
	FALSE	Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt.
	TRUE	Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert
<RetVal>:	Rückgabewert der Funktion	
	Datentyp:	INT
	Wertebereich:	0, -1, -2, ... -7
	Wert	Bedeutung
	0	Kein Fehler aufgetreten.
	-1	Aufruf der Funktion ohne Parameter. Mindestens der Parameter <CompType> muss angegeben werden.
	-2	<CompType> bezeichnet unbekannte Komponente
	-3	<Index1> ist kleiner als -1
	-4	<Index1> ist größer als die projektierte Anzahl Komponenten
	-5	<Index1> hat beim Löschen einer Komponentengruppe einen Wert ungleich -1
	-6	<Index2> ist kleiner als <Index1>
-7	<Index2> ist größer als die projektierte Anzahl Komponenten	

8.2 Indexermittlung per Namen (NAMETOINT)

In Systemvariablenfeldern vom Typ STRING sind anwenderspezifische Namen eingetragen. Anhand des Bezeichners der Systemvariablen und des Namens, ermittelt die Funktion `NAMETOINT()` den zum Namen gehörenden Indexwert, unter dem er im Systemvariablenfeld abgelegt ist.

Syntax

```
<RetVal> = NAMETOINT (<SysVar>, <Name> [, <NoAlarm>])
```

Bedeutung

NAMETOINT:	Ermittlung des Systemvariablenindex	
<SysVar>:	Name des Systemvariablenfeldes vom Typ STRING	
	Datentyp:	STRING
	Wertebereich:	Namen aller Systemvariablenfelder der NC vom Typ STRING
<Name>:	Zeichenkette bzw. Name, zu dem der Systemvariablenindex ermittelt werden soll.	
	Datentyp:	STRING
<NoAlarm>:	Alarmunterdrückung (optional)	
	Datentyp:	BOOL
	Defaultwert:	FALSE
	Wert	Bedeutung
	TRUE	Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert
FALSE	Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt.	
<RetVal>:	Systemvariablenindex oder Fehlermeldung	
	Datentyp:	INT
	Wertebereich:	-1 ≤ x ≤ (max. Anzahl projektierter Komponenten -1)
	Wert	Bedeutung
	≥ 0	Der gesuchte Name wurde unter dem angegebenen Systemvariablenindex gefunden.
-1	Der gesuchte Name wurde nicht gefunden bzw. es ist ein Fehler aufgetreten.	

Beispiel

Programmcode	Kommentar
DEF INT INDEX	
\$NP_PROT_NAME[27] = "Abdeckung"	
...	
INDEX = NAMETOINT("\$NP_PROT_NAME", "Abdeckung")	; INDEX == 27

Kollisionsvermeidung mit kinematischen Ketten

Hinweis

Schutzbereiche

Die in den nachfolgenden Kapiteln genannten Schutzbereiche beziehen sich auf die Funktion "Geometrische Maschinenmodellierung"

Literatur:

Funktionshandbuch Sonderfunktionen, Kapitel "Geometrische Maschinenmodellierung"

9.1 Prüfen auf Kollisionspaar (COLLPAIR)

Bei der Kollisionsvermeidung mit kinematischen Ketten kann mit der Funktion COLLPAIR ermittelt werden, ob zwei Schutzbereiche ein Kollisionspaar bilden, d. h. ob sie, wenn beide aktiv sind, auf Kollision überwacht werden.

Syntax

```
[<RetVal>=]COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>])
```

Bedeutung

COLLPAIR:	Prüfen auf Zugehörigkeit zu einem Kollisionspaar			
<RetVal>:	Rückgabewert der Funktion			
	Datentyp:	INT		
	Wert:	≥ 0	Die beiden Schutzbereiche bilden ein Kollisionspaar. Rückgabewert == Kollisionspaarindex m	
		-1	Es wurden entweder weniger als zwei Strings angegeben, oder mindestens einer der beiden ist der Null-String.	
		-2	Der im ersten Parameter angegebene Schutzbereich wurde nicht gefunden.	
		-3	Der im zweiten Parameter angegebene Schutzbereich wurde nicht gefunden.	
		-4	Keiner der beiden angegebenen Schutzbereiche wurde gefunden.	
-5		Beide angegebenen Schutzbereiche wurden gefunden, aber nicht gemeinsam in einem Kollisionspaar.		
<Name_1>:	Name des ersten Schutzbereichs			
	Datentyp:	STRING		
	Wertebereich:	Parametrierte Schutzbereichsnamen		

<Name_2>:	Name des zweiten Schutzbereiches		
	Datentyp:	STRING	
	Wertebereich:	Parametrierte Schutzbereichsnamen	
<NoAlarm>:	Alarmunterdrückung (optional)		
	Datentyp:	BOOL	
	Wert:	FALSE	Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt.
		TRUE	Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert
Defaultwert:	FALSE		

Literatur

Funktionshandbuch Sonderfunktionen; Kapitel K9: Kollisionsvermeidung

9.2 Neuberechnung des Kollisionsmodells anfordern (PROTA)

Durch das Schreiben von Systemparametern für kinematische Ketten, geometrische Maschinenmodellierung und/oder Kollisionsvermeidung im Teileprogramm oder über die BTSS-Schnittstelle kann das Kollisionmodell geändert werden. Damit die Änderungen wirksam werden, muss mit der Prozedur PROTA eine Neuberechnung des Modells angefordert werden.

Schutzbereichsstatus

Durch den Aufruf der Prozedur PROTA wird eine Neuberechnung des Kollisionmodells **ohne** Änderung des Schutzraumstatus durchgeführt. Ein **neu** definierter Schutzbereich befindet sich danach im Initialisierungsstatus. Ein Schutzbereich, dessen Name geändert wurde, gilt **nicht** als neu definiert. Sein Status bleibt unverändert.

Durch den Aufruf der Prozedur PROTA mit Parameter "R" wird eine Neuberechnung des Kollisionmodells **mit** Änderung des Schutzraumstatus durchgeführt. Dabei werden alle Schutzbereiche wieder in ihren jeweiligen Initialisierungsstatus entsprechend \$NP_INIT_STAT versetzt.

Hinweis

Werkzeugwechsel und automatisch erzeugte Werkzeugschutzbereiche

Nach einem Werkzeugwechsel wird bei automatisch erzeugten Werkzeugschutzbereichen das Kollisionsmodells implizit immer neu berechnet.

Syntax

```
PROTA
PROTA (<Par>)
```

Bedeutung

PROTA:	Neuberechnung des Kollisionsmodells anfordern	
	Vorlaufstopp:	ja
<Par>:	Parameter	
	Datentyp:	STRING
	Wertebereich:	"R" Mit der Neuberechnung des Kollisionsmodells werden alle Schutzbereiche wieder in ihren jeweiligen Initialisierungsstatus versetzt.

Randbedingungen**Simulation**

Die Prozedur PROTA darf in Teileprogrammen nicht im Zusammenhang mit der Simulation (simNCK) verwendet werden.

Beispiel:

Programmcode	Kommentar
...	
IF \$P_SIM == FALSE	; IF Simulation nicht aktiv
PROTA	; THEN Kollisionsmodell neu berechnen
ENDIF	; ENDIF
...	

Literatur

Funktionshandbuch Sonderfunktionen; Kapitel K9: Kollisionsvermeidung

9.3 Schutzbereichszustand setzen (PROTS)

Bei der Kollisionsvermeidung mit kinematischen Ketten kann mit der Prozedur PROTS der Zustand von Schutzbereichen gesetzt werden.

Syntax

```
PROTS (<State>)
PROTS (<State>, <Name_1>)
PROTS (<State>, <Name_1>, ..., <Name_n>)
```

Bedeutung

PROTS:	Setzen des Schutzbereichszustandes	
	Vorlaufstopp:	nein
	Alleine im Satz:	ja

9.4 Abstandsbestimmung zweier Schutzbereiche (PROTD)

<State>: <Name_1> ... <Name_n>:	Status, auf den die angegebenen Schutzbereiche gesetzt werden sollen		
	Datentyp:	CHAR	
	Wert:	"A"oder "a"	Status: Aktiv
		"I"oder "i"	Status: Inaktiv
		"P"oder "p"	Status: Voraktiviert (Aktivierung erfolgt über NC/PLC-Nahtstelle)
"R"oder "r"		Status = \$NP_INIT_STAT (Initialisierungsstatus)	
Name eines oder mehrerer Schutzbereiche, die auf den angegebenen Status gesetzt werden sollen			
Hinweis: Die maximale Anzahl von Schutzbereichen, die als Parameter angegeben werden können, ist nur abhängig von der maximal möglichen Anzahl von Zeichen pro Programmzeile.			
Datentyp:	STRING		
Wertebereich:	Parametrisierte Schutzbereichsnamen Hinweis: Ist kein Name angegeben, wird der angegebenen Status für alle definierten Schutzbereiche gesetzt.		

Literatur

Funktionshandbuch Sonderfunktionen; Kapitel K9: Kollisionsvermeidung

9.4 Abstandsbestimmung zweier Schutzbereiche (PROTD)

Bei der Kollisionsvermeidung mit kinematischen Ketten kann mit der Funktion PROTD der Abstand von zwei Schutzbereichen berechnet werden.

Funktionseigenschaften:

- Die Abstandsberechnung erfolgt unabhängig vom Status der Schutzbereiche (aktiviert, deaktiviert, voraktiviert).
- Die Abstandsberechnung erfolgt mit den am Ende des Vorgängersatzes gültigen Positionen.
- Überlagerungen, die im Hauptlauf eingerechnet werden (z. B. DRF-Verschiebung oder externe Nullpunktverschiebungen) gehen mit den zum **Interpretationszeitpunkt** der Funktion gültigen Werten in die Abstandsberechnung ein.

Hinweis

Synchronisation

Bei Verwendung der Funktion PROTD liegt es ausschließlich in der Verantwortung des Anwenders, Haupt- und Vorlauf gegebenenfalls mittels Vorlaufstopp STOPRE zu synchronisieren.

Funktionswert bei Kollision

Eine Kollision zweier Schutzbereiche liegt vor, wenn für ihren Abstand gilt:

- $0,0 < \text{Abstand} < \text{MD10619 } \$\text{MN_COLLISION_TOLERANCE}$ (Toleranz für Kollisionsprüfung)

Liegt zwischen den angegebenen Schutzbereichen eine Kollision vor, liefert die Funktion PROTD den Wert 0,0.

Hinweis**Sicherheitsabstand**

Der Sicherheitsabstand für die Kollisionsprüfung (MD10622 $\$ \text{MN_COLLISION_SAFETY_DIST}$) wird bei der Abstandsrechnung nicht berücksichtigt.

Syntax

```
[<RetVal>=] PROTD(<Name_1>,<Name_2>,VAR <Vector>[,<System>])
```

```
[<RetVal>=] PROTD(, ,VAR <Vector>[,<System>])
```

Bedeutung

PROTD:	Berechnet den Abstand der beiden angegebenen Schutzbereiche.	
	Vorlaufstopp:	nein
	Alleine im Satz:	ja
<RetVal>:	Rückgabewert der Funktion: Absolutwert des Abstandes der beiden Schutzbereiche oder 0,0 bei Kollision (siehe oben: Absatz Kollision)	
	Datentyp:	REAL
	Wertebereich:	$0,0 \leq x \leq +\text{max. REAL-Wert}$
<Name_1>, <Name_2>:	Namen der beiden Schutzbereiche deren Abstand zueinander berechnet werden soll. (optional)	
	Datentyp:	STRING
	Wertebereich:	Parametrierte Schutzbereichnamen
	Defaultwert:	Sind keine Schutzbereiche angegeben, berechnet die Funktion den aktuell kleinsten Abstand aus allen im Kollisionsmodell enthaltenen aktivierten und voraktivierten Schutzbereichen.
<Vector>:	Rückgabewert: 3-dimensionaler Abstandsvektor von Schutzbereich <Name_2> zu Schutzbereich <Name_1> mit:	
	<ul style="list-style-type: none"> • <Vector>[0]: X-Koordinate im Weltkoordinatensystem • <Vector>[1]: Y-Koordinate im Weltkoordinatensystem • <Vector>[2]: Z-Koordinate im Weltkoordinatensystem 	
	Bei Kollision: <Vector> == Nullvektor	
	Datentyp:	VAR REAL[3]
	Wertebereich:	<Vector>[n]: $0,0 \leq x \leq \pm \text{max. REAL-Wert}$

<System>:	Maßsystem (inch / metrisch) für Abstand und Abstandsvektor (optional)		
	Datentyp:	BOOL	
	Wert:	FALSE	Maßsystem entsprechend der aktuell aktiven G-Funktion aus G-Gruppe 13 (G70, G71, G700, G710).
		TRUE	Maßsystem entsprechend des eingestellten Grundsystems:MD52806 \$MN_ISO_SCALING_SYSTEM
Defaultwert:	FALSE		

Literatur

Funktionshandbuch Sonderfunktionen; Kapitel K9: Kollisionsvermeidung

Werkzeugkorrekturen

10.1 Korrekturspeicher

Aufbau des Korrekturspeichers

Jedes Datenfeld ist mit einer T- und D-Nummer aufrufbar (außer "Flache D-Nr.") und enthält neben den geometrischen Angaben für das Werkzeug noch weitere Einträge, z. B. den Werkzeugtyp.

Flache D-Nummern-Struktur

Die "flache D-Nummern-Struktur" wird verwendet, wenn die Werkzeugverwaltung außerhalb des NCK erfolgt. In diesem Fall werden die D-Nummern mit den zugehörigen Werkzeugkorrektursätzen ohne Zuordnung zu Werkzeugen angelegt.

Im Teileprogramm kann weiterhin T programmiert werden. Dieses T hat aber keinen Bezug zur programmierten D-Nummer.

Anwender-Schneidendaten

Über Maschinendatum können Anwender-Schneidendaten konfiguriert werden. Bitte beachten Sie die Angaben des Maschinenherstellers.

Werkzeugparameter

Hinweis

Einzelne Werte im Korrekturspeicher

Die einzelnen Werte des Korrekturspeichers P1 bis P25 sind über Systemvariable vom Programm les- und schreibbar. Alle übrigen Parameter sind reserviert.

Die Werkzeugparameter \$TC_DP6 bis \$TC_DP8, \$TC_DP10 und \$TC_DP11 sowie \$TC_DP15 bis \$TC_DP17, \$TC_DP19 und \$TC_DP20 haben abhängig vom Werkzeugtyp eine andere Bedeutung.

Werkzeugparameter Nummer (DP)	Bedeutung der Systemvariablen	Bemerkung
\$TC_DP1	Werkzeugtyp	Übersicht siehe Liste
\$TC_DP2	Schneidenlage	nur für Drehwerkzeuge
Geometrie	Längenkorrektur	
\$TC_DP3	Länge 1	Verrechnung nach
\$TC_DP4	Länge 2	Typ und Ebene
\$TC_DP5	Länge 3	
Geometrie	Radius	

Werkzeugparameter Nummer (DP)	Bedeutung der Systemvariablen	Bemerkung
\$TC_DP6 ¹⁾ \$TC_DP6 ²⁾	Radius 1 / Länge 1 Durchmesser d	Fräs-/Dreh-/Schleifwerkz. Nutsäge
\$TC_DP7 ¹⁾ \$TC_DP7 ²⁾	Länge 2 / Eckenradius kegelige Fräser Nutbreite b Eckenradius	Fräswerkzeuge Nutsäge
\$TC_DP8 ¹⁾ \$TC_DP8 ²⁾	Verrundungsradius 1 für Fräswerkzeuge Überstand k	Fräswerkzeuge Nutsäge
\$TC_DP9 ^{1) 3)}	Verrundungsradius 2	reserviert
\$TC_DP10 ¹⁾	Winkel 1 Stirnseite des Werkzeugs	kegelige Fräswerkzeuge
\$TC_DP11 ¹⁾	Winkel 2 Werkzeug-Längsachse	kegelige Fräswerkzeuge
Verschleiß	Längen- und Radiuskorrektur	
\$TC_DP12	Länge 1	
\$TC_DP13	Länge 2	
\$TC_DP14	Länge 3	
\$TC_DP15 ¹⁾ \$TC_DP15 ²⁾	Radius 1 / Länge 1 Durchmesser d	Fräs-/Dreh-/Schleifwerkz. Nutsäge
\$TC_DP16 ¹⁾ \$TC_DP16 ³⁾	Länge 2 / Eckenradius kegelige Fräser Nutbreite b Eckenradius	Fräswerkzeuge Nutsäge
\$TC_DP17 ¹⁾ \$TC_DP17 ²⁾	Verrundungsradius 1 für Fräswerkzeuge Überstand k	Fräsen / 3D Stirnfräsen Nutsäge
\$TC_DP18 ^{1) 3)}	Verrundungsradius 2	reserviert
\$TC_DP19 ¹⁾	Winkel 1 Stirnseite des Werkzeugs	kegelige Fräswerkzeuge
\$TC_DP20 ¹⁾	Winkel 2 Werkzeug-Längsachse	kegelige Fräswerkzeuge
Basismaß/Adapter	Längenkorrekturen	
\$TC_DP21	Länge 1	
\$TC_DP22	Länge 2	
\$TC_DP23	Länge 3	
Technologie		
\$TC_DP24	Freiwinkel	nur für Drehwerkzeuge
\$TC_DP25		reserviert

¹⁾ Gilt auch bei Fräswerkzeugen für das 3D-Stirnfräsen

²⁾ Bei Nutsäge Werkzeugtyp

³⁾ reserviert (wird von SINUMERIK 840D sl nicht benutzt)

Anmerkungen

Für die geometrischen Größen (z. B. Länge 1 oder Radius) bestehen mehrere Eintragskomponenten. Diese werden zu einer resultierenden Größe additiv verrechnet (z. B. Gesamtlänge 1, Gesamtradius), die dann zur Wirkung kommt.

Nicht benötigte Korrekturen sind mit dem Wert Null zu belegen.

Werkzeugparameter \$TC-DP1 bis \$TC-DP23 mit Konturwerkzeugen

Hinweis

Die Werkzeugparameter, die in der Tabelle nicht aufgeführt sind wie z. B. \$TC_DP7, werden nicht ausgewertet, d. h. ihr Inhalt ist bedeutungslos.

Werkzeugparameter Nummer (DP)	Bedeutung	Schneiden Dn		Bemerkung
\$TC_DP1	Werkzeugtyp			400 bis 599
\$TC_DP2	Schneidenlage			
Geometrie	Längenkorrektur			
\$TC_DP3	Länge 1			
\$TC_DP4	Länge 2			
\$TC_DP5	Länge 3			
Geometrie	Radius			
\$TC_DP6	Radius			
Geometrie	Grenzwinkel			
\$TC_DP10	minimaler Grenzwinkel			
\$TC_DP11	maximaler Grenzwinkel			
Verschleiß	Längen- und Radiuskorrektur			
\$TC_DP12	Verschleiß Länge 1			
\$TC_DP13	Verschleiß Länge 2			
\$TC_DP14	Verschleiß Länge 3			
\$TC_DP15	Verschleiß Radius			
Verschleiß	Grenzwinkel			
\$TC_DP19	Verschleiß min. Grenzwinkel			
\$TC_DP20	Verschleiß max. Grenzwinkel			
Basismaß/Adapter	Längenkorrekturen			
\$TC_DP21	Länge 1			
\$TC_DP22	Länge 2			
\$TC_DP23	Länge 3			

Grundwert und Verschleißwert

Die resultierenden Größen ergeben sich jeweils als Summe aus Grundwert und Verschleißwert (z. B. \$TC_DP6 + \$TC_DP15 für den Radius). Zur Werkzeuglänge der ersten Schneide wird außerdem noch das Basismaß (\$TC_DP21 – \$TC_DP23) addiert. Zusätzlich wirken auf diese Werkzeuglänge alle anderen Größen, die auch bei einem herkömmlichen Werkzeug die effektive Werkzeuglänge beeinflussen können (Adapter, orientierbarer Werkzeugträger, Settingdaten).

Grenzwinkel 1 und 2

Die Grenzwinkel 1 bzw. 2 beziehen sich jeweils auf den Vektor vom Schneidenmittelpunkt zum Schneidenbezugspunkt und werden im Gegenuhrzeigersinn gezählt.

10.2 Additive Korrekturen

10.2.1 Additive Korrekturen anwählen (DL)

Additive Korrekturen können als in der Bearbeitung programmierbare Prozesskorrekturen betrachtet werden. Sie beziehen sich auf die geometrischen Daten einer Schneide und sind somit Bestandteil der Werkzeugschneidendaten.

Die Daten einer additiven Korrektur werden über eine DL-Nummer angesprochen (DL: Location dependent; Korrekturen bezüglich des jeweiligen Einsatzorts) und über die Bedienoberfläche eingegeben.

Anwendung

Durch additive Korrekturen können einsatzortbedingte Maßfehler ausgeglichen werden.

Syntax

DL=<Nummer>

Bedeutung

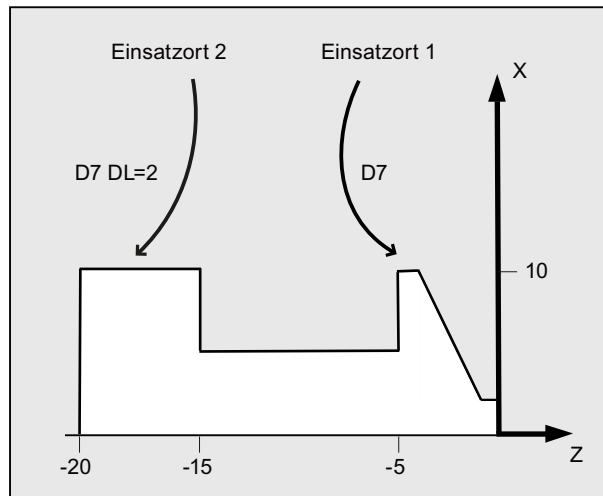
DL:	Befehl zur Aktivierung einer additiven Korrektur
<Nummer>:	Über den Parameter <Nummer> wird der zu aktivierende additive Werkzeugkorrekturdatensatz angegeben

Hinweis

Die Festlegung von Anzahl und Aktivierung der additiven Korrekturen erfolgt über Maschinendaten (→ Angaben des Maschinenherstellers beachten!).

Beispiel

Die gleiche Schneide wird für 2 Lagersitze verwendet:



Programmcode	Kommentar
N110 T7 D7	; Der Revolver wird auf Platz 7 positioniert. D7 und DL=1 werden aktiviert und im nächsten Satz herausgefahren.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; Additiv zu D7 wird DL=2 aktiviert und im nächsten Satz herausgefahren.
N150 G1 Z-21	
N160 G0 X200 Z200	; Werkzeugwechsellpunkt anfahren.
...	

10.2.2 Verschleiß- und Einrichtewerte festlegen (\$TC_SCPxy[t,d], \$TC_ECPxy[t,d])

Verschleiß- und Einrichtewerte können über Systemvariablen gelesen und geschrieben werden. Dabei orientiert sich die Logik an der Logik der entsprechenden Systemvariablen für Werkzeuge und Schneiden.

Systemvariablen

\$TC_SCPxy[<t>,<d>]:	Verschleißwerte, die über xy dem jeweiligen Geometrieparameter zugeordnet sind, wobei x die Nummer des Verschleißwerts entspricht und y den Bezug zum Geometrieparameter herstellt.
\$TC_ECPxy[<t>,<d>]:	Einrichtewerte, die über xy dem jeweiligen Geometrieparameter zugeordnet sind, wobei x die Nummer des Einrichtewerts entspricht und y den Bezug zum Geometrieparameter herstellt.
<t>: T-Nummer des Werkzeugs	
<d>: D-Nummer der Schneide des Werkzeugs	

Hinweis

Die festgelegten Verschleiß- und Einrichtewerte werden zu den Geometrieparametern und den übrigen Korrekturparametern (D-Nummer) addiert.

Beispiel

Der Verschleißwert der Länge 1 wird für die Schneide <d> des Werkzeugs <t> auf den Wert 1.0 festgelegt.

Parameter: \$TC_DP3 (Länge 1, bei Drehwerkzeugen)

Verschleißwerte: \$TC_SCP13 bis \$TC_SCP63

Einrichtewerte: \$TC_ECP13 bis \$TC_ECP63

\$TC_SCP43 [<t>,<d>] = 1.0

10.2.3 Additive Korrekturen löschen (DELDL)

Mit dem Befehl `DELDL` werden additive Korrekturen für die Schneide eines Werkzeugs gelöscht (Freigabe von Speicher). Dabei werden sowohl die festgelegten Verschleißwerte als auch die Einrichtewerte gelöscht.

Syntax

```
DELDL [<t>,<d>]
DELDL [<t>]
DELDL
<Status>=DELDL [<t>,<d>]
```

Bedeutung

DELDL:	Befehl zum Löschen additiver Korrekturen
<t>:	T-Nummer des Werkzeugs
<d>:	D-Nummer der Schneide des Werkzeugs
DELDL [<t>,<d>]:	Es werden alle additiven Korrekturen der Schneide <d> des Werkzeugs <t> gelöscht.
DELDL [<t>]:	Es werden alle additiven Korrekturen aller Schneiden des Werkzeugs <t> gelöscht.
DELDL:	Es werden alle additiven Korrekturen aller Schneiden aller Werkzeuge der TO-Einheit gelöscht (für den Kanal, in dem der Befehl programmiert wird).

<Status>:	Lösch-Status	
	Wert:	Bedeutung:
	0	Das Löschen wurde erfolgreich durchgeführt.
	-	Das Löschen wurde nicht durchgeführt (wenn die Parametrierung genau eine Schneide bezeichnet), oder das Löschen erfolgte nicht vollständig (wenn die Parametrierung mehrere Schneiden bezeichnet).

Hinweis

Verschleiß- und Einrichtewerte aktiver Werkzeuge können nicht gelöscht werden (verhält sich analog zum Löschverhalten von D bzw. Werkzeugdaten).

10.3 Werkzeugkorrektur - Sonderbehandlung

Mit den Settingdaten SD42900 bis SD42960 lässt sich die Bewertung der Vorzeichen für Werkzeuglänge und Verschleiß steuern.

Das gilt ebenfalls für das Verhalten der Verschleißkomponenten beim Spiegeln von Geometrieachsen oder beim Wechsel der Bearbeitungsebene und auch zur Temperaturkompensation in Werkzeugrichtung.

Verschleißwerte

Wenn im Folgenden auf Verschleißwerte Bezug genommen wird, ist darunter jeweils die Summe aus den eigentlichen Verschleißwerten (\$TC_DP12 bis \$TC_DP20) und den Summenkorrekturen mit den Verschleißwerten (\$SCPX3 bis \$SCPX11) und Einrichtewerten (\$ECPX3 bis \$ECPX11) zu verstehen.

Näheres zu den Summenkorrekturen siehe:

Literatur:

Funktionshandbuch Werkzeugverwaltung

Settingdaten

SD42900 \$SC_MIRROR_TOOL_LENGTH	Spiegeln von Werkzeuglängenkomponenten und Komponenten des Basismaßes.
SD42910 \$SC_MIRROR_TOOL_WEAR	Spiegeln von Verschleißwerten der Werkzeuglängenkomponenten.
SD42920 \$SC_WEAR_SIGN_CUTPOS	Vorzeichenbewertung der Verschleißkomponenten in Abhängigkeit von der Schneidenlage.
SD42930 \$SC_WEAR_SIGN	Invertiert die Vorzeichen der Verschleißmaße.
SD42935 \$SC_WEAR_TRANSFORM	Transformation der Verschleißwerte.
SD42940 \$SC_TOOL_LENGTH_CONST	Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen.

SD42950 \$SC_TOOL_LENGTH_TYPE	Zuordnung der Werkzeuglängenkomponenten unabhängig vom Werkzeugtyp.
SD42960 \$SC_TOOL_TEMP_COMP	Temperaturkompensationswert in Werkzeugrichtung. Ist auch bei vorhandener Werkzeugorientierung wirksam.

Literatur

Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1)

Weitere Informationen

Wirksamwerden der veränderten Settingdaten

Die Neubewertung von Werkzeugkomponenten bei einer Änderung der beschriebenen Settingdaten wird erst wirksam, wenn das nächste Mal eine Werkzeugschneide angewählt wird. Ist ein Werkzeug bereits aktiv und die Bewertung der Daten dieses Werkzeugs soll verändert wirksam werden, muss dieses Werkzeug erneut angewählt werden.

Entsprechendes gilt für den Fall, dass sich die resultierende Werkzeuglänge ändert, weil der Spiegelungszustand einer Achse geändert wurde. Das Werkzeug muss nach dem Spiegelbefehl erneut angewählt werden, damit die geänderten Werkzeuglängenkomponenten wirksam werden.

Orientierbare Werkzeugträger und neue Settingdaten

Die Settingdaten SD42900 bis SD42940 wirken nicht auf die Komponenten eines eventuell aktiven orientierbaren Werkzeugträgers. Ein Werkzeug geht jedoch immer mit seiner gesamten resultierenden Länge (Werkzeuglänge + Verschleiß + Basismaß) in die Berechnung mit einem orientierbaren Werkzeugträger ein. Bei der Berechnung der resultierenden Gesamtlänge werden alle Änderungen berücksichtigt, die durch die Settingdaten verursacht wurden; d.h. Vektoren des orientierbaren Werkzeugträgers sind unabhängig von der Bearbeitungsebene.

Hinweis

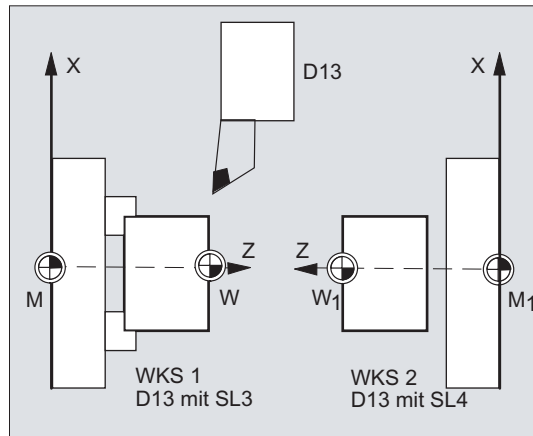
Häufig wird es beim Einsatz orientierbarer Werkzeugträger sinnvoll sein, alle Werkzeuge für ein nicht gespiegeltes Grundsystem zu definieren, auch diejenigen, die nur bei Spiegelbearbeitung verwendet werden. Bei Bearbeitung mit gespiegelten Achsen wird dann der Werkzeugträger so gedreht, dass die tatsächliche Lage des Werkzeugs richtig beschrieben wird. Alle Werkzeuglängenkomponenten wirken dann automatisch in der richtigen Richtung, so dass sich eine Steuerung der Bewertung einzelner Komponenten über Settingdaten abhängig vom Spiegelungszustand einzelner Achsen erübrigt.

Weitere Anwendungsmöglichkeiten

Die Verwendung der Funktionalität orientierbarer Werkzeugträger kann auch dann sinnvoll sein, wenn an der Maschine physikalisch keine Möglichkeit vorgesehen ist, Werkzeuge zu drehen, Werkzeuge aber mit verschiedenen Orientierungen fest installiert sind. Die Werkzeugvermessung kann dann einheitlich in einer Grundorientierung vorgenommen werden, und die für die Bearbeitung relevanten Maße ergeben sich durch Drehungen eines virtuellen Werkzeugträgers.

10.3.1 Werkzeuglängen spiegeln

Mit gesetzten Settingdaten SD42900 \$SC_MIRROR_TOOL_LENGTH und SD42910 \$SC_MIRROR_TOOL_WEAR ungleich Null können Sie Werkzeuglängenkomponenten und Komponenten der Basismaße mit Verschleißwerten deren zugehörigen Achsen spiegeln.



SD42900 \$SC_MIRROR_TOOL_LENGTH

Settingdatum **ungleich** Null:

Es werden die Werkzeuglängenkomponenten (\$TC_DP3, \$TC_DP4 und \$TC_DP5) und die Komponenten der Basismaße (\$TC_DP21, \$TC_DP22 und \$TC_DP23), deren zugehörige Achsen gespiegelt sind, ebenfalls gespiegelt - durch Vorzeicheninvertierung.

Die Verschleißwerte werden **nicht** mitgespiegelt. Sollen diese ebenfalls gespiegelt werden, muss das Settingdatum SD42910 \$SC_MIRROR_TOOL_WEAR gesetzt sein.

SD42910 \$SC_MIRROR_TOOL_WEAR

Settingdatum **ungleich** Null:

Es werden die Verschleißwerte der Werkzeuglängenkomponenten, deren zugehörige Achsen gespiegelt sind, ebenfalls gespiegelt - durch Vorzeicheninvertierung.

10.3.2 Vorzeichenbewertung Verschleiß

Mit gesetzten Settingdaten SD42920 \$SC_WEAR_SIGN_CUTPOS und SD42930 \$SC_WEAR_SIGN ungleich Null können Sie die Vorzeichenbewertung der Verschleißkomponenten invertieren.

SD42920 \$SC_WEAR_SIGN_CUTPOS

Settingdatum **ungleich** Null:

Bei Werkzeugen mit relevanter Schneidenlage (Dreh- und Schleifwerkzeuge, Werkzeugtypen 400) hängt die Vorzeichenbewertung der Verschleißkomponenten in der Bearbeitungsebene von der Schneidenlage ab. Bei Werkzeugtypen ohne relevanter Schneidenlage ist dieses Settingdatum bedeutungslos.

In folgender Tabelle sind die Maße durch ein X gekennzeichnet, deren Vorzeichen über das SD42920 (ungleich 0) invertiert wird:

Schneidenlage	Länge 1	Länge 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Hinweis

Die Vorzeichenbewertung durch SD42920 und SD42910 sind voneinander unabhängig. Wenn z. B. das Vorzeichen einer Maßangabe durch beide Settingdaten geändert wird, bleibt das resultierende Vorzeichen unverändert.

SD42930 \$SC_WEAR_SIGN

Settingdatum **ungleich** Null:

Das Vorzeichen aller Verschleißmaße wird invertiert. Es wirkt sowohl auf die Werkzeuglänge als auch auf die übrigen Größen wie Werkzeugradius, Verrundungsradius usw.

Wird ein positives Verschleißmaß eingegeben, wird somit das Werkzeug "kürzer" und "dünner", siehe Kapitel "Werkzeugkorrektur, Sonderbehandlung", Wirksamwerden der veränderten Settingdaten".

10.3.3 Koordinatensystem der aktiven Bearbeitung (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

Abhängig von der Kinematik der Maschine oder vom Vorhandensein eines orientierbaren Werkzeugträgers werden die in einem dieser Koordinatensysteme gemessenen Verschleißwerte in ein geeignetes Koordinatensystem überführt bzw. transformiert.

Koordinatensysteme der aktiven Bearbeitung

Aus den folgenden Koordinatensystemen können Offsets der Werkzeuglänge hervorgehen, welche die Werkzeuglängenkomponente Verschleiß über den entsprechenden G-Code der Gruppe 56 in ein aktives Werkzeug eingerechnet werden.

- Maschinenkoordinatensystem (MKS)
- Basiskoordinatensystem (BKS)
- Werkstückkoordinatensystem (WKS)

- Werkzeugkoordinatensystem (TCS)
- Werkzeugkoordinatensystem der kinematischen Transformation (KCS)

Syntax

TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS

Bedeutung

TOWSTD:	Grundstellungswert für Korrekturen in der Werkzeuglänge Verschleißwert
TOWMCS:	Korrekturen in der Werkzeuglänge im MKS
TOWWCS:	Korrekturen in der Werkzeuglänge im WKS
TOWBCS:	Korrekturen in der Werkzeuglänge im BKS
TOWTCS:	Korrekturen der Werkzeuglänge am Werkzeugträgerbezugspunkt (orientierbarer Werkzeugträger)
TOWKCS:	Korrekturen der Werkzeuglänge des Werkzeugkopfes (kinematischer Transformation)

Weitere Informationen

Unterscheidungsmerkmale

In der folgenden Tabelle sind die wichtigsten Unterscheidungsmerkmale dargestellt:

G-Code	Verschleißwert	Aktiver orientierbarer Werkzeugträger
TOWSTD:	Grundstellungswert, Werkzeuglänge	Verschleißwerte unterliegen der Drehung.
TOWMCS:	Verschleißwert Im MKS. TOWMCS ist mit TOWSTD identisch, wenn kein orientierbarer WZ-Träger aktiv ist.	Es dreht nur der Vektor der resultierenden Werkzeuglänge ohne Berücksichtigung des Verschleißes.
TOWWCS:	Der Verschleißwert wird Im WKS auf das MKS umgerechnet.	Der Werkzeugvektor wird ohne Berücksichtigung des Verschleißes wie bei TOWMCS berechnet.
TOWBCS:	Der Verschleißwert wird Im BKS auf das MKS umgerechnet.	Der Werkzeugvektor wird ohne Berücksichtigung des Verschleißes wie bei TOWMCS berechnet.
TOWTCS:	Der Verschleißwert wird Im Werkzeugkoordinatensystem auf das MKS umgerechnet.	Der Werkzeugvektor wird ohne Berücksichtigung des Verschleißes wie bei TOWMCS berechnet.

TOWWCS , TOWBCS, TOWTCS: Der Verschleißvektor wird zum Werkzeugvektor addiert.

Lineare Transformation

Die Werkzeuglänge ist im MKS nur sinnvoll definierbar, wenn das MKS aus dem BKS durch eine lineare Transformation hervorgeht.

Nicht lineare Transformation

Ist z. B. mit TRANSMIT eine nicht lineare Transformation aktiv, dann wird bei Angabe des MKS als gewünschtes Koordinatensystem automatisch das BKS verwendet.

Keine kinematische Transformation und kein orientierbarer Werkzeugträger

Ist weder eine kinematische Transformation noch ein orientierbarer Werkzeugträger aktiv, dann fallen bis auf das WKS alle weiteren vier Koordinatensysteme zusammen. Damit unterscheidet sich nur das WKS von den übrigen. Da ausschließlich Werkzeuglängen zu bewerten sind, haben Translationen zwischen den Koordinatensystemen keine Bedeutung.

Literatur:

Weitere Informationen zur Werkzeugkorrektur siehe:
Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1)

Einrechnung der Verschleißwerte

Das Settingdatum **SD42935 \$SC_WEAR_TRANSFORM** legt fest, welche der drei Verschleißkomponenten:

- Verschleiß
- Summenkorrekturen fein
- Summenkorrekturen grob

einer Drehung durch eine Adaptertransformation oder einen orientierbaren Werkzeugträger unterworfen werden soll, wenn einer der folgenden G-Codes aktiv ist:

- TOWSTD Grundstellung
für Korrekturen in der Werkzeuglänge
- TOWMCS Verschleißwerte
im Maschinenkoordinatensystem (MKS)
- TOWWCS Verschleißwerte
im Werkstückkoordinatensystem (WKS)
- TOWBCS Verschleißwerte (BKS)
im Basiskoordinatensystem

- TOWTCS Verschleißwerte im Werkzeugkoordinatensystem an der Werkzeughalteraufnahme (T Werkzeugträgerbezug)
- TOWKCS Verschleißwerte im Koordinatensystem des Werkzeugkopfes bei kinetischer Transformation

Hinweis

Die Bewertung der einzelnen Verschleißkomponenten (Zuordnung zu den Geometrieachsen, Vorzeichenbewertung) wird beeinflusst durch:

- die aktive Ebene
 - die Adaptertransformation
 - folgende Settingdaten:
 - SD42910 \$SC_MIRROW_TOOL_WEAR
 - SD42920 \$SC_WEAR_SIGN_CUTPOS
 - SD42930 \$SC_WEAR_SIGN
 - SD42940 \$SC_TOOL_LENGTH_CONST
 - SD42950 \$SC_TOOL_LENGTH_TYPE
-

10.3.4 Werkzeuglänge und Ebenenwechsel

Mit gesetzten Settingdaten SD42940 \$SC_TOOL_LENGTH_CONST ungleich Null können Sie Werkzeuglängenkomponenten wie Länge, Verschleiß und Basismaß zu den Geometrieachsen für Dreh- und Schleifwerkzeuge bei einem Ebenenwechsel zuordnen.

SD42940 \$SC_TOOL_LENGTH_CONST

Settingdatum **ungleich** Null:

Die Zuordnung der Werkzeuglängenkomponenten (Länge, Verschleiß und Basismaß) zu den Geometrieachsen beim Wechsel der Bearbeitungsebene (G17 - G19) wird nicht verändert.

Folgende Tabelle zeigt die Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen für Dreh- und Schleifwerkzeuge (WZ-Typ 400 bis 599):

Inhalt	Länge 1	Länge 2	Länge 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Jeder Wert ungleich 0, der nicht gleich einem der sechs aufgeführten Werte ist, wird wie der Wert 18 bewertet.

Folgende Tabelle zeigt die Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen für alle anderen Werkzeuge (WZ-Typ < 400 bzw. > 599):

Bearbeitungsebene	Länge 1	Länge 2	Länge 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Jeder Wert ungleich 0, der nicht gleich einem der sechs aufgeführten Werte ist, wird wie der Wert 17 bewertet.

Hinweis

Bei der Darstellung in den Tabellen wird davon ausgegangen, dass die Geometrieachsen bis 3 mit X, Y, Z bezeichnet werden. Für die Zuordnung einer Korrektur zu einer Achse ist nicht der Achsbezeichner, sondern die Achsreihenfolge maßgebend.

10.4 Online-Werkzeugkorrektur

10.4.1 Polynom-Funktion definieren (FCTDEF)

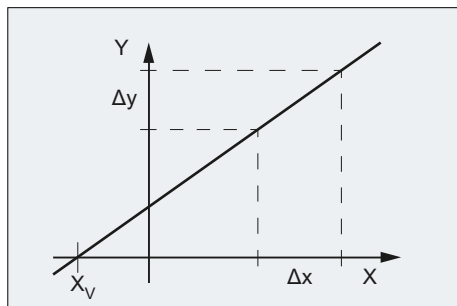
Bestimmte Abrichtstrategien (z. B. Abrichtrolle) zeichnen sich dadurch aus, dass die Schleifscheibe kontinuierlich (linear) mit der Zustellung der Abrichtrolle am Radius abnimmt. Hierfür benötigt man eine lineare Funktion zwischen der Zustellung der Abrichtrolle und dem Schreiben des Verschleißwerts der jeweiligen Länge. Die Definition der linearen Funktion erfolgt über die vordefinierte Prozedur FCTDEF(...) für Polynom-Funktionen bis maximal dritten Grades.

Geradengleichung

$$y = f(x) = a_0 + a_1 * x_1$$

a_1 : Steigung der Geraden, mit $a_1 = \Delta x / \Delta y$

a_0 : Verschiebung der Geraden entlang der X-Achse, mit $a_0 = -a_1 * X_v$



Syntax

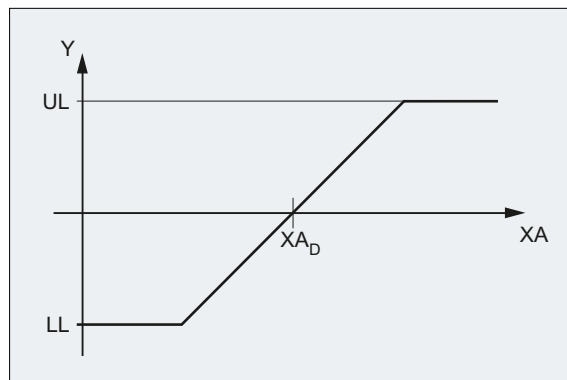
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)

Bedeutung

FCTDEF(...):	Definition einer Polynom-Funktion für PUTFTOCF(...): $y = f(x) = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3$	
<Func>:	Funktionsnummer	
	Datentyp:	INT
	Wertebereich:	1, 2, 3
<LLimit>:	Unterer Begrenzungswert	
	Datentyp:	REAL
<ULimit>:	Oberer Begrenzungswert	
	Datentyp:	REAL
<a0>,<a1>,<a2>,<a3>:	Koeffizienten der Polynom-Funktion	
	Datentyp:	REAL

Beispiel**Festlegungen**

- Funktionsnummer: 1
- Unterer und Oberer Begrenzungswert: -100, 100
- Steigung der Kennlinie: $a_1 = 1$
- Der Arbeitspunkt soll in der Mitte der Kennlinie liegen. Die Kennlinie muss dazu anhand der Sollposition der Achse XA im WKS zum Zeitpunkt der Funktionsdefinition im NC-Programm in negativer Y-Richtung verschoben werden: $a_0 = -a_1 * XA_D = -1 * \AA_IW
- $a_2 = a_3 = 0$

Kennlinie

UL Oberer Begrenzungswert

LL Unterer Begrenzungswert

XA_D Sollwert der Achse XA zum Zeitpunkt der Funktionsdefinition im NC-Programm

Programmierung

Programmcode	Kommentar
FCTDEF (1, -100, 100, -\$AA_IW[XA], 1)	; Funktionsdefinition

10.4.2 Online-Werkzeugkorrektur schreiben, kontinuierlich (PUTFTOCF)

Mit der vordefinierten Prozedur PUTFTOCF(...) wird eine Online-Werkzeugkorrektur über eine zuvor mit FCTDEF(...) (Seite 388) definierte Polynom-Funktion vorgenommen.

Hinweis

Die Online-Werkzeugkorrektur kann auch über eine Synchronaktion erfolgen.

Weitere Informationen siehe Funktionshandbuch Synchronaktionen.

Syntax

PUTFTOCF (<Func>, <RefVal>, <ToolPar>, <Chan>, <Sp>)

Bedeutung

PUTFTOCF (. . .):	Online-WZK schreiben, kontinuierlich satzweise anhand der mit FCTDEF(...) definierten Polynom-Funktion	
<Func>:	Funktionsnummer, festgelegt bei der Funktionsdefinition mit FCTDEF(...)	
	Datentyp:	INT
	Wertebereich:	1, 2, 3
<RefVal>:	Bezugswert, von dem die Korrektur abgeleitet werden soll (z. B. Sollwert einer Achse)	
	Datentyp:	VAR REAL
<ToolPar>:	Nummer des Verschleißparameters (Länge 1, 2 oder 3), in dem der Korrekturwert verrechnet werden soll	
	Datentyp:	INT
<Chan>:	Nummer des Kanals, in dem die Online-WZK wirksam werden soll	
	Hinweis: Nur erforderlich, wenn die Korrektur nicht im aktiven Kanal wirksam werden soll.	
	Datentyp:	INT
<Sp>:	Nummer der Spindel, für die die Online-WZK wirksam werden soll	
	Hinweis: Nur erforderlich, wenn statt dem aktiven, im Einsatz befindlichen Werkzeug eine nicht aktive Schleifscheibe korrigiert werden soll.	
	Datentyp:	INT

10.4.3 Online-Werkzeugkorrektur schreiben, diskret (PUTFTOC)

Funktion

Mit der vordefinierten Prozedur PUTFTOC(...) wird eine Online-Werkzeugkorrektur mit einem festen Korrekturwert vorgenommen.

Syntax

```
PUTFTOC (<CorrVal>, <ToolPar>, <Chan>, <Sp>)
```

Bedeutung

PUTFTOC (...):	Online-WZK schreiben	
<CorrVal>:	Korrekturwert, der zum Verschleißparameter addiert wird	
	Datentyp:	REAL
<ToolPar>:	Nummer des Verschleißparameters (Länge 1, 2 oder 3), in dem der Korrekturwert verrechnet werden soll	
	Datentyp:	INT
<Chan>:	Nummer des Kanals, in dem die Online-WZK wirksam werden soll Hinweis: Nur erforderlich, wenn die Korrektur nicht im aktiven Kanal wirksam werden soll.	
	Datentyp:	INT
<Sp>:	Nummer der Spindel, für die die Online-WZK wirksam werden soll Hinweis: Nur erforderlich, wenn statt dem aktiven, im Einsatz befindlichen Werkzeug eine nicht aktive Schleifscheibe korrigiert werden soll.	
	Datentyp:	INT

10.4.4 Online-Werkzeugkorrektur ein-/ausschalten (FTOCON/FTOCOF)

Mit den G-Befehlen FTOCON und FTOCOF wird die Online-Werkzeugkorrektur ein- bzw. ausgeschaltet.

Syntax

```
FTOCON
...
FTOCOF
```

Bedeutung

FTOCON:	Online-Werkzeugkorrektur einschalten Der Befehl muss in dem Kanal programmiert werden, in dem die Online-Werkzeugkorrektur eingeschaltet werden soll.
FTOCOF:	Online-Werkzeugkorrektur ausschalten Der Befehl muss in dem Kanal programmiert werden, in dem die Online-Werkzeugkorrektur ausgeschaltet werden soll. Hinweis: Mit FTOCOF wird die Werkzeugkorrektur nicht weiter herausgefahren. In den schneidenspezifischen Korrekturdaten bleibt aber der mit PUTFTOC/PUTFTOCF eingerechnete Wert erhalten. Zum endgültigen Deaktivieren der Online-WZK muss nach FTOCOF noch eine An-/Abwahl des Werkzeugs (T...) erfolgen.

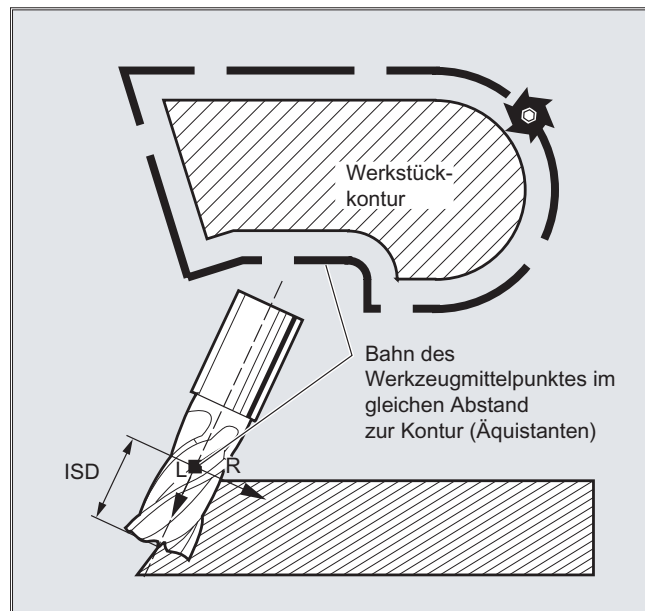
10.5 Aktivierung 3D-Werkzeugkorrekturen (CUT3DC..., CUT3DF...)**10.5.1 Aktivierung von 3D-Werkzeugkorrekturen (CUT3DC, CUT3DCD, CUT3DF, CUT3DFS, CUT3DFF, ISD)**

Bei der 3D-Werkzeugradiuskorrektur für zylindrische Werkzeuge wird im Gegensatz zur 2½ D-Werkzeugradiuskorrektur die veränderliche Werkzeugorientierung berücksichtigt.

Das Anfahrverhalten ist immer *NORM*.

Mit *G41/G42* wird die Werkzeugradiuskorrektur in Bewegungsrichtung links/rechts von der programmierten Bahn angewählt.

Die 3D-Werkzeugradiuskorrektur wirkt nur bei angewählter 5-Achs-Transformation. Daher wird die 3D-Werkzeugradiuskorrektur auch als 5D-Korrektur bezeichnet, da in diesem Fall 5 Freiheitsgrade für die Lage des Werkzeugs im Raum zur Verfügung stehen.



Unterschied zwischen 2½ D- und 3D-Werkzeugradiuskorrektur

Bei der 3D-Werkzeugradiuskorrektur ist die Werkzeug-Orientierung veränderlich.

Bei der 2½ D-Werkzeugradiuskorrektur wird mit einem Werkzeug mit konstanter Orientierung gerechnet.

Werkzeugradiuskorrektur bezogen auf ein Differenzwerkzeug

Die auf ein Differenzwerkzeug bezogene 3D-Werkzeugradiuskorrektur wird durch den Befehl CUT3DCD aktiviert. Sie ist anzuwenden, wenn sich die programmierte Kontur auf die Mittelpunktsbahn eines Differenzwerkzeugs bezieht und die Bearbeitung mit einem davon abweichenden Werkzeug erfolgt. Bei der Berechnung der 3D-Werkzeugradiuskorrektur werden dann nur der Verschleißwert des Radius des aktiven Werkzeugs (\$TC_DP_15) und die gegebenenfalls programmierten Werkzeugkorrekturoffsets OFFN und TOFFR eingerechnet. Der Grundradius (\$TC_DP6) des aktiven Werkzeugs wird **nicht** eingerechnet.

Syntax

```
CUT3DC
CUT3DCD
CUT3DFS
CUT3DFF
CUT3DF
ISD=<Wert>
```

Bedeutung

CUT3DC:	Aktivierung der 3D-Radiuskorrektur für das Umfangsfräsen
CUT3DCD:	Aktivierung der auf ein Differenzwerkzeug bezogenen 3D-Radiuskorrektur für das Umfangsfräsen
CUT3DFS:	D-Werkzeugkorrektur für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist durch G17 - G19 festgelegt und wird durch Frames nicht beeinflusst.
CUT3DFF:	D-Werkzeugkorrektur für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist die durch G17 - G19 festgelegte und gegebenenfalls durch einen Frame gedrehte Richtung.
CUT3DF:	D-Werkzeugkorrektur für das Stirnfräsen mit Orientierungsänderung (nur bei aktiver 5-Achs-Transformation).
G40 X... Y... Z... :	Zum Ausschalten: Linearsatz G0 / G1 mit Geometrieachsen
ISD:	Eintauchtiefe

Hinweis

Die Befehle sind modal wirksam und stehen in der gleichen Gruppe wie CUT2D und CUT2DF. Die Abwahl findet erst mit der nächsten Bewegung in der aktuellen Ebene statt. Dies gilt immer für G40 und ist unabhängig vom CUT-Befehl.

Zwischensätze bei aktiver 3D-Werkzeugradiuskorrektur sind erlaubt. Es gelten die Festlegungen der 2 1/2D-Werkzeugradiuskorrektur.

Randbedingungen**G450 / G451 und DISC**

An Außenecken wird immer ein Kreissatz eingefügt. G450/G451 haben keine Bedeutung. Der Befehl DISC wird nicht ausgewertet.

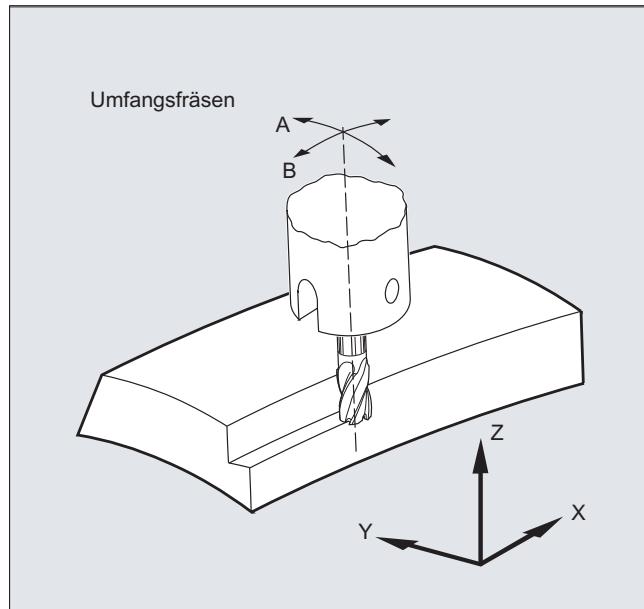
Beispiel

Programmcode	Kommentar
N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	; Werkzeug 1, Schneide 1
N30 TRAORI(1)	; Orientierungstrafo
N40 CUT3DC	; 3D-Werkzeugradiuskorrektur
N50 G42 X10 Y10	; Werkzeugradiuskorrektur rechts von der Kontur
N60 X60	
N70 ...	

10.5.2 3D-Werkzeugkorrektur: Umfangfräsen, Stirnfräsen

Umfangsfräsen

Die hier benutzte Variante des Umfangsfräsens ist durch die Vorgabe einer Bahn (Leitlinie) und der zugehörigen Orientierung realisiert. Bei dieser Art der Bearbeitung ist auf der Bahn die Werkzeugform ohne Bedeutung. Entscheidend ist allein der Radius am Werkzeugeingriffspunkt.

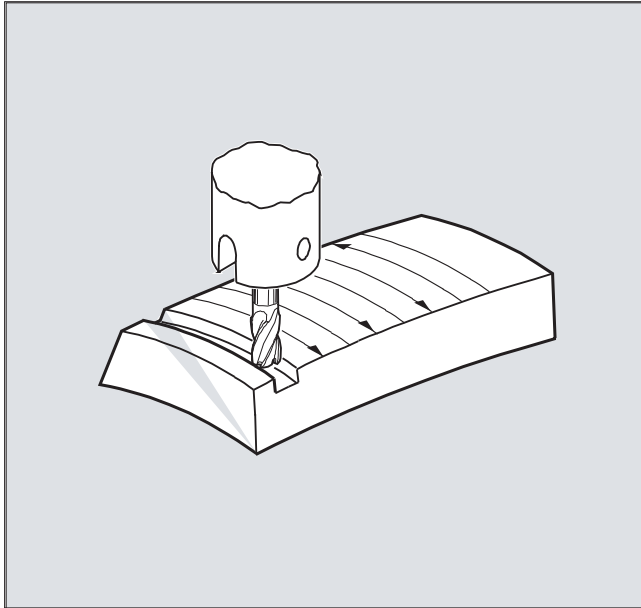


Hinweis

Die Funktion 3D-WRK beschränkt sich auf zylindrische Werkzeuge.

Stirnfräsen

Für diese Art des 3D-FräSENS benötigen Sie die zeilenweise Beschreibung der 3D-Bahnen auf der Werkstückoberfläche. Die Berechnungen werden unter Berücksichtigung der Werkzeugform und Werkzeugabmessungen - üblicherweise im CAM durchgeführt. Der Postprozessor schreibt in das Teileprogramm - neben den NC-Sätzen - die Werkzeugorientierungen (bei aktiver 5-Achstransformation) und den G-Code für die gewünschte 3D-Werkzeugkorrektur. Hierdurch hat der Maschinenbediener die Möglichkeit - abweichend von dem für die Berechnung der NC-Bahnen verwendeten Werkzeug -, geringfügig kleinere Werkzeuge einzusetzen.



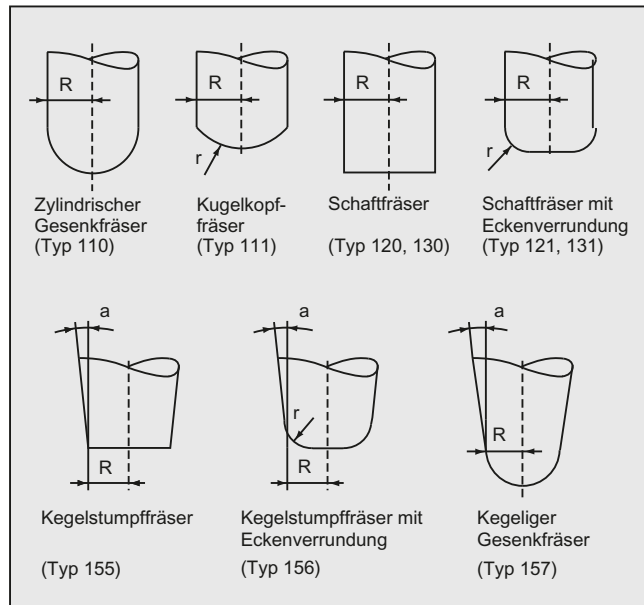
Beispiel:

NC-Sätze wurden mit Fräser 10 mm berechnet. Hier könnte auch mit Fräserdurchmesser 9,9 mm gefertigt werden, wobei dann mit verändertem Rauheitsprofil zu rechnen ist.

10.5.3 3D-Werkzeugkorrektur: Werkzeugformen und Werkzeugdaten für Stirnfräsen

Fräserformen, Werkzeugdaten

Im Folgenden sind die für Stirnfräsen möglichen Werkzeugformen und Grenzwerte der Werkzeugdaten zusammengestellt. Die Form des Werkzeugschafts wird nicht berücksichtigt. Die Werkzeugtypen 120 und 156 sind in ihrer Wirkung identisch.



Wird im NC-Programm eine andere als in der Abbildung gezeigte Typ-Nummer angegeben, verwendet das System automatisch den Werkzeugtyp 110 (Zylindrischer Gesenkfräser). Bei Verletzung der Grenzwerte für die Werkzeugdaten wird ein Alarm ausgegeben.

Fräsertyp	Typ-Nr.	R	r	a
Zylindrischer Gesenkfräser	110	> 0	-	-
Kugelkopffräser	111	> 0	> R	-
Schafffräser, Winkelkopffräser	120, 130	> 0	-	-
Schafffräser, Winkelkopffräser mit Eckenverrundung	121, 131	> r	> 0	-
Kugelstumpffräser	155	> 0	-	> 0
Kugelstumpffräser mit Eckenverrundung	156	> 0	> 0	> 0
Kegeliges Gesenkfräser	157	> 0	-	> 0

R = Schaftradius (Werkzeugradius)

r = Eckenradius

a = Winkel zwischen Werkzeuglängsachse und oberem Ende der Torusfläche

- = wird nicht ausgewertet

Werkzeugdaten	Werkzeugparameter	
Werkzeugmaße	Geometrie	Verschleiß
R	\$TC_DP6	\$TC_DP15
r	\$TC_DP7	\$TC_DP16
a	\$TC_DP11	\$TC_DP20

Werkzeuglängenkorrektur

Als Bezugspunkt für die Längenkorrektur gilt die Werkzeugspitze (Schnittpunkt Längsachse/Oberfläche).

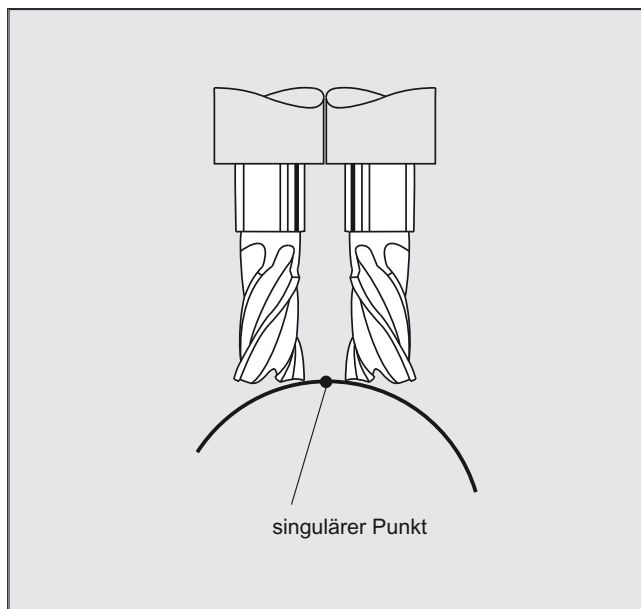
3D-Werkzeugkorrektur, Werkzeugwechsel

Ein neues Werkzeug mit veränderten Abmessungen (R, r, a) oder anderer Form darf nur mit Programmierung von G41 bzw. G42 angegeben werden (Übergang G40 nach G41 bzw. G42, erneute Programmierung von G41 bzw. G42). Alle anderen Werkzeugdaten, z. B. Werkzeuglängen, bleiben von dieser Regel unberücksichtigt, so dass solche Werkzeuge auch ohne erneutes G41 bzw. G42 eingewechselt werden können.

10.5.4 3D-Werkzeugkorrektur: Korrektur auf der Bahn, Bahnkrümmung, Eintauchtiefe (CUT3DC, CUT3DCD, ISD)

Korrektur auf der Bahn

Beim Stirnfräsen muss der Fall betrachtet werden, dass der Berührungspunkt auf der Werkzeugoberfläche springt. Wie in diesem Beispiel bei der Bearbeitung einer konvexen Fläche mit senkrecht stehendem Werkzeug. Die im Bild gezeigte Anwendung kann als Grenzfall betrachtet werden.



Dieser Grenzfall wird von der Steuerung überwacht, indem auf Basis der Winkelanstellungen zwischen Werkzeug und Flächennormalenvektoren sprunghafte Änderungen des Bearbeitungspunktes erkannt werden. An diesen Stellen fügt die Steuerung Linearsätze ein, so dass die Bewegung ausgeführt werden kann.

Für die Berechnung der Linearsätze sind in Maschinendaten für den Seitwärtswinkel zulässige Winkelbereiche hinterlegt. Falls die in Maschinendaten festgelegten Grenzwerte für zugelassene Winkelbereiche überschritten werden, wird ein Alarm angezeigt.

- MD21082 \$MC_CUTCOM_PLANE_ORI_LIMIT (Minimaler Winkel zw. Flächennormnormalenv. und WZ-Orientierung)
- MD21092 \$MC_MAX_TILT_ANGLE (Maximalbetrag des zul. Seitwärtswinkels bei Orientierungsprogr.)

Bahnkrümmung

Die Bahnkrümmung wird nicht überwacht. Auch hier empfiehlt es sich, nur solche Werkzeuge zu verwenden, mit denen ohne Konturverletzung gearbeitet werden kann.

Eintauchtiefe (ISD)

Die Eintauchtiefe ISD wird nur bei aktiver 3D-Werkzeugradiuskorrektur ausgewertet.

Mit dem Programmbefehl `ISD` (Insertion Depth) wird die Eintauchtiefe des Werkzeugs beim Umfangfräsen programmiert. Damit ist es möglich, die Lage des Bearbeitungspunktes auf der Mantelfläche des Werkzeugs zu verändern.

Werkzeugradiuskorrektur bezogen auf ein Differenzwerkzeug

Die auf ein Differenzwerkzeug bezogene 3D-Werkzeugradiuskorrektur wird durch den Befehl `CUT3DCD` aktiviert. Sie ist anzuwenden, wenn sich die programmierte Kontur auf die Mittelpunktsbahn eines Differenzwerkzeugs bezieht und die Bearbeitung mit einem davon abweichenden Werkzeug erfolgt. Bei der Berechnung der 3D-Werkzeugradiuskorrektur werden dann nur der Verschleißwert des Radius des aktiven Werkzeugs (`$TC_DP_15`) und die gegebenenfalls programmierten Werkzeugkorrekturoffsets `OFFN` und `TOFFR` eingerechnet. Der Grundradius (`$TC_DP6`) des aktiven Werkzeugs wird **nicht** eingerechnet.

Syntax

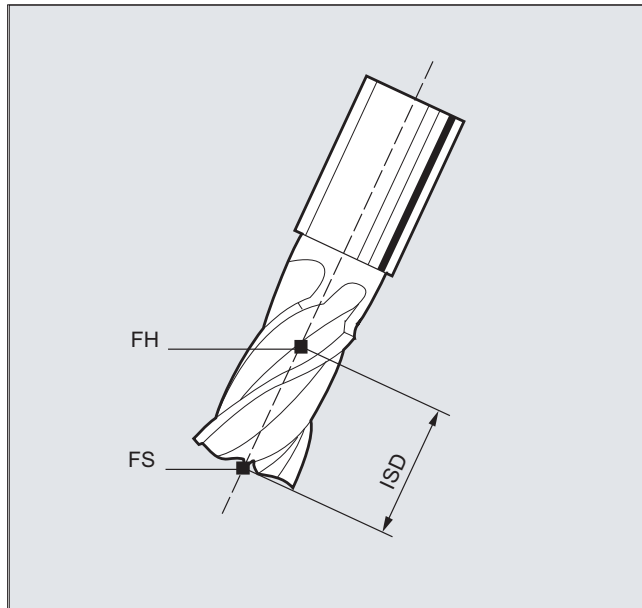
```
CUT3DC
CUT3DCD
ISD=<Wert>
```

Bedeutung

CUT3DC:	Aktivierung der 3D-Radiuskorrektur für das Umfangfräsen
CUT3DCD:	Aktivierung der auf ein Differenzwerkzeug bezogenen 3D-Radiuskorrektur für das Umfangfräsen
ISD:	Mit dem Befehl <code>ISD</code> wird der Abstand (<Wert>) zwischen Fräaserspitze (FS) und dem Fräserhilfspunkt (FH) angegeben.

Fräserhilfspunkt

Der Fräserhilfspunkt (FH) entsteht durch Projektion des programmierten Bearbeitungspunkts auf die Werkzeugachse.



Weitere Informationen

Taschenfräsen mit schrägen Seitenwänden für Umfangsfräsen mit CUT3DC

Bei dieser 3D-Werkzeugradiuskorrektur wird eine Abweichung des Fräserradius kompensiert, indem in Richtung der Flächennormalen der zu bearbeitenden Fläche zugestellt wird. Dabei bleibt die Ebene, in der die Stirnseite des Fräasers liegt unverändert, wenn die Eintauchtiefe ISD gleich geblieben ist. Ein Fräser mit z. B. kleinerem Radius gegenüber einem Normwerkzeug würde dann den Taschenboden, der auch die Begrenzungsfläche darstellt, nicht erreicht werden. Für eine automatische Zustellung des Werkzeugs muss der Steuerung diese Begrenzungsfläche bekannt sein, siehe Kapitel "3D-Umfräsen mit Begrenzungsflächen".

Weitere Informationen zur Kollisionsüberwachung siehe:

Literatur:

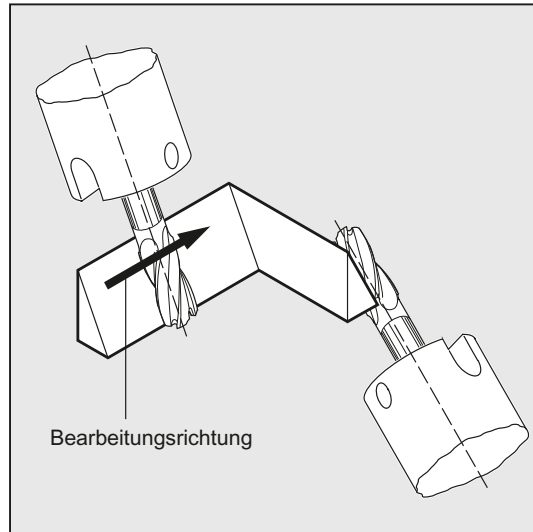
Programmierhandbuch Grundlagen; Kapitel "Werkzeugkorrekturen".

10.5.5 3D-Werkzeugkorrektur: Innenecken/Außenecken und Schnittpunktverfahren (G450/G451)

Innenecken/Außenecken

Außen- und Innenecken werden getrennt behandelt. Die Bezeichnung Innen- oder Außenecke ist abhängig von der Werkzeugorientierung.

Bei Orientierungsänderungen an einer Ecke kann der Fall auftreten, dass sich der Eckentyp während der Bearbeitung ändert. Tritt dieser Fall auf, wird die Bearbeitung mit einer Fehlermeldung abgebrochen.



Syntax

G450
G451

Bedeutung

G450:	Übergangskreis (Werkzeug umfährt Werkstückecken auf einer Kreisbahn)
G451:	Schnittpunkt der Äquidistanten (Werkzeug schneidet in der Werkstückecke frei)

Weitere Informationen

Schnittpunktverfahren für 3D-Korrektur

Bei 3D-Umfangsfräsen wird jetzt an Außenecken der G-Code G450/G451 ausgewertet, d. h. es kann der Schnittpunkt der Offset-Kurven angefahren werden. Bis SW 4 wurde an Außenecken immer ein Kreis eingefügt. Das verfügbare Schnittpunktverfahren ist bei typischen CAD-erzeugten 3D-Programmen besonders vorteilhaft. Diese bestehen häufig aus kurzen Geradensätzen (zur Approximation glatter Kurven), bei denen die Übergänge zwischen benachbarten Sätzen nahezu tangential sind.

Bei Werkzeugradiuskorrektur an der Außenseite der Kontur wurden bislang grundsätzlich Kreise zum Umfahren der Außenecken eingefügt. Da diese Sätze bei nahezu tangentialen Übergängen sehr kurz werden, ergeben sich unerwünschte Geschwindigkeitseinbrüche.

In diesen Fällen werden analog zur 2 ½ D-Radiuskorrektur die beiden beteiligten Kurven verlängert, der Schnittpunkt der beiden verlängerten Kurven wird angefahren.

Der Schnittpunkt wird bestimmt, indem die Offsetkurven der beiden beteiligten Sätze verlängert werden und deren Schnittpunkt in der Ebene senkrecht zur Werkzeugorientierung

an der Ecke bestimmt wird. Existiert kein derartiger Schnittpunkt, wird die Ecke wie bisher behandelt, d. h. es wird ein Kreis eingefügt.

Weitere Informationen zum Schnittpunktverfahren siehe:

Literatur:

Funktionshandbuch Sonderfunktionen; 3D-Werkzeugradiuskorrektur (W5)

10.5.6 3D-Werkzeugkorrektur: 3D-Umfangsfräsen mit Begrenzungsflächen

Anpassungen von 3D-Umfangsfräsen an Gegebenheiten von CAD-Programmen

Von CAD-Systemen generierte NC-Programme approximieren in der Regel die Mittelpunktsbahn eines Normwerkzeuges mit einer großen Anzahl kurzer Linearsätze. Damit diese so erzeugten Sätze vieler Teilekonturen die ursprüngliche Originalkontur möglichst genau nachbilden, ist es notwendig im Teileprogramm gewisse Anpassungen vorzunehmen.

Wichtige Informationen, die für eine optimale Korrektur erforderlich wären, aber im Teilprogramm nicht mehr zur Verfügung stehen, müssen durch geeignete Maßnahmen ersetzt werden. Nachfolgend werden typische Methoden dargestellt, um kritische Übergänge entweder direkt im Teileprogramm oder bei Ermittlung der realen Kontur (z. B. durch Zustellung des Werkzeugs) auszugleichen.

Anwendungen

Zusätzlich zu den typischen Anwendungsfällen, bei denen anstelle des Normwerkzeugs ein reales Werkzeug die Mittelpunktsbahn beschreibt, werden auch zylindrische Werkzeuge mit 3D-Werkzeugkorrektur behandelt. Hierbei bezieht sich die programmierte Bahn auf die Kontur an der Bearbeitungsfläche. Die hierfür zutreffende Begrenzungsfläche ist werkzeugunabhängig. Es wird wie bei der herkömmlichen Werkzeugradiuskorrektur der Gesamtradius zur Berechnung des senkrechten Offsets zur Begrenzungsfläche herangezogen.

10.5.7 3D-Werkzeugkorrektur: Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD)

3D-Umfangsfräsen mit realen Werkzeugen

Beim 3D-Umfangsfräsen mit kontinuierlicher oder konstanter Veränderung der Werkzeugorientierung wird häufig die Werkzeugmittelpunktsbahn für ein definiertes Normwerkzeug programmiert. Da in der Praxis oft nicht die passenden Normwerkzeuge zur Verfügung stehen, kann ein von einem Normwerkzeug nicht allzu stark abweichendes Werkzeug eingesetzt werden.

Mit CUT3DCCD wird für ein reales Differenzwerkzeug eine Begrenzungsfläche berücksichtigt, die das programmierte Normwerkzeug beschreiben würde. Das NC-Programm beschreibt die Mittelpunktsbahn des Normwerkzeuges.

Mit CUT3DCC wird bei Verwendung von zylindrischen Werkzeugen eine Begrenzungsfläche berücksichtigt, die das programmierte Normwerkzeug erreicht hätte. Das NC-Programm beschreibt die Kontur auf der Bearbeitungsfläche.

Syntax

```
CUT3DCCD
CUT3DCC
```

Bedeutung

CUT3DCCD:	Aktivierung der 3D-Werkzeugkorrektur für das Umfangsfräsen mit Begrenzungsflächen mit Differenzwerkzeug auf der Werkzeugsmittelpunktbahn: Zustellung zur Begrenzungsfläche.
CUT3DCC:	Aktivierung der 3D-Werkzeugkorrektur für das Umfangsfräsen mit Begrenzungsflächen mit 3D-Radiuskorrektur: Kontur an der Bearbeitungsfläche

Hinweis

Werkzeugradiuskorrektur mit G41, G42

Für die Werkzeugradiuskorrektur mit G41, G42 bei aktivem CUT3DCCD oder CUT3DCC muss die Option "Orientierungstransformation" vorhanden sein.

Normwerkzeuge mit Eckenverrundung

Die Eckenverrundung des Normwerkzeugs wird durch den Werkzeugparameter \$TC_DP7 beschrieben. Aus den Werkzeugparameter \$TC_DP16 ergibt sich die Abweichung der Eckenverrundung des realen Werkzeugs gegenüber dem Normwerkzeug.

Beispiel

Werkzeugabmessungen für einen Torusfräser mit verringertem Radius gegenüber dem Normwerkzeug.

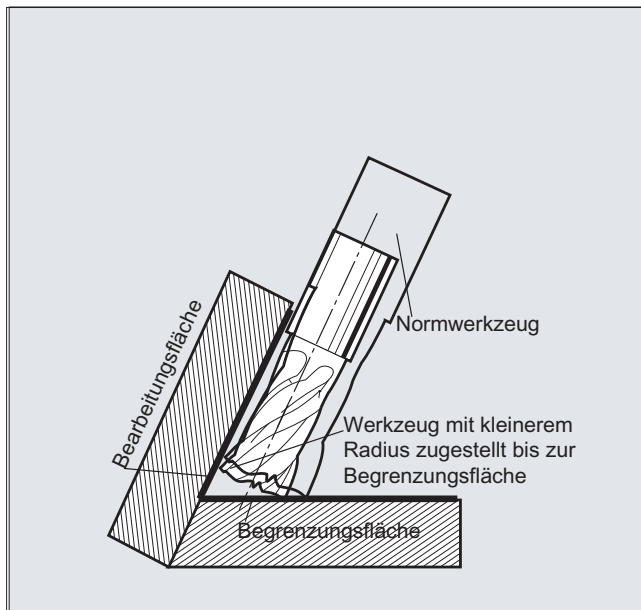
Werkzeugtyp	R = Schaftradius	r = Eckenradius
Normwerkzeug mit Eckenverrundung	$R = \$TC_DP6$	$r = \$TC_DP7$
Reales Werkzeug mit Eckenverrundung: Werkzeugtypen 121 und 131 Torusfräser (Schafffräser)	$R' = \$TC_DP6 + \$TC_DP15 + OFFN$	$r' = \$TC_DP7 + \TC_DP16
In diesem Beispiel sind sowohl \$TC_DP15 + OFFN als auch \$TC_DP16 negativ. Der Werkzeugtyp (\$TC_DP1) wird ausgewertet.		

Werkzeugtyp	R = Schafradius	r = Eckenradius
Zugelassen sind nur Fräserarten mit zylindrischen Schaft (Zylinder- oder Schaffräser) sowie Torusfräser (Typ 121 und 131) und im Grenzfall der zylindrische Gesenkfräser (Typ 110).	Bei diesen zugelassenen Fräserarten ist der Eckenradius r gleich dem Schafradius R. Alle anderen zugelassenen Werkzeugtypen werden als Zylinderfräser interpretiert und ein eventuell angegebenes Maß für die Eckenverrundung wird nicht ausgewertet.	
Zugelassen sind alle Werkzeugtypen der Nummern 1 – 399 mit Ausnahme der Nummern 111 und 155 bis 157.		

Weitere Informationen

Werkzeugmittelpunktsbahn mit Zustellung bis zur Begrenzungsfläche CUT3DCCD

Wird ein Werkzeug verwendet, welches im Vergleich zum passenden Normwerkzeug einen kleineren Radius aufweist, dann wird ein in Längsrichtung zugestellter Fräser soweit weiter geführt, bis dieser den Taschenboden wieder berührt. Damit wird die Ecke, die von der Bearbeitungs- und der Begrenzungsfläche gebildet wird so weit ausgeräumt, wie dies das Werkzeug zulässt. Es handelt sich dabei um eine gemischte Bearbeitungsweise aus Umfangs- und Stirnfräsen. Analog zu einem Werkzeug mit verringertem Radius, wird beim Werkzeug mit vergrößertem Radius, in die entgegengesetzte Richtung entsprechend zugestellt.



Gegenüber allen anderen Werkzeugkorrekturen der G-Code Gruppe 22 hat ein für CUT3DCCD angegebener Werkzeugparameter $\$TC_DP6$ keine Bedeutung für den Werkzeugradius und beeinflusst die resultierende Korrektur nicht.

Der Korrekturoffset ergibt sich aus der Summe von:

- Verschleißwert des Werkzeugradius (Werkzeugparameter $\$TC_DP15$)
- und einem zur Berechnung des senkrechten Offsets zur Begrenzungsfläche programmierten Werkzeugoffset $OFFN$.

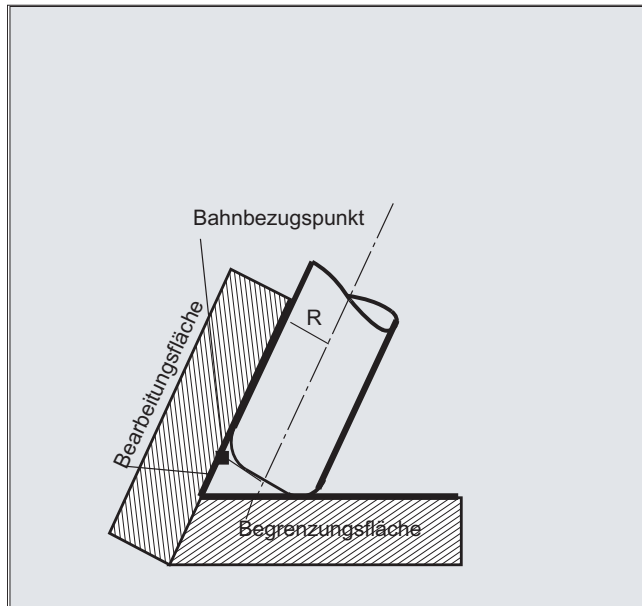
Ob die zu bearbeitende Fläche links oder rechts von der Bahn liegt, kann aus dem erzeugten Teileprogramm nicht entnommen werden. Es wird deshalb von einem positiven Radius und einem negativen Verschleißwert des Originalwerkzeuges ausgegangen. Ein negativer Verschleißwert beschreibt immer ein Werkzeug mit verringertem Durchmesser.

Verwendung von zylindrischen Werkzeugen

Bei der Verwendung von zylindrischen Werkzeugen ist eine Zustellung nur dann erforderlich, wenn die Bearbeitungsfläche und die Begrenzungsfläche einen spitzen Winkel (kleiner als 90 Grad) bilden. Werden Torusfräser (Zylinder mit Eckverrundung) verwendet, dann erfordert dies sowohl bei spitzen als auch bei stumpfen Winkeln eine Zustellung in Längsrichtung des Werkzeugs.

3D-Radiuskorrektur mit CUT3DCC, Kontur an der Bearbeitungsfläche

Ist `CUT3DCC` mit einem Torusfräser aktiv, so bezieht sich die programmierte Bahn auf einen fiktiven Zylinderfräser gleichen Durchmessers. Der hieraus resultierende Bahnbezugspunkt ist bei Verwendung eines Torusfräser im folgenden Bild dargestellt.



Es ist zulässig, dass der Winkel zwischen Bearbeitungs- und Begrenzungsfläche auch innerhalb eines Satzes von einem spitzen in einem stumpfen Winkel oder umgekehrt übergeht.

Gegenüber dem Normwerkzeug darf das verwendete reale Werkzeug sowohl größer als auch kleiner sein. Dabei darf der resultierende Eckenradius nicht negativ werden und das Vorzeichen des resultierenden Werkzeugradius muss erhalten bleiben.

Bei `CUT3DCC` bezieht sich das NC-Teileprogramm auf die Kontur an der Bearbeitungsfläche. Es wird hierbei wie bei der herkömmlichen Werkzeugradienkorrektur der Gesamtradius herangezogen, der sich zusammensetzt aus der Summe von:

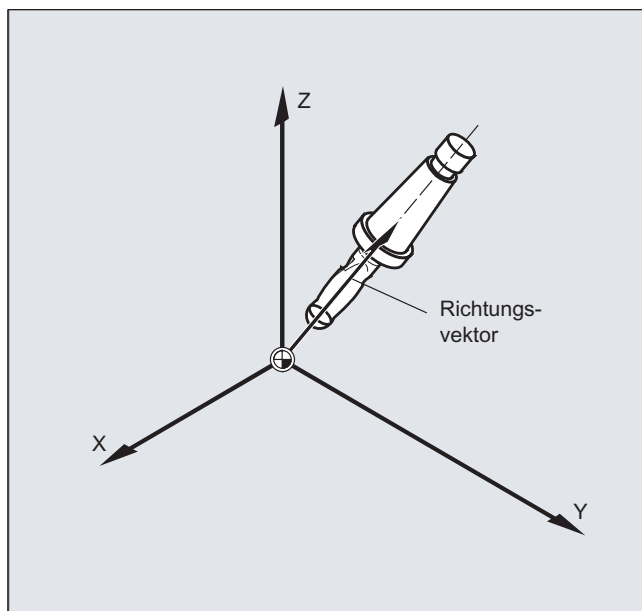
- Werkzeugradius (Werkzeugparameter `$TC_DP6`)
- Verschleißwert (Werkzeugparameter `$TC_DP15`)
- und einem zur Berechnung des senkrechten Offsets zur Begrenzungsfläche programmierten Werkzeugoffset `OFFN`.

Die Lage der Begrenzungsfläche wird bestimmt aus der Differenz der beiden Werte:

- Abmessungen des Normwerkzeugs
- Werkzeugradius (WZ-Parameter $\$TC_DP6$)

10.6 Werkzeugorientierung (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Unter Werkzeugorientierung versteht man die geometrische Ausrichtung des Werkzeugs im Raum. Bei einer 5-Achs-Bearbeitungsmaschine ist die Werkzeugorientierung über Programmbefehle einstellbar.



Mit `OSD` und `OST` aktivierte Überschiefbewegungen der Orientierung werden je nach Interpolationsart für die Werkzeugorientierung unterschiedlich gebildet.

Bei aktiver Vektorinterpolation wird der geglättete Orientierungsverlauf auch mittels Vektorinterpolation interpoliert. Dagegen wird bei aktiver Rundachsinterpolation die Orientierung direkt mittels Rundachsbewegungen geglättet.

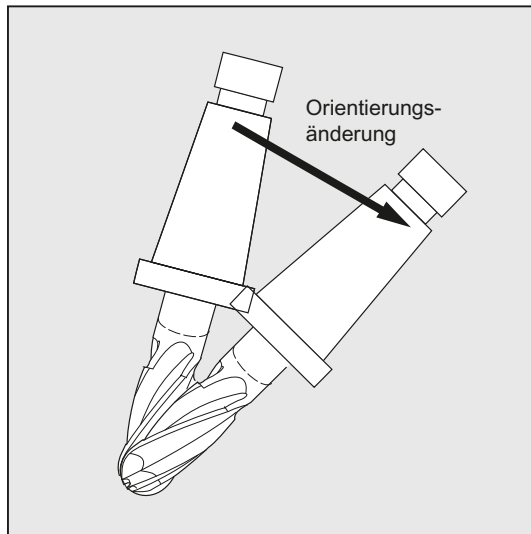
Programmierung

Programmierung der Orientierungsänderung:

Eine Orientierungsänderung des Werkzeugs kann programmiert werden durch:

- direkte Programmierung der Rundachsen A, B, C (Rundachsinterpolation)
- Euler- oder RPY-Winkel
- Richtungsvektor (Vektorinterpolation durch Angabe von `A3` oder `B3` oder `C3`)
- `LEAD/TILT` (Stirnfräsen)

Das Bezugskordinatensystem ist entweder das Maschinenkoordinatensystem (ORIMKS) oder das aktuelle Werkstückkoordinatensystem (ORIWKS).



Programmierung der Werkzeugorientierung:

ORIC:	Orientierung und Bahnbewegung parallel
ORID:	Orientierung und Bahnbewegung nacheinander
OSOF:	keine Orientierungsglättung
OSC:	Orientierung konstant
OSS:	Orientierungsglättung nur am Satzanfang
OSSE:	Orientierungsglättung am Satzanfang und -ende
ORIS:	Geschwindigkeit der Orientierungsänderung bei eingeschalteter Orientierungsglättung in Grad pro mm (gilt für OSS und OSSE)
OSD:	Überschleifen der Orientierung durch Vorgabe der Überschleiflänge mit dem Settingdatum: SD42674 \$SC_ORI_SMOOTH_DIST
OST:	Überschleifen der Orientierung durch Vorgabe der Winkeltoleranz in Grad bei Vektorinterpolation mit dem Settingdatum: SD42676 \$SC_ORI_SMOOTH_TOL Bei Rundachsinterpolation wird die vorgegebene Toleranz als maximale Abweichung der Orientierungsachsen angenommen.

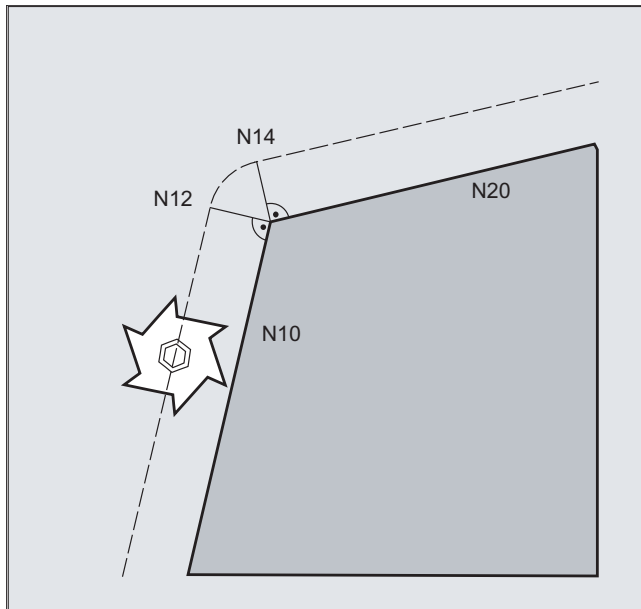
Hinweis

Alle Befehle zum Überschleifen der Werkzeugorientierung (OSOF, OSC, OSS, OSSE, OSD und OST) sind in der G-Funktionsgruppe 34 zusammengefasst. Sie sind modal wirksam, d. h. es kann immer nur einer dieser Befehle wirken.

Beispiele

Beispiel 1: ORIC

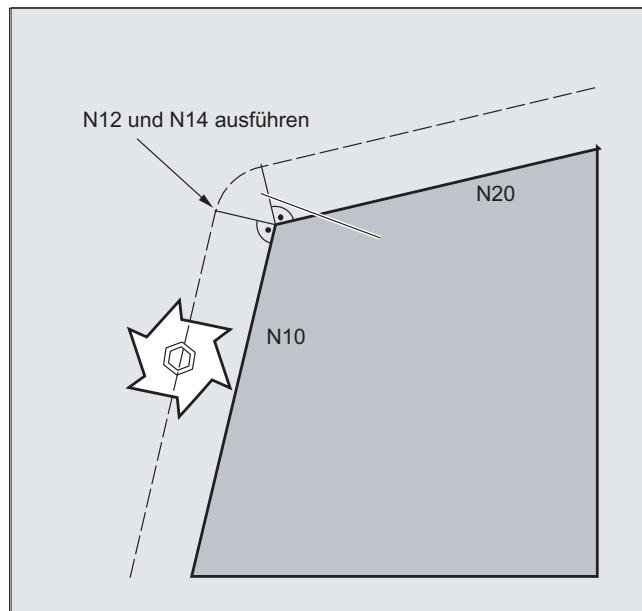
Sind zwischen den Verfahrssätzen N10 und N20 zwei oder mehrere Sätze mit Orientierungsänderungen (z. B. A2=... B2=... C2=...) programmiert und ORIC ist aktiv, so wird der eingefügte Kreissatz entsprechend dem Betrag der Winkeländerungen auf diese Zwischensätze aufgeteilt.



Programmcode	Kommentar
ORIC	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 C2=... B2=...	; Der Kreissatz, der an der Außenecke eingefügt wird, verteilt sich auf N12 und N14, entsprechend der Orientierungsänderung. Kreisbewegung und Orientierungsänderung werden hierbei parallel ausgeführt.
N14 C2=... B2=...	
N20 X =...Y=... Z=... G1 F200	

Beispiel 2: ORID

Ist ORID aktiv, so werden alle Sätze zwischen den beiden Verfahrssätzen am Ende des ersten Verfahrssatzes ausgeführt. Der Kreissatz mit konstanter Orientierung wird unmittelbar vor dem zweiten Verfahrssatz ausgeführt.



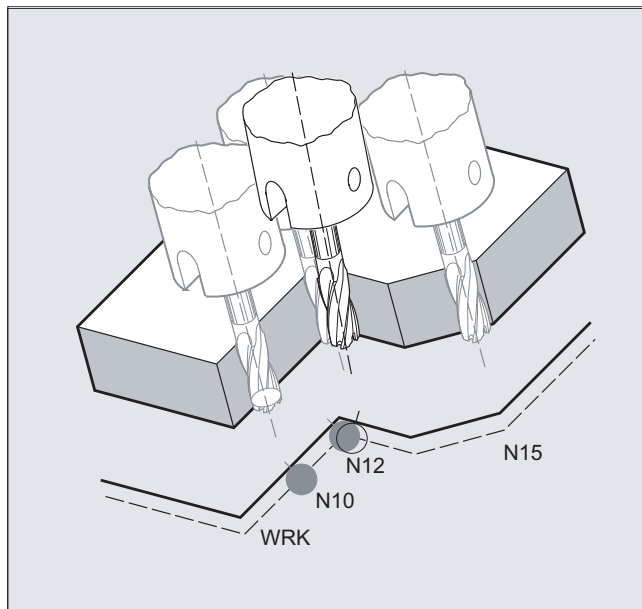
Programmcode	Kommentar
ORID	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 A2=... B2=... C2=...	; Der Satz N12 und N14 wird am Ende von N10 ausgeführt. Danach wird der Kreissatz mit der aktuellen Orientierung ausgefahren.
N14 M20	; Hilfsfunktionen etc.
N20 X... Y... Z...	

Hinweis

Für die Art der Orientierungsänderung an einer Außenecke ist der Programmbefehl maßgebend, welcher im ersten Verfahrtsatz einer Außenecke aktiv ist.

Ohne Orientierungsänderung: Wird die Orientierung an der Satzgrenze nicht verändert, so ist der Werkzeugquerschnitt ein Kreis, der die beiden Konturen berührt.

Beispiel 3: Änderung der Orientierung an einer Innenecke



Programmcode

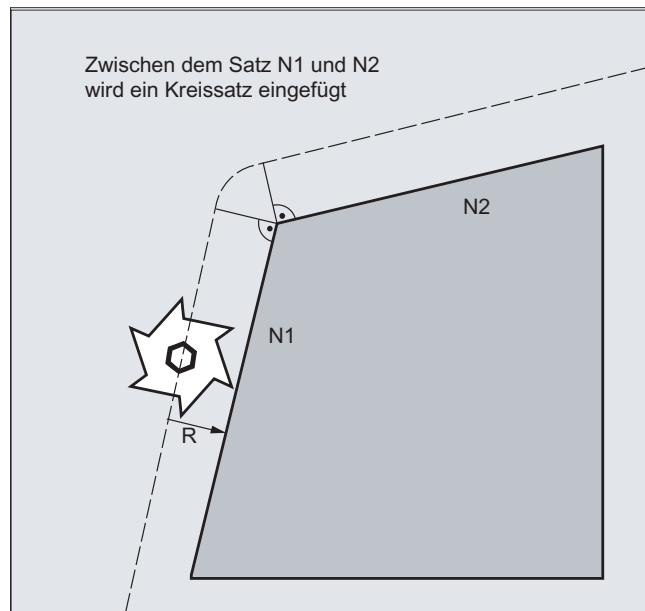
```
ORIC  
N10 X ...Y... Z... G1 F500  
N12 X ...Y... Z... A2=... B2=... C2=...  
N15 X ...Y... Z... A2=... B2=... C2=...
```

Weitere Informationen

Verhalten an Außenecken

An einer Außenecke wird immer ein Kreissatz mit dem Radius des Fräsers eingefügt.

Mit den Programmbefehlen `ORIC` bzw. `ORID` kann festgelegt werden, ob Orientierungsänderungen, die zwischen Satz `N1` und `N2` programmiert wurden, vor Beginn des eingefügten Kreissatzes oder gleichzeitig mit diesem ausgeführt werden.



Ist an Außenecken eine Orientierungsänderung notwendig, so kann diese wahlweise parallel zur Interpolation oder getrennt mit der Bahnbewegung erfolgen.

Bei **ORID** werden zunächst die eingefügten Sätze ohne Bahnbewegung ausgeführt. Der Kreissatz wird unmittelbar vor dem zweiten der beiden Verfahrssätze eingefügt, durch welche die Ecke gebildet wird.

Sind an einer Außenecke mehrere Orientierungssätze eingefügt und **ORIC** ist angewählt, so wird die Kreisbewegung entsprechend den Beträgen der Orientierungsänderungen der einzelnen eingefügten Sätze auf diese verteilt.

Überschleifen der Orientierung mit **OSD** bzw. **OST**

Beim Überschleifen mit **G642** kann die maximale Abweichung für die Konturachsen und die Orientierungsachsen nicht sehr unterschiedlich sein. Die kleinere Toleranz von beiden bestimmt die Form der Überschleifbewegung bzw. Winkeltoleranz, den Orientierungsverlauf relativ stark zu glätten, ohne dabei größere Konturabweichungen hinnehmen zu müssen.

Durch Aktivierung von **OSD** bzw. **OST** ist es möglich, mit einer vorgegebenen Überschleiflänge bzw. Winkeltoleranz sehr geringe Abweichungen des Orientierungsverlaufs ohne gravierende Konturabweichungen "großzügig" zu glätten.

Hinweis

Im Unterschied zum Überschleifen der Kontur (und dem Orientierungsverlauf) mit **G642** wird beim Überschleifen der Orientierung mit **OSD** bzw. **OST** kein eigener Satz gebildet, sondern die Überschleifbewegung wird direkt in die programmierten Originalsätze eingefügt.

Mit **OSD** bzw. **OST** können keine Satzübergänge überschleifen werden bei denen ein Wechsel der Interpolationsart für die Werkzeugorientierung (Vektor → Rundachse, Rundachse → Vektor) stattfindet. Diese Satzübergänge können gegebenenfalls mit den herkömmlichen Überschleiffunktionen **G641**, **G642** bzw. **G643** überschleifen werden.

10.7 Freie D-Nummernvergabe, Schneidenummer

10.7.1 Freie D-Nummernvergabe, Schneidenummer (Adresse CE)

D-Nummer

Die D-Nummern können als Korrekturnummern verwendet werden. Zusätzlich kann über die Adresse CE die Nummer der Schneide adressiert werden. Über die Systemvariable \$TC_DPCE kann die Schneidenummer beschrieben werden.

Voreinstellung: Korrekturr. == Schneidennr.

Über Maschinendaten werden die maximale Anzahl der D-Nummern (Schneidenummern) und die maximale Schneidenanzahl pro Werkzeug festgelegt (→ Maschinenhersteller). Die folgenden Befehle sind nur sinnvoll, wenn die maximale Schneidenummer (MD18105) größer als die Anzahl der Schneiden pro Werkzeug (MD18106) festgelegt wurde. Beachten Sie die Angaben des Maschinenherstellers.

Hinweis

Neben der relativen D-Nummernvergabe können die D-Nummern auch als "flache" bzw. "absolute" D-Nummern (1-32000) ohne Bezug zu einer T-Nummer vergeben werden (innerhalb der Funktion "Flache D-Nummernstruktur").

Literatur

Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1)

10.7.2 Freie D-Nummernvergabe: D-Nummern prüfen (CHKDNO)

Mit dem Befehl `CKKDNO` prüfen Sie, ob die vorhandenen D-Nummern eindeutig vergeben worden sind. Die D-Nummern aller innerhalb einer TO-Einheit definierten Werkzeuge dürfen nur einmal auftreten. Ersatzwerkzeuge werden dabei nicht berücksichtigt.

Syntax

```
state=CHKDNO (Tno1, Tno2, Dno)
```

Bedeutung

state:	=TRUE:	Die D-Nummern wurden für den überprüften Bereich eindeutig vergeben.
	=FALSE:	Es erfolgte eine D-Nummernkollision oder die Parametrierung ist ungültig. Über Tno1, Tno2 und Dno werden die Parameter übergeben, die zur Kollision führten. Diese Daten können im Teileprogramm ausgewertet werden.
CHKDNO (Tno1, Tno2):	Es werden alle D-Nummern der genannten Werkzeuge geprüft.	
CHKDNO (Tno1):	Es werden alle D-Nummern von Tno1 gegen alle anderen Werkzeuge geprüft.	
CHKDNO:	Es werden alle D-Nummern aller Werkzeuge gegen alle anderen Werkzeuge geprüft.	

10.7.3 Freie D-Nummernvergabe: D-Nummern umbenennen (GETDNO, SETDNO)

D-Nummern müssen eindeutig vergeben werden. Zwei verschiedene Schneiden eines Werkzeuges können nicht dieselbe D-Nummer haben.

GETDNO

Dieser Befehl liefert die D-Nummer einer bestimmten Schneide (ce) eines Werkzeuges mit der T-Nummer t. Existiert keine D-Nummer zu den eingegebenen Parametern, wird d=0 gesetzt. Ist die D-Nummer ungültig wird ein Wert größer 32000 zurückgegeben.

SETDNO

Mit diesem Befehl weisen Sie den Wert d der D-Nummer einer Schneide ce des Werkzeuges t zu. Über state wird das Ergebnis dieser Anweisung zurückgegeben (TRUE oder FALSE). Existiert kein Datensatz zu den eingegebenen Parametern wird FALSE zurückgegeben. Syntaxfehler erzeugen einen Alarm. Die D-Nummer kann nicht explizit auf 0 gesetzt werden.

Syntax

```
d = GETDNO (t, ce)
```

```
state = SETDNO (t, ce, d)
```

Bedeutung

d:	D-Nummer der Schneide des Werkzeuges
t:	T-Nummer des Werkzeuges
ce:	Schneidenummer (CE-Nummer) des Werkzeuges
state:	Gibt an, ob der Befehl fehlerfrei ausgeführt werden konnte (TRUE oder FALSE).

Beispiel Umbenennen einer D-Nummer

Programmierung	Kommentar
\$TC_DP2[1,2] = 120	
\$TC_DP3[1,2] = 5.5	
\$TC_DPCE[1,2] = 3	; Schneidenummer CE
...	
N10 def int DNrAlt, DNrNeu = 17	
N20 DNrAlt = GETDNO(1,3)	
N30 SETDNO(1,3, DNrNeu)	

Damit wird der Schneide CE=3 der neue D-Wert 17 zugewiesen. Jetzt werden die Daten dieser Schneide über die D-Nummer 17 angesprochen; sowohl über die Systemvariablen als auch in der Programmierung mit der NC-Adresse.

10.7.4 Freie D-Nummernvergabe: T-Nummer zur vorgegebenen D-Nummer ermitteln (GETACTTD)

Mit dem Befehl `GETACTTD` ermitteln Sie zu einer absoluten D-Nummer die dazugehörige T-Nummer. Es erfolgt keine Prüfung auf Eindeutigkeit. Gibt es mehrere gleiche D-Nummern innerhalb einer TO-Einheit, wird die T-Nummer des ersten gefundenen Werkzeugs zurückgegeben. Bei Verwendung "flacher" D-Nummern ist die Verwendung des Befehls nicht sinnvoll, da hier immer der Wert "1" zurückgegeben wird (keine T-Nummer in der Datenhaltung).

Syntax

`status=GETACTTD(Tnr, Dnr)`

Bedeutung

Dnr:	D-Nummer, für die die T-Nummer gesucht werden soll.	
Tnr:	Gefundene T-Nummer	
status:	Wert:	Bedeutung:
	0	Die T-Nummer wurde gefunden. Tnr erhält den Wert der T-Nummer.
	-1	Zur angegebenen D-Nummer existiert keine T-Nummer; Tnr=0.
	-2	Die D-Nummer ist nicht absolut. Tnr erhält den Wert des ersten gefundenen Werkzeugs, das die D-Nummer mit dem Wert Dnr enthält.
	-5	Die Funktion konnte aus einem anderen Grund nicht ausgeführt werden.

10.7.5 Freie D-Nummernvergabe: D-Nummern ungültig setzen (DZERO)

Der Befehl `DZERO` dient zur Unterstützung während dem Umrüsten. So gekennzeichnete Korrekturdatensätze werden nicht mehr vom Befehl `CHKDNO` geprüft. Um sie wieder zugänglich zu machen, muss die D-Nummer wieder mit `SETDNO` gesetzt werden.

Syntax

DZERO

Bedeutung

DZERO:	Kennzeichnet alle D-Nummern der TO-Einheit als ungültig.
--------	--

10.8 Werkzeugträgerkinematik

Voraussetzungen

Ein Werkzeugträger kann ein Werkzeug nur dann in alle möglichen Raumrichtungen orientieren, wenn

- zwei Drehachsen V_1 und V_2 vorhanden sind.
- die Drehachsen aufeinander senkrecht stehen.
- die Werkzeuglängsachse senkrecht auf der zweiten Drehachse V_2 steht.

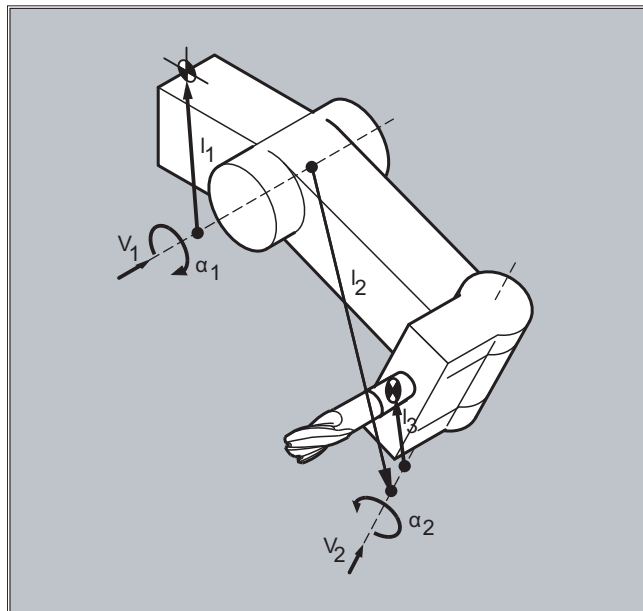
Zusätzlich gilt bei Maschinen, bei denen alle möglichen Orientierungen einstellbar sein müssen, folgende Forderung:

- die Werkzeugorientierung muss senkrecht auf der ersten Drehachse V_1 stehen.

Funktion

Die Werkzeugträgerkinematik mit maximal zwei Drehachsen v_1 oder v_2 wird über die 17 Systemvariablen \$TC_CARR1[m] bis \$TC_CARR17[m] beschrieben. Die Beschreibung des Werkzeugträgers besteht aus:

- dem vektoriellen Abstand von der ersten Drehachse zum Bezugspunkt des Werkzeugträgers I_1 , dem vektoriellen Abstand von erster zu zweiter Drehachse I_2 , dem vektoriellen Abstand von zweiter Drehachse zum Bezugspunkt des Werkzeugs I_3 .
- den Richtungsvektoren beider Drehachsen v_1, v_2 .
- den Drehwinkeln α_1, α_2 um die beiden Achsen. Die Drehwinkel werden mit Blickrichtung in Richtung der Drehachsvektoren im Uhrzeigersinn positiv gezählt.



Für Maschinen mit **aufgelöster Kinematik** (sowohl Werkzeug als auch Werkstück sind drehbar) wurden die Systemvariablen um die Einträge \$TC_CARR18[m] bis \$TC_CARR23[m] erweitert.

Parameter

Funktion der Systemvariablen für orientierbare Werkzeugträger			
Bezeichnung	x-Komponente	y-Komponente	z-Komponente
I_1 Offsetvector	\$TC_CARR1[m]	\$TC_CARR2[m]	\$TC_CARR3[m]
I_2 Offsetvector	\$TC_CARR4[m]	\$TC_CARR5[m]	\$TC_CARR6[m]
v_1 Drehachse	\$TC_CARR7[m]	\$TC_CARR8[m]	\$TC_CARR9[m]
v_2 Drehachse	\$TC_CARR10[m]	\$TC_CARR11[m]	\$TC_CARR12[m]
α_1 Drehwinkel	\$TC_CARR13[m]		
α_2 Drehwinkel	\$TC_CARR14[m]		
I_3 Offsetvector	\$TC_CARR15[m]	\$TC_CARR16[m]	\$TC_CARR17[m]

Erweiterungen der Systemvariablen für orientierbare Werkzeugträger			
Bezeichnung	x-Komponente	y-Komponente	z-Komponente
l_4 Offsetvector	\$TC_CARR18[m]	\$TC_CARR19[m]	\$TC_CARR20[m]
Achsbezeichner Drehachse v_1 Drehachse v_2	Achsbezeichner der Drehachsen v_1 und v_2 (Vorbelegung ist Null) \$TC_CARR21[m] \$TC_CARR22[m]		
Kinematiktyp	\$TC_CARR23[m]		
Tool	Kinematiktyp-T ->	Kinematiktyp-P ->	Kinematiktyp-M
Part Mixed mode	Nur das Werkzeug ist drehbar (Vorbelegung)	Nur das Werkstück ist drehbar	Werkstück & Werkzeug sind drehbar
Offset der Drehachse v_1 Drehachse v_2	Winkel in Grad der Drehachsen v_1 und v_2 bei Einnahme der Grundstellung \$TC_CARR24[m] \$TC_CARR25[m]		
Winkeloffset der Drehachse v_1 Drehachse v_2	Offset der Hirth-Verzahnung in Grad der Drehachsen v_1 und v_2 \$TC_CARR26[m] \$TC_CARR27[m]		
Winkelnkrem. v_1 Drehachse v_2 Drehachse	Inkrement der Hirth-Verzahnung in Grad der Drehachsen v_1 und v_2 \$TC_CARR28[m] \$TC_CARR29[m]		
Min.-Position Drehachse v_1 Drehachse v_2	Software-Limit für Minimalposition der Drehachsen v_1 und v_2 \$TC_CARR30[m] \$TC_CARR31[m]		
Max.-Position Drehachse v_1 Drehachse v_2	Software-Limits für Maximalposition der Drehachsen v_1 und v_2 \$TC_CARR32[m] \$TC_CARR33[m]		
Werkzeugträger Name	Anstelle einer Zahl kann ein Werkzeugträger einem Namen bekommen. \$TC_CARR34[m]		
Anwender: Achsnamen 1 Achsnamen 2 Kennung	Beabsichtigte Verwendung innerhalb der Messzyklen vom Anwender. \$TC_CARR35[m] \$TC_CARR36[m] \$TC_CARR37[m]		
Position	\$TC_CARR38[m]	\$TC_CARR39[m]	\$TC_CARR40[m]
Feinverschiebung	Parameter, die zu den Werten in den Basisparametern addiert werden können.		
l_1 Offsetvector	\$TC_CARR41[m]	\$TC_CARR42[m]	\$TC_CARR43[m]
l_2 Offsetvector	\$TC_CARR44[m]	\$TC_CARR45[m]	\$TC_CARR46[m]
l_3 Offsetvector	\$TC_CARR55[m]	\$TC_CARR56[m]	\$TC_CARR57[m]
l_4 Offsetvector	\$TC_CARR58[m]	\$TC_CARR59[m]	\$TC_CARR60[m]
v_1 Drehachse	\$TC_CARR64[m]		
v_2 Drehachse	\$TC_CARR65[m]		

Hinweis**Erklärungen zu den Parametern**

Mit "m" wird jeweils die Nummer des zu beschreibenden Werkzeugträgers angegeben.

\$TC_CARR47 bis \$TC_CARR54 sowie \$TC_CARR61 bis \$TC_CARR63 sind nicht definiert und führen beim Versuch hierauf lesend oder schreiben zuzugreifen, zu einem Alarm.

Die Anfangs- bzw. Endpunkte der Abstandsvektoren auf den Achsen können frei gewählt werden. Die Drehwinke α_1 , α_2 um die beiden Achsen werden im Grundzustand des Werkzeugträgers mit 0° definiert. Die Kinematik eines Werkzeugträgers kann so auf beliebig viele Möglichkeiten beschrieben werden.

Werkzeugträger mit nur einer oder keiner Drehachse können durch Nullsetzen der Richtungsvektoren einer oder beider Drehachsen beschrieben werden.

Bei einem Werkzeugträger ohne Drehachse wirken die Abstandsvektoren wie zusätzliche Werkzeugkorrekturen, deren Komponenten beim Umschalten der Bearbeitungsebenen (G17 bis G19) $^\circ$ nicht beeinflusst werden.

Erweiterungen der Parameter**Parameter der Drehachsen**

Die Systemvariablen wurden um die Einträge \$TC_CARR24[m] bis \$TC_CARR33[m] erweitert und wie folgt beschrieben:

Den Offset der Drehachsen v_1 , v_2	Veränderung der Position der Drehachse v_1 oder v_2 bei Grundstellung des orientierbaren Werkzeugträgers.
Den Winkeloffset/ Winkelinkrement Drehachsen v_1 , v_2	Offset oder Winkelinkrement der Hirth-Verzahnung der Drehachsen v_1 und v_2 . Programmierter oder berechneter Winkel wird auf den nächstliegenden Wert gerundet, der sich bei ganzzahligem n aus $\phi = s + n \cdot d$ ergibt.
Minimal- und Maximalposition Drehachsen v_1 , v_2	Der Minimalposition/Maximalposition der Drehachse Grenzwinkel (Software-Limit) der Drehachse v_1 und v_2 .

Parameter für den Anwender

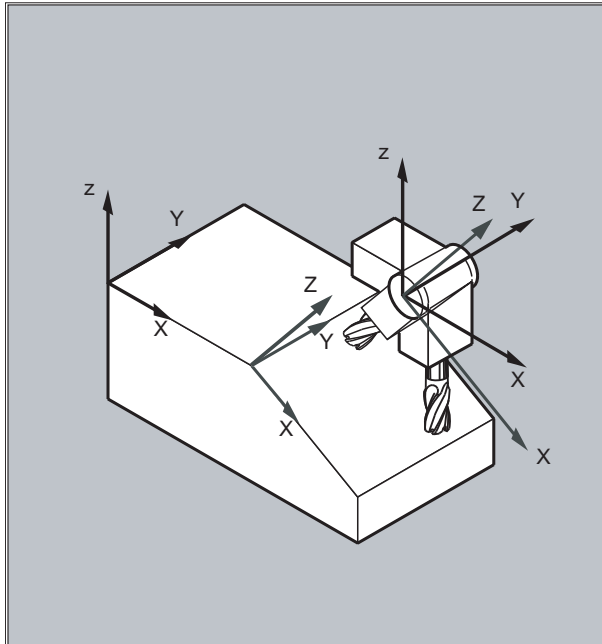
\$TC_CARR34 bis \$TC_CARR40 enthalten Parameter, die den Anwender zur freien Verfügung stehen und bis zum SW 6.4 standardmäßig innerhalb der NCK nicht weiter ausgewertet werden oder keine Bedeutung haben.

Parameter der Feinverschiebung

\$TC_CARR41 bis \$TC_CARR65 enthalten Feinverschiebungsparameter, die zu den Werten in den Basisparametern addiert werden können. Der einem Basisparameter zugeordnete Feinverschiebungswert ergibt sich, wenn zur Parameternummer der Wert 40 addiert wird.

Beispiel

Der im folgenden Beispiel verwendete Werkzeugträger lässt sich durch eine Drehung um die Y-Achse vollständig beschreiben.



Programmcode	Kommentar
N10 \$TC_CARR8[1]=1	; Definition der Y-Komponente der ersten Drehachse des Werkzeugträgers 1.
N20 \$TC_DP1[1,1]=120	; Definition eines Schaftfräasers.
N30 \$TC_DP3[1,1]=20	; Definition eines Schaftfräasers mit Länge 20 mm.
N40 \$TC_DP6[1,1]=5	; Definition eines Schaftfräasers mit Radius 5 mm.
N50 ROT Y37	; Framedefinition mit Drehung von 37° um die Y-Achse.
N60 X0 Y0 Z0 F10000	; Ausgangsposition anfahren.
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; Radiuskorrektur, Werkzeuglängenkorrektur im gedrehten Frame einstellen, Werkzeugträger 1, Werkzeug 1 anwählen.
N80 X40	; Bearbeitung unter einer Drehung von 37° durchführen.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Weitere Informationen

Aufgelöste Kinematik

Für Maschinen mit aufgelöster Kinematik (sowohl Werkzeug als auch Werkstück sind drehbar) wurden die Systemvariablen um die Einträge `$TC_CARR18 [m]` bis `$TC_CARR23 [m]` erweitert und wie folgt beschrieben:

Der drehbare Werkzeuggestisch bestehend aus:

- dem vektoriellen Abstand der zweiten Drehachse v_2 zum Bezugspunkt eines drehbaren Werkzeuggestisches I_4 der dritten Drehachse.

Die Rundachsen bestehend aus:

- den beiden Kanalbezeichnern für den Bezug der Drehachsen v_1 und v_2 , auf deren Position gegebenenfalls bei der Bestimmung der Orientierung des orientierbaren Werkzeugträgers zugegriffen wird.

Der Kinematiktyp mit einem der Werte T, P oder M:

- Kinematiktyp T: Nur das Werkzeug ist drehbar.
- Kinematiktyp P: Nur das Werkstück ist drehbar.
- Kinematiktyp M: Werkzeug und Werkstück sind drehbar.

Löschen der Werkzeugträgerdaten

Mit `$TC_CARR1 [0]=0` können die Daten aller Werkzeugträgerdatensätze gelöscht werden.

Der Kinematiktyp `$TC_CARR23 [T]=T` muss mit einen der drei zulässigen Groß- oder Kleinbuchstaben (T,P,M) belegt werden und sollte aus diesen Grund nicht gelöscht werden.

Ändern der Werkzeugträgerdaten

Jeder der beschriebenen Werte kann durch Zuweisung eines neuen Wertes im Teileprogramm verändert werden. Jedes andere Zeichen als T, P oder M führt bei dem Versuch, den orientierbaren Werkzeugträger zu aktivieren, zu einen Alarm.

Lesen der Werkzeugträgerdaten

Jeder der beschriebenen Werte kann durch Zuweisung an eine Variable im Teileprogramm gelesen werden.

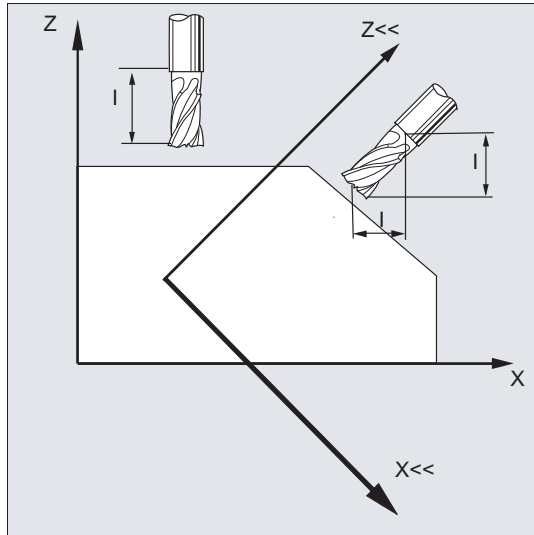
Feinverschiebungen

Ein unzulässiger Feinverschiebungswert wird erst erkannt, wenn ein orientierbarer Werkzeugträger aktiviert wird, der solch einen Wert enthält und gleichzeitig das Settingdatum SD42974 `$SC_TOCARR_FINE_CORRECTION = TRUE` ist.

Der Betrag der zulässigen Feinverschiebung wird über Maschinendaten auf einen maximal zulässigen Wert begrenzt.

10.9 Werkzeuglängenkorrektur für orientierbare Werkzeugträger (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Mit veränderter Raumorientierung des Werkzeugs ändern sich auch dessen Werkzeuglängenkomponenten.



Nach Umrüsten, z. B. durch manuelle Einstellung oder Wechsel des Werkzeugträgers mit fester räumlicher Ausrichtung, müssen daher die Werkzeuglängenkomponenten neu ermittelt werden. Dies erfolgt mit den Wegbefehlen TCOABS und TCOFR.

Bei einem orientierbaren Werkzeugträger eines aktiven Frames kann bei Werkzeuganwahl mit TCOFRZ, TCOFRY und TCOFRX die Richtung, in die das Werkzeug zeigen soll, bestimmt werden.

Syntax

```
TCARR= [<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

Bedeutung

TCARR= [<m>]:	Werkzeugträger mit der Nummer "m" anfordern
TCOABS:	Werkzeuglängenkomponenten aus der aktuellen Werkzeugträgerorientierung berechnen
TCOFR:	Werkzeuglängenkomponenten aus der Orientierung des aktiven Frames bestimmen
TCOFRZ:	Orientierbarer Werkzeugträger aus aktiven Frame, dessen Werkzeug in Z-Richtung zeigt

TCOFRY:	Orientierbarer Werkzeugträger aus aktiven Frame, dessen Werkzeug in Y-Richtung zeigt
TCOFRX:	Orientierbarer Werkzeugträger aus aktiven Frame, dessen Werkzeug in X-Richtung zeigt

Weitere Informationen

Werkzeuglängenkorrektur aus Trägerorientierung (TCOABS)

TCOABS berechnet die Werkzeuglängenkorrektur aus den aktuellen Orientierungswinkeln des Werkzeugträgers; abgelegt in den Systemvariablen \$TC_CARR13 und \$TC_CARR14.

Zur Definition der Werkzeugträgerkinematik mit Systemvariablen siehe "Werkzeugträgerkinematik (Seite 415)".

Zur Neuberechnung der Werkzeuglängenkorrektur bei Frame-Wechsel muss das Werkzeug nochmals ausgewählt werden.

Werkzeugrichtung aus aktiven Frame

Der orientierbare Werkzeugträger kann so eingestellt werden, dass das Werkzeug in folgende Richtungen zeigt:

- mit TCOFR bzw. TCOFRZ in Z-Richtung
- mit TCOFRY in Y-Richtung
- mit TCOFRX in X-Richtung

Ein Umschalten zwischen TCOFR und TCOABS bewirkt eine Neuberechnung der Werkzeuglängenkorrektur.

Werkzeugträger anfordern (TCARR)

Mit TCARR werden mit der Werkzeugträgernummer m dessen Geometriedaten angefordert (Korrekturspeicher).

Mit m=0 wird der aktive Werkzeugträger ausgewählt.

Die Geometriedaten des Werkzeugträgers werden erst nach Aufruf eines Werkzeugs aktiv. Das angewählte Werkzeug bleibt über den Wechsel eines Werkzeugträgers hinaus aktiv.

Die aktuellen Geometriedaten des Werkzeugträgers können auch im Teileprogramm über die entsprechenden Systemvariablen definiert werden.

Neuberechnung der Werkzeuglängenkorrektur (TCOABS) bei Frame-Wechsel

Zur Neuberechnung der Werkzeuglängenkorrektur bei Frame-Wechsel muss das Werkzeug nochmals ausgewählt werden.

Hinweis

Die Werkzeugorientierung muss dem aktiven Frame manuell angepasst werden.

Bei der Berechnung der Werkzeuglängenkorrektur werden in einem Zwischenschritt auch die Drehwinkel des Werkzeugträgers berechnet. Da bei Werkzeugträgern mit zwei Drehachsen im Allgemeinen zwei Drehwinkelpaare existieren, mit denen die Werkzeugorientierung dem aktiven Frame angepasst werden kann, müssen die in den Systemvariablen abgelegten

Drehwinkelwerte zumindest annähernd den mechanisch eingestellten Drehwinkeln entsprechen.

Hinweis**Werkzeugorientierung**

Die Steuerung kann die über die Frame-Orientierung berechneten Verdrehwinkel nicht auf die Einstellbarkeit an der Maschine überprüfen.

Sind die Drehachsen des Werkzeugträgers konstruktiv so angeordnet, dass die durch die Frame-Orientierung berechnete Werkzeugorientierung nicht erreicht werden kann, wird ein Alarm ausgegeben.

Die Kombination von Werkzeugfeinkorrektur und den Funktionalitäten zur Werkzeuglängenkorrektur bei beweglichen Werkzeugträgern ist nicht zulässig. Beim Versuch beide Funktionen gleichzeitig aufzurufen, erfolgt eine Fehlermeldung.

Mit `TOFRAME` ist es möglich, einen Frame aufgrund der Orientierungsrichtung des angewählten Werkzeugträgers zu definieren. Genauere Informationen siehe Kapitel "Frames".

Bei aktiver Orientierungstransformation (3-, 4-, 5-Achstransformation) kann ein Werkzeugträger mit von der Null-Lage abweichender Orientierung angewählt werden, ohne dass dabei ein Alarm ausgegeben wird.

Übergabeparameter von Standard- und Messzyklen

Für die Übergabeparameter von Standard- und Messzyklen gelten definierte Wertebereiche.

Bei Winkelwerten ist der Wertebereich wie folgt festgelegt:

- Drehung um 1. Geometrieachse: -180 Grad bis +180 Grad
- Drehung um 2. Geometrieachse: -90 Grad bis +90 Grad
- Drehung um 3. Geometrieachse: -180 Grad bis +180 Grad

Siehe Kapitel Frames, "Programmierbare Drehung (ROT, AROT, RPL)".

Hinweis

Bei der Übergabe von Winkelwerten an einen Standard- oder Messzyklus ist zu beachten:

Werte kleiner als die Rechenfeinheit der NC sind auf Null zu runden!

Die Rechenfeinheit der NC für Winkelpositionen ist festgelegt im Maschinendatum:

`MD10210 $MN_INT_INCR_PER_DEG`

10.10 Online-Werkzeuglängenkorrektur (TOFFON, TOFFOF)

Über die Systemvariable `$AA_TOFF[<n>]` können die effektiven Werkzeuglängen entsprechend der drei Werkzeugrichtungen dreidimensional in Echtzeit überlagert werden.

Als Index `<n>` werden die drei Geometrieachsbezeichner verwendet. Damit ist die Anzahl der aktiven Korrekturrichtungen durch die zur selben Zeit aktiven Geometrieachsen festgelegt.

Alle Korrekturen können gleichzeitig aktiv sein.

Die Funktion Online-Werkzeuglängenkorrektur ist anwendbar bei:

- Orientierungstransformation TRAORI
- Orientierbare Werkzeugträger TCARR

Hinweis

Die Online-Werkzeuglängenkorrektur ist eine **Option**, die vorher frei geschaltet werden muss. Nur in Verbindung mit einer aktiven Orientierungstransformation oder einem aktiven orientierbaren Werkzeugträger ist diese Funktion sinnvoll.

Syntax

```
TRAORI
TOFFON(<Korrekturrichtung>[,<Offsetwert>])
WHEN TRUE DO $AA_TOFF[<Korrekturrichtung>] ; In Synchronaktionen.
...
TOFFOF(<Korrekturrichtung>)
```

Weitere Erläuterungen zur Programmierung der Online-Werkzeuglängenkorrektur in Bewegungssynchronaktionen siehe "Synchronaktionen (Seite 531)".

Bedeutung

TOFFON:	Online-Werkzeuglängenkorrektur aktivieren	
	<Korrekturrichtung>:	Werkzeugrichtung (X, Y, Z), in der die Online-Werkzeuglängenkorrektur wirksam sein soll.
	<Offsetwert>:	Bei der Aktivierung kann für die entsprechende Korrekturrichtung ein Offsetwert angegeben werden, der sofort herausgefahren wird.
TOFFOF:	Online-Werkzeuglängenkorrektur zurücksetzen Die Korrekturwerte in der angegebenen Korrekturrichtung werden zurückgesetzt und es wird ein Vorlaufstopp ausgelöst.	

Beispiele

Beispiel 1: Anwahl der Werkzeuglängenkorrektur

Programmcode	Kommentar
MD21190 \$MC_TOFF_MODE =1	; Absolute Werte werden angefahren.
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)	; Transformation ein.
N20 TOFFON(Z)	; Aktivierung der Online-WZL-Korrektur für die Z-Werkzeugrichtung.
N30 WHEN TRUE DO \$AA_TOFF[Z]=10 G4 F5	; Für die Z-Werkzeugrichtung wird eine WZL-Korrektur von 10 interpoliert.
...	
N100 XOFFSET=\$AA_TOFF_VAL[X]	; Aktuelle Korrektur in X-Richtung zuweisen.
N120 TOFFON(X,-XOFFSET) G4 F5	; Für die X-Werkzeugrichtung wird die WZL-Korrektur wieder zu 0 zurückgefahren.

Beispiel 2: Abwahl der Werkzeuglängenkorrektur

Programmcode	Kommentar
N10 TRAORI(1)	; Transformation ein.
N20 TOFFON(X)	; Aktivierung der Online-WZL-Korrektur für die X-Werkzeugrichtung.
N30 WHEN TRUE DO \$AA_TOFF[X]=10 G4 F5	; Für die X-Werkzeugrichtung wird eine WZL-Korrektur von 10 interpoliert.
...	
N80 TOFFOF(X)	; Positionsoffset der X-Werkzeugrichtung wird gelöscht: ...\$AA_TOFF[X]=0 Es wird keine Achse verfahren. Zur aktuellen Position im WKS wird der Positionsoffset entsprechend der aktuellen Orientierung hinzugerechnet.

Weitere Informationen

Satzaufbereitung

Bei der Satzaufbereitung im Vorlauf wird der im Hauptlauf wirksame aktuelle Werkzeiglängenoffset mit berücksichtigt. Um die maximal zulässigen Achsgeschwindigkeiten weitgehend ausnutzen zu können, ist es erforderlich, die Satzaufbereitung mit einem Vorlaufstopp `STOPRE` anzuhalten, während ein Werkzeugoffset aufgebaut wird.

Der Werkzeugoffset ist zum Vorlaufzeitpunkt auch immer dann bekannt, wenn die Werkzeiglängenkorrekturen nach Programmstart nicht mehr verändert werden, oder wenn nach einer Veränderung der Werkzeiglängenkorrekturen mehr Sätze abgearbeitet wurden als der IPO-Buffer zwischen Vorlauf und Hauptlauf aufnehmen kann.

Variable `$AA_TOFF_PREP_DIFF`

Das Maß für die Differenz zwischen der aktuellen im Interpolator wirksamen Korrektur und der Korrektur, die zum Zeitpunkt der Satzaufbereitung wirksam war, kann in der Variablen `$AA_TOFF_PREP_DIFF[<n>]` abgefragt werden.

Maschinendaten und Settingdaten einstellen

Für die Online-Werkzeiglängenkorrektur stehen folgende Systemdaten zur Verfügung:

- MD20610 `$MC_ADD_MOVE_ACCEL_RESERVE` (Beschleunigungsreserve für überlagerte Bewegung)
- MD21190 `$MC_TOFF_MODE`
Inhalt der Systemvariable `$AA_TOFF[<n>]` wird als absoluter Wert herausgefahren oder aufintegriert.
- MD21194 `$MC_TOFF_VELO` (Geschwindigkeit der Online-Werkzeiglängenkorrektur)
- MD21196 `$MC_TOFF_ACCEL` (Beschleunigung der Online-Werkzeiglängenkorrektur)
- Settingdatum zur Vorgabe von Grenzwerten:
SD42970 `$SC_TOFF_LIMIT` (Obergrenze des Werkzeiglängenkorrekturwertes)

Literatur:

Funktionshandbuch Sonderfunktionen; F2: Mehrachstransformationen

10.11 Schneidendaten-Modifikation bei drehbaren Werkzeugen (CUTMOD)

Mit der Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen" können die veränderten geometrischen Verhältnisse, die sich bei der Drehung von Werkzeugen (vorwiegend Drehwerkzeuge, aber auch Bohr- und Fräswerkzeuge) relativ zum bearbeiteten Werkstück ergeben, bei der Werkzeugkorrektur berücksichtigt werden.

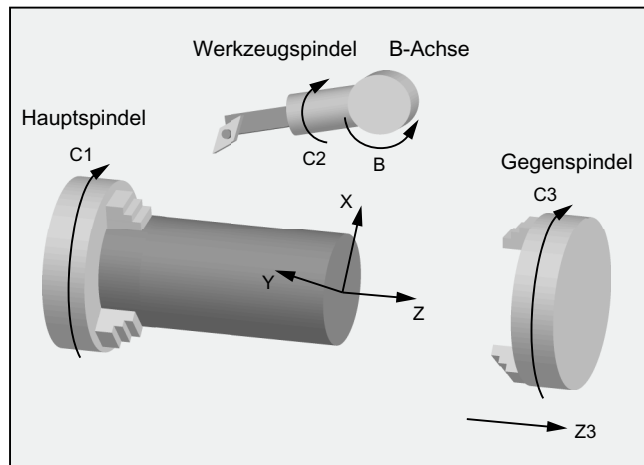


Bild 10-1 Drehbares Werkzeug bei einer Drehmaschine

Die aktuelle Drehung des Werkzeugs wird dabei immer aus einem aktuell aktiven orientierbaren Werkzeugträger (siehe "Werkzeuglängenkorrektur für orientierbare Werkzeugträger (Seite 421)") ermittelt.

Die Funktion wird aktiviert mit dem Befehl `CUTMOD`.

Syntax

`CUTMOD=<Wert>`

Bedeutung

CUTMOD:	Befehl zum Einschalten der Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen"	
<Wert>:	Dem CUTMOD-Befehl können folgende Werte zugewiesen werden:	
	0	Die Funktion ist deaktiviert. Die von den Systemvariablen \$P_AD... gelieferten Werte sind gleich den korrespondierenden Werkzeugparametern.
	> 0	Die Funktion wird aktiviert, falls ein orientierbarer Werkzeugträger mit der angegebenen Nummer aktiv ist, d. h. die Aktivierung ist an einen bestimmten orientierbaren Werkzeugträger gebunden. Die von den Systemvariablen \$P_AD... gelieferten Werte sind gegenüber den korrespondierenden Werkzeugparametern abhängig von der aktiven Drehung gegebenenfalls modifiziert. Die Deaktivierung des bezeichneten orientierbaren Werkzeugträgers deaktiviert die Funktion temporär, die Aktivierung eines anderen orientierbaren Werkzeugträgers deaktiviert sie permanent. Im ersten Fall wird die Funktion deshalb bei erneuter Anwahl des gleichen orientierbaren Werkzeugträgers wieder aktiviert, im zweiten Fall ist eine erneute Anwahl notwendig, auch dann wenn zu einem späteren Zeitpunkt der orientierbare Werkzeugträger mit der angegebenen Nummer erneut aktiviert wird. Die Funktion wird durch Reset nicht beeinflusst.
	-1	Die Funktion wird immer aktiviert, falls ein orientierbarer Werkzeugträger aktiv ist. Beim Wechsel des Werkzeugträgers oder bei dessen Abwahl und einer späteren erneuten Anwahl muss CUTMOD nicht erneut gesetzt werden.
	-2	Die Funktion wird immer aktiviert, falls ein orientierbarer Werkzeugträger aktiv ist, dessen Nummer gleich der des aktuell aktiven orientierbaren Werkzeugträgers ist. Ist kein orientierbarer Werkzeugträger aktiv, ist das gleichbedeutend mit CUTMOD=0. Ist ein orientierbarer Werkzeugträger aktiv, ist das gleichbedeutend mit der unmittelbaren Angabe der aktuellen Werkzeugträgernummer.
	< -2	Werte kleiner -2 werden ignoriert, d. h. dieser Fall wird so behandelt, als wäre CUTMOD nicht programmiert. Hinweis: Dieser Wertebereich sollte nicht verwendet werden, weil er für eventuelle spätere Erweiterungen reserviert ist.

Hinweis**SD42984 \$SC_CUTDIRMOD**

Die über den Befehl CUTMOD aktivierbare Funktion ersetzt die über das Settingdatum SD42984 \$SC_CUTDIRMOD aktivierbare Funktion. Diese Funktion steht jedoch weiterhin unverändert zur Verfügung. Da es aber nicht sinnvoll ist, beide Funktionen parallel zu nutzen, kann sie nur aktiviert werden, wenn CUTMOD gleich Null ist.

Beispiel

Das folgende Beispiel bezieht sich auf ein Werkzeug mit der Schneidenlage 3 und einem orientierbaren Werkzeugträger, der das Werkzeug um die B-Achse drehen kann.

10.11 Schneidendaten-Modifikation bei drehbaren Werkzeugen (CUTMOD)

Die Zahlenwerte in den Kommentaren geben jeweils die Satzendpositionen in Maschinenkoordinaten (MKS) in der Reihenfolge X, Y, Z an.

Programmcode	Kommentar		
N10 \$TC_DP1[1,1]=500			
N20 \$TC_DP2[1,1]=3	Schneidenlage		
N30 \$TC_DP3[1,1]=12			
N40 \$TC_DP4[1,1]=1			
N50 \$TC_DP6[1,1]=6			
N60 \$TC_DP10[1,1]=110	; Halterwinkel		
N70 \$TC_DP11[1,1]=3	; Schnitttrichtung		
N80 \$TC_DP24[1,1]=25	; Freiwinkel		
N90 \$TC_CARR7[2]=0 \$TC_CARR8[2]=1 \$TC_CARR9[2]=0	; B-Achse		
N100 \$TC_CARR10[2]=0 \$TC_CARR11[2]=0 \$TC_CARR12[2]=1	; C-Achse		
N110 \$TC_CARR13[2]=0			
N120 \$TC_CARR14[2]=0			
N130 \$TC_CARR21[2]=X			
N140 \$TC_CARR22[2]=X			
N150 \$TC_CARR23[2]="M"			
N160 TCOABS CUTMOD=0			
N170 G18 T1 D1 TCARR=2	X	Y	Z
N180 X0 Y0 Z0 F10000	12.000	0.000	1.000
N190 \$TC_CARR13[2]=30			
N200 TCARR=2			
N210 X0 Y0 Z0	10.892	0.000	-5.134
N220 G42 Z-10	8.696	0.000	-17.330
N230 Z-20	8.696	0.000	-21.330
N240 X10	12.696	0.000	-21.330
N250 G40 X20 Z0	30.892	0.000	-5.134
N260 CUTMOD=2 X0 Y0 Z0	8.696	0.000	-7.330
N270 G42 Z-10	8.696	0.000	-17.330
N280 Z-20	8.696	0.000	-21.330
N290 X10	12.696	0.000	-21.330
N300 G40 X20 Z0	28.696	0.000	-7.330
N310 M30			

Erläuterungen:

In Satz N180 wird zunächst das Werkzeug bei CUTMOD=0 und nicht gedrehtem orientierbaren Werkzeugträger angewählt. Da alle Offsetvektoren des orientierbaren Werkzeugträgers 0 sind, wird die Position angefahren, die den in \$TC_DP3[1,1] und \$TC_DP4[1,1] angegebenen Werkzeuglängen entspricht.

In Satz N200 wird der orientierbare Werkzeugträger mit einer Drehung von 30° um die B-Achse aktiviert. Da die Schneidenlage wegen CUTMOD=0 nicht modifiziert wird, ist nach wie vor der alte Schneidenbezugspunkt maßgebend. Deshalb wird in Satz N210 die Position angefahren, die den alten Schneidenbezugspunkt im Nullpunkt beibehält (d. h. der Vektor (1, 12) wird in der Z/X-Ebene um 30° gedreht).

In Satz N260 ist im Unterschied zu Satz N200 CUTMOD=2 wirksam. Aufgrund der Drehung des orientierbaren Werkzeugträgers wird die modifizierte Schneidenlage 8. Daraus folgen auch abweichende Achspositionen.

In den Sätzen N220 bzw. N270 wird jeweils die Werkzeugradiuskorrektur (WRK) aktiviert. Die unterschiedliche Schneidenlage in beiden Programmstücken hat auf die Endpositionen der Sätze, in denen die WRK aktiv ist, keinen Einfluss, die entsprechenden Positionen sind deshalb identisch. Erst in den Abwahlsätzen N260 bzw. N300 wirken sich die unterschiedlichen Schneidenlagen wieder aus.

Weitere Informationen

Wirksamkeit der modifizierten Schneidendaten

Die modifizierte Schneidenlage und der modifizierte Schneidenbezugspunkt werden bei Programmierung auch für ein bereits aktives Werkzeug sofort wirksam. Eine Werkzeugneuanwahl ist dazu nicht notwendig.

Einfluss der aktiven Arbeitsebene

Für die Bestimmung von modifizierter Schneidenlage, Schnittrichtung und Halter- bzw. Freiwinkel ist die Betrachtung der Schneide in der jeweils aktiven Ebene (G17 - G19) maßgebend.

Enthält jedoch das Settingdatum SD42940 \$SC_TOOL_LENGTH_CONST (Wechsel der Werkzeuglängenkomponenten bei Ebenenwechsel) einen gültigen Wert ungleich Null (plus oder minus 17, 18 oder 19), so bestimmt dessen Inhalt die Ebene, in der die relevanten Größen betrachtet werden.

Systemvariablen

Folgende Systemvariablen stehen zur Verfügung:

Systemvariablen	Bedeutung
\$P_CUTMOD_ANG / \$AC_CUTMOD_ANG	Liefert den (nicht gerundeten) Winkel in der aktiven Bearbeitungsebene, der für die Modifikation der Schneidendaten (Schneidenlage, Schnittrichtung, Freiwinkel und Halterwinkel) bei den mit CUTMOD bzw. \$SC_CUTDIRMOD aktivierten Funktionen zugrunde gelegt wurde. \$P_CUTMOD_ANG bezieht sich auf den aktuellen Zustand im Vorlauf, \$AC_CUTMOD_ANG auf den aktuellen Hauptlaufsatz.
\$P_CUTMOD / \$AC_CUTMOD	Liest den aktuell gültigen Wert, der zuletzt mit dem Befehl CUTMOD programmiert wurde (Nummer des Werkzeugträgers, für den die Schneidendaten-Modifikation aktiviert werden soll). War der letzte programmierte CUTMOD-Wert = -2 (Aktivierung mit dem aktuell aktiven orientierbaren Werkzeugträger), dann wird in \$P_CUTMOD nicht der Wert -2, sondern die Nummer des zum Zeitpunkt der Programmierung aktiven orientierbaren Werkzeugträgers zurückgeliefert. \$P_CUTMOD bezieht sich auf den aktuellen Zustand im Vorlauf, \$AC_CUTMOD auf den aktuellen Hauptlaufsatz.
\$P_CUT_INV / \$AC_CUT_INV	Liefert den Wert TRUE, wenn das Werkzeug so gedreht ist, dass die Spindeldrehrichtung invertiert werden muss. Dazu müssen in dem Satz, auf den sich die jeweilige Leseoperation bezieht, die folgenden vier Bedingungen erfüllt sein: <ol style="list-style-type: none"> 1. Es ist ein Dreh- oder Schleifwerkzeug aktiv (Werkzeugtypen 400 bis 599 und / oder SD42950 \$SC_TOOL_LENGTH_TYPE = 2). 2. Die Schneidenbeeinflussung wurde mit dem Sprachbefehl CUTMOD aktiviert. 3. Es ist ein orientierbarer Werkzeugträger aktiv, der durch den numerischen Wert von CUTMOD bezeichnet wurde. 4. Der orientierbare Werkzeugträger dreht das Werkzeug um eine Achse in der Bearbeitungsebene (typischerweise die C-Achse) so, dass die resultierende Normale der Werkzeugschneide gegenüber der Ausgangslage um mehr als 90° (typischerweise 180°) gedreht ist. Ist mindestens eine der genannten vier Bedingungen nicht erfüllt, ist der Inhalt der Variablen FALSE. Für Werkzeuge, deren Schneidenlage nicht definiert ist, ist der Wert der Variablen immer FALSE. \$P_CUT_INV bezieht sich auf den aktuellen Zustand im Vorlauf und \$AC_CUT_INV auf den aktuellen Hauptlaufsatz.

Alle Hauptlaufvariablen (\$AC_CUTMOD_ANG, \$AC_CUTMOD und \$AC_CUT_INV) können in Synchronaktionen gelesen werden. Ein Lesezugriff aus dem Vorlauf generiert einen Vorlaufstopp.

Modifizierte Schneidendaten:

Falls eine Werkzeugdrehung aktiv ist, werden die modifizierten Daten in den folgenden Systemvariablen zur Verfügung gestellt:

Systemvariable	Bedeutung
\$P_AD[2]	Schneidenlage
\$P_AD[10]	Halterwinkel

Systemvariable	Bedeutung
\$P_AD[11]	Schnitttrichtung
\$P_AD[24]	Freiwinkel

Hinweis

Die Daten sind gegenüber den korrespondierenden Werkzeugparametern (\$TC_DP2[... , ...] usw.) immer dann modifiziert, wenn die Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen" mit dem Befehl CUTMOD aktiviert wurde und ein orientierbarer Werkzeugträger aktiv ist, der eine Werkzeugdrehung bewirkt.

Literatur

Weitere Informationen zur Funktion "Schneidendaten-Modifikation bei drehbaren Werkzeugen" siehe:

Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1)

Bahnverhalten

11.1 Tangentialsteuerung

11.1.1 Kopplung definieren (TANG)

Über die vordefinierte Prozedur TANG(...) wird eine Tangentialkopplung zwischen einer Rundachse als Folgeachse und zwei Geometrieachsen als Leitachsen definiert. Die Folgeachse wird dabei kontinuierlich zur Bahntangente der Leitachsen ausgerichtet.

Syntax

TANG(<Folgeachse>, <Leitachse_1>, <Leitachse_2>, <Koppelfaktor>, <Koordinatensystem>, <Optimierung>)

Bedeutung

TANG (...):	Definieren einer Tangentialkopplung		
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)		
	Datentyp:	AXIS	
	Wertebereich:	Kanalachsnamen	
<Leitachse_1> <Leitachse_2>:	Achsennamen der Leitachsen (Geometrieachsen) ¹⁾		
	Datentyp:	AXIS	
	Wertebereich:	Geometrieachsnamen des Kanals	
<Koppelfaktor>:	Faktor n der Winkeländerung der Folgeachse zur Änderung der Bahntangente der Leitachsen: Winkeländerung _{Folgeachse} = Winkeländerung _{Bahntangente} * n		
	Datentyp:	REAL	
	Defaultwert:	1,0	
<Koordinatensystem>:	Wirksames Koordinatensystem ²⁾		
	Datentyp:	CHAR	
	Wert:	"B":	Basiskoordinatensystem (Defaultwert)
		"W":	Werkstückkoordinatensystem (nicht verfügbar)

<Optimierung>:	Optimierungsart	
	Datentyp:	CHAR
	Wert:	"S": Standard (Defaultwert) Die Dynamik der Rundachse hat keine Rückwirkung auf die Leitachsen. Ist die Dynamik der Rundachse höher als für die Nachführung erforderlich, ist dieses Verfahren ausreichend genau. Ist die Dynamik der Rundachse nicht hoch genug, um der Änderung der Bahntangenten zu folgen, weicht die Ausrichtung der Rundachse entlang eines nicht definierten Überschleifwegs von der Sollausrichtung ab.
	"P":	Die Dynamik der Rundachse wird bei der Bahnplanung der Leitachsen berücksichtigt. Dazu müssen beim Einschalten der Tangentialkopplung mit TANGON() zwei zusätzliche Parameter angegeben werden: <ul style="list-style-type: none"> • Überschleifweg • Winkeltoleranz Siehe Kapitel "Kopplung einschalten (TANGON) (Seite 436)" Hinweis Im Zusammenhang mit kinematischen Transformationen wird empfohlen, die Optimierungsart "P" zu verwenden.
Hinweis Defaultwerte müssen nicht explizit programmiert werden.		
¹⁾ Hinweis Als Leitachsen der Tangentialkopplung müssen die Geometrieachsen verwendet werden, die, bezogen auf die Grundstellung der Maschine, im Maschinenkoordinatensystem (MKS) die programmierte Bahn abfahren. Wird z. B. auf einer Fräsmaschine mit Schwenkkopf der Schwenkzyklus CYCLE800 verwendet, erfolgt, abhängig von der Konfiguration des Zyklus, die Interpolation im WKS z. B. mit den Geometrieachsen X und Y. Die Tangentialkopplung muss aber mit den Geometrieachsen als Leitachsen definiert werden, die im MKS die programmierte Bahn abfahren. Dazu sind als Leitachsen die Geometrieachsen im ungeschwenkten Zustand der Maschine zu verwenden.		
²⁾ Hinweis Das Basiskoordinatensystem (BKS) darf gegenüber dem MKS nicht gedreht sein. Wird z. B. über den Befehl ROT oder über den Schwenkzyklus CYCLE800 das BKS gedreht, arbeitet die Tangentialsteuerung nicht mehr korrekt.		

11.1.2 Zwischensatzerzeugung einschalten (TLIFT)

Überschreitet die Tangentenänderung der Folgeachse an einer Stelle der programmierten Bahn der Leitachsen den im Maschinendatum MD37400 \$MA_EPS_TLIFT_TANG_STEP parametrisierten Grenzwert, erfolgt die weitere Bahnplanung abhängig vom eingestellten Eckenverhalten. Ohne Verwendung der vordefinierten Prozedur TLIFT(...) wird entsprechend des im Zusammenhang mit TANG(...) (Seite 433) und TANGON(...) (Seite 436) programmierten Überschleifenverhaltens verfahren.

Zwischensatzerzeugung einschalten

Mit Programmierung von TLIFT(...) im Anschluss an TANG(...) wird vom Vorlauf beim Erkennen einer Ecke an dieser Stelle der Bahn ein von der Steuerung automatisch generierter Zwischensatz eingefügt.

Bei der Abarbeitung des Programms werden dann bei Erreichen des Zwischensatzes die Leitachsen angehalten. Im Zwischensatz wird die Folgeachse mit maximaler Achsdynamik in Richtung der Bahntangente des nachfolgenden Satzes gedreht. Danach werden die Leitachsen weiter auf der programmierten Bahn verfahren.

Zwischensatzerzeugung ausschalten

Zum Ausschalten der Zwischensatzerzeugung muss die Tangentialkopplung mittels TANG(...), aber ohne nachfolgendes Einschalten der Zwischensatzerzeugung mittels TLIFT(...), erneut definiert werden.

Syntax

TLIFT (<Folgeachse>)

Bedeutung

TLIFT(...):	Einschalten der Eckenerkennung mit Zwischensatzberechnung	
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsenamen

Drehgeschwindigkeit der Folgeachse

Bahnachse

Wurde die Folgeachse vor der Aktivierung der Tangentialkopplung schon als Bahnachse verfahren, erfolgt die Drehbewegung im Zwischensatz als Bahnachse.

Durch die Vorgabe des Bezugsradius mit FREF [<Achse>]=0.001, erfolgt die Drehbewegung mit der parametrisierten maximalen Achsgeschwindigkeit:

MD32000 \$MA_MAX_AX_VELO[<Folgeachse>]

Positionierachse

Wurde die Folgeachse vor der Aktivierung der Tangentialkopplung noch nicht als Bahnachse verfahren, erfolgt die Drehung im Zwischensatz als Positionierachse.

Die Drehbewegung erfolgt dabei mit der parametrisierten Positionierachsgeschwindigkeit:

MD32060 \$MA_POS_AX_VELO[<Folgeachse>]

11.1.3 Kopplung einschalten (TANGON)

Über die vordefinierte Prozedur TANGON(...) wird eine zuvor mit TANG(...) (Seite 433) definierte Tangentialkopplung eingeschaltet. Die Folgeachse wird dann beim nachfolgenden Verfahren der Leitachsen kontinuierlich zur Bahntangente ausgerichtet.

Winkel der Folgeachse

Der Winkel, den die Folgeachse in Bezug zur Bahntangente einnimmt, ist abhängig vom in TANG(...) vorgegebenen Übersetzungsverhältnis, dem im Maschinendatum MD37402 \$MA_TANG_OFFSET parametrisierten Offsetwinkel und dem dazu additiv wirkenden bei TANGON(...) vorgegebenen Offsetwinkel.

Optimierung "P"

Wurde bei der Definition der Tangentialkopplung (TANG(...)) als Optimierungsparameter der Wert "P" angegeben, muss beim Einschalten der Kopplung der Parameter "Überschleifweg" und optional der Parameter "Winkeltoleranz" angegeben werden.

Wird als Winkeltoleranz der Wert 0 vorgegeben, wirkt nur der Parameter "Überschleifweg".

Wird als Winkeltoleranz ein Wert größer 0 vorgegeben, ergibt sich der wirksame Überschleifweg aus dem Minimum des parametrisierten Überschleifwegs und des Überschleifwegs aufgrund der parametrisierten Winkeltoleranz.

Ist die Dynamik der Folgeachse nicht ausreichend, um den parametrisierten Bedingungen zu folgen, wird die Bahngeschwindigkeit der Leitachsen entsprechend reduziert.

Syntax

```
TANGON(<Folgeachse>, <Offsetwinkel>, <Überschleifweg>,
<Winkeltoleranz>)
```

Bedeutung

TANGON(...):	Einschalten einer Tangentialkopplung	
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsenamen
<Offsetwinkel>:	Offsetwinkel der Folgeachse zur Bahntangente Bezugspunkt ist der Nullpunkt der Rundachse.	
	Datentyp:	REAL
<Überschleifweg>:	Maximal zulässiger Überschleifweg Würde der Überschleifweg aufgrund der Dynamikbedingungen größer, wird die Bahngeschwindigkeit der Leitachsen vermindert.	
	Datentyp:	REAL
<Winkeltoleranz>:	Maximal zulässige Toleranz bezüglich des vorgegebenen Winkels zwi- schen der Folgeachs-Nullstellung und der Bahntangente	
	Datentyp:	REAL

11.1.4 Kopplung ausschalten (TANGOF)

Über die vordefinierte Prozedur TANGOF(...) wird eine mit TANG(...) (Seite 433) definierte und mit TANGON(...) (Seite 436) eingeschaltete Tangentialkopplung ausgeschaltet. Die Folgeachse wird dann nicht mehr auf die Bahntangente der Leitachse ausgerichtet. Die Kopplung der Folgeachse an die Leitachsen bleibt aber auch nach dem Ausschalten weiterhin bestehen, was z.B. folgende Funktionen verhindert:

- Ebenenwechsel
- Geometrieachsumschaltung
- Definition einer neuen Tangentialkopplung für die Folgeachse

Das vollständige Aufheben der Bindung der Folgeachse an die Leitachsen erfolgt erst nach dem Löschen der Kopplung durch TANGDEL(...) (Seite 437).

Programmierung

TANGOF (<Folgeachse>)

Bedeutung

TANGOF (. . .):	Ausschalten einer Tangentialkopplung	
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsnamen

11.1.5 Kopplung löschen (TANGDEL)

Eine mit TANG(...) (Seite 433) definierte Tangentialkopplung bleibt auch nach dem Ausschalten der Tangentialkopplung mit TANGOF(...) (Seite 437) hinaus bestehen. Die bestehende Tangentialkopplung verhindert dann z.B. weiterhin folgende Funktionen:

- Ebenenwechsel
- Geometrieachsumschaltung
- Definition einer neuen Tangentialkopplung für die Folgeachse

Mit der vordefinierten Prozedur TANGDEL(...) wird nach dem Ausschalten der Tangentialkopplung mit TANGOF(...) die bestehende Tangentialkopplung gelöscht.

Syntax

TANGDEL (<Folgeachse>)

Bedeutung

TANGDEL (. . .):	Löschen einer mit TANG() definierten Tangentialkopplung	
	Wirksamkeit:	Satzweise

<Folgeachse>:	Achsenname der Folgeachse, deren Tangentialkopplung gelöscht werden soll	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsenamen

Beispiele

Leitachwechsel

Bevor für die Folgeachse eine neue Tangentialkopplung mit einer anderen Leitachse definiert werden kann, muss die bestehende Tangentialkopplung zuerst gelöscht werden.

Programmcode	Kommentar
N10 TANG (A, X, Y, 1)	; Tangentialkopplung für Folgeachse A definieren: A zu X und Y
N20 TANGON(A)	; Tangentialkopplung für Folgeachse A einschalten
N30 X10 Y20	
...	
N80 TANGOF(A)	; Tangentialkopplung für Folgeachse A ausschalten
N90 TANGDEL (A)	; Tangentialkopplung für Folgeachse A löschen
...	
N120 TANG (A, X, Z)	; Neue Tangentialkopplung für Folgeachse A definieren
N130 TANGON(A)	; Neue Tangentialkopplung für Folgeachse A einschalten
...	

Geometrieachsumschaltung

Bevor eine Geometrieachsumschaltung für eine bestehende Kopplung vorgenommen werden kann, muss die Kopplung zuerst gelöscht werden.

Programmcode	Kommentar
N10 GEOAX(2, Y1)	; 2. Geometrieachse = Maschinenachse Y1
N20 TANG(A, X, Y)	; Tangentialkopplung für Folgeachse A definieren
N30 TANGON(A, 90)	; Tangentialkopplung für Folgeachse A einschalten
N40 G2 F8000 X0 Y0 I0 J50	; Bewegungssatz
N50 TANGOF(A)	; Tangentialkopplung für Folgeachse A ausschalten
N60 TANGDEL (A)	; Tangentialkopplung für Folgeachse A löschen
N70 GEOAX (2, Y2)	; 2. Geometrieachse = Maschinenachse Y2
N80 TANG(A, X, Y)	; Neue Tangentialkopplung für Folgeachse A definieren
N90 TANGON(A, 90)	; Neue Tangentialkopplung für Folgeachse A einschalten
...	

11.2 Vorschubverlauf (FNORM, FLIN, FCUB, FPO)

Zur flexibleren Vorgabe des Vorschubverlaufs wird die Vorschubprogrammierung nach DIN 66025 um lineare und kubische Verläufe erweitert.

Die kubischen Verläufe können direkt oder als interpolierende Splines programmiert werden. Hierdurch lassen sich - abhängig von der Krümmung des zu bearbeitenden Werkstücks - kontinuierlich glatte Geschwindigkeitsverläufe programmieren.

Diese Geschwindigkeitsverläufe ermöglichen ruckfreie Beschleunigungsänderungen und hierdurch Fertigung gleichmäßiger Werkstückoberflächen.

Syntax

F... FNORM
 F... FLIN
 F... FCUB
 F=FPO (... , ... , ...)

Bedeutung

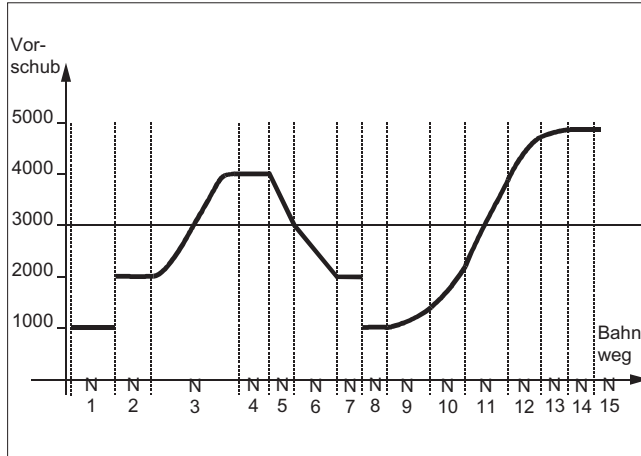
FNORM:	Grundeinstellung. Der Vorschubwert wird über den Bahnweg des Satzes vorgegeben und gilt danach als modaler Wert.
FLIN:	Bahngeschwindigkeitsprofil linear: Der Vorschubwert wird vom aktuellen Wert am Satzanfang bis zum Satzende über den Bahnweg linear eingefahren und gilt danach als modaler Wert. Dieses Verhalten kann mit G93 und G94 kombiniert werden.
FCUB:	Bahngeschwindigkeitsprofil kubisch: Die satzweise programmierten F-Werte werden - bezogen auf den Satzende - durch einen Spline verbunden. Der Spline beginnt und endet tangential zur vorhergehenden bzw. nachfolgenden Vorschubangabe und wirkt mit G93 und G94. Fehlt in einem Satz die F-Adresse, so wird hierfür der zuletzt programmierte F-Wert verwendet.
F=FPO... :	Bahngeschwindigkeitsprofil über Polynom: Die F-Adresse bezeichnet den Vorschubverlauf über ein Polynom vom aktuellen Wert bis zum Satzende. Der Endwert gilt danach als modaler Wert.

Vorschuboptimierung bei gekrümmten Bahnstücken

Vorschub-Polynom $F=FPO$ und Vorschubspline $FCUB$ sollten immer mit konstanter Schnittgeschwindigkeit CFC abgefahren werden. Hierdurch lässt sich ein beschleunigungsstetiges Sollvorschubprofil erzeugen.

Beispiel: Verschiedene Vorschubprofile

In diesem Beispiel finden Sie die Programmierung und grafische Darstellung verschiedener Vorschubprofile.

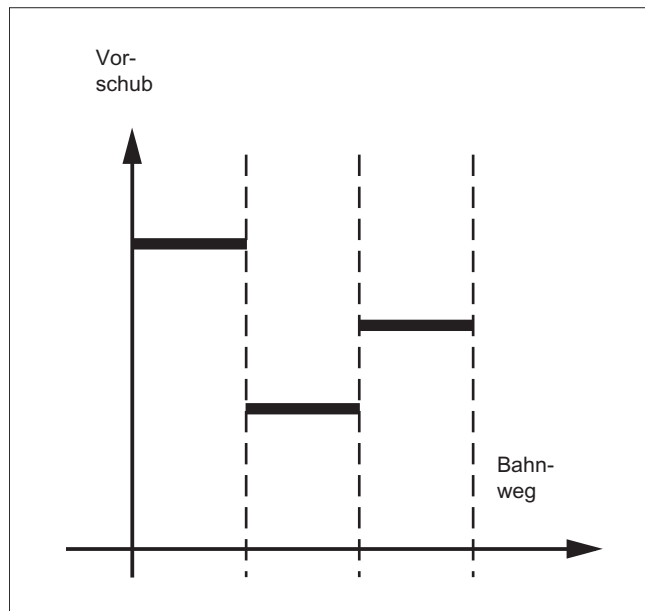


Programmcode	Kommentar
N1 F1000 FNORM G1 X8 G91 G64	; Konstantes Vorschubprofil, Kettenmaßangabe
N2 F2000 X7	; Sprunghafte Sollgeschwindigkeitsänderung
N3 F=FPO(4000, 6000, -4000)	; Vorschubprofil über Polynom mit Vorschub 4000 am Satzende.
N4 X6	; Polynomvorschub 4000 gilt als modaler Wert.
N5 F3000 FLIN X5	; Lineares Vorschubprofil
N6 F2000 X8	; Lineares Vorschubprofil
N7 X5	; Linearer Vorschub gilt als modaler Wert
N8 F1000 FNORM X5	; Konstantes Vorschubprofil mit sprunghafter Beschleunigungsänderung.
N9 F1400 FCUB X8	; Alle folgenden satzweise programmierten F-Werte werden mit Splines verbunden.
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; Splineprofil ausschalten.
N14 FNORM X5	
N15 X20	

Weitere Informationen

FNORM

Die Vorschubadresse F bezeichnet den Bahnvorschub als konstanten Wert nach DIN 66025. Mehr Informationen hierzu finden Sie im Programmierhandbuch "Grundlagen".

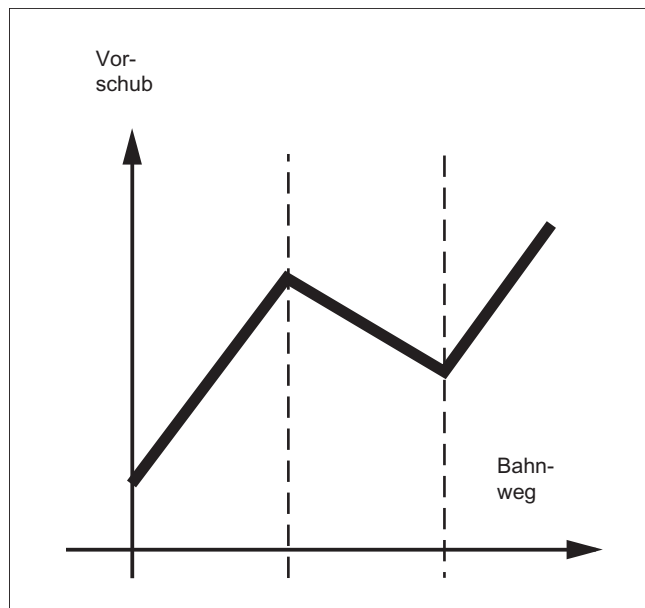


FLIN

Der Vorschubverlauf wird vom aktuellen Vorschubwert zum programmierten F-Wert linear bis Satzende eingefahren.

Beispiel:

```
N30 F1400 FLIN X50
```



FCUB

Der Vorschub wird vom aktuellen Vorschubwert zum programmierten F-Wert bis Satzende im kubischen Verlauf eingefahren. Die Steuerung verbindet alle mit aktivem FCUB satzweise

11.2 Vorschubverlauf (FNORM, FLIN, FCUB, FPO)

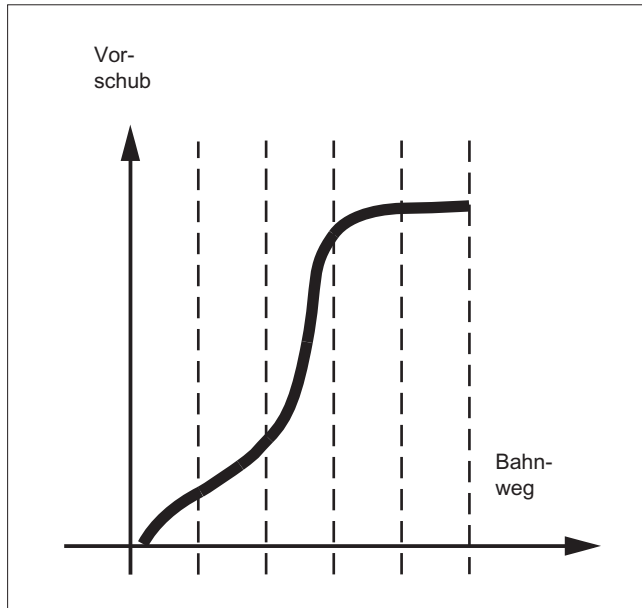
programmierten Vorschubwerte durch Splines. Die Vorschubwerte dienen hier als Stützpunkte zur Berechnung der Splineinterpolation.

Beispiel:

N50 F1400 FCUB X50

N60 F2000 X47

N70 F3800 X52



F=FPO(.....)

Der Vorschubverlauf wird über ein Polynom direkt programmiert. Die Angabe der Polynomkoeffizienten erfolgt analog zur Polynominterpolation.

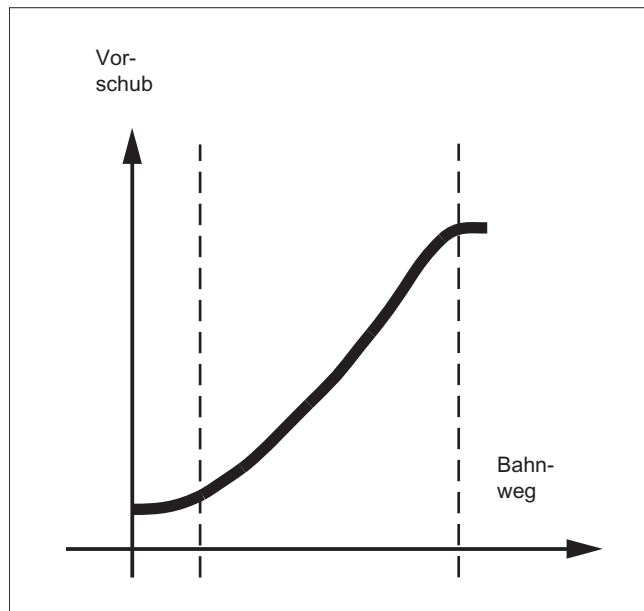
Beispiel:

F=FPO(endfeed, quadf, cubf)

endfeed, quadf und cubf sind vorher definierte Variable.

endfeed:	Vorschub am Satzende
quadf:	Quadratischer Polynomkoeffizient
cubf:	Kubischer Polynomkoeffizient

Bei aktivem FCUB schließt der Spline am Satzanfang und Satzende tangential an den über FPO festgelegten Verlauf an.



Randbedingungen

- Unabhängig vom programmierten Vorschubverlauf gelten die Funktionen zur Programmierung des Bahnfahrverhaltens.
- Der programmierbare Vorschubverlauf gilt grundsätzlich absolut - unabhängig von G90 oder G91.
- Der Vorschubverlauf FLIN und FCUB wirkt mit G93 und G94, **nicht** bei G95, G96/G961 und G97/G971.
- Bei aktivem Kompressor COMPON und Zusammenfassung mehrerer Sätze zu einem Splinesegment gilt Folgendes:

FNORM:	Für das Splinesegment gilt das F-Wort des letzten zugehörigen Satzes.
FLIN:	Für das Splinesegment gilt das F-Wort des letzten zugehörigen Satzes. Der programmierte F-Wert gilt zum Ende des Segments und wird dann linear angefahren.
FCUB:	Der erzeugte Vorschubspline weicht maximal um den im Maschinendatum MD20172 \$MC_COMPRESS_VELO_TOL definierten Wert von den programmierten Endpunkten ab.
F=FPO (... , ... , ...) :	Diese Sätze werden nicht komprimiert.

11.3 Beschleunigungsverhalten

11.3.1 Beschleunigungsmodus (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Zur Programmierung des Beschleunigungsmodus stehen folgende Teileprogrammbeefehle zur Verfügung:

- "BRISK, BRISKA"
Die Einzelachsen bzw. die Bahnachsen verfahren mit maximaler Beschleunigung bis zum Erreichen der programmierten Vorschubgeschwindigkeit (**Beschleunigung ohne Ruckbegrenzung**).
- "SOFT, SOFTA"
Die Einzelachsen bzw. die Bahnachsen verfahren mit stetiger Beschleunigung bis zum Erreichen der programmierten Vorschubgeschwindigkeit (**Beschleunigung mit Ruckbegrenzung**).
- "DRIVE, DRIVEA"
Die Einzelachsen bzw. die Bahnachsen verfahren mit maximaler Beschleunigung bis zu einer projektierten Geschwindigkeitsgrenze (MD-Einstellung!). Danach erfolgt eine Beschleunigungsreduktion (MD-Einstellung!) bis zum Erreichen der programmierten Vorschubgeschwindigkeit.

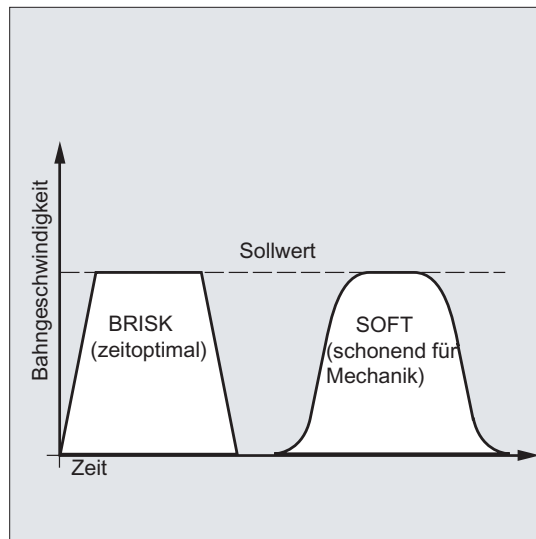


Bild 11-1 Verlauf der Bahngeschwindigkeit bei BRISK und SOFT

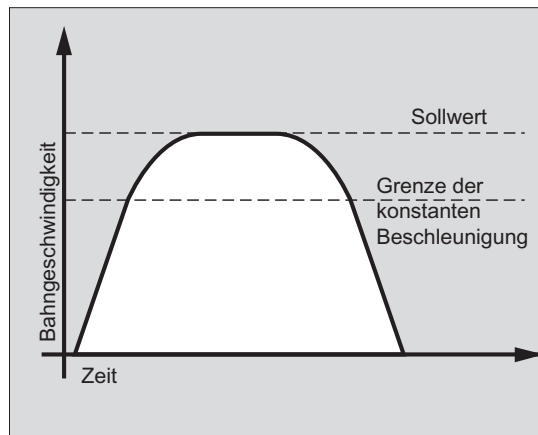


Bild 11-2 Verlauf der Bahngeschwindigkeit bei DRIVE

Syntax

```
BRISK
BRISKA (<Achse1>, <Achse2>, ...)
SOFT
SOFTA (<Achse1>, <Achse2>, ...)
DRIVE
DRIVEA (<Achse1>, <Achse2>, ...)
```

Bedeutung

BRISK:	Befehl zum Einschalten der "Beschleunigung ohne Ruckbegrenzung" für die Bahnachsen.
BRISKA:	Befehl zum Einschalten der "Beschleunigung ohne Ruckbegrenzung" für Einzelachs-bewegungen (JOG, JOG/INC, Positionierachse, Pendelachse, etc.).
SOFT:	Befehl zum Einschalten der "Beschleunigung mit Ruckbegrenzung" für die Bahnachsen.
SOFTA:	Befehl zum Einschalten der "Beschleunigung mit Ruckbegrenzung" für Einzelachs-bewegungen (JOG, JOG/INC, Positionierachse, Pendelachse, etc.).
DRIVE:	Befehl zum Einschalten der reduzierten Beschleunigung oberhalb einer projektierten Geschwindigkeitsgrenze (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT) für die Bahnachsen.
DRIVEA:	Befehl zum Einschalten der reduzierten Beschleunigung oberhalb einer projektierten Geschwindigkeitsgrenze (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT) für Einzelachs-bewegungen (JOG, JOG/INC, Positionierachse, Pendelachse, etc.).
(<Achse1>, <Achse2>, ...):	Einzelachsen, für die der aufgerufene Beschleunigungsmodus gelten soll.

Randbedingungen

Wechsel des Beschleunigungsmodus während Bearbeitung

Wenn in einem Teileprogramm der Beschleunigungsmodus während der Bearbeitung gewechselt wird (BRISK ↔ SOFT), dann erfolgt auch bei Bahnsteuerbetrieb am Übergang ein Satzwechsel mit Genauhalt am Satzende.

Beispiele

Beispiel 1: SOFT und BRISKA

Programmcode

```
N10 G1 X... Y... F900 SOFT  
N20 BRISKA (AX5, AX6)  
...
```

Beispiel 2: DRIVE und DRIVEA

Programmcode

```
N05 DRIVE  
N10 G1 X... Y... F1000  
N20 DRIVEA (AX4, AX6)  
...
```

Literatur

Funktionshandbuch Grundfunktionen; Beschleunigung (B2)

11.3.2 Beeinflussung der Beschleunigung bei Folgeachsen (VELOLIMA, ACCLIMA, JERKLIMA)

Bei Achskopplungen (Tangentiale Nachführung, Mitschleppen, Leitwertkopplung, Elektronisches Getriebe; siehe "Achskopplungen (Seite 477)") werden Folgeachsen/-spindeln abhängig von einer oder mehreren Leitachsen/-spindeln verfahren.

Die Dynamikbegrenzungen der Folgeachsen/-spindeln können mit den Funktionen VELOLIMA, ACCLIMA und JERKLIMA aus dem Teileprogramm oder aus Synchronaktionen beeinflusst werden, auch bei bereits aktiver Achskopplung.

Hinweis

Die Funktion JERKLIMA ist nicht für alle Kopplungsarten verfügbar.

Literatur:

- Funktionshandbuch Sonderfunktionen; Achskopplungen (M3)
 - Funktionshandbuch Erweiterungsfunktionen; Synchronspindel (S3)
-

Hinweis

Verfügbarkeit bei SINUMERIK 828D

Die Funktionen VELOLIMA, ACCLIMA und JERKLIMA können bei SINUMERIK 828D nur in Verbindung mit der Funktion "Mitschleppen" verwendet werden!

Syntax

VELOLIMA (<Achse>)=<Wert>
 ACCLIMA (<Achse>)=<Wert>
 JERKLIMA (<Achse>)=<Wert>

Bedeutung

VELOLIMA:	Befehl zur Korrektur der parametrisierten Maximal geschwindigkeit
ACCLIMA:	Befehl zur Korrektur der parametrisierten Maximal beschleunigung
JERKLIMA:	Befehl zur Korrektur des parametrisierten Maximal rucks
<Achse>:	Folgeachse, deren Dynamikbegrenzungen korrigiert werden sollen
<Wert>:	Prozentualer Korrekturwert

Beispiele

Beispiel 1: Korrektur der Dynamikbegrenzungen für eine Folgeachse (AX4)

Programmcode	Kommentar
...	
VELOLIMA[AX4]=75	; Begrenzungskorrektur auf 75% der im Maschinendatum hinterlegten axialen Maximalgeschwindigkeit.
ACCLIMA[AX4]=50	; Begrenzungskorrektur auf 50% der im Maschinendatum hinterlegten axialen Maximalbeschleunigung.
JERKLIMA[AX4]=50	; Begrenzungskorrektur auf 50% des im Maschinendatum hinterlegten axialen Maximalrucks bei Bahnbewegung.
...	

Beispiel 2: Elektronisches Getriebe

Achse 4 wird über eine Kopplung "Elektronisches Getriebe" an Achse X gekoppelt. Das Beschleunigungsvermögen der Folgeachse wird auf 70 % der maximalen Beschleunigung begrenzt. Die maximal zulässige Geschwindigkeit wird auf 50 % der maximalen Geschwindigkeit begrenzt. Nach erfolgter Einschaltung der Kopplung wird die maximal zulässige Geschwindigkeit wieder auf 100 % gesetzt.

Programmcode	Kommentar
...	
N120 ACCLIMA[AX4]=70	; Reduzierte maximale Beschleunigung.
N130 VELOLIMA[AX4]=50	; Reduzierte maximale Geschwindigkeit.
...	
N150 EGON(AX4,"FINE",X,1,2)	; Einschalten der EG-Kopplung.

11.3 Beschleunigungsverhalten

Programmcode	Kommentar
...	
N200 VELOLIMA[AX4]=100	; Volle Maximalgeschwindigkeit.
...	

Beispiel 3: Leitwertkopplung per statische Synchronaktion beeinflussen

Achse 4 wird mittels Leitwertkopplung an X gekoppelt. Das Beschleunigungsverhalten wird per statische Synchronaktion 2 ab Position 100 auf 80 % begrenzt.

Programmcode	Kommentar
...	
N120 IDS=2 WHENEVER \$AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	; Synchronaktion
N130 LEADON(AX4, X, 2)	; Leitwertkopplung ein
...	

11.3.3 Aktivierung von Technologie-spezifischen Dynamikwerten (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)

Mittels der G-Gruppe "Technologie" kann für 5 unterschiedliche technologische Bearbeitungsschritte die dazu passende Dynamik aktiviert werden.

Dynamikwerte und G-Befehle sind projektierbar und damit von Maschineneinstellungen abhängig (→ Maschinenhersteller!).

Literatur:

Funktionshandbuch Grundfunktionen; Bahnsteuerbetrieb, Genauhalt, LookAhead (B1)

Syntax

Dynamikwerte aktivieren:

DYNNORM
 DYNPOS
 DYNROUGH
 DYNSEMIFIN
 DYNFINISH

Hinweis

Die Dynamikwerte werden bereits in dem Satz wirksam, in dem der zugehörige G-Befehl programmiert wird. Es folgt kein Bearbeitungsstopp.

Bestimmtes Feldelement lesen oder schreiben:

R<m>=\$MA... [n, X]
 \$MA... [n, X]=<Wert>

Bedeutung

DYNNORM:	G-Befehl zur Aktivierung der normalen Dynamik	
DYNPOS:	G-Befehl zur Aktivierung der Dynamik für Positionierbetrieb, Gewindebohren	
DYNROUGH:	G-Befehl zur Aktivierung der Dynamik für Schruppen	
DYNSEMIFIN:	G-Befehl zur Aktivierung der Dynamik für Schichten	
DYNFINISH:	G-Befehl zur Aktivierung der Dynamik für Feinschichten	
R<m>:	Rechenparameter mit Nummer <m>	
\$MA... [n, X]:	Maschinendatum mit dynamikbestimmendem Feldelement	
<n>:	Feldindex	
	Wertebereich:	0 ... 4
	0	Normale Dynamik (DYNNORM)
	1	Dynamik für Positionierbetrieb (DYNPOS)
	2	Dynamik für Schruppen (DYNROUGH)
	3	Dynamik für Schichten (DYNSEMIFIN)
4	Dynamik für Feinschichten (DYNFINISH)	
<X>:	Achsadresse	
<Wert>:	Dynamikwert	

Beispiele

Beispiel 1: Dynamikwerte aktivieren

Programmcode	Kommentar
DYNNORM G1 X10	; Grundstellung
DYNPOS G1 X10 Y20 Z30 F...	; Positionierbetrieb, Gewindebohren
DYNROUGH G1 X10 Y20 Z30 F10000	; Schruppen
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; Schichten
DYNFINISH G1 X10 Y20 Z30 F1000	; Feinschichten

Beispiel 2: Bestimmtes Feldelement lesen oder schreiben

Maximale Beschleunigung für das Schruppen, Achse X.

Programmcode	Kommentar
R1=\$MA_MAX_AX_ACCEL[2, X]	; Lesen
\$MA_MAX_AX_ACCEL[2, X]=5	; Schreiben

11.4 Fahren mit Vorsteuerung (FFWON, FFWOF)

Durch die Vorsteuerung wird der geschwindigkeitsabhängige Nachlaufweg beim Bahnfahren gegen Null reduziert. Fahren mit Vorsteuerung ermöglicht höhere Bahngenaugkeit und damit bessere Fertigungsergebnisse.

11.5 Programmierbare Konturgenauigkeit (CPRECON, CPRECOF)

Syntax

FFWON

FFWOF

Bedeutung

FFWON:	Befehl zum Einschalten der Vorsteuerung
FFWOF:	Befehl zum Ausschalten der Vorsteuerung

Hinweis

Über Maschinendaten wird die Art der Vorsteuerung festgelegt und welche Bahnachsen vorgesteuert verfahren werden sollen.

Standard: Geschwindigkeitsabhängige Vorsteuerung

Option: Beschleunigungsabhängige Vorsteuerung

Beispiel

Programmcode
N10 FFWON
N20 G1 X... Y... F900 SOFT

11.5 Programmierbare Konturgenauigkeit (CPRECON, CPRECOF)

Die Funktion "Programmierbare Konturgenauigkeit" reduziert den Bahnfehler an gekrümmten Konturen durch automatische Anpassung der Geschwindigkeit.

Die einzuhaltende Konturgenauigkeit wird abhängig von der Projektierung der Maschine (MD20470 \$MC_MC_CPREC_WITH_FFW; siehe Angaben des Maschinenherstellers) entweder über das Settingdatum \$SC_CONTPREC oder über die programmierte Konturtoleranz CTOL vorgegeben. Je kleiner der Wert und je kleiner der K_v -Faktor der Geometrieachsen, umso stärker wird der Bahnvorschub auf gekrümmten Konturen abgesenkt.

Die Funktion "Programmierbare Konturgenauigkeit" wird über die Anweisungen CPRECON und CPRECOF im NC-Programm ein- bzw. ausgeschaltet.

Syntax

CPRECON

...

CPRECOF

Bedeutung

CPRECON:	G-Funktionsaufruf: "Programmierbare Konturgenauigkeit" einschalten	
	Wirksamkeit:	modal
CPRECOF:	G-Funktionsaufruf: "Programmierbare Konturgenauigkeit" ausschalten	
	Wirksamkeit:	modal

CPRECON und CPRECOF bilden zusammen die G-Funktionsgruppe 39 (Programmierbare Konturgenauigkeit).

Hinweis

Über das Settingdatum \$SC_MINFEED (Mindestbahnvorschub bei CPRECON) kann der Anwender eine Mindestgeschwindigkeit für den Bahnvorschub vorgeben.

Der Vorschub wird nicht unter diesen Wert begrenzt, es sei denn, ein niedrigerer F-Wert wurde programmiert oder die dynamischen Begrenzungen der Achsen erzwingen eine niedrigere Bahngeschwindigkeit.

Hinweis

Die Funktion "Programmierbare Konturgenauigkeit" betrachtet nur die Geometrieachsen der Bahn. Auf die Geschwindigkeiten von Positionierachsen hat sie keinen Einfluss.

Beispiel

Programmcode	Kommentar
N10 G0 X0 Y0	
N20 CPRECON	; Einschalten der "Programmierbaren Konturgenauigkeit".
N30 G1 G64 X100 F10000	; Bearbeitung mit 10 m/min im Bahnsteuerbetrieb.
N40 G3 Y20 J10	; Automatische Vorschubbegrenzung im Kreissatz.
N50 G1 X0	; Vorschub wieder ohne Begrenzung (10 m/min).
...	
N100 CPRECOF	; Ausschalten der "Programmierbaren Konturgenauigkeit".
N110 G0 ...	

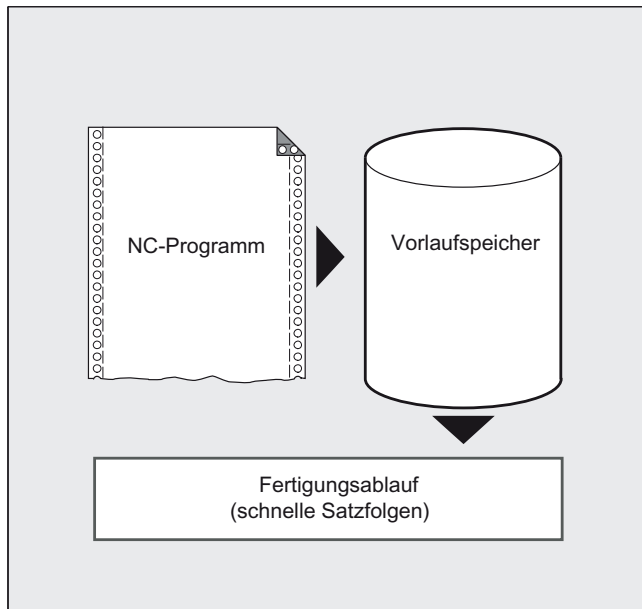
Literatur

Zur Programmierung von CTOL siehe "Programmierbare Kontur-/Orientierungstoleranz (CTOL, OTOL, ATOL) (Seite 471)"

Für ausführlichere Informationen zur Funktion "Programmierbare Konturgenauigkeit" siehe: Funktionshandbuch Sonderfunktionen; Konturtunnel-Überwachung (K6), Kapitel: "Programmierbare Konturgenauigkeit"

11.6 Programmablauf mit Vorlaufspeicher (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Je nach Ausbaustufe verfügt die Steuerung über eine bestimmte Menge sog. Vorlaufspeicher, die fertig aufbereitete Sätze vor der Abarbeitung speichern und im Fertigungsablauf als schnelle Satzfolgen ausgeben. Hierdurch lassen sich kurze Wege mit hohen Geschwindigkeiten abfahren. Soweit die Restzeit der Steuerung es zulässt, wird der Vorlaufspeicher grundsätzlich gefüllt.



Bearbeitungsabschnitt kennzeichnen

Der Bearbeitungsabschnitt, der im Vorlaufspeicher zwischengespeichert werden soll, wird im Teileprogramm am Anfang mit "STOPFIFO" und am Ende mit "STARTFIFO" gekennzeichnet. Die Abarbeitung der aufbereiteten und zwischengespeicherten Sätze beginnt erst nach dem Befehl "STARTFIFO" oder wenn der Vorlaufspeicher voll ist.

Automatische Vorlaufspeichersteuerung

Die automatische Vorlaufspeichersteuerung wird mit dem Befehl "FIFOCTRL" aufgerufen. "FIFOCTRL" wirkt zunächst genauso wie "STOPFIFO". Bei jeder Programmierung wird gewartet, bis der Vorlaufspeicher voll ist, dann beginnt die Abarbeitung. Unterschiedlich ist dagegen das Verhalten beim Leerlaufen des Vorlaufspeichers: mit "FIFOCTRL" wird ab einem Füllstand von 2/3 die Bahngeschwindigkeit zunehmend reduziert, um ein komplettes Leerlaufen und ein Abbremsen bis zum Stillstand zu verhindern.

Vorlaufstopp

Die Satzaufbereitung und -zwischenspeicherung wird angehalten, wenn im Satz der Befehl "STOPRE" programmiert ist. Der nachfolgende Satz wird erst dann ausgeführt, wenn alle

 11.6 Programmablauf mit Vorlaufspeicher (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

vorher aufbereiteten und gespeicherten Sätze vollständig abgearbeitet sind. Der vorherige Satz wird im Genauhalt angehalten (wie G9).

ACHTUNG

Programmabbruch

Bei eingeschalteter Werkzeugkorrektur und bei Spline-Interpolationen sollte kein "STOPRE" programmiert werden, da sonst zusammengehörige Satzfolgen unterbrochen werden.
--

Syntax

Tabelle 11-1 Bearbeitungsabschnitt kennzeichnen:

STOPFIFO
...
STARTFIFO

Tabelle 11-2 Automatische Vorlaufspeichersteuerung:

...
FIFOCTRL
...

Tabelle 11-3 Vorlaufstopp:

...
STOPRE
...

Hinweis

Die Befehle "STOPFIFO", "STARTFIFO", "FIFOCTRL" und "STOPRE" müssen im eigenen Satz programmiert werden.

Bedeutung

STOPFIFO:	"STOPFIFO" kennzeichnet den Beginn eines Bearbeitungsabschnitts, der im Vorlaufspeicher zwischengespeichert werden soll. Mit "STOPFIFO" wird die Bearbeitung angehalten und der Vorlaufspeicher gefüllt, bis: <ul style="list-style-type: none"> • "STARTFIFO" oder "STOPRE" erkannt wird oder • der Vorlaufspeicher voll ist oder • das Programmende erreicht ist.
STARTFIFO:	Mit "STARTFIFO" startet die schnelle Abarbeitung des Bearbeitungsabschnitts, parallel dazu erfolgt das Auffüllen des Vorlaufspeichers
FIFOCTRL:	Einschalten der automatischen Vorlaufspeichersteuerung
STOPRE:	Vorlauf stoppen

Hinweis

Das Auffüllen des Vorlaufspeichers wird nicht ausgeführt bzw. unterbrochen, wenn der Bearbeitungsabschnitt Befehle enthält, die einen ungepufferten Betrieb erzwingen (Referenzpunktfahren, Messfunktionen, ...).

Hinweis

Beim Zugriff auf Zustandsdaten der Maschine (\$SA...) erzeugt die Steuerung internen Vorlaufstopp.

Beispiel: Vorlauf stoppen

Programmcode	Kommentar
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; Messsatz mit Messtaster des ersten Messeingangs und Geradeninterpolation.
N40 STOPRE	; Vorlaufstopp.
...	

11.7 Beeinflussung von Stopp-Ereignissen durch Stop-Delay-Bereiche (DELAYFSTON, DELAYFSTOF)

Zur Definition eines Stopp-Delay-Bereichs im Teileprogramm dienen die vordefinierten Prozeduren DELAYFSTON und DELAYFSTOF.

Hinweis

In Synchronaktionen sind DELAYFSTON und DELAYFSTOF **nicht** zulässig!

Syntax

```
DELAYFSTON
...
DELAYFSTOF
```

Hinweis

DELAYFSTON und DELAYFSTOF stehen allein in einer Teileprogrammzeile!

Bedeutung

DELAYFSTON:	Beginn eines Bereichs definieren, in dem "sanfte" Stopps verzögert werden, bis das Ende des Stop-Delay-Bereichs erreicht wird
DELAYFSTOF:	Ende eines Stop-Delay-Bereichs definieren

Systemvariablen

Ob sich die Teileprogrammbearbeitung zum aktuellen Zeitpunkt in einem Stop-Delay-Bereich befindet, kann mittels folgender Systemvariablen abgefragt werden:

- im Teileprogramm mit \$P_DELAYFST
- in Synchronaktionen mit \$AC_DELAYFST

Wert	Bedeutung
0	Delay-Stop-Bereich nicht aktiv
1	Delay-Stop-Bereich aktiv

Randbedingungen

Unterprogrammende

Mit dem Ende des Unterprogramms, in dem DELAYFSTON gerufen wurde, wird implizit DELAYFSTOF aktiviert.

Schachtelung

Ruft Unterprogramm 1 in einem Stop-Delay-Bereich Unterprogramm 2, so ist Unterprogramm 2 komplett ein Stop-Delay-Bereich. Insbesondere ist DELAYFSTOF in Unterprogramm 2 wirkungslos.

Beispiel

Der folgende Programmausschnitt zeigt ein Beispiel für die Verschachtelung von Stop-Delay-Bereichen in zwei Programmebenen:

Programmcode	Kommentar
N10010 DELAYFSTON()	; Sätze mit N10xxx Programmebene 1.

11.8 Programmstelle für SERUPRO verhindern (IPTRLOCK, IPTRUNLOCK)

Programmcode	Kommentar
N10020 R1 = R1 + 1	
N10030 G4 F1	; Stop-Delay-Bereich beginnt.
...	
N10040 Unterprogramm2	
...	
...	; Interpretation des Unterprogramms 2.
N20010 DELAYFSTON()	; Unwirksam, wiederholter Beginn, 2. Ebene.
...	
N20020 DELAYFSTOF()	; Unwirksam, Ende in anderer Ebene.
N20030 RET	
N10050 DELAYFSTOF()	; Stop-Delay-Bereichs-Ende in gleicher Ebene.
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; Stop-Delay-Bereich endet. Stopps wirken ab jetzt unmittelbar.

11.8 Programmstelle für SERUPRO verhindern (IPTRLOCK, IPTRUNLOCK)

Für bestimmte komplizierte mechanische Situationen an der Maschine ist es erforderlich, den Satzsuchlauf SERUPRO zu verhindern.

Mit einem programmierbaren Unterbrechungszeiger besteht eine Eingriffsmöglichkeit, beim "Suchen auf der Unterbrechungsstelle", vor der suchunfähigen Stelle aufzusetzen.

Es können auch suchunfähige Bereiche in Teileprogrammgebieten definiert werden, in denen die NCK noch nicht wieder einsteigen kann. Mit dem Programmabbruch vermerkt der NCK den zuletzt verarbeiteten Satz, auf den über die Bedienoberfläche HMI gesucht werden kann.

Syntax

```
IPTRLOCK
IPTRUNLOCK
```

Die Befehle stehen allein in einer Teleprogrammzeile und ermöglichen einen programmierbaren Unterbrechungszeiger

Bedeutung

IPTRLOCK:	Beginn des suchunfähigen Programmabschnitts
IPTRUNLOCK:	Ende des suchunfähigen Programmabschnitts

Beide Befehle sind nur in Teileprogrammen, **nicht** jedoch in Synchronaktionen zulässig.

Beispiel

Verschachtelung suchunfähiger Programmabschnitte in zwei Programmebenen mit impliziten "IPTRUNLOCK". Das implizite "IPTRUNLOCK" in Unterprogramm 1 beendet den suchunfähigen Bereich.

Programmcode	Kommentar
N10010 IPTRLOCK()	
N10020 R1 = R1 + 1	
N10030 G4 F1	; Haltesatz, der suchunfähige Programmabschnitt beginnt.
...	
N10040 Unterprogramm2	
...	; Interpretation des Unterprogramms 2.
N20010 IPTRLOCK ()	; Unwirksam, wiederholter Beginn.
...	
N20020 IPTRUNLOCK ()	; Unwirksam, Ende in anderer Ebene.
N20030 RET	
...	
N10060 R2 = R2 + 2	
N10070 RET	; Ende des suchunfähigen Programmabschnitts.
N100 G4 F2	; Hauptprogramm wird fortgesetzt.

Eine Unterbrechung auf 100 liefert dann wieder der Unterbrechungszeiger.

Weitere Informationen**Suchunfähige Bereiche erfassen und suchen**

Die suchunfähigen Programmabschnitte werden mit dem Sprachbefehlen "IPTRLOCK" und "IPTRUNLOCK" gekennzeichnet.

Der Befehl "IPTRLOCK" friert den Unterbrechungszeiger auf ein im Hauptlauf ausführbaren Einzelsatz (SBL1) ein. Dieser Satz wird im Folgenden als Haltesatz bezeichnet. Tritt nach "IPTRLOCK" ein Programmabbruch ein, so kann auf der Bedienoberfläche HMI nach diesen sogenannten Haltesatz gesucht werden.

Auf den aktuellen Satz wieder aufsetzen

Der Unterbrechungszeiger wird mit "IPTRUNLOCK" für den nachfolgenden Programmabschnitt auf den aktuellen Satz zum Unterbrechungspunkt gesetzt werden.

Nach einem gefundenen Suchziel kann mit dem selben Haltesatz ein neues Suchziel wiederholt werden.

Ein vom Benutzer editierter Unterbrechungszeiger, muss über HMI wieder entfernt werden.

Regeln bei Schachtelung

Folgende Punkte regeln das Zusammenspiel der Sprachbefehle "IPTRLOCK" und "IPTRUNLOCK" mit Verschachtelungen und dem Unterprogrammende:

1. Mit dem Ende des Unterprogramms, in dem "IPTRLOCK" gerufen wurde, wird implizit "IPTRUNLOCK" aktiviert.
2. "IPTRLOCK" in einem suchunfähigen Bereich bleibt ohne Wirkung.
3. Ruft Unterprogramm1 in einem suchunfähigen Bereich Unterprogramm2, so bleibt Unterprogramm2 komplett suchunfähig. Insbesondere ist "IPTRUNLOCK" in Unterprogramm2 wirkungslos.

Weitere Informationen hierzu siehe

/FB1/ Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1).

Systemvariable

Ein suchunfähiger Bereich kann mit "\$P_IPTRLOCK" im Teileprogramm erkannt werden.

Automatischer Unterbrechungszeiger

Die Funktion automatischer Unterbrechungszeiger legt automatisch eine vorher festgelegte Kopplungsart als suchunfähig fest. Mittels Maschinendatum wird für

- Elektronisches Getriebe bei "EGON"
- Axiale Leitwertkopplung bei "LEADON"

der automatische Unterbrechungszeiger aktiviert. Überschneiden sich der programmierte und der über Maschinendatum aktivierbare automatische Unterbrechungszeiger, so wird der größtmögliche suchunfähige Bereich gebildet.

11.9

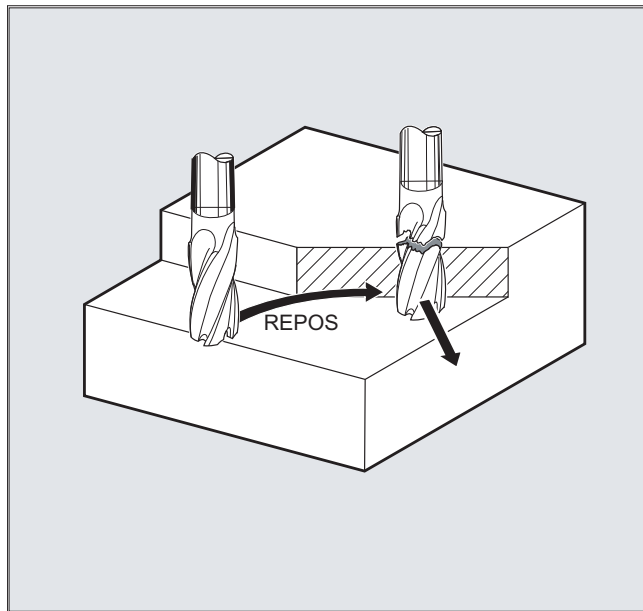
Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Wenn Sie während der Bearbeitung das laufende Programm unterbrechen und das Werkzeug freifahren – z.B. wegen Werkzeugbruchs oder zum Vermessen des Werkstücks – können Sie anschließend die Kontur an einem wählbaren Punkt programmgesteuert wieder anfahren.

Der Befehl `REPOS` wirkt in einem ASUP wie ein Unterprogramm-Rücksprung (z.B. M17).

Nachfolgende Sätze werden nicht mehr ausgeführt. Zur Unterbrechung des Programmlaufs siehe auch "Interruptroutine (ASUP) (Seite 120)".

11.9 Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)



Syntax

```

REPOSA RMIBL DISPR=...
REPOSA RMBBL
REPOSA RMEBL
REPOSA RMNBL
REPOSL RMIBL DISPR=...
REPOSL RMBBL
REPOSL RMEBL
REPOSL RMNBL
REPOSQ RMIBL DISPR=... DISR=...
REPOSQ RMBBL DISR=...
REPOSQ RMEBL DISR=...
REPOSQA DISR=...
REPOSH RMIBL DISPR=... DISR=...
REPOSH RMBBL DISR=...
REPOSH RMEBL DISR=...
REPOSHA DISR=...
    
```

Bedeutung

Auswahl des Anfahrwegs

REPOSA:	Wiederanfahren an die Kontur mit den Geometrieachsen auf einer Geraden. Alle anderen Kanalachsen werden ebenfalls repositioniert.
REPOSL:	Wiederanfahren an die Kontur mit den Geometrieachsen auf einer Geraden . Alle anderen Kanalachsen müssen explizit programmiert werden.

11.9 Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

REPOSQ DISR=... :	Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Viertelkreis mit Radius DISR. Alle anderen Kanalachsen müssen explizit programmiert werden.
REPOSQA DISR=... :	Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Viertelkreis mit Radius DISR. Alle anderen Kanalachsen werden ebenfalls repositioniert.
REPOSH DISR=... :	Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Halbkreis mit Durchmesser DISR. Alle anderen Kanalachsen müssen explizit programmiert werden.
REPOSHA DISR=... :	Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Halbkreis mit Radius DISR. Alle anderen Kanalachsen werden ebenfalls repositioniert.

Auswahl des Wiederanfahrpunkts

RMIBL:	Unterbrechungspunkt anfahren
RMIBL DISPR=...:	Eintrittspunkt im Abstand DISPR in mm/inch vor Unterbrechungspunkt
RMBBL:	Satzanfangspunkt anfahren
RMEBL:	Satzendpunkt anfahren
RMEBL DISPR=... :	Satzendpunkt anfahren im Abstand DISPR vor Endpunkt
RMNBL:	An den nächstliegenden Bahnpunkt anfahren
A0 B0 C0 :	Achsen, in denen angefahren werden soll

Hinweis

Kompatibilität

Um kompatibel mit älteren Softwareständen zu bleiben, können Sie den REPOS-Anfahrmodus auch weiterhin über die modalen Befehle RMI, RMB, RME und RMN programmieren. Bei Verwendung innerhalb eines ASUP sollte dieses mit dem Attribut SAVE in der PROC-Anweisung versehen werden. Andernfalls wirkt der im ASUP verwendete modale REPOS-Anfahrmodus, wenn er von der Voreinstellung RMI abweicht, auch bei nachfolgenden REPOS-Vorgänge.

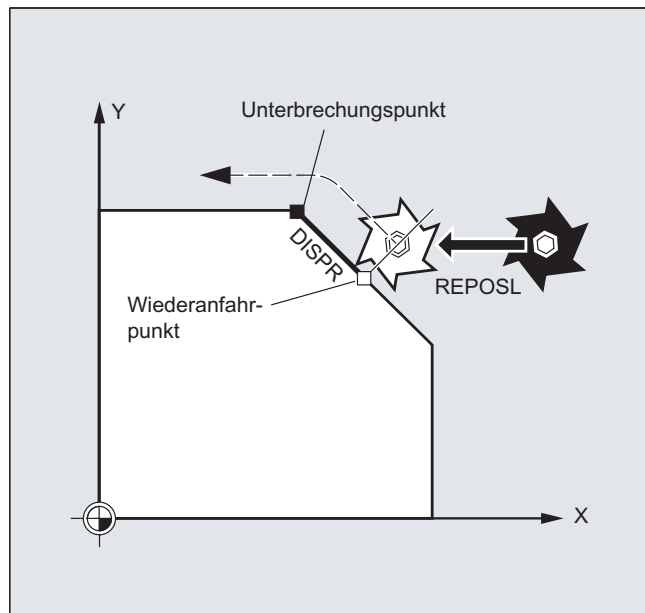
Wiederanfahren an die Kontur auf einer Geraden, REPOSA, REPOSL

Das Werkzeug fährt den Wiederanfahrpunkt direkt auf einer Geraden an.

Beispiel

REPOSL RMIBL DISPR=6 F400

11.9 Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

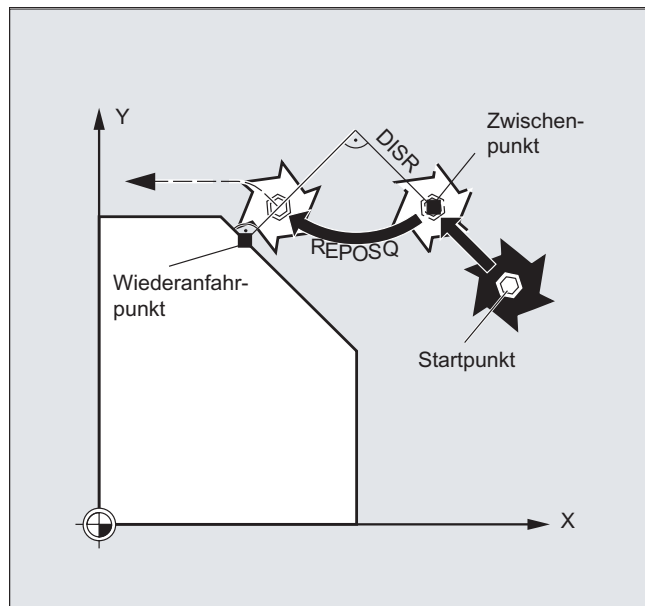


Wiederanfahren an die Kontur mit Viertelkreis, REPOSQ, REPOSQA

Das Werkzeug fährt den Wiederanfahrpunkt auf einem Viertelkreis mit Radius $DISR = \dots$ an. Den notwendigen Zwischenpunkt zwischen Start- und Wiederanfahrpunkt berechnet die Steuerung automatisch.

Beispiel

REPOSQ RMIBL DISR=10 F400

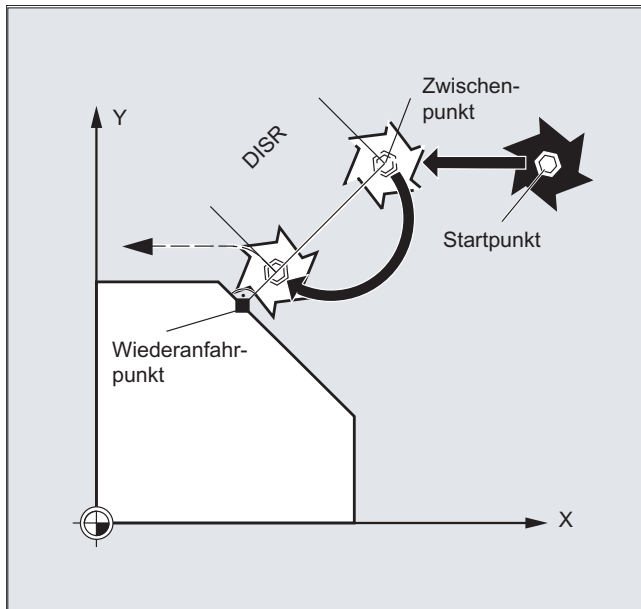


Wiederanfahren an die Kontur mit Halbkreis, REPOSH, REPOSHA

Das Werkzeug fährt den Wiederanfahrpunkt auf einem Halbkreis mit Durchmesser $DISR = \dots$ an. Den notwendigen Zwischenpunkt zwischen Start- und Wiederanfahrpunkt berechnet die Steuerung automatisch.

Beispiel

REPOSH RMIBL DISR=20 F400

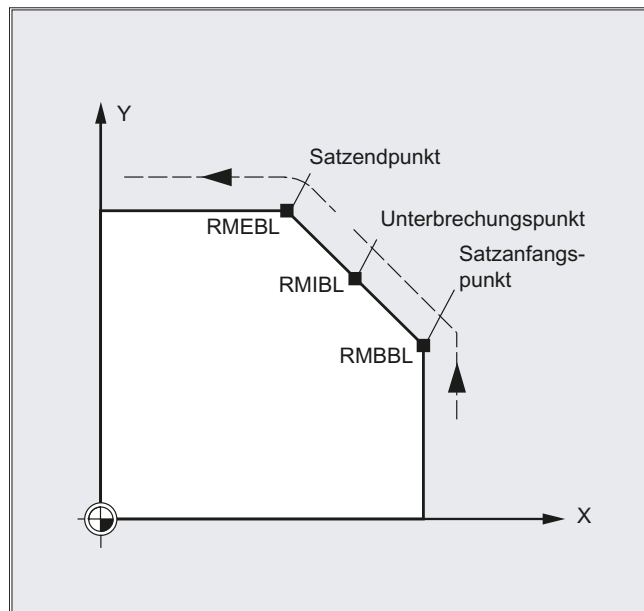


Wiederanfahrpunkt festlegen (nicht für SERUPRO Anfahren mit RMNBL)

Bezogen auf den NC-Satz, in dem der Programm-Ablauf unterbrochen wurde, können Sie zwischen drei Wiederanfahrpunkten wählen:

- RMIBL, Unterbrechungspunkt
- RMBBL, Satzanfangspunkt bzw. letzter Endpunkt
- RMEBL, Satzendpunkt

11.9 Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISPR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)



Mit RMIBL DISPR=... bzw. mit RMEBL DISPR=... können Sie einen Wiederanfahrpunkt festlegen, der vor dem Unterbrechungspunkt bzw. vor dem Satzsendepunkt liegt.

Mit DISPR=... beschreiben Sie den Konturweg in mm/inch, um den der Wiederanfahrpunkt vor dem Unterbrechungs- bzw. Endpunkt liegt. Dieser Punkt kann - auch für größere Werte - maximal im Satzanfangspunkt liegen.

Wird kein DISPR=... programmiert, gilt DISPR=0 und damit der Unterbrechungspunkt (bei RMIBL) bzw. der Satzsendepunkt (bei RMEBL).

Vorzeichen von DISPR

Das Vorzeichen von DISPR wird ausgewertet. Bei positivem Vorzeichen ist das Verhalten wie bisher.

Bei negativem Vorzeichen wird hinter dem Unterbrechungspunkt bzw. bei RMBBL hinter dem Startpunkt wieder aufgesetzt.

Der Abstand Unterbrechungspunkt-Aufsetzpunkt ergibt sich aus dem Betrag von DISPR. Dieser Punkt kann auch für betragsmäßig größere Werte maximal im Satzsendepunkt liegen.

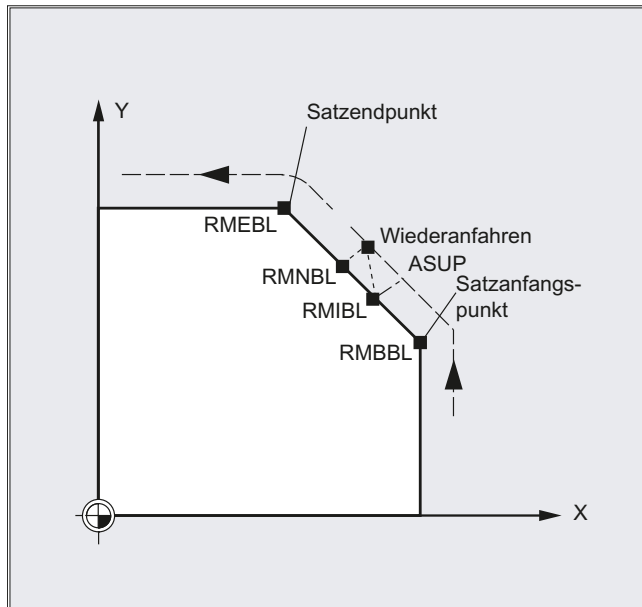
Anwendungsbeispiel:

Durch einen Sensor wird die Annäherung an eine Spannpratze erkannt. Es wird ein ASUP ausgelöst, mit dem die Spannpratze umfahren wird.

Anschließend wird mit negativem DISPR auf einen Punkt hinter der Spannpratze repositioniert und das Programm fortgesetzt.

Anfahren vom nächstliegenden Bahnpunkt RMNBL

Zum Interpretationszeitpunkt von REPOSA wird nach einer Unterbrechung der Wiederanfahrtsatz mit RMNBL nicht noch einmal komplett begonnen, sondern nur der Restweg abgearbeitet. Es wird der nächstliegende Bahnpunkt des unterbrochenen Satzes angefahren.



Status für den gültigen REPOS-Mode

Der gültige REPOS-Mode des unterbrochenen Satzes kann über Synchronaktionen mit der Variablen \$AC_REPOS_PATH_MODE gelesen werden:

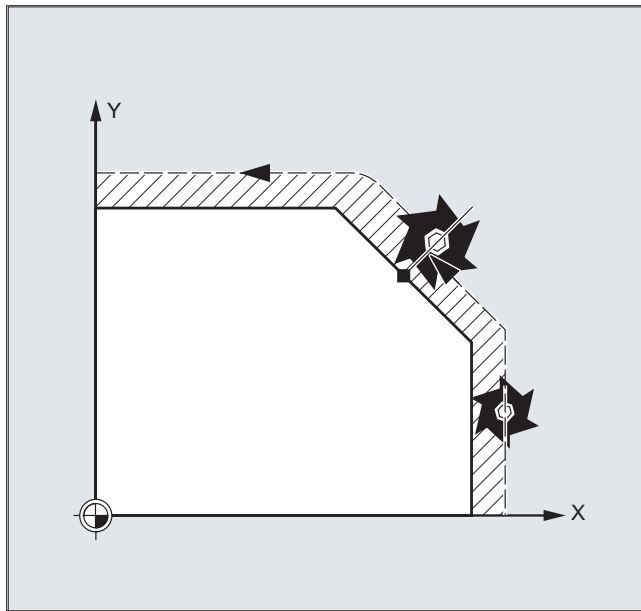
- 0 Anfahren nicht definiert
- 1 RMBBL: Anfahren auf den Beginn
- 2 RMIBL: Anfahren auf den Unterbrechungspunkt
- 3 RMEBL: Anfahren auf den Satzendpunkt
- 4 RMNBL: Anfahren auf den nächstliegenden Bahnpunkt des unterbrochenen Satzes

Anfahren mit neuem Werkzeug

Falls Sie den Programmablauf wegen Werkzeugbruch gestoppt haben:

Mit Programmierung der neuen D-Nummer wird das Programm ab Wiederanfahrpunkt mit den geänderten Werkzeugkorrekturwerten fortgesetzt.

Bei geänderten Werkzeugkorrekturwerten kann der Unterbrechungspunkt möglicherweise nicht mehr angefahren werden. In diesem Fall wird der dem Unterbrechungspunkt nächstgelegene Punkt auf der neuen Kontur angefahren (gegebenenfalls um DISPR modifiziert).



Kontur anfahren

Die Bewegung, mit der das Werkzeug wieder an die Kontur heranfährt, ist programmierbar. Die Adressen der zu verfahrenen Achsen geben Sie mit Wert Null an.

Mit den Befehlen REPOSA, REPOSQA und REPOSHA werden automatisch alle Achsen repositioniert. Es ist keine Achsangabe notwendig.

Bei Programmierung von REPOSL, REPOSQ und REPOSH fahren alle Geometrieachsen automatisch, also auch ohne Angabe im Befehl, an. Alle anderen Achsen müssen im Befehl angegeben werden.

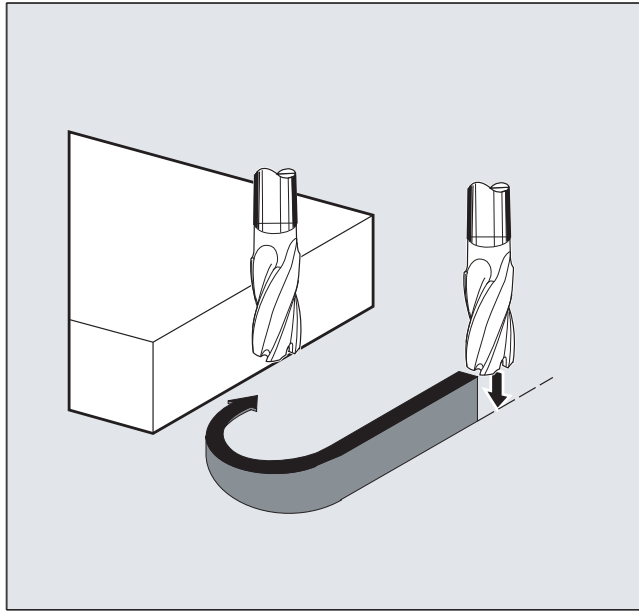
Für die Kreisbewegungen REPOSH und REPOSQ gilt:

Der Kreis wird in der angegebenen Arbeitsebene G17 bis G19 gefahren.

Falls Sie im Anfahrsatz die dritte Geometrieachse (Zustellrichtung) angeben, wird der Wiederanfahrpunkt für den Fall, dass Werkzeugposition und programmierte Position in Zustellrichtung nicht übereinstimmen, auf einer Schraubenlinie angefahren.

In folgenden Fällen wird automatisch auf lineares Anfahren REPOS L umgeschaltet:

- Sie haben keinen Wert für DISR angegeben.
- Es gibt keine definierte Anfahrri chtung (Programmunterbrechung in einem Satz ohne Verfahrinformation).
- Bei Anfahrri chtung senkrecht zur aktuellen Arbeitsebene.



11.10 Beeinflussung der Bewegungsführung

11.10.1 Prozentuale Ruckkorrektur (JERKLIM)

Mit dem NC-Befehl "JERKLIM" kann der per Maschinendatum eingestellte maximal mögliche Ruck einer Achse bei Bahnbewegung in kritischen Programmabschnitten reduziert oder überhöht werden.

Voraussetzung

Der Beschleunigungsmodus SOFT muss aktiv sein.

Wirksamkeit

Die Funktion wirkt:

- in den AUTOMATIK-Betriebsarten.
- nur auf Bahnachsen.

Syntax

JERKLIM[<Achse>]=<Wert>

Bedeutung

JERKLIM:	Befehl zur Ruckkorrektur	
<Achse>:	Maschinenachse, deren Ruckgrenzwert angepasst werden soll.	
<Wert>:	Prozentualer Korrekturwert, bezogen auf den projizierten maximalen Achsruck bei Bahnbewegung (MD32431 \$MA_MAX_AX_JERK).	
	Wertebereich:	1 ... 200
	Der Wert 100 bewirkt keine Beeinflussung des Rucks.	

Hinweis

Das Verhalten von JERKLIM bei Teileprogrammende und Kanal-Reset wird projiziert mit Bit 0 im Maschinendatum MD32320 \$MA_DYN_LIMIT_RESET_MASK:

- Bit 0 = 0:
Der programmierte Wert für JERKLIM wird mit Kanal-Reset/M30 auf 100 % zurückgesetzt.
- Bit 0 = 1:
Der programmierte Wert für JERKLIM bleibt über Kanal-Reset/M30 hinaus erhalten.

Beispiel

Programmcode	Kommentar
...	
N60 JERKLIM[X]=75	; Der Achsschlitten in X-Richtung soll nur mit maximal 75% des für die Achse zulässigen Rucks beschleunigt/verzögert werden.
...	

11.10.2 Prozentuale Geschwindigkeitskorrektur (VELOLIM)

Mit dem Befehl VELOLIM kann im Teileprogramm oder Synchronaktion die über Maschinendatum eingestellte maximal mögliche Geschwindigkeit einer Achse bzw. die maximal mögliche getriebestufenabhängige Drehzahl einer Spindel reduziert werden.

Wirksamkeit

Die Funktion wirkt:

- in den AUTOMATIK-Betriebsarten.
- auf Bahn- und Positionierachsen.
- auf Spindeln im Spindel-/Achsbetrieb

Syntax

VELOLIM[<Achse/Spindel>]=<Wert>

Bedeutung

VELOLIM:	Befehl zur Geschwindigkeitskorrektur
<Achse/Spindel>:	<p>Achse oder Spindel, deren Geschwindigkeits- oder Drehzahlgrenzwert angepasst werden soll.</p> <p>VELOLIM für Spindeln</p> <p>Über Maschinendatum (MD30455 \$MA_MISC_FUNCTION_MASK, Bit 6) kann für die Programmierung im Teileprogramm eingestellt werden, ob "VELOLIM" unabhängig von der aktuellen Verwendung als Spindel oder Achse wirkt (Bit 6 = 1) oder getrennt für jede Betriebsart programmierbar sein soll (Bit 6 = 0). Ist eine getrennte Wirkung projektiert, dann wird die Auswahl über den Bezeichner bei der Programmierung getroffen:</p> <ul style="list-style-type: none"> • Spindelbezeichner S<n> für Spindelbetriebsarten • Achsbezeichner, z. B. "c", für den Achsbetrieb
<Wert>:	<p>Prozentualer Korrekturwert</p> <p>Der Korrekturwert bezieht sich:</p> <ul style="list-style-type: none"> • bei Achsen / Spindeln im Achsbetrieb (MD30455, Bit 6 == 0): auf die projektierte maximale Achsgeschwindigkeit (MD32000 \$MA_MAX_AX_VELO). • bei Spindeln im Spindel- oder Achsbetrieb (MD30455 Bit 6 == 1): auf die Maximaldrehzahl der aktiven Getriebestufe (MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])
Wertebereich:	1 ... 100
Der Wert 100 bewirkt keine Beeinflussung der Geschwindigkeit bzw. Drehzahl.	

Hinweis

Verhalten bei Teileprogrammende und Kanal-Reset

Das Verhalten von "VELOLIM" bei Teileprogrammende und Kanal-Reset ist einstellbar über das Maschinendatum: MD32320 \$MA_DYN_LIMIT_RESET_MASK, Bit 0

Erkennen einer aktiven Drehzahlbegrenzung im Spindelbetrieb

Eine Drehzahlbegrenzung durch "VELOLIM" (kleiner 100 %) kann im Spindelbetrieb über folgende Systemvariable erkannt werden:

- \$AC_SMAXVELO (maximal mögliche Spindeldrehzahl)
- \$AC_SMAXVELO_INFO (Kennung für die drehzahlbegrenzende Ursache)

Beispiele

Beispiel 1: Geschwindigkeitsbegrenzung Maschinenachse

Programmcode	Kommentar
...	
N70 VELOLIM[X]=80	; Der Achsschlitten in X-Richtung soll nur mit maximal 80 % der für die Achse zulässigen Geschwindigkeit verfahren werden.
...	

Beispiel 2: Drehzahlbegrenzung Spindel

Programmcode	Kommentar
N05 VELOLIM[S1]=90	; Begrenzung der Maximaldrehzahl von Spindel 1 auf 90 % von 1000 U/min.
...	
N50 VELOLIM[C]=45	; Begrenzung der Drehzahl auf 45 % von 1000 U/min, C sei der Achsbezeichner von S1.
...	

Maschinendateneinstellungen für Spindel 1 (AX5)

- Maximaldrehzahl der Getriebstufe 1 = 1000 U/min:
MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- Die Programmierung von "VELOLIM" wirkt gemeinsam für Spindel- und Achsbetrieb unabhängig vom programmierten Bezeichner:
MD30455 \$MA_MISC_FUNCTION_MASK[AX5], Bit 6 = 1

11.10.3 Programmbeispiel für JERKLIM und VELOLIM

Das folgende Programm stellt ein Anwendungsbeispiel für die prozentuale Ruck- und Geschwindigkeitsbegrenzung dar:

Programmcode	Kommentar
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20 ACC[Y]=30	
N1200 G1 Y20 VELOLIM[X]=5	; Der Achsschlitten in X-Richtung soll nur mit max. 5% der für die Achse zulässigen Geschwindigkeit verfahren werden.
JERKLIM[Y]=200	; Der Achsschlitten in Y-Richtung kann mit max. 200% des für die Achse zulässigen Rucks beschleunigt/verzögert werden.
N1300 G1 X0 JERKLIM[X]=2	; Der Achsschlitten in X-Richtung soll nur mit max. 2% des für die Achse zulässigen Rucks beschleunigt/verzögert werden.
N1400 G1 Y0	
M30	

11.11 Programmierbare Kontur-/Orientierungstoleranz (CTOL, OTOL, ATOL)

Mit den Befehlen "CTOL", "OTOL" und "ATOL" können die über Maschinen- und Settingdaten festgelegten Bearbeitungstoleranzen für die Kompressor-Funktionen (COMPON, COMPCURV, COMPCAD), die Überschleifarten G642, G643, G645, OST und die Orientierungsglättung ORISON im NC-Programm angepasst werden.

Die programmierten Werte gelten, bis sie neu programmiert oder durch Zuweisung eines negativen Werts gelöscht werden. Sie werden ferner gelöscht bei Programmende, Kanal-Reset, BAG-Reset, NCK-Reset (Warmstart) und Power On (Kaltstart). Nach dem Löschen gelten wieder die Werte aus den Maschinen- und Settingdaten.

Syntax

```
CTOL=<Wert>
OTOL=<Wert>
ATOL [<Achse>]=<Wert>
```

Bedeutung

CTOL:	Befehl zum Programmieren der Konturtoleranz		
	"CTOL" ist gültig für:		
	<ul style="list-style-type: none"> • alle Kompressor-Funktionen • alle Überschleifarten außer G641 und G644 		
	<Wert>:	Der Wert für die Konturtoleranz ist eine Längenangabe.	
	Typ:	REAL	
	Einheit:	Inch/mm (abhängig von der aktuellen Einstellung der Maßangabe)	
OTOL:	Befehl zum Programmieren der Orientierungstoleranz		
	"OTOL" ist gültig für:		
	<ul style="list-style-type: none"> • alle Kompressor-Funktionen • Orientierungsglättung ORISON • alle Überschleifarten außer G641, G644, OSD 		
	<Wert>:	Der Wert für die Orientierungstoleranz ist eine Winkelangabe.	
	Typ:	REAL	
	Einheit:	Grad	

11.11 Programmierbare Kontur-/Orientierungstoleranz (CTOL, OTOL, ATOL)

ATOL:	Befehl zum Programmieren einer achsspezifischen Toleranz		
	"ATOL" ist gültig für:		
	<ul style="list-style-type: none"> • alle Kompressor-Funktionen • Orientierungsglättung ORISON • alle Überschleifarten außer G641, G644, OSD 		
	<Achse>:	Name der Achse, für die eine Achstoleranz programmiert werden soll	
	<Wert>:	Der Wert für die Achstoleranz ist je nach Achstyp (Linear- oder Rundachse) eine Längen- oder Winkelangabe.	
Typ:	REAL		
Einheit:	für Linearachsen:	Inch/mm (abhängig von der aktuellen Einstellung der Maßangabe)	
	für Rundachsen:	Grad	

Hinweis

"CTOL" und "OTOL" haben Vorrang vor "ATOL".

Randbedingungen

Skalierende Frames

Skalierende Frames wirken auf die programmierten Toleranzen in gleicher Weise wie auf die Achspositionen, d. h. die relative Toleranz bleibt gleich.

Beispiel

Programmcode	Kommentar
COMPCAD G645 G1 F10000	; Kompressor-Funktion COMPCAD aktivieren.
X... Y... Z...	; Hier wirken die Maschinen- und Settingdaten.
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; Ab hier wirkt eine Konturtoleranz von 0,02 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; Ab hier wirkt eine Konturtoleranz von 0,005 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; Ab hier wirken wieder Maschinen- und Settingdaten.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Weitere Informationen

Toleranzwerte lesen

Für weitergehende Anwendungsfälle oder zur Diagnose sind die aktuell gültigen Toleranzen für die Kompressor-Funktionen (COMPON, COMPCURV, COMPCAD), die Überschleifarten G642, G643, G645, OST und die Orientierungsglättung ORISON unabhängig von der Art des Zustandekommens über Systemvariablen lesbar.

- In Synchronaktionen oder mit Vorlauf-Stopp im Teileprogramm über die Systemvariablen:

\$AC_CTOL	Konturtoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war Falls keine Konturtoleranz wirksam ist, liefert \$AC_CTOL die Wurzel aus der Summe der Quadrate der Toleranzen der Geometrieachsen.
\$AC_OTOL	Orientierungstoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war Falls keine Orientierungstoleranz wirksam ist, liefert \$AC_OTOL während einer aktiven Orientierungstransformation die Wurzel aus der Summe der Quadrate der Toleranzen der Orientierungsachsen, ansonsten den Wert "-1".
\$AA_ATOL[<Achse>]	Achstoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war Falls eine Konturtoleranz aktiv ist, liefert \$AA_ATOL[<Geometrieachse>] die Konturtoleranz geteilt durch die Wurzel aus der Anzahl der Geometrieachsen. Falls eine Orientierungstoleranz und eine Orientierungstransformation aktiv sind, liefert \$AA_ATOL[<Orientierungsachse>] die Orientierungstoleranz geteilt durch die Wurzel aus der Anzahl der Orientierungsachsen.

Hinweis

Wenn keine Toleranzwerte programmiert wurden, dann sind die \$A-Variablen nicht differenziert genug, um die möglicherweise verschiedenen Toleranzen der einzelnen Funktionen zu unterscheiden, da sie ja nur einen Wert nennen können.

Solche Fälle können auftreten, wenn die Maschinen- und Settingdaten unterschiedliche Toleranzen für Kompressor-Funktionen, Überschleifen und Orientierungsglättung einstellen. Die Variablen liefern dann den größten Wert, der bei den gerade aktiven Funktionen auftritt.

Ist z. B. eine Kompressor-Funktion mit Orientierungstoleranz 0,1° und eine Orientierungsglättung ORISON mit 1° aktiv, liefert die Variable \$AC_OTOL den Wert "1". Wird die Orientierungsglättung ausgeschaltet, liest man nur noch den Wert "0.1".

- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariablen:

\$P_CTOL	Programmierte Konturtoleranz
\$P_OTOL	Programmierte Orientierungstoleranz
\$PA_ATOL	Programmierte Achstoleranz

Hinweis

Wenn keine Toleranzwerte programmiert sind, dann liefern die \$P-Variablen den Wert "-1".

11.12 Toleranz bei G0-Bewegungen (STOLF)

G0-Toleranzfaktor

G0-Bewegungen (Eilgang, Zustellbewegungen) können im Unterschied zur Werkstückbearbeitung mit größerer Toleranz verfahren werden. Dies hat den Vorteil, dass sich die Abfahrzeiten für G0-Bewegungen verkürzen.

Die Einstellung der Toleranzen bei G0-Bewegungen erfolgt durch Projektierung des G0-Toleranzfaktors (MD20560 \$MC_G0_TOLERANCE_FACTOR).

Der G0-Toleranzfaktor wird nur wirksam, wenn:

- eine der folgenden Funktionen aktiv ist:
 - Kompressorfunktionen: COMPON, COMPCURV und COMPCAD
 - Überschleiffunktionen: G642 und G645
 - Orientierungsüberschleifen: OST
 - Orientierungsglättung: ORISON
 - Glättung bei bahnrelativer Orientierung: ORIPATH
- mehrere (≥ 2) G0-Sätze aufeinanderfolgen.
Bei einem einzelnen G0-Satz wird der G0-Toleranzfaktor nicht wirksam, da **beim Übergang** von einer Nicht-G0-Bewegung zu einer G0-Bewegung (und umgekehrt) grundsätzlich die "**kleinere Toleranz**" (Werkstückbearbeitungstoleranz) wirkt!

G0-Toleranzfaktor über das Teileprogramm anpassen

Durch Programmierung von STOLF im Teileprogramm kann der projektierte G0-Toleranzfaktor (MD20560) temporär überschrieben werden. Der Wert im MD20560 wird dabei nicht verändert. Nach Reset bzw. Teileprogrammende wird der projektierte Toleranzfaktor wieder wirksam.

Syntax

STOLF=<Toleranzfaktor>

Bedeutung

STOLF:	Befehl zur Programmierung des G0-Toleranzfaktors
<Toleranzfaktor>:	G0-Toleranzfaktor Der Faktor kann sowohl größer 1 als auch kleiner 1 sein. Normalerweise werden jedoch für G0-Bewegungen größere Toleranzen einstellbar sein. Bei STOLF=1.0 (entspricht dem projektierten Standardwert) sind für G0-Bewegungen dieselben Toleranzen wirksam wie für Nicht-G0-Bewegungen.

Systemvariablen

Der im Teileprogramm bzw. im aktuellen IPO-Satz wirksame G0-Toleranzfaktor ist über Systemvariablen lesbar.

- In Synchronaktionen oder mit Vorlauf-Stopp im Teileprogramm über die Systemvariable:

\$AC_STOLF Aktiver G0-Toleranzfaktor
G0-Toleranzfaktor, der bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.

- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariable:

\$P_STOLF Programmierter G0-Toleranzfaktor

Ist im aktiven Teileprogramm kein Wert mit STOLF programmiert, dann liefern diese beiden Systemvariablen den durch MD20560 \$MC_G0_TOLERANCE_FACTOR eingestellten Wert.

Ist in einem Satz kein Eilgang (G0) aktiv, dann liefern diese Systemvariablen immer den Wert 1.

Beispiel

Programmcode	Kommentar
COMPCAD G645 G1 F10000	; Kompressor-Funktion COMPCAD
X... Y... Z...	; Hier wirken die Maschinen- und Settingdaten.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
G0 X... Y... Z...	; Hier wirkt das Maschinendatum \$MC_G0_TOLERANCE_FACTOR (z.B. =3), also eine Überschleiftoleranz von \$MC_G0_TOLERANCE_FACTOR*\$MA_COMPRESS_POS_TOL.
CTOL=0.02	
STOLF=4	
G1 X... Y... Z...	; Ab hier wirkt eine Konturtoleranz von 0,02mm.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	

11.13 Satzwechselverhalten bei aktiver Kopplung (CPBC)

Programmcode	Kommentar
X... Y... Z...	; Ab hier wirkt ein G0-Toleranzfaktor von 4, also eine Konturtoleranz von 0,08mm.

11.13 Satzwechselverhalten bei aktiver Kopplung (CPBC)

Mit dem Befehl CPBC wird das Satzwechselkriterium vorgegeben das erfüllt sein muss, damit im Teilprogramm bei aktiver Kopplung ein Satzwechsel durchgeführt wird.

Syntax

CPBC[<Folgeachse>] = <Kriterium>

Bedeutung

CPBC:	Satzwechselkriterium bei aktiver Kopplung	
<Folgeachse>:	Achsbezeichner der Folgeachse	
<Kriterium>:	Satzwechselkriterium	
	Typ:	STRING
	Wert	Bedeutung: Satzwechsel erfolgt
	"NOC"	unabhängig vom Kopplungszustand
	"IPOSTOP"	bei sollwertseitigem Synchronlauf
	"COARSE"	bei istwertseitigem Synchronlauf "Grob"
"FINE"	bei istwertseitigem Synchronlauf "Fein"	

Beispiel

Programmcode
<pre> ; Satzwechsel erfolgt bei: ; - Kopplung zu Folgeachse X2 == aktiv ; - sollwertseitiger Synchronlauf == aktiv CPBC[X2]="IPOSTOP" </pre>

Achskopplungen

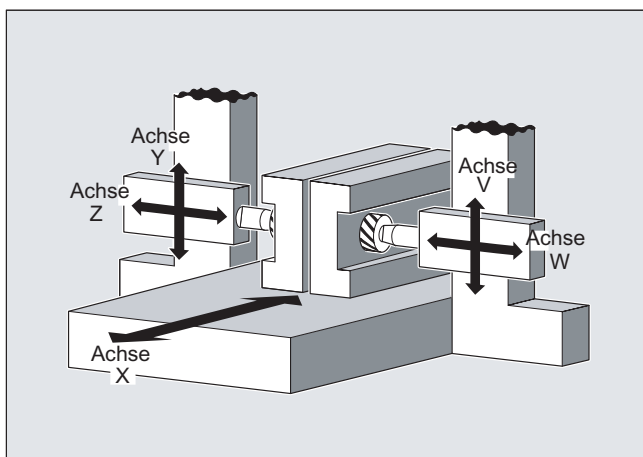
12.1 Mitschleppen (TRAILON, TRAILOF)

Beim Bewegen einer definierten Leitachse fahren ihr zugeordnete Mitschleppachsen (= Folgeachsen) unter Berücksichtigung eines Koppelfaktors die von der Leitachse abgeleiteten Verfahrenswege ab.

Leitachse und Folgeachsen bilden zusammen einen Mitschleppverband.

Anwendungsbereiche

- Verfahren einer Achse durch eine simulierte Achse. Die Leitachse ist eine simulierte Achse und die Mitschleppachse eine reale Achse. Damit kann die reale Achse mit Berücksichtigung des Koppelfaktors verfahren werden.
- Zweiseitenbearbeitung mit 2 Mitschleppverbänden:
 1. Leitachse Y, Mitschleppachse V
 2. Leitachse Z, Mitschleppachse W



Syntax

```
TRAILON (<Folgeachse>, <Leitachse>, <Koppelfaktor>)
TRAILOF (<Folgeachse>, <Leitachse>, <Leitachse 2>)
TRAILOF (<Folgeachse>)
```

Bedeutung

TRAILON:	Befehl zum Einschalten und Definieren eines Mitschleppverbandes	
	Wirksamkeit:	modal

12.1 Mitschleppen (TRAILON, TRAILOF)

<Folgeachse>:	Parameter 1: Achsbezeichnung der Mitschleppachse Hinweis: Eine Mitschleppachse kann auch Leitachse für weitere Mitschleppachsen sein. Auf diese Weise können unterschiedliche Mitschleppverbände aufgebaut werden.
<Leitachse>:	Parameter 2: Achsbezeichnung der Leitachse
<Koppelfaktor>:	Parameter 3: Koppelfaktor Der Koppelfaktor gibt das gewünschte Verhältnis der Wege von Mitschleppachse und Leitachse an: $\text{<Koppelfaktor>} = \text{Weg der Mitschleppachse} / \text{Weg der Leitachse}$
	Typ: REAL
	Voreinstellung: 1
	Die Eingabe eines negativen Wertes bewirkt eine entgegengesetzte Verfahrbewegung der Leit- und Mitschleppachse. Wird der Koppelfaktor bei der Programmierung nicht angegeben, so gilt automatisch der Koppelfaktor 1.
TRAILOF:	Befehl zum Ausschalten eines Mitschleppverbandes
	Wirksamkeit: modal
	TRAILOF mit 2 Parametern schaltet nur die Kopplung zur angegebenen Leitachse aus: <code>TRAILOF(<Folgeachse>, <Leitachse>)</code> Besitzt eine Mitschleppachse 2 Leitachsen, kann zum Ausschalten der beiden Kopplungen TRAILOF mit 3 Parametern aufgerufen werden: <code>TRAILOF(<Folgeachse>, <Leitachse>, <Leitachse 2>)</code> Das gleiche Ergebnis liefert die Programmierung von TRAILOF ohne Angabe einer Leitachse: <code>TRAILOF(<Folgeachse>)</code>

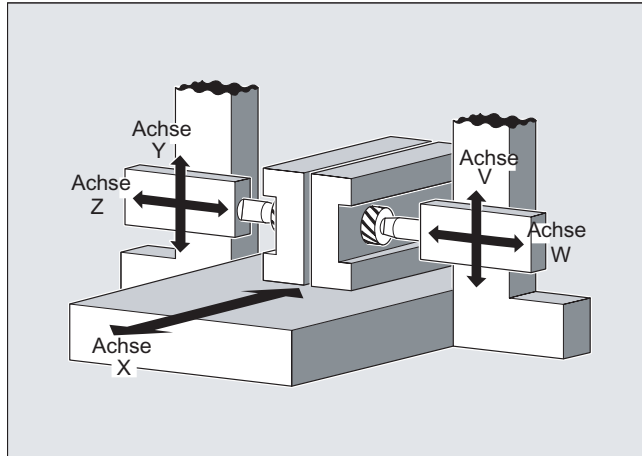
Hinweis

Das Mitschleppen erfolgt immer im Basiskoordinatensystem (BKS).

Die Anzahl der gleichzeitig aktivierbaren Mitschleppverbände wird nur begrenzt durch die Kombinationsmöglichkeiten der an der Maschine vorhandenen Achsen.

Beispiel

Das Werkstück soll zweiseitig mit der dargestellten Achskonstellation bearbeitet werden. Dazu bilden Sie 2 Mitschleppverbände.



Programmcode	Kommentar
...	
N100 TRAILON(V,Y)	; Einschalten des 1. Mitschleppverbandes
N110 TRAILON(W,Z,-1)	; Einschalten des 2. Mitschleppverbandes. Koppelfaktor negativ: Mitschleppachse fährt jeweils in entgegengesetzter Richtung wie Leitachse.
N120 G0 Z10	; Zustellung der Z- und W-Achse in entgegengesetzter Achsrichtung.
N130 G0 Y20	; Zustellung der Y- und V-Achse in gleicher Achsrichtung.
...	
N200 G1 Y22 V25 F200	; Überlagerung einer abhängigen und unabhängigen Bewegung der Mitschleppachse V.
...	
TRAILOF(V,Y)	; Ausschalten des 1. Mitschleppverbandes.
TRAILOF(W,Z)	; Ausschalten des 2. Mitschleppverbandes.

Weitere Informationen**Achstypen**

Ein Mitschleppverband kann aus beliebigen Kombinationen von Linear- und Rundachsen bestehen. Als Leitachse kann dabei auch eine simulierte Achse definiert werden.

Mitschleppachsen


Einer Mitschleppachse können gleichzeitig maximal 2 Leitachsen zugeordnet werden. Die Zuordnung erfolgt in unterschiedlichen Mitschleppverbänden.

Eine Mitschleppachse kann mit allen zur Verfügung stehenden Bewegungsbefehlen programmiert werden (G0, G1, G2, G3, ...). Zusätzlich zu den unabhängig definierten Wegen fährt die Mitschleppachse die mit den Koppelfaktoren aus ihren Leitachsen abgeleiteten Wege.

Dynamikbegrenzung

Die Dynamikbegrenzung ist abhängig von der Art der Aktivierung des Mitschleppverbandes:

- **Aktivierung im Teileprogramm**
Erfolgt die Aktivierung im Teileprogramm und sind alle Leitachsen als Programmachsen im aktivierenden Kanal, wird beim Verfahren der Leitachsen die Dynamik aller Mitschleppachsen so berücksichtigt, dass keine Mitschleppachse überlastet wird. Erfolgt die Aktivierung im Teileprogramm mit Leitachsen, die nicht als Programmachsen im aktivierenden Kanal aktiv sind (\$AA_TYP ≠ 1), wird beim Verfahren der Leitachsen die Dynamik der Mitschleppachse nicht berücksichtigt. Dadurch kann es bei Mitschleppachsen mit einer geringeren als der für die Kopplung benötigten Dynamik zu einer Überlastung kommen.
- **Aktivierung in Synchronaktion**
Erfolgt die Aktivierung in einer Synchronaktion, wird beim Verfahren der Leitachsen die Dynamik der Mitschleppachsen nicht berücksichtigt. Dadurch kann es bei Mitschleppachsen mit einer geringeren als der für die Kopplung benötigten Dynamik zu einer Überlastung kommen.

 **VORSICHT**

Achsüberlastung

Wird ein Mitschleppverband

- in Synchronaktionen
- im Teileprogramm mit Leitachsen, die nicht Programmachsen im Kanal der Mitschleppachse sind,

aktiviert, dann liegt es in der besonderen Verantwortung des Anwenders/ Maschinenherstellers, geeignete Maßnahmen vorzusehen, damit es durch die Verfahrbewegungen der Leitachse nicht zu einer Überlastung der Mitschleppachsen kommt.

Kopplungsstatus

Der Kopplungsstatus einer Achse kann im Teileprogramm abgefragt werden mit der Systemvariablen:

\$AA_COUP_ACT[<Achse>]

Wert	Bedeutung
0	Keine Kopplung aktiv
8	Mitschleppen aktiv

Restweganzeige der Mitschleppachse bei Modulo-Rundachsen

Sind Leit- und Mitschleppachse Modulo-Rundachsen, summieren sich Verfahrbewegungen in der Leitachse von n * 360° mit n = 1, 2, 3... , in der Restweganzeige der Mitschleppachse bis zum Ausschalten der Kopplung auf.

Beispiel: Programmabschnitt mit TRAILON und Leitachse B und Folgeachse C

Programmcode	Kommentar
TRAILON(C,B,1)	; Kopplung einschalten
G0 B0	; Ausgangsposition

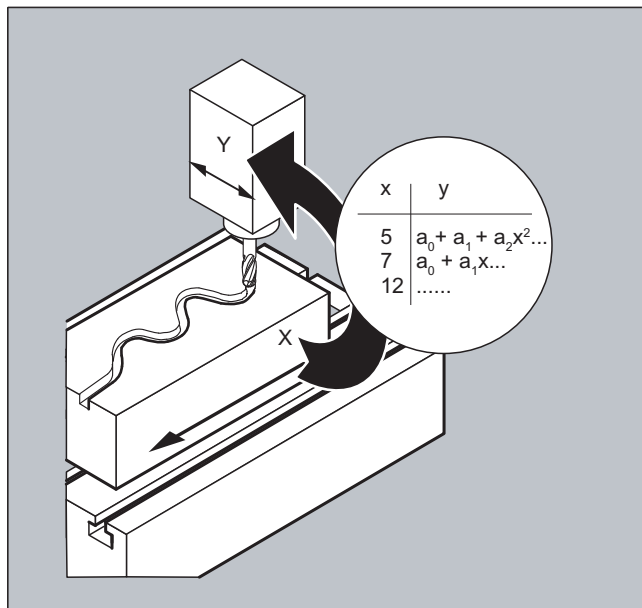
Programmcode	Kommentar
	; Restweganzeige am Satzanfang:
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

12.2 Kurventabellen (CTAB)

Mit Hilfe von Kurventabellen können Positions- und Geschwindigkeitsbeziehungen zwischen zwei Achsen (Leit- und Folgeachse) programmiert werden. Die Kurventabellendefinition erfolgt im Teileprogramm.

Anwendung

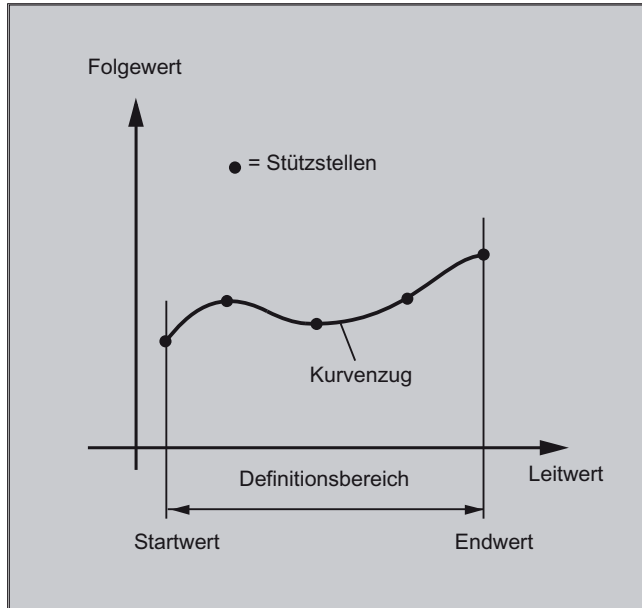
Kurventabellen ersetzen mechanische Kurvenscheiben. Die Kurventabelle bildet dabei die Grundlage für die axiale Leitwertkopplung, indem sie den funktionellen Zusammenhang zwischen Leit- und Folgewert schafft: Die Steuerung berechnet bei entsprechender Programmierung aus einander zugeordneten Positionen von Leit- und Folgeachse ein Polynom, das der Kurvenscheibe entspricht.



12.2.1 Kurventabellen definieren (CTABDEF, CATBEND)

Eine Kurventabelle stellt ein Teileprogramm oder einen Teileprogrammabschnitt dar, welcher durch Voranstellen von CTABDEF und den abschließenden Befehl CTABEND gekennzeichnet ist.

Innerhalb dieses Teileprogrammabschnitts werden durch Bewegungsanweisungen einzelnen Positionen der Leitachse eindeutige Folgeachspositionen zugeordnet, die als Stützstellen für die Berechnung eines Kurvenzugs in Form eines Polynoms bis zu maximal 5. Grades dienen.



Voraussetzung

Für die Definition von Kurventabellen muss durch entsprechende MD-Projektierung Speicherplatz reserviert sein (→ Maschinenhersteller!).

Syntax

```
CTABDEF (<Folgeachse>, <Leitachse>, <n>, <Periodizität>[, <Speicherort>])
...
CTABEND
```

Bedeutung

CTABDEF ():	Beginn der Kurventabellendefinition
CTABEND:	Ende der Kurventabellendefinition
<Folgeachse>:	Achse, deren Bewegung über die Kurventabelle berechnet werden soll
<Leitachse>:	Achse, die die Leitwerte zur Berechnung der Folgeachsbelegung liefert
<n>:	Nummer (ID) der Kurventabelle Die Nummer einer Kurventabelle ist eindeutig und unabhängig vom Speicherort. Es können keine Tabellen mit der gleichen Nummer im statischen und dynamischen NC-Speicher liegen.

<Periodizität>:	Tabellenperiodizität	
	0	Tabelle ist nicht periodisch (wird nur einmal abgearbeitet, auch bei Rundachsen)
	1	Tabelle ist periodisch bezüglich Leitachse
	2	Tabelle ist periodisch bezüglich Leitachse und Folgeachse
<Speicherort>:	Angabe des Speicherorts (optional)	
	"SRAM"	Die Kurventabelle wird im statischen NC-Speicher angelegt.
	"DRAM"	Die Kurventabelle wird im dynamischen NC-Speicher angelegt.
	Hinweis: Wenn für diesen Parameter kein Wert programmiert wird, dann wird der mit MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE eingestellte Standard-Speicherort verwendet.	

Hinweis

Überschreiben

Eine Kurventabelle wird überschrieben, sobald bei einer neuen Tabellendefinition deren Nummer (<n>) benutzt wird (Ausnahme: die Kurventabelle ist in einer Achskopplung aktiv oder mit CTABLOCK gesperrt). **Beim Überschreiben wird keine entsprechende Warnung ausgegeben!**

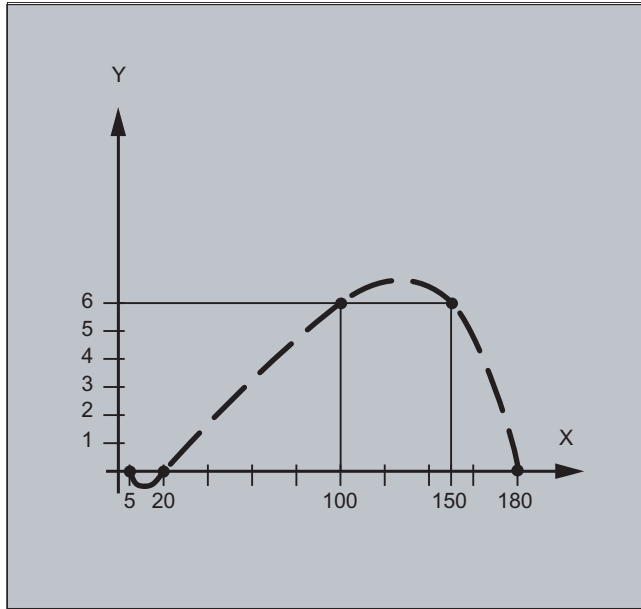
Beispiele

Beispiel 1: Programmabschnitt als Kurventabellendefinition

Ein Programmabschnitt soll unverändert zur Definition einer Kurventabelle benutzt werden. Der darin auftretende Befehl zum Vorlaufstopp STOPRE kann stehen bleiben und wird sofort wieder aktiv, sobald der Programmabschnitt nicht mehr zur Tabellendefinition benutzt wird und CTABDEF und CTABEND entfernt wurden.

Programmcode	Kommentar
...	
CTABDEF(Y,X,1,1)	; Definition einer Kurventabelle.
...	
IF NOT (\$P_CTABDEF)	
STOPRE	
ENDIF	
...	
CTABEND	

Beispiel 2: Definition einer nichtperiodischen Kurventabelle



Programmcode	Kommentar
N100 CTABDEF (Y,X,3,0)	; Beginn der Definition einer nichtperiodischen Kurventabelle mit der Nummer 3.
N110 X0 Y0	; 1.Bewegungsanweisung, legt Startwerte und 1. Stützstelle fest: Leitwert: 0, Folgewert: 0
N120 X20 Y0	; 2.Stützstelle: Leitwert: 0...20, Folgewert: Startwert...0
N130 X100 Y6	; 3.Stützstelle: Leitwert: 20...100, Folgewert: 0...6
N140 X150 Y6	; 4.Stützstelle: Leitwert: 100...150, Folgewert: 6...6
N150 X180 Y0	; 5.Stützstelle: Leitwert: 150...180, Folgewert: 6...0
N200 CTABEND	; Ende der Definition. Die Kurventabelle wird in ihrer internen Darstellung als Polynom maximal 5.Grades erzeugt. Die Berechnung des Kurvenzugs mit den angegebenen Stützstellen ist abhängig von der modal gewählten Interpolationsart (Kreis-, Linear-, Spline-Interpolation). Der Teileprogrammzustand vor Beginn der Definition wird wiederhergestellt.

Beispiel 3: Definition einer periodischen Kurventabelle

Definition einer periodischen Kurventabelle mit Nummer 2, Leitwertbereich von 0 bis 360, Folgeachsabewegung von 0 nach 45 und zurück nach 0:

Programmcode	Kommentar
N10 DEF REAL DEPP0S	

Programmcode	Kommentar
N20 DEF REAL GRADIENT	
N30 CTABDEF (Y, X, 2, 1)	; Beginn der Definition.
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	; Ende der Definition.
;Test der Kurve durch Kopplung von Y an 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)	; Lesen der Tabellenfunktion beim Leitwert 75.0.
N180 G0 X75 Y=DEPPOS	; Positionieren von Leit- und Fol- geachse.
;Nach Einschalten der Kopplung ist kein Synchronisieren der Folgeachse nötig.	
N190 LEADON (Y, X, 2)	
N200 G1 X110 F1000	
N210 LEADOF (Y, X)	
N220 M30	

Weitere Informationen

Start- und Endwert der Kurventabelle

Als Startwert für den Beginn des Definitionsbereichs der Kurventabelle gilt die erste Angabe von zusammengehörigen Achspositionen (die erste Bewegungsanweisung) innerhalb der Kurventabellendefinition. Der Endwert des Definitionsbereichs der Kurventabelle wird entsprechend durch den letzten Verfabrbefehl bestimmt.

Verfügbarer Sprachumfang

Innerhalb der Definition der Kurventabelle steht der gesamte NC-Sprachumfang zur Verfügung.

Hinweis

Folgende Angaben sind In Kurventabellendefinitionen nicht zulässig:

- Vorlaufstopp
- Sprünge in der Leitachsenbewegung (z. B. beim Wechsel von Transformationen)
- Bewegungsanweisung allein für die Folgeachse
- Bewegungsumkehr der Leitachse, d. h. Position der Leitachse muss immer eindeutig sein
- CTABDEF- und CTABEND-Anweisung in unterschiedlichen Programmebenen.

Wirksamkeit von modalen Anweisungen

Sämtliche modal wirksamen Anweisungen, die innerhalb der Kurventabellendefinition getroffen werden, sind mit Abschluss der Tabellendefinition ungültig. Das Teileprogramm, in dem die Tabellendefinition erfolgt, befindet sich damit vor und nach der Tabellendefinition im gleichen Zustand.

Zuweisungen an R-Parameter

Zuweisungen an R-Parameter innerhalb der Tabellendefinition werden nach CTABEND zurückgesetzt.

Beispiel:

Programmcode	Kommentar
...	
R10=5 R11=20	; R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	; R10=25
X=R10	
CTABEND	
...	; R10=5

Aktivierung von ASPLINE, BSPLINE, CSPLINE

Wird innerhalb einer Kurventabellendefinition CTABDEF ... CTABEND ein ASPLINE, BSPLINE oder CSPLINE aktiviert, so sollte vor dieser Spline-Aktivierung mindestens ein Startpunkt programmiert werden. Eine sofortige Aktivierung nach CTABDEF sollte vermieden werden, da sonst der Spline von der aktuellen Achsposition vor der Kurventabellendefinition abhängt.

Beispiel:

Programmcode
...

Programmcode

```
CTABDEF (Y, X, 1, 0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND
```

Wiederholte Verwendung von Kurventabellen

Der über die Kurventabelle berechnete funktionelle Zusammenhang von Leit- und Folgeachse bleibt unter der gewählten Tabellenummer über das Teileprogrammende und über POWER OFF hinaus erhalten, falls die Tabelle im statischen NC-Speicher (SRAM) abgelegt ist.

Eine Tabelle, die im dynamischen Speicher (DRAM) angelegt wurde, wird bei POWER ON gelöscht und muss eventuell noch einmal erzeugt werden.

Die einmal erstellte Kurventabelle lässt sich auf beliebige Achskombinationen von Leit- und Folgeachse anwenden und ist unabhängig davon, welche Achsen zur Erstellung der Kurventabelle benutzt wurden.

Überschreiben von Kurventabellen

Eine Kurventabelle wird überschrieben, sobald bei einer erneuten Tabellendefinition deren Nummer benutzt wird.

Ausnahme: Eine Kurventabelle ist in einer Achskopplung aktiv oder mit CTABLOCK gesperrt.

Hinweis

Beim Überschreiben von Kurventabellen wird keine entsprechende Warnung ausgegeben!

Kurventabellendefinition aktiv?

Mit der Systemvariablen `$P_CTABDEF` kann aus dem Teileprogramm heraus jederzeit abgefragt werden, ob eine Kurventabellendefinition aktiv ist.

Aufheben der Kurventabellendefinition

Der Teileprogrammabschnitt ist nach Ausklammern der Anweisungen zur Kurventabellendefinition wieder als reales Teileprogramm verwendbar.

Laden von Kurventabellen über "Abarbeiten von Extern"

Beim externen Abarbeiten von Kurventabellen muss die Größe des Nachladebuffers (DRAM) über `MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE` so gewählt werden, dass die gesamte Kurventabellendefinition gleichzeitig im Nachladebuffer abgelegt werden kann. Die Teileprogrammbearbeitung wird anderenfalls mit einem Alarm abgebrochen.

Sprünge der Folgeachse

Abhängig von der Einstellung im Maschinendatum:
`MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION`
können Sprünge der Folgeachse bei fehlender Bewegung der Leitachse toleriert werden.

12.2.2 Vorhandensein einer Kurventabelle prüfen (CTABEXISTS)

Mit dem Befehl CTABEXISTS kann geprüft werden, ob eine bestimmte Kurventabellennummer im NC-Speicher vorhanden ist.

Syntax

```
CTABEXISTS (<n>)
```

Bedeutung

CTABEXISTS:	Prüft, ob die Kurventabelle mit Nummer <n> im statischen oder dynamischen NC-Speicher vorhanden ist	
	0	Tabelle existiert nicht
	1	Tabelle existiert
<n>:	Nummer (ID) der Kurventabelle	

12.2.3 Kurventabellen löschen (CTABDEL)

Mit CTABDEL können Kurventabellen gelöscht werden.

Hinweis

Kurventabellen, die in einer Achskopplung aktiv sind, können nicht gelöscht werden.

Syntax

```
CTABDEL (<n>)
CTABDEL (<n>, <m>)
CTABDEL (<n>, <m>, <Speicherort>)
CTABDEL ()
CTABDEL (, , <Speicherort>)
```

Bedeutung

CTABDEL:	Befehl zum Löschen von Kurventabellen
<n>:	Nummer (ID) der zu löschenden Kurventabelle Beim Löschen eines Kurventabellenbereichs CTABDEL (<n>, <m>) wird mit <n> die Nummer der ersten Kurventabelle des Bereichs angegeben.
<m>:	Beim Löschen eines Kurventabellenbereichs CTABDEL (<n>, <m>) wird mit <m> die Nummer der letzten Kurventabelle des Bereichs angegeben. <m> muss größer <n> sein!

<Speicherort>:	Angabe des Speicherorts (optional)	
	Beim Löschen ohne Speicherort-Angabe werden die angegebenen Kurventabellen im statischen und dynamischen NC-Speicher gelöscht.	
	Beim Löschen mit Speicherort-Angabe werden von den angegebenen Kurventabellen nur diejenigen gelöscht, die im angegebenen Speicher liegen. Die übrigen bleiben bestehen.	
"SRAM"	Löschen im statischen NC-Speicher	
"DRAM"	Löschen im dynamischen NC-Speicher	

Wird CTABDEL ohne Angabe der zu löschenden Kurventabelle programmiert, dann werden **alle** Kurventabellen bzw. alle Kurventabellen im angegebenen Speicher gelöscht:

CTABDEL () :	Löscht alle Kurventabellen im statischen und dynamischen NC-Speicher
CTABDEL (, , "SRAM") :	Löscht alle Kurventabellen im statischen NC-Speicher
CTABDEL (, , "DRAM") :	Löscht alle Kurventabellen im dynamischen NC-Speicher

Hinweis

Wenn beim Mehrfachlöschen CTABDEL (<n>, <m>) oder CTABDEL () wenigstens eine der zu löschenden Kurventabellen in einer Kopplung aktiv ist, dann wird der Löschbefehl nicht ausgeführt, d. h. **keine** der adressierten Kurventabellen wird gelöscht.

12.2.4 Kurventabellen gegen Löschen und Überschreiben sperren (CTABLOCK, CTABUNLOCK)

Kurventabellen können durch Setzen von Sperren vor unbeabsichtigtem Löschen und Überschreiben geschützt werden. Eine gesetzte Sperre kann jederzeit auch wieder aufgehoben werden.

Syntax

Sperre setzen:

```
CTABLOCK (<n>)
CTABLOCK (<n>, <m>)
CTABLOCK (<n>, <m>, <Speicherort>)
CTABLOCK ( )
CTABLOCK ( , , <Speicherort>)
```

Sperre aufheben:

```
CTABUNLOCK (<n>)
CTABUNLOCK (<n>, <m>)
CTABUNLOCK (<n>, <m>, <Speicherort>)
CTABUNLOCK ( )
CTABUNLOCK ( , , <Speicherort>)
```

Bedeutung

CTABLOCK:	Befehl zum Setzen einer Sperre gegen Löschen/Überschreiben
CTABUNLOCK:	Befehl zum Aufheben einer Sperre gegen Löschen/Überschreiben CTABUNLOCK gibt die mit CTABLOCK gesperrten Kurventabellen wieder frei. Kurventabellen, die in einer aktiven Kopplung wirken, bleiben weiterhin gesperrt und können nicht gelöscht werden. Die Sperre mit CTABLOCK ist aufgehoben, sobald die Sperrung durch die aktive Kopplung mit Deaktivierung der Kopplung aufgehoben wird. Damit kann diese Tabelle gelöscht werden. Ein nochmaliger CTABUNLOCK-Aufruf ist nicht notwendig.
<n>:	Nummer (ID) der zu sperrenden/entsperrenden Kurventabelle Beim Sperren/Entsperren eines Kurventabellenbereichs CTABLOCK (<n>, <m>)/CTABUNLOCK (<n>, <m>) wird mit <n> die Nummer der ersten Kurventabelle des Bereichs angegeben.
<m>:	Beim Sperren/Entsperren eines Kurventabellenbereichs CTABLOCK (<n>, <m>)/CTABUNLOCK (<n>, <m>) wird mit <m> die Nummer der letzten Kurventabelle des Bereichs angegeben. <m> muss größer <n> sein!
<Speicherort>:	Angabe des Speicherorts (optional) Beim Setzen/Aufheben einer Sperre ohne Speicherort-Angabe werden die angegebenen Kurventabellen im statischen und dynamischen NC-Speicher gesperrt/entsperrt. Beim Setzen/Aufheben einer Sperre mit Speicherort-Angabe werden von den angegebenen Kurventabellen nur diejenigen gesperrt/entsperrt, die im angegebenen Speicher liegen. Die übrigen werden nicht gesperrt/entsperrt.
	"SRAM" Sperre setzen/aufheben im statischen NC-Speicher
	"DRAM" Sperre setzen/aufheben im dynamischen NC-Speicher

Wird CTABLOCK/CTABUNLOCK ohne Angabe der zu sperrenden/entsperrenden Kurventabelle programmiert, dann werden **alle** Kurventabellen bzw. alle Kurventabellen im angegebenen Speicher gesperrt/entsperrt:

CTABLOCK () :	Sperrt alle Kurventabellen im statischen und dynamischen NC-Speicher
CTABLOCK (, , "SRAM") :	Sperrt alle Kurventabellen im statischen NC-Speicher
CTABLOCK (, , "DRAM") :	Sperrt alle Kurventabellen im dynamischen NC-Speicher
CTABUNLOCK () :	Entsperrt alle Kurventabellen im statischen und dynamischen NC-Speicher
CTABUNLOCK (, , "SRAM") :	Entsperrt alle Kurventabellen im statischen NC-Speicher
CTABUNLOCK (, , "DRAM") :	Entsperrt alle Kurventabellen im dynamischen NC-Speicher

12.2.5 Kurventabellen: Tabelleneigenschaften ermitteln (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Mit diesen Befehlen können wichtige Eigenschaften einer Kurventabelle (Tabellenummer, Sperrzustand, Speicherort, Periodizität) abgefragt werden.

Syntax

```
CTABID (<p>)
CTABID (<p>, <Speicherort>)
CTABISLOCK (<n>)
CTABMEMTYP (<n>)
TABPERIOD (<n>)
```

Bedeutung

CTABID:	Liefert die Tabellennummer , die im angegebenen Speicher als die <p>-te Kurventabelle eingetragen ist. Beispiel: CTABID(1, "SRAM") liefert die Nummer der ersten Kurventabelle im statischen NC-Speicher. Die erste Kurventabelle entspricht dabei der Kurventabelle mit der höchsten Tabellennummer. Hinweis: Wird zwischen aufeinander folgenden Aufrufen von CTABID die Reihenfolge der Kurventabellen im Speicher geändert, z. B. durch Löschen von Kurventabellen mit CTABDEL, kann CTABID (<p>, . . .) mit derselben Nummer <p> eine andere Kurventabelle liefern als vorher.	
CTABISLOCK:	Gibt den Sperrzustand der Kurventabelle mit Nummer <n> zurück:	
	0	Tabelle ist nicht gesperrt
	1	Tabelle ist gesperrt durch CTABLOCK
	2	Tabelle ist gesperrt durch aktive Kopplung
	3	Tabelle ist gesperrt durch CTABLOCK und aktive Kopplung
	-1	Tabelle existiert nicht
CTABMEMTYP:	Liefert den Speicherort der Kurventabelle mit Nummer <n>:	
	0	Tabelle im statischen NC-Speicher
	1	Tabelle im dynamischen NC-Speicher
	-1	Tabelle existiert nicht
CTABPERIOD:	Liefert die Periodizität der Kurventabelle mit Nummer <n>:	
	0	Tabelle ist nicht periodisch
	1	Tabelle ist periodisch in der Leitachse
	2	Tabelle ist periodisch in der Leit- und Folgeachse
	-1	Tabelle existiert nicht
<p>:	Eintragsnummer im Speicher	
<n>:	Nummer (ID) der Kurventabelle	
<Speicherort>:	Angabe des Speicherorts (optional)	
	"SRAM"	Statischer NC-Speicher
	"DRAM"	Dynamischer NC-Speicher
	Hinweis: Wenn für diesen Parameter kein Wert programmiert wird, dann wird der mit MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE eingestellte Standard-Speicherort verwendet.	

12.2.6 Kurventabellenwerte lesen (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Folgende Kurventabellenwerte können im Teileprogramm gelesen werden:

- Folgeachs- und Leitachswerte am Anfang und Ende einer Kurventabelle
- Folgeachswerte am Anfang und Ende eines Kurvensegments
- Folgeachswert zu einem Leitachswert
- Leitachswert zu einem Folgeachswert
- Minimal- und Maximalwert der Folgeachse
 - im gesamten Definitionsbereich der Kurventabelle
 - oder
 - in einem definierten Kurventabellenintervall

Syntax

```
CTABTSV (<n>, <Gradient> [, <Folgeachse>])
CTABTEV (<n>, <Gradient> [, <Folgeachse>])
CTABTSP (<n>, <Gradient> [, <Leitachse>])
CTABTEP (<n>, <Gradient> [, <Leitachse>])
CTABSSV (<Leitwert>, <n>, <Gradient> [, <Folgeachse>])
CTABSEV (<Leitwert>, <n>, <Gradient> [, <Folgeachse>])
CTAB (<Leitwert>, <n>, <Gradient> [, <Folgeachse>, <Leitachse>])
CTABINV (<Folgewert>, <Näherungswert>, <n>, <Gradient> [, <Folgeachse>, <Leitachse>])
CTABTMIN (<n> [, <Folgeachse>])
CTABTMAX (<n> [, <Folgeachse>])
CTABTMIN (<n>, <a>, <b> [, <Folgeachse>, <Leitachse>])
CTABTMAX (<n>, <a>, <b> [, <Folgeachse>, <Leitachse>])
```

Bedeutung

CTABTSV:	Folgeachswert am Anfang der Kurventabelle Nr. <n> lesen
CTABTEV:	Folgeachswert am Ende der Kurventabelle Nr. <n> lesen
CTABTSP:	Leitachswert am Anfang der Kurventabelle Nr. <n> lesen
CTABTEP:	Leitachswert am Ende der Kurventabelle Nr. <n> lesen
CTABSSV:	Folgeachswert am Anfang des zum angegebenen Leitachswert (<Leitwert>) gehörenden Kurvensegments lesen
CTABSEV:	Folgeachswert am Ende des zum angegebenen Leitachswert (<Leitwert>) gehörenden Kurvensegments lesen
CTAB:	Folgeachswert zum angegebenen Leitachswert (<Leitwert>) lesen
CTABINV:	Leitachswert zum angegebenen Folgeachswert (<Folgewert>) lesen
CTABTMIN:	Minimalwert der Folgeachse bestimmen: <ul style="list-style-type: none"> • im gesamten Definitionsbereich der Kurventabelle oder • in einem definierten Intervall <a> ...

CTABTMAX:	Maximalwert der Folgeachse bestimmen: <ul style="list-style-type: none"> im gesamten Definitionsbereich der Kurventabelle oder in einem definierten Intervall <a> ...
<n>:	Nummer (ID) der Kurventabelle
<Gradient>:	Im Parameter <Gradient> wird die Steigung der Kurventabellenfunktion an der ermittelten Position zurückgegeben
<Folgeachse>:	Achse, deren Bewegung über die Kurventabelle berechnet werden soll (optional)
<Leitachse>:	Achse, die die Leitwerte zur Berechnung der Folgeachsbewegung liefert (optional)
<Folgewert>:	Folgeachswert zum Lesen des zugehörigen Leitachswerts bei CTABINV
<Leitwert>:	Leitachswert: <ul style="list-style-type: none"> zum Lesen des zugehörigen Folgeachswerts bei CTAB oder für die Auswahl des Kurvensegments bei CTABSSV/CTABSEV
<Näherungswert>:	Die Zuordnung eines Leitachswerts zu einem Folgeachswert bei CTABINV muss nicht immer eindeutig sein. CTABINV benötigt daher als Parameter einen Näherungswert für den erwarteten Leitachswert.
<a>:	Untere Grenze des Leitwertintervalls bei CTABTMIN/CTABTMAX
:	Obere Grenze des Leitwertintervalls bei CTABTMIN/CTABTMAX
	Hinweis: Das Leitwertintervall <a> ... muss innerhalb des Definitionsbereichs der Kurventabelle liegen.

Beispiele

Beispiel 1:

Folgeachs- und Leitachswerte am Anfang und Ende der Kurventabelle sowie Minimal- und Maximalwert der Folgeachse im gesamten Definitionsbereich der Kurventabelle bestimmen.

Programmcode	Kommentar
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)	; Beginn der Tabellendefinition
N110 X0 Y10	; Startposition 1.Tabellensegment
N120 X30 Y40	; Endposition 1.Tabellensegment = Startposition 2.Tabellensegment
N130 X60 Y5	; Endposition 2.Tabellensegment = ...
N140 X70 Y30	

Programmcode	Kommentar
N150 X80 Y20	
N160 CTABEND	; Ende der Tabellendefinition.
...	
N200 STARTPOS=CTABTSV(1,GRADIENT)	; Folgeachswert am Kurventabellenanfang = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)	; Folgeachswert am Kurventabellenende = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)	; Leitachswert am Kurventabellenanfang = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)	; Leitachswert am Kurventabellenende = 80
N240 MINVAL=CTABTMIN(1)	; Minimalwert der Folgeachse bei Y=5
N250 MAXVAL=CTABTMAX(1)	; Maximalwert der Folgeachse bei Y=40

Beispiel 2:

Bestimmung der Folgeachswerte am Anfang und Ende des zum Leitachswert X=30 gehörenden Kurvensegments.

Programmcode	Kommentar
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; Beginn der Tabellendefinition.
N110 X0 Y0	; Startposition 1.Tabellensegment
N120 X20 Y10	; Endposition 1.Tabellensegment = Startposition 2.Tabellensegment
N130 X40 Y40	; Endposition 2.Tabellensegment = ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; Ende der Tabellendefinition.
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; Startposition Y im 2.Segment = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	; Endposition Y im 2.Segment = 40

Weitere Informationen**Verwendung in Synchronaktionen**

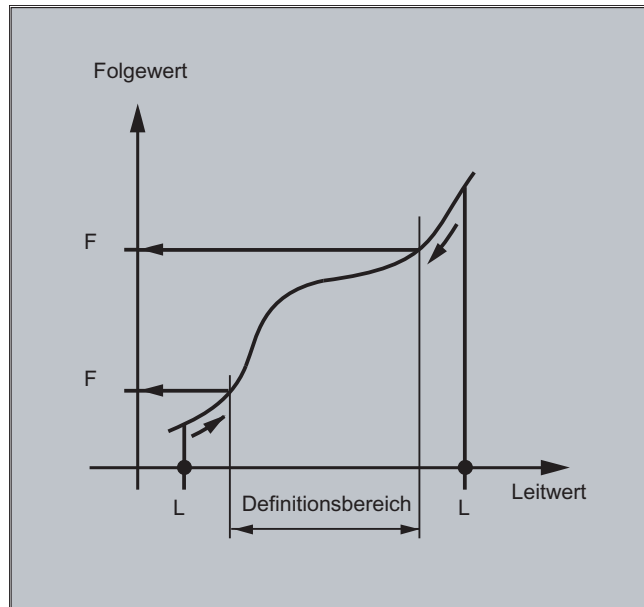
Alle Befehle zum Lesen von Kurventabellenwerten können auch in Synchronaktionen verwendet werden (siehe auch Kapitel "Bewegungssynchronaktionen").

Bei Verwendung der Befehle CTABINV, CTABTMIN und CTABTMAX ist darauf zu achten, dass:

- zum Ausführungszeitpunkt ausreichend NC-Leistung verfügbar ist oder
- vor dem Aufruf die Anzahl der Segmente der Kurventabelle abgefragt wird, um gegebenenfalls die betreffende Tabelle unterteilen zu können

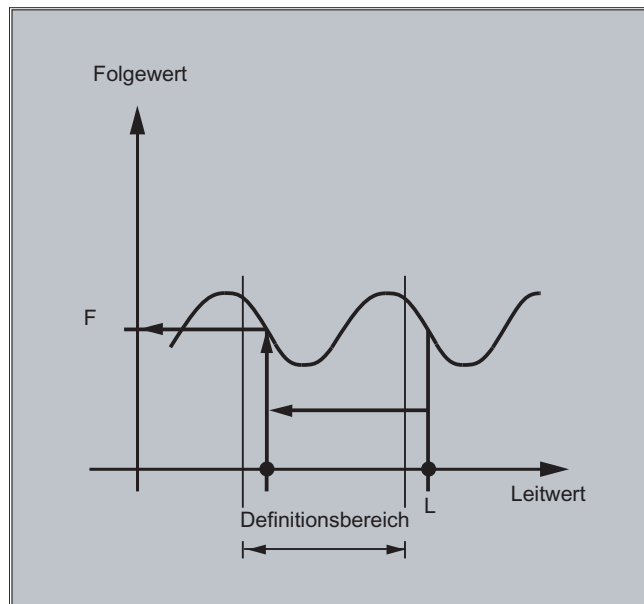
CTAB bei nichtperiodischen Kurventabellen

Liegt der angegebene <Leitwert> außerhalb des Definitionsbereichs, wird als Folgewert die obere bzw. untere Grenze ausgegeben:



CTAB bei periodischen Kurventabellen

Liegt der angegebene <Leitwert> außerhalb des Definitionsbereichs, wird der Leitwert Modulo des Definitionsbereichs bewertet und der entsprechende Folgewert ausgegeben:



Näherungswert für CTABINV

Der Befehl CTABINV benötigt einen Näherungswert für den erwarteten Leitwert. CTABINV gibt den Leitwert zurück, der dem Näherungswert am nächsten liegt. Der Näherungswert kann z. B. der Leitwert aus dem vorherigen Interpolatortakt sein.

Steigung der Kurventabellenfunktion

Die Ausgabe der Steigung (<Gradient>) ermöglicht es, die Geschwindigkeit der Leit- oder Folgeachse an der entsprechenden Position zu berechnen.

Angabe der Leit- oder Folgeachse

Die optionale Angabe der Leit- und/oder Folgeachse ist wichtig, falls Leit- und Folgeachse in verschiedenen Längeneinheiten projiziert sind.

CTABSSV, CTABSEV

Die Befehle CTABSSV und CTABSEV sind in folgenden Fällen **nicht** dazu geeignet, programmierte Segmente abzufragen:

- Kreise oder Evolventen sind programmiert.
- Fasen bzw. Runden mit CHF/RND ist aktiv.
- Überschleifen mit G643 ist aktiv.
- NC-Satz-Kompression mit COMPON/COMPCURV/COMPCAD ist aktiv.

12.2.7

Kurventabellen: Ressourcennutzung prüfen (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Mit diesen Befehlen hat der Programmierer die Möglichkeit, sich aktuell über die Belegung der Ressourcen für Kurventabellen, Tabellensegmente und Polynome zu informieren.

Syntax

```
CTABNO
CTABNOMEM(<Speicherort>)
CTABFNO(<Speicherort>)
CTABSEGID(<n>,<Speicherort>)
CTABSEG(<Speicherort>,<Segmentart>)
CTABFSEG(<Speicherort>,<Segmentart>)
CTABMSEG(<Speicherort>,<Segmentart>)
CTABPOLID(<n>)
CTABPOL(<Speicherort>)
CTABFPOL(<Speicherort>)
CTABMPOL(<Speicherort>)
```


Bedeutung

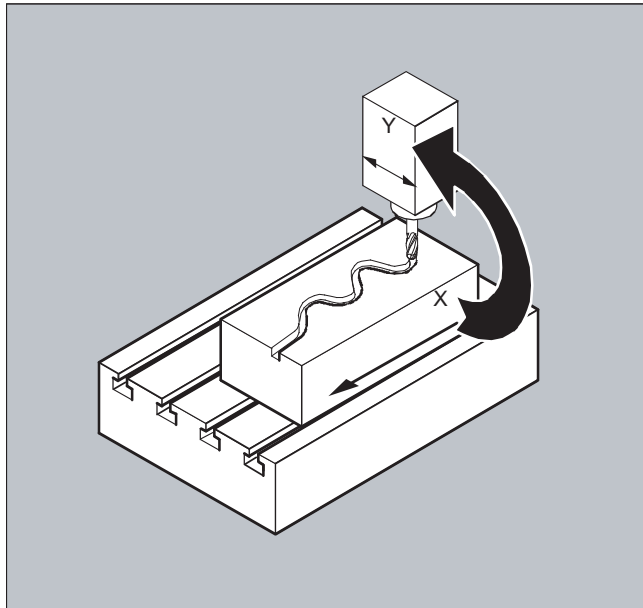
CTABNO:	Gesamtanzahl der definierten Kurventabellen bestimmen (im statischen und dynamischen NC-Speicher)	
CTABNOMEM:	Anzahl der definierten Kurventabellen im angegebenen <Speicherort> bestimmen	
CTABFNO:	Anzahl der noch möglichen Kurventabellen im angegebenen <Speicherort> bestimmen	
CTABSEGID:	Anzahl der Kurvensegmente der angegebenen <Segmentart> bestimmen, die von der Kurventabelle mit Nummer <n> verwendet werden	
CTABSEG:	Anzahl der verwendeten Kurvensegmente der angegebenen <Segmentart> im angegebenen <Speicherort> bestimmen	
CTABFSEG:	Anzahl der noch möglichen Kurvensegmente der angegebenen <Segmentart> im angegebenen <Speicherort> bestimmen	
CTABMSEG:	Anzahl der maximal möglichen Kurvensegmente der angegebenen <Segmentart> im angegebenen <Speicherort> bestimmen	
CTABPOLID:	Anzahl der Kurvenpolynome bestimmen, die von der Kurventabelle mit Nummer <n> verwendet werden	
CTABPOL:	Anzahl der verwendeten Kurvenpolynome im angegebenen <Speicherort> bestimmen	
CTABFPOL:	Anzahl der noch möglichen Kurvenpolynome im angegebenen <Speicherort> bestimmen	
CTABMPOL:	Anzahl der maximal möglichen Kurvenpolynome im angegebenen <Speicherort> bestimmen	
<n>:	Nummer (ID) der Kurventabelle	
<Speicherort>:	Angabe des Speicherorts (optional)	
	"SRAM"	Statischer NC-Speicher
	"DRAM"	Dynamischer NC-Speicher
	Hinweis: Wenn für diesen Parameter kein Wert programmiert wird, dann wird der mit MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE eingestellte Standard-Speicherort verwendet.	
<Segmentart>:	Angabe der Segmentart (optional)	
	"L"	Lineare Segmente
	"P"	Polynomsegmente
	Hinweis: Wenn für diesen Parameter kein Wert programmiert wird, dann wird die Summe aus Linear- und Polynom-Segmenten ausgegeben.	

12.3 Axiale Leitwertkopplung (LEADON, LEADOF)

Hinweis

Diese Funktion steht für SINUMERIK 828D nicht zur Verfügung!

Bei der axialen Leitwertkopplung werden eine Leit- und eine Folgeachse synchron verfahren. Dabei ist die jeweilige Position der Folgeachse über eine Kurventabelle bzw. ein daraus berechnetes Polynom eindeutig einer - ggf. simulierten - Position der Leitachse zugeordnet.



Leitachse heißt diejenige Achse, die die Eingangswerte für die Kurventabelle liefert.
Folgeachse heißt die Achse, die die über die Kurventabelle errechneten Positionen einnimmt.

Ist- und Sollwertkopplung

Als Leitwerte, also Ausgangswerte zur Positionsermittlung der Folgeachse können verwendet werden:

- Istwerte der Leitachsposition: Istwertkopplung
- Sollwerte der Leitachsposition: Sollwertkopplung

Die Leitwertkopplung gilt immer im Basiskoordinatensystem.

Zur Erstellung von Kurventabellen siehe Kapitel "Kurventabellen".

Syntax

LEADON (<Folgeachse>, <Leitachse>, <n>)

LEADOF (<Folgeachse>, <Leitachse>)

oder Ausschalten ohne Angabe der Leitachse:

LEADOF (<Folgeachse>)

Die Leitwertkopplung kann sowohl vom Teileprogramm als auch während der Bewegung aus Synchronaktionen heraus ein- und ausgeschaltet werden.

Bedeutung

LEADON:	Leitwertkopplung einschalten
LEADOF:	Leitwertkopplung ausschalten

<Folgeachse>:	Folgeachse
<Leitachse>:	Leitachse
<n>:	Kurventabellen-Nummer
\$SA_LEAD_TYPE:	Umschaltung zwischen Soll- und Istwertkopplung

Leitwertkopplung ausschalten, LEADOF

Mit dem Ausschalten der Leitwertkopplung wird die Folgeachse wieder zur normalen Kommandoachse!

Axiale Leitwertkopplung und verschiedene Betriebszustände, RESET

Abhängig von der Einstellung im Maschinendatum werden Leitwertkopplungen mit RESET ausgeschaltet.

Beispiel Leitwertkopplung aus Synchronaktion

Bei einer Pressenanlage soll eine herkömmliche mechanische Kopplung zwischen einer Leitachse (Stempelwelle) und Achsen eines Transfersystems aus Transferachsen und Hilfsachsen durch ein elektronisches Koppelsystem ersetzt werden.

Es demonstriert, wie bei einer Pressenanlage ein mechanisches Transfersystem durch ein elektronisches Transfersystem ersetzt wird. Die Kopplungs- und Entkopplungsvorgänge sind als **statische Synchronaktionen** realisiert.

Von der Leitachse LW (Stempelwelle) werden Transferachsen und Hilfsachsen als Folgeachsen über Kurventabellen definiert gesteuert.

Folgeachsen

X Vorschub- bzw. Längsachse
 YL Schließ- bzw. Querachse
 ZL Hubachse
 U Walzenvorschub, Hilfsachse
 V Richtkopf, Hilfsachse
 W Befettung, Hilfsachse

Aktionen

Als Aktionen treten in den Synchronaktionen z. B. auf:

- Einkoppeln, LEADON (<Folgeachse>, <Leitachse>, <Kurventabellen-Nummer>)
- Auskoppeln, LEADOF (<Folgeachse>, <Leitachse>)
- Istwertsetzen, PRESETON (<Achse>, <Wert>)
- Merker setzen, \$AC_MARKER[i]=<Wert>
- Kopplungsart: reeller/virtueller Leitwert
- Anfahren von Achspositionen, POS [<Achse>]=<Wert>

Bedingungen

Als Bedingungen werden digitale schnelle Eingänge, Echtzeitvariablen \$AC_MARKER und Positionsvergleiche, mit dem logischen Operator AND verknüpft, ausgewertet.

Hinweis

Im folgenden Beispiel wurden Zeilenwechsel, Einrückungen und **Fettsatz** ausschließlich dafür verwendet, die Lesbarkeit der Programmierung zu erhöhen. Für die Steuerung ist alles unter einer Zeilennummer stehende einzeilig.

Kommentar

Programmcode	Kommentar
	; Definiert sämtliche statische Synchronaktionen.
	; ****Marker rücksetzen
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 Kopplung Transfer EIN
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 Kopplung Walzenvorschub EIN
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 Kopplung Richtkopf EIN
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 Kopplung Befettung EIN
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 Kopplung AUS
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	

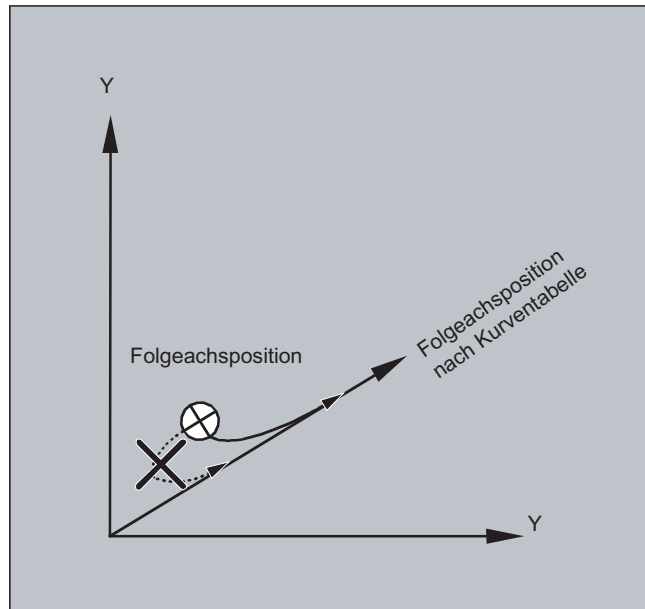
Beschreibung

Die Leitwertkopplung erfordert die Synchronisation von Leit- und Folgeachse. Diese Synchronisation kann nur erreicht werden, wenn die Folgeachse bei Einschalten der Leitwertkopplung innerhalb des Toleranzbereiches des aus der Kurventabelle berechneten Kurvenzugs steht.

Der Toleranzbereich für die Stellung der Folgeachse ist über Maschinendatum MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE definiert.

Befindet sich die Folgeachse mit dem Einschalten der Leitwertkopplung noch nicht an der entsprechenden Position, wird der Synchronlauf automatisch hergestellt, sobald sich der berechnete Positionssollwert für die Folgeachse der tatsächlichen Folgeachsposition nähert. Die Folgeachse wird dabei während des Synchronisationsvorganges in die Richtung

verfahren, die durch die Sollgeschwindigkeit der Folgeachse (berechnet aus Leitachsgeschwindigkeit und nach Kurventabelle CTAB) definiert ist.

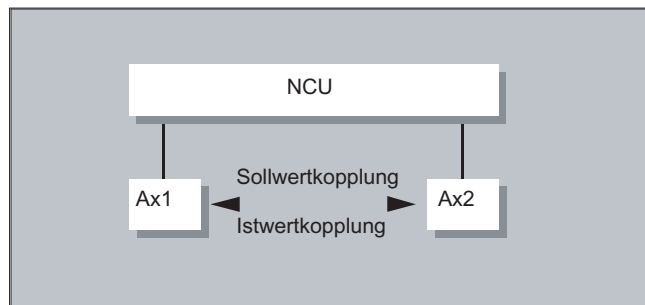


Kein Synchronlauf

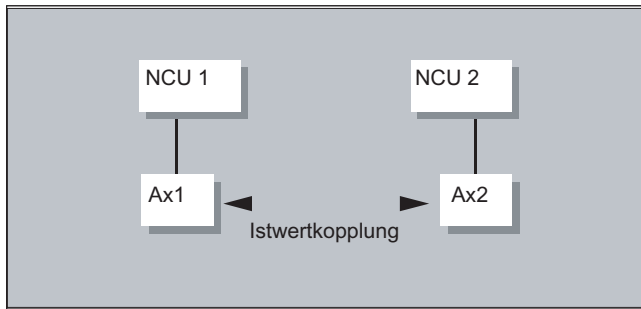
Entfernt sich die berechnete Folgeachssollposition mit Einschalten der Leitwertkopplung von der aktuellen Folgeachseposition, wird kein Synchronlauf hergestellt.

Ist- und Sollwertkopplung

Die Sollwertkopplung liefert im Vergleich zur Istwertkopplung einen besseren Synchronlauf zwischen Leit- und Folgeachse und ist deshalb standardmäßig voreingestellt.



Sollwertkopplung ist nur möglich, wenn Leit- und Folgeachse von derselben NCU interpoliert werden. Bei einer externen Leitachse kann die Folgeachse nur über Istwerte an die Leitachse gekoppelt werden.



Eine **Umschaltung** ist über das Settingdatum \$SA_LEAD_TYPE möglich.

Das Umschalten zwischen Ist- und Sollwertkopplung sollte immer bei Stillstand der Folgeachse erfolgen. Denn nur im Stillstand wird nach dem Umschalten neu synchronisiert.

Anwendungsbeispiel

Das Lesen der Istwerte kann bei großen Maschinenerschütterungen nicht fehlerfrei erfolgen. Beim Einsatz der Leitwertkopplung im Pressentferner kann es daher in den Arbeitsschritten mit größten Erschütterungen notwendig werden, von Istwertkopplung auf Sollwertkopplung umzuschalten.

Leitwertsimulation bei Sollwertkopplung

Über Maschinendatum lässt sich der Interpolator für die Leitachse vom Servo trennen. Damit können bei Sollwertkopplung Sollwerte ohne tatsächliche Bewegung der Leitachse erzeugt werden.

Die über Sollwertkopplung erzeugten Leitwerte sind zur Benutzung z. B. in Synchronaktionen aus folgenden Variablen lesbar:

- \$AA_LEAD_P Leitwert Position
- \$AA_LEAD_V Leitwert Geschwindigkeit

Leitwerte erzeugen

Leitwerte können wahlweise mit anderen selbst programmierten Verfahren erzeugt werden. Die so erzeugten Leitwerte werden in die Variable

- \$AA_LEAD_SP Leitwert Position
- \$AA_LEAD_SV Leitwert Geschwindigkeit

geschrieben und aus ihnen gelesen. Zur Benutzung dieser Variablen muss das Settingdatum \$SA_LEAD_TYPE = 2 gesetzt werden.

Status der Kopplung

Im NC-Teilprogramm können Sie den Kopplungsstatus mit folgender Systemvariablen abfragen:

\$AA_COUP_ACT[Achse]

0: Keine Kopplung aktiv

16: Leitwertkopplung aktiv

Status-Verwaltung bei Synchronaktionen

Schalt- und Koppelvorgänge werden über Echtzeitvariablen:

`$AC_MARKER[i] = n`
verwaltet mit:
i Merker-Nummer
n Statuswert

12.4 Elektronisches Getriebe (EG)

Mit Hilfe der Funktion "Elektronisches Getriebe" ist es möglich, die Bewegung einer **Folgeachse** nach linearem Bewegungssatz abhängig von bis zu fünf **Leitachsen** zu steuern. Die Zusammenhänge zwischen den Leitachsen und der Folgeachse sind je Leitachse durch den Koppelfaktor definiert.

Der berechnete Folgeachs-Bewegungsanteil wird aus den einzelnen Leitachsen-Bewegungsanteilen multipliziert mit den jeweiligen Koppelfaktoren durch Addition gebildet. Bei der Aktivierung eines EG-Achsverbundes kann die Synchronisation der Folgeachse auf eine definierte Position veranlasst werden. Ein Getriebeverband kann aus dem Teileprogramm:

- definiert,
- eingeschaltet,
- ausgeschaltet,
- gelöscht

werden.

Die Folgeachsbewegung kann wahlweise abgeleitet werden aus den

- Sollwerten der Leitachsen sowie den
- Istwerten der Leitachsen.

Als Erweiterung können auch nichtlineare Zusammenhänge zwischen den Leitachsen und der Folgeachse über **Kurventabellen** (siehe Kapitel Bahnverhalten) realisiert werden. Elektronische Getriebe können kaskadiert werden, d. h. die Folgeachse eines Elektronischen Getriebes kann Leitachse für ein weiteres Elektronisches Getriebe sein.

12.4.1 Elektronisches Getriebe definieren (EGDEF)

Ein EG-Achsverband wird durch die Angabe der Folgeachse und mindestens einer, jedoch höchstens fünf Leitachsen mit dem jeweiligen Kopplungstyp festgelegt.

Voraussetzung

Voraussetzung für eine EG-Achsverband-Definition:

Für die Folgeachse darf noch keine Achskopplung definiert sein (ggf. muss eine bestehende vorher mit EGDEL gelöscht werden).

Syntax

EGDEF (Folgeachse, Leitachse1, Kopplungstyp1, Leitachse2, Kopplungstyp2, ..)

Bedeutung

EGDEF:	Definition eines elektronischen Getriebes	
Folgeachse:	Achse, die von Leitachsen beeinflusst wird	
Leitachse1 Leitachse5	Achsen, die die Folgeachse beeinflussen	
Kopplungstyp1 Kopplungstyp5	Kopplungstyp Der Kopplungstyp muss nicht für alle Leitachsen gleich sein und ist daher für jede Leitachse einzeln anzugeben.	
	Wert:	Bedeutung:
	0	Die Folgeachse wird beeinflusst vom Istwert der entsprechenden Leitachse.
	1	Die Folgeachse wird beeinflusst vom Sollwert der entsprechenden Leitachse.

Hinweis

Die Koppelfaktoren werden bei der Definition des EG-Kopplungsverbandes mit Null vorbesetzt.

Hinweis

EGDEF löst Vorlaufstopp aus. Die Getriebedefinition mit EGDEF ist auch dann unverändert zu verwenden, wenn bei Systemen eine oder mehrere Leitachsen über **Kurventabelle** auf die Folgeachse einwirken.

Beispiel

Programmcode	Kommentar
EGDEF (C,B,1,Z,1,Y,1)	; Definition eines EG-Achsverbandes. Die Leitachsen B, Z, Y beeinflussen die Folgeachse C über den Sollwert.

12.4.2 Elektronisches Getriebe einschalten (EGON, EGONSYN, EGONSYNE)

Für das Einschalten eines EG-Achsverbandes existieren 3 Varianten.

Syntax

Variante 1:

Der EG-Achsverband wird ohneSynchronisation selektiv eingeschaltet mit:

EGON (FA, "Satzwechselmodus", LA1, Z1, N1, LA2, Z2, N2, ..., LA5, Z5, N5)

Variante 2:

Der EG-Achsverband wird mit Synchronisation selektiv eingeschaltet mit:

EGONSYN (FA, "Satzwechselmodus", SynPosFA, [, LAi, SynPosLAi, Zi, Ni])

Variante 3:

Der EG-Achsverband wird mit Synchronisation selektiv eingeschaltet und der Anfahrmodus vorgegeben mit:

EGONSYNE (FA, "Satzwechselmodus", SynPosFA, Anfahrmodus [, LAi, SynPosLAi, Zi, Ni])

Bedeutung**Variante 1:**

FA	Folgeachse	
Satzwechselmodus:	Folgende Modi können benutzt werden:	
	"NOC"	Satzwechsel erfolgt sofort
	"FINE"	Satzwechsel erfolgt bei "Synchronlauf fein"
	"COARSE"	Satzwechsel erfolgt bei "Synchronlauf grob"
	"IPOSTOP"	Satzwechsel erfolgt bei sollwertseitigem Synchronlauf
LA1, ... LA5	Leitachsen	
Z1, ... Z5	Zähler für den Koppelfaktor i	
N1, ... N5	Nenner für den Koppelfaktor i Koppelfaktor i = Zähler i/Nenner i	

Es dürfen nur die Leitachsen programmiert werden, die zuvor mit EGDEF spezifiziert worden sind. Es muss mindestens eine Leitachse programmiert werden.

Variante 2:

FA	Folgeachse	
Satzwechselmodus:	Folgende Modi können benutzt werden:	
	"NOC"	Satzwechsel erfolgt sofort
	"FINE"	Satzwechsel erfolgt bei "Synchronlauf fein"
	"COARSE"	Satzwechsel erfolgt bei "Synchronlauf grob"
	"IPOSTOP"	Satzwechsel erfolgt bei sollwertseitigem Synchronlauf
[, LAi, SynPosLAi, Zi, Ni]	(Eckige Klammern nicht schreiben) Mind. 1, max. 5 Folgen von:	
LA1, ... LA5	Leitachsen	
SynPosLAi	Synchronposition für die i. Leitachse	
Z1, ... Z5	Zähler für den Koppelfaktor i	
N1, ... N5	Nenner für den Koppelfaktor i Koppelfaktor i = Zähler i/Nenner i	

Es dürfen nur Leitachsen programmiert werden, die zuvor mit EGDEF spezifiziert worden sind. Durch die programmierten "Synchronpositionen" für die Folgeachse (SynPosFA) und für die

Leitachsen (SynPosLA) werden Positionen definiert, in denen der Koppelverband als *synchron* gilt. Sofern sich das elektronische Getriebe beim Einschalten nicht in synchronem Zustand befindet, fährt die Folgeachse auf ihre definierte Synchronposition.

Variante 3:

Die Parameter entsprechen denen der Variante 2 zuzüglich:

Anfahrmodus:	Folgende Modi können benutzt werden:	
	"NTGT"	Nächste Zahnücke zeitoptimiert anfahren
	"NTGP"	Nächste Zahnücke wegoptimiert anfahren
	"ACN"	Rundachse in negativer Drehrichtung verfahren absolut
	"ACP"	Rundachse in positiver Drehrichtung verfahren absolut
	"DCT"	Zeitoptimiert zur programmierten Synchronposition
	"DCP"	Wegoptimiert zur programmierten Synchronposition

Die Variante 3 hat nur Auswirkungen auf Modulo-Folgeachsen, die an Modulo-Leitachsen gekoppelt sind. Zeitoptimierung berücksichtigt die Geschwindigkeitsgrenzen der Folgeachse.

Weitere Informationen

Beschreibung der Einschaltvarianten

Variante 1:

Die Positionen der Leitachsen sowie der Folgeachse zum Zeitpunkt des Einschaltens werden gespeichert als "Synchronpositionen". Die "Synchronpositionen" können mit den Systemvariablen \$AA_EG_SYN gelesen werden.

Variante 2:

Wenn Moduloachsen im Koppelverband sind, werden ihre Positionswerte modulo reduziert. Damit ist gewährleistet, dass die nächstmögliche Synchronposition angefahren wird (sog. *relative Synchronisation*: z. B. die nächste Zahnücke). Wenn für die Folgeachse nicht "Freigabe Folgeachsüberlagerung" Nahtstellensignal DB(30 +Achsnnummer), DBX 26 Bit 4 gegeben ist, wird nicht auf die Synchronposition gefahren. Stattdessen wird das Programm beim EGONSYN-Satz angehalten und es wird der selbstlöschende Alarm 16771 gemeldet, solange bis das o.g. Signal gesetzt wird.

Variante 3:

Der Zahnabstand (Grad) ergibt sich aus: $360 * Zi/Ni$. Für den Fall, dass die Folgeachse zum Aufrufzeitpunkt steht, liefert wegoptimiert das gleiche Verhalten wie zeitoptimiert.

Bei bereits fahrender Folgeachse wird mit NTGP unabhängig von der aktuellen Geschwindigkeit der Folgeachse auf die nächste Zahnücke synchronisiert. Bei bereits fahrender Folgeachse wird mit NTGT abhängig von der aktuellen Geschwindigkeit der Folgeachse auf die nächste Zahnücke synchronisiert. Die Achse wird dazu ggf. auch abgebremst.

Kurventabellen

Wird für eine der Leitachsen eine **Kurventabelle** verwendet, so muss:

- | | |
|-----|---|
| Ni | der Nenner des Koppelfaktors linearer Kopplungen auf 0 gesetzt werden. (Nenner 0 wäre für lineare Kopplungen unzulässig). Nenner Null ist für die Steuerung das Kennzeichen, dass |
| Zi | als Nummer der zu verwendenden Kurventabelle interpretiert werden soll. Die Kurventabelle mit der angegebenen Nummer muss zum Einschaltzeitpunkt bereits definiert sein. |
| LAi | Die Angabe der Leitachse entspricht der Leitachsangabe bei Kopplung über Koppelfaktor (lineare Kopplung). |

Weitere Hinweise über die Nutzung von Kurventabellen und das Kaskadieren von Elektronischen Getrieben und deren Synchronisierung finden Sie in:

Literatur:

Funktionshandbuch Sonderfunktionen; Achskopplungen und ESR (M3), Kapitel "Mitschleppen und Leitwertkopplung".

Verhalten des Elektronischen Getriebes bei Power On, RESET, Betriebsartenwechsel, Suchlauf

- Nach Power On ist **keine** Kopplung aktiv.
- Aktive Kopplungen bleiben über RESET und Betriebsartenwechsel erhalten.
- Bei Satzsuchlauf werden Befehle zum Schalten, Löschen, Definieren des Elektronischen Getriebes nicht ausgeführt und nicht aufgesammelt, sondern übergangen.

Systemvariablen des Elektronischen Getriebes

Mit Hilfe der Systemvariablen des Elektronischen Getriebes kann das Teileprogramm aktuelle Zustände eines EG-Achsverbandes ermitteln und ggf. darauf reagieren.

Die Systemvariablen des Elektronischen Getriebes sind wie folgt gekennzeichnet:

\$AA_EG_ ...

oder

\$VA_EG_ ...

Literatur:

Handbuch der Systemvariablen

12.4.3 Elektronisches Getriebe ausschalten (EGOFS, EGOFC)

Für das Ausschalten eines aktiven EG-Achsverbandes existieren 3 Varianten.

Programmierung

Variante 1:

Syntax	Bedeutung
EGOFS (Folgeachse)	Das elektronische Getriebe wird ausgeschaltet. Die Folgeachse wird zum Stillstand abgebremst. Der Aufruf löst Vorlaufstopp aus.

Variante 2:

Syntax	Bedeutung
EGOFS (Folgeachse, Leitachse1, ..., Leitachse5)	Diese Parametrierung des Befehls erlaubt selektiv den Einfluss einzelner Leitachsen auf die Bewegung der Folgeachse zu beseitigen.

Es muss wenigstens eine Leitachse angegeben werden. Der Einfluss der angegebenen Leitachsen auf die Folgeachse wird gezielt ausgeschaltet. Der Aufruf löst Vorlaufstopp aus. Verbleiben noch aktive Leitachsen, so läuft die Folgeachse unter deren Einfluss weiter. Sind alle Leitachseneinflüsse auf diese Weise ausgeschaltet, so wird die Folgeachse zum Stillstand abgebremst.

Variante 3:

Syntax	Bedeutung
EGOFC (Folgespindel)	Das elektronische Getriebe wird ausgeschaltet. Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/Geschwindigkeit weiter. Der Aufruf löst Vorlaufstopp aus.

Hinweis

Diese Variante ist nur für Spindeln erlaubt.

12.4.4 Definition eines Elektronischen Getriebes löschen (EGDEL)

Ein EG-Achsverband muss ausgeschaltet sein, bevor seine Definition gelöscht werden kann.

Programmierung

Syntax	Bedeutung
EGDEL (Folgeachse)	Die Kopplungsdefinition des Achsverbandes wird gelöscht. Es wird bis zum Erreichen der maximalen Anzahl gleichzeitig aktivierter Achsverbände wieder möglich, weitere Achsverbände mit EGDEF neu zu definieren. Der Aufruf löst Vorlaufstopp aus.

12.4.5 Umdrehungsvorschub (G95) / Elektronisches Getriebe (FPR)

Mit dem FPR-Befehl kann auch die Folgeachse eines Elektronischen Getriebes als vorschubbestimmende Achse des Umdrehungsvorschubes angegeben werden. Für diesen Fall gilt folgendes Verhalten:

- Der Vorschub ist abhängig von der Sollgeschwindigkeit der Folgeachse des Elektronischen Getriebes.
- Die Sollgeschwindigkeit wird berechnet aus den Geschwindigkeiten der Leitspindeln und Modulo-Leitachsen (die nicht Bahnachsen sind) und deren zugeordneten Koppelfaktoren.
- Geschwindigkeitsanteile von linearen bzw. nicht Modulo-Leitachsen und überlagerte Bewegungen der Folgeachse werden nicht berücksichtigt.

12.5 Synchronspindel

Im Synchronbetrieb gibt es eine Leitspindel (LS) und eine Folgespindel (FS), das sog. **Synchronspindel**paar. Die Folgespindel folgt bei aktiver Kopplung (Synchronbetrieb) den Bewegungen der Leitspindel entsprechend dem festgelegten Funktionszusammenhang.

Die Synchronspindelpaare lassen sich für jede Maschine sowohl mit Hilfe von kanalspezifischen Maschinendaten fest projektieren oder über das CNC-Teileprogramm anwendungsspezifisch definieren. Je NC-Kanal sind bis zu 2 Synchronspindelpaare gleichzeitig betreibbar.

Die Kopplung kann aus dem Teileprogramm

- definiert bzw. geändert
- eingeschaltet
- ausgeschaltet
- gelöscht

werden.

Darüber hinaus kann abhängig vom Softwarestand

- auf die Synchronlaufbedingung gewartet
- das Satzwechselverhalten verändert
- die Kopplungsart entweder Sollwertkopplung oder Istwertkopplung ausgewählt oder der Winkelversatz zwischen Leit- und Folgespindel vorgegeben
- beim Einschalten der Kopplung eine vorhergehende Programmierung der Folgespindel übernommen
- entweder eine gemessene oder eine bereits bekannte Synchronlaufabweichung korrigiert werden.

12.5.1 Synchronspindel: Programmierung (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

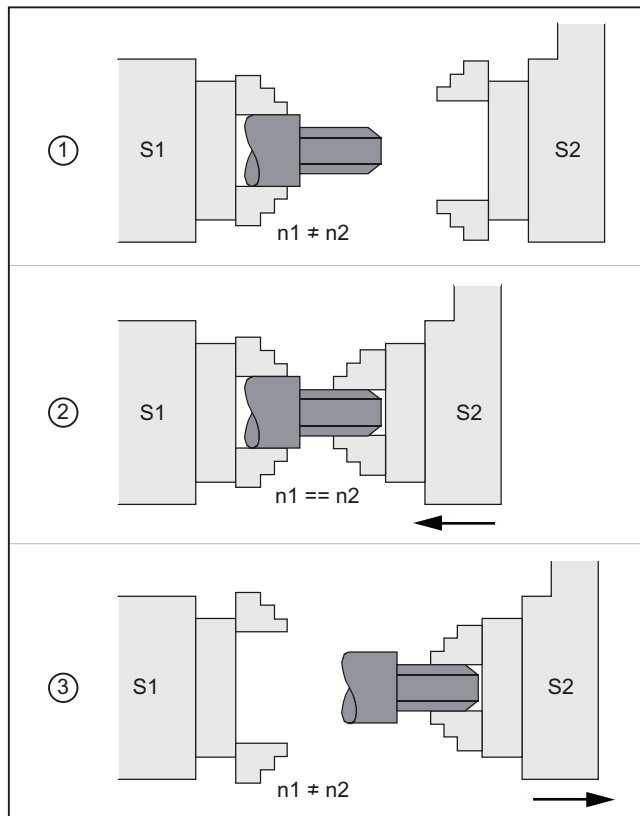
Die Funktion "Synchronspindel" ermöglicht ein drehzahlsynchrones Verfahren von Folge- (FS) und Leitspindel (LS) mit programmierbarem Übersetzungsverhältnis.

Die Funktion bietet folgende Modi:

- Drehzahlsynchronität ($n_{FS} = n_{LS}$)
- Lagesynchronität ($\phi_{FS} = \phi_{LS}$)
- Lagesynchronität mit Winkelversatz ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

Anwendungsbeispiele:

- Fliegende Werkstückübergabe z.B. zur Rückseitenbearbeitung, Übersetzungsverhältnis: 1:1

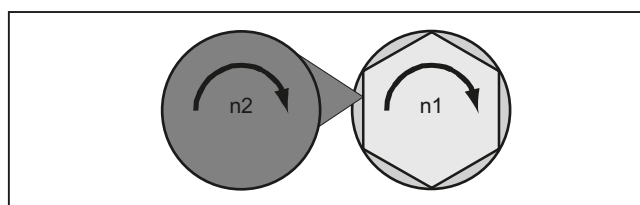


① Drehzahl synchronisieren

② Werkstück übergeben

③ Rückseite bearbeiten

- Mehrkantbearbeitung (Polygondrehen), Drehzyhsynchronität, Übersetzungsverhältnis: $n_1:n_2$



Syntax

COUPDEF (<FS>, <LS>, <ZFS>, <NLS>, <Satzwechsel>, <Koppelart>)
 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>, <Satzwechsel>, <LS>, <Satzwechsel>)

Hinweis

Verkürzte Schreibweise

Bei den Anweisungen COUPOF, COUPOFS, COUNPRES und COUPDEL ist eine verkürzte Schreibweise ohne Angabe der Leitspindel möglich.

Bedeutung

COUPDEF:	Kopplung anwenderspezifisch definieren/ändern
COUPON:	Kopplung einschalten. Ausgehend von der aktuellen Drehzahl synchronisiert sich die Folgespindel auf die Leitspindel
COUPONC:	Kopplung beim Einschalten mit vorhergehender Programmierung von M3 S... oder M4 S... übernehmen. Eine Differenzdrehzahl der Folgespindel wird sofort übernommen.
COUPOF:	Kopplung ausschalten. <ul style="list-style-type: none"> mit sofortigem Satzwechsel: COUPOF (<S2>, <S1>) Satzwechsel erst nach Überfahren der Ausschaltposition(en) <POSFS> bzw. <POSLS>: COUPOF (<S2>, <S1>, <POSFS>) COUPOF (<S2>, <S1>, <POSFS>, <POSLS>)
COUPOFS:	Ausschalten einer Kopplung mit Stopp der Folgespindel. Satzwechsel schnellstmöglich mit sofortigen Satzwechsel: COUPOFS (<S2>, <S1>) Satzwechsel erst nach Überfahren der Ausschaltposition: COUPOFS (<S2>, <S1>, <POSFS>)
COUNPRES:	Kopplungsparameter zurücksetzen auf projektierte MD und SD
COUPDEL:	Anwenderdefinierte Kopplung löschen
WAITC:	Warten Synchronlaufbedingung (NOC werden auf IPO bei Satzwechsel aufgehoben)
<FS>:	Bezeichnung der Folgespindel
Optionale Parameter:	
<LS>:	Bezeichnung der Leitspindel Angabe mit Spindelnummer: z. B. S2, S1
<ZFS>, <NLS>:	Übersetzungsverhältnis zwischen FS und LS. <ZFS> / <NLS> = Zähler / Nenner Voreinstellung: <ZFS> / <NLS> = 1.0 ; Angabe des Nenners optional

<Satzwechsel>:	Satzwechselverhalten	
	Der Satzwechsel erfolgt:	
	"NOC"	sofort
	"FINE"	mit Erreichen von "Synchronlauf fein"
	"COARSE"	mit Erreichen von "Synchronlauf grob"
	"IPOSTOP"	mit Erreichen von IPOSTOP, d. h. nach sollwertseitigem Synchronlauf (Voreinstellung)
Das Satzwechselverhalten ist modal wirksam.		
<Koppelart>:	Kopplungsart: Kopplung zwischen FS und LS	
	"DV"	Sollwertkopplung (Voreinstellung)
	"AV"	Istwertkopplung
	"VV"	Geschwindigkeitskopplung
	Die Kopplungsart ist modal wirksam.	
<POSFS>:	Winkelversatz zwischen Leit- und Folgespindel	
	Wertebereich:	0°... 359,999°
<POSFS>, <POSLS>:	Ausschaltpositionen von Folge- und Leitspindel	
	"Der Satzwechsel wird nach überfahren der POS _{FS} , POS _{LS} freigegeben"	
	Wertebereich:	0°... 359,999°

Beispiele

Arbeiten mit Leit- und Folgespindel

Programmcode	Kommentar
	Leitspindel = Masterspindel = Spindel 1
	Folgespindel = Spindel 2
N05 M3 S3000 M2=4 S2=500	Leitspindel dreht mit 3000 U/min, Folgespindel dreht mit 500 U/min.
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	Definition der Kopplung (kann auch projiziert werden).
...	
N70 SPCON	Leitspindel in Lageregelung nehmen (Sollwertk.).
N75 SPCON(2)	Folgespindel in Lageregelung nehmen.
N80 COUPON(S2,S1,45)	Fliegend auf Offsetposition = 45 Grad einkoppeln.
...	
N200 FA[S2]=100	Positioniergeschwindigkeit = 100 grd/min
N205 SPOS[2]=IC(-90)	90 Grad überlagert in negative Richtung fahren.
N210 WAITC(S2,"Fine")	Warten auf Synchronlauf "fein".
N212 G1 X... Y... F...	Bearbeitung
...	
N215 SPOS[2]=IC(180)	180 Grad überlagert in positive Richtung fahren.
N220 G4 S50	Verweilzeit = 50 Umdrehungen der Masterspindel
N225 FA[S2]=0	Projektierte Geschw. (MD) aktivieren.
N230 SPOS[2]=IC(-7200)	20 Umdrehungen. Mit projektierte Geschwindigkeit in negative Richtung fahren.

Programmcode	Kommentar
...	
N350 COUPOF(S2,S1)	Fliegend auskoppeln, S=S2=3000
N355 SPOSA[2]=0	FS bei Null Grad stoppen.
N360 G0 X0 Y0	
N365 WAITS(2)	Warten auf Spindel 2.
N370 M5	FS stoppen.
N375 M30	

Programmierung einer Differenzdrehzahl

Programmcode	Kommentar
	Leitspindel = Masterspindel = Spindel 1
	Folgespindel = Spindel 2
N01 M3 S500	Leitspindel dreht mit 500 U/min.
N02 M2=3 S2=300	Folgespindel dreht mit 300 U/min.
...	
N10 G4 F1	Verweilzeit der Masterspindel.
N15 COUPDEF (S2,S1,-1)	Koppelfaktor mit Übersetzungsverhältnis -1:1
N20 COUPON (S2,S1)	Kopplung aktivieren. Die Drehzahl der Folgespindel ergibt sich aus der Drehzahl der Leitspindel und dem Koppelfaktor.
...	
N26 M2=3 S2=100	Programmierung einer Differenzdrehzahl.

Beispiele der Übernahme einer Bewegung zur Differenzdrehzahl

1. Kopplung bei vorhergehender Programmierung der Folgespindel mit COUPON einschalten

Programmcode	Kommentar
	Leitspindel = Masterspindel = Spindel 1
	Folgespindel = Spindel 2
N05 M3 S100 M2=3 S2=200	Leitspindel dreht mit 100 U/min, Folgespindel mit 200 U/min.
N10 G4 F5	Verweilzeit = 5 Sekunden der Masterspindel
N15 COUPDEF(S2,S1,1)	Übersetzungsverhältnis FS zu LS ist 1,0 (Voreinstellung).
N20 COUPON(S2,S1)	Fliegend auf Leitspindel einkoppeln.
N10 G4 F5	Folgespindel dreht mit 100 U/min.

2. Kopplung bei vorhergehender Programmierung der Folgespindel mit COUPONC einschalten

Programmcode	Kommentar
	Leitspindel = Masterspindel = Spindel 1
	Folgespindel = Spindel 2
N05 M3 S100 M2=3 S2=200	Leitspindel dreht mit 100 U/min, Folgespindel mit 200 U/min.
N10 G4 F5	Verweilzeit = 5 Sekunden der Masterspindel

Programmcode	Kommentar
N15 COUPDEF(S2,S1,1)	Übersetzungsverhältnis FS zu LS ist 1,0 (Voreinstellung).
N20 COUPONC(S2,S1)	Fliegend auf Leitspindel einkoppeln und vorhergehende Drehzahl zu S2 übernehmen.
N10 G4 F5	S2 dreht mit 100U/min + 200U/min = 300U/min

3. Kopplung bei stehender Folgespindel mit COUPON einschalten

Programmcode	Kommentar
	Leitspindel = Masterspindel = Spindel 1
	Folgespindel = Spindel 2
N05 SPOS=10 SPOS[2]=20	Folgespindel S2 im Positionierbetrieb.
N15 COUPDEF(S2,S1,1)	Übersetzungsverhältnis FS zu LS ist 1,0 (Voreinstellung).
N20 COUPON(S2,S1)	Fliegend auf Leitspindel einkoppeln.
N10 G4 F1	Kopplung wird geschlossen, S2 bleibt auf 20 Grad stehen.

4. Kopplung bei stehender Folgespindel mit COUPONC einschalten

Hinweis

Positionier- oder Achsbetrieb

Befindet sich die Folgespindel vor dem Einkoppeln im Positionier- oder Achsbetrieb, dann verhält sich die Folgespindel bei COUPON (<FS>, <LS>) und COUPONC (<FS>, <LS>) gleich.

Hinweis

Leitspindel und Achsbetrieb

Befindet sich die Leitspindel vor der Definition der Kopplung im Achsbetrieb, wirkt auch nach dem Einschalten der Kopplung der Geschwindigkeitsgrenzwert aus Maschinendatum:

MD32000 \$MA_MAX_AX_VELO (maximale Achsgeschwindigkeit)

Zur Vermeidung dieses Verhaltens muss die Achse vor der Definition der Kopplung in den Spindelbetrieb (M3 S... oder M4 S...) geschaltet werden.

Weitere Informationen

Projektierte Kopplung

Bei der projektierten Kopplung werden LS und FS über Maschinendaten festgelegt. Die projektierten Spindeln können im Teileprogramm nicht verändert werden. Die Parametrierung der Kopplung kann mit COUPDEF im Teileprogramm erfolgen (Voraussetzung: kein Schreibschutz festgelegt).

Anwenderdefinierte Kopplung

Mit COUPDEF kann eine Kopplung im Teileprogramm neu definiert oder verändert werden. Ist bereits eine Kopplung aktiv, muss diese vor der Definition einer neuen Kopplung zuerst mit COUPDEL gelöscht werden.

Eine Kopplung wird vollständig definiert durch:

`COUPDEF (<FS>, <LS>, <ÜFS>, <ÜLS>, Satzwechselverhalten, Koppelart)`

Folgespindel (FS) und Leitspindel (LS)

Mit den Achsnamen für die FS und LS wird die Kopplung eindeutig bestimmt. Die Achsnamen müssen mit jeder Anweisung `COUPDEF` programmiert werden. Die anderen Kopplungsparameter sind modal wirksam und müssen nur programmiert werden, wenn sie geändert werden.

Beispiel:

`COUPDEF (S2, S1)`

Übersetzungsverhältnis

Das Übersetzungsverhältnis wird als Drehzahlverhältnis zwischen FS und LS angegeben:

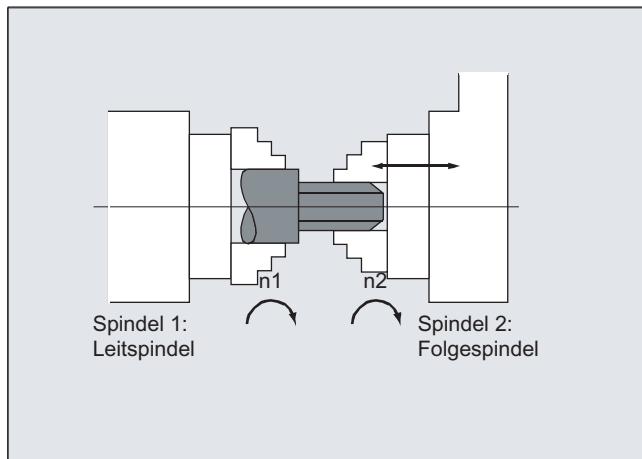
Folgespindel / Leitspindel = Zähler / Nenner

Der Zähler muss programmiert werden. Der Nenner muss nicht programmiert werden. Für den Nenner wird dann der Defaultwert 1.0 gesetzt.

Beispiel:

Folgespindel S2 und Leitspindel S1, Übersetzungsverhältnis = 1 / 1

`COUPDEF (S2, S1, 1.0)`



Hinweis

Das Übersetzungsverhältnis kann auch bei eingeschalteter Kopplung und drehenden Spindeln verändert werden.

Satzwechselverhalten NOC, FINE, COARSE, IPOSTOP

Bei der Programmierung des Satzwechselverhaltens ist folgende verkürzte Schreibweise möglich:

- "NO": sofort (Voreinstellung)
- "FI": mit Erreichen von "Synchronlauf fein"

- "CO": mit Erreichen von "Synchronlauf grob"
- "IP": mit Erreichen von IPOSTOP, d. h. nach sollwertseitigem Synchronlauf

Kopplungsart

Hinweis

Die Kopplungsart darf nur bei ausgeschalteter Kopplung verändert werden.

Synchronbetrieb einschalten COUPON, <POSFS>

- Einschalten der Kopplung mit beliebigem Winkelversatz zwischen LS und FS:
 - COUPON (S2, S1)
 - COUPON (S2)
- Einschalten der Kopplung mit Winkelversatz <POSFS>
<POSFS> bezieht sich auf die 0°-Position der Leitspindel in positiver Drehrichtung
Wertebereich <POSFS>: 0°... 359,999°
 - COUPON (S2, S1, 30)

Hinweis

Der Winkelversatz kann auch bei aktiver Kopplung geändert werden.

Positionieren der Folgespindel

Auch bei eingeschalteter Synchronspindelkopplung lassen sich FS unabhängig von der LS im Bereich $\pm 180^\circ$ positionieren.

- Spindelpositionierung der FS mit SPOS
Beispiel: SPOS [2] = IC (-90)
Weitere Informationen zu SPOS finden sich in:
Literatur:
Programmierhandbuch Grundlagen

Differenzdrehzahl

Eine Differenzdrehzahl entsteht im Drehzahlsteuerbetrieb und aktiver Synchronspindelkopplung durch vorzeichenbehaftete Überlagerung einer FS-Drehzahl aufgrund LS-Bewegung und einer FS-Drehzahl aufgrund Spindelprogrammierung:

- Synchronspindelkopplung mit COUPONC
 - $S<FS>=<Drehzahl> [M<FS>=<Drehrichtung>]$
-

Hinweis

Randbedingungen

- Mit der Drehrichtung M3 / M4 muss auch die Drehzahl S . . . neu programmiert werden.
- Die Überlagern einer Spindeldrehzahl (M<Drehrichtung> S<FS>) durch die LS-Bewegung bei Synchronspindelkopplung COUPONC wird nur wirksam, wenn die Überlagerung freigegeben ist.
- Die Dynamik der Leitspindel muss so weit eingeschränkt werden, dass bei einer Überlagerung der Folgespindel deren Dynamikgrenzwerte nicht überschritten werden.

Weitere Informationen zur Differenzdrehzahl siehe:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Synchronspindel (S3)

Geschwindigkeit, Beschleunigung: FA, ACC, OVRA, VELOLIMA

Axiale Geschwindigkeit und Beschleunigung einer Folgespindeln sind programmierbar mit:

- FA [SPI (S<n>)] bzw. FA [S<n>] (axiale Geschwindigkeit)
- ACC [SPI (S<n>)] bzw. ACC [S<n>] (axiale Beschleunigung)
- OVRA [SPI (S<n>)] bzw. OVRA [S<n>] (axialer Override)
- VELOLIMA [SPI (S<n>)] bzw. VELOLIMA [S<n>] (axiale Geschwindigkeitsüberhöhung bzw. -reduktion)

Mit <n> = 1, 2, 3, ... (Spindelnummer der Folgespindeln)

Literatur:

Programmierhandbuch Grundlagen

Hinweis

Eine Reduktion oder Überhöhung des maximalen axialen Rucks ist bei Spindeln nicht wirksam.

Weitere Informationen zur axialen Dynamik finden sich in:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Rundachsen (R2)

Programmierbares Satzwechselverhalten WAITC

Mit WAITC kann das Satzwechselverhalten, z. B. nach Änderung von Kopplungsparametern oder Positioniervorgängen, mit unterschiedlichen Synchronlaufbedingungen (grob, fein, IPOSTOP) vorgegeben werden. Sind keine Synchronlaufbedingungen angegeben, gilt das bei der Definition COUPDEF angegebene Satzwechselverhalten.

Beispiele

- Warten auf das Erreichen der Synchronlaufbedingung FINE bei Folgespindel S2 und COARSE bei Folgespindel S4: WAITC (S2, "FINE", S4, "COARSE")
- Warten auf das Erreichen der Synchronlaufbedingung entsprechend COUPDEF: WAITC ()

Kopplung ausschalten COUPOF

Mit COUPOF kann das Ausschaltverhalten der Kopplung vorgegeben werden:

- Ausschalten der Kopplung mit sofortigem Satzwechsel:
 - COUPOF (S2, S1) (mit Angabe der Leitspindel)
 - COUPOF (S2) (ohne Angabe der Leitspindel)
- Ausschalten der Kopplung nach Überfahren von Ausschaltpositionen. Der Satzwechsel erfolgt nach dem Überfahren der Ausschaltpositionen.
 - COUPOF (S2, S1, 150) (Ausschaltposition FS: 150°)
 - COUPOF (S2, S1, 150, 30) (Ausschaltposition FS: 150°, LS: 30°)

Kopplung ausschalten mit Stopp der Folgespindel COUPOFS

Mit COUPOFS kann das Ausschaltverhalten der Kopplung mit Stopp der Folgespindel vorgegeben werden:

- Ausschalten der Kopplung mit Stopp der Folgespindel und sofortigem Satzwechsel:
 - COUPOFS (S2, S1) (mit Angabe der Leitspindel)
 - COUPOFS (S2) (ohne Angabe der Leitspindel)
- Ausschalten der Kopplung nach Überfahren von Ausschaltpositionen mit Stopp der Folgespindel. Der Satzwechsel erfolgt nach dem Überfahren der Ausschaltpositionen.
 - COUPOFS (S2, S1, 150) (Ausschaltposition FS: 150°)

Kopplungen löschen COUPDEL

Mit COUPDEL wird die Kopplung gelöscht:

- COUPDEL (S2, S1) (mit Angabe der Leitspindel)
- COUPDEL (S2) (ohne Angabe der Leitspindel)

Kopplungsparameter zurücksetzen COUPRES

Mit COUPRES werden die in den Maschinen- und Settingdaten parametrisierten Werte der Kopplung aktiviert:

- COUPRES (S2, S1) (mit Angabe der Leitspindel)
- COUPRES (S2) (ohne Angabe der Leitspindel)

Systemvariablen

- Aktueller Kopplungszustand der Folgespindel
Der aktuelle Kopplungszustand einer Folgespindel kann bitcodiert gelesen werden über:
<Wert> = \$AA_COUP_ACT[<FS>]

Bit	<Wert>	Bedeutung
-	0	keine Kopplung aktiv
2	4	Synchronspindelkopplung aktiv

Hinweis

- Alle andere Werte beziehen sich auf Achsbetrieb
- Ist die Spindel eine Folgespindel von mehreren Kopplungen, wird als Wert der Kopplungszustand aller Kopplungen als Summenzustand zurückgegeben.

- **Aktueller Winkelversatz**
Der aktuelle Winkelversatz der Folgespindel zur Leitspindel kann gelesen werden über:
 - \$AA_COUP_OFFS [<FS>] (Sollwertseitiger Winkelversatz)
 - \$VA_COUP_OFFS [<FS>] (Istwertseitiger Winkelversatz)

Anwendungsbeispiel

Korrektur der Winkelversatzdifferenz im NC-Programm nach Aufheben des Nachführbetriebs:

Winkelversatzdifferenz = Programmierter Winkelversatz - Systemvariable

Literatur

Ausführliche Informationen zu Systemvariablen finden sich in:

Listenhandbuch Systemvariablen

12.6 Generische Kopplung (CP...)

Die "Generische Kopplung" ist eine allgemeine Kopplungsfunktion, in der alle Kopplungseigenschaften der bestehenden Kopplungsarten (Mitschleppen, Leitwertkopplung, Elektronisches Getriebe und Synchronspindel) zusammengefasst sind.

Die Funktion ermöglicht eine flexible Art der Programmierung:

- Der Anwender kann die für seine Applikation notwendigen Kopplungseigenschaften frei auswählen (Baukastenprinzip).
- Jede Kopplungseigenschaft ist einzeln programmierbar.
- Die Kopplungseigenschaften einer definierten Kopplung (z. B. Koppelfaktor) sind änderbar.
- Eine spätere Nutzung weiterer Kopplungseigenschaften ist möglich.
- Das Koordinatenbezugssystem der Folgeachse (Basiskoordinatensystem oder Maschinenkoordinatensystem) ist programmierbar.
- Bestimmte Kopplungseigenschaften können auch in Synchronaktionen programmiert werden.

Literatur: Funktionshandbuch Synchronaktionen

Hinweis

Die bisherigen Kopplungsaufrufe für das Mitschleppen (TRAIL*), Leitwertkopplung (LEAD*), Elektronisches Getriebe (EG*) und Synchronspindel (COUP*) werden über Anpasszyklen weiterhin unterstützt.

Übersicht aller Schlüsselwörter und Kopplungseigenschaften

Die folgende Tabelle stellt eine Übersicht aller Schlüsselwörter der Generischen Kopplung und der damit programmierbaren Kopplungseigenschaften dar:

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Syntax
CPDEF	Anlegen eines Koppelmoduls	CPDEF= (<FAx>)
CPDEL	Löschen eines Koppelmoduls	CPDEL= (<FAx>)
CPLA	Definition einer Leitachse	CPLA [<FAx>] = (<LAx>)
CPLDEF	Definition einer Leitachse und Anlegen eines Koppelmoduls (auch möglich mit CPDEF + CPLA)	CPLDEF [<FAx>] = (<LAx>) oder CPDEF= (<FAx>) CPLA [<FAx>] = (<LAx>)
CPLDEL	Löschen einer Leitachse eines Koppelmoduls (auch möglich mit CPDEL + CPLA)	CPLDEL [<FAx>] = (<LAx>) oder CPDEL= (<FAx>) CPLA [<FAx>] = (<LAx>)
CPON	Einschalten eines Koppelmoduls	CPON= (<FAx>)
CPOF	Ausschalten eines Koppelmoduls	CPOF= (<FAx>)
CPLON	Einschalten einer Leitachse eines Koppelmoduls	CPLON [<FAx>] = <LAx>
CPLOF	Ausschalten einer Leitachse eines Koppelmoduls	CPLOF [<FAx>] = <LAx>
CPLNUM	Zähler des Koppelfaktors	CPLNUM [FAx, LAx] = <Wert>
CPLDEN	Nenner des Koppelfaktors	CPLDEN [FAx, LAx] = <Wert>
CPLCTID	Nummer der Kurventabelle	CPLCTID [FAx, LAx] = <Wert>
CPLSETVAL	Kopplungsbezug	CPLSETVAL [FAx, LAx] = "<Kopplungsbezug>" "<Kopplungsbezug>": "CMDPOS" Sollwertkopplung "CMDVEL" Geschwindigkeitskopplung "ACTPOS" Istwertkopplung
CPFRS	Koordinatenbezugssystem	CPFRS [FAx] = "<Koordinatenbezug>" "<Koordinatenbezug>": "BCS" Basiskoordinatensystem "MCS" Maschinenkoordinatensystem

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Syntax		
CPBC	Satzwechselkriterium	CPBC [FAx]="<Satzwechselkriterium>"		
		"<Satzwechselkriterium>":	"NOC"	Satzwechsel erfolgt unabhängig vom Kopplungszustand.
			"IPOSTOP"	Satzwechsel erfolgt beim sollwertseitigen Synchronlauf.
			"COARSE"	Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Grob".
		"FINE"	Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Fein".	
CPFPOS + CPON	Synchronposition der Folgeachse beim Einschalten	CPON=FAx CPFPOS [FAx]=<Wert>		
CPLPOS + CPON	Synchronposition der Leitachse beim Einschalten	CPLPOS [FAx, LAx]=<Wert>		

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Syntax		
CPFMSON	Synchronisationsmodus	CPFMSON[FAx]="<Synchronisationsmodus>"		
		"<Synchronisation smodus>":	"CFAST"	Die Kopplung wird zeitopti- miert geschlossen.
			"CCOARSE"	Die Kopplung wird erst ein- geschaltet, wenn sich die gemäß Koppelgesetz ge- forderte Folgeachsposition im Bereich der aktuellen Folgeachsposition befindet.
			"NTGT"	Die nächste Zahnücke wird zeitoptimiert angefah- ren.
			"NTGP"	Die nächste Zahnücke wird wegoptimiert angefah- ren.
			"NRGT"	Das nächste Segment ge- mäß Verhältnis aus Gang- zahl zu Zähnezahzahl wird zeit- optimiert angefahren.
			"NRGP"	Das nächste Segment ge- mäß Verhältnis aus Gang- zahl zu Zähnezahzahl wird weg- optimiert angefahren.
			"ACN"	Nur bei Rundachsen! Die Rundachse verfährt in negativer Achsrichtung die Synchronposition an. Die Synchronisation erfolgt so- fort.
			"ACP"	Nur bei Rundachsen! Die Rundachse verfährt in positiver Achsrichtung die Synchronposition an. Die Synchronisation erfolgt so- fort.
			"DCT"	Nur bei Rundachsen! Die Rundachse verfährt zeit- optimiert zur pen Synchron- position an. Die Synchroni- sation erfolgt sofort.
"DCP"	Nur bei Rundachsen! Die Rundachse verfährt wegoptimiert zur program- mierten Synchronposition an. Die Synchronisation er- folgt sofort.			

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Syntax		
CPFMON	Verhalten der Folgeachse beim Einschalten	CPFMON [FAx] = "<Einschaltverhalten>"		
		"<Einschaltverhalten>":	"STOP"	Nur bei Spindeln! Eine aktive Bewegung der Folgespindel wird vor dem Einschalten gestoppt.
		"CONT"	Nur bei Spindeln und Hauptlaufachsen! Die aktuelle Bewegung der Folgeachse/-spindel wird in die Kopplung als Startbewegung übernommen.	
CPFMOF	Verhalten der Folgeachse beim vollständigen Ausschalten	CPFMOF [FAx] = "<Ausschaltverhalten>"		
		"<Ausschaltverhalten>":	"STOP"	Stopp der Folgeachse/-spindel. Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung geöffnet
CPFMOF			"CONT"	Nur bei Spindeln und Hauptlaufachsen! Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/ Geschwindigkeit weiter.
			CPFPOS + CPOF	Ausschaltposition der Folgeachse beim Ausschalten

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Syntax
CPMRESET	Kopplungsverhalten bei RESET	CPMRESET [FAx]="<Reset-Verhalten>"
		"<Reset-Verhalten>":
		"NONE"
		Der aktuelle Zustand der Kopplung bleibt erhalten.
		"ON"
		Ist das entsprechende Koppelmodul angelegt, so wird die Kopplung eingeschaltet. Es werden alle definierten Leitachsbeziehungen aktiviert. Dies erfolgt auch, wenn bereits alle oder Teile dieser Leitachsbeziehungen aktiv sind, d. h. auch bei einer vollständig aktivierten Kopplung erfolgt eine Neusynchronisation.
		"OF"
		Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden.
		"OFC"
		Nur bei Spindeln möglich! Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/ Geschwindigkeit weiter. Die Kopplung wird ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden.
		"DEL"
		Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung deaktiviert und anschließend gelöscht.
		"DELC"
		Nur bei Spindeln möglich! Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Syntax		
				Geschwindigkeit weiter. Die Kopplung wird deaktiviert und anschließend gelöscht.
CPMSTART	Kopplungsverhalten beim Teilprogrammstart	CPMSTART [FAx] = "<Start-Verhalten>"		
		"<Start-Verhalten>":	"NONE"	Der aktuelle Zustand der Kopplung bleibt erhalten.
			"ON"	Kopplung eingeschaltet. Es werden alle definierten Leitachsbeziehungen aktiviert. Dies erfolgt auch, wenn bereits alle oder Teile dieser Leitachsbeziehungen aktiv sind, d. h. auch bei einer vollständig aktivierten Kopplung erfolgt eine Neusynchronisation.
			"OF"	Die Kopplung wird ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden.
		"DEL"	Die Kopplung wird deaktiviert und anschließend gelöscht.	
CPMPRT	Kopplungsverhalten beim Teilprogrammstart unter Suchlauf via Programmtest	CPMPRT [FAx] = "<Start-Verhalten>"		
		"<Start-Verhalten>":	siehe CPMSTART	
CPLINTR	Verschiebewert für den Eingangswert einer Leitachse	CPLINTR [FAx, LAx] = <Wert>		
CPLINSC	Skalierfaktor für den Eingangswert einer Leitachse	CPLINSC [FAx, LAx] = <Wert>		
CPLOUTTR	Verschiebewert für den Ausgangswert einer Kopplung	CPLOUTTR [FAx, LAx] = <Wert>		
CPLOUTSC	Skalierfaktor für den Ausgangswert einer Kopplung	CPLOUTSC [FAx, LAx] = <Wert>		
CPSYNCOF	Schwellwert für den Positionssynchronlauf "Grob"	CPSYNCOF [FAx] = <Wert>		
CPSYNFIP	Schwellwert für den Positionssynchronlauf "Fein"	CPSYNFIP [FAx] = <Wert>		
CPSYNCOF2	Zweiter Schwellwert für den Positionssynchronlauf "Grob"	CPSYNCOF2 [FAx] = <Wert>		

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Syntax										
CPSYNFIP2	Zweiter Schwellwert für den Positionssynchronlauf "Fein"	CPSYNFIP2 [FAx]=<Wert>										
CPSYNCOV	Schwellwert für den Geschwindigkeitssynchronlauf "Grob"	CPSYNCOV [FAx]=<Wert>										
CPSYNFIV	Schwellwert für den Geschwindigkeitssynchronlauf "Fein"	CPSYNFIV [FAx]=<Wert>										
CPMBRAKE	Verhalten der Folgeachse bei bestimmten Stoppsignalen und -kommandos	CPMBRAKE [FAx]=<Bitcodierter Wert>										
CPMVDI	Verhalten der Folgeachse auf bestimmte NC/PLC-Nahtstellensignale	CPMVDI [FAx]=<Bitcodierter Wert>										
CPMALARM	Unterdrückung spezieller kopplungsbezogener Alarmausgaben	CPMALARM [FAx]=<Bitcodierter Wert>										
CPSETTYPE	Kopplungstyp	CPSETTYPE [FAx]="<Kopplungstyp>" "<Kopplungstyp>": <table border="1" data-bbox="1007 910 1482 1217"> <tbody> <tr> <td>"CP"</td> <td>Freie Programmierbarkeit</td> </tr> <tr> <td>"TRAIL"</td> <td>Kopplungstyp "Mitschleppen"</td> </tr> <tr> <td>"LEAD"</td> <td>Kopplungstyp "Leitwertkopplung"</td> </tr> <tr> <td>"EG"</td> <td>Kopplungstyp "Elektronisches Getriebe"</td> </tr> <tr> <td>"COUP"</td> <td>Kopplungstyp "Synchronspindel"</td> </tr> </tbody> </table>	"CP"	Freie Programmierbarkeit	"TRAIL"	Kopplungstyp "Mitschleppen"	"LEAD"	Kopplungstyp "Leitwertkopplung"	"EG"	Kopplungstyp "Elektronisches Getriebe"	"COUP"	Kopplungstyp "Synchronspindel"
"CP"	Freie Programmierbarkeit											
"TRAIL"	Kopplungstyp "Mitschleppen"											
"LEAD"	Kopplungstyp "Leitwertkopplung"											
"EG"	Kopplungstyp "Elektronisches Getriebe"											
"COUP"	Kopplungstyp "Synchronspindel"											

FAx: Folgeachse/-spindel

LAx: Leitachse/-spindel

Hinweis

Kopplungseigenschaften, die nicht explizit programmiert werden (im Teileprogramm oder in Synchronaktionen), werden mit ihren Standardeinstellungen wirksam.

Abhängig von der Einstellung des Schlüsselworts CPSETTYPE können statt der Standardeinstellungen (CPSETTYPE="CP") auch voreingestellte Kopplungseigenschaften wirksam werden.

Literatur

Für ausführliche Informationen zur Generischen Kopplung siehe:

- Funktionshandbuch Sonderfunktionen; M3: Achskopplungen, Kapitel: "Generische Kopplung"

12.7 Master/Slave-Kopplung (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Die "Master/Slave-Kopplung" ermöglicht:

- das Einkoppeln der Slave-Achsen auf die Master-Achse im Stillstand der beteiligten Achsen.
- das Koppeln und Trennen von **drehenden**, drehzahlgesteuerten Spindeln.
- die dynamische Projektierung.

Hinweis

Positionierbetrieb

Bei Achsen und Spindeln im Positionierbetrieb wird die Kopplung nur im Stillstand geschlossen und getrennt.

Syntax

```
MASLON(<Slave_1>,<Slave_2>,...)
MASLOF(<Slave_1>,<Slave_2>,...)
MASLOFS(<Slave_1>,<Slave_2>,...)
```

Dynamische Projektierung:

```
MASLDEF(<Slave_1>,<Slave_2>,...,<Master>)
MASLDEL(<Slave_1>,<Slave_2>,...)
```

Bedeutung

MASLON:	Eine temporäre Master/Slave-Kopplung einschalten	
	<Slave_x>, ...:	Slave-Achse 1 ... n
MASLOF:	Eine aktive Master/Slave-Kopplung trennen	
	<Slave_1>, ...:	Slave-Achse 1 ... n
MASLOFS:	Master/Slave-Kopplung trennen und Slave-Spindeln automatisch abbremsten (siehe Hinweis "Koppelverhalten bei Spindeln im Drehzahlsteuerbetrieb!")	
	<Slave_1>, ...:	Slave-Achse 1 ... n
MASLDEF:	Master/Slave-Verband aus dem Teileprogramm heraus anlegen/ändern	
	<Slave_1>, ...:	Slave-Achse 1 ... n
	<Master>:	Master-Achse
MASLDEL:	Master/Slave-Kopplung trennen und Definition des Verbandes löschen	
	<Slave_1>, ...:	Slave-Achse 1 ... n
	Hinweis: Die in den Maschinendaten projektierten Master-/Slave-Definitionen bleiben erhalten.	

Hinweis**Koppelverhalten bei Spindeln im Drehzahlsteuerbetrieb**

Bei Spindeln im Drehzahlsteuerbetrieb wird das Koppelverhalten von MASLON, MASLOF, MASLOFS und MASLDEL explizit über das folgende Maschinendatum festgelegt:

MD37263 \$MA_MS_SPIND_COUPLING_MODE

In der Standardeinstellung mit MD37263 = 0 erfolgt das Einkoppeln und Trennen der Slave-Achsen ausschließlich im Stillstand der beteiligten Achsen. MASLOFS entspricht dem MASLOF.

Bei MD37263 = 1 wird die Koppelanweisung unmittelbar, und damit auch in der Bewegung ausgeführt. Die Kopplung wird bei MASLON sofort geschlossen und bei MASLOFS oder MASLOF sofort getrennt. Die zu diesem Zeitpunkt drehenden Slave-Spindeln behalten bei MASLOF ihre Drehzahlen bis zur erneuten Drehzahlprogrammierung bei. Bei MASLOFS werden sie dagegen automatisch abgebremst.

Hinweis

Bei MASLOF/MASLOFS entfällt der implizite Vorlaufstopp. Bedingt durch den fehlenden Vorlaufstopp liefern die \$P-Systemvariablen für die Slave-Achsen bis zum Zeitpunkt erneuter Programmierung keine aktualisierten Werte.

Hinweis

Für die Slave-Achse kann der Istwert durch PRESETON auf den gleichen Wert der Master-Achse synchronisiert werden. Dazu muss die dauerhafte Master/Slave-Kopplung kurzfristig ausgeschaltet werden, um den Istwert der nicht referierten Slave-Achse mit POWER ON auf den Wert der Master-Achse zu setzen. Danach wird die dauerhafte Kopplung wieder hergestellt.

Die dauerhafte Master/Slave-Kopplung wird mit folgender MD-Einstellung aktiviert:
MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE = 1
Sie hat keine Auswirkung auf die Sprachbefehle der temporären Kopplung.

Beispiele**Beispiel 1: Istwertsetzen bei den Slave-Achse einer Master/Slave-Kopplung**

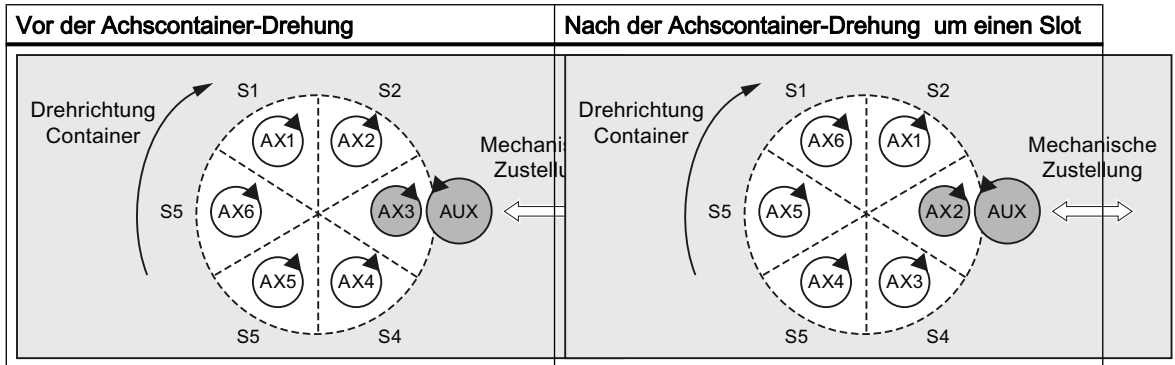
Bei einer permanenten Master/Slave-Kopplung wird mit PRESETON der Istwert der Slave-Achse auf den Wert der Master-Achse gesetzt.

Programmcode	Kommentar
\$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; Permanente Kopplung der Slave-Achse ausschalten
NEWCONF	; Maschinendatenänderung aktivieren
STOPRE	
MASLOF (Y1)	; Temporäre Kopplung ausschalten
PRESETON (AX2, \$VA_IM(M_AX))	; Istwert der Slave-Achse = Istwert der Master-Achse

Programmcode	Kommentar
\$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; Permanente Kopplung der Slave-Achse einschalten
NEWCONF	; Maschinendatenänderung aktivieren

Beispiel 2: Dynamische Projektierung einer Master/Slave-Kopplung

Damit die Kopplung nach der Achscontainer-Drehung mit einer anderen Spindel geschlossen werden kann, muss vorher die alte Kopplung getrennt, die Projektierung gelöscht und die neue Kopplung projiziert werden.



Programmcode	Kommentar
MASLDEF (AUX, S3)	; AUX: Slave, S3: Master = AX3
MASLON (AUX)	; Kopplung ein
M3=3 S3=4000	; Master drehen
MASLDEL (AUX)	; Kopplung trennen und löschen
AXCTSWE (CT1)	; Achscontainer-Drehung freigeben
MASLDEF (AUX, S3)	; AUX: Slave, S3: Master = AX2

Literatur

- Funktionshandbuch Sonderfunktionen, Kapitel "TE3: Drehzahl-/Drehmomentkopplung, Master-Slave"
- Funktionshandbuch Erweiterungsfunktionen, Kapitel "B3: Dezentrale Systeme - nur 840D sl" > "NCU-Link" > "Achscontainer"

Synchronaktionen

13.1 Definition einer Synchronaktion

Eine Synchronaktion wird in einem Satz eines Teileprogramms definiert. Innerhalb dieses Satzes dürfen keine weiteren Befehle programmiert werden, die nicht Bestandteil der Synchronaktion sind.

Eine Synchronaktion besteht aus folgenden Komponenten:

Gültigkeit, Ident-Nr. (optional)	Bedingungsteil (optional)			Aktionsteil		
	Häufigkeit	G-Funktion (optional)	Bedingung	Schlüsselwort	G-Funktion (optional)	Aktionen
--- ¹⁾ ID=<Nr> IDS=<Nr>	--- ¹⁾ WHENEVER FROM WHEN EVERY	G...	Logischer Ausdruck	DO	G...	Aktion 1 ... Aktion n

¹⁾ nicht programmiert

Syntax

```

DO <Aktion 1> ... <Aktion n>
<Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ... <Aktion n>
ID=<Nr> <Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ...
<Aktion n>
IDS=<Nr> <Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ...
<Aktion n>

```

Literatur

Eine ausführliche Beschreibung der Funktionalität von Synchronaktionen findet sich in:
 Funktionshandbuch Synchronaktionen

Pendeln

14.1 Asynchrones Pendeln (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Eine Pendelachse fährt zwischen den zwei Umkehrpunkten 1 und 2 mit gegebenem Vorschub hin und her, bis die Pendelbewegung abgeschaltet wird.

Andere Achsen können während der Pendelbewegung beliebig interpoliert werden. Über eine Bahnbewegung oder mit einer Positionierachse kann eine kontinuierliche Zustellung erreicht werden. Dabei besteht jedoch **kein Zusammenhang** zwischen der Pendel- und der Zustellbewegung.

Eigenschaften des asynchronen Pendelns

- Das asynchrone Pendeln ist achsspezifisch über Satzgrenzen hinweg wirksam.
- Über das Teileprogramm ist ein satzsynchrones Einschalten der Pendelbewegung gewährleistet.
- Eine gemeinsame Interpolation von mehreren Achsen und eine Überlagerung von Pendelstrecken sind nicht möglich.

Programmierung

Über die folgenden Befehle ist ein der Abarbeitung des NC-Programms entsprechendes Einschalten und Beeinflussen des asynchronen Pendelns vom Teileprogramm her möglich.

Die programmierten Werte werden satzsynchron im Hauptlauf in die entsprechenden Settingdaten eingetragen und bleiben bis zur nächsten Änderung wirksam.

Syntax

```
OSP1 [<Achse>]=<Wert>  OSP2 [<Achse>]=<Wert>
OST1 [<Achse>]=<Wert>  OST2 [<Achse>]=<Wert>
FA [<Achse>]=<Wert>
OSCTRL [<Achse>]=(<Setzoption>,<Rücksetzoption>)
OSNSC [<Achse>]=<Wert>
OSE [<Achse>]=<Wert>
OSB [<Achse>]=<Wert>
OS [<Achse>]=1
OS [<Achse>]=0
```

Bedeutung

<Achse>:	Name der Pendelachse	
OS:	Pendeln ein-/ausschalten	
	Wert:	1 Pendeln e inschalten
		0 Pendeln a usschalten
OSP1:	Position von Umkehrpunkt 1 festlegen	

OSP2:	Position von Umkehrpunkt 2 festlegen Hinweis: Falls ein inkrementelles Verfahren aktiv ist, so wird die Position inkrementell zur letzten im NC-Programm programmierten entsprechenden Umkehrposition berechnet.																									
OST1:	Haltezeit im Umkehrpunkt 1 in [s] festlegen																									
OST2:	Haltezeit im Umkehrpunkt 2 in [s] festlegen																									
	<Wert>:	<table border="1"> <tr> <td>-2</td> <td>Interpolation wird ohne Warten auf Genauhalt fortgesetzt</td> </tr> <tr> <td>-1</td> <td>Warten auf Genauhalt grob</td> </tr> <tr> <td>0</td> <td>Warten auf Genauhalt fein</td> </tr> <tr> <td>>0</td> <td>Warten auf Genauhalt fein und anschließend Abwarten der angegebenen Haltezeit Hinweis: Die Einheit für die Haltezeit ist identisch mit der über G4 programmierten Haltezeit.</td> </tr> </table>	-2	Interpolation wird ohne Warten auf Genauhalt fortgesetzt	-1	Warten auf Genauhalt grob	0	Warten auf Genauhalt fein	>0	Warten auf Genauhalt fein und anschließend Abwarten der angegebenen Haltezeit Hinweis: Die Einheit für die Haltezeit ist identisch mit der über G4 programmierten Haltezeit.																
-2	Interpolation wird ohne Warten auf Genauhalt fortgesetzt																									
-1	Warten auf Genauhalt grob																									
0	Warten auf Genauhalt fein																									
>0	Warten auf Genauhalt fein und anschließend Abwarten der angegebenen Haltezeit Hinweis: Die Einheit für die Haltezeit ist identisch mit der über G4 programmierten Haltezeit.																									
FA:	Vorschubgeschwindigkeit festlegen Als Vorschubgeschwindigkeit gilt die definierte Vorschubgeschwindigkeit der Positionierachse. Ist keine Vorschubgeschwindigkeit definiert, gilt der im Maschinendatum hinterlegte Wert.																									
OSCTRL:	Setz- und Rücksetzoptionen angeben Die Optionswerte 0 - 3 verschlüsseln das Verhalten an den Umkehrpunkten beim Ausschalten. Es kann eine der Varianten 0 - 3 ausgewählt werden. Die übrigen Einstellungen sind nach Bedarf kombinierbar mit der gewählten Variante. Mehrere Optionen werden durch Pluszeichen (+) aneinandergefügt.																									
	<Wert>:	<table border="1"> <tr> <td>0</td> <td>Beim Abschalten der Pendelbewegung im nächsten Umkehrpunkt stoppen (Voreinstellung) Hinweis: Nur durch Rücksetzen der Werte 1 und 2 möglich.</td> </tr> <tr> <td>1</td> <td>Beim Abschalten der Pendelbewegung in Umkehrpunkt 1 stoppen</td> </tr> <tr> <td>2</td> <td>Beim Abschalten der Pendelbewegung in Umkehrpunkt 2 stoppen</td> </tr> <tr> <td>3</td> <td>Beim Abschalten der Pendelbewegung keinen Umkehrpunkt anfahren, falls keine Ausfeuerungshübe programmiert sind</td> </tr> <tr> <td>4</td> <td>Nach dem Ausfeuern Endposition anfahren</td> </tr> <tr> <td>8</td> <td>Wird die Pendelbewegung durch Restweglöschen abgebrochen, sollen anschließend Ausfeuerungshübe abgearbeitet und ggf. die Endposition angefahren werden.</td> </tr> <tr> <td>16</td> <td>Wird die Pendelbewegung durch Restweglöschen abgebrochen, soll wie beim Abschalten die entsprechende Umkehrposition angefahren werden.</td> </tr> <tr> <td>32</td> <td>Geänderter Vorschub ist erst ab dem nächsten Umkehrpunkt aktiv</td> </tr> <tr> <td>64</td> <td>FA gleich 0, FA = 0: Wegüberlagerung ist aktiv FA ungleich 0, FA <> 0: Geschwindigkeitsüberlagerung ist aktiv</td> </tr> <tr> <td>128</td> <td>Bei Rundachse DC (kürzester Weg)</td> </tr> <tr> <td>256</td> <td>Ausfeuerhub wird als Doppelhub ausgeführt.(Standard) 1=Ausfeuerhub wird als Einzelhub ausgeführt.</td> </tr> <tr> <td>512</td> <td>Zuerst Startposition anfahren</td> </tr> </table>	0	Beim Abschalten der Pendelbewegung im nächsten Umkehrpunkt stoppen (Voreinstellung) Hinweis: Nur durch Rücksetzen der Werte 1 und 2 möglich.	1	Beim Abschalten der Pendelbewegung in Umkehrpunkt 1 stoppen	2	Beim Abschalten der Pendelbewegung in Umkehrpunkt 2 stoppen	3	Beim Abschalten der Pendelbewegung keinen Umkehrpunkt anfahren, falls keine Ausfeuerungshübe programmiert sind	4	Nach dem Ausfeuern Endposition anfahren	8	Wird die Pendelbewegung durch Restweglöschen abgebrochen, sollen anschließend Ausfeuerungshübe abgearbeitet und ggf. die Endposition angefahren werden.	16	Wird die Pendelbewegung durch Restweglöschen abgebrochen, soll wie beim Abschalten die entsprechende Umkehrposition angefahren werden.	32	Geänderter Vorschub ist erst ab dem nächsten Umkehrpunkt aktiv	64	FA gleich 0, FA = 0: Wegüberlagerung ist aktiv FA ungleich 0, FA <> 0: Geschwindigkeitsüberlagerung ist aktiv	128	Bei Rundachse DC (kürzester Weg)	256	Ausfeuerhub wird als Doppelhub ausgeführt.(Standard) 1=Ausfeuerhub wird als Einzelhub ausgeführt.	512	Zuerst Startposition anfahren
0	Beim Abschalten der Pendelbewegung im nächsten Umkehrpunkt stoppen (Voreinstellung) Hinweis: Nur durch Rücksetzen der Werte 1 und 2 möglich.																									
1	Beim Abschalten der Pendelbewegung in Umkehrpunkt 1 stoppen																									
2	Beim Abschalten der Pendelbewegung in Umkehrpunkt 2 stoppen																									
3	Beim Abschalten der Pendelbewegung keinen Umkehrpunkt anfahren, falls keine Ausfeuerungshübe programmiert sind																									
4	Nach dem Ausfeuern Endposition anfahren																									
8	Wird die Pendelbewegung durch Restweglöschen abgebrochen, sollen anschließend Ausfeuerungshübe abgearbeitet und ggf. die Endposition angefahren werden.																									
16	Wird die Pendelbewegung durch Restweglöschen abgebrochen, soll wie beim Abschalten die entsprechende Umkehrposition angefahren werden.																									
32	Geänderter Vorschub ist erst ab dem nächsten Umkehrpunkt aktiv																									
64	FA gleich 0, FA = 0: Wegüberlagerung ist aktiv FA ungleich 0, FA <> 0: Geschwindigkeitsüberlagerung ist aktiv																									
128	Bei Rundachse DC (kürzester Weg)																									
256	Ausfeuerhub wird als Doppelhub ausgeführt.(Standard) 1=Ausfeuerhub wird als Einzelhub ausgeführt.																									
512	Zuerst Startposition anfahren																									
OSNSC:	Anzahl der Ausfeuerungshübe festlegen																									

14.1 Asynchrones Pendeln (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

OSE:	Endposition (im WKS) festlegen, die nach Ausschalten des Pendelns angefahren werden soll Hinweis: Bei Programmierung von "OSE" wird für "OSCTRL" implizit Option 4 wirksam.
OSB:	Startposition (im WKS) festlegen, die vor Einschalten des Pendelns angefahren werden soll Die Startposition wird vor Umkehrpunkt 1 angefahren. Stimmt die Startposition mit der Umkehrposition 1 überein, so wird als nächstes die Umkehrposition 2 angefahren. Beim Erreichen der Startposition wirkt keine Haltezeit, auch wenn die Startposition mit der Umkehrposition 1 übereinstimmt, stattdessen wird auf Genauhalt fein gewartet. Eine eingestellte Genauhaltbedingung wird eingehalten. Hinweis: Damit die Startposition angefahren wird, muss im Settingdatum SD43770 \$SA_OS-CILL_CTRL_MASK Bit 9 gesetzt sein.

Beispiele

Beispiel 1: Pendelachse soll zwischen zwei Umkehrpunkten pendeln

Die Pendelachse Z soll zwischen Position 10 und 100 pendeln. Umkehrpunkt 1 soll mit Genauhalt fein, Umkehrpunkt 2 mit Genauhalt grob angefahren werden. Der Vorschub für die Pendelachse soll 250 betragen. Am Ende der Bearbeitung sollen 3 Ausfeuerhübe erfolgen und die Pendelachse soll die Endposition 200 ansteuern. Der Vorschub für die Zustellachse soll 1 betragen, das Ende der Zustellung in X-Richtung soll bei Position 15 erreicht sein.

Programmcode	Kommentar
WAITP(X, Y, Z)	; Ausgangsstellung.
G0 X100 Y100 Z100	; Umschalten in Positionierachsbetrieb.
WAITP(X, Z)	
OSP1[Z]=10 OSP2[Z]=100	; Umkehrpunkt 1, Umkehrpunkt 2.
OSE[Z]=200	; Endposition.
OST1[Z]=0 OST2[Z]=-1	; Haltezeit an U1: Genauhalt fein ; Haltezeit an U2: Genauhalt grob
FA[Z]=250 FA[X]=1	; Vorschub Pendelachse, Vorschub Zustellachse.
OSCTRL[Z]=(4, 0)	; Setzoptionen.
OSNSC[Z]=3	; 3 Ausfeuerhübe.
OS[Z]=1	; Pendeln starten.
WHEN \$A_IN[3]==TRUE DO DELDTG(X)	; Restweglöschen.
POS[X]=15	; Ausgangsstellung X-Achse
POS[X]=50	; Endstellung X-Achse.
OS[Z]=0	; Pendeln stoppen.
M30	

Hinweis

Der Befehlsfolge "OSP1[Z]=..." bis "OSNCS[Z]=..." kann auch in einem Satz programmiert werden.

Beispiel 2: Pendeln mit Online-Änderung der Umkehrposition

Die für das asynchrone Pendeln erforderlichen Settingdaten können im Teileprogramm eingestellt werden.

Werden im Teileprogramm die Settingdaten direkt beschrieben, so wird die Änderung schon zum Vorlaufzeitpunkt wirksam. Synchrones Verhalten kann über einen Vorlaufstopp (STOPRE) erreicht werden.

Programmcode	Kommentar
\$SA_OSCILL_REVERSE_POS1[Z]=-10	
\$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER \$AA_IM[Z] < \$\$AA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[X]=0	; Wenn der Istwert der Pendelachse den Umkehrpunkt überschritten hat, wird die Zustellachse angehalten.
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	; Pendeln einschalten.
OS[Z]=0	; Pendeln ausschalten.
M30	

Weitere Informationen**Pendelachse**

Für die Pendelachse gilt:

- Jede Achse kann als Pendelachse benutzt werden.
- Gleichzeitig können mehrere Pendelachsen aktiv sein (maximal: Anzahl der Positionierachsen).
- Für die Pendelachse ist immer - unabhängig vom im Programm aktuell gültigen G-Befehl - Linearinterpolation G1 aktiv.

 14.1 Asynchrones Pendeln (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Die Pendelachse kann:

- Eingangssachse für die dynamische Transformation sein
- Führungssachse bei Gantry- und Mitschleppachsen sein
- verfahren werden:
 - ohne Ruckbegrenzung "BRISK"
oder
 - mit Ruckbegrenzung "SOFT"
oder
 - mit geknickter Beschleunigungskennlinie (wie Positionierachsen)

Pendelumkehrpunkte

Bei der Festlegung der Pendelpositionen sind die aktuellen Verschiebungen zu beachten:

- Absolute Angabe
"OSP1[Z]=<Wert>"
Position Umkehrpunkt = Summe der Verschiebungen + programmierter Wert
- Relative Angabe
"OSP1[Z]=IC(<Wert>)"
Position Umkehrpunkt = Umkehrpunkt 1 + programmierter Wert

Beispiel:

Programmcode
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]=IC(3)

WAITP

Soll mit einer Geometrieachse gependelt werden, so muss diese mit "WAITP" zum Pendeln freigegeben werden.

Nach beendetem Pendeln wird mit "WAITP" die Pendelachse wieder als Positionierachse eingetragen und kann wieder normal verwendet werden.

Pendeln mit Bewegungssynchronaktionen und Haltezeiten

Nach Ablauf der eingestellten Haltezeiten findet beim Pendeln der interne Satzwechsel statt (sichtbar an den neuen Restwegen der Achsen). Beim Satzwechsel wird die Ausschaltfunktion überprüft. Dabei wird nach der eingestellten Steuereinstellung für den Bewegungsablauf (OSCTRL) die Ausschaltfunktion festgelegt. *Dieses Zeitverhalten ist durch den Vorschuboverride beeinflussbar.*

Unter Umständen wird danach noch ein Pendelhub ausgeführt, bevor die Ausfeuerungs hübe gestartet oder die Endposition angefahren wird. *Es entsteht dabei der Eindruck, es verändert sich das Ausschaltverhalten. Dies ist aber nicht der Fall.*

14.2 Über Synchronaktionen gesteuertes Pendeln (OSCILL)

Bei dieser Art des Pendelns ist nur an den Umkehrpunkten bzw. innerhalb definierter Umkehrbereiche eine Zustellbewegung zugelassen.

Je nach Anforderung kann die Pendelbewegung während der Zustellung

- fortgeführt oder
- angehalten werden, bis die Zustellung vollständig ausgeführt ist.

Syntax

1. Parameter für das Pendeln festlegen
2. Bewegungssynchronaktionen definieren
3. Achsen zuordnen, Zustellung festlegen

Bedeutung

OSP1 [<Pendelachse>]=	Position des Umkehrpunkts 1
OSP2 [<Pendelachse>]=	Position des Umkehrpunkts 2
OST1 [<Pendelachse>]=	Haltezeit in Umkehrpunkt 1 in Sekunden
OST2 [<Pendelachse>]=	Haltezeit in Umkehrpunkt 2 in Sekunden
FA [<Pendelachse>]=	Vorschub der Pendelachse
OSCTRL [<Pendelachse>]=	Setz- bzw. Rücksetzoptionen
OSNSC [<Pendelachse>]=	Anzahl der Ausfeuerungshübe
OSE [<Pendelachse>]=	Endposition
WAITP (<Pendelachse>)	Achse für das Pendeln freigeben

Achszuordnung, Zustellung

OSCILL [<Pendelachse>]=(<Zustellachse 1>,<Zustellachse 2>,<Zustellachse 3>)

POSP [<Zustellachse>]=(<Endpos>,<Teillänge>,<Modus>)

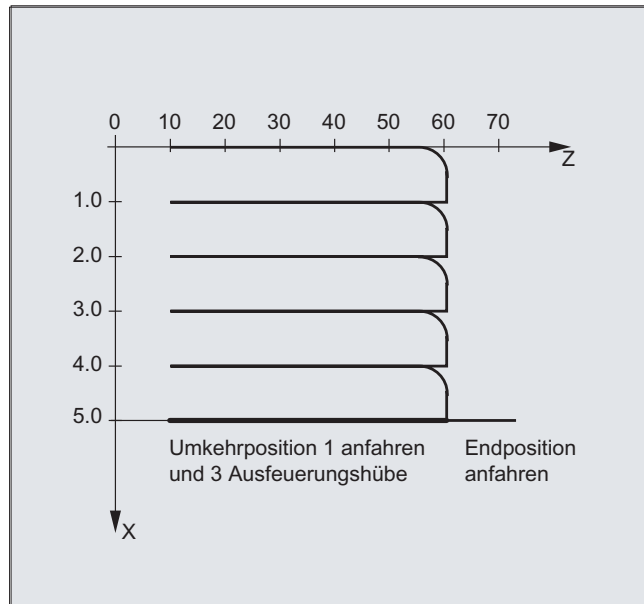
OSCILL:	Zustellachse(n) der Pendelachse zuordnen
POSP:	Gesamt- und Teilzustellungen festlegen (siehe Kapitel Datei- und Programmverwaltung)
Endpos:	Endposition für die Zustellachse, nachdem alle Teilzustellungen abgefahren sind.
Teillänge:	Größe der Teilzustellung am Umkehrpunkt/Umkehrbereich
Modus:	Aufteilung der Gesamtzustellung in Teilzustellungen = zwei gleich große Restschritte (Voreinstellung); = alle Teilzustellungen gleich groß

Bewegungssynchronaktionen

WHEN... .. DO	wenn..., dann...
WHENEVER ... DO	immer wenn..., dann...

Beispiel

Im Umkehrpunkt 1 soll keine Zustellung erfolgen. Beim Umkehrpunkt 2 soll die Zustellung bereits im Abstand $ii2$ vor dem Umkehrpunkt 2 erfolgen und die Pendelachse im Umkehrpunkt nicht auf das Beenden der Teilzustellung warten. Die Achse Z ist Pendelachse und die Achse X Zustellachse.



1. Parameter für das Pendeln

Programmcode	Kommentar
DEF INT ii2	; Variable für Umkehrbereich 2 definieren
OSP1[Z]=10 OSP2[Z]=60	; Umkehrpunkt 1 und 2 definieren
OST1[Z]=0 OST2[Z]=0	; Umkehrpunkt 1: Genauhalt fein Umkehrpunkt 2: Genauhalt fein
FA[Z]=150 FA[X]=0.5	; Vorschub Pendelachse Z, Vorschub Zustellachse X
OSCTRL[Z]=(2+8+16,1)	; Pendelbewegung abschalten im Umkehrpunkt 2; nach RWL Ausfeuern und Endposition anfahren; nach RWL entsprechende Umkehrposition anfahren
OSNC[Z]=3	; Ausfeuerungshübe
OSE[Z]=70	; Endposition = 70
ii2=2	; Umkehrbereich einstellen
WAITP(Z)	; Erlaube Pendeln für Z-Achse

2. Bewegungssynchronaktion

Programmcode	Kommentar
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z] DO -> \$AA_OVR[X]=0 \$AC_MARKER[0]=0	; Immer wenn die aktuelle Position der Pendelachse Z im MKS kleiner als der Beginn des Umkehrbereichs 2 ist, dann setze den axialen Override der Zustellachse X auf 0% und den Merker mit dem Index 0 auf den Wert 0.

Programmcode	Kommentar
WHENEVER \$AA_IM[Z]>=\$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[Z]=0	; Immer wenn die aktuelle Position der Pendelachse Z im MKS größer gleich der Umkehrposition 2 ist, dann setze den axialen Override der Pendelachse Z auf 0%.
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[0]=1	; Immer wenn der Restweg der Teilzustellung gleich 0 ist, dann setze den Merker mit dem Index 0 auf den Wert 1.
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	; Immer wenn der Merker mit dem Index 0 gleich 1 ist, dann setze den axialen Override der Zustellachse X auf 0%. Damit wird eine zu frühe Zustellung verhindert (Pendelachse Z hat den Umkehrbereich 2 noch nicht wieder verlassen, die Zustellachse X ist aber bereit für eine erneute Zustellung). Setze den axialen Override der Pendelachse Z von 0% (Aktion der 2. Synchronaktion) zum Verfahren wieder auf 100%.

-> muss in einem Satz programmiert werden

3. Pendeln starten

Programmcode	Kommentar
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; Starten der Achsen Der Pendelachse Z wird die Achse X als Zustellachse ; zugewiesen. Die Achse X soll bis Endposition 5 in Schritten von 1 fahren.
M30	; Programmende

Weitere Informationen

- Pendelparameter festlegen**
Vor dem Bewegungssatz, der die Zuordnung von Zustell- und Pendelachse sowie die Festlegung der Zustellung enthält, sind die Parameter für das Pendeln festzulegen (siehe "Asynchrones Pendeln").
- Bewegungssynchronaktionen festlegen**
Über Synchronbedingungen erfolgt:
Zustellung unterdrücken, bis sich die Pendelachse innerhalb eines Umkehrbereichs (ii1, ii2) oder an einem Umkehrpunkt (U1, U2) befindet.
Pendelbewegung während der Zustellung im Umkehrpunkt **anhalten**.
Pendelbewegung nach beendeter Teilzustellung wieder **starten**.
Start der nächsten Teilzustellung festlegen.
- Pendeln- und Zustellachse zuordnen** sowie **Gesamt- und Teilzustellung** festlegen.

Pendelparameter festlegen

Zuordnung von Pendel- und Zustellachse: OSCILL

OSCILL[<Pendelachse>]=(<Zustellachse1>,<Zustellachse2>,<Zustellachse3>)

Mit dem Befehl "OSCILL" erfolgen die Achszuordnungen und der Start der Pendelbewegung. Maximal können einer Pendelachse 3 Zustellachsen zugewiesen werden.

Hinweis

Vor dem Start des Pendelns müssen die Synchronbedingungen für das Verhalten der Achsen festgelegt sein.

Zustellungen festlegen: POSP

POSP[<Zustellachse>]=(<Endpos>,<Teillänge>,<Modus>)

Mit dem Befehl "POSP" werden der Steuerung mitgeteilt:

- Gesamtzustellung (über die Endposition)
- Die Größe der jeweiligen Teilzustellung am Umkehrpunkt bzw. im Umkehrbereich
- Das Teilzustellverhalten bei Erreichen der Endposition (über den Modus)

Modus = 0	Für die beiden letzten Teilzustellungen erfolgt eine Aufteilung des verbleibenden Weges bis zum Zielpunkt auf 2 gleich große Restschritte (Voreinstellung).
Modus = 1	Alle Teilzustellungen sind gleich groß. Sie werden aus der Gesamtzustellung berechnet.

Bewegungssynchronaktionen festlegen

Die im folgenden ausgeführten Bewegungssynchronaktionen werden ganz allgemein zum Pendeln verwendet.

Sie finden Beispiellösungen für die Lösung von einzelnen Anforderungen, die Ihnen als Bausteine für die Erstellung von anwenderspezifischen Pendelbewegungen dienen.

Hinweis

Im Einzelfall können die Synchronbedingungen auch anders programmiert werden.

Schlüsselwörter

WHEN ... DO ...	wenn..., dann...
WHENEVER ... DO	immer wenn..., dann...

Funktionen

Mit den im folgenden detailliert beschriebenen Sprachmitteln können Sie folgende Funktionen realisieren:

1. Zustellung im Umkehrpunkt.
2. Zustellung im Umkehrbereich.
3. Zustellung in beiden Umkehrpunkten.
4. Anhalten der Pendelbewegung im Umkehrpunkt.
5. Pendelbewegung wieder starten.
6. Teilzustellung nicht zu früh starten.

Für alle hier beispielhaft dargestellten Synchronaktionen gelten die Annahmen:

- Umkehrpunkt 1 < Umkehrpunkt 2
- Z = Pendelachse
- X = Zustellachse

Hinweis

Für nähere Erläuterungen, siehe Kapitel Bewegungssynchronaktionen.

Pendeln- und Zustellachse zuordnen sowie Gesamt- und Teilzustellung festlegen**Zustellung im Umkehrbereich**

Die Zustellbewegung soll innerhalb eines Umkehrbereichs beginnen, bevor der Umkehrpunkt erreicht ist.

Diese Synchronaktionen verhindern die Zustellbewegung, bis sich die Pendelachse in einem Umkehrbereich befindet.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

Umkehrbereich 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X]=0
```

Immer wenn die aktuelle Position der Pendelachse im MKS größer als der Beginn des Umkehrbereichs 1 ist, dann setze den axialen Override der Zustellachse auf 0%.

Umkehrbereich 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X]=0
```

Immer wenn die aktuelle Position der Pendelachse im KS kleiner als der Beginn des Umkehrbereichs 2 ist, dann setze den axialen Override der Zustellachse auf 0%.

Zustellung im Umkehrpunkt

Solange die Pendelachse den Umkehrpunkt nicht erreicht hat, findet keine Bewegung der Zustellachse statt.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

Umkehrbereich 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse Z im MKS größer oder kleiner als die Position des Umkehrpunkts 1 ist, dann setze den axialen Override der Zustellachse X auf 0% und den axialen Override der Pendelachse Z auf 100%.

Umkehrbereich 2:

Für Umkehrpunkt 2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse Zu im MKS größer oder kleiner als die Position des Umkehrpunkts 2 ist, dann setze den axialen Override der Zustellachse X auf 0% und den axialen Override der Pendelachse Z auf 100%.

Anhalten der Pendelbewegung im Umkehrpunkt

Die Pendelachse wird am Umkehrpunkt angehalten, gleichzeitig beginnt die Zustellbewegung. Die Pendelbewegung wird fortgesetzt, wenn die Zustellbewegung vollständig ausgeführt ist.

Gleichzeitig kann diese Synchronaktion dazu benutzt werden, die Zustellbewegung zu starten, falls diese durch eine vorhergehende Synchronaktion, die noch wirksam ist, gestoppt wurde.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

Umkehrbereich 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse im MKS gleich der Umkehrposition 1 ist, dann setze den axialen Ovberrie der Pendelachse auf 0% und den axialen Override der Zustellachse auf 100%.

Umkehrbereich 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse Zu im MKS gleich der Umkehrposition 2 ist, dann setze den axialen Override der PendelachseX auf 0% und den axialen Override der Zustellachse auf 100%.

Online-Auswertung des Umkehrpunkts

Steht auf der rechten Seite des Vergleichs eine mit \$\$ gekennzeichnete Hauptlaufvariable, so werden die beiden Variablen im IPO-Takt laufend ausgewertet und miteinander verglichen.

Hinweis

Mehr Informationen hierzu siehe Kapitel "Bewegungssynchronaktionen".

Pendelbewegung wieder starten

Diese Synchronaktion wird dazu benutzt, die Bewegung der Pendelachse fortzusetzen, wenn die Teilzustellbewegung abgeschlossen ist.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]=100
```

Immer wenn der Restweg für die Teilzustellung der Zustellachse X im WKS gleich Null ist, dann setze den axialen Override der Pendelachse auf 100%.

Nächste Teilzustellung

Nach erfolgter Zustellung muss ein zu frühes Starten der nächsten Teilzustellung verhindert werden.

Dazu wird ein kanalspezifischer Merker (`$AC_MARKER[Index]`) verwendet, der am Ende der Teilzustellung (Teilrestweg $\equiv 0$) gesetzt wird und beim Verlassen des Umkehrbereichs gelöscht wird. Dann wird mit einer Synchronaktion die nächste Zustellbewegung verhindert.

Unter den gegebenen Annahmen (siehe oben) ergeben sich z. B. für Umkehrpunkt 1 folgende Anweisungen:

1. Marker setzen:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1]=1
```

Immer wenn der Restweg für die Teilzustellung der Zustellachse X im WKS gleich Null ist, dann setze den Merker mit Index 1 auf 1.

2. Marker löschen

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1]=0
```

Immer wenn die aktuelle Position der Pendelachse Z im MKS größer oder kleiner als die Position des Umkehrpunkts 1 ist, dann setze den Merker 1 auf 0.

3. Zustellung verhindern

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

Immer wenn der Merker 1 gleich ist, dann setze den axialen Override der Zustellachse X auf 0%.

Stanzen und Nibbeln

15.1 Aktivierung, Deaktivierung

15.1.1 Stanzen und Nibbeln ein oder aus (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Stanzen bzw. Nibbeln aktivieren/deaktivieren

Mit `PON` und `SON` wird die Stanz- bzw. Nibbelfunktion aktiviert. `SPOF` beendet alle stanz- und nibbelspezifischen Funktionen. Die modal wirksamen Befehle `PON` und `SON` schließen sich gegenseitig aus, d. h. `PON` deaktiviert `SON` und umgekehrt.

Stanzen/Nibbeln mit Vorspann

Die Funktionen `SONS` und `PONS` schalten ebenfalls die Stanz- bzw. Nibbelfunktionen ein.

Im Gegensatz zu der bei `SON/PON` wirksamen Hubsteuerung auf Interpolationsebene erfolgt bei diesen Funktionen die signaltechnische Steuerung der Hubauslösung auf Servoebene. Hierdurch kann mit höheren Hubfrequenzen und damit höherer Stanzleistung gearbeitet werden.

Während der Signalauswertung im Vorspann sind alle Funktionen verriegelt, die zur Positionsänderung der Nibbel- oder Stanzachsen führen (z. B. Handradfahren, Änderungen von Frames über PLC, Messfunktionen).

Stanzen mit Verzögerung

`PDELAYON` bewirkt eine verzögerte Ausgabe des Stanzhubs. Der modal wirksame Befehl hat vorbereitende Funktion und steht damit in der Regel vor `PON`. Nach `PDELAYOF` wird normal weitergestanz.

Hinweis

Die Verzögerungszeit wird eingestellt im Settingdatum `SD42400 $SC_PUNCH_DWELLTIME`.

Wegabhängige Beschleunigung

Mit `PUNCHACC` kann eine Beschleunigungskennlinie festgelegt werden, die je nach Lochabstand unterschiedliche Beschleunigungen definiert.

Zweites Stanz-Interface

Maschinen, die abwechselnd ein zweites Stand-Interface (zweite Stanzeinheit oder ein vergleichbares Medium) nutzen sollen, können auf ein zweites Paar der schnellen digitalen Ein- und Ausgänge der Steuerung (I/O-Paar) umgeschaltet werden. Für beide Stand-

Interfaces ist die volle Stanz-/Nibbel-Funktionalität nutzbar. Die Umschaltung zwischen erstem und zweitem Stanz-Interface erfolgt über die Befehle SPIF1 und SPIF2.

Hinweis

Voraussetzung: Über Maschinendaten muss ein zweites I/O-Paar für die Stanzfunktionalität definiert sein (→ siehe Angaben des Maschinenherstellers!).

Syntax

```
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
```

Bedeutung

PON:	Stanzen aktivieren
SON:	Nibbeln aktivieren
PONS:	Stanzen mit Vorspann aktivieren
SONS:	Nibbeln mit Vorspann aktivieren
SPOF:	Stanzen/Nibbeln deaktivieren
PDELAYON:	Stanzen mit Verzögerung aktivieren
PDELAYOF:	Stanzen mit Verzögerung deaktivieren
PUNCHACC:	Wegabhängige Beschleunigung aktivieren
	Parameter:
	<Smin> Kleinster Lochabstand
	<Amin> Anfangsbeschleunigung <Amin> kann größer als <Amax> sein.
	<Smax> Größter Lochabstand
	<Amax> Endbeschleunigung <Amax> kann größer als <Amin> sein.
SPIF1:	Erstes Stanz-Interface aktivieren Die Hubsteuerung erfolgt über das erste Paar der schnellen I/O.
SPIF2:	Zweites Stanz-Interface aktivieren Die Hubsteuerung erfolgt über das zweite Paar der schnellen I/O.
	Hinweis: Nach RESET oder Steuerungshochlauf ist immer das erste Stanz-Interface aktiv. Wird nur ein Stanz-Interface benutzt, so muss dieses nicht programmiert werden.

Beispiele

Beispiel 1: Nibbeln aktivieren

Programmcode	Kommentar
...	
N70 X50 SPOF	; Positionieren ohne Stanzauslösung.
N80 X100 SON	; Nibbeln aktivieren, Auslösung eines Hubs vor der Bewegung (X=50) und am Ende der programmierten Bewegung (X=100).
...	

Beispiel 2: Stanzen mit Verzögerung

Programmcode	Kommentar
...	
N170 PDELAYON X100 SPOF	; Positionieren ohne Stanzauslösung, Aktivierung der verzögerten Stanzauslösung.
N180 X800 PON	; Stanzen aktivieren. Nach Erreichen der Endposition wird Stanzhub verzögert ausgegeben.
N190 PDELAYOF X700	; Stanzen mit Verzögerung deaktivieren, normale Stanzauslösung am Ende der programmierten Bewegung.
...	

Beispiel 3: Stanzen mit zwei Stand-Interfaces

Programmcode	Kommentar
...	
N170 SPIF1 X100 PON	; Am Ende des Satzes erfolgt eine Hubauslösung auf dem ersten schnellen Ausgang. Das Signal "Hub aktiv" wird auf dem ersten Eingang überwacht.
N180 X800 SPIF2	; Die zweite Hubauslösung erfolgt auf dem zweiten schnellen Ausgang. Das Signal "Hub aktiv" wird auf dem zweiten Eingang überwacht.
N190 SPIF1 X700	; Die Hubsteuerung für alle weiteren Hübe erfolgt mit dem ersten Interface.
...	

Weitere Informationen

Stanzen und Nibbeln mit Vorspann (PONS/SONS)

Stanzen und Nibbeln mit Vorspann ist nicht gleichzeitig in mehreren Kanälen möglich. PONS bzw. SONS kann nur jeweils in einem Kanal aktiviert werden.

Wegabhängige Beschleunigung (PUNCHACC)

Beispiel:

PUNCHACC (2, 50, 10, 100)

Lochabstände unter 2mm:

Es wird mit einer Beschleunigung von 50% der Maximalbeschleunigung verfahren.

Lochabstände von 2mm bis 10mm:

Die Beschleunigung wird proportional zum Abstand auf 100% gesteigert.

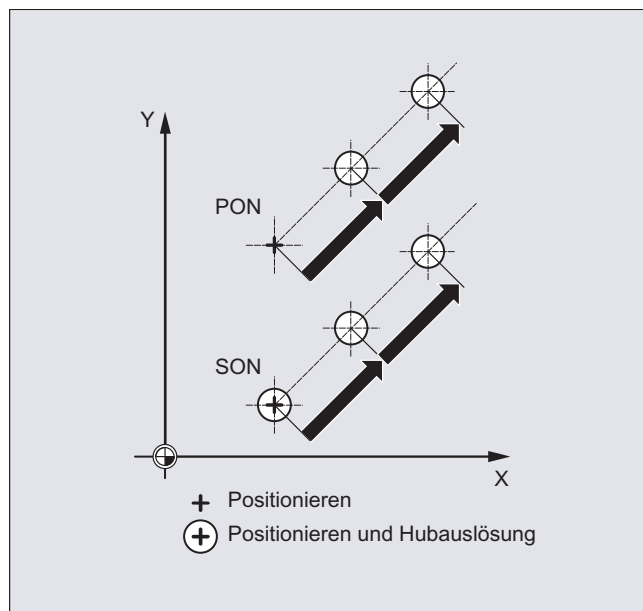
Lochabstände größer als 10mm:

Verfahren mit einer Beschleunigung von 100%.

Auslösung des ersten Hubs

Die Auslösung des ersten Hubs nach Aktivierung der Funktion erfolgt beim Nibbeln und Stanzen zeitlich unterschiedlich:

- PON/ PONS:
 - Alle Hübe - auch der des ersten Satzes nach Aktivierung - erfolgen im Satzende.
- SON/ SONS:
 - Der erste Hub nach Aktivierung des Nibbelns erfolgt bereits im Satzanfang.
 - Alle weiteren Hübe werden jeweils im Satzende ausgelöst.



Stanzen und Nibbeln auf der Stelle

Eine Hubauslösung erfolgt nur dann, wenn der Satz eine Verfahrinformation für die Stanz- oder Nibbelachsen (Achsen der aktiven Ebene) enthält.

Um dennoch einen Hub an gleicher Stelle auszulösen, wird eine der Stanz-/Nibbelachsen mit Verfahrweg 0 programmiert.

Arbeiten mit drehbaren Werkzeugen

Hinweis

Um drehbare Werkzeuge tangential an die programmierte Bahn anzustellen, verwenden Sie die Tangentialsteuerung.

Verwendung von M-Befehlen

Mit Hilfe der Makrotechnik ist es nach wie vor möglich, spezielle M-Funktionen statt der Sprachbefehle zu benutzen (Kompatibilität). Dabei gelten die folgenden Entsprechungen zu älteren Systemen:

M20, M23	△	SPOF
M22	△	SON
M25	△	PON
M26	△	PDELAYON

Beispiel für Makrodatei:

Programmcode	Kommentar
DEFINE M25 AS PON	; Stanzen ein
DEFINE M125 AS PONS	; Stanzen mit Vorspann ein
DEFINE M22 AS SON	; Nibbeln ein
DEFINE M122 AS SONS	; Nibbeln mit Vorspann ein
DEFINE M26 AS PDELAYON	; Stanzen mit Verzögerung ein
DEFINE M20 AS SPOF	; Stanzen, Nibbeln aus
DEFINE M23 AS SPOF	; Stanzen, Nibbeln aus

Programmierbeispiel:

Programmcode	Kommentar
...	
N100 X100 M20	; Positionieren ohne Stanzauslösung.
N110 X120 M22	; Nibbeln aktivieren, vor und nach Bewegung Hubauslösung.
N120 X150 Y150 M25	; Stanzen aktivieren, Hubauslösung am Ende der Bewegung.
...	

15.2 Automatische Wegaufteilung

Unterteilung in Teilstrecken

Bei aktiviertem Stanzen bzw. Nibbeln bewirken sowohl SPP als auch SPN eine Aufteilung der für die Bahnachsen programmierten Gesamtverfahrstrecke in eine Anzahl von gleichlangen Teilstrecken (äquidistante Wegaufteilung). Intern entspricht jede Teilstrecke einem Satz.

Anzahl der Hübe

Beim Stanzen erfolgt der erste Hub am Endpunkt der ersten Teilstrecke, beim Nibbeln dagegen am Startpunkt der ersten Teilstrecke. Über die Gesamtfahrstrecke ergeben sich damit folgende Zahlen:

Stanzen: Anzahl der Hübe = Anzahl der Teilstrecken

Nibbeln: Anzahl der Hübe = Anzahl der Teilstrecken + 1

Hilfsfunktionen

Hilfsfunktionen werden im ersten der erzeugten Sätze ausgeführt.

Syntax

SPP=

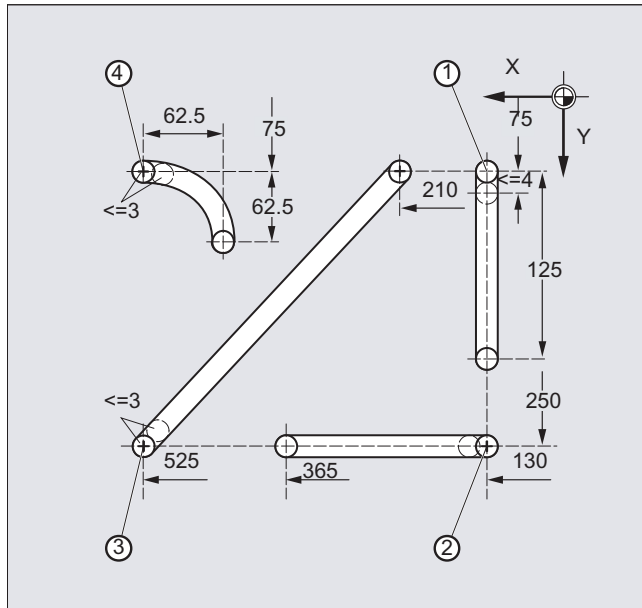
SPN=

Bedeutung

SPP:	Größe der Teilstrecke (maximaler Hubabstand); modal wirksam
SPN:	Anzahl der Teilstrecken pro Satz; satzweise wirksam

Beispiel 1

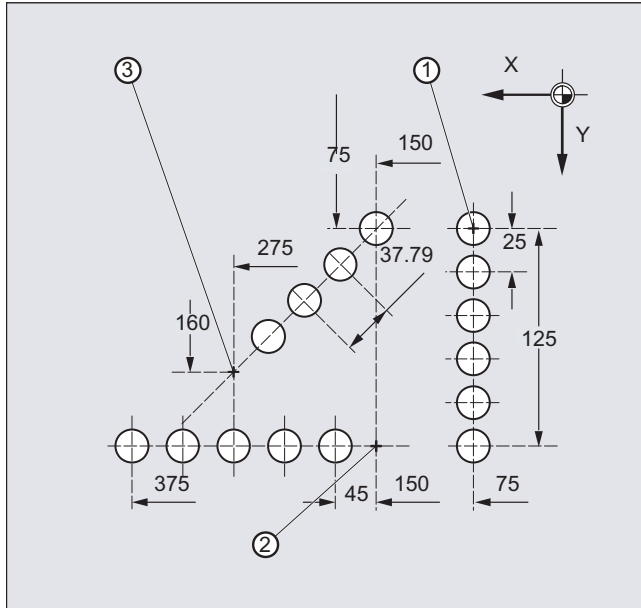
Die programmierten Nibbelstrecken sollen automatisch in gleichgroße Teilstrecken aufgeteilt werden.



Programmcode	Kommentar
N100 G90 X130 Y75 F60 SPOF	; Positionieren auf Startpunkt 1
N110 G91 Y125 SPP=4 SON	; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 4 mm
N120 G90 Y250 SPOF	; Nibbeln aus; Positionieren auf Startpunkt 2
N130 X365 SON	; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 4 mm
N140 X525 SPOF	; Nibbeln aus; Positionieren auf Startpunkt 3
N150 X210 Y75 SPP=3 SON	; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 3 mm
N160 X525 SPOF	; Nibbeln aus; Positionieren auf Startpunkt 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 3 mm
N180 G00 G90 Y300 SPOF	; Nibbeln aus

Beispiel 2

Für die einzelnen Lochreihen soll eine automatische Wegaufteilung erfolgen. Für die Aufteilung wird jeweils die maximale Teilstreckenlänge (SPP-Wert) angegeben.



Programmcode	Kommentar
N100 G90 X75 Y75 F60 PON	; Positionieren auf Startpunkt 1; Stanzen ein Einzelloch stanzen
N110 G91 Y125 SPP=25	; Maximale Teilstreckenlänge für automatische Wegaufteilung: 25 mm
N120 G90 X150 SPOF	; Stanzen aus; Positionieren auf Startpunkt 2
N130 X375 SPP=45 PON	; Stanzen ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 45 mm
N140 X275 Y160 SPOF	; Stanzen aus; Positionieren auf Startpunkt 3
N150 X150 Y75 SPP=40 PON	; Stanzen ein; anstelle der programmierten Teilstre- ckenlänge von 40 mm wird die ;berechnete Teilstrecken- länge von 37,79 mm verwendet.
N160 G00 Y300 SPOF	; Stanzen aus; Positionieren

15.2.1 Wegaufteilung bei Bahnachsen

Länge der Teilstrecke SPP

Mit *SPP* geben Sie den maximalen Hubabstand und damit die maximale Länge der Teilstrecken an, in die die Gesamtverfahrstrecke aufgeteilt werden soll. Das Ausschalten des Befehls erfolgt mit *SPOF* oder *SPP=0*.

Beispiel:

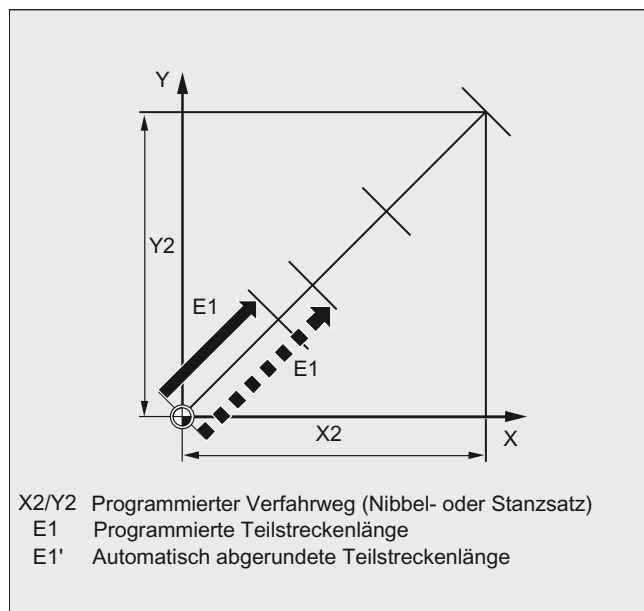
```
N10 SON X0 Y0
```

```
N20 SPP=2 X10
```

Die Gesamtverfahrstrecke von 10 mm wird in 5 Teilstrecken von je 2 mm ($SPP=2$) aufgeteilt.

Hinweis

Die Wegaufteilung mit SPP erfolgt immer äquidistant: alle Teilstrecken sind gleich lang. Das heißt, die programmierte Teilstreckengröße (Wert von SPP) ist nur dann gültig, wenn der Quotient aus Gesamtverfahrstrecke und SPP -Wert ganzzahlig ist. Ist das nicht der Fall, so wird die Größe der Teilstrecke intern so reduziert, dass sich ein ganzzahliger Quotient ergibt.



Beispiel:

```
N10 G1 G91 SON X10 Y10
```

```
N20 SPP=3.5 X15 Y15
```

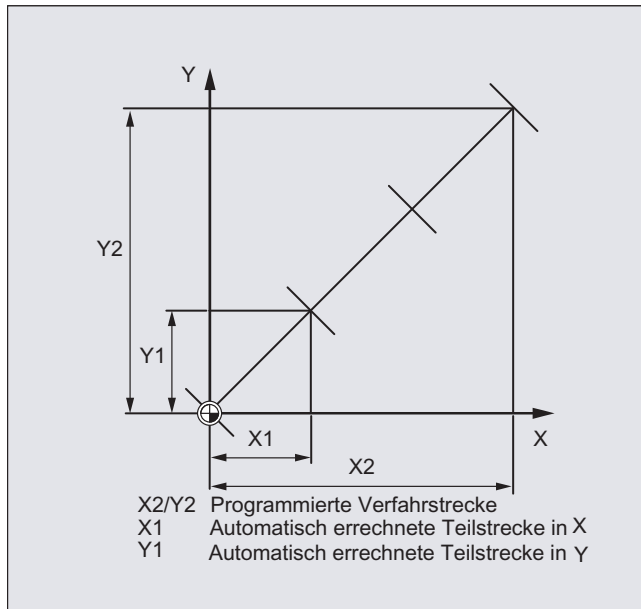
Bei der Gesamtverfahrstrecke von 15 mm und einer Teilstreckenlänge von 3,5 mm ergibt sich ein nicht ganzzahliger Quotient (4,28). Somit erfolgt eine Reduktion des SPP -Werts bis zum nächstmöglichen ganzzahligen Quotienten. In diesem Fall ergibt sich eine Teilstreckenlänge von 3 mm.

Anzahl der Teilstrecken SPN

Mit SPN definieren Sie die Anzahl der Teilstrecken, die aus der Gesamtverfahrstrecke erzeugt werden soll. Die Länge der Teilstrecken wird automatisch berechnet. Da SPN satzweise wirksam ist, muss vorher Stanzen oder Nibbeln mit PON oder SON aktiviert werden.

SPP und SPN im gleichen Satz

Programmieren Sie in einem Satz sowohl die Teilstreckenlänge (*SPP*) als auch Anzahl der Teilstrecken (*SPN*), so gilt für diesen Satz *SPN*, für alle weiteren *SPP*. Wurde *SPP* schon vor *SPN* aktiviert, so wird es nach dem Satz mit *SPN* wieder wirksam.



Hinweis

Sofern Stanzen/Nibbeln grundsätzlich in der Steuerung verfügbar ist, ist die Programmierung der automatischen Wegaufteilung mit *SPN* bzw. *SPP* auch unabhängig von dieser Technologie aktivierbar.

15.2.2 Wegaufteilung bei Einzelachsen

Sind neben den Bahnachsen auch Einzelachsen als Stanz-Nibbel-Achse definiert, so können auch sie der automatischen Wegaufteilung unterliegen.

Verhalten der Einzelachse bei SPP

Die programmierte Länge der Teilstrecke (*SPP*) bezieht sich grundsätzlich auf die Bahnachsen. Daher wird in einem Satz, in dem neben der Einzelachsbewegung und dem *SPP*-Wert keine Bahnachse programmiert ist, der *SPP*-Wert ignoriert.

Sind sowohl Einzel- als auch Bahnachse im Satz programmiert, so richtet sich das Verhalten der Einzelachse nach der Einstellung des entsprechenden Maschinendatums.

1. Standardeinstellung

Der Weg der Einzelachse wird gleichmäßig auf die durch *SPP* erzeugten Zwischensätze verteilt.

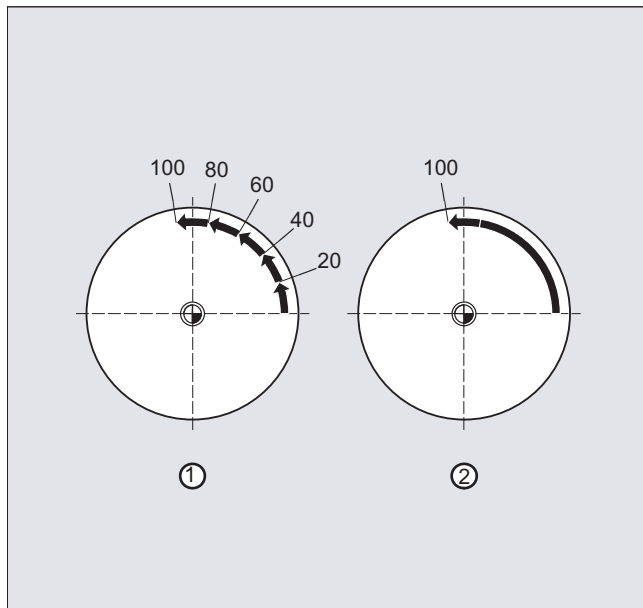
Beispiel:

N10 G1 SON X10 A0

N20 SPP=3 X25 A100

Durch die Hubstrecke von 3 mm werden bei der Gesamtverfahrstrecke der X-Achse (Bahnachse) von 15 mm 5 Sätze erzeugt.

Die A-Achse dreht sich damit in jedem Satz um 20°.



1. Einzelachse ohne Wegaufteilung
Die Einzelachse verfährt ihren Gesamtweg im ersten der erzeugten Sätze.
2. Unterschiedliche Wegaufteilung
Das Verhalten der Einzelachse ist abhängig von der Interpolation der Bahnachsen:
 - Kreisinterpolation: Wegaufteilung
 - Linearinterpolation: keine Wegaufteilung

Verhalten bei SPN

Die programmierte Anzahl von Teilstrecken gilt auch, wenn nicht gleichzeitig eine Bahnachse programmiert ist.

Voraussetzung: Einzelachse ist als Stanz-Nibbel-Achse definiert.

Schleifen

16.1 Schleifenspezifische Werkzeugüberwachung ein-/ausschalten (TMON, TMOF)

Mit den vordefinierten Prozeduren TMON(...) und TMOF(...) wird die schleifenspezifische Werkzeugüberwachung (Geometrie- und Drehzahlüberwachung) ein- bzw. ausgeschaltet.

Voraussetzung

Die werkzeugspezifischen Parameter \$TC_TPG1 bis \$TC_TPG9 müssen gesetzt sein.

Syntax

```
TMON (<Tnr>
...
TMOF (<Tnr>)
```

Bedeutung

TMON (. . .) :	Schleifenspezifische Werkzeugüberwachung einschalten Der Befehl muss in dem Kanal programmiert werden, in dem die schleifenspezifische Werkzeugüberwachung eingeschaltet werden soll.
TMOF (. . .) :	Schleifenspezifische Werkzeugüberwachung ausschalten Der Befehl muss in dem Kanal programmiert werden, in dem die schleifenspezifische Werkzeugüberwachung ausgeschaltet werden soll.
<Tnr>:	T-Nummer Hinweis: Nur erforderlich, wenn die Überwachung statt für das aktive, im Einsatz befindliche Werkzeug für eine nicht aktive Schleifscheibe ein- bzw. ausgeschaltet werden soll.
TMOF (0) :	Überwachung für alle Werkzeuge ausschalten

Weitere Funktionen

17.1 Achsfunktionen (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

"AXNAME" wird z. B. bei der Erstellung allgemeingültiger Zyklen verwendet, wenn die Namen der Achsen nicht bekannt sind.

"AX" wird für die indirekte Programmierung von Geometrie- und Synchronachsen verwendet. Der Achsbezeichner wird dabei in einer Variablen vom Typ AXIS hinterlegt oder von einem Befehl wie "AXNAME" oder "SPI" geliefert.

"SPI" wird verwendet, wenn Achsfunktionen für eine Spindel, z. B. Synchronspindel, programmiert werden.

"AXTOSPI" wird verwendet, um einen Achsbezeichner in einen Spindelindex zu wandeln (Umkehrfunktion zu "SPI").

"AXSTRING" wird verwendet, um einen Achsbezeichner (Datentyp AXIS) in einen String zu wandeln (Umkehrfunktion zu "AXNAME").

"ISAXIS" wird in allgemeingültigen Zyklen verwendet, um sicherzustellen, dass eine bestimmte Geometrieachse vorhanden ist und damit ein nachfolgender Aufruf von \$P_AXNX nicht mit Fehler abgebrochen wird.

"MODAXVAL" wird verwendet, um bei Modulo-Rundachsen die Modulo-Position zu ermitteln.

Syntax

```
AXNAME ("String")
AX[AXNAME ("String")]
SPI (n)

AXTOSPI (A) oder AXTOSPI (B) oder AXTOSPI (C)
AXSTRING (SPI (n))
ISAXIS (<Geometrieachsnummer>)
<Modulo-Position>=MODAXVAL (<Achse>, <Achsposition>)
```

Bedeutung

AXNAME:	Konvertiert einen Eingangsstring in Achsbezeichner; der Eingangsstring muss gültigen Achsnamen enthalten.
AX:	Variabler Achsbezeichner
SPI:	Konvertiert Spindelnummer in Achsbezeichner; der Übergabeparameter muss eine gültige Spindelnummer enthalten.
n:	Spindelnummer
AXTOSPI:	Wandelt einen Achsbezeichner in einen Spindelindex vom Typ Integer um. "AXTOSPI" entspricht der Umkehrfunktion zu "SPI".
X, Y, Z:	Achsbezeichner vom Typ AXIS als Variable oder Konstante
AXSTRING:	Es wird der String mit zugeordneter Spindelnummer ausgegeben.

ISAXIS:	Prüft, ob die angegebene Geometrieachse vorhanden ist.
MODAXVAL:	Ermittelt bei Modulo-Rundachsen die Modulo-Position; diese entspricht dem Modulo-Rest bezogen auf den parametrisierten Modulo-Bereich (beträgt in der Standardeinstellung 0 bis 360 Grad; über MD30340 MODULO_RANGE_START und MD30330 \$MA_MODULO_RANGE können Beginn und Größe des Modulo-Bereichs verändert werden).

Hinweis

SPI-Erweiterungen

Die Achsfunktion SPI(n) ist auch für das Lesen und Schreiben von Framekomponenten einsetzbar. Damit können Frames z. B. mit der Syntax \$P_PFRAME[SPI(1),TR]=2.22 geschrieben werden.

Durch die zusätzliche Programmierung von Achspositionen über die Adresse AX[SPI(1)]=<Achsposition> kann eine Achse verfahren werden. Voraussetzung dafür ist, dass sich die Spindel im Positionier- oder Achsbetrieb befindet.

Beispiele

Beispiel 1: AXNAME, AX, ISAXIS

Programmcode	Kommentar
OVRA[AXNAME("Planachse")]=10	; Override für Planachse
AX[AXNAME("Planachse")]=50.2	; Endposition für Planachse
OVRA[SPI(1)]=70	; Override für Spindel 1
AX[SPI(1)]=180	; Endposition für Spindel 1
IF ISAXIS(1)==FALSE GOTOF WEITER	; Abszisse vorhanden?
AX[\$P_AXN1]=100	; Abszisse verfahren
WEITER:	

Beispiel 2: AXSTRING

Bei der Programmierung mit AXSTRING[SPI(n)] wird nicht mehr der Achsindex der Achse, der die Spindel zugeordnet ist, als Spindelnummer ausgegeben, sondern es wird der String "Sn" ausgegeben.

Programmcode	Kommentar
AXSTRING[SPI(2)]	; Es wird der String "S2" ausgegeben.

Beispiel 3: MODAXVAL

Die Modulo-Position der Modulo-Rundachse A soll ermittelt werden.

Ausgangswert für die Berechnung ist die Achsposition 372.55.

Der parametrisierte Modulo-Bereich beträgt 0 bis 360 Grad:

MD30340 MODULO_RANGE_START = 0

MD30330 \$MA_MODULO_RANGE = 360

Programmcode	Kommentar
R10=MODAXVAL(A,372.55)	; Berechnete Modulo-Position R10 = 12.55.

Beispiel 4: MODAXVAL

Wenn sich der programmierte Achsbezeichner nicht auf eine Modulo-Rundachse bezieht, dann wird der zu wandelnde Wert (<Achsposition>) unverändert zurückgegeben.

Programmcode	Kommentar
R11=MODAXVAL(X,372.55)	; X ist Linearachse; R11 = 372.55.

17.2 Umschaltbare Geometrieachsen (GEOAX)

Mit der Funktion "Umschaltbare Geometrieachsen" können die über Maschinendaten parametrisierten Geometrieachsen durch andere Kanalachsen ersetzt werden.

Syntax

GEOAX (<n>, <Kanalachse>, <n>, <Kanalachse>, <n>, <Kanalachse>)
 GEOAX ()

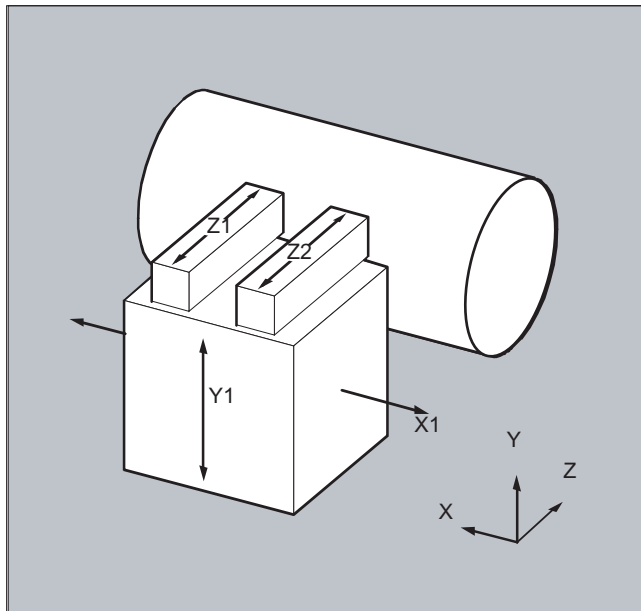
Bedeutung

GEOAX (...)	Funktion zum Umschalten von Geometrieachsen. Hinweis: GEOAX () ohne Parameterangabe aktiviert wieder die in den Maschinendaten parametrisierte Grundkonfiguration der Geometrieachsen.
<n>	Nummer der Geometrieachse, die durch die angegebene Kanalachse ersetzt wird. Wertebereich: 0, 1, 2, 3 Hinweis: 0: Die angegebene Kanalachse wird ersatzlos aus dem Geometrieachsverbund entfernt 1: 1. Geometrieachse \triangleq Koordinatenachse X (Abszisse) des WKS 2: 2. Geometrieachse \triangleq Koordinatenachse Y (Ordinate) des WKS 3: 3. Geometrieachse \triangleq Koordinatenachse Z (Applikate) des WKS
<Kanalachse>	Name der Kanalachse, die in den Geometrieachsverbund aufgenommen werden soll.

Beispiele

Beispiel 1: Zwei Achsen wechselweise als Geometrieachse schalten

Ein Werkzeugschlitten kann über die Kanalachsen X1, Y1, Z1, Z2 verfahren werden:



Die Geometrieachsen sind so projiziert, dass nach dem Einschalten zunächst Z1 als 3. Geometrieachse unter dem Geometrieachsennamen "Z" wirksam ist und zusammen mit X1 und Y1 den Geometrieachsverbund bildet.

Im Teileprogramm sollen nun die Achsen Z1 und Z2 wechselweise als Geometrieachse Z zum Einsatz kommen:

Programmcode	Kommentar
...	
N100 GEOAX(3,Z2)	; Als 3. Geometrieachse (Z) fungiert Kanalachse Z2.
N110 G1 ...	
N120 GEOAX(3,Z1)	; Als 3. Geometrieachse (Z) fungiert Kanalachse Z1.
...	

Beispiel 2: Umschalten der Geometrieachsen bei 6 Kanalachsen

Eine Maschine besitzt 6 Kanalachsen mit den Namen XX, YY, ZZ, U, V, W.

Die Grundeinstellung der Geometrieachskonfiguration über Maschinendaten ist:

Kanalachse XX = 1. Geometrieachse (X-Achse)

Kanalachse YY = 2. Geometrieachse (Y-Achse)

Kanalachse ZZ = 3. Geometrieachse (Z-Achse)

Programmcode	Kommentar
N10 GEOAX()	; Grundkonfiguration der Geometrieachsen ist wirksam.
N20 G0 X0 Y0 Z0 U0 V0 W0	; Alle Achsen im Eilgang auf Position 0.
N30 GEOAX(1,U,2,V,3,W)	; Kanalachse U wird zur ersten (X), V zur zweiten (Y) ; und W zur dritten Geometrieachse (Z).
N40 GEOAX(1,XX,3,ZZ)	; Kanalachse XX wird zur ersten (X), ZZ zur dritten ; Geometrieachse (Z). Kanalachse V bleibt zweite ; Geometrieachse (Y).

Programmcode	Kommentar
N50 G17 G2 X20 I10 F1000	; Vollkreis in der X/Y-Ebene. Es fahren die ; Kanalachsen XX und V.
N60 GEOAX(2,W)	; Kanalachse W wird zweite Geometrieachse (Y).
N80 G17 G2 X20 I10 F1000	; Vollkreis in der X/Y-Ebene. Es fahren die ; Kanalachsen XX und W.
N90 GEOAX()	; Zurücksetzen auf Grundzustand.
N100 GEOAX(1,U,2,V,3,W)	; Kanalachse U wird zur ersten (X), V zur zweiten ; (Y) und W zur dritten Geometrieachse (Z).
N110 G1 X10 Y10 Z10 XX=25	; Kanalachsen U, V, W fahren jeweils auf die ; Position 10. XX als Zusatzachse fährt auf Position ; 25.
N120 GEOAX(0,V)	; V wird aus Geometrieachsverbund herausgenommen. ; U und W sind weiterhin erste (X) und dritte ; Geometrieachse (Z). ; Die zweite Geometrieachse (Y) bleibt unbelegt.
N130 GEOAX(1,U,2,V,3,W)	; Kanalachse U bleibt erste (X), V wird zur ; zweiten (Y), W bleibt dritte Geometrieachse (Z).
N140 GEOAX(3,V)	; V wird zur dritten Geometrieachse (Z), wobei W ; überschrieben und damit aus dem Geometrieachsver- ; bund ; herausgenommen wird. Die zweite Geometrieachse (Y) ; ist nach wie vor unbelegt.

Maschinendaten

Achskonfiguration

Zuordnung von Geometrie-, Zusatz- und Maschinenachsen zu Kanalachsen:

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB
- MD20070 \$MC_AXCONF_MACHAX_USED
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX

Reset-Verhalten

Reset-Verhalten von geänderten Geometrieachszuordnungen:

- MD20110 \$MC_RESET_MODE_MASK, Bit 12
- MD20118 \$MC_GEOAX_CHANGE_RESET

NC-Start-Verhalten

- MD20112 \$MC_START_MODE_MASK, Bit 12

Mitteilung an PLC-Anwenderprogramm

Parametriermöglichkeit des M-Befehls der bei einer Geometrieachsumschaltung an die NC/ PLC-Nahtstelle ausgegeben wird:

- MD22532 \$MC_GEOAX_CHANGE_M_CODE

Randbedingungen

Keine Geometrieachsumschaltung

- Ist eine der folgenden Funktionen aktiv, ist keine Geometrieachsumschaltung möglich:
 - Transformation
 - Spline-Interpolation
 - Werkzeugradiuskorrektur
 - Werkzeugfeinkorrektur
- Die Geometrieachse und eine weitere Kanalachse haben den gleichen Namen.
- Eine der an der Geometrieachsumschaltung beteiligten Achsen ist an einer Aktion beteiligt, die über Satzgrenzen hinweg andauert. Z. B. Satzübergreifende Positionierachse oder Folgeachse einer Achskopplung.

Rundachsen

Rundachsen können nicht zu Geometrieachsen gemacht werden.

Achszustand nach dem Ersetzen

Eine durch die Umschaltung im Geometrieachsverbund ersetzte Achse ist nach dem Umschaltvorgang über ihren Kanalachsenamen als Zusatzachse programmierbar.

Frames, Schutzbereiche, Arbeitsfeldbegrenzungen

Mit dem Umschalten der Geometrieachsen werden alle Frames, Schutzbereiche und Arbeitsfeldbegrenzungen gelöscht.

Polarkoordinaten

Ein Tausch der Geometrieachsen mit `GEOAX` setzt analog einem Ebenenwechsel mit `G17-G19` die modalen Polarkoordinaten auf den Wert 0.

DRF, NPV

Eine eventuelle Handrad-Verschiebung (DRF) oder eine externe Nullpunktverschiebung (NPV) bleibt nach der Umschaltung wirksam.

Grundkonfiguration der Geometrieachsen

Der Befehl `GEOAX()` ruft die Grundkonfiguration des Geometrieachsverbunds auf.

Nach POWER ON und bei Umschalten in die Betriebsart "Referenzpunktfahren" wird automatisch auf die Grundkonfiguration zurückgeschaltet.

Werkzeuglängenkorrektur

Eine aktive Werkzeuglängenkorrektur ist auch nach dem Umschaltvorgang wirksam. Sie gilt jedoch für die neu aufgenommenen bzw. positionsgetauschten Geometrieachsen als noch nicht herausgefahren. Beim ersten Bewegungsbefehl für diese Geometrieachsen besteht der resultierende Verfahrweg dementsprechend aus der Summe von Werkzeuglängenkorrektur und programmiertem Verfahrweg.

Geometrieachsen, die bei einer Umschaltung ihre Position im Achsverband beibehalten, behalten auch ihren Status bezüglich der Werkzeuglängenkorrektur.

Geometrieachskonfiguration bei aktiver Transformation

- Die für eine aktive Transformation über Transformations-Maschinendaten parametrisierte Geometrieachskonfiguration ist über die Funktion "Umschaltbare Geometrieachsen" nicht änderbar.
- Unterschiedliche Geometrieachskonfiguration für eine Transformation, müssen in den Transformations-Maschinendaten unterschiedliche Datensätze parametrisiert werden.
- Eine über `GEOAX` geänderte Geometrieachskonfiguration wird durch Aktivierung einer Transformation gelöscht.
- Bezüglich der Geometrieachsen haben die Transformations-spezifischen Geometrieachskonfigurationen von aktiven Transformationen Vorrang vor den die Umschaltung von Geometrieachsen betreffenden Parametrisierungen.
Beispiel: Eine Transformation ist aktiv. Laut Maschinendaten soll die Transformation bei Kanal-Reset erhalten bleiben. Gleichzeitig soll bei einem Kanal-Reset jedoch die Grundkonfiguration der Geometrieachsen hergestellt werden. Es bleibt die Geometrieachskonfiguration erhalten, die für die Transformation festgelegt ist.
- Mit dem Ausschalten einer Transformation wird wieder die parametrisierte Grundeinstellung der Geometrieachskonfiguration wirksam.

Betriebsart JOG, Maschinenfunktion REF

Beim Umschalten in die Betriebsart JOG, Maschinenfunktion REF (Referenzpunkt fahren), wird die in den Maschinendaten parametrisierte Geometrieachskonfiguration wirksam..

17.3 Achscontainer (AXCTSWE, AXCTSWED, AXCTSWEC)

Über die Befehle "AXCTSWE" bzw. "AXCTSWED" wird die Drehung des angegebenen Achscontainers freigegeben.

Über den Befehl "AXCTSWEC" wird eine bereits gegebene Freigabe zur Achscontainer-Drehung wieder zurückgenommen.

Syntax

```
AXCTSWE (<ID>)  
AXCTSWED (<ID>)  
AXCTSWEC (<ID>)
```

Bedeutung

AXCTSWE:	Freigabe zum Drehen des Achscontainers Die Programmverarbeitung wird durch "AXCTSWE" nicht angehalten. Die Drehung erfolgt, sobald alle am Achscontainer beteiligten Kanäle die Freigabe erteilt haben.	
AXCTSWED:	Freigabe zum Drehen des Achscontainers ohne Berücksichtigung der anderen am Achscontainer beteiligten Kanäle Hinweis <ul style="list-style-type: none"> • Befehlsvariante zur Vereinfachung der Inbetriebnahme des Teileprogramms bzw. Synchronaktion. • Das Verhalten bezüglich der anderen am Achscontainer beteiligten Kanäle, kann vorgegeben werden über: MD12760 \$MN_AXCT_FUNCTION_MASK, Bit 0 	
AXCTSWEC:	Rücknahme der Freigabe zum Drehen des Achscontainers Hinweis Die Freigabe zum Drehen des Achscontainers kann nur zurück genommen werden, wenn die Drehung noch nicht begonnen wurde: \$AN_AXCTSWA[<Achscontainer>] == 0 Zu Systemvariable siehe "Achscontainer (AXCTSWE, AXCTSWED, AXCTSWEC) (Seite 565)"	
<ID>:	Bezeichner des Achscontainers oder einer Containerachse:	
	CT<Nummer>:	Defaultbezeichner eines Achscontainers: MD12750 \$MN_AXCT_NAME_TAB Beispiel: "CT1"
	<Container>:	Anwenderspezifischer Bezeichner eines Achscontainers: MD12750 \$MN_AXCT_NAME_TAB Beispiel: "CONTAINER_1"
	<Achse>:	Bezeichner einer im Kanal bekannten Containerachse

Hinweis

Schrittweite

Die Schrittweite einer Achscontainer-Drehung wird eingestellt über das Settingdatum:

SD41700 \$SN_AXCT_SWWIDTH

Weitere Informationen

Diagnose

Der aktuelle Status eines Achscontainers kann über folgende Systemvariable gelesen werden:

Systemvariable	Typ	Beschreibung
\$AC_AXCTSWA[<Name>]	BOOL	Kanal-spezifischer Status des Achscontainers
\$AN_AXCTSWA[<Achscontainer>]	BOOL	NCU-spezifischer Status des Achscontainer

Systemvariable	Typ	Beschreibung
\$AN_AXCTSWE[<Achscontainer>]	INT	Slot-spezifischer Status der Achscontainer-Drehung Die Systemvariable liefert Bit-weise den Status der Slots des Achscontainers. Jedes Bit entspricht einem Slot.
\$AN_AXCTAS[<Achscontainer>]	INT	Anzahl der Plätze (Slots) um die der Achscontainer aktuell weitergeschaltet wurde.

Achscontainer-Drehung mit implizitem GET / GETD

Über nachfolgendes Maschinendatum kann eingestellt werden, dass mit dem Befehl "AXCTSWE" alle Container-Achsen des Kanals mittels impliziten "GET / GETD" in den Kanal geholt werden. Ein Achstausch ist erst nach erfolgter Container-Drehung wieder möglich.

MD10722 \$MN_AXCHANGE_MASK, Bit 1 = 1

Hinweis

Die Achscontainer-Drehung mit impliziten "GET / GETD" wird **nicht** für eine Achse im Zustand "Hauptlaufachse" (z. B. PLC-Achse) durchgeführt, da die Achse für die Achscontainer-Drehung den Zustand verlassen müsste.

17.4 Warten auf gültige Achsposition (WAITENC)

Mit dem Sprachbefehl "WAITENC" kann im NC-Programm gewartet werden, bis für die mit MD34800 \$MA_WAIT_ENC_VALID = 1 projektierten Achsen synchronisierte bzw. restaurierte Achspositionen zur Verfügung stehen.

Im Wartezustand kann eine Unterbrechung erfolgen, z. B. durch Start eines ASUPs oder durch Betriebsartenwechsel nach JOG. Mit der Programmfortsetzung wird der Wartezustand ggf. wieder eingenommen.

Hinweis

Der Wartezustand wird in der Bedienoberfläche durch den Haltezustand "Warten auf Mess-System" angezeigt.

Syntax

"WAITENC" kann im Programmteil eines beliebigen NC-Programms programmiert werden.

Die Programmierung muss in einem eigenen Satz erfolgen:

```
...
WAITENC
...
```

Beispiel

"WAITENC" wird z. B. im ereignisgesteuerten Anwenderprogramm .../_N_CMA_DIR/_N_PROG_EVENT_SPF verwendet, wie das folgende Anwendungsbeispiel zeigt.

Anwendungsbeispiel: Werkzeugrückzug nach POWER OFF mit Orientierungstransformation

Eine Bearbeitung mit Werkzeugorientierung wurde durch Spannungsausfall abgebrochen. Beim anschließenden Hochlauf wird das ereignisgesteuerte Anwenderprogramm .../_N_CMA_DIR/_N_PROG_EVENT_SPF aufgerufen.

Im ereignisgesteuerten Anwenderprogramm wird mit "WAITENC" auf synchronisierte bzw. restaurierte Achspositionen gewartet, um danach einen Frame berechnen zu können, der das WKS in Werkzeugrichtung ausrichtet.

Programmcode	Kommentar
...	
IF \$P_PROG_EVENT == 4	; Hochlauf.
IF \$P_TRAFO <> 0	; Transformation wurde angewählt.
WAITENC	; Warten auf gültige Achspositionen der Orientierungsachsen.
TOROTZ	; Z-Achse des WKS in Richtung der Werkzeugachse drehen.
ENDIF	
M17	
ENDIF	
...	

Danach kann das Werkzeug in der Betriebsart JOG durch eine Rückzugsbewegung in Richtung der Werkzeugachse freigefahren werden.

17.5 Programmierbare Parametersatzumschaltung (SCPARA)

Mit dem Befehl "SCPARA" kann für eine Achse die Umschaltung auf einen bestimmten Parametersatz angefordert werden.

Hinweis

Keine Parametersatzumschaltung während Gewindebearbeitung

Bei Gewindeschneiden G33 und Gewindebohren G331 / G332 wird der Parametersatz von der Steuerung ausgewählt und kann nicht verändert werden.

Gesperrte Parametersatzumschaltung

Eine Parametersatzumschaltung kann auch über die NC/PLC-Nahstelle angefordert werden. Um Umschaltkonflikte zu vermeiden, kann über die NC/PLC-Nahstelle die Parametersatzumschaltung der NC (SCPARA) gesperrt werden:

DB31, ... DBX9.3 (Parametersatzvorgabe durch NC gesperrt)

Hinweis

Wird eine Parametersatzumschaltung durch "SCPARA" angefordert, während die Parametersatzumschaltung über die NC/PLC-Nahstelle gesperrt ist, wird die Umschaltung ohne Fehlermeldung abgewiesen.

Syntax

SCPARA[<Achse>] = <Wert>

Bedeutung

SCPARA:	Befehl: Parametersatz umschalten	
<Achse>:	Achsbezeichner (Kanalachse)	
	Typ:	AXIS
<Wert>:	Parametersatznummer: 1, 2, 3, ... max. Parametersatznummer	

Beispiel

Programmcode	Kommentar
...	
N110 SCPARA[X] = 3	; Anwahl: Achse X, 3. Parametersatz
...	

Weitere Informationen

Freigabe der Parametersatzumschaltung

Die Parametersatzumschaltung der Achse muss explizit freigegeben werden:

MD35590 \$MA_PARAMSET_CHANGE_ENABLE[<Achse>]

Parametersatznummer lesen

Die Nummer des angewählten Parametersatzes (Soll-Parametersatz) kann über die Systemvariable \$AA_SCPAR gelesen werden.

Literatur

Ausführliche Informationen zu den Parametersätzen finden sich in:

Funktionshandbuch Grundfunktionen; Kapitel "Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2)" > "Regelung" > "Parametersätze des Lagereglers"

17.6 Vorhandenen NC-Sprachumfang prüfen (STRINGIS)

Mit der Funktion "STRINGIS(...)" kann geprüft werden, ob der angegebene String als Element der NC-Programmiersprache im aktuellen Sprachumfang zur Verfügung steht.

Definition

INT STRINGIS (STRING <Name>)

Syntax

STRINGIS (<Name>)

Bedeutung

STRINGIS:	Funktion mit Rückgabewert
<Name>:	Name des zu prüfenden Elementes der NC-Programmiersprache
Rückgabewert:	Das Format des Rückgabewertes ist yxx (Dezimal).

Elemente der NC-Programmiersprache

Folgende Elemente der NC-Programmiersprache können geprüft werden:

- G-Codes aller existierenden G-Funktionsgruppen z.B. "G0", "INVCW", "POLY", "ROT", "KONT", "SOFT", "CUT2D", "CDON", "RMBBL", "SPATH"
- DIN- oder NC-Adressen wie z.B. "ADIS", "RNDM", "SPN", "SR", "MEAS"
- Funktionen z.B. "TANG(...)" oder "GETMDACT"
- Prozeduren z.B. "SBLOF".
- Schlüsselworte z.B. "ACN", "DEFINE" oder "SETMS"
- Systemdaten z.B. Maschinendaten \$M... , Settingdaten \$S... oder Optionsdaten \$O...
- Systemvariable \$A... , \$V... , \$P...
- Rechenparameter R...
- Zyklennamen von aktivierten Zyklen
- GUD- und LUD-Variablen
- Makro-Namen
- Label-Namen

Rückgabewert

Der Rückgabewert ist nur in den erst 3 Dezimalstellen relevant. Das Format des Rückgabewertes ist yxx, mit y = Basisinformation und xx = Detailinformation.

Rückgabewert	Bedeutung
000	Der String 'name' ist im vorliegenden System nicht bekannt ¹⁾
100	Der String 'name' ist ein Element der NC-Programmiersprache, aber aktuell nicht programmierbar (Option/Funktion ist inaktiv)

17.6 Vorhandenen NC-Sprachumfang prüfen (STRINGIS)

Rückgabewert	Bedeutung																										
2xx	Der String 'name' ist ein programmierbares Element der NC-Programmiersprache (Option/Funktion ist aktiv). Die Detailinformation xx enthält weitere Informationen über die Art des Elements:																										
	<table border="1"> <thead> <tr> <th>xx</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>01</td> <td>DIN-Adresse oder NC-Adresse ²⁾</td> </tr> <tr> <td>02</td> <td>G-Code (z.B. G04, INVCW)</td> </tr> <tr> <td>03</td> <td>Funktion mit Rückgabewert</td> </tr> <tr> <td>04</td> <td>Funktion ohne Rückgabewert</td> </tr> <tr> <td>05</td> <td>Schlüsselwort (z.B. DEFINE)</td> </tr> <tr> <td>06</td> <td>Maschinen- (\$M...), Setting- (\$S...) oder Optionsdatum (\$O...)</td> </tr> <tr> <td>07</td> <td>Systemparameter, z.B. Systemvariable (\$...) oder Rechenparameter (R...)</td> </tr> <tr> <td>08</td> <td>Zyklus (Der Zyklus muss im NCK geladen und die Zyklenprogramme aktiv sein ³⁾)</td> </tr> <tr> <td>09</td> <td>GUD-Variable (Die GUD-Variable muss der in GUD-Definitionsdateien definierten und die GUD-Variablen aktiviert sein)</td> </tr> <tr> <td>10</td> <td>Makroname (Das Makro muss in der Makro-Definitionsdateien definierten und Makros aktivierten sein) ⁴⁾</td> </tr> <tr> <td>11</td> <td>LUD-Variable des aktuellen Teileprogramms</td> </tr> <tr> <td>12</td> <td>ISO G-Code (ISO Sprachmodus muss aktiv ist)</td> </tr> </tbody> </table>	xx	Bedeutung	01	DIN-Adresse oder NC-Adresse ²⁾	02	G-Code (z.B. G04, INVCW)	03	Funktion mit Rückgabewert	04	Funktion ohne Rückgabewert	05	Schlüsselwort (z.B. DEFINE)	06	Maschinen- (\$M...), Setting- (\$S...) oder Optionsdatum (\$O...)	07	Systemparameter, z.B. Systemvariable (\$...) oder Rechenparameter (R...)	08	Zyklus (Der Zyklus muss im NCK geladen und die Zyklenprogramme aktiv sein ³⁾)	09	GUD-Variable (Die GUD-Variable muss der in GUD-Definitionsdateien definierten und die GUD-Variablen aktiviert sein)	10	Makroname (Das Makro muss in der Makro-Definitionsdateien definierten und Makros aktivierten sein) ⁴⁾	11	LUD-Variable des aktuellen Teileprogramms	12	ISO G-Code (ISO Sprachmodus muss aktiv ist)
xx	Bedeutung																										
01	DIN-Adresse oder NC-Adresse ²⁾																										
02	G-Code (z.B. G04, INVCW)																										
03	Funktion mit Rückgabewert																										
04	Funktion ohne Rückgabewert																										
05	Schlüsselwort (z.B. DEFINE)																										
06	Maschinen- (\$M...), Setting- (\$S...) oder Optionsdatum (\$O...)																										
07	Systemparameter, z.B. Systemvariable (\$...) oder Rechenparameter (R...)																										
08	Zyklus (Der Zyklus muss im NCK geladen und die Zyklenprogramme aktiv sein ³⁾)																										
09	GUD-Variable (Die GUD-Variable muss der in GUD-Definitionsdateien definierten und die GUD-Variablen aktiviert sein)																										
10	Makroname (Das Makro muss in der Makro-Definitionsdateien definierten und Makros aktivierten sein) ⁴⁾																										
11	LUD-Variable des aktuellen Teileprogramms																										
12	ISO G-Code (ISO Sprachmodus muss aktiv ist)																										
400	Der String 'name' ist eine NC-Adresse, die nicht als xx == 01 oder xx == 10 erkannt wurde und die nicht G oder R ist ²⁾																										
y00	Keine spezifische Zuordnung möglich																										
<p>1) Steuerungs-abhängig ist unter Umständen nur eine Untermenge der Siemens NC-Sprachbefehle bekannt, z.B. SINUMERIK 802D sl. Auf diesen Steuerungen wird für Strings, die prinzipiell Siemens NC-Sprachbefehle sind, der Wert 0 zurückgegeben. Dieses Verhalten kann über MD10711 \$MN_NC_LANGUAGE_CONFIGURATION verändert werden. Bei MD10711 = 1 wird dann für Siemens NC-Sprachbefehle immer der Wert 100 zurückgegeben.</p> <p>2) NC-Adressen sind folgende Buchstaben: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Diese NC-Adressen können auch mit einer Adresserweiterung programmiert werden. Die Adresserweiterung kann bei der Prüfung mit STRINGIS angegeben werden. Beispiel: 201 == STRINGIS("A1"). Die Buchstaben: D, F, H, L, M, N, O, P, S, T sind NC-Adressen oder Hilfsfunktionen die anwenderdefiniert verwendet werden. Für sie wird immer der Wert 400 zurückgegeben. Beispiel: 400 == STRINGIS("D"). Diese NC-Adressen können bei der Prüfung mit STRINGIS nicht mit Adresserweiterung angegeben werden. Beispiel: 000 == STRINGIS("M02"), aber 400 == STRINGIS("M").</p> <p>3) Namen von Zyklenparametern können mit STRINGIS nicht geprüft werden.</p> <p>4) Als Makro definierte Adress z.B. G, H, M, L werden als Makro identifiziert</p>																											

Beispiele

In den folgenden Beispielen wird angenommen, dass die als String angegebenen NC-Sprachelement, sofern nicht besonders vermerkt, in der Steuerung prinzipiell programmierbar sind.

1. Der String "T" ist als Hilfsfunktion definiert:

```
400 == STRINGIS ("T")
000 == STRINGIS ("T3")
```

2. Der String "X" ist als Achse definiert:

```
201 == STRINGIS ("X")
201 == STRINGIS ("X1")
```

3. Der String "A2" ist als NC-Adresse mit Erweiterung definiert:

```
201 == STRINGIS ("A")
201 == STRINGIS ("A2")
```

17.6 Vorhandenen NC-Sprachumfang prüfen (STRINGIS)

4. Der String "INVCW" ist als benannter G-Code definiert:
202 == STRINGIS("INVCW")
5. Der String "\$MC_GCODE_RESET_VALUES" ist als Maschinendatum definiert:
206 == STRINGIS("\$MC_GCODE_RESET_VALUES")
6. Der String "GETMDACT" ist eine NC-Sprachfunktion:
203 == STRINGIS("GETMDACT ")
7. Der String "DEFINE" ist ein Schlüsselwort:
205 == STRINGIS("DEFINE")
8. Der String "\$TC_DP3" ist ein Systemparameter (Werkzeuiglängenkomponente):
207 == STRINGIS("\$TC_DP3")
9. Der String "\$TC_TP4" ist ein Systemparameter (Werkzeuggröße):
207 == STRINGIS("\$TC_TP4")
10. Der String "\$TC_MPP4" ist ein Systemparameter (Magazinplattzustand):
 - Die Werkzeugmagazin-Verwaltung ist aktiv: 207 == STRINGIS("\$TC_MPP4") ;
 - Die Werkzeugmagazin-Verwaltung ist nicht aktiv: 000 == STRINGIS("\$TC_MPP4")Siehe auch unten Absatz: Werkzeugmagazin-Verwaltung.
11. Der String "MACHINERY_NAME" ist als GUD-Variable definiert:
209 == STRINGIS("MACHINERY_NAME")
12. Der String "LONGMACRO" ist als Makro definiert:
210 == STRINGIS("LONGMACRO")
13. Der String "MYVAR" ist als LUD-Variable definiert:
211 == STRINGIS("MYVAR")
14. Der String "XYZ" ist kein im NCK bekannter Befehl, GUD-Variable, Makro- oder Zyklus-Name:
000 == STRINGIS("XYZ")

Werkzeugmagazin-Verwaltung

Ist die Funktion Werkzeugmagazin-Verwaltung nicht aktiv, liefert STRINGIS für die Systemparameter der Werkzeugmagazin-Verwaltung, unabhängig vom Maschinendatum

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION

immer den Wert 000.

ISO Modus

Ist die Funktion "ISO Modus" aktiv:

- MD18800 \$MN_MM_EXTERN_LANGUAGE (Aktivierung externer NC-Sprachen)
- MD10880 \$MN_MM_EXTERN_CNC_SYSTEM (zu adaptierendes Steuerungssystem)

überprüft STRINGIS den angegebenen String zuerst als SINUMERIK G-Code. Ist der String kein SINUMERIK G-Code wird er anschließend als ISO G-Code überprüft.

Programmierte Umschaltungen (G290 (SINUMERIK Mode), G291 (ISO Mode)) haben auf STRINGIS keine Auswirkung.

Beispiel

Die für die Funktion STRINGIS(...) relevanten Maschinendaten haben folgende Werte:

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION = 2 (Es werden nur die NC-Sprachbefehle als bekannt angesehen, deren Optionen gesetzt sind)
- MD19410 \$ON_TRAFO_TYPE_MASK = 'H0' (Option: Transformationen)
- MD10700 \$MN_PREPROCESSING_LEVEL='H43' (Vorverarbeitung für Zyklen aktiv)

Das folgende Beispielprogramm wird ohne Fehlermeldung abgearbeitet:

Programmcode	Kommentar
N1 R1=STRINGIS("TRACYL")	; R1 == 0, da TRACYL wegen der fehlenden Transformations-Option als "nicht bekannt" erkannt wird
N2 IF STRINGIS("TRACYL") == 204	
N3 TRACYL(1,2,3)	; N3 wird übersprungen
N4 ELSE	
N5 G00	;und stattdessen N5 ausgeführt
N6 ENDIF	
N7 M30	

17.7 Fenster aus dem Teileprogramm interaktiv aufrufen (MMC)

Über den Befehl "MMC" können aus dem Teileprogramm auf dem HMI anwenderdefinierte Dialogfenster (Dialogbilder) angezeigt werden.

Das Aussehen der Dialogfenster wird durch rein textuelle Projektierung festgelegt (COM-Datei im Zyklenverzeichnis), die HMI -System-Software bleibt dabei unverändert.

Anwenderdefinierte Dialogfenster können nicht zeitgleich in verschiedenen Kanälen aufgerufen werden.

Syntax

MMC (<Kommando>, <Quittungsmodus>)

Bedeutung

MMC:	Unterprogrammbezeichner	
<Kommando>:	Parameter vom Typ STRING	
	Enthält das MMC-Kommando, z. B. in folgender Form: "CYCLES, PICTURE_ON, T_SK.COM, BILD, MGUD.DEF, BILD_3.AWB, TEST_1, A1"	
	CYCLES:	Bedienbereich, in dem die projektierten Anwenderdialoge ausgeführt werden.
	PICTURE_ON: bzw. PICTURE_OFF:	Befehl: Bildanwahl bzw. Bildabwahl
	T_SK.COM:	Com-Datei: Name der Dialogbild-Datei (Anwenderzyklen). Hier ist das Aussehen der Dialogbilder festgelegt. Im Dialogbild können Anwendervariablen und/oder Kommentartexte angezeigt werden.
	BILD:	Dialogbildname: Die einzelnen Bilder werden durch den Dialogbildnamen ausgewählt.
	MGUD.DEF:	Anwenderdatendefinitionsdatei, auf die beim Lesen/Schreiben von Variablen zugegriffen wird.
	BILD_3.AWB:	Grafikdatei
	TEST_1:	Anzeigezeit oder Quittungsvariable
	A1:	Textvariablen...",
<Quittungsmodus>:	Parameter vom Typ CHAR	
	Wert:	"N": ohne Quittung Die Programmbearbeitung wird nach Absenden des Kommandos fortgeführt. Der Absender wird nicht benachrichtigt, wenn das Kommando nicht erfolgreich ausgeführt werden kann.
		"S": synchrone Quittung Die Programmabarbeitung wird solange angehalten, bis die Empfängerkomponente das Kommando quittiert hat. Bei positiver Quittung wird der nächste Befehl abgearbeitet. Bei negativer Quittung wird eine Fehlermeldung ausgegeben.
		"A": asynchrone Quittung Die Programmabarbeitung wird nach Absetzen des Befehls fortgeführt. Die Quittung wird in einer Quittungsvariablen (vordefinierte Systemvariable) abgelegt und ist explizit aus dem Programm abzufragen. Der auf den Quittungsmodus folgende Parameter ist die Nummer der Quittungsvariablen.
Wird kein <Quittungsmodus> programmiert, wird als Standardeinstellung die synchrone Quittung verwendet.		

17.8 Programmlaufzeit / Werkstückzähler

17.8.1 Programmlaufzeit / Werkstückzähler (Übersicht)

Zur Unterstützung des Werkzeugmaschinenbedieners werden Informationen zur Programmlaufzeit und Werkstückzahl bereitgestellt.

Diese Informationen können als Systemvariablen im NC- und/oder PLC-Programm bearbeitet werden. Gleichzeitig stehen diese Informationen für die Anzeige auf der Bedienoberfläche zur Verfügung.

17.8.2 Programmlaufzeit

Die Funktion "Programmlaufzeit" stellt NC-interne Timer zur Überwachung technologischer Prozesse zur Verfügung, die über NC- und Kanal-spezifische Systemvariablen im Teileprogramm und in Synchronaktionen gelesen werden können.

Der Trigger zur Laufzeitmessung (\$AC_PROG_NET_TIME_TRIGGER) ist die einzige schreibbare Systemvariable der Funktion und dient zur selektiven Messung von Programmabschnitten. D. h. durch Beschreiben des Triggers im NC-Programm kann die Zeitmessung ein- und wieder ausgeschaltet werden.

Systemvariable	Bedeutung	Aktivität
NC-spezifisch		
\$AN_SETUP_TIME	Zeit seit dem letzten Steuerungshochlauf mit Standardwerten ("Kaltstart") in Minuten Wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.	• immer aktiv
\$AN_POWERON_TIME	Zeit seit dem letzten Normalhochlauf der Steuerung ("Warmstart") in Minuten Wird bei jedem Normalhochlauf der Steuerung automatisch auf "0" zurückgesetzt.	
Kanal-spezifisch		

Systemvariable	Bedeutung	Aktivität
\$AC_OPERATING_TIME	Gesamtlaufzeit von NC-Programmen in der Betriebsart Automatik in Sekunden Der Wert wird mit jedem Steuerungshochlauf automatisch auf "0" zurückgesetzt.	<ul style="list-style-type: none"> • Aktivierung über MD27860 • nur Betriebsart AUTOMATIK
\$AC_CYCLE_TIME	Laufzeit des angewählten NC-Programms in Sekunden Der Wert wird mit dem Start eines neuen NC-Programms automatisch auf "0" zurückgesetzt.	
\$AC_CUTTING_TIME	Bearbeitungszeit in Sekunden Gemessen wird die Laufzeit der Bahnachsen (mindestens eine ist aktiv) ohne aktiven Eilgang in allen NC-Programmen zwischen NC-Start und Programmende / NC-Reset. Die Messung wird zusätzlich bei aktiver Verweilzeit unterbrochen. Der Wert wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.	
\$AC_ACT_PROG_NET_TIME	Aktuelle Netto-Laufzeit des aktuellen NC-Programms in Sekunden Wird mit dem Start eines NC-Programms automatisch auf "0" zurückgesetzt.	<ul style="list-style-type: none"> • immer aktiv • nur Betriebsart AUTOMATIK
\$AC_OLD_PROG_NET_TIME	Netto-Laufzeit des gerade korrekt mit M30 beendeten Programms in Sekunden	
\$AC_OLD_PROG_NET_TIME_COUNT	Änderungen auf \$AC_OLD_PROG_NET_TIME Nach POWER ON steht \$AC_OLD_PROG_NET_TIME_COUNT auf "0". \$AC_OLD_PROG_NET_TIME_COUNT wird immer dann erhöht, wenn die Steuerung \$AC_OLD_PROG_NET_TIME neu geschrieben hat.	

Systemvariable	Bedeutung	Aktivität
\$AC_PROG_NET_TIME_TRIGGER	Trigger zur Laufzeitmessung:	<ul style="list-style-type: none"> • nur Betriebsart AUTOMATIK
	0 Neutraler Zustand Der Trigger ist nicht aktiv.	
	1 Beenden Beendet die Messung und kopiert den Wert aus \$AC_ACT_PROG_NET_TIME in \$AC_OLD_PROG_NET_TIME. \$AC_ACT_PROG_NET_TIME wird auf "0" gesetzt und läuft danach weiter.	
	2 Start Startet die Messung und setzt dabei \$AC_ACT_PROG_NET_TIME auf "0". \$AC_OLD_PROG_NET_TIME wird nicht verändert.	
	3 Stopp Stoppt die Messung. Verändert \$AC_OLD_PROG_NET_TIME nicht und hält \$AC_ACT_PROG_NET_TIME bis zum Fortsetzen konstant.	
4 Fortsetzen Fortsetzen der Messung, d. h. eine vorher gestoppte Messung wird wieder aufgenommen. \$AC_ACT_PROG_NET_TIME läuft weiter. \$AC_OLD_PROG_NET_TIME wird nicht verändert.		
Durch POWER ON werden alle Systemvariablen auf "0" zurückgesetzt!		
Literatur: Eine ausführliche Beschreibung der aufgelisteten Systemvariablen findet sich in: Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1), Kapitel: Programmlaufzeit		

Hinweis

Maschinenhersteller

Das Einschalten der aktivierbaren Timer erfolgt über das Maschinendatum MD27860 \$MC_PROCESSTIMER_MODE.

Das Verhalten der aktiven Zeitmessungen bei bestimmten Funktionen (z. B. GOTOS, Override = 0%, aktiver Probelaufvorschub, Programmtest, ASUP, PROG_EVENT, ...) wird konfiguriert über die Maschinendaten MD27850 \$MC_PROG_NET_TIMER_MODE und MD27860 \$MC_PROCESSTIMER_MODE.

Literatur:

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1), Kapitel: Programmlaufzeit

Hinweis

Restzeit für ein Werkstück

Wenn nacheinander gleiche Werkstücke produziert werden, kann aus den Timerwerten:

- Bearbeitungszeit für das zuletzt produzierte Werkstück (siehe \$AC_OLD_PROG_NET_TIME)

und

- aktuelle Bearbeitungszeit (siehe \$AC_ACT_PROG_NET_TIME)

die verbleibende Restzeit für ein Werkstück ermittelt werden.

Die Restzeit wird zusätzlich zur aktuellen Bearbeitungszeit auf der Bedienoberfläche angezeigt.

Hinweis

Verwendung von STOPRE

Die Systemvariablen \$AC_OLD_PROG_NET_TIME und \$AC_OLD_PROG_NET_TIME_COUNT erzeugen keinen impliziten Vorlaufstopp. Bei der Verwendung im Teileprogramm ist das unkritisch, wenn der Wert der Systemvariablen aus dem vorangegangenen Programmlauf stammt. Wenn aber der Trigger zur Laufzeitmessung (\$AC_PROG_NET_TIME_TRIGGER) hochfrequent geschrieben wird und sich dadurch \$AC_OLD_PROG_NET_TIME sehr oft ändert, dann sollte im Teileprogramm ein explizites STOPRE verwendet werden.

Randbedingungen

- **Satzsuchlauf**
Bei Satzsuchlauf werden keine Programmlaufzeiten ermittelt.
- **REPOS**
Die Zeitdauer eines REPOS-Vorgangs wird der aktuellen Bearbeitungszeit (\$AC_ACT_PROG_NET_TIME) angerechnet.

Beispiele

Beispiel 1: Zeitdauer von "mySubProgrammA" messen

```
Programmcode
...
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
```

Nachdem das Programm die Zeile N80 verarbeitet hat, steht in \$AC_OLD_PROG_NET_TIME die Nettolaufzeit von "mySubProgrammA".

Der Wert von \$AC_OLD_PROG_NET_TIME:

- bleibt über M30 hinaus erhalten.
- wird nach jedem Schleifendurchlauf aktualisiert.

Beispiel 2: Zeitdauer von "mySubProgrammA" und "mySubProgrammC" messen

```

Programmcode
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
    
```

17.8.3 Werkstückzähler

Die Funktion "Werkstückzähler" stellt diverse Zähler zur Verfügung, die insbesondere für die steuerungsinterne Zählung von Werkstücken verwendet werden können.

Die Zähler existieren als kanalspezifische Systemvariablen mit Schreib- und Lese-Zugriff im Wertebereich von 0 bis 999 999 999.

Systemvariable	Bedeutung
\$AC_REQUIRED_PARTS	Anzahl der zu fertigenden Werkstücke (Soll-Werkstückzahl) In diesem Zähler kann die Anzahl der Werkstücke definiert werden, bei dessen Erreichen die Ist-Werkstückzahl (\$AC_ACTUAL_PARTS) auf "0" zurückgesetzt wird.
\$AC_TOTAL_PARTS	Anzahl der insgesamt gefertigten Werkstücke (Ist-Werkstückzahl Gesamt) Dieser Zähler gibt die Anzahl aller ab Startzeitpunkt gefertigten Werkstücke an. Der Wert wird nur bei einem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.
\$AC_ACTUAL_PARTS	Anzahl der gefertigten Werkstücke (Ist-Werkstückzahl) In diesem Zähler wird die Anzahl aller ab Startzeitpunkt gefertigten Werkstücke registriert. Bei einem Erreichen der Soll-Werkstückzahl (\$AC_REQUIRED_PARTS) wird der Zähler automatisch auf "0" zurückgesetzt (\$AC_REQUIRED_PARTS > 0 vorausgesetzt).
\$AC_SPECIAL_PARTS	Anzahl der vom Anwender gezählten Werkstücke Dieser Zähler erlaubt dem Anwender eine Werkstückzählung nach eigener Definition. Definiert werden kann eine Alarmausgabe bei Erreichen der Soll-Werkstückzahl (\$AC_REQUIRED_PARTS). Eine Nullung des Zählers muss der Anwender selbst vornehmen.

Hinweis

Alle Werkstückzähler werden bei einem Steuerungshochlauf mit Standardwerten auf "0" gesetzt und können unabhängig von ihrer Aktivierung gelesen und geschrieben werden.

Hinweis

Über kanalspezifische Maschinendaten kann auf die Zähler-Aktivierung, den Zeitpunkt der Nullung und den Zählalgorithmus Einfluss genommen werden.

Hinweis

Werkstückzählung mit anwenderdefiniertem M-Befehl

Über Maschinendaten kann eingestellt werden, dass die Zählimpulse für die verschiedenen Werkstückzähler statt über das Programmende M2/M30 über anwenderdefinierte M-Befehle ausgelöst werden.

Literatur

Weitere Informationen zur Funktion "Werkstückzähler" siehe:

- Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1), Kapitel: Werkstückzähler

17.9 Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE)

Das Schreiben von Daten aus einem Teileprogramm heraus auf ein externes Gerät/Datei erfolgt in drei Schritten:

1. Externes Gerät/Datei öffnen
Mit dem Befehl EXTOPEN wird das externe Gerät/Datei für den Kanal zum Schreiben geöffnet.
2. Daten schreiben
Das Ausgabedatum kann mit den Stringfunktionen der NC-Sprache, z. B. SPRINT, aufbereitet werden. Das Schreiben selbst erfolgt über den WRITE-Befehl.
3. Externes Gerät/Datei schließen
Mit dem Befehl EXTCLOSE oder durch Erreichen des Programmendes (M30) sowie bei Kanal-Reset wird das im Kanal belegte externe Gerät/Datei wieder freigegeben.

Syntax

```
DEF INT <Result>  
DEF STRING[<n>] <Output>  
...  
EXTOPEN (<Result>, <ExtDev>, <SyncMode>, <AccessMode>, <WriteMode>)  
...
```

17.9 Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE)

```
<Output>="Ausgabe Daten"
WRITE(<Result>,<ExtDev>,<Output>)
...
EXTCLOSE(<Result>,<ExtDev>)
```

Bedeutung

EXTOPEN:	Vordefinierte Prozedur zum Öffnen eines externen Geräts/Datei		
<Result>:	Parameter 1: Ergebnisvariable Anhand des Ergebnisvariablenwerts kann im Programm das Gelingen der Operation ausgewertet und entsprechend fortgefahren werden.		
	Typ:	INT	
	Werte:	0	kein Fehler
		1	externes Gerät kann nicht geöffnet werden
		2	externes Gerät ist nicht projiziert
		3	externes Gerät mit ungültigem Pfad projiziert
		4	keine Zugriffsrechte für externes Gerät
		5	Benutzungsmodus: externes Gerät bereits "exklusiv" belegt
		6	Benutzungsmodus: externes Gerät bereits "shared" belegt
		7	Dateilänge größer als LOCAL_DRIVE_MAX_FILESIZE
		8	maximale Anzahl externer Geräte überschritten
		9	Option für LOCAL_DRIVE nicht gesetzt
		11	V.24-Schnittstelle bereits durch Easy-Message-Funktion belegt (nur 828D)
12		Schreibmodus: Angabe widersprüchlich zu extdev.ini	
16	ungültiger externer Pfad programmiert		
22	externes Gerät nicht gemountet		

<Ext.Dev>:	Parameter 2: Symbolischer Bezeichner für das zu öffnende externe Gerät/Datei	
	Typ:	STRING
	Der symbolische Bezeichner besteht aus:	
	1. dem logischen Gerätenamen 2. ggf. gefolgt von einem Dateipfad (angehängt mit "/").	
	Folgende logische Gerätenamen sind definiert:	
	"LOCAL_DRIVE":	Lokale CompactFlash Card (vordefiniert)
	"CYC_DRIVE":	reservierte Laufwerksangabe zur Verwendung in SIEMENS-Zyklen (vordefiniert)
	"/dev/ext/1", ... "/dev/ext/9":	Verfügbare Netzlaufwerke Hinweis: Projektierung in der Datei extdev.ini erforderlich!
	"/dev/cyc/1", "/dev/cyc/2":	reservierte Laufwerksangaben zur Verwendung in SIEMENS-Zyklen Hinweis: Projektierung in der Datei extdev.ini erforderlich!
	"/dev/v24":	V.24-Schnittstelle Hinweis: Projektierung in der Datei extdev.ini erforderlich!
Dateipfad:		
<ul style="list-style-type: none"> • Zu "LOCAL_DRIVE" und "CYC_DRIVE" muss ein Dateipfad angegeben werden, z. B.: "LOCAL_DRIVE/my_dir/my_file.txt" • Die logischen Gerätenamen "/dev/ext/1...9" und "/dev/cyc/1...2" können per Projektierung: <ul style="list-style-type: none"> – schon auf eine Datei verweisen, dann darf nur der logische Gerätenamen angegeben werden, z. B.: "/dev/ext/4" – oder auf ein Verzeichnis, dann muss ein Dateipfad angegeben werden, z. B.: "/dev/ext/5/my_dir/my_file.txt" • Zu "/dev/v24" darf kein Dateipfad angehängt sein. 		
Hinweis:		
Für die logischen Gerätenamen "/dev/ext/1...9", "/dev/v24" und "/dev/cyc/1...2" wird Groß-/Kleinschreibung ignoriert, bei der Pfadangabe zu einer Datei ist Groß-/Kleinschreibung signifikant. Für "LOCAL_DRIVE" und "CYC_DRIVE" ist nur Großschreibung zulässig.		

17.9 Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE)

<SyncMode>:	Parameter 3: Bearbeitungsmodus für die WRITE-Befehle zu diesem Gerät/Datei				
	Typ:	STRING			
	Werte:	<table border="1"> <tr> <td>"SYN":</td> <td> <p>Synchrones Schreiben</p> <p>Die Programmausführung wird angehalten, bis der Schreibvorgang abgeschlossen ist.</p> <p>Die erfolgreiche Beendigung des synchronen Schreibens kann durch Auswerten der Fehlervariablen des WRITE-Befehls überprüft werden.</p> </td> </tr> <tr> <td>"ASYN":</td> <td> <p>Asynchrones Schreiben</p> <p>Die Programmausführung wird durch den WRITE-Befehl nicht unterbrochen.</p> <p>Hinweis: Die Ergebnisvariable des WRITE-Befehls ist in diesem Modus nicht aussagekräftig und hat immer den Wert 0 (kein Fehler). Es gibt in diesem Modus keine Sicherheit, dass der WRITE-Befehl erfolgreich war.</p> </td> </tr> </table>	"SYN":	<p>Synchrones Schreiben</p> <p>Die Programmausführung wird angehalten, bis der Schreibvorgang abgeschlossen ist.</p> <p>Die erfolgreiche Beendigung des synchronen Schreibens kann durch Auswerten der Fehlervariablen des WRITE-Befehls überprüft werden.</p>	"ASYN":
"SYN":	<p>Synchrones Schreiben</p> <p>Die Programmausführung wird angehalten, bis der Schreibvorgang abgeschlossen ist.</p> <p>Die erfolgreiche Beendigung des synchronen Schreibens kann durch Auswerten der Fehlervariablen des WRITE-Befehls überprüft werden.</p>				
"ASYN":	<p>Asynchrones Schreiben</p> <p>Die Programmausführung wird durch den WRITE-Befehl nicht unterbrochen.</p> <p>Hinweis: Die Ergebnisvariable des WRITE-Befehls ist in diesem Modus nicht aussagekräftig und hat immer den Wert 0 (kein Fehler). Es gibt in diesem Modus keine Sicherheit, dass der WRITE-Befehl erfolgreich war.</p>				
<AccessMode>:	Parameter 4: Benutzungsmodus für dieses Gerät/Datei				
	Typ:	STRING			
	Werte:	<table border="1"> <tr> <td>"SHARED":</td> <td>Gerät/Datei wird im "Shared"-Modus angefordert. Andere Kanäle können das Gerät mitverwenden, d. h. ebenfalls in diesem Modus öffnen.</td> </tr> <tr> <td>"EXCL":</td> <td>Gerät/Datei wird in dem Kanal exklusiv verwendet, kein anderer Kanal kann das Gerät mitverwenden.</td> </tr> </table>	"SHARED":	Gerät/Datei wird im "Shared"-Modus angefordert. Andere Kanäle können das Gerät mitverwenden, d. h. ebenfalls in diesem Modus öffnen.	"EXCL":
"SHARED":	Gerät/Datei wird im "Shared"-Modus angefordert. Andere Kanäle können das Gerät mitverwenden, d. h. ebenfalls in diesem Modus öffnen.				
"EXCL":	Gerät/Datei wird in dem Kanal exklusiv verwendet, kein anderer Kanal kann das Gerät mitverwenden.				
<WriteMode>:	Parameter 5: Schreibmodus für die WRITE-Befehle zu diesem Gerät/Datei (optional)				
	Typ:	STRING			
	Werte:	"APP":	<p>Anhängen</p> <p>Die Datei bleibt in ihrem Inhalt erhalten, Schreibaufrufe fügen an das Ende an.</p>		
		"OVR":	<p>Überschreiben</p> <p>Der Inhalt der Datei wird gelöscht und durch nachfolgende Schreibaufrufe neu erstellt.</p>		
	Hinweis: Mit diesem Parameter kann der in der Datei extdev.ini projektierte Schreibmodus nicht überschrieben werden. Im Konfliktfall wird der EXTOPEN-Aufruf mit Fehler quittiert.				

WRITE:	Vordefinierte Prozedur zum Schreiben der Ausgabedaten
--------	---

EXTCLOSE:	Vordefinierte Prozedur zum Schließen eines geöffneten externen Geräts/Datei		
<Result>:	Parameter 1: Ergebnisvariable		
	Typ:	INT	
	Werte:	0	kein Fehler
		16	ungültiger externer Pfad programmiert
21		Fehler beim Schließen des externen Geräts	
<ExtDev>:	Parameter 2: Symbolischer Bezeichner für das zu schließende externe Gerät/DateiBeschreibung siehe unter EXTOPEN! Hinweis: Der Bezeichner muss identisch zu dem im EXTOPEN-Aufruf angegebenen Bezeichner sein!		

Beispiel

```

Programmcode
N10      DEF INT RESULT
N20      DEF BOOL EXTDEVICE
N30      DEF STRING[80] OUTPUT
N40      DEF INT PHASE
N50      EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60      IF RESULT > 0
N70          MSG("Fehler bei EXTOPEN:" << RESULT)
N80      ELSE
N90          EXTDEVICE=TRUE
N100     ENDIF
...
N200     PHASE=4
N210     IF EXTDEVICE
N220         OUTPUT=SPRINT("Ende Phase: %D",PHASE)
N230         WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240     ENDIF
...
    
```

Siehe auch

Stringoperationen (Seite 79)

Datei schreiben (WRITE) (Seite 138)

17.10 Alarme (SETAL)

In einem NC-Programm können Alarme gesetzt werden. Diese werden in der Bedienoberfläche in einem besonderen Feld dargestellt. Mit einem Alarm ist jeweils eine Reaktion der Steuerung entsprechend der Alarmkategorie verbunden.

Literatur:

Weiterführende Informationen zu den Alarmreaktionen siehe Inbetriebnahmehandbuch.

Syntax

SETAL(<Alarmnummer>[, <Zeichenkette>])

Bedeutung

SETAL:	Schlüsselwort zur Programmierung eines Alarms. SETAL muss in einem eigenen NC-Satz programmiert werden.	
<Alarmnummer>:	Variable vom Typ INT. Enthält die Alarmnummer. Der gültige Bereich für Alarmnummern liegt zwischen 60000 und 69999, wovon 60000 bis 64999 für SIEMENS-Zyklen reserviert sind und 65000 bis 69999 für den Anwender zur Verfügung stehen.	
<Zeichenkette>:	Bei der Programmierung von Anwenderzyklenalarmen kann zusätzlich eine Zeichenkette mit bis zu 4 Parametern angegeben werden. In diesen Parametern können variable Anwendertexte definiert werden. Es stehen aber auch folgende vordefinierte Parameter zur Verfügung:	
	Parameter	Bedeutung
	%1	Kanalnummer
	%2	Satznummer, Label
	%3	Textindex für Zyklenalarme
	%4	zusätzlicher Alarmparameter

Hinweis

Alarmtexte müssen in der Bedienoberfläche projiziert werden.

Hinweis

Soll eine Alarmausgabe in der auf der Bedienoberfläche aktiven Sprache erfolgen, benötigt der Anwender Informationen über die aktuell auf dem HMI eingestellte Sprache. Diese Information kann im Teileprogramm und in Synchronaktionen über die Systemvariable \$AN_LANGUAGE_ON_HMI abgefragt werden (siehe "Aktuelle Sprache im HMI (Seite 796)").

Beispiel

Programmcode	Kommentar
...	
N100 SETAL(65000)	; Alarm Nr. 65000 setzen
...	

17.11 Erweitertes Stillsetzen und Rückziehen (ESR)

Die Funktion "Erweitertes Stillsetzen und Rückziehen", im weiteren Verlauf mit ESR bezeichnet, bietet die Möglichkeit, in Fehlersituationen prozessabhängig flexibel zu reagieren:

- **Erweitertes Stillsetzen**
Soweit es die spezifische Fehlersituation erlaubt, werden alle für das Erweiterte Stillsetzen freigegebenen Achsen geordnet stillgesetzt.
- **Rückziehen**
Das sich im Eingriff befindliche Werkzeug wird schnellstmöglich vom Werkstück zurückgezogen.
- **Generatorbetrieb (SINAMICS-Antriebsfunktion "Vdc-Regelung")**
Bei Unterschreitung eines parametrierbaren Werts der Zwischenkreisspannung, z. B. bei Ausfall der Netzspannung, wird die für das Rückziehen benötigte elektrische Energie durch Rückspeisen der Bremsenergie eines dafür vorgesehenen Antriebs (Generatorbetrieb) erzeugt.

Auslösequellen

Allgemeine Quellen (NC-extern/global oder BAG-/kanalspezifisch)

- Digitale Eingänge (z. B. auf NCU-Baugruppe) bzw. das steuerungsinterne, rücklesbare Abbild digitaler Ausgänge (\$A_IN, \$A_OUT)
- Kanalzustand (\$AC_STAT)
- VDI-Signale (\$A_DBB)
- Sammelmeldungen einer Anzahl von Alarmen (\$AC_ALARM_STAT)

Axiale Quellen

- Notrückzugsschwelle der Folgeachse (Synchronlauf der elektronischen Kopplung, \$VC_EG_SYNCDIFF[<Folgeachse>])
- Antrieb: Zwischenkreis-Warnschwelle (drohende Unterspannung), \$AA_ESR_STAT[<Achse>]
- Antrieb: Generator-Minimaldrehzahl-Schwelle (keine rückspeisbare Rotationsenergie mehr vorhanden), \$AA_ESR_STAT[<Achse>].

Verknüpfungslogik der statischen Synchronaktionen: Quellen-/Reaktions-Verknüpfung

Die flexiblen Verknüpfungsmöglichkeiten der statischen Synchronaktionen werden genutzt, um aufgrund von Quellen relativ zeitnah bestimmte Reaktionen auszulösen.

Die Verknüpfung aller relevanten Quellen mit Hilfe statischer Synchronaktionen liegt in den Händen des Anwenders. Er kann die Quellen-Systemvariablen als Ganzes oder mit Hilfe von Bit-Masken auch selektiv auswerten und hieran seine gewünschten Reaktionen knüpfen. Die statischen Synchronaktionen sind in allen Betriebsarten wirksam.

Literatur:

Funktionshandbuch Synchronaktionen

Aktivierung

Funktionsfreigabe

Die Funktionen Generatorbetrieb, Stillsetzen, Rückziehen werden durch Setzen des zugehörigen Steuersignals \$AA_ESR_ENABLE freigegeben. Dieses Steuersignal kann von Synchronaktionen verändert werden.

Funktionsauslösung

ESR wird gemeinsam für alle freigegebenen Achsen ausgelöst durch Setzen der Systemvariablen \$AC_ESR_TRIGGER.

Der Generatorbetrieb wird "automatisch" im Antrieb bei Erkennung drohender Zwischenkreis-Unterspannung aktiv.

Antriebsautarkes Stillsetzen und/oder Rückziehen werden aktiv bei Erkennung eines Kommunikationsausfalls (zwischen NC und Antrieb) sowie bei Erkennung einer Zwischenkreis-Unterspannung im Antrieb (Konfiguration und Freigabe vorausgesetzt).

Antriebsautarkes Stillsetzen und/oder Rückziehen können zusätzlich auch von der NC-Seite her ausgelöst werden durch das Setzen des entsprechenden Steuersignals \$AN_ESR_TRIGGER (Broadcast-Kommando an alle Antriebe).

Literatur

Ausführliche Informationen zu ESR siehe:

Funktionshandbuch Sonderfunktionen; Erweitertes Stillsetzen und Rückziehen (R3)

17.11.1 NC-geführtes ESR

17.11.1.1 NC-geführtes Rückziehen (POLF, POLFA, POLFMASK, POLFMLIN)

Für NC-geführtes Rückziehen sind bestimmte Ausgangsbedingungen erforderlich (siehe "NC-geführtes Rückziehen (POLF, POLFA, POLFMASK, POLFMLIN) (Seite 587)"). Sind diese Voraussetzungen erfüllt, dann wird für die im Kanal konfigurierte(n) Rückzugsachse(n) durch Setzen der Systemvariablen \$AC_ESR_TRIGGER (bzw. \$AA_ESR_TRIGGER für Einzelachsen) Schnellabheben (LIFTFAST) aktiviert.

Syntax

```
POLF(<Achse>)=<Position>
POLFA(<Achse>,<Typ>,<Position>)
POLFMASK(<Achse_1>,<Achse_2>,...)
POLFMLIN(<Achse_1>,<Achse_2>,...)
```

Für POLFA sind folgende Kurzformen erlaubt:

```
POLFA(<Achse>,<Typ>) ; Kurzform für Einzelachs-Rückziehen
POLFA(Achse,0/1/2) ; Schnelle Deaktivierung oder Aktivierung
POLFA(Achse,0,$AA_POLFA[Achse]) ; Bewirkt einen Vorlaufstopp
```

POLFA(Achse,0) ; Bewirkt keinen Vorlaufstopp

Bedeutung

POLF:	Adresse zur Angabe der Zielposition der Rückzugsachse POLF ist modal wirksam.	
	<Achse>:	Name der Geometrie- oder Kanal-/Maschinenachse, die zurückzieht
	<Position>:	Rückzugsposition Typ: REAL Für Geometrieachse gilt WKS, sonst MKS. Bei gleichen Bezeichnern für Geometrie- und Kanal-/Maschinenachse wird im WKS zurückgezogen.
POLFA:	Vordefinierter Unterprogrammaufruf zur Angabe der Rückzugsposition von Einzelachsen	
	<Achse>:	Kanalachsbezeichner
	<Typ>:	Positionsangabemodus
		Typ: INT
		Wert:
Hinweis: Ist eine Achse keine Einzelachse oder fehlt der Typ bzw. Typ=0, dann wird ein entsprechender Alarm gemeldet.		
<Position>:	Rückzugsposition (s. o.) Hinweis: Der Positionswert wird auch mit Typ=0 übernommen. Nur ist dieser Wert dann als ungültig markiert und muss für das Rückziehen neu programmiert werden.	
POLFMASK:	Vordefinierter Unterprogrammaufruf zur Auswahl der Achsen, die nach Auslösung des Schnellabhebens unabhängig voneinander zurückgezogen werden sollen.	
	<Achse_1>, ...:	Namen der Achsen, die beim Schnellabheben auf ihre mit POLF definierten Positionen fahren sollen. Alle angegebenen Achsen müssen sich im gleichen Koordinatensystem befinden.
POLFMASK () ohne Angabe einer Achse deaktiviert das Schnellabheben für alle Achsen, die unabhängig voneinander zurückgezogen wurden.		
POLFMLIN:	Vordefinierter Unterprogrammaufruf zur Auswahl der Achsen, die nach Auslösung des Schnellabhebens im linearen Zusammenhang zurückgezogen werden sollen.	
	<Achse_1>, ...:	s. o.
POLFMLIN () ohne Angabe einer Achse deaktiviert das Schnellabheben für alle Achsen, die im linearen Zusammenhang zurückgezogen wurden.		

Hinweis

Bevor über POLFMASK oder POLFMLIN das Schnellabheben auf eine feste Position freigegeben werden kann, muss für die ausgewählten Achsen eine Position mit POLF programmiert worden sein.

Hinweis

Werden Achsen nacheinander mit POLFMASK, POLFMLIN oder POLFMLIN, POLFMASK freigegeben, gilt für die jeweilige Achse immer die letzte Festlegung.

Hinweis

Die mit POLF programmierten Positionen und die Aktivierung durch POLFMASK oder POLFMLIN werden bei Teileprogrammstart gelöscht. Das heißt, der Anwender muss in jedem Teileprogramm die Werte für POLF und die selektierten Achsen in POLFMASK bzw. POLFMLIN neu programmieren.

Hinweis

Wenn bei Verwendung der Kurzformen POLFA nur der Typ geändert wird, dann muss der Anwender sicherstellen, dass entweder die Rückzugsposition oder der Rückzugsweg einen sinnvollen Wert enthält. Insbesondere sind die Rückzugsposition und der Rückzugsweg nach POWER ON neu zu setzen.

Beispiel

Rückziehen einer Einzelachse:

Programmcode	Kommentar
MD37500 \$MA_ESR_REACTION[AX1]=21	; NC-geführtes Rückziehen.
...	
\$AA_ESR_ENABLE[AX1]=1	
POLFA(AX1,1,20.0)	; AX1 wird die axiale Rückzugsposition 20.0 (absolut) zugeteilt.
\$AA_ESR_TRIGGER[AX1]=1	; Ab hier beginnt der Rückzug.

Weitere Informationen

Voraussetzungen für NC-geführtes Rückziehen

- Im Kanal ist eine Rückzugsachse für das NC-geführte Rückziehen projektiert:
MD37500 \$MA_ESR_REACTION = 21
- ESR muss für diese Achse freigegeben sein:
\$AA_ESR_ENABLE = 1
- Verzögerungszeiten sind definiert:
MD21380 \$MC_ESR_DELAY_TIME1
MD21381 \$MC_ESR_DELAY_TIME2

- Im Teileprogramm sind die achsspezifischen Rückzugspositionen mit `POLF` programmiert.
- Die Achsen sind mit `POLFMASK/POLFMLIN` für das NC-geführte Rückziehen ausgewählt.
- Für die Rückzugbewegung müssen die Freigabesignale gesetzt sein und gesetzt bleiben.

NC-geführtes Rückziehen freigeben und starten

Wenn die Systemvariable `$AC_ESR_TRIGGER = 1` gesetzt wird, und wenn in diesem Kanal eine Rückzugsachse konfiguriert ist (d. h. `MD37500 $MA_ESR_REACTION = 21`) und für diese Achse `$AA_ESR_ENABLE = 1` gesetzt ist, dann wird in diesem Kanal Schnellabheben (LIFTFAST) aktiviert.

Die mit `POLF` (bzw. `LFPOS`) konfigurierte Abhebebewegung der mit `POLFMASK` oder `POLFMLIN` selektierte(n) Achse(n) ersetzt die für diese Achse(n) im Teileprogramm festgelegte Bahnbewegung.

Für den Rückzug steht maximal die Summe der Zeiten `MD21380 $MC_ESR_DELAY_TIME1` und `MD21381 $MC_ESR_DELAY_TIME2` zur Verfügung. Nach Ablauf dieser Zeitspanne wird auch für die Rückzugachse Schnellbremsen eingeleitet mit anschließendem Nachführen.

Hinweis

Die Erweiterte Rückzugbewegung (d. h. durch `$AC_ESR_TRIGGER` ausgelöstes `LIFTFAST/LFPOS`) ist **nicht unterbrechbar** und kann nur durch Not-Halt vorzeitig beendet werden.

Hinweis

Durch `$AC_ESR_TRIGGER` ausgelöstes Rückziehen ist gegen Mehrfach-Rückzug verriegelt.

Einzelachs-Rückziehen

Beim Einzelachs-Rückziehen muss mit `POLFA` die Rückzugsposition der Einzelachse programmiert worden sein und die folgenden Bedingungen müssen eingehalten werden:

- `$AA_ESR_ENABLE = 1`
- <Achse> muss zum Triggerzeitpunkt (`$AA_ESR_TRIGGER = 1`) eine Einzelachse sein.
- <Typ> muss entweder 1 oder 2 sein.

Rückzugsrichtung beim Schnellabheben

Zum Zeitpunkt der Aktivierung des Schnellabhebens wird der gültige Frame berücksichtigt.

Hinweis

Frames mit Drehung beeinflussen über `POLF` auch die Richtung, in welche abgehoben wird.

Achstausch

Rückzugsachsen sind immer genau einem NC-Kanal zuzuordnen und dürfen nicht zwischen den Kanälen getauscht werden. Beim Versuch, eine Rückzugsachse in einen anderen Kanal zu tauschen, wird ein Alarm gemeldet. Erst nachdem diese Achse mit `$AA_ESR_ENABLE[AX] = 0` wieder deaktiviert wurde, kann die Achse in einen neuen Kanal getauscht werden. Nach dem erfolgten Achstausch können Achsen wieder mit `$AA_ESR_ENABLE[AX] = 1` beaufschlagt werden.

Neutrale Achsen

Neutrale Achsen können kein NC-geführtes ESR ausführen.

17.11.1.2 NC-geführtes Stillsetzen

Für die im Kanal konfigurierte(n) Stillsetzachse(n) wird durch Setzen der Systemvariablen \$AC_ESR_TRIGGER (bzw. \$AA_ESR_TRIGGER für Einzelachsen) das NC-geführte Stillsetzen aktiviert.

Voraussetzungen

- Im Kanal ist eine Stillsetzachse für das NC-geführte Stillsetzen projektiert:
MD37500 \$MA_ESR_REACTION = 22
- ESR muss für diese Achse freigegeben sein:
\$AA_ESR_ENABLE = 1
- Verzögerungszeiten sind definiert:
MD21380 \$MC_ESR_DELAY_TIME1 (Verzögerungszeit ESR-Achsen)
MD21381 \$MC_ESR_DELAY_TIME2 (ESR-Zeit für interpolatorisches Bremsen)

Ablauf

Für die Dauer der Zeitspanne in MD21380 interpoliert die Achse ungestört weiter wie programmiert. Nach Ablauf der Zeitspanne in MD21380 wird interpolatorisch geführtes Bremsen (Rampenstopp) eingeleitet. Für das interpolatorisch geführte Bremsen steht dann maximal die Zeitspanne in MD21381 zur Verfügung. Nach Ablauf dieser Zeitspanne wird Schnellbremsen mit anschließendem Nachführen eingeleitet.

Beispiel

Stillsetzen einer Einzelachse:

Programmcode	Kommentar
MD37500 \$MC_ESR_REACTION[AX1]=22	; NC-geführtes Stillsetzen.
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	; Ab hier beginnt das Stillsetzen.

17.11.2 Antriebsautarkes ESR

17.11.2.1 Antriebsautarkes Stillsetzen projektieren (ESRS)

Mit der Funktion `ESRS (...)` werden die Antriebsparameter für das "Stillsetzen" der antriebsautarken ESR-Funktion projektiert.

17.11 Erweitertes Stillsetzen und Rückziehen (ESR)

Syntax

ESRS (<Achse_1>, <Stillsetzzeit_1>[, ..., <Achse_n>, <Stillsetzzeit_n>])

Bedeutung

ESRS (...):	Funktion zum Schreiben der Antriebsparameter für die ESR-Funktion "Stillsetzen" Die Funktion: <ul style="list-style-type: none"> • muss alleine im Satz stehen. • löst Vorlaufstopp aus. • kann nicht in Synchronaktionen verwendet werden. 	
<Achse_1>, ..., <Achse_n>:	Achse, für die antriebsautarkes Stillsetzen projiziert werden soll Im Antrieb wird für diese Achse der Antriebsparameter p0888 (Konfiguration) geschrieben: p0888 = 1	
	Typ:	AXIS
	Wertebereich:	Kanalachsbezeichner
<Stillsetzzeit_1>, ..., <Stillsetzzeit_n>:	Zeitdauer, über die der Antrieb nach dem Auftreten eines Fehlers mit dem aktuellen Drehzahlsollwert konstant weiter fährt Im Antrieb wird für die angegebene Achse der Antriebsparameter p0892 (Zeitstufe) geschrieben: p0892 = <Stillsetzzeit>	
	Einheit:	s
	Typ:	REAL
	Wertebereich:	0.00 - 20.00
In einem Funktionsaufruf können maximal 5 Achsen programmiert werden; n = 5		

17.11.2.2 Antriebsautarkes Rückziehen projizieren (ESRR)

Mit der Funktion ESRR (...) werden die Antriebsparameter für das "Rückziehen" der antriebsautarken ESR-Funktion projiziert.

Syntax

ESRR (<Achse_1>, <Rückzugsweg_1>, <Rückzugsgeschwindigkeit_1>[, ..., <Achse_n>, <Rückzugsweg_n>, <Rückzugsgeschwindigkeit_n>])

Bedeutung

ESRR (...):	Funktion zum Schreiben der Antriebsparameter für die ESR-Funktion "Rückziehen" Die Funktion: <ul style="list-style-type: none"> • muss alleine im Satz stehen. • löst Vorlaufstopp aus. • kann nicht in Synchronaktionen verwendet werden. 	
<Achse_1>, ..., <Achse_n>:	Achse, für die antriebsautarkes Rückziehen projiziert werden soll Im Antrieb wird für diese Achse der Antriebsparameter p0888 (Konfiguration) geschrieben: p0888 = 2	
	Typ:	AXIS
	Wertebereich:	Kanalachsbezeichner
<Rückzugsweg_1>, ..., <Rückzugsweg_n>:	Der Rückzugsweg wird für den Antrieb in eine Rückzugsdrehzahl umgerechnet. Der Wert wird für die angegebene Achse in den Antriebsparameter p0893 (Drehzahl) geschrieben: p0893 = (<Rückzugsweg_n> umgerechnet in Rückzugsdrehzahl)	
	Einheit:	mm/min, inch/min, Grad/min (abhängig von der Einheit der Achse)
	Typ:	REAL
	Wertebereich:	MIN - MAX
<Rückzugsgeschwindigkeit_1>, ..., <Rückzugsgeschwindigkeit_n>:	Die Rückzugsgeschwindigkeit wird für den Antrieb in eine Zeitdauer umgerechnet. Der Wert wird für die angegebene Achse in den Antriebsparameter p0892 (Zeitstufe) [s] geschrieben: p0892 = <Rückzugsweg_n> / <Rückzugsgeschwindigkeit_n>	
	Einheit:	mm/min, inch/min, Grad/min (abhängig von der Einheit der Achse)
	Typ:	REAL
	Wertebereich:	0.00 - MAX
In einem Funktionsaufruf können maximal 5 Achsen programmiert werden; n = 5		

Eigene Abspanprogramme

18.1 Unterstützende Funktionen für das Abspannen

Für das Abspannen werden Ihnen fertige Bearbeitungszyklen angeboten. Darüber hinaus haben Sie die Möglichkeit, mit den nachfolgend aufgeführten Funktionen eigene Abspanprogramme zu erstellen:

- Konturtabelle erstellen (CONTPRON)
- Codierte Konturtabelle erstellen (CONTDCON)
- Konturaufbereitung ausschalten (EXECUTE)
- Schnittpunkt zwischen zwei Konturelementen ermitteln (INTERSEC)
(Nur für Tabellen, die durch CONTPRON erstellt wurden.)
- Konturelemente einer Tabelle satzweise abarbeiten (EXECTAB)
(Nur für Tabellen, die durch CONTPRON erstellt wurden.)
- Kreisdaten berechnen (CALCDAT)

Hinweis

Sie können diese Funktionen nicht nur zum Abspannen, sondern universell einsetzen.

Voraussetzungen

Vor dem Aufruf der Funktionen CONTPRON oder CONTDCON müssen:

- ein Startpunkt angefahren werden, der eine kollisionsfreie Bearbeitung erlaubt.
- die Schneidenradiuskorrektur mit G40 ausgeschaltet sein.

18.2 Konturtabelle erstellen (CONTPRON)

Mit CONTPRON schalten Sie die Konturaufbereitung ein. Die nachfolgend aufgerufenen NC-Sätze werden nicht abgearbeitet, sondern in einzelne Bewegungen aufgeteilt und in der Konturtabelle abgelegt. Jedem Konturelement entspricht eine Tabellenzeile im zweidimensionalen Feld der Konturtabelle. Die Anzahl der ermittelten Hinterschnitte wird zurückgeliefert.

Syntax

Konturaufbereitung einschalten:

```
CONTPRON (<Konturtabelle>, <Bearbeitungsart>, <Hinterschnitte>,
<Bearbeitungsrichtung>)
```

Konturaufbereitung ausschalten und in den normalen Abarbeitungsmodus zurückkehren:

```
EXECUTE (<FEHLER>)
```

Siehe " Konturaufbereitung ausschalten (EXECUTE) (Seite 609) "

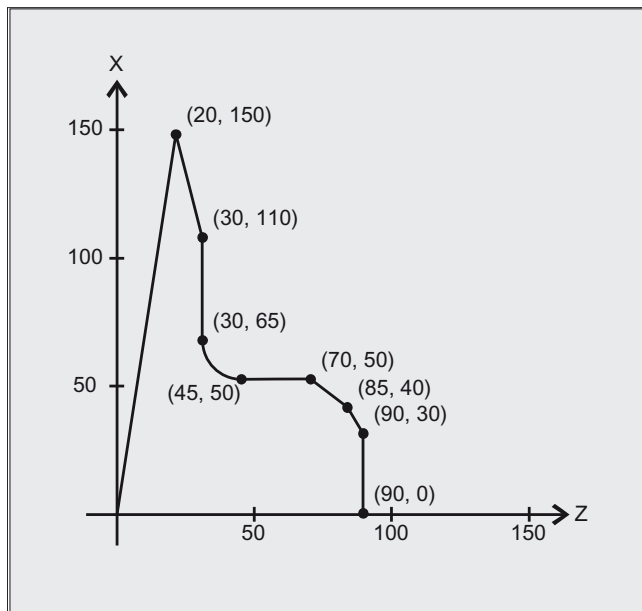
Bedeutung

CONTPRON:	Vordefinierte Prozedur zum Einschalten der Konturaufbereitung zur Erstellung einer Konturtable			
<Konturtable>:	Name der Konturtable			
<Bearbeitungsart>:	Parameter für die Bearbeitungsart			
	Typ:	CHAR		
	Wert:	"G":	Längsdrehen: Innenbearbeitung	
		"L":	Längsdrehen: Außenbearbeitung	
		"N":	Plandrehen: Innenbearbeitung	
"P":		Plandrehen: Außenbearbeitung		
<Hinterschnitte>:	Ergebnisvariable für die Anzahl auftretender Hinterschnittelemente			
	Typ:	INT		
<Bearbeitungsrichtung>:	Parameter für die Bearbeitungsrichtung			
	Typ:	INT		
	Wert:	0	Konturaufbereitung vorwärts (Standardwert)	
		1	Konturaufbereitung in beiden Richtungen	

Beispiel 1

Erstellen einer Konturtable mit:

- Namen "KTAB"
- max. 30 Konturelementen (Kreise, Geraden)
- einer Variablen für die Anzahl auftretender Hinterschnittelemente
- einer Variablen für Fehlermeldungen

**NC-Programm:**

Programmcode	Kommentar
N10 DEF REAL KTAB[30,11]	; Konturtabelle mit Namen KTAB und max. 30 Konturelementen, Parameterwert 11 (Spaltenzahl der Tabelle) ist eine feste Größe.
N20 DEF INT ANZHINT	; Variable für die Anzahl der Hinter-schnittelemente mit Namen ANZHINT.
N30 DEF INT FEHLER	; Variable für die Fehlerrückmeldung (0=kein Fehler, 1=Fehler).
N40 G18	
N50 CONTPRON(KTAB,"G",ANZHINT)	; Konturaufbereitung einschalten.
N60 G1 X150 Z20	; N60 bis N120: Konturbeschreibung
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(FEHLER)	; Füllen der Konturtabelle beenden, Umschalten auf normalen Programmbetrieb.
N140 ...	; Weitere Bearbeitung der Tabelle.

Konturtabelle KTAB:

Index Zeile	Spalte									
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(0)	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

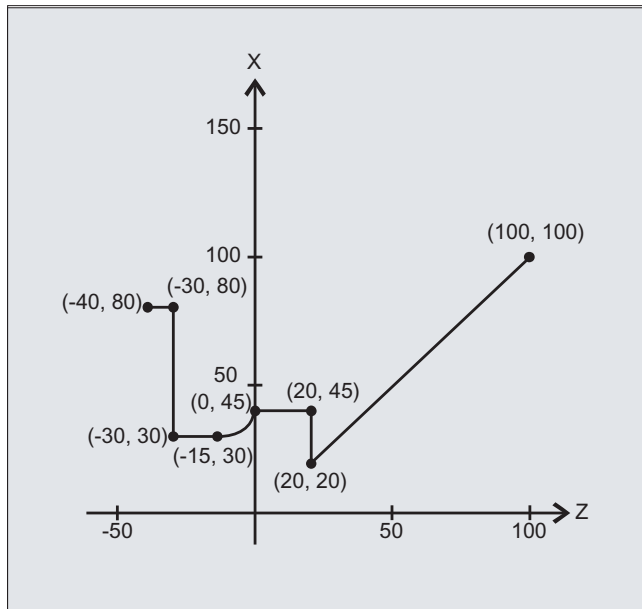
Erläuterung der Spalteninhalte:

- (0) Zeiger auf nächstes Konturelement (auf die Zeilennummer desselben)
- (1) Zeiger auf vorhergehendes Konturelement
- (2) Codierung des Konturmodus für die Bewegung
Mögliche Werte für X = abc
a = 10² G90 = 0 G91 = 1
b = 10¹ G70 = 0 G71 = 1
c = 10⁰ G0 = 0 G1 = 1 G2 = 2 G3 = 3
- (3), (4) Anfangspunkt der Konturelemente
(3) = Abszisse, (4) = Ordinate in der aktuellen Ebene
- (5), (6) Endpunkt der Konturelemente
(5) = Abszisse, (6) = Ordinate in der aktuellen Ebene
- (7) Max-/min-Anzeiger: kennzeichnet lokale Maxima und Minima in der Kontur
- (8) Maximaler Wert zwischen Konturelement und Abszisse (bei Längsbearbeitung) bzw. Ordinate (bei Planbearbeitung). Der Winkel ist abhängig von der programmierten Bearbeitungsart.
- (9), (10) Mittelpunktskoordinaten des Konturelements, wenn es ein Kreissatz ist.
(9) = Abszisse, (10) = Ordinate

Beispiel 2

Erstellen einer Konturtabelle mit

- Namen KTAB
- max. 92 Konturelementen (Kreise, Geraden)
- Betriebsart: Längsdrehen, Außenbearbeitung
- Aufbereitung vorwärts und rückwärts

**NC-Programm:**

Programmcode	Kommentar
N10 DEF REAL KTAB[92,11]	; Konturtabelle mit Namen KTAB und max. 92 Konturelementen, Parameterwert 11 ist eine feste Größe.
N20 DEF CHAR BT="L"	; Betriebsart für CONTPRON: Längsdrehen, Außenbearbeitung
N30 DEF INT HE=0	; Anzahl der Hinterschnittelemente=0
N40 DEF INT MODE=1	; Aufbereitung vorwärts und rückwärts
N50 DEF INT ERR=0	; Fehlerrückmeldung
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; Konturaufbereitung einschalten.
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)	; Füllen der Konturtabelle beenden, Umschalten auf normalen Programmbetrieb.
...	

Konturtabelle KTAB:

Nach Ende der Konturaufbereitung steht die Kontur in beiden Richtungen zur Verfügung.

Index	Spalte										
Zeile	(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

Erläuterung der Spalteninhalte und der Anmerkungen zu den Zeilen 0, 1, 6, 8, 83, 85 und 91

Es gelten die im Beispiel 1 genannten Erläuterungen der Spalteninhalte.

Immer in Tabellen-Zeile 0:

- 1) Vorgänger: Zeile n enthält das Konturende vorwärts
- 2) Nachfolger: Zeile n ist das Konturtabellenende vorwärts

Je einmal innerhalb der Konturelemente vorwärts:

- 3) Vorgänger: Konturbeginn (vorwärts)
- 4) Nachfolger: Konturende (vorwärts)

Immer auf Zeile Konturtabellenende (vorwärts) +1:

- 5) Vorgänger: Anzahl der Hinterschnitte vorwärts
- 6) Nachfolger: Anzahl der Hinterschnitte rückwärts

Je einmal innerhalb der Konturelemente rückwärts:

- 7) Nachfolger: Konturende (rückwärts)
- 8) Vorgänger: Konturbeginn (rückwärts)

Immer in letzter Tabellen-Zeile:

- 9) Vorgänger: Zeile n ist der Konturtabellenanfang (rückwärts)

10) Nachfolger: Zeile n enthält den Konturanfang (rückwärts)

Weitere Informationen

Erlaubte Verfahrensbefehle, Koordinatensystem

Für die Konturprogrammierung sind folgende G-Befehle zulässig:

- G-Gruppe 1: G0, G1, G2, G3

Zusätzlich möglich sind:

- Rundung und Fase
- Kreisprogrammierung über CIP und CT

Die Funktionen Spline, Polynom und Gewinde führen zu Fehlern.

Änderungen des Koordinatensystems durch Einschalten eines Frames sind zwischen `CONTPRON` und `EXECUTE` nicht zulässig. Gleiches gilt für einen Wechsel zwischen G70 und G71 bzw. G700 und G710.

Ein Tausch der Geometrieachsen mit `GEOAX` während der Aufbereitung der Konturtabelle führt zu einem Alarm.

Hinterschnittlelemente

Die Konturbeschreibung der einzelnen Hinterschnittlelemente kann wahlweise in einem Unterprogramm oder in einzelnen Sätzen erfolgen.

Abspannen unabhängig von der programmierten Konturrichtung

Die Konturaufbereitung mit `CONTPRON` wurde so erweitert, dass nach ihrem Aufruf die Konturtabelle unabhängig von der programmierten Richtung zur Verfügung steht.

18.3 Codierte Konturtabelle erstellen (CONTDCON)

Bei der mit `CONTDCON` eingeschalteten Konturaufbereitung werden die nachfolgend aufgerufenen NC-Sätze in einer 6-spaltigen Konturtabelle speichergünstig codiert abgelegt. Jedem Konturelement entspricht eine Tabellenzeile in der Konturtabelle. Aus Kenntnis der unten angegebenen Codierungsregeln können Sie z. B. für Zyklen aus den Tabellenzeilen DIN-Code-Programme zusammenstellen. In der Tabellenzeile mit der Nummer 0 werden die Daten des Ausgangspunkts gespeichert.

Syntax

Konturaufbereitung einschalten:

```
CONTDCON (<Konturtabelle>, <Bearbeitungsrichtung>)
```

Konturaufbereitung ausschalten und in den normalen Abarbeitungsmodus zurückkehren:

```
EXECUTE (<FEHLER>)
```

Siehe " Konturaufbereitung ausschalten (EXECUTE) (Seite 609) "

Bedeutung

CONTDCON:	Vordefinierte Prozedur zum Einschalten der Konturaufbereitung zur Erstellung einer codierten Konturtable		
<Konturtable>:	Name der Konturtable		
<Bearbeitungsrichtung>:	Parameter für Bearbeitungsrichtung		
	Typ:	INT	
	Wert:	0	Konturaufbereitung gemäß der Folge der Kontursätze (Standardwert)
		1	unzulässig

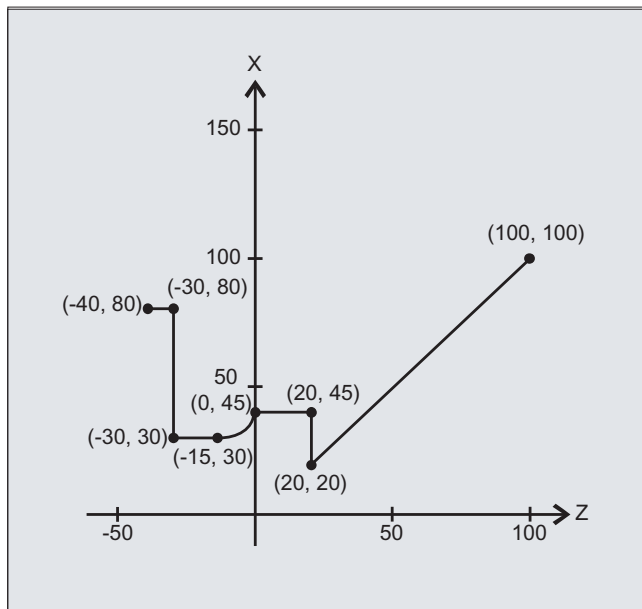
Hinweis

Die für CONTDCON zugelassenen G-Codes im zu tabellierenden Programmstück sind umfangreicher als bei CONTPRON. Darüber hinaus werden Vorschübe und Vorschubtyp pro Konturstück mitgespeichert.

Beispiel

Erstellen einer Konturtable mit:

- Namen "KTAB"
- Konturelementen (Kreise, Geraden)
- Betriebsart: Drehen
- Bearbeitungsrichtung: vorwärts



NC-Programm:

Programmcode	Kommentar
N10 DEF REAL KTAB[9,6]	; Konturtabelle mit Namen KTAB und 9 Tabellenzeilen. Diese erlauben 8 Kontursätze. Der Parameterwert 6 (Spaltenzahl der Tabelle) ist eine feste Größe.
N20 DEF INT MODE = 0	; Variable für die Bearbeitungsrichtung. Standardwert 0: nur in programmierter Richtung der Kontur.
N30 DEF INT ERROR = 0	; Variable für die Fehlerrückmeldung.
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; Aufruf Konturaufbereitung (MODE darf weggelassen werden).
N110 G1 Z20 X20 F200	; Konturbeschreibung.
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)	; Füllen der Konturtabelle beenden, Umschalten auf normalen Programmbetrieb.
...	

Konturtabelle KTAB:

	Spaltenindex					
	0	1	2	3	4	5
Zeilenindex	Konturmodus	Endpunkt Abszisse	Endpunkt Ordinate	Mittelpunkt Abszisse	Mittelpunkt Ordinate	Vorschub
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

Erläuterung der Spalteninhalte:

Zeile 0: Codierungen für den **Startpunkt**:

Spalte 0:	10^0 (Einerstelle): G0 = 0 10^1 (Zehnerstelle): G70 = 0, G71 = 1, G700 = 2, G710 = 3
Spalte 1:	Startpunkt Abszisse
Spalte 2:	Startpunkt Ordinate
Spalte 3-4:	0
Spalte 5:	Zeilenindex des letzten Konturstückes in der Tabelle

Zeilen 1-n: Einträge der **Konturstücke**

Spalte 0:	10^0 (Einerstelle): G0 = 0, G1 = 1, G2 = 2, G3 = 3 10^1 (Zehnerstelle): G70 = 0, G71 = 1, G700 = 2, G710 = 3 10^2 (Hunderterstelle): G90 = 0, G91 = 1 10^3 (Tausenderstelle): G93 = 0, G94 = 1, G95 = 2, G96 = 3 10^4 (Zehntausenderstelle): G60 = 0, G44 = 1, G641 = 2, G642 = 3 10^5 (Hunderttausender Stelle): G9 = 1
Spalte 1:	Endpunkt Abszisse
Spalte 2:	Endpunkt Ordinate
Spalte 3:	Mittelpunkt Abszisse bei Kreisinterpolation
Spalte 4:	Mittelpunkt Ordinate bei Kreisinterpolation
Spalte 5:	Vorschub

Weitere Informationen

Erlaubte Verfahrbefehle, Koordinatensystem

Für die Konturprogrammierung sind folgende G-Gruppen und G-Befehle zulässig:

G-Gruppe 1:	G0, G1, G2, G3
G-Gruppe 10:	G60, G64, G641, G642
G-Gruppe 11:	G9
G-Gruppe 13:	G70, G71, G700, G710
G-Gruppe 14:	G90, G91
G-Gruppe 15:	G93, G94, G95, G96, G961

Zusätzlich möglich sind:

- Rundung und Fase
- Kreisprogrammierung über CIP und CT

Die Funktionen Spline, Polynom und Gewinde führen zu Fehlern.

Änderungen des Koordinatensystems durch Einschalten eines Frames sind zwischen CONTDCON und EXECUTE nicht zulässig. Gleiches gilt für einen Wechsel zwischen G70 und G71 bzw. G700 und G710.

Ein Tausch der Geometrieachsen mit GEOAX während der Aufbereitung der Konturtabelle führt zu einem Alarm.

Bearbeitungsrichtung

Die mit CONTDCON erzeugte Konturtabelle ist zum Abspannen in der programmierten Richtung der Kontur vorgesehen.

18.4 Schnittpunkt zwischen zwei Konturelementen ermitteln (INTERSEC)

INTERSEC ermittelt den Schnittpunkt von zwei normierten Konturelementen aus mit CONTPRON erzeugten Konturtabellen.

Syntax

```
<Status>=INTERSEC (<Konturtabelle_1>[<Konturelement_1>],
<Konturtabelle_2>[<Konturelement_2>], <Schnittpunkt>, <Bearbeitungsart
>)
```

Bedeutung

INTERSEC:	Vordefinierte Funktion zur Ermittlung des Schnittpunkts zweier Konturelemente aus mit CONTPRON erzeugten Konturtabellen		
<Status>:	Variable für den Schnittpunktstatus		
	Typ:	BOOL	
	Wert:	TRUE	Schnittpunkt gefunden
		FALSE	kein Schnittpunkt gefunden
<Konturtabelle_1>:	Name der ersten Konturtabelle		
<Konturelement_1>:	Nummer des Konturelements der ersten Konturtabelle		
<Konturtabelle_2>:	Name der zweiten Konturtabelle		
<Konturelement_2>:	Nummer des Konturelements der zweiten Konturtabelle		
<Schnittpunkt>:	Schnittpunkt-Koordinaten in der aktiven Ebene (G17 / G18 / G19)		
	Typ:	REAL	
<Bearbeitungsart>:	Parameter für die Bearbeitungsart		
	Typ:	INT	
	Wert:	0	Schnittpunktberechnung in der mit Parameter 2 aktiven Ebene (Standardwert)
		1	Schnittpunktberechnung unabhängig der übergebenen Ebene

Hinweis

Beachten Sie, dass die Variablen vor ihrer Verwendung definiert sein müssen.

Die Übergabe der Konturen erfordert die Einhaltung der mit `CONTPRON` definierten Werte:

Parameter	Bedeutung
2	Codierung des Kontur-Mode für die Bewegung
3	Kontur-Anfangpunkt Abszisse
4	Kontur-Anfangpunkt Ordinate
5	Kontur-Endpunkt Abszisse
6	Kontur-Endpunkt Ordinate
9	Mittelpunktskoordinate für die Abszisse (nur bei Kreis-Kontur)
10	Mittelpunktskoordinate für die Ordinate (nur bei Kreis-Kontur)

Beispiel

Schnittpunkt von Konturelement 3 der Tabelle `TABNAME1` und Konturelement 7 der Tabelle `TABNAME2` ermitteln. Die Schnittpunkt-Koordinaten in der aktiven Ebene werden in der Variablen `ISCOORD` (1. Element = Abszisse, 2. Element = Ordinate) abgelegt. Existiert kein Schnittpunkt, erfolgt ein Sprung zu `KEINSCH` (kein Schnittpunkt gefunden).

Programmcode	Kommentar
<code>DEF REAL TABNAME1[12,11]</code>	<code>; Konturtabelle 1</code>
<code>DEF REAL TABNAME2[10,11]</code>	<code>; Konturtabelle 2</code>
<code>DEF REAL ISCOORD[2]</code>	<code>; Variable für Schnittpunkt-Koordinaten.</code>
<code>DEF BOOL ISPOINT</code>	<code>; Variable für Schnittpunktstatus.</code>
<code>DEF INT MODE</code>	<code>; Variable für Bearbeitungsart.</code>
<code>...</code>	
<code>MODE=1</code>	<code>; Berechnung unabhängig von der aktiven Ebene.</code>
<code>N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)</code>	<code>; Aufruf Schnittpunkt der Konturelemente.</code>
<code>N20 IF ISPOINT==FALSE GOTOF KEINSCH</code>	<code>; Sprung zu KEINSCH.</code>
<code>...</code>	

18.5 Konturelemente einer Tabelle satzweise abfahren (EXECTAB)

Mit `EXECTAB` können Sie Konturelemente einer Tabelle, die z. B. mit dem `CONTPRON` erzeugt wurde, satzweise abfahren.

Syntax

```
EXECTAB (<Konturtabelle>[<Konturelement>])
```

Bedeutung

EXECTAB:	Vordefinierte Proedur zum Abfahren eines Konturelements
<Konturtabelle>:	Name der Konturtabelle
<Konturelement>:	Nummer des Konturelements

Beispiel

Die Konturelemente 0 bis 2 der Tabelle KTAB sollen satzweise abgefahren werden.

Programmcode	Kommentar
N10 EXECTAB(KTAB[0])	; Element 0 der Tabelle KTAB verfahren.
N20 EXECTAB(KTAB[1])	; Element 1 der Tabelle KTAB verfahren.
N30 EXECTAB(KTAB[2])	; Element 2 der Tabelle KTAB verfahren.

18.6 Kreisdaten berechnen (CALCDAT)

Mit `CALCDAT` können Sie aus drei oder vier bekannten Kreispunkten den Radius und die Kreismittelpunkt-Koordinaten berechnen. Die angegebenen Punkte müssen unterschiedlich sein.

Bei 4 Punkten, die nicht exakt auf dem Kreis liegen, wird für Kreismittelpunkt und Radius ein Mittelwert gebildet.

Hinweis**Rechenvorschrift zur Mittelwertbildung**

Die Kreisbogenberechnung wird 4 x ausgeführt:

1. mit Kreispunkt 1, 2, 3
2. mit Kreispunkt 1, 2, 4
3. mit Kreispunkt 1, 3, 4
4. mit Kreispunkt 2, 3, 4

Die Kreismittelpunkt-Koordinaten Abszissenwert und Ordinatenwert werden berechnet, indem die Abszissenwerte bzw. Ordinatenwerte der vier Kreisbogenberechnungen addiert und durch 4 geteilt werden.

Der Radius wird berechnet, indem die Wurzel aus der Summe der vier Radien der Kreisbogenberechnungen gebildet und das Ergebnis mit 0,5 multipliziert wird.

Syntax

```
<Status>=CALCDAT (<Kreispunkte> [<Anzahl>,<Art>] ,<Anzahl>,<Ergebnis>)
```

Bedeutung

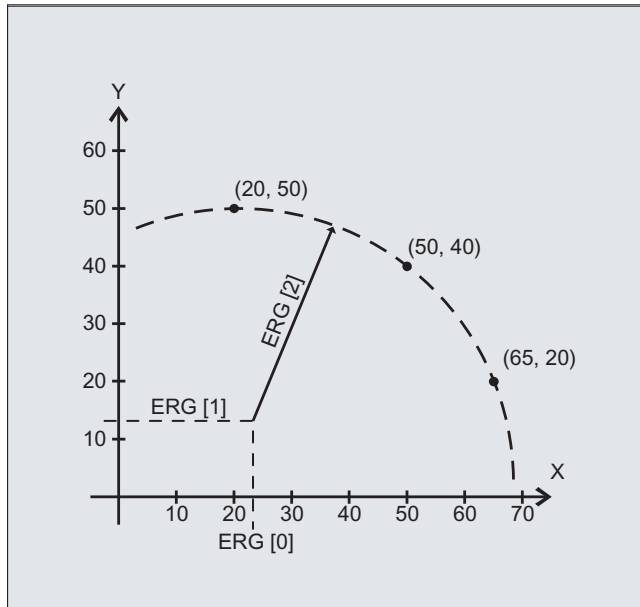
CALCDAT:	Vordefinierte Funktion zur Berechnung von Radius und Mittelpunkt-Koordinaten eines Kreises aus 3 oder 4 Punkten		
<Status>:	Variable für den Kreisberechnungsstatus		
	Typ:	BOOL	
	Wert:	TRUE	Die angegebenen Punkte liegen auf einem Kreis.
FALSE		Die angegebenen Punkte liegen nicht auf einem Kreis.	
<Kreispunkte>[]:	Variable zur Angabe der Kreispunkte mit den Parametern:		
	<Anzahl>:	Anzahl der Kreispunkte (3 oder 4)	
	<Art>:	Art der Koordinatenangabe, z. B. 2 für 2-Punkt-Koordinaten	
<Anzahl>:	Parameter für die Anzahl der zur Berechnung verwendeten Punkte (3 oder 4)		
<Ergebnis>[3]:	Variable für Ergebnis: Angabe von Kreismittelpunkt-Koordinaten und Radius		
	0	Kreismittelpunkt-Koordinate: Abszissenwert	
	1	Kreismittelpunkt-Koordinate: Ordinatenwert	
	2	Radius	

Hinweis

Beachten Sie, dass die Variablen vor ihrer Verwendung definiert sein müssen.

Beispiel

Von drei Punkten soll ermittelt werden, ob sie auf einem Kreisabschnitt liegen.



Programmcode	Kommentar
N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)	; Variable zur Angabe der Kreispunkte.
N20 DEF REAL ERG[3]	; Variable für Ergebnis.
N30 DEF BOOL STATUS	; Variable für Status.
N40 STATUS=CALCDAT(PKT,3,ERG)	; Aufruf der ermittelten Kreisdaten.
N50 IF STATUS == FALSE GOTOF ERROR	; Sprung zu Fehler.

18.7 Konturaufbereitung ausschalten (EXECUTE)

Mit EXECUTE wird die Konturaufbereitung abgeschaltet und gleichzeitig in den normalen Abarbeitungsmodus zurückgeschaltet.

Syntax

```
EXECUTE (<FEHLER>)
```

Bedeutung

EXECUTE:	Vordefinierte Prozedur zum Beenden der Konturaufbereitung	
<FEHLER>:	Variable für Fehlerrückmeldung	
	Typ:	INT
	Der Wert der Variablen zeigt an, ob die Kontur fehlerfrei aufbereitet werden konnte:	
	0	Fehler
1	kein Fehler	

Beispiel

Programmcode

```
...  
N30 CONTPRON(...)  
N40 G1 X... Z...  
...  
N100 EXECUTE(...)  
...
```

Zyklen extern programmieren

19.1 Technologische Zyklen

19.1.1 Einleitung

Inhalt

Dieses Kapitel enthält die Beschreibung der Zyklen für die Technologien Drehen, Fräsen, Schleifen.

Aufbau

Die Beschreibung eines Zyklus ist wie folgt aufgebaut:

- **Syntax**
Zyklusname und Aufrufreihenfolge der Übergabeparameter
- **Parameter**
Tabelle zur Erläuterung der einzelnen Parameter

Parameterbeschreibung

In der Tabelle sind zu einem Parameter folgende Daten angegeben: Name, Beschreibung, Wertebereich und Abhängigkeiten zu anderen Parametern.

Die Spalte zum Verweis auf den Parameter in der Maske dient der Erleichterung zum Wiederauffinden von an der Steuerung programmierte Werten beim Rückübersetzen von extern generierten Zyklenaufrufen.

Parameter "nur für die Oberfläche"

In der Tabelle sind Parameter mit "nur für die Oberfläche" gekennzeichnet. Diese sind für die Funktion des Zyklus nicht von Bedeutung. Sie werden nur benötigt, um Zyklenaufrufe vollständig rückübersetzen zu können. Sind sie nicht programmiert, kann der Zyklus trotzdem rückübersetzt werden, die Felder sind dann entsprechend farblich gekennzeichnet und müssen in der Maske ausgefüllt werden.

Parameter "reserviert"

Parameter, die mit "reserviert" beschrieben sind, müssen mit Wert 0 oder Leerkomma programmiert werden, damit die Zuordnung der folgenden Aufrufparameter zu den internen Zyklusparametern stimmt. Ausnahme: bei Stringparametern Wert "" oder Leerkomma.

Zyklen auf Positionsmuster wiederholen

Bohr- und Fräszyklen können auf Positionsmuster wiederholt werden (modale Aufrufe). Vor dem Zyklus ist dann in derselben Zeile `MCALL` zu schreiben, z. B. `MCALL CYCLE83 (. . .)`.

Hinweis

Sind bestimmte Übergabeparameter (z. B. `<_VARI>`, `<_GMODE>`, `<_DMODE>`, `<_AMODE>`) indirekt als Parameter programmiert, wird die Eingabemaske beim Rückübersetzen geöffnet, kann aber nicht abgespeichert werden, weil es zu bestimmtem Auswahlfeldern keine eindeutige Zuordnung gibt.

19.1.2 Technologie-spezifische Übersicht

In der folgenden Übersichtstabelle sind alle verfügbaren extern programmierbaren technologischen Zyklen aufgelistet und der jeweiligen Technologie zugeordnet:

Technologie	Technologischer Zyklus
Bohren	<ul style="list-style-type: none"> • CYCLE81 - Bohren, Zentrieren (Seite 651) • CYCLE82 - Bohren, Plansenken (Seite 652) • CYCLE85 - Reiben (Seite 661) • CYCLE86 - Ausdrehen (Seite 662) • CYCLE83 - Tieflochbohren (Seite 655) • CYCLE830 - Tieflochbohren 2 (Seite 686) • CYCLE84 - Gewindebohren ohne Ausgleichfutter (Seite 658) • CYCLE840 - Gewindebohren mit Ausgleichfutter (Seite 695) • CYCLE78 - Bohrgewinde fräsen (Seite 647) • CYCLE802 - Beliebige Positionen (Seite 683) • HOLES1 - Lochreihe (Seite 614) • CYCLE801 - Gitter oder Rahmen (Seite 681) • HOLES2 - Lochkreis (Seite 614)
Drehen	<ul style="list-style-type: none"> • CYCLE951 - Abspannen (Seite 706) • CYCLE930 - Einstich (Seite 701) • CYCLE940 - Freistich Formen (Seite 703) • CYCLE99 - Gewindedrehen (Seite 671) • CYCLE98 - Gewindekette (Seite 667) • CYCLE92 - Abstich (Seite 663)
Konturdrehen	<ul style="list-style-type: none"> • CYCLE62 - Konturaufruf (Seite 633) • CYCLE952 - Konturstechen (Seite 709)

Technologie	Technologischer Zyklus
Fräsen	<ul style="list-style-type: none"> • CYCLE61 - Planfräsen (Seite 631) • POCKET3 - Rechtecktasche fräsen (Seite 616) • POCKET4 - Kreistasche fräsen (Seite 619) • CYCLE76 - Rechteckzapfen fräsen (Seite 643) • CYCLE77 - Kreiszapfen fräsen (Seite 645) • CYCLE79 - Mehrkant (Seite 649) • SLOT1 - Längsnut (Seite 621) • SLOT2 - Kreisnut (Seite 624) • CYCLE899 - Offene Nut fräsen (Seite 698) • LONGHOLE - Langloch (Seite 626) • CYCLE70 - Gewindefräsen (Seite 637) • CYCLE60 - Gravurzyklus (Seite 628)
Konturfräsen	<ul style="list-style-type: none"> • CYCLE62 - Konturaufruf (Seite 633) • CYCLE72 - Bahnfräsen (Seite 639) • CYCLE63 - Konturtasche fräsen (Seite 634) • CYCLE64 - Konturtasche vorbohren (Seite 636)
Schleifen	<ul style="list-style-type: none"> • CYCLE495 - Profilieren (Seite 676) • CYCLE435 - Abrichterposition berechnen (Seite 676) • CYCLE4071 - Längsschleifen mit Zustellung am Umkehrpunkt (Seite 715) • CYCLE4072 - Längsschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal (Seite 716) • CYCLE4073 - Längsschleifen mit kontinuierlicher Zustellung (Seite 720) • CYCLE4074 - Längsschleifen mit kontinuierlicher Zustellung und Abbruchsignal (Seite 721) • CYCLE4075 - Flachsleifen mit Zustellung am Umkehrpunkt (Seite 724) • CYCLE4077 - Flachsleifen mit Zustellung am Umkehrpunkt und Abbruchsignal (Seite 727) • CYCLE4078 - Flachsleifen mit kontinuierlicher Zustellung (Seite 730) • CYCLE4079 - Flachsleifen mit intermittierender Zustellung (Seite 732)
Sonstige	<ul style="list-style-type: none"> • CYCLE800 - Schwenken (Seite 678) • CYCLE832 - High Speed Settings (Seite 692)

19.1.3 HOLES1 - Lochreihe

Syntax

HOLES1 (<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>, <_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung																				
1	X0	<SPCA>	REAL	Bezugspunkt für Lochreihe in der 1. Achse (abs)																				
2	Y0	<SPCO>	REAL	Bezugspunkt für Lochreihe in der 2. Achse (abs)																				
3	α0	<STA1>	REAL	Grund-Drehwinkel (Winkel zur 1. Achse)																				
4	L0	<FDIS>	REAL	Abstand der 1. Bohrung vom Bezugspunkt																				
5	L	<DBH>	REAL	Abstand zwischen den Bohrungen																				
6	N	<NUM>	INT	Anzahl der Bohrungen																				
7		<_VARI>	INT	reserviert																				
8		<_UMODE>	INT	reserviert																				
9		<_HIDE>	STRING [200]	Ausgeblendete Positionen <ul style="list-style-type: none"> max. 198 Zeichen Angabe der fortlaufenden Positionsnummer, z. B. "1,3" (Positionen 1 und 3 werden nicht ausgeführt) 																				
10		<_NSP>	INT	reserviert																				
11		<_DMODE>	INT	Anzeigemodus <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2">EINER:</td> <td colspan="2">Bearbeitungsebene G17/18/19</td> </tr> <tr> <td style="width: 50px;">0 =</td> <td colspan="3">Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv</td> </tr> <tr> <td>1 =</td> <td colspan="3">G17 (nur im Zyklus aktiv)</td> </tr> <tr> <td>2 =</td> <td colspan="3">G18 (nur im Zyklus aktiv)</td> </tr> <tr> <td>3 =</td> <td colspan="3">G19 (nur im Zyklus aktiv)</td> </tr> </table>	EINER:		Bearbeitungsebene G17/18/19		0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv			1 =	G17 (nur im Zyklus aktiv)			2 =	G18 (nur im Zyklus aktiv)			3 =	G19 (nur im Zyklus aktiv)		
EINER:		Bearbeitungsebene G17/18/19																						
0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv																							
1 =	G17 (nur im Zyklus aktiv)																							
2 =	G18 (nur im Zyklus aktiv)																							
3 =	G19 (nur im Zyklus aktiv)																							

19.1.4 HOLES2 - Lochkreis

Syntax

HOLES2 (<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>, <_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung			
1	X0	<CPA>	REAL	Mittelpunkt für Lochkreis in der 1. Achse (abs) Bezugspunkt in der 1. Achse (bei XY) (bei XA, YB, ZC)			
2	Y0	<CPO>	REAL	Mittelpunkt für Lochkreis in der 2. Achse (abs) Bezugspunkt in der 2. Achse (bei XY) (bei XA, YB, ZC)			
3	R	<RAD>	REAL	Radius des Lochkreises (bei XY)			
4	α_0	<STA1>	REAL	Anfangswinkel oder 1. Rundachsposition (bei XY) (bei XA, YB, ZC)			
5	α_1	<INDA>	REAL	Fortschaltwinkel (nur bei Teilkreis) (bei XY, XA, YB, ZC)			
				< 0 = Uhrzeigersinn > 0 = Gegenuhrzeigersinn			
6	N	<NUM>	INT	Anzahl der Positionen			
7		<_VARI>	INT	Bearbeitungsart			
				EINER:	reserviert		
				ZEHNER:	Positionierungsart		
				0 =	Position anfahren - Linear		
				1 =	Position anfahren - auf Kreisbahn		
				HUNDETER:	reserviert		
				TAUSENDER:	Kreismuster		
				0 =	Kompatibilitätsmodus, wenn INDA = 0 dann Vollkreis, INDA <> 0 dann Teilkreis		
				1 =	Vollkreis		
				2 =	Teilkreis		
				ZEHNTAUSENDER:	Positionsmuster mit Rundachse		
				0 =	XY (ohne Rundachse)	(bei XY)	
				1 =	XA (X-Achse und Rundachse um X)	(nur bei XA)	
				2 =	YB (Y-Achse und Rundachse um Y)	(nur bei YB)	
3 =	ZC (Z-Achse und Rundachse um C)	(nur bei ZC)					
EINEMILLION + HUNDERTTAUSENDER:	Offset (bei mehreren Rundachsen um die gleiche Achse; wenn Index zu groß, dann 1. Achse)						
00 =	1. A, B oder C-Achse						
01 =	2. A, B oder C-Achse						
...							
10 =	20. A, B oder C-Achse						
8		<_UMODE>	INT	reserviert			
9		<_HIDE>	STRING [200]	reserviert			
10		<_NSP>	INT	reserviert			

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
11		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/18/19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
	3 =	G19 (nur im Zyklus aktiv)			

19.1.5 POCKET3 - Rechtecktasche fräsen

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<_RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Taschentiefe (abs/ink), siehe <_AMODE>
5	L	<_LENG>	REAL	Taschenlänge (ink, mit Vorzeichen einzugeben)
6	W	<_WID>	REAL	Taschenbreite (ink, mit Vorzeichen einzugeben)
7	R	<_CRAD>	REAL	Eckenradius der Tasche
8	X0	<_PA>	REAL	Bezugspunkt, 1. Achse (abs)
9	YO	<_PO>	REAL	Bezugspunkt, 2. Achse (abs)
10	α0	<_STA>	REAL	Drehwinkel, Winkel zwischen Längsachse (L) und 1. Achse
11	DZ	<_MID>	REAL	maximale Tiefenzustellung
12	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene
13	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe
14	F	<_FFP1>	REAL	Vorschub in der Ebene
15	FZ	<_FFD>	REAL	Zustellvorschub Tiefe
16		<_CDIR>	INT	Fräsrichtung:
				0 =
	1 =	Gegenlauf		

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
17		<_VARI>	INT	Bearbeitungsart	
				EINER:	1 = Schruppen
				2 = Schlichten	
				4 = Schlichten Rand	
				5 = Anfassen	
				ZEHNER:	0 = vorgebohrt, Zustellung mit G0
				1 = senkrecht, Zustellung mit G1	
				2 = helikal	
				3 = pendeln auf Taschenlängsachse	
				HUNDERTER:	reserviert
18	DXY	<_MIDA>	REAL	maximale Ebenenzustellung, Einheit siehe <_AMODE>	
19	L1	<_AP1>	REAL	Länge der Vorbearbeitung (ink)	
20	W1	<_AP2>	REAL	Breite der Vorbearbeitung (ink)	
21	AZ	<_AD>	REAL	Tiefe der Vorbearbeitung (ink)	
22	ER	<_RAD1>	REAL	Radius der Helixbahn beim Eintauchen helikal	
	EW			maximaler Eintauchwinkel für pendelnd	
23	EP	<_DP1>	REAL	Helixsteigung bei Eintauchen helikal	
24		<_UMODE>	INT	reserviert	
25	FS	<_FS>	REAL	Fasenbreite (ink)	
26	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfassen (abs/ink), siehe <_AMODE>	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
27		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)		
				EINER:	reserviert	
				ZEHNER:	reserviert	
				HUNDERTER:	Auswahl Bearbeitung/nur Startpunktbe- rechnung	
					0 =	Kompatibilitätsmodus
					1 =	normale Bearbeitung
				TAUSENDER:	Bemaßung über Mitte/Ecke	
					0 =	Kompatibilitätsmodus
					1 =	Bemaßung über Mitte
					2 =	Bemaßung Eckpunkt, Taschenla- ge +LENG/+WID
					3 =	Bemaßung Eckpunkt, Taschenla- ge -LENG/+WID
					4 =	Bemaßung Eckpunkt, Taschenla- ge +LENG/-WID
				ZEHNTAUSENDER:	Komplettbearbeitung/Nachbearbeitung	
					0 =	Kompatibilitätsmodus (<_AP1>, <_AP2> und <_AD> wie bisher behandeln)
1 =	Komplettbearbeitung					
2 =	Nachbearbeitung					
28		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/G18/G19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene ak- tiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
				ZEHNER:	3 =	G19 (nur im Zyklus aktiv)
					Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub	
				HUNDERTER:	0 =	Kompatibilitätsmodus
					1 =	G-Code wie vor Zyklusaufruf. G94/G95 für Flächen- und Tiefen- vorschub gleich
				TAUSENDER:	---	reserviert
				ZEHNTAUSENDER:	---	reserviert
					Technologieskalierung innerhalb von Zy- klenmasken (Seite 734)	
					0 =	Eingabe: komplett
				1 =	Eingabe: einfach	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
29		<_AMODE>	INT	Alternativmodus		
				EINER:	Taschentiefe (Z1)	
					0 =	absolut (Kompatibilitätsmodus)
					1 =	inkrementell
				ZEHNER:	Einheit für Ebenenzustellung (DXY)	
					0 =	mm
					1 =	% vom Werkzeugdurchmesser
				HUNDERTER:	Eintauchtiefe bei Anfasen (ZFS)	
					0 =	absolut
	1 =	inkrementell				

19.1.6 POCKET4 - Kreistasche fräsen

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<_RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Taschentiefe (abs/ink), siehe <_AMODE>
5	Ø	<_CDIAM>	REAL	Taschendurchmesser oder Taschenradius, siehe <_DMODE>
6	X0	<_PA>	REAL	Bezugspunkt 1. Achse (abs)
7	Y0	<_PO>	REAL	Bezugspunkt 2. Achse (abs)
8	DZ	<_MID>	REAL	maximale Tiefenzustellung, siehe <_VARI> = ebenenweise maximale Helixsteigung, siehe <_VARI> = helikal
9	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene
10	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe
11	F	<_FFP1>	REAL	Vorschub für Flächenbearbeitung
12	FZ	<_FFD>	REAL	Zustellvorschub Tiefe
13		<_CDIR>	INT	Fräsrichtung
				0 =
	1 =	Gegenlauf		

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
14		<_VARI>	INT	Bearbeitungsart	
				EINER:	Bearbeitung
					1 = Schruppen
					2 = Schlichten
					4 = Schlichten Rand
					5 = Anfasen
				ZEHNER:	Zustellungsart (Schruppen und Schlichten)
					0 = vorgebohrt, Zustellung mit G0 (Tasche ist vorgearbeitet)
					1 = senkrecht, Zustellung mit G1
					2 = helikal
	HUNDERTER:	reserviert			
	TAUSENDER:				
		0 = ebenenweise			
		1 = helikal			
15	DXY	<_MIDA>	REAL	maximale Ebenenzustellung, siehe <_AMODE>, 0 = 0,8 x WZG-Durchmesser	
16	∅	<_AP1>	REAL	Durchmesser/Radius der Vorbearbeitung (ink)	
17	AZ	<_AD>	REAL	Tiefe der Vorbearbeitung (ink)	
18	ER	<_RAD1>	REAL	Radius der Helixbahn beim Eintauchen helikal	
19	EP	<_DP1>	REAL	Helixsteigung bei Eintauchen auf Helixbahn	
20		<_UMODE>	INT	reserviert	
21	FS	<_FS>	REAL	Fasenbreite (ink)	
22	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE>	
23		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	Bearbeitung/Startpunktberechnung
					0 = Kompatibilitätsmodus
					1 = normale Bearbeitung
				TAUSENDER:	reserviert
				ZEHNTAUSENDER:	Komplettbearbeitung/Nachbearbeitung
					0 = Kompatibilitätsmodus (<_AP1> und <_AD> wie bisher behandeln)
					1 = Komplettbearbeitung
	2 = Nachbearbeitung				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
24		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/18/19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
					3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub	
					0 =	Kompatibilitätsmodus
					1 =	G-Code wie vor Zyklusaufwurf. G94/G95 für Flächen- und Tiefenvorschub gleich
				HUNDERTER:		
					0 =	Kompatibilitätsmodus (<_CDIAM>/<_AP1> als Radius eingegeben)
					1 =	<_CDIAM>/<_AP1> als Durchmesser eingegeben
TAUSENDER:	---	reserviert				
ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)					
	0 =	Eingabe: komplett				
	1 =	Eingabe: einfach				
25		<_AMODE>	INT	Alternativmodus		
				EINER:	Taschentiefe (Z1)	
					0 =	absolut (Kompatibilitätsmodus)
					1 =	inkrementell
				ZEHNER:	Einheit für Zustellbreite (DXY)	
					0 =	mm
					1 =	% vom Werkzeugdurchmesser
				HUNDERTER:	Eintauchtiefe bei Anfassen (ZFS)	
					0 =	absolut
1 =	inkrementell					

19.1.7 SLOT1 - Längsnut

Syntax

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>

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<RTP>	REAL	Rückzugsebene (abs)
2	Z0	<RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Nuttiefe (abs)
5		<_DPR>	REAL	Nuttiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben)
6		<NUM>	INT	Anzahl der Nuten = 1
7	L	<LENG>	REAL	Länge der Nut
8	W	<WID>	REAL	Breite der Nut
9	X0	<_CPA>	REAL	Bezugspunkt, 1. Achse der Ebene
10	Y0	<_CPO>	REAL	Bezugspunkt, 2. Achse der Ebene
11		<RAD>	REAL	reserviert
12	α	<STA1>	REAL	Drehwinkel
13		<INDA>	REAL	reserviert
14	FZ	<FFD>	REAL	Zustellvorschub Tiefe
15	F	<FFP1>	REAL	Vorschub
16	DZ	<_MID>	REAL	maximale Tiefenzustellung
17		<CDIR>	INT	Fräsrichtung 0 = Gleichlauf 1 = Gegenlauf
18	UXY	<_FAL>	REAL	Schlichtaufmass Ebene oder Nutrand
19		<VARI>	INT	Bearbeitungsart EINER: 0 = reserviert 1 = Schruppen 2 = Schlichten 4 = Schlichten Rand (nur Rand bearbeiten) 5 = Anfasen ZEHNER: Anfahren 0 = vorgebohrt, Zustellung mit G0 (Nut ist vorbearbeitet) 1 = senkrecht, Zustellung mit G1 2 = helikal 3 = pendelnd HUNDETER: reserviert
20	DZF	<_MIDF>	REAL	reserviert
21	FF	<FFP2>	REAL	reserviert
22	SF	<SSF>	REAL	reserviert

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
23	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe	
24	ER	<_STA2>	REAL	Radius der Helixbahn beim Eintauchen helikal	
	EW			maximaler Eintauchwinkel für pendelnd	
25	EP	<_DP1>	REAL	Eintauchtiefe pro Umdrehung für Helix	
26		<_UMODE>	INT	reserviert	
27	FS	<_FS>	REAL	Fasenbreite (ink) bei Anfasen	
28	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugschärfe) bei Anfasen (abs/ink), siehe <_AMODE>	
29		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	Auswahl Bearbeitung oder nur Startpunktberechnung
					1 = normale Bearbeitung
				TAUSENDER:	Bemaßung Bezugspunkt, Nutlage
					0 = Mitte
					1 = links innen +L
					2 = rechts innen -L
					3 = linker Rand +L
4 = rechter Rand -L					
30		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/18/19
					0 = Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
					1 = G17 (nur im Zyklus aktiv)
					2 = G18 (nur im Zyklus aktiv)
					3 = G19 (nur im Zyklus aktiv)
				ZEHNER:	reserviert
				HUNDERTER:	reserviert
				TAUSENDER:	Kennung SW-Version
					1 = Funktionserweiterung SLOT1
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)
					0 = Eingabe: komplett
					1 = Eingabe: einfach

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
31		<_AMODE>	INT	Alternativmodus		
				EINER:	Endtiefe Z1 (abs/ink)	
					0 =	Kompatibilität
					1 =	Z1 (ink)
					2 =	Z1 (abs)
				ZEHNER:	reserviert	
				HUNDERTER:	Eintauchtiefe bei Anfasen ZFS	
0 =	ZFS (abs)					
	1 =	ZFS (ink)				

Hinweis

Der Zyklus ist gegenüber früheren SW-Versionen mit neuen Funktionen ausgestattet. Das hat zur Folge, dass bestimmte Parameter in der Eingabemaske nicht mehr angezeigt werden (<NUM>, <RAD>, <INDA>). Mehrere Nuten auf einem Positionsmuster sind mittels "MCALL" und Aufruf des gewünschten Positionsmusters, z B HOLES2, programmierbar.

19.1.8 SLOT2 - Kreisnut

Syntax

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>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<RTP>	REAL	Rückzugsebene (abs)
2	Z0	<RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Nuttiefe (abs)
5		<_DPR>	REAL	Nuttiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben)
6	N	<NUM>	INT	Anzahl der Nuten
7	α1	<AFSL>	REAL	Öffnungswinkel der Nut
8	W	<WID>	REAL	Breite der Nut
9	X0	<_CPA>	REAL	Bezugspunkt = Mittelpunkt des Kreises, 1. Achse der Ebene
10	Y0	<_CPO>	REAL	Bezugspunkt = Mittelpunkt des Kreises, 2. Achse der Ebene
11	R	<RAD>	REAL	Radius des Kreises

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
12	$\alpha 0$	<STA1>	REAL	Anfangswinkel	
13	$\alpha 2$	<INDA>	REAL	Fortschaltwinkel	
14	FZ	<FFD>	REAL	Zustellvorschub Tiefe	
15	F	<FFP1>	REAL	Vorschub	
16	DZ	<_MID>	REAL	maximale Tiefenzustellung	
17		<CDIR>	INT	Fräsrichtung	
				0 = Gleichlauf 1 = Gegenlauf	
18	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene oder Nutrand	
19		<VARI>	INT	Bearbeitungsart	
				EINER:	
				0 = Komplettbearbeitung	
				1 = Schruppen	
				2 = Schichten	
				3 = Schichten Rand	
				5 = Anfasen	
				ZEHNER:	
				0 = Zwischenpositionieren mit G0-Gerade	
				1 = Zwischenpositionieren auf Kreisbahn	
HUNDERTER:	reserviert				
TAUSENDER:					
0 = Kompatibilitätsmode, wenn <INDA> = 0 dann Vollkreis, <INDA> <> 0 dann Teilkreis)					
1 = Vollkreis					
2 = Teilkreis					
20	DZF	<_MIDF>	REAL	reserviert	
21		<FFP2>	REAL	reserviert	
22		<SSF>	REAL	reserviert	
23	FF	<_FFCP>	REAL	reserviert	
24		<_UMODE>	INT	reserviert	
25	FS	<_FS>	REAL	Fasenbreite (Ink)	
26	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE>	
27		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	Auswahl Bearbeitung oder nur Startpunktberechnung
				0 = Kompatibilitätsmodus 1 = normale Bearbeitung	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung					
28		<_DMODE>	INT	Anzeigemodus					
				EINER:	Bearbeitungsebene G17/18/19				
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv			
					1 =	G17 (nur im Zyklus aktiv)			
					2 =	G18 (nur im Zyklus aktiv)			
								3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	reserviert				
				HUNDERTER:	reserviert				
				TAUSENDER:	Kennung SW-Version				
					1 =	Funktionen SLOT2 ab SW 2.5			
ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)								
	0 =	Eingabe: komplett							
	1 =	Eingabe: einfach							
29		<_AMODE>	INT	Alternativmodus					
				EINER:	Endtiefe Z1 (abs/ink)				
					0 =	Kompatibilität			
					1 =	Z1 (ink)			
								2 =	Z1 (abs)
				ZEHNER:	reserviert				
				HUNDERTER:	Eintauchtiefe bei Anfasen ZFS				
0 =	ZFS (abs)								
1 =	ZFS (ink)								

19.1.9 LONGHOLE - Langloch

Syntax

LONGHOLE (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>, <_CPO>, <RAD>, <STAL>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugabe)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
4	Z1	<_DP>	REAL	Langlochtiefe (abs)
5		<_DPR>	REAL	Langlochtiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben)
6		<NUM>	INT	Anzahl der Langlöcher = 1
7	L	<LENG>	REAL	Länge Langloch
8	X0	<_CPA>	REAL	Bezugspunkt, 1. Achse der Ebene
9	Y0	<_CPO>	REAL	Bezugspunkt, 2. Achse der Ebene
10		<RAD>	REAL	reserviert
11	α0	<STAL>	REAL	Drehwinkel
12		<INDA>	REAL	reserviert
13	FZ	<FFD>	REAL	Zustellvorschub Tiefe
14	F	<FFP1>	REAL	Vorschub
15	DZ	<MID>	REAL	maximale Tiefenzustellung
16		<_VARI>	INT	Bearbeitungsart
			EINER:	Zustellart
				1 = senkrecht mit G1
				3 = pendelnd
			HUNDERTER:	reserviert
17		<_UMODE>	INT	reserviert
18		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)
			EINER:	reserviert
			ZEHNER:	reserviert
			HUNDERTER:	Auswahl Bearbeitung oder nur Startpunktberechnung
				0 = Kompatibilitätsmode
				1 = normale Bearbeitung
			TAUSENDER:	Bemaßung Bezugspunkt, Nutlage
				0 = Mitte
				1 = links innen +L
				2 = rechts innen -L
				3 = linker Rand +L
				4 = rechter Rand -L

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
19		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/18/19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)	
				ZEHNER:	Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub	
					0 =	Kompatibilitätsmode
				1 =	G-Code wie vor Zyklusaufwurf. G94/G95 für Flächen- und Tiefenvorschub gleich	
				HUNDERTER:	reserviert	
TAUSENDER:	Kennung SW-Version					
	1 =	Funktionserweiterung LONGHOLE (Bemaßung Bezugspunkt)				
20		<_AMODE>	INT	Alternativmodus		
				EINER:	Endtiefe Z1 (abs/ink)	
					0 =	Kompatibilität
					1 =	Z1 (ink)
					2 =	Z1 (abs)

Hinweis

Der Zyklus ist gegenüber früheren SW-Versionen mit neuen Funktionen ausgestattet. Das hat zur Folge, dass bestimmte Parameter in der Eingabemaske nicht mehr angezeigt werden (<NUM>, <RAD>, <INDA>). Mehrere Nuten auf einem Positionsmuster sind mittels "MCALL" und Aufruf des gewünschten Positionsmusters, z. B. HOLES2, programmierbar.

19.1.10 CYCLE60 - Gravurzyklus

Syntax

```
CYCLE60 (<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1		<_TEXT>	STRING [200]	zu gravierender Text (maximal 100 Zeichen)
2	RP	<_RTP>	REAL	Rückzugsebene (abs)
3	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
4	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Referenzebene, ohne Vorzeichen eingeben)
5	Z1	<_DP>	REAL	Tiefe (abs), siehe <_AMODE>
6	Z1	<_DPR>	REAL	Tiefe (ink), siehe <_AMODE>
7	X0	<_PA>	REAL	Bezugspunkt, 1. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI>
	R			Bezugspunkt, Länge (Radius) - polar, siehe <_VARI>
8	Y0	<_PO>	REAL	Bezugspunkt, 2. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI>
	α 0			Bezugspunkt, Winkel bezogen auf die 1. Achse - polar, siehe <_VARI>
9	α 1	<_STA>	REAL	Textrichtung, Winkel der Textlinie bezogen auf die 1. Achse), siehe <_VARI>
10	XM	<_CP1>	REAL	Mittelpunkt des Textkreises, 1. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI>
	LM			Mittelpunkt des Textkreises, Länge (Radius) bezogen auf WNP - polar, siehe <_VARI>
11	YM	<_CP2>	REAL	Mittelpunkt des Textkreises, 2. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI>
	α M			Mittelpunkt des Textkreises, Winkel bezogen auf die 1. Achse - polar, siehe <_VARI>
12	W	<_WID>	REAL	Zeichenhöhe (ohne Vorzeichen eingeben)
13	DX1	<_DF>	REAL	Zeichenabstand / Gesamtbreite, siehe <_VARI>
	DX2			
	α 2			Öffnungswinkel, siehe <_VARI>
14	FZ	<_FFD>	REAL	Zustellvorschub Tiefe, siehe <_DMODE>
15	F	<_FFP1>	REAL	Vorschub für Flächenbearbeitung

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
16		<_VARI>	INT	Bearbeitung (Ausrichtung und Bezugspunkt des Gravurtextes)		
				EINER:	Bezugspunkt	
					0 =	rechtwinklig
					1 =	polar
				ZEHNER:	Textausrichtung	
					0 =	Text auf einer Linie
					1 =	Text auf einem Kreisbogen oben
					2 =	Text auf einem Kreisbogen unten
				HUNDERTER:	reserviert	
				TAUSENDER:	Bezugspunkt des Textes horizontal	
					0 =	links
					1 =	mittig
					2 =	rechts
				ZEHNTAUSENDER:	Bezugspunkt des Textes vertikal	
					0 =	unten
					1 =	mittig
					2 =	oben
HUNDERTTAUSENDER:	Textlänge					
	0 =	Zeichenabstand				
	1 =	Gesamtbreite des Textes (nur bei Text linear)				
	2 =	Öffnungswinkel (nur bei Text auf Kreisbogen)				
EINEMILLION:	Kreismittelpunkt					
	0 =	rechtwinklig (kartesisch)				
	1 =	polar				
ZEHNMILLIONEN:	Spiegelschrift					
	0 =	Kompatibilität				
	1 =	Spiegelschrift EIN				
	2 =	Spiegelschrift AUS				
17		<_CODEP>	INT	Nummer der Codepage für die Schrift (z. Z. nur 1252)		
18		<_UMODE>	INT	reserviert		
19		_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)		
				EINER:	reserviert	
				ZEHNER :	reserviert	
				HUNDERTER:	Auswahl Bearbeitung / nur Startpunktbe- rechnung	
0 =	Kompatibilitätsmodus					
	1 =	normale Bearbeitung				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
20		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/18/19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)	
				ZEHNER :	Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub	
0 =	Kompatibilitätsmodus					
1 =	G-Code wie vor Zyklusaufwurf. G94/G95 für Flächen- und Tiefenvorschub gleich					
21		<_AMODE>	INT	Alternativmodus		
				EINER:	Endtiefe (<_DP>, <_DPR>)	
					0 =	Kompatibilität
					1 =	inkrementell (<_DPR>)
					2 =	absolut (<_DP>)

19.1.11 CYCLE61 - Planfräsen

Syntax

```
CYCLE61 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<_RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse, Höhe Rohteil (abs)
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Höhe Fertigteil (abs/ink), siehe <_AMODE>
5	X0	<_PA>	REAL	Eckpunkt 1 in der 1. Achse (abs)
6	Y0	<_PO>	REAL	Eckpunkt 1 in der 2. Achse (abs)
7	X1	<_LENG>	REAL	Eckpunkt 2 in der 1. Achse (abs/ink), siehe <_AMODE>
8	Y1	<_WID>	REAL	Eckpunkt 2 in der 2. Achse (abs/ink), siehe <_AMODE>
9	DZ	<_MID>	REAL	maximale Tiefenzustellung
10	DXY	<_MIDA>	REAL	maximale Ebenenzustellung (Einheit, siehe <_AMODE>)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
11	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe	
12	F	<_FFP1>	REAL	Bearbeitungsvorschub	
13		<_VARI>	INT	Bearbeitungsart	
				EINER:	Bearbeitung
				1 =	Schruppen
				2 =	Schlichten
				ZEHNER:	Bearbeitungsrichtung
				1 =	parallel zur 1. Achse, eine Richtung
				2 =	parallel zur 2. Achse, eine Richtung
				3 =	parallel zur 1. Achse, wechselnde Richtung
4 =	parallel zur 2. Achse, wechselnde Richtung				
14		<_LIM>	INT	Eingrenzungen	
				EINER:	Eingrenzung 1. Achse minus
				0 =	nein
				1 =	ja
				ZEHNER:	Eingrenzung 1. Achse plus
				0 =	nein
				1 =	ja
				HUNDERTER:	Eingrenzung 2. Achse minus
				0 =	nein
				1 =	ja
				TAUSENDER:	Eingrenzung 2. Achse plus
				0 =	nein
1 =	ja				
15		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/18/19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
3 =	G19 (nur im Zyklus aktiv)				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
16		<_AMODE>	INT	Alternativmodus		
				EINER:	Endtiefe (<_DP>)	
					0 =	absolut
					1 =	inkrementell
				ZEHNER:	Einheit für Ebenenzustellung (<_MIDA>)	
					0 =	mm
					1 =	% vom Werkzeugdurchmesser
				HUNDERTER:	reserviert	
				TAUSENDER:	Länge der Fläche	
					0 =	inkrementell
	1 =	absolut				
ZEHNTAUSENDER:	Breite der Fläche					
	0 =	inkrementell				
	1 =	absolut				

19.1.12 CYCLE62 - Konturaufruf

Syntax

CYCLE62 (<_KNAME>, <_TYPE>, <_LAB1>, <_LAB2>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
1	PRG/CON	<_KNAME>	STRING [140]	Konturname oder Unterprogrammname, muss nicht programmiert werden bei _TYPE = 2	
2		<_TYPE>	INT	Bestimmung der Kontureingabe	
				0 =	Unterprogramm
				1 =	Konturname
				2 =	Labels
	3 =	Labels in Unterprogramm			
3	LAB1	<_LAB1>	STRING[32]	Label 1, Konturanfang	
4	LAB2	<_LAB2>	STRING[32]	Label 2, Konturende	

19.1.13 CYCLE63 - Konturtasche fräsen

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung																							
1	PRG	<_PRG>	STRING [100]	Name Ausräumprogramm																							
2		<_VARI>	INT	<table border="1"> <tr> <td colspan="2">Bearbeitungsart</td> </tr> <tr> <td rowspan="5">EINER:</td> <td>Technologische Bearbeitung</td> </tr> <tr> <td>1 = Schruppen</td> </tr> <tr> <td>3 = Schlichten Boden</td> </tr> <tr> <td>4 = Schlichten Rand</td> </tr> <tr> <td>5 = Anfasen</td> </tr> <tr> <td rowspan="4">ZEHNER:</td> <td>Zustellungsart</td> </tr> <tr> <td>0 = Eintauchen mittig</td> </tr> <tr> <td>1 = Eintauchen helikal</td> </tr> <tr> <td>2 = Eintauchen pendelnd</td> </tr> <tr> <td>HUNDERTER:</td> <td>reserviert</td> </tr> <tr> <td rowspan="3">TAUSENDER:</td> <td>Abhebemodus</td> </tr> <tr> <td>0 = Abheben auf Rückzugsebene</td> </tr> <tr> <td>1 = Abheben auf Bezugspunkt + Sicherheitsabstand</td> </tr> <tr> <td rowspan="3">ZEHNTAUSENDER:</td> <td>Startpunkt bei Schruppen und Schlichten Boden</td> </tr> <tr> <td>0 = auto</td> </tr> <tr> <td>1 = manuell</td> </tr> </table>	Bearbeitungsart		EINER:	Technologische Bearbeitung	1 = Schruppen	3 = Schlichten Boden	4 = Schlichten Rand	5 = Anfasen	ZEHNER:	Zustellungsart	0 = Eintauchen mittig	1 = Eintauchen helikal	2 = Eintauchen pendelnd	HUNDERTER:	reserviert	TAUSENDER:	Abhebemodus	0 = Abheben auf Rückzugsebene	1 = Abheben auf Bezugspunkt + Sicherheitsabstand	ZEHNTAUSENDER:	Startpunkt bei Schruppen und Schlichten Boden	0 = auto	1 = manuell
Bearbeitungsart																											
EINER:	Technologische Bearbeitung																										
	1 = Schruppen																										
	3 = Schlichten Boden																										
	4 = Schlichten Rand																										
	5 = Anfasen																										
ZEHNER:	Zustellungsart																										
	0 = Eintauchen mittig																										
	1 = Eintauchen helikal																										
	2 = Eintauchen pendelnd																										
HUNDERTER:	reserviert																										
TAUSENDER:	Abhebemodus																										
	0 = Abheben auf Rückzugsebene																										
	1 = Abheben auf Bezugspunkt + Sicherheitsabstand																										
ZEHNTAUSENDER:	Startpunkt bei Schruppen und Schlichten Boden																										
	0 = auto																										
	1 = manuell																										
3	RP	<_RP>	REAL	Rückzugsebene (abs)																							
4	Z0	<_Z0>	REAL	Bezugspunkt der Werkzeugachse (abs)																							
5	SC	<_SC>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)																							
6	Z1	<_Z1>	REAL	Endtiefe, (siehe <_AMODE> EINER)																							
7	F	<_F>	REAL	Vorschub in der Ebene Schruppen/Schlichten																							
8	FZ	<_FZ>	REAL	Zustellvorschub Tiefe																							
9	DXY	<_DXY>	REAL	Zustellung Ebene - Einheit, (siehe <_AMODE> ZEHNER)																							
10	DZ	<_DZ>	REAL	Zustellung Tiefe																							
11	UXY	<_UXY>	REAL	Schlichtaufmaß Ebene																							
12	UZ	<_UZ>	REAL	Schlichtaufmaß Tiefe																							

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
13		<_CDIR>	INT	Fräsrichtung	
				0 = Gleichlauf 1 = Gegenlauf	
14	XS	<_XS>	REAL	Startpunkt X, absolut	
15	YS	<_YS>	REAL	Startpunkt Y, absolut	
16	ER	<_ER>	REAL	Eintauchen helikal: Radius	
17	EP	<_EP>	REAL	Eintauchen helikal: Steigung	
18	EW	<_EW>	REAL	Eintauchen pendelnd: maximaler Eintauchwinkel	
19	FS	<_FS>	REAL	Fasenbreite (ink) bei Anfassen	
20	ZFS	<_ZFS>	REAL	Eintauchtiefe Werkzeugspitze bei Anfassen, (siehe <_AMODE> HUNDERTER)	
21	TR	<_TR>	STRING[32]	Referenzwerkzeugname bei Restmaterialbearbeitung	
22	DR	<_DR>	INT	Referenzwerkzeug D-Nummer bei Restmaterialbearbeitung	
23		<_UMODE>	INT	reserviert	
24		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	Auswahl Bearbeitung/nur Startpunktberechnung
				0 =	normale Bearbeitung (kein Kompatibilitätsmodus erforderlich)
				1 =	normale Bearbeitung
2 =	reserviert				
25		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	reserviert
				HUNDERTER:	Technologiemodus
				1 =	Tasche
				2 =	Zapfen
				TAUSENDER:	Restmaterial bearbeiten
				0 =	nein
				1 =	ja
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)
0 =	Eingabe: komplett				
1 =	Eingabe: einfach				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
26		<_AMODE>	INT	Alternativmodus		
				EINER:	Endtiefe (Z1)	
					0 =	absolut (Kompatibilitätsmodus)
					1 =	inkrementell
				ZEHNER:	Einheit für Ebenenzustellung (DXY)	
					0 =	mm
					1 =	% vom Werkzeugdurchmesser
				HUNDETER:	Eintauchtiefe bei Anfasen (ZFS)	
					0 =	absolut
					1 =	inkrementell
TAUSENDER:	---	reserviert				

19.1.14 CYCLE64 - Konturtasche vorbohren

Syntax

```
CYCLE64 (<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
1	PRG	<_PRG>	STRING [100]	Name Bohr-/Zentrierprogramm	
2		<_VARI>	INT	Bearbeitungsart	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDETER:	reserviert
				TAUSENDER:	Abhebemodus
0 =	Abheben auf Rückzugsebene				
1 =	Abheben auf Bezugspunkt + Sicherheitsabstand				
3	RP	<_RP>	REAL	Rückzugsebene (abs)	
4	Z0	<_Z0>	REAL	Bezugspunkt (abs)	
5	SC	<_SC>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)	
6	Z1	<_Z1>	REAL	Bohr-/Zentriertiefe, (siehe <_AMODE> EINER)	
7	F	<_F>	REAL	Vorschub Bohren/Zentrieren	
8	DXY	<_DXY>	REAL	Zustellung Ebene - Einheit, (siehe <_AMODE> ZEHNER)	
9	UXY	<_UXY>	REAL	Schlichtaufmaß Ebene	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung																
10	UZ	<_UZ>	REAL	Schlichtaufmaß Tiefe																
11		<_CDIR>	INT	Fräsrichtung <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>Gleichlauf</td> </tr> <tr> <td>1 =</td> <td>Gegenlauf</td> </tr> </table>	0 =	Gleichlauf	1 =	Gegenlauf												
0 =	Gleichlauf																			
1 =	Gegenlauf																			
12	TR	<_TR>	STRING[20]	Referenzwerkzeugname																
13	DR	<_DR>	INT	Referenzwerkzeug D-Nummer																
14		<_UMODE>	INT	reserviert																
15		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte) <table border="1" style="margin-left: 20px;"> <tr> <td>EINER:</td> <td>reserviert</td> </tr> <tr> <td>ZEHNER:</td> <td>reserviert</td> </tr> <tr> <td>HUNDERTER:</td> <td>Auswahl Bearbeitung/nur Startpunktbe- rechnung <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>normale Bearbeitung (kein Kom- patibilitätsmodus erforderlich)</td> </tr> <tr> <td>1 =</td> <td>normale Bearbeitung</td> </tr> <tr> <td>2 =</td> <td>reserviert</td> </tr> </table> </td> </tr> </table>	EINER:	reserviert	ZEHNER:	reserviert	HUNDERTER:	Auswahl Bearbeitung/nur Startpunktbe- rechnung <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>normale Bearbeitung (kein Kom- patibilitätsmodus erforderlich)</td> </tr> <tr> <td>1 =</td> <td>normale Bearbeitung</td> </tr> <tr> <td>2 =</td> <td>reserviert</td> </tr> </table>	0 =	normale Bearbeitung (kein Kom- patibilitätsmodus erforderlich)	1 =	normale Bearbeitung	2 =	reserviert				
EINER:	reserviert																			
ZEHNER:	reserviert																			
HUNDERTER:	Auswahl Bearbeitung/nur Startpunktbe- rechnung <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>normale Bearbeitung (kein Kom- patibilitätsmodus erforderlich)</td> </tr> <tr> <td>1 =</td> <td>normale Bearbeitung</td> </tr> <tr> <td>2 =</td> <td>reserviert</td> </tr> </table>	0 =	normale Bearbeitung (kein Kom- patibilitätsmodus erforderlich)	1 =	normale Bearbeitung	2 =	reserviert													
0 =	normale Bearbeitung (kein Kom- patibilitätsmodus erforderlich)																			
1 =	normale Bearbeitung																			
2 =	reserviert																			
25		<_DMODE>	INT	Anzeigemodus <table border="1" style="margin-left: 20px;"> <tr> <td>EINER:</td> <td>Bearbeitungsebene G17/G18/G19 <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene ak- tiv</td> </tr> <tr> <td>1 =</td> <td>G17 (nur im Zyklus aktiv)</td> </tr> <tr> <td>2 =</td> <td>G18 (nur im Zyklus aktiv)</td> </tr> <tr> <td>3 =</td> <td>G19 (nur im Zyklus aktiv)</td> </tr> </table> </td> </tr> <tr> <td>ZEHNER:</td> <td>Technologiemodus) <table border="1" style="margin-left: 20px;"> <tr> <td>1 =</td> <td>Vorbohren</td> </tr> <tr> <td>2 =</td> <td>Zentrieren</td> </tr> </table> </td> </tr> </table>	EINER:	Bearbeitungsebene G17/G18/G19 <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene ak- tiv</td> </tr> <tr> <td>1 =</td> <td>G17 (nur im Zyklus aktiv)</td> </tr> <tr> <td>2 =</td> <td>G18 (nur im Zyklus aktiv)</td> </tr> <tr> <td>3 =</td> <td>G19 (nur im Zyklus aktiv)</td> </tr> </table>	0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene ak- tiv	1 =	G17 (nur im Zyklus aktiv)	2 =	G18 (nur im Zyklus aktiv)	3 =	G19 (nur im Zyklus aktiv)	ZEHNER:	Technologiemodus) <table border="1" style="margin-left: 20px;"> <tr> <td>1 =</td> <td>Vorbohren</td> </tr> <tr> <td>2 =</td> <td>Zentrieren</td> </tr> </table>	1 =	Vorbohren	2 =	Zentrieren
EINER:	Bearbeitungsebene G17/G18/G19 <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene ak- tiv</td> </tr> <tr> <td>1 =</td> <td>G17 (nur im Zyklus aktiv)</td> </tr> <tr> <td>2 =</td> <td>G18 (nur im Zyklus aktiv)</td> </tr> <tr> <td>3 =</td> <td>G19 (nur im Zyklus aktiv)</td> </tr> </table>	0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene ak- tiv	1 =	G17 (nur im Zyklus aktiv)	2 =	G18 (nur im Zyklus aktiv)	3 =	G19 (nur im Zyklus aktiv)											
0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene ak- tiv																			
1 =	G17 (nur im Zyklus aktiv)																			
2 =	G18 (nur im Zyklus aktiv)																			
3 =	G19 (nur im Zyklus aktiv)																			
ZEHNER:	Technologiemodus) <table border="1" style="margin-left: 20px;"> <tr> <td>1 =</td> <td>Vorbohren</td> </tr> <tr> <td>2 =</td> <td>Zentrieren</td> </tr> </table>	1 =	Vorbohren	2 =	Zentrieren															
1 =	Vorbohren																			
2 =	Zentrieren																			
26		<_AMODE>	INT	Alternativmodus <table border="1" style="margin-left: 20px;"> <tr> <td>EINER:</td> <td>Bohr-/Zentriertiefe Z1 <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>absolut (Kompatibilitätsmodus)</td> </tr> <tr> <td>1 =</td> <td>inkrementell</td> </tr> </table> </td> </tr> <tr> <td>ZEHNER:</td> <td>Einheit für Ebenenzustellung (DXY) <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>mm</td> </tr> <tr> <td>1 =</td> <td>% vom Werkzeugdurchmesser</td> </tr> </table> </td> </tr> </table>	EINER:	Bohr-/Zentriertiefe Z1 <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>absolut (Kompatibilitätsmodus)</td> </tr> <tr> <td>1 =</td> <td>inkrementell</td> </tr> </table>	0 =	absolut (Kompatibilitätsmodus)	1 =	inkrementell	ZEHNER:	Einheit für Ebenenzustellung (DXY) <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>mm</td> </tr> <tr> <td>1 =</td> <td>% vom Werkzeugdurchmesser</td> </tr> </table>	0 =	mm	1 =	% vom Werkzeugdurchmesser				
EINER:	Bohr-/Zentriertiefe Z1 <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>absolut (Kompatibilitätsmodus)</td> </tr> <tr> <td>1 =</td> <td>inkrementell</td> </tr> </table>	0 =	absolut (Kompatibilitätsmodus)	1 =	inkrementell															
0 =	absolut (Kompatibilitätsmodus)																			
1 =	inkrementell																			
ZEHNER:	Einheit für Ebenenzustellung (DXY) <table border="1" style="margin-left: 20px;"> <tr> <td>0 =</td> <td>mm</td> </tr> <tr> <td>1 =</td> <td>% vom Werkzeugdurchmesser</td> </tr> </table>	0 =	mm	1 =	% vom Werkzeugdurchmesser															
0 =	mm																			
1 =	% vom Werkzeugdurchmesser																			

19.1.15 CYCLE70 - Gewindefräsen

Syntax

```
CYCLE70 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<_RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Gewindelänge (abs, ink), siehe <_AMODE> Auslauf am Bohrungsgrund berücksichtigen (minimum halbe Steigung)
5	Ø	<_DIATH>	REAL	Nenndurchmesser des Gewindes
6	H1	<_H1>	REAL	Gewindetiefe
7	U	<_FAL>	REAL	Schlichtaufmaß
8	P	<_PIT>	REAL	Gewindesteigung (Auswahl <_PITA>: mm, inch, MODUL, Gänge/Zoll)
9	NT	<_NT>	INT	Anzahl Zähne auf der Schneidplatte Werkzeuiglänge immer auf unteren Zahn bezogen!
10	DXY	<_MID>	REAL	Maximale Zustellung pro Schnitt <_MID> > <_H1>: alles mit einem Schnitt
11	F	<_FFR>	REAL	Fräsvorschub
12		<_TYPTH>	INT	Gewindetyp 0 = Innengewinde 1 = Außengewinde
13	X0	<_PA>	REAL	Kreismittelpunkt, 1. Achse (abs)
14	Y0	<_PO>	REAL	Kreismittelpunkt, 2. Achse (abs)
15	αS	<_NSP>	REAL	Startwinkel (mehrgängige Gewinde)
16		<_VARI>	INT	Bearbeitungsart EINER: 1 = Schruppen 2 = Schlichten ZEHNER: 1 = von oben nach unten 2 = von unten nach oben HUNDERTER: 0 = Rechtsgewinde 1 = Linksgewinde
17		<_PITA>	INT	Bewertung der Gewindesteigung 0 = Kompatibilitätsmodus 1 = Steigung in mm 2 = Steigung in Gänge pro Zoll (TPI) 3 = Steigung in inch 4 = Steigung als MODUL
18		<_PITM>	STRING[15]	String als Merker für Gewindesteigungseingabe (nur für die Oberfläche)
19		<_PTAB>	STRING[20]	String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") (nur für die Oberfläche)
20		<_PTABA>	STRING[20]	String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...) (nur für die Oberfläche)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
21		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	Bearbeitung/Startpunktberechnung
				0 =	Kompatibilitätsmodus
1 =	normale Bearbeitung				
22		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
3 =	G19 (nur im Zyklus aktiv)				
23		<_AMODE>	INT	Alternativmodus	
				EINER:	Gewindelänge (<_DP>)
				0 =	absolut
				1 =	inkrementell

19.1.16 CYCLE72 - Bahnfräsen

Syntax

```
CYCLE72 (<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1		<_KNAME>	STRING [141]	Name des Kontur-Unterprogrammes
2	RP	<_RTP>	REAL	Rückzugsebene (abs)
3	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
4	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
5	Z1	<_DP>	REAL	Endpunkt, Endtiefe(abs/ink), siehe <_AMODE>
6	DZ	<_MID>	REAL	maximale Tiefenzustellung (ink, ohne Vorzeichen einzugeben)
7	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene (ink), Aufmaß an der Randkontur
8	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe (ink), Aufmaß am Grund ohne Vorzeichen einzugeben)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung																						
9	FX	<_FFP1>	REAL	Vorschub an der Kontur																						
10	FZ	<_FFD>	REAL	Vorschub für Tiefenzustellung (oder räumliche Zustellung)																						
11		<_VARI>	INT	<table border="1"> <tr> <td colspan="2">Bearbeitungsart</td> </tr> <tr> <td rowspan="3">EINER:</td> <td>Bearbeitung</td> </tr> <tr> <td>1 = Schruppen</td> </tr> <tr> <td>2 = Schlichten</td> </tr> <tr> <td>5 = Anfasen</td> </tr> <tr> <td rowspan="2">ZEHNER:</td> <td>0 = Zwischenwege mit G0</td> </tr> <tr> <td>1 = Zwischenwege mit G1</td> </tr> <tr> <td rowspan="4">HUNDERTER:</td> <td>Rückzug am Konturende</td> </tr> <tr> <td>0 = Rückzug am Konturende auf Bezugspunkt</td> </tr> <tr> <td>1 = Rückzug am Konturende auf Bezugspunkt + <_SDIS></td> </tr> <tr> <td>2 = Rückzug am Konturende um <_SDIS></td> </tr> <tr> <td>3 = kein Rückzug am Konturende, nächster Startpunkt wird mit Konturvorschub angefahren</td> </tr> <tr> <td>TAUSENDER:</td> <td>reserviert</td> </tr> <tr> <td rowspan="3">ZEHNTAUSENDER:</td> <td>Kontur bearbeiten</td> </tr> <tr> <td>0 = Kontur vorwärts bearbeiten</td> </tr> <tr> <td>1 = Kontur rückwärts bearbeiten Einschränkungen bei rückwärts: <ul style="list-style-type: none"> max. 170 Konturelemente (inclusive Fasen oder Verrundungen) es werden nur Werte in der Ebene (X/Y) und F ausgewertet </td> </tr> </table>	Bearbeitungsart		EINER:	Bearbeitung	1 = Schruppen	2 = Schlichten	5 = Anfasen	ZEHNER:	0 = Zwischenwege mit G0	1 = Zwischenwege mit G1	HUNDERTER:	Rückzug am Konturende	0 = Rückzug am Konturende auf Bezugspunkt	1 = Rückzug am Konturende auf Bezugspunkt + <_SDIS>	2 = Rückzug am Konturende um <_SDIS>	3 = kein Rückzug am Konturende, nächster Startpunkt wird mit Konturvorschub angefahren	TAUSENDER:	reserviert	ZEHNTAUSENDER:	Kontur bearbeiten	0 = Kontur vorwärts bearbeiten	1 = Kontur rückwärts bearbeiten Einschränkungen bei rückwärts: <ul style="list-style-type: none"> max. 170 Konturelemente (inclusive Fasen oder Verrundungen) es werden nur Werte in der Ebene (X/Y) und F ausgewertet
Bearbeitungsart																										
EINER:	Bearbeitung																									
	1 = Schruppen																									
	2 = Schlichten																									
5 = Anfasen																										
ZEHNER:	0 = Zwischenwege mit G0																									
	1 = Zwischenwege mit G1																									
HUNDERTER:	Rückzug am Konturende																									
	0 = Rückzug am Konturende auf Bezugspunkt																									
	1 = Rückzug am Konturende auf Bezugspunkt + <_SDIS>																									
	2 = Rückzug am Konturende um <_SDIS>																									
3 = kein Rückzug am Konturende, nächster Startpunkt wird mit Konturvorschub angefahren																										
TAUSENDER:	reserviert																									
ZEHNTAUSENDER:	Kontur bearbeiten																									
	0 = Kontur vorwärts bearbeiten																									
	1 = Kontur rückwärts bearbeiten Einschränkungen bei rückwärts: <ul style="list-style-type: none"> max. 170 Konturelemente (inclusive Fasen oder Verrundungen) es werden nur Werte in der Ebene (X/Y) und F ausgewertet 																									
12		<_RL>	INT	<table border="1"> <tr> <td colspan="2">Bearbeitungsrichtung</td> </tr> <tr> <td>40 =</td> <td>mittig der Kontur (G40, An- und Abfahren: Gerade oder senkrecht)</td> </tr> <tr> <td>41 =</td> <td>links der Kontur (G41, An- und Abfahren: Gerade oder Kreis)</td> </tr> <tr> <td>42 =</td> <td>rechts der Kontur (G42, An- und Abfahren: Gerade oder Kreis)</td> </tr> </table>	Bearbeitungsrichtung		40 =	mittig der Kontur (G40, An- und Abfahren: Gerade oder senkrecht)	41 =	links der Kontur (G41, An- und Abfahren: Gerade oder Kreis)	42 =	rechts der Kontur (G42, An- und Abfahren: Gerade oder Kreis)														
Bearbeitungsrichtung																										
40 =	mittig der Kontur (G40, An- und Abfahren: Gerade oder senkrecht)																									
41 =	links der Kontur (G41, An- und Abfahren: Gerade oder Kreis)																									
42 =	rechts der Kontur (G42, An- und Abfahren: Gerade oder Kreis)																									

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
13		<_AS1>	INT	Kontur-Anfahrbewegung	
				EINER:	1 = Gerade
					2 = Viertelkreis
					3 = Halbkreis
					4 = senkrecht An- und Abfahren
				ZEHNER:	0 = letzte Bewegung, in der Ebene
	1 = letzte Bewegung, räumlich				
14	L1	<_LP1>	REAL	Anfahrweg, oder Anfahradius (ink, ohne Vorzeichen einzugeben)	
15	FZ	<_FF3>	REAL	Vorschub für Zwischenwege (G94/G95 wie an Kontur)	
16		<_AS2>	INT	Kontur-Abfahrbewegung (nicht bei senkrecht An-/Abfahren)	
				EINER:	1 = Gerade
					2 = Viertelkreis
					3 = Halbkreis
				ZEHNER:	0 = letzte Bewegung, in der Ebene
					1 = letzte Bewegung, räumlich
17	L2	<_LP2>	REAL	Abfahrweg, oder Abfahradius (ink, ohne Vorzeichen einzugeben)	
18		<_UMODE>	INT	reserviert	
19	FS	<_FS>	REAL	Fasenbreite (Ink)	
20	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfassen (abs/ink), siehe <_AMODE>	
21		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	Auswahl Bearbeitung/nur Startpunktbe- rechnung
				0 =	Kompatibilitätsmodus
1 =	normale Bearbeitung				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung					
22		<_DMODE>	INT	Anzeigemodus					
				EINER:	Bearbeitungsebene G17/18/19				
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv			
					1 =	G17 (nur im Zyklus aktiv)			
					2 =	G18 (nur im Zyklus aktiv)			
								3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub				
					0 =	Kompatibilitätsmodus			
								1 =	G-Code wie vor Zyklusaufwurf. G94/G95 für Flächen- und Tiefenvorschub gleich
				TAUSENDER:					
					0 =	Kompatibilitätsmode: Konturname steht in <_KNAME>			
								1 =	Konturname wird im CYCLE62 programmiert und in <_SC_CONT_NAME> übergeben
23		<_AMODE>	INT	Alternativmodus					
				EINER:	Endpunkt Z1 (<_DP>)				
					0 =	absolut (Kompatibilitätsmodus)			
								1 =	inkrementell
				ZEHNER:	Einheit für Ebenenzustellung				
					0 =	mm, inch			
								1 =	reserviert
				HUNDERTER:	Eintauchtiefe bei Anfassen (<_ZFS>)				
					0 =	absolut			
								1 =	inkrementell

Hinweis

Sind folgende Übergabeparameter indirekt (als Parameter) programmiert, wird die Eingabemaske nicht rückübersetzt:

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

19.1.17 CYCLE76 - Rechteckzapfen fräsen

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<_RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen eingeben)
4	Z1	<_DP>	REAL	Zapfentiefe(abs)
5		<_DPR>	REAL	Zapfentiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben)
6	L	<_LENG>	REAL	Zapfenlänge, siehe <_GMODE> (ohne Vorzeichen einzugeben)
7	W	<_WID>	REAL	Zapfenbreite, siehe <_GMODE> (ohne Vorzeichen einzugeben)
8	R	<_CRAD>	REAL	Eckenradius des Zapfen (ohne Vorzeichen einzugeben)
9	X0	<_PA>	REAL	Bezugspunkt Zapfen, 1. Achse der Ebene (abs)
10	Y0	<_PO>	REAL	Bezugspunkt Zapfen, 2. Achse der Ebene (abs)
11	α0	<_STA>	REAL	Drehwinkel, Winkel zwischen Längsachse (L) und 1. Achse der Ebene
12	DZ	<_MID>	REAL	maximale Tiefenzustellung (ink, ohne Vorzeichen einzugeben)
13	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene (ink), Aufmaß an der Randkontur
14	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe (ink), Aufmaß am Grund (ohne Vorzeichen eingeben)
15	FX	<_FFP1>	REAL	Vorschub an der Kontur
16	FZ	<_FFD>	REAL	Zustellvorschub Tiefe
17		<_CDIR>	INT	Fräsrichtung (ohne Vorzeichen einzugeben) EINER: 0 = Gleichlauf 1 = Gegenlauf
18		<_VARI>	INT	Bearbeitung EINER: 1 = Schruppen 2 = Schlichten 5 = Anfasen
19	L1	<_AP1>	REAL	Länge des Rohteilzapfens
20	W1	<_AP2>	REAL	Breite des Rohteilzapfens
21	FS	<_FS>	REAL	Fasenbreite (ink)
22	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs, ink), siehe <_AMODE>

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
23		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)		
				EINER:	reserviert	
				ZEHNER:	reserviert	
				HUNDERTER:	Auswahl Bearbeitung oder nur Startpunktberechnung	
					0 =	Kompatibilitätsmodus
					1 =	normale Bearbeitung
				TAUSENDER:	Bemaßung des Zapfens über Mitte oder Ecke	
					0 =	Kompatibilitätsmodus
					1 =	Bemaßung über Mitte
					2 =	Bemaßung Eckpunkt, Zapfen +L +W
					3 =	Bemaßung Eckpunkt, Zapfen -L +W
					4 =	Bemaßung Eckpunkt, Zapfen +L -W
				ZEHNTAUSENDER:	Komplettbearbeitung oder Nachbearbeitung	
					0 =	Kompatibilitätsmodus
1 =	Komplettbearbeitung					
2 =	Nachbearbeitung					
24		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/18/19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)	
				ZEHNER:	---	reserviert
				HUNDERTER:	---	reserviert
				TAUSENDER:	---	reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklusmasken (Seite 734)	
0 =	Eingabe: komplett					
1 =	Eingabe: einfach					

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
25		<_AMODE>	INT	Alternativmodus		
				EINER:	Endtiefe Z1 (DP)	
					0 =	Kompatibilität
					1 =	inkrementell
				ZEHNER:	reserviert	
					HUNDERTER:	Eintauchtiefe bei Anfassen (ZFS)
				0 =		absolut
1 =	inkrementell					

19.1.18 CYCLE77 - Kreiszapfen fräsen

Syntax

CYCLE77 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>, <_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
1	RP	<_RTP>	REAL	Rückzugsebene (abs)		
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)		
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen eingeben)		
4	Z1	<_DP>	REAL	Zapftiefe(abs)		
5		<_DPR>	REAL	Zapftiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben)		
6	Ø	<_CDIAM>	REAL	Durchmesser des Zapfens (ohne Vorzeichen einzugeben)		
7	X0	<_PA>	REAL	Bezugspunkt Zapfen, 1. Achse der Ebene (abs)		
8	Y0	<_PO>	REAL	Bezugspunkt Zapfen, 2. Achse der Ebene (abs)		
9	DZ	<_MID>	REAL	maximale Tiefenzustellung (ink, ohne Vorzeichen einzugeben)		
10	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene (ink), Aufmaß an der Randkontur		
11	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe (ink), Aufmaß am Grund (ohne Vorzeichen eingeben)		
12	FX	<_FFP1>	REAL	Vorschub an der Kontur		
13	FZ	<_FFD>	REAL	Zustellvorschub Tiefe		
14		<_CDIR>	INT	Fräsrichtung (ohne Vorzeichen einzugeben)		
				EINER:		
					0 =	Gleichlauf
1 =	Gegenlauf					

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
15		<_VARI>	INT	Bearbeitungsart	
				EINER:	Bearbeitung
				1 =	Schruppen bis auf Schlichtaufmaß
				2 =	Schichten (Aufmaß X/Y/Z = 0)
				5 =	Anfasen
16	Ø1	<_AP1>	REAL	Durchmesser des Rohteilzapfens	
17	FS	<_FS>	REAL	Fasenbreite (ink)	
18	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink) siehe <_AMODE>	
19		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	Auswahl Bearbeitung/nur Startpunktbe- rechnung
				0 =	Kompatibilitätsmodus
				1 =	normale Bearbeitung
				TAUSENDER:	reserviert
				ZEHNTAUSENDER:	Komplettbearbeitung/Nachbearbeitung
				0 =	Kompatibilitätsmodus (<_AP1> wie bisher behandeln)
				1 =	Komplettbearbeitung
				2 =	Nachbearbeitung
20		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/18/19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene ak- tiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	--- reserviert
				HUNDERTER:	--- reserviert
				TAUSENDER:	--- reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zy- klenmasken (Seite 734)
0 =	Eingabe: komplett				
1 =	Eingabe: einfach				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
21		<_AMODE>	INT	Alternativmodus		
				EINER:	Endtiefe Z1 (DP)	
					0 =	absolut (Kompatibilitätsmodus)
					1 =	inkrementell
				ZEHNER:	2 =	absolut
					reserviert	
				HUNDERTER:	Eintauchtiefe bei Anfassen (ZFS)	
0 =	absolut					
1 =	inkrementell					

19.1.19 CYCLE78 - Bohrgewinde fräsen

Syntax

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>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
1	RP	<_RTP>	REAL	Rückzugsebene (abs)	
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)	
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen eingeben)	
4	Z1	<_DP>	REAL	Endbohrtiefe (abs/ink), siehe <_AMODE>	
5		<_ADPR>	REAL	Anbohrtiefe mit reduziertem Bohrvorschub (ink), mit <_VARI> ZEHN-TAUSENDER wirksam	
6	D	<_FDPR>	REAL	maximale Tiefenzustellung (ink) D ≥ Z1 ⇒ eine Zustellung auf Endbohrtiefe D < Z1 ⇒ Tiefbohrzyklus mit mehreren Zustellungen und Entspanen	
7	ZR	<_LDPR>	REAL	Restbohrtiefe beim Durchbohren (ink), mit Vorschub FR	
8	∅	<_DIAM>	REAL	Nenndurchmesser des Gewindes	
9	P	<_PIT>	REAL	Gewindesteigung als Zahlenwert	
10		<_PITA>	INT	Bewertung der Gewindesteigung P	
				1 =	Steigung in mm/U
				2 =	Steigung in Gänge/Zoll
				3 =	Steigung in inch/U
				4 =	Steigung als MODUL

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
11	DF	<_DAM>	REAL	Betrag/Prozentsatz für jede weitere Zustellung (Degression), siehe <_AMODE>	
12	V1	<_MDEP>	REAL	minimale Zustellung (ink), nur bei Degression wirksam	
13		<_VARI>	INT	Bearbeitungsart	
				EINER:	reserviert
				ZEHNER:	Entspanen vor Gewindefräsen
				0 =	kein Entspanen vor Gewindefräsen (wirkt nur auf Endbohrtiefe)
				1 =	Entspanen vor Gewindefräsen (wirkt nur auf Endbohrtiefe)
				HUNDERTER:	Rechts-/Linksgewinde
				0 =	Rechtsgewinde
				1 =	Linksgewinde
				TAUSENDER:	Restbohrtiefe mit Bohrvorschub
				0 =	keine Restbohrtiefe mit Bohrvorschub FR
				1 =	Restbohrtiefe mit Bohrvorschub FR
				ZEHNTAUSENDER:	Anbohren mit verringertem Vorschub
				0 =	kein Anbohren mit verringertem Vorschub
1 =	Anbohren mit verringertem Vorschub Anbohrvorschub = 0,3 F1, wenn F1 < 0,15 mm/U Anbohrvorschub = 0,1 mm/U, wenn F1 ≥ 0,15 mm/U				
14		<_CDIR>	INT	Fräsrichtung	
				0 =	Gleichlauf
				1 =	Gegenlauf
				4 =	Gegenlauf + Gleichlauf (Kombination Schruppen + Schlichten)
15	Z2	<_GE>	REAL	Rückzugsbetrag vor Gewindefräsen (ink)	
16	F1	<_FFD>	REAL	Bohrvorschub (mm/min bzw. in/min oder mm/U)	
17	FR	<_FRDP>	REAL	Bohrvorschub für Restbohrtiefe (mm/min oder mm/U)	
18	F2	<_FFR>	REAL	Vorschub für Gewindefräsen (mm/min oder mm/Zahn)	
19	FS	<_FFP2>	REAL	Schlichtvorschub für <_CDIR> = 4 (mm/min oder mm/Zahn)	
20		<_FFA>	INT	Bewertung Vorschübe	
				EINER:	Bohrvorschub F1
				ZEHNER:	Bohrvorschub für Restbohrtiefe FR
				HUNDERTER:	Vorschub für Gewindefräsen F2
				TAUSENDER:	Schlichtvorschub FS
21		<_PITM>	STRING[15]	String als Merker für Gewindesteigungseingabe (nur für die Oberfläche) ¹⁾	
22		<_PTAB>	STRING[20]	String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") (nur für die Oberfläche) ¹⁾	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
23		<_PTABA>	STRING[20]	String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...) (nur für die Oberfläche) ¹⁾	
24		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte), reserviert	
25		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/18/19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
3 =	G19 (nur im Zyklus aktiv)				
26		<_AMODE>	INT	Alternativmodus	
				EINER:	Bohrtiefe = Endbohrtiefe Z1 abs/ink
				0 =	absolut
				1 =	inkrementell
				ZEHNER:	Betrag/Prozentsatz DF für jede weitere Zustellung (Degression)
0 =	Betrag				
1 =	Prozentsatz (0.001 bis 100 %)				

Hinweis

¹⁾ Die Parameter 21, 22 und 23 werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet. Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich.

19.1.20 CYCLE79 - Mehrkant

Syntax

CYCLE79 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>, <_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<_RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse (abs)

Zyklen extern programmieren

19.1 Technologische Zyklen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Mehrkanttiefe (abs/ink), siehe <_AMODE>
5	N	<_NUM>	INT	Anzahl der Kanten (1...n)
6	SW/L	<_SWL>	REAL	Schlüsselweite oder Kantenlänge (je nach <_VARI>) ("SW" bei Schlüsselweite, "L" bei Kantenlänge) Schlüsselweite nur bei gerader Anzahl Kanten, und Einkant
7	X0	<_PA>	REAL	Zapfenbezugspunkt, 1. Achse (abs)
8	Y0	<_PO>	REAL	Zapfenbezugspunkt, 2. Achse (abs)
9	α0	<_STA>	REAL	Drehwinkel Kantenmitte gegen 1. Achse (X-Achse)
10	R1/FS1	<_RC>	REAL	Eckenverrundung bei <_NUM> > 2 (Radius/Fase, siehe <_AMODE>) (ink, ohne Vorzeichen einzugeben) ("R1" bei Radius, "FS1" bei Fase)
11	∅	<_AP1>	REAL	Rohdurchmesser des Zapfens
12	DXY	<_MIDA>	REAL	maximale Zustellbreite (Einheit, siehe <_AMODE>)
13	DZ	<_MID>	REAL	maximale Tiefenzustellung
14	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene
15	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe
16	F	<_FFP1>	REAL	Bearbeitungsvorschub
17		<_CDIR>	INT	Fräsrichtung 0 = Gleichlauf 1 = Gegenlauf
18		<_VARI>	INT	Bearbeitungsart EINER: Bearbeitung 1 = Schruppen 2 = Schlichten 3 = Schlichten Rand 5 = Anfasen ZEHNER: Schlüsselweite oder Kantenlänge 0 = Schlüsselweite 1 = Kantenlänge
19	FS	<_FS>	REAL	Fasenbreite (ink)
20	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE>
21		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte) EINER: reserviert ZEHNER: reserviert HUNDERTER: Auswahl Bearbeitung oder nur Startpunktberechnung 1 = normale Bearbeitung

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung					
22		<_DMODE>	INT	Anzeigemodus					
				EINER:	Bearbeitungsebene G17/18/19				
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv			
					1 =	G17 (nur im Zyklus aktiv)			
					2 =	G18 (nur im Zyklus aktiv)			
								3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	---	reserviert			
				HUNDERTER:	---	reserviert			
				TAUSENDER:	---	reserviert			
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)				
0 =	Eingabe: komplett								
1 =	Eingabe: einfach								
23		<_AMODE>	INT	Alternativmodus					
				EINER:	Endtiefe (<_DP>)				
					0 =	absolut			
					1 =	inkrementell			
				ZEHNER:	Einheit für Ebenenzustellung (<_MIDA>)				
					0 =	mm			
					1 =	% vom Werkzeugdurchmesser			
				HUNDERTER:	Eintauchtiefe bei Anfassen (<_ZFS>)				
					0 =	absolut			
					1 =	inkrementell			
				TAUSENDER:	Eckenverrundung (<_RC>)				
					0 =	Radius			
					1 =	Fase			

19.1.21 CYCLE81 - Bohren, Zentrieren

Syntax

CYCLE81 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<RTP>	REAL	Rückzugsebene (abs)
2	Z0	<RFP>	REAL	Bezugspunkt (abs)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugabe)	
4	Z1/Ø	<DP>	REAL	Bohrtiefe (abs) / Durchmesser der Zentrierung (abs), siehe <_GMODE>	
5	Z1	<DPR>	REAL	Bohrtiefe (ink)	
6	DT	<DTB>	REAL	Verweilzeit auf Endbohrtiefe, siehe <_AMODE>	
7		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	Zentrierung bezogen auf die Tiefe / den Durchmesser
				0 =	Kompatibilität, Tiefe
1 =	Durchmesser				
8		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
3 =	G19 (nur im Zyklus aktiv)				
9		<_AMODE>	INT	Alternativmodus	
				EINER:	Bohrtiefe Z1 (abs/ink)
				0 =	Kompatibilität, aus Programmierung DP/DPR
				1 =	inkrementell
				2 =	absolut
				ZEHNER:	Verweilzeit auf Endbohrtiefe DT in Sekunden/Umdrehungen
				0 =	Kompatibilität, aus Vorzeichen von DTB (> 0 Sekunden oder < 0 Umdrehungen)
1 =	in Sekunden				
2 =	in Umdrehungen				

19.1.22 CYCLE82 - Bohren, Plansenken

Syntax

CYCLE82 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>, <_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
1	RP	<RTP>	REAL	Rückzugsebene (abs)	
2	Z0	<RFP>	REAL	Bezugspunkt (abs)	
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)	
4	Z1	<DP>	REAL	Bohrtiefe (abs), siehe <_AMODE>	
5	Z1	<DPR>	REAL	Bohrtiefe (ink), siehe <_AMODE>	
6	DT	<DTB>	REAL	Verweilzeit auf Endbohrtiefe, siehe <_AMODE>	
7		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	Bohrtiefe bezogen auf Spitze/Schaft
				0 =	Kompatibilität, Spitze
				1 =	Schaft
8		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	reserviert
				HUNDERTER:	reserviert
				TAUSENDER:	reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)
				0 =	Eingabe: komplett
				1 =	Eingabe: einfach

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
9		<_AMODE>	INT	Alternativmodus	
				EINER:	Bohrtiefe Z1 (abs/ink)
				0 =	Kompatibilität aus Programmierung DP/DPR
				1 =	inkrementell
				2 =	absolut
				ZEHNER:	Verweilzeit DT auf Endbohrtiefe in Sekunden/Umdrehungen
				0 =	Kompatibilität, aus Vorzeichen von DT (> 0 Sekunden / < 0 Umdrehungen)
				1 =	in Sekunden
				2 =	in Umdrehungen
				HUNDERTER:	Anbohrtiefe ZA abs/ink
				0 =	inkrementell
				1 =	absolut
				TAUSENDER:	Bewertung Anbohrvorschub
				0 =	in % vom Bohrvorschub
				1 =	F/min
				2 =	F/U
				ZEHNTAUSENDER:	Restbohrtiefe ZD abs/ink
				0 =	inkrementell
				1 =	absolut
				HUNDERTTAUSENDER:	Bewertung Restbohrvorschub
				0 =	in % vom Bohrvorschub
1 =	F/min				
2 =	F/U				
10		<_VARI>	INT	Bearbeitungsart Anbohren/Durchbohren	
				EINER:	reserviert
				ZEHNER:	reserviert
				HUNDERTER:	reserviert
				TAUSENDER:	Durchbohren
				0 =	Durchbohren "Nein"
				1 =	Durchbohren "Ja"
				ZEHNTAUSENDER:	Anbohren
				0 =	Anbohren "Nein"
1 =	Anbohren "Ja"				
11	ZA	<S_ZA>	REAL	Anbohrtiefe inkrementell bezogen auf Bezugspunkt oder absolut (siehe <_AMODE> HUNDERTER)	
12	FA	<S_FA>	REAL	Anbohrvorschub als Wert oder in % (in Verbindung mit <_AMODE> TAUSENDER)	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
13	ZD	<S_ZD>	REAL	Restbohrtiefe inkrementell bezogen auf Endbohrtiefe oder absolut (in Verbindung mit <_AMODE> ZEHNTAUSENDER)
14	FD	<S_FD>	REAL	Restbohrvorschub als Wert oder in % (in Verbindung mit <_AMODE> HUNDERTTAUSENDER)

19.1.23 CYCLE83 - Tieflochbohren

Syntax

CYCLE83 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>, <DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>, <_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung										
1	RP	<RTP>	REAL	Rückzugsebene (abs)										
2	Z0	<RFP>	REAL	Bezugspunkt (abs)										
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen eingeben)										
4	Z1	<DP>	REAL	Endbohrtiefe (abs), siehe <_AMODE>										
5	Z1	<DPR>	REAL	Endbohrtiefe (ink), siehe <_AMODE>										
6	D	<FDEP>	REAL	1. Bohrtiefe (abs), siehe <_AMODE>										
7	D	<FDPR>	REAL	1. Bohrtiefe (ink), siehe <_AMODE>										
8	DF	<_DAM>	REAL	Degressionsbetrag/Prozentsatz für jede weitere Zustellung, siehe <_AMODE>										
9	DTB	<DTB>	REAL	Verweilzeit auf Bohrtiefe, siehe <_AMODE>										
10	DTS	<DTS>	REAL	Verweilzeit am Anfangspunkt (nur bei Entspanen), siehe <_AMODE>										
11	FD1	<FRF>	REAL	Prozentsatz für den Vorschub bei der ersten Zustellung, siehe <_AMODE>										
12		<VARI>	INT	<table border="1"> <tr> <td colspan="2">Bearbeitungsart</td> </tr> <tr> <td>EINER:</td> <td>Spänebrechen / Entspanen</td> </tr> <tr> <td>0 =</td> <td>Spänebrechen</td> </tr> <tr> <td>1 =</td> <td>Entspanen</td> </tr> </table>	Bearbeitungsart		EINER:	Spänebrechen / Entspanen	0 =	Spänebrechen	1 =	Entspanen		
Bearbeitungsart														
EINER:	Spänebrechen / Entspanen													
0 =	Spänebrechen													
1 =	Entspanen													
13		<_AXN>	INT	<table border="1"> <tr> <td colspan="2">Werkzeugachse</td> </tr> <tr> <td>0 =</td> <td>3. Geometrieachse</td> </tr> <tr> <td>1 =</td> <td>1. Geometrieachse</td> </tr> <tr> <td>2 =</td> <td>2. Geometrieachse</td> </tr> <tr> <td>> 2</td> <td>3. Geometrieachse</td> </tr> </table>	Werkzeugachse		0 =	3. Geometrieachse	1 =	1. Geometrieachse	2 =	2. Geometrieachse	> 2	3. Geometrieachse
Werkzeugachse														
0 =	3. Geometrieachse													
1 =	1. Geometrieachse													
2 =	2. Geometrieachse													
> 2	3. Geometrieachse													
14	V1	<_MDEP>	REAL	minimale Zustellung (nur bei Prozentsatz für Degression)										

Zyklen extern programmieren

19.1 Technologische Zyklen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
15	V2	<_VRT>	REAL	Rückzugsbetrag nach jeder Bearbeitung (nur bei Spänebrechen)	
				> 0	variabler Rückzugsbetrag
				0 =	Standardwert 1 mm
16	DT	<_DTD>	REAL	Verweilzeit auf Endbohrtiefe, siehe <_AMODE>	
17	V3	<_DIS1>	REAL	Vorhalteabstand (nur bei Entspanen), siehe <_AMODE>	
18		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	Bohrtiefe bezogen auf Spitze/Schaft
				0 =	Spitze
				1 =	Schaft
19		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	--- reserviert
				HUNDERTER:	--- reserviert
				TAUSENDER:	--- reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)
				0 =	Eingabe: komplett
1 =	Eingabe: einfach				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
20		<_AMODE>	INT	Alternativmodus	
				EINER:	Bohrtiefe = Endbohrtiefe Z1 (abs/ink)
				0 =	Kompatibilität, aus Programmierung <DP>/<DPR>
				1 =	inkrementell
				2 =	absolut
				ZEHNER:	Verweilzeit auf Bohrtiefe DTB in Sekunden/Umdrehungen
				0 =	Kompatibilität aus Vorzeichen von DTB (> 0 Sekunden oder < 0 Umdrehungen)
				1 =	in Sekunden
				2 =	in Umdrehungen
				HUNDERTER:	Verweilzeit am Anfangspunkt von DTS in Sekunden/Umdrehungen
				0 =	Kompatibilität aus Vorzeichen von DTS (> 0 Sekunden oder < 0 Umdrehungen)
				1 =	in Sekunden
				2 =	in Umdrehungen
				TAUSENDER:	Verweilzeit auf Endbohrtiefe DTD in Sekunden/Umdrehungen
				0 =	Kompatibilität aus Vorzeichen von DTD (> 0 Sekunden oder < 0 Umdrehungen)
				1 =	in Sekunden
				2 =	in Umdrehungen
				ZEHNTAUSENDER:	1. Bohrtiefe D (abs/ink)
				0 =	Kompatibilität, aus Programmierung <FDEPF>/<DPR>
				1 =	inkrementell
				2 =	absolut
				HUNDERTTAUSENDER:	Degressionsbetrag/Prozentsatz <_DAM> für jede weitere Zustellung
				0 =	Kompatibilität, aus Vorzeichen von <_DAM> (> 0 Degressionsbetrag oder < 0 Faktor 0,001 bis 1,0)
				1 =	Degressionsbetrag
2 =	Prozentsatz (0,001 bis 100 %)				
EINEMILLION:	Vorhalteabstand V3 automatisch/manuell				
0 =	Kompatibilität aus Vorzeichen von <_DIS1> (= 0 automatisch oder > 0 manuell)				
1 =	automatisch (wird im Zyklus berechnet)				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
				2 = manuell (programmierter Wert)
				ZEHNMILLIONEN: Vorschubfaktor für erste Zustellung <FRF> als Faktor/Prozentsatz
				0 = Kompatibilität, als Faktor (0,001 bis 1,0, FRF = 0 bedeutet 100 %)
				1 = Prozentsatz (0,001 bis 999,999 %)

19.1.24 CYCLE84 - Gewindebohren ohne Ausgleichsfutter

Syntax

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>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<RTP>	REAL	Rückzugsebene (abs)
2	Z0	<RFP>	REAL	Bezugspunkt (abs)
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<DP>	REAL	Bohrtiefe = Endbohrtiefe (abs), siehe <_AMODE>
5	Z1	<DPR>	REAL	Bohrtiefe = Endbohrtiefe (ink), siehe <_AMODE>
6	DT	<DTB>	REAL	Verweilzeit auf Bohrtiefe in Sekunden
7	SDE	<SDAC>	INT	Drehrichtung nach Zyklusende
8		<MPIT>	REAL	Gewindegröße nur für "ISO metrisch" (Steigung wird intern während der Laufzeit berechnet)
9	P	<PIT>	REAL	Gewindesteigung als Wert, Maßeinheit siehe <_PITA>
10	αS^1)	<POSS>	REAL	Spindelposition für orientierten Spindelstop
11	S	<SST>	REAL	Spindeldrehzahl für Gewindebohren
12	SR	<SST1>	REAL	Spindeldrehzahl für Rückzug
13		<_AXN>	INT	Bohrachse
				0 = 3. Geometrieachse
				1 = 1. Geometrieachse
				2 = 2. Geometrieachse
				≥ 3 = 3. Geometrieachse

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
14		<_PITA>	INT	Maßeinheit der Gewindesteigung (Auswertung <PIT> und <MPIT>)		
				0 =	Steigung in mm	- Auswertung <MPIT>/<PIT>
				1 =	Steigung in mm	- Auswertung <PIT>
				2 =	Steigung in TPI	- Auswertung <PIT> (Gewindegänge pro inch)
				3 =	Steigung in inch	- Auswertung <PIT>
				4 =	MODUL	- Auswertung <PIT>
15		<_TECHNO>	INT	Technologie ¹⁾		
				EINER:	Genauhaltverhalten	
				0 =	Genauhaltverhalten wie vor Zyklusauf- ruf aktiv	
				1 =	Genauhalt G601	
				2 =	Genauhalt G602	
				3 =	Genauhalt G603	
				ZEHNER:	Vorsteuerung	
				0 =	mit/ohne Vorsteuerung wie vor Zyklusauf- ruf aktiv	
				1 =	mit Vorsteuerung FFWON	
				2 =	ohne Vorsteuerung FFWOF	
				HUNDERTER:	Beschleunigung	
				0 =	SOFT/BRISK/DRIVE wie vor Zyklusauf- ruf aktiv	
				1 =	mit Ruckbegrenzung SOFT	
				2 =	ohne Ruckbegrenzung BRISK	
				3 =	reduzierte Beschleunigung DRIVE	
				TAUSENDER:	MCALL Spindelbetrieb	
0 =	bei MCALL Spindelbetrieb wieder aktivie- ren					
1 =	bei MCALL in Lageregelung bleiben					
16		<_VARI>	INT	Bearbeitungsart		
				EINER:	0 =	1 Schnitt
				1 =	Spänebrechen (Tieflochgewindebohren)	
				2 =	Entspanen (Tieflochgewindebohren)	
				TAUSENDER:	ISO/SIEMENS Modus für Eingabemaske nicht relevant	
				0 =	Aufruf aus ISO-Kompatibilität	
1 =	Aufruf aus SIEMENS-Kontext					
17	D	<_DAM>	REAL	maximale Tiefenzustellung (nur bei Entspanen/Spänebrechen)		

Zyklen extern programmieren

19.1 Technologische Zyklen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
18	V2	<_VRT>	REAL	Rückzugsbetrag nach jeder Bearbeitung (nur bei Spänebrechen), siehe <_AMODE>	
19		<_PTM>	STRING[15]	String als Merker für Gewindesteigungseingabe ²⁾	
20		<_PTAB>	STRING[5]	String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") ²⁾	
21		<_PTABA>	STRING[20]	String für Auswahl in der Gewindetabelle (z. B. "M 10", "M 12", ...) ²⁾	
22		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	reserviert
23		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	reserviert
				HUNDETER:	reserviert
				TAUSENDER:	Kompatibilitätsmode (nur für Eingabemaske zu Rückübersetzen), wenn MD 52216 Bit0 = 1 ¹⁾
				0 =	Technologieparameter werden angezeigt (Kompatibilität): TECHNO-Parameter wirken
				1 =	Technologieparameter werden nicht angezeigt: Technologie "wie vor Zyklusaufruf programmiert" wirkt
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklensmasken (Seite 734)
				0 =	Eingabe: komplett
1 =	Eingabe: einfach				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
24		<_AMODE>	INT	Alternativmodus	
				EINER:	Bohrtiefe = Endbohrtiefe Z1 (abs/ink)
				0 =	Kompatibilität, aus Programmierung <DP>/<DPR>
				1 =	inkrementell
				2 =	absolut
				ZEHNER:	reserviert
				HUNDERTER:	reserviert
				TAUSENDER:	Gewindedrehrichtung rechts/links
				0 =	Kompatibilität, aus Vorzeichen PIT/MPIT
				1 =	rechts
				2 =	links
				ZEHNTAUSENDER:	reserviert
				HUNDERTTAUSENDER:	reserviert
				EINEMILLION:	Rückzugsbetrag nach jeder Bearbeitung V2 manuell/automatisch
0 =	Kompatibilität, aus Programmierung <_VRT> (> 0 variabler Wert oder ≤ 0 Standardwert 1 mm/0,0394 inch)				
1 =	automatisch (Standardwert 1 mm/0,0394 inch)				
2 =	manuell (wie unter V2 programmiert)				
<p>¹⁾ Felder Technologie können ausgeblendet sein in Abhängigkeit von Settingdatum SD52216 \$MCS_FUNCTION_MASK_DRILL</p> <p>²⁾ Die Parameter 19, 20, und 21 werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet. Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich.</p>					

19.1.25 CYCLE85 - Reiben

Syntax

CYCLE85 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>, <_GMODE>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<RTP>	REAL	Rückzugsebene (abs)
2	Z0	<RFP>	REAL	Bezugspunkt (abs)
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
4	Z1	<DP>	REAL	Bohrtiefe (abs), siehe <_AMODE>	
5	Z1	<DPR>	REAL	Bohrtiefe (ink), siehe <_AMODE>	
6	DT	<DTB>	REAL	Verweilzeit auf Endbohrtiefe, siehe <_AMODE>	
7	F	<FFR>	REAL	Vorschub	
8	FR	<RFF>	REAL	Vorschub beim Rückzug	
9		<_GMODE>	INT	reserviert	
10		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
3 =	G19 (nur im Zyklus aktiv)				
11		<_AMODE>	INT	Alternativmodus (Bohren)	
				EINER:	Bohrtiefe Z1 (abs/ink)
				0 =	Kompatibilität, aus Programmierung DP/DPR
				1 =	inkrementell
				2 =	absolut
				ZEHNER:	Verweilzeit DT auf Endbohrtiefe in Sekunden/Umdrehungen
				0 =	Kompatibilität, aus Vorzeichen von DT (> 0 Sekunden oder < 0 Umdrehungen)
1 =	in Sekunden				
2 =	in Umdrehungen				

19.1.26 CYCLE86 - Ausdrehen

Syntax

CYCLE86 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>, <RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<RTP>	REAL	Rückzugsebene (abs)
2	Z0	<RFP>	REAL	Bezugspunkt (abs)
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
4	Z1	<DP>	REAL	Bohrtiefe (abs), siehe <_AMODE>
5	Z1	<DPR>	REAL	Bohrtiefe (ink), siehe <_AMODE>
6	DT	<DTB>	REAL	Verweilzeit auf Endbohrtiefe, siehe <_AMODE>
7	DIR	<SDIR>	INT	Spindeldrehrichtung 3 = M3 4 = M4
8	DX	<RPA>	REAL	Abhebebetrag in X-Richtung
9	DY	<RPO>	REAL	Abhebebetrag in Y-Richtung
10	DZ	<RPAP>	REAL	Abhebebetrag in Z-Richtung
11	SPOS	<POSS>	REAL	Spindelposition zum Abheben (für orientierten Spindelhalt, in Grad)
12		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte) EINER: Abhebemodus 0 = abheben, Kompatibilität 1 = nicht abheben
13		<_DMODE>	INT	Anzeigemodus EINER: Bearbeitungsebene G17/G18/G19 0 = Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv 1 = G17 (nur im Zyklus aktiv) 2 = G18 (nur im Zyklus aktiv) 3 = G19 (nur im Zyklus aktiv)
14		<_AMODE>	INT	Alternativmodus EINER: Bohrtiefe Z1 (abs/ink) 0 = Kompatibilität, aus Programmierung <DP>/<DPR> 1 = inkrementell 2 = absolut ZEHNER: Verweilzeit auf Endbohrtiefe DT in Sekunden/Umdrehungen 0 = Kompatibilität, aus Vorzeichen von DT (> 0 Sekunden oder < 0 Umdrehungen) 1 = in Sekunden 2 = in Umdrehungen

19.1.27 CYCLE92 - Abstich

Syntax

CYCLE92 (<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>, <_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	X0	<_SPD>	REAL	Bezugspunkt (abs, immer Durchmesser)
2	Y0	<_SPL>	REAL	Bezugspunkt (abs)
3	X1	<_DIAG1>	REAL	Tiefe für Drehzahlreduzierung, siehe <_AMODE> (EINER)
4	X2	<_DIAG2>	REAL	Endtiefe, siehe <_AMODE> (ZEHNER)
5	R/FS	<_RC>	REAL	Verrundusradius oder Fasebreite, siehe <_AMODE> (TAUSENDER)
6	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
7	S	<_SV1>	REAL	konstante Spindeldrehzahl, siehe <_AMODE> (ZEHNTAUSENDER)
	V			konstante Schnittgeschwindigkeit
8	SV	<_SV2>	REAL	Maximaldrehzahl bei konstanter Schnittgeschwindigkeit
9	DIR	<_SDAC>	INT	Spindeldrehrichtung
				3 = für M3
				4 = für M4
10	F	<_FF1>	REAL	Vorschub bis Tiefe für Drehzahlreduzierung
11	FR	<_FF2>	REAL	reduzierter Vorschub bis Endtiefe
12	SR	<_SS2>	REAL	reduzierte Drehzahl bis Endtiefe
13	XM	<_DIAGM>	REAL	Tiefe Teilfänger ausfahren (abs, immer Durchmesser)
14		<_VARI>	INT	Bearbeitungsart
				EINER:
				Rückzug
				0 = Rückzug auf <_SPD> + <_SDIS>
				1 = kein Rückzug am Ende
				ZEHNER:
				Teilfänger
				0 = Nein, kein M-Befehl ausführen
				1 = Ja, Aufruf von CUST_TECHCYC(101)-Schublade ausfahren, CUST_TECHCYC(102)-Schublade schließen
15		<_DN>	INT	D-Nummer für 2. Schneide, wenn nicht programmiert ⇒ D+1
20		<_DMODE>	INT	Anzeigemodus
				EINER:
				Bearbeitungsebene G17/G18/G19
				0 = Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 = G17 (nur im Zyklus aktiv)
				2 = G18 (nur im Zyklus aktiv)
				3 = G19 (nur im Zyklus aktiv)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
21		<_AMODE>	INT	Alternativmodus	
				EINER:	Tiefe für Drehzahlreduzierung (<_DIAG1>)
					0 = absolut, Wert der Planachse im Durchmesser
				1 = inkrementell, Wert der Planachse im Radius	
				ZEHNER:	Endtiefe (<_DIAG2>)
					0 = absolut, Wert der Planachse im Durchmesser
				1 = inkrementell, Wert der Planachse im Radius	
				HUNDERTER:	reserviert
				TAUSENDER:	Radius/Fase (<_RC>)
					0 = Radius
				1 = Fase	
				ZEHNTAUSENDER:	Spindeldrehzahl/Schnittgeschwindigkeit (<_SV1>)
0 = konstante Spindeldrehzahl					
1 = konstante Schnittgeschwindigkeit					

19.1.28 CYCLE95 - Konturabspanen

Syntax

CYCLE95 (<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>, <_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	CON	<NPP>	STRING [140]	Konturname
2	D	<MID>	REAL	maximale Tiefenzustellung beim Schruppen, siehe <_GMODE>
3	UZ	<FALZ>	REAL	Schlichtaufmaß in Z
4	UX	<FALX>	REAL	Schlichtaufmaß in X
5	U	<FAL>	REAL	Schlichtaufmaß konturparallel (wirkt in beiden Achsen)
6	F	<FF1>	REAL	Vorschub für Schruppen
7	FY	<FF2>	REAL	Eintauchvorschub Hinterschnitte
8	FS	<FF3>	REAL	Schlichtvorschub

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung								
9		<_VARI>	INT	Bearbeitungsart								
				EINER und ZEHNER:								
				1 =	Schruppen, längs, Außen							
				2 =	Schruppen, plan, Außen							
				3 =	Schruppen, längs, Innen							
				4 =	Schruppen, plan, Innen							
				5 =	Schlichten, längs, Außen							
				6 =	Schlichten, plan, Außen							
				7 =	Schlichten, längs, Innen							
				8 =	Schlichten, plan, Innen							
				9 =	Komplettbearbeitung, längs, Außen							
				10 =	Komplettbearbeitung, plan, Außen							
				11 =	Komplettbearbeitung, längs, Innen							
				12 =	Komplettbearbeitung, plan, Innen							
10	DT	<DT>	REAL	Verweilzeit bei Vorschubunterbrechung								
				11	DI	<DAM>	REAL	Abstand Vorschubunterbrechungen				
								12	VRT	<_VRT>	REAL	Abhebeweg von der Kontur
												0 =
> 0 =	Abhebeweg											
13		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)								
				EINER:	Bewertung der Zustelltiefe							
				0 =	Zustelltiefe wird entsprechend G-Gruppe DIAMON/DIAMOF verrechnet							
				1 =	Zustelltiefe wirkt als Radiuswert (unabhängig vom DIAMON/DIAMOF)							

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
14		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/G18/G19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)	
				TAUSENDER:		
					0 =	Kompatibilitätsmodus: Konturname steht in NPP
1 =	Konturname wird im CYCLE62 programmiert und in _SC_CONT_NAME übergeben					

19.1.29 CYCLE98 - Gewindegänge

Syntax

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>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	Z0	<_PO1>	REAL	Bezugspunkt in Z (abs)
2	X0	<_DM1>	REAL	Bezugspunkt in X (abs), im Durchmesser
3	Z1	<_PO2>	REAL	Zwischenpunkt 1 in Z, (abs/ink), siehe <_AMODE> (EINER)
4	X1	<_DM2>	REAL	Zwischenpunkt 1 in X (abs/ink), siehe <_AMODE> (ZEHNER) oder
	X1α			Gewindegänge 1 (-90° bis 90°) abs immer Durchmesser, ink immer Radius
5	Z2	<_PO3>	REAL	Zwischenpunkt 2 in Z, (abs/ink), siehe <_AMODE> (HUNDERTER)
6	X2	<_DM3>	REAL	Zwischenpunkt 2 in X (abs/ink), siehe <_AMODE> (TAUSENDER) oder
	X2α			Gewindegänge 2 (-90° bis 90°) abs immer Durchmesser, ink immer Radius
7	Z3	<_PO4>	REAL	Endpunkt in Z (abs/ink), siehe <_AMODE> (ZEHNTAUSENDER)
8	X3	<_DM4>	REAL	Endpunkt in X, (abs/ink), siehe <_AMODE> (HUNDERTTAUSENDER) oder
	X3α			Gewindegänge 3 (-90° bis 90°) abs immer Durchmesser, ink immer Radius

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
9	LW	<APP>	REAL	Gewindevorlauf (ink, ohne Vorzeichen einzugeben)	
10	LR	<ROP>	REAL	Gewindeauslauf (ink, ohne Vorzeichen einzugeben)	
11	H1	<TDEP>	REAL	Gewindetiefe (ink, ohne Vorzeichen einzugeben)	
12	U	<FAL>	REAL	Schlichtaufmaß in X und Z	
13	DP	<_IANG>	REAL	Zustellschräge als Abstand oder Winkel, siehe <_AMODE> (EINEMILLION)	
	αP			Die Zustellschräge wirkt entsprechend der Einstellung Parameter <_VARI> (HUNDERTER).	
				Definition für <_VARI_HUNDERTER = 0 - Kompatibilitätsmode:	
				> 0 =	Flankenzustellung an einer Flanke
				0 =	Zustellung senkrecht im Gewinde
				< 0 =	Flankenzustellung mit alternierenden Flanken
				Definition für <_VARI_HUNDERTER<>0:	
> 0 =	Zustellung an der positiven Flanke				
0 =	mittige Zustellung				
< 0 =	Zustellung an der negativen Flanke				
14	α0	<NSP>	REAL	Startwinkelversatz für den 1. Gewindegang	
15		<NRC>	INT	Anzahl der Schruppschnitte, siehe <_VARI> (ZEHNTAUSENDER)	
16	NN	<NID>	INT	Anzahl Leerschnitte	
17	P0	<_PP1>	REAL	Gewindesteigung 1. Gewindeabschnitt, siehe <_PITA>	
18	P1	<_PP2>	REAL	Gewindesteigung 2. Gewindeabschnitt, siehe <_PITA>	
19	P2	<_PP3>	REAL	Gewindesteigung 3. Gewindeabschnitt, siehe <_PITA>	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
20		<_VARI>	INT	Bearbeitung		
				EINER:	Technologie	
					1 =	Außengewinde mit linearer Zustellung
					2 =	Innengewinde mit linearer Zustellung
					3 =	Außengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant
				4 =	Innengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant	
				ZEHNER:	reserviert	
				HUNDERTER:	Zustellungsart	
					0 =	Kompatibilitätsmode für <_IANG>
					1 =	einseitige Zustellung
				2 =	wechselnde Zustellung	
				TAUSENDER:	reserviert	
				ZEHNTAUSENDER:	Alternative Tiefenzustellung	
					0 =	Kompatibilität, Vorgabe Anzahl der Schruppschnitte (<_NRC>)
					1 =	Vorgabe Wert für 1. Zustellung (<_MID>)
				HUNDERTTAUSENDER:	Bearbeitungsart	
					0 =	Kompatibilität (Schruppen und Schichten)
1 =	Schruppen					
2 =	Schichten					
3 =	Schruppen und Schichten					
EINEMILLION:	Bearbeitungsreihenfolge bei mehrgängigen Gewinde					
	0 =	Reihenfolge der Gänge aufsteigend				
	1 =	Reihenfolge der Gänge gegenüber				
21	N	<_NUMTH>	INT	Anzahl Gewindegänge		
22		<_VRT>	REAL	Rücklaufabstand (ink)		
				0 =	intern wird ein Abhebeweg von 1 mm unabhängig vom Maßsystem inch/metrisch verwendet	
				> 0 =	Abhebeweg	
23	D1	<_MID>	REAL	erste Zustellung, siehe <_VARI> (ZEHNTAUSENDER)		

Zyklen extern programmieren

19.1 Technologische Zyklen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
24	DA	<_GDEP>	REAL	Gangwechselliefe (wirkt nur bei "mehrgängig")	
				0 =	keine Gangwechselliefe berücksichtigen
				> 0 =	Gangwechselliefe berücksichtigen
25		<_IFLANK>	REAL	Zustellschräge als Breite (nur für die Oberfläche)	
26		<_PITA>	INT	Bewertung der Gewindesteigung	
				0 =	Kompatibilitätsmode für Gewindesteigung, Auswertung <_PP1> bis <_PP3> wie bisher entsprechend aktivem System metrisch/ inch
				1 =	Steigung in mm
				2 =	Steigung in TPI (Gewindgänge pro inch)
				3 =	Steigung in inch
4 =	MODUL				
27		<_PITM1>	STRING[15]	String als Merker für Gewindesteigungseingabe (nur für die Oberfläche)	
28		<_PITM2>	STRING[15]	String als Merker für Gewindesteigungseingabe (nur für die Oberfläche)	
29		<_PITM3>	STRING[15]	String als Merker für Gewindesteigungseingabe (nur für die Oberfläche)	
30		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	--- reserviert
				HUNDERTER:	--- reserviert
				TAUSENDER:	--- reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)
0 =	Eingabe: komplett				
1 =	Eingabe: einfach				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
31		<_AMODE>	INT	Alternativmodus	
				EINER:	1. Zwischenpunkt in Z (Z1)
				0 =	absolut
				1 =	inkrementell
				ZEHNER:	1. Zwischenpunkt in X (X1)
				0 =	absolut
				1 =	inkrementell
				2 =	α
				HUNDERTER:	2. Zwischenpunkt in Z (Z2)
				0 =	absolut
				1 =	inkrementell
				TAUSENDER:	2. Zwischenpunkt in X (X2)
				0 =	absolut
				1 =	inkrementell
				2 =	α
				ZEHNTAUSENDER:	Endpunkt in Z (Z3)
				0 =	absolut
				1 =	inkrementell
				HUNDERTTAUSENDER:	Endpunkt in X (X3)
				0 =	absolut
				1 =	inkrementell
				2 =	α
				EINEMILLION:	Auswahl Zustellschräge als Winkel oder Breite
				0 =	Zustellwinkel <_IANG>
1 =	Zustellschräge <_IFLANK>				
ZEHNMILLIONEN:	eingängig/mehrgängig				
0 =	Kompatibilitätsmodus (Startwinkel <_NSP> wird ausgewertet)				
1 =	eingängig (mit Startwinkelversatz <_NSP>)				
2 =	mehrgängig				

19.1.30 CYCLE99 - Gewindedrehen

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
1	Z0	<_SPL>	REAL	Bezugspunkt (abs)	
2	X0	<_SPD>	REAL	Bezugspunkt (abs, immer Durchmesser)	
3	Z1	<_FPL>	REAL	Endpunkt in Verbindung mit <_AMODE> (EINER)	
4	X1	<_FPD>	REAL	Endpunkt, in Verbindung mit <_AMODE> (ZEHNER)	
5	LW/LW2	<_APP>	REAL	Gewindevorlauf in Verbindung mit <_AMODE> (HUNDERTER) oder Gewindeeinlauf = Gewindeauslauf in Verbindung mit <_AMODE> (HUNDERTER)	
6	LR	<_ROP>	REAL	Gewindeauslauf	
7	H1	<_TDEP>	REAL	Gewindetiefe	
8	U	<_FAL>	REAL	Schlichtaufmaß in X und Z	
9	DP αP	<_IANG>	REAL	Zustellschräge als Abstand oder Winkel, in Verbindung mit <_AMODE> (TAUSENDER)	
				> 0 =	Zustellung an der positiven Flanke
				< 0 =	Zustellung an der negativen Flanke
				0 =	mittige Zustellung
10	α0	<_NSP>	REAL	Startwinkelversatz (wirkt nur bei "eingängig")	
11	ND	<_NRC>	INT	Anzahl Schruppschnitte, in Verbindung mit <_VARI> (ZEHNTAUSENDER)	
12	NN	<_NID>	INT	Anzahl Leerschnitte	
13	P	<_PIT>	REAL	Gewindesteigung als Wert, in Verbindung mit <_PITA>	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
14		<_VARI>	INT	Bearbeitungsart		
				EINER:	Technologie	
					1 =	Außengewinde mit linearer Zustellung
					2 =	Innengewinde mit linearer Zustellung
					3 =	Außengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant
				4 =	Innengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant	
				ZEHNER:	reserviert	
				HUNDERTER:	Zustellungsart	
					1 =	einseitige Zustellung
					2 =	wechselnde Zustellung
				TAUSENDER:	reserviert	
				ZEHNTAUSENDER:	Alternative Tiefenzustellung	
					0 =	Vorgabe Anzahl der Schruppschnitte (<_NRC>)
					1 =	Vorgabe Wert für 1. Zustellung (<_MID>)
HUNDERTTAUSENDER:	Bearbeitungsart					
	1 =	Schruppen				
	2 =	Schichten				
	3 =	Schruppen und Schichten				
EINEMILLION:	Bearbeitungsreihenfolge bei mehrgängigen Gewinde					
	0 =	Reihenfolge der Gänge aufsteigend				
	1 =	Reihenfolge der Gänge gegenüber				
15	N	<_NUMTH>	INT	Anzahl Gewindegänge		
16	VR	<_SDIS>	REAL	Rücklaufabstand, ink		
17	D1	<_MID>	REAL	Erste Zustelltiefe, in Verbindung mit <_VARI> (ZEHNTAUSENDER)		
18	DA	<_GDEP>	REAL	Gangwechselliefe (wirkt nur bei "mehrgängig")		
				0 =	keine Gangwechselliefe berücksichtigen	
				> 0 =	Gangwechselliefe berücksichtigen	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
19	G	<_PIT1>	REAL	Änderung der Steigung pro Umdrehung		
				0 =	Gewindesteigung ist konstant (G33)	
				> 0 =	Gewindesteigung wird größer (G34)	
				< 0 =	Gewindesteigung wird kleiner (G35)	
20		<_FDEP>	REAL	Eintauchtiefe (ohne Vorzeichen einzugeben)		
21	N1	<_GST>	INT	Startgang N1 = 1...N, in Verbindung mit <_AMODE> (HUNDERTTAU-SENDER)		
22		<_GUD>	INT	reserviert		
23		<_IFLANK>	REAL	Zustellschräge als Breite (nur für die Oberfläche)		
24		<_PITA>	INT	Maßeinheit der Gewindesteigung (Auswertung PIT und/oder MPIT)		
				0 =	Steigung in mm - Auswertung MPIT/PIT	
				1 =	Steigung in mm - Auswertung PIT	
				2 =	Steigung in TPI - Auswertung PIT (Gewindegänge pro inch)	
				3 =	Steigung in inch - Auswertung PIT	
				4 =	MODUL - Auswertung PIT	
25		<_PITM>	STRING[15]	String als Merker für Gewindesteigungseingabe (nur für die Oberfläche) ¹⁾		
26		<_PTAB>	STRING[20]	String für Gewindetabelle (nur für die Oberfläche) ¹⁾		
27		<_PTABA>	STRING[20]	String für Auswahl in der Gewindetabelle (nur für die Oberfläche) ¹⁾		
28		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/G18/G19	
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv	
				1 =	G17 (nur im Zyklus aktiv)	
				2 =	G18 (nur im Zyklus aktiv)	
				3 =	G19 (nur im Zyklus aktiv)	
				ZEHNER:	Gewindeart	
				0 =	Längsgewinde	
				1 =	Plangewinde	
				2 =	Kegelgewinde	
				HUNDERTER:	---	reserviert
				TAUSENDER:	---	reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)	
				0 =	Eingabe: komplett	
1 =	Eingabe: einfach					

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung			
29		<_AMODE>	INT	Alternativmodus			
				EINER:	Gewindelänge in Z		
					0 =	absolut	
					1 =	inkrementell	
				ZEHNER:	Gewindelänge in X		
					0 =	absolut, Wert der Planachse im Durchmesser	
					1 =	inkrementell, Wert der Planachse im Radius	
					2 =	α	
				HUNDERTER:	Vorlauf-/Einlaufwegbewertung <_APP>		
					0 =	Gewindevorlauf <_APP>	
					1 =	Gewindeeinlauf = Gewindeauslauf <_APP> = -<_ROP>	
					2 =	Gewindeeinlaufweg vorgeben <_APP> = -<_APP>	
				TAUSENDER:	Auswahl Zustellschräge als Winkel oder Breite		
					0 =	Zustellwinkel <_IANG>	
					1 =	Zustellschräge <_IFLANK>	
				ZEHNTAUSENDER:	eingängig/mehrgängig		
					0 =	eingängig (mit Startwinkelversatz <_NSP>)	
					1 =	mehrgängig	
HUNDERTTAUSENDER:	Startgang <_GST>						
	0 =	komplette Bearbeitung					
	1 =	ab diesem Gang die Bearbeitung starten					
	2 =	nur diesen Gang bearbeiten					
EINEMILLION:	Durchhangskompensation bei Längsgewinde						
	0 =	Segmenthöhe balliges Gewinde XS					
	1 =	Radius balliges Gewinde RS					
30	XS / RS	<_S_XRS>	REAL	Durchhangskompensation bei Längsgewinde in Verbindung mit <_AMODE>: EINEMILLION			

Hinweis

¹⁾ Die Parameter <_PITM>, <_PTAB> und <_PTABA> werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet.
Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich.

19.1.31 CYCLE435 - Abrichterposition berechnen

Syntax

CYCLE435 (<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>, <S_PVL>, <S_PD>, <S_PL>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1		<_T>	STRING[32]	Werkzeugname der Schleifscheibe
2		<_DD>	INT	Schneidenummer der Schleifscheibe
3		<S_TA>	STRING[32]	Bezugspunkt Abrichter - Name des Abrichters
4		<S_DA>	INT	Schneidenummer des Abrichters
5		<S_AD>	REAL	Abrichtbetrag Durchmesser
6		<S_AL>	REAL	Abrichtbetrag plan
7		<S_PVD>	REAL	Profilierverschiebung Durchmesser
8		<S_PVL>	REAL	Profilierverschiebung plan
9		<S_PD>	REAL	Profilieraufmaß Durchmesser
10		<S_PL>	REAL	Profilieraufmaß plan
11		<_AMODE>	INT	Alternativmodus
			EINER:	aktives Werkzeug bei Zyklusende
				0 = Abrichter aktiv
				1 = Scheibe aktiv

19.1.32 CYCLE495 - Profilieren

Syntax

CYCLE495 (<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>, <S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1		<_T>	STRING[20]	Werkzeugname der Schleifscheibe
2		<_DD>	INT	Schneidenummer der Schleifscheibe
3		<_SC>	REAL	Abhebebetrag zum Umfahren von Hindernissen, inkrementell
4		<_F>	REAL	Vorschub Profilieren

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung			
5		<_VARI>	INT	Bearbeitungsart			
				EINER:	Profilierungsart		
					1 =	achsparell	
					2 =	konturparell	
				ZEHNER:	Bearbeitungsrichtung		
					0 =	ziehend möglich mit den Schneidenlagen 1 bis 4	
					1 =	stoßend möglich mit den Schneidenlagen 1 bis 4	
					2 =	wechselnd möglich mit den Schneidenlagen 1 bis 8	
					3 =	Anfang → Ende möglich mit den Schneidenlagen 1 bis 8	
					4 =	Ende → Anfang möglich mit den Schneidenlagen 1 bis 8	
				HUNDERTER:	Zustellrichtung		
					1 =	Zustellung X- bei G18 bzw. Y- bei G19	
					2 =	Zustellung X+ bei G18 bzw. Y+ bei G19	
3 =	Zustellung Z- bei G18 und bei G19						
4 =	Zustellung Z+ bei G18 und bei G19						
6		<_D>	REAL	Abrichtbetrag bei der Profilierart achsparell			
7		<_DX>	REAL	Abrichtbetrag X bei G18 bzw. Y bei G19 bei der Profilierart konturparell			
8		<_DZ>	REAL	Abrichtbetrag Z bei G18 und G19 bei der Profilierart konturparell			
9		<S_PA>	REAL	Profilieraufmaß			
10		<S_N>	INT	Anzahl der Hübe im Profilierprogramm			
11		<_DMODE>	INT	Anzeigemodus			
				EINER:	Bearbeitungsebene G17/G18/G19		
					0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv	
					1 =	G17 (nur im Zyklus aktiv)	
					2 =	G18 (nur im Zyklus aktiv)	
3 =	G19 (nur im Zyklus aktiv)						

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
12		<_AMODE>	INT	Alternativmodus		
				EINER:	Auswahl Profilieren Neu/Fortsetzen	
					1 =	Neu
					2 =	Fortsetzen
				ZEHNER:	Auswahl Profilieraufmaß	
					0 =	von der Rohkontur bis zum tiefsten Punkt der Kontur
1 =	von der Rohkontur bis zum höchsten Punkt der Kontur					
13		<S_FW>	REAL	Freiwinkel des Abrichters		
14		<S_HW>	REAL	Halterwinkel des Abrichters		

19.1.33 CYCLE800 - Schwenken

Syntax

CYCLE800 (<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>, <_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
1		<_FR>	INT	Freifahrmodus:	
				0 =	kein Freifahren
				1 =	Freifahren Maschinenachse Z
				2 =	Freifahren Maschinenachse Z und danach XY
				3 =	reserviert
				4 =	Freifahren in Werkzeugrichtung maximal
	5 =	Freifahren in Werkzeugrichtung inkrementell			
2		<_TC>	STRING[32]	Name Schwenkdatensatz:	
				""	(kein Name) wenn nur 1 Schwenkdatensatz vorhanden
				"0"	Abwahl Schwenkdatensatz (Löschen der Schwenkframes)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
3		<_ST>	INT	Status Transformationen	
				EINER:	0 = Neu , Schwenkebene wird gelöscht und mit den aktuellen Parametern neu berechnet
				1 = Additiv , Schwenkebene setzt additiv auf aktive Schwenkebene auf	
				ZEHNER:	Werkzeugspitze nachführen ja/nein (nur aktiv, wenn in IBN SCHWENKEN Funktion eingerichtet ist)
				0 = Werkzeugspitze nicht nachführen	
				1 = Werkzeugspitze nachführen (TRAORI)	
				HUNDERTER:	Werkzeug anstellen / ausrichten (Funktion wird in Eingabemaske SCHWENKEN Werkzeug angezeigt)
				0 = Werkzeug nicht anstellen	
				1 = Werkzeug (vorzugweise Radienfräser) anstellen	
				2 = Dreh werkzeug ausrichten (wenn B-Achskinematik für Drehtechnologie in IBN Schwenken eingerichtet ist)	
				3 = Fräs werkzeug ausrichten (wenn B-Achskinematik für Drehtechnologie in IBN Schwenken eingerichtet ist)	
				TAUSENDER:	interner Parameter Schwenken in JOG
				ZEHNTAUSENDER:	siehe Parameter Richtung <_DIR>
				0 = Schwenken "ja"	
				1 = Schwenken "nein" Richtung "Minus" ³⁾	
				2 = Schwenken "nein" Richtung "Plus" ³⁾	
				HUNDERTTAUSENDER:	siehe Parameter Richtung <_DIR>
				0 = Kompatibilität	
1 = Richtungsauswahl "Minus" optimiert ⁴⁾					
2 = Richtungsauswahl "Plus" optimiert ⁴⁾					

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
4		<_MODE> ⁵⁾	INT	Schwenkmodus: Auswertung der Schwenkwinkel und der Schwenksequenzfolge (bitcodiert!)		
				Bit: 7 6	0 0:	Schwenkwinkel achsweise -> siehe Parameter <_A>, <_B>, <_C>
					0 1:	Raumwinkel -> siehe Parameter <_A>, <_B> ¹⁾
					1 0:	Projektionswinkel -> siehe Parameter <_A>, <_B>, <_C> ¹⁾
					1 1:	Schwenkmodus Rundachsen direkt -> siehe Parameter <_A>, <_B> ¹⁾
				Bit: 5 4 3 2 1 0 (bei Raumwinkeln ohne Bedeutung!)	x x x x 0 1	1.Drehung _A um X
					x x x x 1 0	1.Drehung _A um Y
					x x x x 1 1	1.Drehung _A um Z
					x x 0 1 x x	2.Drehung _B um X
					x x 1 0 x x	2.Drehung _B um Y
x x 1 1 x x	2.Drehung _B um Z					
0 1 x x x x	3.Drehung _C um X					
1 0 x x x x	3.Drehung _C um Y					
1 1 x x x x	3.Drehung _C um Z					
5	X0	<_X0>	REAL	Bezugspunkt X vor der Drehung		
6	Y0	<_Y0>	REAL	Bezugspunkt Y vor der Drehung		
7	Z0	<_Z0>	REAL	Bezugspunkt Z vor der Drehung		
8	X(A)	<_A>	REAL	1. Drehung laut Einstellung in Parameter <_MODE>		
9	Y(B)	<_B>	REAL	2. Drehung laut Einstellung in Parameter <_MODE>		
10	Z(C)	<_C>	REAL	3. Drehung laut Einstellung in Parameter <_MODE>		
11	X1	<_X1>	REAL	Bezugspunkt X nach der Drehung		
12	Y1	<_Y1>	REAL	Bezugspunkt Y nach der Drehung		
13	Z1	<_Z1>	REAL	Bezugspunkt Z nach der Drehung		
14	- oder +	<_DIR>	INT	Verfahrbewegung der Rundachsen auslösen (default = -1!)		
				-1 =	auf kleineren Wert der Rundachse 1 oder 2 positionieren ²⁾	
				+1 =	auf größeren Wert der Rundachse 1 oder 2 positionieren ²⁾	
				0 =	Schwenken nein (nur Schwenkfräse berechnen) ^{1) 3)}	
15	FR	<_FR_I>	REAL	Wert (ink) Freifahren in Werkzeugrichtung inkrementell		

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
16		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	Darstellung des Beta-Wertes bei Ausrichten Werkzeug
				0 =	Wert
1 =	Pfeil				

Hinweis

Sind folgende Übergabeparameter indirekt (als Parameter) programmiert, wird die Eingabemaske nicht rückübersetzt: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

¹⁾ Auswahl möglich, wenn in IBN SCHWENKEN Funktion eingerichtet ist.

²⁾ Auswahl möglich, wenn in IBN SCHWENKEN Richtungsbezug auf Rundachse 1 oder 2 eingestellt ist.

kein Auswahlfeld bei Richtungsbezug nein

³⁾ Auswahl Schwenken "nein" kann ausgeblendet sein SD 55221 Bit 0

Schwenken "nein" Richtung "Minus" entspricht <_DIR> = 0 und _ST ZEHNTAUSENDER = 1

Schwenken "nein" Richtung "Plus" entspricht <_DIR> = 0 und _ST ZEHNTAUSENDER = 2

⁴⁾ Die Richtungsauswahl der Rundachse 1 oder 2 erfolgt auch, wenn sich die Rundachse mit dem Richtungsbezug in Polstellung (Positionswert gleich Null) befindet.

⁵⁾ Beispiel Codierung: Drehung achsweise, Drehreihenfolge ZYX

Binär: 00011011; Dezimal: 27

Die Achsbezeichner XYZ entsprechen den Geometrieachsen des NC-Kanals. Drehungen um die Achsen XYZ dürfen einzeln ausgeführt werden. Z.B. Drehreihenfolge um ZXZ ist in einem Aufruf des CYCLE800 nicht erlaubt

19.1.34 CYCLE801 - Gitter oder Rahmen

Syntax

```
CYCLE801 (<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung										
1	X0	<_SPCA>	REAL	Bezugspunkt für Positionsmuster (Gitter/Rahmen) in der 1. Achse (abs)										
2	Y0	<_SPCO>	REAL	Bezugspunkt für Positionsmuster (Gitter/Rahmen) in der 2. Achse (abs)										
3	$\alpha 0$	<_STA>	REAL	Grund-Drehwinkel (Winkel zur 1. Achse) <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>< 0 =</td> <td>Drehung im Uhrzeigersinn</td> </tr> <tr> <td>> 0 =</td> <td>Drehung im Gegenuhrzeigersinn</td> </tr> </table>	< 0 =	Drehung im Uhrzeigersinn	> 0 =	Drehung im Gegenuhrzeigersinn						
< 0 =	Drehung im Uhrzeigersinn													
> 0 =	Drehung im Gegenuhrzeigersinn													
4	L1	<_DIS1>	REAL	Abstand der Spalten (Positionsabstand 1. Achse, ohne Vorzeichen eingeben)										
5	L2	<_DIS2>	REAL	Abstand der Zeilen (Positionsabstand 2. Achse, ohne Vorzeichen eingeben)										
6	N1	<_NUM1>	INT	Anzahl der Spalte										
7	N2	<_NUM2>	INT	Anzahl der Zeilen										
8		<_VARI>	INT	Bearbeitungsart <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>EINER:</td> <td>Positionsmuster</td> </tr> <tr> <td>0 =</td> <td>Gitter</td> </tr> <tr> <td>1 =</td> <td>Rahmen</td> </tr> <tr> <td>ZEHNER:</td> <td>reserviert</td> </tr> <tr> <td>HUNDERTER:</td> <td>reserviert</td> </tr> </table>	EINER:	Positionsmuster	0 =	Gitter	1 =	Rahmen	ZEHNER:	reserviert	HUNDERTER:	reserviert
EINER:	Positionsmuster													
0 =	Gitter													
1 =	Rahmen													
ZEHNER:	reserviert													
HUNDERTER:	reserviert													
9		<_UMODE>	INT	reserviert										
10	αX	<_ANG1>	REAL	Scherwinkel zur 1. Achse (Schrägstellung der Zeilen bezogen auf die 1. Achse) <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>< 0 =</td> <td>Messung im Uhrzeigersinn (0 bis -90 Grad)</td> </tr> <tr> <td>> 0 =</td> <td>Messung im Gegenuhrzeigersinn (0 bis 90 Grad)</td> </tr> </table>	< 0 =	Messung im Uhrzeigersinn (0 bis -90 Grad)	> 0 =	Messung im Gegenuhrzeigersinn (0 bis 90 Grad)						
< 0 =	Messung im Uhrzeigersinn (0 bis -90 Grad)													
> 0 =	Messung im Gegenuhrzeigersinn (0 bis 90 Grad)													
11	αY	<_ANG2>	REAL	Scherwinkel zur 2. Achse (Schrägstellung der Spalten bezogen auf die 2. Achse) <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>< 0 =</td> <td>Messung im Uhrzeigersinn (0 bis -90 Grad)</td> </tr> <tr> <td>> 0 =</td> <td>Messung im Gegenuhrzeigersinn (0 bis 90 Grad)</td> </tr> </table>	< 0 =	Messung im Uhrzeigersinn (0 bis -90 Grad)	> 0 =	Messung im Gegenuhrzeigersinn (0 bis 90 Grad)						
< 0 =	Messung im Uhrzeigersinn (0 bis -90 Grad)													
> 0 =	Messung im Gegenuhrzeigersinn (0 bis 90 Grad)													
12		<_HIDE>	STRING [200]	Ausgeblendete Positionen <ul style="list-style-type: none"> • max. 198 Zeichen • Angabe der fortlaufenden Positionsnummer, z. B. "1,3" (Positionen 1 und 3 werden nicht ausgeführt) 										
13		<_NSP>	INT	reserviert										

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
14		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
	3 =	G19 (nur im Zyklus aktiv)			

19.1.35 CYCLE802 - Beliebige Positionen

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
1		<_XA>	INT	Alternative für alle X-Positionen (9-stelliger Dezimalwert) Stellenzahl: 876543210 (Die Stelle entspricht der Bohrposition Xn)		
				Stellenwert:	1 =	absolut (1. programmierte Position immer absolut)
					2 =	inkrementell
2		<_YA>	INT	Alternative für alle Y-Positionen (9-stelliger Dezimalwert) Stellenzahl: 876543210 (Die Stelle entspricht der Bohrposition Yn)		
				Stellenwert:	1 =	absolut (1. programmierte Position immer absolut)
					2 =	inkrementell
3	X0	<_X0>	REAL	1. Position X		
4	Y0	<_Y0>	REAL	1. Position Y		
5	X1	<_X1>	REAL	2. Position X		
6	Y1	<_Y1>	REAL	2. Position Y		
7	X2	<_X2>	REAL	3. Position X		
8	Y2	<_Y2>	REAL	3. Position Y		
9	X3	<_X3>	REAL	4. Position X		
10	Y3	<_Y3>	REAL	4. Position Y		
11	X4	<_X4>	REAL	5. Position X		

Zyklen extern programmieren

19.1 Technologische Zyklen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
12	Y4	<_Y4>	REAL	5. Position Y
13	X5	<_X5>	REAL	6. Position X
14	Y5	<_Y5>	REAL	6. Position Y
15	X6	<_X6>	REAL	7. Position X
16	Y6	<_Y6>	REAL	7. Position Y
17	X7	<_X7>	REAL	8. Position X
18	Y7	<_Y7>	REAL	8. Position Y
19	X8	<_X8>	REAL	9. Position X
20	Y8	<_Y8>	REAL	9. Position Y

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
21		<_VARI>	INT	Bearbeitung	
				HUNDERTER:	(nur für Aufruf aus Jobshop) (vorerst nur 0 und 2 ausgewertet)
				0 =	kein Spindel klemmen
				1 =	Spindel klemmen nur bei senkrechtem Eintauchen mit G00 oder G01
				2 =	Spindel klemmen während der gesamten Bearbeitung
				TAUSENDER:	reserviert
				ZEHNTAUSENDER:	Positionsmuster mit/ohne Rundachse – Achskombination (mit <_VARI> HUNDERTTAUSENDER)
				0 =	XY (nur XY ohne Rundachse, Kompatibilität)
				1 =	X,Y oder Z und Rundachse: XA, YB, ZC (1 Rundachse mit Geo-Achse, um die die Rundachse dreht)
				2 =	XY und Rundachse: XYA, XYB, XYC (1 Rundachse mit 1. und 2. Geo-Achse, ohne TRACYL)
				HUNDERTTAUSENDER:	Rundachse
				0 =	ohne Rundachse (nur XY, Kompatibilität)
				1 =	A-Achse (Rundachse um X)
				2 =	B-Achse (Rundachse um Y)
				3 =	C-Achse (Rundachse um Z)
				ZEHNMILLIONEN + EINMILLION:	Positionsmuster mit Rundachse – Offset (bei mehreren Rundachsen um die gleiche Achse; wenn Index zu groß, dann 1. Achse)
				00 =	1. A, B oder C-Achse oder bei Kompatibilität
01 =	2. A, B oder C-Achse				
...					
19 =	20. A, B oder C-Achse				
22		<_UMODE>	INT	Auswahl der zu klemmenden Spindel: (nur für Aufruf aus Jobshop) (Aufruf Anwenderzyklus CUST_TECHCYC)	
				3 =	Klemmen/Lösen Hauptspindel
				23 =	Klemmen/Lösen Gegenspindel

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
23		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
24		<S_ABA>	INT	Alternative für alle AB-Positionen (9-stelliger Dezimalwert)	
				Stellenzahl: 876543210 (Die Stelle entspricht der Position ABn)	
				Stellenwert:	1 = absolut (1. programmierte Position immer absolut)
				2 =	inkrementell
25	A0	<S_AB0>	REAL	1. Rundachseposition bei Positionsmuster mit Rundachse (in Verbindung mit <_VARI>)	
26	A1	<S_AB1>	REAL	2. Rundachseposition bei Positionsmuster mit Rundachse	
27	A2	<S_AB2>	REAL	3. Rundachseposition bei Positionsmuster mit Rundachse	
28	A3	<S_AB3>	REAL	4. Rundachseposition bei Positionsmuster mit Rundachse	
29	A4	<S_AB4>	REAL	5. Rundachseposition bei Positionsmuster mit Rundachse	
30	A5	<S_AB5>	REAL	6. Rundachseposition bei Positionsmuster mit Rundachse	
31	A6	<S_AB6>	REAL	7. Rundachseposition bei Positionsmuster mit Rundachse	
32	A7	<S_AB7>	REAL	8. Rundachseposition bei Positionsmuster mit Rundachse	
33	A8	<S_AB8>	REAL	9. Rundachseposition bei Positionsmuster mit Rundachse	

Hinweis

Nicht benötigte Positionen der Parameter X1/Y1/A1 bis X8/Y8/A8 können weggelassen werden. Die Alternativwerte für <_XA>, <_YA> und <S_ABA> sind jedoch vollständig für alle 9 Positionen anzugeben.

Bei Positionsmuster XA, YB oder ZC (eine Geometrieachse und Rundachse) ist die Achse der Bearbeitungsebene, die über das Positionsmuster nicht verfahren wird (bei G17 und XA ist das Y) vor Zyklusaufwurf zu positionieren.

19.1.36 CYCLE830 - Tieflochbohren 2

Syntax

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>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung																													
1	RP	<RTP>	REAL	Rückzugsebene (abs)																													
2	Z0	<RFP>	REAL	Bezugspunkt (abs)																													
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen)																													
4	Z1	<_DP>	REAL	Endbohrtiefe abs/ink (siehe <_AMODE> EINER)																													
5	D	<FDEP>	REAL	1. Bohrtiefe zum Spänebrechen/Entspanen absolut oder inkrementell bezogen auf den Bezugspunkt bei Anbohren/ohne Anbohren oder bezogen auf Pilotbohrungstiefe (siehe <_AMODE> ZEHNTAUSENDER)																													
6	DF	<_DAM>	REAL	Betrag/Prozentsatz für jede weitere Zustellung Degressionsbetrag oder -Prozentsatz (siehe <_AMODE> HUNDERTTAUSENDER)																													
7	DTB	<DTB>	REAL	Verweilzeit auf jeder Bohrtiefe (siehe <_AMODE> ZEHNER)																													
8	DTS	<DTS>	REAL	Verweilzeit beim Entspanen am Anfangspunkt (siehe <_AMODE> HUNDERTER)																													
9	FD1	<FRF>	REAL	Prozentsatz für den Vorschub bei der ersten Zustellung (siehe <_AMODE> ZEHNMILLIONEN)																													
10		<VARI>	INT	<table border="1"> <tr> <td colspan="2">Bearbeitung</td> </tr> <tr> <td rowspan="4">EINER:</td> <td>Spänebrechen/Entspanen</td> </tr> <tr> <td>0 = in einem Schnitt</td> </tr> <tr> <td>1 = Spänebrechen</td> </tr> <tr> <td>2 = Entspanen</td> </tr> <tr> <td>3 = Spänebrechen und Entspanen</td> </tr> <tr> <td rowspan="3">ZEHNER:</td> <td>Rückzug bei Entspanen</td> </tr> <tr> <td>0 = auf Pilotbohrtiefe</td> </tr> <tr> <td>1 = auf Sicherheitsabstand</td> </tr> <tr> <td rowspan="3">HUNDERTER:</td> <td>weicher Anschnitt</td> </tr> <tr> <td>0 = nein</td> </tr> <tr> <td>1 = ja</td> </tr> <tr> <td rowspan="3">TAUSENDER:</td> <td>Durchbohren</td> </tr> <tr> <td>0 = nein</td> </tr> <tr> <td>1 = ja</td> </tr> <tr> <td rowspan="4">ZEHNTAUSENDER:</td> <td>Anbohren/Pilotbohrung</td> </tr> <tr> <td>0 = ohne Anbohren</td> </tr> <tr> <td>1 = mit Anbohren</td> </tr> <tr> <td>2 = mit Pilotbohrung</td> </tr> <tr> <td rowspan="3">HUNDERTTAUSENDER:</td> <td>Rückzug</td> </tr> <tr> <td>0 = auf Pilotbohrtiefe</td> </tr> <tr> <td>1 = auf Rückzugsebene</td> </tr> </table>	Bearbeitung		EINER:	Spänebrechen/Entspanen	0 = in einem Schnitt	1 = Spänebrechen	2 = Entspanen	3 = Spänebrechen und Entspanen	ZEHNER:	Rückzug bei Entspanen	0 = auf Pilotbohrtiefe	1 = auf Sicherheitsabstand	HUNDERTER:	weicher Anschnitt	0 = nein	1 = ja	TAUSENDER:	Durchbohren	0 = nein	1 = ja	ZEHNTAUSENDER:	Anbohren/Pilotbohrung	0 = ohne Anbohren	1 = mit Anbohren	2 = mit Pilotbohrung	HUNDERTTAUSENDER:	Rückzug	0 = auf Pilotbohrtiefe	1 = auf Rückzugsebene
Bearbeitung																																	
EINER:	Spänebrechen/Entspanen																																
	0 = in einem Schnitt																																
	1 = Spänebrechen																																
	2 = Entspanen																																
3 = Spänebrechen und Entspanen																																	
ZEHNER:	Rückzug bei Entspanen																																
	0 = auf Pilotbohrtiefe																																
	1 = auf Sicherheitsabstand																																
HUNDERTER:	weicher Anschnitt																																
	0 = nein																																
	1 = ja																																
TAUSENDER:	Durchbohren																																
	0 = nein																																
	1 = ja																																
ZEHNTAUSENDER:	Anbohren/Pilotbohrung																																
	0 = ohne Anbohren																																
	1 = mit Anbohren																																
	2 = mit Pilotbohrung																																
HUNDERTTAUSENDER:	Rückzug																																
	0 = auf Pilotbohrtiefe																																
	1 = auf Rückzugsebene																																
11	V1	<_MDEP>	REAL	minimale Zustellung inkrementell (nur bei Prozentsatz für Degression)																													
12	V2	<_VRT>	REAL	<table border="1"> <tr> <td colspan="2">Rückzugsbetrag nach jeder Bearbeitung inkrementell (nur bei Spänebrechen)</td> </tr> <tr> <td>0 =</td> <td>Standardwert 1 mm</td> </tr> <tr> <td>> 0 =</td> <td>variabler Rückzugsbetrag</td> </tr> </table>	Rückzugsbetrag nach jeder Bearbeitung inkrementell (nur bei Spänebrechen)		0 =	Standardwert 1 mm	> 0 =	variabler Rückzugsbetrag																							
Rückzugsbetrag nach jeder Bearbeitung inkrementell (nur bei Spänebrechen)																																	
0 =	Standardwert 1 mm																																
> 0 =	variabler Rückzugsbetrag																																
13	DT	<_DTD>	REAL	Verweilzeit auf Endbohrtiefe (siehe <_AMODE> TAUSENDER)																													

Zyklen extern programmieren

19.1 Technologische Zyklen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
14	V3	<_DIS1>	REAL	Vorhalteabstand inkremental nur bei Entspanen (siehe <_AMODE> EINEMILLION)	
15	FP	<S_FP>	REAL	Vorschub zum Einfahren in die Pilotbohrung als Wert oder in % (in Verbindung mit <S_AMODE2> HUNDERTER)	
16		<S_SDAC2>	INT	Spindeldrehrichtung beim Anfahren	
				3 =	M3
				4 =	M4
				5 =	M5 (Default)
17	SP	<S_SV2>	REAL	Anfahren mit	
	V4			konstanter Spindeldrehzahl (siehe <S_AMODE2> ZEHNMILLIONEN)	
				konstanter Schnittgeschwindigkeit	
				Spindeldrehzahl in % von der Bohrdrehzahl	
18	F	<S_FB>	REAL	Bohrvorschub (siehe <S_AMODE2> EINER)	
19		<_SDAC>	REAL	Spindeldrehrichtung beim Bohren	
				3 =	M3
				4 =	M4
20	S	<_SV1>	REAL	Bohren mit	
	V5			konstanter Spindeldrehzahl (siehe <S_AMODE2> EINEMILLION)	
				konstanter Schnittgeschwindigkeit	
21	SPOS	<S_SPOS>	REAL	Spindelposition, nur wenn Anfahren mit M5	
22	ZA	<S_ZA>	REAL	Anbohrtiefe inkrementell bezogen auf Bezugspunkt oder absolut (siehe <S_AMODE3> EINER)	
23	FA	<S_FA>	REAL	Anbohrvorschub als Wert oder in % (in Verbindung mit <S_AMODE2> ZEHNER)	
24	ZP	<S_ZP>	REAL	Pilotbohrung inkrementell bezogen auf Bezugspunkt, absolut oder Faktor des Bohrungsdurchmessers (siehe <S_AMODE3> ZEHNER)	
25	FS	<S_FS>	REAL	Vorschub Anschnitt als Wert oder in % (in Verbindung mit <S_AMODE2> TAUSENDER)	
26	ZS1	<S_ZS1>	REAL	Tiefe jedes Anschnitts mit konstantem Vorschub (ink)	
27	ZS2	<S_ZS2>	REAL	Tiefe jedes Anschnitts für Vorschuberrhöhung (ink)	
28	N	<S_N>	INT	Anzahl der Spänebrechhübe vor jedem Entspanen	
29	ZD	<S_ZD>	REAL	Restbohrtiefe inkrementell bezogen auf Endbohrtiefe oder absolut (in Verbindung mit <S_AMODE3> HUNDERTER)	
30	FD	<S_FD>	REAL	Restbohrvorschub als Wert oder in % (in Verbindung mit <S_AMODE2> ZEHNTAUSENDER)	
31	FR	<S_FR>	REAL	Vorschub Rückzug (in Verbindung mit <S_AMODE2> HUNDERTTAUSENDER)	
32		<S_SDAC3>	INT	Spindeldrehrichtung beim Rückzug	
				3 =	M3
				4 =	M4
				5 =	M5
33	SR	<S_SV3>	REAL	Rückzug mit	
	V6			konstanter Spindeldrehzahl (siehe <S_AMODE2> HUNDERTMILLIONEN)	
				konstanter Schnittgeschwindigkeit	
				Spindeldrehzahl in % von der Bohrdrehzahl	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
34	Kühlmittel ein	<S_CON>	STRING[10]	Kühlmittel ein, M-Befehl oder Unterprogrammaufruf	
35	Kühlmittel aus	<S_COFF>	STRING[10]	Kühlmittel aus, M-Befehl oder Unterprogrammaufruf	
36		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)	
				EINER:	reserviert
				ZEHNER:	Bohrtiefe bezogen auf Spitze/Schaft
				0 =	Spitze
				1 =	Schaft
37		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
				3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	reserviert
				HUNDERTER:	reserviert
				TAUSENDER:	reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)
	0 =	Eingabe: komplett			
	1 =	Eingabe: einfach			

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
38		<_AMODE>	INT	Alternativmodus 1		
				EINER:	Bohrtiefe = Endbohrtiefe Z1 abs/ink	
					0 =	inkrementell
					1 =	absolut
				ZEHNER:	Verweilzeit auf jeder Bohrtiefe DTB in Sekunden/Umdrehungen	
					0 =	in Sekunden
					1 =	in Umdrehungen
				HUNDETER:	Verweilzeit zum Entspanen DTS in Sekunden/Umdrehungen	
					0 =	in Sekunden
					1 =	in Umdrehungen
				TAUSENDER:	Verweilzeit auf Endbohrtiefe DT in Sekunden/Umdrehungen	
					0 =	in Sekunden
					1 =	in Umdrehungen
				ZEHNTAUSENDER:	1. Bohrtiefe D abs/ink	
0 =	inkrementell					
1 =	absolut					
HUNDERTTAUSENDER:	Betrag/Prozentsatz DF für jede weitere Zustellung (Degression)					
	0 =	Betrag				
	1 =	Prozentsatz (0,001 bis 100 %)				
EINEMILLION:	Vorhalteabstand V3 automatisch/manuell					
	0 =	automatisch (wird im Zyklus berechnet)				
	1 =	manuell (programmierter Wert)				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
39		<S_ AMODE2 >	INT	Alternativmodus 2	
				EINER:	EINER: Bohrvorschub F
				0 =	F/min
				1 =	F/U
				ZEHNER:	Bewertung Anbohrvorschub FA
				0 =	in % vom Bohrvorschub
				1 =	F/min
				2 =	F/U
				HUNDERTER:	Bewertung Vorschub zum Einfahren in Pilotbohrung FP
				0 =	in % vom Bohrvorschub
				1 =	F/min
				2 =	F/U
				TAUSENDER:	Bewertung Vorschub Anschnitt FS
				0 =	in % vom Bohrvorschub
				1 =	F/min
				2 =	F/U
				ZEHNTAUSENDER:	Bewertung Vorschub Durchbohren FD
				0 =	in % vom Bohrvorschub
				1 =	F/min
				2 =	F/U
				HUNDERTTAUSENDER:	Vorschub Rückzug FR
				0 =	F/min
				1 =	Eilgang
				EINEMILLION:	Bohren - Spindeldrehzahl/Schnittgeschwindigkeit (S/V5)
0 =	konstante Spindeldrehzahl				
1 =	konstante Schnittgeschwindigkeit				
ZEHNMILLIONEN:	Anfahren mit Spindeldrehzahl/Schnittgeschwindigkeit (SP/V4)				
0 =	konstante Spindeldrehzahl				
1 =	konstante Schnittgeschwindigkeit				
2 =	Spindeldrehzahl in % von der Bohrdrehzahl				
HUNDERTMILLIONEN:	Rückzug - Spindeldrehzahl/Schnittgeschwindigkeit (SR/V6)				
0 =	konstante Spindeldrehzahl				
1 =	konstante Schnittgeschwindigkeit				
2 =	Spindeldrehzahl in % von der Bohrdrehzahl				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
40		<S_AMODE3 >	INT	Alternativmodus 3		
				EINER:	Anbohrtiefe ZA abs/ink	
					0 =	inkrementell
					1 =	absolut
				ZEHNER:	Tiefe der Pilotbohrung ZP	
					0 =	inkrementell
					1 =	absolut
				HUNDETER:	Restbohrtiefe ZD ink/abs	
					0 =	inkrementell
					1 =	absolut
				2 =	Faktor des Bohrungsdurchmessers	

19.1.37 CYCLE832 - High Speed Settings

Syntax

CYCLE832 (<S_TOL>, <S_TOLM>, <S_OTOL>)

Hinweis

Der CYCLE832 entlastet den Maschinenhersteller nicht von notwendigen Optimierungsaufgaben bei der Inbetriebnahme der Maschine. Dies betrifft die Optimierung der an der Bearbeitung beteiligten Achsen und die Einstellungen der NCU (Vorsteuerung, Ruckbegrenzung, usw.).

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
1	Toleranz	<S_TOL>	REAL	Konturtoleranz Die Konturtoleranz entspricht der Achstoleranz der Geometrieachsen.	
2		<S_TOLM>	INT	Bearbeitungsart (Technologie)	
				EINER:	
				0 =	Abwahl
				1 =	Schlichten (Finish)
				2 =	Vorschlichten (Semifinish)
				3 =	Schruppen (Rough)
				ZEHNER:	
				0 =	Kompatibilität ¹⁾ bzw. keine Orientierungstoleranz
				1 =	Orientierungstoleranz im Parameter <S_OTOL>
				HUNDERTER ... HUNDERTTAUSENDER	belegt wegen Kompatibilität
EINEMILLION:					
	0 =	Kompatibilität. Es wird automatisch die beste zur Verfügung stehende Formenbaufunktion verwendet: <ul style="list-style-type: none"> • Option Top Surface nicht aktiv: ⇒ Advanced Surface • Option Top Surface aktiv: ⇒ Top Surface mit Glättung 			
	1 =	Top Surface ohne Glättung			
	2 =	Top Surface mit Glättung			
3	ORI-Toleranz	<S_OTOL>	REAL	Orientierungstoleranz bzw. Versionskennzeichen CYCLE832 Toleranzparameter für die Orientierung des Werkzeugs. Wird benötigt bei der Abarbeitung eines Hochgeschwindigkeitsbearbeitungsprogramms auf Maschinen mit dynamischer Orientierungstransformation (z. B. der 5-Achs-Bearbeitung). Der Parameter <S_OTOL> muss programmiert werden. Dies gilt auch für Anwendungen auf 3-Achs-Maschinen bei Programmen ohne Orientierung des Werkzeugs (<S_OTOL> = 1).	

¹⁾ Orientierungstoleranz abgeleitet von der Konturtoleranz multipliziert mit dem Faktor aus den Zyklen-Settingdaten SD55441 bis SD55443.

Literatur:

Inbetriebnahmehandbuch Basesoftware und Bedien-Software; SINUMERIK Operate (IM9), Kapitel "Funktion High Speed Setting (CYCLE832) konfigurieren"

Klartexteingabe

Um die Lesbarkeit des Zyklenaufrufs zu verbessern, kann der Parameter <S_TOLM> (Bearbeitungsart) auch im Klartext eingegeben werden. Klartexte sind sprachunabhängig. Folgende Eingaben sind zulässig:

_OFF	für	0	Abwahl
_FINISH	für	1	Schlichten
_SEMIFIN	für	2	Vorschlichten
_ROUGH	für	3	Schruppen
_ORI_FINISH	für	11	Schlichten mit Eingabe einer Orientierungstoleranz
_ORI_SEMIFIN	für	12	Vorschlichten mit Eingabe einer Orientierungstoleranz
_ORI_ROUGH	für	13	Schruppen mit Eingabe einer Orientierungstoleranz
_TOP_SURFACE_SMOOTH_OFF	für	1000000	Top Surface ohne Glättung
_TOP_SURFACE_SMOOTH_ON	für	2000000	Top Surface mit Glättung

Für die Klartexteingabe bei Top Surface werden die Klartexte wie im folgenden Beispiel kombiniert:

```
CYCLE832 (0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

Hinweis

Die Klartexte sind an die Funktionsnamen der G-Gruppe 59 (Dynamikmodus für Bahninterpolation) angelehnt. Mit diesen Klartexten werden 3-Achs-Maschinen und Maschinen mit Mehrachsorientierungstransformation (TRAORI) in der Anwendung klar getrennt.

Abwahl von CYCLE832

Bei Abwahl von CYCLE832 muss der Parameter <S_TOL> mit Null übergeben werden.

```
Beispiel: CYCLE832 (0, 0, 1)
```

Die Syntax CYCLE832 () ist für die Abwahl von CYCLE832 ebenfalls erlaubt.

Beispiele

Beispiel 1: CYCLE832 auf 3-Achs-Maschine ohne Orientierungstransformation

a) Zyklenaufruf mit Klartexteingabe

Programmcode	Kommentar
G710	; Maßsystem ist metrisch.
CYCLE832 (0.004, _FINISH, 1)	; Aufruf CYCLE832 mit: Konturtoleranz = 0,004 mm, Bearbeitungsart: Schlichten

Programmcode	Kommentar
...	; Abarbeitung eines Hochgeschwindigkeitsbearbeitungsprogramms

b) Zyklenuufruf ohne Klartexteingabe

Programmcode	Kommentar
G710	; s. o.
CYCLE832(0.004,1,1)	; s. o.
...	; s. o.

Beispiel 2: CYCLE832 auf 5-Achs-Maschine mit Orientierungstransformation

a) Zyklenuufruf und Abwahl mit Klartexteingabe

Programmcode	Kommentar
G710	; Maßsystem ist metrisch.
TRAORI	; Orientierungstransformation aktivieren.
CYCLE832(0.3,_ORI_ROUGH,0.8)	; Aufruf CYCLE832 mit: Konturtoleranz = 0,3 mm, Bearbeitungsart: Schruppen mit Eingabe einer Orientierungstoleranz, Orientierungstoleranz = 0,8 Grad
...	; Abarbeitung eines Hochgeschwindigkeitsbearbeitungsprogramms
CYCLE832(0,_OFF,1)	; Konturtoleranz = 0, Bearbeitungsart: Abwahl CYCLE832, Orientierungstoleranz = 0 Grad

b) Zyklenuufruf und Abwahl ohne Klartexteingabe

Programmcode	Kommentar
G710	; s. o.
TRAORI	; s. o.
CYCLE832(0.3,13,0.8)	; s. o.
...	; s. o.
CYCLE832(0,0,1)	; s. o.

19.1.38 CYCLE840 - Gewindebohren mit Ausgleichsfutter**Syntax**

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung															
1	RP	<RTP>	REAL	Rückzugsebene (abs)															
2	Z0	<RFP>	REAL	Bezugspunkt (abs)															
3	SC	<SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)															
4	Z1	<DP>	REAL	Bohrtiefe (abs), siehe <_AMODE>															
5	Z1	<DPR>	REAL	Bohrtiefe (ink) , siehe <_AMODE>															
6	DT	<DTB>	REAL	Verweilzeit auf Bohrtiefe/auf Sicherheitsabstand nach Rückzug in Sekunden, siehe <ENC>															
7		<SDR>	INT	Drehrichtung für den Rückzug															
8	SDE	<SDAC>	INT	Drehrichtung nach Zyklusende															
9		<ENC>	INT	<table border="1"> <tr> <td colspan="3">Gewindebohren mit Spindelgeber (G33)/Gewindebohren ohne Spindelgeber(G63)</td> </tr> <tr> <td>0 =</td> <td>mit Spindelgeber</td> <td>- Steigung aus <MPIT>/<PIT> - ohne DT</td> </tr> <tr> <td>20 =</td> <td>mit Spindelgeber</td> <td>- Steigung aus <MPIT>/<PIT> - mit DT nach Rückzug auf Sicherheitsabstand</td> </tr> <tr> <td>11 =</td> <td>ohne Spindelgeber</td> <td>- Steigung aus <MPIT>/<PIT> - mit DT auf Bohrtiefe</td> </tr> <tr> <td>1 =</td> <td>ohne Spindelgeber</td> <td>- Steigung aus programmierten Vorschub - mit DT auf Bohrtiefe (Vorschub = Drehzahl · Steigung)</td> </tr> </table>	Gewindebohren mit Spindelgeber (G33)/Gewindebohren ohne Spindelgeber(G63)			0 =	mit Spindelgeber	- Steigung aus <MPIT>/<PIT> - ohne DT	20 =	mit Spindelgeber	- Steigung aus <MPIT>/<PIT> - mit DT nach Rückzug auf Sicherheitsabstand	11 =	ohne Spindelgeber	- Steigung aus <MPIT>/<PIT> - mit DT auf Bohrtiefe	1 =	ohne Spindelgeber	- Steigung aus programmierten Vorschub - mit DT auf Bohrtiefe (Vorschub = Drehzahl · Steigung)
Gewindebohren mit Spindelgeber (G33)/Gewindebohren ohne Spindelgeber(G63)																			
0 =	mit Spindelgeber	- Steigung aus <MPIT>/<PIT> - ohne DT																	
20 =	mit Spindelgeber	- Steigung aus <MPIT>/<PIT> - mit DT nach Rückzug auf Sicherheitsabstand																	
11 =	ohne Spindelgeber	- Steigung aus <MPIT>/<PIT> - mit DT auf Bohrtiefe																	
1 =	ohne Spindelgeber	- Steigung aus programmierten Vorschub - mit DT auf Bohrtiefe (Vorschub = Drehzahl · Steigung)																	
10		<MPIT>	REAL	<p>Gewindegröße nur für "ISO metrisch" (Steigung wird intern während der Laufzeit berechnet)</p> <p>Wertebereich: 3 bis 48 (für M3 bis M48), alternativ zu <PIT></p>															
11		<PIT>	REAL	<p>Gewindesteigung als Wert, Maßeinheit siehe <_PITA></p> <p>Wertebereich: > 0, alternativ zu MPIT</p>															
12		<_AXN>	INT	<table border="1"> <tr> <td rowspan="4">Bohrachse</td> <td>0 =</td> <td>3. Geometrieachse</td> </tr> <tr> <td>1 =</td> <td>1. Geometrieachse</td> </tr> <tr> <td>2 =</td> <td>2. Geometrieachse</td> </tr> <tr> <td>≥ 3 =</td> <td>3. Geometrieachse</td> </tr> </table>	Bohrachse	0 =	3. Geometrieachse	1 =	1. Geometrieachse	2 =	2. Geometrieachse	≥ 3 =	3. Geometrieachse						
Bohrachse	0 =	3. Geometrieachse																	
	1 =	1. Geometrieachse																	
	2 =	2. Geometrieachse																	
	≥ 3 =	3. Geometrieachse																	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
13		<_PITA>	INT	Maßeinheit der Gewindesteigung (Auswertung <PIT> und <MPIT>)		
				0 =	Steigung in mm	- Auswertung <MPIT>/<PIT>
				1 =	Steigung in mm	- Auswertung <PIT>
				2 =	Steigung in TPI	- Auswertung <PIT> (Gewindegänge pro inch)
				3 =	Steigung in inch	- Auswertung <PIT>
4 =	MODUL	- Auswertung <PIT>				
14		<_TECHNO>	INT	Technologie ¹⁾		
				EINER:	Genauhaltverhalten	
				0 =	Genauhaltverhalten wie vor Zyklusaufruf aktiv	
				1 =	Genauhalt G601	
				2 =	Genauhalt G602	
				3 =	Genauhalt G603	
				ZEHNER:	Vorsteuerung	
0 =	mit/ohne Vorsteuerung wie vor Zyklusaufruf aktiv					
1 =	mit Vorsteuerung FFWON					
2 =	ohne Vorsteuerung FFWOF					
15		<_PITM>	STRING[15]	String als Merker für Gewindesteigungseingabe ²⁾		
16		<_PTAB>	STRING[5]	String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") ²⁾		
17		<_PTABA>	STRING[20]	String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...) ²⁾		
18		<_GMODE>	INT	reserviert		

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung			
19		<_DMODE>	INT	Anzeigemodus			
				EINER:	Bearbeitungsebene G17/G18/G19		
					0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv	
					1 =	G17 (nur im Zyklus aktiv)	
					2 =	G18 (nur im Zyklus aktiv)	
					3 =	G19 (nur im Zyklus aktiv)	
				ZEHNER:	reserviert		
				HUNDERTER:	reserviert		
				TAUSENDER:	Kompatibilitätsmode (nur für Eingabemaske zu Rückübersetzen), wenn MD 52216 Bit0 = 1 ¹⁾		
					0 =	Technologieparameter werden angezeigt (Kompatibilität): TECHNO-Parameter wirken	
1 =	Technologieparameter werden nicht angezeigt: Technologie "wie vor Zyklusaufruf programmiert" wirkt						
ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zyklensmasken (Seite 734)						
	0 =	Eingabe: komplett					
	1 =	Eingabe: einfach					
20		<_AMODE>	INT	Alternativmodus			
				EINER:	Bohrtiefe Z1 (abs/ink)		
					0 =	Kompatibilität, aus Programmierung <DP>/<DPR>	
					1 =	inkrementell	
					2 =	absolut	
¹⁾ Felder Technologie können ausgeblendet sein in Abhängigkeit von Settingdatum SD52216 MCS_FUNCTION_MASK_DRILL ²⁾ Die Parameter 15, 16, und 17 werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet. Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich!							

19.1.39 CYCLE899 - Offene Nut fräsen

Syntax

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	RP	<_RTP>	REAL	Rückzugsebene (abs)
2	Z0	<_RFP>	REAL	Bezugspunkt der Werkzeugachse(abs)
3	SC	<_SDIS>	REAL	Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben)
4	Z1	<_DP>	REAL	Nuttiefe(abs/ink), siehe <_AMODE>
5	L	<_LENG>	REAL	Länge der Nut (ink)
6	W	<_WID>	REAL	Breite der Nut (ink)
7	X0	<_PA>	REAL	Bezugspunkt/Startposition 1. Achse (abs)
8	Y0	<_PO>	REAL	Bezugspunkt/Startposition 2. Achse (abs)
9	α0	<_STA>	REAL	Drehwinkel zur 1.Achse
10	DZ	<_MID>	REAL	maximale Zustelltiefe (ink), nur für Wirbelfräsen
11	DXY	<_MIDA>	REAL	maximale Ebenenzustellung, siehe <_AMODE>
12	UXY	<_FAL>	REAL	Schlichtaufmaß Ebene
13	UZ	<_FALD>	REAL	Schlichtaufmaß Tiefe
14	F	<_FFP1>	REAL	Vorschub
15		<_CDIR>	INT	Fräsrichtung EINER: 0 = Gleichlauf 1 = Gegenlauf 4 = Wechselweise
16		<_VARI>	INT	Bearbeitung EINER: 1 = Schruppen 2 = Schlichten 3 = Schlichten Boden 4 = Schlichten Rand 5 = Vorschlichten 6 = Anfasen ZEHNER: reserviert HUNDERTER: reserviert TAUSENDER: 1 = Wirbelfräsen 2 = Tauchfräsen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
17		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)		
				EINER:	reserviert	
				ZEHNER:	reserviert	
				HUNDERTER:	Auswahl Bearbeitung/nur Startpunktbe- rechnung	
				1 =	normale Bearbeitung	
				TAUSENDER:	Bemaßung über Mitte / Kante	
				0 =	Bemaßung über Mitte	
				1 =	Bemaßung über Kante "links" ("-" Richtung der 1.Achse)	
2 =	Bemaßung über Kante "rechts" ("+" Richtung der 1.Achse)					
18		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/G18/G19	
				0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene ak- tiv	
				1 =	G17 (nur im Zyklus aktiv)	
				2 =	G18 (nur im Zyklus aktiv)	
				3 =	G19 (nur im Zyklus aktiv)	
				ZEHNER:	---	reserviert
				HUNDERTER:	---	reserviert
				TAUSENDER:	---	reserviert
				ZEHNTAUSENDER:	Technologieskalierung innerhalb von Zy- klenmasken (Seite 734)	
0 =	Eingabe: komplett					
1 =	Eingabe: einfach					
19		<_AMODE>	INT	Alternativmodus		
				EINER:	Nuttiefe Z1	
				0 =	absolut	
				1 =	inkrementell	
				ZEHNER:	Einheit für Ebenenzustellung (<_MIDA> DXY	
				0 =	mm	
				1 =	% vom Werkzeugdurchmesser	
				HUNDERTER:	Eintauchtiefe bei Anfasen ZFS	
0 =	absolut					
1 =	inkrementell					
20		<_UMODE>	INT	reserviert		
21	FS	<_FS>	REAL	Fasenbreite (ink)		
22	ZFS	<_ZFS>	REAL	Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE>		

19.1.40 CYCLE930 - Einstich

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	X0	<_SPD>	REAL	Bezugspunkt in der Planachse (immer Durchmesser)
2	Z0	<_SPL>	REAL	Bezugspunkt in der Längsachse
3	B1	<_WIDG>	REAL	Einstichbreite unten
4	B2	<_WIDG2>	REAL	Einstichbreite oben (nur für die Oberfläche)
5	T1	<_DIAG>	REAL	Einstichtiefe am Bezugspunkt, bei abs und Längsbearbeitung = Durchmesser, sonst ink
6	T2	<_DIAG2>	REAL	Einstichtiefe gegenüber dem Bezugspunkt (nur für die Oberfläche), bei abs und Längsbearbeitung = Durchmesser, sonst ink
7	$\alpha 0$	<_STA>	REAL	Winkel der Schräge ($-180 \leq <_STA> \leq 180$)
8	$\alpha 1$	<_ANG1>	REAL	Flankenwinkel 1 ($0 \leq <_ANG1> < 90$) an der durch den Bezugspunkt bestimmten Seite des Einstichs
9	$\alpha 2$	<_ANG2>	REAL	Flankenwinkel 2 ($0 \leq <_ANG2> < 90$) gegenüber dem Bezugspunkt
10	R1/FS1	<_RCO1>	REAL	Verrundungsradius oder Fasenbreite 1, außen am Bezugspunkt
11	R2/FS2	<_RCI1>	REAL	Verrundungsradius oder Fasenbreite 2, innen am Bezugspunkt
12	R3/FS3	<_RCI2>	REAL	Verrundungsradius oder Fasenbreite 3, innen gegenüber dem Bezugspunkt
13	R4/FS4	<_RCO2>	REAL	Verrundungsradius oder Fasenbreite 4, außen gegenüber dem Bezugspunkt
14	U	<_FAL>	REAL	Schlichtaufmaß in X und Z, siehe <_VARI> (ZEHNTAUSENDER) (ohne Vorzeichen einzugeben)
15	D	<_IDEP1>	REAL	maximale Tiefenzustellung beim Eintauchen (ohne Vorzeichen einzugeben)
				0 = 1. Schnitt direkt auf ganze Tiefe
				> 0 = 1. Schnitt <_IDEP1>, 2. Schnitt $2 \cdot <_IDEP1>$ usw.
16	SC	<_SDIS>	REAL	Sicherheitsabstand (ohne Vorzeichen einzugeben)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
17		<_VARI>	INT	Bearbeitungsart		
				EINER:	reserviert	
				ZEHNER:	Technologische Bearbeitung	
					1 =	Schruppen
					2 =	Schlichten
				HUNDETER:	3 =	Schruppen und Schlichten
					Lage Längs/Plan Außen/Innen +Z/+Z bzw. +X/-X	
					1 =	Längs/Außen +Z
					2 =	Plan/Innen -X
					3 =	Längs/Innen +Z
					4 =	Plan/Innen +X
				TAUSENDER:	5 =	Längs/Außen -Z
					6 =	Plan/Außen -X
					7 =	Längs/Innen -Z
ZEHNTAUSENDER:	8 =	Plan/Außen +X				
	Lage Bezugspunkt					
0 =	Bezugspunkt oben					
1 =	Bezugspunkt unten					
Definition wie Schlichtaufmaße wirken						
0 =	Schlichtaufmaß U konturparallel					
1 =	Schlichtaufmaße UX und UZ getrennt					
18		<_DN>	INT	D-Nummer für 2. Schneide des Werkzeugs		
				> 0 =	D-Nummer für Werkzeugkorrektur der 2. Schneide des Einstechers	
				0 =	keine 2. Schneide programmiert	
19	N	<_NUM>	INT	Anzahl der Einstiche (0 = 1 Einstich)		
20	DP	<_DBH>	REAL	Abstand der Einstiche (nur erforderlich, wenn <_NUM> > 1)		
21	F	<_FF1>	REAL	Vorschub		
22		<_NR>	INT	Kennung für die Einstichform entspricht den Vertikal-Softkeys zur Auswahl der Form		
				0 =	90°-Flanken ohne Fasen/Verrundungen	
				1 =	schräge Flanken mit Fasen/Verrundungen (ohne α0)	
				2 =	wie 1, aber auf Kegel (mit α0)	
23	UX	<_FALX>	REAL	Schlichtaufmaß in X Achse, siehe <_VARI> (ZEHNTAUSENDER) (ohne Vorzeichen einzugeben)		
24	UZ	<_FALZ>	REAL	Schlichtaufmaß in Z Achse, siehe <_VARI> (ZEHNTAUSENDER) (ohne Vorzeichen einzugeben)		

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
25		<_DMODE>	INT	Anzeigemodus	
				EINER:	Bearbeitungsebene G17/G18/G19
				0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
				1 =	G17 (nur im Zyklus aktiv)
				2 =	G18 (nur im Zyklus aktiv)
3 =	G19 (nur im Zyklus aktiv)				
26		<_AMODE>	INT	Alternativmodus	
				EINER:	Bemaßung Tiefe (nur für die Oberfläche)
				0 =	am Bezugspunkt
				1 =	gegenüber dem Bezugspunkt
				ZEHNER:	Tiefe
				0 =	absolut
				1 =	inkrementell
				HUNDERTER:	Bemaßung Breite (nur für die Oberfläche)
				0 =	am Außendurchmesser (oben)
				1 =	am Innendurchmesser (unten)
				TAUSENDER:	Radius/Fase 1 (<_RCO1>)
				0 =	Radius
				1 =	Fase
				ZEHNTAUSENDER:	Radius/Fase 2 (<_RCI1>)
				0 =	Radius
				1 =	Fase
				HUNDERTTAUSENDER:	Radius/Fase 3 (<_RCI2>)
				0 =	Radius
				1 =	Fase
				EINEMILLION:	Radius/Fase 4 (<_RCO2>)
0 =	Radius				
1 =	Fase				

19.1.41 CYCLE940 - Freistich Formen

Mit dem Zyklus CYCLE940 können verschiedene Freistiche programmiert werden. Diese unterscheiden sich in der Parametrierung zum Teil erheblich.

Die zusätzlichen Spalten in der Tabelle zeigen, welcher Parameter bei welcher Freistichform benötigt wird. Sie entsprechen den vertikalen Auswahlsoftkeys in der Zyklenmaske:

- E: Freistich Form E
- F: Freistich Form F
- A-D: Freistich Gewinde DIN (Formen A-D)
- T: Freistich Gewinde (freie Definition der Form)

Syntax

CYCLE940 (<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>, <_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>, <_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Progr. bei Form				Bedeutung		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	Bezugspunkt in der Planachse (immer Durchmesser)		
2	Z0	<_SPL>	REAL	x	x	x	x	Bezugspunkt Längsachse (abs)		
3	FORM	<_FORM>	CHAR	x	x	x	x	Form des Freistichs (Großbuchstaben, z.B. "T") Auswahl, aus welcher Tabelle die Freistichwerte genommen werden sollen		
								A =	außen, Bezug DIN76, A = normal	
								B =	außen, Bezug DIN76, B = kurz	
								C =	innen, Bezug DIN76, C = normal	
								D =	innen, Bezug DIN76, D = kurz	
								E =	Bezug DIN509	
								F =	Bezug DIN509	
T =	Freiform									
4	LAGE	<_LAGE>	INT	x	x	x	x	Lage des Freistichs (parallel Z)		
								0 =	Außen +Z: ___	
								1 =	Außen -Z: ___/	
								2 =	Innen +Z: /-----	
3 =	Innen -Z: -----\									
5	SC	<_SDIS>		x	x	x	x	Sicherheitsabstand (ink)		
6	F	<_FFP>		x	x	x	x	Bearbeitungsvorschub (mm/U)		
7		<_VARI>	INT	-	-	x	x	Bearbeitungsart		
								EINER:	Bearbeitung	
									1 =	Schruppen
									2 =	Schichten
								ZEHNER:	Bearbeitungsstrategie	
									0 =	konturparallel
1 =	längs									
Freistichform E und F werden immer in einem Zug wie Schichten bearbeitet.										
8	X1	<_EPD>		x	x	-	-	Aufmaß X (abs/ink), siehe <_AMODE>		
								-	-	-
9	Z1	<_EPL>		-	x	-	-	Aufmaß Z		
								-	-	-

Nr.	Parameter Maske	Parameter intern	Datentyp	Progr. bei Form				Bedeutung	
10	R1	<_R1>		-	-	-	x	Verrundungsradius an der Schrägen	
11	R2	<_R2>		-	-	-	x	Verrundungsradius in der Ecke	
12	α	<_STA>		-	-	x	x	Eintauchwinkel	
13	VX	<_VRT>		x	x	-	-	Planzug X (abs/ink), siehe <_AMODE>	
				-	-	x	x	Planzug X bei Schlichten, (abs/ink), siehe <_AMODE>	
14	D	<_MID>		-	-	x	x	Tiefenzustellung	
15	U	<_FAL>		-	-	x	x	Schlichtaufmaß konturparallel, siehe <_AMODE>	
16	UX	<_FALX>		-	-	x	x	Schlichtaufmaß X	
17	UZ	<_FALZ>		-	-	x	x	Schlichtaufmaß Z	
18	P	<_PITI>	INT	-	-	x	-	Auswahl der Steigung, Form A-D, entspricht M1 ... M68	
								0 = 0.20 6 = 0.50 12 = 1.25 18 = 3.50 1 = 0.25 7 = 0.60 13 = 1.50 19 = 4.00 2 = 0.30 8 = 0.70 14 = 1.75 20 = 4.50 3 = 0.35 9 = 0.75 15 = 2.00 21 = 5.00 4 = 0.40 10 = 0.80 16 = 2.50 22 = 5.50 5 = 0.45 11 = 1.00 17 = 3.00 23 = 6.00	
				x	x	-	-	Auswahl Radius/Tiefe, Form E, F	
								0 = 0.6 · 0.3 4 = 2.5 · 0.4 8 = 0.1 · 0.1 1 = 1.0 · 0.4 5 = 4.0 · 0.5 9 = 0.2 · 0.1 2 = 1.0 · 0.2 6 = 0.4 · 0.2 3 = 1.6 · 0.3 7 = 0.6 · 0.2	
19		<_PTAB>	STRING [5]					String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") (nur für Oberfläche)	
20		<_PTABA>	STRING [20]					String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...) (nur für die Oberfläche)	
21		<_DMODE>	INT					Anzeigemodus	
				x	x	x	x	EINER:	Bearbeitungsebene G17/G18/G19
									0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv
									1 = G17 (nur im Zyklus aktiv)
									2 = G18 (nur im Zyklus aktiv)
					3 = G19 (nur im Zyklus aktiv)				

Nr.	Parameter Maske	Parameter intern	Datentyp	Progr. bei Form	Bedeutung	
22		<_AMODE>	INT	Alternativmodus		
				x x - x	EINER:	Parameter <_EPD> Aufmaß X oder Freistichtiefe
						0 = absolut (immer Durchmesser)
						1 = inkrementell
				x x - x	ZEHNER:	Parameter <_EPL> Aufmaß Z oder Freistichbreite
						0 = absolut
						1 = inkrementell
				x x x x	HUNDETER:	Parameter <_VRT> Planzug X
						0 = absolut (immer Durchmesser)
						1 = inkrementell
				- - x x	TAUSENDER:	Schlichtaufmaß
						0 = Schlichtaufmaß konturparallel (<_FAL>)
		= Schlichtaufmaß getrennt (<_FALX>/<_FALZ>)				

19.1.42 CYCLE951 - Abspannen

Syntax

CYCLE951 (<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>, <_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>, <_FF1>, <_NR>, <_DMODE>, <_AMODE>)

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	X0	<_SPD>	REAL	Bezugspunkt (abs, immer Durchmesser)
2	Z0	<_SPL>	REAL	Bezugspunkt (abs)
3	X1	<_EPD>	REAL	Endpunkt
4	Z1	<_EPL>	REAL	Endpunkt
5	XM α1 α2	<_ZPD>	REAL	Zwischenpunkt, siehe <_DMODE> (ZEHNER)
6	ZM α1 α2	<_ZPL>	REAL	Zwischenpunkt, siehe <_DMODE> (ZEHNER)

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
7	Lage	<_LAGE>	INT	Lage der Abspanecke	
				0 = außen/hinten	
				1 = außen/vorne	
				2 = innen/hinten	
				3 = innen/vorne	
8	D	<_MID>	REAL	maximale Tiefenzustellung beim Eintauchen	
9	UX	<_FALX>	REAL	Schlichtaufmaß in X	
10	UZ	<_FALZ>	REAL	Schlichtaufmaß in Z	
11		<_VARI>	INT	Bearbeitungsart	
				EINER:	Abspanrichtung (längs oder plan) im Koordinatensystem
				1 =	längs
				2 =	plan
				ZEHNER:	
				1 =	Schuppen bis Schlichtaufmaß
				2 =	Schichten
				HUNDERTER:	reserviert
TAUSENDER:	reserviert				
ZEHNTAUSENDER:	reserviert				
12	R1/FS1	<_RF1>	REAL	Verrundungsradius oder Fasenbreite 1, siehe <_AMODE> (ZEHNTAUSENDER)	
13	R2/FS2	<_RF2>	REAL	Verrundungsradius oder Fasenbreite 2, siehe <_AMODE> (HUNDERTTAUSENDER)	
14	R3/FS3	<_RF3>	REAL	Verrundungsradius oder Fasenbreite 3, siehe <_AMODE> (EINEMILLION)	
15	SC	<_SDIS>	REAL	Sicherheitsabstand	
16	F	<_FF1>	REAL	Vorschub für Schuppen/Schichten	
17		<_NR>	INT	Kennung der Abspanart (entspricht den Vertikal-Softkeys zur Auswahl der Form):	
				0 =	Abspanen 1, 90 Grad Ecke ohne Fasen/Verrundungen
				1 =	Abspanen 2, 90 Grad Ecke mit Fasen/Verrundungen
				2 =	Abspanen 3, beliebige Ecke mit Fasen/Verrundungen

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
18		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/G18/G19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufwurf wirksame Ebene aktiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
					3 =	G19 (nur im Zyklus aktiv)
				ZEHNER:	Eingabeform <_ZPD>/<_ZPL>	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
5 =	α1/α2					
21		<_AMODE>	INT	Alternativmodus		
				EINER:	Zwischenpunkt in X	
					0 =	absolut, Wert der Planachse im Durchmesser
					1 =	inkrementell, Wert der Planachse im Radius
				ZEHNER:	Zwischenpunkt in Z	
					0 =	absolut
					1 =	inkrementell
				HUNDETER:	Endpunkt in X	
					0 =	absolut, Wert der Planachse im Durchmesser
					1 =	inkrementell, Wert der Planachse im Radius
				TAUSENDER:	Endpunkt in Z	
					0 =	absolut
					1 =	inkrementell
				ZEHNTAUSENDER:	Radius/Fase 1	
					0 =	Radius
					1 =	Fase
				HUNDERTTAUSENDER:	Radius/Fase 2	
					0 =	Radius
					1 =	Fase
				EINEMILLION:	Radius/Fase 3	
					0 =	Radius
					1 =	Fase

19.1.43 CYCLE952 - Konturstechen

Syntax

```
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>)
```

Parameter

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung
1	PRG	<_PRG>	STRING[100]]	Name des Abspanprogramms
2	CON	<_CON>	STRING[100]]	Name des Programms, aus dem die aktualisierte Rohteilkontur gelesen wird (bei Restbearbeitung)
3	CONR	<_CONR>	STRING[100]]	Name des Programms, in das die aktualisierte Rohteilkontur (siehe <_AMODE> ZEHNTAUSENDER) geschrieben wird

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
4		<_VARI>	INT	Bearbeitungsart	
				EINER:	Abspanart
					1 = längs
					2 = plan
					3 = konturparallel
				ZEHNER:	Technologische Bearbeitung (siehe <_GMODE> HUNDERTER)
					1 = Schruppen
					2 = Schichten
					3 = reserviert
					4 = Schruppen zweikanalig
					5 = Schichten zweikanalig
				HUNDERTER:	Bearbeitungsrichtung
					1 = Bearbeitungsrichtung X -
					2 = Bearbeitungsrichtung X +
					3 = Bearbeitungsrichtung Z -
					4 = Bearbeitungsrichtung Z +
				TAUSENDER:	Zustellrichtung
					1 = Außen X -
					2 = Innen X +
					3 = Stirnseite Z -
					4 = Rückseite Z +
ZEHNTAUSENDER:	Definition, wie Schlichtaufmaße wirken				
	0 = Schlichtaufmaße UX und UZ getrennt				
	1 = Schlichtaufmaß U konturparallel				
HUNDERTTAUSENDER:	Nachziehen				
	0 = Kompatibilität, Nachziehen automatisch				
	1 = mit Nachziehen an der Kontur				
	2 = ohne Nachziehen				
	3 = Nachziehen automatisch				
EINEMILLION:	Hinterschnitte				
	0 = Stelle wird nicht ausgewertet bei Stechen, - Rest und Stechdrehen, - Rest				
	1 = Hinterschnitte bearbeiten				
	2 = Hinterschnitte nicht bearbeiten				
ZEHNMILLIONEN:	hintervor Drehmitte				
	0 = Bearbeitung vor der Drehmitte				
	1 = reserviert				
5	F	<_F>	REAL	Vorschub für Schruppen/Schichten	
	FZ			Vorschub Abszisse Stechdrehen	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
6	FR	<_FR>	REAL	Vorschub zum Eintauchen in Hinterschnitte Schruppen	
	FX			Vorschub Ordinate Stechdrehen	
7	RP	<_RP>	REAL	Rückzugsebene bei Innenbearbeitung (abs., immer Durchmesser)	
8	D	<_D>	REAL	Zustellung Schruppen (siehe <_AMODE> EINER)	
9	DX	<_DX>	REAL	Zustellung X (siehe <_AMODE> EINER)	
10	DZ	<_DZ>	REAL	Zustellung Z (siehe <_AMODE> EINER)	
11	UX	<_UX>	REAL	Schlichtaufmaß X (siehe <_VARI> ZEHNTAUSENDER)	
12	UZ	<_UZ>	REAL	Schlichtaufmaß Z (siehe <_VARI> ZEHNTAUSENDER)	
13	U	<_U>	REAL	Schlichtaufmaß konturparallel (siehe <_VARI> ZEHNTAUSENDER)	
14	U1	<_U1>	REAL	zusätzliches Schlichtaufmaß beim Schlichten (siehe <_AMODE> TAU-SENDER)	
15	BL	<_BL>	INT	Rohteildefinition	
				1 =	Zylinder mit Aufmaß
				2 =	Aufmaß auf Fertigteilkontur
3 =	Rohteilkontur ist angegeben				
16	XD	<_XD>	REAL	Rohteildefinition X (siehe <_AMODE> HUNDERTTAUSENDER)	
17	ZD	<_ZD>	REAL	Rohteildefinition Z (siehe <_AMODE> EINEMILLION)	
18	XA	<_XA>	REAL	Grenze 1 X (abs., immer Durchmesser)	
19	ZA	<_ZA>	REAL	Grenze 1 Z (abs.)	
20	XB	<_XB>	REAL	Grenze 2 X (siehe <_AMODE> ZEHNMILLIONEN)	
21	ZB	<_ZB>	REAL	Grenze 2 Z (siehe <_AMODE> HUNDERTMILLIONEN)	
22	XDA	<_XDA>	REAL	Einstechgrenze 1 für die 1. Einstichposition auf der Stirnseite (abs., immer Durchmesser)	
23	XDB	<_XDB>	REAL	Einstechgrenze 2 für die 1. Einstichposition auf der Stirnseite (abs., immer Durchmesser)	
24	N	<_N>	INT	Anzahl der Einstiche	
25	DP	<_DP>	REAL	Abstand der Einstiche	
				Längs-Einstich: parallel zur Z-Achse Plan-Einstich: parallel zur X-Achse	
26	DI	<_DI>	REAL	Abstand für Vorschubunterbrechung	
				0 = keine Unterbrechung > 0 = mit Unterbrechung	
27	SC	<_SC>	REAL	Sicherheitsabstand zum Umfahren von Hindernissen, inkremental	
28	D2	<_DN>	INT	D-Nummer für 2. Schneide, wenn nicht programmiert ⇒ D+1	

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
29		<_GMODE>	INT	Geometriemodus (Bewertung der programmierten Geometriewerte)		
				EINER:	re- ser- viert	
				ZEHNER:	re- ser- viert	
				HUNDERTER:	Auswahl Bearbeitung / nur Startpunktbe- rechnung	
					0 =	normale Bearbeitung (kein Kom- patibilitätsmodus erforderlich)
					1 =	normale Bearbeitung
					2 =	Startposition berechnen - keine Bearbeitung (nur für Aufruf aus ShopMill/ShopTurn)
				TAUSENDER:	Eingrenzung	
					0 =	nein
					1 =	ja
				ZEHNTAUSENDER:	Grenze 1 X eingeben	
					0 =	nein
					1 =	ja
				HUNDERTTAUSENDER:	Grenze 2 X eingeben	
					0 =	nein
					1 =	ja
				EINEMILLION:	Grenze 1 Z eingeben	
0 =	nein					
1 =	ja					
ZEHNMILLIONEN:	Grenze 2 Z eingeben					
	0 =	nein				
	1 =	ja				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung		
30		<_DMODE>	INT	Anzeigemodus		
				EINER:	Bearbeitungsebene G17/G18/G19	
					0 =	Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv
					1 =	G17 (nur im Zyklus aktiv)
					2 =	G18 (nur im Zyklus aktiv)
				ZEHNER:	3 =	G19 (nur im Zyklus aktiv)
					Technologiemodus	
					1 =	Konturabspannen
				HUNDERTER:	2 =	Konturstechen
					3 =	Stechdrehen
					Restmaterial bearbeiten	
				TAUSENDER:	0 =	nein
					1 =	ja
				ZEHNTAUSENDER:	---	reserviert
ZEHNTEUSENDER:	Technologieskalierung innerhalb von Zyklenmasken (Seite 734)					
	0 =	Eingabe: komplett				
	1 =	Eingabe: einfach				

Nr.	Parameter Maske	Parameter intern	Datentyp	Bedeutung	
31		<_AMODE>	INT	Alternativmodus	
				EINER:	Auswahl Zustellung
					0 = Zustellung DX und DZ bei Abspanart konturparallel
				1 = Zustellung D	
				ZEHNER:	Zustellstrategie
					0 = variable Schnitttiefe (90 ... 100 %)
				1 = konstante Schnitttiefe	
				HUNDERTER:	Schnittaufteilung
					0 = gleichmäßig
				1 = an Kanten ausrichten	
				TAUSENDER:	Auswahl Konturaufmaß U1, Doppelschichten
					0 = nein
				1 = ja	
				ZEHNTAUSENDER:	Auswahl Rohteil aktualisieren
					0 = nein
				1 = ja	
				HUNDERTTAUSENDER:	Auswahl Aufmaß auf Rohteil XD
					0 = absolut, Wert der Planachse im Durchmesser
1 = inkremental, Wert der Planachse im Radius					
EINEMILLION:	Auswahl Aufmaß auf Rohteil ZD				
	0 = absolut				
1 = inkrementell					
ZEHNMILLIONEN:	Auswahl Grenze 2 XB				
	0 = absolut, Wert der Planachse im Durchmesser				
1 = inkremental, Wert der Planachse im Radius					
HUNDERTMILLION:	Auswahl Grenze 2 ZB				
	0 = absolut				
1 = inkrementell					
EINEMILLIARDE:					
	0 = Führungskanal				
1 = Folgekanal					
32		<_PK>	INT	Nummer des Partnerkanals, wenn mehr als 2 Kanäle an der Maschine vorhanden sind	
33	DCH	<_DCH>	REAL	Kanalversatz	
34	FS	<_FS>	REAL	Vorschub Schichten bei Komplettbearbeitung	

19.1.44 CYCLE4071 - Längsschleifen mit Zustellung am Umkehrpunkt

Syntax

```
CYCLE4071 (<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>, <S_A2>)
```

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_A>	REAL	Zustelltiefe am Anfang
2	<S_B>	REAL	Zustelltiefe am Ende
3	<S_W>	REAL	Schleifbreite
4	<S_U>	REAL	Ausfeuerzeit
5	<S_I>	REAL	Vorschub für Zustellung
6	<S_K>	REAL	Vorschub für Querststellung
7	<S_H>	INT	Anzahl Wiederholungen
8	<S_A1>	AXIS	Zustellachse (optional) bzw. 1. Geometrieachse
9	<S_A2>	AXIS	Pendelachse (optional) bzw. 2. Geometrieachse

Funktion

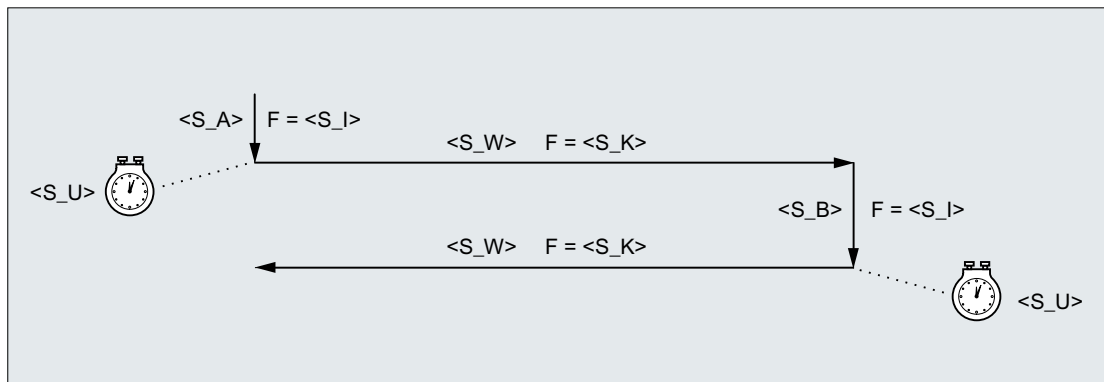
Der Zyklus dient zum Abarbeiten von <S_H> Zustellungen. Dabei kann die Zustelltiefe am Anfang und am Ende unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse.
2. Verfahren der Zustellachse auf die Zustelltiefe am Anfang P1 <S_A> mit dem Vorschub für Zustellung P5 <S_I>.
3. Ausfeuern mit der Ausfeuerzeit P4 <S_U>.
4. Verfahren der Pendelachse mit der Schleifbreite P3 <S_W> als Fahrweg und dem Vorschub für Querststellung P6 <S_K>.
5. Verfahren der Zustellachse auf die Zustelltiefe am Ende P2 <S_B> mit dem Vorschub für Zustellung P5 <S_I>.
6. Ausfeuern mit der Ausfeuerzeit P4 <S_U>.
7. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg auf den Anfangspunkt und dem Vorschub für Querststellung P6 <S_K>.

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

Der Ablauf wird entsprechend der programmierten Anzahl Wiederholungen P7 (<S_H>) wiederholt.



Beispiel

Zwei Pendelbewegung mit folgenden Zyklenparameter ausführen:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Vorschub Zustellung: 1 mm/min
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Programmcode

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

19.1.45 CYCLE4072 - Längsschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal

Syntax

```
CYCLE4072 (<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_GAUGE>	STRING	Abbruchbedingungen für Zustellung: 1. Nummer eines schnellen Eingangs 2. Logischer Ausdruck
2	<S_A>	REAL	Zustelltiefe am Anfang
3	<S_B>	REAL	Zustelltiefe am Ende
4	<S_W>	REAL	Schleifbreite
5	<S_U>	REAL	Ausfeuerzeit
6	<S_I>	REAL	Vorschub für Zustellung
7	<S_K>	REAL	Vorschub für Querststellung
8	<S_H>	INT	Anzahl Wiederholungen
9	<S_A1>	AXIS	Zustellachse (optional) bzw. 1. Geometrieachse
10	<S_A2>	AXIS	Pendelachse (optional) bzw. 2. Geometrieachse

Funktion

Der Zyklus dient zum Abarbeiten von <S_H> Zustellungen unter Berücksichtigung eines externen Abbruchsignals. Die Zustelltiefe kann am Anfang und am Ende unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung. Die Tiefenzustellung wird abgebrochen, wenn die Abbruchbedingung erfüllt ist. Nach dem Abbruch der Tiefenzustellung wird immer ein kompletter Hub durchgeführt.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse.
 2. Verfahren der Zustellachse auf die Zustelltiefe am Anfang P2 <S_A> mit dem Vorschub für Zustellung P6 <S_I>.
 3. Ausfeuern mit der Ausfeuerzeit P5 <S_U>.
 4. Verfahren der Pendelachse mit der Schleifbreite P4 <S_W> als Fahrweg und dem Vorschub für Querststellung P7 <S_K>.
 5. Verfahren der Zustellachse auf die Zustelltiefe am Ende P3 <S_B> mit dem Vorschub für Zustellung P6 <S_I>.
 6. Ausfeuern mit der Ausfeuerzeit P5 <S_U>.
 7. Verfahren der Pendelachse mit der Schleifbreite P4 <S_W> als Fahrweg auf den Anfangspunkt und dem Vorschub für Querststellung P7 <S_K>.
 8. Ohne Abbruch: Der oben beschriebene Ablauf wird so oft wiederholt, bis die programmierten Anzahl Wiederholungen P7 (<S_H>) erreicht ist.
Mit Abbruch: Die Bearbeitung wird Erreichen des nächsten Startpunkts beendet.
- Der Ablauf ist durch Einzelsatz nicht unterbrechbar.

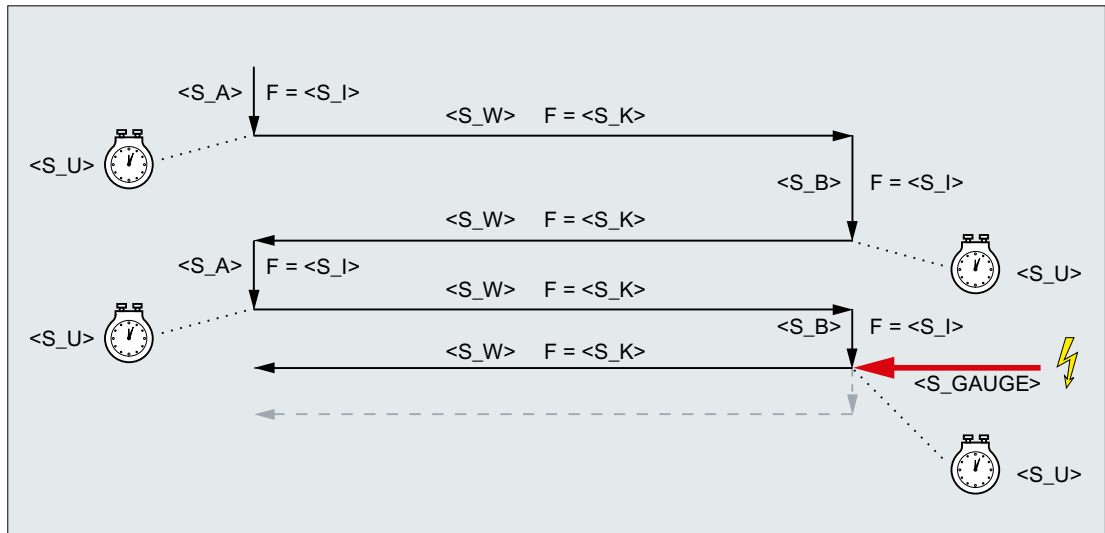


Bild 19-1 Abbruch der Zustellung am Ende

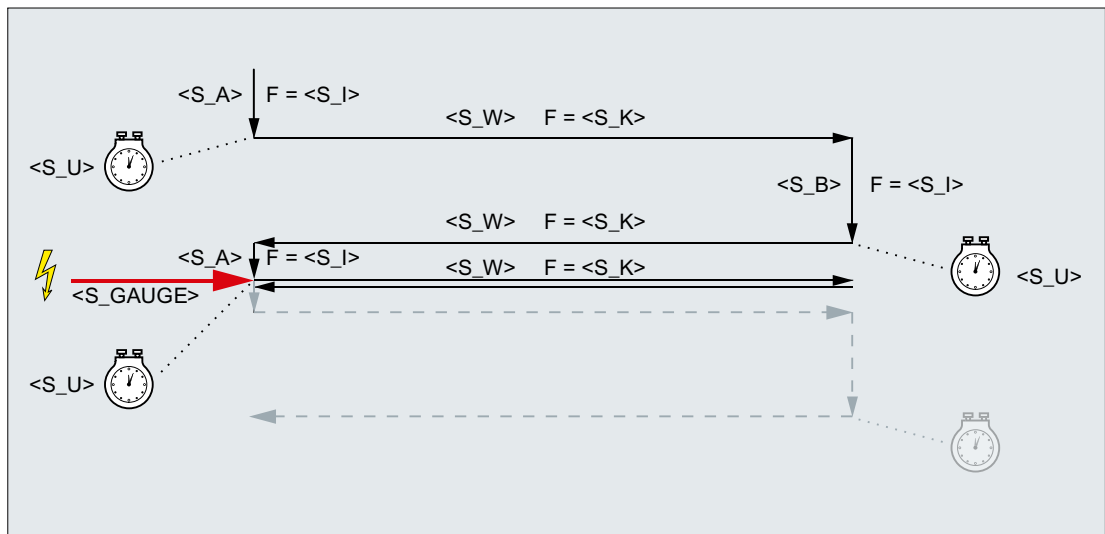


Bild 19-2 Abbruch der Zustellung am Anfang

Ressourcen

Als Ressourcen verwendet der Zyklus eine satzübergreifende Synchronaktion und eine Synchronaktionsvariable. Die Synchronaktion wird dynamisch aus dem freien Bereich des Synchronaktionsbandes ermittelt (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR - 1199 ...). Als Synchronaktionsvariable wird SYG_IS[1] verwendet.

Beispiele

Beispiel 1: Oszillieren mit zwei Hüben

Zyklusparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Vorschub Zustellung: 1 mm/min
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: schneller Eingang 1 (\$A_IN[1])

Programmcode

```
N10 T1 D1
N20 CYCLE4072 ("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

Beispiel 2: Oszillieren mit zwei Hüben

Zyklusparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Vorschub Zustellung: 1 mm/min
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: Variable \$A_DBR[20] < 0,01

Programmcode

```
N10 T1 D1
N20 CYCLE4072 ("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

19.1.46 CYCLE4073 - Längschleifen mit kontinuierlicher Zustellung

Syntax

CYCLE4073 (<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_A>	REAL	Zustelltiefe am Anfang
2	<S_B>	REAL	Zustelltiefe am Ende
3	<S_W>	REAL	Schleifbreite
4	<S_U>	REAL	Ausfeuerzeit
5	<S_K>	REAL	Vorschub für Querstellung
6	<S_H>	INT	Anzahl Wiederholungen
7	<S_A1>	AXIS	Zustellachse (optional) bzw. 1. Geometrieachse
8	<S_A2>	AXIS	Pendelachse (optional) bzw. 2. Geometrieachse

Funktion

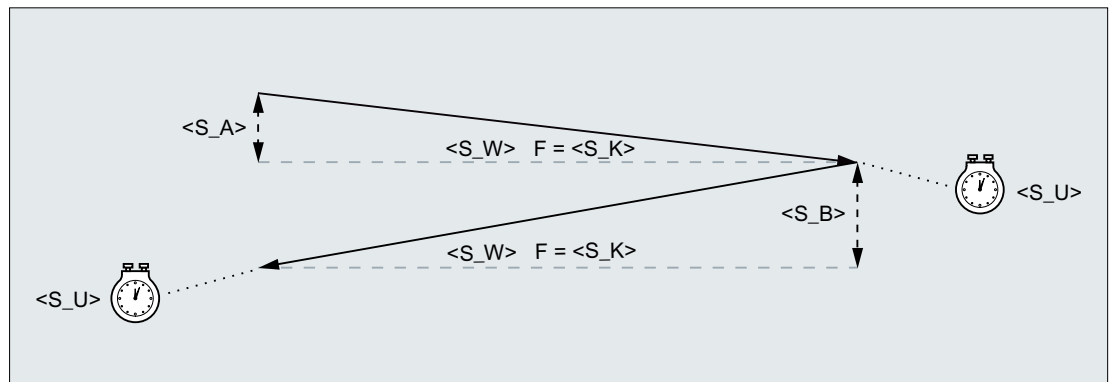
Der Zyklus dient zum Abarbeiten von <S_H> Zustellungen. Dabei kann die Zustellung vom Anfang zum Ende und vom Ende zum Anfang unterschiedlich sein.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse mit Zustelltiefe 0
2. Verfahren der Pendelachse mit der Schleifbreite P3 <S_W> als Fahrweg und Vorschub für Querstellung P5 <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Anfang P1 <S_A>.
3. Ausfeuern mit der Ausfeuerzeit P4 <S_U>.
4. Verfahren der Pendelachse mit der Schleifbreite P3 <S_W> als Fahrweg auf den Anfangspunkt und Vorschub für Querstellung P5 <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Ende P2 <S_B>.
5. Ausfeuern mit der Ausfeuerzeit P4 <S_U>.

Der Ablauf ist durch Einzelsatz nicht unterbrechbar.

Der Ablauf wird entsprechend der programmierten Anzahl Wiederholungen P7 (<S_H>) wiederholt.



Beispiel

Oszillieren mit zwei Hübungen

Zyklusparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Programmcode

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

19.1.47 CYCLE4074 - Längschleifen mit kontinuierlicher Zustellung und Abbruchsignal

Syntax

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_GAUGE>	STRING	Abbruchbedingungen für Zustellung: 1. Nummer eines schnellen Eingangs 2. Logischer Ausdruck
2	<S_A>	REAL	Zustelltiefe am Anfang
3	<S_B>	REAL	Zustelltiefe am Ende
4	<S_W>	REAL	Schleifbreite
5	<S_U>	REAL	Ausfeuerzeit
6	<S_K>	REAL	Vorschub für Querststellung
7	<S_H>	INT	Anzahl Wiederholungen
8	<S_A1>	AXIS	Zustellachse (optional) bzw. 1. Geometrieachse
9	<S_A2>	AXIS	Pendelachse (optional) bzw. 2. Geometrieachse

Funktion

Der Zyklus dient zum Abarbeiten von <S_H> Zustellungen unter Berücksichtigung z.B. eines externen Abbruchsignals. Die Zustelltiefe kann am Anfang und am Ende unterschiedlich sein. Die Tiefenzustellung wird abgebrochen, wenn die Abbruchbedingung erfüllt ist. Nach dem Abbruch der Tiefenzustellung wird immer ein kompletter Hub durchgeführt.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse mit Zustelltiefe 0
2. Verfahren der Pendelachse mit der Schleifbreite P4 <S_W> als Fahrweg und Vorschub für Querststellung P6 <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Anfang P2 <S_A>.
3. Ausfeuern mit der Ausfeuerzeit P5 <S_U>.
4. Verfahren der Pendelachse mit der Schleifbreite P4 <S_W> als Fahrweg auf den Anfangspunkt und Vorschub für Querststellung P6 <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Ende P3 <S_B>.
5. Ausfeuern mit der Ausfeuerzeit P5 <S_U>.
6. Ohne Abbruch: Der oben beschriebene Ablauf wird so oft wiederholt, bis die programmierten Anzahl Wiederholungen P7 (<S_H>) erreicht ist.
Mit Abbruch: Die Tiefenzustellung wird abgebrochen. Die Bearbeitung wird mit Erreichen des nächsten Startpunkts beendet.

Der Ablauf ist durch Einzelsatz nicht unterbrechbar.

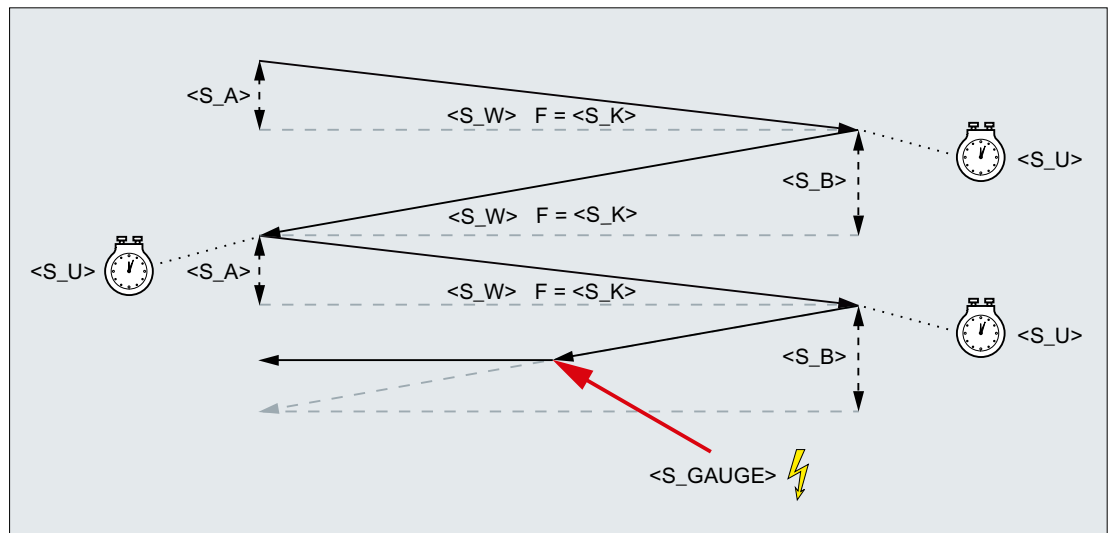


Bild 19-3 Abbruch der Zustellung vom Ende zum Anfang

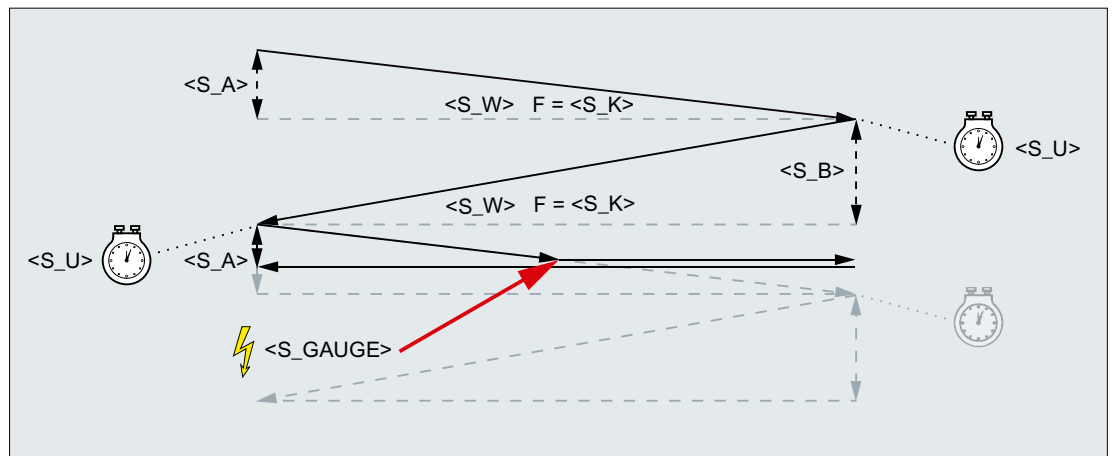


Bild 19-4 Abbruch der Zustellung vom Anfang zum Ende

Ressourcen

Als Ressourcen verwendet der Zyklus eine satzübergreifende Synchronaktion und eine Synchronaktionsvariable. Die Synchronaktion wird dynamisch aus dem freien Bereich des Synchronaktionsbandes ermittelt (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR - 1199 ...). Als Synchronaktionsvariable wird SYG_IS[1] verwendet.

Beispiele

Beispiel 1: Oszillieren mit zwei Hüben

Zyklusparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm

- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: schneller Eingang 1 (\$A_IN[1])

Programmcode

```
N10 T1 D1
N20 CYCLE4074 ("1", 0.02, 0.01, 100, 1, 1000, 2)
N30 M30
```

Beispiel 2: Oszillieren mit zwei Hüben

Zyklusparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: Variable \$A_DBR[20] < 0,01

Programmcode

```
N10 T1 D1
N20 CYCLE4074 ("($A_DBR[20]<0.01)", 0.02, 0.01, 100, 1, 1000, 2)
N30 M30
```

19.1.48 CYCLE4075 - Flachsleifen mit Zustellung am Umkehrpunkt

Syntax

```
CYCLE4075 (<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>, <S_A2>)
```

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_I>	REAL	Zustelltiefe am Anfang
2	<S_J>	REAL	Zustelltiefe am Ende

Nr.	Parameter	Datentyp	Bedeutung
3	<S_K>	REAL	Gesamtzustelltiefe
4	<S_A>	REAL	Schleifbreite
5	<S_R>	REAL	Vorschub für Zustellung
6	<S_F>	REAL	Vorschub für Querstellung
7	<S_P>	REAL	Ausfeuerzeit
8	<S_A1>	AXIS	Zustellachse (optional)
9	<S_A2>	AXIS	Pendelachse (optional)

Funktion

Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe P3 <S_K> in Zustellschritten. Die Zustelltiefen am Anfang P1 <S_I> und am Ende P2 <S_J> können unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung.

Die Wegangaben P1 bis P4 können negativ oder positiv sein.

Die Angabe der Zustellachse P8 <S_A1> und/oder Pendelachse P9 <S_A2> sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

Ist die Summe aus Zustelltiefe am Anfang P1 <S_I> und am Ende P2 <S_J> gleich 0 bzw. die Gesamtzustelltiefe P3 <S_K> gleich 0 wird nur ein Ausfeuerhub durchgeführt.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse.
2. Verfahren der Zustellachse auf die Zustelltiefe am Anfang P1 <S_I> mit dem Vorschub für Zustellung P5 <S_R>.
3. Ausfeuern mit der Ausfeuerzeit P7 <S_P>
4. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg und dem Vorschub für Querstellung P6 <S_F>.
5. Verfahren der Zustellachse auf die Zustelltiefe am Ende P2 <S_J> mit dem Vorschub für Zustellung P5 <S_R>.
6. Ausfeuern mit der Ausfeuerzeit P7 <S_P>.
7. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg auf den Anfangspunkt und dem Vorschub für Querstellung P6 <S_F>.

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

Der Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe P3 <S_K> erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.

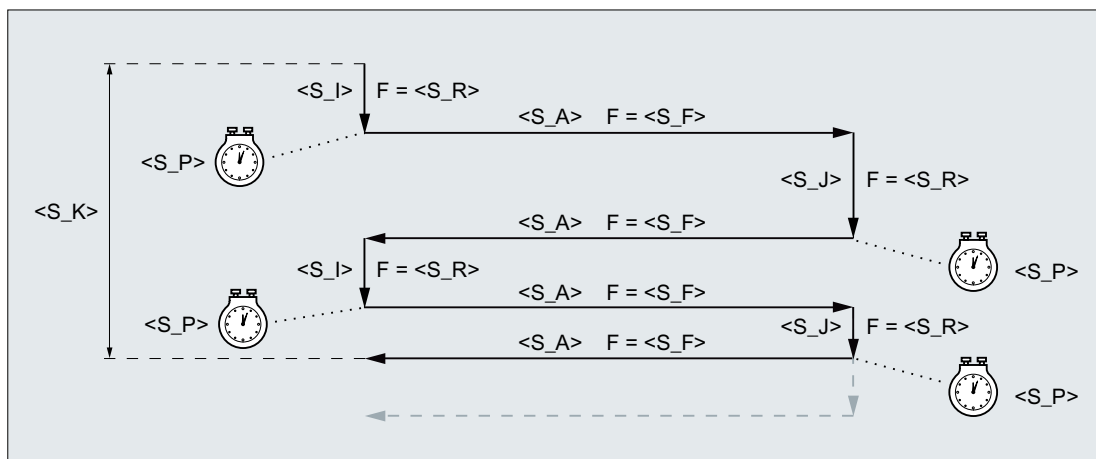


Bild 19-5 Gesamtzustelltiefe erreicht bei Zustellung am zweiten Umkehrpunkt

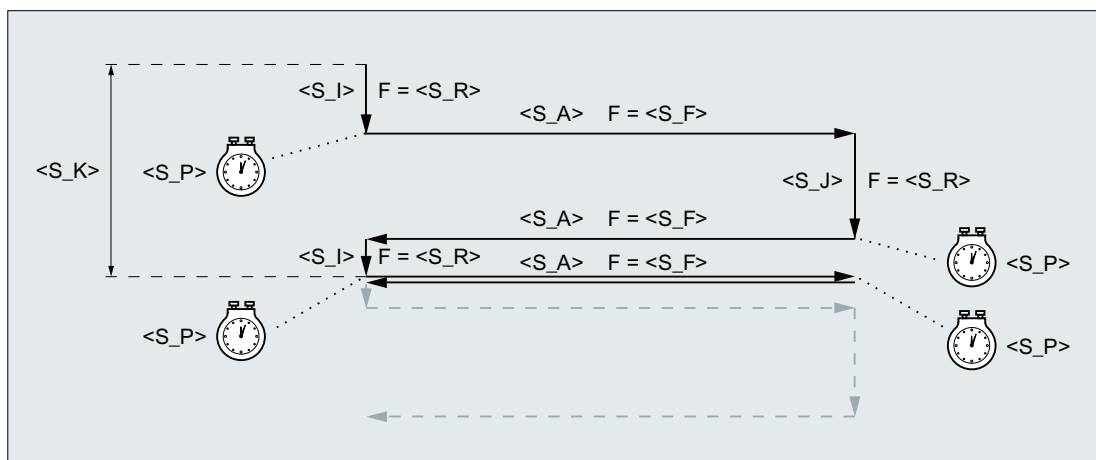


Bild 19-6 Gesamtzustelltiefe erreicht bei Zustellung am ersten Umkehrpunkt

Beispiel

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

Programmcode
N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30

19.1.49 CYCLE4077 - Flachsleifen mit Zustellung am Umkehrpunkt und Abbruchsignal

Syntax

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_GAUGE>	STRING	Abbruchbedingung für Zustellung: <ul style="list-style-type: none"> • Nummer eines schnellen Eingangs • Logischer Ausdruck
2	<S_I>	REAL	Zustelltiefe am Anfang
3	<S_J>	REAL	Zustelltiefe am Ende
4	<S_K>	REAL	Gesamtzustelltiefe
5	<S_A>	REAL	Schleifbreite
6	<S_R>	REAL	Vorschub für Zustellung
7	<S_F>	REAL	Vorschub für Querstellung
8	<S_P>	REAL	Ausfeuerzeit
9	<S_A1>	AXIS	Zustellachse (optional)
10	<S_A2>	AXIS	Pendelachse (optional)

Funktion

Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe P4 <S_K> in Zustellschritten. Die Zustelltiefen am Anfang P2 <S_I> und am Ende P3 <S_J> können unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung. Die Tiefenzustellung wird abgebrochen, wenn das Abbruchsignal des schnellen Eingangs gleich 1 oder die Abbruchbedingung erfüllt ist. Nach dem Abbruch wird ein kompletter Hub ausgeführt.

Die Wegangaben P2 bis P5 können negativ oder positiv sein.

Die Angabe der Zustellachse P9 <S_A1> und/oder Pendelachse P10 <S_A2> sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

Ist die Summe aus Zustelltiefe am Anfang P2 <S_I> und am Ende P3 <S_J> gleich 0 bzw. die Gesamtzustelltiefe P4 <S_K> gleich 0 wird nur ein Ausfeuerhub durchgeführt.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse.
2. Verfahren der Zustellachse auf die Zustelltiefe am Anfang P2 <S_I> mit dem Vorschub für Zustellung P6 <S_R>.
3. Ausfeuern mit der Ausfeuerzeit P8 <S_P>.
4. Verfahren der Pendelachse mit der Schleifbreite P5 <S_A> als Verfahrweg und dem Vorschub für Querststellung P7 <S_F>.
5. Verfahren der Zustellachse auf die Zustelltiefe am Ende P3 <S_J> mit dem Vorschub für Zustellung P6 <S_R>.
6. Ausfeuern mit der Ausfeuerzeit P8 <S_P>.
7. Verfahren der Pendelachse mit der Schleifbreite P5 <S_A> als Verfahrweg auf den Anfangspunkt und dem Vorschub für Querststellung P7 <S_F>.
8. Ohne Abbruch: Der oben beschriebene Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe P4 <S_K> erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.
Mit Abbruch: Die Bearbeitung wird am Startpunkt beendet.

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

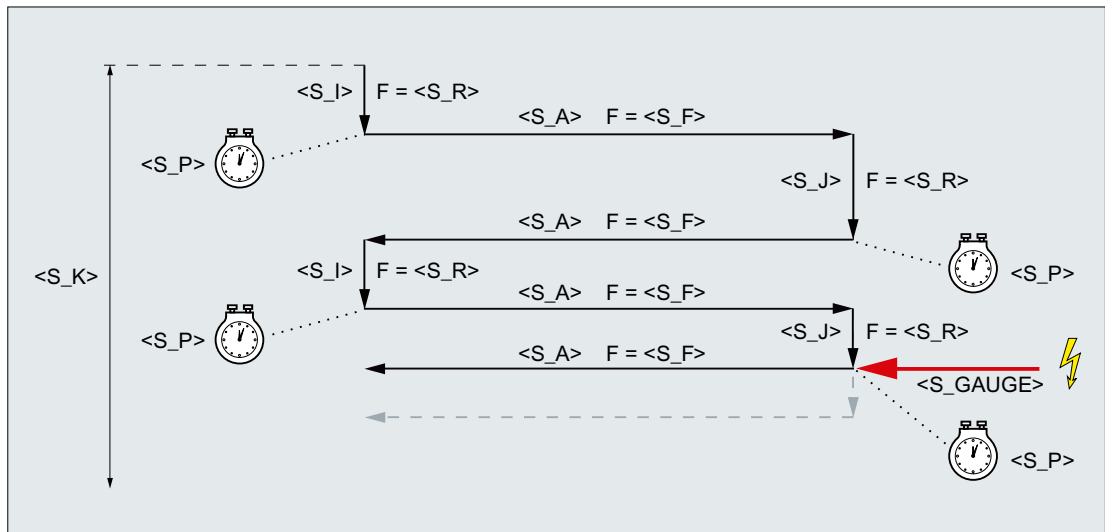


Bild 19-7 Abbruch der Zustellung am Ende

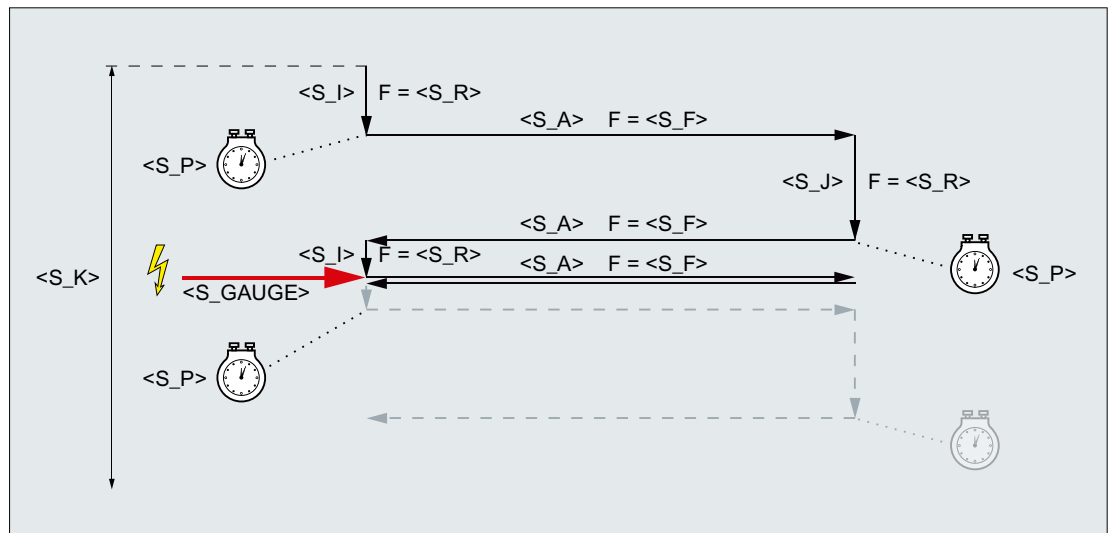


Bild 19-8 Abbruch der Zustellung am Anfang

Ressourcen

Als Ressourcen verwendet der Zyklus eine satzübergreifende Synchronaktion und eine Synchronaktionsvariable. Die Synchronaktion wird dynamisch aus dem freien Bereich des Synchronaktionsbandes ermittelt (CUS.DIR - 1 ..., CMA.DIR - 1000 ... , CST.DIR - 1199 ...). Als Synchronaktionsvariable wird SYG_IS[1] verwendet.

Beispiele

Beispiel 1

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

Abbruchsignal: schneller Eingang 1 (\$A_IN[1])

Programmcode

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

Beispiel 2

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

Abbruchsignal: Dualport-RAM-Variable 20 kleiner als 0,01 (\$A_DBR[20] < 0,01)

Programmcode

```
N10 T1 D1
N20 CYCLE4077 ("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
N30 M30
```

19.1.50 CYCLE4078 - Flachsleifen mit kontinuierlicher Zustellung

Syntax

CYCLE4078 (<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_I>	REAL	Zustelltiefe vom Anfang zum Ende
2	<S_J>	REAL	Zustelltiefe vom Ende zum Anfang
3	<S_K>	REAL	Gesamtzustelltiefe
4	<S_A>	REAL	Schleifbreite
5	<S_F>	REAL	Vorschub
6	<S_P>	REAL	Ausfeuerzeit
7	<S_A1>	AXIS	Zustellachse (optional)
8	<S_A2>	AXIS	Pendelachse (optional)

Funktion

Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe P3 <S_K> mittels kontinuierlicher Zustellung. Die Zustelltiefen vom Anfang zum Ende P1 <S_I> und vom Ende zum Anfang P2 <S_J> können unterschiedlich sein.

Die Wegangaben P1 bis P4 können negativ oder positiv sein.

Die Angabe der Zustellachse P8 <S_A1> und/oder Pendelachse P9 <S_A2> sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

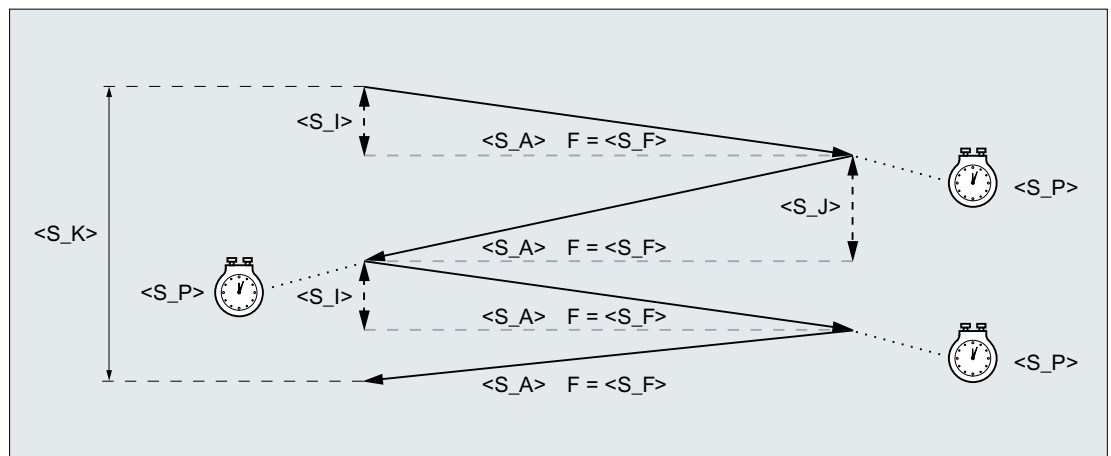
Ist die Summe der Zustelltiefen P1 <S_I> und P2 <S_J> gleich 0 bzw. die Gesamtzustelltiefe P3 <S_K> gleich 0 wird nur ein Ausfeuerhub durchgeführt.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse mit Zustelltiefe 0
2. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg und Vorschub P5 <S_F> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Anfang P1 <S_I>.
3. Ausfeuern mit der Ausfeuerzeit P7 <S_P>
4. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg auf den Anfangspunkt und Vorschub P5 <S_F> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Ende P2 <S_J>.
5. Ausfeuern mit der Ausfeuerzeit P7 <S_P>.
6. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg auf den Anfangspunkt und Vorschub P5 <S_F>.

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

Der Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe P3 <S_K> erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.



Beispiel

Oszillieren mit:

- 20 mm Zustelltiefe am Anfang
- 10 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 100 mm
- Hub 100 mm

- Vorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

```

Programmcode
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
    
```

19.1.51 CYCLE4079 - Flachschleifen mit intermittierender Zustellung

Syntax

CYCLE4079 (<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>, <S_A2>)

Parameter

Nr.	Parameter	Datentyp	Bedeutung
1	<S_I>	REAL	Zustelltiefe am Anfang
2	<S_J>	REAL	Zustelltiefe am Ende
3	<S_K>	REAL	Gesamtzustelltiefe
4	<S_A>	REAL	Schleifbreite
5	<S_R>	REAL	Vorschub für Zustellung
6	<S_F>	REAL	Vorschub für Querstellung
7	<S_P>	REAL	Ausfeuerzeit
8	<S_A1>	AXIS	Zustellachse (optional)
9	<S_A2>	AXIS	Pendelachse (optional)

Funktion

Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe P3 <S_K> in Zustellschritten. Die Zustelltiefen am Anfang P1 <S_I> und am Ende P2 <S_J> können unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung.

Die Wegangaben P1 bis P4 können negativ oder positiv sein.

Die Angabe der Zustellachse P8 <S_A1> und/oder Pendelachse P9 <S_A2> sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

Ist die Summe aus Zustelltiefe am Anfang P1 <S_I> und am Ende P2 <S_J> gleich 0 bzw. die Gesamtzustelltiefe P3 <S_K> gleich 0 wird nur ein Ausfeuerhub durchgeführt.

Ablauf

1. Start des Zyklus an der aktuellen Position der Pendelachse.
2. Verfahren der Zustellachse auf die Zustelltiefe am Anfang P1 <S_I> mit dem Vorschub für Zustellung P5 <S_R>.
3. Ausfeuern mit der Ausfeuerzeit P7 <S_P>
4. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg und dem Vorschub für Querstellung P6 <S_F>.
5. Verfahren der Zustellachse auf die Zustelltiefe am Ende P2 <S_J> mit dem Vorschub für Zustellung P5 <S_R>.
6. Ausfeuern mit der Ausfeuerzeit P7 <S_P>.
7. Verfahren der Pendelachse mit der Schleifbreite P4 <S_A> als Fahrweg auf den Anfangspunkt und dem Vorschub für Querstellung P6 <S_F>.

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

Der Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe P3 <S_K> erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.

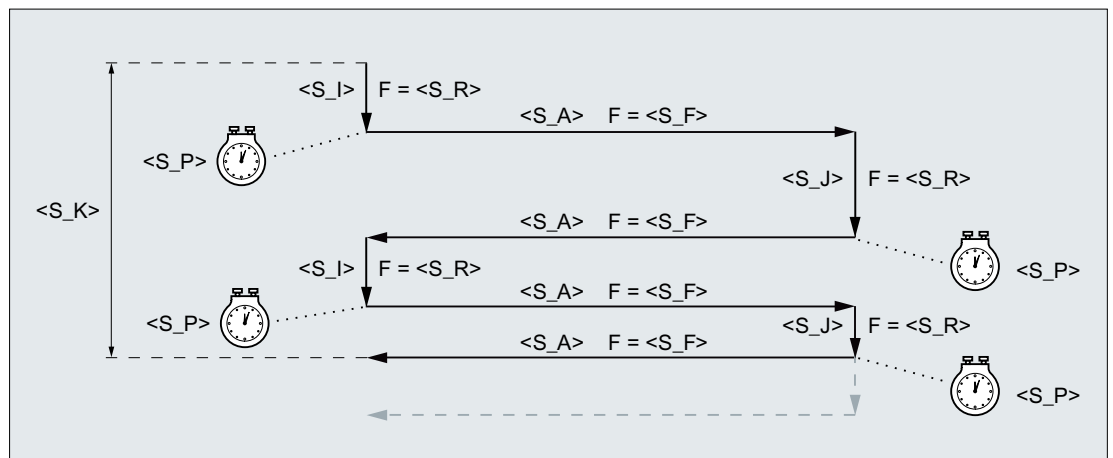


Bild 19-9 Gesamtzustelltiefe erreicht bei Zustellung am zweiten Umkehrpunkt

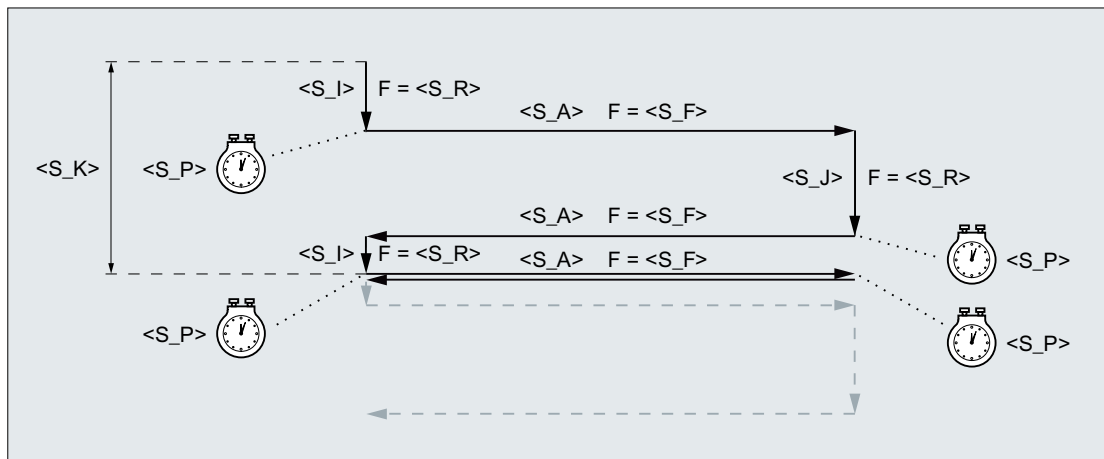


Bild 19-10 Gesamtzustelltiefe erreicht bei Zustellung am ersten Umkehrpunkt

Beispiel

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

Programmcode

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

19.1.52 Randbedingungen

19.1.52.1 Technologieskalierung innerhalb von Zyklenmasken

Ist die Technologieskalierung aktiv, kann bei verschiedenen Zyklenmasken die vereinfachte Eingabe angewählt werden, bei der nur die wesentlichsten Zyklenparameter angezeigt werden.

Bei folgenden Zyklenmasken kann z.B. die vereinfachte Eingabe angewählt werden:

Technologie	Zyklenmaske
Bohren	Tieflochbohren
	Gewindebohren
Fräsen	Rechtecktasche
	Konturfräsen: Tasche
Drehen	Gewindedrehen: Längs
	Konturdrehen: Abspannen
	Konturdrehen: Stechen
	Konturdrehen: Stechdrehen

Bei den betreffenden Zyklenmasken steht an der Bedienoberfläche die Auswahlmöglichkeit "Eingabe: **einfach**" und "Eingabe: **komplett**" zur Verfügung.

Nicht angezeigte Zyklenparameter

Die bei der vereinfachten Eingabe nicht angezeigten Zyklenparameter werden entweder mit festen, technologisch sinnvollen, aber nicht änderbaren Werten vorbelegt. Oder den Zyklenparametern werden über kanalspezifische Zyklen-Settingdaten parametrierbare Werte zugewiesen. Siehe unten Absatz "Inbetriebnahme" > "Kanalspezifische Zyklen-Settingdaten"

Umschaltung "Eingabe: komplett" > "Eingabe: einfach"

Wird eine Zyklenmasken unter der Einstellung "Eingabe komplett" ausgefüllt und anschließend auf "Eingabe einfach" umgeschaltet, werden bei der Generierung des Zyklenaufrufs für die dann nicht mehr angezeigten Parameter, die Default- oder Settingdatenwerte verwendet.

Inbetriebnahme

Kanalspezifische Konfigurations-Maschinendaten

Mit dem Maschinendatum wird die Möglichkeit zur Technologieskalierung innerhalb von Zyklenmasken aktiviert:

MD52210 \$MCS_FUNCTION_MASK_DISP, Bit 9 = 1 (Auswahl "Eingabe einfach" anzeigen)

Kanalspezifische Zyklen-Settingdaten

Ist die vereinfachte Eingabe innerhalb von Zyklenmasken aktiv, können die Werte für bestimmte Zyklenparameter über folgende Settingdaten vorgegeben werden:

Nummer	Bezeichner	Bedeutung
SD55300	\$SCS_EASY_SAFETY_CLEARANCE	Sicherheitsabstand
SD55301	\$SCS_EASY_DWELL_TIME	Verweilzeit
SD55305	\$SCS_EASY_DRILL_DEEP_FD1	Tieflochbohren: Prozentsatz: 1. Vorschub
SD55306	\$SCS_EASY_DRILL_DEEP_DF	Tieflochbohren: Prozentsatz: Zustellung
SD55307	\$SCS_EASY_DRILL_DEEP_V1	Tieflochbohren: minimale Tiefenzustellung
SD55308	\$SCS_EASY_DRILL_DEEP_V2	Tieflochbohren: Rückzugsbetrag
SD55309	\$SCS_EASY_THREAD_RETURN_DIST	Gewindedrehen: Rücklaufabstand

19.2 Messzyklen

Messzyklen sind spezielle von Siemens bereitgestellte Unterprogramme zur Lösung bestimmter Messaufgaben. Wie bei Zyklen allgemein üblich, können auch die Messzyklen über Parameter an das konkrete Problem angepasst werden.

Messzyklen stehen für Messungen in folgenden Bereichen und Technologien zur Verfügung:

- Werkzeugmessungen Drehen / Fräsen
- Werkstückmessungen Drehen / Fräsen

Literatur

Die ausführliche Beschreibung der Messzyklen findet sich in:

Programmierhandbuch Messzyklen

Tabellen

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
:	O	NC-Hauptsatznummer, Sprungmarkenabschluss, Kettungsoperator		+		PGAsI
*	O	Operator für Multiplikation		+		PGAsI
+	O	Operator für Addition		+		PGAsI
-	O	Operator für Subtraktion		+		PGAsI
<	O	Vergleichsoperator, kleiner		+		PGAsI
<<	O	Verkettungsoperator für Strings		+		PGAsI
<=	O	Vergleichsoperator, kleiner gleich		+		PGAsI
=	O	Zuweisungsoperator		+		PGAsI
>=	O	Vergleichsoperator, größer gleich		+		PGAsI
/	O	Operator für Division		+		PGAsI
/0		Satz wird ausgeblendet (1. AusblendeEbene) ^o		+		PGsI
...		...				
...		...				
/7		Satz wird ausgeblendet (8. AusblendeEbene)				
A	A	Achsname	m/s	+		PGAsI
A2	A	Werkzeugorientierung: RPY- oder Eulerwinkel	s	+		PGAsI
A3	A	Werkzeugorientierung: Vektorkomponente Richtung-/Flächennormal	s	+		PGAsI
A4	A	Werkzeugorientierung: Flächennormalvektor für den Satzanfang	s	+		PGAsI
A5	A	Werkzeugorientierung: Flächennormalenvektor für das Satzende	s	+		PGAsI
ABS	F	Absolutwert (Betrag)		+	+	PGAsI
AC	K	absolute Maßangabe von Koordinaten/Positionen	s	+		PGsI
ACC	K	Beeinflussung der aktuellen axialen Beschleunigung	m	+	+	PGsI
ACCLIMA	K	Beeinflussung der aktuellen maximalen axialen Beschleunigung	m	+	+	PGAsI
ACN	K	absolute Maßangabe für Rundachsen, Position in negativer Richtung anfahren	s	+		PGsI
ACOS	F	Arcus-Cosinus (Trigon. Funktion)		+	+	PGAsI
ACP	K	absolute Maßangabe für Rundachsen, Position in positiver Richtung anfahren	s	+		PGsI
ACTBLOCNO	P	Ausgabe der aktuellen Satznummer eines Alarmsatzes, auch wenn "aktuelle Satzanzeige unterdrückt" (DISPLOF) aktiv ist!		+		PGAsI

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
ADDFRAME	F	Einrechnung und evtl. Aktivierung eines gemessenen Frames		+	-	PGAsl, FB1sl (K2)
ADIS	A	Überschleifabstand für Bahnfunktionen G1, G2, G3, ...	m	+		PGsl
ADISPOS	A	Überschleifabstand für Eilgang G0	m	+		PGsl
ADISPOSA	P	Größe des Toleranzfenster für IPOBRKA	m	+	+	PGAsl
ALF	A	Schnellabhebewinkel	m	+		PGAsl
AMIRROR	G	Programmierbare Spiegelung	s	+		PGsl
AND	K	Logisches UND		+		PGAsl
ANG	A	Konturzug-Winkel	s	+		PGsl
AP	A	Polarwinkel	m/s	+		PGsl
APR	K	Zugriffsschutz lesen / anzeigen		+		PGAsl
APRB	K	Zugriffsrecht lesen, BTSS		+		PGAsl
APRP	K	Zugriffsrecht lesen, Teileprogramm		+		PGAsl
APW	K	Zugriffsschutz schreiben		+		PGAsl
APWB	K	Zugriffsrecht schreiben, BTSS		+		PGAsl
APWP	K	Zugriffsrecht schreiben, Teileprogramm		+		PGAsl
APX	K	Definition des Zugriffsschutzes für die Ausführung des angegebenen Sprachelements		+		PGAsl
AR	A	Öffnungswinkel	m/s	+		PGsl
AROT	G	Programmierbare Drehung	s	+		PGsl
AROTS	G	Programmierbare Framedrehungen mit Raumwinkeln	s	+		PGsl
AS	K	Makro-Definition		+		PGAsl
ASCALE	G	Programmierbare Skalierung	s	+		PGsl
ASIN	F	Rechenfunktion, Arcussinus		+	+	PGAsl
ASPLINE	G	Akima-Spline	m	+		PGAsl
ATAN2	F	Arcus-Tangens2		+	+	PGAsl
ATOL	K	achs-spezifische Toleranz für Kompressor-Funktionen, Orientierungsglättung und Überschleifarten		+		PGAsl
ATRANS	G	additive programmierbare Verschiebung	s	+		PGsl
AUXFUDEL	P	Hilfsfunktion kanalspezifisch aus der globalen Liste löschen		+	-	FB1sl (H2)
AUXFUDELG	P	Alle Hilfsfunktionen einer Hilfsfunktionsgruppe kanalspezifisch aus der globalen Liste löschen		+	-	FB1sl (H2)
AUXFUMSEQ	P	Ausgabe-Reihenfolge für M-Hilfsfunktionen ermitteln		+	-	FB1sl (H2)
AUXFUSYNC	P	Aus der globalen Liste der Hilfsfunktionen einen vollständigen Teileprogrammsatz für das kanalspezifische SERUPRO-Ende-ASUP als String generieren		+	-	FB1sl (H2)
AX	K	Variabler Achsbezeichner	m/s	+		PGAsl
AXCTSWE	P	Achscontainer drehen		+	-	PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
AXCTSWEC	P	Freigabe zur Achscontainer-Drehung zurücknehmen		+	+	PGAsI
AXCTSWED	P	Achscontainer drehen (Befehlsvariante für die Inbetriebnahme!)		+	-	PGAsI
AXIS	K	Achsbezeichner, Achsadresse		+		PGAsI
AXNAME	F	Konvertiert Eingangsstring in Achsbezeichner		+	-	PGAsI
AXSTRING	F	Konvertiert den String Spindelnummer		+	-	PGAsI
AXTOCHAN	P	Achse für einen bestimmten Kanal anfordern. Ist vom NC-Programm und aus Synchronaktion möglich.		+	+	PGAsI
AXTOSPI	F	konvertiert Achsbezeichner in einen Spindelindex um		+	-	PGAsI
B	A	Achsname	m/s	+		PGAsI
B2	A	Werkzeugorientierung: RPY- oder Eulerwinkel	s	+		PGAsI
B3	A	Werkzeugorientierung: Vektorkomponente Richtung-/Flächen-normal	s	+		PGAsI
B4	A	Werkzeugorientierung: Flächennormalvektor für den Satzanfang	s	+		PGAsI
B5	A	Werkzeugorientierung: Flächennormalenvektor für das Satzende	s	+		PGAsI
B_AND	O	Bitweises UND		+		PGAsI
B_OR	O	Bitweises ODER		+		PGAsI
B_NOT	O	Bitweise Negierung		+		PGAsI
B_XOR	O	Bitweises Exklusiv-ODER		+		PGAsI
BAUTO	G	Definieren des ersten Spline-Abschnitts durch die nachfolgenden 3 Punkte	m	+		PGAsI
BLOCK	K	Definiert zusammen mit dem Schlüsselwort TO den abzuarbeitenden Programmteil in einem indirekten Unterprogrammlauf		+		PGAsI
BLSYNC	K	Bearbeitung der Interruptroutine soll erst mit dem nächsten Satzwechsel beginnen		+		PGAsI
BNAT ⁶⁾	G	Natürlicher Übergang zum ersten Spline-Satz	m	+		PGAsI
BOOL	K	Datentyp: Wahrheitswerte TRUE/FALSE bzw. 1/0		+		PGAsI
BOUND	F	Prüft, ob Wert innerhalb des definierten Wertebereichs liegt. Gleichheit gibt Prüfwert zurück.		+	+	PGAsI
BRISK ⁶⁾	G	Sprungförmige Bahnbeschleunigung	m	+		PGAsI
BRISKA	P	Sprungförmige Bahnbeschleunigung für die programmierten Achsen einschalten		+	-	PGAsI
BSPLINE	G	B-Spline	m	+		PGAsI
BTAN	G	Tangentialem Übergang zum ersten Spline-Satz	m	+		PGAsI
C	A	Achsname	m/s	+		PGAsI
C2	A	Werkzeugorientierung: RPY- oder Eulerwinkel	s	+		PGAsI
C3	A	Werkzeugorientierung: Vektorkomponente Richtung-/Flächen-normal	s	+		PGAsI

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
C4	A	Werkzeugorientierung: Flächennormalvektor für den Satzanfang	s	+		PGAsl
C5	A	Werkzeugorientierung: Flächennormalvektor für das Satzende	s	+		PGAsl
CAC	K	Absolutes Anfahren einer Position		+		PGAsl
CACN	K	In Tabelle abgelegter Wert wird absolut in negativer Richtung angefahren		+		PGAsl
CACP	K	In Tabelle abgelegter Wert wird absolut in positiver Richtung angefahren		+		PGAsl
CALCDAT	F	Berechnet Radius und Mittelpunkt eines Kreises aus 3 oder 4 Punkten		+	-	PGAsl
CALCPOSI	F	Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldbegrenzung und Softwarelimits		+	-	PGAsl
CALL	K	Indirekter Unterprogrammaufruf		+		PGAsl
CALLPATH	P	Programmierbarer Suchpfad bei Unterprogrammaufrufen		+	-	PGAsl
CANCEL	P	Modale Synchronaktion abbrechen		+	-	FBSYsl
CASE	K	Bedingte Programmverzweigung		+		PGAsl
CDC	K	Direktes Anfahren einer Position		+		PGAsl
CDOF ⁶⁾	G	Kollisionsüberwachung AUS	m	+		PGsl
CDOF2	G	Kollisionsüberwachung AUS, bei 3D-Umfangsfräsen	m	+		PGsl
CDON	G	Kollisionsüberwachung EIN	m	+		PGsl
CFC ⁶⁾	G	Konstanter Vorschub an der Kontur	m	+		PGsl
CFIN	G	Konstanter Vorschub nur bei Innenkrümmung, nicht bei Außenkrümmung	m	+		PGsl
CFINE	F	Zuweisung der Fein-Verschiebung an eine FRAME-Variable		+	-	PGAsl
CFTCP	G	Konstanter Vorschub im Werkzeugschneiden-Bezugspunkt, Mittelpunktsbahn	m	+		PGsl
CHAN	K	Spezifizierung des Gültigkeitsbereichs von Daten		+		PGAsl
CHANDATA	P	Kanalnummer für Kanaldatenzugriffe einstellen		+	-	PGAsl
CHAR	K	Datentyp: ASCII-Zeichen		+		PGAsl
CHF	A	Fase; Wert = Länge der Fase	s	+		PGsl
CHKDM	F	Prüfung der Eindeutigkeit innerhalb eines Magazins		+	-	FBWsl
CHKDNO	F	Eindeutigkeitsprüfung der D-Nummern		+	-	PGAsl
CHR	A	Fase; Wert = Länge der Fase in Bewegungsrichtung		+		PGsl
CIC	K	Inkrementelles Anfahren einer Position		+		PGAsl
CIP	G	Kreisinterpolation über Zwischenpunkt	m	+		PGsl
CLEARM	P	Rücksetzen einer/mehrerer Marken für Kanalkoordinierung		+	+	PGAsl
CLRINT	P	Interrupt abwählen		+	-	PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
CMIRROR	F	Spiegeln an einer Koordinatenachse		+	-	PGAsI
COARSEA	K	Bewegungsende beim Erreichen von "Genauhalt Grob"	m	+		PGAsI
COLLPAIR	F	Prüfen auf Zugehörigkeit zu einem Kollisionspaar		+		PGAsI
COMPCAD	G	Kompressor EIN: Optimierte Oberflächen-güte bei CAD-Programmen	m	+		PGAsI
COMPCURV	G	Kompressor EIN: krümmungsstetige Polynome	m	+		PGAsI
COMPLETE		Steueranweisung für das Aus- und Einlesen von Daten		+		PGAsI
COMPOF ⁶⁾	G	Kompressor AUS	m	+		PGAsI
COMPON	G	Kompressor EIN		+		PGAsI
CONTDCON	P	Konturdecodierung in Tabellenform EIN		+	-	PGAsI
CONTPRON	P	Referenzaufbereitung einschalten		+	-	PGAsI
CORROF	P	Alle aktiven überlagerten Bewegungen werden abgewählt.		+	-	PGsI
COS	F	Cosinus (Trigon. Funktion)		+	+	PGAsI
COUPDEF	P	Definition ELG-Verband / Synchronspindel-Verband		+	-	PGAsI
COUPDEL	P	ELG-Verband löschen		+	-	PGAsI
COUPOF	P	ELG-Verband / Synchronspindel-paar AUS		+	-	PGAsI
COUPOFS	P	Ausschalten ELG-Verband / Synchronspindel-paar mit Stopp der Folgespindel		+	-	PGAsI
COUPON	P	ELG-Verband / Synchronspindel-paar EIN		+	-	PGAsI
COUPONC	P	Einschalten ELG-Verband / Synchronspindel-paar mit vorhergehender Programmierung übernehmen		+	-	PGAsI
COUPRES	P	ELG-Verband rücksetzen		+	-	PGAsI
CP ⁶⁾	G	Bahnbewegung	m	+		PGAsI
CPBC	K	Generische Kopplung: Satzwechselkriterium		+	+	FB3sl (M3)
CPDEF	K	Generische Kopplung: Anlegen eines Koppelmoduls		+	+	FB3sl (M3)
CPDEL	K	Generische Kopplung: Löschen eines Koppelmoduls		+	+	FB3sl (M3)
CPFMOF	K	Generische Kopplung: Verhalten der Folgeachse beim vollständigen Ausschalten		+	+	FB3sl (M3)
CPFMON	K	Generische Kopplung: Verhalten der Folgeachse beim Einschalten		+	+	FB3sl (M3)
CPFMON	K	Generische Kopplung: Synchronisationsmodus		+	+	FB3sl (M3)
CPFPOS	K	Generische Kopplung: Synchronposition der Folgeachse		+	+	FB3sl (M3)
CPFRS	K	Generische Kopplung: Koordinatenbezugssystem		+	+	FB3sl (M3)
CPLA	K	Generische Kopplung: Definition einer Leitachse		+	-	FB3sl (M3)

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
CPLCTID	K	Generische Kopplung: Nummer der Kurventabelle		+	+	FB3sl (M3)
CPLDEF	K	Generische Kopplung: Definition einer Leitachse und Anlegen eines Koppelmoduls		+	+	FB3sl (M3)
CPLDEL	K	Generische Kopplung: Löschen einer Leitachse eines Koppelmoduls		+	+	FB3sl (M3)
CPLDEN	K	Generische Kopplung: Nenner des Koppelfaktors		+	+	FB3sl (M3)
CPLINSC	K	Generische Kopplung: Skalierfaktor für den Eingangswert einer Leitachse		+	+	FB3sl (M3)
CPLINTR	K	Generische Kopplung: Verschiebewert für den Eingangswert einer Leitachse		+	+	FB3sl (M3)
CPLNUM	K	Generische Kopplung: Zähler des Koppelfaktors		+	+	FB3sl (M3)
CPLOF	K	Generische Kopplung: Ausschalten einer Leitachse eines Koppelmoduls		+	+	FB3sl (M3)
CPLON	K	Generische Kopplung: Einschalten einer Leitachse eines Koppelmoduls		+	+	FB3sl (M3)
CPLOUTSC	K	Generische Kopplung: Skalierfaktor für den Ausgangswert einer Kopplung		+	+	FB3sl (M3)
CPLOUTTR	K	Generische Kopplung: Verschiebewert für den Ausgangswert einer Kopplung		+	+	FB3sl (M3)
CPLPOS	K	Generische Kopplung: Synchronposition der Leitachse		+	+	FB3sl (M3)
CPLSETVAL	K	Generische Kopplung: Kopplungsbezug		+	+	FB3sl (M3)
CPMALARM	K	Generische Kopplung: Unterdrückung spezieller kopplungsbezogener Alarmausgaben		+	+	FB3sl (M3)
CPMBRAKE	K	Generische Kopplung: Verhalten der Folgeachse bei bestimmten Stoppsignalen und -kommandos		+	-	FB3sl (M3)
CPMPRT	K	Generische Kopplung: Kopplungsverhalten beim Teileprogrammstart unter Suchlauf via Programmtest		+	+	FB3sl (M3)
CPMRESET	K	Generische Kopplung: Kopplungsverhalten bei RESET		+	+	FB3sl (M3)
CPMSTART	K	Generische Kopplung: Kopplungsverhalten beim Teileprogrammstart		+	+	FB3sl (M3)
CPMVDI	K	Generische Kopplung: Verhalten der Folgeachse auf bestimmte NC/PLC-Nahtstellensignale		+	+	FB3sl (M3)
CPOF	K	Generische Kopplung: Ausschalten eines Koppelmoduls		+	+	FB3sl (M3)
CPON	K	Generische Kopplung: Einschalten eines Koppelmoduls		+	+	FB3sl (M3)
CPRECOF ⁶⁾	G	Programmierbare Konturgenauigkeit AUS	m	+		PGAsl
CPRECON	G	Programmierbare Konturgenauigkeit EIN	m	+		PGAsl
CPRES	K	Generische Kopplung: Aktiviert die projektierten Daten der Synchronspindelkopplung		+	-	FB3sl (M3)
CPROT	P	Kanalspezifischer Schutzbereich EIN/AUS		+	-	PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
CPROTDEF	P	Definition eines kanalspezifischen Schutzbereichs		+	-	PGAsI
CPSETTYPE	K	Generische Kopplung: Kopplungstyp		+	+	FB3sl (M3)
CPSYNCOF	K	Generische Kopplung: Schwellwert für den Positionssynchronlauf "Grob"		+	+	FB3sl (M3)
CPSYNCOF2	K	Generische Kopplung: Schwellwert für den Positionssynchronlauf "Grob" 2		+	+	FB3sl (M3)
CPSYNCOV	K	Generische Kopplung: Schwellwert für den Geschwindigkeitssynchronlauf "Grob"		+	+	FB3sl (M3)
CPSYNFIP	K	Generische Kopplung: Schwellwert für den Positionssynchronlauf "Fein"		+	+	FB3sl (M3)
CPSYNFIP2	K	Generische Kopplung: Schwellwert für den Positionssynchronlauf "Fein" 2		+	+	FB3sl (M3)
CPSYNFIV	K	Generische Kopplung: Schwellwert für den Geschwindigkeitssynchronlauf "Fein"		+	+	FB3sl (M3)
CR	A	Kreisradius	s	+		PGsI
CROT	F	Drehung des aktuellen Koordinatensystems		+	-	PGAsI
CROTS	F	Programmierbare Framedrehungen mit Raumwinkeln (Drehung in den angegebenen Achsen)	s	+	-	PGsI
CRPL	F	Frame-Drehung in einer beliebigen Ebene		+	-	FB1sl (K2)
CSCALE	F	Maßstabsfaktor für mehrere Achsen		+	-	PGAsI
CSPLINE	F	Kubischer Spline	m	+		PGAsI
CT	G	Kreis mit tangentialem Übergang	m	+		PGsI
CTAB	F	Ermittle Folgeachsposition anhand der Leitachsposition aus Kurventabelle		+	+	PGAsI
CTABDEF	P	Tabellendefinition EIN		+	-	PGAsI
CTABDEL	P	Kurventabelle löschen		+	-	PGAsI
CTABEND	P	Tabellendefinition AUS		+	-	PGAsI
CTABEXISTS	F	Prüft die Kurventabelle mit der Nummer n		+	+	PGAsI
CTABFNO	F	Anzahl der noch möglichen Kurventabellen im Speicher		+	+	PGAsI
CTABFPOL	F	Anzahl der noch möglichen Polynome im Speicher		+	+	PGAsI
CTABFSEG	F	Anzahl der noch möglichen Kurvensegmente im Speicher		+	+	PGAsI
CTABID	F	Liefert Tabellen-Nummer der n-ten Kurventabelle		+	+	PGAsI
CTABINV	F	Ermittle Leitachsposition anhand der Folgeachsposition aus Kurventabelle		+	+	PGAsI
CTABISLOCK	F	Gibt den Sperrzustand der Kurventabelle mit der Nummer n zurück		+	+	PGAsI
CTABLOCK	P	Löschen und Überschreiben, sperren		+	+	PGAsI
CTABMEMTYP	F	Gibt den Speicher zurück, in dem die Kurventabelle mit der Nummer n angelegt ist.		+	+	PGAsI
CTABMPOL	F	Anzahl der maximal möglichen Polynome im Speicher		+	+	PGAsI

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
CTABMSEG	F	Anzahl der maximal möglichen Kurvensegmente im Speicher		+	+	PGAsl
CTABNO	F	Anzahl der definierten Kurventabellen im SRAM oder DRAM		+	+	FB3sl (M3)
CTABNOMEM	F	Anzahl der definierten Kurventabellen im SRAM oder DRAM		+	+	PGAsl
CTABPERIOD	F	Gibt die Tabellenperiodizität der Kurventabelle mit der Nummer n zurück		+	+	PGAsl
CTABPOL	F	Anzahl der bereits verwendeten Polynome im Speicher		+	+	PGAsl
CTABPOLID	F	Anzahl der von der Kurventabelle mit der Nummer n verwendeten Kurvenpolynome		+	+	PGAsl
CTABSEG	F	Anzahl der bereits verwendeten Kurvensegmente im Speicher		+	+	PGAsl
CTABSEGID	F	Anzahl der von der Kurventabelle mit der Nummer n verwendeten Kurvensegmente		+	+	PGAsl
CTABSEV	F	Liefert den Endwert der Folgeachse eines Segments der Kurventabelle		+	+	PGAsl
CTABSSV	F	Liefert den Startwert der Folgeachse eines Segments der Kurventabelle		+	+	PGAsl
CTABTEP	F	Liefert den Wert der Leitachse am Kurventabellen-Ende		+	+	PGAsl
CTABTEV	F	Liefert den Wert der Folgeachse am Kurventabellen-Ende		+	+	PGAsl
CTABTMAX	F	Liefert Maximalwert der Folgeachse der Kurventabelle		+	+	PGAsl
CTABTMIN	F	Liefert Minimalwert der Folgeachse der Kurventabelle		+	+	PGAsl
CTABTSP	F	Liefert den Wert der Leitachse am Kurventabellen-Anfang		+	+	PGAsl
CTABTSV	F	Liefert den Wert der Folgeachse am Kurventabellen-Anfang		+	+	PGAsl
CTABUNLOCK	P	Aufheben der Lösch- und Überschreibsperre		+	+	PGAsl
CTOL	K	Konturtoleranz für Kompressor-Funktionen, Orientierungsglättung und Überschleifarten		+		PGAsl
CTRANS	F	Nullpunktverschiebung für mehrere Achsen		+	-	PGAsl
CUT2D ⁶⁾	G	2D-Werkzeugkorrektur	m	+		PGsl
CUT2DF	G	2D-Werkzeugkorrektur Die Werkzeugkorrektur wirkt relativ zum aktuellen Frame (schräge Ebene).	m	+		PGsl
CUT3DC	G	3D-Werkzeugkorrektur Umfangsfräsen	m	+		PGAsl
CUT3DCC	G	3D-Werkzeugkorrektur Umfangsfräsen mit Begrenzungsflächen	m	+		PGAsl
CUT3DCCD	G	3D-Werkzeugkorrektur Umfangsfräsen mit Begrenzungsflächen mit Differenzwerkzeug	m	+		PGAsl
CUT3DF	G	3D-Werkzeugkorrektur Stirnfräsen	m	+		PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
CUT3DFF	G	3D-Werkzeugkorrektur Stirnfräsen mit konstanter Werkzeugorientierung abhängig vom aktiven Frame	m	+		PGAsI
CUT3DFS	G	3D-Werkzeugkorrektur Stirnfräsen mit konstanter Werkzeugorientierung unabhängig vom aktiven Frame	m	+		PGAsI
CUTCONOF ⁶⁾	G	Konstante Radiuskorrektur AUS	m	+		PGsI
CUTCONON	G	Konstante Radiuskorrektur EIN	m	+		PGsI
CUTMOD	K	Funktion "Modifikation der Korrekturdaten bei drehbaren Werkzeugen" einschalten		+		PGAsI
CYCLE60	C	Gravurzyklus		+		PGAsI
CYCLE61	C	Planfräsen		+		PGAsI
CYCLE62	C	Konturaufruf		+		PGAsI
CYCLE63	C	Konturtasche fräsen		+		PGAsI
CYCLE64	C	Konturtasche vorbohren		+		PGAsI
CYCLE70	C	Gewindefräsen		+		PGAsI
CYCLE72	C	Bahnfräsen		+		PGAsI
CYCLE76	C	Rechteckzapfen fräsen		+		PGAsI
CYCLE77	C	Kreiszapfen fräsen		+		PGAsI
CYCLE78	C	Bohrgewinde fräsen		+		PGAsI
CYCLE79	C	Mehrkant		+		PGAsI
CYCLE81	C	Bohren, Zentrieren		+		PGAsI
CYCLE82	C	Bohren, Plansenken		+		PGAsI
CYCLE83	C	Tieflochbohren		+		PGAsI
CYCLE84	C	Gewindebohren ohne Ausgleichsfutter		+		PGAsI
CYCLE85	C	Reiben		+		PGAsI
CYCLE86	C	Ausdrehen		+		PGAsI
CYCLE92	C	Abstich		+		PGAsI
CYCLE95	C	Konturabspannen		+		PGAsI
CYCLE98	C	Gewindekette		+		PGAsI
CYCLE99	C	Gewindedrehen		+		PGAsI
CYCLE495	C	Profilieren		+		PGAsI
CYCLE751	C	Optimierungssitzung öffnen / ausführen / schließen		+		FB3sI (T4)
CYCLE752	C	Achse zu einer Optimierungssitzung hinzufügen		+		FB3sI (T4)
CYCLE753	C	Optimierungsmodus auswählen		+		FB3sI (T4)
CYCLE754	C	Datensatz hinzufügen / entfernen		+		FB3sI (T4)
CYCLE755	C	Datensatz sichern / wiederherstellen		+		FB3sI (T4)
CYCLE756	C	Optimierungsergebnisse aktivieren		+		FB3sI (T4)
CYCLE757	C	Optimierungsdaten speichern		+		FB3sI (T4)
CYCLE758	C	Parameterwert ändern		+		FB3sI (T4)
CYCLE759	C	Parameterwert lesen		+		FB3sI (T4)
CYCLE800	C	Schwenken		+		PGAsI

Tabellen

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
CYCLE801	C	Gitter oder Rahmen		+		PGAsI
CYCLE802	C	Beliebige Positionen		+		PGAsI
CYCLE830	C	Tieflochbohren 2		+		PGAsI
CYCLE832	C	High Speed Settings		+		PGAsI
CYCLE840	C	Gewindebohren mit Ausgleichsfutter		+		PGAsI
CYCLE899	C	Offene Nut fräsen		+		PGAsI
CYCLE930	C	Einstich		+		PGAsI
CYCLE940	C	Freistich Formen		+		PGAsI
CYCLE951	C	Abspannen		+		PGAsI
CYCLE952	C	Konturstechen		+		PGAsI
CYCLE4071	C	Längschleifen mit Zustellung am Umkehrpunkt		+		PGAsI
CYCLE4072	C	Längschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal		+		PGAsI
CYCLE4073	C	Längschleifen mit kontinuierlicher Zustellung		+		PGAsI
CYCLE4074	C	Längschleifen mit kontinuierlicher Zustellung und Abbruchsignal		+		PGAsI
CYCLE4075	C	Flachschleifen mit Zustellung am Umkehrpunkt		+		PGAsI
CYCLE4077	C	Flachschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal		+		PGAsI
CYCLE4078	C	Flachschleifen mit kontinuierlicher Zustellung		+		PGAsI
CYCLE4079	C	Flachschleifen mit intermittierender Zustellung		+		PGAsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
D	A	Werkzeugkorrekturnummer		+		PGsI
D0	A	Bei D0 sind die Korrekturen für das Werkzeug unwirksam		+		PGsI
DAC	K	Absolut satzweise achsspezifische Durchmesserprogrammierung	s	+		PGsI
DC	K	Absolute Maßangabe für Rundachsen, Position direkt anfahren	s	+		PGsI
DCI	K	Datenklasse I (= Individual) zuordnen (nur SINUMERIK 828D!)		+		PGAsI
DCM	K	Datenklasse M (= Manufacturer) zuordnen (nur SINUMERIK 828D!)		+		PGAsI
DCU	K	Datenklasse U (= Anwender) zuordnen (nur SINUMERIK 828D!)		+		PGAsI
DEF	K	Variablendefinition		+		PGAsI
DEFAULT	K	Zweig in der CASE-Verzweigung		+		PGAsI
DEFINE	K	Schlüsselwort für Makrodefinitionen		+		PGAsI
DELAYFSTOF	P	Ende eines Stopp-Delay-Bereichs definieren	m	+	-	PGAsI
DELAYFSTON	P	Beginn eines Stopp-Delay-Bereichs definieren	m	+	-	PGAsI
DELDL	F	Additive Korrekturen löschen		+	-	PGAsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
DELDTG	P	Restweglöschen		-	+	FBSYsl
DELETE	P	Die angegebene Datei löschen. Der Dateiname kann mit Pfad und Datei-Kennung angegeben werden.		+	-	PGAsl
DELMOWNER	F	Eigentümergebiet des Werkzeugs löschen		+	-	FBWsl
DELMRES	F	Magazinsplatz-Reservierung löschen		+	-	FBWsl
DELMT	P	Multitool löschen		+	-	FBWsl
DELOBJ	F	Löschen von Elementen von kinematischen Ketten, Schutzbereichen, Schutzbereichselementen, Kollisionspaaren und Transformationsdaten		+		PGAsl
DELT	P	Werkzeug löschen		+	-	FBWsl
DELTC	P	Werkzeugträgerdatensatz löschen		+	-	FBWsl
DELTOOLNV	F	Datensätze zur Beschreibung von Werkzeugumgebungen löschen		+	-	FB1sl (W1)
DIACYCOFA	K	Achsspezifische modale Durchmesserprogrammierung: AUS in Zyklen	m	+		FB1sl (P1)
DIAM90	G	Durchmesserprogrammierung für G90, Radiusprogrammierung für G91	m	+		PGAsl
DIAM90A	K	Achsspezifische modale Durchmesserprogrammierung für G90 und AC, Radiusprogrammierung für G91 und IC	m	+		PGsl
DIAMCHAN	K	Übernahme aller Achsen aus MD Achsfunktionen in den Kanalzustand der Durchmesserprogrammierung		+		PGsl
DIAMCHANA	K	Übernahme Kanalzustand der Durchmesserprogrammierung		+		PGsl
DIAMCYCOF	G	Kanalspezifische Durchmesserprogrammierung: AUS in Zyklen	m	+		FB1sl (P1)
DIAMOF ⁶⁾	G	Durchmesserprogrammierung: AUS Grundstellung siehe Maschinenhersteller	m	+		PGsl
DIAMOFA	K	Achsspezifische modale Durchmesserprogrammierung: AUS Grundstellung siehe Maschinenhersteller	m	+		PGsl
DIAMON	G	Durchmesserprogrammierung: EIN	m	+		PGsl
DIAMONA	K	Achsspezifische modale Durchmesserprogrammierung: EIN Freischaltung siehe Maschinenhersteller	m	+		PGsl
DIC	K	Relativ satzweise achsspezifische Durchmesserprogrammierung	s	+		PGsl
DILF	A	Rückzugsweg (Länge)	m	+		PGsl
DISABLE	P	Interrupt AUS		+	-	PGAsl
DISC	A	Überhöhung Übergangskreis Werkzeug-Radiuskorrektur	m	+		PGsl
DISCL	A	Abstand des Endpunkts der schnellen Zustellbewegung, von der Bearbeitungsebene		+		PGsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
DISPLOF	PA	Aktuelle Satzanzeige unterdrücken		+		PGAsI
DISPLON	PA	Unterdrückung der aktuellen Satzanzeige aufheben		+		PGAsI
DISPR	A	Repos-Bahndifferenz	s	+		PGAsI
DISR	A	Repos-Abstand	s	+		PGAsI
DISRP	A	Abstand der Rückzugsebene von der Bearbeitungsebene beim weichen An- und Abfahren		+		PGsI
DITE	A	Gewindeauslaufweg	m	+		PGsI
DITS	A	Gewindeeinlaufweg	m	+		PGsI
DIV	K	Integer-Division		+		PGAsI
DL	A	Ortsabhängige additive Werkzeugkorrektur an-wählen (DL, Summen- Einrichtekorrektur)	m	+		PGAsI
DO	A	Schlüsselwort für Synchronaktion, löst bei erfüllter Bedingung Aktion aus		-	+	FBSYsI
DRFOF	P	Ausschalten der Handradverschiebungen (DRF)	m	+	-	PGsI
DRIVE	G	Geschwindigkeitsabhängige Bahnbeschleunigung	m	+		PGAsI
DRIVEA	P	Geknickte Beschleunigungskennlinie für die programmierten Achsen einschalten		+	-	PGAsI
DYNFINISH	G	Dynamik für Feinschlichten	m	+		PGAsI
DYNNORM ⁶⁾	G	Normale Dynamik	m	+		PGAsI
DYNPOS	G	Dynamik für Positionierbetrieb, Gewindebohren	m	+		PGAsI
DYNROUGH	G	Dynamik für Schruppen	m	+		PGAsI
DYNSEMIFIN	G	Dynamik für Schlichten	m	+		PGAsI
DZERO	P	Kennzeichnet alle D-Nummern der TO-Einheit als ungültig		+	-	PGAsI
EAUTO	G	Festlegung des letzten Spline-Abschnitts durch die letzten 3 Punkte	m	+		PGAsI
EGDEF	P	Definition eines elektronischen Getriebes		+	-	PGAsI
EGDEL	P	Kopplungsdefinition für die Folgeachse löschen		+	-	PGAsI
EGOFC	P	Elektronisches Getriebe kontinuierlich ausschalten		+	-	PGAsI
EGOFS	P	Elektronisches Getriebe selektiv ausschalten		+	-	PGAsI
EGON	P	Elektronisches Getriebe einschalten		+	-	PGAsI
EGONSYN	P	Elektronisches Getriebe einschalten		+	-	PGAsI
EGONSYNE	P	Elektronisches Getriebe einschalten, mit Vorgabe von Anfahrmodus		+	-	PGAsI
ELSE	K	Programmverzweigung, wenn IF-Bedingung nicht erfüllt		+		PGAsI
ENABLE	P	Interrupt EIN		+	-	PGAsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
ENAT ⁶⁾	G	Natürlicher Kurvenübergang zum nächsten Verfahrtsatz	m	+		PGAsl
ENDFOR	K	Endezeile der FOR-Zählschleife		+		PGAsl
ENDIF	K	Endezeile der IF-Verzweigung		+		PGAsl
ENDLABEL	K	Endmarke für Teilprogrammwiederholungen über REPEAT		+		PGAsl, FB1sl (K1)
ENDLOOP	K	Endezeile der Endlos-Programmschleife LOOP		+		PGAsl
ENDPROC	K	Endezeile eines Programms mit der Anfangszeile PROC		+		
ENDWHILE	K	Endezeile der WHILE-Schleife		+		PGAsl
ESRR	P	Antriebsautarkes ESR-Rückziehen im Antrieb parametrieren		+		PGAsl
ESRS	P	Antriebsautarkes ESR-Stilsetzen im Antrieb parametrieren		+		PGAsl
ETAN	G	Tangentialer Kurvenübergang zum nächsten Verfahrtsatz bei Spline-Beginn	m	+		PGAsl
EVERY	K	Synchronaktion ausführen bei Übergang der Bedingung von FALSE zu TRUE		-	+	FBSYsl
EX	K	Schlüsselwert für die Wertzuweisung in exponentieller Schreibweise		+		PGAsl
EXECSTRING	P	Übergabe einer String-Variablen mit der auszuführenden Teileprogrammzeile		+	-	PGAsl
EXECTAB	P	Ein Element aus einer Bewegungstabelle abarbeiten		+	-	PGAsl
EXECUTE	P	Programmausführung EIN		+	-	PGAsl
EXP	F	Exponentialfunktion ex		+	+	PGAsl
EXTCALL	A	Externes Unterprogramm abarbeiten		+	+	PGAsl
EXTCLOSE	P	Zum Schreiben geöffnetes externes Gerät/ Datei schließen		+	-	PGAsl
EXTERN	K	Bekanntmachung eines Unterprogramms mit Parameterübergabe		+		PGAsl
EXTOPEN	P	Externes Gerät/Datei für den Kanal zum Schreiben öffnen		+	-	PGAsl
F	A	Vorschubwert (in Verbindung mit G4 wird mit F auch die Verweilzeit programmiert)		+	+	PGsl
FA	K	Axialer Vorschub	m	+	+	PGsl
FAD	A	Zustell-Vorschub für Weiches An- und Abfahren		+		PGsl
FALSE	K	Logische Konstante: falsch		+	+	PGAsl
FB	A	Satzweiser Vorschub		+		PGsl
FCTDEF	P	Polynomfunktion definieren		+	-	PGAsl
FCUB	G	Vorschub nach kubischem Spline veränderlich	m	+		PGAsl
FD	A	Bahnvorschub für Handradüberlagerung	s	+		PGsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
FDA	K	Axialer Vorschub für Handradüberlagerung	s	+		PGsI
FENDNORM ⁶⁾	G	Eckenverzögerung AUS	m	+		PGAsI
FFWOF ⁶⁾	G	Vorsteuerung AUS	m	+		PGAsI
FFWON	G	Vorsteuerung Ein	m	+		PGAsI
FGREF	K	Bezugsradius bei Rundachsen oder Bahnbezugsfaktoren bei Orientierungsachsen (Vektorinterpolation)	m	+		PGsI
FGROUP	P	Festlegung der Achse(n) mit Bahnvorschub		+	-	PGsI
FI	K	Parameter für Zugriff auf Framedaten: Feinverschiebung		+		PGAsI
FIFOCTRL	G	Steuerung des Vorlaufpuffers	m	+		PGAsI
FILEDATE	P	Liefert Datum des zuletzt schreibenden Zugriffs auf die Datei		+	-	PGAsI
FILEINFO	P	Liefert Summe von FILEDATE, FILESIZE, FILESTAT und FILETIME zusammen		+	-	PGAsI
FILESIZE	P	Liefert aktuelle Größe der Datei		+	-	PGAsI
FILESTAT	P	Liefert Filestatus der Rechte Lesen, Schreiben, Execute, Anzeigen, Löschen (rwxsd)		+	-	PGAsI
FILETIME	P	Liefert Uhrzeit des zuletzt schreibenden Zugriffs auf die Datei		+	-	PGAsI
FINEA	K	Bewegungsende beim Erreichen von "Genauhalt Fein"	m	+		PGAsI
FL	K	Grenzgeschwindigkeit für Synchronachsen	m	+		PGsI
FLIN	G	Vorschub linear veränderlich	m	+		PGAsI
FMA	K	Mehrere Vorschübe axial	m	+		PGsI
FNORM ⁶⁾	G	Vorschub normal nach DIN66025	m	+		PGAsI
FOC	K	Satzweise wirksame Moment/Kraft-Begrenzung	s	-	+	FBSYsI
FOCOF	K	Modale Moment/Kraft-Begrenzung ausschalten	m	-	+	FBSYsI
FOCON	K	Modale Moment/Kraft-Begrenzung einschalten	m	-	+	FBSYsI
FOR	K	Zählschleife mit fester Anzahl von Durchläufen		+		PGAsI
FP	A	Festpunkt: Nummer des anzufahrenden Festpunkts	s	+		PGsI
FPO	K	Über ein Polynom programmierter Vorschubverlauf		+		PGAsI
FPR	P	Kennzeichnung Rundachse		+	-	PGsI
FPRAOF	P	Umdrehungsvorschub ausschalten		+	-	PGsI
FPRAON	P	Umdrehungsvorschub einschalten		+	-	PGsI
FRAME	K	Datentyp zur Festlegung von Koordinatensystemen		+		PGAsI
FRC	A	Vorschub für Radius und Fase	s	+		PGsI
FRCM	A	Vorschub für Radius und Fase modal	m	+		PGsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
FROM	K	Die Aktion wird ausgeführt, wenn die Bedingung einmal erfüllt ist und solange die Synchronaktion aktiv ist		-	+	FBSYsl
FTOC	P	Werkzeugfeinkorrektur ändern		-	+	FBSYsl
FTOCOF ⁶⁾	G	Online wirksame Werkzeugfeinkorrektur AUS	m	+		PGAsl
FTOCON	G	Online wirksame Werkzeugfeinkorrektur EIN	m	+		PGAsl
FXS	K	Fahren auf Festanschlag ein	m	+	+	PGsl
FXST	K	Momentgrenze für Fahren auf Festanschlag	m	+	+	PGsl
FXSW	K	Überwachungsfenster für Fahren auf Festanschlag		+	+	PGsl
FZ	K	Zahnvorschub	m	+		PGsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
G0	G	Linearinterpolation mit Eilgang (Eilgangsbewegung)	m	+		PGsl
G1 ⁶⁾	G	Linearinterpolation mit Vorschub (Geradeninterpolation)	m	+		PGsl
G2	G	Kreisinterpolation im Uhrzeigersinn	m	+		PGsl
G3	G	Kreisinterpolation gegen Uhrzeigersinn	m	+		PGsl
G4	G	Verweilzeit, zeitlich vorbestimmt	s	+		PGsl
G5	G	Schrägeinstechschleifen	s	+		PGAsl
G7	G	Ausgleichsbewegung beim Schrägeinstechschleifen	s	+		PGAsl
G9	G	Genauhalt - Geschwindigkeitsabnahme	s	+		PGsl
G17 ⁶⁾	G	Wahl der Arbeitsebene X/Y	m	+		PGsl
G18	G	Wahl der Arbeitsebene Z/X	m	+		PGsl
G19	G	Wahl der Arbeitsebene Y/Z	m	+		PGsl
G25	G	Untere Arbeitsfeldbegrenzung	s	+		PGsl
G26	G	Obere Arbeitsfeldbegrenzung	s	+		PGsl
G33	G	Gewindeschneiden mit konstanter Steigung	m	+		PGsl
G34	G	Gewindeschneiden mit linear zunehmender Steigung	m	+		PGsl
G35	G	Gewindeschneiden mit linear abnehmender Steigung	m	+		PGsl
G40 ⁶⁾	G	Werkzeugradiuskorrektur AUS	m	+		PGsl
G41	G	Werkzeugradiuskorrektur links von der Kontur	m	+		PGsl
G42	G	Werkzeugradiuskorrektur rechts von der Kontur	m	+		PGsl
G53	G	Unterdrückung der aktuellen Nullpunktverschiebung (satzweise)	s	+		PGsl
G54	G	1. Einstellbare Nullpunktverschiebung	m	+		PGsl
G55	G	2. Einstellbare Nullpunktverschiebung	m	+		PGsl
G56	G	3. Einstellbare Nullpunktverschiebung	m	+		PGsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
G57	G	4. Einstellbare Nullpunktverschiebung	m	+		PGsl
G58 (840D sl)	G	Axiale programmierbare Nullpunktverschiebung absolut, Grobverschiebung	s	+		PGsl
G58 (828D)	G	5. Einstellbare Nullpunktverschiebung	m	+		PGsl
G59 (840D sl)	G	Axiale programmierbare Nullpunktverschiebung additiv, Feinverschiebung	s	+		PGsl
G59 (828D)	G	6. Einstellbare Nullpunktverschiebung	m	+		PGsl
G60 ⁶⁾	G	Genauhalt - Geschwindigkeitsabnahme	m	+		PGsl
G62	G	Eckenverzögerung an Innenecken bei aktiver Werkzeugradiuskorrektur (G41, G42)	m	+		PGAsl
G63	G	Gewindebohren mit Ausgleichsfutter	s	+		PGsl
G64	G	Bahnsteuerbetrieb	m	+		PGsl
G70	G	Inch-Maßangabe für geometrische Angaben (Längen)	m	+	+	PGsl
G71 ⁶⁾	G	Metrische Maßangabe für geometrische Angaben (Längen)	m	+	+	PGsl
G74	G	Referenzpunktanfahren	s	+		PGsl
G75	G	Festpunktanfahren	s	+		PGsl
G90 ⁶⁾	G	Maßangabe absolut	m/s	+		PGsl
G91	G	Kettenmaßangabe	m/s	+		PGsl
G93	G	Zeitreziproker Vorschub 1/min	m	+		PGsl
G94 ⁶⁾	G	Linearvorschub F in mm/min oder inch/min und Grad/min	m	+		PGsl
G95	G	Umdrehungsvorschub F in mm/U oder inch/U	m	+		PGsl
G96	G	konstante Schnittgeschwindigkeit (wie bei G95) EIN	m	+		PGsl
G97	G	konstante Schnittgeschwindigkeit (wie bei G95) AUS	m	+		PGsl
G110	G	Polprogrammierung relativ zur letzten programmierten Sollposition	s	+		PGsl
G111	G	Polprogrammierung relativ zum Nullpunkt des aktuellen Werkstück-Koordinatensystems	s	+		PGsl
G112	G	Polprogrammierung relativ zum letzten gültigen Pol	s	+		PGsl
G140 ⁶⁾	G	Anfahrriechtung WAB festgelegt durch G41/G42	m	+		PGsl
G141	G	Anfahrriechtung WAB links der Kontur	m	+		PGsl
G142	G	Anfahrriechtung WAB rechts der Kontur	m	+		PGsl
G143	G	Anfahrriechtung WAB tangentialabhängig	m	+		PGsl
G147	G	Weiches Anfahren mit Gerade	s	+		PGsl
G148	G	Weiches Abfahren mit Gerade	s	+		PGsl
G153	G	Unterdrückung aktueller Frames inklusive Basisframe	s	+		PGsl
G247	G	Weiches Anfahren mit Viertelkreis	s	+		PGsl
G248	G	Weiches Abfahren mit Viertelkreis	s	+		PGsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
G290 ⁶⁾	G	Umschalten auf SINUMERIK-Mode EIN	m	+		FBWsl
G291	G	Umschalten auf ISO2/3-Mode EIN	m	+		FBWsl
G331	G	Gewindebohren ohne Ausgleichsfutter, positive Steigung, Rechtslauf	m	+		PGsl
G332	G	Gewindebohren ohne Ausgleichsfutter, negative Steigung, Linkslauf	m	+		PGsl
G335	G	Drehen eines balligen Gewindes im Uhrzeigersinn	m	+		PGsl
G336	G	Drehen eines balligen Gewindes gegen den Uhrzeigersinn	m	+		PGsl
G340 ⁶⁾	G	Anfahrstanz räumlich (Tiefe und in der Ebene zugleich (Helix))	m	+		PGsl
G341	G	Zuerst in der senkrechten Achse zustellen (z), dann Anfahren in der Ebene	m	+		PGsl
G347	G	Weiches Anfahren mit Halbkreis	s	+		PGsl
G348	G	Weiches Abfahren mit Halbkreis	s	+		PGsl
G450 ⁶⁾	G	Übergangskreis	m	+		PGsl
G451	G	Schnittpunkt der Äquidistanten	m	+		PGsl
G460 ⁶⁾	G	Einschalten der Kollisionsüberwachung für An- und Abfahrstanz	m	+		PGsl
G461	G	Einfügen eines Kreises im WRK-Satz	m	+		PGsl
G462	G	Einfügen einer Geraden im WRK-Satz	m	+		PGsl
G500 ⁶⁾	G	Ausschalten aller einstellbaren Frames, Basisframes sind aktiv	m	+		PGsl
G505 ... G599	G	5 ... 99. Einstellbare Nullpunktverschiebung	m	+		PGsl
G601 ⁶⁾	G	Satzwechsel bei Genauhalt fein	m	+		PGsl
G602	G	Satzwechsel bei Genauhalt grob	m	+		PGsl
G603	G	Satzwechsel bei IPO-Satzende	m	+		PGsl
G621	G	Eckenverzögerung an allen Ecken	m	+		PGAsl
G641	G	Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium (= programmierbarer Überschleifabstand)	m	+		PGsl
G642	G	Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen	m	+		PGsl
G643	G	Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen (satzintern)	m	+		PGsl
G644	G	Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Dynamik	m	+		PGsl
G645	G	Bahnsteuerbetrieb mit Überschleifen von Ecken und tangentialer Satzübergänge unter Einhaltung definierter Toleranzen	m	+		PGsl
G700	G	Inch-Maßangabe für geometrische und technologische Angaben (Längen, Vorschub)	m	+	+	PGsl
G710 ⁶⁾	G	Metrische Maßangabe für geometrische und technologische Angaben (Längen, Vorschub)	m	+	+	PGsl
G810 ⁶⁾ , ..., G819	G	Für den OEM-Anwender reservierte G-Gruppe		+		PGAsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
G820 ⁶⁾ , ..., G829	G	Für den OEM-Anwender reservierte G-Gruppe		+		PGAsl
G931	G	Vorschubvorgabe durch Verfahrszeit	m	+		
G942	G	Linear-Vorschub und konstante Schnittgeschwindigkeit oder Spindeldrehzahl einfrieren	m	+		
G952	G	Umdrehungsvorschub und konstante Schnittgeschwindigkeit oder Spindeldrehzahl einfrieren	m	+		
G961	G	konstante Schnittgeschwindigkeit und Linear-Vorschub	m	+		PGsl
G962	G	Linear-Vorschub oder Umdrehungsvorschub und konstante Schnittgeschwindigkeit	m	+		PGsl
G971	G	Spindeldrehzahl einfrieren und Linear-Vorschub	m	+		PGsl
G972	G	Linear-Vorschub oder Umdrehungsvorschub und konstante Spindeldrehzahl einfrieren	m	+		PGsl
G973	G	Umdrehungsvorschub ohne Spindeldrehzahlbegrenzung	m	+		PGsl
GEOAX	P	Den Geometrieachsen 1 - 3 neue Kanalachsen zuordnen		+	-	PGAsl
GET	P	Freigegebene Achse zwischen Kanälen tauschen		+	+	PGAsl
GETACTT	F	Bestimmt das aktive Werkzeug aus einer Gruppe von gleichnamigen Werkzeugen		+	-	FBWsl
GETACTTD	F	Bestimmt zu einer absoluten D-Nummer die zugehörige T-Nummer		+	-	PGAsl
GETD	P	Achse direkt zwischen Kanälen tauschen		+	-	PGAsl
GETDNO	F	Liefert D-Nummer einer Schneide (CE) eines Werkzeugs (T)		+	-	PGAsl
GETEXET	P	Lesen der eingewechselten T-Nummer		+	-	FBWsl
GETFREELOC	P	Für ein gegebenes Werkzeug einen Leerplatz in den Magazinen suchen		+	-	FBWsl
GETSELT	P	Vorgewählte T-Nummer liefern		+	-	FBWsl
GETT	F	T-Nummer zu Werkzeugnamen bestimmen		+	-	FBWsl
GETTCOR	F	Werkzeu glängen bzw. Werkzeu glängenkomponenten auslesen		+	-	FB1sl (W1)
GETTENV	F	T-, D-, und DL-Nummern lesen		+	-	FB1sl (W1)
GETVARAP	F	Zugriffsrecht auf eine System-/Anwendervariable lesen		+	-	PGAsl
GETVARDFT	F	Standardwert einer System-/Anwendervariablen lesen		+	-	PGAsl
GETVARLIM	F	Grenzwerte einer System-/Anwendervariablen lesen		+	-	PGAsl
GETVARPHU	F	Physikalische Einheit einer System-/Anwendervariablen lesen		+	-	PGAsl
GETVARTYP	F	Datentyp einer System-/Anwendervariablen lesen		+	-	PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
GOTO	K	Sprunganweisung erst vorwärts dann rückwärts (Richtung erst zum Programm-Ende und dann zum Programm-Anfang)		+		PGAsI
GOTOB	K	Sprunganweisung rückwärts (Richtung Programm-Anfang)		+		PGAsI
GOTOC	K	Wie GOTO, aber Alarm 14080 "Sprungziel nicht gefunden" unterdrücken		+		PGAsI
GOTOF	K	Sprunganweisung vorwärts (Richtung Programm-Ende)		+		PGAsI
GOTOS	K	Rücksprung auf Programmanfang		+		PGAsI
GP	K	Schlüsselwort zur indirekten Programmierung von Positionsattributen		+		PGAsI
GWPSOF	P	Konstante Scheibenumfangsgeschwindigkeit (SUG) abwählen	s	+	-	PGsI
GWPSON	P	Konstante Scheibenumfangsgeschwindigkeit (SUG) anwählen	s	+	-	PGsI
H...	A	Hilfsfunktionsausgabe an die PLC		+	+	PGsI/FB1sI (H2)
HOLES1	C	Lochreihe		+		PGAsI
HOLES2	C	Lochkreis		+		PGAsI
I	A	Interpolationsparameter	s	+		PGsI
I1	A	Zwischenpunktcoordinate	s	+		PGsI
IC	K	Kettenmaßeingabe	s	+		PGsI
ICYCOF	P	Alle Sätze eines Technologiezyklus nach ICYCOF in einem IPO-Takt abarbeiten		+	+	FBSYsI
ICYCON	P	Jeden Satz eines Technologiezyklus nach ICYCON in einem separaten IPO-Takt abarbeiten		+	+	FBSYsI
ID	K	Kennzeichnung für modale Synchronaktionen	m	-	+	FBSYsI
IDS	K	Kennzeichnung für modale statische Synchronaktionen		-	+	FBSYsI
IF	K	Einleitung eines bedingten Sprungs im Teileprogramm / Technologiezyklus		+	+	PGAsI
INDEX	F	Index eines Zeichens im Eingangsstring bestimmen		+	-	PGAsI
INICF	K	Initialisierung der Variablen bei NewConfig		+		PGAsI
INIPO	K	Initialisierung der Variablen bei PowerOn		+		PGAsI
INIRE	K	Initialisierung der Variablen bei Reset		+		PGAsI
INIT	P	Anwahl eines bestimmten NC-Programms zur Abarbeitung in einem bestimmten Kanal		+	-	PGAsI
INITIAL		Erzeugen eines INI-Files über alle Bereiche		+		PGAsI
INT	K	Datentyp: Ganzzahliger Wert mit Vorzeichen		+		PGAsI
INTERSEC	F	Schnittpunkt zwischen zwei Konturelementen berechnen		+	-	PGAsI
INVCCW	G	Evolvente fahren, gegen den Uhrzeigersinn	m	+		PGsI
INVCW	G	Evolvente fahren, im Uhrzeigersinn	m	+		PGsI

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
INVFRAME	F	Aus einem Frame den inversen Frame berechnen		+	-	FB1sl (K2)
IP	K	Variabler Interpolationsparameter		+		PGAsl
IPOBRKA	P	Bewegungskriterium ab Einsatzpunkt der Bremsrampe	m	+	+	
IPOENDA	K	Bewegungsende beim Erreichen von "IPO-Stopp"	m	+		PGAsl
IPTRLOCK	P	Beginn des suchunfähigen Programmabschnitts auf nächsten Maschinenfunktionssatz einfrieren.	m	+	-	PGAsl
IPTRUNLOCK	P	Ende des suchunfähigen Programmabschnitts auf aktuellen Satz zum Unterbrechungszeitpunkt setzen.	m	+	-	PGAsl
IR	A	Kreismittelpunktcoordinate (X-Richtung) beim Drehen balliger Gewinde		+		PGsl
ISAXIS	F	Prüfen, ob die als Parameter angegebene Geometrieachse 1 ist		+	-	PGAsl
ISD	A	Eintauchtiefe	m	+		PGAsl
ISFILE	F	Prüfen, ob eine Datei im NCK-Anwendungsspeicher vorhanden ist		+	-	PGAsl
ISNUMBER	F	Prüfen, ob Eingangsstring in Zahl umgewandelt werden kann		+	-	PGAsl
ISOCALL	K	Indirekter Aufruf eines in ISO-Sprache programmierten Programms		+		PGAsl
ISVAR	F	Prüfen, ob der Übergabeparameter eine in der NC bekannte Variable enthält		+	-	PGAsl
J	A	Interpolationsparameter	s	+		PGsl
J1	A	Zwischenpunktcoordinate	s	+		PGsl
JERKA	P	Über MD eingestelltes Beschleunigungsverhalten für die programmierten Achsen aktivieren		+	-	
JERKLIM	K	Reduktion oder Überhöhung des maximalen axialen Rucks	m	+		PGAsl
JERKLIMA	K	Reduktion oder Überhöhung des maximalen axialen Rucks	m	+	+	PGAsl
JR	A	Kreismittelpunktcoordinate (Y-Richtung) beim Drehen balliger Gewinde		+		PGsl
K	A	Interpolationsparameter	s	+		PGsl
K1	A	Zwischenpunktcoordinate	s	+		PGsl
KONT	G	Kontur umfahren bei der Werkzeugkorrektur	m	+		PGsl
KONTC	G	Mit krümmungsstetigem Polynom an-/abfahren	m	+		PGsl
KONTT	G	Mit tangentenstetigem Polynom an-/abfahren	m	+		PGsl
KR	A	Kreismittelpunktcoordinate (Z-Richtung) beim Drehen balliger Gewinde		+		PGsl
L	A	Unterprogramm-Nummer	s	+	+	PGAsl
LEAD	A	Voreilwinkel 1. Werkzeugorientierung 2. Orientierungspolynome	m	+		PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
LEADOF	P	Axiale Leitwerkkopplung AUS		+	+	PGAsl
LEADON	P	Axiale Leitwerkkopplung EIN		+	+	PGAsl
LENTOAX	F	Liefert Informationen über die Zuordnung der Werkzeuglängen L1, L2 und L3 des aktiven Werkzeugs zur Abszisse, Ordinate und Applikate		+	-	FB1sl (W1)
LFOF ⁶⁾	G	Schnellrückzug für Gewindeschneiden AUS	m	+		PGsl
LFON	G	Schnellrückzug für Gewindeschneiden EIN	m	+		PGsl
LFPOS	G	Rückzug der mit POLFMASK oder POLFMLIN bekannt gemachten Achse auf die mit POLF programmierte absolute Achsposition	m	+		PGsl
LFTXT ⁶⁾	G	Ebene der Rückzugsbewegung beim Schnellabheben wird bestimmt aus der Bahntangente und der aktuellen Werkzeugrichtung	m	+		PGsl
LFWP	G	Ebene der Rückzugsbewegung beim Schnellabheben wird bestimmt durch die aktuelle Arbeitsebene (G17/G18/G19)	m	+		PGsl
LIFTFAST	K	Schnellabheben		+		PGsl
LIMS	K	Drehzahlbegrenzung bei G96/G961 und G97	m	+		PGsl
LLI	K	Unterer Grenzwert von Variablen		+		PGAsl
LN	F	Natürlicher Logarithmus		+	+	PGAsl
LOCK	P	Synchronaktion mit ID sperren (Technologiezyklus stoppen)		-	+	FBSYsl
LONGHOLE	C	Langloch		+		PGAsl
LOOP	K	Einleitung einer Endlosschleife		+		PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
M0		Programmierer Halt		+	+	PGsl
M1		Wahlweiser Halt		+	+	PGsl
M2		Programmende Hauptprogramm (wie M30)		+	+	PGsl
M3		Spindeldrehrichtung rechts		+	+	PGsl
M4		Spindeldrehrichtung links		+	+	PGsl
M5		Spindel Halt		+	+	PGsl
M6		Werkzeugwechsel		+	+	PGsl
M17		Programmende Unterprogramm		+	+	PGsl
M19		Spindelpositionierung auf die im SD43240 eingetragene Position		+	+	PGsl
M30		Programmende Hauptprogramm (wie M2)		+	+	PGsl
M40		Automatische Getriebeschaltung		+	+	PGsl
M41 ... M45		Getriebestufe 1 ... 5		+	+	PGsl
M70		Übergang in Achsbetrieb		+	+	PGsl
MASLDEF	P	Master/Slave-Achsverband definieren		+	+	PGAsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
MASLDEL	P	Master/Slave-Achsverband trennen und Definition des Verbandes löschen		+	+	PGAsI
MASLOF	P	Ausschalten einer temporären Kopplung		+	+	PGAsI
MASLOFS	P	Ausschalten einer temporären Kopplung mit automatischem Stillsetzen der Slave-Achse		+	+	PGAsI
MASLON	P	Einschalten einer temporären Kopplung		+	+	PGAsI
MATCH	F	Suchen eines String im String		+	-	PGAsI
MAXVAL	F	Größerer Wert zweier Variablen (arithm. Funktion)		+	+	PGAsI
MCALL	K	Modaler Unterprogrammaufruf		+		PGAsI
MEAC	K	Axiales kontinuierliches Messen ohne Restweglöschen	s	+	+	PGAsI
MEAFRAME	F	Frame-Berechnung aus Messpunkten		+	-	PGAsI
MEAS	A	Messen mit Restweglöschen	s	+		PGAsI
MEASA	K	Axiales Messen mit Restweglöschen	s	+	+	PGAsI
MEASURE	F	Berechnungsmethode für die Werkstück- und Werkzeugvermessung		+	-	FB1sI (M5)
MEAW	A	Messen ohne Restweglöschen	s	+		PGAsI
MEAWA	K	Axiales Messen ohne Restweglöschen	s	+	+	PGAsI
MI	K	Zugriff auf Frame-Daten: Spiegelung		+		PGAsI
MINDEX	F	Index eines Zeichens im Eingangsstring bestimmen		+	-	PGAsI
MINVAL	F	Kleinerer Wert zweier Variablen (arithm. Funktion)		+	+	PGAsI
MIRROR	G	Programmierbare Spiegelung	s	+		PGAsI
MMC	P	Aus dem Teileprogramm interaktiv Dialogfenster am HMI aufrufen		+	-	PGAsI
MOD	K	Modulo-Division		+		PGAsI
MODAXVAL	F	Modulo-Position einer Modulo-Rundachse ermitteln		+	-	PGAsI
MOV	K	Positionierachse starten		-	+	FBSYsI
MOVT	A	Endpunkt einer Verfahrbewegung in Werkzeugrichtung angeben				FB1(K2)
MSG	P	Programmierbare Meldungen	m	+	-	PGsI
MVTOOL	P	Sprachbefehl zum Bewegen eines Werkzeugs		+	-	FBWsI
N	A	NC-Nebensatznummer		+		PGsI
NAMETOINT	F	Systemvariablenindex ermitteln		+		PGAsI
NCK	K	Spezifizierung des Gültigkeitsbereichs von Daten		+		PGAsI
NEWCONF	P	Geänderte Maschinendaten übernehmen (entspricht "Maschinendatum wirksam setzen")		+	-	PGAsI
NEWMT	F	Neues Multitool anlegen		+	-	FBWsI
NEWT	F	Neues Werkzeug anlegen		+	-	FBWsI
NORM ⁶⁾	G	Normaleinstellung im Anfangs-, Endpunkt bei der Werkzeugkorrektur	m	+		PGsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
NOT	K	Logisches NICHT (Negation)		+		PGAsl
NPROT	P	Maschinenspezifischer Schutzbereich EIN/AUS		+	-	PGAsl
NPROTDEF	P	Definition eines maschinenspezifischen Schutzbereichs		+	-	PGAsl
NUMBER	F	Eingangsstring in Zahl umwandeln		+	-	PGAsl
OEMIPO1	G	OEM-Interpolation 1	m	+		PGAsl
OEMIPO2	G	OEM-Interpolation 2	m	+		PGAsl
OF	K	Schlüsselwort in der CASE-Verzweigung		+		PGAsl
OFFN	A	Aufmaß zur programmierten Kontur	m	+		PGsl
OMA1	A	OEM-Adresse 1	m	+		PGAsl
OMA2	A	OEM-Adresse 2	m	+		PGAsl
OMA3	A	OEM-Adresse 3	m	+		PGAsl
OMA4	A	OEM-Adresse 4	m	+		PGAsl
OMA5	A	OEM-Adresse 5	m	+		PGAsl
OR	K	Logischer Operator, ODER-Verknüpfung		+		PGAsl
ORIXES	G	Lineare Interpolation der Maschinenachsen oder Orientierungsachsen	m	+		PGAsl
ORIXPOS	G	Orientierungswinkel über virtuelle Orientierungsachsen mit Rundachspositionen	m	+		PGAsl
ORIC ⁶⁾	G	Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert	m	+		PGAsl
ORICONCCW	G	Interpolation auf einer Kreismantelfläche im Gegenuhrzeigersinn	m	+		PGAsl/FB3sl (F3)
ORICONCW	G	Interpolation auf einer Kreismantelfläche im Uhrzeigersinn	m	+		PGAsl/FB3sl (F4)
ORICONIO	G	Interpolation auf einer Kreismantelfläche mit Angabe einer Zwischenorientierung	m	+		PGAsl/FB3sl (F4)
ORICONTO	G	Interpolation auf einer Kreismantelfläche im tangentialen Übergang (Angabe der Endorientierung)	m	+		PGAsl/FB3sl (F5)
ORICURVE	G	Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeugs	m	+		PGAsl/FB3sl (F6)
ORID	G	Orientierungsänderungen werden vor dem Kreissatz ausgeführt	m	+		PGAsl
ORIEULER ⁶⁾	G	Orientierungswinkel über Euler-Winkel	m	+		PGAsl
ORIMKS	G	Werkzeugorientierung im Maschinen-Koordinatensystem	m	+		PGAsl
ORIPATH	G	Werkzeugorientierung bezogen auf die Bahn	m	+		PGAsl
ORIPATHS	G	Werkzeugorientierung bezogen auf die Bahn, ein Knick im Orientierungsverlauf wird geglättet	m	+		PGAsl
ORIPANE	G	Interpolation in einer Ebene (entspricht ORIVECT) Großkreisinterpolation	m	+		PGAsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
ORIRESET	P	Grundstellung der Werkzeugorientierung mit bis zu 3 Orientierungsachsen		+	-	PGAsI
ORIROTA ⁶⁾	G	Drehwinkel zu einer absolut vorgegebenen Drehrichtung	m	+		PGAsI
ORIROTC	G	Tangentialer Drehvektor zur Bahntangente	m	+		PGAsI
ORIROTR	G	Drehwinkel relativ zur Ebene zwischen Start- und Endorientierung	m	+		PGAsI
ORIROTT	G	Drehwinkel relativ zur Änderung des Orientierungsvektors	m	+		PGAsI
ORIRPY	G	Orientierungswinkel über RPY-Winkel (XYZ)	m	+		PGAsI
ORIRPY2	G	Orientierungswinkel über RPY-Winkel (ZYX)	m	+		PGAsI
ORIS	A	Orientierungsänderung	m	+		PGAsI
ORISOF ⁶⁾	G	Glättung des Orientierungsverlaufs AUS	m	+		PGAsI
ORISON	G	Glättung des Orientierungsverlaufs EIN	m	+		PGAsI
ORIVECT ⁶⁾	G	Großkreisinterpolation (identisch mit ORIPLANE)	m	+		PGAsI
ORIVIRT1	G	Orientierungswinkel über virtuelle Orientierungsachsen (Definition 1)	m	+		PGAsI
ORIVIRT2	G	Orientierungswinkel über virtuelle Orientierungsachsen (Definition 1)	m	+		PGAsI
ORIWKS ⁶⁾	G	Werkzeugorientierung im Werkstück-Koordinatensystem	m	+		PGAsI
OS	K	Pendeln ein/aus		+		PGAsI
OSB	K	Pendeln: Startpunkt	m	+		FB1sI (P5)
OSC	G	Konstante Glättung Werkzeugorientierung	m	+		PGAsI
OSCILL	K	Axis: 1 - 3 Zustellachsen	m	+		PGAsI
OSCTRL	K	Optionen pendeln	m	+		PGAsI
OSD	G	Überschleifen der Werkzeugorientierung durch Vorgabe der Überschleiflänge mit SD	m	+		PGAsI
OSE	K	Pendeln Endpunkt	m	+		PGAsI
OSNSC	K	Pendeln: Ausfunktanzahl	m	+		PGAsI
OSOF ⁶⁾	G	Glättung der Werkzeugorientierung AUS	m	+		PGAsI
OSP1	K	Pendeln: linker Umkehrpunkt	m	+		PGAsI
OSP2	K	Pendeln rechter Umkehrpunkt	m	+		PGAsI
OSS	G	Glättung der Werkzeugorientierung am Satzende	m	+		PGAsI
OSSE	G	Glättung der Werkzeugorientierung am Satz-anfang und Satzende	m	+		PGAsI
OST	G	Überschleifen der Werkzeugorientierung durch Vorgabe der Winkeltoleranz in Grad mit dem SD (maximale Abweichung vom programmiert. Orientierungsverlauf)	m	+		PGAsI
OST1	K	Pendeln: Haltepunkt im linken Umkehrpunkt	m	+		PGAsI
OST2	K	Pendeln: Haltepunkt im rechten Umkehrpunkt	m	+		PGAsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
OTOL	K	Orientierungstoleranz für Kompressor-Funktionen, Orientierungsglättung und Überschleifarten		+		PGAsl
OVR	K	Drehzahlkorrektur	m	+		PGAsl
OVRA	K	Axiale Drehzahlkorrektur	m	+	+	PGAsl
OVRRAP	K	Eilgang-Korrektur	m	+		PGAsl
P	A	Anzahl Unterprogrammdurchläufe		+		PGAsl
PAROT	G	Werkstückkoordinatensystem am Werkstück ausrichten	m	+		PGsl
PAROTOF ⁶⁾	G	Werkstückbezogene Frame-Drehung ausschalten	m	+		PGsl
PCALL	K	Unterprogramme mit absoluter Pfadangabe und Parameterübergabe aufrufen		+		PGAsl
PDELAYOF	G	Verzögerung beim Stanzen AUS	m	+		PGAsl
PDELAYON ⁶⁾	G	Verzögerung beim Stanzen EIN	m	+		PGAsl
PHI	K	Drehwinkel der Orientierung um die Richtungssachse des Kegels		+		PGAsl
PHU	K	Physikalische Einheit einer Variablen		+		PGAsl
PL	A	1. B-Spline: Knotenabstand 2. Polynom-Interpolation: Länge des Parameterintervalls bei Polynom-Interpolation	s	+		PGAsl
PM	K	pro Minute		+		PGsl
PO	K	Polynomkoeffizient bei Polynom-Interpolation	s	+		PGAsl
POCKET3	C	Rechtecktasche fräsen		+		PGAsl
POCKET4	C	Kreistasche fräsen		+		PGAsl
POLF	K	Rückzugsposition LIFTFAST	m	+		PGsl/PGAsl
POLFA	P	Rückzugsposition von Einzelachsen mit \$AA_ESR_TRIGGER starten	m	+	+	PGsl
POLFMASK	P	Achsen für den Rückzug ohne Zusammenhang zwischen den Achsen freigeben	m	+	-	PGsl
POLFMLIN	P	Achsen für den Rückzug mit linearem Zusammenhang zwischen den Achsen freigeben	m	+	-	PGsl
POLY	G	Polynom-Interpolation	m	+		PGAsl
POLYPATH	P	Polynom-Interpolation selektierbar für die Achsgruppen AXIS oder VECT	m	+	-	PGAsl
PON	G	Stanzen EIN	m	+		PGAsl
PONS	G	Stanzen EIN im IPO-Takt	m	+		PGAsl
POS	K	Achse positionieren		+	+	PGsl
POSA	K	Achse positionieren über Satzgrenze		+	+	PGsl
POSM	P	Magazin positionieren		+	-	FBWsl
POSMT	P	Multi-tool auf WZ-Halter auf Platznummer positionieren		+	-	FBWsl
POSP	K	Positionieren in Teilstücken (Pendeln)		+		PGsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
POSRANGE	F	Ermitteln, ob sich die aktuell interpolierte Sollposition einer Achse in einem Fenster um eine vorgegebene Referenzposition befindet		+	+	FBSYsl
POT	F	Quadrat (Arithmetische Funktion)		+	+	PGAsl
PR	K	Pro Umdrehung		+		PGsl
PREPRO	PA	Unterprogramme mit Vorbereitung kennzeichnen		+		PGAsl
PRESETON	P	Istwertsetzen mit Verlust des Referenzierstatus		+	+	PGAsl
PRESETONS	P	Istwertsetzen ohne Verlust des Referenzierstatus		+	+	PGAsl
PRIO	K	Schlüsselwort zum Setzen der Priorität bei der Behandlung von Interrupts		+		PGAsl
PRLOC	K	Initialisierung der Variablen bei Reset nur nach lokaler Änderung		+		PGAsl
PROC	K	Erste Anweisung eines Programms		+		PGAsl
PROTA	P	Neuberechnung des Kollisionsmodells anfordern		+		PGAsl
PROTD	F	Abstand zweier Schutzbereiche berechnen		+		PGAsl
PROTS	P	Setzen des Schutzbereichszustandes		+		PGAsl
PSI	K	Öffnungswinkel des Kegels		+		PGAsl
PTP	G	Punkt-zu-Punkt-Bewegung	m	+		PGAsl
PTPG0	G	Punkt-zu-Punkt-Bewegung nur bei G0, sonst CP	m	+		PGAsl
PUNCHACC	P	Wegabhängige Beschleunigung beim Nibbeln		+	-	PGAsl
PUTFTOC	P	Werkzeugfeinkorrektur für paralleles Abrichten		+	-	PGAsl
PUTFTOCF	P	Werkzeugfeinkorrektur in Abhängigkeit einer mit FCTDEF festgelegten Funktion für paralleles Abrichten		+	-	PGAsl
PW	A	B-Spline, Punkt-Gewicht	s	+		PGAsl
QU	K	Schnelle Zusatz-(Hilfs-)funktionsausgabe		+		PGsl
R...	A	Rechenparameter auch als einstellbarer Adressbezeichner und mit numerischer Erweiterung		+		PGAsl
RAC	K	Absolut satzweise achsspezifische Radiusprogrammierung	s	+		PGsl
RDISABLE	P	Einlesesperre		-	+	FBSYsl
READ	P	Liest in der angegebenen Datei eine oder mehrere Zeilen ein und legt gelesene Informationen im Feld ab		+	-	PGAsl
REAL	K	Datentyp: Gleitpunktvariable mit Vorzeichen (reale Zahlen)		+		PGAsl
REDEF	K	Redefinition von Systemvariablen, Anwendervariablen und NC-Sprachbefehlen		+		PGAsl
RELEASE	P	Maschinenachsen zum Achstausch freigeben		+	+	PGAsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
REP	K	Schlüsselwort zur Initialisierung aller Elemente eines Feldes mit demselben Wert		+		PGAsl
REPEAT	K	Wiederholung einer Programmschleife		+		PGAsl
REPEATB	K	Wiederholung einer Programmzeile		+		PGAsl
REPOSA	G	Wiederanfahen an die Kontur linear mit allen Achsen	s	+		PGAsl
REPOSH	G	Wiederanfahen an die Kontur mit Halbkreis	s	+		PGAsl
REPOSHA	G	Wiederanfahen an die Kontur mit allen Achsen; Geometrieachsen im Halbkreis	s	+		PGAsl
REPOSL	G	Wiederanfahen an die Kontur linear	s	+		PGAsl
REPOSQ	G	Wiederanfahen an die Kontur im Viertelkreis	s	+		PGAsl
REPOSQL	G	Wiederanfahen an die Kontur linear mit allen Achsen; Geometrieachsen im Viertelkreis	s	+		PGAsl
RESET	P	Technologiezyklus rücksetzen		-	+	FBSYsl
RESETMON	P	Sprachbefehl zur Sollwertaktivierung		+	-	FBWsl
RET	P	Unterprogrammende		+	+	PGAsl
RETB	P	Unterprogrammende		+	+	PGAsl
RIC	K	Relativ satzweise achsspezifische Radiusprogrammierung	s	+		PGsl
RINDEX	F	Index eines Zeichens im Eingangsstring bestimmen		+	-	PGAsl
RMB	G	Wiederanfahen an Satzanfangspunkt	m	+		PGAsl
RMBBL	G	Wiederanfahen an Satzanfangspunkt	s	+		PGAsl
RME	G	Wiederanfahen an Satzende	m	+		PGAsl
RMEBL	G	Wiederanfahen an Satzende	s	+		PGAsl
RMI ⁶⁾	G	Wiederanfahen an Unterbrechungspunkt	m	+		PGAsl
RMIBL ⁶⁾	G	Wiederanfahen an Unterbrechungspunkt	s	+		PGAsl
RMN	G	Wiederanfahen an nächstliegenden Bahnpunkt	m	+		PGAsl
RMNBL	G	Wiederanfahen an nächstliegenden Bahnpunkt	s	+		PGAsl
RND	A	Konturecke verrunden	s	+		PGsl
RNDM	A	Modales Verrunden	m	+		PGsl
ROT	G	Programmierbare Drehung	s	+		PGsl
ROTS	G	Programmierbare Frame-Drehungen mit Raumwinkeln	s	+		PGsl
ROUND	F	Runden der Nachkommastellen		+	+	PGAsl
ROUNDUP	F	Aufrunden eines Eingabewerts		+	+	PGAsl
RP	A	Polarradius	m/s	+		PGsl
RPL	A	Drehung in der Ebene	s	+		PGsl
RT	K	Parameter für Zugriff auf Framedaten: Drehung		+		PGAsl
RTLIOF	G	G0 ohne Linearinterpolation (Einzelachsinterpolation)	m	+		PGsl
RTLION ⁶⁾	G	G0 mit Linearinterpolation	m	+		PGsl

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
S	A	Spindeldrehzahl (bei G4, G96/G961 andere Bedeutung)	m/s	+	+	PGsI
SAVE	PA	Attribut zur Rettung von Informationen bei Unterprogrammaufrufen		+		PGAsI
SBLOF	P	Einzelsatz unterdrücken		+	-	PGAsI
SBLON	P	Einzelsatzunterdrückung aufheben		+	-	PGAsI
SC	K	Parameter für Zugriff auf Framedaten: Skalierung		+		PGAsI
SCALE	G	Programmierbare Skalierung	s	+		PGsI
SCC	K	Selektive Zuordnung einer Planachse zu G96/G961/G962. Achsbezeichner können Geo-, Kanal oder Maschinenachse sein.		+		PGsI
SCPARA	K	Servo-Parameterersatz programmieren		+	+	PGAsI
SD	A	Spline-Grad	s	+		PGAsI
SET	K	Schlüsselwort zur Initialisierung aller Elemente eines Feldes mit aufgelisteten Werten		+		PGAsI
SETAL	P	Alarm setzen		+	+	PGAsI
SETDNO	F	D-Nummer der Schneide (CE) eines Werkzeugs (T) zuordnen		+	-	PGAsI
SETINT	K	Festlegung, welche Interruptroutine aktiviert werden soll, wenn ein NCK-Eingang ansteht		+		PGAsI
SETM	P	Setzen von Markern im eigenen Kanal		+	+	PGAsI
SETMS	P	Zurückschalten auf die im Maschinendatum festgelegte Masterspindel		+	-	PGsI
SETMS(n)	P	Spindel n soll als Masterspindel gelten		+		PGsI
SETMTH	P	Masterwerkzeughaltnummer setzen		+	-	FBWsl
SETPIECE	P	Stückzahl für alle Werkzeuge berücksichtigen, die der Spindel zugeordnet sind		+	-	FBWsl
SETTA	P	Werkzeug aus Verschleißverbund aktiv setzen		+	-	FBWsl
SETTCOR	F	Veränderung von Werkzeugkomponenten unter Berücksichtigung aller Randbedingungen		+	-	FB1sl (W1)
SETTIA	P	Werkzeug aus Verschleißverbund inaktiv setzen		+	-	FBWsl
SF	A	Startpunktversatz für Gewindeschneiden	m	+		PGsI
SIN	F	Sinus (Trigon. Funktion)		+	+	PGAsI
SIRELAY	F	Die mit SIRELIN, SIRELOUT und SIRELTIME parametrisierten Sicherheitfunktionen aktivieren		-	+	FBSIsI
SIRELIN	P	Eingangsgrößen des Funktionsbausteins initialisieren		+	-	FBSIsI
SIRELOUT	P	Ausgangsgrößen des Funktionsbausteins initialisieren		+	-	FBSIsI
SIRELTIME	P	Timer des Funktionsbausteins initialisieren		+	-	FBSIsI
SLOT1	C	Längsnut		+		PGAsI
SLOT2	C	Kreisnut		+		PGAsI
SOFT	G	Ruckbegrenzte Bahnbeschleunigung	m	+		PGsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
SOFTA	P	Ruckbegrenzte Achsbeschleunigung für die programmierten Achsen einschalten		+	-	PGsl
SON	G	Nibbeln EIN	m	+		PGAsl
SONS	G	Nibbeln EIN im IPO-Takt	m	+		PGAsl
SPATH ⁶⁾	G	Bahnbezug für FGROUP-Achsen ist Bogenlänge	m	+		PGAsl
SPCOF	P	Masterspindel oder Spindel (n) von Lageregelung in Drehzahlregelung umschalten	m	+	-	PGsl
SPCON	P	Masterspindel oder Spindel (n) von Drehzahlregelung in Lageregelung umschalten	m	+	-	PGAsl
SPI	F	Konvertiert Spindelnummer in Achsbezeichner		+	-	PGAsl
SPIF1 ⁶⁾	G	Schnelle NCK-Ein-/Ausgänge für Stanzen/Nibbeln Byte 1	m	+		FB2sl (N4)
SPIF2	G	Schnelle NCK-Ein-/Ausgänge für Stanzen/Nibbeln Byte 2	m	+		FB2sl (N4)
SPLINEPATH	P	Spline-Verband festlegen		+	-	PGAsl
SPN	A	Anzahl der Teilstrecken pro Satz	s	+		PGAsl
SPOF ⁶⁾	G	Hub AUS, Stanzen, Nibbeln AUS	m	+		PGAsl
SPOS	K	Spindelposition	m	+	+	PGsl
SPOSA	K	Spindelposition über Satzgrenzen hinweg	m	+		PGsl
SPP	A	Länge einer Teilstrecke	m	+		PGAsl
SPRINT	F	Liefert einen Eingangsstring formatiert zurück		+		PGAsl
SQRT	F	Quadratwurzel (arithmetische Funktion) (square root)		+	+	PGAsl
SR	A	Pendelrückzugsweg für Synchronaktion	s	+		PGsl
SRA	K	Pendelrückzugsweg bei externem Eingang axial für Synchronaktion	m	+		PGsl
ST	A	Pendelausfeuerzeit für Synchronaktion	s	+		PGsl
STA	K	Pendelausfeuerzeit axial für Synchronaktion	m	+		PGsl
START	P	Starten der ausgewählten Programme in mehreren Kanälen gleichzeitig aus dem laufenden Programm		+	-	PGAsl
STARTFIFO ⁶⁾	G	Abarbeiten; parallel dazu Auffüllen des Vorlaufpuffers	m	+		PGAsl
STAT		Stellung der Gelenke	s	+		PGAsl
STOLF	K	G0-Toleranzfaktor	m	+		PGAsl
STOPFIFO	G	Anhalten der Bearbeitung; Auffüllen des Vorlaufpuffers, bis STARTFIFO erkannt wird, Vorlaufpuffer voll oder Programmende	m	+		PGAsl
STOPRE	P	Vorlaufstopp, bis alle vorbereiteten Sätze vom Hauptlauf abgearbeitet sind		+	-	PGAsl
STOPREOF	P	Vorlaufstopp aufheben		-	+	FBSYsl
STRING	K	Datentyp: Zeichenkette		+		PGAsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
STRINGIS	F	Prüft vorhandenen NC-Sprachumfang und speziell für diesen Befehl gehörende NC-Zyklennamen, Anwendervariablen, Makros und Labelnamen, ob diese existieren, gültig, definiert oder aktiv sind.		+	-	PGAsI
STRLEN	F	Länge eines Strings bestimmen		+	-	PGAsI
SUBSTR	F	Index eines Zeichens im Eingangsstring bestimmen		+	-	PGAsI
SUPA	G	Unterdrückung der aktuellen Nullpunktverschiebung, einschließlich programmierter Verschiebungen, Systemframes, Handradverschiebungen (DRF), externer Nullpunktverschiebung und überlagerte Bewegung	s	+		PGsI
SVC	K	Werkzeug-Schnittgeschwindigkeit	m	+		PGsI
SYNFCT	P	Auswertung eines Polynoms abhängig von einer Bedingung in der Bewegungssynchronaktion		-	+	FBSYsI
SYNR	K	Lesen der Variable erfolgt synchron, d. h. zum Abarbeitungszeitpunkt		+		PGAsI
SYNRW	K	Lesen und Schreiben der Variable erfolgt synchron, d. h. zum Abarbeitungszeitpunkt		+		PGAsI
SYNW	K	Schreiben der Variable erfolgt synchron, d. h. zum Abarbeitungszeitpunkt		+		PGAsI
T	A	Werkzeug aufrufen (wechseln nur, wenn im Maschinendatum festgelegt; ansonsten M6-Befehl nötig)		+		PGsI
TAN	F	Tangens (Trigon. Funktion)		+	+	PGAsI
TANG	P	Definition des Achsverbandes Tangentiales Nachführen		+	-	PGAsI
TANGDEL	P	Löschen der Definition des Achsverbandes Tangentiales Nachführen		+	-	PGAsI
TANGOF	P	Tangentielles Nachführen AUS		+	-	PGAsI
TANGON	P	Tangentielles Nachführen EIN		+	-	PGAsI
TCA (828D: _TCA)	P	Werkzeugwahl / Werkzeugwechsel unabhängig vom Status des Werkzeugs		+	-	FBWsl
TCARR	A	Werkzeugträger (Nummer "m") anfordern		+		PGAsI
TCI	P	Wechsle Werkzeug aus Zwischenspeicher in das Magazin		+	-	FBWsl
TCOABS ⁶⁾	G	Werkzeuglängenkomponenten aus der aktuellen Werkzeugorientierung bestimmen	m	+		PGAsI
TCOFR	G	Werkzeuglängenkomponenten aus der Orientierung des aktiven Frames bestimmen	m	+		PGAsI
TCOFRX	G	Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in X-Richtung	m	+		PGAsI
TCOFRY	G	Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in Y-Richtung	m	+		PGAsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
TCOFRZ	G	Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in Z-Richtung	m	+		PGAsl
THETA	A	Drehwinkel	s	+		PGAsl
TILT	A	Seitwärtswinkel	m	+		PGAsl
TLIFT	P	Bei Tangentialsteuerung Zwischensatz an Konturrecken einfügen		+	-	PGAsl
TML	P	Werkzeuganwahl mit Magazin-Platznummer		+	-	FBWsl
TMOF	P	Werkzeugüberwachung abwählen		+	-	PGAsl
TMON	P	Werkzeugüberwachung anwählen		+	-	PGAsl
TO	K	Bezeichnet den Endwert in einer FOR-Zähl-schleife		+		PGAsl
TOFF	K	Werkzeuglängen-Offset in Richtung der Werkzeuglängenkomponente, die parallel zu der im Index angegebenen Geometrieachse wirkt.	m	+		PGsl
TOFFL	K	Werkzeuglängen-Offset in Richtung der Werkzeuglängenkomponente L1, L2 bzw. L3	m	+		PGsl
TOFFOF	P	Online-Werkzeuglängenkorrektur rücksetzen		+	-	PGAsl
TOFFON	P	Online-Werkzeuglängenkorrektur aktivieren		+	-	PGAsl
TOFFR	A	Werkzeugradius-Offset	m	+		PGsl
TOFRAME	G	Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten	m	+		PGsl
TOFRAMEX	G	X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten	m	+		PGsl
TOFRAMEY	G	Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten	m	+		PGsl
TOFRAMEZ	G	wie TOFRAME	m	+		PGsl
TOLOWER	F	Buchstaben eines Strings in Kleinbuchstaben wandeln		+	-	PGAsl
TOOLENV	F	Alle aktuellen Zustände speichern, die für die Bewertung der im Speicher abgelegten Werkzeugdaten von Bedeutung sind		+	-	FB1sl (W1)
TOOLGNT	F	Anzahl der Werkzeuge einer Werkzeuggruppe ermitteln		+	-	FBWsl
TOOLGT	F	T-Nummer eines Werkzeugs aus einer Werkzeuggruppe ermitteln		+	-	FBWsl
TOROT	G	Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten	m	+		PGsl
TOROTOF ⁶⁾	G	Framedrehungen in Werkzeugrichtung AUS	m	+		PGsl
TOROTX	G	X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten	m	+		PGsl
TOROTY	G	Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten	m	+		PGsl
TOROTZ	G	wie TOROT	m	+		PGsl
TOUPPER	F	Buchstaben eines Strings in Großbuchstaben wandeln		+	-	PGAsl

20.1 Anweisungen

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
1) 2) 3) 4) 5) Erläuterungen siehe Legende (Seite 770).						
TOWBCS	G	Verschleißwerte im Basiskoordinatensystem (BKS)	m	+		PGAsI
TOWKCS	G	Verschleißwerte im Koordinatensystem des Werkzeugkopfes bei kinetischer Transformation (unterscheidet sich vom MKS durch Werkzeugdrehung)	m	+		PGAsI
TOWMCS	G	Verschleißwerte im Maschinen-Koordinatensystem (MKS)	m	+		PGAsI
TOWSTD ⁶⁾	G	Grundstellungswert für Korrekturen in der Werkzeuglänge	m	+		PGAsI
TOWTCS	G	Verschleißwerte im Werkzeug-Koordinatensystem (Werkzeugträgerbezugspunkt T an der Werkzeughalteraufnahme)	m	+		PGAsI
TOWWCS	G	Verschleißwerte im Werkstück-Koordinatensystem (WKS)	m	+		PGAsI
TR	K	Verschiebungskomponente einer Frame-Variablen		+		PGAsI
TRAANG	P	Transformation schräge Achse		+	-	PGAsI
TRACON	P	Kaskadierte Transformation		+	-	PGAsI
TRACYL	P	Zylinder: Mantelflächen-Transformation		+	-	PGAsI
TRAFOOF	P	Im Kanal aktive Transformationen ausschalten		+	-	PGAsI
TRAILOF	P	Achssynchrones Mitschleppen AUS		+	+	PGAsI
TRAILON	P	Achssynchrones Mitschleppen EIN		+	+	PGAsI
TRANS	G	Programmierbare Verschiebung	s	+		PGsI
TRANSMIT	P	Polar-Transformation (Stirnflächenbearbeitung)		+	-	PGAsI
TRAORI	P	4-, 5-Achstransformation, Generische Transformation		+	-	PGAsI
TRUE	K	Logische Konstante: wahr		+		PGAsI
TRUNC	F	Abschneiden der Nachkommastellen		+	+	PGAsI
TU		Achswinkel	s	+		PGAsI
TURN	A	Windungszahl für Schraubenlinie	s	+		PGsI
ULI	K	Oberer Grenzwert von Variablen		+		PGAsI
UNLOCK	P	Synchronaktion mit ID freigeben (Technologiezyklus fortsetzen)		-	+	FBSySl
UNTIL	K	Bedingung zur Beendigung einer REPEAT-Schleife		+		PGAsI
UPATH	G	Bahnbezug für FGROUP-Achsen ist Kurvenparameter	m	+		PGAsI
VAR	K	Schlüsselwort: Art der Parameterübergabe		+		PGAsI
VELOLIM	K	Reduktion der maximalen axialen Geschwindigkeit	m	+		PGAsI
VELOLIMA	K	Reduktion oder Überhöhung der maximalen axialen Geschwindigkeit der Folgeachse	m	+	+	PGAsI
WAITC	P	Warten, bis Kopplungssatzwechselkriterium für die Achsen/Spindeln erfüllt ist		+	-	PGAsI

Anweisung	Art ¹⁾	Bedeutung	W ²⁾	TP ³⁾	SA ⁴⁾	Beschreibung siehe ⁵⁾
^{1) 2) 3) 4) 5)} Erläuterungen siehe Legende (Seite 770).						
WAITE	P	Warten auf das Programmende in einem anderen Kanal.		+	-	PGAsl
WAITENC	P	Warten auf synchronisierte bzw. restaurierte Achspositionen		+	-	PGAsl
WAITM	P	Warten auf Marker im angegebenen Kanal; vorhergehenden Satz mit Genauhalt beenden.		+	-	PGAsl
WAITMC	P	Warten auf Marker im angegeben. Kanal; Genauhalt nur, wenn die anderen Kanäle den Marker noch nicht erreicht haben.		+	-	PGAsl
WAITP	P	Warten auf Verfahrende der Positionierachse		+	-	PGsl
WAITS	P	Warten auf Erreichen der Spindelposition		+	-	PGsl
WALCS0 ⁶⁾	G	WKS-Arbeitsfeldbegrenzung abgewählt	m	+	-	PGsl
WALCS1	G	WKS-Arbeitsfeldbegrenzungsgruppe 1 aktiv	m	+	-	PGsl
WALCS2	G	WKS-Arbeitsfeldbegrenzungsgruppe 2 aktiv	m	+	-	PGsl
WALCS3	G	WKS-Arbeitsfeldbegrenzungsgruppe 3 aktiv	m	+	-	PGsl
WALCS4	G	WKS-Arbeitsfeldbegrenzungsgruppe 4 aktiv	m	+	-	PGsl
WALCS5	G	WKS-Arbeitsfeldbegrenzungsgruppe 5 aktiv	m	+	-	PGsl
WALCS6	G	WKS-Arbeitsfeldbegrenzungsgruppe 6 aktiv	m	+	-	PGsl
WALCS7	G	WKS-Arbeitsfeldbegrenzungsgruppe 7 aktiv	m	+	-	PGsl
WALCS8	G	WKS-Arbeitsfeldbegrenzungsgruppe 8 aktiv	m	+	-	PGsl
WALCS9	G	WKS-Arbeitsfeldbegrenzungsgruppe 9 aktiv	m	+	-	PGsl
WALCS10	G	WKS-Arbeitsfeldbegrenzungsgruppe 10 aktiv	m	+	-	PGsl
WALIMOF	G	BKS-Arbeitsfeldbegrenzung AUS	m	+	-	PGsl
WALIMON ⁶⁾	G	BKS-Arbeitsfeldbegrenzung EIN	m	+	-	PGsl
WHEN	K	Die Aktion wird einmal ausgeführt, wenn die Bedingung einmal erfüllt ist.		-	+	FBSYsl
WHENEVER	K	Die Aktion wird zyklisch in jedem Interpolatortakt ausgeführt, wenn die Bedingung erfüllt ist.		-	+	FBSYsl
WHILE	K	Beginn der WHILE-Programmschleife		+		PGAsl
WRITE	P	Text ins Dateisystem schreiben. Fügt einen Satz am Ende der angegebenen Datei an.		+	-	PGAsl
WRTPR	P	Verzögert den Bearbeitungsauftrag ohne dabei den Bahnsteuerbetrieb zu unterbrechen		+	-	PGAsl
X	A	Achsname	m/s	+	+	PGsl
XOR	O	Logisches Exklusiv- ODER		+		PGAsl
Y	A	Achsname	m/s	+	+	PGsl
Z	A	Achsname	m/s	+	+	PGsl

 Legende

1) Art der Anweisung:

A Adresse

Bezeichner, dem ein Wert zugewiesen wird (z. B. OVR=10). Es gibt auch einige Adressen, die ohne Wertzuweisung eine Funktion ein- oder ausschalten (z. B. CPLON und CPLOF).

C Technologischer Zyklus

Vordefiniertes Teileprogramm, in dem ein bestimmter Zyklus (Bearbeitungsvorgang), wie z. B. das Bohren eines Gewindes oder das Fräsen einer Tasche, allgemeingültig programmiert ist. Die Anpassung an die konkrete Bearbeitungssituation erfolgt über Parameter, die dem Zyklus beim Aufruf übergeben werden.

F Vordefinierte Funktion (liefert Rückgabewert)

Der Aufruf der vordefinierten Funktion kann als Operand im Ausdruck stehen.

G G-Funktion

G-Funktionen sind in Funktionsgruppen eingeteilt. Es kann nur eine G-Funktion einer Gruppe in einem Satz geschrieben werden. Eine G-Funktion kann modal wirksam sein (bis auf Widerruf durch eine andere Funktion derselben Gruppe), oder sie ist nur für den Satz wirksam, in dem sie steht (satzweise wirksam).

K Schlüsselwort

Bezeichner, der die Syntax eines Satzes bestimmt. Einem Schlüsselwort wird kein Wert zugewiesen und mit einem Schlüsselwort kann auch keine NC-Funktion ein-/ausgeschaltet werden.

Beispiele: Kontrollstrukturen (IF, ELSE, ENDIF, WHEN, ...), Programmablauf (GOTOB, GOTO, RET ...)

O Operator

Operator für eine mathematische, Vergleichs- oder logische Operation

P Vordefinierte Prozedur (liefert keinen Rückgabewert)

PA Programmattribut

Programmattribute stehen am Ende der Definitionszeile eines Unterprogramms:

```
PROC <Programmname>(...) <Programmattribut>
```

Sie bestimmen das Verhalten beim Ablauf des Unterprogramms.

2) Wirksamkeit der Anweisung:

m modal

s satzweise

3) Programmierbarkeit im Teileprogramm:

+ programmierbar

- nicht programmierbar

4) Programmierbarkeit in Synchronaktionen:

+ programmierbar

- nicht programmierbar

T programmierbar nur in Technologiezyklen

- 5) Verweis auf das Dokument, das die ausführliche Beschreibung der Anweisung enthält:
- PGsI* Programmierhandbuch Grundlagen
PGAsI Programmierhandbuch Arbeitsvorbereitung
BNMsI Programmierhandbuch Messzyklen
BHDsI Bedienhandbuch Drehen
BHFsl Bedienhandbuch Fräsen
FB1sl () Funktionshandbuch Grundfunktionen (mit dem alphanumerischen Kürzel der betreffenden Funktionsbeschreibung in Klammern)
FB2sl () Funktionshandbuch Erweiterungsfunktionen (mit dem alphanumerischen Kürzel der betreffenden Funktionsbeschreibung in Klammern)
FB3sl () Funktionshandbuch Sonderfunktionen (mit dem alphanumerischen Kürzel der betreffenden Funktionsbeschreibung in Klammern)
FBSIsI Funktionshandbuch Safety Integrated
FBSYsI Funktionshandbuch Synchronaktionen
FBWsl Funktionshandbuch Werkzeugverwaltung
- 6) Standardeinstellung bei Programmanfang (im Auslieferungsstand der Steuerung, wenn nichts anderes programmiert ist).

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
• Standard						
○ Option						
- nicht verfügbar						
:	•	•	•	•	•	•
*	•	•	•	•	•	•
+	•	•	•	•	•	•
-	•	•	•	•	•	•
<	•	•	•	•	•	•
<<	•	•	•	•	•	•
<=	•	•	•	•	•	•
=	•	•	•	•	•	•
>=	•	•	•	•	•	•
/	•	•	•	•	•	•
/0	•	•	•	•	•	•
...						
... /7	○	○	○	○	○	○
A	•	•	•	•	•	•
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
A5	-	-	-	-	-	-
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	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
• Standard ○ Option - nicht verfügbar						
AX	•	•	•	•	•	•
AXCTSWE	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-
AXIS	•	•	•	•	•	•
AXNAME	•	•	•	•	•	•
AXSTRING	•	•	•	•	•	•
AXTOCHAN	•	•	•	•	•	•
AXTOSPI	•	•	•	•	•	•
B	•	•	•	•	•	•
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B_AND	•	•	•	•	•	•
B_OR	•	•	•	•	•	•
B_NOT	•	•	•	•	•	•
B_XOR	•	•	•	•	•	•
BAUTO	-	○	-	○	-	○
BLOCK	•	•	•	•	•	•
BLSYNC	•	•	•	•	•	•
BNAT	-	○	-	○	-	○
BOOL	•	•	•	•	•	•
BOUND	•	•	•	•	•	•
BRISK	•	•	•	•	•	•
BRISKA	•	•	•	•	•	•
BSPLINE	-	○	-	○	-	○
BTAN	-	○	-	○	-	○
C	•	•	•	•	•	•
C2	-	-	-	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
CAC	•	•	•	•	•	•
CACN	•	•	•	•	•	•
CACP	•	•	•	•	•	•
CALCDAT	•	•	•	•	•	•
CALCPOSI	•	•	•	•	•	•
CALL	•	•	•	•	•	•

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
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	-	○	-	○	-	○
CONTDCON	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●
CORROF	●	●	●	●	●	●
COS	●	●	●	●	●	●
COUPDEF	○	-	○	-	○	-
COUPDEL	○	-	○	-	○	-
COUPOF	○	-	○	-	○	-
COUPOFS	○	-	○	-	○	-
COUPON	○	-	○	-	○	-

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
• Standard						
○ Option						
- nicht verfügbar						
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	•	•	•	•	•	•
CPMRESET	•	•	•	•	•	•
CPMSTART	•	•	•	•	•	•
CPMVDI	•	•	•	•	•	•
CPOF	•	•	•	•	•	•
CPON	•	•	•	•	•	•
CPRECOF	•	•	•	•	•	•
CPRECON	•	•	•	•	•	•
CPRES	•	•	•	•	•	•
CPROT	•	•	•	•	•	•
CPROTDEF	•	•	•	•	•	•
CPSETTYPE	•	•	•	•	•	•

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
CTABTMAX	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-
CTOL	-	○	-	○	-	○
CTRANS	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●
CYCLE60	●	●	●	●	●	●
CYCLE61	●	●	●	●	●	●
CYCLE62	●	●	●	●	●	●
CYCLE63	●	●	●	●	●	●
CYCLE64	●	●	●	●	●	●
CYCLE70	●	●	●	●	●	●
CYCLE72	●	●	●	●	●	●
CYCLE76	●	●	●	●	●	●
CYCLE77	●	●	●	●	●	●
CYCLE78	●	●	●	●	●	●
CYCLE79	●	●	●	●	●	●
CYCLE81	●	●	●	●	●	●
CYCLE82	●	●	●	●	●	●
CYCLE83	●	●	●	●	●	●
CYCLE84	●	●	●	●	●	●
CYCLE85	●	●	●	●	●	●
CYCLE86	●	●	●	●	●	●
CYCLE92	●	●	●	●	●	●
CYCLE95	●	●	●	●	●	●
CYCLE98	●	●	●	●	●	●
CYCLE99	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
CYCLE495	●	●	●	●	●	●
CYCLE751	-	-	-	-	-	-
CYCLE752	-	-	-	-	-	-
CYCLE753	-	-	-	-	-	-
CYCLE754	-	-	-	-	-	-
CYCLE755	-	-	-	-	-	-
CYCLE756	-	-	-	-	-	-
CYCLE757	-	-	-	-	-	-
CYCLE758	-	-	-	-	-	-
CYCLE759	-	-	-	-	-	-
CYCLE800	●	●	●	●	●	●
CYCLE801	●	●	●	●	●	●
CYCLE802	●	●	●	●	●	●
CYCLE830	●	●	●	●	●	●
CYCLE832	●	●	●	●	●	●
CYCLE840	●	●	●	●	●	●
CYCLE899	●	●	●	●	●	●
CYCLE930	●	●	●	●	●	●
CYCLE940	●	●	●	●	●	●
CYCLE951	●	●	●	●	●	●
CYCLE952	●	●	●	●	●	●
CYCLE4071	●	●	●	●	●	●
CYCLE4072	●	●	●	●	●	●
CYCLE4073	●	●	●	●	●	●
CYCLE4074	●	●	●	●	●	●
CYCLE4075	●	●	●	●	●	●
CYCLE4077	●	●	●	●	●	●
CYCLE4078	●	●	●	●	●	●
CYCLE4079	●	●	●	●	●	●

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
D	●	●	●	●	●	●
D0	●	●	●	●	●	●
DAC	●	●	●	●	●	●
DC	●	●	●	●	●	●
DCI	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
DCM	●	●	●	●	●	●
DCU	●	●	●	●	●	●
DEF	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●
DELDL	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●
DELETE	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●
DEMLRES	●	●	●	●	●	●
DELMT	●	●	●	●	●	●
DELOBJ	-	-	-	-	-	-
DELT	●	●	●	●	●	●
DELTC	●	●	●	●	●	●
DELTOOLENV	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●
DIC	●	●	●	●	●	●
DILF	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●
DISC	●	●	●	●	●	●
DISCL	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●
DISPR	●	●	●	●	●	●
DISR	●	●	●	●	●	●
DISRP	●	●	●	●	●	●
DITE	●	●	●	●	●	●
DITS	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
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	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●
EXP	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
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	-	-	-	-	-	-
FPR	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●
FRAME	●	●	●	●	●	●
FRC	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
FRCM	●	●	●	●	●	●
FROM	●	●	●	●	●	●
FTOC	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●
FXS	●	●	●	●	●	●
FXST	●	●	●	●	●	●
FXSW	●	●	●	●	●	●
FZ	●	●	●	●	●	●
G0	●	●	●	●	●	●
G1	●	●	●	●	●	●
G2	●	●	●	●	●	●
G3	●	●	●	●	●	●
G4	●	●	●	●	●	●
G5	●	●	●	●	●	●
G7	●	●	●	●	●	●
G9	●	●	●	●	●	●
G17	●	●	●	●	●	●
G18	●	●	●	●	●	●
G19	●	●	●	●	●	●
G25	●	●	●	●	●	●
G26	●	●	●	●	●	●
G33	●	●	●	●	●	●
G34	●	●	●	●	●	●
G35	●	●	●	●	●	●
G40	●	●	●	●	●	●
G41	●	●	●	●	●	●
G42	●	●	●	●	●	●
G53	●	●	●	●	●	●
G54	●	●	●	●	●	●
G55	●	●	●	●	●	●
G56	●	●	●	●	●	●
G57	●	●	●	●	●	●
G58	●	●	●	●	●	●
G59	●	●	●	●	●	●
G60	●	●	●	●	●	●
G62	●	●	●	●	●	●
G63	●	●	●	●	●	●
G64	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
G70	●	●	●	●	●	●
G71	●	●	●	●	●	●
G74	●	●	●	●	●	●
G75	●	●	●	●	●	●
G90	●	●	●	●	●	●
G91	●	●	●	●	●	●
G93	●	●	●	●	●	●
G94	●	●	●	●	●	●
G95	●	●	●	●	●	●
G96	●	●	●	●	●	●
G97	●	●	●	●	●	●
G110	●	●	●	●	●	●
G111	●	●	●	●	●	●
G112	●	●	●	●	●	●
G140	●	●	●	●	●	●
G141	●	●	●	●	●	●
G142	●	●	●	●	●	●
G143	●	●	●	●	●	●
G147	●	●	●	●	●	●
G148	●	●	●	●	●	●
G153	●	●	●	●	●	●
G247	●	●	●	●	●	●
G248	●	●	●	●	●	●
G290	●	●	●	●	●	●
G291	●	●	●	●	●	●
G331	●	●	●	●	●	●
G332	●	●	●	●	●	●
G335	●	●	●	●	●	●
G336	●	●	●	●	●	●
G340	●	●	●	●	●	●
G341	●	●	●	●	●	●
G347	●	●	●	●	●	●
G348	●	●	●	●	●	●
G450	●	●	●	●	●	●
G451	●	●	●	●	●	●
G460	●	●	●	●	●	●
G461	●	●	●	●	●	●
G462	●	●	●	●	●	●
G500	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
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	●	●	●	●	●	●

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
GET	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●
GETD	●	●	●	●	●	●
GETDNO	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●
GETT	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
• Standard						
○ Option						
- nicht verfügbar						
GETVARAP	•	•	•	•	•	•
GETVARDFT	•	•	•	•	•	•
GETVARLIM	•	•	•	•	•	•
GETVARPHU	•	•	•	•	•	•
GETVARTYP	•	•	•	•	•	•
GOTO	•	•	•	•	•	•
GOTOB	•	•	•	•	•	•
GOTOC	•	•	•	•	•	•
GOTOF	•	•	•	•	•	•
GOTOS	•	•	•	•	•	•
GP	•	•	•	•	•	•
GWPSOF	•	•	•	•	•	•
GWPSON	•	•	•	•	•	•
H...	•	•	•	•	•	•
HOLES1	•	•	•	•	•	•
HOLES2	•	•	•	•	•	•
I	•	•	•	•	•	•
I1	•	•	•	•	•	•
IC	•	•	•	•	•	•
ICYCOF	•	•	•	•	•	•
ICYCON	•	•	•	•	•	•
ID	•	•	•	•	•	•
IDS	•	•	•	•	•	•
IF	•	•	•	•	•	•
INDEX	•	•	•	•	•	•
INIPO	•	•	•	•	•	•
INIRE	•	•	•	•	•	•
INICF	•	•	•	•	•	•
INIT	-	-	-	-	-	-
INITIAL	•	•	•	•	•	•
INT	•	•	•	•	•	•
INTERSEC	•	•	•	•	•	•
INVCCW	-	-	-	-	-	-
INVCW	-	-	-	-	-	-
INVFRAME	•	•	•	•	•	•
IP	•	•	•	•	•	•
IPOBRKA	•	•	•	•	•	•
IPOENDA	•	•	•	•	•	•
IPTRLOCK	•	•	•	•	•	•

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
IPTRUNLOCK	●	●	●	●	●	●
IR	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●
ISD	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●
ISNUMBER	●	●	●	●	●	●
ISOCALL	●	●	●	●	●	●
ISVAR	●	●	●	●	●	●
J	●	●	●	●	●	●
J1	●	●	●	●	●	●
JERKA	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●
JR	●	●	●	●	●	●
K	●	●	●	●	●	●
K1	●	●	●	●	●	●
KONT	●	●	●	●	●	●
KONTC	●	●	●	●	●	●
KONTT	●	●	●	●	●	●
KR	●	●	●	●	●	●
L	●	●	●	●	●	●
LEAD						
Werkzeugorientierung	-	-	-	-	-	-
Orientierungspolyn.	-	-	-	-	-	-
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
● Standard ○ Option - nicht verfügbar						
M0	●	●	●	●	●	●
M1	●	●	●	●	●	●
M2	●	●	●	●	●	●
M3	●	●	●	●	●	●
M4	●	●	●	●	●	●
M5	●	●	●	●	●	●
M6	●	●	●	●	●	●
M17	●	●	●	●	●	●
M19	●	●	●	●	●	●
M30	●	●	●	●	●	●
M40	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●
M70	●	●	●	●	●	●
MASLDEF	●	●	●	●	●	●
MASLDEL	●	●	●	●	●	●
MASLOF	●	●	●	●	●	●
MASLOFS	●	●	●	●	●	●
MASLON	●	●	●	●	●	●
MATCH	●	●	●	●	●	●
MAXVAL	●	●	●	●	●	●
MCALL	●	●	●	●	●	●
MEAC	-	-	-	-	-	-
MEAFRAME	●	●	●	●	●	●
MEAS	●	●	●	●	●	●
MEASA	-	-	-	-	-	-
MEASURE	●	●	●	●	●	●
MEAW	●	●	●	●	●	●
MEAWA	-	-	-	-	-	-
MI	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●
MMC	●	●	●	●	●	●
MOD	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●
MOV	●	●	●	●	●	●
MOVT	●	●	●	●	●	●
MSG	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
N	●	●	●	●	●	●
NAMETOINT	-	-	-	-	-	-
NCK	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●
NEWT	●	●	●	●	●	●
NORM	●	●	●	●	●	●
NOT	●	●	●	●	●	●
NPROT	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-
OF	●	●	●	●	●	●
OFFN	●	●	●	●	●	●
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
OR	●	●	●	●	●	●
ORIXES	-	-	-	-	-	-
ORIXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
ORIROTT	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-
ORIS	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-
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	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
● Standard ○ Option - nicht verfügbar						
PL	-	○	-	○	-	○
	-	-	-	-	-	-
PM	●	●	●	●	●	●
PO	-	-	-	-	-	-
POCKET3	●	●	●	●	●	●
POCKET4	●	●	●	●	●	●
POLF	●	●	●	●	●	●
POLFA	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●
POLY	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-
PON	-	-	-	-	-	-
PONS	-	-	-	-	-	-
POS	●	●	●	●	●	●
POSA	●	●	●	●	●	●
POSM	●	●	●	●	●	●
POSMT	●	●	●	●	●	●
POSP	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●
POT	●	●	●	●	●	●

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
● Standard ○ Option - nicht verfügbar						
PR	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●
PRIO	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●
PROC	●	●	●	●	●	●
PROTA	-	-	-	-	-	-
PROTD	-	-	-	-	-	-
PROTS	-	-	-	-	-	-
PSI	-	-	-	-	-	-
PTP	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
• Standard ○ Option - nicht verfügbar						
PUNCHACC	-	-	-	-	-	-
PUTFTOC	•	•	•	•	•	•
PUTFTOCF	•	•	•	•	•	•
PW	-	○	-	○	-	○
QU	•	•	•	•	•	•
R...	•	•	•	•	•	•
RAC	•	•	•	•	•	•
RDISABLE	•	•	•	•	•	•
READ	•	•	•	•	•	•
REAL	•	•	•	•	•	•
REDEF	•	•	•	•	•	•
RELEASE	•	•	•	•	•	•
REP	•	•	•	•	•	•
REPEAT	•	•	•	•	•	•
REPEATB	•	•	•	•	•	•
REPOSA	•	•	•	•	•	•
REPOSH	•	•	•	•	•	•
REPOSHA	•	•	•	•	•	•
REPOSL	•	•	•	•	•	•
REPOSQ	•	•	•	•	•	•
REPOSQA	•	•	•	•	•	•
RESET	•	•	•	•	•	•
RESETMON	•	•	•	•	•	•
RET	•	•	•	•	•	•
RETB	•	•	•	•	•	•
RIC	•	•	•	•	•	•
RINDEX	•	•	•	•	•	•
RMB	•	•	•	•	•	•
RME	•	•	•	•	•	•
RMI	•	•	•	•	•	•
RMN	•	•	•	•	•	•
RND	•	•	•	•	•	•
RNDM	•	•	•	•	•	•
ROT	•	•	•	•	•	•
ROTS	•	•	•	•	•	•
ROUND	•	•	•	•	•	•
ROUNDUP	•	•	•	•	•	•
RP	•	•	•	•	•	•
RPL	•	•	•	•	•	•

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
RT	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●
RTLION	●	●	●	●	●	●
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	●	●	●	●	●	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
• Standard ○ Option - nicht verfügbar						
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	•	•	•	•	•	•
STRINGVAR	-	-	-	-	-	-
STRLEN	•	•	•	•	•	•
SUBSTR	•	•	•	•	•	•
SUPA	•	•	•	•	•	•
SVC	•	•	•	•	•	•
SYNFCT	•	•	•	•	•	•
SYNR	•	•	•	•	•	•
SYNRW	•	•	•	•	•	•
SYNW	•	•	•	•	•	•
T	•	•	•	•	•	•
TAN	•	•	•	•	•	•
TANG	-	-	-	-	-	-
TANGDEL	-	-	-	-	-	-
TANGOF	-	-	-	-	-	-

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
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	-	●	-	●	-	●
TOWMCS	-	●	-	●	-	●

20.2 Anweisungen: Verfügbarkeit bei SINUMERIK 828D

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
TOWSTD	-	●	-	●	-	●
TOWTCS	-	●	-	●	-	●
TOWWCS	-	●	-	●	-	●
TR	●	●	●	●	●	●
TRAANG	-	-	-	-	○	-
TRACON	-	-	-	-	○	-
TRACYL	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●
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	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●

20.3 Aktuelle Sprache im HMI

Anweisung ● Standard ○ Option - nicht verfügbar	828D-Steuerungsvariante					
	PPU240.3 / 241.3		PPU260.3 / 261.3		PPU280.3 / 281.3	
	Drehen	Fräsen	Drehen	Fräsen	Drehen	Fräsen
WALCS8	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●
WHEN	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●
WHILE	●	●	●	●	●	●
WRITE	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●
X	●	●	●	●	●	●
XOR	●	●	●	●	●	●
Y	●	●	●	●	●	●
Z	●	●	●	●	●	●

20.3 Aktuelle Sprache im HMI

Die folgende Tabelle enthält alle auf der Bedienoberfläche verfügbaren Sprachen.

Die aktuell eingestellte Sprache ist im Teileprogramm und in Synchronaktionen über folgende Systemvariable abfragbar:

\$AN_LANGUAGE_ON_HMI = <Wert>

<Wert>	Sprache	Sprachkürzel
1	Deutsch (Deutschland)	DEU
2	Französisch	FRA
3	Englisch (Vereinigtes Königreich)	ENG
4	Spanisch	ESP
6	Italienisch	ITA
7	Niederländisch	NLD
8	Chinesisch (vereinfacht)	CHS
9	Schwedisch	SVE
18	Ungarisch	HUN
19	Finnisch	FIN
28	Tschechisch	CSY
50	Portugiesisch (Brasilien)	PTB
53	Polnisch	PLK
55	Dänisch	DAN
57	Russisch	RUS

<Wert>	Sprache	Sprachkürzel
68	Slowakisch	SKY
72	Rumänisch	ROM
80	Chinesisch (traditionell)	CHT
85	Koreanisch	KOR
87	Japanisch	JPN
89	Türkisch	TRK

Hinweis

Eine Aktualisierung von \$AN_LANGUAGE_ON_HMI erfolgt:

- nach Systemhochlauf.
 - nach NCK- und/oder PLC-Reset.
 - nach dem Umschalten auf einen anderen NCK im Rahmen von M2N.
 - nach Sprachumschaltung auf HMI.
-

A.1 Liste der Abkürzungen

A	
A	Ausgang
ADI4	Analog Drive Interface for 4 Axes
AC	Adaptive Control
ALM	Active Line Module
ARM	Asynchroner rotatorischer Motor
AS	Automatisierungssystem
ASCII	American Standard Code for Information Interchange: Amerikanische Code-Norm für den Informationsaustausch
ASIC	Application Specific Integrated Circuit: Anwender-Schaltkreis
ASUP	Asynchrones Unterprogramm
AUXFU	Auxiliary Function: Hilfsfunktion
AWL	Anweisungsliste
AWP	Anwenderprogramm

B	
BA	Betriebsart
BAG	Betriebsartengruppe
BCD	Binary Coded Decimals: Im Binärcode verschlüsselte Dezimalzahlen
BERO	Berührungsloser Näherungsschalter
BI	Binector Input
BICO	Binector Connector
BIN	Binary Files: Binärdateien
BIOS	Basic Input Output System
BKS	Basiskoordinatensystem
BO	Binector Output
BTSS	Bedientafelschnittstelle

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Compile-Zyklen
CI	Connector Input
CF-Card	Compact Flash-Card
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung

Anhang

A.1 Liste der Abkürzungen

C	
CO	Connector Output
CoL	Certificate of License
COM	Communication
CPA	Compiler Projecting Data: Projektierdaten des Compilers
CRT	Cathode Ray Tube: Bildröhre
CSB	Central Service Board: PLC-Baugruppe
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: Zentrale Rechereinheit
CR	Carriage Return
CTS	Clear To Send: Meldung der Sendebereitschaft bei seriellen Daten-Schnittstellen
CUTCOM	Cutter Radius Compensation: Werkzeugradiuskorrektur

D	
DAU	Digital-Analog-Umwandler
DB	Datenbaustein (PLC)
DBB	Datenbaustein-Byte (PLC)
DBD	Datenbaustein-Doppelwort (PLC)
DBW	Datenbaustein-Wort (PLC)
DBX	Datenbaustein-Bit (PLC)
DDE	Dynamic Data Exchange
DDS	Drive Data Set: Antriebsdatensatz
DIN	Deutsche Industrie Norm
DIO	Data Input/Output: Datenübertragungs-Anzeige
DIR	Directory: Verzeichnis
DLL	Dynamic Link Library
DO	Drive Object
DPM	Dual Port Memory
DPR	Dual Port RAM
DRAM	Dynamischer Speicher (ungepuffert)
DRF	Differential Resolver Function: Differential-Drehmelder-Funktion (Handrad)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run: Probelaufvorschub
DSB	Decoding Single Block: Dekodierungseinzelsatz
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Datenwort
DWORD	Doppelwort (aktuell 32 Bit)

E	
E	Eingang
EES	Execution from External Storage

E	
E/A	Ein-/Ausgabe
ENC	Encoder: Istwertgeber
EFP	Einfach Peripheriemodul (PLC–E/A–Baugruppe)
EGB	Elektronisch gefährdete Baugruppen/Bauelemente
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Norm
ENC	Encoder: Istwertgeber
EnDat	Geberschnittstelle
EPROM	Erasable Programmable Read Only Memory: Löschbarer, elektrisch programmierbarer nur Lesespeicher
ePS Network Services	Dienste zur internetgestützten Maschinen-Fernwartung
EQN	Typbezeichnung eines Absolutwertgebers mit 2048 Sinussignalen/Umdrehung
ES	Engineering System
ESR	Erweitertes Stillsetzen und Rückziehen
ETC	ETC–Taste ">"; Erweiterung der Softkeyleiste im gleichen Menü

F	
FB	Funktionsbaustein (PLC)
FC	Function Call: Funktionsbaustein (PLC)
FEPROM	Flash–EPROM: Les– und schreibbarer Speicher
FIFO	First In First Out: Speicher, der ohne Adressangabe arbeitet und dessen Daten in derselben Reihenfolge gelesen werden, in der sie gespeichert wurden
FIPO	Feininterpolator
FPU	Floating Point Unit: Gleitpunkteinheit
FRK	Fräsradiuskorrektur
FST	Feed Stop: Vorschub Halt
FUP	Funktionsplan (Programmiermethode für PLC)
FW	Firmware

G	
GC	Global Control (PROFIBUS: Broadcast-Telegramm)
GDIR	Globaler Teileprogrammspeicher
GEO	Geometrie, z.B. Geometrieachse
GIA	Gear Interpolation Data: Getriebeinterpolationsdaten
GND	Signal Ground
GP	Grundprogramm (PLC)
GS	Getriebestufe
GSD	Gerätestammdatei zur Beschreibung eines PROFIBUS Slaves
GSDML	Generic Station Description Markup Language: XML-basierte Beschreibungssprache zur Erstellung einer GSD-Datei
GUD	Global User Data: Globale Anwenderdaten

H	
HEX	Kurzbezeichnung für hexadezimale Zahl
HiFu	Hilfsfunktion
HLA	Hydraulischer Linearantrieb
HMI	Human Machine Interface: SINUMERIK-Bedienoberfläche
HSA	Hauptspindeltrieb
HW	Hardware

I	
IBN	Inbetriebnahme
IKA	Interpolatorische Kompensation
IM	Interface-Modul: Anschaltungsbaugruppe
IMR	Interface-Modul Receive: Anschaltungsbaugruppe für Empfangsbetrieb
IMS	Interface-Modul Send: Anschaltungsbaugruppe für Sendebetrieb
INC	Increment: Schrittmaß
INI	Initializing Data: Initialisierungsdaten
IPO	Interpolator
ISA	International Standard Architecture
ISO	International Standard Organization

J	
JOG	Jogging: Einrichtbetrieb

K	
K_V	Verstärkungsfaktor des Regelkreises
K_P	Proportionalverstärkung
K_U	Übersetzungsverhältnis
KOP	Kontaktplan (Programmiermethode für PLC)

L	
LAI	Logic Machine Axis Image: Logisches Maschinenachsen-Abbild
LAN	Local Area Network
LCD	Liquid-Crystal Display: Flüssigkristallanzeige
LED	Light Emitting Diode: Leuchtdiode
LF	Line Feed
LMS	Lagemesssystem
LR	Lageregler
LSB	Least Significant Bit: Niederwertigstes Bit
LUD	Local User Data: Anwenderdaten (lokal)

M	
MAC	Media Access Control
MAIN	Main program: Hauptprogramm (OB1, PLC)
MB	Megabyte
MCI	Motion Control Interface
MCIS	Motion–Control–Information–System
MCP	Machine Control Panel: Maschinensteuertafel
MD	Maschinendatum bzw. Maschinendaten
MDA	Manual Data Automatic: Handeingabe
MDS	Motor Data Set: Motordatensatz
MELDW	Meldungswort
MKS	Maschinenkoordinatensystem
MM	Motor Module
MPF	Main Program File: Hauptprogramm (NC)
MSTT	Maschinensteuertafel

N	
NC	Numerical Control: Numerische Steuerung
NCK	Numerical Control Kernel: Numerik-Kern mit Satzaufbereitung, Verfahrbereich usw.
NCU	Numerical Control Unit: Hardware-Einheit des NCK
NRK	Bezeichnung des Betriebssystems des NCK
NST	Nahtstellensignal
NURBS	Non-Uniform Rational B-Spline
NV	Nullpunktverschiebung
NX	Numerical Extension: Achserweiterungsbaugruppe

O	
OB	Organisationsbaustein in der PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Bedieneinrichtung
OPI	Operation Panel Interface: Bedientafel-Anschaltung
OPT	Options: Optionen
OLP	Optical Link Plug: Busstecker für Lichtleiter
OSI	Open Systems Interconnection: Normung für Rechnerkommunikation

P	
PAA	Prozessabbild der Ausgänge
PAE	Prozessabbild der Eingänge
PC	Personal Computer
PCIN	Name der SW für den Datenaustausch mit der Steuerung

P	
PCMCIA	Personal Computer Memory Card International Association: Speichersteckkarten-Normierung
PCU	PC Unit: PC-Box (Rechneinheit)
PG	Programmiergerät
PKE	Parameterkennung: Teil eines PKW
PKW	Parameterkennung: Wert (Parametrierteil eines PPO)
PLC	Programmable Logic Control: Anpass-Steuerung
PN	PROFINET
PNO	PROFIBUS-Nutzerorganisation
PO	POWER ON
POE	Programmorganisationseinheit
POS	Position/Positionieren
POSMO A	Positioning Motor Actuator: Positioniermotor
POSMO CA	Positioning Motor Compact AC: Komplette Antriebseinheit mit integrierter Leistungs- und Reglungsbaugruppe sowie Positioniereinheit und Programmspeicher; Wechselstrom-Einspeisung
POSMO CD	Positioning Motor Compact DC: wie CA, jedoch Gleichstromspeisung
POSMO SI	Positioning Motor Servo Integrated: Positioniermotor; Gleichstromspeisung
PPO	Parameter Prozessdaten Objekt ; Zyklisches Datentelegramm bei der Übertragung mit PROFIBUS-DP und Profil "Drehzahlveränderbare Antriebe"
PPU	Panel Processing Unit (zentrale Hardware einer Panel-basierten CNC-Steuerung z.B. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Serieller Datenbus
PRT	Programmtest
PSW	Programmsteuerwort
PTP	Point to Point: Punkt zu Punkt
PUD	Program Global User Data: Programmglobale Anwendervariable
PZD	Prozessdaten: Prozessdatenteil eines PPO

Q	
QFK	Quadrantenfehler Kompensation

R	
RAM	Random Access Memory: Schreib-/Lese-Speicher
REF	Funktion Referenzpunkt anfahren
REPOS	Funktion Repositionieren
RISC	Reduced Instruction Set Computer: Prozessortyp mit kleinem Befehlssatz und schnellem Befehlsdurchsatz
ROV	Rapid Override: Eingangskorrektur
RP	R-Parameter, Rechenparameter, vordefinierte Anwendervariable
RPA	R-Parameter Active: Speicherbereich in NCK für R-Parameternummern
RPY	Roll Pitch Yaw: Drehungsart eines Koordinatensystems
RTL	Rapid Traverse Linear Interpolation: Lineare Interpolation bei Eilgangbewegung

R	
RTS	Request To Send: Sendeteil einschalten, Steuersignal von seriellen Daten-Schnittstellen
RTCP	Real Time Control Protocol

S	
SA	Synchronaktion
SBC	Safe Break Control: Sichere Bremsenansteuerung
SBL	Single Block: Einzelsatz
SBR	Subroutine: Unterprogramm (PLC)
SD	Settingdatum bzw. Settingdaten
SDB	System Datenbaustein
SEA	Setting Data Active: Kennzeichnung (Dateityp) für Settingdaten
SERUPRO	Search-Run by Program Test: Suchlauf via Programmtest
SFB	System Funktionsbaustein
SFC	System Function Call
SGE	Sicherheitsgerichteter Eingang
SGA	Sicherheitsgerichteter Ausgang
SH	Sicherer Halt
SIM	Single in Line Module
SK	Softkey
SKP	Skip: Funktion zum Ausblenden eines Teileprogrammsatzes
SLM	Synchroner Linearmotor
SM	Schrittmotor
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Sub Program File: Unterprogramm (NC)
SPS	Speicherprogrammierbare Steuerung = PLC
SRAM	Statischer Speicher (gepuffert)
SRK	Schneidenradiuskorrektur
SRM	Synchron rotatorischer Motor
SSFK	Spindelsteigungsfehlerkompensation
SSI	Serial Synchron Interface: Serielle synchrone Schnittstelle
SSL	Satzsuchlauf
STW	Steuerwort
SUG	Scheibenumfangsgeschwindigkeit
SW	Software
SYF	System Files: Systemdateien
SYNACT	Synchronized Action: Synchronaktion

T	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point: Werkzeugspitze
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: Kennung für Maschinendaten
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Werkzeugkorrektur
TOA	Tool Offset Active: Kennzeichnung (Dateityp) für Werkzeugkorrekturen
TRANSMIT	Transform Milling Into Turning: Koordinatentransformation für Fräsbearbeitungen an einer Drehmaschine
TTL	Transistor–Transistor–Logik (Schnittstellen–Typ)
TZ	Technologiezyklus

U	
UFR	User Frame: Nullpunktverschiebung
UP	Unterprogramm
USB	Universal Serial Bus
USV	Unterbrechungsfreie Stromversorgung

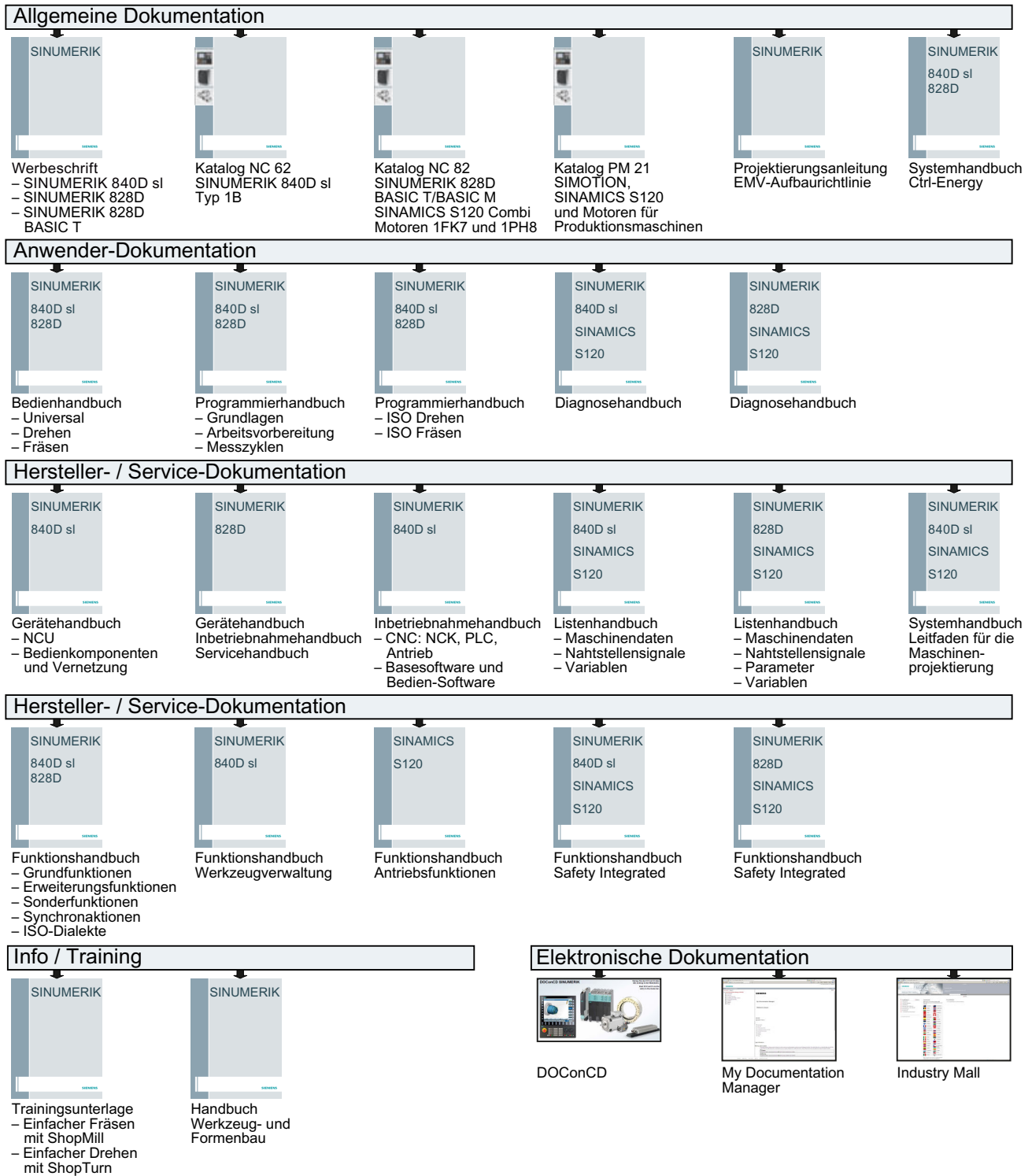
V	
VDI	Interne Kommunikationsschnittstelle zwischen NCK und PLC
VDI	Verein Deutscher Ingenieure
VDE	Verband Deutscher Elektrotechniker
VI	Voltage Input
VO	Voltage Output
VSA	Vorschubantrieb

W	
WAB	Funktion Weiches An- und Abfahren
WKS	Werkstückkoordinatensystem
WKZ	Werkzeug
WLK	Werkzeuglängenkorrektur
WOP	Werkstatt-orientierte Programmierung
WPD	Work Piece Directory: Werkstückverzeichnis
WRK	Werkzeug-Radius-Korrektur
WZ	Werkzeug
WZK	Werkzeugkorrektur
WZV	Werkzeugverwaltung
WZW	Werkzeugwechsel

X	
XML	Extensible Markup Language

Z	
ZOA	Zero Offset Active: Kennung für Nullpunktverschiebungen
ZSW	Zustandswort (des Antriebs)

A.2 Dokumentationsübersicht



Glossar

Absolutmaß

Angabe des Bewegungsziels einer Achsbewegung durch ein Maß, das sich auf den Nullpunkt des momentan gültigen Koordinatensystems bezieht. Siehe → Kettenmaß.

Achsadresse

Siehe → Achsname

Achsen

Die CNC-Achsen werden entsprechend ihres Funktionsumfangs abgestuft in:

- Achsen: interpolierende Bahnachsen
- Hilfsachsen: nicht interpolierende Zustell- und Positionierachsen mit achsspezifischem Vorschub. Hilfsachsen sind an der eigentlichen Bearbeitung nicht beteiligt, z. B. Werkzeugzubringer, Werkzeugmagazin.

Achsname

Zur eindeutigen Identifikation müssen alle Kanal- und → Maschinenachsen der Steuerung mit kanal- bzw. steuerungsweit eindeutigen Namen bezeichnet werden. Die → Geometrieachsen werden mit X, Y, Z benannt. Die um die Geometrieachsen drehenden → Rundachsen werden mit A, B, C benannt.

Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden oder Operandenbereich, z. B. Eingang, Ausgang usw.

Alarmer

Alle → Meldungen und Alarmer werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

1. Alarmer und Meldungen im Teileprogramm
Alarmer und Meldungen können direkt aus dem Teileprogramm im Klartext zur Anzeige gebracht werden.
2. Alarmer und Meldungen von PLC
Alarmer- und Meldungen der Maschine können aus dem PLC-Programm im Klartext zur Anzeige gebracht werden. Dazu sind keine zusätzlichen Funktionsbaustein-Pakete notwendig.

Antrieb

Der Antrieb ist diejenige Einheit der CNC, welche die Drehzahl- und Momentenregelung aufgrund der Vorgaben der NC ausführt.

Anwenderdefinierte Variable

Anwender können für beliebige Nutzung im → Teileprogramm oder Datenbaustein (globale Anwenderdaten) anwenderdefinierte Variablen vereinbaren. Eine Definition enthält eine Datentypangabe und den Variablennamen. Siehe → Systemvariable.

Anwenderprogramm

Anwenderprogramme für Automatisierungssysteme S7-300 werden mit der Programmiersprache STEP 7 erstellt. Das Anwenderprogramm ist modular aufgebaut und besteht aus einzelnen Bausteinen.

Die grundlegenden Bausteintypen sind:

- Code-Bausteine
Diese Bausteine enthalten die STEP 7-Befehle.
- Datenbausteine
Diese Bausteine enthalten Konstanten und Variablen für das STEP 7-Programm.

Anwenderspeicher

Alle Programme und Daten wie Teileprogramme, Unterprogramme, Kommentare, Werkzeugkorrekturen, Nullpunktverschiebungen/Frames sowie Kanal- und Programmanwenderdaten können in den gemeinsamen CNC-Anwenderspeicher abgelegt werden.

Arbeitsfeldbegrenzung

Mit der Arbeitsfeldbegrenzung kann der Verfahrbereich der Achsen zusätzlich zu den Endschaltern eingeschränkt werden. Je Achse ist ein Wertepaar zur Beschreibung des geschützten Arbeitsraumes möglich.

Arbeitsraum

Dreidimensionaler Raum, in den die Werkzeugspitze aufgrund der Konstruktion der Werkzeugmaschine hineinfahren kann. Siehe → Schutzraum.

Arbeitsspeicher

Der Arbeitsspeicher ist ein RAM-Speicher in der → CPU, auf den der Prozessor während der Programmbearbeitung auf das Anwenderprogramm zugreift.

Archivieren

Auslesen von Dateien und/oder Verzeichnissen auf ein **externes** Speichergerät.

Asynchrones Unterprogramm

Teileprogramm, das asynchron (unabhängig) zum aktuellen Programmzustand durch ein Interruptsignal (z. B. Signal "schneller NC-Eingang") gestartet werden kann.

Automatik

Betriebsart der Steuerung (Satzfolgebetrieb nach DIN): Betriebsart bei NC-Systemen, in der ein → Teileprogramm angewählt und kontinuierlich abgearbeitet wird.

Bahnachse

Bahnachsen sind alle Bearbeitungsachsen des → Kanals, die vom → Interpolator so geführt werden, dass sie gleichzeitig starten, beschleunigen, stoppen und den Endpunkt erreichen.

Bahngeschwindigkeit

Die maximal programmierbare Bahngeschwindigkeit ist abhängig von der Eingabefineinheit. Bei einer Auflösung von beispielsweise 0,1 mm beträgt die maximal programmierbare Bahngeschwindigkeit 1000 m/min.

Bahnsteuerbetrieb

Ziel des Bahnsteuerbetriebes ist es, ein größeres Abbremsen der → Bahnachsen an den Teileprogramm-Satzgrenzen zu vermeiden und mit möglichst gleicher Bahngeschwindigkeit in den nächsten Satz zu wechseln.

Bahnvorschub

Bahnvorschub wirkt auf → Bahnachsen. Er stellt die geometrische Summe der Vorschübe der beteiligten → Geometrieachsen dar.

Basisachse

Achse, deren Soll- oder Istwert für die Berechnung eines Kompensationswertes herangezogen wird.

Basiskoordinatensystem

Kartesisches Koordinatensystem, wird durch Transformation auf das Maschinenkoordinatensystem abgebildet.

Im → Teileprogramm verwendet der Programmierer Achsnamen des Basiskoordinatensystems. Es besteht, wenn keine → Transformation aktiv ist, parallel zum → Maschinenkoordinatensystem. Der Unterschied zu diesem liegt in den → Achsnamen.

Baudrate

Geschwindigkeit bei der Datenübertragung (Bit/s).

Baustein

Als Bausteine werden alle Dateien bezeichnet, die für die Programmerstellung und Programmverarbeitung benötigt werden.

Bearbeitungskanal

Über eine Kanalstruktur können durch parallele Bewegungsabläufe Nebenzeiten verkürzt werden, z. B. Verfahren eines Ladeportals simultan zur Bearbeitung. Ein CNC-Kanal ist dabei als eigene CNC-Steuerung mit Dekodierung, Satzaufbereitung und Interpolation anzusehen.

Bedienoberfläche

Die Bedienoberfläche (BOF) ist das Anzeigemedium einer CNC-Steuerung in Gestalt eines Bildschirms. Sie ist mit horizontalen und vertikalen Softkeys gestaltet.

Beschleunigung mit Ruckbegrenzung

Zur Erzielung eines optimalen Beschleunigungsverhaltens an der Maschine bei gleichzeitiger Schonung der Mechanik kann im Bearbeitungsprogramm zwischen sprunghafter Beschleunigung und stetiger (ruckfreier) Beschleunigung umgeschaltet werden.

Betriebsart

Ablaufkonzept für den Betrieb einer SINUMERIK-Steuerung. Es sind die Betriebsarten → Jog, → MDA, → Automatik definiert.

Betriebsartengruppe

Technologisch zusammengehörige Achsen und Spindeln können zu einer Betriebsartengruppe (BAG) zusammengefasst werden. Achsen/Spindeln einer BAG können von einem oder mehreren → Kanälen gesteuert werden. Den Kanälen der BAG ist immer die gleiche → Betriebsart zugeordnet.

Bezeichner

Die Wörter nach DIN 66025 werden durch Bezeichner (Namen) für Variable (Rechenvariable, Systemvariable, Anwendervariable), für Unterprogramme, für Schlüsselwörter und Wörter mit mehreren Adressbuchstaben ergänzt. Diese Ergänzungen kommen in der Bedeutung den Wörtern beim Satzaufbau gleich. Bezeichner müssen eindeutig sein. Derselbe Bezeichner darf nicht für verschiedene Objekte verwendet werden.

Booten

Laden des Systemprogramms nach Power On.

C-Achse

Achse, um die eine gesteuerte Drehbewegung und Positionierung mit der Werkstückspindel erfolgt.

CNC

Siehe → NC

Computerized Numerical Control: umfasst die Komponenten → NCK, → PLC, HMI, → COM.

CNC

Siehe → NC

Computerized Numerical Control: umfasst die Komponenten → NCK, → PLC, HMI, → COM.

COM

Komponente der NC-Steuerung zur Durchführung und Koordination von Kommunikation.

CPU

Central Processing Unit, siehe → Speicherprogrammierbare Steuerung

C-Spline

Der C-Spline ist der bekannteste und am meisten verwendete Spline. Die Übergänge an den Stützpunkten sind tangential- und krümmungstetig. Es werden Polynome 3. Grades verwendet.

Datenbaustein

1. Dateneinheit der → PLC, auf die → HIGHSTEP-Programme zugreifen können.
2. Dateneinheit der → NC: Datenbausteine enthalten Datendefinitionen für globale Anwenderdaten. Die Daten können bei der Definition direkt initialisiert werden.

Datenübertragungsprogramm PCIN

PCIN ist ein Hilfsprogramm zum Senden und Empfangen von CNC-Anwenderdaten über die serielle Schnittstelle, wie z. B. Teileprogramme, Werkzeugkorrekturen etc. Das PCIN-Programm ist unter MS-DOS auf Standard-Industrie-PCs lauffähig.

Datenwort

Zwei Byte große Dateneinheit innerhalb eines → Datenbausteins.

Diagnose

1. Bedienbereich der Steuerung
2. Die Steuerung besitzt sowohl ein Selbstdiagnose-Programm als auch Testhilfen für den Service: Status-, Alarm- und Serviceanzeigen

DRF

Differential Resolver Function: NC-Funktion, die in Verbindung mit einem elektronischen Handrad eine inkrementale Nullpunktverschiebung im Automatik-Betrieb erzeugt.

Editor

Der Editor ermöglicht das Erstellen, Ändern, Ergänzen, Zusammenschieben und Einfügen von Programmen/Texten/Programmsätzen.

Eilgang

Schnellste Verfahrgeschwindigkeit einer Achse. Sie wird z. B. verwendet, wenn das Werkzeug aus einer Ruhestellung an die → Werkstückkontur herangefahren oder von der Werkstückkontur zurückgezogen wird. Die Eilganggeschwindigkeit wird maschinenspezifisch über Maschinendatum eingestellt.

Externe Nullpunktverschiebung

Von der → PLC vorgegebene Nullpunktverschiebung.

Fertigteilkontur

Kontur des fertig bearbeiteten Werkstücks. Siehe → Rohteil.

Festpunkt-Anfahren

Werkzeugmaschinen können feste Punkte wie Werkzeugwechsellpunkt, Beladepunkt, Palettenwechsellpunkt etc. definiert anfahren. Die Koordinaten dieser Punkte sind in der Steuerung hinterlegt. Die Steuerung verfährt die betroffenen Achsen, wenn möglich, im → Eilgang.

Frame

Ein Frame stellt eine Rechenvorschrift dar, die ein kartesisches Koordinatensystem in ein anderes kartesisches Koordinatensystem überführt. Ein Frame enthält die Komponenten → Nullpunktverschiebung, → Rotation, → Skalierung, → Spiegelung.

Führungsachse

Die Führungsachse ist die → Gantry-Achse, die aus Sicht des Bedieners und des Programmierers vorhanden und damit entsprechend wie eine normale NC-Achse beeinflussbar ist.

Genauhalt

Bei programmierter Genauhalt-Anweisung wird die in einem Satz angegebene Position genau und ggf. sehr langsam angefahren. Zur Reduktion der Annäherungszeit werden für Eilgang und Vorschub → Genauhaltsgrenzen definiert.

Genauhaltgrenze

Erreichen alle Bahnachsen ihre Genauhaltgrenze, so verhält sich die Steuerung als habe sie einen Zielpunkt exakt erreicht. Es erfolgt Satzweitschaltung des → Teileprogramms.

Geometrie

Beschreibung eines → Werkstücks im → Werkstückkoordinatensystem.

Geometrieachse

Die Geometrieachsen bilden das 2- bzw. 3-dimensionale → Werkstückkoordinatensystem in dem in → Teileprogrammen die Geometrie des Werkstücks programmiert wird.

Geradeninterpolation

Das Werkzeug wird auf einer Geraden zum Zielpunkt verfahren und dabei das Werkstück bearbeitet.

Geschwindigkeitsführung

Um bei Verfahrbewegungen um sehr kleine Beträge je Satz eine akzeptable Verfahrgeschwindigkeit erreichen zu können, kann vorausschauende Auswertung über mehrere Sätze (→ Look Ahead) eingestellt werden.

Gewindebohren ohne Ausgleichsfutter

Mit dieser Funktion können Gewinde ohne Ausgleichsfutter gebohrt werden. Durch das interpolierende Verfahren der Spindel als Rundachse und der Bohrachse werden Gewinde exakt auf Endbohrtiefe geschnitten, z. B. Sacklochgewinde (Voraussetzung: Achsbetrieb der Spindel).

Gleichlaufachse

Die Gleichlaufachse ist die → Gantry-Achse, deren Sollposition stets von der Verfahrbewegung der → Führungsachse abgeleitet und damit synchron verfahren wird. Aus Sicht des Bedieners und des Programmierers ist die Gleichlaufachse "nicht vorhanden".

Grenzdrehzahl

Maximale/minimale (Spindel-)Drehzahl: Durch Vorgaben von Maschinendaten, der → PLC oder → Settingdaten kann die maximale Drehzahl einer Spindel begrenzt sein.

Hauptprogramm

Die Bezeichnung Hauptprogramm stammt noch aus der Zeit, als Teileprogramm fest in Haupt- und → Unterprogramme unterteilt waren. Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Prinzipiell kann jedes Teileprogramm im Kanal angewählt und gestartet werden. Es läuft dann in der → Programmebene 0 (Hauptprogramm-Ebene) ab. Im Hauptprogramm können weitere Teileprogramme oder → Zyklen als Unterprogramme aufgerufen werden

Hauptsatz

Durch ":" eingeleiteter Satz, der alle Angaben enthält, um den Arbeitsablauf in einem → Teileprogramm starten zu können.

HIGHSTEP

Zusammenfassung der Programmiermöglichkeiten für die → PLC des Systems AS300/AS400.

Hilfsfunktionen

Mit Hilfsfunktionen können in → Teileprogrammen → Parameter an die → PLC übergeben werden, die dort vom Maschinenhersteller definierte Reaktionen auslösen.

Hochsprache CNC

Die Hochsprache dient zum Schreiben von NC-Programmen, → Synchronaktionen und → Zyklen. Sie bietet: Kontrollstrukturen, → Anwenderdefinierte Variable, → Systemvariable, → Makrotechnik.

HW-Konfig

SIMATIC S7-Tool zum Konfigurieren und Parametrieren von Hardware-Komponenten innerhalb eines S7-Projekts.

Interpolator

Logische Einheit des → NCK, die nach Angaben von Zielpositionen im Teileprogramm Zwischenwerte für die in den einzelnen Achsen zu fahrenden Bewegungen bestimmt.

Interpolatorische Kompensation

Über interpolatorische Kompensationen wie → Spindelsteigungsfehler-, Durchhang-, Winkligkeits- und Temperaturkompensation werden mechanische Fehler der Maschine kompensiert.

Interruptroutine

Interruptroutinen sind spezielle → Unterprogramme, die durch Ereignisse (externe Signale) vom Bearbeitungsprozess gestartet werden können. Ein in Abarbeitung befindlicher Teileprogrammabsatz wird abgebrochen, die Unterbrechungsposition der Achsen wird automatisch gespeichert.

JOG

Betriebsart der Steuerung (Einrichtebetrieb): In der Betriebsart JOG kann die Maschine eingerichtet werden. Einzelne Achsen und Spindeln können über die Richtungstasten im Tippbetrieb verfahren werden. Weitere Funktionen in der Betriebsart JOG sind das → Referenzpunktfahren, → Repos sowie → Preset (Istwert setzen).

Kanal

Ein Kanal ist dadurch gekennzeichnet, dass er unabhängig von anderen Kanälen ein → Teileprogramm abarbeiten kann. Ein Kanal steuert exklusiv die ihm zugeordneten Achsen und Spindeln. Teileprogrammabläufe verschiedener Kanäle können durch → Synchronisation koordiniert werden.

Kettenmaß

Auch Inkrementmaß: Angabe eines Bewegungsziels einer Achse durch eine zu verfahrenende Wegstrecke und Richtung bezogen auf einen bereits erreichten Punkt. Siehe → Absolutmaß.

Kompensationsachse

Achse, deren Soll- oder Istwert durch den Kompensationswert modifiziert wird.

Kompensationstabelle

Tabelle von Stützpunkten. Sie liefert für ausgewählte Positionen der Basisachse die Kompensationswerte der Kompensationsachse.

Kompensationswert

Differenz zwischen der durch den Messgeber gemessenen Achsposition und der gewünschten, programmierten Achsposition.

Kontur

Umriss des → Werkstücks

Konturüberwachung

Als Maß für die Konturtreue wird der Schleppfehler innerhalb eines definierbaren Toleranzbandes überwacht. Ein unzulässig hoher Schleppfehler kann sich z. B. durch

Überlastung des Antriebs ergeben. In diesem Fall kommt es zu einem Alarm und die Achsen werden stillgesetzt.

Koordinatensystem

Siehe → Maschinenkoordinatensystem, → Werkstückkoordinatensystem

Korrekturspeicher

Datenbereich in der Steuerung, in dem Werkzeugkorrekturdaten hinterlegt sind.

Kreisinterpolation

Das → Werkzeug soll zwischen festgelegten Punkten der Kontur mit einem gegebenen Vorschub auf einem Kreis fahren und dabei das Werkstück bearbeiten.

Krümmung

Die Krümmung k einer Kontur ist das Inverse des Radius r des anschmiegenden Kreises in einem Konturpunkt ($k = 1/r$).

KÜ

Übersetzungsverhältnis

KV

Kreisverstärkungsfaktor, regelungstechnische Größe eines Regelkreises

Ladespeicher

Der Ladespeicher ist bei der CPU 314 der → SPS gleich dem → Arbeitsspeicher.

Linearachse

Die Linearachse ist eine Achse, welche im Gegensatz zur Rundachse eine Gerade beschreibt.

Look Ahead

Mit der Funktion **Look Ahead** wird durch das "Vorausschauen" über eine parametrierbare Anzahl von Verfahrssätzen ein Optimum an Bearbeitungsgeschwindigkeit erzielt.

Losekompensation

Ausgleich einer mechanischen Maschinenlose, z. B. Umkehrlose bei Kugelrollspindeln. Für jede Achse kann die Losekompensation getrennt eingegeben werden.

Makrotechnik

Zusammenfassung einer Menge von Anweisungen unter einem Bezeichner. Der Bezeichner repräsentiert im Programm die Menge der zusammengefassten Anweisungen.

Maschinenachsen

In der Werkzeugmaschine physikalisch existierende Achsen.

Maschinenfestpunkt

Durch die Werkzeugmaschine eindeutig definierter Punkt, z. B. Maschinen-Referenzpunkt.

Maschinenkoordinatensystem

Koordinatensystem, das auf die Achsen der Werkzeugmaschine bezogen ist.

Maschinennullpunkt

Fester Punkt der Werkzeugmaschine, auf den sich alle (abgeleiteten) Messsysteme zurückführen lassen.

Maschinensteuertafel

Bedientafel der Werkzeugmaschine mit den Bedienelementen Tasten, Drehschalter usw. und einfachen Anzeigeelementen wie LEDs. Sie dient der unmittelbaren Beeinflussung der Werkzeugmaschine über die PLC.

Maßangabe metrisch und inch

Im Bearbeitungsprogramm können Positions- und Steigungswerte in inch programmiert werden. Unabhängig von der programmierbaren Maßangabe (*G70/G71*) wird die Steuerung auf ein Grundsystem eingestellt.

Masse

Als Masse gilt die Gesamtheit aller untereinander verbundenen inaktiven Teile eines Betriebsmittels, die auch im Fehlerfall keine gefährliche Berührungsspannung annehmen können.

MDA

Betriebsart der Steuerung: Manual Data Automatic. In der Betriebsart MDA können einzelne Programmsätze oder Satzfolgen ohne Bezug auf ein Haupt- oder Unterprogramm eingegeben und anschließend über die Taste NC-Start sofort ausgeführt werden.

Meldungen

Alle im Teileprogramm programmierten Meldungen und vom System erkannte → Alarme werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

Metrisches Messsystem

Genormtes System von Einheiten: für Längen z. B. mm (Millimeter), m (Meter).

NC

Numerical Control Komponente der → CNC, die → Teileprogramme abarbeitet und die Bewegungsvorgänge der Werkzeugmaschine koordiniert.

Nebensatz

Durch "N" eingeleiteter Satz mit Informationen für einen Arbeitsschritt, z. B. eine Positionsangabe.

Netz

Ein Netz ist die Verbindung von mehreren S7-300 und weiteren Endgeräten, z. B. einem PG, über → Verbindungskabel. Über das Netz erfolgt ein Datenaustausch zwischen den angeschlossenen Geräten.

NRK

Numeric Robotic Kernel (Betriebssystem des → NCK)

Nullpunktverschiebung

Vorgabe eines neuen Bezugspunkts für ein Koordinatensystem durch Bezug auf einen bestehenden Nullpunkt und ein → Frame.

1. Einstellbar
Es steht eine projektierbare Anzahl von einstellbaren Nullpunktverschiebungen für jede CNC-Achse zur Verfügung. Die über G-Befehle anwählbaren Verschiebungen sind alternativ wirksam.
2. Extern
Zusätzlich zu allen Verschiebungen, die die Lage des Werkstücknullpunkts festlegen, kann eine externe Nullpunktverschiebung durch Handrad (DRF-Verschiebung) oder von der PLC überlagert werden.
3. Programmierbar
Mit der Anweisung `TRANS` sind für alle Bahn- und Positionierachsen Nullpunktverschiebungen programmierbar.

NURBS

Die steuerungsinterne Bewegungsführung und Bahninterpolation wird auf Basis von NURBS (**N**on **U**niform **R**ational **B**-**S**plines) durchgeführt. Damit steht steuerungsintern für alle Interpolationen ein einheitliches Verfahren zur Verfügung.

OEM

Für Maschinenhersteller, die ihre eigene Bedienoberfläche erstellen oder technologiespezifische Funktionen in die Steuerung einbringen wollen, sind Freiräume für individuelle Lösungen (OEM-Applikationen) vorgesehen.

Orientierter Spindelhalt

Halt der Werkstückspindel in vorgegebener Winkellage, z. B. um an bestimmter Stelle eine Zusatzbearbeitung vorzunehmen.

Override

Manuelle bzw. programmierbare Eingriffsmöglichkeit, die es dem Bediener gestattet, programmierte Vorschübe oder Drehzahlen zu überlagern, um sie einem bestimmten Werkstück oder Werkstoff anzupassen.

Peripheriebaugruppe

Peripheriebaugruppen stellen die Verbindung zwischen CPU und Prozess her.

Peripheriebaugruppen sind:

- → Digital-Ein-/Ausgabebaugruppen
- → Analog-Ein-/Ausgabebaugruppen
- → Simulatorbaugruppen

PLC

Programmable Logic Control: → Speicherprogrammierbare Steuerung. Komponente der → NC: Anpass-Steuerung zur Bearbeitung der Kontroll-Logik der Werkzeugmaschine.

PLC-Programmierung

Die PLC wird mit der Software **STEP 7** programmiert. Die Programmiersoftware STEP 7 basiert auf dem Standardbetriebssystem **WINDOWS** und enthält die Funktionen der STEP 5 - Programmierung mit innovativen Weiterentwicklungen.

PLC-Programmspeicher

SINUMERIK 840D sl: Im PLC-Anwenderspeicher werden das PLC-Anwenderprogramm und die Anwenderdaten gemeinsam mit dem PLC-Grundprogramm abgelegt.

Polarkoordinaten

Koordinatensystem, das die Lage eines Punktes in einer Ebene durch seinen Abstand vom Nullpunkt und den Winkel festlegt, den der Radiusvektor mit einer festgelegten Achse bildet.

Polynom-Interpolation

Mit der Polynom-Interpolation können die unterschiedlichsten Kurvenverläufe erzeugt werden, wie **Gerade-, Parabel-, Potenzfunktionen** (SINUMERIK 840D sl).

Positionierachse

Achse, die eine Hilfsbewegung an einer Werkzeugmaschine ausführt. (z. B. Werkzeugmagazin, Palettentransport). Positionierachsen sind Achsen, die nicht mit den → Bahnachsen interpolieren.

Programmbaustein

Programmbausteine enthalten die Haupt- und Unterprogramme der → Teileprogramme.

Programmebene

Ein im Kanal gestartetes Teileprogramm läuft als → Hauptprogramm auf Programmebene 0 (Hauptprogramm-Ebene). Jedes im Hauptprogramm aufgerufene Teileprogramm läuft als → Unterprogramm auf einer eigenen Programmebene 1 ... n.

Programmierbare Arbeitsfeldbegrenzung

Begrenzung des Bewegungsraumes des Werkzeugs auf einen durch programmierte Begrenzungen definierten Raum.

Programmierbare Frames

Mit programmierbaren → Frames können dynamisch im Zuge der Teileprogramm-Abarbeitung neue Koordinatensystem-Ausgangspunkte definiert werden. Es wird unterschieden nach absoluter Festlegung anhand eines neuen Frames und additiver Festlegung unter Bezug auf einen bestehenden Ausgangspunkt.

Programmierschlüssel

Zeichen und Zeichenfolgen, die in der Programmiersprache für → Teileprogramme eine festgelegte Bedeutung haben.

Pufferbatterie

Die Pufferbatterie gewährleistet, dass das → Anwenderprogramm in der → CPU netzausfallsicher hinterlegt ist und festgelegte Datenbereiche und Merker, Zeiten und Zähler remanent gehalten werden.

Quadrantenfehlerkompensation

Konturfehler an Quadrantenübergängen, die durch wechselnde Reibverhältnisse an Führungsbahnen entstehen, sind mit der Quadrantenfehlerkompensation weitgehend eliminierbar. Die Parametrierung der Quadrantenfehlerkompensation erfolgt durch einen Kreisformtest.

Referenzpunkt

Punkt der Werkzeugmaschine, auf den sich das Messsystem der → Maschinenachsen bezieht.

Rohteil

Teil, mit dem die Bearbeitung eines Werkstücks begonnen wird.

Rotation

Komponente eines → Frames, die eine Drehung des Koordinatensystems um einen bestimmten Winkel definiert.

R-Parameter

Rechenparameter, kann vom Programmierer des → Teileprogramms für beliebige Zwecke im Programm gesetzt oder abgefragt werden.

Rundachse

Rundachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine vorgegebene Winkellage.

Rundungsachse

Rundungsachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine einem Teilungsraster entsprechende Winkellage. Beim Erreichen eines Rasters ist die Rundungsachse "in Position".

Satzsuchlauf

Zum Austesten von Teileprogrammen oder nach einem Abbruch der Bearbeitung kann über die Funktion "Satzsuchlauf" eine beliebige Stelle im Teileprogramm angewählt werden, an der die Bearbeitung gestartet oder fortgesetzt werden soll.

Schlüsselschalter

Der Schlüsselschalter auf der → Maschinensteuertafel besitzt 4 Stellungen, die vom Betriebssystem der Steuerung mit Funktionen belegt sind. Zum Schlüsselschalter gehören drei verschiedenfarbige Schlüssel, die in den angegebenen Stellungen abgezogen werden können.

Schlüsselwörter

Wörter mit festgelegter Schreibweise, die in der Programmiersprache für → Teileprogramme eine definierte Bedeutung haben.

Schneidenradiuskorrektur

Bei der Programmierung einer Kontur wird von einem spitzen Werkzeug ausgegangen. Da dies in der Praxis nicht realisierbar ist, wird der Krümmungsradius des eingesetzten Werkzeugs der Steuerung angegeben und von dieser berücksichtigt. Dabei wird der Krümmungsmittelpunkt um den Krümmungsradius verschoben äquidistant um die Kontur geführt.

Schnellabheben von der Kontur

Beim Eintreffen eines Interrupts kann über das CNC-Bearbeitungsprogramm eine Bewegung eingeleitet werden, die ein schnelles Abheben des Werkzeugs von der gerade bearbeiteten Werkstückkontur ermöglicht. Zusätzlich kann der Rückzugwinkel und der Betrag des Weges parametrisiert werden. Nach dem Schnellabheben kann zusätzlich eine Interruptroutine ausgeführt werden.

Schnelle digitale Ein-/Ausgänge

Über die digitalen Eingänge können z. B. schnelle CNC-Programmroutinen (Interruptroutinen) gestartet werden. Über die digitalen CNC-Ausgänge können schnelle, programmgesteuerte Schalfunktionen ausgelöst werden.

Schrägenbearbeitung

Bohr- und Fräsbearbeitungen an Werkstückflächen, die nicht in den Koordinatenebenen der Maschine liegen, können mit Unterstützung der Funktion "Schrägenbearbeitung" komfortabel ausgeführt werden.

Schraubenlinien-Interpolation

Die Schraubenlinien-Interpolation eignet sich besonders zum einfachen Herstellen von Innen- oder Außengewinden mit Formfräsern und zum Fräsen von Schmiernuten.

Dabei setzt sich die Schraubenlinie aus zwei Bewegungen zusammen:

- Kreisbewegung in einer Ebene
- Linearbewegung senkrecht zu dieser Ebene

Schrittmaß

Verfahrweglängenangabe über Inkrementanzahl (Schrittmaß). Inkrementanzahl kann als → Settingdatum hinterlegt sein bzw. durch entsprechend beschriftete Tasten 10, 100, 1000, 10000 gewählt werden.

Schutzraum

Dreidimensionaler Raum innerhalb des → Arbeitsraumes, in den die Werkzeugspitze nicht hineinreichen darf.

Settingdaten

Daten, die Eigenschaften der Werkzeugmaschine auf durch die Systemsoftware definierte Weise der NC-Steuerung mitteilen.

Sicherheitsfunktionen

Die Steuerung enthält ständig aktive Überwachungen, die Störungen in der → CNC, der Anpass-Steuerung (→ PLC) und der Maschine so frühzeitig erkennen, dass Schäden an Werkstück, Werkzeug oder Maschine weitgehend ausgeschlossen werden. Im Störfall wird der Bearbeitungsablauf unterbrochen und die Antriebe werden stillgesetzt, die Störungsursache gespeichert und als Alarm angezeigt. Gleichzeitig wird der PLC mitgeteilt, dass ein CNC-Alarm ansteht.

Skalierung

Komponente eines → Frames, die achsspezifische Maßstabsveränderungen bewirkt.

Softkey

Taste, deren Beschriftung durch ein Feld im Bildschirm repräsentiert wird, das sich dynamisch der aktuellen Bediensituation anpasst. Die frei belegbaren Funktionstasten (Softkeys) werden softwaremäßig definierten Funktionen zugeordnet.

Software-Endschalter

Software-Endschalter begrenzen den Verfahrbereich einer Achse und verhindern ein Auffahren des Schlittens auf die Hardware-Endschalter. Je Achse sind 2 Wertepaare vorgebar, die getrennt über die → PLC aktiviert werden können.

Speicherprogrammierbare Steuerung

Speicherprogrammierbare Steuerungen (SPS) sind elektronische Steuerungen, deren Funktion als Programm im Steuerungsgerät gespeichert ist. Aufbau und Verdrahtung des Gerätes hängen also nicht von der Funktion der Steuerung ab. Die speicherprogrammierbare Steuerung hat die Struktur eines Rechners; sie besteht aus CPU (Zentralbaugruppe) mit Speicher, Ein-/Ausgabebaugruppen und internem Bus-System. Die Peripherie und die Programmiersprache sind auf die Belange der Steuerungstechnik ausgerichtet.

Spiegelung

Bei Spiegelung werden die Vorzeichen der Koordinatenwerte einer Kontur bezüglich einer Achse vertauscht. Es kann bezüglich mehrerer Achsen zugleich gespiegelt werden.

Spindelsteigungsfehler-Kompensation

Ausgleich mechanischer Ungenauigkeiten einer am Vorschub beteiligten Kugelrollspindel durch die Steuerung anhand von hinterlegten Messwerten der Abweichungen.

Spline-Interpolation

Mit der Spline-Interpolation kann die Steuerung aus nur wenigen vorgegebenen Stützpunkten einer Sollkontur einen glatten Kurvenverlauf erzeugen.

Standardzyklen

Für häufig wiederkehrende Bearbeitungsaufgaben stehen Standardzyklen zur Verfügung:

- Für die Technologie Bohren/Fräsen
- Für die Technologie Drehen

Im Bedienbereich "Programm" werden unter dem Menü "Zyklenunterstützung" die zur Verfügung stehenden Zyklen aufgelistet. Nach Anwahl des gewünschten Bearbeitungszyklus werden die notwendigen Parameter für die Wertzuweisung im Klartext angezeigt.

Synchronachsen

Synchronachsen benötigen für ihren Weg die gleiche Zeit wie die Geometrieachsen für ihren Bahnweg.

Synchronaktionen

1. Hilfsfunktionsausgabe
Während der Werkstückbearbeitung können aus dem CNC-Programm heraus technologische Funktionen (→ Hilfsfunktionen) an die PLC ausgegeben werden. Über diese Hilfsfunktionen werden beispielsweise Zusatzeinrichtungen der Werkzeugmaschine gesteuert, wie Pinole, Greifer, Spannfutter etc.
2. Schnelle Hilfsfunktionsausgabe
Für zeitkritische Schaltfunktionen können die Quittierungszeiten für die → Hilfsfunktionen minimiert und unnötige Haltepunkte im Bearbeitungsprozess vermieden werden.

Synchronisation

Anweisungen in → Teileprogrammen zur Koordination der Abläufe in verschiedenen → Kanälen an bestimmten Bearbeitungsstellen.

Systemspeicher

Der Systemspeicher ist ein Speicher in der CPU, in der folgende Daten abgelegt werden:

- Daten, die das Betriebssystem benötigt
- Die Operanden Zeiten, Zähler, Merker

Systemvariable

Ohne Zutun des Programmierers eines → Teileprogramms existierende Variable. Sie ist definiert durch einen Datentyp und dem Variablennamen, der durch das Zeichen \$ eingeleitet wird. Siehe → Anwenderdefinierte Variable.

Teileprogramm

Folge von Anweisungen an die NC-Steuerung, die insgesamt die Erzeugung eines bestimmten → Werkstücks bewirken. Ebenso Vornahme einer bestimmten Bearbeitung an einem gegebenen → Rohteil.

Teileprogrammsatz

Teil eines → Teileprogramms, durch Line Feed abgegrenzt. Es werden → Hauptsätze und → Nebensätze unterschieden.

Teileprogrammverwaltung

Die Teileprogrammverwaltung kann nach → Werkstücken organisiert werden. Die Größe des Anwenderspeichers bestimmt die Anzahl der zu verwaltenden Programme und Daten. Jede Datei (Programme und Daten) kann mit einem Namen von maximal 24 alphanumerischen Zeichen versehen werden.

Text-Editor

Siehe → Editor

TOA-Bereich

Der TOA-Bereich umfasst alle Werkzeug- und Magazindaten. Standardmäßig fällt der Bereich bzgl. der Reichweite der Daten mit dem Bereich → Kanal zusammen. Über Maschinendaten kann jedoch festgelegt werden, dass sich mehrere Kanäle eine → TOA-Einheit teilen, so dass diesen Kanälen dann gemeinsame WZV-Daten zur Verfügung stehen.

TOA-Einheit

Jeder → TOA-Bereich kann mehrere TOA-Einheiten enthalten. Die Anzahl der möglichen TOA-Einheiten wird über die maximale Anzahl aktiver → Kanäle begrenzt. Eine TOA-Einheit umfasst genau einen WZ-Daten-Baustein und einen Magazindaten-Baustein. Zusätzlich kann noch ein WZ-Trägerdaten-Baustein enthalten sein (optional).

Transformation

Additive oder absolute Nullpunktverschiebung einer Achse.

Unterprogramm

Die Bezeichnung Unterprogramm stammt noch aus der Zeit, als Teileprogramm fest in → Haupt- und Unterprogramme unterteilt waren. Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Prinzipiell kann jedes Teileprogramm oder jeder → Zyklus innerhalb eines anderen Teileprogramms als Unterprogramm aufgerufen werden. Es läuft dann in der nächsten → Programmebene (x+1) (Unterprogrammebene (x+1)) ab.

Urlöschen

Beim Urlöschen werden folgende Speicher der → CPU gelöscht:

- → Arbeitsspeicher
- Schreib-/Lesebereich des → Ladespeichers
- → Systemspeicher
- → Backup-Speicher

V.24

Serielle Schnittstelle für die Dateneingabe/-ausgabe. Über diese Schnittstelle können Bearbeitungsprogramme sowie Hersteller- und Anwenderdaten geladen und gesichert werden.

Variablendefinition

Eine Variablendefinition umfasst die Festlegung eines Datentyps und eines Variablennamens. Mit dem Variablennamen kann der Wert der Variablen angesprochen werden.

Verfahrbereich

Der maximal zulässige Verfahrbereich bei Linearachsen beträgt ± 9 Dekaden. Der absolute Wert ist abhängig von der gewählten Eingabe- und Lageregelfeinheit und dem Einheitensystem (inch oder metrisch).

Vorkoinzidenz

Satzwechsel bereits, wenn Bahnweg um ein vorgegebenes Delta der Endposition nahe gekommen ist.

Vorschub-Override

Der programmierten Geschwindigkeit wird die aktuelle Geschwindigkeitseinstellung über → Maschinensteuertafel oder von der → PLC überlagert (0-200%). Die Vorschubgeschwindigkeit kann zusätzlich im Bearbeitungsprogramm durch einen programmierbaren Prozentfaktor (1-200%) korrigiert werden.

Vorsteuerung, dynamisch

Ungenauigkeiten der → Kontur, bedingt durch Schleppfehler, lassen sich durch die dynamische, beschleunigungsabhängige Vorsteuerung nahezu eliminieren. Dadurch ergibt sich auch bei hohen → Bahngeschwindigkeiten eine hervorragende Bearbeitungsgenauigkeit. Die Vorsteuerung kann achsspezifisch über das → Teileprogramm an- und abgewählt werden.

Werkstück

Von der Werkzeugmaschine zu erstellendes/zu bearbeitendes Teil.

Werkstückkontur

Sollkontur des zu erstellenden/bearbeitenden → Werkstücks.

Werkstückkoordinatensystem

Das Werkstückkoordinatensystem hat seinen Ausgangspunkt im → Werkstücknullpunkt. Bei Programmierung im Werkstückkoordinatensystem beziehen sich Maße und Richtungen auf dieses System.

Werkstücknullpunkt

Der Werkstücknullpunkt bildet den Ausgangspunkt für das → Werkstückkoordinatensystem. Er ist durch Abstände zum → Maschinennullpunkt definiert.

Werkzeug

An der Werkzeugmaschine wirksames Teil, das die Bearbeitung bewirkt (z. B. Drehmeißel, Fräser, Bohrer, LASER-Strahl ...).

Werkzeugkorrektur

Berücksichtigung der Werkzeug-Abmessungen bei der Berechnung der Bahn.

Werkzeugradiuskorrektur

Um eine gewünschte → Werkstückkontur direkt programmieren zu können, muss die Steuerung unter Berücksichtigung des Radius des eingesetzten Werkzeugs eine äquidistante Bahn zur programmierten Kontur verfahren (G41/G42).

WinSCP

WinSCP ist ein frei verfügbares Open Source-Programm für Windows zum Transferieren von Dateien.

Zeitreziproker Vorschub

Anstelle der Vorschubgeschwindigkeit kann für die Achsbewegung auch die Zeit programmiert werden, die der Bahnweg eines Satzes benötigen soll (G93).

Zoll-Maßsystem

Maßsystem, das Entfernungen in "inch" und Bruchteilen davon definiert.

Zwischensätze

Verfahrbewegungen mit angewählter → Werkzeugkorrektur (G41/G42) dürfen durch eine begrenzte Anzahl Zwischensätze (Sätze ohne Achsbewegungen in der Korrektorebene) unterbrochen werden, wobei die Werkzeugkorrektur noch korrekt verrechnet werden kann. Die zulässige Anzahl Zwischensätze, die die Steuerung vorausliest, ist über Systemparameter einstellbar.

Zyklen

Geschützte Unterprogramme zur Ausführung von wiederholt auftretenden Bearbeitungsvorgängen am → Werkstück.

Index

\$

\$AA_ATOL, 473
\$AA_COUP_ACT
 bei axialer Leitwertkopplung, 502
 beim Mitschleppen, 480
\$AA_ESR_ENABLE, 587
\$AA_LEAD_SP, 502
\$AA_LEAD_SV, 502
\$AC_ACT_PROG_NET_TIME, 576
\$AC_ACTUAL_PARTS, 579
\$AC_AXCTSWA, 566
\$AC_AXCTSWE, 567
\$AC_CTOL, 473
\$AC_CUT_INV, 431
\$AC_CUTMOD, 431
\$AC_CUTMOD_ANG, 431
\$AC_CUTTING_TIME, 576
\$AC_CYCLE_TIME, 576
\$AC_DELAYFST, 455
\$AC_ESR_TRIGGER, 587
\$AC_OLD_PROG_NET_TIME, 576
\$AC_OLD_PROG_NET_TIME_COUNT, 576
\$AC_OPERATING_TIME, 576
\$AC_OTOL, 473
\$AC_PROG_NET_TIME_TRIGGER, 577
\$AC_REPOS_PATH_MODE, 465
\$AC_REQUIRED_PARTS, 579
\$AC_SMAXVELO, 469
\$AC_SMAXVELO_INFO, 469
\$AC_SPECIAL_PARTS, 579
\$AC_STOLF, 475
\$AC_TOTAL_PARTS, 579
\$AN_AXCTAS, 567
\$AN_AXCTSWA, 566
\$AN_ESR_TRIGGER, 587
\$AN_LANGUAGE_ON_HMI, 796
\$AN_POWERON_TIME, 575
\$AN_SETUP_TIME, 575
\$P_ACTBFRAME, 296
\$P_AD, 431
\$P_BFRAME, 296
\$P_CHBFRAME, 295
\$P_CHBFRMASK, 296
\$P_CTOL, 474
\$P_CUT_INV, 431
\$P_CUTMOD, 431
\$P_CUTMOD_ANG, 431

\$P_DELAYFST, 455
\$P_IFRAME, 297
\$P_IS_EES_PATH, 216
\$P_NCBFRAME, 295
\$P_NCBFRMASK, 296
\$P_OTOL, 474
\$P_PATH, 215
\$P_PFRAME, 297
\$P_PROG, 215
\$P_PROGPATH, 215
\$P_SIM, 267
\$P_STACK, 215
\$P_STOLF, 475
\$P_SUBPAR, 154
\$PA_ATOL, 474
\$SA_LEAD_TYPE, 502
\$SC_CONTPREC, 450
\$SC_MINFEED, 451
\$SC_PA_ACTIV_IMMED, 227
\$SN_PA_ACTIV_IMMED, 227
\$TC_CARR1...14, 416
\$TC_CARR18...23, 416
\$TC_CARR18[m], 420
\$TC_DP1 ... 25, 375
\$TC_ECPxy, 379
\$TC_SCPxy, 379

*

* (Rechenfunktion), 71

/

/ (Rechenfunktion), 71

+

+ (Rechenfunktion), 71

<

< (Vergleichsoperator), 73

<< (Verkettungsoperator), 81

<= (Vergleichsoperator), 73

<> (Vergleichsoperator), 73

- =
- == (Vergleichsoperator), 73
- >
- > (Vergleichsoperator), 73
- >= (Vergleichsoperator), 73
- 0**
- 0-Zeichen, 79
- 3**
- 3D-Werkzeugkorrektur, 397
 - Bahnkrümmung, 399
 - Eintauchtiefe, 399
 - Korrektur auf der Bahn, 398
 - Schnittpunktverfahren, 401
- 3D-Werkzeugradiuskorrektur, 392
 - 3DSchnittpunkt der Äquidistanten, 401
 - Innenecken/Außenecken, 400
 - Stirnfräsen, 396
 - Übergangskreis, 401
 - Umfangsfräsen, 395
- A**
- Abrichterposition berechnen - CYCLE435
 - extern programmieren, 676
- ABS, 71
- Abspanen
 - unterstützende Funktionen, 595
- Abspanen - CYCLE951
 - extern programmieren, 706
- Abstich - CYCLE92
 - extern programmieren, 663
- ACCLIMA, 446
- Achs
 - tausch, 131
- Achsen
 - Mitschlepp-, 479
- ACOS, 71
- ACTBLOCNO, 166
- ACTFRAME, 275
- ADISPOSA, 269
- Adressierung, 209
- Alarme
 - im NC-Programm setzen, 584
- ALF
 - für Schnellabheben von der Kontur, 127
- AND, 73
- APR, 43
- APRB, 43
- APRP, 43
- APW, 43
- APWB, 43
- APWP, 43
- Arbeitsspeicher, 216
- Array, 50
- AS, 202
- ASIN, 71
- ASPLINE, 234
- A-Spline, 240
- Asynchrones Pendeln, 533
- ATAN2, 71
- ATOL, 471
- Aufrunden, 147
- Ausdrehen - CYCLE86
 - extern programmieren, 662
- Ausgabe
 - auf externes Gerät/Datei, 580
- Automatische Wegaufteilung, 550
- Automatischer Unterbrechungszeiger, 458
- AV, 510
- AX, 559
- AXCTSWE, 565
- AXCTSWEC, 565
- AXCTSWED, 565
- Axiale Leitwertkopplung, 498
- AXIS, 27
- AXNAME, 80
- AXSTRING, 559
- AXTOCHAN, 135
- AXTOSPI, 559
- B**
- B_AND, 73
- B_NOT, 73
- B_OR, 73
- B_XOR, 73
- Bahnfräsen - CYCLE72
 - extern programmieren, 639
- Bahnrelative Werkzeugorientierung, 335
- BAUTO, 234
- Bearbeitungszeit, 576
- Beliebige Positionen - CYCLE802
 - extern programmieren, 683

Beschleunigungsmodus, 444
Betriebsmodus
 beim Messen, 262
Bewegungsendekriterium
 programmierbar, 269
BFRAME, 275
BLOCK, 190
BLSYNC, 122
BNAT, 234
Bohren - CYCLE82
 extern programmieren, 652
Bohrgewinde fräsen - CYCLE78
 extern programmieren, 647
BOOL, 27
BOUND, 76
BRISK, 444
BRISKA, 444
BSPLINE, 234
B-Spline, 241
BTAN, 234

C

CAC, 233
CACN, 233
CACP, 233
CALL, 189
CALLPATH, 193
CASE, 99
case-insensitiv, 209
CDC, 233
CFINE, 284
CHAN, 27
CHANDATA, 216
CHAR, 27
CHKDNO, 412
CIC, 233
CLEARM, 114
CLRINT, 124
COARSE, 510
COARSEA, 269
COLLPAIR, 369
COMPCAD, 245
COMPCURV, 245
COMPLETE, 216
COMPOF, 245
COMPON, 245
COMPSURF, 245
CONTDCON, 601
CONTPRON, 595
COS, 71
COUPDEF, 510
COUPDEL, 510
COUPOF, 510
COUPOFS, 510
COUPON, 510
COUPONC, 510
COUPRES, 510
CPBC, 522
CPDEF, 521
CPDEL, 521
CPFMOF, 524
CPFMON, 524
CPFMSON, 523
CPFPOS, 522, 524
CPFRS, 521
CPLA, 521
CPLCTID, 521
CPLDEF, 521
CPLDEL, 521
CPLDEN, 521
CPLINSC, 526
CPLINTR, 526
CPLNUM, 521
CPLOF, 521
CPLON, 521
CPLOUTSC, 526
CPLOUTTR, 526
CPLPOS, 522
CPLSETVAL, 521
CPMALARM, 527
CPMBRAKE, 527
CPMPRT, 526
CPMRESET, 525
CPMSTART, 526
CPMVDI, 527
CPOF, 521
CPON, 521
CPRECOF, 450
CPRECON, 450
CPROT, 224
CPROTDEF, 221
CPSETTYPE, 527
CPSYNCOF, 526
CPSYNCOF2, 526
CPSYNCOV, 527
CPSYNFIP, 526
CPSYNFIP2, 527
CPSYNFIV, 527
CSPLINE, 234
C-Spline, 242
CTAB, 492
CTABDEF, 482
CTABDEL, 488

- CTABEND, 482
- CTABEXISTS, 488
- CTABFNO, 496
- CTABFPOL, 496
- CTABFSEG, 496
- CTABID, 490
- CTABINV, 492
- CTABISLOCK, 490
- CTABLOCK, 489
- CTABMEMTYP, 490
- CTABMPOL, 496
- CTABMSEG, 496
- CTABNO, 496
- CTABNOMEM, 496
- CTABPERIOD, 490
- CTABPOL, 496
- CTABPOLID, 496
- CTABSEG, 496
- CTABSEGID, 496
- CTABSEV, 492
- CTABSSV, 492
- CTABTEP, 492
- CTABTEV, 492
- CTABTMAX, 492
- CTABTMIN, 492
- CTABTSP, 492
- CTABTSV, 492
- CTABUNLOCK, 489
- CTOL, 471
- CTRANS, 284
- CUT3DC, 393, 399
- CUT3DCC, 402
- CUT3DCCD, 402
- CUT3DCD, 393, 399
- CUT3DF, 393
- CUT3DFF, 393
- CUT3DFS, 393
- CUTMOD, 427
- CYCLE4071
 - extern programmieren, 715
- CYCLE4072
 - extern programmieren, 716
- CYCLE4073
 - extern programmieren, 720
- CYCLE4074
 - extern programmieren, 721
- CYCLE4075
 - extern programmieren, 724
- CYCLE4077
 - extern programmieren, 727
- CYCLE4078
 - extern programmieren, 730
- CYCLE4079
 - extern programmieren, 732
- CYCLE435 - Abrichterposition berechnen
 - extern programmieren, 676
- CYCLE495 - Profilieren
 - extern programmieren, 676
- CYCLE60 - Gravur
 - extern programmieren, 628
- CYCLE61 - Planfräsen
 - extern programmieren, 631
- CYCLE62 - Konturaufruf
 - extern programmieren, 633
- CYCLE63 - Konturtasche fräsen
 - extern programmieren, 634
- CYCLE64 - Konturtasche vorbohren
 - extern programmieren, 636
- CYCLE70 - Gewindefräsen
 - extern programmieren, 637
- CYCLE72 - Bahnfräsen
 - extern programmieren, 639
- CYCLE76 - Rechteckzapfen
 - extern programmieren, 643
- CYCLE77 - Kreiszapfen
 - extern programmieren, 645
- CYCLE78 - Bohrgewinde fräsen
 - extern programmieren, 647
- CYCLE79 - Mehrkant
 - extern programmieren, 649
- CYCLE800 - Schwenken
 - extern programmieren, 678
- CYCLE801 - Positionsmuster Gitter/Rahmen
 - extern programmieren, 681
- CYCLE802 - Beliebige Positionen
 - extern programmieren, 683
- CYCLE81 - Zentrieren
 - extern programmieren, 651
- CYCLE82 - Bohren
 - extern programmieren, 652
- CYCLE83 - Tieflochbohren
 - extern programmieren, 655
- CYCLE830 - Tieflochbohren 2
 - extern programmieren, 686
- CYCLE832 - High Speed Settings
 - extern programmieren, 692
- CYCLE84 - Gewindebohren ohne Ausgleichsfutter
 - extern programmieren, 658
- CYCLE840 - Gewindebohren mit Ausgleichsfutter
 - extern programmieren, 695
- CYCLE85 - Reiben
 - extern programmieren, 661
- CYCLE86 - Ausdrehen
 - extern programmieren, 662

CYCLE899 - Offene Nut fräsen
 extern programmieren, 698
 CYCLE92 - Abstich
 extern programmieren, 663
 CYCLE930 - Einstich
 extern programmieren, 701
 CYCLE940 - Freistich
 extern programmieren, 704
 CYCLE95 - Konturabspanen
 extern programmieren, 665
 CYCLE951 - Abspanen
 extern programmieren, 706
 CYCLE952 - Stechen
 extern programmieren, 709
 CYCLE98 - Gewindegewinde
 extern programmieren, 667
 CYCLE99 - Gewindedrehen
 extern programmieren, 671

D

Datei
 -informationen, 145
 Dateiname, 213
 Datenklasse, 48
 DCI, 48
 DCM, 48
 DCU, 48
 DEF, 27
 DEFAULT, 99
 DEFINE ... AS, 202
 DELAYFSTOF, 454
 DELAYFSTON, 454
 DELDL, 380
 DELETE, 141
 DELOBJ, 365
 DIN Unterprogrammname, 213
 DISABLE, 124
 DISPLOF, 166
 DISPLON, 166
 DISPR, 458
 DIV, 71
 DL, 378
 D-Nummer
 frei vergeben, 412
 D-Nummern
 prüfen, 412
 umbenennen, 413
 DO, 531
 Drehachsen
 Abstandsvektoren, 416

Richtungsvektoren, 416
 Verdrehwinkel, 416
 Drehung
 des Orientierungsvektors, 333
 Drehzahlsynchronität, 510
 DRIVE, 444
 DRIVEA, 444
 DV, 510
 DYNFINISH, 448
 DYNNORM, 448
 DYNPOS, 448
 DYNROUGH, 448
 DYNSEMIFIN, 448

E

EAUTO, 234
 Eckenverzögerung an allen Ecken, 268
 Eckenverzögerung an Innenecken, 268
 EES, 207
 EES-Notation, 209
 EG
 Elektronisches Getriebe, 503
 EGDEF, 503
 EGDEL, 508
 EGOFC, 507
 EGOF, 507
 EGON, 504
 EGONSYN, 504
 EGONSYNE, 504
 Einrichtewert, 379
 Einstich - CYCLE930
 extern programmieren, 701
 Eintauchtiefe, 399
 Einzelachsbewegung, 554
 Einzelsatz
 -unterdrückung, 161
 Elektronisches Getriebe, 503
 ELSE, 108
 ENABLE, 124
 ENAT, 234
 ENDFOR, 110
 ENDIF, 108
 ENDLABEL, 101
 ENDLOOP, 110
 Endlosschleife, 110
 ENDWHILE, 112
 ESR, 586
 ESRR, 592
 ESRS, 591
 ETAN, 234
 Eulerwinkel, 317

EVERY, 531
 EXECSTRING, 69
 EXECTAB, 606
 EXECUTE, 609
 EXP, 71
 EXTCALL, 194, 198
 EXTCLOSE, 580
 EXTERN, 184
 Externe Nullpunktverschiebung, 285
 EXTOPEN, 580

F

F10, 221
 FALSE, 27
 FCTDEF, 388
 FCUB, 439
 Feld
 -definition, 50
 -element, 50
 Feldindex, 52
 FENDNORM, 268
 FFWOF, 449
 FFWON, 449
 FIFOCTRL, 452
 FILEDATE, 145
 FILEINFO, 145
 FILESIZE, 145
 FILESTAT, 145
 FILETIME, 145
 FINE, 510
 FINEA, 269
 FLIN, 439
 FNORM, 439
 Folgeachse
 bei axialer Leitwertkopplung, 498
 FOR, 110
 FPO, 439
 Frame
 aufrufen, 281
 -Kettung, 299
 FRAME, 27
 Framekomponente
 FI, 280
 MI, 280
 RT, 280
 SC, 280
 TR, 280
 Frames
 Frameketten, 283
 Kanalspezifische, 294
 NCU-globale, 293

System-, 295
 zuweisen, 282
 Framevariable
 Aufruf von Koordinatentransformationen, 273
 Vordefinierte Framevariable, 275, 281
 Werte zuweisen, 278
 Fräser
 -hilfspunkt (FH), 400
 -spitze (FS), 400
 Fräserformen, 397
 Freistich - CYCLE940
 extern programmieren, 704
 FROM, 531
 FTOCOF, 391
 FTOCON, 391

G

G0-Toleranzfaktor, 474
 G5, 350
 G62, 268
 G621, 268
 G7, 350
 G810 ... G819, 267
 G820 ... G829, 267
 G-Code
 indirekt programmieren, 65
 GEOAX, 561
 Geometrieachse
 umschalten, 561
 Geschwindigkeitskopplung, 513
 GET, 131
 GETACTTD, 414
 GETD, 131
 GETDNO, 413
 GETVARAP, 59
 GETVARDFT, 61
 GETVARLIM, 60
 GETVARPHU, 58
 GETVARTYP, 62
 Gewindebohren mit Ausgleichsfutter - CYCLE840
 extern programmieren, 695
 Gewindebohren ohne Ausgleichsfutter - CYCLE84
 extern programmieren, 658
 Gewindedrehen - CYCLE99
 extern programmieren, 671
 Gewindefräsen - CYCLE70
 extern programmieren, 637
 Gewindegewinde - CYCLE98
 extern programmieren, 667
 G-Gruppe
 Technologie, 448

Glättung
 des Orientierungsverlaufs, 344
 Globaler R-Parameter, 23
 Globaler Teileprogrammspeicher (GDIR), 207
 GOTO, 96
 GOTOB, 96
 GOTOC, 96
 GOTOF, 96
 GOTOS, 95
 GP, 67
 Gravur - CYCLE60
 extern programmieren, 628
 Grundstellung der Werkzeugorientierung
 ORIRESET, 313
 GUD, 27

H

Haltesatz, 457
 High Speed Settings – CYCLE832
 extern programmieren, 692
 Hilfsfunktionen, 550
 HOLES1 - Positionsmuster Linie
 extern programmieren, 614
 HOLES2 - Positionsmuster Kreis
 extern programmieren, 614
 Hubauslösung, 548

I

ID, 531
 IDS, 531
 IF, 108
 IFRAME, 275
 INDEX, 84
 Indirekte Programmierung
 von Adressen, 63
 von G-Codes, 65
 von Positionsattributen, 67
 von Teileprogrammzeilen, 69
 INICF, 27
 INIPO, 27
 INIRE, 27
 INIT, 114
 INITIAL, 216
 INITIAL_INI, 216
 Initialisierung
 von Feldern, 50
 Initialisierungsprogramm, 217
 INT, 27
 Interpolation des Drehvektors, 333

Interruptroutine
 Aus-/Einschalten, 124
 Löschen, 124
 Neu Zuordnen, 123
 Programmierbare Verfahrrichtung, 127
 Rückzugsbewegung, 127
 Schnellabheben von der Kontur, 125
 INTERSEC, 605
 IPOBRKA, 269
 IPOENDA, 269
 IPOSTOP, 510
 IPTRLOCK, 456
 IPTRUNLOCK, 456
 ISAXIS, 559
 ISD, 393, 399
 ISFILE, 144
 ISNUMBER, 80
 ISOCALL, 192
 Istwertkopplung, 513
 ISVAR, 57

J

JERKLIM, 467
 JERKLIMA, 446

K

Kartesisches PTP-Fahren, 305
 Kinematik
 Aufgelöste, 420
 Kinematiktyp, 420
 Kontroll
 -strukturen, 107
 Kontur
 -aufbereitung, 595
 -codierung, 601
 -tabelle, 595, 601
 wieder anfahren, 458
 Konturabspannen - CYCLE95
 extern programmieren, 665
 Konturaufbereitung
 Fehlerrückmeldung, 609
 Konturaufruf - CYCLE62
 extern programmieren, 633
 Konturelement
 abfahren, 606
 Konturgenauigkeit
 programmierbare, 450
 Konturtasche fräsen – CYCLE63
 extern programmieren, 634

Konturtasche vorbohren – CYCLE64
 extern programmieren, 636
Koppelfaktor, 477
Kopplung
 Generische, 520
Kopplungsart, 513
Kopplungsstatus
 bei axialer Leitwertkopplung, 502
 beim Mitschleppen, 480
Korrekturspeicher, 375
Kreisdaten
 berechnen, 607
Kreisnut - SLOT2
 extern programmieren, 624
Kreistasche - POCKET4
 extern programmieren, 619
Kreiszapfen - CYCLE77
 extern programmieren, 645

L

L..., 182
Label, 101
Lagesynchronität, 510
Lagesynchronität mit Winkelversatz, 510
Langloch - LONGHOLE
 extern programmieren, 626
Längsnut - SLOT1
 extern programmieren, 621
Laufwerksname, 211
Laufzeit
 -verhalten von Kontrollstrukturen, 107
LEAD, 314
LEADOF, 498
LEADON, 498
Leitachse
 bei axialer Leitwertkopplung, 498
Leitwertkopplung
 Ist- und Sollwertkopplung, 501
 Synchronisation Leit- und Folgeachse, 500
Leitwertsimulation, 502
LIFTFAST, 125
Link
 -Variablen, 25
LLI, 39
LN, 71
Logische Operatoren, 73
LONGHOLE - Langloch
 extern programmieren, 626
LOOP, 110
LUD, 27

M

M17, 170
M30, 170
Makro, 202
MASLDEF, 528
MASLDEL, 528
MASLOF, 528
MASLOFS, 528
MASLON, 528
MATCH, 84
MAXVAL, 76
MCALL, 187
MD15800, 24
MD18156, 24
MEAC, 257
MEAFRAME, 289
MEAS, 254
MEASA, 257
MEAW, 254
MEAWA, 257
Mehrkant - CYCLE79
 extern programmieren, 649
Messauftragsstatus, 265
MINDEX, 84
MINVAL, 76
Mitschleppachsen, 479
Mitschleppen, 477
Mitschleppverband, 477
MMC, 573
MOD, 71
MODAXVAL, 559
MPF, 206

N

NAMETOINT, 368
NCK, 27
NCK-Notation, 209
Nennerpolynom, 250
NEWCONF, 137
Nibbeln
 aktivieren/deaktivieren, 545
 automatische Wegaufteilung, 550
NOC, 510
NOT, 73
NPROT, 224
NPROTDEF, 221
Nullpunktverschiebung
 Externe Nullpunktverschiebung, 285

NUMBER, 80
NUT, 326

O

OEM-Adressen, 267
OEM-Funktionen, 267
OEMIPO1/2, 267
Offene Nut fräsen - CYCLE899
 extern programmieren, 698
OFFN, 348
OMA1 ... OMA5, 267
Online-Werkzeuglängenkorrektur, 423
OR, 73
ORIXES, 324
ORIC, 406
ORICONCCW, 326
ORICONCW, 326
ORICONIO, 326
ORICONTO, 326
ORICURVE, 329
ORID, 406
Orientierbare Werkzeugträger, 416
Orientierungsachsen, 324
Orientierungsprogrammierung, 324
Orientierungstransformation TRAORI
 Generische 5/6-Achs Transformation, 305
 Maschinenkinematik, 304
 Orientierungsprogrammierung, 313
 Varianten der Orientierungsprogrammierung, 313
 Verfahrbewegungen und
 Orientierungsbewegungen, 303
Orientierungsvektor THETA, 333
ORIEULER, 324
ORIMKS, 322
ORIPATH, 337
ORIPATHS, 337
ORIPANE, 326
ORIRESET(A, B, C), 313
ORIROTA, 333
ORIROTC
 bei Drehung der Werkzeugorientierung, 333
 bei Interpolation der Werkzeugdrehung, 338
ORIROTR, 333
ORIROTT, 333
ORIRPY, 324
ORIRPY2, 324
ORIS, 406
ORISOF, 344
ORISON, 344
ORIVECT, 324
ORIVIRT1, 324

ORIVIRT2, 324
ORIWKS, 322
OS, 533
OSB, 533
OSC, 406
OSCILL, 538
OSCTRL, 533
OSD, 406
OSE, 533
OSNSC, 533
OSOF, 406
OSP1, 533
OSP2, 533
OSS, 406
OSSE, 406
OST, 406
OST1, 533
OST2, 533
OTOL, 471

P

P..., 186
P_ACTFRAME, 297
Parameter
 Aktual-, 152
 Formal-, 152
 -übergabe bei Unterprogrammaufruf, 184
 -übergabe beim Unterprogrammaufruf, 153
 Werkzeug-, 375
PCALL, 192
PDELAYOF, 545
PDELAYON, 545
Pendelbewegung
 Umkehrbereich, 540
 Umkehrpunkt, 540
 Zustellung im Umkehrpunkt, 542
Pendeln
 Asynchrones, 533
 Asynchrones Pendeln, 533
 Synchrones Pendeln, 538
 Teilstellung, 540
 über Synchronaktion steuern, 538
Pfadangabe, 210
PFRAME, 275
PHI
 bei Orientierung entlang einer
 Kegelmantelfläche, 326
 Orientierungspolynome, 332
PHU, 40

- PL
 - bei Polynom-Interpolation, 246
 - bei Spline-Interpolation, 234
- Planfräsen - CYCLE61
 - extern programmieren, 631
- PO, 246
- PO[PHI]
 - bei Drehung der Werkzeugorientierung, 337
 - bei Orientierung entlang einer Kegelmantelfläche, 326
 - Orientierungspolynome, 332
- PO[PSI]
 - bei Drehung der Werkzeugorientierung, 337
 - bei Orientierung entlang einer Kegelmantelfläche, 326
 - Orientierungspolynome, 332
- PO[THT]
 - bei Drehung der Werkzeugorientierung, 337
 - Orientierungspolynome, 332
- PO[XH]
 - bei Orientierungsvorgabe zweier Kontaktpunkte, 329
 - Orientierungspolynome, 332
- PO[YH]
 - bei Orientierungsvorgabe zweier Kontaktpunkte, 329
 - Orientierungspolynome, 332
- PO[ZH]
 - bei Orientierungsvorgabe zweier Kontaktpunkte, 329
 - Orientierungspolynome, 332
- POCKET3 - Rechtecktasche
 - extern programmieren, 616
- POCKET4 - Kreistasche
 - extern programmieren, 619
- Polar-Transformation, 305
- POLF
 - für NC-geführtes Rückziehen, 587
- POLFA, 587
- POLFMASK
 - für NC-geführtes Rückziehen, 587
- POLFMLIN
 - für NC-geführtes Rückziehen, 587
- POLY, 246
- Polynom-Interpolation, 246
- Polynomkoeffizient, 247
- POLYPATH, 246
- PON, 553
- PONS, 545
- POSFS, 510
- Positionsattribute
 - indirekt programmieren, 67
- Positionsmuster Gitter/Rahmen - CYCLE801
 - extern programmieren, 681
- Positionsmuster Kreis - HOLES2
 - extern programmieren, 614
- Positionsmuster Linie - HOLES1
 - extern programmieren, 614
- POT, 71
- PREPRO, 169
- PRESETON, 287
- PRESETONS, 288
- PRIO, 122
- PRLOC, 27
- Process DataShare, 580
- Profilieren - CYCLE495
 - extern programmieren, 676
- Programm
 - adressierung, 209
 - Initialisierungs-, 217
 - koordinierung, 114
 - laufzeiten, 575
 - speicher, 206
 - sprünge, 96
 - verzweigung, 99
 - wiederholung, 186
- Programmschleife
 - Endschleife, 110
 - IF-Schleife, 108
 - REPEAT-Schleife, 112
 - WHILE-Schleife, 112
 - Zählschleife, 110
- Programmspeicher
 - Dateitypen, 206
 - Standard-Verzeichnisse, 206
- Programmteil
 - wiederholung, 101
- Programmteilwiederholung
 - mit indirekter Programmierung CALL, 190
- PROTA, 370
- PROTD, 372
- PROTS, 371
- PSI
 - bei Orientierung entlang einer Kegelmantelfläche, 326
 - Orientierungspolynome, 332
- PUD, 27
- PUNCHACC, 545
- PUTFTOC, 391
- PUTFTOCF, 390
- PW, 234

R

R..., 22
 Randbedingungen bei Transformationen, 362
 READ, 142
 REAL, 27
 Rechenparameter
 -nummer n, 22
 Rechenparameter (R), 22
 Rechtecktasche - POCKET3
 extern programmieren, 616
 Rechteckzapfen - CYCLE76
 extern programmieren, 643
 REDEF, 33
 Reiben - CYCLE85
 extern programmieren, 661
 RELEASE, 131
 REP, 50
 REPEAT, 101
 REPEATB, 101
 REPOSA, 458
 REPOSH, 458
 REPOSHA, 458
 REPOSL, 458
 REPOSQ, 458
 REPOSQA, 458
 Restweglöschen, 263
 Restzeit
 für ein Werkstück, 578
 RET, 171
 RET (parametrierbar), 172
 RETB (parametrierbar), 178
 RG, 23
 Richtungsvektor, 319
 RINDEX, 84
 RMBBL, 458
 RMEBL, 458
 RMIBL, 458
 RMNBL, 458
 ROUND, 71
 ROUNDUP, 147
 R-Parameter
 Globaler, 23
 RPY-Winkel, 318
 Ruck
 -begrenzung, 444
 -korrektur, 467
 Rückziehen
 antriebsautarkes, 592
 NC-geführtes, 587

S

Satzanzeige, 192
 unterdrücken, 166
 SAVE, 159
 SBLOF, 161
 SBLON, 161
 Schachtelungstiefe
 von Kontrollstrukturen, 107
 Schneidenummer, 412
 Schnellabheben von der Kontur, 125
 Schräge Achse (TRAANG), 349
 Schrägeinsteichschleifen, 350
 Schutz
 -bereiche, 221
 Schwenken – CYCLE800
 extern programmieren, 678
 SCPARA, 568
 SD, 234
 SD42475, 342
 SD42476, 342
 SD42477, 342
 SD42678, 344
 SD42680, 344
 SD42900, 383
 SD42910, 383
 SD42920, 383
 SD42930, 384
 SD42935, 386
 SD42940, 387
 SD42984, 428
 Seitwärtswinkel, 316
 SET, 50
 SETAL, 584
 SETDNO, 413
 SETINT, 122
 SETM, 114
 SIN, 71
 Singuläre Stellen, 323
 SLOT1 - Längsnut
 extern programmieren, 621
 SLOT2 - Kreisnut
 extern programmieren, 624
 SOFT, 444
 SOFTA, 444
 Sollwertkopplung, 513
 SON, 545
 SONS, 545
 Speicher
 Arbeits-, 216

Programm-, 205
 Vorlauf-, 452
 SPF, 206
 SPI, 559
 SPIF1, 545
 SPIF2, 545
 Spindel
 -tausch, 131
 Spline
 -Interpolation, 234
 -Typen, 240
 SPLINEPATH, 244
 Spline-Verbund, 244
 SPN, 550
 SPOF, 545
 SPP, 550
 SPRINT, 87
 Sprung
 auf Programmanfang, 95
 auf Sprungmarken, 96
 Sprungmarke
 bei Programmsprüngen, 97
 bei Programmteiwiederholungen, 101
 SQRT, 71
 Stanzen
 aktivieren/deaktivieren, 545
 automatische Wegaufteilung, 550
 START, 114
 STARTFIFO, 452
 STAT, 354
 Stechen - CYCLE952
 extern programmieren, 709
 Stillsetzen
 antriebsautarkes, 591
 NC-geführtes, 591
 Stirnfräsen, 321
 STOLF, 474
 STOPFIFO, 452
 STOPRE, 452
 String
 formatieren, 87
 -länge, 83
 -operationen, 79
 -Verkettung, 81
 STRING, 27
 STRINGIS, 570
 STRLEN, 83
 SUBSTR, 85
 Suchpfad
 bei Unterprogrammaufruf, 214
 Programmierbarer Suchpfad, 193
 Suchunfähige Bereiche erfassen und suchen, 457

Synchrones Pendeln
 Auswertung IPO-Takt, 543
 Nächste Teilzustellung, 544
 Synchronaktionen, 541
 Zuordnung von Pendel- und Zustellachse, 541
 Zustellbewegung, 542
 Zustellung im Umkehrbereich, 542
 Zustellungen festlegen, 541
 Synchronlauf
 fein, 513
 grob, 513
 Synchronspindel
 Kopplung, 509
 -paar festlegen, 515
 SYN, 27
 SYN, 27
 SYN, 27
 SYN, 27
 System
 -abhängige Verfügbarkeit, 5
 Systemframes, 295
 Systemvariablen
 Messtasterbegrenzung, 265
 Messtasterstatus, 264

T

TAN, 71
 TANG, 433
 TANGDEL, 437
 TANGOF, 437
 TANGON, 436
 TCARR, 421
 TCOABS, 421
 TCOFR, 421
 TCOFRX, 421
 TCOFRY, 421
 TCOFRZ, 421
 Teilstrecken - automatische Wegaufteilung, 550
 THETA
 bei Drehung der Werkzeugorientierung, 333
 bei Interpolation der Werkzeugdrehung, 338
 Tieflochbohren - CYCLE83
 extern programmieren, 655
 Tieflochbohren 2 - CYCLE830
 extern programmieren, 686
 TILT, 314
 TLIFT, 434
 TMOF, 557
 TMON, 557
 TOFFOF, 424
 TOFFON, 424
 Toleranz";"bei G0, 474

TOWER, 83
 TOUPPER, 83
 TOWBCS, 384
 TOWKCS, 384
 TOWMCS, 384
 TOWSTD, 384
 TOWTCS, 384
 TOWWCS, 384
 TRAANG, 349
 TRACON, 351
 TRACYL, 346
 TRAFEOF, 363
 TRAILOF, 477
 TRAILON, 477
 Transformation mit schwenkbarer Linearachse, 310
 Transformationen

- Drei- Vier- und Fünf- Achstransformation, 311
- Kinematikunabhängige Grundstellung der Werkzeugorientierung, 302
- Kinematische Transformationen, 302
- Orientierungstransformation, 301
- verkettete, 351
- Verkettete Transformationen, 303

 Transformationsarten

- Allgemeine Funktion, 301

 TRANSMIT, 346
 TRAORI, 311
 Trigger-Ereignis

- beim Messen, 261

 TRUE, 27
 TRUNC, 71
 TU, 357

U

ULI, 39
 Umfangsfräsen

- mit Begrenzungsflächen, 402

 Umkehr

- punkt, 538

 Umschaltbare Geometrieachsen, 561
 Unterprogramm

- Anwendung, 149
- aufruf indirekt, 189
- aufruf mit Parameterübergabe, 184
- aufruf modal, 187
- aufruf ohne Parameterübergabe, 182
- name, 150
- Programmierbarer Suchpfad, 193
- rücksprung parametrierbar, 172, 178
- wiederholung, 186

Unterprogrammaufruf mit Pfadangabe und Parametern, 192
 UNTIL, 112

V

Variable

- Typenkonvertierung, 78

 Variablen

- anwenderdefiniert, 27
- Typkonvertierung, 80

 VELOLIM, 468
 VELOLIMA, 446
 Verfügbarkeit

- System-abhängige, 5

 Vergleichsoperatoren, 73
 Verkettung

- von Strings, 81

 Verschleißwert, 379
 Verzeichnispfad, 212
 Voreilwinkel, 316
 Vorlauf

- speicher, 452

W

WAITC, 510
 WAITE, 114
 WAITENC, 567
 WAITM, 114
 Wegaufteilung, 554
 Wegaufteilung bei Bahnachsen, 552
 Werkstück

- Hauptverzeichnis, 206
- verzeichnisse, 206
- zähler, 579

 Werkzeug

- längenkorrektur, 421
- orientierung, 314, 406
- orientierung bei Framewechsel, 423
- orientierung Glättung, 344
- parameter, 375
- radiuskorrektur, 381

 Werkzeugkorrektur

- Koordinatensystem für Verschleißwerte, 384
- Korrekturspeicher, 375

 Werkzeugkorrekturen

- additive, 378

 Werkzeugorientierung

- bahnrelativ, 335

Werkzeugradiuskorrektur
 Eckenverzögerung, 268
Werkzeugträger
 -kinematik, 416
 -Orientierbare, 421
WHEN, 531
WHEN-DO, 541
WHENEVER, 531
WHENEVER-DO, 541
WHILE, 112
WRITE, 138

X

xe ye ze, 329
XH YH ZH, 329
xi yi zi, 329
XOR, 73

Z

Zählschleife, 110
Zentrieren - CYCLE81
 extern programmieren, 651
Zustell
 -achse, 539
Zyklenalarme, 585
Zylindermanteltransformation, 305